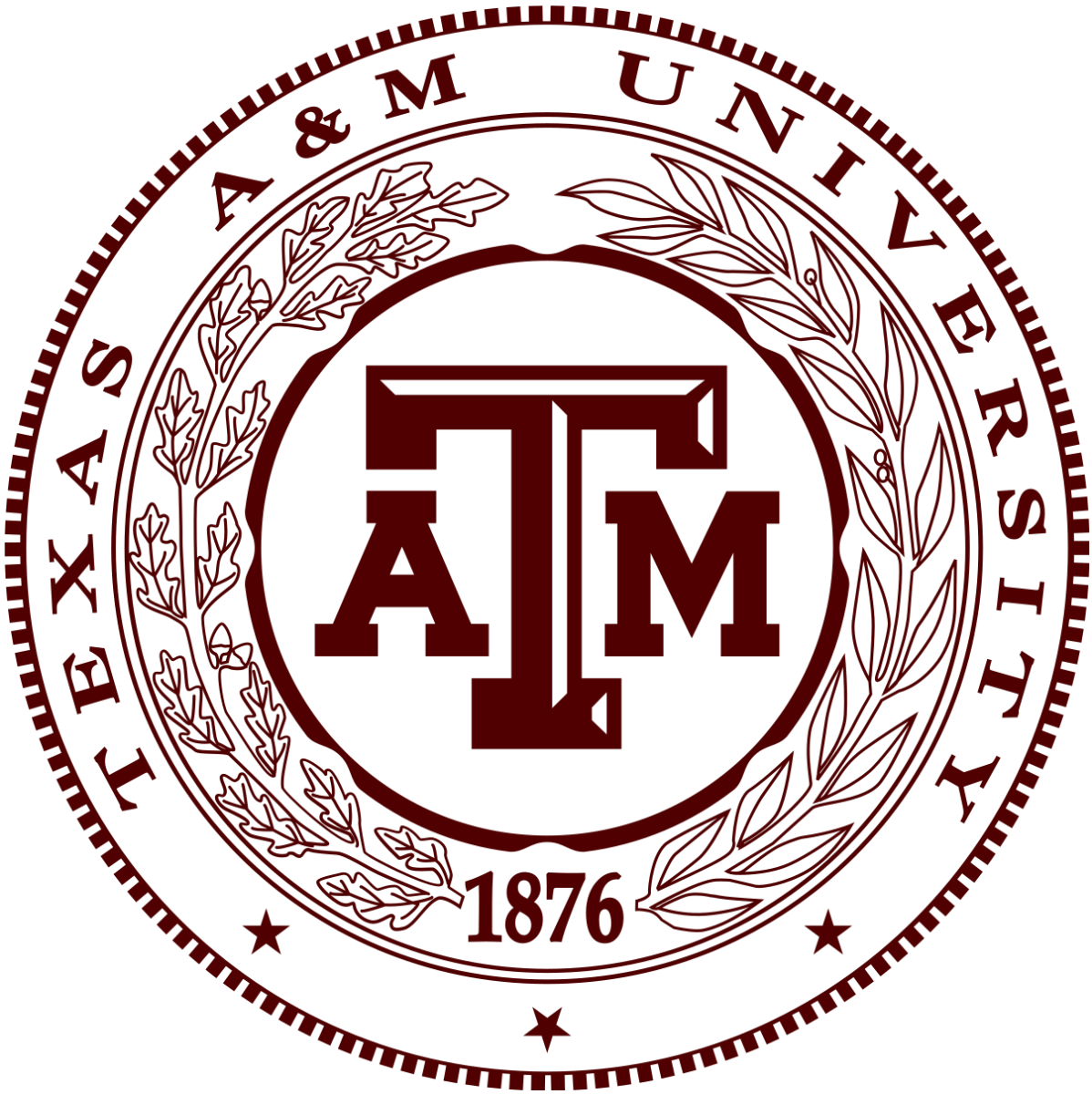


Project: Human Radar

CSCE 462 Professor: Dr. Jyh Liu

Dhiraj Rishi Atmakuri - rishi.atmakuri@tamu.edu

Alex Do - alexthanhdo2003@tamu.edu

Arun Natarajan - arun.annamalai4@tamu.edu

Table of Contents

Executive Summary

The **Human Radar System** is an innovative robotic system designed to provide safety and support for elderly individuals living alone. Built to address the challenges of monitoring and responding to emergencies, this compact and intelligent device ensures peace of mind for families and caregivers alike.

The system features a robotic car equipped with a high-definition camera, night vision capability, LiDAR sensors, and GPS, all seamlessly powered by a Raspberry Pi 5. Its primary function is to autonomously follow the user, monitor their movements, and detect falls in real-time. Upon detecting a fall, it immediately sends an alert to a connected smartphone, enabling swift assistance when it's needed most.

This practical and straightforward device offers an effective solution for elder care. With the **Human Radar System**, safety is always within reach for caretakers, no matter where they might be.

Introduction

The **Human Radar System** project was inspired by the real-life challenges faced by elderly individuals, particularly those with conditions such as dementia or limited mobility. One of our team members shared the story of their grandmother, who often gets lost due to dementia and struggles with balance, increasing her risk of falls. Recognizing the critical need for a reliable safety system, we set out to develop a solution that could address these concerns while offering independence and security to users.

What began as a simple camera-based monitoring system evolved into a comprehensive robotic solution. The system now includes GPS tracking for real-time location monitoring, autonomous user tracking, fall detection, and smartphone alerts. These enhancements not only ensure safety but also provide peace of mind for caregivers by enabling swift responses to emergencies.

Designed for practicality and versatility, the **Human Radar System** operates seamlessly in various home environments and is easy to set up and transport. This project represents a meaningful response to a growing societal need, combining technical innovation with a commitment to improving lives.

