



AUTONOMOUS ROBOTICS

Design Book



Milton Hinnant

Jan 18th

"Square Bots"

Build guide

Parts

36-tooth gear

60-tooth gear

Metal Chassis

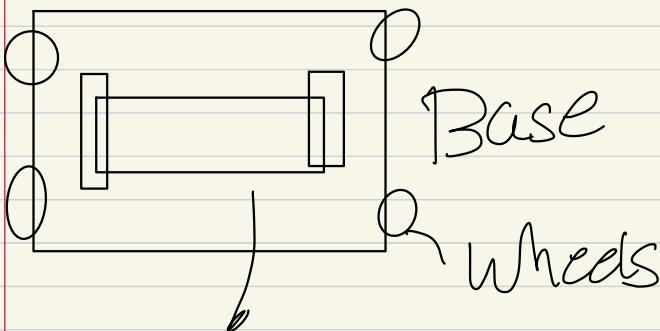
Code

pre

loaded

on

micro -
controller



Micro Controller

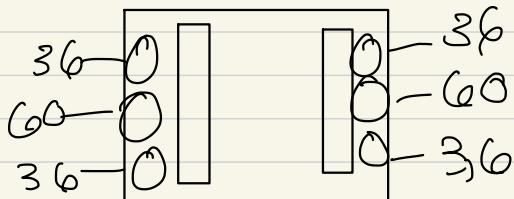
Today we started with the
the initial build of the
square

Following the design guide
we got the metal chassis
built.



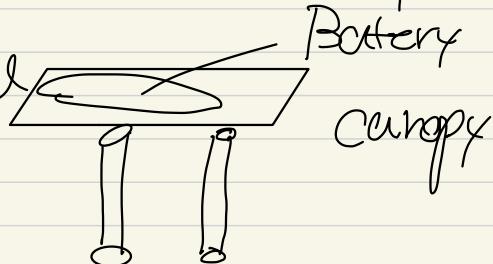
added the motors next
put them in the middle
of the metal chassis
next metal shafts.
added collars + 60 tooth
Gear. have a left + right

Side. Put everythy to
Gether



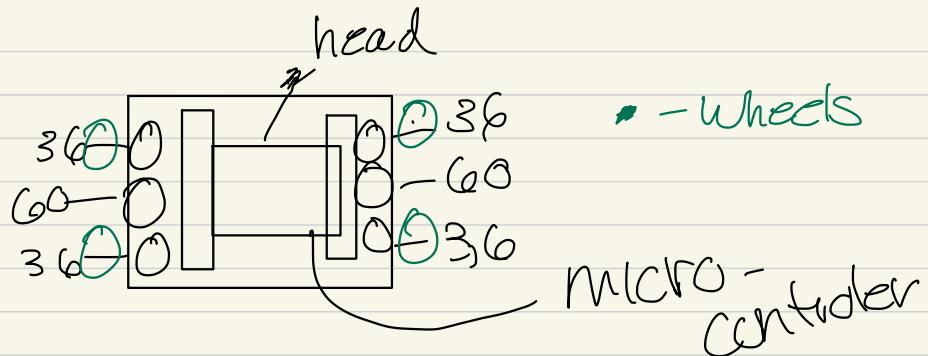
adding the battery canopy

We constructed ~~the~~ Battery canopy
the canopy



added on top of Base

adding the Wheels to the
outside now



Wheels add on the cogs

adding micro controller. Pot

the micro controller the
wrong way.

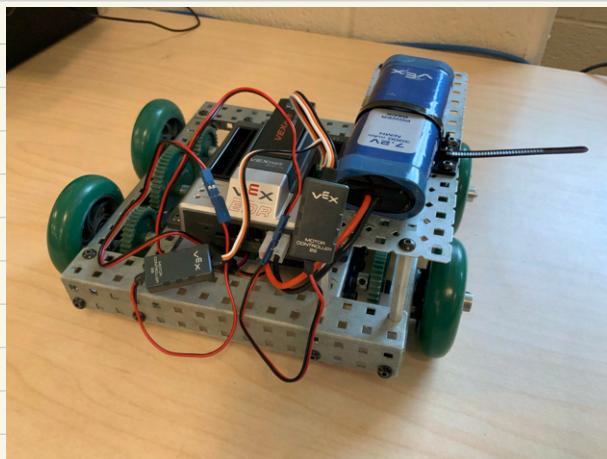
adding battery to canopy.

Installing battery canopy to
base. time to wire
every thing.

the motor wires need an
adaptor. the direction
matters!

Wired the battery up.

After switching the motor
wires around we are done
with the build.



Jan 20th

teleop

- joystick

- Bump Sensors



pairing the Micro Controlle
With The controller. Issues

With the pairing have to
re try it again. Done

We can now move our
Robot around with the
remote. tele-op done

Continue

Autonomous Mode

Bump Sensors



When pressed it does something

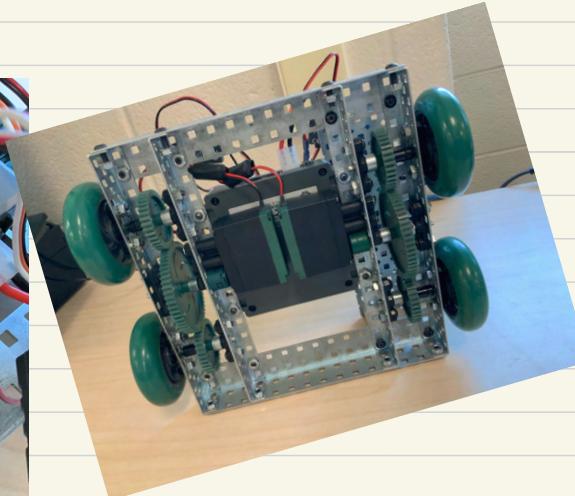
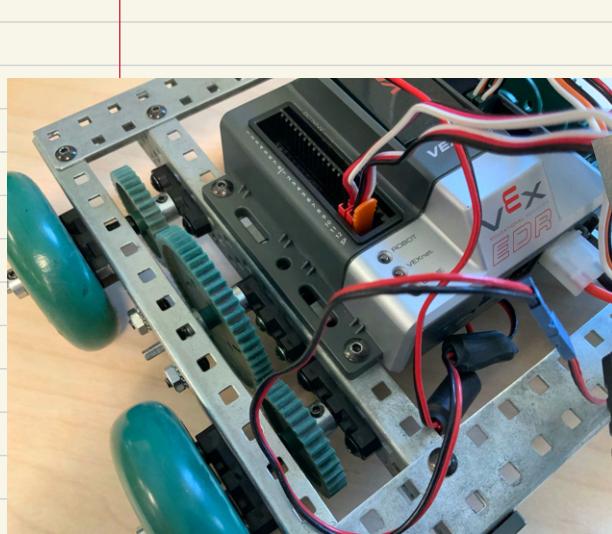
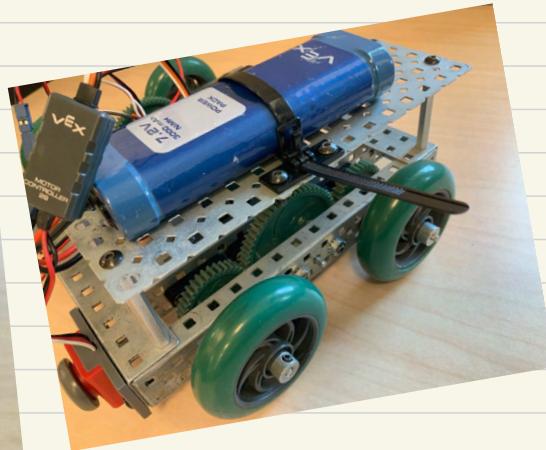
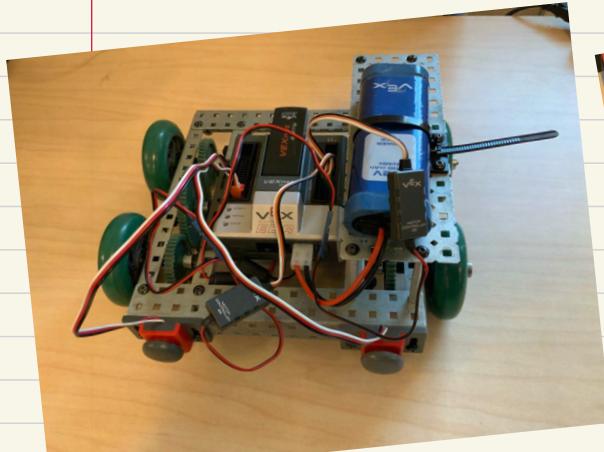
We add two to the front left and right of the base.

