BIO

Team Members:

1. Celeste Tan (Leader, Software, Electrical)
2. Lim Kae Sophie (Hardware, Electrical)
3. Xue Minjie (Software, Hardware)
4. Aseera Jannath (Software, Hardware)

Operational Logistics:
Forum for communication: Whatsapp, Discord
Meetings frequency: Almost every weekday
Location: School (Raffles Institution)

Navigation

PCB V1, PCB V2, PCB V3, PCB V4, 3D Prints Appendix, Unknown Terms Appendix

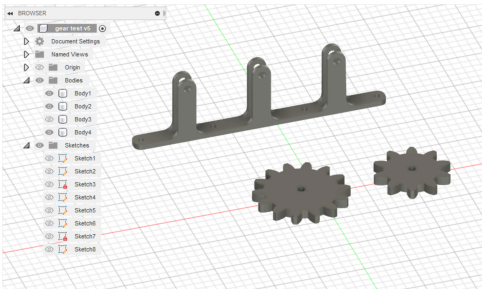
PCB V1

[14/11/2022]
[Kae]

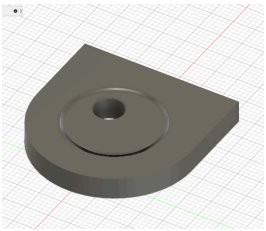
MECHANICAL

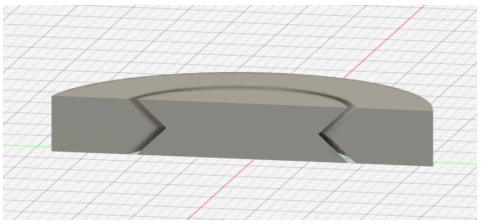
Tasks Done

1. Modelled and printed **gearbox**
    a. using screws and surrounding gear on both sides



2. Modelled and printed **wheel mount**
    a. using print in place design
    b. mainly checking for smoothness and durability





	Issues	Solutions
gearbox v1	• Functional but many issues	• consider print in place for gearbox • make mount walls thicker

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>

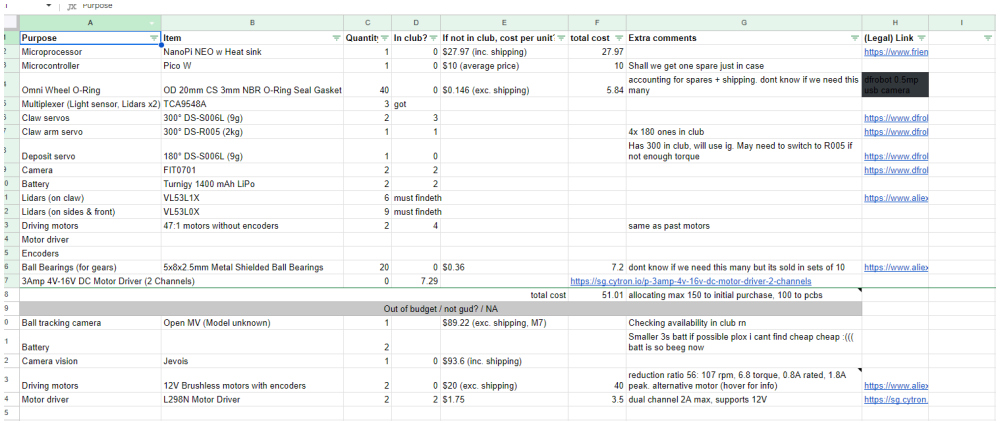
- Screws produce too much friction
- Metal-on-plastic mounting is not durable
- Mounting is too flimsy, gears tend to rotate on multiple axes

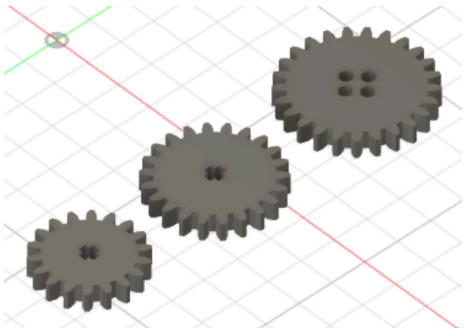
</td><td></td></tr>
<tr><td>wheel mount v1</td><td>

- Certain parts of the mount (i.e. the top) will experience more wear and tear due to weight of the robot acting on it
- Plastic-on-plastic might cause friction

</td><td>

- consider lubrication to fix friction issues
- change angle of print in place design

</td></tr>
</table>
</td></tr>
<tr><td>[15/11/2022]
[Kae, Celeste]</td><td>
ADMIN
<u>Tasks Done</u>
1. Discussed BOM and budget

<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th><th>G</th><th>H</th><th>I</th></tr>
<tr><td>1</td><td>Purpose</td><td>Item</td><td>Quantity</td><td>In club?</td><td>If not in club, cost per unit</td><td>total cost</td><td>Extra comments</td><td>(Legal) Link</td><td></td></tr>
<tr><td>2</td><td>Microprocessor</td><td>NanoPi NEO w Heat sink</td><td>1</td><td>0</td><td>$27.97 (inc. shipping)</td><td>27.97</td><td></td><td>https://www.frien</td><td></td></tr>
<tr><td>3</td><td>Microcontroller</td><td>Pico W</td><td>1</td><td>0</td><td>$10 (average price)</td><td>10</td><td>Shall we get one spare just in case</td><td></td><td></td></tr>
<tr><td>4</td><td>Omni Wheel O-Ring</td><td>OD 20mm CS 3mm NBR O-Ring Seal Gasket</td><td>40</td><td>0</td><td>$0.146 (exc. shipping)</td><td>5.84</td><td>accounting for spares + shipping. dont know if we need this many</td><td>alrobot 0.5mp usb camera</td><td></td></tr>
<tr><td>5</td><td>Multiplexer (Light sensor, Lidars x2)</td><td>TCA9548A</td><td>3</td><td>got</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>Claw servos</td><td>300° DS-S006L (9g)</td><td>2</td><td>3</td><td></td><td></td><td></td><td>https://www.dfrol</td><td></td></tr>
<tr><td>7</td><td>Claw arm servo</td><td>300° DS-R005 (2kg)</td><td>1</td><td>1</td><td></td><td></td><td>4x 180 ones in club</td><td>https://www.dfrol</td><td></td></tr>
<tr><td>8</td><td>Deposit servo</td><td>180° DS-S006L (9g)</td><td>1</td><td>0</td><td></td><td></td><td>Has 300 in club, will use ig. May need to switch to R005 if not enough torque</td><td>https://www.dfrol</td><td></td></tr>
<tr><td>9</td><td>Camera</td><td>FIT0701</td><td>2</td><td>2</td><td></td><td></td><td></td><td>https://www.dfrol</td><td></td></tr>
<tr><td>10</td><td>Battery</td><td>Turnigy 1400 mAh LiPo</td><td>2</td><td>2</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>11</td><td>Lidars (on claw)</td><td>VL53L1X</td><td>6</td><td>must findeth</td><td></td><td></td><td></td><td>https://www.alie</td><td></td></tr>
<tr><td>12</td><td>Lidars (on sides & front)</td><td>VL53L0X</td><td>9</td><td>must findeth</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>13</td><td>Driving motors</td><td>47:1 motors without encoders</td><td>2</td><td>4</td><td></td><td></td><td>same as past motors</td><td></td><td></td></tr>
<tr><td>14</td><td>Motor driver</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>15</td><td>Encoders</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>16</td><td>Ball Bearings (for gears)</td><td>5x8x2.5mm Metal Shielded Ball Bearings</td><td>20</td><td>0</td><td>$0.36</td><td>7.2</td><td>dont know if we need this many but its sold in sets of 10</td><td>https://www.alie</td><td></td></tr>
<tr><td>17</td><td>3Amp 4V-16V DC Motor Driver (2 Channels)</td><td></td><td>0</td><td>7.29</td><td></td><td>https://sg.cytron.io/p-3amp-4v-16v-dc-motor-driver-2-channels</td><td></td><td></td><td></td></tr>
<tr><td>18</td><td></td><td></td><td></td><td></td><td>total cost</td><td>51.01</td><td>allocating max 150 to initial purchase, 100 to pcbs</td><td></td><td></td></tr>
<tr><td>19</td><td colspan="9">Out of budget / not gud? / NA</td></tr>
<tr><td>20</td><td>Ball tracking camera</td><td>Open MV (Model unknown)</td><td>1</td><td></td><td>$89.22 (exc. shipping, M7)</td><td></td><td>Checking availability in club rn</td><td></td><td></td></tr>
<tr><td>21</td><td>Battery</td><td></td><td>2</td><td></td><td></td><td></td><td>Smaller 3s batt if possible plox i cant find cheap cheap :(((batt is so beeg now</td><td></td><td></td></tr>
<tr><td>22</td><td>Camera vision</td><td>Jevois</td><td>1</td><td>0</td><td>$93.6 (inc. shipping)</td><td></td><td></td><td></td><td></td></tr>
<tr><td>23</td><td>Driving motors</td><td>12V Brushless motors with encoders</td><td>2</td><td>0</td><td>$20 (exc. shipping)</td><td>40</td><td>reduction ratio 56: 107 rpm, 6.8 torque, 0.8A rated, 1.8A peak. alternative motor (hover for info)</td><td>https://www.alie</td><td></td></tr>
<tr><td>24</td><td>Motor driver</td><td>L298N Motor Driver</td><td>2</td><td>2</td><td>$1.75</td><td>3.5</td><td>dual channel 2A max, supports 12V</td><td>https://sg.cytron.</td><td></td></tr>
<tr><td>25</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>[16/11/2022]
[Kae]</td><td>
MECHANICAL
<u>Tasks Done</u>
1. Modelled gears v1 with the correct gear ratio and properly calculated parameters
a. used Helix Gear Generator extension for Fusion360
</td></tr>
</table>

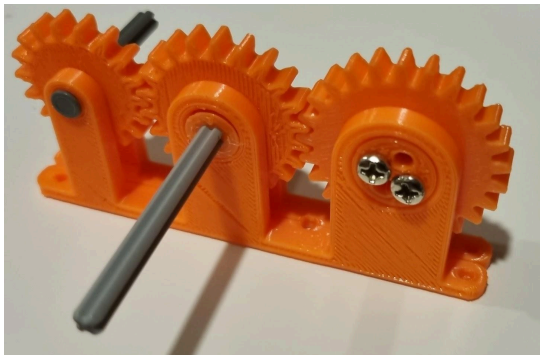


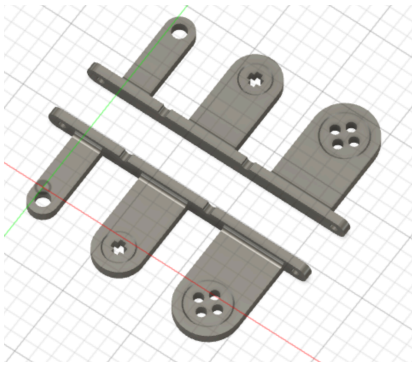
[17/11/2022]
[Kae]

MECHANICAL

Tasks Done

1. Printed **gears v1**
2. Modelled and printed 3 types of **gearboxs** (v2, 3, 4) for testing
    a. 2: gearboxed on lego axel that is free spinning in empty holes
    b. 3: gearboxed on lego axel inserted in rotating print-in-place holes
    c. 4: Gear screwed into rotating print-in-place holes





3. Decided to use the same type of mount for wheel and gears

	Issues	Solution
gears v1	• gear smoothness is affected by printing with elephant's foot	• watch first layer of printing • increase backlash in design
gearbox v2	• Causes too much friction, will wear down axels over time	• modify v2 to use ball bearings instead of an hole for a smoother rotation
gearbox v3	• Rough, bumpy rotation due to imperfections in 3D print • Too much friction produced • Due to the print-in-place mechanism, the mount would	• develop version 2 instead

		have to be printed in 2 parts, reducing stability	
	gearbox v4	• Print-in-place mount produces too much friction • Due to the print-in-place mechanism, the mount would have to be printed in 2 parts, reducing stability	• develop version 2 instead

[10/12/2022]
[Celeste, Kae]

ADMIN

Tasks Done

1. Due to tight budget constraints, shortlisted parts
2. Reviewed and purchased all relevant parts: motors, pico, motor driver, etc. from BOM

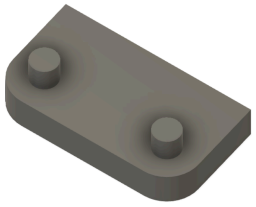
	A	B	C	D	E	F
1	Item	Quantity	Unit Cost	SGD	USD	Link
2	Pico (No presoldered headers)	2	6.4	12.8		https://sg.cytron.io/p-raspberry-pi-pico-board
3	OD 20mm CS 3mm NBR O-Ring Seal Gasket (	2	2.92	5.84		https://shopee.sg/Fast-delivery-CS-3mm-Black-Nitrile-O-Ring-NBR-Rubber-Seal-Washer-O-Ring-OD-10mm-80mm-i.578645436.120
4	0.3MP USB Camera	1	9.9		9.9	https://www.dfrobot.com/product-2089.html
5	Metal DC Geared Motor w/Encoder - 6V 210RP	3	19.9		59.7	https://www.dfrobot.com/product-1617.html
6	2 channel motor driver	2	8		16	https://www.aliexpress.com/item/1005004734230943.html?spm=a2g0o.order_list.order_list_main.5.451d1802FdFpzc
7				0	0	
8				0	0	
9				0	0	
10				0	0	
11				0	0	
12				0	0	
13	Estimated cost of PCBs (order later after confirm parts)			50	0	
14	Shipping buffer			20	0	
15	Total individual currency			88.64	85.6	
16	Total with 1:1.36 exchange rate			205.056	150.7764706	
17						

[13/12/2022]
[Kae]

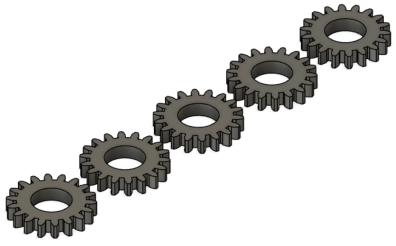
MECHANICAL

Tasks Done

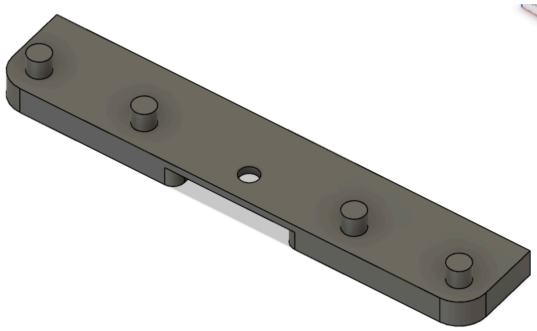
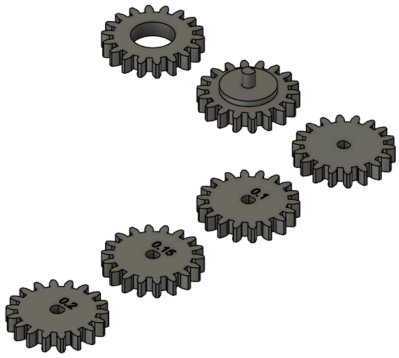
1. scrapped old gearbox idea with axels, new **gearbox v5** incorporating ball bearings

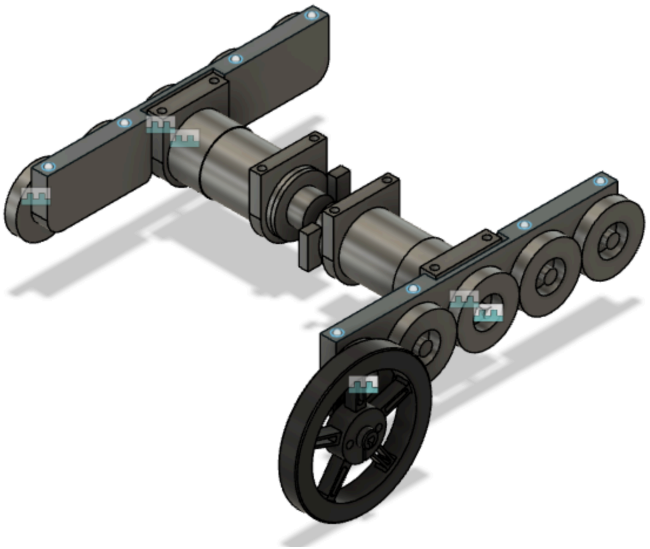


2. modelled 27.45mm pitch diameter gears (**gears v2**)

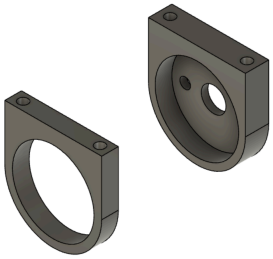


	Issues	Solution

<table>
<tr><td></td><td>
<table>
<tr><td>gearbox v5</td><td>• ball bearings may fall off with wear and tear
• gears may be misaligned if spread out over multiple mounts</td><td>• tight fit: glue/hammer in ball bearings into print
• combine gear train into one long mount</td></tr>
</table>
</td></tr>
<tr><td>[31/12/2022]
[Kae]</td><td>
MECHANICAL
<u>Tasks Done</u>
1. extended the gearbox in v6 to mount all 5 gears
a. mount the whole gear train onto one mount
b. decided on a 5-gear train with motor driving gears in the middle

2. modelled, printed and tested how to mount gear onto motor + wheel onto gear in gears v3

3. started mockup of whole robot in Fusion360
</td></tr>
</table>



4. modelled **motor mounts v1**



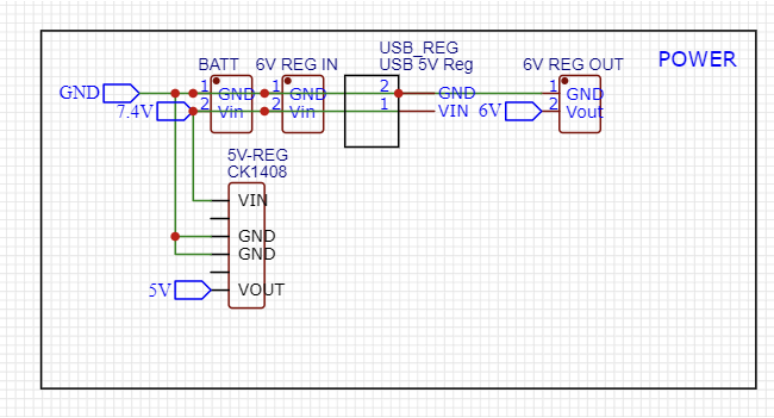
	Issues	Solution

[02/01/2023]
[Celeste]

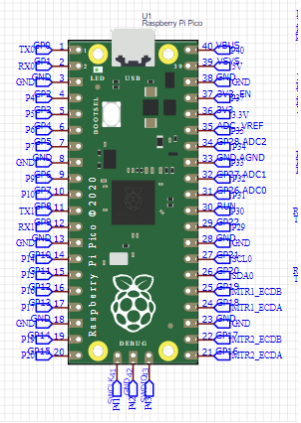
ELECTRICAL

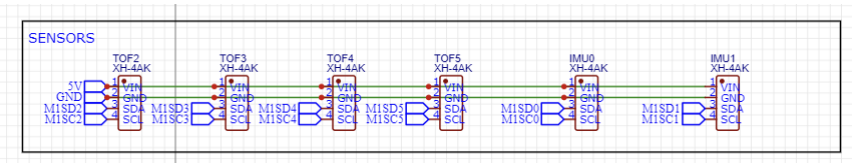
Tasks Done

1. **Schematics** in EasyEDA (PCB V1)
    a. Calculated widths of main power lines from battery to 5V, 6V reg using P = IV and a trace width calculator. Used external layers in air and 1 oz copper as settings.
    b. Created main net ports of each different potential (3.3V, 5V, 6V, 7.4V)



c. Inserted components from JLCPCB library: Pico, JST XH for LiDARs



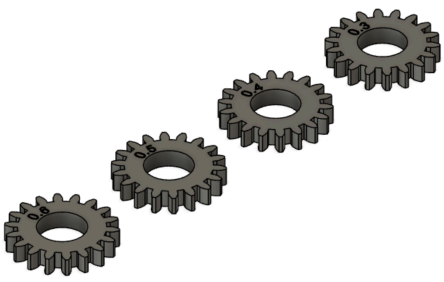


[04/01/2023]
[Kae, Celeste]

MECHANICAL

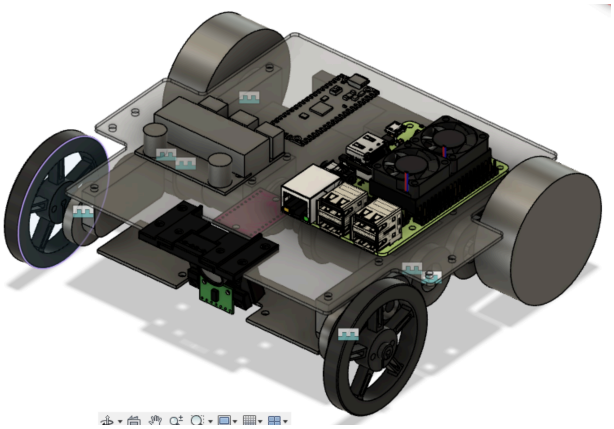
Tasks Done

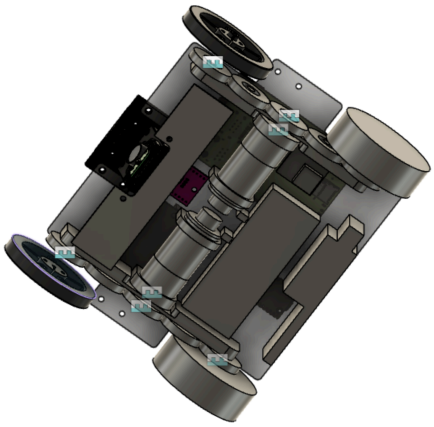
1. decided on metal shielded ball bearing to purchase
    a. outer diameter: 13mm, inner diameter: 6mm, thickness: 5mm
    b. https://shopee.sg/6mm-x-13mm-x-5mm-Miniature-Deep-Groove-Radial-Ball-Bearings-686Z-10Pcs-i.110868311.1947445015
2. modelled and printed **gears v4** to test fit onto ball bearings
    a. decided on 1:1 motor:wheel gear ratio

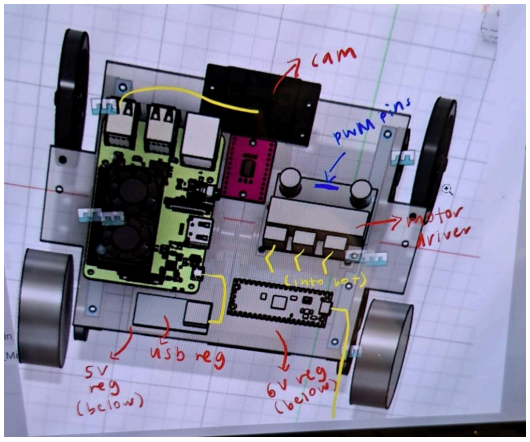


3. modelled **bottom camera mount v1**

4. decided on added a **Bottom PCB** below the main PCB for light sensors for linetrack
5. added CAD of Raspberry Pi 3b+, Raspberry Pi Pico, TCA9548A, motor driver, bottom camera mount, battery, 6V Regulator, 5V Regulator, omniwheels and both PCBs' outline to the main robot CAD
    a. might need to rearrange later







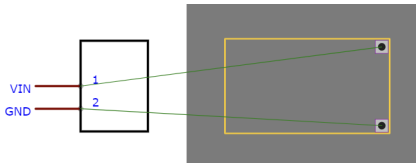
	Issues	Solution
gears v4	• gears may fall off ball	• considering epoxy glue

	bearings with repeated use	or using a very tight fit (hammer into place)
bottom camera mount v1	• may be too bulky	
PCB v1	• Too little space • Pico usb inaccessible	• Realised we did not need 3 multiplexers • Move pico