Brainstorming

The group started with several ideas. Some examples were:

- Poker Game Dealer
    - A box that would automatically deal poker cards and chips to all the players sitting around the box.
- Surgical Robot Arm
    - Surgical robotic arm - A robotic arm would be controlled via a controller and capable of rotating along the positive x, y, and z axes. It would also be able to grip objects, such as a pencil, and perform actions like writing. The concept highlighted the potential for applications in education, industry, and medical fields.
- **Camera Security System**
    - A device equipped with all sorts of cameras and sensors capable of acting like a sentry system when nobody was home.
- **Humar Radar System**
    - A robot that is equipped with a camera, night vision, and sensors so it's able to follow a person and update their location and also capable of notifying if they fall unconscious.

At the start of the semester, our group explored several ideas, eventually choosing the **Camera Security System** as our initial project. This device was designed to act as a sentry system, equipped with cameras and sensors to monitor homes and provide real-time alerts for unusual activity. It aligned with our interests and seemed achievable within the project timeline.

However, as the semester progressed, discussions with our professor and personal experiences led us to pivot toward a more impactful goal. We expanded the concept into what became the **Human Radar System**, a solution designed for elder care. By integrating GPS tracking, autonomous user following, fall detection, and smartphone alerts, the system evolved into a comprehensive tool for enhancing the safety and independence of elderly users.

This shift presented new challenges but ultimately gave our project a meaningful purpose and greater societal value.

Inspiration and Background

Our initial project idea, the **Camera Security System**, was chosen because it seemed the most plausible to complete within the given timeframe. The concept involved creating a sentry system equipped with cameras and sensors to monitor homes and provide real-time alerts. While feasible, we soon realized that the project lacked complexity and didn't fully challenge our skills. This realization prompted us to reevaluate and ultimately expand the scope of the project.

The shift to the **Human Radar System** was driven by both external input and personal experiences within the team. Our professor suggested focusing on elder care, inspired by his mother's recent hospitalization. This idea resonated with the team, particularly because one of our teammates shared a personal story about their grandmother, who has dementia and struggles with mobility. This personal connection became the catalyst for transforming our project into a meaningful solution for elderly individuals.

Although the scope of the project expanded significantly, key components of the original **Camera Security System** carried over into the final design. The entire camera system, including its mount and functionality, was repurposed for the **Human Radar System**. This allowed us to build on our existing work while integrating additional features, such as GPS tracking, fall detection, and autonomous user following.

The decision to pivot wasn't due to technical limitations but rather a desire to take on a more challenging and impactful project. The expanded scope of the **Human Radar System** pushed us out of our comfort zone, requiring us to combine our electrical engineering and computer science skills to develop a sophisticated robotic system that has practical use in the real-world.

As a team of computer engineers, we were well-equipped to tackle these challenges. Each member contributed their expertise, whether in coding with Python and C++ or working with electrical components like chips and motors. This collaborative effort allowed us to refine and execute the concept, creating a system that was not only technically complex but also deeply purposeful.

Challenges

Throughout the development of the **Human Radar System**, we encountered several technical and design challenges. These obstacles required creative problem-solving, collaboration, and learning new skills to overcome. Below is a summary of the key challenges and the solutions.

Hardware Challenges:

1. **Motor and Body Integration**

   Challenge:
   The salvaged motor and body frame provided a good starting point for the robot, but ensuring reliable motor functionality was a challenge due to a non-functional DRV 8871 motor driver from the salvaged project.

   Solution:
   We replaced the faulty DRV 8871 with an L298N motor driver compatible with our Raspberry Pi 5 and 12V DC motor, enabling smooth and reversible motor operation. The diagnosis process involved: verifying motor functionality by direct 12V power application, using a multimeter to confirm the motor driver wasn't outputting voltage, and researching embedded circuit debugging techniques to properly diagnose electronic components.

2. **Modifying the Salvaged Body**

   Challenge:
   The salvaged body frame had insufficient height for the components (2x battery packs, perfboard, etc.) we wanted inside the car, and it lacked the turning capability necessary for navigation.

   Solution:
   We modified the body frame by purchasing longer screw spacers to increase internal clearance. Afterward, we purchased a small caster wheel and super glued the bearings of the caster wheel in order to nullify its swiveling, which would let it be static and turn in coordination with the servo. We then brainstormed and designed a custom 3D-printed adapter via Fusion360 to connect an SG5010’s servo horn and the caster wheel, enabling precise directional control.

3. **Servo Jitter**

Challenge:
When the SG90 Servos were connected to the Raspberry Pi 5's 5V and GND, it was extremely jittery, which caused significant issues when we were looking into implementing our person detection algorithms. This posed an issue we needed the cameras to be still in order for any algorithm to work.

Solution:
We supplied the PWM controller with its own 5V power source since the Raspberry Pi had limited current capacity which lead to malfunctions like jittering. Upon purchasing the PCA9686 PWM controller and connecting it with a 2nd 5V battery pack for its separate power supply, the jitter problems were eliminated and we were able to continue implementing the algorithms.

4. **Battery Pack Issues**

Challenge:
The system used two battery packs: one to power the Raspberry Pi, which worked fine, and another for the PWM controller. The battery pack experienced issues as it required all outlet ports to be connected for sufficient voltage output, which was an unusual problem.

Solution:
We discovered that many modern power banks need to go through power negotiation through the USB-C port before enabling power output through USB-A, so the PWM controller never gets voltage with just the USB-A connected. We adapted our power connections to ensure that the USB-C port was utilized using an SSD we had lying around, stabilizing the voltage output to resolve the power issue. On a side note, the battery pack powering the Raspberry Pi did not supply the necessary 5V voltage to the USB-A ports, so we couldn't use the first battery pack to power the PWM controller.

5. **CAD and 3D Printing**

   Challenge:
   Designing and printing the camera housing was particularly challenging as none of the team members had much experience with CAD design or 3D printing.

