Olá robô - Configuração

- Pré-configuração
- Motor
- Servo
- Dispositivos Digitais
- Dispositivos I2C
- Salvando a configuração
- Erros comuns

Pré-configuração

A configuração é uma das etapas mais comumente mal compreendidas ou esquecidas necessárias para programar um robô. Esta seção tem o objetivo de explicar a importância da configuração e dissipar equívocos comuns sobre ela, respondendo às seguintes perguntas:

1. O que é configuração?
2. Como você configura elementos de hardware?
3. Quais são os problemas comuns causados por um problema com o arquivo de configuração?

A importância da configuração

Embora cada REV Control Hub seja o mesmo, os robôs controlados pelo Control Hub não são. Cada Control Hub tem o mesmo número de portas de motor, portas de servo, portas digitais, e assim por diante, mas como cada usuário utiliza essas portas varia de sistema para sistema. Por exemplo, um Sensor de Cor V3 pode ser conectado à I2C Bus 1 no Hub de um usuário, mas outro usuário pode usar o mesmo barramento para conectar um Sensor de Distância de 2m.

O Control Hub sabe que há um dispositivo I2C conectado à porta, mas não possui naturalmente as informações necessárias para traduzir essas informações para um Op Mode ou informar ao Op Mode quais drivers precisam ser acessados para usar esse sensor. O usuário precisa fornecer informações adicionais, para que o software interno no Hub possa pegar informações do Op Mode e aplicá-las a uma porta de hardware externa correspondente e vice-versa. Esse processo é conhecido como mapeamento de hardware. O mapeamento de hardware é um processo de duas etapas que inclui: a criação de um arquivo legível conhecido como arquivo de configuração e chamadas ao mapeamento de hardware dentro de um Op Mode.

Arquivo de configuração

O arquivo de configuração é um arquivo legível criado pelo usuário por meio do aplicativo Driver Station. Ao criar um arquivo de configuração, os usuários são obrigados a atribuir cada dispositivo a uma porta, selecionar o tipo de dispositivo a partir das opções fornecidas pelo SDK e dar a ele um nome exclusivo.

> Na programação, é importante distinguir entre variáveis, dando a cada variável um nome diferente.

1. Assim que um arquivo de configuração é salvo ou ativado, o robô será reiniciado. Esse reinício é feito para que o SDK possa ler o arquivo, determinar quais dispositivos estão presentes e adicionar os dispositivos à classe hardwareMap.

Mapeamento de hardware

No lado do usuário, no modo de operação criado, está a classe hardwareMap. Esta classe é onde as informações criadas na configuração estão disponíveis para uso no código Blocks, OnBot Java ou Android Studio.

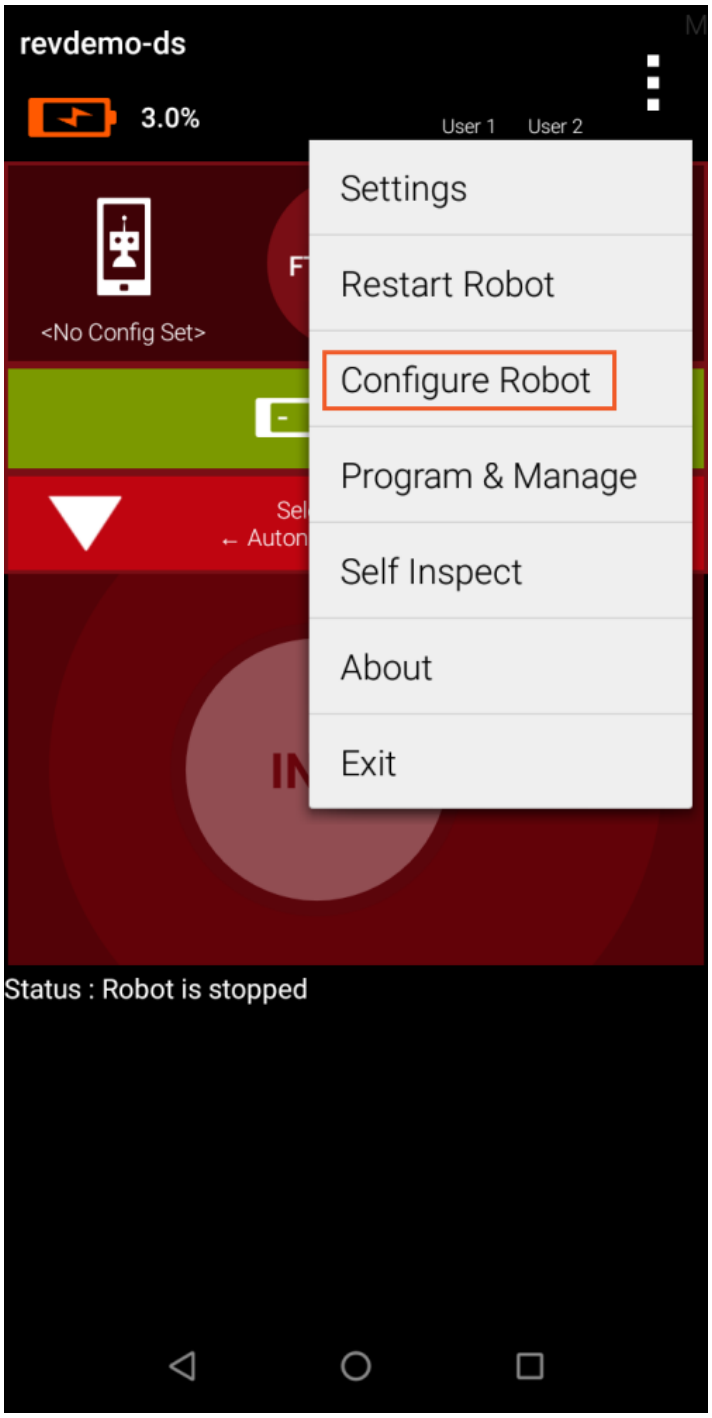
O nível de acesso ou interação que um usuário tem com a classe hardwareMap depende da ferramenta de programação que estão usando. Como o Blocks é uma coleção de trechos de código predefinidos, ele cria referências ao hardwareMap sempre que um trecho de código de variável, correspondente a um hardware externo, é referenciado pela primeira vez. No entanto, com OnBot Java e Android Studio, a referência ao hardwareMap requer a criação de uma variável atribuída a uma unidade de hardware externa dentro do hardwareMap.

> As informações sobre como fazer referência à classe hardwareMap em Java serão explicadas com mais detalhes na seção Banco de dados - OnBot Java.

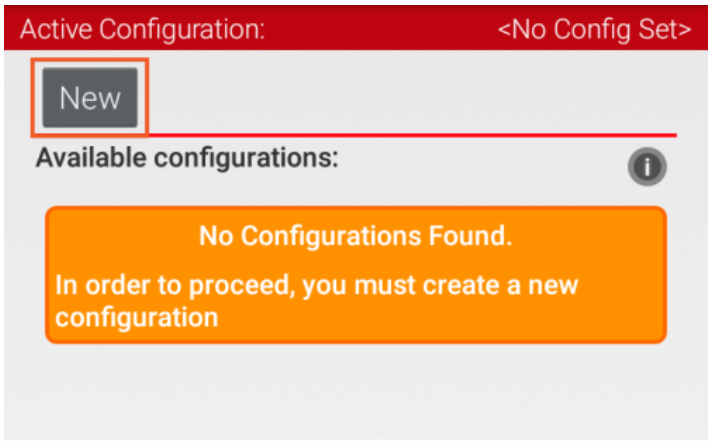
Configurando dispositivos de hardware comum

Acessando o utilitário de configuração

Selecione o menu no canto superior direito da Estação do Motorista. Em seguida, selecione "Configure Robot" (Configurar Robô).

