



# Projeto 09

## Controle Serial – Prática

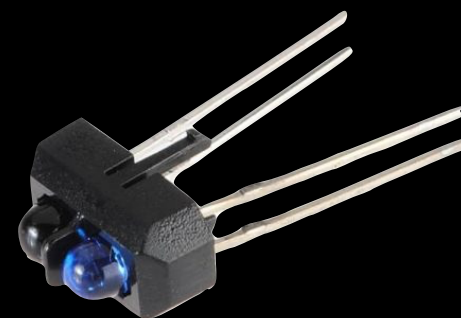
Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

ENG1419 – Programação de Microcontroladores

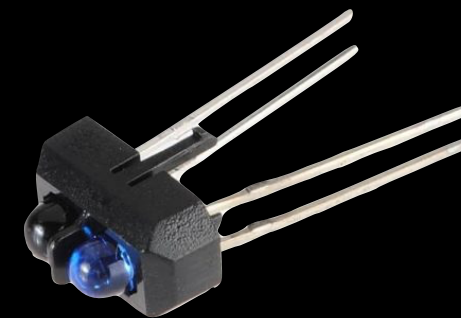
# Testes Iniciais



motor 3  
(esquerda)

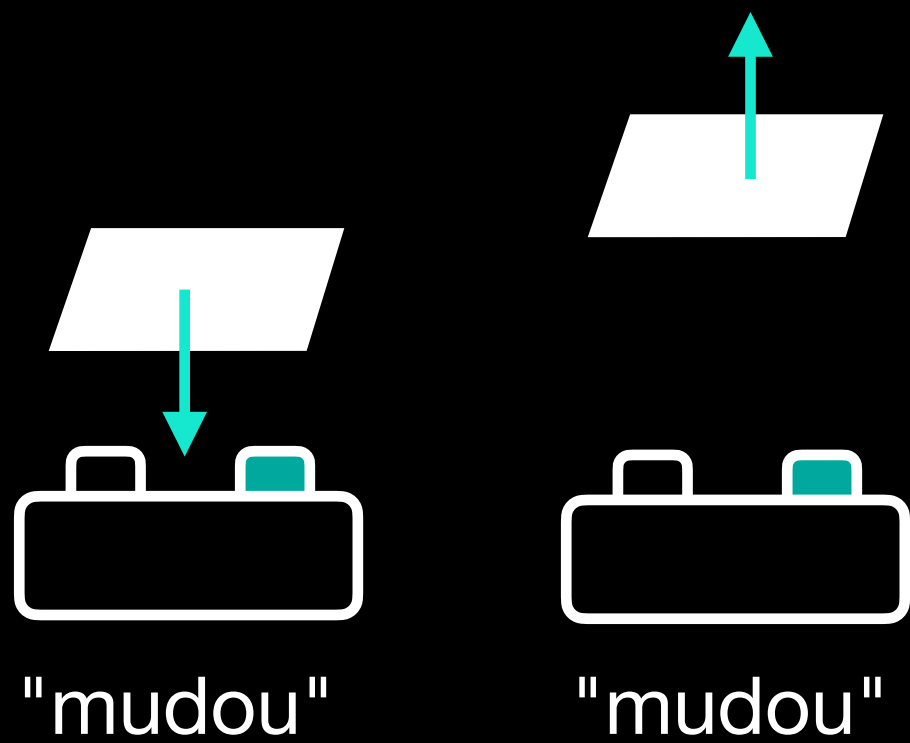


pino A11  
(esquerda)

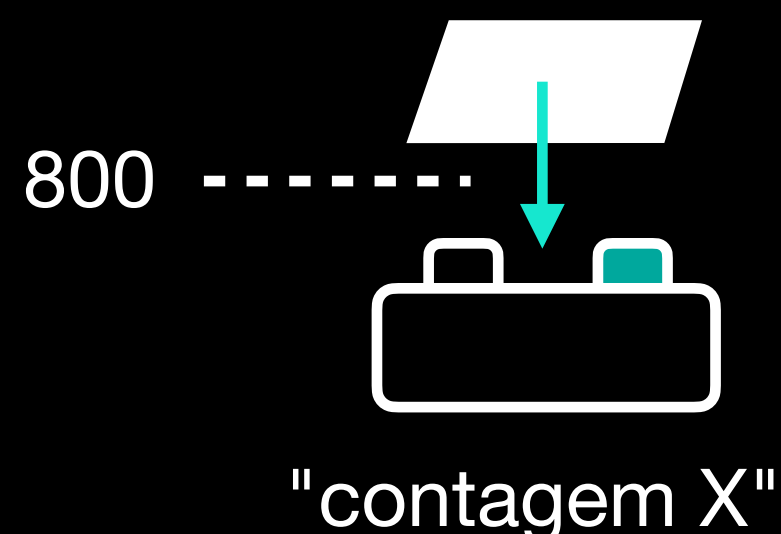


pino A12  
(direita)

Pinos Usados pelos Componentes



## Testes Iniciais



Cada vez que o **valor digital do sensor ótico 1** mudar, **envie o texto "mudou"** para a Serial.

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

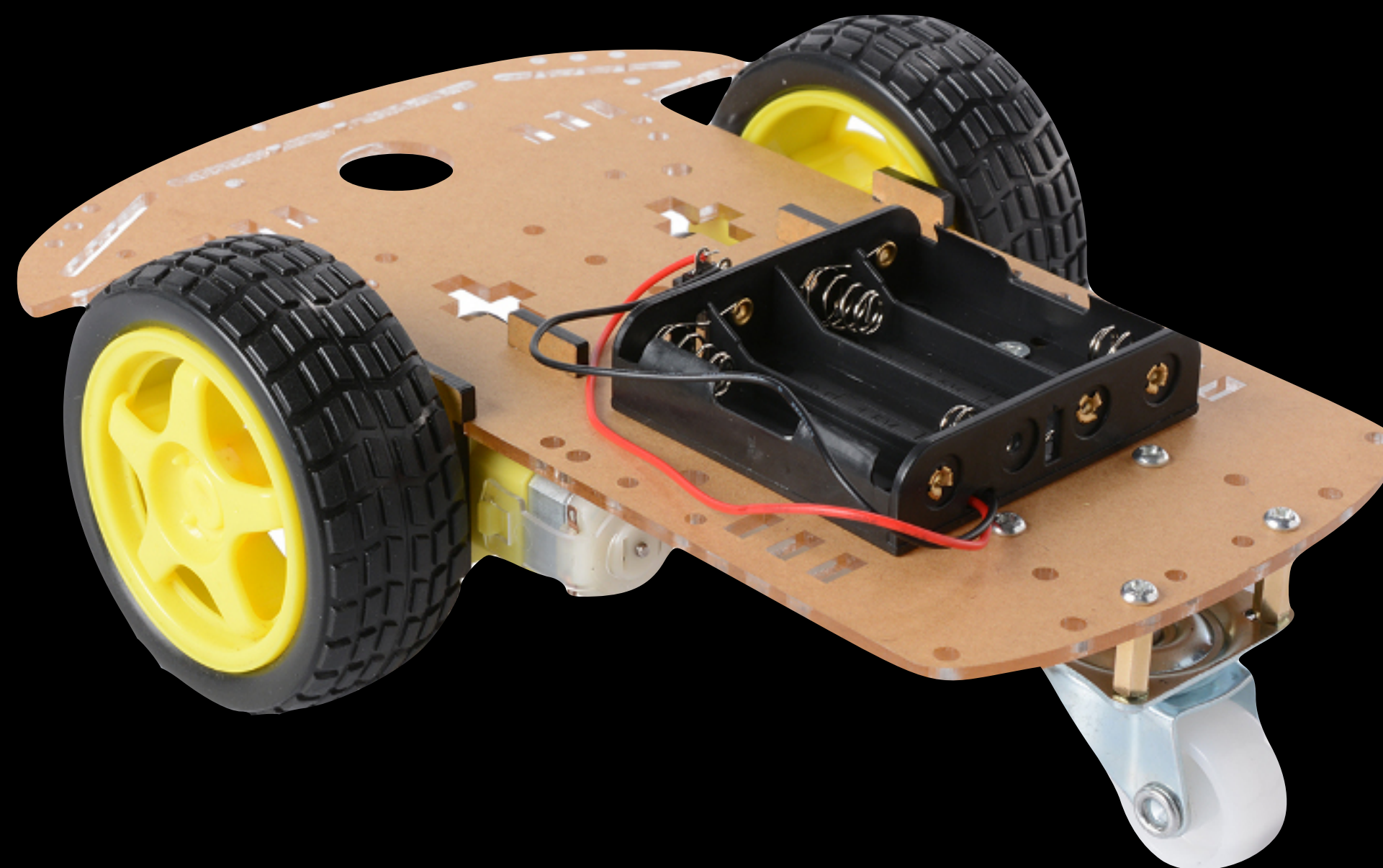
Ao receber o comando "frente N" da Serial, **gire o Motor 3 para frente com a velocidade N**. Ao receber "tras N", **gire para trás com a velocidade N**.

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

**Aumente uma vez o contador X** cada vez que o **valor analógico do sensor ótico 2 ultrapassar o limiar de 800**. Em seguida, **envie o texto "contagem X"** (com o valor de X) pela Serial.

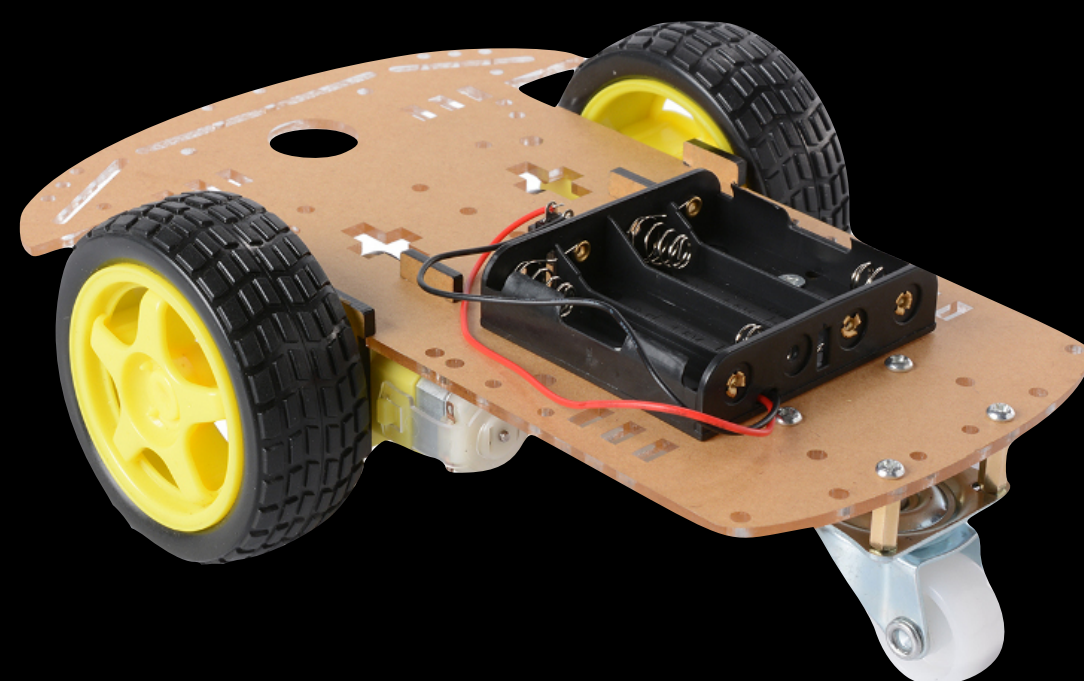
↳ DICA: crie uma outra variável global para armazenar o valor anterior do sensor.

# Implementação

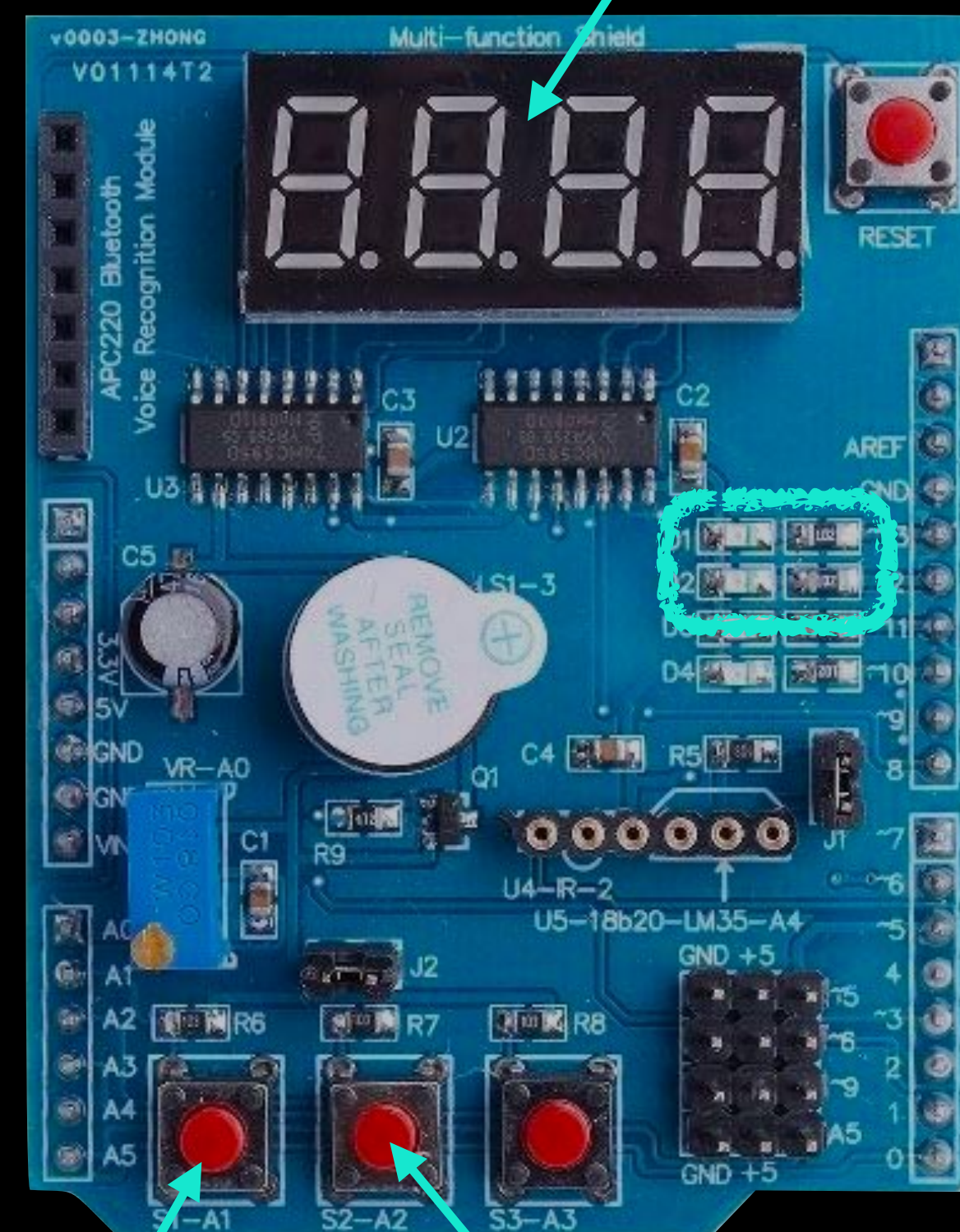


Carrinho de Duas Rodas





serial



mostra direção

mostra leitura  
dos sensores  
do carrinho

muda direção

andar / parar

Controle Manual do Carrinho



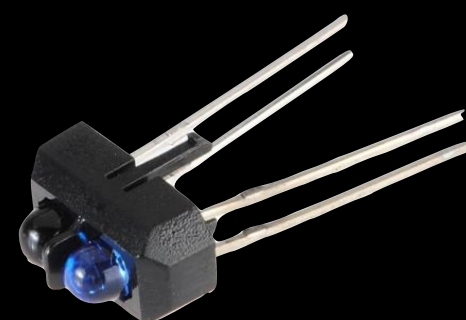
## Implementação 1



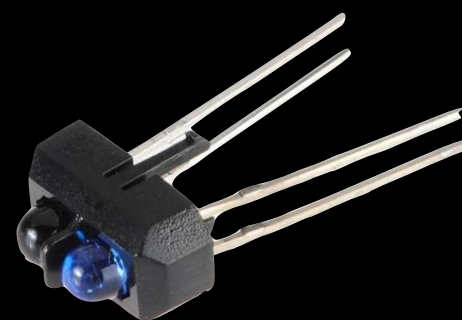
motor 3  
(esquerda)



motor 4  
(direita)



pino A11  
(esquerda)



pino A12  
(direita)

## Implementação 2

pinos 4, 7 e 8

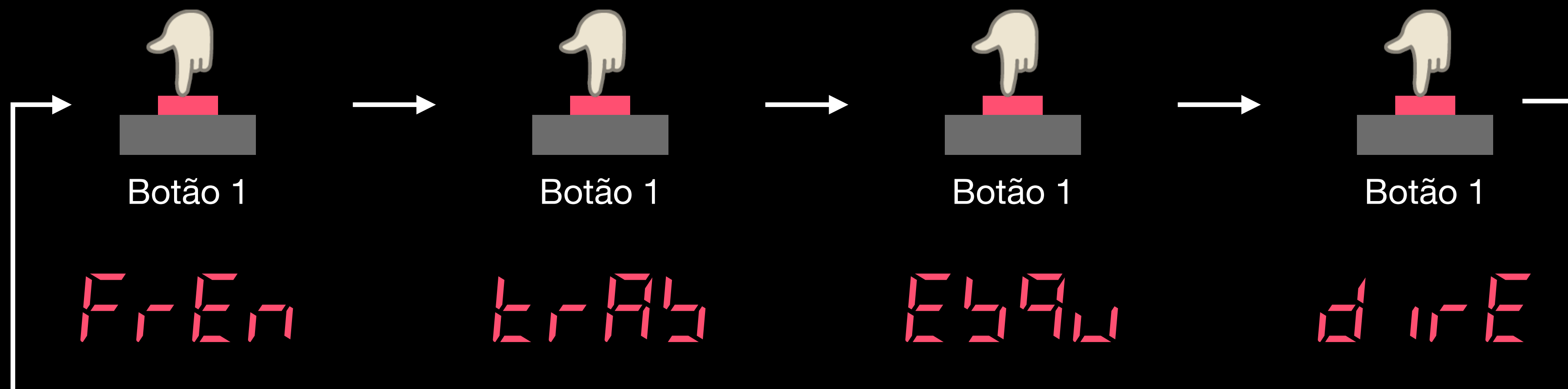


pinos 13,  
12, 11 e 10

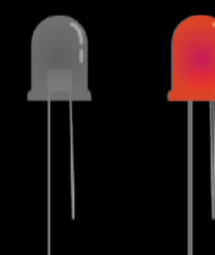
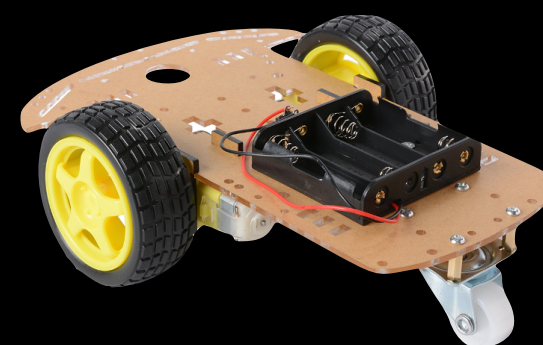
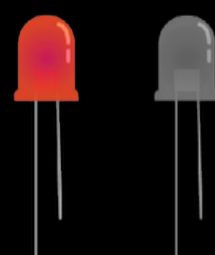
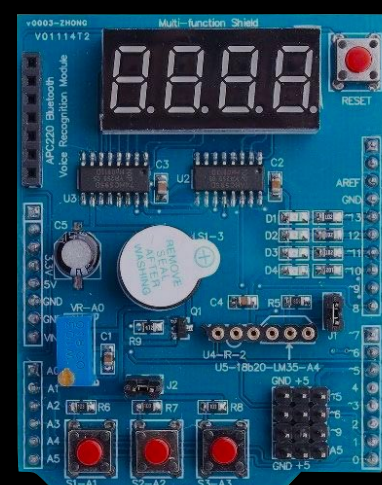
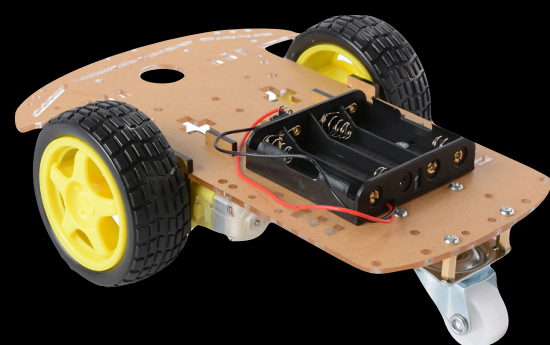
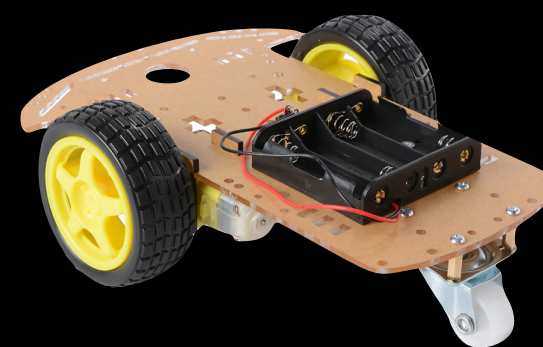
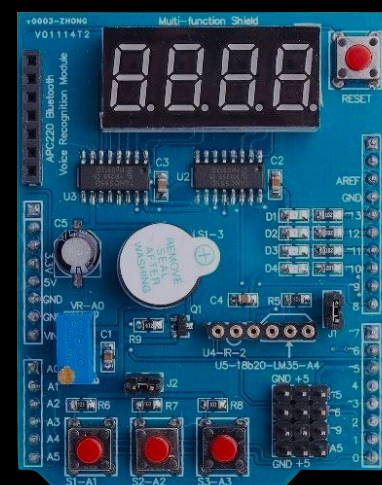
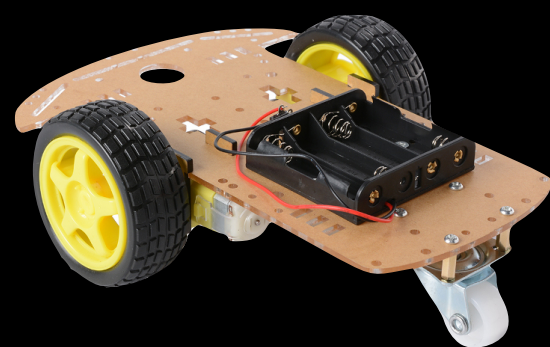
pinos A1, A2 e A3

Pinos Usados pelos Componentes





Ajuste e Envio da Direção pela Serial



Envio da Leitura do Sensores pela Serial



## Implementação 01

Crie as funções `frente`, `tras`, `esquerda`, `direita` e `parar`, controlando devidamente os motores do carrinho. Fixe as velocidades como 160.

Chame as funções acima ao receber os comandos `"frente"`, `"trás"`, `"esquerda"`, `"direita"` e `"parar"` via Serial.

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.



## Implementação 02

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

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

Ao receber 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 enviados pela serial na Implementação 01.

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



Aperfeiçoamento



09b\_implementacao1.ino

cópia  
----->



09c\_aperfeicoamento1.ino



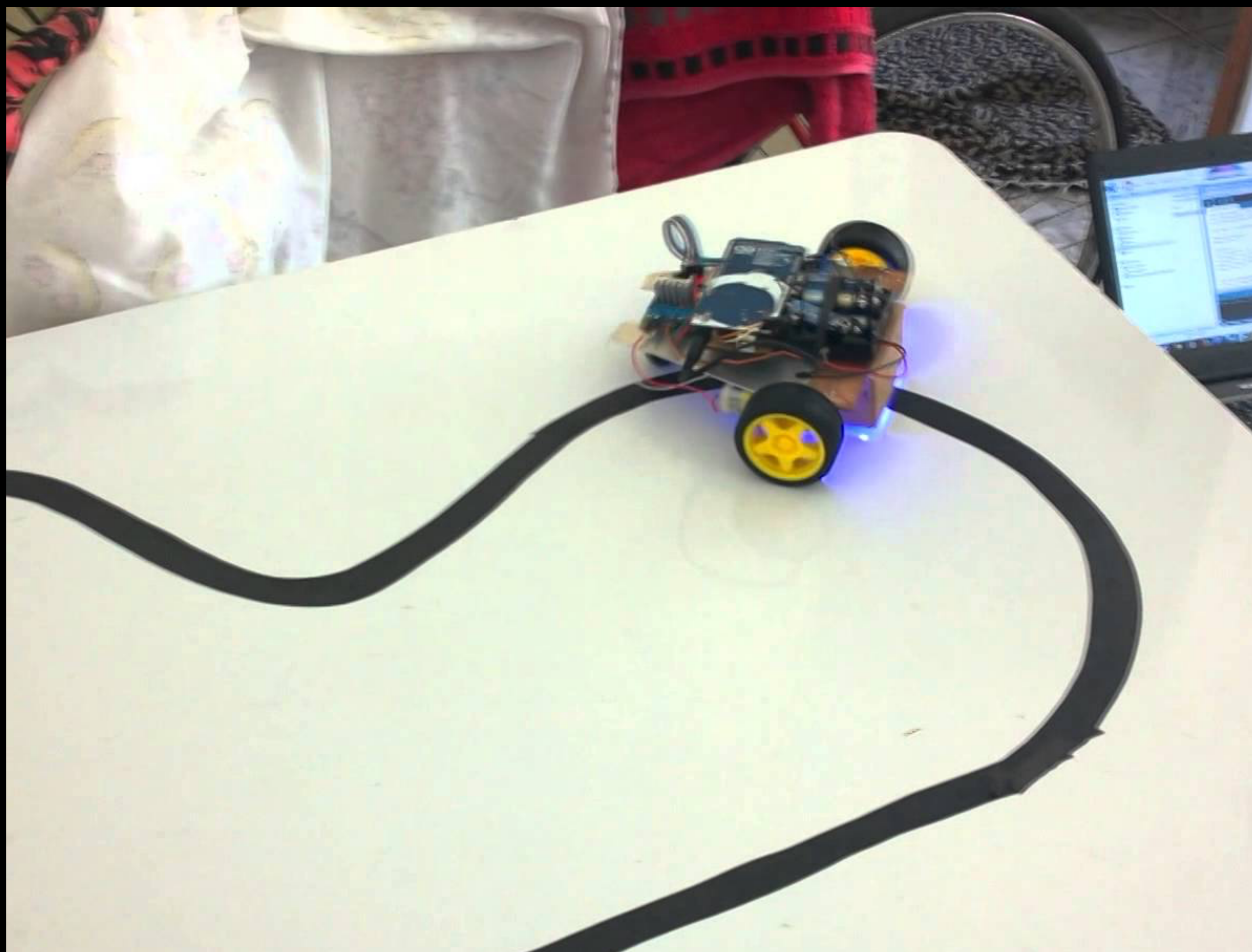
09b\_implementacao2.ino

cópia  
----->



09c\_aperfeicoamento2.ino

Cópia do Código da Implementação para o Aperfeiçoamento



Modo Automático Seguidor de Linha

sensor 1

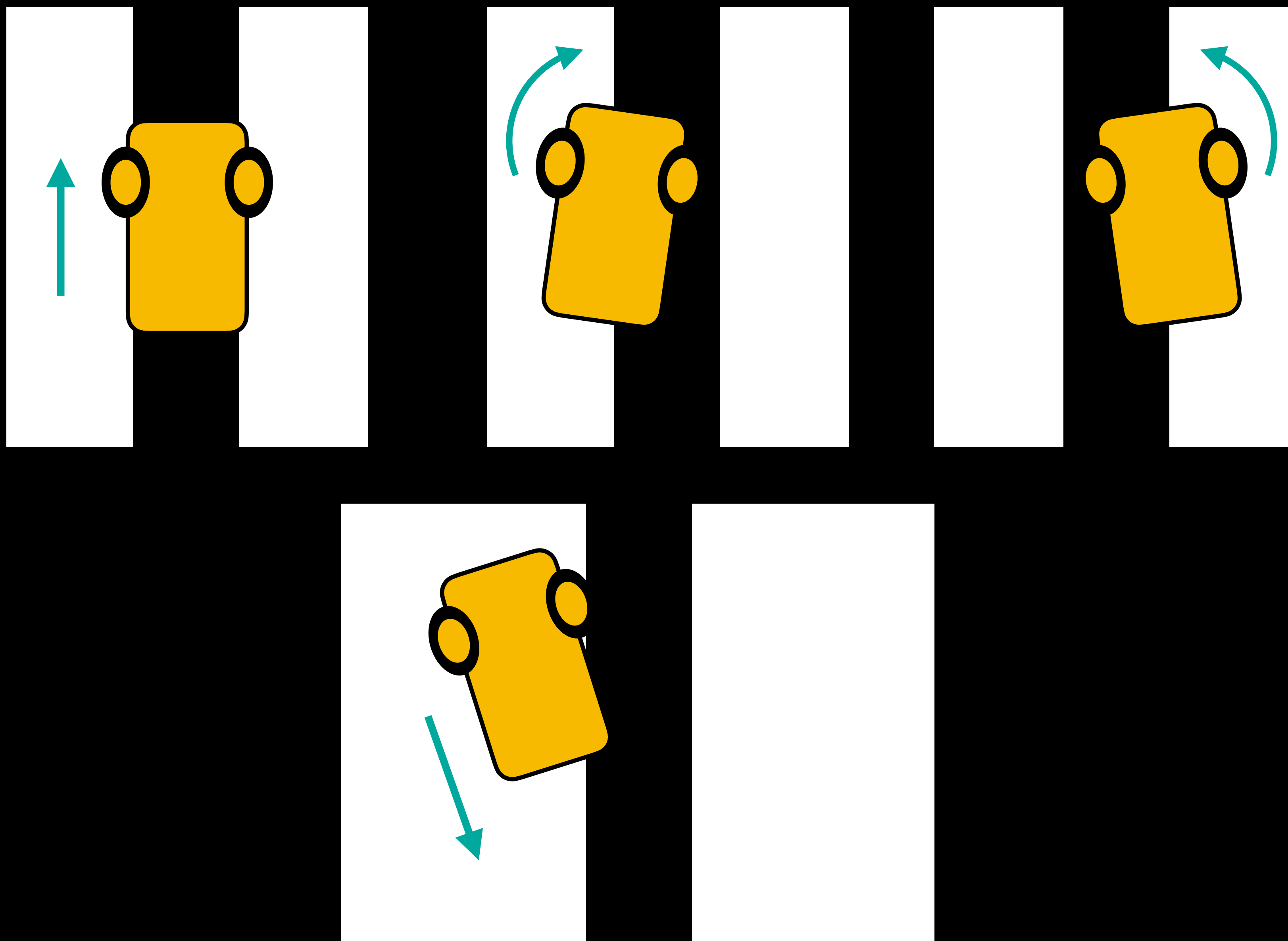


sensor 2



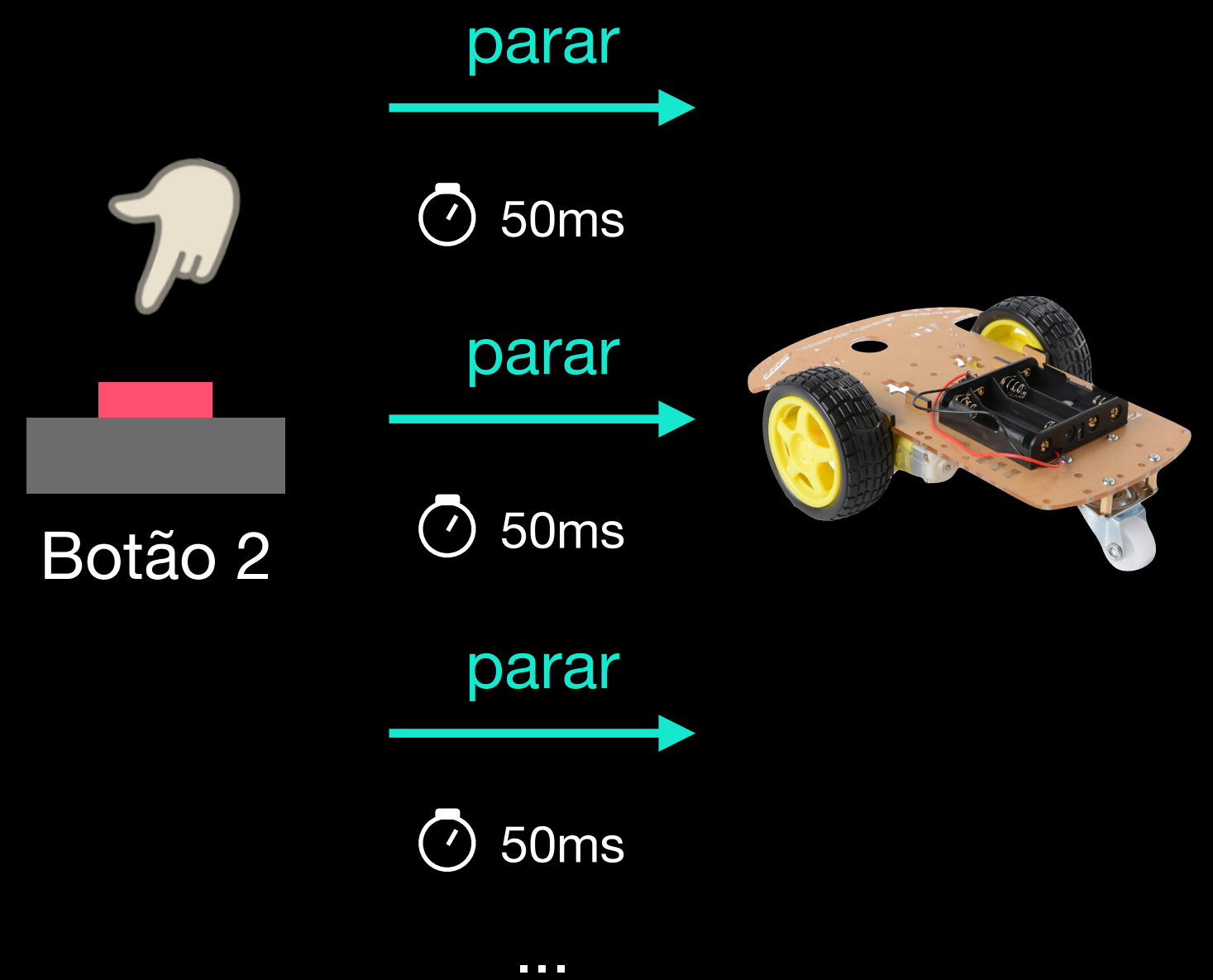
Dois Sensores Rastreando a Linha Preta



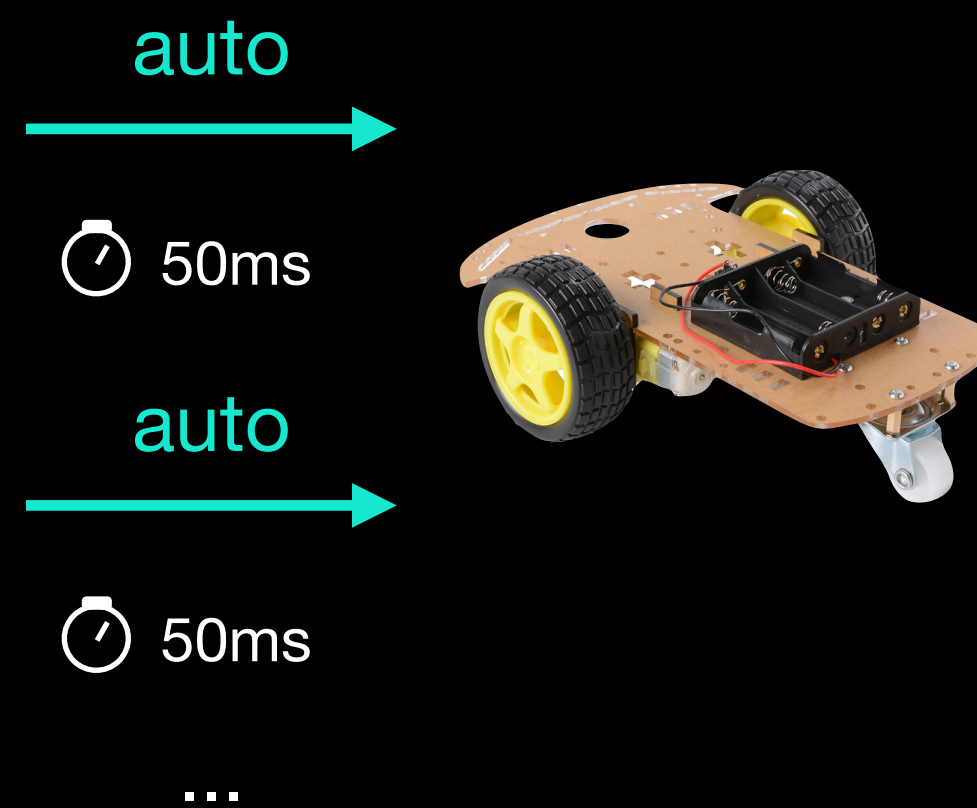


Controle do Carrinho de Acordo com o Sensor

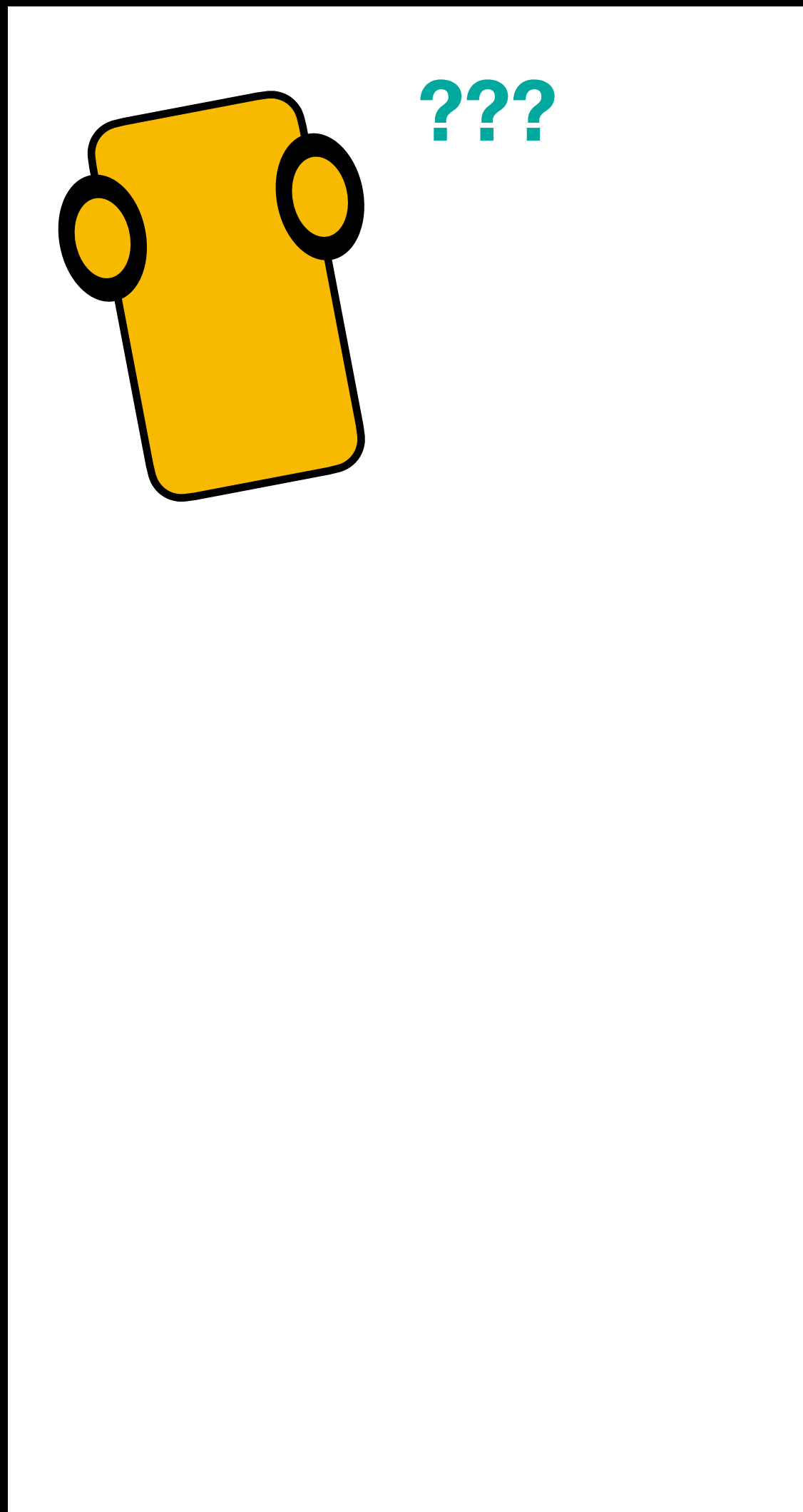
FREN



Auto



Envio Contínuo de Comandos pela Serial



Interrupção do Modo Automático

leitura do sensores



...



último instante  
na linha preta

🕒 5 segundos



PARA O CARRINHO!



## Aperfeiçoamento 01

Mude o código de volta da Serial1 para Serial, para facilitar os testes separados.

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.

Ignore possíveis comandos de direção vindos da serial ("frente", "tras", etc) durante o modo automático.

Ao receber o comando "parar", pare o carrinho e saia do modo automático.





## Aperfeiçoamento 02

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.

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`.

Caso o modo atual seja o automático e os dois valores recebidos dos sensores fiquem 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.

Altere o código das duas partes do Aperfeiçoamento para usarem a `Serial1`. Em seguida, teste a comunicação de dados entre os dois Arduinos.



[janks.link/micro/projeto09.zip](https://janks.link/micro/projeto09.zip)

Material do Projeto 09