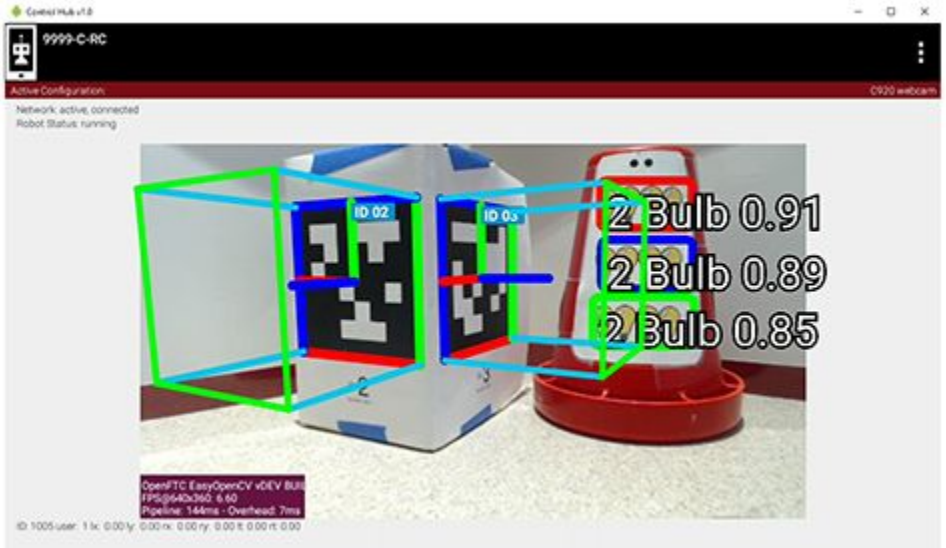
Vision Portal Workshop

Vision Portal

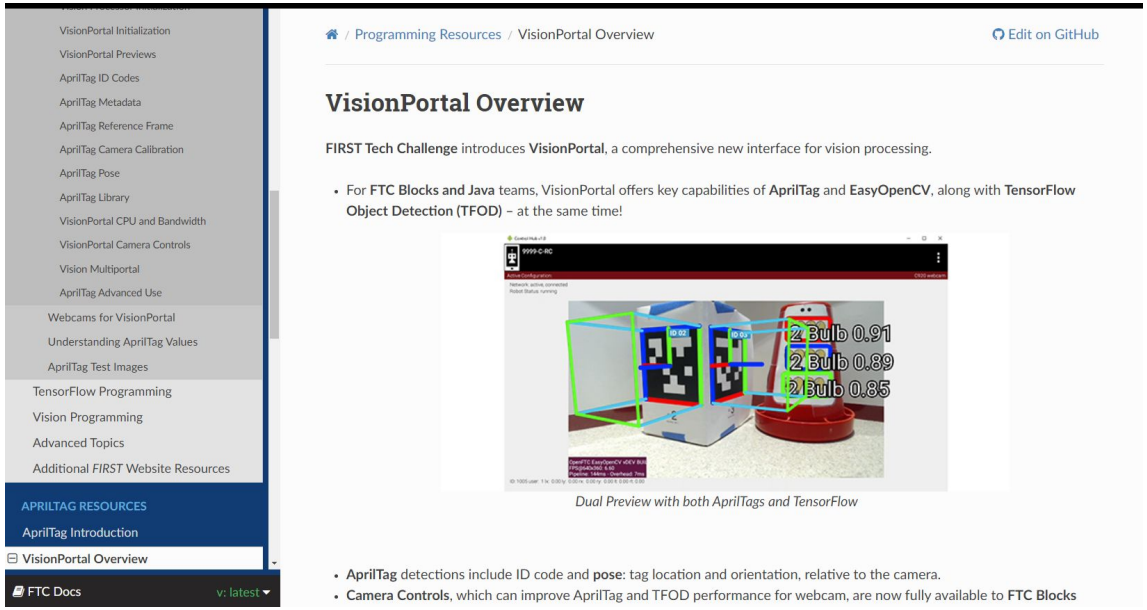
Vision Portal is a comprehensive new interface for vision processing introduced this year.



April Tags
TensorFlow
Webcam Controls

FTC Docs

FTC Docs has step-by-step instructions on how to use the Vision Portal.



https://ftc-docs.firstinspires.org/en/latest/apriltag/vision_portal/visionportal_overview/visionportal-overview.html

Webcams

	Logitech C270 HD	Logitech C920 HD Pro	Depstech DW49
			
Diagonal Angle of View:	55°	78°	80°
Max. Resolution:	1280x720	1920x1080	3840 x 2160

FIRST Tech Challenge Computer Vision

FIRST designs the game to include features that lend themselves to computer vision processing.

In CENTERSTAGE computer vision can be used on

- Pixels
- Team Props
- April Tags on the front perimeter
- April Tags on each backdrop

Centerstage Autonomous

Navigating:

Parked In Alliance Backstage: ..... 5 points

Randomization Tasks based on white Pixel:

Purple Pixel in Spike Mark location: .....10 points

Yellow Pixel in correct column on Backdrop: .....10 points

Randomization Tasks based on Team Art:

Purple Pixel in Spike Mark location: .....20 points

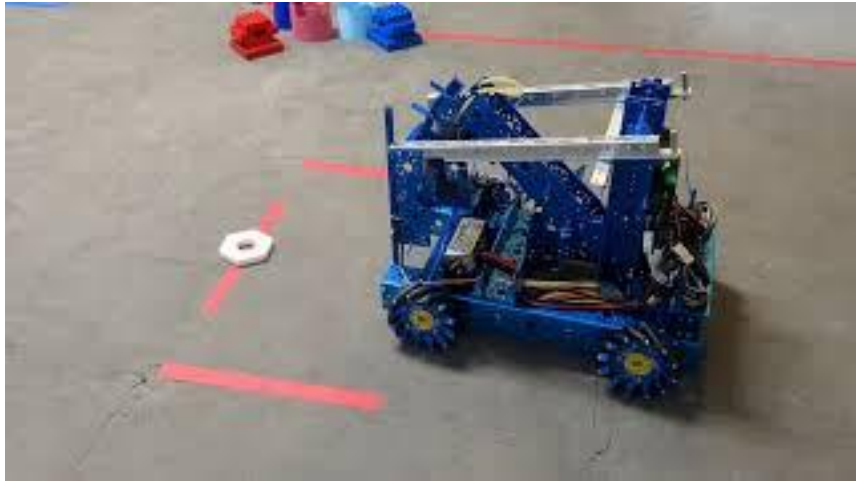
Yellow Pixel in correct column on Backdrop: .....20 points

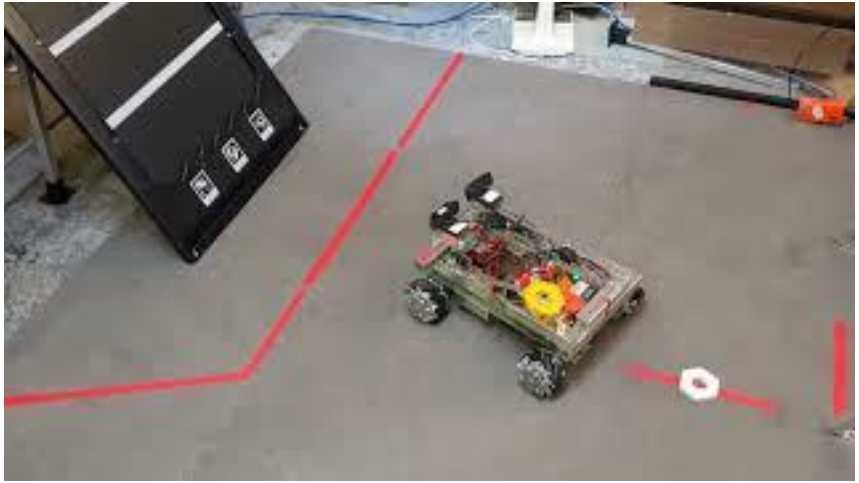
Pixels:

Placed in Backstage: ..... 3 points

Placed on Backdrop: ..... 5 points

Autonomous Programs





The FTC Software Development Kit (SDK)

The FTC SDK is the software that runs on the Robot Controller.

