

EA871 – Laboratório de Programação Básica de Sistemas Digitais

Projeto Final – 2s2020

1. Objetivos

- Utilizar modulação por largura de pulso para ajustar a velocidade de motores de corrente contínua.
- Empregar interrupções periódicas para gerenciar a temporização de sistemas.
- Exercitar o uso da UART.
- Projetar sistemas com tarefas concorrentes.

2. Descrição do sistema

O projeto final consiste no desenvolvimento do sistema de acionamento de um carrinho dotado de dois motores, tração diferencial e um sensor de distância para evitar colisões, conforme ilustrado na Figura 1. Os comandos de controle do sistema são enviados através da interface serial. De acordo com o sentido com que os motores giram, definido pela polaridade da tensão aplicada em seus terminais, o carrinho pode ir para frente, para trás, girar no sentido horário ou anti-horário. É possível também controlar a frequência de rotação dos motores através de sinais PWM de modo a modificar a velocidade de deslocamento do carrinho.

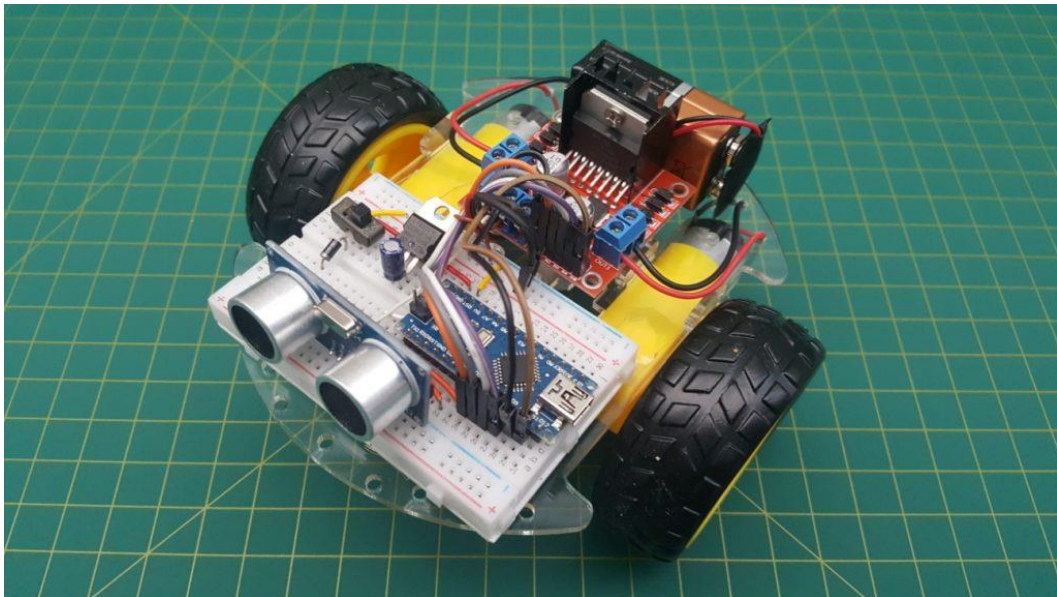


Figura 1: Carrinho com dois motores, tração diferencial e sensor de distância ultrassônico.

Vamos implementar e simular o funcionamento do sistema no Tinkercad, utilizando o circuito mostrado na Figura 2 e que pode ser copiado no link: <https://www.tinkercad.com/things/79dSArEP15s>

Uma vez recebido um comando pela UART, o sistema deve executar as ações necessárias e enviar uma mensagem de confirmação. A mensagem deve ser **reenviada periodicamente** a intervalos de **um segundo** até que um novo comando seja recebido. A Tabela 1 mostra os comandos que devem ser implementados e as correspondentes mensagens de confirmação. Comandos inválidos devem ser ignorados e não alteram o comportamento do sistema.

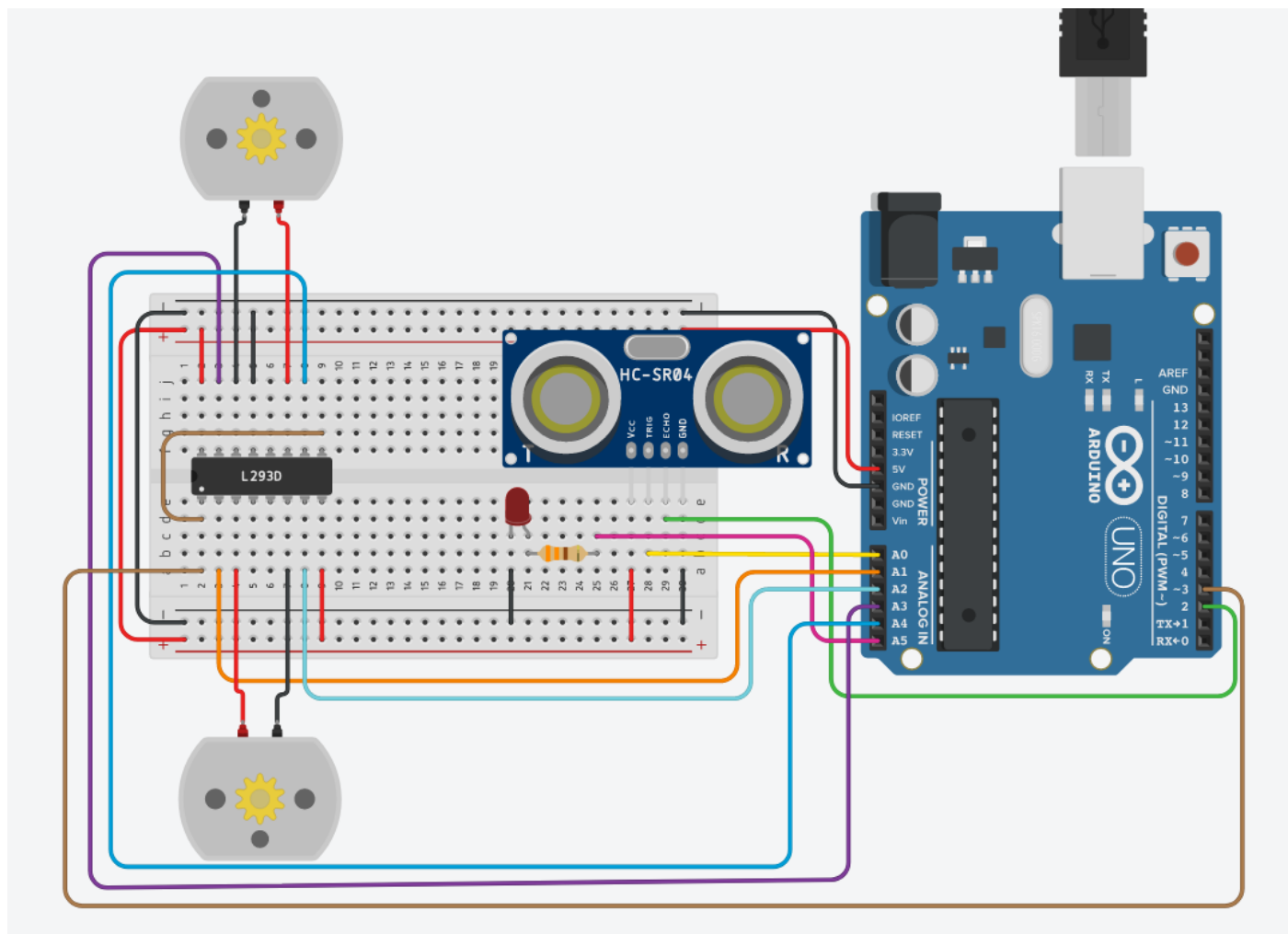


Figura 2: Circuito do sistema de acionamento do carrinho.

Comando	Ação	Mensagem
w	Para frente. <i>Duty cycle</i> inalterado.	"FRENTE\n"
s	Para trás. <i>Duty cycle</i> inalterado.	"TRAS\n"
a	Giro anti-horário. <i>Duty cycle</i> inalterado.	"ANTI-HORARIO\n"
d	Giro horário. <i>Duty cycle</i> inalterado.	"HORARIO\n"
q	Parar os motores. <i>Duty cycle</i> inalterado.	"PARADO\n"
e	Retorna a última medida de distância realizada (em centímetros). Na mensagem ao lado, a string DDD deve ser substituída pelo valor da distância (centena, dezena e unidade). Direção de movimento e <i>duty cycle</i> inalterados.	"DDDcm\n"
6	<i>Duty cycle</i> dos sinais PWM em 60%. Direção de movimento inalterada.	"Velocidade 60%\n"
8	<i>Duty cycle</i> dos sinais PWM em 80%. Direção de movimento inalterada.	"Velocidade 80%\n"
0	<i>Duty cycle</i> dos sinais PWM em 100%. Direção de movimento inalterada.	"Velocidade 100%\n"

Tabela 1. Lista de comandos e de mensagens.

2.1. Acionamento dos motores

Cada motor é acionado através de uma ponte H (L293D), que fornece a potência necessária para alimentá-lo e permite o controle do sentido e frequência de rotação do eixo. No nosso circuito, os movimentos previstos nos comandos são obtidos a partir da combinação dos níveis lógicos aplicados nos terminais A1, A2, A3 e A4 da placa de desenvolvimento, conforme mostrado na Tabela 2.

A velocidade de deslocamento é controlada por meio do *duty cycle* do sinal PWM gerado no terminal 3 da placa de desenvolvimento pelo temporizador 2. A frequência do sinal PWM deve ser alta o suficiente, da ordem de kHz, para que não haja oscilações na rotação do motor. Escolha um valor de frequência e um modo de operação do modulador apropriados para se obter as especificações do sistema.

Ação	A1	A2	A3	A4
Para frente	1	0	1	0
Para trás	0	1	0	1
Giro anti-horário	1	0	0	1
Giro horário	0	1	1	0
Parar	0	0	0	0

Tabela 2. Acionamento dos motores.

2.2. Sensor de distância

O sensor ultrassônico HC-SR04 é usado para medir continuamente a distância entre o carrinho e obstáculos frontais de modo a evitar colisões. Ao receber um pulso de no mínimo 10µs de duração no terminal TRIG (conectado ao terminal A0 da placa de desenvolvimento), o sensor dispara uma onda ultrassônica modulada que se propaga à sua frente. Ao colidir com um obstáculo, parte da onda retorna ao sensor que capta o eco e gera um pulso em seu terminal ECHO (conectado ao terminal 2 da placa de desenvolvimento) de largura igual ao tempo de trânsito da onda, ou seja, com duração igual ao intervalo transcorrido entre a emissão da onda e a recepção do eco. A Figura 3 mostra um diagrama de tempo que ilustra o processo.

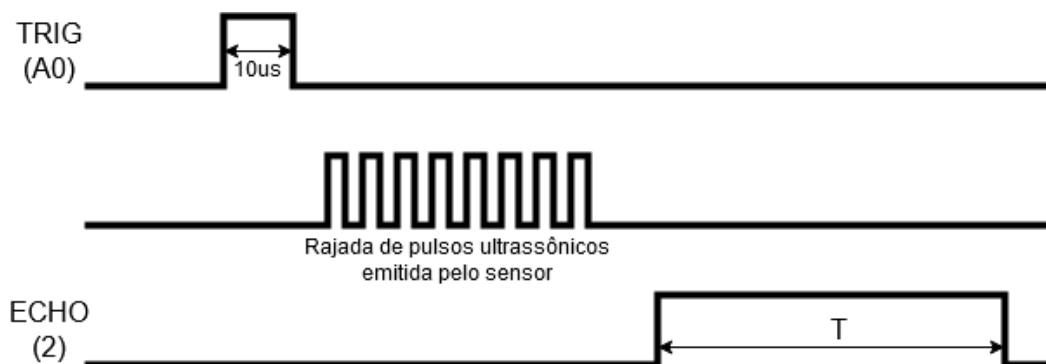


Figura 3: Diagrama de tempo do acionamento do sensor HC-SR04.

A relação entre a duração T do pulso gerado no terminal ECHO e a distância entre o sensor e o obstáculo é dada pela equação

$$\text{distância} = \text{velocidade do som} \times T/2$$

em que o valor da velocidade do som no ar é de aproximadamente 343m/s.

Portanto, para estimar a distância usando essa equação, é necessário medir a duração T do pulso gerado no terminal ECHO após a realização de um disparo. Podemos detectar os instantes em que as bordas de subida e de descida do pulso acontecem por meio de interrupções externas associadas a esses eventos e

medir o tempo entre eles usando um temporizador. De acordo com o manual do sensor, a duração do pulso pode assumir valores na faixa de $118\mu\text{s}$ a $23529\mu\text{s}$, que corresponde a distâncias de 2cm a 4m.

Para prevenir colisões, o sistema deve medir periodicamente, a cada 200ms, a distância entre o carrinho e obstáculos frontais. Caso a distância seja menor ou igual a 10cm, o carrinho deve parar e ser impedido de se movimentar para a frente, ignorando o comando 'w'. Nessa situação, o sistema passa a enviar a mensagem "OBSTACULO!\n" ao invés da mensagem do comando 'w'. O sistema responde normalmente aos demais comandos.

Para sinalizar a proximidade de obstáculos, o LED conectado ao terminal A5 deve piscar com frequência inversamente proporcional à distância medida pelo sensor, ou seja, quanto mais perto de um obstáculo, mais rápido o LED pisca. Defina uma relação entre frequência e distância que possibilite uma boa percepção visual da proximidade a partir do LED.

2.3. Configurações da UART

As operações de transmissão e de recepção via interface serial devem ser implementadas utilizando a estratégia de interrupção para lidar com os eventos assíncronos à execução do programa. Para simplificar o sistema, não usaremos o buffer circular nesse projeto. As especificações de configuração são:

- Velocidade de transmissão normal (i.e., modo *double-speed* desativado);
- Modo de transmissão multi-processador desabilitado;
- Número de bits de dados por *frame* igual a 8;
- Modo assíncrono de funcionamento da USART;
- Sem bits de paridade;
- Uso de um bit de parada;
- *Baud rate* igual a 9.600 bps.

2.4. Resumo do sistema

Em suma, o sistema é composto das seguintes partes principais:

- Recepção e execução de comandos;
- Envio de mensagens: a mensagem associada ao comando em execução deve ser enviada periodicamente a cada 1s;
- Acionamento dos motores de acordo com o sentido de rotação especificado no comando em execução;
- Ajuste da frequência de rotação dos motores de acordo com o comando em execução;
- Atualização da medida da distância entre o carrinho e os obstáculos frontais a cada 200ms;
- Acionamento do LED indicador de distância de modo que a frequência de cintilação aumente com a proximidade do obstáculo.

Observações

- Na atividade como um todo, **não é** permitido usar rotinas de atraso, com exceção ao atraso usado no laço principal para acelerar a simulação. Utilize os temporizadores!
- Como os comandos serão enviados um por vez, não é necessário trabalhar com *buffer* circular.
- O sistema deve ser iniciado com o carrinho executando o comando 'q' (parado), *duty cycle* 60% e enviando a mensagem "PARADO\n", conforme a Tabela 1.
- Como necessitamos de uma base de tempo da ordem de μs para estimar a distância, construam rotinas de serviço de interrupção o mais breves possível para que não haja interferência nas interrupções do temporizador. Façam uso de variáveis de sinalização que permitam tratar as tarefas no laço principal.

- Para acelerar a simulação, utilizem um prescaler igual a 32 para gerar a forma de onda no temporizador 2.
- Usem o temporizador 0 ou o 1 para gerar a base de tempo empregada na estimativa da distância.
- Para imprimir a distância, é necessário primeiro converter o valor para a cadeia ASCII correspondente que o representa e depois enviar cada caractere pela UART.

Instruções para a submissão do projeto

- 1) A atividade pode ser feita em duplas.
- 2) Nos comentários do código-fonte, explique o funcionamento geral do programa e justifique as operações e os valores carregados em todos os registradores. Explique detalhadamente o mecanismo de medida da distância. Lembre-se que seus comentários fazem parte do relatório. Programas sem comentários terão nota máxima 5. Justificativas de configuração incorretas, mesmo que a configuração esteja certa, serão penalizadas fortemente, principalmente se tiverem sido discutidas em exemplos disponibilizados em vídeo. Incluam o nome e RA dos integrantes do grupo nos comentários no início do programa.
- 3) Ao final do projeto, salve o código-fonte com o nome “seu_ra.c” (Exemplo: 025304.c) se o projeto for feito individualmente. No caso de projetos feitos em duplas, salve o código-fonte com o nome “ra1_ra2.c” (Exemplo: 025304_012345.c).
- 4) Faça o upload do código fonte comentado no Google Classroom. Se o projeto foi feito em duplas, os dois alunos devem submeter o mesmo arquivo.