


# Setting up a Configuration

Before we can truly dive into programming our robot we need to help our robot's brain, the Control Hub, to know what is connected to it. Through the configuration process we can tell the Control Hub which port sensors, motors, servos, and any other connected devices can be found.

This is one of the most important steps to always complete BEFORE you can start programming!

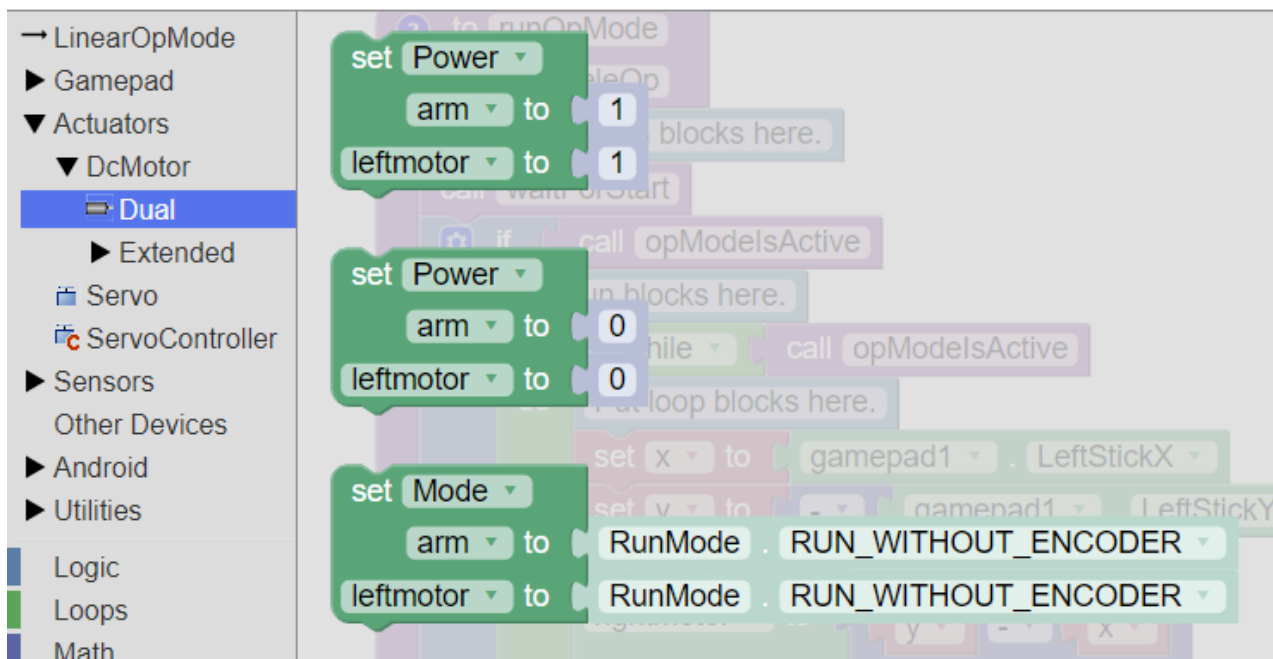
 When programming in Blocks, some blocks may be hidden UNTIL the configuration process is completed and activated on the Driver Hub.

## The Importance of Configuration

While every REV Control Hub is the same, the robots being controlled are each unique. Each Control Hub has the same number of input and output ports for motors, sensors and cameras, but, how you may utilize these ports varies from system to system. For instance, a Color Sensor V3 may be plugged in to I2C Bus 1 on one person's Hub, but another might use the same bus to host a 2m Distance Sensor.

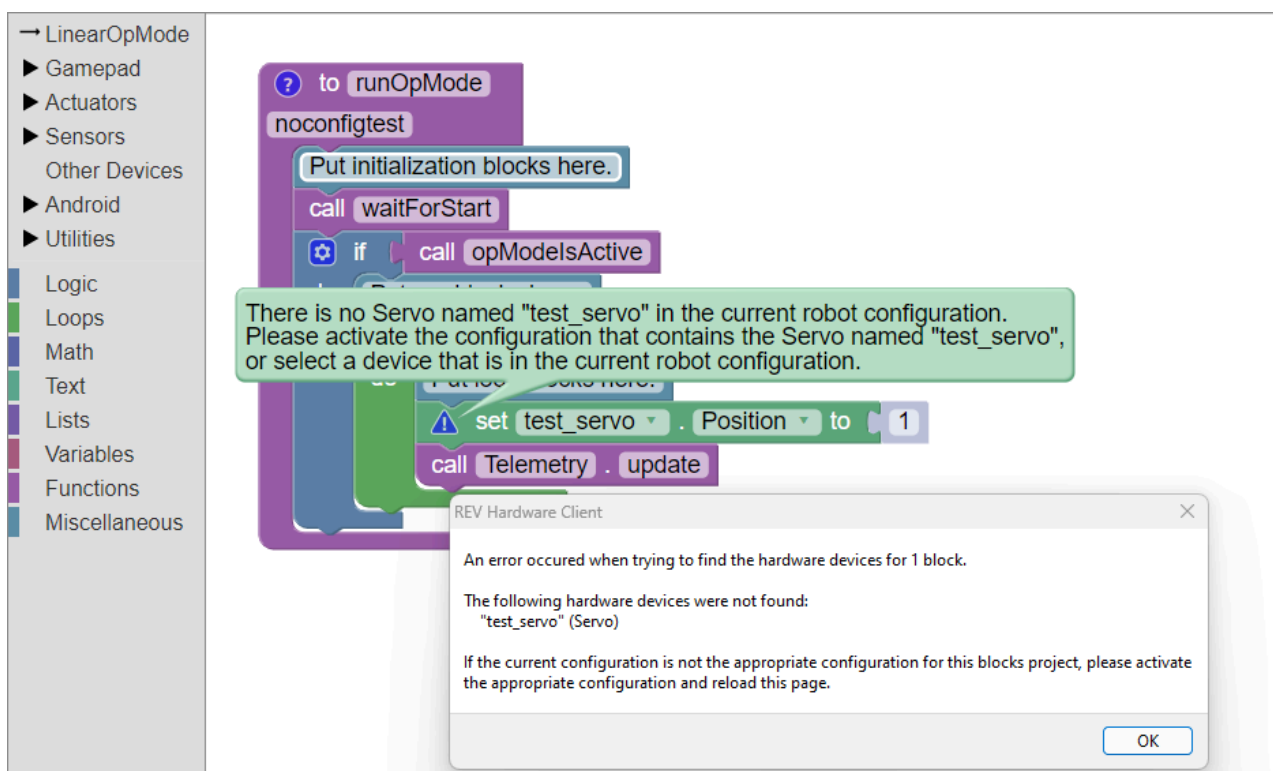
While the Control Hub may know there is a device attached to a port, it doesn't instinctively know which information needs to be transferred back for use in an [OpMode](#). To help our robot out we need to complete a process called hardware mapping. Generally, this is a two step process that includes creating our configuration file using the Driver Hub and calling the hardware map within our OpMode.

