MEGN540 Project Progress Report

Sebastian Negrete-Alamillo, Keenan Buckley, Chris Larson

1. Problem:

Many people are unable or unwilling to get up, walk to, and grab an item they need or desire around their home. Examples include elderly persons who need to take medication at a specific time daily but may be forgetful, and college students who are thirsty but too incapacitated to get their next beverage themselves.

We are designing and building a product that can deliver a necessary item to those people when they need it.

2. Design Concept:

We are building a two-platform tank-drive mobile delivery robot. The lower platform will secure our electronics, and the upper platform will carry the delivery payload. Upon activation, our robot will identify the person nearest it, drive to that person, and deliver the payload. Our robot will have the ability to:

1. Listen for and react to an activation signal. To begin, this will be a serial command issued via SSH.
2. Identify persons in its FOV and target the nearest person to it (if any).
3. Orient itself toward the person and drive to them in a straight line on a flat, carpeted surface.
4. Carry a payload of at least 16oz.
5. Stop within arm's reach of the target person to deliver the payload.

2.1. Sensors

- *Stereo camera:* Luxonis Oak-D Lite for visual odometry, depth estimation, and object detection.
- *IMU:* MPU-6050 for pose estimation and motion control feedback.
- *Load sensor:* MPS20N00400 sensor and HX711 amplifier to measure payload.
- *Wheel encoder (x2):* Hall encoders for motion control feedback.

2.2. Actuators

- *DC motor (x2):* 12V, 150rpm DC motors to power the robot's drivetrain.

2.3. PCB

- *LED Demuxer:* Our circuit board's primary purpose is to enable us to indicate our robot's status (locating a target, planning, delivery in progress, delivered, ready, error, etc.) visually without occupying too many pins on our microcontroller. Our PCB will incorporate a 3 to 8 demuxer, a resistor, and ports for 8 LEDs to achieve this. The PCB will also clean up our robot's wiring by supplying 5V power rails for our sensors. Our initial PCB design is in Figure 1.

2.4. Software

- *Visual odometry* module for depth perception and pose estimation.
- *Object detection* module for person identification.
- *Path planning* module for trajectory generation.
- *Motion control* module to power motors and follow the trajectory.

3. System Integration:

We worked as a team to assemble our robot's chassis and prototype mechatronic system. We are working in parallel on our robot's software modules. To simplify integration, we use Docker containers to ensure environment compatibility and a GitHub branch/pull/merge workflow to manage our codebase. During April, we will work together as a team to first integrate our software modules (running on Nvidia Jetson) with our mechatronic system (controlled on Arduino Uno), and then to implement our motion controller. We will assign debugging tasks individually as necessary. We intend to have a working product by 4/24 and one week to refine/optimize.

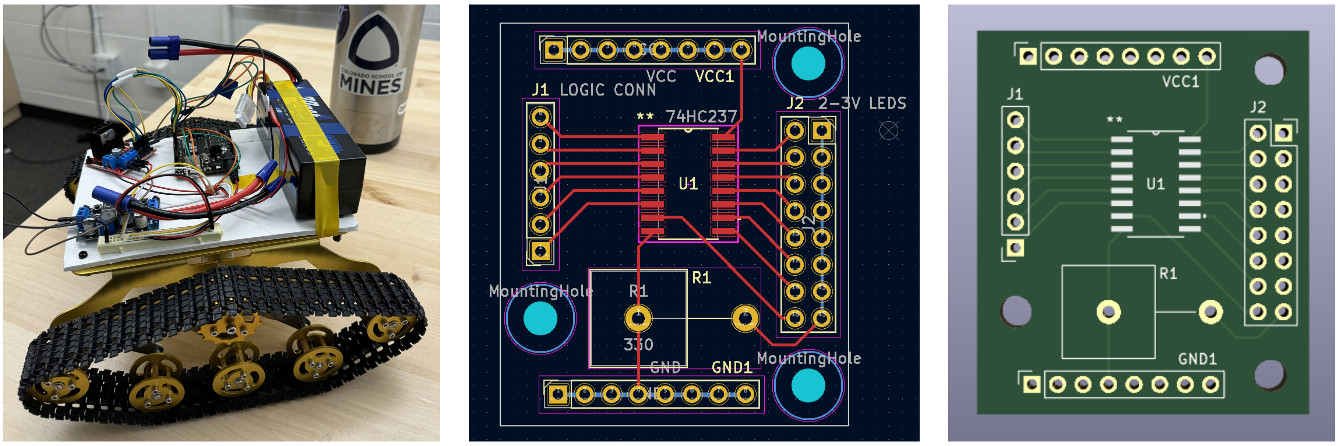

Figure 1: Works completed to date. (L) Prototype mechatronic system, (R) PCB design.

