Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Summer 2021

Parking Lot Rover Team
Lot Striping Rover

Liliana Diaz
Sean-Michael Woerner
Abiria Placide
Mariane Sanchez
Bryan Bidjocka

Revision History

Revision	Date	Author(s)	Description
0.1	6.21.2021	LD	document creation
0.2	6.28.2021	SW, AP, LD, BB, MS	Version 1 Completion
0.2	8.02.2021	SW, AP, LD, BB, MS	Final Version Completion

CONTENTS

List of Figures

List of Tables

1 Introduction

The purpose of this project is to design and create a parking lot striping rover. This rover will be autonomous with GUI interfaces and Bluetooth connections to decrease costs and increase efficiency in parking lot striping. The rover will have more accuracy and precision in creating lines for parking lots. This rover will be small, compact and durable enough that it will require minimal human interaction. This rover will increase performance compared to the manually operated parking lot striping machines already in the market.

The rover will be made utilizing an electric wheelchair base and will use a RKT GPS for getting co-ordinates of it's current location. A Raspberry Pi will be used to send the commands used for knowing where to go, where to paint, and when to release the paint. These commands will be sent and received using Mission Planner Software.There will be three main components for creating the parking lot striping rover. The first component will be focused on the GPS connections, the second will be focused on the paint aspects, and the third will be the rover's mobility as a whole.

2 System Overview

Figure 1 displays the system architecture of the parking lot striping rover. There are four main layers that interact with the parking lot striping rover, the user interface layer, GPS RTK Board layer, the motor control layer, and the paint layer. The rover is created using a wheelchair base which is part of the central layer. The central layer's main purpose is to serve as the base between all the layers, utilizing a PX4 PixHawk board as the heart of the system. The base will receive the way-point data from the user interface layer and send it to the GPS layer to calculate it's location coordinates. The motor control layer contains the motors and their batteries, along with a Sabertooth motor controller. The Sabertooth controls the speed and power of the motors. Figure 2 displays the interactions between the user interface, GPS RTK board, and motor control layers with the central rover layer.

