

Servos

- Conexão elétrica
- Programação
- Lógica
- Exercício

Conexão elétrica

Os servos tem uma conexão simples, no sentido de que os cabos geralmente são integrados no corpo do *hardware*. Entretanto, é preciso ficar atento com o sentido da conexão, visto que é o único contato da placa que é possível inverter.

Assim o primeiro passo para conectar o servo é identificar três pinos:

- GND - *Ground*
- Vcc - *Alimentação*
- Signal - *Sinal de controle*

Existem cores padrões para certas fabricantes, mas na dúvida sempre cheque o manual dos equipamentos.

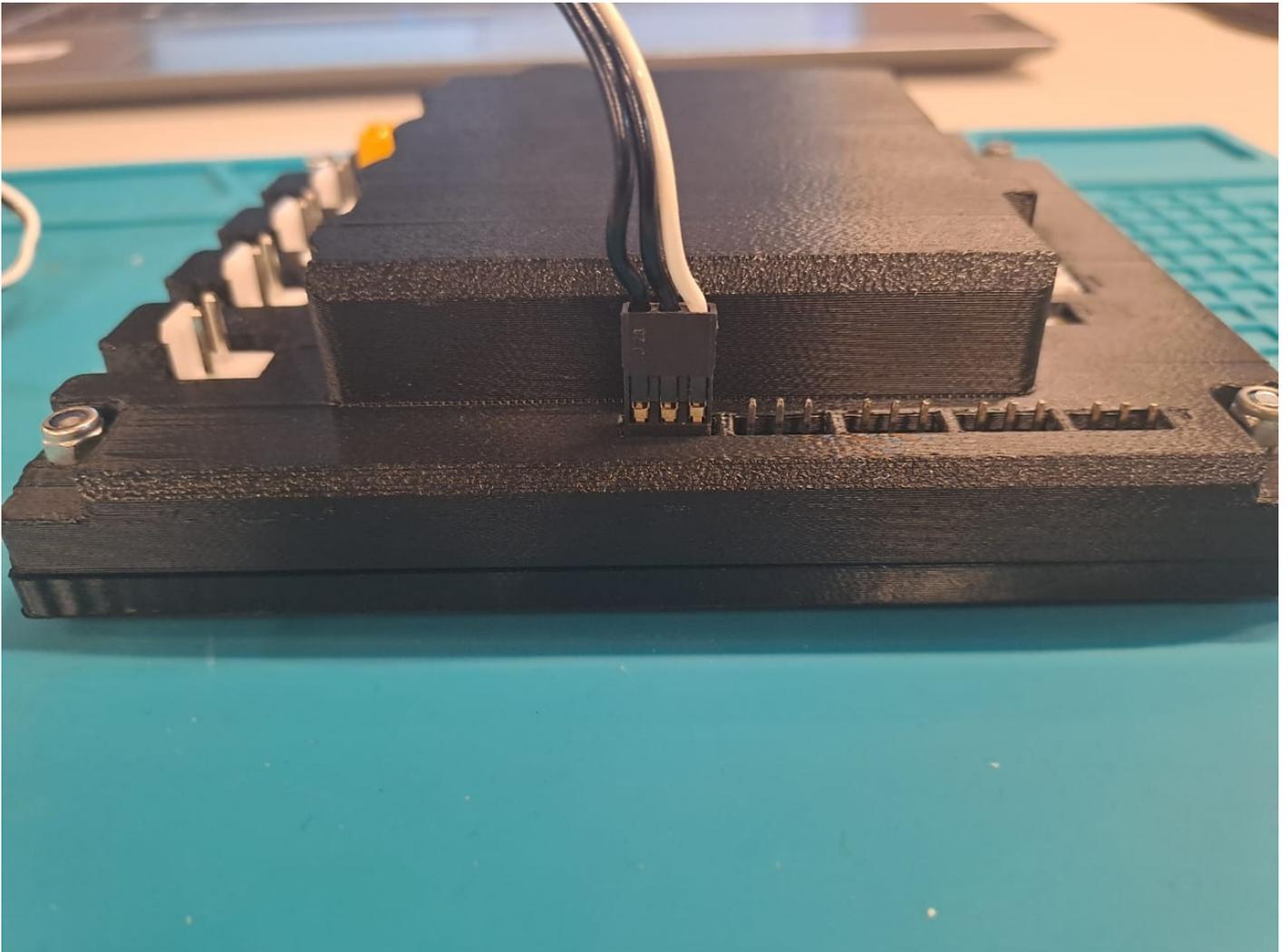
Nessa seção será utilizado um [Smart Servo](#) da REV Robotics

