Programming in FTC with Android Studio:
Overview

Presented by #8668 Error 404: Team Name Not Found

Welcome to Programming in FTC: Overview presented by Team 8668 Error 404: Team Name Not Found. This is the second of three workshops on programming basics being presented by Error 404 this week.

Forewarning: These classes are aimed towards team who will be programming with Java in Android Studio. I believe that there is/was/will be another class focused on Blocky, but Error 404 will not be teaching that one and I honestly have no idea.
hope to have blocky class in second set of classes
So with that, we're ready to roll.

Agenda

- Control Systems
- Phones/Electronics
- Hardware Configuration
- Programming
- Creating and Running an OpMode

Here’s what we’ll be covering today and tomorrow. We’re just going to go until time runs out and start up again tomorrow.

Control Systems

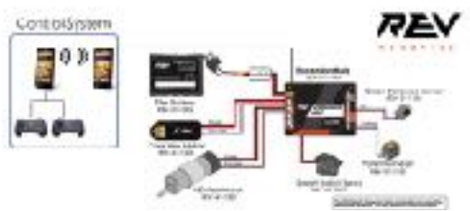
Control Systems - Overview

- Controls Individual Robot Components

First we’re going to talk about the Robot Control System. Now technically FIRST classifies the control system is just the phones and gamepads, but everyone else uses the term to describe everything. The Control System is basically what tells all the motors and servos what to do and what receives data from the sensors.

Control Systems - Overview

- Controls Individual Robot Components
- Two Legal Control Systems in FTC
  - Modern Robotics
  - REV



So the Control System is what is used to run the robot. The phones relay instructions, but the phones can’t connect directly to motors and servos. There needs to be a middle man to connect to the phone and to the motors, servos, and sensors. This middle man is primarily what we think of as the control system. FIRST allows two control systems for FTC: Modern Robotics and REV.

Modern Robotics - Overview

- Uses a Series of Modules to Control Different Aspects of the Robot

First we'll cover the Modern Robotics stuff. The Modern Robotics Control System, as you can see, is quite sprawling. It uses a bunch of different modules to control motors, servos, and sensors. ***NEXT*** The Power Distribution Module connects to the phone and battery and sends power out to up to seven control modules. These modules include the Device Interface Module (for sensors), the CORE Legacy Module (for super outdated sensors like LEGO NXT stuff), the CORE Servo Controller (for servos), and the CORE Motor Controller (for DC Motors). These various modules have a limited number of slots so often teams end up having to use two of each module. This is especially true for the motor and servo controllers as teams typically require a large number of each. All of these Modules have to be mounted on the robot and having so many modules drastically increases the amount of wiring and consumes valuable space inside limited volume of the robot.

Lastly, as of this coming Season, the FIRST is no longer supporting the Modern Robotics Control System. What does this mean? Basically it means that FIRST won't be releasing anymore software updates for the Modern Robotics stuff. Translation: FIRST is beginning to phase out the Modern Robotics stuff in favor of the newer REV system.

Modern Robotics - Overview

- Uses a Series of Modules to Control Different Aspects of the Robot
  - Core Power Distribution module has 6 12V Outputs as well as well as slots for up to 7

The Power Distribution Module connects to the phone and battery and sends power out to up to seven control modules. These modules include the ***NEXT*** CORE Motor Controller (for DC Motors), ***NEXT*** the CORE Servo Controller (for servos), ***NEXT*** the Device Interface Module (for sensors), and the CORE Legacy Module (for super outdated sensors like LEGO NXT stuff)

These various modules have a limited number of slots so often teams end up having to use two of each module. This is especially true for the motor and servo controllers as teams typically require a large number of each. ***NEXT*** All of these Modules have to be mounted on the robot and having so many modules drastically increases the amount of wiring and consumes valuable space inside limited volume of the robot.

Lastly, as of this coming Season, the FIRST is no longer supporting the Modern Robotics Control System. What does this mean? Basically it means that FIRST won't be releasing anymore software updates for the Modern Robotics stuff. Translation: FIRST is beginning to phase out the Modern Robotics stuff in favor of the newer REV system.

Modern Robotics - Overview

- Uses a Series of Modules to Control Different Aspects of the Robot
  - Core Power Distribution module has 6 12V Outputs as well as well as slots for up to 7
    - motor

CORE Motor Controller (for DC Motors), ***NEXT*** the CORE Servo Controller (for servos), ***NEXT*** the Device Interface Module (for sensors), and the CORE Legacy Module (for super outdated sensors like LEGO NXT stuff)

These various modules have a limited number of slots so often teams end up having to use two of each module. This is especially true for the motor and servo controllers as teams typically require a large number of each. ***NEXT*** All of these Modules have to be mounted on the robot and having so many modules drastically increases the amount of wiring and consumes valuable space inside limited volume of the robot.

Lastly, as of this coming Season, the FIRST is no longer supporting the Modern Robotics Control System. What does this mean? Basically it means that FIRST won't be releasing anymore software updates for the Modern Robotics stuff. Translation: FIRST is beginning to phase out the Modern Robotics stuff in favor of the newer REV system.

Modern Robotics - Overview

- Uses a Series of Modules to Control Different Aspects of the Robot
  - Core Power Distribution module has 6 12V Outputs as well as well as slots for up to 7
    - motor
    - servo

the CORE Servo Controller (for servos), ***NEXT*** the Device Interface Module (for sensors), and the CORE Legacy Module (for super outdated sensors like LEGO NXT stuff)

These various modules have a limited number of slots so often teams end up having to use two of each module. This is especially true for the motor and servo controllers as teams typically require a large number of each. ***NEXT*** All of these Modules have to be mounted on the robot and having so many modules drastically increases the amount of wiring and consumes valuable space inside limited volume of the robot.

Lastly, as of this coming Season, the FIRST is no longer supporting the Modern Robotics Control System. What does this mean? Basically it means that FIRST won't be releasing anymore software updates for the Modern Robotics stuff. Translation: FIRST is beginning to phase out the Modern Robotics stuff in favor of the newer REV system.

Modern Robotics - Overview

- Uses a Series of Modules to Control Different Aspects of the Robot
  - Core Power Distribution module has 6 12V Outputs as well as well as slots for up to 7
    - motor
    - servo
    - or device interface controllers

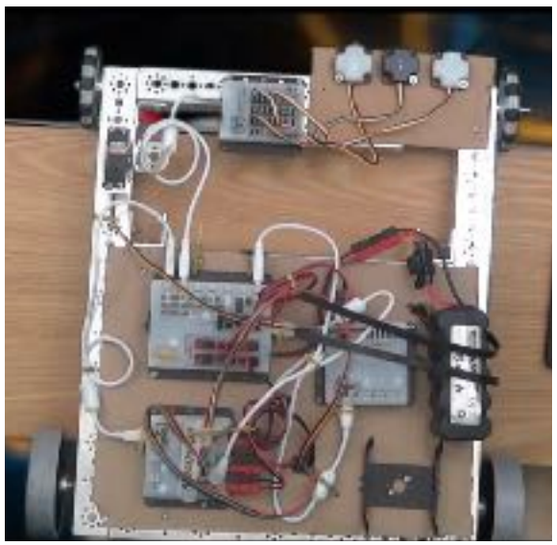
the Device Interface Module (for sensors), and the CORE Legacy Module (for super outdated sensors like LEGO NXT stuff)

These various modules have a limited number of slots so often teams end up having to use two of each module. This is especially true for the motor and servo controllers as teams typically require a large number of each. ***NEXT*** All of these Modules have to be mounted on the robot and having so many modules drastically increases the amount of wiring and consumes valuable space inside limited volume of the robot.

Lastly, as of this coming Season, the FIRST is no longer supporting the Modern Robotics Control System. What does this mean? Basically it means that FIRST won’t be releasing anymore software updates for the Modern Robotics stuff. Translation: FIRST is beginning to phase out the Modern Robotics stuff in favor of the newer REV system.

