

Projeto II
Controle de trajetória de ônibus elétrico

1 Introdução

Este projeto da disciplina visa a implementação técnica de controle em um microcontrolador ATmega 328p montado em circuito próprio. Neste documento, apresenta-se a descrição de um sistema dinâmico para aplicação do controle, consistindo de modelo de ônibus elétrico simulado em kit Arduino. Outros sistemas a controlar podem ser escolhidos para o projeto em comum acordo com os professores da disciplina.

1.1 Requisitos do projeto

1.1.1 Requisito Funcional

A tarefa de controle do ônibus elétrico requer o cumprimento de itinerário pré-definido no menor tempo possível, respeitadas as restrições de energia disponível. Nos pontos de parada, a bateria do ônibus pode ser recarregada até o nível adequado definido pelo controlador. Assim, apesar do alto consumo do ônibus em relação à energia acumulável na bateria, o itinerário pode ser completado se as recargas nos pontos forem suficientes.

Detalhes do modelo do ônibus são apresentados nos capítulos 2 e seguintes.

1.1.2 Requisito de Sistema

O controlador do sistema deve ser implementado em circuito montado a partir dos materiais fornecidos em laboratório, que são:

- circuito integrado do microcontrolador (ATmega 328 ou equivalente);
- placa de contatos (protoboard);
- gravador USBasp;
- conectores, resistores, capacitores;

- kit Arduino para simulação do sistema a controlar.

Os aspectos da montagem do circuito para uso do simulador são apresentados nos capítulos 2 e seguintes.

1.2 Equipe e Cronograma

- O projeto deve ser realizado em equipe de dois ou três estudantes.
- O prazo de conclusão do sistema completo é 16 de dezembro, sexta-feira, com defesa presencial no período de 9:30 h a 12:30 h.
- A documentação técnica com a descrição das soluções adotadas deve ser entregue via Moodle até as 23:59 h do dia 16 de dezembro.

2 Simulador de ônibus elétrico

O simulador de ônibus elétrico implementado em Arduino fornece uma plataforma para controle da trajetória de um ônibus elétrico ao longo de um itinerário. A Fig. 1 apresenta um exemplo de simulação do sistema com a saída gráfica no “plotter serial” do ambiente (IDE) Arduino. Observa-se que:

- as retas horizontais correspondem às posições dos pontos de parada;
- a curva em lilás representa a posição do ônibus ao longo do itinerário;
- a curva em cinza representa a velocidade de percurso; e
- a curva celeste representa o nível de energia estocada na bateria.

Os sinais mostrados no gráfico indicam que a energia é gasta ao longo do percurso e pode ser parcialmente recuperada em frenagens. Além disso, recargas ocorrem sempre que o ônibus estiver parado nos pontos. Esse comportamento é similar à operação com baterias de recarga rápida com “supercapacitores”. Nota-se, ainda, que a energia pode esgotar-se ao longo do percurso, causando a parada do ônibus.

3 Implementação do simulador do ônibus em Arduino

A implementação do simulador em ambiente Arduino usa o código fornecido no endereço seguinte: <https://www.tinkercad.com/things/loENfvDRUhq-simulador-de-onibus-eletrico/editel>.

