

NAME : SAURABH SALVI

ROLL NO: ME22B1033

TASK #2
MARS CLUB



- Subtask – 1 : Documenting what have I learnt through provided youtube playlists

1) Lipo Batteries (Lithium Polymer Battery)

S in the batteries represents the number of cells in the battery as we know that the battery can be defined as the combination of cells.

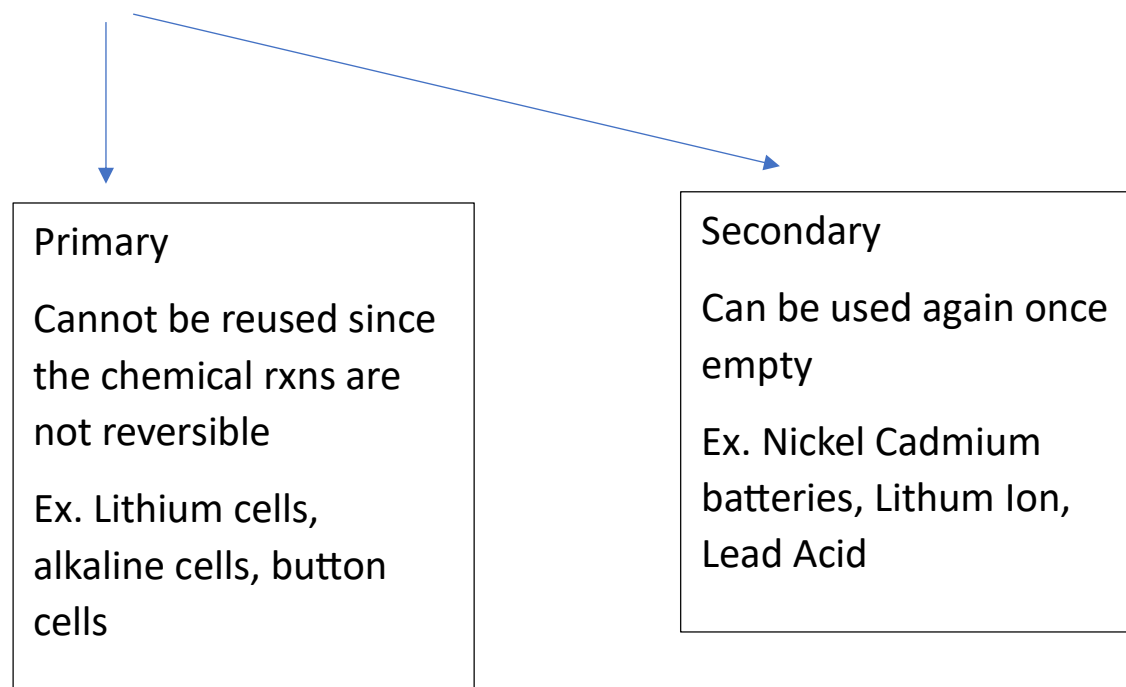
So 7S represents 7 cells combined in the battery

mAh is basically the time duration for which the battery would last

C represents the current which I can draw out from the battery safely at a given amount of time

XT60 connectors (female & golden coloured) for power supply and balance connector(white) for charging the battery and by looking at the ports available to balance connector we can guess S number of battery

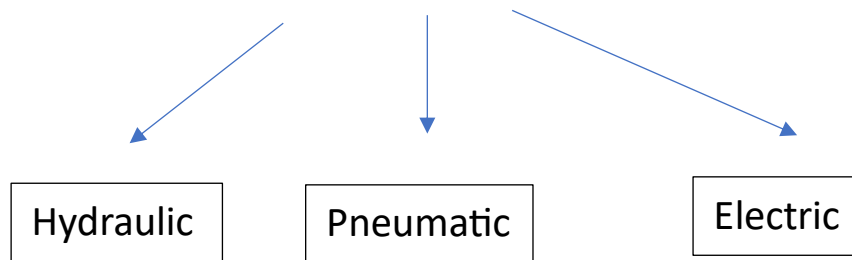
2) Batteries



3) Actuators

There are many types : Linear and rotary mainly. It basically receives a source of energy and uses it to move something

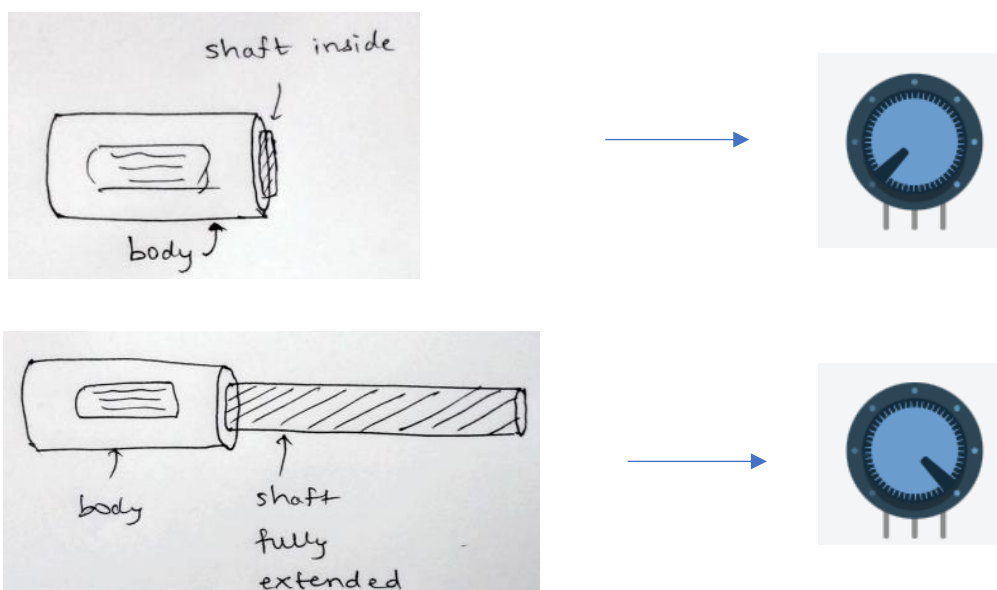
The sources of energy are :