No Smart Servo, o cabo de sinal tem a coloração branca, o do meio é o Vcc e o restante é o GND. Como mostra a imagem abaixo:



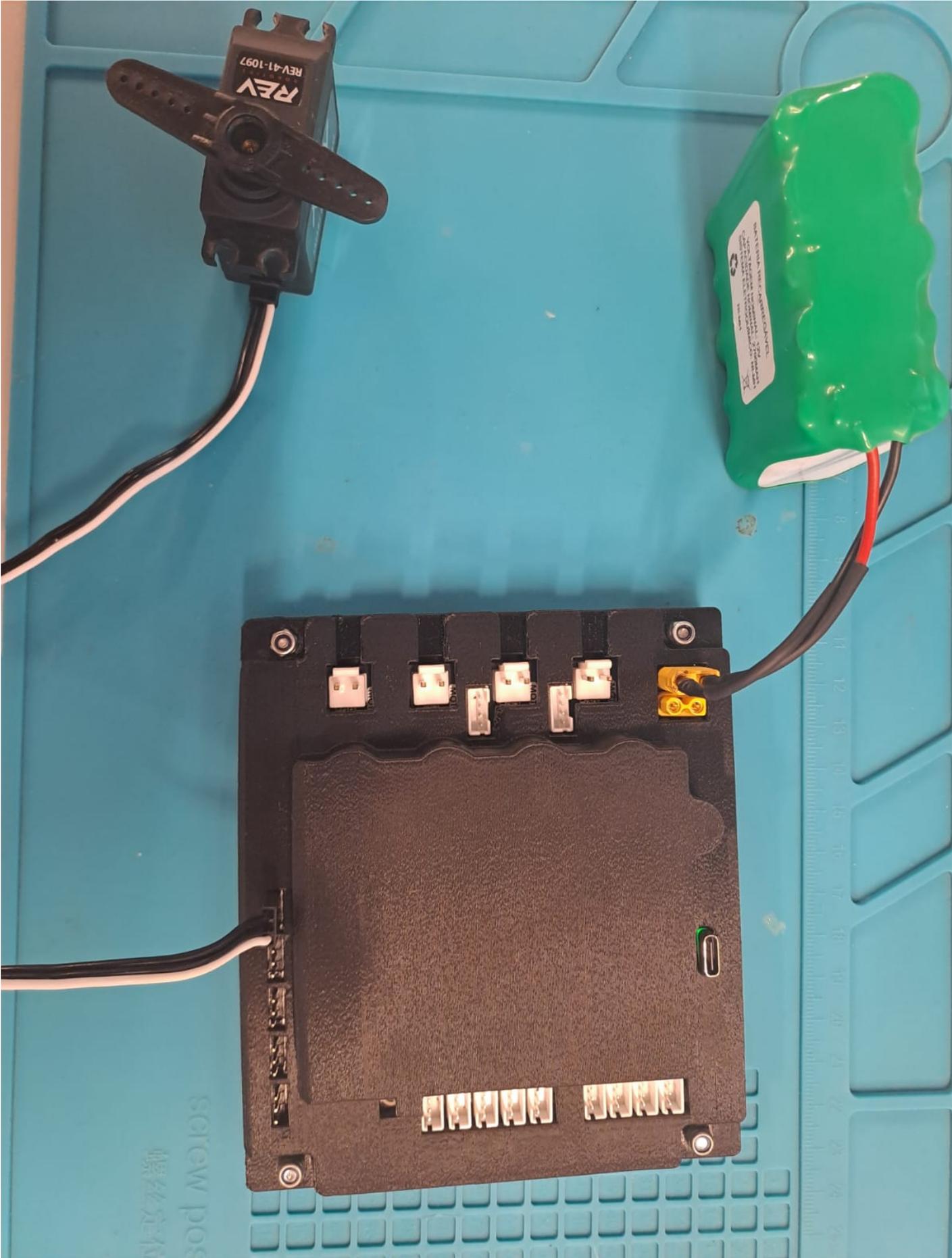
Quanto a conexão na placa EasySTEAM, é necessário verificar quais pinos correspondem ao terminal de saída do servo - para não acabar invertendo sinal com GND por exemplo. Mas visto isso, já é possível conectar, como seguem as demonstrações abaixo:





Alimentação

Apesar de os servos atuarem com uma tensão elétrica de 5V, é recomendado que se tenha uma bateria de 12V conectada a placa - igual ao que foi feito para operar os motores - para não sobrecarregar todo o equipamento envolvido.



