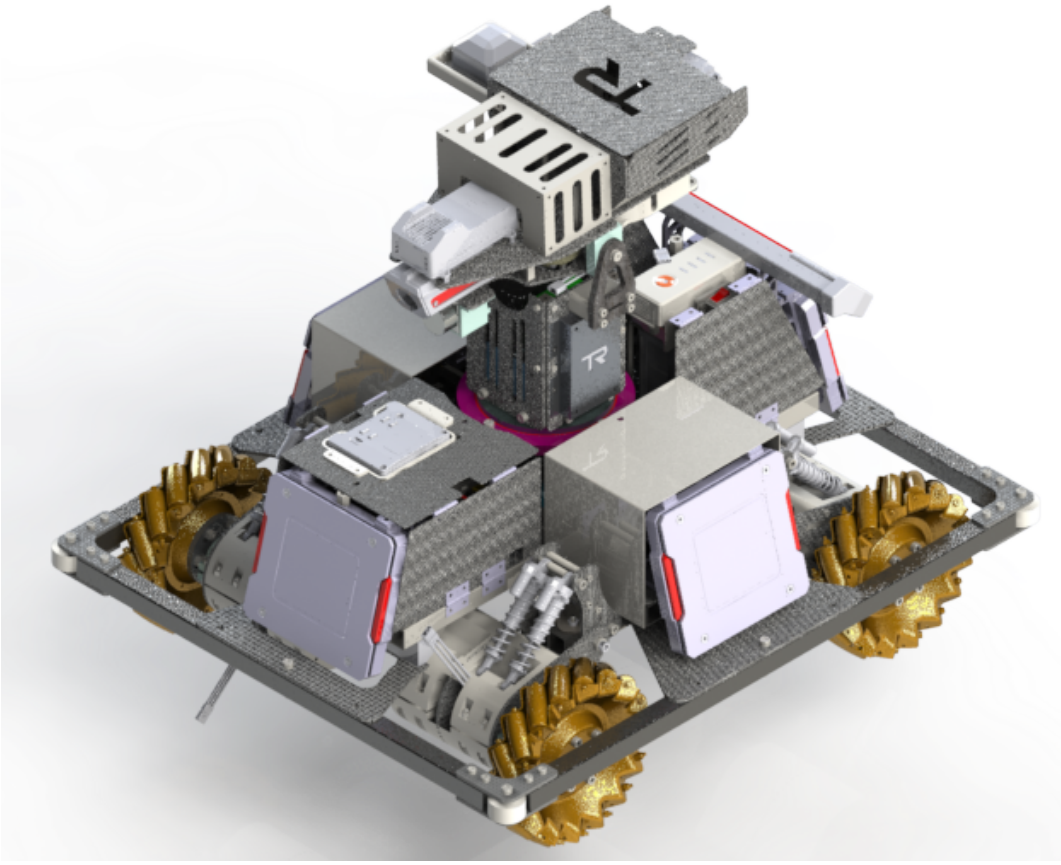
Infantry Technical Report

RoboMaster 2022

Written by Roger Nguyen
Contributors: Ankit ‘Ace’ Bhatia, Anshal Jain, Hongbin Miao, Michael Shao



Website:tritonrobotics.org
E-mail: tritonrobotics@ucsd.edu

Table of Contents

TRITON ROBOTICS

[1] Infantry Robot

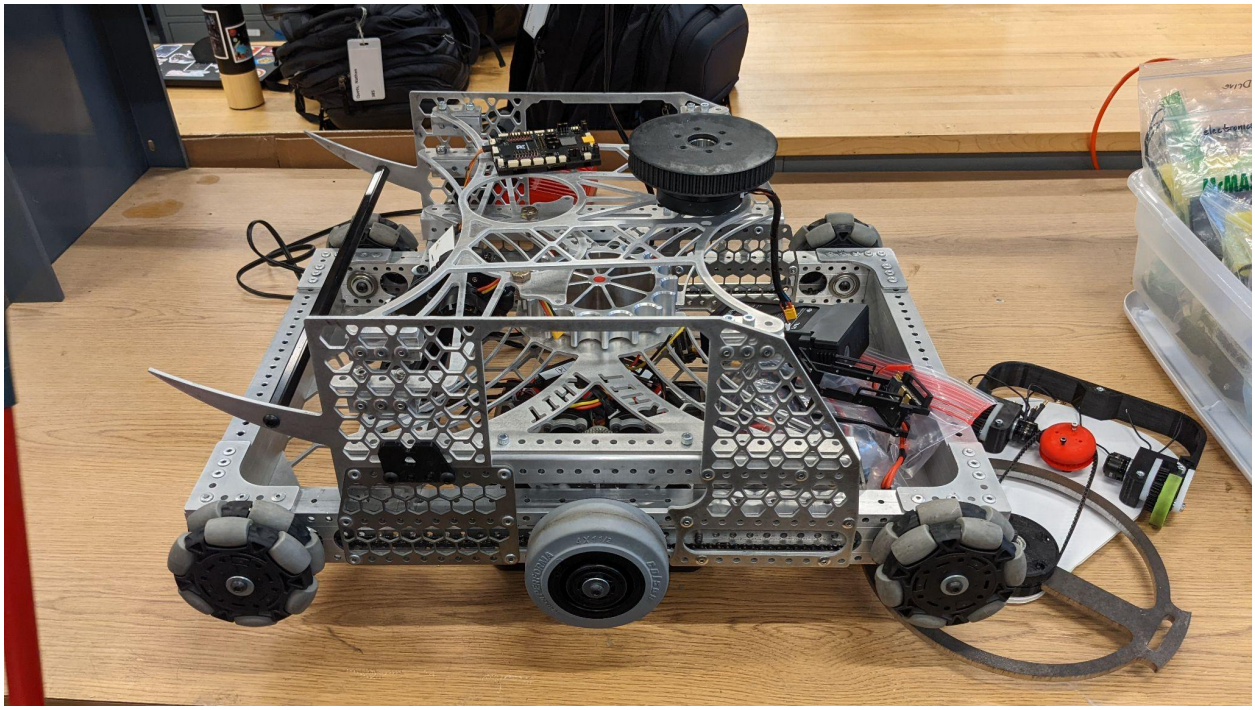
2021-2022 | University of California, San Diego

