Mechatronics

Introduction to Mechatronics
MCT131s

Final project

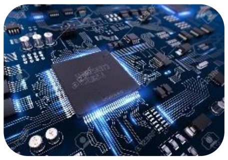
Team:9

Contents

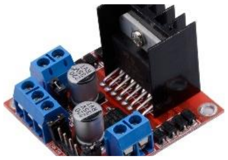
Bill of materials:

car

Arduino


PCB

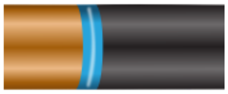
IR Sensors


H-bridge

Servomotor

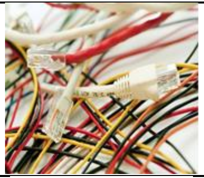
DC motors

color sensor


batter

jumpers

Name of component	No. of used pieces	Picture of component
Car chassis	2	
Arduino	2	
PCB	1	
IR Sensor	3	
H-bridge	1	

Servomotor	1	
DC motors	4	
Color sensors	1	
Battery	3	
jumpers	-	
M3x30 Screw	8	
M3x8 Screw	8	
M3 Nut	8	
T stand	8	
L30+6 Spacer	6	
Section four battery box	1	

Car body:

This kit for the car is our choice to make the project as it is light and suitable.

Outer dia of wheel:	65mm
Width:	28mm
Internal Axe Diamter:	3.7x5.3mm
External Axe Diameter:	13.8mm

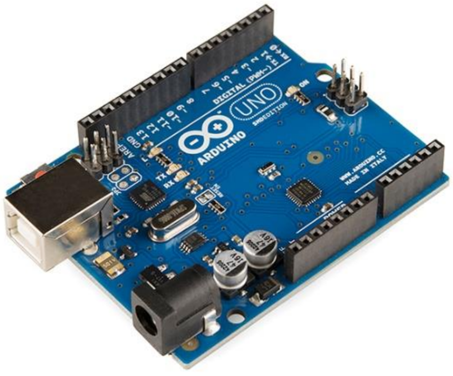
Motors:

To move the car, we used 4 DC motors and we fed them from the H-bridge as we have batteries with a total of 12v.

Specifications

- Voltage: 4- 9 V DC
- No Load Speed: 90 +/- 10rpm
- No Load Current: 190mA (max.250mA)
- Minimum Torque: 800 gm.cm

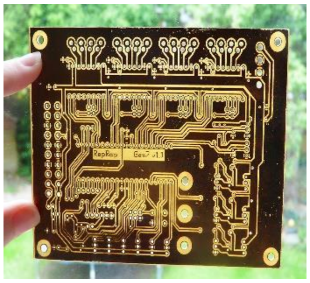
The microcontroller [Arduino]:



Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards can read inputs - a light on a sensor, a finger on a button, or a Twitter message - and turn them into an output - activating a motor, turning on an LED, or publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. You use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

PCB:

Our PCB is straightforward it has 5v and ground terminals.



IR Sensor:

Are necessary in the system to make sure the desired motion exists.

We use two types of sensors the IR to make the car move in the track as wanted and the colour sensor to make the car stop at the required positions.

- IR sensor: TCRT5000

  Specifications

- using infrared reflective sensor TCRT5000
- detecting the reflected distance: 1mm ~ 25mm applicable
- the comparator output signal clean, good waveform driving ability more than 15mA.
- with multi-turn precision potentiometer adjustable sensitivity adjustment
- the working voltage of 3.3V-5V
- the output format: digital switching output (0 and 1)
- a fixed bolt holes for easy installation
- small plates PCB size: 3.2cm x 1.4cm
- using a wide voltage LM393 comparator

Batteries:

We chose batteries with a total voltage of 12v

H-bridge:

An H-bridge is an electronic circuit that switches the polarity of a voltage applied to a load. These circuits are often used in robotics and other applications to allow DC motors to run forward or backward.

We use it to control the motion of the DC motors.

Specifications

- Driver: L298N Dual H Bridge DC Motor Driver IC
- Driven part of the terminal supply area Vs: +5 V ~ +35 V;
- Peak current Io: 2A
- The logical part of the terminal supply area Vss: +5 V ~ +7 V (can take power within the board +5 V)
- Storage temperature: -25 Deg C ~ +130 Deg C
- Other Extensions: control of direction indicators, the logic part of the plate to take power interface.
- Driver Board Size: 55mm * 60mm * 30mm
- Maximum power consumption: 20W (when the temperature T = 75 Deg C)

The servomotor:

is used for the shooting mechanism. And the bat cave

Specifications

- Operating Voltage : 4.8~6.0V
- Operating Speed : 0.12sec/60 degree(4.8V)~0.1sec/60 degree(6.0V)
- Torque : 1.6kg/cm(4.8V)
- Dead Band Width : 5usec
- Temperature Range : -30~+60â„ƒ
- Cable Length : 25cm
- Servo Type : Analog Servo

Color sensor:

Color sensors are used to detect the color of a material in RGB (red, green, blue) scale while rejecting unwanted infrared or ultraviolet light. The ultimate challenge with color sensing has been to detect subtle differences among similar or highly reflective surfaces.

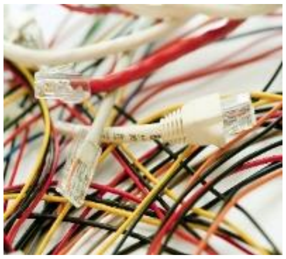
- Color sensor: TCS3200

  Specifications

- Chip: TCS3200
- Input voltage: DC 3 – 5V
- Use bright white LED lights
- Can be connected directly with Microcontroller
- Static detection of the measured object color
- Best detection distance: 1cm

Wires[jumpers]:

We used it to connect the system elements.

