Ready-to-Serve Trash Bin
Final Report

Team #19

Dongming Liu
Josh Litao
Owen Xu

TA: Jason Zhang

May 1, 2024
ECE 445

Abstract

The Ready-to-Serve Trash Bin is a project that aims to serve people with limited mobility in disposing of their trash. This project uses a camera and computer vision (CV) to detect the hand gesture that calls the bin, and this system communicates with the trash bin over a WiFi connection. The bin is equipped with wheels controlled by DC motors and a linear actuator to control the lid. This project is successful and can identify hand gestures, move towards the person requesting the bin, collect the trash, and return to its original location.

Table of Contents

1 Introduction

1.1 Problem

Throwing away trash is a simple task that many people take for granted. However, those with little to no mobility as a result of a disability, hospitalization, natural aging, or other health conditions struggle to carry out this necessary task. According to the CDC, 12.1 percent of U.S. adults have a mobility disability with serious difficulty walking or climbing stairs [1]. As a result, these people either require an assistant to dispose of their trash for them, which may not always be feasible, or they are forced to hold their trash and let it accumulate by their side. Letting trash accumulate is a sanitary concern that could escalate into further problems. A trash bin could be placed next to the person, but this solution has various problems. An open trash bin would allow the odor of the trash to spread throughout the room, and a bin with a lid could pose difficulty for users whose conditions make them unable to open the lid directly with their hands or use their foot to press the pedal.

1.2 Solution

In order to eliminate the problems with existing trash bins for people with limited mobility, we propose a trash bin that would be ready to take a user's trash once they perform a particular hand gesture to call it. A camera will be part of the motion and object detection system. This system to detect the hand gesture would be placed somewhere in the room where it would be able to monitor whether the user needs the trash bin to pick up their trash. The trash bin would be attached to a set of wheels to allow it to move. The lid of the bin would also be controlled to open and close. Once the camera detects that the user wishes to dispose of trash, the camera system would wirelessly communicate with the bin to prompt the bin to move toward the user. Upon arriving at the user, the lid would open, ready to collect the user's trash. Once the user has disposed of trash into the bin, the lid would close, and the bin would return to its resting position. This solution simplifies the process of throwing trash by only requiring the users to call it and drop their trash into the bin.

1.3 Visual Aid

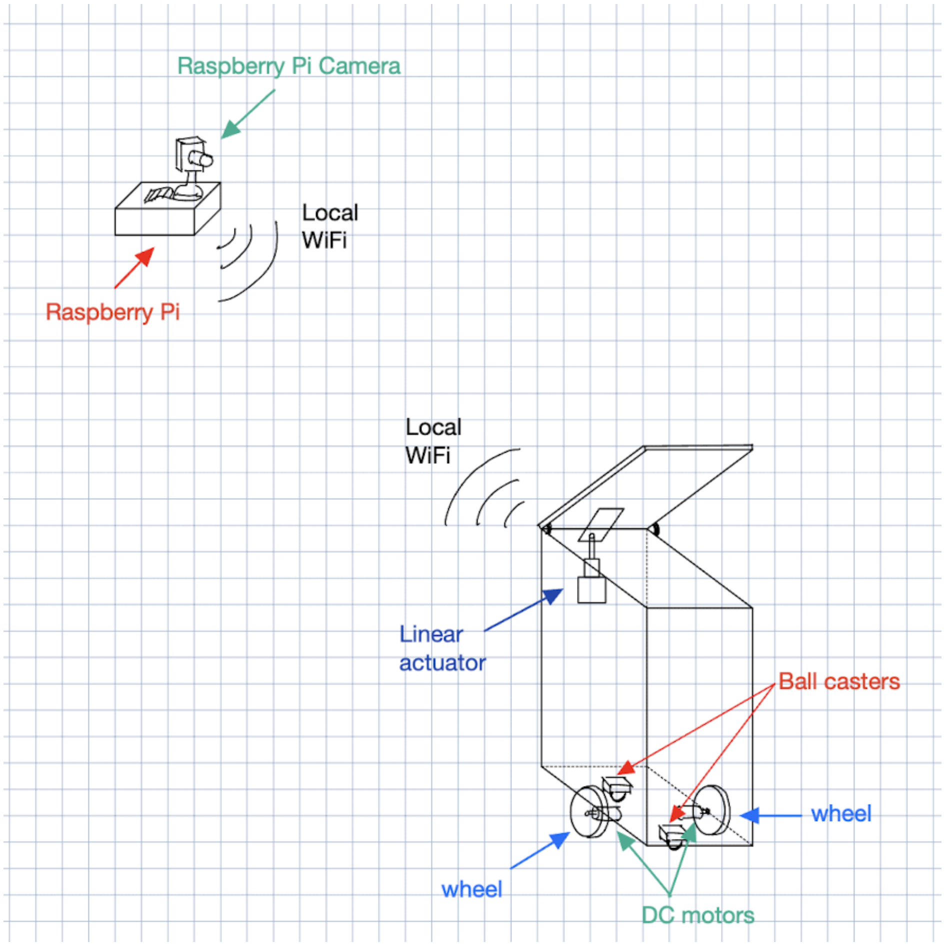

Figure 1: High Level Overview of the Ready-to-Serve Trash Bin

1.4 High-Level Requirements

- The camera system must be able to recognize a person's gesture for calling the trash bin. The camera will only recognize one single gesture, which will be similar to the one when someone is hailing a taxi.
- The trash bin must be able to travel close to the user and return back to its resting position. The distance the bin stops is 25 cm ± 5 cm, a side to side distance.
- The lid of the trash bin must open when collecting trash from the user and remain closed at all other times, including traveling to and from the user. The bin should react in 2 seconds before it begins to travel. And the speed of the bin should be in the range of 1-1.5 m/s. Once the bin reaches the destination, the lid should open in less than 1 second. The lid will remain open for 30 seconds before gradually closing in 1 second and then travel back to the bin station at the same speed as it comes.