4) Linear actuator

Here, as we know a motor would just rotate but the linear actuators actually make the shaft go 'in and out'

We have a feedback mechanism which helps like keeps sending signals regarding the location of shaft because of which it helps to stop the shaft from over extending mostly using potentiometer(maybe using diode).



Just like Dc motor, the direction of travel changes if the terminals of motor are given supply in opposite sense

Motor controller is used in order to control the speed of actuator (roboclaw)

In the interior, to mark the endings of extension, there is small extended part which actually presses the switch which in stops the shaft from further moving.

Terms :

- i. Stroke : Total amount which shaft can move
- ii. Static load : When the shaft is not moving, how much force can we keep on it
- iii. Dynamic load : When the shaft is moving, how much force can we keep on it
- iv. Speed of moving of shaft

5) Linear and Rotary Encoder

Linear encoder uses the distance travelled from the sensor or something as reference to actually get the position of object at that time instant

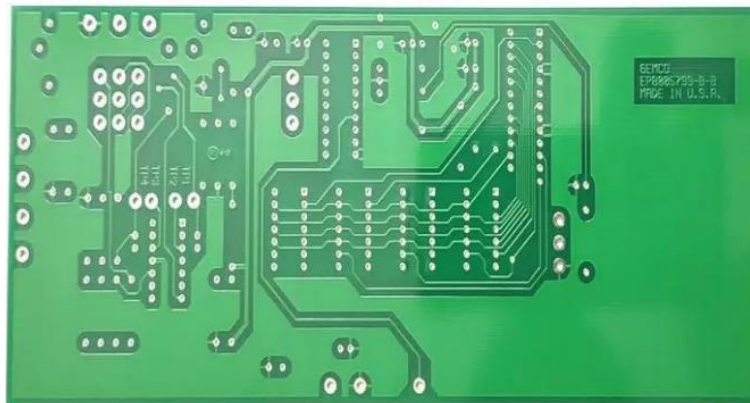
Rotary encoder can measure the angle rotation or basically tell angular position of object(absolute) while in incremental, we can measure distance, speed and linear position. In incremental, we can't measure the exact position of the object

6) PCB (Printing circuit board)

To make pcb's, the main thing is that we first take layers of fibreglass, put it together, then we keep copper plates layers on both side

Then the traces are done on the circuit board to match the required circuit.

Then the whole thing is exposed to UV light and after washing, we start getting the final prints of PCB board



7) Electrical components

Capacitors :

- The capacitors actually do the work of provided only the required amount of voltage to the circuit devices in order to avoid damage due to excess voltage application and hence overheating.
- Capacitors also act as the one which filter the voltage. They also keep the constant DC for devices by filtering out the AC signal (ripples)

Resistors :

- They limit the flow of current for the devices

Inductors :

- Smoothes out current based on how much the device can manage and work perfectly.

Transistors and MOSFTEs :

- They actually act as logic switches which help to carry out task
 - Also they act as a communication medium which helps them to figure out what voltage is perfect at that time instant according to the requirement.
-

- Subtask – 2 : Using Interrupt driven programming

For this subtask, I have used tinkercad software for Arduino programming.

