

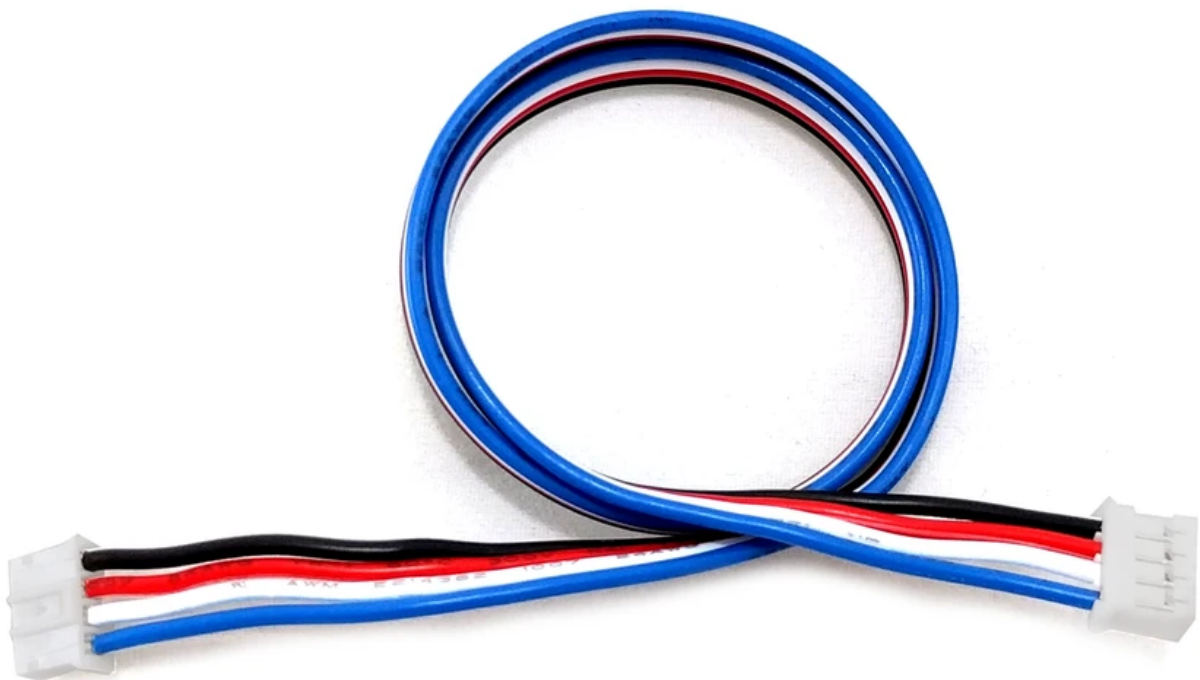
# Sensores Digitais

- [Conexão elétrica](#)
- [Categoria Digital](#)
- [Categoria Text](#)
- [Exercício](#)

# Conexão elétrica

---

As conexões de sensores digitais é feita utilizando cabos JST-PH de 4 pinos, conforme imagem:



Esses contatos encaixam em apenas um sentido, portanto não é possível inverter a conexão.

Nesta página será utilizado um [sensor de limite magnético](#), segue a imagem abaixo mostrando a conexão do cabo JST-PH com o sensor:



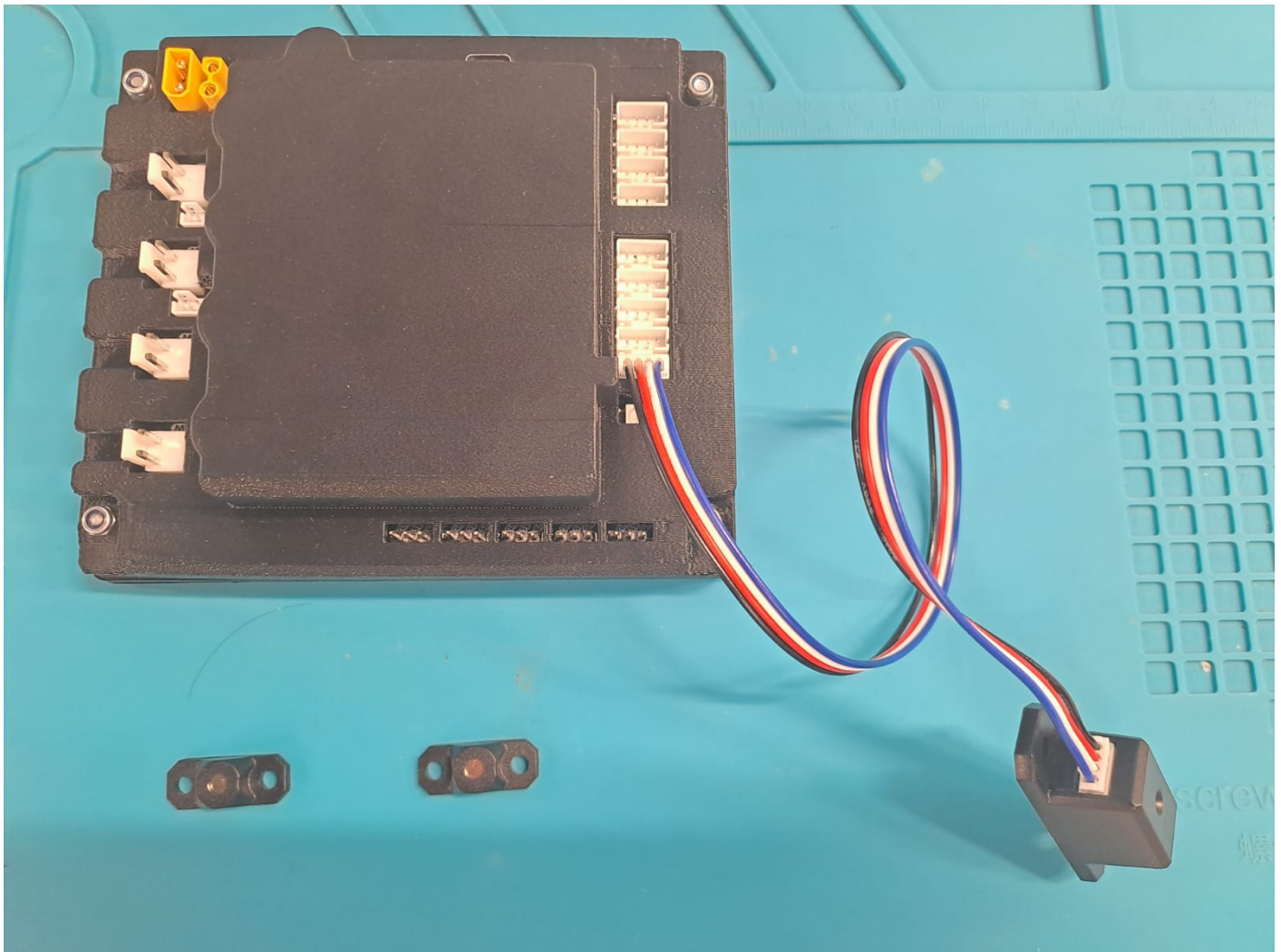
As conexões correspondentes na placa EasySTEAM estão indicadas abaixo:





Considerando uma ordem da direita para a esquerda, temos a porta Digital 1, 2, 3, 4 e 5. É importante diferenciar qual estamos usando por causa do código mostrado posteriormente

Segue abaixo uma imagem mostrando a conexão geral do sensor:



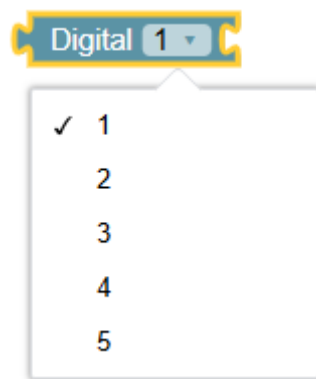
Aqui o sensor foi conectado na porta 5, mas nos exemplos seguintes estaremos usando a porta 1.

# Categoria Digital

A categoria digital é talvez a mais fácil de utilização, visto que são apenas dois blocos, um objeto e um que retorna uma expressão boolean (veja a seção [Lógica](#))

## Objeto digital

O objeto de sensor digital é semelhante aos blocos de motores e servos, no sentido de que eles são colocados na [Área de trabalho](#) e é fornecem uma porta que pode ser escolhida.



## Bloco de input

Esse bloco tem como objetivo retornar o valor que o sensor está no momento. No caso de um sensor de limite (como o usado nesse capítulo), se um objeto está dentro do alcance, ele retorna verdadeiro, se não, falso.



Esses blocos são muito utilizados juntamente da categoria "Logical", portanto, dê uma olhada no capítulo do servo para entender de que forma isso é possível.

Veremos nas páginas seguintes a forma de se trabalhar com essa categoria.

