The SKU 3110-0003-0001 is a USB Camera with goBILDA® Case. It is a UVC compatible USB Webcam which can record up to 4k @ 15fps. It is autofocus equipped, with USB camera controls including focus, brightness, white balance, and exposure. Its 76° recording angle strikes a great balance between low distortion and wide angle.

Table of Contents:

Summary of Product Ratings:

Operating Voltage	5V		Dynamic Range	72.5dB
Input Current	160mA - 260mA		Sensor	Sony IMX179
Interface Type	USB 2.0 High Speed		USB ID	VID: 0x0BDA, PID: 0x5805
Shutter Type	Electronic Rolling shutter		Wire Length	920mm

Supported Resolutions:				
Resolution	**Frame Rate**		**Resolution**	**Frame Rate**
3264x2448	15 fps		960x540	30 fps
2592x1944	15 fps		800x600	30 fps
1920x1080	30 fps		640x480	30 fps
1280x720	30 fps			

Camera Calibration Characteristics:			
Resolution	**Focal Length (fx, fy)**	**Principal Point (cx, cy)**	**Distortion Coefficients**
1920, 1080	1432.032f, 1432.032f	997.085f, 1432.032f	0.1289180, -0.3621222, 0, 0, 0.2872672, 0, 0, 0
1280, 720	964.146f, 964.146f	637.101f, 369.345f	0.1345473, -0.3987214, 0, 0, 0.3413483, 0, 0, 0
800, 600	602.426f, 602.426f	416.250f, 302.576f	0.1402389, -0.4099823, 0, 0, 0.3524094, 0, 0, 0
640, 480	481.985f, 481.985f	334.203f, 241.948f	0.1293532, -0.3755644, 0, 0, 0.3164797, 0, 0, 0

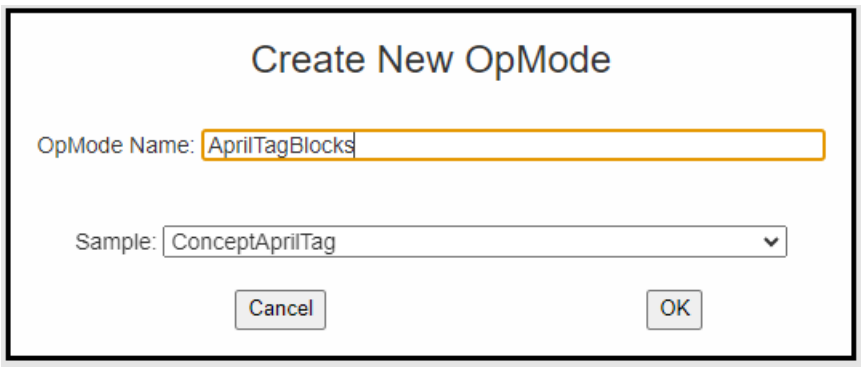
These were calibrated using 3DF Zephyr https://ftc-docs.firstinspires.org/en/latest/programming_resources/vision/camera_calibration/camera-calibration.html
We encourage you to run these tests for yourself, on your camera. These are a great starting point, but manufacturing variances can mean that these characteristics are subtly different camera-to-camera. And accurate calibration characteristics are the most important part of a good vision-based localization system.

Supported USB Camera Parameters:
- Brightness/Exposure
- Focus (autofocus on/off, manual focus)
- White Balance