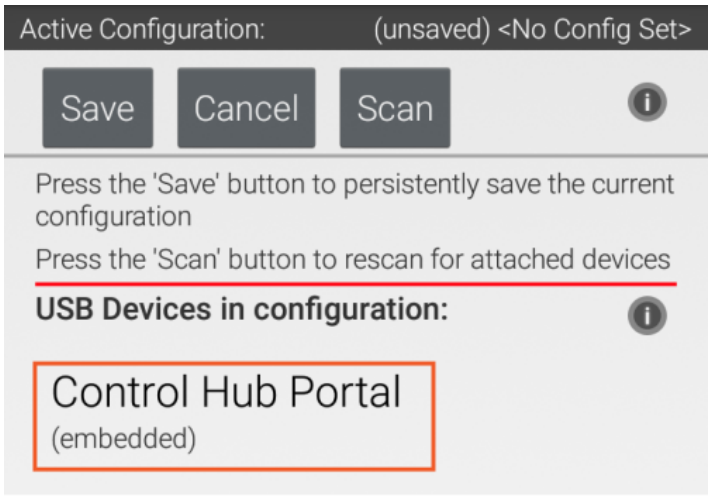


Na página de configurações disponíveis, selecione "New" (Novo).



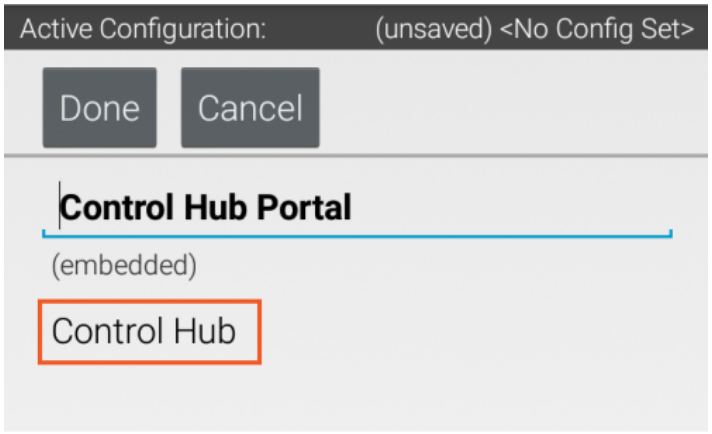
Na página de configuração de dispositivos USB, selecione "Control Hub Portal" (Portal do Control Hub).

Observação: Se você tiver um Expansion Hub, ele aparecerá como "Expansion Hub Portal" (Portal do Expansion Hub).



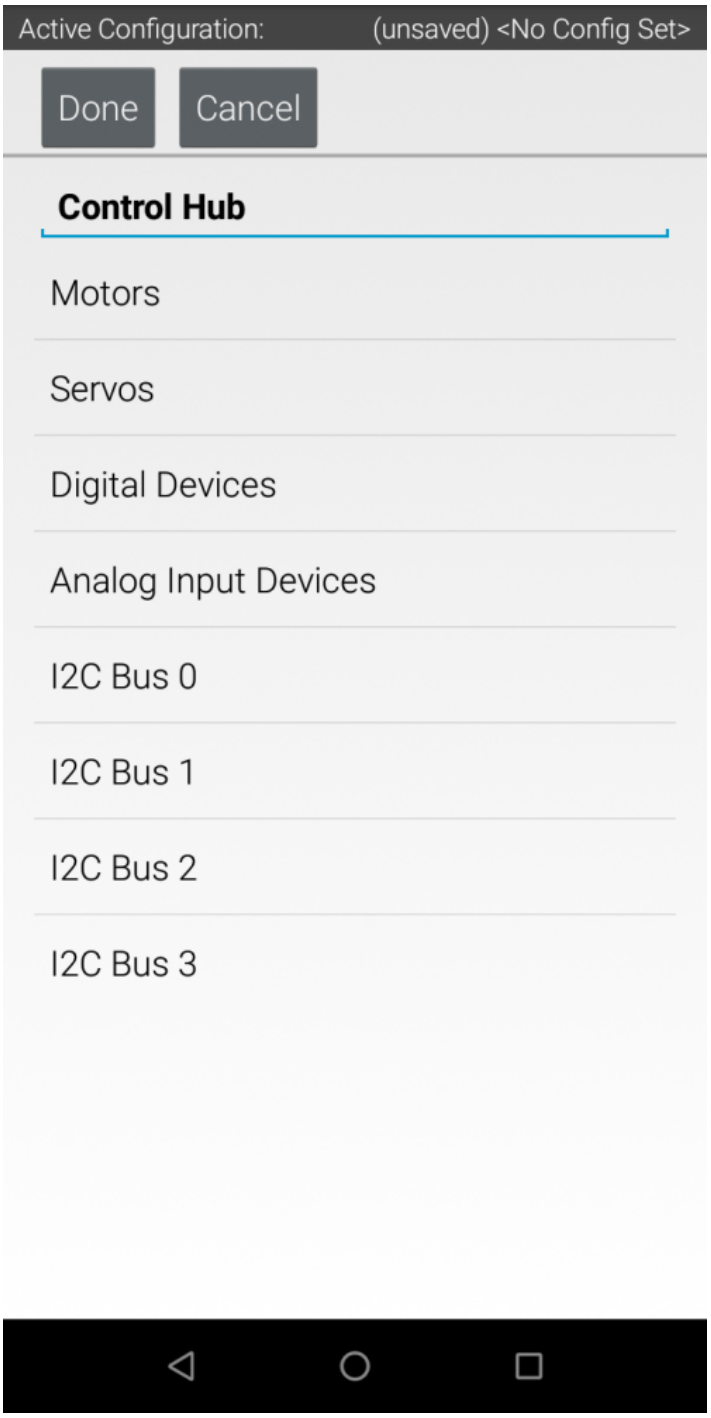
Dentro do Portal do Hub, selecione o dispositivo que deseja configurar. Neste caso, selecione o Control Hub.

Observação: se você tiver um Expansion Hub conectado a um Control Hub, o Expansion Hub também aparecerá como um dispositivo configurável no portal.



Isso o levará para a página mostrada na imagem. A partir daqui, você pode configurar motores, servos e sensores que está usando. Siga o restante do guia para descobrir como configurar dispositivos que serão usados na seção de Teste.

Observação: A forma como os dispositivos Digitais e Analógicos são configurados difere significativamente da forma como os dispositivos I2C são configurados. Isso ocorre porque cada porta física I2C é um barramento diferente que pode hospedar vários sensores diferentes. Para obter mais informações sobre os diferentes tipos de sensores, consulte a seção de sensores.