Website:tritonrobotics.org

E-mail: tritonrobotics@ucsd.edu

[1.1] Analysis of Infantry of Other Teams

[1.1.1] Rose-Hulman Institute of Technology (RHIT)'s Infantry



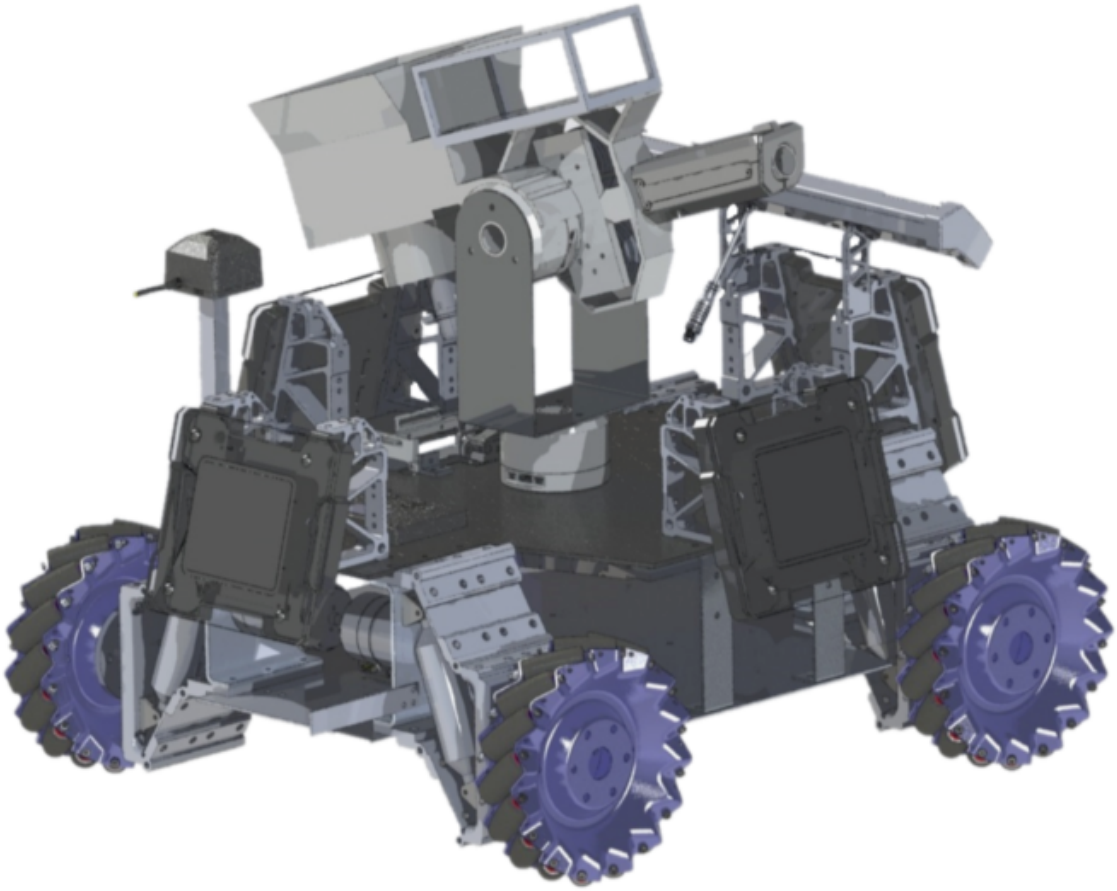
Based on this image of RHIT's infantry, this team decided to use omni-wheels on the four corners, as well as, two additional wheels on the sides and a center wheel rather than using standard mecanum wheels. Compared to other infantries, this wheel configuration is not standard. The advantage of this design compared to the standard 4-mecanum wheels is the reduced weight since the mecanum wheels are 660 grams each. The disadvantage of this design is an increase in the number of components used to achieve what the mecanum wheels can do and as a result, this increases the complexity.

RHIT uses square tubing for their outer frame and sheet metal for the core part of their robot. UC San Diego employs a similar tactic with their infantry but opts to use carbon fiber plates rather than sheet metal to reduce the weight of the robot. RHIT decided to use sheet metal but also reduce the weight of each plate by cutting out patterns. The main form of RHIT's robot is box-like. UC San Diego's infantry used to employ this form, but the difficulty of access to electronics made it difficult to continue on this path. This robot is not currently complete.

[1.1.2] University of Washington (UW)'s Infantry

Based on this image of UW's infantry, it has a similar form to UC San Diego's infantry. Both utilize mecanum wheels with a suspension system parallel to the wheels. The protective frame of the robot utilizes square tubing. There's a protective shield for electronics on the chassis to protect the robots. The main differences that can be viewed are seen in the turret base and turret design. UC San Diego decided to use a rectangular turret base and put the pitch GM6020 motor along with the Jetson inside. UW decided to use a trapezoidal turret base and left it empty. As a result, the pitch motor directly drives the turret rather than a linkage mechanism that UC San Diego employs. The Jetson is also very close to the turret. The disadvantage of this is a larger moment of inertia that was taken into account when UC San Diego decided to house these hardware components closer to the robot's center of mass. More information about this robot could not be found.

[1.1.3] Texas A&M University (TAMU)'s Infantry



TAMU's robot utilizes elements that are very similar to UC San Diego's previous year's robot. The suspension system is double wishbone-esque. The reason UC San Diego decided not to use this design was due to the fact that this type of suspension caused too much wobbling when it moved forwards and backwards. This wobble was apparent on the hero robot and less so for the infantry robot; however, in order to standardize suspension systems, the infantry robot adopted a new design. The core frame of this robot uses plates which are good for housing electronics and a simple design that works well with the suspension system. UC San Diego decided to move away from this design in order to allow for easier access to electronics. In a similar regard, the pitch GM6020 motor directly drives the turret; the location increases the moment of inertia of the turret. This may be negligible if the yaw GM6020 motor has enough torque but less power is needed to generate the same angular velocity. A new shooting design is vertical flywheels. UC San Diego still uses the standard horizontal flywheel design and has decided to stick with the standard design over the avant-garde one.

[1.2] Function

Chassis

Complete XY planar movement
Capable of climbing a 16 degree sloped ramp
Capable of jumping off a ramp
Beyblade mode with slip-ring
Stable on bumpy surfaces
Modular suspension system
Suspension system with shock absorbers parallel to front and back

Gimbal

Three axis gimbal with good shooting stability

Shooting System

Auto aiming/Assistant aiming with Computer Vision

Other Functions

Easy access of Electronics
Stable under violent crash
Industry Design/Nice Looking

[1.3] Parameters

Mechanical Parameters:

Weight: 11.3 kg
Gravity Center: 100 mm above ground level
*Dimensions (L*W*H):* 599.72mm x 513.72mm x 492.8mm (LWH).

