# Engineering Notebook

TEAM 186

Project 4: Autonomous Product Retriever

February 11th, 2023

Team members:

Ben Smith | smith9bi

Luke Raters | raterslm

Taha Momayiz | momayita

Yatharth Bajaj | bajajyh

# Table of Contents

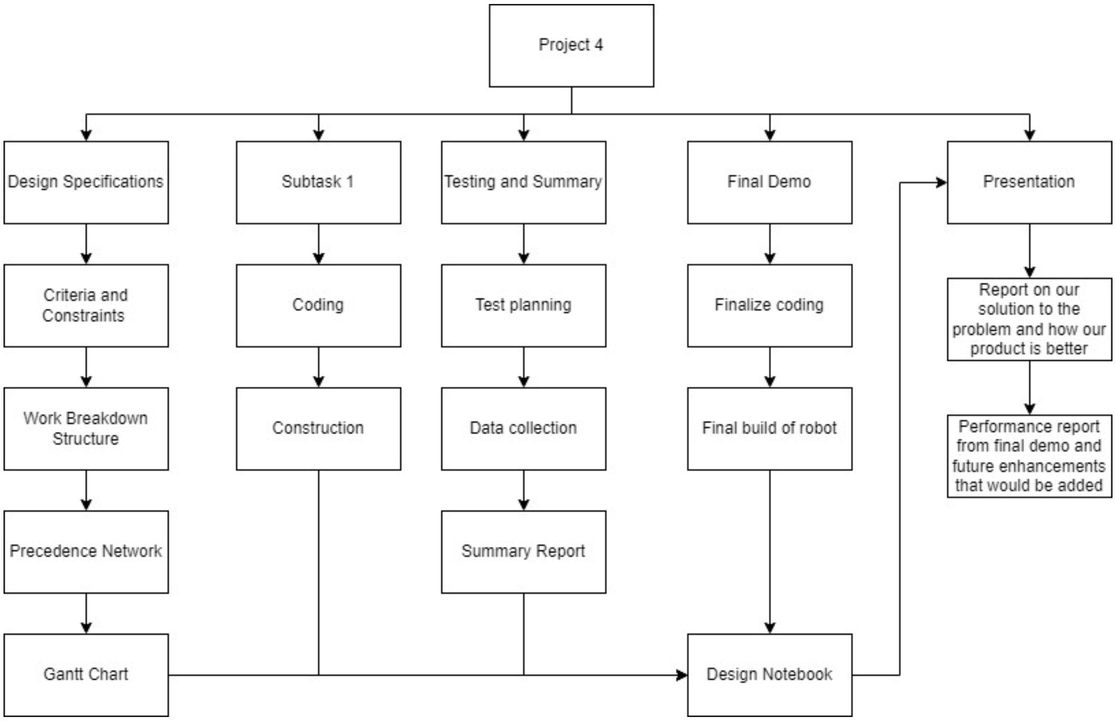
# Meeting 1: 02/11/2023

6:00pm to 8:00pm

All the members were present. We completed the design specifications, made a work breakdown structure, precedence network, and Gantt chart for the project, identified our stakeholders and criteria and constraints, and began discussing the first subtask.

-BS

Work Breakdown Structure:



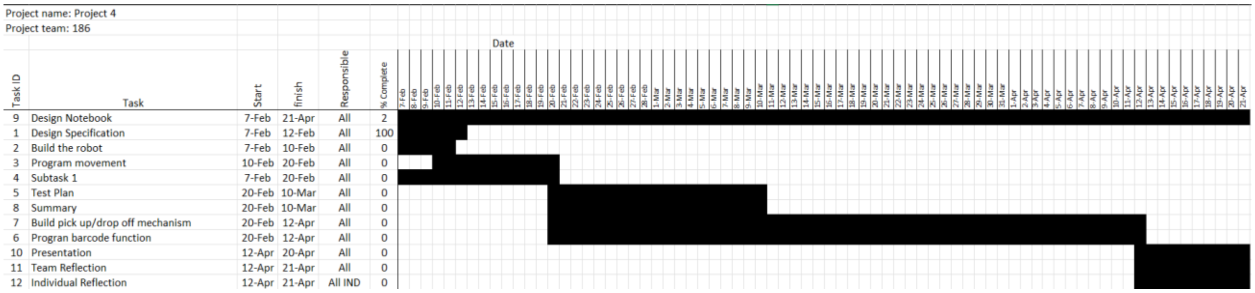
Precedence Network:

| Task # | Description | Precedence | Duration |
|---|---|---|---|
| **1** | Design Specifications | None | 1 Week |
| **2** | Build Initial Design | None | 1 Week |
| **3** | Program Movement | None | 1 Week |
| **4** | Finalize for Subtask 1 | 2, 3 | 1 Week |
| **5** | Test Plan | 4 | 2 Weeks |
| **6** | Mid-Project Summary | 4 | 2 Weeks |
| **7** | Pickup/Drop Mechanisms | 4 | 6 Weeks |
| **8** | Barcode Scanning Functions | 4 | 6 Weeks |
| **9** | Design Notebook | None | Continuous |
| **10** | Finalize for Status Update 1 | 7, 8 | 1 Week |
| **11** | Finalize for Final Demonstration | 10 | 1 Week |
| **12** | Presentation | 11 | 1 Week |
| **13** | Team Reflection | 11 | 0.5 Weeks |
| **14** | Individual Reflection | 11 | 0.5 Weeks |

The critical path, which is fairly linear, is highlighted in yellow.

-LR

The initial Gantt Chart:



Project name: Project 4
Project team: 186

| Task ID | Task | Start | finish | Responsible | % Complete |
|---|---|---|---|---|---|
| 9 | Design Notebook | 7-Feb | 21-Apr | All | 2 |
| 1 | Design Specification | 7-Feb | 12-Feb | All | 100 |
| 2 | Build the robot | 7-Feb | 10-Feb | All | 0 |
| 3 | Program movement | 10-Feb | 20-Feb | All | 0 |
| 4 | Subtask 1 | 7-Feb | 20-Feb | All | 0 |
| 5 | Test Plan | 20-Feb | 10-Mar | All | 0 |
| 8 | Summary | 20-Feb | 10-Mar | All | 0 |
| 7 | Build pick up/drop off mechanism | 20-Feb | 12-Apr | All | 0 |
| 6 | Progran barcode function | 20-Feb | 12-Apr | All | 0 |
| 10 | Presentation | 12-Apr | 20-Apr | All | 0 |
| 11 | Team Reflection | 12-Apr | 21-Apr | All | 0 |
| 12 | Individual Reflection | 12-Apr | 21-Apr | All IND | 0 |

-BS

The stakeholders involved in this project are:

1. **Buy More**: They are the primary stakeholders.
2. **The building team**: Who will submit the proposal for the APR system.
3. **Customers**: Who will benefit from the increased efficiency and speed due to the APR.

| **Criteria or Constraint:** | **Specification:** |
|---|---|
| The robot needs to be able to move forward and back for a specified distance. | It should operate within a +- 20 cm corridor. |
| The robot needs to be able to turn 180 degrees and then come back to the initial position. | It should do so with a <1 degree error to not misalign the robot for the movement. |
| The robot needs to scan the barcode to identify the box. | It should identify the box correctly 100% of the time before advancing. |
| The robot needs to pick up the box and drop it at the desired location. | It should drop the box within +-10cm of the desired location. |
| The robot needs to complete the tasks in a timely manner. | Each task should take no longer than 2 minutes. |
| The robot must avoid obstacles. | The robot should stop 100% of time if there is something blocking its path. |
| The robot must navigate to the correct location in the most efficient way possible. | It must minimize the energy used for each task. |

-YB

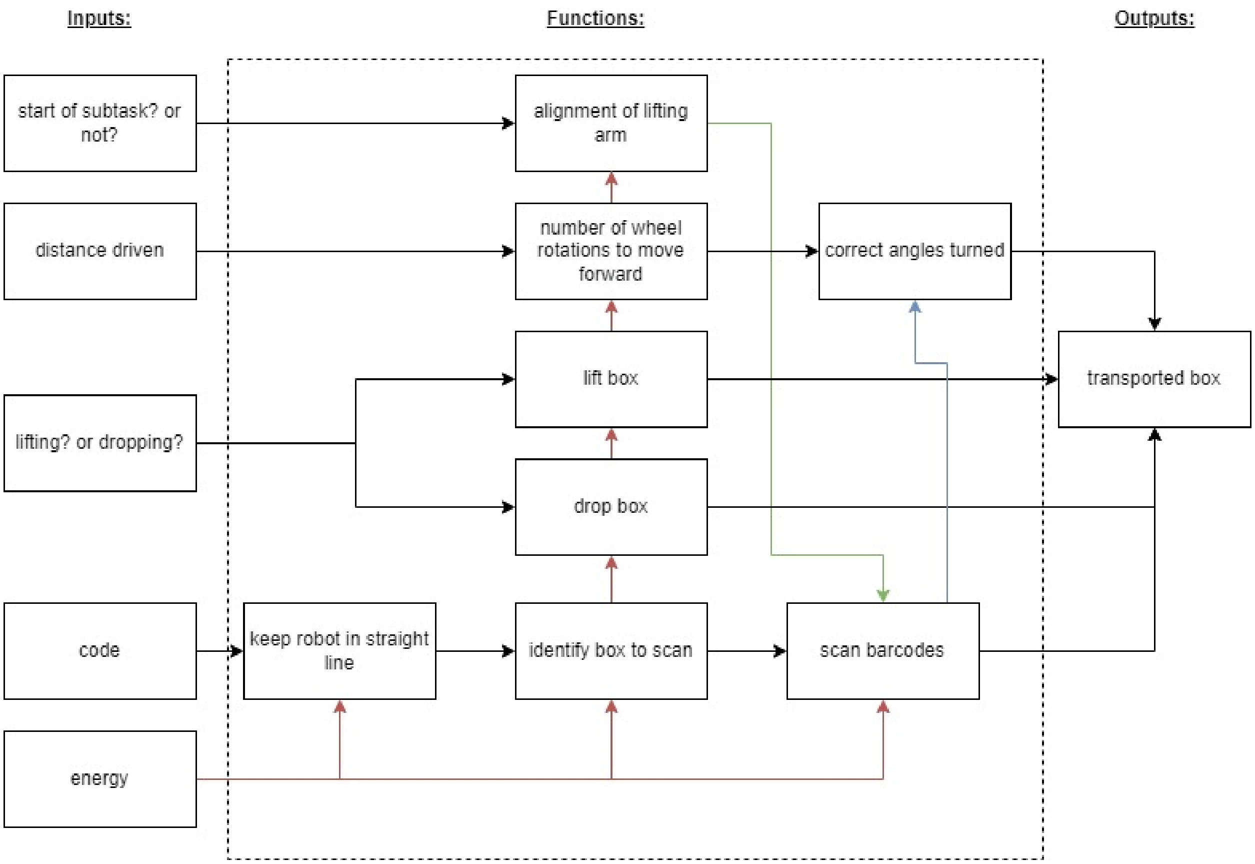
## Meeting 2: 02/14/2023

7:00pm to 8:30pm

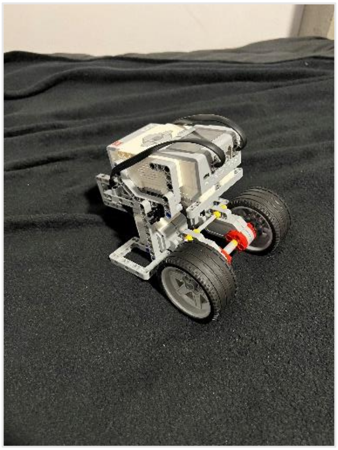
All the members were present. The initial robot design was brainstormed and built. Also, we set up python and GitHub so we could all work on the code, which we did starting during the meeting. Ben was assigned to work on the physical robot outside of the meeting, and Taha was assigned to work on the code.

| Criteria | Weight | 1 Wheel + Pivot | Legs | 4 Wheels | 2 Wheels + Pivot | 2 Wheels |
|---|---|---|---|---|---|---|
| Simplicity to Construct | 3 | 3 | 1 | 5 | 4 | 3 |
| Effectiveness | 5 | 1 | 1 | 5 | 5 | 3 |
| Efficiency | 4 | 3 | 1 | 5 | 4 | 3 |
| Difficultly to Code | 4 | 1 | 1 | 3 | 5 | 5 |
| Weighted Score | [redacted] | 32 | 16 | 72 | 73 | 56 |

We created a functional block diagram to help determine exact what was needed for the code:

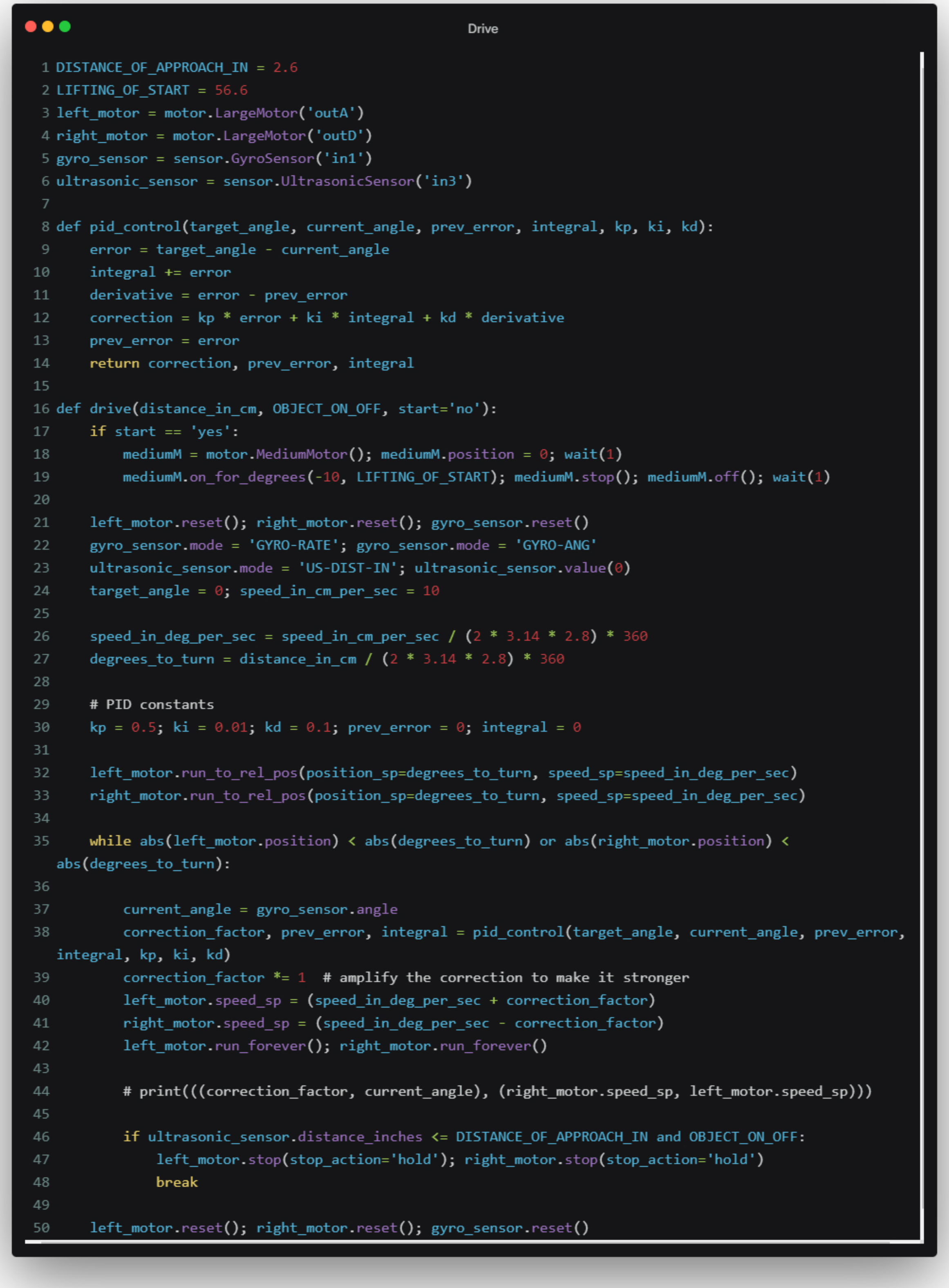


This design was started at the meeting and finished outside by Ben. The design is simple, with two wheels in the front and a ball bearing in the back, this allows the robot to move effectively and turn very well. Taha also finished the preliminary coding after the meeting.



-BS

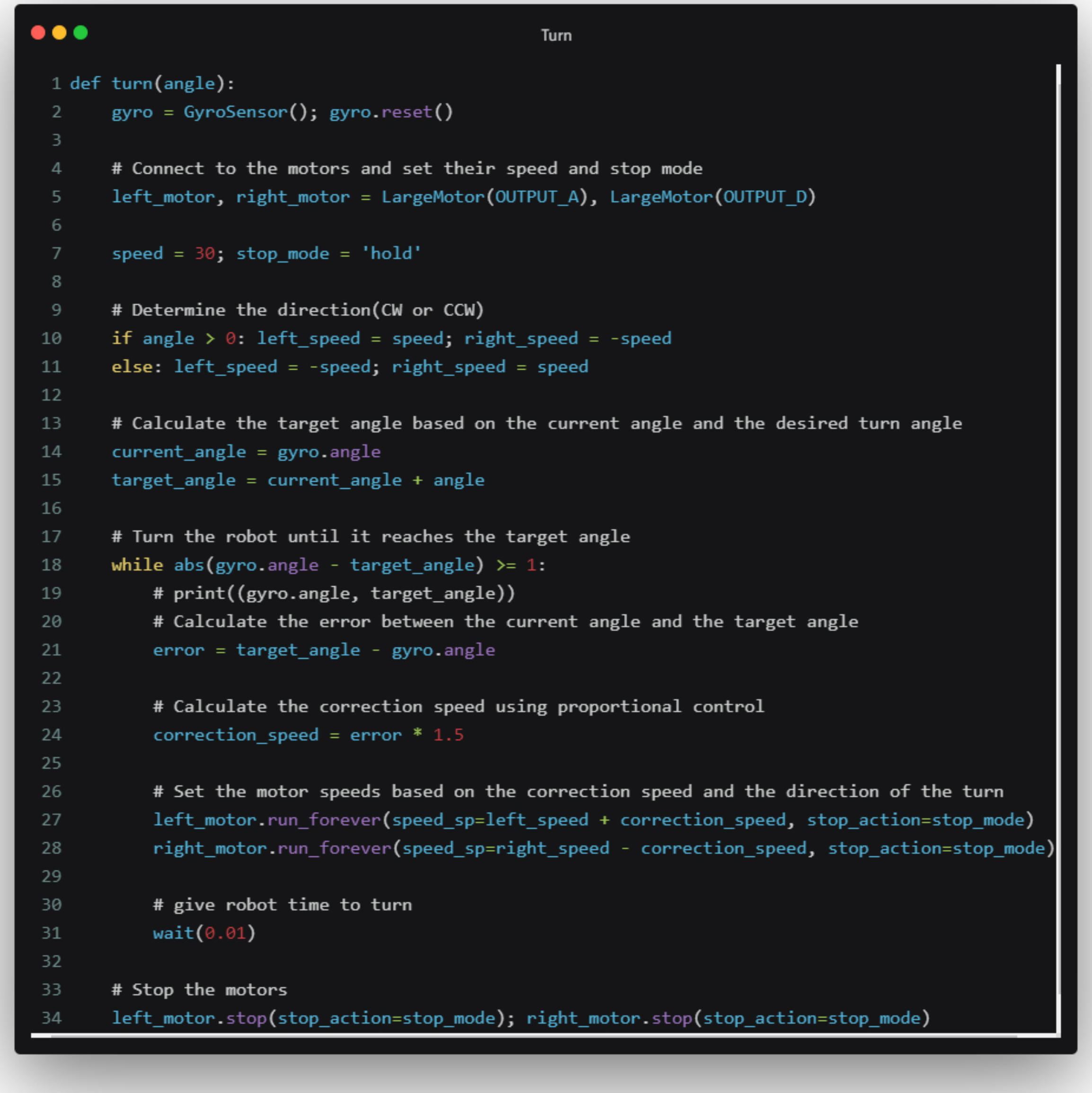
The project was assigned then to Taha to work on the code that was used for driving:



```
DISTANCE_OF_APPROACH_IN = 2.6
LIFTING_OF_START = 56.6
left_motor = motor.LargeMotor('outA')
right_motor = motor.LargeMotor('outD')
gyro_sensor = sensor.GyroSensor('in1')
ultrasonic_sensor = sensor.UltrasonicSensor('in3')

def pid_control(target_angle, current_angle, prev_error, integral, kp, ki, kd):
    error = target_angle - current_angle
    integral += error
    derivative = error - prev_error
    correction = kp * error + ki * integral + kd * derivative
    prev_error = error
    return correction, prev_error, integral

def drive(distance_in_cm, OBJECT_ON_OFF, start='no'):
    if start == 'yes':
        mediumM = motor.MediumMotor(); mediumM.position = 0; wait(1)
        mediumM.on_for_degrees(-10, LIFTING_OF_START); mediumM.stop(); mediumM.off(); wait(1)

    left_motor.reset(); right_motor.reset(); gyro_sensor.reset()
    gyro_sensor.mode = 'GYRO-RATE'; gyro_sensor.mode = 'GYRO-ANG'
    ultrasonic_sensor.mode = 'US-DIST-IN'; ultrasonic_sensor.value(0)
    target_angle = 0; speed_in_cm_per_sec = 10

    speed_in_deg_per_sec = speed_in_cm_per_sec / (2 * 3.14 * 2.8) * 360
    degrees_to_turn = distance_in_cm / (2 * 3.14 * 2.8) * 360

    # PID constants
    kp = 0.5; ki = 0.01; kd = 0.1; prev_error = 0; integral = 0

    left_motor.run_to_rel_pos(position_sp=degrees_to_turn, speed_sp=speed_in_deg_per_sec)
    right_motor.run_to_rel_pos(position_sp=degrees_to_turn, speed_sp=speed_in_deg_per_sec)

    while abs(left_motor.position) < abs(degrees_to_turn) or abs(right_motor.position) < abs(degrees_to_turn):

        current_angle = gyro_sensor.angle
        correction_factor, prev_error, integral = pid_control(target_angle, current_angle, prev_error, integral, kp, ki, kd)
        correction_factor *= 1  # amplify the correction to make it stronger
        left_motor.speed_sp = (speed_in_deg_per_sec + correction_factor)
        right_motor.speed_sp = (speed_in_deg_per_sec - correction_factor)
        left_motor.run_forever(); right_motor.run_forever()

        # print(((correction_factor, current_angle), (right_motor.speed_sp, left_motor.speed_sp)))

        if ultrasonic_sensor.distance_inches <= DISTANCE_OF_APPROACH_IN and OBJECT_ON_OFF:
            left_motor.stop(stop_action='hold'); right_motor.stop(stop_action='hold')
            break

    left_motor.reset(); right_motor.reset(); gyro_sensor.reset()
```

This is the PID control algorithm that allow the robot to freely calculate the error that was made while driving using previous error, derivation and integral concepts to create a much more effective and reliable correction speed which the drive function uses to fix its path each time the gyroscope detect an angle variation.

The code that was used for turning:



```
def turn(angle):
    gyro = GyroSensor(); gyro.reset()

    # Connect to the motors and set their speed and stop mode
    left_motor, right_motor = LargeMotor(OUTPUT_A), LargeMotor(OUTPUT_D)

    speed = 30; stop_mode = 'hold'

    # Determine the direction(CW or CCW)
    if angle > 0: left_speed = speed; right_speed = -speed
    else: left_speed = -speed; right_speed = speed

    # Calculate the target angle based on the current angle and the desired turn angle
    current_angle = gyro.angle
    target_angle = current_angle + angle

    # Turn the robot until it reaches the target angle
    while abs(gyro.angle - target_angle) >= 1:
        # print((gyro.angle, target_angle))
        # Calculate the error between the current angle and the target angle
        error = target_angle - gyro.angle

        # Calculate the correction speed using proportional control
        correction_speed = error * 1.5

        # Set the motor speeds based on the correction speed and the direction of the turn
        left_motor.run_forever(speed_sp=left_speed + correction_speed, stop_action=stop_mode)
        right_motor.run_forever(speed_sp=right_speed - correction_speed, stop_action=stop_mode)

        # give robot time to turn
        wait(0.01)

    # Stop the motors
    left_motor.stop(stop_action=stop_mode); right_motor.stop(stop_action=stop_mode)
```

This function takes the desired angle and uses gyroscope to calculate the current angle, it then adjusts the speed of the motors to proportionally turn the motors to acquire the desire angle and the overcome any overshooting errors.

These functions were upgraded throughout the semester and weren’t primarily used in subtasks 1A and 1B, they were adapted after the failure of our old precision algorithms.

