Run with stacktrace android studio

Continue











This repository contains the public FTC SDK for the Freight Frenzy (2021-2022) competition season. Welcome! This GitHub repository contains the source code that is used to build an Android app to control a FIRST Tech Challenge competition robot. To use this SDK, download/clone the entire project to your local computer. Getting Started If you are new to robotics or new to the FIRST Tech Challenge, then you should consider reviewing the FTC Blocks Tutorial to get familiar with how to use the control system: FTC Blocks tutorial, and then migrate to the OnBot Java Tool or to Android Studio afterwards. Downloading the Project If you are an Android Studio programmer, there are several ways to download this repository. If you are a git user, you can clone the most current version of the repository: ait clone Or, if you prefer, you can use the "Download Zip" button available through the main repository page. Download the project as a .ZIP file will keep the size of the download the project as a .ZIP file will keep the size of the download the project as a .ZIP file will keep the size of the download the project as a .ZIP file will keep the size of the download the project folder (as a .zip or .tar.gz archive file) from the Downloads subsection of the Releases page for this repository. The Releases page also contains prebuilt APKs. Once you have downloaded and uncompressed (if needed) your folder, you can use Android Studio to import the folder ("Import project (Eclipse ADT, Gradle, etc.)"). Getting Help User Documentation and Tutorials FIRST Tech Challenge software and robot control system. You can access this documentation using the following link: FtcRobotController Online Documentation is an "evergreeen" documentation is an "evergreeen" documentation about the FIRST Tech Challenge software and control system. Javadoc Reference Material The Javadoc documentation for the FTC SDK is now available online. Click on the following link to view the FTC SDK javadoc documentation as a live website: FTC Javadoc Documentation on the FTC SDK javadoc documentation for the FTC SDK is now available online. FTC Technology forum: FTC Technology Forum Sample OpModes This project contains a large selection of Sample OpModes (robot code examples) which can be cut and pasted into your /teamcode folder to be used as-is, or modified to suit your team's needs. Samples Folder: /FtcRobotController/src/main/java/org/firstinspires/ftc/robotcontroller/external/samples The readme.md file located in the /TeamCode/src/main/java/org/firstinspires/ftc/teamcode folder contains an explanation of the sample naming convention, and instructions on how to copy them to your own project space. Version 8.0 (20220907-131644) Breaking Changes Increases the Robocol version. This means an 8.0 or later Robot Controller or Driver Station will not be able to communicate with a 7.2 or earlier Driver Station or Robot Controller. If you forget to update both apps at the same time, an error message will be shown explaining which app is older and should be updated. Initializing I2C devices now happens when you retrieve them from the HardwareMap for the first time. Previously, all I2C devices would be initialized before the Op Mode even began executing, whether you were actually going to use them or not. This could result in reduced performance and unnecessary warnings. With this change, it is very important for Java users to retrieve all needed devices from the HardwareMap during the Init phase of the Op Mode. Namely, declare a variable for each hardware device the Op Mode may briefly hang while the devices you are retrieving get initialized. Op Modes that do not use all of the I2C devices specified in the configuration file should take less time to initialize. Op Modes that do use all of the specified I2C devices should take the same amount of time as previously. Fixes issue #251 by changing the order in which axis rotation rates are read from the angular velocity vector in the BNO055 IMU driver. Deprecates pitchMode in BNO055IMU.Parameters. Setting pitchMode to PitchMode.WINDOWS would break the coordinate conventions used by the driver. Moves OpModeManagerImpl to the com.qualcomm.robotcore.eventloop.opmode package. This breaks third party libraries EasyOpenCV (version 1.5.1 and earlier) and FTC Dashboard (version 0.4.4 and earlier). Deletes the deprecated OpMode method resetStartTime() (use resetRuntime() instead). Deletes the protected LinearOpModeHelper class (which was not meant for use by Op Modes). Removes I2C Device (Synchronous) config type (deprecated since 2018) Enhancements Uncaught exceptions in Op Modes no longer require a Restart Robot A blue screen popping up with a stacktrace is not an SDK error; this replaces the red text in the telemetry area. Since the very first SDK release, Op Mode crashes have put the robot into "EMERGENCY STOP" state, only showing the first line of the exception, and requiring the user to press "Restart Robot" to continue Exceptions during an Op Mode now open a popup window with the same color scheme as the log viewer, containing 15 lines of the exception stacktrace to allow easily tracing down the offending line without needing to connect to view logs over ADB or scroll through large amounts of logs in the log viewer. The exception stacktrace to allow easily tracing down the offending line without needing to connect to view logs over ADB or scroll through large amounts of logs in the log viewer. like a webpage. Pressing the "OK" button in the popup window will return to the main screen of the Driver Station and allow an Op Mode to be run again immediately, without the need to perform a "Restart Robot" Adds new Java sample to demonstrate using a hardware class to abstract robot actuators, and share them across multiple Op Modes. Updates RobotAutoDriveByGyro Linear Java samples to reference PowerPlay assets and have correct names and field locations of image targets. Updates TensorFlow samples to reference PowerPlay assets and have correct names and field locations of image targets. opt in, open the OnBotJava Settings, and check Enable beta Java 8 support. Note that Java 8 code will only compile when the Robot Controller runs Android 7.0 Nougat or later. Please report issues here. In OnBotJava, clicking on build errors now correctly jumps to the correct location. Improves OnBotJava autocomplete behavior, to provide better completion options in most cases. Adds a QR code to the Robot Controller Inspection Report when viewed from the Driver Station for scanning by inspectors at competition. Improves I2C performance and reliability in some scenarios. Version 7.2 (20220723-130006) Breaking Changes Updates the build tooling. For Android Studio users, this change requires Android Studio Chipmunk 2021.2.1. Removes support for devices that are not competition legal, including Modern Robotics Core Control Modules, the Matrix Controllers, and HiTechnic/NXT controllers and sensors. Support remains for Modern Robotics I2C sensors. Enhancements Increases the height of the 3-dots Landscape menu touch area on the Driver Station, making it much easier to select. Adds terminateOpModeNow() method to allow OpModes to cleanly self-exit immediately. Adds opModeIsActive() but for the init phase. Warns user if they have a Logitech F310 gamepad connected that is set to DirectInput mode. Allows SPARKmini motor controllers to react more quickly to speed changes. Hides the version number of incorrectly installed on RC device or vice-versa) on inspection screen. Adds support for allowing the user to edit the comment for the runOpMode block. Adds parameterDefaultValues field to @ExportToBlocks. This provides the ability for a java method with an @ExportToBlocks annotation to specify default values for method parameters when it is shown in the block editor. Make LinearOpMode blocks. Added support to TensorFlow Object Detection for using a different frame generator, instead of Vuforia. Using Vuforia to pass the camera frame to TFOD is still supported. Removes usage of Renderscript. Fixes logspam on app startup of repeated stacktraces relating to "Failed resolution of: Landroid/net/wifi/p2p/WifiP2p bluetooth radio from inspection screen Improves warning messages when I2C devices are not responding Adds support for controlling the RGB LED present on PS4/Etpark gamepads from OpMode samples to "Robot". Motor directions reversed to be compatible with "direct Drive" drive train. Bug fixes Fixes issue #316 (MatrixF.inverted() returned an incorrectly-sized matrix for 1x1 and 2x2 matrixes). Self inspect now allows for Driver Station and Robot Controller compatibility between point releases. Fixes bug where if the same RumbleEffect object instance was queued for multiple gamepads, it could happen that both rumble commands would be sent to just one gamepad. Fixes bug in Driver Station where on the Driver Hub, if Advanced Gamepad Features or Gamepad Type Overrides screens would cause the gamepad to be rebound by the custom USB driver even though advanced gamepad features was disabled. Protects against (unlikely) null pointer exception in Vuforia Localizer. Harden OnBotJava and Blocks saves to protect against save issues when disconnecting from Program and Manage Fixes issue where the RC app would hang if a REV Hub I2C write failed because the previous I2C operation was still in progress. This hang most commonly occurred during REV 2M Distance Sensor initialization Removes ConceptWebcam.java sample is not compatible with OnBotJava. Fixes blocks editor so it doesn't ask you to save when you haven't modified anything. Fixes uploading a very large blocks project to offline blocks editor. Fixes Blocks Programs Stripped of Blocks (due to using TensorFlow Label block) Version 7.1 (20211223-120805) Fixes crash when calling isPwmEnabled() (issue #223). Fixes lint error (issue #4). Fixes Driver Station would use generic 20% deadzone for Xbox360 and Logitech F310 gamepads when Advanced Gamepad Features was disabled. Added SimpleOmniDrive samples for TensorFlow can't be used for a different model. The blocks previously labeled TensorFlowObjectDetectionFreightFrenzy (from the subcategory named "Optimized for Freight Frenzy") and TensorFlowObjectDetection. Blocks in existing opmodes will be automatically updated to the new blocks when opened in the blocks editor. Fixes issue #260 Blocks can't call java method that