Electronics Parameters:

Sensor(s): RFID (up to 70 mm reading distance)
Power Consumption of Circuits: 6 Watts
Total Capacitance of Capacitors: 20 Farads
Working Voltage Range: 3.3-24 Volts
Actuators: 2 GM6020, 4 M3508 P19, 1 M2006 P36, and Snail 2305 Brushless DC Motors

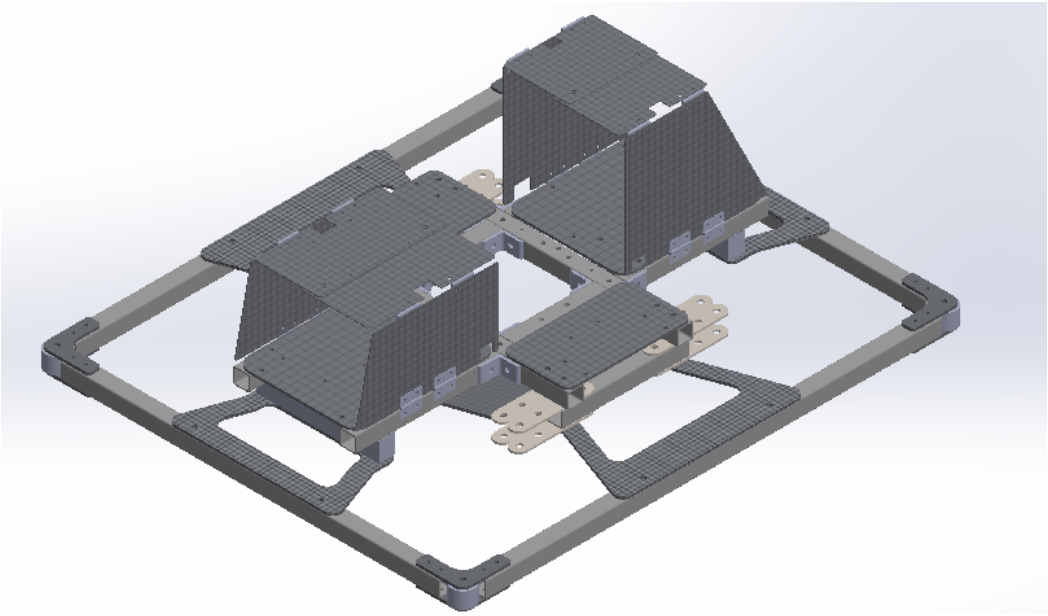
Performance Parameters:

Maximum Moving Speed: 8 meters per second
Climbing Angle: 15 degrees
Rotatable Angle of Gimbal: +/- 30 degrees
Shooting Rate: 10 balls over 4 seconds
Hitting Rate (# out of #): 7 out of 10 balls