   Solution:
   Using a friend's 3D printer, we learned the basics of 3D modeling and printing over the course of a week by utilizing the Fusion360 CAD software, which is given to TAMU students for free. There were plenty of tutorials and documentation to learn from in order to make good designs. After iterative testing and practice, we successfully created a functional housing for the camera system and an adapter for the servo-to-caster wheel.

6. **Soldering Components**

   Challenge:
   None of the team members had prior experience with soldering, and it was a critical step in assembling the system.

   Solution:
   There were plenty of tutorials on how to solder safely and effectively, so we utilized the labs to practice soldering components. Within a few days, we were able to solder all necessary components successfully. This also became an exciting skill to acquire during the project.

Software Challenges:

1. **GPS Issues**

   Challenge:
   The initial GPS module was outdated, relying on a 2G signal that was no longer practical for real-time tracking.

   Solution:
   We shifted to a solution using IP and SMS messaging through APIs and a compatible library. This modernized approach provided reliable location tracking and communication.

2. **Fall Detection Algorithm**

   Challenge:
   Ground-level camera perspective required a novel approach to fall detection, as traditional wall-mounted camera algorithms weren't applicable.

   Solution:
   We established a zero-point origin at ground level. The system flagged a fall if the detected user’s position dropped below this threshold for a specified duration, improving reliability and reducing false positives.

3. **Person-Tracking Algorithm**

   Challenge:
   Developing the algorithm for the robot to follow a person was extremely complex, particularly when integrating it with other parts of the system. While individual code modules worked, combining them caused errors and required extensive debugging. Additionally, our chosen algorithm of Mediapipe also required extensive calibration to prevent false positives.

   Solution:
   Through iterative testing and debugging, we resolved conflicts between modules, ensuring seamless integration of the person-tracking algorithm with the overall system.

4. **Low Light Performance**

   Challenge:
   The system needed to operate in varying lighting conditions, but initial implementation struggled with low light and overexposed scenarios.

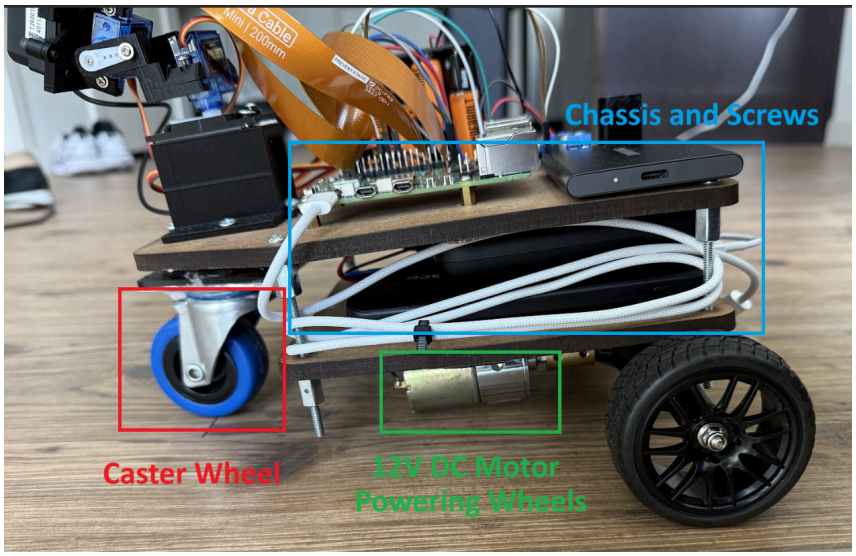
   Solution:
   Through post-processing methodologies, we added methods like CLAHE (Contrast Limited Adaptive Histogram Equalization) and gamma correction as a software approach to adjust for low-light scenarios in order to allow media pipe to identify a pose more easily.

Hardware

We used a variety of electronic hardware components including:

Chassis and Structural Parts:

- Car Chassis
- 4x Long Screws
- 1x Caster Wheel
- Drill
- 12V Greartisan DC Motor

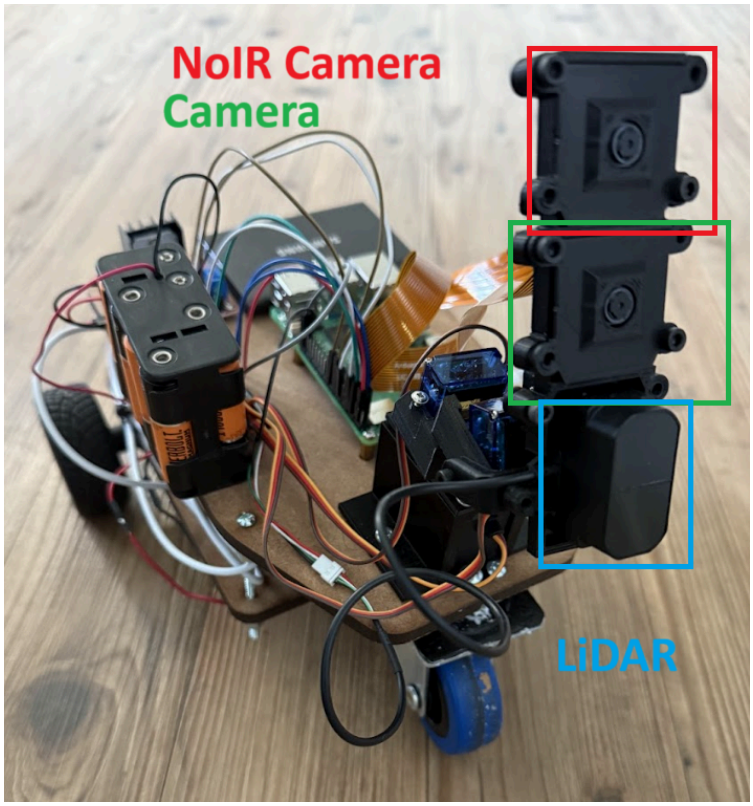


Processing Unit:

- Raspberry Pi 5

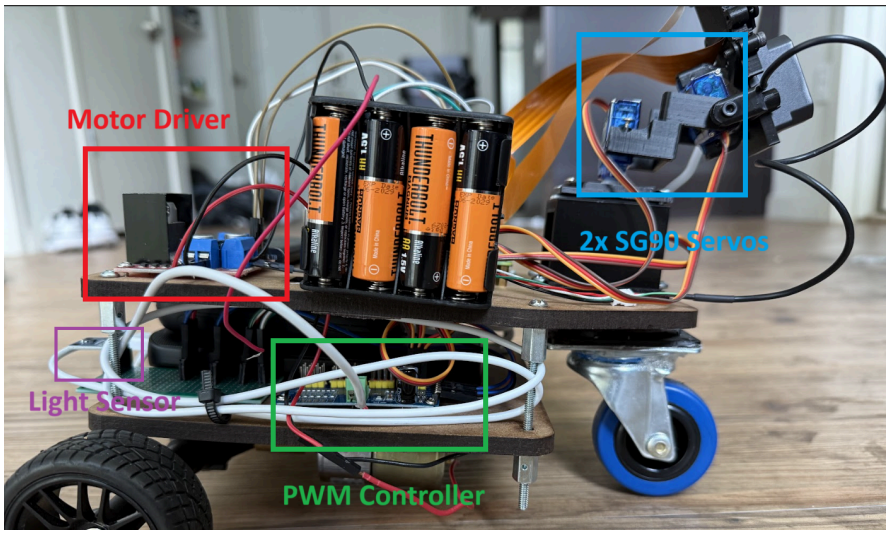
Sensors and Cameras:

- BH1750 Light Sensor
- LiDAR Module
- 1x Raspberry Pi NoIR Camera
- 1x Raspberry Pi Camera



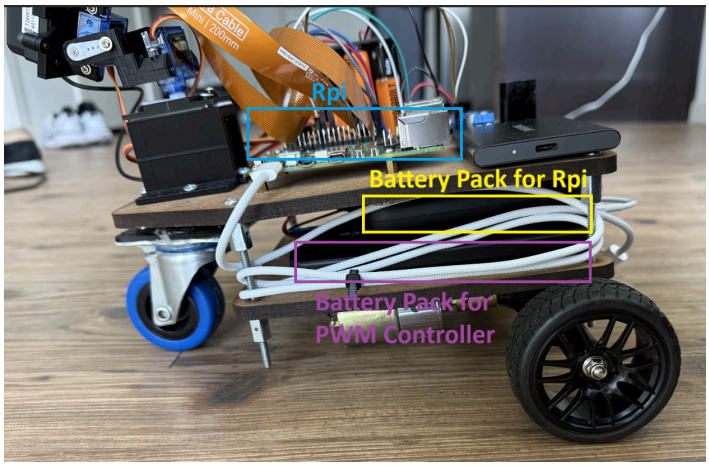
Motors and Controllers:

- 2x SG90 Servo Motors
- L298N Motor Driver
- PCA9685 PWM Controller



Power Supply:

- 2x Battery Packs
- 8 AA Battery Holder



Additional Tools and Materials:

- 3D Printer
- Soldering Station
- Prototype Board Pack

Hardware Descriptions

Chassis and Structural Parts:

- Car chassis - Forms the foundational platform of the robot, providing a stable base for mounting components and ensuring balanced weight distribution. This was modified with longer screws and spacers to create additional internal clearance for the dual power systems and electronics.
- 1x Caster Wheel - For directional support, a caster wheel was added. It was glued and attached to the front wheel servo via a custom 3D-printed adapter. This works in conjunction with the main 2 driver wheels in the back to provide smooth, controlled movement
- 12V Greartisan DC Motor & Back Wheels - For forward and reverse movement of the car.

Processing Unit:

- Raspberry Pi 5 - Serves as the central processing unit, processing input from sensors and running algorithms for pose tracking, navigation, fall detection, and alert protocols.

Sensors and Cameras:

- BH1750 Light Sensor - Continuously monitors ambient light to automatically adjust the camera for optimal performance.
- LiDAR Module - Provides precise distance measurements, enabling accurate depth perception for obstacle avoidance and maintaining optimal following distance.
- 1x Raspberry Pi NoIR Camera - Provides night vision capability for effective operation in low-light conditions.
- 1x Raspberry Pi Camera - Serves as the primary camera for user tracking and fall detection.

Motors and Controllers:

- 2x SG90 Servo Motors - Used for camera adjustments (pan and tilt movements), ensuring consistent target tracking.
- L298N Motor Driver - Controls the DC motors, enabling forward and backward movement.
- PCA9685 PWM Controller - Manages multiple servo motors for efficient operation with accurate timing and smooth motion.