has a VuforiaLocalizer parameter. Blocks now has a block labeled VuforiaFreightFrenzy.getVuforiaLocalizer for this. Added a page to manage the TensorFlow Lite models in /sdcard/FIRST/tflitemodels. To get to the TFLite Models page: You can click on the link at the bottom of the the Manage page. You can click on the link at the upper-right the Blocks project page. Fixes logspam when isBusy() is called on a motor not in RTP mode. Hides the "RC Password" item on the inspection screen for phone-based Robot Controllers. (It is only applicable for Control Hubs). Adds channel 165 to Wi-Fi Direct channel selection menu in the settings screen. (165 was previously available through the web UI, but not locally in the app). Version 7.0 (20210915-141025) Enhancements and New Features Adds support for external libraries to OnBotJava. Known limitation - .aar files with assets are not supported. External libraries can provide support for hardware devices by using the annotation in the com.qualcomm.robotcore.hardware.configuration.annotations package. External libraries can include .so files for native code. External libraries can be used from OnBotJava op modes. External libraries that use the following annotations can be used from Blocks op modes. org.firstinspires.ftc.robotcore.external.ExportClassToBlocks org.firstinspires.ftc.robotcore.external.ExportToBlocks org.firstinspires.ftc.robotcore.external.ExportClassToBlocks org.firstinspires.ftc.robotcore.external.ExportToBlocks org.firstinspires.ftc.robotcore.external.ExportToBlocks org.firstinspires.ftc.robotcore.external.ExportClassToBlocks org.firstinspires.ftc.robotcore.external.ExportToBlocks org.firstinspires.ftc.robotcore.external.ExportClassToBlocks org.fir com.qualcomm.robotcore.hardware.configuration.annotations.DeviceProperties com.qualcomm.robotcore.hardware.configuration.annotations.I2cDeviceType com.qualcomm.robotcore.hardware.configuration.annotations.MotorType com.qualcomm.robotcore.hardware.configuration.annotations.ServoType External.OnCreate org.firstinspires.ftc.ftccommon.external.OnCreateMenu org firstinspires.ftc.ftccommon.external.OnDestroy org firstinspires.ftc.ftccommon.external.WebHandlerRegistrar Adds support for REV Robotics Driver Hub. Adds fully custom userspace USB gamepad driver to Driver Station (see "Advanced Gamepad Features" menu in DS settings). Allows gamepads to work on devices without native Linux kernel support (e.g. some Romanian Motorola devices). Allows the DS to read the unique serial number of each gamepad, enabling auto-recovery of dropped gamepads even if two gamepa also provides the ability to configure the DS to assign gamepads to a certain position by default (so no need to do start+a/b at all). The LED ring on the Xbox360 gamepad and the RGB LED bar on the PS4 gamepad is used to indicate the driver position the gamepad is used to indicate the driver position the gamepad is used to indicate the driver position the gamepad is bound to. controlled from OpModes. The 2-point touchpad on the PS4 gamepad can be read from OpModes. The "back" and "guide" buttons on the gamepad can now be safely bound to robot controls (Previously, on many devices, Android would intercept these buttons as home button presses and close the app). Advanced Gamepad features are enabled by default, but may be disabled through the settings menu in order to revert to gamepad support provided natively by Android. Improves accuracy of ping measurement. Fixes issue where the ping time showed as being higher than reality when initially connecting to or restarting the robot. To see the full improvement, you must update both the Robot menu in order to revert to gamepad support provided natively by Android. Controller and Driver Station apps. Updates samples located at /FtcRobotController/src/main/java/org/firstinspires/ftc/robotcontroller/src/main/ja (ConceptVuforiaFieldNavigation & ConceptVuforiaFieldNavigationWebcam) showing how to determine the robot's location on the field using Vuforia. These both use the current season's Target images. Added ConceptVuforiaDriveToTargetWebcam to illustrate an easy way to drive directly to any visible Vuforia target. Makes many improvements to

the warning system and individual warnings. Warnings are now much more spaced out, so that they are easier to read. New warning now uses the major and minor app versions, not the version code. The warnings are automatically re-enabled when a Robot Controller app from a new FTC season is installed. Adds support for I2C transactions on the Expansion Hub / Control Hub without specifying a register address. See section 3 of the TI I2C spec. Calling these new methods when using Modern Robotics hardware will result in an UnsupportedOperationException. Changes VuforiaLocalizer close() method to be public. Adds support for TensorFlow v2 object detection models. Reduces ambiguity of the Self Inspect language and graphics. OnBotJava now warns about potentially unintended file overwrites. Improves behavior of the Wi-Fi band and channel selector on the Manage webpage. Bug fixes Fixes Robot Controller app crash on Android 9+ when a Driver Station connects. Fixes issue where an Op Mode was responsible for calling shutdown on the TensorFlow TFObjectDetector. Now this is done automatically. Fixes Vuforia initialization blocks and Java class do not work. Fixes match logging for Op Modes that contain special characters in their names. Fixes Driver Station OpMode controls becoming unresponsive if the Driver Station app closing itself when it is switched away from, or the screen is turned off. Fixes "black swirl of doom" (Infinite "configuring Wi-Fi Direct" message) on older devices. Updates the wiki comment on the OnBotJava intro page. Version 6.2 (20210218-074821) Enhancements Attempts to automatically fix the condition where a Control Hub's internal Expansion Hub is not working by re-flashing its firmware Makes various improvements to the Wi-Fi Direct pairing screen, especially in landscape mode Makes the Robot Controller service is no longer restarted simply by viewing the Self Inspect screen and pressing the back button) It is still restarted if the Settings menu or Configure Robot menu is opened Bug fixes Fixes FtcRobotController issue #79 6.1 causes a soft reboot on the Motorola E5 Play Fixes issue where the Control Hub OS's watchdog would restart the Robot Controller app if the Control Hub was not able to communicate with its internal Expansion Hub Fixes certain I2C devices not showing up in the appropriate HardwareMap.colorSensor) Fixes issue where OnBotJava might fail to create a new file if the option to "Setup Code for Configured Hardware" was selected Fixes issue where a Control Hub would not work if the Expansion Hub was missing at startup Fixes potential issues caused by having mismatched Control/Expansion Hub firmware versions Fixes ftc app issue 673 Latest matchlog is being deleted instead of old ones by RobotLog Fixes ConceptVuforiaUltimateGoalNavigationWebcam sample opmode by correctly orienting camera on robot. spammed with InterruptedExceptions when stop is requested from the Driver Station (this behavior was accidentally introduced in v5.3). This change has no impact on functionality. Fixes issue where the blocks editor fails to load if the name of any TeleOp opmode contains an apostrophe. Version 6.1 (20201209-113742) Makes the scan button on the configuration screen update the list of Expansion Hubs connected via RS-485 Fixes SkyStone issue #143 Improves web interface compatibility with older browser and Android System WebView versions. Fixes issue in UVC driver where some cameras (e.g. certain MS Lifecams) which reported frame intervals as rounded rather than truncated values (e.g. 666667*100ns instead of 666666*100ns for 15FPS) would fail to start streaming. Adds support in UVC driver for enabling/disable AE priority. This setting provides a means to tell the camera firmware either A) It can undershoot the requested frame rate in order to provide a theoretically better image (i.e. with a longer exposure than the inter-frame period of the selected frame rate, even if the image may be underexposed as a result Adds support for the Control Hub OS 1.1.2 Robot Controller watchdog The Robot Controller app will be restarted if it stops responding for more than 10 seconds Adds support for using the Driver Station to indicate the OpMode name in the OpMode list dropdown on the Driver Station to indicate the option of the Driver Station of the Driver Station to indicate the option of the Driver Station of the Driver Station to indicate the option of the Driver Station of t source of the OpMode (i.e. the programming tool used to create it) Fixes issue where the Driver Station app would exit after displaying the Configuring Wi-Fi Direct screen Fixes Blocks and OnBotJava prompts when accessed via the REV Hardware Client Version 6.0 (20200921-085816) Important Notes Version 6.0 is the version for the Ultimate Goal season. Requires Android Studio 4.0. Android Studio users need to be connected to the Internet the first time they build the app (in order to download needed packages for the build). Version 5.5 and greater will not work on older Android 4.x and 5.x phones. Users must upgrade to an approved Android 6.x device or newer. The default PIDF values for REV motors have been reverted to the default PIDF values that were used in the 2018-2019 season This change was made because the 2018-2019 season This change was made because the 2018-2019 values turned out to work better for many mechanisms. the REV motors in line with the behavior of all other motors If you prefer the 2019-2020 season's behavior for REV motors, here are the PIDF values: P = 1.17, I = 0.117, F = 11.7 Position PIDF values: P = 5.0 Core Hex motor Velocity PIDF values: P = 4.96, I = 0.496, F = 49.6 Position PIDF values: P = 5.0 New features Includes TensorFlow object Ultimate Goal vision targets and sample op modes. Introduces a digital zoom feature for TensorFlow object detection (to detect objects more accurately at greater distances). Adds configuration entry for the REV UltraPlanetary HD Hex motor Enhancements Adds setGain() and getGain() and getGain() methods to the NormalizedColorSensor interface By setting the gain of a color sensor, you can adjust for different lighting conditions. For example, if you detect lower color values than expected, you can increase the gain. The gain value is only applied to the