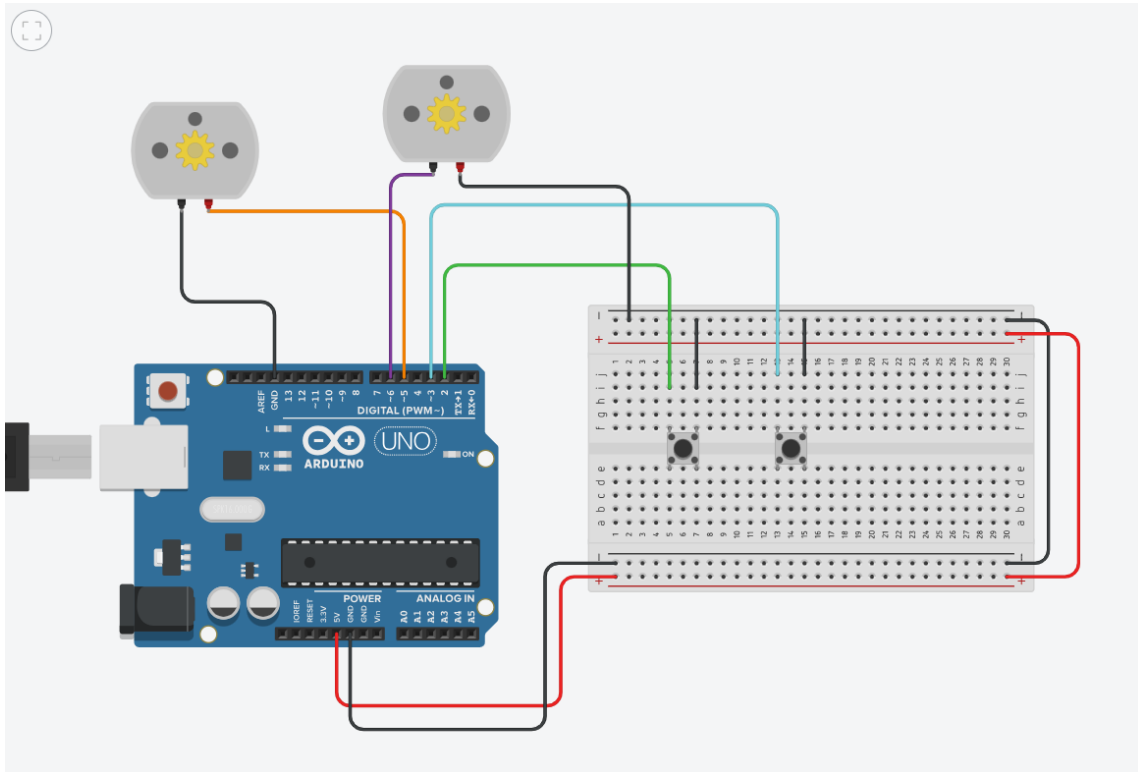


Fig. clockwise and anticlockwise motor

Code :

```
1 int motorclock = 5;  
2 int motoranticlock = 6;  
3 int interruptPin1 = 2;  
4 int interruptPinanti1 = 3;  
5 int state = LOW;  
6 int state1 = HIGH;  
7
```

```

8 void setup() {
9   pinMode(motorclock, OUTPUT);
10  pinMode(motoranticlock, OUTPUT);
11  pinMode(interruptPincl, INPUT_PULLUP);
12  pinMode(interruptPinanticl, INPUT_PULLUP);
13  attachInterrupt(digitalPinToInterrupt(interruptPincl), clock, RISING);
14
15  attachInterrupt(digitalPinToInterrupt(interruptPinanticl), anticlock, RISING);
16 }
17
18 void loop() {
19   digitalWrite(motorclock, state);
20   digitalWrite(motoranticlock, state1);
21 }
22
23 void clock() {
24   state = !state;
25 }
26
27 void anticlock() {
28   state1 = !state1;
29 }

```

Another Project for subtask 2:

Just using one switch to control the motor direction of clockwise and anticlockwise

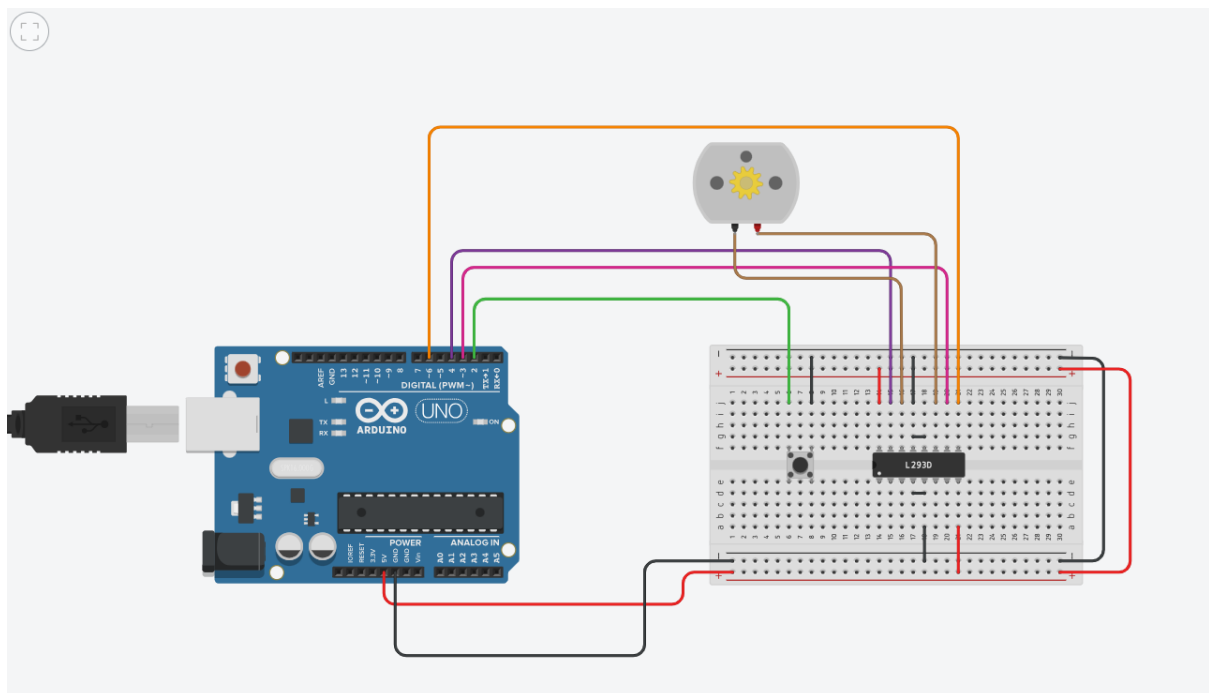


Fig. Another project for subtask-2

Code :

