Project Title: Smart Carpark

BUILDING A SMART CAR-PARK USING AN ARDUINO

Module Code:EGx154
Module Description: Introduction to Engineering
Student Name: Kwa Jian Quan
Admin No. : 202525R
Module Group : EC2001

BACKGROUND/PROBLEMS/ISSUES

- Drivers spend time looking for parking lots in car park
- Drivers go to a carpark only to find out that it is full
- Dark and stuffy carpark
- Drivers find it hard to park their car in a carpark as the mirrors are obstructed by nearby car
- Drivers forgot where they parked their car

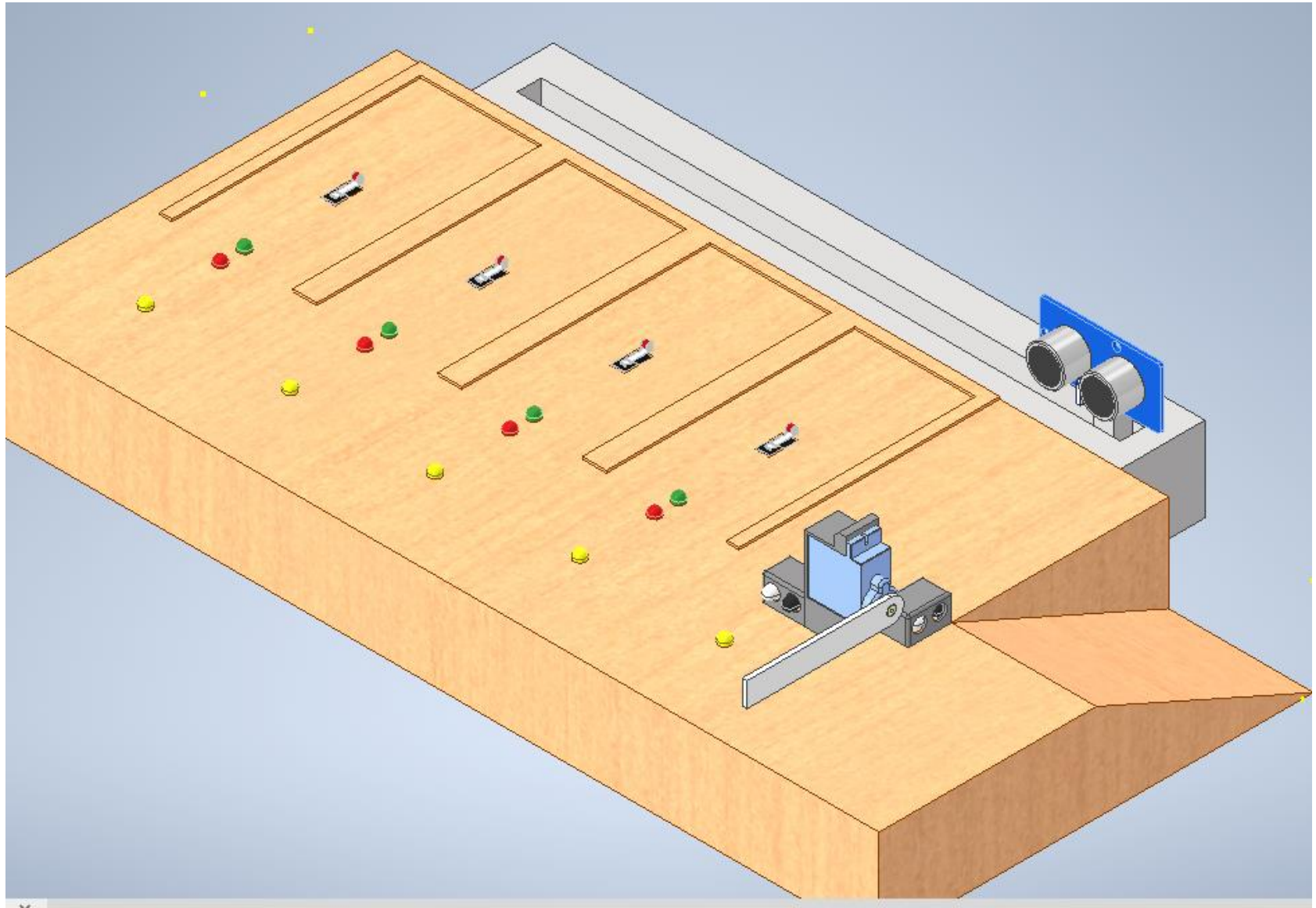
PROJECT OBJECTIVE

In line with Singapore's goal to become a smart nation, I propose a miniature car park model to explore the idea of a smart carpark, which:

1. Assign lots to each driver and guide them there
2. Open-design with retractable roof (activates during rain/sunny days)
3. Allows drivers to check on the number of slots available beforehand
4. Not allow cars to enter when slots are fully filled
5. Inform drivers on distance from wall when parking
6. Indicates which parking lot is empty/filled using red and green LEDs

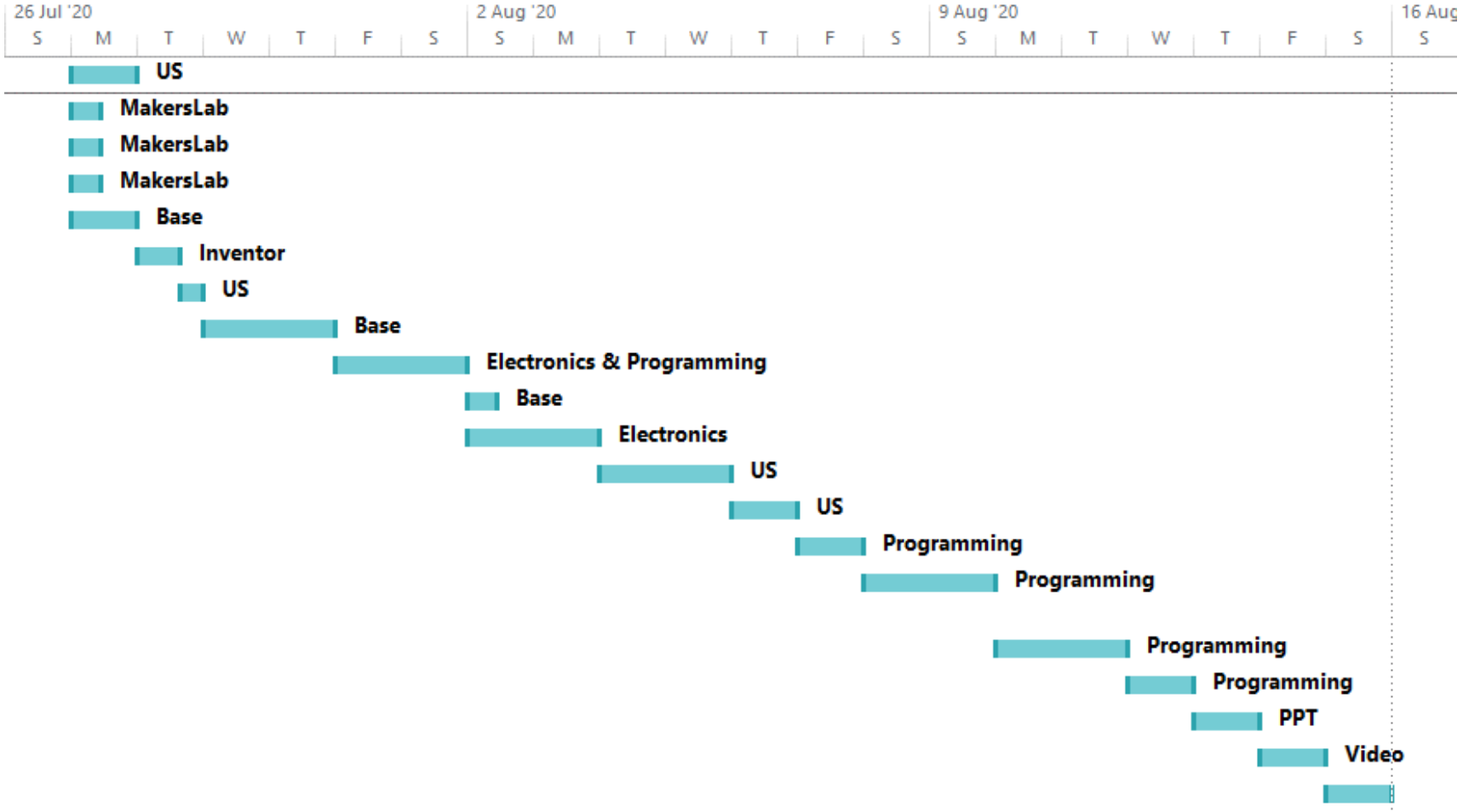
This will solve most of the problem/issues discussed in the previous slide

PICTURE ILLUSTRATION OF SMART CAR PARK



PROJECT GANN CHART

ID	Task Mode	Task Name	Duration	Start	Finish
1		Separate scanner from spoilt printer	1 day	Mon 27/7/20	Mon 27/7/20
2		Test Model of small Servo	0.1 days	Mon 27/7/20	Mon 27/7/20
3		Test IR holder	0.1 days	Mon 27/7/20	Mon 27/7/20
4		Test small servo holder	0.1 days	Mon 27/7/20	Mon 27/7/20
5		Pick Cardboard	1 day	Mon 27/7/20	Mon 27/7/20
6		Get dimensions of scanner, modify assembly size	0.5 days	Tue 28/7/20	Tue 28/7/20
7		Test Stepper Motor of scanner with driver	0.5 days	Tue 28/7/20	Tue 28/7/20
8		Build Car Park Base	2 days	Wed 29/7/20	Thu 30/7/20
9		Build Electronics and coding (without stepper)	2 days	Fri 31/7/20	Sat 1/8/20
10		Build/Print Car Park lots	0.1 days	Sun 2/8/20	Sun 2/8/20
11		Mount Electronics to Car Park Base and test	2 days	Sun 2/8/20	Mon 3/8/20
12		Mount Ultrasonic Sensor to Scanner	2 days	Tue 4/8/20	Wed 5/8/20
13		Mount Scanner to Car Park Base	1 day	Thu 6/8/20	Thu 6/8/20
14		Create functions for Car Park lots distances	1 day	Fri 7/8/20	Fri 7/8/20
15		Sub-code to measure distances, alert when distance reached	2 days	Sat 8/8/20	Sun 9/8/20
16		Combine Codes	2 days	Mon 10/8/20	Tue 11/8/20



GANN CHART

US: Ultrasonic Sensor
Inventor: 3D CAD

THE ENVIRONMENT

When creating this prototype, I strive to reduce the environmental impact.

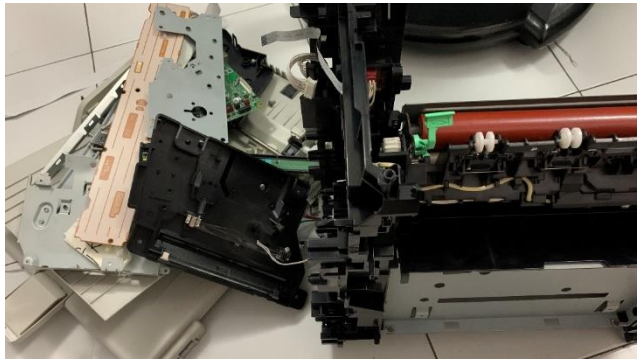
This is done by:

1. Obtaining the stepper motor and pully system from a spoilt printer scanner
2. 3D printing part of the completed product for testing
3. Using CAD (Autodesk Inventor) to create sketches
4. Hot Glue sticks, used to stick the cardboard prototype together, have been used sparingly. Approximately 120mm of glue stick is used.

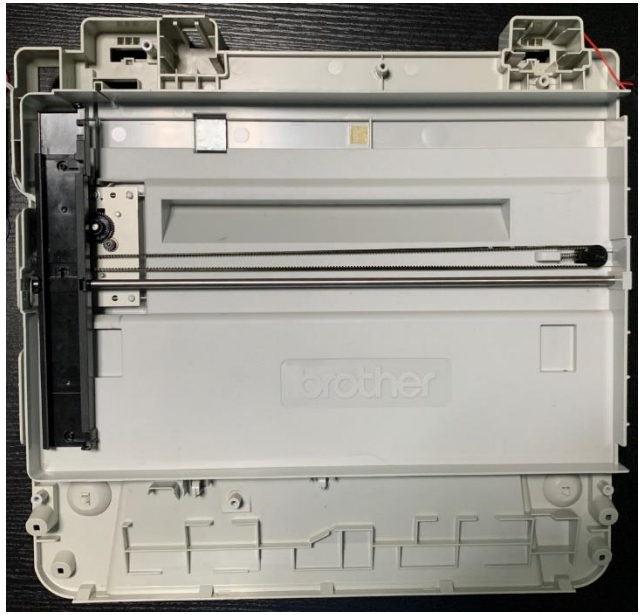
OBTAINING THE SERVO MOTOR AND PULLY SYSTEM

I found a spoilt printer, MFC 7360, and began to disassemble the scanner portion. The other components will be recycled, either in the Blue recycle bin (plastic) or Green recycle bin (E-waste)

Top:
Plastic
Recyclables



Bottom:
Electronics/metal
Recyclables

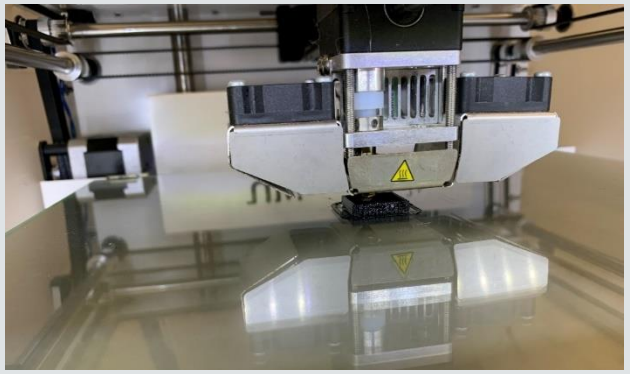


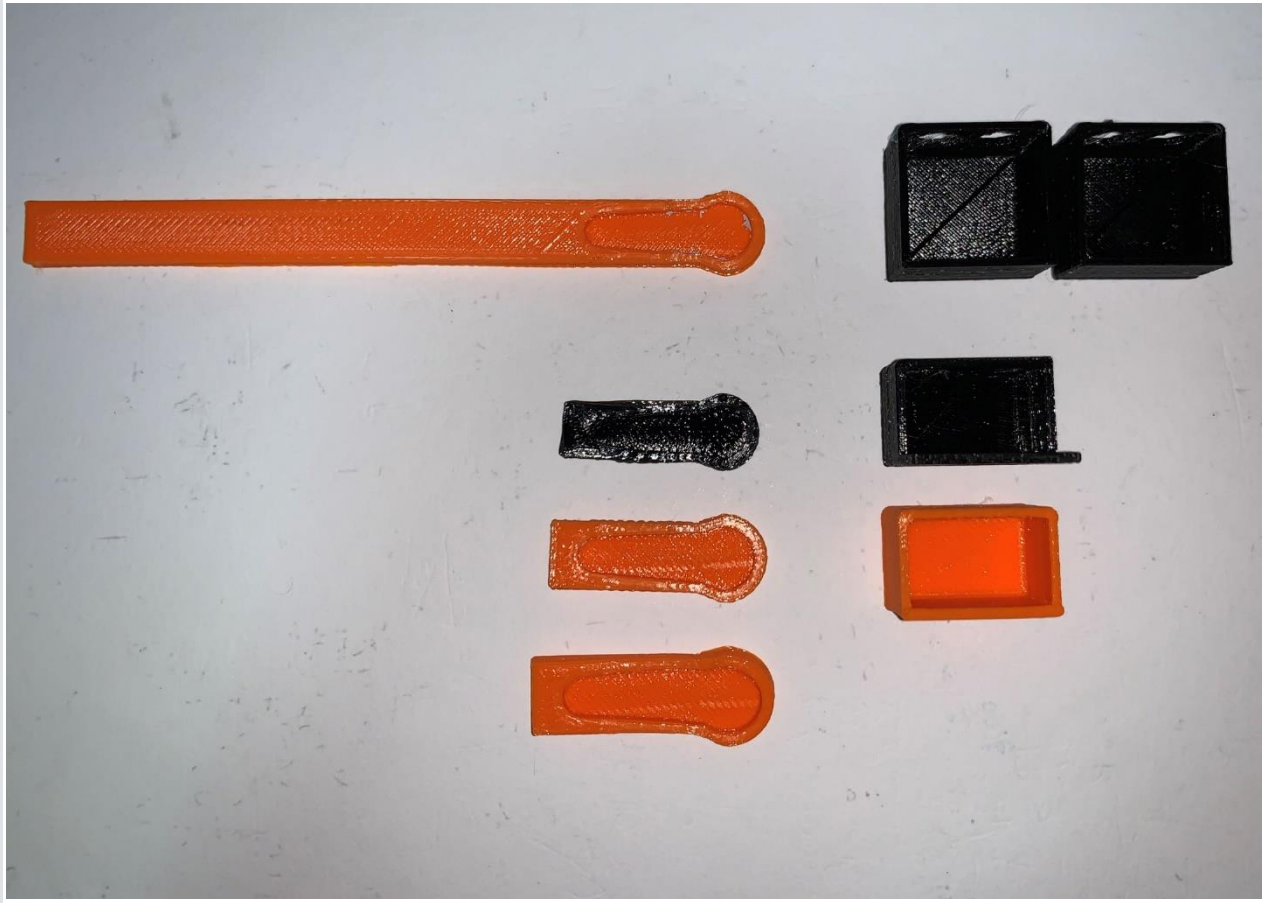
Disassembled scanner (with pully, belt, stepper motor intact)

3D PRINTING

I have printed a few accessories to assist in my project, such as the car gate arm and servo mount. I used the 3D printers in Makers Lab to produce my designs.

Total materials used in failed prints: 15 grams





Top: Final successful prints
Bottom: Failed prints

CAD

As I am new to Autodesk Inventor, I went through a 4-hour course through Lynda.com to learn the basics skills. The link of the course will be in the appendix slide.

This will reduce the need for paper, which will reduce the environmental impact.

Note: 3D files are designed from scratch, not copied from the internet, except existing electrical components such as servos and LEDs

AUTODESK® INVENTOR®

Assembly	28/7/2020 11:14 am	
IR sensor	24/7/2020 11:34 am	
Led	25/7/2020 10:43 am	
OldVersions	31/7/2020 8:16 pm	
Servos	2/8/2020 12:21 am	
STL	2/8/2020 12:22 am	
Ultrasonic	25/7/2020 10:40 am	
Carpark base (for scanner).ipt	29/7/2020 11:29 am	212 KB
Carpark base.ipt	24/7/2020 1:19 pm	235 KB
CarPark Plan.iam	28/7/2020 11:04 am	86 KB
display mount.ipt	31/7/2020 8:01 pm	126 KB
IR head.ipt	27/7/2020 3:17 pm	291 KB
micro switch.ipt	24/7/2020 1:30 pm	1,071 KB
Parking lot.ipt	24/7/2020 10:20 am	236 KB
Smart Car Park.ipj	23/7/2020 10:57 pm	10 KB
US mount.ipt	31/7/2020 8:16 pm	162 KB

Files for the designs, saving at least 140 sheets of paper saved

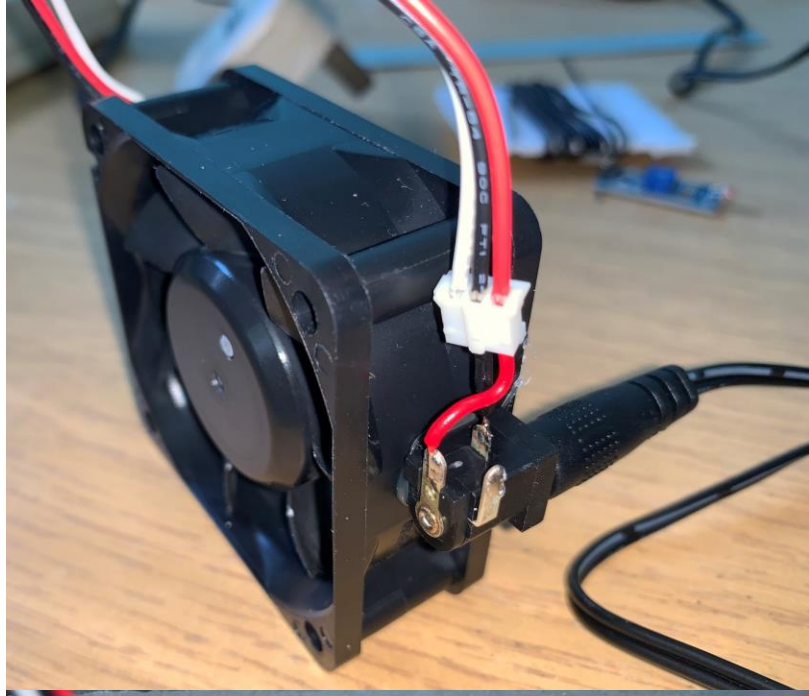
SAFETY AND ORGANIZATION

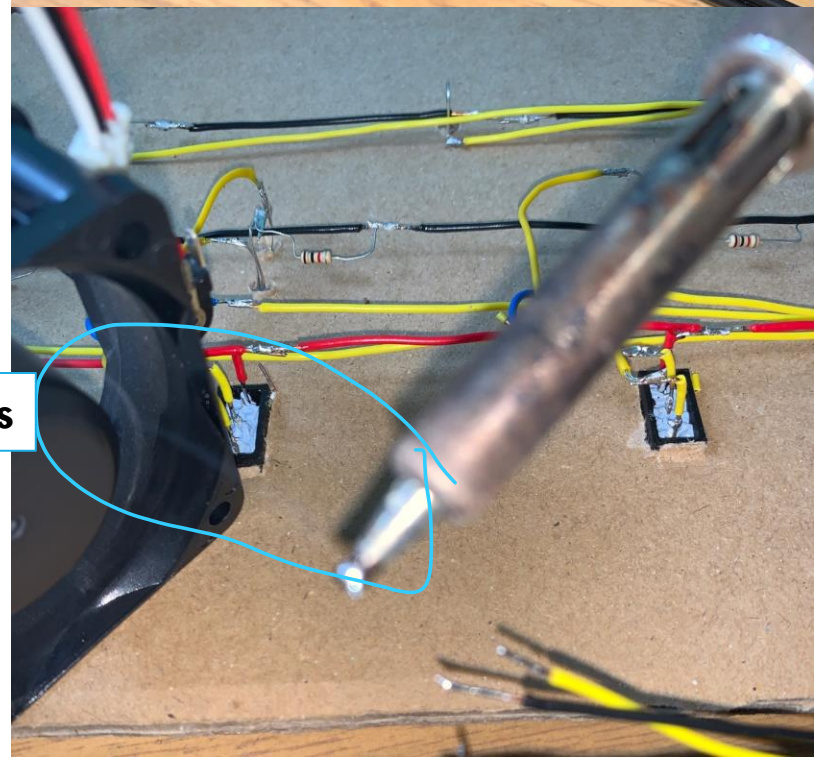
Due to the frequent (up to 12 hours) use of the soldering iron, as well as the hot glue gun, I have decided to create a workbench using an extra table. Hence, I can Ensure organisation and neatness, hence preventing accidental burning.

