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

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2024

Heals On Wheels
HMETV

Christian Brown
Dyana Rahhal
Nicholas Doerfler
Rency Kansagra
Tristan Moses

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	11.02.2024	CB	document creation
0.9	11.11.24	ALL	complete draft
1.0	11.11.24	CB	public release
1.1	4.23.25	CB	final draft

Contents

List of Figures

List of Tables

1 INTRODUCTION

The HMETV (Hospital Medical Equipment Transport Vehicle) is a system of autonomous carts that deliver medical supplies, tests, or any other needed resources that would usually be delivered by a human nurse, to where they need to go in an efficient and reliable manner. The system takes orders through a mobile application, and then summons a supply cart to the desired destination with the requested supplies

2 System Overview

The system is interfaced via a mobile app on a cellular device where a Doctors, Nurse, or other hospital staff member makes a request for an item to be delivered to a specified location with specified urgency by a cart. This request is then encrypted and sent to the application backend, the brains of the operation. The application backend decrypts the request, caches it, and checks for any idle carts, to fulfill the request. If there is one available, then the cart is sent a location of where to first pick up the requested item and then a location to drop it off at. If a cart is not available it is placed in a priority queue, where it will wait until a cart is available. Once the cart is ready, it will use a pathfinding algorithm to determine the most efficient route to its destinations which it will then traverse using a drive system of 4 motors attached to omnidirectional wheels, while actively sensing obstacles which it will avoid using input from a LIDAR sensor. Once it has finished its delivery it will then report to the application backend that it is now idle and go back to the charging dock. If the cart is stuck or malfunctioning it is equipped with an emergency stop button which will kill all power to the drive system when pressed and alert the application backend that it has been stopped. Upon receiving the stop notification, the application backend will alert a human maintenance worker to the location of the stopped cart.

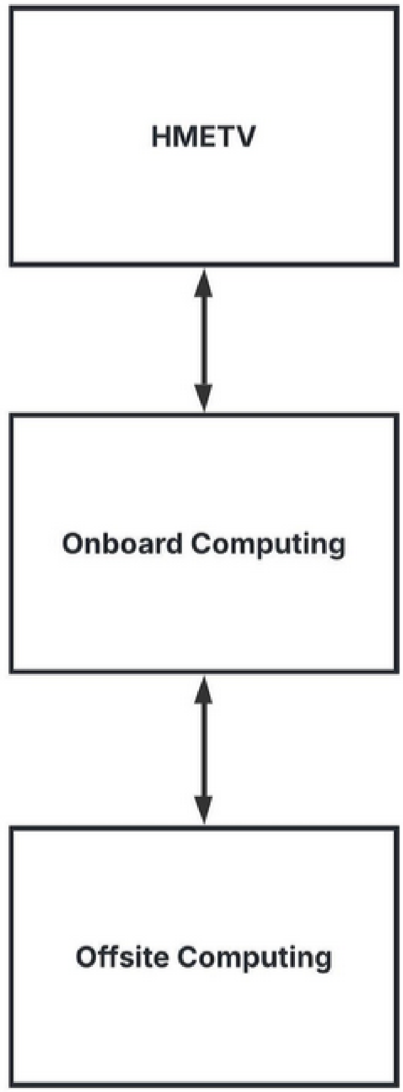

Figure 1: A simple architectural layer diagram

2.1 Onboard Computing

This layer is responsible for the HMETV's pathing and decision making, and interprets commands sent by the Application Backend in the Offsite Computing layer. This layer also gathers data from the environment to detect obstacles and orient itself using the Hector SLAM algorithm. The three components that comprise this layer are the Raspberry Pi, the Arduino, and the LiDAR Sensor. The Raspberry Pi does all the decision making and processes all data coming to the HMETV from various sources, all pathing decisions and communications with the Application Backend are through this subsystem. The Arduino translates commands from the Raspberry Pi into commands that can be read by the Drive System. The LiDAR sensor scans the area around the HMETV and reports the position at which an object was detected to the Raspberry Pi, this data is then used for navigation and mapping.

