A Byte Irrational

FTC 15065

ENGINEERING PORTFOLIO

Freight Frenzy

2021-2022 Season

Table of Contents

Team Introduction 3
Team Sustainability Plan 3
Outreach and Funding 3
Budget and Finances 4
Season Highlights 5
Engineering Overview 7
Strategy 10
Software 11

Team Introduction

Our team, **A Byte Irrational**, chartered by Orlando Science Schools, adopted our team number in 2019 with 5 members who were new to FTC. In 2020, we had 2 returning members from the previous year, with 5 members in total. In the 2021-2022 season, we began with 4 returning members and 3 new members. We have 4 engineering mentors: **Mr. Don Boughton, Ms. Karen Oliver, Mr. Dhruv Sharma**, and FIRST/OSS alumni **Mr. Carlos Barea** and very supportive parents. We have 7 student team members:

Sarah -9th grade – Mechanical, Drive Team

Diyar -9th grade-Mechanical and Electrical, Drive Team Coach

Joshua -9th grade – Software and Electrical, Drive Team Alternate

Kaavya -9th grade – CAD and Mechanical, Drive Team Alternate

Shourya -9th grade –Mechanical and Software, Drive Team

Hamsika-8th grade - Mechanical

Dilan -8th grade - Business and Software

We highly value having a cohesive team that works well together, cooperates, and balances hard work and fun to keep our members motivated and interested in robotics.

Team Sustainability Plan

Our plan to keep the team sustainable for the years to come is to continue applying for grants that we have gotten in previous years. Using our progress, we can show the companies that we apply for grants from how their funding is being put into use and is helping us improve. Additionally, we plan to develop and put into use a new system for accepting and rejecting applications to join our team.

Outreach and Funding

LAKE SILVER ELEMENTARY SCHOOL OUTREACH PROJECT

Our team is very passionate about enabling future generations to pursue STEM careers, and we would like to expose children to robotics from a young age. To achieve this, we planned a trip to a local public elementary school, Lake Silver, where we would have a hands-on presentation allowing kids to interact with STEM components such as model circuits. We were able to communicate with the school principal and were able to confirm they were interested and receptive towards the idea. Unfortunately, due to challenges with COVID restrictions we have been unable to decide upon a time to hold this outreach event. We will continue working with the school to determine a better date for our team to go to Lake Silver

MAKER FAIRE 2021

Our team participated in the Orlando 2021 Maker Faire. We promoted the FIRST organization, demonstrated our robot and explained how robots work to the many children and families who stopped by the field. We allowed the children to drive the robot around the field and attempt to score game elements in a make-shift goal. It was exciting to see so many young children interested in robotics.

3-D PRINTER DONATION

Our team purchased and donated a 3-D printer to our high school. The previous 3-D printer was too small to make a wide range of robotics parts and was always being used by other classes in the school. To minimize these issues, we decided to buy another larger 3-D printer that was dedicated to the school's robotics teams.

Budget and Finances

Parts		Subtotal:	$1,372.78
Electronics		Subtotal:	$504.25
Game Field items		Subtotal:	$1,139.10
Hardware Tools		Subtotal:	$240.00
Administrative		Subtotal:	$389.00
Registration		Subtotal:	$575.00
	TOTAL BUDGETED:	**$4,220.13**	
	Lockheed Martin Grants	$1,750.00	
	Raytheon Registration Grant	$275.00	
	Collins STEM Grant:	$1,000.00	
	HRSoft Grant:	$500.00	
	Team Funded:	$1,400.00	
	TOTAL INCOME:	**$4,925.00**	

We have 4 financial sponsors this season: **Lockheed Martin, Collins Aerospace, HRSoft** and **Raytheon Technologies**. We are also supported by our school -**OSS**, which provides a facility to work on our robot and practice. HRSoft was a late season sponsor so we were unable to include them on our shirt design for this season. With the 4 sponsors and our parent contributions we had a season income of just under $5000. We took the opportunity to use some of the team finances to buy a perimeter field kit and are holding a reserve to allow us to buy the game elements on the day of kick-off. We typically wait for the school to buy the game elements but this year due to supply chain issues, the order was significantly delayed and we didn't get a setup until the day before the 2nd meet. If we can order earlier in the season then all the teams at our school will be able to practice more efficiently.