Wiring them to the microcontroller. dig.10, 11. adding the jumper now done! we are having an issue with the bumpers.

bumpers put in wrong spot

issue resolved

done



Jan 25th

Hailstone Numbers

$n_{i+1} = n_i / 2$	if n_i is even
$n_{i+1} = 3n_i + 1$	if n_i is odd

function return $n/2$
return $3 * n + 1$

```
hailstone(int n){  
    if(n % 2 == 0){  
        return n/2  
    }  
    return 3 * n + 1  
}
```

hailstone Sequence

7, 22, 11, 34, 17, 52, 26, 13

"Light Controlled Motor"

Motor that Speed up depending
on the Intensity of the light
Source.

light sensor needed

Added a light sensor to the
Micro controller

Most of today
is programming.



Pseudo code

int sensVal

int adjusted sensor value

int motor Speed

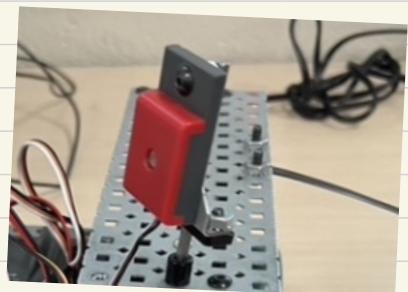
adjusted sensorValue()
sensorValue()

while loop through
pull values for

between -127 - 127 light convert
-127 - 1000 light number to
invert values
-1000 - 1000 motor speed
S

our light is not working consistently
we can not get our micro computer
to connect with the computer
consistently

got new batteries in flashlight.
done with part two



Jan 27th

An Anticipatory Obstacle Avoidance

- Ultra sonic sensor
- Micro controller
- port 2

Back away for obstacles

Computer will not connect to

Micro controller been having
this issue for 30 minutes

15 mins left got it working

Jan 28th

yesterday we mounted the
Ultra sonic Sensor then
we had computer issues most
of class.

pseudo Code

While(true)

 While{ get starting distance

 "do something when number

 is small"

 move motors

 }

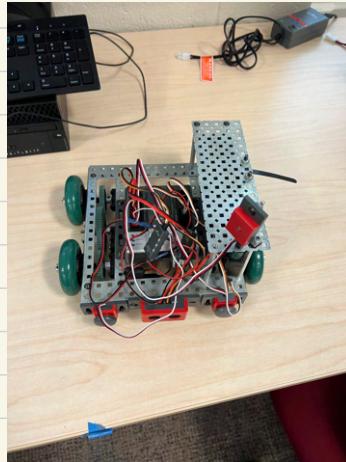
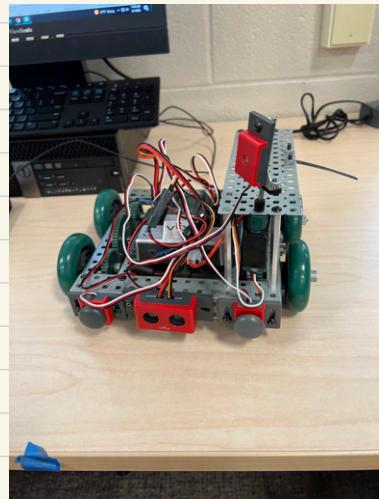
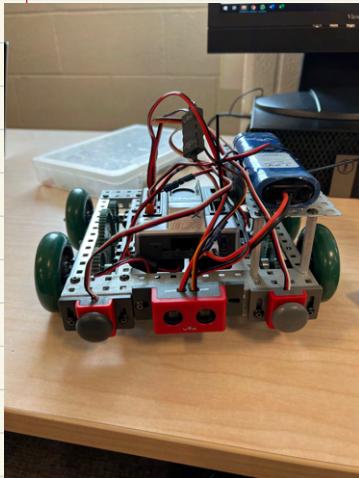
 Bump Sensors Check for press

 ?

Between -1 & 50 while

Using Bump Sensors

done



Move camera to full depth.
feb 1st High + low freq chirps - ultra

lecture notes modulate chirps

Part Build a light sensor that can figure out the direction

- flash light higher # turn to direction

Part 2

- how accurate is the gyro sensor.
- Which direction is the beam painted If statement

Part 3 Photo axes

- Optical shaft encoder

- When wheels move so does it

Wandering Mode

Keep Bumpers

Part 4 - Optical Shaft encoder

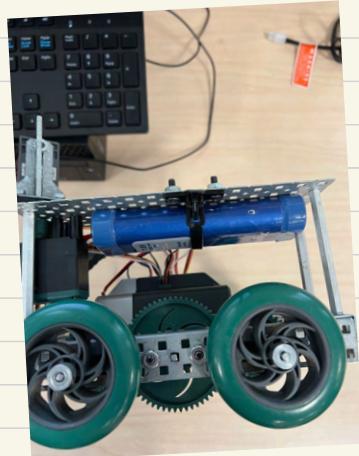
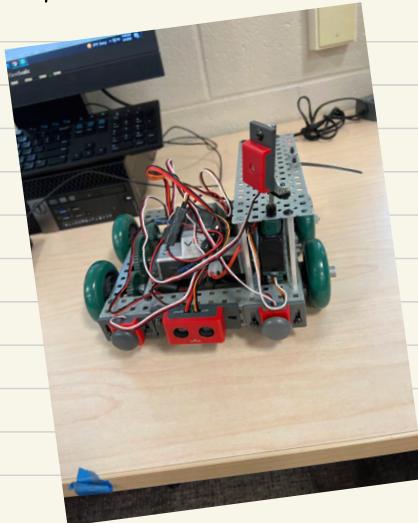
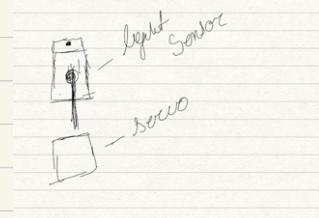
positive Counter Clock wise

negative Clockwise

Feb 3 - 11

Active Directionality Sensing
we wanted a light sensor on
top of a serv

we added this
to the front of
of our Robot and moved
the battery to underneath the
Canopy,

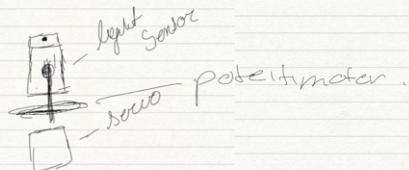


With the code we started with the initial position of the servo the made a while loop through all of the positions the servo can move between $-127 \leq 127$ So we can find the position that finds the brightest light source and stopping once it finds the source.

