FOMI Robot Log - Cutting Board & Tupperware | May Bubolz Thomas & Iris Nicholson
So far I've had to switch out the Arduino board at least 4 times. I'm not giving a date for this because it's happened over a long period of time.

I've been working on this robot for a while now, but this is the first log entry.
03/20/2023
I finally printed a mount for my ultrasonic sensor and I'm currently troubleshooting a problem with the right motor. I've determined that it's not the motor itself, I think it's somewhere in the code. But then I ran into another problem, my code will not upload to the board so I'm trying to figure out how to update the program. Which sounds silly, but it's a lot more complicated than I thought it would be.

Okay figured it out. It doesn't like my servo. Which makes sense since I downloaded a library for my code that works specifically with the servo. Hopefully, I just plugged it in the wrong port. [Here's the problem shooting link I used.]

Last update before I go to bed. Servo is now working and it looks like it's turning in response to where my hand is. The ultrasonic sensor is being a little weird but again, the servo was moving accordingly so that's promising. Still haven't figured out the right motor issue but that'll be for tomorrow.

03/21/2023
So the robot is moving its head. I tried using PWM to slow the DC motors down but so far… it hasn't worked. But what does sorta work, is that one wheel will spin if the robot detects an obstacle. Except that it's the wrong wheel that's spinning. So if it senses a corner in front of it with the wall on its left, the right wheel will spin… meaning it's driving itself straight into the wall. Should be an easy coding fix though.

After I fix that the next order of business will be creating a second story to be able to put the battery on the robot. I also want to do some wire management because it's a mess!

It is 9 o'clock at night right now. And I have successfully gotten control of the speed of the dc motors. It turns significantly slower, which makes sense, but I like it. Because this is supposed to example a wheelchair, we wouldn't want to be flying through a turn. Here's the video I watched to learn about PWM, Arduino Motor Control and PWM Signal with L298N H-bridge Motor Driver. Pretty simple, but for some reason it took me a while to figure out how. Now I'm addressing the problem that the robot will detect an object… but drive right into it. It's a simple fix.

```
40  void loop() {
41    analogWrite(enA, 100);
42    analogWrite(enB, 100);
```

```
else if (distances[0]>=distances[1])
  turnDir = 0;
else if (distances[0]<distances[1])
  turnDir = 2;
return turnDir;
```

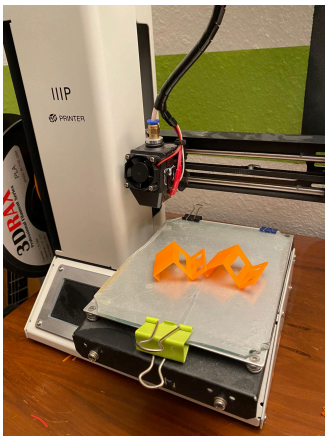
Wrong one: the first else if statement was for turning left and the second was for turning right. I simply switched the number values to make it right.

Right one:

```
else if (distances[0]>=distances[1])
  turnDir = 2;
else if (distances[0]<distances[1])
  turnDir = 0;
```

03/30/2023

Today I didn't do too much. I did hop onto OnShape and made a quick holder for my DC motors because the wood that was holding them split. It took them 1 hour and 43 minutes to print and I'm not sure if they will work or not.



03/31/2023

UPDATE: It did not work. It probably would've but I split the wood… I'm currently printing them again and making the walls thicker. Previously they were 2mm thick and now they will be 4mm. I've also made one of them longer, so that I can still screw it in but avoid the split wood area.

I've been doing some research and I found a tutorial on how to make a DIY LIDAR sensor. On Amazon, a sensor starts at $100, which I don't want to pay. Here is the YouTube video I found: Homemade LIDAR sensor with Arduino & Processing. Hopefully, I'll be able to figure out how to implement it into my entire project but it's a step in the right direction. While I wait for the new holders to finish printing, I'm going to make a list of the things I will need to make it.

I'm also thinking about getting the bluetooth module. It's on sale for $10 so I think I'm going to go for it.