Season Highlights

1ST MEET - REMOTE

Date: 11/06/21 **Time:** 3:30PM – 4:30PM
Members in Attendance: Sarah, Diyar, Kaavya, Dilan, Hamsika, Joshua, and Shourya
Drive Team: Sarah (driver), Shourya (driver), and Diyar (coach)

Match	Score	Autonomous Points	Teleop Points	Endgame Points
1	**52**	**5**	**29**	**18**
Reflection: In match one, the alliance hub was unbalanced, and one freight intended for the top layer of the hub landed on the middle layer. In addition, while transporting a freight to the hub it was dropped, but a smooth recovery was made.				
2	**47**	**5**	**18**	**24**
Reflection: During this match we left the alliance hub unbalanced during endgame and we dropped the freight twice while trying to place it on the hub.				
3	**65**	**5**	**26**	**34**
Reflection: In match three our hub was balanced, and we got 4 ducks, but one freight landed on the bottom layer.				
4	**67**	**5**	**28**	**34**
Reflection: We scored 4 ducks and our hub was balanced, but 2 of the freight we were trying to place on the alliance hub landed on the bottom layer instead of the intended top layer.				
5	**57**	**5**	**24**	**28**
Reflection: In this match our hub was imbalanced, and we scored 3 ducks instead of our usual 4.				
6	**69**	**5**	**40**	**24**
Reflection: This was our highest scoring match due to our Teleop score being higher than the other matches.				

2ND MEET - REMOTE

Date: 12/11/21 **Time:** 3:30PM – 5:00PM
Members in Attendance: Sarah, Diyar, Kaavya, Dilan, Hamsika, Joshua, and Shourya
Drive Team: Sarah (driver), Shourya (driver), and Diyar (coach)

Match	Score	Autonomous Points	Teleop Points	Endgame Points
1	112	36	28	48
Reflection: In autonomous the 4-bar lifter was wobbly and so we hit the Alliance shipping hub but we still got it on the right level. During the tele-op we had hit the alliance shipping hub multiple times during our match.				
2	121	31	32	58
Reflection: This match had gone with minimal issues during the entire thing.				
3	97	26	29	52
Reflection: At the start of the match the cube had stuck to our gripper and unbalanced our shipping hub. In the middle of teleop we had fixed our shipping during it as well.				
4	102	20	24	58
Reflection: For our fourth match, during autonomous the stickiness of the gripper held on to the block too much and ended up knocking the block out of the alliance shipping hub. During tele-op we had multiple occurrences of the same problem: the gripper's stickiness.				
5	124	36	30	58
Reflection: During match five we also encountered a problem with the gripper's stickiness and a duck ended up being flung off the alliance shipping hub. Another problem we noticed was that the robot was being				

moved away from the hub before a freight was fully dropped onto the hub, causing it to fall.				
6	129	31	40	58
Reflection: For our final match, we had our first experience with rebalancing the hub during a match. We ended up optimizing most of our strategy for this match, thus setting our new record of 129.				

3RD MEET- LIVE

Date: **1/22/22** **Time:** 7:45 AM – 4:00 PM
Members in Attendance: Sarah, Diyar, Kaavya, Dilan, Hamsika, and Joshua
Drive Team: Sarah (driver), Joshua (driver), and Diyar (coach)

Match	Score	Autonomous Points	Teleop Points	Endgame Points
1 (Qual. 3)	56*	16	34	6
Reflection: On the red alliance with Team 16249, Excaliper. Focused on the shared shipping hub and successful capstone detention and placement of pre-loaded box. Did not cap or completely park in warehouse. Total of 5 freight objects including element on alliance hub.				
2 (Qual. 5)	138*	22	48	64
Reflection: On the blue alliance with Team 9277, Hazmat Robotics Explosives, we were initially placed by the warehouse. Autonomous – parked completely in warehouse, successful duck detection and placement (pts not counted) Teleop- 4 pieces of freight placed on 3rd level. Endgame – 4 ducks and unbalanced shipping hub for blue alliance.				
3 (Qual. 8)	99*	26	18	55
Reflection: Red alliance with Team 15033, Bug Bytes. We started by carousel, shipping element detection and placement successful. We were blocked from accessing the shared shipping hub by the blue alliance and Bug Bytes shipping marker was in the warehouse blocking us from going over the barrier to the shared hub. This was our first time experiencing a team using a defensive tactic during the match.				
4 (Qual. 10)	141	0	52	89
Reflection: On the blue alliance with Team 17288, Rubber Bandits, we were unable to practice with our alliance partner prior to the match to update the timing for Autonomous. Unfortunately, we had a collision with our alliance partner when we both ran our autonomous modes.				
5 (Qual. 13)	70	46	28	6
Reflection: On the blue alliance with Team 17613, RobotSakers. Received a 10-point penalty. Our capping was unsuccessful, and our alliance partner was not able to spin the carousel, so we attempted to but ran out of time, resulting in a low endgame score.				
6 (Qual. 16)	60*	16	14	30
Reflection: On the red alliance with Team 17305, Ducks In a Row. We ran autonomous from the carousel side, but our alliance partner did not move into the warehouse as planned and our robot collided with the edge of their robot when it was going over the barrier and it flipped. We were unable to flip back up and our alliance partner continued to play to try to score points. Although it was a low score, we still won the match.				