When using Blocks, this second step is handled by the SDK itself. Once you've created your configuration file on the Driver Hub your robot's devices are ready to use and you will see their names in the dropdown menus of related blocks.



In this configuration the motors have been named "arm" and "leftmotor"

- ❗ If you receive an error (example below) when opening a Blocks OpMode that the Control Hub could not locate a device, double check that the correct configuration file is currently active!



- ✓ Click to Learn More about the Configuration File

The configuration file is a readable file created by the user through the Driver Station Application. When creating a configuration file users are required to assign each device to a port, select the type of device it is from options provided by the SDK, and give it a **unique** name.

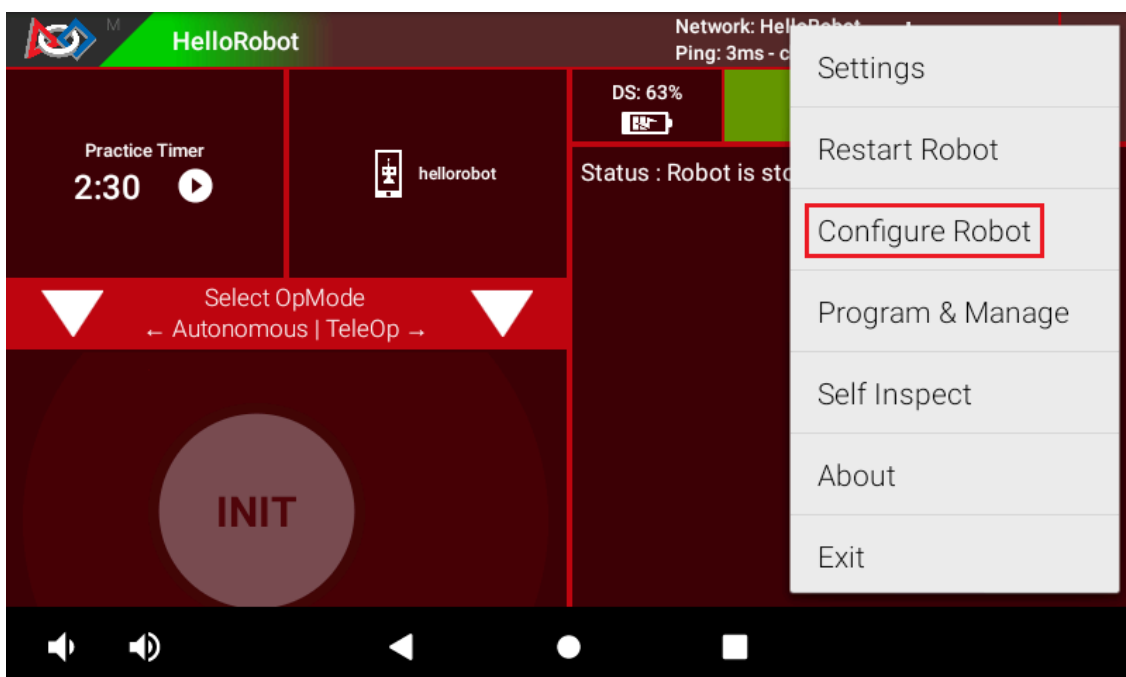
It's important to name each device something recognizable and distinguishable!

Once a configuration file is saved or activated the robot will restart. This restart is so the SDK can read the file, determine what devices are present, and add the devices to the hardwareMap class.

