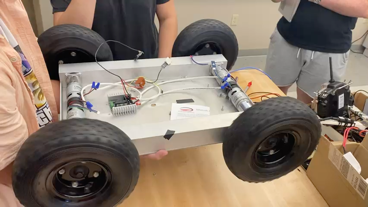
Name: Andrea Gamble

Date: October 7, 2024

Today I worked on the preliminary controls for the robot movement. The hardware team has established that the motor works and all four wheels turn:



Double click to play video

Since the Raspberry Pi has not come in yet, I used the Arduino to set up the motor controls. Using the hardware manual for the Kangaroo, all of the variables were set up, including:

<https://drive.google.com/file/d/1rsKM1XLbuf_G3P5VXRqTkzsoTTBM8DyI/view?usp=sharing>

* Baud Rate
* Full forward for motors 1 and 2
* Full Reverse for motors 1 and 2
* Full stop for motors 1 and 2

A person working on a machine

Description automatically generated

Testing concluded with the understanding of forward and backwards direction for the robot as well as the distinguishing between motors 1 and 2. From testing, the software team has learned that one side of the board is not sending enough voltage for essential functionality. This board has been added to the list of parts.

Further development includes:

* Creating methods for forward movement, backward movement, turning left and turning right.
* Implementation of these programs for autonomous movement

Date: October 20, 2024

Today I worked on the modifications of the software proposal. The team has switched back to the original project: Autonomous Survival Detection Vehicle, which incorporates body scanning to determine if a human is alive or not. The original project also implements a camera, which plays a big role in the software team. Many of our objective remained the same:

* Connecting two Raspberry Pis using P2P communication
* Developing algorithms for motor movements and obstacle avoidance
* Test and maintain software including error handling and recovery

However, a few of our objectives have changed. Since we are no longer reading qr codes to determine a friendly or enemy robot, OpenCV implementation with LiDAR is no longer in the scope of the project. Additionally, with the new implementation of the camera for scanning bodies, the software team is currently researching ways for the added camera part for object detection and tracking in python. The semester-based sprint goals have only changed subtly. These changes include removing CloudCompare research and implementation.

The software team will still need to continuously refine the project to create a clear objective and address any upcoming issues that present throughout the year.

Date: October 24, 2024

Today, I worked on research for live object detection using a laptop in python. I learned that there are several trained object detection models and for utilizing object detection. The difference between the object detection models depends on what the project needs are. The trained model decided would be best suited for object detection is YOLOv3. <https://pjreddie.com/darknet/yolo/#google_vignette> This trained model is used primarily for real-time object detection algorithms. It is beneficial because it provides high accuracy and speed and allows for multiple objects, which is necessary for our project since we need to detect objects, people, and identify their classification for each. This is especially useful for autonomous vehicles, which is the goal of the AGV-HSD project.

The next step is to use the trained model and develop an algorithm suitable for the AGV object detection. Following this step, we can move on to classifying objects in the camera view with testing to ensure accuracy in classification. Lastly, we need to implement the algorithm using the camera sensor on the AGV itself.

October 26, 2024

Today, I did research on how to connect two Raspberry Pi 5’s using P2P communication. A main component of the AGV-HSD is the use of wireless connection to establish secure and efficient communication between the AGV and ground station. From research, I have learned the process in connecting these two devices. They would work through a client-server architecture, which involves setting up a direct network link between them without needing an intermediate network. For the AGV-HSD project, the direct link will be through Wi-Fi. We will need to create a configuration file for the Wi-Fi on both devices. They will both need to be assigned an IP address, and the connection will need to be tested to ensure the connection is feasible.

The next step is to develop the algorithm for the device connections and test the algorithm to ensure it is still sufficient using the LoRa sensor.

