



Projeto 09

Controle Serial – Prática

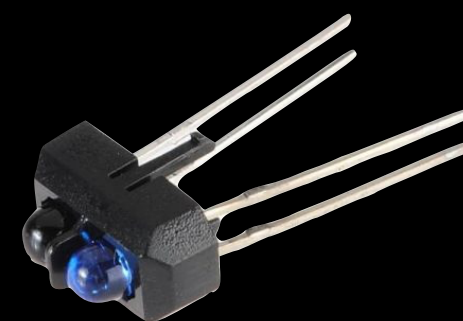
Jan K. S. – janks@puc-rio.br

ENG1419 – Programação de Microcontroladores

Testes Iniciais



motor 3
(esquerda)

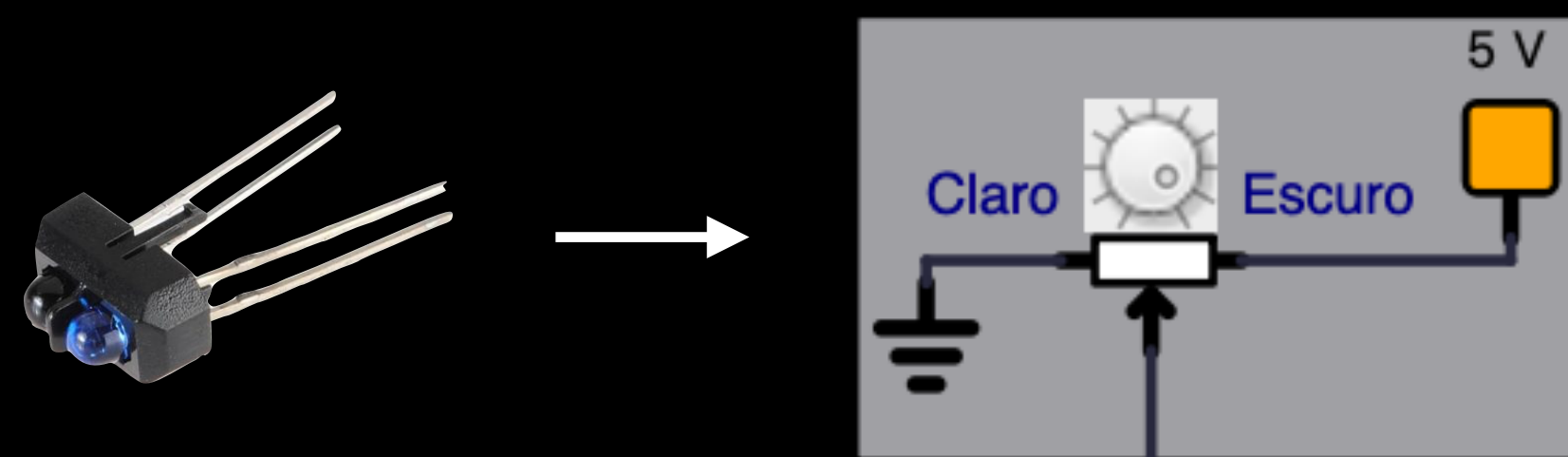


pino A11
(esquerda)

