INDIAN INSTITUTE OF TECHNOLOGY, DELHI

INTERNSHIP REPORT

DEVELOPMENT OF ROBOMUSE 3XT

Ayush Goel
Department of Mechanical Engineering
Thapar University, Patiala

Shubham Gupta
Department of Mechanical Engineering
Thapar University, Patiala

Supervisor:
Prof. Subir Kumar Saha
Department of Mechanical Engineering
Indian Institute of Technology,
Delhi

Mentor:
Sasanka Sekhar Sinha
Department of Mechanical Engineering
Indian Institute of Technology,
Delhi

Mentor:
Rohit Patel
Department of Mechanical Engineering
Indian Institute of Technology,
Delhi

DECLARATION

This is to certify that Ayush Goel an undergraduate student of Thapar University, Patiala Mechanical Engineering Department, (Roll Number: 101409012), had taken up a project 'Development of ROBOMUSE 3XT' under my guidance, during a period of six months as a part of an internship programme from Jan 10,2017 to July 09,2017. This certifies hisenthusiastic participation and completion of the project, within the stipulated time.

Date: ___________________

Prof. S.K. Saha

Professor and Head

Department of Mechanical Engineering

Indian Institute of Technology Delhi

DECLARATION

This is to certify that Shubham Gupta an undergraduate student of Thapar University, Patiala Mechanical Engineering Department, (Roll Number: 101409034), had taken up a project 'Development of ROBOMUSE 3XT' under my guidance, during a period of six months as a part of an internship programme from Jan 10,2017 to July 09,2017. This certifies his enthusiastic participation and completion of the project, within the stipulated time.

Date: __________________

Prof. S.K. Saha

Professor and Head

Department of Mechanical Engineering

Indian Institute of Technology Delhi

Acknowledgement

We take this opportunity to express our sincere thanks to Prof. Subir Kumar Saha, Department of Mechanical Engineering, I.I.T Delhi, for giving us the opportunity to work on RoboMuse 3XT and his timely inputs on the projects. We consider ourselves extremely lucky to have been associated with someone like Prof. Saha whose personality inspires people to imbibe his morality, style of work and professionality. It has been a wonderful learning experience working on the project.

We are also deeply indebted to Mr. Sasanka Sekhar Sinha and Mr. Rohit Patel for their invaluable time and inputs during the complete course of the project semester. Without their constructive criticism from time to time, it would have been difficult for us to come so far.

We would also like to express our gratitude to Dr. Ashish Singla, Mechanical Engineering Department, Thapar University, Patiala for his guidance as the faculty advisor for the project semester.

We would also like to thank Mr. Abdullah Aamir Hayat, Mrs. Shraddha Chaudhary, Mr. Riby Abraham Boby, Mr. Arun Sharma, Mr. Vishal Abhishek, Mr. Rajeevlochana G. Chittawadigi and other members of Program in Autonomous Robotics Laboratory for their timely guidance and ensuring smooth going of our project work and stay at IIT Delhi.

Last but not the least, we would like to thank Mr. Amal George and other members of the Robotics Club, IIT Delhi for their suggestions and for lending their equipment and workspace for the development of the project.

Contents

List of Figures

List of Tables

1 Introduction

A semi-autonomous mobile robot RoboMuse 3X has been developed by the students of robotics club at IIT Delhi that is capable of traversing a pre-defined path autonomously. It is also capable of detecting obstacles and differentiate between a known and unknown object.

The aim of this project is to make a similar robot,RoboMuse 3XT, with added functionalities.

Figure 1: RoboMuse 3XT

1.1 Objectives

RoboMuse 3XT was to be designed to have the following capabilities:

1. A semi-autonomous mobile robot capable of being operated either manually or autonomously.
2. The robot should have the capability of live video surveillance of indoor environments.

3. It should have the capability of carrying a payload of minimum 15 Kgs.
4. Expandability for future research.
5. It should be able to detect obstacles and humans.
6. It should interact with humans whenever it senses them.
7. It should be able to guide people in environments unknown to them.

The development of this robot required review of various aspects of an autonomous mobile robot that includes mechanical design, perception, sensor selection, localization, path planning and navigation.

1.2 Specifications

Table 1: Specifications of the Robot

S.No	Description	
1.	**Height**	720 mm
2.	**Width**	440 mm
3.	**Length**	700mm
4.	**Sensing Capability**	Detect obstacles up to 2000 mm
5.	**Size of Payload**	500 x 350 (l x b) mm x mm
6.	**Weight of the RoboMuse Assembly**	25 Kgs
7.	**Payload carrying capacity**	15 Kg
8.	**Range of Movement**	Either X or Y but not together
9.	**Workspace**	Rectangular configuration wherein the RoboMuse can move

2 Design

2.1 Introduction

Before starting the design of a mobile robot, one should be sufficiently aware of the requirements the robot needs to fulfil. In our case we need to design the mobile robot capable of:

1. Working on smooth indoor surfaces
2. Carrying a payload of 15Kg
3. Carrying multiple sensors on board
4. Being modified easily to be used in future research and
5. Negotiating slopes

The final CAD generated on SOLIDWORKS is shown in Figure 2.

Figure 2: CAD of ROBOMUSE 3XT

2.2 Methodology

The mechanical design of the robot was designed in two stages:

In the first stage, a preliminary design was created on SOLIDWORKS that included the initial design of the base, placement of various electrical components, motor mounting and placement of sensors.

In the second and final stage, improvements were made in the initial design on the basis of agility, appearance and payload capacity.

The CAD of chassis of first stage is shown in Figure 3.

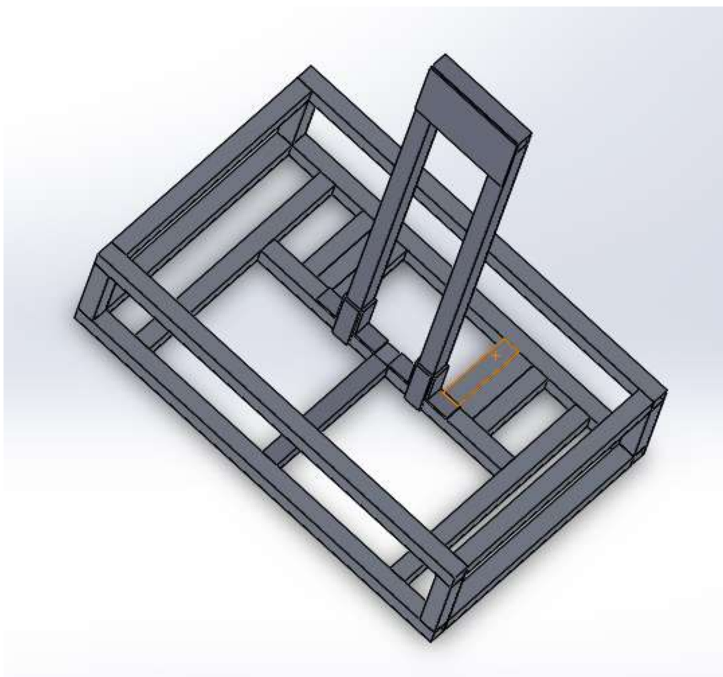


Figure 3: Cad of First Stage of Mechanical Design

2.2.1 First stage development

Before starting the design process, we need to know certain specifications of the robot like dimensions, shape, wheel configuration etc. The following subsections explain in detail the process we used to find these specifications.

2.2.1.1 Robot Configuration

2.2.1.1.1 Dimensions of the robot

The outer base dimensions of ROBOMUSE 3X are: 620 x 406.4 (in mm).

The dimensions of ROBOMUSE 3X were sufficient enough to carry all the electronics and sensors on board. So, there was no need to increase the dimensions of ROBOMUSE 3XT. The outer dimensions weren't decreased as well so as to provide scope for future additions.

After the CAD model was complete, the outer base dimensions of the ROBOMUSE 3XT came out to be: 620 x 400 (in mm).

2.2.1.1.2 Material used in robot

The chassis of ROBOMUSE 3XT was fabricated with hollow Aluminium rods (square cross-section with side of 25.4 mm) welded together by TIG Welding. Hollow Aluminium rods were used because they were easily available in standard sizes and they have good strength-to-weight ratio. The grade of Aluminium used was **Al-6061** with composition as shown in Table 2.

Table 2:Composition of ***Al- 6061***

Component	Weight %
Aluminium	95.8-98.6
Magnesium	0.8-1.2
Silicon	0.4-0.8
Others (Manganese, Tin, Copper)	Less than 2%

2.2.1.1.3 Shape of the robot

The shape of a mobile robot is of great importance and can have an impact on the robot's performance. As this robot has to be used in future for research purposes, the shape of the robot should be selected so that it provides maximum space for the components and the payload.The area available to place components in this configuration is 248000 mm^2. Considering atriangular configuration, to achieve the same area, length had to be doubled keeping the width same and vice versa. The reason for not using other configurations (circular, hexagonal, pentagonetc.) was the manufacturing complexity in getting them fabricated in the desired shape because of the use of hollow tubes.

Thus, the dimensions, material and shape of the base of the robot was finalisedupto this step.The final base representation is shown in Figure 4.

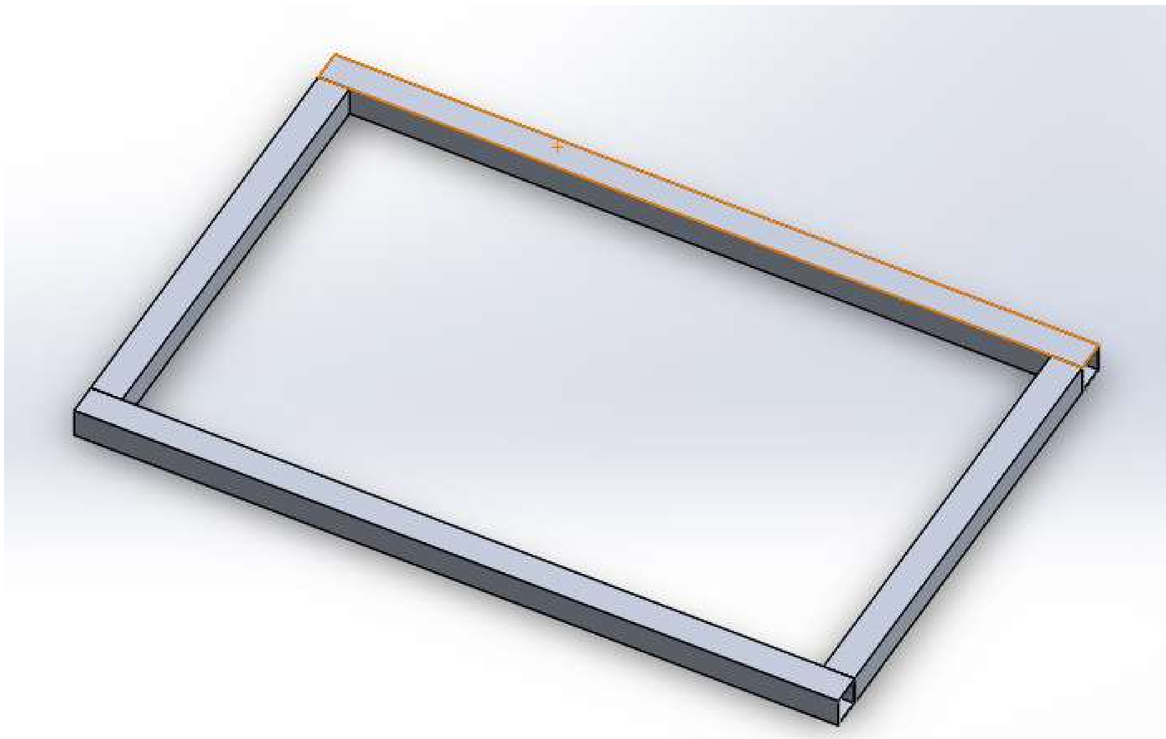


Figure 4: Base Representation

2.2.1.2 Wheel configuration

Before designing the mechanical system to support wheels and motors, wheel configuration has to be finalised. Wheeled robots are almost always designed so that all wheels are in ground contact at all times. Three wheels are sufficient to guarantee stable balance, although two-wheeled robots can also be stable. When more than three wheels are used, a suspension system is required to allow all wheels to maintain ground contact at all times [1].

In case of a two-wheeled robot, static stability can be achieved if the centre of mass of the whole system is below the wheel axle. However, for a three-wheeled robot, static stability is achieved when the centre of gravity of the system is within the triangle formed by the ground contact points of the wheels. To improve stability, more wheels can be added but as mentioned above if the number of wheels is more than three, a suspension system is required to ensure contact of wheels with ground. Suspension system adds to the complexity of the system, therefore a three-wheeled system was finalised. For a three-wheeled system, various drive systems are possible which are discussed in detail in [2].

The wheel configuration used in this project consists of two independently driven wheels in the rear with one castor wheel in the front as shown in Figure 5.

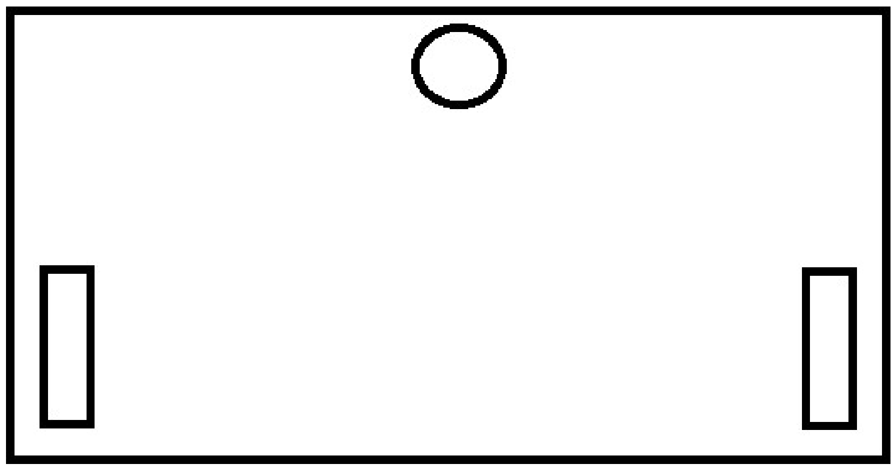


Figure 5: Three Wheel Configuration

The advantages of this configuration are:

- Simple mechanism that allows only two motors to drive the robot. Each motor powers the two rear wheels separately.
- No suspension needed because three wheels are always in contact with ground at all times.
- The omni-directional behaviour: The robot can move to a certain position in space and achieve a certain orientation.
- Zero turn radius: To change the orientation, the two wheels have to be simply rotated in the opposite directions.

The disadvantages of this configuration are:

- Despite its simplicity, the controllability is rather difficult. For a straight-line motion, the drive wheels need to be turned at exactly the same rate which is difficult to achieve because of slight difference in the motors, friction differences in the wheel-ground interface, and misalignment of wheels.

As the wheel configuration was finalised, CAD was updated to accommodate the wheels. This is shown in Figure 6.

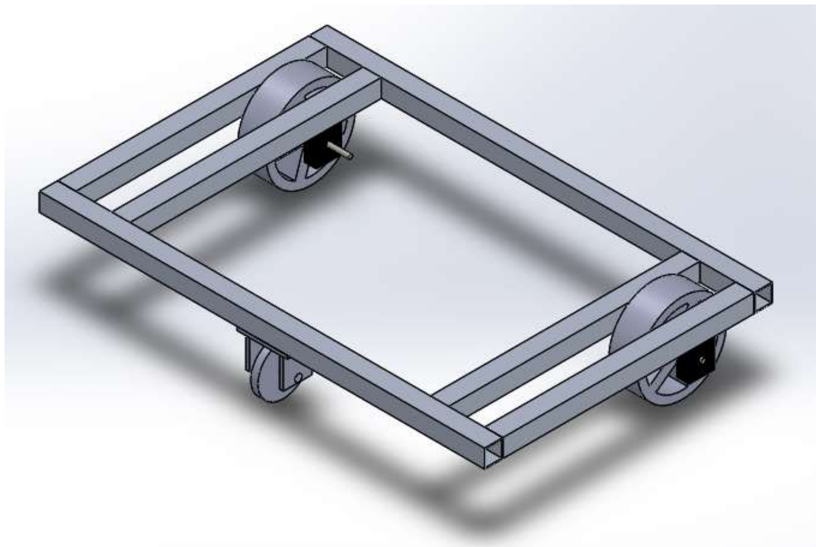


Figure 6: Base with wheels and castor

2.2.1.3 Drive system

Since there are two separate motors for the two wheels, a simple drive mechanism is used in the assembly as shown in Figure 7.

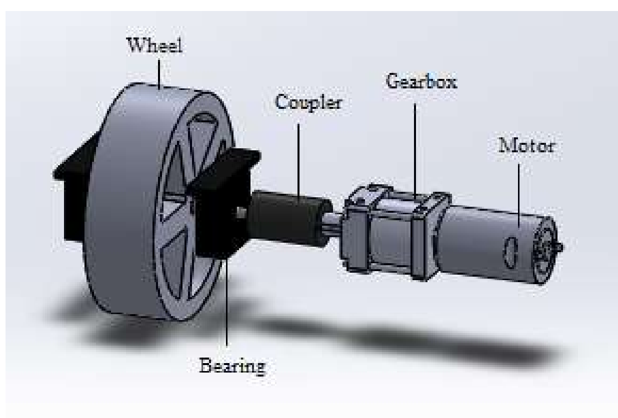


Figure 7: Drive System

In the above drive system, motor is mounted directly to the gearbox which is coupled with the wheel for power transmission. The gearbox needs to be attached to the assembly with the help of screws and bearings with the help of nuts and bolts. The updated CAD is shown in Figure 8.

This base will be covered with acrylic sheet to support battery and motor driver.

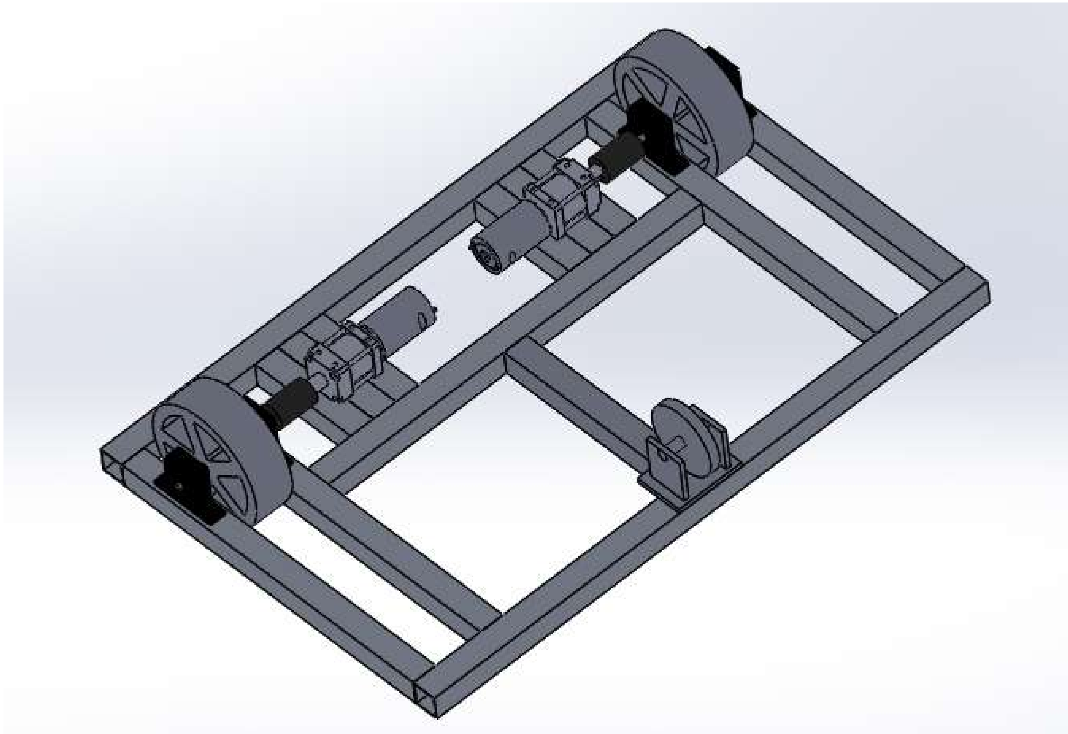
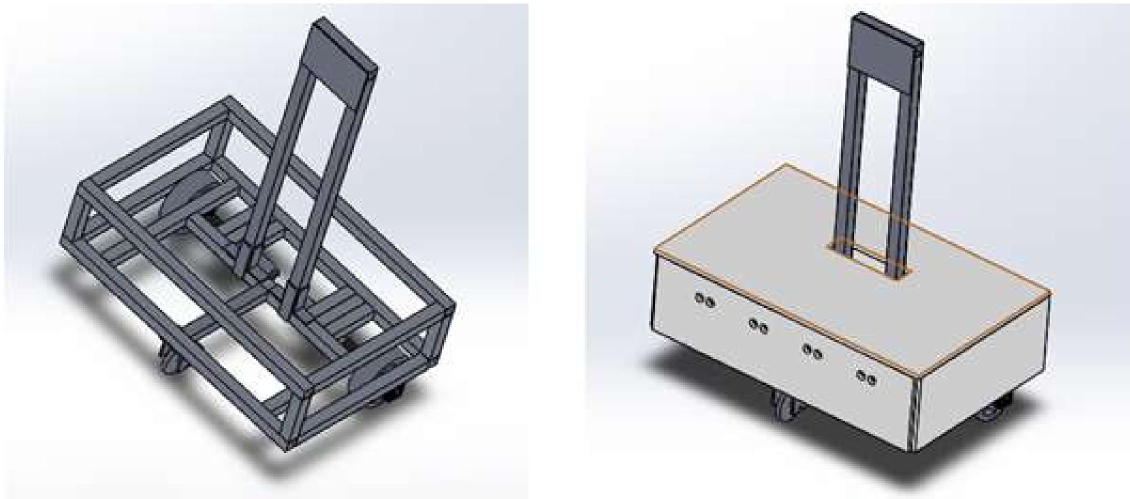


Figure 8: Base with Motors and Gearbox

2.2.1.4 Final Structure of First Stage

The final structure of the first stage is shown in Figure 9. As can be seen in the figure, there is a top subassembly to support the payload that will be kept on the robot. The vertical rods that originate from the base support a rectangular plate to hold the interactive screen and the camera. The vertical rods are fixed to the base assembly with the help of L-clamps and a rectangular plate.

The sides of this robot would be covered with PVC sheets. The material to cover the top will beacrylic because it has to support payload as well.The same structure covered with acrylic sheets is also shown in Figure 9.



(a) (b)

Figure 9: (a) Final Structure of First Stage

(b) Structure Covered with Acrylic

2.2.1.5 Summary of First Stage development

The flowchart of the design process of first stage is shown in Figure 10.

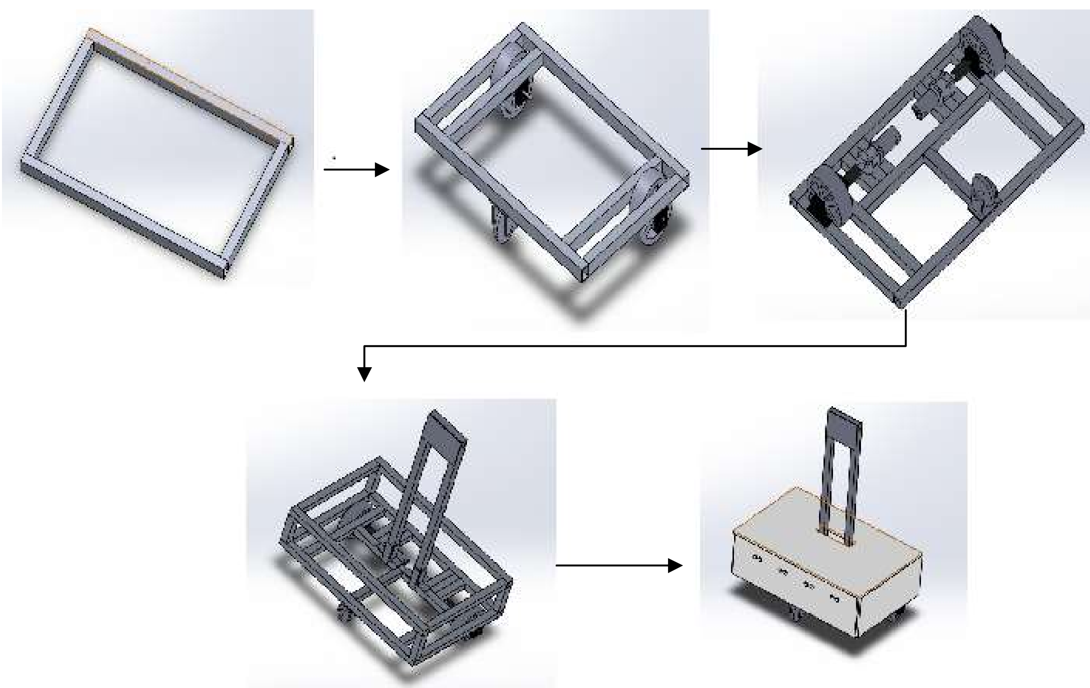


Figure 10: Flowchart of the Design Process

2.2.2 Second Stage development

Design process involves design, analysis and redesign of the system. After the CAD of first stage was developed, various issues were identified after analysis which are discussed in the subsection. The redesign of the system is further discussed in the next subsection.

2.2.2.1 Various issues identified

1. As can be seen in Figure 9, the vertical rods rise from the centre of the base assembly. After the full robot will be manufactured, there would be a lot of electronic components attached to the base of assembly. So in case the robot is to be transported, the vertical supports wouldn't be removed easily.

2. Welding three rods together and then mounting motor on those has several problems. When the robot moves, motor applies a torque on the rods due to which there is stress concentration in welds. This may lead to failure in long run. Moreover if the motors needs to be removed from the structure, remounting may lead to alignment problems.

3. The castor wheel comes with a plate that has to mounted on some solid structure. As can be seen in Figure 9, only a single rod is used to support castor which is not enough. Also towards the front portion of base assembly, batteries will be placed. Since the batteries are quite heavy, an additional support would be needed to support the acrylic.

4. The top subassembly has to carry a payload. As can be seen in Figure 9, the acrylic that will be mounted over the top has no support at the centre. Acrylic sheet is only supported by the boundaries of the top subassembly. So in case the load increases at the centre, bending may take place.

2.2.2.2 Changes implemented

The final CAD of the base and chassis of the robot is shown in Figure 11 and Figure 12 respectively.

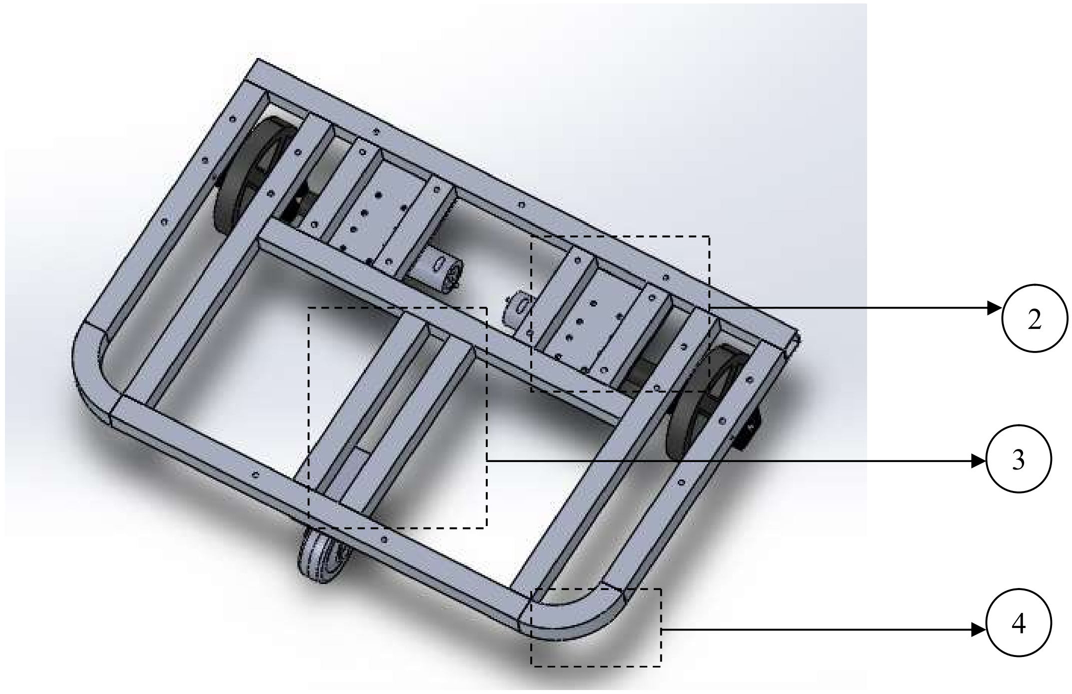


Figure 11: Base Design of the Final Model

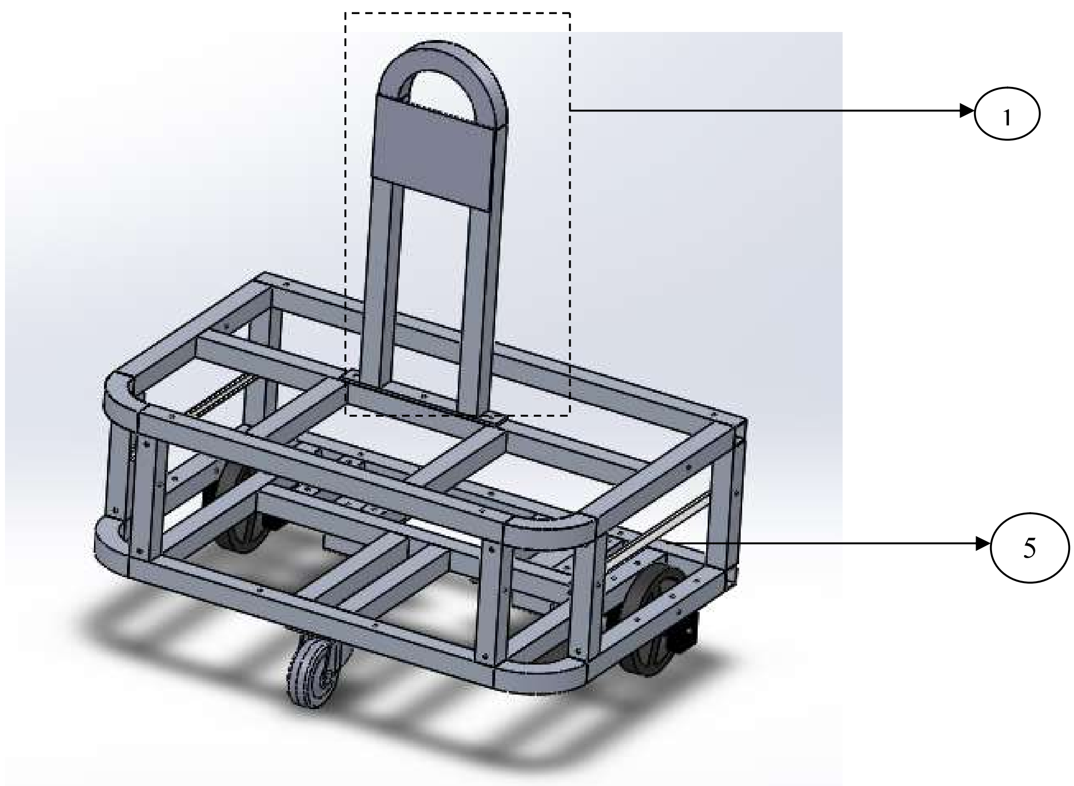


Figure 12: Chassis Design of the Final Model

The various changes implemented in the design are discussed in Table 3.

