A PROJECT REPORT

on

**BatMobile – IOT Rover**

*Submitted by*

**Mr. Ganesh Umesh Tiwari**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF SCIENCE**

in

**COMPUTER SCIENCE**

*under the guidance of*

**Prof. Vipul Saluja**

**Department of Computer Science**



**R. D. & S.H. National College & S. W. A. Science College**

**Bandra, Mumbai – 400050.**

**(Sem V)**

**(2018 – 2019)**

**DECLARATION**

I, Mr. Ganesh Umesh Tiwari, hereby declare that the project entitled “**Quiz App**” submitted in the partial fulfillment for the award of **Bachelor of Science** in **Computer Science** during the academic year **2018 – 2019** is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

**Signature of the Student:**

**Place:**

Bandra West Mumbai 400050

**Date:**

21/03/2019

You don’t get another chance, life is no Nintendo game.

**Marshall Mathers**

**TABLE OF CONTENTS**

**Chapter 1 – Project Definition**

|  |  |  |
| --- | --- | --- |
| 1.1 | Introduction |  |
| 1.2 | Objective |  |
| 1.3 | Purpose of Project |  |
| 1.4 | Components Used |  |

**Chapter 2 – System Analysis**

|  |  |  |
| --- | --- | --- |
| 2.1 | Introduction |  |
| 2.2 | Advantages and Disadvantages of Arduino |  |
| 2.3 | Advantages and Disadvantages of Raspberry Pi |  |

**Chapter 3 – System Design**

|  |  |  |
| --- | --- | --- |
| 3.1 | Design of the project |  |
| 3.2 | Circuit Diagram |  |
| 3.3 | Design of project and placement |  |
| 3.4 | Applications of Arduino and Raspberry |  |

**Chapter 4 – Requirement Analysis**

|  |  |  |
| --- | --- | --- |
| 4.1 | Arduino UNO |  |
| 4.2 | Raspberry Pi 3 |  |
| 4.3 | RaspiCam Module |  |
| 4.4 | Ultrasonic Sensor |  |
| 4.5 | DHT Sensor |  |
| 4.6 | Power Supply |  |

**Chapter 5 – System Installation Process and Particular Requirements**

|  |  |  |
| --- | --- | --- |
| 5.1 | Installing Raspbian OS |  |
| 5.2 | Installing Arduino IDE |  |
| 5.3 | Installing MJPEG Streamer |  |
| 5.4 | Installation of Mosquitto (Mqtt Broker) |  |
| 5.5 | Installation of Python Libraries of Mqtt and PySerial |  |
| 5.6 | Android Libraries for Application (Mqtt, Barcode Scanning, Simple Mjpeg Viewer) |  |

**Chapter 6 – Implementation and Testing**

|  |  |  |
| --- | --- | --- |
| 5.1 | Programming Arduino |  |
| 5.2 | Programming Raspberry Pi |  |
| 5.4 | Android App Development |  |

**Chapter 7 – Conclusion**

**Chapter 8 – Future Enhancement**

**Chapter 9 – References**

**Introduction**

This project is entitled for the development of a fully functional mobile prototype “the Rover Station”, responsible for environmental data capture as Temperature, Humidity and Luminosity. The idea is in the future aggregate other functions to emulate what would be a Mars Rover emulator.

This project is built upon with help of growing technological advancements in the fields of Internet of things and Wi-Fi technology. This project utilizes Wireless LAN technology for connection between objects and MQTT protocol for transfer of data between the devices.

