ME2400 - Project Group B103

- R S Harish Krishnan (ME20B081)
- Harshavardhan Srinivas Pentakota (ME20B083)
- Muthu Saran C V (ME20B115)
- Pilli Vasanthi (ME20B128)

OBJECTIVE (taken from course site, change some words)

To Build a “mini-Segway” that can house ALL the components, provided in the project kit, on a chassis that can balance itself (any part of the chassis should not touch the ground), without any external power supply, and without toppling when subjected to small imposed external disturbances on (only) two axially aligned, independently actuated motorised wheels.

COMPONENTS USED:

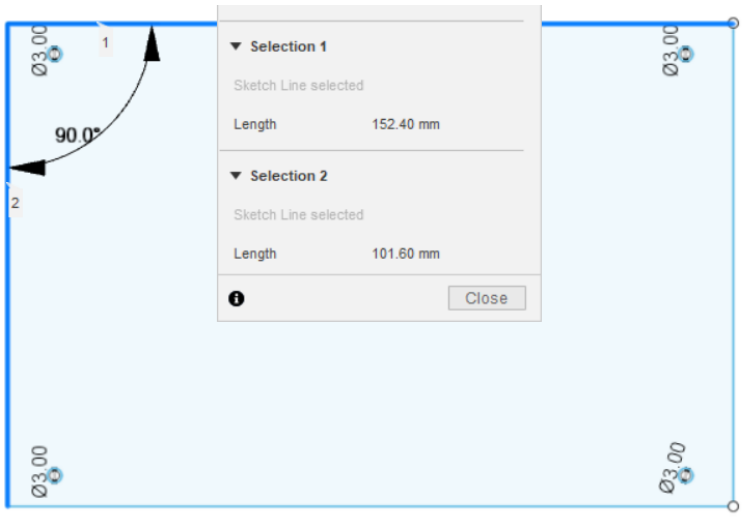
ESP 32 Dev Board
MPU 6050 Sensor
L293D Motor shield custom designed by Electronics club,IITM
Battery Operated DC Motor
4 Alkaline Batteries
Insulation Tape
Acrylic Board for Chassis
USB Cable

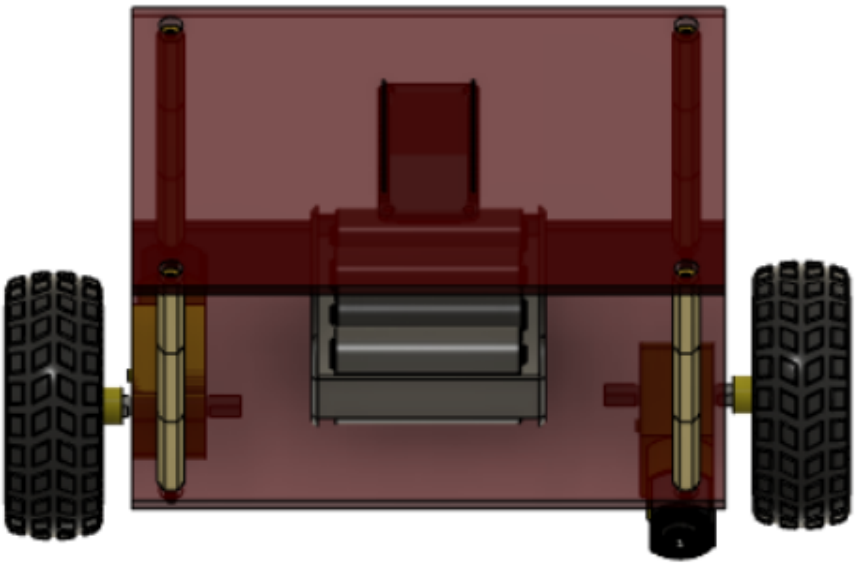
Jumper Wires
Wheels (*2)
Miscellaneous (screws/bolts/clamps) for Chassis

MECHANICAL DESIGN (need to check, will add pics tomorrow)

In our mechanical design we have placed all our components such that the centre of gravity remains as low as possible and also, such that it remains on the same vertical plane as the centres of the wheels. To maintain this we have positioned our DC motors on the underside of our bottom surface and the DC motors are placed facing opposite ways.
The ESP 32 is placed on the underside of the top layer, and the external MPU 6050 sensor is placed on the top flat surface as it has to be placed on the top most layer as it needs a decent height to get steady readings.
The battery pack is placed on the upper side of our bottom layer and is placed in the middle.This layer is kept only for the battery pack, since as a single entity it is the heaviest component on our Chassis.

