ECE 445 Design Document - Fall 2025

A Follow-Me Cart: A smart shopping assistant

Project #3

Zixuan Huang, Jiaming Gu, Shi Qiao
{zixuan23, jgu13, shiqiao2} @ illinois.edu

Professor: Arne Fliflet

Table of Contents

1. Introduction

1.1. Problem

In supermarkets, shopping centers, and malls, shoppers often experience inconvenience when pushing heavily loaded carts while simultaneously browsing items, handling phones, or managing children. This situation not only reduces comfort but also efficiency, leading to a poorer shopping experience.

With the growing trend of smart retail and service robots, there is a demand for semi-automatic shopping carts that can reduce physical strain and allow customers to shop more comfortably and efficiently. Prior works such as autonomous luggage and robotic carts have demonstrated feasibility, further justifying the importance of developing such a system.

There is a need for a semi-automatic shopping cart system that follows customers and The user-friendly mobile application allows customers to control the shopping cart, increasing convenience, reducing physical strain, and improving the shopping experience for a wide range of customers.

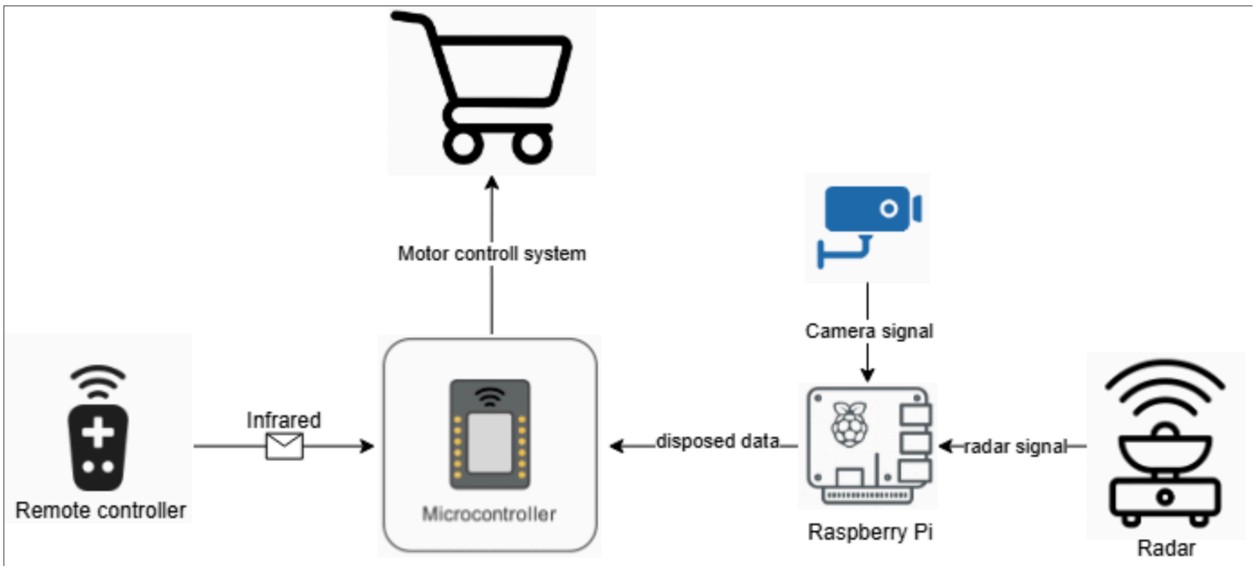
1.2. Solution

To address this challenge, we propose a Follow-Me Cart system controlled via a remote controller. The architecture integrates a Raspberry Pi for The system includes high-level processing, such as sensor fusion, user tracking, and communication with the app, while an ESP32 microcontroller is responsible for low-latency motor control and ultrasonic safety features. This hybrid design integrates these components effectively balances computational intelligence with real-time reliability.

The Pi processes data from an mmWave radar and camera to track the user, while the ESP32 enforces safety and ensures low-latency motor response. This architecture will balance high-level intelligence with reliable real-time control.

Instead of being completely automatic, the cart connects to a remote controller through Wi-Fi or infrared. Customers can enable follow-me mode, control the car's speed, adjust the following distance and stop or start the cart as needed. The user carries an infrared remote, while the shopping cart prototype is equipped with a receiver and simple motor control. The cart is able to maintain a following distance of 1–2 meters in an open experimental area, and it can stop or bypass when encountering obstacles. The navigation system can be based on a fixed map, since supermarket layouts are generally static. Instead of dynamic pathfinding, the cart only needs basic obstacle avoidance when encountering people or objects. As long as the paths remain open, the cart can reach its destination. In cases where a path is completely blocked, the cart may stop, which is acceptable since it is outside the system's responsibility. To remain affordable and safe, we will construct the system at a smaller scale. The size is scaled down so it can hold roughly 50–70% of the weight of a real shopping cart, with a payload capacity of 5–10 kg, compared to the usual capacity of 30–40 kg.

