

Professor

Massaki de Oliveira Igarashi

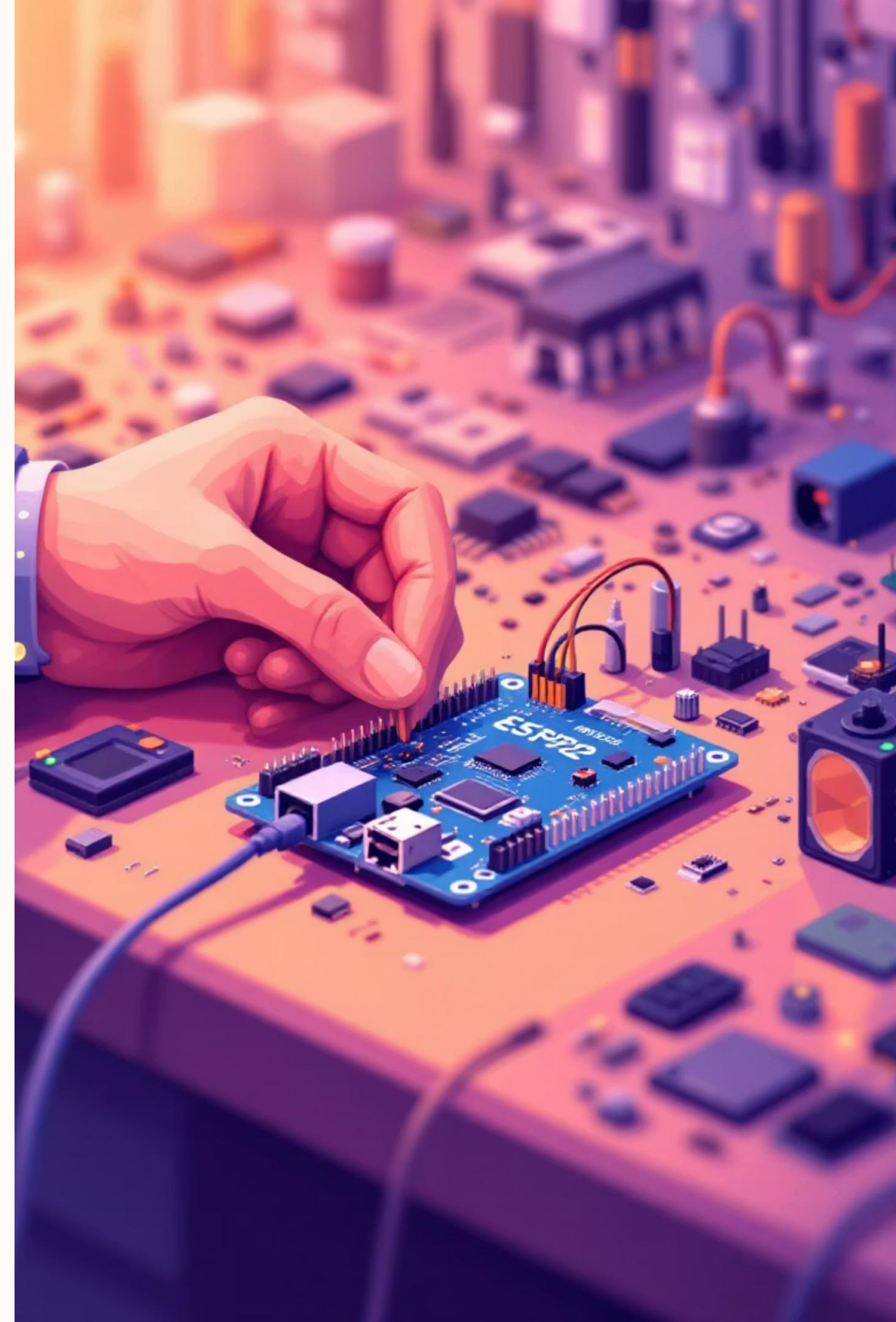
Soluções Integradas com IoT

Dia 3 de 8



Leitura Analógica com ESP32 usando Arduino IDE

Nesta apresentação, exploraremos o uso do Conversor Analógico-Digital (ADC) do ESP32 com a familiar interface da Arduino IDE. Entenda como transformar sinais do mundo real em dados digitais para seus projetos de IoT.



O que é ADC no ESP32?



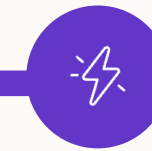
Medição de Tensão

O ADC (Conversor Analógico-Digital) permite medir tensões elétricas, idealmente na faixa de 0 a 3,3 V.



