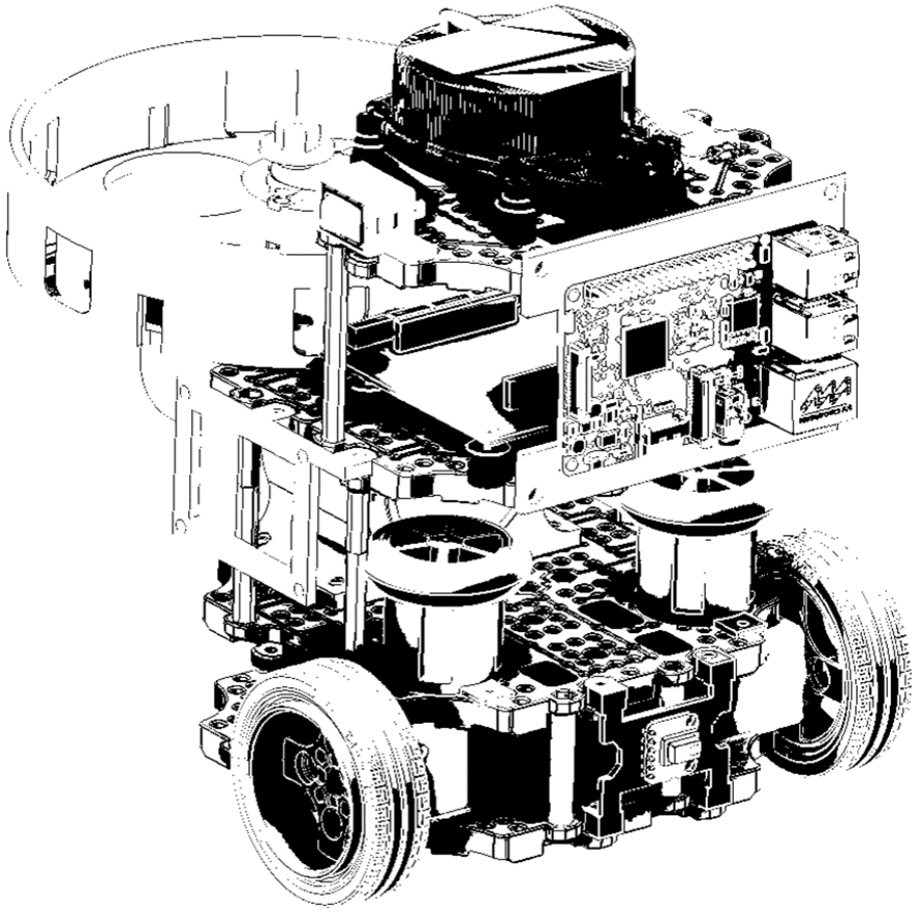
EG2310: Fundamentals of System Design

G2 DOCUMENTATION



GROUP 9

Tran Phuoc Huy Khang	A0237161L
Nguyen Minh Nguyen	A0200786E
Chew Yong Zhang	A0239856L
Sean Low Wei Jian	A0235153M
Niwa Allie	A0240270W

TABLE OF CONTENTS

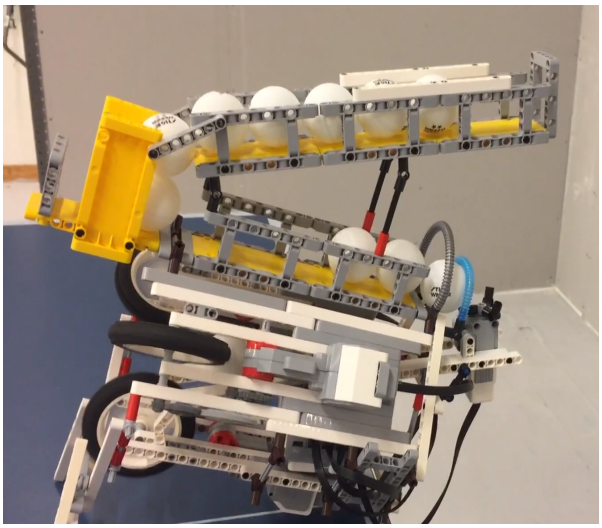
1. DESIGN PROCESS

The final robot is the integration result of multiple mechanical, electrical, and software systems, each of which has been developed through an iterative prototyping process that aims to balance all stakeholder's expectations. The prototyping process, as well as the iterative improvements, are described below.

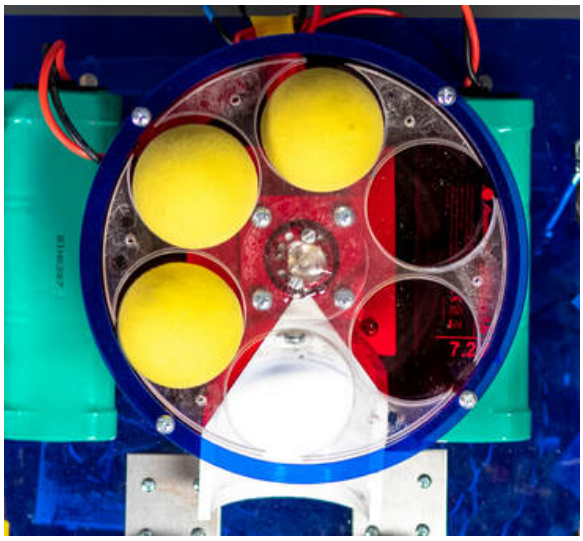
1.1 Payload

Type of Ball Feeder

To begin with, with the goal of carrying a payload of 5 ping pong balls, different designs were considered.



A slanted or vertical slide that takes up space vertically. Balls in the slide require a servo motor for feeding.



An indexing feeder requires large horizontal space. Balls in an indexing feeder can be fed using a stepper motor.

We found out that the dimension of 5 ping pong balls placed side by side in a slanted or vertical slide was quite large (about 20cm length).

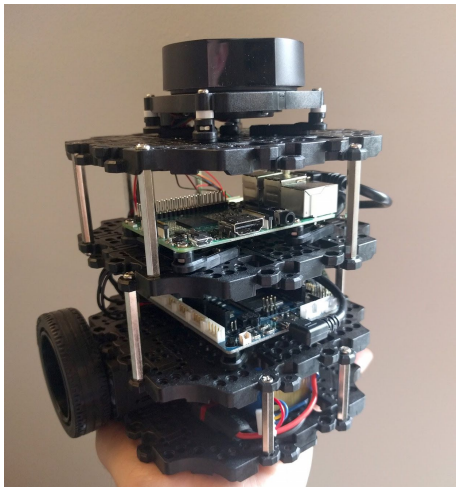
It was our intention to keep our robot's height as close to the original Turtlebot as possible in order to achieve stability, compactness as well as accuracy in the detection of the maze walls.

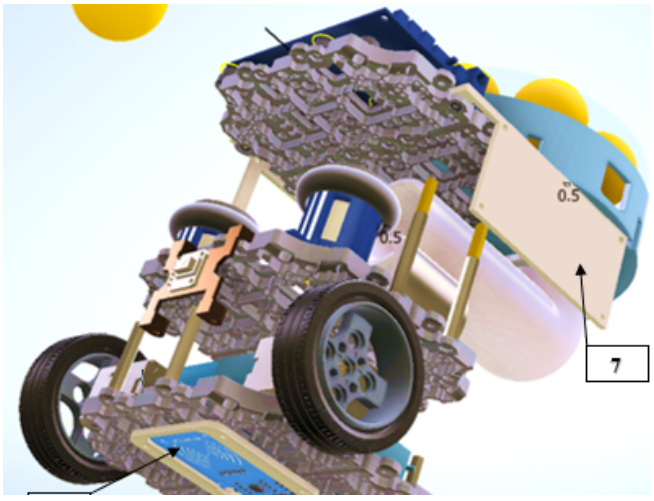
Hence, we decided to use an **indexing feeder** so as to keep the balls in a circular arrangement to save space.

Placement of Ball Feeder

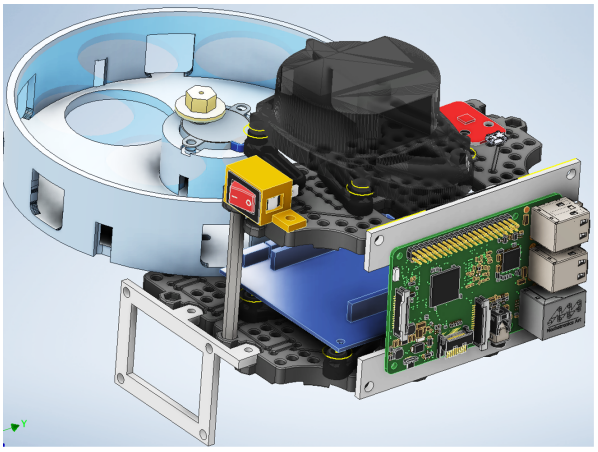
The placement of the indexing feeder also presented a challenge: the feeder should be placed relatively high on the robot to rely on gravity to feed the ping pong ball into the shooting barrel **(Right picture)**.

However, there was no space to mount it on the original Turtlebot. Looking at the original Turtlebot, the third waffle layer was used quite inefficiently with only a Raspberry Pi and the USB2LDS module **(Left picture)**.





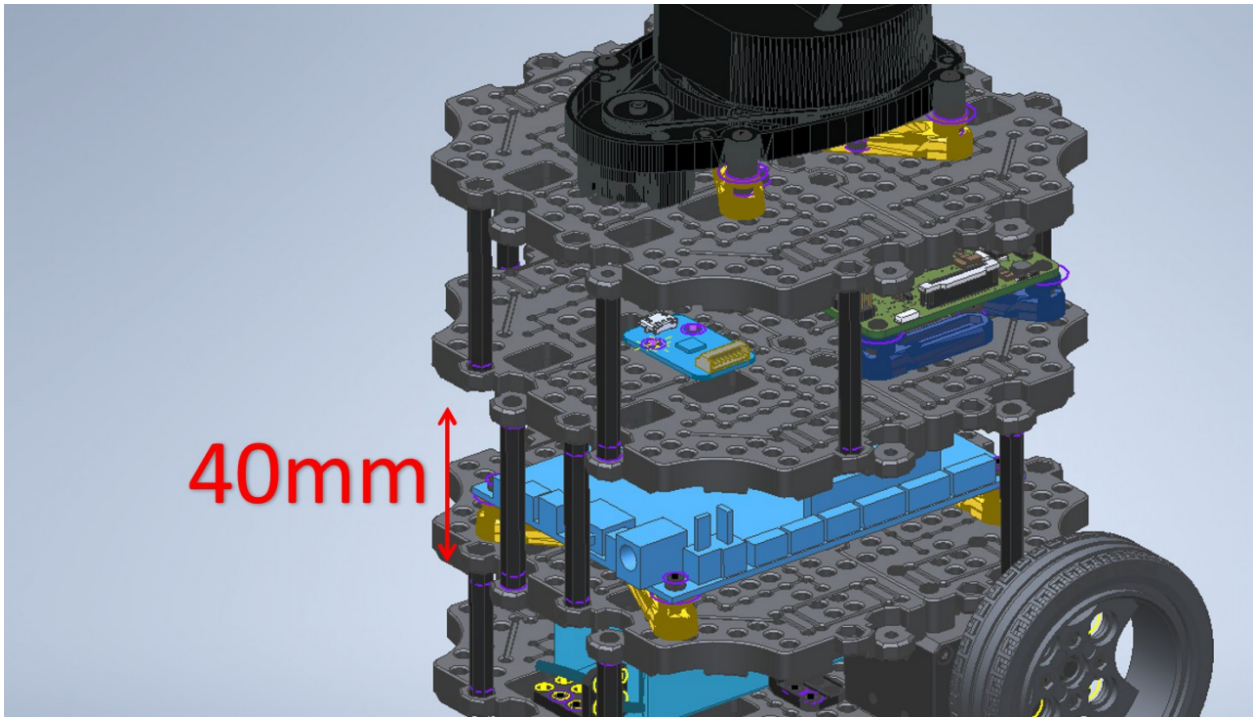
The Raspberry Pi, then, was mounted forward, while the USB2LD module was mounted next to the LiDAR.



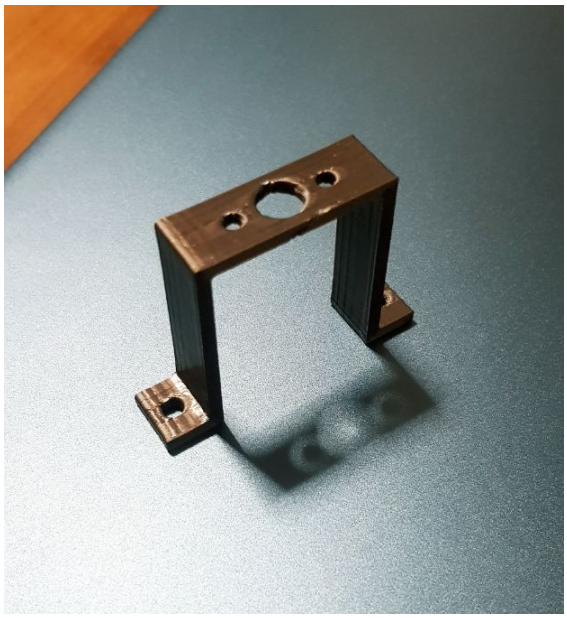
1.2 Ball launcher

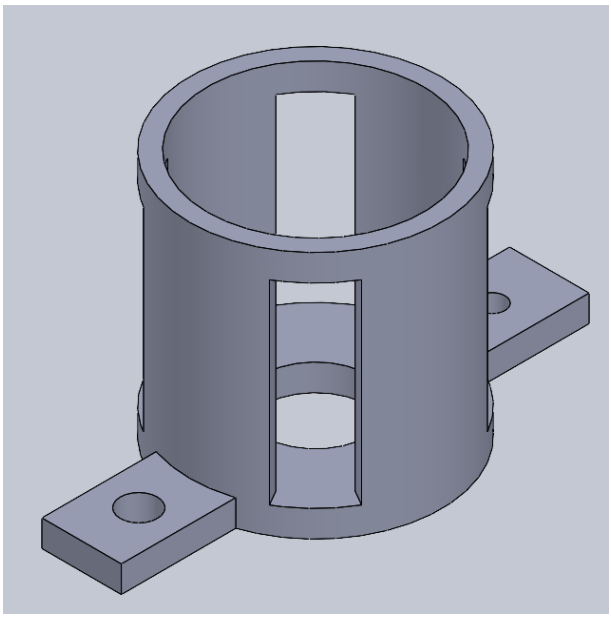
Only 2 firing mechanisms were considered for shooting the ping pong balls: Solenoid and Flywheel. We had a chance to create a prototype and conduct a shooting test based on the solenoid mechanism in one of our homework, and the result was a low accuracy, short-range shot; the force of ejecting the ball was also uncontrollable. Hence, we chose the flywheel concept for our final robot as it did not have the same problems: the range and ejecting force can be controlled by varying the DC motor's speed, and at a higher speed, the shooting accuracy would also be higher and more controllable. The flywheel mechanism's specifications can be found in Sections 2.2 and 4.3.