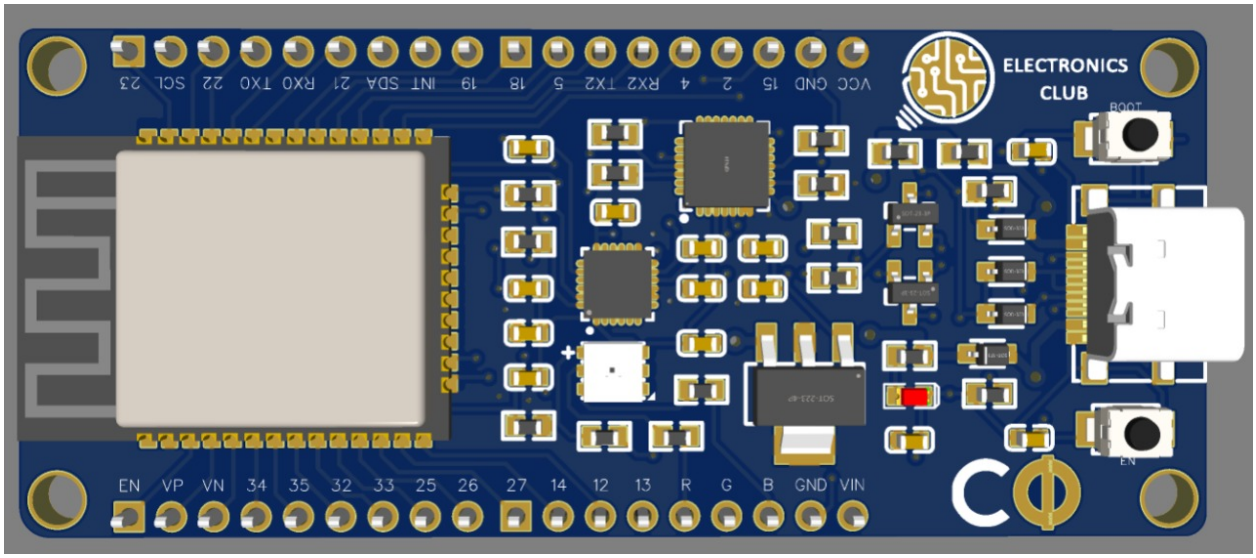
Dimensions of our Chassis Design :
Acrylic board : 4in * 6in
Drill Bit size for our spacer bolts and L-clamps : M3 (3mm dia)


Ø3.00
1
90.0°
2
Selection 1
Sketch Line selected
Length 152.40 mm
Selection 2
Sketch Line selected
Length 101.60 mm
Close
Ø3.00
Ø3.00
Ø3.00



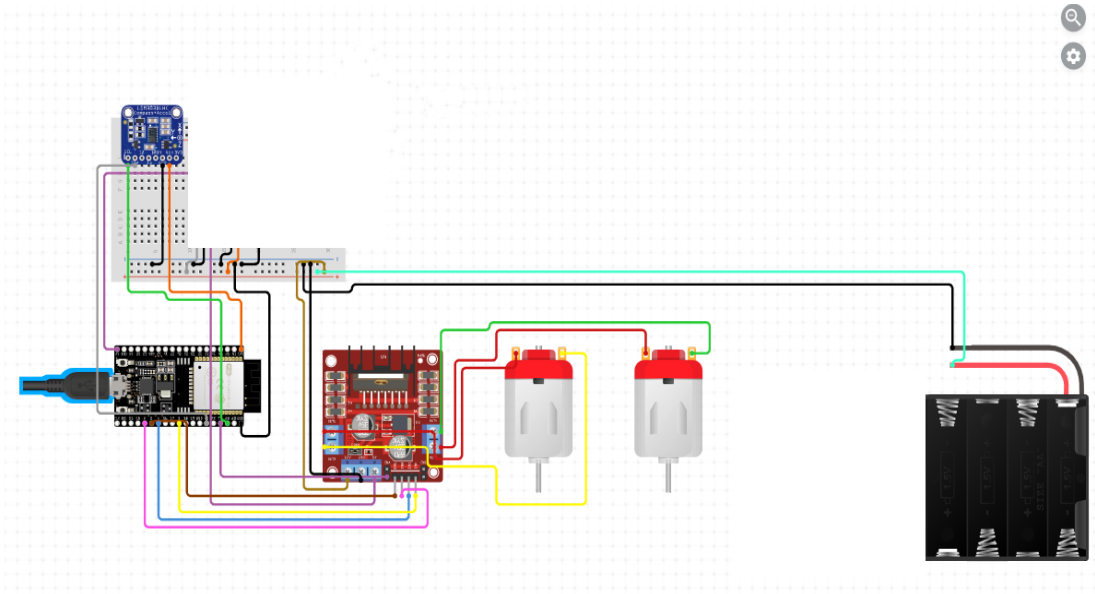
ELECTRONIC CIRCUIT DIAGRAM:

The controller diagram[ESP32 development board custom designed by electronics club of IITM.



Picture source:https://cfi-electronics-club.github.io/

The Circuit diagram:



CONTROL BLOCK DIAGRAM:

TRANSFER FUNCTION:

CONTROLLER DESIGN:

The controller we used in this project is an ESP32 development board custom designed by Electronics Club IITM. We Implemented PID controller in the microcontroller by using the PID algorithim.

