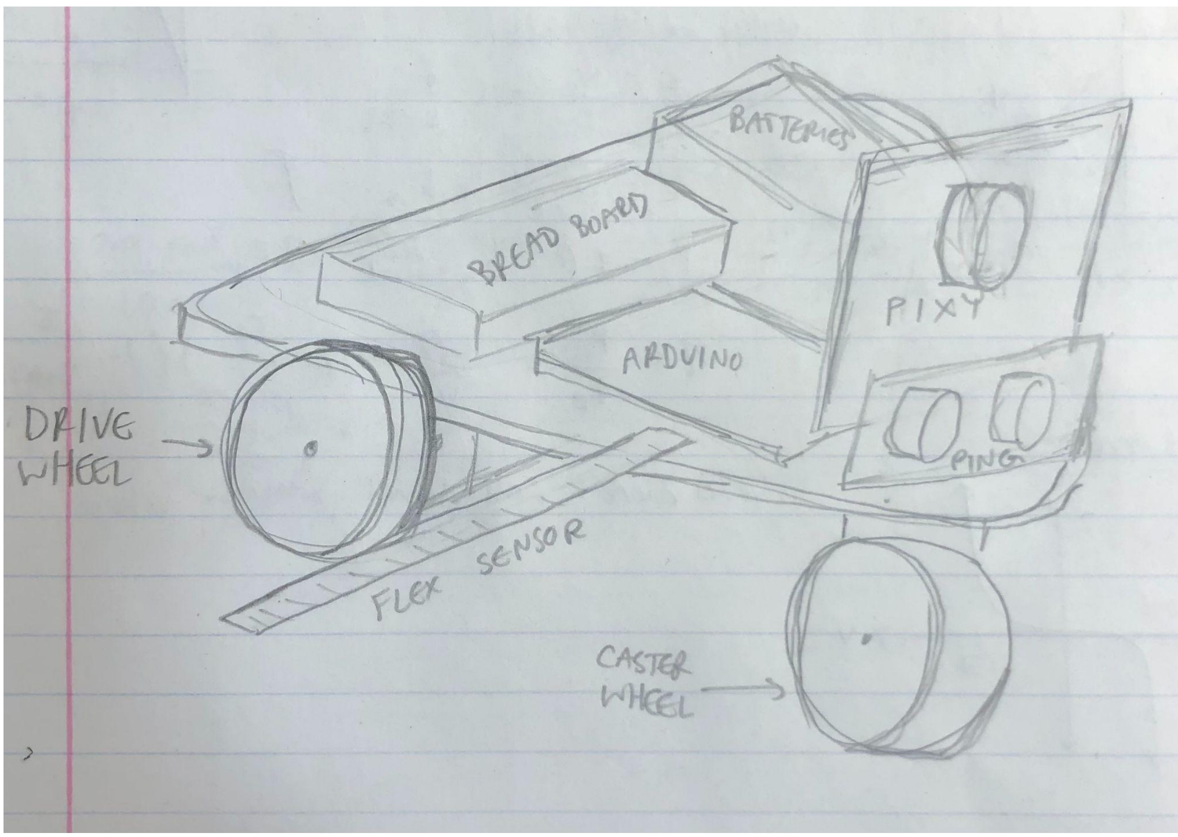
EN.530.421 Mechatronics, Section 1
Lab 3: State Machines and Labyrinth
Trisha Pakkala, Alex Alessi, Nyeli Kratz
3/2/2022

Design Requirements:

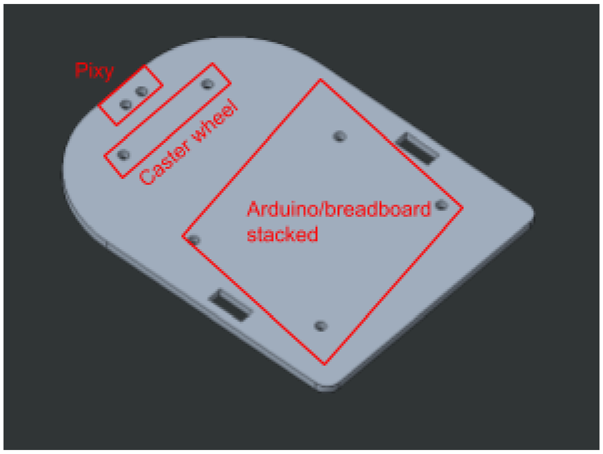
The objective of this lab was to create a robot which navigates a labyrinth using navigation cues from color-coded markers placed on the walls. The robot must be able to drive forwards and backwards, turn left and right. The robot must be fully autonomous and must not be tethered.