Modern Robotics - Overview

- Uses a Series of Modules to Control Different Aspects of the Robot
  - Core Power Distribution module has 6 12V Outputs as well as well as slots for up to 7
    - motor
    - servo
    - or device interface controllers
- Takes a lot of space



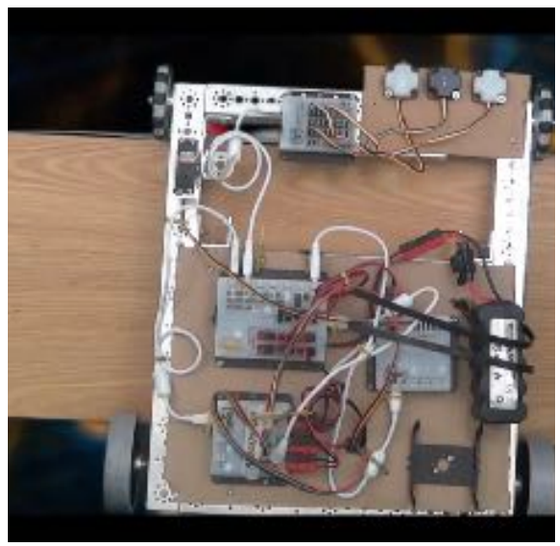
All of these Modules have to be mounted on the robot and having so many modules drastically increases the amount of wiring and consumes valuable space inside the limited volume of the robot.

NEXT

Lastly, as of this coming Season, the FIRST is no longer supporting the Modern Robotics Control System. What does this mean? Basically it means that FIRST won't be releasing anymore software updates for the Modern Robotics stuff. Translation: FIRST is beginning to phase out the Modern Robotics stuff in favor of the newer REV system.

Modern Robotics - Overview

- Uses a Series of Modules to Control Different Aspects of the Robot
  - Core Power Distribution module has 6 12V Outputs as well as well as slots for up to 7
    - motor
    - servo
    - or device interface controllers
- Takes a lot of space
- No Longer Supported by FIRST



Lastly, as of this coming Season, the FIRST is no longer supporting the Modern Robotics Control System. What does this mean? Basically it means that FIRST won't be releasing anymore software updates for the Modern Robotics stuff. Translation: FIRST is beginning to phase out the Modern Robotics stuff in favor of the newer REV system. They did this with NXT, so I anticipate Modern Robotics being completely phased out next season.

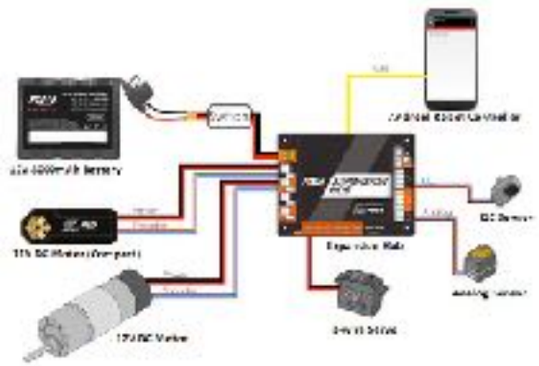
Modern Robotics - Overview

- Uses a Series of Modules to Control Different Aspects of the Robot
  - Core Power Distribution module has 6 12V Outputs as well as well as slots for up to 7
    - motor
    - servo
    - or device interface controllers
- Takes a lot of space
- No Longer Supported by FIRST
- Teams using M.R. are advised to stock up on Advil as extensive headaches will follow use of the M.R. Control System :)

Oh, and you'll want some Advil too

REV - Overview

- One Central Box



The Second legal Control System option, is the REV Control System. It uses the REV Expansion Hub and dispenses with the bakers dozen of modules that the Modern Robotics system uses.

NEXT

REV - Overview

- One Central Box
- 1 REV Expansion Hub =

One REV Expansion Hub is the equivalent of
NEXT

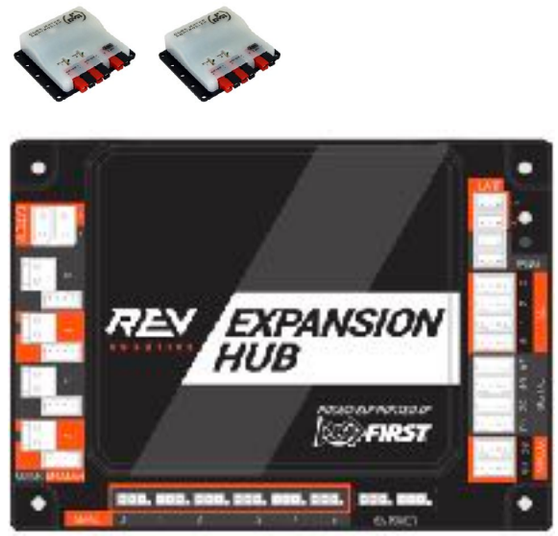
REV - Overview

- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module

1 power distribution module
NEXT

REV - Overview

- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module
  - 2 Motor Controllers



2 motor controllers

NEXT

REV - Overview

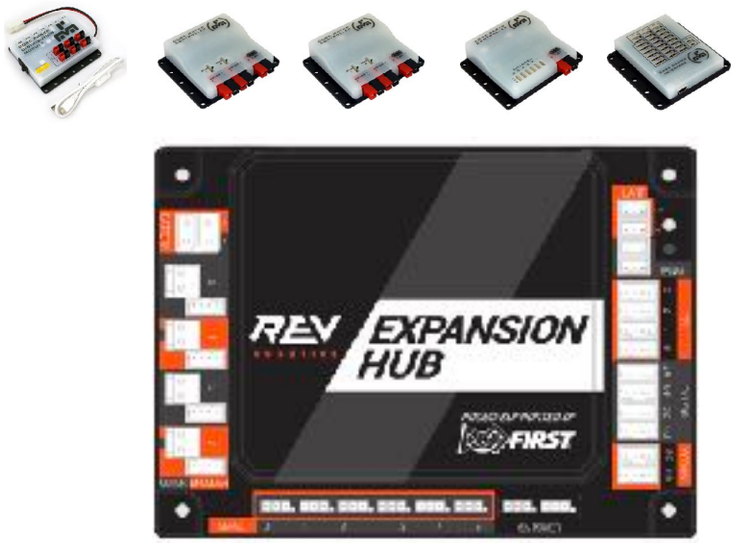
- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module
  - 2 Motor Controllers
  - 1 Servo Controller

1 servo controller

NEXT

REV - Overview

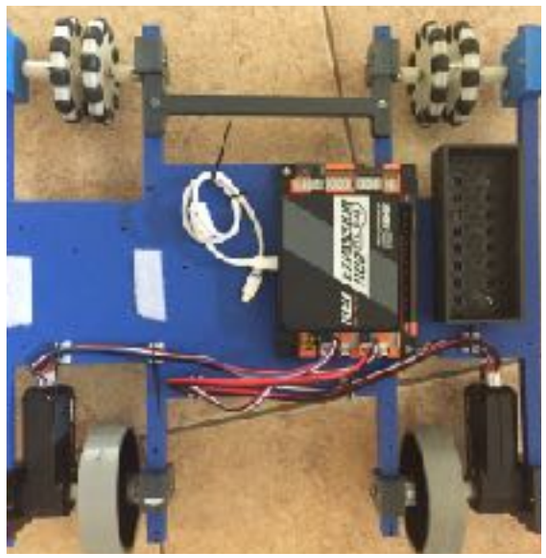
- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module
  - 2 Motor Controllers
  - 1 Servo Controller
  - 1 Device Interface Module



and 1 device interface module
NEXT

REV - Overview

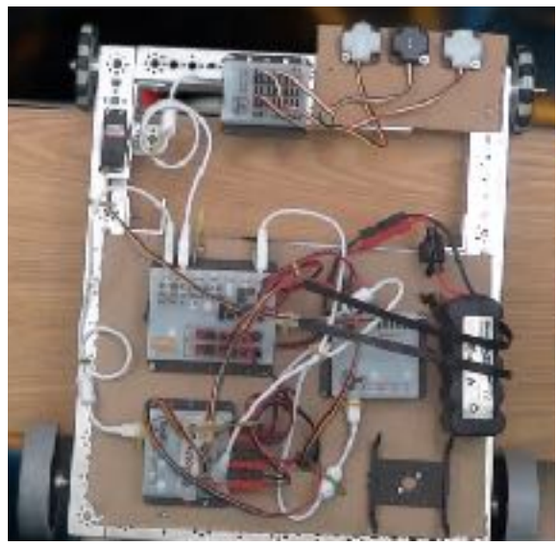
- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module
  - 2 Motor Controllers
  - 1 Servo Controller
  - 1 Device Interface Module
- Compact



The Rev Expansion Hub is compact and super easy to store away inside the robot. You can see that the encoders and motor power wires plug straight into the REV Hub.

REV - Overview

- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module
  - 2 Motor Controllers
  - 1 Servo Controller
  - 1 Device Interface Module
- Compact



This in contrast to the Modern Robotics system. Obviously the robot shown has more on it than the REV one had, but you get the idea. Just to have two motors, one servo, and three sensors requires four separate modules that all have to have interconnecting wires and yeah. You can see how it multiplies quickly.