Table 3: Various Changes Implemented

S. No.	Initial Model Feature	Final Model Feature	Reason to change
1	Vertical road rises from the base	Separate subassembly	1. Easy removal in case of transportation 2. More space available for payload 3. Added support to top subassembly
2	Three rods welded to mount a motor	Separate plate for motor mounting	1. No stress concentration 2. No need to remove motor screws to remove the motor 3. Easy alignment
3	Single rod to support castor wheel	2 rods to support castor wheel	1. Rigid mounting of castor wheel 2. Added support to acrylic sheet to bear the stresses induced due to heavy batteries. 3. Increases payload capacity
4	Sharp edges at front	Curved edges at front	Aesthetic reasons
5	Electronic components placed at base subassembly	Electronic components placed on separate slider sheet	1. Easy separation between electronic components and mechanical structure 2. Increased space at base subassembly

3 Static Studies

Now that the CAD was finalised, stress studies were performed on the base assembly, top assembly which carries the payload and on the full chassis of the robot. The results are shown in figures below.

3.1 Analysis on Base Assembly

3.1.1 Stress Analysis

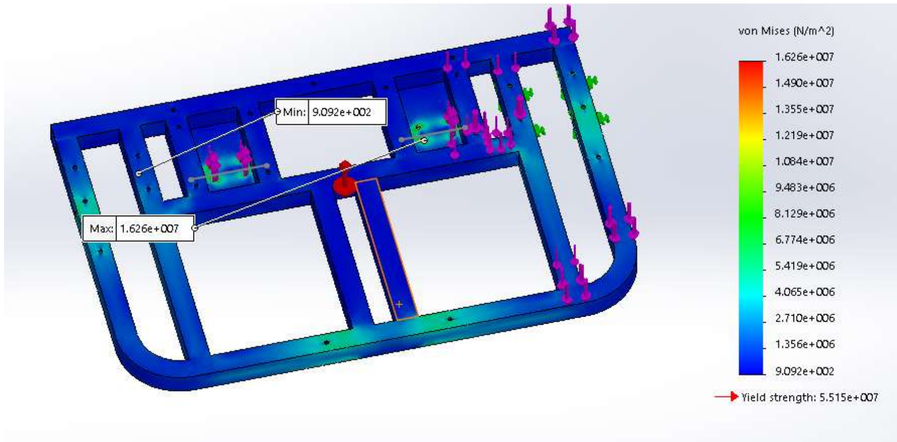


Figure 13: Stress Analysis on Base Assembly

Maximum induced stress: $1.626*10^7$ N/m^2

Yield strength of material: $5.515*10^7$ N/m^2

Factor of Safety: 3.39

3.1.2 Displacement Analysis

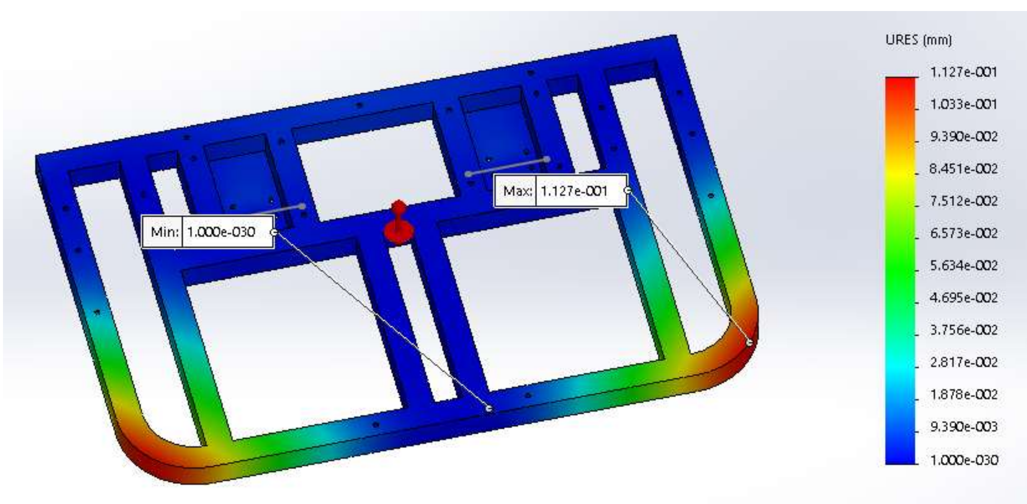


Figure 14: Displacement Analysis on Base Assembly

Maximum Displacement Induced: $1.127*10^{-1}$ mm

3.2 Analysis on Top Assembly

3.2.1 Stress Analysis

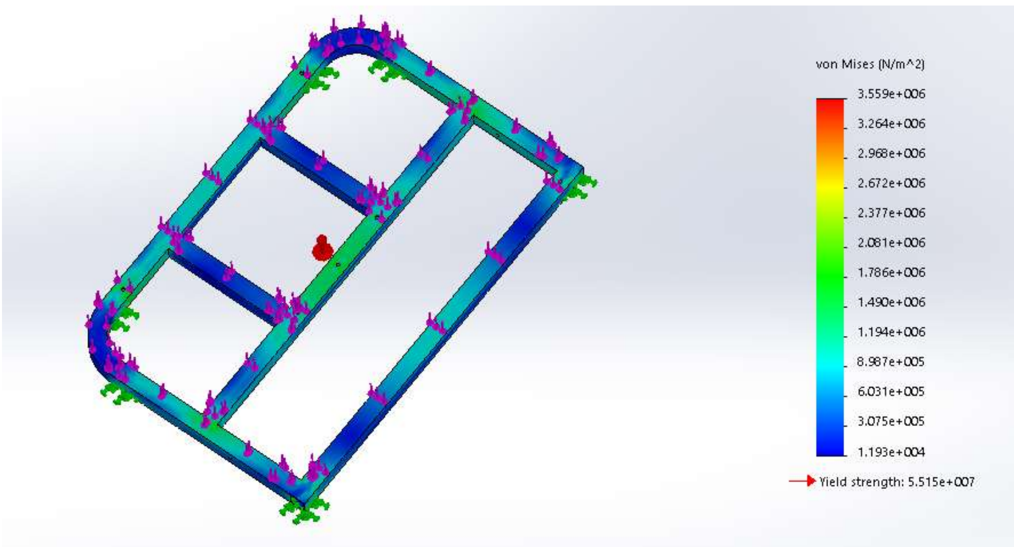


Figure 15: Stress Analysis on Top Assembly

Maximum induced stress: $3.559*10^6$ N/m^2

Yield strength of material: $5.515*10^7$ N/m^2

Factor of Safety: 15.49

3.2.2 Displacement Analysis

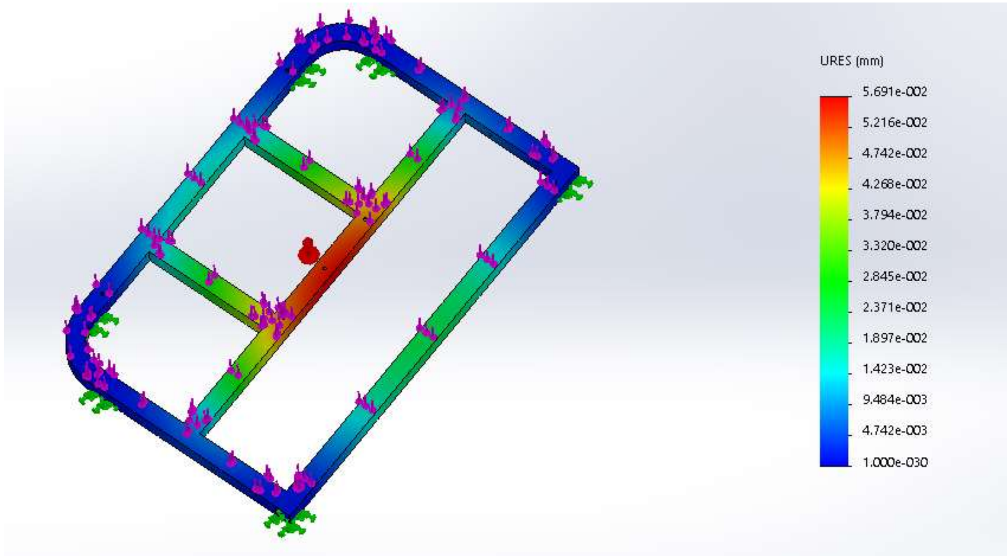


Figure 16: Displacement Analysis on Top Assembly

Maximum Displacement Induced: 5.691*10^{-2} mm

3.3 Analysis on Full Chassis

3.3.1 Stress Analysis

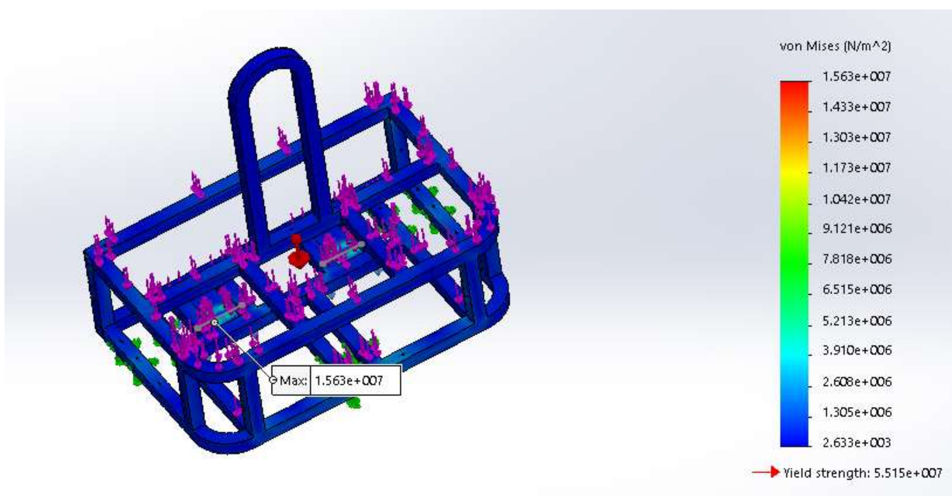


Figure 17: Stress Analysis on Full Chassis

Maximum induced stress: 1.563*10^7 N/m^2

Yield strength of material: 5.515*10^7 N/m^2

Factor of Safety: 3.52

3.3.2 Displacement Analysis

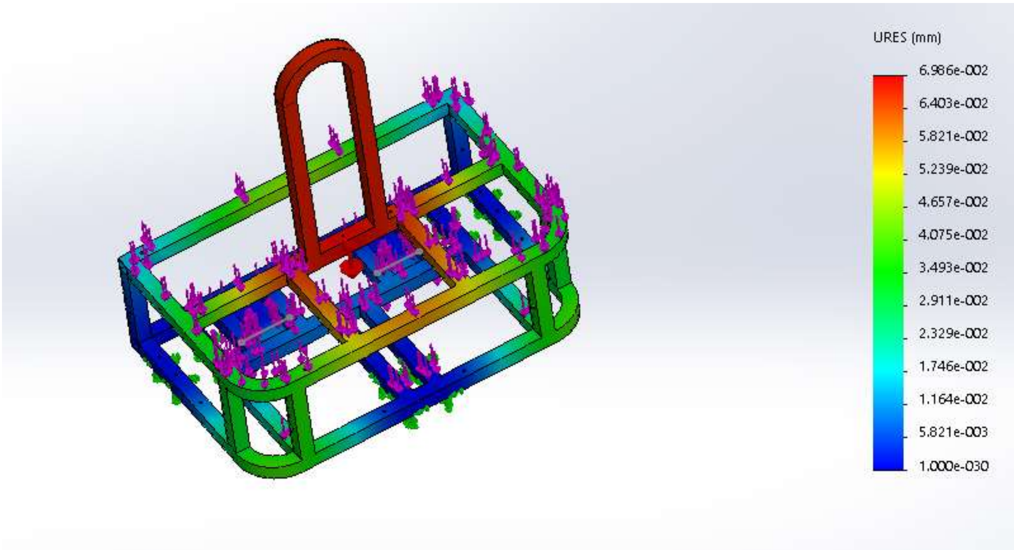


Figure 18: Displacement Analysis on Full Chassis

Maximum Displacement Induced: $6.986*10^{-2}$ m

4 Component Selection

4.1 Motors

The selection of a motor for an application depends on the torque and speed requirements of the application keeping in mind several other motor parameters. The various parameters to be considered while selecting a motor are as follows:

- Nominal Voltage
- No Load RPM
- Power Rating
- Stall Torque
- Stall Current
- Speed – Torque characteristics
- Torque vs current characteristics

After the RoboMuse assembly was developed, the weight of the entire assembly was measured and an approximation was made to have an estimate of the overall mass of the assembly. With some researching and the recommendations from the members of the Robotics Club, IIT Delhi for the different motors available in the market, BaneBots LLC was selected and the task was to choose the appropriate motor that fits the application perfectly.

4.1.1 Motor Calculations

Before selecting the motors, the task is to calculate the torque and the rpm required for our robot. For calculating the torque requirements and the rpm requirements, some parameters had to be defined. One of the parameteris the mass of the overall assembly after all the components have been integrated. So, data of weight of all the components were gathered from different sources and an approximation was made based upon this data and the cad model. The robot was supposed to carry a payload apart from its weight. So, an initial estimate of 10kg was made. The next task was to decide about the maximum velocity of the robot and thereby decide the time required to reach such velocities. This was meant to calculate the maximum acceleration that the robot could achieve.

Following are the parameter values that were finalised.

- Mass of robot: **25 kg**
- Payload capacity: **10 kg**
- Maximum Velocity: **1 m/s**
- Time required to attain maximum velocity: **500 ms**
- Diameter of wheel: **127 mm (0.0635 m)**

After the above values were finalised, the torque required by the assembly was calculated. The calculation was performed as follows.

4.1.1.1 Torque Calculation

Torque Required = m x a x r

where,

m = mass of the robot

a = maximum acceleration

r = radius of the wheel

Thus, Torque Required = 35 x (1 / 0.5) x 0.0635 Nm

= **4.445 Nm**

= **627 oz-in**

Then the angular velocity requirement was calculated as follows.

4.1.1.2 No Load Angular Velocity

No load Angular Velocity = v / r

where,

v = maximum velocity to be achieved by the robot

r = radius of the wheel

Thus, No load Angular Velocity = 1 / 0.0635 rad/s

= **15.75 rad/s**

= **150.3 RPM**

4.1.2 Motor Selection

After the torque requirement and angular velocity requirement were calculated, the task was to select a motor from a list of available motors that fitted our requirements perfectly. Since the robot was to be controlled using two 12V batteries, it was decided to select a 12V motor. From the BaneBots website, a list of all the motors running on 12V supply was made. An appropriate motor had to be selected from this list based on our requirements.

Table 4: Available BaneBotsMotors [3]

MOTOR	STALL TORQUE (OZ-IN)	RPM	STALL CURRE NT	EFFICIE NCY
RS395	16.65	15500	15	76%
RS540	32.64	17200	37	70%
RS550	69.16	19300	85	76%
RS555	29.16	7750	15	67%
RS775	61.1	8700	30	77%

The above list consists of motors that run on 12V supply. From this list RS550 was straightaway rejected as its stall current was very high. From the remaining list, a motor had to be selected based on the torque requirement and angular velocity requirement. From this list two motors were selected – RS540 and RS 775 based on requirements. First it was decided to go with RS775 motor as it had very high stall torque but this motor was not available at the BaneBots website at that time. So, it was decided to go with RS 540 motor. The power requirements of the motor were also calculated.

The specifications of the BaneBots RS-540 motor are given in Figure 19.

Specifications

Model	: RS-540 12V	Weight	: 5.4 oz
Operating v	: 4.5v - 12v	Length of motor	: 1.97 in
Nominal v	: 12v		
No Load RPM	: 17200	Diameter	: 1.41 in
No Load A	: 1A	Shaft Diameter	: 0.125 in
Stall Torque	: 32.64 oz-in	Shaft Length	: 0.3 in
Stall Current	: 37A	Mounting Screws (2)	: M3
Kt	: oz-in/A		
Kv	: rpm/V		
Efficiency	: 70%		
RPM - Peak Eff	: 14840		
Torque - Peak Eff	: 4.47 oz-in/A		
Current - Peak Eff	: 5.85A		

Figure 19: Specifications of RS-540 motor [4]

The speed-torque characteristics of the BaneBots RS-540 motor is given in Figure 20.

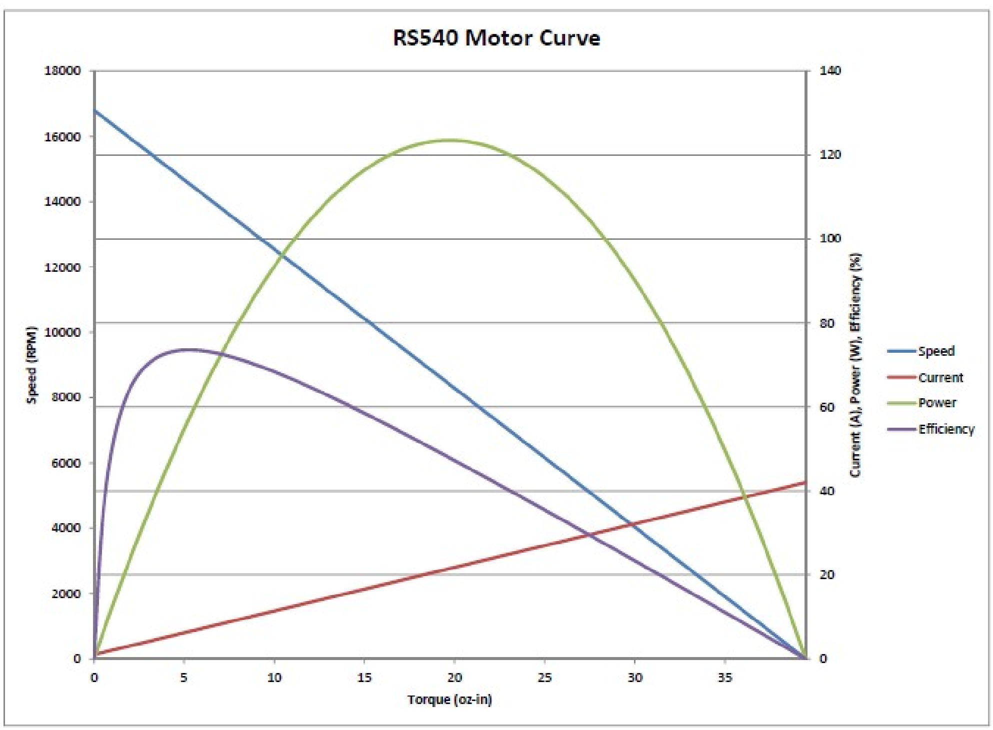


Figure 20: Speed-Torque curve of RS-540 motor [5]

After the motor was finalised, the torque and rpm of the motor was noted. The values of the above parameters were found to be:

Stall torque = **32.64 oz-in**

No load rpm = **17200 RPM**

4.1.3 Gearbox Selection

Since the stall torque of the motor is quite low in comparison to the torque required by the robot, so a gearbox had to be used to increase the torque of the motor. The BaneBots website was referred for a suitable gearbox. For this the gear ratio for the gearbox had to be calculated.

Gear-ratio required	=	Stall torque (required) / Stall torque (generated)
	=	627 / 32.64
	=	**19.21**
Gear-ratio required	=	No load RPM (generated) / No load RPM (required)
	=	17200 / 150
	=	**114.67**

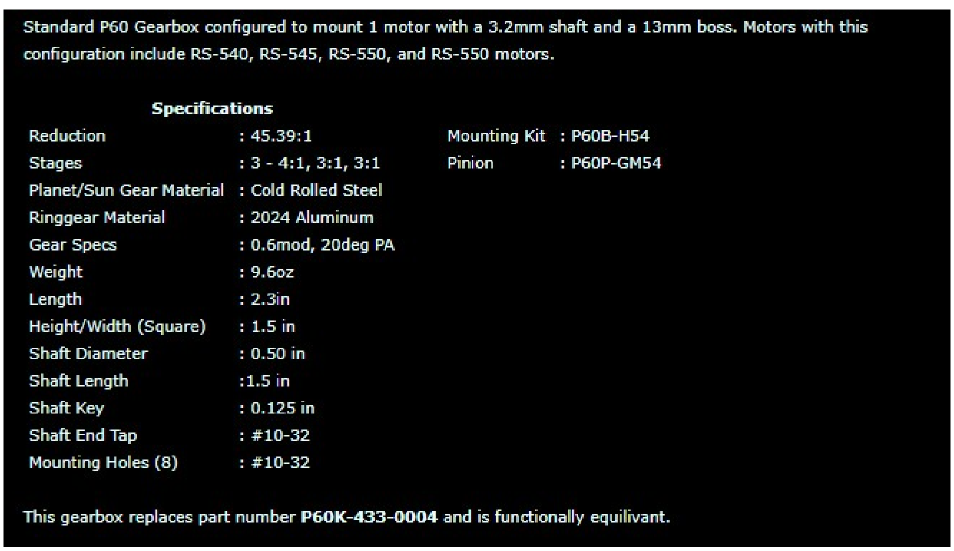
With the two values indicating the acceptable range of values of gear-ratio required for the gearbox, the task was to select a gearbox that has gear-ratio between these two values and is less costly.

The options available for the gearboxes that could fit the above range are given in Figure 21.

Figure 21: Available BaneBots Gearboxes [6]

We decided to go with the 45: 1 ratio gearbox since it provided enough torque and did not reduce the rpm of the motor to a great extent.

The specifications of BaneBots P60 gearbox is given by Figure 22.



Standard P60 Gearbox configured to mount 1 motor with a 3.2mm shaft and a 13mm boss. Motors with this configuration include RS-540, RS-545, RS-550, and RS-550 motors.

Specifications

Reduction	: 45.39:1	Mounting Kit	: P60B-H54
Stages	: 3 - 4:1, 3:1, 3:1	Pinion	: P60P-GM54
Planet/Sun Gear Material	: Cold Rolled Steel		
Ringgear Material	: 2024 Aluminum		
Gear Specs	: 0.6mod, 20deg PA		
Weight	: 9.6oz		
Length	: 2.3in		
Height/Width (Square)	: 1.5 in		
Shaft Diameter	: 0.50 in		
Shaft Length	:1.5 in		
Shaft Key	: 0.125 in		
Shaft End Tap	: #10-32		
Mounting Holes (8)	: #10-32		

This gearbox replaces part number **P60K-433-0004** and is functionally equilivant.

Figure 22: Specifications of Gearbox P60 45:1 [7]

With the gearbox attached, calculations were performed to calculate the maximum torque generated when the gearbox is attached to the motor and the maximum RPM output of the motor.

The values were:

Maximum torque (taking into account the efficiency of the motor)

= 32.64 * 0.7 * 45

= **1028.16 oz-in**

Maximum RPM output (taking into account the efficiency of the motor)

= 17200 * 0.7 / 45

= **267.56RPM**

Figure 23: BaneBots RS-540 motor [8]

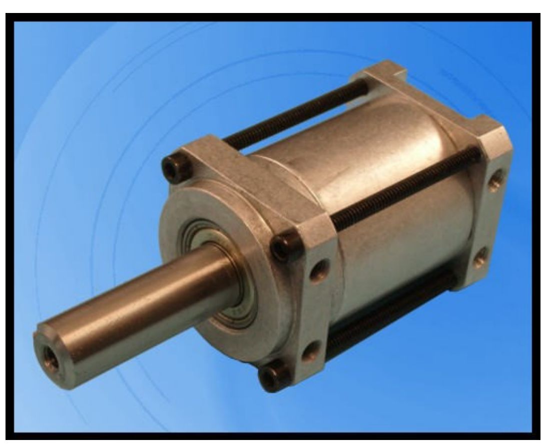


Figure 24: BaneBots P60 45:1 Gearbox [7]

4.2 Motor Driver

The DC motor used in the project has the stall current of 37A and operating voltage as 4.5V - 12V. So, a motor driver that could sustain a current of 37A without getting damaged was to be selected. The robotics club of IIT Delhi had already used motor drivers from Dimension Engineering in their projects. Since they assured us of the reliability of these drivers, we decided to go by the same. So, we had two options from Dimension Engineering:

1. SyRen 50A regenerative motor driver

2. Sabertooth dual 60A motor driver

We decided to go by rthe latter because the former one could only be used for a single DC motor.

The features of Sabertooth dual 60A motor driver are as follows [9]:

- Synchronous regenerative drive
- Ultra-sonic switching frequency
- Thermal and overcurrent protection
- Easy mounting and setup
- Compact size
- Carefree reversing
- Multiple operating modes

The motor driver can be seen inFigure 25. The documentation of the driver can be found at[9].

Figure 25: Sabertooth dual 60A motor driver[9]

4.3 Encoders

As mentioned in 2.2.1.2 the current wheel configuration doesn't ensure straight line motion due to difficulty in running the motors at same speed. However, if some sensor, for example rotary encoders, can measure the rotation of both the motors independently, we can calculate the relative rotation between the two motors instantaneously. Then some control algorithm can be implemented on the instantaneous rotation of the two motors so that the robot moves in the desired path. A rotary encoder makes it possible to measure the motor rotation instantaneously.

An encoder is an electromechanical device that is capable of measuring motion or position. Most of the encoders use optical sensors to provide electrical signals in the form of pulse trains that can be converted into motion or position. The encoder used for RoboMuse 3XT is an **incremental quadrature encoder**. An encoder was selected from a list of available encoders shown inTable 5.

Table 5: Available Encoders

ENCODER	DEGREE OF RESOLUTION	INDEX PIN	PRICE(robokits.co.in)
QUAD. ENCODER 600 PPR / 2400 CPR	0.15˚	NO	₹1500
QUAD. ENCODER 1000 PPR / 4000 CPR	0.09˚	YES	₹2500
QUAD. ENCODER 2500 PPR / 10000 CPR	0.036˚	YES	₹4100

The encoder selected for RoboMuse 3XT is quadrature encoder with PPR value being 1000 and CPR value being 4000. The reason being that this encoder had an index pin as compared to its previous counterpart. The degree of resolution of selected encoder met our requirements of atleast 0.1˚ resolution. Keeping the cost of the encoder in mind, the encoder selected had less cost than the next encoder in the list.

4.4 Screen

The screen is interfaced with the Raspberry Pi and is responsible for accepting input from the user and communicating the response with the Raspberry Pi. A Graphical User Interface is made that would be run on the screen. For this there were two options available. The first option was to use a standard touchscreen and the second option was to use a car monitor.

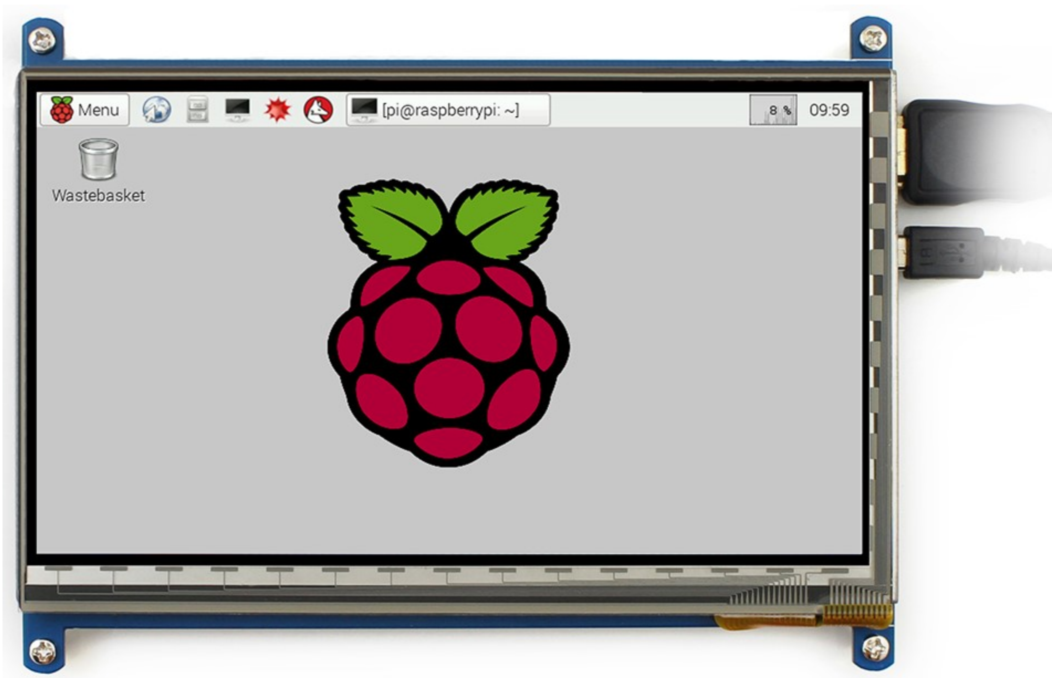


Figure 26: Touchscreen for Raspberry PI [10]

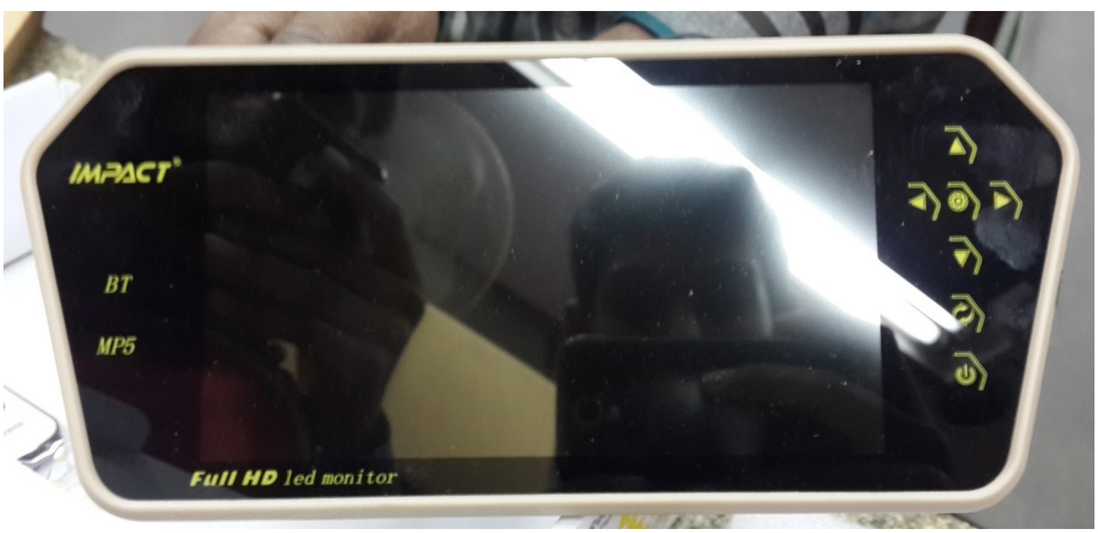


Figure 27: LED monitor

It was decided to go with the touchscreen over the led monitor due to the following reasons:

1. The touchscreen could be easily interfaced to the Raspberry Pi using a HDMI cable and a USB cable. The same was not possible with the LED monitor as it could be connected using only RCA connectors.
2. The touchscreen consumed less power as compared to the LED monitor.
3. The touchscreen provided an extra added functionality of 10 finger touch whereas in case of LED monitor only the buttons were touch.