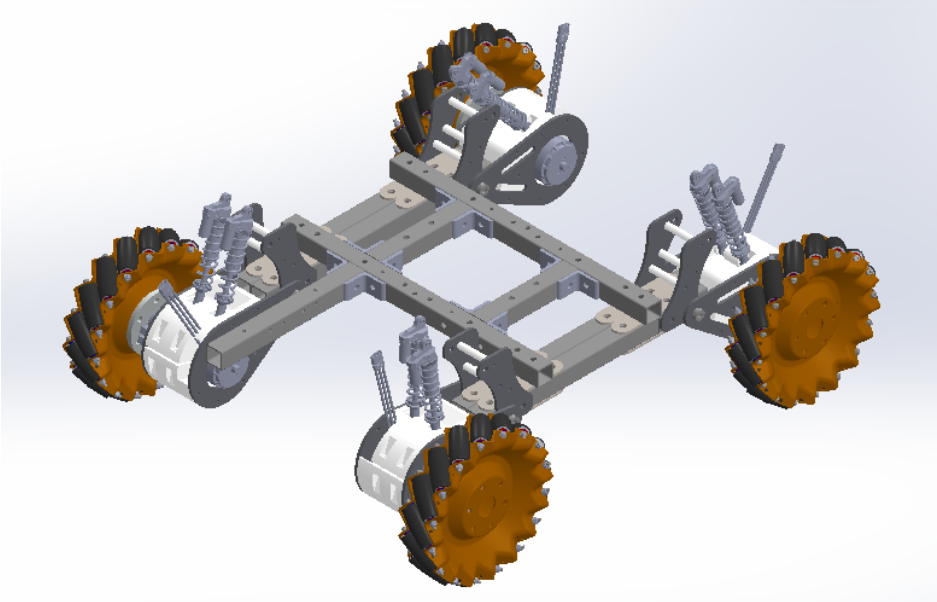
[1.4] Design

[1.4.1] Mechanical Structure Design

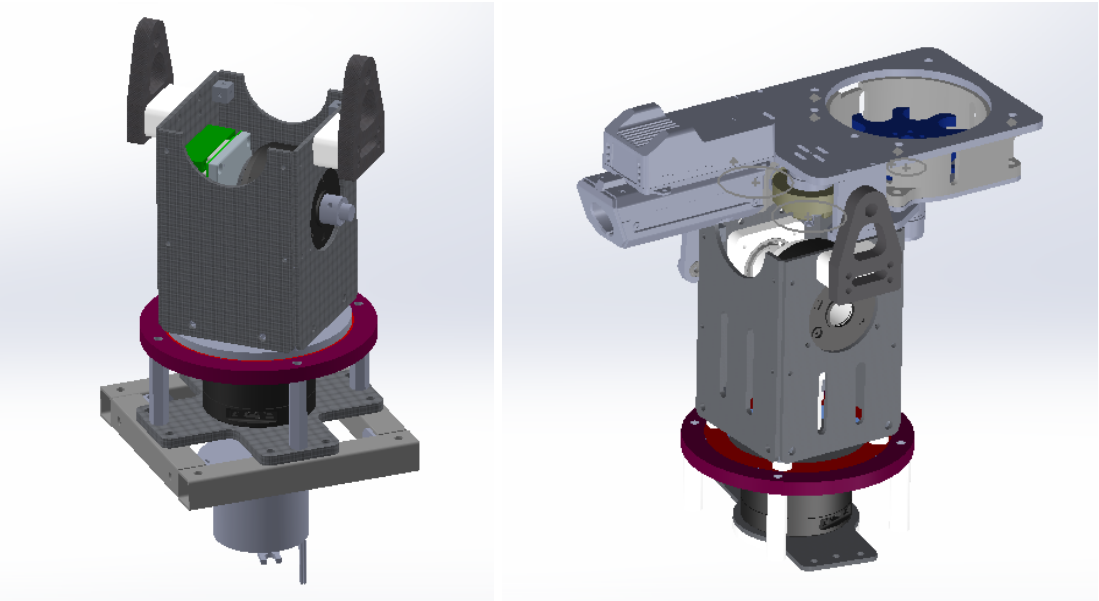
[1.4.1.1] Design Overview



The 2021-2022 chassis frame utilizes square tubing and carbon fiber/acrylic plates. Using L-brackets, the square tubing forms the foundation and the frame of the chassis, while the plates are used to mount various electronics. In order to protect the electronics from projectiles, an outer casing was made consisting of plates and a hinge design. In a similar regard to the chassis frame, the protective frame enclosing the chassis is also made of square tubing and plates with rollers at the end to prevent damage to the armor modules during collisions.



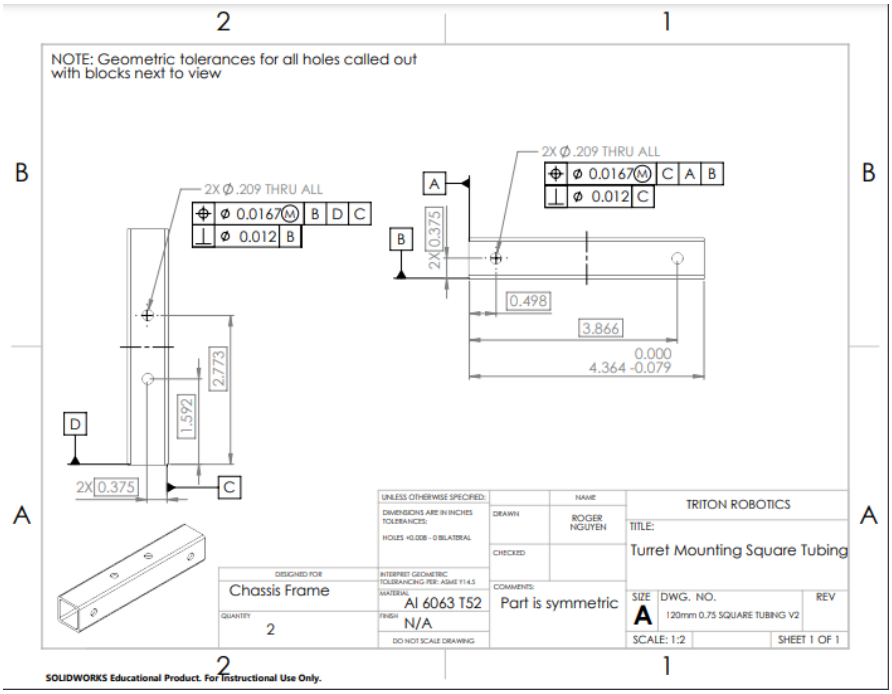
The suspension system uses a design where the shock absorbers are parallel to the mecanum wheels. In order to prevent tilting due to the weight of the chassis on the suspension assembly, a cover with a clamping design is used in hopes that it would distribute the load more evenly across the suspension arm. Compared to previous years, the suspension system is modular.



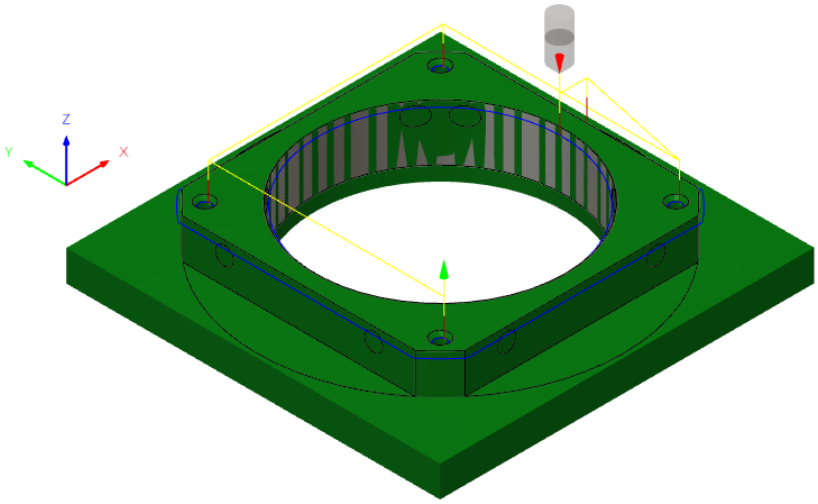
The image on the left is the second design of the turret that is a work in progress, while the image on the right is the first design of the turret excluding components like the ammo box and proper mounting for the DEV board. This turret design will incorporate the Jetson for auto-aiming purposes and a new larger slip-ring that was deemed necessary due to the added electrical components compared to last year. The new turret design uses a 4-bar linkage mechanism to pitch the turret rather than using direct drive. This new design was used in order to make the turret base more

compact by moving the pitch motor inside the turret base, as well as reducing the moment of inertia about the yaw axis as a result.

[1.4.1.2] Manufacturing Overview



The square tubing was cut on a horizontal bandsaw and put onto a manual milling machine to drill the holes. The image above is a drawing that was used to manufacture one of square tubing parts.



Some parts were machined on the CNC milling machine out of aluminum 6061 stock. The image above is a picture of the program used to create one of these machined parts.

Some parts were made of sheet metal. These parts were first cut with a waterjet machine and bent.

Other manufacturing techniques that were used to create the Infantry robot was FDM 3D printing to create unique parts and laser cutting to cut carbon fiber plates. The laser cutting was outsourced to Vinatech Engineering Inc, a manufacturing company not affiliated with UC San Diego.

[1.4.1.3] FEA Analysis

FEA analysis was done on the suspension system and protective frame to ensure that they would not break during competition settings.

Expected Weight of Infantry in RMUL 2021	20 kg
Height of Drop Test	0.2 m

Drag force will be ignored for this calculation.

The potential energy of the Infantry robot at the max height of the drop test is 39.2 Joules, using the equation $P = mgh$.