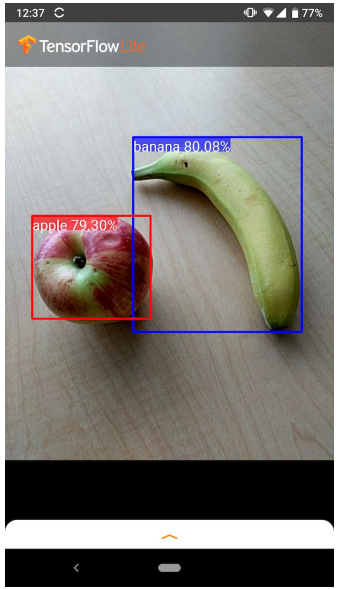
Vision Processing features of the SDK:

1. TensorFlow
2. April Tags
3. Camera controls

Java programmers can use EasyOpenCV.

TensorFlow Concepts

Google created a machine learning software library called TensorFlow.



To use TensorFlow:

- Create a set of images with objects
- Put a label on bounding box on game objects
- Train the model to recognize game objects

TensorFlow Lite is designed to run on smaller devices like Phones or the android device in a Control Hub.

TensorFlow Inputs

FIRST has provided a TensorFlow model that recognizes the Pixel in this year's game.

TensorFlow Inputs:

- confidence level, default 75%
- cropping/zoom
- model to use

https://ftc-docs.firstinspires.org/en/latest/programming_resources/index.html#tensorflow-programming

TensorFlow Outputs

Camera Stream preview showing TensorFlow detection of a pixel

TensorFlow Outputs:

List of objects, each object will have

- Label e.g. Pixel
- bounding box
- confidence percentage

TensorFlow for CENTERSTAGE

By default TensorFlow can detect the pixel from above

https://ftc-docs.firstinspires.org/en/latest/programming_resources/vision/tensorflow_cs_2023/tensorflow-cs-2023.html

TensorFlow Advantages

- TensorFlow detects objects
  - in spite of different backgrounds
  - in varied lighting conditions
  - in varied orientation

- TensorFlow can distinguish between similar looking (but still distinct) objects

TensorFlow Disadvantages

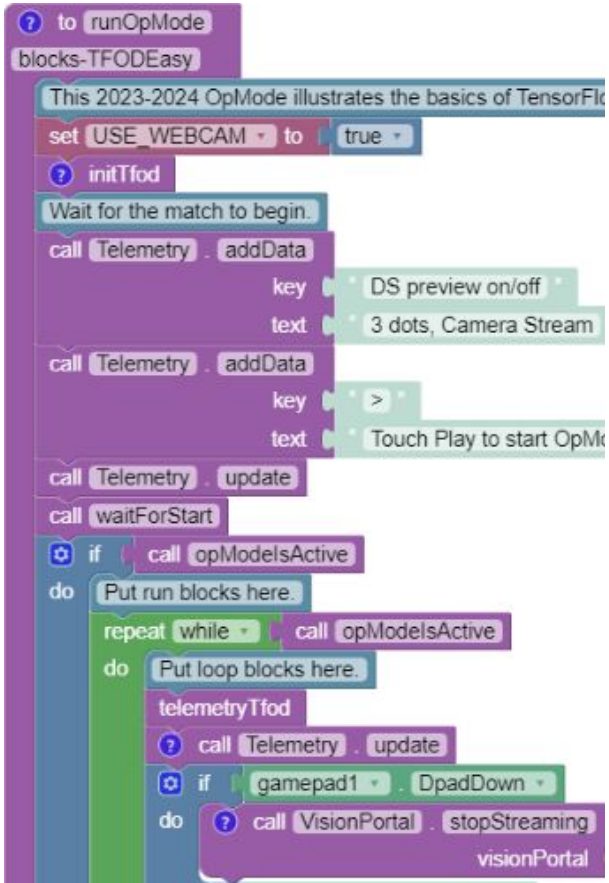
- Training a TensorFlow model seems daunting
- TensorFlow is computationally intensive and has a low detection rate
- TensorFlow only detects objects

SDK TensorFlow Sample Blocks Programs

- **ConceptTensorFlowObjectDetectionEasy** - easy to use sample program that displays what objects are detected.
- **ConceptTensorFlowObjectDetection** - sample program with comments that show how to enable or set various parameters
- **ConceptTensorFlowObjectDetectionCustomModel** - sample program where you provide your own TensorFlow model.

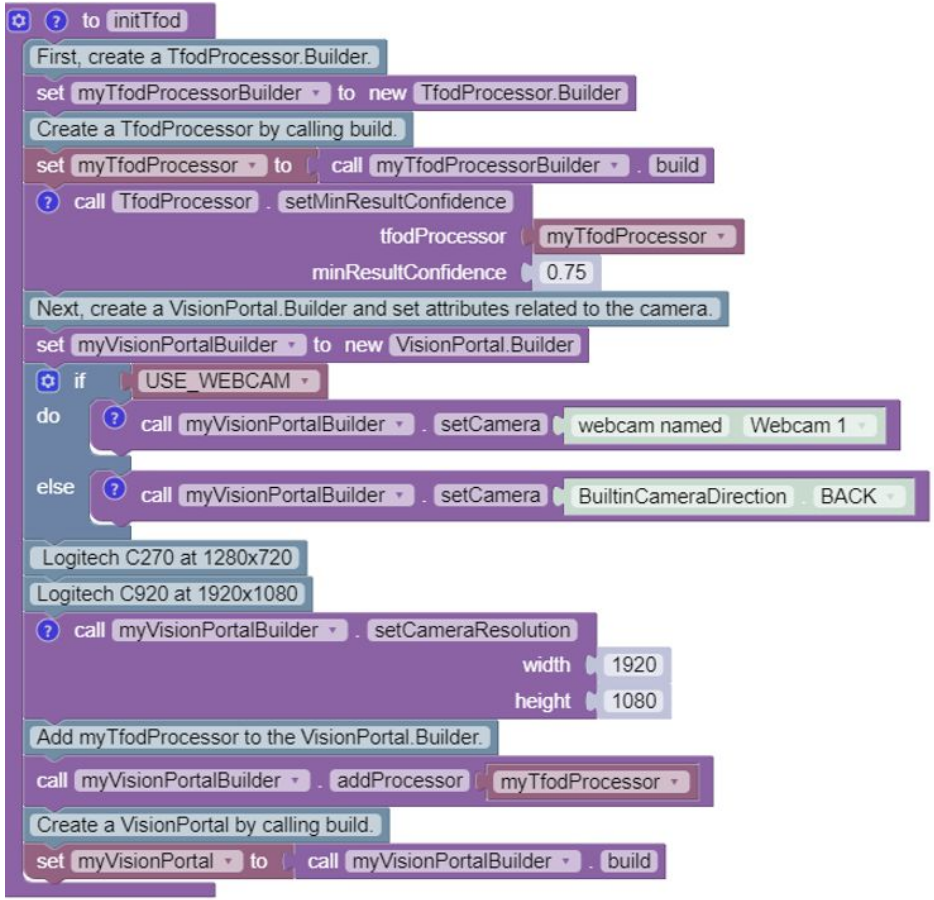
TensorFlow Easy (Blocks)

Lists know objects (i.e. pixels) detected in the image from the webcam



blocks-TFOD

Allows user to set camera resolution and other parameters.



blocks-TFODcustom

Sample program using custom Tensorflow model.

Used for Team Props.

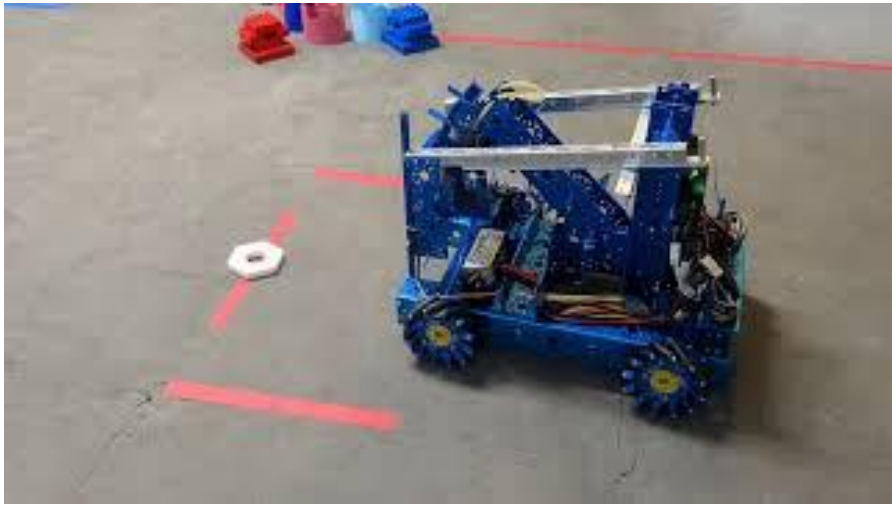
```
to initTfod
  First, create a TfodProcessor.Builder.
  set myTfodProcessorBuilder to new TfodProcessor.Builder
  call myTfodProcessorBuilder . setModelFileName "redBlueDuplo.tflite"
  call myTfodProcessorBuilder . setModelLabels create list with "blueDuplo"
                                                                "redDuplo"
  Create a TfodProcessor by calling build.
  set myTfodProcessor to call myTfodProcessorBuilder . build
  call TfodProcessor . setMinResultConfidence
                 tfodProcessor myTfodProcessor
                 minResultConfidence 0.6
  Next, create a VisionPortal.Builder and set attributes related to the camera.
  set myVisionPortalBuilder to new VisionPortal.Builder
  if USE_WEBCAM
  do call myVisionPortalBuilder . setCamera webcam named Webcam 2
  else call myVisionPortalBuilder . setCamera BuiltinCameraDirection BACK
  call myVisionPortalBuilder . setCameraResolution
                 width 1920
                 height 1080
  Add myTfodProcessor to the VisionPortal.Builder.
  call myVisionPortalBuilder . addProcessor myTfodProcessor
  Create a VisionPortal by calling build.
  set myVisionPortal to call myVisionPortalBuilder . build
```