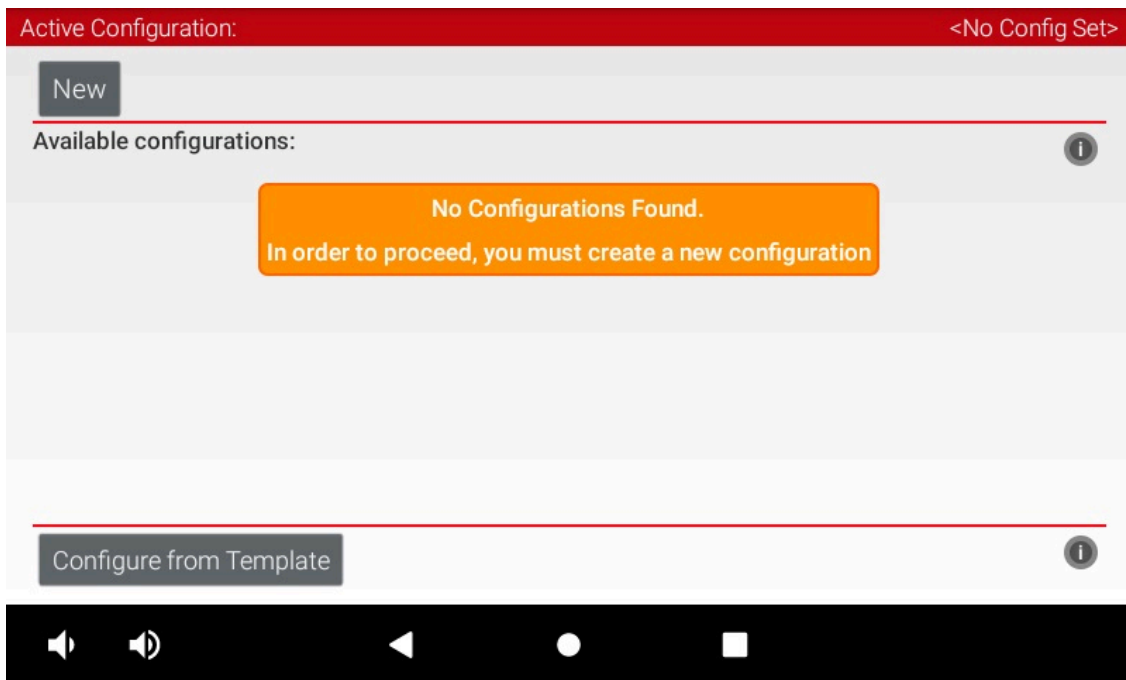
## Configuring Common Hardware Devices

### Accessing the Configuration Utility

1. Select the menu in the stop right corner of the Driver Station app. Then select Configure Robot.

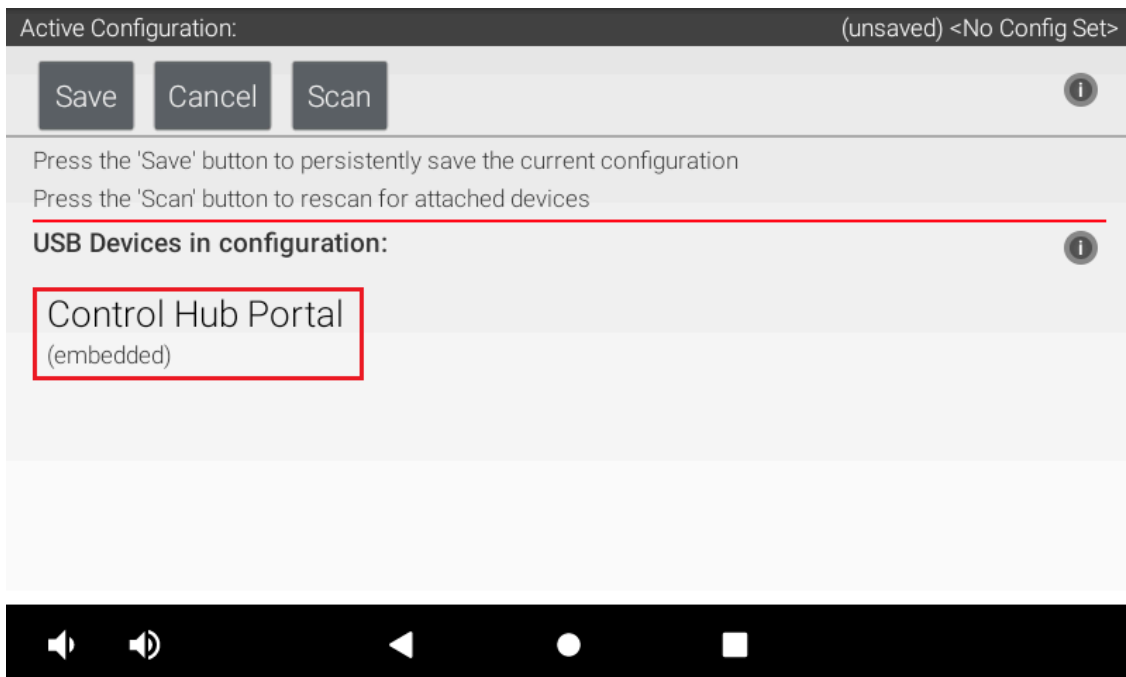


2. In the Available configurations page, select New.



3. In the USB Devices in configuration page select the Control Hub Portal.

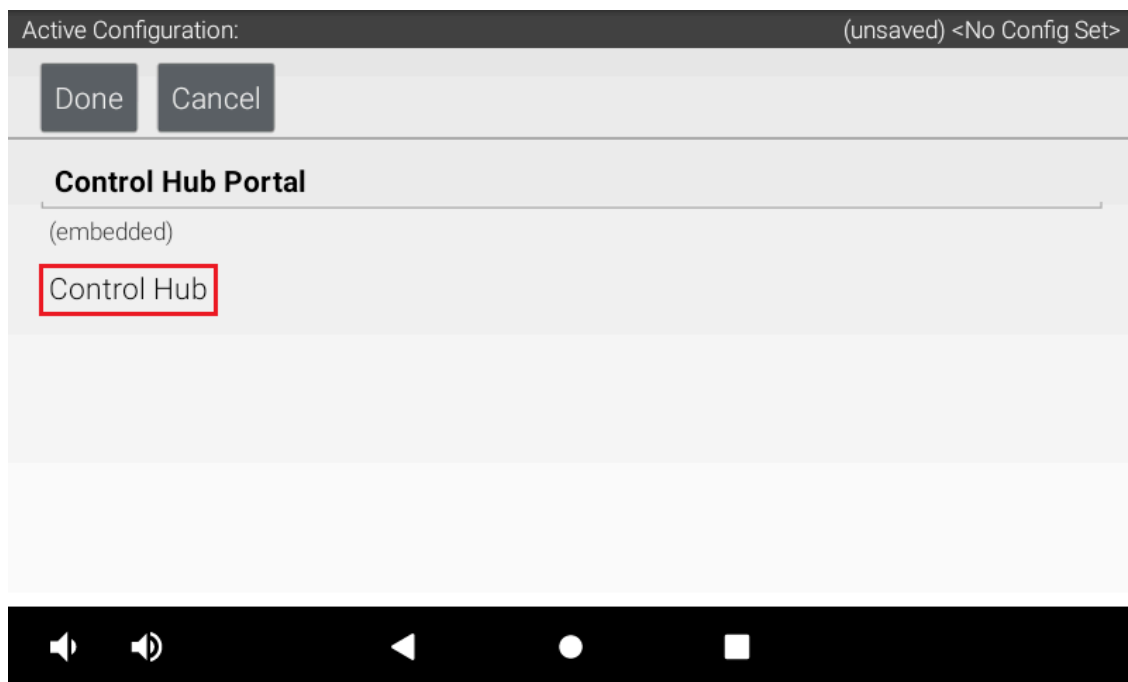
**Note:** If you have an Expansion Hub connected it will appear as an Expansion Hub Portal.



