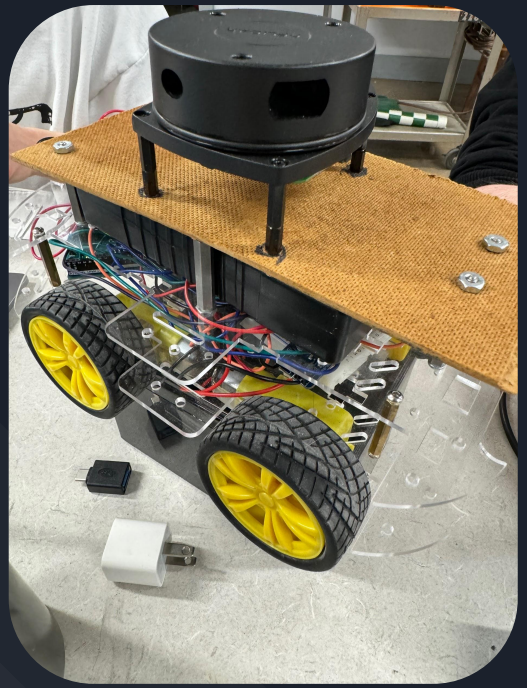
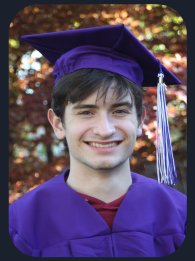
TEAM 8 MDR





John, Ian, Andrew, Harry,
Advisor: Professor Xu

"Thomas the Delivery Train!"

PROBLEM STATEMENT

UMass has the best campus dining among the colleges in the United States. My favorite dining hall is Franklin! Sometimes when we are busy studying we don't have time even for the Grab & Go. Our goal is to build a food robot that can deliver packaged food around campus (library etc.) safely and efficiently.

WHERE WE ARE NOW

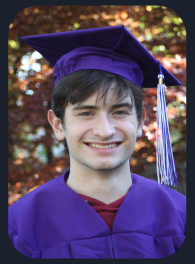
What is finished

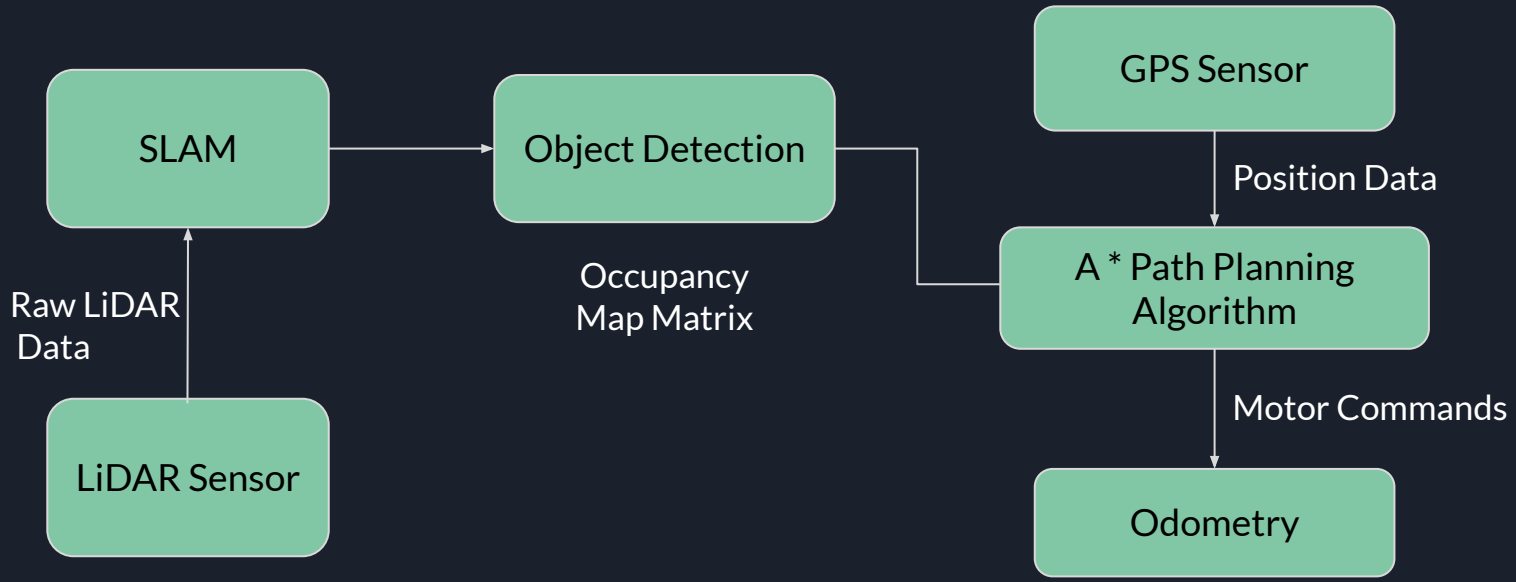
- Small scale:
  - Arduino (command detection & handling, IMU checking, motor driving)
  - Path planning (Returns lowest cost path to goal)
  - LiDAR detection (Scans environment, creates map)
  - Power distribution (motor drivers, power bank)