2 Design

2.1 Block Diagram

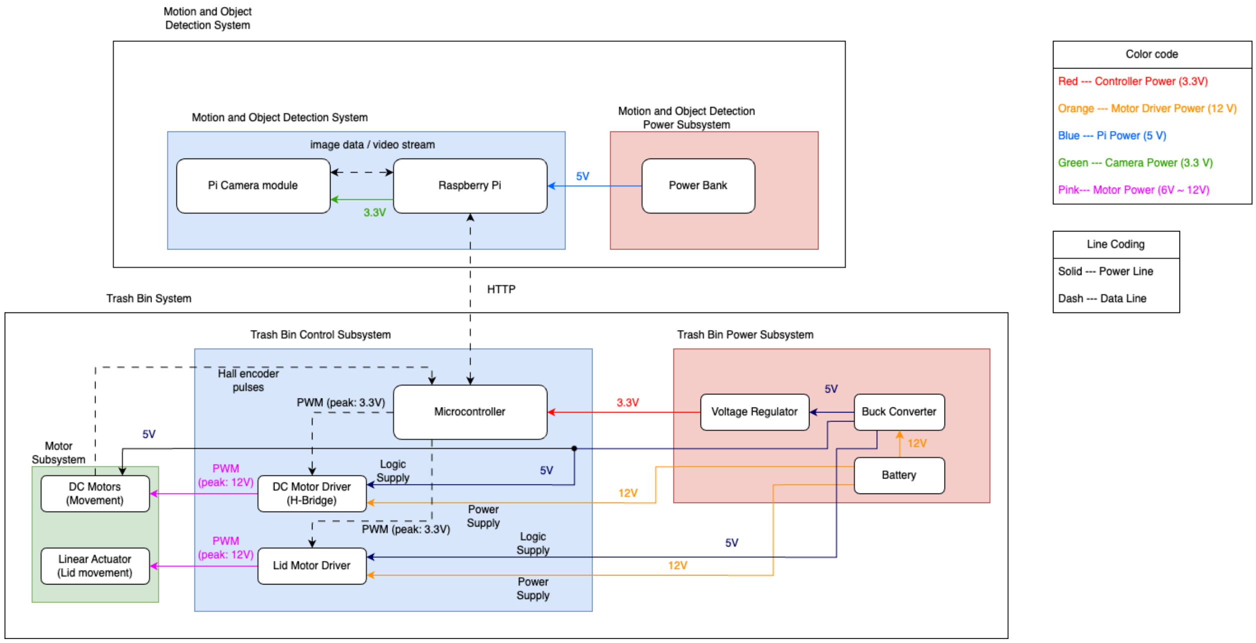

Figure 2: Block Diagram

2.2 Physical Design

The trash bin used in the project has a dimension (L × W × H) of 14.17 × 10.62 × 16.92 Inches (0.3599 × 0.2697 × 0.4298 m). The wheels used in the project have a radius of about 3 inches (0.0762 m). We asked the machine shop to build a chassis for the base (bottom) of the trash bin. The chassis holds a battery holder, PCB with MCU and motor drivers, two DC motors, two metal ball casters, and the trash bin. The DC motors are placed at the centers along the long side (L) of the trash bin. Both metal ball casters are placed in the center of the width (W). Each of the caster is placed on each edge of the long side. The linear actuator is placed vertically near the hinge of the trash lid; it is responsible for opening and closing the lid. The Raspberry Pi and the connected camera module are placed on the ceiling. It will be positioned to a point where the camera is able to capture the entire moving space.

2.3 Motion and Object Detection System Overview

2.3.1 Control System

This system is mainly responsible for recognizing users' hand gestures, in our case, the V-sign; if this gesture is detected, the system estimates the position of the user to find the shortest available path and leads the bin to that position. The system is built with two components: a Raspberry Pi 4 GB and a Smraza Raspberry Pi 4 Camera Module. The camera captures the real time image of the room and sends this as input to the CV model which is loaded into the Raspberry Pi, where the whole CV model is implemented using Convolutional Neural Network (CNN), a powerful tool in the field of computer vision which instead of checking if the model has seen the input before, computes how similar the image is to the labeled training data. Below is the basic finite state machine showing how the system works.

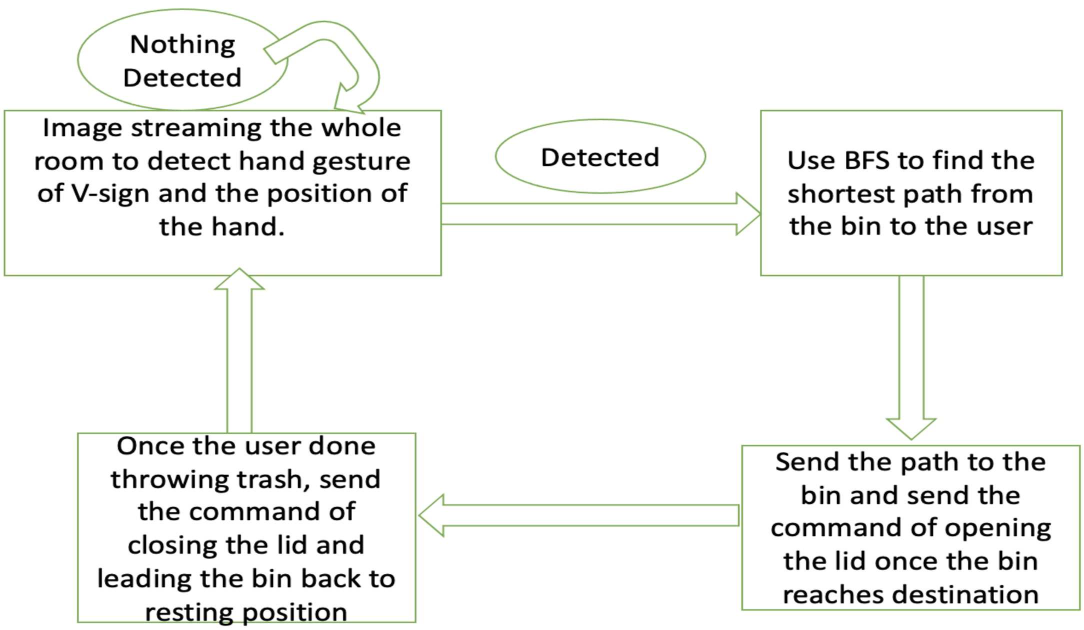

Figure 3: State Diagram for Motion and Object Detection System

A couple of changes we made to this system include the gesture to ask for the trash bin, the position of the camera, and the implementation of the model. The gesture that is used to notify the CV model is switched from a hailing-taxi gesture to a V-sign. This is because the CV model has some problems recognizing people's waving arms. Some reasons causing the issue can be due to the fact that the size, the skin-color, or even the cloth-color vary from person to person. If the model has not trained with a certain color before, it will have trouble recognizing people's arms when they are wearing a long-sleeve cloth with that color. Also, with the time and resource limitations, it's not possible for us to do such a large scale training. Instead of recognizing arms, we choose to focus on people's hands. The variation among people's hands is relatively smaller. When classifying users' hands given the image of the whole room, the task for the CV model is still complicated. This is due to the fact that users' hands are only a small part of the whole image, the CV model can focus on some other patterns of the room during the training and testing period. Whereas faces have high contrast patterns like nose, eyes, and mouth, the lack of such features in hands makes it comparatively difficult to detect them reliably from their visual features alone. Given the fact that the model will also need to classify the type of the hand gesture, this makes the task impossible for one model to accomplish. Thus, we decide to break this task down into two subparts, and have two models each being responsible for one task. The first model will find the position of the hand in the input image, crop that part so that the new image is one with most pixels portraying that hand, and send this to the second model, which is a CNN classifier on hand gestures. In our first model, we use a deep neural network called Single Shot Multibox Detector (SSD) [9].

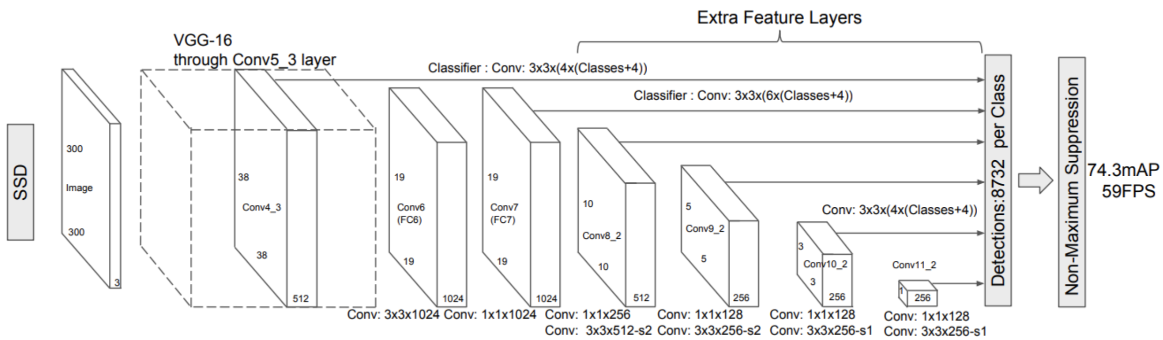

Figure 4: Layers of SSD Used For the First Model

The key idea of this is to combine predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. With SSD, regardless how far the object is from the camera or how small/big the object is in the image, the SSD method will always be able to recognize it. After getting the image of the hand, the second model, implemented using the following layer, will classify the type of the hand gestures.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Input((21 * 2, )),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(20, activation='relu'),
    tf.keras.layers.Dropout(0.4),
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(NUM_CLASSES, activation='softmax')
])
```

Figure 5: Layers Used For the Second Model