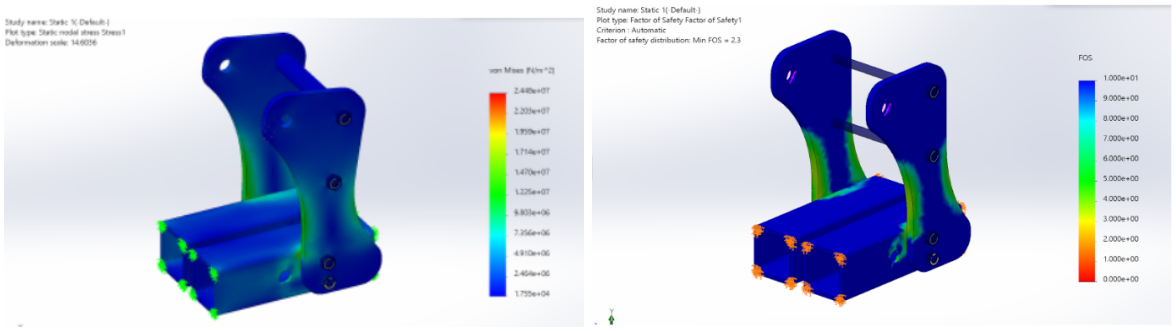
Assuming that all this potential energy gets directly converted to kinetic energy, the speed of the robot at the ground is 1.98 meters per second, using the conservation of energy and the equation $mgh = \frac{1}{2}mv^2$.

If the robot is going at 1.98 meters per second and collides into the ground at a complete stop, the magnitude of the change in momentum or impulse is 39.6 Newton-seconds using the equation $J = \Delta p = |0 - mv_0|$.

The force of the collision can be found by assuming a duration of 0.1 seconds and constant force using $J = F\Delta t$. This gives us a force of 396 Newtons.

Since there are eight shock support brackets, the total force will be distributed across all eight brackets, leaving each bracket to face 49.5 Newtons of force.

Calculations were done to determine the boundary conditions for the drop test and survivability of the suspension system.



A static simulation was completed: the part deforms as expected and it doesn't fail. The factor of safety was above one but fatigue study was not accounted for.

Maximum Weight of Infantry in RMUL 2021	25 kg
Maximum Chassis Power Consumption at Level 3 for Power Focus	100 W

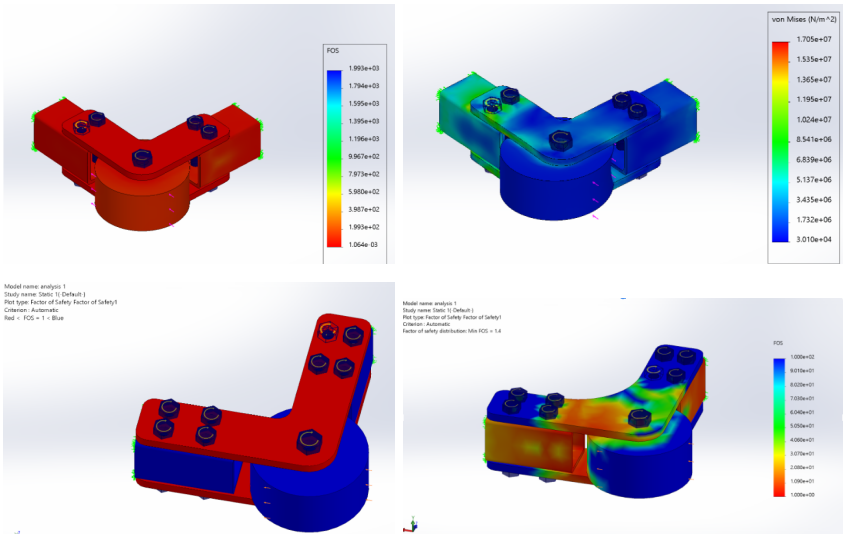
Assuming that all the power goes into the chassis in one second, the energy provided is 100 Joules using $P = E/t$.

Provided that all 100 Joules goes into translational movement, this means that the speed of the infantry, using the equation $E = \frac{1}{2}mv^2$, is 2.83 meters per second.

If the robot is going at 2.83 meters per second and collides into the playing field at a complete stop, the magnitude of the change in momentum or impulse is 70.71 Newton-seconds using the equation $J = \Delta p = |0 - mv_o|$.

The force of the collision can be found by assuming a duration of 0.1 seconds and constant force using $J = F\Delta t$. This gives us a force of 707 Newtons.

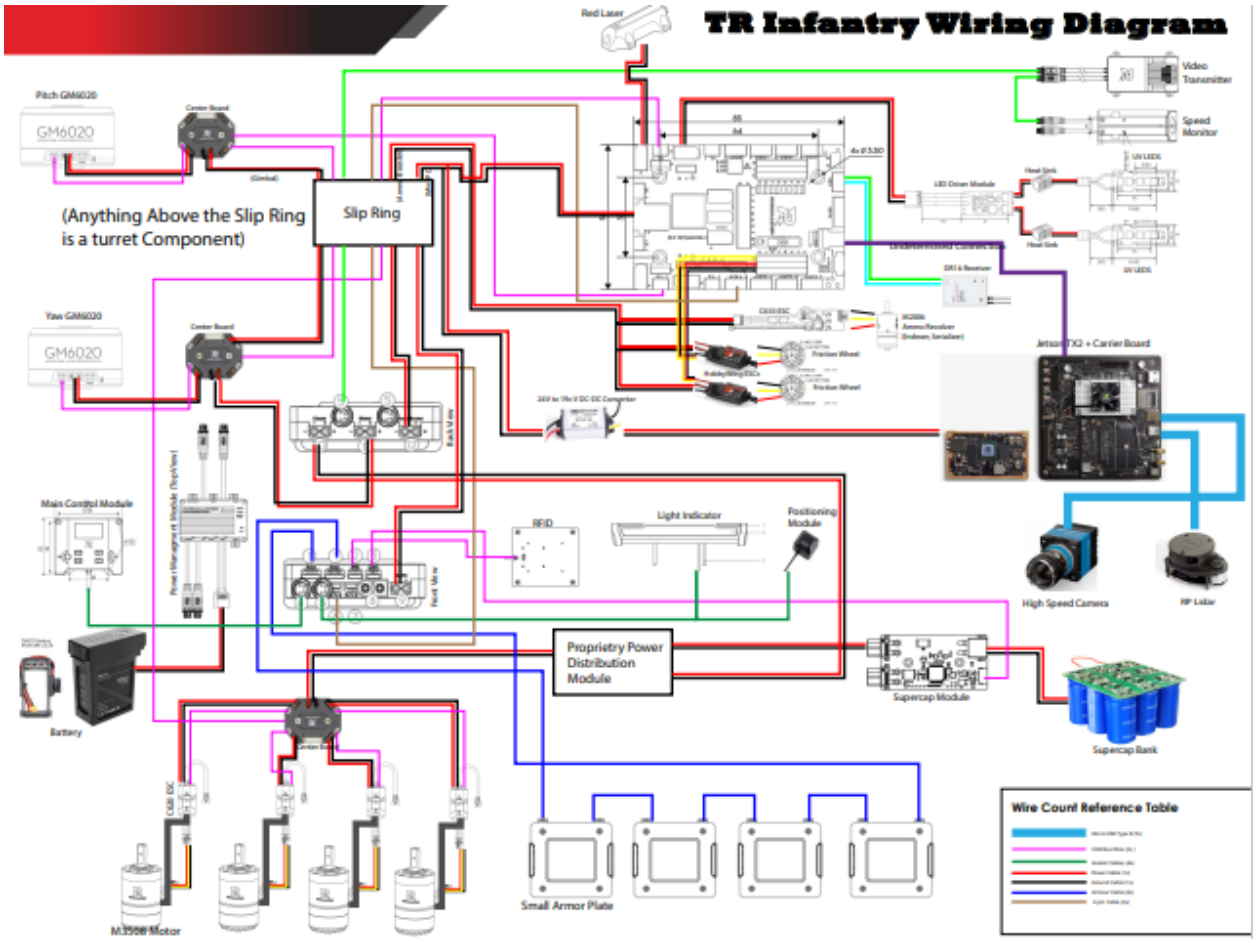
Calculations were done to determine the boundary conditions for an impact collision on the protective frame.