blocks-RobotAutoDriveTFODpixel

https://github.com/acharraggi/Centerstage-Blocks

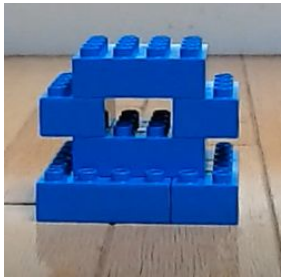
TensorFlow Autonomous

blocks-RobotAutoDriveTFODpixel



Team Prop

See section 7.4 Team Game Element Construction Rules in the game manual part 1

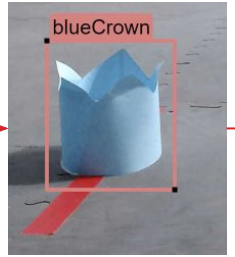


https://ftc-docs.firstinspires.org/en/latest/programming_resources/vision/tensorflow_cs_2023/tensorflow-cs-2023.html

TensorFlow *FIRST* Machine Learning Toolchain (FTC-ML)

Create Videos

Upload To FTC-ML



Label Objects

TensorFlow

Create Model

Download and Program

https://ftc-docs.firstinspires.org/en/latest/ftc_ml/index.html

FTC-ML Videos

- One hour to set up and take videos.
- One hour to upload and the videos to the FTC ML website and label the Team Props
- FTC ML model training step will take another hour.

https://ftc-docs.firstinspires.org/en/latest/ftc_ml/managing_tool/create_videos/create-videos.html

TensorFlow Summary

TensorFlow can be a useful way to recognize objects on the field.

It's not good for navigating, April Tags are designed for that.

TensorFlow should do a good job recognizing Team Props at the start of the autonomous period.

Questions?

April Tag Concepts

Developed at the University of Michigan.

AprilTag is like a 2D barcode or a simplified QR Code.

Contains a numeric ID code and can be used for location and orientation.

Camera Ready

==== (ID 0) Nemo
XYZ -6.6 24.9 -5.7 (inch)
PRY 2.0 0.1 4.9 (deg)
RBE 25.7 14.8 -12.8 (inch, deg, deg)

==== (ID 1) Jonah
XYZ -1.5 25.5 -5.7 (inch)
PRY 0.7 -0.0 5.0 (deg)
RBE 25.6 3.3 -12.6 (inch, deg, deg)

key:
XYZ = X (Right), Y (Forward), Z (Up) dist.
PRY = Pitch, Roll & Yaw (XYZ Rotation)
RBE = Range, Bearing & Elevation

https://ftc-docs.firstinspires.org/en/latest/apriltag/vision_portal/apriltag_intro/apriltag-intro.html

April Tag Pose

"pose" is the combination of:

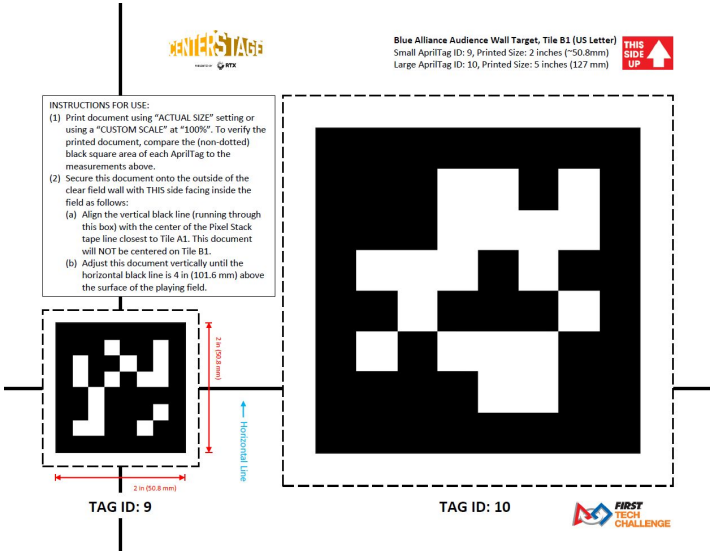
- relative position from the camera to April Tag; and
- orientation of the April Tag

The SDK also calculates Range, Bearing and Elevation

```
Camera Ready

==== (ID 0) Nemo
XYZ  -6.6  24.9  -5.7  (inch)
PRY   2.0   0.1   4.9  (deg)
RBE  25.7  14.8 -12.8  (inch, deg, deg)

==== (ID 1) Jonah
XYZ  -1.5  25.5  -5.7  (inch)
PRY   0.7  -0.0   5.0  (deg)
RBE  25.6   3.3 -12.6  (inch, deg, deg)

key:
XYZ = X (Right), Y (Forward), Z (Up) dist.
PRY = Pitch, Roll & Yaw (XYZ Rotation)
RBE = Range, Bearing & Elevation
```

https://ftc-docs.firstinspires.org/en/latest/apriltag/vision_portal/apriltag_pose/apriltag-pose.html

CENTERSTAGE April Tags



April Tag Advantages

- Fast detection rate
- Provides accurate, relative position information
- Is less prone to fluctuating or varied lighting conditions on the field.

April Tag Disadvantages

- The entire April Tag must be in the camera view
- April Tags must be included in the tag library
- Cameras require calibration data

April Tag Webcam Calibration

To provide good pose estimates, each webcam requires calibration data, for each specific resolution you use.

The SDK includes calibration for some webcams.

Logitech often creates new versions of their webcams with the same name but different firmware so the SDK can't recognize them.

C270 should be calibrated.

https://ftc-docs.firstinspires.org/en/latest/apriltag/vision_portal/apriltag_camera_calibration/apriltag-camera-calibration.html

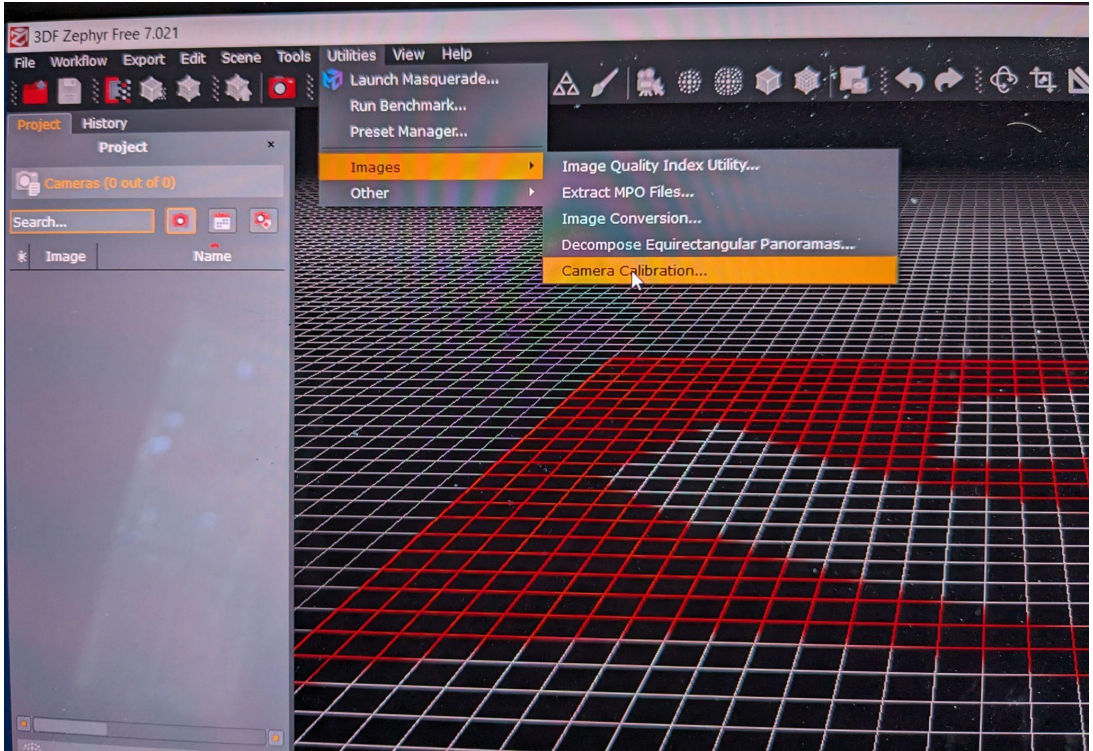
Calibrating with 3DF Zephyr

Allow about an hour to do this.

