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: 16/11/18

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

LIST OF TABLES

LIST OF FIGURES

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.

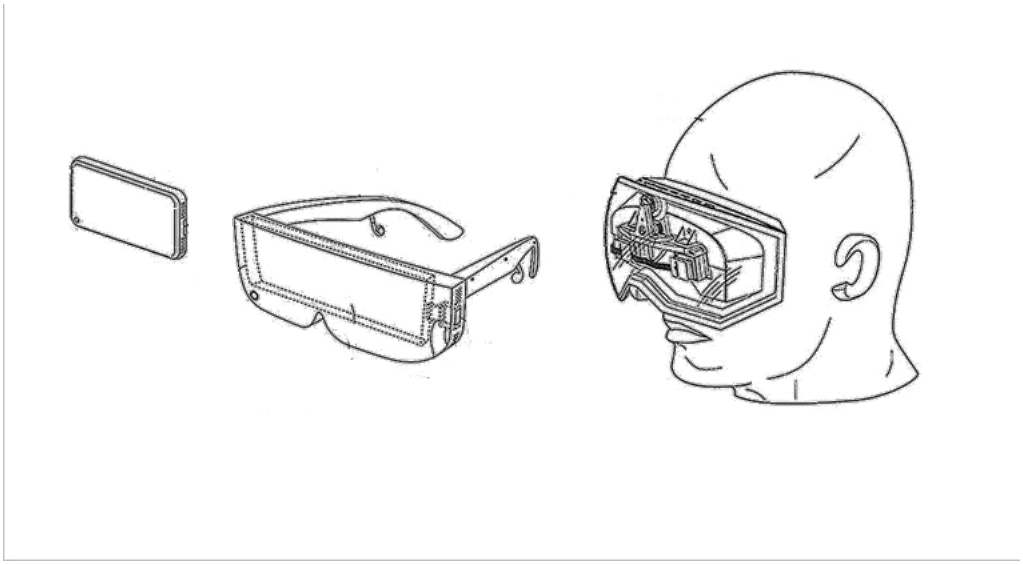


Figure 1.1: VR headset and smartphone

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 experience taking place within a simulated environment, that incorporates mainly auditory and visual, but also other types of sensory feedback like haptic. 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 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

2. 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 video 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.
- All the data except for video feed will be shared over MQTT channels.
- 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.

3. Technical Description

3.1 Software Environment

Xcode 9

Xcode is an IDE for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS, and tvOS. Using the iOS SDK, Xcode can be used to compile and debug applications for iOS that run on ARM architecture processors.

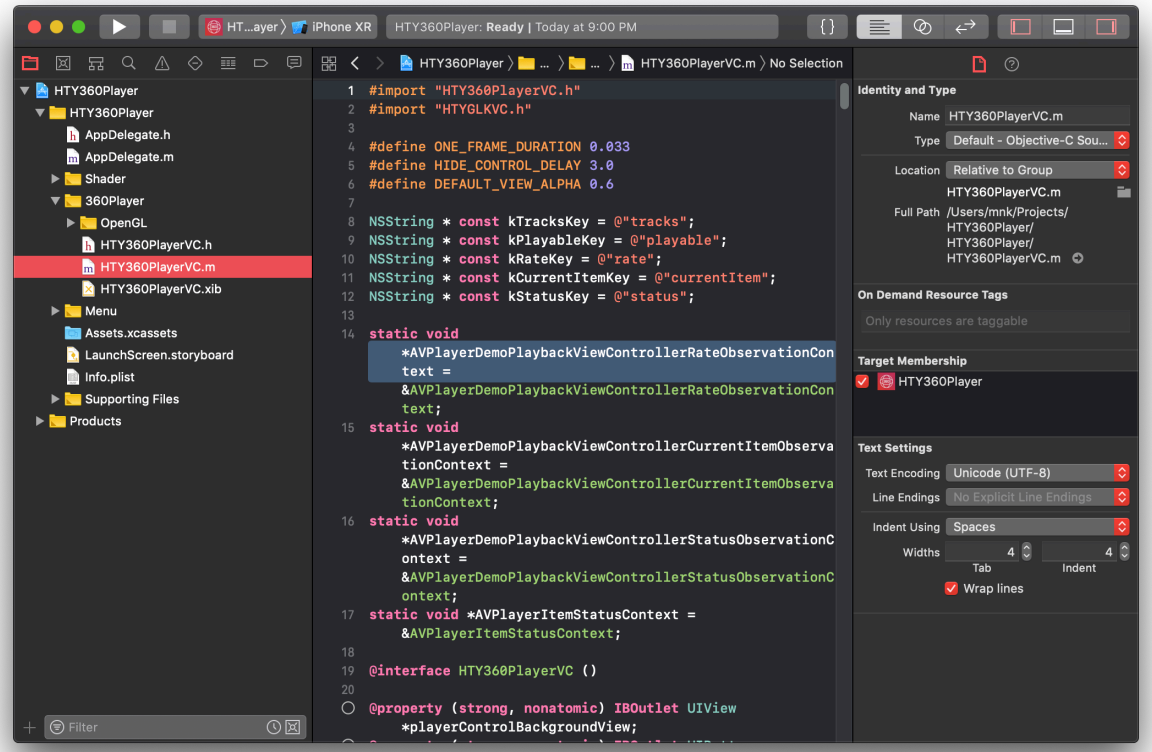


Figure 3.1 : X-Code with a project open.

Xcode was used to develop a significant part of the project, the mobile application which directly interfaces with the vehicle.

Objective-C and Swift

Objective-C is a general-purpose, object-oriented programming language that adds Smalltalk-style messaging to the C programming language and Swift is a further modified and advanced version of Objective-C. These are the main programming language used by Apple for the OS X and iOS operating systems, and these two are the languages used to develop the mobile application in Xcode.

Python

Python is an interpreted high-level programming language for general-purpose programming, python was used all-around the project for various tasks. Most significant use of python was in the software being run on the on-board computer on vehicle, responsible for interfacing with everything connected to the vehicle.