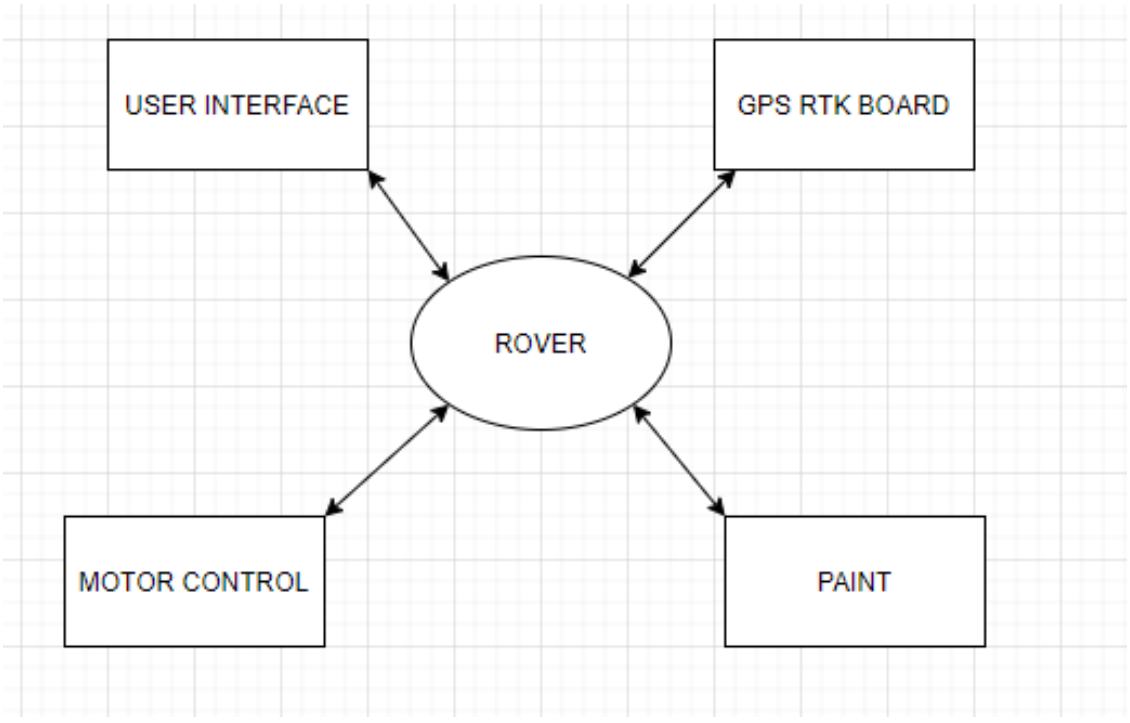

Figure 1: System architecture of the Rover

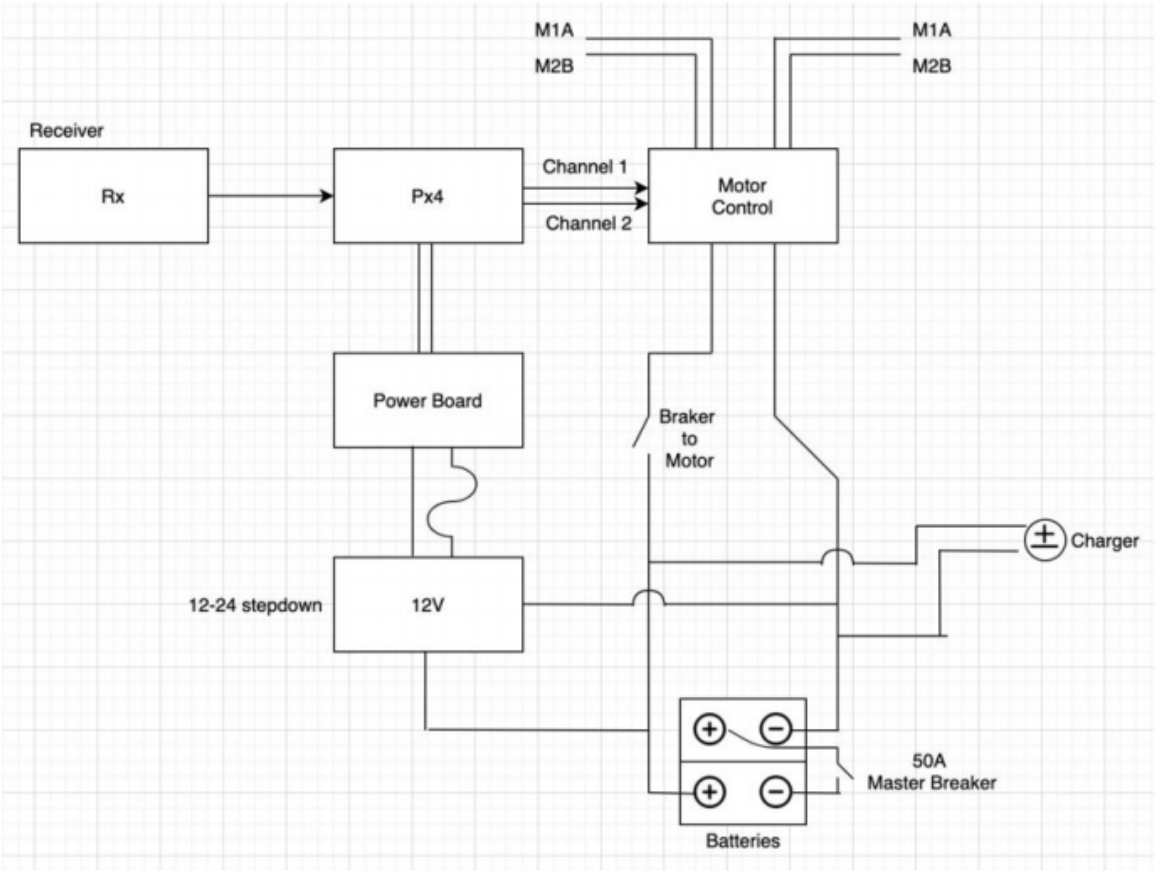

Figure 2: Motor Control, GPS RTK Board, and the user interface layers as a system.

3 Central Layer Subsystems

The central layer, the rover, is the main layer of the system. This layer controls the other layers.

3.1 Layer Hardware

The hardware components for this layer contain the wheelchair base used to operate the rover, which has already built. No further hardware has been used for the base.

3.2 Layer Operating System

Because the central layer has only one subsystem, there is no operating systems that are required for operating the wheelchair base for the rover.

3.3 Layer Software Dependencies

There are no software dependencies used for this part of the layer.