4.5 Obstacle detection sensor

The robot is supposed to avoid obstacles in its path. The purpose was such that the robot does not collide with the obstacle.

Various techniques to detect obstacles using a minimum number of sensors were compared to pros and cons. The sensor which has to be used on the robot was compared against each other for pros and cons keeping in the mind the constraints of electrical and computational power.

The options available to us were to either use an object detection module like a Kinect sensor or the other option was to use ultrasonic sensors.

4.5.1 Kinect sensor

Kinect is a motion sensing input device by Microsoft for Xbox 360 and Xbox One video game consoles and Windows PCs. The device features an RGB camera, depth sensor and multi-array microphone which provides full-body 3D motion capture, facial recognition and voice recognition capabilities. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions.

Figure 28: Kinect Sensor [11]

This device can also be modified to act as an RGBD sensor, the availability of open source software enables this modification.

But there are some cons with this device:

1. Large power consumption 12Watts.
2. The requirement of complex OS and higher computational power to handle libraries and data stream.
3. The Infrared sensor becomes unreliable in presence of sunlight.
4. Cost is also high.

So, it was decided to switch over to some other sensor. We decided to go with Ultrasonic sensors.

4.5.2 Ultrasonic Sensors

Ultrasonic sensors are based on the measurement of the properties of acoustic waves with frequencies above the human audible range, often at roughly 40 kHz. They typically operate by generating a high-frequency pulse of sound and then receiving and evaluating the properties of the echo pulse.

In reflection mode (also known as “echo ranging”), an ultrasonic transmitter emits a short burst of sound in a particular direction. The pulse bounces off a target and returns to the receiver after a time interval t. The receiver records the length of this time interval, and calculates the distance travelled d based on the speed of sound c:

$$d = c*t$$

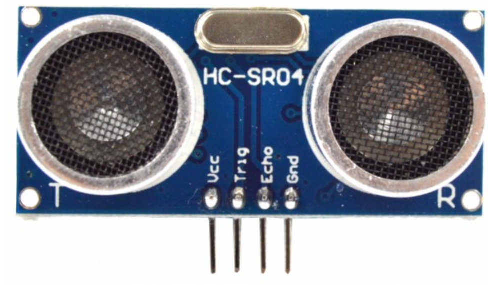


Figure 29: Ultrasonic sensor [12]

4.5.2.1 Demerits:

1. **Angle**. For the transmitted wave to echo back to the receiver, the target surface must be perpendicular to the transmitter. Round objects are therefore most easily sensed since they always show some perpendicular face. When targeting a flat object, care must be taken to ensure that its angle with respect to the sensor does not exceed a particular range.

2. **Dead-zone**. Ultrasonic sensors typically have a "dead zone" immediately in front of them in which objects cannot be detected because they deflect the wave back before the receiver is operational.

3. **Material**. Some materials are more absorbent than others, and these will reflect less ultrasound. This complicates using the attenuation method to measure the distance of arbitrary objects.

4.5.2.2 Merits

1. **Fast detection**. The time required for detection of an obstacle is very less.

2. **Cheap**. Ultrasonic sensors are cheaply available in the market so the overall cost can be reduced

3. **Modularity**. Even if one sensor is not working as a whole the module still functions thus giving a higher reliability

In order to overcome the demerits, we had to modify the way in which we use them. The problem of covering a large area with a narrow cone of ultrasonic was not possible, so multiple ultrasonic sensors were mounted on the front side of the RoboMuse 3XT.

When used simultaneously, the echo from one ultrasonic sensor interfered with other ultrasonic sensors and gave false values. This problem was solved by synchronizing the operation of ultrasonic sensors such that no two sensors work together. A loop time of 130 milliseconds was achieved after using a dedicated Arduino Uno for the detection module.

The Arduino Uno acts as a detection module with four sensors. One of the digital pins in the Arduino Uno acts as a command pin. When any obstacle is encountered within a range of 80 cm the command pin goes to 'High' state (on) else the pin goes to 'Low' state (off) after one loop.

5 PCB Design

The project had three major circuitries. These were the power distribution board, Arduino Uno board and Arduino mega board. To make the system more reliable and ensure rigid connections, we decided to develop printed circuit boards for each of the major circuitries. First of all, it was decided to make a single PCB that could supplement all of the above boards. But then if there is problem in some part of PCB, the entire PCB would not work. So, we decided to make three separate PCB each for the three boards. There were 2 options: either to design PCB in Eagle CAD software or in DIPTRACE software. As the members of robotics club were designing their PCBs in DIPTRACE software, so we decided to work in DIPTRACE in order to maintain uniformity.

5.1 Power PCB

The PowerPCB is developed for power distribution throughout the electrical systems. it consists of capacitors to give a steady output of 12V from the fluctuating dc supply. There are headers for 12V as well as 5V. 12v to 5v conversion is done using a buck convertor. The PCB also has a fuse so that when the current exceeds 40A, the current does not to the electronic components, otherwise they might burn out.

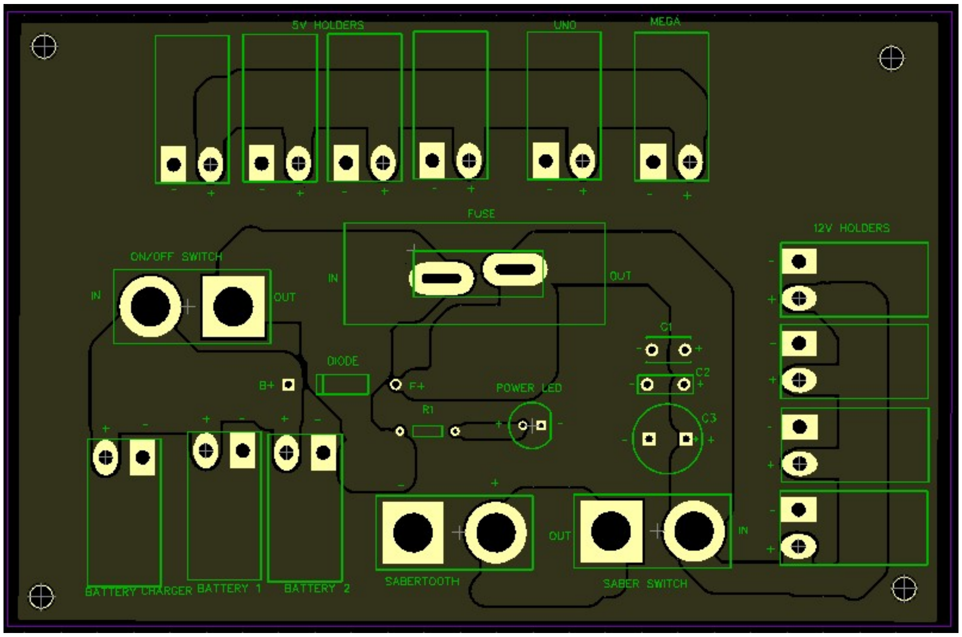


Figure 30: Power PCB - Top Layer

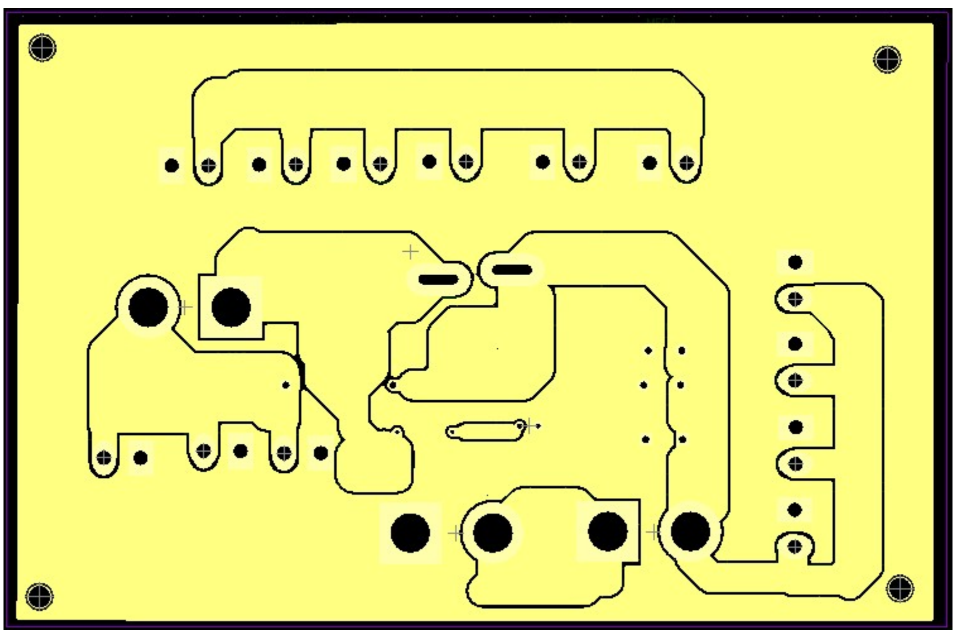


Figure 31: Power PCB - Bottom Layer

Features:

- Length of the PCB: **130mm**
- Width of the PCB: **85mm**
- Thickness of copper pour: **0.07 mm**

- Single-sided PCB
- It has **six** 5V outputs and **four** 12V outputs
- A **fuse** to prevent excessive current flow
- **Diode** to allow backward flow of current
- Capacitors act as **de-coupling capacitors**

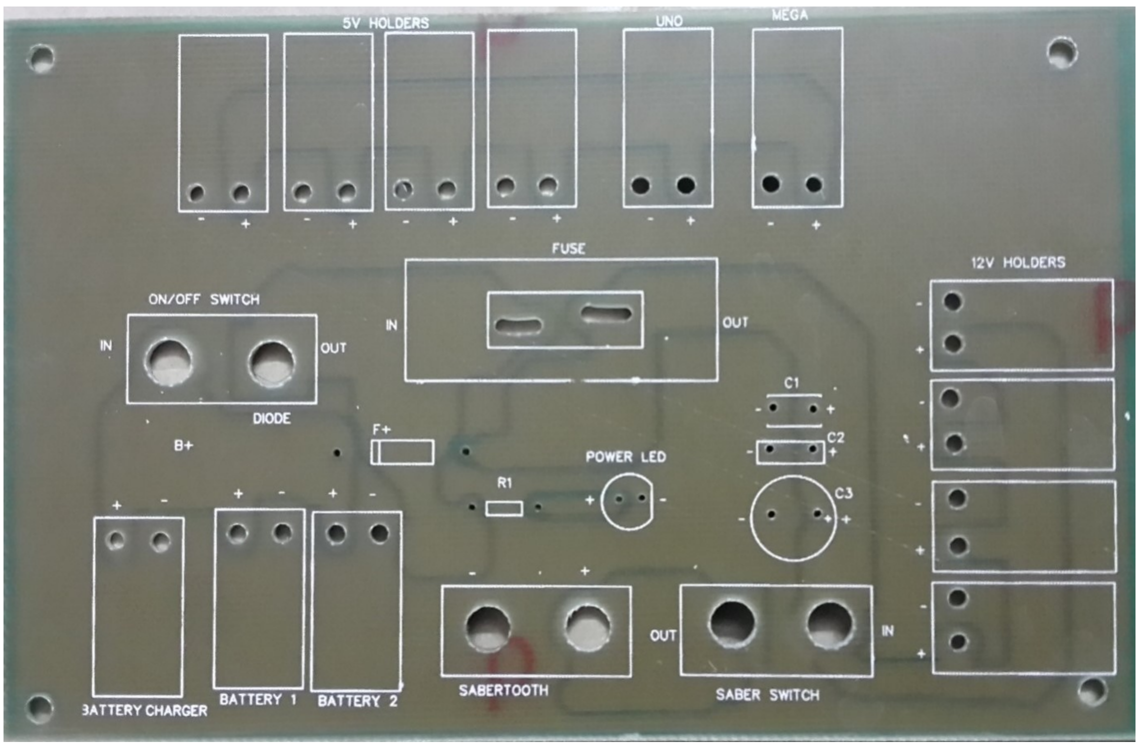


Figure 32: Power PCB after fabrication - Top Layer

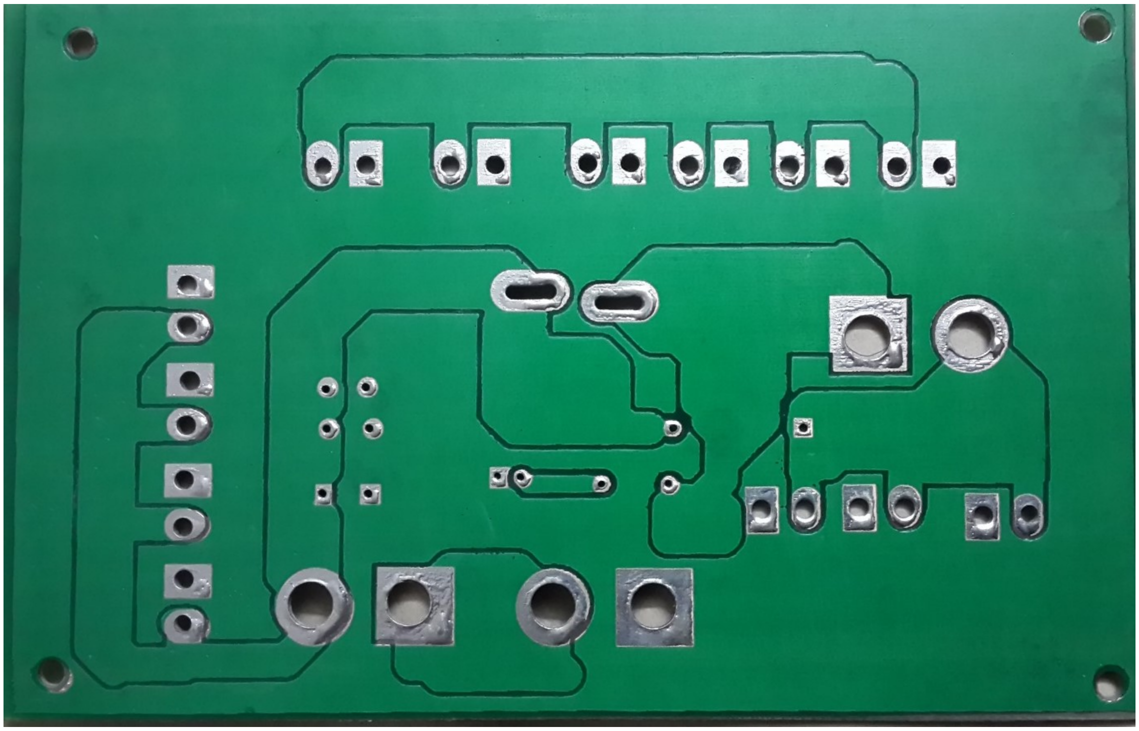


Figure 33: Power PCB after fabrication - Bottom Layer

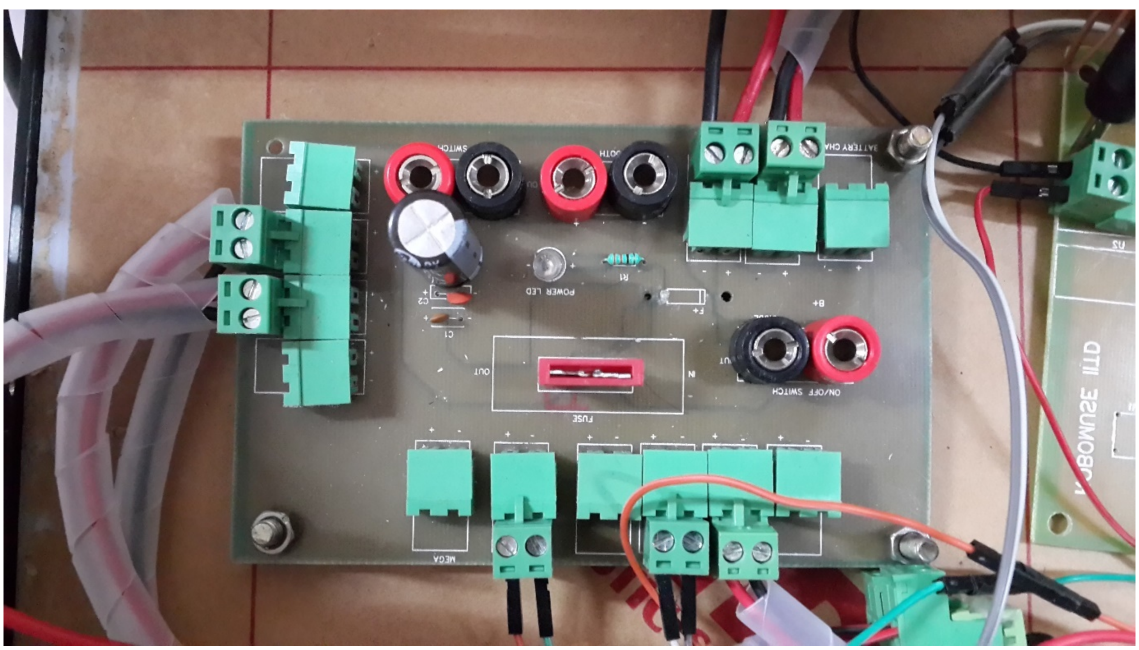


Figure 34: Power PCB mounted on RoboMuse

5.2 Mega PCB

The Arduino Mega PCB is fabricated to mount Arduino mega on it and add support for the various other components and sensors on the board. The PCB has connectors to mount Arduino mega rigidly on the board and it also supports the use of Bluetooth module. The module is designed specifically keeping in mind that it can be used if others sensors have to be integrated in the future.

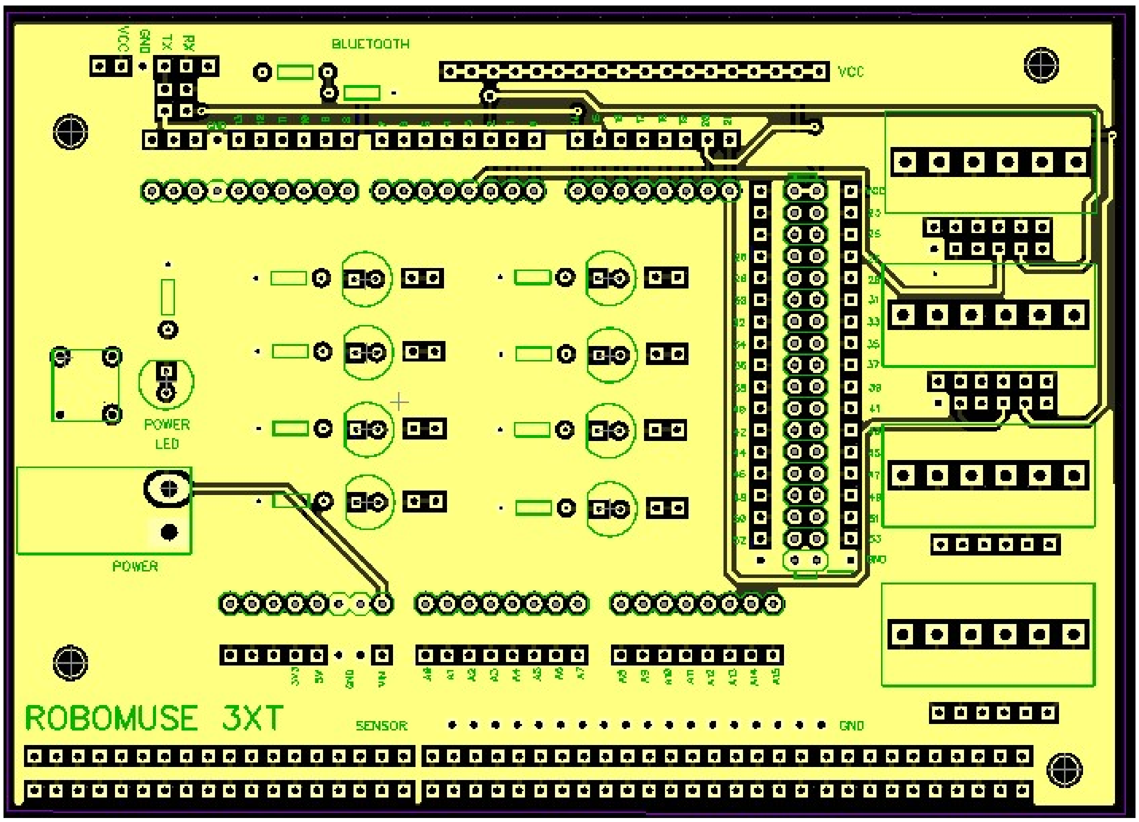


Figure 35: Arduino Mega PCB - Top Layer

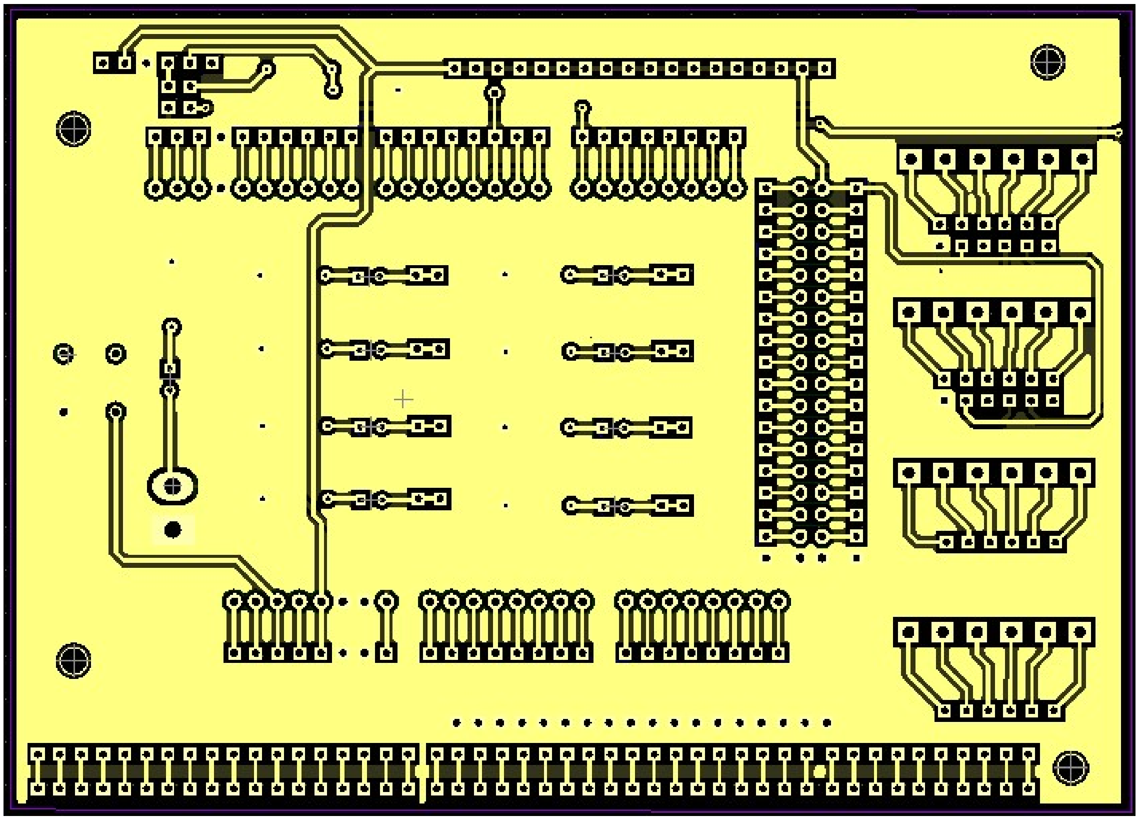


Figure 36: Arduino Mega PCB - Bottom Layer

Features:

- Length of the PCB: **131mm**
- Width of the PCB: **94mm**
- Thickness of copper pour: **0.035 mm**
- Double-sided PCB
- Support for **Bluetooth module**
- Support for extra **Encoders**
- **Debugging leds** to debug Arduino pins
- Support for extra **sensors**

