Table of Content

1 Project Description

Robotics is a branch of engineering that focuses on designing, manufacturing, and operation of robots. They are employed in various industries, including manufacturing, healthcare, and transportation. The hardware and software of the robot are entirely linked. The robot's hardware supplies its physical structure and capabilities, while the software governs its activities. Actuators, sensors, control systems, and the power source comprise the hardware components. Operating system, control, planning, and user interface software are some examples of robot software components.

Actuators are the components that move robot parts. They could be motors, hydraulic cylinders, or something else entirely. Sensors such as cameras, proximity sensors, and force sensors receive information about the robot's environment. The control system evaluates sensor data and transmits commands to actuators. To power the robot components, a power supply must be provided.

The program's requirements must be met by the hardware, and the hardware must be properly controlled by the software. The robot's sensors gather data about its environment, including the location and existence of objects. This information is used by the robot's control software to plan its movements. The actuators move the robot components in response to orders supplied by the control software. The operating system oversees the hardware and resources of the robot, such as the power supply and memory. The planning software determines the sequence of movements that the robot must make to complete the assignment. The user interface helps to provide commands or change the robot settings.

For robot operations to be successful, hardware and software must be coordinated. A robot's capabilities and dependability increase with its coordination. With ongoing hardware and software advancements, the field of robotics is always expanding. As a result, more sophisticated and capable robots that can carry out a wider range of tasks are being created.

A robot is a machine that can operate faster than people; also, robots aid humans in their daily lives by reducing work complexity and workforce. Robots, in my perspective, are robots with minds developed by merging artificial intelligence, sensors, and machine vision. Humanoid robots, autonomous robots, teleoperated robots, and augmenting robots are just a handful of the several types of robots that are currently available.

Robots are becoming more efficient, flexible, and independent as AI and machine learning develop. In a variety of fields and applications, they are anticipated to become more significant in the future. However, the widespread adoption of robotics across a range of industries raises potential worries about job losses. Finally, robotics is a fascinating subject that creates innovative ideas that alter the present world.

1.1 Competitive Information

We, the blue team (team 4), As members of the blue team, we (Teams 3 and 4) worked together to get to the field's backstage area to park the robot during the autonomous drive and Driver controller drive. To earn high points in autonomous mode and as we are moving in the same direction to avoid colliding with one another, we talked it over internally and decided on safe parking so as to prevent robot collisions while they are performing autonomous tasks.

1.2 Relationship to Other Applications/Projects

The combination of OpenCV, webcam technology, AprilTag detection, Java SDK, Android Studio, FTC Control, and TensorFlow presents an adaptable and potent toolkit suitable for an array of uses. This stack of technologies makes it easier for robots to navigate and localize precisely. It also integrates AprilTags with FTC Control to improve robot performance in competitions. Although using TensorFlow for object identification increases complexity, it is still beneficial for locating and modifying aspects unique to a game.

1.3 Assumptions and Dependencies

Assumption during the Autonomous Game Mode:

The robot's success during the Autonomous Period and our team's strategy are largely determined by the autonomous game regulations for the 2023 FTC competition. Following these guidelines, our robot follows preprogrammed commands for a predetermined amount of time i.e. 30 seconds in as 12/12 square play field, and human control is not allowed at all during this time.

During the thirty-second Autonomous Period that starts the competition, our robot can only be operated by commands that have been preprogrammed.

Drive Team utilizes the "start" command found on the Driver Station touch screen to engage our robot, guaranteeing a hands-off approach.

Robot gains points on its own during the Autonomous Period in the following categories:

Points are given for effective navigation, which includes parking in the Backstage area. For completing this assignment on its own, each robot in our alliance receives points.

Tasks involving Randomization: Completing tasks involving randomization effectively yields points. The preloaded Pixels' locations and the related randomization objects—the white Pixel or Team Prop are crucial to these activities. To enable the robot to score points on its own, we place Pixels in key locations on the Backdrop and the assigned Spike Mark.

Pixel Placement:

Based on the autonomous placement of Pixels in certain areas of the Alliance Backdrop, points are provided. Points are awarded for each pixel put in the recessed Scoring section of our Alliance Backdrop, and points are awarded for each pixel placed in the Alliance Backstage.

Approach to the Autonomous Period is determined by these autonomous rules and categories, where our robot carefully plans and programs its behaviors to optimize our score without the need for human participation.

Autonomous Mode Execution based on the Assumptions:

Selection of Autonomous Op Modes:

Based on the objectives of the match, our team strategically decided before the match which Autonomous Op Mode to use. The robot's actions during the Autonomous Period were affected by this choice.

Robot's Initial Position:

The Blue Alliance designated A2 as our robot's beginning position, according to the game handbook. We made sure the robot was positioned exactly in the designated A2 spot, adjusted it as needed, and aligned it so it could recognize the April tag that was fastened to it.

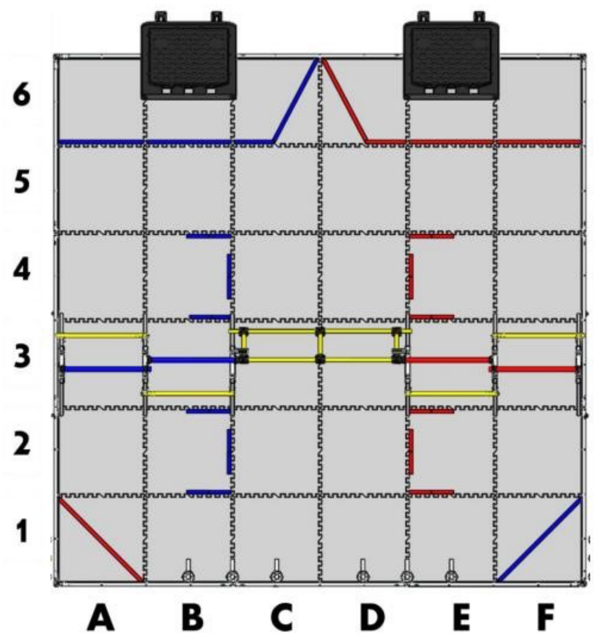

Fig 1.3.1 Initial Field for Blue and Red Teams

Randomization and Detection of Pixels:

We built our robot's algorithms and sensors to identify all three probable locations for dropping the Pixel, accounting for the potential randomization of Pixels. One crucial component of our plan was the robot's capacity to recognize various locations and modify its behavior accordingly.