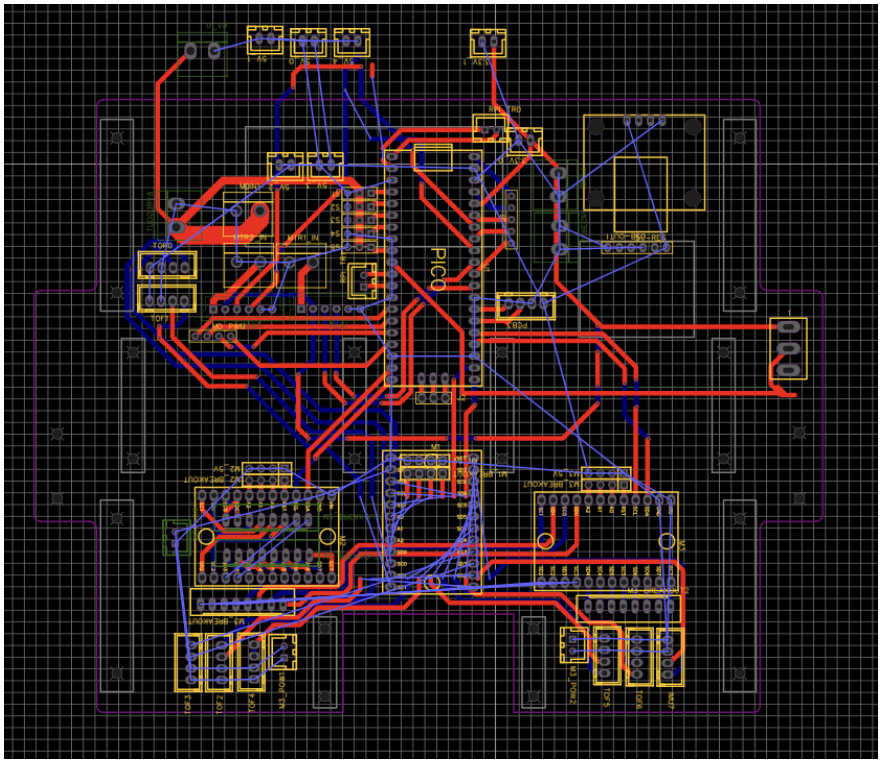
ELECTRICAL

Tasks Done

1. **PCB V1 Schematics** in EasyEDA (PCB V1)
    a. Custom made a simple component symbol and footprint for USB 5V Reg



    b. Traced the rest of the signal traces with 30 mil
2. **PCB V1**
    a. Arranged the components in a first draft of the PCB



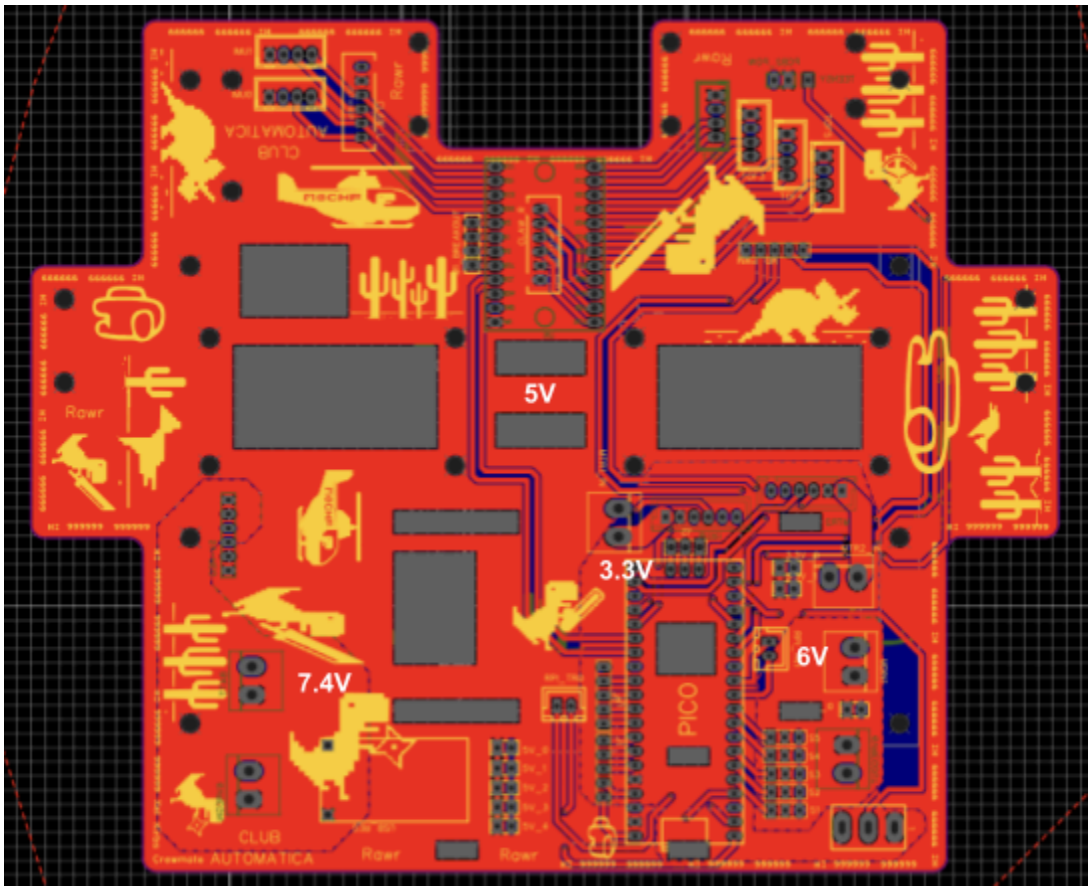
[05/01/2023]
[Celeste, Kae]

ELECTRICAL

Tasks Done

1. **PCB V1**

a. Reorganised PCB Rerouted traces using copper pour for each different potential (3.3V, 5V, 6V, 7.4V)
b. Removed unneeded multiplexers as the IMU has a different address from the VL53L1X (0x29) and the VL53L0X (0x29)
c. Copper pour into ground
d. Added designs to silkscreen for personalisation



	Issues	Solution
PCB v1	• Not enough space for thick traces	• Used versatile copper pour to stay flexible in tight spaces
PCB v1	• Pico USB not accessible from front (will be blocked by gripper)	• Relocated the pico to allow USB space for connection

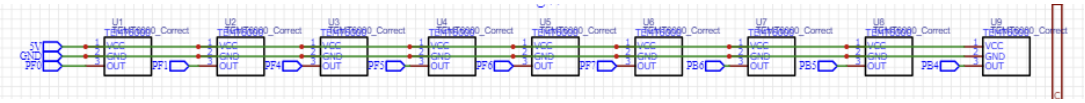
[07/01/2023]
[Kae, Celeste]

ELECTRICAL

Tasks Done

1. **Bottom PCB V1**
   a. Planned for 6 sensors, 3 on each side of the camera

b. Wired the schematics for the sensors to connect to the teensy



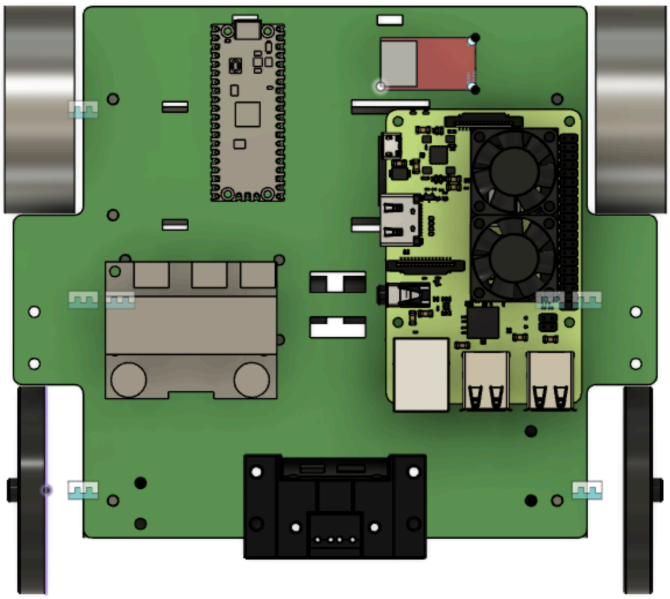
	Issues	Solution
Bottom PCB V1	• Not enough pins on the pico to read from 6 sensors	• Used a separate microcontroller teensy to interface between the two and transfer data via UART

[07/01/2023]
[Kae, Celeste]

MECHANICAL

Tasks Done

1. added USB-A 5V Regulator to **PCB v1 CAD**
2. added holes for zip ties and velcro to **PCB v1** outline



[08/01/2023]
[Kae, Celeste]

ELECTRICAL

Tasks Done

1. Ran Design Rule Check on **PCB v1** and resolved Via diameter problems

	Issues	Solution
PCB v1	• Via diameters are too small	• Changed all vias to a larger diameter to pass DRC

ADMIN

Tasks Done

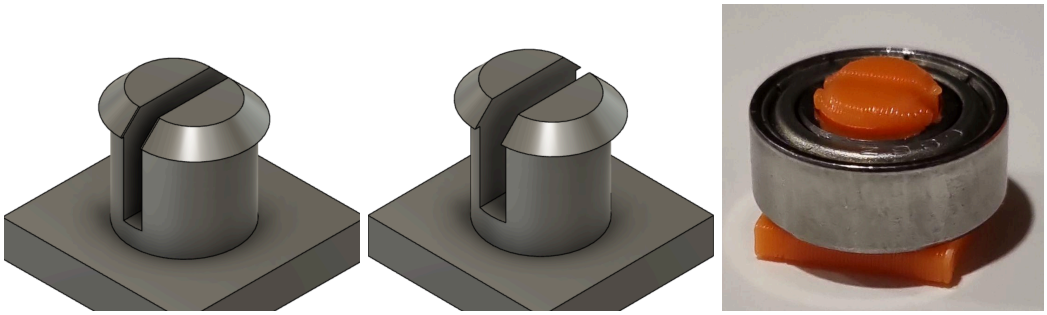
1. Ordered **PCB v1** and **Bottom PCB v1** through JLCPCB

[09/01/2023]
[Kae]

MECHANICAL

Tasks Done

1. modelled, printed and tested a **snap-fit mechanism** for the ball bearings



Design Concepts

- snap-fit mechanism
  - easy for repeated mounting and removal of ball bearings
  - any edits made to gear design can be easily swapped out
  - compared to using glue or friction to hold ball bearings to the gearbox, this is a more versatile and convenient design

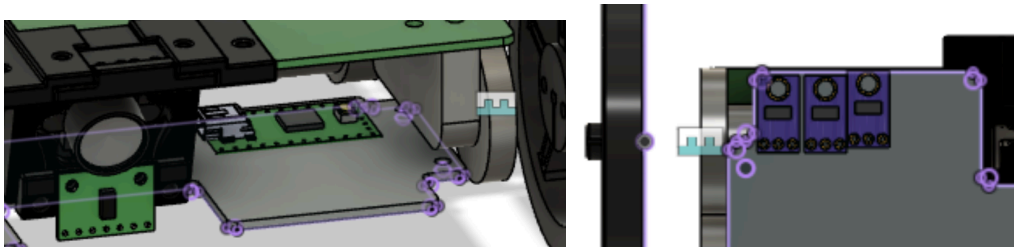
	Issues	Solution
snap-fit mechanism	• too stiff • ball bearing will fall out due to plastic bending • ball bearing becomes loose over time, causing misalignment of gears • mechanism's prongs snap under too much force	• expanded gap between the moving parts • increased size of lip • consider a brace for wheels since they experience the most force • change 3d print orientation so layer lines are not along the stress points of the print

[11/01/2023]
[Kae]

MECHANICAL

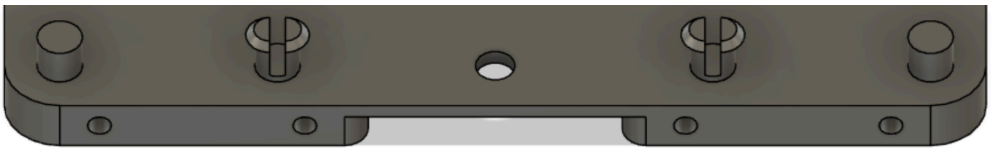
Tasks Done

1. added Teensy 2.0 and TEMT6000 to the robot CAD on the **Bottom PCB v1**

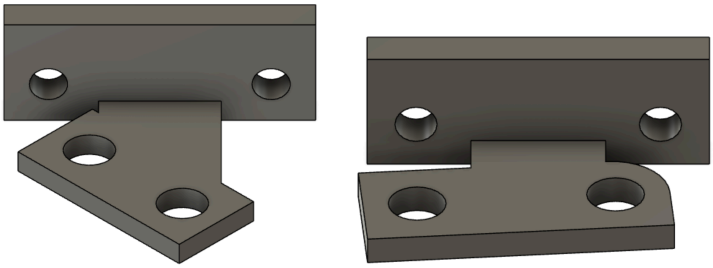


2. changed outline of the **Bottom PCB v1**

3. modified **gearbox v7**
    a. added holes to mount the Bottom PCB onto the gearbox
    b. incorporated snap-fit mechanism



4. modelled and printed **diagonal lidar** (vl53l1x) **mounts v1**



[15/01/2023]
[Celeste]

SOFTWARE

Tasks Done

1. Experimented with different softwares and ways to program the pico
2. Upload blink code and basic protocols (UART and I2C) to test the pico using Earl E. Philhower's core

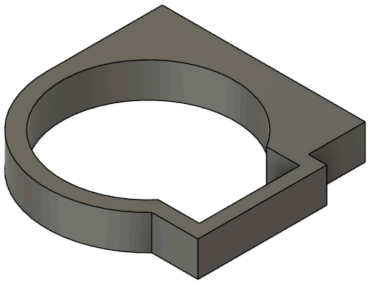
	Issues	Solution
Pico SDK	• Compiled extremely slowly • Had to reset and put the pico in bootloader mode every single time code is to be uploaded • Steep learning curve for unfamiliar beginners	• Using the Earl E Philhower's arduino-pico core to code and upload in Arduino IDE

[17/01/2023]
[Kae]

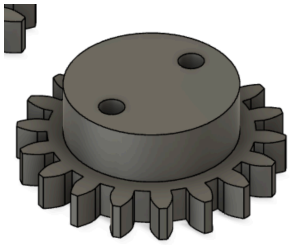
MECHANICAL

Tasks Done

1. modelled and printed new back motor mounts in **motor mounts v2** to fit around connector on motor



2. modelled and printed new wheel gear design (**gears v5**) to mount front wheels and omniwheels



ELECTRICAL

Tasks Done

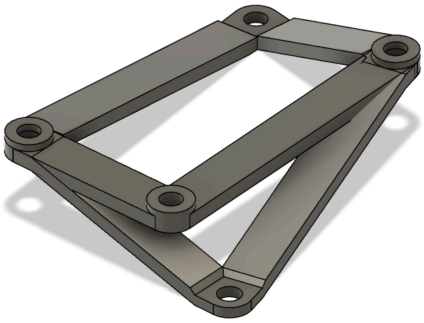
1. PCB v1 delivered
2. Tested connections on **PCB v1** with main power lines using digital multimeter

[18/01/2023]
[Kae]

MECHANICAL

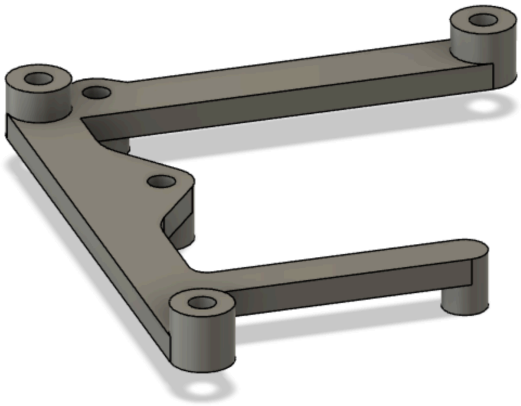
Tasks Done

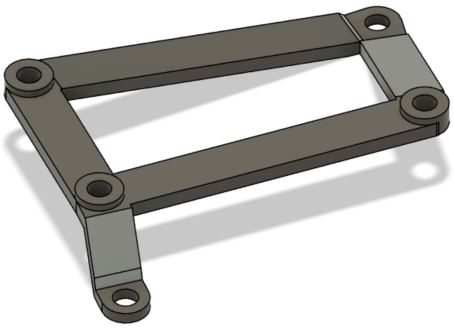
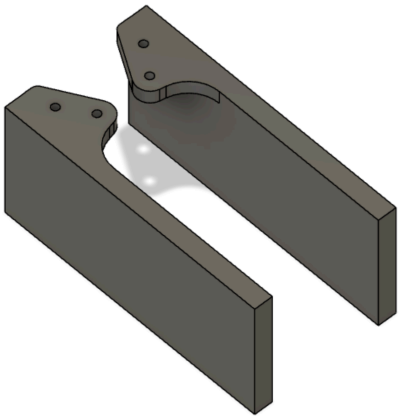
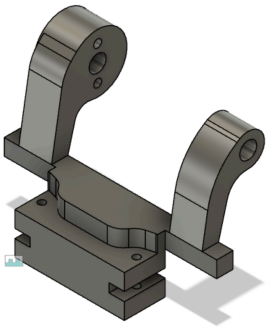
1. Modelled motor driver mount v1

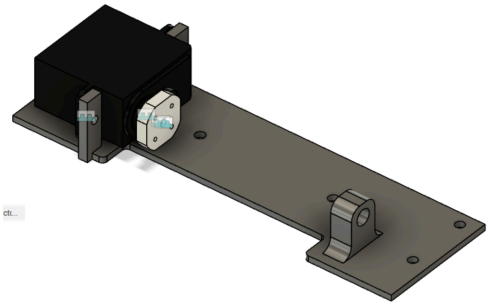


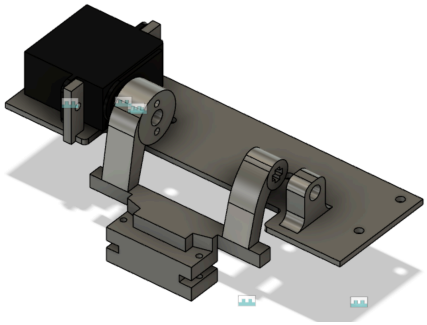
	Issues	Solution
motor driver mount v1	• possibly too flimsy	• increase thickness of the model • add more supporting structures

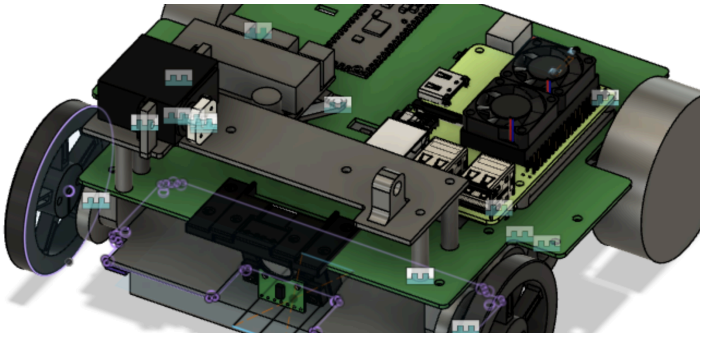
<table>
<tr><td></td><td>
<table>
<tr><td></td><td>• difficult to remove supports after printing</td><td>• remove overlapping structures, simplify design</td></tr>
</table>

ELECTRICAL
<u>Tasks Done</u>
1. Soldered headers and JSTs onto PCB
2. Soldered the 5V regulator and the 5V USB Regulator</td></tr>
<tr><td>[19/01/2023]
[Kae]</td><td>MECHANICAL
<u>Tasks Done</u>
1. modelled, printed and tested rpi 3 mount v1
 
<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>rpi 3 mount v1</td><td>• components on the PCB prevent the mount from being secured to all 4 screw holes on the rpi 3
• the mount tilts and flexes due to the weight of the rpi 3</td><td>• a 3 point connection should be stable
• consider adding more screw holes to mount the print to the PCB for increased stability</td></tr>
</table></td></tr>
<tr><td>[20/01/2023]
[Kae]</td><td>MECHANICAL
<u>Tasks Done</u>
1. modelled motor driver mount v2
a. simplified design to allow for easier and more consistent 3d prints</td></tr>
</table>

	 2. modelled **grippers v1** for servo hand and 9g clutch servo 
[23/01/2023] [Kae, Celeste]	**MECHANICAL** <u>Tasks Done</u> 1. modelled, printed and tested **gripper arm v1** a. decided that gripper arm is mounted to 300 degree 9g clutch servo and supported by lego axel b. considered a L-shape arm instead of a curved arm, rejected due to weaker stress points and possibility of cracking   2. modelled, printed and tested **servo mount v1** a. used to mount the servo driving the gripper arm b. decided to mount the servo mount to the PCB with metal standoffs







	Issues	Solution
gripper arm v1	• gripper arm structure might be too weak, may not be able to withstand long term use • unable to get perfect fit of servo screw holes into the gripper arm, connection is not secure	• increase wall thickness for 3d printing • use hot glue to reinforce and prevent wanted rotation

SOFTWARE

Tasks Done

1. Wrote basic skeleton of **drivebase library**
    a. Classes for motor and entire robot consisting of the two instances of motor

```
class Vroom
{
    public:
        Vroom(Motor *left, Motor *right);
        void setSteer(double rpm, double rotation), //when speed is negative, robot heading switches
            reset(),
            resetPID(),
            stop();

    private:
        Motor *_left, *_right;
};
```

    b. Wrote initialization functions for each class
    c. Interrupt functions for the encoders
    d. Wrote functions:

i. Read from encoders
ii. Get speed (calculate from encoder values using the intervals between the pulses)

	Issues	Solution

Encoder interrupts	Interrupts in library return an error in arduino on the pico core	Call attachInterrupt() in the void setup() in main file
		`//^ MOTOR ENCODERS`
		`attachInterrupt(MotorL.getEncAPin(), ISRLA, RISING);`
		`attachInterrupt(MotorL.getEncBPin(), ISRLB, RISING);`
		`attachInterrupt(MotorR.getEncAPin(), ISRRA, RISING);`
		`attachInterrupt(MotorR.getEncBPin(), ISRRB, RISING);`
		`Serial.println("Encoders initialised");`

ELECTRICAL

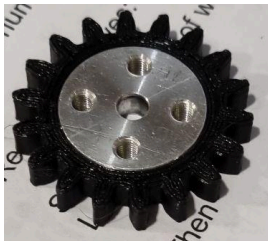
	Issues	Solution
PCB V1	BURNT THE MOTOR DRIVER AND 6V REG while testing motor: ● Realised we wired it wrong and connected one of the motor outputs of the motor driver to gnd, which resulted in a direct short from 6V (this was not figured out until 3 days later) ● Resultant casualties of the 3 days: ○ 2 motor drivers ○ Adjustable Reg 20A	● Manually connect the motor's wires into the screw terminal of the motor driver ● Run without the 6V Reg and deliver 7.4V straight, with care not to overstall it

[24/01/2023]
[Kae]

MECHANICAL

Tasks Done

1. tested mounting motor gear onto aluminium motor mounting hub



2. added a small ring extrusion to gears in **gears v6** to prevent friction against gearbox

3. modelled and printed new motor gear design (**gears v6**) to screw into mounting hub
    a. mounting hub is smaller than base diameter of gear, so no gear realignment required

	Issues	Solution
motor gear (gears v6)	• easily falls off especially due to the smoothness of the metal	• consider epoxy ○ prevents mounting hub from being reused • screw gear into the mounting hub
wheel gear (gears v6)	• ring extrusion results in print warping 	• remove ring form wheel gears, use only on normal gears

