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

Detailed Design Specification
CSE 4317: Senior Design II
Fall 2024

IGVC-002
Autonomous Ground Vehicle

Dezean Gardener
Hayden Ansell
Nestor Arteaga
Jocsan Cano
Debin Babykutty
Chau Nguyen

CONTENTS

List of Figures

List of Tables

1 Introduction

For these next few months, we will be working as a team to build an unmanned ground vehicle. The work will be divided into 3 sections, The physical components of the robot, the vision system that will detect objects, and finally the path-finding system, that will calculate a safe path for the vehicle to travel.

The physical components refers to the motor control, camera system, lidar, power management, and the frame itself. Vision will be done using the camera and lidar working together. Finally, path-finding will be done in software, using the vision components and code written. At a minimum, the path finding system should be able to, within its physical constraints, avoid obstacles that are in it's immediate path. As a whole, the vehicle should be able to traverse a semi-random static environment, locate and identify potential obstacles, and then avoid them.

2 System Overview

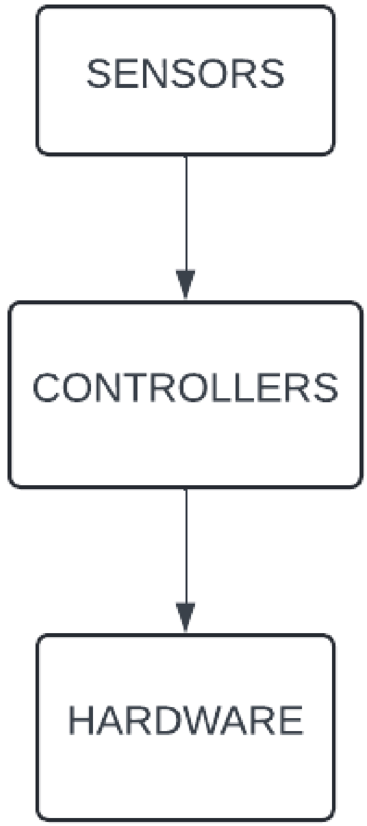

Figure 1: A simple architectural layer diagram

2.1 Sensors Layer Description

These sensors gather information from the environment and then send that information to the controller layer for processing. The camera gathers images of the environment to detect lanes. The LiDAR creates a map of the environment that we can crosscheck with the camera's data to understand where every object is in relation to the vehicle.

2.2 Controller Layer Description

This layer will use ROS2 to traverse the environment. Once an adequately safe path is found, the controller layer will send a signal to the hardware layer for the robot to move.

2.3 Hardware Layer Description

This layer will execute the demanded control signals from the software along with being responsible for all low-level operations such as powering, circuit current regulations, motors, safety standards.

3 Sensor Layer Subsystems

3.1 LiDAR Subsystem

This subsystem consists of an RP LiDAR S2. The LiDAR will provide live information about any protruding obstacles such as cones, barrels, and walls.

