

Environment analyzing rover based on Raspberry Pi – construction and analysis

Rohan Tomar

Bachelor in Technology (Computer Engineering), Department of Computer Science & Engineering, Delhi Technological University, Shahbad Daultapur, Main Bawana road, Delhi-110042, India
rohantomar98@gmail.com

Abstract

Background: This study is to help reader to understand Robotics by developing a robotic system which is practically useful by collecting, generating environmental quantities and analyzing it.

Methods: Chassis can be made in CAD format and wiring is tested beforehand. Constraint in Pi is limited GPIO, so we use two Pi's for different uses. Accessing possible via wireless system, SSH, VNC or port forwarding. Calculation of speed, rotation time done. We also deduce the relation between the distance travelled and the temperature of the CPU of the Raspberry Pi. We also compare and suggest best cooling solutions.

Findings: We can construct a rover which can detect the environmental quantities like temperature, human presence, animal presence, humidity etc. with the help of two raspberry pi's and an H-bridge, which can be accessed over the internet and the live stream can be obtained on a remote screen. The rover we construct can travel on the speed of 0.18 meter per second. It will take nearly 1.10 seconds to turn at a 90 degree rotation and double of that (2.20 seconds) for a 180 degree rotation. When we perform temperature analysis we find out that temperature almost linearly increases with the distance travelled, and the base temperature of the raspberry pi can be reduced by adding the cooling solutions like small heat sink, fan and larger heats ink. Equations are deduced using method of regression. We conclude that a bigger heats ink, enough to cover the CPU can be the best method to cool a rover which is based on raspberry pi as its single board computer.

Applications: Detection of environmental quantities to detect problem/hazard/performing disaster mitigation is the application of the rover and the paper would also help in developing cooling solutions for the rover further.

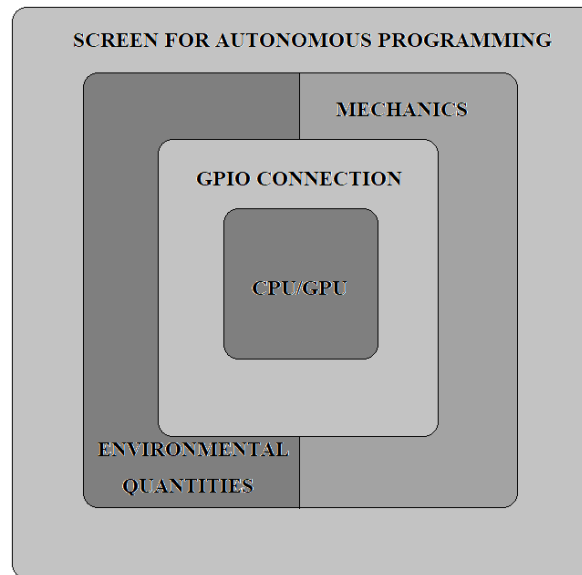
Keyword: Raspberry Pi, Robotics, Single board computer, Environment, Sensors, Analyzer

1. Introduction

The aim of this particular study is to help reader to understand robotics by developing a robotic system which is practically useful in real life by collecting and generating real life quantities and further analyzing it on speed, data, and temperature and suggest best cooling solutions by deducing some relation on distance and temperature. This study presents a variety of concepts on the synthesis of Single Board Computers and Robotics. Single board computers are highly useful as we can program them to a high degree and perform a large amount of functions with simple addition of modules. We here attempting to learn about robotics in aspects like from where to start, how to program, and knowledge about rover chassis. Here we are beginning with something called a 'Single Board Computer'. A single-board computer is, as indicated, a computer, but contains memory, and other functions of a computer like microprocessor. The not-so-advanced architecture of a single board computer provides low power but on the contrary also provides input and output [1]. We ourselves can perceive a Single Board Computer or a S.B.C. as compact computer which may max out to its potential in field you require. We may look at various types of Single Board Computers. Raspberry Pi, Asus Tinker board, Latte Panda, Rock Pro 64, ODROID XU4 etc. All of these S.B.C.'s may be used for various kinds of purposes but most importantly they can be used for Robotics. Now we attempt to define robotics: Robotics is a branch of technology which deals with robots. Robots are programmable machines which are usually able to carry out a series of actions autonomously, or semi-autonomously. Autonomous / Semi-Autonomous behavior, Using Sensors / actuators to establish a link between the robot and the physical world and programmability are majorly three functions of a robot. In [2] this study we establish the link between Robotics and Single Board Computers.

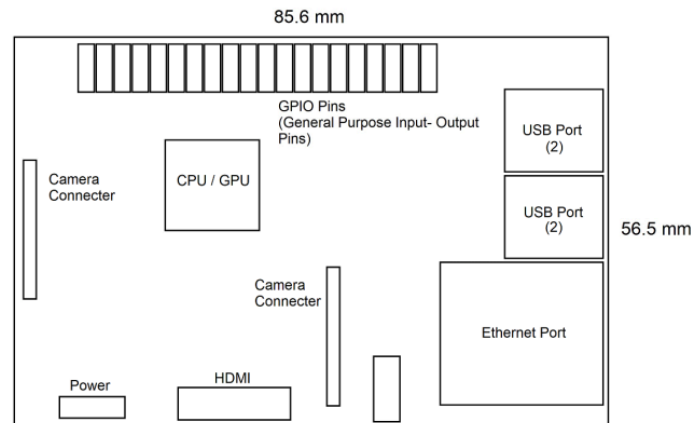
On the technical aspect of it, we need to look at the structure of a Single Board Computer and how exactly we plan to turn this into some system with Robotic facilities. Out of the many S.B.C.'s (Single Board Computer) available in the market right now we plan to use the Raspberry Pi. We now look at the Raspberry Pi. Raspberry Pi Foundation defines it as a low cost, a small sized fully fledged computer that can work with screen like television or a monitor and has complete functions of a computer in a small package. Apart from providing some abstract features like enabled learning of computing for all ages, the most amazing thing about the Raspberry Pi is, it provides all the features a normal computer would provide from saving files, programming, photos, videos, text documents, playing games to surfing the internet. (<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>). Now we look at the technical structure of the Raspberry Pi. We here try to sketch a rough diagram (Figure 1) of our Pi. (Here we are considering The Raspberry Pi Model 3 B). The Rover we intend to construct is based on this Raspberry Pi for basic functions like mechanics and complex functions like measuring temperature, humidity etc. We can simply work on the raspberry pi to enable these functions using simple connection and various lines of code we write.

Figure 1. The schematic representation of the rover's components



(This is just a conceptual and abstract portrayal of the working principles of the E.A.R. The most basic level of this vehicle is the CPU/GPU. This requires basic maintenance like cooling. Then the most important is the GPIO connection, then the functions assigned to the two raspberry pi's that are mechanics and the collection of quantities. The collection of quantities means the obtaining of the environmental quantities like temperature, humidity, water etc. And in the end a screen at the very outside for the autonomous programming of the E.A.R)

Figure 2. Raspberry Pi



(This is just a basic pictorial representation of a Raspberry Pi mentioning its major components in the correct places. This is not very much technical, but it's the most efficient, simple and nicest way possible to understand the simplicity of the Raspberry Pi.)