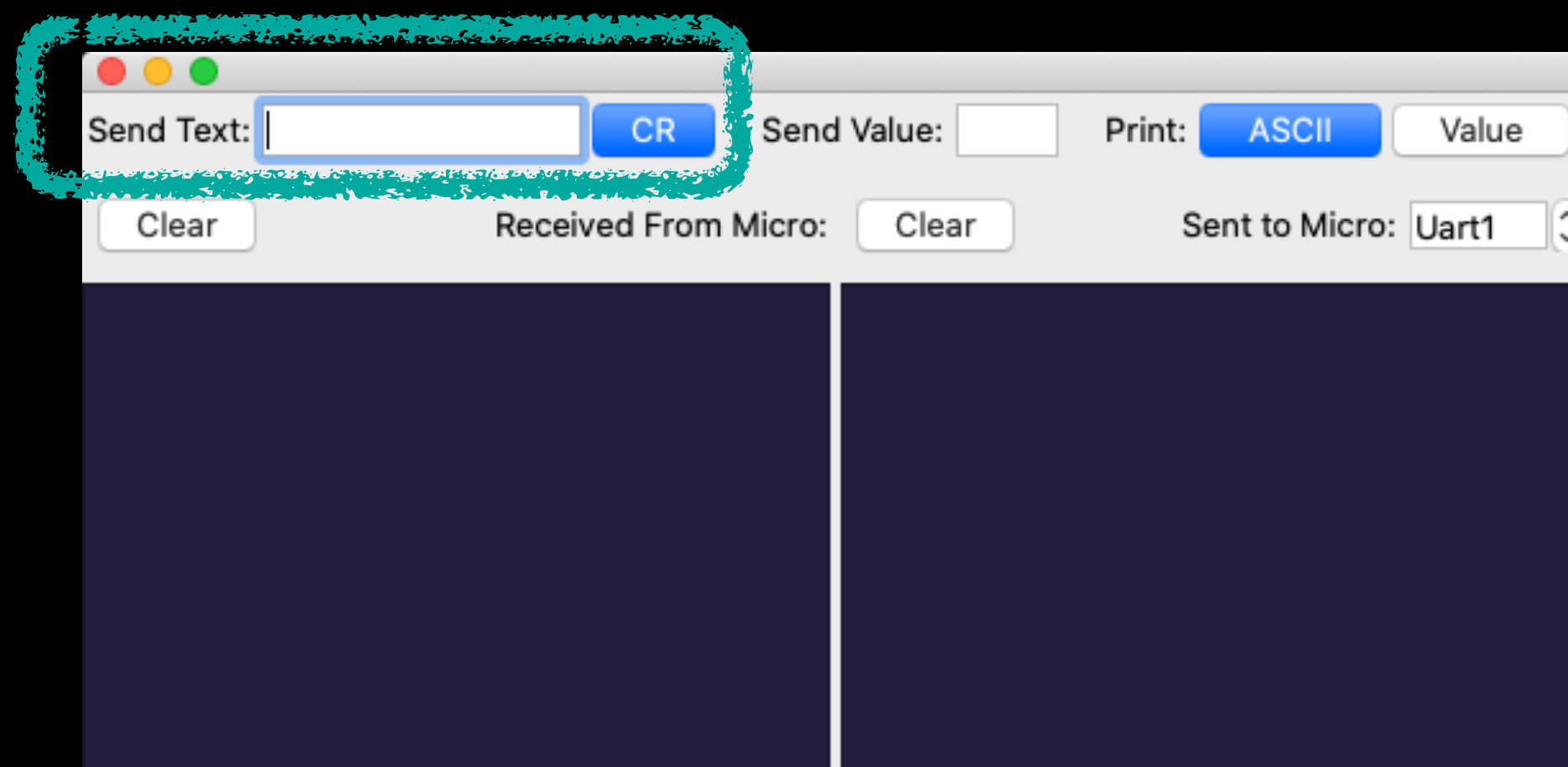


pino A12
(direita)

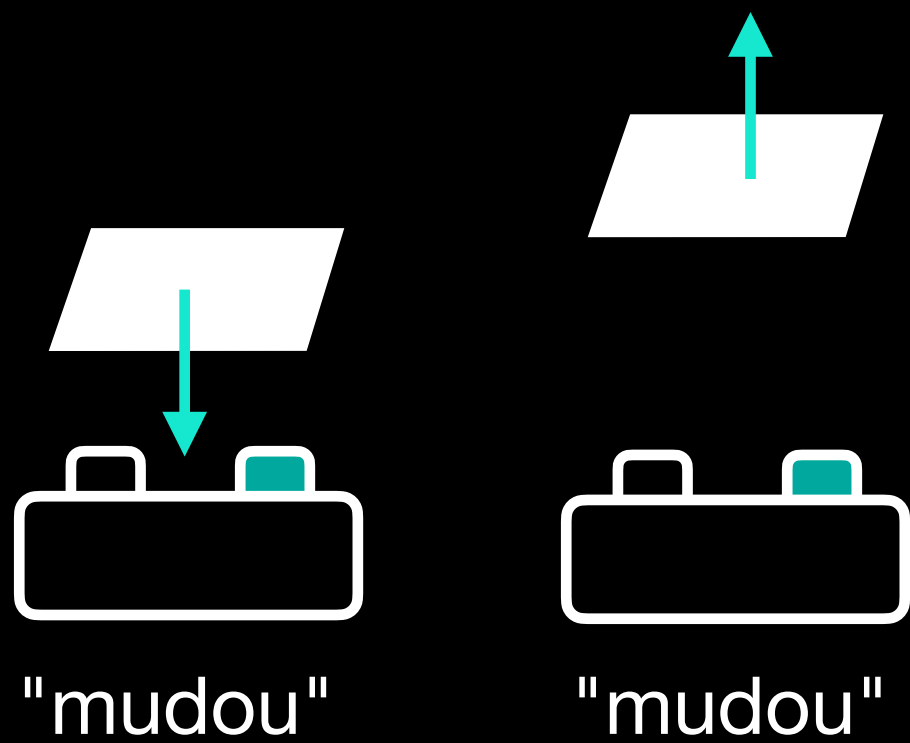
Pinos Usados pelos Componentes



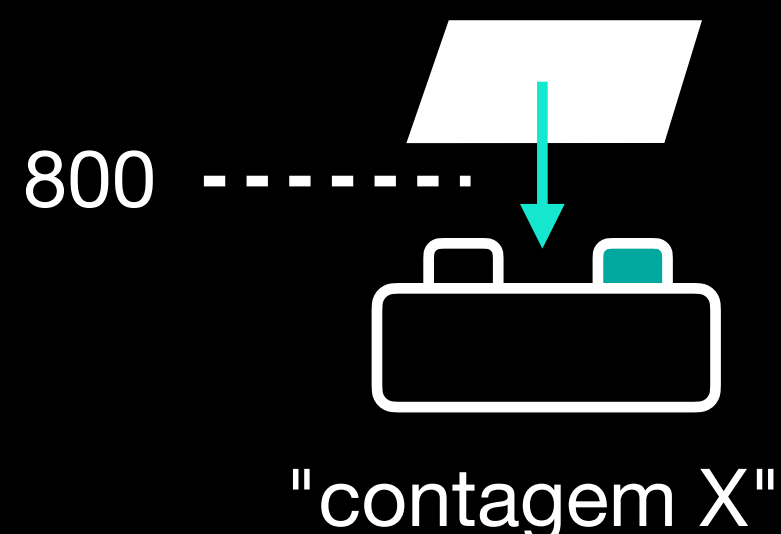
enviar texto para serial do Arduino



Simulação dos Sensores e da Serial



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.