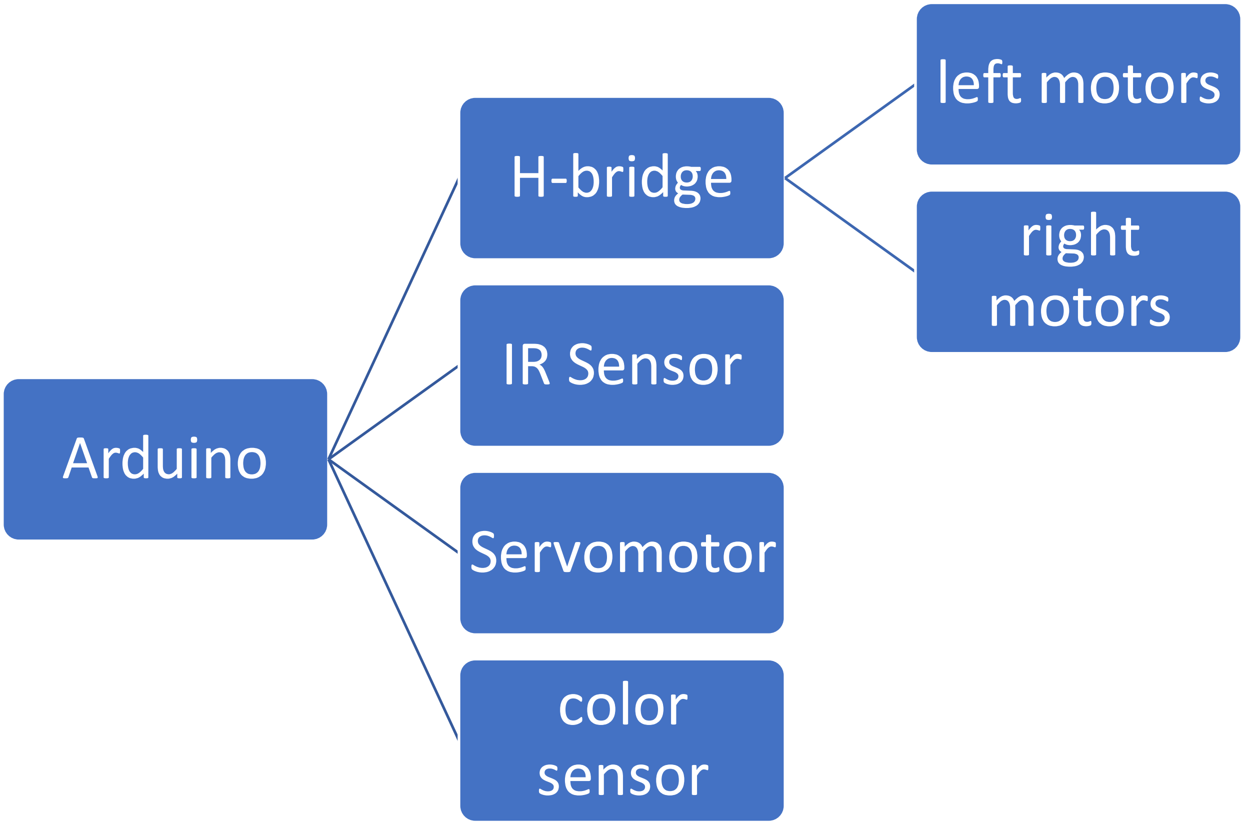


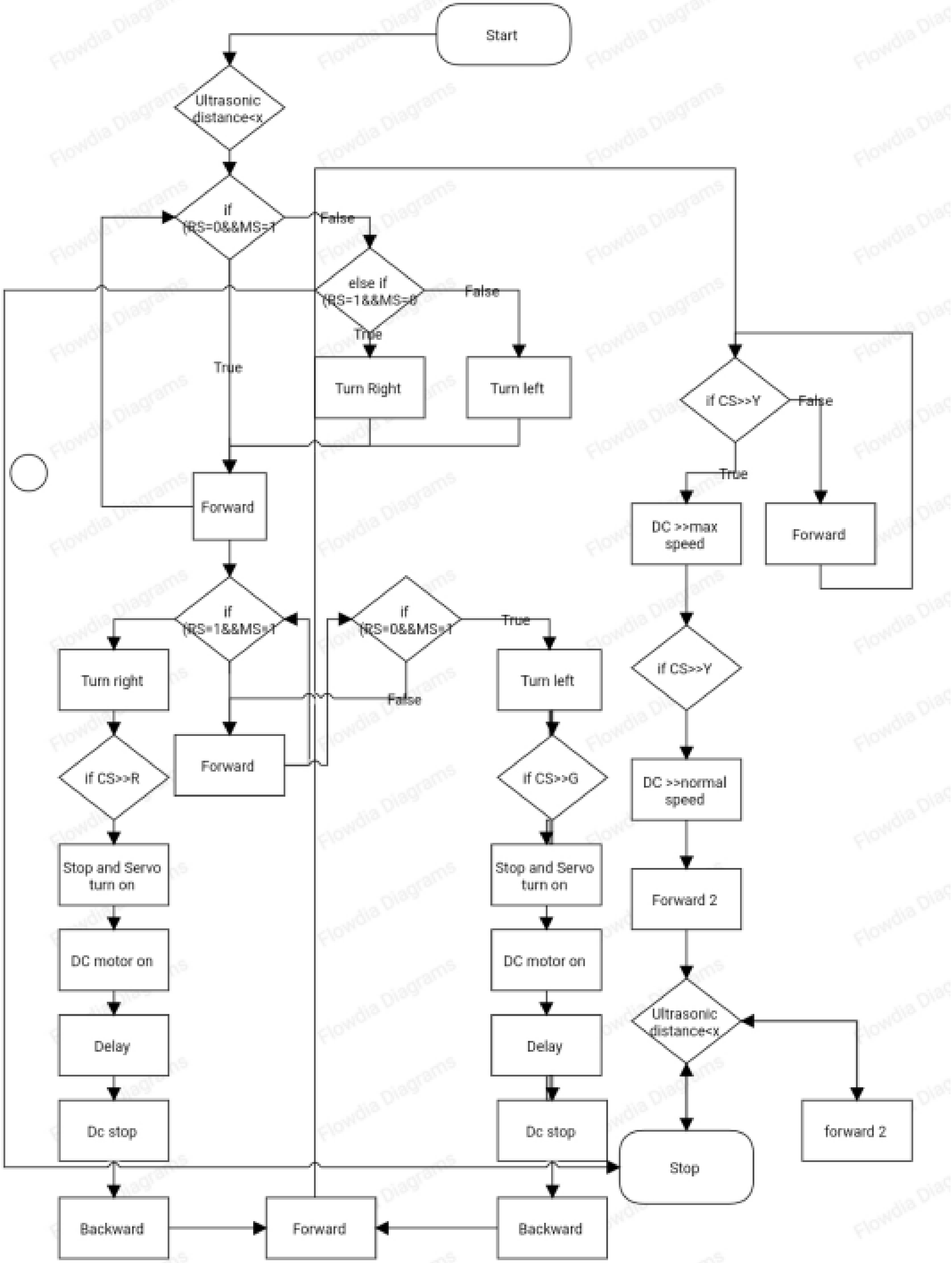
Control system:

Software:

Arduino IDE:

To control our system, we used the Arduino IDE to make our code debugging it and then burn it on the Arduino.