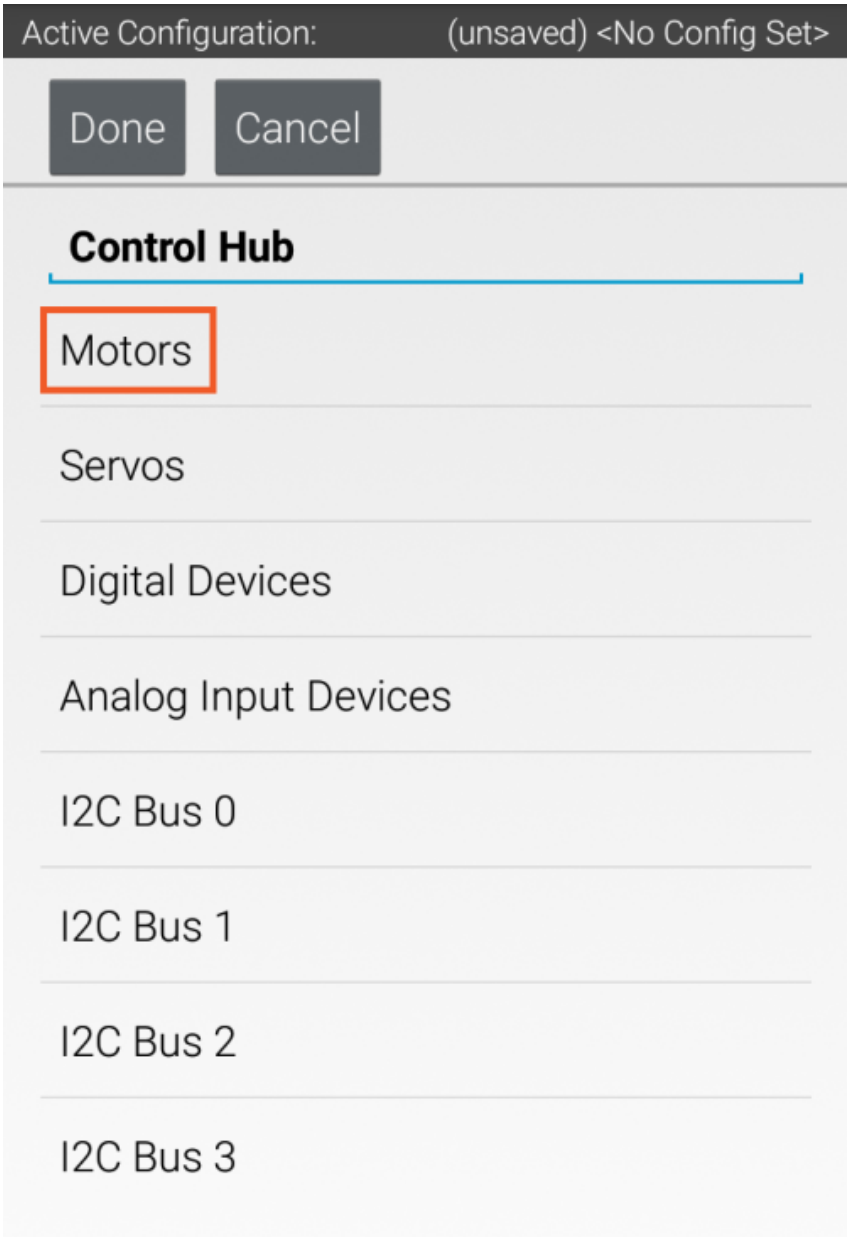
Configurando o hardware

A próxima seção mostrará como configurar componentes que serão usados no Test Bed. O tipo de hardware e os nomes foram escolhidos levando em consideração o plano de aula Hello World. Os usuários devem observar as notas dentro das etapas para considerar ao criar arquivos de configuração para outras instâncias.

Motor

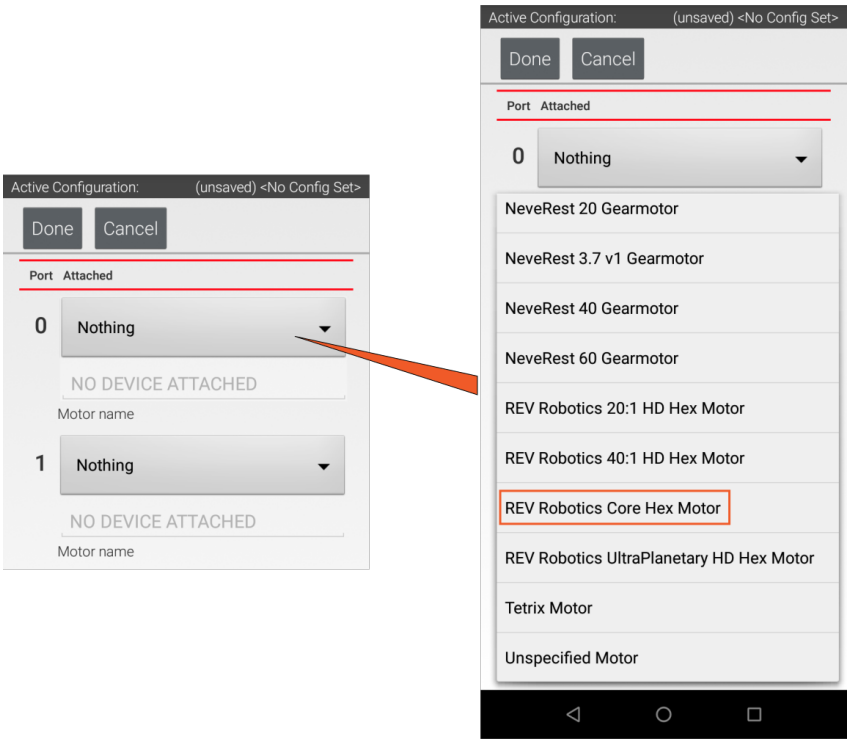
Motor

Selecione *Motors*



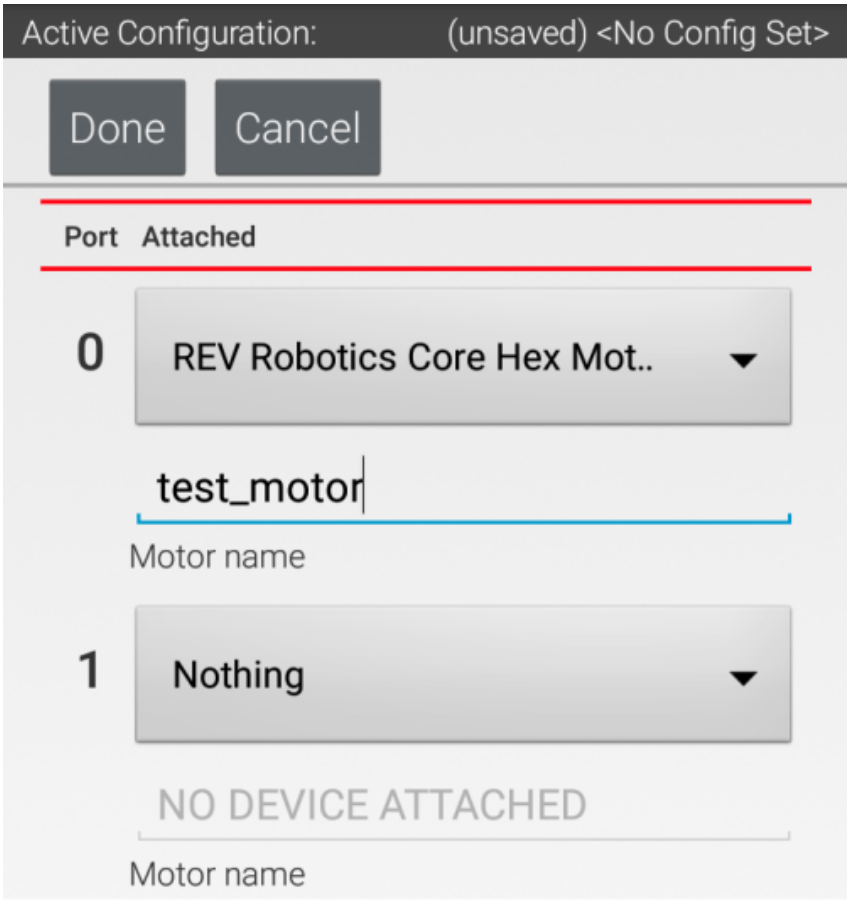
A página do Motor permitirá que você configure todas as quatro portas de motor no Hub. Na Porta 0, abra o menu suspenso e selecione "REV Robotics Core Hex Motor" (Motor Hexagonal Core da REV Robotics).

Observação: No seu arquivo de configuração, você deve configurar as portas de motor para o tipo de motor que está usando.



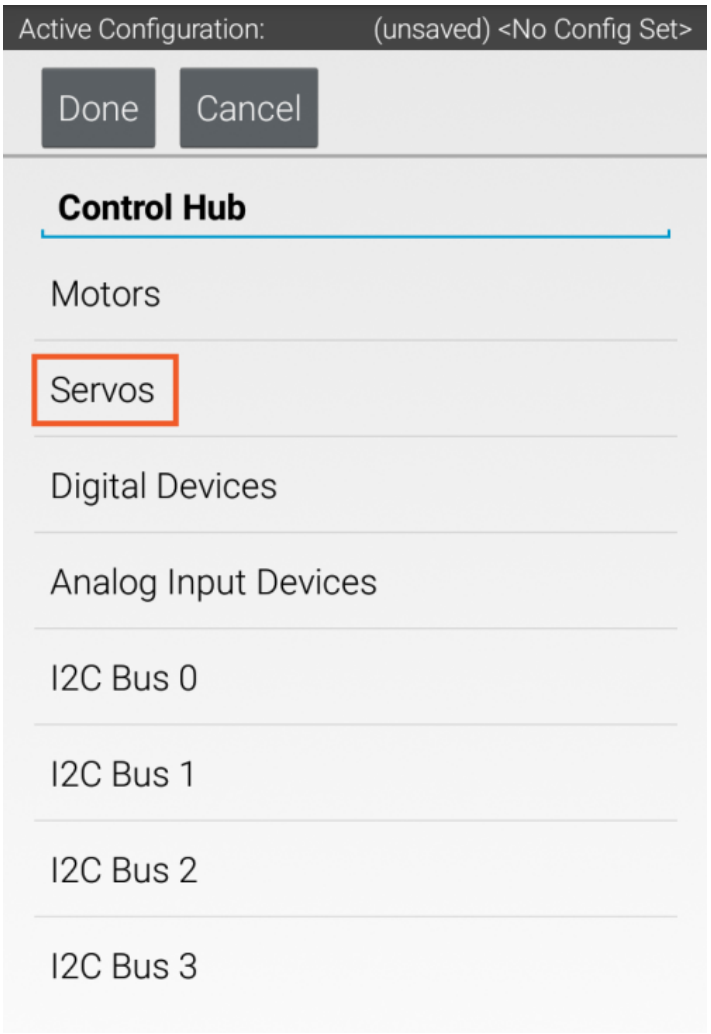
Dê o nome de "test_motor" ao motor. Selecione "done" (concluído).