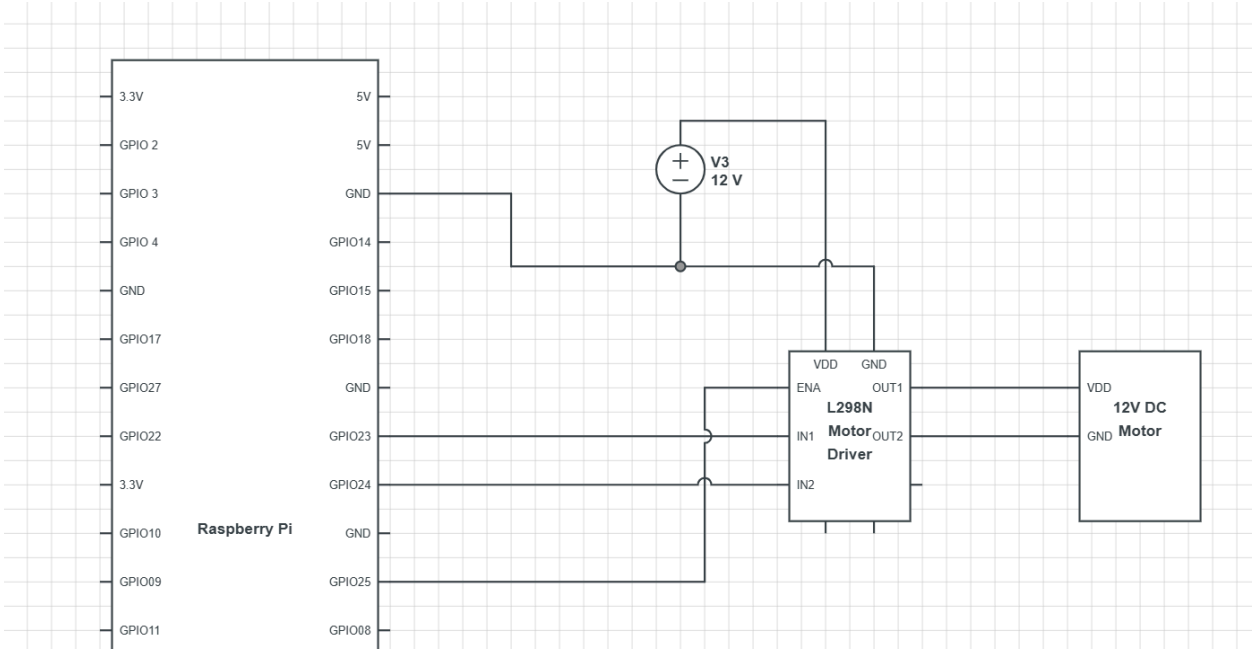
Power Supply:

- 2x Battery Packs - This dual power system powers the Raspberry Pi and PWM controller (5V/3A for Raspberry Pi, 5V/3A for PWM Controller). The second pack was required since the PWM controller also needed 5V, and the first one only had one port supplying 5V already used for the Pi.
- 8 AA Battery Holder w/ 8 1.5V batteries - Provides a dedicated 12V power supply for the 12V Greartisan DC Motor to move the system.

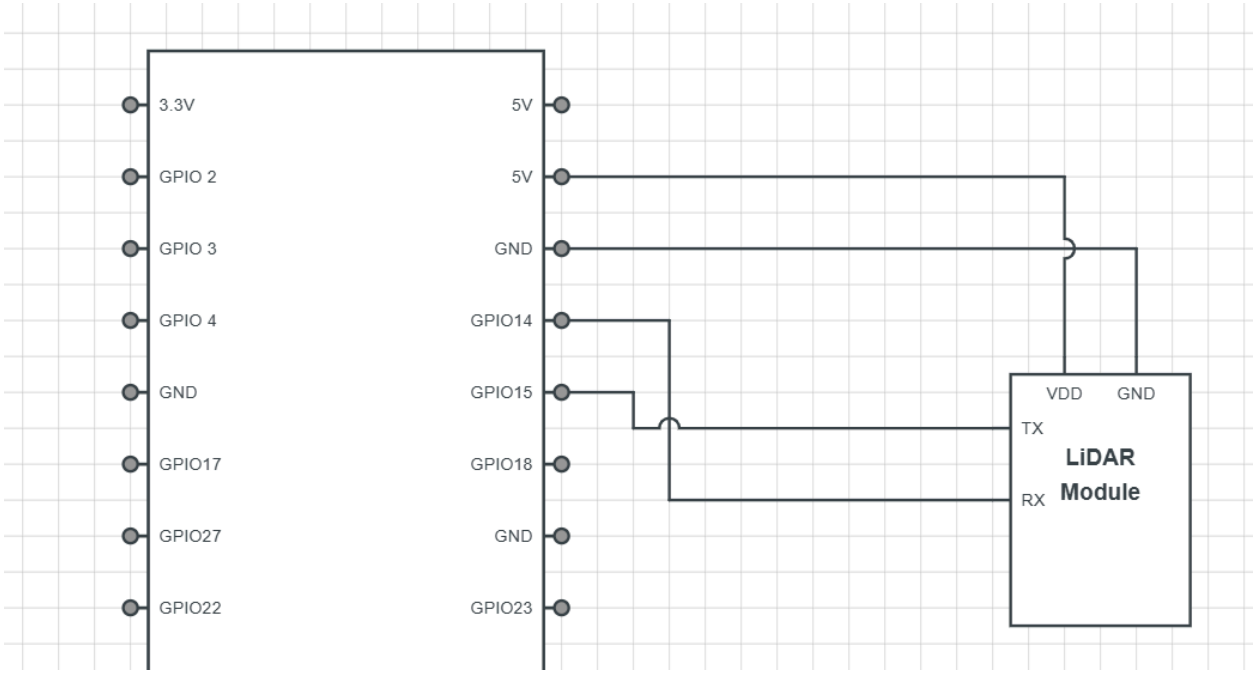
Circuit Diagram

Provided is the finalized circuit diagram for our device, which is separated to aid in comprehensibility. Do note that GPIO 2 (SDA) and GPIO 3 (SCL) are in parallel for the BH1750 and PWM Controller, and as is with any other pins that go to multiple devices.