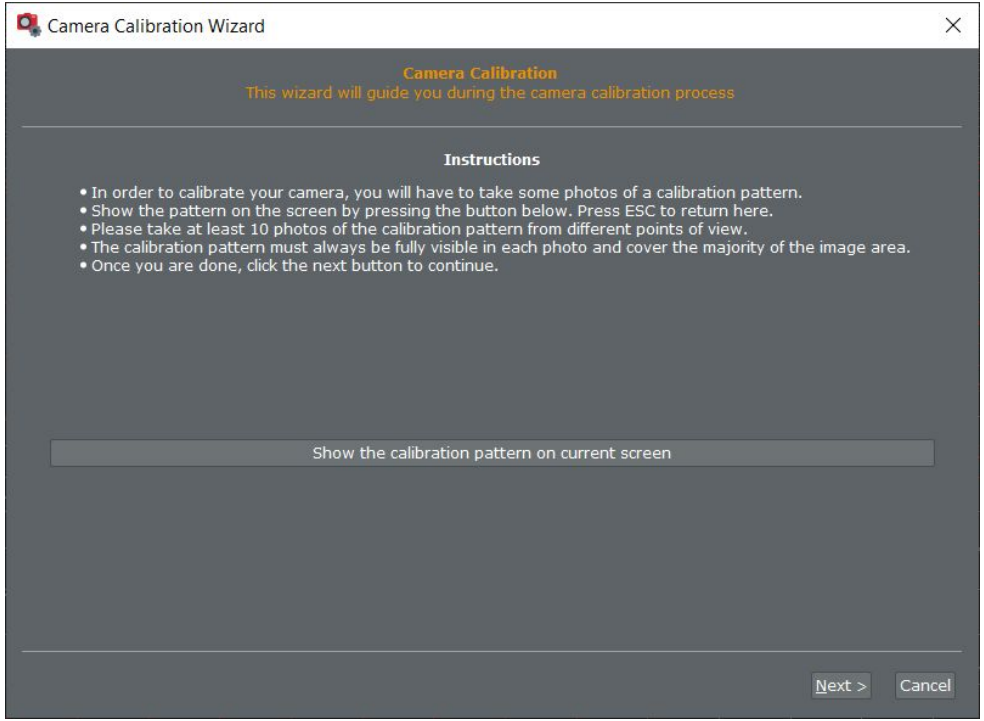
1. Download and install 3DF Zephyr Free Edition. https://www.3dflow.net/3df-zephyr-free/
2. Create an OpMode from the sample UtilityCameraFrameCapture
3. Use 3DF Zephyr to display the calibration target and take pictures
4. Copy the captured frames to your computer
5. Add the images to 3DF Zephyr
6. Run the calibration target analysis in 3DF Zephyr, save calibration values

https://ftc-docs.firstinspires.org/en/latest/programming_resources/vision/camera_calibration/camera-calibration.html

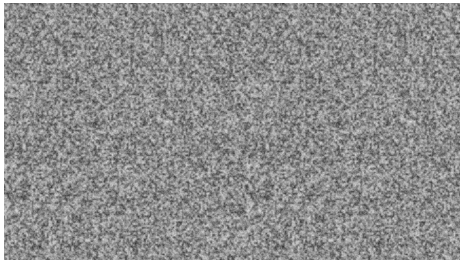
Start Calibration



Calibrating with 3DF Zephyr

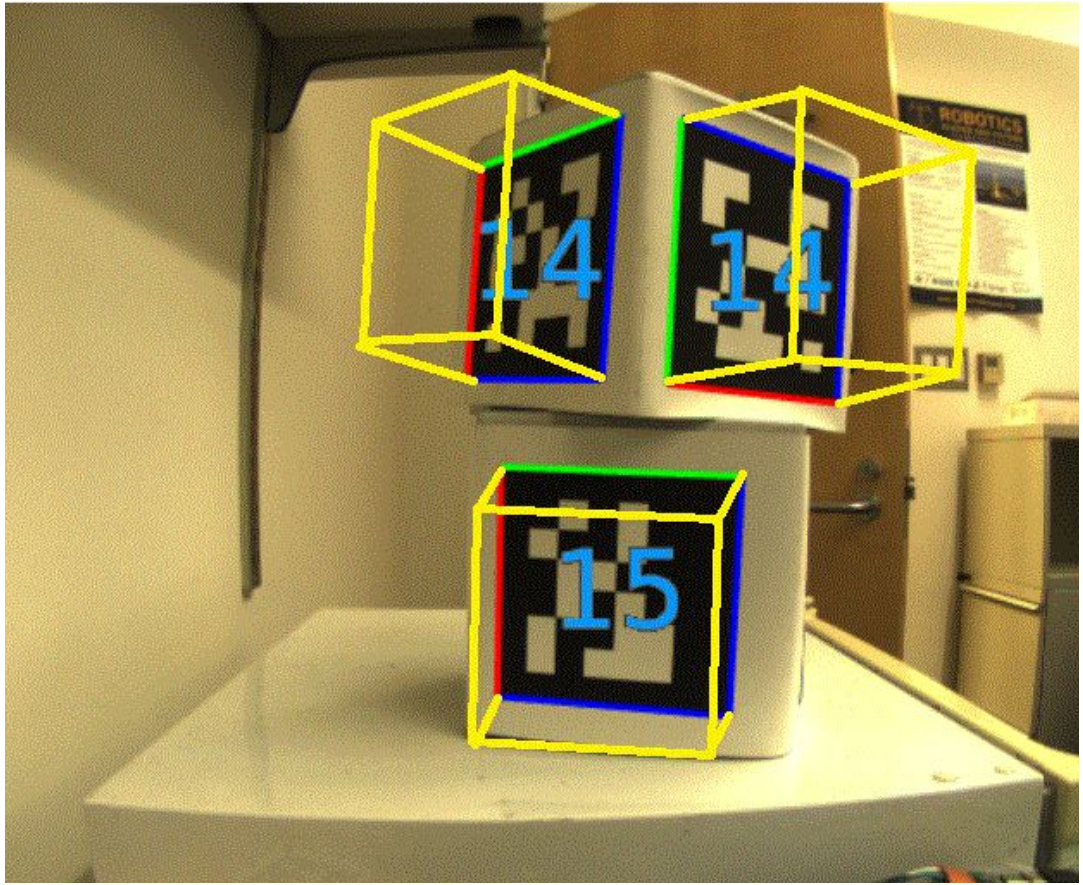


Calibration Target

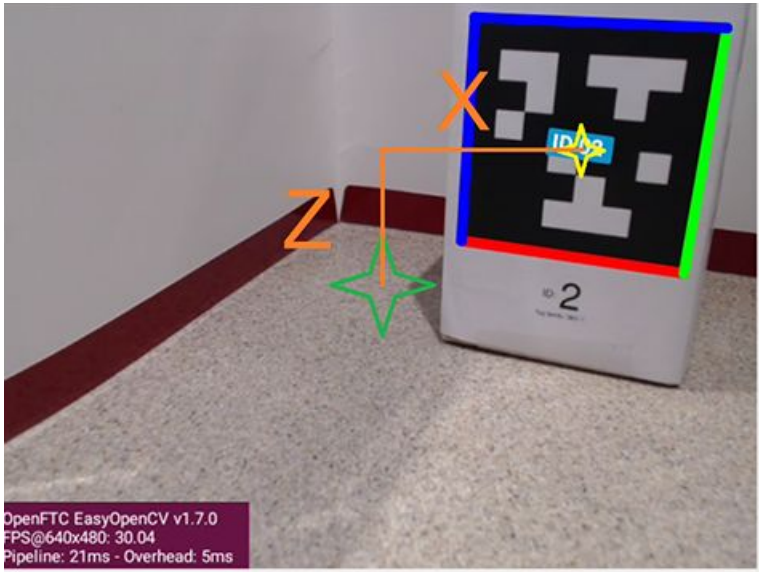


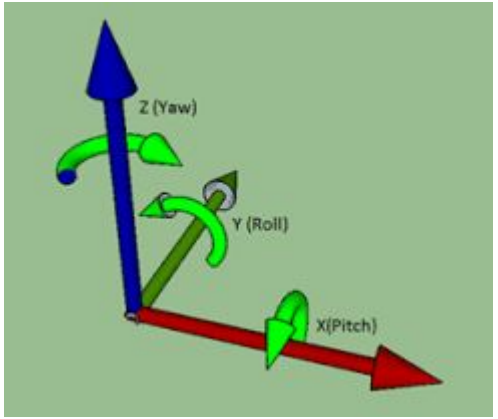
Take at least 10 captures using the Utility Camera Frame Capture program

April Tags



April Tag Coordinate System





The green star is the centre of the webcam image, the yellow star is at the April Tag centre.