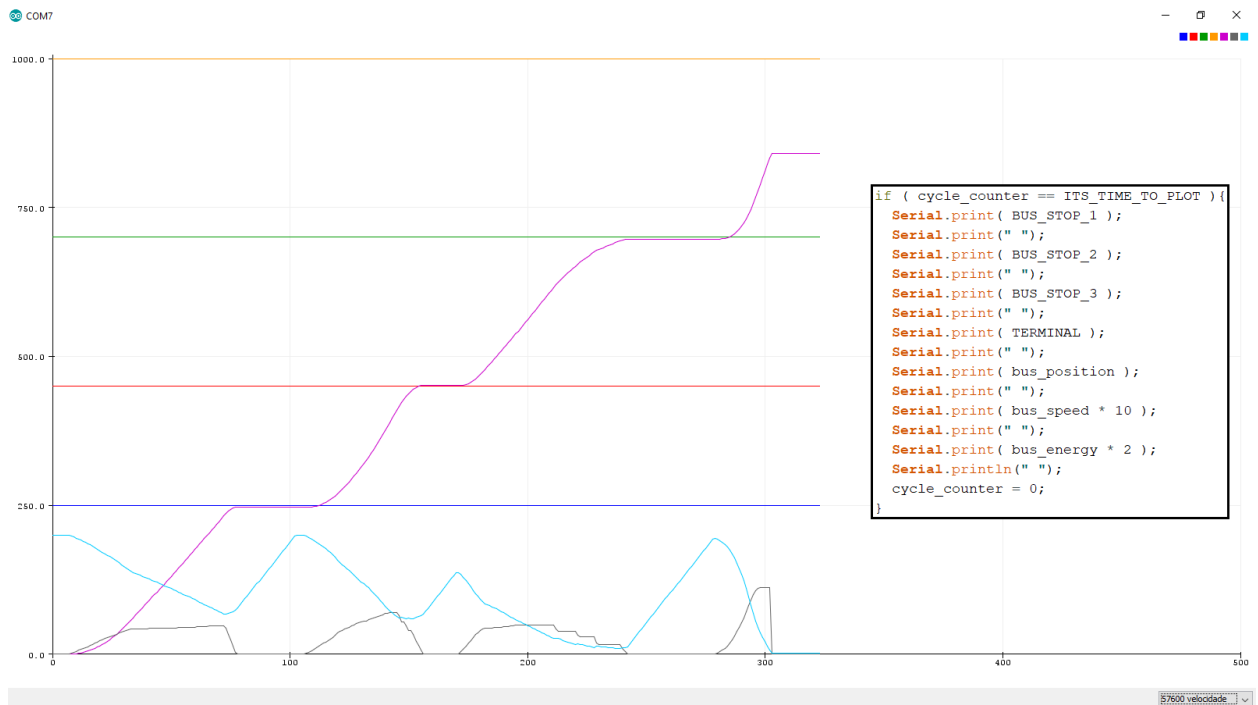


Figura 1: Trecho de código (à direita) e gráfico resultante no “plotter serial” do ambiente Arduino durante a simulação do ônibus elétrico; a legenda de cores no canto superior direito segue a ordenação indicada no código. A taxa de comunicação é de 57600 bits/s.

Para controle da trajetória, há diversos sinais implementados no circuito com kit Arduino, conforme mostrado na Fig. 2. Os sinais podem ser classificados como de “saída” (sinais que fornecem medições do estado do ônibus) e de “entrada” (sinais de atuação para controle do estado do ônibus), sendo dados por:

- Sinais de saída (medição do estado do ônibus), atualizados a cada 1 segundo:

Pino 5: representa a posição do ônibus, que varia de 0 a 1.000 m.

Tipo de sinal: PWM gerado por *analogWrite()*, 256 níveis, frequência de base de 980 Hz.

Codificação da posição: 0 (origem) = 10 (PWM); 1.000 m (terminal) = 250 (PWM)

Pino 6: indica o valor da velocidade, que varia de 0 a 72 km/h, i. é, de 0 a 20 m/s.

Tipo de sinal: PWM gerado por *analogWrite()*, 256 níveis, frequência de base de 980 Hz.

Codificação da velocidade: 0 = 50 (PWM); 20 m/s = 250 (PWM).

Pino 9: fornece o estado da bateria, a qual pode ser recarregada nos pontos de parada.

Tipo de sinal: PWM gerado por *analogWrite()*, 256 níveis, frequência de base de 490 Hz.

Codificação da energia: 0 = 50 (PWM); 100% = 250 (PWM).

- Sinais de entrada (atuação para controle):

Pino A0: aceleração propriamente dita, definindo valores positivos ou zero.

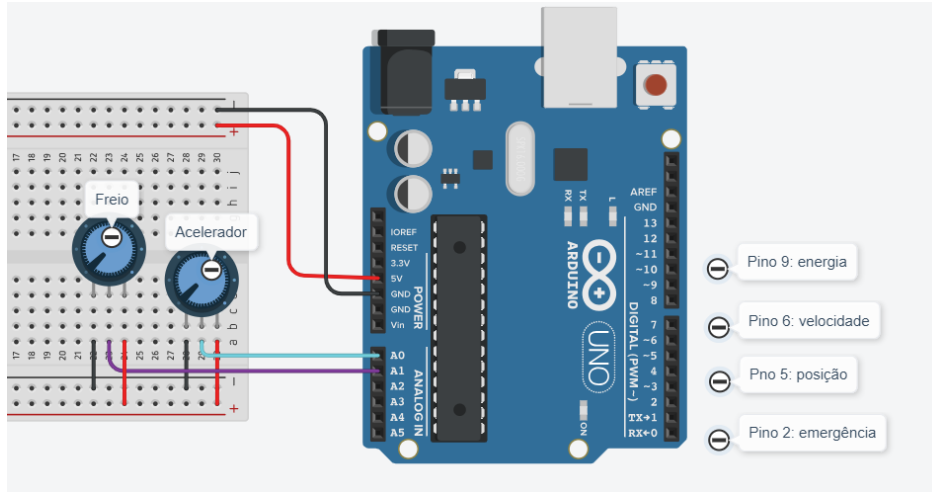


Figura 2: Kit Arduino Uno com marcação dos sinais relevantes para a simulação do ônibus elétrico.

Tipo de sinal: analógico, de 0 a 5 V.

Codificação da aceleração: $0 = 0 \text{ V}$, $1 \text{ m/s}^2 = 5 \text{ V}$.

Pino A1: frenagem, definindo valores negativos de aceleração.

Tipo de sinal: analógico, de 0 a 5 V.

Codificação da frenagem: $0 = 0 \text{ V}$, $-4 \text{ m/s}^2 = 5 \text{ V}$.

Pino 2: sinal de emergência, não usado na implementação. Caso o programador decida criar funcionalidades associadas à esta entrada, deverá codificá-la no modelo de simulação do ônibus.

Leitura e escrita dos sinais PWM em modo analógico

O simulador do ônibus recebe, por padrão, os sinais de aceleração e de frenagem em modo analógico (tensões de 0 a 5 V), conforme ilustrado pelos potenciômetros da Fig. (2). Portanto, o controlador que vier a substituir tais potenciômetros deve fornecer sinal contínuo, em vez do sinal PWM criado pela função *analogWrite()* do Arduino.

Para realizar a passagem do sinal PWM para sinal contínuo, usa-se o circuito elétrico RC (resistor + capacitor) mostrado na Fig. 3, também denominado de filtro RC. No circuito, o capacitor armazena energia durante o tempo em alta do sinal e a descarrega durante o tempo em baixa, atenuando as variações. Os valores dos componentes da figura são de $C = 100 \text{ nF}$ e $R = 47 \text{ k}\Omega$, porém recomenda-se usar $C = 1 \mu\text{F}$ para haver menor flutuação do sinal. A Fig. 4 mostra o diagrama completo de montagem com filtros RC (resistor + capacitor) e o segundo kit Arduino no qual se implementa a lógica de controle.

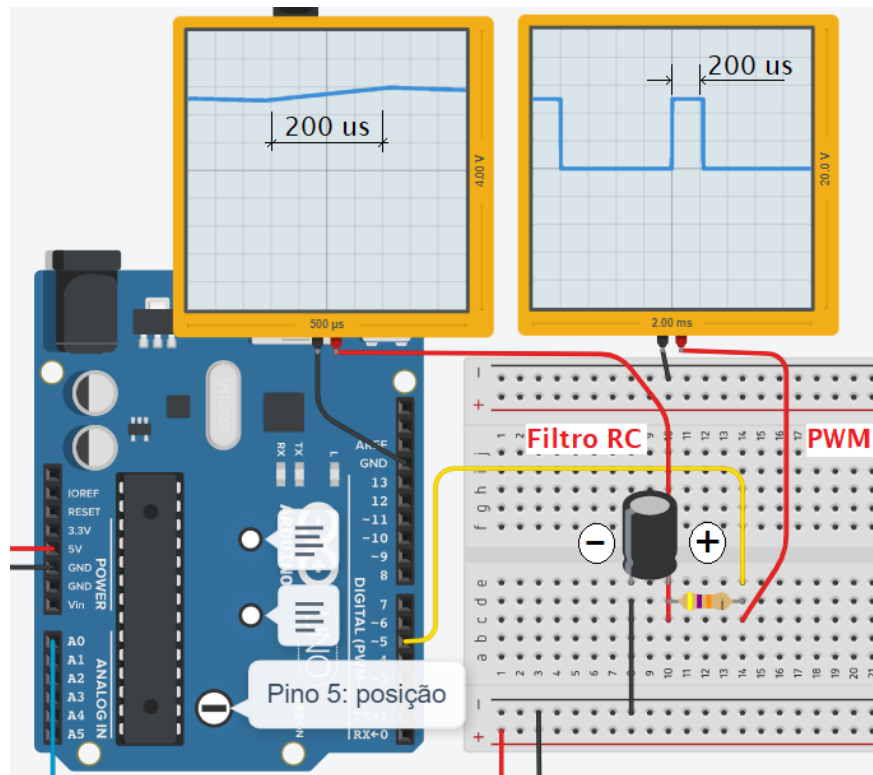


Figura 3: Sinais elétricos da posição do ônibus com PWM (direita) e com sinal analógico na saída do filtro com circuito RC (esquerda); note a variação do sinal analógico sobre o capacitor durante o carregamento (tempo de $200 \mu s$ de PWM em alta).

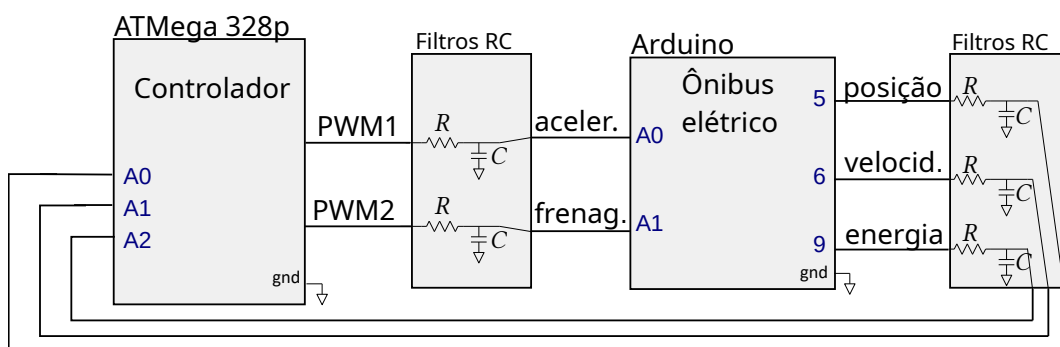


Figura 4: Diagrama do sistema de controle automático da simulação do ônibus elétrico com sinais analógicos obtidos por filtragem dos sinais PWM; recomenda-se usar $R = 47 \text{ k}\Omega$ e $C = 1 \mu\text{F}$. (Observar a necessidade de conectar todos os componentes a uma referência comum, representada por “gnd”).

4 Modelo do ônibus

4.1 Modelo cinemático

O modelo do ônibus descreve o comportamento cinemático do veículo, relacionando acelerações instantâneas com a velocidade e a posição ao longo do tempo. Supõe-se que efeitos dinâmicos (atritos, inclinação do trajeto, potência dos motores) sejam tratados pelo sistema mecânico de tal forma que valores de aceleração comandados pelo controlador sejam sempre aplicados ao veículo.

Desta forma, o modelo do ônibus é dado por:

$$v(t) = \int_0^t a(\sigma) d\sigma, \quad (1)$$

$$s(t) = \int_0^t v(\sigma) d\sigma. \quad (2)$$

Portanto, a simulação consiste da integração iterativa da aceleração instantânea $a(t)$ para cálculo da velocidade $v(t)$, e da velocidade para cálculo da posição $s(t)$.

A aceleração é calculada por:

$$a(t) = K_a P_a(t) - K_b P_b(t), \quad (3)$$

com $P_a(t)$ e $P_b(t)$ representando os sinais de aceleração e de frenagem e K_a e K_b constantes.

Considera-se, ainda, que a velocidade máxima do ônibus é de 72 km/h, ou 20 m/s.

4.2 Modelo de consumo energético

Para cálculo do consumo instantâneo de energia do ônibus, usa-se fórmula *ad hoc* dada por:

$$C_e(t) = -K_{v1}v(t) - K_{v2}v(t)^2 - K_{acel}a(t), \quad \text{se } a(t) \geq 0 \quad (4)$$

$$C_e(t) = K_{fren}a(t), \quad \text{se } a(t) < 0. \quad (5)$$

Note que, neste modelo, não há penalização para uso simultâneo de acelerador e freio, limitando o realismo da representação.

A energia estocada é calculada como a integral do consumo instantâneo.

4.3 Atribuição de parâmetros

Para a integração numérica do modelo, usa-se passo de simulação de 10 ms. A integração é feita com a fórmula de Euler. As constantes de aceleração foram definidas com base nos valores aplicados nas entradas de aceleração e de frenagem do Arduino, isto é, valores de 0 a 1023. Optou-se por definir a aceleração de 1 m/s² para valor máximo, e de -4 m/s² como mínimo. Assim, tem-se que

$$K_a = 0.001$$

$$K_b = 0.004.$$

Os parâmetros relacionados ao consumo de energia são dados por:

$$K_{v1} = 0.1$$

$$K_{v2} = 0.02$$

$$K_{accel} = 5$$

$$K_{fren} = 1.$$