Open-GL

Open Graphics Library is a cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering. Open-GL had its significant involvement in rendering the 360-degree video in the mobile application.

HLS

HTTP Live Streaming is an HTTP-based media streaming communications protocol implemented by Apple Inc. as part of its QuickTime, Safari, OS X, and iOS software. HLS was used to stream the video off the vehicle.

MQTT

Message Queuing Telemetry Transport is an ISO standard publish-subscribe-based messaging protocol. It works on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. MQTT was responsible for a great amount of information exchange between the devices that occurs on the system, MQTT allowed for well-defined and distinct channels to share information without congesting the network too much.

3.2 Hardware Environment

Arduino Uno

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. An Arduino Uno was used in the project as the microcontroller responsible for driving the vehicle.

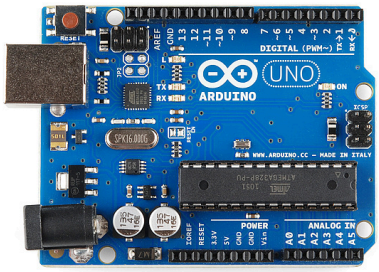


Figure 3.2: Arduino

Source: Adafruit

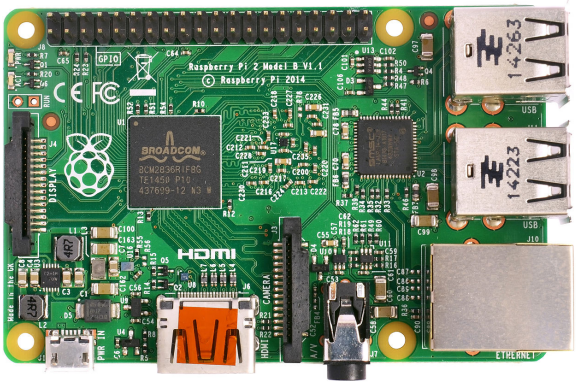


Figure 3.3: Raspberry Pi

Source: Raspberrpi.com

Raspberry Pi 3 Model B

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. Raspberry Pi 3 Model B is used as the on-board computer on the vehicle, responsible for everything from data transfer to video encoding.

PICAM360

PICAM360 is a 360-degree panoramic camera with the open source hardware based on Raspberry Pi. PICAM360 captures upto 3.7k footage and upto 30fps and supports streaming out of the box. Two units of the PICAM360 are being used in the project.

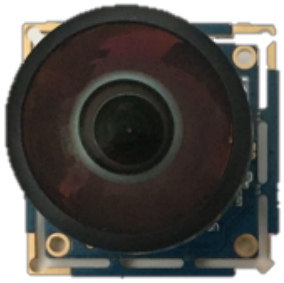


Figure 3.4: PICAM360

Source: PICAM.com

Gyroscope

Gyroscopes are used for measuring orientation and angular velocity. Gyroscopes implement their functionality using the principle of conservation of angular momentum.

Figure 3.5: A Gyroscope

Source: ThinkGeek

As shown in the diagram above, a gyroscope consists of a spinning wheel that is supported on an axis that moves freely. Precession is defined as the change in the orientation of the rotational axis of a body that is rotating. A pivot that allows rotation around one axis supports the spinning wheel. The axis is termed as gimbal. Having two gimbals, gives the wheel or the rotor three degrees of rotational freedom. When the rotor of the gyroscope is being spun, the gyroscope will point in same direction. Gyroscopes work through precession. The rotor rotates on its axis, when a force is applied on the wheel that attempts to rotate its spin axis; the gyroscope reacts to that force in an axis perpendicular to that applied force. When force is applied to the gimbal, the gyroscope will move towards the left while the bottom will move towards the right. This causes precession.

Accelerometer

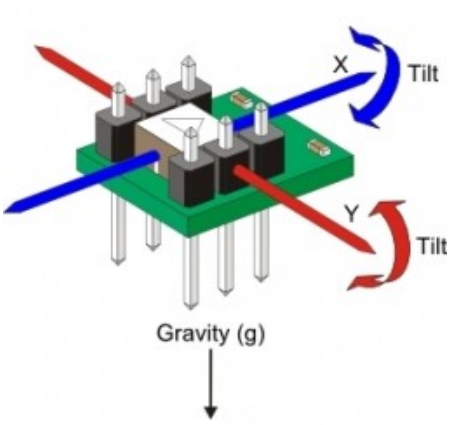


Figure 3.6: An accelerometer

Source: IoT One

Accelerometer is an electromechanical sensor device that measures acceleration forces. Acceleration is defined as the change in velocity over time.

Accelerometer consists of a piezoelectric device and a sensor to measure capacitance. The piezoelectric uses microscopic crystal structures for its function. These tiny structures become stressed in presence of acceleration forces and create a voltage. The accelerometer then interprets this voltage to identify the value velocity and orientation.

4. 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.