SDK April Tag Programs

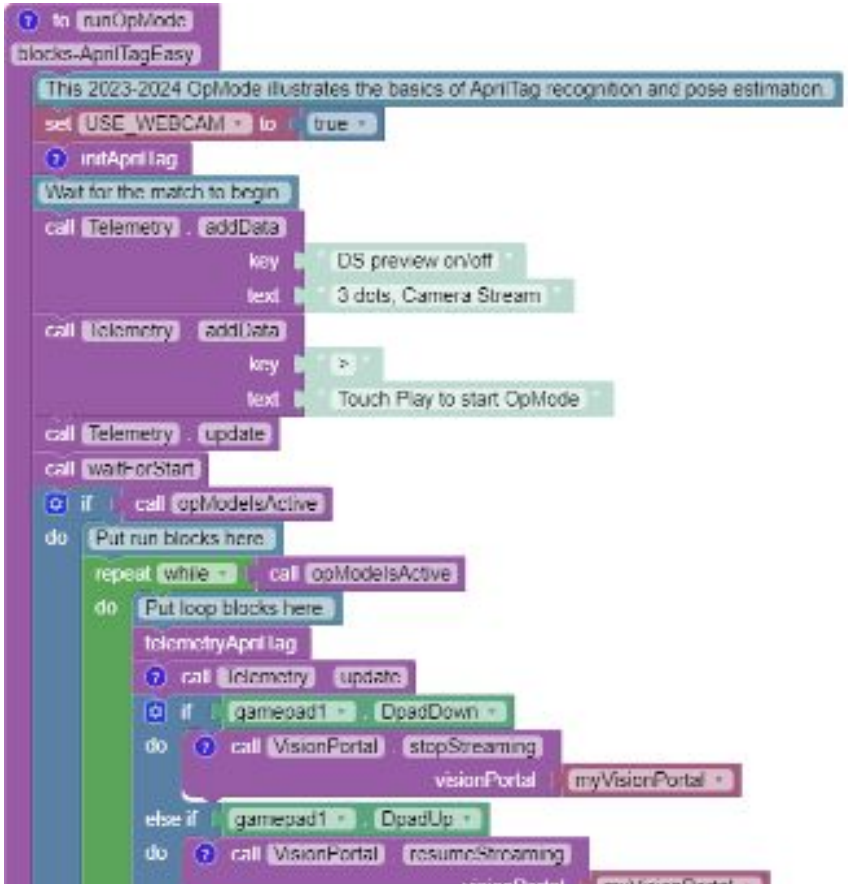
AprilTagEasy - start here

AprilTag

UtilityCameraFrameCapture - for calibration

ConceptAprilTagEasy

Detects and displays all April Tags in the camera view with their position information.



ConceptAprilTag

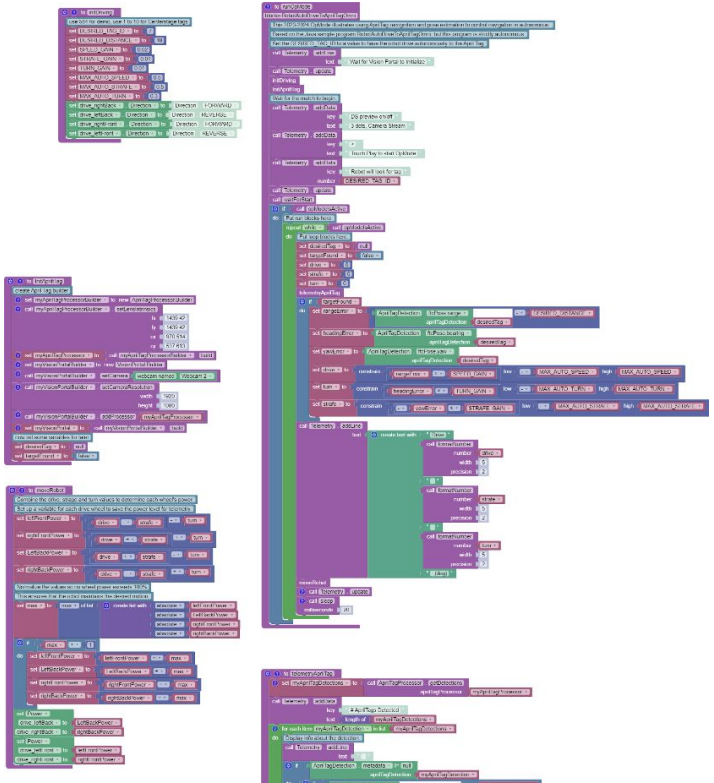
Similar but allows you to set calibration and resolution parameters.

```
to initAprilTag
  First, create an AprilTagProcessor.Builder.
  set myAprilTagProcessorBuilder to new AprilTagProcessor.Builder
  call myAprilTagProcessorBuilder . setLensIntrinsics
      fx 1439.42
      fy 1439.42
      cx 970.514
      cy 537.613
  call Telemetry . addLine
      text "Lens Intrinsics set for Logitech c920"
  Create an AprilTagProcessor by calling build.
  set myAprilTagProcessor to call myAprilTagProcessorBuilder . build
  Next, create a VisionPortal.Builder and set attributes related to the camera.
  set myVisionPortalBuilder to new VisionPortal.Builder
  if USE_WEBCAM
  do call myVisionPortalBuilder . setCamera webcam named Webcam 2
  else call myVisionPortalBuilder . setCamera BuiltinCameraDirection . BACK
  call myVisionPortalBuilder . setCameraResolution
      width 1920
      height 1080
  call Telemetry . addLine
      text "webcam resolution set to 1920x1080"
```

RobotAutoDriveToAprilTagOmni Program

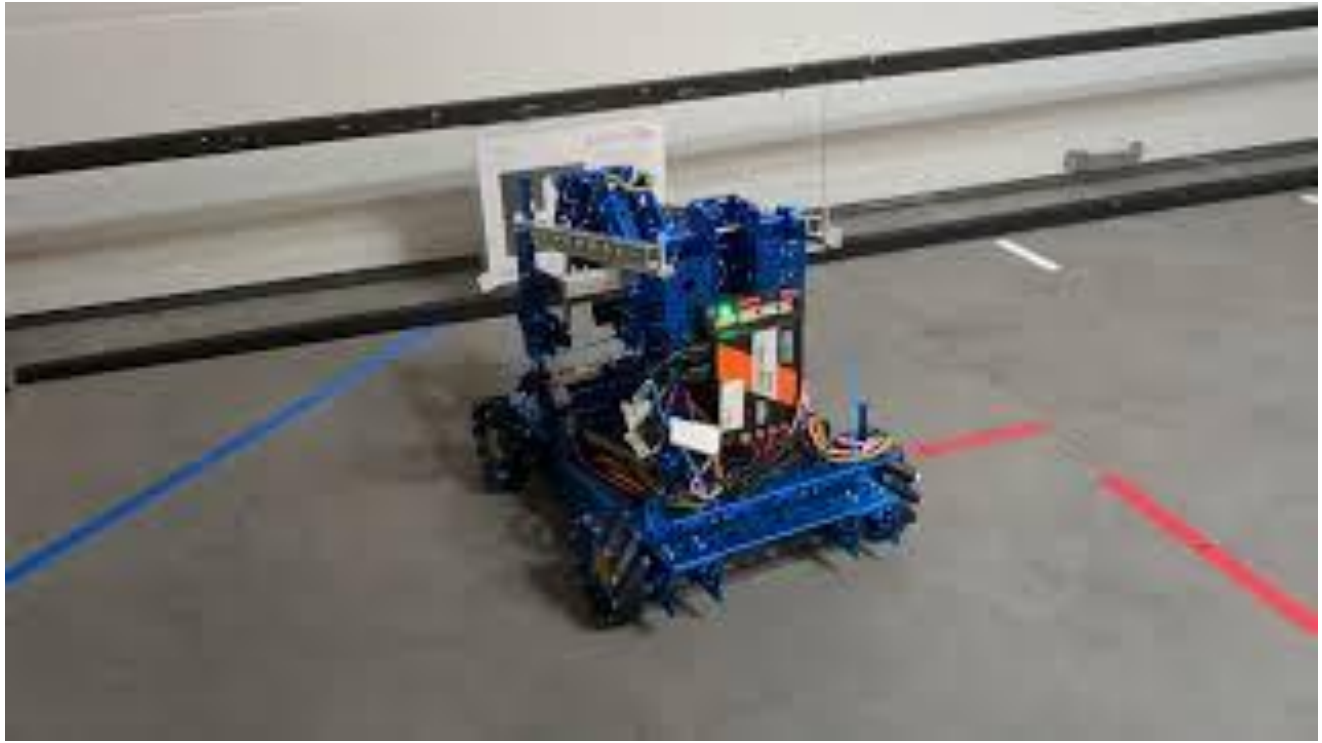
Combines April Tag Detection and Driving.