A safety glasses is also used to prevent solder from hitting the eye.

SAFETY

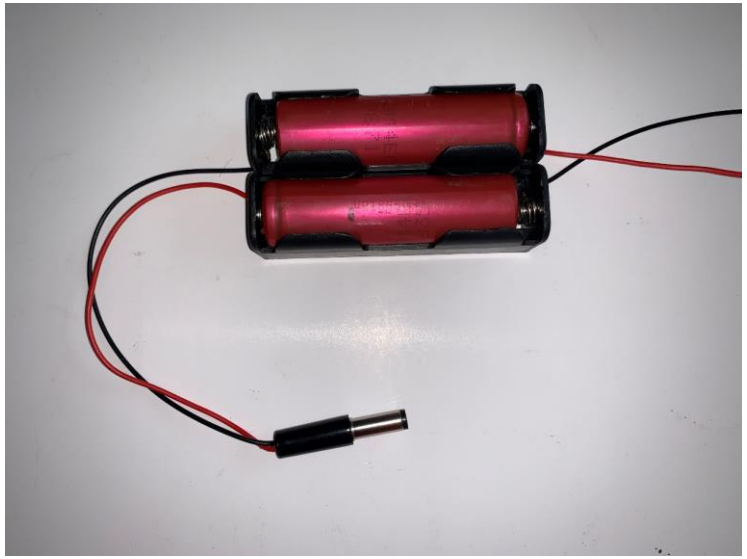
Fumes are produced when I am soldering electronics. Although it does not really affect me, I still put some safety precautions to prevent me from breathing too much of these fumes. This is done by replicating the MakerSpaceNYP solution, by using a fan to redirect the fumes. From the disassembly of the printer, I managed to find this cooling fan, which can be powered with the Arduino DC adapter. I soldered the female dc jack to the fan, and hot glued it on the side of the fan.





SAFETY (18650 BATTERY)

Initially, when I require the use of the battery, I will have to twist the wires together. However, short circuit can occur easily and cause an electrical fire. Hence, I soldered the 18650 wires to a DC adapter, to Ensure that the wires will not short circuit as easily. Furthermore, when the battery is not in use, an insulating material is placed to create an open circuit.



PHASES OF PROJECT

Phase 1: Building of Car Park with Green and Red LEDs to indicate if slot is occupied.

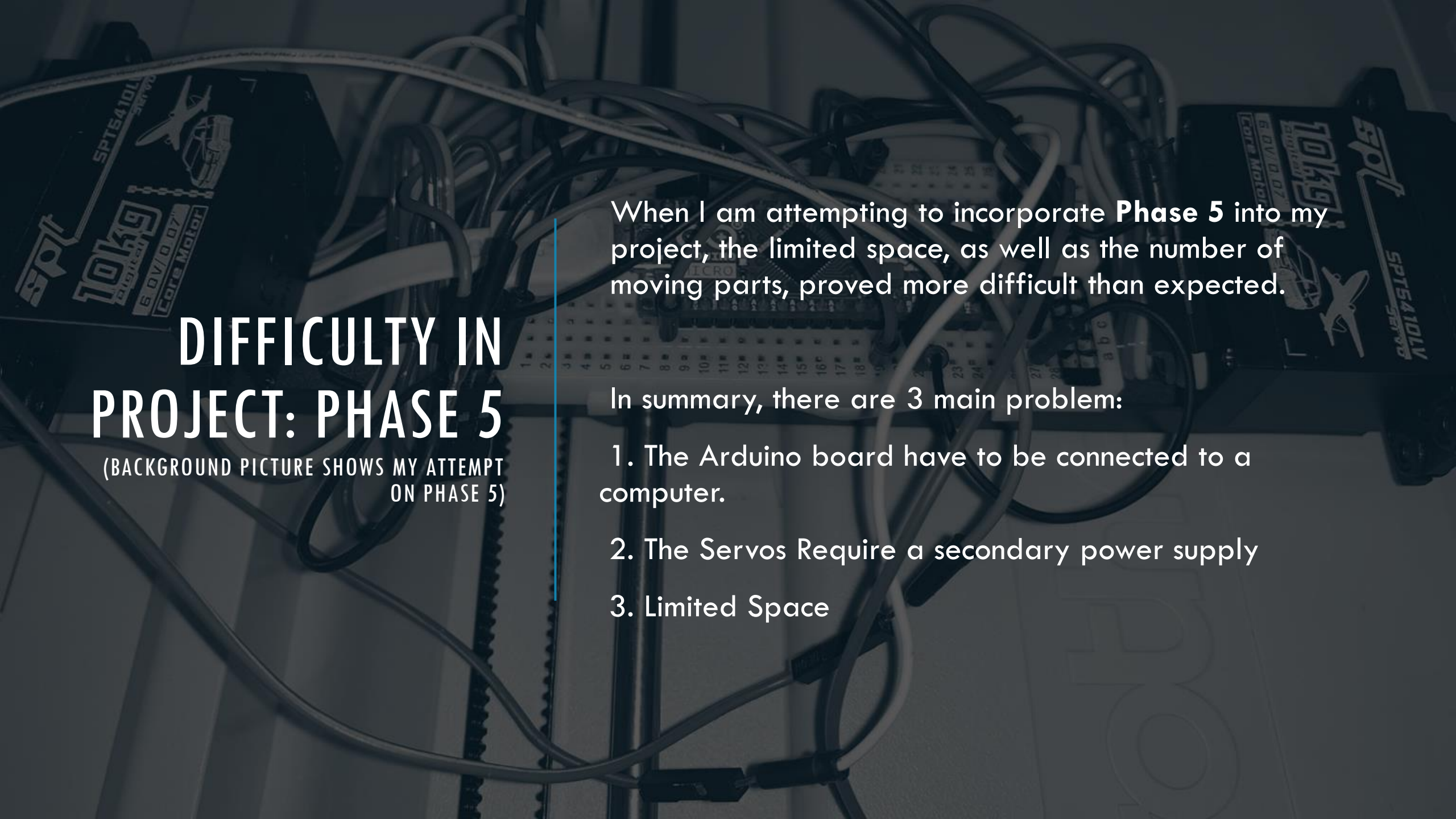
Phase 2: Incorporating Yellow LEDs to guide car into their assigned slot (closest) slot, as well as building a barrier which only allows a car to enter when there are slots available

Phase 3: Incorporating an ultrasonic sensor and a buzzer to indicate to the driver the vehicle distance away from the wall

Phase 4: Incorporating Green, Yellow and Red LED to indicate how close the vehicle is from the wall

Phase 5: Include 2 servos to move the ultrasonic sensor and LED display downwards when not in use, for a cleaner, more aesthetically pleasing look

Phase 5 was not completed due to difficulties, further explained in slide 19



DIFFICULTY IN PROJECT: PHASE 5

(BACKGROUND PICTURE SHOWS MY ATTEMPT ON PHASE 5)

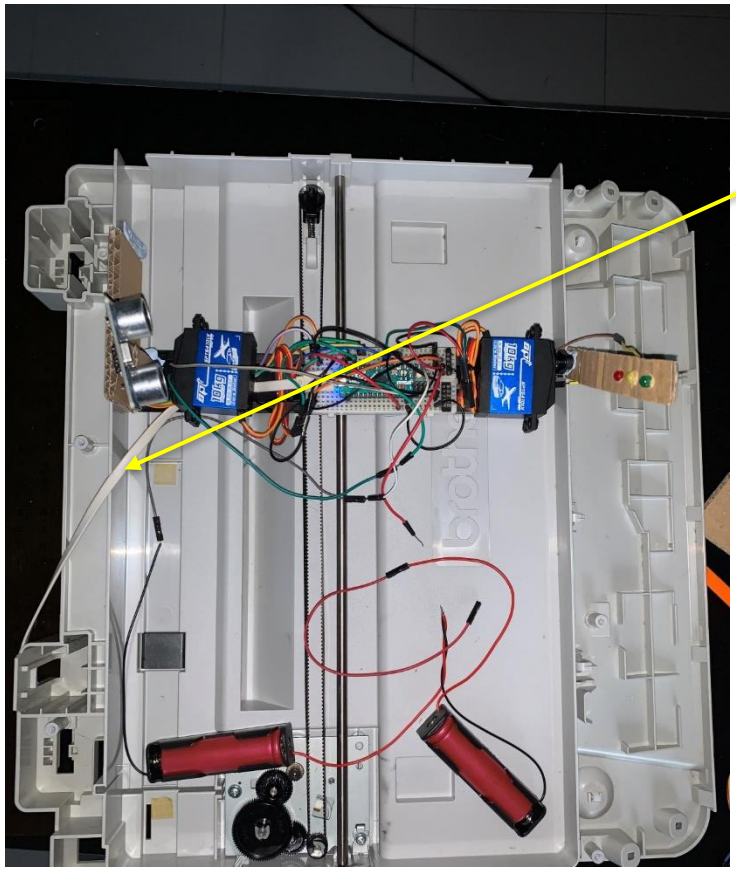
When I am attempting to incorporate **Phase 5** into my project, the limited space, as well as the number of moving parts, proved more difficult than expected.

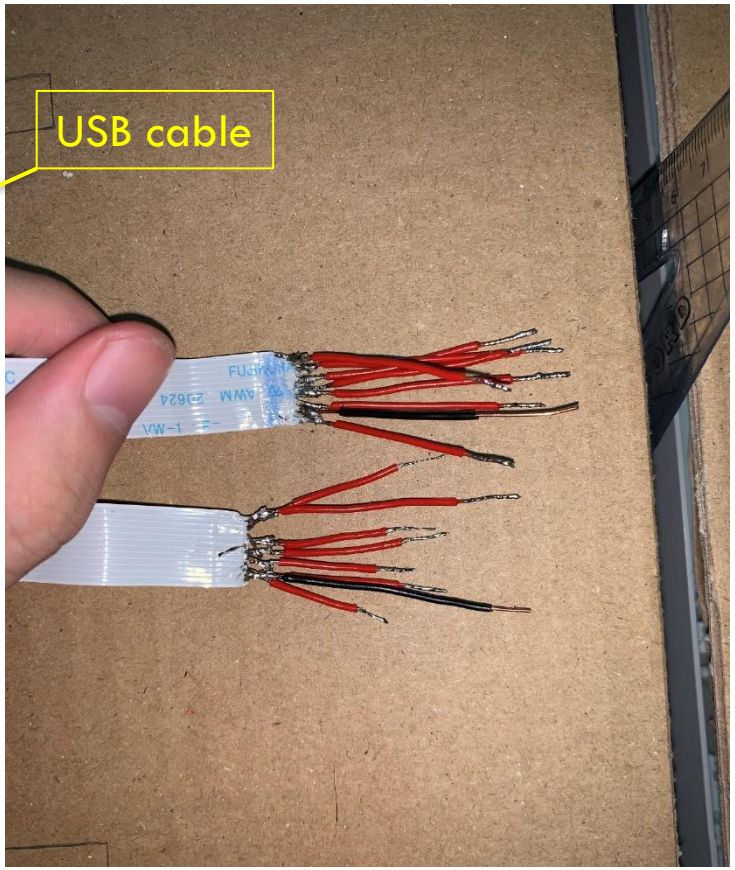
In summary, there are 3 main problem:

1. The Arduino board have to be connected to a computer.

2. The Servos Require a secondary power supply

3. Limited Space





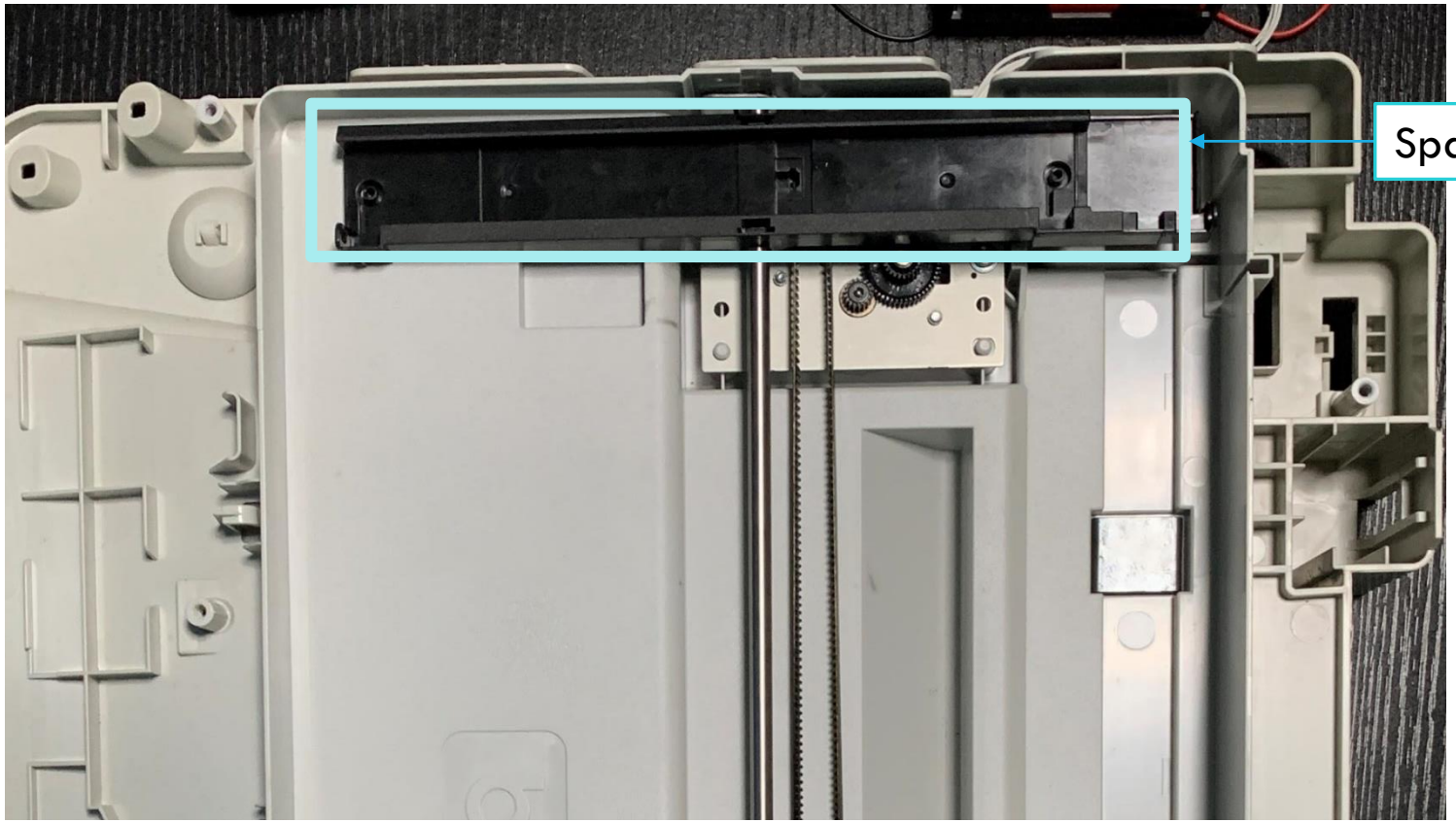
PHASE 5: PROBLEM 1

As this project requires the Arduino to be run by python, a USB cable must always be connected to the Arduino . However, since the platform is on a linearly moving pully, it means that the cable needs to be able to accommodate the changes in distance.

I have attempted using ribbon cables to overcome this difficulty; however, I do not have a port for it, and soldering it makes it fragile.

PHASE 5: PROBLEM 2

As **Phase 5** requires 2 SPT5410LV servos, they can draw up to 2 Amp of current, which cannot be supplied by an Arduino. If supplied solely by an Arduino, stuttering and spinning will occur. Hence, I decided to use two 18650 batteries, which will supply the necessary voltage (6.4V). However, the batteries are heavy and big which can be a safety hazard as well.



PHASE 5: PROBLEM 3

2 Servo motors, 1 Arduino microcontroller, 2 18650 batteries and a breadboard must fit on a small 20cm by 5cm spot. The need to balance the weight across the small space, as well as managing the cables, proved to be extremely difficult.

SUMMARY OF PHASE 5

I must put phase 5 on hold as, according to the Gannt chart, I will not be able to complete this project on time.

Nevertheless, I have learnt a lot from this attempt. While I am unable to fit the various part into the platform, I have managed to produce the code needed, which will be included in the annex section. Furthermore, a short video showing the code running will be uploaded to YouTube, under a separate link.

Electrical Principals, as well as mechanics applied, will still be relevant up till Phase 4.

ELECTRICAL PRINCIPALS

This project require the use of 17 LEDs, 4 micro switch, L298N stepper motor driver, 2 IR sensors, 1 buzzer, 1 LED strip, 1 servo, 1 LDR and an ultrasonic sensor. By using a micro switch, I have reduced the number of pins needed by 8.

Furthermore, I have combined the Arduino Uno with an Arduino Micro controller, which managed to provide me with all the input/output pins needed.

Power consumption calculations: Slide 45

Schematic: Slide 36 and 37

Total I/O pins used: 17 (uno) + 6 (micro)

FUNCTION OF COMPONENTS:

Component	Quantity	Function
Red LED	5	1. To inform the user if the carpark slot is occupied 2. To inform the user that he should stop his car [DISPLAY PANEL]
Yellow LED	7	1. To guide the user into his/her assigned parking lot 2. To inform the user that he should slow down his car [DISPLAY PANEL]
Green LED	5	1. To inform the user if the carpark slot is vacant 2. To inform the user that he should proceed into the lot [DISPLAY PANEL]
Servo Motor	1	1. To block car form entering the Carpark when it is full 2. To allow car to enter when there is lot in the Carpark 3. Allow cars to leave the Carpark
Push Button	1	1. Ensures that the stepper motor position is properly calibrated
LDR	1	1. Finds out the surrounding light level
LED Strip	3	1. When the light level is too low, LED Strip will activate to illuminate the Carpark

FUNCTION OF COMPONENTS:

Component	Quantity	Function
Boost Converter	1	1. To increase the voltage of DIGITAL OUT from 5V to 12V
Obstacle Sensor	2	1. To detect if a car is entering the Carpark 2. To detect if a car is leaving the Carpark
Buzzer	1	1. To alert the driver when he/she is too close to the edge of the Carpark
Ultrasonic Sensor	1	1. To detect the distance of the car, away from the edge
L298N	1	1. To drive the stepper motor
Micro Switch	4	1. To detect if the parking lot is occupied or filled

DECLARING WIRE COLORS

Standard wire colors are used, such as red for positive; black for negative; to ensure safety when manipulating with the electrical components. Other colors, which are not standard, will be defined in the table below.

Color	Usage
Yellow	Data cable
Blue	Connection between components
White	Connect Arduino Uno pins to YELLOW data cable
Red (Exception)	[Only for Display Panel] for red LED
Yellow (Exception)	[Only for Display Panel] for yellow LED
Green (Exception)	[Only for Display Panel] for green LED

RESISTOR (LED)

My prototype uses resistors to limit the current flowing into the LEDs.

