

# Conceitos iniciais

- [Código inicial](#)
- [Upload de código](#)
- [LED status](#)

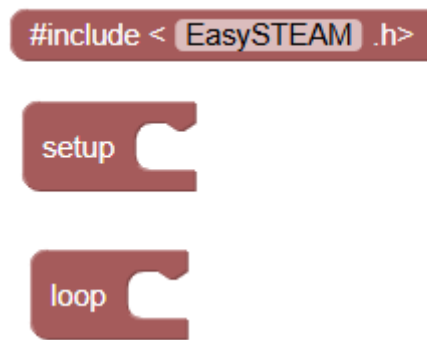
# Código inicial

---

Veja a primeira parte para entender como mexer na [interface](#) - pré-requisito para esse trecho.

Essa seção tem como objetivo mostrar qual a configuração inicial necessária para programar o controlador EasySTEAM.

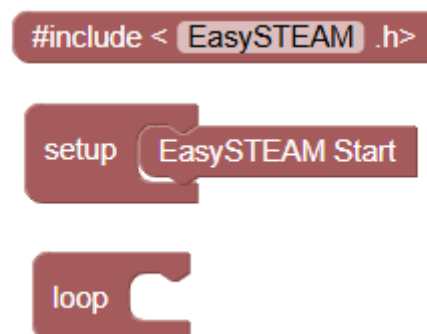
Na categoria *General* da interface (geral, traduzindo), temos os blocos que possibilitam que o controlador funcione como esperado. Todos devem ser incluídos na **seguinte ordem** para funcionar corretamente.



O bloco “*EasySTEAM Start*” vai dentro do bloco **setup**, isso pode ser feito arrastando o bloco solto para baixo da abertura dele. Como mostrado:



Dessa forma, temos a estrutura final como:



## Bloco Setup

Esse bloco opera apenas uma vez durante a execução do programa. Portanto, é aqui que ficariam as rotinas de inicialização do seu robô - caso haja alguma. Por isso que o bloco *“EasySTEAM Start”* é posto dentro de *setup*.



## Bloco Loop

Como será demonstrado nas seções seguintes, esse bloco opera repetidamente durante a execução do programa. Por isso que as instruções sobre como os acionadores se movem é geralmente realizada aqui.

# Upload de código

A placa EasySTEAM precisar ser conectada ao computador para conseguir receber o código desenvolvido, isso é feito por meio de um cabo USB-C. Então, caso o código anteriormente mostrado fosse passado, as seguintes etapas seriam seguidas.

1. Conectar o controlador no computador;
  - a. Por meio de um USB-C;
2. Fazer o código inicial;
3. Escolher um nome;
4. Clicar em Salvar;
  - a. Quando esse botão for clicado os arquivos do código serão criados;
  - b. Juntamente com a compilação do arquivo - necessária para efetuar o *upload*.
  - c. Mensagens indicando o processo aparecerão na área monitor serial.

```
Compiling... wait a time
Sketch uses 829449 bytes (26%) of program storage space. Maximum is 3145728 bytes.
Global variables use 70240 bytes (21%) of dynamic memory, maximum is 262144 bytes.

Used library
EasySTEAM
ESP32Encoder
ArduinoJson
ServoESP32
SparkFun 9DoF IMU Breakout - ICM 20948 - Arduino Library
Wire
SPI
WiFi
```

5. Clicar em Executar;
  - a. Uma das certificações de que funcionou é a mensagem de carregamento percentual;

```
Writing at 0x00000000... (100 %)
Wrote 3072 bytes (144 compressed) at 0x00000000 in 0.1 seconds [Erasing 4096 bytes]
Hash of data verified.
Compressed 8192 bytes to 47...
Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.1 seconds [Erasing 4096 bytes]
Hash of data verified.
Compressed 836032 bytes to 528405...
Writing at 0x00010000... (3 %)
Writing at 0x0001bac7... (6 %)
Writing at 0x0002564f... (9 %)
Writing at 0x00037659... (12 %)
Writing at 0x0003cc69... (15 %)
Writing at 0x00042a0d... (18 %)
Writing at 0x00047f5f... (21 %)
Writing at 0x0004dfec... (24 %)
Writing at 0x000539e1... (27 %)
Writing at 0x000594e2... (30 %)
Writing at 0x0005ea00... (33 %)
Writing at 0x00063cfc... (36 %)
```

Apesar de ser um exemplo, experimente realizar esse processo para verificar se está tudo em ordem.

# LED status

---

A presente página tem como objetivo explicar os significados do LED embarcado na placa EasySTEAM.

Segue abaixo uma tabela relacionando as cores com os significados:

Cor	Estado	Significado
	Sólido	Driver Station conectada a placa
	Piscante	EasySTEAM sem Driver Station Conectada
	Sólido	Erro