MICROPROCESSADORES
(TEORIA E PRÁTICA)

Edição
Julho/2024

Explicação do Código DAC (fake)

1. **Configuração dos LEDs**:
    - O código configura três pinos para LEDs: **vermelho**, **verde** e **amarelo**.
    - Dependendo da faixa de temperatura:
        - **0 a 26°C**: O LED verde acende.
        - **26,1 a 30°C**: O LED amarelo acende.
        - **Acima de 30°C**: O LED vermelho acende.
2. **Funções digitalWrite()**:
    - Cada LED é controlado pela função `digitalWrite()`, que liga ou desliga os LEDs conforme a temperatura lida.

Resultados Esperados:

- À medida que você ajusta o potenciômetro (ou altera a entrada do sensor de temperatura), a cor do LED correspondente deve mudar de acordo com as faixas de temperatura definidas:
    - **0 a 26°C**: Verde
    - **26,1 a 30°C**: Amarelo
    - **Acima de 30°C**: Vermelho

Razões para Usar Resistores Individuais

1. **Limitação de Corrente**:
    - Cada LED tem uma corrente nominal que não deve ser excedida (geralmente em torno de 20 mA para LEDs comuns). O resistor limita a corrente que passa por cada LED, evitando que ele queime.
2. **Variação de Tensão e Corrente**:
    - Quando você conecta LEDs em paralelo com um único resistor, a corrente não se divide igualmente entre os LEDs, o que pode resultar em um LED recebendo mais corrente do que o outro, levando a queimar um deles ou a uma intensidade desigual.
3. **Confiabilidade**:
    - Usar resistores separados para cada LED garante que se um LED falhar (por exemplo, se ele se queimar), os outros LEDs continuarão funcionando corretamente.

Conexão dos LEDs com Resistores

- Para cada LED, você deve conectar:
  - **Ânodo** (perna longa) do LED ao pino de controle do ESP32.
  - **Catodo** (perna curta) do LED ao resistor (220 ohms).
  - O outro terminal do resistor ao **GND**.

Montagem do Circuito no Wokwi:

1. **LED Verde**:
   - Conecte o ânodo ao **GPIO 25**.
   - Conecte o catodo ao resistor de 220Ω.
   - Conecte o outro terminal do resistor ao **GND**.
2. **LED Amarelo**:
   - Conecte o ânodo ao **GPIO 26**.
   - Conecte o catodo ao resistor de 220Ω.
   - Conecte o outro terminal do resistor ao **GND**.
3. **LED Vermelho**:
   - Conecte o ânodo ao **GPIO 27**.
   - Conecte o catodo ao resistor de 220Ω.
   - Conecte o outro terminal do resistor ao **GND**.

No exemplo que estamos utilizando, os pinos **25**, **26**, e **27 não estão sendo usados como saídas DAC**, mas sim como **saídas digitais** para controlar LEDs. Esses pinos estão configurados para fornecer **sinais digitais** (HIGH ou LOW) usando a função `digitalWrite()` para ligar ou desligar os LEDs, e não estão realizando uma conversão digital-analógica (DAC).

Diferença entre Saída Digital e Saída DAC

1. **Saída Digital**:
    - A saída digital só tem dois estados: **HIGH** (3.3V ou 5V, dependendo do sistema) ou **LOW** (0V).
    - Estamos usando `digitalWrite()` para controlar os LEDs, ou seja, ligando-os (HIGH) ou desligando-os (LOW).

2. **Saída DAC**:
    - A saída DAC (Conversor Digital-Analógico) converte um valor digital em um sinal analógico, que pode variar dentro de uma faixa contínua de valores (por exemplo, 0V a 3.3V).
    - O ESP32 possui dois pinos que podem funcionar como DAC: **GPIO 25** e **GPIO 26**.
    - Para usar a funcionalidade DAC, você utilizaria funções como `dacWrite()` para gerar um sinal analógico nesses pinos.

Pinos DAC no ESP32

O ESP32 tem apenas **dois pinos** que podem ser usados como DAC:

1. **GPIO 25 (DAC1)**
2. **GPIO 26 (DAC2)**

Esses pinos podem ser usados para gerar sinais analógicos diretamente.

Se Quiser Usar GPIO 25 e 26 como Saída DAC

Se você quiser usar os pinos **25** e **26** como saídas DAC para gerar sinais analógicos em vez de sinais digitais, pode fazer algo como o seguinte:

```
int dacValue = map(temperature, 0, 100, 0, 255);  // Mapeia a temperatura para 0-255
dacWrite(25, dacValue);  // Saída DAC no GPIO 25
```

Bibliografia:

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html
https://randomnerdtutorials.com/
http://www.practical-arduino.com/