

EA701 A - Sinais Analógicos, ADC, PWM e DAC

Raphael Salles Vitor de Souza - RA 223641
Vinicius Patriarca Miranda Miguel - RA 260731

September 16, 2025

Material usado

Neste experimento, utilizamos os seguintes materiais para analisar o funcionamento do PWM e como o aumento de resistências influenciava a impedância e a forma de onda gerada pelo Raspberry Pi Pico:

- BitDogLab (com Raspberry Pi Pico/Pico 2)
- Protoboard pequena
- Multímetro digital
- Osciloscópio digital (compartilhado, se disponível)
- Resistor 10 kΩ (1/4 W)
- Capacitor 100 nF (cerâmico)
- Resistor 1 kΩ (opcional, proteção)
- 10 jumpers macho-macho e macho-fêmea
- Resistores extras (1 kΩ, 47 kΩ) (*opcional*)
- Capacitores extras (10 nF, 1 μF) (*opcional*)

1 Leitura do Joystick

A leitura do joystick é feita da seguinte forma, usando a função `read_analog()`:

```
uint16_t read_analog(uint pin) {  
    adc_gpio_init(pin);  
    adc_select_input(pin - 26);  
    return adc_read();  
}
```

e usamos essa linha para imprimir os valores lidos:

```
printf("Joystick X: %u (%.2f V) | Joystick Y: %u (%.2f V)\n", x_analog, x_volts, y_analog, y_volts);
```

Os valores lidos estão na Figura 1:

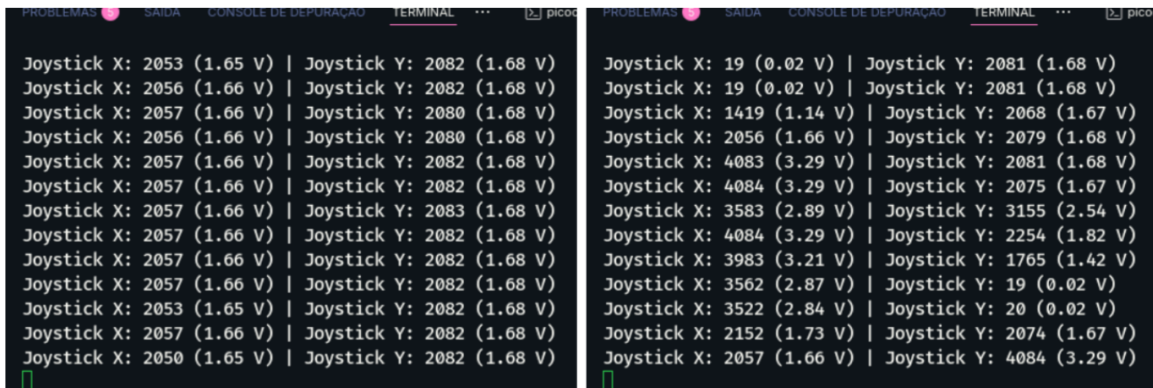


Figure 1: Valores lidos do joystick exibidos no terminal.

e quando o joystick está na "posição inicial" o valor observado para x e para y são 1.66V e 1.68V respectivamente. Para conseguir ler o tempo todo usamos a seguinte lógica:

```

while (true) {
    uint16_t x_analog = read_analog(JOYSTICK_X);
    uint16_t y_analog = read_analog(JOYSTICK_Y);

    float x_volts = (x_analog / 4095.0) * 3.3;
    float y_volts = (y_analog / 4095.0) * 3.3;

    printf("Joystick X: %u (%.2f V) | Joystick Y: %u (%.2f V)\n", x_analog, x_volts, y_analog, y_volts);
    sleep_ms(500);
}

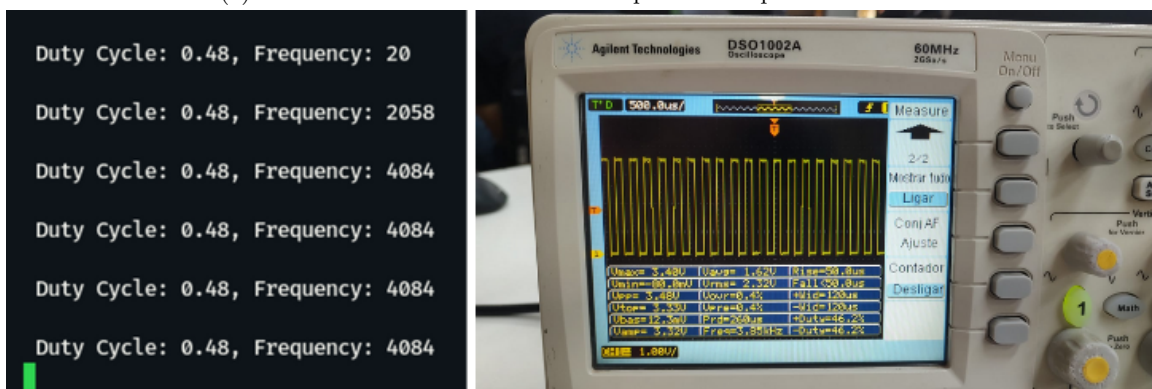
```