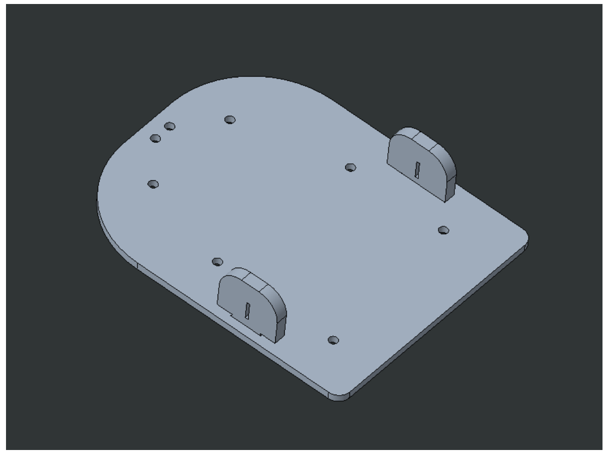
Initial Sketch:



Our initial design used a Pixy camera to read the color cues on the walls of the labyrinth and a ping sensor to detect the distance from obstacles because we did not want to rely on using the Pixy to read the width of colors to sense distance in case the robot approached a wall without color cues. We planned to continuously read the angle of two flex sensors on either side of the robot and use this information to make slight turns when the robot gets too close to the side walls.

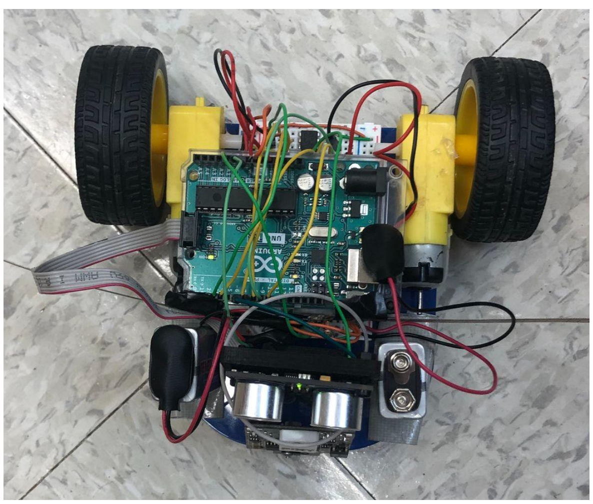
Chassis Design:

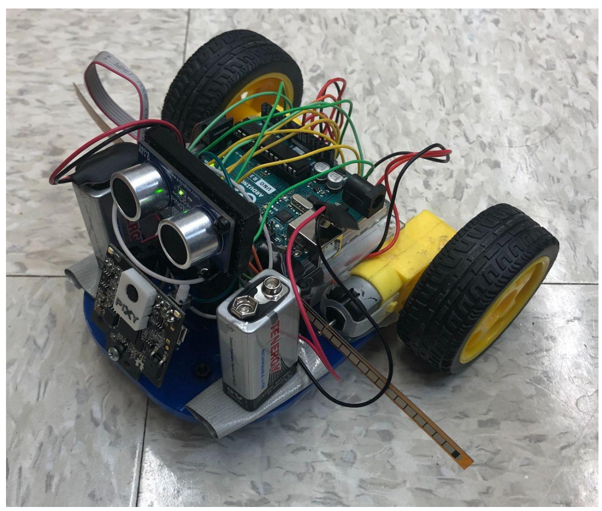


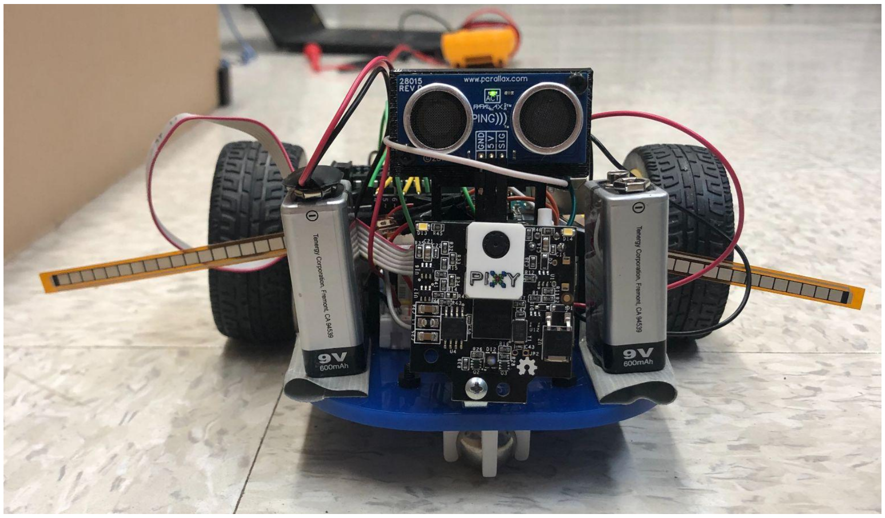


We designed this chassis with the Arduino and breadboard stacked to minimize the area of our chassis, thus minimizing our turning radius. We wanted to minimize the turning radius of our robot because we were concerned about running into walls when turning, particularly with the flex sensors extending out from the chassis. We used a 3D-printed mount for the ping sensor that we were given by one of the TAs from a previous project and modified it with a simple laser cut rectangle to increase the height.

