A project report on

Virtual Reality Integrated Surveillance Vehicle

*(VR Car)*

*Submitted by*

Manik(15CSU109),

Pallak(15CSU144),

Vinay(15CSU258),

and Vineet(15CSU259)



Department of CSE & IT

The NorthCap University

Gurugram

A project report on

Virtual Reality Integrated Surveillance Vehicle

*(VR Car)*

*Of*

**Bachelor of Technology**

**in Computer Science Engineering**

*by*

Manik(15CSU109),

Pallak(15CSU144),

Vinay(15CSU258),

and Vineet(15CSU259)

Under Supervision of

Dr. Jyotsna Singh



Department of CSE & IT

The NorthCap University

Gurugram

**CERTIFICATE**

This is to certify that the Project Report entitled, “Virtual Reality Integrated Surveillance Vehicle*(VR Car)*” submitted by “**Manik, Pallak, Vinay and Vineet”** to **The NorthCap University, Gurugram, India,** is a record of bonafide project work carried out by him/her under my supervision and guidance and is worthy of consideration for the award of the degree of **Bachelor of Technology** in **Computer Science Engineering** of the University.

Dr. Jyotsna Singh

Date: 12/10/18

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our supervisor, Dr. Jyotsna Singh, whose contribution in stimulating suggestions and encouragement as well as her active guidance helped us to make progress on our project especially in writing this report. We would like to extend our deepest appreciation to all those who provided us their support and helped us in this endeavor.

We are indebted to our teachers and friends who have provided us with the knowledge and encouragement to help us bring in our best to this project.

Manik

Pallak Singh

Vinay Garg

Vineet Jain

**ABSTRACT**

Virtual Reality is a rapidly developing technology that has found itself being employed in various domains. VR provides an immersive experience by filling the entire field of view with an image where the head movements control where and what the user sees. This project is going to bring a virtual reality experience to small scale robotic vehicles.

There are places in the world or sensitive projects where it is not safe for humans to go themselves but require devices controlled from a distant location to give the relevant and required information. We propose to bring an immersive experience to make information gathering more efficient and lifelike. The main focus of this project is to create an economical system that combines a robotic car with integrated VR so that such devices can be used widely in such fields. The project will be having many open source designs incorporated into it. The project will be using an Arduino and a Raspberry Pi to perform the functions it’s meant to. Further a mobile app will be developed which will offer many device settings and information (such as battery information), the mobile app will also be used to display the VR video, where one could the mobile phone as the VR headset.

**CONTENTS**

Certificate …………………………………………………………………………..i

Acknowledgement ...……………………………………………………………….ii

Abstract………………….……………….…………….…….….….….….….……iii

List of Tables …...……….…………………………………….…………………...iv

List of Figures ….…………………………………………….……………………iv

List of Symbols and Abbreviations ………….……………….……………………iv

1.Introduction.….………………………….……………………………………….1

* System Objective…………………………………………………………….1
* Market Landscape…………………………………………………………...2

2.System Design……………………………………………………………………2

* Hardware Interfaces…………………………………………………………2
* Software Interfaces…………………………………………………………..3

3. Design and Implementation……………………………………………………..4

4.Constraints, Problems and Cost

* Design and Implementation Constraints……………………………………5
* Safety Issues…………………………………………………………………6
* Security Issues……………………………………………………………….6
* Cost estimation…………………………...………………………………….6

5.Timeline……………………………………………..…………………………....7

6.References………………………………………………………………………...8

7.Glossary…………………………………………………………………………..8

**LIST OF TABLES**

Table 1 Estimated Cost………………………………………………………..6

Table 2 Major Milestones……………………………………………………..7

**LIST OF FIGURES**

Figure 1.1 Figure depicting VR headsets with smartphones placed inside……..1

Figure 2.1 Figure depicting the UML Use-Case diagram………………………3

Figure 3.1 Figure depicting the Data Flow Level 0 diagram…………………...4

Figure 3.2 Figure depicting the Data Flow Level 1 diagram…………………...4

Figure 3.2 Figure depicting the Arduino circuit diagram ……………………....5

Figure 4.1 Figure depicting latency and lag……………………………………..5

Figure 5.1 Gantt chart…………………………………………………………....7

**LIST OF SYMBOLS AND ABBREVATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Definition** |
| VR | Virtual Reality |
| HTTP | Hyper Text Transfer Protocol |
| IOT | Internet of Things |
| MOSFET | Metal-Oxide-Semiconductor Field-effect |
| MTP | Motion to Photon Latency |

1. **Introduction**

There are a variety of applications that require quick and real time action-response in the world of robots. We require rapid information with the most meticulous observations from the robots sent in the remote areas we cannot reach ourselves. This project proposes a way to make the user controlling the robot to experience those remote areas like he was there himself. Virtual Reality(VR) is no longer a part of science fiction or something that gets battle around in laboratories. VR focuses on creating an interactive and immersive experience that engages the body and mind. This project will utilize virtual reality for just this: an immersive and interactive experience.