2 PWM controlado pelo Joystick

Os valores de frequência e duty observados variaram conforme o ajuste do joystick, que controla o ciclo de trabalho do PWM gerado pelo Raspberry Pi Pico foram:



(a) Sinal PWM observado no osciloscópio com frequência de 2 kHz.



(b) Sinal PWM observado no osciloscópio com frequência de 4 kHz.

Figure 2: Formas de onda PWM para diferentes posições do joystick.

3 Filtro RC e Conversão Digital–Analógica

3.1 Montagem

Para a montagem do passa baixa, usamos um resistor de 9.9 kΩ e um capacitor de 101.5 μF

Logo, usando a formula do passa-baixa, temos que a frequência de corte é:

$$f_c = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 9.9 \cdot 10^3 \cdot 101.5 \cdot 10^{-9}} = 158Hz$$

Como o valor calculado é muito baixo, utilizamos um resistor de 1 kΩ, onde obtemos uma frequência de corte nova no valor de 1.6 k Hz, permitindo que a primeira harmônica não seja descartada pelo filtro, garantindo uma saída mais nítida.

Para o código, fizemos algumas pequenas modificações na `main`:

```
while (true) {
    duty += 0.1;
    if (duty > 1.09)
        duty = 0;

    configure_pwm(slice_num, channel, freq, duty);

    button_controller(GPIO16, &status_button, &last_state, 1, 0);

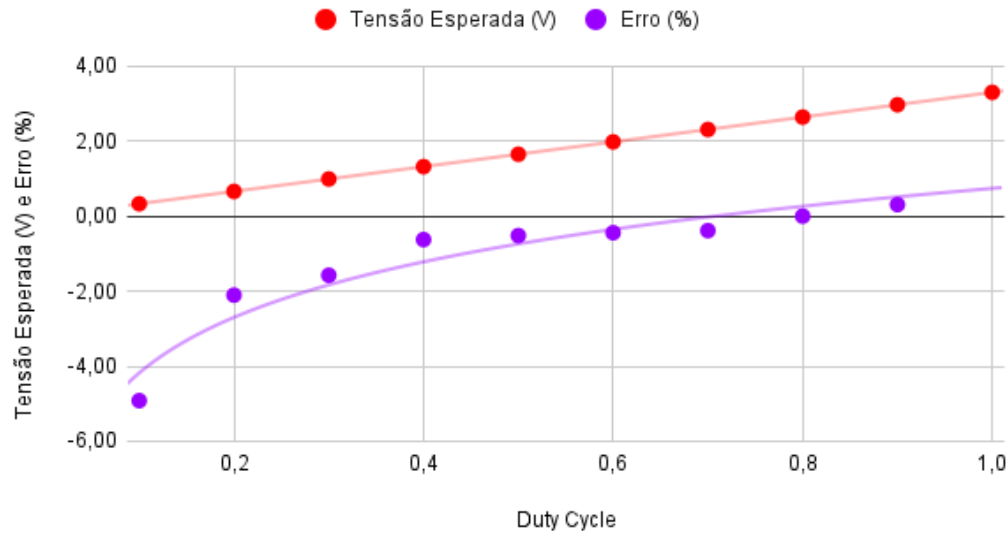
    static absolute_time_t last_print_time = {0};
    if (absolute_time_diff_us(last_print_time, get_absolute_time()) >= 1000000) {
        printf("Duty Cycle: %.2f, Frequency: %d, GPIO Frequency: %d\n", duty, freq, status_button - counter);
        last_print_time = get_absolute_time();
        counter = status_button;
    }
    sleep_ms(3000);
}
```

Os valores medidos e esperados de tensão para cada duty se encontram na seguinte tabela, para 100uF, 1k Ω e 1000Khz.