** Awaiting 20-point score adjustments for missed Autonomous bonus for correct level of preloaded block.*

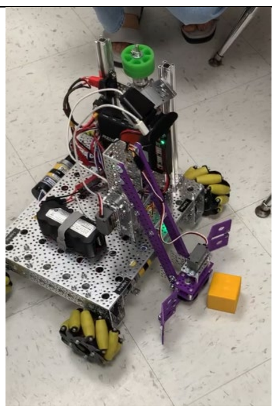
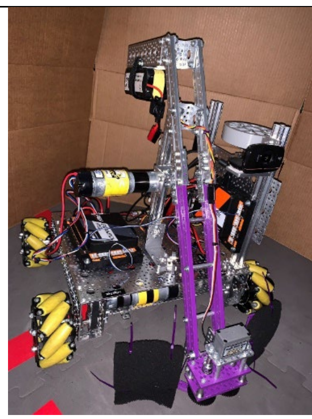
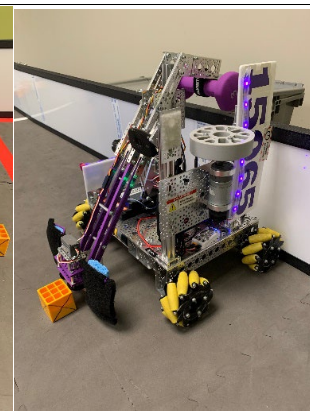
Engineering Overview

"ABI"

Overall design

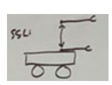
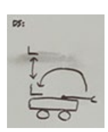
Our robot "Abi" has 5 major components:

Chassis – for movement
Gripper – for grabbing the freight/cargo/elements
Lifter – for raising the freight/cargo/elements into the shipping hubs
Spinner – for spinning the carousel to deliver the ducks to the field
Webcam – for detecting the duck or team shipping element location

Robot Meet 1	Robot Meet 2	Robot Meet 3
		 

Robot Mechanisms Trades

Since the chassis design is a reuse from the prior season and can meet the needs of going over the barrier, we focused our first build meeting on brainstorming 3 main concepts for constructing robot mechanisms to perform the game: SSA, SSL, and DS.

	Info	Pros	Cons
SSA (Same Side Angular) **[Chosen]** 	Gripper and lifter on same side, lifter moves up and back	• Easy • Light • Easily transition to DS later	Slow Less height
SSL (Same Side Linear) 	Gripper and lifter on different side, lifter moves directly up and down	Easy (relative to DS) Can lift high More precise Can cap	Heavy Slow
DS (Different Side) [Planned for future] 	Gripper and lifter on opposite sides, with gripper moving back and dropping object, and lifter moving up and down	Quickest	Complex

CHASSIS

The chassis design uses 4- 312 rpm motors with mecanum 100mm diameter wheels. We used this design in the prior season. The motors are mounted horizontally and staggered to allow for a smaller area chassis. The chassis is approximately 17" wide x 13" deep (which is narrow enough to allow us to strafe between the game barrier and the field perimeter into the warehouse. The base chassis frame is also 3" from the ground allowing enough clearance for the robot to go over the barrier.