```
model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)
```

Figure 6: Function Used For the Second Model

If the model did not detect any V-sign, it will keep looping this process. Once it does, it will use a breadth-first search (BFS) algorithm to find the shortest path from the current position of the trash bin to the position of the user. This process is done by discretizing the room into many squares, with the size of 40 cm × 40 cm. We build the map representing the room treating each square as a vertex. The square where the trash bin position is at will be the starting vertex and the square where the user position is at will be the ending vertex. We construct an undirected edge between two squares if they are adjacent and neither of them is a wall or an obstacle. So in this case, we can avoid the BFS algorithm returning us an unavailable path.

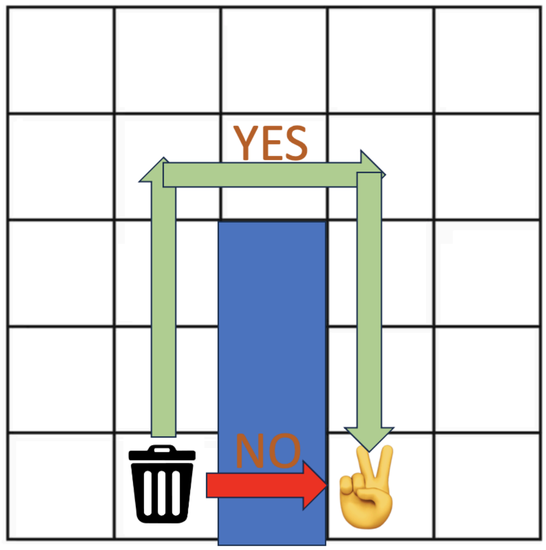

Figure 7: An Example Path

Once the path is found, the Raspberry Pi unit will send commands with HTTP requests to the trash bin control system via WiFi. The bin, after reaching the destination will report to Raspberry Pi, which will then send the command of opening the lid. After 30 seconds, Raspberry Pi will send another command to close the lid. Then, depending on whether the bin has a new destination to travel to, Raspberry Pi will either send the bin to the new location or back to resting points using the same BFS algorithm mentioned before. For each command Raspberry Pi sends to the bin, the Raspberry Pi unit waits until the trash bin responds (an HTTP response).

I also changed the position of where the camera will be installed in the room from the corner of to the center of the ceiling so that the view of the camera is now approximately perpendicular to the floor. We did this because having a perpendicular view makes the task of localization much easier. As the users of this trash bin are people with immobility, the z-coordinates of their hand when showing the V-sign to the camera are relatively low, making it possible for the camera to treat the 2-D graph it captures as a 3-D graph to find the position of the hand. This also makes the cv model more robust to light interference, as change in light intensity may influence the transpose[10] matrix if we are using view perspective transform.

In the validation period, our CV model reaches an accuracy of 97%, and the localization has an offset less than 5%. Both reach our estimation and justify the functionality of our motion and object detection system. The details of the validation will be discussed more in the verification section.

2.3.2 Power System

A power bank will power the Raspberry Pi 4B and the Raspberry Pi camera module. The Raspberry Pi 4B will receive power from its USB-C port and supply power to the camera module through its onboard camera module port.

2.4 Trash Bin System Overview

2.4.1 Control System

The control system of the trash bin system is responsible for receiving commands from the motion and object detection system and sending control signals to the motors. It also collects the feedback from the motors to better adjust the control. It is powered by the power system of the trash bin system. The MCU maintains feedback recollections and control calculation at a rate of 100 ± 10 Hz. The MCU distinguishes different commands and adjusts the speed, by controlling the duty cycle of the PWM signals to the motor drivers. The motor drivers amplify the incoming PWM signals based on their referenced voltages and output to the motors. The MCU we planned to use is ESP32-S3-WROOM-1-N16R8. This model has 16 MB on-chip flash and 8 MB PSRAM, and it helps provide more space for storing our program. ESP32-S3 has a built-in WiFi and Bluetooth module, introducing remote control and wireless communication [2]. The motor drivers controller we use is L298N. The L298N chip supports a large range of voltage source and current [3]. It is also easy to use. The schematic in Figure 10 is based on the module schematic from [4]. However, during debugging, we found out that the SENSE_B pin on the L298N module controlling the two DC motors has to be grounded, not floating, so the current through the motor is not inhibited.

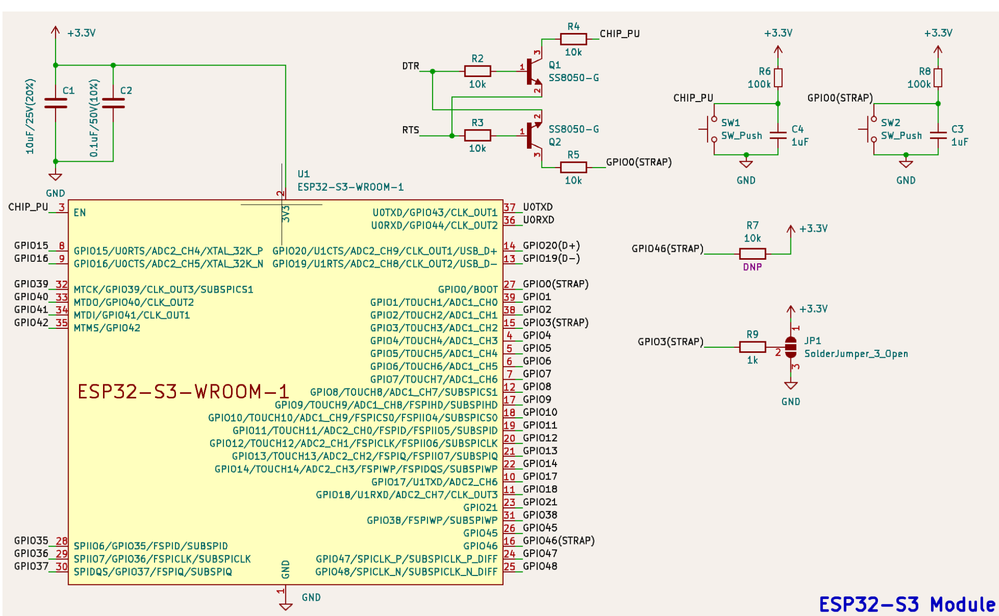

Figure 8: Microcontroller Schematic

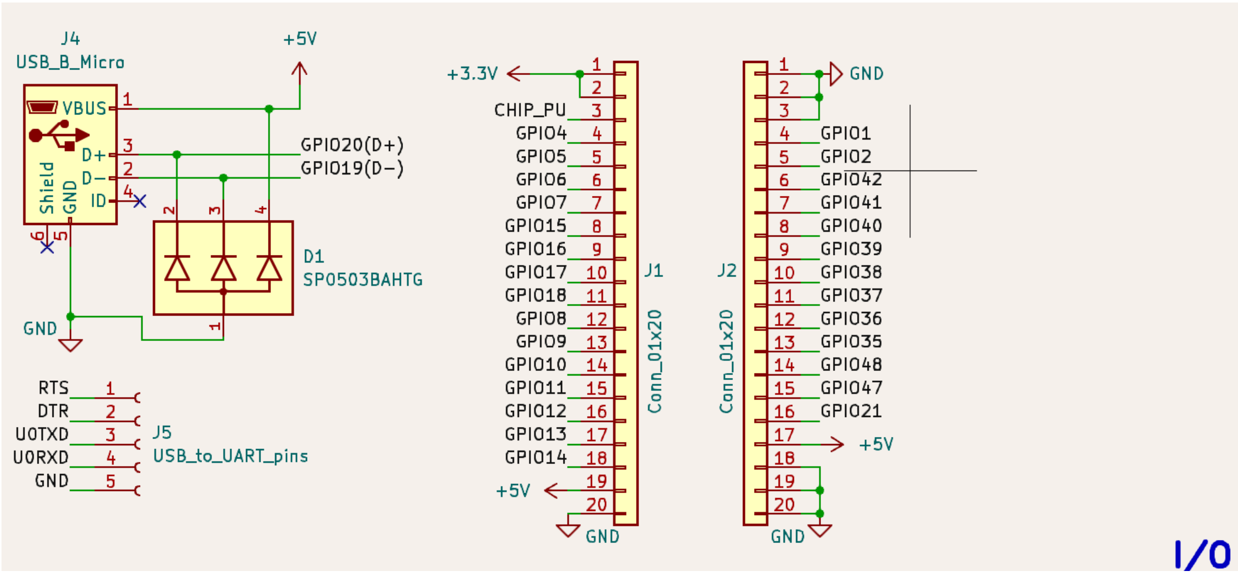

Figure 9: I/O Schematic

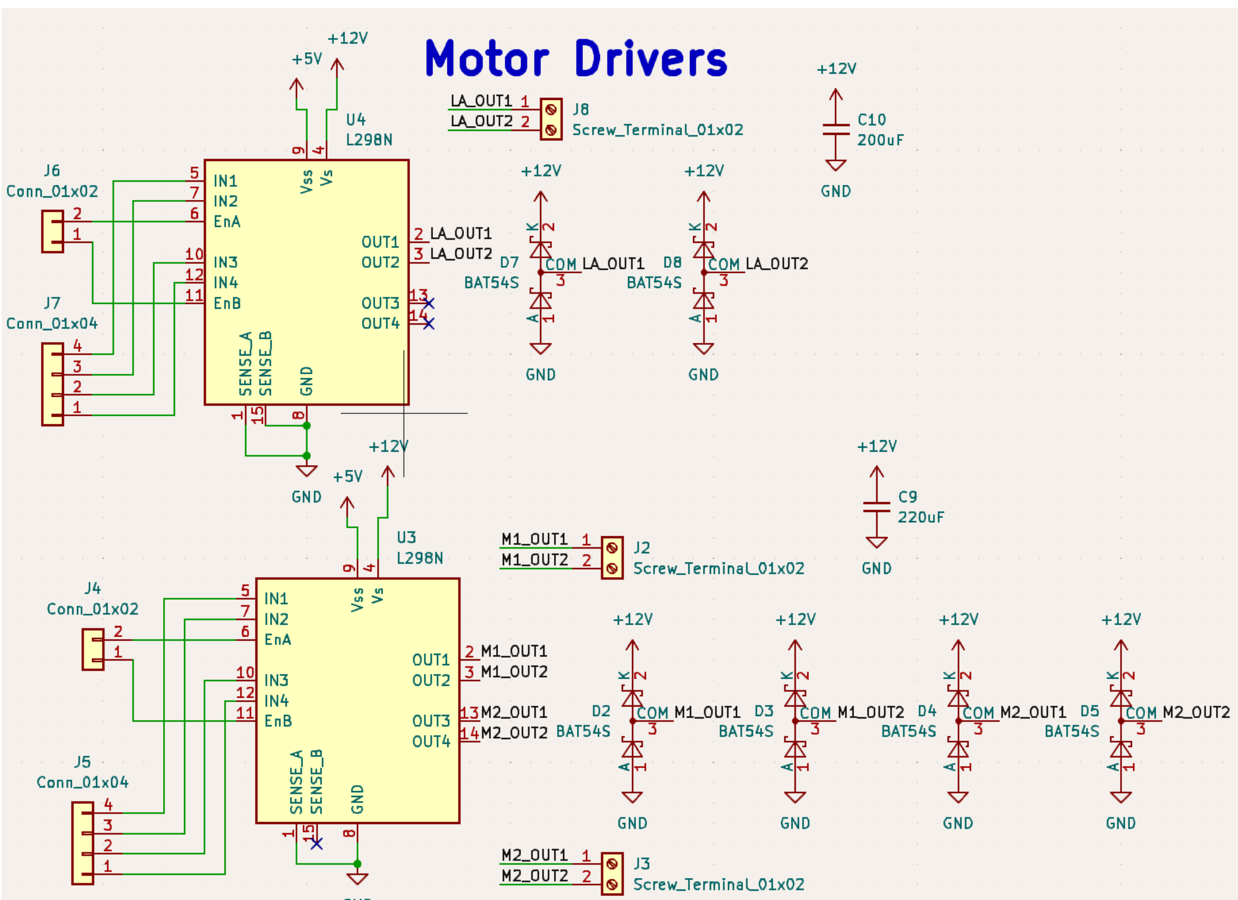

Figure 10: Motor Drivers Schematic

2.4.2 Power System

The power system of the trash bin system needs to provide the right voltage levels to the ESP32 microcontroller and the motor drivers. Our motors are rated for 12 V DC. Additionally, the motor driver runs on 5 V logic, whereas the microcontroller runs on 3.3 V logic. As a result, we will have to convert between different voltage levels. We decided that we will power the trash bin with a battery voltage around 12 V since we are using 12 V motors. In our final product, we used an 11.1-V LiPo battery as it was small and light enough to fit on the chassis and its capacity is sufficient for our needs. The battery output would directly connect to the motor drivers to provide power to the motors. Our microcontroller runs on 3.3 V and requires a current input of about 0.5 A [2]. If we perform this voltage conversion through a linear regulator, it would dissipate 4.35 W of power, which would be inefficient and significantly heat up the board. These concerns are discussed more in detail in Section 2.5 Tolerance Analysis. Because of the large voltage drop and the need for an intermediate voltage, we need to use a buck converter to first step down the voltage to 5 V. The TPS562201 buck converter can take in 11.1 V and output 5 V with a maximum output current of 2 A. The circuit in Figure 11 is based off of [5]. This 5-V output then connects to the motor drivers to supply the logic voltage. Finally, we connect the 5-V output from the buck converter to the voltage regulator to output a regulated 3.3 V to the microcontroller. We chose the TLV1117LV as it has a variation that can output 3.3 V at 1 A, which is sufficient for the microcontroller [6].

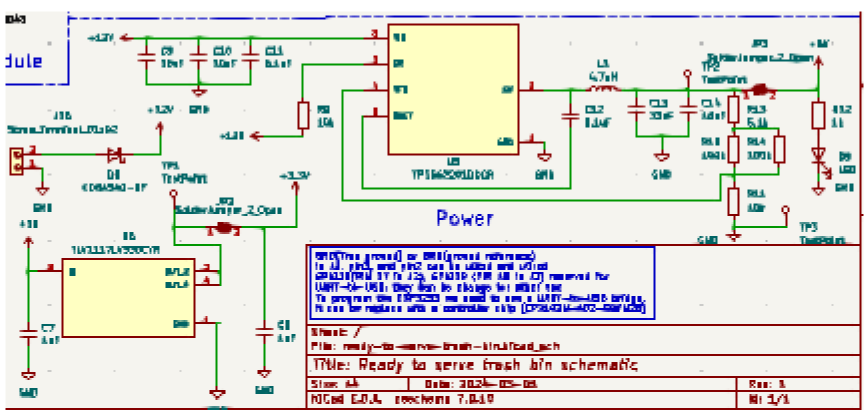

Figure 11: Power Schematic

2.4.3 Motor System

The motor system controls the movement of the trash bin. It is powered by the power subsystem and controlled by the control subsystem. The two DC-geared motors help the trash bin move around in the space (forward, backward, left turn, right turn). The linear actuator controls the opening and closing motion of the trash bin lid.