Beginning and 30-Second Time Limit:

We first chose the Autonomous Op Mode, and then we started the 30-second countdown that the robot had to finish its autonomous tasks in. We gave the "start" instruction via the Driver Station and made sure the robot worked within the allotted time frame.

April Tag Detection and Pixel Positioning:

Our robot used the webcam to find the April tag linked to the desired location for Pixel placement during the Autonomous Period. The robot then made exact movements to align the Pixel with the designated line while dodging obstructions.

Servo-Based Pixel Drop Mechanism:

Our robot used a servo motor as a mechanism to drop the Pixel at the predetermined spot. This servo motor was calibrated and managed to guarantee precise positioning of the pixels.

Control of the Driver in an Autonomous:

Upon activating the Autonomous Period by clicking the "play" button, the robot functioned autonomously without requiring human intervention. The robot's capacity to locate the April tag and move to its designated place was the driving force for its autonomous behavior.

Planning the Path and Avoiding Obstacles:

The robot was able to go around the field and avoid obstacles since we had programmed it to follow a predetermined route. The accomplishment of autonomous tasks depended heavily on the robot's precise and effective navigation.

Careful planning, programming, and sensor integration led to our robot's Autonomous Mode. We were able to finish certain tasks in the allotted thirty seconds, showcasing our accuracy in April tag detection and Pixel insertion.

By including these specifics in our project report, you can provide readers with a clear and comprehensive picture of how our team approached the competition's autonomous mode and completed the necessary tasks. It displays the technical know-how and strategic judgment of our team.

1. After coordinating with team3(Blue teams) we identified the below paths during the randomization task.
2. As per the discussion with team 3, they are going to use the A and B field path for parking. So, to avoid collision we choose the below paths for safe parking in the backdrop Based on the object located on the field.

a. When the object is located in the center of the 2nd tile as mentioned in the below diagram. b. April tag we used here is below:

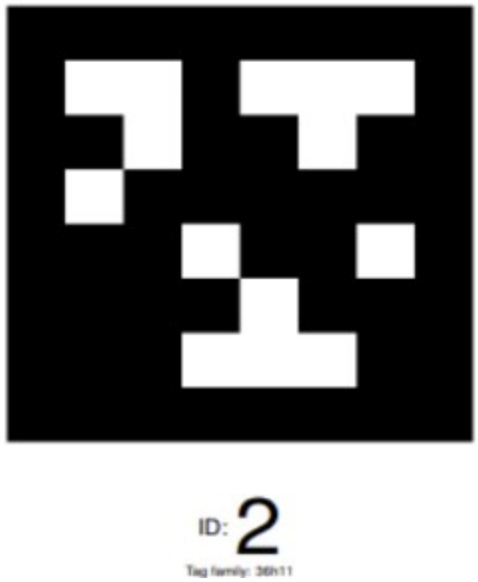

Fig 1.3.2.b Center position April tag

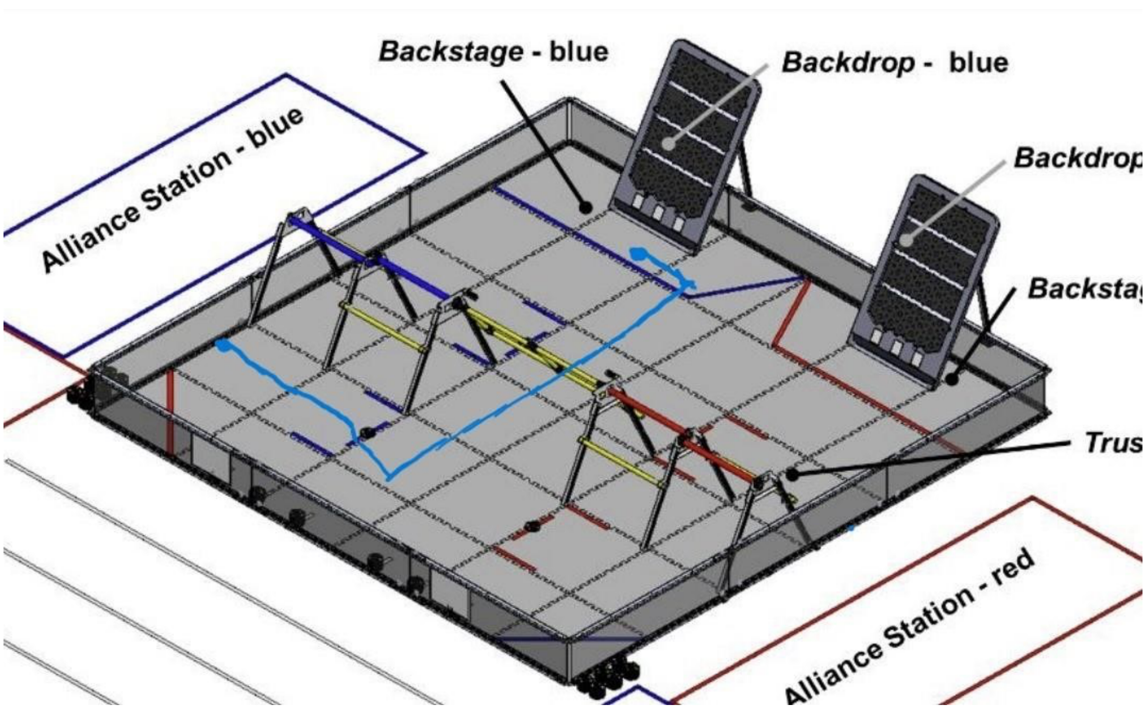

Fig 1.3.2.a center pixel drop and parking path

c. When the object located on the left side of the 2nd Tile.

d. April tag we used here is below:

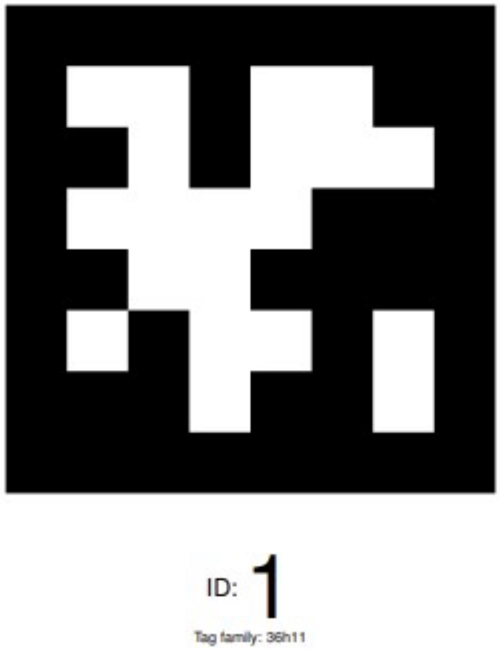

Fig 1.3.3.d Left position April tag

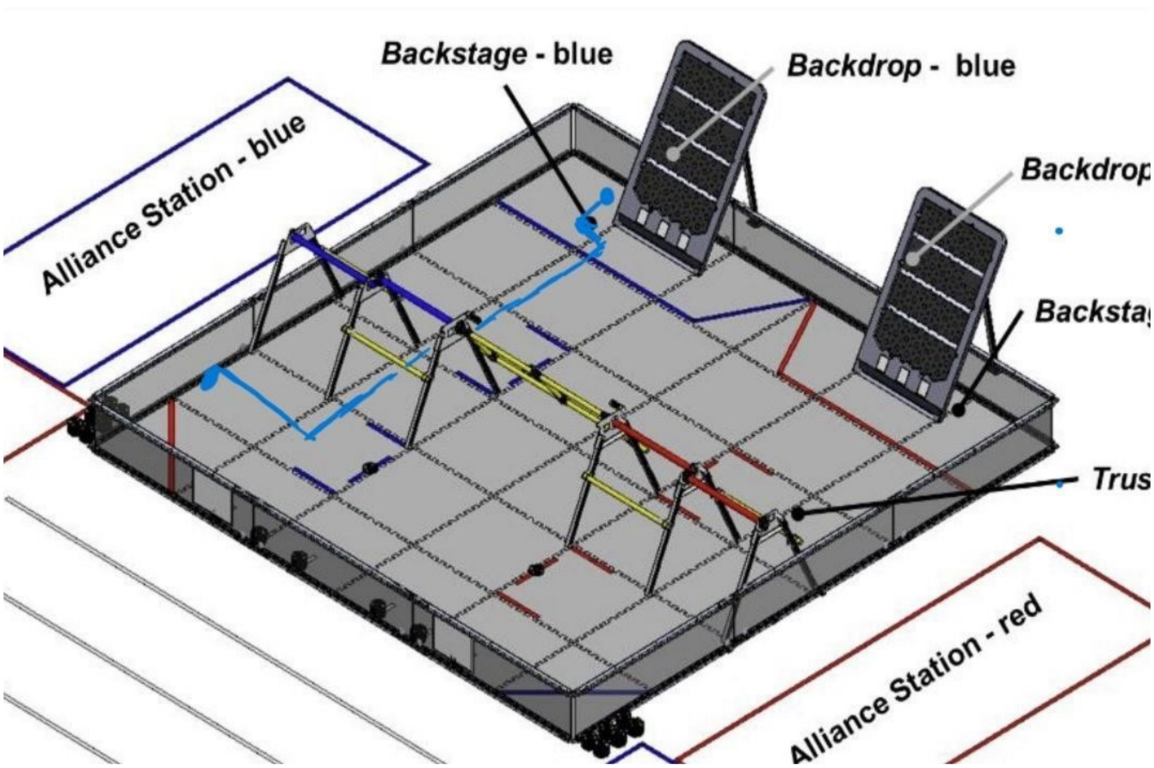

Fig 1.3.3.c Left side pixel drop and parking path

e. When the object is located on the right side of the 2nd tile on the field.

f. April tag we used here is below:

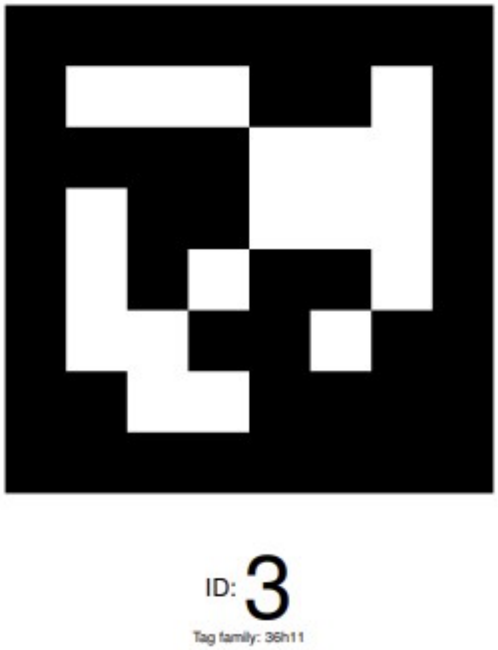

Fig 1.3.4.f Right position April tag

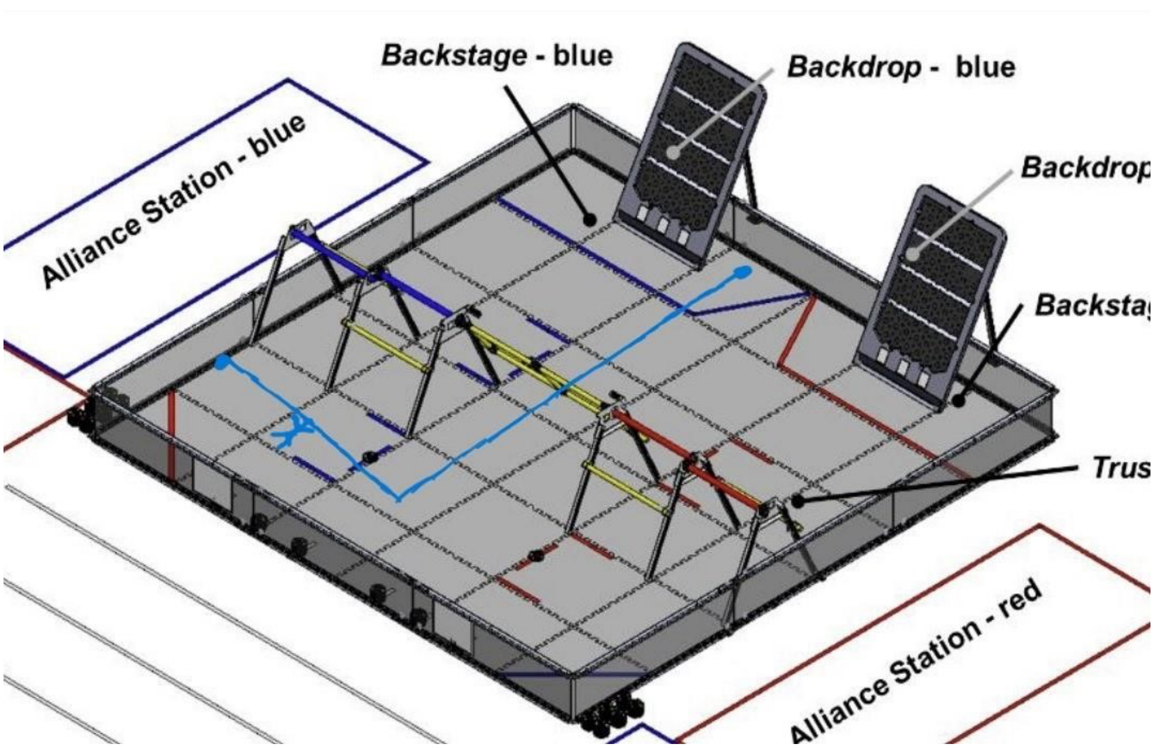

Fig 1.3.4.e Right side pixel drop parking path Driver-Controlled Period and Scoring:

Drive Teams get five (5) seconds after the Autonomous Period ends to set up their driver stations, during which they will also be issued a "3-2-1-go" countdown. In order to guarantee a smooth transition into the 120-second Driver-Controlled Period, this preparatory stage is essential. Drive Teams restart their active participation in the match by pressing the start button on their Driver Stations as soon as the word "go" is announced.

Driver-Controlled Tasks and Scoring at Rest:

Pixel Positioning:

Our team receives three points for each pixel that is carefully positioned on the recessed Scoring region of our Alliance Backdrop. Each pixel in our Alliance Backstage adds one point to our total score.

Artist's Bonus:

Each mosaic that our team creates by arranging Pixels in particular ways is worth an extra ten (10) points.

Bonus Set:

Pixels on a backdrop must be scored in a manner that creates a horizontal Set Line in order to receive a Set Bonus. Our alliance gains ten (10) points for each Set Line. Regardless of how many Pixels cross a vertical crossing in a Set Line, there is an additional one (1) Set Bonus contributed. Thirty (30) points is the maximum Set Bonus that our alliance can earn.

Strategy and Execution:

During the Driver-Controlled Period, our Drive Team skillfully arranges and places Pixels to optimize our score. In addition to placing individual pixels, the emphasis is also on making mosaics and reaching set bonuses. During this time, the robot must be precisely controlled and coordinated in order to navigate the field, place Pixels, and maximize scoring opportunities.

For our team to perform at our best during the Driver-Controlled Period, we must be able to quickly modify our plans in response to the changing dynamics of the game, as well as the activities of our opponents and alliance.

To succeed in the 2023 FTC competition, our team intends to demonstrate a blend of talent, teamwork, and strategic decision-making by comprehending and utilizing the scoring criteria for Driver-Controlled assignments.

End Game and Robot Location:

End Game Overview:

The End Game, which lasts for thirty seconds during the Driver-Controlled Period, is a crucial stage where precise movements and calculated actions can have a big impact on the result of the match. The

DriverControlled Period continues to be scored during the End Game, with some feats having significant point values.

Robot Location:

There are two location-based scoring possibilities that are mutually exclusive during the End Game, and a robot can only receive points for one of these jobs.

a) Parked Behind the Scenes:

Robots can gain five (5) points per robot by strategically parking in the Backstage area for the corresponding Alliance. This accomplishment promotes effective alliance member placement and coordination.

Strategic Considerations:

Our group is aware of how important the End Game is in determining how the match turns out in general. We carefully organize our moves, taking into account the point values attached to each assignment as well as the exclusive nature of Robot Location awards. This involves determining whether it is possible to effectively park in the backstage area or suspend our robot from the rigging.

Penalties:

Points Deducted for Robots or Alliances Induced to Violate a Regulation:
Action: It's unclear what particular actions are connected to this point, but it implies that robots or alliances may not always receive penalty points directly and that there may be circumstances in which breaking the rules necessitates taking a penalty or action.

Violation of the Competition Area Entry or Exit Rule:
Action: A warning is issued for breaking any of the rules pertaining to entering or leaving the competition area. Repeat offenders receive a Yellow Card. This highlights how crucial it is to follow the rules regarding entry and leave during the tournament.

Drive Team Absent Safety Equipment:
If any safety equipment is absent from the driving team, a notice is sent out. The offending drive team member(s) must leave the competition area and cannot be replaced if the problem is not fixed in 30 seconds. This emphasizes how important safety precautions are and how quickly safety gear problems should be resolved.

Dangerous Robot or Harm to the Field of Play:
Action: The robot is disabled if it is operating in a dangerous manner and the problem is likely to continue. For risky operation, a Yellow Card is an optional feature. The issue might get worse and result in a Red Card if there is a lot of damage and/or delays. This highlights how crucial it is to keep robot operation safe and reduce harm to the playing ground.

1.4 Future Enhancements

Advanced Object Recognition:

Improve the robot's capacity to identify and engage with a greater variety of objects. This might entail using more advanced machine learning models to identify and manage items other than AprilTags, perhaps learned on different datasets.

Integration of Multiple Sensors:

By including extra sensors, such as LiDAR, ultrasonic, or inertial measurement units (IMUs), the robot's sensory capabilities can be increased. The navigation, obstacle avoidance, and general situational awareness could all be enhanced by this multi-sensor fusion.

Autonomous Decision-Making:

Give the robot's decision-making mechanisms higher-level autonomy. Give the robot the ability to decide on more complicated tasks by combining sensor data, outside signals, and preset goals. This might be especially helpful in situations where there are several goals or changing needs for the activity at hand.

1.5 Definitions and Acronyms

OpMode :

Operational Mode, often known as Op Mode or opmode, is a class found in the FTC SDK (source code for robot controller apps). You add your code to the controller app using this class. Actually, your code only makes up a portion of the controller app; the rest is found in the FTC source code.

Autonomous Period:

the first thirty seconds of the match, during which the robots only move and respond in response to directives that have been pre-programmed onto the onboard robot control system by the Team and inputs from sensors.

It is not allowed for humans to operate robots at this time.