Digitalização de Sinais

Ele converte esses sinais analógicos em valores digitais, variando de 0 a 4095, com 12 bits de resolução.



Para Sensores Analógicos

É fundamental para integrar sensores analógicos, como potenciômetros, termistores e sensores de luz, que fornecem saídas de tensão variáveis.

Como usar a função `analogRead()`

Sintaxe Simples

A função `analogRead(GPIO)` é direta e fácil de usar, lendo o valor analógico do pino GPIO especificado.

Canais ADC do ESP32

O ESP32 possui 18 canais ADC no total, com 15 tipicamente disponíveis no DEVKIT V1 para suas aplicações.

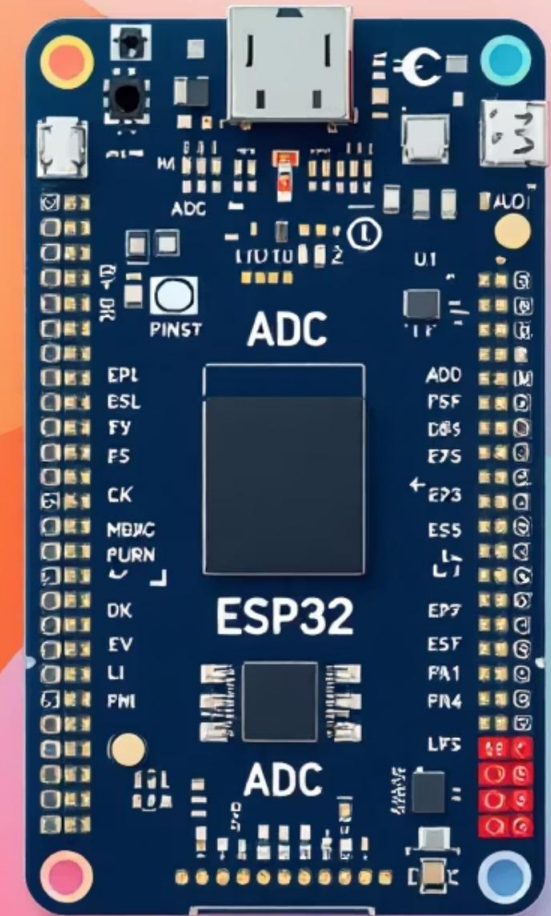
Atenção aos Pinos ADC2

- ❌ Pinos ADC2 não funcionam quando o Wi-Fi está ativo. Para projetos que utilizam Wi-Fi, priorize os pinos ADC1 para garantir leituras consistentes.

Essa limitação é importante para evitar problemas de compatibilidade e garantir o funcionamento correto do seu projeto.

Localize os Pinos ADC

no seu ESP32 para leituras confiáveis.



Comportamento não linear do ADC do ESP32

Atenção aos Limites

Valores próximos aos extremos da faixa de leitura (0V e 3.3V) podem apresentar comportamentos inesperados. Por exemplo, tanto 3.2V quanto 3.3V podem retornar o valor máximo de 4095. Similarmente, 0V e 0.1V podem ambos resultar em 0.

Essa não linearidade é uma característica do hardware e deve ser **considerada durante o desenvolvimento**. Recomenda-se realizar uma calibração para garantir a precisão das leituras, mapeando os valores digitais para as tensões reais correspondentes. Ignorar essa característica pode levar a imprecisões significativas, especialmente em aplicações que exigem alta sensibilidade.

Configurações avançadas do ADC

Resolução (bits)

`analogSetWidth(bits)`: Define a resolução da leitura analógica entre 9 e 12 bits (o padrão é 12). Uma resolução maior oferece mais detalhes na medição.



Atenuação (faixa de tensão)

`analogSetAttenuation(ADC_0db a ADC_11db)`: Ajusta a faixa de tensão que o ADC pode medir. Use para expandir a capacidade de leitura até aproximadamente 3.3V com maior precisão.



Amostras (sensibilidade)

`analogSetSamples(samples)`: Aumenta o número de amostras tiradas para cada leitura, o que pode resultar em maior sensibilidade e estabilidade do sinal.

Exemplo prático: leitura de um potenciômetro

Para demonstrar a leitura analógica, vamos usar um potenciômetro, que é um excelente exemplo de sensor analógico que varia a resistência e, conseqüentemente, a tensão.

- Conecte um potenciômetro a um pino ADC1 do ESP32, por exemplo, o **GPIO 34**.
- O pino central do potenciômetro vai para o GPIO 34.
- As extremidades vão para 3.3V e GND, respectivamente.