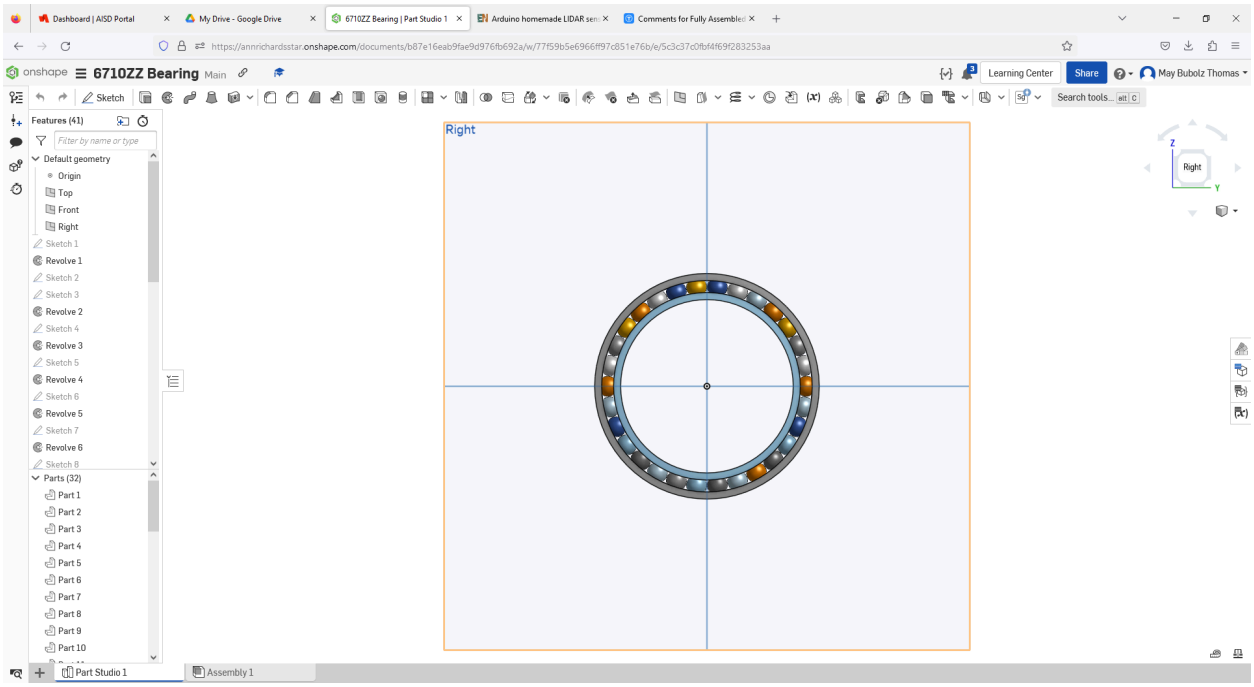
- Infrared sensor
- Arduino Nano
- Bluetooth Module
- Stepper motor - makerspace
- Driver
- Slip Ring
- Hall Sensor - makerspace
- Drill PCB - makerspace
- 100uF Capacitor - makerspace?
- M3 Screws + Nuts - not neccesary
- M3 Inserts - not neccesary
- Magnet

The new mounts for the motors. They work really well and seem pretty sturdy at the moment.

There are 7 pieces that need to be printed.

04/01/2023

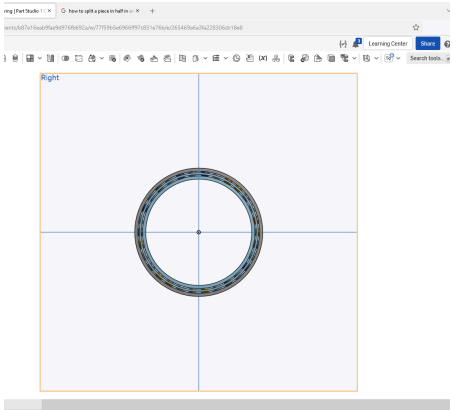
Today, I 3D modeled multiple types of bearings. The DIY LIDAR sensor I'm making requires a 6710ZZ bearing. Which is either really cheap, but I have to wait 2-3 weeks for it, or really expensive ($56) and I can get it in a few days. I did not want to do either option so I decided to 3D model it. After 2-3 hours of work I got this.



I started printing it at 1:30am and we will see if it works tomorrow.

04/02/2023

The bearing does work!! It took a while to take the supports off. I also think, that next time I will sand the balls to make sure they are spherical and don't have any remaining bumps from the supports. There will be a next time, because it doesn't fit. Whoops. Shouldn't be too hard to fix though. After 2 and a half hours. I made the new bearing (hopefully) the right size. I also added an inner ring to hold the balls in place. The inner ring will be printed separately so I can decide if I want it or not when I print it.