October 27, 2024

Today, I worked on the context diagram draft for the SRS. Based on SE310, I created a very simple system that utilized only the AGV, the ground station, and the disaster environment. The main functions of the AGV-HSD project is to detect and identify human survivors in he disaster zone of the AGV. From this broad scope, the main functionalities were included in the high-level context diagram:

* The sensor data inputs come from the disaster zone and are received by the AGV-HSD system.
* The AGV-HSD system will communicate status updates back to the ground station. This can include AGV status, sensor status, or survivor detection information.
* The ground station will send commands to the AGV-HSD system depending on what is needed in the moment. This can be manual motor control commands, dialog from a speaker to communicate with detected survivors, or commands recalling the AGV to a recharging station.

A diagram of a system

Description automatically generated

Now that a foundational understanding of the system is established, this can be used to create a draft of functional and non-functional requirements of the system.

The next step will be to use the level 0 context diagram to create the level 1 context diagram for further understanding of the AGV-HSD system.

October 28, 2024

Today, I worked on the level 1 context diagram for the SRS. I developed this diagram based on the initial context diagram of the AGV-HSD system. The complexities developed form the main functionalities of the system include:

* A communication module to process incoming data from the ground station into data for the AGV-HSD system.
* A navigation module that will use processed data from the AGV environment to determine the movements and path of the AGV.
* The data processing module that will receive the input data from the AGV environment and process it into data for the ground station to understand and utilize.
* A data collection module to collect sensor data from the AGV environment to send to the data processing module.
* A sensor data store to temporarily store raw sensor data such as heartbeat data, body temperature data and time stamps.
* A processed data module to temporarily store processed data such as survivor identities and GPS locations of the AGV.

*A diagram of a data processing system

Description automatically generated*

Now that a foundational understanding of the system is established, this can be used to create a draft of functional and non-functional requirements of the system.

October 29, 2024

Today, I worked on creating the level 2 context diagram for the SRS. The level 2 illustrates in more complexity the modules and processes for the AGV-HSD system. Because there wasn’t a full foundational understanding of the project at this time, a lot of this complex diagram are a draft and need to be further understood. The main implementations added from the level 0 and level 1 diagram are as follows:

* The ground station will be able to send commands to control the AGV navigation module
* The ground station commands will be received and process to control the AGV nagvation module
* The AGV environment will allow sensor data inputs to be collected from the temperature sensor, heartbeat sensor, speaker, and GPS and LiDAR data.
* The raw sensor data will still be temporarily stores in the sensor data store
* The temperature, heartbeat, and speaker data will be used to detect and classify survivors
* The detected and identified survivors will be temporarily stored in the detection classification store
* The classification data will be processed to transmit this information back to the ground station for further investigation

*A diagram of a system

Description automatically generated*

The next step is to talk to the software team and see if they have any modifications to the current diagram. Then, I will use the diagrams created to make some functional and non-functional requirements for the AGV-HSD system.

November 1, 2024

Today I worked on one of the sprint backlog items to determine feasibility. During the last class, the parts for the AGV-HSD project came in. Since I have never worked on a Raspberry Pi 5, I did some research to understand exactly what the process of using it is. I watched a video on installing ubuntu onto a Raspberry Pi 5.

[Video: Raspberry Pi 5 Ubuntu install in 8 minutes. David Bombal Tech. YouTube](https://www.youtube.com/watch?v=5CBYGz_mO9U)

I found out the process for installing Ubuntu onto a Raspberry Pi 5 is not difficult.

* Need to use a laptop that can take in an SD card reader. I don’t have one on my laptop, so I need to order a SD card reader for laptop. I let my group know, and they said that you can also use an HDMI cable to connect your device to the Raspberry Pi 5. I will try to do this sometime before the weekend.