From the datasheet (link found in annex) of the LEDs: I_{MAX}: 20mA V_{DROP}: 2.2V $R = \frac{V_s - V_f}{I_f}$

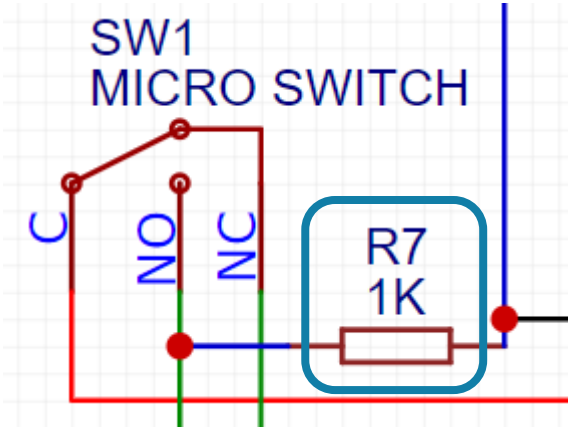
Hence, given that my input voltage is 5V, I found out that the resister should be higher than **140 ohms**, a power rating of at least **0.056W**. However, since this resistor value will be pushing the current to the peak limit, I will be using a resistor of at least **330 ohms** instead.

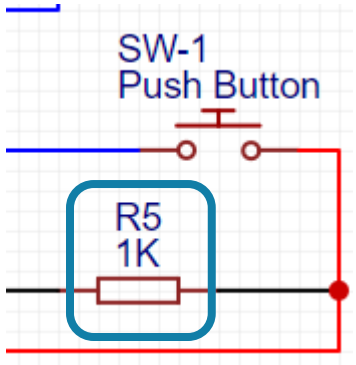
However, as for the guide lights part of the circuit, 3 to 6 LEDs could be connected. This means, 60mA to 120mA of current could flow at any time. The resistor should be higher than **46.7mA.** Since this resistor will be lighted frequently, I will be using a resistor of at least **110 ohms** instead. However, since the Arduino kit only comes with **1k** and **330 ohms,** I will be using 3 **330 ohms** resister in parallel in order to create and equivalent resistor of **110 ohms.**

RESISTOR (BUTTON)

My prototype uses resistors to act as a **pull-down** resistor for the push button, as well as the Micro Switch.

This pull-down resistor is connected to ground, providing a baseline ground voltage when there is no power connected. Hence the Arduino can detect when the switch is activated, increasing the voltage to 5V, to be read by an input pin. Hence, a resistor is needed as a load so that the current flowing to the ground will not cause a short-circuit.





Analog Input values:

```
826
1012
636
701
['✓', '✓', '✕', '✓']
Closest lot: Lot 1
```

```
1015
1012
832
871
['✓', '✓', '✕', '✕']
Closest lot: Lot 1
```

```
977
1012
782
810
['✓', '✓', '✕', '✓']
Closest lot: Lot 1
```

Before pull-down resistor is added (only highlighted switch is pressed)

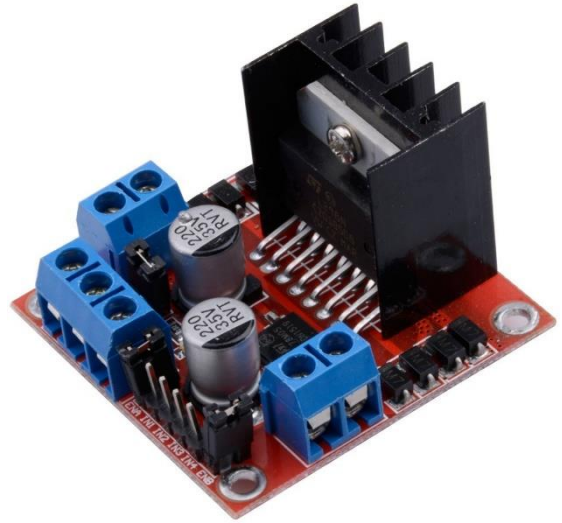
```
19
19
19
19
['✓', '✓', '✓', '✓']
Closest lot: Lot 1
```

```
21
1011
21
21
['✓', '✓', '✕', '✓']
Closest lot: Lot 1
```

```
14
14
15
14
['✓', '✓', '✓', '✓']
Closest lot: Lot 1
```

After pull-down resistor is added (only highlighted switch is pressed)

STEPPER MOTOR



I require the use of a stepper motor to linearly move the ultrasonic sensor to a precise spot. This can be easily done with a stepper motor and belt, which can be found in a scanner. The stepper motor I have obtained is M35SP-11NK.

From datasheet:

Resistance : 15 Ω Max Current: 360mA

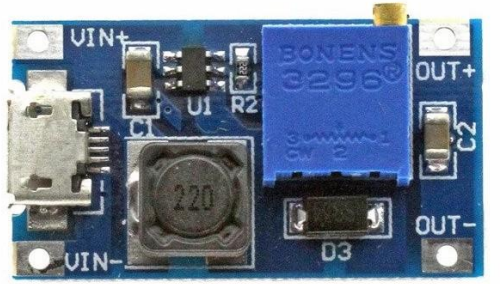
Hence, using ohm's law, I managed to find out that my input voltage should be 7.4V, considering a 2V voltage drop on the driver, L298N.

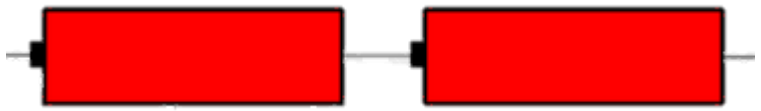
POWER SUPPLY

As an Arduino Uno could not supply the necessary Voltage and Current to the stepper motor driver and led strip, a separate power source is needed.

By using two 18650 batteries in series, I managed to create a voltage of 7.4V (6V – 8V) at 2.4Ah to power the stepper motor.

I have also considered using a step-up converter to increase the voltage. However, due to the constant power draw required by the stepper motor, the MT3608 converter is not suitable.





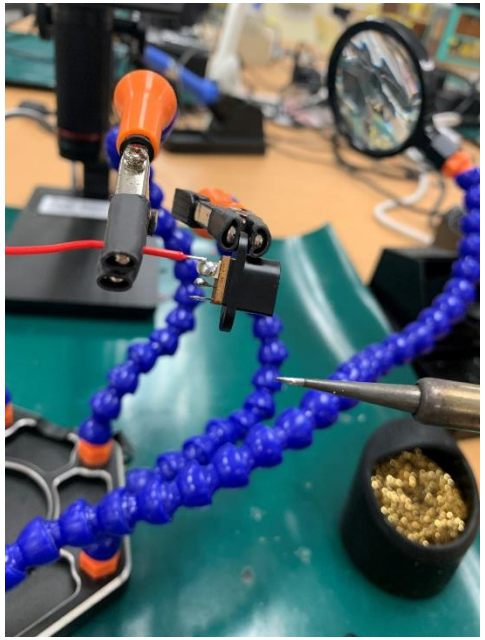
The converter is used to supply 12V to the LED strip instead

POWER SUPPLY

Even though the battery could theoretically last a day, a back-up power supply should be available. Hence, with a *12V/1A* AC to DC power supply, I managed to create a backup power source. The stepper motor is built to work with 24V.

The DC power adapter will be used during presentation, as it is safer and simpler to use than a battery.

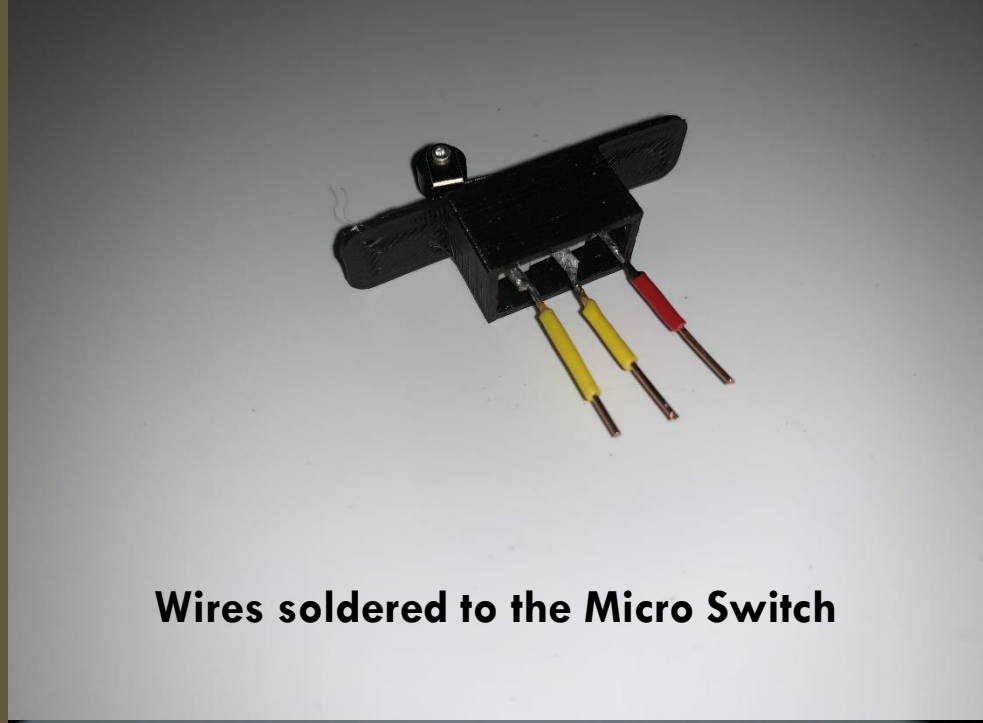
Finding the output pins



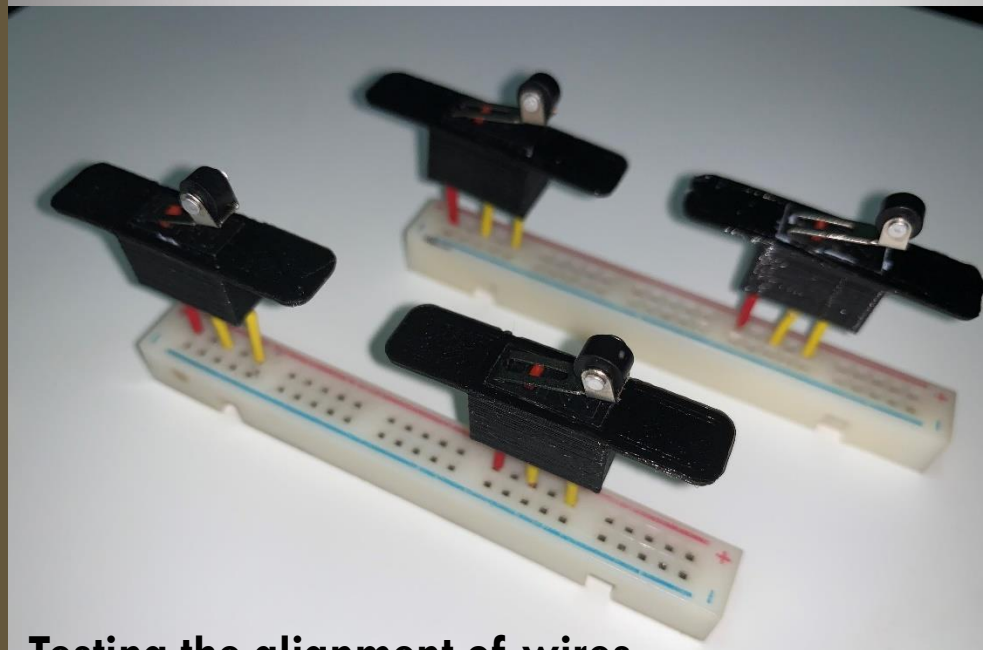
Soldering of wires

MICRO SWITCH WITH MOUNT

In order to integrate the Micro Switch into the model Cardboard, I need to create a 3D printed mount to hold the Micro Switch. However this presented a problem where the pins of the Micro Switch could not reach the Breadboard. Hence, I soldered wires to extend the pins of the Micro Switch.



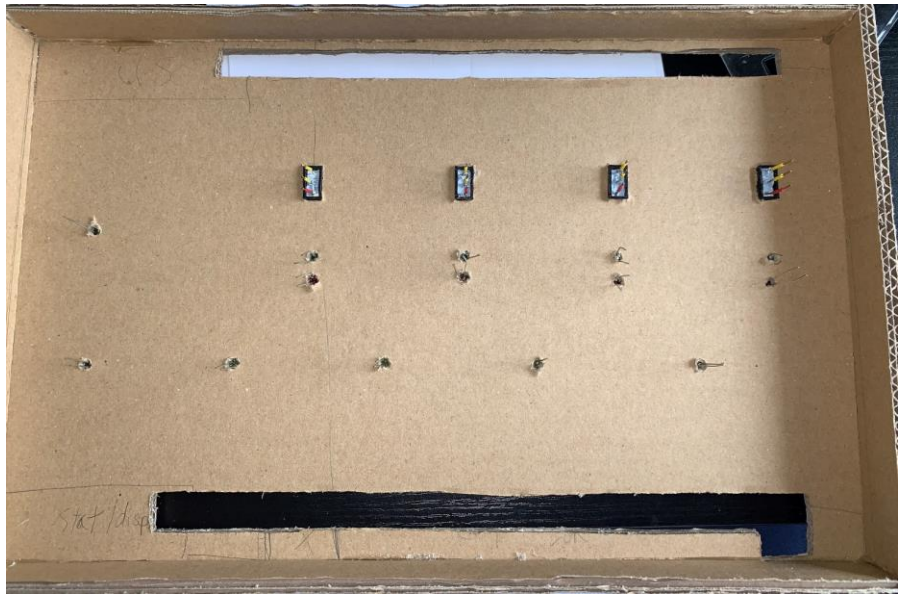
Wires soldered to the Micro Switch

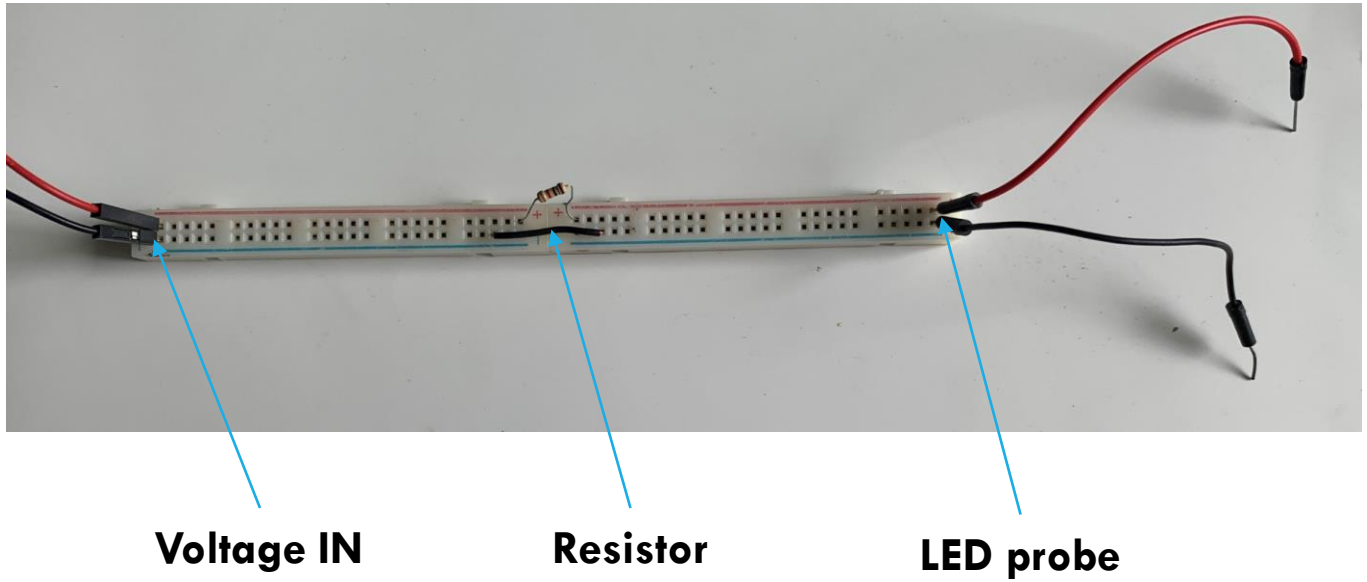


Testing the alignment of wires

DETERMINING POLARITY OF LED RESISTORS

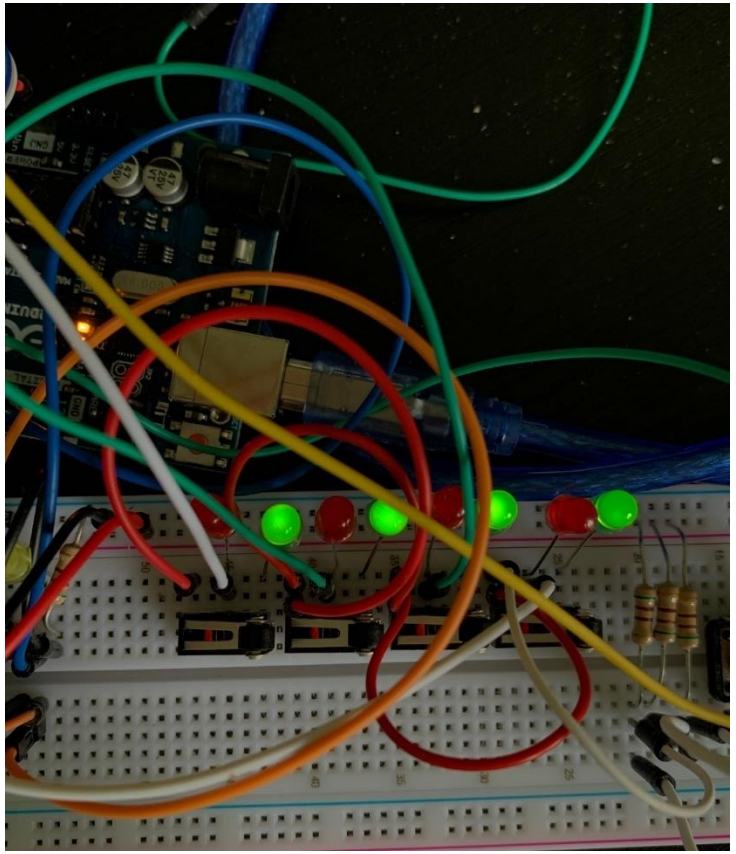
After transferring the LEDs from the breadboard to the cardboard prototype, I encounter a difficulty where the polarity of the resistors is unknown. Hence, I created a small circuit to power the LED, in order to find out the polarity of the LEDs.

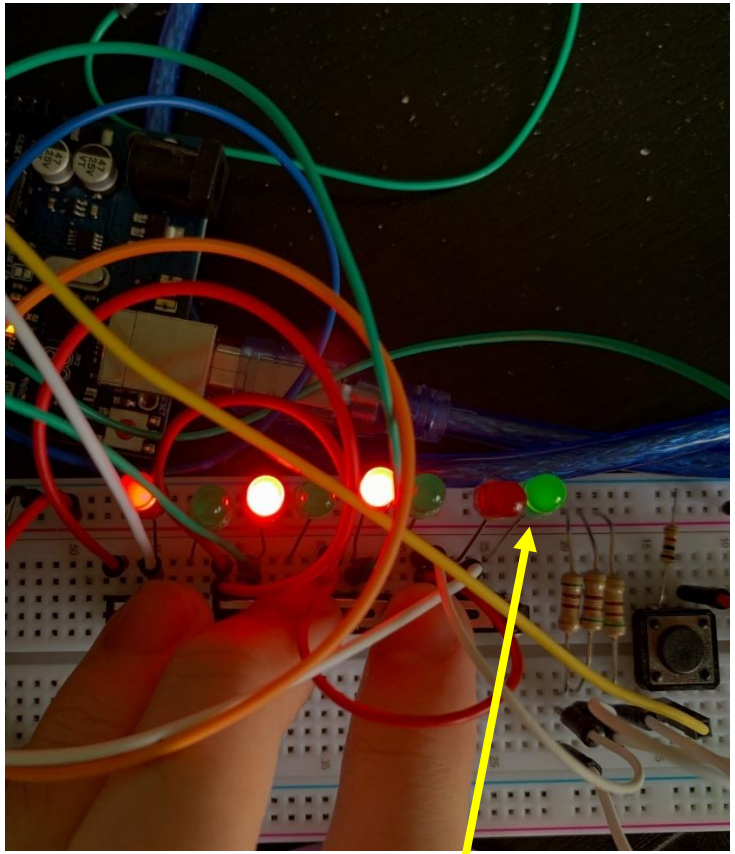




LED DISPLAY PANEL

Using the schematics, I have soldered 3 LEDs to 1 resistor, as part of my effort to be more environmentally friendly and reduce resource usage. Since only one LED will be activated at any time, I do not have to worry about the total resistance of the circuit changing. The left photo shows that top view, I have increased the distance between the positive wire and the ground, to prevent short-circuit. The right photo shows the side view, I have bent the wires to make them lie flat with the mount. This ensures that the mount can move within the cut-out made on the carpark





LED INDICATOR AND RESISTORS

During the testing of my electrical circuits, I realised that the green LED will become dimmer. This is because the red LED have a lower resistance, hence, resulting in a different current draw. I have decided to use two separate resistors, one (with higher resistance) to power the red LEDs and one (with lower resistance) to power the green LEDs.