Start
Ultrasonic distance<x
if (RS=0&&MS=1
False
else if (RS=1&&MS=0
False
True
True
Turn Right
Turn left
Forward
if CS>>Y
False
True
DC >>max speed
Forward
if (RS=1&&MS=1
if (RS=0&&MS=1
True
False
Turn right
Forward
Turn left
if CS>>Y
if CS>>R
if CS>>G
DC >>normal speed
Stop and Servo turn on
Stop and Servo turn on
Forward 2
DC motor on
DC motor on
Delay
Delay
Ultrasonic distance<x
Dc stop
Dc stop
Stop
forward 2
Backward
Forward
Backward

The Code:

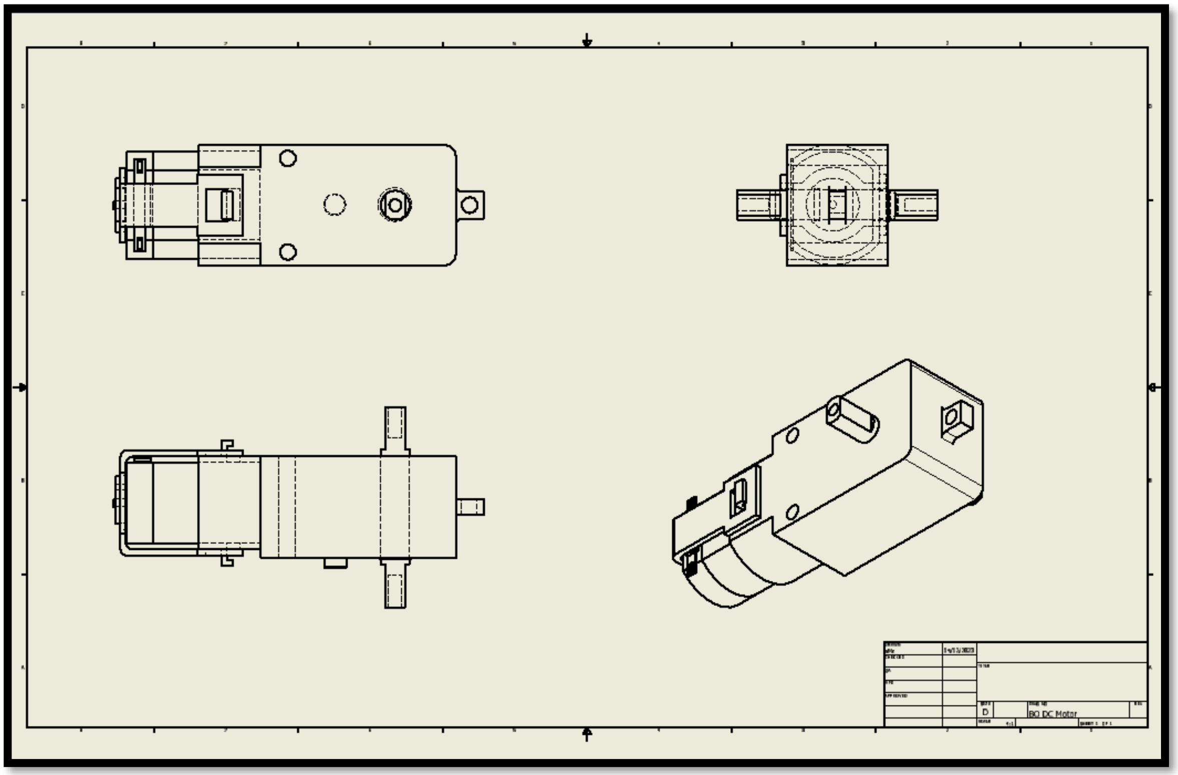
```
#define enA 10//Enable1 L298 Pin enA
#define in1 9 //Motor1  L298 Pin in1
#define in2 8 //Motor1  L298 Pin in1
#define in3 7 //Motor2  L298 Pin in1
#define in4 6 //Motor2  L298 Pin in1
#define enB 5 //Enable2 L298 Pin enB
#define L_S 13//ir sensor Left
#define R_S 12 //ir sensor Right
int Set=15;
int distance_L, distance_F, distance_R;
void setup(){ // put your setup code here, to run once
Serial.begin(9600); // start serial communication at 9600bps
pinMode(R_S, INPUT); // declare if sensor as input
pinMode(L_S, INPUT); // declare ir sensor as input
pinMode(enA, OUTPUT); // declare as output for L298 Pin enA
pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4
pinMode(enB, OUTPUT); // declare as output for L298 Pin enB
analogWrite(enA, 200); // Write The Duty Cycle 0 to 255 Enable
Pin A for Motor1 Speed
analogWrite(enB, 200); // Write The Duty Cycle 0 to 255 Enable
Pin B for Motor2 Speed
}
void loop(){
//===============================================
//     Line Follower and Obstacle Avoiding
//===============================================
//if Right Sensor and Left Sensor are at White color then it
will call forword function
 if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0)){
{forword();}
```