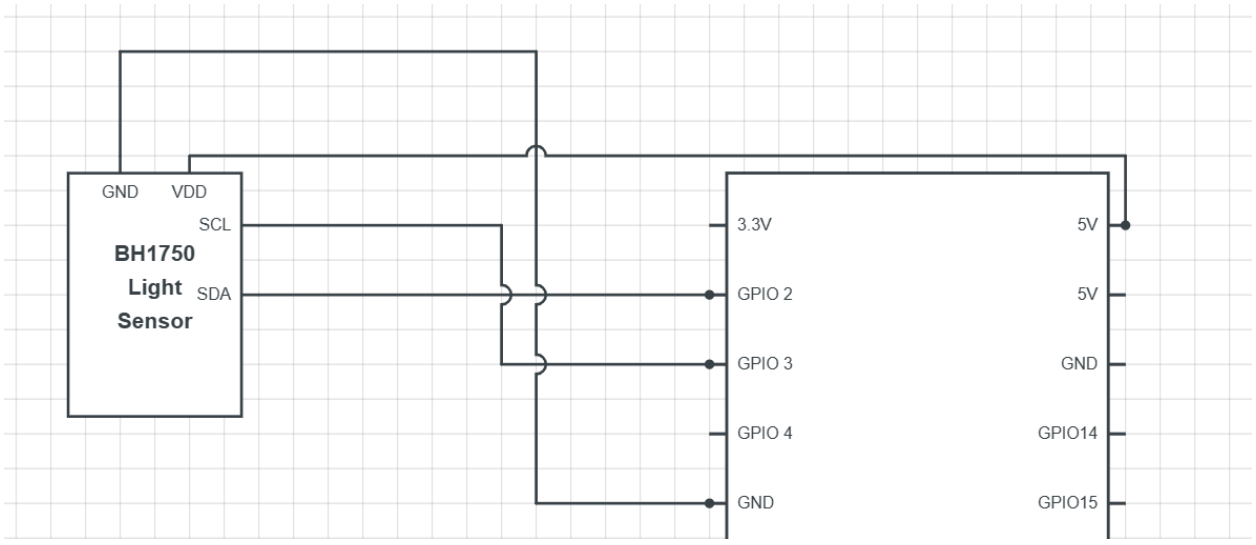
L298N Motor Driver and 12V Greartisan DC Motor:



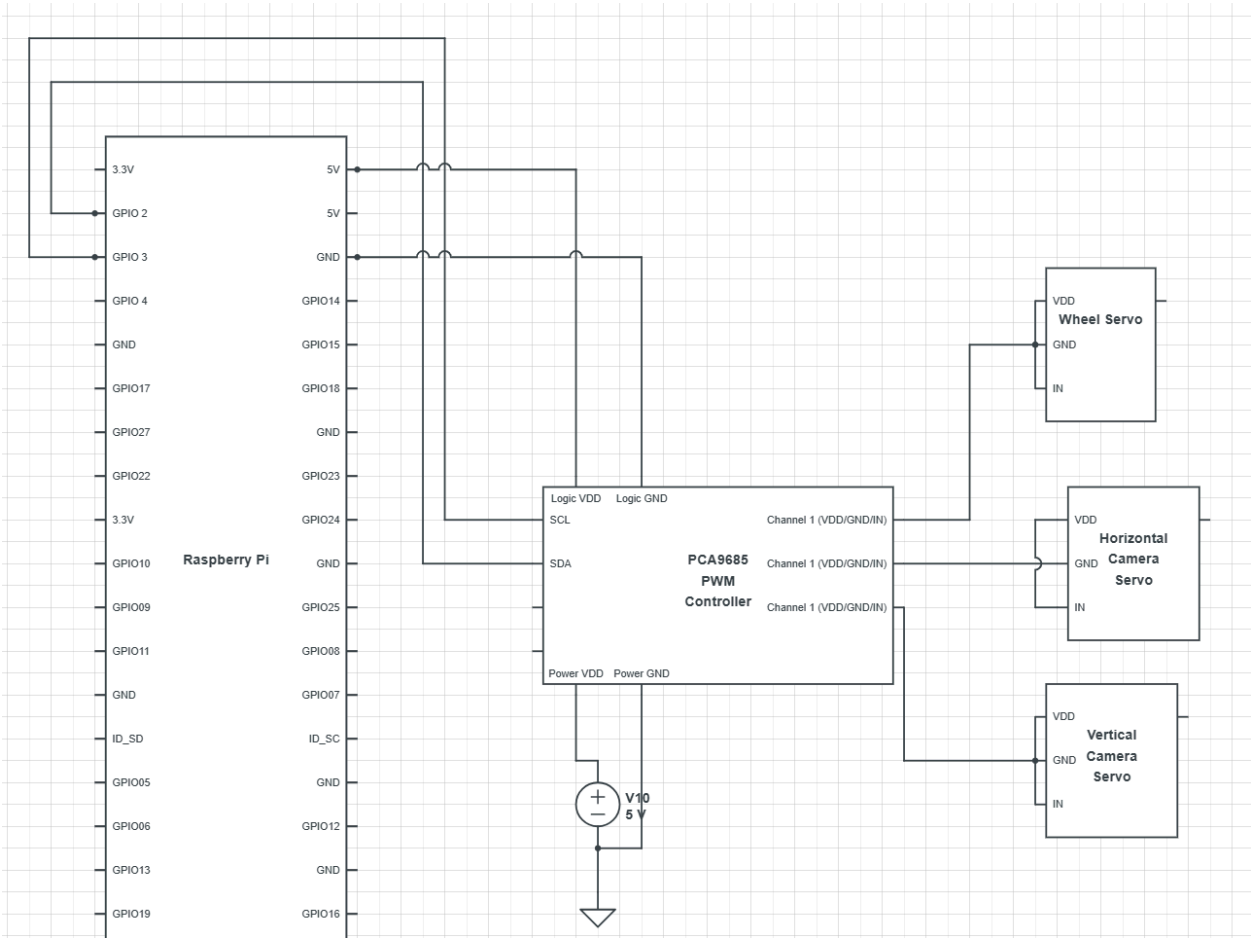
LiDAR module:



BH1750:



PWM Controller (Combined Servo Pins for readability):



Software Descriptions

The program begins by importing all the necessary libraries and modules required for computer vision, hardware control, communication, and logging. Most importantly, it includes picamera2 for camera operations, mediapipe for pose detection, cv2 for image processing, numpy for numerical operations, RPi.GPIO for GPIO pin control on the Raspberry Pi, adafruit_pca9685 and adafruit_servokit for servo control, Twilio for sending messages via WhatsApp, and others. The program initializes the MediaPipe pose detection module with specified confidence levels for detection and tracking. This sets up the environment for detecting human poses in camera frames. The I2C bus is initialized to communicate with the PCA9685 PWM controller, which controls the servos (2x SG90 Camera Servos and 1x SG5010 Turning Servo). A ServoKit instance is created for controlling multiple servos connected to the PWM controller. Constants defining servo channels, servo angle limits, center positions, and movement thresholds are established. These will be used to control the camera's pan and tilt and the robot's steering. GPIO pins controlling the motors are configured, and PWM is initialized to control motor speed.