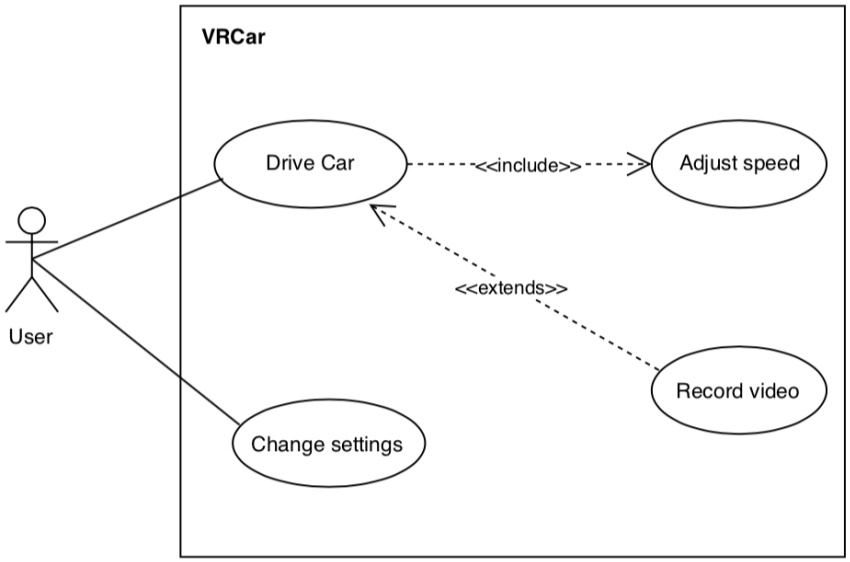


Figure 4.1: Use-Case Diagram

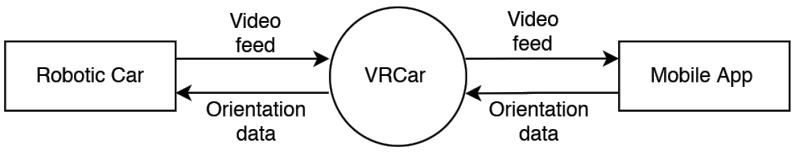


Figure 4.2: Context Diagram

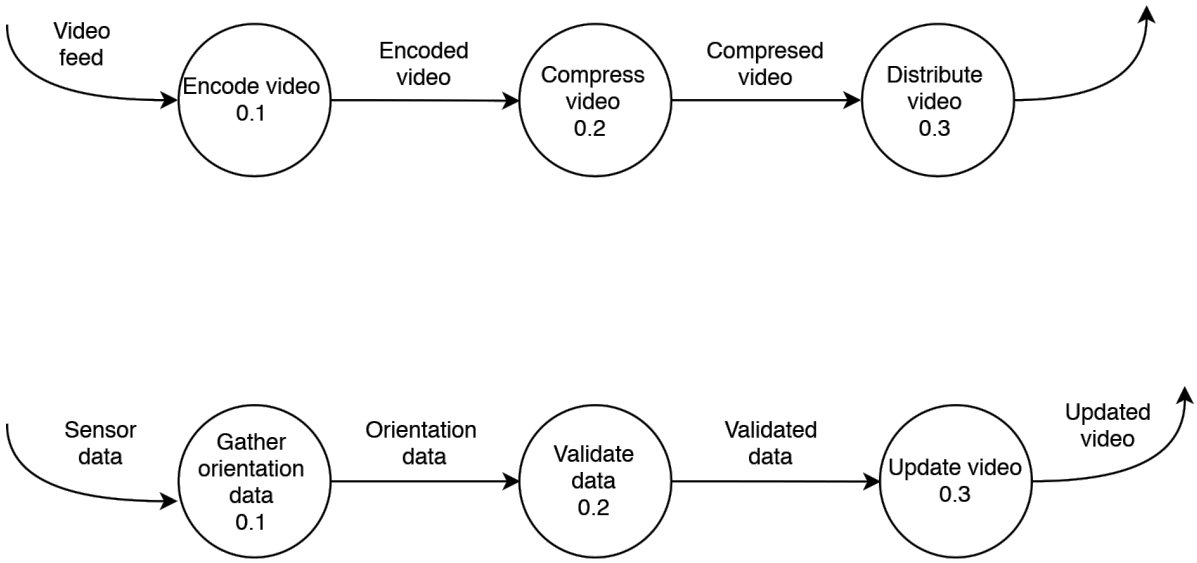


Figure 4.3: DFD Level 1

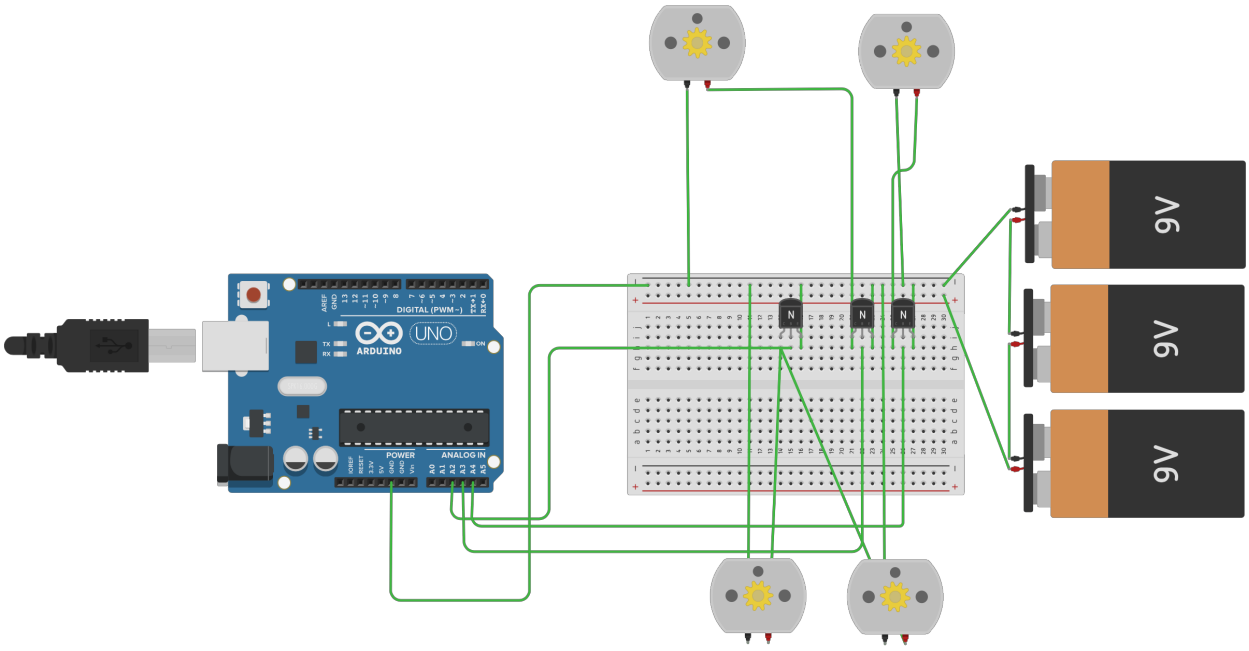
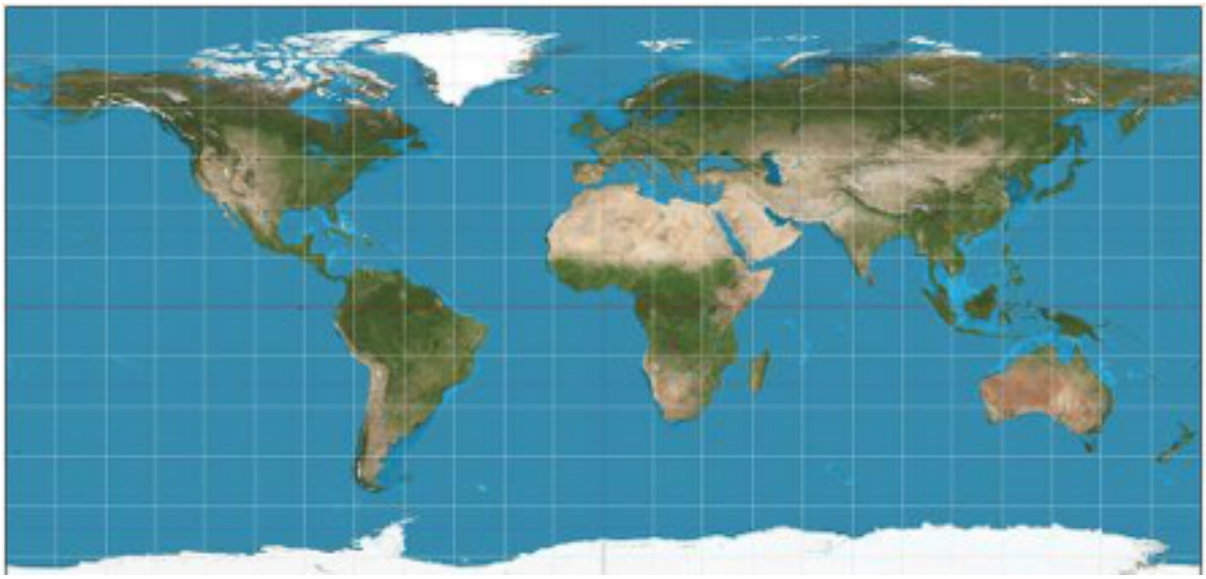