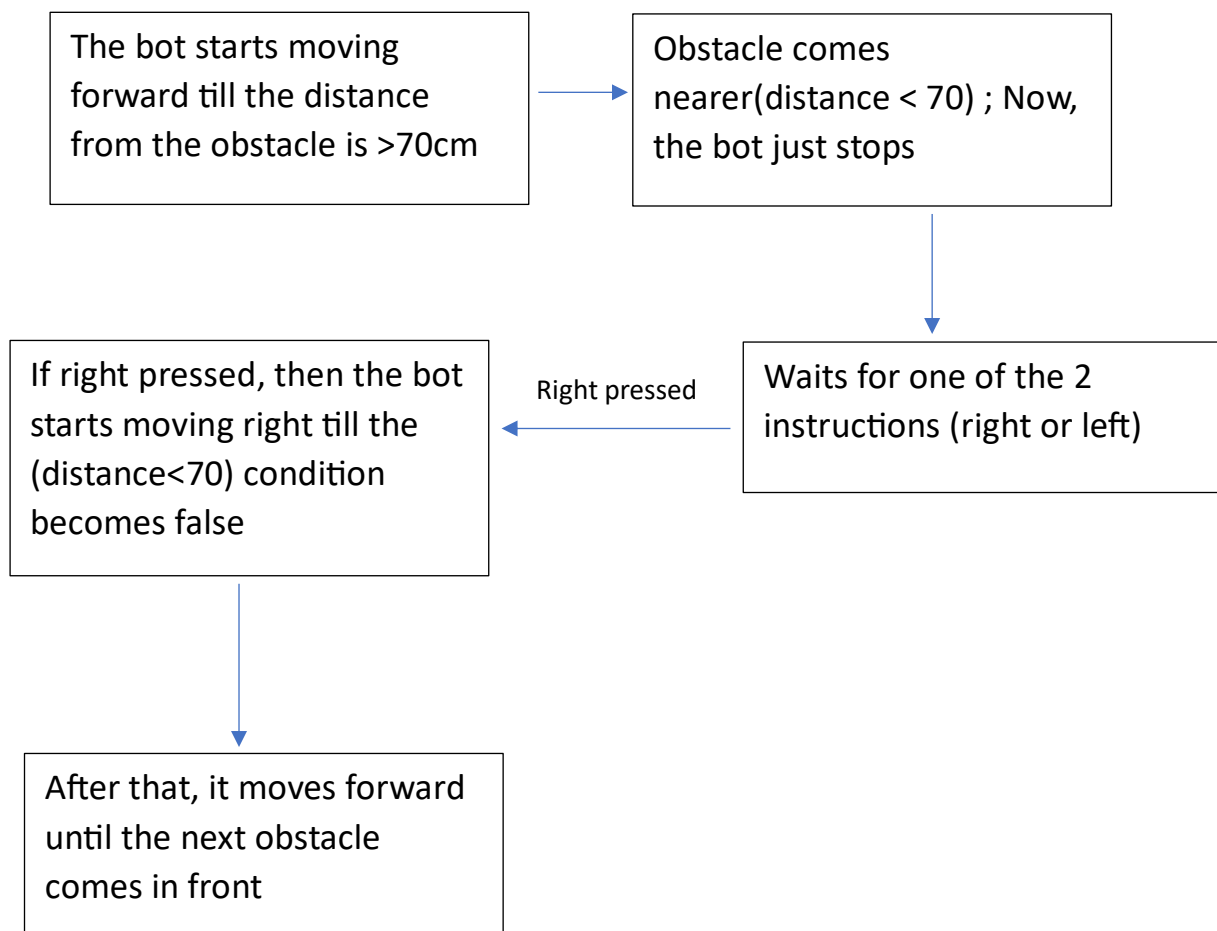
```
1  /*
2  Saurabh Salvi
3  ME22B1033
4  Another option for subtask-2 in Task#2
5  i.e. with just 1 button, control clockwise and anticlockwise
6      for single motor
7  */
8
9  int motorclock = 3;
10 int motoranticlock = 4;
11 int interruptPin1 = 2;
12 int en = 6;
13 //int interruptPinanticl = 3;
14 int state = HIGH;
15 //int state1 = HIGH;
16
17 void setup() {
18     pinMode(motorclock, OUTPUT);
19     pinMode(motoranticlock, OUTPUT);
20     pinMode(interruptPin1, INPUT_PULLUP);
21     //pinMode(interruptPinanticl, INPUT_PULLUP);
22     attachInterrupt(digitalPinToInterrupt(interruptPin1), clock, FALLING);
23
24     //attachInterrupt(digitalPinToInterrupt(interruptPinanticl), anticlock, FALLING);
25 }
26
27 void loop() {
28     if(state == LOW){
29         digitalWrite(en, !state);
30         digitalWrite(motorclock, HIGH);
31         digitalWrite(motoranticlock, LOW);
32     }
33     else{
34         digitalWrite(en, state);
35         digitalWrite(motorclock, LOW);
36         digitalWrite(motoranticlock, HIGH);
37     }
38 }
39
40 void clock() {
41     state = !state;
42 }
43
```

- Subtask – 3 : Building an obstacle avoiding robot

For this subtask, I have decided to build a obstacle avoiding robot with the help of Bluetooth module hc-05.