SOFTWARE

Tasks Done

1. Added **encoder-based motor speed control** to the **drivebase library**

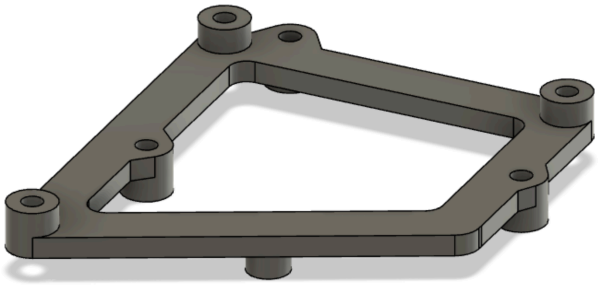
- Used a contributed PID library and integrated it into library using the required speed and the speed obtained from encoders, correcting for errors in motor speed during line track

[27/01/2023]
[Celeste, Kae]

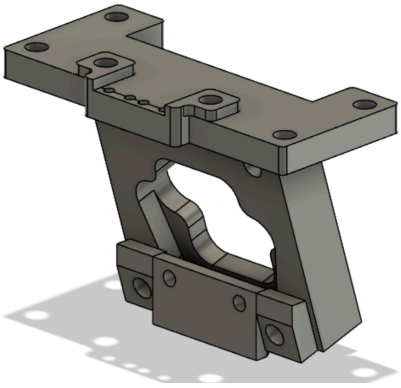
MECHANICAL

Tasks Done

1. modelled and printed **rpi 3 mount v2**
    a. uses triangular shape for maximum support
    b. increased the number of points of the mount being screwed into the PCB
    c. added support structures touching the PCB to minimise flexing of mount



2. modelled **new bottom camera mount v2** with lidar (vl53l1x) and led mounts



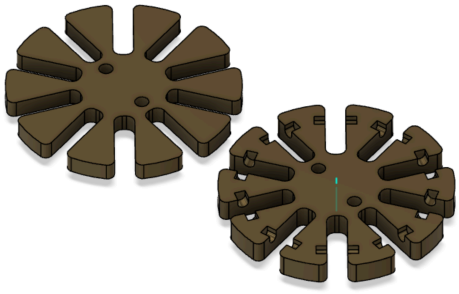
	Issues	Solution
bottom camera mount	• led mount walls are thin, might snap	• sacrifice saving space, make walls thicker

[29/01/2023]
[Kae]

MECHANICAL

Tasks Done

1. confirmed purchase of nitrile rubber o-rings
    a. 20mm diameter, 3mm thickness
2. confirmed purchase of metal rods
    a. 8mm length, 2mm diameter
3. modelled **omniwheel v1**
    a. 10x 20mm diameter o-rings
    b. 60mm diameter



Design Concepts

- custom-made omniwheels
  - 60mm diameter
    - matches diameter of front wheels to prevent excessive scraping
    - PCB will remain parallel to the ground
  - 20mm diameter o-rings
    - diameter is double the height of speed bumps to cross them more effectively when approached perpendicularly

	Issues	Solutions
omniwheel v1	• Distance between adjacent o-rings might be too thin, might break • omniwheel housing might be too flimsy	• increasing wall thickness when 3d printing

[30/01/2023]
[Celeste]

SOFTWARE

Tasks Done

1. Updated **drivebase library**
   a. Tuned P and I for the encoder-based motor speed control
   b. Wrote setRpm and getRps functions for a different measurement system, as the max rpm of each motor was known to be 210Rpm@ no load
   c. Tested the encoder-based motor speed control in a simple forward drive

```
double Motor::setRpm(double rpm) //* rev min^-1
{
    _wantedRpm = fabs(rpm); //setpoint
    noInterrupts();
    _realRpm = this->getRpm(); //input
    interrupts();
    _motorPID.Compute();
    if (rpm >= 0) {
        digitalWrite(_pwmPin1, LOW);
        analogWrite(_pwmPin2, (int)(_neededRpm)); //output
    } else {
        analogWrite(_pwmPin1, (int)(_neededRpm));
        digitalWrite(_pwmPin2, LOW);
    }
    return _neededRpm;
}
```

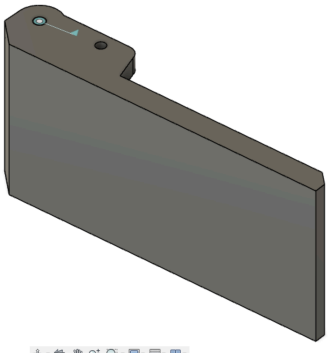
	Issues	Solution
encoder-based motor speed control	• Robot was not reacting well to the PID	• Tune P and I

[31/01/2023]
[Celeste, Kae]

MECHANICAL

Tasks Done

1. modeled **grippers v2**
    a. changed pivot point to be closer to the edge of the gripper



SOFTWARE

Tasks Done

1. Updated the **drivebase library**
    a. Added the function of setSteer(rpm, rotation) for differential steer.
    b. Tested the function setSteer()
    c. Added function resetPID()

Design Concepts

Slower is the rpm of the slower wheel, calculated with the following formula

```
double slower = rpm*(1-2*fabs(rotation));
```

- At 0.5, this is a one wheel turn to the right. At -0.5, this is a one wheel turn to the left
- 1 and -1 is a two wheel turn to the right and left respectively.
- Any number in between is a scaled rotation value where rotation determines the robot's speed of change of angle with the same given rpm

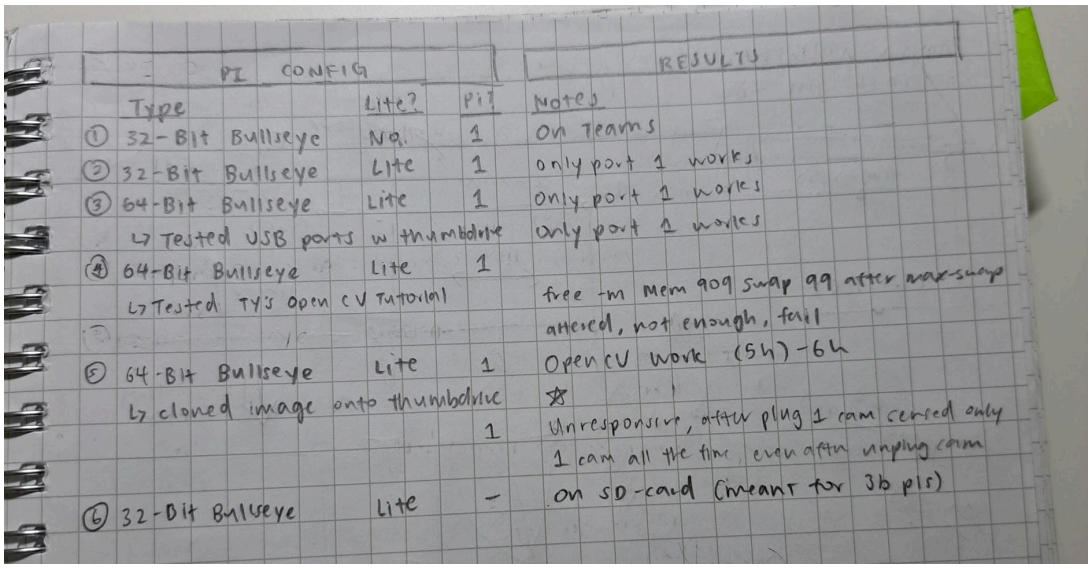
	Issues	Solution
Encoder-based motor speed control	High pitched noise when the robot is supposed to stop and coasting when switch is turned off because of the PID attempting to correct for the error	Clear the existing error using a resetPID function

[01/02/2023]
[Celeste]

SOFTWARE

Tasks Done

1. Attempted to stream **two camera**'s video feed
    a. Flashed the raspberry pi OS onto the SD card with auto connect to SSH using the raspberry pi imager
    b. Downloaded **OpenCV** onto the raspberry pi 3b+
    c. Used filezilla to connect to the pi to upload files with port 22
    d. Used putty to run commands on the pi
    e. Wrote testcam.py and testdualcam.py to stream two cameras
    f. Used multithreading to allow only the most relevant camera frame to be processed as the python script runs slowly due to the heavy load



	Issues	Solution
OpenCV	MANY ISSUES with the downloading of opencv: ● Missing builds, unknown errors, etc. ● After going through about 6 tutorials or so, finally found the one	USE QENGINEERING opencv tutorial - the only problem free one https://qengineering.eu/install-opencv-4.5-on-raspberry-pi-4.html

ELECTRICAL

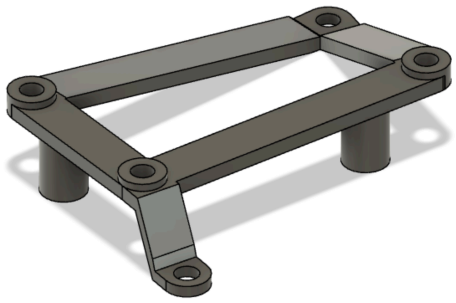
Issues	Solution
When camera is disconnected, the camera is sometimes unable to be recognised or detected again when it is plugged back in: Suspicions: ● the camera driver is not very good ● the ports are slowly dying (most likely this reason) ● Opencv did not build properly and thus was unable to stream ● Processing power not enough to handle two video streams (but sometimes it doesn't even appear as a connected stream so that shouldn't matter)	Switch to **Raspberry Pi 4B** instead of the 3B+ which may have fried ports. Ensure the cameras are both plugged into USB 3.0 instead of 2.0

[06/02/2023]
[Kae]

MECHANICAL

Tasks Done

1. added more support structures resting on the PCB to **motor driver mount v3**



	Issues	Solution
motor driver mount v1	● motor driver began rotating and shifting due to flexing of	● add more support structures for

		the print	stability

[09/02/2023]
[Celeste, Kae]

MECHANICAL

Tasks Done

1. Tilted bottom camera down to 25 degrees (**bottom camera mount 3**), printed
2. received rods and o-rings, assembled omniwheels (**omniwheel v1**)

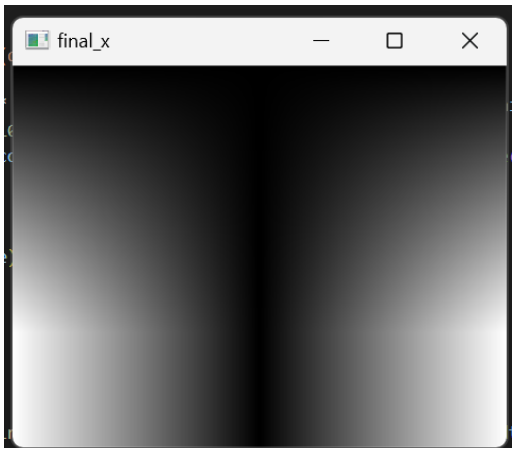
SOFTWARE

Tasks Done

1. Created bwcalib.py to tune the threshold value of the black in the grayscale image using a slider from the live video feed

2. **Linetrack** with **camera**
    a. The image is read using a frame from the stream utilizing multithreading
    b. Convert the image to grayscale

c. Masked the image using the tuned value of black
d. Used a pre populated vector matrix which varies based on distance of the pixel to the origin



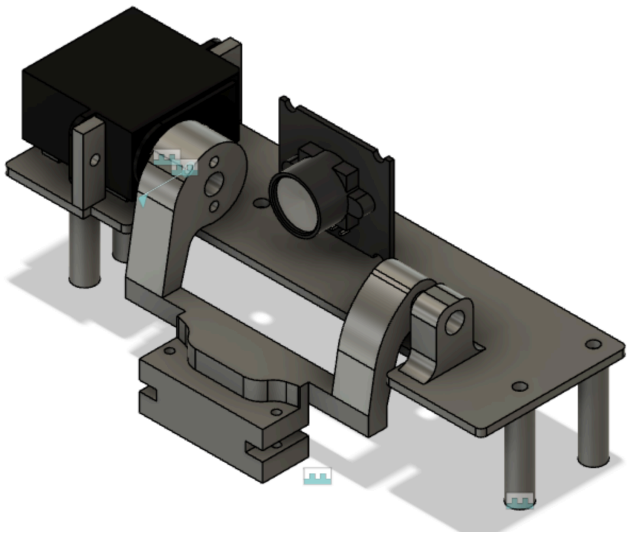
e. The weighted sum is the overall output vector that determines the target angle and thus the rotation value

3. Wrote testserial.py for two way serial communication, from pi to pico and from pico to pi
    a. Downloaded pyserial on the raspberry pi
    b. Connected the TX RX pins for UART
    c. Managed to print numbers sent over from pi and vice versa

<u>Design Concepts</u>

A list of [type, data, type, data, type, data] is sent over to the pico from the pi. The type falls above 200 and the data is from 0 - 200. On the pico side, it understands that anything above 200 is a type and anticipates the incoming data as of that type. Example: [255, 80] might indicate as the rpm is 80 if 255 is preset to be rpm.

	<u>Issues</u>	<u>Solution</u>
Camera stream	• Many frames are read fast but the loop time of processing that image is slow, resulting in a lag in camera frames	• Use multithreading so that only the most recent image is being utilised
Serial from pi to pico	• Complicated to process on the pico's side, many wrong data and rubbish data transfers	• Send over a list [a,b,c] which will end up being processed one by one on the pico side as a 8 bit unsigned integer (0 - 255 effective range). This can be converted directly into an (int) using type casting
Serial from pico	• Pi is missing some data	• Allow for the data to complete

<table>
<tr><td></td><td>
<table>
<tr><td>to pi</td><td>due to the code finishing earlier than it was supposed to, dropping some of the data</td><td>by waiting a short amt of time (about 0.03s), and then read using inWaiting()</td></tr>
</table>
</td></tr>
<tr><td>[11/02/2023]
[Celeste]</td><td>SOFTWARE
<u>Tasks Done</u>
1. Created and hosted both the Pi and Pico code on a github repository
a. Integrated github into vscode environment and setup gitlens for a more visual interface
2. Added VL53L0X and VL53L1X library (LiDARs)</td></tr>
<tr><td>[12/02/2023]
[Kae]</td><td>MECHANICAL
<u>Tasks Done</u>
1. discussed and started modeling top camera mount v1 (mounted on servo mount)
 
<u>Design Concepts</u>
● top camera mount
○ height and angle of bottom camera distorts and/or cuts off the shapes of victims and evacuation point
○ top camera used for more reliable identification of victims and evacuation point
<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>top camera mount v1</td><td>● gripper arm may block part of the frame of the camera</td><td>● tilt the top camera mount to look past the gripper arm</td></tr>
</table>
</td></tr>
</table>

[13/02/2023]
[Celeste]

SOFTWARE

Tasks Done

1. Tested first **line track** linetrack.py with the working serial comms
    a. Sent over line track variables through serial (rpm and rotation)
    b. Tuned the linetrack PID

Design concept

- Rotation is a double but the simplicity of the method of serial comms we use, [task, value, task, value], allows for 8 bit unsigned integers as task and value
- The resultant angle (-90 to 90) of the image processing is an integer and offsetted to be 0-180 instead. On the pico side, it reverses this conversion back to -90 to 90 and then dividends it by 90 to obtain a rotation from -1 to 1 of the type double

ELECTRICAL

	Issues	Solution
motor driver	• Realised the capacitor on the motor driver was smoothing out any surges in voltage, which was smoothing out the sharp changes in PID, decreasing reactiveness such that the correction of errors are not instantaneous any longer	• Change motor driver to one without such a large capacitor (MDD3A): make **PCB V2**

[14/02/2023]
[Celeste]

SOFTWARE

Tasks Done

1. Attempted making lidar and multiplexer library to read from lidars and store into variables easily → eventually discarded

	Issues	Solution
Lidar library	• The wire.h included in the self made lidar library seems to be using the default arduino wire and not the pico wire	• Don't use a lidar library, instead implement it directly in the code • Cons: code readability

ELECTRICAL

Tasks Done

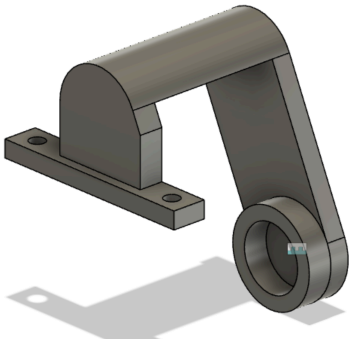
1. Reviewed schematic, edited motor driver connections to have the motors directly connected into the driver's screw terminals instead of through the pcb (will transfer to **PCB v2**)

[15/02/2023]
[Kae]

MECHANICAL

Tasks Done

1. modelled and printed **omniwheel brace v1**
    a. test brace for one of the wheels first



	Issues	Solution
gearbox v7	• wheels under enough pressure to bend gearbox's snap-fit mechanism	• use a brace to provide additional support to the wheels
omniwheel brace v1	• too bulky, may hinder mounting other components • glaring weak points are present no matter the print orientation • difficult to remove brace to access wheel if needed	• change design to not have to reach around the wheel, use different mount holes

[16/02/2023]
[Celeste]

SOFTWARE

Tasks Done

1. Added **lidar readings** into the main code, so it reads from lidars every loop

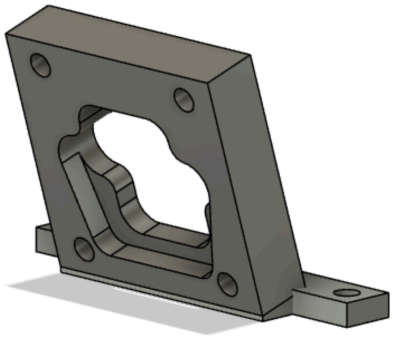
```
//* LIDARS SETUP
const int defaultLidarReading = 200;
//^ VL53L0X
const int L0XNum = 4;
VL53L0X lidarsl0x[L0XNum];
int l0x_readings[L0XNum] = {defaultLidarReading, defaultLidarReading, defaultLidarReading, defaultLidarReading};
const int l0x_pins[L0XNum] = {1, 5, 6, 0};
String l0x_labels[L0XNum] = {"FRONT: ", "FRONT LEFT: ", "LEFT: ", "RIGHT: "}; //for print debugging
You, 2 weeks ago | 1 author (You)
namespace L0X {
    enum L0X { FRONT, FRONT_LEFT, LEFT, RIGHT };
}
//^ VL53L1X
const int L1XNum = 1;
VL53L1X lidarsl1x[L1XNum];
int l1x_readings[L1XNum] = {defaultLidarReading};
const int l1x_pins[L1XNum] = {4};
String l1x_labels[L1XNum] = {"FRONT BOTTOM: "}; //for print debugging
You, 2 weeks ago | 1 author (You)
namespace L1X {
    enum L1X { FRONT_BOTTOM };
}
```

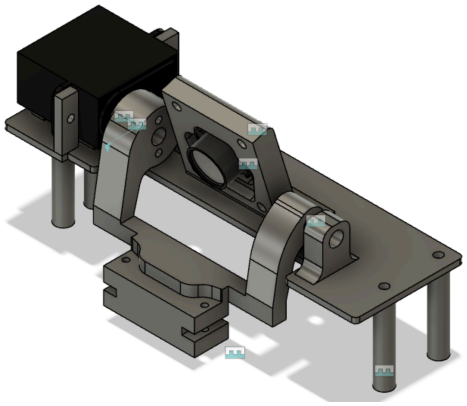
[17/02/2023]
[Celeste, Kae]

MECHANICAL

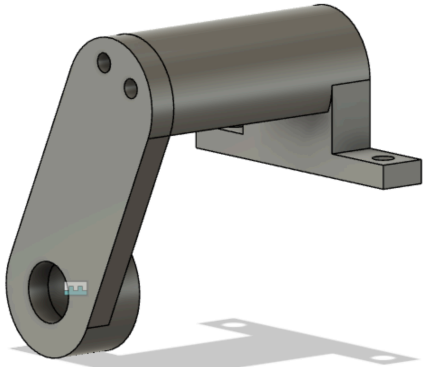
Tasks Done

1. finished modelling and printed **top camera mount v1**

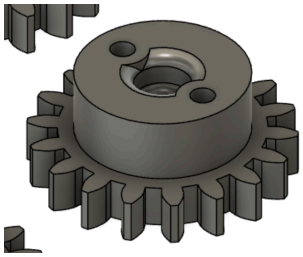




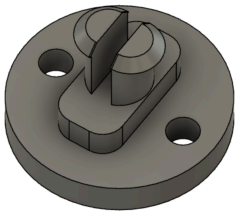
2. modelled and printed **omniwheel brace v2**
    a. added removable part mounted with screws for easier access to wheel

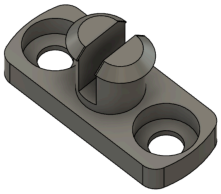


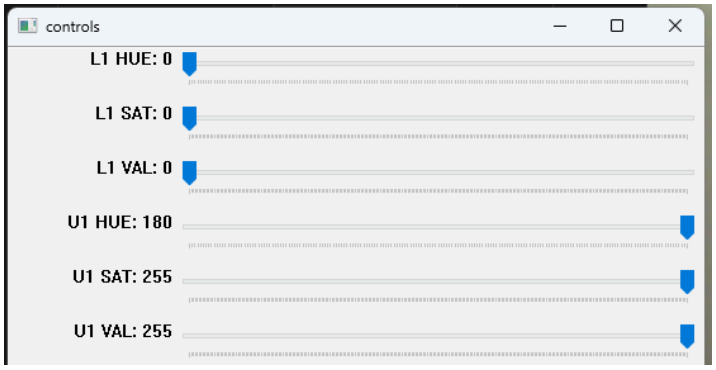
3. modelled pololu wheel gear (**gears v7**) for brace compatibility
    a. extruding part of pololu gear faces inwards



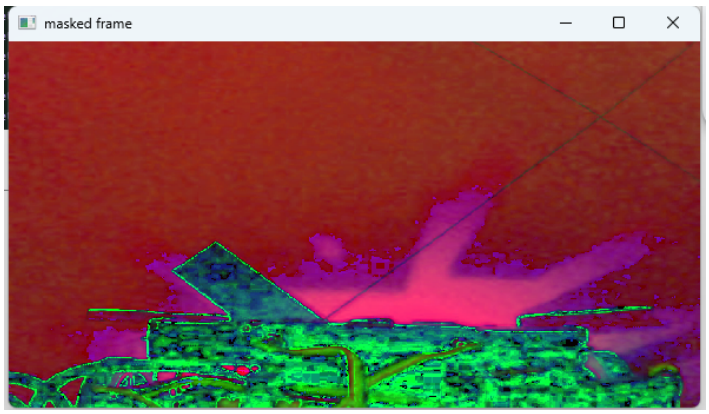
4. modelled **pololu wheel** and **omniwheel brace adaptors v1**\
    a. omniwheel brace adaptor not combined into omniwheel design due to difference in print orientation





<table>
<tr><td></td><td>
ELECTRICAL
<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>I2C bus</td><td>• I2c bus crashes sometimes when running, possibly because of wire connection issues</td><td>• Hot glue all wires at its point of solder (e.g. onto the lidar), to secure and prevent fraying</td></tr>
</table>
</td></tr>
<tr><td>[20/02/2023]
[Celeste]</td><td>
SOFTWARE
<u>Tasks Done</u>
1. Added hsv calibration on the pi, for red, green and blue tuning of values in two parts:
a. Hsvcalib1.py takes a picture of the camera’s current frame
b. Hsvcalib2.py is meant to reside on the local computer to run on the image downloaded from the pi, to tune the values with sliders
<pre>
l_hsv[0] = cv2.getTrackbarPos('L HUE', 'controls')
l_hsv[1] = cv2.getTrackbarPos('L SAT', 'controls')
l_hsv[2] = cv2.getTrackbarPos('L VAL', 'controls')
u_hsv[0] = cv2.getTrackbarPos('U HUE', 'controls')
u_hsv[1] = cv2.getTrackbarPos('U SAT', 'controls')
u_hsv[2] = cv2.getTrackbarPos('U VAL', 'controls')
</pre>


</td></tr>
</table>



Design Concepts

HSV space proves more advantageous as it allows for effective color masking by thresholding using the hue channel. RGB values are also highly sensitive to variations in lighting conditions, making it even more challenging to accurately precisely determine colors.