⚠ Pressing "**Scan**" on an existing configuration may result in the already named devices being erased. A new configuration file is needed when adding a camera or Expansion Hub.

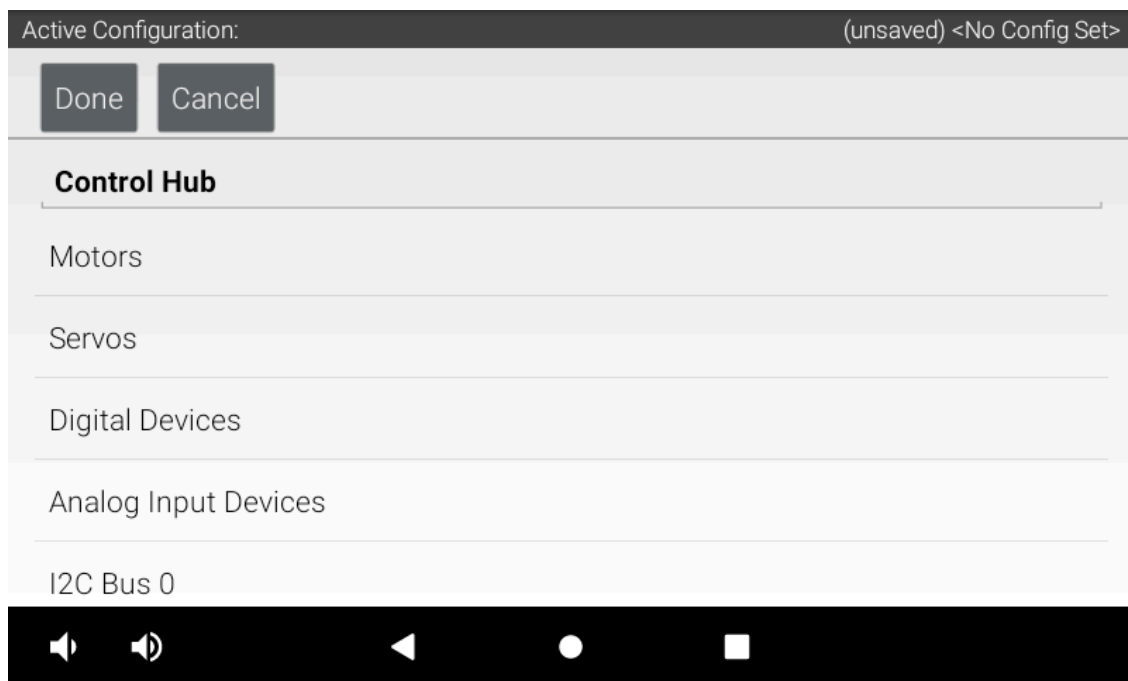
4. Within the Hub Portal select the device you want to configure. In this use case, select the Control Hub.

**Note:** if you have an Expansion Hub connected to a Control Hub, the Expansion Hub will also appear as a configurable device in the portal.



5. This will bring you to the page shown in the image. From here you can configure motors, servos and sensors that you are using. Follow through the rest of the guide to figure out how to configure devices that will be used in the Test Bed section.

**Note:** The way that Digital and Analog devices are configured versus how I2C devices are configured differ significantly. This is because each physical I2C port is a different bus that can host multiple different sensors. For more information on the different types of sensors check out the [sensors](#) section.

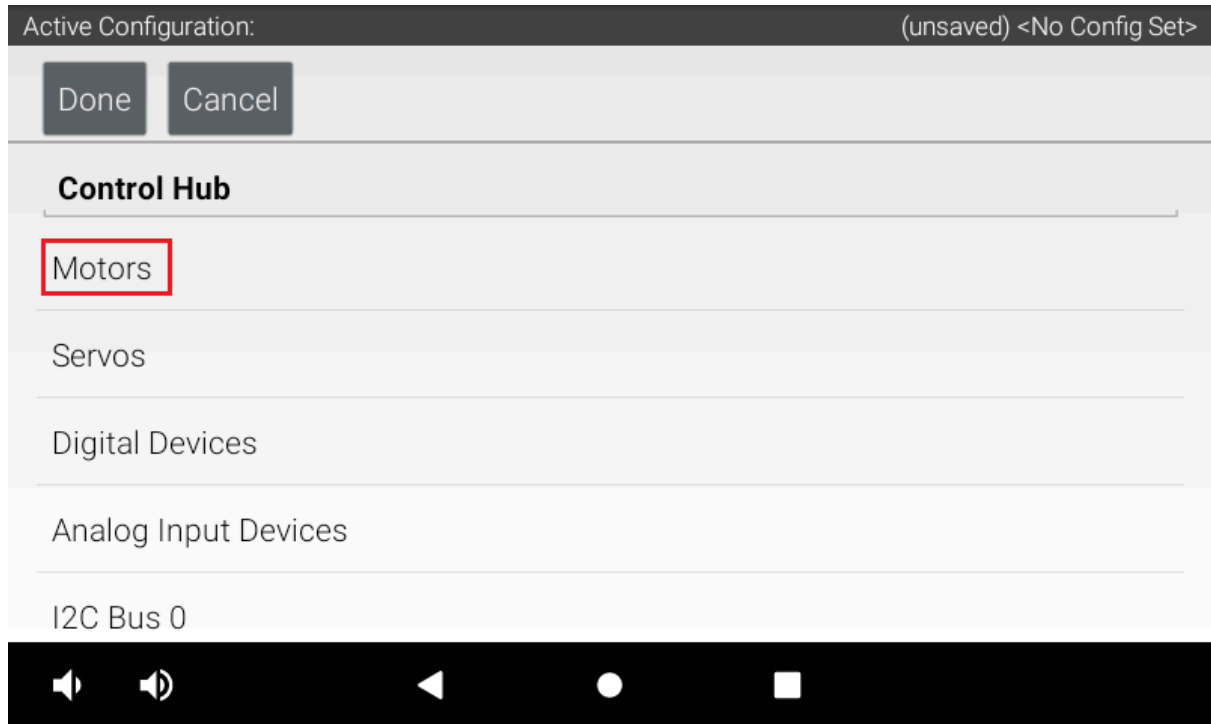


## Configuring Hardware

The following section will show how to configure components that will be used in the Test Bed. The hardware type and names have been chosen in consideration for the Hello World lesson plan. Users should heed notes within the steps to consider when creating configuration files for other instances.