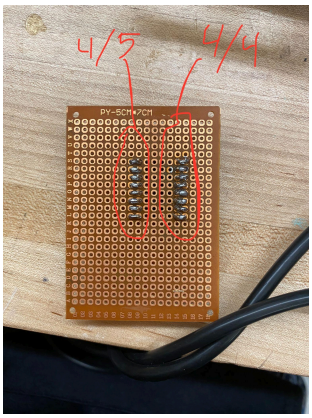
Finally printed! It took some finessing to be able to put the beads into the inner ring. After filing and trimming the beads, I couldn't figure out how to put them into the holding ring with the outer and inner ring. My mom came up with the idea to put a surface in the middle of the rings so the beads would be level to the holes. I cut out a ring, the diameter was big enough that the outer and holding rims could fit in it. Then the holes were level and we placed the beans semi-in them, and then shoved the inner ring in. And now I have a perfect fitting bearing for the LiDAR sensor.

I found this YouTube video that I think can help me in the future.
How to make a voice control Robot Car | Motor Experiment | Science Project

04/03/2023
Today, I began to build the LiDAR sensor. Starting at 10:30 pm, I cut wires, assembled the cap parts, the sensor and slip ring. And soldered the wires into their appropriate spots. After, I used a very thin (I learned my lesson from last time) line of hot glue around the outer ring of the bearing and slid the cap into place.

04/04/2023
Iris and I soldered the voice recognition board today. After that we did some playing around with the wires to try to organize them better and keep them in place. I didn't do too much today, but then, as I was moving the RedBoard around, the connectors fell off. So, a little while later we switched out the board for a new one… which turned out to not take our code for some reason, and after switchin boards two more times, we finally got the code uploaded.

04/05/2023
Today I came early-ish into school and quickly soldered together the stepper motor driver to my PCB board. I'm using a different driver than the video but it has the same pins, just in different locations, so hopefully it should be fine. Here's a comparison on the previous day's soldering on the board, to today's. I can see a pretty nice difference



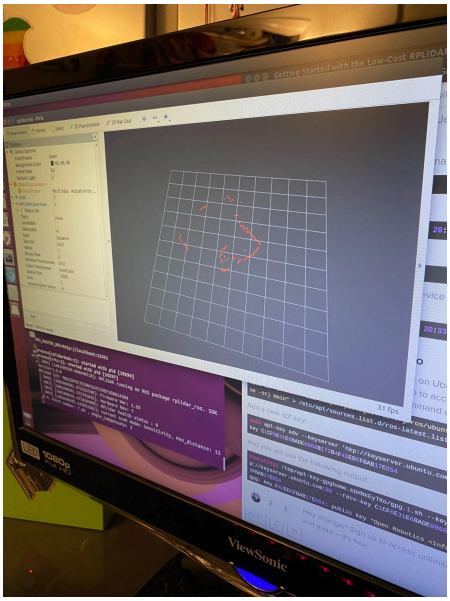
04/12/2023 | Moving to the NVIDIA Jetson Nano

Today, Mr. Oren (our teacher for the Makerspace) said that he didn't think that my cutting board robot would work. Ouch. I'm just kidding, I'm not upset but I am using that to fuel me to prove him wrong. After doing some research, I think it's time to move to a different heart for my robot. Actually, I think it's time to make a whole new robot. I'm planning to use NVIDIA Jetson Nano, which has 4GB of memory, so I can actually save the maps I would be making, plus it's more advanced. It's $150 but luckily I have some money saved up so I'm going to get it. I also found out that you can use Arduino and Jetson, and have them talk to each other. Here's the link: Arduino x Jetson.

Anyways, it's quite likely that it won't turn out and it will be another dead end. But I know it can work, because I've seen YouTube videos of Jetson robots doing what I'm hoping to get mine to do. I also know, that if it doesn't work out, the Jetson can be used for a ton more projects! I'm going to start planning on the structure for my new robot and order the Jetson.