Table 1: Relação entre Duty Cycle, Tensões Medidas, Tensão Esperada e Erro

Duty Cycle	Tensão de Entrada Osiloscópio (V)	Vc Medido (V)	Tensão Esperada (V)	Erro (%)
0.1	0.280	0.290	0.33	-14.29
0.2	0.610	0.630	0.66	-4.92
0.3	0.945	0.970	0.99	-2.11
0.4	1.27	1.30	1.32	-1.57
0.5	1.6	1.64	1.65	-0.63
0.6	1.93	1.97	1.98	-0.52
0.7	2.26	2.30	2.31	-0.44
0.8	2.59	2.63	2.64	-0.39
0.9	2.92	2.97	2.97	0.00
1.0	3.25	3.31	3.30	0.31

Tensão Esperada (V) e Erro(%) por Duty Cycle



O circuito usado para essas medições foi o que se encontra na Figura 3.

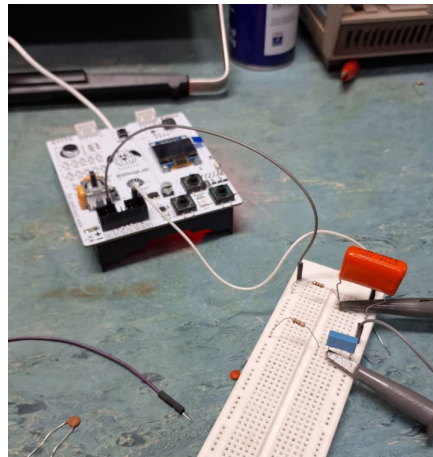


Figure 3: Circuito utilizado para a conversão digital-analógica via filtro RC, composto por resistor de 1 kΩ e capacitor de 100 μ F, conectado à saída PWM do Raspberry Pi Pico.

4 Conclusão e Comentários

Os resultados obtidos confirmam a relação teórica entre o ciclo de trabalho do sinal PWM e a tensão média medida após a filtragem pelo circuito RC. Observamos que, para baixos valores de duty cycle, o erro relativo foi mais significativo, chegando a cerca de -14%, mas rapidamente se estabilizou com valores próximos a zero para duty cycles acima de 0,5. Isso se explica pela limitação prática do filtro RC, que apresenta uma resposta mais imprecisa em tensões muito baixas devido ao tempo de carga e descarga do capacitor.

Outro ponto relevante é que a frequência de corte calculada (1,6k Hz) se mostrou adequada para suavizar o sinal PWM de 1k Hz, aproximando-o de uma tensão contínua. Ainda assim, pequenas ondulações (ripple) podem ser observadas, especialmente em duty cycles intermediários, evidenciando que um filtro de ordem maior poderia reduzir ainda mais esse efeito.

Adicionalmente, verificamos que ao diminuir a capacitância do filtro RC, o ripple observado no sinal de saída tornou-se mais "íngreme" e visível no osciloscópio. Esse comportamento está de acordo com a teoria: quanto menor a capacitância, menor a constante de tempo $\tau = RC$, e portanto menor a capacidade do capacitor de armazenar carga entre os pulsos do PWM. Como consequência, o capacitor se carrega e descarrega mais rapidamente, resultando em uma variação de tensão mais acentuada entre os ciclos, o que aumenta o ripple. Esse efeito de carga e descarga pode ser observado na Figura 4, onde mantivemos o mesmo circuito variando a capacitância de 100 nF (linha amarela) para 1 μ F (linha verde), para isso mantivemos a menor frequência de corte entre as duas frequências de corte. Além disso, subimos a frequência para ser