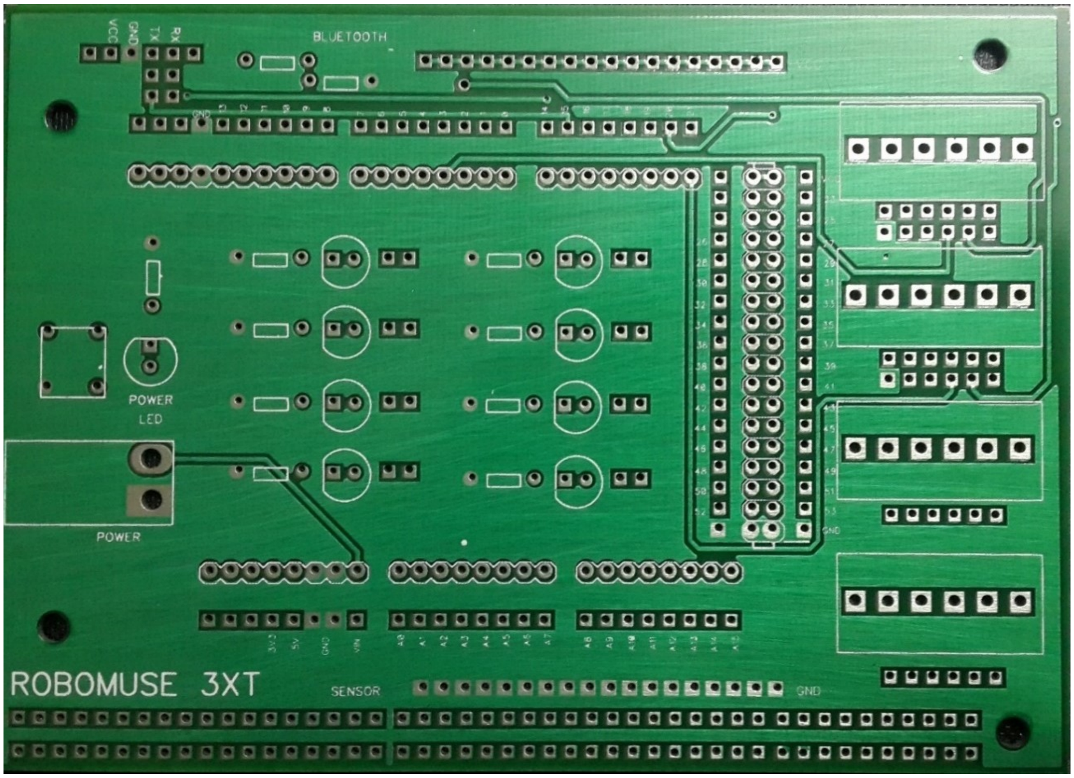


Figure 37: Arduino Mega PCB after fabrication - Top Layer

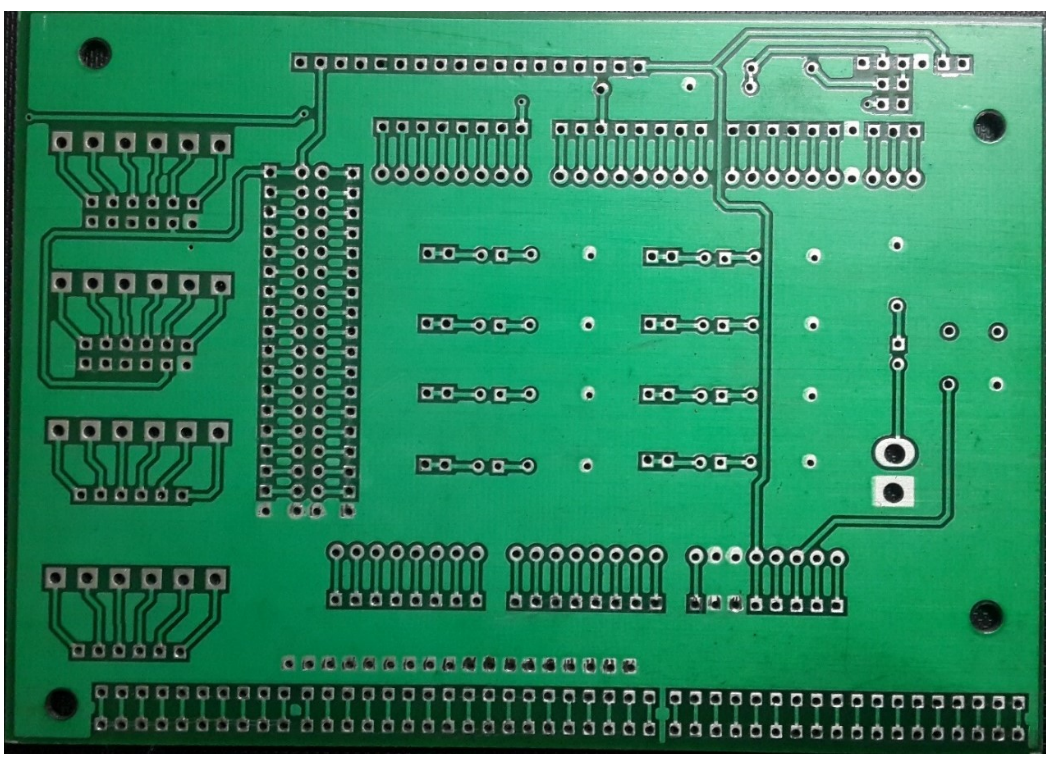


Figure 38: Arduino Mega PCB after fabrication - Bottom Layer

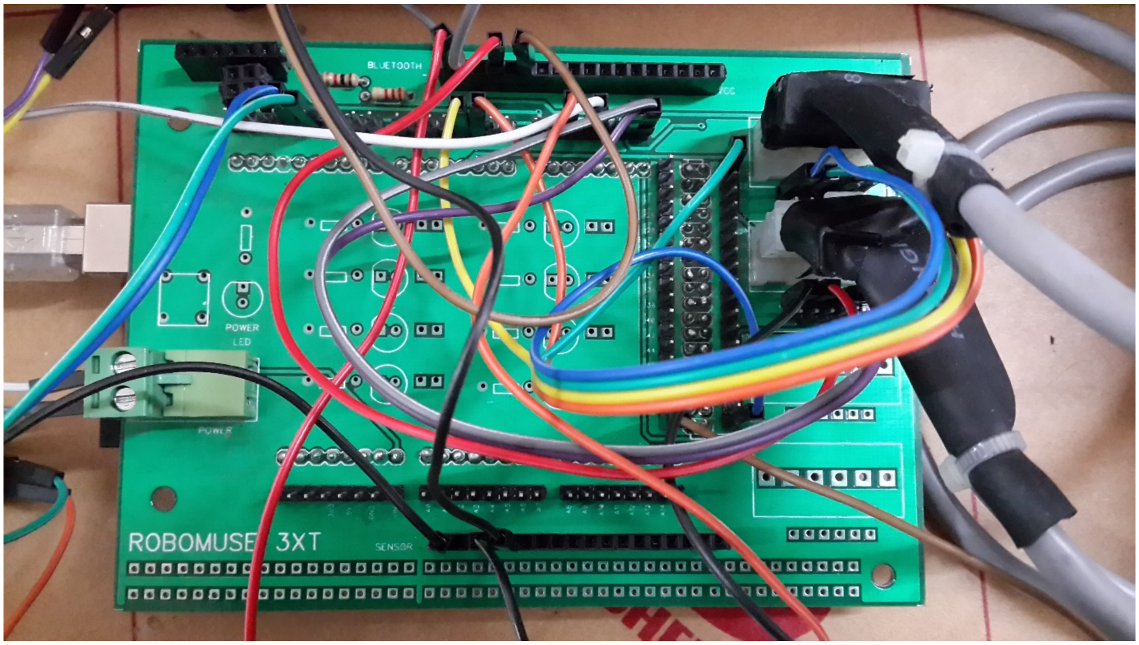


Figure 39: Arduino Mega PCB mounted on RoboMuse

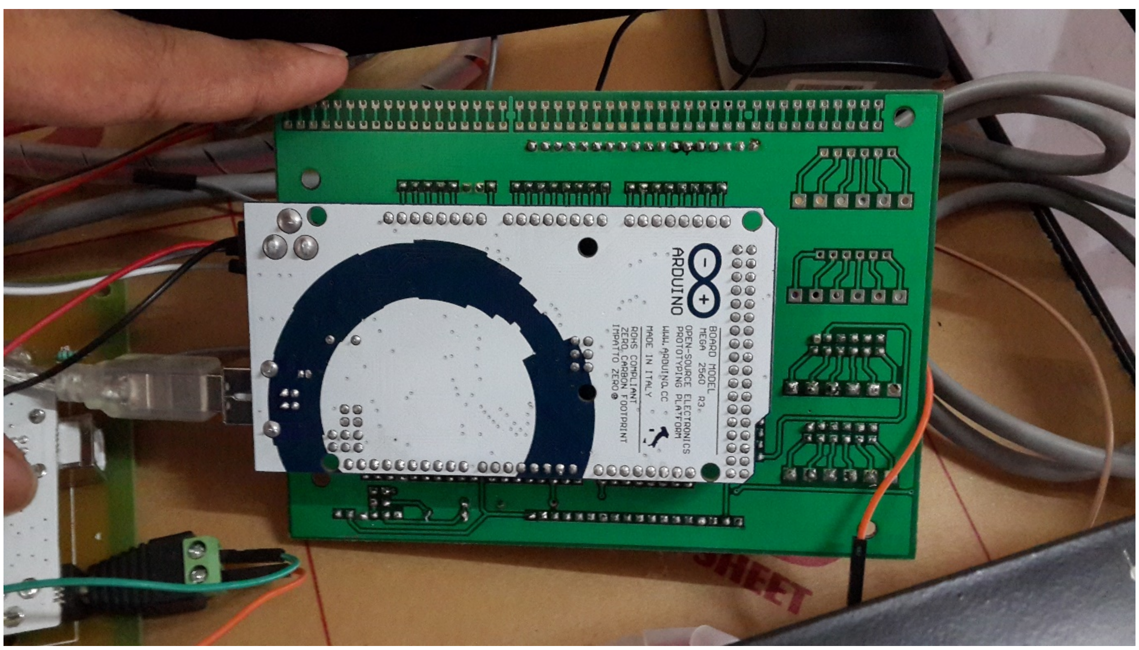


Figure 40: Arduino Mega attached to PCB

5.3 Uno PCB

The Arduino Uno PCB is designed to attach different ultrasonic sensors with the Arduino uno board.

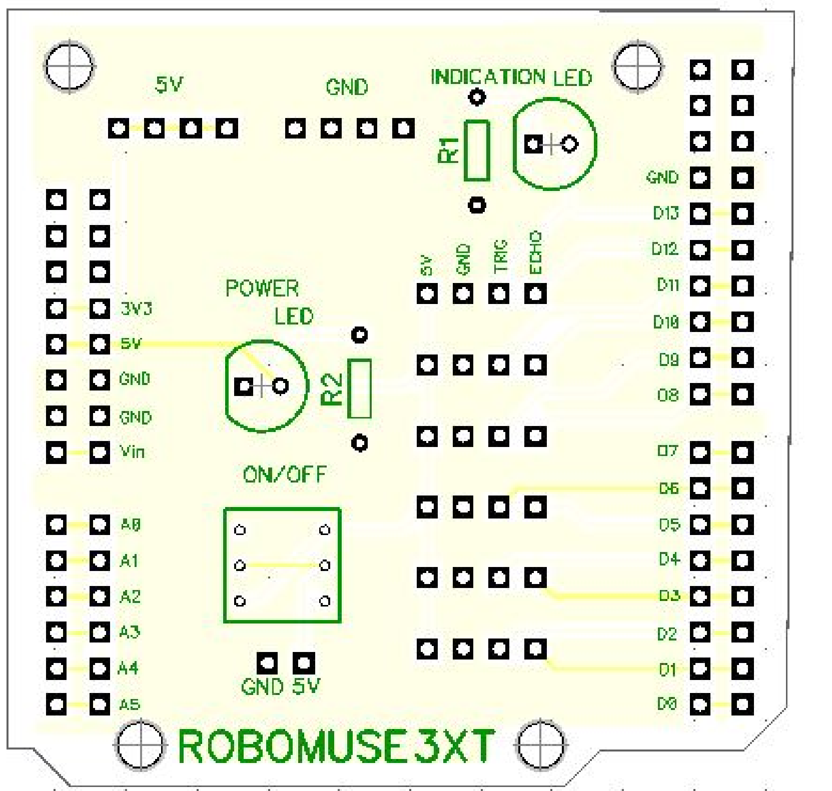


Figure 41: Arduino Uno PCB

Features:

- Length of the PCB: **70mm**
- Width of the PCB: **54mm**
- Thickness of copper pour: **0.035 mm**
- Single-sided PCB
- Support for upto 6 Ultrasonic sensors

6 Control System

Our robot operates on 3 controlling boards, namely, Arduino Uno, Arduino Mega and Raspberry Pi.

6.1 Arduino Uno

- Arduino Uno has been used for Obstacle detection.
- RoboMuse 3XT has four Ultrasonic sensors in the front that are used to detect an obstacle.
- Whenever an obstacle is detected, it interrupts Arduino Mega so that motors could be stopped.

Specifications of the Arduino UNO[13]**:**

Table 6: Specifications of Arduino Uno

S.No.	Specification	
1	**Microcontroller**	ATmega328
2	**Operating Voltage**	5V
3	**Input Voltage (recommended)**	7-12V
4	**Input Voltage (limits)**	6-20V
5	**Digital I/O pins**	14 (of which 6 provide PWM output)
6	**Analog Input pins**	6
7	**DC current per I/O pin**	40mA
8	**DC current for 3.3V pin**	50mA
9	**Flash memory**	32 KB of which 0.5 KB used by bootloader
10	**SRAM**	2 KB (ATmega328)
11	**EEPROM**	1 KB (ATmega328)
12	**Clock Speed**	16 MHz

6.2 Arduino Mega

Arduino Mega controls the motor driver, thereby controlling the motion of the robot. Apart from controlling the motors, Arduino mega also performs the following functions:

- It is responsible for reading serial data from raspberry pi which tells it how to traverse in the environment.
- It implements PID control over the motion of the robot to eliminate any errors caused due to slipping of robot or due to inertia of the robot.
- It performs odomertic calculations from the data fed to it via the encoders so as to localise itself with respect to an initial reference frame.

Specifications of the Arduino MEGA[14]**:**

Table 7: Specifications of Arduino Mega

S.No.	Specification	
1	**Microcontroller**	ATmega1280
2	**Operating Voltage**	5V
3	**Input Voltage (recommended)**	7-12V
4	**Input Voltage (limits)**	6-20V
5	**Digital I/O pins**	54 (of which 15 provide PWM output)
6	**Analog Input pins**	16
7	**DC current per I/O pin**	40mA
8	**DC current for 3.3V pin**	50mA
9	**Flash memory**	128 KB of which 4 KB used by bootloader
10	**SRAM**	8 KB (ATmega1280)
11	**EEPROM**	4 KB (ATmega1280)
12	**Clock Speed**	16 MHz

6.3 Raspberry PI

- A GUI is developed in Python to interact with the user. The GUI will read data from the user and then fed it serially to Arduino mega so that it can pass some value to the motor driver making the robot to perform an action.
- The GUI consists of all the modes that the robot can execute.
- It executes Face Detection and Face Recognition algorithms to recognize the person in its vicinity and thereby indicate the presence of an intruder in a fixed environment.

Specifications of Raspberry Pi [15]:

Table 8: Specifications of Raspberry Pi

S.No.	Specification	
1	**Processor**	Broadcom BCM2837 64bit ARMv7 Quad Core Processor powered Single Board Computer running at 1.2GHz
2	**RAM**	1GB
3	**WIFI**	Yes (BCM43143)
4	**Bluetooth**	Yes
5	**GPIO pins**	40
6	**USB 2.0 ports**	4
7	**HDMI Output**	Yes
8	**Audio Output**	Yes
9	**Camera Port**	Yes
10	**Power Source**	Micro USB power source (supports up to 2.4 Amps)

7 Control Theory

7.1 PID

- Most commonly used control algorithm.
- Robust performance in a wide range of operating conditions.
- Three coefficients, proportional, integral and derivative, varied to get optimal response.
- PID algorithm constantly calculates error value and updates the output based on Kp, Ki and Kd terms.
- $u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + \frac{K_d de(t)}{dt}$

7.1.1 Block Diagram

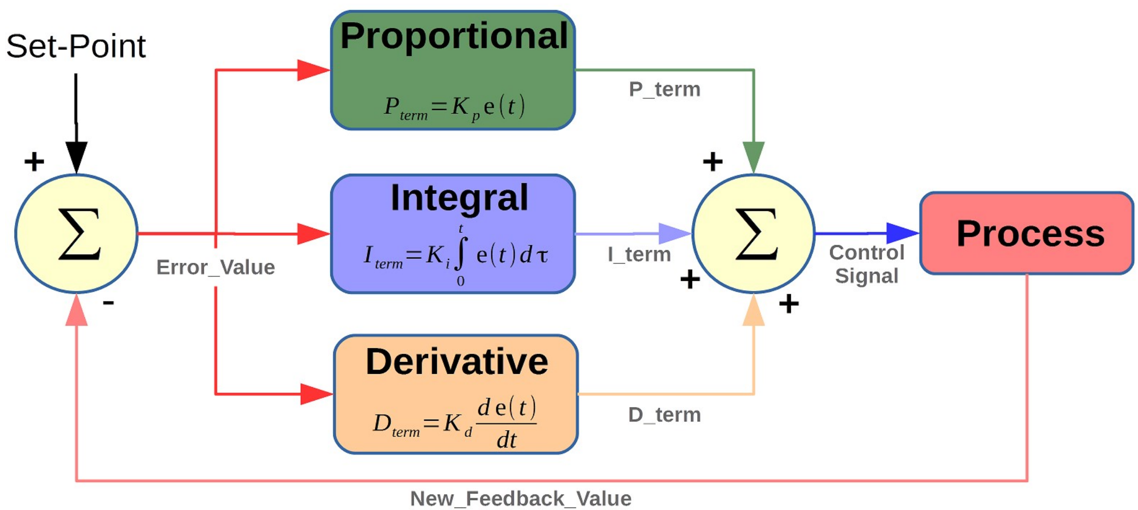


Figure 42: Block diagram for PID controller [16]

7.1.2 Types of Responses

PID controller consists of three kinds of responses. These are:

- Proportional Response
- Derivative Response
- Integral Response

7.1.2.1 Proportional Response

- Proportional component determines the output that is proportional to the error term.
- Proportional control acts like a spring (or capacitive) response.
- Increasing Kp, increases the speed of response.
- With higher values of Kp, the oscillatory motion of the system increases.
- Proportional term: $\boldsymbol{P_{OUT} = K_P e(t)}$

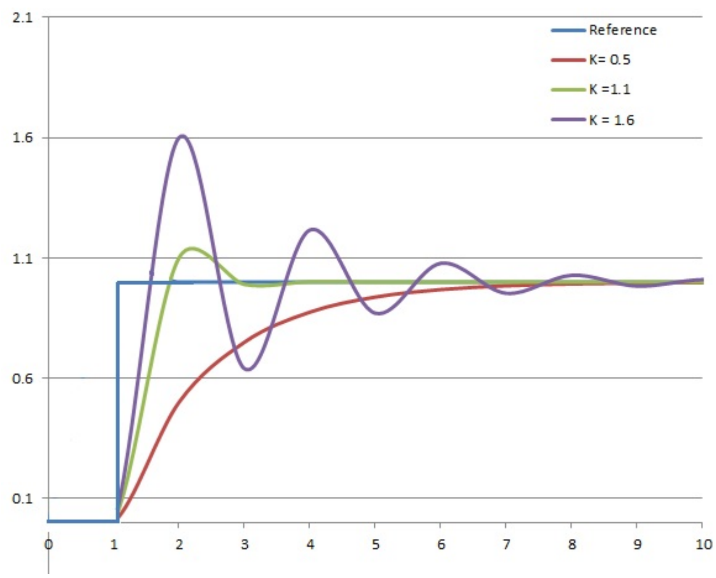


Figure 43: Plot of PV vs time; for three values of Kp(Ki and Kd being held constant) [17]

7.1.2.2 Derivative Response

- Derivative response is proportional to rate of change of error of system.
- Increasing derivative time parameter causes system to react more strongly to changes in error term.
- Derivative Response highly susceptible to noise in the process variable.
- Derivative term has effect on transient response of system.
- Derivative term: $D_{OUT} = K_d \frac{de(t)}{dt}$

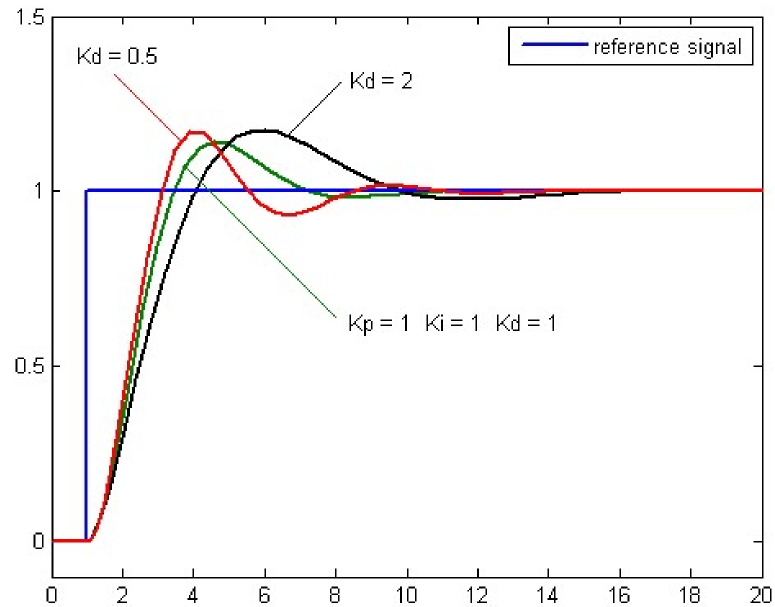


Figure 44: Plot of PV vs time; for three values of Kd (Kp and Ki being held constant) [17]

7.1.2.3 Integral Response

- Integral component accumulates the error over the course of action.
- It is sum of instantaneous errors that gives accumulated offset that must be corrected.
- Integral windup - results when integral action saturates a controller without the controller driving the error signal toward zero.
- It affects the steady state response of the system.
- Integral term: $$I_{OUT} = K_i \int_0^t e(\tau) d\tau$$

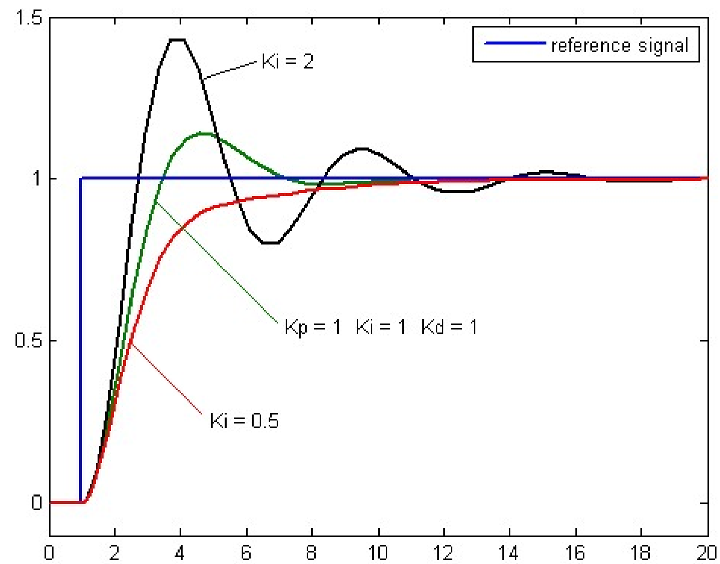


Figure 45: Plot of PV vs time; for three values of Ki (Kp and Kd being held constant) [17]

7.1.3 TUNING PID PARAMETERS

- First of all, set all the parameters, namely, K_p, K_i and K_d to zero.
- **To set K_p :**

  Increase the K_p value until the output of the system oscillates. Increasing the $K_p$value, increases the speed of response of the system. But the $K_p$value must not be too large, otherwise the system becomes over-damped. The value of K_p is noted when the oscillations begins in the system.

- **To set K_i :**

  After setting the K_p value, K_i value is increased so that oscillations in the system could be eliminated. The integral term reduces the steady state error, but it increases the overshoot of the system. So, the value of K_i must be selected such as the oscillations are eliminated and the overshoot is very low.

- **To set K_d :**

After the K_p and the K_i values have been set to desired values, the K_d values are increased. Increasing the k_d term, decreases the overshoot of the system and provides higher stability to the system. But care must be taken as increasing the $K_d$value makes the system susceptible to noises. Thus, K_d value must be selected as low as possible.

7.2 Localisation

In mobile robotics, two basic positioning-estimation positioning methods are used:

1. Absolute Positioning Methods
2. Relative Positioning Methods

Absolute positioning methods usually rely on navigation beacons, active or passive landmarks, map matching, or satellite-based navigation signals whereas relative positioning is usually based on odometry (i.e., monitoring the wheel revolutions to compute the offset from a known startingposition). Another approach to the position determination of mobile robots is based on inertial navigation with gyros and/or accelerometers.

Odometry is the most widely used method for determining the momentary position of a mobilerobot and also the method used in RoboMuse 3XT due to the following reasons:

1. Navigation beacons, landmarks and map-matching require that the work environment is known.
2. Navigation beacons and landmarks usually require costly installations and maintenance.
3. Map-matching methods are slower than odometry.
4. Satellite- based navigation has poor accuracy (on the order of 10-30 meters) when the robot is working indoors.
5. Data from accelerometers and gyros must be integrated to yield position which makes theses sensors exceedingly sensitive to drift.

This has been discussed in[18].

7.2.1 Odometry

As mentioned in 4.3 , rotary encoders installed on the robot can measure the rotation of the two motors. This can be used to measure the relative position of the robot with respect to some initial frame as discussed below [18]:

Suppose that at sampling interval t, the left and right wheel encoders show the encoder count as N_L and N_R respectively.

We compute a conversion factor that translates the encoder counts into linear wheel displacement.

$$C_m = \pi . D_n / n . C_e$$

where,

C_m = conversion factor that translates encoder pulses into linear wheel displacement

D_n = nominal wheel diameter (in mm)

n = gear ratio of the reduction gear between the motor(where encoder is attached) and wheel

C_e = encoder resolution (in pulses per revolution)

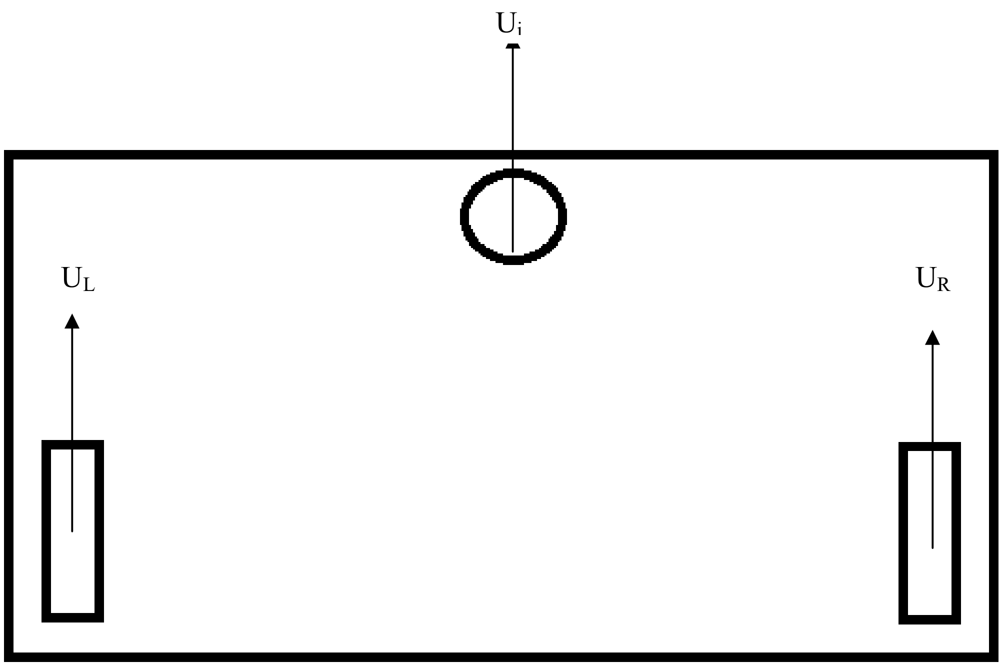


Figure 46: Variables for Odometric Calculations

The incremental travel distance for the left and the right wheels is as follows,

$$\Delta U_L = C_m . N_L$$

$$\Delta U_R = C_m .\ N_R$$

The incremental linear displacement of the robot's centre point is given by:

$$\Delta U_i = (\Delta U_L + \Delta U_R) / 2$$

Change in orientation of the robot's centre-point is given as:

$$\Delta \theta_i = (\Delta U_R - \Delta U_L) / 2$$

Where, b = wheel base (distance between the two drive wheels)

The relative orientation of the robot is given by:

$$\theta_i = \theta_{i-1} + \Delta \theta_i$$

The relative position of the robot's centre-point is given by:

$$x_i = x_{i-1} + \Delta U_i .\ cos(\theta_{i-1} + \Delta \theta_i / 2)$$

$$y_i = y_{i-1} + \Delta U_i .\ sin(\theta_{i-1} + \Delta \theta_i / 2)$$

8 Software

8.1 GUI

The GUI is built to interact with the user. The user could use it to get desired motion from the robot. The GUI can be operated via laptop or phone.

- Coding Platform: **Python**
- Modules Used: **PyQt, Google Text-to-Speech**

PyQt is a set of Python v2 and v3 bindings for The Qt Company's Qt application framework and runs on all platforms supported by Qt including Windows, OS X, Linux, iOS and Android. PyQt5 supports Qt v5. PyQt4 supports Qt v4 and will build against Qt v5. The bindings are implemented as a set of Python modules and contain over 1,000 classes.

PyQt brings together the Qt C++ cross-platform application framework and the cross-platform interpreted language Python.

Qt is more than a GUI toolkit. It includes abstractions of network sockets, threads, Unicode, regular expressions, SQL databases, SVG, OpenGL, XML, a fully functional web browser, a help system, a multimedia framework, as well as a rich collection of GUI widgets.

The GUI is made using PyQt module in Python programming language. The PyQt modules has various objects that could be integrated in the code and linked together to form a Graphical User Interface.

The GUI along with the face detection code is invoked at the start. The GUI interface looks as shown in the Figure 47.

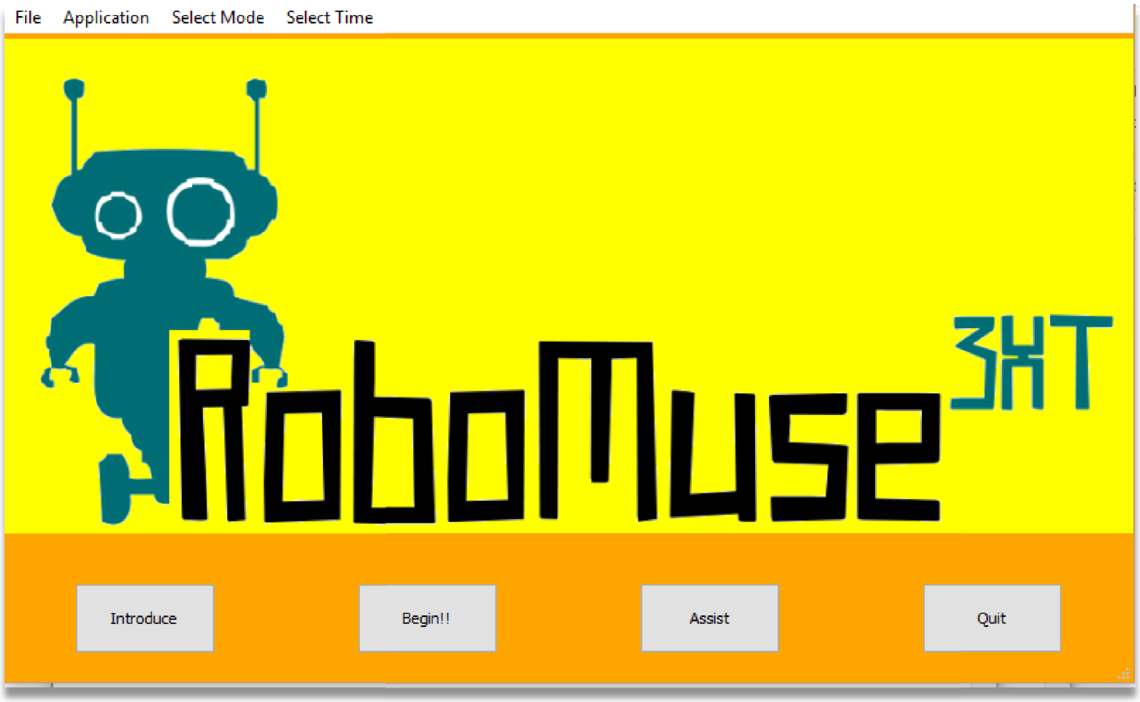


Figure 47: The GUI Interface

The GUI has the following options:

- Clicking on the **Introduce** button, the robot will introduce itself.
- Clicking on the **Begin** button will launch the Face Detection GUI.
- Clicking on the **Assist** button will launch another GUI wherein appropriate destination can be selected and the robot will give the directions.
- Clicking on the **Application** Menu, the robot will move in some pre-defined mode.
- Clicking on the **Select Mode** Menu will present three options:
    - Manual Mode
    - Autonomous Mode
    - Navigation Mode
- In the **Manual Mode**, a controller window appears and the robot could be controlled from a laptop or a phone. The obstacle avoidance code runs simultaneously and while an obstacle is detected, the robot stops executing its operation.

- In the **Autonomous Mode**, the robot will traverse in a square path of fixed length and for a fixed number of turns. Whenever an obstacle is detected, it halts and waits until the obstacle has been removed.
- In the **Navigation Mode**, a GUI window pops up where the user can type the coordinate to which the robot must travel.
- Clicking on the **Select Time** Menu, will present two options – either the **Day Mode** or the **Night Mode**. In the Day Mode, the lights are disabled and in the Night Mode, the lights would be enabled.

8.2 Face Detection

The Face Detection is a module that would identify a human face in an environment. This module is used for one objective of discretising the human being from a non-human obstacle.

- Coding Platform:**OpenCV** in **Python**
- Module Used:**Haar Cascade Classifier**
- What is **Haar Cascade Classifier**?
    - It is a database of 6000 features that are used for facial feature detection.
    - It groups the feature into different stages of classifier and applies them one by one. If the image fails any stage, it is discarded.
    - The image that passes all the stages has the face region.

8.2.1 Haar-like features recognition

This classifier is based on a publication by Papageorgiou et al [19]. which discussed working with a substitute feature set based on Haar wavelets instead of the usual image intensities. Viola and Jones [20] improved the idea of using Haar wavelets and developed the so-called Haar-like features. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. For example, let us say we have an image database with human faces. It is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore, a common haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

Demerits:

- The requirement of large number or data set to teach the classifier.
- Computational costs for training the classifier is huge.

