|  |
| --- |
|  |
| Capstone Project Document |

**Mini Explorer System**

----------------------------------------------------------------

|  |  |  |
| --- | --- | --- |
| **MEx Team** | | |
| **Group Members** | Luyện Bảo Anh | SE03747 |
| Phạm Minh Hoàng | SE03769 |
| Lê Xuân Hướng | SE03388 |
| Phùng Đức Luật | SE03164 |
| Đỗ Cao Phong | SE03196 |
| Đặng Ngọc Tú | SE03591 |
| **Supervisor** | Hoàng Xuân Sơn | |
| **Project code** | MEx | |

**- Hoa Lac, 01/2017 –**

Contents

[1. INTRODUCTION 3](#_Toc490131398)

[**1.1** Purpose 3](#_Toc490131399)

[**1.2** Acronyms and Definitions 3](#_Toc490131400)

[**1.3** The People 3](#_Toc490131401)

[1.3.1 Supervisor 3](#_Toc490131402)

[1.3.2 Team member 4](#_Toc490131403)

[**1.4** Project information 4](#_Toc490131404)

[**1.5** The idea 4](#_Toc490131405)

[**1.6** Proposal of system 4](#_Toc490131406)

[1.6.1 The scope 4](#_Toc490131407)

[1.6.2 Existing system 5](#_Toc490131408)

[1.8 Benefit from project 6](#_Toc490131409)

[1.8.1 For team members 6](#_Toc490131410)

[1.8.2 For community 6](#_Toc490131411)

[**1.9** Critical assumption and constraints 6](#_Toc490131412)

[1.9.1 Critical assumption 6](#_Toc490131413)

[1.9.2 Critical assumption 6](#_Toc490131414)

[**1.10** Potential risks 6](#_Toc490131415)

[2. Project management plan 7](#_Toc490131416)

[3. Requirements 17](#_Toc490131417)

[3.1.1 User requirement 17](#_Toc490131418)

[3.1.2 Car requirement 17](#_Toc490131419)

[3.1.3 Android application requirement 17](#_Toc490131420)

[3.1.4 Remote Controller requirement 17](#_Toc490131421)

[3.2.1 Interface requirement 17](#_Toc490131422)

[3.2.1.1 User Interface 18](#_Toc490131423)

[3.2.1.2 Hardware interface 18](#_Toc490131424)

[3.2.1.3 Software interface 18](#_Toc490131425)

[3.3.2 Functional requirement 19](#_Toc490131426)

[3.3.2.1 Connect smartphone 19](#_Toc490131427)

[3.3.2.2 Disconnect Smartphone 20](#_Toc490131428)

[3.3.2.3. Enter the VR mode 21](#_Toc490131429)

[3.3.2.4. Enter Android controller mode 22](#_Toc490131430)

[3.3.2.5 Android Speed control 23](#_Toc490131431)

[3.3.2.6 Android Gear Switch 24](#_Toc490131432)

[3.3.2.7 Android light signal 25](#_Toc490131433)

[3.3.2.8 Android Sound signal 26](#_Toc490131434)

[3.3.2.9 Remote Connect 27](#_Toc490131435)

[3.3.2.10 Remote Speed control 28](#_Toc490131436)

[3.3.2.11 Remote Gear Switch 29](#_Toc490131437)

[3.3.2.12 Remote Light Signal 30](#_Toc490131438)

[3.3.2.13 Remote Sound Signal 31](#_Toc490131439)

[3.2.3 Non-Functional Requirement 32](#_Toc490131440)

[3.2.3.1 Reliability 32](#_Toc490131441)

[3.2.3.2 Availability 32](#_Toc490131442)

[3.2.3.3 Maintainability 32](#_Toc490131443)

[3.2.3.4 Performance 32](#_Toc490131444)

[3.3.1 Hardware 32](#_Toc490131445)

[3.3.2 Software and tool 33](#_Toc490131446)

[4. IMPLEMENTATION 33](#_Toc490131447)

[4.1. Proposed system architecture 33](#_Toc490131448)

[4.2. Analysis and selection of tools, devices 34](#_Toc490131449)

[4.2.1. Motor controller 34](#_Toc490131450)

[4.2.2. Servo car 35](#_Toc490131451)

[4.2.3. Servo camera 35](#_Toc490131452)

[4.2.4. Remote Controller 36](#_Toc490131453)

[4.2.5. Supply power for Raspberry PI 36](#_Toc490131454)

[4.3 Interface design 36](#_Toc490131455)

[4.3.1. Hardware design 36](#_Toc490131456)

[4.3.2. Software design 36](#_Toc490131457)

[4.4. Testing 36](#_Toc490131458)

# INTRODUCTION

## Purpose

This document is created as the introduction for project MEx – our Capstone Project at FPT University. In this document, we will descript the overview of some existing systems, the initial idea for our project, a brief description about our expected system and some potential risks, critical assumptions, constrains. Moreover, this document also shows opportunities what it offers for users.

## Acronyms and Definitions

|  |  |
| --- | --- |
| **Acronym & Abbreviation** | **Definition** |
| MEx | Mini Explorer |
| FU | FPT University |
| VR | Virtual Reality |

Table 1-1: *Definitions and Acronyms*

## The People

### Supervisor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Full name** | **Phone** | **Email** | **Title** |
| Supervisor | Hoàng Xuân Sơn | 0936232008 | [SonhHX@fe.edu.vn](mailto:SonhHX@fe.edu.vn) |  |

1. Supervisor’s information

### Team member

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Full name** | **StudentID** | **Phone** | **Email** | **Role** |
| 1 | Luyện Bảo Anh | SE03747 | 01672788452 | [anhlbse03747@fpt.edu.vn](mailto:anhlbse03747@fpt.edu.vn) | Team Leader |
| 2 | Phạm Minh Hoàng | SE03769 | 0904882411 | [hoangpmse03769@fpt.edu.vn](mailto:hoangpmse03769@fpt.edu.vn) | Team Member |
| 3 | Lê Xuân Hướng | SE03388 | 01649132648 | [huonglxse03388@fpt.edu.vn](mailto:huonglxse03388@fpt.edu.vn) | Team Member |
| 4 | Phùng Đức Luật | SE03164 | 01656885023 | [luatpdse03164@fpt.edu.vn](mailto:luatpdse03164@fpt.edu.vn) | Team Member |
| 5 | Đỗ Cao Phong | SE03196 | 0979064208 | [phongdcse03196@fpt.edu.vn](mailto:phongdcse03196@fpt.edu.vn) | Team Member |
| 6 | Đặng Ngọc Tú | SE03591 | 0868463132 | [tudnse03591@fpt.edu.vn](mailto:tudnse03591@fpt.edu.vn) | Team Member |

1. Team member information

## Project information

* Project name: Mini Explorer System
* Project code: MEx
* Project group name: MEx Team
* Product type: Embedded System
* Timeline: From May 8th to August 26th, 2017

## The idea

Nowadays, the rapid development of technology has a strong impact on the life of human beings. Along the rapid expansion of economic, the improvement of living standard, the demand of people about a comfort, safe and convenience life, car is going to one of main means of transport . But in Vietnam, the prices of cars are still too high for people to own one. Therefore, they have to take driving courses so that they can practice in the real car at relatively high prices. Not to mention, during the practice, can cause accidents for the user when they are not proficient yet.

MEx is the idea of first-person view - driving simulation system through virtual reality (VR) technology. The user controls an automobile model by wheel controller, pedals as in the real car. The camera will be set up and provide first-person view to a virtual reality lens to observe all vehicle movements.

The system simulates the whole process of driving a car so that user can learn how to drive easily at home rather than going to the driving courses.

## Proposal of system

### The scope

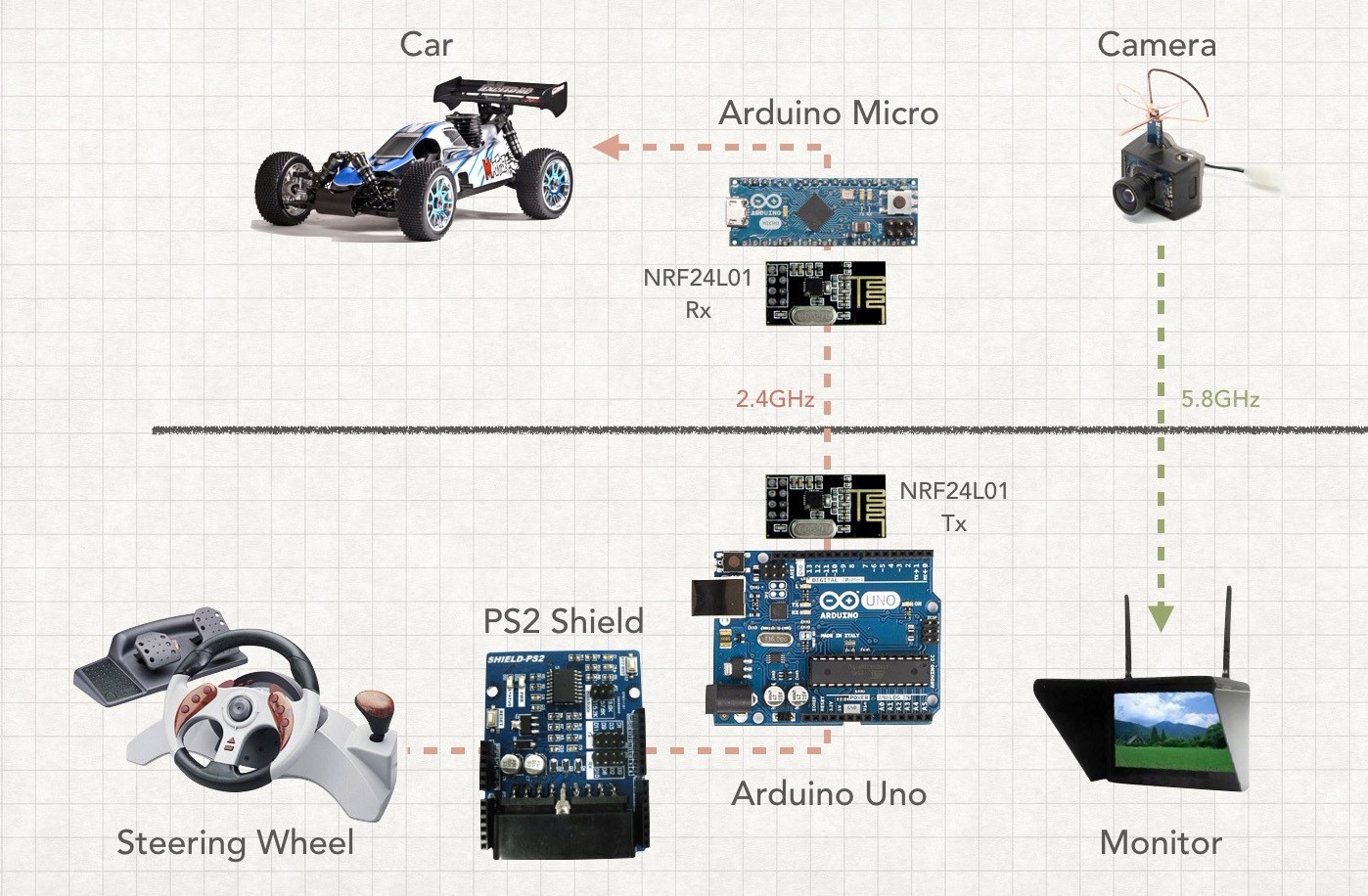
The scope of MEx is a prototype of control device. It includes both hardware and software. Finally, product must be satisfied some below specification.