Figure 4.4: Circuit Design of the car.

5. 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.



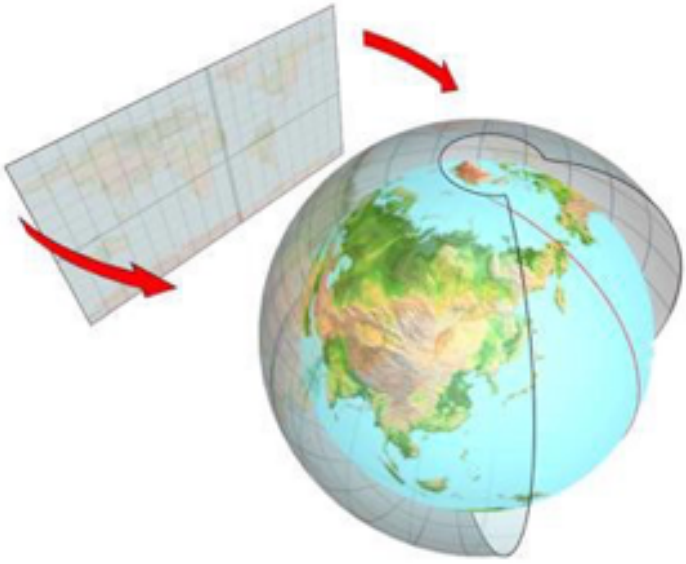


Figure 5.1: How an isometric image is mapped to a sphere.

Source: Wikipedia

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. Source: Medium

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.

5.5 Core Motion

Core Motion is a framework by Apple that allows you to access the values of motion and environmental sensors present in the iOS devices. Some of these devices include accelerometers and gyroscopes, and from the pedometer, magnetometer, and barometer. This framework provides both the raw and the processed values. The device-motion service by apple provides processed motion data for the iOS Device. It processes Raw accelerometer and gyroscope data must be processed to remove bias from other factors, such as gravity to give refined data that can be used with no further processing required.

The attitude of the device gives The device's orientation relative to a known frame of reference at a point in time in the form of roll pitch and yaw.

- Rotation around the front-to-back axis is called roll.
- Rotation around the side-to-side axis is called pitch.

