

Configurando Servo Hub

Esta página abordará informações sobre conceitos de configuração específicos do Servo Hub. Para obter mais informações sobre a configuração geral no REVLlib, consulte esta página.

Classes de Configuração

Embora o Servo Hub tenha sua própria classe de configuração, `ServoHubConfig`, a maior parte da configuração ocorre na classe `ServoChannelConfig`.

API Documentation

ServoHubConfig	Java	C++
--------------------------------	----------------------	---------------------

Parâmetros Persistentes

A configuração de um Servo Hub mantém automaticamente as definições de configuração entre os ciclos de energia ao chamar o método `configure()`.

A persistência dos parâmetros envolve salvá-los na memória do Servo Hub, o que consome muito tempo e bloqueia a comunicação com o dispositivo.

Casos de uso

Recomenda-se atualizar a maioria dos parâmetros do dispositivo durante a configuração inicial do dispositivo no início do programa para garantir que o controlador mantenha sua configuração no caso de um ciclo de energia durante a operação, por exemplo, devido a um disparo do disjuntor ou a uma queda de energia.

Em geral, recomenda-se não fazer atualizações na configuração durante a operação para evitar o bloqueio do programa e afetar o desempenho do robô.

Abaixo está um exemplo de um desses casos:

■ ■

JAVA

```
Robot() {  
    ServoHubConfig config = new ServoHubConfig();  
    config  
        .channel0.pulseRange(500, 1500, 2500)  
        .disableBehavior(ServoChannelConfig.BehaviorWhenDisabled.kSupplyPower);  
}
```

```

// Persist parameters and reset any not explicitly set above to
// their defaults.
servoHub.configure(config, ServoHub.ResetMode.kResetSafeParameters);
}

void reduceRange() {
    ServoHubConfig config = new ServoHubConfig();
    config.channel0.pulseRange(500, 1500, 2500)

    // Don't reset the parameters the at are not explicitly set above
    servoHub.configure(config, ServoHub.ResetMode.kNoResetSafeParameters);
}

```

Copy

```

using namespace rev::servohub;

Robot() {
    ServoHubConfig config;
    config
        .channel0.pulseRange(500, 1500, 2500)
        .disableBehavior(ServoChannelConfig::BehaviorWhenDisabled::kSupplyPower);

    // Persist parameters and reset any not explicitly set above to
    // their defaults.
    servoHub.configure(config, ServoHub::ResetMode::kResetSafeParameters);
}

void reduceRange() {
    ServoHubConfig config;
    config.channel0.pulseRange(500, 1500, 2500)

    // Don't reset the parameters the at are not explicitly set above
    servoHub.configure(config, ResetMode.kNoResetSafeParameters);
}

```

■ ■

JAVA

```

Robot() {
    ServoHubConfig config = new ServoHubConfig();

```

```

config
    .channel0.pulseRange(500, 1500, 2500)
    .disableBehavior(ServoChannelConfig.BehaviorWhenDisabled.kSupplyPower);

// Persist parameters and reset any not explicitly set above to
// their defaults.
servoHub.configure(config, ServoHub.ResetMode.kResetSafeParameters);
}

void reduceRange() {
    ServoHubConfig config = new ServoHubConfig();
    config.channel0.pulseRange(500, 1500, 2500)

    // Don't reset the parameters the at are not explicitly set above
    servoHub.configure(config, ServoHub.ResetMode.kNoResetSafeParameters);
}

```

C++

```

using namespace rev::servohub;

Robot() {
    ServoHubConfig config;
    config
        .channel0.pulseRange(500, 1500, 2500)
        .disableBehavior(ServoChannelConfig::BehaviorWhenDisabled::kSupplyPower);

    // Persist parameters and reset any not explicitly set above to
    // their defaults.
    servoHub.configure(config, ServoHub::ResetMode::kResetSafeParameters);
}

void reduceRange() {
    ServoHubConfig config;
    config.channel0.pulseRange(500, 1500, 2500)

    // Don't reset the parameters the at are not explicitly set above
    servoHub.configure(config, ResetMode.kNoResetSafeParameters);
}

```

Revisão #2

Criado 9 maio 2025 19:10:02 por Team stemOS

Atualizado 9 maio 2025 20:23:40 por Team stemOS