* Streaming video with following feature:
  + Frame rate: 24 frames/second.
  + Delay: Depend on network rate.
  + Resolution: 640x480
* Controllers and models must have minimum functions such as running (forward, backward), steering, braking.
* Working on the terrain is relatively flat, not too rough, not too complex, no waves or interference obstructions.
* Tracking the motion of the head to provide the most sensible viewing angle

### Existing system

Recently, there are various products like MEx with good functions, attractive design interface. Below are some of these products:

* + - 1. *FPW Driving – Drive an RC Car with First person view*

**** FPW Driving is a prototype of Paul Yan – Arduino Team member. With an old PS2 wheel controller, two Arduinos, a mini FPV camera, and a headset as a standalone monitor, he made a system to control a RC car with first-person view very smoothly. The RC car–which is equipped with a Micro–interfaces with the wheel using an Uno and a PS2 Shield. Both Arduinos communicate via a pair of nRF24L01 modules.

***Figure 1****: FPW-Driving Diagram*

Without tracking the movement of the head, this system can only provide the front sight from straight view of the car. It will become difficult for users when they want to look to the right or left. This is the biggest shortcoming of Yan’s FPW Driving.

* + - 1. *RACEROOM with Oculus Rift*

Raceroom is the one of the most typical driving simulation games. This game offers players a system of more than 20 arcades and more than 60 vehicles to enter the race. With Oculus Rift (VR lens from Microsoft), playing Raceroom will be more fun and realistic with live sound and various gameplay modes.

Player can skip racing mode, and practice driving in a virtual environment. With Driving Force from Logitech or steering wheel from other suppliers, Raceroom would be suitable for those who want to practice driving.

***Figue 2:*** *Screen from Playing Raceroom with Oculus Rift*

As a product that has been commercialized, Racerook is being sold on Steam for around $ 20. Along with the expensive equipment that comes with the Oculus glass, Logitech's controllers make the price of this kit close to $2000. It’s very expensive for drive learner.

## Benefit from project

### For team members

* Have more experiences in working in project, project management.
* Have more knowledge about Arduino, Raspberry Pi, Android and mechanical.
* Improve skill about communicate with team members and how to work in team more effective.

### For community

* Have a new feeling of driving.
* Learn or get more driving practice anytime.
* Explore the distant area without the need for actual movement.

## Critical assumption and constraints

### Critical assumption

* Training: Developers can self-training Arduino and Raspberry Pi Programming with Python in 3 Weeks.
* Human resources: Assume that all members in team have a good healthy to work

### Critical assumption

* Time & Deadline: We must complete task on time. We work on 14 weeks, each member works 5 hours/day and 5 days/week. We do not have more time for us to complete developing and deliver application to teachers. Besides, we have to submit report documents before deadline to teacher can review.
* Quality: The products must be run well
* Process: We have to follow the software processing of FPT Software
* Human resources: There are 6 member in our team, each member have to study 4 subjects at school.

## Potential risks

After studying about this project, we find out some problem that we may be encountered:

* Under-estimate scope and time or miss deadline because lack of experience in group working, managing and controlling work.
* Equipment got broken because of careless or accident.
* Human resources: Team member cannot complete their works because of health reasons, key member leave team or un-cooperating on team.
* Change requirements: Requirement changed when some functions cannot be completed or some technologies is not suitable.

# Project management plan

**2.1. Definition Problem**

The report 1 is clearly specified reason why MEx project was chose to develop. It is an overview concept about MEx system and be discussed some main function of existing system.

You now have the knowledge of the system’s scope. This document will present project planning to get the target. All the tasks and time to implement, the resource of the system, and the risk maybe meet during development.

***2.1.1. Name of this Capstone Project***

This Capstone project named Mini Explorer System, abbreviated as MEx.

1. ***Boundaries of the System***
2. *Boundaries of the System*

The system under development of this Capstone Project will include:

* The controller has the task of sending the request via wireless, saving control information, controlling device.
* Wireless is an information bridge between the controller and car.
* A central circuit board in the car has responsible for data exchange with the gateway through Arduino to transmit, receive and process information from user.
* User manual, Test Document
* Design circuit broad, Design Document
* Source code Android App and Arduino

*2. Development Environment*

Below is the list of hardware and software requirements needed for development environment:

***Hardware requirements***

o Develop:

* + Arduino/WeMOS
  + Raspberry Pi
  + Sensor, servo motor, resistors, capacitors, wire…
  + Personal Computers with 4 Gigabytes of RAM or more
  1. Test:
     + Personal Computers

***Software requirements***

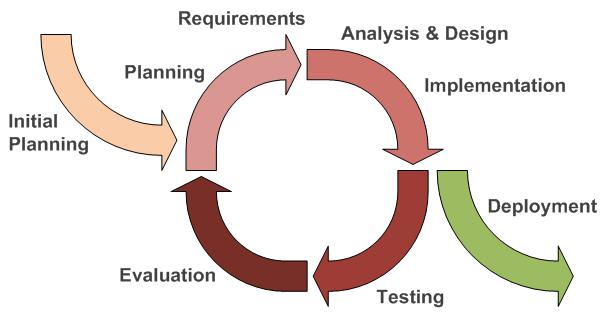
o Operating System: Windows 8.1, 10 Pro – 64bit

o Design software: Proteus 7.8

o IDEs: Android Studio v5.0 and SDK tools, JDK 7, Arduino-1.6.4-windows,Python

o Document: Microsoft Office 2016, Microsoft Project 2016

1. **Project organization**
   1. ***System Process Model***



*Figure 2-1: Iterative and Incremental Software Process Model*

This figure above describes the information and products flow lifecycle process model. MEx project uses the Iterative and Incremental Software Process Model.

Iterative and Incremental Software Process Model is a method of software development that is model around a gradual increase in feature additions and a cyclical release and upgrade pattern.

The Iterative and Incremental Software Process Model is most use when the scope of the project is big, the major requirements were defined clearly, some more detail will be added in time, and for the newbie group in software development.

By using this software process model, we break down the developing system task into series of smaller tasks are completed separately, evaluated, and subsequently re-worked until the system’s performance adequately. In addition, the iterative model is easier than other models when the issues were discover. They are feedback to the team, and solution found while the project is still in development.

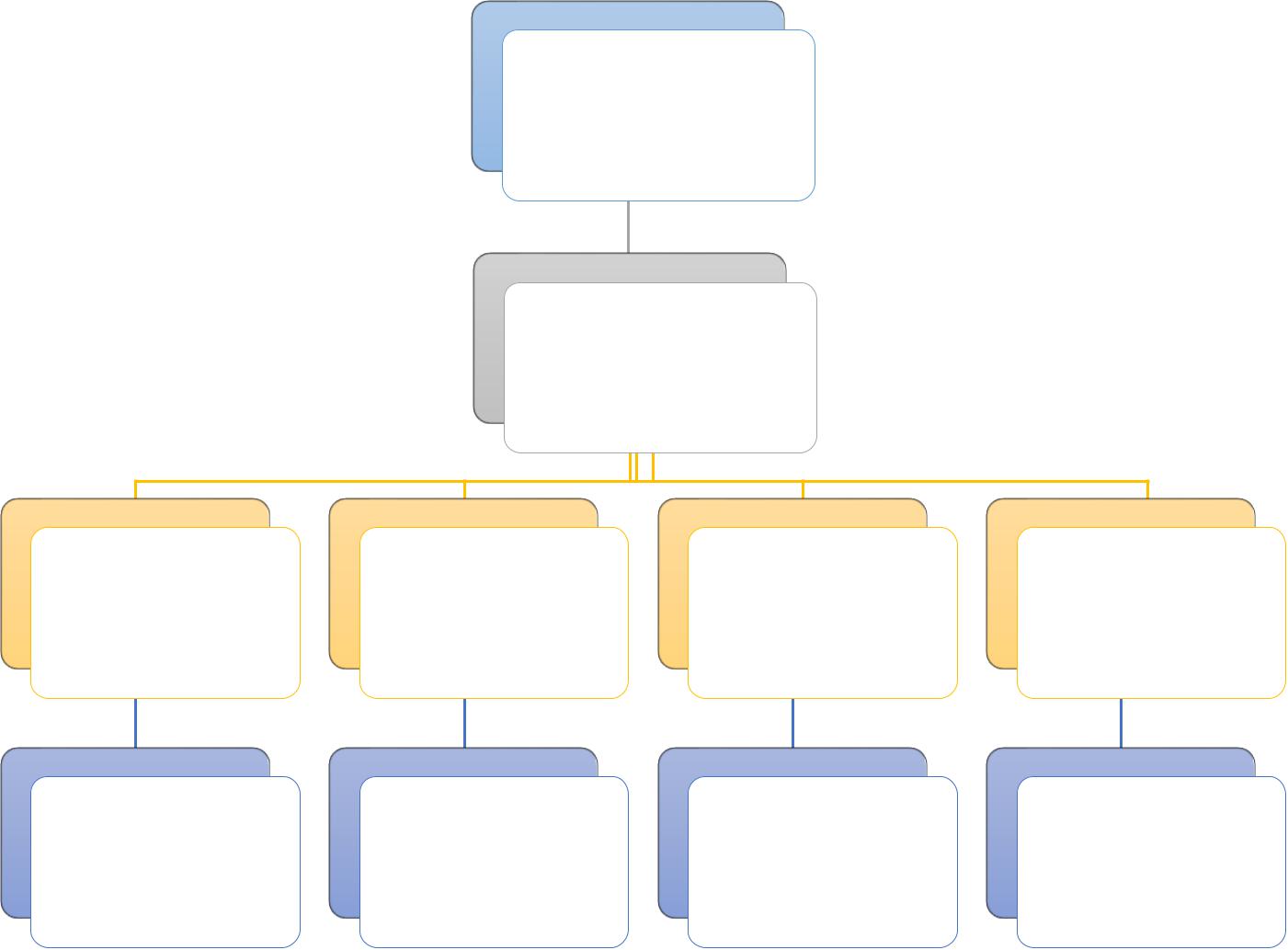
1. ***Roles and Responsibilities***
   1. *Organization and Structure*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Roles** |  |  | **Responsibility** |  |  |
|  |  |  |  |  |  |  |
|  | Project Manager |  |  | Planning developing schedules, allocating resources, keeping | |  |
|  |  |  |  | on schedule, coordinating communication, generally | |  |
|  |  |  |  | responsible for keeping the team’s focus on main goal and tries | |  |
|  |  |  |  | to keep the project team focused on the right goal at a time. | |  |
|  |  |  |  |  | |  |
|  | Technical Leader |  |  | Responsible for the underlying architecture for the hardware |  |  |
|  | (Hardware and Software) |  |  | system and software program, assigning tasks, mentoring |  |  |
|  |  |  | people, reporting. Technical leaders is a reference book for |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | other team members. |  |  |
|  |  |  |  |  |  |  |
|  | Quality Assurance Manager |  |  | Ensuring the products meet the certain standards of quality | |  |
|  |  |  |  | from requirements. | |  |
|  |  |  |  |  | |  |
|  | Test Leader |  |  | Responsible for test execution, including test set-up and test |  |  |
|  |  |  |  | run, evaluation of test run and error recovery, defect logging |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | and test results recording. |  |  |
|  |  |  |  |  |  |  |
|  | Developer |  |  | Involve to code product | |  |
|  |  |  |  |  | |  |
|  | Designer |  |  | Involve to design product |  |  |
|  |  |  |  |  |  |  |
|  | Tester |  |  | Involve to test product | |  |
|  |  |  |  |  |  |  |