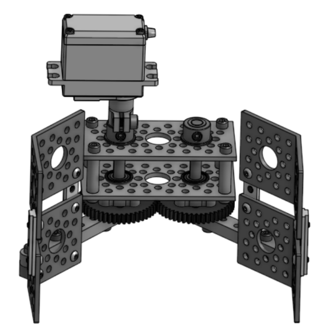
GRIPPER

Design 1:

Our first design was entirely made of metal and was composed of 4 small metal plates slightly bent that were connected with gears. We liked the overall design, however the metal caused it to be too heavy for the servo. The gripper did not have a firm hold on the freight, even when we added sponges to the metal plates. The balls and blocks continued to occasionally slip.

Design 2:

We replaced the metal with similar plastic parts and changed the gears for press-fit gears. However, the press-fit gears caused the axle and ball bearings to slip, and it would affect the gripper when we tried to open and close it while grabbing the freight.



Gripper Design 2 & 3 CAD

Design 3:

The plastic parts and foam on the gripper remained but needed the extra support picking up the freight. We added grip tape and put extra sponges between the foam and the gripper plate.

Design 4:

During the 2nd meet we found the gripper was pushing the freight/cargo around when we tried to grab them using a claw-like motion. We thought a more parallel grabbing action would be more accurate with picking up the freight so we designed a new gripper using the a similar 4-bar mechanism as the lifter. However, this design ended up not working because it took too much time to develop.

LIFTER

Design 1:

The first design was made of a simple Actobotics U-Channel rotated by a high torque servo.

Lifter Design 1

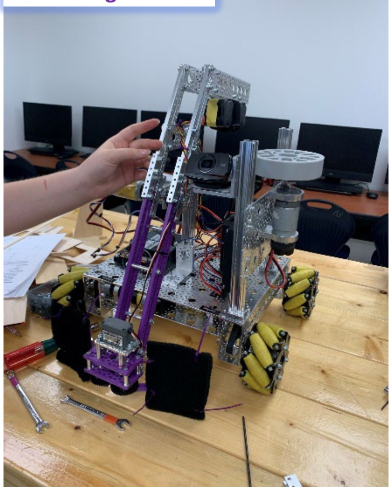
Design 2:

This design utilized a 4-bar lifter, and was composed of metal bars, which were initially our original plan. However, the metal bars made the lifter too heavy for the servo motor we were using. We calculated the weight that the servo could handle, using a torque equation, and found it to be less than a pound, so we had to redesign for lighter weight lifting and gripping mechanisms.

Design 3:

This design consisted of the same design as prototype 2 but we replaced the metal parts with 3-D printed plastic parts. Even with the plastic parts we could not get the weight to be lower than a pound so we switched the servo to regular DC motor.

Lifter Design 4



Design 4:

Design 4 was similar to designs 2 and 3 and is composed of the plastic 4-bar lifter. We also added a counterweight using our battery as the weight. We also extended the length of lifter arm so we could reach higher to cap with our team shipping element.

Lifter Design 4 CAD

Design 5:

Design 5 included replacing the higher speed motor with a slower motor (84 rpm vs 223 rpm), high torque motor with the addition of a potentiometer to support more accurate control of the arm position for autonomous modes. We also added 3 lb. dumbbell for safer counterweight to ensure arm is balanced.

SPINNER

Design 1

We used a soft 2” diameter wheel mounted on a vertical motor at the height of the carousel.

Design 2

During Meet 1 we found the small wheel was not always able to reliably spin the carousel so we replaced it with a larger 4” diameter compliant wheel that has been much more reliable.

WEBCAM

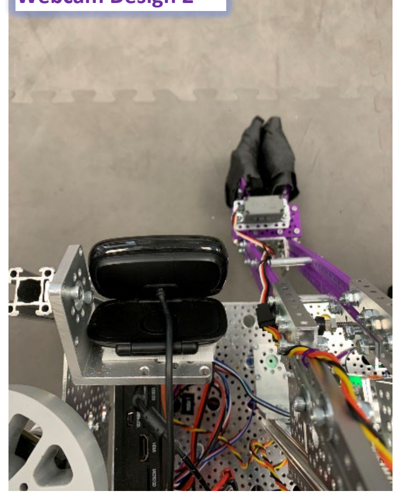
Design 1

We started with a CODI webcam mounted on an angle bracket which as angled down to see only the barcode on the field.

Design 2

We replace the CODI webcam with a Logitech webcam. Through trial and error, we found the CODI camera was not reliable in detecting the game elements. We decided to use the recommended Logitech camera and it has been much more reliable.

Webcam Design 2



TEAM SHIPPING ELEMENT

We wanted a shipping element that could be picked up by the robot and capped to the alliance hub in any direction.