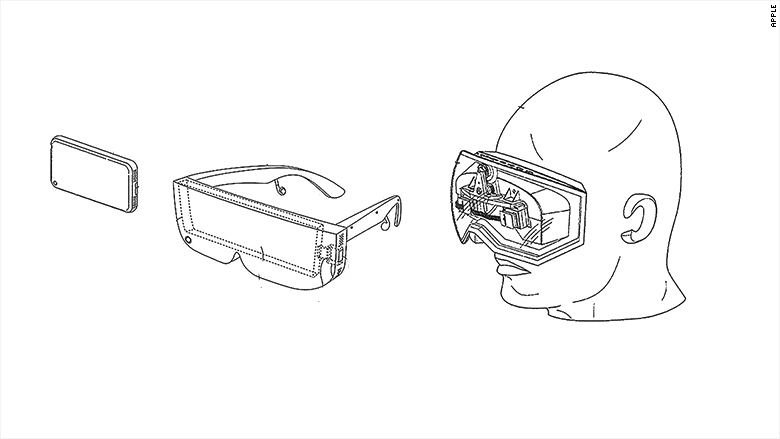


Figure 1.1: VR headset and smartphone

Source: http://pdfaiw.uspto.gov/

The project will be implementing hardware that will be able to broadcast a video feed from the dual cameras on board to a VR Headset. Further a mobile app to interact with the robotic car will be developed.

**1.1 System Objective**

This product in basic terminology is a Robotic Car that has VR implementation build into it. The main aim for this project is to create the hardware and the software required for the car. We aim to select optimum hardware such as microcontrollers, cameras and other parts for best possible performance. Software a VR headsets will be developed along the car which would let you interact with the car.

**1.2 Market Landscape**

We currently have identified a group of people could benefit or enjoy the product being developed by us if deployed in their workflows.

**Indented Audience**

* Gamer

Gamers can use the project to simulate a game environment in real life, for example: real life racing games between multiple similar cars where the gamer would have a more immersive and a real physical experience.

* Scientist

Scientists would be able to use to this explore alien or not easy to reach places with an immersive experience and further produce data for further research by adding customized sensors or add-ons as the designs will be open source.

* Differently Abled People

Differently abled can experience the world around them with little to no movements on their part

**1.3 What is 360-degree Video?**

360-Degree video allows the viewer to have a more immersive experience when watching a video, a typical video is usually just one camera focused on a scene what the camera man wants to show the viewer, but a 360-degree video records all the angles possible from the camera’s location, allowing the viewer to look at the presented scene from different perspectives or even move around the video to look around the scene.



Figure 1.2: How a 360-Degree video looks straight out of camera.

Source: Youtube

360-degree videos are typically generating using outputs of multiple wide angle cameras, and merging them together in a panoramic manner and producing a very distorted 16:9 video in using already pre-existing video formats and codecs, 360-degree video does not require new formats or codecs. 360-degree videos does benefit from having a high resolution output video as that allows more data to be compressed and stored in the 16:9 video.

What changes with 360-degree videos is in the manner they are played back, playing back a 360-degree video in a non-360 compatible player would playback the 16:9 file, where everything would seem distorted, making it incredibly hard to enjoy the video experience. Special piece of software is required, which manipulates 360-degree videos in a manner so that they seem normal to the eye when playing back while simultaneously allowing the viewer to move around the video.



Figure 1.3: How a 360-Degree video looks in a video capable of playing back 360-degree video.

Source: Youtube

**1.4 What is VR Video?**

Virtual reality (VR) is an interactive [computer-generated](https://en.wikipedia.org/wiki/Computer_generated_reality) experience taking place within a simulated environment, that incorporates mainly auditory and visual, but also other types of [sensory feedback](https://en.wikipedia.org/wiki/Perception#Types) like [haptic](https://en.wikipedia.org/wiki/Haptic_perception). This immersive environment can be similar to the real world or it can be fantastical, creating an experience that is not possible in ordinary physical reality.