2.2 Offsite Computing

This layer is responsible for taking data and requests from users, and doing any record keeping and database management required. The two components that comprise this layer are the Application Backend and the Client Mobile Application. The Application Backend is a Microsoft Azure cloud service that sends commands to the HMETV, handles user requests, and performs all database operations. The Client Mobile Application is an application that allows hospital staff to make supply requests, view the status of a job currently in progress, and view the logs of past jobs. All subsystems of the Offsite Computing Layer communicate with other systems via RESTful API's.

2.3 HMETV

This layer is the physical vehicle that carries out the tasks of the system, such as locomotion, and also powers the system's electrical components. It consists of the Drive System, the Power System, and the Chassis. The Drive System is the collection of motors and motor control boards that cause the device to move. The Chassis was built around an ELAFROS Heavy Duty Plastic Utility Cart, and also contains the HMETV's Emergency Stop Button. The Power System supplies power to the Raspberry Pi and Drive System.

3 Subsystem Definitions & Data Flow

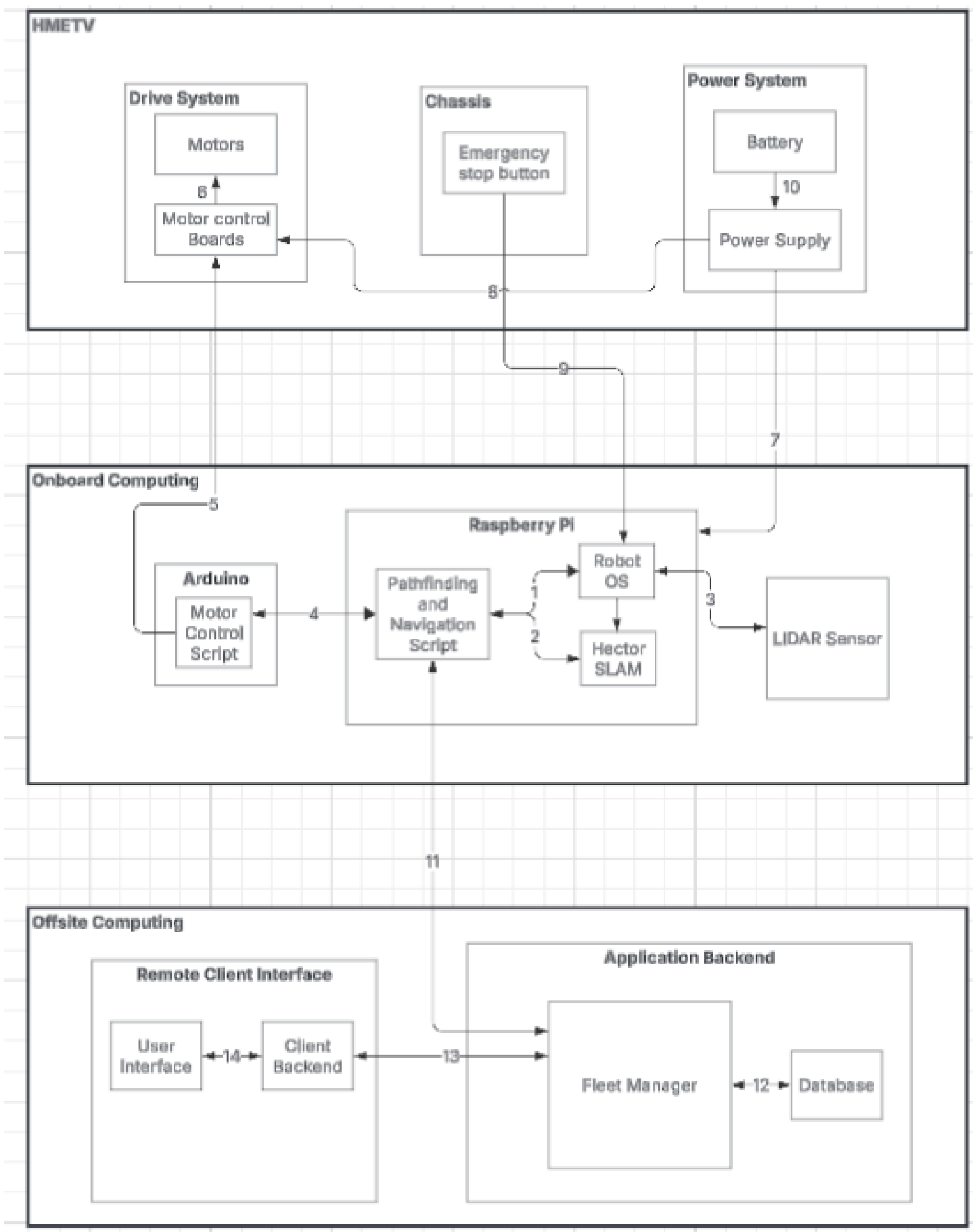

Figure 2: A simple data flow diagram

4 Onboard Computing Layer Subsystems