Observação: Lembre-se de que, ao nomear hardware no arquivo de configuração, o REV Control System diferencia maiúsculas de minúsculas (case-sensitive).



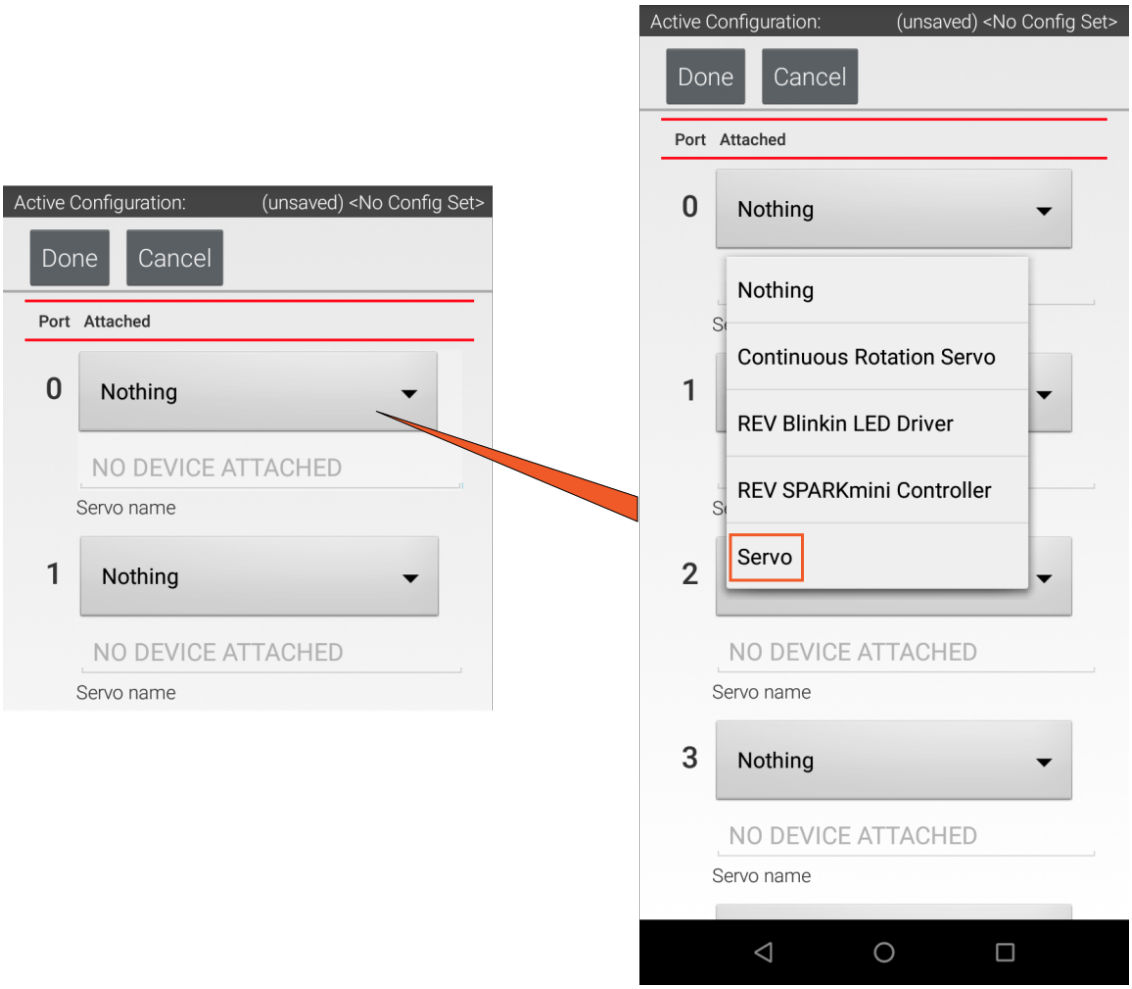
Servo

Selecione *Servos*



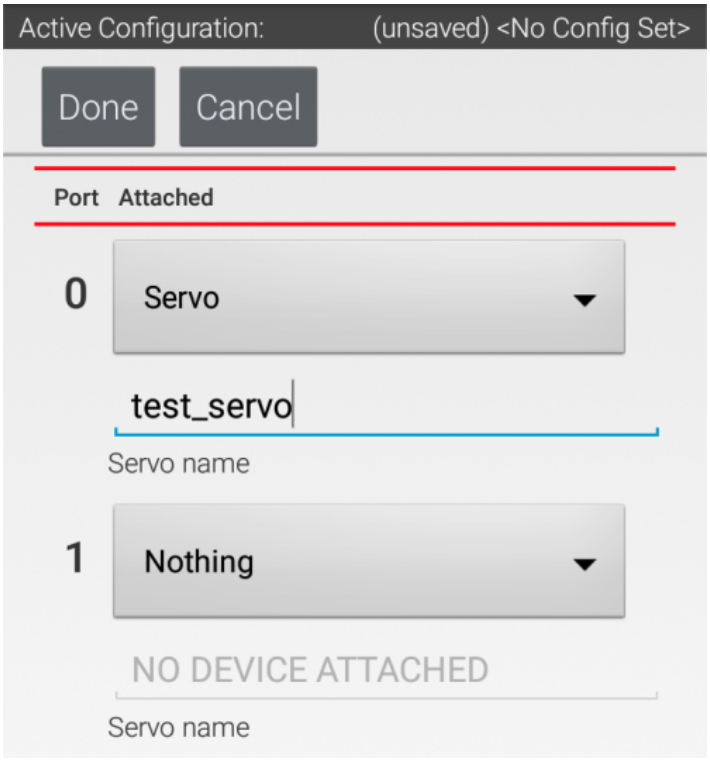
A página do Servo permitirá que você configure todas as seis portas de servo no Hub. Na Porta 0, abra o menu suspenso e selecione Servo.

Observação: Os Servos da REV podem ser configurados como um Servo comum ou como um Servo de Rotação Contínua. O tipo de dispositivo ao qual um servo é configurado deve corresponder ao modo no qual o sensor está. Para obter mais informações sobre os modos de sensor, consulte a seção de Sensores da REV Robotics.



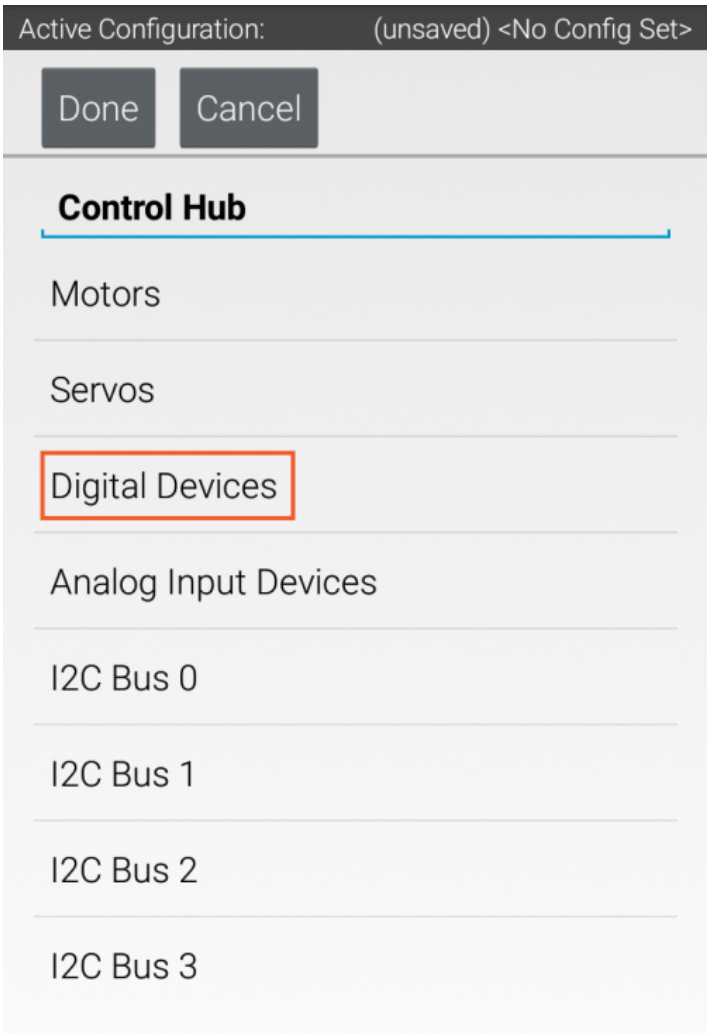
Nomeie o servo como test_servo. Selecione "Done" (Concluído).