# Categoria Text

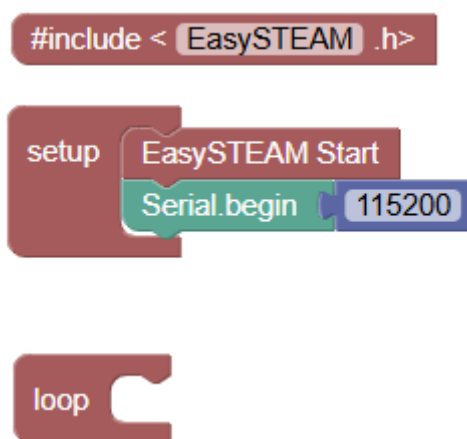
---

Geralmente, quando um código é feito, é interessante e desejável gravar a saída dele para que se possa entender o que está acontecendo - esteja funcionando (ou não).

Para fazer isso, dentro da interface foi criado uma categoria para mostrar mensagens no [monitor serial](#), que é a categoria desta página.

## Inicializando o monitor serial

Apesar de o monitor serial ser parte da interface, é preciso informar a placa EasySTEAM que é desejado abrir uma comunicação com ela. Isso é feito da forma demonstrada abaixo:



Utilizamos para iniciar a comunicação dois blocos, os quais são encontrados na categoria "Math" e "Text", em resumo, o número 115200 indica que o objetivo é abrir uma comunicação com a placa. Entender o que exatamente faz o número 115200 é um conteúdo mais difícil, que não será abordado nesse documento, mas uma explicação pode ser obtida no seguinte documento:

[Comunicação Serial](#)

## Mostrando mensagens

Para imprimir mensagens no monitor serial utilizamos algum dos seguintes blocos:

`Serial.print`

`Serial.println`

Esses blocos aceitam como argumento a maioria dos tipos que a interface possui.

Na página seguinte mostraremos como utilizar essa categoria juntamente da "Digital".



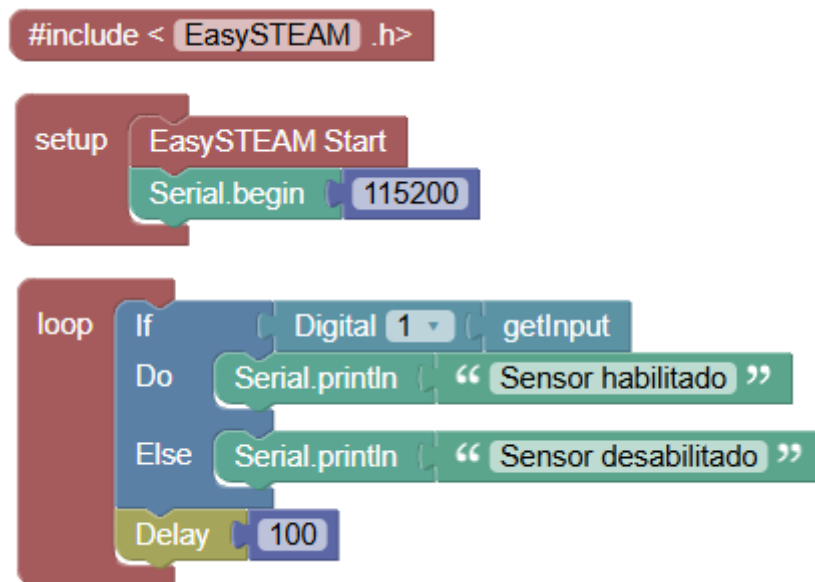
# Exercício

Leia a seção [Lógica](#) antes fazer esse exercício.

A ideia principal desta página é executar um código que faça o seguinte:

- Receba o estado do sensor de limite magnético;
- Imprima no monitor serial se estiver pressionado ou solto.

O seguinte código pode ser feito para testar o sensor:



Adicionei o delay apenas para ficar mais legível, em uma montagem que não usa monitor serial não é necessário utilizar esse função, a qual simplesmente paralisa o código durante uma certa quantidade de milissegundos.

Caso você tenha feito o código, foi possível perceber que ele informa que o "Sensor está habilitado" quando o ímã não chegou perto do sensor - o que era para na perspectiva do usuário ser "desabilitado". Em resumo, está invertido, e porque isso ocorre? Como supracitado ao longo de todo documento, condicionais trabalham com verdadeiro e falso, e do ponto de vista do sensor o falso é quando ele é acionado - isso pode ser melhor visto nos diagramas de montagem dos sensores.

Assim, a solução esperada é inverter - ou melhor, negativar - essa expressão, isso pode ser feito utilizando o operador **NOT** presente na categoria "Logical"

```
#include < EasySTEAM .h>
```

```
setup EasySTEAM Start
```

```
Serial.begin 115200
```

```
loop
```

```
If NOT Digital 1 getInput
```

```
Do Serial.println " Sensor habilitado "
```

```
Else Serial.println " Sensor desabilitado "
```

```
Delay 100
```

Resultado: o código acima funciona perfeitamente.