November 2, 2024

Today, I worked more on the object detection aspect using a laptop camera using Python and OpenCV. The website for YOLOv3 helped with created the code for the object detection file. However, I had some difficulties creating my environment for the file to run in. I need to have the yolov3.weights file running, but the GitHub was a link to a much larger file. I spent about an hour trying to get find out how to access to the file but couldn’t figure it out. I’m not very savvy with GitHub, so I might just try to find another way to get the object detection for the laptop. <https://github.com/patrick013/Object-Detection---Yolov3/blob/master/model/yolov3.weights>

November 3, 2024

Today, I continued working on the object detection using Python and OpenCV. I found a simple tutorial on how to get the object detection done. <https://dipankarmedh1.medium.com/real-time-object-detection-with-yolo-and-webcam-enhancing-your-computer-vision-skills-861b97c78993> They have the code laid out with the explanation. Initially, I was having trouble getting the python file to run because I have not worked in terminal much before this project. I realized that I needed to download PyTorch into my virtual environment and make sure I didn’t have a version of Python higher than 3.11 installed. I also needed to have pip3 installed and ultralytics as a efficient way to access the YOLO files. I had the most trouble getting the ultralytics installed. I kept getting an error saying the resolution not possible for the environment I was it. After some research, I found out I only needed to install ultralytics without the dependencies because it made the file too large. Finally, I was able to get the file running through my environment. <https://drive.google.com/file/d/1_XVrM5R7IBTNJnlzYppvgcL_uhfXX98r/view?usp=sharing>

The next step is to add this code to the GitHub. I will wait until I have a better understanding of the structure of the code environment before I add it to a random place. I will also need to refine the file to specify what objects I want to be classified determined by the scope of the project. The file currently has a template of objects that can be classified, so I will delve into more research focused on how to modify this.

November 4, 2024

Today, I did research on the SQL database for storing user information. I watched a YouTube video on a person who went through the steps and infrastructure to SQL. I have worked with MySQL before, but I’m not sure the difference between MySQL and other RDMSs, so I thought I should do some research before jumping into it. However, following this, I realized we are using SQLite 3, which is not the same as the video I watched. I watched a video on Microsoft SQL.

The next step is to figure out how to work SQLite 3 so that we can implement this into our project for saving data, user login information, and identified survivors.

November 5, 2024

Today, I worked on the collaboration graph for the SDD. As part of the Software Detailed Design implementation for the AGV-HSD project, I labeled the following modules with their functions, conditions of use, overall processing and logic, interfaces, and security requirements:

* Navigation Module
* Survivor Detection Module
* Communication Module

Using these modules, I was able to create a draft of the collaboration diagram to show the interconnections between subsystems and how they collaborate to the overall goal of human detection and identification.

*A diagram of a security system

Description automatically generated*

I then went on to give a brief overview of each of the subsystems and their purpose in contribution to the AGV-HSD project. Overall, I feel like this diagram will give the software team a better understanding of the direction the project is going in in terms of the scope of the project.

The next step will include going further into detail with each specific subsystem and their contracts in detail. I plan to get this done hopefully by the weekend.

November 5, 2024

Today during class I learned more about how to add items to Git and the process of committing and pull requests. Matthew created a development branch, so all commits will need to go through the development branch before being added to the main branch. I was able to look at the code for the P2P connection that Matthew did so that I could have a better understand of how the wireless connection between the two Pi’s will work. Overall, I have a better understanding of the code implementation since I have been heavily focused on documentation these past few weeks. I was also able to see the SQL database that Matthew created, and the software team went over the scope of the database so we all could have a better understanding of the direction we are going in.

The next step is to add the object detection code into the document without getting any merge conflicts. However, the last time I tried to add my engineering notebook to the GitHub, for some reason the commit wasn’t adding. This time I will just drag to upload my engineering notebook to the GitHub rather than committing the document through Git.

November 6, 2024

Today, I worked on the subcomponents of the subsystem diagram. I added contracts for the navigation subsystem and the survivor detection subsystem. Since the AGV-HSD project will be using database management for storing survivor information, sensor data, and user login information, the database management subsystem and contract remained the same from the template. The only difference in the database management subsystem is that type of database management system we are using. We will be using the SQL instead of MySQL, so the information in this module was modified accordingly.

The next step will be to finish the documentation for the subsystem diagram including the communication subsystem and the ground station subsystem.