Observação: lembre-se de que ao nomear hardware no arquivo de configuração, o REV Control System diferencia maiúsculas de minúsculas.



Dispositivos Digitais

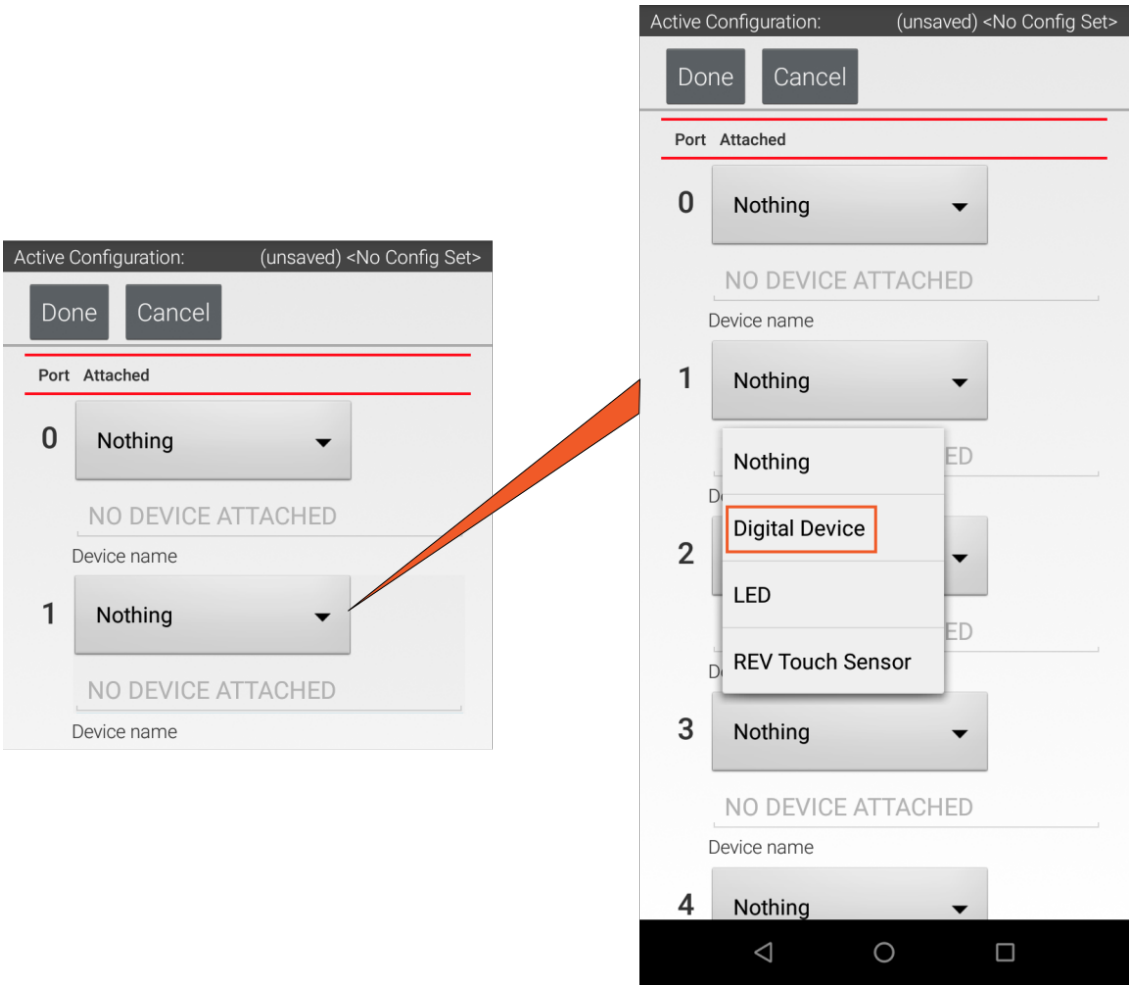
Selecione *Digital Devices*



A página de Dispositivos Digitais permitirá que você configure todas as oito portas digitais no Hub. Na Porta 1, abra o menu suspenso e selecione Dispositivo Digital.

Observação: Sensores de Toque (Touch Sensors) devem sempre ser configurados em portas de números ímpares. Consulte a seção de Sensores Digitais para obter mais informações.

Observação: Sensores de Toque podem ser configurados como um REV Touch Sensor ou como um Dispositivo Digital. No FTC SDK, o tipo de dispositivo configurado altera as classes e métodos que podem ser usados.



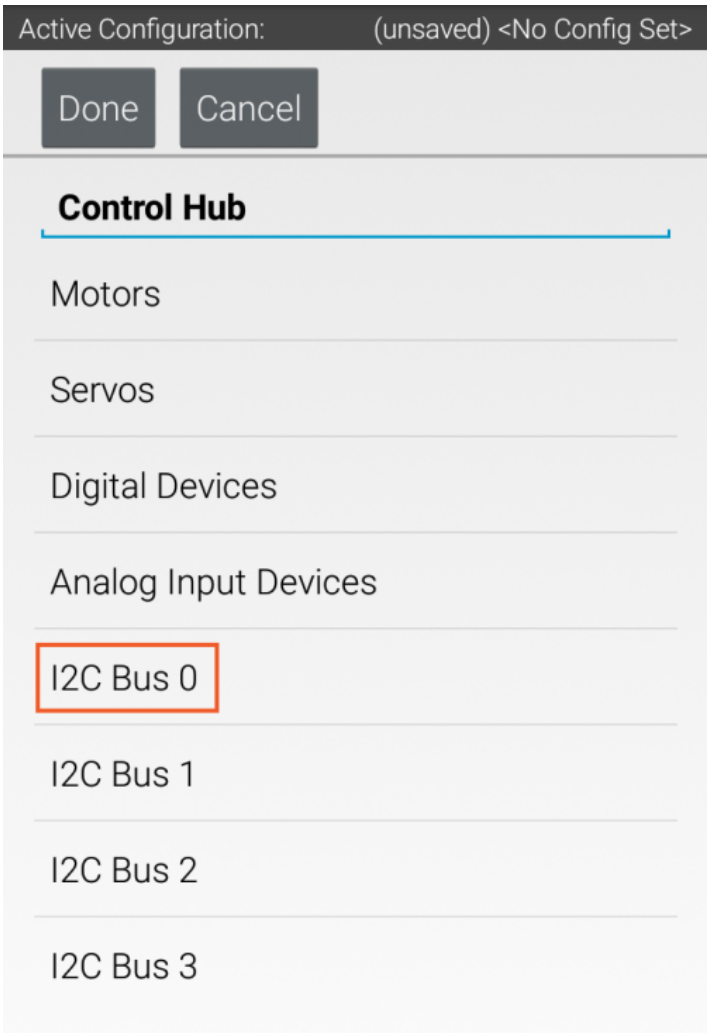
Nomeie o dispositivo digital como test_touch. Selecione "Done" (Concluído).

Observação: lembre-se de que ao nomear hardware no arquivo de configuração, o REV Control System diferencia maiúsculas de minúsculas.