[Virtual Reality](https://filmora.wondershare.com/virtual-reality/how-does-vr-work.html) videos are immersive video content accessed through the use of virtual reality headsets. Unlike the ordinary videos, virtual reality videos create in illusion of the user being part of the video through the rendition of 3D images. The viewing of such videos is not restricted to computer or television screens and depending on the video or the kind of content being watched, the user can also interact with the video.

Just like with the ordinary videos, virtual reality videos can be created at the beginning when a video is being recorded or captured using 360 degrees video cameras. The videos can also be made from computer generated content using VR immersive video enabled applications. The formats adopted by virtual reality videos will vary from the software and the device used in the production of the videos.



Figure 1.4: How a VR video appears in the headset.

Source: Youtube

1. **System Design**

**2.1 Hardware Interfaces**

To develop a system like this we had to decide among a lot of hardware selection and design the hardware and the circuitry. We needed to decide on the following fronts before continuing:

1. Microcontroller to drive the car.
2. On-Board computer to handle VR related tasks.
3. Sensors to provide various physical data to manage the virtual reality.
4. The Cameras to actually record and send the video.

**2.2 Software Interfaces**

* The Mobile software will provide the sensor data to the car hardware.
* HTTP is a publish-subscribe based messaging protocol. The Video transmission will be done  over HTTP on a local network.
* The Programming for the robot car will be done in Python, C and further software for the  headsets will be written in their respective languages and frameworks like Swift, Java, Unity etc.

1. **Technical Description**

**3.1 Software Environment**

**Xcode 9**

Xcode is an IDE for [macOS](https://en.wikipedia.org/wiki/MacOS) containing a suite of [software development](https://en.wikipedia.org/wiki/Software_development) tools developed by [Apple](https://en.wikipedia.org/wiki/Apple_Inc.) for developing software for macOS, [iOS](https://en.wikipedia.org/wiki/IOS), [watchOS](https://en.wikipedia.org/wiki/WatchOS), and [tvOS](https://en.wikipedia.org/wiki/TvOS). Using the [iOS SDK](https://en.wikipedia.org/wiki/IOS_SDK), Xcode can be used to compile and debug applications for [iOS](https://en.wikipedia.org/wiki/IOS) that run on [ARM architecture](https://en.wikipedia.org/wiki/ARM_architecture) processors.

**Objective-C**

Objective-C is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [programming language](https://en.wikipedia.org/wiki/Programming_language) that adds [Smalltalk](https://en.wikipedia.org/wiki/Smalltalk)-style [messaging](https://en.wikipedia.org/wiki/Message_passing) to the [C](https://en.wikipedia.org/wiki/C_(programming_language)) programming language. It was the main programming language used by [Apple](https://en.wikipedia.org/wiki/Apple_Inc.) for the [OS X](https://en.wikipedia.org/wiki/OS_X) and [iOS](https://en.wikipedia.org/wiki/IOS) operating systems.

**Python**

Python is an interpreted high-level programming language for general-purpose programming.

**Open-GL**

Open Graphics Library is a [cross-language](https://en.wikipedia.org/wiki/Language-independent_specification), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface) for rendering [2D](https://en.wikipedia.org/wiki/2D_computer_graphics) and [3D](https://en.wikipedia.org/wiki/3D_computer_graphics) [vector graphics](https://en.wikipedia.org/wiki/Vector_graphics). The API is typically used to interact with a [graphics processing unit](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPU), to achieve [hardware-accelerated](https://en.wikipedia.org/wiki/Hardware_acceleration) [rendering](https://en.wikipedia.org/wiki/Rendering_(computer_graphics)).

**HLS**

HTTP Live Streaming (also known as HLS) is an [HTTP](https://en.wikipedia.org/wiki/HTTP)-based media streaming communications protocol implemented by [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.) as part of its [QuickTime](https://en.wikipedia.org/wiki/QuickTime), [Safari](https://en.wikipedia.org/wiki/Safari_(web_browser)), [OS X](https://en.wikipedia.org/wiki/OS_X), and [iOS](https://en.wikipedia.org/wiki/IOS) software. Client implementations are also available in [Microsoft Edge](https://en.wikipedia.org/wiki/Microsoft_Edge), [Firefox](https://en.wikipedia.org/wiki/Firefox) and some versions of [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome). Support is widespread in streaming media servers.