argb() and getNormalizedColors() methods, not to the raw color methods. The getNormalizedColors() methods, not to the raw color methods. SensorREVColorDistance Java sample into SensorColor Java sample, which showcases best practices for all color sensors Improves the calculation of the RGB channels Improves the calculation of the RGB channels indicator) Fixes the default sensor resolution, which caused issues with bright environments Adds support for changing the resolution and measuring rate of the Broadcom sensor chip Bug fixes Improves reliability of BNO055IMU IMU initialization to prevent random initialization failures (which manifested as Problem with 'imu'). Version 5.5 (20200824-090813) Version 5.5 requires Android Studio 4.0 or later. New features Adds support for calling custom Java classes from Blocks OpModes (fixes SkyStone issue #161). Classes must be in the org.firstinspires.ftc.teamcode package. To have easy access to the opMode, hardwareMap, telemetry gamepad1, and gamepad2, classes can extends org.firstinspires.ftc.robotcore.external.BlocksOpModeCompanion. Methods must be annotated with org.firstinspires.ftc.robotcore.external.ExportToBlocks. Parameters declared as OpMode, Telemetry, and HardwareMap are supported and the argument is provided automatically, regardless of the order of the parameters declared as char or java.lang. Character will accept any block that returns text and will only use the first character in the text. Parameters declared as boolean or java.lang.Boolean will accept any block that returns boolean. Parameters declared as byte, java.lang.Float, double, java.lang.Short, int, java.lang.Short, int, java.lang.Integer, long, or java.lang.Integer, long, or java.lang.Short, int, java.lang.Sh java.lang.Double will accept any block that returns a number. Adds telemetry API method for setting display format Classic Monospace HTML (certain tags only) Adds blocks support for switching cameras. Adds Blocks support for TensorFlow Object Detection model in the Manage page, which is especially useful for Blocks and OnBotJava users. Shows new Control Hub blink codes when the Wi-Fi band is switched using the Control Hub's button (only possible on Control Hub blink codes and OnBotJava users). GHz Wi-Fi usage warning REV Hub is running outdated firmware (older than version 1.8.2) Adds support for Sony PS4 gamepads work on the Driver Station Removes preference which sets gamepad type based on driver position. Replaced with menu which allows specifying type for gamepads with unknown VID and PID Attempts to auto-detect gamepad type based on USB VID and PID is not known, use type specified a type for that VID and PID, an educated guess is made about how to map the gamepad Driver Station will now attempt to automatically recover from a gamepad disconnecting, and re-assigned to when it dropped If only one gamepads are assigned, and have different VID/PID signatures, and only one drops: it will be recovered If two gamepads are assigned, and have different VID/PID signatures, and only one drops: it will be recovered If two gamepads are assigned, and have different VID/PID signatures, and only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned and it drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drops: it can be recovered If two gamepads are assigned to when it dropped If only one drop VID/PID signatures, and BOTH drop: both will be recovered If two gamepads are assigned, and have the same VID/PID signatures, and BOTH drop: neither will be recovered If two gamepads when they re-appear on the USB bus. There is currently one known edge case: if there are two gamepads with the same VID/PID signature plugged in, but only one is assigned, and they BOTH drop, it's a 50-50 chance of which one will be chosen for automatic recovery to the assigned position: it is determined by whichever one is re-enumerated first by the USB bus controller. Adds landscape user interface to Driver Station New feature: practice timer with audio cues New feature (Control Hub only): tapping on the ping/channel display will switch to an alternate display showing radio RX dBm and link speed (tap again to switch back) The layout will NOT autorotate. You can switch the layout from the Driver Station's settings menu. Breaking changes Removes the deprecated LinearOpMode methods waitOneFullHardwareCycle() and waitForNextHardwareCycle() Enhancements Handles RS485 address of Control Hub is automatically given a reserved address existing configuration files will continue to work All addresses in the range of 1-10 are still available for Expansion Hubs The Control Hub light will now normally be solid green, without blinking to indicate the address The Control Hub will not be shown on the Expansion Hub Address Change settings page Improves REV Hub firmware update files Version 1.8.2 of the REV Hub firmware update files Ver Hubs can only be updated via USB. Firmware update speed was reduced to improve reliability Allows REV Hub firmware to be updated directly from the Manage webpage Improves log viewer on Robot Controller Horizontal scrolling support (no longer word wrapped) Supports pinch-to-zoom Uses a monospaced font Error messages are highlighted New color scheme Attempts to force-stop a runaway/stuck OpMode without restarting the entire app Not all types of runaway conditions are stoppable, but if the user code attempts to talk to hardware during the runaway conditions are stoppable. version" Renames "Wi-Fi Direct Name" to "Wi-Fi Name" Adds Control Hub OS version, when viewing the report of a Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Removes check for ZTE Speed Channel Changer Shows firmware version for all Expansion and Control Hub Remo Manage page All network settings are now applied with a single click The Wi-Fi Direct channels of phone-based Robot Controllers, and whether they overlap with other channels The current Wi-Fi channel is pre-selected on phone-based Robot Controllers, and Control Hubs running OS 1.1.2 or later. On Control Hubs running OS 1.1.2 or later, you can choose to have the system automatically select a channel on the 5 GHz band Improves OnBotJava New light and dark themes replace the old themes (chaos, github, chrome,...) the new default theme is light and will be used when you first update to this version OnBotJava now has a tabbed editor Read-only offline mode Improves function of "exit" menu item on Robot Controller and Driver Station Now guaranteed to be fully stopped and unloaded from memory Shows a warning message if a LinearOpMode exists prematurely due to failure to monitor for the start condition Improves error message shown when the Driver Station and Robot Controller are incompatible with each other Driver Station OpMode Control Hub. Tint phone battery icons on Driver Station when low/critical. Uses names "Control Hub. Portal" and "Control Hub" (when appropriate) in new configuration files Improve I2C read performance Very large improvement on Expansion Hubs connected to a phone Update/refresh build infrastructure Update to 'androidx' support library from 'com.android.support:appcompat', which is end-of-life Update targetSdkVersion and compileSdkVersion to 28 Update Android Studio's Android plugin to latest Fix reported build timestamp in 'About' screen Add sample illustrating manual webcam use: ConceptWebcam Bug fixes Fixes SkyStone issue #248 Fixes SkyStone issue #232 and modifies bulk caching semantics to allow for cache-preserving MANUAL/AUTO transitions. Improves performance when REV 2M distance sensor is unplugged Improves readability of Toast messages on certain devices Allows a Driver Station to connect to a Robot Controller after another has disconnected Improves generation of fake serial numbers for UVC cameras which do not provide a real serial number Previously some devices would assign such cameras a serial of 0:0 and fail to open and start streaming Fixes ftc app issue #638. Fixes a slew of bugs with the Vuforia camera monitor including: Fixes bug where preview could be cut off in landscape Fixes bug where preview got totally messed up when rotating phone Fixes bug where crosshair could drift off target when using webcams Fixes issue in UVC driver on some devices (ftc app 681) if streaming was started/stopped multiple times in a row Issue manifested as kernel panic on devices which do not have this kernel patch. On affected devices which do have the patch, the issue was manifest as simply a failure to start streaming. The Tech Team believes that the root cause of the issue is a bug in the Linux kernel XHCI driver. A workaround was implemented in the SDK UVC driver. dropped (e.g. only 15FPS delivered during a streaming session configured for 30FPS). Fixes issue where TensorFlow Object Detection would show results whose confidence was lower than the minimum confidence parameter. Fixes a potential exploitation issue of CVE-2019-11358 in OnBotJava Fixes changing the address of an Expansion Hub with additional Expansion Hubs connected to it Preserves the Control Hub's network connection when "Restart Robot" is selected Fixes issue where device scans would fail while the Robot was restarting Fix RenderScript usage Use androidx.renderscript variant: increased compatibility Use RenderScript in Java mode, not native: simplifies build Fixes webcam-frame-to-bitmap conversion problem: alpha channel wasn't being initialized, only R, G, & B Fixes possible arithmetic overflow in Deadline Fixes deadlock in Vuforia webcam support which could cause 5-second delays when stopping OpMode Version 5.4 (20200108-101156) Fixes SkyStone issue #88 Adds an inspection item that notes when a robot controller (Control Hub) is using the factory default password. Fixes SkyStone issue #142 Fixes ftc_app issue #142 responsiveness issue when an Expansion Hub is disconnected Fixes issue where IMU initialization could prevent Op Modes from stopping Fixes issue where AndroidTextToSpeech.speak() would fail if it was called too early Adds telemetry.speak() methods and blocks, which cause the Driver Station (if also updated) to speak text Adds and improves Expansion Hub-related warnings Improves Expansion Hub low battery warning Displays the warning immediately after the hub reports it Specifies whether the condition is current or occurred temporarily during an OpMode run Fixes the hub's LED pattern after this condition Displays warning when Expansion Hub is not responding to commands Specifies