Regarding the placement of DC motors and barrel, with the indexing feeder mounted on layer 3 and the motors and battery mounted on layer 1, the barrel and DC motors must be mounted on layer 2, where the OpenCR was. The clearance between the 2nd and 3rd layers was smaller than the ping pong ball and the barrel, so we extended it with standoff screws. Additionally, we also moved the spacers and standoff screws to the sides so as not to block the shooting path, and the OpenCR was moved to the 3rd waffle layer.



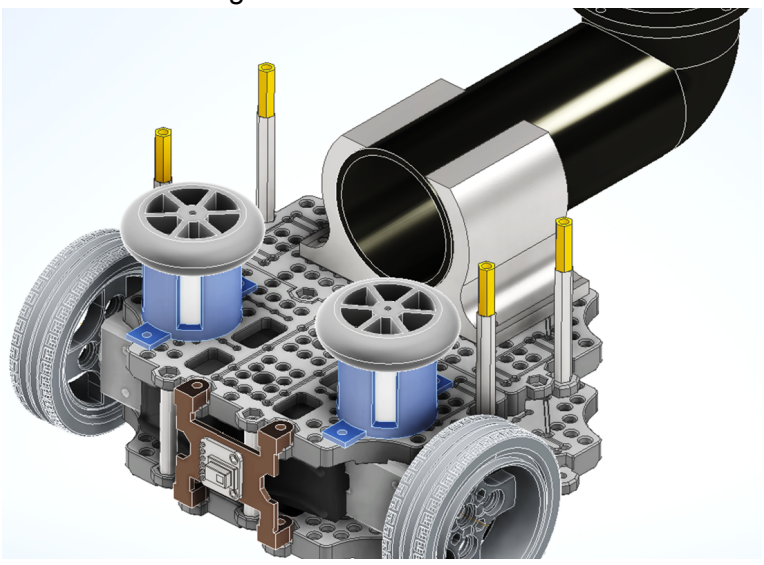
The design for the DC motor brackets was also changed during the prototyping process. Originally, we placed an emphasis on fast printing time, and we weren't sure about the barrel design and dimensions, hence we designed it like the picture on the left to be able to control the motor's height relative to the barrel. This design, however, was prone to failure as there was no proper mounting point for the DC motor. This made the motor wobble greatly until the point where the bracket broke. We, therefore, change the bracket design to the one in the right picture. With this design, the motor sits inside the bracket tightly and therefore won't break the bracket.



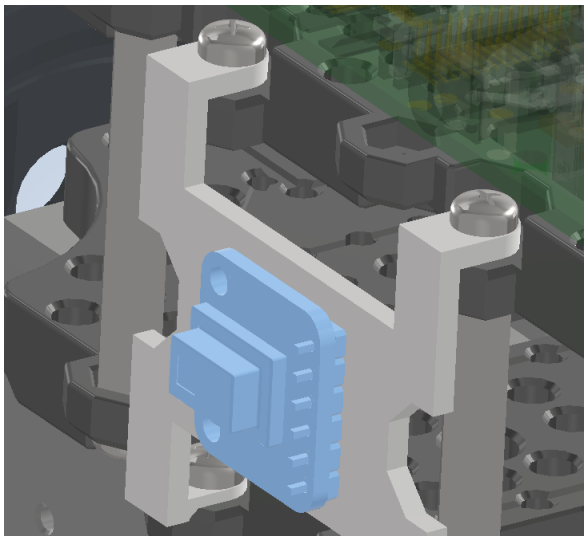


1.3 Infrared (IR) Sensor and PCBs

We decided that the thermal sensor would be mounted in front of the robot at a low position (between the 1st and 2nd layers). This is such that we do not obstruct the shooting barrel and also allow for detection of the hot target.

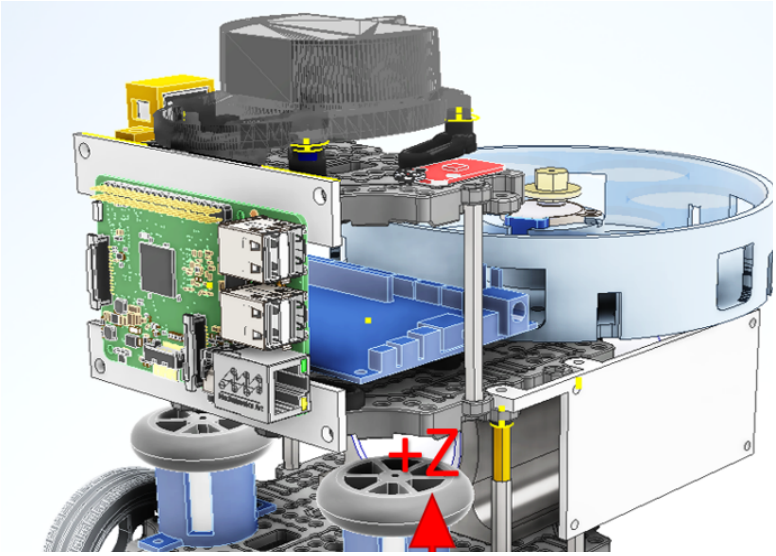


Additionally, it is positioned close to the middle, aligning relatively close with the shooting barrel to simplify the calibration process of the firing mechanism. To do this, we designed and 3D printed a special mount for the IR sensor that can be attached to the spacer mounting position.



This is a close-up view of the mount and the IR sensor together. The mount could also be easily accessed, maintained, and replaced if necessary.

With the same idea, mounts for the PCBs and Raspberry Pi were also designed and 3D printed. To balance the robot's weight, the Raspberry Pi was mounted at the front, while the PCBs at the sides.

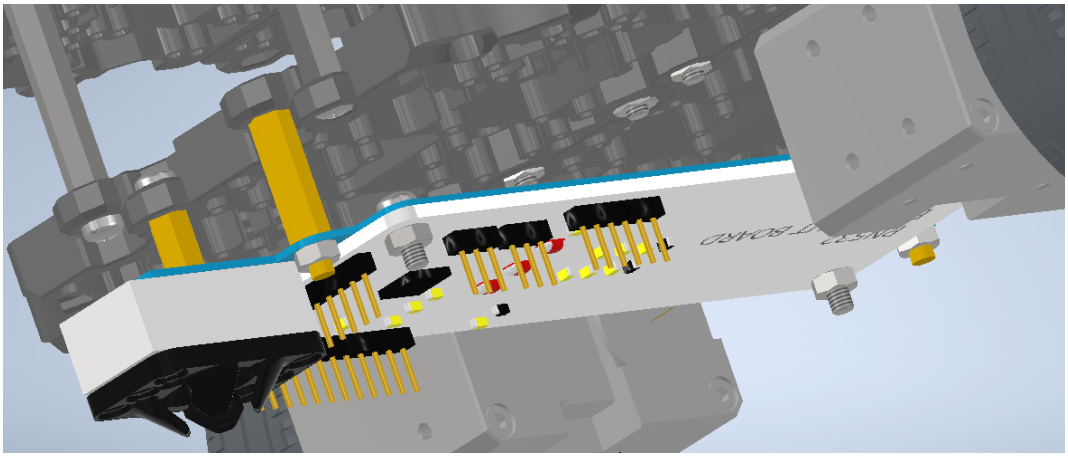


1.4 NFC Tag Reader and Motors

Initially, we used Adafruit PN532 NFC Breakout Board. We mount it below the 1st layer, as close to the ground as possible to detect the NFC tag. However, if it is too close to the ground, the wiring for the NFC reader will be difficult.

To do this, we shifted the motors to the bottom of the 1st layer to create more space for the NFC reader, while the NFC reader was attached to a laser-cut mount, which was then mounted to the

bottom of the 1st layer with spacers so we could fine-tune the distance between the reader and the ground to determine the best distance for reading NFC, as shown below.



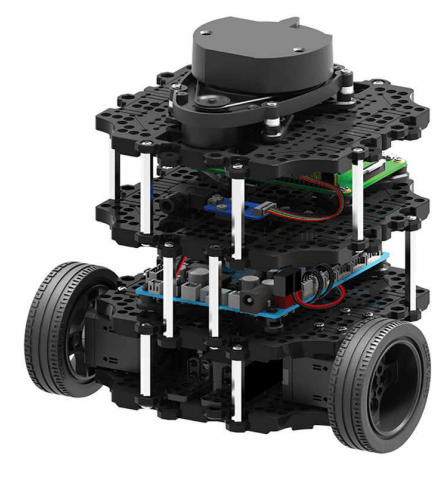
Due to the NFC reader taking up all the space, the steering ball had to be attached to the same acrylic mount since there was no more space to attach it to the waffle. Attaching it like so created a structural weakness and may cause the acrylic mount to break, hence the plan was not approved.

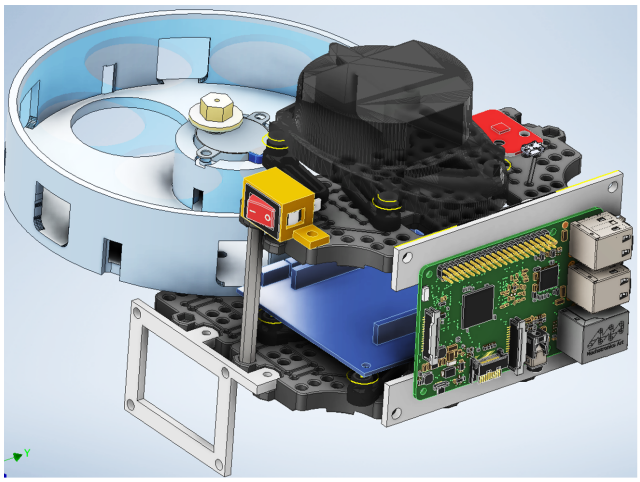
Instead, we decided to use a smaller NFC reader (RC522) so we could mount the steering wheel under the waffle. We also chose not to shift the Turtlebot's motors from their original positions to the bottom of the waffle, as doing so would require a separate spacer for the steering wheel, and it would also increase the robot's height significantly.

There was, however, a change in the orientation of the main motors and steering wheel: we moved the main motors forward. This would make the front part of the robot heavier and help to counterweight the weight of the indexing feeder. Although this would negate the robot's ability to rotate about its centre and increase the turning radius, after some discussion regarding the maze navigation algorithm, it was decided that this would not cause too much trouble.

1.5 Lidar and Detecting Loaded Balls Mechanism

With the Lidar mounted in the original position on the Turtlebot, not only would the 4th layer's space be used inefficiently, but there might also be difficulty with loading balls into the feeder since it was obstructed by the waffle plate ***(Left picture)***. Hence, we decided to use only half of the original waffle plate, and mount the Lidar forward ***(Right picture)***.





With regards to the continuation of the robot's operation after being loaded with 5 ping pong balls, we narrowed our choices down to either (Left to Right).

1. Rocker switch
2. Photosensor
3. Laser transmitter and Laser detector module combination.

In the end, a rocker switch was chosen, as it was less prone to failure. Additionally, it would be much easier to create a mount for the rocker switch as shown in the picture above.

For the photosensor and Laser modules, it will require mounting within the indexing feeder itself such that once the ping pong ball is placed, the light/laser is blocked. These methods may be more complex and more prone to failure.

The rocker switch would also provide tactile feedback to the person loading the ball. He would immediately know what to expect after flipping the switch (robot moving away) instead of wondering whether something went wrong.

1.6 Final Design

After the iterative prototyping process, the final robot's chassis is quite similar to the original Turtlebot in terms of dimensions. Its new features include a flywheel firing mechanism, an indexing feeder, an NFC reader mounted at the bottom, and a thermal sensor mounted at the front.

2. SYSTEM TECHNICAL SPECIFICATIONS

2.1 General System Specifications

Max Translational Velocity	~ 0.19 m/s
Max Rotational Velocity	~ 155°/s (2.7 rad/s)
Dimensions (L x W x H) in mm	~ [257 x 181 x 219]
Total Weight in Grams	~ 1989.6
Center of Gravity (CG)	~ [9.1, 10.6, 8.6]
Wheel Base	~ 90.5 [Horizontal distance between the center of the 2 sprocket wheels & the metal ball caster wheel.]
Expected Charging Time	~30 mins per LiPo
Expected Operating Time	Main battery 1800mAh 3S1P 5C LiPo (23 min) Backup battery Turnigy 2500mAh 3S1P 5C LiPo (32 min)
Single Board Computer	Raspberry Pi 3B+
Microcontroller Unit (MCU)	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
Wheel Motors	Dynamixel XL430-W250
Lidar sensor	LDS-01 360 Laser Distance Sensor
Inertial Measurement Unit	Gyroscope 3-Axis, Accelerometer 3-Axis, Magnetometer 3-Axis. ICM-20648 Digital Motion Processor.
Communication interface	Wifi or Mobile data Hotspot. Laptops and SD cards on the Raspberry Pi 3B+ (RPi) must connect to the same Wifi/hotspot to allow for SSH into the RPi.