NOT a sample program.



https://github.com/acharraggi/Centerstage-Blocks

RobotAutoDriveToAprilTagOmni



April Tag Summary

An April Tag is like a 2D barcode or a simplified QR Code. It contains a numeric ID code and can be used for location and orientation.

With a calibrated camera and a tag of known size you get:

- X,Y,Z distances to the tag
- Roll, Pitch, Yaw rotations of the tag
- Range, Bearing and Elevation calculated values

The CENTERSTAGE field has tags at the front and back of the field.

Questions?

Vision Portal - Part 3 - Webcam Controls

Vision Portal also allows some control over webcams

- Exposure
- Gain
- Switch between two webcams
- Virtual pan, tilt, zoom function

Webcam Sample Programs

- **ConceptAprilTagOptimizeExposure.java** - adjust the exposure and gain values
- **Concept Double Vision** - control switching between TensorFlow and April Tag processing
- **April Tag Switchable Cameras (blocks)** - switch between 2 webcams
- **Concept TensorFlow Object Detection Switchable Cameras (blocks)**

Questions?

FTC Team Resources

Start here: https://firstroboticsbc.org/ftc/ftc-team-resources/

FIRST Robotics BC

About | Programs | Educators | How to Help | Events | Contact | Donate

FIRST TECH CHALLENGE

About FTC | Game & Season | Start a Team | Team Resources

FTC Team Resources

There's lots of information and help available for *FIRST* Tech Challenge teams, everything from FTC event results to Preparing for Competitions. Let us know if there are more resources you'd like added to this page, or you can contact us if you have any questions.

- FTC Workshops – videos and slideshows from past workshops.
- Managing a FTC Team – *FIRST* Team Management Resources page, Event Preparation, Finances/Fundraising, Mentors, including how to start a team.
- FTC Game Manuals and Resources – official game and season materials, updated to new season after game revealed.
- The official FIRST Tech Challenge Blog
- Official FTC Discussion Forums
- Official FTC Question and Answer System
- How to Create a Virtual Pit – Workshop on How to Create Virtual Pit on a platform like ZOOM.
- Preparing for Competition (including Robot Inspection Check Lists)
- Engineering Notebook Portfolio – required for judged awards at a competition.

Building FTC Robots

- Robot Building Resources – official *FIRST* robot building resources page.
- Robot Wiring Guide – highly recommended guide to wiring your robot. Teams should follow best practices when wiring their robots. This will help to ensure that the placement, connections, and security of their wires will lead to improved robot performance, eliminate intermittent electrical problems, and allow for easy troubleshooting and resolution of electrical and/or signal-

Sample programs at: https://github.com/acharraggi/Centerstage-Blocks

Appendix - extra material

1. Demonstration Robot
2. Known Vision Processing Issues
3. Java Autonomous Programs
4. Good Programming Practices
5. Programming Tools

A1 - Demonstration Robot StudicaBot2

This demonstration robot uses mecanum wheels and four motors. It's based on the Studica robot kit which we recommend.

- It has two webcams, one points down for Pixel detection, one points forward for April Tag detection.
- It has a small arm that can attempt to place the yellow pixel on the backdrop. It is not able to pick up pixels.
- It has a hanging arm that can be released and used to hang the robot
- It also has a drone launcher

A2 - Known Vision Processing Issues

https://github.com/FIRST-Tech-Challenge/FtcRobotController/issues

The GitHub repository is where issues with the Robot Controller can be found. There are a couple of vision related issues, one that is quite serious.

- loss of robot control due to internal vision error
  Note: this also results in a short driver hub disconnect. This appears to be an fairly frequent a problem running an autonomous program with vision processing
- when using switchable cameras, Gain control returns a NULL pointer

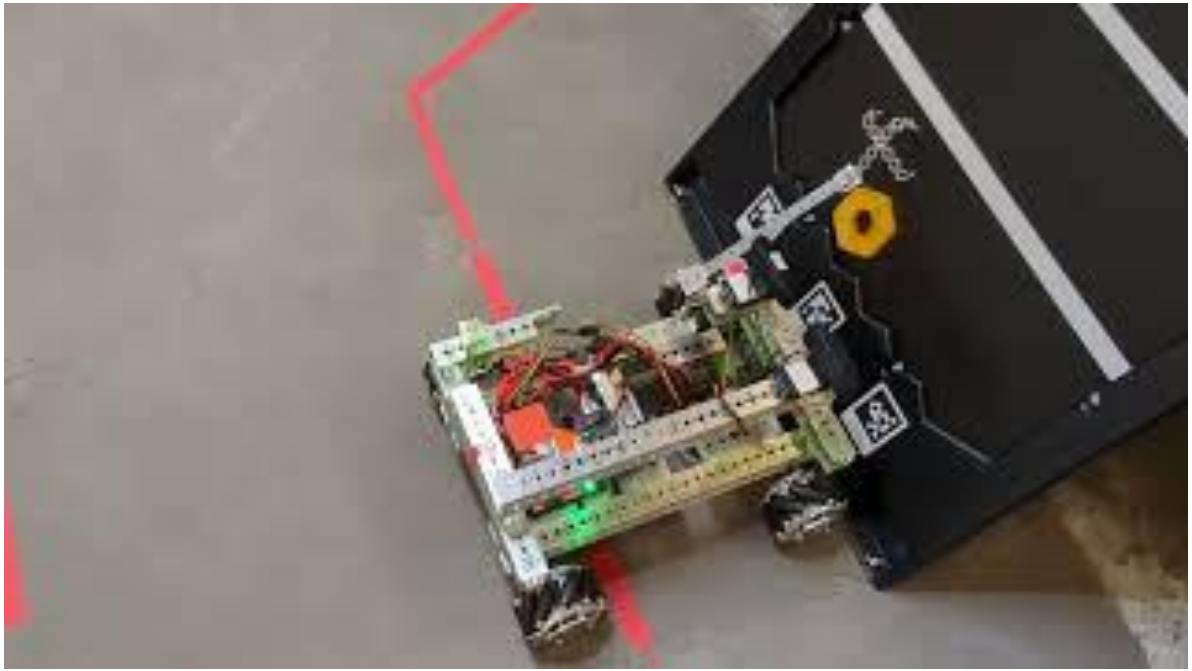
Autonomous Programs

CENTERSTAGE autonomous programs likely need to use a webcam and vision processing

- inspect the spike marks to determine which has the pixel or team prop
- navigate from your starting position to the backdrop
- if your quick enough doing that you might be able to go to the front wall and pick up a couple of pixels that you can score backstage