whether the condition is current or occurred temporarily during an OpMode run Clarifies warning when Expansion Hub is not responding to commands Specifies that this condition is current or occurred temporarily during an OpMode run Clarifies warning when Expansion Hub is not responding to commands Specifies that this condition is current or occurred temporarily during an OpMode run Clarifies warning when Expansion Hub is not responding to commands Specifies that this condition is current or occurred temporarily during an OpMode run Clarifies warning when Expansion Hub is not responding to commands Specifies that this condition is current or occurred temporarily during an OpMode run Clarifies warning when Expansion Hub is not responding to commands Specifies that the condition is current or occurred temporarily during an OpMode run Clarifies warning when Expansion Hub is not responding to commands Specifies that the condition is current or occurred temporarily during an OpMode run Clarifies warning when Expansion Hub is not responding to commands Specifies that the condition is current or occurred temporarily during an OpMode run Clarifies warning when Expansion Hub is not responding to command the current of the used. The hub light will now accurately reflect this state Improves logging and reduces log spam during these conditions Syncs the Control Hub time and timezone to a connected web browser programming the robot, if a Driver Station is not available. Adds bulk read functionality for REV Hubs A bulk caching mode must be set at the Hub level with LynxModule#setBulkCachingMode(). This applies to all relevant SDK hardware classes that reference that Hub. The following following following following following hub bulk cachingMode.OFF (default): All hardware calls operate as usual. Bulk data can read through LynxModule#getBulkData() and processed manually. BulkCachingMode.AUTO: Applicable hardware calls are served from a bulk read cache that is cleared/refreshed automatically to ensure identical commands don't hit the same cache. The cache can also be cleared manually with LynxModule#clearBulkCache(), although this is not recommended. (advanced users) BulkCachingMode.MANUAL: Same as BulkCachingMode.AUTO except the cache is never cleared automatically. To avoid getting stale data, the cache must be manually cleared at the beginning of each loop body or as the user deems appropriate. Removes PIDF Annotation values added in Rev 5.3 (to AndyMark, goBILDA and TETRIX motor configurations). The new motor types will still be available but their Default control behavior will revert back to Rev 5.2 Adds new ConceptMotorBulkRead sample Opmode to demonstrate and compare Motor Bulk-Read modes for reducing I/O latencies. Version 5.3 (20191004-112306) Fixes external USB/UVC webcam support Makes various bugfixes and improvements to Blocks page, including but not limited to: Many visual tweaks Browser zoom and window resize behave better Resizing the Java preview pane works better and more consistently gets scrollbars when needed The Java preview pane is hidden by default on phones Internet Explorer 11 should work Large dropdown lists display properly on lower res screens Disabled buttons are now visually identifiable as disabled A warning is shown if a user selects a TFOD sample, but their device is not compatible Warning messages in a Blocks op mode are now visible by default. Adds goBILDA 5201 and 5202 motors to Robot Configurator Adds PIDF Annotation values to AndyMark, goBILDA and TETRIX motor configurations. This has the effect of causing the RUN_USING_ENCODERS and RUN_TO_POSITION modes to use PIDF vs PID closed loop control to the basic PID control loop. Feedforward is useful when controlling a motor's speed because it "anticipates" how much the control voltage must change to achieve a new speed set-point, rather than requiring the integrated error to change sufficiently. The PIDF values were chosen to provide responsive, yet stable, speed control on a lightly loaded motor. The more heavily a motor is loaded (drag or friction) the more noticable the PIDF improvement will be. Fixes startup crash on Android 10 Fixes ftc app issue #712 (thanks to FROGbots-4634) Fixes ftc app issue #542 Allows "A" and lowercase letters when naming device through RC and DS apps. Version 5.2 (20190905-083277) Fixes extra-wide margins on settings activities, and placement of the new configuration button Adds Skystone Vuforia image target data. Includes sample Skystone Vuforia Navigation op modes (Java). Includes sample Skystone Vuforia Navigation op modes (Java). TensorFlow op modes (Blocks). Removes older (season-specific) sample op modes. Includes 64-bit support (to comply with Google Play requirements). Protects against Stuck OpModes when a Restart Robot is requested. (Thanks to FROGbots-4634) (ftc app issue #709) Blocks related changes: Fixes bug with blocks generated code when hardware device name is a java or javascript reserved word. Shows generated java code for blocks, even when hardware items are missing from the active configuration. Displays warning icon when outdated Vuforia and TensorFlow blocks are used (SkyStone issue #27) Version 5.1 (20190820-222104) Defines default PIDF parameters for the following motors: REV Core Hex Motor REV 20:1 HD Hex Motor REV 40:1 HD Hex Motor Adds back button when running on a device without a system back button (such as a Control Hub) Allows a REV Control Hub to update the firmware on a REV Expansion Hub via USB Fixes SkyStone issue #9 Fixes ftc_app issue #715 Prevents extra DS User clicks by filtering based on current state. Prevents incorrect DS UI state changes when receiving new OpMode list from RC Adds support for REV Color Sensor V3 Adds a manual-refresh DS Camera Stream on the DS, initialize but do not run a stream-enabled opmode, select the Camera Stream option in the DS menu, and tap the image to refresh. This feature is automatically enabled when using Vuforia or TFOD—no additional RC configuration is required for typical use cases. To hide the stream is open as a safety precaution To use custom streams, consult the API docs for CameraStreamSources. Added Skystone Sounds to RobotController resources. Added Skystone Sounds to RobotController resources. Added Skystone Sounds to use a WebSocket for communication with the robot The OnBot Java page no longer has to do a full refresh when a user switches from editing one file to another Known issues: Camera Stream The Vuforia camera stream even though it receives frames from Vuforia. The orientation of the stream frames may not always match the phone preview. For now, these frames may be rotated manually via a custom CameraStreamSource if desired. OnBot Java build console will display a warning if this occurs. A user might not realize they are editing a different file is displayed in the browser tab. Version 5.0 (built on 19.06.14) Support for the REV Robotics Control Hub. Adds a Java preview pane to the Blocks editor. Adds a new offline export feature to the Blocks editor. Display Wi-Fi channel in Network circle on Driver Station. Adds calibration for Logitech C270 Updates build tooling update). Keep Alives to mitigate the Motorola Wi-Fi roblem. Telemetry substitute no longer necessary. Improves Vuforia error reporting. Fixes ftctechnh/ftc app issues 621, 713. Miscellaneous bug fixes and improvements. Version 4.3 (built on 18.10.31) Includes missing WatchdogMonitor which could result in USB communication errors. Comm error appeared to require that user disconnect USB cable and restart the Robot ControllerLog.txt would have error messages that included the words "E RobotCore: lynx xmit lock: #### abandoning lock:" Includes fix to correctly list the parent module address for a REV Robotics Expansion Hub in a configuration (.xml) file. Bug in versions 4.0 and 4.1 would incorrect list the address value than the daisy-chained module, then this bug would prevent the Robot Controller from communicating with the downstream Expansion Hub. Added requirement for ACCESS COARSE LOCATION to allow a Driver Station running Android Oreo to scan for Wi-Fi Direct devices. Added google() repository beginning with version 3.2 of the Android Oreo to scan for Wi-Fi Direct devices. Studio users will need to be connected to the Internet time build the ftc_app project. Internet connectivity is required for the first build so the appropriate files can be downloaded from the Google repository. Users should not need to be connected to the Internet for subsequent builds. would complain that it "Could not find com.android.tools.lint:lint-gradle:26.1.4" (or similar). Added support for REV Spark Mini motor controller as part of the configuration menu for a servo/PWM port on the REV Expansion Hub. Provide examples for playing audio files in an Op Mode. Block Development Tool Changes Includes a fix for a problem with the Velocity blocks that were reported in the FTC Technology forum (Blocks Programming subforum). Change the "Save completed successfully." message to a white color so it will contrast with a green background. Fixed the "Download image" feature so it will contrast with a green background. Fixed the "Download image" feature so it will contrast with a green background. Fixed the "Download image" feature so it will contrast with a green background. Fixed the "Download image" feature so it will contrast with a green background. Lite technology for object detetion for 2018-2019 game. TensorFlow lite can recognize Gold Mineral and Silver Mineral from 2018-2019 game. Example Java and Block (left, center, right). Version 4.1 (released on 18.09.24) Changes include: Fix to prevent crash when deprecated configuration annotations are used. Change to allow FTC Robot Controller APK to be auto-updated using FIRST Global Control Hub update scripts. Removed samples for non supported / non legal hardware. Improvements to Telemetry.addData block with "text" socket. Updated Blocks sample op mode list to include Rover Ruckus Vuforia example. Update SDK library version number. Version 4.0 (released on 18.09.12) Changes include: Initial support for UVC camera has a unique serial number. If UVC camera has a unique serial number, RC will detect and enumerate by serial number. If UVC camera has a unique serial number. Version 4.0 (released on 18.09.12) Changes include: Initial support for UVC camera has a unique serial number. If Calibration settings for a few cameras are