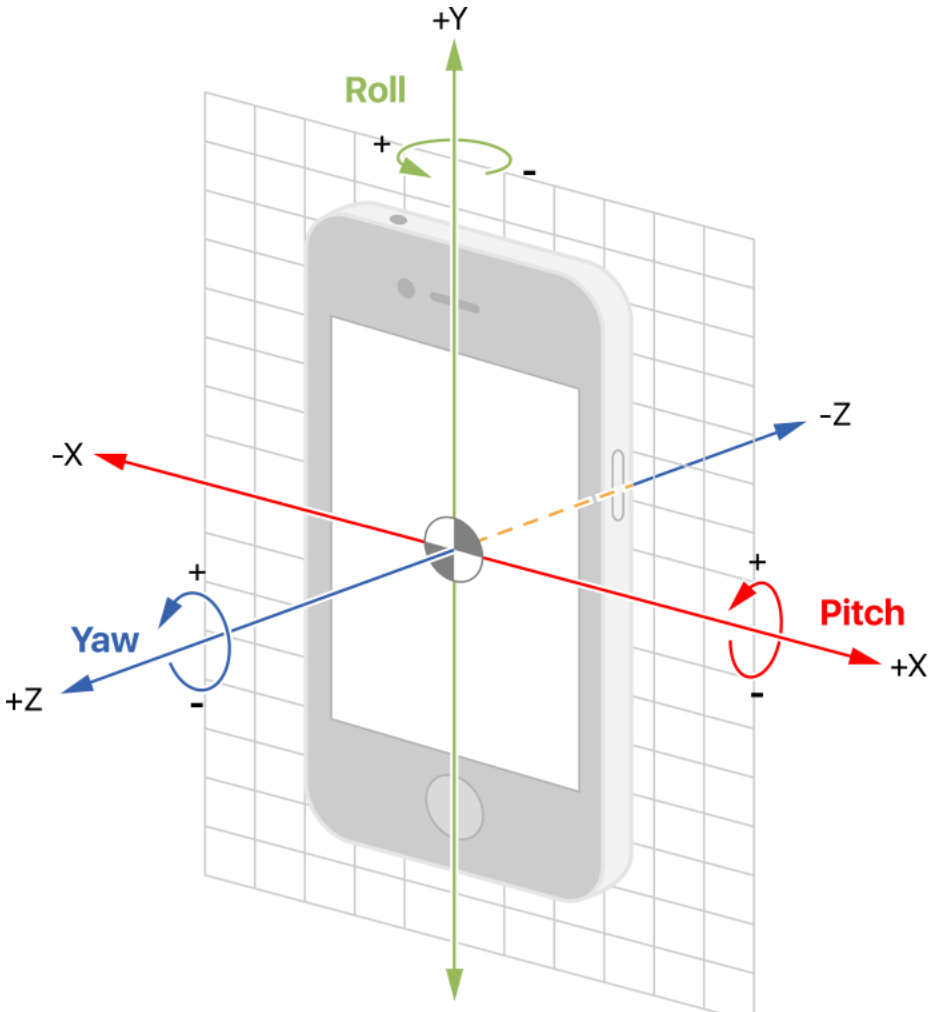


Figure 5.4: Pitch and Yaw

Source: Apple

5.6 Core Data

Core Data is a object and persistence framework provided by Apple for iOS Devices for storing data on phone locally. It allows data organized by the relational entity–attribute model to be manipulated using higher level objects representing entities and their relationships. We use core data for storing user preferences locally on the device