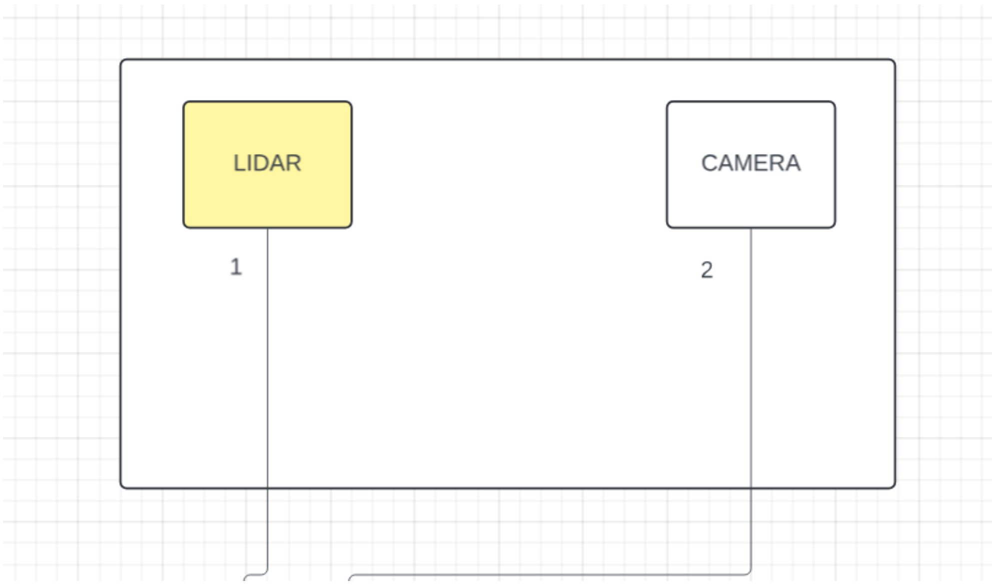

Figure 2: LiDAR description diagram

3.1.1 Subsystem Hardware

The rpLiDAR S2 will be the only hardware used in this layer

3.1.2 Subsystem Operating System

Ubuntu 22.04 is required to run ROS2 and subsequently to use the LiDAR in conjunction with the rest of the vehicle.

3.1.3 Subsystem Software Dependencies

RVIZ and or gazebo are needed to visualize the LiDAR data. These programs allow for the simulation and visualization of many sensors.

3.1.4 Subsystem Programming Languages

No programming languages are required to run the LiDAR

3.1.5 Subsystem Data Structures

The LiDAR sends information in a YAML format that includes a lot of information, the most important being the range values for each point mapped, and the angle of increment for every cycle the LiDAR runs.

3.2 Camera Subsystem

In an IGVC competition, the ground vehicle's camera subsystem consists of hardware and software components for obtaining, analyzing, and using visual data for perception and navigational tasks. It communicates with the control system of the car to allow for navigation and autonomous operation in changing conditions..

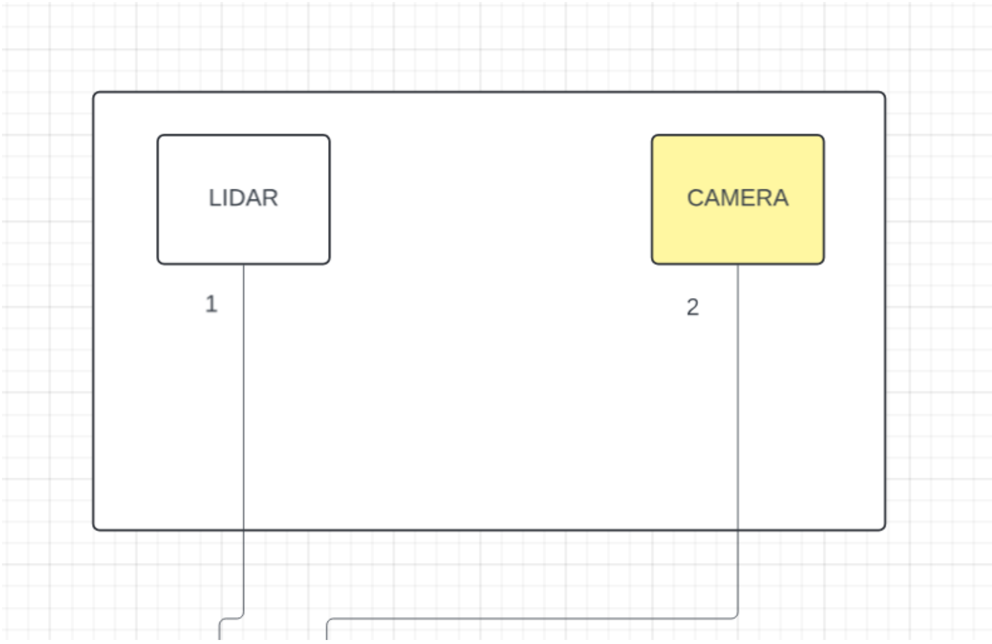

Figure 3: Camera description diagram

3.2.1 Subsystem Hardware

Power system, sensors, computer platform, communication systems, control interfaces, and mounting hardware are just a few of the many parts that make up an IGVC ground vehicle's system hardware.

3.2.2 Subsystem Operating System

The subsystems operating systems of the IGVC contain real-time operating systems and frameworks

3.2.3 Subsystem Data Structures

The camera will send data in an array format that will allow the model to categorize all the pixels into categories (obstacles, lanes, drivable area) for image processing and lane detection.

4 Controller Layer Subsystems

4.1 Layer Hardware

The hardware is controlled solely through interfacing over the WIFI through the NVIDIA Jetson Orin Nano sending signals to the ESP 32 which is used to drive the motors through the 2x32 Sabertooth.