REV - Overview

- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module
  - 2 Motor Controllers
  - 1 Servo Controller
  - 1 Device Interface Module
- Compact
- Half the Price

Interestingly, the REV Hub comes in at half the price of its Modern Robotics equivalent at each Expansion Hub costing $175.
The equivalent in Modern Robotics costs $388

M.R. also sells an Electronics Bundle (1 of each module plus some wires — not equivalent to 1 REV Hub) and costs $455
mr is not an option on the FIRST store anymore.
So you can see that REV is much cheaper than Modern Robotics. Some of y'all may have done the math and figured out that the cost of one REV Hub in Modern Robotics stuff is more than double the price of two REV Hubs.

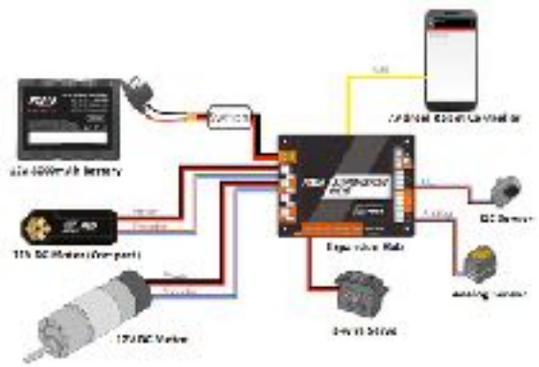
REV - Overview

- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module
  - 2 Motor Controllers
  - 1 Servo Controller
  - 1 Device Interface Module
- Compact
- Half the Price
- Teams can have up to 2 REV Hubs

And on that note, teams are allowed to have up to 2 REV Hubs. The Hubs can be daisy chained together as shown and they share power and essentially become one, so the Robot Controller phone and the battery only have to plug into one Hub, yet are able to use both Hubs.

REV - Overview

- One Central Box
- 1 REV Expansion Hub =
  - 1 Power Distribution Module
  - 2 Motor Controllers
  - 1 Servo Controller
  - 1 Device Interface Module
- Compact
- Half the Price
- Teams can have up to 2 REV Hubs
- Supported by FIRST



The REV Control System is currently the only Control System supported by FIRST. Meaning that FIRST will be releasing software updates periodically to be able to use newer motors, sensors, and servos.

REV - Compatibility

- The REV Hub can support Digital, Analog, I2C, UART, and RS485 type sensor inputs

Going in deeper about the REV Hub, we'll take al look as the Expansion Hub's compatibility with other non-REV stuff. The REV Hub has 4 analog ports, 8 digital ports, 4 I2C buses, 2 RS485 ports, 2 UART ports, and 2 5-volt power output ports.

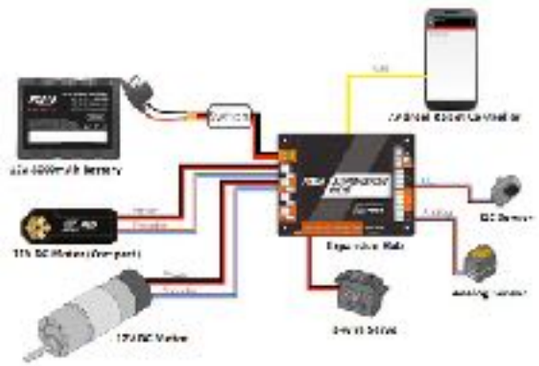
NEXT

Because REV runs sensors at 3.3v and Modern Robotics stuff uses 5v, teams using a REV Control System and Modern Robotics (or any non 3.3v) sensor will need to use a Logic Level Shifter sold by REV to make the transition. But, once the sensor voltage is shifted, pretty much everything sensor-wise Modern Robotics or anyone else sells works with the REV hub.

you may have questions about the different kinds of wiring we'll talk about it later

REV - Compatibility

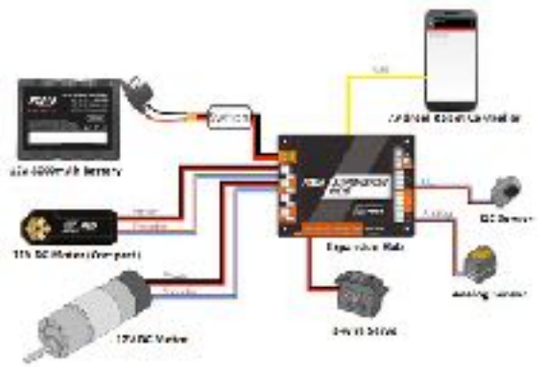
- The REV Hub can support Digital, Analog, I2C, UART, and RS485 type sensor inputs
- Modern Robotics Sensors are Compatible with REV
  - Level Shifter Required



Because REV runs sensors at 3.3v and Modern Robotics stuff uses 5v, teams using a REV Control System and Modern Robotics (or any non 3.3v) sensor will need to use a Logic Level Shifter sold by REV to make the transition. But, once the sensor voltage is shifted, pretty much everything sensor-wise Modern Robotics or anyone else sells works with the REV hub.

REV - Compatibility

- The REV Hub can support Digital, Analog, I2C, UART, and RS485 type sensor inputs
- Modern Robotics Sensors are Compatible with REV
  - Level Shifter Required
- REV is Compatible with non-REV Motors and Servos
  - Level Shifter required for Motors



REV also works with non-dev motors and servos. You will just need the level shifter for the encoders and an adapter for the power wire, but other than that, your pitsco or andymark motors are fine with the REV Hub. Servos on the other hand don’t require a level shifter. You just plug them in.

REV - Usage

- Logic Level Shifter - $3.00
  - SKU: REV-31-1389

So Here’s the Level Shifter. It is required to change the voltage between the REV Hub and third-party sensors. This is because, as I stated earlier, the REV Hub and REV Sensors use 3.3 volts and most other sensors use 5 volts or there about.

new point: Used to change shift voltage from 3.3 to 5. Only needed for non-REV sensors

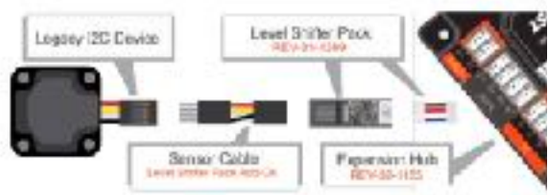
REV - Usage

- Logic Level Shifter - $3.00
  - SKU: REV-31-1389
- Sensor Cable Adapter for Level Shifter - $1.00
  - SKU: REV-31-1384

Because the level shifter has a different pin-in on one side than standard modern robotics sensors, REV sells this city little adapter that changes the pins to match that of the level shifter. The other option of course is to just pull one end of the sensor cable apart and rebuild the connector with the pins in the right order. ***Next***

REV - Usage

- Logic Level Shifter - $3.00
  - SKU: REV-31-1389
- Sensor Cable Adapter for Level Shifter - $1.00
  - SKU: REV-31-1384



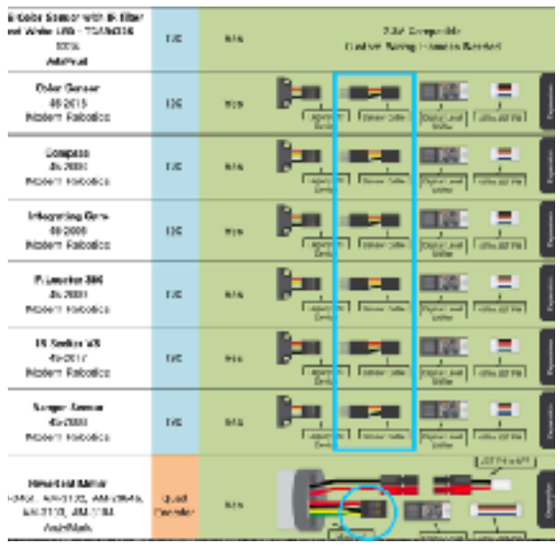
The Top-right is the REV Sensor pinout and the top-left is the pinout for the third-party connector. (reference bottom chart) The only problem with that is that ninety percent of third-party connectors have the GND and Ch B/SCL wires (Black and White) switched (Black on bottom, white on top). Therefore either you have to take the connector apart and reorder the wires or you have to buy REV's little adapter… I know, not fishy at all, but that's the way it is. By the way, these charts are in the REV Hub Guide which is included in the workshop packets.

Next

Here is a chart showing a bunch of Modern Robotics sensors and motor encoder.