Part 2

We add a potentiometer to the bottom of the light sensor

Servo combo



today we are gathering the angles from the potentiometers with the light source

By taking the servo values we plotted the angles.

$$y = 0.68x - 39.786$$

holding the light at an angle

$$40^\circ \quad 31 - 1006$$

$$80^\circ$$

measured at
12"

$$110^\circ$$

$$130^\circ - 126 - 2896$$

distance
15"

$$60^\circ$$

$$9^\circ - 1397$$

$$73^\circ$$

$$77^\circ - 1534$$

$$124$$

$$135^\circ$$

$$125^\circ - 2311$$

$$142^\circ - 2545$$

$$136^\circ - 2494$$

184.3

$$60^\circ$$

$$99 - 2144$$

15

$$100 - 1861$$

10

$$105 - 1956$$

15

$$95 - 1774$$

15

$$95 - 1798$$

$$40^\circ \quad 37 - 1114$$

Best one

$$36 - 1116$$

$$36 - 1092$$

36.3

$$110^\circ - 124$$

$$130^\circ - 2406$$

$$123^\circ - 2304$$

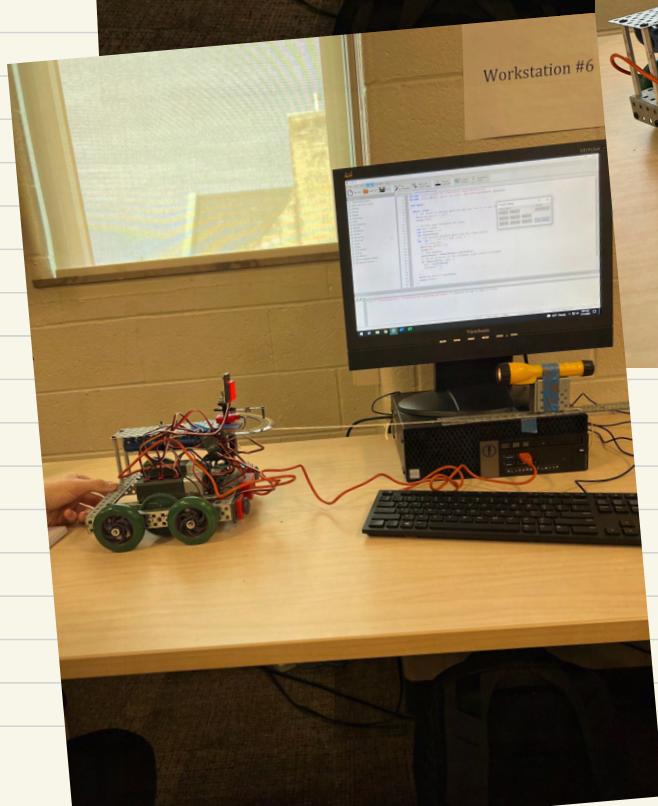
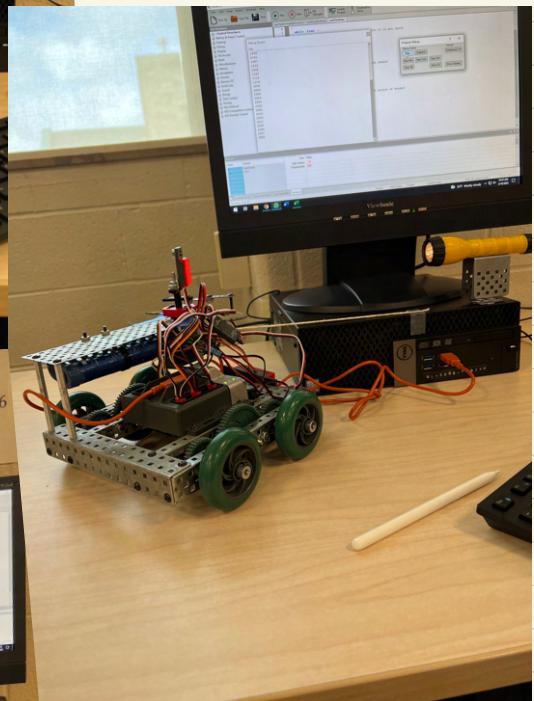
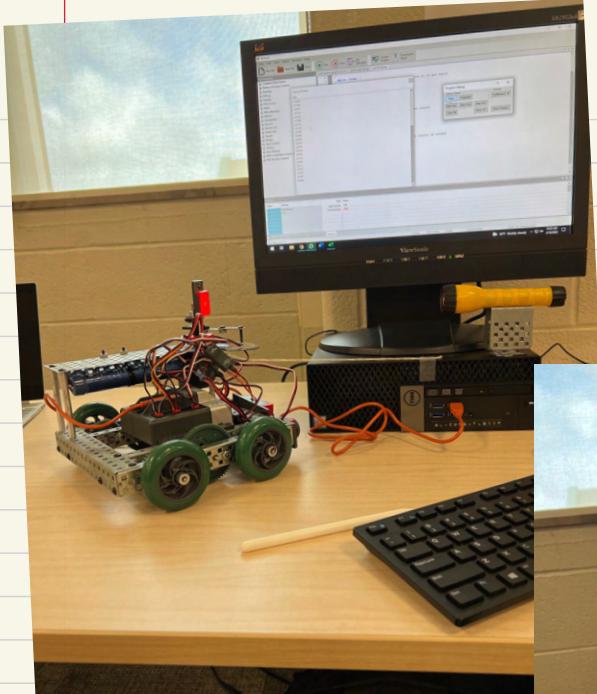
$$126^\circ - 2294$$

$$120^\circ - 2193$$

$$118^\circ - 2190$$

$$119^\circ - 2191$$

We made a tool on the protractor to get the angles used screws & brackets



today we continued with the same calculations

the start.