1.3. Visual aid



This diagram shows the end-to-end data flow for the cart. The camera and the mmWave radar feed raw sensor streams to the Raspberry Pi, which runs the perception and decision logic (detection, tracking, and range fusion). The Pi outputs compact motion targets—desired speed V_target and ψ_target—to the microcontroller (ESP32). The microcontroller executes the real-time control loops at 100-200 Hz, which will translate those targets into differential PWM for the motor driver. Operator input enters through the **infrared remote (IR)** instead of a phone: an IR receiver connected to the microcontroller decodes commands (Follow/Hold/E-stop, speed cap, distance setpoint), and the RTC arbitrates these with autonomy and safety rules. In short, sensing -> Raspberry Pi for perception/decisions -> microcontroller for real-time control -> motor system, with an IR remote providing user commands.

1.4. High-level Requirements List

1.4.1 Maneuvers

We aim for the cart to achieve a 90% success rate in avoiding collisions with other objects in environments similar to grocery stores. The abovementioned objects include but are not limited to people, shelves, tools, and other shopping utilities.

1.4.2 Uptime

According to a recent market survey by Drive Research, about 74% of U.S. consumers report that their full grocery shopping trip takes under 44 minutes. Drive Research While this includes all activities (entering store, browsing, checkout, etc.), it provides an empirical basis for requiring our cart to operate continuously for 40-60 minutes to cover the majority of typical shopping sessions.

1.4.3 Accuracy

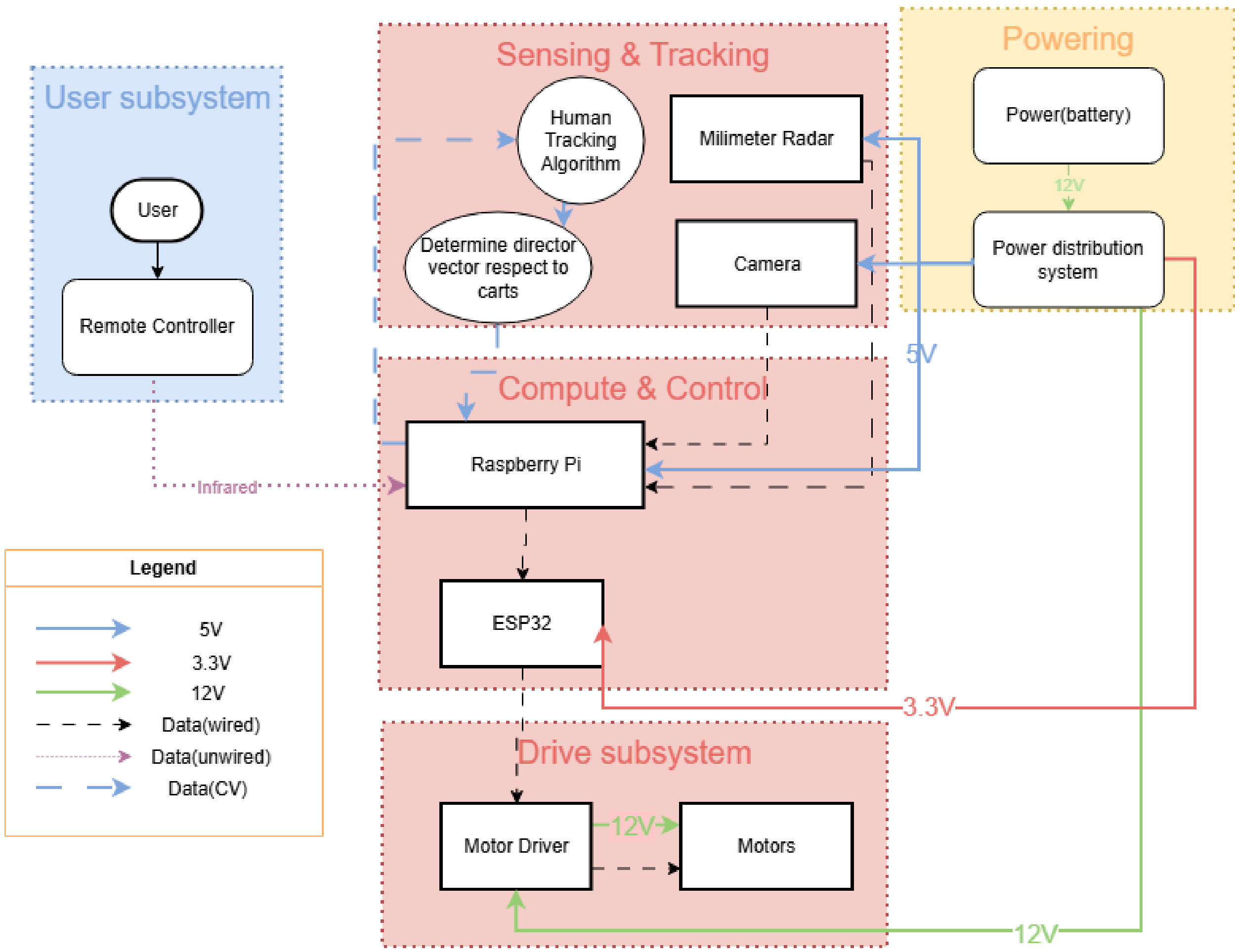
The cart is designed to maintain a following distance of 1–2 meters from the designated customer within a pre-mapped supermarket-like environment. This range ensures that the cart remains close enough to be convenient while avoiding excessive proximity that could inconvenience the user or risk collisions.