2.2 Firing Mechanism Specifications

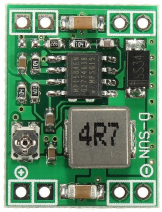
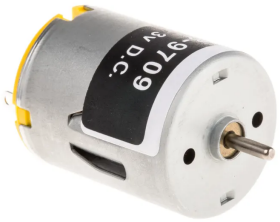
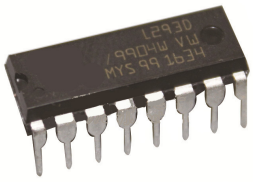
Intended object to be shot	(38 to 40) mm diameter Ping Pong Balls
Maximum capacity	5 Ping Pong Balls
Shooting angle relative to the ground plane	0°
Flywheel Motor	RS PRO Geared DC Motor, 2mm Shaft Diameter
Operating Voltage	1.5V to 3V
Maximum Rotational speed	7800 rpm
Maximum Torque	23.2 gcm
Thermal Sensor	8x8 AMG8833 Infrared Sensor module

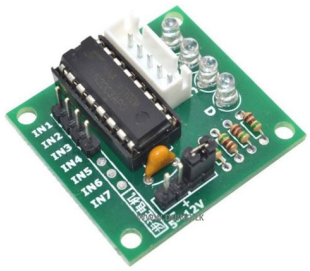
3. ELECTRONICS

3.1 Electronics Component List

Besides electronic components like LiDAR, Raspberry Pi 3b+, OpenCR 1.0 and Dynamixels, these are the other Electronic components listed to **fulfil the main objectives.**

No.	Component	Model	Image	Specifications
1	Battery	3S1P 1800mAh 5C LiPo		• 3S1P • Rechargeable • Low discharge rate at 5C • Mass : 140 g
2	Backup Battery	3S1P 1800mAh 5C LiPo		• 3S1P • Rechargeable • Low discharge rate at 5C • Mass : 140 g

3	NFC Reader	RC522		• Operating Voltage : 2.5V~3.3V • Supports SPI, I2C, UART • Reading Distance up to 5 cm • Dimension : 39 x 60 mm
4	IR Sensor	AMG8833		• Supports I2C • Compatible with Raspberry Pi • Operating Voltage : 3.3V • Temperature accuracy : 2.5°C~3.0°C • Temperature range : -20°C~100°C • Arrays : 8 x 8 pixels • Lense Viewing Angle (FOV) : 60°(horizontal) and 60° (vertical) • Dimensions : 30 x 30 x 5 mm
5	Buck (Step down) Converters [For Protoboard]	MP1584EN		• Adjustable voltage with adjustment potentiometer (clockwise buck, counterclockwise boost) • Output current: 1.5~2A • Input voltage range: 4.5~28V • Output voltage range: 0.8~20V • Dimensions: 22 x 17 x 4 mm
6	Motor for Flywheel	RS PRO Geared DC Motor		• Shaft diameter: 2 mm • Motor Type: Gear • Operating Voltage: 1.5V~3V • Current Rating: 1.06A • Torque Range: 20~23.2 gcm • Rotational Speed: 7800 rpm • Dimensions: 30.5 x 23.8 x 30.5 mm • Mass: 44 g
7	Corresponding Motor Driver	L293D Motor Driver		• Bipolar rotation using H-Bridge • Motor speed control using PWM • Control up to 4 motors

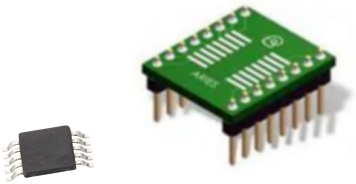
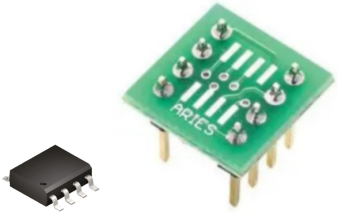
8	Motor to rotate Indexing Ball Feeder	28BYJ-48 Stepper Motor		• Shaft diameter: 5 mm • Motor Type: Gear • Number of Phase: 4 • Operating Voltage: 5V • Frequency 100Hz • Friction torque: 600~1200 gf.cm • Pull-in torque: 300 gf.cm • Speed variation ratio: 1/64 • Stride angle: 5.625°/64 • Dimensions: 35 x 28 (dia.) x 19 mm
9	Corresponding Motor Driver	ULN2003 Stepper Motor Driver Board		• Supply Voltage: 5~12V • 4 input pins (connect to microcontroller output pins) • Maximum Current per output:: 500mA • 4 Step State LEDs
10	Rocker Switch (Activate Robot to move off after receiving balls)	2-PIN Rocker Switch		• 2 Pin • On/Off settings

Load Balancer

Besides the aforementioned components, our team also decided to **optimise the powering system** to allow for longer Turtlebot operation.

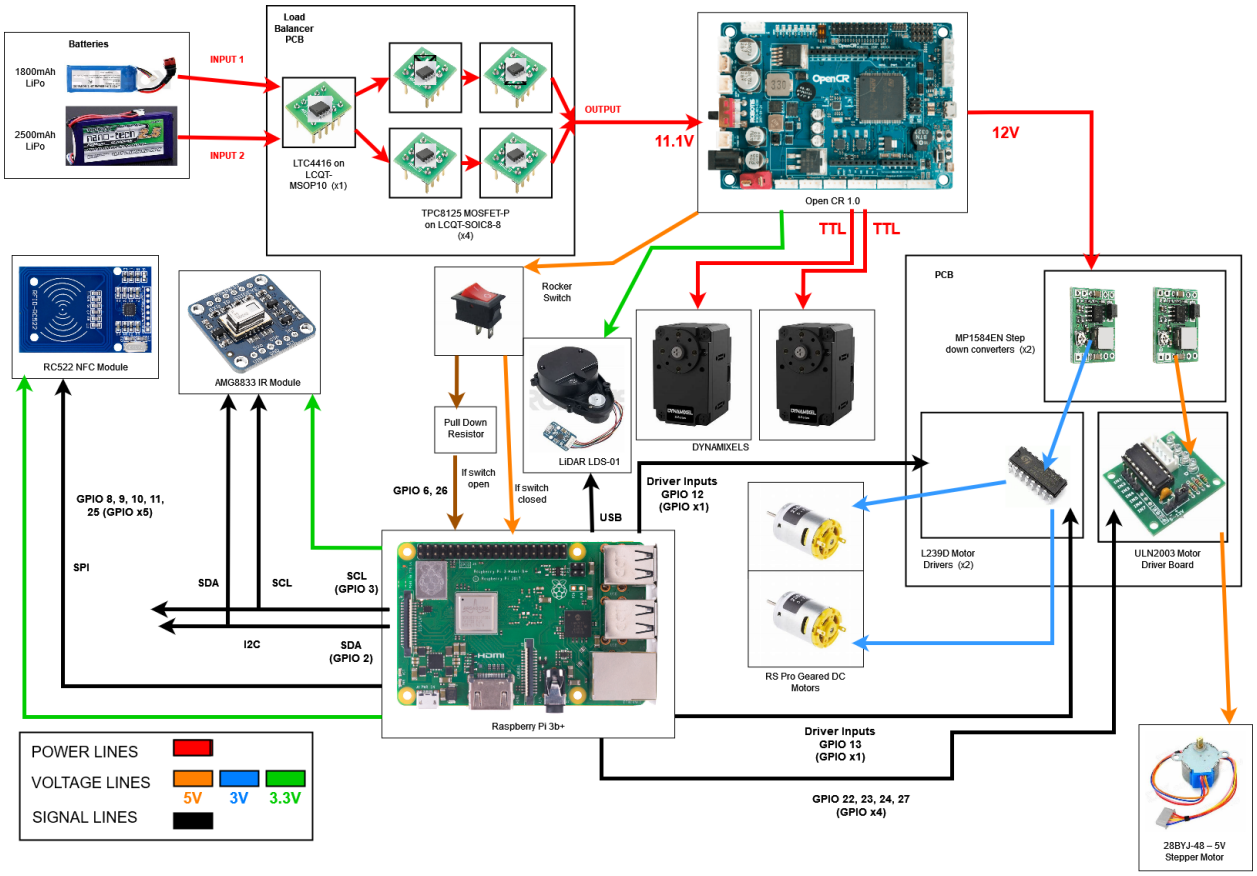
We plan to implement a power path controller system which allows 2 different power sources (LiPo Batteries) to supply power to the system without time lag. There will be alternating usage between the two batteries, where only one battery will supply power to the system at one time.

Below are the electrical components needed to make this Load Balancer.

	
LCQT-MSOP10 (left) will be used to adapt the LTC4416 Power Path Controller IC (x1) (right) to the load balancer PCB/Breadboard.	The LCQT-SOIC8-8 (left) will be used to adapt the TPC8125 MOSFET-P (x4) (right) to the load balancer PCB/Breadboard.
Purpose: IC to control the on and off of the TPC8125 MOSFET-P (x4)	Purpose: Act as a switch between the 2 power supply

3.2 Electronics Systems Architecture

The diagram below shows the overview of the whole electrical system.



This is the summary table of GPIO pins connected from the respective components onto RasPi.

```
#################################################
#                   PINOUT                      #
# GPIO 6: Button Enable                         #
# GPIO 12 (PWM1) : DC Motor1                    #
# GPIO 13 (PWM1) : DC Motor2                    #
# GPIO 22, 23, 24, 27: Stepper Motor Control    #
# GPIO 26: Button                               #
# GPIO 2, 3: SDA, SCL (I2C) for IR camera       #
# GPIO 10, 09, 11, 08: MOSI, MISO, SCK, SS for NFC  #
# GPIO 25: RST for NFC RC522                    #
#                                               #
#################################################
```

GPIO PINOUT

3.2.1 Protoboard

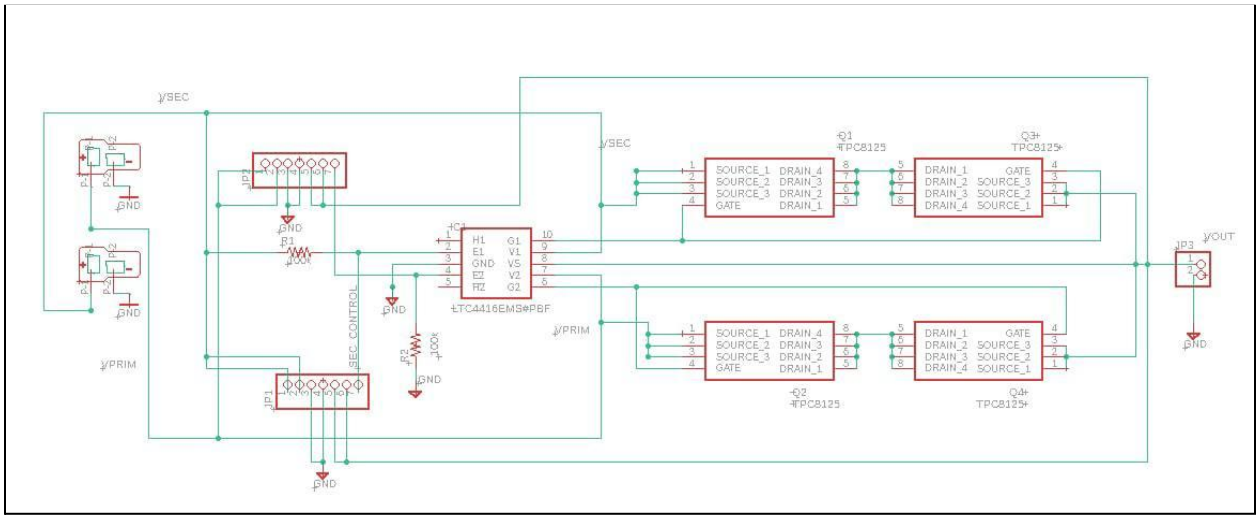
Taking into consideration the mechanical space constraints, we included 2 Protoboards in our system.

Protoboard 1: Load Balancer + L293D Motor Driver + [Flywheel Motor]

The first protoboard will have a power control system to allow instant switching of power supply between the Main battery and the Backup battery. The main battery has primary voltage input (Vprim) and the Backup battery has secondary voltage input (Vsec) to operate at the same time.

However, the LTC4416 IC, which acts as a controller, will trigger the higher voltage supply once the voltage difference between the two power sources reaches more than 100mV. By using the TPC8125 MOSFET-P, it allows the switching process to take place swiftly and eventually produce output at Vout. This allows for a longer operating time as this power system ensures a long-period power supply.

The 11.1V Vout will be channelled to OpenCR to further step up to 12V, which then further connects to Protoboard 2.



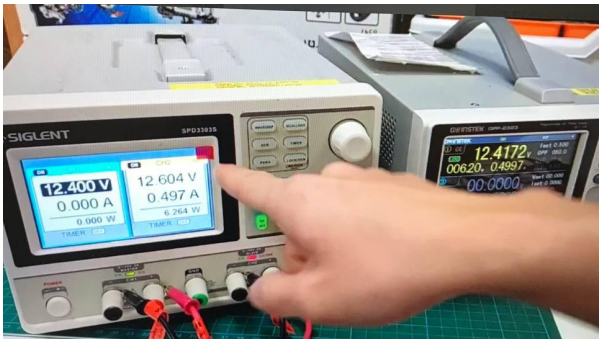
Schematic for Load Balancer

To test if the load balancer is working, we used Channel 1 and 2 from a power supply as Vprim and Vsec, then probing using a multimeter at the testing pin is done to collect the data for Vprim, Vsec and Vout while changing the voltage input at Channel 1 and 2 **alternatingly**. There will be a current-drawing device to test whether it can work normally under current-drawing mode.

Testing of a functional load balancer

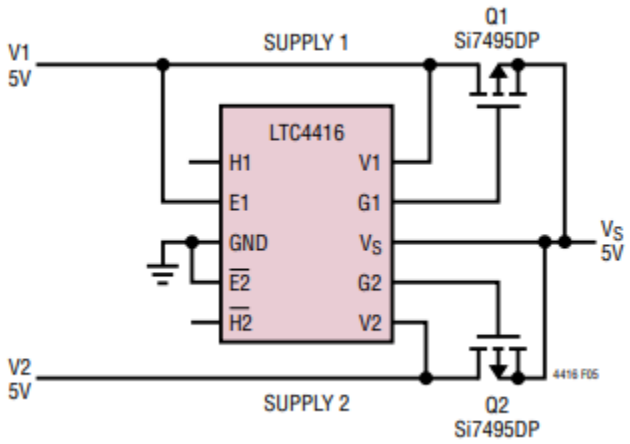
Let Channel 1 == Vprim, Channel 2 == Vsec, Vout==voltage at current-drawing device (rightmost).