90°

89
89
94

1708
1724
1779



80°

89
92

72
71
75
77

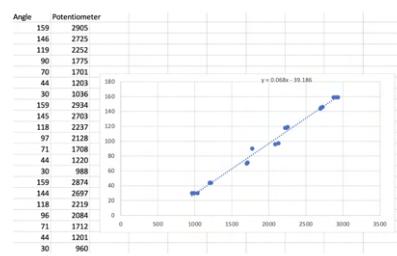
0°

1718
1746

1464
1449
1507
1514

dist 17.5

We tested the two enclosures and got better readings from the no enclosure. going that way.



excel 1. / Line Graph

10°	Angle of Light	Angle of Sensor	Potentiometer Reading	Average Sensor Angle	Error	Error Squared	Sum of Squared Error	No. of Observations	Sum of Squared Error / No. of Observed Error
10	40	34	1154	1152	1480	36.00000000	1.29600000	11	0.32720000
10	49	43	1157	1157	1520	73	5329	11	484.500000
10	58	51	1161	1174	1700	49	2401	11	445.5455
10	67	59	1165	1174	1700	54	2916	11	265.0909
10	76	68	1169	1174	1700	59	3481	11	317.3636
10	85	77	1174	1174	1700	64	4096	11	372.3636
10	94	86	1178	1174	1700	69	4701	11	427.3636
10	103	95	1182	1174	1700	74	5476	11	486.0000
10	112	104	1186	1174	1700	79	6241	11	553.7273
10	121	113	1190	1174	1700	84	6945	11	622.2727
10	130	122	1194	1174	1700	89	7641	11	691.9091
10	139	131	1198	1174	1700	94	8344	11	761.5455
10	148	140	1202	1174	1700	99	9049	11	831.7273
10	157	149	1206	1174	1700	104	9756	11	901.4545
10	166	158	1210	1174	1700	109	10461	11	971.1818
10	175	167	1214	1174	1700	114	11169	11	1040.9091
10	184	176	1218	1174	1700	119	11876	11	1110.6364
10	193	185	1222	1174	1700	124	12584	11	1180.3636
10	202	194	1226	1174	1700	129	13291	11	1250.0909
10	211	203	1230	1174	1700	134	14000	11	1320.8182
10	220	212	1234	1174	1700	139	14709	11	1390.5455
10	229	221	1238	1174	1700	144	15416	11	1460.2727
10	238	230	1242	1174	1700	149	16123	11	1530.0000
10	247	239	1246	1174	1700	154	16830	11	1600.7273
10	256	248	1250	1174	1700	159	17537	11	1670.4545
10	265	257	1254	1174	1700	164	18244	11	1740.1818
10	274	266	1258	1174	1700	169	18951	11	1810.9091
10	283	275	1262	1174	1700	174	19658	11	1880.6364
10	292	284	1266	1174	1700	179	20365	11	1950.3636
10	301	293	1270	1174	1700	184	21072	11	2020.0909
10	310	302	1274	1174	1700	189	21779	11	2090.8182
10	319	311	1278	1174	1700	194	22486	11	2160.5455
10	328	320	1282	1174	1700	199	23193	11	2230.2727
10	337	329	1286	1174	1700	204	23800	11	2300.0000
10	346	336	1290	1174	1700	209	24507	11	2370.7273
10	355	345	1294	1174	1700	214	25214	11	2440.4545
10	364	354	1298	1174	1700	219	25921	11	2510.1818
10	373	363	1302	1174	1700	224	26628	11	2580.9091
10	382	372	1306	1174	1700	229	27335	11	2650.6364
10	391	381	1310	1174	1700	234	28042	11	2720.3636
10	400	390	1314	1174	1700	239	28749	11	2790.0909
10	409	399	1318	1174	1700	244	29456	11	2860.8182
10	418	408	1322	1174	1700	249	30163	11	2930.5455
10	427	417	1326	1174	1700	254	30870	11	3000.2727
10	436	426	1330	1174	1700	259	31577	11	3070.0000
10	445	435	1334	1174	1700	264	32284	11	3140.7273
10	454	444	1338	1174	1700	269	32991	11	3210.4545
10	463	453	1342	1174	1700	274	33698	11	3280.1818
10	472	462	1346	1174	1700	279	34405	11	3350.9091
10	481	471	1350	1174	1700	284	35112	11	3420.6364
10	490	479	1354	1174	1700	289	35819	11	3490.3636
10	499	488	1358	1174	1700	294	36526	11	3560.0909
10	508	507	1362	1174	1700	299	37233	11	3630.8182
10	517	516	1366	1174	1700	304	37940	11	3700.5455
10	526	525	1370	1174	1700	309	38647	11	3770.2727
10	535	534	1374	1174	1700	314	39354	11	3840.0000
10	544	543	1378	1174	1700	319	40061	11	3910.7273
10	553	552	1382	1174	1700	324	40768	11	3980.4545
10	562	561	1386	1174	1700	329	41475	11	4050.1818
10	571	570	1390	1174	1700	334	42182	11	4120.9091
10	580	579	1394	1174	1700	339	42889	11	4190.6364
10	589	588	1398	1174	1700	344	43596	11	4260.3636
10	598	597	1402	1174	1700	349	44303	11	4330.0909
10	607	606	1406	1174	1700	354	45010	11	4400.8182
10	616	615	1410	1174	1700	359	45717	11	4470.5455
10	625	624	1414	1174	1700	364	46424	11	4540.2727
10	634	633	1418	1174	1700	369	47131	11	4610.0000
10	643	642	1422	1174	1700	374	47838	11	4680.7273
10	652	651	1426	1174	1700	379	48545	11	4750.4545
10	661	660	1430	1174	1700	384	49252	11	4820.1818
10	670	669	1434	1174	1700	389	50059	11	4890.9091
10	679	678	1438	1174	1700	394	50766	11	4960.6364
10	688	687	1442	1174	1700	399	51473	11	5030.3636
10	697	696	1446	1174	1700	404	52180	11	5100.0909
10	706	705	1450	1174	1700	409	52887	11	5170.8182
10	715	714	1454	1174	1700	414	53594	11	5240.5455
10	724	723	1458	1174	1700	419	54301	11	5310.2727
10	733	732	1462	1174	1700	424	55008	11	5380.0000
10	742	741	1466	1174	1700	429	55715	11	5450.7273
10	751	750	1470	1174	1700	434	56422	11	5520.4545
10	760	759	1474	1174	1700	439	57129	11	5590.1818
10	769	768	1478	1174	1700	444	57836	11	5660.9091
10	778	777	1482	1174	1700	449	58543	11	5730.6364
10	787	786	1486	1174	1700	454	59250	11	5800.3636
10	796	795	1490	1174	1700	459	60057	11	5870.0909
10	805	804	1494	1174	1700	464	60764	11	5940.8182
10	814	813	1498	1174	1700	469	61471	11	6010.5455
10	823	822	1502	1174	1700	474	62178	11	6080.2727
10	832	831	1506	1174	1700	479	62885	11	6150.0000
10	841	840	1510	1174	1700	484	63592	11	6220.7273
10	850	849	1514	1174	1700	489	64309	11	6290.4545
10	859	858	1518	1174	1700	494	65016	11	6360.1818
10	868	867	1522	1174	1700	499	65723	11	6430.9091
10	877	876	1526	1174	1700	504	66430	11	6500.6364
10	886	885	1530	1174	1700	509	67137	11	6570.3636
10	895	894	1534	1174	1700	514	67844	11	6640.0909
10	904	903	1538	1174	1700	519	68551	11	6710.8182
10	913	912	1542	1174	1700	524	69258	11	6780.5455
10	922	921	1546	1174	1700	529	70065	11	6850.2727
10	931	930	1550	1174	1700	534	70772	11	6920.0000
10	940	939	1554	1174	1700	539	71479	11	6990.7273
10	949	948	1558	1174	1700	544	72186	11	7060.4545
10	958	957	1562	1174	1700	549	72893	11	7130.1818
10	967	966	1566	1174	1700	554	73500	11	7200.9091
10	976	975	1570	1174	1700	559	74207	11	7270.6364
10	985	984	1574	1174	1700	564	74914	11	7340.3636
10	994	993	1578	1174	1700	569	75621	11	7410.0909
10	1003	1002	1582	1174	1700	574	76328	11	7480.8182
10	1012	1011	1586	1174	1700	579	77035	11	7550.5455
10	1021	1020	1590	1174	1700	584	77742	11	7620.2727
10	1030	1029	1594	1174	1700	589	78449	11	7690.0000
10	1039	1038	1598	1174	1700	594	79156	11	7760.7273
10	1048	1047	1602	1174	1700	599	79863	11	7830.4545
10	1057	1056	1606	1174	1700	604	80570	11	7900.1818
10	1066	1065	1610	1174	1700	609	81277	11	7970.9091
10	1075	1074	1614	1174	1700	614	81984		