The part deforms as expected but ended up failing. Design changes were made to ensure this part would not fail during a direct collision in the worst case scenario. Future iterations were made to maintain the same integrity while reducing the number of fasteners needed.

[1.4.2] Hardware Design

[1.4.2.1] Wiring Diagram



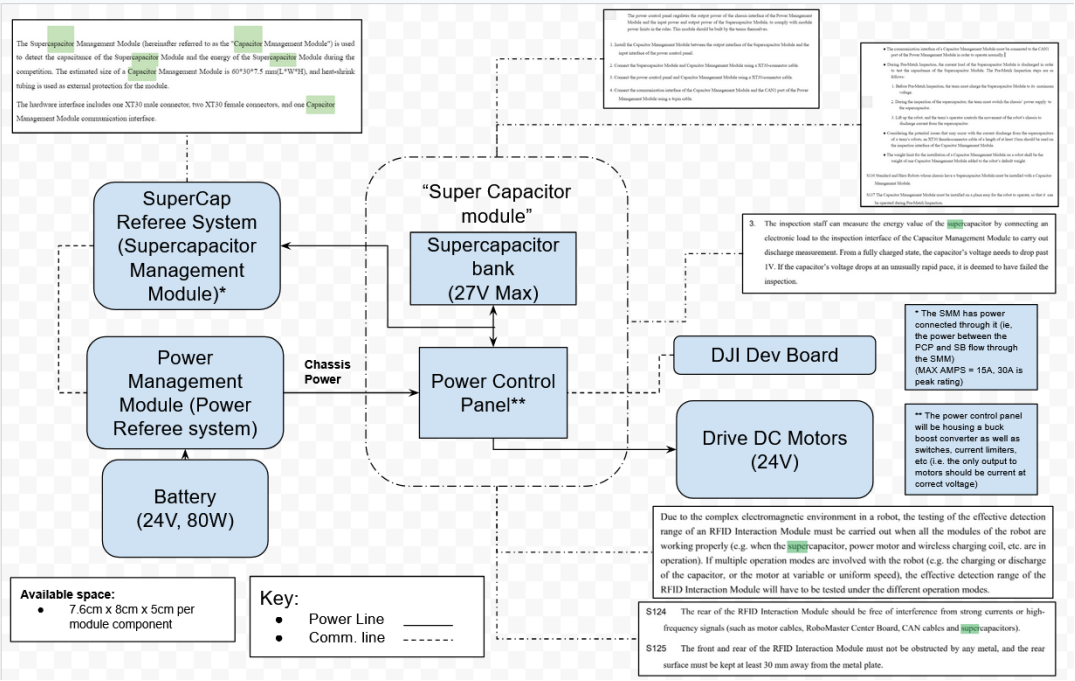
1. Development Board

We run everything off the PMM, and wire our M3508s and GM6020s with CAN. For our

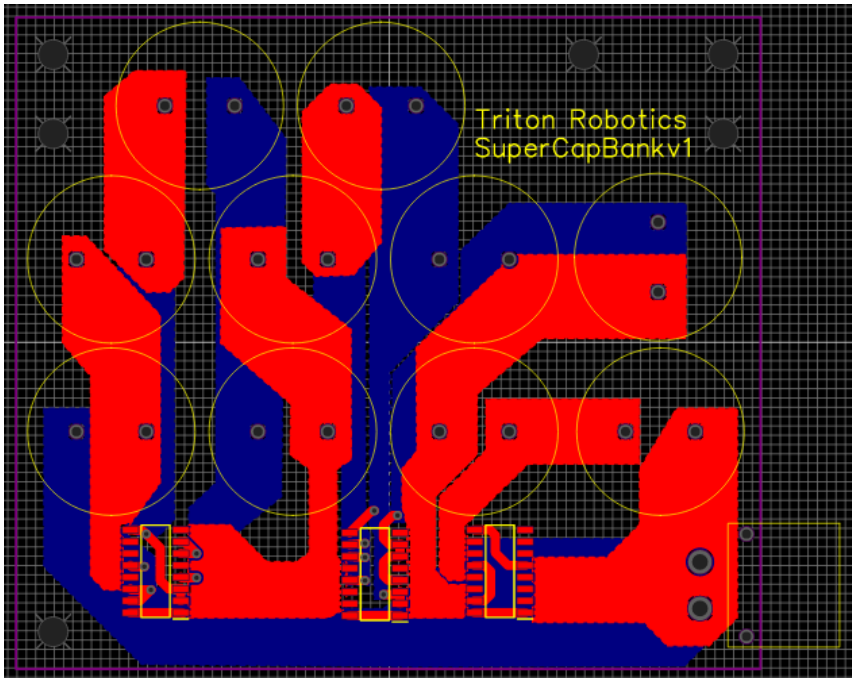
We use a Nucleo F446RE for our control systems, and our robot is run with M3508s, GM6020s, and snails.