What we need to work on

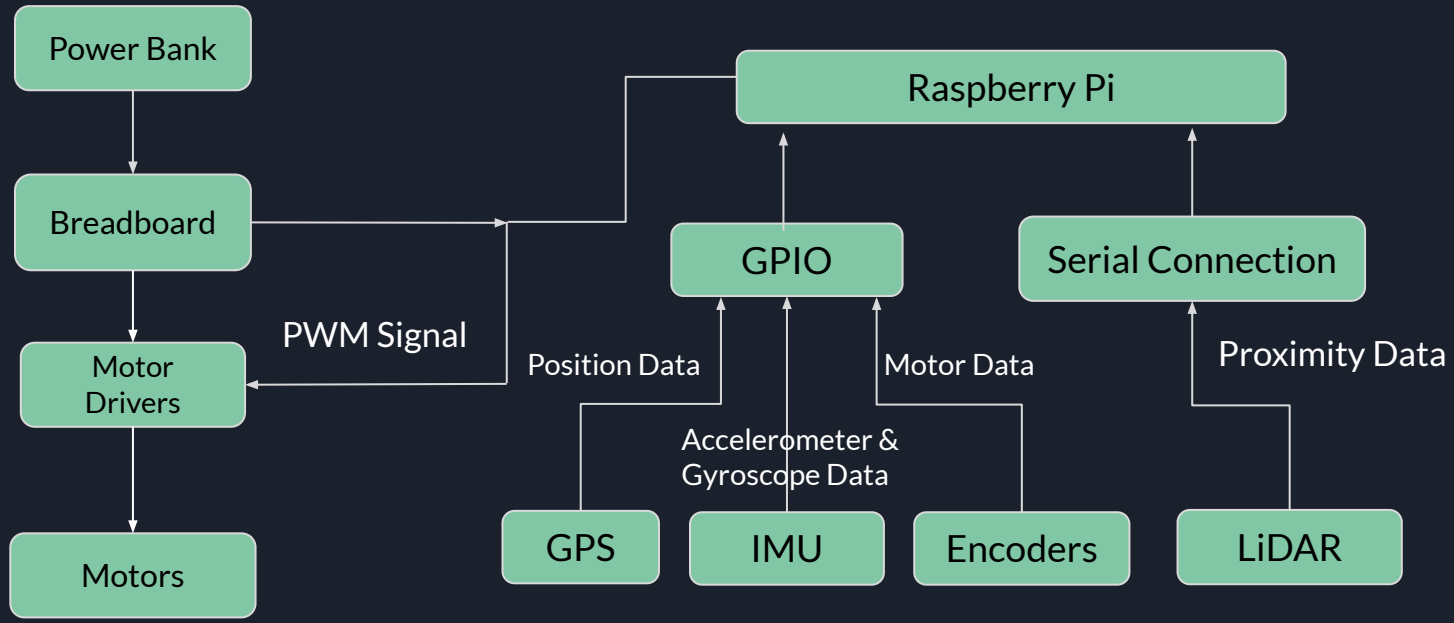
- Large scale:
  - PCB (motor control & breadboard replacement)
  - Power distribution (power bank)
  - Scaling the systems to work with a larger base
  - Odometry (encoders, GPS)

SOFTWARE BLOCK DIAGRAM



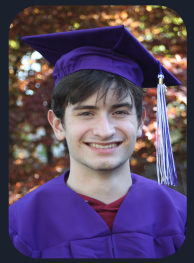


ELECTRICAL BLOCK DIAGRAM



TESTING PLAN

1. Autonomous travel from start location to chosen destination on flat terrain
2. LIDAR map tested to detect obstructions between start and destination
3. Path planning tested in ability to route around obstructions
4. Turns with <20% error to angle given.