Merits:

- Calculation speed is better compared to other classifiers.
- Availability of open source trained classifiers.

8.2.2 Block Diagram

The Figure 48shows the process that is employed to find whether there is a face in an image or not.

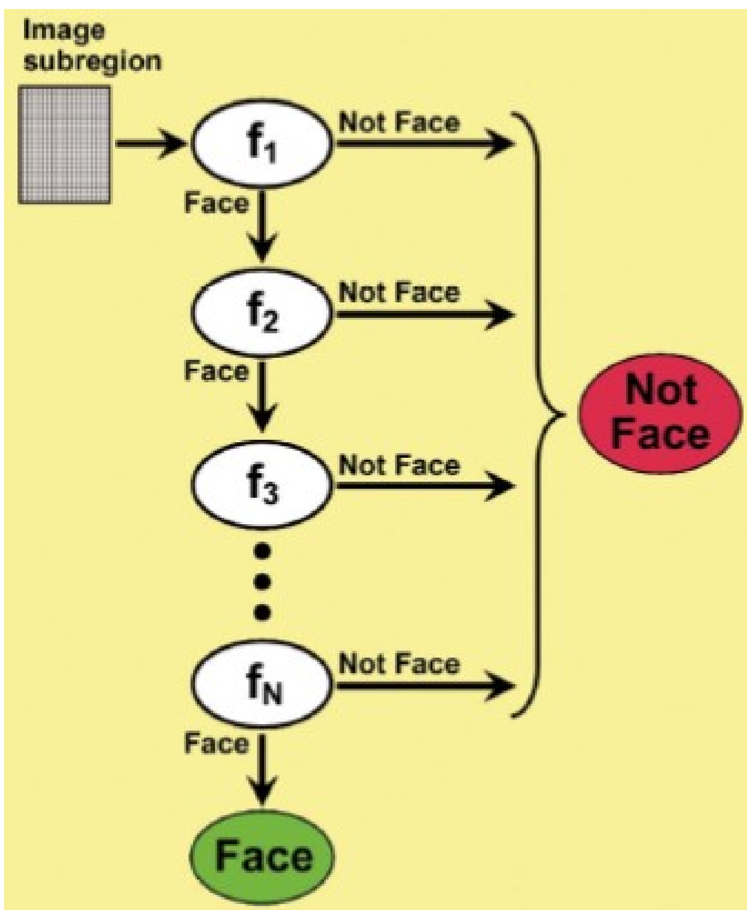


Figure 48: Block Diagram of Haar Cascade Classifier [21]

8.2.3 Demonstration

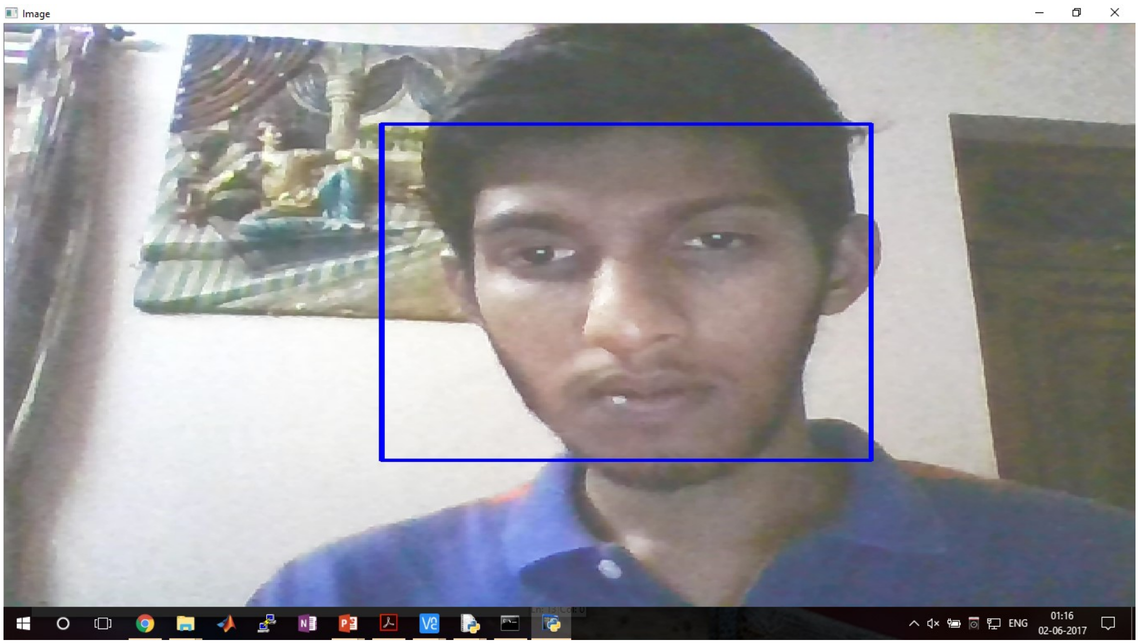


Figure 49: Face Detection

8.3 Face Recognition

The face recognition is a module that can be used to recognise a human face. If the face does matches to a known person, the user of the robot is made aware of the presence of an intruder. The code is developed to help the user keep the track of an area, thus keeping surveillance of the region.

- Coding Platform: **OpenCV** in **Python**
- Module Used: **LOCAL BINARY PATTERN HISTOGRAM**
- What is **Local Binary Pattern Histogram**?

  It is an important feature descriptor that is used in computer vision for template matching. It compares the histogram of the known image to the detected image and tells if there is a match or not.

8.3.1 LBP Descriptor

- In this technique, for each pixel in a grayscale image, a neighbourhood is selected and a LBP value is computed for that pixel.

- **Rule for calculating LBP value:**

  If the current pixel value is greater or equal to the neighbouring pixel value, the corresponding bit in the binary array is set to 1 else if the current pixel value is less than the neighbouring pixel value, the corresponding bit in the binary array is set to 0.

8.3.2 LBP Value Calculation

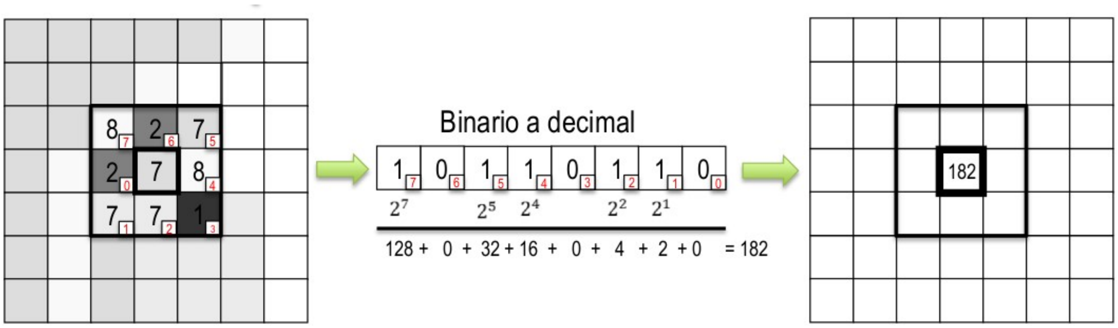


Figure 50: LBP value calculation [22]

- Once the LBP value is computed for all the pixels, a LBP histogram is computed using these values.
- The LBP value range from 0 – 255
- Thus, the size of LBP Descriptor becomes 1 x 256
- The histogram computed is then matched with the histograms of the trained images. If a match is found, the face is recognized.

8.3.3 Flowchart

Figure 51: Face recognition Algorithm

8.3.4 Demonstration

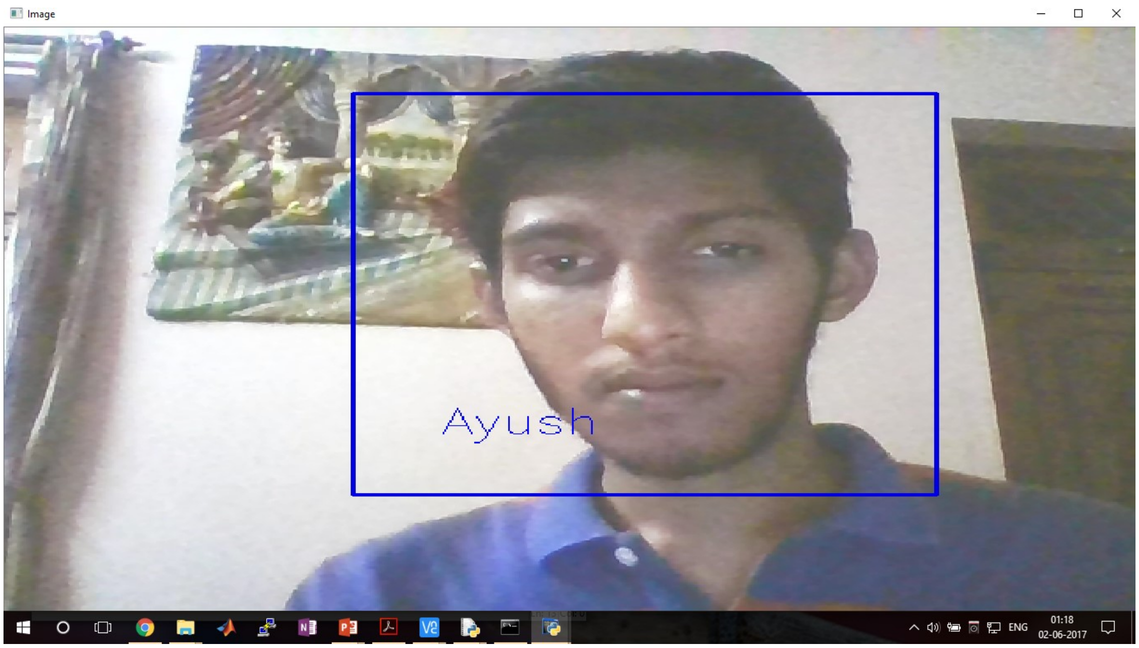


Figure52: Face Recognition Demonstration

8.4 Coding Flowchart

8.4.1 Obstacle Detection

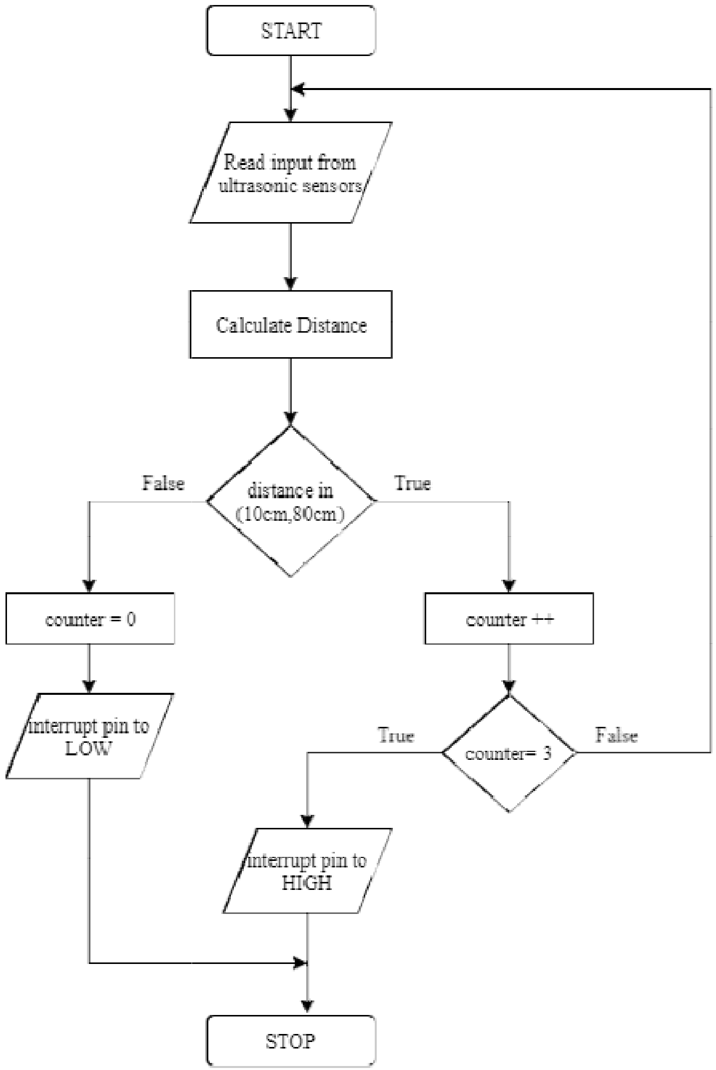


Figure 53: Flowchart for Obstacle Detection

8.4.1.1 Algorithm

1. The ultrasonic sensors are triggered and they transmit a sonic burst.
2. Arduino Uno measures the time difference between the transmission of wave and the reception of the wave. With this it calculates the distance the wave has traversed.
3. If the wave has traversed a distance less than 80 cm, then there is an obstacle in its path.

4. Arduino Uno measures these multiple times. If the obstacle is detected for three consecutive times, it commands the Arduino Mega interrupt pin HIGH.
5. On receiving this interrupt, Arduino Mega stops the motor.

8.4.2 Semi-Autonomous Control

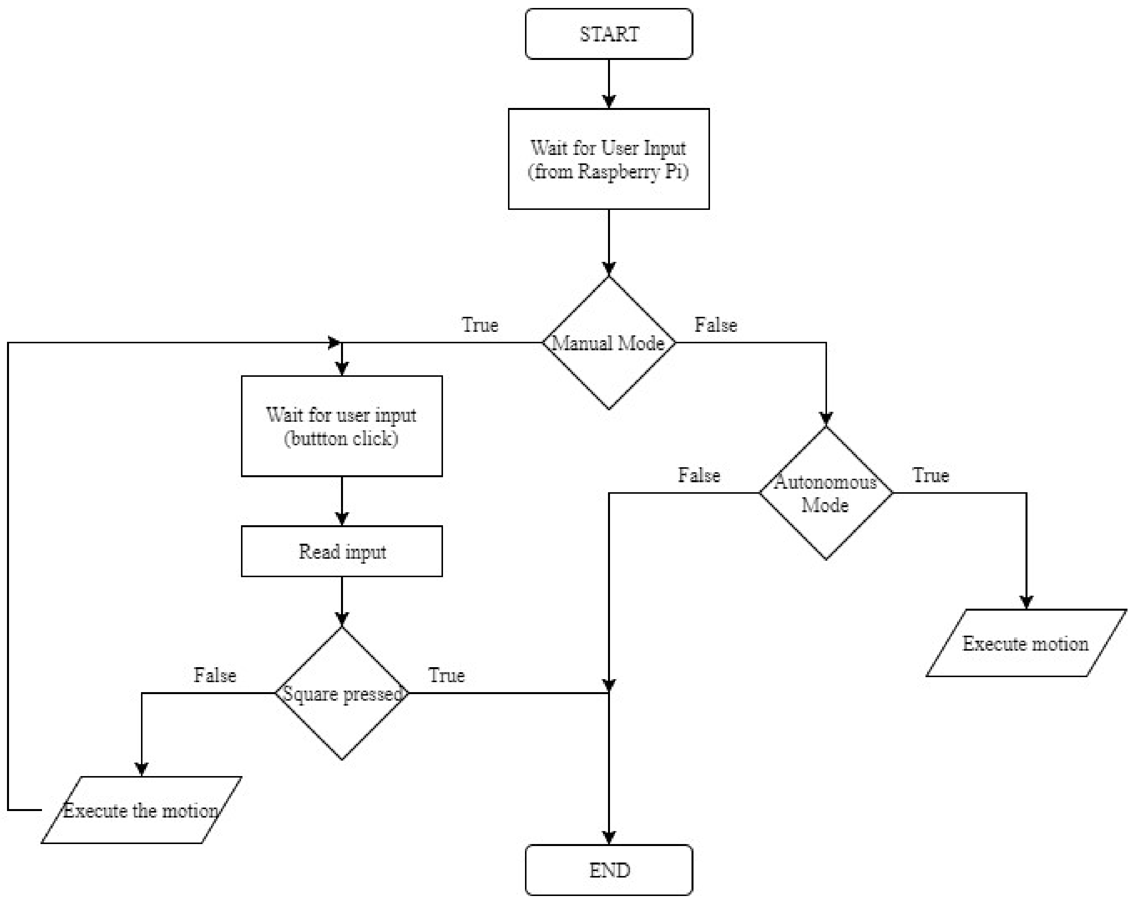


Figure 54: Flowchart for Semi- Autonomous Control

8.4.2.1 Algorithm

1. The Arduino Mega is configured to receive input from Raspberry Pi.
2. If the user presses 'Manual Control' button on GUI, Arduino Mega receives the input and executes motion as the user commands.
3. If the user presses 'Autonomous Control' button on GUI, Arduino Mega executes the motion in a square path and meanwhile detects the obstacle.
4. If the user presses 'Navigation Control' button on GUI, Raspberry Pi asks for the coordinates of the point and Arduino Mega traverses to that point.

9 Demonstration at IIT Delhi

At IIT Delhi, we had been presented with several opportunities by Prof. S.K. Saha to showcase our project. All the demonstrations proved to be quite useful as we could know about the improvements that were still needed in the robot and also have some suggestions and feedbacks from the people that saw the demonstration.

The first opportunity we were presented was at the IIT Delhi Open House organised on April 22, 2017, a platform to exhibit an extensive collection of innovative research and product development projects. Many people visited the IIT Delhi premises. There we also presented our robot.

Figure 55: Demonstration at Open House

Then we presented the robot at 3rd SERB Summer School on Robotics from June 23, 2017 to June 28,2017. About 40 participants came from different engineering colleges and industries. The school is planned mainly to introduce basic topics of robotics from the domain of Computer Science and Engineering, Electrical Engineering, and Mechanical Engineering.

Figure 56: Demonstration at SERB Summer School

We also had the opportunity to present our robot on Advances in Robotics Conference held at IIT Delhi. Advances in Robotics (AIR) is a series of biennial conference organized by the Robotics Society of India. The conference aims to create a forum to present and exchange new ideas by researchers and developers from India and abroad working in the fields of robotics and its applications. Renowned professors from national and international universities visited the conference and saw our project.

Figure 57: Advances in Robotics Conference

10 Conclusions and Future Work

The robot is capable of traversing in manual and autonomous mode. It can lift up a payload of up to 15kg. It can detect an obstacle and avoid itself from colliding with it. The robot is capable of distinguishing between a known and an unknown obstacle. It can detect a human figure and acknowledge the person.

Until now the robot is navigating based on the data received from the encoders. In the near future, the navigation capabilities of the robot can be improved. An autonomous charging dock can be designed so that the robot could move to the dock whenever it is low on battery. As this robot has been developed as a mobile robot platform, various researches could be performed on this robot. Various sensors could be integrated within the robot and thus various applications could be tested. If the robot is to be used in surveillance, work could be done on security features of the robot.

References

[1] Roland Siegwart, Illah R. Nourbakhsh "Introduction to Autonomous MobileRobots", The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2004.

[2] K. Goris, "Autonomous mobile robot mechanical design," Thesis academic degree Civil Mechanical-Electrical Engineering, Vrije Universiteit Brussel, 2005.

[3] http://www.banebots.com/category/MOTORS.html

[4]http://www.banebots.com/product/M5-RS540-12.html

[5] www.sfhsrobotics.com/Documents/Game_Manuals/.../Kit.../Motor_Curves.pdf

[6] http://www.banebots.com/category/P60_5.html

[7] http://www.banebots.com/product/P60S-433-5.html

[8] http://www.robotshop.com/en/banebots-rs-540-motor.html

[9] https://www.dimensionengineering.com/products/sabertooth2x60

[10]https://www.amazon.es/pantalla-t%C3%A1ctil-800%C3%97480-Raspberry-sistemas/dp/B00Y84K25W

[11]http://circuitfreak.net/make-your-own-kinect-calibration-card/

[12] http://www.robotshop.com/ca/en/hc-sr04-ultrasonic-range-finder.html

[13] http://www.hobbytronics.co.uk/arduino-uno-r3

[14]https://www.arduino.cc/en/Main/arduinoBoardMega

[15]https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/2020826.pdf

[16]http://coder-tronics.com/pid-tutorial-c-code-example-pt1/

[17]https://en.wikipedia.org/wiki/PID_controller

[18] J. Borenstein, H. R. Everett and L. Feng, Where am I?" -Systems and Methods for Mobile RobotPositioning, 1996.

[19] C. P. Papageorgiou, M. Oren and T. Poggio, "A general framework for object detection," in *International Conference on Computer Vision*, 1988.

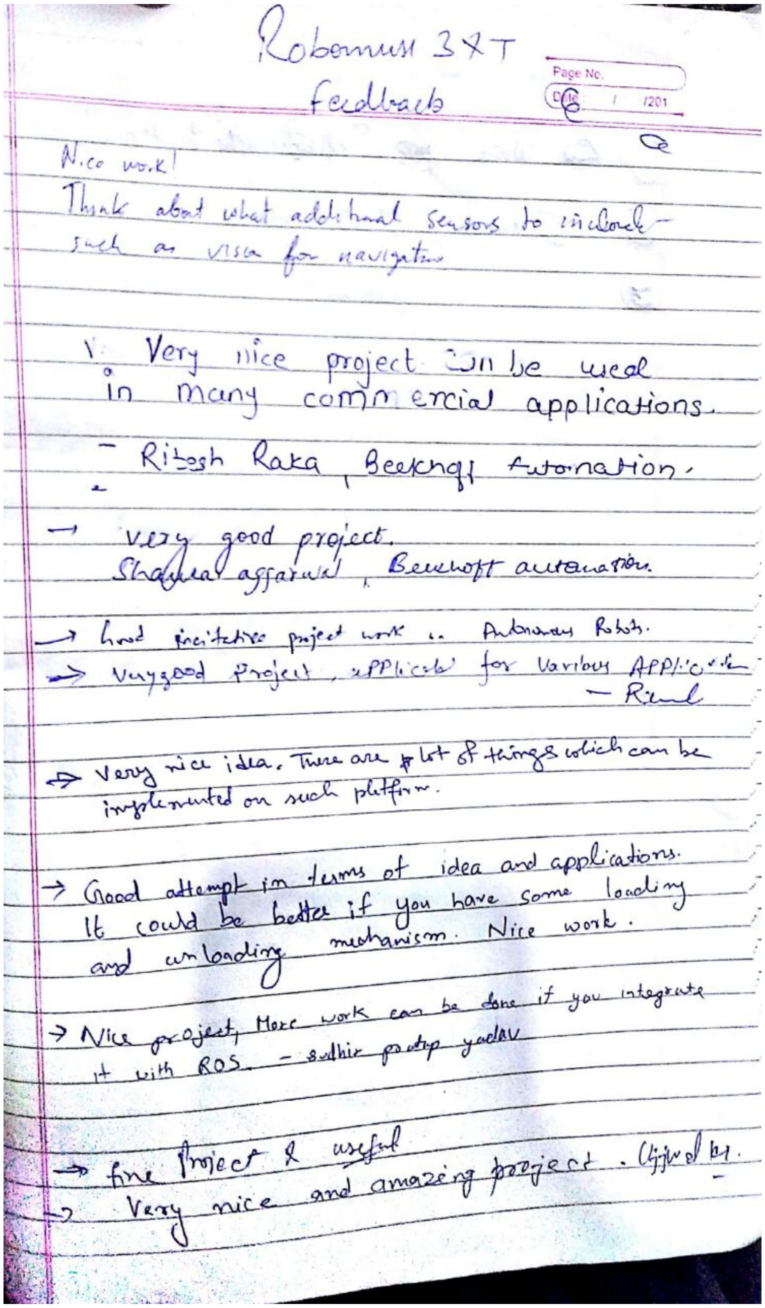
[20] M. Jones and P. Viola, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition*, 2001

[21]http://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Object_Detection_Face_Detection_Haar_Cascade_Classifiers.php

[22] http://hanzratech.in/2015/02/03/face-recognition-using-opencv.html

Appendix

A Reviews, Comments and Feedback



Robomuss 3XT
Feedback

Page No.
Date / /201

Nice work!
Think about what additional sensors to include – such as vision for navigation

Very nice project can be used in many commercial applications.
– Ritesh Raka, Beckhoff Automation.

→ very good project.
Shashank aggarwal, Beckhoff automation.

→ Good initiative project work .. Autonomous Robots.
→ Very good project, applicable for various applications
– Rahul

→ Very nice idea. There are a lot of things which can be implemented on such platform.

→ Good attempt in terms of idea and applications. It could be better if you have some loading and unloading mechanism. Nice work.

→ Nice project, More work can be done if you integrate it with ROS. – sudhir pratap yadav

→ fine Project & useful
→ Very nice and amazing project. Ujjwal M.

Robomuus 3XT

Page No
Date / /201

Feedback

· Excellent work on ~~int~~ fusion of sensors in motion control.
Best wishes.

Good working project & demo

Good Start learn from stretch instead of using arduino Library. You can look for e-Yantra robotics Competition to learn more. Optimize Hardware, might design your own PCB & power optimization.

Suraiv Shandilya

Really excited that students are working on practical, reliable, product based design. I hope you go forward, first considering the requirements and working towards achieving those goals.

ASHWIN KURUTTUKULAM

Exciting work. Nice Platform. Wish to see localization as future work. - Thanks

Abhijit Biswas

Flawless running nice! Put it in some real use. Thank you

Feedback

Robomuse 3XT

Page No.

Date : / /201

① Video Record ② Target, facial recognisation

★ Nice Idea, Great, keep going.

Awesome work. Keep it up.

Page No.

Date : / /201

Feedback

Impressive work. Future scope of this system is really promising one. All the best for the team.

— Robomuse —

1/8/2017

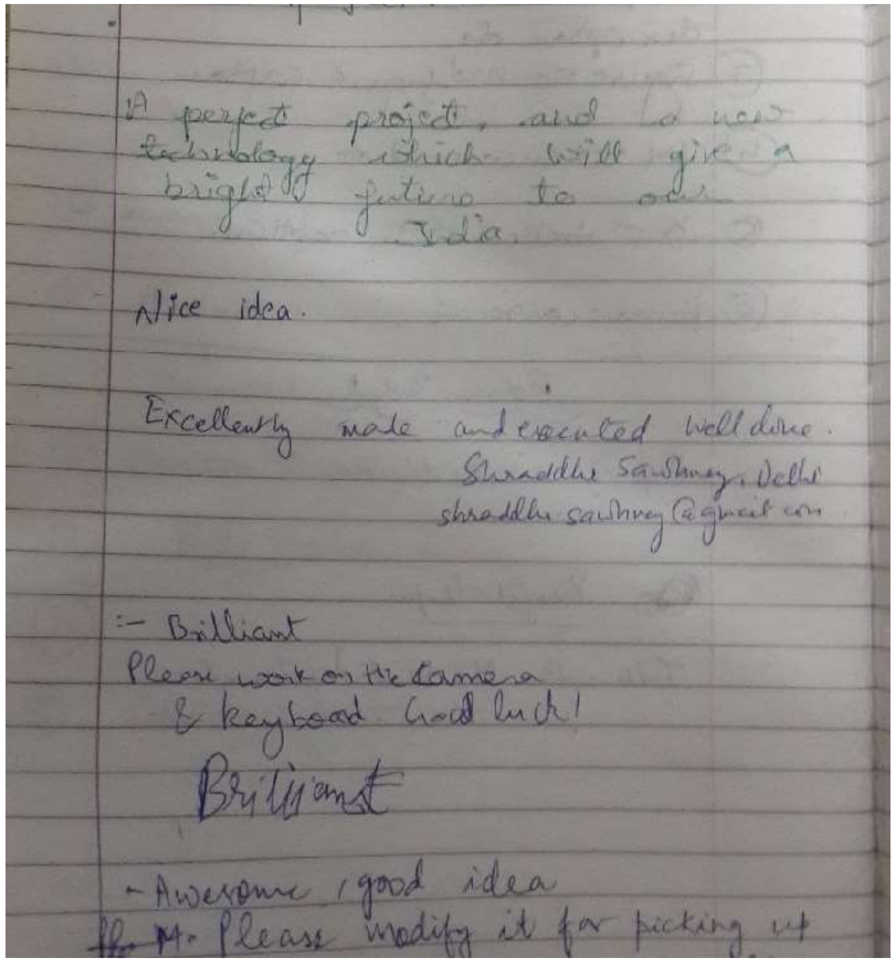
(Dr. S. S. Ohol)

COEP.

It was excellent getting to know about the work/project on Intruder Detection System. Good luck to the Project Team. Plenty of Scope for way forward.

(NEERAJ LAL)

DGM, ONGC.


A perfect project, and a new technology which will give a bright future to our India.

Nice idea.

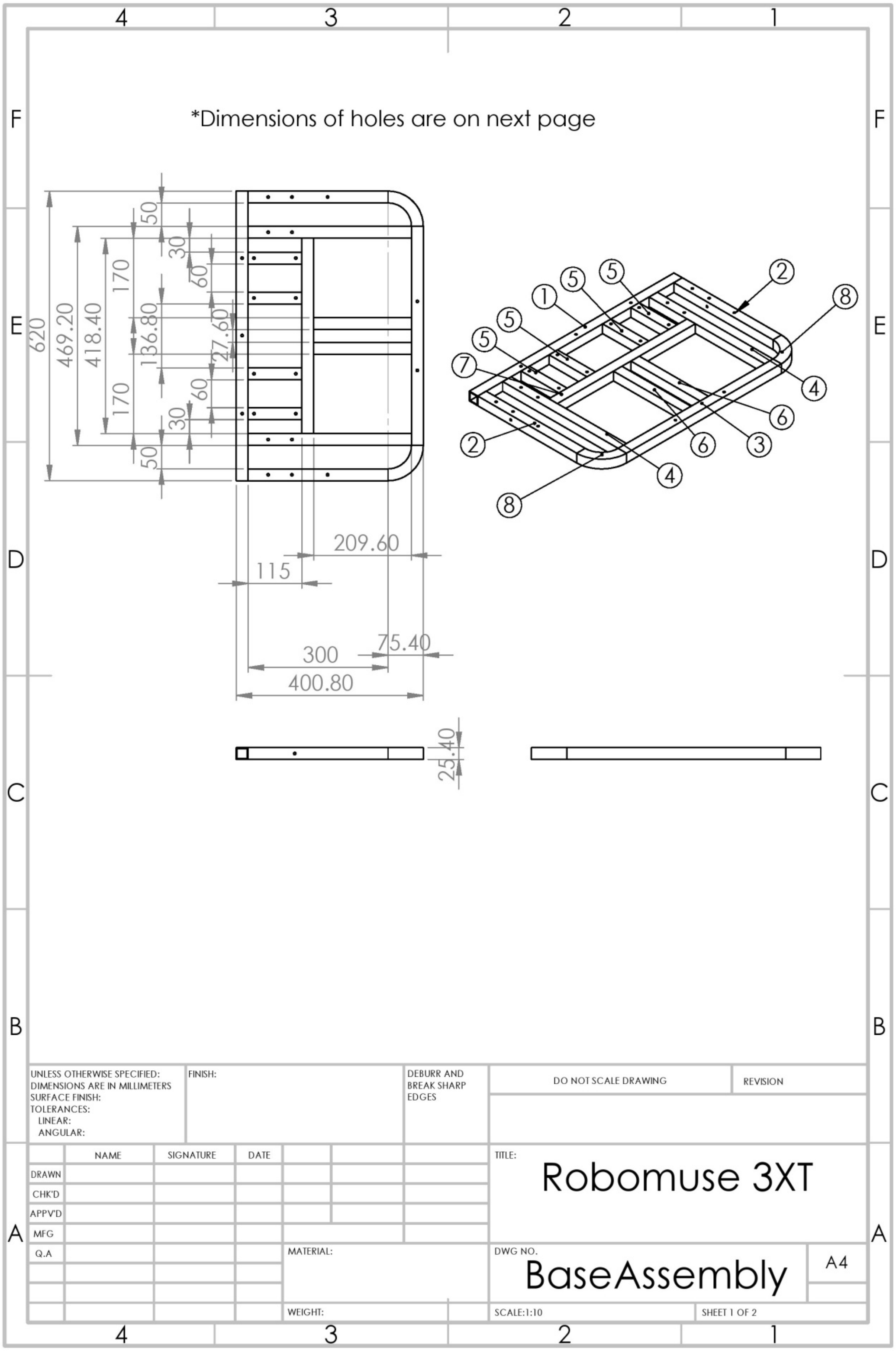
Excellently made and executed well done.
Shraddha Sawhney, Delhi
shraddha.sawhney@gmail.com

:- Brilliant
Please work on the camera & keyboard Good luck!