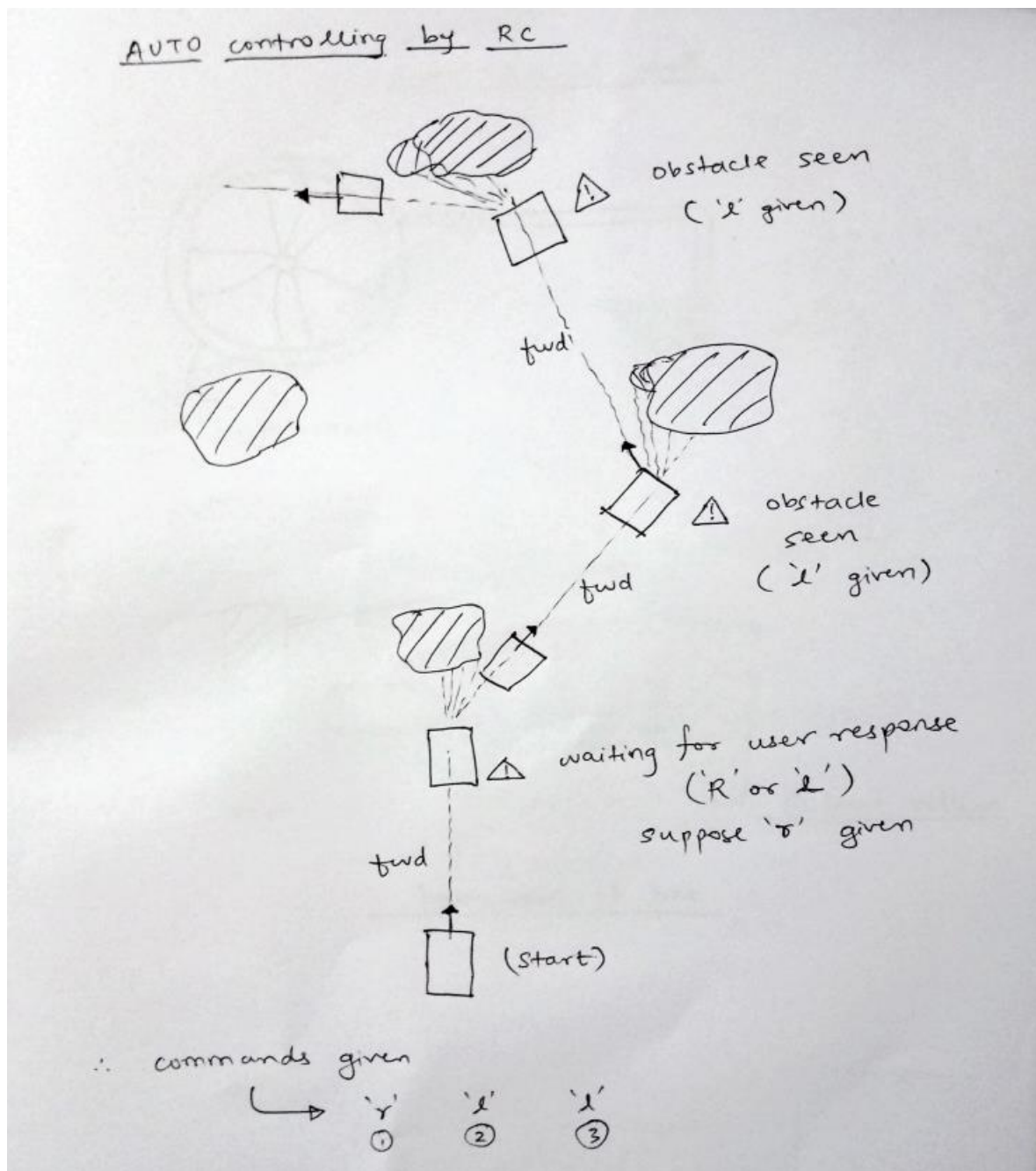
In this bot, I have kept two pushbuttons, 1 is for start/stop that will turn on and off the motors respectively and the 2nd is for auto and manual controlling of bot

In AUTO controlling,



Also there is an LED (also can be buzzer) in order to alarm or warn the user/controller about the obstacle coming near the bot

Diagram :



In MANUAL controlling,

Basically here, suppose we feel like in that specific situation, its better to control the bot by ourselves, then in that case we can just shift to manual controlling by pressing 2nd push button

Here controlling can be just done by pressing the respective controlling keys on controller. We have set following characters (received by Serial monitor) to represent the direction

Direction	Corresponding character
Forward	'f'
Backward	'b'
Right	'r'
Left	'l'

In this controlling, alarming using the LED is the only use of the ultrasonic sensor since all else can be controlled by the user itself.