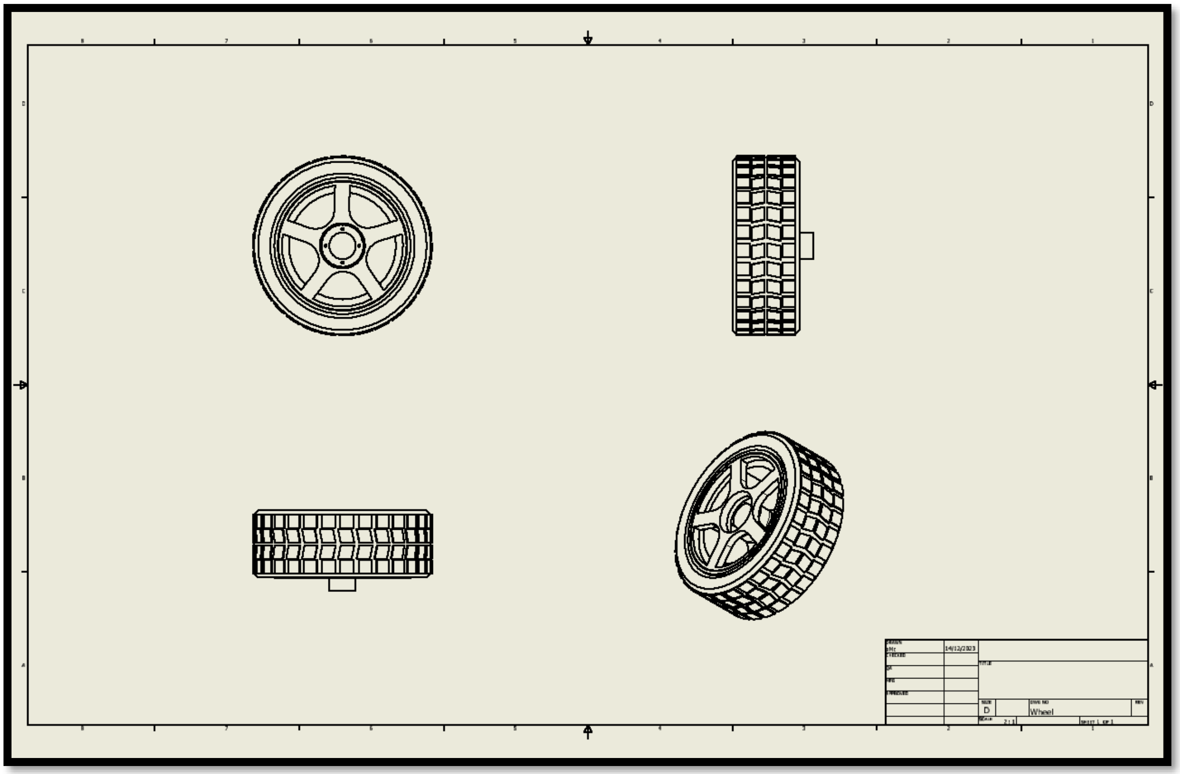
```
 }

//if Right Sensor is Black and Left Sensor is White then it
will call turn Right function
else if((digitalRead(R_S) == 1)&&(digitalRead(L_S) ==
0)){turnRight();}
//if Right Sensor is White and Left Sensor is Black then it
will call turn Left function
else if((digitalRead(R_S) == 0)&&(digitalRead(L_S) ==
1)){turnLeft();}

delay(10);
}

void forword(){  //forword
digitalWrite(in1, LOW); //Left Motor backword Pin
digitalWrite(in2, HIGH); //Left Motor forword Pin
digitalWrite(in3, HIGH); //Right Motor forword Pin
digitalWrite(in4, LOW); //Right Motor backword Pin
}
void backword(){ //backword
digitalWrite(in1, HIGH); //Left Motor backword Pin
digitalWrite(in2, LOW); //Left Motor forword Pin
digitalWrite(in3, LOW); //Right Motor forword Pin
digitalWrite(in4, HIGH); //Right Motor backword Pin
}
void turnRight(){ //turnRight
digitalWrite(in1, LOW); //Left Motor backword Pin
digitalWrite(in2, HIGH); //Left Motor forword Pin
digitalWrite(in3, LOW); //Right Motor forword Pin
digitalWrite(in4, HIGH); //Right Motor backword Pin
}
void turnLeft(){ //turnLeft
digitalWrite(in1, HIGH); //Left Motor backword Pin
digitalWrite(in2, LOW); //Left Motor forword Pin
digitalWrite(in3, HIGH); //Right Motor forword Pin
digitalWrite(in4, LOW); //Right Motor backword Pin
```

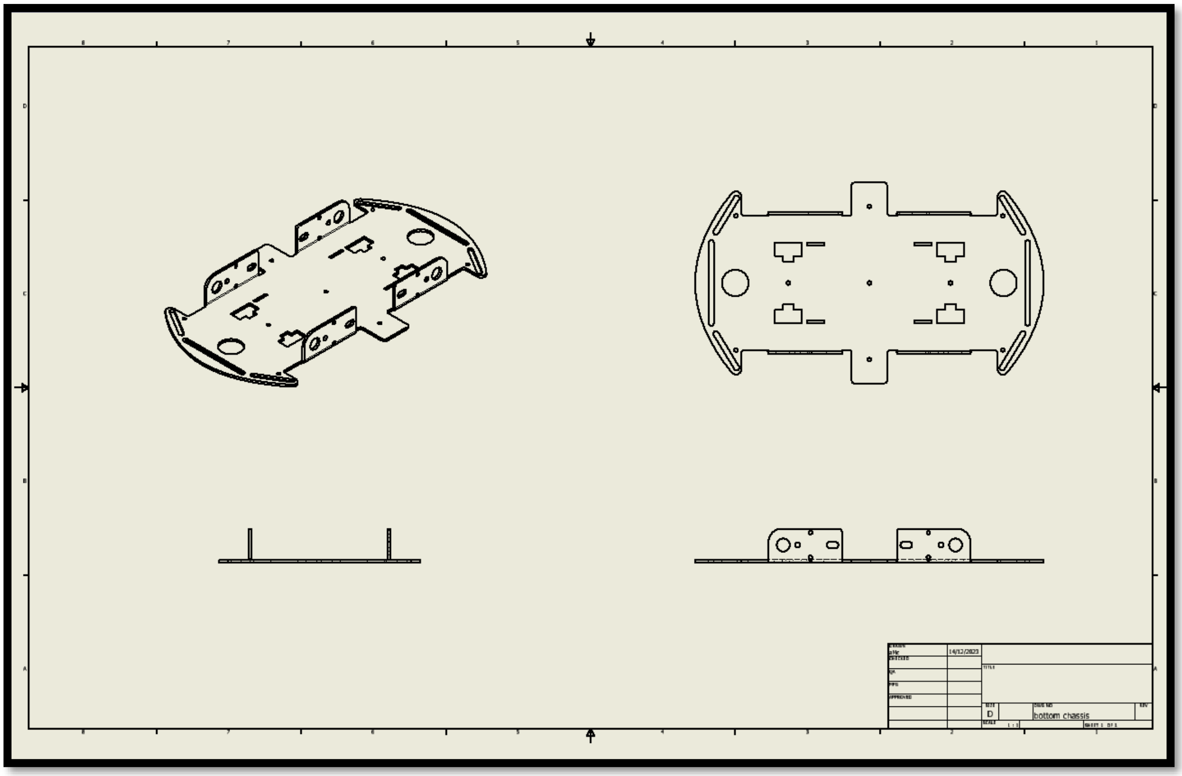
```
}
void Stop(){ //stop
digitalWrite(in1, LOW); //Left Motor backword Pin
digitalWrite(in2, LOW); //Left Motor forword Pin
digitalWrite(in3, LOW); //Right Motor forword Pin
digitalWrite(in4, LOW); //Right Motor backword Pin
}
```