**Vout has around 0.1V~ 0.2V of voltage drop which is acceptable due to the internal resistance of the components.

	• Vprim=12.383V • Vsec=12.202V • Vout= 12.1082V • Current drawn: 0.495A • **Current drawn from Channel 1 (Vprim)**
	• Vprim= 12.400V • Vsec= 12.604V • Vout= 12.4172V • Current drawn: 0.497A • **Current drawn from Channel 2 (Vsec)**
	• Vprim= 12.601V • Vsec=12.604V • Vout= 12.4585V • Current drawn: 0.219A for Channel 1 and 0.272A for Channel 2 • **Current drawn from both Channel 1 and 2 evenly (Both Vprim and Vsec evenly)**

Defects and Troubleshooting

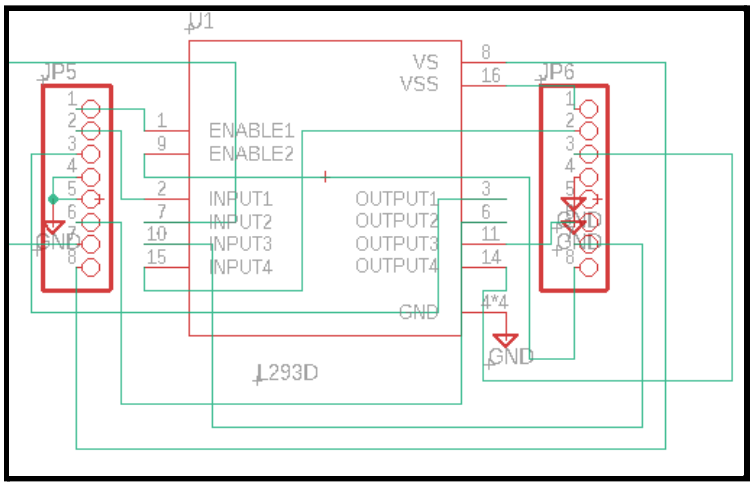
Notably, the team had accidentally burnt 2 out of 5 of the TPC8125 PMosfet due to the shorted terminal while probing and reverse the polarity of power input. To salvage the predicament as the team is financially not viable to afford additional IC Chip procurement, the team had referred to the Datasheet and found an alternative possible circuit using only 2 PMosfet instead of 4. By using Jumper wires across the Drain and Source Pins, the load balancer is successfully modified and is tested to be functional even under 0.5A current-drawing mode.



Alternative circuit using only 2 Pmosfets

L293D and Gear Motors

L293D allows driving of at most 4 motors at the same time and different directions of motor rotation and speed control.

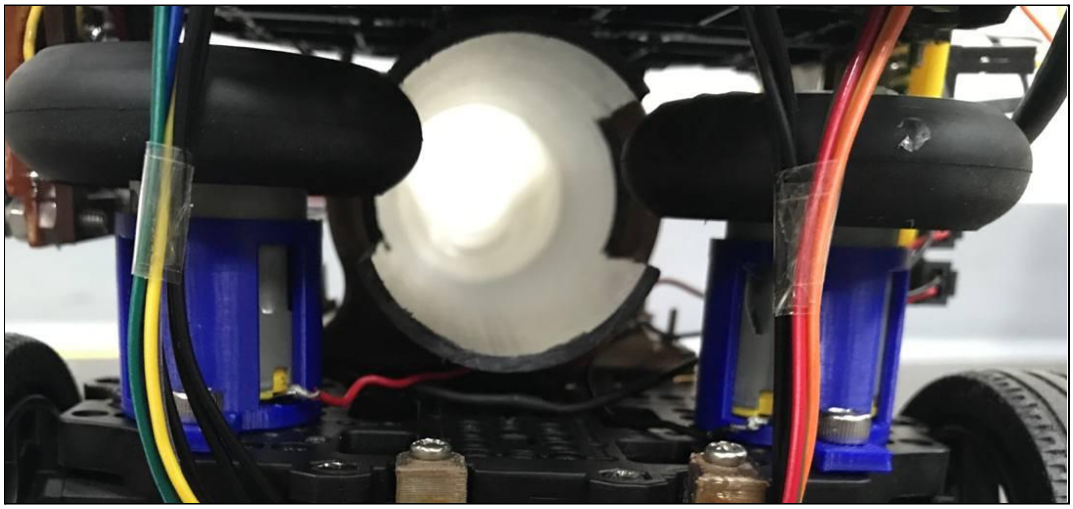


Schematic for L293D Motor Driver

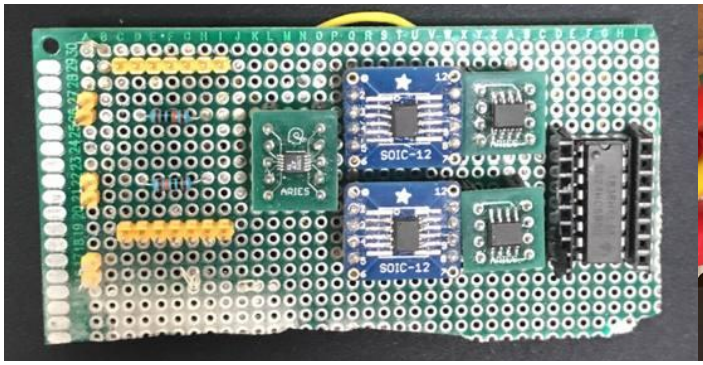
As it is not wise to provide a high current control signal from the Raspberry Pi, L293D is used to drive the two gear motors (RS PRO Geared DC Motor) for flywheels. L293D will combine your PWM control signals from the Raspberry Pi with the high current from Vcc2 supplied to the IC. PWM signals are also used to control the speed of the flywheel motor so that you can decide the range for your ball shooter.

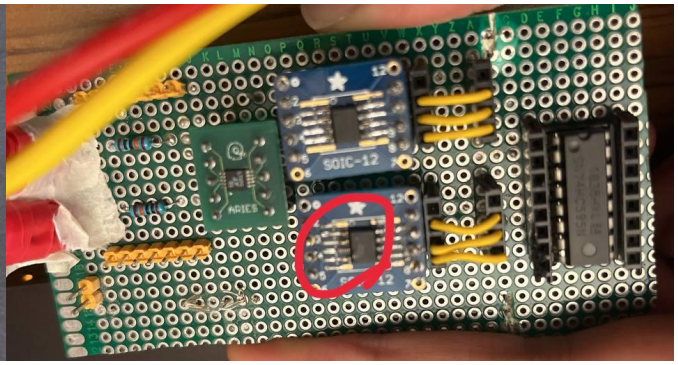
As it is a geared motor, it could amplify the torque produced using the gear ratio of the internal gear system with only very low power consumption, which facilitates the actuation. This can be shown through the maximum torque produced at 23.2gcm among the three motors with the operating voltage range from 1.5V to 3V and a comparatively low current rated at 1.06A.

Also, its high rpm up to 7800rpm allows the ping pong ball to possess a high initial horizontal speed after being launched, therefore the ball could travel a longer horizontal distance with negligible vertical displacement change, hence aims and shoots more accurately. On top of that, it operates at a low voltage ranging from 1.5V ~ 3V and draws a relatively low current at 1.01A.



The two gear motors in the flywheel mechanism



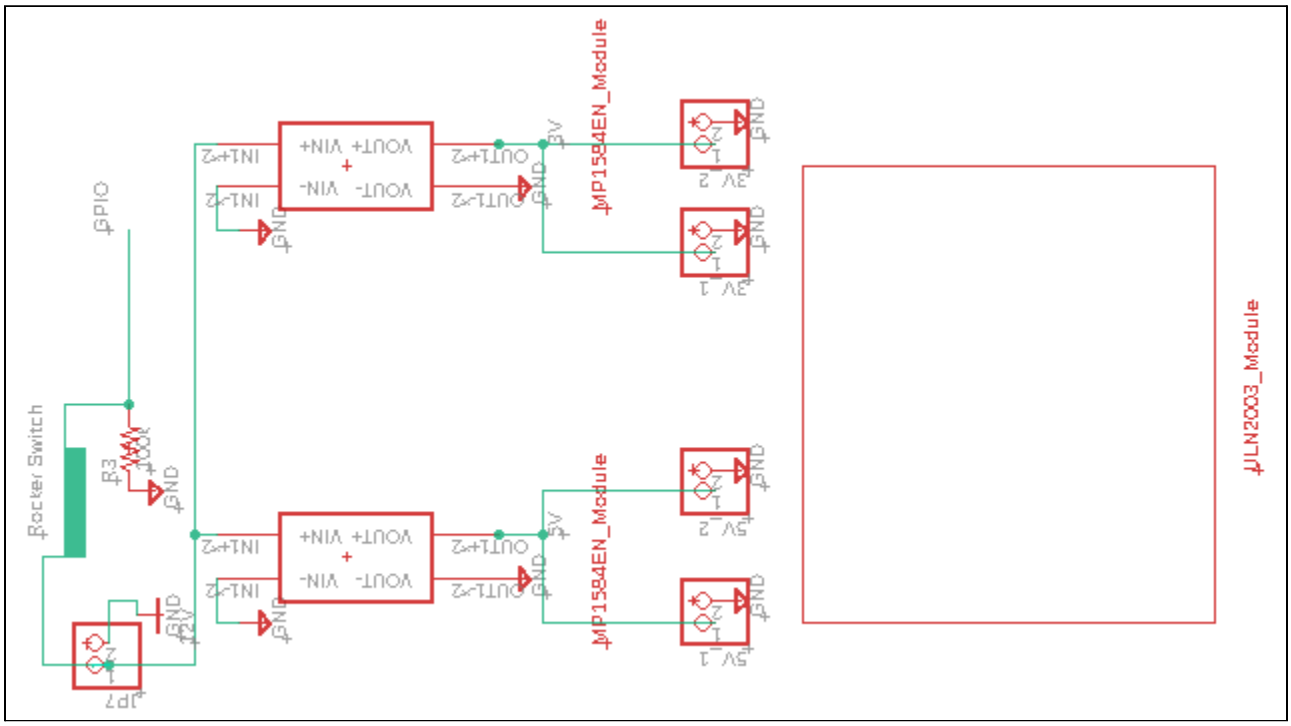


The actual Protoboard 1 (Complete version) **The reduced version of Protoboard 1**

P/s: In the reduced version, we don't use the load balancer. This is because we wanted to reduce the weight of our robot in exchange for the speed to map the maze.

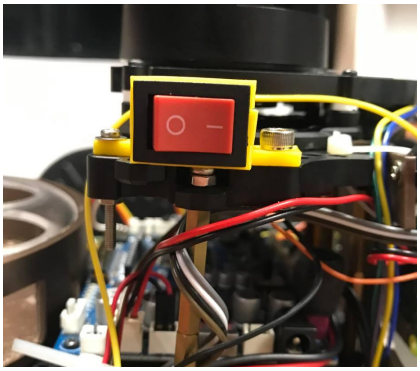
Protoboard 2: Buck Converter + Pull Down Resistor Circuit (Rocker Switch) + ULN2003 Stepper Motor Driver + [28BYJ-48 Stepper Motor]

The second protoboard mainly comprises buck converters, pull-down resistor circuit (Rocker Switch) and mechanically-mounted Stepper Motor Driver.



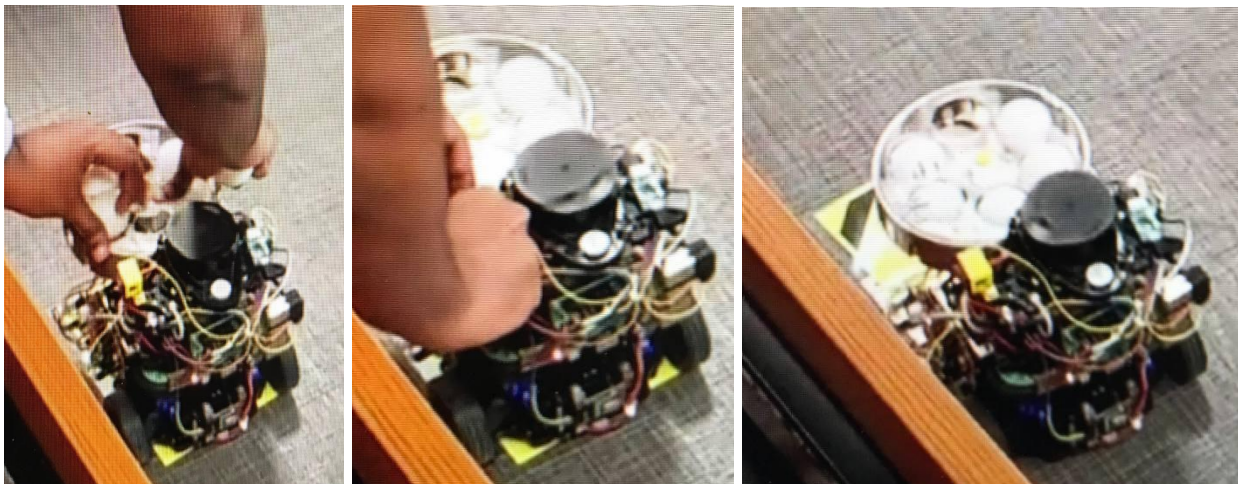
Schematic of Protoboard 2

Protoboard 2 will receive 12V input from the OpenCR, then step down the voltage to 5V and 3.3V respectively using different MP1584E Buck Converters. The output of the buck converters will be branched out parallelly 2 times to provide more output pins, so as to accommodate more components. The reason for supplying 12V to Protoboard 2 before stepping down the voltage is to ensure sufficient power distribution to the components on Protoboard 2 despite low voltage operation.

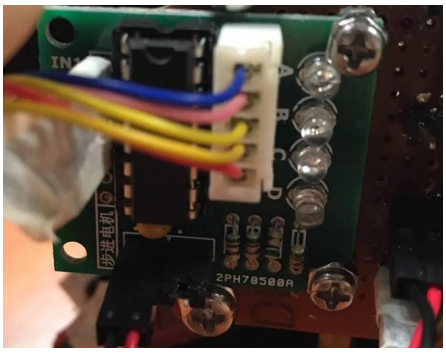


Meanwhile, the team connected a rocker switch to a pull-down resistor circuit. The resistor was connected to a voltage supply, rocker switch and GPIO. If the rocker switch is off, it will be an open circuit and GPIO detects nothing. If the rocker switch is on, it will now be a closed circuit and GPIO will detect a positive signal. This logic design will help to actuate the Turtlebot manually to continue its auto-navigation after it has stopped and collected 5 ping pong balls.

Rocker switch mounted on Turtlebot

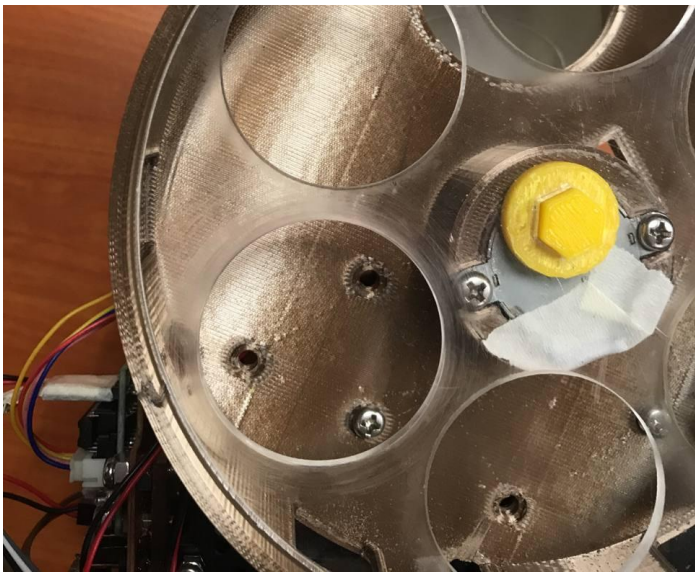


Load ping pong balls onto the stationary Turtlebot, turned on the rocker switch and actuate again the auto-navigation



For the ULN2003 Stepper driver module, it is a ready-to-use board and hence, it is mechanically mounted on Protoboard 2.

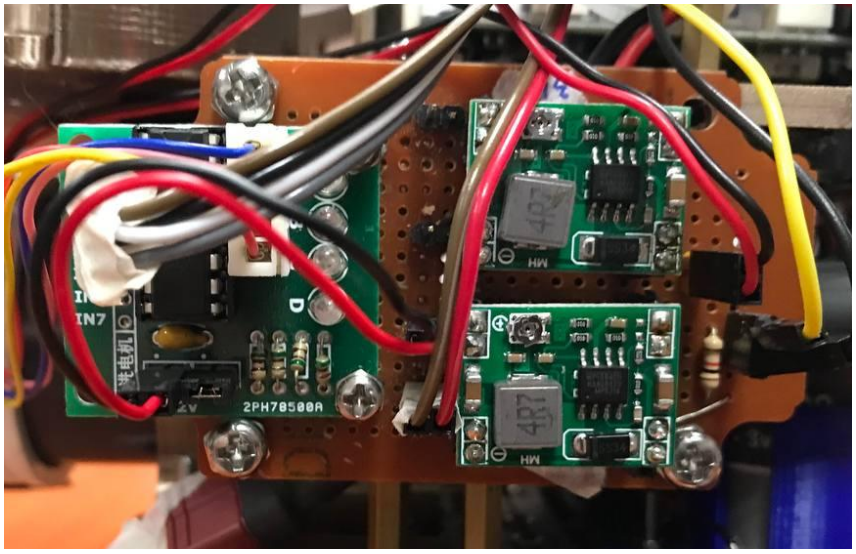
ULN2003 Module mounted on Protoboard 2



For the motor that controls the acrylic disk ping pong holder at specific step angles, 28BYJ-48 Stepper Motor is chosen.

As it is a stepper motor, it allows precise speed control, positioning and produces low-speed torque. Moreover, it has a metal shaft to produce high friction torque ranging from 600gf.cm~1200gf.cm. It is owned by the members and thus could reduce the budget. On top of that, it is specifically designed to work with ULN2003 stepper driver module.

28BYJ-48 Stepper Motor mounted on the rotary plate



The actual Protoboard 2

3.2.2 Infrared Sensor (AMG8833)

To choose the ideal IR sensor for this project, the properties would be:

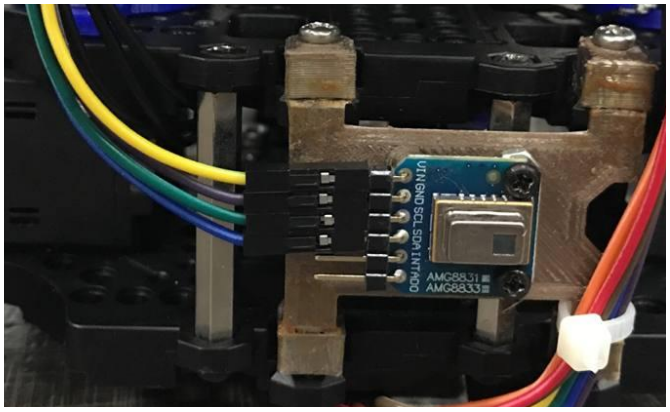
- Greater number of pixels
- Higher maximum temperature
- Higher temperature precision
- Working well with Raspberry Pi
- Has I2C protocol
- Longer distance of detection
- Greater Field Of View (FOV)
- Most importantly, the price should be affordable.

The team has selected AMG8833 as the thermal sensor.

AMG8833 works well with Raspberry Pi and has an I2C protocol. On top of that, it comes with 8x8 pixels which allow precise location. Though its maximum temperature is comparatively lower at 100℃, it is still sufficient for this project. Also, the temperature precision is considered high at 2.5 ℃~3.0 ℃. Importantly, its FOV is essentially wide at 60 ° (horizontal) and 60 ° (vertical), which will provide a huge scopic view. Notably, the average price of AMG8833 from local distributors is around SGD50++, but the member is able to source from a Chinese supplier on Shopee at SGD33(Delivery fee inclusive).

Operating AMG8833

Connect the SDA, and SCL pins on AMG8833 to GPIO 2, and 3 on the Raspberry Pi 3B+ respectively. Connect 3.3V and GND pins on RC522 to 3.3V and GND pins on the Raspberry Pi 3B+. Once properly connected, run the code. The AMG8833 should be able to detect a temperature surface within the -20°C~100°C range.



Connection of AMG8833 on the actual Turtlebot

3.2.3 NFC Tag Reader (RC522)

The team had done a technical comparison to decide whether to use the given Adafruit PN532 Breakout board or the RC522.

Component Model	Adafruit PN532 NFC Breakout Board	RC522
Picture		
Similarities :	-Have host interfaces (SPI, I2C, UART)	
Differences:	-Size 5.1 x 11.77 x 0.11cm -2.7V~5.5V power supply operating range -6 Different Operating Modes -Types of cards supported (ISO/IEC 14443A/MIFARE) -Reading distance up to 10 cm depending on mode and antenna: (1)Typical operating distance in ISO/IEC 14443A/MIFARE or FeliCa card emulation mode of approximately 10 cm. (2)Typical operating distance in Read/Write mode up to 5 cm. (3) Working distance up to 5 cm in NFCIP-1 mode	-Size 3.9 x 6 cm -2.5V~3.3V power supply operating range -1 Operating Mode -Types of cards supported (ISO/IEC 14443 A/MIFARE and NTAG.) -Reading Distance with ISO 14443A / MIFARE® in the reader mode is up to 5 cm, depending on the length and tuning of the antenna

Overall, though there are slight differences in their technical properties, it will bring negligible differences in the outcome. Therefore, the determinant of choice for the team is the size. As the **RC522** module is smaller, it would be mechanically feasible to locate it at the bottom of Turtlebot.

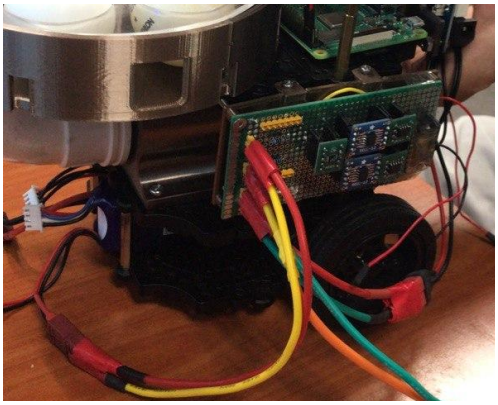
Operating RC522

Connect the MOSI, MISO, SCK, SS, and RST pins on RC522 to GPIO 10, 9, 11, 8 and 25 on the Raspberry Pi 3B+ respectively. Connect 3.3V and GND pins on RC522 to 3.3V and GND pins on the Raspberry Pi 3B+. Once properly connected, run the code. The RC522 should be able to detect an NFC Tag within a 5cm distance.

RC522 is mounted at the bottom of Turtlebot and wire connections

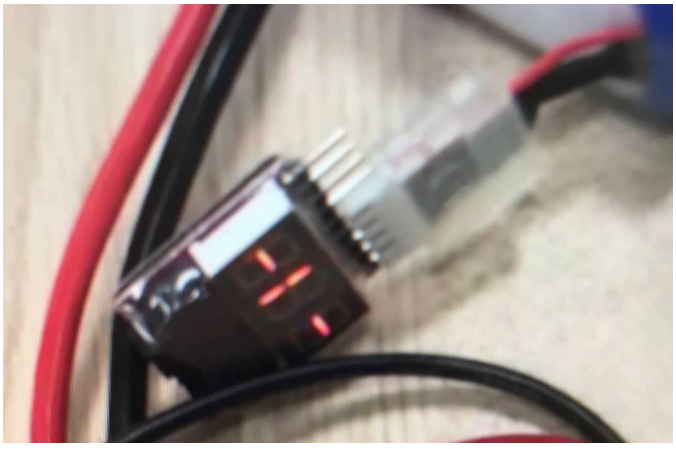
3.2.4 Battery

The 2 batteries are placed in the first layer of the Turtlebot. There are 2 possible installation modes for the battery – 2 batteries or 1 battery.

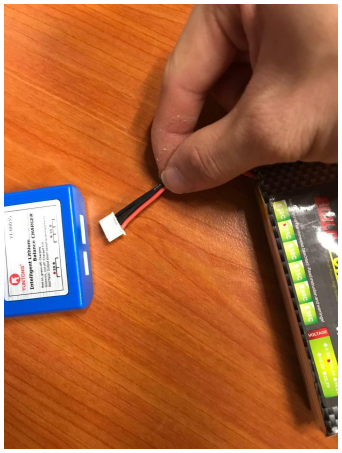
2 Batteries (With Load Balancer)	1 Battery (Without Load Balancer)
 The two batteries are connected to the load balancer using 16AWG wires with Dean Connectors.	The single battery is directly connected to the OpenCR board as the Turtlebot Emanual

Charging

Check the voltage level of the battery using the LiPo Battery voltage tester as shown in the picture below. Ensure the pins are correctly connected.

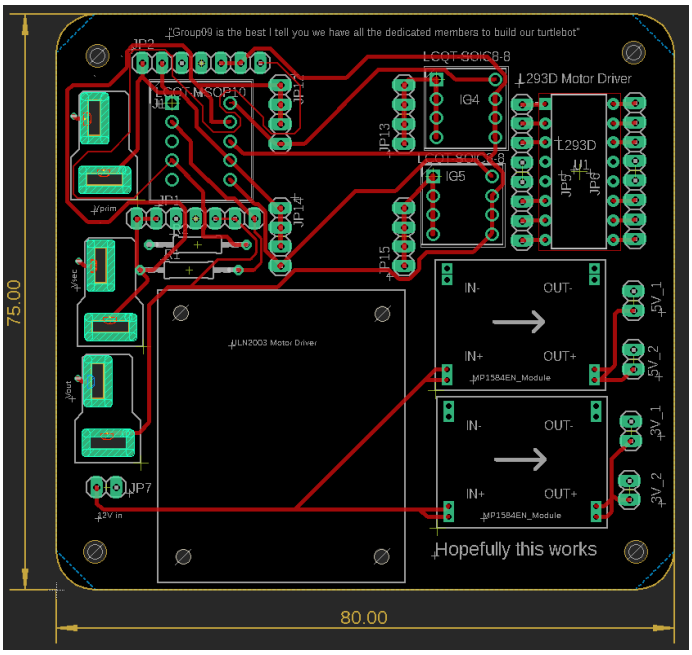


If the battery has insufficient energy, you will need to charge the LiPo Battery.

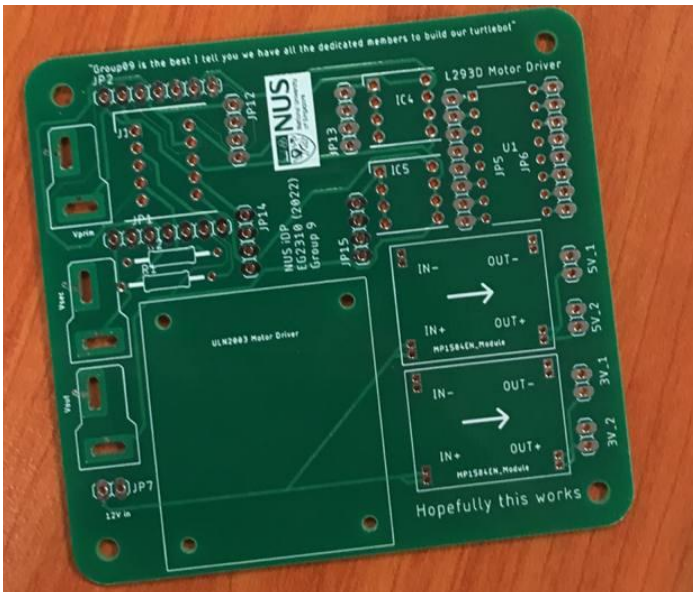
			
Components.	First, connect the LiPo Battery to the Charger.	Secondly, connect the charger to a power outlet.	Lastly, the red light on the Charger means charging is in progress. Green light on the Charger means the battery is fully charged.

3.2.5 Printed Circuit Board (PCB)

The team has also planned to install a Printed Circuit Board to save space and increase the efficacy of the electrical systemic operation. The diagram below shows the design of the PCB using Eagle.



PCB Drawing on Eagle



Our actual PCB

Unfortunately, the Vprim and Vsec have no soldering point on the actual PCB due to unknown reasons. However, this can be resolved easily using a jumper wire to connect to the test pins of Vprim and Vsec. Eventually, due to the faulty manufacturing process, which result in our board missing power trace, the team decided to go for the protoboards instead as they have gone through many rounds of testing and were proven to work consistently.

3.3 Power Calculations

Typical and Surge power consumption of turtlebot

Initial Boot up

Lipo Capacity / %	Typical Voltage / V	Typical current / mA	Typical power / W	Typical power consumption / Wh	Surge Voltage / V	Surge Current / mA	Surge power / W	Surge power consumption / Wh
100	12.596	388	4.887248	22.6728	12.596	692	8.716432	22.6728
50	11.506	413	4.751978	20.7108	11.506	760	8.74456	20.7108
15	11.097	430	4.77171	19.9746	11.097	784	8.700048	19.9746

Standby/Idle

Lipo Capacity / %	Typical Voltage / V	Typical current / mA	Typical power / W	Typical power consumption / Wh	Surge Voltage / V	Surge Current / mA	Surge power / W	Surge power consumption / Wh
100	12.596	388	4.887248	22.6728	12.596	488	6.146848	22.6728
50	11.506	413	4.751978	20.7108	11.506	560	6.44336	20.7108
15	11.097	430	4.77171	19.9746	11.097	512	5.681664	19.9746

During Operation

Lipo Capacity / %	Typical Voltage / V	Typical current / mA	Typical power / W	Typical power consumption / Wh	Surge Voltage / V	Surge Current / mA	Surge power / W	Surge power consumption / Wh
100	12.596	435	5.47926	22.6728	12.596	488	6.146848	22.6728
50	11.506	465	5.35029	20.7108	11.506	500	5.753	20.7108
15	11.097	479	5.315463	19.9746	11.097	549	6.092253	19.9746

For Turtlebot:

1. **During the initial boot up, the current will first increase and then fall back to a typical value.**
   Surge current is defined as the highest possible value of current drawn before it falls back to a typical value.
   Typical Current is defined as the average estimated value during the booting up process.