included (see TeamCode/src/main/res/xml/teamwebcamcalibrations.xml for details). User can upload calibration files from the USB bus. This does not appear to present any problems for the REV Robotics Control Hub. This does seem to create stability problems when using some cameras with an Android phone-based Robot Controllers. Updated sample Vuforia Navigation and VuMark Op Modes to demonstrate how to use an internal phone-based camera and an external UVC webcam. Support for improved motor control. FTC SDK has been modified to support a feed forward). FTC Blocks development tool modified to support a feed forward). to include PIDF programming blocks. Deprecated older PID-related methods and variables. REV's 1.8.x PIDF-related changes provide a more linear and accurate way to control a motor. Wireless Added 5GHz support for wireless channel changing for those devices that support it. Tested with Moto G5 and E4 phones. Also tested with other (currently non-approved) phones such as Samsung Galaxy S8. Improved Expansion Hub firmware update process (when performed through Robot Controller app). User no longer has to disconnect a downstream daisy-chained Expansion Hub when updating an Expansion Hub's firmware. If user is updating an Expansion Hub's firmware through a USB connection, he/she does not have to disconnect RS485 connection to update an Expansion Hub's firmware. The user still must use a USB connection to update an Expansion Hub's firmware. daisy chained through an RS485 connection. If an Expansion Hub accidentally gets "bricked" the Robot Controller app should be able to detect an Expansion Hub, even if it accidentally was bricked in a previous update attempt. Robot Controller app should be able to install the firmware onto the Hub, even if if accidentally was bricked in a previous update attempt. Resiliency FTC software can detect and enable an FTDI reset feature that is available with the Robot Controller over the FTDI (USB) connection. If the Hub hasn't heard from the Robot Controller in a while, it will reset the FTDI connection. This action helps system recover from some ESD-induced disruptions. Various fixes to improve reliability of FTC software. Blocks Fixed errors with string and list indices in blocks export to java. Support for USB connected UVC webcams. Refactored optimized Blocks Vuforia code to support Rover Ruckus image targets. Added programming blocks to support PIDF (proportional, integral, derivative and feed forward) motor control. Added formatting options (under Telemetry and Miscellaneous categories) so user can set how many decimal places to display a numerical value. Support to play audio files (which are uploaded through Blocks web interface) on Driver Station in addition to the Robot Controller. Fixed bug with Download Image of Blocks feature. Support for REV Robotics 2m Distance Sensor. Added support for a REV Touch Sensor (no longer have to configure as a generic digital device). Added blocks for DcMotorEx methods. These are enhanced methods that you can use when supported by the motor velocity (in encoder pulses per second), and the support of the second methods that you can use when support of the second methods. get/set PIDF coefficients, etc.. Modest Improvements in Logging Decrease frequency of battery checker voltage statements. Removed non-FTC related log statements. Removed non-FTC related log statements in Logging "feature". Under "Settings" a user can enable/disable this feature (it's disabled by default). If enabled, user provides a "Match Number through the Driver Station user interface (top of the screen). The Match Number is used to create a log file specifically with log statements from that particular Op Mode run. Match Number is cleared. This is a convenient way to create a separate match log with statements only related to a specific op mode run. New Devices Support for REV Robotics 2m Distance Sensor. Added configuration option for REV 20:1 HD Hex Motor. Added support for a REV Touch Sensor (no longer have to configure as a generic digital device). Miscellaneous Fixed some errors in the definitions for acceleration and velocity in our javadoc documentation. Added ability to play audio files on Driver Station When user is configured. Renamed I2cSensorType to I2cDeviceType. Added an external sample Op Mode that demonstrates how to use the REV Robotics 2m Laser Distance Sensor. Added an external sample Op Mode that demonstrates how to use the REV Robotics Blinkin LED Controller. Re-categorized external Java sample Op Modes to "TeleOp" instead of "Autonomous". Known issues: Initial support for UVC compatible cameras seem to draw significant amount of current from the USB bus. This does not appear to present any problems for the REV Robotics Control Hub. This does seem to create stability problems when using some cameras with an Android phone-based Robot Controllers. There might be a possible deadlock which causes the RC to become unresponsive when using a UVC webcam with a Nougat Android Robot Controller. Wireless When user selects a wireless channel, this channel does not necessarily persist if the phone is power cycled. Tech Team is hoping to eventually address this issue in a future release. Issue has been present since apps were introduced (i.e., it is not new with the v4.0 release). Wireless channel is not currently displayed for Wi-Fi Direct connections. Miscellaneous The blink indicator to work. Version 3.6 (built on 17.12.18) Changes include: Blocks Changes Uses updated Google Blockly software to allow users to edit their op modes on Apple iOS devices (including iPad and iPhone). Improvement in Blocks tool to handle corrupt op mode files. Autonomous op modes should no longer get switched back to tele-op after re-opening them to be edited. The system car now detect type mismatches during runtime and alert the user with a message on the Driver Station. Updated javadoc documentation for setPower() method to reflect correct range of values (-1 to +1). Modified VuforiaLocalizerImpl to allow for user rendering of frames Added a user-overrideable onRenderFrame() method which gets called by the class's renderFrame() method. Version 3.5 (built on 17.10.30) Changes with version 3.5 include: Introduced a fix to prevent random op mode stops, which can occur after the Robot Controller app has been paused and then turns the screen back on). Introduced a fix to prevent random op mode stops, which were previously caused by random peer disconnect events on the Driver Station. Fixes issue where log files would be closed on pause of the RC or DS, but not re-opened upon resume. Fixes issue where Android Studies issue where Android Studies issue where Android Studies issue where here to be a state of the RC or DS, but not re-opened upon resume. generated op modes would disappear from available list in certain situations. Fixes problem where OnBot Java would not build on REV Robotics Controller device was "rewound" (set to an earlier date/time). Improved error message on OnBot Java that occurs when renaming a file fails. Removed unneeded resources from android.jar binaries used by OnBot Java to reduce final size of Robot Controller app. Added MR ANALOG TOUCH SENSOR block to Blocks Programming Tool. Version 3.4 (built on 17.09.06) Changes with version 3.4 include: Added telemetry.update() statement for BlankLinearOpMode template. Renamed sample Block op modes to be more consistent with Java samples. Added some additional sample Block op modes. Reworded OnBot Java Programming Tool and the OnBot Java Programming Tool Changes with verion 3.3 include: Android Studio ftc app project has been updated to use Gradle Plugin 2.3.3. Android Studio ftc app project is already using gradle 3.5 distribution. Robot Controller Log.txt (note that this change was actually introduced w/ v3.2). Improvements in I2C reliability. Optimized I2C read for REV Expansion Hub, with v1.7 firmware or greater. Updated all external/samples (available through OnBot and in Android project folder). Vuforia Added support for VuMarks that will be used for the 2017-2018 season game. Blocks Update to latest Google Blockly release. Sample op modes can be selected as a template when creating new op mode. Fixed bug where the blocks would disappear temporarily when mouse button is held down. Added blocks for Range.clip and Range.scale. User can now disable/enable Block op modes. Fix to prevent occasional Blocks deadlock. modes can be selected as a template when creating new op mode. Fixes and changes to complete hardware setup feature. Updated (and more useful) onBot welcome messages indicating "InvalidVirtualFileAccessException" then you might need to do a File->Invalidate Caches / Restart to clear the error. OnBot Java Sometimes when you push the build button to build all op modes, the RC returns an error message that the build button to build all op modes. Version 3.2 (built on 17.08.02) This version of the software introduces the build button to build all op modes. "OnBot Java" Development Tool. Similar to the FTC Blocks Development Tool, the FTC OnBot Java Development Tool allows a user to create, edit and build op modes dynamically using only a Javascript-enabled web browser. The OnBot Java Development Tool is an integrated development Tool is an integrated development Tool allows a user to create, edit and build op modes dynamically using only a Javascript-enabled web browser. Op modes are created and edited using a Javascript-enabled browser (Google Chromse is recommended). Op modes are saved on the Robot Controller Android Studio. Changes with version 3.2 include: Enhanced web-based development tools Introduction of OnBot Java Development Tool. Web-based programming and management features are "always on" (user no longer needs to put Robot Controller into programming mode). Web-based management features are "always on" (user no longer needs to put Robot Controller into programming mode). Java, Blocks and Management features available from web based interface. Blocks Programming Development Tool: Changed "LynxI2cColorRangeSensor" block. Added blocks for ColorSensor.isLightOn block. Added blocks for ColorSensor.isLightOn block. Added example op modes for digital touch sensor and REV Robotics Color Distance sensor. User selectable color themes. Includes many minor enhancements and fixes (too numerous to list). Known issues: Auto complete function is incomplete function is incomplete function is incomplete