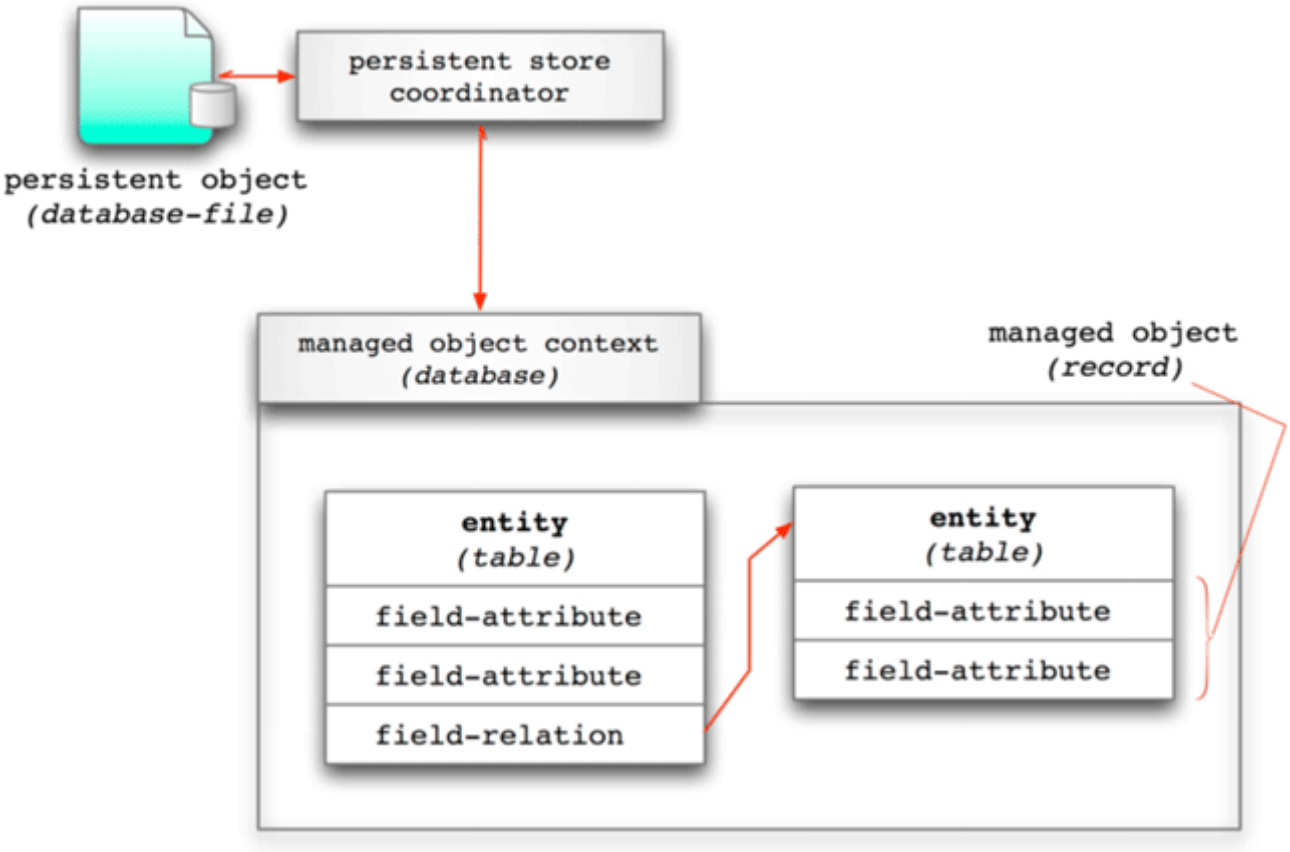


Figure 5.5: How Core Data Works

Source: DRdobbs

5.7 App Screenshots

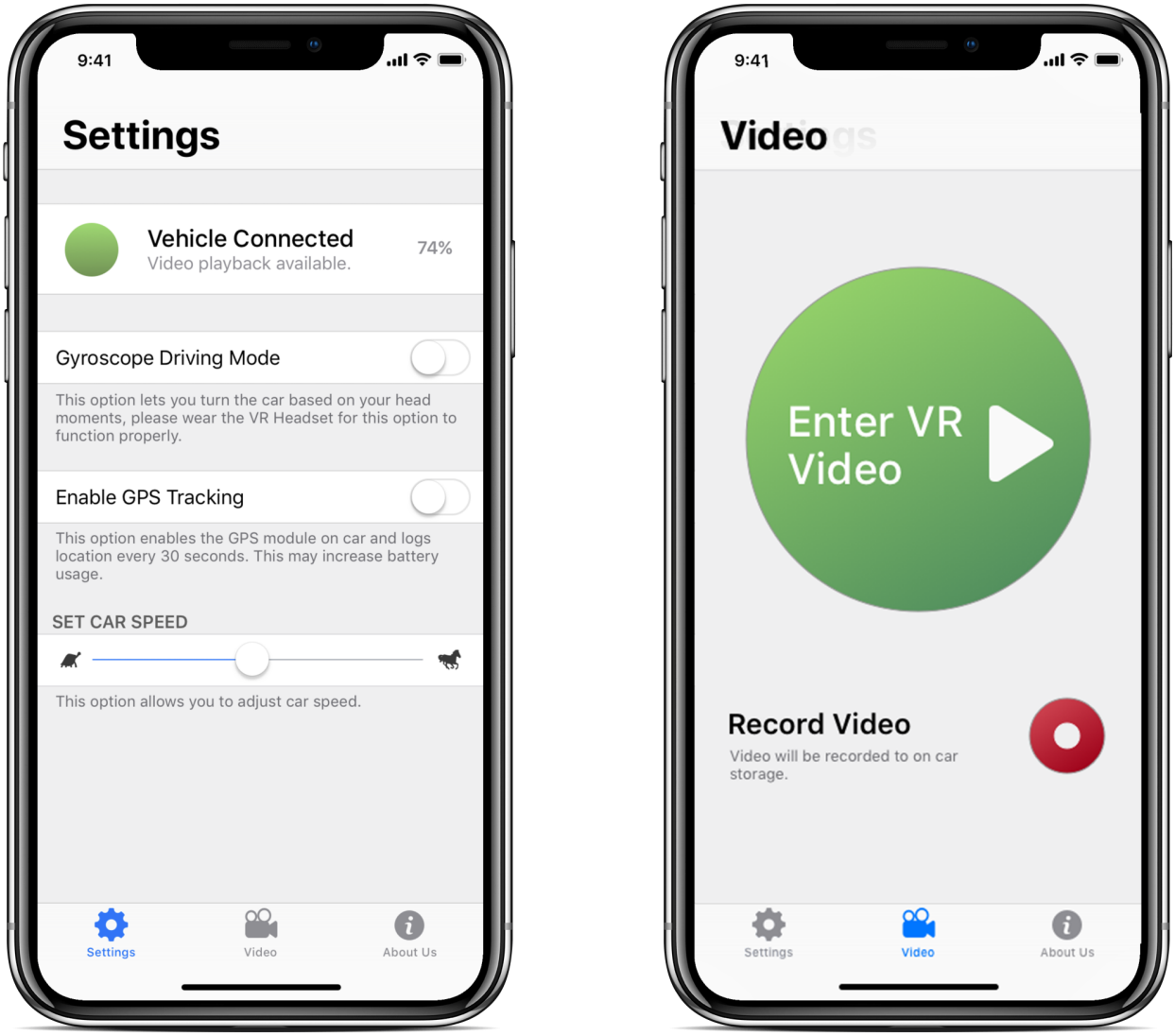


Figure 5.3: Screenshots of the App

6. Technical Description of the Network Structure

6.1 Network Structure

There are two protocols responsible for the data transfer and dealing with networking side of things. The use of two protocols essentially created two separate data streams responsible for different things.

- **HTTP/HLS:** Responsible for dealing with video transmission.
- **MQTT:** Responsible for pretty much everything else.

6.2 HTTP/HLS Structure

The raw input footage when input is first encoded to an encoding schema suitable for streaming, MPEG-2 in our case. The encoded footage is then subject to a stream segmenter which segments up the input encoded footage into various smaller clips with an extension .m3u8, while keeping an index which tracks all the .m3u8 whiles.

The segmented files are then sent to the receiving end, which are then further broken up. The .m3u8 files consists of an index and a series .t3 files which contain individual frames stored in them.