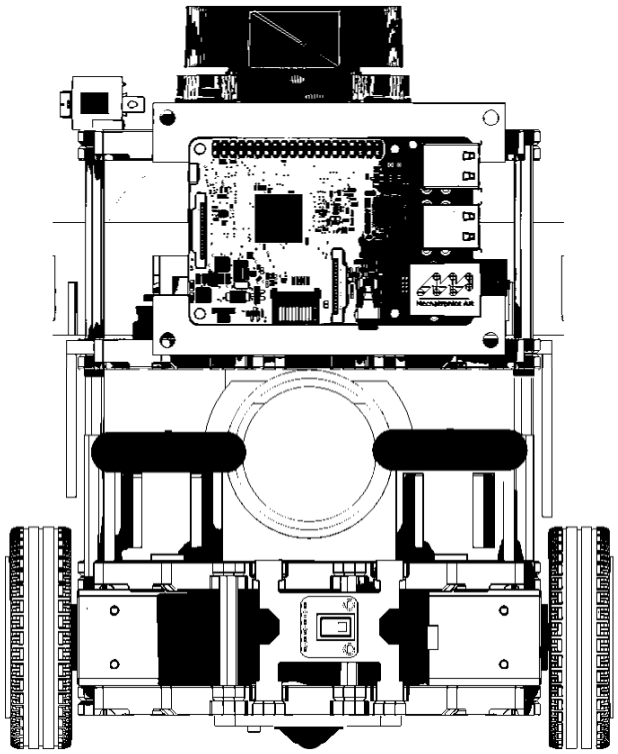
2. **During the idle/standby state, the current drawn should be at a relatively steady value. Surge Current is defined as the highest possible value of current drawn when there is a simple operation executed. (In our context, we use "ssh into RPI from laptop" as the operation).**
   Typical Current is defined as the average estimated steady value during the idle/standby state.

3. **During the operation state, the current drawn will be higher than the idling stage. To define an operation state, it is a state when there is an action conducted by the Turtlebot (eg execution of software) (In our context, we choose r2mover)**
   Surge Current is defined as the highest possible value of current when a vigorous operation is ongoing. (In our context, it is when "w" is executed in r2 mover.)
   Typical Current is defined as the estimated average value of current under an idling operation state. (In our context, it is when r2mover is activated but no further command is given.)

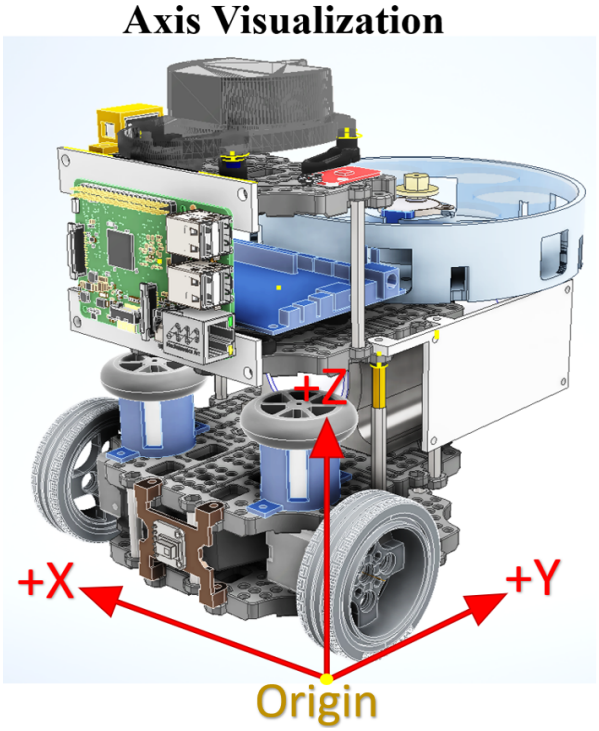
*Using max V in given range	Component	Typical Voltage / V	Max Current / mA	Max Current / A	Power / W	Qty
	RC522 NFC Reader Module	3.3	26	0.026	0.0858	1
	AMG8833 IR Sensor Module	3.3	10	0.01	0.033	1
	MP1584EN Buck Converter	12	0.125	0.000125	0.0045	3
	RS PRO Geared DC Motor	3	1060	1.06	6.36	2
	28BYJ-48 Stepper Motor (to rotate wheel)	5	40	0.04	0.2	1
	ULN2003 Board (motor driver)	5	500	0.5	2.5	1
	L293D (motor driver)	3	1200	1.2	7.2	2
	Dynamixel Motor (XL430-W250-T)	11.1	52	0.052	1.1544	2
	Open CR 1.0	11.1	800	0.8	8.88	1
	Rasberry Pi 3b+	5	2500	2.5	12.5	1
	Servo Motor	5	150	2.2	11	1
	LiDAR (LDS-01)	5	400	0.4	2	1
	Total power				51.9177	

Choose appropriate battery based on component with highest voltage and highest current				
Battery choices	Battery Voltage (V)	Battery Capacity (Ah)		Operation time (t)
Turnigy 2500mAh 3S1P 5C LiPo	11.1	2.5		32 min
1800mAh 3S1P 5C LiPo	11.1	1.8		23 min

4. MECHANICAL



4.1 Design Calculations



System Total Mass and Center of Gravity(CG)

Start of +X and +Z Axis are taken at the edge of the Left wheel. While the start of the +Y axis is 2cm from the left wheel.

This is so that all the distance measurements are Positive as the OpenCR is mounted at the front of the robot.

The ORIGIN is depicted in the MAIN VIEW on the right.

Components	Mass/g	Position X /cm	Position Y /cm	Position Z /cm	CG X	CG Y	CG Z
Left wheel	14.5	0	2	0	0	0.01457579413	0
Right wheel	14.5	18	2	0	0.1311821472	0.01457579413	0
Waffle plate 1 (most top)	78.5	9	9	16.4	0.3550965018	0.3550965018	0.6470647366
Waffle plate 2 + 3 (combined)	78.5	9	9	11.7	0.3550965018	0.3550965018	0.4616254524
Waffle plate 3 + 4 (combined)	78.5	9	9	5.5	0.3550965018	0.3550965018	0.2170034178
Waffle plate 5 + 6 (most bottom)	78.5	9	9	1.5	0.3550965018	0.3550965018	0.0591827503
LiPo Battery 1	143	7	8.75	3	0.5031162043	0.6288952553	0.2156212304
LiPo Battery 2	160	11	8.75	3	0.8845999196	0.7036590269	0.2412545235
360 Laser Distance Sensor	112	9	6.5	20	0.5066344994	0.365902694	1.125854443
Open CR 1.0	66	9	1	14.2	0.2985524729	0.03317249698	0.4710494572
Open CR 1.0 Mount	15	9	1.75	14.2	0.06785283474	0.01319360676	0.1070566948
Raspi 3	49	9	5.5	12.9	0.2216525935	0.1354543627	0.3177020507
Left DYNAMIXEL XL430	114	3.5	6.5	3.3	0.2005428227	0.3724366707	0.1890832328
Right DYNAMIXEL XL430	114	13.5	6.5	3.3	0.773522316	0.3724366707	0.1890832328
AMG8833	20	9	1.9	4	0.09047044632	0.01909931645	0.04020908725
AMG8833 Holder	5	9	2	4	0.02261761158	0.00502613590	0.01005227181
Left RS Pro Motor	120	4	8.75	8.6	0.2412545235	0.5277442702	0.5186972256
Left Motor Mount	5.2	4	8.75	8.6	0.01045436269	0.02286891838	0.02247687977
Left Flywheel	15	4	8.75	9.5	0.03015681544	0.06596803378	0.07162243667
Right RS pro Motor	120	13.5	8.75	8.6	0.8142340169	0.5277442702	0.5186972256
Right Motor Mount	5.2	13.5	8.75	8.6	0.03528347407	0.02286891838	0.02247687977
Right Flywheel	15	13.5	8.75	9.5	0.1017792521	0.06596803378	0.07162243667
Left PCB	22	0.6	11	9.7	0.006634499397	0.1216324889	0.1072577402
Left PCB Mount	8	1.2	11	9.7	0.00482509047	0.04422999598	0.03900281464
Right PCB	37	20.7	11.5	8.5	0.3849517491	0.2138620828	0.1580719743
Right PCB Mount	14	19.9	11.5	8.5	0.1400281464	0.0809207881	0.05981101729
RC522 NFC	15	9	9.2	0.6	0.06785283474	0.06936067551	0.004523522316
NFC Mount	5	9	9.2	0.6	0.02261761158	0.02312022517	0.001507840772
Shooting Barrel	132	9	27	8.7	0.5971049457	1.791314837	0.5772014475
Shooting Barrel Mount	55	9	15.5	8.7	0.2487937274	0.428478086	0.2405006031
Indexing Feeder Container	153	9	17.5	12.9	0.6920989144	1.345747889	0.9920084439
Stepper Motor	95	9	17.5	14.1	0.42973462	0.8355950945	0.6732509047
Acryllic Disk	32.2	9	17.5	14.3	0.1456574186	0.2832227583	0.231433454

System total Mass/g	1989.6		
Total CG in the X-Axis/cm	9.094591878	rounded to	9.1cm (1dp)
Total CG in the Y-Axis/cm	10.5694612	rounded to	10.6cm (1dp)
Total CG in the Z-Axis/cm	8.602005428	rounded to	8.6cm (1dp)
COORDINATE OF SYSTEM CG	**(9.1, 10.6, 8.6)**		

The CG of the entire system for each axis is calculated using the formula:

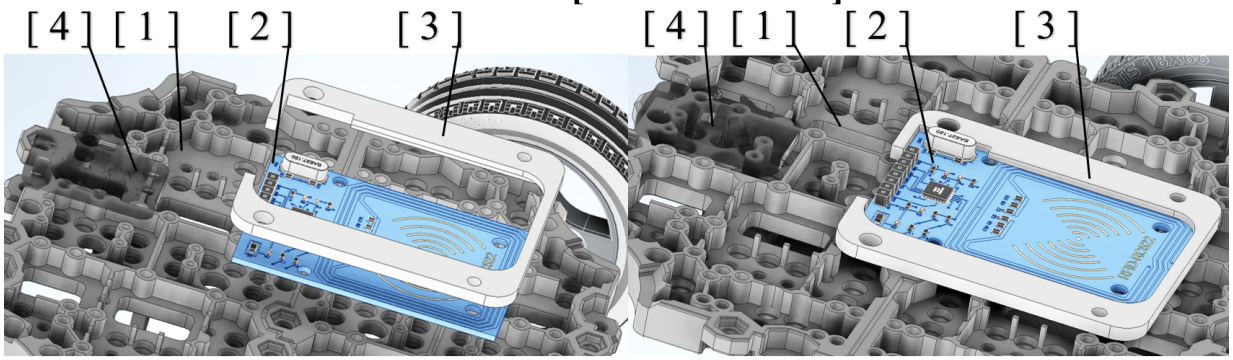
$$X_{CG} = \frac{\sum W_i X_i}{\sum W_i}$$ **[Same formula is used for total CG along Y and Z axis]**

The centre of gravity is estimated by summing the product of weights and locations (Moments) of individual masses and then dividing the total system mass. The distance is computed from the origin.

4.2 Mechanical System Assembly

****Nuts and Bolts are excluded from the CAD assembly for Simplification****

STAGE 1 [Bottom View]



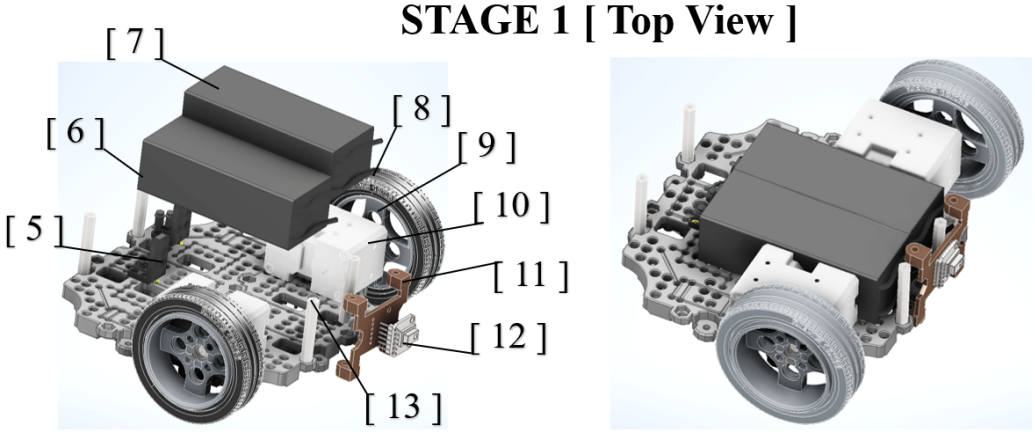
Exploded View

Assembled View

Part Number	Part Name	Quantity
1	Waffle Plate	2
2	RC522 NFC Reader	1
3	NFC Mount	1
4	Steering Wheel Support	1

Screws and Nuts	Quantity
M3x12 Screw	2
M3 Nut	6
M4x12 Screw	2
M4 Nut	2
M3x8 Screw	4

STAGE 1 [Top View]