Design 1

We designed a 4" cube with holes on all the sides. We used 4" to meet design criteria in the rules. We 3-D printed in pieces and glued it together with superglue. The walls of the cube were thick. It was a good concept and we used in Meet 2 but found it to be too big to pick up reliably and we were delivering to the field in endgame, and it broke during practice.

Design 2

We re-designed to make a lighter width and narrower, 3"x3"x4", rectangular element. We left the top and bottom open, had holes on the 4 sides and a surface in the middle to allow the element to hang from the alliance hub pole. We prototyped it with cardboard and then 3-D printed it. It is much thinner than design 1 but it is too thin, it has cracked several times and will need to be thickened to be sturdier for the final events.

Strategy

AUTONOMOUS

Our robot is programmed to use the webcam to locate the duck or team shipping element on the barcode, storing its location. Then it moves to the carousel, delivering a duck onto the playing field. It then places the pre-loaded box onto the shipping hub using the stored level, and finally parking completely inside the warehouse.

DRIVER-CONTROLLED PERIOD

Two controllers are used for this period, one controls the forward/backward/strafe movements, the other controls the gripper, lifter and spinner. After grabbing the freight, the robot moved through the small 13.7" space between the wall of the playing field and the boundary, sideways.

The main objective is to get as much freight on the third level of the shipping hub and try to keep the hub as balanced as possible.

ENDGAME

We focus on delivering as many ducks as possible from the carousel onto the playing field, and if fast enough moving to park in the warehouse. While we have built the capability to cap our team element, we have found we don't have enough time in endgame to quickly cap so we attempt to get to the warehouse to get more points.

Software

Software is developed in Java using Android Studio (Arctic Fox). The code for Freight Frenzy is version controlled using GitHub. Add link:

Using the standard FTC architecture, the core classes developed by the team are divided in to 4 groups: **Hardware Control , Autonomous, Teleop** and **Test/Development**.

HARDWARE CONTROL CLASSES

HardwareAbiBot:

ABIMU:

Wrapper for the IMU. It adds utility functions for easy access to the heading and position data from the IMU to be used in AutonomousOpmodes

Gripper:

Wrapper for the gripper servo control. Adds easy access methods for opening and closing the gripper, using Auto and Telop Opmodes

Spinner:

Wrapper for the spinner motor control. Adds easy access methods for spinning the wheel to turn the carousel, used in Auto and Telop Opmodes

Lifter:

Wrapper for the lifter arm motor and the potentiometer (v3 addition). It includes a **self-developed PI (Proportional-Integral) controller for the arm** and adds easy access to the angle of the arm from the potentiometer data. Pot added to have more reliable measurement of arm position vs motor encoder.

Measured values from the potentiometer to set lifter alliance hub level heights:

Hub Level	Pot Voltage
Capping	1.17V
Level 3	1.48V
Level 2	1.85V
Level 1	2.35V
Ground / Init	2.85V

PI Controller Tuned Coefficients

Autonomous Kp = 0.2; Ki = 0.15;   **Teleop** Kp = 0.5; Ki = 0.1

Chassis:

Class that contains the control of the 4 drive train motors. Includes the movement methods:

`encoderDrive()`: a key method to control movement. Inputs the 4 wheel's distances (based on wheel circumference) and speed. Uses these parameters and the motor encoder data to drive the wheels using the FTC methods setTargetPosition() and setMode() and setPower().
`encoderForward()`, `encoderStrafe()`, `encoderTranslate()` all use encoderDrive() and are only used in Autonomous OpModes
`imuRotateAbsolute()` drives the robot using the imu data from the ABIMU class and custom proportional controller
`imuRotate()` is derived from imuRotateAbsolute() to use relative angles rather than absolute angles.
Both rotate methods are only used in Autonomous OpModes.
`teleopMove()` does standard Teleop movement of the drive train based on gamepad input. Controls shown in Figure TBD.