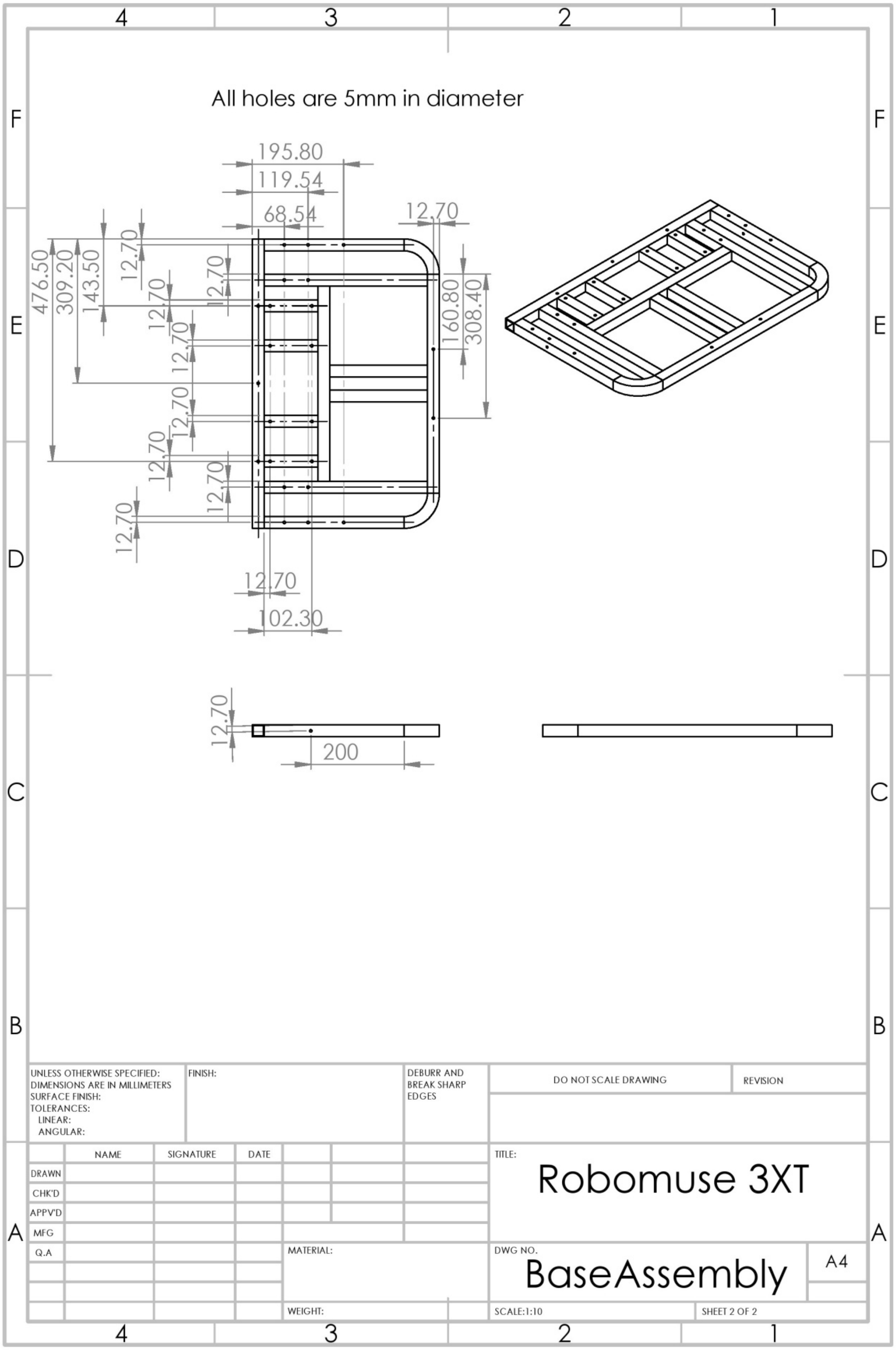
Brilliant

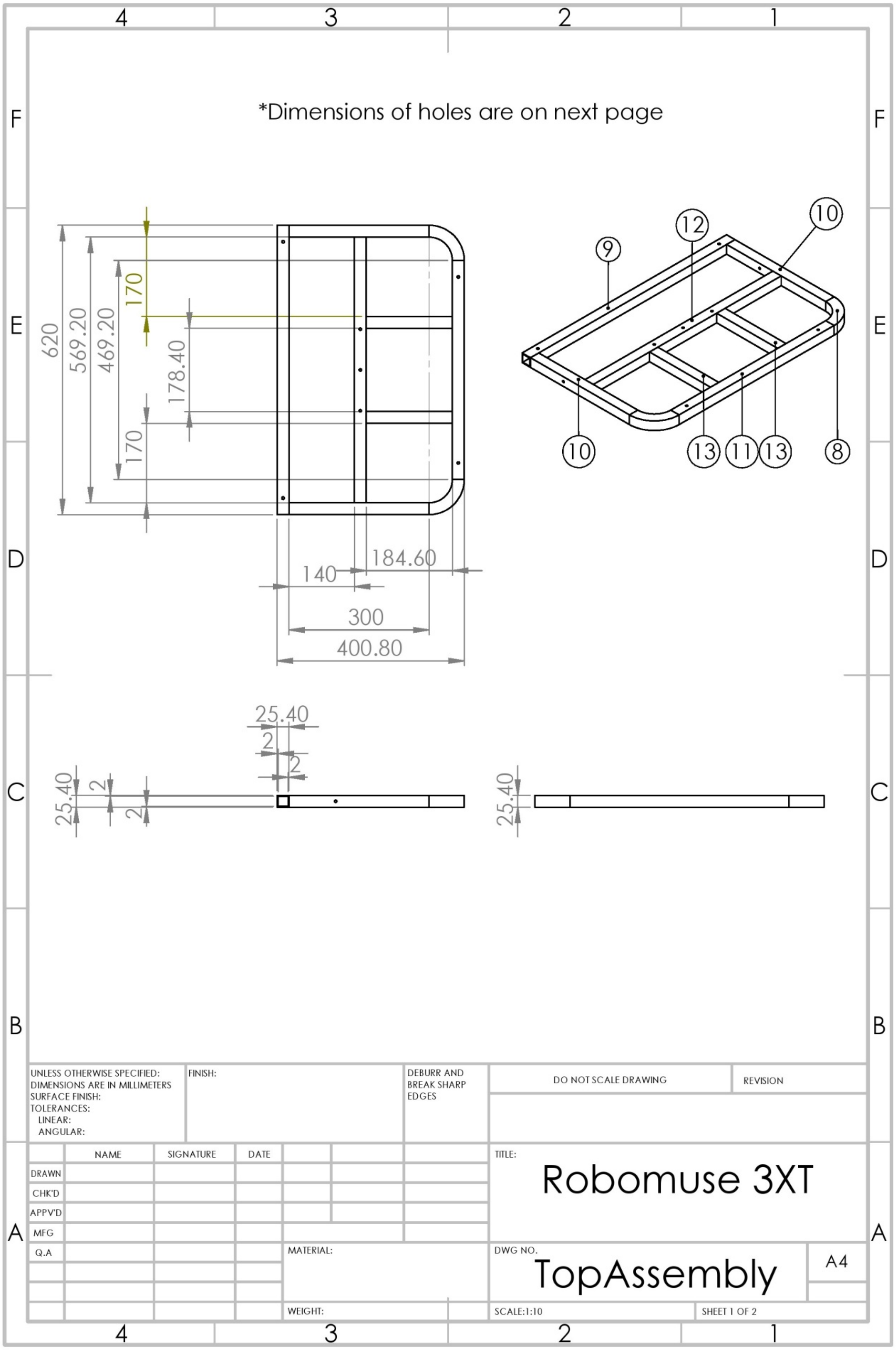
- Awesome / good idea
~~the M.~~ Please modify it for picking up

B Drawings

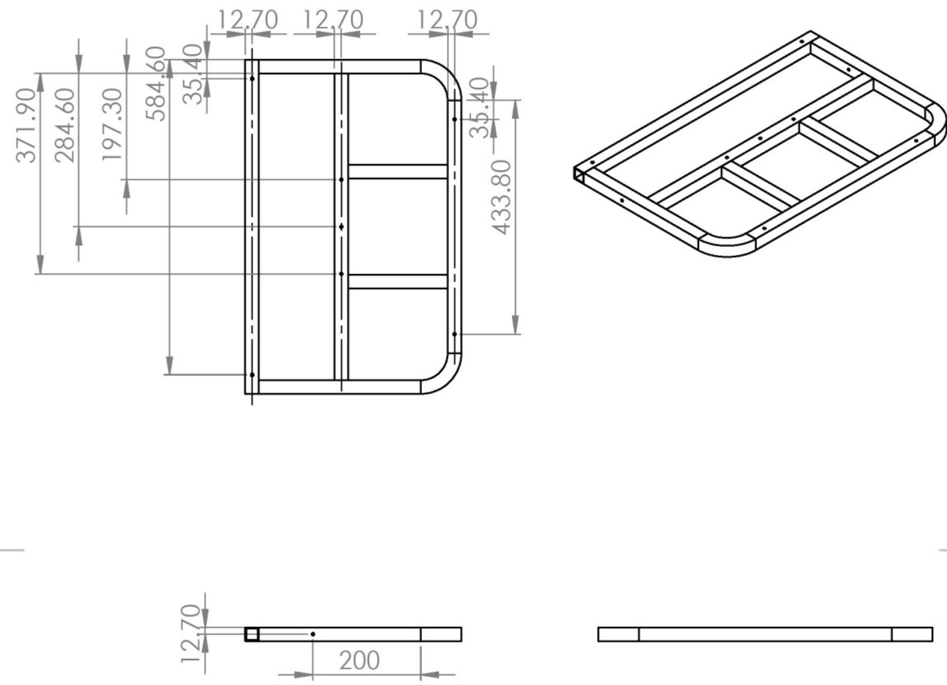


*Dimensions of holes are on next page
620
469.20
418.40
170
50
30
60
136.80
27.60
209.60
115
300
75.40
400.80
25.40
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
Robomuse 3XT
DWG NO.
BaseAssembly
A4
SCALE:1:10
SHEET 1 OF 2


All holes are 5mm in diameter
195.80
119.54
68.54
12.70
476.50
309.20
143.50
160.80
308.40
102.30
200
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
BaseAssembly
A4
WEIGHT:
SCALE:1:10
SHEET 2 OF 2


*Dimensions of holes are on next page
620
569.20
469.20
170
178.40
170
140
184.60
300
400.80
25.40
2
2
2
25.40
2
2
25.40
9
12
10
10
13
11
13
8
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
Robomuse 3XT
DWG NO.
TopAssembly
A4
SCALE:1:10
SHEET 1 OF 2

All holes are 5 mm in diameter



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE: Robomuse 3XT

MATERIAL:

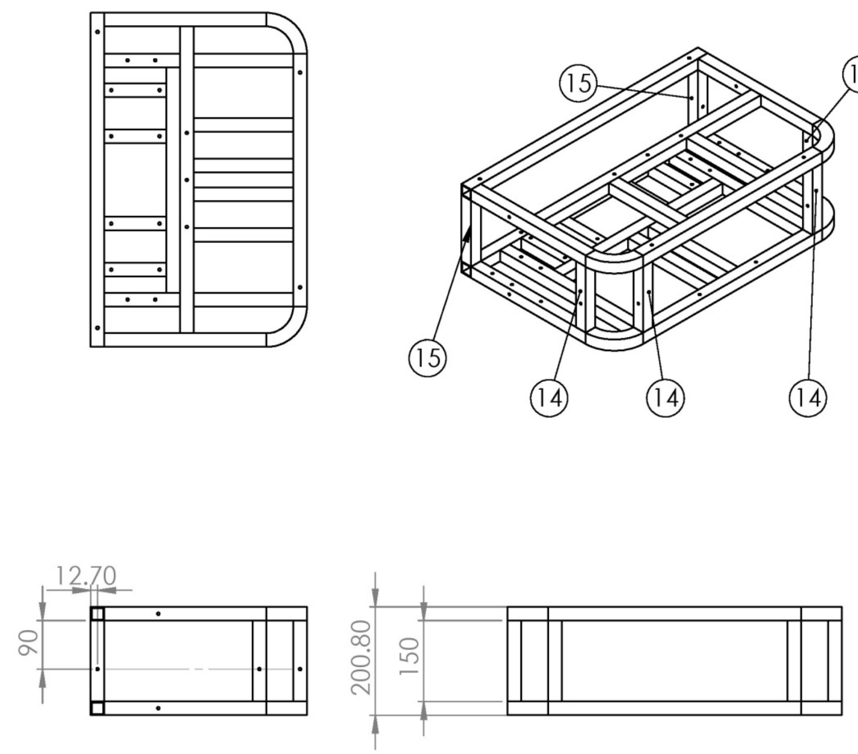
DWG NO. TopAssembly

A4

WEIGHT:

SCALE:1:10

SHEET 2 OF 2



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE: Robomuse 3XT

MATERIAL:

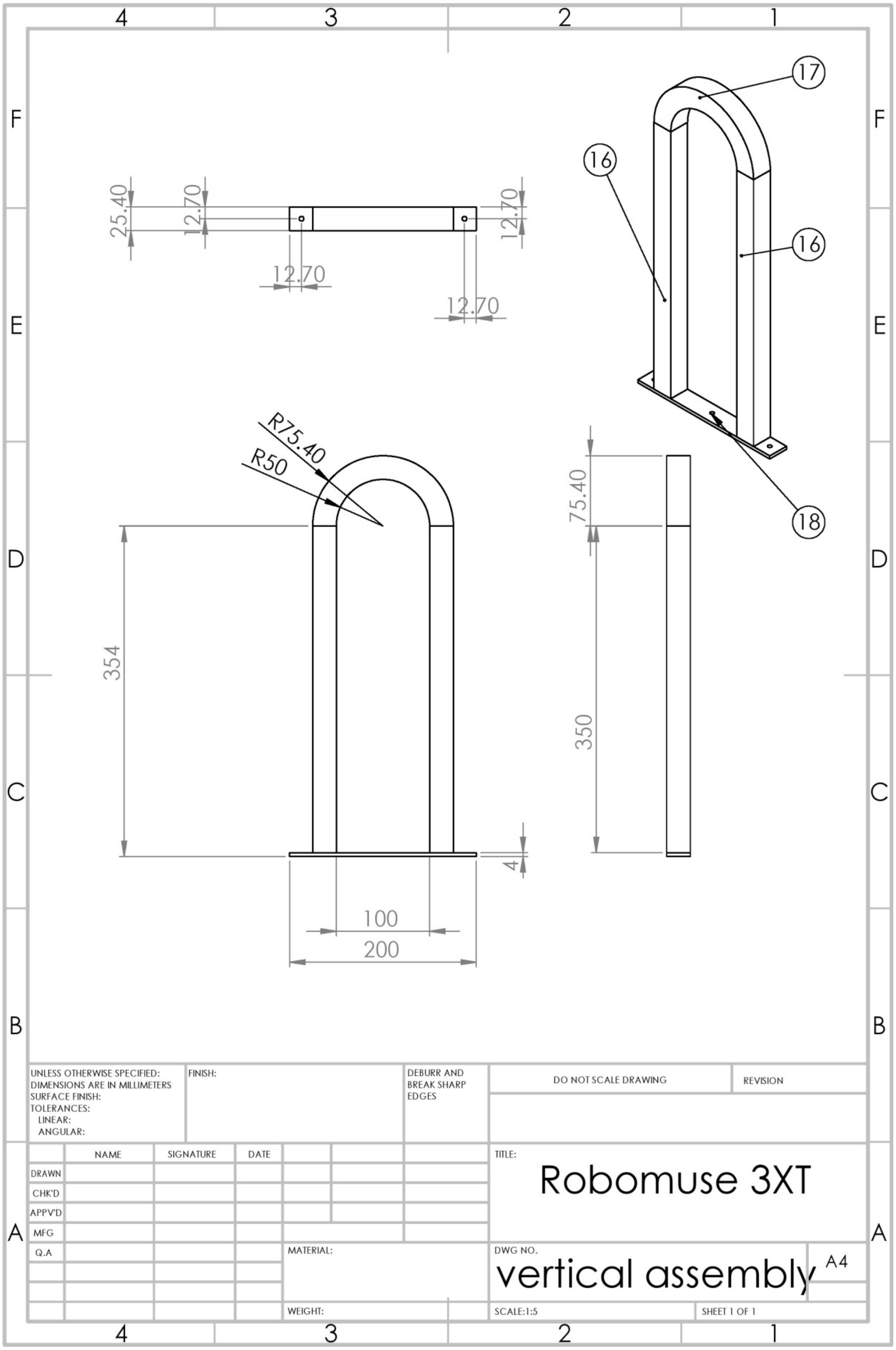
DWG NO. Chassis

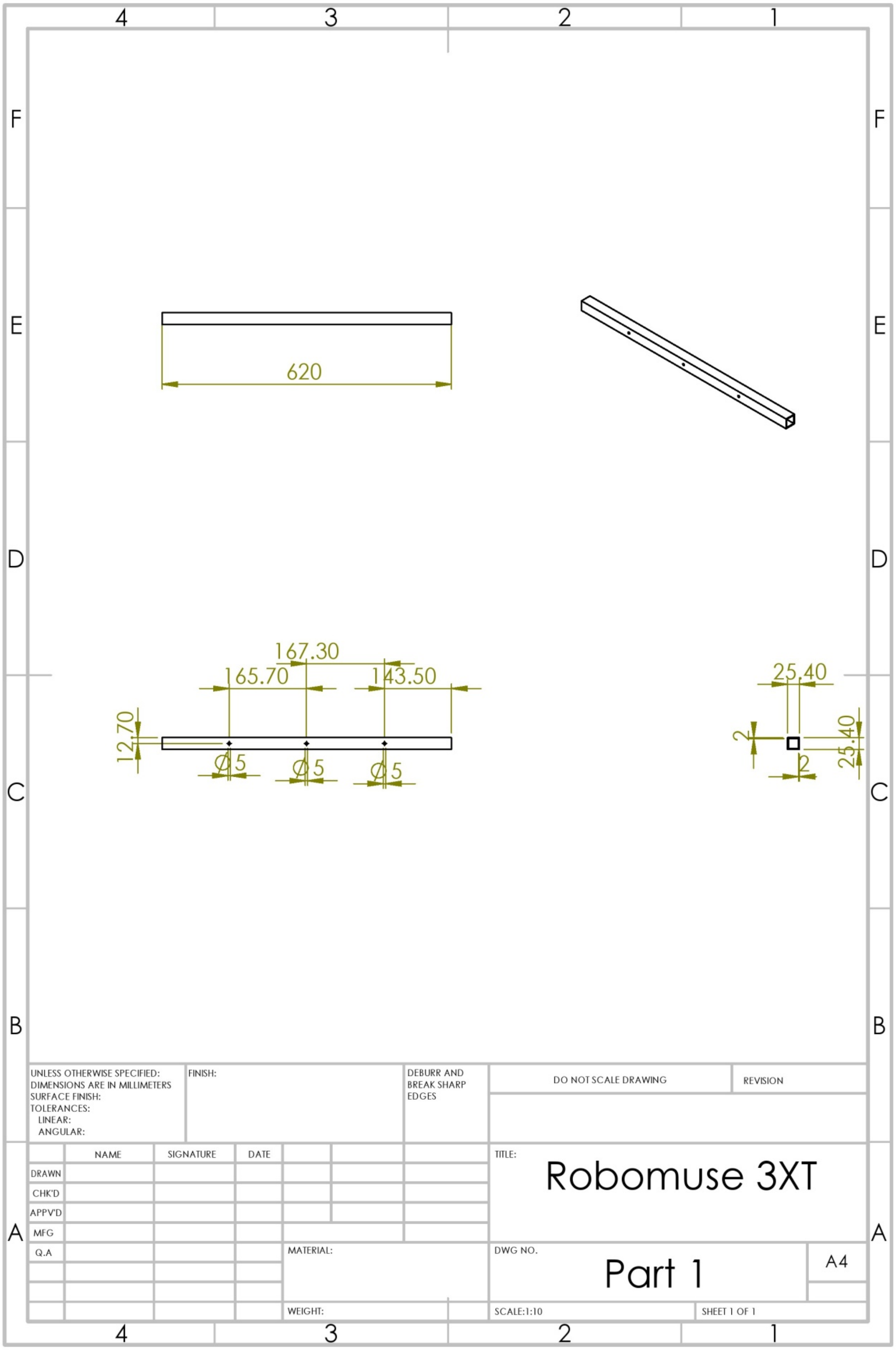
A4

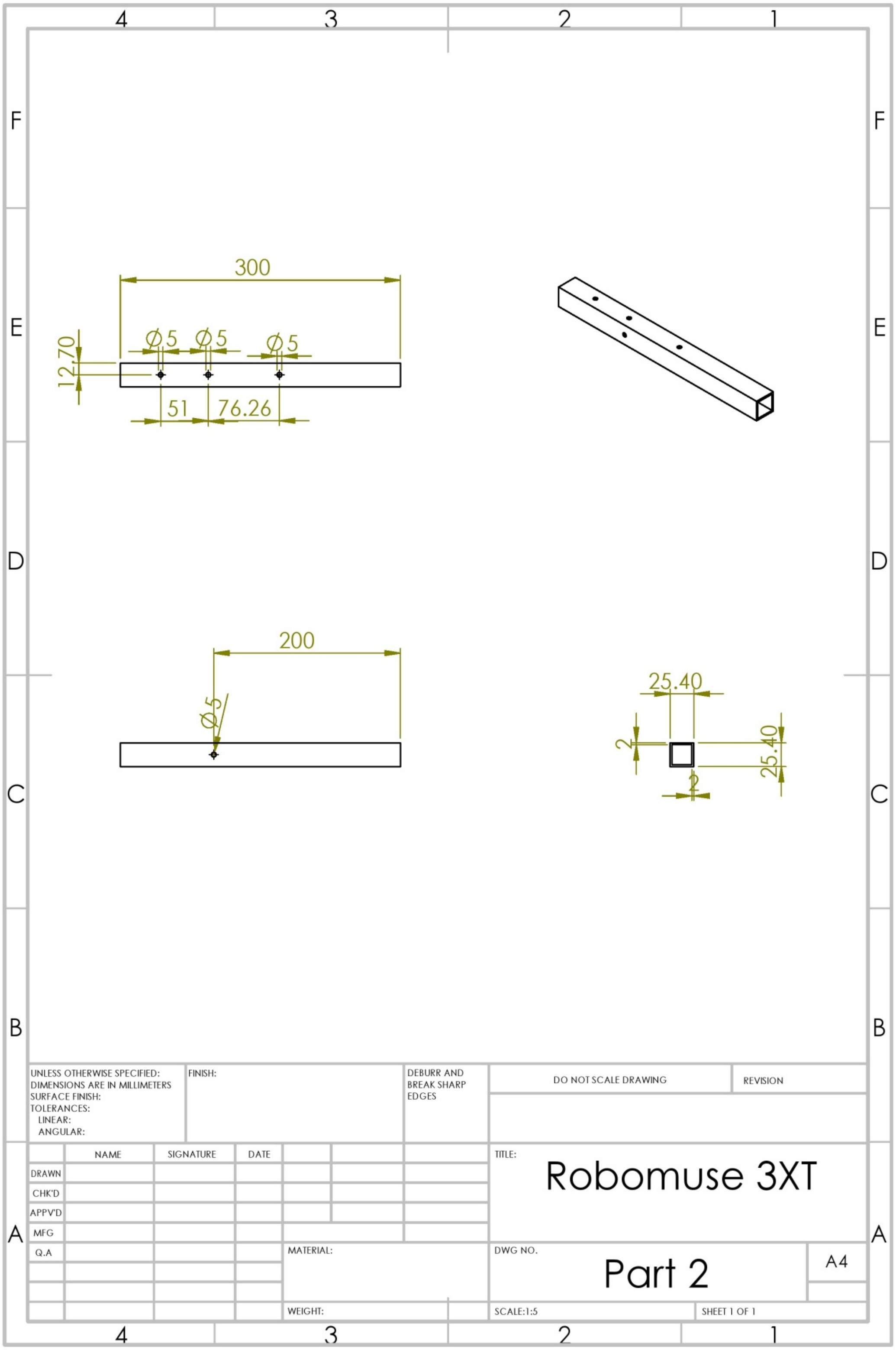
WEIGHT:

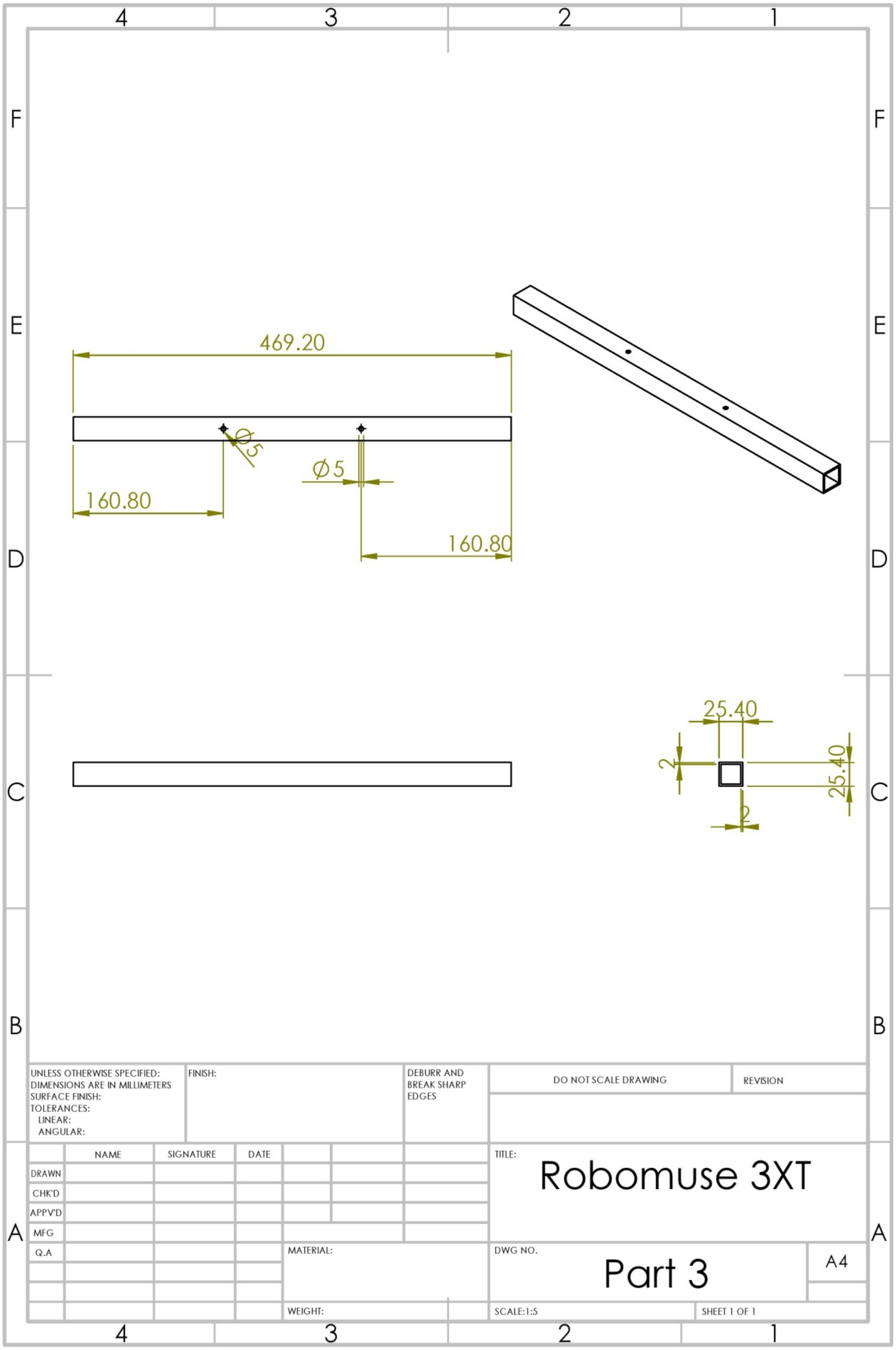
SCALE:1:10

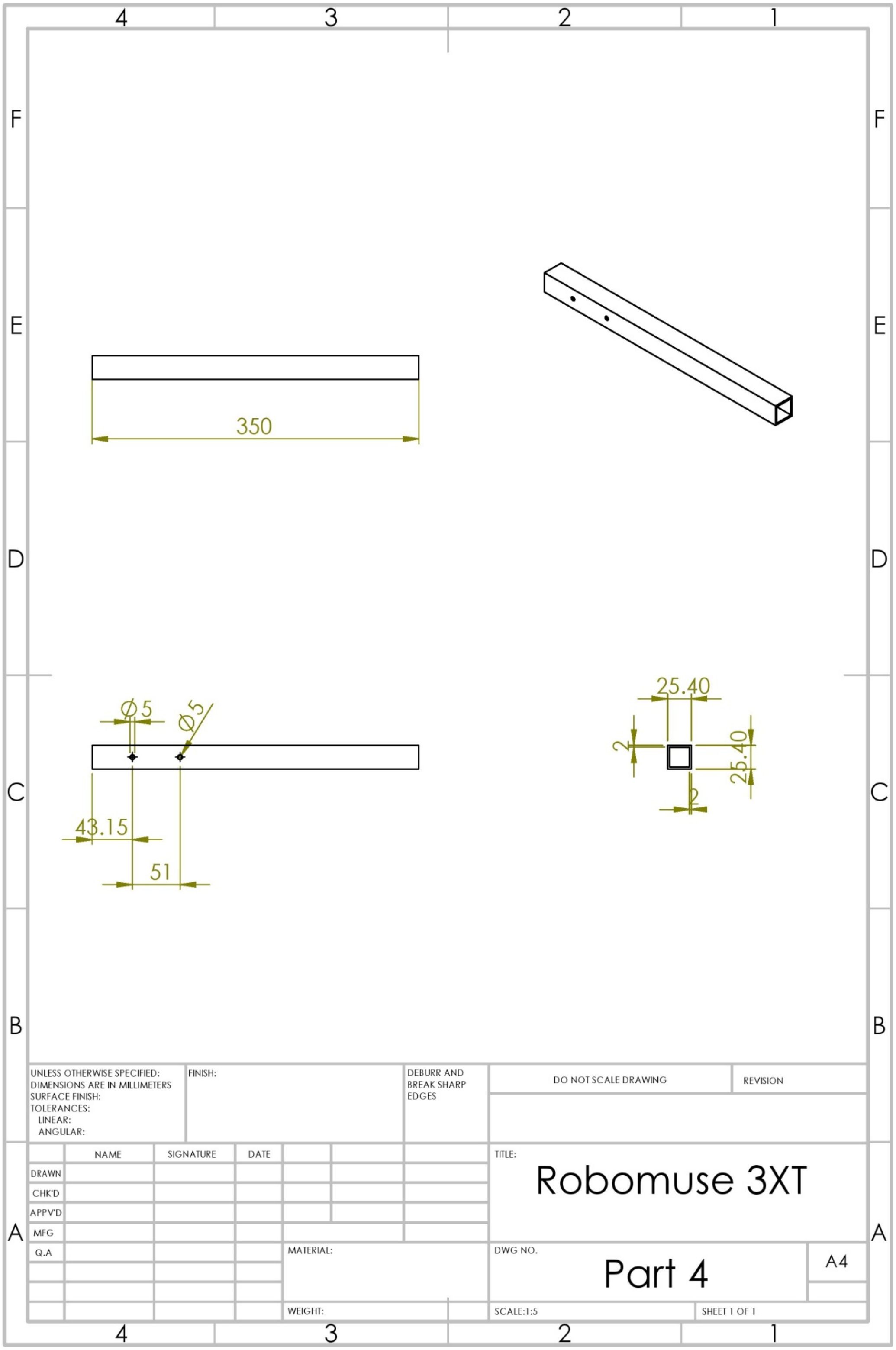
SHEET 1 OF 1

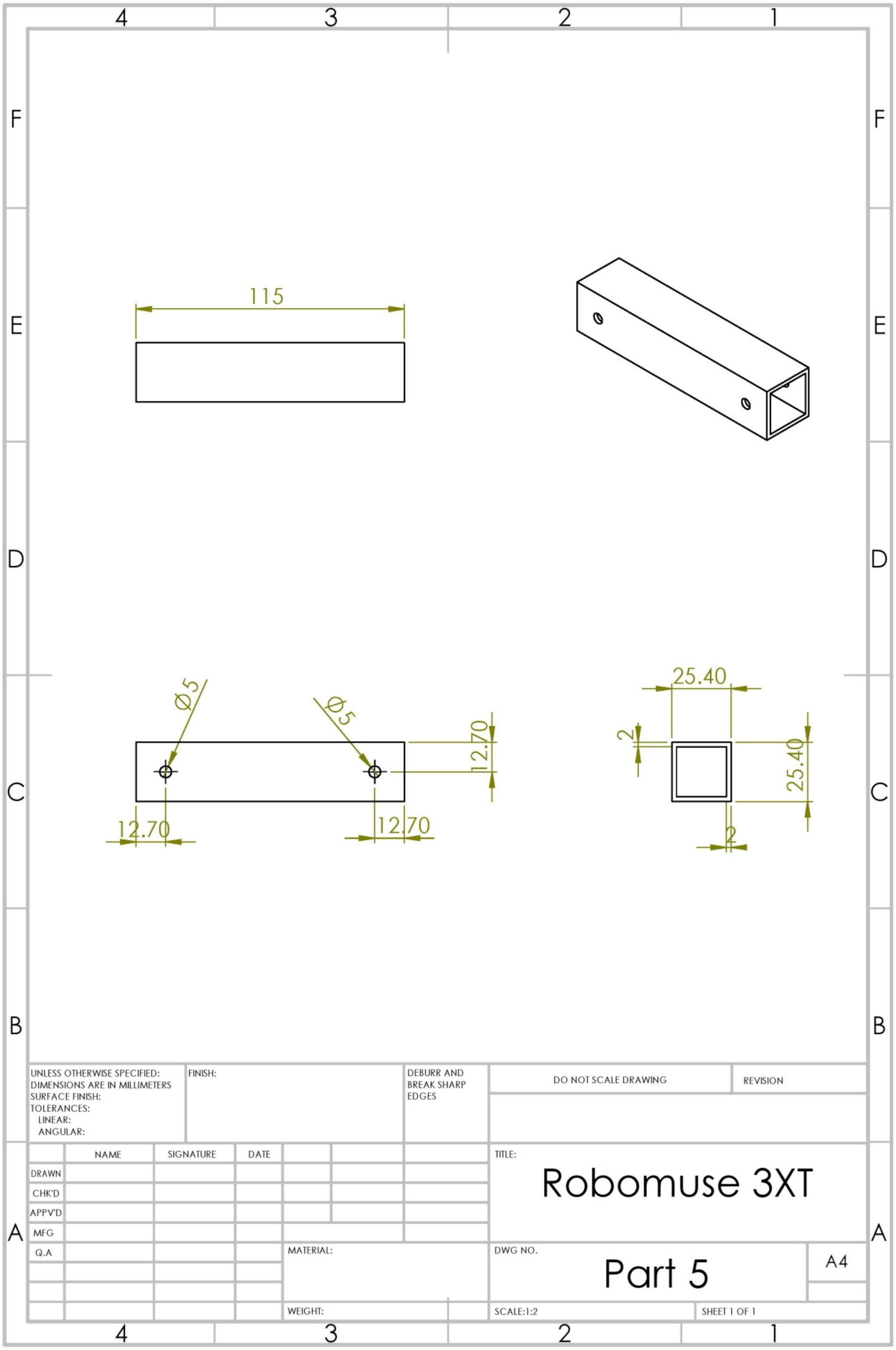

25.40
12.70
12.70
12.70
12.70
R75.40
R50
354
75.40
350
4
100
200
16
17
16
18
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
Robomuse 3XT
DWG NO.
vertical assembly
A4
SCALE:1:5
SHEET 1 OF 1

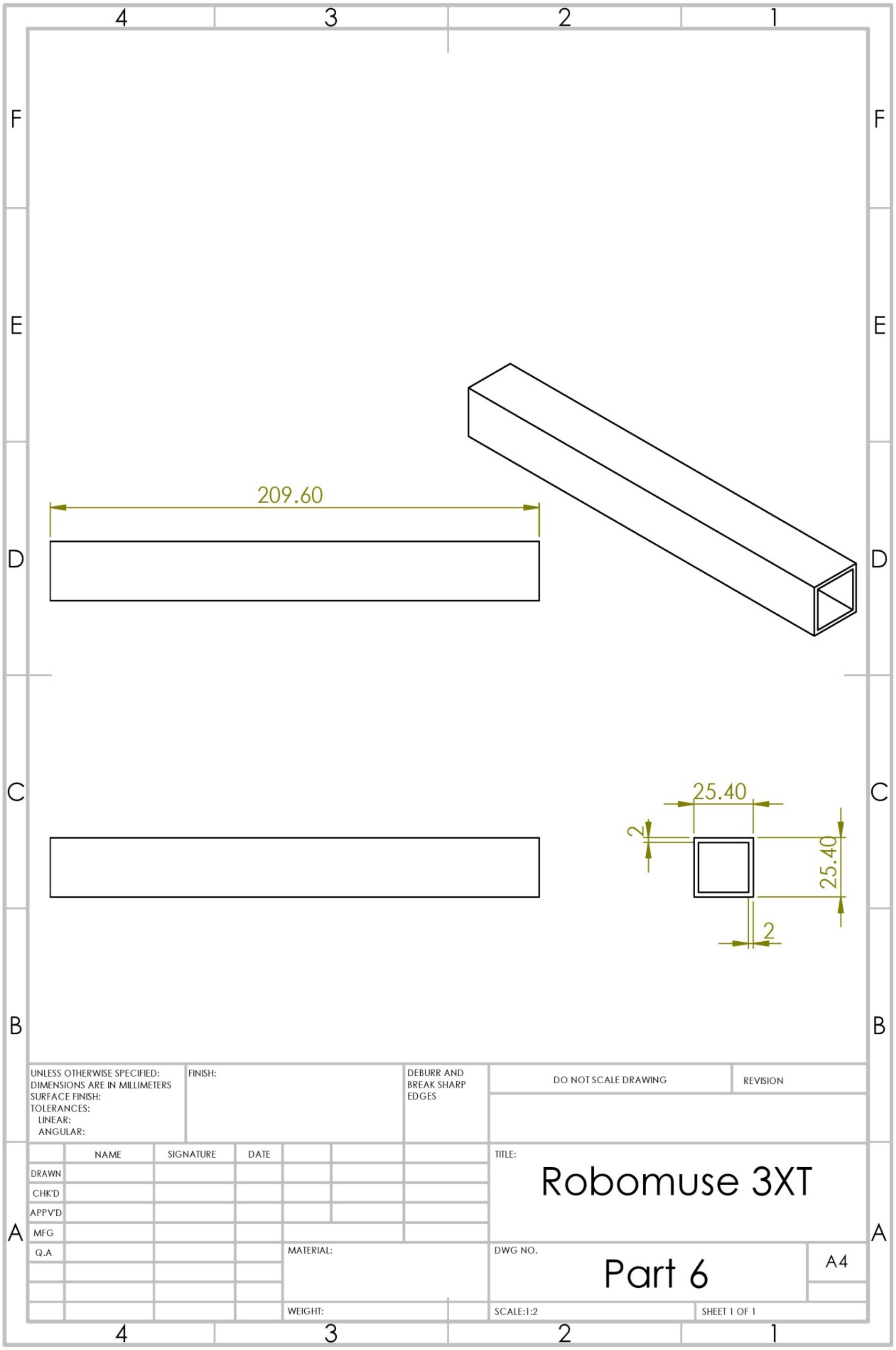

620
167.30
165.70
143.50
12.70
Ø5
Ø5
Ø5
25.40
25.40
2
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 1
A4
WEIGHT:
SCALE:1:10
SHEET 1 OF 1


300
Ø5
Ø5
Ø5
12.70
51
76.26
200
Ø5
25.40
25.40
2
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
Robomuse 3XT
DWG NO.
Part 2
A4
SCALE:1:5
SHEET 1 OF 1

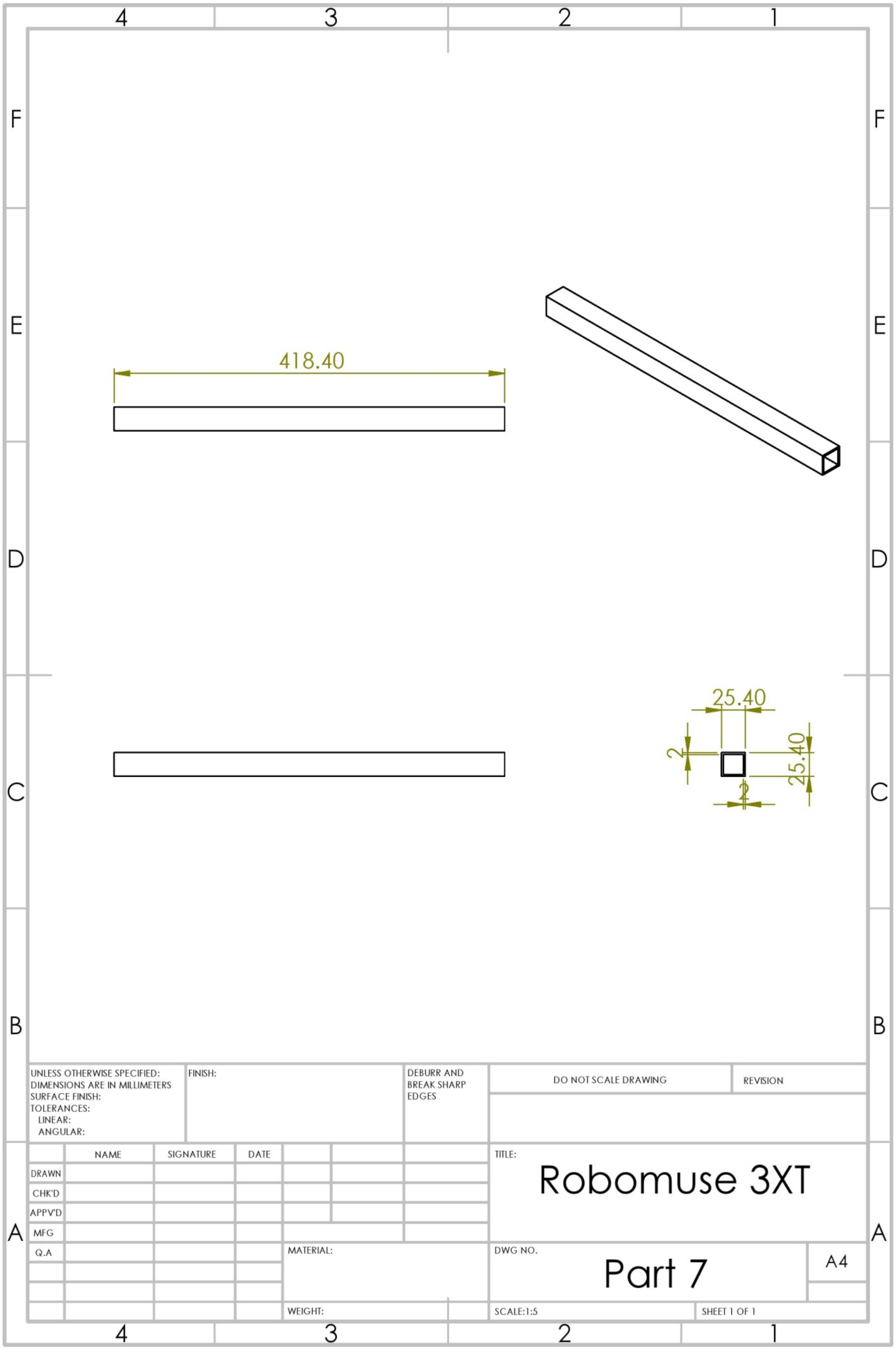

469.20
Ø5
Ø5
160.80
160.80
25.40
2
25.40
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 3
A4
WEIGHT:
SCALE:1:5
SHEET 1 OF 1

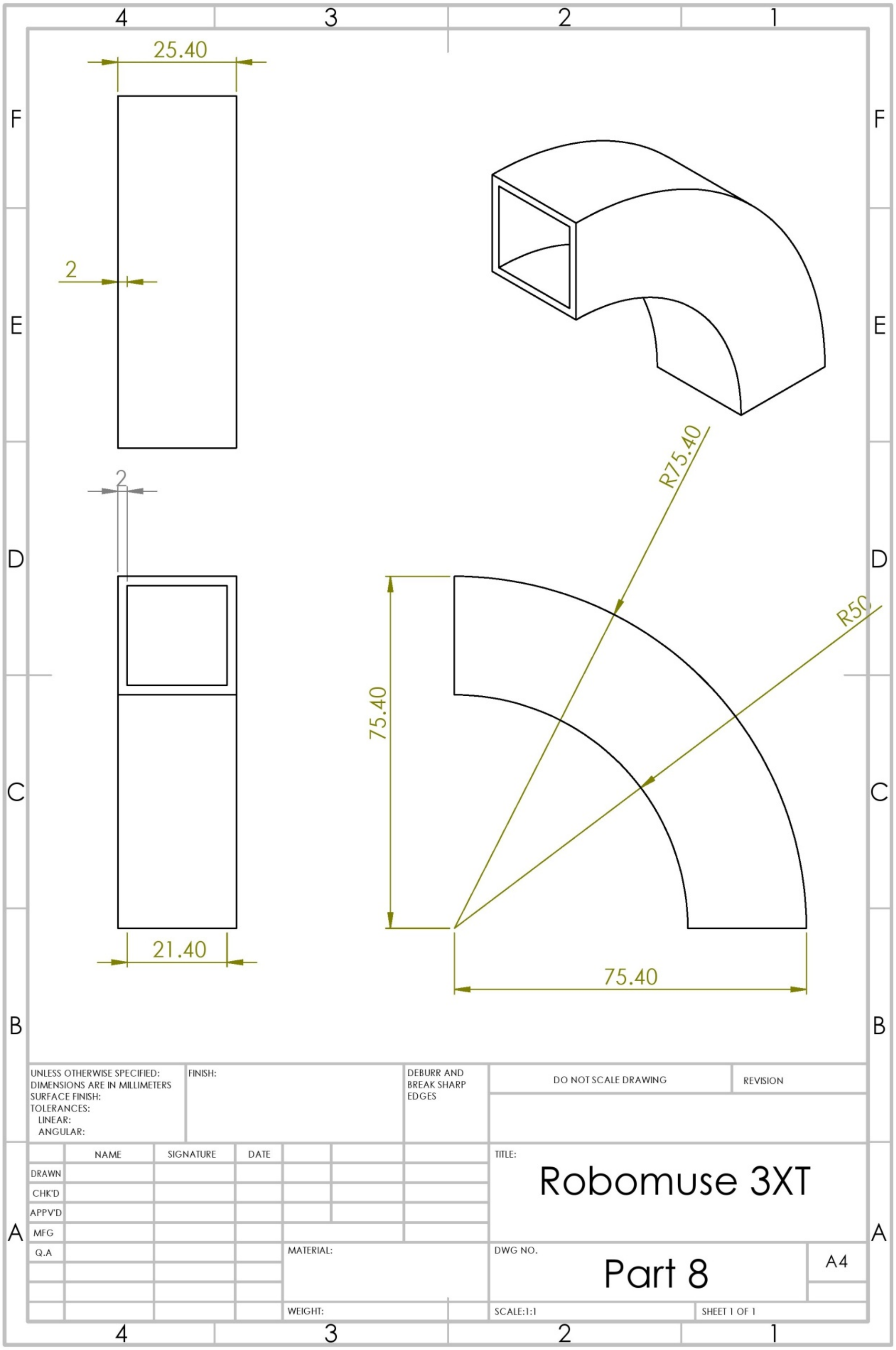

350
⌀5
⌀5
43.15
51
25.40
2
25.40
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 4
A4
WEIGHT:
SCALE:1:5
SHEET 1 OF 1

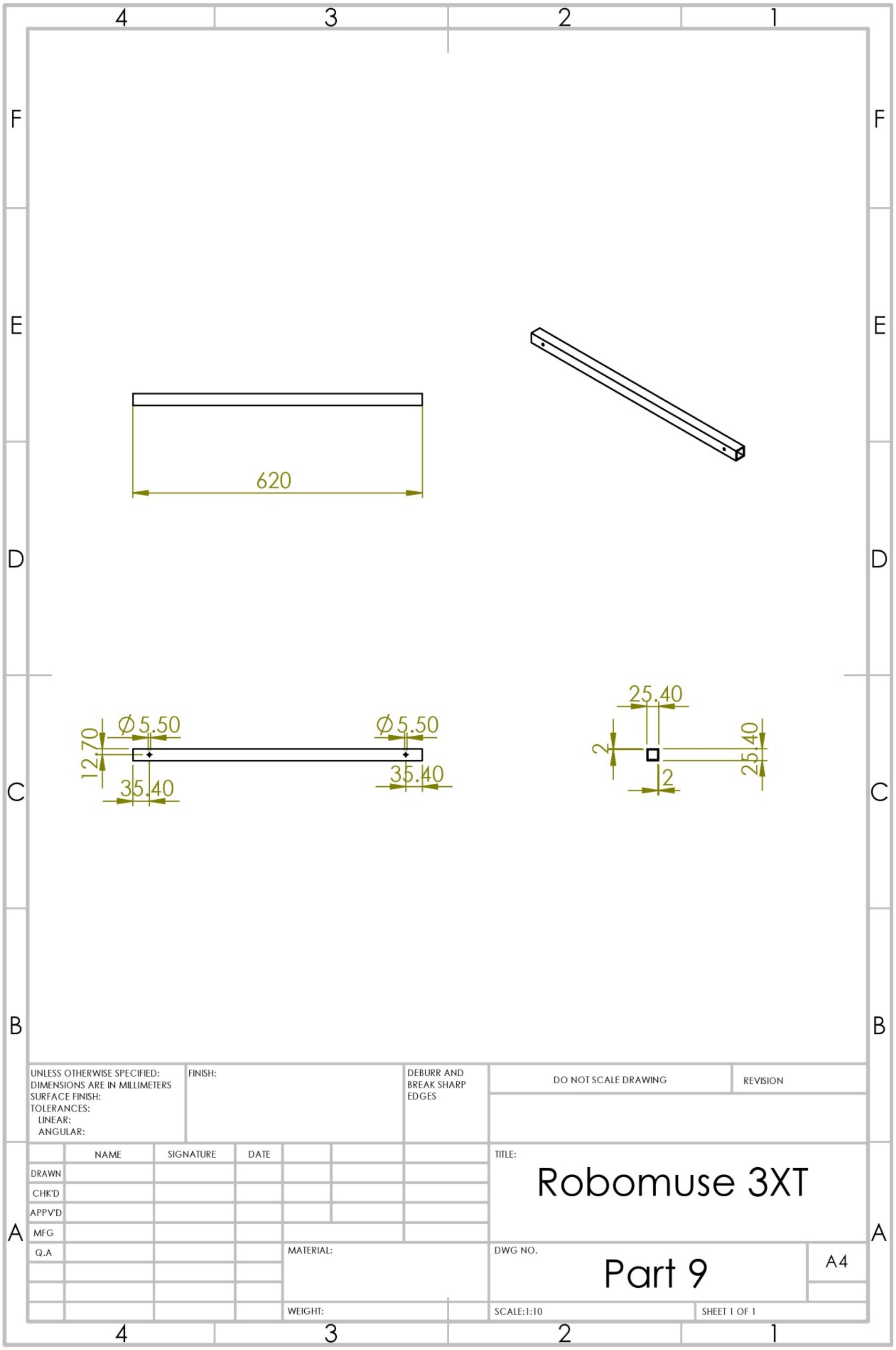

115
Ø5
Ø5
12.70
12.70
12.70
25.40
25.40
2
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
Robomuse 3XT
DWG NO.
Part 5
A4
SCALE:1:2
SHEET 1 OF 1

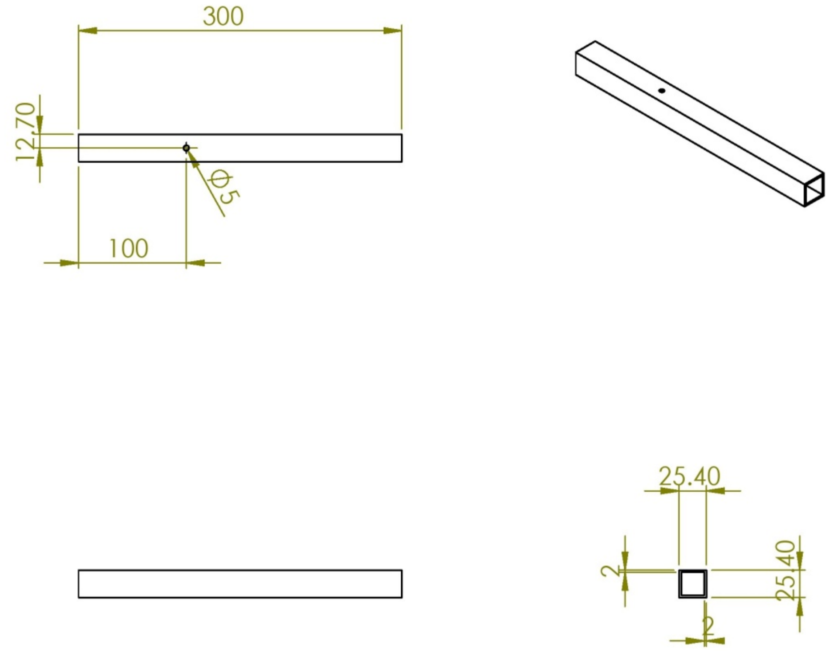


209.60
25.40
25.40
2
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 6
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1

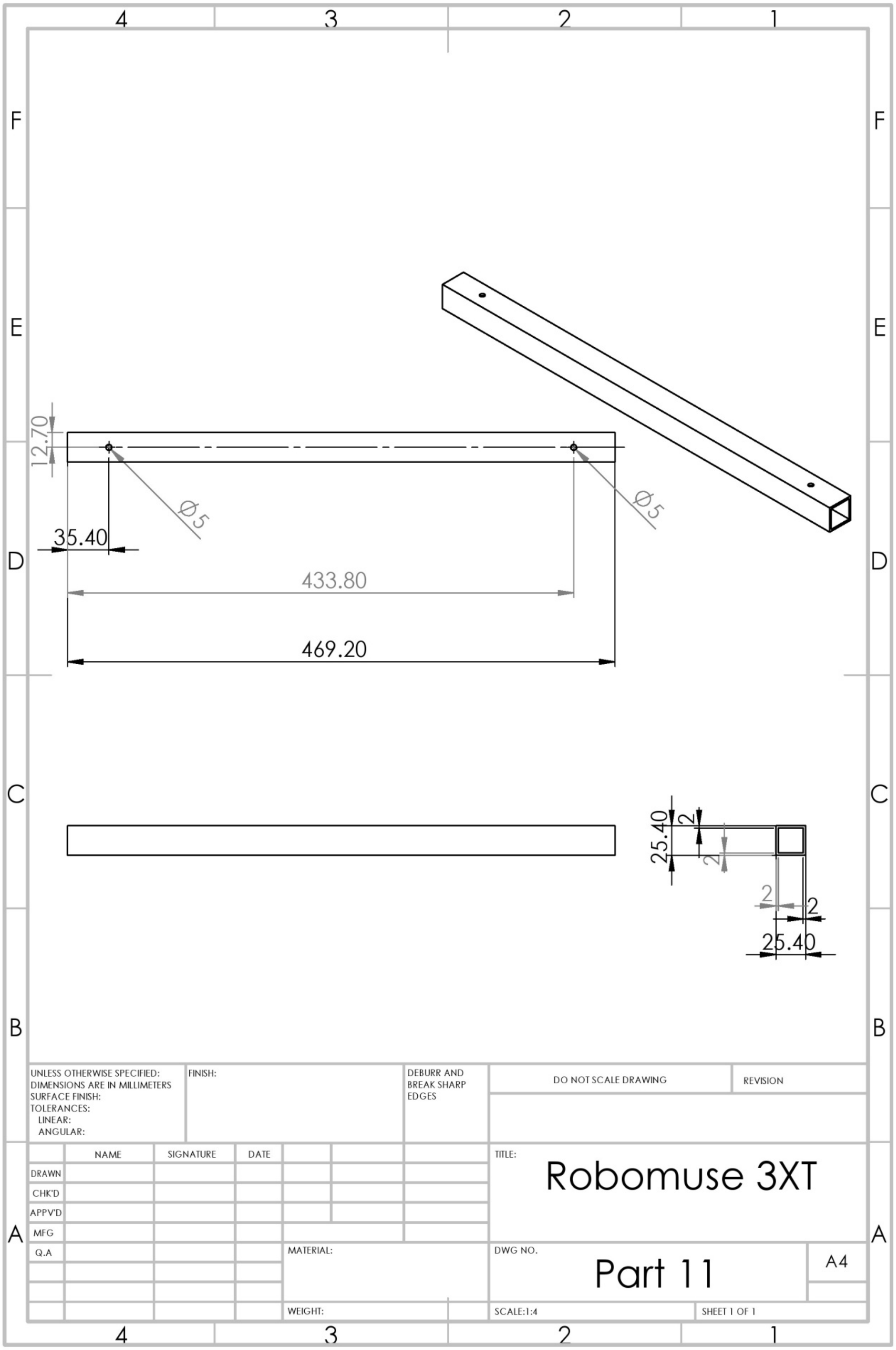

418.40
25.40
2
25.40
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND
BREAK SHARP
EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 7
A4
WEIGHT:
SCALE:1:5
SHEET 1 OF 1


4
3
2
1
F
E
D
C
B
A
25.40
2
2
21.40
75.40
75.40
R75.40
R50
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 8
A4
WEIGHT:
SCALE:1:1
SHEET 1 OF 1