## Configuring a Motor

1. Select Motors.



2. The Motor page will allow you to configure all four motor ports on the Hub. On Port 0 open the drop down menu and select REV Robotics Core Hex Motor.

**Note:** In your configuration file you should configure the motor ports to the type of motor you are using.

Active Configuration: (unsaved) <No Config Set>

Done Cancel

Port	Attached
0	Nothing
NO DEVICE ATTACHED	
Motor name	
1	Nothing
NO DEVICE ATTACHED	

Android navigation bar

Active Configuration: (unsaved) <No Config Set>

Done Cancel

Port	Attached
0	<div> REV Robotics 20:1 HD Hex Motor  REV Robotics 40:1 HD Hex Motor  REV Robotics Core Hex Motor  REV Robotics UltraPlanetary HD Hex Motor </div>
1	

Android navigation bar

3. Name the motor test\_motor. Select done.

**Note:** remember when naming hardware in the configuration file that the REV Control System is Case Sensitive.



Active Configuration:

(unsaved) <No Config Set>

Done

Cancel

Port Attached

0

REV Robotics Core Hex Motor

test\_motor

Motor name

1

Nothing

NO DEVICE ATTACHED



## Configuring a Servo

1. Select Servos.

2. The Servo page will allow you to configure all six servo ports on the Hub. On Port 0 open the drop down menu and select **Servo**.

**Note:** REV Servos can be configured as a Servo or a Continuous Rotation Servo. The type of device a servo is configured as should correspond with the mode the sensor is in. For more information on Sensor modes visit the [Sensor section ↗](#).

3. Name the servo test\_servo. Select done.

**Note:** remember when naming hardware in the configuration file that the REV Control System is Case Sensitive.



# Configuring a Digital Device

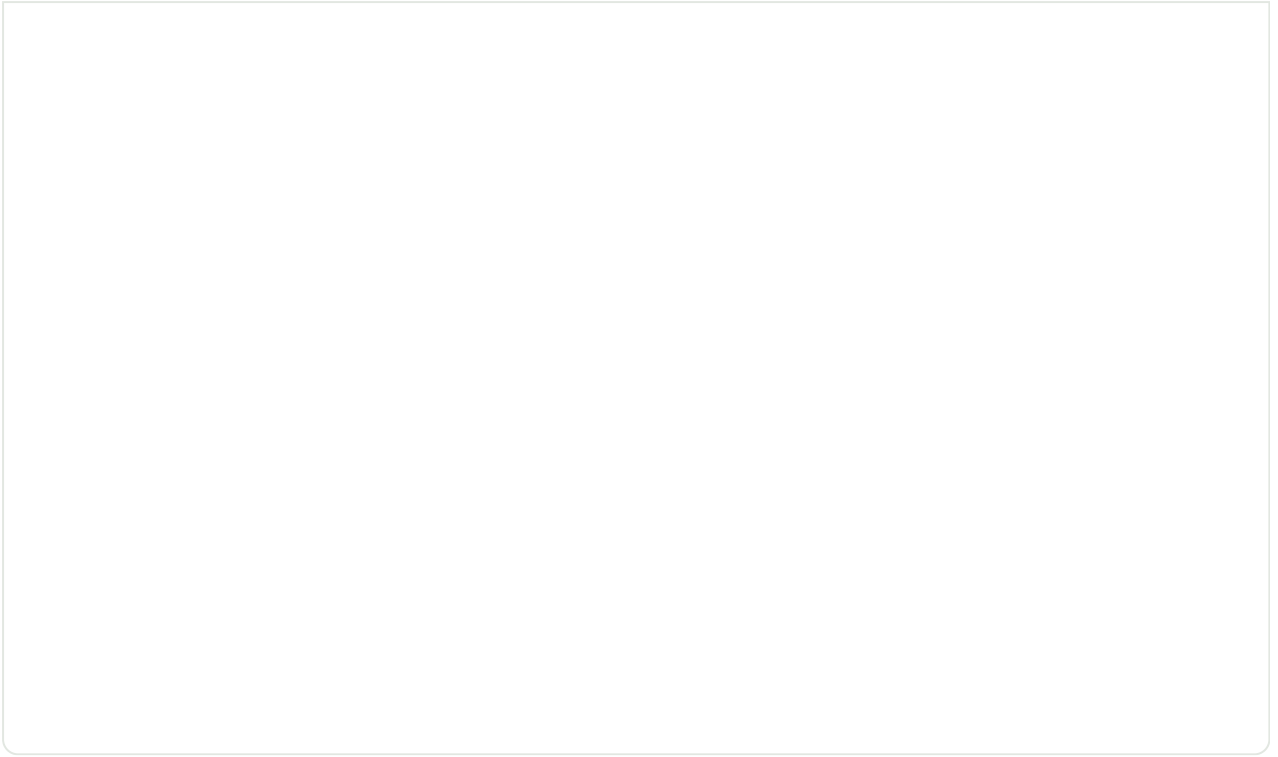
- Select Digital Devices.

**Note:** Touch Sensors must always be configured on odd number ports.

**Note:** Touch Sensors can be configured as a REV Touch Sensor or a Digital Device. In the FTC SDK the type of device it is configured as changes the classes and methods that can be used.

- Name the motor test\_touch. Select done.

**Note:** remember when naming hardware in the configuration file that the REV Control System is Case Sensitive.



## Configuring an I2C Device

- Select I2C Bus 0.

- Select Add.

**Note:** Each I2C Bus can host more than one I2C sensor as long as the I2C addresses do not conflict. Bus 0 will always host the internal IMU. For more information on I2C sensors visit the [I2C section](#).



- On Port 1, which was created in the previous step, open the drop down menu and select REV Color Sensor V3.