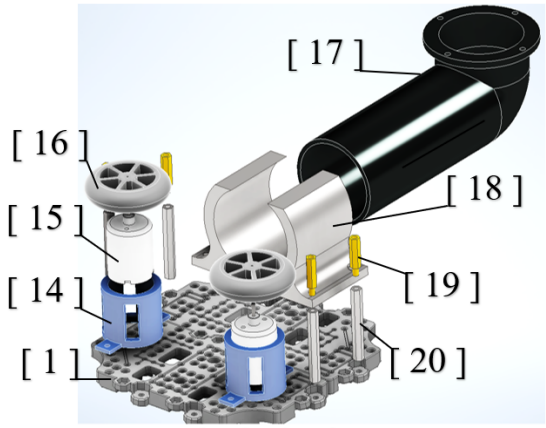


Exploded View

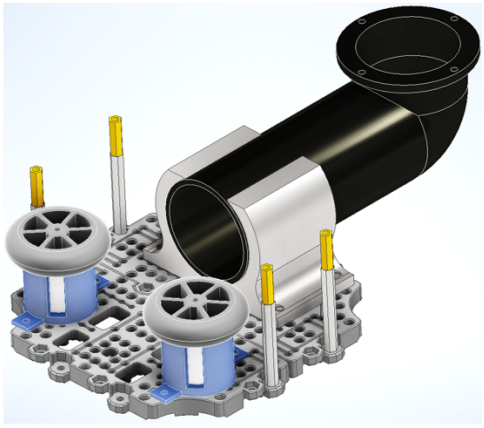
Assembled View

Part Number	Part Name	Quantity	Screws, Nuts and Rivets	Quantity
5	L Bracket mount	2	M3x12 Screw	2
6	Li-Po Battery 1	1	M3x8 Screw	4
7	Li-Po Battery 2	1	M3 Nut	6
8	Tire	2	PH_T2.6x12	16
9	Wheel	2	M2.5x8 Screw	2
10	Dynamixel XL430-W250	2	M2.5 Nut	2
11	Infrared Sensor Mount	1	Rivet	4
12	AMG8833 Infrared Sensor	1		
13	Beam [M3x35]	4		

STAGE 2



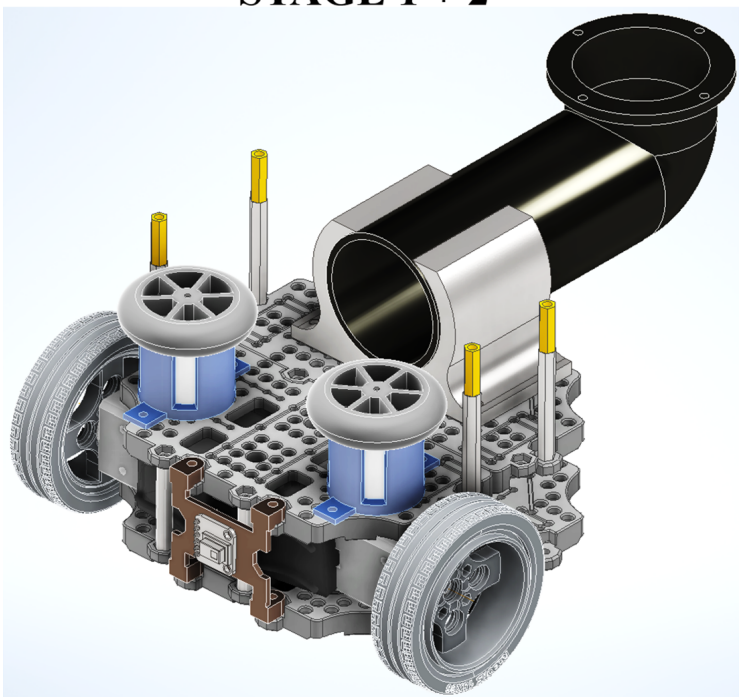
Exploded View


Assembled View

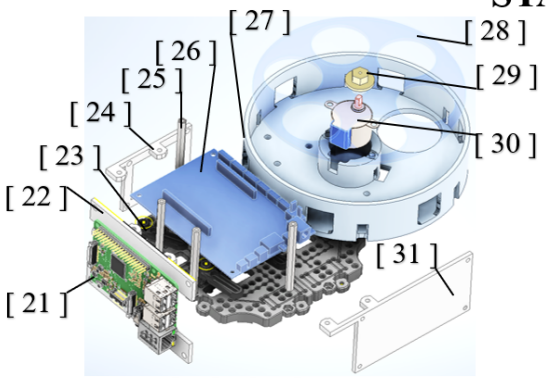
Part Number	Part Name	Quantity
1	Waffle Plate	2
14	DC Motor Mount	2
15	RS Pro geared DC Motor	2
16	Flywheel	2
17	Shooting Barrel	1
18	Barrel Mount	1
19	Beam [M3x15]	4
20	Beam [M3x40]	4

Screws, Nuts and Rivets	Quantity
M3x8 Screw	16
M3x12 Screw	2
M3 Nut	18
M4x8 Screw	2
M4x12 Screw	2
M4 Nut	4

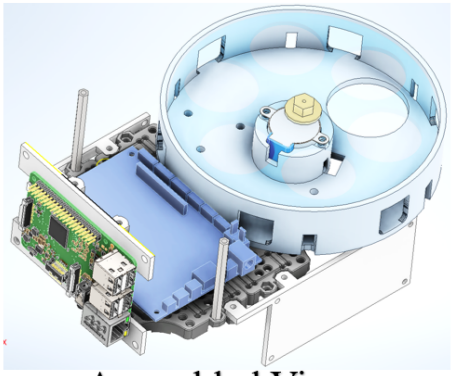
STAGE 1 + 2



STAGE 3



Exploded View

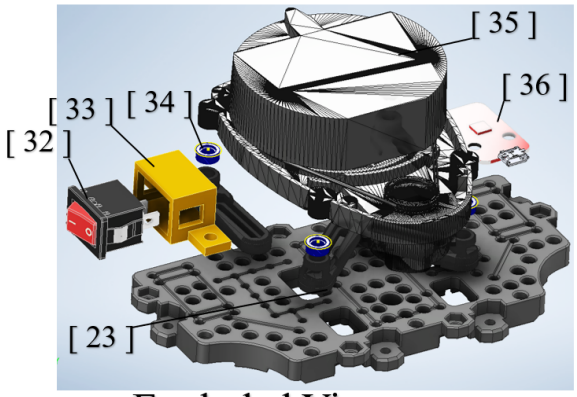


Assembled View

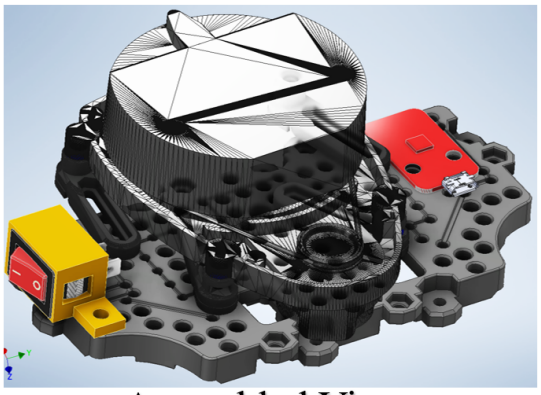
Part Number	Part Name	Quantity
21	Raspberry Pi 3B+ (RPi)	1
22	RPi Mount	1
23	Board Bracket	4
24	Protoboard 1 Mount	1
25	Beam [M3x60]	4
26	OpenCR 1.0	1
27	Ball Storage Container	1
28	Acrylic Disk	1
29	Stepper Motor Shaft	1
30	28BYJ-48 Stepper Motor	1
31	Protoboard 2 Mount	1

Screws, Nuts and Rivets	Quantity
M2.5x8 Screw	6
M2.5x12	4
M2.5 Nut	10
M3x8 Screw	12
M3x12 Screw	6
M3 Nut	10
M4x8 Screw	4
M4x16 Screw	2
M4 Nut	6

STAGE 4



Exploded View

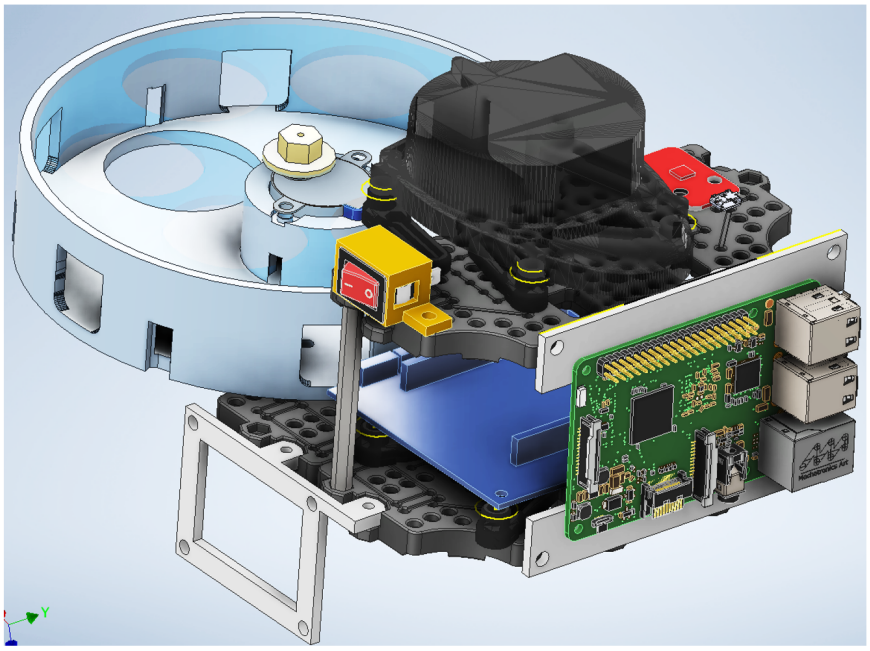


Assembled View

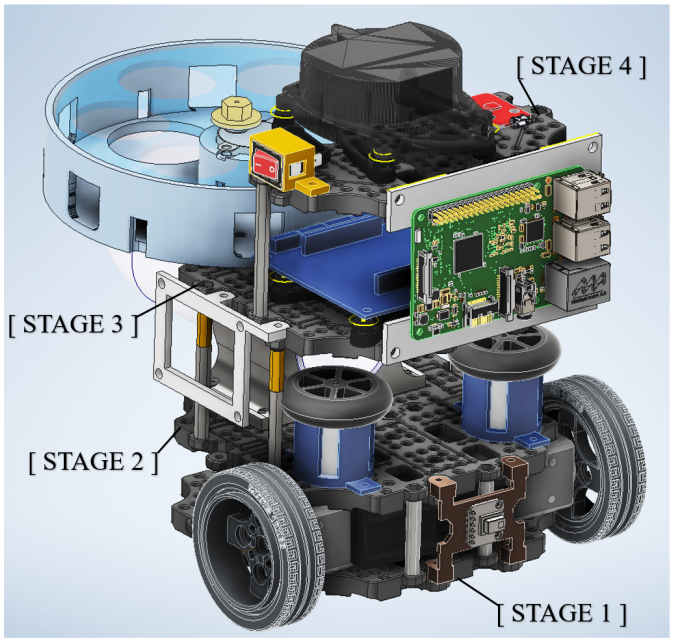
Part Number	Part Name	Quantity
23	Board Bracket	4
32	Rocker Switch	1
33	Switch Mount	1
34	Spacer	4
35	360 Laser Distance Sensor [LDS-01]	1
36	USB2LDS Connector	1

Screws, Nuts and Rivets	Quantity
Rivet	2
M2.5x8 Screw	4
M2.5x16 Screw	4
M2.5 Nut	8
M3x8 Screw	6
M4x8 Screw	2
M4 Nut	2

STAGE 3 + 4

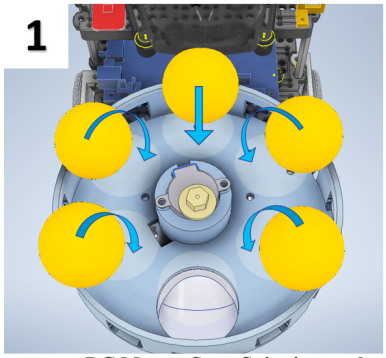


FINAL ASSEMBLY

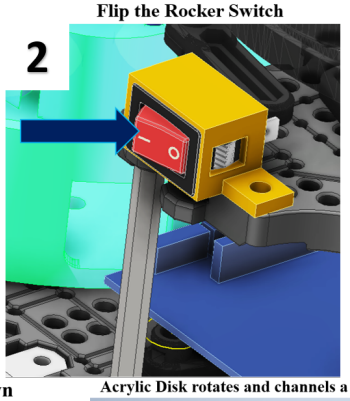


4.3 Firing Mechanism

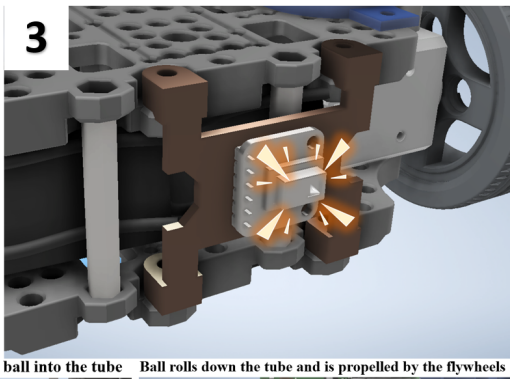
Ball Loading [Max Capacity: 5 balls]