*Table 2-1: Project Structure*

|  |  |
| --- | --- |
| *2.2.2.2. Project Team Member* | |
|  |  |
|  |  |
| **Team Member** | **Roles** |
|  |  |
| AnhLB | Project Manager, Technical Leader, Developer, Tester |
|  |  |
| HuongLX | Designer, Technical Leader, Developer, Test |
|  |  |
| PhongDC | Technical Leader, Developer, Tester Leader |
|  |  |
| HoangPM | Designer, Developer, Tester |
|  |  |
| LuatPD | Designer, Developer, Tester |
|  |  |
| TuDN | Designer, Developer, Tester |
|  |  |

*Table 2-2: Project Team Member*

SonHX (Supervisor)

AnhLB

(Project Manager)

Technical Designer Developer Testing and QA

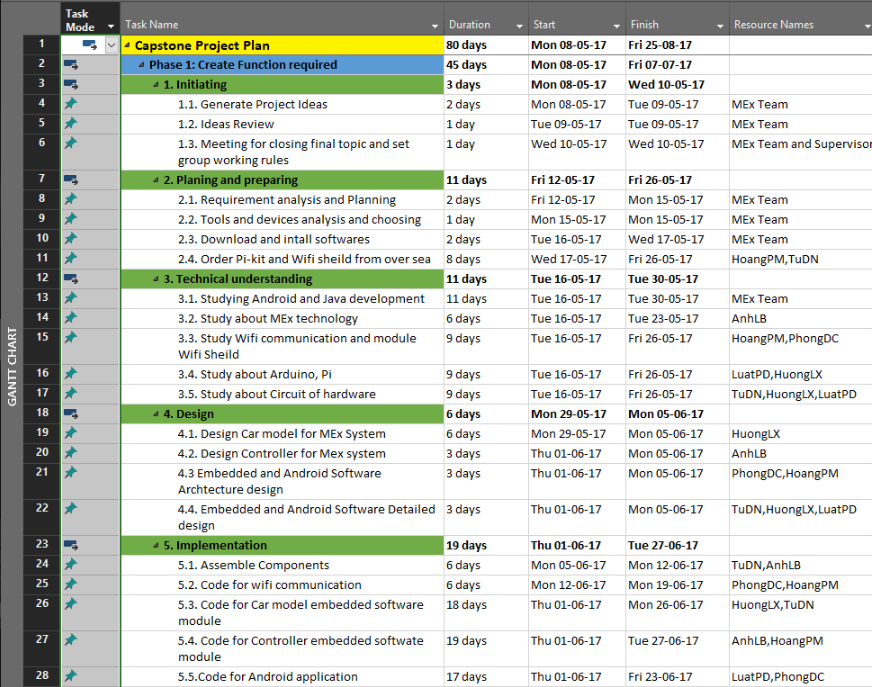
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| - AnhLB | - HoangPM | All of the | All of the |  |
| - HuongLX | - LuatPD |  |
| member | member |  |
| - PhongDC | - TuDN |  |
|  |  |  |

*Figure 2-2: Project Team Member*

***2.2.3. Tools and Techniques***

* Programing language : Android, Python 3.5, Arduino
* Process Model: Iterative and Incremental Software Process Model.
* IDEs: Android Studio, Arduino IDE, IDLE
* Design tool: Fritzing, Photoshop, Proteus
* Other:
  + Tortoise GIT
  + Microsoft Word 2016
  + Microsoft Excel 2016
  + Microsoft PowerPoint 2016
  + Microsoft Project 2016

1. **Project management plan**





*Figure 2-3: Project Management Plan*

Refer to [MEx\_ProjectPlan.mpp]

***2.3.2. Human Resource***

Human resource

* + - Team member
    - Supervisor

Non – human resource

* + Equipment: Personal Computers, Arduino, Raspberry PI
  + Building: FPT University, Thachhoa, Thachthat, Hanoi
  + Building: FPT University’s Library, Thachhoa, Thachthat, Hanoi

***2.3.3. Meeting Minutes***

All meeting minutes will be written follow this template:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Meeting/Project*** | Mex | |  |  |  |
|  |  |  |  |  |  |
| ***Date of Meeting:*** | *15/5/2017* |  | ***Time: (Type)*** | *2hours* |  |
|  |  |  |  |  |  |
| ***Meeting Called By:*** | *AnhLB* | | ***Location:*** | *FPT University* |  |
|  |  |  |  |  |  |
| ***Note Taker:*** | *TuDN* | | ***Time Keeper:*** | *PhongDC* |  |
|  |  |  |  |  |  |
| 1. Meeting Objective: |  |  |  |  |  |
|  |  |  |  |  |  |
| Brainstorming all functions of systems | | |  |  |  |
|  |  |  |  |  |  |
| 2. Attendance |  |  |  |  |  |
|  |  |  |  |  |  |
| Name | Roles |  | E-mail | Phone |  |
|  |  |  |  |  |  |
|  | Project |  | [@fpt.edu.vn](mailto:Duclqse02946@fpt.edu.vn) |  |  |
|  |  |  |  |  |  |
|  | Tester |  | [@fpt.edu.v](mailto:Tungntse02945@fpt.edu.vn)n |  |  |
|  |  |  |  |  |  |
|  | Developer |  | [@fpt.edu.vn](mailto:Anhpvse02918@fpt.edu.vn) |  |  |
|  |  |  |  |  |  |
| 3. Content: |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | *Table 2-3: Meeting Minutes Template* | | |  |  |

***2.3.4. Risk Management Plan***

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Name** | **Probability** | |  | **Prevention** |  |  | **Correction** |  | **Impact** |  |
|  |  |  | |  |  |  |  |  |  |  |  |
| 1 | **Miscommunication** | Medium | |  | After a meeting, one |  |  | When it becomes |  | High |  |
|  |  |  | |  | group member |  |  | clear that |  |  |  |
|  |  |  | |  | creates an interview |  |  | miscommunication |  |  |  |
|  |  |  | |  | report. Every |  |  | is causing problem, |  |  |  |
|  |  |  | |  | participant or |  |  | the team members |  |  |  |
|  |  |  | |  | absence person |  |  | are gathered in a |  |  |  |
|  |  |  | |  | should get a copy of |  |  | meeting to clear |  |  |  |
|  |  |  | |  | this report. Team |  |  | thing up. |  |  |  |
|  |  |  | |  | members should not |  |  |  |  |  |  |
|  |  |  | |  | hesitate to ask |  |  |  |  |  |  |
|  |  |  | |  | questions if they are |  |  |  |  |  |  |
|  |  |  | |  | unclear. |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |
| 2 | **Design Error** | High | |  | The design should be |  |  | When error in the |  | Medium |  |
|  |  |  | |  | reviewed very |  |  | design are noticed |  |  |  |
|  |  |  | |  |  |  |  |  |  |
|  |  |  | |  | critically. Team leader |  |  | PM or team leader |  |  |  |
|  |  |  | |  | should be consulted |  |  | should be consulted |  |  |  |
|  |  |  | |  | frequency on his |  |  | to help correct the |  |  |  |
|  |  |  | |  | opinion about the |  |  | design errors as soon |  |  |  |
|  |  |  | |  | feasibility and the |  |  | as possible. Also all |  |  |  |
|  |  |  | |  | correctness of certain |  |  | the work, that |  |  |  |
|  |  |  | |  | design decisions. |  |  | depends on the |  |  |  |
|  |  |  | |  |  |  |  | faulty design, should |  |  |  |
|  |  |  | |  |  |  |  | be halted until the |  |  |  |
|  |  |  | |  |  |  |  | error is corrected. |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |
| 3 | **Hardware Failure** | Low | |  | Check all of hardware |  |  | Creating a list of |  | High |  |
|  |  |  | |  | before buying. Being |  |  | store that is selling |  |  |  |
|  |  |  | |  | sure and testing about |  |  | this hardware. |  |  |  |
|  |  |  | |  | current and volt of this |  |  | Checking it exist if |  |  |  |
|  |  |  | |  | hardware before |  |  | having plan goes to |  |  |  |
|  |  |  | |  | using. |  |  | buy. |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |
| 4 | **Illness or absence** | Medium |  |  | Team members should |  |  | By ensuring that |  | Medium |  |
|  | **of team member** |  |  |  | warn their team leader |  |  | knowledge is shared |  |  |  |
|  |  |  |  |  | timely before a |  |  | between team |  |  |  |
|  |  |  |  |  | planned period of |  |  | members, work can |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | absence. |  |  | be taken over |  |  |  |
|  |  |  |  |  |  |  |  | quickly by someone |  |  |  |
|  |  |  |  |  |  |  |  | else if a person gets |  |  |  |
|  |  |  |  |  |  |  |  | ill. |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | **Requirement** | Medium |  |  | Carefully brainstorm |  |  | Team meetings with |  | High |  |
|  | **change** |  |  |  | system’s features |  |  | supervisor to |  |  |  |
|  |  |  |  |  | among team members. |  |  | determine whether |  |  |  |
|  |  |  |  |  | Regularly hold |  |  | new feature should |  |  |  |
|  |  |  |  |  | meeting to define and |  |  | be implemented or |  |  |  |
|  |  |  |  |  | discuss all the features |  |  | not. Team leaders |  |  |  |
|  |  |  |  |  | of systems. Design |  |  | create |  |  |  |
|  |  |  |  |  | system carefully. |  |  | implementation plan |  |  |  |
|  |  |  |  |  | Analyze all the |  |  | for implemented |  |  |  |
|  |  |  |  |  | possible cases to |  |  | features and sent to |  |  |  |
|  |  |  |  |  | minimize the change |  |  | team members. |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | **Time shortage** | High |  |  | Project manager |  |  | Lacking time is the |  | High |  |
|  |  |  |  |  | should create more |  |  | fatal problem, can |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | spare time and |  |  | run project to failure. |  |  |  |
|  |  |  |  |  | calculate plus 20% |  |  | PM should analysis |  |  |  |
|  |  |  |  |  | buffer time. |  |  | and has change on |  |  |  |
|  |  |  |  |  |  |  |  | the next phase. |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

*Table 2-4: Risk Management Plan*

***2.3.5. Communication Plan***

* 1. *Communication between team members*
* *Weekly meeting schedule*: By using Iterative and Incremental Process Model, MExProject System will be divided into a series of small tasks, each task will be assigned to team members by Technical Leader and depend on difficulty, and Technical Leader will assigned deadlines for each task. We will have a meeting every Thursday, Friday and Monday to report

the progress of whole team’s tasks. Any member who doesn’t finish his/her task (without reasonable explanation), will be fined. If there is any issue, we will discuss and find solution together. If it is too difficult and can’t be solved by ourselves, we will ask our supervisor for advises.