TeleOp Mode:

Teleop mode, also known as teleoperation mode, is a robotics control technique in which a human operator remotely guides and controls a robot. With a range of input devices, including joysticks and other controllers, the operator can direct the robot's movements and actions in real-time while using this mode. Applications for teleop mode can be found in a variety of contexts, such as industrial settings where accuracy is essential or robotics tournaments where robot operators guide their machines to complete predetermined tasks. This mode facilitates tasks that might be dangerous or inaccessible to humans directly, demonstrating the synergy between human decision-making and robotic capabilities.

Back stage:

The Specialized Area for Alliances below a backdrop. The Backstage is roughly 72 inches (183 cm) long by 23 inches (58.4 cm) deep, and it is defined by a nominal 1-inch (24 mm) broad tape. One (1) red and one (1) blue Alliance Specific Backstage are available.

2 Project Technical Description

The Complete technical Architecture explained as below.

2.1 Application Architecture

We will use Android studio that offers our team flexibility to create Java programs for competition robots. The control system can have two phases. An autonomous phase where the robot operates without any human input, and Driver controlled phase where the robot can receive input from up to two human drivers. We can use two android devices.

One is mounted to the robot, called the Robot Controller, which acts as the robot's brain. There are two hardware options: the REV Robotics Expansion Hub or the REV Robotics Control Hub. The REV Robotics Control Hub is an integrated version of the robot controller. It combines an Android device built into the same case as a REV Robotics Expansion Hub. It runs the robot controller app.

A second Android device sits with the team drivers that can have one or two gamepads. This second device is known as the Driver Station. It helps to issue commands to the Robot Controller. It runs the driver station app

Op modes (Operational modes) are computer programs used to customize a competition robot's behavior. Android Studio integrated development environment (IDE) is used to create their op modes.

Driver Station Overview:

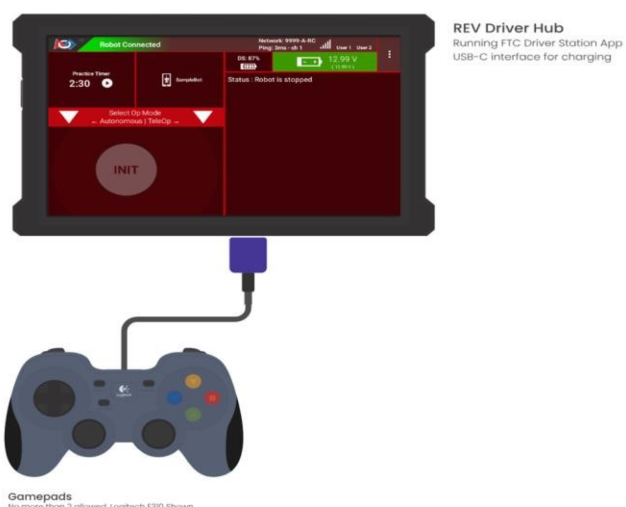

Fig 2.1.1 Driver station

The heart of the Driver Station is the Android device that runs the Driver Station App (Option A in the above image). USB-OTG Adapter cable connects the Android device and the USB Hub where the gamepads are connected. This Android Device requirement can be fulfilled by using a REV Driver Hub (Option B in the above image). The gamepads can connect directly to the REV Driver Hub, and an external power supply must be connected to the REV Driver Hub.

Robot Controller Overview:

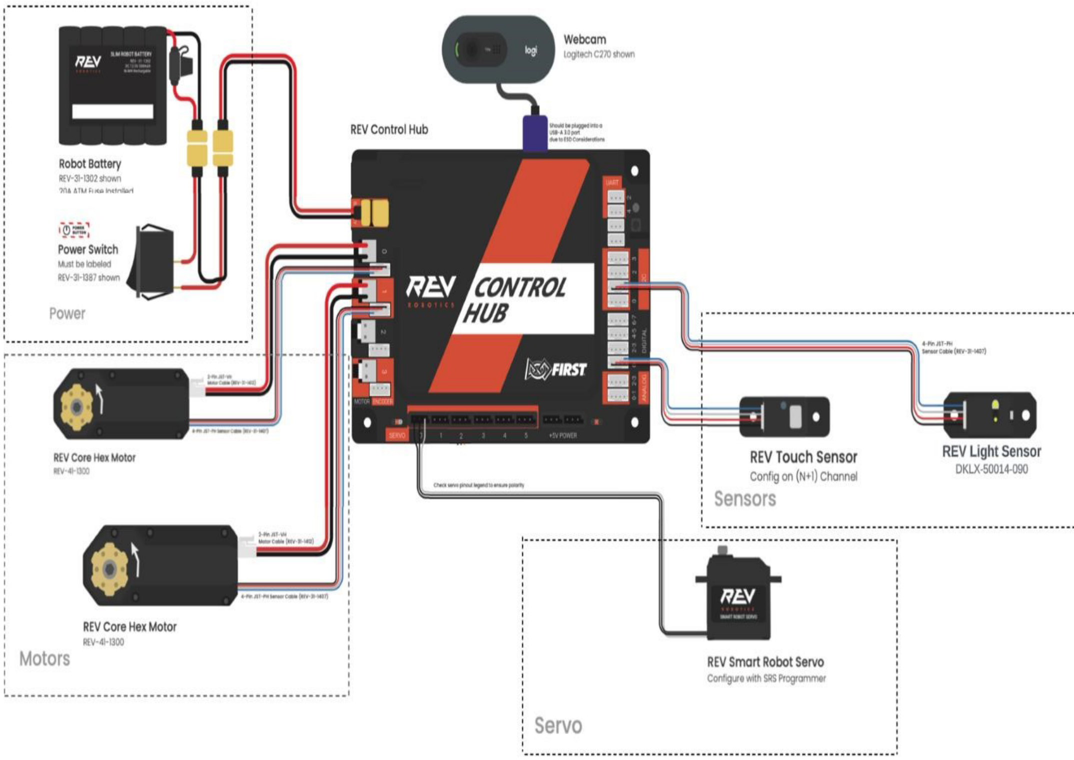

Fig 2.1.2 Control Hub

Configuration Details:

Device	Name used in our code	Port number
REV Core Hex Motor	left_drive	MOTOR ENCODER - 0
REV Core Hex Motor	right_drive	MOTOR ENCODER - 1
REV Touch Sensor	sensor_touch	DIGITAL 0-1
REV Light Sensor	sensor_distance	I2C - 1
REV Smart Robot Servo	right_hand	SERVO - 0
Webcam Logitech C270 HD 720p	webcam	USB-A 3.0

The main power of a robot comes from one 12V battery. A main power switch must control all power provided by the main battery. We can use a Power Distribution block (like REV XT30) to help distribute the power to devices. REV Servo Power Module can boost the power supplied to 3-wire servos. REV Control Hub and REV Expansion Hub has Battery, Motor, Encoder, Servo, +5V Power, Analog, Digital, I2C, RS485, UART and USB-B ports.