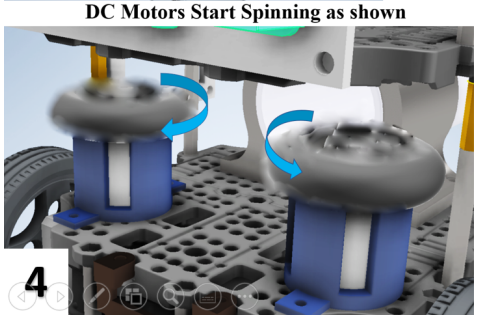
Flip the Rocker Switch



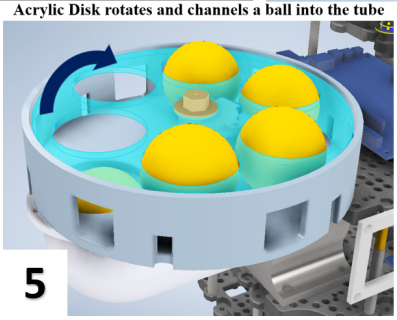
AMG8833 Detects Temperature > Threshold



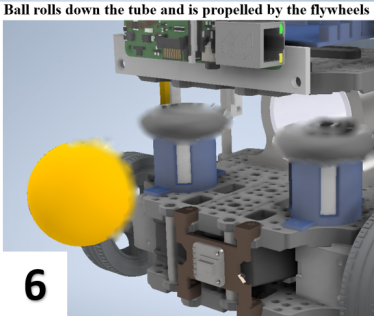
DC Motors Start Spinning as shown



Acrylic Disk rotates and channels a ball into the tube

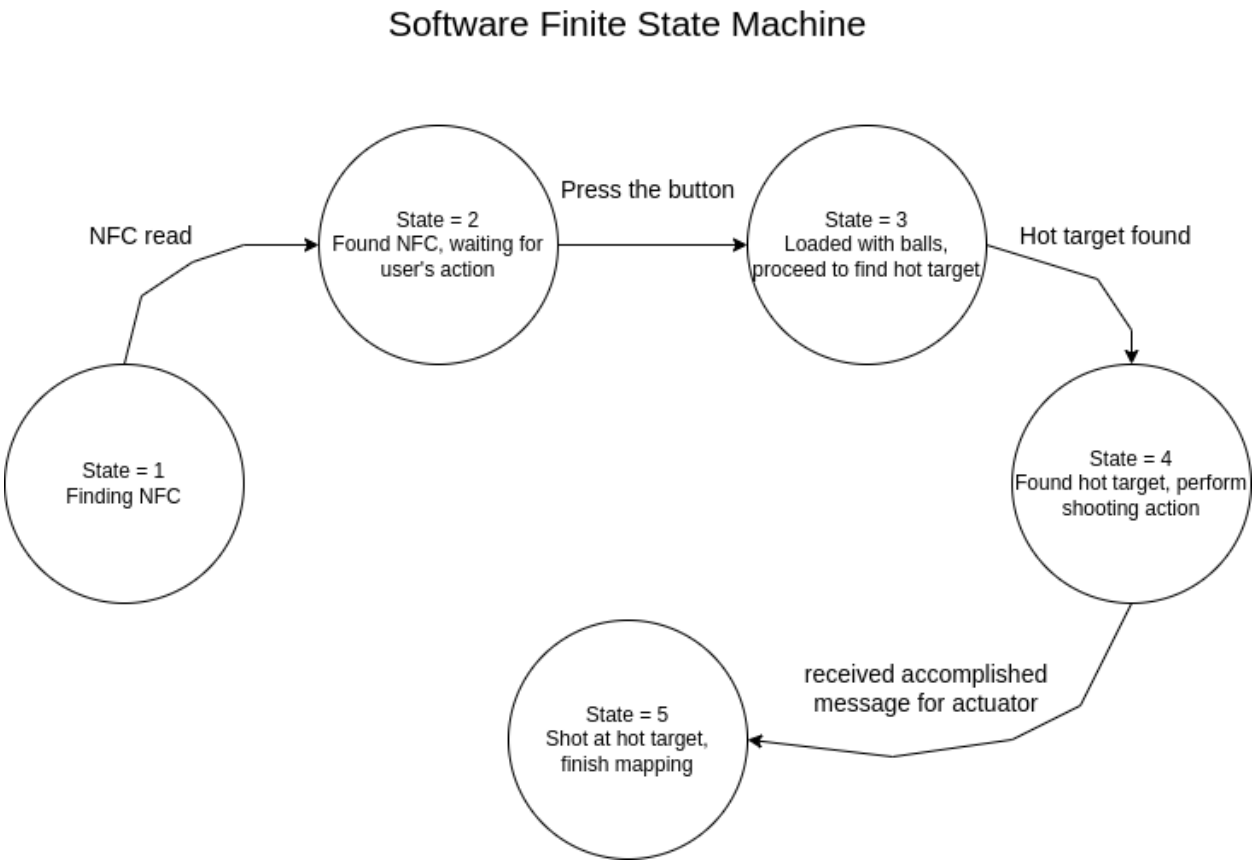


Ball rolls down the tube and is propelled by the flywheels



5.SOFTWARE

5.1 Block Diagram for Software Flow

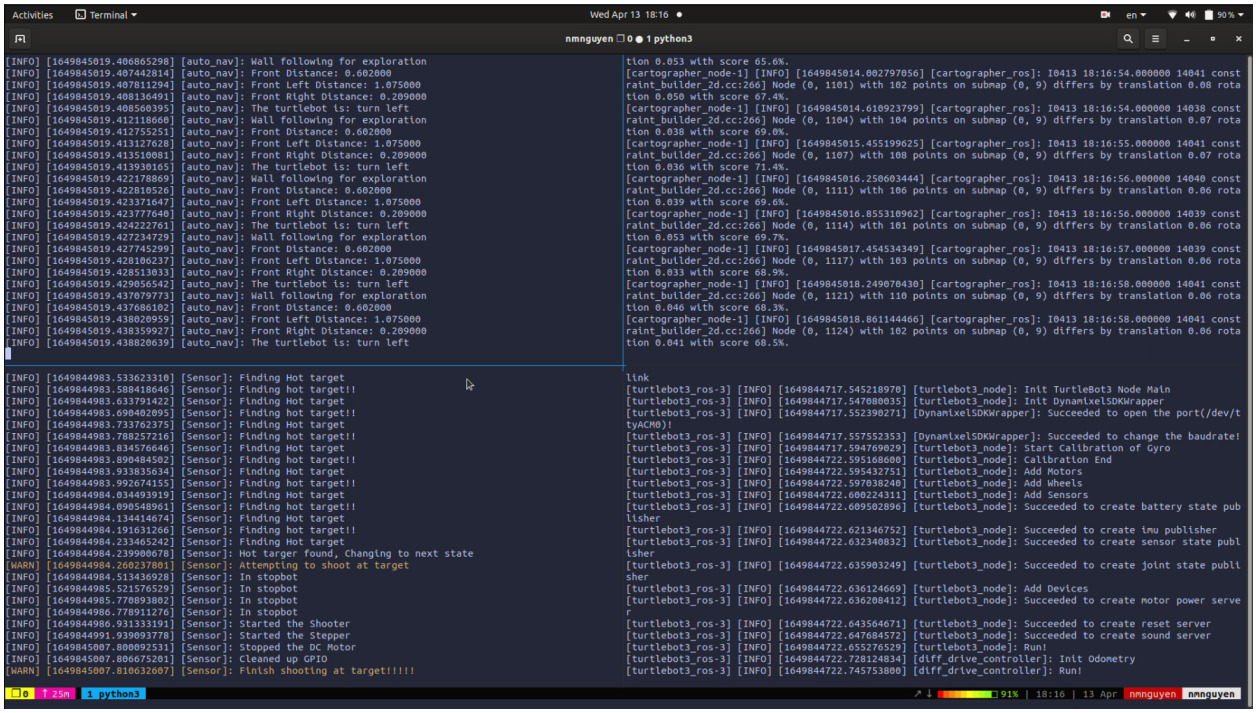


5.2 Running your code

Running Instructions:

- Clone the Github repo from https://github.com/nguyen2001ag/r2autonav and follow the instructions.
- SSH into the Raspberry Pi on the Turtlebot.
- Start the bring-up from the RPi on the TurtleBot: roslaunch turtlebot3_bringup turtlebot3_robot.launch
- Start SLAM from your laptop: ros2 launch turtlebot3_cartographer cartographer.launch.py
- Start the actuation code from the RPi on the TurtleBot: ros2 run <package_name> <entry_point_specified_in_the_setup.py>
- Start the wall following code from your laptop: ros2 run <package_name> self_exploration

Your running window should look like this for the ease of debugging. I am running SLAM and exploration node on my laptop, bring-up and actuation node on the Raspberry Pi.



Terminal windows on your laptop

6. BEFORE YOUR RUN (PRE-OPS CHECK)

6.1 Cable Check

For proper wiring management, the team has measured and crimped the respective cable with the adequate pinhead and length, and they are arranged in a manner such that it will neither be too tense nor too loose. Also, it shouldn't be blocking or impeding the actuation path (the ball-shooting mechanism. On top of that, the pinheads are to be double-checked to ensure electrically insulated by heat shrink for safety concerns and to ensure a tight connection with the different pins to allow more consistent power/signal transmission.

6.2 Software check

Inside your colcon workspace, build the package and observe for any errors.
On your computer, build your package using:

~/colcon_ws$ colcon build –packages-select auto_nav
~/colcon_ws$ source install/setup.bash

After you have built your package, you might want to test it in Simulation first. To use simulation:

Launch the Gazebo world:
ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py

Launch simulated SLAM:
ros2 launch turtlebot3_cartographer cartographer.launch.py use_sim_time:=True

Start the exploration code:
ros2 run auto_nav self_exploration

Observe the robot movement in the simulated environment and ensure it follows your own configuration.

6.3 Components Check

Components:
- 28BYJ-48 Stepper Motor: Motor should be able to do a full 360° rotation
- RS PRO Geared DC Motor: Motors should be able to spin at a relatively high velocity
- 8x8 AMG8833 Infrared Sensor Module: A 8x8 array of decimal values should appear depicting the estimated temperature of the relative regions the camera is pointed at.

We also provide a few scripts for you to test individual components:

Components	Command	Expected output
Thermal camera (AMG8833)	Ros2 run actuation test_amg	8x8 array of floats representing temperature are displayed at 1Hz
DC Motor (RS PRO)	Ros2 run actuation test_dc	The 2 DC motor is running
Stepper Motor (28BYJ-48)	Ros2 run actuation test_stepper	The loading part of the robot should be moving and the ping pong should come down the tube
NFC Reader (RC522)	Ros2 run actuation test_nfc	The robot should print out whether it has detected an NFC tag

6.4 Calibration of IR Camera

CALIBRATION OF 8x8 AMG8833 Infrared Sensor Module

For best accuracy before beginning operation, it is best to point the AMG8833 sensor facing the target and adjusting the distance between the target and the AMG8833 to have an accurate gauge of the detected temperature at different distances from the target.

Once this is done, under the **actuation.py** file, the user can then modify the 2 parameters:

1) **heat_threshold**, state 3 (self.fsm = 3) of the **Software Finite State Machine Diagram** will be achieved once the detected temperature goes above this value signifying that the target is detected.
2) **firing_threshold**, state 4 (self.fsm = 4) of the **Software Finite State Machine Diagram** will be achieved once the detected temperature goes above this value, sequence 4 to 6 will occur under **the Firing Mechanism (Section 4.3)**.

It is recommended to measure the temperature at a distance of 5cm to 15cm. This will allow the user to anticipate the approximate temperature of the target during operation and key the appropriate temperature threshold values into the **software**.

7. GETTING STARTED

7.1 Wiring

Make sure your wiring from the sensors follows the exact table below:

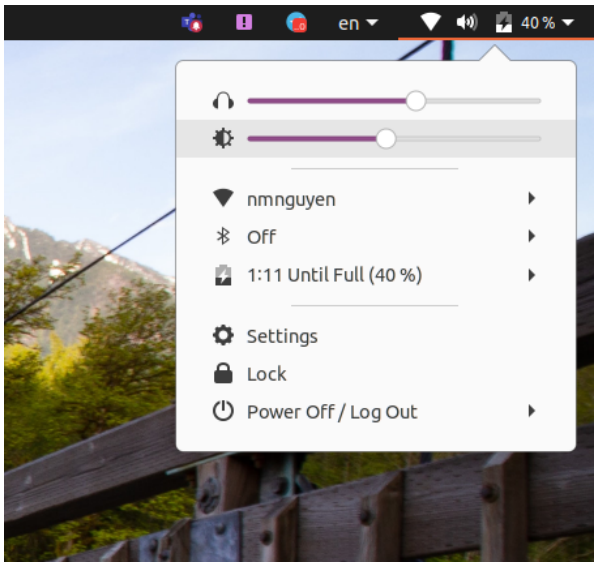
```
#####################################################
#                   PINOUT                          #
# GPIO 6: Button Enable                             #
# GPIO 12 (PWM1) : DC Motor1                        #
# GPIO 13 (PWM1) : DC Motor2                        #
# GPIO 22, 23, 24, 27: Stepper Motor Control        #
# GPIO 26: Button                                   #
# GPIO 2, 3: SDA, SCL (I2C) for IR camera           #
# GPIO 10, 09, 11, 08: MOSI, MISO, SCK, SS for NFC  #
# GPIO 25: RST for NFC RC522                        #
#                                                   #
#####################################################
```

7.2 Powering Up

Ensure the battery is connected to the OpenCR and flick the OpenCR on/off switch.

7.3 Connecting to the Turtlebot

Ensure that your laptop and the robot are on the same network as your laptop. To do this, I would suggest you create your own wifi hotspot and connect bot your Turtlebot and laptop to it.



Then you can connect to the Turtlebot using SSH:

Now you can follow Section 5.2 to perform your task.

8. PURCHASE LIST AND ITEMS REQUEST LIST

S/N	Item Description & Part/Model No.	Class of Item (Normal / Chemical / Biological / Hazardous*)	Unit price	Quantity	Sub Total	Distributor/Supplier/URL
1	IC Adapter LCQT-MSOP10	Normal	$ 5.62	1	$ 5.62	https://sg.element14.com/aries/lcqt-msop10/ic-adaptor-10msop-to-dip-2-54mm/dp/2476034
2	IC Adapter LCQT-SOIC8-8	Normal	$ 5.32	2	$ 10.64	https://sg.element14.com/aries/lcqt-soic8-8/ic-adaptor-8-soic-to-dip-2-54mm/dp/2476033
3	AMG8833 IR Sensor module	Normal	$ 31.88	1	$ 31.88	https://shopee.sg/AMG8833-IR-8*8-MLX90640-32*24-Camera-Module-Thermal-Imager-Array-Temperature-Sensor-8x8-32x24-Infrared-MLX90640BAA-MLX90640BAB-i.56539845.4997730286?sp_atk=c0d622d8-37fc-4a0a-
4	RS PRO Geared DC Motor, 1.6 W, 1.5 → 3 V, 20 gcm, 7800 rpm, 2mm Shaft Diameter	Normal	$ 5.34	2	$ 10.68	https://sg.rs-online.com/web/p/dc-motors/2389709
5	TPC8125 MOSFET-P	Normal	$ 2.02	5	$ 10.10	https://sg.rs-online.com/web/p/mosfets/7965115
6	LTC4416 Power Path Controller IC	Normal	$ 9.66	1	$ 9.66	https://sg.rs-online.com/web/p/gate-drivers/7619581
7	MP1584EN	Normal	$ 1.36	3	$ 4.08	https://shopee.sg/MP1584EN-Ultra-small-Size-DC-DC-Step-down-Power-Supply-Module-3A-Adjustable-Step-down-Module-Super-LM2596-i.161750523.3508610430?g
	Delivery fee from RS Component		$ 8.56		$ 8.56	
				Grand Total	$ 91.22	

APPENDIX

Github Repo: https://github.com/nguyen2001ag/r2autonav

Published by: EG2310 Group 9
Address: Engineering Design & Innovation Centre
Block E2A #04-05
5 Engineering Drive 2
Singapore 117579