REV - Usage

- Logic Level Shifter - $3.00
  - SKU: REV-31-1389
- Sensor Cable Adapter for Level Shifter - $1.00
  - SKU: REV-31-1384



Here is a chart showing a bunch of Modern Robotics sensors and motor encoder. And you'll see that all of them either have to use an adapter or have the wires reordered. Keep in mind that in order to use ANY third party sensor with the REV Hub, you have to use the level shifter. And with the level shifter comes a funky pinout which necessitates the adapter or wire surgery. Last season 404 didn't actually use a whole lot of the adapters as we used all REV motors (and therefore all REV encoder wires) and since we did completely custom wiring, we were able to reorder the wires easily enough in production. And just FYI, for all of those people who just perked their heads up, doing all custom wiring, while space efficient, takes a super long time. And I mean a REALLY long time. I know because it did it all. Again, that's something for your team to decide, but if you have access to them, just use the adapters.

REV - Usage

- Logic Level Shifter - $3.00
  - SKU: REV-31-1389
- Sensor Cable Adapter for Level Shifter - $1.00
  - SKU: REV-31-1384
- FTC Cable Conversion Kit - $25.00
  - SKU: REV-31-1173

REV also sells a cable conversion kit. The kit comes with 3 Level Shifters, 3 REV Sensor Cables, 1 Adapter Cable for the Level Shifter, 4 motor adapters (converts the Anderson Two Pole cable to the REV Motor Type Cable — lets you use your Pitsco and andymark motors with the REV Hub).

REV - Usage

- Logic Level Shifter - $3.00
  - SKU: REV-31-1389
- Sensor Cable Adapter for Level Shifter - $1.00
  - SKU: REV-31-1384
- FTC Cable Conversion Kit - $25.00
  - SKU: REV-31-1173
- Sensor Splitter Cable (2 pck) - $3.00
  - SKU: REV-31-1386

The REV Hub is neat in that each of the Analog and Digital sensor inputs has two channels. Meaning that you can plug two totally separate sensors into the same digital or analog port. To make things easier, REV sells a sensor splitter cable.

REV - Usage

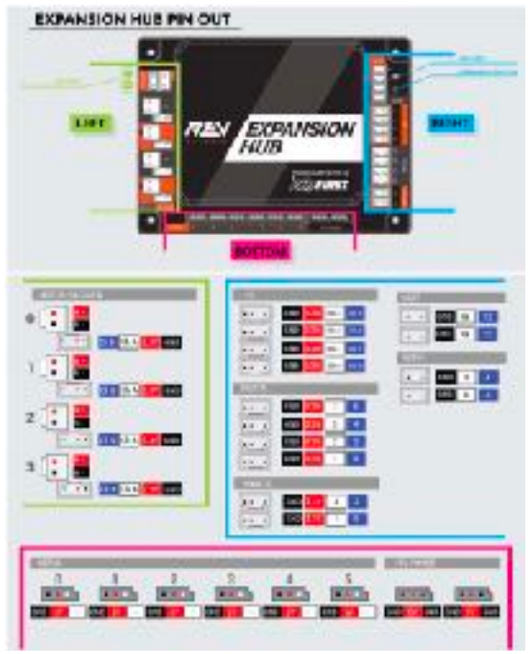
- Logic Level Shifter - $3.00
  - SKU: REV-31-1389
- Sensor Cable Adapter for Level Shifter - $1.00
  - SKU: REV-31-1384
- FTC Cable Conversion Kit - $25.00
  - SKU: REV-31-1173
- Sensor Splitter Cable (2 pck) - $3.00
  - SKU: REV-31-1386
- Anderson Power Pole to XT30 Adapter (2 pck) - $8.00
  - SKU: REV-31-1385

REV sells adapter cables to let teams use their Pitsco 12v Batteries with the REV Hub. REV also sells an adapter that lets teams use their non-REV motors with the REV Hub.

And no, this is not a thirty minute-long advertisement for REV. REV just has a ton of neat, and wickedly useful, ding dongs that I had to include.

REV - Usage

- Pin Out



This is just a pinout chart for the REV Hub. I found it dreadfully useful and referenced it constantly throughout the season. This'll be in the packet we're giving out.

Does anyone have any questions before we move on?

Phones/Electronics

Now I’m going to transition to the more non-robotics electronics. The Phones, the gamepads, and how it all works.

Phones - Which Ones?

- Allowed Phones:

So, which phones are you allowed to use? Well according to the freshly released Rover Ruckus Rules Part 1, these phones:

Phones - Which Ones?

- Allowed Phones:
  - ZTE Speed

The ZTE Speed,

Phones - Which Ones?

- Allowed Phones:
  - ZTE Speed
  - Moto G 2nd Gen

the Motorola Moto-G 2nd Generation,

Phones - Which Ones?

- Allowed Phones:
  - ZTE Speed
  - Moto G 2nd Gen
  - Moto G 3rd Gen

the Motorola Moto-G 3rd Generation,

Phones - Which Ones?

- Allowed Phones:
  - ZTE Speed
  - Moto G 2nd Gen
  - Moto G 3rd Gen
  - Moto G4 Play

the Motorola Moto G4 Play,

Phones - Which Ones?

- Allowed Phones:
  - ZTE Speed
  - Moto G 2nd Gen
  - Moto G 3rd Gen
  - Moto G4 Play
  - Moto G5

the Motorola Moto G5,

Phones - Which Ones?

- Allowed Phones:
  - ZTE Speed
  - Moto G 2nd Gen
  - Moto G 3rd Gen
  - Moto G4 Play
  - Moto G5
  - Moto E4

The Motorola Moto E4,

Phones - Which Ones?

- Allowed Phones:
  - ZTE Speed
  - Moto G 2nd Gen
  - Moto G 3rd Gen
  - Moto G4 Play
  - Moto G5
  - Moto E4
  - Nexus 5

The Google Nexus 5, and

Phones - Which Ones?

- Allowed Phones:
  - ZTE Speed
  - Moto G 2nd Gen
  - Moto G 3rd Gen
  - Moto G4 Play
  - Moto G5
  - Moto E4
  - Nexus 5
  - Galaxy S5

the Samsung Galaxy S5

Gamepads - I Get to Choose?

- Allowed Gamepads:

But wait, you can choose your gamepad too…

Gamepads - I Get to Choose?

- Allowed Gamepads:
  - Logitech F310 Gamepad Controller

FIRST allows the Logitech F310 Gamepoad Controller and

Gamepads - I Get to Choose?

- Allowed Gamepads:
  - Logitech F310 Gamepad Controller
  - Xbox 360 Controller

the Xbox 360 Controller.
They’re both Available on Amazon.

Gamepads - I Get to Choose?

- Allowed Gamepads:
  - Logitech F310 Gamepad Controller
  - Xbox 360 Controller
- Team May Use Any Combination of the Two Legal Gamepads
- Mark Controllers with Team Number

FIRST also says that you can use any combination of the grand total of two gamepad choices. We use two Logitechs, but they're just standard gamepads, so I suppose you could use two different ones. I don't know if using different ones would confuse the Driver's Station at all. The varying controllers may map different, I don't know, I've never used the Xbox one. But there you are. If you feel like it, bait and switch is allowed.

Tip: mark your controllers with your team number. And I mean mark those suckers up. We've got 8668 plastered on in like four different places in silver sharpie on ours.

How It All Connects

- Phones Communicate Via WiFi Direct

So now we have the phones, a control system, gamepads, and android studio. But how does everything work?

How It All Connects

- Phones Communicate Via WiFi Direct
- Driver's Station Tells Robot Controller Which OpMode to Run

Well it goes like this: The Robot Controller actually stores the programs you code and it is the one doing all of the direct interfacing with the robot control System (the REV Hub).

The Robot Controller has all the code on it and can tell the REV Hub to tell the motors, servos, and sensors what to do. But it's actually the Driver's station's job to tell the robot controller which program to do and in the case of a driver controlled program, provide moment by moment control feedback (drive the robot around).

So the Driver's Station tells the Robot Controller which program or OpMode to run.

How It All Connects

- Phones Communicate Via WiFi Direct
- Driver's Station Tells Robot Controller Which OpMode to Run
- Driver's Station Can Relay Commands *from* Gamepads *to* Robot Controller

The Driver's Station can also relay drive commands from the Gamepads to the Robot Controller.

How It All Connects

- Phones Communicate Via WiFi Direct
- Driver's Station Tells Robot Controller Which OpMode to Run
- Driver's Station Can Relay Commands *from* Gamepads *to* Robot Controller
- Robot Controller Runs OpMode and Interfaces with Robot Control System

Either way, once the Robot Controller receives the instructions from the Driver's Station, it interprets those interactions and talks to the Robot Control System (the REV Hub) about which motors and whatnot are supposed to do what.

Does anyone have any questions?