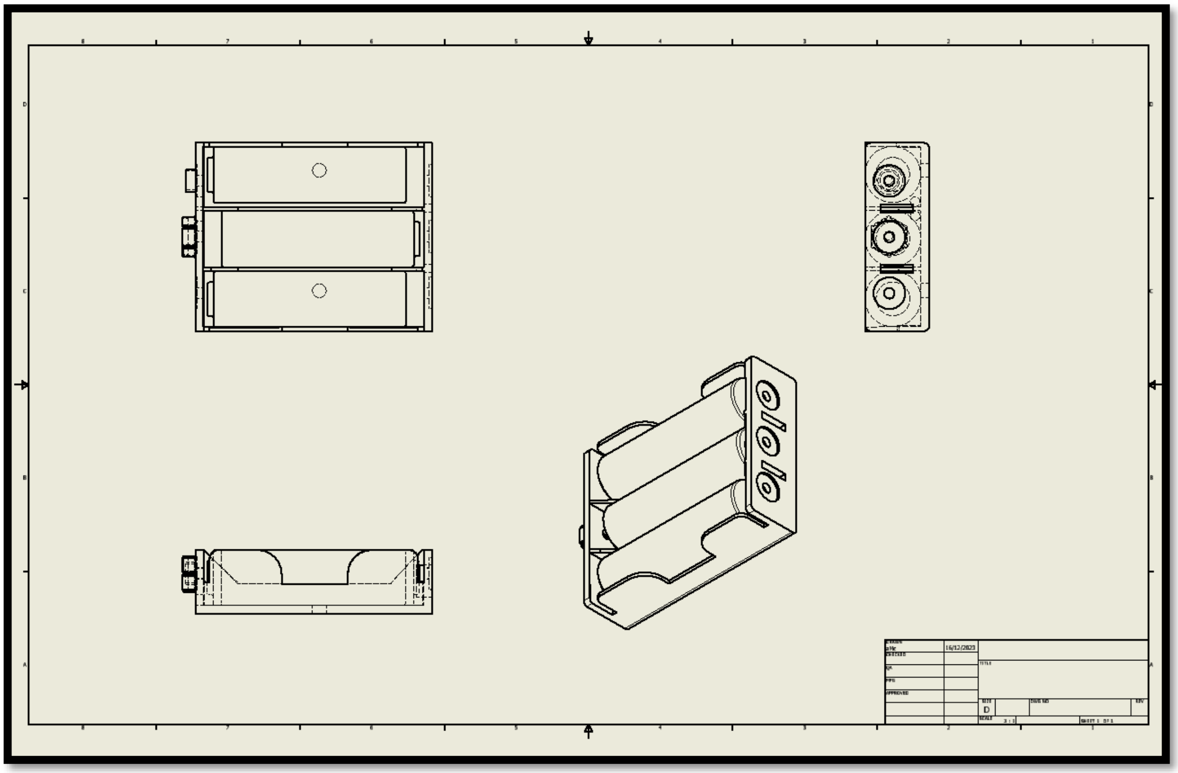
CAD-Drawings:

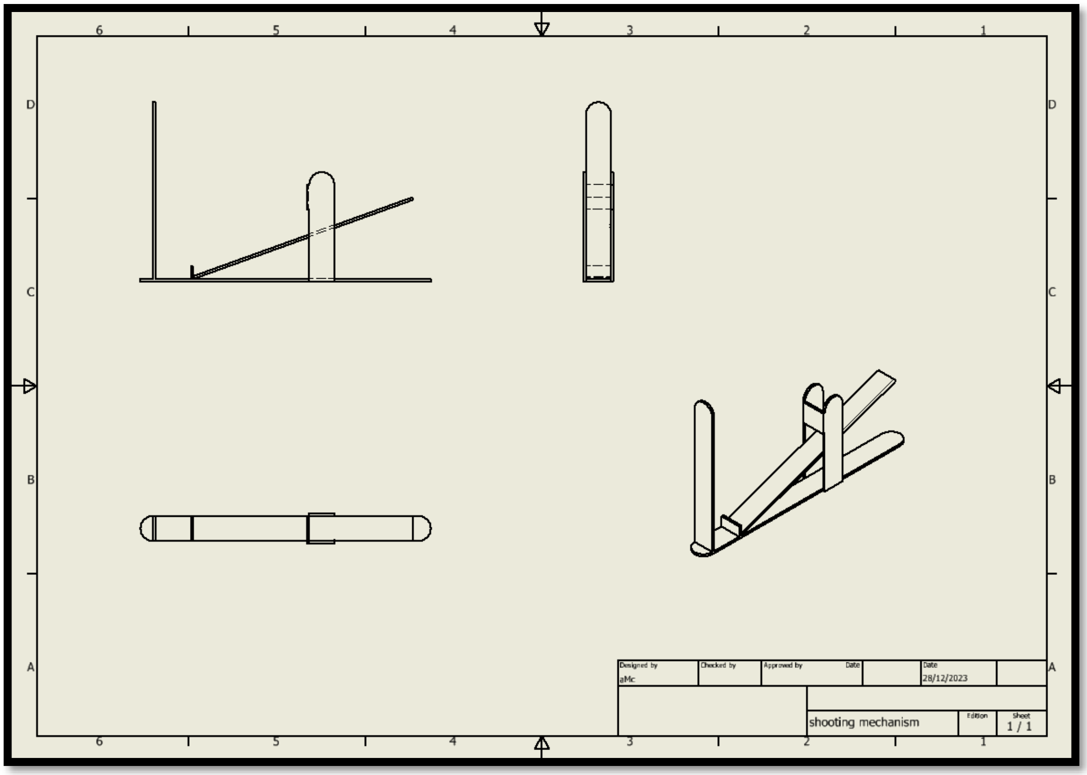
1. Working Drawings:



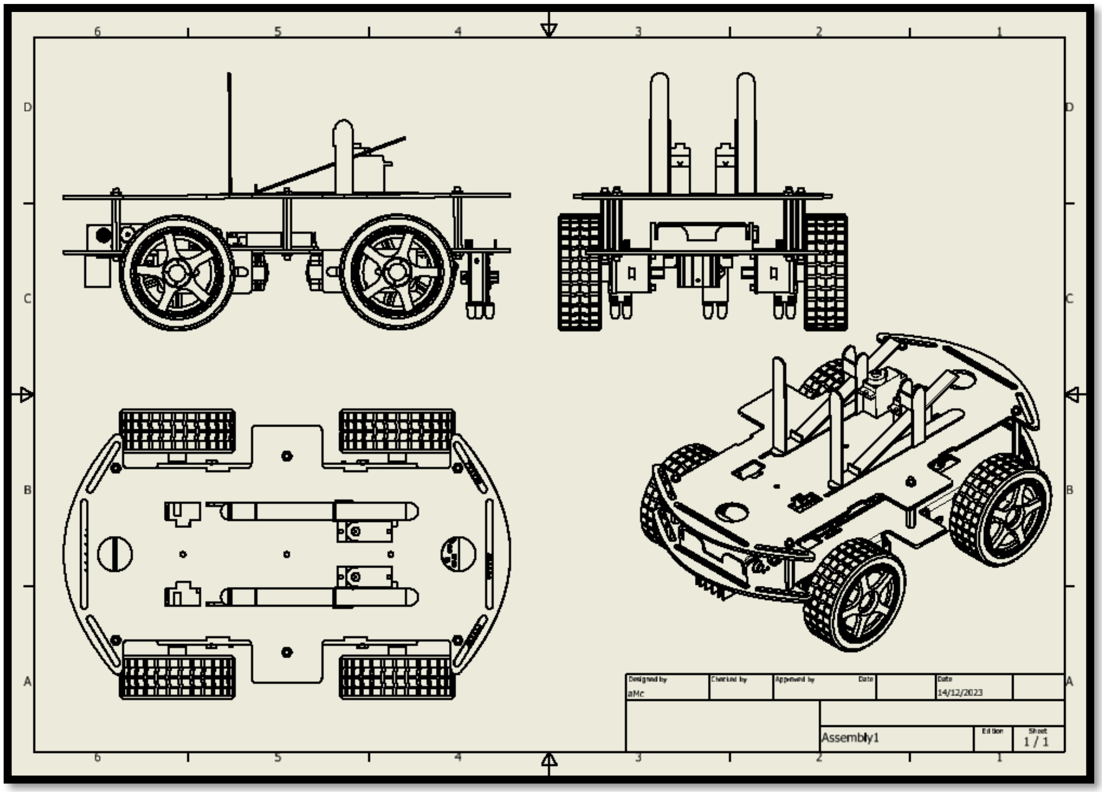



DRAWN
aMc
14/12/2023
CHECKED
QA
MFG
APPROVED
TITLE
SIZE
D
DWG NO
bottom chassis
REV
SCALE
1 : 1
SHEET 1 OF 1

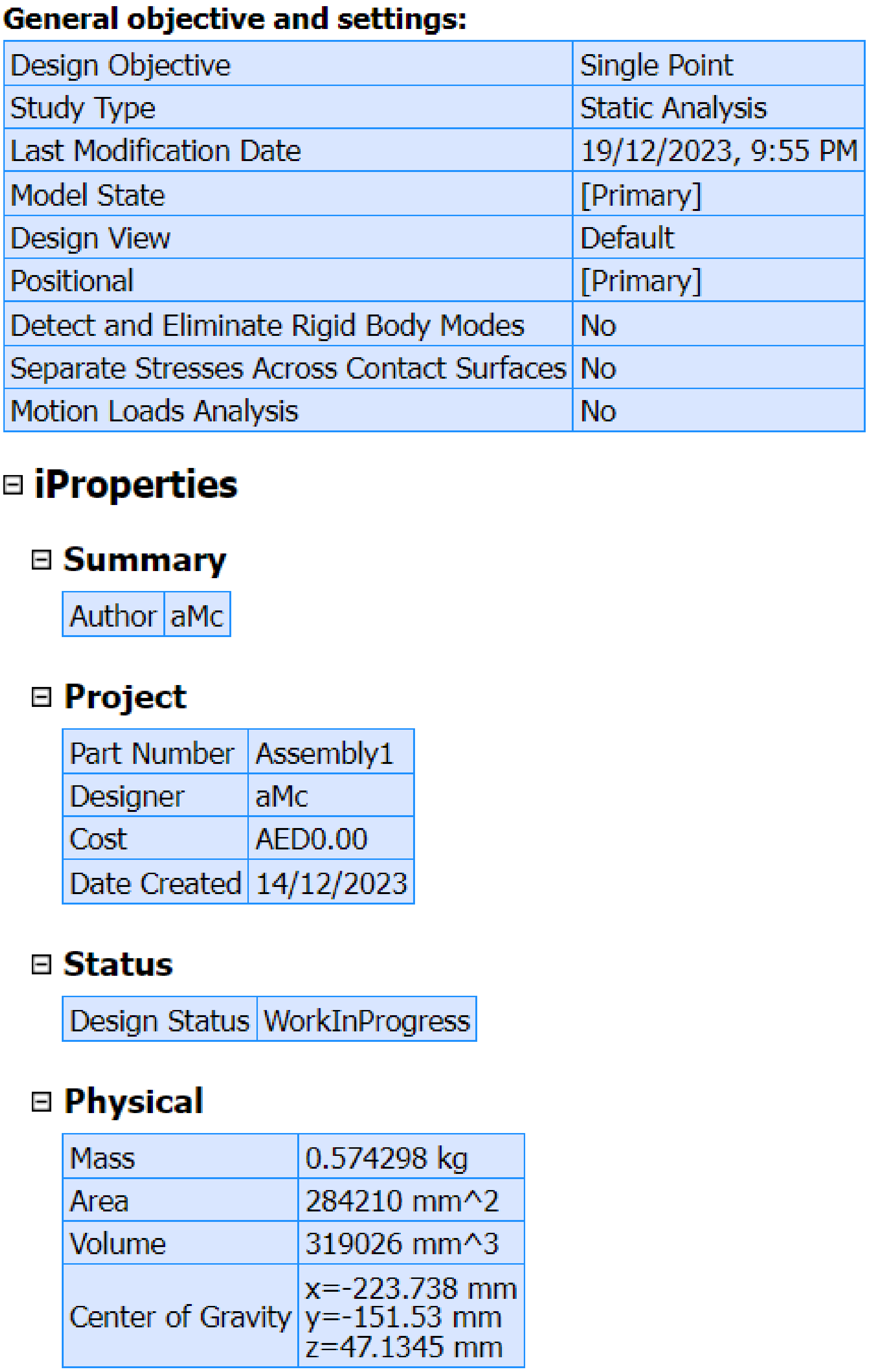

DRAWN
aMc
16/12/2023
CHECKED
QA
MFG
APPROVED
TITLE
SIZE
D
DWG NO
REV
SCALE
3 : 1
SHEET 1 OF 1



2. Construction Drawings:



3. Stress Analysis:



General objective and settings:

Design Objective	Single Point
Study Type	Static Analysis
Last Modification Date	19/12/2023, 9:55 PM
Model State	[Primary]
Design View	Default
Positional	[Primary]
Detect and Eliminate Rigid Body Modes	No
Separate Stresses Across Contact Surfaces	No
Motion Loads Analysis	No