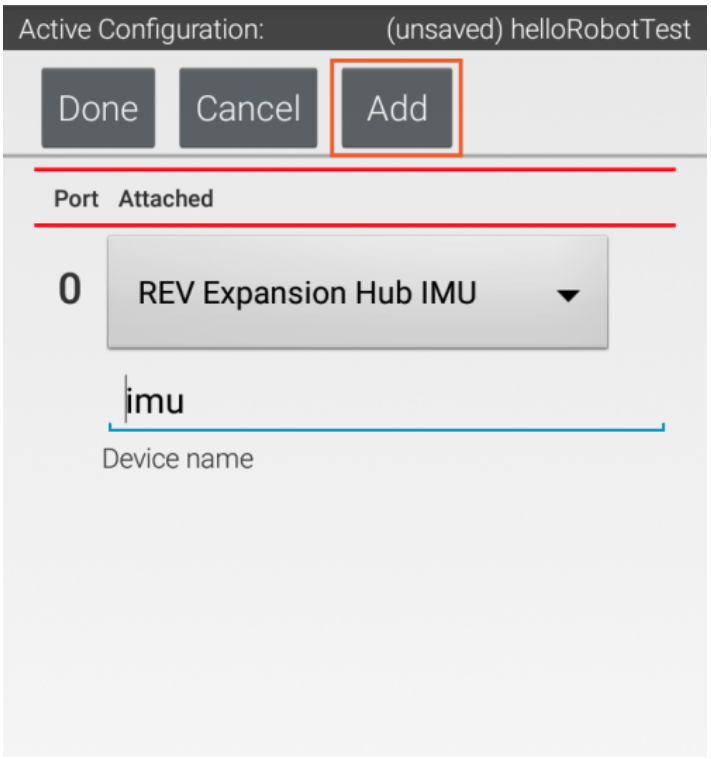
Dispositivos I2C

Selecione *I2C Bus 0*



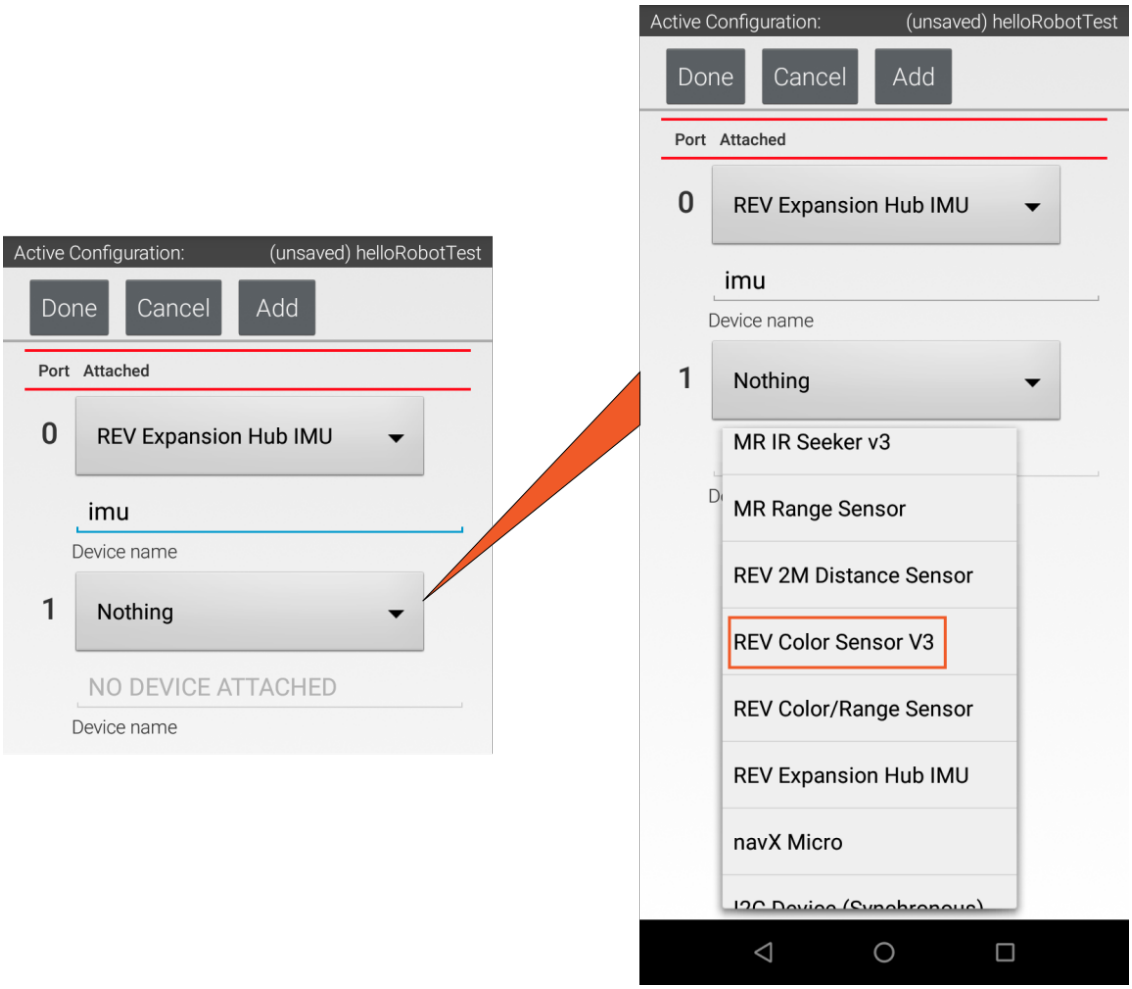
Selecione "Add" (Adicionar).

Observação: Cada barramento I2C pode hospedar mais de um sensor I2C, desde que os endereços I2C não entrem em conflito. O Barramento 0 sempre hospedará o IMU interno. Para obter mais informações sobre sensores I2C, consulte a seção de I2C.



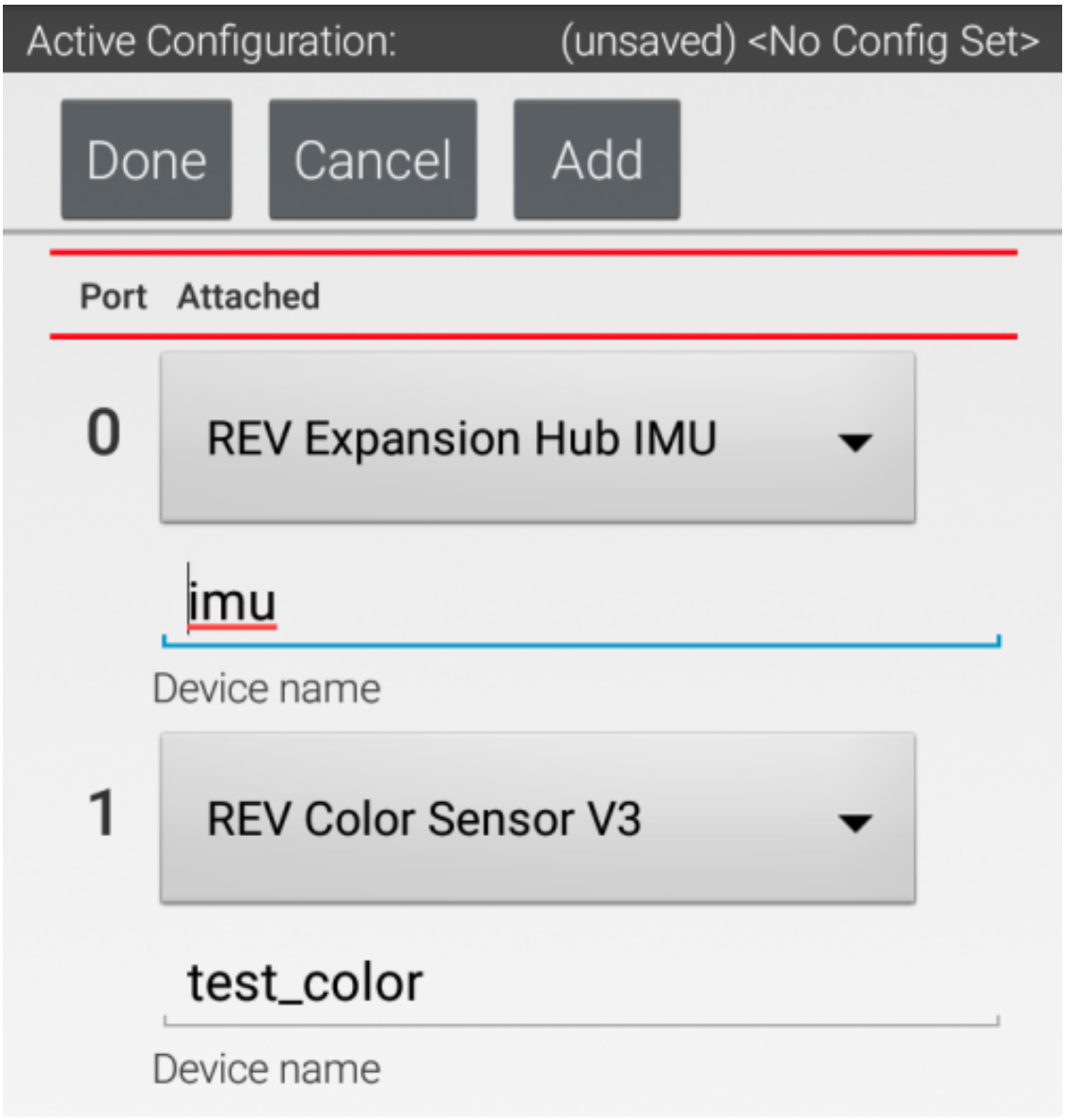
Na Porta 1, que foi criada na etapa anterior, abra o menu suspenso e selecione REV Color Sensor V3.