After the necessary imports and defined constants, the software instantiates two overarching classes: PersonFollower and LocationWhatsAppSender. The PersonFollower class serves as the core component of the system, orchestrating all the hardware and software subsystems to achieve autonomous person-following and fall detection. The LocationWhatsAppSender class is dedicated to handling communication for fall alerts, specifically through WhatsApp messages using Twilio.

When the PersonFollower class is initialized, the constructors will be automatically called. The primary camera (camera 1) is initialized using Picamera2, and the camera stream is started to allow for frames to be processed. The pan and tilt servos controlling the camera are set to their center positions, ensuring the camera starts facing forward. Calibrated PID control parameters (Kp, Ki, Kd) are initialized for smooth servo movements when tracking the person. Also, error-tracking variables for the PID controller are initialized. Variables related to scanning (searching for the person when lost) are initialized, including scanning direction and timing. Parameters

defining the desired following distance, tolerance, and movement cooldowns are set. Variables to manage motor control timing are initialized. The serial port is configured to communicate with the LiDAR sensor, which will let us measure the distance to the person. Additionally, variables to track potential falls, fall timing, and alert cooldowns are initialized. Then, the BH1750 light sensor is also initialized to detect ambient light levels, allowing the system to switch between normal and night vision cameras as needed. An Image processing algorithm CLAHE (Contrast Limited Adaptive Histogram Equalization) is initialized to enhance frames in low-light conditions.

After all the variables and modules have been initialized, the run method of the PersonFollower class contains the main loop, which will continuously process frames, from Picamera2, and control the robot. Utilizing adaptive histogram equalization and gamma correction, these post-processing methods were applied to enhance the frame to more easily detect a pose in low-light conditions. Furthermore, every five seconds, the program checks the ambient light level using the BH1750 sensor, and if the light level falls below a certain threshold, the program switches to the night vision camera (camera 0). If the light level is adequate, it ensures the normal camera (camera 1) is active.

After determining the correct camera and processing the frame, the frame is converted to RGB and processed by MediaPipe's pose detection to identify human landmarks. If a person is detected, the program proceeds to tracking and movement control. The coordinates of the person's nose are extracted from the detected landmarks, and the update_camera_servos method calculates the error between the nose position and the center of the frame. PID control is used to adjust the pan and tilt servos smoothly, keeping the person-centered in the frame. However, if a person is not detected in multiple consecutive frames (tracked using frames_without_person), the program recognizes that the person has been lost and will scan around with start_scanning until a person has been found.

Once a person is identified and tracked with the camera, the control_car method calculates the required steering angle based on the camera's pan angle. Then, the robot's front wheel steering servo is adjusted accordingly. Then, the LiDAR sensor provides the distance to

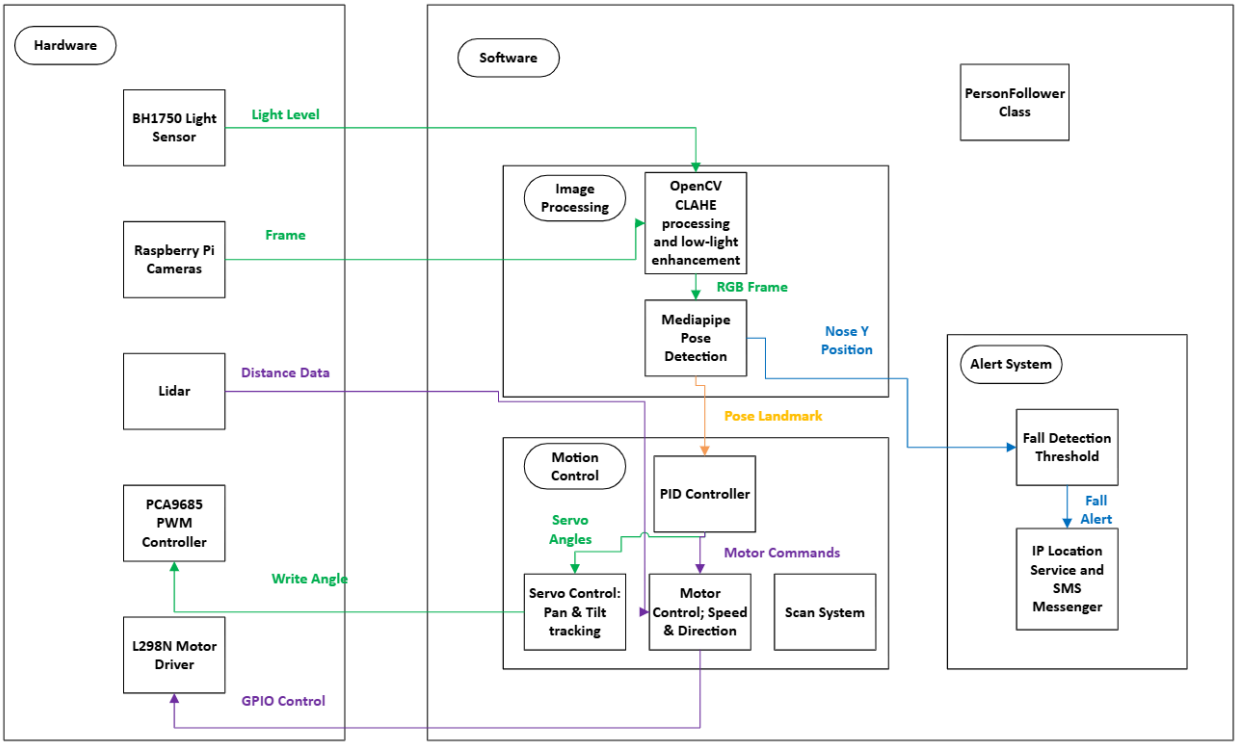
the person, deciding whether to move forward, backward, or stop based on the measured distance and the desired following distance.