⊟ iProperties

⊟ Summary

Author	aMc

⊟ Project

Part Number	Assembly1
Designer	aMc
Cost	AED0.00
Date Created	14/12/2023

⊟ Status

Design Status	WorkInProgress

⊟ Physical

Mass	0.574298 kg
Area	284210 mm^2
Volume	319026 mm^3

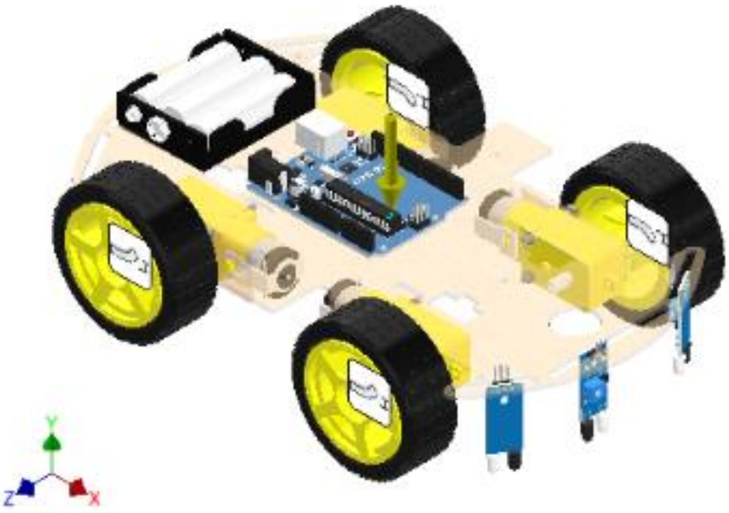
Center of Gravity	x=-223.738 mm
	y=-151.53 mm
	z=47.1345 mm

⊟ Operating conditions

⊟ Force:1

Load Type	Force
Magnitude	4.905 N
Vector X	0.031 N
Vector Y	-4.905 N
Vector Z	0.000 N

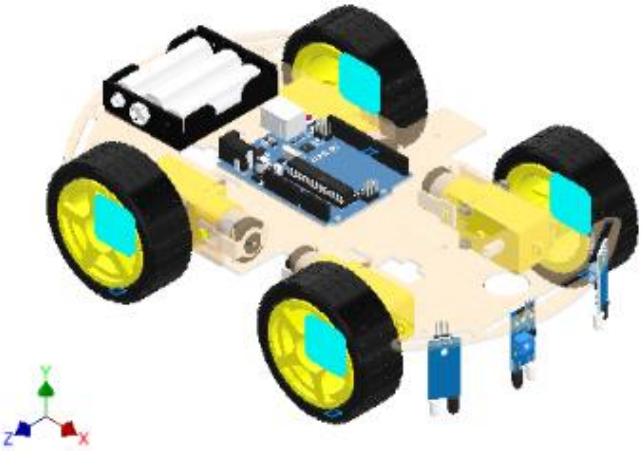
⊟ Selected Face(s)

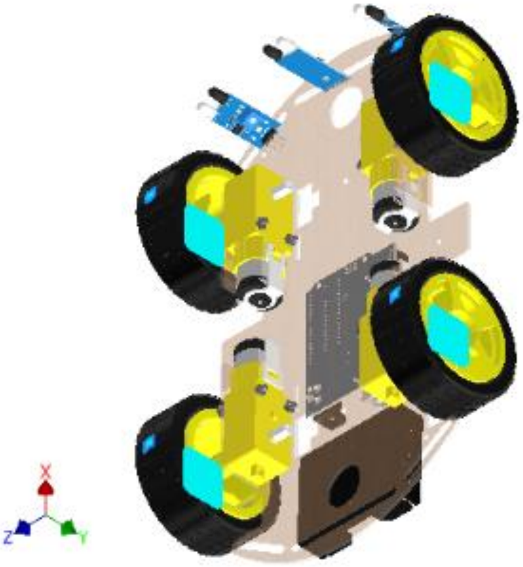


⊟ Fixed Constraint:1

Constraint Type	Fixed Constraint

⊟ Selected Face(s)





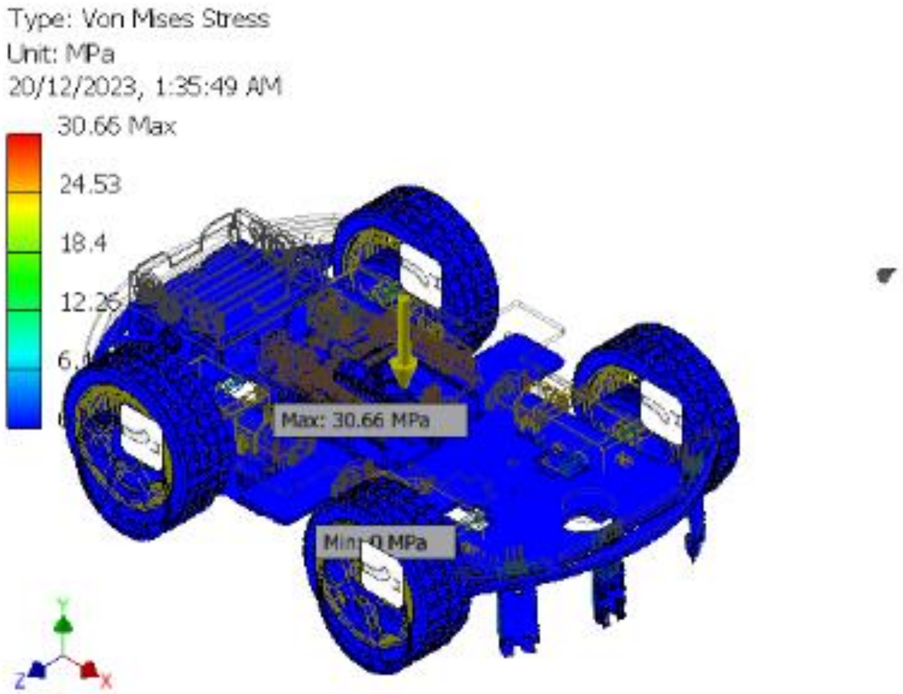
⊟ Reaction Force and Moment on Constraints