2. SuperCapacitor System

*Wiring diagram denoted with communication port and the power flow denoted by arrows

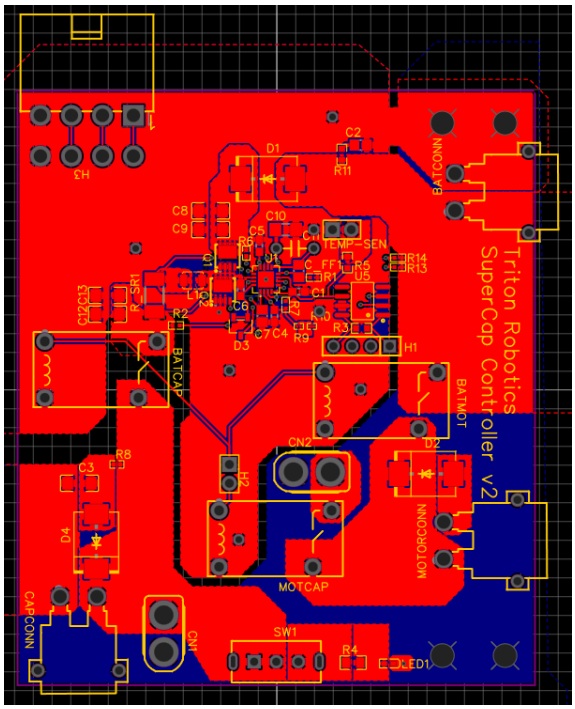


The Supercapacitor system allows the robot to store unused energy from our power supply budget (80W) - specifically power not going to the drive motors. On demand, the system can then unleash said energy to allow the robot to briefly consume more power than it is normally allowed to from the battery. For example, in the competition, this bank, in combination with the control board, can allow a robot, like Infantry, to move faster briefly. As can be seen above, the system sits between the Power Management Module and the drive motors. There are two modules - the supercapacitor bank and the control board.



The PCB file to the left shows the supercapacitor bank with 10 capacitors in series, as well as 3 ALD811024SCL ICs in between, which help evenly distribute the charge between the capacitors. Note that each individual capacitor has a capacitance of 60 F, so the equivalence of this super capacitor bank is 6F.

The PCB file to the right shows the super capacitor control board.The main feature of the board is a BQ24640VAR IC, which, using the surrounding passive components, manages the charging of the supercapacitor bank via the battery. We can control to an extent the charge rate via a digital potentiometer, as shown to the left, which itself is controlled via an Arduino which communicates using SPI. Also important features are the relays that we use to switch between connecting the battery to the motor and supercapacitor, and the supercapacitor to the battery, in order to control when we are charging/discharging our supercapacitors. Additional features on the board are an inrush current limiter and discharging circuit for safety.



The control board interfaces with an Arduino UNO to compartmentalize the low level operation of the control board and leave a high level interface for communication with the main Nucleo Board.

[1.4.3] Software Design

1. System Structure

Mini-Computer: Jetson TX2
Operating System: Ubuntu 18.04
Software System / Libraries: ROS2 Foxy, OpenCV2, micro-ROS-Agent
Programming Language: C++

Dev-Board: Nucleo F446RE, STM32F427II
Software System / Libraries: Mbed 6, Micro-ROS, STM32 HAL
Programming Language: C++

Supercapacitor module: Arduino UNO
Software System / Libraries: N/A
Programming Language: C++

2. Execution process

First, the streams of image will be fed into the program by the industrial camera (with pre-tuned camera parameters: exposure, gain, height, width …) . Next, we will do image pre-processing on the input (more specifically, tuning the color threshold and binary threshold of a gray image in order to select out the target color and target brightness). After the pre-process, we will have a binary image suitable for feature extraction for finding the contour of the armor module. After finding all the possible contours, we select the contour with the optimal matches by defining a set of matching constraints, such as height-width ratio, angle offset between the two led bars on the armor module. Once the optimal matches have been found, we find the optimal armor by the armor that has the closest euclidean distance to the center of the frame.

3. Core Function

Each robot runs a dual microcontroller system, outfitted with two STM32 boards from the F4 family.

STM32F427II (DJI Dev Board type A):

Inputs:

- DR16 Receiver

Outputs:

- Serial message containing the parsed information of the DR16 receiver.

Protocol:

- Followed from the DR16 manual, a series of bit shifts and/or masks in order to parse four joystick channels, switch and wheel data, and mouse/keyboard inputs.

STM32F446RE (Nucleo F446RE):

Inputs:

- Serial input from DJI Dev Board A
- 2x CAN input using the Waveshare CAN Transceiver (SN65HVD230)
- Note: one transceiver for each CAN bus
- Serial input from Referee System User Port
- Serial input from onboard mini-pc (Jetson TX2)

Outputs:

- CAN protocol messages to the CAN network
- Serial data to onboard mini-pc (Jetson TX2)

Protocol:

- Receive data from DJI Dev Board A, use parsed data from DR16 as input for remote tele-operation
- Receive and send CAN messages to CAN Transceiver, which in turn communicates with all connected motor controllers of type C620, C610, and GM6020

- Receive and send Serial/UART information from Referee System User Port, allowing for robot functionality to ensure robot follows all referee specifications
- Receive and send Serial/UART information from mini-pc, allowing for armor module tracking and auto lock on via computer vision

Nucleo F446RE: Motor Protocol:

All motor control is done by sending and receiving CAN messages to and from CAN Transceivers (SN65HVD230).

Inputs:

- Aggregated motor data, including torque, position, velocity, and temperature

Outputs:

- Raw current value to send via CAN in order for each motor to reach desired target values

Protocol:

- Define three modes for motor control: Speed, Current, and Position
- Run PID algorithms with various constants depending on the motor type (M3508, M2006, GM6020).
- Error will be calculated by subtracting aggregated data from desired input.
- PID constants are determined using the Ziegler-Nichols algorithm.
- Track absolute encoder position by storing data upon encoder position reset.

Computer Vision:

Inputs:

- Current team color (red, blue)
- Aiming mode (armor module, center star)

Outputs:

- Desired turret pitch and yaw

Protocol:

- The core function of the task in computer vision is to develop a stable autonomous aiming system that can help the robot driver to target at the opponent's armor plate in a fast and precise manner in the competition. The aiming system will be running on Jetson TX2, a compact and powerful minicomputer.