November 7, 2024

Today, I worked on the documentation of the communication subsystem for the collaboration graph including the contracts. The communication subsystem only had one contract, which is data transmission to the ground station, received through the LoRa Hat. The information received from the data transmission contract will determine the navigation of the AGV. Additionally, the Ground Station can alter navigation from the data received by communication subsystem. The communication interface will display the information for the Ground Station to view. The Ground Station can then use the information received and add it to the database for further analysis and/or machine learning.

The next step will be to finish the final documentation for the subsystem diagram which includes the Ground station subsystem.

November 8, 2024

Today, I worked on the documentation for the Ground Station subsystem for the collaboration graph, including the contracts. Similarly to the communication subsystem, the ground station subsystem has two contracts. The first contract is to receive data coming from the communication subsystem. The data collection will come in raw form for the database implementation. The other contract has to do with the interface for the ground station to collect this data and process it for logging into the database.

The next step will be to finish any documentation not yet completed from the software team. Additionally, we are also working to install Ubuntu onto the Raspberry Pi 5 so we can begin the real implementation and testing of the project.

November 9, 2024

Today, I worked on research for installing Ubuntu onto the Raspberry Pi 5. The communication team was previously working on the installation process. However, they notified us that the microSD card was not reading properly, and we needed to get a new one for the project. After some quick research, I found out that you can use either a microSD card or a USB drive to boot ubuntu onto the raspberry pi. I found this video : <https://www.youtube.com/watch?v=5CBYGz_mO9U> to for a seamless installation process so that I had some background knowledge about how the pi works before going to class to test it.

The next step will be to get a USB drive with enough storage to install it onto the Raspberry Pi 5.

November 12, 2024

Today, I worked with the software team to get ubuntu installed onto the Raspberry Pi 5. The communication team found us a USB drive with enough storage, so we began working on that. We tried using a laptop and Raspberry Pi Imager to download ubuntu onto the drive. The process of installation took the entire class time, so we didn’t get to test if the installation process was successful or not. However, we did get ubuntu installed onto the device. When we tried to boot it, there was a problem with connecting the HDMI cord to the computer monitor because we didn’t know how to see the Raspberry Pi OS.

The next step will be to use a monitor to see if Ubuntu has been installed on the pi. Additionally, I will continue working on the SDD for submission.

November 13, 2024

Today, I worked on the documentation for the SDD submission. I haven’t worked with software interface detailed design previously, so I had to do some research to understand what to implement. After talking to the team, I was able to understand how the interface section works. I included the initial handshake protocol, which includes making sure both the AGV and Ground Station have an established connection before beginning. There was also a template created in the event of an error. The same process was conducted for the query and response protocol, which entails the ground station requestion information from the AGV through the communication subsystem. This data includes heartbeat data, body temperature data, and vocal response data queries with their response protocol.

A diagram of a server

Description automatically generated

The next step will include continuing with the installation of ubuntu onto the Raspberry Pi 5 and any other documentation for the SDD submission.

November 14, 2024

Today, I continued working with the Software team to install ubuntu onto the Raspberry Pi 5. With the software installed, we connected the Raspberry Pi 5 to the lab computer monitor to see the Raspberry Pi 5 OS running. From there, we were able to boot Ubuntu onto the Raspberry Pi 5. However, there was an issue with the initial boot. The Raspberry Pi 5 needed more voltage to continue running Ubuntu on the device. We believe the cause of this low voltage pull is the size of the storage device, causing too much power needed for the storage. We informed the communication team to buy the microSD card with the idea that the storage won’t be enough to cause low voltage errors.

The next step is to do some research on Ubuntu to understand the system before the microSD card arrives so we can smoothly transition into the implementation for the project. Additionally, the documentation for the SDD submission.

November 15, 2024

Today I worked on the Interface Architecture for the AGV-HSD project. I came up with 4 interfaces. The interface needs a way for the sensors to be grouped together, a link between the AGV and main server, and a manager for the facilitation of communication between the AGV and ground station.