AUTONOMOUS V1.0

AutonomousTest()

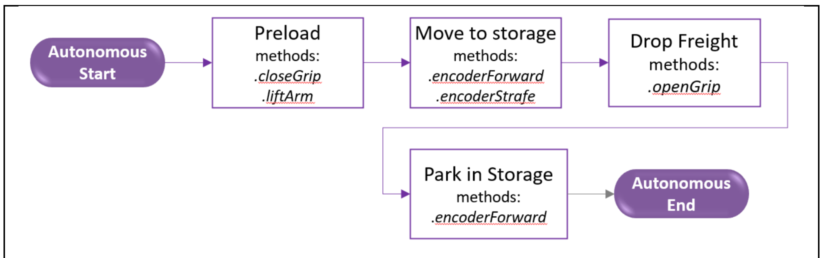


Figure 1 Autonomous V1.0 Flow Diagram

The initial autonomous code was developed to run a high-point strategy but just before the matches started the robot was unable to turn reliably to the correct angle toward the alliance shipping hub, so the code was removed and parking in the storage was added to ensure some autonomous points were scored and avoiding penalties.

TELEOP V1.0

FullTeleop()

Figure 2 Teleop Controls V1.0

FullTeleop is the mode used for the drivers to control the robot during Teleop/endgame. All strategy methods are working for a full Teleop scoring run. The driver controls are shown in Figure 2.

Turbo Mode: The Right Trigger allows the driver to run the robot drive train at full speed of the motors when needed. The regular wheel speeds operate and 50% power.

Driving: With the mecanum wheel chassis, the regular drive train operation allows the driver to strafe the robot, which is used for faster access in and out of the warehouse.

AUTONOMOUS V2.0

AutonomousDetect()

With more time to tune autonomous, the IMU code was updated to perform better angle measurements to turn toward the warehouse. Necessary adjustments to tune the included PID controller to control the arm levels were made (this also affected TeleOp).

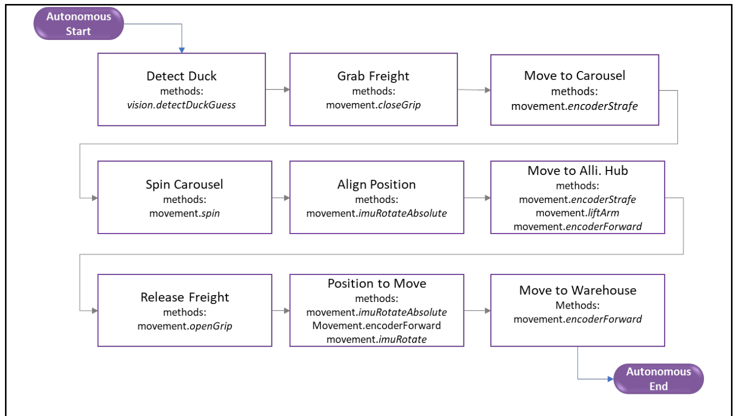


Figure 3 Autonomous V2.0 Flow Diagram

TELEOP V2.0

FullTeleopShourya()

Minor updates to Teleop were made to add the Slow Mode (left trigger) to allow for smoother driving for less experienced drivers. Button controls modified for specific drivers for their ease of use.

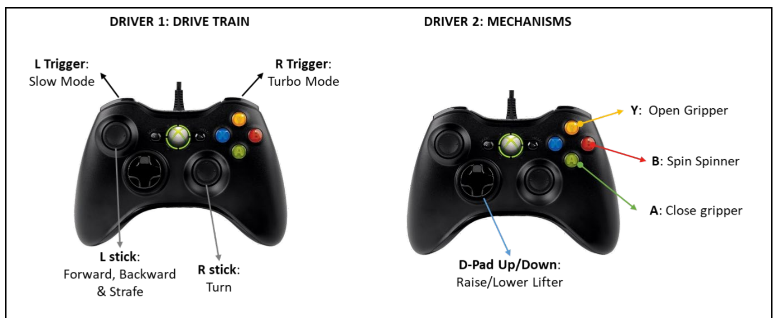


Figure 4 Teleop Controls V2.0

AUTONOMOUS V3.0

AutonomousDetectPurplePlus()

Telemetry added which allows the driver/operator to see progress throughout the autonomous run as well as the pre-load box detection level during Init.