This layer handles the pathing, sensory data gathering, and remote control aspects of the HMETV. It is essentially the device's "brain". The Onboard Computing Layer consists of 3 devices working together in tandem, the Raspberry Pi takes care of wireless communication and navigation, the Arduino relays commands from the Raspberry Pi to the Motor Control Boards, and the LIDAR Sensor gathers sensory data to send to the Raspberry Pi

4.1 Raspberry Pi Subsystem

This is the core subsystem of this layer. It uses ROS as a central interface to communicate with the LIDAR Sensor, the Arduino, and the Application Backend. Data gathered by the LIDAR sensor is sent to ROS and then to Hector SLAM, which generates a map that will be used by the Pathfinding and Navigation Script. Pathing decisions are output to the Arduino, which will control the motors.

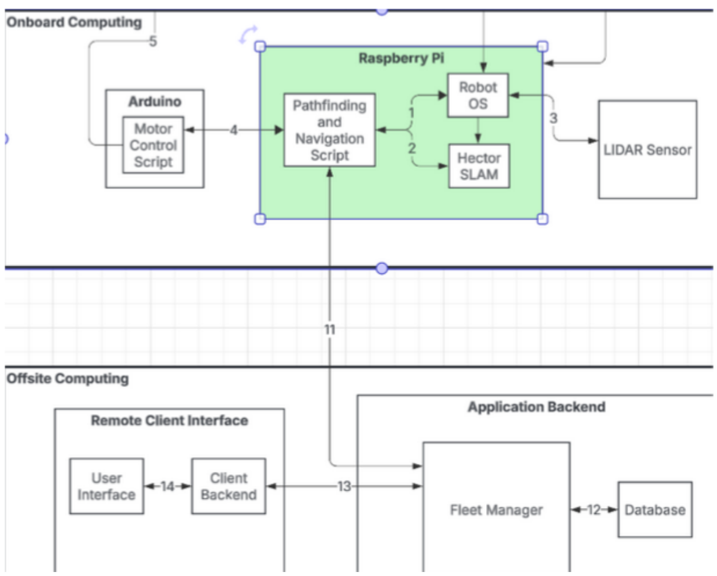

Figure 3: Example subsystem description diagram

4.1.1 Assumptions

- It is assumed that the hardware we use in this layer will have enough compute power to use Hector SLAM and perform its various other tasks.

4.1.2 Responsibilities

The Raspberry Pi is responsible for navigation and mapping, as well as communicating with the Offsite Computing layer for instructions and data processing.

4.1.3 Subsystem Interfaces

Interface	Input	Output
Application Backend	Commands, data updates, and system configuration	Acknowledgment, status updates
LIDAR Sensor	Mapping Data, Status	Scan Commands
Arduino	Health Status	Commands for Motor Control
Power System	Status	Status

4.2 Arduino

This subsystem is responsible for motor control and executing commands provided by the Raspberry Pi layer. It translates commands from the Raspberry Pi to instructions to give to the Motor Control Boards.