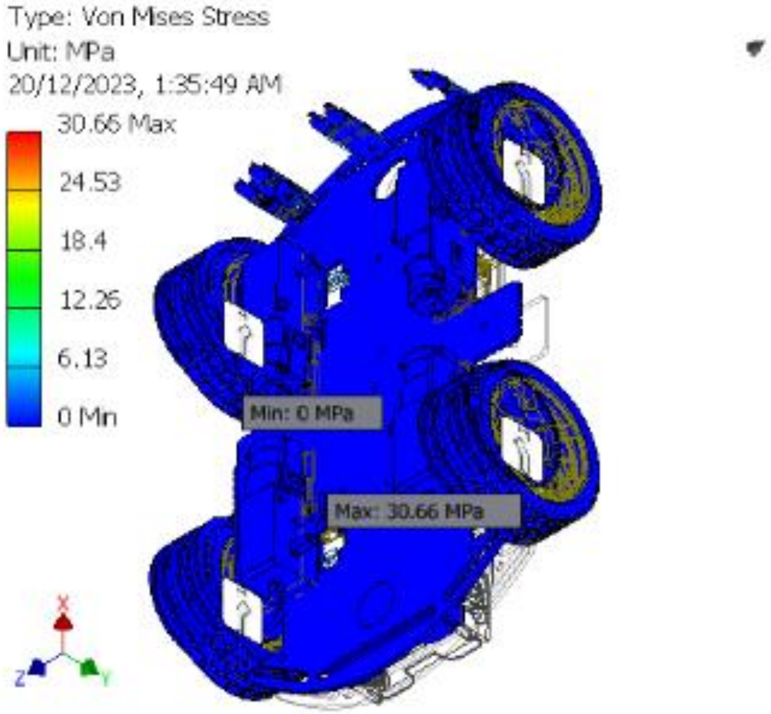
Constraint Name	Reaction Force		Reaction Moment	
	Magnitude	Component (X,Y,Z)	Magnitude	Component (X,Y,Z)
Fixed Constraint:1	0 N	0 N	0 N m	0 N m
		0 N		0 N m
		0 N		0 N m

⊟ Result Summary

Name	Minimum	Maximum
Volume	319026 mm^3	
Mass	0.574298 kg	
Von Mises Stress	0 MPa	30.6595 MPa
1st Principal Stress	-7.83849 MPa	34.7813 MPa
3rd Principal Stress	-35.2041 MPa	6.34203 MPa
Displacement	0 mm	0.124145 mm
Safety Factor	4.15926 ul	15 ul
Stress XX	-20.0501 MPa	20.1963 MPa
Stress XY	-17.4143 MPa	13.4885 MPa
Stress XZ	-5.7087 MPa	5.97731 MPa
Stress YY	-34.0677 MPa	33.5771 MPa
Stress YZ	-7.09307 MPa	5.81006 MPa
Stress ZZ	-10.8276 MPa	9.56987 MPa
X Displacement	-0.0063375 mm	0.0216413 mm
Y Displacement	-0.124139 mm	0 mm
Z Displacement	-0.00495701 mm	0.00555622 mm
Equivalent Strain	0 ul	0.00209482 ul
1st Principal Strain	-0.00000334338 ul	0.00191421 ul
3rd Principal Strain	-0.0017883 ul	0.00000279581 ul
Strain XX	-0.00100783 ul	0.000853712 ul
Strain XY	-0.00120769 ul	0.00161028 ul
Strain XZ	-0.000538554 ul	0.000355433 ul
Strain YY	-0.00142658 ul	0.00139815 ul
Strain YZ	-0.000817194 ul	0.000747075 ul
Strain ZZ	-0.000341036 ul	0.000361838 ul
Contact Pressure	0 MPa	123.546 MPa
Contact Pressure X	-58.6866 MPa	46.3282 MPa
Contact Pressure Y	-112.268 MPa	106.063 MPa
Contact Pressure Z	-28.243 MPa	40.2844 MPa

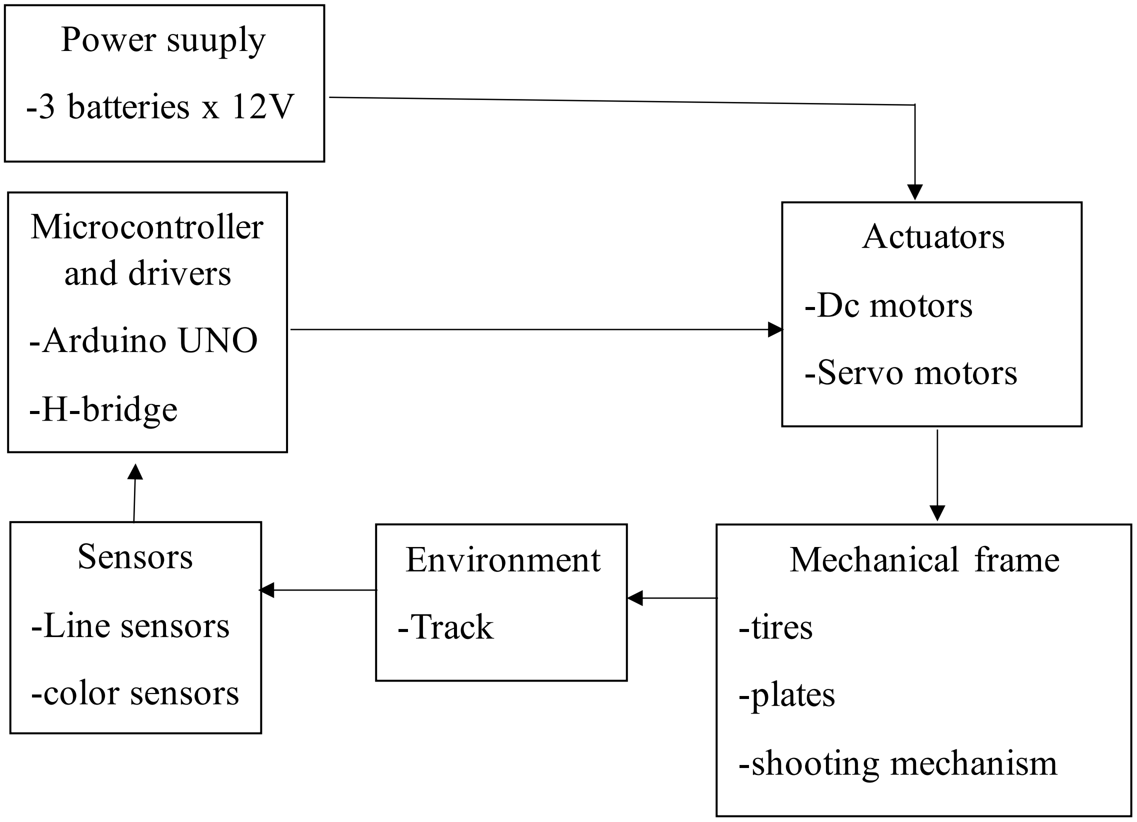
⊟ Von Mises Stress





System Integration and Performance Integration:

Mechatronics System Configuration Block diagram



References:

- Arduino Official website: www.arduino.cc
- Commponent search engine: https://componentsearchengine.com/
- Future electronics (for data sheets): https://store.fut-electronics.com/
- Ram electronics (for data sheets): https://ram-e-shop.com/
- Autodesk instructables: https://www.instructables.com/