2.5 Tolerance Analysis

The diameter (d) of the wheel is about 6 inches (0.1524 m). When the motor is not loaded, the RPM of the motor is at least 150.

$$C = \pi d = 0.1524 * \pi = 0.47878\, m$$
$$Speed\ per\ second = RPM * C = 150 * 0.47878 / 60 = 1.19634\, m/s$$

With no load, the speed of the DC motor is 1.19634 m/s. It is close to the lower end of the average walking speed.

Torque and force analysis for the DC gear motors of overall design:

- τ_w = the torque of motor (Nm)
- R_w = the wheel radius (m)
- μ_w = static friction coefficient of the wheels against the floor
- μ_c = static friction coefficient of the metal caster against the floor
- m_T = total mass (kg)
- F_N = normal force (N)
- F_g = weight/gravitational force (N)
- F_{wf} = static friction of the wheel (N)
- F_{cf} = static friction of the metal caster (N)
- g = gravitational acceleration constant (m/s^2)

Two DC gear motors are at the same position, and they exert the same amount of torque. We can derive the following formula based the Newton's Second Law of rotational motion:

$$2\tau_w = F_{wf} * R_w \tag{1}$$

In rolling without slipping, the static friction of the wheels cannot exceed the static friction of the total mass. The static friction of the total mass is the normal force times the friction coefficients. Given that we have two different types of wheels (one is metal, and the other is plastic). Assuming the internal friction and friction from the bearing are negligible, we have:

$$F_{wf} \le F_N * (\mu_w + \mu_c) \tag{2}$$

Based on Newton's Second Law, $F_g = m_T * g$. Given that the entire trash bin is not accelerating up and down, $F_g = F_N$.

So the static friction of the total mass can be computed with the following formula:

$$m_T * g * (\mu_w + \mu_c) \tag{3}$$

Putting this into the inequality equation (2)

$$\frac{2\tau_w}{R_w} \le m_T * g * (\mu_w + \mu_c) \tag{4}$$

Rearranging (4), we can determine the required torque of the DC gear motors based on our target mass

$$\frac{2\tau_w}{R_w * g * (\mu_w + \mu_c)} \le m_T \tag{5}$$

The wheels have a radius of 3 inches (0.0762m). The wheels and metal caster have a friction coefficient of approximately 0.5 when rolling on the hospital flooring. We estimate that the summing mass of our finished product and its load is about 5 to 6 lbs (2.3 kg to 2.8 kg). With this information, in theory, we need both motors to provide a torque of at least 0.373 Nm.

Since we are using a voltage regulator to step down the battery voltage to a fixed voltage for the microcontroller, we have to verify that we will not draw too much power and risk overheating the component. The microcontroller needs about 0.5 A of current delivered from its power source, and it takes in a voltage of 3.3 V. Our battery voltage is 12 V to meet the needs of the motors. From these requirements, we can calculate the power that the regulator would have to dissipate:

$$P_D = i_{out}(v_{in} - v_{out}) \tag{6}$$

Using (6) with the parameters the regulator would have, we find that it would consume 4.35 W of power, which is a large amount for a small component. Knowing how much power the regulator would dissipate, we calculate the junction temperature to see how it would compare to the maximum junction temperature on the datasheet. For this analysis, we use [7] as a representative for typical voltage regulators we would consider.

$$T_j = P_D \Theta_{ja} + T_a \tag{7}$$

The junction-to-ambient thermal resistance is about 100 ℃/W. We use an ambient temperature of 38℃, which could happen if the circuit heats up. Therefore, the junction temperature, according to (7), will be 473℃, which is higher than the maximum temperature of the linear regulator. Therefore, we would need a buck converter to step down the voltage before applying it to the voltage regulator.

3 Verification

3.1 Motion and Object Detection System Overview

3.1.1 Control System

There are three requirements for the control subsystem. More details are shown in Appendix A.1.1.

The Raspberry Pi needs to provide power to the camera module. The voltage requirement for the camera module is about 3.3 V. We used a voltmeter to measure the voltage from the Pi camera port. The measurement ranged from 3 V to 3.3 V. This effectively fulfills our requirement for powering the camera module.

The round trip time (RTT) for wireless communication is important because the control is in real-time. We put the MCU in WiFi station mode and used the ping command on the Raspberry

Pi to inspect the RTT of each packet between the Pi and the MCU. The average RTT is within 500ms, which fulfills our requirement.

The Computer Vision system needs to correctly classify people's hand gestures so that it only sends commands to the trash bin when a V-sign is shown. It also needs to accurately estimate the position of the user so that it can lead the trash bin to the proper destination.
We collect 7000 data images of hands showing various gestures. To train the CV model, we split the data into 2 parts: 5200 images as training data and 1800 images as validation data. After fine tuning different hyperparameters, the best result we reached is about 97% accuracy on validation data. To further improve the accuracy, we set a counter initialized to 0 for each square in our code so that every time an image is sent from the camera, if a V-sign is detected in the certain square, we increment the counter for that square and clear all the other counters. We only find the path from the bin to the user and send it once a counter reaches 5. This can eliminate the probability of false alarm. As our accuracy is about 97%, it's not really possible that the CV model makes wrong judgments 5 times in a row. And since each iteration takes less than 0.3 seconds, users only need to hold their hand for less than 1.5 seconds before the CV starts to send the path to the bin. Even considering the fact that the model may miss the gesture, a 97% accuracy guarantee the miss probability is low and even if it occurs, the gesture needs to be held for no more than 3 seconds, which is a reasonable tolerance.
For the localization part, we did the testing in a room of size 3.6 m × 2.4 m. We did about 20 testing examples in different places in the room, and the localization algorithm returns the correct square representing the position we are at for all 20 examples. And the model starts to send commands to the trash bin after 1-2.5 seconds once the gesture is shown.

3.1.2 Power System