3.4 PX4 PixHawk

The PixHawk (PX4) is a lightweight advanced autopilot flight controller. The PX4 is the brain of the entire system and plays the converting role between the layers, letting the layers to communicate with one another. It receives data from the user interface system and sends it to the GPS RTK layer for navigation. Receiving the signals from the GPS, it guides the rover to move to a certain direction, and sends the moving commands to the motor control layer of the rover.

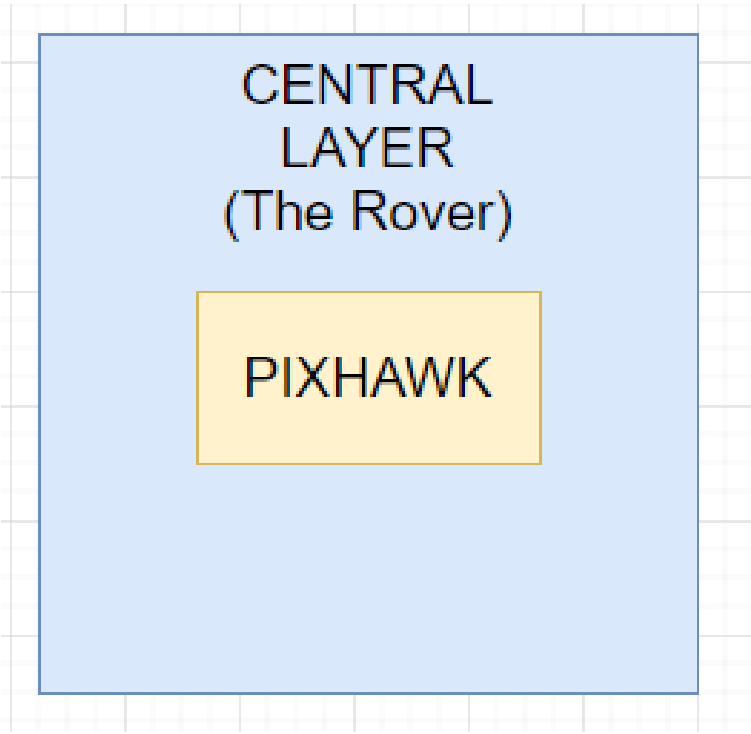

Figure 3: PixHawk Subsystem Diagram

3.4.1 Subsystem Hardware

The Pixhawk uses an ARM Cortex microcontroller base, a 32-bit ARM Cortex M4 core with FPU. It also has a 32-bit fail safe co-processor with a 168MHz RAM and 2MB flash memory. The PixHawk used for the rover requires two telemetry radios that let the paint striping rover connect to a computer via radio link. It also requires a radio control system so that it may be manually controlled from a handheld transmitter. This radio control system, the receiver, will allow to control the rover in case the autonomous function fails. The PixHawk also requires a buzzer, which is used with the PX4 autopilot to annunciate status tones. This can be used to know when the rover is turned on, off, or if any errors occur with the rover. The PixHawk requires a power management board, which serves the purpose of a power module as well as a power distribution board. This piece of hardware allows regulated power to the PixHawk and sends information to the autopilot about the battery's voltage and current supplied to the controller and motors. It can have a max current output of 120 amps with a 5 volts output.

3.4.2 Subsystem Operating System

The PixHawk requires a Windows Operating System.

3.4.3 Subsystem Software Dependencies

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem. The PixHawk depends on two software sources, the Mission Planner software and the ArduPilot software. ArduPilot is open source autopilot software that is capable of controlling autonomous rovers, which is needed for this project. ArduPilot consists of navigation software that can be used on a running rover, along with a ground station controlling software, which is Mission Planner. Mission Planner is compatible with Windows only, and is used as a configuration utility or as a dynamic control supplement for the rover. This software allows to setup, configure, and

tune the rover for optimum performance.

3.4.4 Subsystem Programming Languages

The main programming language used for the PixHawk is written in C++, but Python can be used as well. The main flight code for the ArduPilot software is written in C++, and the supporting tools are commonly written in python.

3.4.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

3.4.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

4 Motor Control Layer Subsystems

The motor control layer contains the subsystems that give the rover the ability to move using the PixHawk hardware.

4.1 Layer Hardware

The hardware used for the motor control layer are mainly two 12 volt batteries to supply sufficient power needed to allow the rover movement. They will supply power to the entire rover, and will be DC batteries. These batteries can be recharged with a given power charger cord that can charge both at once. Two motors are used for each of the back wheels of the rover and will be supplied power by the two batteries. These motors will give the rover the ability to accelerate forwards, backwards, or be able to turn in different directions based on the given instructions from the user. The movement commands are given from the PixHawk, the central layer.