Left: ALL SLOTS AVAILABLE

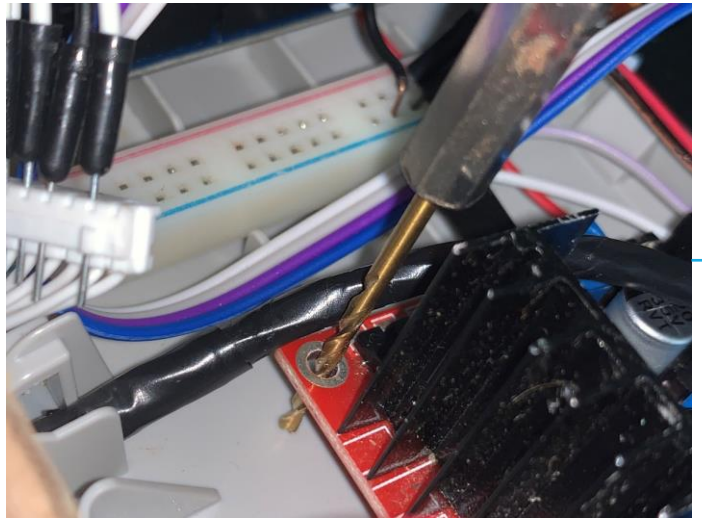
Right: SLOTS 1, 2 AND 3 OCCUPIED

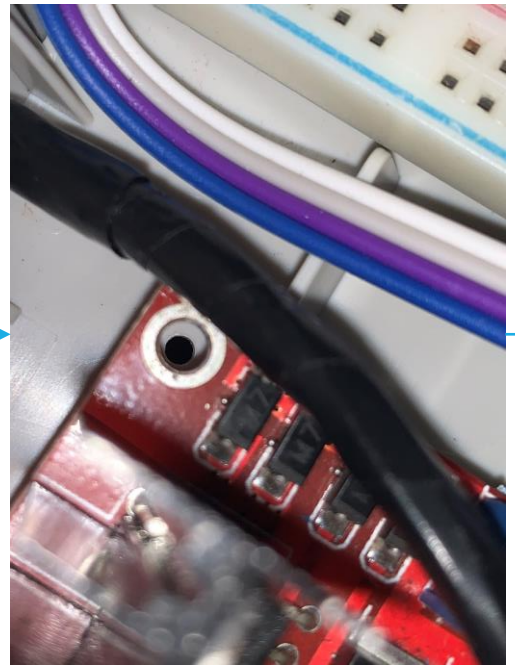
Note: The exposure of the camera is kept constant, hence any difference in lighting is due to the current of the circuit.

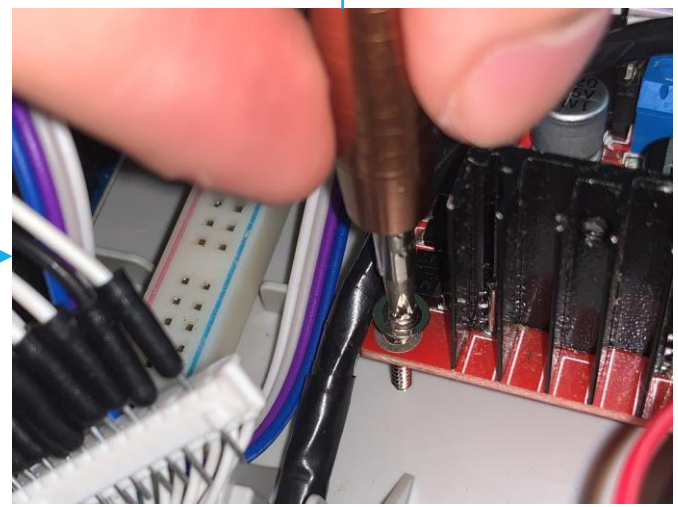
MOUNTING OF COMPONENTS

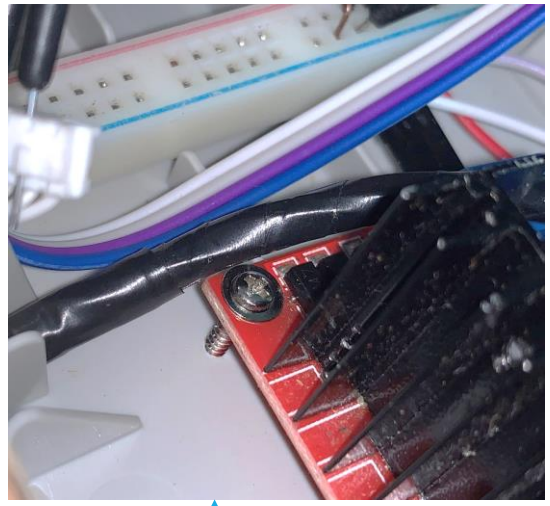
Screws are used to secure the electrical components. This is done by:

1. Placing the component on top of the surface
2. Drilling a hole through
3. Screwing the screw



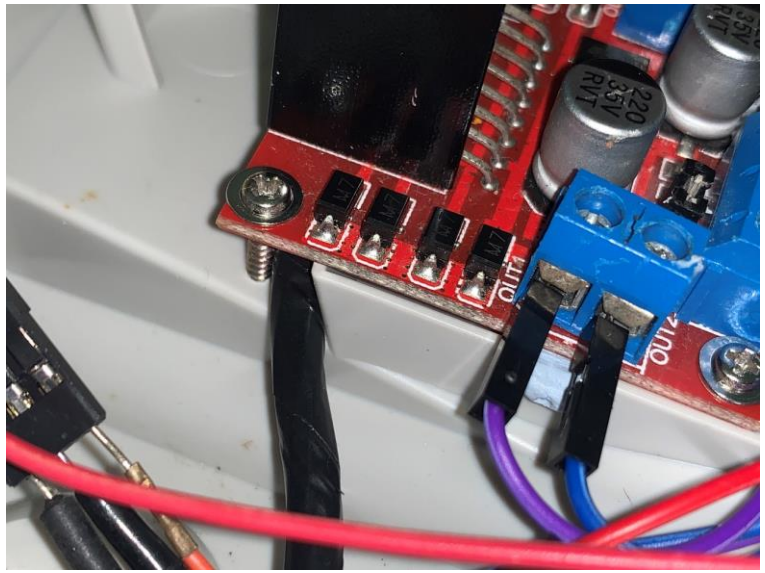


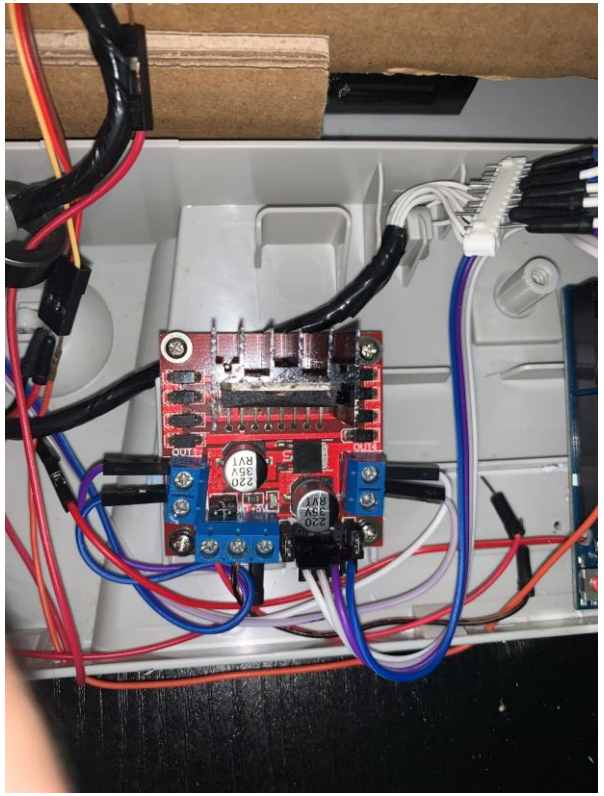


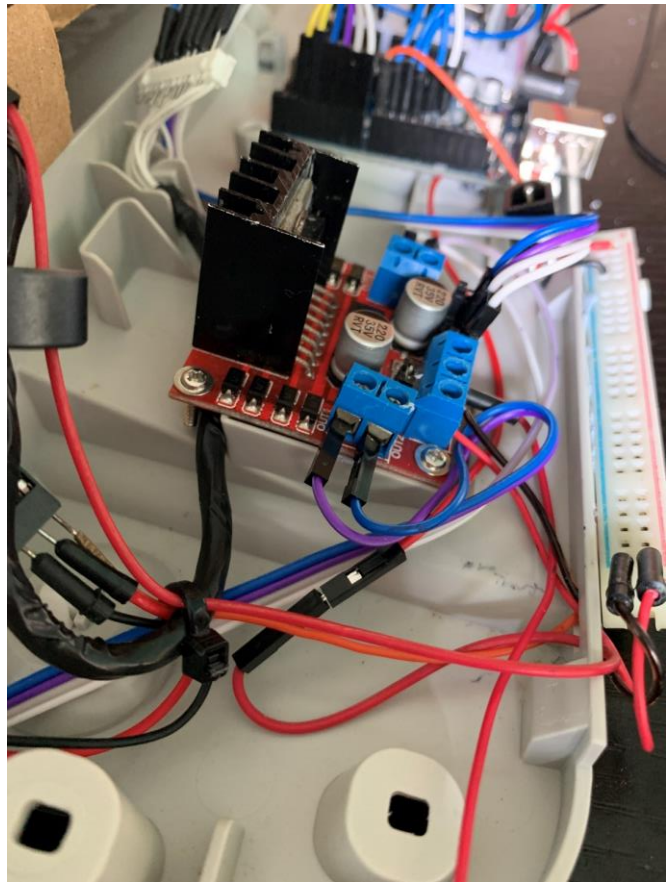


CABLE MANAGEMENT

Zip ties are used to organize wires together. Furthermore, holes are drilled to allow wires to pass in an organized fashion.

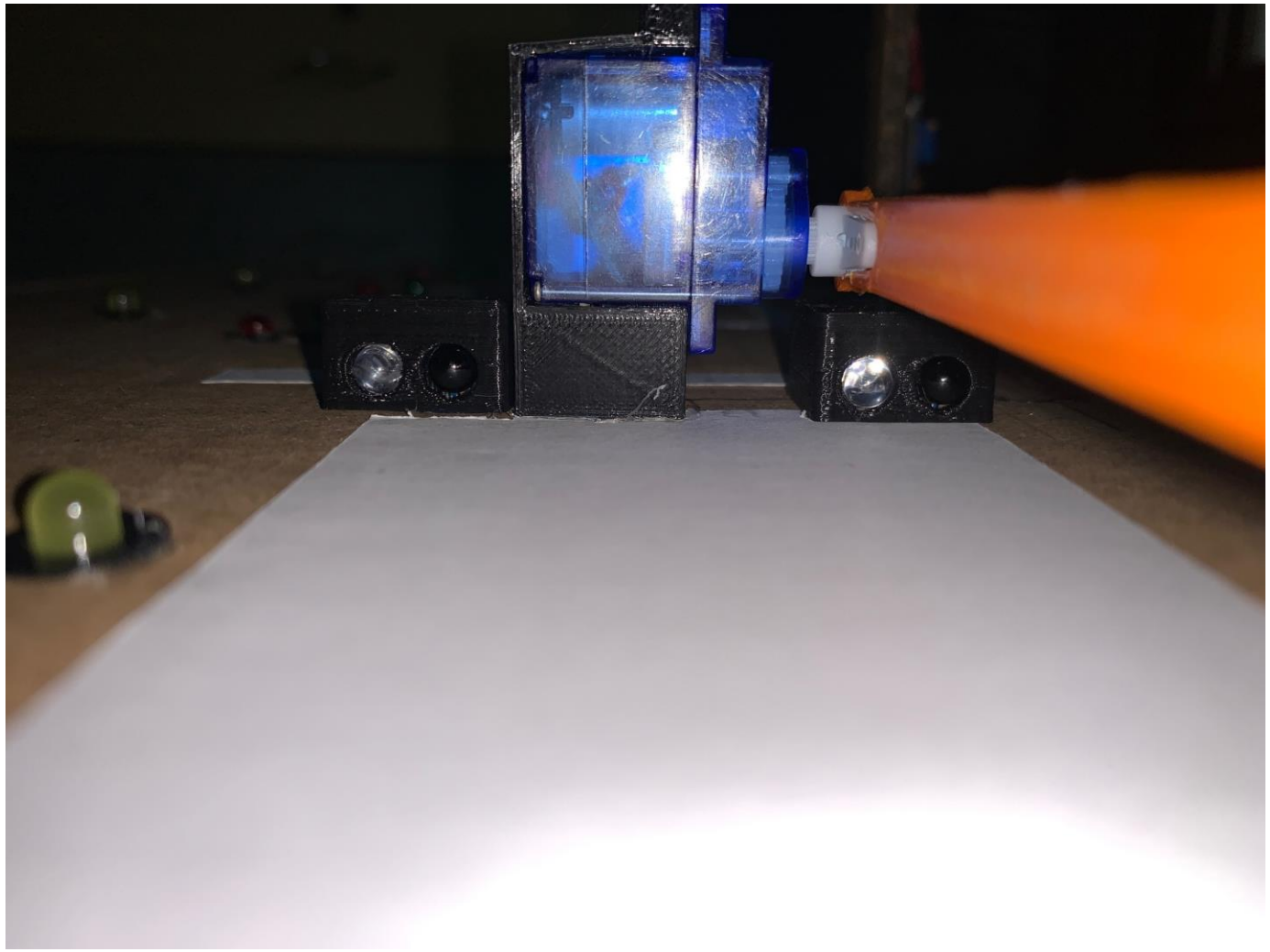


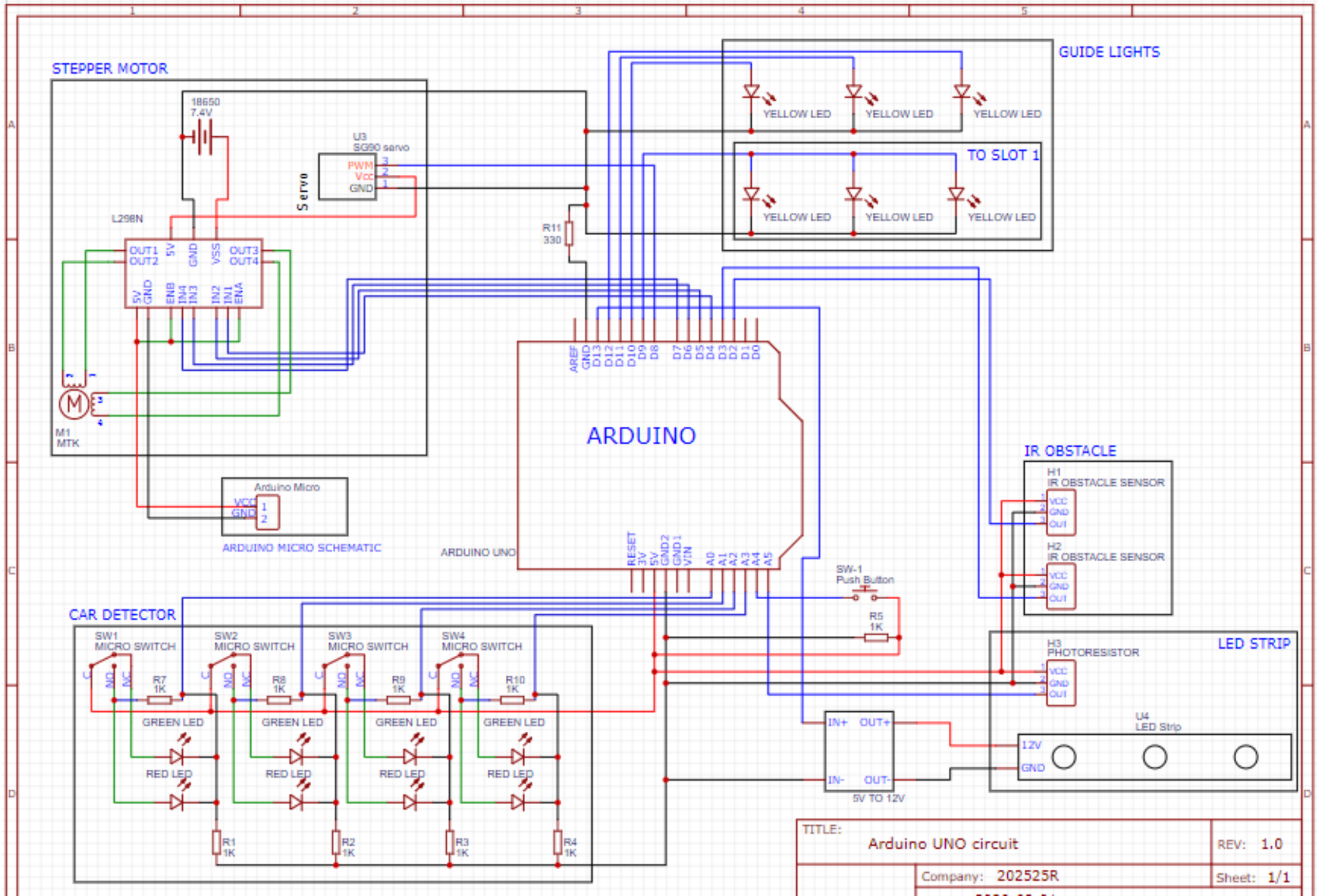


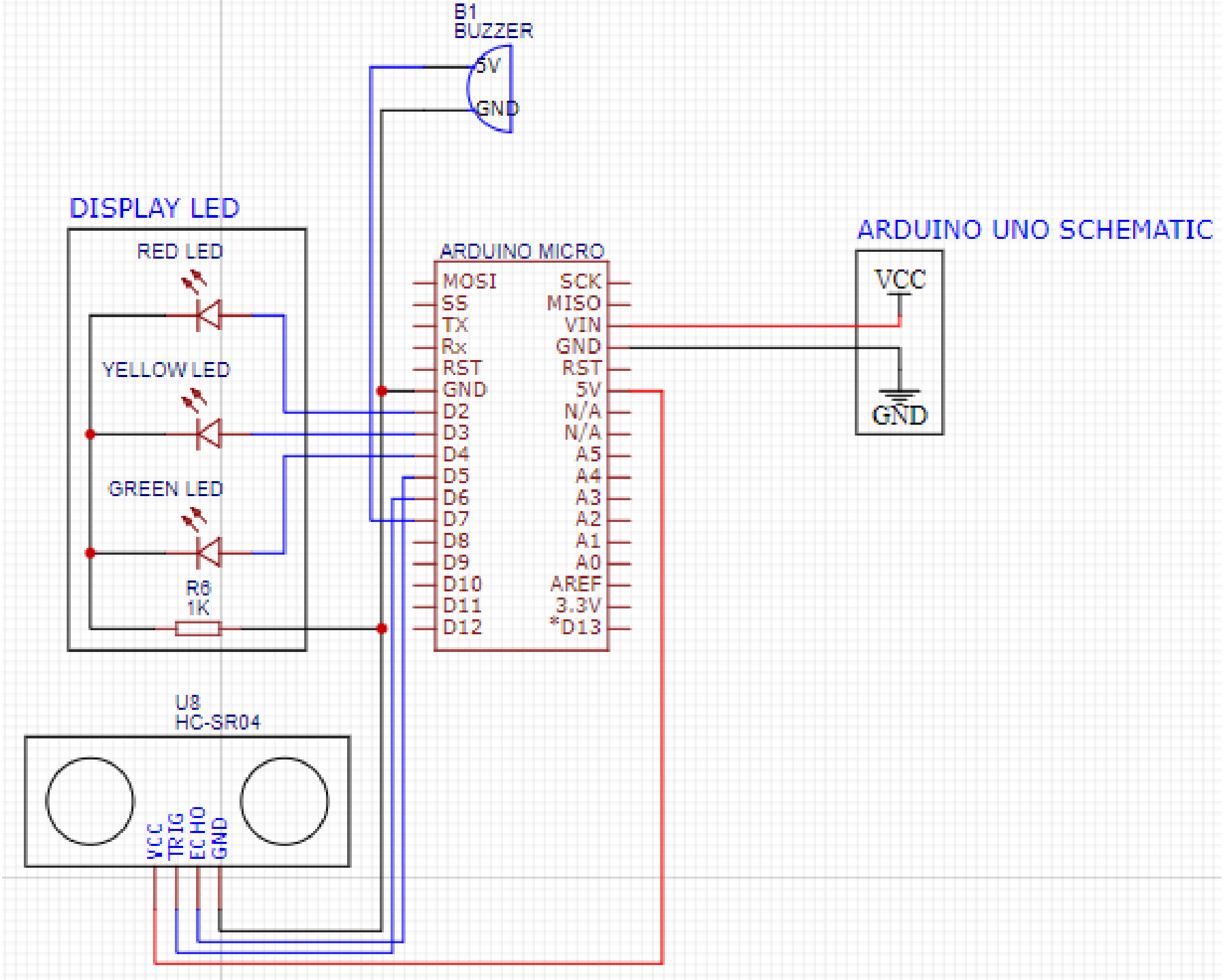


IR SENSORS

During the testing of the IR obstacle sensor, I realised that a reflective surface could help improve the accuracy of the IR sensor. Hence, I added a piece of paper underneath the sensor, in order to improve its accuracy.