A3 - Java Autonomous Programs

AutoPixel1

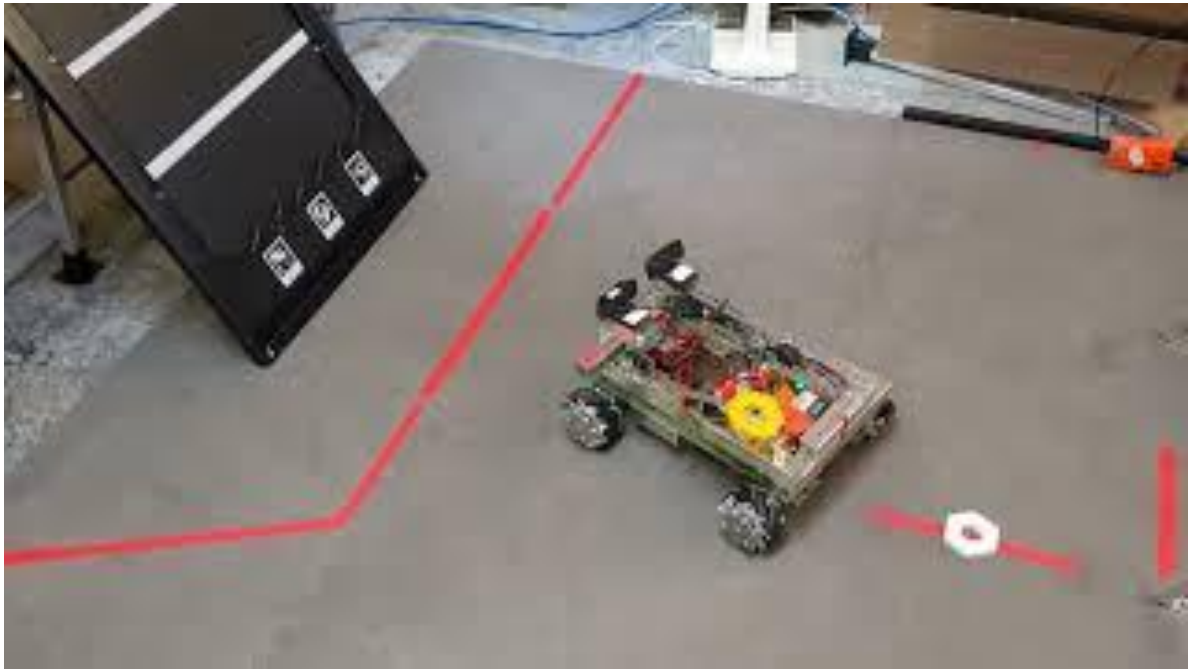


AutoPixel1

[show Java Program] - discuss structure

https://github.com/acharraggi/Centerstage-Samples/blob/main/AutoPixel1.java

AutoPixelFront



AutoPixelFront

[show Java Program] - discuss structure

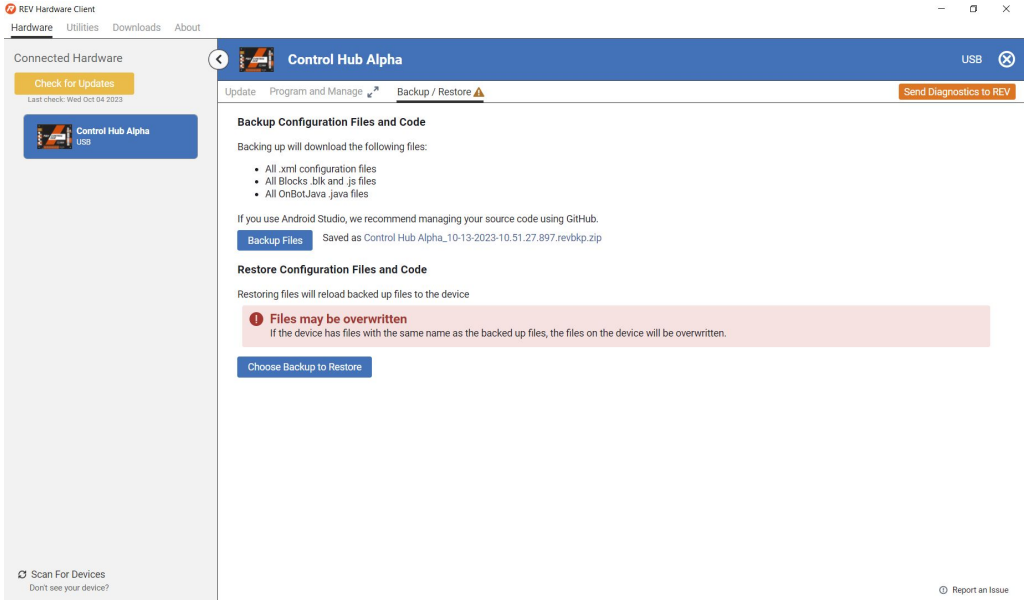
https://github.com/acharraggi/Centerstage-Samples/blob/main/AutoPixelFront.java

A4 - Good Programming Practices

- add comments
- save your program often, create backups or even versions.
- provide Telemetry to the driver station
- write to the Robot Controller log
- autonomous programs should always do something. Don't get stuck in a loop waiting for something that might not happen.

Backups

- Select Download to save each program in your PC's downloads folder.
- Use the Rev Hardware Client to save a zip file of all programs and your config files.



Telemetry

Initialization

- Display message indicating progress of Vision Portal or IMU start up
- You could display warnings e.g Voltage too low, or errors in initialization
- should indicate initialization is complete with a Ready message

Autonomous Telemetry

- You might want to indicate what step or state the program is in.
- probably key points in the autonomous program like where did it find the pixel or what April Tag is currently being tracked.

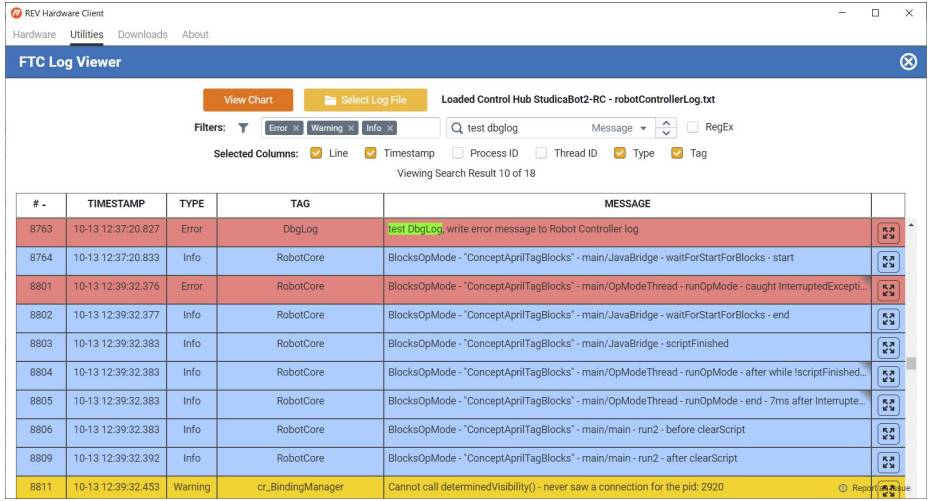
Driver Control Telemetry

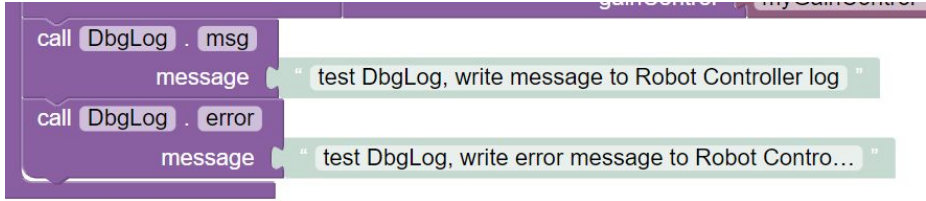
- You might want to indicate the position of an arm or gripper or whether or not you collector has picked up a game element.
- commonly things like the motor power levels are displayed