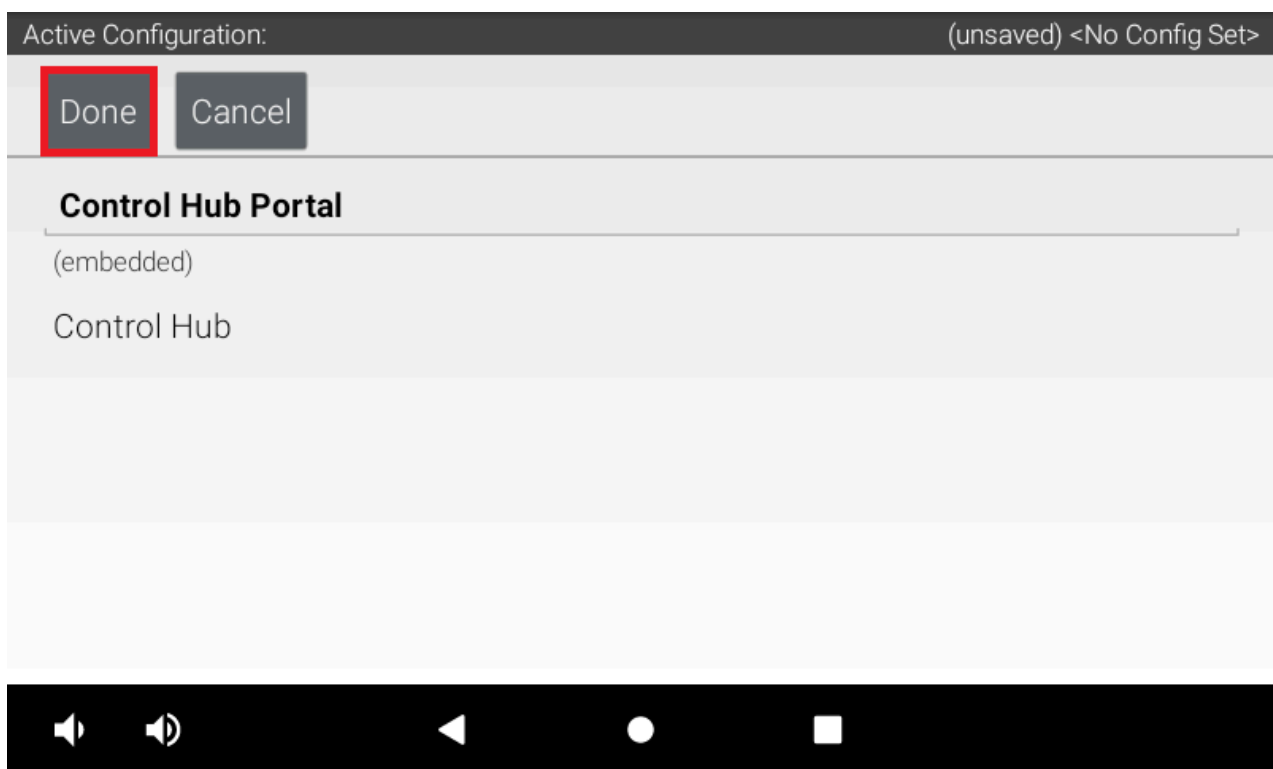
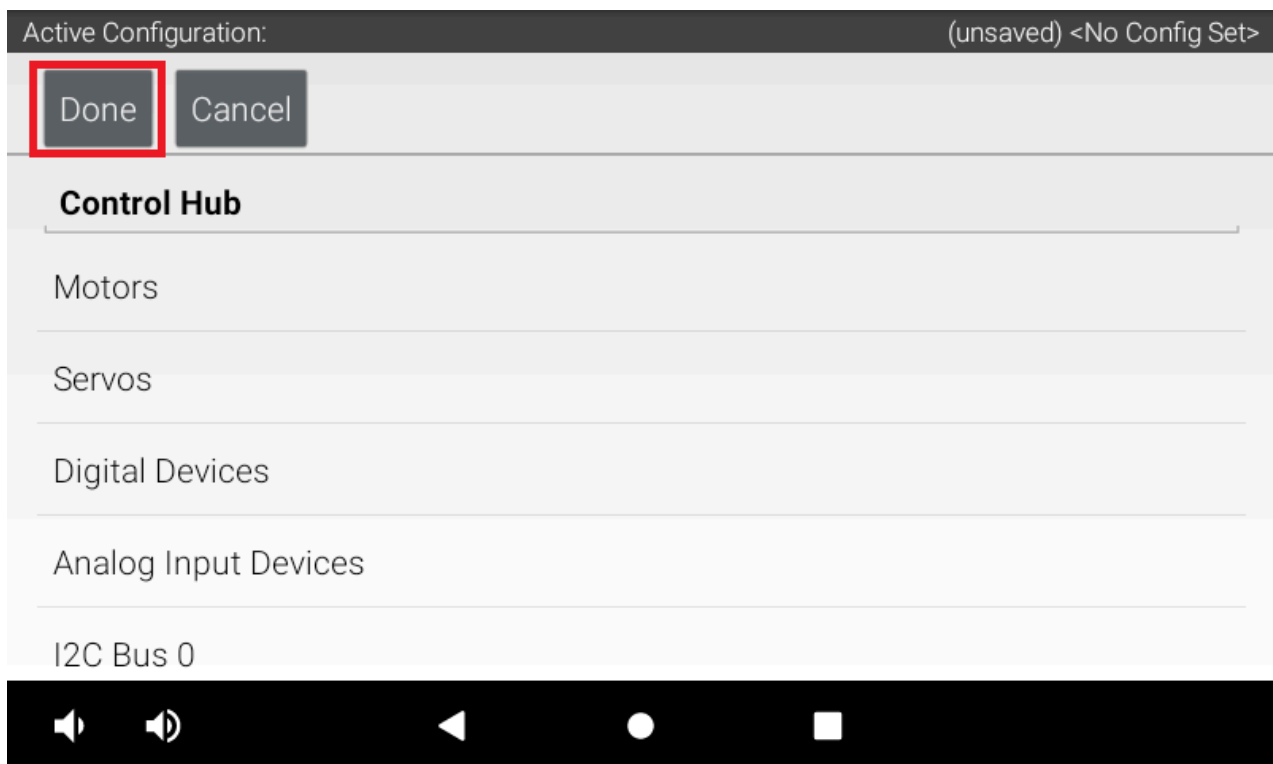
**Note:** If you are using Color Sensors V1 or V2 select REV Color/Range Sensor. For more information on configuring with the REV Color Sensors visit the [Color Sensor Datasheets ↗](#).