4.2 Layer Operating System

Ubuntu 22.04 is needed to run ROS2

4.3 Layer Software Dependencies

ROS2 must be set up correctly for proper execution of tasks.

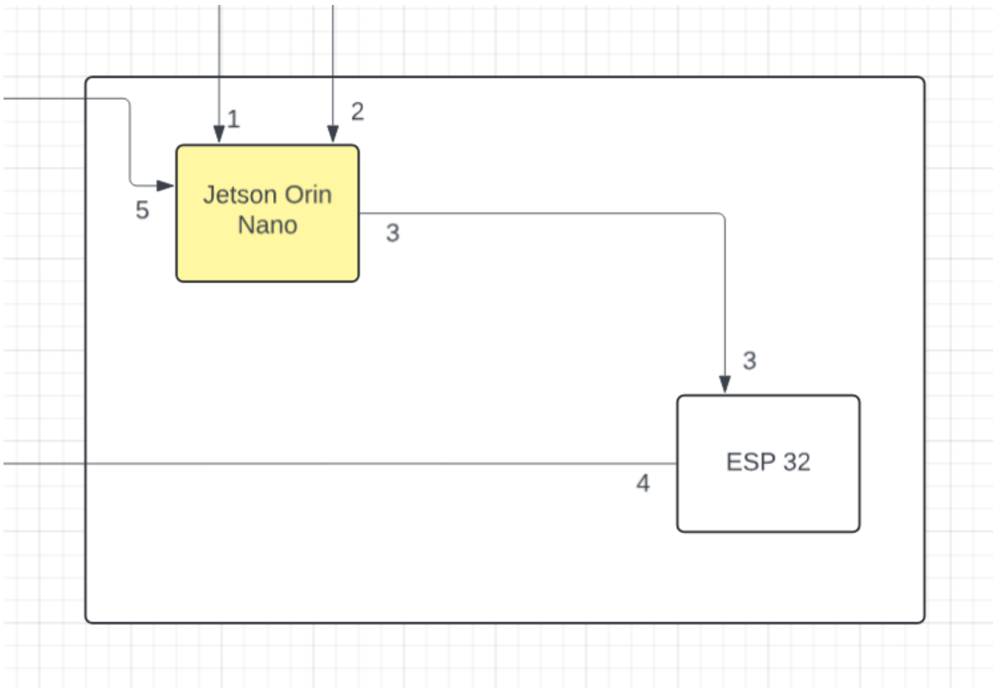

Figure 4: Controller subsystem diagram

4.3.1 Subsystem Hardware

The RPLidar and OpenCV logic will run on a Jetson Nano

4.3.2 Subsystem Operating System

The Jetson will be running Ubuntu 22.04

4.3.3 Subsystem Software Dependencies

Node and subscriptions will be implemented in ROS2 Humble.

4.3.4 Subsystem Programming Languages

The ROS2 packages used by the subsystem contain Python code and openCV for lane detection.

4.3.5 Subsystem Data Structures

The subsystem will receive data from sensors in the form of sensor_msgs/LaserScan. These are standard message formats used by ROS2.

4.4 Arduino Subsystem

The Arduino subsystem consists of a micro controller that will be used control the speed and direction of the motors.