4.2 Layer Operating System

This layer requires a Windows Operating System

4.3 Layer Software Dependencies

There are no software dependencies for the motors or the batteries, but are needed for a subsystem.

4.4 Motor Controller - Sabertooth

The Sabertooth 2x32 is a dual channel motor driver capable of supplying 32 amps to two motors, with peak currents up to 64 amps per motor. It can be operated from radio control, along, TTL serial or USB inputs. The Sabertooth communicates with the central layer via the PixHawk, where the pixHawk will then take inputs on/off signals and forward them back to the Sabertooth that will control the rover motors. The Sabertooth converts the design of high level programming to low level programming so taht the motor can understand and execute the user command.

4.4.1 Subsystem Hardware

The Sabertooth motor controller requires the rover's two motors and the two batteries to make the rover move. The motors receive the commands from the Sabertooth, which can cause the wheels of the rover to rotate or make movement.

4.4.2 Subsystem Operating System

The Sabertooth requires a Windows Operating System.

4.4.3 Subsystem Software Dependencies

The Sabertooth requires the DEScribe software to function. DEScribe is Dimension Engineering's PC software that adjusts, modifies, monitors, and/or updates the motor drivers, as well as other similar devices.

4.4.4 Subsystem Programming Languages

Sabertooth is compatible with most programming languages, but for this project, it is focused on C++ and Python.

4.4.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

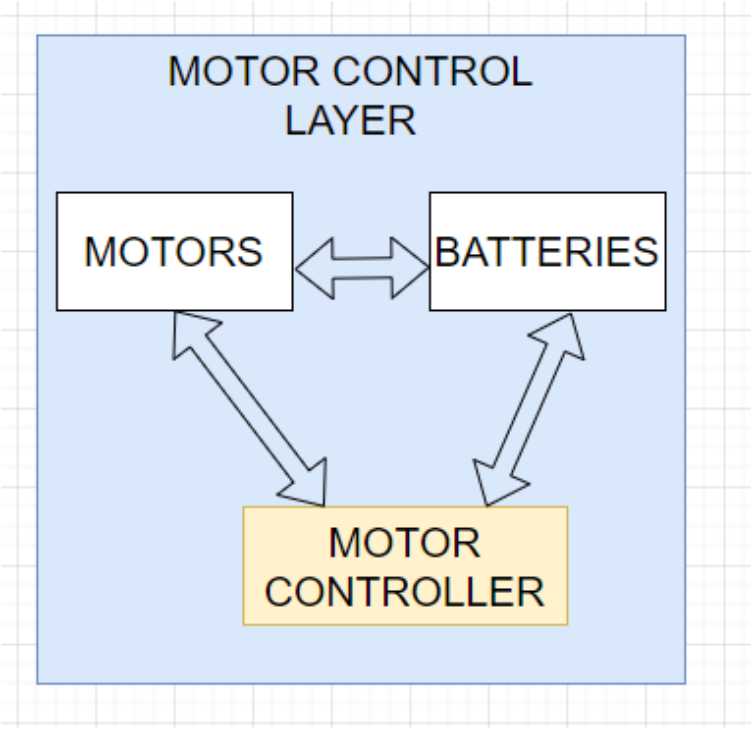

Figure 4: The Motor Controller-Sabertooth diagram

4.4.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

5 GPS RTK Board Layer Subsystems

The GPS RTK board layer interacts with the PixHawk and Sabertooth to know of the rover's location, in order to know when and where to move to.

5.1 Layer Hardware

The GPS RTK board contains two pieces of hardware, an antenna and a Zed F9P RTK board that will correct a GPS signal received to the rover. This will cause the rover to be more accurate in movement and for way-point tracking of the rover system. The antenna is necessary to receive a strong signal and with the creation of another RTK GNSS base station, the signal can be received from different locations.

5.2 Layer Operating System

The operating system necessary for this layer is a Windows Operating System.

5.3 Layer Software Dependencies

Mission planner is necessary to interact with the rest of the rover layers and hardware.

5.4 Zed F9P RTK Board

Descibe at a high level the purpose and basic design of this subsystem. Is it a piece of hardware, a class, a web service, or something else? Note that each of the subsystem items below are meant to be specific to that subystem and not a repeat of anything discussed above for the overall layer. The Zed-F9P board is used to correct a GPS signal with a correction signal from an RTK base station, providing a 1cm accuracy for the way-point tracking of the rover system. This piece of hardware will take data from the GNSS and a correction signal from a base station and calculate the movements necessary to follow way points received from Mission Planner. The movements will be sent to the PixHawk controller which will be connected to the DC controller that controls the motors and wheels.