04/16/2023

Oh my goodness, today has been a big day! My dad got home yesterday and helped me out a ton with the project. Both the LiDAR and NVIDIA Nano came today and I was very eager to set them up. Today involved learning about the Terminal. I had never used it prior. But first I had to setup the SD card to put onto the Nano. And once that was setup (took a little bit) I set out to conquer the LiDAR sensor. Which I quickly realized I needed my Dad's help. Following this blog: Hackster.io RPLiDAR, after A LOT of trouble shooting and going back and forth we got it to work. My main issue was… that I was typing each command when I could've been copying and pasting. I am familiar with copying and pasting. …But it never occurred to me to do that with the Terminal because control c and v weren't working. Face palm. Anyways, once my dad showed me it was possible it was fine. Eventually we got to the end and it was saying it couldn't connect to the serial port. (The port the sensor was connected to). But after a lot of wondering, I realized we needed to tell the computer that we could rewrite the usb port. Then it worked like a charm. Now, we have a ton of commands that need to be run in a specific order. So, next up will be making a start up syntax to run all of it so I don't have to type it out.
Here's what our end product was:



04/17/2023

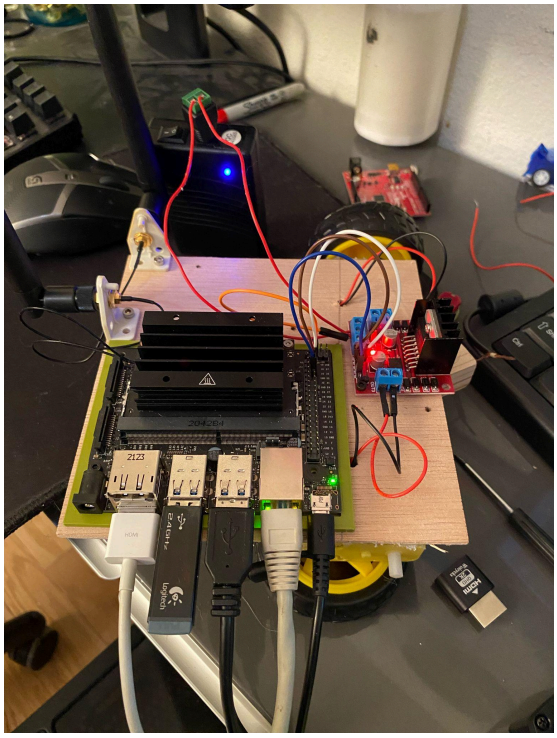
So we got the startup to work after a lot of testing. My dad read on a blog, that in order to run multiple programs from a shell script we needed to use &. Previously, the script was getting stuck on the roscore command and wouldn't carry on to the next ones. But after adding an & sign to the end and a sleep for 5 seconds, it worked perfectly. Next, I'm going to learn how to make a map through ROS and then you can download it. This is the blog I will follow:
https://www.makeuseof.com/how-to-create-a-2d-map-of-your-room-with-lidar/ then from there the idea would be, the map would be downloaded to the bot and sent via wifi and/or Bluetooth, where then the user could "draw" in boxes to resemble furniture or put important points down. Then following this blog, hopefully I can get it to work:
https://automaticaddison.com/how-to-set-up-the-ros-navigation-stack-on-a-robot/
What I really need is a way to import an image into the program and then say, if there's a black line on the image, that's a collision, avoid it, and then if there's, let's say a red dot, that's a labeled point.

4/18/2023

I set up a VNC for my laptop to hook up to. It works but it's laggy.

4/19/2023

Today I built a trashy robot, just to see if I could get the nano to control it and correspond with the Arduino. I never actually got to that part because there were so many issues. In the end, I somehow messed up the password for the entire admin user, and somehow the nano reset the password to something else… except I have no idea and no way of knowing what it changed the password to. So I'm restarting by erasing the SD card, and flashing it again, then setting up the system, the setting up the LiDAR stuff. Again. I'm very frustrated right now.