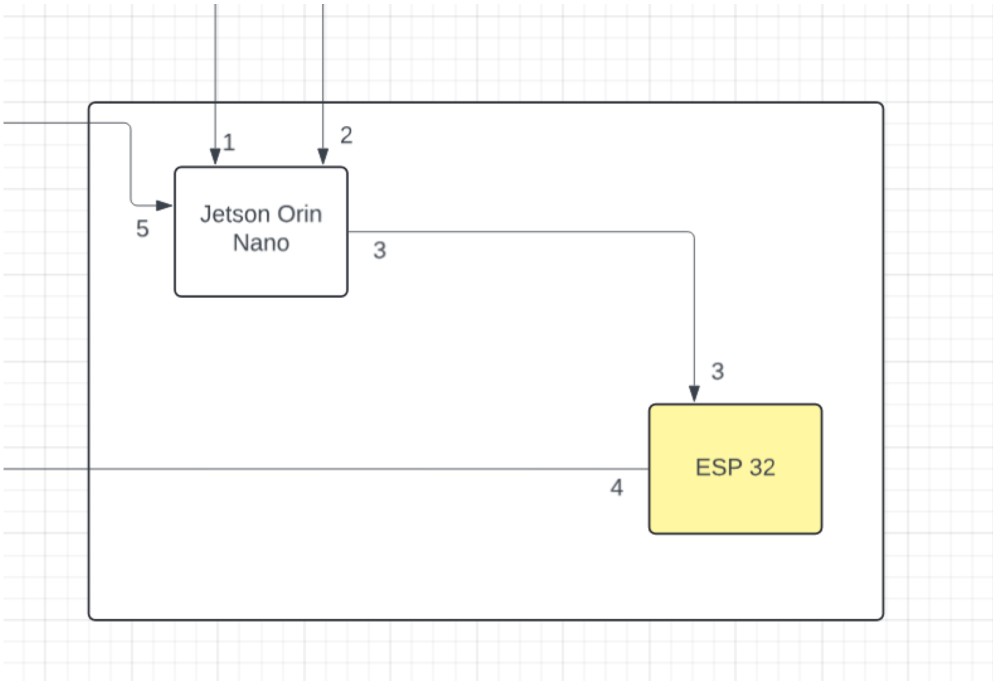

Figure 5: Power supply subsystem description diagram

4.4.1 Subsystem Hardware

The Arduino subsystem consists of an Arduino ESP 32 controller.

4.4.2 Subsystem Software Dependencies

This subsystem requires the use of the Sabertooth libraries available on an Arduino.

4.4.3 Subsystem Programming Languages

The programming language used is a variant of C/C++ that runs on the Arduino IDE.

5 Hardware Layer Subsystems

This layer consists of two lead acid batteries that act as the power supply of the entire system connected directly to the dual channel motor controller, and a relay and switching unit to supply power to all other components.

5.1 Layer Hardware

This layer consists of two lead acid batteries, two sabertooth 2x32 drivers, a 12 volts to 5 volts converter.

5.2 Layer Software Dependencies

This subsystem requires the use of the Sabertooth libraries available on an Arduino.

5.3 Relay and Switching Controls Subsystem

The hardware consists of the Flight Sensory unit comprising of a wide range of sensors and connections with the responsibility to regulate that there is adequate information for guided or unguided missions and help autonomous controls.

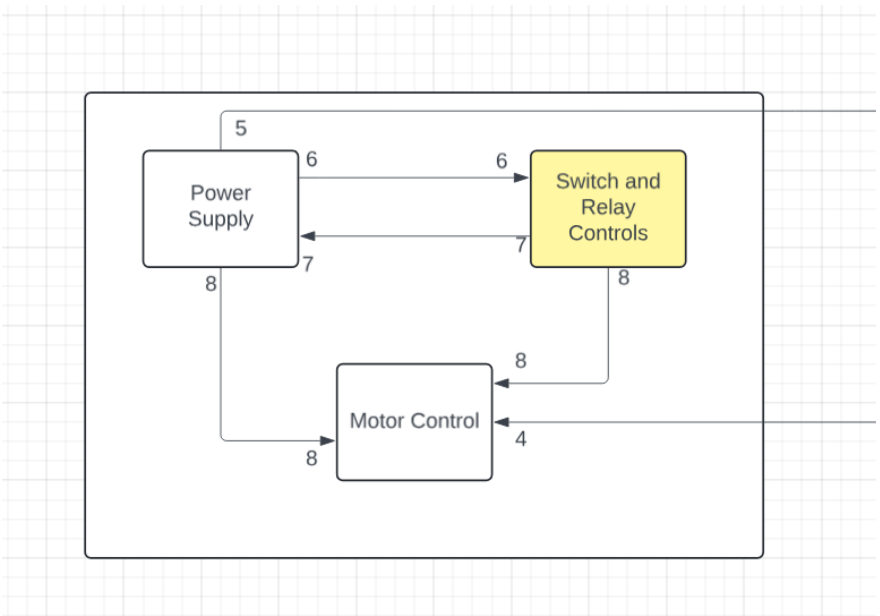

Figure 6: Hardware subsystem diagram

5.3.1 Subsystem Programming Languages

No programming languages are needed for this hardware layer.

5.4 Motor Control Block Subsystem

The motor control block subsystem consists of two dual H-bridge motor driver that receives input from an Arduino or radio control transmitters to control the speed and direction of the motors.

5.4.1 Subsystem Hardware

The motor control block subsystem consists of two Sabertooth 2x32 dual channel motor drivers.