MDR DELIVERABLES

John:

- Chassis Design, LiDAR, SLAM Algorithm Implementation

Ian:

- Chassis, Path Planning, Serial Communication

Andrew:

- Motor Control Code, IMU, Power Distribution

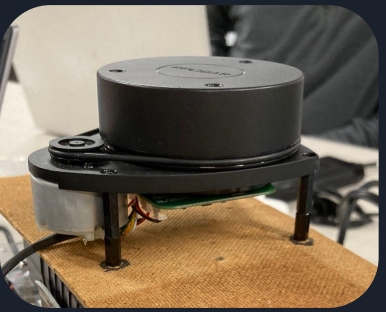
Harry:

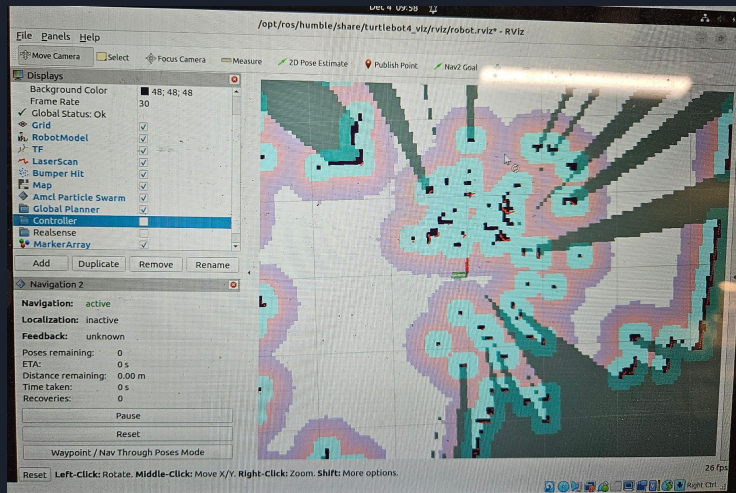
- Breadboard Wiring, Power Distribution

LiDAR & SLAM Algorithm

John:

- ROS2 SLAM Toolbox/Turtlebot4
  - Converts LiDAR Laser Scan data to Occupancy Grid
  - Visualize the Laser Scan Data
  - Simulate the Robot Position

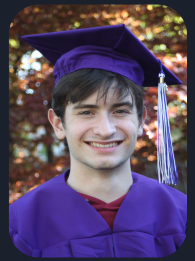




Path Planning & Serial Communication

Ian:

- A* search algorithm written in Python
- Pyserial communication between Raspberry Pi and Arduino
- Planning to switch to using ROS2 navigation stack for next semester

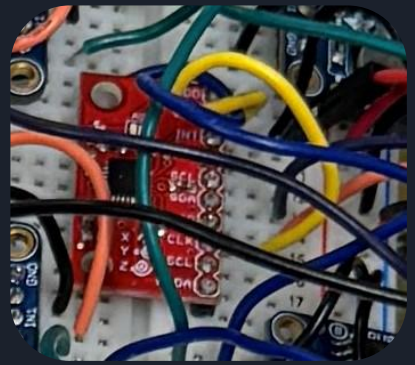


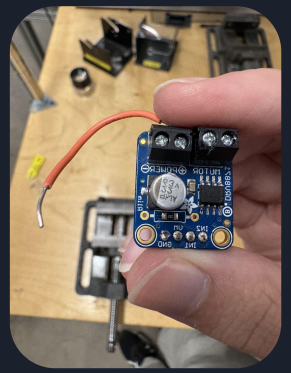
```
ited = []
queue = PriorityQueue()
lastAxis = "y"
position = self.start
node = [position, [["DD 00 00", 1]], lastAxis, 0.0, [position]] #makes node for starting position, starting movement i
queue.push(node, 0.0)    #add start node to queue
while not queue.isEmpty():
    position, actions, lastAxis, cost, coord = queue.pop() #removes top of queue
    if position not in visited and grid[position[0]][position[1]] < 99 : #if node is visited, move onto next in queue
        visited.append(position)
        if self.checkGoal(position):
            #print(position) #for demo purposes, prints goal node when reached
            #print()
            coord += [position]
            #print(coord)#prints coordinates ordered in path
            #actions.pop()
            actions = actions + [["DD 00 00", 1]]
            actions[0] = coord[2]
            return actions
        else:
            """n's in successors are different from nodes, only one action is passed through a successor.
            Nodes in the queue have a list of actions leading up to their position from start
            """
            for n in self.getSuccessors(position): #goes through each possible successor
                if lastAxis != n[2]:
                    #print(n[2], lastAxis)
                    tmp = actions + [[self.computeTurn(lastAxis, actions, n[1]), n[1][1]]] + [n[1]]
                else:
                    tmp = actions + [n[1]]
                tmp2 = coord + [n[4]]
                h = self.calcManhattan(n[0])
```