In the Figure 1 we have attempted to explain to sketch the schematic representation for the rover and the central ideas on it. The components to the inner sides are things like the CPU and the connections and the outer side will obviously be the screen to program the Raspberry Pi. In the Figure 2 we have sketched a simple sketch for the raspberry pi for the easy understanding of the board as directly looking at the board may confuse the people who are not so technical and hence hinder the aim of this paper of being simple and informative. Here we completed a schematic sketch of the Pi and now we are able to understand the working of the single board computers. With some knowledge and working we can convert this simple looking device into a robotic functionality. This will act as the central unit of our E.A.R. (Environment Analyzing Rover) for the very basic functions like motion. Although for justifying the name of this Rover we need the help of many other functions like live streaming and Detection of quantities.

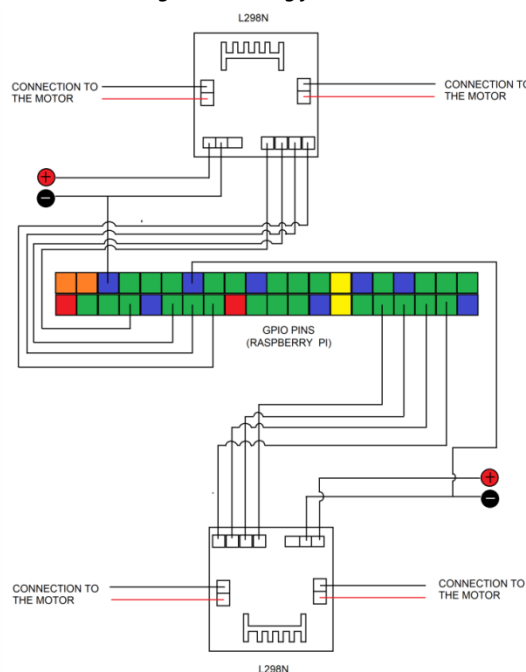
1.1. Operating system

The Pi uses an Operating System called Raspbian. Raspbian is the Foundation's official supported operating system. Raspbian comes pre-installed with plenty of software for education, programming and general use. It has Python, Scratch, Sonic Pi, Java and more [3]. We simply have a download a copy of the Operating System from the link (<https://www.raspberrypi.org/downloads/raspbian/>, 2018), mount it to a SD Card, and mount the SD Card to back of the Pi, Connect a display to the Pi through HDMI port and install the Operating System.

2. Adding mechanics to control raspberry pi

To add mechanics and any kind of motion to the Raspberry Pi we use Motor Controller. This device will be used here in this particular paper and Robotic Functionality development as a primary method to induce motion in the scenario. This will easily be workable with our Pi. The motor controller will be connected to the GPIO Pins of the Raspberry Pi through jumper wires. The Motor controller is programmable to a great extent hence we may add our code to run the motor controller in our fashion. A Motor Controller is a device that acts as intermediary between your robot's microcontroller, batteries and motors. A motor controller is necessary because a microcontroller can usually only provide roughly 0.1 Amps of current whereas most actuators (DC motors, DC gear motors, servo motors etc.) require several Amps [4]. Here we attempt to connect the Pi to two motor controllers. Through these we will be able to connect a total of four motors to the Raspberry Pi, and be able to control these with the ability of simple programming. The motor controller we use here is called the L298N motor controller.

Figure 3. Wiring for motion



This is just a simple wiring diagram representation of how the motor controller and the Raspberry Pi interact with each other physically

The following circuit diagram (Figure3) demonstrates the connection between the Raspberry Pi and the L298n Motor controller. Figure3 demonstrates a qualitative and simple way to set a link between the motor controller and the raspberry pi. Now we have successfully added mechanics to the Raspberry Pi and the motor controllers are connected to the Raspberry Pi. What we mean here by “SUCCESSFULLY CONNECTED” is the fact that we are in the position to program the Pi in order to control the motor controllers. We will control the controllers with simply are keyboard which can be connected to Pi. We will control the GPIO Pin via the code we write through which the motors can be controlled. We write the following code in the program specified as “idle” in the Raspberry Pi.

2.1. Algorithm (Motion)

1. Import the module for Raspberry Pi in Python, Time and Curses.
2. Set up the General purpose Input Output pins for the Pins which are involved in the connection as shown in Figure (1) using the commands like GPIO. Setup (<Number of the pin involved in the connection>, GPIO.OUT). And set up the keys and enable GPIO pins for them from statements like
3. Set up the Stop and various motion keys using curses (c break) for controlling by switching on and off the pins.

char=screen.getch()	if char==ord('q'):	break
elif char==curses.KEY_UP:	GPIO.output (7, False)	GPIO.output (11, True)
GPIO.output (13, False)	GPIO.output (15, True)	GPIO.output (31, False)
GPIO.output (33, True)	GPIO.output (35, False)	GPIO.output (37, True)
elif char==curses.KEY_DOWN:	GPIO.output (7, True)	GPIO.output (11, False)
GPIO.output(13,True)	GPIO.output(15,False)	GPIO.output(31,True)
GPIO.output(33,False)	GPIO.output(35,True)	GPIO.output(37,False)
elif char==curses.KEY_RIGHT:	GPIO.output(7,True)	GPIO.output(11,False)
GPIO.output(13,False)	GPIO.output(15,False)	GPIO.output(31,True)
GPIO.output(33,False)	GPIO.output(35,False)	GPIO.output(37,False)
elif char==curses.KEY_LEFT:	GPIO.output(7,False)	GPIO.output(11,False)
GPIO.output(13,False)	GPIO.output(15,True)	GPIO.output(31,False)
GPIO.output(33,False)	GPIO.output(35,False)	GPIO.output(37,True)
elif char==10:	GPIO.output(7,False)	GPIO.output(11,False)
GPIO.output(13,False)	GPIO.output(15,False)	GPIO.output(31,False)
GPIO.output(33,False)	GPIO.output(35,False)	GPIO.output(37,False)

3. Chassis construction

We have to create a body for the vehicle to fit a suitable amount of circuit for on it. We have to use designing software available in use to create it and we plan to attach two raspberry pis on it. Basically the structure or the frame of the rover is being referred to as chassis here [5]. We used an open source 3D CAD modeler. In this we design our software. Free CAD can be defined as 3D CAD application for modeling physical structures. We may use this application for a wide array of functions wherever the modeling may be required despite the fact the application was firstly used for mechanical purposes [6]. Through the advancements made in the designing communities, designing and making quick custom structures is no big deal now. Hence we have completed our work here.

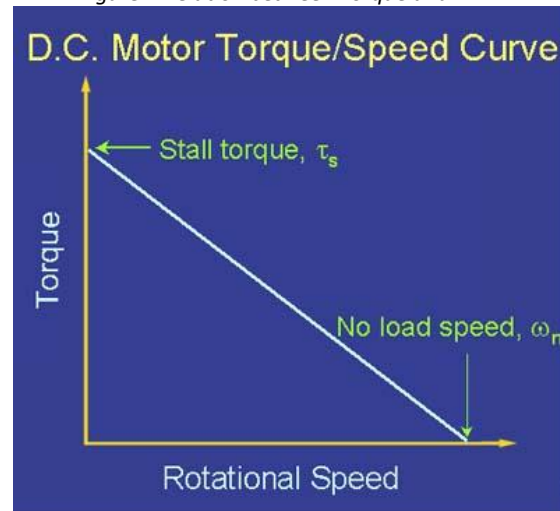
4. Assessing motors and torque

We have to assess the torque and the rotations per minute of the motor we attach to it so that it would be able to carry the load of our rover we place on it. We need enough torque to carry forward the payload and rotations per minute of the motor we attach on it. Torque as we know it, is the force responsible for rotation. The word “Torque” comes from Latin vocabulary and initially means “To twist”. Regular application of twisting likes screwing a bolt with a screwdriver or using a wrench, with our intention being to twist something, we are applying the vector called torque. The SI unit of torque is the newton meter (N+m) [7]. To pull our rover forward we do need enough torque but we do need some good rpm also.