4/20/2023

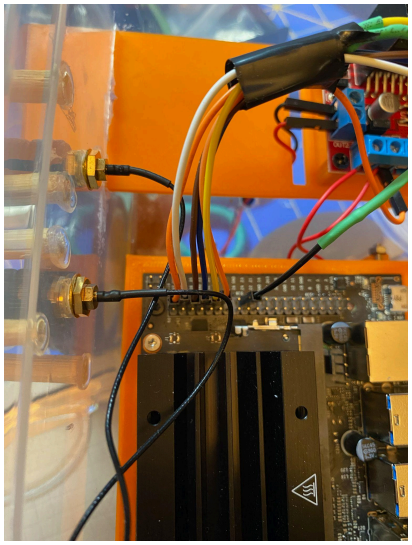
I re-flashed the sd card last night and I'm going to set it up again. Hopefully it won't take as long to set up the LiDAR stuff since I've done it before. I also need to re-setup the VNC stuff. I ordered a dummy HDMI that got here today. Basically, I could only connect through the VNC to the nano if the nano had been previously connected to a direct HDMI connection during the period it was on. (I don't know if that makes sense). We have this dummy HDMI, to trick the system into thinking it does have an HDMI connection, so I won't actually have to connect it to a monitor every time to get the VNC to work. The plan for today is to get everything setup again and then hopefully get the Arduino and Nano to work with each other to do a simple task. I'm not sure what the task will be yet though…

SUCCESS! I can now connect to the VNC without having to have a monitor plugged in to the nano. The dummy HDMI is great! On to the LiDAR sensor.

4/26/2025

I GOT A MOTOR TO MOVE!!!! I found a really sketchy video on YouTube and hey! It lead me right where I needed to go. I'm using the L289N and GPIO pins on the board. I also got another SD card today, so I officially have a backup. And I swapped my SD card out for a faster one.

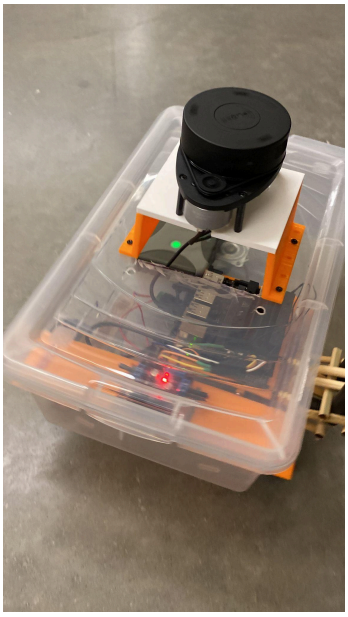
To call a python script in the terminal you put the command: python [name of python script]



5/5/2023

So it's been a hot minute since I've written a log. But here's what I've done:

- built a new body with a tupperware and custom 3D printed parts. I'll insert a picture of it somewhere
- Took me a while to get all the custom 3D parts together but now they're mostly complete
- Just finished the LiDAR mount. Which has a custom stand. The top of the stand has slits in it for a M2.5 screw to go through and to screw into the LiDAR's legs. And then on the bottom of the top part of the stand, it has a rectangular wedge on the ends. The legs are kinda at an angle and have screw holes fo r M3 screws to go into the top of the Tupperware. The legs have a hole in them that are a tiny bit bigger than the wedges. And they snap together very nicely!



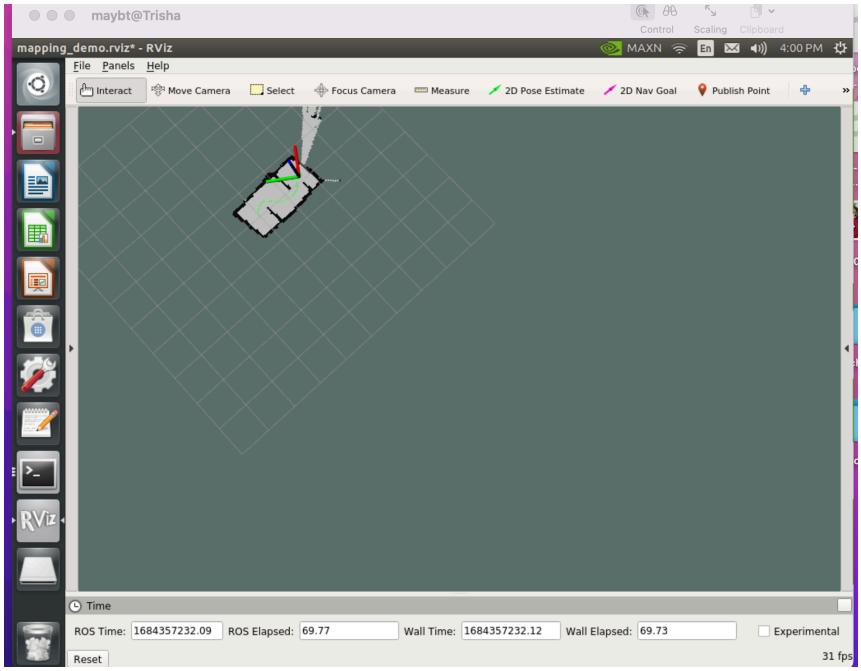
5/15/2023
Today, after 2 hours, I got my Xbox controller to hook up to the board. It took me awhile because the commands the tutorial I followed was using, wasn't working for me. I still don't entirely understand what happened but basically, I was trying to install pygame. Pip install and pip3 install wasn't working, which was strange because I have python3 on my nano. But turns out I somehow downloaded Python3 and Python 2.7 onto the nano. I think it comes with Python 2.7 and I later installed Python3. Anyhow, I didn't realize I had done this until later. I finally came across a forum talking about having Python3 and Python 2.7 and after ONE command. It all worked perfectly.