Telemetry

here's how to do telemetry in blocks

Robot Controller Log





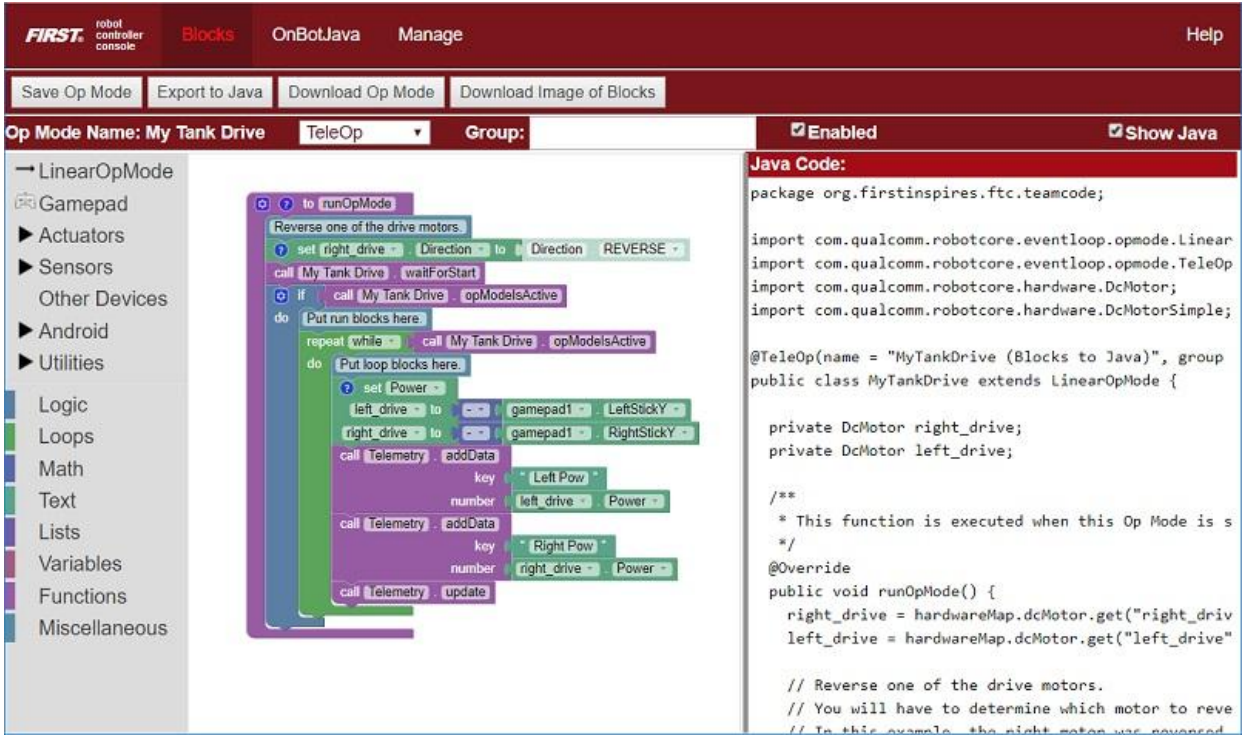
Rev Hardware Client - Log Viewer program

A5 - Programming Tools

The FTC robot controller is essentially an Android phone application. There are three programming tools provided:

1. The Blocks Programming Tool
2. The OnBot Java Programming Tool
3. Android Studio

The Blocks Programming Tool



The OnBot Java Programming Tool

FIRST robot controller console | Blocks | OnBotJava | Manage | Help

Project Files

- org.firstinspires.ftc.teamcode
  - MyTankDrive.java
  - SimpleOnBotIterative.java
  - SimpleOnBotLinear.java
  - testmotor.java

```
1  package org.firstinspires.ftc.teamcode;
2
3  import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
4  import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
5  import com.qualcomm.robotcore.hardware.DcMotor;
6  import com.qualcomm.robotcore.hardware.DcMotorSimple;
7
8  @TeleOp(name = "MyTankDrive (Blocks to Java)", group = "")
9  public class MyTankDrive extends LinearOpMode {
10
11   private DcMotor right_drive;
12   private DcMotor left_drive;
13
14   /**
15    * This function is executed when this Op Mode is selected fr
16    */
17   @Override
18   public void runOpMode() {
```

```
Build started at Mon Jun 17 2019 15:45:25 GMT-0400 (Eastern Daylight Time)


Build finished in 2.9 seconds
Build succeeded!
```

Android Studio

Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help 100% Sun 12:29 PM

PushbotTeleopPOV_Linear.java - ftc_app - [~/Documents/workspace/ftctechnh/ftc_app]

TeamCode

ftc_app › FtcRobotController › src › main › java › org › firstinspires › ftc › robotcontroller › external › samples › PushbotTeleopPOV_Linear

Android

- FtcRobotController
- TeamCode
  - manifests
  - java
    - org.firstinspires.ftc.teamcode
      - readme.md
  - jniLibs
  - res
- Gradle Scripts

PushbotTeleopPOV_Linear.java

PushbotTeleopPOV_Linear

```
 * All device access is managed through the HardwarePushbot class.
 * The code is structured as a LinearOpMode
 *
 * This particular OpMode executes a POV Game style Teleop for a PushBot
 * In this mode the left stick moves the robot FWD and back, the Right stick turns left and right.
 * It raises and lowers the claw using the Gampad Y and A buttons respectively.
 * It also opens and closes the claws slowly using the left and right Bumper buttons.
 *
 * Use Android Studios to Copy this Class, and Paste it into your team's code folder with a new name.
 * Remove or comment out the @Disabled line to add this opmode to the Driver Station OpMode list
 */

@TeleOp(name="Pushbot: Teleop POV", group="Pushbot")
@Disabled
public class PushbotTeleopPOV_Linear extends LinearOpMode {

    /* Declare OpMode members. */
    HardwarePushbot robot           = new HardwarePushbot();   // Use a Pushbot's hardware
                                                               // could also use HardwarePushbotMatrix class.
    double          clawOffset      = 0;                       // Servo mid position
    final double    CLAW_SPEED      = 0.02 ;                   // sets rate to move servo

    @Override
    public void runOpMode() {
        double left;
        double right;
        double max;

        /* Initialize the hardware variables.
         * The init() method of the hardware class does all the work here
         */
        robot.init(hardwareMap);

        // Send telemetry message to signify robot waiting;
        telemetry.addData("Say", "Hello Driver");    //
        telemetry.update();

        // Wait for the game to start (driver presses PLAY)
        waitForStart();
```

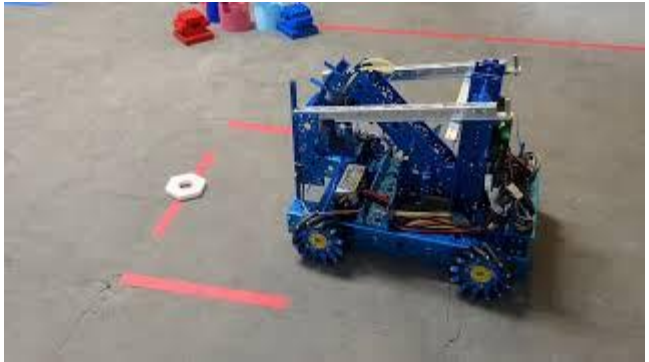
0: Messages | Terminal | 6: Android Monitor | TODO | 2: Event Log | Gradle Console

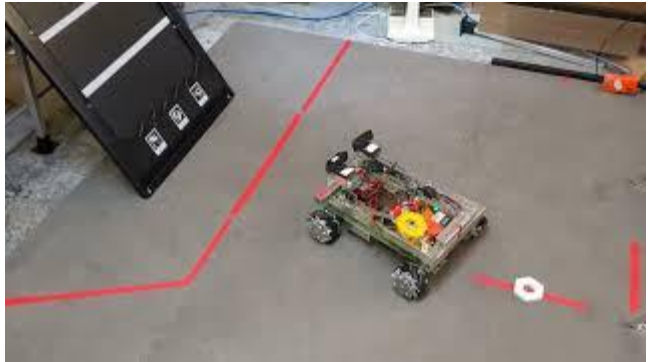
Class 'PushbotTeleopPOV_Linear' is never used

60:14 LF UTF-8 Context: <no context>

Programming the Robot

- Allow time to program the robot. All robots need a driver controlled program.
- The robot should be able to do something from all four positions in the autonomous period.
- Detecting the Team Prop and pushing a purple pixel to the indicated spike mark is a simple program all teams should aim to create. That's one program that would work from all starting positions and earns 20 points.
- Navigating to the backdrop is more of a challenge, especially from the front of the field.





Programs

You create programs that are called OpModes (Operational Modes).

There are two styles of OpModes that you can create.

- Autonomous - you select one of this type of program to run during the 30 second autonomous period at the start of a match. The program runs without any input from the drive team, but it can use cameras and sensors to navigate and perform actions on the field. You can also provide inputs to the program during the initialization period.
- Teleop - after the autonomous period, there is a two minute driver controlled period where the drive team uses the gamepad(s) to issue commands to drive the robot and have it perform actions on the field.

OpMode vs LinearOpMode

LinearOpMode is a class derived from OpMode that was meant to simplify Java programming. Blocks and the sample Java programs operate using LinearOpMode. That can be seen where the program flows from initialization to wait for start to the loop and then finally exits the loop.

OpMode operates in more of a real-time programming style where you need to maintain state about what action you want to do next. There are also separate Initialize, start and stop functions you need to write.

Both Autonomous and TeleOp programs often end up with state machine like logic which can operate better as a non-linear op mode.