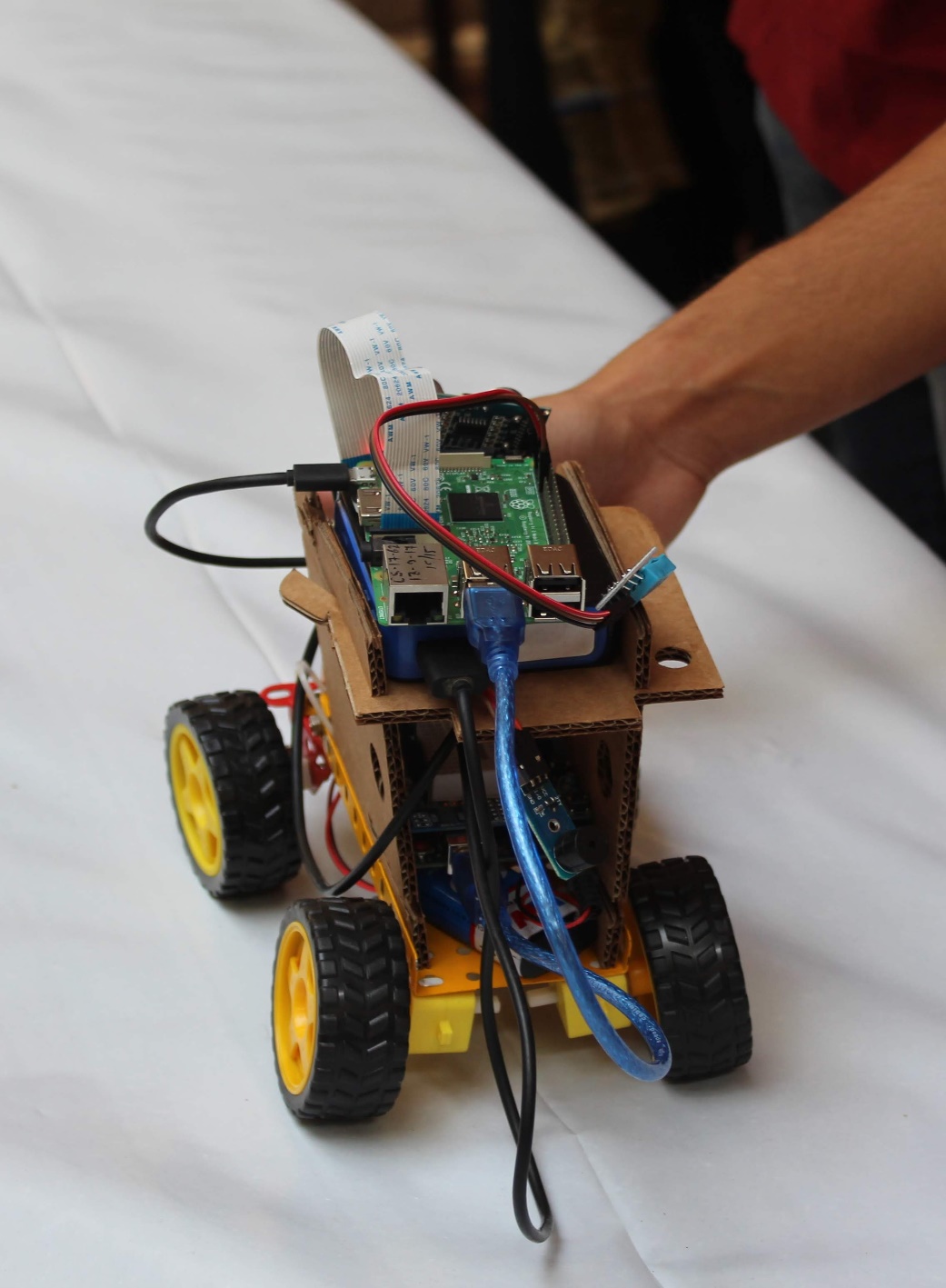


Figure Earlier Beta Model Built by us.

**Objective**

This project was built with having certain ideas in mind, it can be used for some ideal situations like monitoring over things where humans can’t reach or it is difficult to reach there. It can also be used as a harmless transport vehicle for small things to transport things from one place to another. It can be used to gather information about a place which is in remote location so that certain tasks can be achieved without actually getting a human intervention.