5/16/2023
Today I spent my time implementing the controller into moving the robot around. After a little bit of trying it works semi-perfectly. You have to jiggle the joystick to move forward consistently. You can't hold the joystick down in a direction and it'll move constantly, it only moves a certain duration. I definitely will fix this later but I decided to direct my focus because our cornerstone presentation is next week.

5/17/2023
Originally, I wanted to program one of the buttons to activate the mapping software, but that proved to be harder. So I took a brainbreak from that and decided to embark on the wonderful journey of using Hector SLAM. It is basically the same thing as ROS and uses ROS still. BUT it creates a 2D map and saves the values. So previously, when I was driving the robot around on RVIZ (the mapping software), it would show the obstacles it was currently detecting. But now with Hector SLAM, it shows the previous obstacles it detected even if it can't detect them directly.

This took me 4 hours. I had to uninstall ALL of ROS and RVIZ multiple times. Which I later discovered that I had had a massive brain fart because I had put some files in the trash… and didn't empty the trash. I

had checked when I put the files in the trash if they had for sure been deleted but couldn't find the trash aaaaaand it said they were deleted. #MacOSuser
Not my brightest moment. And after much more problem solving incidences I finally got it working!



07/20/2023
Hi, it's me again. I forgot to write about how our Cornerstone day went, but it was fantastic. We let the kids drive the robot around and watch it create a map. The professionals that came by, seemed to be impressed with what we had done. The last guy we talked to, was actually working with LiDAR at his job. But instead he was mapping stuff from helicopters and SUVs. He told us to reach out to him after college and he would give us a job! I talked with him about a problem with the mapping, where the maps were drawing over themselves. It's because the robot doesn't know where it is. So, he suggested adding GPS, IMU (roll and pitch, to see if there's ramps, etc.), DMI (# of rotation of tires), some kind of LiDAR processing (he said some drones come with it).

It's been a bit since I've worked on it, so I decided to get back into it with something "easy". I just ordered a Raspberry Pi Camera Module V2, which is compatible with the Nano. I'm not sure how to use the LiDAR for obstacle avoidance so I'm going to start with the camera. Luckily, there's a ton of videos and forums on it (unlike the LiDAR). Figured this would be easy, plus important for the robot, as autonomous movement is a huge part of the bot.

I saw this thing online, about college scholarships. Now I'm getting wayyyyyy ahead of myself, but it's a competition where you make a video of a STEM project you're passionate about. Maybe I can enter this? I'm excited to get back to working with Trisha (the robot). I lost the battery chargers (face palm). I won't update again for a bit… going out of town!

SIDE NOTE: My mind is everywhere, but I'm thinking about trying a new body for the bot. I want to try four wheels and to make the body taller. That way the LiDAR can detect more (it sees a lot of furniture legs right now).

P.S. We took apart the cutting board robot. RIP my first robot.

07/21/2023

Okay, so I lack a lot of patience. Or rather… I suck at researching sometimes. I got a little excited to buy the camera yesterday. Which, it is on its way. But I'm thinking for now, using the ultrasonic sensor that is compatible with the Arduino would be better. I know how to code it. Doesn't seem too hard and it works kinda well (?). I will still use the camera. I found a forum about the sensor: https://forums.developer.nvidia.com/t/using-hc-sr04-ultrasonic-sensor-with-jetson-nano/78861/2
I'm definitely going to use this to help me get the Arduino working with the Nano. Since I don't have an Arduino or Ultrasonic sensor at the moment, I think I'll start focusing on the really really daunting task of figuring out how to get the mapping to work. I want to TRY to make my own program that will take the values from the LiDAR and draw it out (no idea how I'll do that). Maybe then I could make my own processing thing to process all the dots, then incorporate GPS, then IMU, then DMI, then… I'm getting my hopes up that this will work out… We'll see how it goes.