1. Relation between torque and RPM

DC motors, as we have used, have two main characteristics in its functioning which are Torque and Rotations per Minute. If we have to completely understand the working of DC motors, we have to understand the characteristic curves. For every motor, there is a specific Torque/Speed curve and Power curve. Graph in Figure 4 gives a pretty good idea about how the two quantities are related. Clearly from this we conclude the they have a very inverse relation. As the torque increases, the rpm decreases and vice versa [8]. We can further this by the relation defined below:

Figure4. Relation between Torque and RPM



(This graph shows the Graphical representation between torque and rpm)

$$HP = \text{Torque} \times \text{RPM} \div 5252$$

At 5252 rpm we get the equivalency between power and torque but we are clearly operating much below than that at 60 rpm. So for our range torque and rpm will have an inverse relation [9].

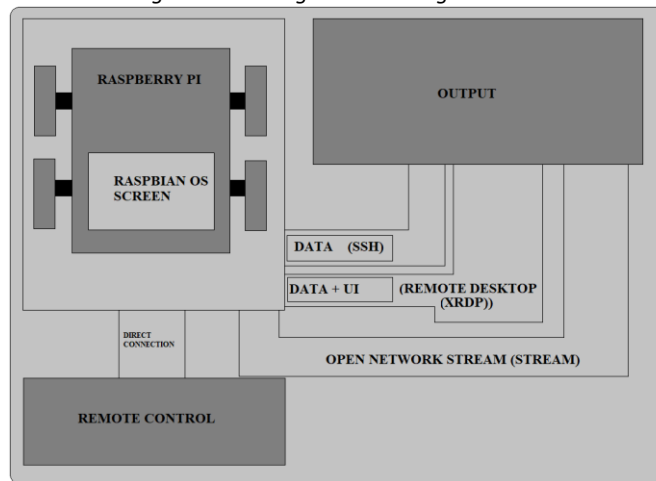
2. Choosing motors

We have to choose motors accordingly to the chassis and the load on it. We first took the four DC motors of 200 rpm. When we attached the four motors of this rpm the results were unsuccessful as the motors were unable to pull the rover. When will lift the rover in our hands the motors ran successfully. We have to choose the motors accordingly, so by the concept derived by the relation between the torque and the rotations per minute, we lower the rpm of the motors and try it. We now try the motors with 60 rpm labeling and we intend to get results. This time we get successful results as the motors produce enough torque to pull the load on it and the rover is successfully working. Now when we lift the rover off the ground, the motors work just as successful as when not lifted off the ground. So we have made sure that 60 rpm labeled motors will be perfect for the construction of this rover.

5. Attempt to access control with laptop and adding remote log in protocol

The Figure 5 gives us an idea about the working of the model and the methods to access it. This is schema of the methods to access the rover in certain ways. We here try to control this with our laptops now. For this a very important concept called "Secure Shell" or commonly known as "SSH" should be studied. The SSH protocol (also referred to as Secure Shell) is a method for secure remote login from one computer to another. Apart from providing some other options for solid authentication, it has strong encryption features to protect communication security and integrity [10]. Before understanding what encryption is we need to take a very short, brief look at what cryptography is? The concept of encryption is included in the concept of cryptography. Cryptography, the word was given by the Greeks and initially meant "Secret Writing". However recently it has been used to refer the concept of making things more secure and making our institutions less prone to attacks [11].

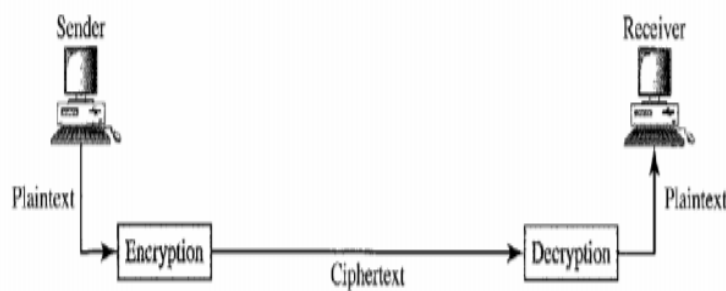
Figure 5. Working and accessing the rover



(This is an extensive pictorial description of the working. We can connect to the output of the Raspberry Pi's using SSH which will help us to access the terminal of the Raspbian OS. We may use Remote desktop connection to access GUI. To live stream we may use VLC player and the camera attached to the E.A.R. We simply can access the E.A.R. using a regular keyboard.)

The Figure6 gives the information about the receiving and sending via performing encryption, and what cipher text and plaintext actually are. The job of an algorithm called "encryption algorithm" is to transform the plaintext into cipher text and similarly the job of a decryption algorithm is to transform the cipher text back into plaintext. The sender uses an encryption algorithm, and the receiver uses a decryption algorithm. To encrypt a message, we need an encryption algorithm, an encryption key, and the plaintext. These create the cipher text. To decrypt a message, we need a decryption algorithm, a decryption key, and the cipher text. These reveal the original plaintext. We have now studied what encryption and what cryptography is. Now we take a deeper look at Secure Shell (SSH). SSH is now established as secured, encrypted, network communication protocol which will provide data communication. SSH not only indicates to the cryptographic network protocol but also to the suite of utilities that implement the particular protocol. Client – Server Model is a highlight of the Secure Shell. It helps connecting a secure shell client application, the end at which the session is displayed, with an SSH server, the end at which the session runs [12]. To access the Raspberry Pi via SSH, we use Putty. Putty is software which will easily implement SSH for our rover.

Figure 6. Data encryption



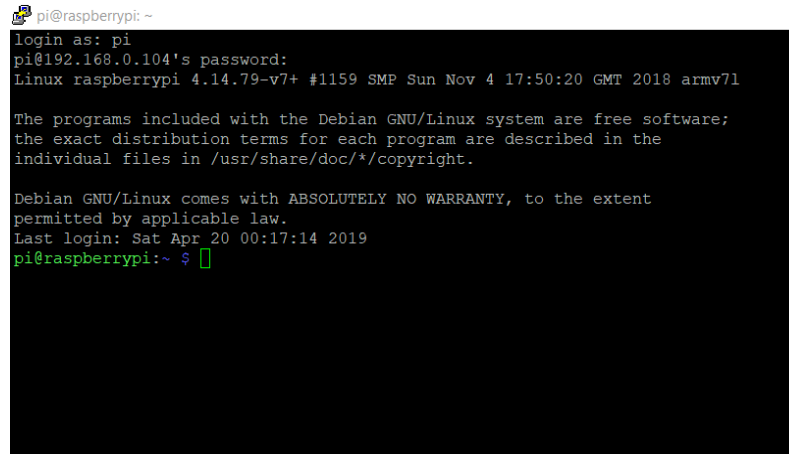
This figure here depicts the concept of encryption and decryption with showing what plaintext and cipher text actually is

Simon Tatham developed the Putty which is basically an SSH and Telnet Client, especially for windows platform. Putty is open source software that is available with source code and is developed and supported by a group of volunteers (This is free and open-source.) This was developed by Simon Tatham and is easily downloadable on <http://www.putty.org>. This is how we log on to the rover with help of putty [13].

