

## Lista de Exercícios

1-Como funciona o processo de ADC, que converte um sinal contínuo (analógico) para um sinal digital?

Incluir o circuito ADC na explicação;

O **ADC (Analog-to-Digital Converter)** é o circuito responsável por converter um sinal **analógico contínuo** em um **sinal digital discreto**, permitindo que sistemas digitais processem informações do mundo real.

O processo de conversão analógico–digital ocorre em **três etapas principais**:

### 1) Amostragem

O sinal analógico é medido em instantes regulares de tempo, definidos pela frequência de amostragem. Para que o sinal seja corretamente reconstruído, a frequência de amostragem deve obedecer ao **Teorema de Nyquist**, ou seja, ser pelo menos o dobro da maior frequência presente no sinal.

Nessa etapa, é utilizado um circuito **Sample and Hold**, formado por uma chave eletrônica e um capacitor, cuja função é capturar e manter a tensão constante durante o processo de conversão.

### 2) Quantização

Cada amostra obtida é aproximada para o nível digital mais próximo. O número de níveis disponíveis depende da **resolução do ADC**, expressa em bits. Por exemplo, um ADC de 8 bits possui 256 níveis de quantização. Esse processo introduz o chamado **erro de quantização**, inerente à conversão.

### 3) Codificação

O valor quantizado é então representado em forma de **número binário**, gerando a saída digital do ADC.

Um dos circuitos mais utilizados é o **ADC de Aproximação Sucessiva (SAR)**, composto por um comparador, um DAC interno, um registrador SAR e um clock. O DAC interno gera tensões de referência que são comparadas com o sinal de entrada, permitindo que o valor digital seja determinado bit a bit, do mais significativo (MSB) ao menos significativo (LSB).

2-Como funciona o processo de DAC, que converte um sinal digital para um sinal contínuo (analógico)?

Incluir o circuito DAC na explicação;

Destacar o funcionamento do PWM (Modulação por Largura de Pulso);

O **DAC (Digital-to-Analog Converter)** realiza o processo inverso ao ADC, convertendo um **sinal digital** em um **sinal analógico contínuo**, geralmente na forma de tensão ou corrente.

O funcionamento do DAC baseia-se na contribuição ponderada de cada bit do valor digital de entrada. Os bits mais significativos possuem maior peso na tensão de saída, enquanto os menos significativos possuem menor influência.

Um circuito bastante comum é o **DAC por rede R-2R**, que utiliza apenas dois valores de resistores ( $R$  e  $2R$ ), chaves eletrônicas e um amplificador operacional. Cada bit conecta sua respectiva rede a uma tensão de referência ou ao terra, e a soma das correntes geradas resulta em uma tensão analógica proporcional ao valor digital aplicado.

Além dos DACs convencionais, destaca-se o **PWM (Modulação por Largura de Pulso)**, que não é um DAC verdadeiro, mas é amplamente utilizado para gerar sinais analógicos de forma simples. No PWM, o sinal permanece digital, alternando entre nível alto e baixo, porém a informação está na **largura do pulso**, chamada de **duty cycle**. Quanto maior o duty cycle, maior será a tensão média do sinal.

Quando a saída PWM é aplicada a um **filtro passa-baixa**, normalmente um circuito RC, o sinal pulsado é suavizado, resultando em uma tensão praticamente contínua. Essa técnica é muito utilizada em microcontroladores devido ao baixo custo e simplicidade, embora apresente limitações como ripple e menor precisão quando comparada a DACs dedicados.

Como funciona o processo de ADC, que converte um sinal contínuo (analógico) para um sinal digital? Incluir o circuito ADC na explicação; Como funciona o processo de DAC, que converte um sinal digital para um sinal contínuo (analógico)? Incluir o circuito DAC na explicação; Destacar o funcionamento do PWM (Modulação por Largura de Pulso);