-TM

## Meeting 3: 02/18/2023

6:00pm to 8:30pm

All the members were present. The final design was completed, and we tested the robot so we could make an accurate error prediction. Our test values were:

| **Run** | **Laps** | **Lap Length** | **Total distance** | **X-error** | **Y-error** |
|---|---|---|---|---|---|
| **1A** | 3 | 200cm | 1200cm | 7cm | 10cm |
| **2A** | 3 | 100cm | 600cm | 2.8cm | 2cm |
| **1B** | 2 | 100cm | 400cm | 6cm | 0cm |
| **2B** | 2 | 100cm | 400cm | 0cm | 0cm |
| **3B** | 2 | 100cm | 400cm | 3cm | 1cm |
| **4B** | 2 | 100cm | 400cm | 1cm | 1cm |
| **3A** | 3 | 100cm | 600cm | 4cm | 3.5cm |
| **4A** | 3 | 100cm | 600cm | 0cm | 1cm |
| **Average error per cm** | [redacted] | [redacted] | [redacted] | A - 0.0046cm<br>B - 0.0063cm | A - 0.0052cm<br>B - 0.0013cm |

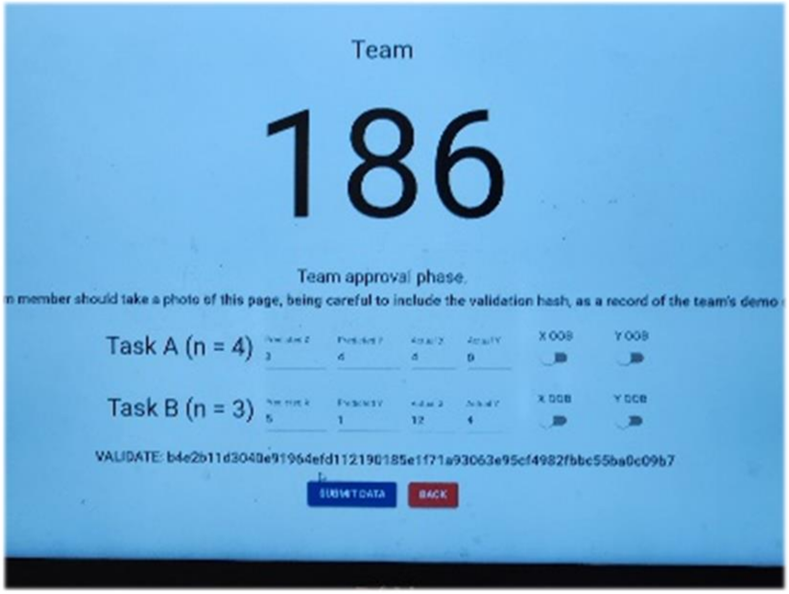
This data provided us with a method to predict the error for subtask 1. The method is as follows:

Multiply the total distance the robot will travel for each subtask by the average x and y error values and that will give us a good guess for the error in the subtask 1 demonstration.

-LR

## Sub-Task Demo: 02/20/2023

4:30pm to 5:30pm



Team

186

Team approval phase.

n member should take a photo of this page, being careful to include the validation hash, as a record of the team's demo

| | Nominal X | Nominal Y | Actual X | Actual Y | X OOB | Y OOB |
|---|---|---|---|---|---|---|
| Task A (n = 4) | 3 | 6 | 6 | 9 | | |
| Task B (n = 3) | 5 | 1 | 12 | 4 | | |

VALIDATE: b4e2b11d3040e91964efd112190185e1f71a93063e95cf4982fbbc55ba0c09b7

Subtask 1 demonstration day. All members of the team were present. When we first tested the robot a few minutes before our demonstration, it was not going straight. Eventually it started working, and we did decently good. One part that did cause more errors than expected was the paper clip and metal ball roller. The roller was very tight, and the paper clip was dragging on the ground a little during the demonstration, so it did not turn as well as it could have. We got a new metal ball roller after and will remember to put the paper clip in a better spot next time, so it goes more like how we expected.

-LR

## Meeting 4: 02/28/2023

7:00pm to 9:30pm

All the members were present. We did all the straight-line tests for the test plan, and properly collected the data in the spreadsheet. Ben was assigned to work on the lifting mechanism outside of the meeting. Luke was assigned to do some research for the CCCD and plan that event out for the team.

-LR

Future test plan-

We feel that the most important aspects of the bot to test are the barcode readings, pickup and move with the box, drop off, and movement around the warehouse.

To test the barcode reading we would place the bot in front of a paper with all the different barcode options printed on it. The bot would be set to read the code and we will record if it was correct, we would do this at least twice for each barcode combination.

To test the pickup and move we would start the robot in front of a box, instruct it to move forward that distance and lift the arm. Then we will tell it to move forward a certain distance to test the stability with the box.

To test the movement around the warehouse we would construct a smaller version of the final demo warehouse with similar dimensions but not necessarily the individual boxes. We would use this to test how well the code works to navigate around the warehouse and to make sure that the x and y error are not too great to cause the bot to hit a shelf.

To test the final sequence, we would set the bot to start at a “home area” similar to the final demo, it will then be given the location of the box and fulfillment area. This will allow us to test the location accuracy, barcode feature, and drop of functionality. This test would be done very close to the final demo and would be used to verify how well the bot works before we put together a final code.

-BS

## 3/04/2023

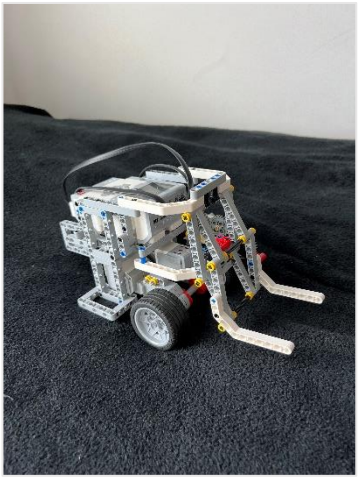
7:00pm-10:00pm

Ben was present. I started by brainstorming design ideas. I came up with a few, but the best ones were a vertical lift where the medium motor would be connected to a gear that would lift a arm straight upwards. Attaching an arm directly to the medium motor. The other idea was to connect the medium motor to gears that ran a lever arm to lift a “parallel lift”. I noticed a lot of other groups doing the vertical lift, and after discussing with Luke about his similar design last semester, we decided that the parallel lift would be both effective and unique. I started working on a protype lifting mechanism.

Decision matrix:

| | **Weight** | **Vertical lift** | **Direct connection** | **Parallel Lift** |
|---|---|---|---|---|
| **Effectiveness** | 5 | 4 | 3 | 4 |
| **Simplicity** | 5 | 4 | 5 | 4 |
| **Efficiency (with motor)** | 4 | 4 | 3 | 4 |
| **Uniqueness** | 2 | 1 | 3 | 5 |
| **Total** | | 58 | 58 | 66 |

This is the final product:



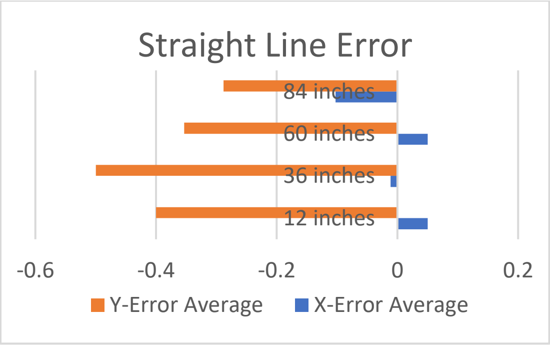
-BS

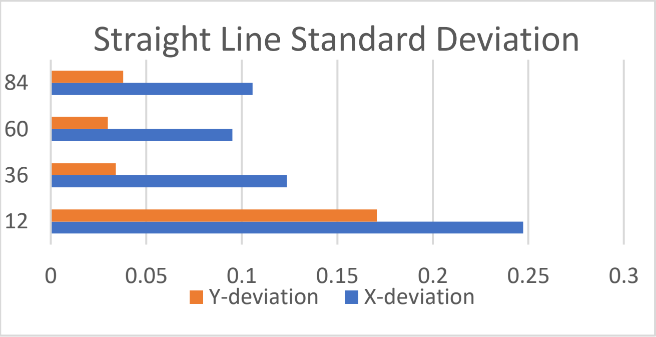
## Meeting 5: 03/09/2023

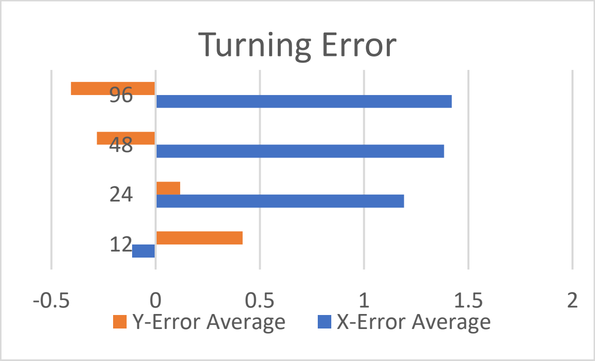
10:30am to 12:30pm

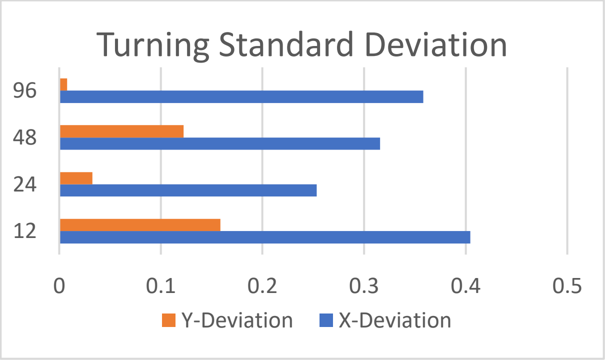
All members present. We finished the testing for the mid project review. As a result, we calculated a coefficient that will give us the desired distance and angle, we will be adding this to the code to improve our accuracy with locomotion. Luke was assigned to set up the testing for the future functions such as carrying the box, scanning the barcode, lifting the box, and avoiding objects.

Here are the locomotion test results:









For straight line motion, we found these coefficients:

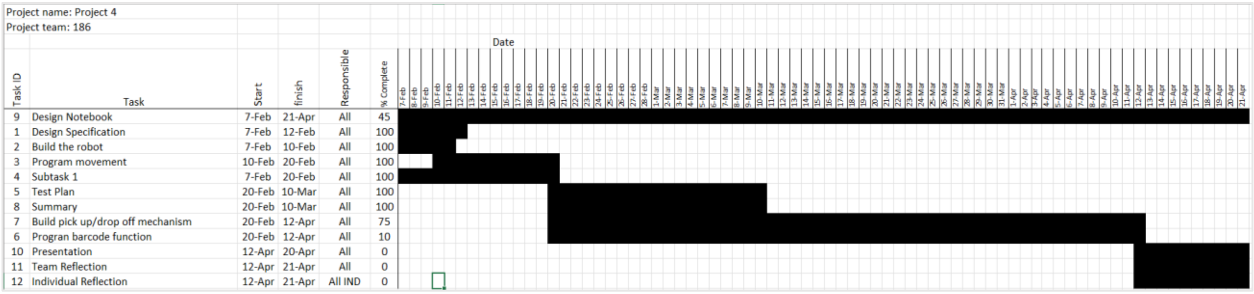
| Weighted Average X error (mm per inch) |
|---|
| -0.0281225 |
| |
| Weighted Average Y error (mm per inch) |
| -0.3552025 |

And for turning, the coefficients we found were as follows:

| | |
|---|---|
| Weighted Average X error (mm per inch) | 1.2133052 |
| Weighted Average Y error (mm per inch) | -0.20390286 |
| Weighted Average Angle error (mm per inch) | 1.452851948 |

-LR

This is also an updated Gantt chart:


Project name: Project 4
Project team: 186

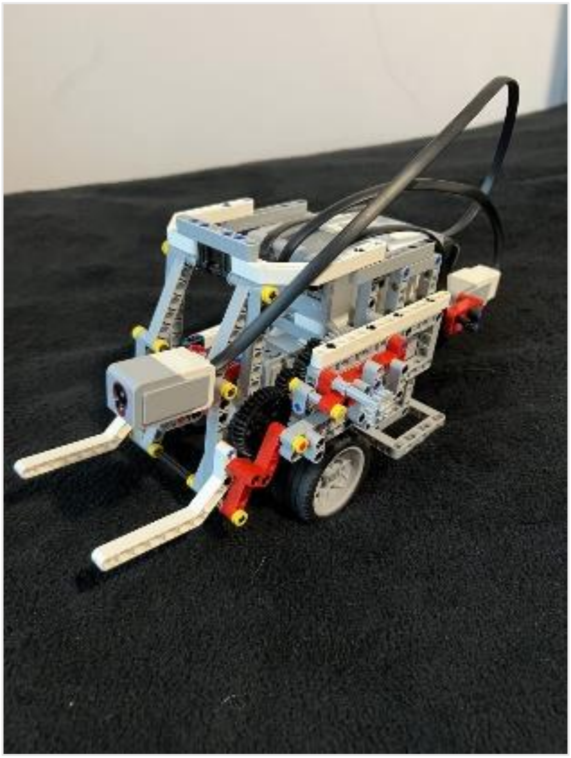
| Task ID | Task | Start | finish | Responsible | % Complete |
|---|---|---|---|---|---|
| 9 | Design Notebook | 7-Feb | 21-Apr | All | 45 |
| 1 | Design Specification | 7-Feb | 12-Feb | All | 100 |
| 2 | Build the robot | 7-Feb | 10-Feb | All | 100 |
| 3 | Program movement | 10-Feb | 20-Feb | All | 100 |
| 4 | Subtask 1 | 7-Feb | 20-Feb | All | 100 |
| 5 | Test Plan | 20-Feb | 10-Mar | All | 100 |
| 8 | Summary | 20-Feb | 10-Mar | All | 100 |
| 7 | Build pick up/drop off mechanism | 20-Feb | 12-Apr | All | 75 |
| 6 | Progran barcode function | 20-Feb | 12-Apr | All | 10 |
| 10 | Presentation | 12-Apr | 20-Apr | All | 0 |
| 11 | Team Reflection | 12-Apr | 21-Apr | All | 0 |
| 12 | Individual Reflection | 12-Apr | 21-Apr | All IND | 0 |