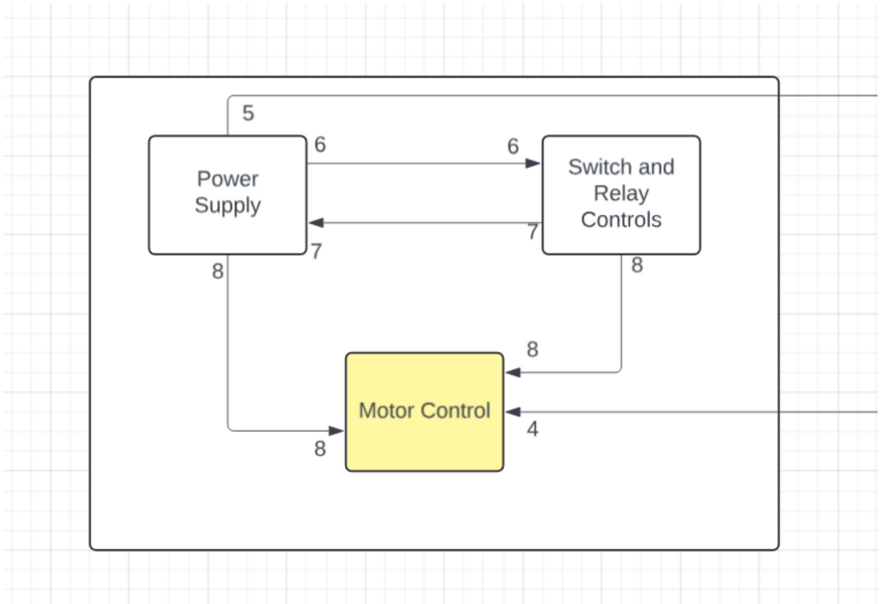

Figure 7: Motor Control description diagram

5.4.2 Subsystem Data Structures

This subsystem is able to receive and send data packets through a serial communication protocol that transmits data as ASCII formatted plain text strings, binary, four character packet serial commands, or servo signals/radio control pulses.

5.5 Power Supply Subsystem

This subsystem controls the source of power for the entire robot, communicating with the relay and switching controls to ensure that every subsystem has the necessary amount of current needed to operate safely.

5.5.1 Subsystem Hardware

Two led acid batteries are used directly connecting with two dual channel motor controllers.

5.5.2 Subsystem Data Structures

All power is to be regulated by the hardware, so no DC smoothing or offsets are present.

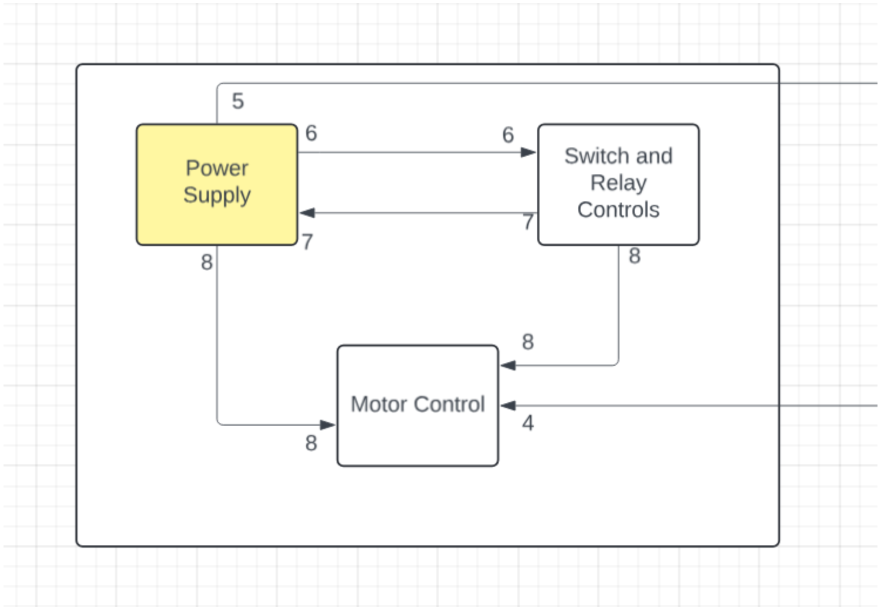

Figure 8: Power Supply interface diagram

REFERENCES