4. Project Plan:

Our work/deliverables are planned in two-week increments to keep tabs on progress and enable us to react quickly to issues that arise. We have created a backlog of deliverables on GitHub and are organizing our work using a Kanban board. Table 1 shows our milestones and their current status. We have completed the first three milestones and are progressing on the rest. Completed works are shown in Figure 1.

Table 1: Project plan and status.

Milestone	Date	Status	Description	Requirements
1	2024-02-07	✓	Material acquisition and planning	• Bill of Materials created • Ordered necessary materials
2	2024-02-21	✓	Software module design	• Project repo instantiated • Module specifications created • ROS framework installed
3	2024-03-06	✓	Prototype build and PCB design	• PCB design finalized • Materials assembled
4	2024-03-20	In Progress	Software implementation	• Visual odometry module implemented • Object detection module implemented
5	2024-04-10	In Progress	Software implementation	• Path planning module implemented • Motion control module implemented
6	2024-04-24	To do	Prototype refinement	• PCB installed/integrated • Path planning and motion control tuned
7	2024-04-30	To do	Deliverables	• Project is demonstrated to the class. • The project report is submitted.

4.1. Work Split

We have combined work in areas where we have less experience and assigned individual work in areas where we have more experience. The work split is as follows:

- Planning and design (team)
- Prototype build (team)

- PCB design (Keenan)
- Visual odometry (Chris)
- Object/Person detection (Sebastian)
- Path planning (Keenan)
- Final assembly (team)
- Motion controller (team)

4.2. Critical Paths

Critical Path 1:

- Software module integration. Our first critical path is end-to-end integration of our software modules. For our system to function as intended, our data flow through sensing, planning, and action phases must be seamless. We are incorporating our module interfaces over the next two weeks to avoid any last-minute surprises.

Critical Path 2:

- PCB integration and assembly. Our second critical path is printing and integrating our PCB into our mechatronic system. We have a prototype mechatronic system, but a successful project outcome requires incorporating a custom PCB, so we are working diligently to order the PCB as soon as possible. That should give us enough time to re-order if necessary.

5. Budget Estimate

We aim to spend $300 or less ($100 per team member) on parts. To date, we have spent $254 acquiring our chassis, motors, Arduino, motor controller, voltage regulators, batteries, and miscellaneous necessities (wiring, acrylic platforms, tape, etc.). The most expensive components we had to purchase were 4S LiPo batteries and a compatible charger to power the Nvidia Jetson and our motors.

The remaining items to be purchased include the PCB parts: the prints, the 3:8 demuxers, and the LEDs.

6. Risks

6.1. Technical

- #1 - Jetson platform coding: NVIDIA drivers and SDKs
- #2 - Real-time processing and control speed
- #3 - Sensor fidelity (cameras in a relatively visually uniform room)

Our highest technical risk is deploying our software modules on the Nvidia Jetson. Given Nvidia has unique hardware and proprietary software integration, there is a risk that as we develop code individually on our machines, it simply won't work on the Jetson with Nvidia SDKs. To mitigate this risk, we compiled Docker images on the Jetson that contain a replica of its environment. We will launch containers from these images for development environments while coding on our machines. We feel this approach will mitigate most risks. Still, to ensure we have our eyes on this issue, we plan to deploy our codebase on the Jetson frequently through the remainder of the semester to ensure no hidden issues show up while changing our code.

6.2. Programmatic

- #1 - PCB delivery timeline
- #2 - Bandwidth constraints over Spring Break

We believe our highest programmatic risk is the PCB delivery timeline. Given our inexperience with PCBs, and our need to solder a small, surface-mounted demuxer to it, we feel that the two-week delivery time will cause issues if we don't approach it carefully. We designed our PCB this week to mitigate this risk and intend to order it before Spring Break, which will leave us enough time to and order a second PCB if something goes wrong.