	Issues	Solution

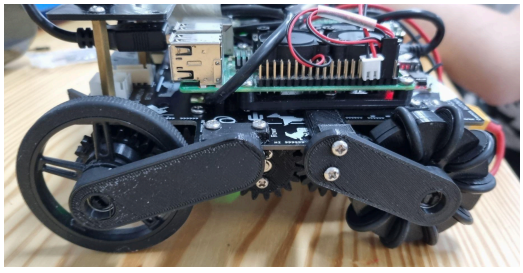
Hsv calibration display	• Window would not hold all the sliders, sliders could not be seen	• Resize window
		`cv2.namedWindow('controls', 2)`
		`cv2.resizeWindow('controls', 550, 10)`

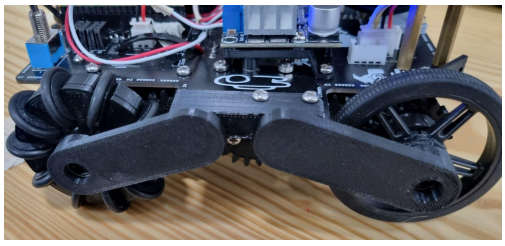
[21/02/2023]
[Celeste, Kae]

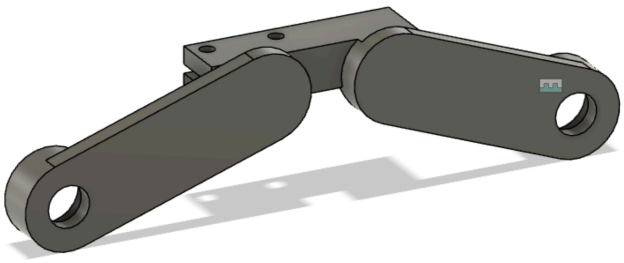
MECHANICAL

Tasks Done

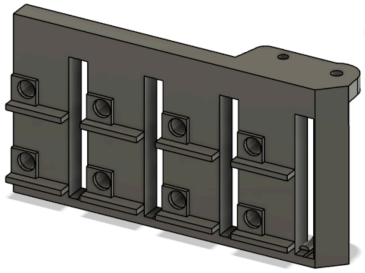
1. modelled, printed and tested **omniwheel brace v3,4** and **wheel brace v1**
    a. brace approach less from the top and more from the side, can use pcb as support
    b. decided on the **wheel brace** design
        i. easier to mount and print







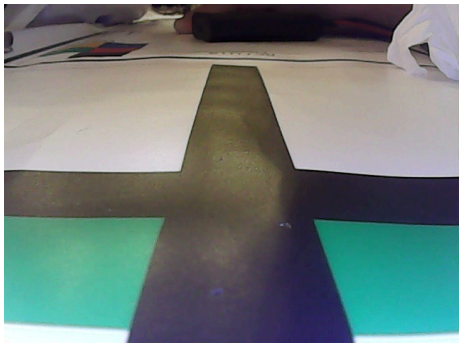
2. modelled and printed **grippers v3** to test lidar (vl53l0x) placement on the gripper
    a. can be used to detect victims



SOFTWARE

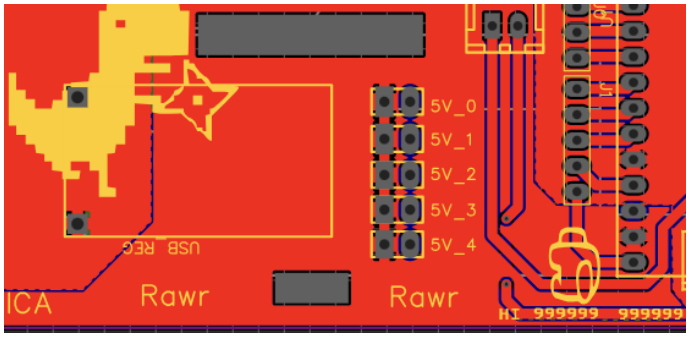
Tasks Done

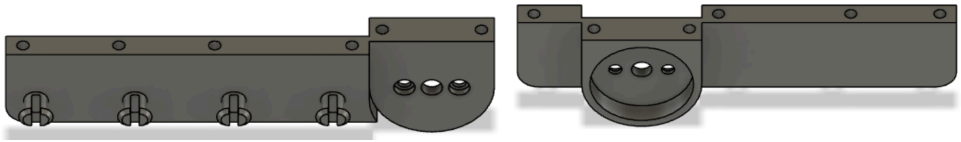
1. **Green squares detection**
    a. Test the presence of green squares thresholding the amount of green in image
    b. Check if the green square is above or below the intersection by checking a sample of area above the green square for black
    c. Identify type of green squares: SINGLE or DOUBLE by seeing if the centroid of black falls within the width of the green mass(es)
    d. If SINGLE, identify if LEFT or RIGHT green square by checking for black on the left and the right of the green square

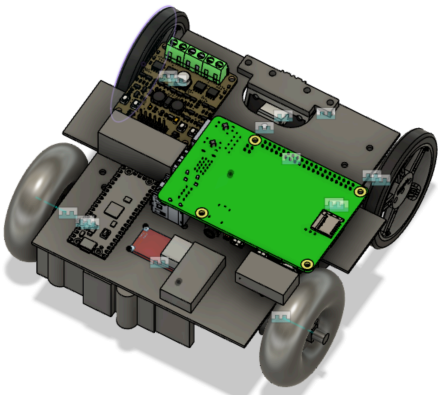
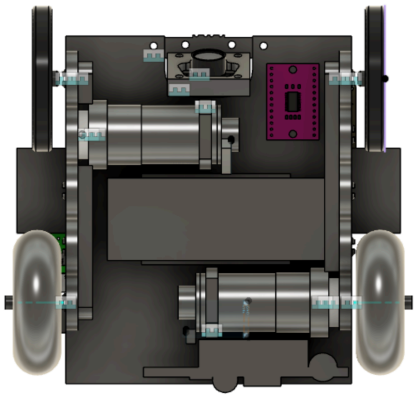
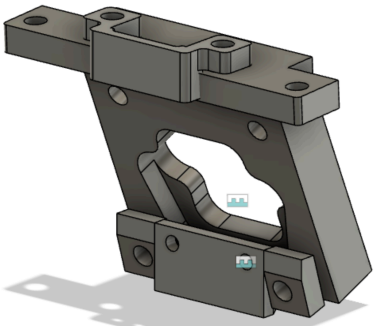


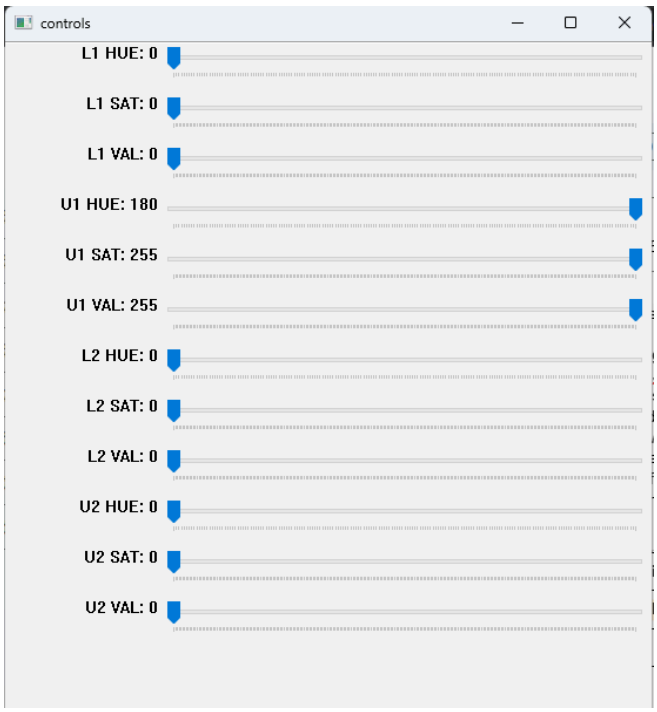
ELECTRICAL

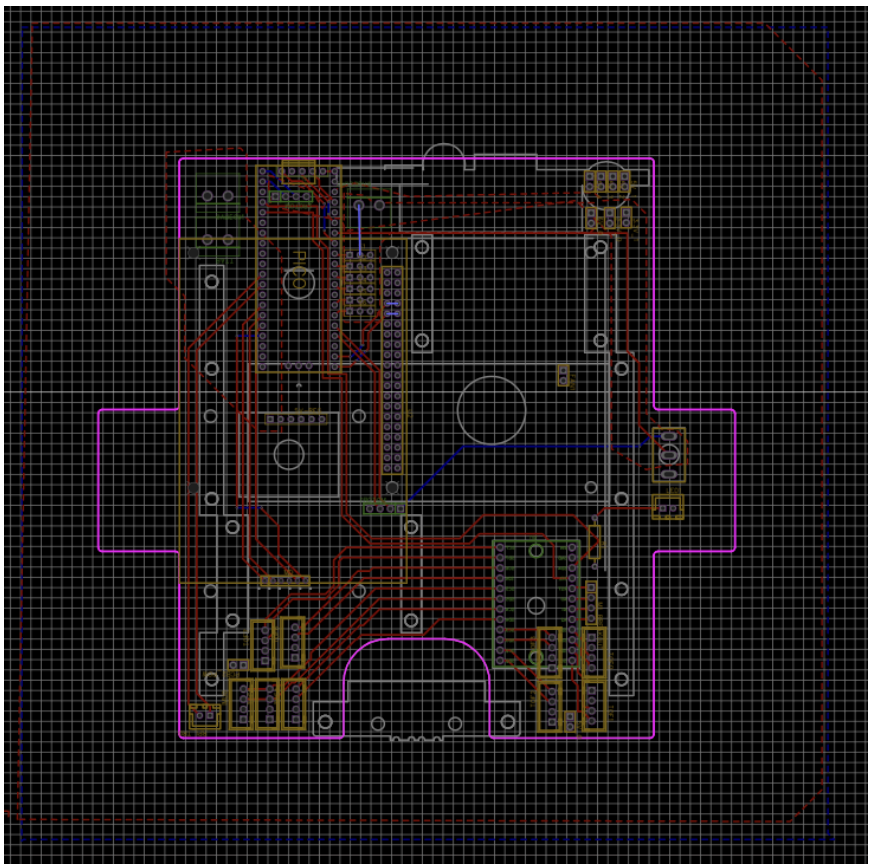
	Issues	Solution
Serial	• Serial connection always transmit the wrong information or rubbish data (could not figure out	• In **PCB V2**, use

		why for 2 days) • Figured out the pi occasionally disconnects because the usb reg is under a lot of tension from the L shaped cable powering it being pushed upwards. This is due to the cable is blocked by the pico 	gpio pins to power it

[22/02/2023] [Celeste]	**SOFTWARE** Tasks Done 1. Tested identifying **green squares** code 2. Integrated green squares into main code with with linetrack so that it automatically identifies green squares
PCB V2	
[23/02/2023] [Kae]	**MECHANICAL** Tasks Done 1. modelled 23.04mm pitch diameter for **gears v8**  2. modelled new **gearbox v8** (Fig 2) a. new staggered motor arrangement b. combined front motor mount into gearbox c. creation of **back motor mount v1** 
	ELECTRICAL Tasks Done 1. did preliminary arrangement and outline in **PCB v2** CAD

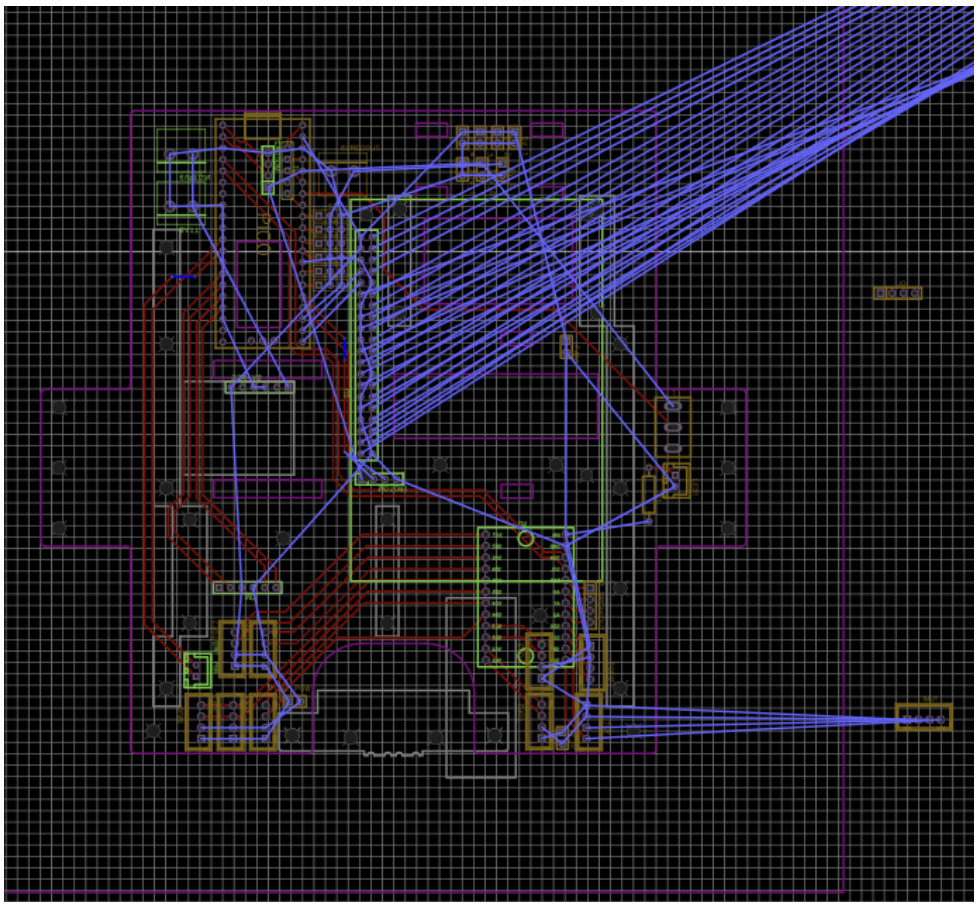
	 
[24/02/2023] [Kae, Minjie]	**MECHANICAL** Tasks Done 1. modelled **bottom camera mount v4** a. removed unnecessary screw holes b. increase thickness of led mount to prevent break c. tilted camera to 20 degrees 
	SOFTWARE Tasks Done 1. Added a second spectrum to **hsv calibration** as red lies on the first part (about 0 to 10) and the last part (about 350 - 360). a. The masks are added together to allow for tuning of red values to have 2 parts

	
[25/02/2023] [Celeste, Kae, Aseera, Minjie]	**ELECTRICAL** Tasks Done 1. Schematics a. Made custom component for MDD3A motor driver in EasyEDA b. Changed Raspberry Pi to be powered through GPIO pins c. Second wire for teensy RX/TX d. Fixed previous mistake of 3.3V extra pins being 6V e. Inserted JST for LED f. Motor driver motor outputs removed, to be wired directly instead of through PCB 2. First draft of **PCB V2** a. Rearranged components in PCB v2 CAD b. Traced the components

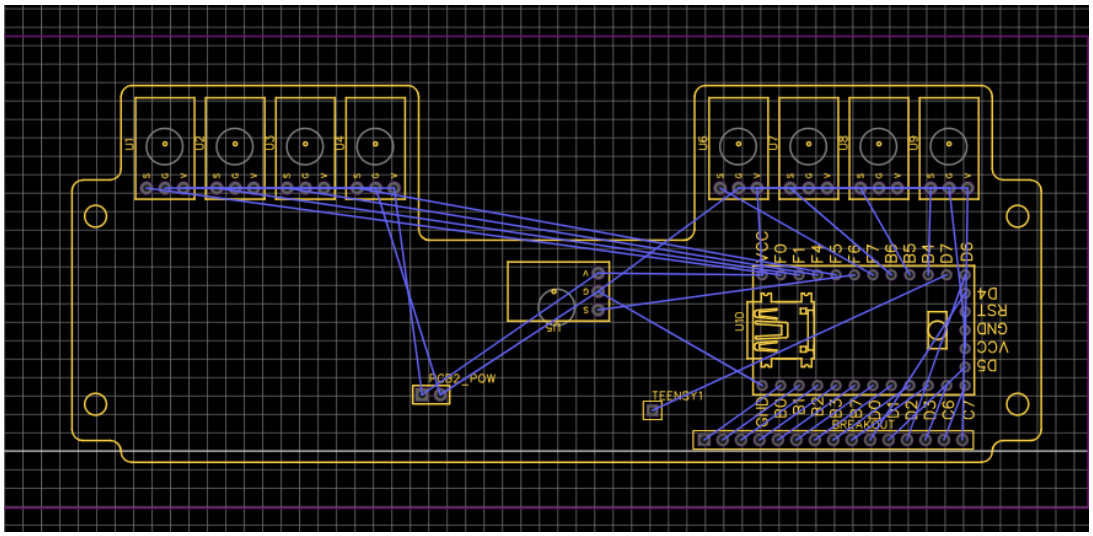
	
[26/02/2023] [Celeste, Minjie]	**SOFTWARE** Tasks Done 1. Edited **drivebase library** a. Added public method getPin for motor class to access private attributes pin1 and pin2 2. **Green squares** a. Added code on the pico's side to react to green squares (turn with a fixed rotation for single, and a coded 180 for double green) b. Added the transfer of a new type of variable via serial, task, to tell the pico what is happening (LINETRACK, LEFT GREEN, RIGHT GREEN, etc.)

	c. Added states “green_now”, in pi side to determine if it is currently in a green square. Only stop turning when the green square is out of sight
	`#* JUST FOUND GREEN SQUARE PREVIOUSLY`
	
	`#~ Turn while still seeing green`
	`elif gs_now and gs_sum < 1000:`
	`  if curr.name == "DOUBLE_GREEN":`
	`  if np.sum(mask_black) > 18000000:`
	`  gs_now = False`
	`  else:`
	`  to_pico = [253, curr.value]`
	`  else:`
	`  gs_now = False`
	3. **Red line**
	a. Added red detection based on amount of red in image
	b. Added red state in the pi side
	`elif red_now == True:`
	`  red_now = False`
	`  rpm = set_rpm`
	c. Added code on pico’s side to react to red line (stop)

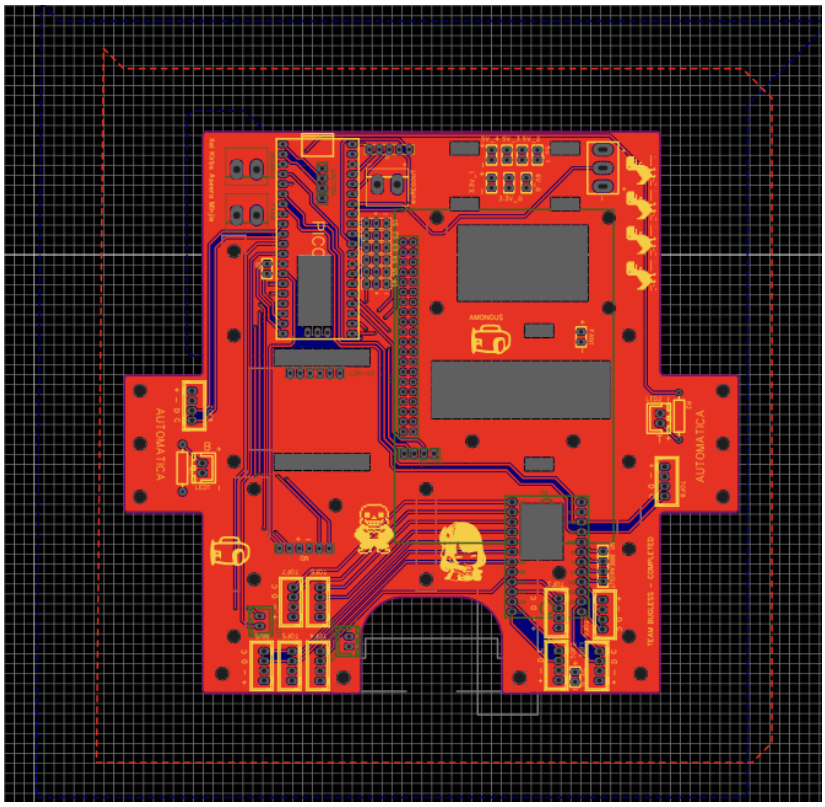
[27/02/2023] [Celeste, Kae, Aseera, Minjie]	**ELECTRICAL** Tasks Done 1. Second draft of **PCB V2**

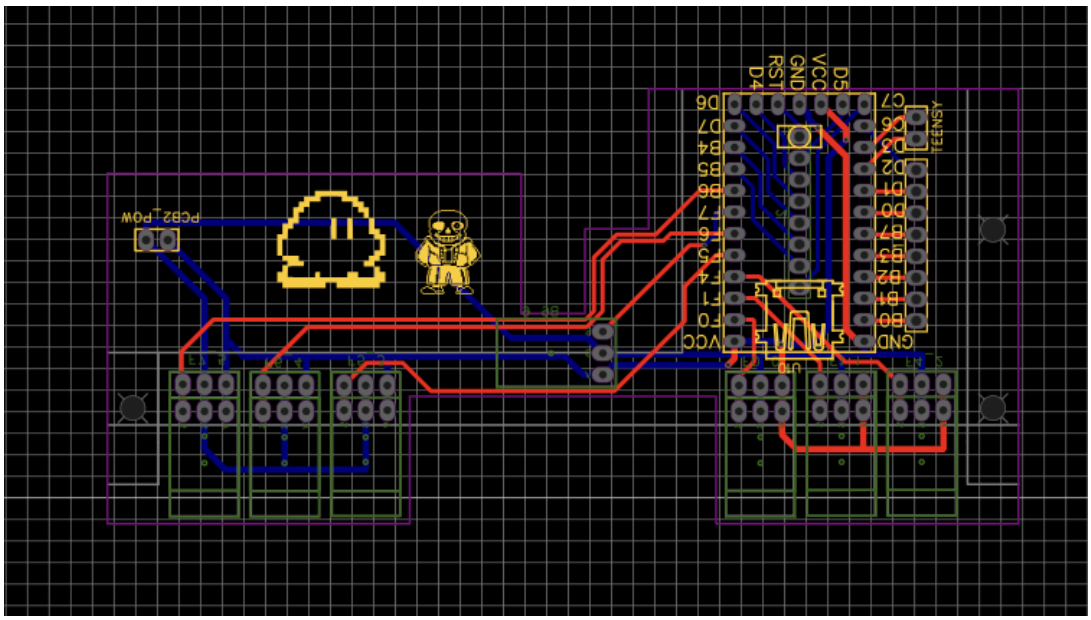


2. Schematics of **Bottom PCB V1**
3. Draft 1 of **Bottom PCB V1**
    a. TEMT6000s are connected to the Teensy 2.0. The screw holes are joined to the bottom of the gearbox



<u>Design Concepts</u>

	Light sensor array plate is to allow the robot to cross the 135 turn that it has trouble doing. Since the pico has a lack of pins, a teensy interfaces between it and the pico and it is planned to use serial comms with the pico on UART 1
[01/03/2023] [Celeste, Kae, Aseera, Minjie]	**SOFTWARE** Tasks Done 1. Started coding evac **identifying balls** using **hough circles** a. Grayscale the image b. Apply canny edges to find the edges of the image c. Apply hough circles to identify those in a circle d. Refer to here for diagrams 2. Tuned hough circles parameters (min dist between the balls, edge thresholds, etc.)
	ELECTRICAL Tasks Done 1. Third and final version of **PCB V2**  2. Final version of **Bottom PCB V1**