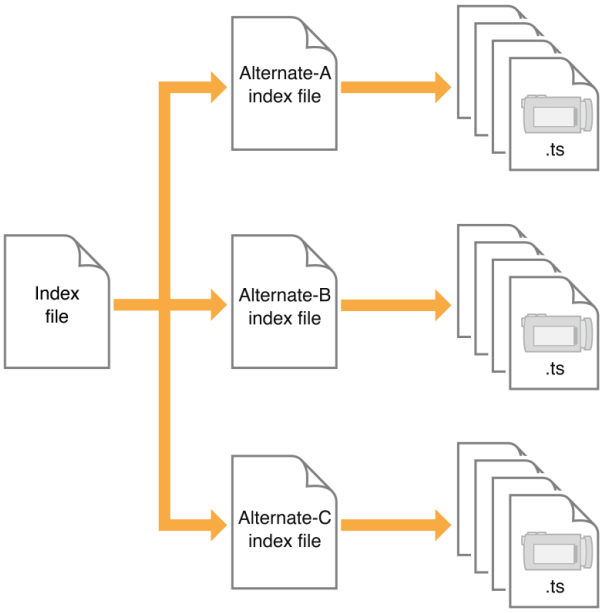


Figure 6.1: Index Files of HLS

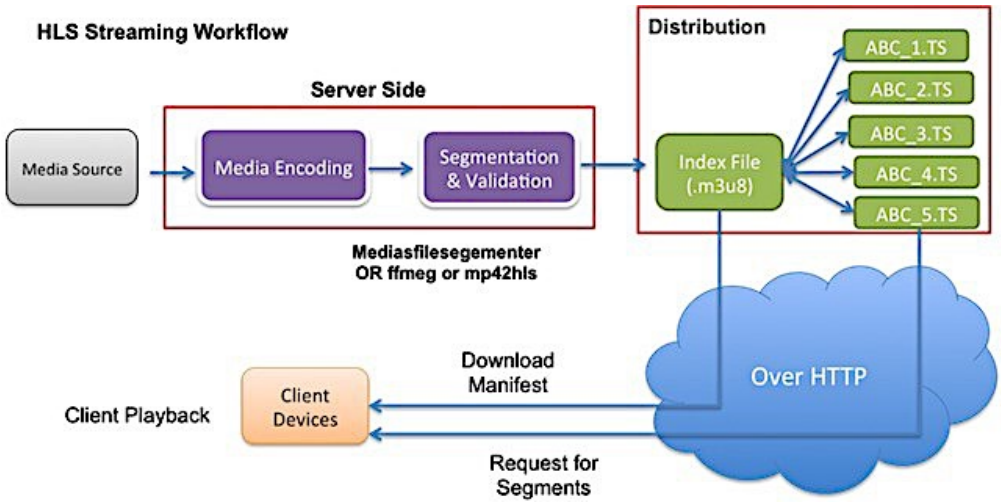
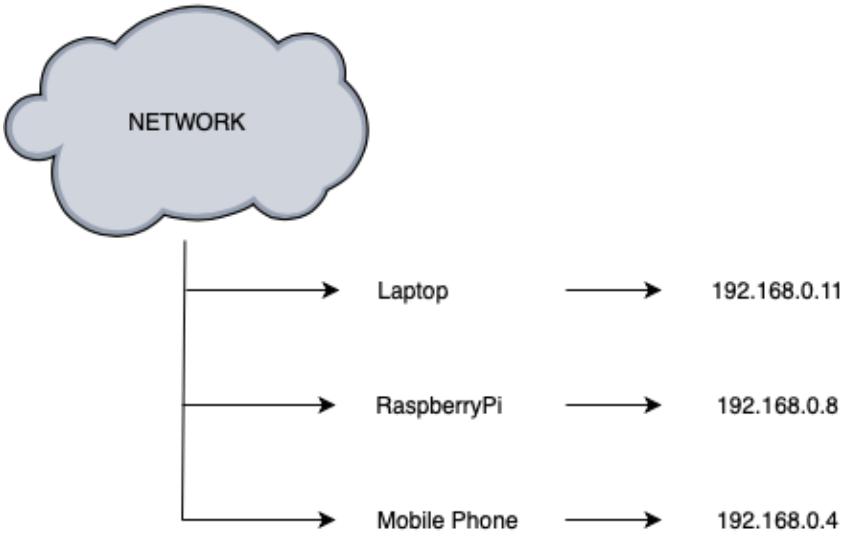


Figure 6.2: How HLS works

Source: Media Entertainment Media

6.3 MQTT Structure

MQTT has been responsible for all the data transfer between the vehicle and connected accessories. To run an MQTT server we run Mosquitto broker on the Raspberry Pi to which devices on the same network can connect to and further interact with the Vehicle and control it.



MQTT uses a subscription model where a publisher can publish messages to any specified topic and any subscriber can read off from any topic they specify.

Figure 6.3: Network Structure

Topics/Channels used in the MQTT structure:

- SettingsChannel
- DriveChannel

SettingsChannel:

Responsible for altering the settings on the vehicle, a JSON packet is sent to the subscriber which contains the altered settings. JSON is then parsed on the receivers end and certain variables are then set accordingly. The sent JSON looks like:

{GyroMode : 1/0,
Record : 1/0}

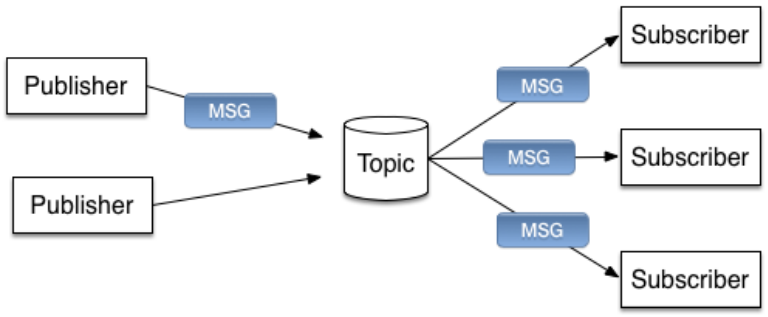


Figure 6.4: MQTT Topic model

DriveChannel:
Responsible for commands which move the vehicle.

- Straight – When the character '1' is sent to the DriveChannel the vehicle moves straight until a Stop command is sent.

- Left – When the character '2' is sent to the DriveChannel the vehicle turns left until a Stop command is sent.

- Right – When the character '3' is sent to the DriveChannel the vehicle turns right until a Stop command is sent.

- Backwards – When the character '4' is sent to the DriveChannel the vehicle moves in the reverse direction until a Stop command is sent.

- Stop – Character '0' is sent to signal the vehicle to Stop.

- SetSpeedLow – When the character '7' is sent to the DriveChannel the vehicle is to a slow speed.

- SetSpeedMid – When the character '8' is sent to the DriveChannel the vehicle is to a medium speed.

- SetSpeedHigh – When the character '9' is sent to the DriveChannel the vehicle is to a high speed.

When any motion key is pressed on any connected keyboard, then the said command is sent to the DriveChannel, and when an release key event is detected, the Stop command is sent. The gyroscopic driving mode works similarly as well. In the gyroscopic driving mode, the mobile device senses the device movement either towards Left/Right or Up/Downwards, and then the mentioned commands are sent based on the detection.