Circuit Schematic :

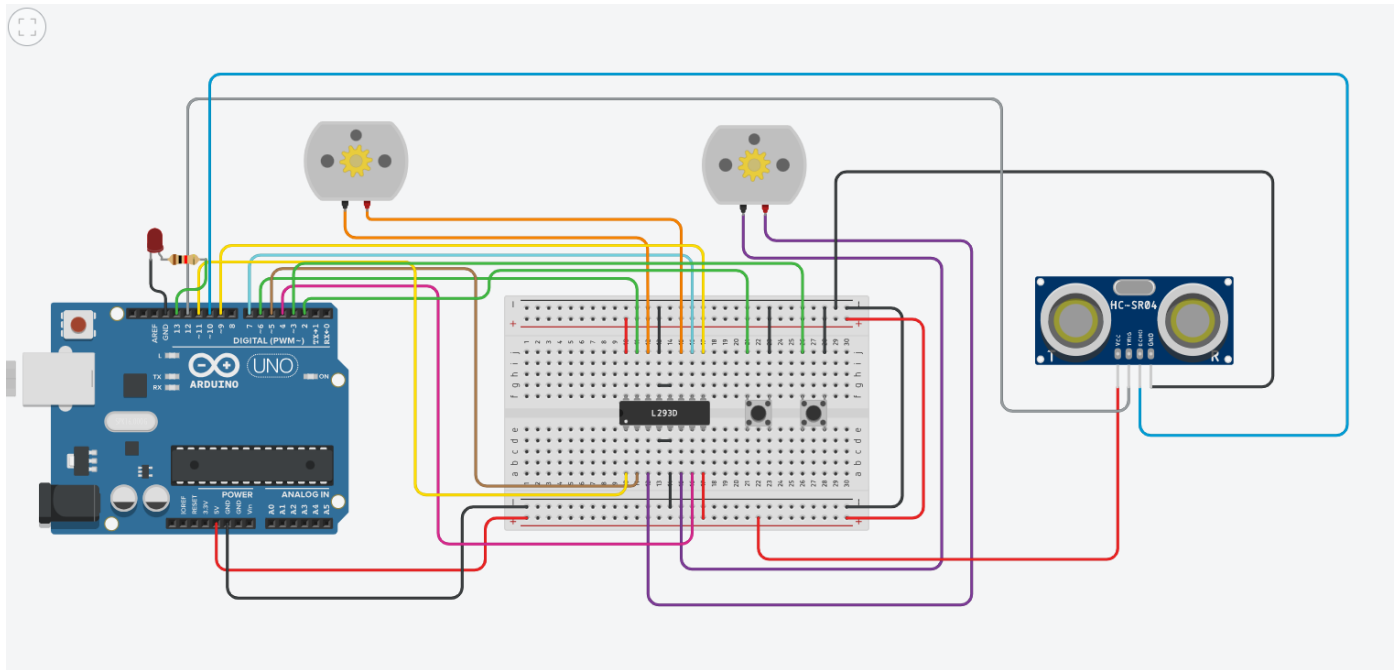


Fig. Circuit connection schematic for obstacle avoiding bot

Code:

```
1  int in1 = 5;
2  int in2 = 4;
3  int enM1 = 9;
4  int enM2 = 11;
5  int in3 = 7;
6  int in4 = 6;
7  int trig = 12;
8  int echo = 10;
9  int interruptPin = 2;
10 int automanual = 3;
11 int control = 0;
12 int state = LOW;
13 float time;
14 float distance;
15 //int buzzer = 8;
16 int led=13;
17
```

```

18 void setup() {
19     Serial.begin(9600);
20     pinMode(in1, OUTPUT);
21     pinMode(in2, OUTPUT);
22     pinMode(enM1, OUTPUT);
23     pinMode(in3, OUTPUT);
24     pinMode(in4, OUTPUT);
25     pinMode(enM2, OUTPUT);
26     pinMode(trig, OUTPUT);
27     pinMode(automanual, INPUT_PULLUP);
28     pinMode(echo, INPUT);
29     pinMode(interruptPin, INPUT_PULLUP);
30     pinMode(8, OUTPUT);
31     pinMode(led, OUTPUT);
32     //pinMode(buzzer, OUTPUT);
33
34     attachInterrupt(digitalPinToInterrupt(interruptPin), startStop,
35     attachInterrupt(digitalPinToInterrupt(automanual), AM, FALLING);
36
37 }
38