The power system has two requirements that we must verify. The voltage requirement for the Raspberry Pi is suggested to be around 5 V. We used a voltmeter to measure the voltage from the power port. The measurement ranged from 5 V to 5.1 V. This effectively fulfills our requirement for powering the Raspberry Pi. To verify the power adapter can constantly provide 15 W, we can use a multimeter to measure the voltage and current at the power port, and multiply these two numbers we get the power. The measurement ranged from 15 W to 15.3 W. This effectively fulfills our requirement. To verify the power adapter can support 5 V and 15 W for at least 3 hours, we just simply leave the device on for 4 hours on, and when we are back, the measurement stays the same compared to the number before we leave.

3.2 Trash Bin System

3.2.1 Control System

There are four requirements we needed to verify for this subsystem, which can be found in the requirements and verification table in Appendix A.2.2.

To ensure that the MCU is able to establish a persistent WiFi connection, we let the MCU connect to the WiFi in the local area network. In its main loop, we made the MCU periodically output the WiFi status via the serial port. We used COMTool, an open-source serial monitor software, to continuously listen to the MCU's serial port and log the output coming from the MCU [12]. We monitored the MCU's serial output for 3 hours, and the MCU didn't have any WiFi connection failure messages. Therefore, the MCU can maintain a persistent WiFi connection.

The MCU needs to be reliable in receiving commands from the Raspberry Pi. We wrote a Python script to simulate our project setup. The Python script periodically (every 4 seconds) sends requests to the MCU, and the MCU needs to log the request in one line in its serial port. We let the Python script and MCU run for one hour. Then, we compared the number of requests sent by the Python script and the number of outputting lines in COMTool [12]. The number of requests and outputting lines were equal. Therefore, the MCU can receive more than 90% of the commands, which effectively fulfill our requirements.

To ensure that the MCU can distinguish between different commands, we wrote a testing script that interacts with the serial port of the MCU. We sent the command and the PWM duty cycle (in integer) via serial port input, and the MCU could output different commands and call the helper function tailed to the commands. To verify the outputting PWM of the MCU, we connected the pins that output the PWM signals to an oscilloscope and inspected the duty cycle of the signals. We found out that the duty cycle of the outputting PWM signals was close to the PWM parameter in the command (with about 0.02 difference in the waveform). This effectively fulfilled our requirement.

The motors in the trash bin system need to perform various motions, so we must verify that the motor driver can turn the motors in different directions (clockwise / counterclockwise). We connected the motors to the motor driver and flashed the control script on the MCU. We probed the h-bridge control ports with an oscilloscope and verified that the MCU could provide logical high (~ 3.3V) and low (~ 0V). The motor driver uses two control inputs (IN1 & IN2 or IN3 & IN4) to manipulate the direction of the turn. We began with a logical low on both inputs and fed in a 50% duty cycle PWM to the enable port of the motor driver. Then, we set IN1 to be logically high. The motor turned in a counterclockwise direction. We measured the output of the motor driver on an oscilloscope and saw that the motor driver generated a PWM with the same duty cycle and a 12V peak-to-peak voltage. Flipping the control inputs (IN1 = logical low, IN2 = logical high), we could see the motor was turning in the opposite direction. By adjusting the control inputs, the linear actuator could extend or contract. Hence, the motor driver works as expected and fulfills our requirements.

3.2.2 Power System

The power system on the trash bin has three requirements that we must verify to ensure our PCB components are receiving the correct power for their operation. The specifics of these requirements can be found in the requirements and verification table in Appendix A.2.2.

To ensure the board is receiving power, we connect the power supply or battery to the connector on the board for power input and use a voltmeter to measure the voltage between the power source output node and the board ground. We consistently read the supply or battery voltage, verifying the first requirement.

The next requirement is that the output voltage of the Buck converter is 5 V ± 0.5 V, which directly connects to the logic reference voltage of the L298N drivers and also feeds into the LDO linear regulator. Using a voltmeter, we measured around 5.12 V, which satisfies this requirement.

The last requirement is for the output voltage of the LDO linear regulator is 3.3V ± 0.2V, which provides power to the ESP32 microcontroller. Using a voltmeter, we measured around 3.32 V. At this point, all requirements for the trash bin power system have been successfully verified.

3.2.3 Motor System

The motor system also has three requirements we need to verify. More details about the requirements are shown in Appendix A.2.3.

The DC motor works at the rated voltage (12 V) and voltages below the rated voltage. The voltage range we selected is from 6 V to 12 V. This range has a good coverage of different speeds of our motors. We connected the positive and negative terminals of the DC motor to the positive and negative terminals of the power supply and varied the voltage from 6 V to 12 V. As the voltage increased, the speed of the motor increased. To ensure that the DC motor could turn in the opposite direction, we switched the terminal connection of the DC motor and repeated the voltage variation. The motor could turn in the other direction and change the speed based on the different voltages. This effectively fulfilled our first requirements.

The linear actuator also has a rated voltage of 12 V. Similar to verifying the DC motor, we must ensure that the linear actuator can work from 6 V to 12 V. We connected the positive and negative terminals of the linear actuator to the positive and negative terminals of the power supply and varied the voltage from 6 V to 12 V. The higher the voltage, the faster the linear actuator extended. We switched the terminal connections and repeated the voltage variation to ensure the linear actuator ould contract. The linear actuator could contract and change the speed based on the different voltages. This effectively fulfilled our second requirement.