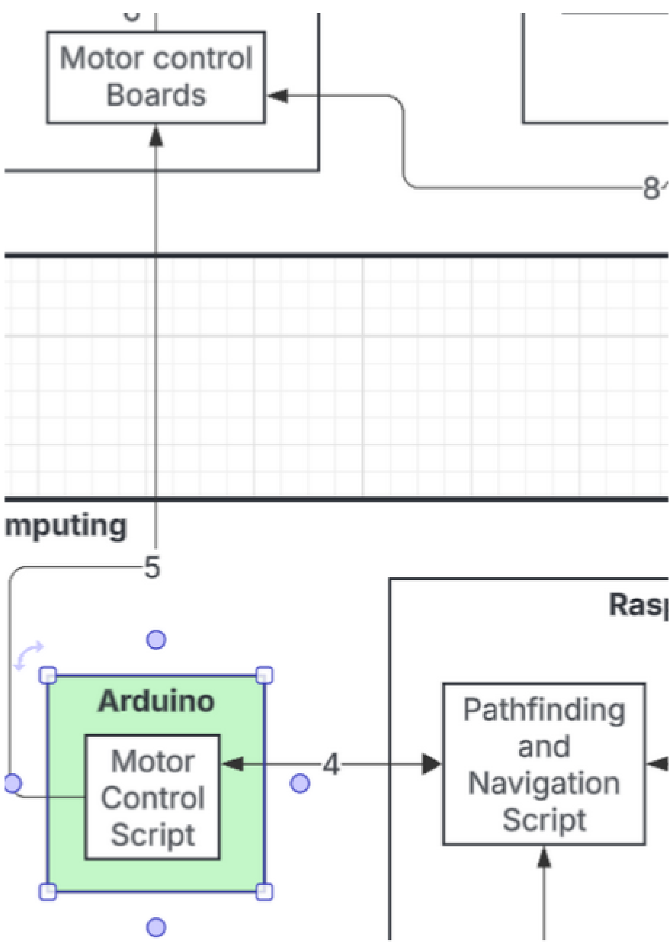

Figure 4: Example subsystem description diagram

4.2.1 Assumptions

- It is assumed that there is a well-defined interface between the Arduino and any hardware it may connect to

4.2.2 Responsibilities

The Arduino subsystem is the main interface between the HMETV and Onboard Computing Layers. It is responsible for interpreting and relaying instructions from the Raspberry Pi to the Sabertooth Motor Control Boards

4.2.3 Subsystem Interfaces

Interface	Input	Output
Raspberry Pi	Commands, Power	Status
Drive System	Status	Commands

4.3 LiDAR Sensor

This subsystem gets data from a rotating Slamtec A1M8 LiDAR scanner and sends that data to the Raspberry Pi over USB connection

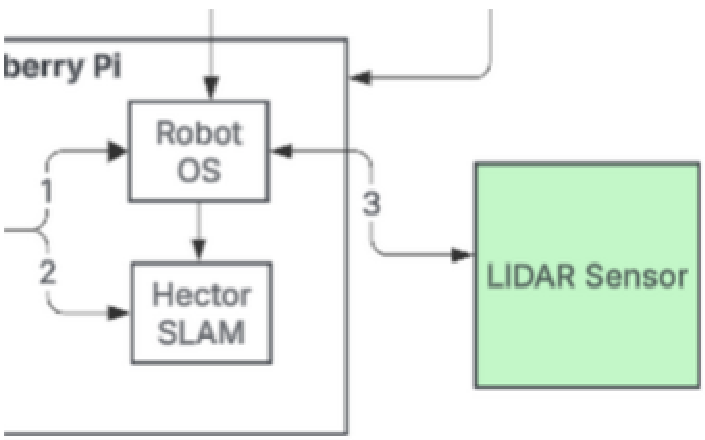

Figure 5: Example subsystem description diagram

4.3.1 Assumptions

- It is assumed that the LiDAR sensor will receive enough power to perform its tasks
- It is assumed that the environment will be reflective enough for the LiDAR to detect objects nearby

4.3.2 Responsibilities

The LiDAR Sensor is responsible for gathering data from the environment to detect obstacles and walls nearby so that the navigation system can use that data to go around said obstacles or use them as reference points

4.3.3 Subsystem Interfaces

Interface	Input	Output
Raspberry Pi	Commands	Mapping Data, Status

5 Offsite Computing Layer Subsystems

The Offsite Computing layer consists of the Client Interface, a mobile application that allows nurses to make requests to the system, and the Application Backend, a cloud service that processes user requests and takes care of all database and record keeping functionality.