1. Connect Rover to a display, log on to the Raspbian desktop and open terminal.
2. Type in a command if config in the terminal and hit enter. And in addition, we have to go into the raspberry pi configuration and we just have to enable the SSH in this section.

3. Now you will be shown two sections here. One for the group of people chooses to connect to the internet via Ethernet and one for Wi-Fi. We might obtain the IP address of the rover connected to the Internet. Figure 7 is the screenshot of our rover's Pi when we have logged in it via Secure Shell. Here we are able to see the terminal of that Pi. We have now found the IP address of the rover, and we can now use putty.
4. Open Putty and enter the IP addresses we found and hit enter. Now you will be logged on to a black screen and demanding your credentials to log on to the Pi. Mostly, username is "Pi" and password is "password". After entering your credentials you will be in the terminal of your rover and you may just use your laptop to browse through terminal of the rover.
5. Run the algorithm using the terminal we gained access through Putty. We have successfully gained access to control our rover through our Laptop. Now Arrow Keys are just as good as Remote Control.

Figure 7. Putty accessed Terminal



```

pi@raspberrypi: ~
login as: pi
pi@192.168.0.104's password:
Linux raspberrypi 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
last login: Sat Apr 20 00:17:14 2019
pi@raspberrypi:~ $

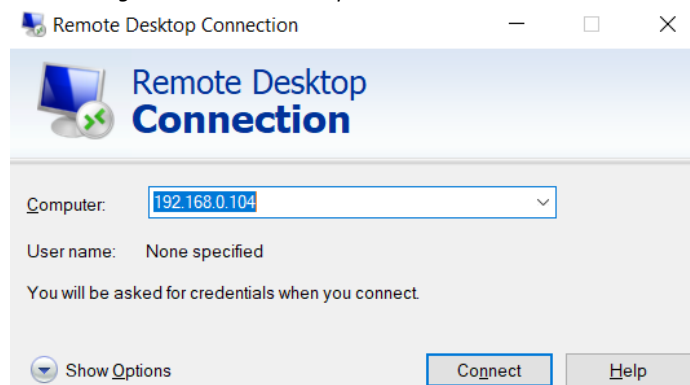
```

(This is the terminal we have accessed the from the IP address of the PI and putty.)

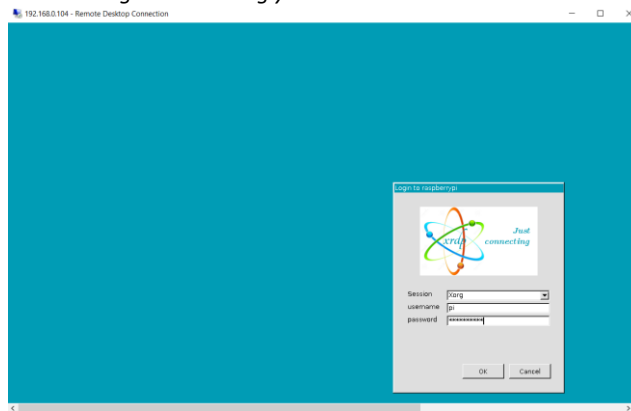
1. Access the graphical user interface (Gui) of the Raspbian from another screen

This is done through the Xrdp and remote desktop connection. Xrdp provides a graphical login to remote machines using RDP (Microsoft Remote Desktop Protocol). xrdp accepts connections from variety of RDP clients: Free RDP, rdesktop, NeutrinoRDP and Microsoft Remote Desktop Client (for Windows, macOS, iOS and Android). Remote desktop is a program or an operating system feature that allows a user to connect to a computer in another location, see that computer's desktop and interact with it as if it were local. Just like the previous section, get the IP address of the raspberry pi, and open "Remote Desktop Connection". Enter the IP address in this bar displayed here and you will be directed to the Gui of the Raspbian Desktop. From here we may just as well open up our terminal and run the code for the working of the rover. Figures 8-10 are the screenshots of the time when we have logged in Pi and we have proved to be successful in accessing the Graphical User Interface of the Pi. Figure 10 is the screenshot of the final step when we have opened the Pi's desktop on our desktop. The various methods to access will help us generate the data of environment as we intend like temperature etc [14-15].

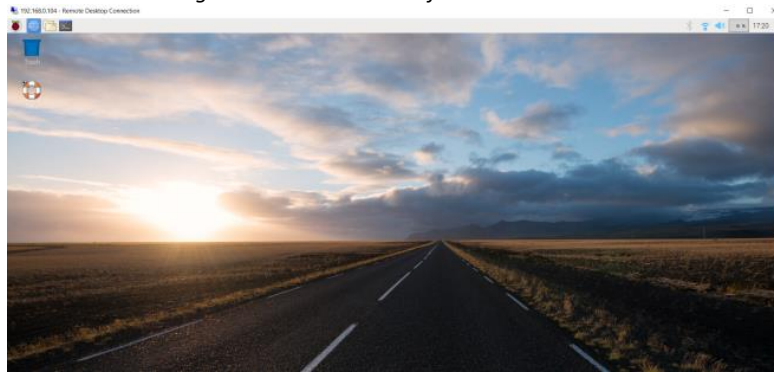
Figure 8. Remote desktop connection in windows OS



This is the window which asks for the IP of the PI to access the GUI of the PI

Figure9. Entering your credentials in windows

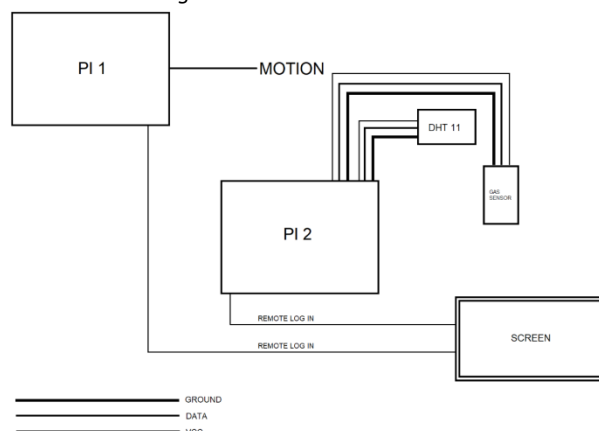
This is the window which asks for the credentials of the PI for accessing the GUI of the Pi

Figure 10. Accessed GUI of the Pi in windows

This is the accessed GUI of the Pi after using remote desktop connection on windows

2. Adding modules and another raspberry pi

Here we are trying to increase and max out potential of our rover. Here we will take the help of modules which can be easily used as the attachments of the Raspberry Pi. When we say “Modules” we are talking about small independent circuits. These can be easily programmed and we may obtain useful data which can be of help to a great extend to us. We try to enable capabilities like detection of temperature, gas levels, Water detection, Sound detection and Human detection to our rover. We are trying to make our rover more capable on its own which can be accomplished by the help if modules. We are here using one more Raspberry Pi and one extension board for the Pi to attach the Modules, because the GPIO pins of the first Pi are already occupied to a great extend.