The hall encoder on the DC motor is essential for forming the close-loop control for our PID controller, so we have to ensure that the hall encoder can generate pulses from both phase ports, phase A and phase B. We powered the hall encoder with 5 V coming from the power supply. We connected phases A and B to channels one and two of the oscilloscope. Then, we spun the magnetic wheel on the motor and observed the square waveform coming from both phases. The hall encoder is able to generate pulses in a square wave, which fulfills the last requirement of the motor subsystem.

4 Cost and Schedule

4.1 Cost Analysis

Detailed cost analysis table for components in our project is shown in the Appendix B.
We assume the hourly salary is $40/hr and expect that we can have a salary of $40/hr × 2.5 × 65 hr = $6500 per person. There are three members in the team, so the total cost for labor is $6500 × 3 = $19500. Adding the costs of the parts, the total cost of our project is $166.66 + $19500 = $19666.66.

4.2 Schedule

Detailed schedule for all team members is shown in Appendix C.

5 Conclusion

5.1 Accomplishments

With some alternatives in the design, we successfully finished the project so that it works in a single-user daily-life environment. We have countless accomplishments in our project. The most difficult part is to implement the hand gesture detection model on the Raspberry Pi, and we have the model working on the Raspberry Pi. The Raspberry Pi can feed the image stream from the Pi camera in the hand gesture recognition detection model and make decisions based on output coming from the model. We have successfully implemented the wireless communication between the Raspberry Pi and the MCU. The Raspberry Pi can command the MCU by issuing HTTP requests and using web service APIs on the MCU. The MCU is able to recognize different commands, parse the data from the requests, and execute the commands properly. The PID controller in the MCU can work smoothly, making the speed control much easier. The final production of the project is successful. The Raspberry Pi is able to drive the trash bin from a resting position to the user's position and guide the trash bin back to the resting position after servicing the user. The trash bin is able to correctly interpret the commands from the Raspberry Pi and execute them.

5.2 Uncertainties

One uncertainty we faced was that the MCU would occasionally skip executing commands. The MCU could receive commands from the Raspberry Pi, but could not process them in a timely manner. This can be caused by processing delays in the service handler and state changes in the main loop. If commands are sent during these delays, the MCU has a chance to ignore them. One workaround is to have the Raspberry Pi delay sending commands by about 1 second. The solution is not ideal, but it makes the uncertainty disappear. There's also some uncertainties in the CV model as well, where it may have some miss and or false alarms. Even the probability of such errors are low, they may still exist in real life situations.

5.3 Ethical Considerations

To ensure a successful project, it is important that we follow the IEEE Code of Ethics. Our project has some safety and privacy concerns that we must address as per Section I.1 of the Code [8]. Since our product is targeted towards those with limited mobility, we need to take care that our product accommodates for their safety as well.

The power subsystems for both the Trash Bin System and the Motion and Object Detection Systems have different safety concerns that must be addressed. The trash bin would be powered by batteries, which could overheat, cause a fire, or create a shock. To minimize the chance of batteries overheating, we will ensure that the power needs of our motors, drivers, microcontroller, do not exceed what the batteries can provide. The Raspberry Pi and the camera would be using a plug to receive power from a 120 VAC outlet. Since electric shocks will be severe with this voltage, we need to warn the user of the dangers.

When the trash bin is moving, it could potentially collide with people and obstacles. Since the trash bin is only moving at walking speeds, which is about 1 m/s, and the trash bin will not be too heavy, the impact of a collision would be minor. Additionally, people would have ample time to react with the trash bin traveling at a low speed. When planning the path to destination, the bin will prefer to walk along the middle of the corridor, as people with immobility tend to walk against the wall, such path planning can reduce the possibility that the bin collides with them. Another concern related to movement is objects getting caught in the wheels and gear motors, which would damage both the drive system and object caught. While there is not much we can do on our end, we would have to warn the user of moving parts and rely on the user to keep the trash bin's path clear.

Since the product is designed to collect trash, various kinds of solid and liquid materials will enter our product, which could potentially create sparks and other hazards if they come in contact with the electronics. To mitigate this risk, the electronics would be housed underneath the trash bin where they will be separated from the trash inside the bin. During normal operation, the user should not be able to easily access the circuit. And it is the duty of the users to check regularly the situation of the battery to ensure the whole system is in good condition.

Since the lid of the bin is automatically controlled, the lid may jam people's fingers when they are throwing trash. To avoid this issue, we will extend the time of the lid remaining open to cooperate with the fact that people with limited mobility may need more time to finish throwing all their trash. Moreover, the lid will close gradually which will give people enough time to react and reduce the impact if a jam occurs. And in the worst case, since the bin is made with light-weight materials, a jam will not be severe.

A privacy concern must be addressed since we are using a camera in this project. The camera is only used to capture users' hand gestures and will have no local or online storage for these photos, which means all the information the camera has is for one-time use. The camera will not need to capture photos to improve the machine learning output, as the model will be trained with a huge data size which significantly outnumbered the number of pictures the camera captures in daily use, making the new learning progress have little effect. The camera will also not transmit photographic data to other devices. The only data that the camera system sends out is whether a gesture to call the trash can was made.

5.4 Future Work

To improve our design, we are considering various aspects. We can enhance the performance of wireless communication between both systems. HTTP/1.1 is not ideal for real-time communication. Thus, we should create a customized protocol using UDP with customized congestion and packet loss control to speed up wireless communication. For bin motion control, we can include an IMU in the design that tracks the heading angle of the bin and integrates that measurement into the PID controller. With this implementation, the trash bin can adjust its direction automatically when it heads in the wrong direction. We can also improve our software system so that it can serve different people at the same time. Now, the Raspberry Pi unit can only serve one person at a time, when it is sending commands to the bin, it will not retrieve images from the camera, meaning that if someone else is calling for the trash bin, this need will be ignored. This can be done by building a synchronization system on the Raspberry Pi. Also, the CV model now can only handle images with one hand, if multiple hands are shown, the model gets confused and will randomly focus on one hand and ignore the others. With these improvements, the bin may have a wider range of use, for example placed in hospitals for patients.

6 Citations

[1] “Disability Impacts All of Us.” Centers for Disease Control and Prevention. https://www.cdc.gov/ncbddd/disabilityandhealth/infographic-disability-impacts-all.html. (accessed February 6, 2024).

[2] *2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth® 5 (LE) modul*e, Espressif Systems, 2023, v1.3. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf.

[3] *Dual full-bridge driver*, STMicroelectronics, 2023, Rev. 5. [Online]. Available: https://www.st.com/content/ccc/resource/technical/document/datasheet/82/cc/3f/39/0a/29/4d/f0/CD00000240.pdf/files/CD00000240.pdf/jcr:content/translations/en.CD00000240.pdf.

[4] *L298N Dual H-Bridge Motor Driver*, Handson Technology. [Online]. Available: https://www.handsontec.com/dataspecs/module/L298N%20Motor%20Driver.pdf

[5] *TPS56220x 4.5-V to 17-V Input, 2-A Synchronous Step-Down Voltage Regulator in 6-Pin SOT-23*, Texas Instruments, 2020, Rev. B. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps562201.pdf.

[6] *TLV1117LV 1-A, Positive Fixed-Voltage, Low-Dropout Regulator*, Texas Instruments, 2023, Rev. C. [Online]. Available: https://www.ti.com/lit/ds/symlink/tlv1117lv.pdf

[7] *Low Dropout Linear Regulator*, Diodes Incorporated, 2022, Rev. 5-2. [Online]. Available: https://www.diodes.com/assets/Datasheets/AZ1117C.pdf

[8] “IEEE Code of Ethics.” IEEE. https://www.ieee.org/about/corporate/governance/p7-8.html. (accessed February 7, 2024).

[9] “Hand Landmarks Detection Guide.” MediaPipe. https://developers.google.com/mediapipe/solutions/vision/hand_landmarker

[10] “Geometric Image Transformations.” OpenCV. https://docs.opencv.org/4.x/da/d54/group__imgproc__transform.html#ga20f62aa3235d869c9956436c870 893ae

[11] “Hand Gesture Recognition Mediapipe”. Github. https://github.com/kinivi/hand-gesture-recognition-mediapipe/tree/main

[12] Neucrack, “Neutree/COMTool,” GitHub, https://github.com/Neutree/COMTool

Appendix A Requirements and Verification Table

A.1 Motion and Object Detection System Overview

A.1.1 Control System

Table 1: Motion and Object Detection Control System Requirements and Verification

Requirement	Verification
The Raspberry Pi needs to continuously power the Raspberry Pi camera module with a voltage of 3.3V ± 0.2V	Power on the Raspberry Pi. Look for the Pi camera connector (a rectangular 15-pin connector near the 3.5mm audio jack with the label “camera”) Turn on a multimeter. Put the negative probe on PIN 1 of the connector and the positive probe on PIN 15. Then, check the voltage readings from the multimeter and see whether it is within the range of 3.3V ± 0.2V. Note: PIN 1 is the left furthest pin from the label “camera” and PIN 15 is the right furthest pin from the label “camera”.
The RTT (Round Trip Time) of the communication should be no more than 700ms.	Power on the MCU and flash the code to the MCU. The code will output the IP address of the MCU on the serial monitor on the Arduino IDE. Power on the Raspberry Pi and open the terminal on Pi. Type **ping ip_address_of_mcu -c <n>** in the terminal, where n is specified as the number of packets. By default, **n = 20**. The **time** section of the ping output is the RTT of the communication between Pi and MCU. Compare the output and see whether it is within the limitation.
The CV system should send commands to the trash bin only when the V sign is detected. The CV system should also be able to localize the position of the hand.	Test with multiple hand gestures at different positions. The CV unit should correctly distinguish the V sign. Trash bin should only move when showing a V sign to the camera. The stop position of the bin should be in a 25 cm ± 5 cm range from the hand.

A.1.2 Power System

Table 2: Motion and Object Detection Power System Requirements and Verification

Requirement	Verification
The power bank should continuously provide at least 5V for at least 3 hours	Connect the USB-C output to a USB-C power tester and connect the USB-C tester to the Raspberry Pi. Then, verify the voltage reading and power reading.
The power bank should continuously supply at least 15 W for at least 3 hours	Connect the USB-C output to a USB-C power tester and connect the USB-C tester to the Raspberry Pi. Then, verify the voltage reading and power reading.

A.2 Trash Bin System Overview

A.2.1 Control System

Table 3: Trash Bin Control System Requirements and Verification

Requirement	Verification
The MCU needs to ensure a persistent WiFi connection	Connect the USB of the MCU board to a laptop. Write and upload a program that continuously verifies the WiFi connection every 1 to 2 seconds using Arduino IDE. The program writes messages of successful connection and connection failure to the Arduino serial monitor. Use a COM tool that records the output of the serial port that connects to the MCU and outputs a log file once execution finishes. Run the MCU program simultaneously for at least 30 minutes. Once the MCU program finishes running, check the log and see if the log contains any message regarding connection failure. To test the stationary and moving condition, we will verify to see if the connection is stable under the following scenarios: - Some obstacles or people are around the bin.(stationary)