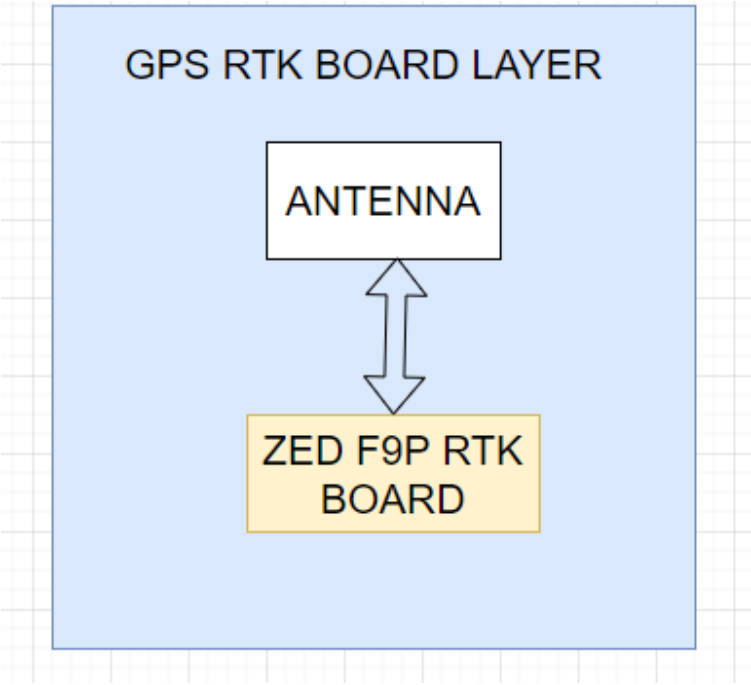

Figure 5: Zed F9P RTK Board description diagram

5.4.1 Subsystem Hardware

A description of any involved hardware components for the subsystem.

5.4.2 Subsystem Operating System

The system requires a Windows Operating System.

5.4.3 Subsystem Software Dependencies

The RTK board requires the ArduPilot software and Mission Planner software.

5.4.4 Subsystem Programming Languages

For this project, this subsystem requires the C++ and Python programming languages to be compatible with the other layer hardware and software.

5.4.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

5.4.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

6 User Interface Layer Subsystems

The user interface layer makes the use of the rover simple without needing a skilled technician to operate.

6.1 Layer Hardware

The purpose of the layer hardware is to give the user access to select points on a map via a GUI, graphic user interface, on a board or remote application and monitor the rover as it paints the stripes on the parking lot.

6.2 Layer Operating System

This layer requires a Windows Operating System.

6.3 Layer Software Dependencies

The user interface layer is dependant on Mission planner to interact with the rest of the rover layers.

6.4 Graphical User Interface- GUI

The main purpose of the GUI is to receive the inputs from the user to indicate the path the rover has to follow, and the major GUI output is the positions of the rover based on the RTK-GPS information.