Figure11. Module connection

(We only have shown the schematic wiring for only two sensors here, but the wiring f the rest of the sensors is really very similar to these, so these should fulfill the appetite of knowing how we attach the modules to the raspberry pi. And as shown and discussed above, remote log in is used to fetch data from our Pi.)

A rover enabled to measure quantities like temperature and Intelligence like detection of Human and things we talked earlier on may go a long way to help us understand robotics. Let's talk about the Modules we have added to our rover to enable the capabilities. DHT11, Sound Sensor Module, Vibration Sensor Module, HC-SR501 PIR Motion Sensor Detector Module, Water detection Module, MQ7 Module, MQ5 Module are some of the modules which we connect to another Raspberry Pi. We can get the data from these modules via remote log in techniques. Here is an example of fetching data via rover. With a very simple diagram we can get an idea of wiring. Figure 11 is an attempt to simplify the connections of the Different Modules present and the Raspberry Pi.

Figure 12. An example of fetching sound detection

```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.105's password:
Linux raspberrypi 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 23 18:46:31 2019 from 192.168.0.106
pi@raspberrypi:~$ dir
Desktop      Pictures          SW420_Vibration_sensor_code.py
DHT11_Python PIR.py           Templates
Documents    Public            thinclient_drives
Downloads    Raspberry-Pi-Gas-Sensor-MQ Third_PIR_Sensor.py
Gas_Sensor.py samplecode1.py   Videos
MagPi        Sample_Water_code.py Wrong_Water_code.py
MQ7.py       Second_PIR_code.py
Music        Sound_Sensor.py

pi@raspberrypi:~$
pi@raspberrypi:~$ python Sound_Sensor.py
Sound Detected
Sound Detected
Sound Detected
Sound Detected
Sound Detected
Sound Detected
Sound Detected
Sound Detected
```

This is the screen of the terminal of the rover when we try to detect any sound in the area. We just access the Pi using method mentioned and "Sound detected" is the message it displays when any sound is detected. Just as a reminder this is the work done on second raspberry pi.

Figure 13. An example of result of PIR sensor

```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.105's password:
Linux raspberrypi 4.14.79-v7+ #1159 SMP Sun Nov 4 17:50:20 GMT 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 23 18:48:46 2019 from 192.168.0.106
pi@raspberrypi:~$ dir
Desktop      Pictures          SW420_Vibration_sensor_code.py
DHT11_Python PIR.py           Templates
Documents    Public            thinclient_drives
Downloads    Raspberry-Pi-Gas-Sensor-MQ Third_PIR_Sensor.py
Gas_Sensor.py samplecode1.py   Videos
MagPi        Sample_Water_code.py Wrong_Water_code.py
MQ7.py       Second_PIR_code.py
Music        Sound_Sensor.py

pi@raspberrypi:~$ python Third_PIR_Sensor.py
Sensor is HIGH
Sensor is LOW
Sensor is HIGH
Sensor is LOW
Sensor is HIGH
Sensor is LOW
Sensor is HIGH
Sensor is LOW
```

This is the screen of the terminal of the rover when we try to detect any Human/Animal Presence in the area with the PIR sensor. We just access the Pi using method mentioned and "Sensor is High" is the message it displays when any presence is detected. Just as a reminder this is the work done on second raspberry pi.

In this diagram, we have tried to represent the connections between the GPIO pins of the raspberry Pi and the pins present on the modules mentioned above. Figure 12 is the working representation of the modules where we have enabled and run our sound sensor and it has successfully detected the sounds in the area. Next in the Figure 13 is the working screenshot of the PIR sensor which can detect humans or animals. It goes "high" when the human life is in proximity to it. It was able to detect a hand of a person which is shown.

6. Speed analysis

We are here trying to analyze the speed analysis of the rover we build so far. We will try out the rover in different paths and try to calculate speed of this S.B.C (Raspberry Pi) based rover. We test the rover in a path of straight line. The line is of 1.5 meters. The rover crosses the mark in about 7.93 seconds. In 1 meter the time taken is approx $7.93/1.5 = 5.2867$ seconds. So we can conclude that from 60 rpm dc motors (four) can provide a speed of 0.18 m/sec. It takes approximately same time to get back. The weight of the rover was calculated about to be Grams.

Time for rotation (90 degree) = 1.1 seconds

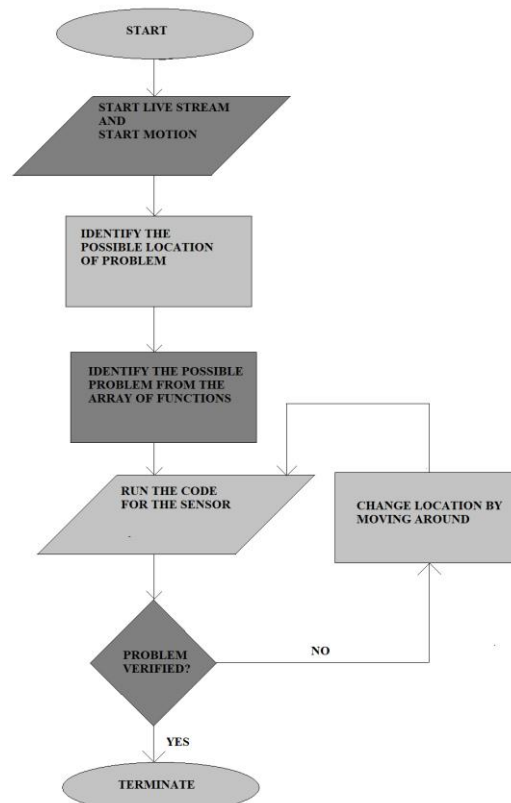
Time for rotation (180 degree) = 2.2 seconds

From this data we can calculate the data for whatever path we want. We just have to add the time of rotations and the path travelled by the rover.

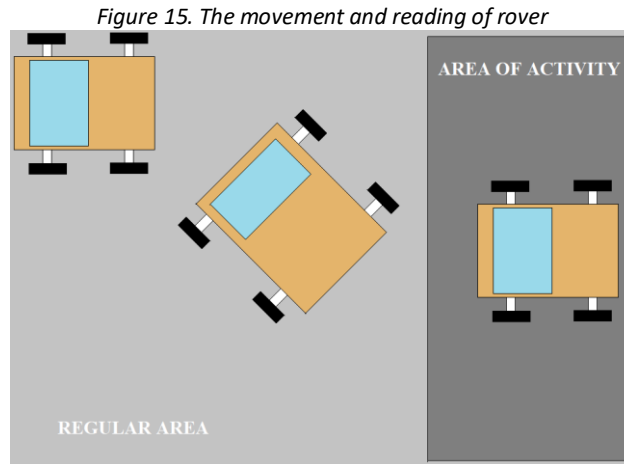
7. Data analysis

This venture would obviously produce a large amount of data. We have to create some mechanism for storing data and making valuable data quantities like temperature secure. Simply we may put the command `command>outputfilenmae.txt` and the output of the terminal will be saved on the Computer in a Text file. Figure 14 is an abstract algorithm for the sequence of working of the rover and shows how to operate it. This very much simplified and can be easily read. Figure 15 is depicts the movement of the rover and how it can take readings in various areas and verify and problem/leak/hazard which we may intend.