Flex Sensor Design:



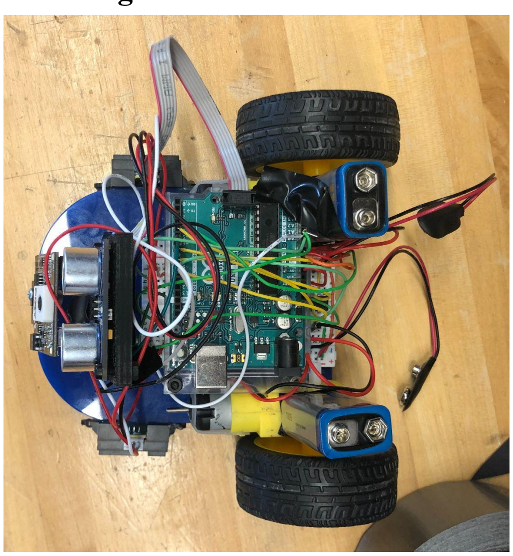


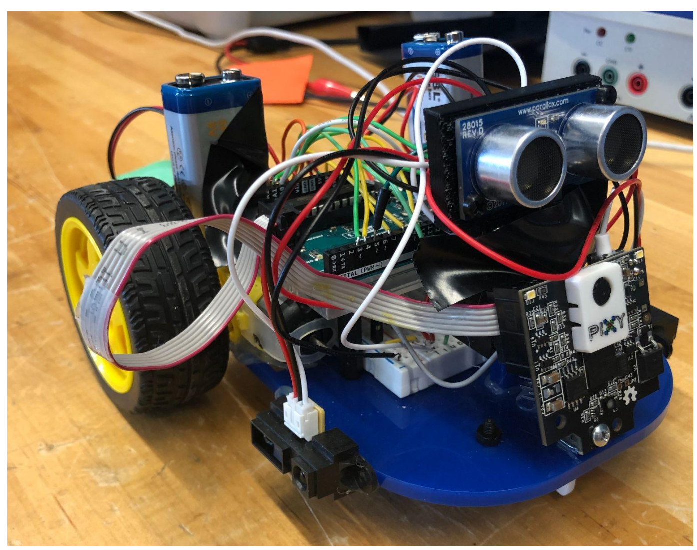


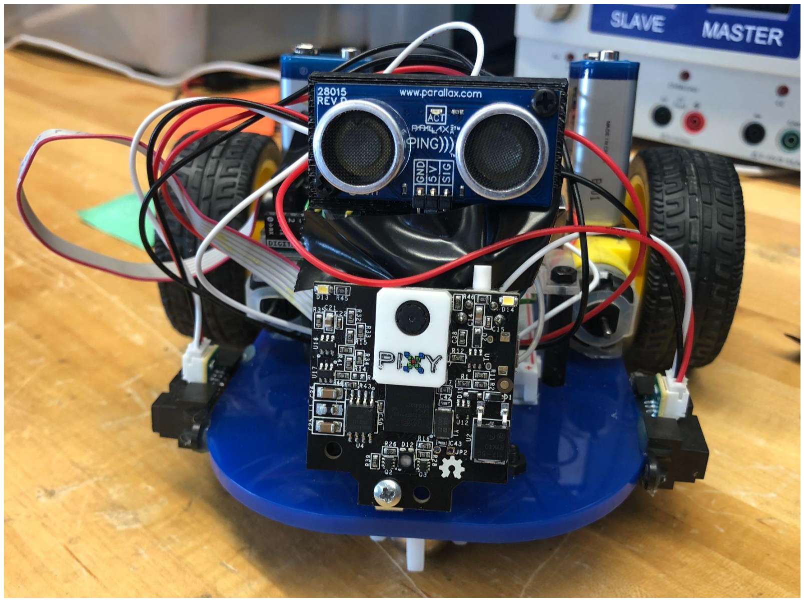
We wired the sensors and motors according to their data sheets and soldered wires to the flex sensors to allow us more control over the placement of the sensors. We wrote code (see appendix and zipped code folder for Mechatronics_Lab3_Version3.ino) which drives the robot forwards, stops when it detects an obstacle close to the ping sensor, and looks for colors using the Pixy when it is stopped. When the flex sensors are flexed beyond 20 degrees, the robot will decrease the power to the left or right motor respectively to avoid running into the side walls.