LEDs also added to the display to the drive team which level was detected from the barcode. LEDs are

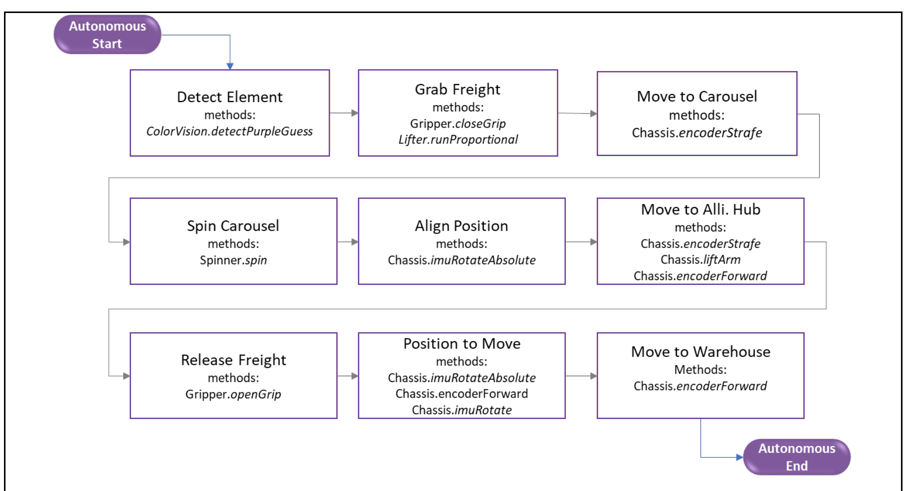


Figure 5 Autonomous V3.0 Flow Diagram

normally purple but will flash up to 3Hz in yellow with the flash count being the detected barcode level.

Additional Autonomous Runtime modes were added to allow operation on the Blue Alliance side and setup nearest the warehouse.

Major portions of the code are:

`Initialize()`

Performs the initialization of the robot (Hardware) and vision class; sets the lifter to default position and begins the detection of team shipping element.

Sets the LED pattern to identify the level of the shipping element detection for use later

`InitLoop()`

Updates the LED pattern, causing it to flash

Run until Start button is pressed

`run()`

Executes autonomous movement to score points see diagram Fig TBD

Vision class

For Meet 2 we used **AutonomousVision()** which used the FTC trained Machine Learning models to detect the duck.

Using the as-is FTC code to detect the custom team shipping element was not reliable so a new more efficient method was developed, the **AutonomousColorVision()** class. Major methods in this vision class are:

`cameraOpen()` sets up camera to interface to the software

`initializeFrameQueue()` setups of the queue to hold frame data

`startCamera()` populates the frameQueue

`purpleLoc()` Grabs a single frame, split into Left, Center, Right sectors. For each sector, iterates through each pixel counting pixels that fall into the color sphere of interest using RGB space. Checks which sector has the highest count of purple pixels and returns that section location. Purple shipping element RGB: 127, 62, 180

`detectPurple()` poll from the queue the current frame, then call purpleLoc on the frame, incrementing a count based on whichever sector is detected. This is then repeated for a input number of seconds. The sector with the highest count is returned to the calling class.

The counts are displayed in the telemetry data, which is helpful during testing and match setup of the robot.

Other robot Teleop mechanisms shown in figure above are executed in the specific Teleop mode classes using the Gripper, Lifter, and Spinner wrappers.

TELEOP V3.0

Update to TeleOp is consistent with changes made in Autonomous to use the new PI controller to move the lifter/arm to specific locations and allow for smooth movement of the arm.

The arm controls on the gamepad were modified to allow specific levels to be chosen using the d-pad.

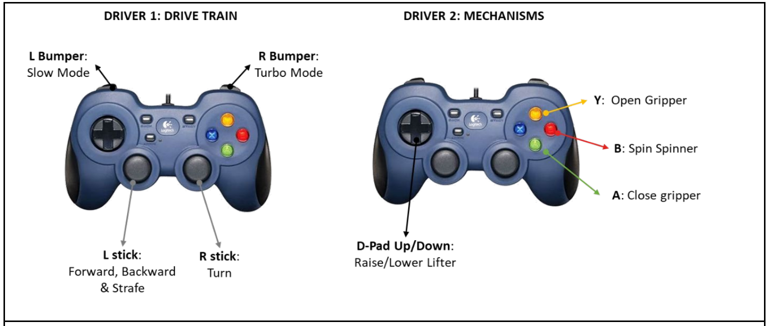


Figure 6 Teleop Controls V3.0

TEST / DEVELOPMENT

LifterTesting()- code to allow the user to adjust values for the gripper and the arm to ensure heights and widths are where needed before committing into the code.

LightTesting() – code to test operation of the gripper

ColorVisionTesting() – code to set parameters and verify **AutonomousColorVision()** is tuned, used when lighting conditions change.