5.1 Application Backend Subsystem

This subsystem is a cloud service that accepts data from the user application and sends jobs to an HMETV as requested. It records all requests and the times they were completed in a MySQL database. It communicates with the user application and the HMETV fleet via simple TCP Networking

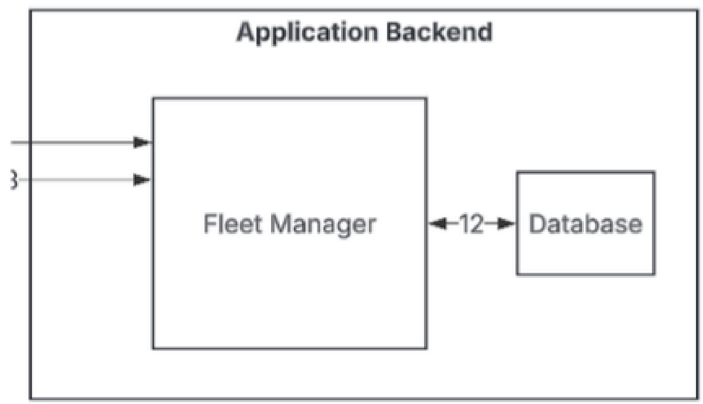

Figure 6: Example subsystem description diagram

5.1.1 Assumptions

- It is assumed that this cloud service will be paid for and maintained by the customer throughout the duration of the project
- It is assumed that this service will have enough storage to host its database and perform its tasks
- It is assumed that this service will have sufficient network connectivity and compute power to execute its tasks in a timely manner

5.1.2 Responsibilities

This system is responsible for sending jobs to the HMETV units, as well as keeping record of any requests made

5.1.3 Subsystem Interfaces

Interface	Input	Output
Raspberry Pi	Jobs	Status Updates, Job Completion Notification
Client Application	Supply Requests	Records of past and currently scheduled jobs

5.2 Client Application

This subsystem is the mobile application that hospital staff can use to schedule a supply order to a room in the hospital. It has user authentication to ensure that all requests come from legitimate hospital staff. This subsystem relays requests to the Application Backend layer, and can request data from the Application Backend if the user wants to view the status of an order or look through the request/delivery history.

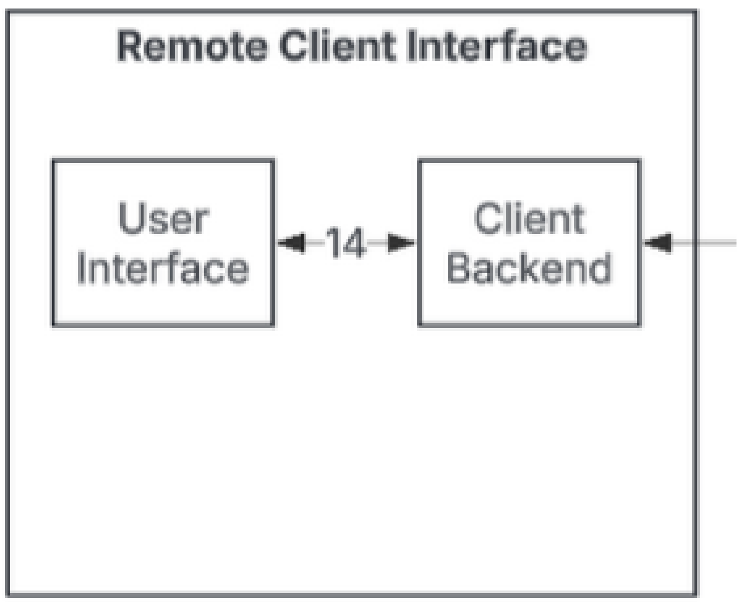

Figure 7: Example subsystem description diagram

5.2.1 Assumptions

- It is assumed that all hospital staff have a device capable of installing and running this mobile application
- It is assumed that the devices running this application will have sufficient network connectivity

5.2.2 Responsibilities

This subsystem is responsible for processing user orders and acting as an interface between hospital staff and the HMETV's database

5.2.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

5.2.4 Subsystem Interfaces

Interface	Input	Output
Application Backend	Order status/records	Job Requests, Data Queries

6 HMETV Layer Subsystems

The HMETV Layer consists of the pieces of the physical product that don't require programming or logic. This layer is entirely hardware, and includes the chassis and other physical/mechanical components of the product.

6.1 Chassis

The chassis subsystem is the frame that holds the onboard computing components, and it is the physical foundation that the rest of the physical components of the system were built around. The chassis also contains the HMETV's emergency stop button, which will halt the system and cause it to begin its recovery procedure when the button is pressed by a human.