Observação: Se estiver usando sensores de cor V1 ou V2, selecione REV Color/Range Sensor. Para obter mais informações sobre a configuração com os sensores de cor REV, consulte as Datasheets do Sensor de Cor.



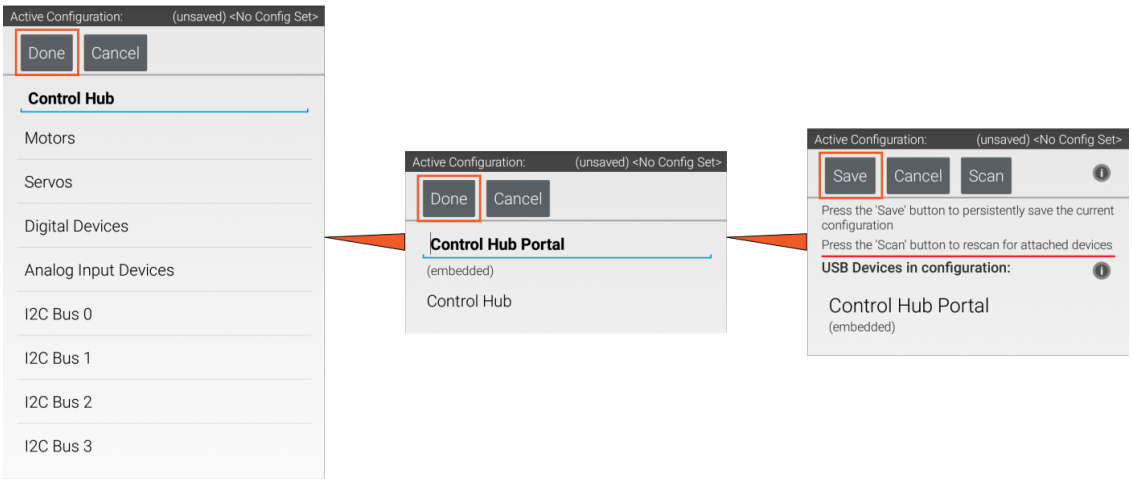
Nomeie o sensor de cor como test_color. Selecione "Done" (Concluído).

Observação: lembre-se de que ao nomear hardware no arquivo de configuração, o REV Control System diferencia maiúsculas de minúsculas.


Active Configuration:
(unsaved) <No Config Set>
Done
Cancel
Add
Port
Attached
0
REV Expansion Hub IMU
imu
Device name
1
REV Color Sensor V3
test_color
Device name

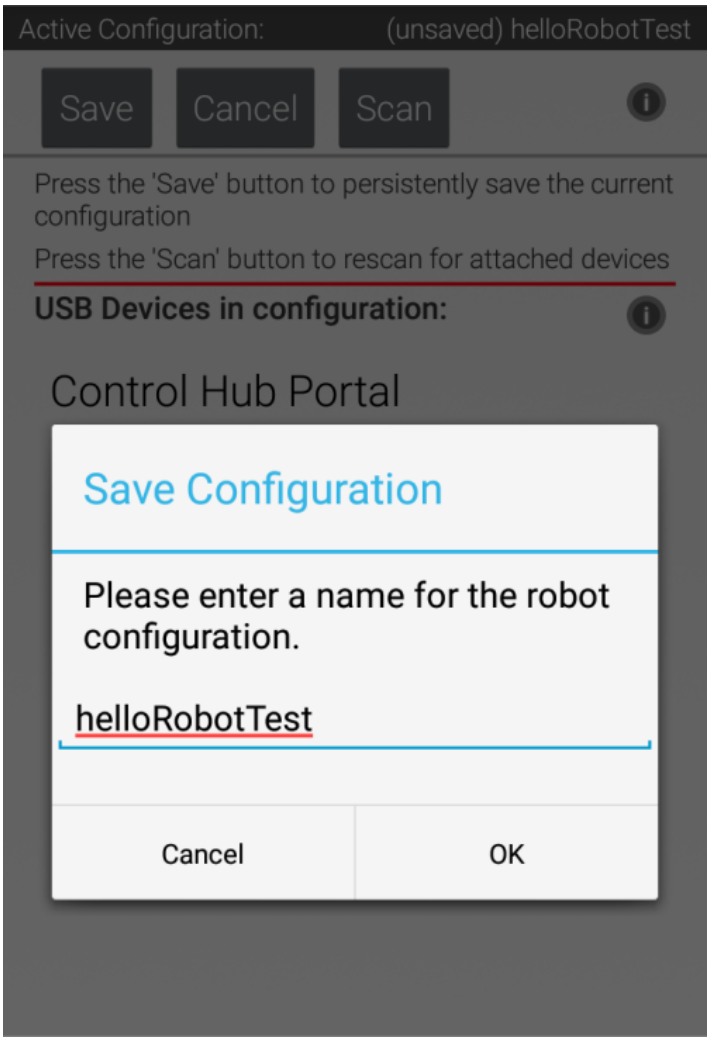
Salvando a configuração

Pressione "Done" duas vezes até chegar à página de Dispositivos USB na configuração. Na página de Dispositivos USB na configuração, pressione "Save" (Salvar).