Photoats

have to re work the code

we are having an issue with
the sensor not finding the light

pseudocode

find light()
{

motor[my_servo] = 127

}

add print statements for debugging
code.

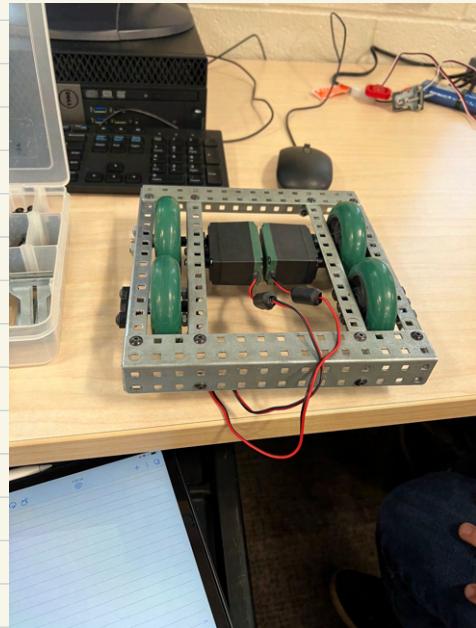
fixed

Check github

Maze traversal

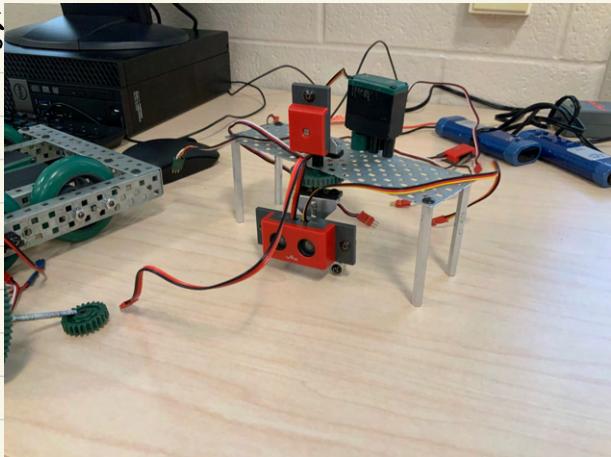
Quick lecture about the maze movement.

today we start from scratch
we take everything down
to the base and start
from there

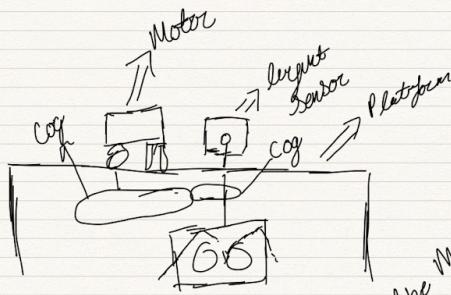


Building a platform for the light sensor and the servos on top using a canopy

Servo gear turns the light sensor

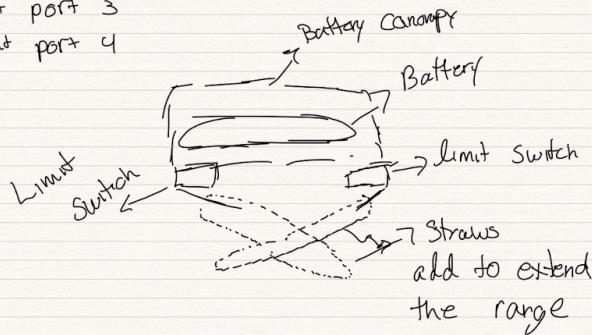


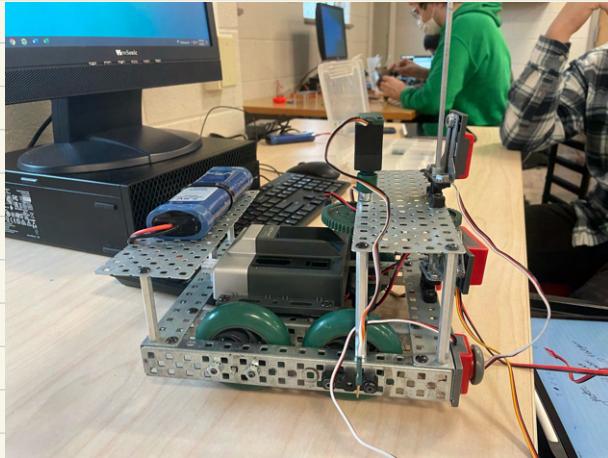
We add two canopies for the battery and the light sensor servo combo.



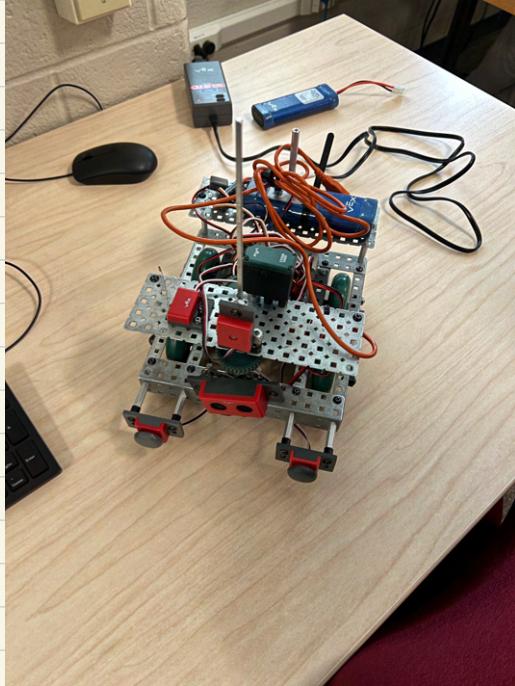
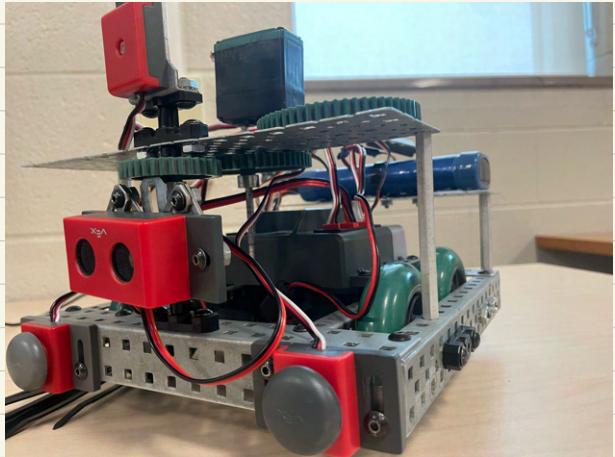
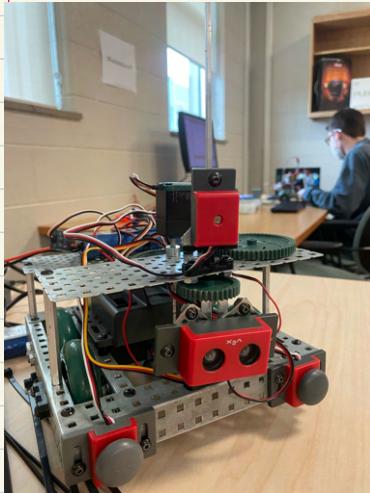
digital
Left port 3
Right port 4