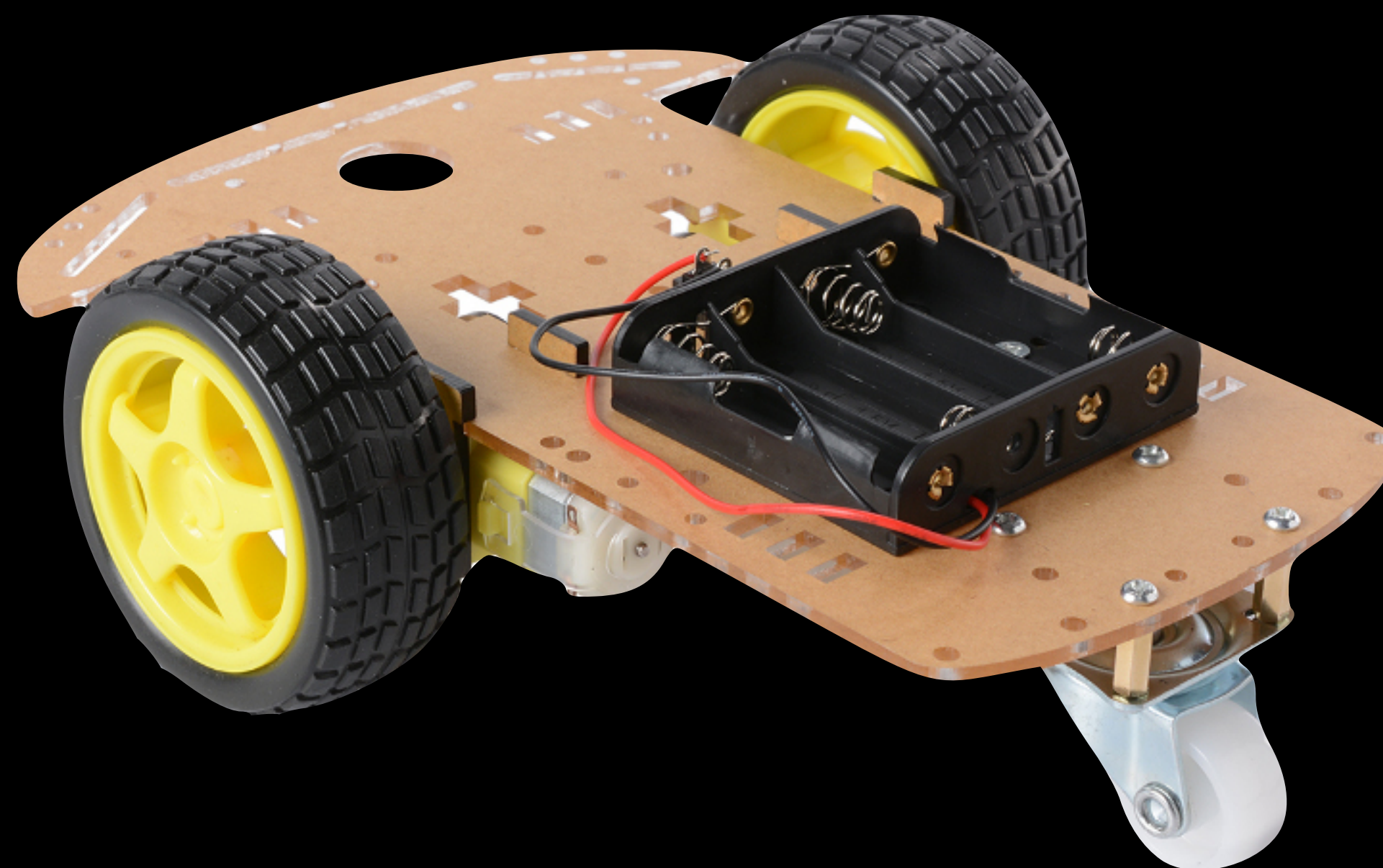
Ao receber o comando "frente N" da Serial, **gire o Motor 3 para frente com a velocidade N**. Ao receber "tras N", **gire para trás com a velocidade N**.