* Unscheduled meeting: If someone has an important problem want to be solved immediately, we will have a meeting for discussion.
* Communication channel: Our main communication channels are face-to-face meeting, email, Facebook, Skype. However, we sometimes can make a phone call or instant message if someone has problem.
  + 1. *Communication with Supervisor*
  + *Face-to-face* meeting: Weekly on every Thursday afternoons to make sure thatsupervisor can keep tracking of the team’s progress.
  + *E-*mail: Gmail is the fastest way to get device and document checking fromsupervisor.
  + *Mobile phone:* is used to get time and place arranged for the meeting every weekly.

1. **Projection Directory**

|  |  |  |  |
| --- | --- | --- | --- |
| Main folder | Sub-folder | Purpose |  |
|  |  |  |  |
|  | Meeting minutes | Store project meeting minutes |  |
|  |  |  |  |
|  | Report 1 | Store final deliverables of report 1 |  |
|  |  |  |  |
|  | Report 2 | Store final deliverables of report 2 |  |
| Project’s |  |  |  |
| Report 3 | Store final deliverables of report 3 |  |
| Document |  |  |  |
| Report 4 | Store final deliverables of report 4 |  |
|  |  |
|  |  |  |  |
|  | Report 5 | Store final deliverables of report 5 |  |
|  |  |  |  |
|  | Report 6 | Store final deliverables of report 6 |  |
|  |  |  |  |
|  | Final Report | Store final deliverables of final report |  |
|  |  |  |  |
| Plan |  | Store project plan, Task list |  |
|  |  |  |  |
| Resource |  | Store template needed in project |  |
|  |  |  |
| Tool | Store tool needed in project |  |
|  |  |
|  |  |  |  |
| Working space | Each team members has a folder | Team member’s working area |  |
|  |  |  |  |
| Reference |  | Store reference needed in project |  |
|  |  |  |  |
|  | *Table 2-5: Projection Directory* | |  |

# Requirements

**3.1. User Requirements Specification**

## 3.1.1 User requirement

* User use smartphone ( or remote controller) connect to car’s raspberry pi through WIFI
* User can use smartphone or remote controller to control the movement of the car and the camera
* User can use smartphone to switch control between smartphone or remote controller
* In remote controller mode, user can use smartphone to see the camera view

## Car requirement

* Right direction under the control of user
* Operation of the car must be stable and safe
* Easy to receive signal from user

## Android application requirement

* Well design with minimum cost and maximum quality
* Has long durability in period of time
* Be simple and user-friendly

## Remote Controller requirement

* Auto connect to system
* Low-cost hardware module

**3.2. System Requirement Specification**

## 3.2.1 Interface requirement

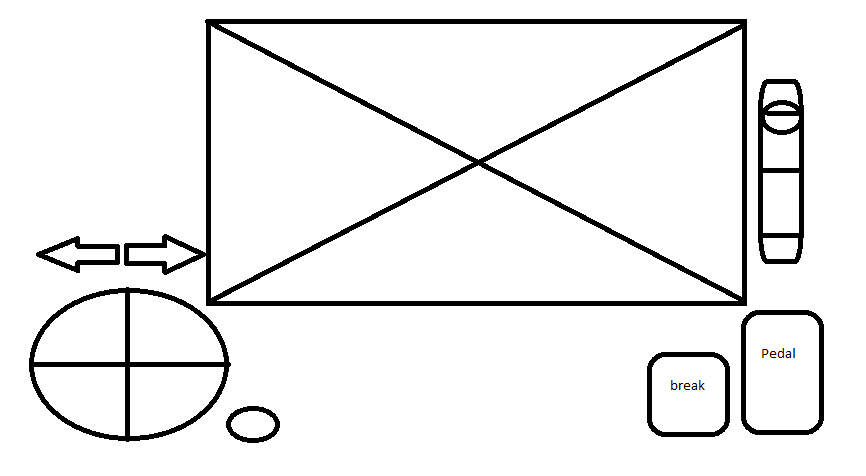
### 3.2.1.1 User Interface

User interface (UI) of mobile application must be design base on Flat UI Design. In this project, we only develops mobile application on Android. We ensure that the navigation options in the application will be similar; and all error occurring and exception handling will be catch and display for user with friendly messages.

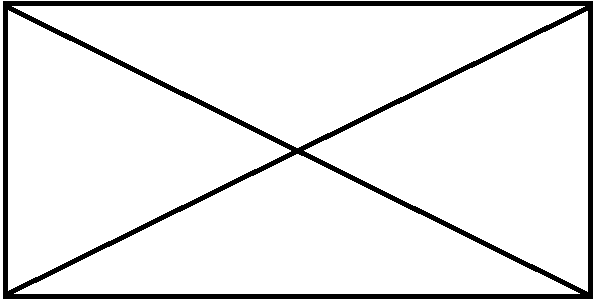
## 3.2.1.2 Hardware interface

* **Smartphone device** : the application only runs on smart phone which support Android 4.4+ ; the component helps user to control the car and also display the “view” from the car’s camera
* **Arduino WEMOS** : Use to control the car in remote controller, the signal will be transfer to Raspberry PI through WIFI
* **Raspberry PI** : Receive control signals from remote controller(or Smartphone device) and control the operation motors/Servo; also transfer video data from camera to smartphone device

## 3.2.1.3 Software interface



**Figure 3-1: Android controller screen**

****

**Figure 3-2: VR mode screen**

This table below will show detail function of all component of **Android application**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Component | Function | Illustrate Image | Description |
| 1 | Steering Wheel | Control movement of the car |  | User can turn left/right to  Control the car turn left/right |
| 2 | Pedal button | Increase the car speed |  | User can press the button to increase car speed |
| 3 | Break button | Decrease the car speed |  | User can press the button to decrease the car speed |
| 4 | Gear switcher | Switch run mode |  | User can pull the switcher down to change the car mode.  Default : parking |
| 5 | Light signal | Light signal |  | User can press an arrow to control car light signal.  Press 1 time to turn on, then press again/or the other button to turn off (or change the light) |
| 6 | Streaming screen | Stream all image info get from camera module in the car |  | Users watch this to know all info that robot’s camera is streaming |
| 7 | VR mode button | Enter the VR mode |  | Press the button to enter the VR mode |

**3.3. System Features**

3.3.1 General use Case Diagram

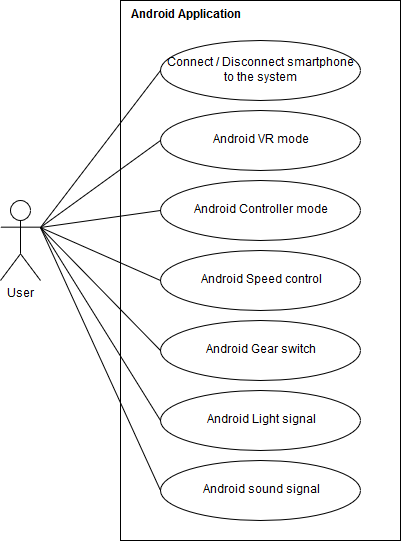
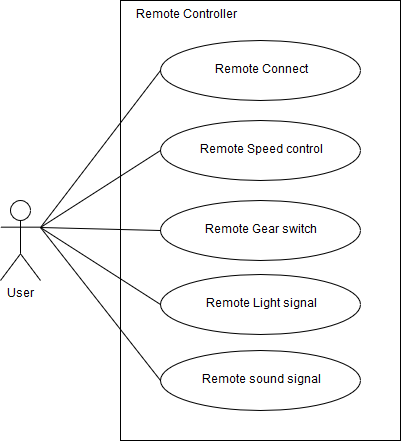


Figure 3-1 : Use case( Android) diagram



**Figure** 3-2: Use case (Remote controller) diagram

## 3.3.2 Functional requirement

### 3.3.2.1 Connect smartphone

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 1 SPECIFICATION | | | |
| Use-case No. | UC001 | **Use-case version** | 1.0 |
| Use-case Name | Connect smartphone | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to connect to system though WIFI  Goal : Successfully connected to the car system  Triggers :   * Connect to the car WIFI * Start Android Application * Tab “Connect device” button   Pre-condition :   * Android application is start successful * Car WIFI is turn on   Post-condition :   * Disable the Remote controller.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Connect to car’s WIFI** | | 2 | **User** | **Tab “Connect device” button** | | 3 | **Car System** | **Check the connection :**   * **Return “OK” if success** * **Return “NO” if fail to connect** * **Return** “**LIMIT” if any device has connected.** | | 4 | **Car System** | **Transfer signal to android device** | | 5 | **Android** | **Display connection message** |   Alternative Scenario: None  Exceptions : N/A | | | |
|  | | | |

### 3.3.2.2 Disconnect Smartphone

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 2 SPECIFICATION | | | |
| Use-case No. | UC002 | **Use-case version** | 1.0 |
| Use-case Name | Disconnect smartphone | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to disconnect to the car system  Goal : Successfully disconnected to the car system  Triggers :   * Tab “disconnect device” button   Pre-condition :   * Android application is connect successful with the car system   Post-condition :   * After disconnect user can connect again.   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab “Disconnect device” button** | | 2 | **System** | **Display disconnects successful message** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.3. Enter the VR mode

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 3 SPECIFICATION | | | |
| Use-case No. | UC003 | **Use-case version** | 1.0 |
| Use-case Name | Android VR mode | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Allow User to enter VR mode to view the camera in the VR mode and control the car by Remote controller  Goal : Successful enter the VR mode  Triggers :   * Tab on “VR mode” button   Pre-condition :   * The Android application is connect successful with the car system * Remote controller is turned on   Post-condition :   * Allow remote controller to control the car * User can turn back to “Android Controller mode”   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab on “VR mode” button** | | 2 | **Android System** | **Send signal to Car system** | | 3 | **Android System** | **Display the “VR mode" screen** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.4. Enter Android controller mode

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 4 SPECIFICATION | | | |
| Use-case No. | UC004 | **Use-case version** | 1.0 |
| Use-case Name | Android controller mode | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Allow User to enter android controller mode to control the car by Android application  Goal : Successful enter the android controller mode  Triggers :   * Tab on “back” button on the Android device   Pre-condition :   * Successful enter the “VR mode”   Post-condition :   * The remote controller is disable. * User can turn back to the “VR mode”   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab on “back” button** | | 2 | **Android System** | **Send signal to the car system** | | 3 | **Car System** | **Disable the “Remote controller”** | | 4 | **Android System** | **Display the “Controller mode” screen** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.5 Android Speed control