Hardware Configuration

Hardware Configuration. In this section we're going to talk about how to tell the phone what ports on the REV Hub have what plugged into them. You see, the phone can't tell *what* is plugged into the REV Hub, it can only tell that say motor ports 1 and 3 are taken. The Config file actually tells the phone that infant the motor in port 1 is a REV Core Hex motor and the one in motor port 3 is a Andymark Neverrest 40. This is the same for sensors and servos. Basically the config file is a map for the phone detailing all the hardware plugged into the REV Hub

You need to configure your phones to match the hardware setup of the robots.

Hardware Configuration - Creating a Config File

- Connect Robot Controller to REV Expansion Hub

The First thing you need to do is plug the Robot Controller into the REV Hub.

Hardware Configuration - Creating a Config File

- Connect Robot Controller to REV Expansion Hub
- Allow REV Hub and Phone to Connect

The phone will ask you if it’s ok to talk to the REV Hub and you say yes.

Hardware Configuration - Creating a Config File

- Connect Robot Controller to REV Expansion Hub
- Allow REB Hub and Phone to Connect
- Open New Config File
  - Hit the Three Dots in the Top Right Corner

Now you need to crate a new config file. You do this by hitting the three dots in the top right corner.

Hardware Configuration - Creating a Config File

- Connect Robot Controller to REV Expansion Hub
- Allow REB Hub and Phone to Connect
- Open New Config File
  - Hit the Three Dots in the Top Right Corner
  - Select “Configure Robot”

this opens a dialogue and you’re going to hit “Configure Robot”

Hardware Configuration - Creating a Config File

- Connect Robot Controller to REV Expansion Hub
- Allow REB Hub and Phone to Connect
- Open New Config File
  - Hit the Three Dots in the Top Right Corner
  - Select “Configure Robot”
  - Hit “New”

Hit “New” to open a new config file

Hardware Configuration - Creating a Config File

- Connect Robot Controller to REV Expansion Hub
- Allow REB Hub and Phone to Connect
- Open New Config File
  - Hit the Three Dots in the Top Right Corner
  - Select “Configure Robot”
  - Hit “New”
  - Hit “Scan”

Then hit “Scan.” This will tell the phone to look for REV Hubs

Hardware Configuration - Creating a Config File

- Connect Robot Controller to REV Expansion Hub
- Allow REB Hub and Phone to Connect
- Open New Config File
  - Hit the Three Dots in the Top Right Corner
  - Select “Configure Robot”
  - Hit “New”
  - Hit “Scan”
  - Select an Expansion Hub Portal

Select a REV Portal (there should only be one since there’s only one Hub plugged in.

Hardware Configuration - Creating a Config File

- Connect Robot Controller to REV Expansion Hub
- Allow REB Hub and Phone to Connect
- Open New Config File
  - Hit the Three Dots in the Top Right Corner
  - Select “Configure Robot”
  - Hit “New”
  - Hit “Scan”
  - Select an Expansion Hub Portal
  - Select a Hub

Then select a Hub. Right now there’s only one hub in the portal (the one that’s plugged into the phone), but later on you will probably need to add on a second Hub. Then both hubs will show up in the same portal and you will need to choose which hub you’re going to configure hardware on.

Hardware Configuration - Configuring the Motors

- In Config File, Select “Motors”

To Configure a motor, select “motors.”

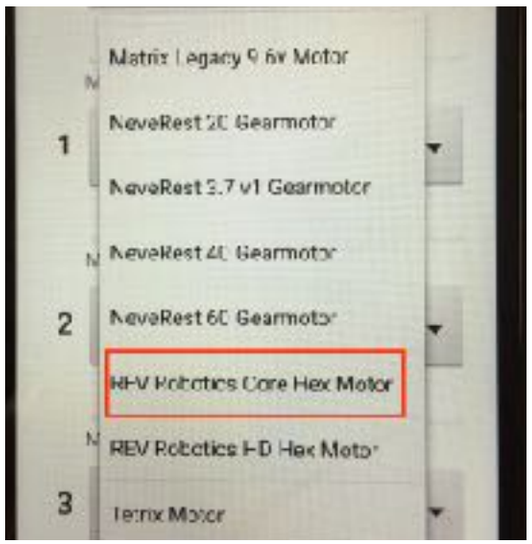
Hardware Configuration - Configuring Motors

- In Config File, Select “Motors”
- Motors are Plugged into Ports 0 and 2

The motors on the robots are plugged into morts 0 and 2. Select port 0 ***NEXT*** and then select the Core Hex Motor.

Hardware Configuration - Configuring a Motor

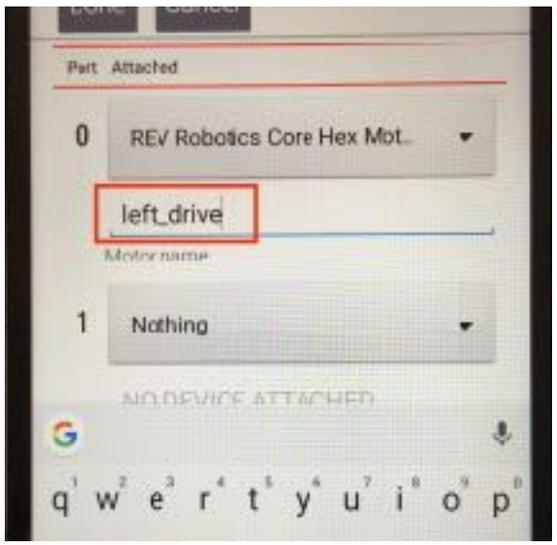
- In Config File, Select “Motors”
- Motors are Plugged into Ports 0 and 2
- Select REV Robotics Core Hex Motor



and then select the Core Hex Motor.

Hardware Configuration - Configuring a Motor

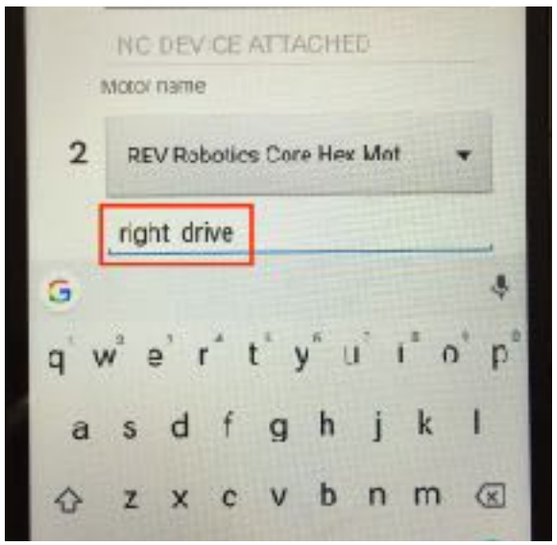
- In Config File, Select “Motors”
- Motors are Plugged into Ports 0 and 2
- Select REV Robotics Core Hex Motor
- Name Motors
  - Port 0: left_drive
  - Port 2: right_drive



Name the port 0 motor “left_drive”

Hardware Configuration - Configuring a Motor

- In Config File, Select “Motors”
- Motors are Plugged into Ports 0 and 2
- Select REV Robotics Core Hex Motor
- Name Motors
  - Port 0: left_drive
  - Port 2: right_drive



Do the same for the port 2 motor and name it “right_motor”

Hardware Configuration - Configuring a Motor

- In Config File, Select “Motors”
- Motors are Plugged into Ports 0 and 2
- Select REV Robotics Core Hex Motor
- Name Motors
  - Port 0: left_drive
  - Port 2: right_drive
- Hit “Done”

Then hit done.

So now you have the two motors in the config file and you’re ready to finish the config file. I will very very briefly go over configuring sensors and servos. They are outside of our scope today, but it’s super easy and you can go back and look at this presentation for reference.

Hardware Configuration - Configuring a Servo

- In Config File, Select “Servos”

For a Servo, it’s essentially the same process. Select servo.

Hardware Configuration - Configuring a Servo

- In Config File, Select “Servos”
- Select Port Number and Servo Type

Select a port number, and servo type.

Hardware Configuration - Configuring a Servo

- In Config File, Select “Servos”
- Select Port Number and Servo Type
- Name Servo (no Whitespace)

Name the servo.

Hardware Configuration - Configuring a Servo

- In Config File, Select “Servos”
- Select Port Number and Servo Type
- Name Servo (no Whitespace)
- Hit “Done”

And then hit done.

Hardware Configuration - Configuring an I2C Sensor

- In Config Menu, Select I2C Bus

Stuff starting to look familiar? To configure an I2C sensor, select an I2C bus.