CODE USED IN THIS PROJECT::https://drive.google.com/file/d/1VdZuw_rYskjDmmGl3aqAUQptrmFCAj9A/view?usp=sharing

PID TUNING,FINAL PARAMETER VALUES

Make Kd,Ki zero. Increase Kp from zero to point where bot starts shaking vigorously

Now increase Kd slowly such that bot reduces its shaking

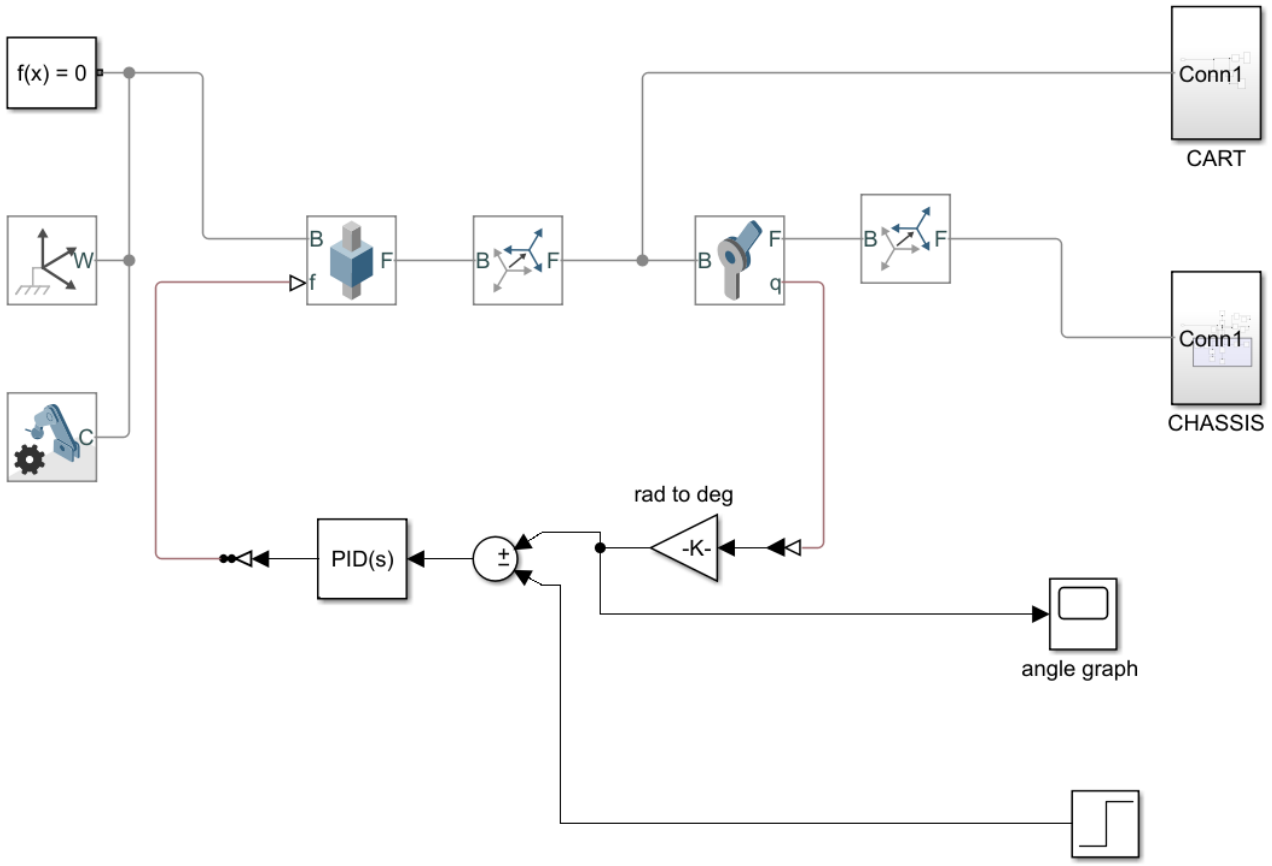
Now increase Ki till the bot balances perfectly

➔ PID gains::

- Kp=30
- Kd= 0.001
- Ki=0.2

SIMULINK MODEL::

We tried to tune the pid controller directly by using an auto tuner in MATLAB SIMSCAPE. For that we made a simscape model of the robot.
We used multibody extension in simscape to model the robot
The multibody simscape design circuit:



RESULT :

CONCLUSION AND LESSONS LEARNT (need to change some wordings)

● We were able to do the mechanical design of our Self-balancing bot, place components, make the necessary electronic circuit connections write an Arduino code in time

● We were also able to do some amount of tuning for our bot to self balance with the help of our Simulink Model.

● While working on the Simulink model, we learnt more about the extra add ons like simscape multibody,driveline etc..

● We were able to test out the Ultimate cycle method for controller tuning in the Simulink model