REV Control Hub has additional USB-A, USB-C and HDMI ports. XT-30 connectors are used to power the REV Hub using Battery ports. We can connect a laptop for loading the SDK using USB-C port. RS485 port can be used if we wish to use a second REV (like in the above image). UART is used only for the shaft. An absolute encoder can indicate the displacement of the shaft from its starting position and the exact angle the shaft is currently at relative to a zero position.

Developer debugging. HDMI port can provide the video output for the device. UVC Cameras can be connected to the Hub using USB-A ports.Encoders can be connected to the Hub using Encoder ports (four ports on each hub). An encoder is a device that measures the rotational displacement around an axis. An incremental encoder can send a tick per partial rotation.

Servos can be connected to the Hub using Servo ports (six ports on each hub). A servo is a type of device that takes a Pulse-Width Modulated (PWM) signal as an input and, with the help of an embedded controller, produces linear or rotational movement based upon the input signal. Servos may take an input signal generated by a REV. Hub (either by a Control Hub or Expansion Hub) which itself provides 5V of power and a limited amount of current.

Sensors can be connected to the Hub using Analog, Digital, and I2C ports. Sensors are devices that detect and measure physical properties or environmental conditions and convert them into signals or data that can be interpreted or displayed.

Computer vision Technologies:

April Tags:

April tags were used in tech challenge 2022 –2033 invented by University of Michigan. These tags are attached to the robots and these tags continuously pose the data through we can get the robots position on the field. With that location the player can easily navigate the robot to the target location.

2.2 Application Information flows

During the autonomous period, Robots perform tasks, navigate the competition field, and score points throughout this period using preprogrammed instructions or algorithms. Points are awarded to players who successfully accomplish goals, such as putting objects in certain positions or getting to specific spots on the field. At the end of the autonomous time, the team with the highest score receives an autonomous bonus.

In robotics competitions, the autonomous phase is vital since it displays robot skills and assesses how well they can work on their own. It also sets the stage for future developments in autonomous systems by showcasing the advances in robotics technology.

The autonomous period in robotics describes a stage of technological development where robots become capable of independent operation and decision-making without continual human supervision. Robots can now navigate complex environments, adjust to changing conditions, and carry out tasks with a level of autonomy that was previously unthinkable thanks to the integration of artificial intelligence, machine learning, and sensor technologies. This is a transformative era.

Teleop mode, also known as teleoperation mode, is a robotics control technique in which a human operator remotely guides and controls a robot. With a range of input devices, including joysticks and other controllers, the operator can direct the robot's movements and actions in real-time while using this mode. Applications for teleop mode can be found in a variety of contexts, such as industrial settings where accuracy is essential or robotics tournaments where robot operators guide their machines to complete predetermined tasks. This mode facilitates tasks that might be dangerous or inaccessible to humans directly, demonstrating the synergy between human decision-making and robotic capabilities.

2.3 Capabilities

The robot should be able to use sensors like encoders and cameras to navigate the competition field on its own. Accurate placement is essential to accomplishing certain goals, such arriving at a predetermined starting location or placing objects in predetermined areas. The robot might be able to identify and locate AprilTags or other markers using computer vision techniques, allowing for precise localization and navigation. During the Autonomous Period, the robot should be programmed to carry out specified duties, like moving to predetermined destinations or scoring game elements. Algorithms for making decisions are frequently used in autonomous programming, enabling the robot to adjust to shifting circumstances on the field.

During the Driver-Controlled Period, teleoperation—in which human drivers operate robots using joysticks or gamepads—is the main capacity. Based on the game's dynamic progression and the robot's sensory input, human operators make decisions in real time.

2.4 Risk Assessment and Management

Give the operators instructions to handle the robot gently, avoiding forceful or needless hits. Give instructions on how to move and elevate the robot such that damage is avoided.

Stress how crucial it is to switch off the robot when not in use. Give precise directions on how to handle batteries and charge them properly. Establish a shut-off process to cut off power sources.

Designate and label the robot's operating prohibited regions explicitly. Adopt a rigorous policy prohibiting field entry while robots are actively operating. To identify the restricted areas, use visual signals or physical obstacles.

teaching operators how to minimize the risk of entanglement or disconnection by properly arranging and securing wiring. To stop the robot's mobility from being hampered, use cable management devices and fasten loose cables.

3. Release and Transition Plan

Update the REV Control Hub firmware according to the manufacturer's instructions and periodically check for updates. Make that the control software on your robot is compatible with the most recent firmware. Updates to the firmware frequently bring new

for updates and new releases on a regular basis. Examine and update the project's third-party dependencies and libraries.

4 Project Code Function

Function that is used to get distance:

```
//Function capable to get Distance
1 usage  harshach
public double getDistance() { return distanceSensor.getDistance(DistanceUnit.INCH); }
```

Fig 4.1 Distance

Function related to Servo:

```
// Robot Servo Rotating Function for Autonomous mode
2 usages  harshach
public void servoRun()
{
    servo.setPosition(0);
    myOpMode.sleep( milliseconds: 1500);
    servo.setPosition(1);
    myOpMode.sleep( milliseconds: 1500);
}
```

Fig 4.2 Servo code to drop the pixel in autonomous mode.

```java
// Turning the servo in direction specified for the Driver Mode.
2 usages  harshach
public void turnServo(String string)
{
    if(string.equals("UP"))
    {
        servo.setPosition(0);
    }
    else if(string.equals("DOWN"))
    {
        servo.setPosition(1);
    }
}
```

Fig 4.3 Servo code to drop the pixel in Driver-Controlled mode

Encoder Move Function:

```java
while (myOpMode.opModeIsActive() &&
        (runtime.seconds() < timeoutS) &&
        (leftWheel.isBusy() && rightWheel.isBusy()))
{
    // Compare the string to use avoidCollision or not
    if(string.equals("GO"))
    {
        telemetry.addLine( lineCaption: "Going");
    }
    else if(string.equals("avoidCollision"))
    {
        while (getDistance()<8){
            telemetry.addData( caption: "Object present in Robot Path;", value: "Waiting till it clears");
            telemetry.update();
            leftWheel.setPower(0);
            rightWheel.setPower(0);
        }
        runtime.reset();
        leftWheel.setPower(Math.abs(speed));
        rightWheel.setPower(Math.abs(speed));
    }
}
```