Hardware Configuration - Configuring an I2C Sensor

- In Config Menu, Select I2C Bus
  - Bus 0 will always be the REV Hub's own internal IMU

So a public service announcement, the REV Hub has an internal IMU (inertial-measurement-sensor — it can determine the robot's orientation and acceleration). This IMU always defaults to I2C bus 0. You can override that, but unless you are out of buses or you really really want port 0, just use a different port.

Hardware Configuration - Configuring an I2C Sensor

- In Config Menu, Select I2C Bus
  - Bus 0 will always be the REV Hub's own internal IMU
  - Hit "Add" to add IMU to Config File

Add the IMU to the config file so that the phone can use it.

Hardware Configuration - Configuring an I2C Sensor

- In Config Menu, Select I2C Bus
  - Bus 0 will always be the REV Hub's own internal IMU
  - Hit "Add" to add IMU to Config File
  - Pick Either Bus 1,2, or 3

Pick another I2C bus

Hardware Configuration - Configuring an I2C Sensor

- In Config Menu, Select I2C Bus
  - Bus 0 will always be the REV Hub's own internal IMU
  - Hit "Add" to add IMU to Config File
  - Pick Either Bus 1,2, or 3
- Name I2C Sensor (no Whitespace)

Select your sensor type and name it.

Hardware Configuration - Configuring an I2C Sensor

- In Config Menu, Select I2C Bus
  - Bus 0 will always be the REV Hub's own internal IMU
  - Hit "Add" to add IMU to Config File
  - Pick Either Bus 1,2, or 3
- Name I2C Sensor
- Hit "Done"

then his "done"

Hardware Configuration - Configuring a Digital Sensor

- In Config Menu, Select “Digital Devices”

Oh look, it’s the exact same for a digital sensor.

Just FYI, the difference between a digital and analog sensor is that the digital sensor can send two states: on and off. It’s basically a light switch. The Analog sensor sends a range. Both of these sensors send back voltage. For the Digital it’s either 3.3 volts or it’s nothing. The analog, however, can send anywhere between 0 and 3.3 volts.

Hardware Configuration - Configuring a Digital Sensor

- In Config Menu, Select “Digital Devices”
- Select Port Number and Name Sensor (no Whitespace)

Select sensor type and name it.

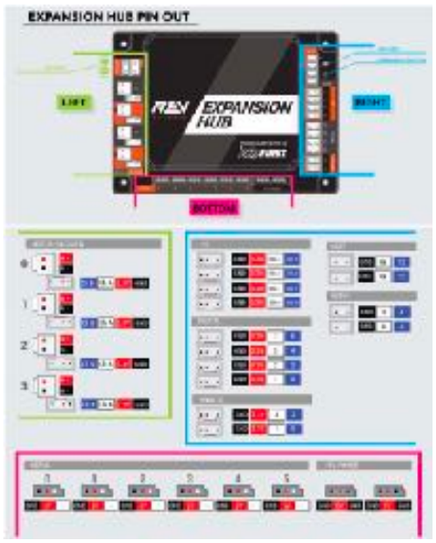
Hardware Configuration - Configuring a Digital Sensor

- In Config Menu, Select “Digital Devices”
- Select Port Number and Name Sensor (no Whitespace)
- Hit “Done”

Hit “Done”

Hardware Configuration - Configuring an Analog Sensor

- In Config Menu, Select “Analog Input Devices”
- Select Port Number and Name Analog Sensor (no Whitespace)
- Hit “Done”



Configuring an Analog sensor is going to be exactly the same as a digital sensor, only you’re going to choose Analog instead of Digital.

Hardware Configuration - Saving Config Info

- After Hardware is Configured, Hit “Done”

Once you have your hardware mapped in the config file, go ahead and save the config file. Hit “done”

Hardware Configuration - Saving Config Info

- After Hardware is Configured, Hit “Done”
- Press “Done” Again

then hit it again…

Hardware Configuration - Saving Config Info

- After Hardware is Configured, Hit “Done”
- Press “Done” Again
- Press “Save”

then hit save….

Hardware Configuration - Saving Config Info

- After Hardware is Configured, Hit “Done”
- Press “Done” Again
- Press “Save”
- Name Config File and Hit “OK”

Name the config file

Hardware Configuration - Saving Config Info

- After Hardware is Configured, Hit “Done”
- Press “Done” Again
- Press “Save”
- Name Config File and Save

and press “ok”

Hardware Configuration - Saving Config Info

- After Hardware is Configured, Hit “Done”
- Press “Done” Again
- Press “Save”
- Name Config File and Save
- Verify that Config File is Active
  - Hit “Back Button”

Now you need to verify that the phone is using your config file as the active one. Because you only have one right now, it should, but once you begin to have multiple config files, this will become much more important. To verify that the correct config file is active, hit the back button.

Hardware Configuration - Saving Config Info

- After Hardware is Configured, Hit “Done”
- Press “Done” Again
- Press “Save”
- Name Config File and Save
- Verify that Config File is Active
  - Hit “Back Button”
  - Verify That Your Config File is the Active Config File

Then look as the red bar near the top. It says “Active Configuration” on the left and the name of the active config file on the right. If it is displaying your config file, then you are good to go.

Does anyone have any questions?

Programming

Programming. Here I won’t be teaching you *how* to program, I’ll be covering *what* you can *use* to program.

Programming Tool/Languages - Overview

- Three Legal Programming Tools/ Languages:

So, FIRST allows three programming tools. These tools use one of two programming languages.

Programming Tool/Languages - Overview

- Three Legal Programming Tools/ Languages:
  - Blocky (FTC Blocks)

The first tool is FTC Blocks (or Blocky). Blocky is its own language.

Programming Tool/Languages - Overview

- Three Legal Programming Tools/ Languages:
  - Blocky (FTC Blocks)
  - FTC OnBot (Java Programing Tool)

The next legal choice is OnBot. Which uses the Java programming language.

Programming Tool/Languages - Overview

- Three Legal Programming Tools/ Languages:
  - Blocky (FTC Blocks)
  - FTC OnBot (Java Programing Tool)
  - Android Studio (Java Android App Development Tool)

And then there’s Android Studio which also uses Java.

Programming Tools/Languages - Blocky

- Visual Programming Tool
- Drag and Drop Interface (Similar in Concept to LEGO Mindstorms EV3 G)
- Browser-Based
- Intuitive and Easy to Learn
- FIRST Recommends for Rookie Programmers

So Blocky. Blocky is a graphical programming tool — meaning you drag and drop objects onto a pogromming canvas. Pretty much the exact same as the LEGO Mindstorms stuff, only for FTC.

FIRST Recommends Blocky for Rookie Programmers due to its drag and drop visual nature, especially when compared to the complexity of Java. We have no experience with Blocky, so everything we say about it is straight from the FIRST website, but we do know that FIRST has a lot of resources for Blocky. There is a youtube playlist of Tutorials for Blocky on the FTC youtube channel and there are also written tutorials on the FTC gitHub page. There are links to all of that on our website and in your packet.

Programming Tools/Languages - OnBot

- Text-Based
- Java OpModes
- Browser Based
- FIRST Recommends for Programmers with Basic to Advanced Java and Text-Based Programming Skills

We have no experience with OnBot. It’s a browser based tool that uses Java. FIRST Recommends for Programmers with Basic to Advanced Java and Text-Based Programming Skills.

Programming Tools/Languages - Android Studio

- Text-Based
- Java OpModes
- Advanced Integrated Development Environment for Creating Android Apps
- Professional App Development Tool
- FIRST Recommends for Advanced Users with Extensive Java Programming Experience

And then Android Studio. Error 404 has only used Android Studio. It’s like the pro-tool of the bunch. Super complicated and hard to get working right, but once it’s up and running, it’s great.

Programming Tools/Languages - Which One?

- Blocky is a Less Capable, Much Simpler Tool, Intuitive

Error 404 has only ever used Android Studio and as such we have very limited experience with Blocky and OnBot (hence why these classes are angled towards teams using Android Studio).

Blocky is designed to be a simpler yet functional language designed by FIRST for FTC. It utilizes the visually intuitive drag and drop environment and is the easiest of the three choices for teams to pick up. Blocky is also browser-based meaning that as long as you have a javascript-enabled web browser, you can access your code via computer (Apple, Windows, Chromebook) or tablet (Apple or Android). We heard from teams last season that they found Blocky limited robot performance and that there were some software limitations

NEXT

OnBot is a middle-of-the-road type tool. It uses Java and requires that students have at least a basic knowledge of Java. OnBot, like Blocky, is browser-based and therefore accessible from computers and tablets alike. This tool assumes that the student has a sound understanding of Java and FIRST recommends that students unfamiliar with Java use Blocky.