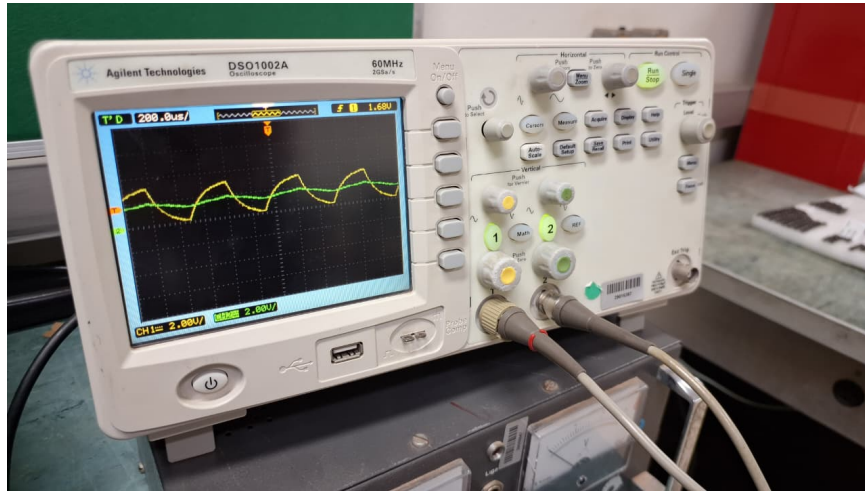


Figure 5: Forma de onda observada na saída do filtro RC com capacitor de $1\ \mu\text{F}$, utilizando uma frequência de PWM acima da frequência de corte. O sinal apresenta menor ripple, aproximando-se de uma tensão contínua.

superior à frequência de corte do circuito com o capacitor de $1\ \mu\text{F}$ e o resultado pode ser observado na Figura 5, mantendo em ambas um resistor de $1\text{K}\Omega$.

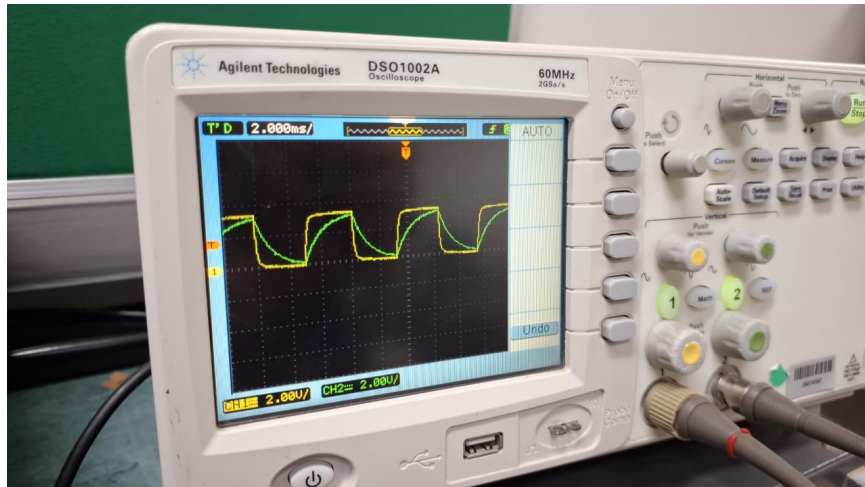


Figure 4: Comparação do ripple na saída do filtro RC para capacitâncias de $100\ \text{nF}$ (linha amarela) e $1\ \mu\text{F}$ (linha verde), mantendo o mesmo circuito e frequência de corte. Observa-se que o ripple é mais acentuado para o capacitor de menor valor.

Além disso, o controle 1 via joystick se mostrou funcional, permitindo variar o duty cycle do PWM e observar diretamente o impacto no valor médio da tensão de saída. Esse tipo de implementação ilustra bem a transição entre o domínio digital e o analógico, destacando a importância dos conversores e filtros em sistemas embarcados.

Comentários finais

- O uso do BitDogLab simplificou a prototipagem, permitindo integrar rapidamente leitura analógica, geração de PWM e medição dos resultados.
- O osciloscópio foi essencial para validar o comportamento real do sinal, mostrando que, apesar da teoria prever valores ideais, fatores como a tolerância dos componentes e o ripple afetam os resultados.
- A análise do efeito da capacitância no filtro evidenciou a importância de escolher valores adequados para reduzir o ripple e aproximar o sinal do valor médio esperado.
- Como sugestão de melhoria, poderia ser testado um filtro ativo ou de segunda ordem, a fim de obter uma aproximação ainda mais fiel entre o valor esperado e o medido.

Em conclusão, o experimento demonstrou com clareza a conversão de sinais digitais em analógicos por meio de PWM e filtragem RC, reforçando conceitos fundamentais de eletrônica e sistemas embarcados.