7. Constraints, Problems and Cost

7.1 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.

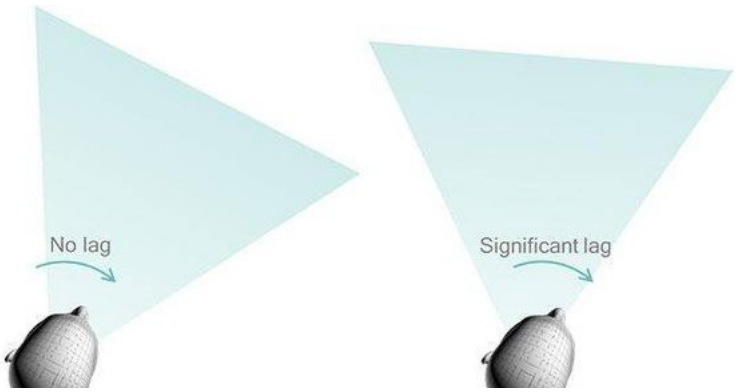


Figure 7.1 Latency and lag

Source: Qualcomm

- 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.

7.2 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.

7.3 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.

7.4 Cost estimation

Table 1&2: Cost Estimation

S. No	Description	Costs
	A. Capital	
1	Equipment	40,000
	Total(A)	40,000
	B. Consumable	
1	Raw Materials, Consumables	4,000
2	Contigency & Others	4,000
	Total(B)	8,000
	C. Miscellaneous	
1	Miscellaneous	2,000
	Total(C)	
	Total(A+B+C)	50,000

Sr.No	Equipment/Software	Costs(Rs.)
1	Arduino Uno	500
2	Raspberry Pi(x2)	10,000
3	Camera(x2)	20,000
4	Circuitry Accessories	2,000
5	Vehicle Body	1,500
6	Networking Hardware	5,000
Total		40,000

8. 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

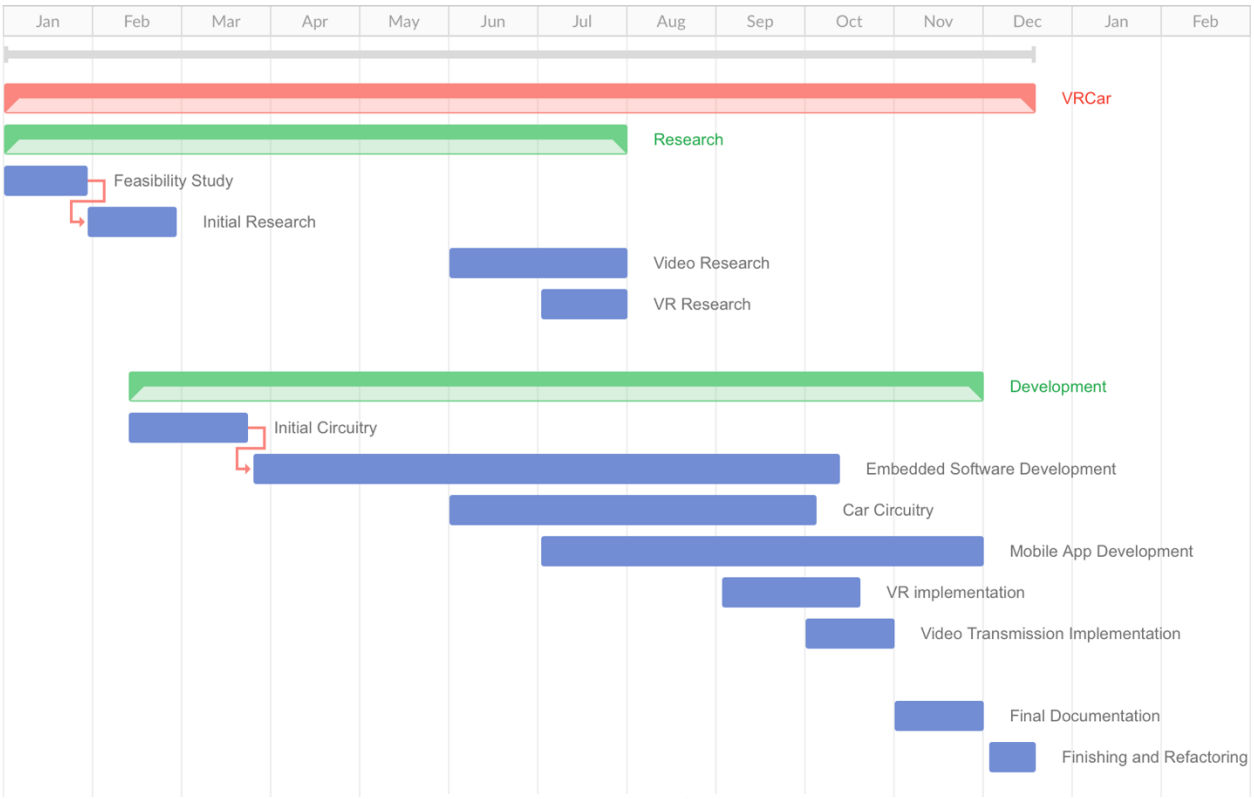


Figure 8: Gantt Chart

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

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

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

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

Nadine E. Miner, Sharon Stansfield, "An Interactive Virtual Reality Simulation System for Robot Control and Operator Training", Intelligent Systems and Robotics Center

Nadine E. Miner, Sharon Stansfield, "An Interactive Virtual Reality Simulation System for Robot Control and Operator Training", Intelligent Systems and Robotics Center

Ray Wenderlich, "OpenGL-ES Transformations gestures", https://www.raywenderlich.com/50398/opengl-es-transformations-gestures

Boonsuk, Wutthigrai, "Evaluation of desktop interface displays for 360-degree video"(2011). *Graduate_Theses_and_Dissertations.*10249. *http://lib.dr.iastate.edu/etd/10249*

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.