3. Added silkscreen designs

ADMIN

Tasks Done

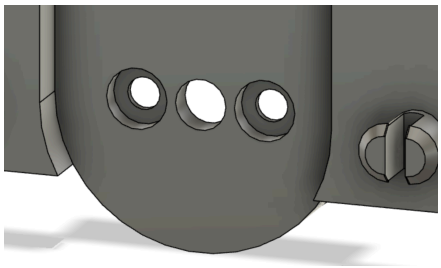
1. Ordered **PCB V2** and **Bottom PCB V1** through JLCPCB

[02/03/2023]
[Celeste, Kae, Aseera]

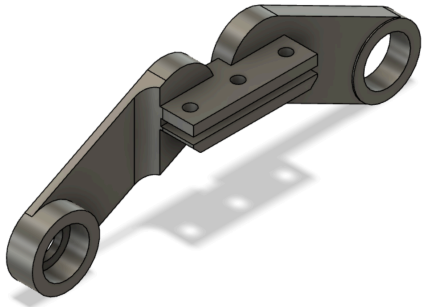
MECHANICAL

Tasks Done

1. expanded recessed holes in **gearbox v9** for screw heads to fit properly



2. modelled and printed **wheel brace v2** for **PCB v2**



	Issues	Solution

	wheel brace v2	• need to minimise robot width, brace limited to around 2mm thickness, worried about stability	• increase printing wall thickness to 4mm, print face downwards

SOFTWARE

Tasks Done

1. Attempted to write a **gripper** library

```
void claw_open() {
    servos_angle[Servos::RIGHT] = 130;
    servos_angle[Servos::LEFT] = 0;
    servos_change = true;
}

void claw_close() { //ball
    servos_angle[Servos::RIGHT] = 25;
    servos_angle[Servos::LEFT] = 105;
    servos_change = true;
}

void claw_close_cube() {
    servos_angle[Servos::RIGHT] = 10;
    servos_angle[Servos::LEFT] = 120;
    servos_change = true;
}

void claw_halfclose() {
    servos_angle[Servos::RIGHT] = 95;
    servos_angle[Servos::LEFT] = 35;
    servos_change = true;
}

void claw_up() {
    servos_angle[Servos::ARM] = 0;
    servos_change = true;
}

void claw_service_up() {
    servos_angle[Servos::ARM] = 140;
    servos_change = true;
}       You, 4 weeks ago • tcs readings

void claw_down() {
    servos_angle[Servos::ARM] = 180;
    servos_change = true;
}
```

2. After issues, put in main code:

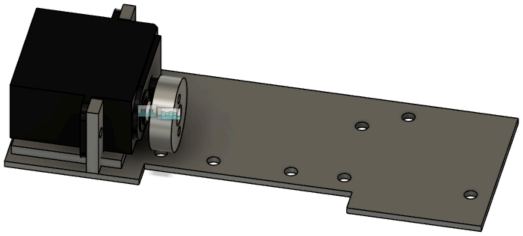
	Issues	Solution
gripper library	Gripper library did not work as there was issues with library and the arduino pico core	Copied all the functions over into main.ino

[03/03/2023]
[Kae]

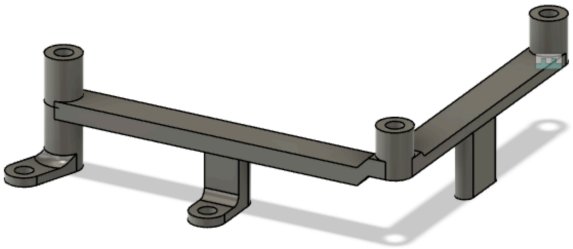
MECHANICAL

Tasks Done

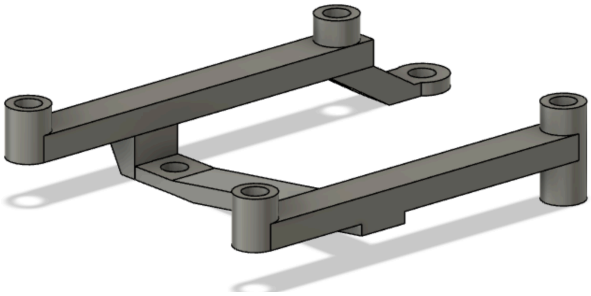
1. started modelling **servo mount v2** for **PCB v2**



2. modelled new **rpi 4 mount v1** for upside down Pi 4



3. modelled and printed **mdd3a mount v1**



	Issues	Solution
servo mount v2	• not sure where to put the screw holes to mount	• screw standoffs into gearbox instead of using screws and nuts against the pcb
rpi 4 mount v1	• screwed into the PCB at 2 points, may not be stable enough	• add more screw holes in later versions

[04/03/2023]
[Celeste, Aseera, Minjie]

SOFTWARE

Tasks Done

1. Green squares made to be 0.5 rotation instead of a greater 0.8 rotation (0.5 is a one wheel turn to the right, while 0.8 is a faster rate of turn)
2. **Rescue kit navigation** started getting written
    a. First method: Used time
        i. Detecting the centroid of blue and using trigonometry to obtain angle of rotation every single loop
        ii. Logging the time spent at each specific loop and the rotation value
        iii. By using the change in heading and position at every single moment, calculate the final heading and x-y displacement relative to original heading and position
        iv. After picking up rescue kit, use trigonometry to face the original position
        v. Drive straight to original position

vi. Return to original heading by correcting for both the final change in heading as well as the change in heading when facing the original position

b. Second method: Storing the movements in an array
   i. Detecting the centroid of blue and using trigonometry to obtain angle of rotation every single loop
   ii. Log the rotations at every single loop
   iii. After picking up rescue kit, reverse movements one by one and remove them from the array until the array is empty

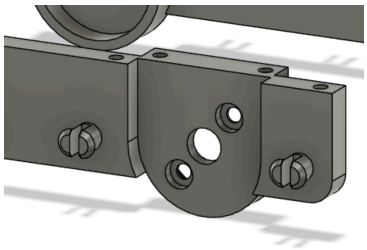
	Issues	Solution
First method of **rescue kit detection**	• The math was difficult to implement • There were many inaccuracies between the wanted rotation and the actual rotation. Multiplied by loop time, this was accentuated greatly	Use an array instead (method 2)
Second method of **rescue kit detection**	• It was not the most accurate either, and the robot ends up over rotating	Use a fixed centering method (method 3 done later)

[06/03/2023]
[Celeste, Kae]

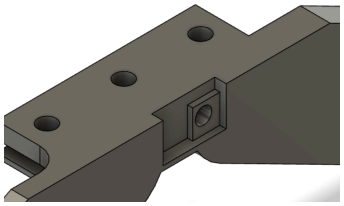
MECHANICAL

Tasks Done

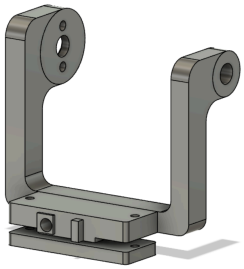
1. Rotated front motor to 50 degrees in **gearbox v10**
   a. move the connector to make more space for battery



2. added side lidar (vl53l0x) mounts to the **wheel brace v3**
   a. added raised 0.9mm rectangle around the screw hole due to components sticking out from the back of the l0x



3. added hole for lego axel on the **servo mount v3**
4. added more mounting holes (just in case) in **servo mount v3**
5. started modelling **gripper arm v2**
    a. L-shape arm to fit around servo mount better
    b. added l0x mount hole at the front



	Issues	Solution
wheel brace v3	• hole for l0x not deep enough, might catch on objects • l0x rotates, falls out	• thicken brace
gripper arm v2	• L-shape arm more prone to cracking	• rotate orientation for 3d printing so stress points are not along layer lines

SOFTWARE

Tasks Done

1. Turns more after **green squares**

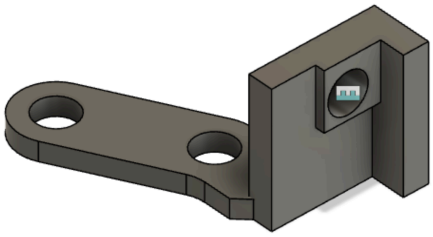
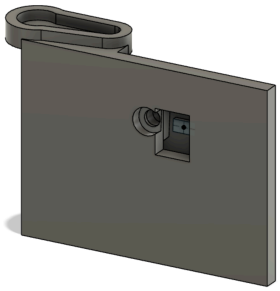
```
if (curr == 1) {
    //~ enter post left green mode after minimum turn time
    if (millis() - startGSMillis > 200) {
        curr = 21;
        lostGSMillis = millis(); }
} else if (curr == 2) {
    //~ enter post right green mode after minimum turn time
    if (millis() - startGSMillis > 200) {
        curr = 22;
        lostGSMillis = millis(); }
```

```
case 21: //^ post left green
    send_pi(0);
    Robawt.setSteer(rpm, -0.6);
    break;

case 22: //^ post right green
    send_pi(0);
    Robawt.setSteer(rpm, 0.6);
    break;      You, 4 weeks ago
```

2. Made getDist functions in vroom library for accurate turns using encoders
    a. getDist returns the encoder values calculated to be in cm, so that we know exactly what distance we traveled and when to stop

	Issues	Solution
green squares	Robot not turning enough after green square	Include a period of time where the robot continues turning

[07/03/2023] [Kae, Aseera]	**ELECTRICAL** Tasks Done 1. Soldered half of **PCB V2** (All the headers and connectors but not the regs)
[08/03/2023] [Kae, Aseera, Minjie]	**MECHANICAL** Tasks Done 1. modeled and printed left and right **diagonal lidar mounts v2** a. designed for vl53l0x  2. increased lego mounting hole wall thickness on **servo mount v4** 3. decrease length of arm in **gripper arm v3** to have a bigger gripper 4. modeled **grippers v4** to have recessed hole for white servo arm 5. added hole and l0x mount on **grippers v4** 
	SOFTWARE Tasks Done 1. **Rescue kit navigation**: Method 3 centering on kit first a. Turning to the cube with a fixed one wheel turn left or right based on position of cube b. When the centroid of the cube is centered (horizontally less than 60 pixels from the center of the frame), record turn angle c. Move forward d. After picking up kit, reverse the same distance as recorded by the motor encoders with getDist() e. Turn back with previous recorded angle

```
#~ If the block isn't centred
if abs(finalx) >= 60: #^ DOM: not refined yet; consider doing abs(...) >= 100
    if cx_blue < centre_x_lt: #if rescue kit is to the left of the centre of the frame
        rotation = -45 # fixed rotation of the bot
        rpm_lt = 20
    elif cx_blue > centre_x_lt: #to the right
        rotation = 45
        rpm_lt = 20

#~ Else if the block is (relatively) centred:
else:
    curr = Task.BLUE
```

	Issues	Solution
rescue kit code	Many issues in rescue kit code: • Robot's turning back is not accurate • Robot's moving back not accurate • Robot does not stop when it is supposed to	Missing break; in case statement 😢

[10/03/2023]
[Celeste, Kae]

ELECTRICAL
Tasks Done
1. Soldered regs onto the new PCB V2 (3.3V, 5V and connectors for 6V)

(busy day in SSEF, not free since tues)

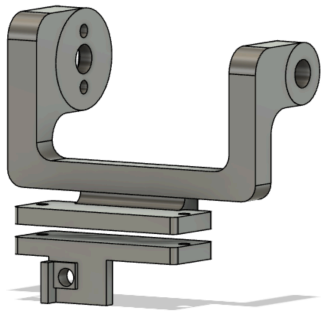
[11/03/2023]
[Celeste, Kae]

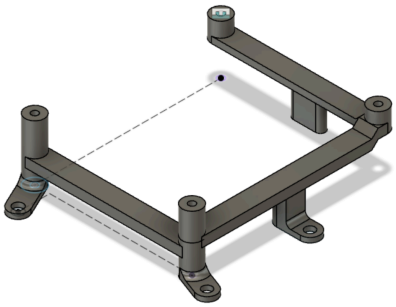
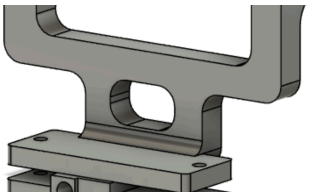
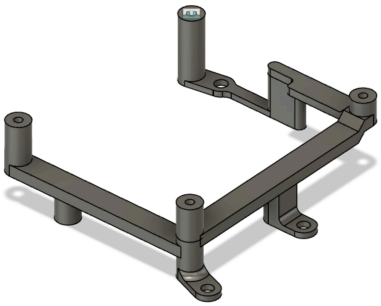
MECHANICAL
Tasks Done
1. Assembled robot
2. modeled **bottom camera mount v5** to reduce thickness again

ELECTRICAL
Tasks Done
1. Finished soldering the robot's new **PCB V2** (Wires)

[12/03/2023]
[Kae]

MECHANICAL
Tasks Done
1. decrease height of arm in **gripper arm v4**



<table>
<tr><td></td><td>2. added more screw holes to the rpi 4 mount v2
  </td></tr>
<tr><td rowspan="2">[13/03/2023]
[Celeste, Kae, Aseera, Minjie]</td><td>MECHANICAL
<u>Tasks Done</u>
1. changed shape of body in gripper arm v5
a. no longer clashes with gripper servo
2. increased height in gripper arm v5
 
3. added more screw holes in rpi 4 mount v3
  </td></tr>
<tr><td>SOFTWARE
<u>Tasks Done</u>
1. Ball detection
a. Hough circles parameters returned for finer control
<table><tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr><tr><td>ball detection</td><td>Using bottom camera,
• Difficult to spot when it only is able to see half of the ball (due to the angle needed for linetrack)
• Noise from outside of the evac zone in the camera affects the</td><td>Install a top down camera to see the ball against a flush white background</td></tr></table></td></tr>
</table>

		ball detection → many false positives	

[14/03/2023]
[Celeste]

SOFTWARE

Tasks Done

1. **Deposit servos** added into code

```
void alive_up() {
    servos_angle[Servos::ALIVE] = 60;
    servos_change = true;
}

void alive_down() {
    servos_angle[Servos::ALIVE] = 140;
    servos_change = true;
}

void dead_up() {
    servos_angle[Servos::DEAD] = 65;
    servos_change = true;
}

void dead_down() {
    servos_angle[Servos::DEAD] = 10;
    servos_change = true;
}

void sort_alive() {
    servos_angle[Servos::SORT] = 130;
    servos_change = true;
}

void sort_neutral() {
    servos_angle[Servos::SORT] = 90;
    servos_change = true;
}

void sort_dead() {
    servos_angle[Servos::SORT] = 50;
    servos_change = true;
}
```

2. Entire code assembled together (line track and evac)

```
while True:
    if top_stream.stopped or bot_stream.stopped:
        break

    received_task = receive_pico()
    if received_task != -1:
        pico_task = int(received_task)

    if pico_task == 0:
        print("Linetrack")
        task0_lt()
    elif pico_task == 1:
        print("Evac looking for ball")
        task_1_ball()
    elif pico_task == 2:
        print("Evac looking for alive deposit")
        task_2_depositalive()
    elif pico_task == 3:
        print("Evac looking for dead deposit")
        task_3_depositdead()
    elif pico_task == 4:
        print("Looking for linetrack")
        task4_backtolt()
    elif pico_task == 5:
        print("Obstacle turning left, looking at right of camera for line")
        task5_leftlookright()
    elif pico_task == 6:
        print("Obstacle turning right, looking at left of camera for line")
        task6_rightlookleft()
    elif pico_task == 9:
        print("Switch off")
        task0_lt()
    else:
        print("Pico task unknown:", pico_task)
```

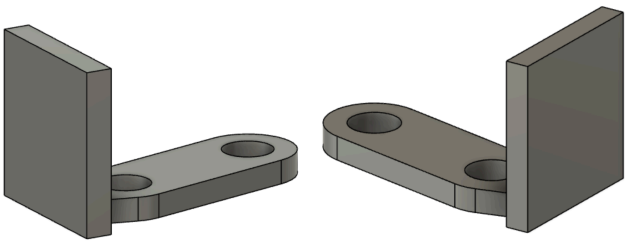
	Issues	Solution

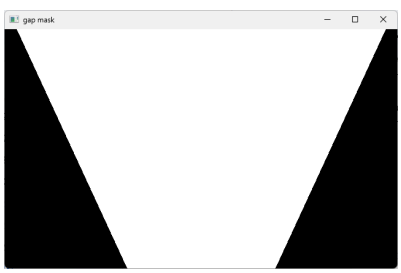
<table>
<tr>
<td></td>
<td>
<table>
<tr>
<td>Assembling code</td>
<td>The robot is unable to reset when using a linear method, which is to break out of the while true loops after the trigger happens. This is because python does not have a goto

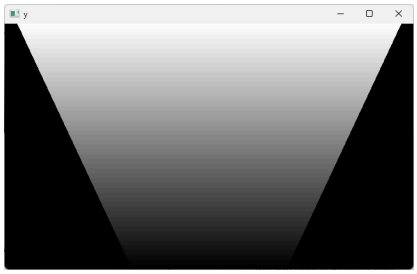
```
while True:
    #insert linetrack logic
    if silver_line:
        break

while True:
    #insert evac logic
    if timer > 300:
        break

while True:
    #looking for alive deposit
    if green_sum > 25000:
        break
```


In this case, if the switch were to be flipped after the timer triggers and it is looking for the alive deposit, the code cannot be restarted back to linetrack</td>
<td>Call the different functions when at different parts of the mission (as shown in diagram above)</td>
</tr>
</table>
</td>
</tr>
<tr>
<td>[15/03/2023]
[Celeste, Kae, Minjie]</td>
<td>
MECHANICAL
<u>Tasks Done</u>
1. increased wheel brace thickness (wheel brace v4) so l0x doesn't stick out
2. modelled diagonal lidar mounts v3
    a. changed angle of lidars to 30 degrees instead of 45 degrees
3. removed screw holes from diagonal lidar mounts v3, can just hole glue

4. decreased height of grippers, moved gripper l0x positions more towards the centre in grippers v5
</td>
</tr>
<tr>
<td></td>
<td>
SOFTWARE
<u>Tasks Done</u>
1. Added a different matrix to use on line track when a line gap is detected
</td>
</tr>
</table>





2. Changed the crop height so that the horizon is cropped out
3. Added erode dilate onto linetrack

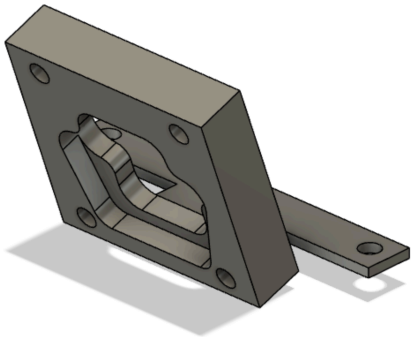
	Issues	Solution
linetrack	The tile are marked out by thin lines, which affects linetrack as they are black too	Erode the lines away, then dilate the image to preserve original thickness and size of notable lines
linetrack	Noise at the top of the image (outside of the mat) was affecting line track	Crop out anything above the horizon line
Line gap	Robot latches on to nearer lines to the side	Create an inverted triangle mask (shown above) so that it ignores anything to the side, and then weigh further pixels more heavily until line gap is over

[16/03/2023]
[Celeste, Kae, Aseera, Minjie]

MECHANICAL

Tasks Done

1. modelled and printed **top camera mount v2**
    a. sticks out more to the front



2. modelled and printed **servo mount v5**
    a. to follow changes to top camera mount
3. modelled and printed **bottom camera mount v6**

a. changed bottom camera angle to 22 degrees
b. moved bottom camera up

4. added hole on **grippers v6** for tapping screw to screw into servo
5. added extra wall to prevent rotation for gripper l0x in **grippers v6**

	Issues	Solution
bottom camera v6	• l1x mounted on camera is scraping against speed bumps	• move bottom camera upwards • reduce tilt angle of camera to compensate

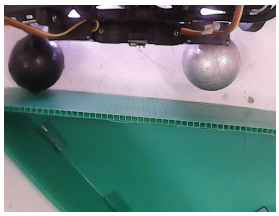
SOFTWARE

Tasks Done

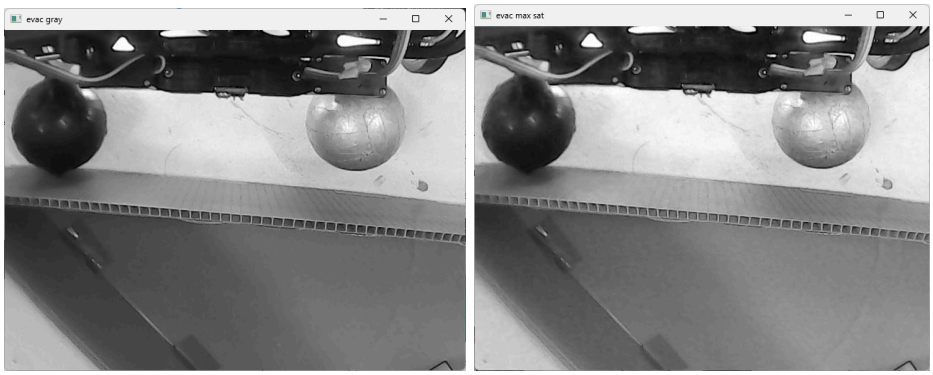
1. Started using gripper functions to pick the balls
2. Read from top evac camera
3. Testing hough circles on both max sat and grayscale on top camera

Design concept: **Hough circles**

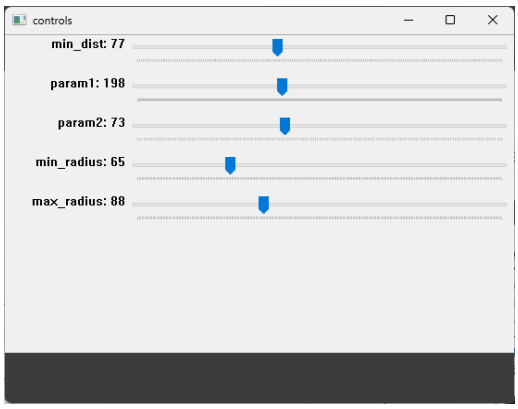
Original image:



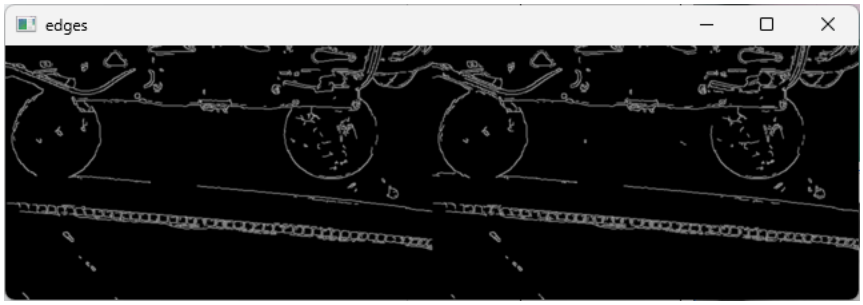
Left: Grayscale, Right: maximum saturation



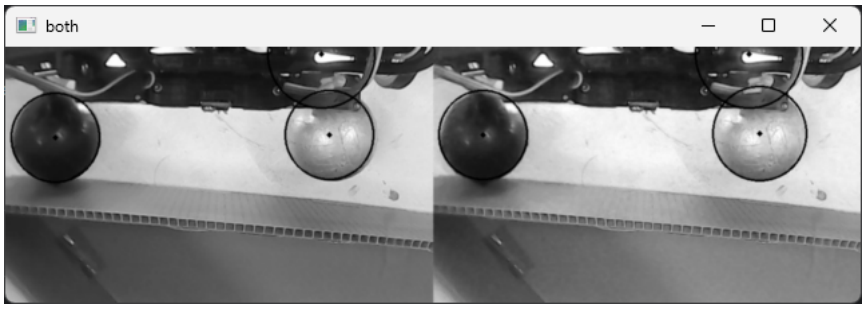
Parameters for hough circles:



After canny edges: (left - grayscale, right - max saturation)



After houghcircles: (left - grayscale, right - max saturation)