And Finally Android Studio. Android Studio is a professional app development tool for creating Android apps and requires a solid working knowledge of Java. Android Studio is not browser-based and instead is an application on your computer. Again, for students unfamiliar with Java and script-base programming, FIRST recommends that they use Blocky. I stress that FIRST recommends because Error 404 has only ever used Android Studio and therefore we don't have any formal experience with Blocky.

So when picking a programming tool/language, each team is going to be different. Teams will need to factor in their collective experience in programming — and text-based programming specifically —, the average age of the member students, and how much time a team is prepared to devote to learning a language and tool environment. Take it from a guy who jumped in the deep end, learning Java from scratch is quite the time commitment. Especially for students unused to text-based

Programming Tools/Languages - Which One?

- Blocky is a Less Capable, Much Simpler Tool, Intuitive
- OnBot Requires a Basic Knowledge of Java and Text-based Programing

OnBot is a middle-of-the-road type tool. It uses Java and requires that students have at least a basic knowledge of Java. OnBot, like Blocky, is browser-based and therefore accessible from computers and tablets alike. This tool assumes that the student has a sound understanding of Java and FIRST recommends that students unfamiliar with Java use Blocky.

NEXT

And Finally Android Studio. Android Studio is a professional app development tool for creating Android apps and requires a solid working knowledge of Java. Android Studio is not browser-based and instead is an application on your computer. Again, for students unfamiliar with Java and script-base programming, FIRST recommends that they use Blocky. I stress that FIRST recommends because Error 404 has only ever used Android Studio and therefore we don't have any formal experience with Blocky.

So when picking a programming tool/language, each team is going to be different. Teams will need to factor in their collective experience in programming — and text-based programming specifically —, the average age of the member students, and how much time a team is prepared to devote to learning a language and tool environment. Take it from a guy who jumped in the deep end, learning Java from scratch is quite the time commitment. Especially for students unused to text-based programming languages. Once students have mastered Java and the text-based programming environment however, Android Studio is a professional-level tool that is superbly functional.

Programming Tools/Languages - Which One?

- Blocky is a Less Capable, Much Simpler Tool, Intuitive
- OnBot Requires a Basic Knowledge of Java and Text-based Programing
- Java in Android Studio Requires an Advanced Working Knowledge of the Java Programming Language, Powerful, Flexible

And Finally Android Studio. Android Studio is a professional app development tool for creating Android apps and requires a solid working knowledge of Java. Android Studio is not browser-based and instead is an application on your computer. Again, for students unfamiliar with Java and script-base programming, FIRST recommends that they use Blocky. ***NEXT*** I stress that FIRST recommends because Error 404 has only ever used Android Studio and therefore we don't have any formal experience with Blocky.

NEXT

So when picking a programming tool/language, each team is going to be different. Teams will need to factor in their collective experience in programming — and text-based programming specifically —, the average age of the member students, and how much time a team is prepared to devote to learning a language and tool environment. Take it from a guy who jumped in the deep end, learning Java from scratch is quite the time commitment. Especially for students unused to text-based programming languages. Once students have mastered Java and the text-based programming environment however, Android Studio is a professional-level tool that is superbly functional.

Programming Tools/Languages - Which One?

- Blocky is a Less Capable, Much Simpler Tool, Intuitive
- OnBot Requires a Basic Knowledge of Java and Text-based Programing
- Java in Android Studio Requires an Advanced Working Knowledge of the Java Programming Language, Powerful, Flexible
- 404 Uses Android Studio

I stress that FIRST recommends because Error 404 has only ever used Android Studio and therefore we don't have any formal experience with Blocky.

NEXT

So when picking a programming tool/language, each team is going to be different. Teams will need to factor in their collective experience in programming — and text-based programming specifically —, the average age of the member students, and how much time a team is prepared to devote to learning a language and tool environment. Take it from a guy who jumped in the deep end, learning Java from scratch is quite the time commitment. Especially for students unused to text-based programming languages. Once students have mastered Java and the text-based programming environment however, Android Studio is a professional-level tool that is superbly functional.

Programming Tools/Languages - Which One?

- Blocky is a Less Capable, Much Simpler Tool, Intuitive
- OnBot Requires a Basic Knowledge of Java and Text-based Programing
- Java in Android Studio Requires an Advanced Working Knowledge of the Java Programming Language, Powerful, Flexible
- 404 Uses Android Studio
- Pick the One That Seems Best for Your Team's Experience/Age/Available Time

When picking a programming tool/language, each team is going to be different. Teams will need to factor in their collective experience in programming — and text-based programming specifically —, the average age of the member students, and how much time a team is prepared to devote to learning a language and tool environment. Take it from a guy who jumped in the deep end, learning Java from scratch is quite the time commitment. Especially for students unused to text-based programming languages. Once students have mastered Java and the text-based programming environment however, Android Studio is a professional-level tool that is superbly functional.

Are there any questions before we move on?

Creating and Running an OpMode (Android Studio)

Now we’re going to transition to actually programming. You will need a fully functional robot, phone set, and computer with Android Studio and requisite SDKs.

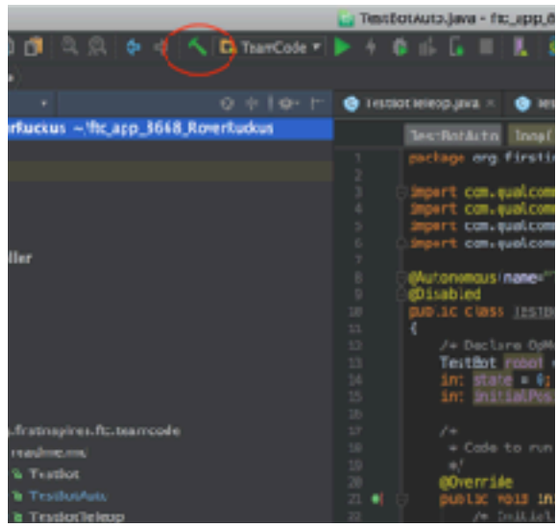
Android Studio - Disable Instant Run

- Disabling Instant Run Prevents Severe and Difficult-to-Troubleshoot Problems

If you are an Android Studio user, one of the most important steps to take is to disable Android Studio Instant Run. Instant Run is a feature that is designed to streamline the development process by reducing the time to apply code changes to your app. Unfortunately, Instant Run is limited in function and when used with the FIRST Tech Challenge Android Studio project folder, can cause severe and difficult-to-troubleshoot problems. Basically, If you’re using Android Studio and don’t want to age prematurely, disable the instant run feature. And here’s how to do that.

Android Studio - Disable Instant Run

- Disabling Instant Run Prevents Severe and Difficult-to-Troubleshoot Problems
- Build Gradle



The first thing is to hit the little green hammer in the top tool bar of Android Studio.

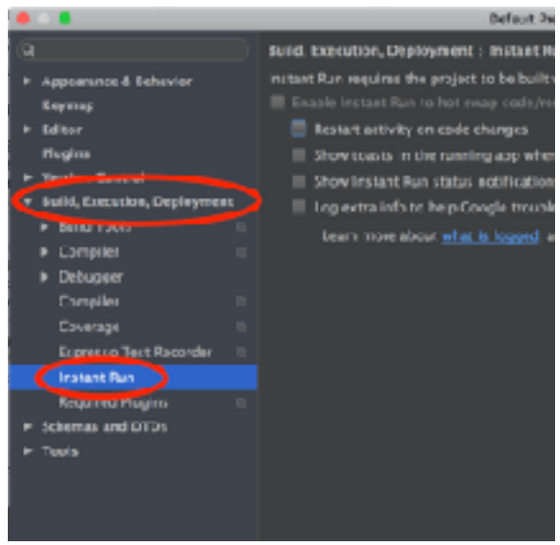
Android Studio - Disable Instant Run

- Disabling Instant Run Prevents Severe and Difficult-to-Troubleshoot Problems
- Build Gradle
- Locate Instant Run Settings
  - Open Android Studio Preferences

Open Android Studio Preferences (or settings depending on your operating system).