```

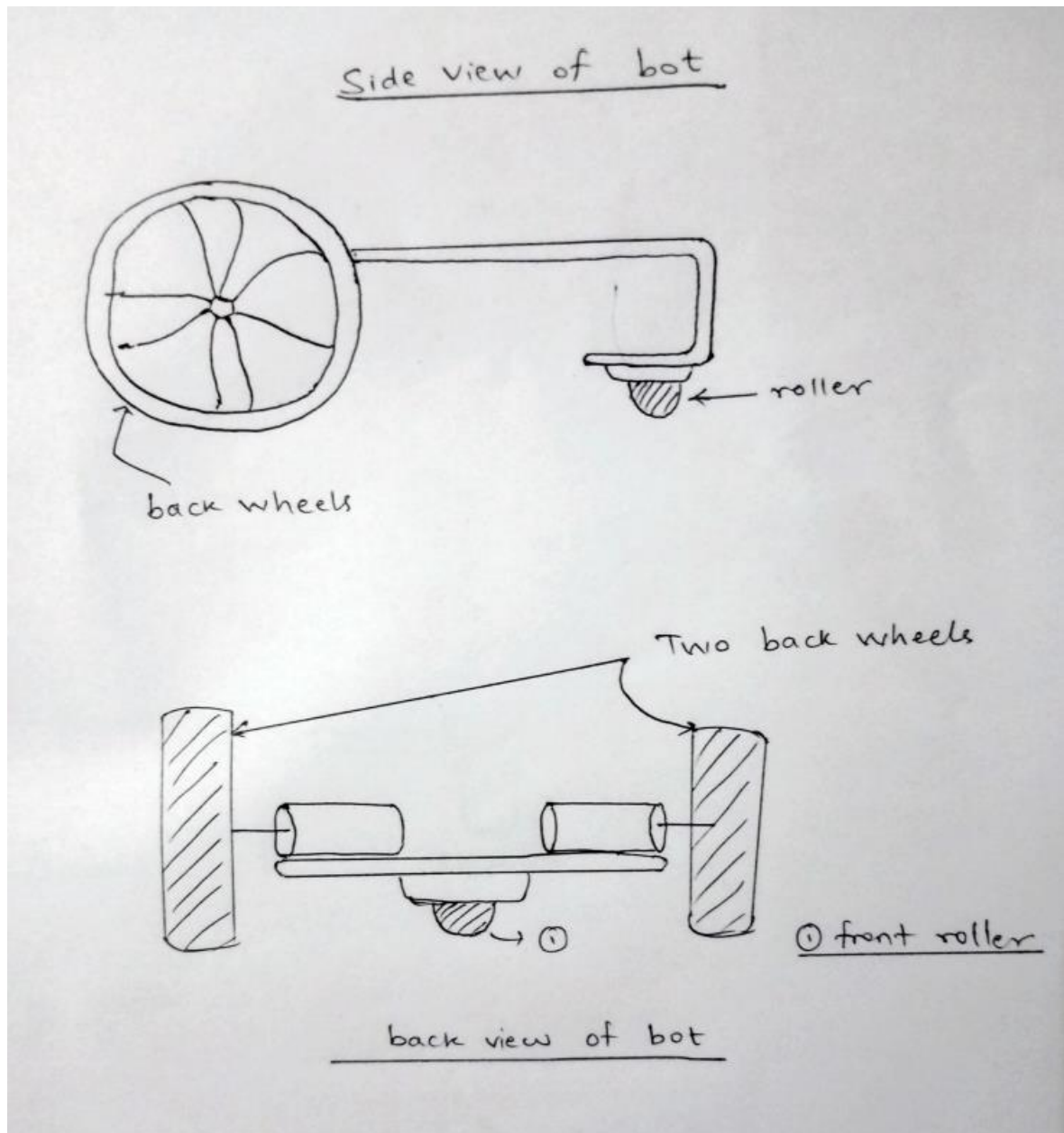
In Void Loop,

- i. First, I wrote the code for measuring the distance with the help of ultrasonic sensor after that based on the distance from obstacle, I am making LED glow as the alarming/warning system.
- ii. After that I write code for the AUTO controlling in which if the distance < 70, then the bot stops till Serial.available()>0 i.e. the Serial monitor gets an input.
- iii. For AUTO (control == 0), there are only 2 options, Right or left, codes for making the bot turn in right or left direction is written under switch statement.
- iv. And if the distance is not under 70cm, then the bot must keep moving forward.

- v. For MANUAL (control == 1), there are 4 options, forward, back, right, left. All of them are kept under switch statements so that when the respective button is pressed, accordingly the motors will be controlled to execute that command
- vi. After that, I have defined 2 functions as 'startStop' and 'AM' (Auto & Manual) which serve as my Interrupts functions which will be executed when one of the two pushbuttons (as external interrupts) are pressed accordingly.

Steering Mechanism :

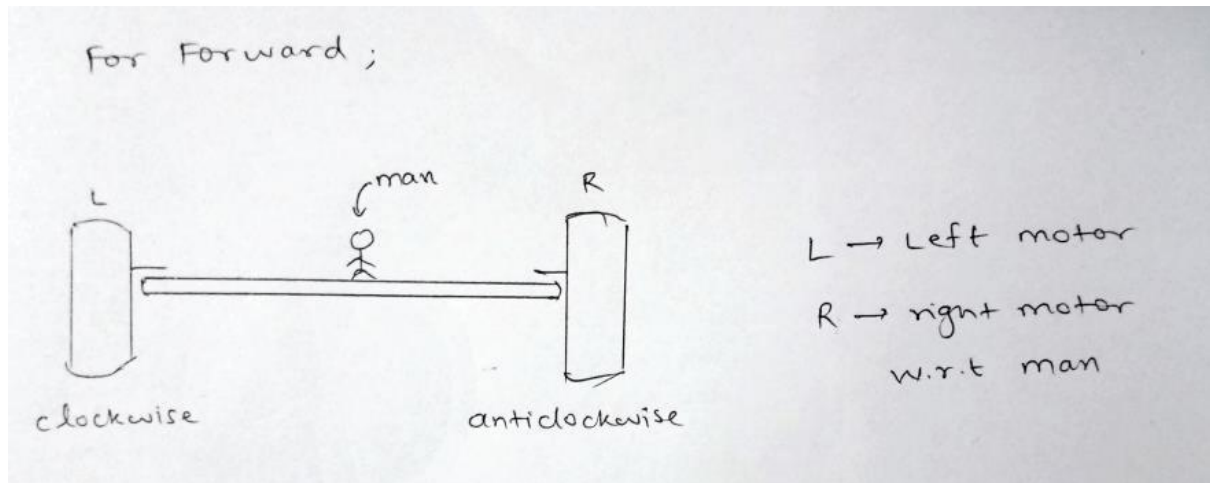
The bot is basically made up of 2 wheels connected to 2 motors which are connected to L293d motor driver and motor driver is interfaced with the Arduino microcontroller and for the front part, we just have a rolling ball in order to maintain the front part.