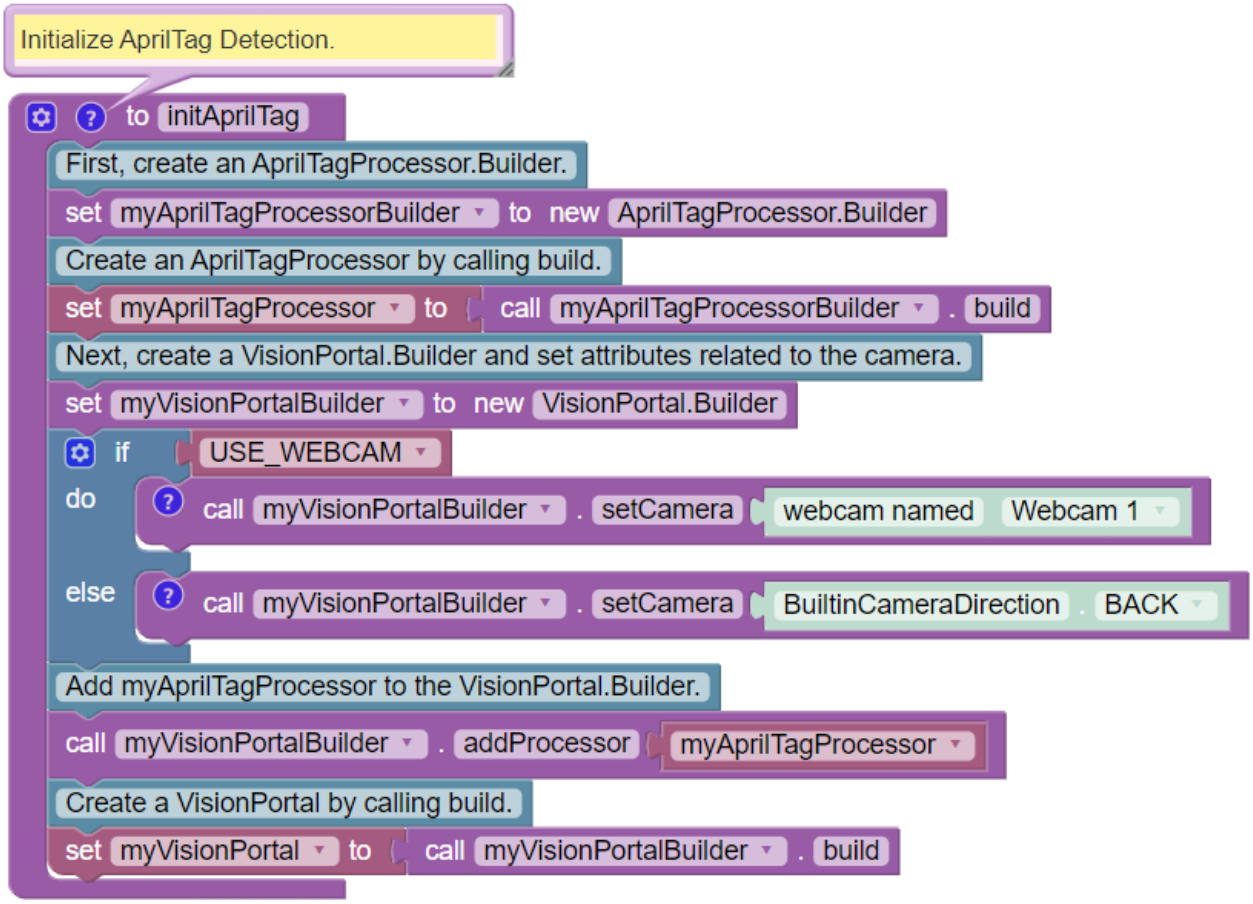
FTC Installation Guide - Blocks:

If you are using FTC Blocks, follow these steps to include these camera calibration characteristics in your code.