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 5 SPECIFICATION | | | |
| Use-case No. | UC005 | **Use-case version** | 1.0 |
| Use-case Name | Android speed control | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Use Android application to control the car movement  Goal : Control movement of the car  Triggers :   * Turn the steering wheel to left or right that user want the car to turn in that direction. * Tab the “Pedal” button to increase the movement speed of the car * Tab the “break” button to decrease the movement speed of the car   Pre-condition :   * Android application is connect successful with the car system * Enter the “Android controller mode” * Gear switch isn’t in “parking mode”   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Turn the steering wheel to left or right and also tab the “Pedal” button to increase the movement speed or tab the “break” button to decrease the movement** | | 2 | **System** | **Send a signal to car’s system to control the movement of the car** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.6 Android Gear Switch

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 6 SPECIFICATION | | | |
| Use-case No. | UC006 | **Use-case version** | 1.0 |
| Use-case Name | Android Gear swich | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to control car run mode (Run forward / backward)  Goal : Control run mode of the car  Triggers :   * Tab “Gear Switcher” button to switch between “parking”, “run forward” and “run backward” mode   Pre-condition :   * Android application is connect successful with the car system * Enter “Android controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Drag the “Switcher” lever.** | | 2 | **Android System** | **Send a signal to car system to switch run mode**   * **“0” : parking** * **“1” : run forward** * **“2” : run backward** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.7 Android light signal

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 7 SPECIFICATION | | | |
| Use-case No. | UC007 | **Use-case version** | 1.0 |
| Use-case Name | Android light signal | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to turn on/off the light signal  Goal : successful to turn on/off the light signal  Triggers :   * Tab the “light signal left” or “light signal right” arrow button   Pre-condition :   * Android application is connect successful with the car system * Enter “Android controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab the “light signal left” or “light signal right” arrow button** | | 2 | **Android System** | **Transfer signal to car system**   * **“1” : light signal left** * **“2” : light signal right** | | 3 | **Car system** | **Receive the signal then turn on the light.** | | 4 | **User** | **Tab the arrow button again to turn off or tab the other arrow button to change the light signal.** | | 5 | **Android system** | **Transfer signal to car system**   * **“0” : turn off** * **“1” : light signal left** * **“2” : light signal right** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.8 Android Sound signal

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 8 SPECIFICATION | | | |
| Use-case No. | UC008 | **Use-case version** | 1.0 |
| Use-case Name | Android Sound signal | | |
| Author | PhongDC | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Allow use to turn on the sound signal  Goal : Successful to turn on the sound signal  Triggers :   * Hold the “Sound” button   Pre-condition :   * Android application is connect successful with the car system * Enter “Android controller” mode   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Hold the “sound” button to turn on the sound then release the button to turn off.** | | 2 | **Android System** | **Transfer signal to the car system**   * **“1” : turn on** * **“0” : turn off** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.9 Remote Connect

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 9 SPECIFICATION | | | |
| Use-case No. | UC009 | **Use-case version** | 1.0 |
| Use-case Name | Remote connect | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : When user turn on the “Remote controller”, it’ll auto connect to the car system  Goal : Successful connect to the car system  Triggers :   * Turn on the “Remote controller”   Pre-condition :   * N/A   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Turn on the “Remote controller”** | | 2 | **Remote System** | **Transfer signal to the car system** | | 3 | **Car system** | **Receive signal. Check if any device has connected.**   * **Return “OK” : Successful to connect** * **Return “NO” : Fail to connect** * **Return “LIMIT” : if any device has connected** | | 4 | **Car system** | **If connect successful. System check the controller mode.**   * **Return “True” if Android device has not connected or user use the “VR mode” in the android app.** * **Return “Fail” if user use the “Android controller mode”.** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.10 Remote Speed control

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 10 SPECIFICATION | | | |
| Use-case No. | UC010 | **Use-case version** | 1.0 |
| Use-case Name | Remote Speed control | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Use Remote controller to control the car movement  Goal : Control movement of the car  Triggers :   * Turn the steering wheel to left or right that user want the car to turn in that direction. * Tab the “Pedal” button to increase the movement speed of the car * Tab the “break” button to decrease the movement speed of the car   Pre-condition :   * Remote Controller is connect successful with the car system * Remote controller has allow to control. * Remote Gear switch lever isn’t in “parking mode”   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Turn the steering wheel to left(or right) and also tab on the “Pedal” button to increase the movement speed or tab the “break” button to decrease the movement** | | 2 | **Remote System** | **Send a signal to car’s system to control the movement of the car** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.11 Remote Gear Switch

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 11 SPECIFICATION | | | |
| Use-case No. | UC011 | **Use-case version** | 1.0 |
| Use-case Name | Remote Gear switch | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use smartphone to control car run mode (Run forward / backward)  Goal : Control run mode of the car  Triggers :   * Pull “Gear Switcher” Lever to switch between “parking”, “run forward” and “run backward” mode   Pre-condition :   * Remote controller is connect successful with the car system * Remote controller has allow to control   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Pull the “Switcher” lever.** | | 2 | **Remote System** | **Send a signal to car system to switch run mode**   * **“0” : parking** * **“1” : run forward** * **“2” : run backward** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.12 Remote Light Signal

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 12 SPECIFICATION | | | |
| Use-case No. | UC012 | **Use-case version** | 1.0 |
| Use-case Name | Remote Light Signal | | |
| Author | HoangPM | | |
| Date |  | **Priority** | High |
| Actor : User  Description : User use Remote controller to turn on/off the light signal  Goal : successful to turn on/off the light signal  Triggers :   * Tab the “light signal left” or “light signal right” button   Pre-condition :   * Remote controller is connect successful with the car system * Remote controller has allow to control   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Tab the “light signal left” or “light signal right” arrow button** | | 2 | **Remote System** | **Transfer signal to car system**   * **“1” : light signal left** * **“2” : light signal right** | | 3 | **Car system** | **Receive the signal then turn on the light.** | | 4 | **User** | **Tab the button again to turn off or tab the other button to change the light signal.** | | 5 | **Remote system** | **Transfer signal to car system**   * **“0” : turn off** * **“1” : light signal left** * **“2” : light signal right** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

### 3.3.2.13 Remote Sound Signal

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE 13 SPECIFICATION | | | |
| Use-case No. | UC013 | **Use-case version** | 1.0 |
| Use-case Name | Remote Sound Signal | | |
| Author | TuDN | | |
| Date |  | **Priority** | High |
| Actor : User  Description : Allow use to turn on the sound signal  Goal : Successful to turn on the sound signal  Triggers :   * Hold the “Sound” button   Pre-condition :   * Remote controller is connect successful with the car system * Remote controller has allow to control   Post-condition :   * N/A   Main Success Scenario:   |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | **User** | **Hold the “sound” button to turn on the sound then release the button to turn off.** | | 2 | **Remote System** | **Transfer signal to the car system**   * **“1” : turn on** * **“0” : turn off** |   Alternative Scenario: None  Exceptions : N/A | | | |
| Note and issues : | | | |

## 3.2.3 Non-Functional Requirement

### 3.2.3.1 Reliability

* Mobile application and Remote Controller must work correctly with car system, have no conflict between devices.
* Only 1 android device and 1 Arduino can connect to the car system at the same time
* Car’s angle, speed, run mode must be the same as the display on the mobile

### 3.2.3.2 Availability

* The mobile application and Remote controller are easy to connect to system to transform data.
* Power of components like Remote controller, Car System, Camera and smart phone is stable and available

### 3.2.3.3 Maintainability

* Microcontroller and all components can be replaced easily
* Implementation code must be follow coding standard, clearly commented for maintaining and enhancing system in the future

### 3.2.3.4 Performance

* System has to response one command in less than 100ms

**3.3. Infrastructure and Tools**

## 3.3.1 Hardware

|  |  |
| --- | --- |
| No | Item |
| 1 | Raspberry Pi 3 |
| 2 | Arduino WEMOS D1 R2 |
| 3 | Arduino UNO R3 |
| 4 | Speed-reducer Motor |
| 5 | Servo MG996R |
| 6 | Rotary resistance |
| 7 | PS1/PS2 motor controller |
| 8 | Webcam Logitech C170 |
| 9 | Pin Cell 3.7 |
| 10 | Frame mica |
| 11 | Module L9110s |
| 12 | Servo SG90 |

## 3.3.2 Software and tool

* Android Studio
* Python IDLE 3.5
* Arduino IDE
* Proteus 8
* Fritzing

## 3.3.3 Hardware study

### 3.3.3.1. Raspberry PI 3



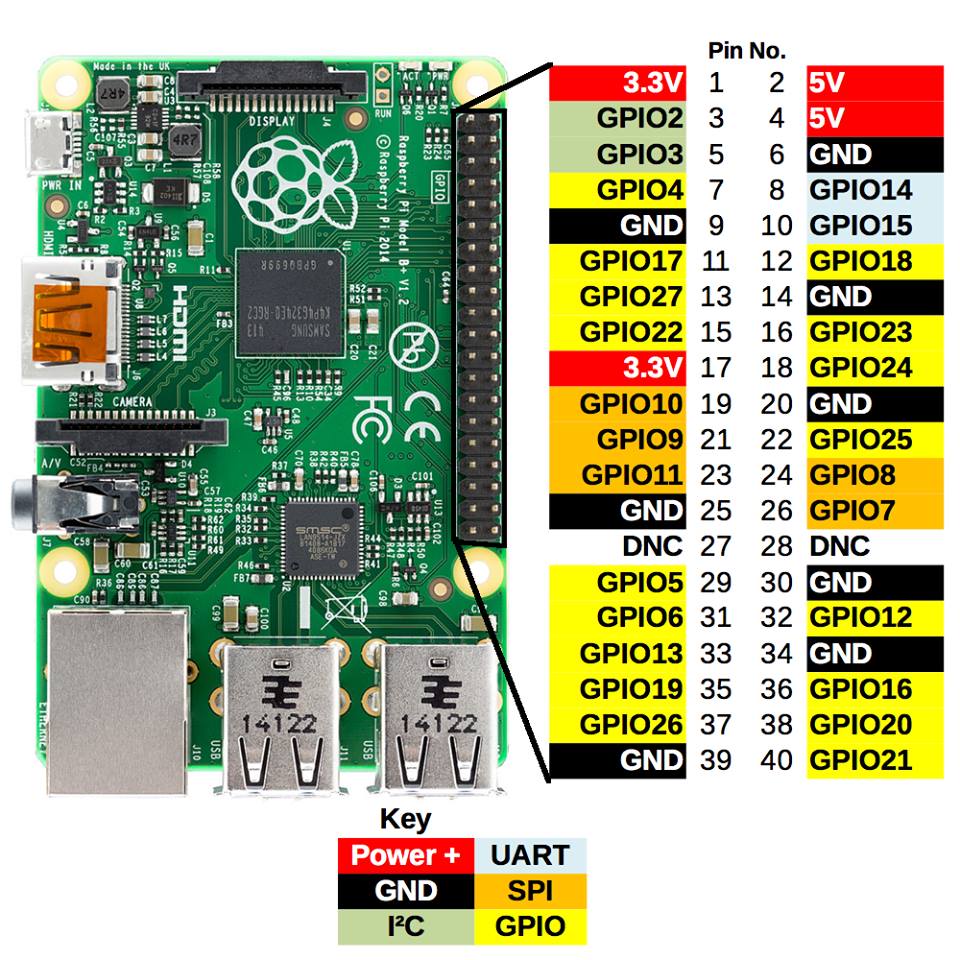
**Figure 3-3: Raspberry PI**

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings us a more powerful processer, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.