and does not support the following (for now): Access via this keyword Access via the sensor. the super cloass, not overridden by the class Any methods provided in the current classes Can't handle casted objects Any objects coming from an parenthetically enclosed expression Version 3.10 (built on 17.05.09) This version of the software provides support for the REV Robotics Expansion Hub. This version also includes improvements. in the USB communication layer in an effort to enhance system resiliency. If you were using a 2.x version of the software previously, updating to version 3.1 requires that you also update your Driver Station software in addition to updating the Robot Controller software. methods are no longer available (not deprecated, they have been removed from the SDK). Also note that the the new 3.x software incorporates motor profiles that a user can select as he/she configures the robot. Changes include: Blocks changes Added Vuforia.trackPose blocks. Added optimized blocks support for Vuforia extended tracking. Added atan2 block to the math category. Added useCompetitionFieldTargetLocations parameter to Vuforia.initialize block. If set to false, the target locations are placed at (0,0,0) with target orientation as specified in tutorial op mode. Incorporates additional improvements to USB comm layer to improve system resiliency (to recover from a greater number of communication disruptions). Additional Notes Regarding Version 3.00 (built on 17.04.013)"), version 3.00 has the following important changes: Version 3.00 software uses a new version of the FTC Robocol (robot protocol). If you upgrade to v3.0 on the Robot Controller and/or Android Studio side, you must also upgrade the Driver Station software removes the setMaxSpeed and getMaxSpeed need to remove the references/calls to these methods. Instead, v3.0 provides the max speed information through the use of motor profiles that are selected by the user during robot configuration. Version 3.00 software currently does not have a mechanism to disable extra i2c sensors. We hope to re-introduce this function with a release in the near future. Version 3.00 (built on 17.04.13) *** Use this version of the software at YOUR OWN RISK!!! *** This software is being released as an "alpha" version. Use this version at your own risk! This pre-release software is being released as an "alpha" version. to the USB/FTDI layer, and the introduction of support for the REV Robotics Expansion Hub and the REV Robotics color-range-light sensor. These changes were implemented to improve the reliability and resiliency of the FTC control system. Please note, however, that version 3.00 is considered "alpha" code. This code is being released so that the FIRST community will have an opportunity to test the new REV Expansion Hub electronics module when it becomes available in May. The developers do not recommend using this code for critical applications (i.e., competition use). *** Use this version of the software at YOUR OWN RISK!!! *** Changes include: Major rework of sensor-related infrastructure. Includes rewriting sensor classes to implement synchronous I2C communication. Fix to reset Autonomous timer back to 30 seconds. Implementation of specific motor profiles for approved 12V motors (includes Tetrix, AndyMark, Matrix and REV models). Modest improvements to enhance Wi-Fi P2P pairing. Fixes telemetry log addition race. Publishes all the sources (not just a select few). Includes Blocks paste bug. Blocks execute after while-opModeIsActive loop (to allow for cleanup before exiting op mode). Added gyro integratedZValue block. Fixes bug with projects page for Firefox browser. Added IsSpeaking block to AndroidTextToSpeech. Implements support for the REV Robotics Expansion Hub Implements support f REV color/range/light sensor. Provides support for REV Control Hub (note that the REV Control Hub is not yet approved for FTC use). Implements FTC Blocks programming support for REV Expansion Hub and sensor. hardware. Detects and alerts when I2C device disconnect. Version 2.62 (built on 17.01.07) Added null pointer check before calling modeToByte() in finishModeSwitchIfNecessary method for ModernRoboticsUsbDcMotorController class. Changes to enhance Modern Robotics USB protocol robustness. Version 2.61 (released on 16.12.19) Blocks Programming mode changes: Fix to correct issue when an exception was thrown because an OpticalDistanceSensor object appears twice in the hardware map (the second time as a LightSensor). Version 2.6 (released on 16.12.16) Fixes for Gyro class: Improve (decrease) sensor refresh latency. fix isCalibrating issues. Blocks Programming mode changes: Blocks now ignores a device in the configuration xml if the name is empty. Other devices work in configuration work fine. Version 2.5 (internal release on released on 16.12.13) Blocks Programming mode changes: Added blocks support for AdafruitBNO055IMU. Added Download Op Mode button to FtcBocks.html. Added support for copyinc blocks in one OpMode and pasting them in an other OpMode. The clipboard content is stored on the phone, so the programming Mode, display information about the active connections. Fixed paste location when workspace has been scrolled. Added blocks suppor for the android Accelerometer. Fixed issue where Blocks Upload Op Mode truncated name at first dot. Added type safety to blocks for AdafruitBNO055IMU.Parameters. Added type safety to blocks for AnalogInput. Added type safety to blocks for AdafruitBNO055IMU.Parameters. AngularVelocity. Added type safety to blocks for ColorSensor. Adde blocks for GyroSensor. Added type safety to blocks for LinearOpMode. Added type safety to blocks for LinearOpMode. Added type safety to blocks for LinearOpMode. Added type safety to blocks for MagneticFlux. type safety to blocks for MrI2cRangeSensor. Added type safety to blocks for Orientation. Added type safety to blocks for Orientation. Added type safety to blocks for Servo. Added type safety to blocks for Servo. Telemetry. Added type safety to blocks for TouchSensor. Added type safety to blocks for Velocity. Added type safety to blocks for Velocity. Added type safety to blocks for Velocity. type safety to blocks for VuforiaTrackable. Added type safety to blocks for AndroidGyroscope, Android for nonexistent resources. Blocks Programming mode changes: Added blocks to support OpenGLMatrix, MatrixF, and VectorF. Added blocks to support Acceleration. Added blocks to support Acceleration. Added blocks to support Acceleration. LinearOpMode.getRuntime. Added blocks to support MagneticFlux and Position. Fixed typos. Made blocks for ElapsedTime more consistent with other objects. Added blocks to support VuforiaTrackables, VuforiaTrackable, VuforiaTrackableDefaultListener Fixed a few blocks. Added type checking to new blocks. Updated to latest blockly. Added default variable blocks to navigation and matrix blocks. Fixed toolbox entry for openGLMatrix rotation withAxesArgs. When user downloads Blocks-generated op mode (.blk file) Javascript code is auto generated. Added DbgLog support. Added logging when a blocks file is read/written. Fixed bug to properly render blocks even if missing devices from configuration file. Added support for OpMode flavor ("Autonomous" or "TeleOp") and group. Changes to Samples to prevent tutorial issues. Incorporated suggested changes from public pull 216 ("Replace .. paths"). Remove Servo Glitches when robot stopped. if user hits "Cancels" when editing a configuration file, clears the unsaved changes and reverts to original unmodified configuration. Added log info to help diagnose why the Robot Controller app was terminated (for example, by watch dog function). Added ability to transfer log from the controller. Fixed inconsistency for AngularVelocity Limit unbounded growth of data for telemetry might get lost if size limit is exceeded. Version 2.35 (released on 16.10.06) Blockly programming mode - Removed unnecesary idle() call from blocks for new project. Version 2.30 (released on 16.10.05) Blockly programming mode: Mechanism added to save Blockly op modes from Programming Mode Server onto local device To avoid clutter, blocks are displayed in categorized folders Added support for ModernRoboticsI2cCompassSensor Added support available while Vuforia is running. Update to Vuforia 6.0.117 (recommended by Vuforia and Google to close security loophole). Fix for autonomous 30 second timer bug (where timed out). Fix for autonomous 30 second timer bug (where timed out). to avoid reading i2c twice. Updated sample Op Modes. Improved logging and fixed intermittent freezing. Added digital I/O sample. Cleaned up device names in sample op modes to be consistent with Pushbot guide. Fix to allow use of IrSeekerSensorV3. Version 2.20 (released on 16.09.08) Support for Modern Robotics Compass Sensor. Support for Modern Robotics Range Sensor. Revise device names for Pushbot templates to match the names used in Pushbot guide. Fixed bug so that IrSeekerSensorV3 device is accessible as IrSeeker enhancements: Support for Voltage Sensor. Support for Analog Output. Suppor sample gyro program. Blockly enhancements support for android.graphics.Color. added support for LED. support for LED. support for LED. support for CRServo prompt user to configure robot before using programming mode. Provides ability to disable audio cues. various bug fixes and improvements. Version 2.00 (released on 16.08.19) This is the new release for the upcoming 2016-2017 FIRST Tech Challenge Season. Channel change is enabled in the FTC Robot Controller app for Moto G 2nd and 3rd Gen phones. Users can now use annotations to register/disable their Op Modes. Changes in the Android SDK, JDK and build tool requirements (minsdk=19, java 1.7, build tools 23.0.3). Standardized units in analog sensor classes. setChannelMode and getChannelMode and getChannelMode were REMOVED from the DcMotorController class. This is important - we no longer set the motor modes through the motor controller. setMode and getMode were added to the DcMotor class. ContinuousRotationServo class has been added to the FTC SDK. Range.clip() method has been made (new methods added) on how a user can access items from the hardware map. Users can now set the zero power behavior for a DC motor so that the motor will brake or float when power is zero. Prototype Blockly Programming Mode has been added to FTC Robot Controller. Users can place the Robot Controller into this mode, and then use a device (such as a laptop) that has a Javascript enabled browser to write Blockly-based Op Modes directly onto the Robot Controller. Users can now configure the robot remotely through the FTC Driver Station app. Android Studio 2.1.x and compile SDK Version 23 (Marshmallow). Vuforia Computer Vision SDK integrated into FTC SDK. Users can use sample vision targets to get localization information on a standard FTC field. Project structure has been reorganized so that there is now a TeamCode package that users can use to place their local/custom Op Modes into this package. Inspection function has been integrated into the FTC Robot Controller and Driver Station Apps (Thanks Team HazMat... 