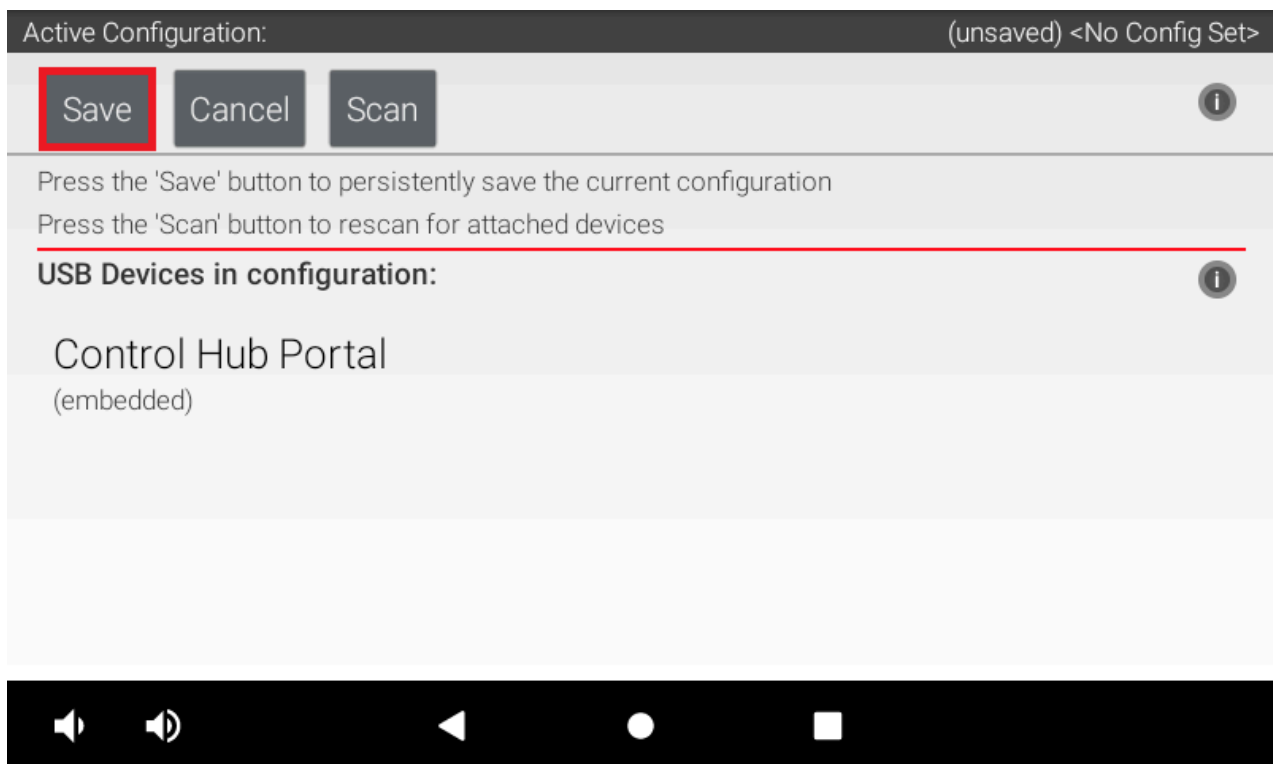
- Name the motor test\_color. Select done.

**Note:** remember when naming hardware in the configuration file, that the REV Control System is Case Sensitive.

## Saving the Configuration File

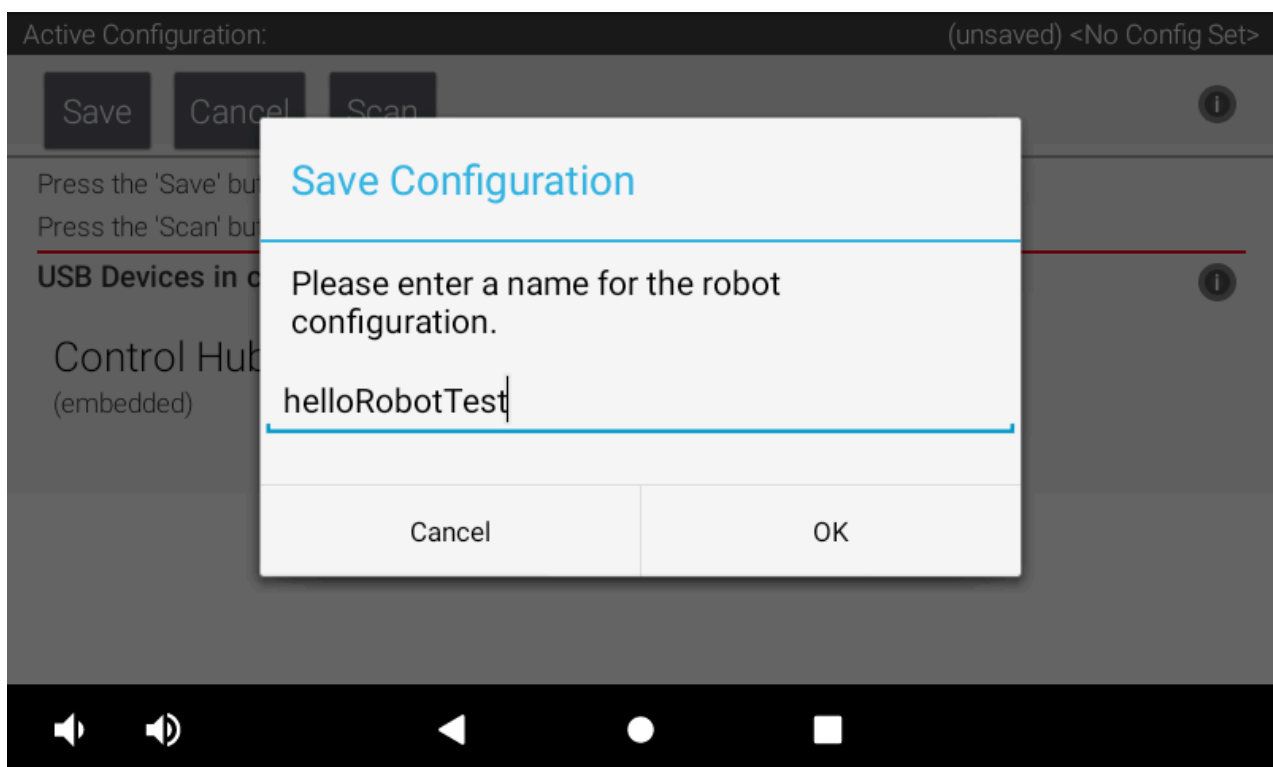
1. Hit Done twice until you reach the USB Devices in configuration page. On the USB Devices in configuration page hit Save.