4. Software Test

Embedded:

Testing is done on the STM32F446RE via a logging console from the host computer. The Mbed framework creates a live console environment which host computers can access via a TTY. Generally, our host computers communicate with microcontrollers from the USB port

(/dev/ttyACM0). Mbed offers a live debugging environment using pyOCD, which allows testers to run GDB on generated hex files.

Computer Vision:

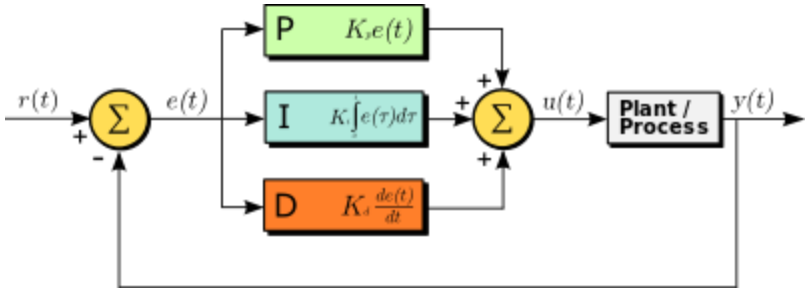
In Computer Vision, we will be testing our algorithm implementation by first observing the detection result via the OpenCV2 GUI on the monitor. More specifically, we point the camera toward the armor with the desired color that we wanted to detect and track and view the result on the monitor. We test the algorithm by performing a similar test in different environments: different lighting conditions, distance and angle between the camera and armor, camera acceleration in different directions to simulate competition environment … Below is a clip of the Computer Vision testing:

https://youtube.com/shorts/yWOVYmLYiRQ

[1.4.4] Algorithm Design

Embedded: Motor Movement Algorithms

1. PID



The PID algorithm is based on three values: Desired value, Actual value, and time difference (dt). Error e(t) is calculated by subtracting the actual value from the desired value. K_p, K_I, and K_D are non-negative coefficients for the Proportional, Integral and Derivative terms. Final output u(t) can be found using the equation as shown:

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(\tau)\,\mathrm{d}\tau + K_\mathrm{d} \frac{\mathrm{d}e(t)}{\mathrm{d}t},$$

PID constants K_p, K_I, and K_D are found using the Ziegler-Nichols tuning method. The K_P term is increased from zero until reaching the ultimate gain K_U, which is the value at which the PID loop outputs a stable and consistent oscillation. Given K_U and the oscillation period T_U, a number of simple mathematical operations are performed to reach the final constants for the desired controller. These operations can be found using the table below:

Ziegler–Nichols method[1]

Control Type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	–	–	–	–
PI	$0.45K_u$	$0.80T_u$	–	$0.54K_u/T_u$	–
PD	$0.8K_u$	–	$0.125T_u$	–	$0.10K_uT_u$
classic PID[2]	$0.6K_u$	$0.5T_u$	$0.125T_u$	$1.2K_u/T_u$	$0.075K_uT_u$
Pessen Integral Rule[2]	$0.7K_u$	$0.4T_u$	$0.15T_u$	$1.75K_u/T_u$	$0.105K_uT_u$
some overshoot[2]	$0.3\bar{3}K_u$	$0.50T_u$	$0.3\bar{3}T_u$	$0.66\bar{6}K_u/T_u$	$0.1\bar{1}K_uT_u$
no overshoot[2]	$0.20K_u$	$0.50T_u$	$0.3\bar{3}T_u$	$0.40K_u/T_u$	$0.066\bar{6}K_uT_u$

2. Matrix Rotations

A matrix rotation is performed for the purpose of creating translation motion while the robot chassis continually rotates. Each of four wheels is given a location in the matrix, corresponding to the physical location of the motor on the robot. Then, using the desired angle *theta*, the following transformation is applied:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

These four points are rotated counterclockwise using the angle *theta* with respect to the positive x-axis. Following this transformation, accurate motor outputs are derived.

[1.5] Iteration

[1.5.1] Iteration Record

Version/Phase	Performance Illustration	Comp
V1.0	Newly built robot is capable of movement but not shooting. Shooting system has not yet been built.	2022.11
V2	Fixes most of the chassis related problems and created the turret. Jamming occurs frequently.	2022.3
V2.1	Fixes turret jamming and other minute problems	2022.4
V2.2	Robot is mechanically finished	2022.5

[1.5.2] Key Problem Solving Record

No.	Description of the Problem	Cause of the Problem	Solution Proposed and Result	Robot Version	Credited Members
1	Suspension system bends inward	Long moment arm and uneven load distribution on suspension system	Reduce moment arm and add a cover to distribute load more evenly	V1.0	Mechanical Member: Kenny Wang

No.	Description of the Problem	Cause of the Problem	Solution Proposed and Result	Robot Version	Credited Members
2	Shooting system jams	Balls are unable to make it to barrel	Adding a cover to reduce “double-stacking” and changing revolver design	v2.1	Mechanical Member: Roger Nguyen
3	The charge on the supercapacitors in series may not necessarily balance , which can affect performance.	Lack of any mechanism to balance charge.	The use of ALD8100xx MOSFETs was implemented.	V2.1 (Supercapacitor Bank v1.0)	Hardware Member: Michael
4	The XT-30 port on the supercapacitor bank was accidentally reversed.	Design error not caught until after production.	Reverse the XT-30 port on the supercapacitor control board.	V2.1 (Supercapacitor Control Board v1.0)	Hardware Member: Guang

[1.6] Team Member Contribution

Name	Basic Info (Major, Grade, Position)	Job Description	Contribution (Total Contribution is 100%)
Roger	Mechanical Engineering, Sophomore, New Infantry Lead	In charge of the mechanical development of the robot after Spring 2022.	25%
Kenny	Mechanical Engineering, Senior, Old Infantry Lead	In charge of the mechanical development of the robot, including chassis, suspension, turret base, and turret up until Spring 2022	30%
Ankit 'Ace'	Computer Engineering, Sophomore, Software Lead	In charge of the embedded development of the whole robot, including chassis control, gimbal control, computer vision embedded environment development, etc.	20%
Hongbin	Math-Computer Science, Sophomore, Computer Vision Lead	In charge of developing computer vision using ROS for the infantry robot.	15%
Michael	Computer Engineering, Junior, Hardware Lead	In charge of development of the supercapacitor system, including the supercapacitor bank and control board.	10%

[1.7] Reference

[1] RoboMaster 2022 University Championship Rules Manual

[2] RoboMaster 2022 University Championship Robot Building Specifications Manual

[3] RoboMaster 2022 University League (North America) Rules Manual