1.4.4 Remote connection

The cart shall connect via infrared to a remote controller to assist user tracking and provide teleoperation/diagnostic capabilities. This connection should allow bidirectional data exchange, enabling the user to issue simple commands (stop/move) and monitor system status.

2. Design

2.1. Block Diagram

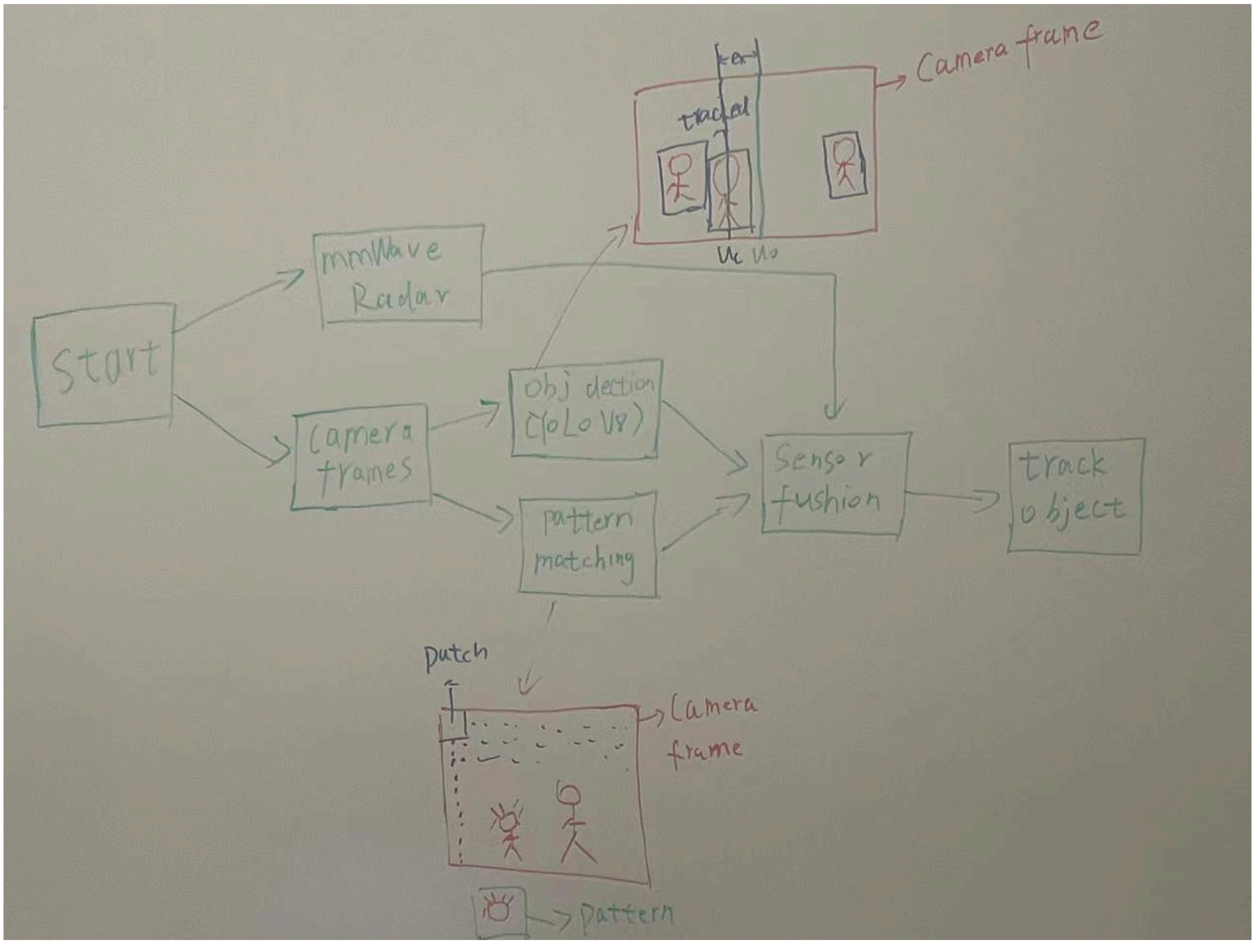


Below are the explanation of each subsystem:

2.2. Subsystem Overview

1. Subsystem 1: Sensing and Tracking subsystem
   This subsystem consists of sensors and a camera. In this project, we will use several mm-wave radars working together with a camera. In this mode, the camera will handle most of the data collection, while mmWave sensors will be activated as needed in situations like turning or approaching other objects to better assist with maneuvering the cart.
2. Subsystem 2: Compute and Control subsystem
   This system will receive data from all other systems, analyze them using Raspberry Pi if needed, then forward the abovementioned processed information to ESP32, while the latter component drives motors on the cart.
3. Subsystem 3: Drive Subsystem
   This system consists of the chassis, wheels, and motors of our cart, it is powered by the on-board battery and will follow instructions given by the ESP32 chip.
4. Subsystem 4: Power Subsystem
   The power system consists of a central battery and a power distribution unit. It provides 12 V to the motor driver, 5 V to the sensing and tracking modules as well as the Raspberry Pi, and 3.3 V to the ESP32 microcontroller.
5. Subsystem 5: User Subsystem
   The User Subsystem consists of the user and remote controller, where the user will send instructions like STOP, HOLD, or FOLLOW to the Compute and Control Subsystem via IR. And IR’s digital output will connect to the GPIO on ESP32 and ESP32 will communicate with the Drive Subsystem to implement the movement accordingly.

2.2.1 Subsystem 1: Sensing and Tracking Subsystem



The sensing and tracking subsystem could detect the intended person in each camera frame, maintains a stable identity over time, fuses camera bearing radar range and gives a direction vector with respect to the cart for control. A lightweight YOLO-style detector(like YOLOv8-nano, YOLOv5-nano) runs at 5-10FPS on the Raspberry Pi to produce person boxes. A tracker(Kalman Filter+Association) locks onto the target instance and smooths jitter. From the tracked box center uc and the calibrated image center u0, the pipeline computes pixel error ex = uc-u0 and converts it to bearing ψ =arctan(ex/fx) using the horizontal focal length fx. Distance d will come from the mmWave radar. A 2D constant-velocity Kalman filter combines(d,ψ) over time and outputs d_hat = [d,ψ] plus a confidence flag for each cycle. If the confidence is low(blur, ID switch), the module freezes ψ and ramps V_target toward zero. The vector d_hat is released at 5-10Hz with latency under 80ms and drives the **compute and control layer.**

Requirements	Verification
Reliable user lock: tracker keeps the same user ID for >=90% of frames during a 5min walk	Record a 5-min run; count ID switches in the log. Checking if overall ID consistency >=90%

Direction vector accuracy: bearing error ≤ 5° and distance MAE ≤ 0.25 m from 1–3 m.	Place markers at 1, 2, 3 m and have the user stand on each marker. Report the mean error.
Being Able to compute the PID error	When there are people showing up in the image,testing whether the PID error are correct or not when people are not standing right in the middle of frame(left/ right)

2.2.2 Subsystem 2: Compute and Control Subsystem

This subsystem performs planning decision-making, and motion control based on perception subsystem outputs so the platform can **maneuver safely and avoid collisions**, meeting high-level requirements HLR-1 (**maneuvers**), HLR-2 (Accuracy), and HLR-4 (user control via remote controller). It ingests sensors/tracking data, classifies humans vs. objects, decides actions (turn, brake, yield), and commands the drive subsystem.

2.2.2.1 Software and Dataflow

The software and dataflow for this subsystem describe how sensing, perception, decision, and control interact continuously between the high-level computer (HLC) such as Raspberry Pi and the real-time controller (RTC) such as ESP32. The **sensor fusion** process, implemented on the ESP32 microcontroller, merges measurements from the IMU and wheel encoders using a Complementary or Multiplicative Extended Kalman Filter (MEKF). This produces stable estimates of the cart's yaw, velocity, and angular rate along with their covariance, providing the foundation for all higher-level motion control.

On the HLC, the **perception layer** runs at 10 Hz and is responsible for interpreting sensor data to understand the environment. Images are pre-processed and fed into a lightweight convolutional neural network, such as YOLO-Nano, to detect and classify both humans and static objects. The perception system then tracks these detections over time using SORT or ByteTrack algorithms, combining them with depth information from radar to compute range and position. The entire perception pipeline is optimized to maintain an end-to-end latency below 80 ms.

Next, the **decision layer**, which operates at 20 – 50 Hz, determines how the cart should react to its surroundings. Detected obstacles are inflated by safety radii (0.75 m for humans and 0.4 m for other objects). When a human is within 2 m, the system's yield logic commands the cart to decelerate smoothly to a stop. Otherwise, the cart follows a Pure Pursuit path-tracking algorithm with curvature limited to $|\kappa| \le 0.25\ m^{-1}$. A cost function(1) is used to calculate the error.

$$J = w1 \times d_{goal} + w2 \times |\kappa| + w3 \times \sum \varphi(di) + w4 \times 1(human\_near) \quad (1)$$

Equation 1 balances distance to goal, turning smoothness, and obstacle proximity to ensure safe navigation. Here d_goal stands for the distance from the center point of the camera to the target user and absolute value of K indicates the smoothness of the turning, encouraging the cart to keep focus on the user and make moderate turns to ensure safety. φ(di) represents the total penalty function based on the proximity of obstacles; it increases sharply when any obstacle is closer than the safety distance threshold, prompting the decision layer to reduce speed or change direction to maintain safe spacing. This strategy aims to balance the hierarchy between following the user and avoiding obstacles. We don't want the cart to be easily blocked by an obstacle between it and the user. It should have the ability to take risky decisions. The decision algorithm should come out with the desired angle and speed in order to minimize the cost J.