STEPPER MOTOR
18650
7.4V
U3
SG90 servo
PWM
Vcc
GND
Servo
L298N
OUT1
OUT2
5V
GND
VSS
OUT3
OUT4
ENB
IN4
IN3
IN2
IN1
ENA
M1
MTK
R11
330
GUIDE LIGHTS
YELLOW LED
YELLOW LED
YELLOW LED
TO SLOT 1
YELLOW LED
YELLOW LED
YELLOW LED
ARDUINO
AREF
GND
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
Arduino Micro
VCC
GND
ARDUINO MICRO SCHEMATIC
ARDUINO UNO
RESET
3V
5V
GND2
GND1
VIN
A0
A1
A2
A3
A4
A5
IR OBSTACLE
H1
IR OBSTACLE SENSOR
H2
IR OBSTACLE SENSOR
SW-1
Push Button
R5
1K
CAR DETECTOR
SW1
MICRO SWITCH
SW2
MICRO SWITCH
SW3
MICRO SWITCH
SW4
MICRO SWITCH
R7
1K
R8
1K
R9
1K
R10
1K
GREEN LED
GREEN LED
GREEN LED
GREEN LED
RED LED
RED LED
RED LED
RED LED
R1
1K
R2
1K
R3
1K
R4
1K
H3
PHOTORESISTOR
LED STRIP
U4
LED Strip
12V
GND
IN+
OUT+
IN-
OUT-
5V TO 12V
TITLE:
Arduino UNO circuit
REV: 1.0
Company: 202525R
Sheet: 1/1


B1
BUZZER
5V
GND
DISPLAY LED
RED LED
YELLOW LED
GREEN LED
R6
1K
ARDUINO MICRO
MOSI
SS
TX
Rx
RST
GND
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
SCK
MISO
VIN
GND
RST
5V
N/A
N/A
A5
A4
A3
A2
A1
A0
AREF
3.3V
*D13
ARDUINO UNO SCHEMATIC
VCC
GND
U8
HC-SR04
VCC
TRIG
ECHO
GND

PRELIMINARY PSEUDOCODE

Note: This pseudocode is created on 25 July, before Project is finalized, hence final code will contain significant differences.

Reset stepper motor location
Reset servo location
Get closest slot
Wait for car to enter/leave

If car leave:

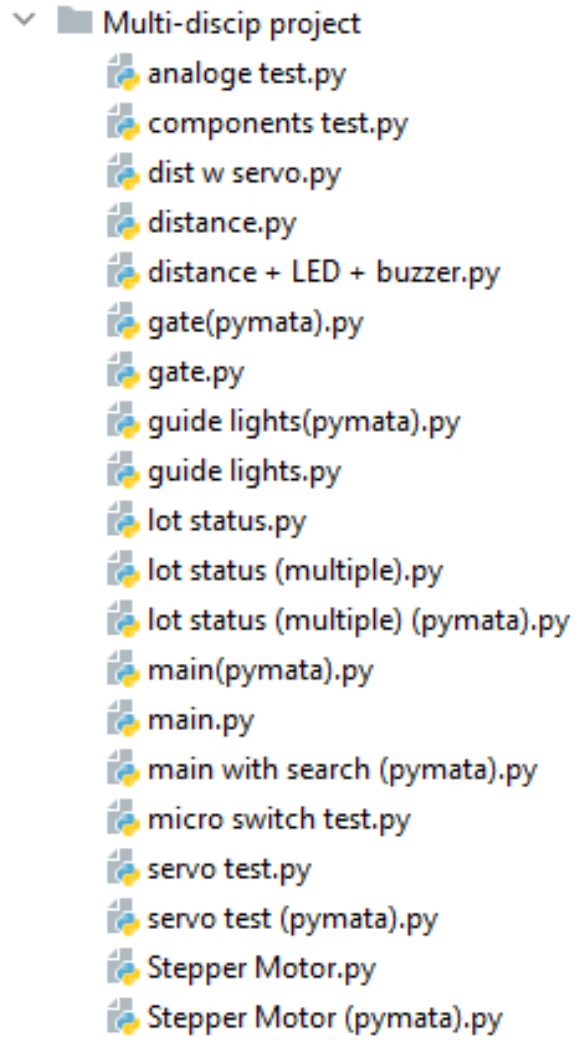
1. Find which slot is empty now
2. Light respective LED to exit (yellow)
3. Wait for IR sensor to activate
4. Servo open/close
5. Off yellow led
6. Update closest slot

If car enter:

1. Move stepper motor to slot
2. Light respective LED to lot (yellow)
3. Wait for slot button to activate
4. Off yellow led
5. Update slot status

PROGRAMMING

The coding of this project is done in several different functions. Due to my project needing a fast response time, pyFirmata make it difficult as every output must be handled on the computer's side, which will cause delays. This is evident when moving the stepper motor, as well as measuring distance using an Ultrasonic sensor. Hence, after much research, I have decided to migrate my codes to pyMata, which uses Arduino libraries, leaving the processing to the Arduino side.



PYMATA

With pyMata, I can offload tasks such as HC-SR04 ultrasonic sensor as well as driving of the stepper motor the Arduino, instead of driving it on python which will introduce a lot of delays with the sleep() function.

Furthermore, a customized version of StandardFirmata, called FirmataExpress is flashed into the Arduino board

- FirmataExpress adds support for:
  - HC-SR04 Ultrasonic Distance Sensors.
  - DHT Humidity/Temperature Sensors.
  - Stepper Motors.
  - Piezo Tone Generation.
  - Baud rate of 115200

USER INTERFACE

When a car enters the carpark, the code will prompt the user to enter his name/plate number. This information is used to help the user track where his car is parked at, preventing forgetful people from spending time searching for their vehicle.

```
Lot 1: ✓          Plate Number/Owner's Name: 0
Lot 2: ✓          Plate Number/Owner's Name: 0
Lot 3: ✓          Plate Number/Owner's Name: 0
Lot 4: ✓          Plate Number/Owner's Name: 0
Closest lot: Lot 1

Please enter your name or vehicle's plate number > Jian Quan
Please follow the Yellow Guide Lights to your Assigned Parking Lot.
Assigned Parking Lot: Lot 1

Lot 1: ✗          Plate Number/Owner's Name: Jian Quan
Lot 2: ✓          Plate Number/Owner's Name: 0
Lot 3: ✓          Plate Number/Owner's Name: 0
Lot 4: ✓          Plate Number/Owner's Name: 0
Closest lot: Lot 2
```

MECHANICS

A huge amount of mechanical calculations, as well as mechanisms (servos) are required to make this prototype. This range from free body diagrams, to shear and bending moment diagrams.

Components involving mechanics:

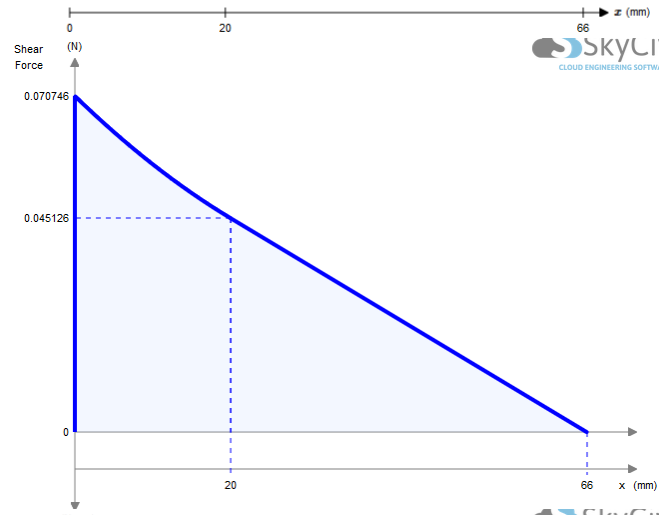
Carpark gate SFD and BMD, and torque experienced by servo motor	Stepper motor Gear Ratio and Speed	Work done by Arduino, battery, power usage (partially ELECTRICAL PRINCIPLES)	Moment and weight distribution of Ultrasonic sensor, distance display and Arduino Micro

1: CARPARK GATE

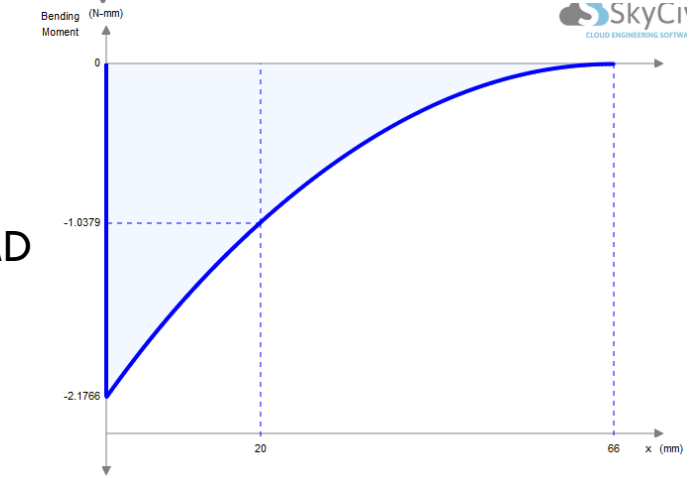
The length of the barrier must be at least 60mm, according to my CAD planning. Given that the servo driving the barrier have a torque of 245N-mm, I must ensure the material used and the thickness will not result in the barrier being too heavy.

FBD

SFD



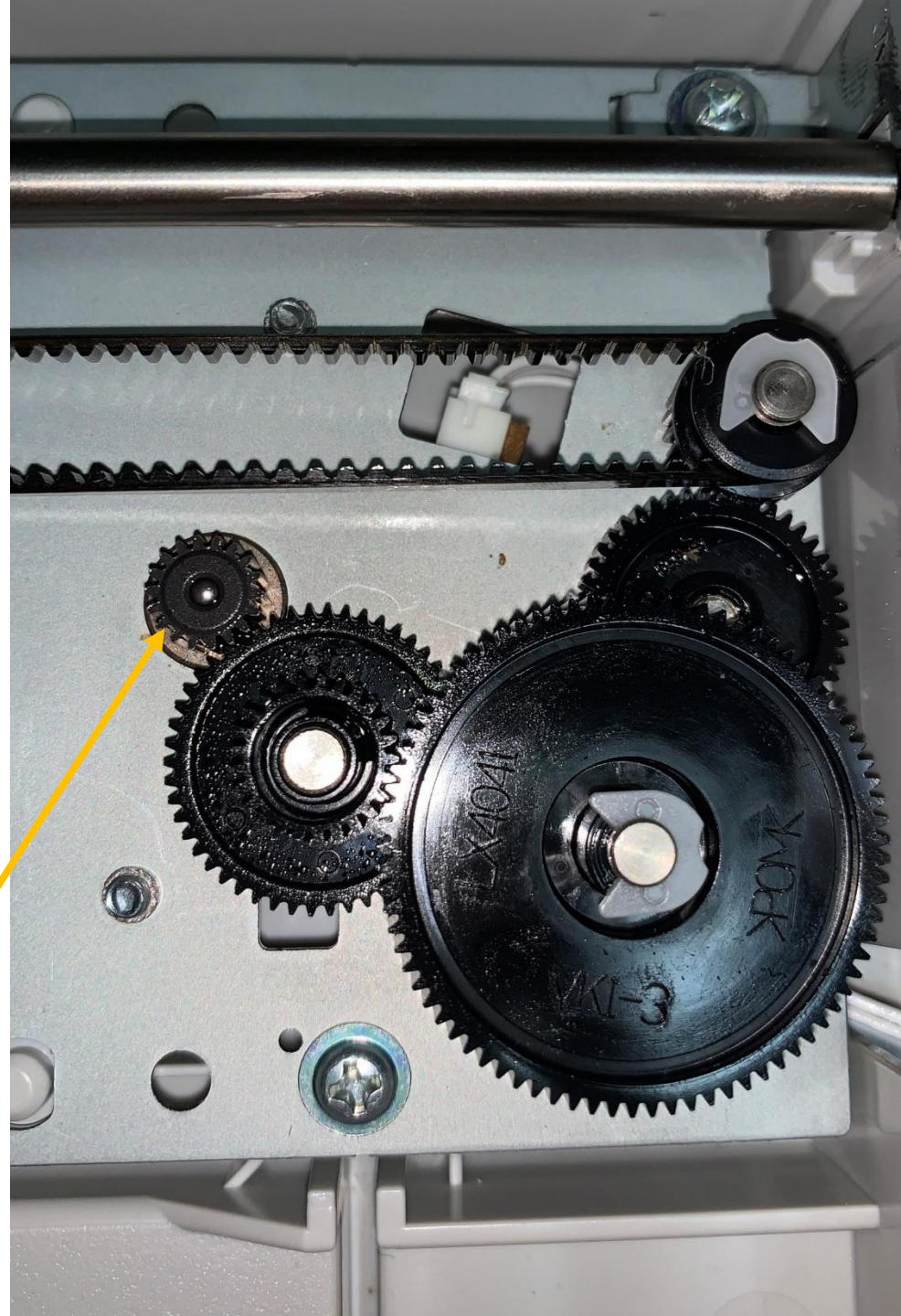
BMD



2: STEPPER MOTOR AND GEARS

The gearing included in the scanner have increased the torque, while reducing the speed. I will have to calculate the gear ratio of the gears and figure out how many steps of a stepper motor is needed to move the rod by a unit distance.

The max speed for 7.4V is **12cm/second**



3: WORK DONE BY ELECTRICAL COMPONENTS

Arduino Uno power draw USB (5V): 100mA (min) | 130mA (max) [Picture in Annex]
Arduino Micro power draw USB (5V): 10mA (min) | 10mA (max) [Picture in Annex]
Stepper Motor (7.4V): 44mA (idle) | 296mA (moving) | 404mA (hold) [Picture in Annex]

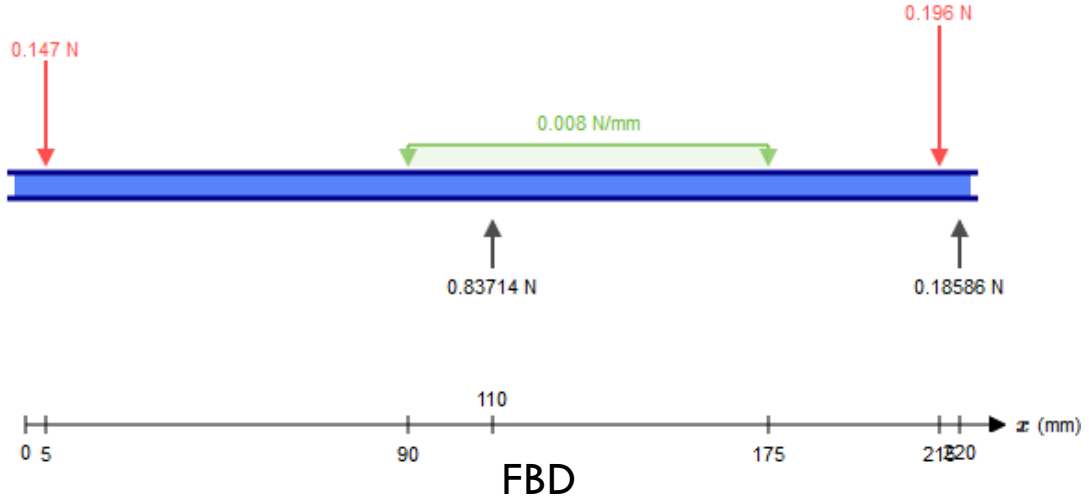
Idle time: 90% (idle), 5% (moving), 5% (hold) [stepper]
40% (on), 60% (off) [LED]

Average Power: (0.9)(7.4V)(0.044A) + (0.05)(7.4V)(0.269A) + (0.05)(7.4V)(0.404A)
(stepper) = 0.542W
(LED & arduino) (0.6)(5V)(0.1A + 0.01A) + (0.4)(5V)(0.13A + 0.01A) = 0.61W
Total Avg Power = 1.152W + 20% heat loss = 1.3824W

2.4 Ah, 7.4V = 17.76 Wh 17.76 ÷ 1.3824 = 12.84 hours / charge

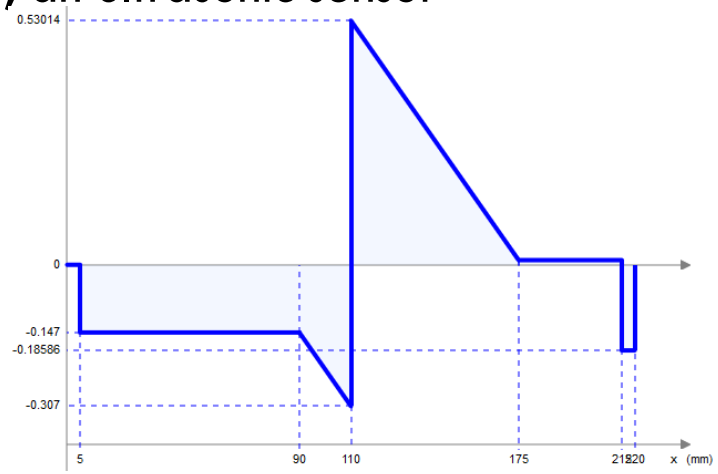
4: MOMENT AND WEIGHT DISTRIBUTION

The size of the platform (which the pulley pulls) is only 32mm by 220mm, which must carry a breadboard containing Arduino Micro, an ultrasonic sensor and a display panel containing 3 LEDs

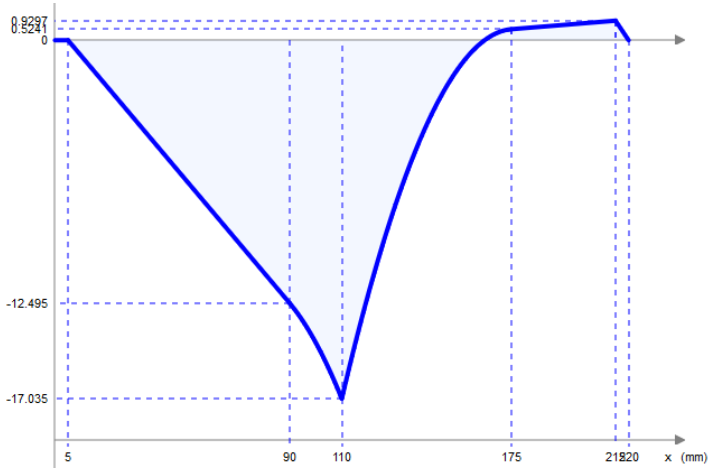


FBD

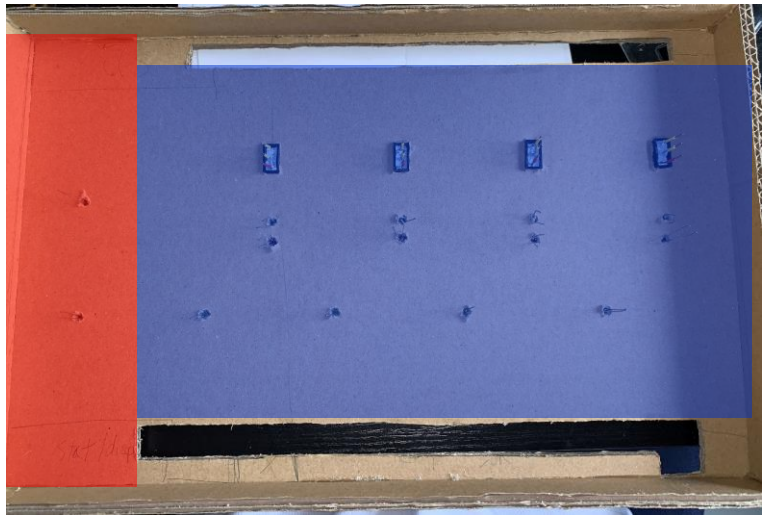
SFD



BMD



CLEARANCE MAP



Key:

RED – 5cm clearance

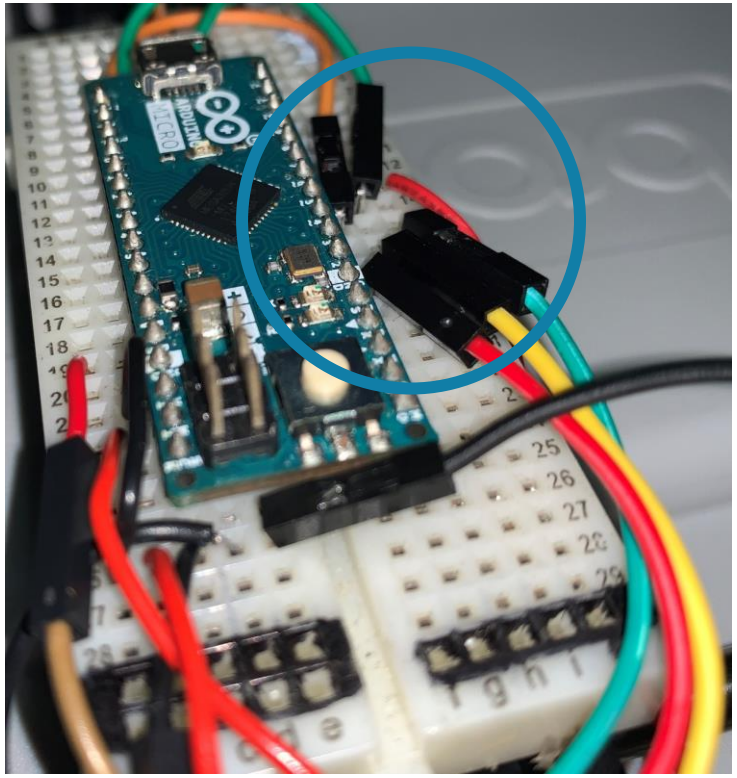
BLUE – 1.5cm clearance

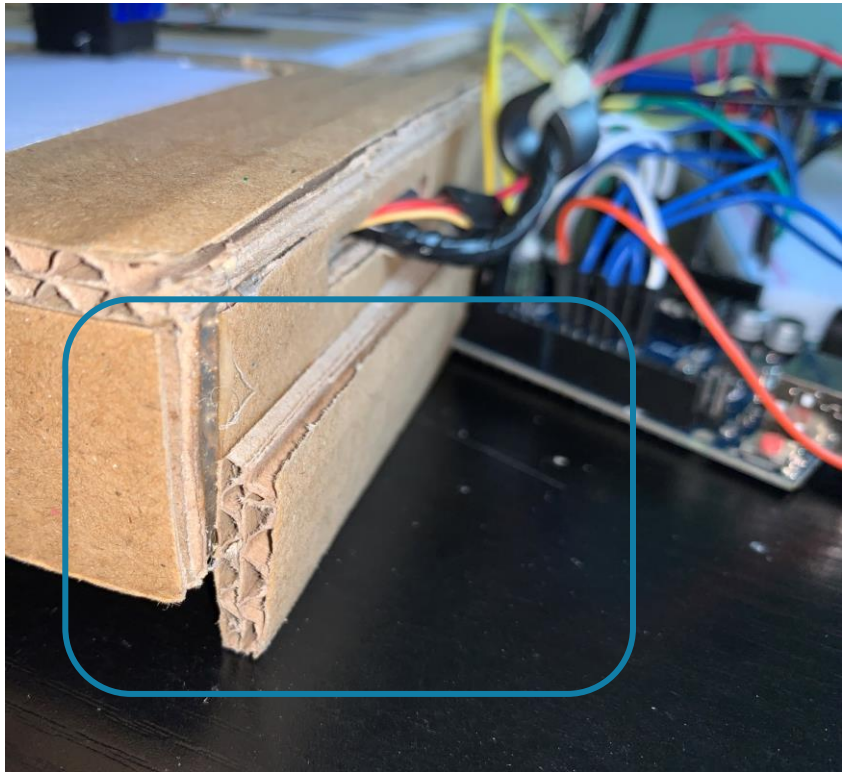
Picture of wires

CLEARANCE SPACE BETWEEN PLATFORM AND MODEL

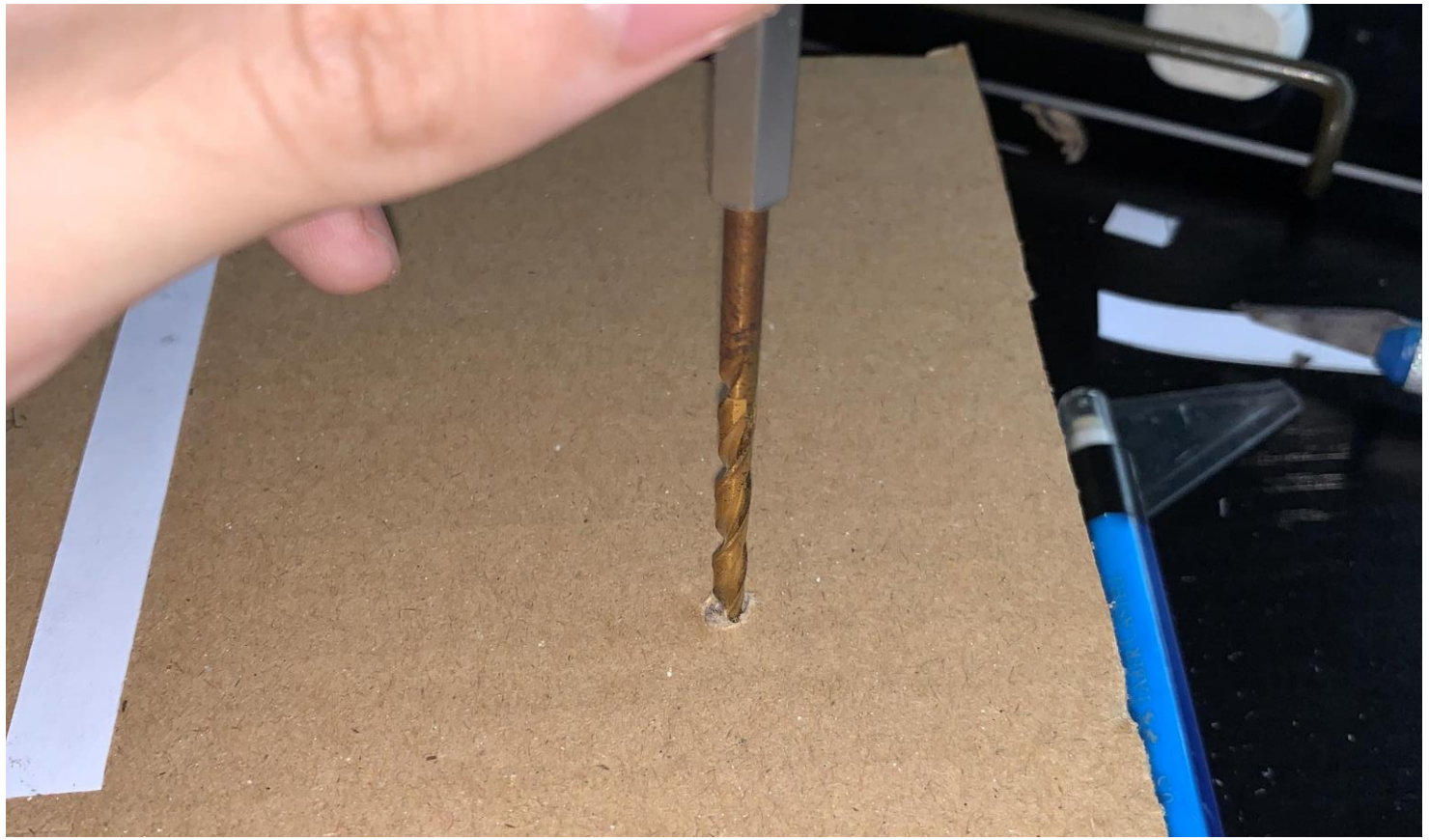
After testing with all the components soldered onto the cardboard model, I realized that I need to increase the clearance between the platform and the model by 4cm. This is done in 2 ways. Firstly, I bent the pins flat, to reduce the height profile of the breadboard. Furthermore, extensions to the cardboard are added to ensure enough clearance room

Pins Bent





Extensions



CUT-OUTS ON CARDBOARD FOR ELECTRONICS

When I tried to drill holes through the cardboard, I realized that the speed of the drill will cause extreme inaccuracies. Hence, I decided to hand drill the holes as shown in the picture. This ensures that the holes are accurately sized, and accurately positioned.

TESTING OF ULTRASONIC SENSOR

To find the best distance away from the ultrasonic sensor for the car to park, I used the microcontroller to print out the value away from the detector. From this reading, I can conclude that the minimum distance should be 3 cm, as it is the minimum point of the readings. Hence, I have set a buzzer to alert the user if he/she goes closer than 3 cm, so that he will not crash onto any obstacle.

```
Distance: 18.92 cm
Distance: 18.94 cm
Distance: 16.88 cm
Distance: 15.74 cm
Distance: 14.98 cm
Distance: 13.24 cm
Distance: 12.29 cm
Distance: 11.22 cm
Distance: 10.25 cm
Distance: 9.15 cm
Distance: 8.50 cm
Distance: 7.16 cm
Distance: 5.58 cm
Distance: 4.40 cm
Distance: 3.35 cm
Distance: 3.04 cm
Distance: 3.06 cm
Distance: 8.72 cm
Distance: 18.94 cm
Distance: 18.94 cm
Distance: 18.94 cm
```

Readings

3D PRINTING

I have 3D printed a Servo arm, a LED display panel and an ultrasonic sensor holder to improve the durability. Unfortunately, the LED display panel model that I created have a slot too small to hold a LED. Due to the environmental impact of reprinting the model, I widened the slot via drilling instead.

Cardboard Model

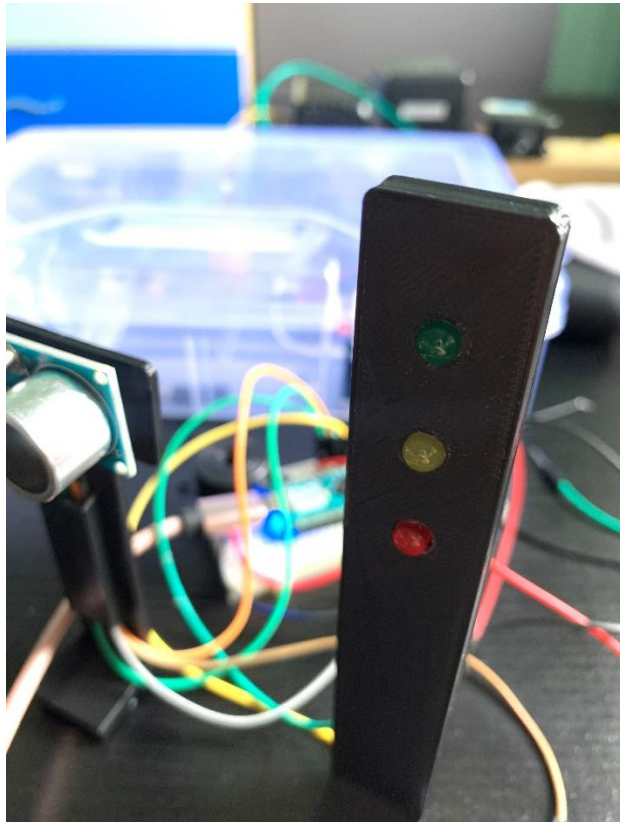
Drilling 3D Print

Printed Model

3D PRINTING - DISPLAY LED

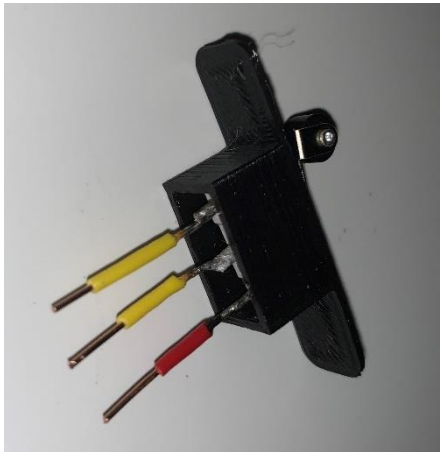
When the surroundings are bright, the LEDs will look no different from a lighted one. This is because the back of the LED is translucent, and hence will allow light to pass through. I fixed this issue my adding an opaque covering at the back of each LED.

Unpowered, only red LED is covered

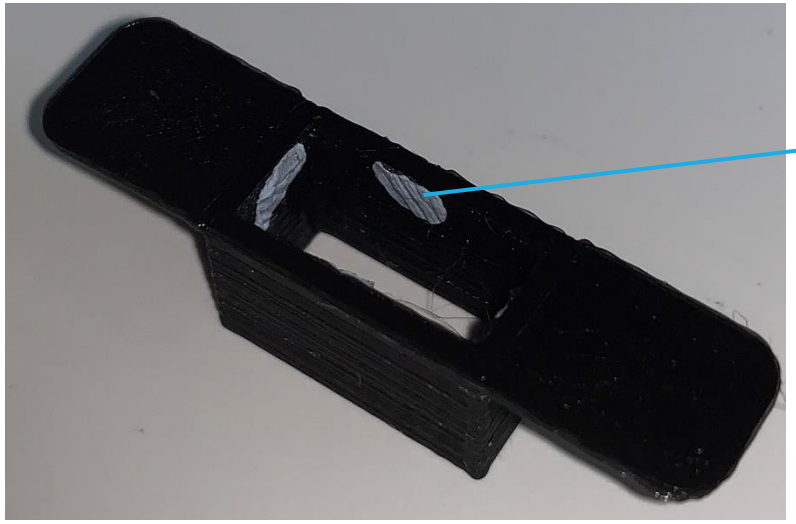


Unpowered, only all LEDs are covered

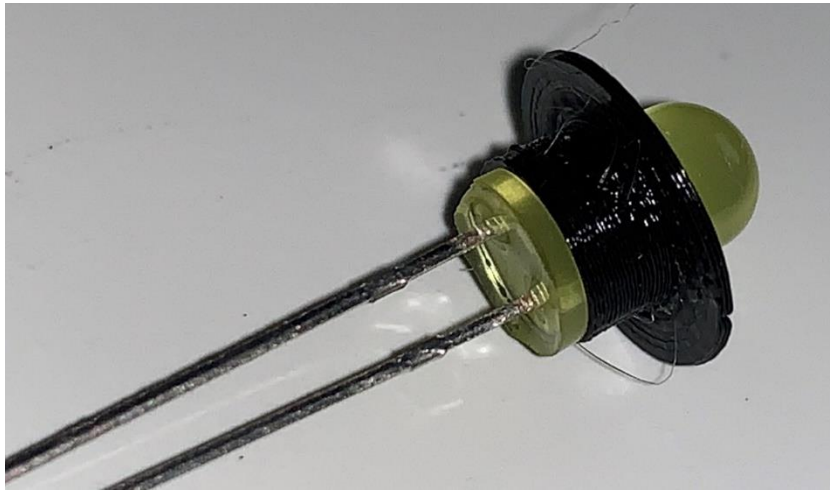
3D PRINTING – ELECTRONICS MOUNT



Micro Switch Mount



Bluetack to reduce allowance



LED Mount

During testing of the Micro Switch Mount, I realized that the force placed on the switch will cause the switch to sink into the mount. I have circumvented this problem by reducing the allowance distance between the mount and the switch, hence allowing the Micro Switch to **resist more force** before sinking.

This method is used for loose LED mount as well.

BOM (ADDITIONAL COMPONENTS)

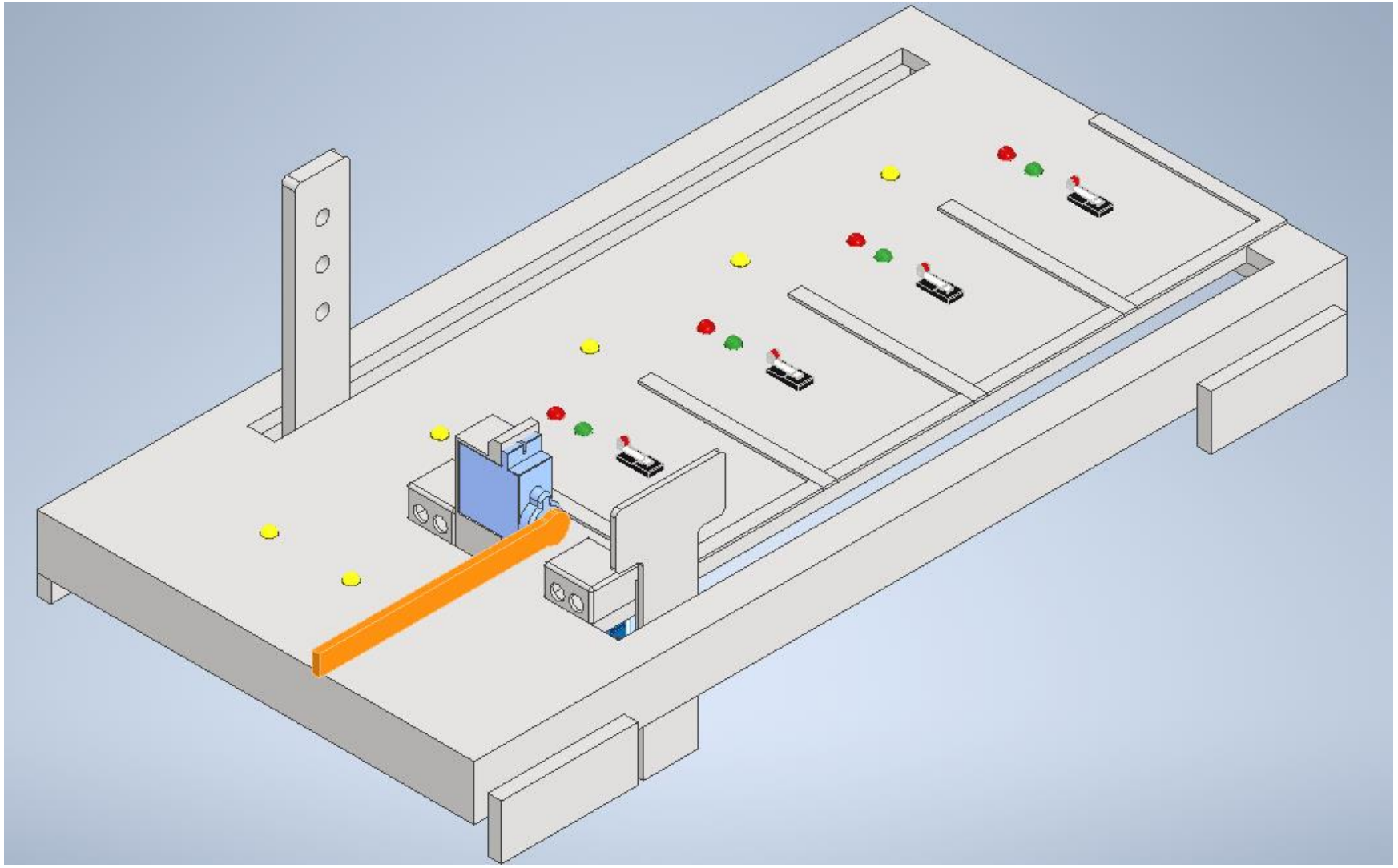
Name	Cost	Quantity	Total
Ultrasonic sensor	$0.82	1	$0.82
Micro Switch	$0.13	4	$0.52
L298N driver	$6	1	$6
Arduino Micro	$4 (Nano alternative)	1	$4
18650 battery holder	$1.75	1 (5 set)	$0.7 (2 used)
LED strip	$3.97	1 (5m)	$0.08 (0.1m)
MT3608	$2.91	1 (5 set)	$0.58
		Total cost (non-provided)	$12.70