Step 1: Create OpMode based on the "ConceptAprilTag" sample



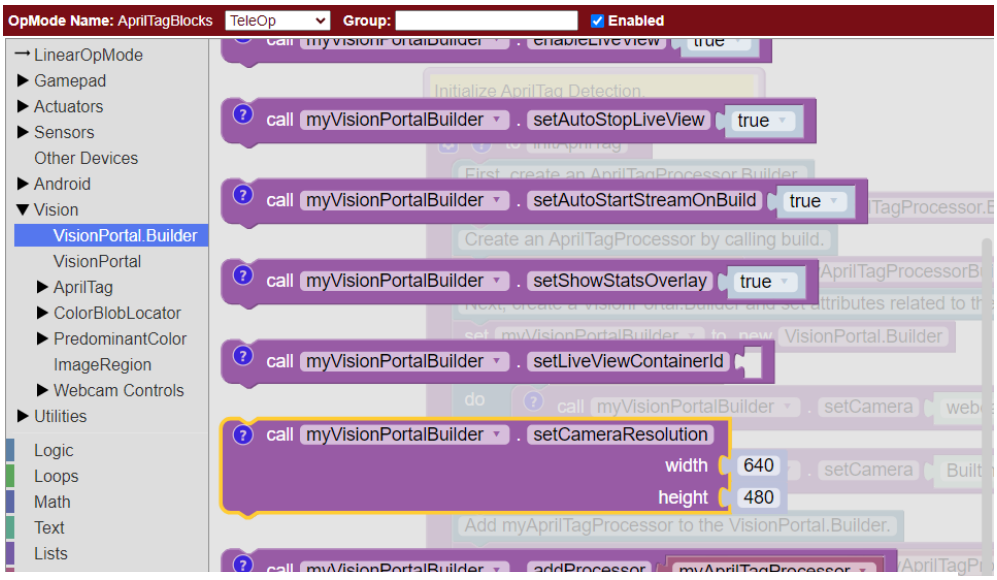
Step 2: Find the "InitAprilTag" function blocks

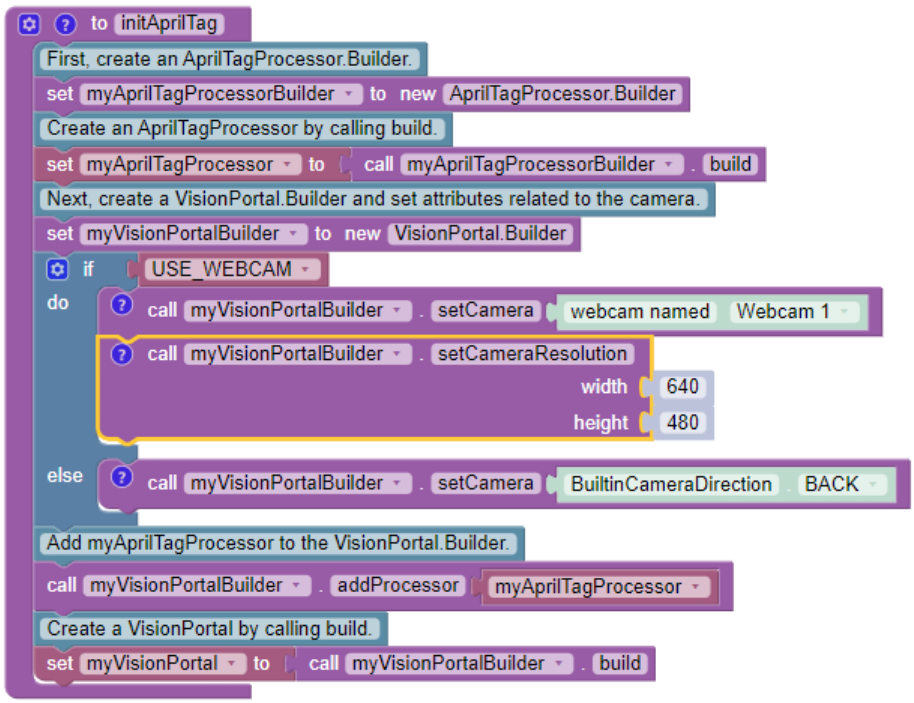


FTC Installation Guide - Blocks Continued:

Step 3: Set Camera Resolution

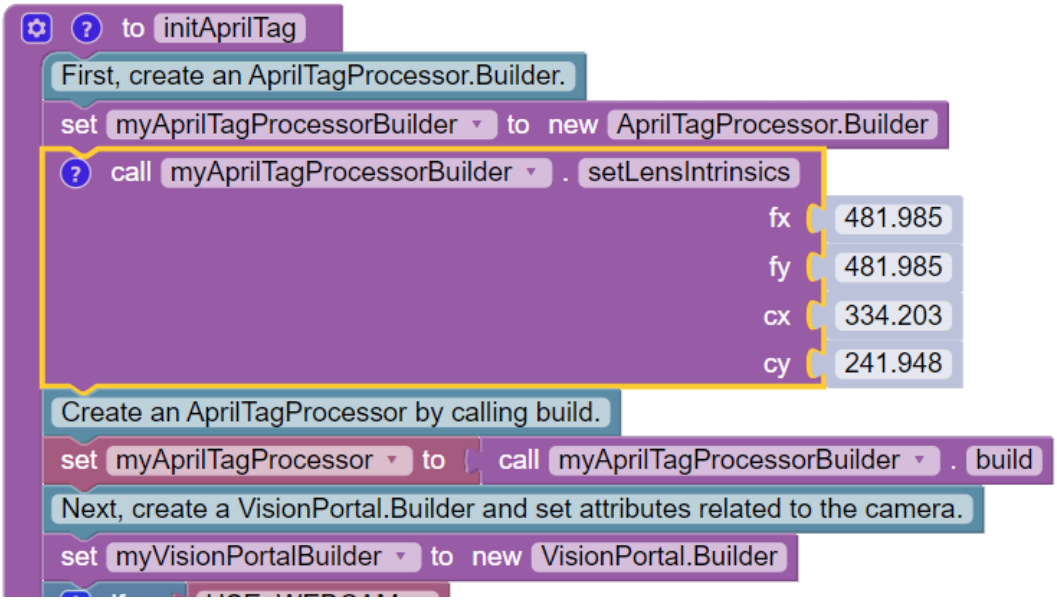
On the left side of your screen, navigate to Vision, then VisionPortalBuider. And drag over the “setCameraResolution” Block into the “initAprilTag” function. Put it inside the if “USE_WEBCAM” statement, and set width to 640, and height to 480.



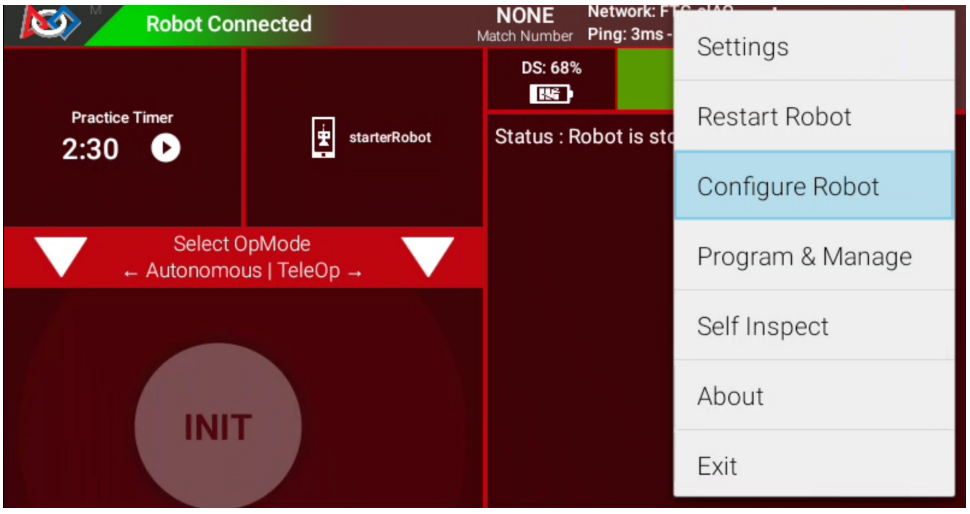


FTC Installation Guide - Blocks Continued:

Step 4: Set Camera Calibration
Navigate to Vision > AprilTag > AprilTagProcessorBuilder and drag over the “setLensIntrinsics” block. Put this under the “set “myAprilTagProcessorBuilder” to new “AprilTagProcessor.Builder”” Block.
Refer to the Camera Calibration Characteristics table on page 2 to find the values you need to set here, based on your resolution. For 640x480 we need to set fx to 481.985, fy to 481.985, cx to 334.203, and cy to 241.948.