The **Control Layer** serves as the PID-based execution stage within the RTC. It receives high-level motion targets (v_target, ψ_target) from the Raspberry Pi's decision system, computes the corresponding errors (e_v, e_ψ), and uses PID equations to generate real-time PWM motor commands. In this unified structure, the control layer itself embodies the PID logic—translating strategic goals into precise actuation signals. The RTC employs a watchdog that halts motion if no valid command is received within 50 ms, ensuring both precision and safety.

2.2.2.2 PID Controller Formulation

The ESP32 executes two PID controllers at **100–200 Hz**:

Longitudinal PID (speed control):

$$u_{v(t)} = Kp_{v} * e_{v(t)} + Ki_{v} * \int e_{v(t)} dt + Kd_{v} * \frac{de_{v(t)}}{dt} \quad (2)$$

where e_v(t) = v_target − v̂. This adjusts motor PWM duty to match desired speed.

Lateral PID (heading/curvature control):

$$u_{\psi(t)} = Kp_{\psi} * e_{\psi(t)} + Ki_{\psi} * \int e_{\psi(t)} dt + Kd_{\psi} * \frac{de_{\psi(t)}}{dt} \quad (3)$$

where $e_\psi(t) = \psi_{target} - \hat{\psi}$. The output controls differential wheel speed or steering actuator.

Initial parameters:

- Longitudinal PID: $Kp_v = 1.0$, $Ki_v = 0.2$, $Kd_v = 0$.
- Lateral PID: $Kp_\psi = 1.4$, $Ki_\psi = 0$, $Kd_\psi = 0.05$.

We will adjust these parameters when building and testing the system in order to get a desired performance.

Requirements	Verification
Following Distance: The cart shall maintain a consistent following distance of 1–2 m from the designated user for ≥ 95 % of time. This ensures user convenience while maintaining a safe margin to avoid collisions.	Use a range sensor or motion-capture system to record distances at 50 Hz during a 5-minute run; compute the percentage of samples within range and verify stability during stops and turns.
Safety Stop: When any radar sensor detects an obstacle closer than 0.7 m, the braking response must occur within 100 ms. This rapid reaction minimizes collision risk and ensures compliance with safety requirements.	Conduct tests by moving a reflective obstacle toward the cart at various speeds and measure the time difference between detection and brake activation using a logic analyzer or oscilloscope.
Emergency Stop: When the E-stop button is pressed, all drive commands must be disabled and motor current reduced to zero within 50 ms. This guarantees immediate shutdown capability in emergencies.	Use an oscilloscope to record the time between E-stop signal activation and the drop in motor current; perform 10 repeated tests to confirm consistent reaction time and mechanical braking reliability.

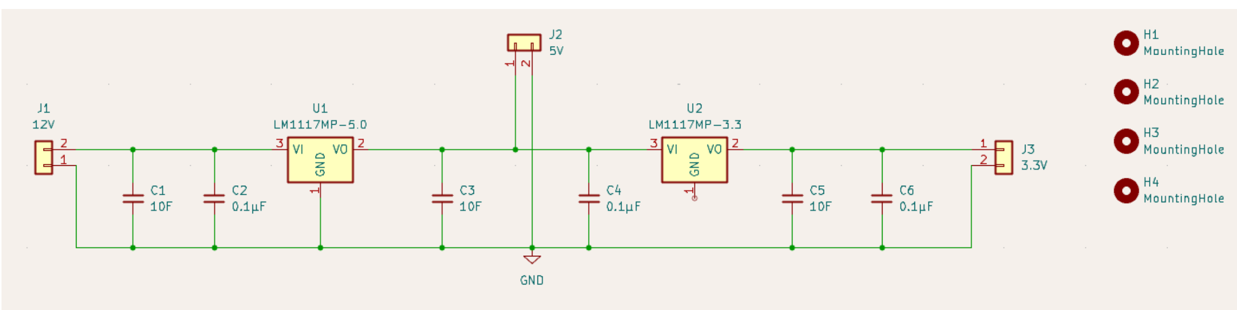
2.2.3 Subsystem 3: Driving Subsystem

The drive subsystem translates **V (speed) and ψ(heading)** from the RTC into wheel torques with dynamics, current limiting and hardwired safety. The mechanical layout is rear-wheel differential drive with a free-swiveling front caster, chosen for excellent low-speed controllability and small swept path in aisles. Each 12V gear motor will yield a no-load wheel speed near 1.4m/s at 12V and a torque sufficient to overcome static friction on smooth flooring at <=10KG payload. Electronic commutation is via a dual H-bridge driver module specified for ≥30 A aggregate peak and ≥12 A continuous; the **PWM** carrier is set to 19.5 kHz to move acoustic energy out of the audible band and to reduce current ripple.