O código para ler o valor é bastante direto, utilizando a função `analogRead()`.

```
valor = analogRead(34);
```

Para converter o valor lido (0-4095) de volta para uma tensão (0-3.3V), aplique uma regra de três simples:

```
tensão = valor * (3.3 / 4095);
```

Este exemplo ilustra a facilidade de integrar sensores analógicos com o ESP32.

Dicas para evitar problemas comuns

ADC2 e Wi-Fi

Evite usar os pinos ADC2 para leituras analógicas quando o Wi-Fi do ESP32 estiver ativo, pois isso pode causar interferências e resultados inconsistentes.

Resistores de Pull-Down

Para sinais analógicos, o uso de resistores de pull-down ajuda a estabilizar as leituras e evitar ruídos, garantindo um valor conhecido quando o sensor não está conectado ou está em estado flutuante.

Calibração de Sensores

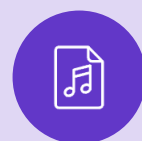
Devido à não linearidade inerente do ADC do ESP32, calibre seus sensores. Isso garante que os valores lidos digitalmente correspondam com precisão às condições físicas reais que você está medindo.

Aplicações reais do ADC no ESP32



Monitoramento Ambiental

Sensores de temperatura, umidade e luz para casas inteligentes e agricultura.



Controle de Dispositivos

Ajuste de volume em sistemas de áudio ou brilho em iluminação ambiente.



Projetos de IoT

Coleta de dados analógicos precisos para análises e tomadas de decisão em tempo real.

Conclusão: Potencial do ESP32 para projetos analógicos

Poder e Precisão

O ADC do ESP32 é uma ferramenta poderosa, mas exige atenção aos detalhes para garantir a precisão das leituras.

Explore e Otimize

Não hesite em explorar as configurações avançadas para otimizar seus projetos e superar os desafios da não linearidade.

Facilidade com Arduino IDE

A Arduino IDE simplifica a programação e experimentação, tornando-o acessível para todos os níveis de conhecimento.

Comece Hoje!

Transforme sinais analógicos em dados digitais e dê vida às suas ideias com o ESP32. O futuro da IoT está em suas mãos!

Aplicações reais do ADC no ESP32

 **Atividade-Desafio Final: Sistema de Automação com Alerta de Emergência**

 **Título: Automação Residencial com Alerta de Segurança e Acionamento de Carga via Relé Eletromecânico**

 **Objetivo Geral:**

Desenvolver um sistema de automação onde o aluno:

- Reaproveita o pino físico do LED_BUILTIN da ESP32 para acionar um **relé eletromecânico**
- Utiliza o **App Inventor** para controlar essa saída remotamente
- Integra um **sensor de alarme (botão, LDR, fio de alarme rompido etc.)**
- Configura o ESP32 para **enviar um e-mail automático** quando o sensor for ativado

 **Competências Desenvolvidas:**

- Integração de sistemas embarcados com automação residencial
- Reutilização inteligente de pinos GPIO
- Lógica de monitoramento e resposta a eventos críticos
- Comunicação via Internet (HTTP + SMTP)
- Segurança em projetos com relés e sensores



Cenário Simulado:

Imagine que você está instalando um sistema de automação em uma casa. A lâmpada do jardim será ligada remotamente via aplicativo. No entanto, caso o **sensor da cerca de segurança** detecte uma violação (fio rompido), o sistema deve:

- Acionar o alarme (simulado ou real) Enviar um **e-mail de emergência automaticamente**

Aplicações reais do ADC no ESP32

🔌 Esquema de Conexão (Resumo):

➤ Controle do relé (via GPIO do LED_BUILTIN, normalmente GPIO 2):

- GPIO 2 → 1kΩ → base do transistor
- Transistor NPN → relé → +5V
- Diodo 1N4007 em paralelo com bobina
- Contatos do relé → carga

➤ Sensor de Alarme:

- Sensor normalmente fechado (NF) → entre GPIO 4 e GND
- Ativado quando **o fio é rompido ou aberto**
- Pode usar resistor pull-up interno do ESP32

📦 Materiais sugeridos:

Item	Quantidade
ESP32 DOIT V1	1
Relé eletromecânico 10A	1
Transistor NPN	1
Diodo 1N4007	1
Resistor de base (1kΩ)	1
Sensor de alarme (ex: botão, fio, reed switch)	1
Carga (ex: lâmpada 12V ou AC)	1
Protoboard, fios	diversos
Fonte 5V (se necessário)	1