**3.2 Hardware Environment**

**Arduino**

Arduino is an [open-source hardware](https://en.wikipedia.org/wiki/Open-source_hardware) and [software](https://en.wikipedia.org/wiki/Open-source_software) company, project and user community that designs and manufactures [single-board microcontrollers](https://en.wikipedia.org/wiki/Single-board_microcontroller) and [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. Its products are licensed under the [GNU Lesser General Public License](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License) (LGPL)

**Raspberry Pi**

The Raspberry Pi is a series of small [single-board computers](https://en.wikipedia.org/wiki/Single-board_computer) developed in the [United Kingdom](https://en.wikipedia.org/wiki/United_Kingdom) by the [Raspberry Pi Foundation](https://en.wikipedia.org/wiki/Raspberry_Pi_Foundation) to promote the teaching of basic [computer science](https://en.wikipedia.org/wiki/Computer_science) in schools and in [developing countries](https://en.wikipedia.org/wiki/Developing_countries).

**Odroid-XU4**

Odroid XU4, is a microcontroller development board similar to Raspberry Pi.

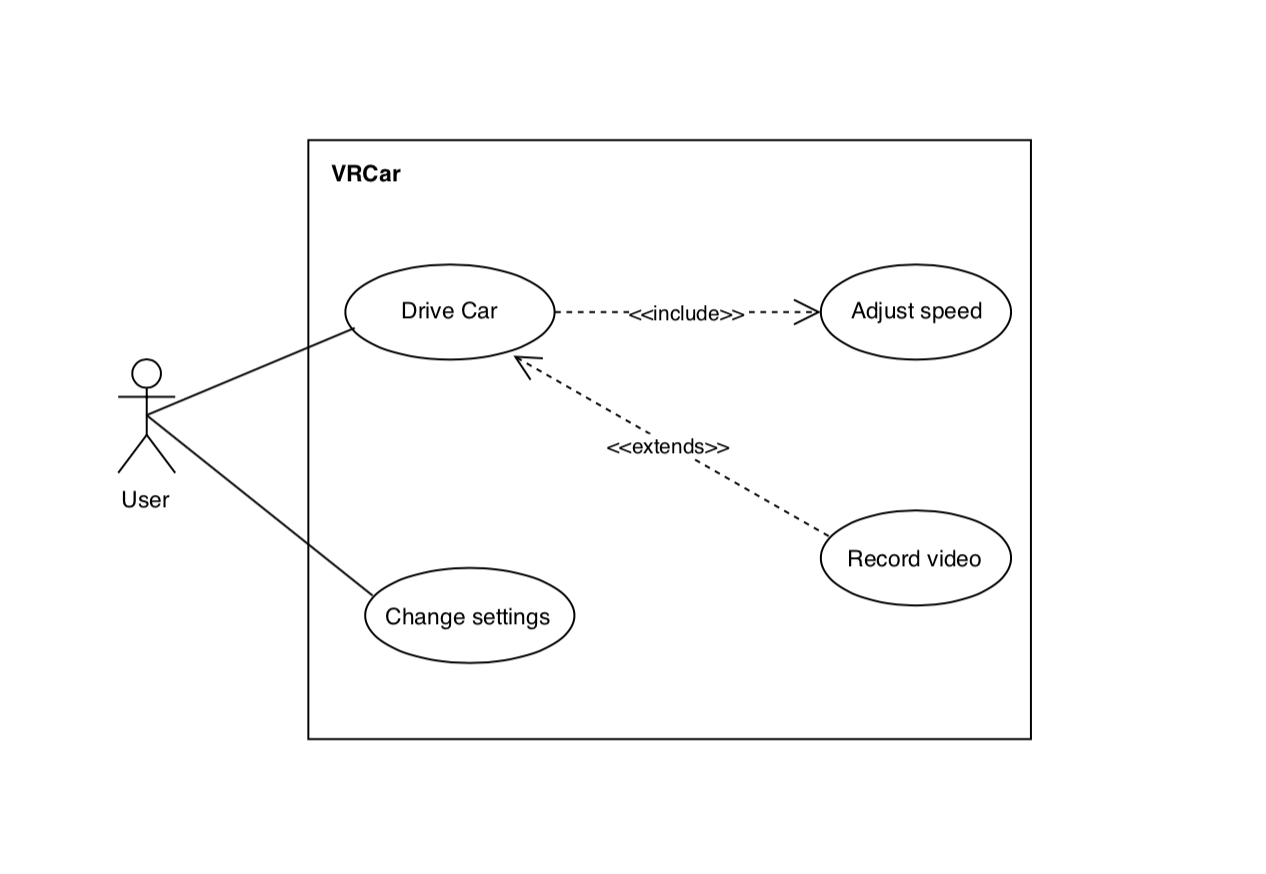
****

Figure 3.1: Use-Case Diagram

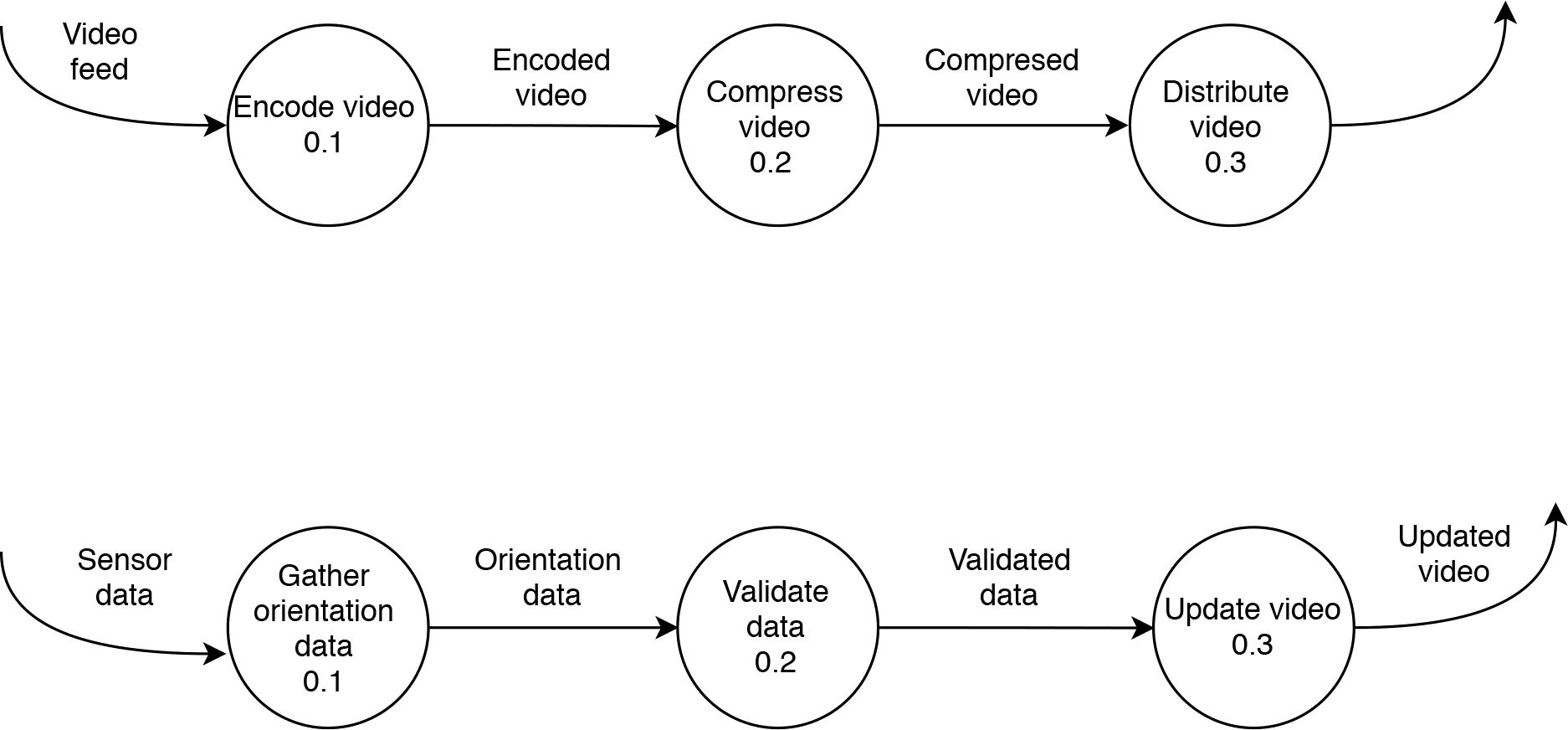
1. **Design and Implementation**

The developed system is going to utilize the sensors from the mobile devices to measure the motion and direction in space. An accelerometer, magnetometer and gyroscope data are combined to provide us the orientation and motion of the robotic car and how to change it accordingly. The sensor data will be sent to the raspberry pi mounted on the robotic car. The live feed from the camera will be broadcasted to the smartphone and hence, be displayed on the video mode of the mobile application. The project divides the system in to the two major parts:

1. Extraction of sensor data from smartphone for orientation and motion control
2. Video broadcast to the mobile application.