Electrical and EMC details ensure reliability in noisy environments. The 12 V rail feeding the drivers includes a low-ESR bulk capacitor bank (≥2200 μF close to the H-bridges), and each motor lead is fitted with 0.1 μF ceramic capacitors to ground to attenuate brush noise. Motor harnesses are twisted and routed separately from camera and radar cables. The subsystem exports wheel-speed and current telemetry to the HLC for logging, and it reports latched fault codes (over-temp, stall, undervoltage) that must be cleared by operator action.

Requirements	Verification
Step from 0->1.0m/s within 1s	Apply speed step and analyze five step response
Minimum turning radius ≤1.5 m at 0.6 m/s;	Drive circular path around the marker and compute its velocity
E-stop removes motor drive within ≤50 ms	Scope E-stop line and motor current sensor and measure delay over 10 trials

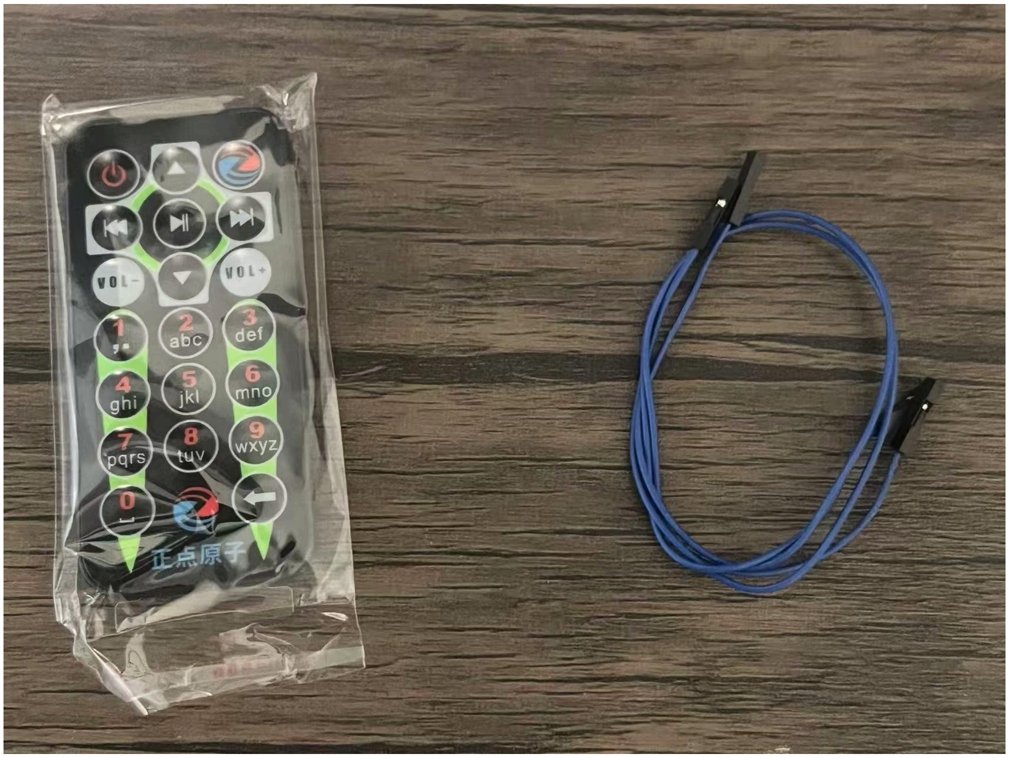
2.2.4 Subsystem 4: Power Subsystem



Here's the schematic diagram we build for voltage regulators within the Power Subsystem. Our power subsystem will accept 12 V input at J1 and locally generate 5V(J2) and 3.3V(J3) using LM117MP-5.0(U1) and LM117MP-3.3(U2) linear regulators. Each regulator follow the decoupling: 10 μF bulk + 0.1μF ceramic at the input(C1/C2, C4) and 10 μF bulk + 0.1μF ceramic at the output(C3/C6, C5) to ensure stability across load steps and cable inductance.Grounds are tied with a low-impedance plane; the regulators' tab pads are connected to large copper areas to spread heat. The LM1117's dropout is around 1.1V at 800mA. Regularization accuracy is typically +-1%——2% with PSRR ~60dB at 120Hz and tens of db into the KHz range. Because these are **linear regulators,** dissipation is (Vin -Vout)I: for 5v, every 100 mA costs **~0.7 W**, and for 3.3 V every 100 mA costs **~0.87 W**.For those high-current consumers(e.g. Raspberry Pi, camera and motor driver) are powered from separate high-efficiency buck modules. This board is the quiet, low-ripple supply for control electronics. An inline **input fuse** and polarity protection live upstream in the system harness; this board provides the local regulation and decoupling close to the loads to minimize noise on the IMU and MCU. Test pads at each rail allow scope probes to verify ripple and transient droop during perception bursts and motor current steps. With these limits and placement, the board provides clean, low-noise rails for the real-time controller while avoiding thermal runaway typical of linear regulators at high current.