Fig 4.4 Distance with encoder code snip

Web cam:

```
int cameraMonitorViewId = myOpMode.hardwareMap.appContext.getResources()
        .getIdentifier( name: "cameraMonitorViewId",  defType: "id", myOpMode.hardwareMap.appContext.getPackageName());
camera = OpenCvCameraFactory.getInstance().createWebcam(myOpMode.hardwareMap
        .get(WebcamName.class,  deviceName: "webcam"), cameraMonitorViewId);
aprilTagDetectionPipeline = new PipelineForAprilTagDetection(tagSize, fx, fy, cx, cy);

camera.setPipeline(aprilTagDetectionPipeline);
harshach
camera.openCameraDeviceAsync(new OpenCvCamera.AsyncCameraOpenListener()
{
    harshach
    @Override
    public void onOpened() { camera.startStreaming( width: 800, height: 448, OpenCvCameraRotation.UPRIGHT); }
    harshach
    @Override
    public void onError(int errorCode) { }
});
```

Fig 4.5 web cam initialization code:

5 Internal/external Interface Impacts and Specification

In the field of robotics, telemetry is the process of gathering information from sensors and other sources on a robot and sending it to a distant location for the purposes of control, analysis, or monitoring. For real-time understanding of a robot's status, surroundings, and performance, telemetry systems are essential. The status of the motor, battery levels, sensor readings (such as temperature, LiDAR, and cameras), location, and other details can all be included in this data.

Through telemetry, operators or controllers can remotely check the health of the robot, identify problems, make deft decisions, and even modify its behavior or mission parameters. It's crucial in situations like space exploration, deep-sea exploration, hazardous environments, and autonomous operations where direct human supervision or intervention may be difficult.

Depending on the range and particular needs of the application, wireless communication technologies like Wi-Fi, Bluetooth, cellular networks, or specialized communication protocols are frequently used to send the telemetry data transmitted from the robot. This information can be processed by other systems to enable automated decision-making, or it can be presented to human operators via control panels or user interfaces.

6 Design Units Impacts

6.1 Functional Area A/Design Unit A

Replace this section with a list of the impacted design units (functional areas). For new products, this would be a list of all the new functional areas and would, therefore, describe the new system architecture. However, when it is done, the design should clearly reflect how the design units fit together to define the project. Each functional area (or design unit) should have its own subsection below. If there are no known impacts to a given functional area, then that should be explicitly stated. For your project, it could be only one design unit that will cover all requirements, or you could have multiple design units (e.g., one for the web interface, one for your application, and one for the maintenance process, etc.).

Touch Sensor:

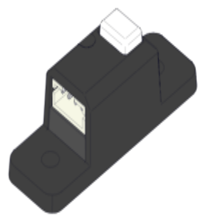

Fig 6.1.1 Touch sensor

A digital sensor that can be used as a button input or a simple mechanical limit switch is the REV Robotics Touch Sensor (REV-31-1425). The touch sensor works like a keyboard button; when the button is pressed, it alerts the Robot Controller and causes a coded action to run. Depending on the use case, this action may occasionally cause the motors to halt or the encoder angle to reset.

The Touch Sensor operates on a binary, just as all digital sensors. The value the Expansion Hub reads when the button is depressed is 3.3V (high), and when the button is depressed, the LED light illuminates, and the Expansion Hub reads 0V (low).

Color and Distance Sensor:

Fig 6.1.2 Color and Distance Sensor

This sensor contains a built-in IR (optical) proximity sensor, a white LED for active target lighting, and the ability to read and compare colors. Supports High Speed I2C Communication (400kHz) and Auto Increment Register Read, allowing the user to return the whole contents of the color register and status register in a single read instruction as opposed to four separate read operations.

Specifications of WebCam:

Fig 6.1.3 WebCam

- HD 720p video calling and HD video recording, 2.4 GigaHertz Intel Core2 Duo, 2 GB RAM,200 MB hard drive space
- Video capture: Upto 1280 x 720 pixels, Logitech fluid crystal. Focus type: Fixed focus
- Crisp 3 MP photos technology, Hi speed USB 2.0
- Compatible with: Windows 8 or later, MacOS 10.10 or later, Chrome OS, with USB-A port.

Detecting April Tag Using Webcam:

April Tags are distinctively patterned 2D barcodes. Using a webcam, find April Tags:

OpenCV records video frames from the camera. To make picture processing easier, convert the frames to grayscale. To find April Tags in the grayscale frames, use the April Tag library (e.g., apriltag). Data out of the detection results, like tag ID and pose. It is optional to add bounding boxes or markers around the identified April Tags on the original video frames to visualize them. **Add the AprilTag data that was found to our robotics application:**

Gather pertinent information for navigation and localization activities, such as pose and tag ID. Based on the information from the identified AprilTag, judgments or modify the robot's behavior. April Tags are 2D barcodes with a unique pattern.

Locate April Tags with a webcam:

To capture video frames from the camera using OpenCV, convert the frames to grayscale to facilitate the processing of pictures. By using the AprilTag library (e.g., apriltag) to locate AprilTags in the grayscale frames, data such as pose, and tag ID are the results. To view the recognized AprilTags on the original video frames, we can choose to include bounding boxes or markers around them. The discovered AprilTag data to our robot’s application. Collects relevant data, like pose and tag ID, for localization and navigation tasks. The robot's actions are based on the data from the recognized AprilTag.