	- The bin is moving towards or away from the router. (moving)
The MCU needs to receive at least 90% of the requests (commands) coming from the Motion and Object Detection System.	Connect the USB of the MCU board to a laptop. Write a Python script that periodically sends different requests (commands) to the MCU. By default the testing period would be every 2 ± seconds. Put a serial print statement in the command handler code, and upload the code to the MCU Use a COM tool that records the output of the serial port that connects to the MCU and outputs a log file once execution finishes. Let the MCU and Python script run for at least 30 minutes. Once the Python script finishes running, check the log and see if total commands received is at least 90% of total commands sent.
The MCU needs to distinguish different commands, set the correct duty cycles of the PWM signals, and output the PWM signals to the motor drivers.	Connect the PWM output pins of the MCU to an oscilloscope and the USB port of the MCU to the laptop. Writing a testing function in Arduino IDE. The function takes in inputs from the serial port and change commands (to simplified the testing, each command has an integer ID) Check the waveform on the oscilloscope and see if the PWM cycle is adjusted accordingly.
The L298 chip can switch the turning direction of DC motors and the linear actuator.	Supply 12 V to the VS (supply voltage) port and 5 V to the VSS (logic supply voltage) port. Ground all the CURRENT SENSING ports. Connect the OUTPUT1 and OUTPUT2 pins to the positive and negative ends of a DC motor / linear actuator. Connect ENABLE A to 3.3V. Power IN1 with a logic high (3.3V) and logic

	low (eg. GND). Observe the turning direction of the motor / linear actuator for 5 seconds. The motor should turn in one direction (clockwise / counterclockwise). The linear actuator should either push or pull back. Then, flip the inputs to IN1 and IN2. Observe the turning direction of the motor / linear actuator for 5 seconds. The motor / linear actuator should turn in the opposite direction. Repeat the process for OUTPUT3, OUTPUT4, IN3, IN4, ENABLE B. Note: ENABLE B, IN 3 (h-bridge control 1), IN 4 (h-bridge control 2), OUTPUT 3 (motor positive end), and OUTPUT 4 (motor negative end)

A.2.2 Power System

Table 4: Trash Bin Power System Requirements and Verification

Requirement	Verification
The power system needs to provide a voltage with a range from 6V to 12V to the DC motor and a voltage with a range from 6V to 12V to the linear actuator.	Using a digital multimeter, measure the output voltage of the battery.
The buck converter of the power subsystem should supply a voltage of 5 V ± 0.5 V.	Using a digital multimeter, measure the output voltage of the converter.
The voltage regulators of the power subsystem should supply a voltage of 3.3V ± 0.2V.	Using a digital multimeter, measure the output voltage of the LDO.

A.2.3 Motor System

Table 5: Trash Bin Motor System Requirements and Verification

Requirement	Verification
The DC motor can work in the range of 6V and 12 V	Connect the positive and the negative ends of the motor to the voltage source and

	multimeter. Set the voltage source to 6V, check the output from the multimeter, and see if the motor is turning Repeat this process for 7 to 12V voltage source
The linear motor can work in the range of 6V to 12 V	Connect the positive and the negative ends of the motor to the voltage source and multimeter. Set the voltage source to 6V, check the output from the multimeter, and see if the motor is turning Repeat this process for 7 to 12V voltage source
The hall encoders on the DC motors can output square waves.	Connect encoder phase A output to channel 1 of an oscilloscope and encoder phase B to channel 2. Supply 5V to the hall encoder logic Spin the magnetic wheels at the end of the DC motors and check the waveform on the oscilloscope.

Appendix B Cost Analysis Table

Table 6: Cost Analysis Table

Description	Manufacture	Part Number	Quantity	Cost
Raspberry Pi Model 4B	Raspberry Pi	SC0193	1	$61.79
Raspberry Pi Camera	Smraza	SM39	1	$16.99
Raspberry Pi 4 Power Adapter	iRasptek	MKE-0513500H	1	$9.99
ESP32-S3-WROOM-1-N16R8	Espressif Systems	ESP32-S3-WROOM-1-N16R8	1	$3.90
Trash Bin	Mainstays	MS202376BLK	1	$10.62
Motor Driver Chip	STMicroelectronics	L298N	2	$11.20
3.3-V LDO Regulator	Texas Instruments	TLV1117LV	1	$0.35
2-A Synchronous Buck Converter	Texas Instruments	TPS562201	1	$0.44
4.7 µH Fixed Inductor	Vishay Dale	IHLP2525CZER4R7M01	1	$1.15
12 V DC Motor	Weiheng Transmission Technology	JGB37-520	2	$5.75
12 V Linear Actuator	Rtisgunpro	MN	1	$19.99
Metal Ball Caster	DFRobot	FIT0007	2	$2.50
Push button	KMR231NG ULC LFS	C&K	2	$0.55
TVS DIODE	SP0503BAHTG	Littelfuse Inc.	1	$0.86
NPN Transistor	SS8050-G	Comchip Technology	2	$0.29
Total				$166.66

Appendix C Schedule

Table 3: Weekly Team Member Schedule

Week	Task	Person
2/26 - 3/1	Design Review	All
	Design PCB (ESP32S3 schematic)	Dongming
	Design CV system	Owen
	Design PCB (Power system and Motor Drivers)	Josh
3/4 - 3/8	First Round PCBway Order, Teamwork Evaluation I	All
	Writing MCU code (motor controlling)	Dongming
	Design and Train the CV system	Owen
	Finish initial PCB design, Assist with MCU motor code	Josh
3/11 - 3/15	Spring Break	All
3/18 - 3/22	Initial testing	All
	Writing MCU code (motor controlling)	Dongming
	Train and test the CV system	Owen
	Test Hardware, PCB Revisions, Assist with MCU motor code	Josh
3/25 - 3/29	Individual Progress Reports, Second PCBway Order	All
	Writing MCU code (communication) and integrating motor code	Dongming
	Test on camera video streaming and signal input/output, design algorithms of finding the path to distance	Owen
	Finish PCB Revisions, Assist with testing	Josh
4/1 - 4/5	Begin integration and tests	All
	Writing MCU code (communication) and integrating motor code	Dongming
	Test on sending output command	Owen
	Start Board Integration Tests	Josh
4/8 - 4/12	Final Integration Tests, Finalize Assembly	All

	Overall testing for MCU controls and MCU communication	Dongming
	Overall testing	Owen
	Overall testing for hardware	Josh
4/15 - 4/19	Mock Demo, Team Contract Fulfillment, Final Debugging	All
4/22 - 4/26	Final Demo, Mock Presentation	All
4/29 - 5/3	Final Presentation, Final Paper Due, Lab Notebooks Due	All

Appendix D PCB Layout

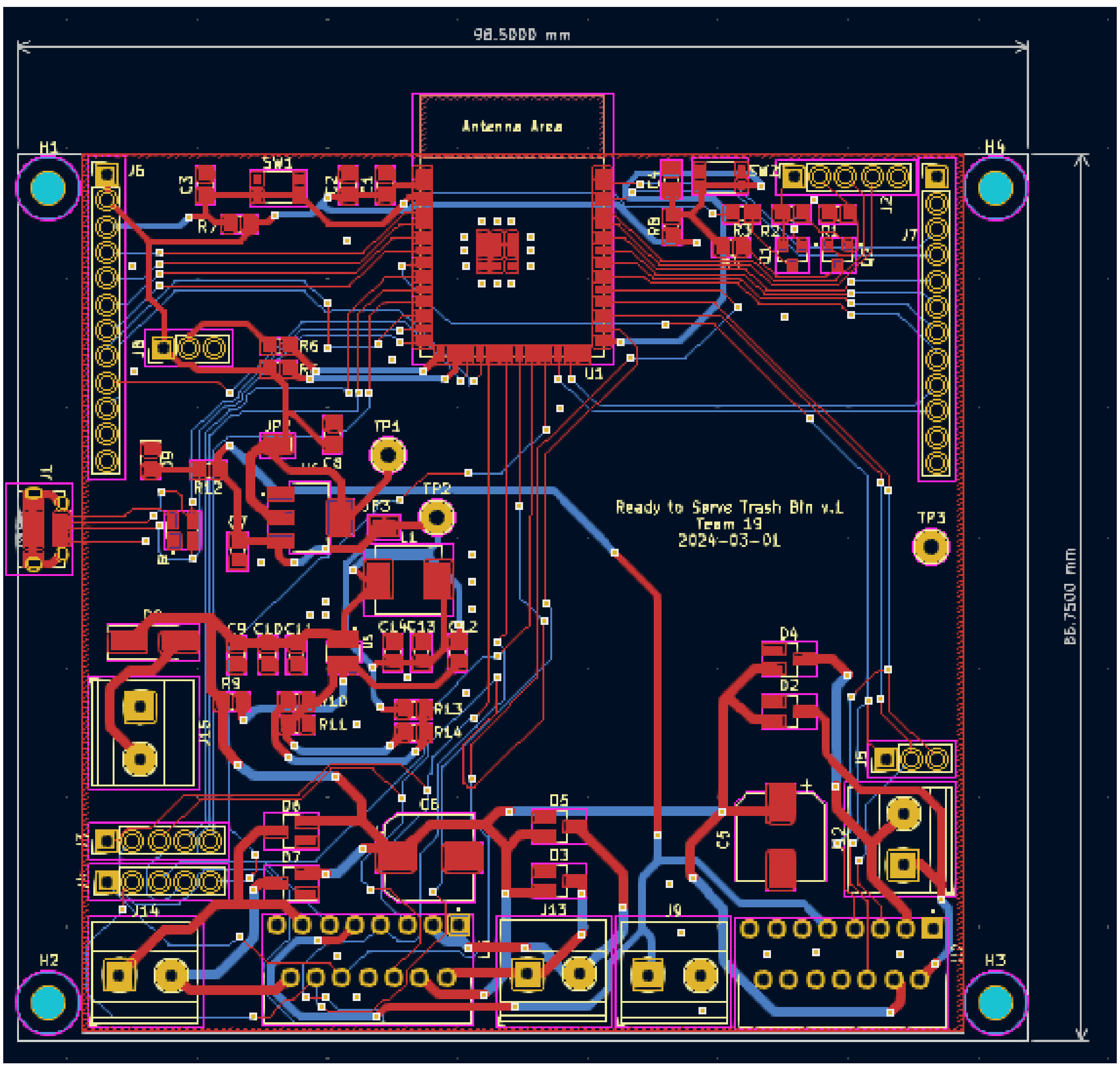

Figure 12: PCB Design