	Issues	Solution
hough circle	Balls were being falsely detected due to the lines on the corrugated board	Attempted to use the maximum saturation instead of a grayscale mask. (results shown above)

[20/03/2023] [Celeste]	**SOFTWARE** Tasks Done 1. Tested pico to pi comms in integrated line track functions
[21/03/2023] [Celeste]	**SOFTWARE** Tasks Done

1. **Servo** positions tuned to be initialized in their starting positions

```
//* SERVOS SETUP
double servos_angle[6] = {0, 140, 180, 0, 130, 90}; //basic states initialised
const double servos_max_angle[6] = {180, 180, 180, 300, 300, 300};
const int servos_pin[6] = {27, 26, 21, 20, 2, 22};
bool servos_change = false;
You, 4 weeks ago | 1 author (You)
namespace Servos {
    enum Servos { DEAD, ALIVE, ARM, LEFT, RIGHT, SORT};
}
```

2. **Gripper** movements tuned so that the claw closes differently on the ball (4-5cm) and the kit (smaller than 4cm)

```
void claw_close() { //ball
    servos_angle[Servos::RIGHT] = 25;
    servos_angle[Servos::LEFT] = 105;
    servos_change = true;
}

void claw_close_cube() {
    servos_angle[Servos::RIGHT] = 10;
    servos_angle[Servos::LEFT] = 120;
    servos_change = true;
}
```

	Issues	Solution
gripper	The kit was unable to be picked up as claw was not closing enough, but it was already very tight for the ball	Use 2 different functions for the different objects

[22/03/2023]
[Aseera]

SOFTWARE

Tasks Done

1. Obstacle code started
2. Detection of see line after obstacle (to signal to robot to turn back)
    a. Based on the amount of black in a specific area

[23/03/2023]
[Celeste, Aseera]

MECHANICAL

	Issues	Solution
camera	Too cramped for wire JST on bottom camera to fit against back motor mount	Need to flip camera upside down → flip frame horizontally and vertically

SOFTWARE

Tasks Done

1. Added debug functions determined by preprocessor defined variables for easy debugging

```
//* DEBUG SETTINGS
//^ Runs with code
#define debug_serial 0
#define debug_led 1
#define debug_looptime 0
#define debug_lidars 1
#define debug_curr 1
#define debug_distance 1
#define debug_evac_speedup 1

//^ Runs without normal code
#define loop_movetime 0
#define loop_movedistance 0
#define loop_pickball 0
#define loop_pickcube 0
#define loop_deposit 0
#define isthishardwarestop 0
//^ Force events
#define debug_evac 0
#define debug_deposit 0
#define debug_evac_exit 0
```

2. Flipped camera stream horizontally and vertically

```
frame_org = bot_stream.read()
frame_org = cv2.flip(frame_org, 0)
frame_org = cv2.flip(frame_org, 1)
```

3. Tuned distance from gripper the ball must be to pick it up i.e. see_ball()

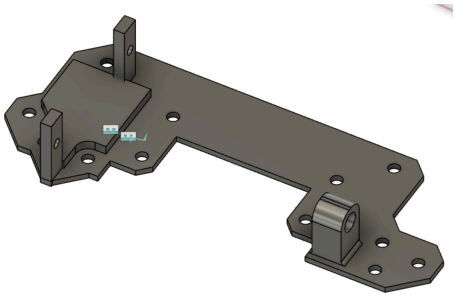
	Issues	Solution
camera	Camera stream kept crashing	In multithread, added magic line: self.vcap.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc('M', 'J', 'P', 'G'))

[24/03/2023]
[Celeste]

MECHANICAL

Tasks Done

1. removed extra empty space from **servo mount v6**
    a. easier to acces JSTs below the servo mount
2. added holes in **servo mount v6** to mount **side lidar mounts v1**
    a. need to mount from the bottom for the right side

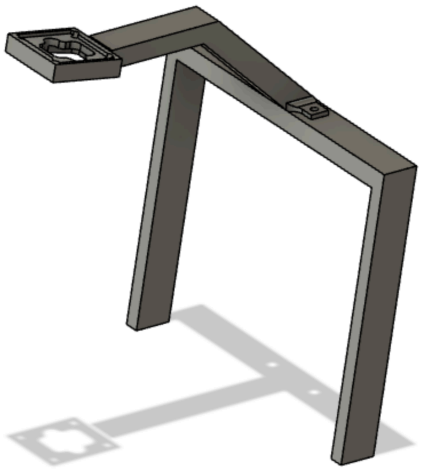


3. started modelling **side lidar mounts v1**
4. move gripper servos closer together in **gripper arm v6**

SOFTWARE

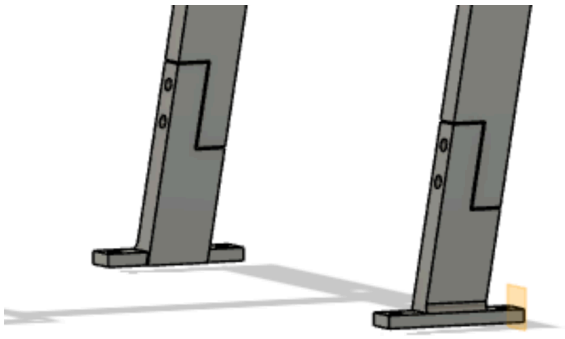
Tasks Done

<table>
<tr><td></td><td>
1. Attempted to use serial to talk to teensy
    a. Used binary data transfer instead
<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>UART1</td><td>Hardware serial UART1 will crash pico every time it is used (
Already tested on different picos, and different laptops uploading only void setup() { Serial2.begin(); })</td><td>Since we only need 4 states (turn left, go straight, turn right, nothing), we can use binary, 00, 01, 10, 11 to convey this information by just manipulating the two pins using HIGH and LOW</td></tr>
</table>
</td></tr>
<tr><td>[25/03/2023]
[Kae]</td><td>
MECHANICAL
<u>Tasks Done</u>
1. started design on top camera mount v3
    a. camera facing almost directly downwards
2. started modelling handle v1 to mount top camera mount v3



<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>handle v1</td><td>• not sure where to mount handle
• might block/restrict deposit
• might break due to weight of break</td><td>• mount with brace mounting holes
• maximise handle size to maximum robot dimensions
• increase wall thickness for 3d printing</td></tr>
</table>
</td></tr>
<tr><td>[26/03/2023]
[Kae]</td><td>
MECHANICAL
<u>Tasks Done</u>
</td></tr>
</table>

1. split handle into 2 bases and 1 main body for easy removal



[27/03/2023]
[Celeste, Kae, Aseera, Minjie]

MECHANICAL

Tasks Done

1. increased distance between gripper servos on **gripper arm v7**

	Issues	Solution
cardboard hat	The hat prevents the robot from being able to go close enough to the obstacle to sense it	Remove the hat

SOFTWARE

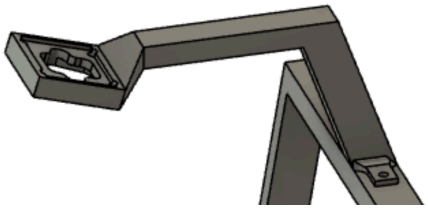
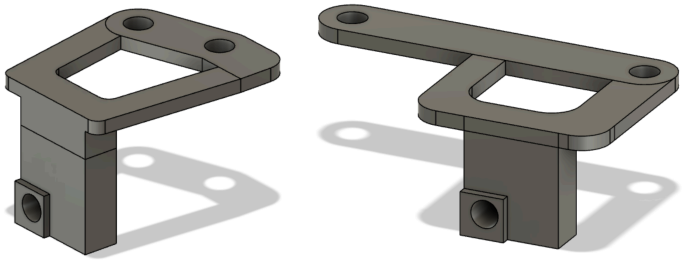
Tasks Done

1. Rescue kit works reliably by solving some bugs in the code
2. Used enum for tasks to clean code (e.g. LINETRACK, RED, etc.)

```
#* VARIABLE INITIALISATIONS
...
class Task(enum.Enum):
    #~ Linetrack
    EMPTY = 0
    LEFT_GREEN = 1
    RIGHT_GREEN = 2
    DOUBLE_GREEN = 3
    RED = 4
    TURN_LEFT = 5
    TURN_RIGHT = 6
    TURN_BLUE = 7
    BLUE = 8
    LINEGAP = 11
    # SILVER = 12
    #~ Evac
    NOBALL = 20
    BALL = 21
    DEPOSITALIVE = 22
    DEPOSITDEAD = 23
    EMPTYEVAC = 24
    FINDINGLINE = 25
```

3. Attempted to detect silver tape
    a. Method 1: using standard deviation to look for the amount of noise
    b. Method 2: use a red LED and look for concentration of red

	Issues	Solution
silver tape	Silver tape using top camera just looks like plain noise, which is	Use a red LED

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>indistinguishable from normal linetrack because normal linetrack is very noisy as well</td><td></td></tr>
<tr><td>Silver tape</td><td>Using a red LED is drowned out by some strong lighting</td><td>Add a cardboard hat to block out the above lighting</td></tr>
<tr><td>distance</td><td>Distance was wrong (currDist - prevDist) should have been negative when going backwards, but was tested for > positive number</td><td>Use fabs() to ensure no confusion, and test only absolute distance instead of displacement</td></tr>
</table>
</td></tr>
<tr><td>[28/03/2023]
[Celeste, Kae, Aseera]</td><td>
MECHANICAL
<u>Tasks Done</u>
1. made top camera moount v4 taller

2. removed l0x mount from grippers v7
3. finished modelled and printed side lidar mounts v1

SOFTWARE
<u>Tasks Done</u>
1. Added left 135 and right 135 detected using the TEMTs on the bottom plate, interfaced with the Teensy 2.0 to the pico
<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>135</td><td>Once 135 has been added, the robot seems to do nothing BUT the 135s because it was triggering too easily.
• Condition was as long as the middle and the sidemost sensor was black.
• However, there is no way to tighten this condition further because of the lack of</td><td>Implemented a time out so that it cannot trigger 135s in a row</td></tr>
</table>
</td></tr>
</table>

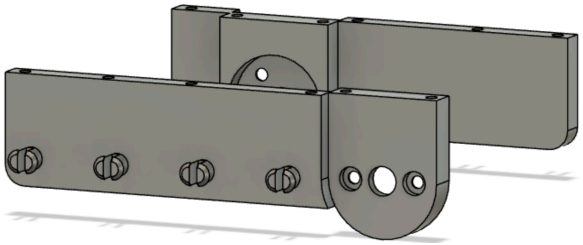
		resolution on the bottom of the robot (only 6 sensors)	

[29/03/2023]
[Kae]

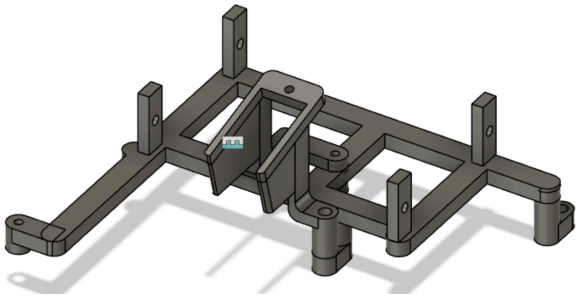
MECHANICAL

Tasks Done

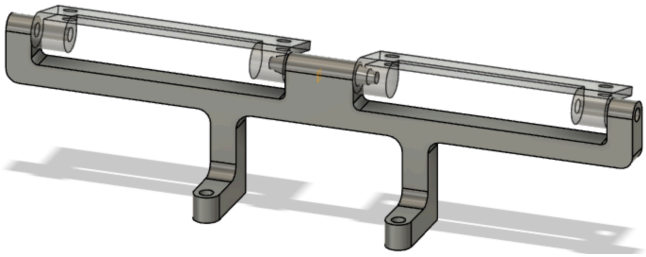
1. added extra 10mm height to **gearbox v11** and **back motor mount v2**



2. made **top camera v5** shorter
    a. keep below 25cm height limit
3. modelled and printed **deposit** and **sorting servo mount v1**
    a. screw onto rpi 4 mount and MDD3A mount



4. modelled **compartment hinge v1**

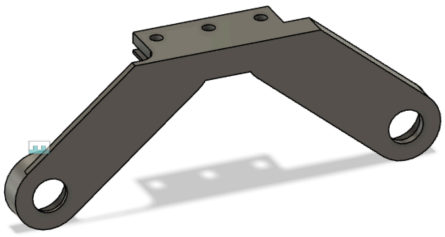


[30/03/2023]
[Celeste, Kae, Minjie]

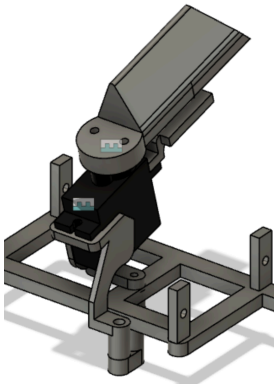
MECHANICAL

Tasks Done

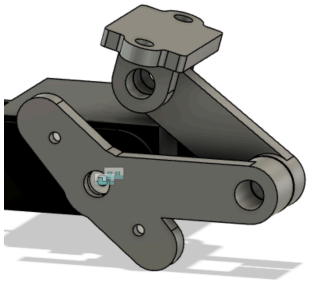
1. removed side lidar mounts from **wheel brace v5**, edited brace for the 10mm height increase



2. decreased wall thickness of **bottom camera mount 7**
3. move height of l0x down on **gripper arm v8**
4. modelled **sorting servo mount v2**, added **sorting servo bridge v1** and **sorting arm v1**



5. started modelling **deposit arm v1**, using a 2-joint system with lego axels



6. built compartment out of cardboard, 53x130x30mm

SOFTWARE

Tasks Done

1. Attempted a whole other linetrack
    a. By comparing whether the line leads left or right based on where the black ends relative to the centre; the furthest is the direction, and the robot is forced to do a fixed rotation towards that direction

	Issues	Solution
Linetrack	Realised it has been abit laggy during the past weeks	Try redoing the algorithm
Linetrack	The new linetrack did not work	Add timeout timer (still did

<table>
<tr><td></td><td>
<table>
<tr><td></td><td></td><td>not work so we reverted back to original code)</td></tr>
</table>
</td></tr>
<tr><td>[31/03/2023]
[Celeste, Aseera, Minjie]</td><td>
SOFTWARE
<u>Tasks Done</u>
1. Reverse the code to find out why the entire linetrack has been lagging for the past week
2. Remove lidar readings in code
3. ROBOT SUCCEEDS!!
4. Now able to straighten out before a line gap to cross it

<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>Lidars</td><td>Lidars were causing the entire code to lag by around 100 milliseconds, as the L0X library we were using is blocking when reading from them. (and we have 5 L0X on our robot)</td><td>Use a non-blocking library</td></tr>
</table>
</td></tr>
<tr><td>[02/04/2023]
[Celeste]</td><td>
SOFTWARE
<u>Tasks Done</u>
1. Added non blocking library for the lidars
    a. Verified all the lidars are working with no lag
    b. Added back lidars into main code
2. Updated main code to be able to control servos again after rehauling of linetrack
    a. Pick up (when the difference between the front op lidar and the front bottom lidar is bigger than 30mm)
    b. Deposit tested
3. Tuned for green squares on rounds by turning a fixed amount instead of when the green square cannot be seen anymore

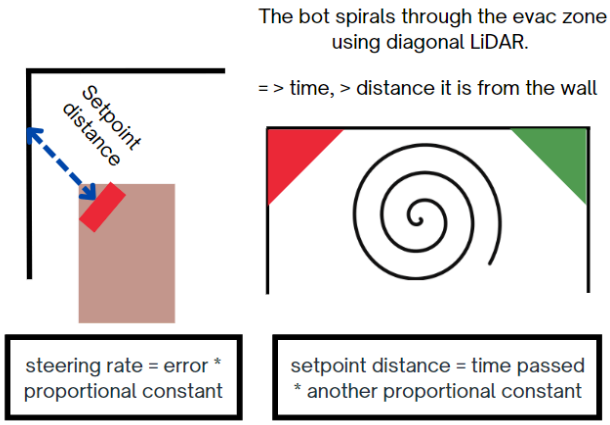
<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>Green squares</td><td>At green squares that are not 90 degrees, the robot's initial heading causes it to lose the following line</td><td>Tuned for middle ground (turns about 80 degrees)</td></tr>
</table>
</td></tr>
<tr><td>[03/04/2023]
[Celeste, Kae Aseera, Minjie]</td><td>
MECHANICAL
<u>Tasks Done</u>
1. changed top camera mount v6 angle to 17 degrees
2. added 2 extrusions to gripper arm v9
    a. prevents victims and rescue kit from touching the camera
3. extended length of deposit arms in v2
4. curved right deposit arm v2 to go around pico
</td></tr>
</table>

	Issues	Solution
deposit arms v2	• tendency to stall or break or bend if it stops or starts at the wrong angle	• consider different mechanism, pushing or sliding maybe

SOFTWARE

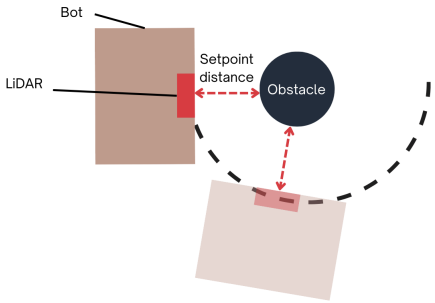
Tasks Done

1. Added **obstacle** logic using the side lidar
    a. PID around obstacle (discarded)
    b. Diff method detailed in table
2. Added rescue kit logic back into main after linetrack rehaul
3. Added **walltrack** by scaling rotation by the distance from the wall.



4. Added ball track with an evac.py script designed to give a rotation based on the ball's coordinates
5. Finished evac till the first deposit

Design Concept: PID Distance (Not using anymore)



steer = (setpoint - current distance) * proportionality constant

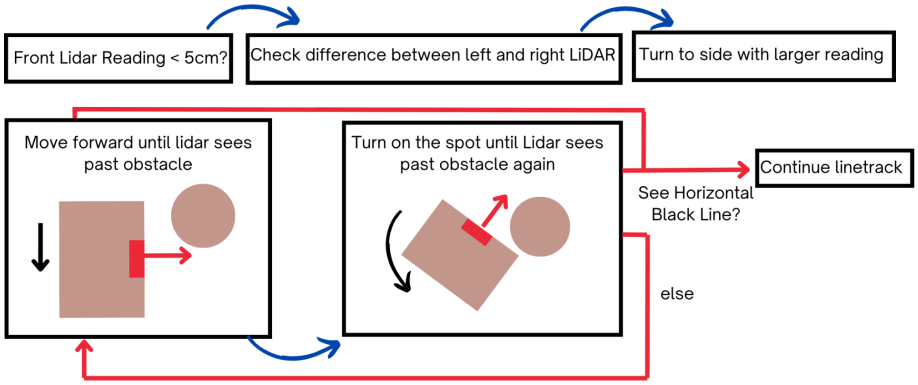
Given the obstacle is on the left of the robot, when the robot is nearer than expected, the positive difference is proportionally controlled and the robot turns right away from the

obstacle. Likewise, when it is too far, the negative difference causes the robot to turn left towards the obstacle with a negative steer.

For the obstacles on the right side, the opposite happens

steer = (current distance - setpoint) * proportionality constant

Design Concepts: New **Obstacle** logic



When the robot moves straight, it will move past the obstacle and the lidar will experience a huge increment in value once it is completely past the obstacle. In that case, the robot will start to turn, waiting for the lidar to overshoot again. Then, it moves straight. Repeat.

	Issues	Solution
Obstacle	PID around obstacle does not work because it turns too much and starts correcting the wrong way	Do a straight, turn, straight, turn method, every single time while waiting for the lidars to overshoot obstacle, until line is seen
Ball track	When tracking the ball, the gripper is half closed to funnel the ball in. However, when it is wall tracking, it is open to avoid collisions with the wall. This causes it to half close when it detects a ball, and if the ball is too near, it essentially flings the ball away	Only close if the ball is far away. If not, turn on the spot and center on ball

[04/04/2023]
[Celeste, Aseera]

SOFTWARE

Tasks Done

1. Find the first deposit point using the camera
    a. Mask the color out (red/green)
    b. Take the sum over a threshold
2. Increased sensitivity of linetrack by increasing its exponent

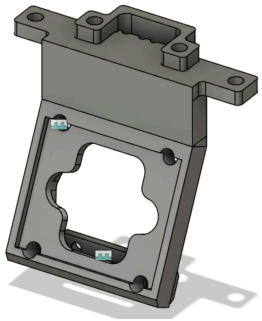
		Issues	Solutions
	Camera	Camera angled too far downwards, unable to see a substantial amount of the evac point when too far away	Angle camera upwards slightly more

[05/04/2023]
[Celeste, Kae Aseera, Minjie]

MECHANICAL

Tasks Done

1. tested different backlashes (0.2, 0.3, 0.35) to increase gear smoothness in **gears v9**
    a. settled on 0.35mm
2. changed **bottom camera mount v8** height to fit the 10mm height increase
3. increase thickness of led mount in **bottom camera mount v8**



4. increased height of extrusions on **gripper arm v10** for the 10mm height increase
5. increase height of **grippers v8** for 10mm increase
6. curved the bottom of the **grippers v8** to pass speed bumps more easily

SOFTWARE

Tasks Done

1. Added **evac** cases (when it sees out of range aka exit or a wall when its tracking a ball, it does a turn)
2. Changed **line gap** code by doing a sweep left, then right. If a close line is found, track the nearest line to its original position. If not, then it is a line gap, where it will start to move forward until it sees a line close enough to it

	Issues	Solution
Line gap	Robot sometimes trigger line gap when it is not line gap	Do a sweep to ensure there's no lines near it. This also solves 135.
Evac	When tracking noise outside of the evac zone, the robot sometimes exits or gets stuck on the wall	Constraint away these situations and prioritize the other stuff mentioned above over the ball because the most important thing is for the robot to stay in the evac zone

<table>
<tr><td></td><td></td></tr>
<tr><td>[06/04/2023]
[Celeste, Minjie]</td><td>

MECHANICAL

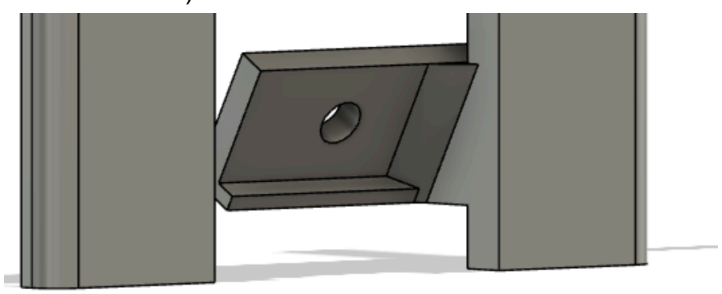
<table>
<tr><th></th><th><u>Issues</u></th><th><u>Solution</u></th></tr>
<tr><td>battery</td><td>Battery at the back causes the cg to move backward, causing it to be
unable to climb the ramp, when the rotation is too large (it goes left and right and ends up slipping down the ramp)
Tip over when the see saw falls to the other side</td><td>Mount the battery at the back
start modelling battery mount</td></tr>
<tr><td>gripper servos</td><td>gripper servos keep falling out even after hot glue</td><td>secure servo with both screw holes instead of just using one</td></tr>
</table>

SOFTWARE
<u>Tasks Done</u>
Tuned distances it moves left and right by during the line gap sweep

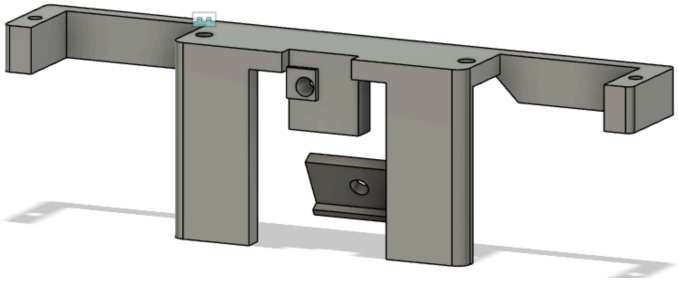
<table>
<tr><th></th><th><u>Issues</u></th><th><u>Solution</u></th></tr>
<tr><td>Encoders</td><td>Sweep is not working: we found it to be the cause of the encoder values going haywire when the battery is placed near it (currently it is placed below the bot, pressing against the encoder wires).
Even though the encoder wires have already been hot glued.</td><td>Make sure linetrack is smoother
Move the battery away (this worked, we mounted it at the back of the robot instead)</td></tr>
</table>

</td></tr>
<tr><td>[07/04/2023]
[Celeste, Kae Aseera, Minjie]</td><td>

MECHANICAL
<u>Tasks Done</u>
added mount position for TEMT6000 on gripper arm v11 for victim identification (if needed)



</td></tr>
</table>

	SOFTWARE Tasks Done 1. Get the type of **ball** a. Masking out the detected ball b. Then get the percentage of black in that mask c. Silver balls should have lesser percentages 2. Sort into correct compartment based on ball type using the sorter
[08/04/2023] [Celeste, Kae Aseera, Minjie]	**MECHANICAL** Tasks Done 1. added the screw holes on the other side of both gripper servos to keep servos in place in **gripper arm v12** 
	SOFTWARE Tasks Done 1. Stream both camera streams and combine the code for the pico and pi all the way from enter **evac** to end of evac a. Pico tells pi when its done with picking balls through a timer b. Pi switches to bottom camera and looks for first deposit point, tells pico where to go c. Pico reaches deposit point, detected by lidar, deposits, and returns to the centre of the evac zone, telling pi it has finished with the first deposit d. Pi then looks for the second deposit point and tells pico where to go e. Afterwards, pico tells pi it has finished with depositing and pi starts looking for line back to the linetrack as the pico walltracks out of the evac zone
[09/04/2023] [Celeste, Aseera, Minjie]	**SOFTWARE** Tasks Done 1. Combined linetrack and evac code together in one single file, to be automatically run by itself 2. Tuned various constants for linetrack (exponents)
[10/04/2023] [Celeste, Kae Aseera, Minjie]	**MECHANICAL** Tasks Done 1. curved right deposit arm more to fit around the pico better
	SOFTWARE Tasks Done 1. **Evac exit** cases added

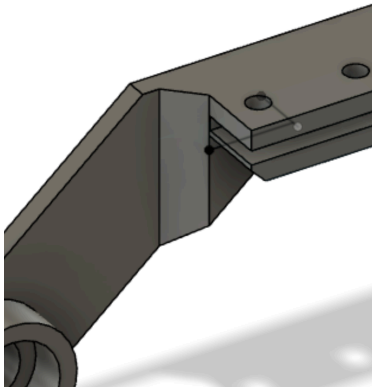
	Issues	Solution
Evac exit	In a corner case, there is no marking to tell that it is a corner	Do many hard coded cases for many permutations 1. Open corner 2. Straight forward but opposite side open 3. Etc etc To differentiate between those and open corner

[11/04/2023]
[Celeste, Kae, Aseera, Minjie]

MECHANICAL

Tasks Done

1. changed fillet to chamfer in **wheel brace v6** to increase structural stability



2. made it easier to screw in **sorting servo mount v3** by making the walls thinner
3. increased thickness of compartment hinge legs

	Issues	Solution
sorting servo mount v3	• inconvenient and difficult to screw in	• make design less bulky to make space for screwdriver
compartment hinge v2	• legs of hingle will flex, makes deposit more difficult	• increase thickness of leg design • increase wall thickness for 3d printing

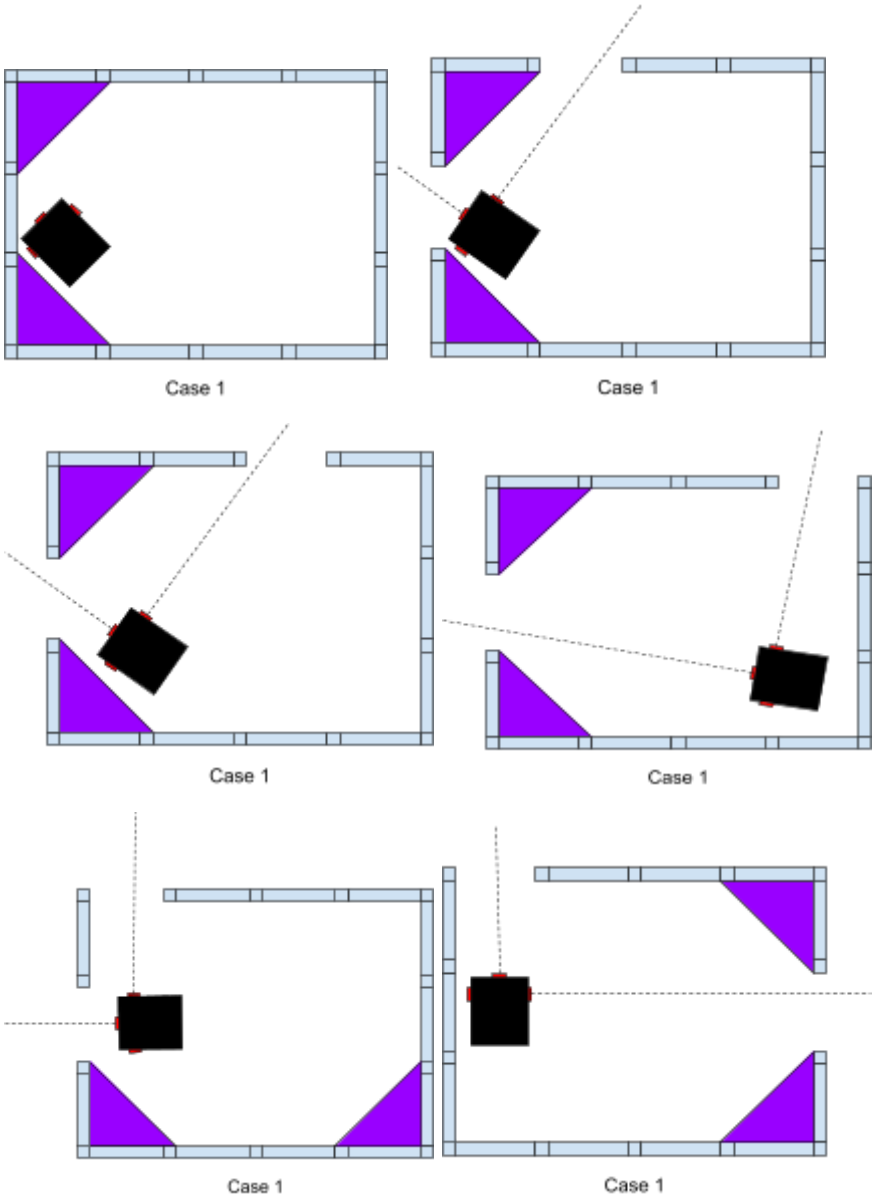
SOFTWARE

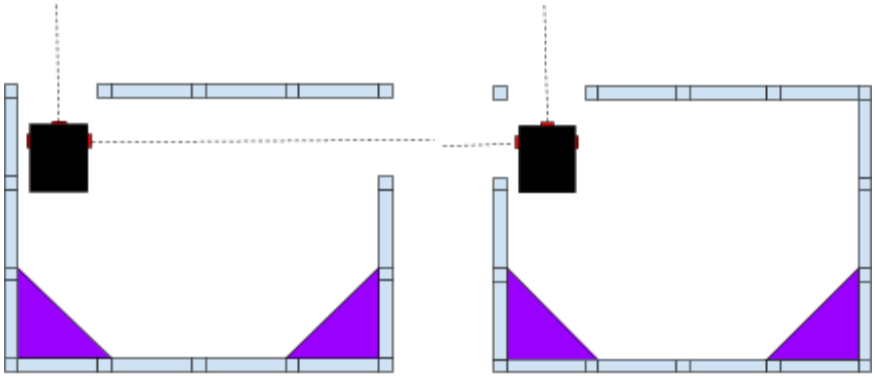
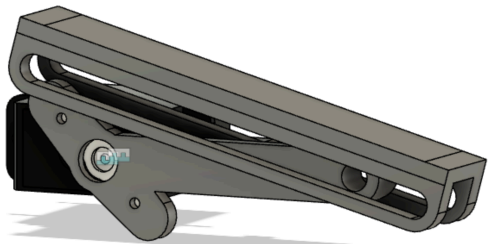
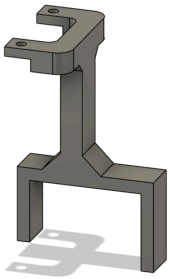
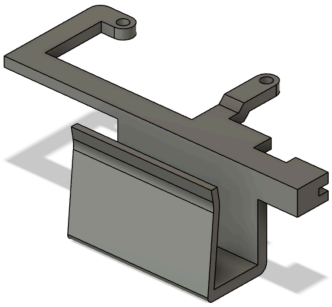
Tasks Done

1. Added even more **evac exit** cases
    a. The robot walltracks using the diagonal left lidar, and turns when it senses an open wall (when the lidar reading exceeds a certain amount)
    b. An example is below in Case 1 where all have the same lidar readings, but are all different configuration. The robot will not know whether to turn further or whether it's already facing the exit. In the examples below, the front and right lidars see outside the evac.

c. To solve this, we move straight until the left lidar also sees out of the evac zone, and then
d. Do a turn until it triggers on the extrusion beside the wall.
e. Afterwards, we turn back a little bit to correct for the tilt
f. Another case is case 2, where left and front lidar sees out

2. Added auto startup script for our raspberry pi



	
[13/04/2023] [Kae]	**MECHANICAL** Tasks Done 1. modelled and printed new **sliding compartment lifter v1**  2. modelled and printed 2 different battery holders a. secure the battery vertically (probably with masking tape) between the compartments (**battery holder v1**)  b. clip for battery at the back of the robot (**battery holder v2**)  3. epoxy glued gears to ball bearings a. only normal gears were glued

b. wheel gears not glued since they are reinforced by the wheel brace

4. marked gears for smoothest configuration

	Issues	Solution
battery holder v1	• battery very difficult and inconvenient to remove, but also not secure	• use version 2
battery holder v2	• mount holes are quite far in, print might break	• increase thickness of design

[14/04/2023]
[Celeste, Aseera, Minjie]

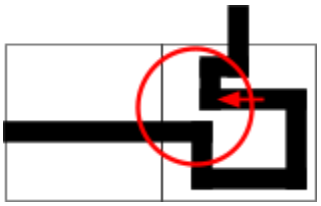
SOFTWARE

Tasks Done

1. Find closest contour and mask out everything else

```
#~ Contours for weird close national tile
if white_start_y < 140:
    # cv2.imshow("before contours", mask_black)
    contours, _ = cv2.findContours(mask_black, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    # cv2.drawContours(frame_org, contours, -1, (0, 255, 0), 3)       You, last week • remove
    # cv2.imshow("with contours", frame_org)
    # print(contours)

    cnts = []
    for cnt in contours:
        x,y,w,h = cv2.boundingRect(cnt)
        cnts.append({
            "x": x,
            "y": y,
            "w": w,
            "h": h
        })
```

	Issues	Solution
contours	For a specific tile, where two lines are very close to each other, the robot sees it as a continuous line and continues straight. 	To solve this, we used opencv contours to find the closest line segment and then mask it out so that it on

[15/04/2023]
[Celeste, Aseera, Minjie]

SOFTWARE

Tasks Done

1. Attempted to add a timer for evac.
2. However, timer didn't work and no one can find the issue

	Issues	Solution
evac	Robot does a fixed 3 minute evac, so if reseted it will not have time to complete the rest of the mission	Add a timer so if the robot has used too much time before evac, just go straight through and deposit rescue kit only

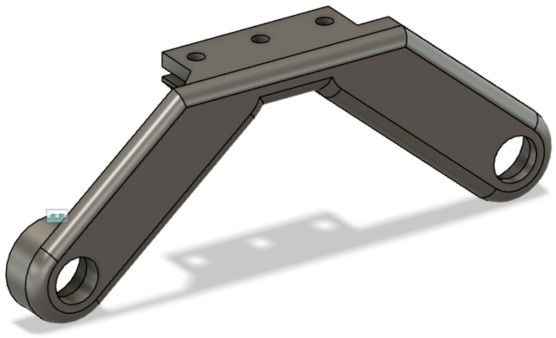
ROBOCUP NATIONALS

[17/04/2023]
[Kae]

MECHANICAL

Tasks Done

1. filleted outer edges of **wheel brace v7**, printed



	Issues	Solution
wheel brace v7	• brace gets stuck on bumps on	• fillet sharp edges of

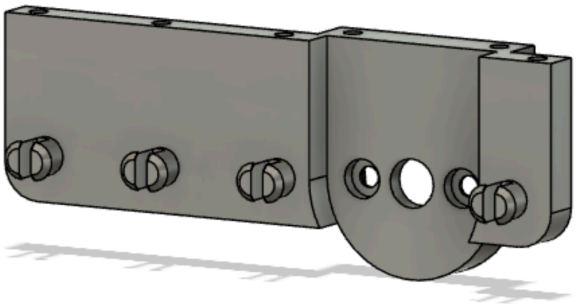
		evac zone walls	wheel brace

END ROBOCUP NATIONALS

[02/05/2023]
[Kae]

MECHANICAL
Tasks Done

1. decided on 5mm height decrease to **gearbox v12**
2. modelled **gearbox v12** for PCB v3
    a. both gearboxes to be driven from the 2nd/4th gear



3. modified **back motor mount v3** for the 5mm height decrease

[04/05/2023]
[Kae]

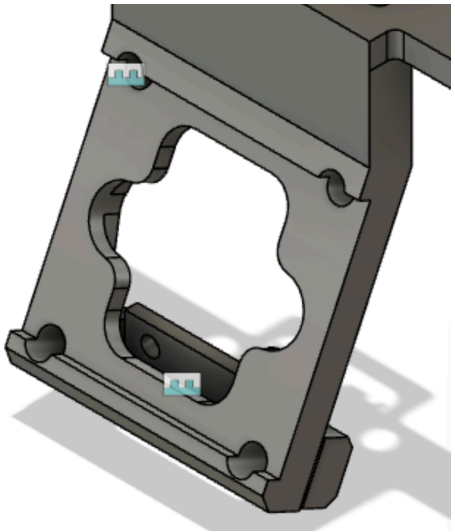
MECHANICAL
Tasks Done

1. added holes to the bottom surface of the gearbox to mount **Bottom PCB v2**

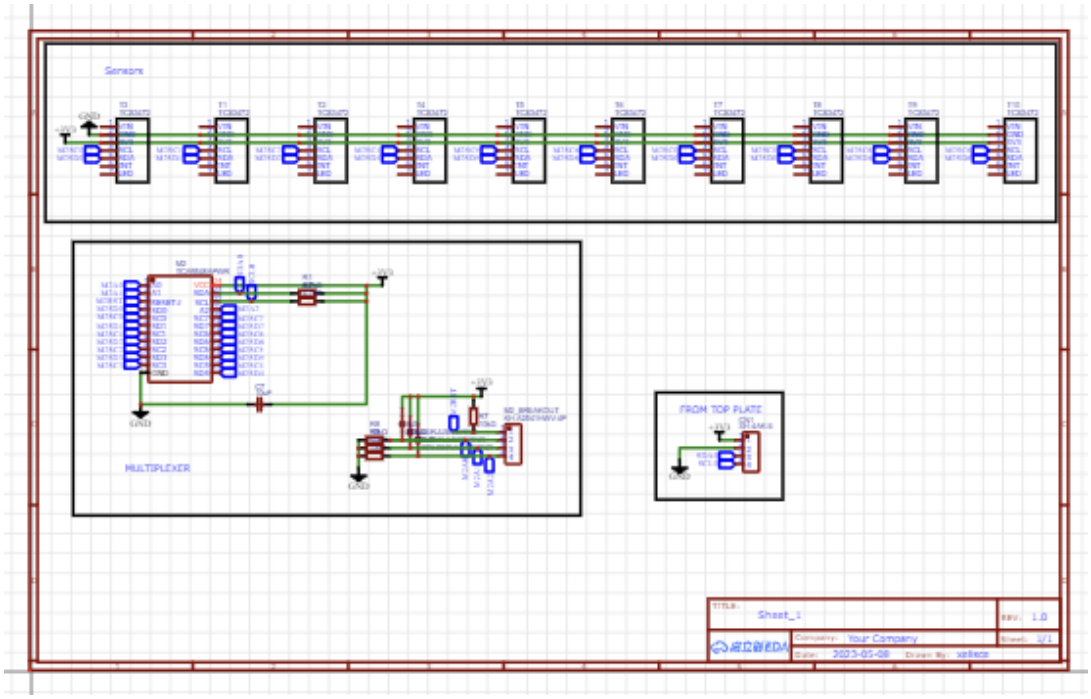
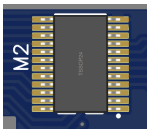
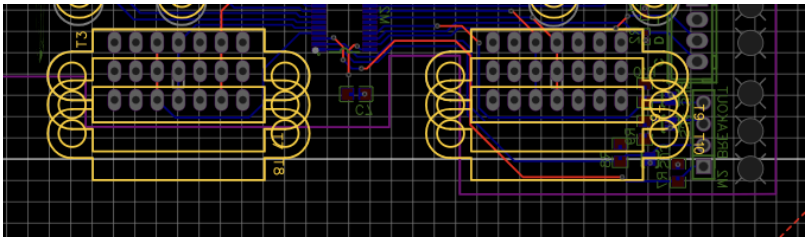
[05/05/2023]
[Kae]

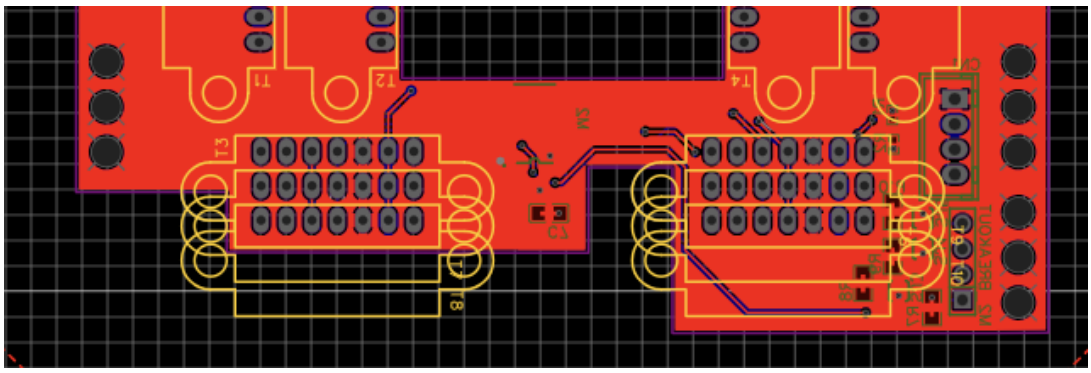
MECHANICAL
Tasks Done

1. removed unnecessary walls on **bottom camera mount v9**



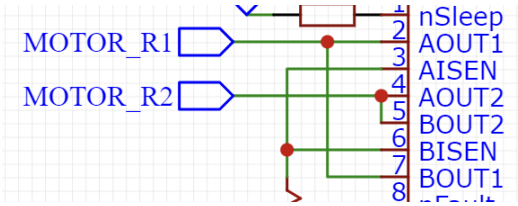
PCB V3

[09/05/2023] [Celeste, Kae, Aseera, Minjie]	**ELECTRICAL** Tasks Done 1. Started schematics for **Bottom PCB V2** a. Changes: i. Use tcs instead of temt  2. Modelled first draft of **Bottom PCB V2** a. Consists of a surface mounted (SMT) multiplexer (tca9458A) connected to the TCS sensors  b. Included variable positions for the TCS sensors, which are connected to the same trace  i. Allows us to move the TCS sensors up and down for the green squares detection, when the front sensors are detecting an intersection c. Included variable positions for the entire plate by widening the mounting hole on the pcb



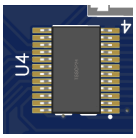
i. The entire plate is connected to the gearbox, so adding more holes would allow the plate to shift forward and backward.
ii. In the z-axis, it can also move by changing the thickness of the spacer connecting the bottom plate pcb to the gearbox

3. Started schematics for the top plate (**PCB V3**) for internationals
   a. SMT the motor driver DRV8833. By connecting the motor outputs in parallel, we can allow a 3A continuous current, 4A peak current load on the motor driver



- Outputs can be in Parallel for
  - 3-A RMS, 4-A Peak (PWP and RTY)
  - 1-A RMS, 4-A Peak (PW)

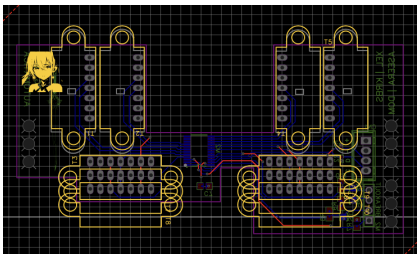
   b. SMT the multiplexer

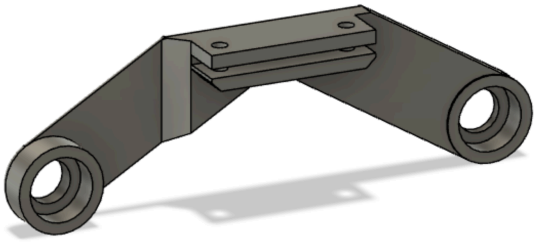
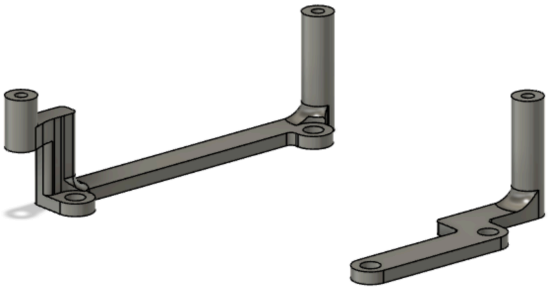
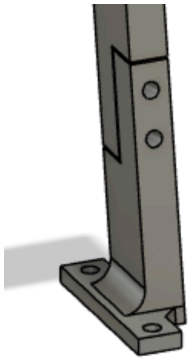


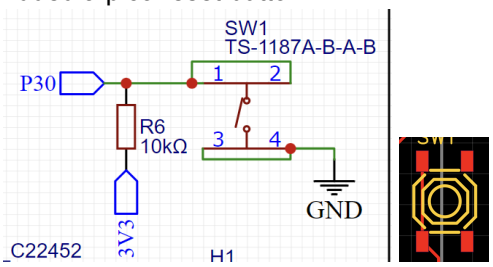
i. The multiplexer is connected to the lidars and the IMUs
ii. The IMU is connected to the same traces as the VLs because they have different addresses

   c. Connected the switch to 3.3v and GND
   d. Changed from the large adjustable 6v reg that spans the entire width of the robot (connected via connectors) to a small 6v reg that can be soldered directly onto the board
   e. Added a 3.3V reg
   f. Added an i2c oled breakout to debug code on a screen on the robot

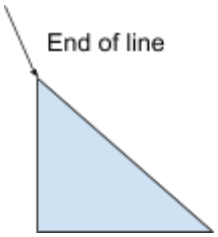
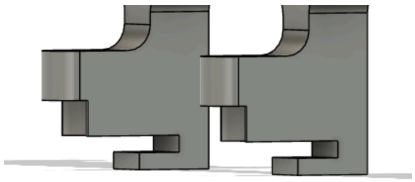
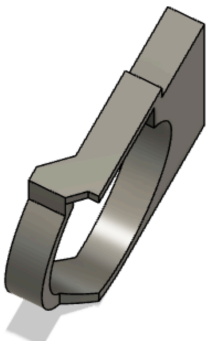
	Issues	Solution

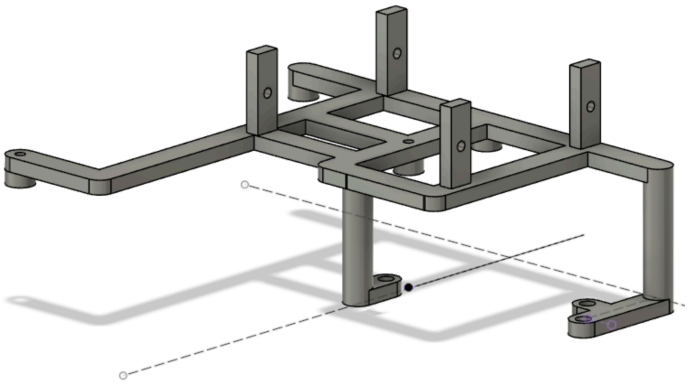
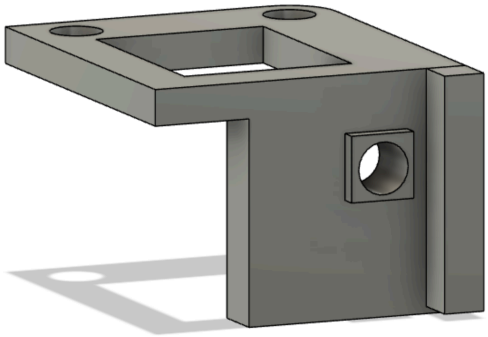
	switch	readings from switch are inconsistent	• Actually connect the switch to 3.3v and GND, instead of leaving it to be freely floating.
	motor driver **multiplexer**	takes up unnecessary space on the robot	• surface mount the chips • to save space and reduce robot size
	tcs sensors	The tcs34725 in the club:  were too big, only allowing us to fit one on each side of the camera, The temts had no led and could only read reflected light, not color (could not use it for green squares). Since the space under the robot is extremely dark, the leds were needed to give an accurate representation of the line.	Purchased a different model of TCS34725: TCS3472  This component has LEDs on each side of the sensor, and is smaller, allowing us to fit 6 sensors on the plate instead: 
	3.3v reg	The pico sometimes fries after prolonged use because of the logic level in its i2c communication The mux and lidars are at 5v logic level, but the pico is at a 3.3v logic level	Since the logic level of the lidars and the mux follows what they are powered with, using a 3.3v to power them all will solve the logic level problem. Since the pico's on board 3.3v reg cannot support the current draw of all these, a 3.3v reg will be added
	MECHANICAL		

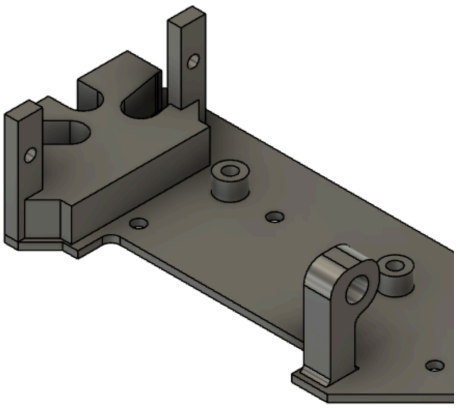
	Tasks Done 1. modelled and printed **wheel brace v8** for **PCB v3** 
[15/05/2023] [Kae]	**MECHANICAL** Tasks Done 1. created **battery holder v3** for **PCB v3** 2. modelled **gripper arm v13** for **PCB v3** a. brought the 2 prongs closer together 3. modelled **servo mount v7** for **PCB v3** a. added extrusions for compartment hinge screws to thread into 4. modelled **rpi 4 mount v4** for **PCB v3** a. decided to split into 2 parts since one corner is supported by header pins into PCB 
[16/05/2023] [Kae, Celeste]	**MECHANICAL** Tasks Done 1. increase distance between prongs of **handle v2** a. aka made handle wider and redid handle base b. for more compartment space 
	ELECTRICAL

	Tasks Done 1. Did a preliminary check after receiving PCB V3 and found that the motor driver trace was too small → 10 mil because someone forgot about it
PCB V4	
[17/05/2023] [Celeste, Kae, Aseera, Minjie]	**ELECTRICAL** Tasks Done 1. PCB V4 schematics a. Added a pico reset button  b. Changed the switch footprint to a jst xh 2 pin to wire it instead of directly solder c. Added a tilt sensor in preparation to include a tilt sensor d. Increased the motor driver to motor trace widths (the main problem) e. Added pullup resistors for the i2c lines to the mux to stabilize the bus (it keeps crashing) f. Removed oled breakout because we will not use it

	Issues	Solution
switch	Switch was under tension as the structural support was its pins, would break easily	Solder wires to the switch and heatshrink around the joint, and then use a jst to connect to the pcb instead of directly solder
pico	Sometimes the pico requires a reset because - the i2c bus has crashed and needs reinitialisation - The code crashed and now it needs to be mounted as a uf2 board before upload	Add a pico reset button that connects to the RUN pin, which stops the pico from running on the hardware level
tilt	If the robot is tilting upwards at an angle, the line ends early due to the camera's angle	Add a tilt sensor

<table>
<tr><td></td><td>
<table>
<tr><td></td><td> 
</td><td></td></tr>
<tr><td>Pcb v4</td><td>Motor traces too small</td><td>Increase trace width</td></tr>
</table>
</td></tr>
<tr><td>[19/05/2023]
[Kae]</td><td>MECHANICAL
<u>Tasks Done</u>
1. created compartment hinge v3 for PCB v3
a. added extrusions to wrap around the servo mount
b. adds space to fit around extrusions on the servo mount
  </td></tr>
<tr><td>[23/05/2023]
[Celeste, Aseera, Minjie]</td><td>SOFTWARE
<u>Tasks Done</u>
1. Tested the TCS sensors: Managed to obtain readings from the sensors</td></tr>
<tr><td>[25/05/2023]
[Kae]</td><td>MECHANICAL
<u>Tasks Done</u>
1. offset the bottom section of the back motor mount v4 to follow the indent in Bottom PCB v2
 
2. modelled deposit servo mount v2 for PCB v3</td></tr>
</table>

	
[27/05/2023] [Kae, Minjie]	**MECHANICAL** Tasks Done 1. modelled **left side lidar mount v2** 
[28/05/2023] [Celeste, Kae, Aseera, Minjie]	**ELECTRICAL** Tasks Done 1. Soldered headers and regulators onto first plate of PCB V4
[29/05/2023] [Celeste, Kae,, Minjie]	**MECHANICAL** Tasks Done 1. Assembled robot 2. increased mount height of gripper arm servo in **servo mount v8** a. to preserve the gripper-servo-compartment height ratio from nationals 3. added extrusions for compartmnet hinge mounting screws to thread into in **servo mount v8**



SOFTWARE

Tasks Done

1. Switch TCS library

	Issues	Solution
TCS library	Used adafruit library at first, which was blocking, which caused a 60ms reading for each lidar, adding a 8x60ms reading for the TCS	1. Amend the adafruit one,to read when available 2. Find a non blocking TCS library
	The non blocking tcs library (https://www.arduinolibraries.info/libraries/tcs34725-auto-gain) has issues with initialisation. In the event that the robot is just powered up, the TCS sensors are not initialised correctly and this library only works after the adafruit library has run once	Modified the Adafruit library to use the settings: (below) The integration time and gainer results in the least loop time for the TCS readings

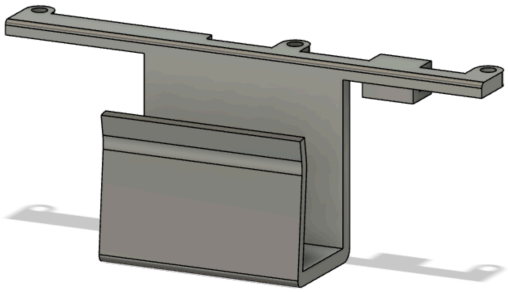
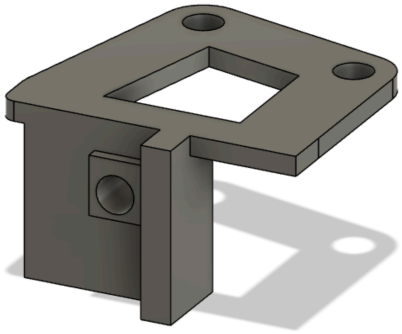
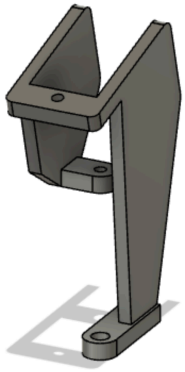
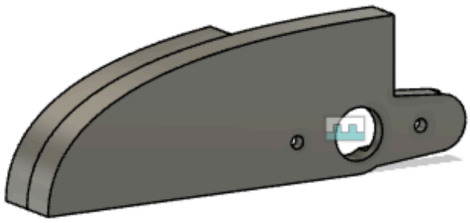
```
Adafruit_TCS34725 tcs[6] = {Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_154MS, TCS34725_GAIN_1X),
                            Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_154MS, TCS34725_GAIN_1X),
                            Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_154MS, TCS34725_GAIN_1X),
                            Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_154MS, TCS34725_GAIN_1X),
                            Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_154MS, TCS34725_GAIN_1X),
                            Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_154MS, TCS34725_GAIN_1X)};
```

ELECTRICAL

Tasks Done

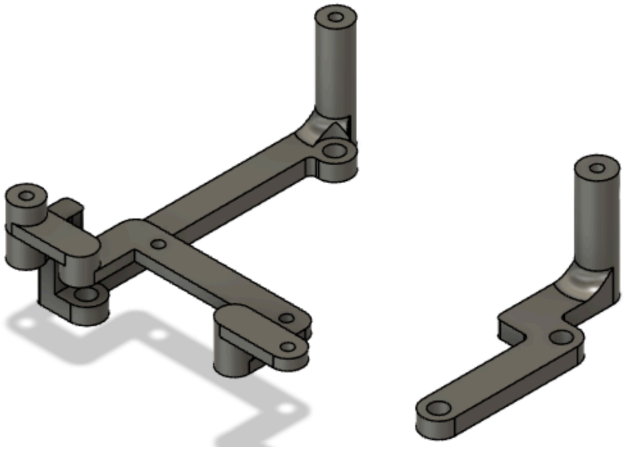
1. Soldered tcs34725 (colour sensors) onto bottom plate

	Issues	Solution
pico	Pico fried The battery wire had been scratched and exposed, delivering 12v on the 6v line	Tape up exposed wire

[31/05/2023] [Celeste, Kae, Minjie]	**MECHANICAL** Tasks Done 1. expanded width of **battery holder v3** from 15mm to 18mm  2. increased height of **grippers v9** to grab below midpoint of ball 3. modelled **right side lidar mount v2**  4. modelled **sorting servo mount v4** for **PCB v3** a. screwed onto deposit servo mount  5. modelled new **deposit arm v3** a. pushes compartment up, is not directly attached 

<table>
<tr>
<td></td>
<td>
<table>
<tr><th></th><th>Issues</th><th>Solution</th></tr>
<tr><td>battery holder v3</td><td>• LiPo battery bloated from use
• battery holder snaps due to being under too much strain</td><td>• expand width of battery holder</td></tr>
<tr><td>grippers v9</td><td>• grippers are too high up to bring balls firmly</td><td>• lower gripper height</td></tr>
</table>

SOFTWARE
Tasks Done
1. Rgb to hsv function for the tcs sensors
2. Started unit tests
<table>
<tr><th>Unit test</th><th>Description</th></tr>
<tr><td>Linetrack</td><td>To test: TCS sensor readings integrated with robot’s movement
Result:
• Still has latency, but TCS reads accurate values
• Linetrack oscillates further from line instead of towards line</td></tr>
<tr><td>Pickup</td><td>To test: gripper height and strength
Result:
• With the debris (toothpicks), the lidar readings are different as more of the sensor returns on the toothpick, and the gripper grips the ball at different points</td></tr>
<tr><td>Walltrack</td><td>To test: PID movement based on lidar readings
Result: No change, SUCCESS!</td></tr>
<tr><td>Deposit</td><td>To test: Able to deposit heavy objects? (x2 80g rescue balls and 50g rescue kit)
Result: Fail → Switch to cam design</td></tr>
</table>
</td>
</tr>
<tr>
<td>[01/06/2023]
[Kae, Celeste]</td>
<td>
MECHANICAL
Tasks Done
1. modelled rpi 4 fan mount v1
2. modelled rpi 4 mount v5 to fit the fan mount
</td>
</tr>
</table>



3. modelled spacers to screw in bottom PCB

	Issues	Solution
rpi 4 fan mount v1	Fan was taped on, and kept getting stuck on the heatsink or nearby pins. Unable to spin, the fan could not cool the pi, resulting in pi overheating	Make **rpi 4 fan mount v1**

SOFTWARE

Tasks Done

1. Finished **unit tests**

Unit test	Description
Obstacle	To test: PID movement based on lidar readings Result: No change, SUCCESS!
Test drive	To test: Oscillations from line during linetrack Result: ??? Still oscillates no matter what

[02/06/2023]
[Kae]

MECHANICAL

Tasks Done

1. built compartment out of cardboard

ELECTRICAL

Tasks done

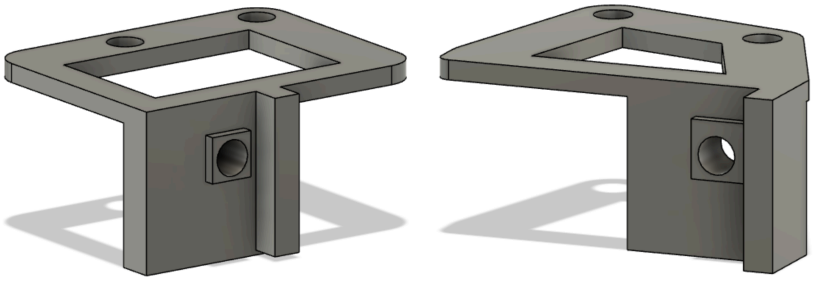
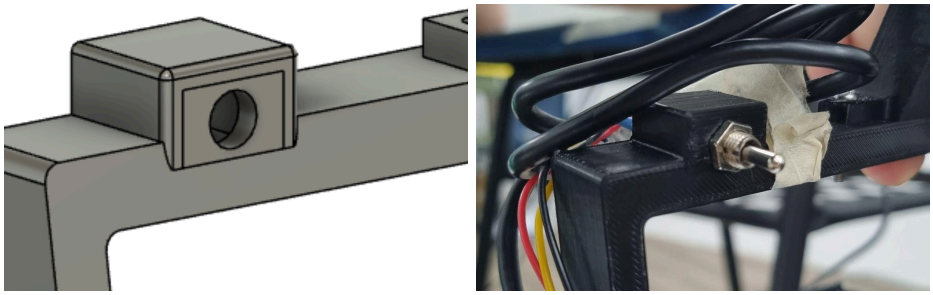
1. Soldered headers onto the second spare plate of PCB V4

[03/06/2023]
[Kae]

MECHANICAL

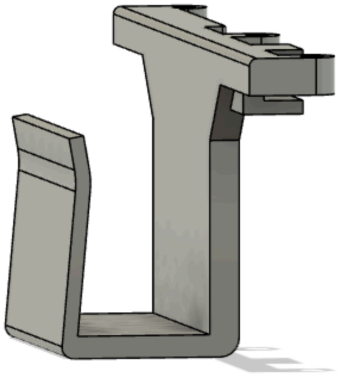
Tasks Done

1. removed the wrap-around extrusions on **compartment hinge v4**

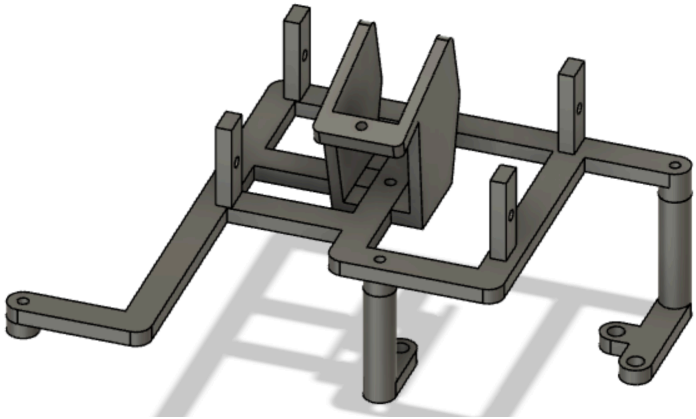
<table>
<tr>
<td></td>
<td>
2. modelled left and right side lidar mounts v3

<table>
<tr><td></td><td><u>Issues</u></td><td><u>Solution</u></td></tr>
<tr><td>compartment hinge v4</td><td>• too tight to wrap around servo mount</td><td>• remove the wrapping extrusions</td></tr>
<tr><td>side lidar mounts v3</td><td>• behind the pivot point of the robot</td><td>• shift forward to be in front of the pivot point</td></tr>
</table>
</td>
</tr>
<tr>
<td>[05/06/2023]
[Kae]</td>
<td>
MECHANICAL
<u>Tasks Done</u>
1. added switch mount to handle v3

2. increased thickness of deposit arm v4
a. more surface area of contact, less pressure on compartment
</td>
</tr>
<tr>
<td>[06/06/2023]
[Celeste]</td>
<td>
SOFTWARE
<u>Tasks Done</u>
1. Tested TCS linetrack with different error multipliers, tried tuning

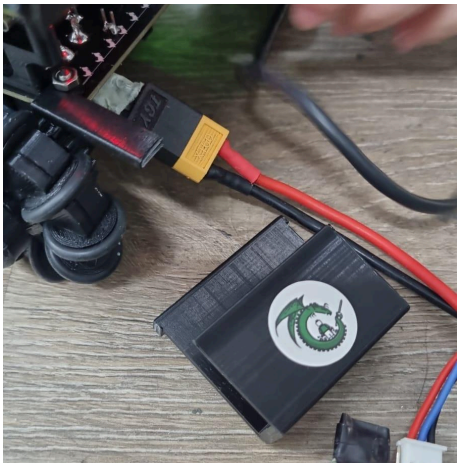
<u>Design concept</u>
Linetrack with PID
Steer = (left - right) * p
The line track sensors are calibrated to ensure that both sensors return the same value for the same colour even at different venues. Then, the cross track error, which is the difference in the left and right readings, is proportionally controlled to give a steer rate. At more drastic cross track errors, the robot will turn more forcefully towards the line, correcting more dramatically.
Further left from line, left on white, right on black, (left - right) larger. Steer is larger, robot turns right.
</td>
</tr>
</table>

<table>
<tr><td>[08/06/2023]
[Celeste]</td><td>SOFTWARE
<u>Tasks Done</u>
1. Rewrote skeleton code with enums for readability on the pico side</td></tr>
<tr><td>[09/06/2023]
[Celeste]</td><td>SOFTWARE
<u>Tasks Done</u>
1. Added forced movement into code (turn and moving distance)
2. Tuned forced movement and distance moved</td></tr>
<tr><td>[12/06/2023]
[Celeste]</td><td>SOFTWARE
<u>Tasks Done</u>
1. Added color hsv thresholds for the tcs sensors (green hues, red hues, etc.)
2. Made a struct for tcs sensors for easy access
<pre>struct tcsSensor
{
    uint16_t r;
    uint16_t g;
    uint16_t b;
    uint16_t c;
    int hue;
    int sat;
    int val;
    bool green = false;
    bool red = false;
    bool black = false;
    bool silver = false;
};
struct tcsSensor tcsSensors[tcsNum];</pre></td></tr>
<tr><td>[14/06/2023]
[Celeste]</td><td>SOFTWARE
<u>Tasks Done</u>
1. Could not make tcs linetrack work, instead used the top camera to line track with the same algorithm as bottom camera
<table><tr><td>Pros</td><td>Cons</td></tr><tr><td>• Tile gap does not seem significant due to the constant difference
• Speed bump will appear as a normal line instead of sometimes seeming like a line gap like on the bottom camera</td><td>• Less immediate information
• Weaker lighting</td></tr></table></td></tr>
<tr><td>[15/06/2023]
[Celeste]</td><td>SOFTWARE
<u>Tasks Done</u>
1. Tested top camera line track, tuned multipliers</td></tr>
<tr><td>[16/06/2023]
[Celeste, Kae]</td><td>MECHANICAL
<u>Tasks Done</u>
1. increased thickness of walls of battery holder v4 to prevent breaking</td></tr>
</table>



2. combined deposit and sorting servo mount into **deposit servo mount v3**
3. detach legs from main body of **deposit servo mount v3**



	Issues	Solution
battery holder v4	• battery holder snapped due to repeated use 	• increase thickness of design
deposit servo mount v3	• sorting servo mount is extremely difficult to mount	• combine it with the deposit servo

		and secure	mount so it is easier to mount
		• the mounting screws are far into the robot	• detach legs from main body • mount legs first, then mount main body to the legs

SOFTWARE

Tasks Done

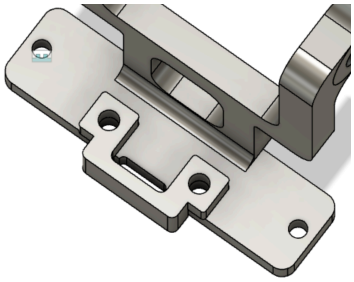
1. Added line gap code, to align to the line gap
    a. Issue: robot did not immediately react when the pi saw the aligned line → seemed to be latency between communication, but later found out to be the motor PID issue

[17/06/2023]
[Kae]

MECHANICAL

Tasks Done

1. added LED mount to front of **gripper arm v14**
2. removed TEMT6000 mount from **gripper arm v14**



	Issues	Solution
gripper arm v14	• light from camera led is blocked by the gripper arm, cannot be used by top camera	• shift led position to gripper arm

[19/06/2023]
[Celeste, Kae]

MECHANICAL

Tasks Done

1. added 1mm offset to left brace and 0.5mm offset to right brace (**wheel brace v9**)

	Issues	Solution
	• brace measurements tilts omniwheels slightly, may be causing jam	• offset the brace to the correct measurements

SOFTWARE

Tasks Done

1. Tested and realised the PID was outputting the same value for hundreds of loops (results shown below)

	Issues	Solution
PID	PID may not have been properly tuned, which may also be a contributing factor in the jam	Use pid auto tuning
	PID lacked direction (the modulus of encoder values were implemented without sign)	Add direction to the encoders, by adding a variable encDir, which logs as 1 or -1 depending on whether encoderA or encoderB triggers first

Appendix:

3D Prints

Name (Ctrl+F to find)	Latest Version
gears	v9
gearbox	v12
back motor mount	v4
omniwheel	v1
bottom camera mount	v9
grippers	v9
gripper arm	v14
servo mount	v8
top camera mount	v6
pololu wheel brace adaptor	v1
omniwheel brace adaptor	v1
wheel brace	v9

rpi 4 mount	v5
rpi 4 fan mount	v1
MDD3A mount	v1
left side lidar mount	v2
right side lidar mount	v2
handle	v3
deposit servo mount	v3
compartment hinge	v4
sorting servo bridge	v1
sorting arm	v1
battery holder	v4
deposit arm	v4
SCRAPPED	
sliding compartment lifter	v1
sorting servo mount	v4
omniwheel brace	v4
motor mounts	v2
diagonal lidar mounts	v3
motor driver mount	v3
rpi 3 mount	v2

Clarification of components used

What we reference to them as	What we used
TCS	TCS34725 (RGB sensor)
TCA/Multiplexer/Mux	TCA9548A (I2C Multiplexer)
TEMT	TEMT6000 (Ambient light sensor)
Pico	Raspberry Pi Pico
Pi	Raspberry Pi 4B

LiDAR	VL53L0X & VL53L1X
L0X	VL53L1X (ToF sensor)
L1X	VL53L1X (ToF sensor)