the motor
cog will turn
the leg
down





adding bump sensors to the front with the micro controller
in the middle of the base
the ultra sonic is on the canap
also the servo turns it all



We have re
worked our
maze code
over 10 times
we did not
do well on the
contest

Maze Race

The design is to have the robot turn first before moving towards any direction.

group 3

- no light sensor

Ultra Sonic Sensor

The Ultra sonic sensor does not move right only left hugs the walls.

Straw bump sensor didn't make it stuck in 2nd turn

Group 8 "mr Charles"

Super close got in the right spot when turned at.

Group 11 minator

designed to hug the wall on the right side. Using the ultra sonic sensor.

Group 2 "geo"

limit switch - broke

- light sensor

Our Group -

Terrible

Group 9 - Bumpcc

Group 5 - G won

March 22 - April

Robot Manipulation

- gears
- PID Control
- Kinematics

Inverse Kinematics

two - joints

Multiple Solutions

more degrees of freedom
you have to pick one

No Solutions

- design constraints

The Solution Space

- Configuration Space

7 degrees of freedom in
the human arm

How to solve enough times
Kinematics

- the law of Sines

- the law of Cosines

Solving for θ^2

- Complete triangle
- Label triangle
- find the law that works (Cosine)
- Define right triangle
- Substitution
- Math

Solving for θ_1

$$\frac{b}{\sin B} = \frac{c}{\sin C}$$

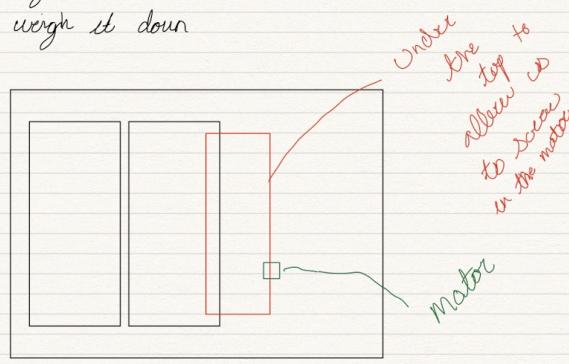
- Finish triangle
- Math

Build table
Implements solve the Problem.

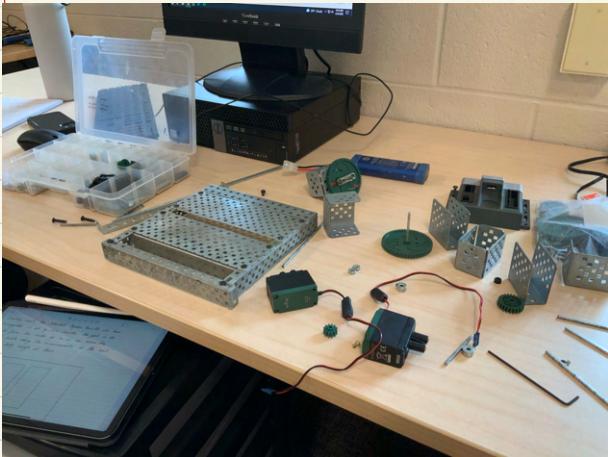
The Build



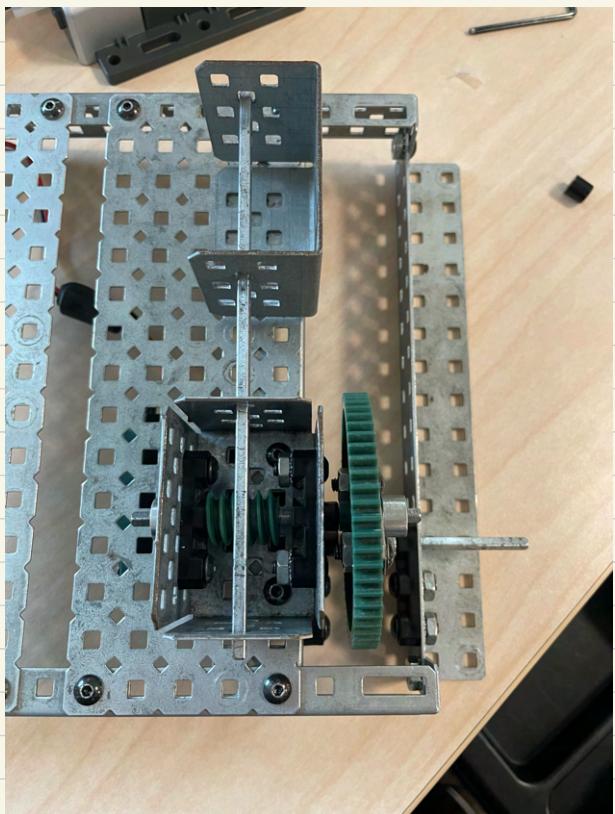
Starting with the Standard Space Build we have incorporated in all of our builds. the goal is to build from the middle up so we can balance the weight. With adding the microcontroller at the back to also help weigh it down