Motor Control, IMU, Power Distribution

Andrew:

- Arduino Code
  - Forward: “A”, Left: “B”, Right: “C”, Stop: “D”
  - AnalogWrite to PWM pins of motor drivers
  - Add time delay that takes in percentage of pi to determine angle needed
  - IMU reads gyroscope measurements for Left or Right, acceleration measurements for Forward
  - Will continue task if IMU considers distance traveled not sufficient



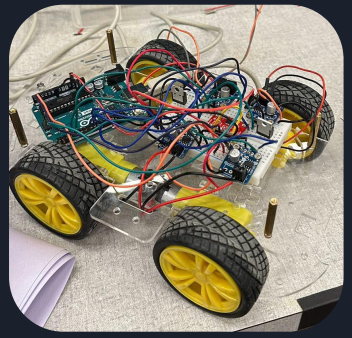


```
driveLeft(int speed, float deg) {
float percentAngle = deg/100;
int timeDelay = percentAngle * 420; // 210 is the time for a 90 degree turn to occur if pulse is 420
int pulse = 130;
if (pulse < 130) {
  pulse = 130;
}
int error = 1;
analogWrite(2, 0);
analogWrite(3, 0);
analogWrite(10, 0);
analogWrite(11, 0);
analogWrite(4, pulse);
analogWrite(5, pulse);
analogWrite(8, pulse);
analogWrite(9, pulse);
delay(timeDelay);
sensors_event_t a, g, temp;
mpu.getEvent(&a, &g, &temp);
float testedAngleEnd = sqrt(pow(g.gyro.x + 0.06, 2) + pow(g.gyro.y - 0.01, 2)) * timeDelay / 1000
```

Breadboard Wiring & Power Distribution

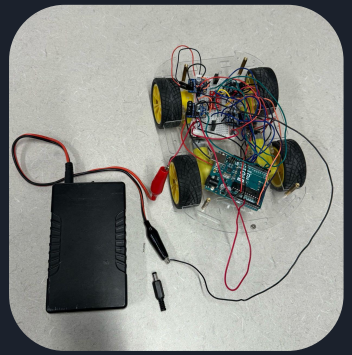
Harry:

- Breadboard wiring
- Hooked up wires connecting the Arduino, Motors, and Motor Controllers to the breadboard.
- Tests of AA batteries, lipo batteries and power supply.



Andrew:

- Power bank
- Arduino takes in between 7-12V from Vin, outputs 12V.
- Run on power rails, connected to motor drivers with 3A.

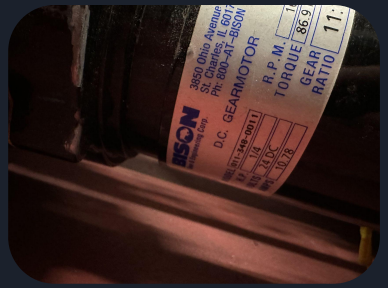


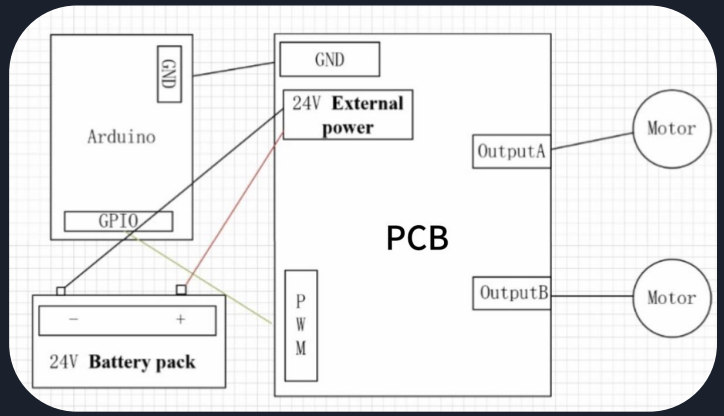
CDR DELIVERABLES-HARDWARE

Harry:

- Power Distribution Solution: 24V, 25A Lithium Iron Battery
- Two DC motors: 24V, 10.78 Amps each
- Installing encoders onto shaft of motors for odometry

- Encoders will be purchased online for 24V motor
- Input: spinning speed of the motors
- Output to the PCB