**Purpose of the project**

Purpose of this project was not actually about for making any discoveries or anything but it was for a Competition which was held in our college. We were excited to participate in the competition and to represent our college in front of other colleges and be able to see what others are building in their project. I was always eager to learn about Internet of Things and how different things if connected together and made available on internet can actually make a difference in the whole world.

**Components Used in the project.**

**Raspberry Pi: -**

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries.[5][6][7] The original model became far more popular than anticipated,[8] selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards and mice) and cases. However, some accessories have been included in several official and unofficial bundles.

**Arduino: -**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

**Jumper Cables: -**

A jump wire (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), 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.

**Raspberry Pi Camera Module: -**

The Raspberry Pi Camera Module is a 5MP CMOS camera with a fixed focus lens that is capable of capturing still images as well as high definition video. Stills are captured at a resolution of 2592 x 1944, while video is supported at 1080p at 30 FPS, 720p at 60 FPS and 640x480 at 60 or 90 FPS. The camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system.

**Ultrasonic Sensor: -**

As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

**DHT Sensor: -**

The DHT sensors are made of two parts, a capacitive humidity sensor and a thermistor. There is also a very basic chip inside that does some analog to digital conversion and spits out a digital signal with the temperature and humidity. The digital signal is fairly easy to read using any microcontroller.

**Power Supply: -**

Power banks can be defined as portable batteries that use circuitry to control any power in and power out. They can charge up using a USB charger when power is available, and then used to charge battery powered items like mobile phones and a host of other devices that would normally use a USB charger.

Lithium batteries are primary batteries that have metallic lithium as an anode. These types of batteries are also referred to as lithium-metal batteries.

They stand apart from other batteries in their high charge density (long life) and high cost per unit. Depending on the design and chemical compounds used, lithium cells can produce voltages from 1.5 V (comparable to a zinc–carbon or alkaline battery) to about 3.7 V. Disposable primary lithium batteries must be distinguished from secondary lithium-ion, lithium iron phosphate and lithium-polymer,[1] which are rechargeable batteries. Lithium is especially useful, because its ions can be arranged to move between the anode and the cathode, using an intercalated lithium compound as the cathode material but without using lithium metal as the anode material. Pure lithium will instantly react with water, or even moisture in the air; the lithium in lithium ion batteries is in a less reactive compound.

Lithium batteries are widely used in portable consumer electronic devices, and in electric vehicles ranging from full sized vehicles to radio controlled toys.

**System Analysis**

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

**Advantages and Disadvantages of Arduino**

**Advantages: -**

**1- Ready to Use:**

The biggest advantage of Arduino is its ready to use structure. As Arduino comes in a complete package form which includes the 5V regulator, a burner, an oscillator, a micro-controller, serial communication interface, LED and headers for the connections. You don't have to think about programmer connections for programming or any other interface. Just plug it into USB port of your computer and that's it. Your revolutionary idea is going to change the world after just few words of coding.

**2- Examples of codes:**

Another big advantage of Arduino is its library of examples present inside the software of Arduino. I'll explain this advantage using an example of voltage measurement. For example, if you want to measure voltage using ATmega8 micro-controller and want to display the output on computer screen then you have to go through the whole process. The process will start from learning the ADC's of micro-controller for measurement, went through the learning of serial communication for display and will end at USB - Serial converters.

**3- Effortless functions:**

During coding of Arduino, you will notice some functions which make the life so easy. Another advantage of Arduino is its automatic unit conversion capability. You can say that during debugging you don't have to worry about the unit’s conversions. Just use your all force on the main parts of your projects. You don't have to worry about side problems.

**4- Large community:**

There are many forums present on the internet in which people are talking about the Arduino. Engineers, hobbyists and professionals are making their projects through Arduino. You can easily find help about everything. Moreover, the Arduino website itself explains each and every functions of Arduino.

**Disadvantages: -**

**1- Structure:**

Yes, the structure of Arduino is its disadvantage as well. During building a project you have to make its size as small as possible. But with the big structures ofArduino we have to stick with big sized PCB’s. If you are working on a small micro-controller like ATmega8 you can easily make your PCB as small as possible.