Figure 4.1: Context Diagram



*Figure 4.2: DFD Level 1*

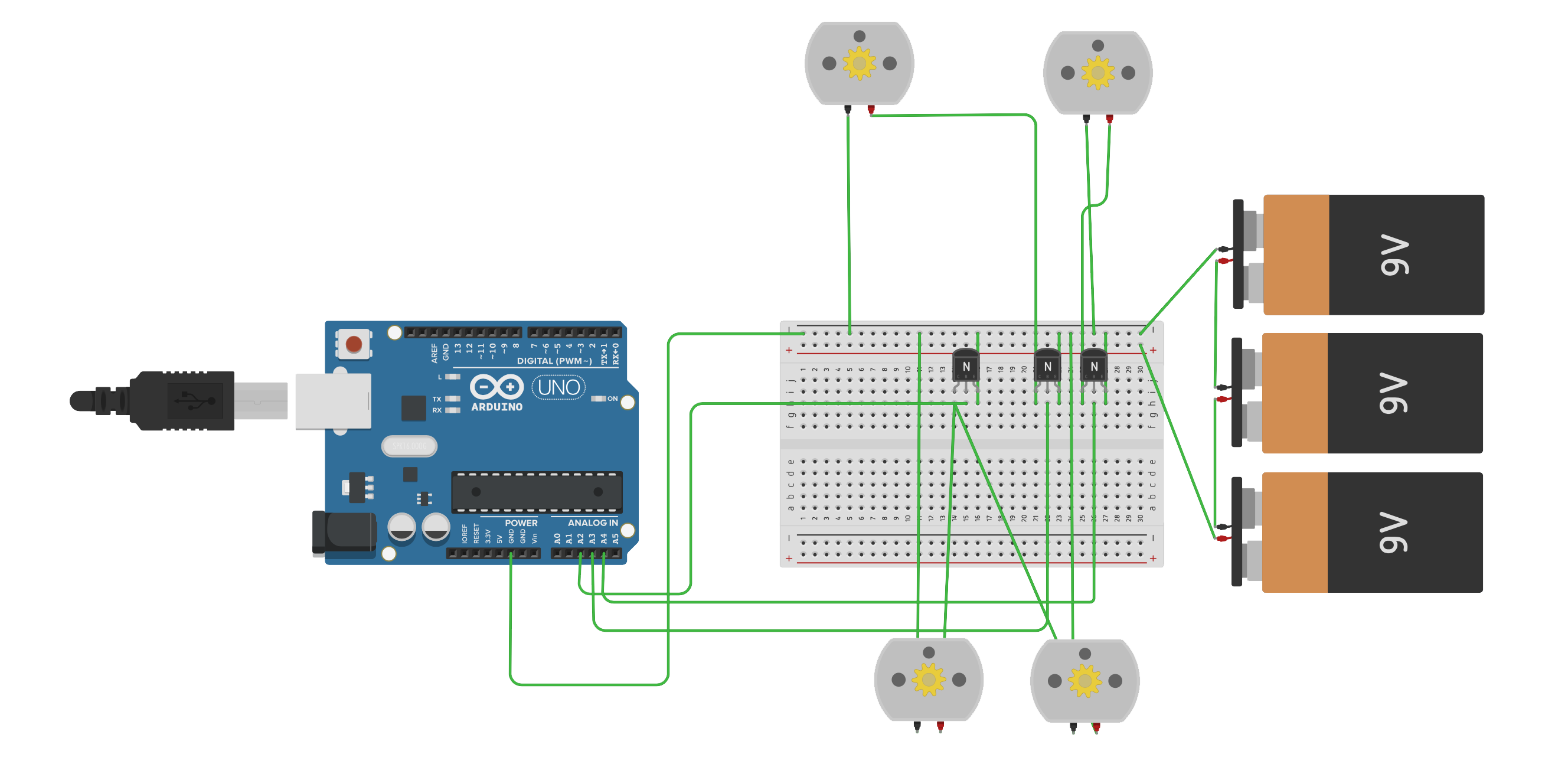
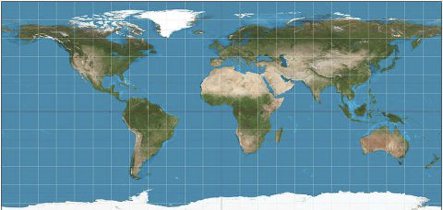


Figure 4.3: Circuit Design of the car.