Forward direction :

LEFT motor → clockwise

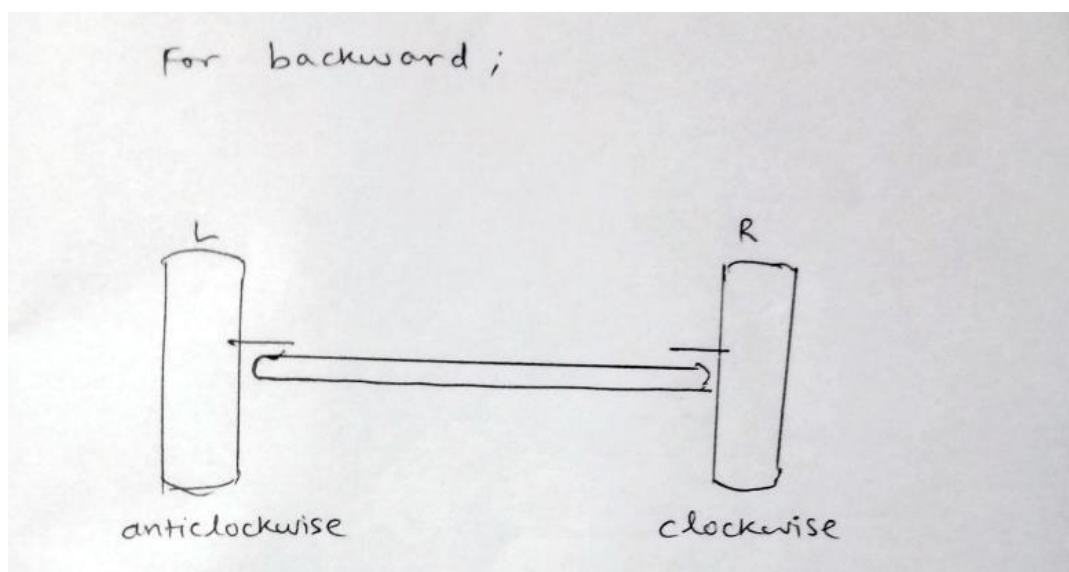
RIGHT motor → anticlockwise



Backward direction :

LEFT motor → anticlockwise

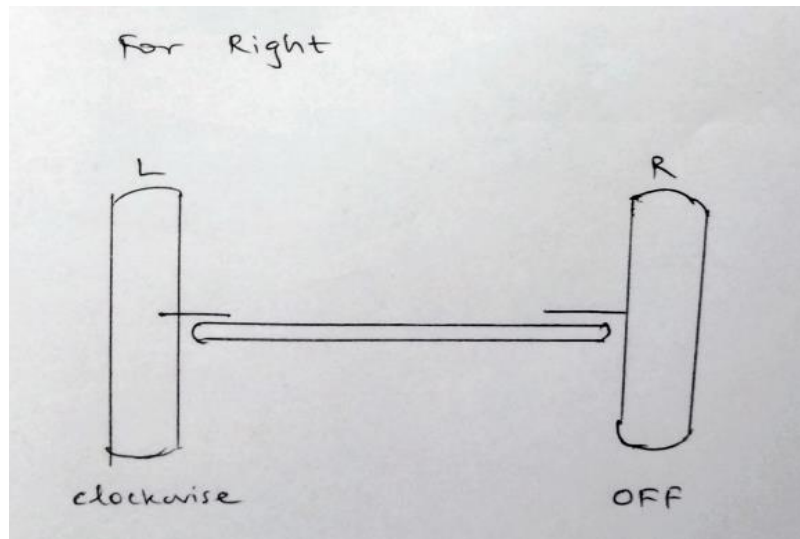
RIGHT motor → clockwise



Right direction :

LEFT motor → clockwise

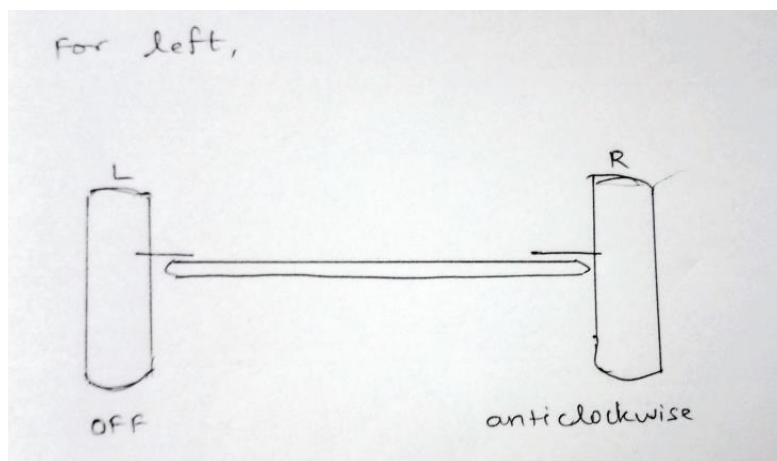
RIGHT motor → OFF



Left direction :

LEFT motor → clockwise

RIGHT motor → anticlockwise



Conclusion :

So through this task -2, I got to learn a lot about control systems and about communications and making a bot as well was really interesting. Overall, It was really fun to do this task.