**2- Easy to use:**

In my opinion, if you started your journey of micro-controllers with Arduino then it will be very difficult for you to make the complex intelligent circuitries in future. The easy to use hardware/software of Arduino unable a person to learn the basics of many things likes Serial communication, ADC, I2C etc. This basically makes you think that microcontrollers are easy which is not the case.

**Advantages and Disadvantages of Raspberry Pi**

**Advantages: -**

* You can install a fully-fledged Operating System (e.g. Raspbian which is Linux-based or Windows IOT Core which is Windows based OS) and use it as a day to day computer
* The presence of GPIO (General Purpose Input Output pins) is what distinguishes a RPi from traditional computers. You can connect these pins to sensors and external components and interact with them programmatically using a language such as Python. This allows you to build and prototype Internet of Things devices that can sense the real world
* Newer models like the RPi has WIFI and Bluetooth built in. This allows you to take projects into wireless mode easily.

**Disadvantages: -**

* There is not any fuse protection on the Rpi, so if you connect pins incorrectly, you can damage the board
* It is not as fast in terms of CPU processing speed nor does it have as much memory as traditional PC or laptops. But for $35, it packs quite a punch
* There is no built in analog to digital conversion on the GPIO pins like there is in Arduino. So you need to have an ADC chip to work with analog signals when you need more accuracy

**Requirement Analysis**

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. Requirements analysis involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore requirements analysis means to analyze, document, validate and manage software or system requirements. High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to a facilitate system design.

* **Requirements analysis process**

The requirements analysis process involves the following steps

* **Eliciting requirements**

The process of gathering requirements by communicating with the customers is known as eliciting requirements.

* **Analyzing requirements**

This step helps to determine the quality of the requirements. It involves identifying whether the requirements are unclear, incomplete, ambiguous, and contradictory. These issues resolved before moving to the next step.

* **Requirements modelling**

In Requirements modelling, the requirements are usually documented in different formats such as use cases, user stories, natural-language documents, or process specification.

* **Review and retrospective**

This step is conducted to reflect on the previous iterations of requirements gathering in a bid to make improvements in the process going forward.

After carrying out requirement analysis for our project we came out with a certain list of the things which I needed to make sure are present in order for this project to be in a working Condition. They are already mentioned in other places as well as in this documentation but I will list them here for convenience.

1. Raspberry Pi Model 3B
2. Arduino Nano
3. Jumper Cables
4. Power Supply (Power Bank as well as rechargeable Lithium Battery’s)
5. Raspberry Camera Module
6. DHT11 Sensor

**System Installation and Processing**

Installing Raspbian Operating System

**Download the image**

Official images for recommended operating systems are available to download from the Raspberry Pi website Downloads page.

Alternative distributions are available from third-party vendors.

If you're not using Etcher (see below), you'll need to unzip .zip downloads to get the image file (.img) to write to your SD card.

Note: the Raspbian with Raspberry Pi Desktop image contained in the ZIP archive is over 4GB in size and uses the ZIP64) format. To uncompress the archive, a unzip tool that supports ZIP64 is required. The following zip tools support ZIP64:

* 7-Zip (Windows)
* The Unarchiver (Mac)
* Unzip (Linux)

**Writing an image to the SD card**

Before you start, don't forget to check the SD card requirements.

You will need to use an image writing tool to install the image you have downloaded on your SD card.

Etcher is a graphical SD card writing tool that works on Mac OS, Linux and Windows, and is the easiest option for most users. Etcher also supports writing images directly from the zip file, without any unzipping required. To write your image with Etcher:

* Download Etcher and install it.
* Connect an SD card reader with the SD card inside.
* Open Etcher and select from your hard drive the Raspberry Pi .img or .zip file you wish to write to the SD card.
* Select the SD card you wish to write your image to.
* Review your selections and click 'Flash!' to begin writing data to the SD card.