-BS

## Meeting 6: 03/10/2023

1:00pm to 2:00pm

Ben was present. I improved the lifting arm mechanism to be much stronger and more structured. This is the result:

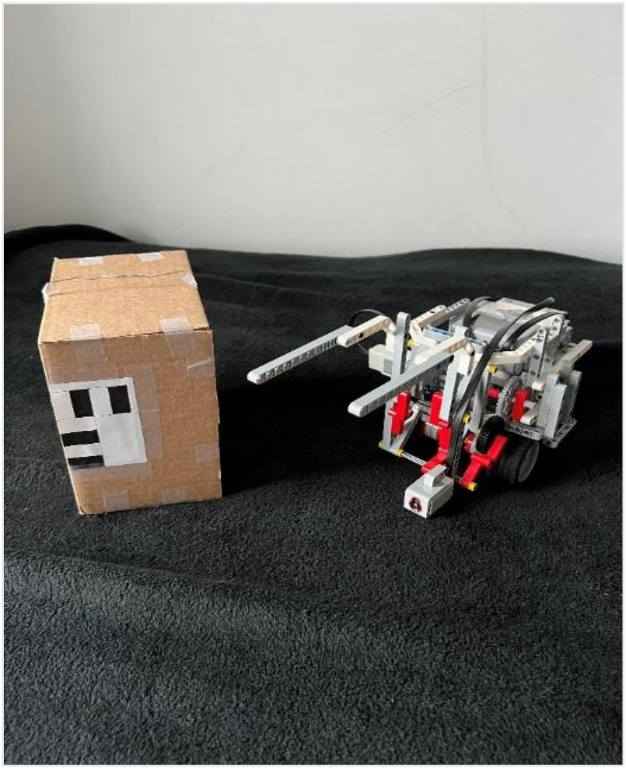


-BS

## Meeting 7: 03/21/2023

8:00pm to 10:00pm

Ben was present. constructed a box that is the same as the boxes to be used in the final demo. We will use it to test the lifting and barcode reading function of the robot. I also made improvements to the lifting arm. I moved the motor to the front of the robot to make a better connection with the lifting arm, this allowed for a much simpler gear assembly. I also positioned the lifting arms so that they can lift the box from the handle, and moved the camera so it can work effectively. This is the result:



-BS

## Meeting 8: 03/25/2023

6:30pm to 9:00pm

All the members were present. We watched the movie “Parasite” and discussed the movie after for the CCCD. We wrote our discussion notes in a PowerPoint to cover for the video assignment.

Next meeting, we plan to record our video for the CCCD assignment.

-LR

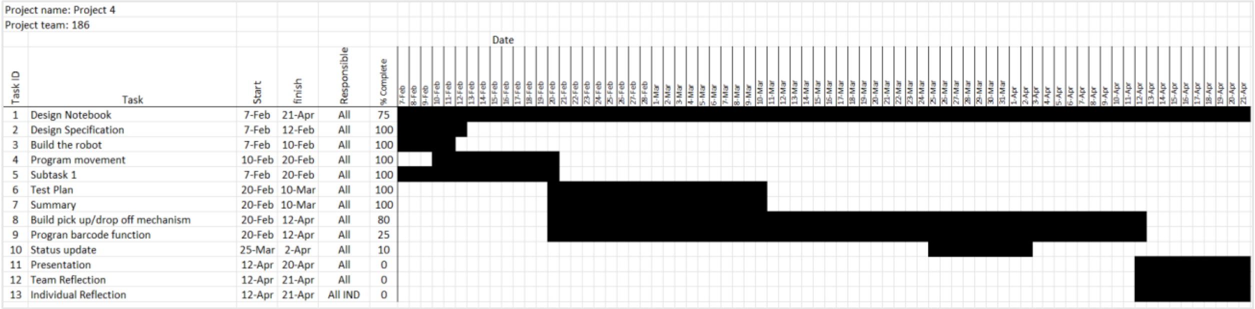
# Meeting 9: 03/28/2023

6:30pm to 8:00pm

All the members were present. We met to make the video for the CCCD assignment and discuss what needed to be worked on for the robot for the status update. For the CCCD video, we all joined a Zoom call and went over the PowerPoint we created as it was screenshared.

Taha stated he would do some work on the robot in preparation for the status update.

-LR



Project name: Project 4

Project team: 186

Date

| Task ID | Task | Start | finish | Responsible | % Complete |
|---|---|---|---|---|---|
| 1 | Design Notebook | 7-Feb | 21-Apr | All | 75 |
| 2 | Design Specification | 7-Feb | 12-Feb | All | 100 |
| 3 | Build the robot | 7-Feb | 10-Feb | All | 100 |
| 4 | Program movement | 10-Feb | 20-Feb | All | 100 |
| 5 | Subtask 1 | 7-Feb | 20-Feb | All | 100 |
| 6 | Test Plan | 20-Feb | 10-Mar | All | 100 |
| 7 | Summary | 20-Feb | 10-Mar | All | 100 |
| 8 | Build pick up/drop off mechanism | 20-Feb | 12-Apr | All | 80 |
| 9 | Progran barcode function | 20-Feb | 12-Apr | All | 25 |
| 10 | Status update | 25-Mar | 2-Apr | All | 10 |
| 11 | Presentation | 12-Apr | 20-Apr | All | 0 |
| 12 | Team Reflection | 12-Apr | 21-Apr | All | 0 |
| 13 | Individual Reflection | 12-Apr | 21-Apr | All IND | 0 |

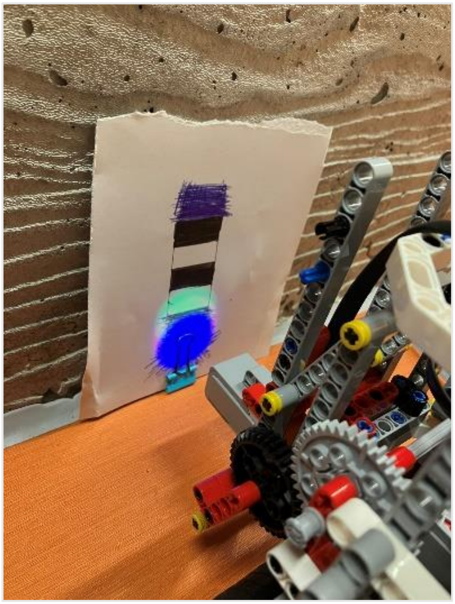
## Meeting 10: 03/31/2023

10:00pm to 4:00am

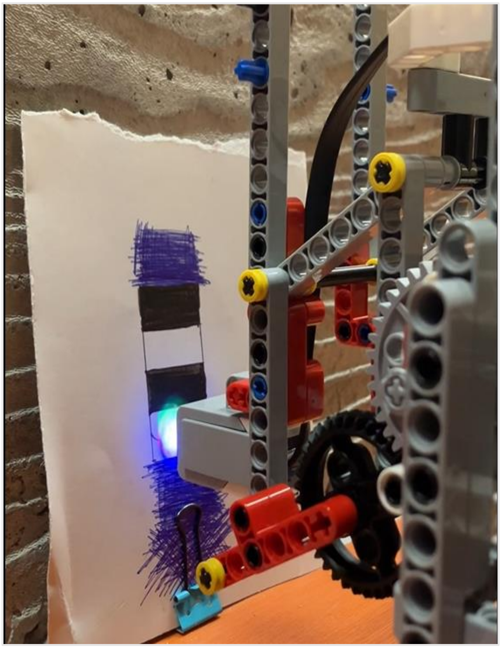
Taha and Yatharth were present. Yatharth was documenting the robot's progress.

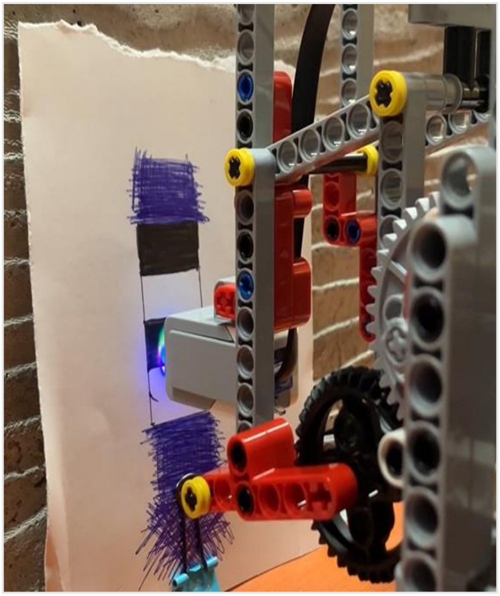
Taha worked on the following mechanisms:

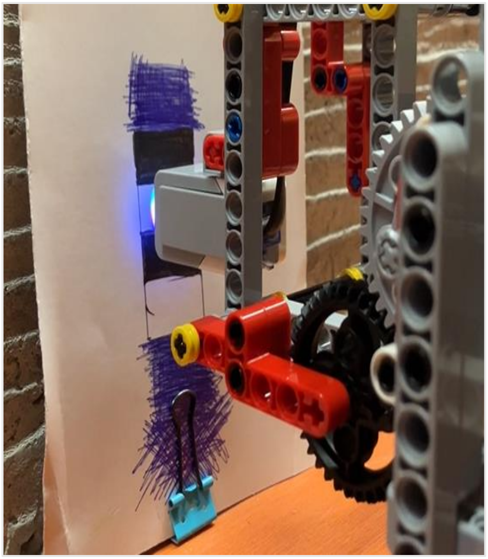
**Scanner mechanism**: The barcode scanner mechanism is a device that uses a color sensor and a motor to scan a barcode consisting of four horizontal colors spaced 0.5 inches apart. The motor moves the sensor along the barcode, stopping at each color to scan its color and record it as a 0 or 1. If the scanned barcode matches one of the four predefined combinations, then checks for the right type, if correct then the mechanism returns a "True" value and performs further actions to lift and go home. If no match is found, the mechanism reverses direction and attempts to scan the barcode again. If a match is still not found, the mechanism repeats the scanning process 4 times.

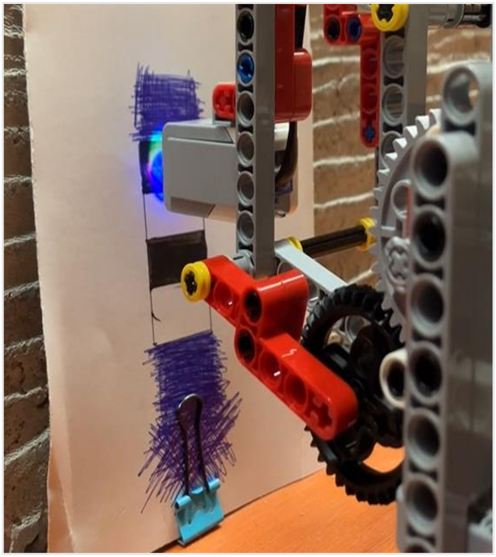


Here is the process:



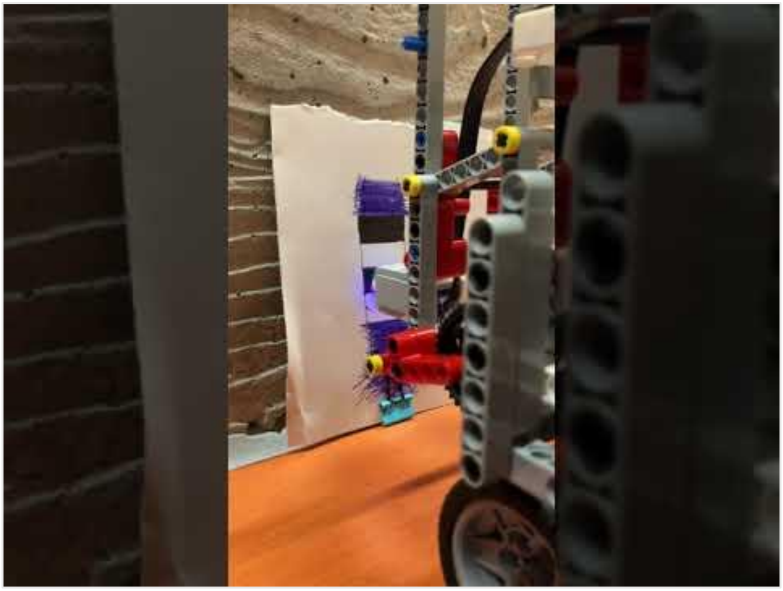




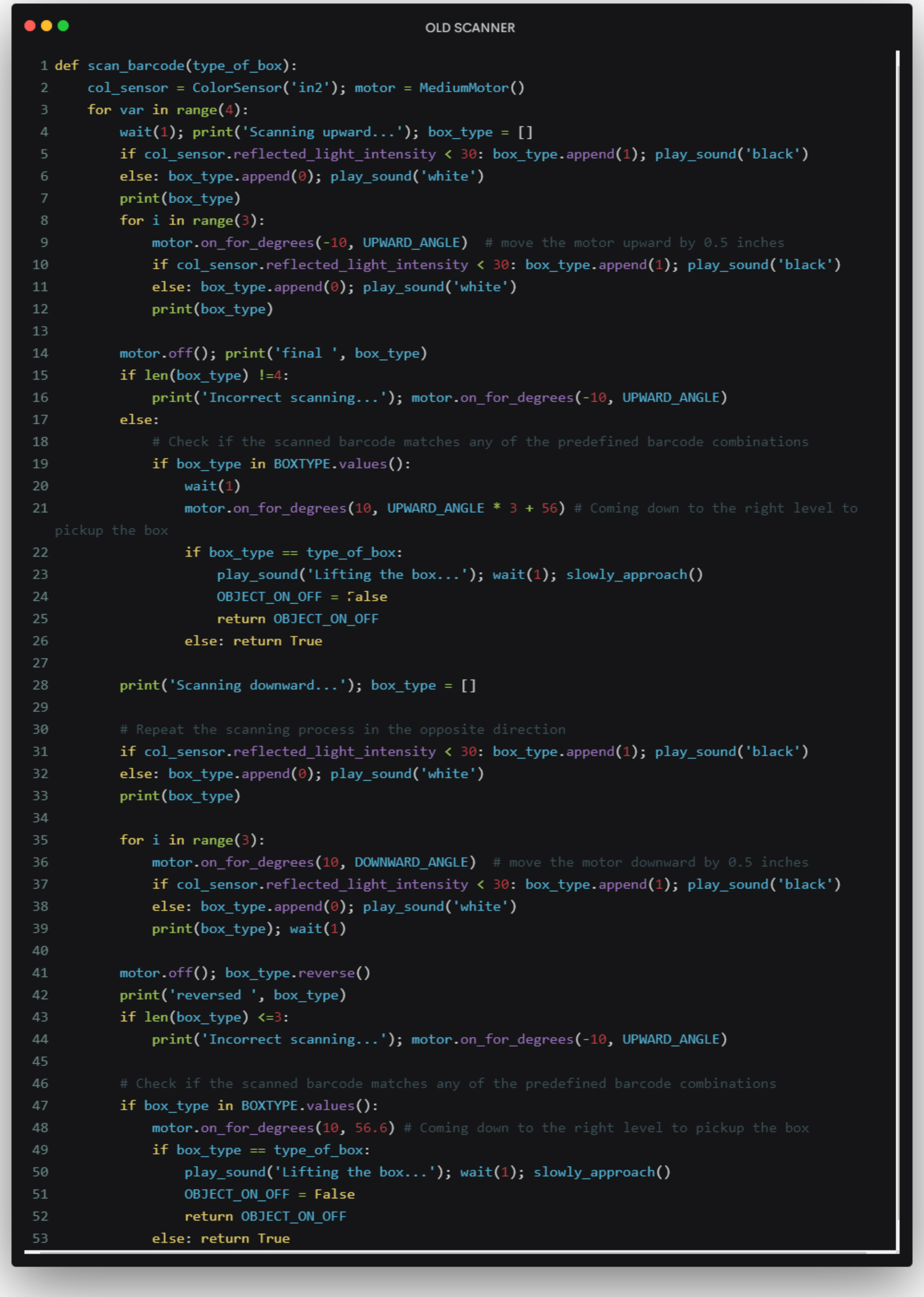


If it wrongly skips a color, it will rescan downward to try to correct its error.

**(CLICK BELOW TO WATCH)**



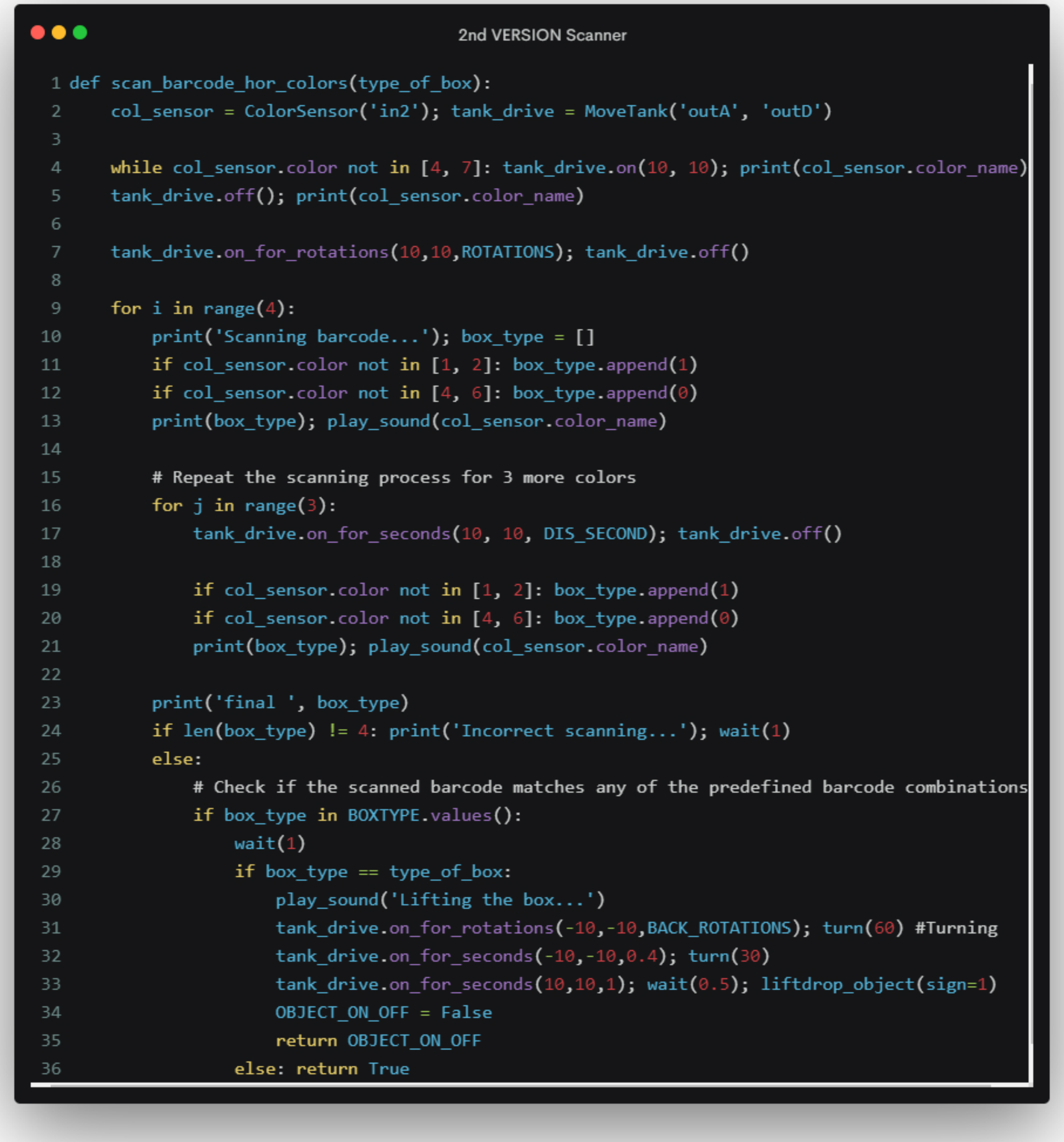
Here is the Code behind it:



OLD SCANNER

```python
def scan_barcode(type_of_box):
    col_sensor = ColorSensor('in2'); motor = MediumMotor()
    for var in range(4):
        wait(1); print('Scanning upward...'); box_type = []
        if col_sensor.reflected_light_intensity < 30: box_type.append(1); play_sound('black')
        else: box_type.append(0); play_sound('white')
        print(box_type)
        for i in range(3):
            motor.on_for_degrees(-10, UPWARD_ANGLE)  # move the motor upward by 0.5 inches
            if col_sensor.reflected_light_intensity < 30: box_type.append(1); play_sound('black')
            else: box_type.append(0); play_sound('white')
            print(box_type)

        motor.off(); print('final ', box_type)
        if len(box_type) !=4:
            print('Incorrect scanning...'); motor.on_for_degrees(-10, UPWARD_ANGLE)
        else:
            # Check if the scanned barcode matches any of the predefined barcode combinations
            if box_type in BOXTYPE.values():
                wait(1)
                motor.on_for_degrees(10, UPWARD_ANGLE * 3 + 56) # Coming down to the right level to pickup the box
                if box_type == type_of_box:
                    play_sound('Lifting the box...'); wait(1); slowly_approach()
                    OBJECT_ON_OFF = False
                    return OBJECT_ON_OFF
                else: return True

        print('Scanning downward...'); box_type = []

        # Repeat the scanning process in the opposite direction
        if col_sensor.reflected_light_intensity < 30: box_type.append(1); play_sound('black')
        else: box_type.append(0); play_sound('white')
        print(box_type)

        for i in range(3):
            motor.on_for_degrees(10, DOWNWARD_ANGLE)  # move the motor downward by 0.5 inches
            if col_sensor.reflected_light_intensity < 30: box_type.append(1); play_sound('black')
            else: box_type.append(0); play_sound('white')
            print(box_type); wait(1)

        motor.off(); box_type.reverse()
        print('reversed ', box_type)
        if len(box_type) <=3:
            print('Incorrect scanning...'); motor.on_for_degrees(-10, UPWARD_ANGLE)

        # Check if the scanned barcode matches any of the predefined barcode combinations
        if box_type in BOXTYPE.values():
            motor.on_for_degrees(10, 56.6) # Coming down to the right level to pickup the box
            if box_type == type_of_box:
                play_sound('Lifting the box...'); wait(1); slowly_approach()
                OBJECT_ON_OFF = False
                return OBJECT_ON_OFF
            else: return True
```

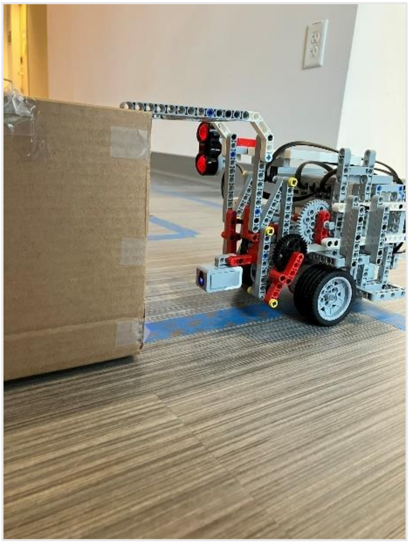
And this is the second version of our code scanner:

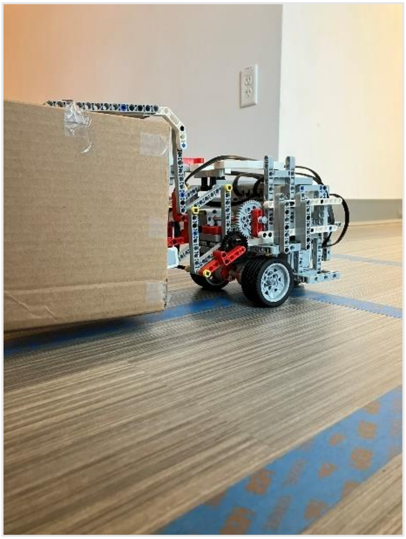


2nd VERSION Scanner

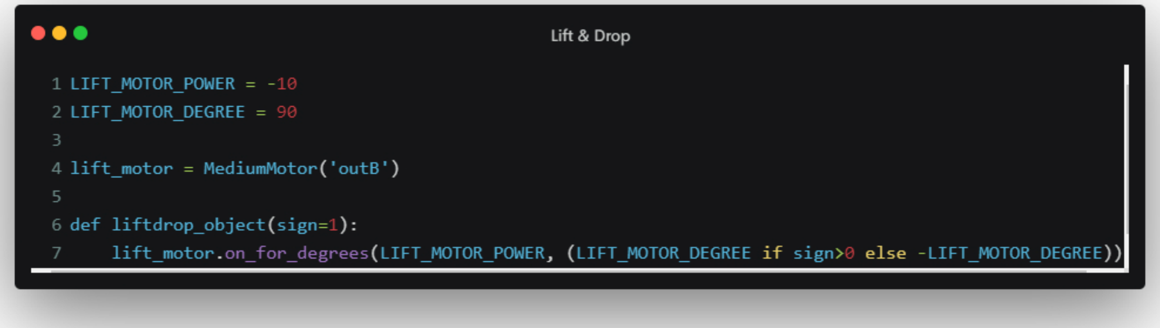
```
def scan_barcode_hor_colors(type_of_box):
    col_sensor = ColorSensor('in2'); tank_drive = MoveTank('outA', 'outD')

    while col_sensor.color not in [4, 7]: tank_drive.on(10, 10); print(col_sensor.color_name)
    tank_drive.off(); print(col_sensor.color_name)

    tank_drive.on_for_rotations(10,10,ROTATIONS); tank_drive.off()

    for i in range(4):
        print('Scanning barcode...'); box_type = []
        if col_sensor.color not in [1, 2]: box_type.append(1)
        if col_sensor.color not in [4, 6]: box_type.append(0)
        print(box_type); play_sound(col_sensor.color_name)

        # Repeat the scanning process for 3 more colors
        for j in range(3):
            tank_drive.on_for_seconds(10, 10, DIS_SECOND); tank_drive.off()

            if col_sensor.color not in [1, 2]: box_type.append(1)
            if col_sensor.color not in [4, 6]: box_type.append(0)
            print(box_type); play_sound(col_sensor.color_name)

        print('final ', box_type)
        if len(box_type) != 4: print('Incorrect scanning...'); wait(1)
        else:
            # Check if the scanned barcode matches any of the predefined barcode combinations
            if box_type in BOXTYPE.values():
                wait(1)
                if box_type == type_of_box:
                    play_sound('Lifting the box...')
                    tank_drive.on_for_rotations(-10,-10,BACK_ROTATIONS); turn(60) #Turning
                    tank_drive.on_for_seconds(-10,-10,0.4); turn(30)
                    tank_drive.on_for_seconds(10,10,1); wait(0.5); liftdrop_object(sign=1)
                    OBJECT_ON_OFF = False
                    return OBJECT_ON_OFF
                else: return True
```

**Lifting mechanism**: The lifting mechanism uses the medium motor to raise and lower the box. The medium motor is connected to a gear system that lifts the object as it rotates. The lifting height can be adjusted by changing the rotation angle of the motor, and the box will remain lifted as the





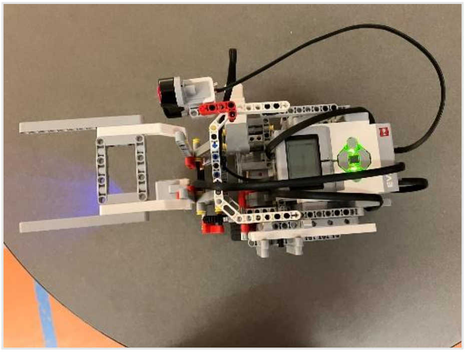
Here is the code behind it:

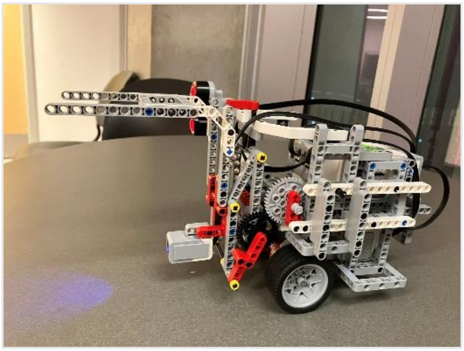


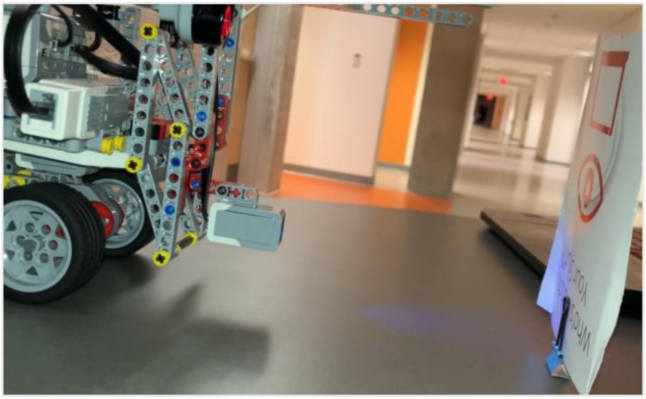
```
LIFT_MOTOR_POWER = -10
LIFT_MOTOR_DEGREE = 90

lift_motor = MediumMotor('outB')

def liftdrop_object(sign=1):
    lift_motor.on_for_degrees(LIFT_MOTOR_POWER, (LIFT_MOTOR_DEGREE if sign>0 else -LIFT_MOTOR_DEGREE))
```

Lifting and Dropping functions were merged into one function, it was easier for us to work with it rather than having two functions. The argument sign is where we can decide if it is picking up the box or dropping it (sign>0 means picking up | sign<0 means dropping off).

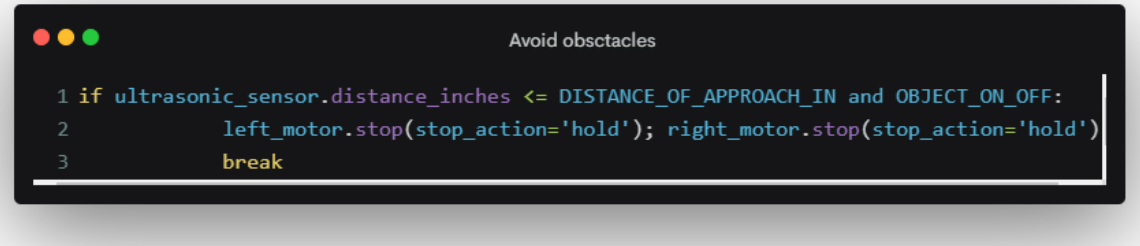
**Avoid Obstacle mechanism**: The avoid obstacle mechanism uses an ultrasonic sensor to detect obstacles in front of the robot. When an obstacle is detected within a certain range, the robot will stop and turn in a different direction to avoid the obstacle. This program allows the robot to navigate around obstacles and avoid collisions while still moving towards its intended destination.







This is the code behind it: (it was implemented inside the drive function)



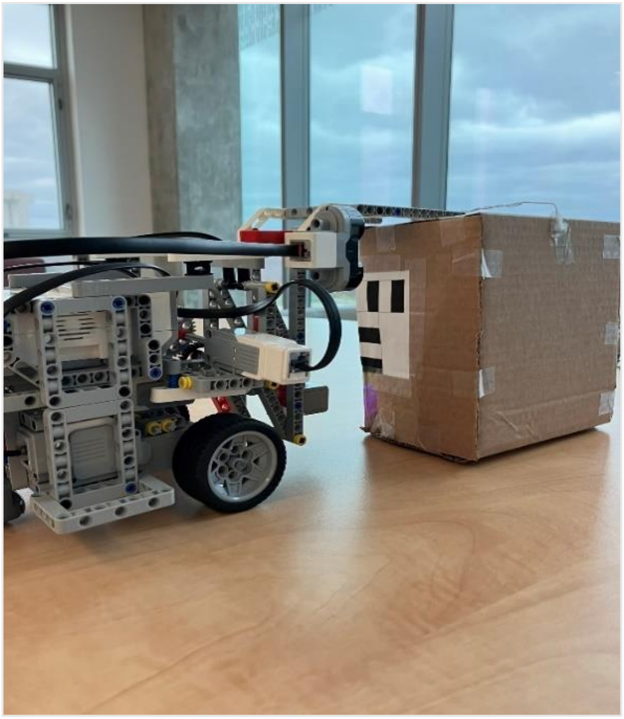
```
if ultrasonic_sensor.distance_inches <= DISTANCE_OF_APPROACH_IN and OBJECT_ON_OFF:
            left_motor.stop(stop_action='hold'); right_motor.stop(stop_action='hold')
            break
```

-TM

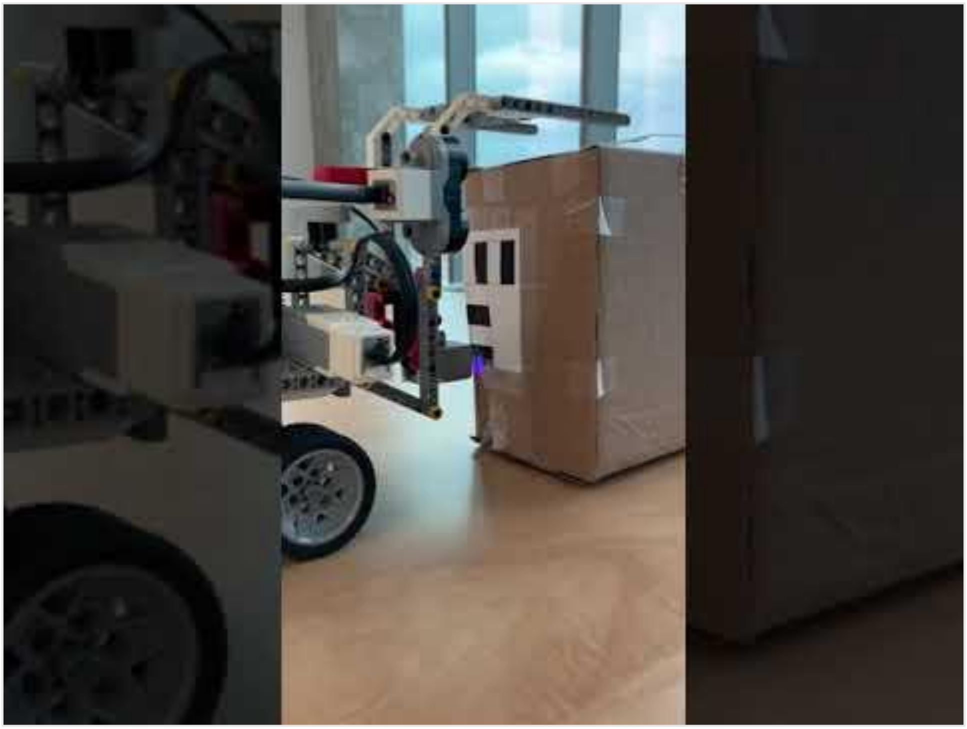
## Meeting 11: 04/01/2023

10:00am to 5:00pm

Taha was present. Taha worked on upgrading and fixing the code for mechanisms.



(CLICK BELOW TO WATCH)



(CLICK BELOW TO WATCH)

-TM

## Meeting 12: 04/01/2023

6:00pm to 9:00pm

The whole team was present, we completed the P4 Status Update. We made a video and talked about the following features:

Robot navigating: Explained in subtasks 1a and 1b.

Barcode scanning and picking up: Taha explains how it works.

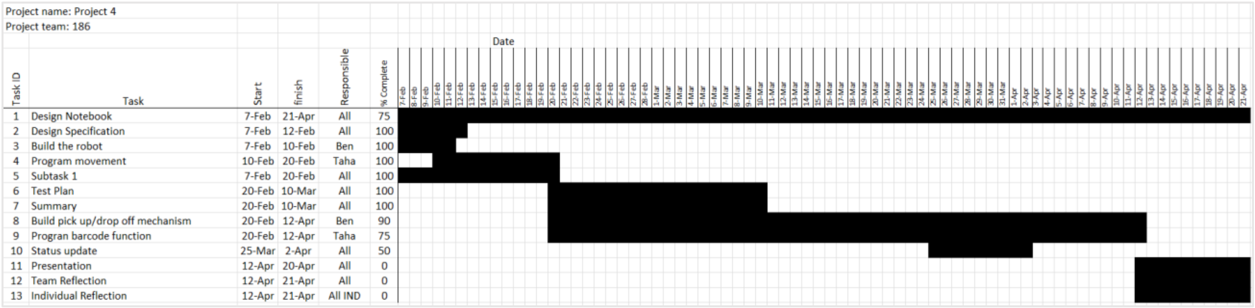
Lifting object and dropping off: Ben explains how it works.

Avoids obstacles: Yatharth explains how it works.

Luke talks about the Gantt Chart and project management.

-YB

Updated Gantt Chart:



Project name: Project 4
Project team: 186

| Task ID | Task | Start | finish | Responsible | % Complete |
|---|---|---|---|---|---|
| 1 | Design Notebook | 7-Feb | 21-Apr | All | 75 |
| 2 | Design Specification | 7-Feb | 12-Feb | All | 100 |
| 3 | Build the robot | 7-Feb | 10-Feb | Ben | 100 |
| 4 | Program movement | 10-Feb | 20-Feb | Taha | 100 |
| 5 | Subtask 1 | 7-Feb | 20-Feb | All | 100 |
| 6 | Test Plan | 20-Feb | 10-Mar | All | 100 |
| 7 | Summary | 20-Feb | 10-Mar | All | 100 |
| 8 | Build pick up/drop off mechanism | 20-Feb | 12-Apr | Ben | 90 |
| 9 | Progran barcode function | 20-Feb | 12-Apr | Taha | 75 |
| 10 | Status update | 25-Mar | 2-Apr | All | 50 |
| 11 | Presentation | 12-Apr | 20-Apr | All | 0 |
| 12 | Team Reflection | 12-Apr | 21-Apr | All | 0 |
| 13 | Individual Reflection | 12-Apr | 21-Apr | All IND | 0 |

## Meeting 13: 04/08/2023

6:00pm to 8:30pm

All the members were present. To prepare for the final demos, we tested the barcode scanning for the robot to iron out some potential issues with the system.