Android Studio - Disable Instant Run

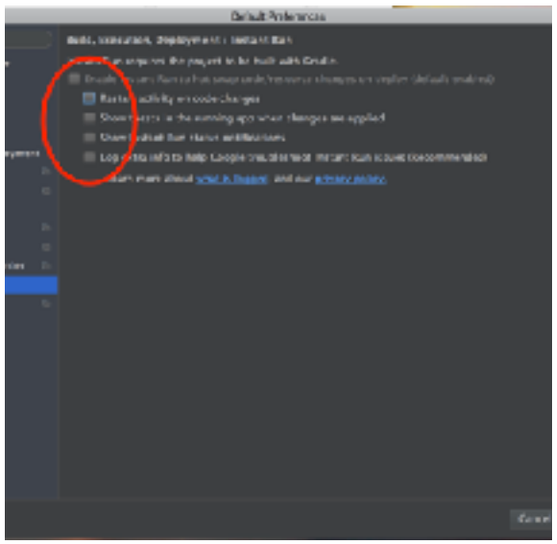
- Disabling Instant Run Prevents Severe and Difficult-to-Troubleshoot Problems
- Build Gradle
- Locate Instant Run Settings
  - Open Android Studio Preferences
  - Under “Build, Execution, Deployment,” Select “Instant Run”



under Build,Execution,Deployment, select instant run

Android Studio - Disable Instant Run

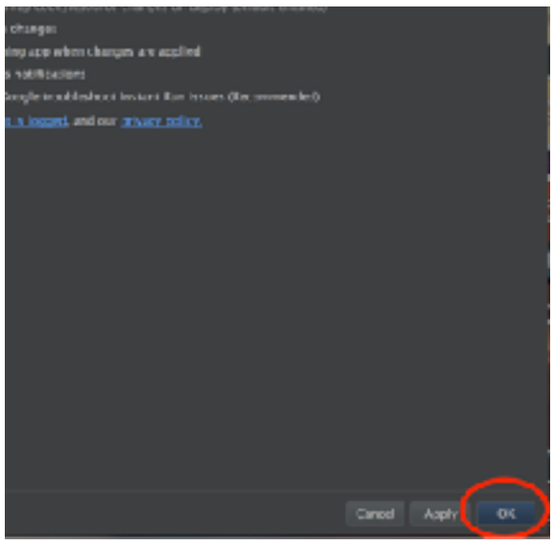
- Disabling Instant Run Prevents Severe and Difficult-to-Troubleshoot Problems
- Build Gradle
- Locate Instant Run Settings
  - Open Android Studio Preferences
  - Under “Build, Execution, Deployment,” Select “Instant Run”
- Uncheck Instant Run



uncheck everything. If the top checkbox is grayed out, then hit the little green hammer at the top again. Then open preferences and check the instant run settings again. If it’s not, call someone for help

Android Studio - Disable Instant Run

- Disabling Instant Run Prevents Severe and Difficult-to-Troubleshoot Problems
- Build Gradle
- Locate Instant Run Settings
  - Open Android Studio Preferences
  - Under “Build, Execution, Deployment,” Select “Instant Run”
- Uncheck Instant Run
- Click “OK”



Once you have everything unchecked, hit ok

What is an OpMode?

- An *Op* mode is a computer program that is used to customize the behavior of a competition robot. The Robot Controller can *execute* a selected op mode to perform certain tasks during a match.

So from here on out you will probably be hearing the term OpMode a lot. An OpMode is basically a program. Technically an opMode is a computer program that is used to customize the behavior of a competition robot. The Robot Controller can execute a selected op mode to perform certain tasks during a match. So an opMode is a program.

Types of OpModes

- Linear OpMode

There are two types of autonomous OpModes. Autonomous and Teleop. And we will talk about both of them. The types of programs will align with the match phases. In FTC the matches are split into two parts: Autonomous and Driver Controlled. The First thirty seconds of the match are called Autonomous and during autonomous the robot runs completely on pre-programmed instructions. the second part of a match is called the Driver Controlled Period. During this period, drivers will use gamepads to control the robot directly.

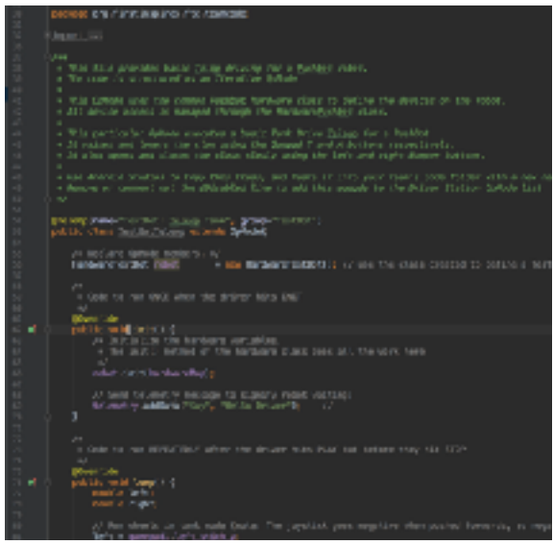
Types of OpModes

- Linear OpMode
  - Step by Step (1,2,3,4,5….,etc)
- Iterative OpMode
  - Traditional Java structure

There are two type of OpMode:
The Linear OpMode and the Iterative or just normal OpMode. The Linear OpMode is specifically structured so that the code executes from top to bottom.
Whereas the normal iterative OpMode can jump around and provide more flexibility and decision making.

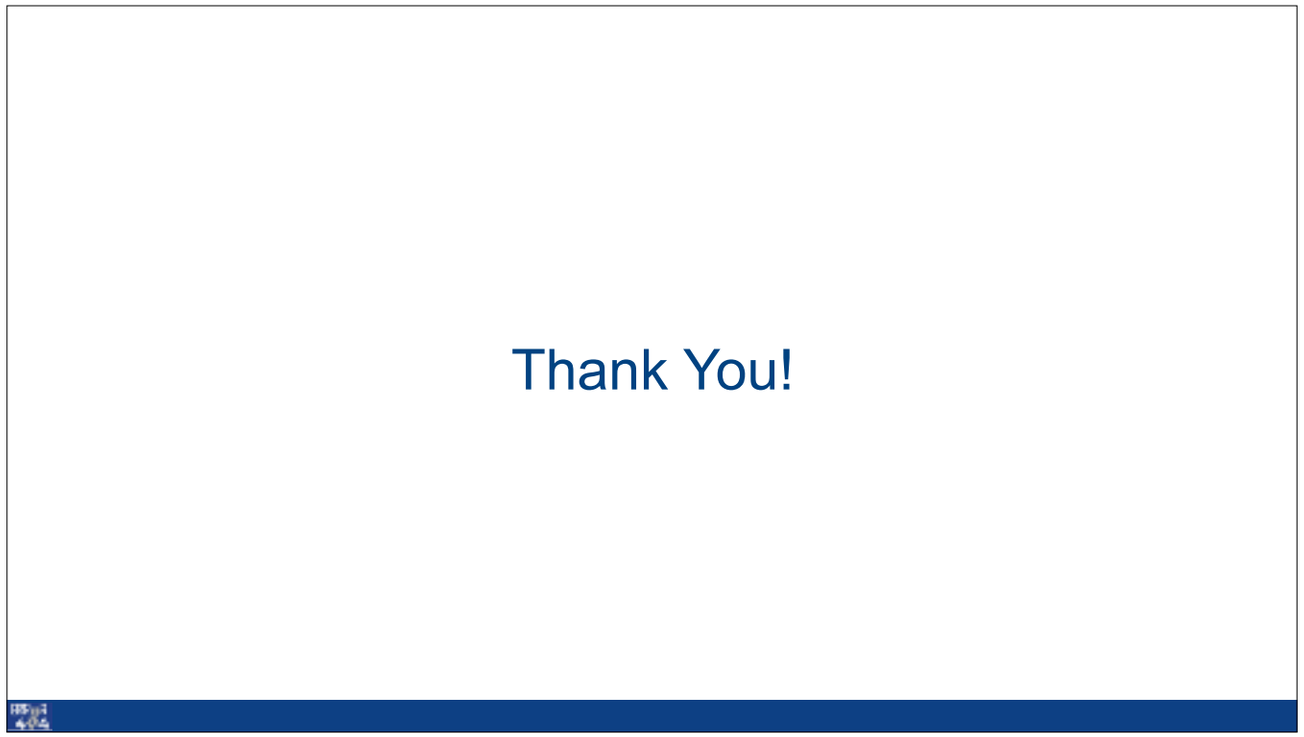
Anatomy of an OpMode

- Init Method - Called When Init Button is Pressed
- Start Method - Called When Start Button is Pressed
- Loop Method - The Main Body of the Functional Code
- Stop Method - Called When the Stop Button is Pressed — It Stops Everything



Now I am going to talk about code structure and some of the methods FIRST has given us. I will be talking about the same code you guys have so you can follow along on your own computers if you want, but I will have Android Studio up on the big screen.

- Move over to Android Studio.
- Show them each method and what it does



Does anyone have any questions?

Cool. This concludes Error 404's Programming in FTC workshop series. I'm Andrew and it has been my pleasure to present to you this week. Thank you for coming.