Blue refers to borrowed items

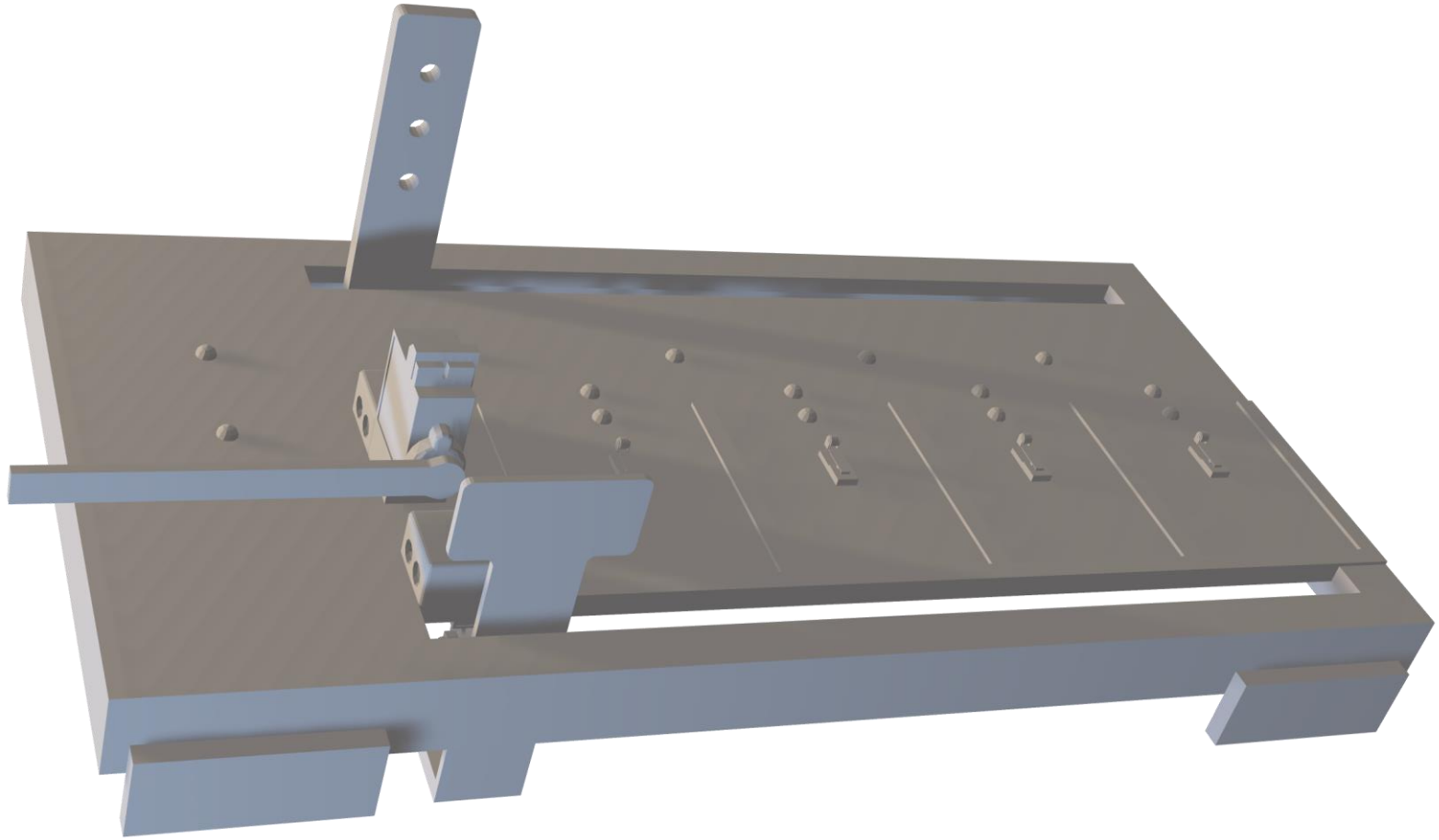
Green refers to items that I have

Receipts for purchased items can be provided, upon request

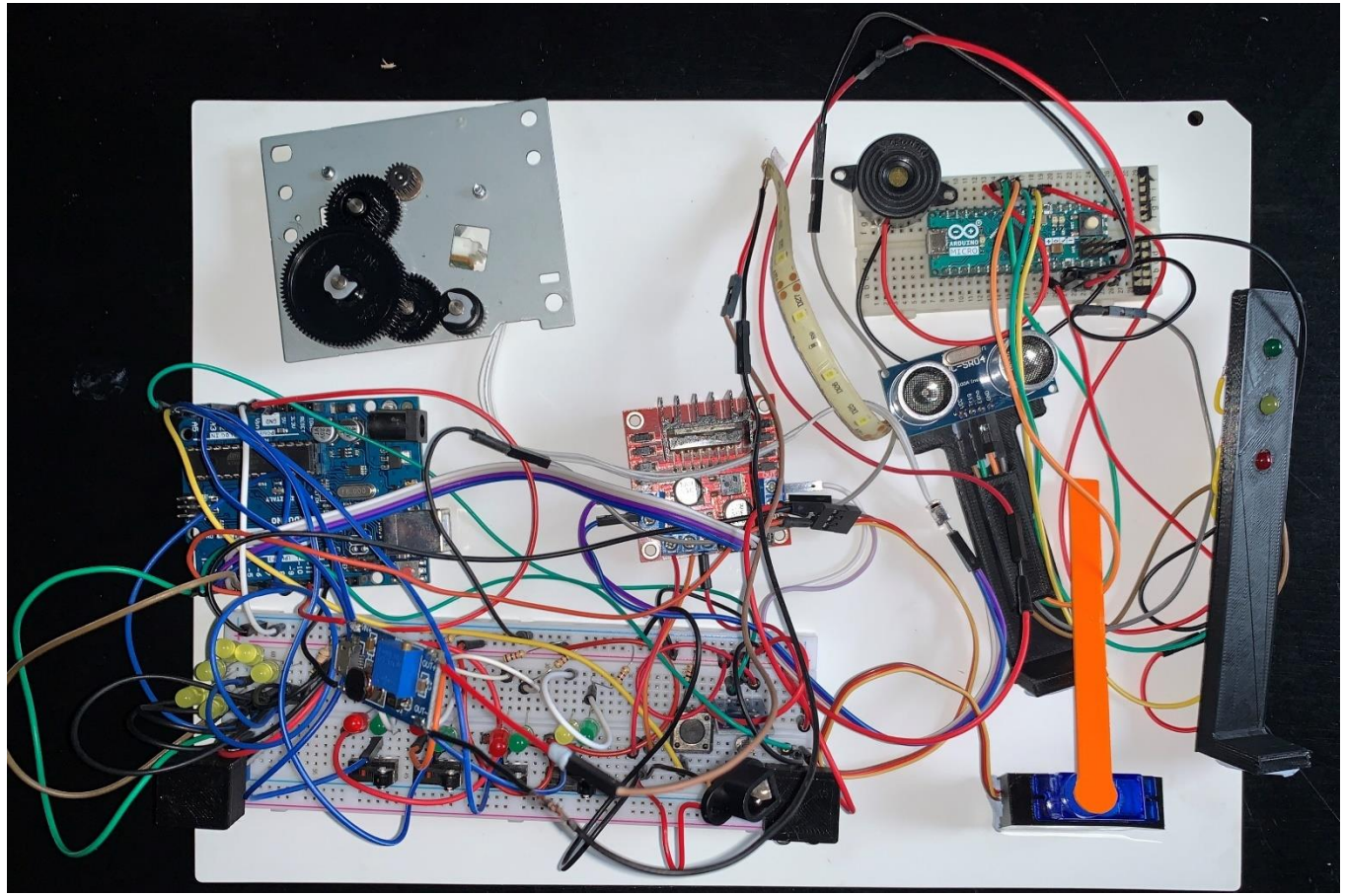
FINAL SKETCH

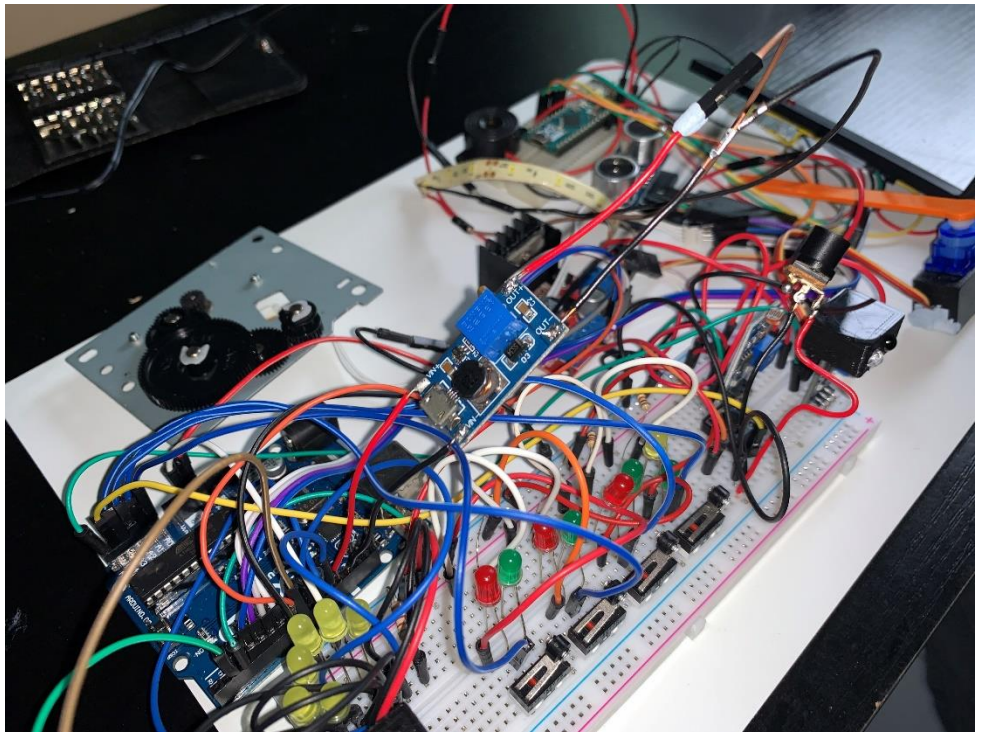


3D MODEL OF FINAL SKETCH

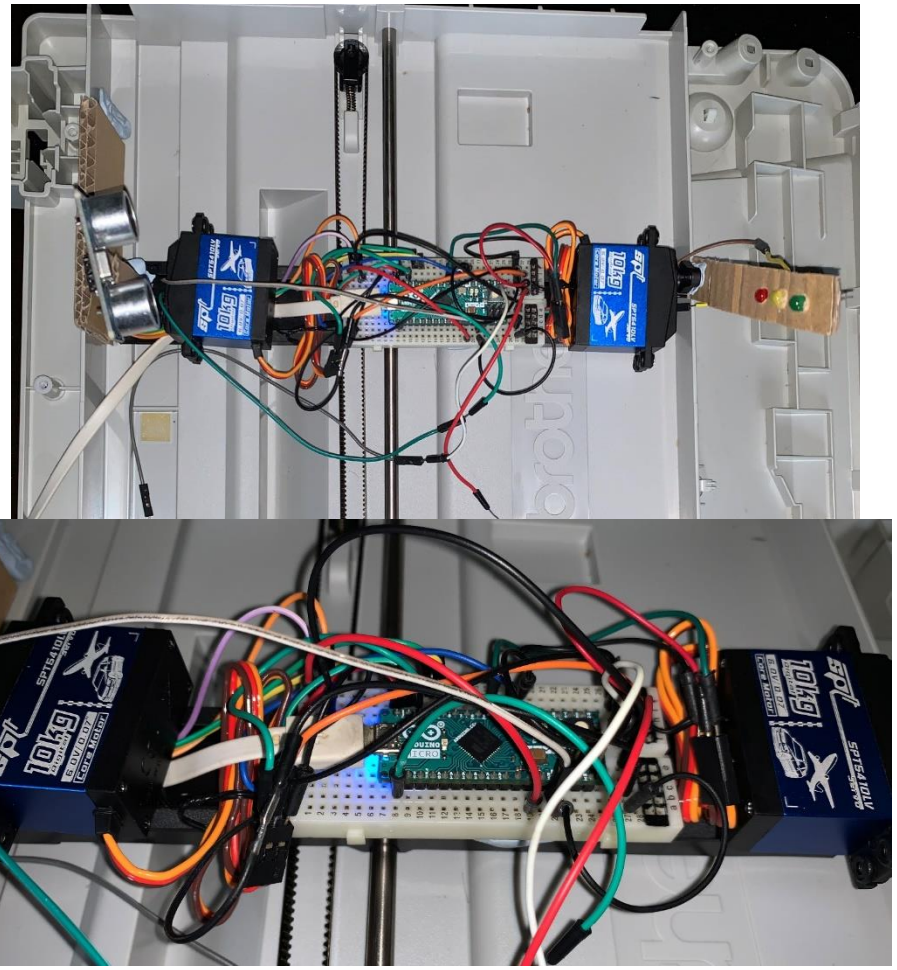


FINAL PROTOTYPE BEFORE INSTALLATION

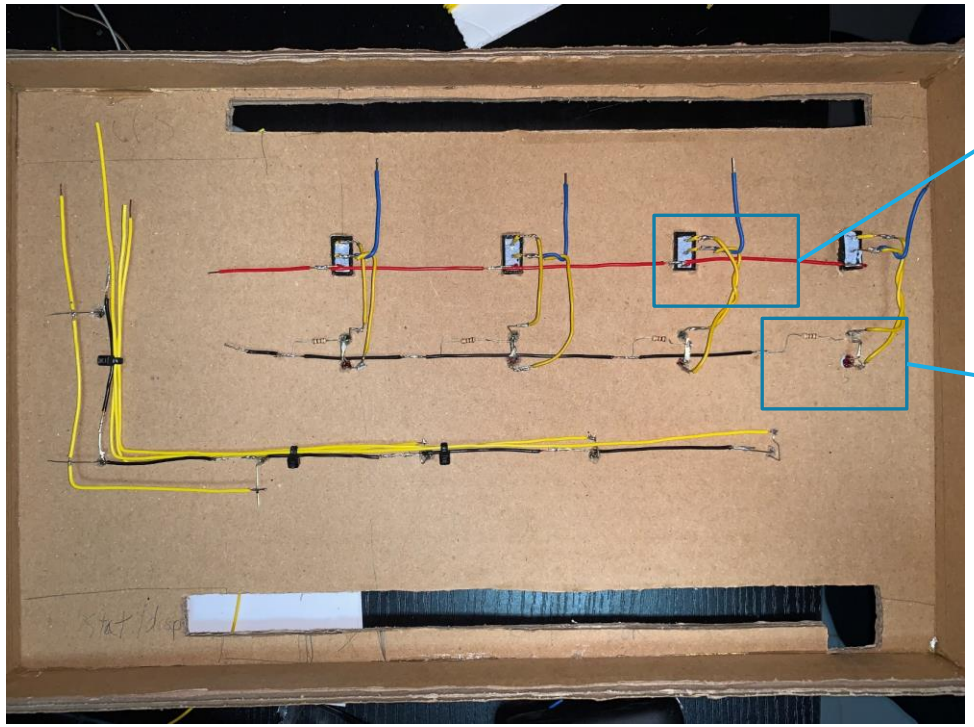


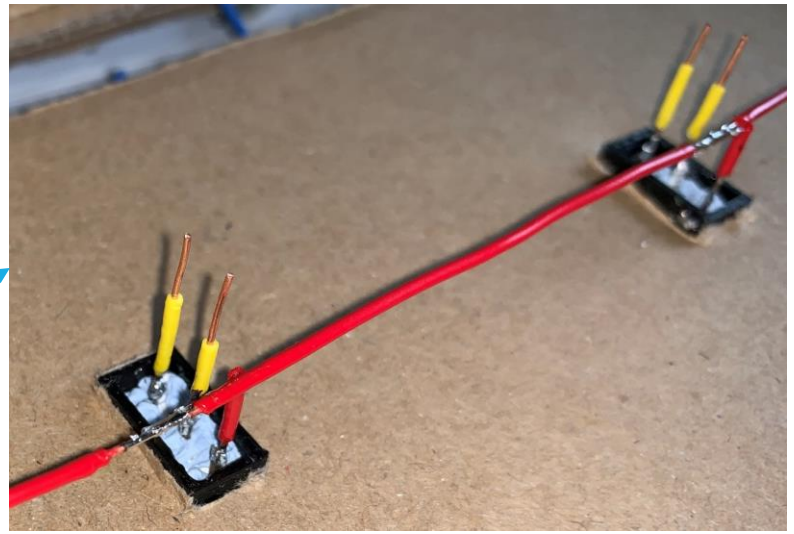


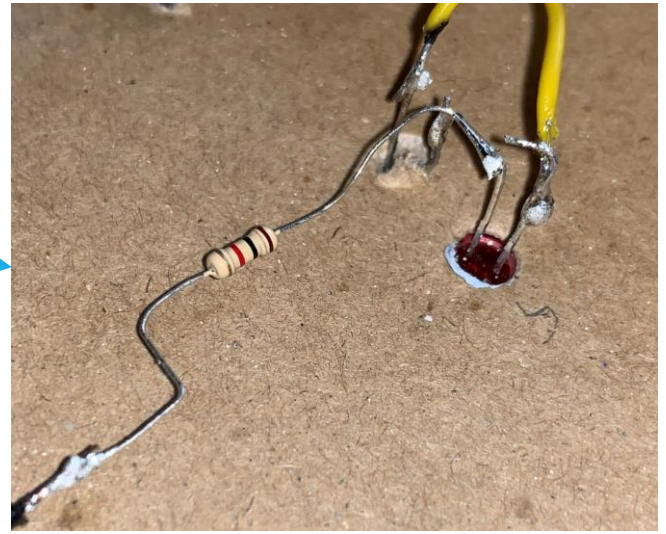
PHASE 5 (DISCONTINUED) PROTOTYPE



SOLDERING OF COMPONENTS

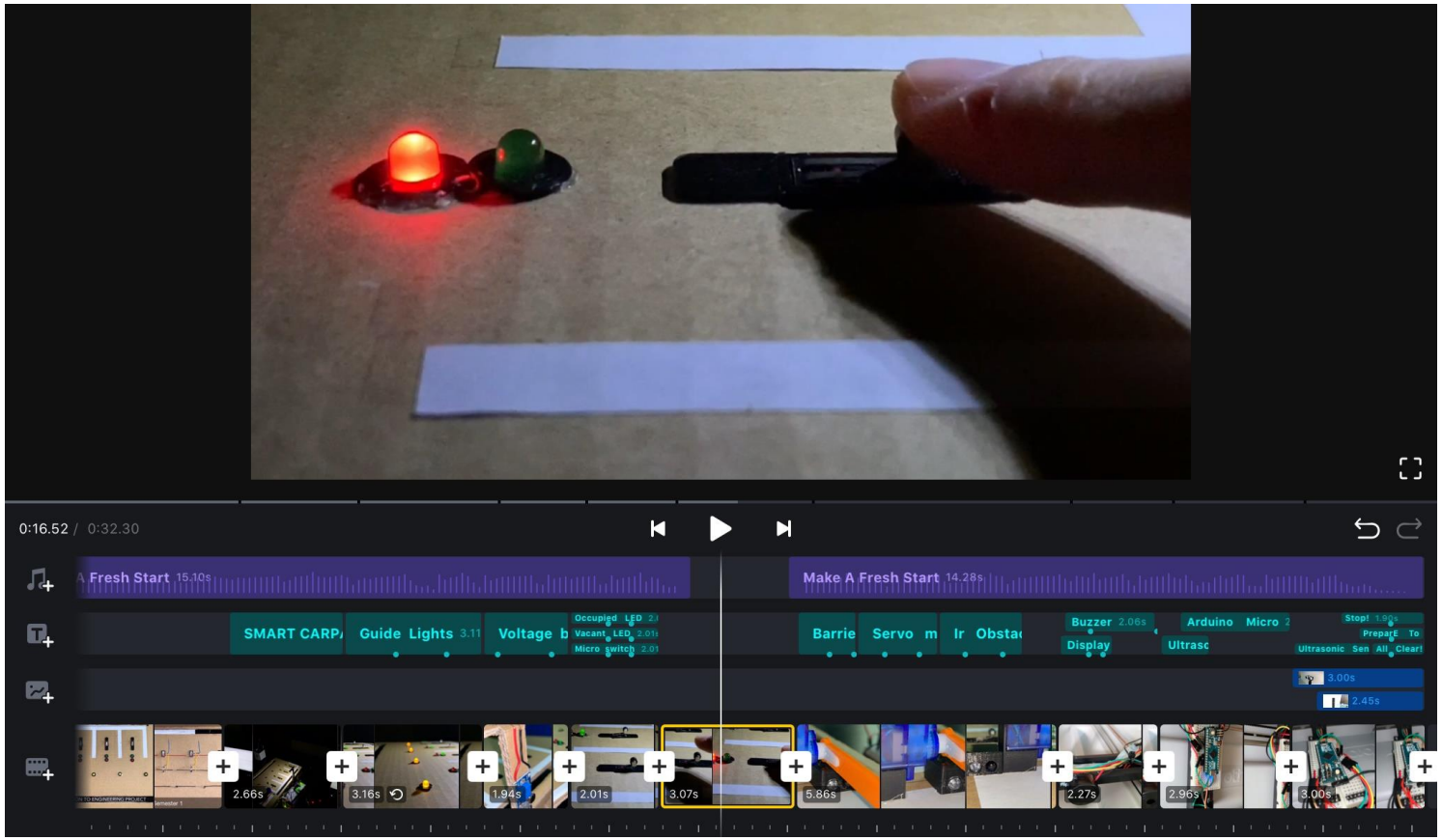






CAR MODEL

In order to create the video for the Carpark, I require “cars” in order to simulate cars parking into the lot. However, miniature cars cost a lot, as they are meant for collectors, not for Carpark testers. Hence, I created a crude version of the car with cardboard and placed weights on top of it to simulate the weight of the car.



VIDEO EDITING

Initially, I used iMovie to edit my video. However, after using it for some time, I realized that there are many restrictions (on iPad) such as being unable to rotate a picture, move the text positions, add animations to text, etc. Hence, after searching the App Store for an alternative app, I found VC, a free app with all these features and does not leave any inconveniences, such as a watermark, albeit a few advertisement when opening the app. The link will be included in the annex section.

PROJECT WORKS…BY STUDENTS

Final Project Link: https://youtu.be/8uZyaMhKqPE

Phase 5 (discontinued) Link: https://youtu.be/f5RWuzyZhAI

CONCLUSION

With reference to slide 3:

(1) Yellow LED will now guide the driver to the closest parking lot, reducing the time spent looking for an empty lot

(2) Carpark barrier will not allow the car to enter when the carpark is full, preventing congestion.

(3) This is solved with an open-air design, and LEDs will light up the surroundings when sensors detect that the surroundings are dark

(4) A display will show the driver the distance from the wall when he/she is parking

(5) The code will display a UI, which will show the parking lot status, as well as the Car Plate Number/Owner name, so that the owner knows where he/she had parked the car.