Figure 14. Flowchart for using the rover



(This is an abstract representation in form of a flowchart of how we take data from the vehicle and how we verify a reading)



(This is a diagram of the E.A.R. moves. We can take a reading in the regular area and if the reading is not identified, we may move around and eventually enter the affected area and the reading will be verified.)

8. Heat analysis

Obviously working of Pi's to such a great extent will induce some heating issues. We want to discuss some heating issues in both the Pi's, particularly the GPU/CPU of both the Raspberry Pi's. We want to discuss some heating issues of the Pi's like Heat-sinks. The natural climate of the place the Pi is being used in also matters. The places warmer in nature such as us, and the places which are colder in nature such as the UK or the European Regions might have different base temperatures. Heat sink can be defined as a small attachment for a microprocessor chip which primarily functions by absorbing heat directly from the chip and dissipating that into the air. A microprocessor's temperature should not run in excess of 50-55 degrees Celsius while under a full load [16]. Table 1 shows the temperature variation of the two raspberry pi's (one used to run the motors and one used to run the modules), without any heating solutions, which means that pi's are on their own with no help for heating, (All Figures Are In Degree Celsius (C)). Here, Raspberry Pi 1 is called the one we put on the lower level of the chassis i.e. the one, which is responsible for the motion of the rover. The Raspberry Pi 2 is the one where we have attached the enhanced capabilities of the rover. We have obtained the temperature of the GPU/CPU via a simple command. Table 1 shows the temperature variation of the two raspberry pi's (one used to run the motors and one used to run the modules), without any heating solutions, which means that pi's are on their own with no help for heating.

Table 1. Temperature results with no solution for heating

	Raspberry Pi 1	Raspberry Pi 2
No use	36.9	39.8
0.5m	37.4	40.3
1m	38.5	41.4
1.5m	38.5	41.4
2m	39.0	41.9
2.5m	39.5	42.4
3m	40.1	42.9

1. vcgencmd measure_temp

Through this simple command this will tell you the temperature of Pi in the terminal. We have to discuss some of the heating solutions of this particular situation. Overheating may damage the Pi as sometimes Overlooking does. We will discuss some of the heat-sinks here. Some heat sinks are available in the market which work just like "Peel and paste", on the other hand some of the bigger heat sinks require a thermal paste to work on the Pi without damaging its components. Table 2 shows the temperature variation of the two raspberry pis with a small heat sink(size equivalent to CPU on Pi) to control heating and provide cooling. We might use a fan to get better results for the temperature analysis results as the cooling factor in a fan is way more direct and better than a heat sink.

We also experimented cooling results with a fan directly next to the CPU/GPU and recorded the results. The following are some temperatures we get. Table 3 shows the temperature variation of the two raspberry pi's with a small fan for CPU's to control heating and provide cooling. Table 4 shows the temperature variation of the two raspberry pi's with a larger heat sink to control heating and provide cooling.

Table 2. Temperature results with small sized heat sink

	Raspberry Pi 1	Raspberry Pi 2
No use	34.9	37.5
0.5m	35.4	38.0
1m	36.0	38.4
1.5m	37.1	39.5
2m	37.6	40.6
2.5m	38.1	41.1
3m	38.6	41.4

Table 2 shows the temperature variation of the two raspberry pi's with a small heats ink (size equivalent to CPU on Pi) to control heating and provide cooling.

Table 3. Temperature results with small fan

	Raspberry Pi 1	Raspberry Pi 2
No use	33.2	35.2
0.5m	33.7	35.2
1m	33.7	35.7
1.5m	34.8	35.7
2m	35.9	36.8
2.5m	36.1	36.8
3m	36.6	37.3

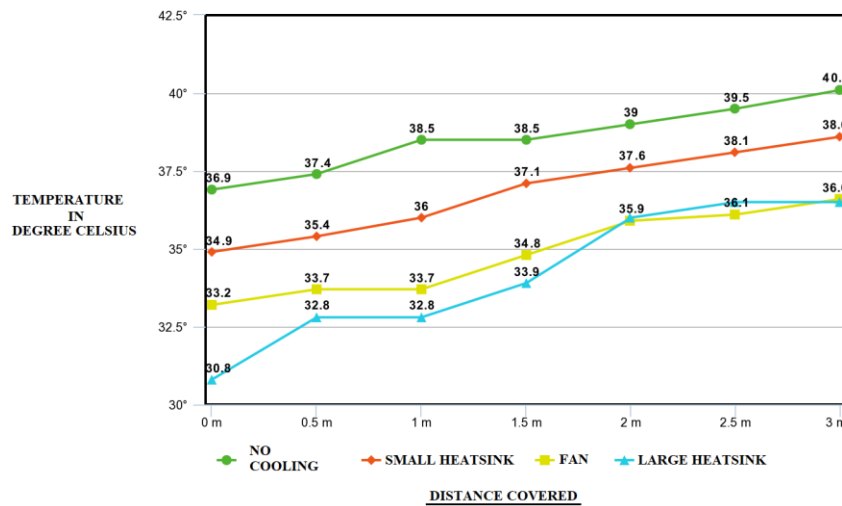
Table 3 shows the temperature variation of the two raspberry pi's with a small fan for CPU's to control heating and provide cooling.

Table 4. Temperature results with bigger heat sink

	Raspberry Pi 1	Raspberry Pi 2
No use	30.8	32.7
0.5m	32.8	34.8
1m	32.8	36.5
1.5m	33.9	36.5
2m	36.0	36.7
2.5m	36.5	37.1
3m	36.5	37.7

Table 4 shows the temperature variation of the two raspberry pi's with a larger heats ink to control heating and provide cooling. Hence we see that fans are the best way to cool the Single Board Computer having a heavy data usage then we might use a big heat sink and then smaller heat sinks. We plot a graph for the temperature and the distance covered for the Pi which is responsible for the motion of the rover which we have been referring to as Raspberry Pi 1. Figure 16 is the graphical representation of the temperatures of the first Pi to track the manner in which the two quantities affect each other. Figure 16 shows the relation between Temperature and Distance. This Pi is responsible for the motion of the rover. We also have to analyze this data for the Pi which will fetch the data for Quantities. We now analyze the similar graphical situation with the Raspberry Pi which is assigned the responsibility of using the sensors. Figure 17 is the graphical representation of the temperatures of the second Pi to track the manner in which the two quantities affect each other.