Next, I developed a preliminary communication path diagram. The main modules for communication are the central server, which will use HTTP to connect to the onboard computer. WebSocket is a communication protocol for real time communication between the AGV (central server) and ground station. UDP is used for communication between the onboard computer and the sensor systems. I was not very familiar with this, so my plan is to come back to this point so that I have a better understanding of exactly what the interface architecture is the importance of this on the project. I believe it will make more sense once there are real implementations of the software for the project.

The next step will be to work on learning ubuntu before the microSD cards arrive. Additionally, the next step will also be to revise the SRS for version 2 updates.

November 20, 2024

Today, I worked on revisions for the SRS version 1 document. After seeing the TA feedback, I rephrased one of the security requirements to make it more understandable for the audience. Additionally, I edited the security requirement for IP addresses since the IP address does not define the amount of security given. Lastly in the security section, the hash algorithms were specified for user authentication with text passwords. In the maintainability section, I added some additional context for when a maintenance notification should occur. We needed some non-functional requirements for the vocal response data as well to give the reader a better understanding of how the interactions of the vocal response data with the system will occur. The data flow diagram was updated with an additional interaction between the navigation module and communication module. The navigation coordinates should interact with the communication module in order to alert the ground station.

The next step will be to work on connecting at least one of the radars to the raspberry pi 5. I do not have any experience with this process, so I will need to do some more research to gain a better understanding.

November 21, 2024

Today, I worked on research on how to connect the radar to the raspberry pi 5. After doing some preliminary research, I believe this is something the communication’s team would oversee since they handle more computer engineering. Once they get the connections to work properly, the software team can utilize the software to implement the two devices.

The next step will be to work on the slides for the10-minute video presentation as well as revisions to the poster and revisions for the SDD. Overall, the project is making steady progress.

December 2, 2024

Today, I worked on the slideshow for the 10-minute presentation. I added all the necessities according to the rubric that were needed from the software team, including the system architecture diagram, collaboration diagram, and UML diagram. I also added some notes alongside the diagrams to give brief but thorough context of the diagrams. I added the software team objectives for the project as well as the interfaces we will be working with into the slides.

The next step will be to record the 10-minute video presentation with the software team. We also need to work on the test plan, so I will begin that as well.

December 3, 2024

Today, I worked with the software team to film to 10-minute presentation for submission. The software team used zoom to record the video. We chose our assigned slides and made sure we each understood what each person was presenting. The entire recording took us about 30 minutes because the internet connection was slow, and we had to test the video to make sure we could see our faces. We also wanted to be sure that the video would download as an mp4. We were able to send the video to the hardware team for editing and final submission.

The next step is to work on the System Test Plan for V1 submission. Additionally, the revisions to the SDD for V2 submission.

December 5, 2024

Today, I worked on the test plan for the submission on V1 of the STP. I made a draft of the test cases according to the requirements 4-10 along with the steps. Peter created an excel sheet for the team to utilize to make filling in the STP easier. The test cases were made for the following requirements:

* The User Interface (UI) shall display real-time sensor readings from the AGV.
* The UI shall display real-time vehicle status from the AGV.
* The UI shall display in real-time when a survivor is detected from the AGV.
* The UI shall provide a graphical representation of heartbeat sensor data in the form of a graph.
* The UI shall provide a graphical representation of body temperature data in the form of a heat map.
* The UI shall require user authentication for access once the system is powered on.
* If the heartbeat sensor detects a pulse within the human’s range of 4 meters, the system shall classify the detection as human to further investigate.

Each test case according to requirements above had at least 5 steps. However, the issue I ran into included the redundancy of certain preliminary steps. These steps included:

* Ensuring the AGV and other test subjects are on and operating correctly.
* Logging into the UI system.

The next step includes creating a UI for the AGV ground station. The UI will have various modules for survivor detection, heartbeat data, body temperature data, status updates, etc. Additionally, the STP will need to be finished so more tests can be done on the system.