How do you take real time values and turn them into a drawing…

I found the video that had given me this idea freshmen year. https://youtu.be/xMMxiAYmLWE?t=307
These links have helped me so far with trying to get data from the LiDAR myself:

- https://github.com/NikodemBartnik/LIDAR-Robot/blob/main/Arduino%20Code/lidar/lidar.ino
- https://stackoverflow.com/questions/62031686/reading-a-serial-signal-from-a-usb-port-with-python

08/01/2023

I'm back! Today, I went to the makerspace with Iris and we got a few new things. We have decided to start on a new body. Mr. Oren (the makerspace teacher) told us it would be good and I agree. We're using stronger motors too. He gave us multiple but I'm thinking we go with the AndyMark NeveRest 20 Motor. It has a 20:1 ratio, lots of torque, and they're pretty beefy. I got my ultrasonic sensors for an idea I have about autonomous driving but I'm redirecting my attention to learning about these new motors and the new motor driver that came with them. The motordriver is a Sabertooth 2x12. The Sabertooth takes 12V and distributes 12V to the motors. 0V = ground, 5V = well, 5 volts (I assume it's outputting 5V), and then I'm still trying to figure out what S1 and S2 are. But I found this link, https://se.inf.ethz.ch/people/wei/robots/arduino_sabertooth2x12/sabertooth.html, which will be super helpful in the future. The motors we have now, have encoders. Basically channel A and channel B, 5Vs to power it, and ground of course. Channel A will lead B when the motor is rotating forward and vice versa when it's going backwards. I'm thinking the wires for the encoders can go straight into the arduino but I'm not a hundred percent sure.

08/05/2023

So for the past 3 days I have been struggling to get the Sabertooth motor driver to work. We've had all kinds of problems with the motors. I didn't realize I had grabbed two motors with a lot of torque and two motors with a lot of speed instead. I later learned that, that was what was causing the motors to go at different speeds (face palm). We've been three different batteries because the motors are drawing a lot of amps. Anyway, today my dad helped me figure out how to use our new battery. Which does 6Ah and a little over 12V. We figured out that the motorcontroller can't support 4 motors because it can't how much draw is happening. But it can handle two motors. So, I'm going to use two different Sabertooth controllers and then it'll be pretty easy to code to go. So next step after I get the two other motors and second motor controller will be encoders.

03/12/2024

Okay so I forgot to update the log on… a bunch of things. Here are the changes since the last log:

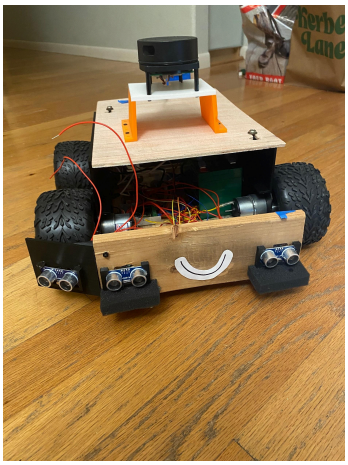
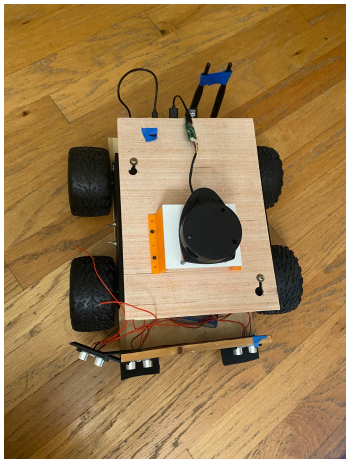
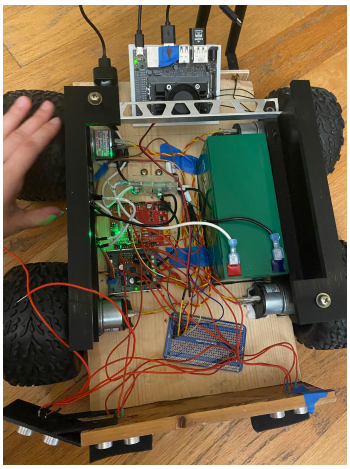
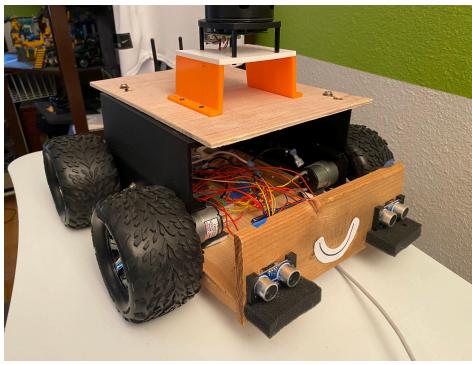
1) I got the Sabertooth to work. Two motors for each port so it's acting like it's controlling one motor but actually controlling two. I don't know if that makes sense but basically, I'm able to have 4 motors with one Sabertooth controller.
2) New body.
3) Pretty good at obstacle avoidance.
4) Beefy battery (12V LiFePO4)
5) Huge wire mess.
6) Such an improvement from all the robots that came before.