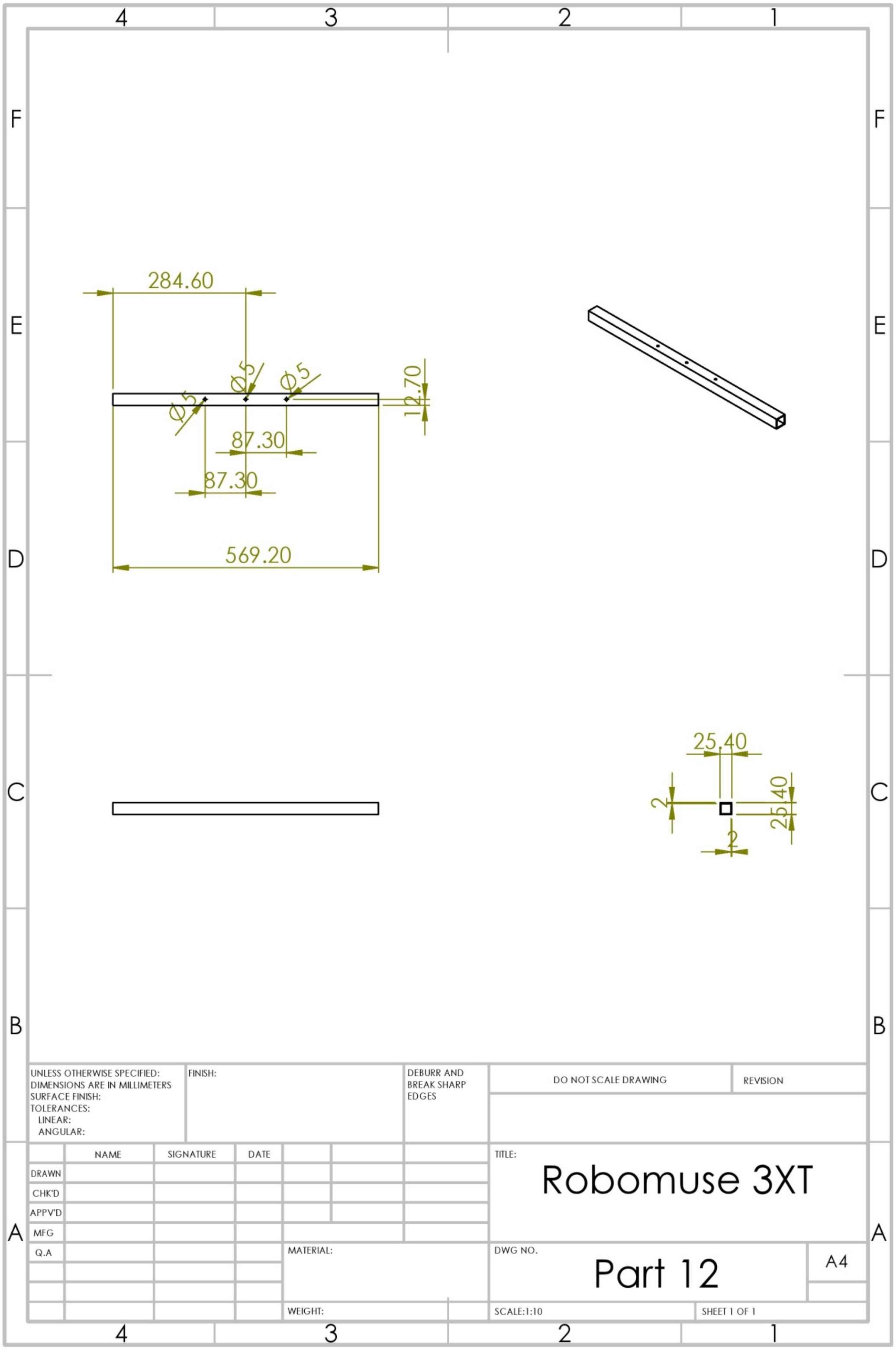

620
Ø5.50
Ø5.50
12.70
35.40
35.40
25.40
2
2
25.40
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 9
A4
WEIGHT:
SCALE:1:10
SHEET 1 OF 1

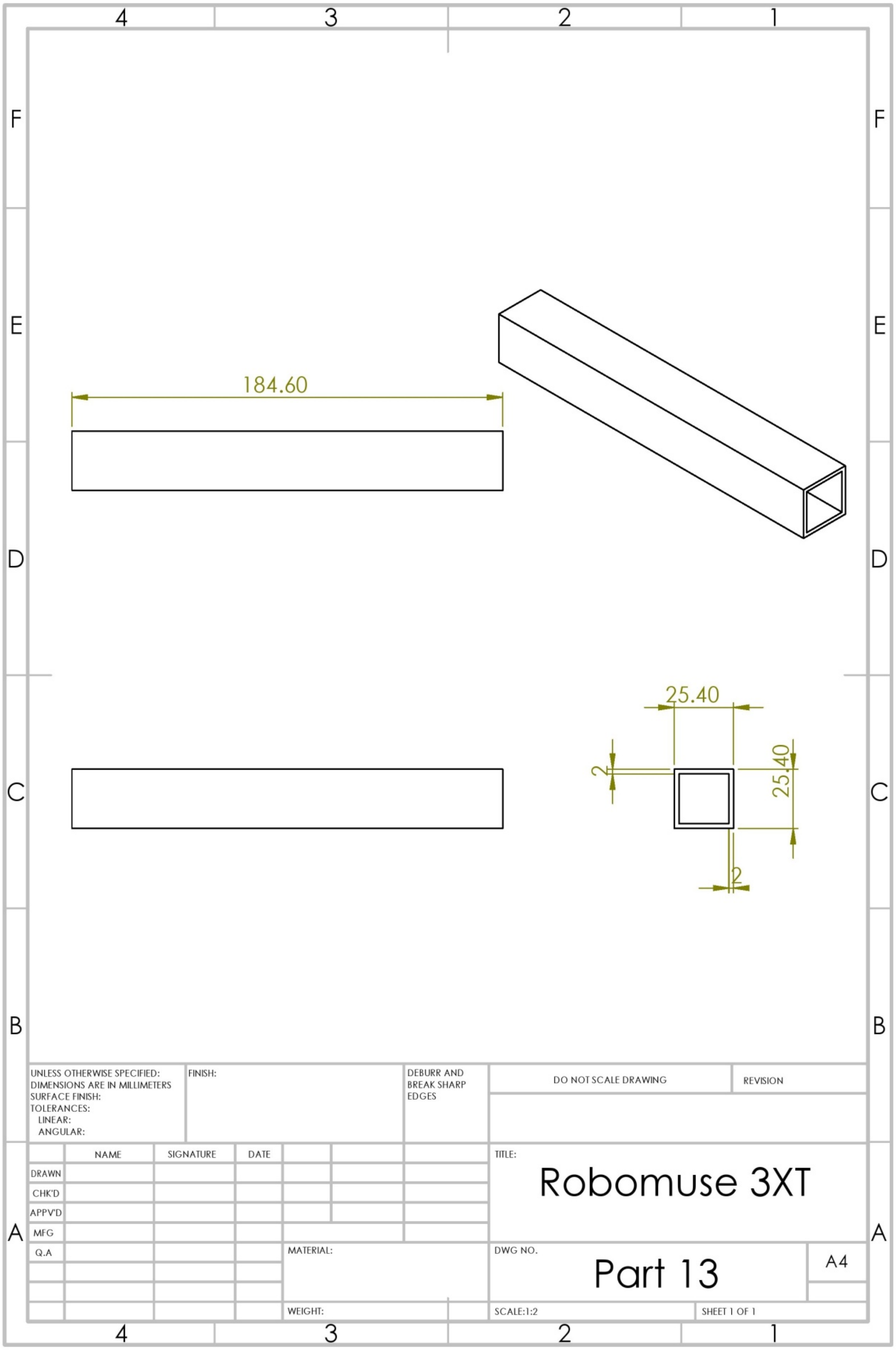


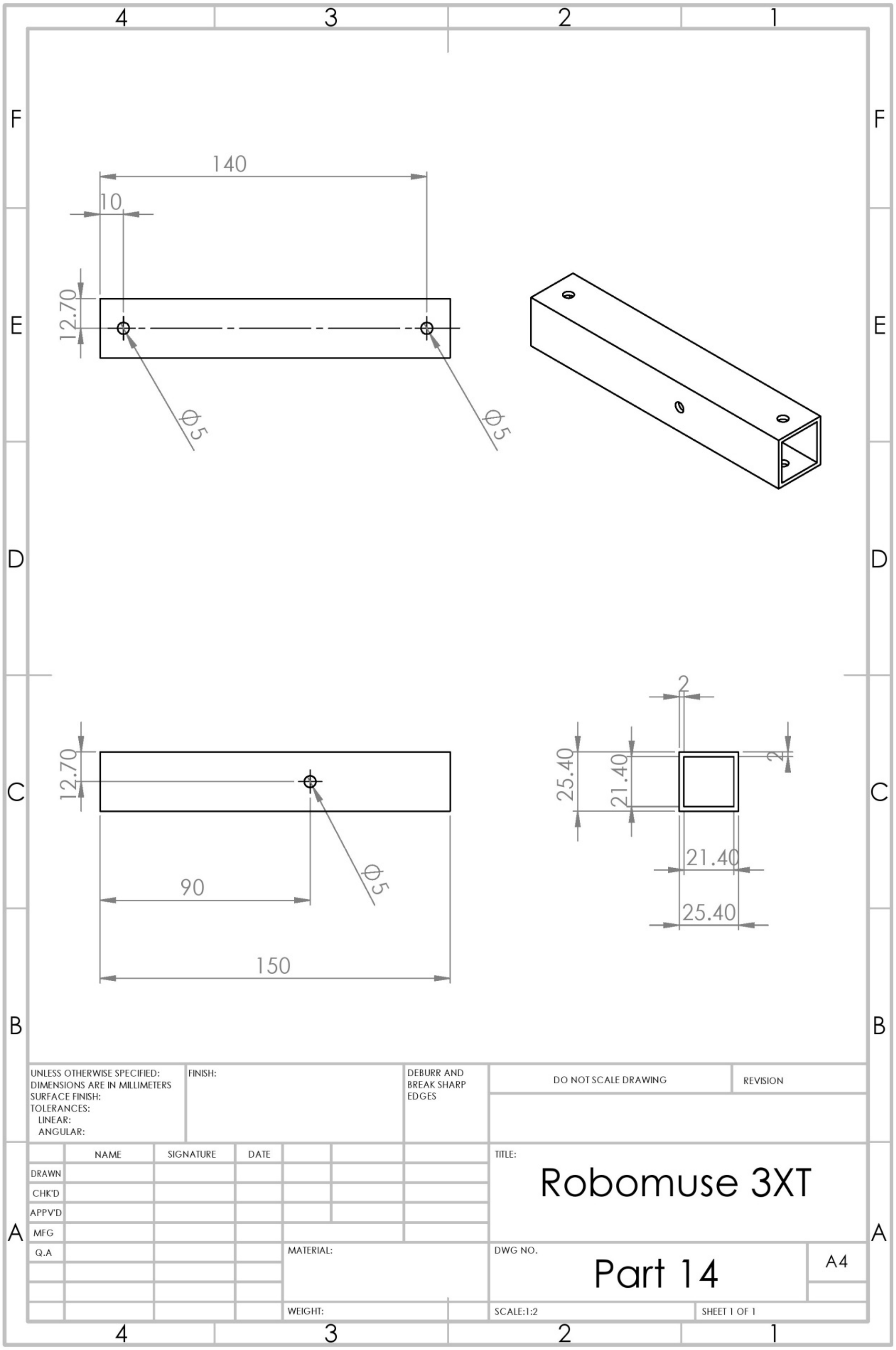
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:		DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING		REVISION
	NAME	SIGNATURE	DATE	TITLE: Robomuse 3XT		
DRAWN						
CHK'D						
APPV'D						
MFG						
Q.A			MATERIAL:	DWG NO. Part 10		A4
			WEIGHT:	SCALE:1:5	SHEET 1 OF 1	

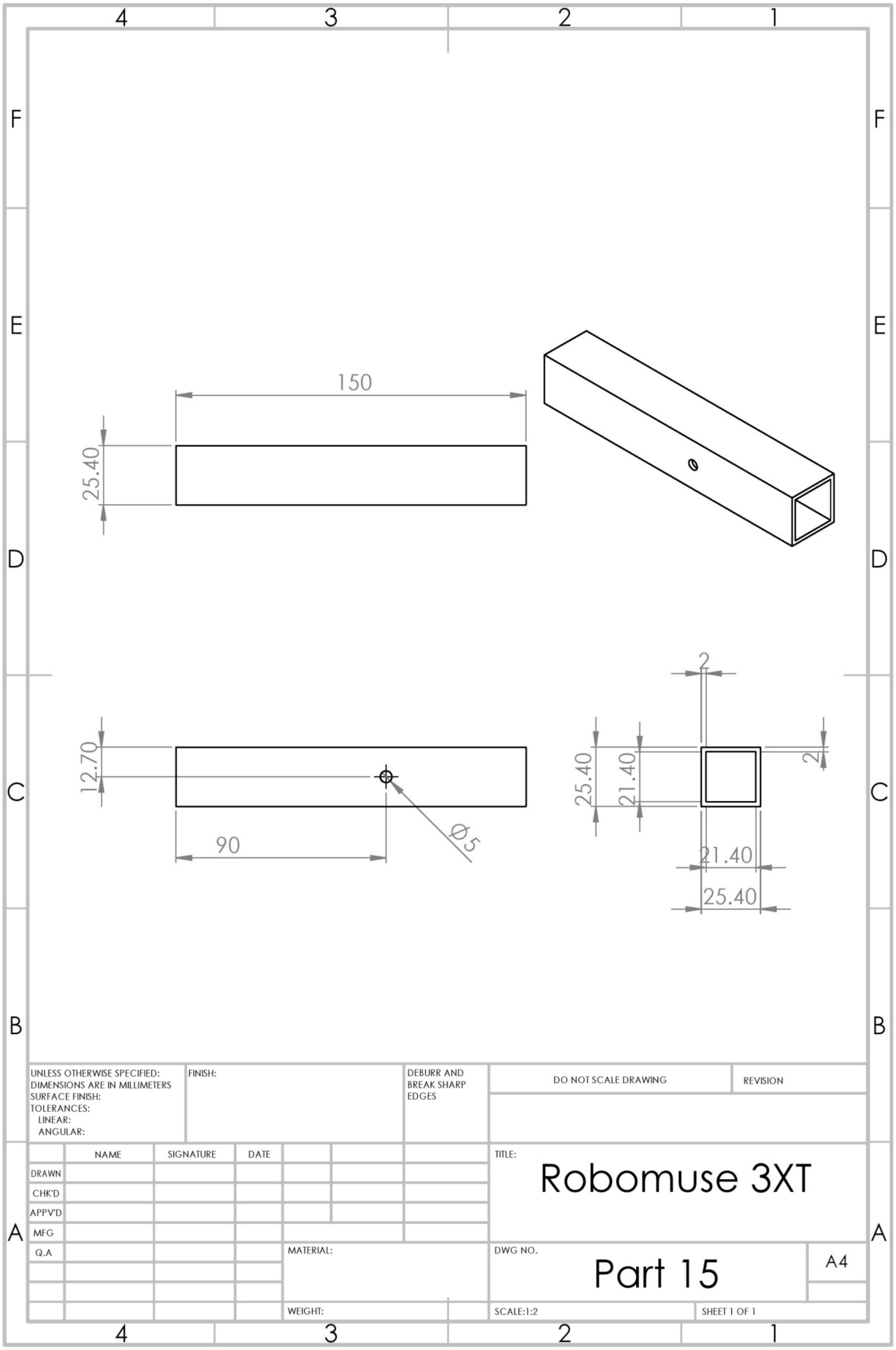


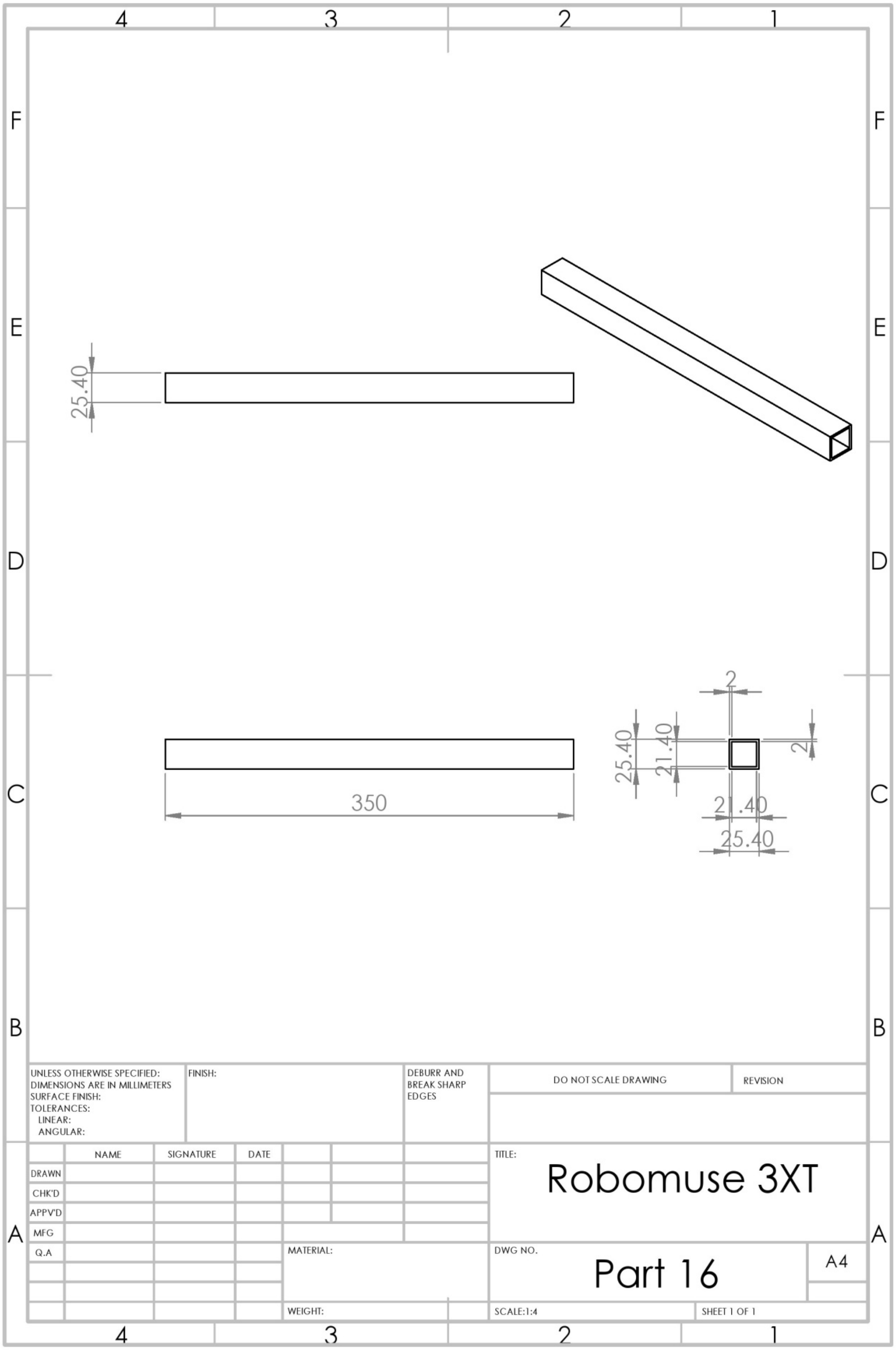
12.70
Ø5
Ø5
35.40
433.80
469.20
25.40
2
2
2
2
25.40
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
Robomuse 3XT
DWG NO.
Part 11
A4
SCALE:1:4
SHEET 1 OF 1

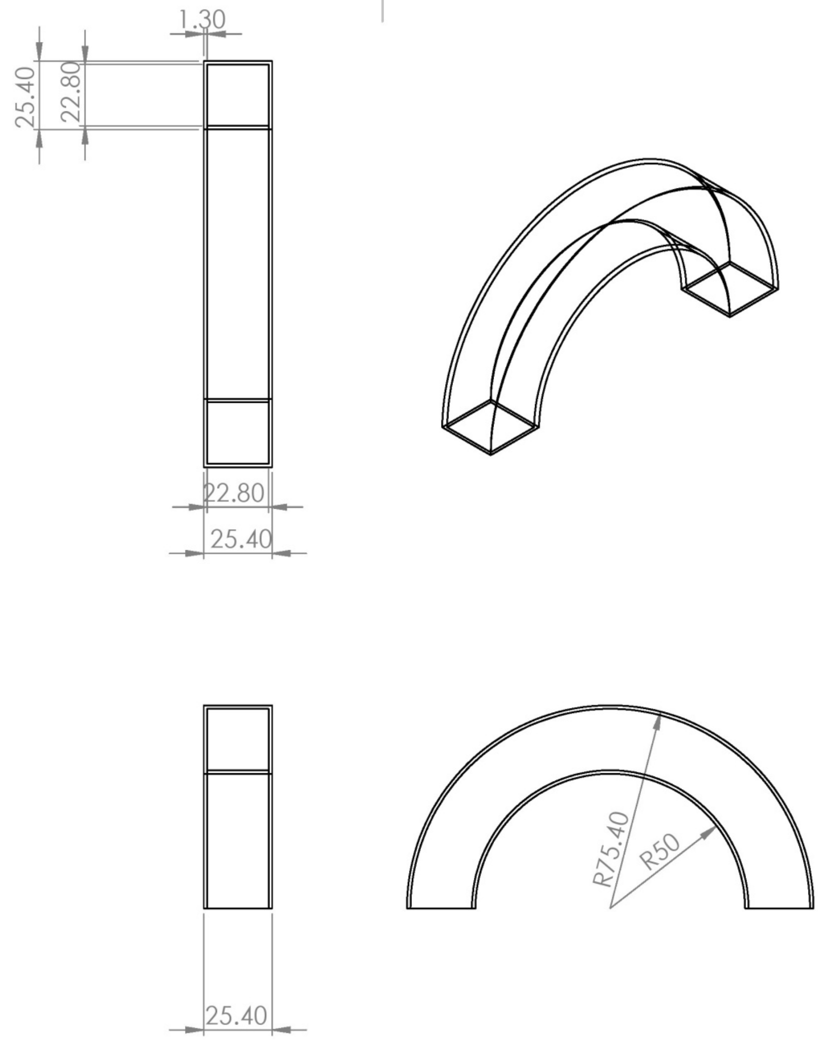

284.60
Ø5
Ø5
Ø5
12.70
87.30
87.30
569.20
25.40
2
25.40
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 12
A4
WEIGHT:
SCALE:1:10
SHEET 1 OF 1


184.60
25.40
25.40
2
2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 13
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1


140
10
12.70
Φ5
Φ5
12.70
Φ5
90
150
2
2
25.40
21.40
21.40
25.40
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 14
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1


150
25.40
12.70
90
Ø5
2
25.40
21.40
2
21.40
25.40
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 15
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1


25.40
350
2
25.40
21.40
2
21.40
25.40
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 16
A4
WEIGHT:
SCALE:1:4
SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION	

	NAME	SIGNATURE	DATE		
DRAWN					
CHK'D					
APPV'D					
MFG					
Q.A				MATERIAL:	
				WEIGHT:	

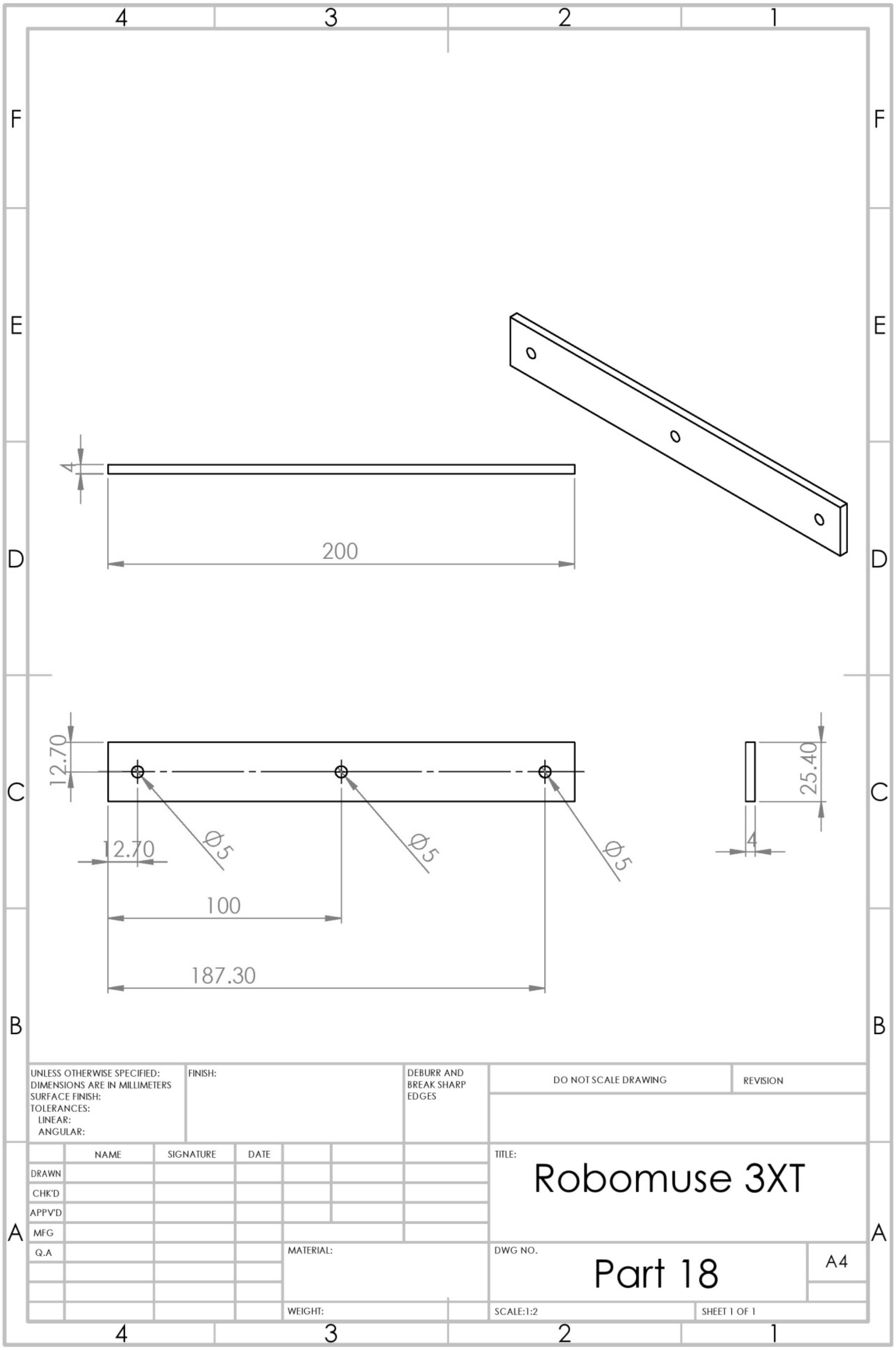
TITLE: Robomuse 3XT

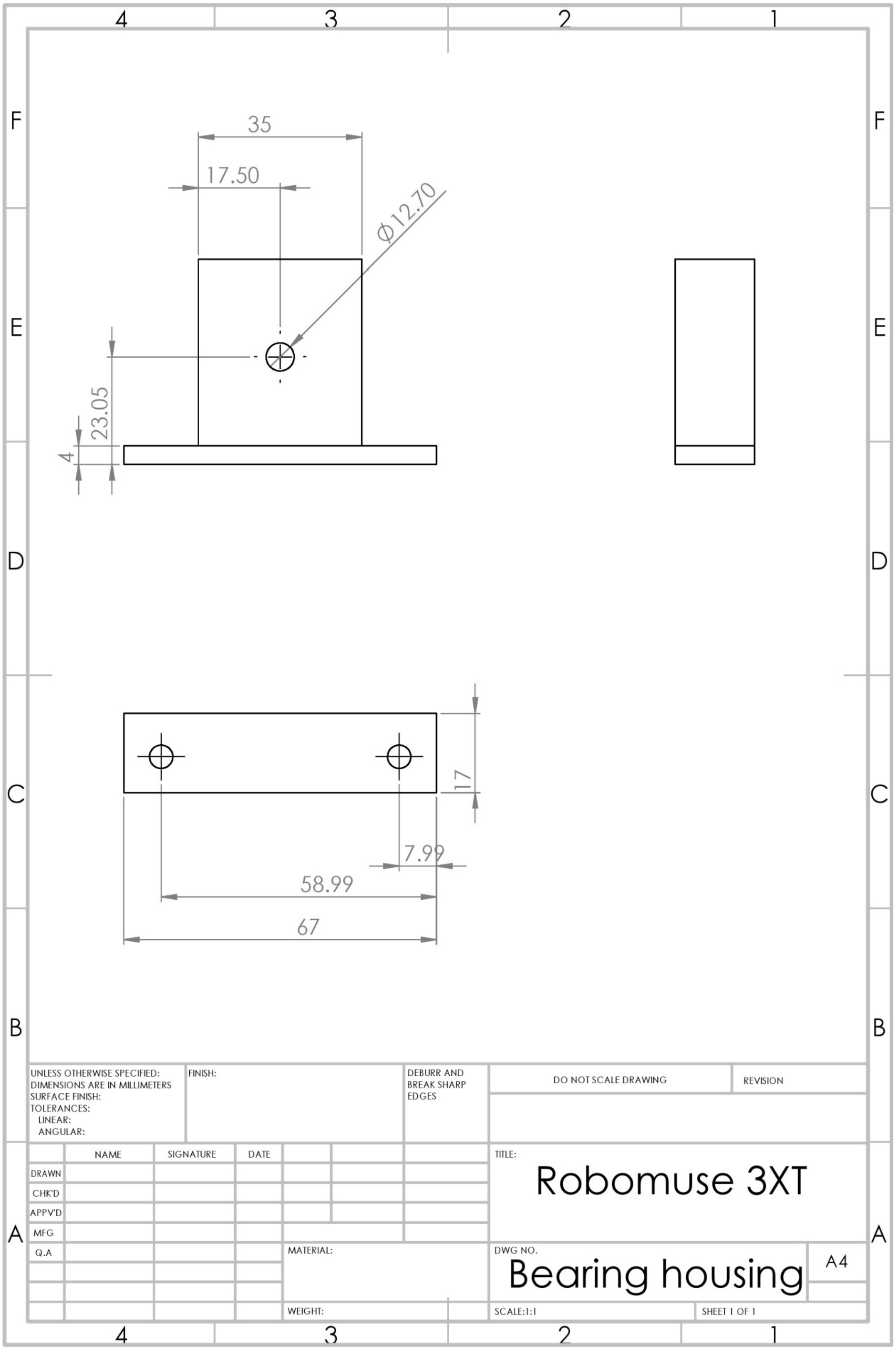
DWG NO. Part 17

A4

SCALE:1:2

SHEET 1 OF 1


4
200
12.70
25.40
Ø5
Ø5
Ø5
12.70
100
187.30
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Part 18
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1


35
17.50
Ø12.70
23.05
4
17
7.99
58.99
67
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
Robomuse 3XT
MATERIAL:
DWG NO.
Bearing housing
A4
WEIGHT:
SCALE:1:1
SHEET 1 OF 1