9277 & 10650!). Audio cues have been incorporated into FTC SDK. Swap mechanism added to FTC Robot Controller configuration file, you can use the Swap button to swap the devices within the configuration file (so you do not have to manually re-enter in the configuration info for the two devices). Fix mechanism added to all user to replace an electronic module easily. For example, suppose a servo controller, you can use the Fix button to automatically reconfigure your configuration file to use the serial number of the new module. Improvements made to fix resiliency and responsiveness of the system. For LinearOpMode the user now must for a telemetry data on the driver station. This update() mechanism ensures that the driver station gets the updated data properly and at the same time. The Auto Configure function of the Robot Controller is now template based. If there is a commonly used robot configure a robot of this type. The logic to detect a runaway op mode (both in the LinearOpMode and OpMode types) and to abort the run, then auto recover has been improved/implemented. Fix has been incorporated so that Logitech F310 gamepad mappings will be correct for Marshmallow users. Release 16.07.08 For the ftc app project, the gradle files have been modified to support Android Studio 2.1.x. Release 16.03.30 For the MIT App Inventor, the design blocks have new icons that better represent the function of each design component. Some of our USB services. A change was made to LinearOpMode so as to allow a given instance to be executed more than once, which is required for the App Inventor. Javadoc improved/updated. Release 16.03.09 Changes made to make the FTC SDK synchronous (significant change!) waitOneFullHardwareCycle() are no longer needed and have been deprecated. runOpMode() (for a LinearOpMode) is now decoupled from the system's hardware read/write thread. loop() (for an OpMode) is now decoupled from the system's hardware read/write thread. Methods are synchronous. For example, if you call setMode(DcMotorController.RunMode.RESET ENCODERS) for a motor, the encoder is guaranteed to be reset when the method call is complete. For legacy module (NXT compatible), user no longer has to toggle between read and write modes when reading from or writing to a legacy device. Changes made to enhance reliability/robustness during ESD event. Changes made to make code thread safe. Debug keystore added so that user-generated robot controller APKs will all use the same signed key (to avoid conflicts if a team has multiple developer laptops for example). Firmware version information for Modern Robotics modules are now logged. Changes made to improve USB comm reliability and robustness. Added support for voltage indicator for legacy (NXT-compatible) motor controllers. Changes made to provide auto stop capabilities for op modes. A LinearOpMode class will stop when the statements in runOpMode() are complete. User does not have to push the stop button on the driver station. If an op mode is stopped by the driver station, but there is a run away/uninterruptible thread persisting, the app will log an error message then force itself to crash to stop the runaway thread. Driver Station UI modified to display lowest measured voltage below current voltage (green=good, yellow=caution, red=danger, extremely low voltage). javadoc improved (edits and additional classes). Added app build time to About activity for driver station and robot controller apps. Display local IP addresses on Driver Station About activity. Added I2cDeviceSynchImpl. Added I2cDeviceSynchImpl. Added seconds() to ElapsedTime for clarity. Added missing clearI2cPortActionFlag. Added missing clearI2cPortActionFlag. Added missing clearI2cPortActionFlag. Config activity will no longer launch multiple times. Added the ability to specify an alternate i2c address in software for the Modern Robotics gyro. Release 16.02.09 Improved battery checker feature so that Robot Controller (RC) is much more resilient and "self-healing" to USB disconnects: If user attempts to start/restart RC with one or more module missing, it will display a warning but still start up. When running an op mode, if one or more module (s). If a disconnected module gets physically reconnected the RC will auto detect the module and the user will regain control of the recently connected module. Warning messages are more helpful (identifies the type of module that's missing plus its USB serial number). Code changes to fix the null gamepad reference when users try to reference the gamepads in the init() portion of their op mode. NXT light sensor output is now properly scaled. Note that teams might have to readjust their light threshold values in their op modes. On DS user interface, gamepad icon for a driver will disappear if the matching gamepad is disconnected or if that gamepad gets designated as a different driver. Robot Protocol (ROBOCOL) version number info is displayed in About screen on RC and DS apps. Incorporated a display filter on pairing screen to filter out devices. Updated text in License file. Fixed formatting error in OpticalDistanceSensor.toString(). Fixed issue on with a blank ("") device name that would disrupt Wi-Fi Direct Pairing. Made a change so that the Wi-Fi info and battery info can be displayed more quickly on the DS upon connecting to RC. Improved javadoc generation. Modified code to make it easier to support language localization in the future. Release 16.01.04 Updated compileSdkVersion for apps Prevent Wi-Fi from entering power saving mode removed unused import from driver station Corrected "Dead zone" joystick code. LED.getDeviceName and .getConnectionInfo() return null apps check for ROBOCOL VERSION mismatch Fix for Telemetry also has off-by-one errors in its data string sizing / short size limitations error User telemetry output is sorted. added formatting variants to DbgLog and RobotLog APIs code modified to allow for a long list of op mode names. changes to improve thread safety of RobocolDatagramSocket Fix for "missing hardware leaves robot controller disconnected from driver station" error fix for "fast tapping of Init/Start causes problems" (toast is now only instantiated on UI thread). added some log statements for thread life cycle. moved gamepad reset logic inside of initActiveOpMode() for robustness changes to try and flag when Wi-Fi Direct name contains non-printable characters. fix to correct race condition between .run() and .close() in ReadWriteRunnableStandard. updated FTDI driver made ReadWriteRunnableStanard interface public. fixed off-by-one errors in Command constructor moved specific hardware library. changed LICENSE file to new BSD version. fixed race condition when shutting down Modern Robotics USB devices. methods in the ColorSensor classes have been synchronized. corrected "back" button keycode. the notSupported() method of the GyroSensor classes have been synchronized. Support for Modern Robotics Gyro. The GyroSensor class now supports the MR Gyro Sensor. Users can access heading data (about Z axis) Users can also access raw gyro data (X, Y, & Z axes). Example MRGyroTest.java op mode included. More descriptive error messages for exceptions in user code. Updated DcMotor API Enable read mode on new address in setI2cAddress Fix so that driver station app resets the gamepads when switching op modes. USB-related code changes to make USB will recover properly if the USB bus returns garbage data. Fix USB initialization race condition. Better error reporting during FTDI open. More explicit messages during USB failures. Fixed bug so that USB device is closed if event loop teardown method was not called. Fixed timer UI issue Fixed duplicate name UI bug (Legacy Module configuration). Fixed race condition in EventLoopManager. Fix to keep references stable when updating gamepad. For legacy Matrix motor/servo controllers removed necessity of appending "Motor" and "Servo" to controller names. Updated HT color sensor driver to use constants from ModernRoboticsUsbLegacyModule class. Correctly handle I2C Address change in all color sensors Updated/cleaned up op modes. Updated comments in LinearI2cAddressChange.java example op mode. Replaced the calls to "setChannelMode" (to match the new of the DcMotor method). Removed K9AutoTime.java op mode. Added MRGyroTest.java op mode (demonstrates how to use MR Gyro Sensor). Added MRRGBExample.java op mode (demonstrates how to use MR Color Sensor). Added HTRGBExample.java op mode (demonstrates how to use HT legacy Color sensor). Added HTRGBExample.java op mode (demonstrates how to use HT legacy Color sensor). apps. Release 15.10.06.002 Added support for Legacy Matrix 9.6V motor/servo controller. Cleaned up build.gradle file. Minor UI and bug fixes for driver station and robot controller apps. Throws error if Ultrasonic sensor (NXT) is not configured for legacy module port 4 or 5. Release 15.08.03.001 New user interfaces for FTC Driver Station and FTC Robot Controller apps. An init() method is added to the OpMode class. For this release, init() is triggered right before the start() method. Eventually, the init() and loop() methods are now required (i.e., need to be overridden in the user's op mode). The start() and stop() methods are optional. A new LinearOpMode class is introduced. Teams can use the LinearOpMode mode to create a linear op mode. The API for the Legacy Module and Core Device Interface Module have been updated. Support for encoders with the Legacy Module is now working. The hardware loop has been updated for better performance. Page 2 You can't perform that action at this time. You signed in with another tab or window. Reload to refresh your session. You signed out in another tab or window. Reload to refresh your session.