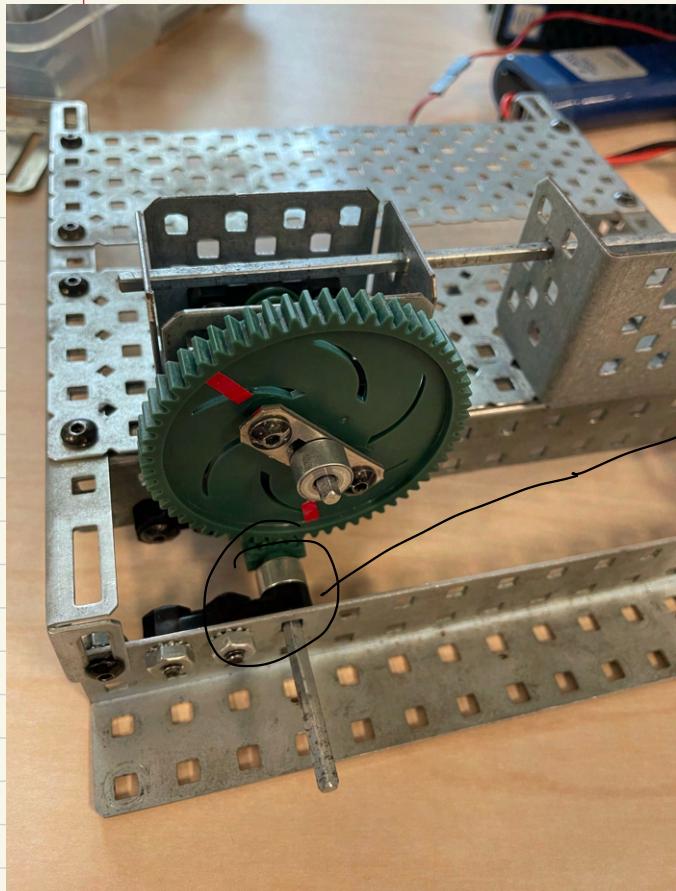


Next we moved to the gears
we need more torque not
Speed



Worked
on a
Worm
gear
Using 1 motor
to move the
whole base

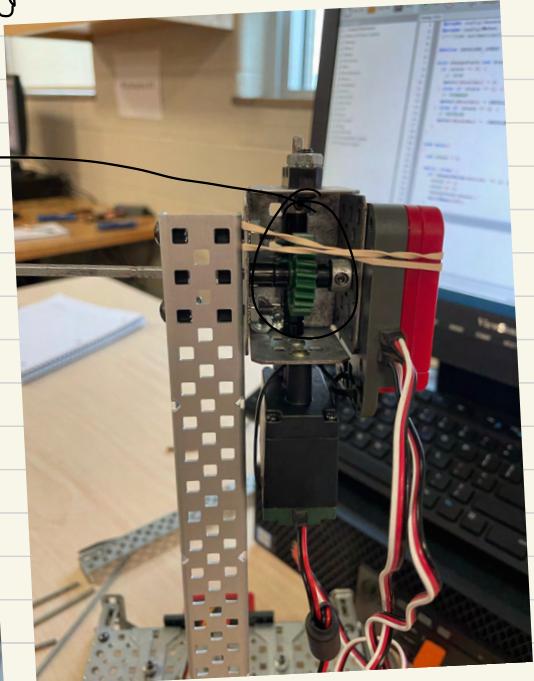
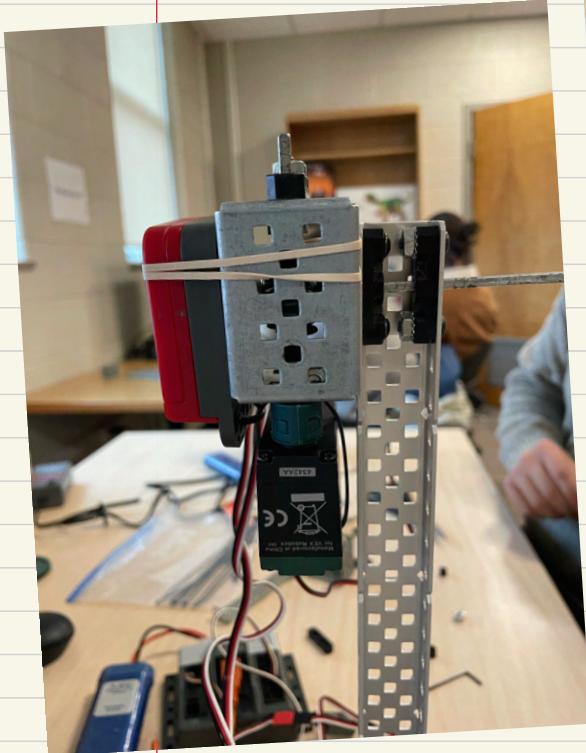




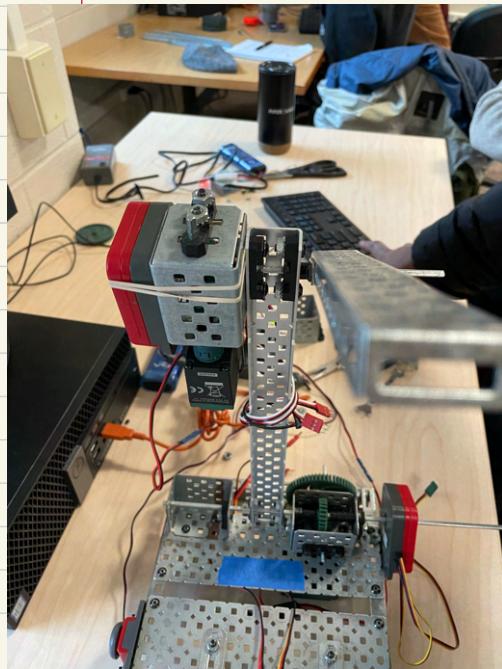
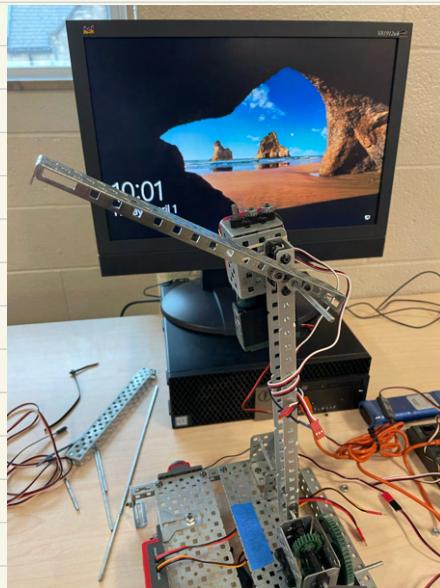
Gear attached
to motor

We are having issues with
the worm gear slipping
we added a rubber band to
the gear to keep it secure.

today we are working on
the Shoulder Using the same
Setup we did for the bottom
shoulder
worm gear



now we are working on our arm

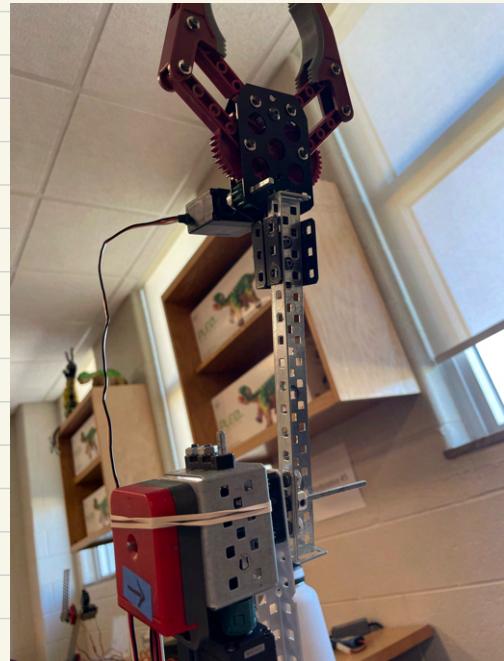
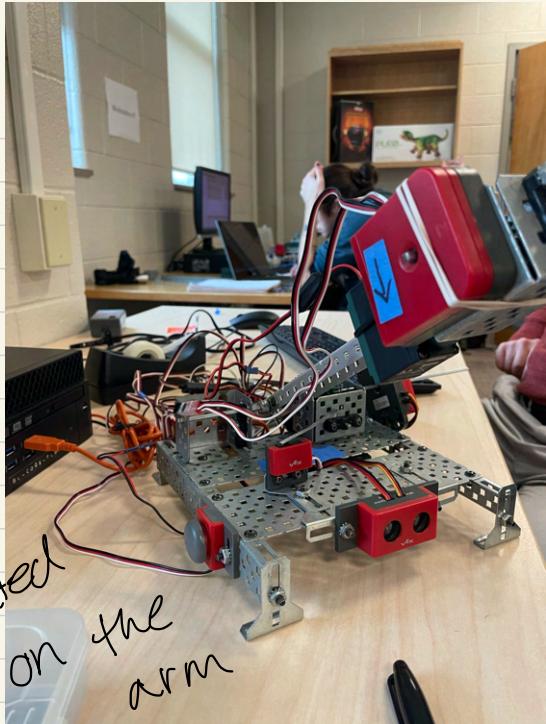