1. **Technical Description of Mobile Application.**

**5.1 Methodology**

360-degree video needs to be manipulated before it could be displayed to the viewer, the basic methodology to manipulate the video in a viewable form is to map the vide on to a sphere, and the viewer would be looking at the sphere from it’s point of origin. The process is similar to mapping equirectangular projection of earth onto a globe.



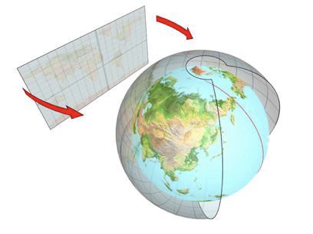


Figure 5.1: How an isometric image is mapped to a sphere.

**5.2 How is the Sphere Drawn**

We draw the Sphere using OpenGL. We can only draw triangles in OpenGL, by connecting Vertex, we’re able to draw triangles. Using more and more triangles, we’re able to build up a sphere that look smooth.

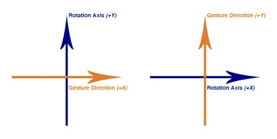


Figure 5.2: Building a Sphere with triangles.

**5.3 How the Video is mapped to the Sphere**

*Figure 2.4* does a good job at presenting how the video is mapped to the sphere, although the actual video isn’t bent around the sphere, but color, texture and co-ordinate information from different pixels of a frame from the video is extracted, and that information is then displayed on the sphere using shaders.

The process repeats itself for every frame of the video displaying the video in a spherical form, which when viewed from the origin of the sphere displays video feed all around it or a 360-degree view.

**5.4 How do we move around in the Video**

* **Using Gestures**

We get the touch distance in X and Y co-ordina tes, and for every pixel the user drags we move rotate the sphere a set amount of radians.

Figure 5.3: Arrows and Directions

* **Using Gyroscope**

Gyroscope approach is similar to the gestures approach, instead of getting the touch distance in X and Y, we receive radians moved in X and Y direction from the gyroscope and rotate the sphere accordingly.

1. **Constraints, Problems and Cost**

**6.1 Design and Implementation Constraints**

* + Latency and lag caused because of transmission of the VR video wirelessly. The VR system has to track movements of the head and render the new image to prevent lag. The VR system should perform the information exchange within 20 ms to maintain 60 frames per second. This metric is the Motion-to-Photon latency. It is the time needed for the user’s movements to reflect onto his screen.

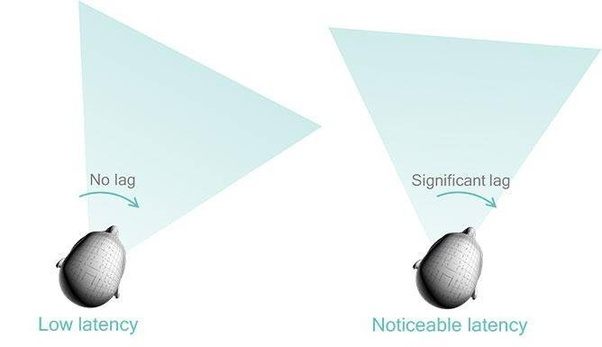


Figure 6.1 Latency and lag

Source: https://www.qualcomm.com/news/onq/2016/06/29/keeping-virtual-world-stable-vr

* + Software development for various platforms means requirement of various different frameworks and knowledge.
  + Accuracy of accelerometer, magnetometer and gyroscope may vary from device to device.

**6.2 Safety Issues**

The hardware should be treated carefully as extensive damage to the hardware might hinder the end user experience, functionality of the VR car might also be affected because of the damage. Driving very fast while wearing the VR headset might induce dizziness in the user’s body, hence longer sessions or sessions with very fast driving should be avoided.

**6.3 Security Issues**

Video Transmission over local networks might be a security issue if unsafe protocols used. Usage on public networks would be on users risk in case of breach of privacy.