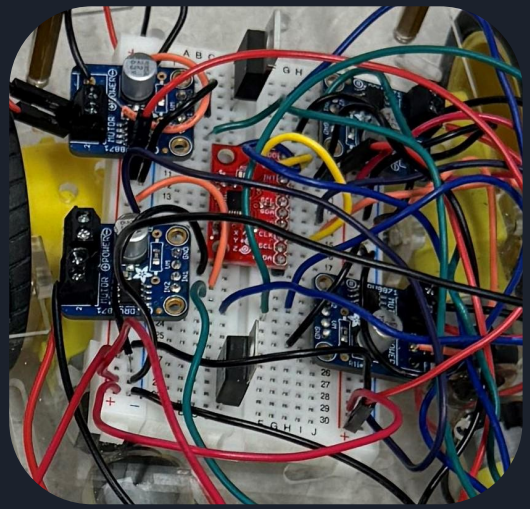




CDR DELIVERABLES-HARDWARE

Andrew:

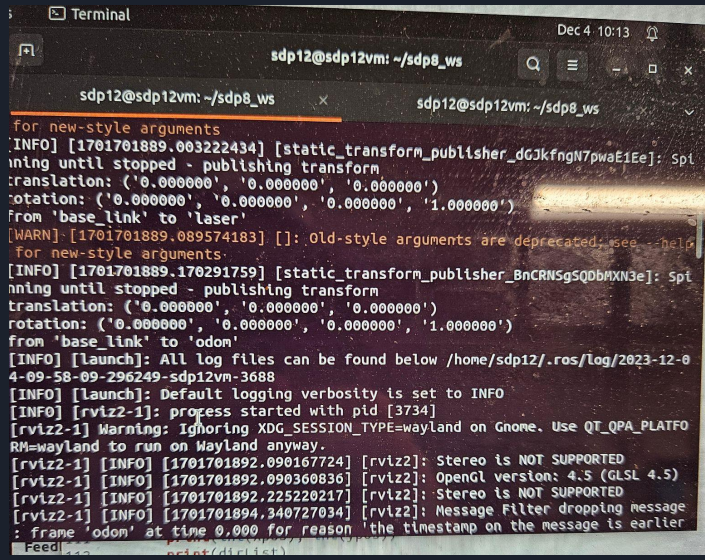
- PCB
  - KiCad to design PCB as breadboard
  - New motor drivers to handle larger current will need PCB to send signals from arduino
  - IMU connections (SDA, SCL, VIO, VDD, GND) through PCB
- Odometry
  - Allow encoder to communicate with IMU to determine position for testing



CDR DELIVERABLES-SOFTWARE

John:

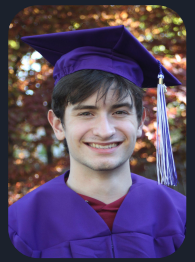
- Integrate Nav2 Stack with current system
  - Generate Goal Node from RVIZ
- Redo the Transform Tree to work with dynamic environment
- Begin Work on Camera



CDR DELIVERABLES-SOFTWARE

Ian:

- Path Planning and Object Detection
  - Path planning should be based on LIDAR occupancy map, with a given goal point
  - The robot should be able to navigate a dynamic environment without collisions
  - A location should be set by GPS that the robot can safely arrive at



PROJECT EXPENDITURES (SO FAR)

Arduino: From M5

IMU (Sparkfun MPU6050): From M5

TT motors: From M5

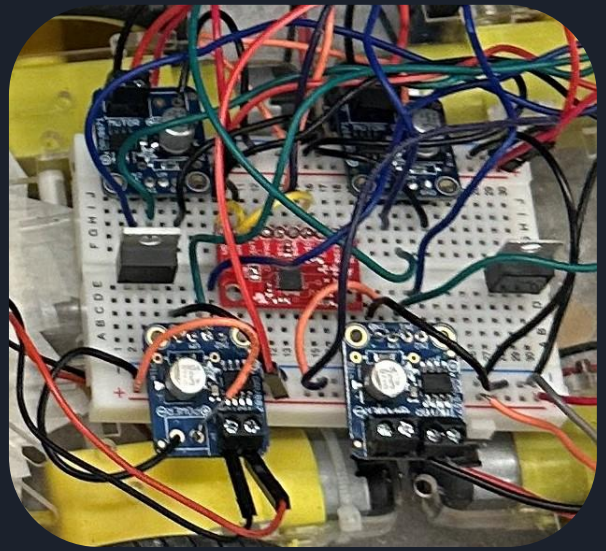
4 Motor Drivers (DRV8871): From M5

Power Bank: Wouter lended

Wheels & Chassis: $25

Total: $25

Budget left: $475



PROJECT TIME MANAGEMENT

| | | |

PERFORMANCE OF MDR PROTOTYPE

Video: https://youtube.com/shorts/weRBlap1zAs

Thank you for your time!

Questions?