When testing this robot, we ran into a lot of problems with running into sidewalls and the robot getting stuck on its flex sensors. We abandoned this design because our flex sensor would often physically get stuck on walls when adjusting to the left and right and cause the robot to be unable to continue driving.

Final Design:







After abandoning the flex sensors, we decided to use IR distance sensors to prevent the robot from running into side walls without having to make contact with the side walls. This approach was much more successful because our robot had nothing to get stuck on except for its wheels. The robot was able to successfully navigate the first half of the maze (up until the 180 degree turn direction). It was able to successfully read colors, make turns, and avoid the walls fairly well using the IR sensors. See videos in this google folder: https://drive.google.com/drive/folders/1ttiMEMIAYoGnQXTRUS7uBoRpL07UjK_d?usp=sharing.

However, there were still many problems with this design. Since we replaced the flex sensors with IR sensors less than 24 hours before our demonstration, we did not have enough time to perfect the control scheme of the slight left and right turn functions. As such, our robot was not able to avoid walls well enough to make it all the way through the maze without hitting side walls. If we had more time to work on this robot, we would adjust this control scheme to make the robot avoid the walls better. We would also change our turning control from time-based turning (e.g. turn wheels in opposite directions for 700ms) to a different control method (e.g. turn until the PING sensor reads its max distance and one of the IR sensors reads below a threshold distance).

Summary of Accomplished/Unaccomplished Objectives & Lessons Learned:

We addressed the objectives of each aspect of our design above, but this is an overall summary of our robot's performance. Overall, our robot was able to navigate the labyrinth autonomously and perform turns based on color cues like turn right, turn left, go backward and do a 180 degrees turn. Our robot was able to do these turns, however, things such as variable voltage in the battery caused there to be issues when we tried to autonomously run the robot. As the battery voltage got lower, our DC motor got less powerful and so it messed up the timings for our turn because it then took more time to move the vehicle to complete a specified turn. We had to increase the turning time as we continued to use the battery and then decrease it again when we recharged the battery. This caused issues in how much the robot turned each time. Lastly, while you can see in the video that the robot was able to complete all the turns after recognizing the color, we were not able to get a good video of a complete maze run through.

As for objectives that we accomplished, we again were successfully able to interface the Pixy with the motors successfully and as shown in the videos, the point turns are smooth. The chassis design enabled the robot to make these tight turns without it getting stuck on the walls. We were able to show the robot successfully complete every type of turn. Additionally, the PixyCam was often very finicky in recognizing the color but we were able to train it well to address this issue by increasing the camera brightness of the Pixy and utilizing a “back-up” function that would be triggered every time the robot was 15 cm from the wall and the PixyCam had not recognized a color. This helped us deal with any technical issues we had with the PixyCam. Additionally, while there were a few issues, on the whole, the IR sensors helped deal with drift of the robot very well which was especially an issue because as the battery voltage decreased, drift increased. Most importantly, we got the chassis laser cut, the wiring and assembly, and the code done early so that we spent the last 2 weeks of the project debugging the code and testing it in practice. Our time management was very instrumental to our progress.

Overall, we learned that we should think more clearly about wires or sensors that could get stuck on side walls. This was a huge issue with our flex sensors and one of the reasons why we changed to IR sensors. Additionally, we should not rely on things like delays to perform functions and instead use sensors to attempt to create a closed-loop control system. A more ideal system as talked about above would be to turn until the PING sensor reads its max distance and one of the IR sensors reads below a threshold distance to make sure the robot is turning correctly and into an open channel because in this control system the turning angle would not be affected by battery voltage.

Power Budget:

Based on our estimation, our robot can run for **3.8 hours**.

Component	**Nominal Current (mA)**	**Voltage Supplied (V)**	**Total Power (mW)**	**Time to Deplete One Battery (hr)**
Ping Sensor	35	5	175	28.3
IR Sensor (x2)	30	5	300	16.5
Arduino Uno R3	98	5	490	10.1
Pixy Cam V2	140	5	700	7.1
L293D H-Bridge	35	5	175	28.2
12V Brushed DC Motor (x2)	42	9	756	6.5

We used a total of two batteries in parallel. Each battery operates at 9V nominally and has a capacity of 550mAh. The total power supplied by these two batteries is thus 9.9Wh. Our system consumes approximately 2.596W and can thus theoretically operate for approximately 3.8 hours.

Deviations from Theory:

Although in theory our robot can operate for 3.8 hours before completely draining the pair of batteries used we observed an effective runtime of approximately 20 minutes. This is a result of the following factors:

1. As the batteries are depleted the voltage supplied drops as a function of time. Batteries operating below the minimum voltage of our components are no longer functional.
2. All current metrics are nominal, the actual current consumed by a given component varies. Motor current consumption was measured in a no-load condition, the current drawn by DC motors increases greatly when motors are placed under load.