6.1.1 Assumptions

- It is assumed that the chassis is structurally sound enough to support the rest of the system

6.1.2 Responsibilities

The chassis is responsible for ensuring the structural integrity of the HMETV, and the emergency stop button ensures that there is a way for a human user to stop the system should it go rogue and start performing undesired behaviors.

6.1.3 Subsystem Interfaces

Interface	Input	Output
Raspberry Pi	N/A	Emergency Stop Signal

6.2 Power Subsystem

The power subsystem is responsible for giving power to the Onboard Computing and HMETV layers.

6.2.1 Assumptions

- It is assumed that the hardware will have enough interfaces to accommodate all the system's electrical components.
- It is assumed that the power supply will output enough power to fully satiate all the system's electrical components running at full capacity.

6.2.2 Responsibilities

The Power Subsystem is responsible for powering the HMETV and Onboard Computing layers

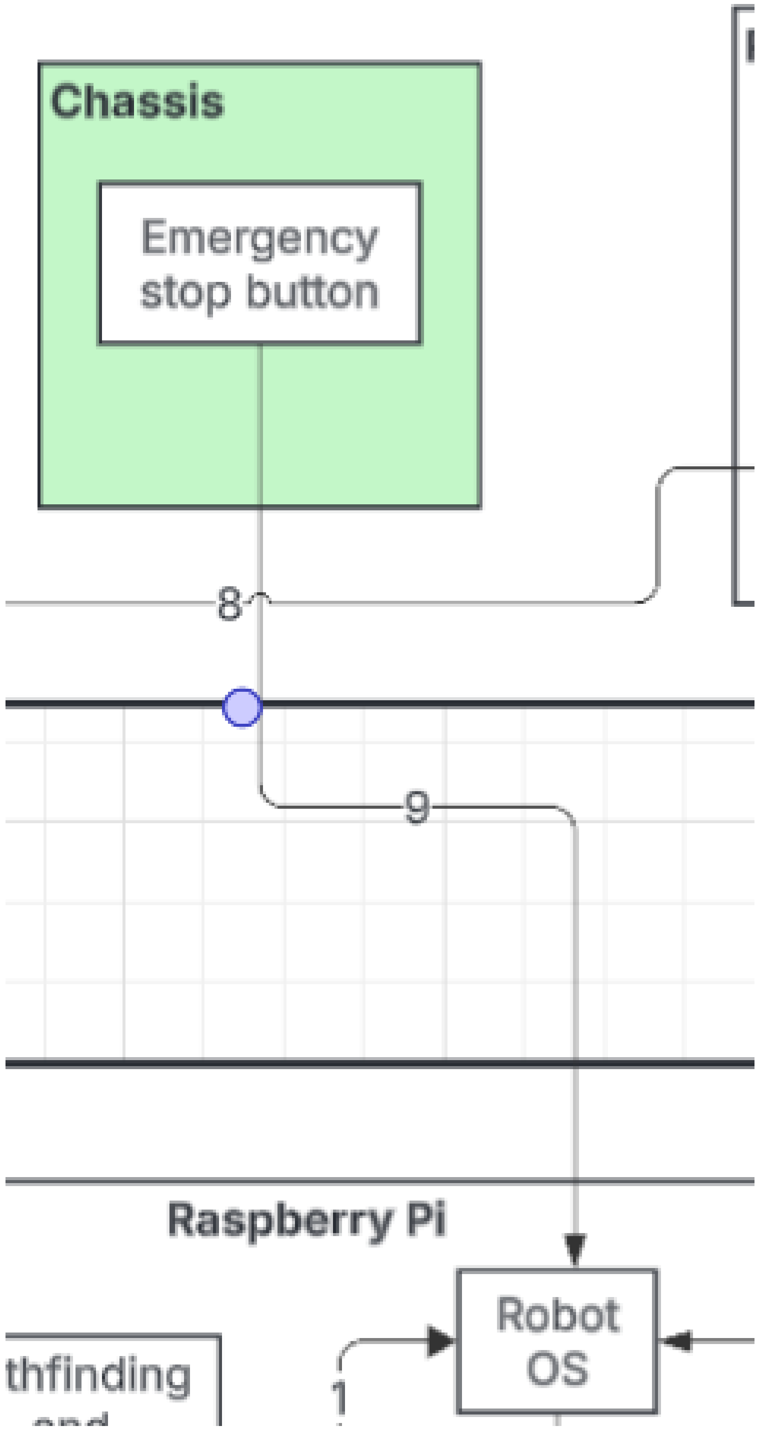

Figure 8: Example subsystem description diagram

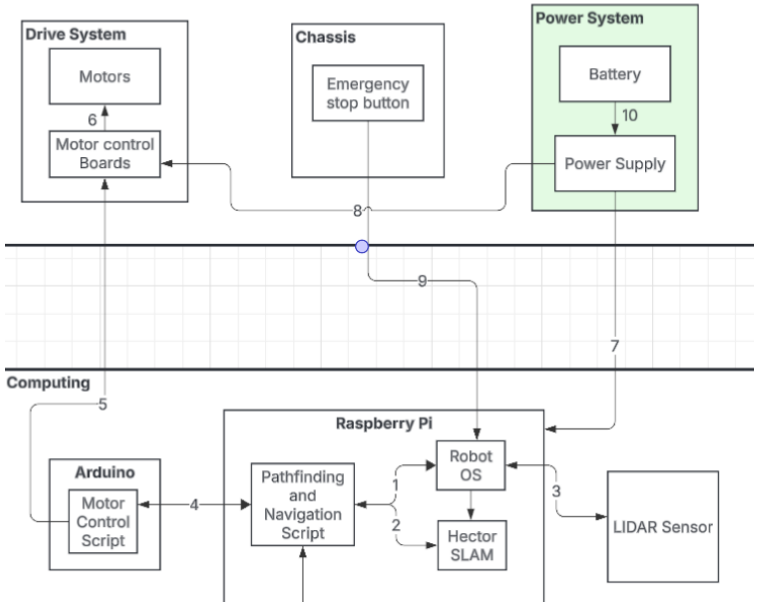

Figure 9: Example subsystem description diagram

6.2.3 Subsystem Interfaces

Interface	Input	Output
Raspberry Pi	Status	Power
Drive System	Status	Power

6.3 Drive System

The Drive Subsystem is responsible for the locomotion of the HMETV, it consists of 2 Sabertooth Motor Control Boards connected to 4 motors

6.3.1 Assumptions

- It is assumed that the motors provided to us will be able to move the HMETV at a speed that meets the requirements we were given

6.3.2 Responsibilities

The Drive System is responsible for moving the HMETV in the direction specified by the Navigation system

6.3.3 Subsystem Interfaces

Interface	Input	Output
Arduino	Commands	Status
Power System	Power	N/A

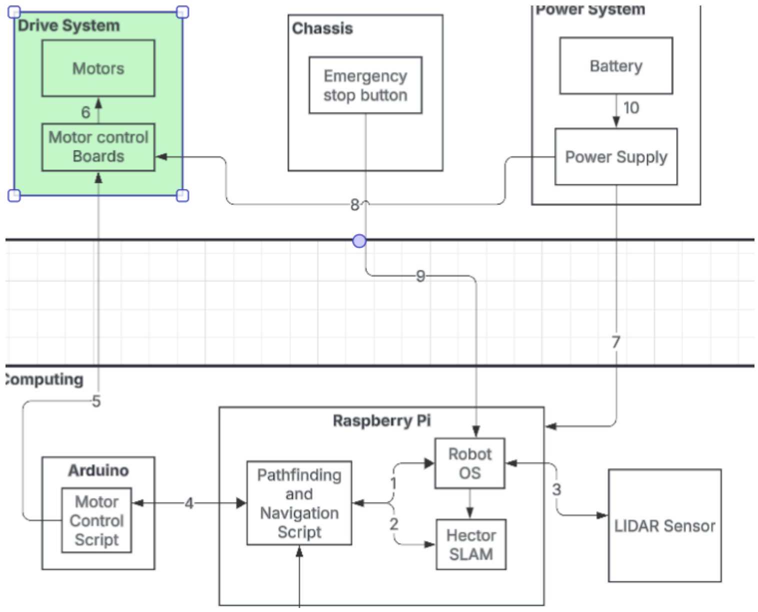

Figure 10: Example subsystem description diagram

REFERENCES