Requirements	Verification
Right voltages at normal use. 5 V = 4.85–5.15 V and 3.3 V = 3.20–3.40 V while supplying typical loads (5 V ≈ 200 mA, 3.3 V ≈ 150 mA).	Power the board from 12 V. Plug simple resistor/electronic loads that draw ~200 mA (5 V) and ~150 mA (3.3 V). Measure J2/J3 with a digital multimeter.
It can successfully output 5v and 3.3V so that other subsystems could work	Test it directly with the multimeter
One rail doesn't kill the other. If the 5 V rail is accidentally shorted, the 3.3 V rail still stays on and returns to normal when the short is removed (and vice-versa).	Briefly short 5 V to GND through a current-limited supply or fast fuse; watch 3.3 V on the Digital multimeter. Repeat by shorting 3.3 V.
No resets when motors run. With the cart driving, the ESP32 stays powered from this board (no brownouts/reboots) for 3 minutes.	Power the whole system, start/stop the motors several times, and drive in place for 3 min.
No big dip when something turns on. When a small device is plugged in (extra 100 mA), the rail stays above 4.9 V (5 V rail) and above 3.25 V (3.3 V rail).	With the board on, plug or unplug an extra 100mA load a few times. Watch the multimeter.

2.2.5 Subsystem 5: User Subsystem



The user subsystem will provide simple and low-latency input via infrared remote control instead of a phone. The IR receiver's digital output connects to an ESP32 GPIO with an interrupt driven decoder. Each key press will yield a 32 bit command frame plus an automatic repeat code while the key is held. Then the ESP32 will map decoded keys to high-level action like E-STOP, FOLLOW, HOLD. A tiny state machine will reject bounce ,ignore invalid frames and forward accepted commands to the decision stack as mode updates. For safety, if no valid command is received for >1s while in manual-follow mode, the ESP32 will issue a stop. The cart provides immediate feedback- an LED flash-whenever a command is accepted so the user knows the button took effect. Typical indoor range should be ranging from 5-8 m line-of-sight. At the same time, the receiver should be angled slightly upward to reduce the effect caused by floor reflections and mounted away from motor wiring to minimize **electromagnetic interference**.

Requirements	Verification
Command work reliably indoors to at least 5m	Standing 1,3,5,7m away from the cart. Pressing the mapped button checks if the command really works.
E-STOP is immediate	While driving at~0.8m/s, press E-STOP. Checking if the motors cut off and cart stops.
No spurious action will be caused by noise or repeats	Hold a button for 3s. Checking if the cart performs one continuous action and no unintended mode changes occur.

3. Tolerance Analysis

With a maximum speed of 1.1 m/s, assuming a conservative control delay of 0.5 s and a deceleration of 0.5 m/s², the total stopping distance is approximately 1.8 m. Given that the radar provides a 10 m detection range, even if the system only makes decisions when an obstacle is detected at 7 m, the cart can still stop safely before reaching the obstacle under worst-case conditions.

$$d_{total} = v * t_{reaction} + \frac{v^2}{(2\times a)} = 1.1 \times 0.5 + \frac{1.1^2}{2\times 0.5} \approx 1.76\, m$$

Another important point concerns the durability of the power system. According to official specifications, the Raspberry Pi 4 may consume up to 3 A at 5 V (≈15 W) under peak load (*Raspberry Pi Foundation*). In addition, other peripherals contribute to the total The ESP32 microcontroller uses about 0.5 A at 3.3 V (Espressif Systems), the Logitech USB camera uses about 0.25 A at 5 V (Logitech), and the TI mmWave radar sensor uses about 0.36 A at 5 V (Texas Instruments). Summing these values, the peak demand of the compute and sensing subsystems is roughly 17.5 W, which establishes the baseline requirement for the power distribution system.

Four MG513 motors were chosen as the propulsion units for the drive system. Assuming each motor draws 0.6 A at 12 V under typical load, the estimated power consumption is calculated as:

$$P = 0.6\, A \times 12\, V \times 4 = 28.8\, W$$

Thus, the total estimated peak power requirements are

$$P_{total} = P_{compute\ and\ sensing} + P_{drive} = 17.5\, W + 28.8\, W = 46.3\, W$$

Since we are using a 2550 mAh battery, the duration will be at least

$$T = 2.55\, Ah \times 12\, V \div 46.3W = 30.6\, Wh \div 46.3W \approx 0.66h \approx 40min$$

Hence, the proposed cart system is guaranteed to maintain operation for no less than 40 minutes under peak power demand. Given that the motors in real-world usage are not always operating at maximum output, the effective endurance is likely to be significantly higher.