CHALLENGES FACED

- Running a stepper motor through pyFirmata will limit the motor's speed
- Difficult to multitask in python
- Not enough pins in a Arduino UNO board to complete the required functions
- Power limits of Arduino (unable to supply required current)
- Voltage limits of Arduino (unable to supply 6V, 7.4V and 12V for servo, stepper motor and led strip respectively)

FUTURE PLAN

- Use IOT to display space availability via a website/app
- Increase number of lots/stories
- Incorporate electric car chargers when Electronic cars becomes popular
- Use a more advanced microcontroller with more IO pins
- Create an anti-theft system which sounds an alarm if a car left a slot unexpectedly
- Vertical parking to reduce space used
- Use solar panels for sustainability and include plants within the Carpark for a more environmentally friendly feel
- Searching of car's location when more slots are added


Thank You

ANNEX (DATASHEET LINKS):

https://store.arduino.cc/usa/arduino-uno-rev3

https://store.arduino.cc/usa/arduino-micro

http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

https://www.mitsumi.co.jp/latest/Catalog/pdf/motor_m35sp_11nk_e.pdf

https://www.sunrom.com/p/dc-dc-boost-2a-microusb-mt3608

http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf

https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf

https://www.sunrom.com/p/light-sensing-module-ldr

https://www.rhydolabz.com/documents/26/IR_line_obstacle_detection.pdf

Micro switch datasheet could not be found, I used this link to learn how it works instead:

https://www.youtube.com/watch?v=q6nP1FjxAMU

ANNEX (PROGRAMMING REFERENCE CODES)

https://diyfactory007.blogspot.com/2018/04/diy-arduino-based-car-parking-assistant.html

https://mryslab.github.io/pymata4/

https://github.com/MrYsLab/pymata4/blob/master/examples/digital_input.py

https://github.com/MrYsLab/pymata4/blob/master/examples/digital_output.py

https://github.com/MrYsLab/pymata4/blob/master/examples/hc-sr04_distance_sensor.py

https://github.com/MrYsLab/pymata4/blob/master/examples/servo.py

https://github.com/MrYsLab/pymata4/blob/master/examples/stepper.py

https://github.com/MrYsLab/FirmataExpress

https://www.sparkfun.com/images/products/09856-04.jpg

https://www.sparkfun.com/images/products/09855-04.jpg

http://www.sparkfun.com/datasheets/Components/LED/COM-09594-YSL-R531Y3D-D2.pdf

ANNEX (OTHER LINKS):

Link of Autodesk Inventor course: https://www.lynda.com/Inventor-tutorials/Autodesk-Inventor-2021-Essential-Training-REVISION/2822351-2.html?srchtrk=index%3a1%0alinktypeid%3a2%0aq%3ainventor%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2

Average car HP: https://www.autolist.com/guides/average-car-horsepower

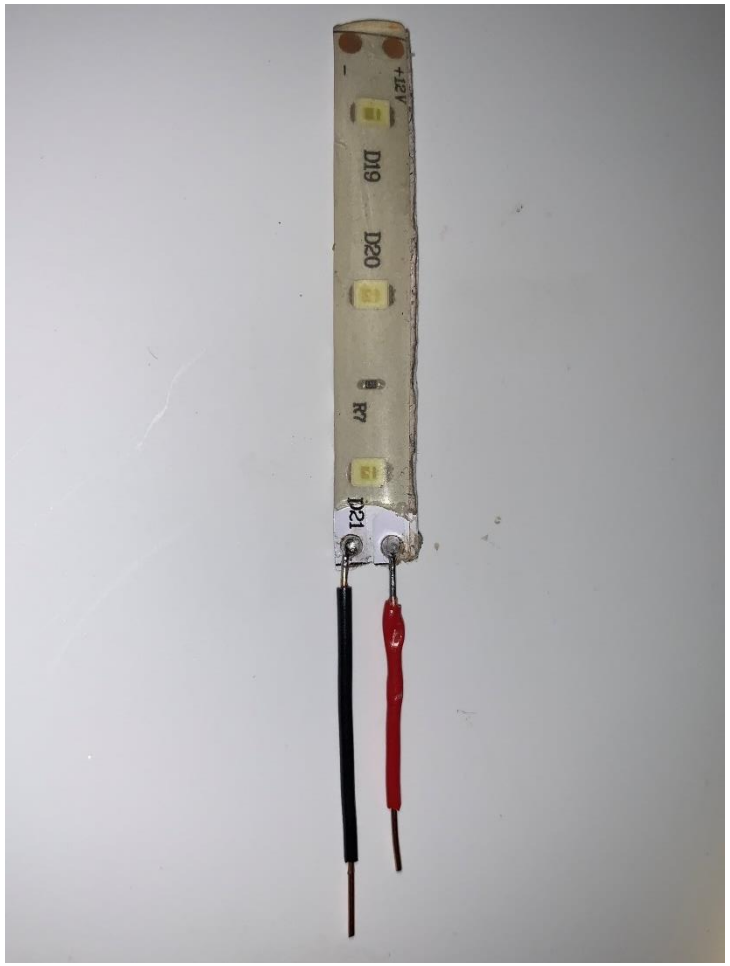
Average weight of car: https://www.onemotoring.com.sg/content/onemotoring/home/driving/commercial-vehicles.html

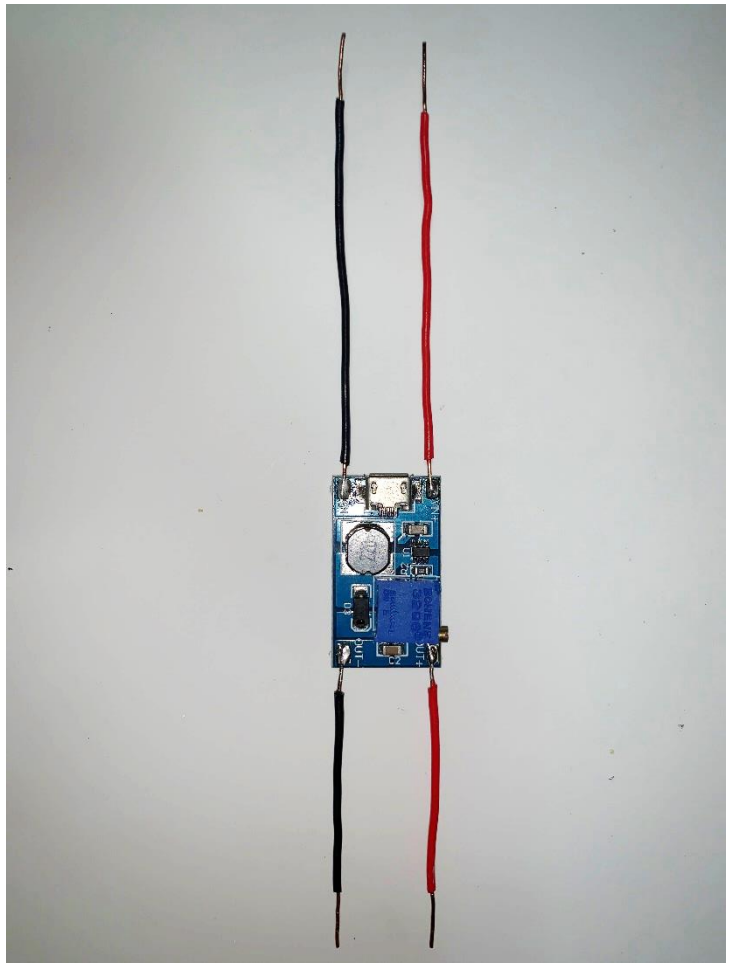
FBD, SFD and BMD calculator: https://skyciv.com/free-beam-calculator/

Grams to Newton converter: http://www.kylesconverter.com/mass/grams-to-newtons

VC video editor: https://www.vlognow.me/

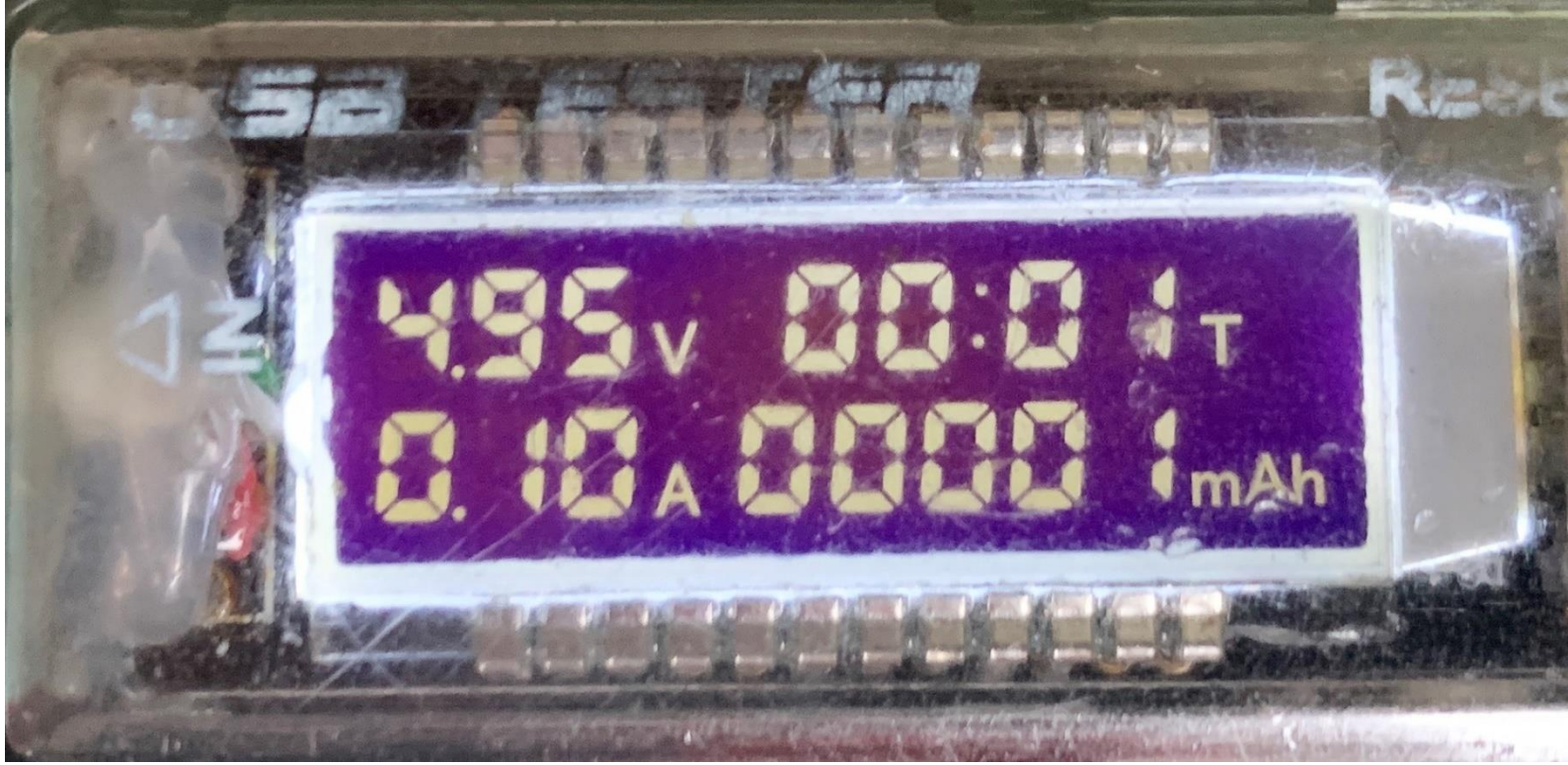
LED STRIP AND VOLTAGE BOOST CONVERTER

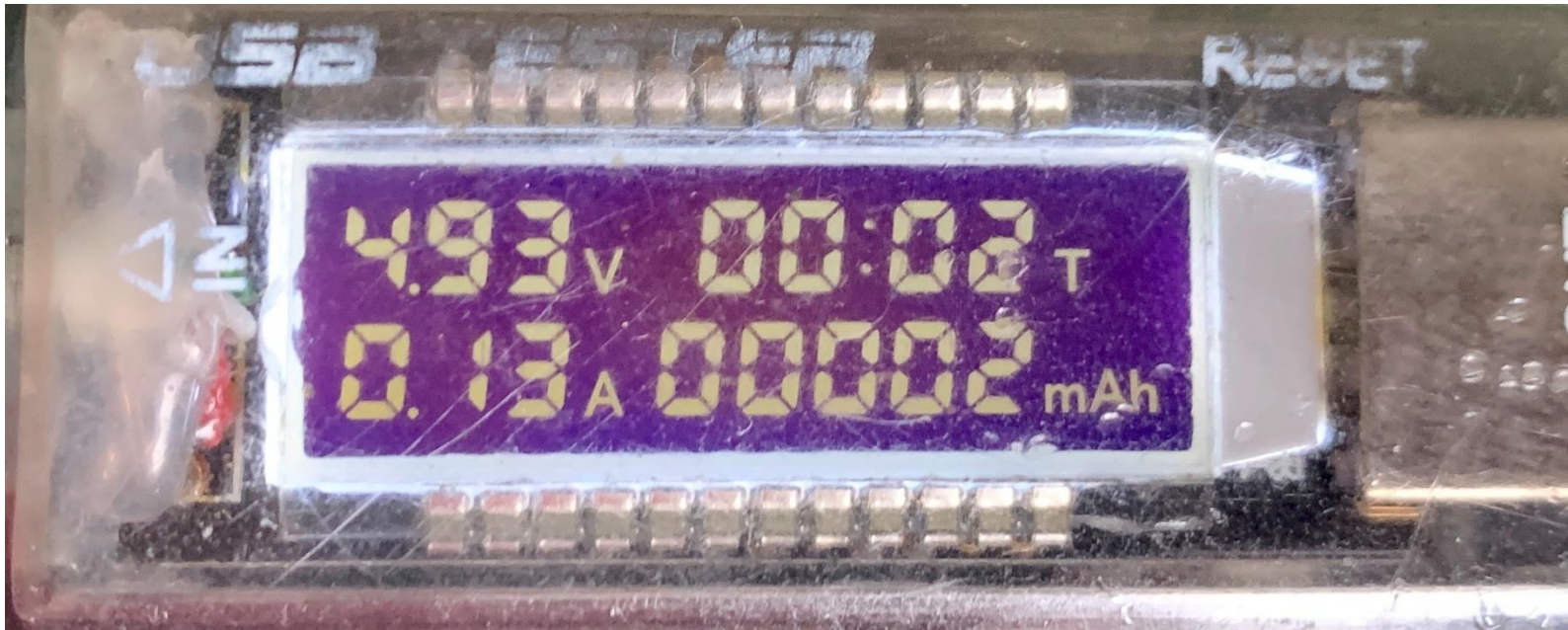




PICTURE OF POWER DRAW

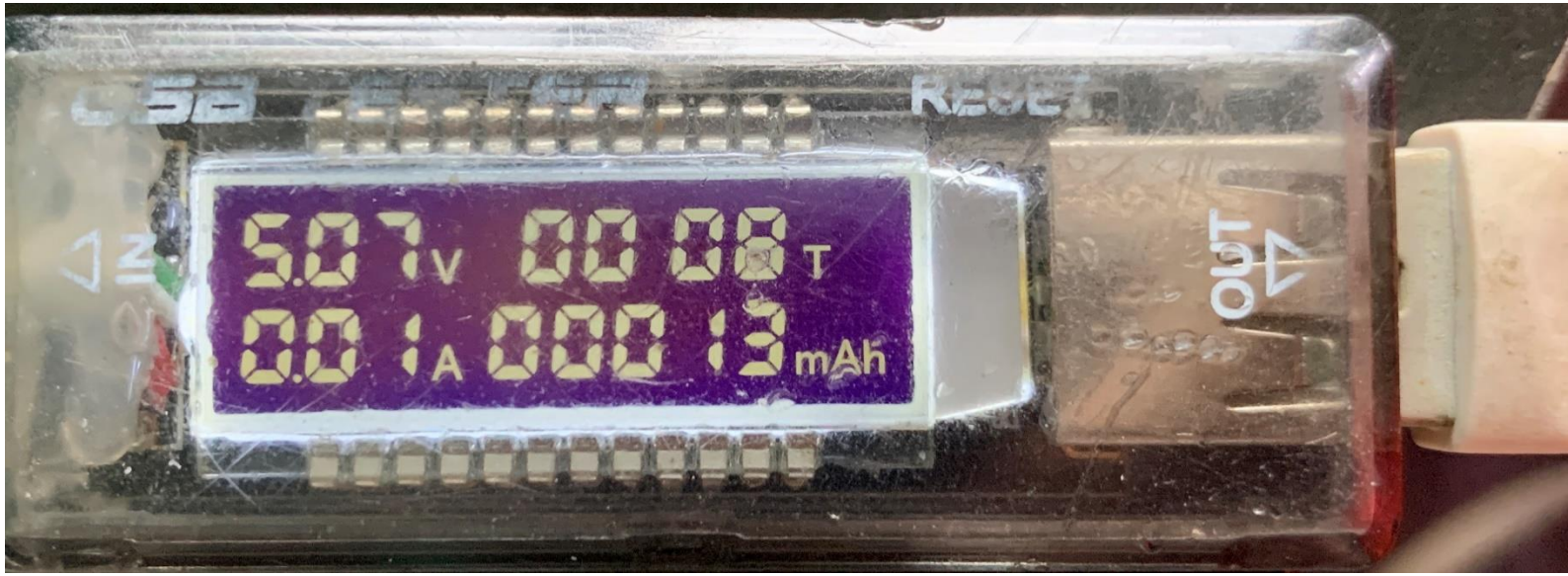
Arduino Uno





PICTURE OF POWER DRAW

Arduino Micro



PICTURE OF POWER DRAW

Stepper Motor
(idle, moving, hold)

ANNEX : MAIN CODE (UNO)

```
from time import sleep
from pymata4 import pymata4

# add resistor to button

board = pymata4.Pymata4()

# analog pins
lot1 = 0
lot2 = 1
lot3 = 2
lot4 = 3
button = 18
ldr = 19

# digital pins
obs1 = 2  # entrance
obs2 = 3  # exit
STEPPER_PINS = [4, 5, 6, 7]
servo = 8

guide_01 = 9  # led from gate up till first lot
guide_02 = 10
guide_03 = 11
guide_04 = 12
lights = 13

lot_status = []

board.set_pin_mode_analog_input(lot1)
board.set_pin_mode_analog_input(lot2)
board.set_pin_mode_analog_input(lot3)
board.set_pin_mode_analog_input(lot4)
```

ANNEX : PLATFORM CODE (MICRO) [IN USE]

```
int redLED = 2;

int yellowLED = 3;

int greenLED = 4;

int echoPin = 5;

int trigPin = 6;

int buzzer = 7;

long TempDistance = 0;

int counter = 0;


void setup() {

Serial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(redLED, OUTPUT);

pinMode(yellowLED, OUTPUT);
```

ANNEX : PLATFORM CODE [PYTHON]

```
from time import sleep
from pymata4 import pymata4

# digital
red_led = 2
yellow_led = 3
green_led = 4
buzzer = 7

# for Ultrasonic sensor
trig = 5
echo = 6

board = pymata4.Pymata4()

board.set_pin_mode_digital_output(red_led)
board.set_pin_mode_digital_output(yellow_led)
board.set_pin_mode_digital_output(green_led)
board.set_pin_mode_digital_output(buzzer)


# A callback function to display the distance
def the_callback(data):
    pass


def get_distance(my_board, callback):
    my_board.set_pin_mode_sonar(trig, echo, callback)
    return my_board.sonar_read(trig)[0]


def main():
    while True:
```

ANNEX : CARPARK (PHASE 1&2) CODE

```
from time import sleep
from pymata4 import pymata4

board = pymata4.Pymata4()

# analog pins
lot1 = 0
lot2 = 1
lot3 = 2
lot4 = 3
button = 4

# digital pins
obs1 = 2  # entrance
obs2 = 3  # exit
STEPPER_PINS = [4, 5, 6, 7]
servo = 8
guide_01 = 9
guide_02 = 10
guide_03 = 11
guide_04 = 12
guide_05 = 13

lot_status = []

board.set_pin_mode_analog_input(lot1)
board.set_pin_mode_analog_input(lot2)
board.set_pin_mode_analog_input(lot3)
board.set_pin_mode_analog_input(lot4)
board.set_pin_mode_analog_input(button)
```

ANNEX : DISTANCE + SERVO (PHASE 5) CODE

```
from time import sleep
from pymata4 import pymata4

board = pymata4.Pymata4()

# analog pins
lot1 = 0
lot2 = 1
lot3 = 2
lot4 = 3
button = 18
ld. = 19

# digital pins
obs1 = 2  # entrance
obs2 = 3  # exit
STEPPER_PINS = [4, 5, 6, 7]
servo = 8

guide_01 = 9  # led from gate up till first lot
guide_02 = 10
guide_03 = 11
guide_04 = 12
lights = 13

lot_status = []

board.set_pin_mode_analog_input(lot1)
board.set_pin_mode_analog_input(lot2)
board.set_pin_mode_analog_input(lot3)
```