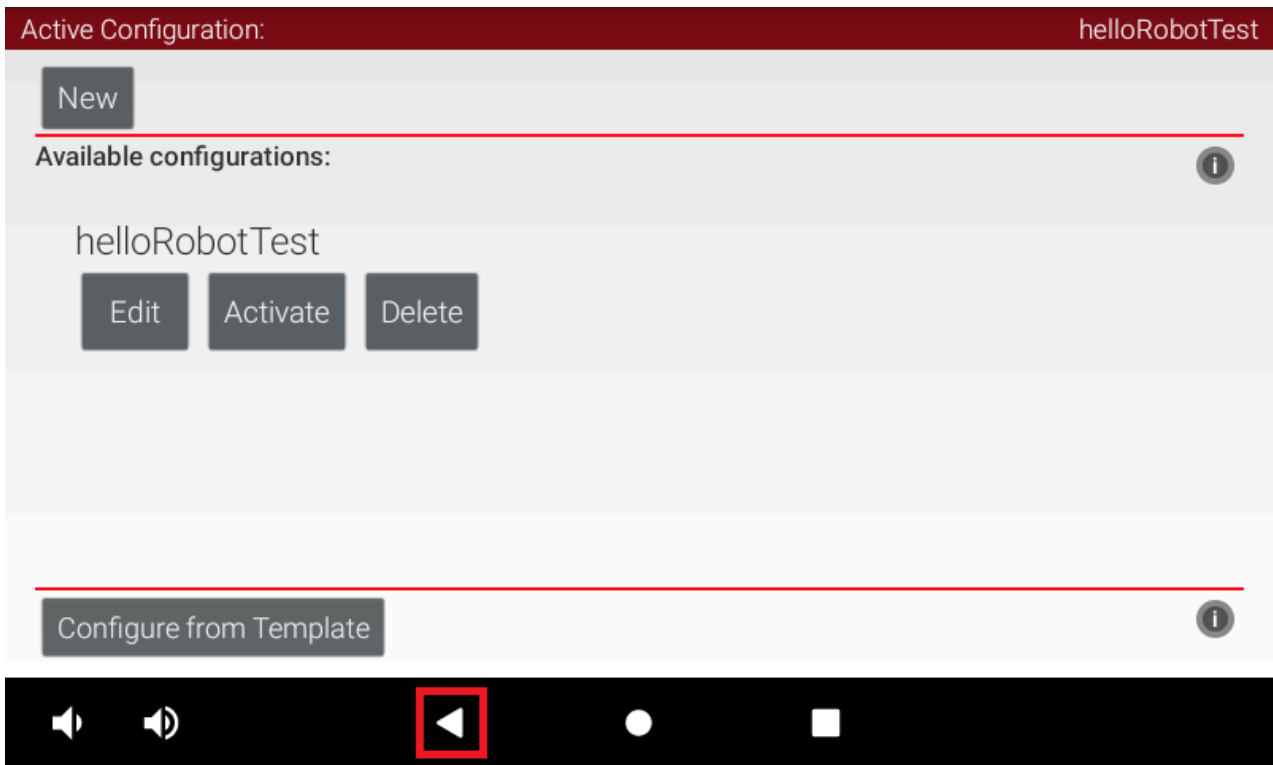


2. Name the configuration **helloRobotTest** and then select Ok.

**Note:** The FTC SDK does not force you to abide by a naming convention for but it is common to name configurations in lowerCamelCase.



3. Press back to activate the saved configuration. Your Robot Controller will restart once you activate a new configuration.



# Common Errors in Configuration

While there are many errors one may run into in the programming and software world we're going to focus for now on the two major errors that may occur when hardware mapping.

- **Interface Errors** - errors between how an interface should work and how it actually behaves
- **Runtime Errors** - errors that occur when a program is being executed


## Interface Errors

**Interface errors** occur in the SDK when the parameters of the SDK interface are not met. In the hardware mapping process, the most common interface error occurs within the Blocks Programming Tool.

Blocks is designed to simplify the user experience by automatically handling the conversation of the hardwareMap to pre-named blocks. This means while no configuration is active the applicable block options are removed from the dropdown menu on the side. Similarly, if there are only sensors configured in your file then Blocks will not show the blocks used for motors to minimize the sidebar.

For this reason it is important to create a configuration file BEFORE trying to code!

Below you can see a comparison of Blocks when a configuration file is not selected (left) versus when one is active (right):

 If you have started a program before configuring or have the wrong configuration selected, activate the correct configuration file on the Driver Hub then reopen your Blocks program after the Robot Controller has restarted.

## Runtime Errors

Within the SDK **runtime errors** occur during initialization or run. One of the most common runtime errors within the Control Hub can be seen below:

There are a few different reasons this error typically occurs:

- No configuration file is currently active or created
- The incorrect configuration file is active
- There is a mismatched name between the configuration file and code (ex: rightmotor vs right\_motor)

What results is when the program begins the robot is forced to stop running when the first hardware device is not properly identified. That first hardware device is the one indicated in the error. If there are multiple issues the next will show once the initial is fixed.

Because Blocks handles importing the hardwareMap for you it will additionally show an error when opening a saved OpMode while the wrong configuration file is selected:

