



Projeto 09

Controle Serial – Prática

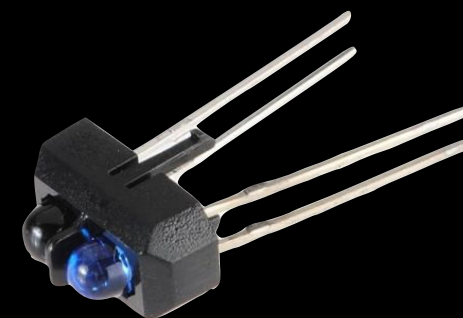
Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

Testes Iniciais



motor 3
(esquerda)



pino A11
(esquerda)

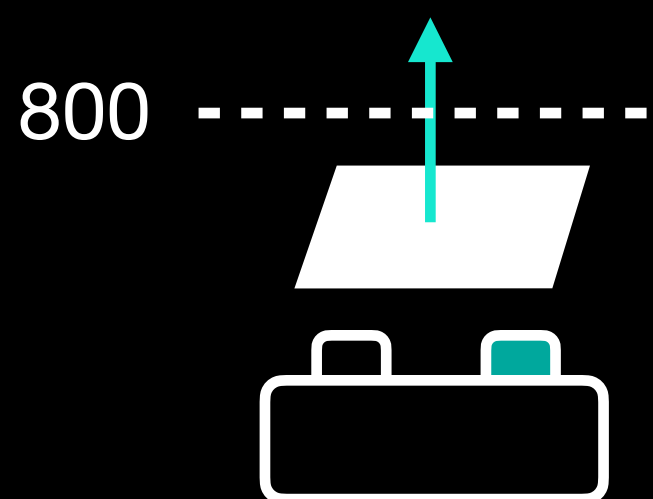


pino A12
(direita)

Pinos Usados pelos Componentes



Testes Iniciais



"contagem X"

1. Ao receber o comando "**frente N**" da Serial, gire o Motor 3 para frente com a velocidade N. Ao receber "**tras N**", gire para trás com a velocidade N.

↳ DICA: use a função `startsWith` e `substring`.

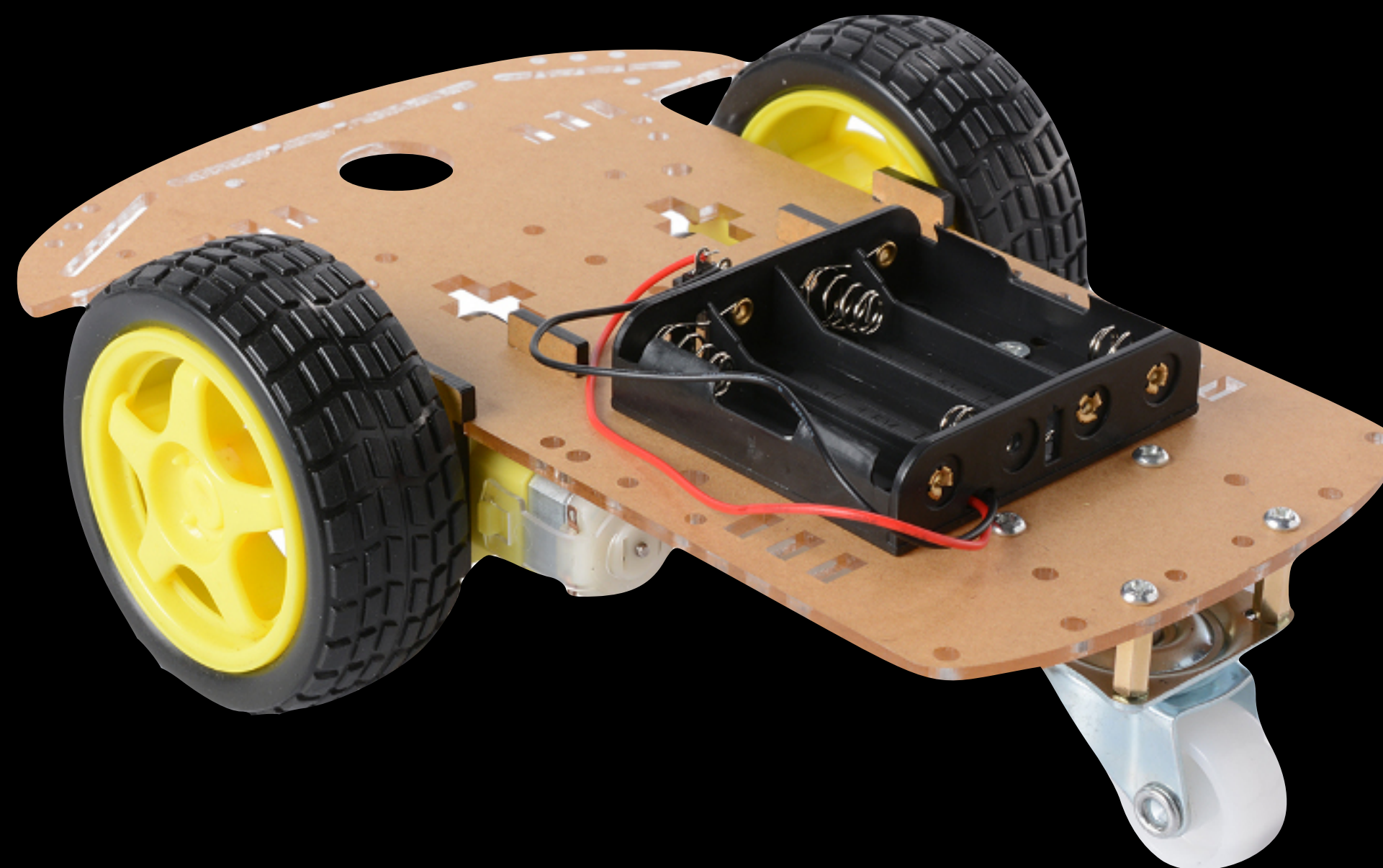
2. A cada 500 milissegundos, **imprima na Serial as leituras analógicas dos dois sensores óticos** numa mesma linha, separadas por vírgula (ex: "529, 98").

↳ DICA: use a `millis`, a conversão para `String()` e a concatenação.

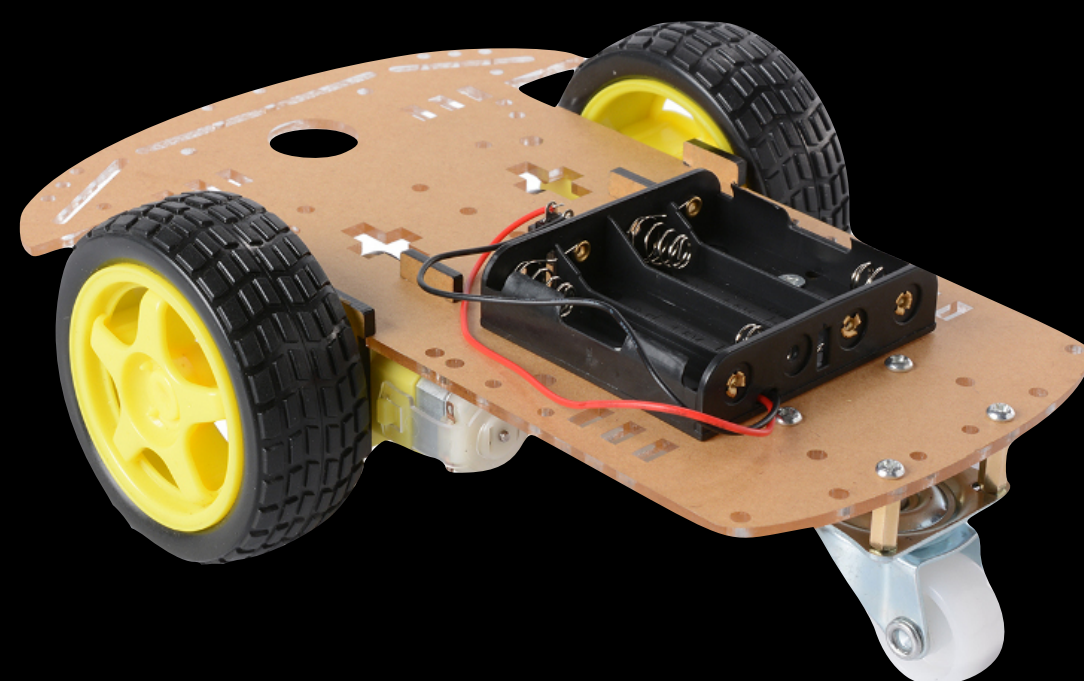
3. Aumente 1 vez um contador X **cada vez que o valor analógico do sensor ótico 2 ultrapassar o limiar de 800** (ou seja, quando o valor passar de algo menor que 800 para algo maior que 800). Em seguida, envie o texto "contagem X" (com o valor de X) pela Serial.

↳ DICA: crie uma outra variável global para armazenar o valor anterior do sensor.

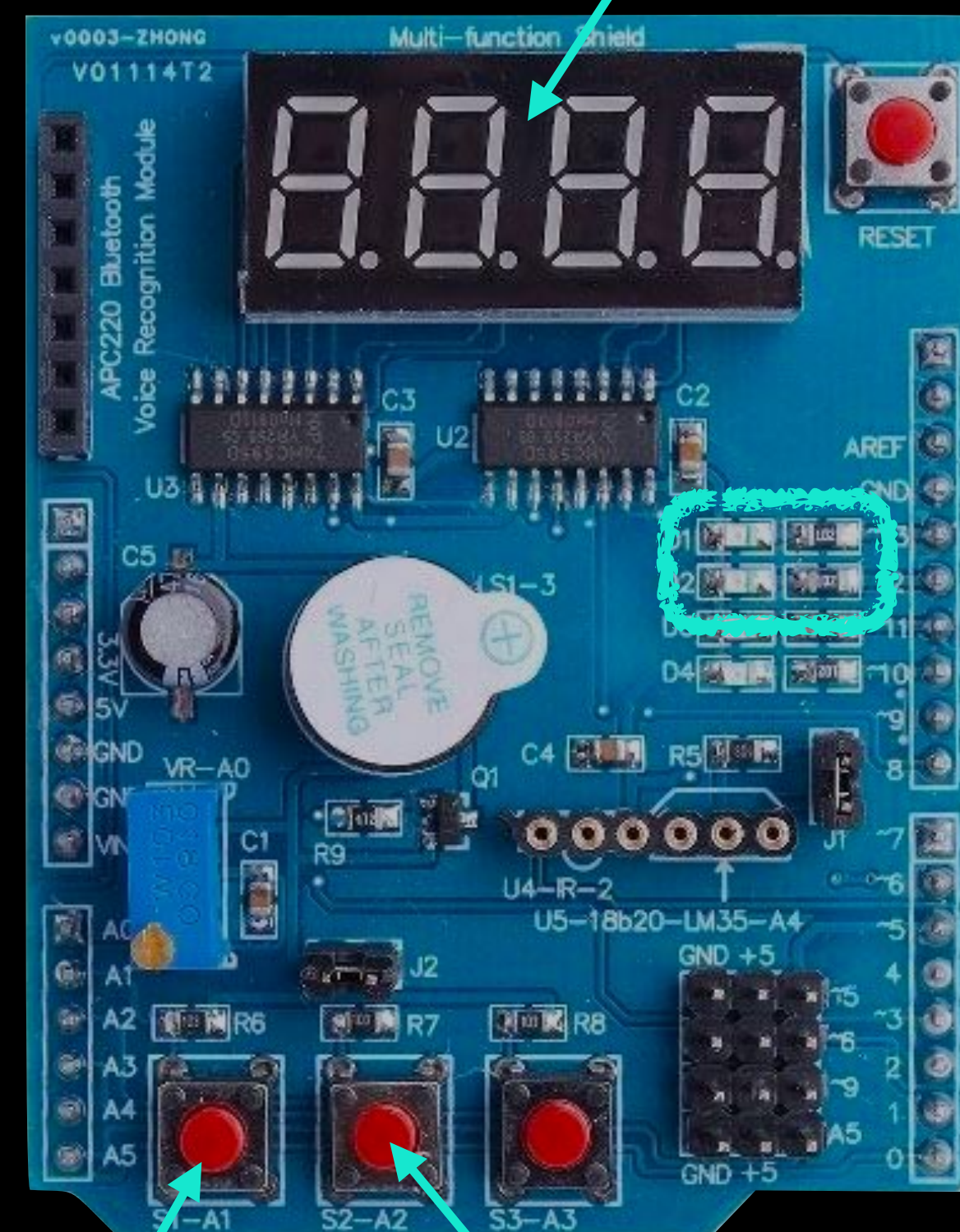
Implementação



Carrinho de Duas Rodas



serial



mostra direção

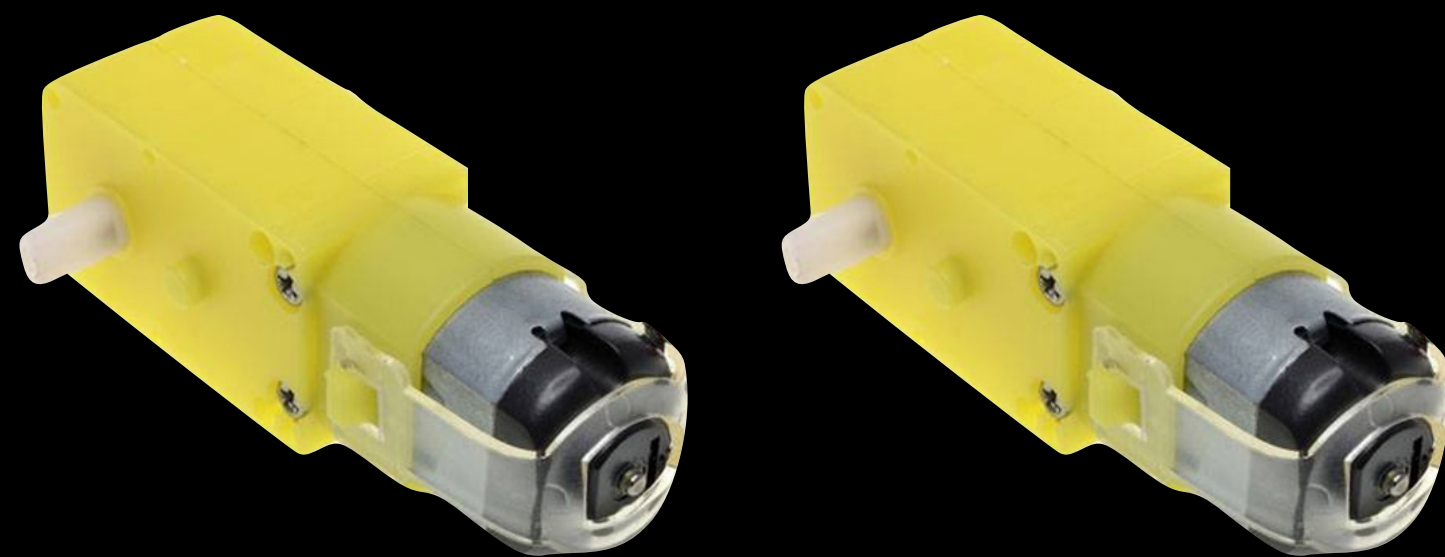
mostra leitura
dos sensores
do carrinho

muda direção

andar / parar

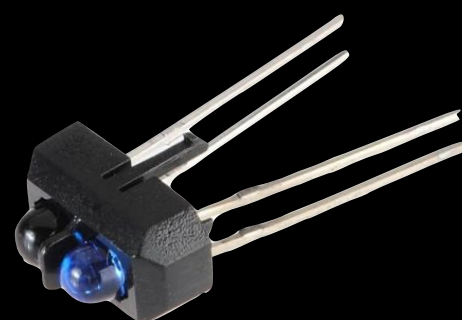
Controle Manual do Carrinho

Implementação 1

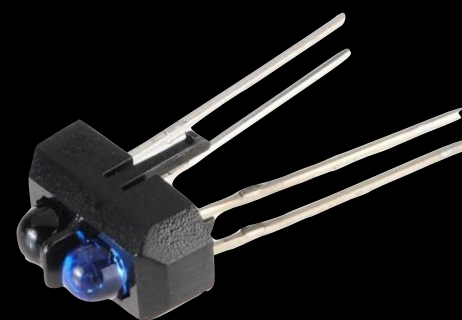


motor 3
(esquerda)

motor 4
(direita)



pino A11
(esquerda)



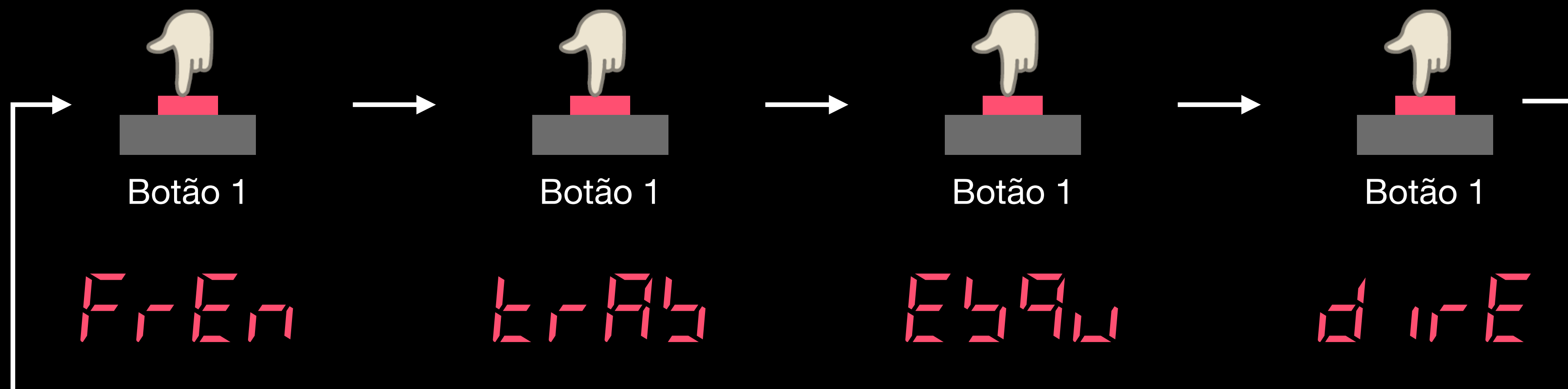
pino A12
(direita)

Implementação 2

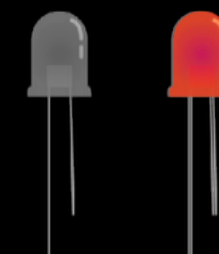
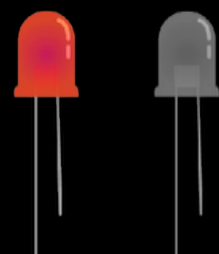
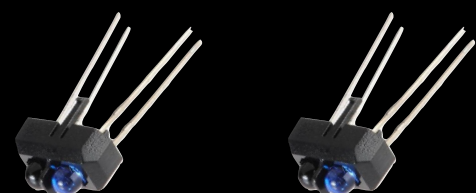
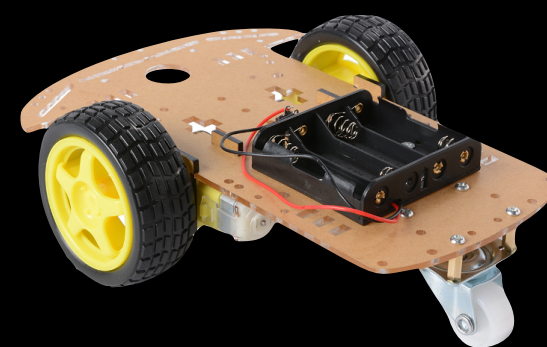
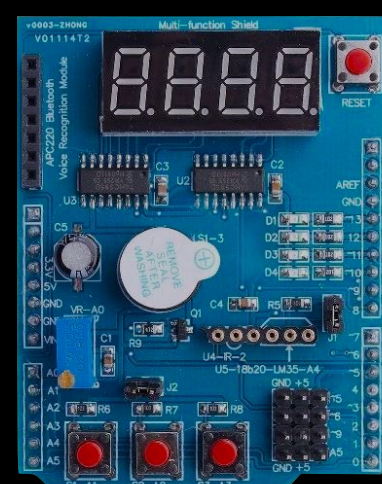
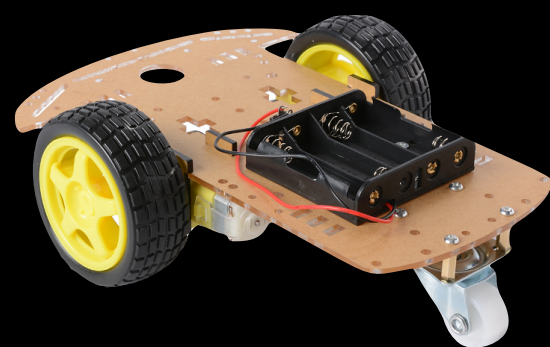
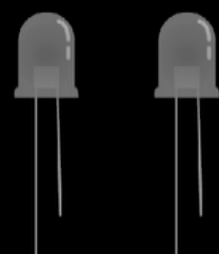
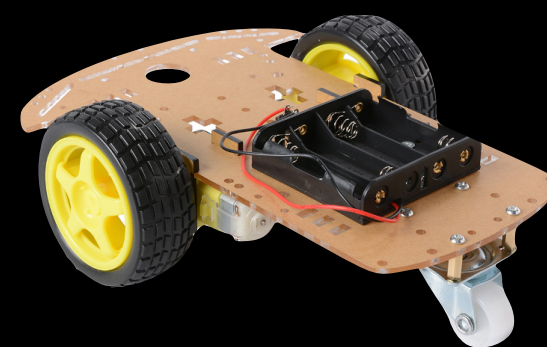
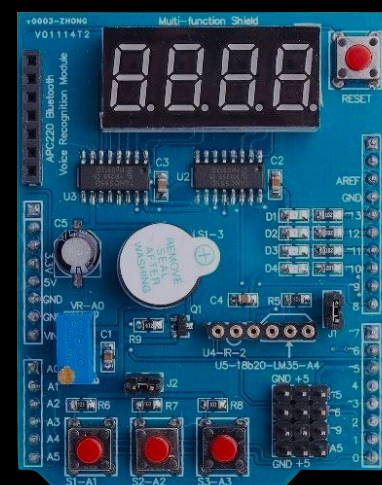
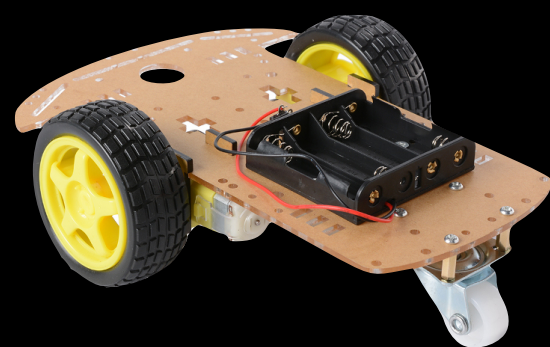


pinos A1, A2 e A3

Pinos Usados pelos Componentes

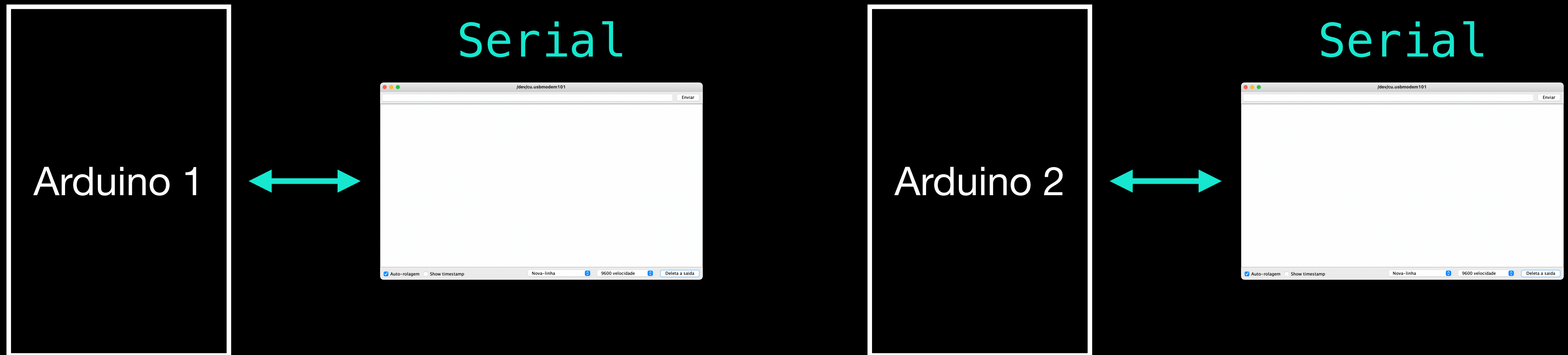


Ajuste e Envio da Direção pela Serial

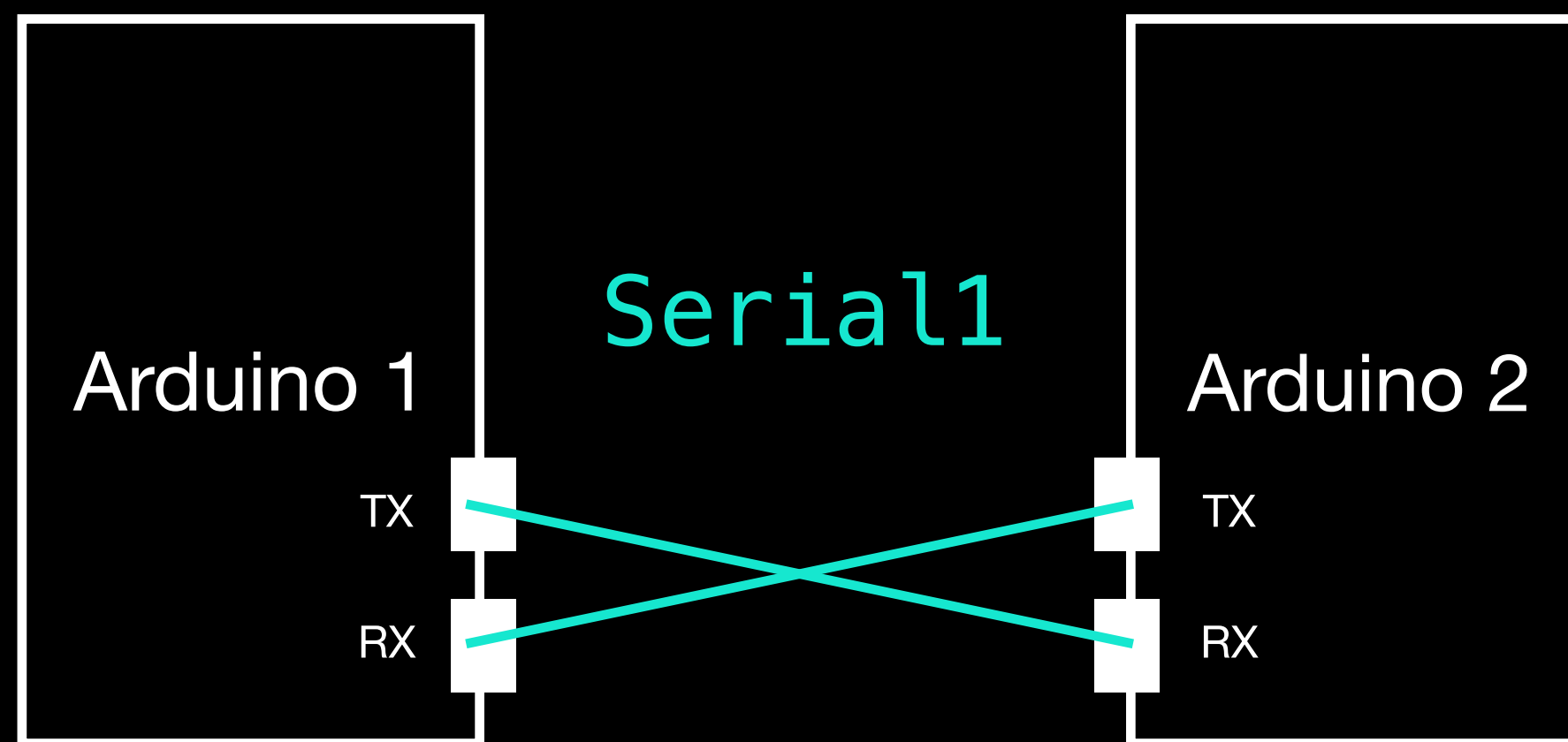


Envio da Leitura do Sensores pela Serial

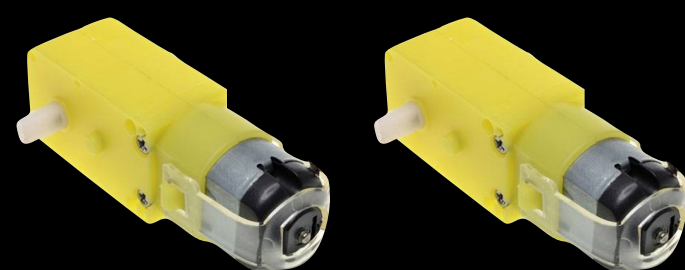
testar separadamente no começo



depois testar a comunicação entre os dois

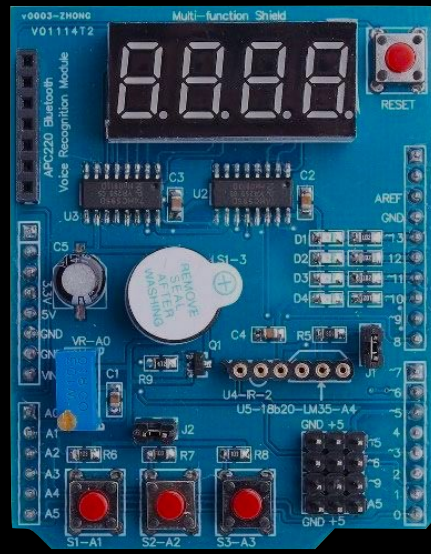


Testes com a Serial e com a Serial1



Implementação 01

1. Crie as funções **frente**, **tras**, **esquerda**, **direita** e **parar**, controlando devidamente os motores do carrinho. Fixe as velocidades como 160.
2. Chame as funções acima **ao receber da Serial** os comandos "frente", "tras", "esquerda", "direita" e "parar".
3. A cada 100 ms, **envie as leituras digitais do sensores óticos pela serial**.
↳ DICA: use a `millis`. Escolha um formato simples para enviar esses dados, para não complicar a leitura deles na Implementação 02!



Implementação 02

1. Ao apertar o Botão 1, alterne uma variável global do comando atual entre "frente", "tras", "esquerda" e "direita". Exiba essa variável no display de 7 segmentos.

2. Ao apertar o Botão 2, envie o comando atual pela serial. Ao soltar o Botão 2, envie o comando "parar".

3. Ao receber pela Serial os valores dos dois sensores, acenda/apague os LEDs 1 e 2 de acordo com o que foi lido (aceso = cor clara, apagado = cor escura). Use o mesmo formato de dados que você enviou pela serial na Implementação 01.

4. Altere o código das duas partes da Implementação para **usarem a Serial1**. Em seguida, teste a comunicação de dados entre os dois Arduinos.

↪ DICA: você pode manter também a Serial tradicional em algumas partes, para ajudar a monitorar o seu programa no PC.

Aperfeiçoamento



09b_implementacao1.ino

cópia
----->



09c_aperfeicoamento1.ino



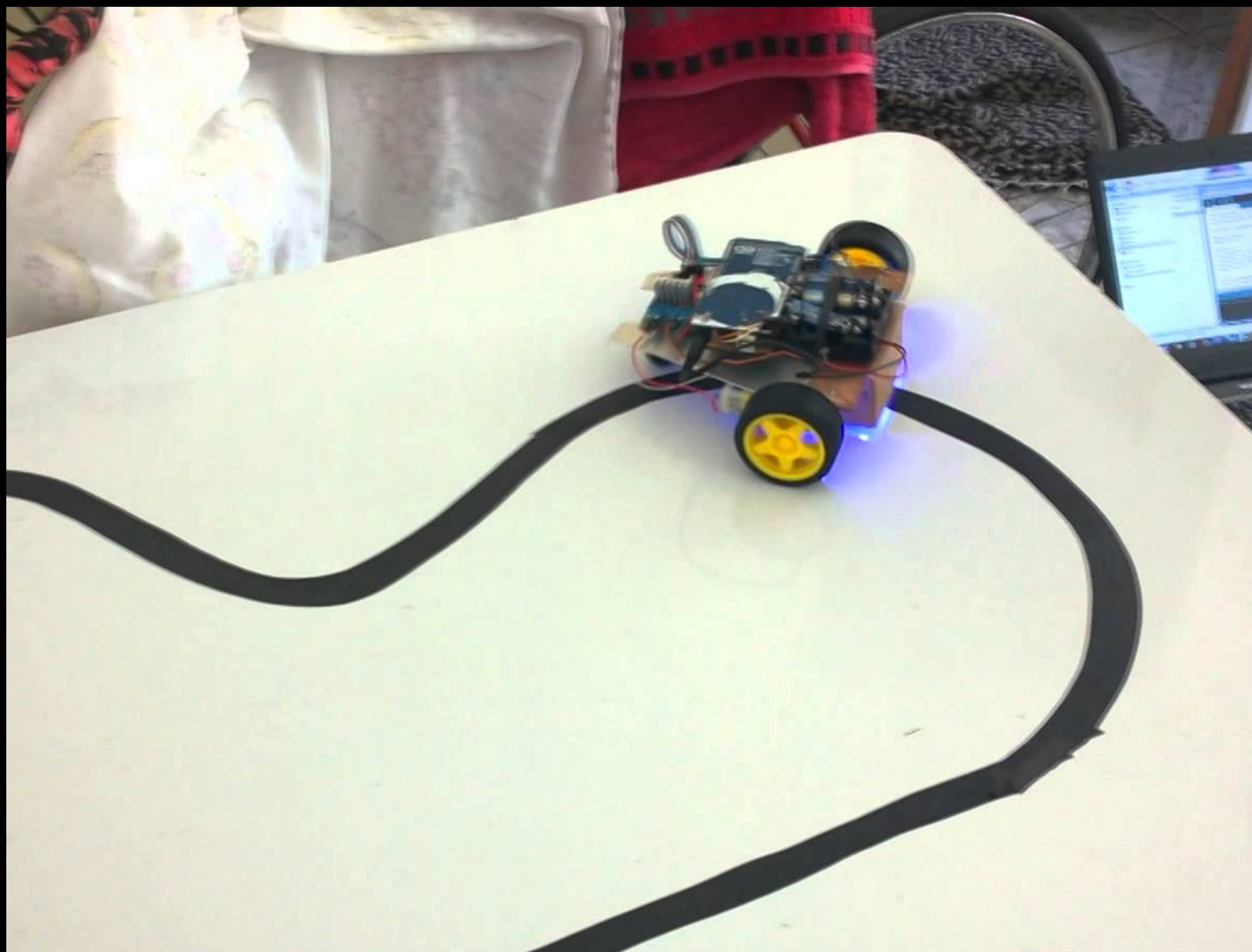
09b_implementacao2.ino

cópia
----->



09c_aperfeicoamento2.ino

Cópia do Código da Implementação para o Aperfeiçoamento



Modo Automático Seguidor de Linha

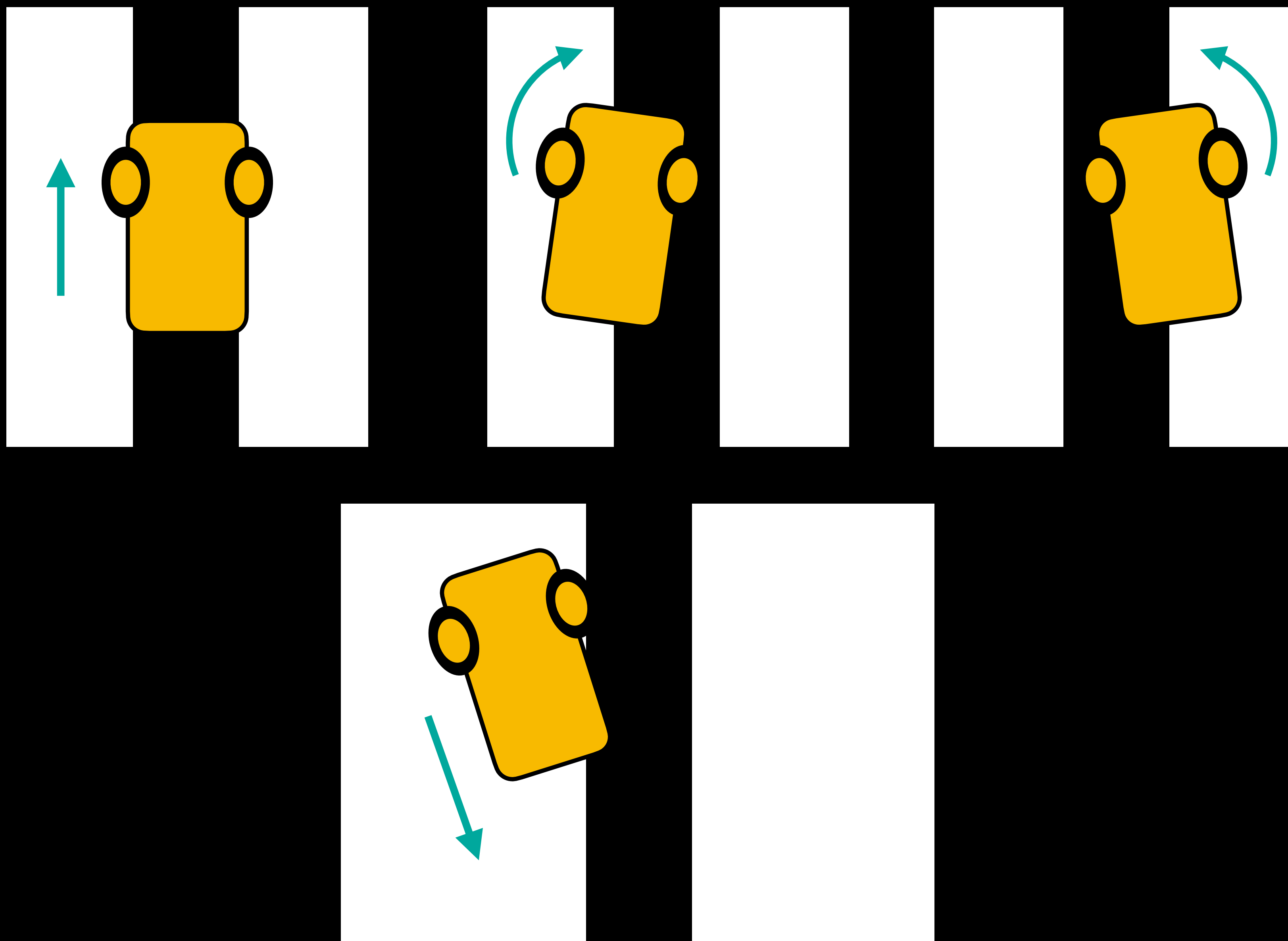
sensor 1



sensor 2

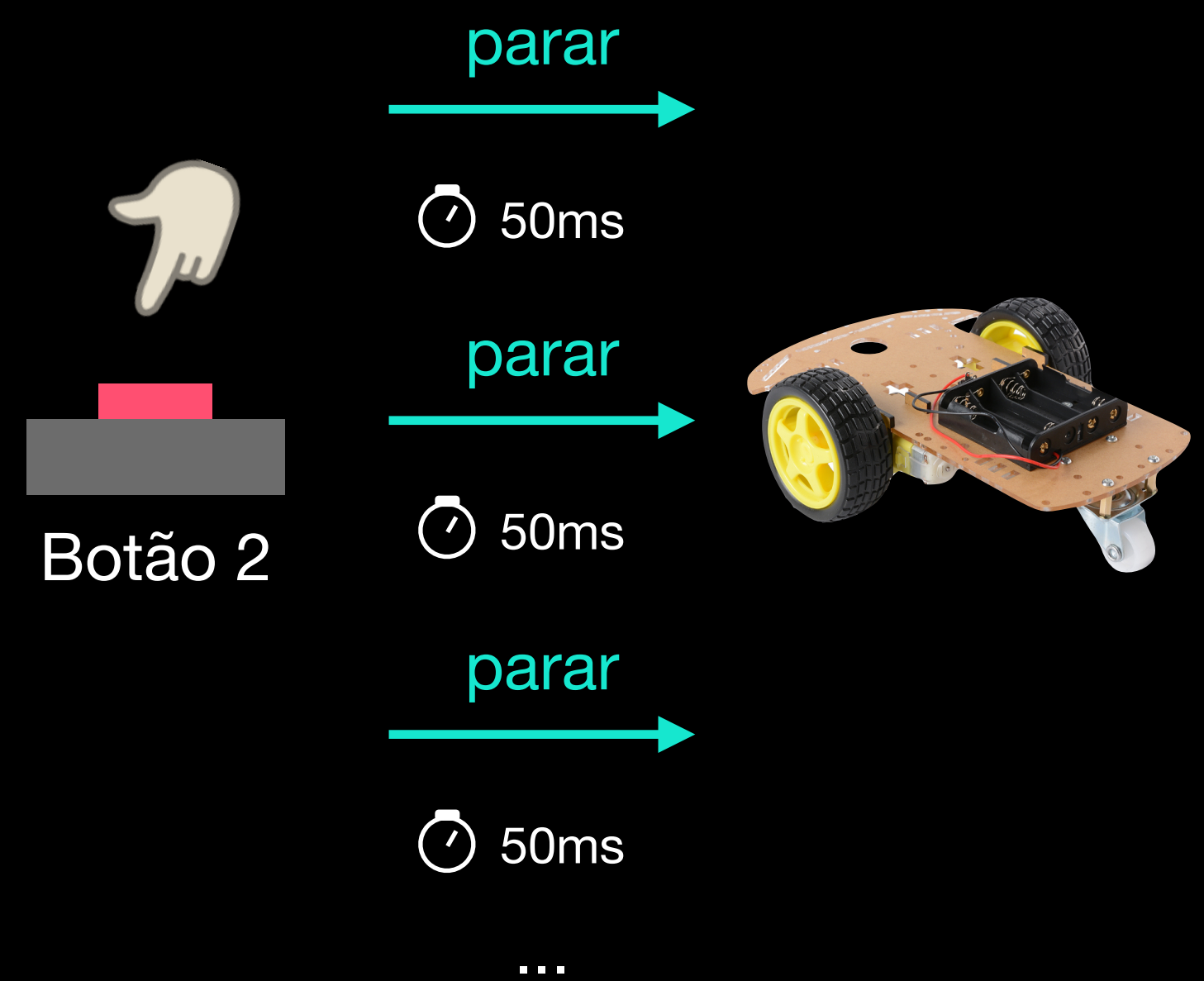


Dois Sensores Rastreando a Linha Preta

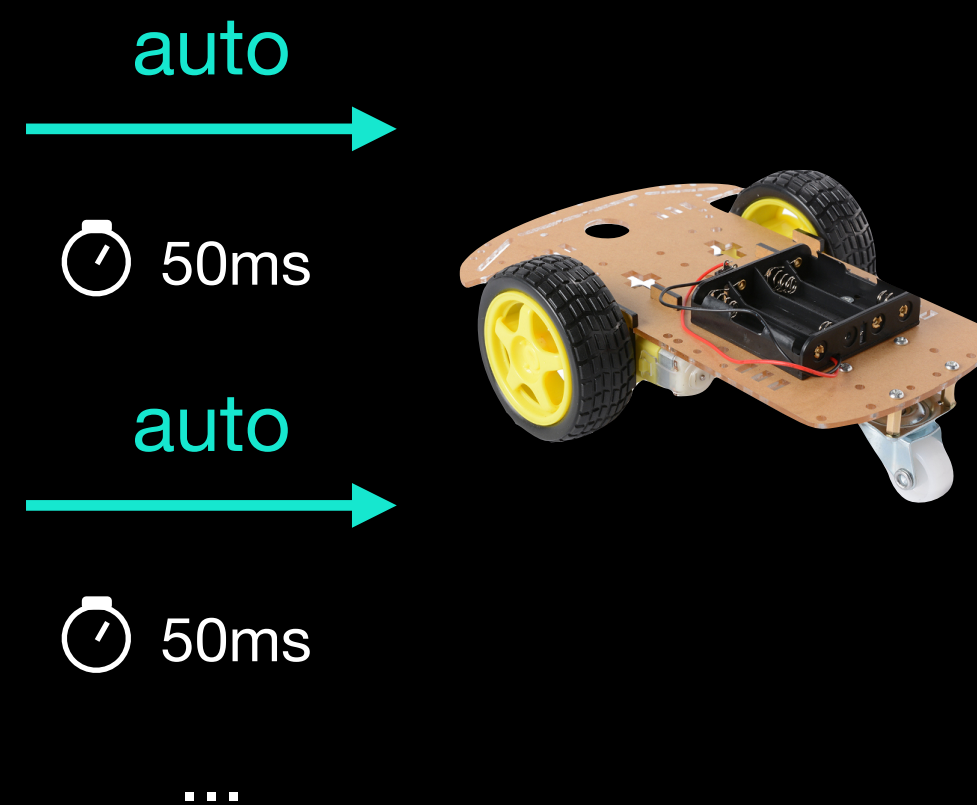


Controle do Carrinho de Acordo com o Sensor

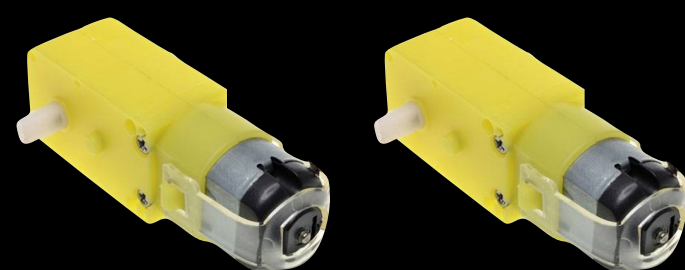
FREN



Auto

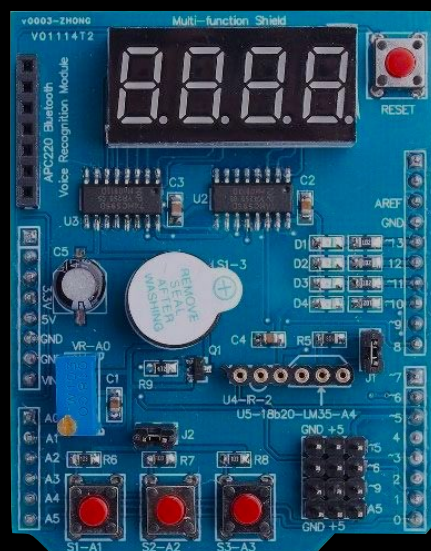


Envio Contínuo de Comandos pela Serial



Aperfeiçoamento 01

1. Mude o código da Serial1 **de volta para Serial**, para facilitar os testes separados.
2. Crie uma variável global booleana para o modo automático. Ao receber o comando "auto" da serial, mude a variável para true. Nesse modo, **controle a direção do carrinho para mantê-lo dentro da linha preta** e garanta que ele nunca fique parado.
↳ DICA: controle a direção de acordo com os valores dos dois sensores. Não complique o algoritmo.
3. Ao receber o comando "parar", **pare o carrinho e saia do modo automático**.



Aperfeiçoamento 02

1. Mude o código da Serial1 **de volta para Serial**, para facilitar os testes separados.
2. Crie uma variável global booleana para o modo automático. Ao apertar o Botão 3, **alterne entre os modos automático e manual**. No modo automático, o texto "auto" deve ser exibido no display. No manual, vale o que era antes.
3. **A cada 50 milissegundos, envie o comando atual de direção** ("frente", "tras", "esquerda", "direita", "parar" ou "auto") pela serial, para proteger o carrinho contra falhas na comunicação. Para isso, verifique o modo atual (automático ou manual) e o estado do Botão 2 (se está pressionado ou não).
↳ DICA: use a `millis`.
4. Altere o código das duas partes do Aperfeiçoamento para **usarem novamente a Serial1**. Em seguida, teste a comunicação de dados entre os dois Arduinos.

Desafio Extra



09b_aperfeicoamento1.ino

cópia
----->



09c_desafio1.ino



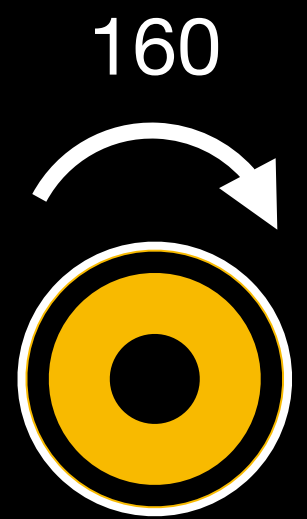
09b_aperfeicoamento2.ino

cópia
----->

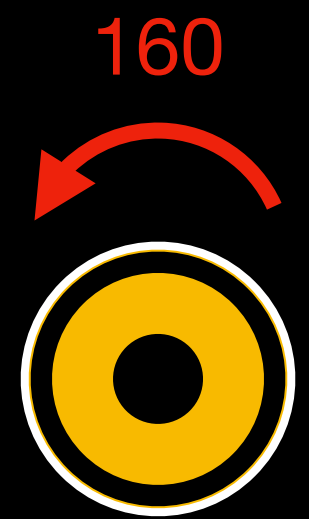


09c_desafio2.ino

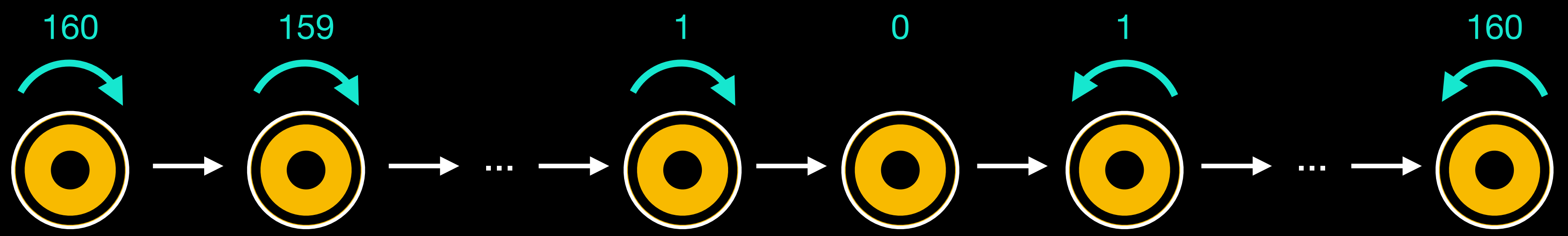
Cópia do Código do Aperfeiçoamento para o Desafio



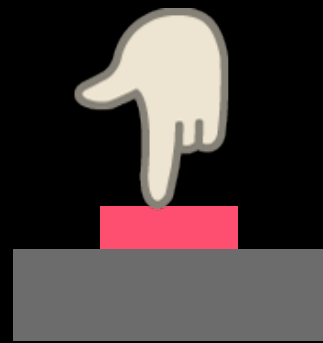
mudança brusca
de direção



Mudanças bruscas
são ruins para o motor
e a bateria.