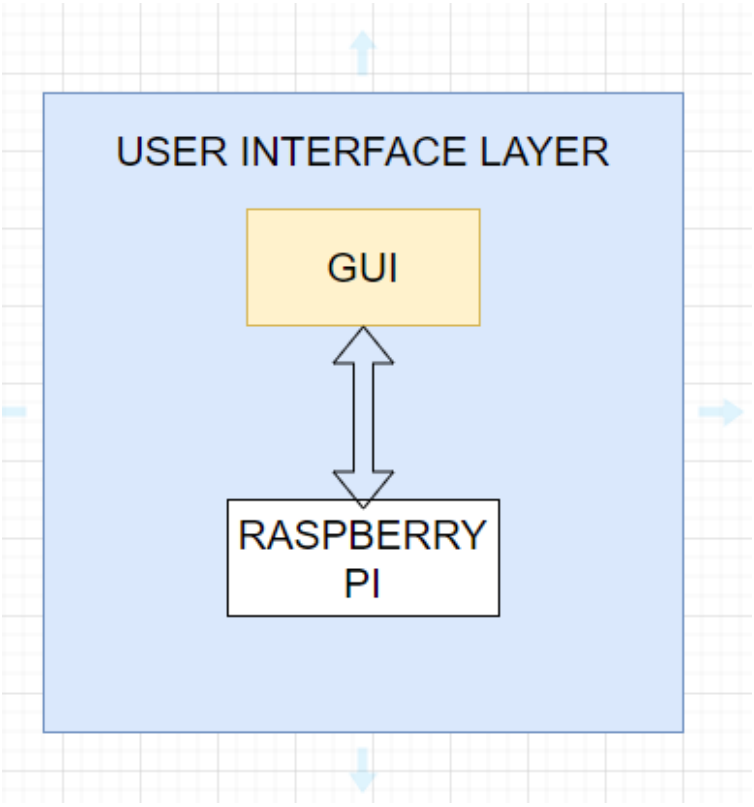

Figure 6: GUI subsystem diagram

6.4.1 Subsystem Hardware

Hardware required for the GUI to be accessible are human interface devices, such as a computer, keyboard, and a mouse, touchpad, or even joystick. The hardware allows the users to interact with the GUI.

6.4.2 Subsystem Operating System

For this project, the GUI requires a Windows Operating System.

6.4.3 Subsystem Software Dependencies

GUI is a system interactive visual component for computer software, this GUI for the project can work with Mission Planner.

6.4.4 Subsystem Programming Languages

There are no programming languages required for the GUI.

6.4.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

6.4.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

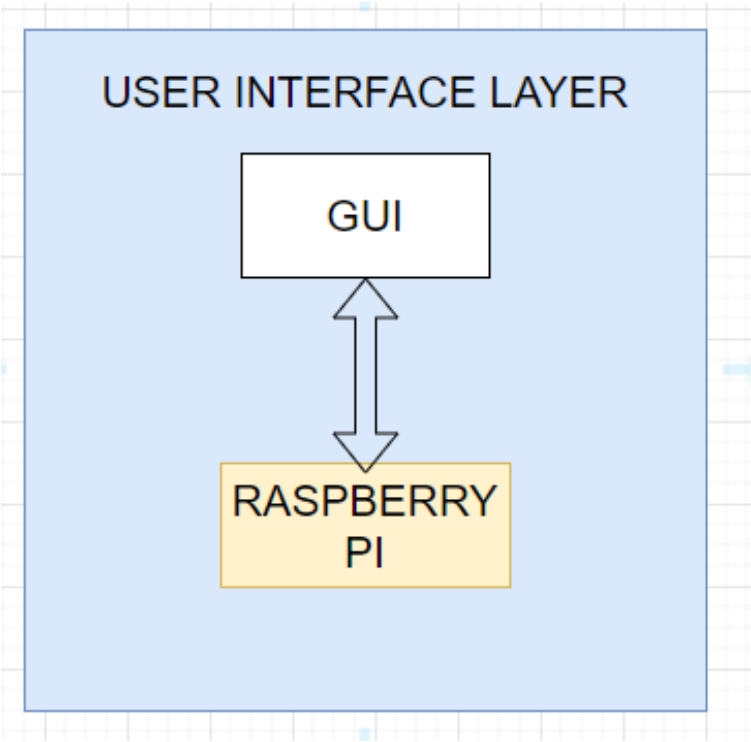

Figure 7: Raspberry Pi description diagram

6.5 Raspberry Pi

The Raspberry Pi is the acing operating system of the rover. It is the "brain" of the system in order to make the commands to lower level. The Raspberry Pi requests the PixHawk to connect to the GPS, then sends it to the motor controller. It also requests the PixHawk to connect to the sensor to detect any

obstacles and sends that data to the motor controller. The Raspberry Pi communicates with the GUI and thus connects tot he rover system as a whole.

6.5.1 Subsystem Hardware

The Raspberry Pi uses a 1.8 GHz 64-bit quad-core Arm Cortex-A72 CPU.

6.5.2 Subsystem Operating System

The Raspberry Pi is the operating system for the rover called Raspberry Pi OS.

6.5.3 Subsystem Software Dependencies

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem.

6.5.4 Subsystem Programming Languages

This subsystem is compatible with most programming languages, but C++ and python are primarily used.

6.5.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

6.5.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

7 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES

[1] Asphalt Kingdom. Asphalt Kingdom Line Stripers.
[2] National Pavement. Parking Lot Striping: Everything You Should Know, 2019.
[3] New Stripe. Parking Lot Striping Equipment.
[4] Texas Parking Lot Striping. Texas Parking Lot Striping
[5] Line Stripping Arlington. Parking lot striping arlington.