Connecting it to a computer with a USB cable or battery to get started simply.

|  |  |
| --- | --- |
| **Processor** | Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache |
| **GPU** | Broadcom VideoCore IV @ 250 MHz  OpenGL ES 2.0  MPEG-2 and VC-1 |
| **Memory** | 1GB |
| **Operating System** | Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT |
| **Network** | 10/100 Mbit/s Ethernet, 802.11n wireless, Bluetooth 4.1 |
| **Video Output** | HDMI |
| **Audio Output** | Audio Output 3.5mm jack |
| **Camera Connector** | 15-pin MIPI Camera Serial Interface |
| **Display Connector** | Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane |
| **Memory Card Slot** | Push/Pull micro SDIO |
| **GPIO Connector** | 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines |

**Table 3-1: Raspberry PI 3 Specifications**

****

**Figure 3-4: Pin Diagrams**

### 3.3.3.2. Arduino Uno

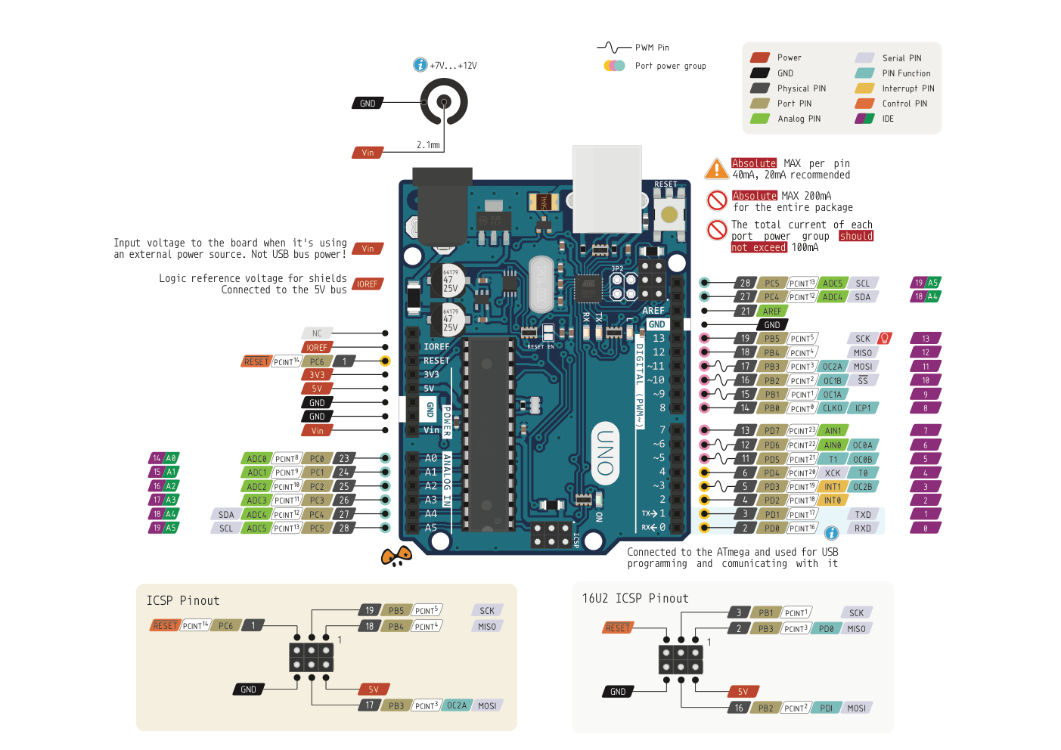


**Figure 3-5: Arduino Uno**

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

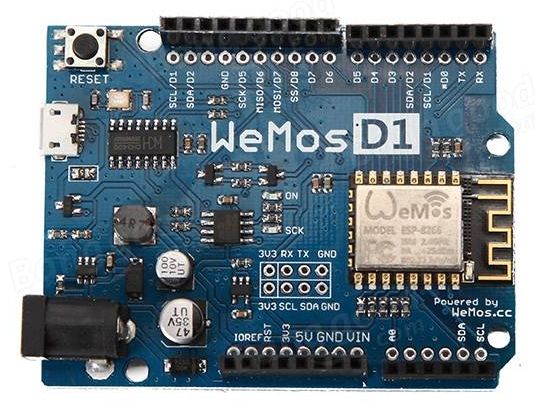
|  |  |
| --- | --- |
| **Microcontroller** | ATmega328 |
| **Architecture** | AVR |
| **Operating Voltage** | 5V |
| **Flash memory** | 32KB of which 0.5 KB used by bootloader |
| **SRAM** | 2KB |
| **Clock Speed** | 16 MHZ |
| **Analog I/O Pins** | 6 |
| **EEPROM** | 1 KB |
| **DC Current per I/O pins** | 40 mA on I/O pins; 50 mA on 3.3V pin |
| **Input Voltage** | 7 – 12 V |
| **Digital I/O pins** | 20 |
| **PWM Output** | 6 |
| **PCB Size** | 53.4 x 68.6 mm |
| **Weight** | 25g |

**Table 3-2: Arduino Uno specifications**

****

**Figure 3-6: Arduino Uno pin diagram**

### 3.3.3.3. Arduino Wemos-D1R2



**Figure 3-7: Wemos-D1R2**

Wemos-D1R2 is an ESP8266-12 based WIFI enabled microprocessor unit on a Arduino-UNO footprint

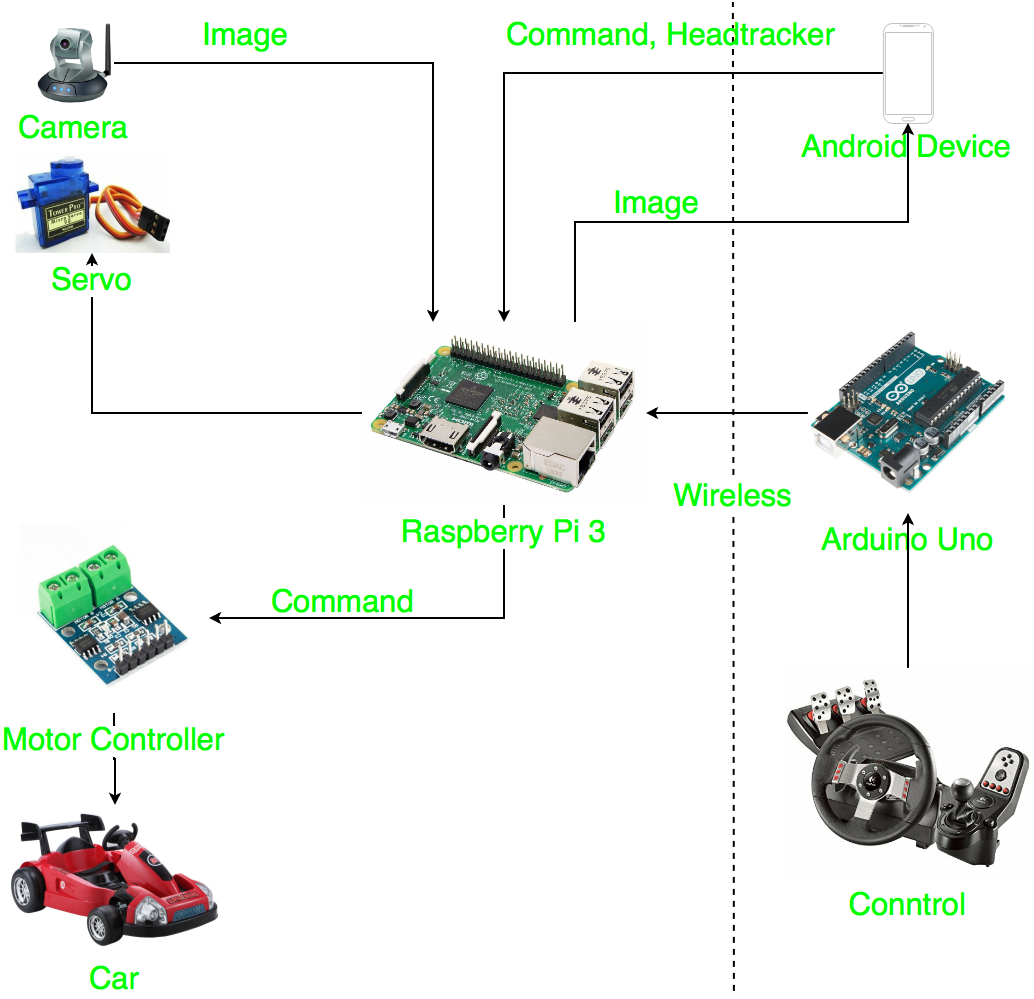
|  |  |
| --- | --- |
| **Microcontroller** | ESP8266EX |
| **Operating Voltage** | 3.3V |
| **Input Voltage Range** | 9V to 12V |
| **Output** | 5V at 1A max |
| **Digital I/O pins** | 11 |
| **Analog input pins** | 1 |
| **Flash Memory** | 4MB |
| **Board Dimensions** | 68.6mm x 53.4mm |
| **Weight** | 21.8g |

**Table 3-3: Wemos-D1R2 specifications**

### 3.3.3.4. Motor

# IMPLEMENTATION

## 4.1. Proposed system architecture

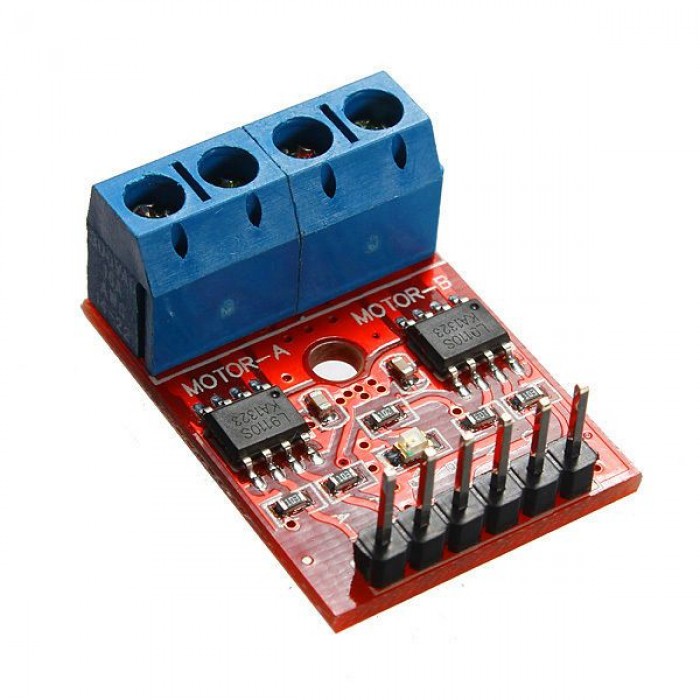


**Figure 4-1: System architecture**

## 4.2. Analysis and selection of tools, devices

### 4.2.1. Motor controller

We used 2 motor combined with L9110s module to control car movement. The module will control 2 motor with customize speed.



**Figure 4-2: Module L9110s Figure 4-3: Motor**

When the car go forward or backward, two motor will turned the same direction and speed.

**Figure 4-4: Run forward**

### 4.2.2. Servo car

We used servo (MG996R) to control the car. When the car turn left (or right), the servo will turn left (or right).We also set the swivel angle of servo between 45 to 135



**Figure 4-5: Servo MG996R**

### 4.2.3. Servo camera

We used servo (SG90) to control the camera (Webcam Logitech C170). The servo combine with the plastic shelf to hold the camera. When driver turn Android device to left (or right), servo will turn in the same direction.

### 

****

**Figure 4-6: camera turn**

### 4.2.4. Remote Controller

We use Arduino Uno to process data combine with WEMOS to transfer data to Raspberry server through WIFI; we also use mechanical parts of Panther-Lord (Car racing game control) to control the car in Remote Controller mode. The Remote include 3 parts:

* Steering : wheel , light and sound signal button
* Transmission : Gear lever, hand-break
* Pedals : throttle pedal, break pedal



**Figure 4-7: Remote controller**

### 4.2.5. Supply power for Raspberry PI

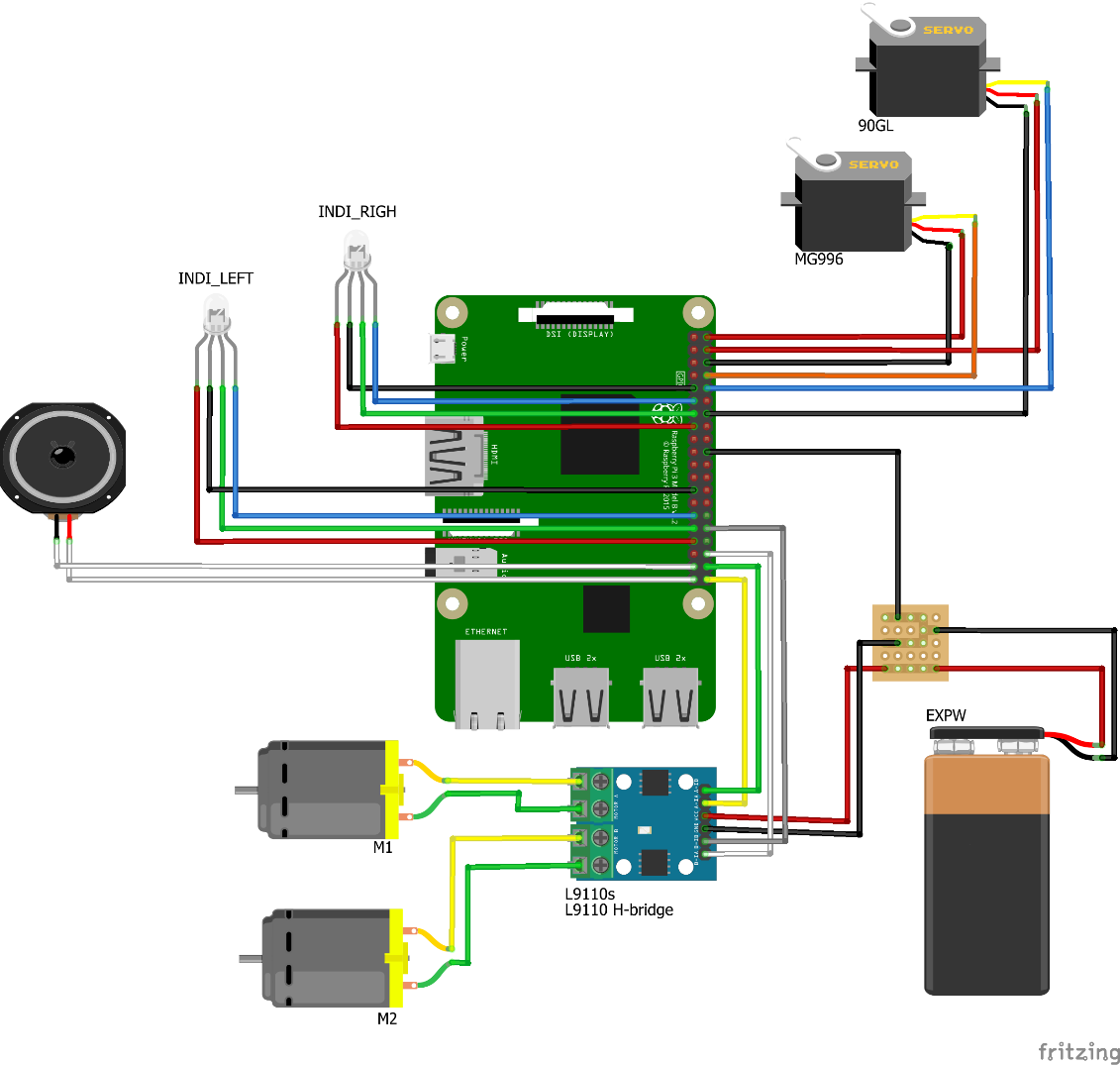
We use LGDB118650 pin cell, a rechargeable battery that can be use and recharge many time, also with 3000mAh and 3.7 voltage, suitable for Raspberry PI and powerful enough for the car system.

**Figure 4-8: LGDB118650**

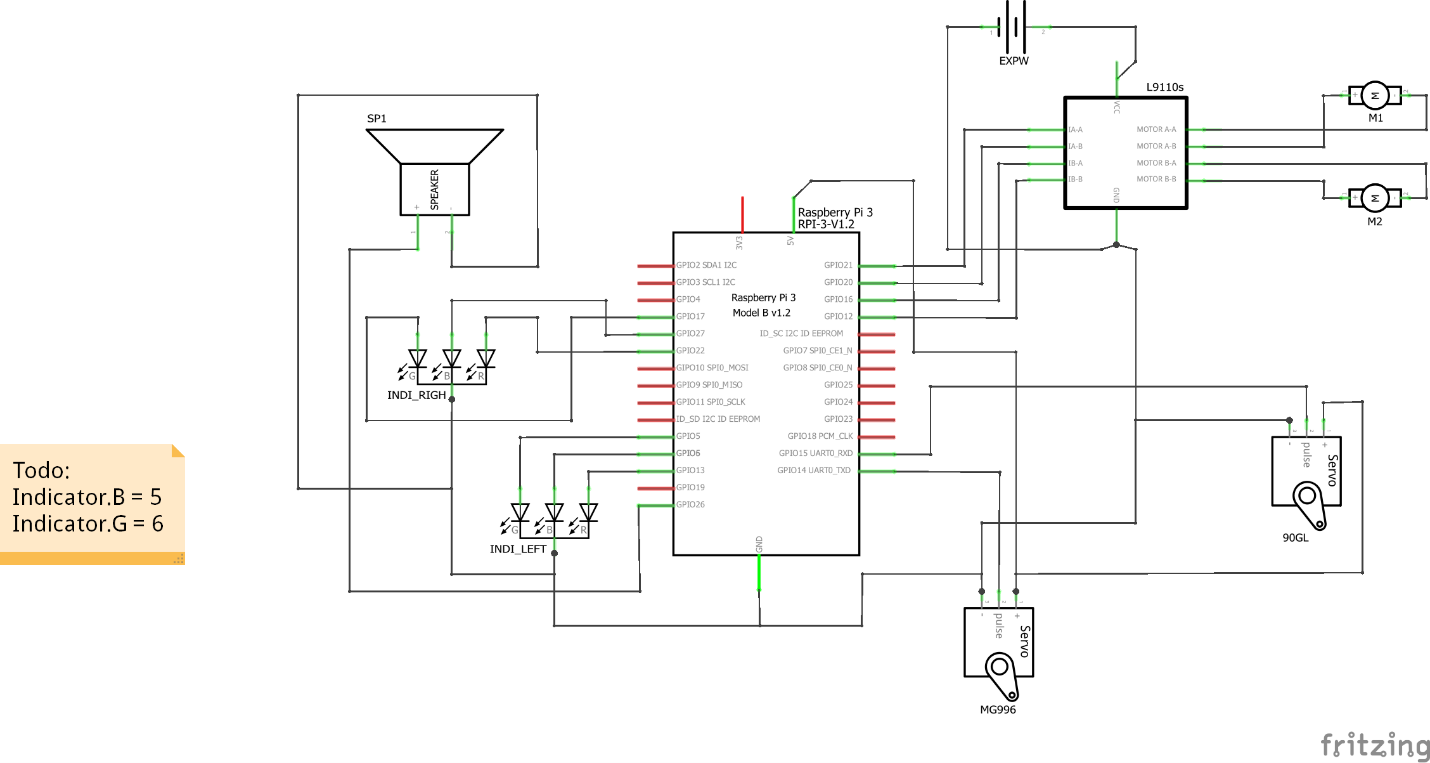
**Figure 4-9: Module**

## 4.3. Circuit design

### 4.3.1. Raspberry PI connector

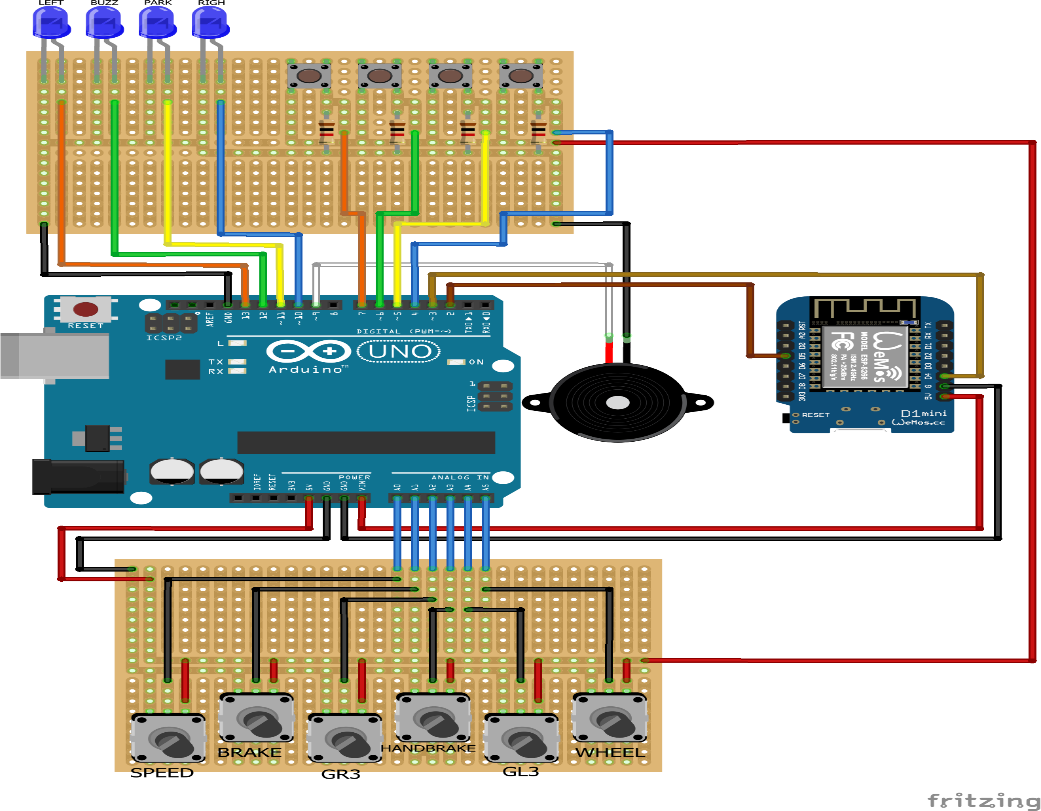


**Figure 4-10: Raspberry PI connecter circuit design**

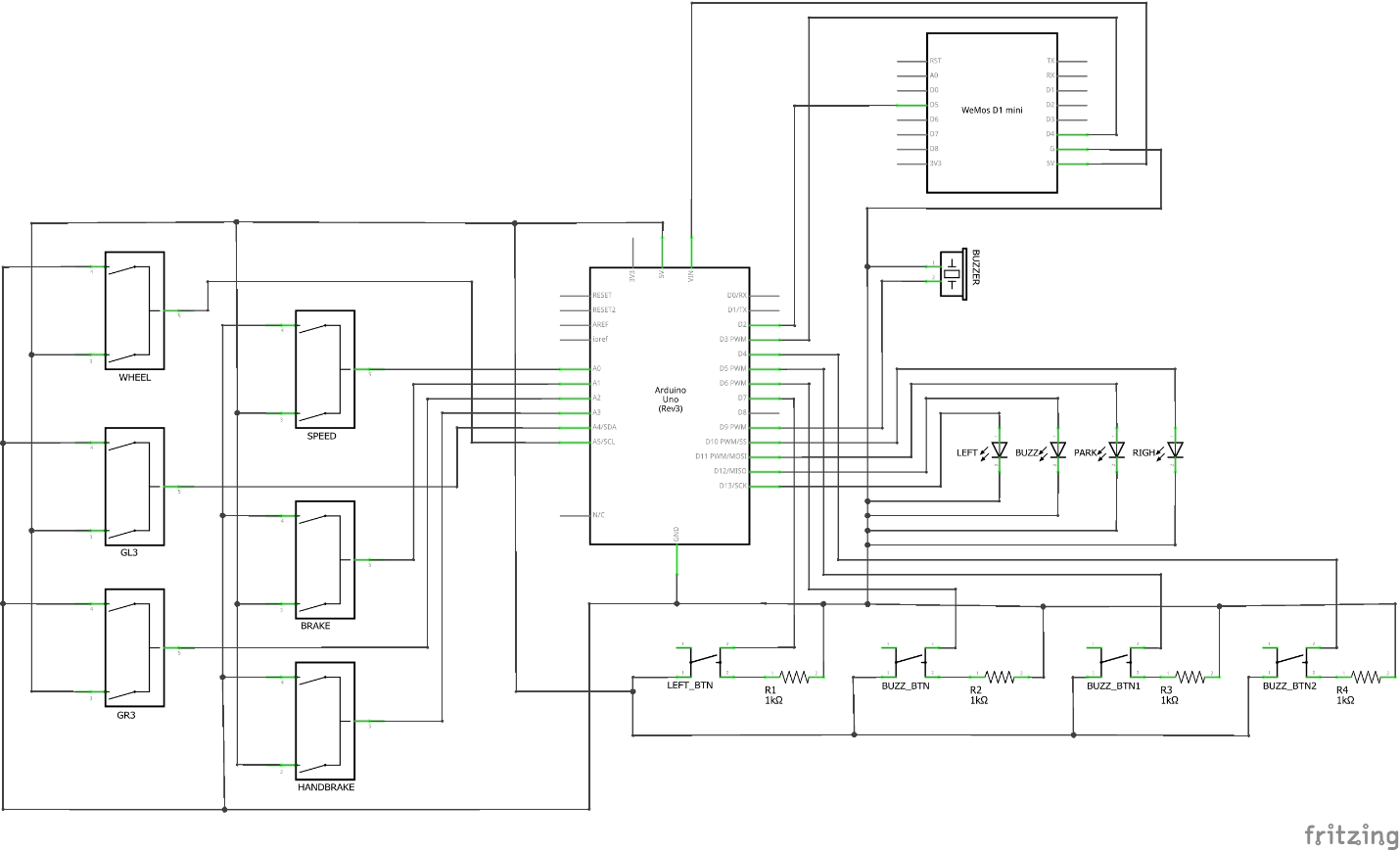
****

**Figure 4-11: Raspberry PI connector schematic**

### 4.3.2. Remote controller



**Figure 4-12: Remote Controller circuit design**



**Figure 4-13: Remote controller schematic**

## 4.4. Interface design

## 4.5. Testing

### 4.5.1. Purpose

The primary purpose of this report is to detect software failures so that detects may

be discovered and correct to ensure that our project is thoroughly tested and resulting in a successful implementation project.

### 4.5.2. Test approach

* Unit testing
* Integration testing
* System testing
* Acceptance testing

### 4.5.3. Test environment:

|  |  |
| --- | --- |
| Resource | Name and type |
| Personal Laptop |  |
| Android smartphone |  |
| WIFI |  |
| Window 10 | Operating system |
| Android | Android 4.4, 5.1, 7.0 |
| Android Development tool | Android Studio |
| Arduino IDE | Arduino 1.8.1 |

### 4.5.4. Test Case

#### Software

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | ID | Test case Description | Test Case Procedure | Expected Output | Inter-Test Case Dependence | Status | Date | Note |
| 1 | Android\_1 | Open [Mini Explorer] main screen | Touch on [Mini Explorer] application icon | Open [Mini Explorer] main screen : screen is displayed with the following informations  -Text : Project name  - 1 connect button |  | Pass | 14/08/2017 |  |
| 2 | Android\_2 | Showing [WIFI is turned off] dialog after press on [connect] button (If WIFI is turned off) | Touch on [Connect] button | Show alert dialog which contain a text: “WIFI is turned off” | [Mini Explorer] screen is displayed; WIFI of android device is off or not connect with Car WIFI | Pass | 14/08/2017 |  |
| 3 | Android\_3 | Open [Android controller] screen | Touch on [Connect] button | Open [Android controller] screen: screen is displayed with the following informations  -1 [Disconnect] button  -5 control buttons  -2 buttons change speed  -1 [VR mode] button  -1 webview | [Mini Explorer] screen is displayed; WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 4 | Android\_4 | Send control car data | Touch on [Gear switcher]/[pedal]/[break]/[light signal]/[sound signal]/[steering wheel] button | Send signal to car system and show message on Raspberry PI in order :  “Speed”:0, ”angle”:0, ”mode”:0, ”light”:0, ”buzzer”:0 | [Android controller] screen is displayed;  WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 5 | Android\_5 | Open [VR mode] screen | Touch on [VR mode] button | Open [VR mode] screen: screen is displayed with the following informations  -1 Webview that display Camera image info | [Android controller] screen is displayed;  WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 6 | Android\_6 | Send control camera module data | Turn android device to left or right | Send signal to car system and show message on Raspberry PI | [VR mode] screen is displayed;  WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |

#### Hardware

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | ID | Test Case Description | Test Case Procedure | Expected Output | Inter-Test Case Dependence | Status | Date | Note |
| 1 | Car\_1 | Change run mode testing | Pull the [Gear switcher] | Raspberry receive correct message and change the car run mode | -[Android controller] screen is displayed; WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 2 | Car\_2 | Car turn left testing | Turn the [steering wheel] to left | Raspberry receive correct message and the car turn left | -[Android controller] screen is displayed; WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 3 | Car\_3 | Car turn right testing | Turn the [steering wheel] to left | Raspberry receive correct message and the car turn right | -[Android controller] screen is displayed; WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 4 | Car\_4 | Car increase speed testing | Push the [Pedal] | Raspberry receive correct message and the car increase speed | -[Android controller] screen is displayed; WIFI of android device is on and connected with the Car WIFI | Pass | 14/10/2017 |  |
| 5 | Car\_5 | Car decrease speed testing | Push the [Break] | Raspberry receive correct message and the car decrease speed | -[Android controller] screen is displayed;  WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 6 | Car\_6 | Sound signal testing | Press on the [Sound signal] | Raspberry receive correct message and turn on the sound signal | -[Android controller] screen is displayed;  WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 7 | Car\_7 | Turn on light signal testing | Press on the [Light signal left]/[Light signal right] | Raspberry receive correct message and turn on the light signal left/right | -[Android controller] screen is displayed; WIFI of android device is on and connected with the Car WIFI | Pass | 14/08/2017 |  |
| 8 | Car\_8 | Turn off light signal testing | Press on the [Light signal left]/[Light signal right] again | Raspberry receive correct message and turn off the light signal left/right | -[Android controller] screen is displayed;  WIFI of android device is on and connected with the Car WIFI  -[Light signal left]/[Light signal right] has turned on | Pass | 14/08/2017 |  |
| 9 | Car\_9 | Turn the [camera module] to left | Turn android device to left | Raspberry receive correct message and turn the [camera module] to left | -[Android controller] screen is displayed;  WIFI of android device is on and connected with the Car WIFI  - Camera module has turned on | Pass | 14/08/2017 |  |
| 10 | Car\_10 | Turn the [camera module] to right | Turn android device to right | Raspberry receive correct message and turn the [camera module] to right | -[Android controller] screen is displayed; WIFI of android device is on and connected with the Car WIFI  - Camera module has turned on | Pass | 14/08/2017 |  |

# Results

# System user’s Manual

## 6.1 Setting up hardware

Press the “power” button on car

**Figure 6-1: Press the “power” button**

Supply power for remote controller.

**Figure 6-2: supply power for Remote**

Remote Controller will be auto connect with the car system.

## 6.2 Android Application

The first, download and install the “Mini explorer” app on android device (Recommendation: Android 4.4+).

### 6.2.1 Connect WIFI

* Open WIFI setting on your smart phone
* Choose “RazzPi” WIFI
* Enter password (The WIFI password will be provided with the product. Example : “12345678” )
* Choose “Connect”

### Connect Car system

* Open “Mini explorer” app.
* Choose “Connect” button. After success, direct move to “Android Controller” screen.

**Figure 6-3: Open “Mini explorer” app**

**Figure 6-4: Choose “Connect” button on the main screen**

**Figure 6-6: Successful**

### Control car movement

* Pull the “gear switcher” down to change the run mode(Default mode : parking)
* Turn the “Steering wheel” to left (or right) that you want the car move in that direction.
* Push the “Pedal” button to increase the car speed
* Push the “Break” button to decrease the car speed

**Figure 6-7: Android Controller Screen**

**Figure 6-8: Run mode**

### VR mode

Push the “VR” button to enter the VR mode, after that put your android device into the VR (Virtual reality) headsets. Now, your android device will only display the camera view and you can control your car by using the “remote controller”. You can also turn the camera to left (or right) by turn your head to the direction that you want.

**Figure 6-9: VR mode**

**Figure 6-10: Remote controller**

# References

* <https://developer.android.com>
* <https://arduino.vn>
* <https://github.com>
* <https://draw.io>
* <http://linhkienhanoi.com>
* <https://python.org>