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

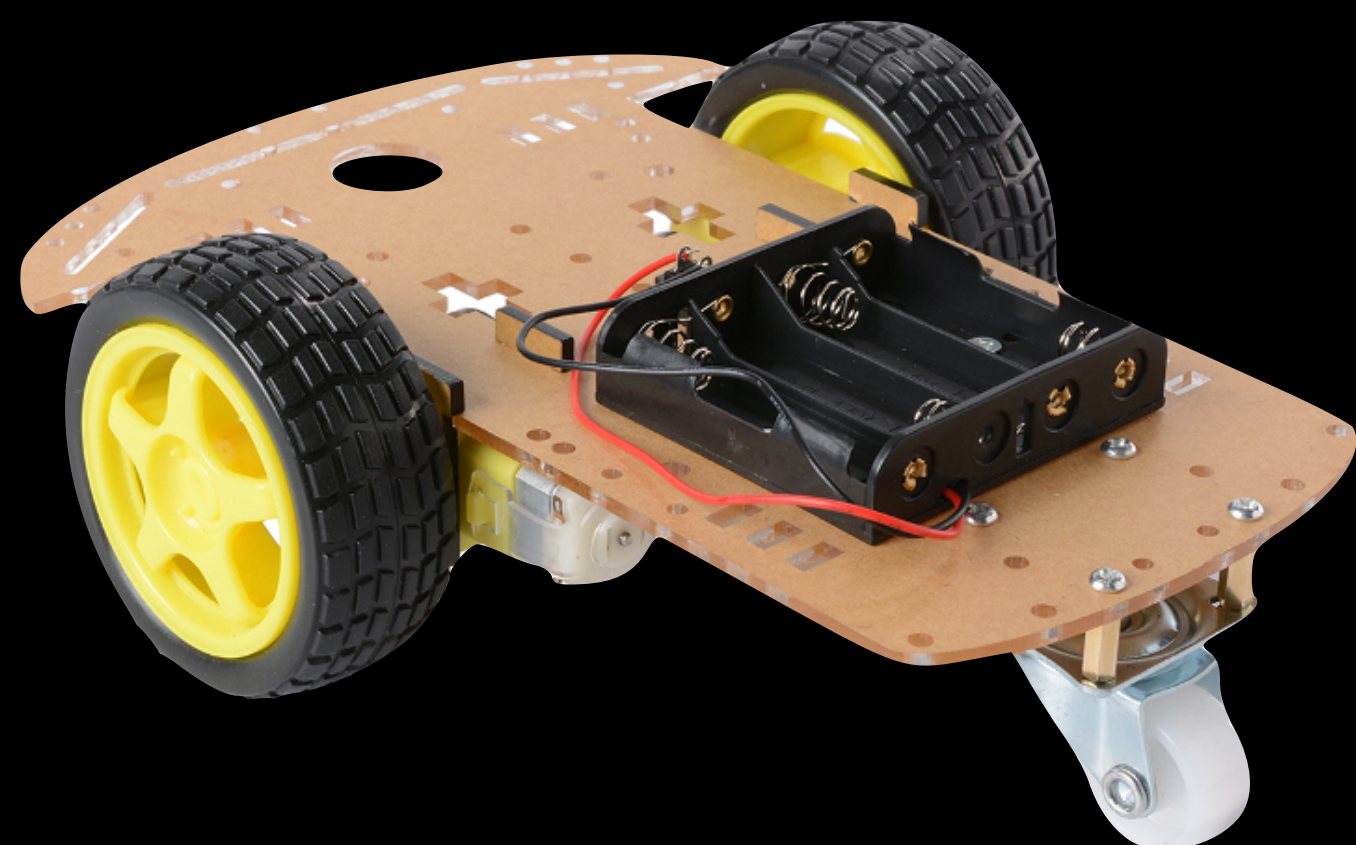
Aumente uma vez o contador X cada vez que o **valor analógico do sensor ótico 2 ultrapassar o limiar de 800**. Em seguida, **envie o texto "contagem X"** (com o valor de X) pela Serial.

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

Implementação



Carrinho de Duas Rodas



serial



Mostra Direção

Muda Direção

Andar / Parar

Controle Manual do Carrinho

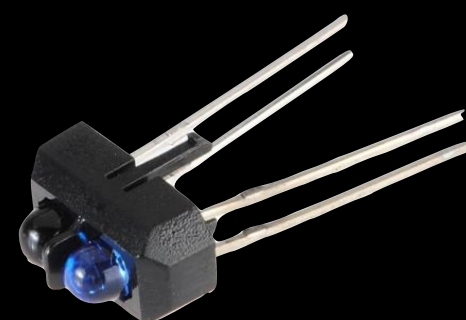
Implementação 1



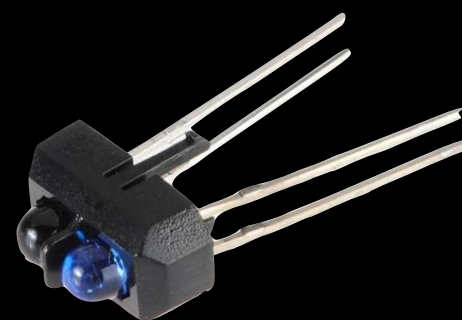
motor 3
(esquerda)



motor 4
(direita)



pino A11
(esquerda)



pino A12
(direita)

Implementação 2

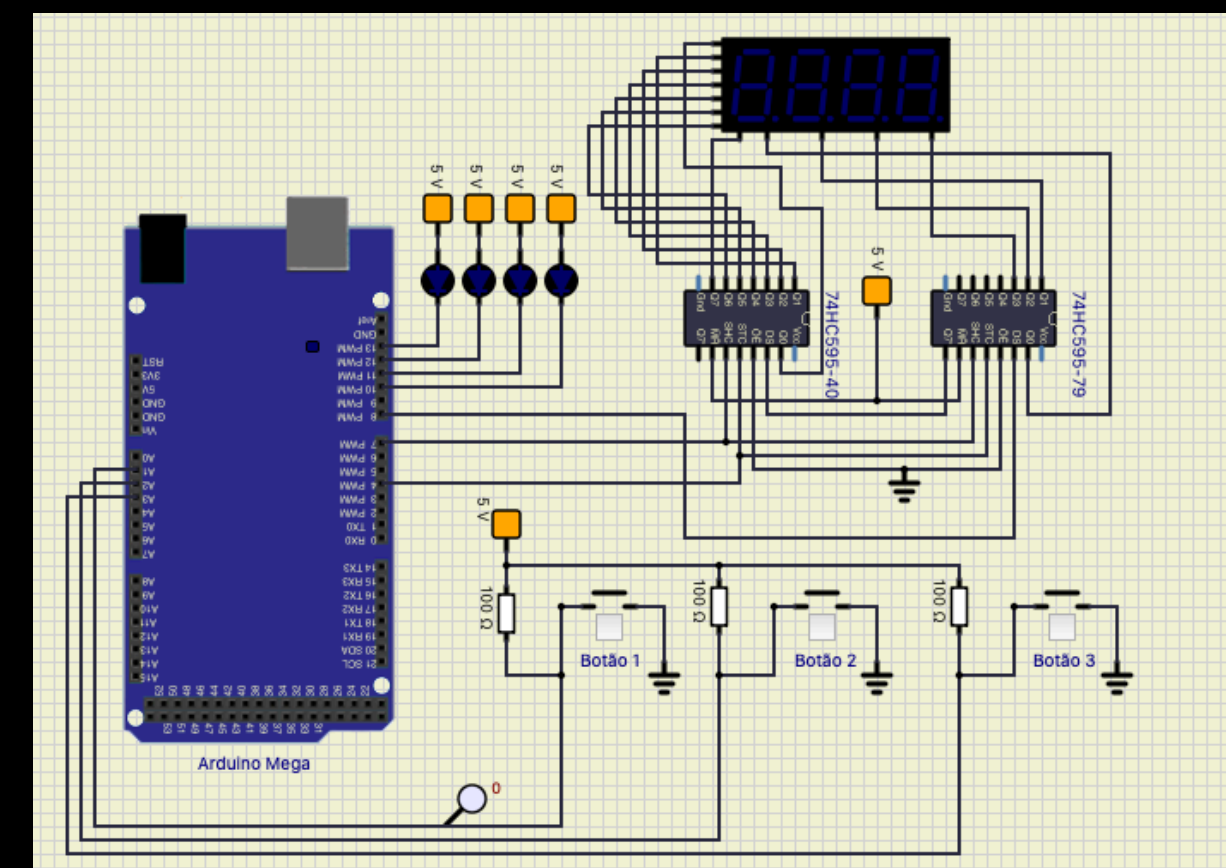
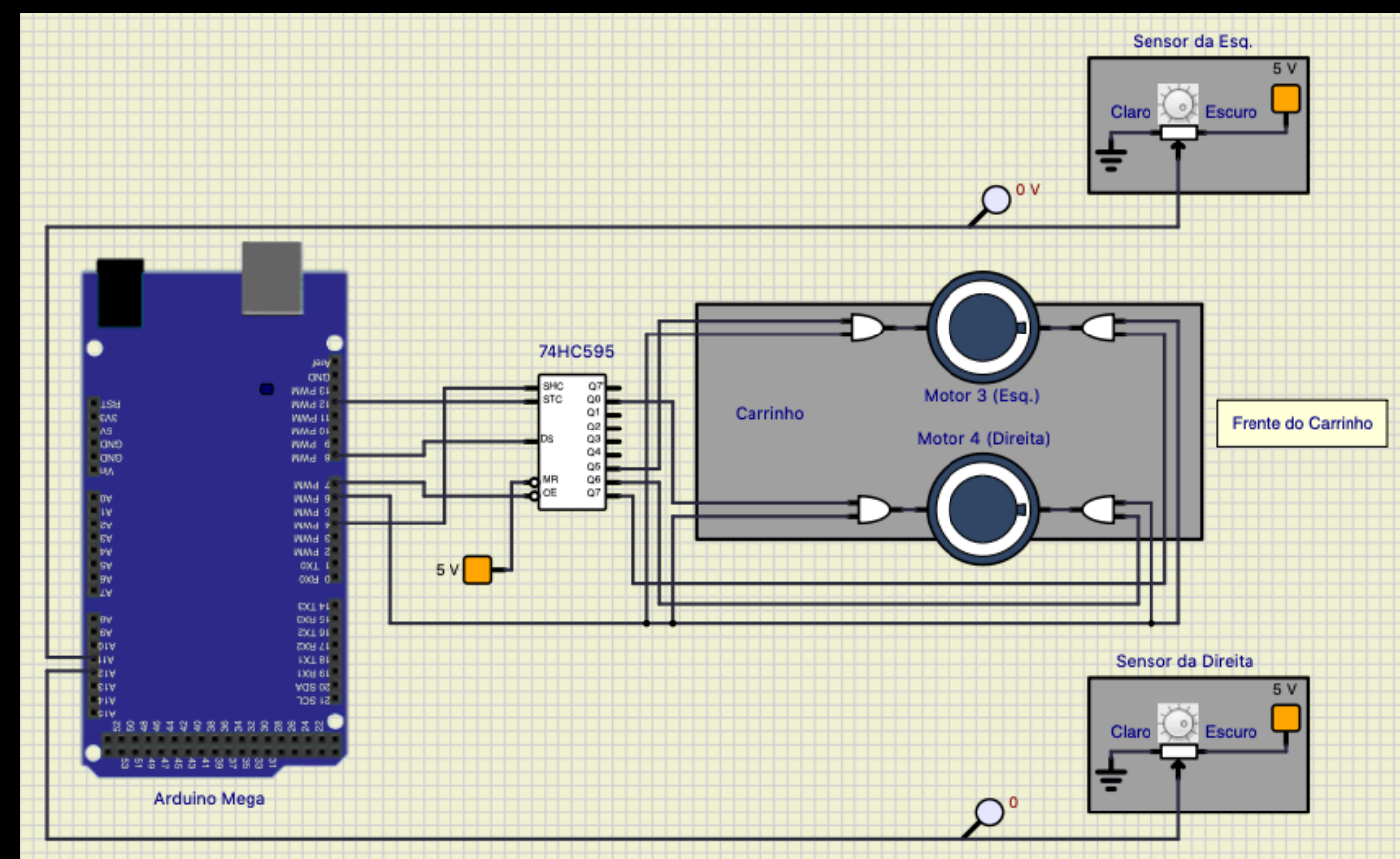
pinos 4, 7 e 8



pinos 13,
12, 11 e 10

pinos A1, A2 e A3

Pinos Usados pelos Componentes



Limitação da Simulação



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 os valores dos 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 da Implementação 01.

Aperfeiçoamento



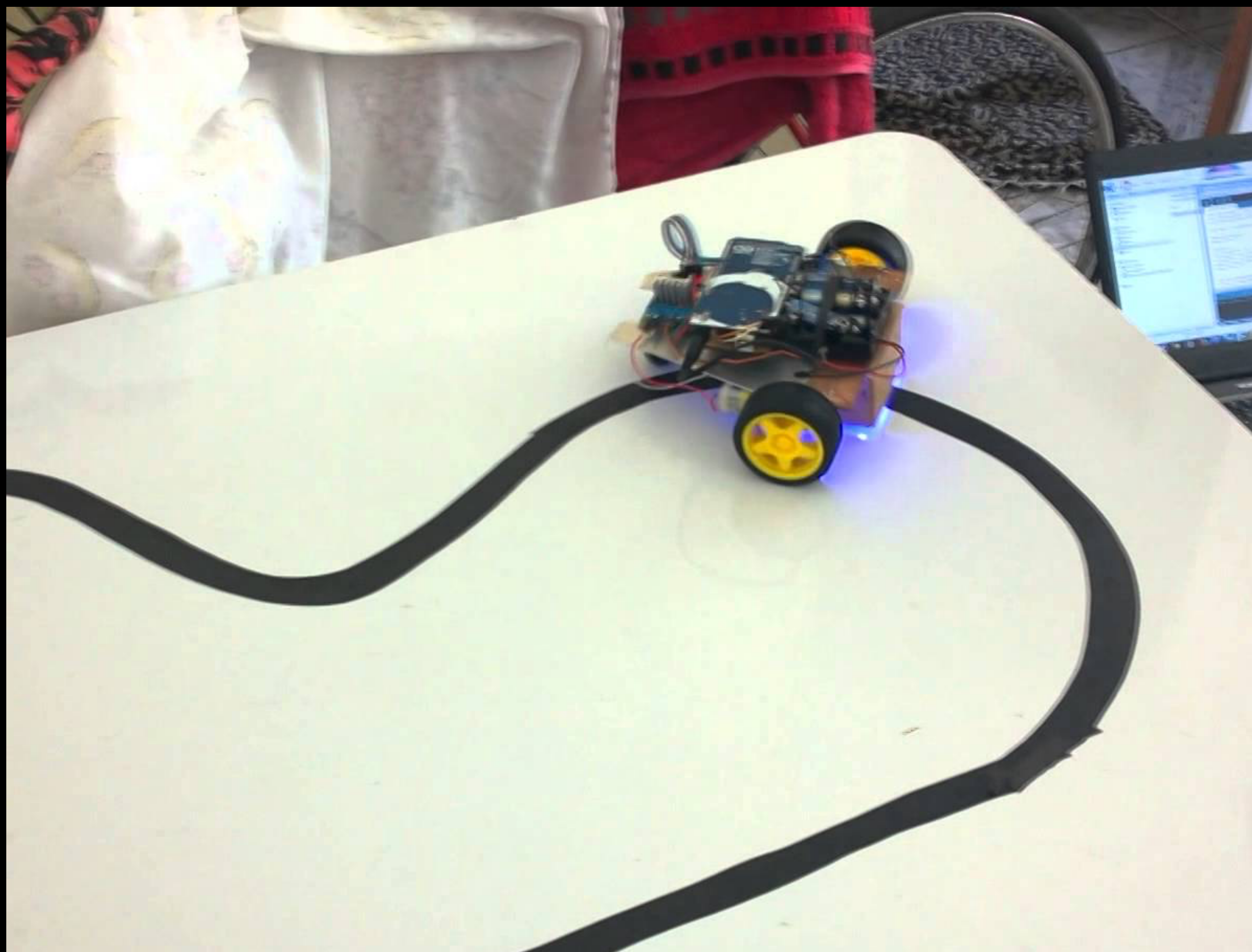
09b_implementacao.ino

cópia
-----▶



09c_aperfeicoamento.ino

Cópia do Código da Implementação para o Aperfeiçoamento



Modo Automático Seguidor de Linha

sensor 1



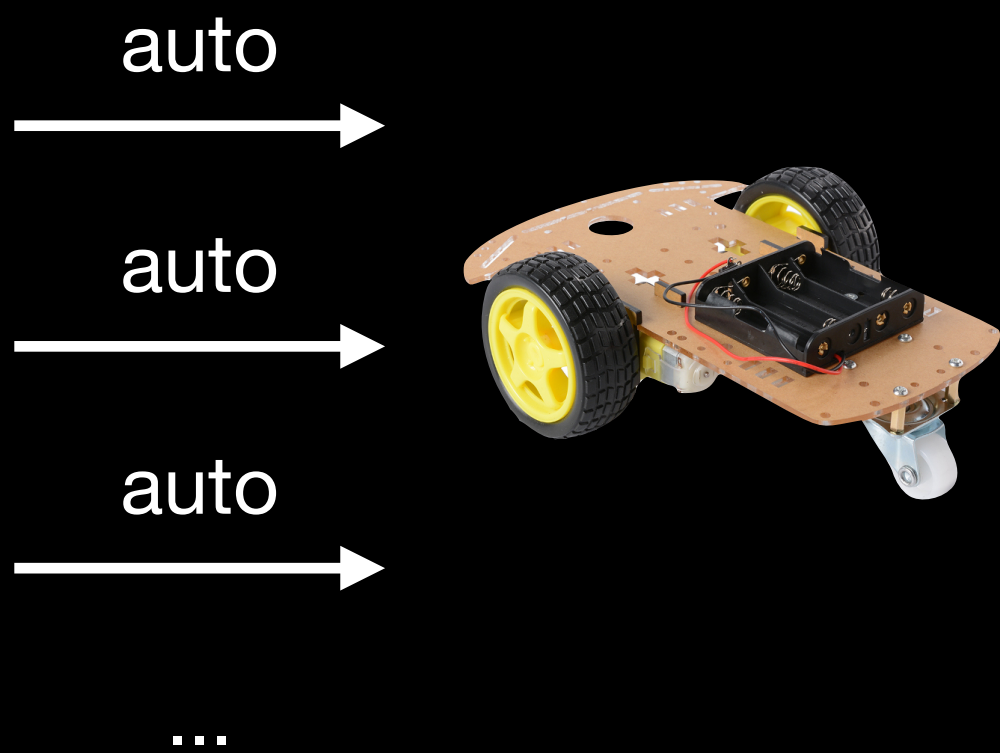
sensor 2



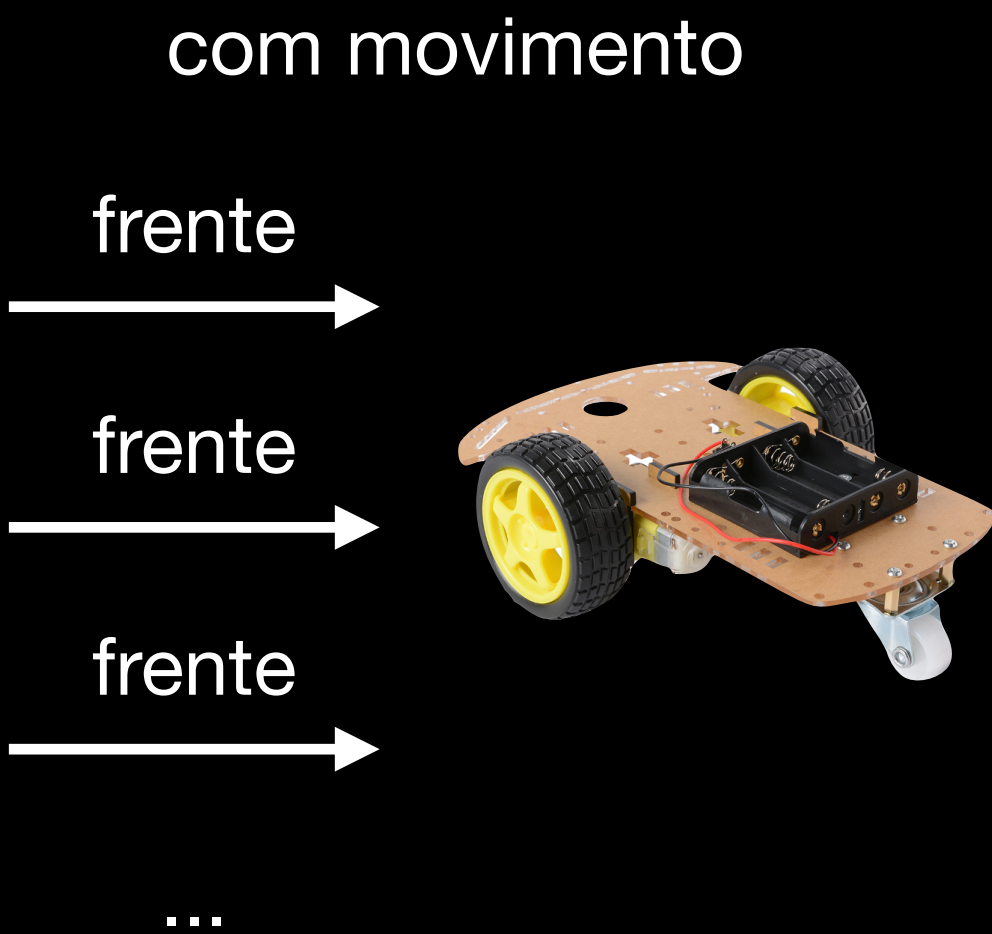
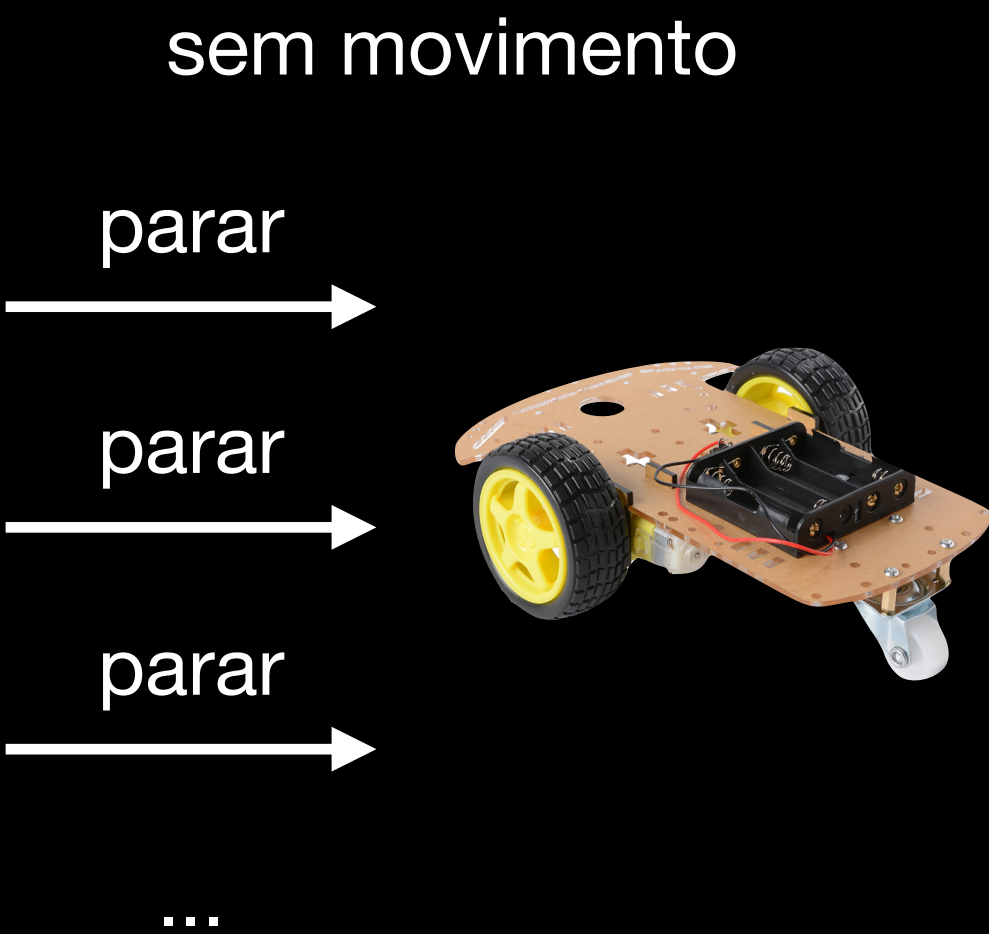
Dois Sensores Rastreando a Linha Preta

Envio contínuo pela Serial

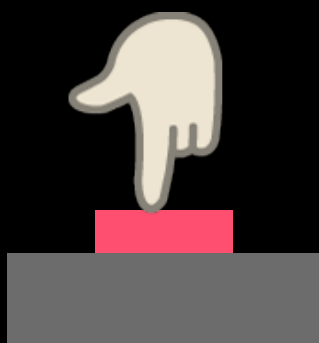
modo automático



modo manual

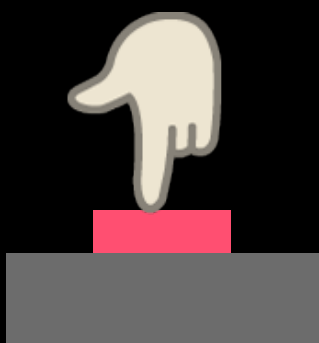


Botão 1

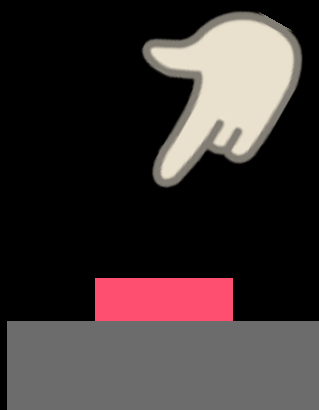


Muda comando atual no display

Botão 2

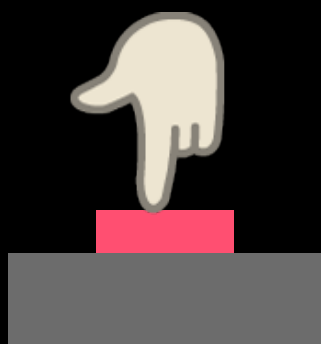


movimento → true
(no caso manual)



movimento → false
(no caso manual)

Botão 3



manual → automático
automático → manual

Mudança entre Modo Manual e Automático



Aperfeiçoamento 01

Ao receber o comando "auto" da serial, controle os motores de modo que o **carrinho mantenha o curso da linha preta**. Garanta que o carrinho nunca fique parado no modo automático.

↳ DICA: controle a direção de acordo com os valores dos dois sensores. Não complique o algoritmo de controle.

Ignore os comandos de direção durante o modo automático.

Ao receber o comando "parar", **pare o carrinho e volte a obedecer os comandos** anteriores de direção.

Ao apertar o Botão 3, alterne entre os modos automático e manual. No modo automático, o texto "auto" deve ser exibido no display, ignorando os botões 1 e 2. No manual, vale o que era antes.



Aperfeiçoamento 02

Envie a direção, o texto "auto" ou o texto "parar" a cada 50 milissegundos, de acordo com o estado atual do controlador, para proteger o carrinho contra falhas na comunicação.

↪ DICA: use a `millis` e uma nova variável global que indique se o carrinho está em movimento.

Caso o modo atual seja o automático e os valores recebidos dos sensores fiquem iguais a LOW (cor branca) durante 5 segundos, 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.



janks.link/micro/projeto09.zip

Material do Projeto 09