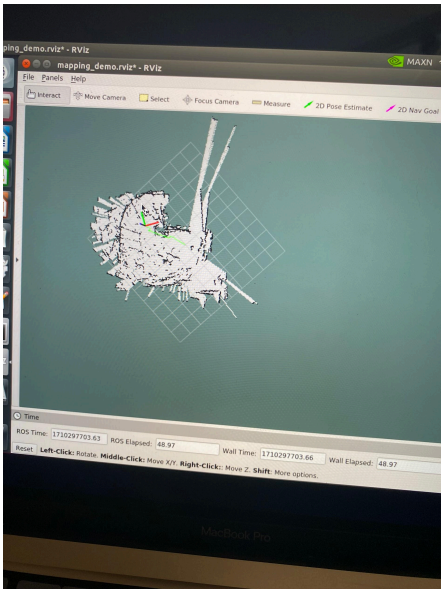
I made this log with the goal of writing down the problems I faced and writing how I solved them. Because I forgot to do that for the past almost 6 months, here we go.

- Sabertooth was fixed only after catching one on fire and busting a second. Third time was a charm 🙂. Mr. Oren found some old code people had used to make an RC car with the controller and gave the code to me and all of a sudden it worked. There are also little switches, called DP switches, that set the controller to certain modes. I was able to copy the right switches from a different controller that was actually working. Also, the online guides and manuals were not very helpful for the dip switches.
- The new body would be best described with pictures. Which I'll put below. But basically, I took an old RC project car body and completely turned it into something else. Very square, very chunky, and of course, there had to be funky materials on the bot (wood). Giving this bot its nickname… Woody! Anyways, it has very beefy tires, the kind that are on the monster truck RC cars. And a smiley face.
- The obstacle avoidance is something I'm pretty proud of. And it's still a work in progress. Basically I took the code that I used for the cutting board bot, and repurposed it so it would work with another ultrasonic sensor and wouldn't spin on a servo. Works pretty well, except it hits chair legs right between the two sensors. And it also gets stuck when an object hits its wheels but I'm fixing that as I type this up.
- The big battery was not necessary but while trying to figure out what was wrong with the Sabertooth, we thought it was the battery supply. So we bought a very big one.
- There are wires everywhere.

Now for the actual update for today:

First I had to get the VNC back to working. For some reason it decided to not work but after a little bit of ‘hacking’ as the website I followed called it, it started to work again. Then after that I discovered that if I turned on the switch supplying the NVIDIA Jetson Nano with power and then turned on the switch supplying the Arduino and motors, the Nano would experience a voltage drop, shutting it off for a second. But long enough for it to well… turn off. So my dad came up with the idea of connecting a capacitor to the battery in parallel and connecting the positive end to the switch pwr and the pwr to the nano. Now it doesn’t shut off. Yay! Okay so once that worked, I was able to get the LiDAR to map while the robot autonomously drove around… but the map was drawing over itself. This was a result from the robot moving too quickly so I slowed it down. Sorta solved the issue, but turning is still too fast. So that’ll be something to fix in the future. Lastly, I’m printing parts to attach to the front of the bot so that I can add two more ultrasonic sensors to improve the obstacle avoidance. And that’s it! I’ll add pictures below and hopefully I’ll get more
on to updating the log.



← Overlapping mapping.

Some of the code for obstacle avoidance →