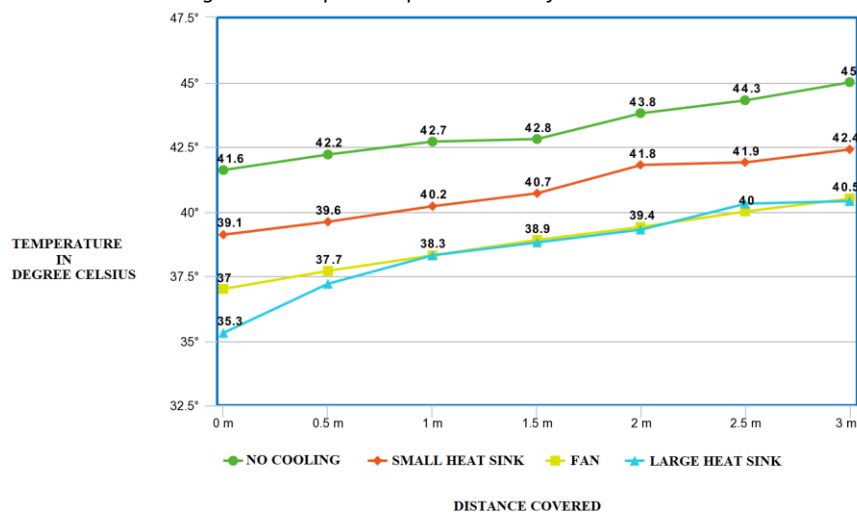
Figure 16. Graphical representation for first Pi



This graph shows the relation between the two quantities in the first raspberry pi in different cooling scenarios. The graph in Fig.16 shows the relation between Temperature and Distance. This Pi is responsible for the motion of the rover

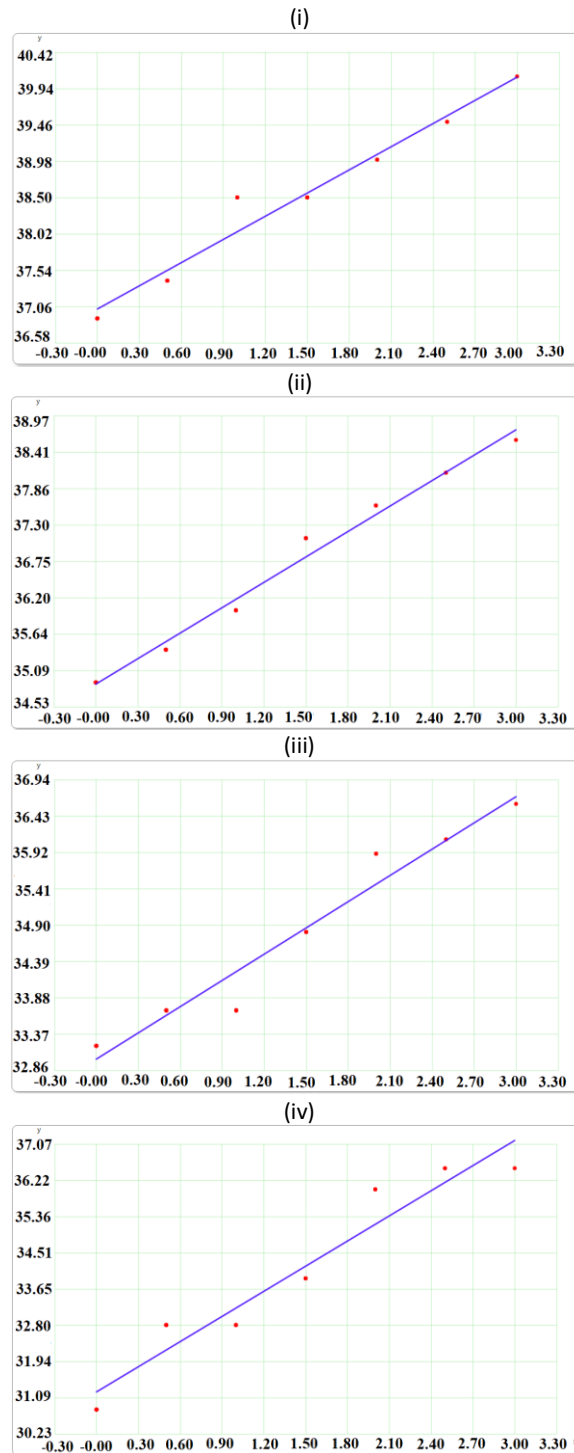
Hence we have graphically represented the relation between the distance covered by the rover car and the temperature of the CPU/GPU of the single board computer in it. The various cooling solutions we employed here worked and were successful to a great degree in cooling down the CPU/GPU. We can derive an equation for the curve using the concept called regression. When we have a set of data we can derive its equation in a liner, parabolic, polynomial form. Linear regression is used for finding linear relationship between target and one or more predictors. There are two types of linear regression- Simple and Multiple. Simple linear regression is useful for finding relationship between two continuous variables. One is predictor or independent variable and other is response or dependent variable. It looks for statistical relationship but not deterministic relationship. Relationship between two variables is said to be deterministic if one variable can be accurately expressed by the other. For example, using temperature in degree Celsius it is possible to accurately predict Fahrenheit. Statistical relationship is not accurate in determining relationship between two variables. For example, relationship between height and weight. If we let y represent delivery time and x represent delivery volume, then the equation of a straight line relating these two variables is $y = \beta_0 + \beta_1 X$. Where β_0 is the intercept and β_1 is the slope. Now the data points do not fall exactly on a straight line, so Eq. (1.1) should be modified to account for this.

Figure 17. Graphical representation of the second Pi



This graph shows the relation between the two quantities in the second raspberry pi in different cooling scenarios

Figure18. Temperature variations on first Pi



(Above is demonstrated the linear graph for the linear regression of the two quantities- Distance and Temperature, and the graphs respectively display (I) No cooling, (II) Small heat sink, (III) Fan and (IV) Large heat sink (Pi-1).)

The linear regression model $y = X\beta + \epsilon$ is a general model for fitting any relationship that is linear in the unknown parameters β . This includes the important class of polynomial regression models. For example, the second - order polynomial in one variable [17-18].

$$y = \beta_0 + \beta_1 x + \beta_2 (x^2) + \epsilon$$

And the second - order polynomial in two variables

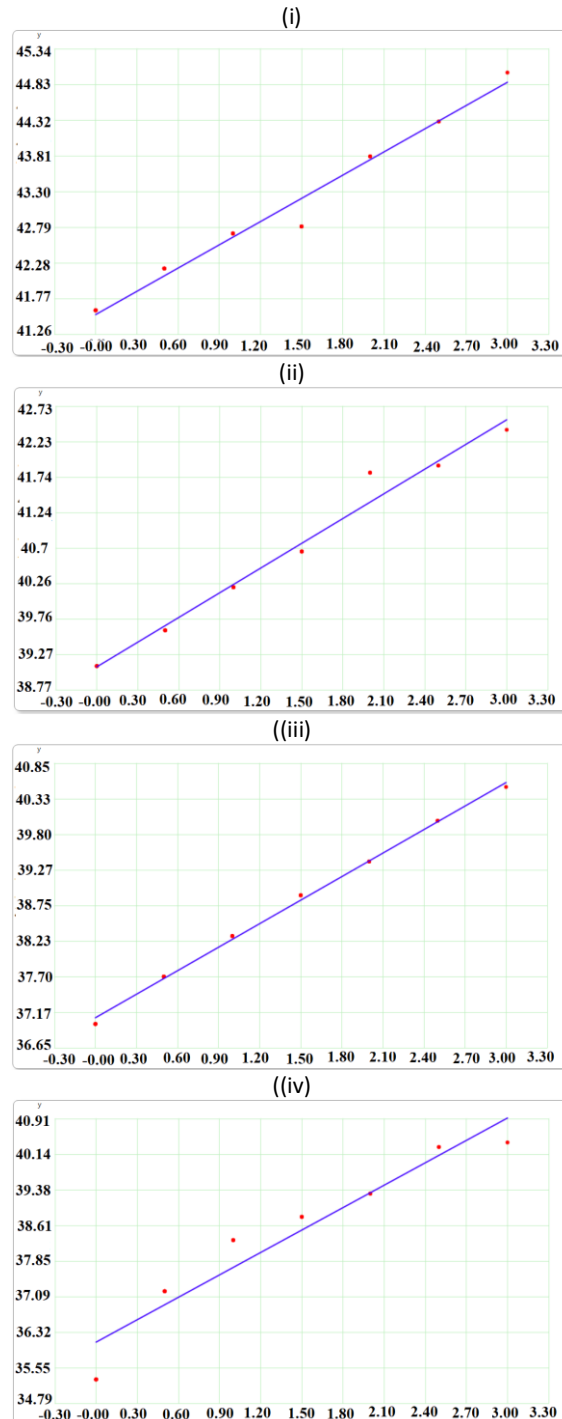
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} ((x_1^2)) + \beta_{22} ((x_2^2)) + \beta_{12} x_1 x_2 + \epsilon$$