Programação

Com a conexão pronta é possível começar a programação do servo. Que consiste em 3 blocos (sendo um para servos contínuos)

Objeto do Servo

Assim como o objeto de motor - leia a seção de [Motores](#), caso não tenha visto ainda - o servo tem um bloco semelhante, no qual podemos escolher quais dos 3 servos estaremos mexendo.



Bloco de posição

Diferente do motor em que definimos um percentual da velocidade de saída, os servos funcionam ditando uma posição (geralmente em graus). Dessa forma, o bloco tem o mesmo significado, ou seja, receber um valor numérico, mas esse valor pode ir de 0 a 270.

O programa não diferencia unidades, mas espera um certo tipo, então no caso dos servos, seria em graus.

Como demonstra:



Bloco de velocidade

Esse bloco só funciona caso o servo tenha modo contínuo, como o Smart Servo utilizado neste exemplo. A função dele é ao invés de definir uma posição é fornecer uma velocidade, semelhante ao motor, mas agora indo de um valor de 0 a 1.0. Como mostrado abaixo:



Não use esse bloco se seu servo não for contínuo

Na próxima página será demonstrado como utilizar esses blocos juntamente com a categoria "*Logical*".

Lógica

Em praticamente qualquer programa de computador é necessário tomar decisões, isso é feito utilizando condicionais, a mesma ideia é necessária ao utilizar essa interface e mecanismos.

As condicionais trabalham com lógica booleana, em resumo verdadeiro ou falso. Isso será melhor visto com a seção abaixo.

Bloco de comparação

Este bloco tem por objetivo fazer comparações numéricas do tipo, **5 > 3** ou **4 > 5** - caso o resultado seja correto ele retorna verdadeiro, se não, falso.



Ele aceita como parâmetros dois tipos numérico que funcionam para a comparação. Como exemplo:

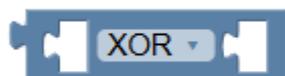


Esses números podem ser substituídos por qualquer outro bloco que retorna um tipo numérico. Como o controle.

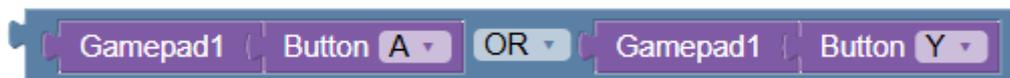
Operadores lógicos

Diferentes dos operadores numéricos que trabalham com números, esses trabalham com expressões de verdadeiro e falso. Para ver mais sobre o assunto acesse: [Operadores lógicos](#)

O bloco que opera com isso é o mostrado abaixo:



Diferente do bloco de comparação, ele recebe valores de verdadeiro e falso (booleanos). Então segue abaixo um exemplo usando o controle:



A expressão acima seria verdadeira caso qualquer um dos dois - ou mesmo os dois - botões estivessem pressionados, portanto, retornando uma instrução verdadeira

Bloco If do

Ainda dentro da categoria "Logical" é possível ver um bloco como o mostrado abaixo:



A ideia principal dele é: "Se esta expressão for verdadeira, faça isso". Para ele verificar a veracidade de algo é necessário uma condição - que em resumo é uma expressão que retorna falso ou verdadeiro.

E como dito nas seções acima, essas expressões podem ser os operadores lógicos, comparadores numéricos, ou até mesmo os valores de botões do controle.

Bloco If Else

Esse bloco é semelhante ao bloco If, com a adição de uma condição "Se não", em resumo, "Se está expressão é verdadeira, faça isso, se não, faça aquilo".



Exercício

Nesta seção será demonstrado como usar a categoria "Logical" para fazer o acionamento de servos.

A ideia principal do programa que será desenvolvido nesta página é:

- Se o botão A for apertado o servo deve girar até 90°;
- Se o botão X for apertado o servo deve girar até 180°;
- Se o botão B for apertado o servo deve girar até 270°;
- Se o botão Y for apertado o servo deve girar até 0°.

Sinta-se a vontade para experimentar outras lógicas, como: Se o botão X for apertado juntamente com o botão B faça isso. A ideia dos exercícios é aprender sobre a implementação de um conjunto de categorias.

```
#include < EasySTEAM .h>

setup EasySTEAM Start

loop
  If Gamepad1 Button A
  Do Servo 1 setPosition 90
  If Gamepad1 Button X
  Do Servo 1 setPosition 180
  If Gamepad1 Button B
  Do Servo 1 setPosition 270
  If Gamepad1 Button Y
  Do Servo 1 setPosition 0
```

O que o código faz é checar qual dos 4 botões está pressionado, sendo que, cada um ao ser apertado define uma posição para o servo.

Faça o carregamento do código para a placa, conforme descrito nos capítulos iniciais, e
TESTE!