Pseudo code

go to E3

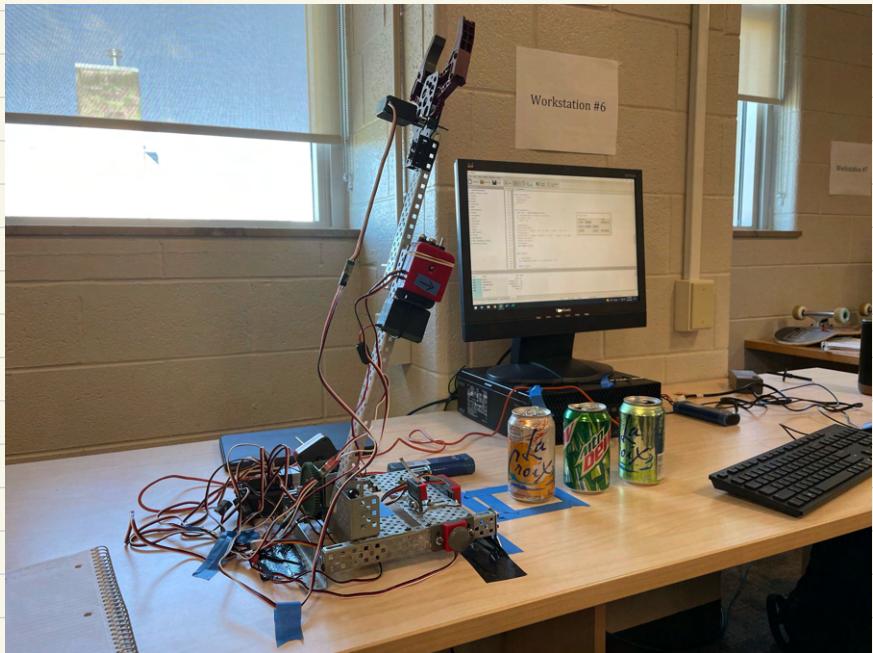
reset()
throwCan()
grabCan()

today
we sat
the
we mounted
it on the
arm



we will test out the claw the
next class

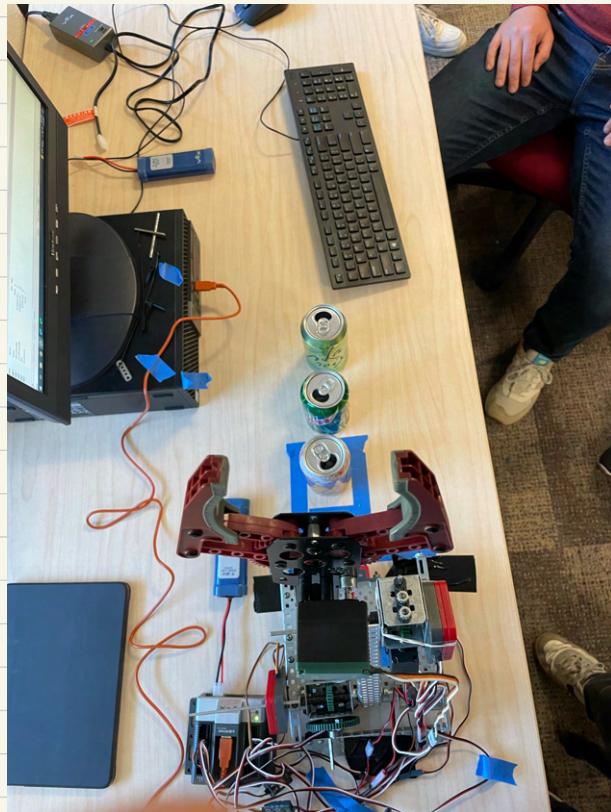
We got the claw to



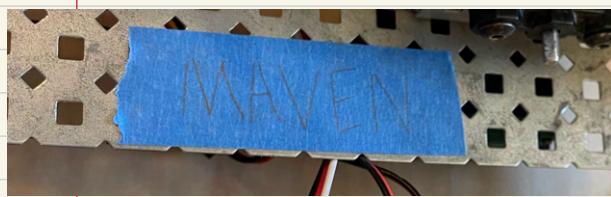
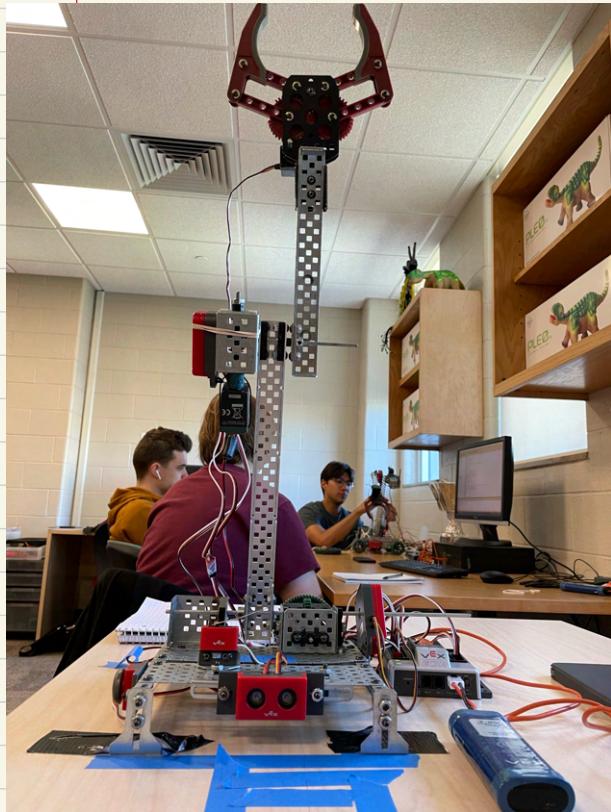
Pick up 1 can trying on
the second can if misses or
when picking up the 1st can
it hits the second one.

We are add to are grabs
function we will grab the can
the automatically move it back wards
towards the base to clear other
cans

we
lowered
down
allowed
space
for the
cans



today we finished our
Manipulation



It's Been
a great
Semester

I have
enjoyed
the course!

- Milton