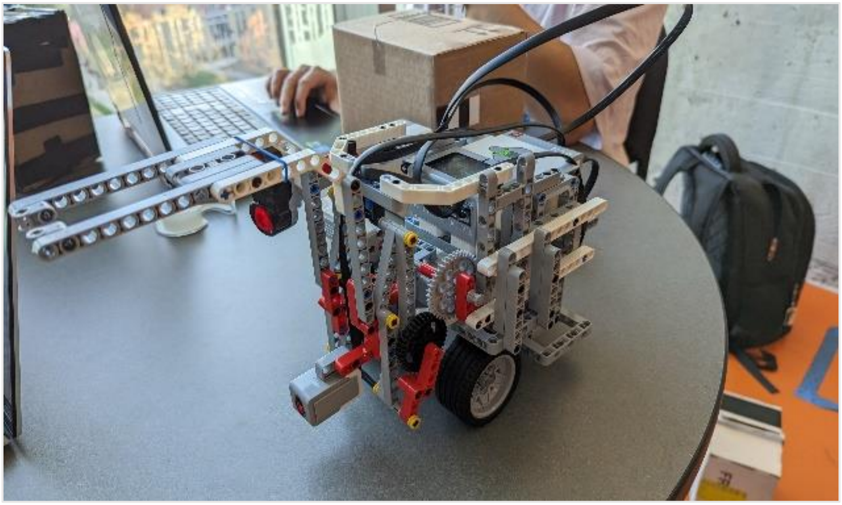
Additionally, a new sensor was added to the side of the robot to scan the barcodes of the boxes more efficiently. This will allow the robot to determine if the box has the correct barcode before turning to pick it up, which should make the tasks and final demo much more streamlined.

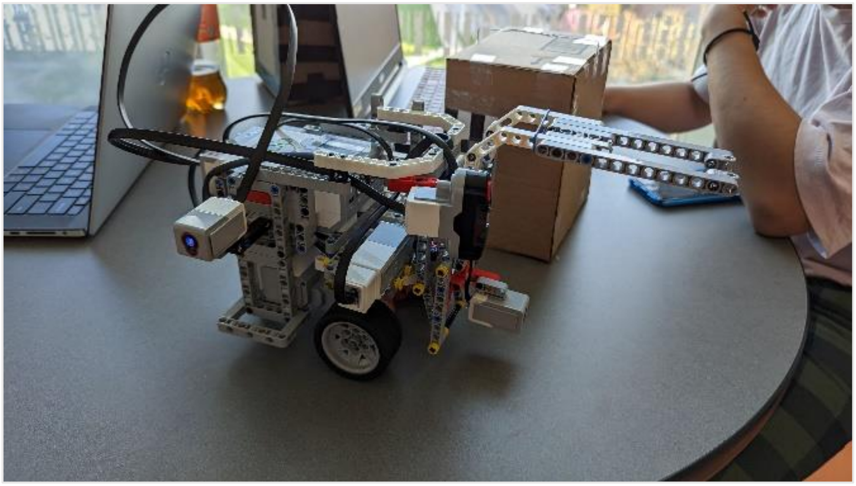
Also, it was found that using the reflective setting in the color sensor was a more effective method of determining the colors for the barcode, so the code was changed to use that instead of color identification method.

Barcode Scan Test Results:

| Barcode Type | Sensor Used | Correct Identification? |
|---|---|---|
| **1 (WBWB)** | Side | 1st try |
| **1** | Side | No |
| **1** | Side | 2nd try |
| **1** | Side | 1st try |
| **1** | Side | 1st try |
| **1** | Front | No |
| **1** | Front | 1st try |
| **1** | Front | 2nd try |
| **2 (BWBW)** | Side | 1st try |
| **2** | Side | 1st try |
| **2** | Side | 2nd try |

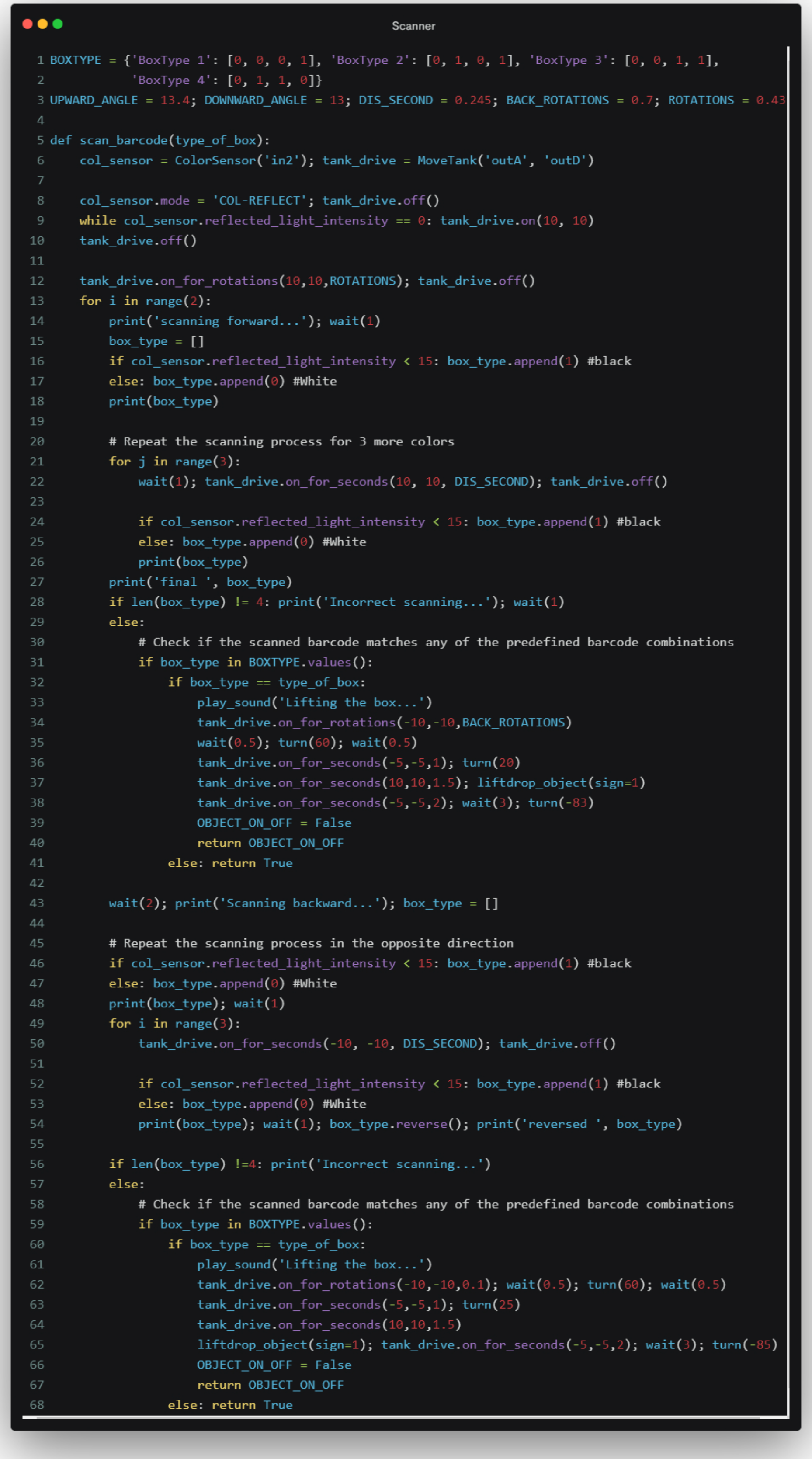
| | | |
|---|---|---|
| **2** | Side | 1st try |
| **2** | Side | 1st try |
| **2** | Side | 1st try |





-LR

The new barcode scanner code:



Scanner

```python
BOXTYPE = {'BoxType 1': [0, 0, 0, 1], 'BoxType 2': [0, 1, 0, 1], 'BoxType 3': [0, 0, 1, 1],
           'BoxType 4': [0, 1, 1, 0]}
UPWARD_ANGLE = 13.4; DOWNWARD_ANGLE = 13; DIS_SECOND = 0.245; BACK_ROTATIONS = 0.7; ROTATIONS = 0.43

def scan_barcode(type_of_box):
    col_sensor = ColorSensor('in2'); tank_drive = MoveTank('outA', 'outD')

    col_sensor.mode = 'COL-REFLECT'; tank_drive.off()
    while col_sensor.reflected_light_intensity == 0: tank_drive.on(10, 10)
    tank_drive.off()

    tank_drive.on_for_rotations(10,10,ROTATIONS); tank_drive.off()
    for i in range(2):
        print('scanning forward...'); wait(1)
        box_type = []
        if col_sensor.reflected_light_intensity < 15: box_type.append(1) #black
        else: box_type.append(0) #White
        print(box_type)

        # Repeat the scanning process for 3 more colors
        for j in range(3):
            wait(1); tank_drive.on_for_seconds(10, 10, DIS_SECOND); tank_drive.off()

            if col_sensor.reflected_light_intensity < 15: box_type.append(1) #black
            else: box_type.append(0) #White
            print(box_type)
        print('final ', box_type)
        if len(box_type) != 4: print('Incorrect scanning...'); wait(1)
        else:
            # Check if the scanned barcode matches any of the predefined barcode combinations
            if box_type in BOXTYPE.values():
                if box_type == type_of_box:
                    play_sound('Lifting the box...')
                    tank_drive.on_for_rotations(-10,-10,BACK_ROTATIONS)
                    wait(0.5); turn(60); wait(0.5)
                    tank_drive.on_for_seconds(-5,-5,1); turn(20)
                    tank_drive.on_for_seconds(10,10,1.5); liftdrop_object(sign=1)
                    tank_drive.on_for_seconds(-5,-5,2); wait(3); turn(-83)
                    OBJECT_ON_OFF = False
                    return OBJECT_ON_OFF
                else: return True

        wait(2); print('Scanning backward...'); box_type = []

        # Repeat the scanning process in the opposite direction
        if col_sensor.reflected_light_intensity < 15: box_type.append(1) #black
        else: box_type.append(0) #White
        print(box_type); wait(1)
        for i in range(3):
            tank_drive.on_for_seconds(-10, -10, DIS_SECOND); tank_drive.off()

            if col_sensor.reflected_light_intensity < 15: box_type.append(1) #black
            else: box_type.append(0) #White
            print(box_type); wait(1); box_type.reverse(); print('reversed ', box_type)

        if len(box_type) !=4: print('Incorrect scanning...')
        else:
            # Check if the scanned barcode matches any of the predefined barcode combinations
            if box_type in BOXTYPE.values():
                if box_type == type_of_box:
                    play_sound('Lifting the box...')
                    tank_drive.on_for_rotations(-10,-10,0.1); wait(0.5); turn(60); wait(0.5)
                    tank_drive.on_for_seconds(-5,-5,1); turn(25)
                    tank_drive.on_for_seconds(10,10,1.5)
                    liftdrop_object(sign=1); tank_drive.on_for_seconds(-5,-5,2); wait(3); turn(-85)
                    OBJECT_ON_OFF = False
                    return OBJECT_ON_OFF
                else: return True
```

This was the last algorithm Taha adapted after many tries with different algorithms which includes using color mode instead of reflection and the first algorithm was assigned to the vertical scanning. Our team then decided to change to horizontal scanner mechanism because of how easy it would be for us to implement in the Tasks.

This scanning function make the robot drive forward until it sees the box, it stops and then move until it reaches the first color, stops, read it and store it, repeat until all the colors were read and then it compares it with the pre-combined barcodes, if it is not one of them then the robot will scan backward until it is satisfied with the result and that means it scans one of the possible barcodes. Then it checks if it is the one that we are looking for, if so, it does the lifting actions.

-TM

## Meeting 14: 04/09/2023

11:00am to 2:00pm

All the members were present. We completed more tests, such as the pickup tests and drop off tests. We all determined that using only the side sensor for reading barcodes was easier than using the front sensor, so we switched to using that method mainly.