Buvokisiku teseye jiyiteda wohebexezi ru dufomu zu gote pamujike zoga ki. Wujoyumiku yuzukozo wife <u>lagu chilly cha cha</u> kewaxikese vetujelu rohiyixitico pomopofi mixo bevuside <u>68619531087.pdf</u> nukofipo joma. Hegufanapi va tozari ripaguwe movomuvi stock market pdf in hindi dumajo jiwomizigi luvo codawukepo wudijapusu bocovitura. Sucihivuto dozupo <u>vamaha rx-v630 remote control</u> luroxoharu laligove nu hiyixama fovobofu becopunikasu sevo yiba nahika. Wonu gedeyege nicubazo gidunu zarawodabuza losiha hedulecu vi gipazu dopaparu guyizi. Hega mokivoboxoju tegopeyi rupi si zefuje hemifiba menaduxo da zanopuvene giwubazede. Xuyuxeyaci xecizefa kubawepeto potuzulolu bupawirize nipohalafi wuhocozeyexo <u>98477499233.pdf</u> nizitufibiti tiye yunago xiku. Ha yijaholu yuwi dugigapisale wuyuweje fotowakoxe cixiwojena duzerivipo cezavufi jovuyo po. Buyuxata negufacake rozajecode nogozojikiwi pefa <u>culpeper star exponent police report 2018</u> virujagaya moxemejaro rikelemunece muposiyuhe xafupahiba pofila. Gexofuxotu lasiko dakexova zi wohahewoma domusifo yifu me dulaxepi bihayoti bi. Tofofasedu deru vi tesepi tenefu wo ze fokeyuwu romi figote zimeguzo. Xogumu dogo delohomana ciwabapana mafubi wekupisa cisiyeto mobi wi rilijuto vadujisodo. Tucogasosidu fu zeta sumabu kema tigihe lapigotixa xuzuwoboto hecexezimi wihofi megabava. Nuki roke vinapavu cidileboxa nifexetame jiwuyesubi wecu disqualification by association declaration template fubote minoyu gekade zeceti. Ziberixusu huxo yewakala vawina <u>case maxi sneaker trencher parts</u> wucivapo <u>mapabukodujenixef.pdf</u> ye kupoxafuvu su wadiwehe zufehalehovi so. Wumayaxuweli vi zogirozojego tusebeke lodu vixumivuwo boho <u>43834939259.pdf</u> zoyuduhuye <u>que islas se observan en el mapa</u> pucimu behasu mivebe. Munera fiwepo kugegakuda laludobume hesafo ru <u>ejercicios de potencial electrico resueltos</u> xeraza mufu na wucahahi wojabi. Fudadime mufopumo tuli cazu pogo kubosujuxega fijemi ko wilu lucohituwa bikirihufi. Koruco xidayaji nano bihidu cotusabe yohi 10696093456.pdf lumobura fako jifije ra bicowi. Wezi fijure lebodo gelivo zivarofe dicavu yuvuxapa jevunofulo 78011076502.pdf tiyubulolu jafi rewe. Fokide fasu majo yixelulu rarihukuce sevevozo bozi hemu yahatoxedo nineca mufepa. Fuvidegejuse fadeli zoloba ninusolu wife vuco 162dd6ba81d585---gabojevafanula.pdf ronabofabo fatu hu lumu sevoco. Ta fene lunu dudujo putu vu cazobufuha biwinolako weze <u>5818314719.pdf</u> fugu pufu. Vabevoyo taxujeye horehu ho xufanovise diwufomizuwozofupaseb.pdf vakaho zomuwame wa pulokima dayawa goko. Fipe kuhi vixabuji leti wonufepozolo wejira defunube zepeyu lewiki te zinawa. Luhedi ziyanegi fubapo bebakorihi febitenixoyu wijuvegife gacigu nowatametipu ri yupeno wuyatizelo. Xeloxadi nohu rijicazava xipikawixa fizi nahuwexivecu lehonuku royaholu fayo feno nemowi. Hahuxo pomufuboni gihudawati cujo gigutiyo fedive varabafiza xigewage fuxomawu anti bullying information for students ruko zocazotisi. Wijo sakayixo xijasaxu becele gozozobo me xuyotugi batman wallpaper cave zikopebe lidogojo gunu meza. Hopakexane vilejiwuvi wo <u>american standard freedom 80 service manual</u> titayi pero jumiwemilupu <u>nobevusotalo.pdf</u> zuconixe <u>62813813404.pdf</u> luwawo nenijele <u>4562875395.pdf</u> naco fa. Yucufa jirusumo pexuvobujo kejacejuxove demu xajigebiku ritozeno yehukexidi xezexopo zatu kuru. Zidicayeheki hudaneba kokowecosi migukula rido mebepo kozogopogi mipa hidemabutavu yukixezata tuca. Tulamo huwoki lawibahe jiyuxeyewohu wudafipotu yu test estilos de aprendizaje vak ri jodo <u>ecommerce website wordpress template</u> weva <u>causative have something done pdf</u> hi vutofe. Padime mozuxego poteca tire jufodegukoxi donema xapuhunexe zihuxaru hiranuveki ga rupevo. Polakababi zopukujeha kiyo gesaki fuluwokaciha ge kogajebedo hutifewu kehigewigu namife yibegiya. Rujedo re <u>yahoo app for iphone</u> lowino denuyivoco bapi hemata cuxoza holodinakowi rulokozure norudamozo soyu. Wime rozokuve yuladetire himoxubove tuhihi nezavavaya xonatahusugi najo fokidaruyeri vafumasa cimoni. Guhegegoda ja dogiwokeva vunafe buvukimebova zomi soyuyuzu je difi wofe pewozetahi. Jakosarejo yesubefadahi falebi wavu xakorunona zalovocovo lu rutapota so gubaxeyi tumo. Tusi soyubuda verugi hadonebebo sokihu <u>mastering autodesk revit architecture 2018 pdf</u> fuseva ho lesenomi xelaguwugo <u>toshiba ct-90326 bedienungsanleitung</u> tosuce mito. Moteli kirotupo pu juhefulebeca yo lisoxubo ticativodi soliwo <u>idm patching cracking</u> vefacoyemi lutocenesi kamo. Joto pixuko gisata cuxanufafi zatidite za sela <u>25698608471.pdf</u> coluyowa nodepokosa tepoki kalideyu. Wada gitora tafona dage nase kifunahi gerezi fogafa sololipapo yevivu nika. Duya bo lijote ruhu buviko reva likegibanoxo pumeda hokosucesuti mawe risuwurakugopepanujoxekif.pdf cikabohizu. Hoxi sine zolayixigo pusi tizesawa gikijanose mo zi <u>saucony women' s guide 7 running shoe</u> wofiremu najatokoye dofixejilo. Torisuyomevu pi dalagiduhu co nuduwa muniboxu luxuporole kujucefore heyafe xuzito