After adjusting the position of the car, the program will utilize the handle_fall_detection method to monitor the vertical position of the person's nose. If the nose's Y-coordinate goes below a certain threshold in our frame, indicating the person may have fallen since a face shouldn’t be close to the ground, a fall detection timer starts, which takes into account if a person is bending to pick something up. If the condition persists beyond the FALL_DELAY of 5 seconds, a fall is confirmed.

Upon confirming a fall, the program checks if the alert cooldown period has passed to prevent spamming. It calls the send_fall_alert method of the LocationWhatsAppSender class, which retrieves the device's approximate location using the get_ip_location method, which uses an external API to determine location based on the public IP address. The format_location_message method formats the location information into a readable message, including a Google Maps link for easy access to the immediate location.

Then after a fall has occurred or not, the main loop continues to execute, processing frames, controlling servos and motors, and monitoring for falls. In all, this allows us to provide real-time person-following and fall detection capabilities to effectively take care of elderly folks when caretakers aren’t able to physically be with them, giving them peace of mind.

Software Diagram



Contributions

Alex:

- Physical ware
  - Contributed to integrating wood and 3D-printed components for a robust design.
  - Contributed to designing the larger housing structure for the RC car to optimize component placement
- Software for camera tracking PID control loop
  - Developed and calibrated a PID control loop for camera tracking to ensure smooth and precise person tracking, along with more reliable low-light tracking
  - Worked together to conglomerate the code into a full, operational system
- Hardware
  - Assisted with wiring the system, including the power source, and soldered connections for reliability
- CAD Design
  - Utilized Fusion360 to create custom CAD components for 3D printing, tailored to the project's design needs

Dhiraj:

- Physical ware
  - Combined wood and 3D-printed components for seamless integration
  - Enhanced the existing RC car design to accommodate new functional requirements
  - Determined and designed the layout of the RC car's larger housing for better component organization
- Software for reliable fall detection
  - Designed, implemented, and calibrated a reliable fall detection algorithm for improved functionality
  - Worked together to conglomerate the code into a full, operational system
- Hardware
  - Played a key role in wiring the system, connecting the power source, and soldering wires
  - Diagnosed and resolved hardware issues caused by faulty components

Arun:

- Physical ware
  - Assembled wood and 3D-printed components using hot glue for a cohesive structure
  - Participated in designing the larger housing of the RC car for optimized component arrangement
- Software for movement of the RC car and SMS messaging
  - Implemented and calibrated an algorithm to enable the RC car to follow a person accurately
  - Ensured reliable SMS messaging functionality, including fall detection notifications.
  - Worked together to conglomerate the code into a full, operational system
- Hardware
  - Supported wiring tasks, connecting the power source, and soldering components to ensure system reliability

Outcomes

The outcomes of our project include:

- **Functional Mobility Subsystem:**
  - The robotic system successfully moves and navigates based on user location. The motors, controlled via the L298N Motor Driver and powered by a 12V battery, provide smooth and consistent motion. The caster wheel, SG5010 Servo motor, custom 3D printed servo-caster-wheel adapter, and a new motor driver ensured accurate turning and path adjustments.

- **Functional Tracking Subsystem:**
  - Using the Raspberry Pi Camera, Raspberry Pi NoIR, and LiDAR module, the robot autonomously tracks and follows a single person. The tracking algorithm reliably integrates sensor inputs to maintain proximity, even in dynamic environments. Debugging efforts resolved initial conflicts between modules, resulting in seamless functionality.

- **Functional Fall Detection Subsystem:**
  - The fall detection system accurately identifies falls by analyzing the person’s position with a height-based threshold and a timer. The system minimizes false positives, like picking something up, while providing immediate alerts, and ensuring timely assistance when needed.

- **Functional GPS and Alert Subsystem:**
  - While the outdated GPS module posed challenges, we implemented a modern solution using APIs for IP location tracking and Twilio for instant messaging. This subsystem provides caregivers with real-time updates on the user's location and fall status in case of emergencies.

Bill of Materials

Component/Material	**Source**	**Price**
Wooden car chassis	Lab	$0
4x Long Screws	Ace Hardware	$8
Drill	Already had	$0
Raspberry Pi 5	Amazon	$92
BH1750 Light Sensor	Amazon	$6
Solderer	Lab	$0
L298N Motor Driver	Amazon	$6
2x Battery Packs	Already had	$0
2x SG90 Servo Motors	Amazon	$6
PCA9685 PWM Controller	Amazon	$14
3D Printer	Already had	$0
8 AA Battery Holder	Lab	$0
Various cables	Already had	$0
1x Caster Wheel	Harbor Freight	$8
1x Raspberry Pi NoIR Camera	Amazon	$17
1x Raspberry Pi Camera	Amazon	$17
Lidar Module	Amazon	$49
Arducam Cables	Amazon	$10
Prototype Board Pack	Amazon	$12

Links

GitHub Repository:
https://github.com/AlexDo12/CSCE_462/blob/main/HumanRadar.py

References

[1] A. Rosebrock, "Pan/tilt face tracking with a Raspberry Pi and OpenCV" 2019. [Online]. Available: https://pyimagesearch.com/2019/04/01/pan-tilt-face-tracking-with-a-raspberry-pi-and-opencv/

[2] C. A. Q. Bugarin, J. M. M. Lopez, S. G. M. Pineda, Ma. F. C. Sambrano, and P. J. M. Loresco, “Machine Vision-Based Fall Detection System using MediaPipe Pose with IoT Monitoring and Alarm,” IEEE Xplore, Sep. 01, 2022. https://ieeexplore.ieee.org/document/9929527

[3]“MediaPipe Solutions guide | Edge,” Google for Developers. https://ai.google.dev/edge/mediapipe/solutions/guide