O ADC (Conversor Analógico-Digital) converte sinais analógicos (como uma tensão contínua) em valores digitais binários. Ele faz isso amostrando o sinal em momentos regulares e comparando-o com níveis de referência. Um circuito ADC comum é o tipo SAR (Aproximação Sucessiva), que usa um comparador e um DAC interno para ajustar bit a bit o valor digital equivalente à entrada. Já o DAC (Conversor Digital-Analógico) realiza o processo inverso, gerando uma tensão contínua proporcional ao valor binário recebido. Um circuito típico é o DAC R-2R, que utiliza uma escada de resistores e chaves digitais para somar tensões em proporção binária. O PWM (Modulação por Largura de Pulso), por sua vez, não converte diretamente valores, mas sim controla a potência média enviada a uma carga, variando o tempo em que a saída digital permanece ligada (duty cycle). Com um filtro, o PWM pode simular uma saída analógica. Em sistemas embarcados, ADCs são usados para ler sensores, DACs para controlar atuadores analógicos, e PWM para regular potência de LEDs, motores ou tensão de forma digital e eficiente.

3-Desenvolva uma aplicação em que um sensor (input) tenha seus valores monitorados por alguma ferramenta serial (monitor ou plotter). Pode usar qualquer IDE para embarcar em placa física OU simulador;

Nesta aplicação, um **sensor analógico** (exemplo: potenciômetro ou sensor de temperatura LM35) é utilizado como **entrada**, e seus valores são monitorados em tempo real por meio do **Serial Monitor** ou **Serial Plotter** da IDE Arduino.

### Descrição do funcionamento

O sensor é conectado a uma entrada analógica do microcontrolador. O ADC interno do Arduino converte a tensão analógica (0 a 5 V) em um valor digital entre 0 e 1023. Esse valor é lido periodicamente e enviado pela comunicação serial para visualização no computador.

### Hardware utilizado

- Arduino Uno
- Sensor analógico (potenciômetro ou LM35)
- Conexão USB

Código:

```
const int sensorPin = A0;
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  int valorSensor = analogRead(sensorPin);  
  Serial.println(valorSensor);  
  delay(200);  
}
```

4-Desenvolva uma aplicação (pode ser continuação da questão 2 desta lista) em que o funcionamento de um atuador seja influenciado pelo sinal lido a partir de um sensor.

Nesta aplicação, o funcionamento de um **atuador** (LED ou motor) é influenciado diretamente pelo valor lido de um **sensor analógico**, caracterizando um sistema de controle simples.

### Descrição do funcionamento

O valor do sensor é lido pelo ADC do microcontrolador e utilizado para controlar a intensidade de um LED por meio de **PWM**. Quanto maior o valor lido no sensor, maior será o duty cycle do PWM e, conseqüentemente, maior o brilho do LED.

## Hardware utilizado

- Arduino Uno
- Sensor analógico (potenciômetro)
- LED + resistor
- Saída PWM

Código:

```
const int sensorPin = A0;  
const int ledPin = 9;
```

```
void setup() {  
  pinMode(ledPin, OUTPUT);  
}
```

```
void loop() {  
  int valorSensor = analogRead(sensorPin);  
  int pwm = map(valorSensor, 0, 1023, 0, 255);  
  analogWrite(ledPin, pwm);  
}
```

5-Desenvolva uma aplicação (pode ser continuação da questão 2 desta lista) em que seja realizada armazenamento não volátil de informações. Pode ser usada a própria memória não volátil da placa (física ou simulador) ou utilizar arquivos (necessita de S.O.);

Nesta aplicação, os valores lidos de um sensor são armazenados em **memória não volátil**, garantindo que os dados sejam preservados mesmo após desligar a alimentação do sistema.

## Descrição do funcionamento

O valor do sensor é lido e armazenado na **EEPROM interna** do Arduino. Sempre que o sistema é reiniciado, o último valor salvo é recuperado e enviado pela comunicação serial.

## Hardware utilizado

- Arduino Uno
- Sensor analógico
- Memória EEPROM interna da placa

Código:

```
#include <EEPROM.h>
```

```
const int sensorPin = A0;
const int enderecoEEPROM = 0;

void setup() {
  Serial.begin(9600);

  int valorSalvo;
  EEPROM.get(enderecoEEPROM, valorSalvo);
  Serial.print("Último valor salvo: ");
  Serial.println(valorSalvo);
}

void loop() {
  int valorSensor = analogRead(sensorPin);
  EEPROM.put(enderecoEEPROM, valorSensor);

  Serial.print("Valor atual: ");
  Serial.println(valorSensor);

  delay(1000);
}
```