Are linear regression models. Polynomials are widely used in situations where the response is curvilinear, as even complex nonlinear relationships can be adequately modeled by polynomials over reasonably small ranges of the x 's. This chapter will survey several problems and issues associated with fitting polynomials. Correlation Coefficients formula is given as following

$$r = \frac{(n \sum xy) - (\sum x)(\sum y)}{[(n \sum x^2 - (\sum x)^2)[(n \sum y^2 - (\sum y)^2)]^{1/2}}$$

In the Figure 18(i)-(iv)) shows the graph obtained by the linear regression of the quantities of the first pi in different cooling scenarios, similarly Figure 19(i)-(iv)) shows the graph obtained by the linear regression of the quantities of the first pi in different cooling scenarios.

Figure 19. Temperature variations on second Pi



(Above is demonstrated the linear graph for the linear regression of the two quantities- Distance and Temperature, and the graphs respectively display(I) No cooling, (II) Small heat sink, (III) Fan and (IV) Large heat sink(Pi-2).)

The Figures 18-19 are demonstrated the linear graphs for the linear regression of the two quantities and the graphs respectively display No cooling, small heat sink, fan and large heat sink (Pi-2). Here are the coefficients we got on solving these linear regressions. Table 5 shows the correlation coefficient of the two pi's in different cooling scenarios. Table 6 deduces the relation equation between the two quantities in all the cases we have discussed. The website <https://arachnoid.com/> helped us to create graphs and the relation models between the two quantities. Table 5 shows the correlation coefficient of the two pi's in different cooling scenarios.

Table 5. This table shows the correlation coefficient in various cases

	Raspberry PI 1	Raspberry PI 2
No cooling	0.9663988657847515	0.9772616335827063
Small heat sink	0.9858269138177704	0.9777082076038671
Fan	0.9521112255402043	0.99771897810261
Large heat sink	0.932444584872753	0.9248940368312355

Table 6 deduces the relation equation between the two quantities in all the cases we have discussed.

Table 6. This table is representation of the relation between distance and temperature after solving the linear regression in the form of a proper relation		
	Raspberry PI 1	Raspberry PI 2
No cooling	$f(x) = 3.702500000000027e+001 * x^0 + 1.0214285714285600e+000 * x^1$	$f(x) = 4.1539285714285718e+001 * x^0 + 1.1071428571428572e+000 * x^1$
Small heat sink	$f(x) = 3.487499999999979e+001 * x^0 + 1.2928571428571527e+000 * x^1$	$f(x) = 3.9089285714285758e+001 * x^0 + 1.1499999999999773e+000 * x^1$
Fan	$f(x) = 3.3014285714285691e+001 * x^0 + 1.2285714285714400e+000 * x^1$	$f(x) = 3.7092857142857142e+001 * x^0 + 1.1571428571428604e+000 * x^1$
Large heat sink	$f(x) = 3.1217857142857149e+001 * x^0 + 1.9785714285714238e+000 * x^1$	$f(x) = 3.6103571428571428e+001 * x^0 + 1.6071428571428572e+000 * x^1$

9. Conclusion

The rover which has the capability of detecting quantities like Temperature, humidity, sound, vibrations, human/animal presence, water, and can deploy sensors like mq5 and mq7 can be build using two different raspberry pi's with separate uses of providing motion and implementing sensors. Detection of environmental quantities to detect problem/hazard or even performing disaster mitigation is an application of the rover and this paper would also help in developing cooling solutions for the rover even further. The rover moves on a speed of 0.18m/s, is capable of clicking photos and live streaming videos on to a remote screen. The rover itself can be accessed via a remote screen. The data generated on a terminal screen can be recorded. On looking at temperature we find out that this particular rover we built contains a linear relationship between distance travelled and the temperature of the CPU. For cooling, a big sized heats ink is recommended for the CPU, as the trends we mark and the equations we derive from the regression of data, indicates the same.

10. References

1. Single Board Computer by Beal, Vangie. https://www.webopedia.com/TERM/S/sbc_single_board_computer.html. Date accessed: 05/04/2019.
2. What's the Difference Between Robotics and Artificial Intelligence? by Alex Owen Hill. <https://blog.robotiq.com/whats-the-difference-between-robotics-and-artificial-intelligence>. Date accessed: 08/04/2019.
3. Raspbian. <https://www.raspberrypi.org/downloads/raspbian/>. Date Accessed: 08.03/2019.
4. Basics: What is a Motor Controller? <https://www.robotshop.com/community/tutorials/show/basics-what-is-a-motor-controller?> Date Accessed: 09/03/2019.
5. Chassis. <https://www.merriam-webster.com/dictionary/chassis>. Date Accessed: 11/03/2019.
6. Getting started. https://www.freecadweb.org/wiki/Getting_started. Date Accessed: 09/03/2019.
7. David Halliday, Robert Resnick, Jearl Walker. Fundamentals of Physics. (9th Edition). Publisher: Wiley. 2011.
8. Understanding D.C. Motor Characteristics. <http://lancet.mit.edu/motors/motors3.html>. Date Accessed: 12/04/2019.
9. Power and Torque. http://www.epi-eng.com/piston_engine_technology/power_and_torque.htm. Date Accessed: 09/04/2019.
10. SSH protocol. <https://www.ssh.com/ssh/protocol/>. Date Accessed: 12/03/2019.
11. Forouzan, Behrouz. Data Communication and Networking. Publisher: McGrawHill. 2007.
12. Secure Shell (SSH) by Rouse, Margaret. <https://searchsecurity.techtarget.com/definition/Secure-Shell>. Date Accessed: 12/04/2019.
13. Putty. <https://www.putty.org/>. Date Accessed: 12/04/2019.
14. Xrdp. <http://www.xrdp.org/>. Date Accessed: 15/03/2019.
15. Remote Desktop by Rouse, Margaret. <https://searchenterprisedesktop.techtarget.com/definition/remote-desktop>. Date Accessed: 14/04/2019.
16. Heat sink by Margaret Rouse. <https://whatis.techtarget.com/definition/heatsink>. Date Accessed: 17/4/2019.
17. Linear Regression-Detailed View. <https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86>. Date Accessed: 26/04/2018.
18. D.C. Montgomery, E.A. Peck, G. Geoffrey Vining. Introduction to linear regression. Publication: John Wiley & Sons, inc. publication. 2012; 672.

The Publication fee is defrayed by Indian Society for Education and Environment (www.iseeadyar.org)

Cite this article as:

Rohan Tomar. Environment analyzing rover based on Raspberry Pi – construction and analysis. *Indian Journal of Automation and Artificial Intelligence*. Vol 6(7), July 2019.