Mudança Gradual da Velocidade



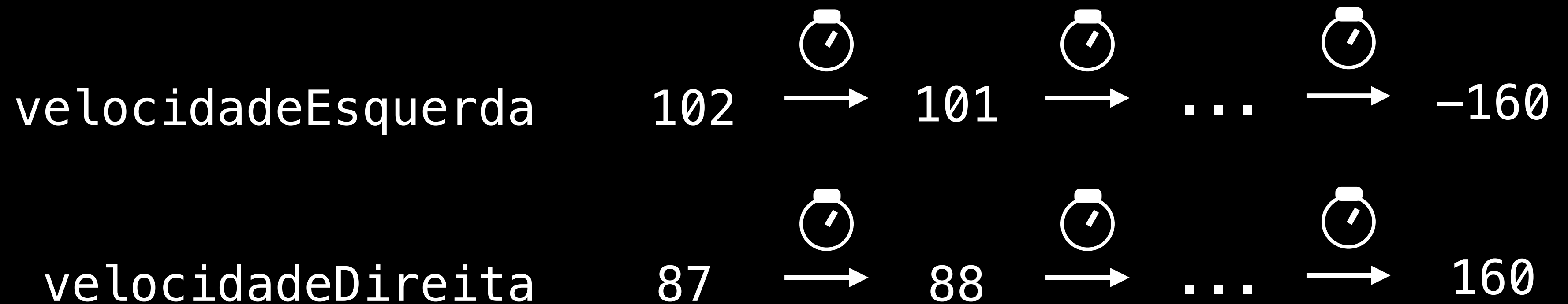
Botão 1



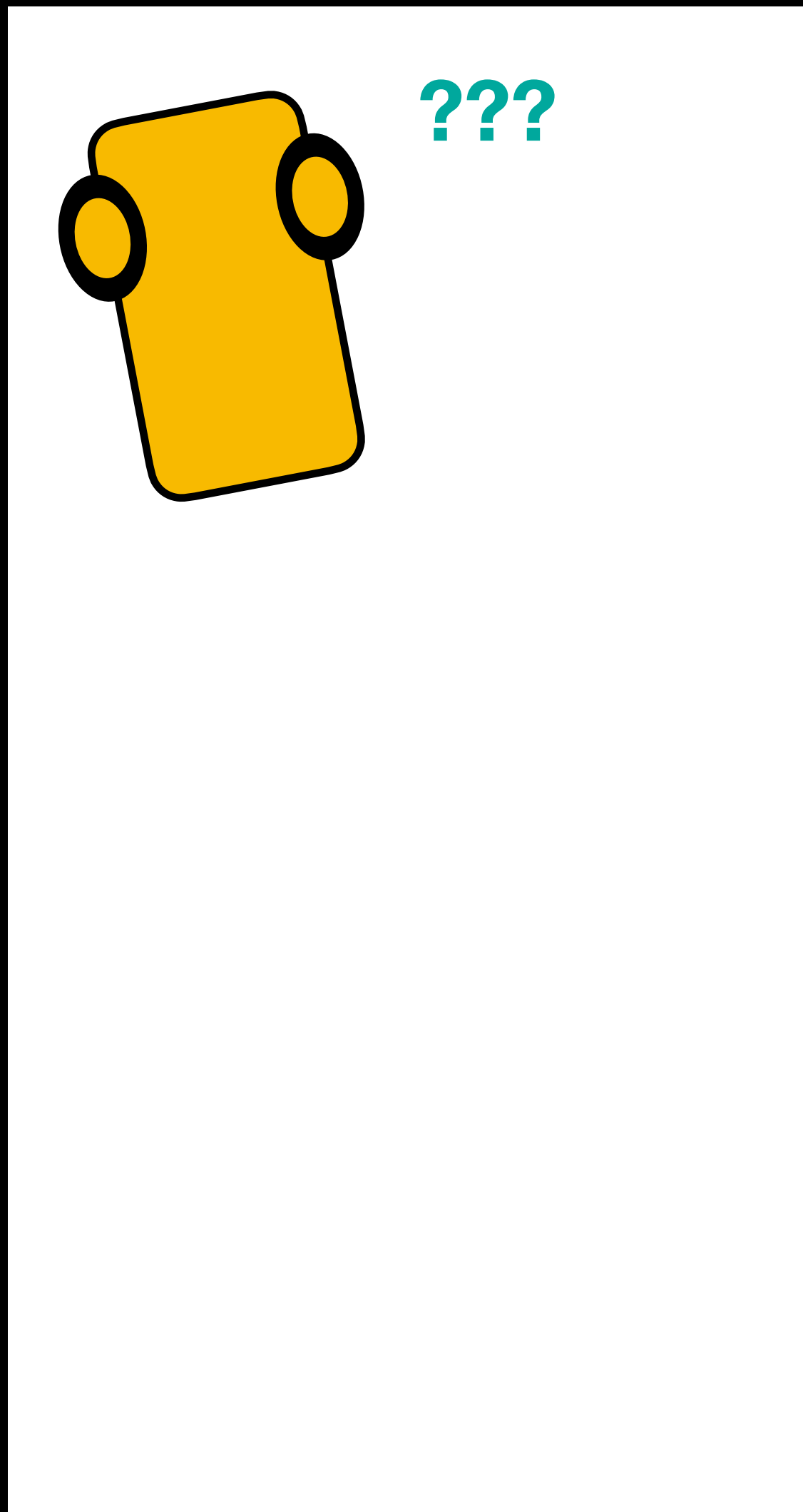
```
metaEsquerda = -160;  
metaDireita = 160;
```

ESQU

loop:



Ajuste Gradual para Meta da Velocidade



leitura do sensores



...



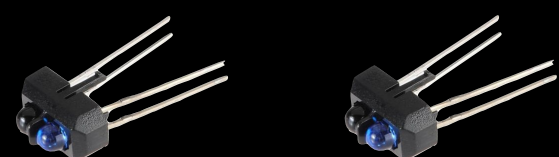
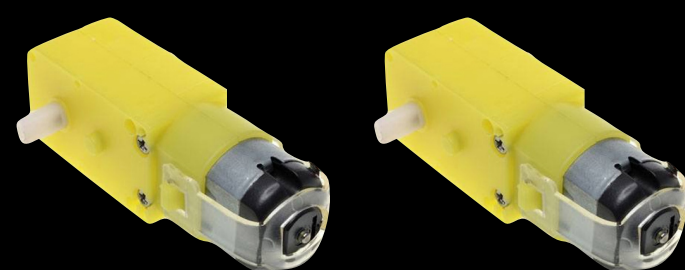
último instante
na linha preta

🕒 5 segundos



PARA O CARRINHO!

Interrupção do Modo Automático



Desafio Extra 01

1. Crie duas variáveis para registrar a meta de velocidade para o motor 1 e do motor 2. Nas funções frente, tras, esquerda, direita e parar, **ajuste essas variáveis para 160, 0 ou -160 em vez de controlar diretamente o motor.**

2. Crie outras duas variáveis para guardar as velocidades atuais dos motores 1 e 2. Dentro da loop, de tempos em tempos, verifique se essas velocidades são diferentes das metas. Se forem, **aumente ou diminua um pouco esses valores** e, em seguida, **ajuste a direção e velocidade do motor.**

↪ DICA: use a `millis` para fazer a transição gradual da velocidade.



Desafio Extra 02

1. Caso o modo atual seja o automático e os dois valores recebidos dos sensores **ficarem 5 segundos sem receber HIGH (preto)**, volte ao modo manual.
↳ DICA: salve o instante de tempo caso pelo menos um dos sensores seja HIGH, e use a `millis` para verificar se já passou o tempo limite. Lembre de reiniciar também esse instante ao apertar o Botão 3.

Possíveis Melhorias



Possíveis Melhorias

Como seria o controle automático com 5 sensores óticos ao invés de 2?

Como seria o controle automático com apenas 1 sensor?



janks.link/micro/projeto09.zip

Material do Projeto 09