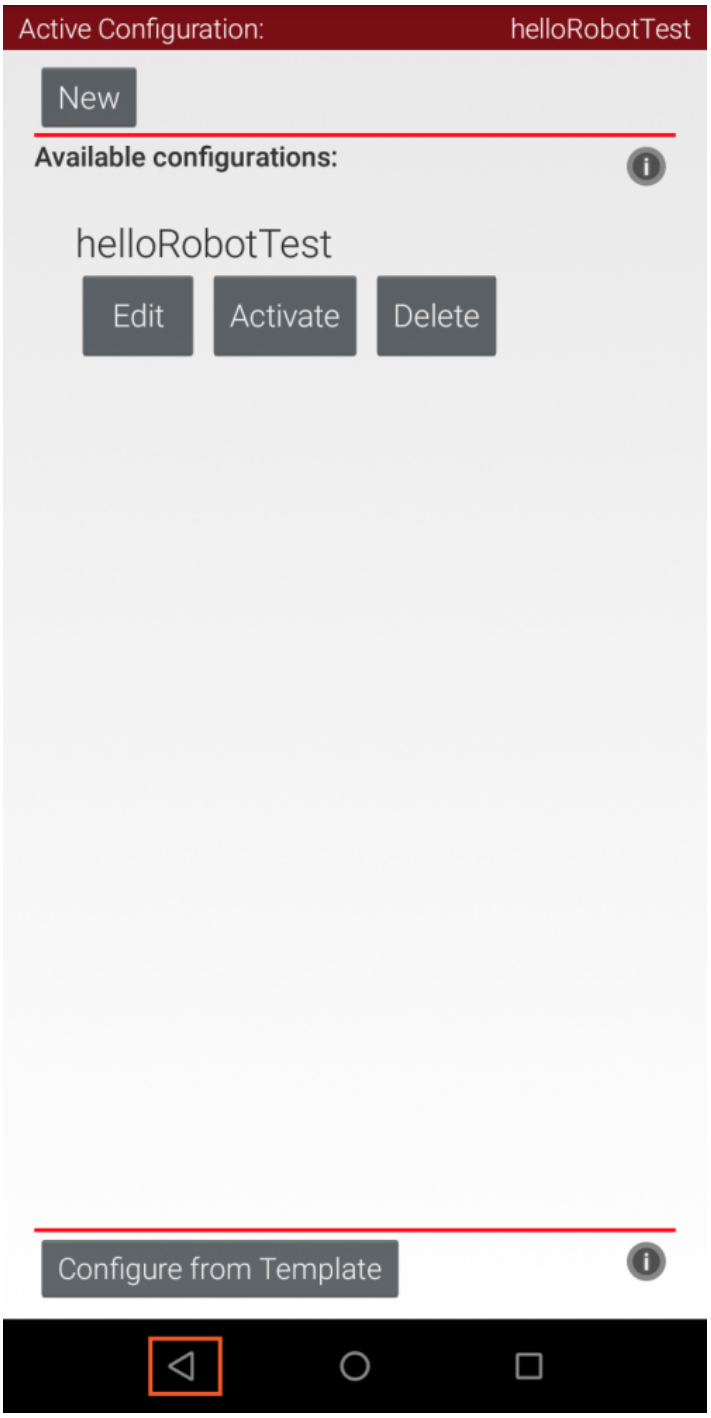


Dê o nome de "helloRobotTest" à configuração e selecione "Ok" (OK).

Observação: O FTC SDK não obriga que você siga uma convenção de nomenclatura específica, mas é comum nomear configurações em lowerCamelCase.



Pressione "back" para ativar a configuração salva. Seu Controlador do Robô reiniciará assim que você ativar uma nova configuração.


Active Configuration:
helloRobotTest
New
Available configurations:
helloRobotTest
Edit
Activate
Delete
Configure from Template

Erros comuns

Dentro do mundo da programação e do software, os erros podem se apresentar em muitas formas e tipos diferentes. Ao mapear hardware, há dois erros principais que você pode encontrar. Ambos os erros se enquadram em categorias comuns de erros de software:

Erros de Interface - são erros entre como uma interface deveria funcionar e como ela realmente se comporta.

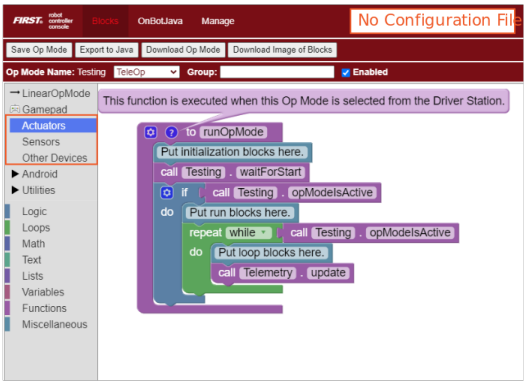
Erros em Tempo de Execução - são erros que ocorrem quando um programa está sendo executado.

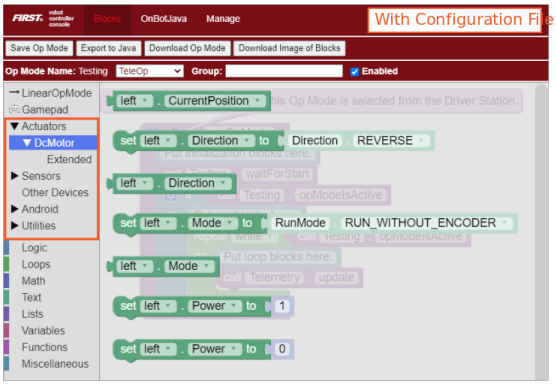
Erro de interface

Os erros de interface ocorrem no SDK quando os parâmetros da interface do SDK não são atendidos. No processo de mapeamento de hardware, o erro de interface mais comum ocorre com a ferramenta de programação Blocks. Como mencionado na seção de Introdução à Programação, o Blocks oculta as complexidades da biblioteca do SDK dos usuários. Uma maneira de fazer isso é criando automaticamente referências ao hardwareMap quando trechos de código para uma unidade de hardware externo são usados.

Para automatizar as chamadas ao hardwareMap, a interface Blocks lê o arquivo de configuração e cria variáveis de hardware com base nas informações encontradas. Por esse motivo, é importante criar um arquivo de configuração antes de tentar codificar.

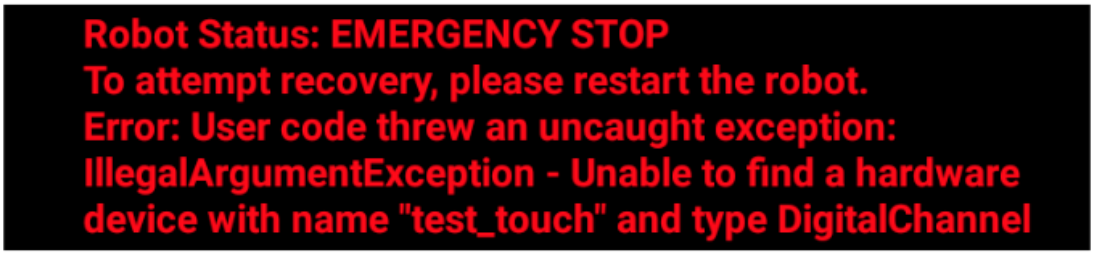
A imagem abaixo mostra duas versões diferentes da interface Blocks. Na versão sem arquivo de configuração, não há menus suspensos para acessar trechos de código específicos para atuadores ou sensores. Na versão da interface com um arquivo de configuração, os menus suspensos estão presentes e os trechos de código específicos para motores estão acessíveis.





Erros de tempo de execução

Dentro do SDK, erros em tempo de execução ocorrem durante a inicialização ou execução. Um dos erros em tempo de execução mais comuns dentro do REV Control System é exemplificado na imagem abaixo.



Esse erro indica uma inconsistência entre como um dispositivo de hardware é chamado no código e como isso se compara ao nome usado no arquivo de configuração. Existem duas maneiras diferentes de ocorrer esse erro.

A primeira ocorrência desse erro é quando não há um arquivo de configuração encontrado. Isso pode significar que um arquivo de configuração não foi criado, um arquivo foi criado, mas não está ativo, ou o arquivo errado está sendo usado. Quando qualquer uma dessas situações acontece, o código está solicitando um nome e tipo de dispositivo que o hardware map não consegue localizar no arquivo de configuração. O programa para no primeiro nome de dispositivo que não consegue localizar.

Uma referência incorreta ao hardwareMap também pode causar esse erro. Ao contrário do Blocks, OnBot Java e Android Studio exigem que um programador codifique manualmente a chamada ao hardwareMap. Se o nome de referência na chamada não corresponder ao nome do dispositivo no arquivo de configuração (é sensível a maiúsculas e minúsculas), o código será compilado sem falhas, mas o erro em tempo de execução ocorrerá. Vamos usar o arquivo de configuração da seção anterior como exemplo, onde há um sensor de toque chamado "test_touch" e um motor chamado "test_motor".

> As aspas indicam que "test_touch" e "test_motor" representam uma variável de string que corresponde aos nomes dos dispositivos na configuração.

```
public class HR_test extends LinearOpMode{
    private DigitalChannel test_touch;
    private DcMotor test_motor;

    @Override
    public void runOpMode(){
```

```
        //get the touch sensor and motor from hardwareMap
        test_touch = hardwareMap.get(DigitalChannel.class, "test_touch");
        test_motor = hardwareMap.get(DcMotor.class, "test_moto");


    }
}
```

Observe no exemplo de linhas de código que o hardwareMap.get() para "test_motor" está escrito como "test_moto" em vez de "test_motor". Quando o código está sendo compilado, não há uma verificação imediata de que o nome solicitado está no hardwareMap. Essa verificação é feita quando o código é executado no robô. Quando a comunicação com o hardwareMap é iniciada, ele procura por "test_moto" e, quando não consegue encontrar, cria o erro em tempo de execução mencionado acima.