Sample April Tags:

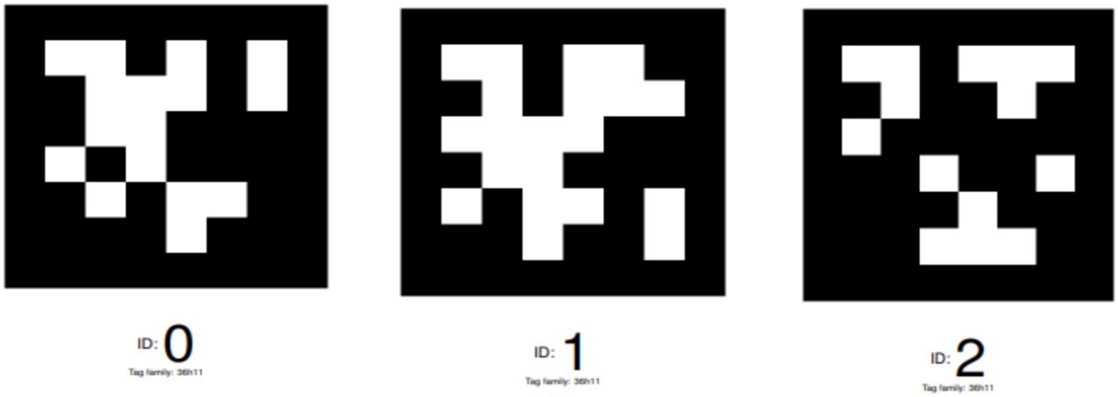

Fig 6.1.4 April Tags

HiTEC HS-485HB SERVO:

In the Current Game, we are using the servo to drop the pixel at the object location in the randomization task during the autonomous mode.

Fig 6.1.5 Servo Motor

Product Description:

The HS-485HB updates the very popular HS-475HB by adding more torque and greater speed. With its top ball bearing, heavy duty, high impact Karbonite™ gears and high-performance circuitry, the HS-485HB offers lots of torque and features excellent centering and resolution.

- **Applications:** Heli's and Aircraft up to 72" or up to 12lbs when using just one servo per control surface.
- **Note:** Be aware that high vibration gas (not glow) engines combined with large contol surfaces and large thows (3D models) can put undue strain on the gears. Metal geared servos are recommended for these applications even if under 12lbs.
- **Limitation:** Do not use thread lockers on Karbonite geared servo as it will cause the output shaft to fail.

Features:

1. Top Selling Heavy Duty Standard Servo
2. High Impact Karbonite Gears
3. Top Ball Bearings
4. Excellent Centering and Resolution

6.2 Requirements

Welcome to CENTERSTAGESM presented by RTX. Matches are played on a Playing Field initially set up as illustrated in Figure 4.2 -1. Two Alliances – one "red" and one "blue", made up of two Teams each – compete in each Match. The object of the game is to earn as many points as possible by performing the achievements outlined below.

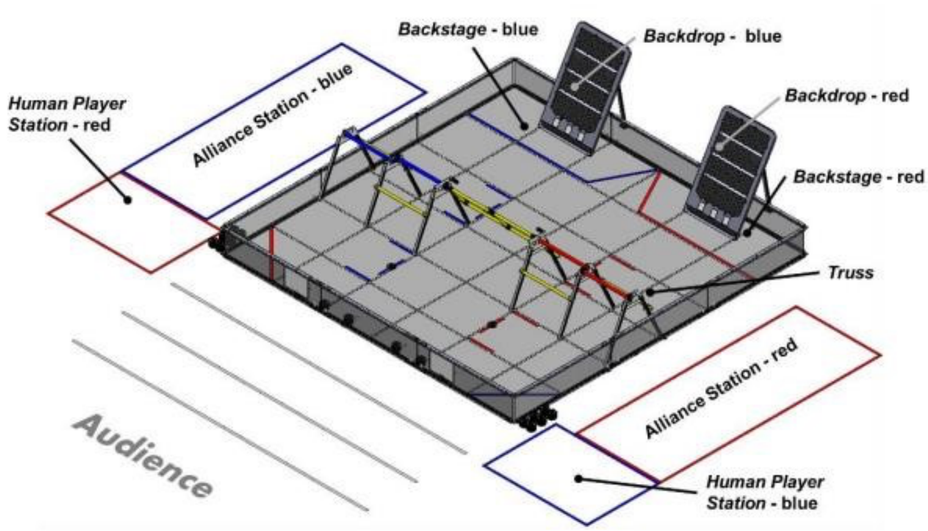

Figure 4.2-1 – Isometric view of the *Playing Field*

Fig 6.2.1 Playing Field

The Match starts with a 30-second Autonomous Period in which Robots operate using only preprogrammed instructions and sensor inputs.

The following Robot actions earn points during the Autonomous Period:

1. Navigating to their Alliance Backstage.
2. Placing Pixels On their Alliance Backdrop or In their Alliance Backstage.
3. Identifying the Randomization Object on the randomly selected Spike Mark. Alliances can earn points by:
    a. Placing a Pixel On the randomly selected Spike Mark.

Additional points are earned for these tasks when a Team uses their Team Prop in place of the tournament provided Pixel.

The two-minute Driver-Controlled Period follows the Autonomous Period. Robots earn points by:

1. Placing Pixels On their Alliance Backdrop or in their Alliance Backstage.
3. Scoring Pixels that cross the Set Line on their Alliance Backdrop.

The final 30 seconds of the Driver-Controlled Period is called the End Game. In addition to the previously listed

Driver-Controlled Period Scoring activities, Alliances earn points by:

1. Suspending Robots from their Alliance Rigging.
2. Parking Robots In their Alliance Backstage.

Each game comes with its own unique set of challenges. In CENTERSTAGESM there are multiple ways Teams can utilize technology to assist them in solving these challenges. Teams may use the built-in technology, or they can come up with their own solutions to solve the challenges.

The challenges include:

1. Object Identification:

a) Robots can use their on-board control system and sensors to identify Game Elements.

b) Robots can use the built-in TensorFlow technology to decode the randomized Autonomous task.

2. Field Navigation:

Built-in AprilTag technology helps the Robot identify and navigate to important locations on the Playing Field.

3. Situational Awareness:

Sensors provide situational awareness during the Autonomous Period, automate operations and provide feedback during the Driver-Controlled Period. Cameras, motor encoders, distance, and color sensors are useful for solving gameplay tasks.

7 References

1. https://www.firstinspires.org/resource-library/ftc/technology-information-and-resources

2. Game manual:

3. How to configure camera detection for FTC Power Play using OpenCV & April Tags:

https://www.youtube.com/watch?v=GXeSsbGXURM

4. 2023-2024 CENTERSTAGE presented by RTX Game Animation Game Video https://www.youtube.com/watch?v=6e-5Uo1dRic

5. Testing REV Touch Sensor https://www.youtube.com/watch?v=aZYBmU6ocec&t=34s

6. Testing REV Color Distance Sensor https://www.youtube.com/watch?v=VAPxENrKRvo

8 Appendices

Git hub repository link https://github.com/Harshachennoor/Grad_Seminar_Project/tree/main/FtcRobotController-9.0