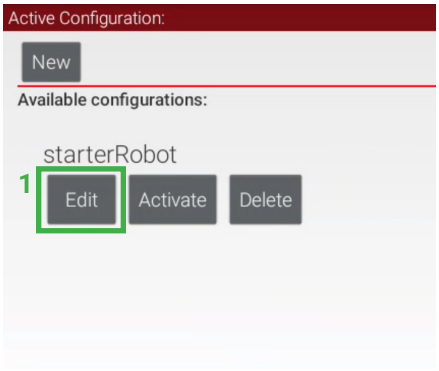


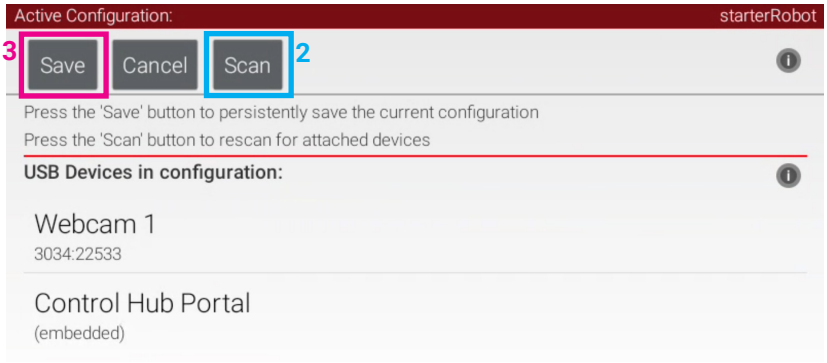
Step 5: Configure Hardware
Plug the Camera into the USB 3.0 port on your Control Hub, click the menu at the top right of your driver’s station, and click “Configure Robot”.



FTC Installation Guide - Blocks Continued:

Step 6: Add Webcam to Robot Configuration
Click “Edit” on the configuration you’d like to use, and then click “Scan”. You should see a device appear called “Webcam 1”. Leave that name as-is, and click the “Save” button at the top left.

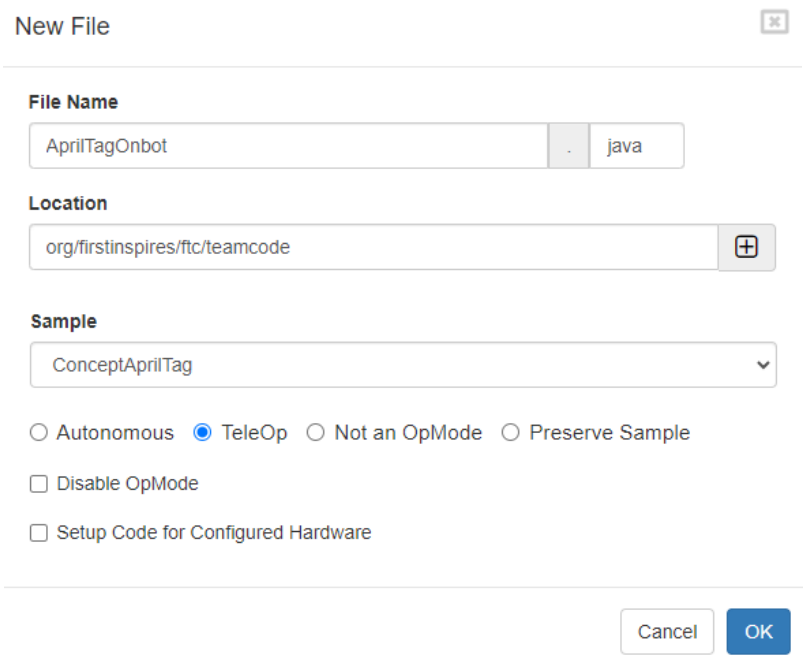




FTC Installation Guide - Onbot Java:

In this section, we’ll follow a very similar set of steps to set up this camera for use with Onbot Java.