4. Cost Analysis

Since the lowest hourly wage for UIUC graduates is approximately **$15/hour**, we adopt a moderate estimate of **$25/hour** for our calculation. Assuming each member works **2.5 × 20 hours per week** over **11 weeks**, the total salary per member is:

$$20\$/hour \times 2.5 \times 15\ hours/week \times 9\ weeks = 6,750\$$$

With **three team members**, the total estimated labor cost is:

$$6,750\$ \times 3 = 20,250\$$$

ITEM	AMOUNT	TOTAL	STATUS/REMARK
Raspberry Pi 5	1	80$	RECEIVED
Raspberry Pi Camera 3	1	27.99$	RECEIVED
Raspberry Pi Case	1	9.99$	RECEIVED
Raspberry Pi Camera Cable	1	4.99$	RECEIVED
Motor Driver	1	6.95$	RECEIVED
Millimeter Radar	8	12$	RECEIVED
IR Receiver	1	3.7$	PENDING PURCHASE
Wheel encoder	2	10$	PENDING PURCHASE
IMU	1	20$	PENDING PURCHASE
TOTAL		175.62$	

Table 4: List of items

As total, the cost will be:

$$20,250\$ + 175.62\$ = 20,425.62\$$$

5. Schedule

ACADEMIC WEEK	DEADLINE	ITEM
WEEK 06	10/04	OBTAIN the following item: ☐ Mount motors ☐ Caster ☐ Wheels ☐ Wire motor drive MEET the following criteria: ☐ Fully assemble the cart and make it moveable ☐ Attend PCB review with finished PCB design ☐ Finalize PCB
WEEK 07	10/11	MEET the following criteria: ☐ Extracting the image taken from camera(send to Raspberry pi) successfully ☐ Target lock on person we are tracking using combination of YOLO and pattern matching(CV)
WEEK 08	10/18	MEET the following criteria ☐ Be able to analyze the data from mmWave radar
WEEK 09	10/25	MEET the following criteria ☐ Be able to apply and test the obstacle avoidance system based on radar and CV.
WEEK 10	11/01	Fix bugs and in case of delayed works.
WEEK 11	11/08	PENDING
WEEK 12	11/15	MEET the following criteria ☐ Test on the obstacle avoidance system and fix bugs
WEEK 13	11/22	MEET the following criteria ☐ Test on the following system and fix bugs
FALL BREAK	11/30	MEET the following criteria ☐ Test on the whole cart and fix bugs
WEEK 14	12/03	MEET the following criteria ☐ Prepare for final demo
WEEK 15	12/08	MEET the following criteria ☐ Prepare for final presentation

6. Ethics and Safety

As our project advances, we stay steadfast in our commitment to comply with the ethical and safety standards set out by the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). We commit to maintaining these standards in the creation of this autonomous shopping cart system, ensuring that all our decisions and actions embody the utmost professionalism and integrity.

In accordance with the IEEE Code of Ethics, Clause 1, we pledge to prioritize the safety, health, and welfare of the public and to endeavor to adhere to ethical design and operating standards. Safety will be paramount during the design process, and ethical considerations will be incorporated at each phase. We shall furnish explicit guidelines to users, including parents and guardians, concerning the appropriate utilization and constraints of the robotic cart to facilitate educated and responsible decision-making.

A critical ethical and practical focus lies in ensuring that the cart operates safely in environments with elderly people and children, who may move unpredictably or be more vulnerable to collisions. To address this, the cart includes multi-sensor obstacle detection (mmWave radar and vision fusion) and real-time safety stops that trigger when any object is detected within 0.7 m. Additionally, soft braking and path-planning algorithms will be tuned to minimize sudden movements or impact forces, further protecting pedestrians in crowded areas.

Adherence to all pertinent rules and regulations governing this technology will remain a primary focus. Additionally, in accordance with Clause 3 of the IEEE Code of Ethics, we commit “to aid colleagues and co-workers in their professional growth and to endorse their adherence to this code of ethics.” We will cultivate a culture of ethical awareness and mutual accountability, promoting open communication that allows team members to express issues and seek help without reservation.

References

[1] Rodgers, Emily. “Grocery Shopping Stats: Where, When & How Much We Spend.” *Drive Research*, 25 Feb. 2025, www.driveresearch.com/market-research-company-blog/grocery-store-statistics-where-when-how-much-people-grocery-shop/?utm_source=chatgpt.com.

[2] Raspberry Pi Foundation. *Raspberry Pi 4 Model B Product Brief*. Raspberry Pi Trading Ltd., 2019. https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf.

[3] Espressif Systems. *ESP32 Series Datasheet*. Version 3.9, Espressif Systems, 2022. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[4] Texas Instruments. *IWR6843 Datasheet: mmWave Sensor*. Rev. D, Texas Instruments, 2020. https://www.ti.com/lit/ds/symlink/iwr6843.pdf.

[5] Logitech. *HD Webcam C270 Product Specifications*. Logitech, 2012. https://www.logitech.com/assets/44810/hd-webcam-c270.pdf.