**Installing Arduino IDE for Raspberry Pi**

The first step in programming an Arduino board with a Raspberry Pi is to install the Arduino IDE (integrated development environment) on your Raspberry Pi. This program checks code and loads it onto the Arduino. Install the latest version of Arduino IDE using apt:

sudo apt-get update && sudo apt-get upgrade  
sudo apt-get install Arduino

Alternatively, open Chrome on your Raspberry Pi, head to magpi.cc/2tPw8ht, and click the Linux ARM link under ‘Download the IDE’. Extract the file to your /opt directory , then open a Terminal and run the install.sh script to install.

cd Downloads/  
tar -xf arduino-1.8.3-linuxarm.tar.xz  
sudo mv arduino-1.8.3 /opt  
sudo /opt/arduino-1.8.3/install.sh

You will find Arduino IDE under Menu > Programming. Open the app to start programming your Arduino board.

**Installing MJPEG Streamer**

To install Mjpeg Streamer it is necessary to have all the components that are installed on the system to be up to date.

sudo apt-get update  
sudo apt-get upgrade -y

In order to make Mjpeg streamer to work properly on your system you should have

* Build essentials
* Libjpeg
* Imagemagick
* Libv4l-devl
* cmake

They can be installed by a single command

sudo apt-get install build-essential libjpeg8-dev imagemagick libv4l-dev cmake -y

Mjpeg Streamer is freely and openly available software developed on github and these are the necessary steps required to build the Mjpeg streamer directly from the source itself

cd /tmp  
git clone https://github.com/jacksonliam/mjpg-streamer.git  
cd mjpg-streamer/mjpg-streamer-experimental

make  
sudo make install

**Installing Mosquitto MQTT Broker**

Installing mosquitto mqtt broker is a piece of cake as it is also a freely available software which is basically used in most of IOT projects.

sudo apt-get install mosquitto mosquitto-clients

This single line of code does all of the work and install mosquitto mqtt broker on raspberry pi

**Installing python libraries of MQTT and PySerial on raspberry**

Pip (pip) is a package management system used to install and manage software packages written in Python. Many packages can be found in the default source for packages and their dependencies — Python Package Index. Python 2.7.9 and later, and Python 3.4 and later include pip by default.

Pip is used to install all the packages and libraries in python.

MQTT can be installed as

pip install paho-mtt

PySerial can be installed as

pip install PySerial

PySerial can be installed from Conda:

conda install pyserial

**Android Libraries for Application (Mqtt, Barcode Scanning, Simple Mjpeg Viewer)**

In order to use a Support Library, you must modify your application's project's classpath dependencies within your development environment. You must perform this procedure for each Support Library you want to use.

**To add a Support Library to your application project:**

Include Google's Maven repository in your top-level build.gradle file.

allprojects {  
    repositories {  
        google()

jcenter()  
  
        // If you're using a version of Gradle lower than 4.1, you must  
        // instead use:  
        //  
        // maven {  
        //     url 'https://maven.google.com'  
        // }  
    }  
}

Add the support library to the dependencies section. For example, to add the v4 core-utils library, add the following lines:

dependencies {  
    implementation fileTree(dir: 'libs', include: ['\*.jar'])  
    implementation"org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin\_version"  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
  
    //exoplayer  
    implementation 'com.google.android.exoplayer:exoplayer:2.9.3'  
    //anko commons  
    implementation "org.jetbrains.anko:anko-commons:$anko\_version"  
    implementation 'com.github.niqdev:mjpeg-view:1.6.0'  
    implementation 'com.android.volley:volley:1.1.1'  
    implementation 'com.google.code.gson:gson:2.8.5'  
  
    implementation 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.1.0'  
    implementation 'org.eclipse.paho:org.eclipse.paho.android.service:1.1.1'  
  
    implementation 'com.github.stealthcopter:AndroidNetworkTools:0.4.3'  
  
    implementation 'com.journeyapps:zxing-android-embedded:3.6.0'  
  
    implementation 'com.github.dydwo92:Android-Simple-MjpegViewer:0.0'  
}