**6.4 Cost estimation**

Table 1: Cost Estimation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **S. No** | | **Description** | |  | **Costs** |
|  |  |  | **A. Capital** | |  |  |
| 1 | |  | Equipment | |  | 40,000 |
|  |  |  | Total(A) | |  |  |
|  |  |  | **B. Consumable** | |  |  |
| 1 | |  | Raw materials, Consumables. | |  | 4,000 |
| 2 | |  | Contingency & Others (Repair & | |  | 4,000 |
|  | Maintenance) | |  |
|  |  |  |  |  |
|  |  |  | Total(B) | |  | 8,000 |
|  |  |  | **C. Miscellaneous** | |  |  |
|  | 1 |  | Miscellaneous | |  | 2,000 |
|  |  |  | Total(C) | |  | 2,000 |
|  |  |  | *Total(A+B+C)* | |  | 50,000 |
|  |  | | |
|  |  |  | Table 2: Cost Estimation of Capital Equipment | |  | |
|  | **Sr.** |  | **Capital** | | **Total** | |
|  | **No.** |  | **Equipment/Software** | | **(Rs. )** | |
|  | **1.** |  | Arduino Nano | | 500 | |
|  | **2.** |  | Odroid XU4 | | 6,500 | |
|  | **3.** |  | Raspberry Pi | | 3,500 | |
|  | **4.** |  | Camera(x2) | | 20,000 | |
|  | **5.** |  | Circuitry Accessories | | 2,000 | |
|  | **6.** |  | Vehicle Body | | 1,500 | |
|  | **7.** |  | Networking Hardware | | 5,000 | |
|  | **Total** |  |  | | 40,000 | |

1. **Timeline**

Table 3: Major Milestones

|  |  |
| --- | --- |
| **Milestone** | **Milestone Goal** |
| Concept Approval | Feasibility studies and basic system concepts have been approved by our mentor and further research into the project has started. |
| Requirement Review | Requirements details for the project are complete and further designing has started. |
| Design Review | Confirming that the design satisfies the project requirements and are capable to fully implement the system and are suitable for code input. |
| Test Plan | Test Plans are Adequate for the testing of all product features, are approved and are suitable for input to the development of test cases. |
| System Test | Software for the system has passed testing and is suitable for further input |
| Product Operational | The Software and Hardware are working the way they were indented too. |

**Gantt Chart**



Figure 7: Gantt Chart

**Glossary**

* Latency: Time taken for a single video frame to from the camera to display
* VR Headset: A virtual reality headset is a head-mounted device that provides virtual reality for the wearer. VR headsets are widely used with computer games but they are also used in other applications, including simulators and trainers. They comprise of a stereoscopic head-mounted display, stereo sound and head motion tracking sensors.
* Accelerometer: An accelerometer is a device that measures proper acceleration of the device movement in X and Y axis.
* Gyroscope: A gyroscope is a device used for measuring or maintaining orientation and angular velocity. The gyroscope measure the rate of change of a particular axis at the current moment in time. This means that to keep track of our angle, we need to sum all of the rates of change over a given period of time. We're essentially looking for the integral of our gyro data.

**Definitions, Acronyms and Abbreviations**

* GPU - Graphics Processing Unit.
* Objective C - A programming language.
* API - Application Programming Interface.
* OpenGL - Graphics API.
* SDK - Software Development Kit.
* iOS - Mobile Operating System.
* ARM - Advanced RISC Machine.
* IDE - Integrated Development Environment.
* HLS – HTTP live streaming.
* VR – Virtual Reality

**References**

[1] Yufeng Shan, UC Berkeley. Cross-Layer Techniques for Adaptive Video Streaming over Wireless Network 2005:2, 220–228

[2] Documentation for UV4L(video streaming service): [*http://www.linux-projects.org*](http://www.linux-projects.org)

[3] Jingwei Huang, Zhili Chen, Duygu Ceylan, Hailin Jin. “6 DOF VR Videos with a 360 Camera”, March 18-22, 2017, Los Angeles, CA, USA 978-1-5090-6647-6/17

[4] Nadine E. Miner, Sharon Stansfield, “An Interactive Virtual Reality Simulation System for Robot Control and Operator Training”, Intelligent Systems and Robotics Center

[5] Nadine E. Miner, Sharon Stansfield, “An Interactive Virtual Reality Simulation System for Robot Control and Operator Training”, Intelligent Systems and Robotics Center

[6]  Ray Wenderlich, ”OpenGL-ES Transformations gestures”, https://www.raywenderlich.com/50398/opengl-es-transformations-gestures

[7]  Boonsuk, Wutthigrai, "Evaluation of desktop interface displays for 360-degree video"(2011). *Graduate\_Theses\_and\_Dissertations*.10249. [*http://lib.dr.iastate.edu/etd/10249*](http://lib.dr.iastate.edu/etd/10249)

[8] Boonsuk, Wutthigrai; Gilbert, Stephen B.; and Kelly, Jonathan W.,” The Impact of Three Interfaces for 360-Degree Video on Spatial Cognition” (2012). Faculty Research & Creative Activity. Paper 13.