Step 1: Create OpMode based on the “ConceptAprilTag” sample



FTC Installation Guide - Onbot Java Continued:

Step 2: Set Camera Resolution
On line 123 you will find the function initAprilTag(). This gets called on line 87 to apply all these settings at once. Using a function here cleans up the code a lot.
Inside the initAprilTag() function, you'll find an if statement on line 157. This checks to see if the user has selected that they are using a webcam, or the built-in camera on an Android device. After line 158, add the statement: builder.setCameraResolution(new Size(640,480));
This sets the webcam resolution to 640x480p.

```
155
156         // Set the camera (webcam vs. built-in RC phone camera).
157         if (USE_WEBCAM) {
158             builder.setCamera(hardwareMap.get(WebcamName.class, "Webcam 1"));
159             builder.setCameraResolution(new Size(640,480));
160         } else {
161             builder.setCamera(BuiltinCameraDirection.BACK);
162         }
163
```

Step 3: Set Camera Calibration
On line 139 you'll find a commented-out line of code that sets the "lens intrinsics" for your camera. "Uncomment" this line. And change the calibration values to: 481.985, 481.985, 334.203, 241.948
.setLensIntrinsics(481.985, 481.985, 334.203, 241.948)

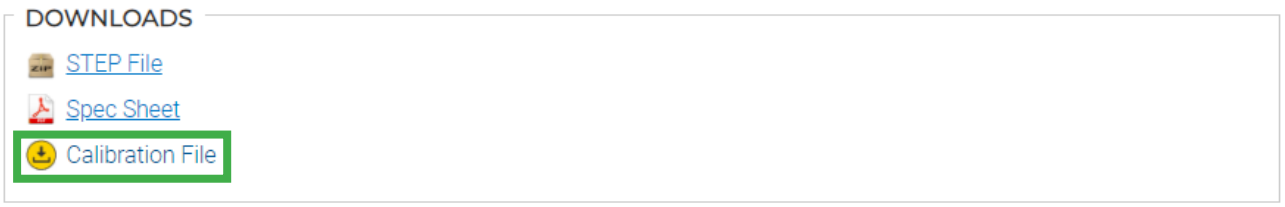
```
124
125         // Create the AprilTag processor.
126         aprilTag = new AprilTagProcessor.Builder()
127
128             // The following default settings are available to un-comment and edit as needed.
129             //.setDrawAxes(false)
130             //.setDrawCubeProjection(false)
131             //.setDrawTagOutline(true)
132             //.setTagFamily(AprilTagProcessor.TagFamily.TAG_36h11)
133             //.setTagLibrary(AprilTagGameDatabase.getCenterStageTagLibrary())
134             //.setOutputUnits(DistanceUnit.INCH, AngleUnit.DEGREES)
135
136             // == CAMERA CALIBRATION ==
137             // If you do not manually specify calibration parameters, the SDK will attempt
138             // to load a predefined calibration for your camera.
139             .setLensIntrinsics(481.985, 481.985, 334.203, 241.948)
140             // ... these parameters are fx, fy, cx, cy.
141
142             .build();
143
```

Step 4: Set Hardware Configuration
Refer to steps 5 and 6 of the Blocks installation guide for a walkthrough of how to add the webcam to your robot's configuration file.

FTC Installation Guide - Android Studio:

The Android Studio installation is very different from either Blocks or Onbot Java. You can still set your camera calibration in your Java code by calling .setLensIntrinsics(). But you can also just upload a webcam config file which will capture our defaults at a number of different resolutions.

Step 1: Download the Calibration File
File is available here: www.gobilda.com/usb-camera-with-gobilda-case-autofocus-8-megapixel-usb-type-a/



Step 2: Upload File to TeamWebCamCalibrations folder
Once you have downloaded the calibration file, unzip it and copy the teamwebcamcalibrations file to your clipboard.
Now in Android Studio, navigate to your TeamCode folder, then to res > xml and you'll find a file called "teamwebcamcalibrations.xml. Click this file and click paste. It will ask if you would like to overwrite the existing file, click overwrite. Once this is complete, any code you build using this webcam will automatically find the intrinics we have calibrated for it.