Drop Off Test Results:

| Distance Travelled | x - error from drop off spot (in) | y - error from drop off spot (in) | Box dropped? |
|---|---|---|---|
| 72 in | 8 | 1 | yes |
| 72 in | 10 | 3 | yes |
| 72 in | 24 | 1 | no |
| 72 in | 0 | 0 | yes |
| 72 in | 0 | 0 | yes |

Pickup and Carry Box Test Results:

| *Trial* | *Pickup Successful?* | *Inches travelled before issues* |
|---|---|---|
| *1* | no | 24 (spot of the turn) |
| *2* | no | 24 |
| *3* | no | 24 |
| *4* | no | 0 |
| *5* | no | 0 |
| *6* | no | 30 (after turn) |
| *7* | no | 30 |
| *8* | no | 36 (at box) |
| *9* | yes | N/A |
| *10* | yes | N/A |
| *11* | yes | 36 |
| *12* | yes | N/A |
| *13* | no | 36 |
| *14* | yes | N/A |
| *15* | no | 30 |
| *16* | yes | N/A |
| *17* | yes | N/A |

As we tested the robot’s abilities, we adjusted the code to help compensate for its errors. We tested until we felt confident in its abilities.

-LR

## Meeting 15: 04/11/2023

6:30pm to 8:00pm

All team members were present. We tested all 4 subtasks in preparation for the final demo. To make the testing experience as realistic as possible, we set up a simulation in one of the dorm hallways. This provided us with a platform to practice and refine our skills. The simulation allowed us to iron out any kinks, making sure that our team was operating at optimal levels. We could assess the efficiency of our team members and make any necessary changes to ensure we were all working seamlessly.

Our results were as follows:

| | **Completed?** | **Issue?** | **Error (if applicable)** |
|---|---|---|---|
| **Subtask 1** | | | |
| **Trial 1** | Yes | N/A | N/A |
| **2** | Yes | N/A | N/A |
| **3** | Yes | N/A | N/A |
| **4** | Yes | N/A | N/A |
| **5** | Yes | N/A | N/A |
| **Subtask 2** | | | |
| **Trial 1** | Partially | Off center | 3" y-direction |
| **2** | Partially | Stopped early | 3" before home |
| **3** | Partially | Stopped early | 3" before home |
| **4** | Yes | N/A | N/A |
| **5** | Yes | N/A | N/A |
| **Subtask 3** | | | |
| **Trial 1** | No | Incorrect readings | Bot too far away from box |
| **2** | No | Incorrect readings | Bot misaligned with barcode |
| **3** | Yes | N/A | N/A |
| **4** | Yes | N/A | N/A |
| **5** | Yes | N/A | N/A |
| **6** | Yes | N/A | N/A |
| **Subtask 4** | | | |
| **Trial 1** | No | Lift didn't raise | Gear malfunction |
| **2** | Partially | Drove too far | 6" from aisle |
| **3** | Partially | Did not stop moving | ← |
| **4*** | Yes | N/A | N/A |

*Tasks 3 + 4 completed together

Throughout these tests, small improvements were made in code regarding the navigation of the bot, so now it travels in a straighter line more consistently.

-LR

Here is a video of our robot from testing. It successfully completed all the individual tasks and also completed the final track.

**(CLICK BELOW TO WATCH)**

-YB

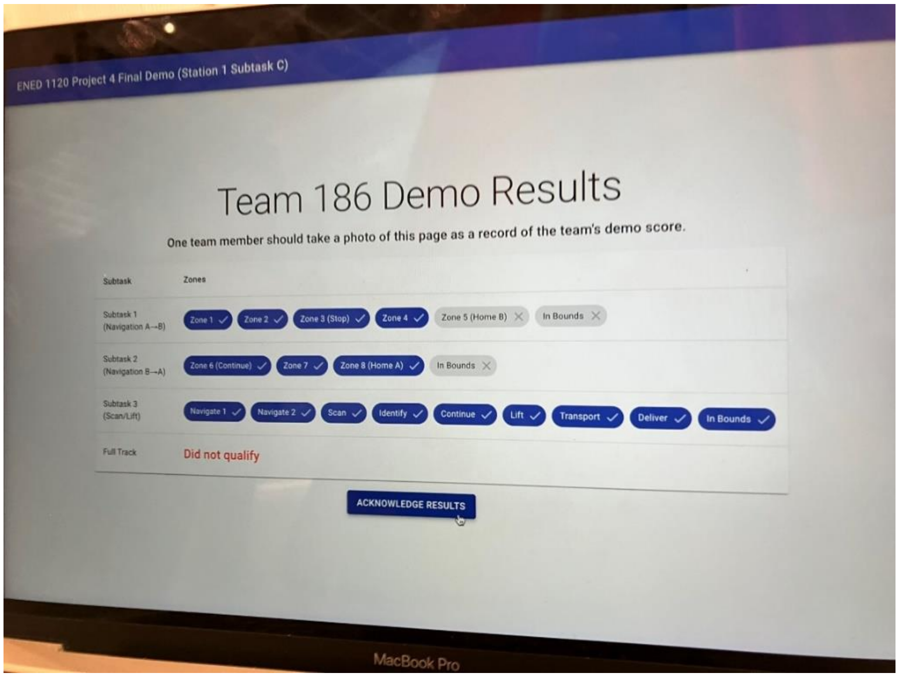
## Final demo: 04/12/2023

6:00pm to 7:00pm

All the members were present. We demonstrated all subtasks.

The first and second subtask weren't as successful as we thought because of gyroscope issues (increasing values of gyro while the robot is in a stable position made the robot drift while moving).

After fixing the gyroscope, our third and fourth subtasks were successful, and we didn't lose any points for them.



ENED 1120 Project 4 Final Demo (Station 1 Subtask C)

Team 186 Demo Results

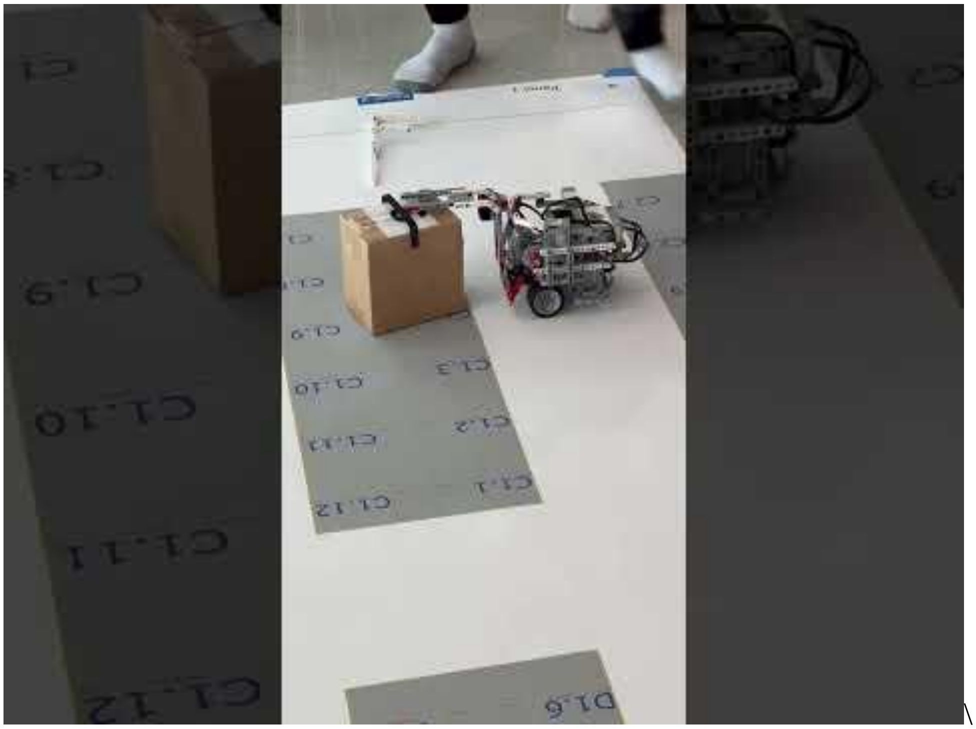
One team member should take a photo of this page as a record of the team's demo score.

| Subtask | Zones |
|---|---|
| Subtask 1 (Navigation A→B) | Zone 1 ✓ Zone 2 ✓ Zone 3 (Stop) ✓ Zone 4 ✓ Zone 5 (Home B) ✗ In Bounds ✗ |
| Subtask 2 (Navigation B→A) | Zone 6 (Continue) ✓ Zone 7 ✓ Zone 8 (Home A) ✓ In Bounds ✗ |
| Subtask 3 (Scan/Lift) | Navigate 1 ✓ Navigate 2 ✓ Scan ✓ Identify ✓ Continue ✓ Lift ✓ Transport ✓ Deliver ✓ In Bounds ✓ |
| Full Track | Did not qualify |

ACKNOWLEDGE RESULTS

Below is a video of our robot doing the 3rd and 4th task successfully at the final demonstration.

**(CLICK BELOW TO WATCH)**



\

The robot was able to successfully identify the box, scan the barcode, pick the box, and drop it off at the drop off location. We believe we could've been able to do well at the final track but unfortunately, we didn't qualify for it.

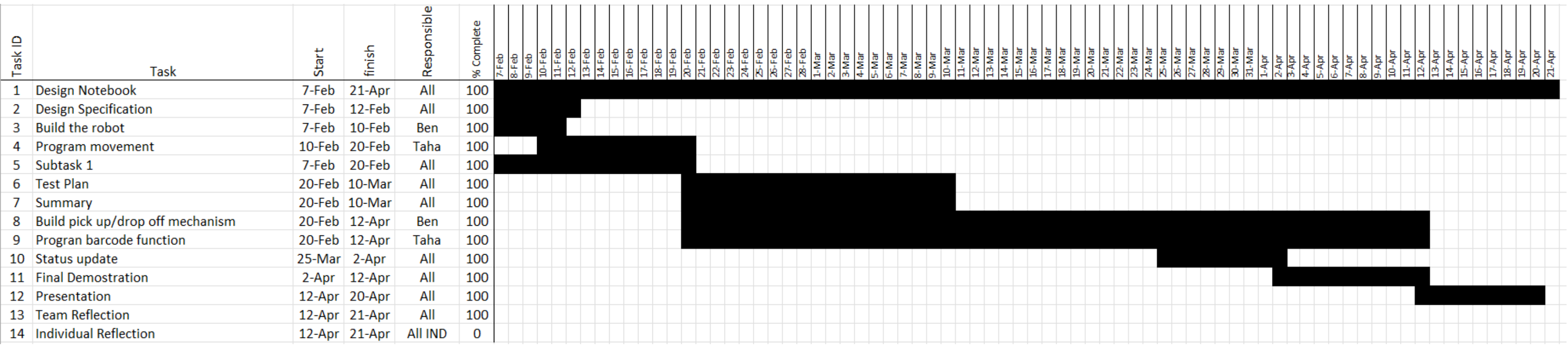
-YB

## Meeting 16: 04/18/2023

6:30pm to 8:00pm

All members were present, we worked on and finished the presentation for class, and continued work on the design notebook. Additionally, we deconstructed the robot and counted all the parts. Luke and Yatharth were assigned to finish compiling the design notebook. All members were assigned to do a part in the team reflection.

Final Gantt Chart:



| Task ID | Task | Start | finish | Responsible | % Complete |
|---|---|---|---|---|---|
| 1 | Design Notebook | 7-Feb | 21-Apr | All | 100 |
| 2 | Design Specification | 7-Feb | 12-Feb | All | 100 |
| 3 | Build the robot | 7-Feb | 10-Feb | Ben | 100 |
| 4 | Program movement | 10-Feb | 20-Feb | Taha | 100 |
| 5 | Subtask 1 | 7-Feb | 20-Feb | All | 100 |
| 6 | Test Plan | 20-Feb | 10-Mar | All | 100 |
| 7 | Summary | 20-Feb | 10-Mar | All | 100 |
| 8 | Build pick up/drop off mechanism | 20-Feb | 12-Apr | Ben | 100 |
| 9 | Progran barcode function | 20-Feb | 12-Apr | Taha | 100 |
| 10 | Status update | 25-Mar | 2-Apr | All | 100 |
| 11 | Final Demostration | 2-Apr | 12-Apr | All | 100 |
| 12 | Presentation | 12-Apr | 20-Apr | All | 100 |
| 13 | Team Reflection | 12-Apr | 21-Apr | All | 100 |
| 14 | Individual Reflection | 12-Apr | 21-Apr | All IND | 0 |

-LR

# Project Costs: 04/20/2023

As a team, we put in a collective 135 hours into project 4. 108 of those hours were as a team in meetings, and 27 hours were as individuals outside of meetings. Using the conversion rate of $40 per hour, the total cost of the hours worked was **$5400**. This cost is less than half of what we originally expected in the design specification review. We believe that we overestimated by so much in the beginning mainly because we expected to be doing much more work individually, however, that amount turned out to be much lower than expected, as we were very efficient and effective in our team meetings. In the future we could generate a better estimate by trusting the effectiveness of our team. We believe we would be much more productive while working in our team meetings than while working individually, so we could put more emphasis on that, which would bring down the estimated final cost.

-LR