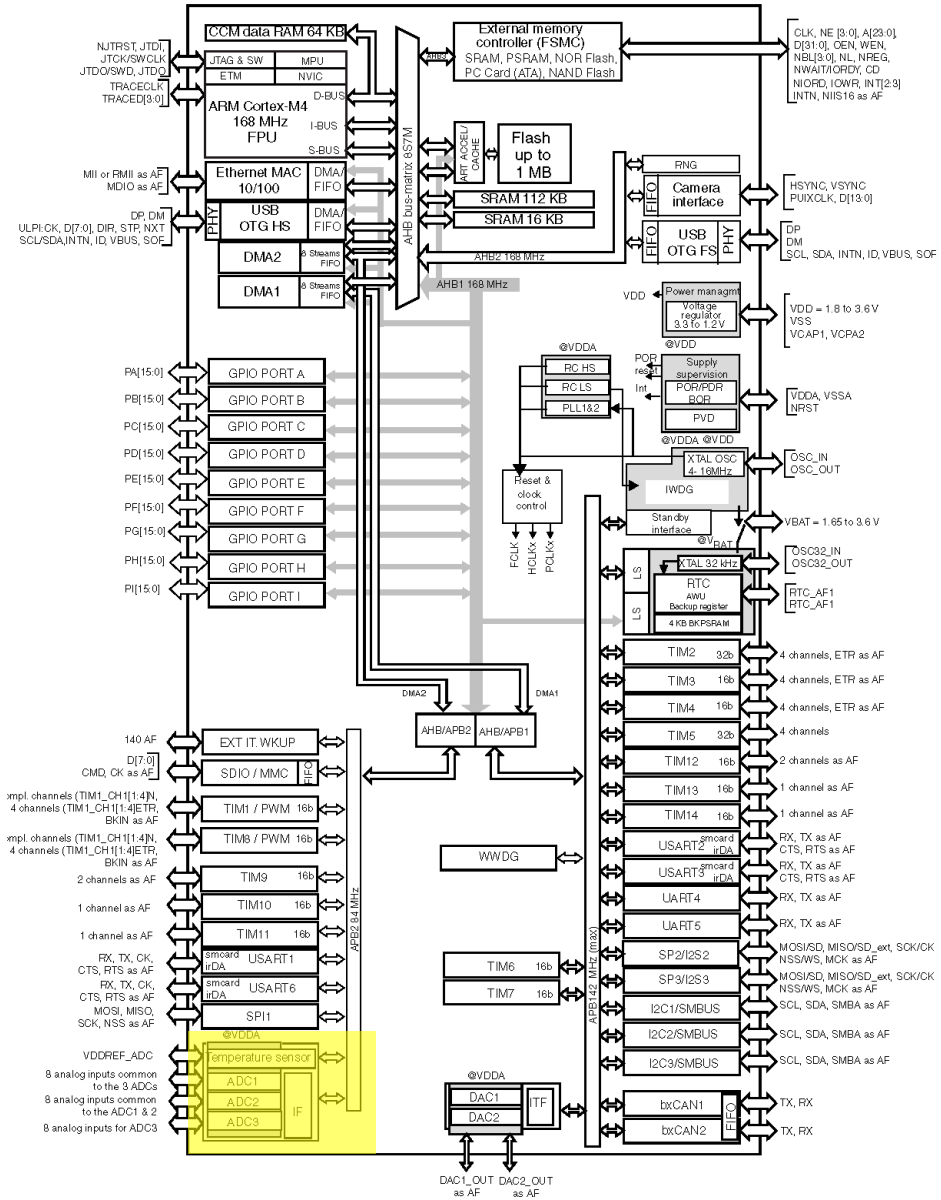
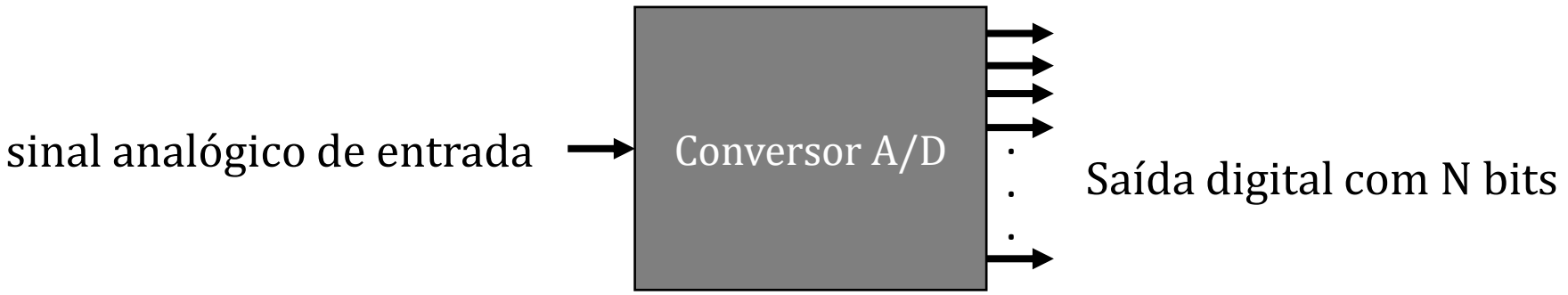
Conversor analógico-digital (AD) no STM32F407

Prof. Fagner de Araujo Pereira
fagner.pereira@ifpb.edu.br
fagnereng@gmail.com

Conversor AD (Analógico-Digital)



Conversor AD (Analógico-Digital)

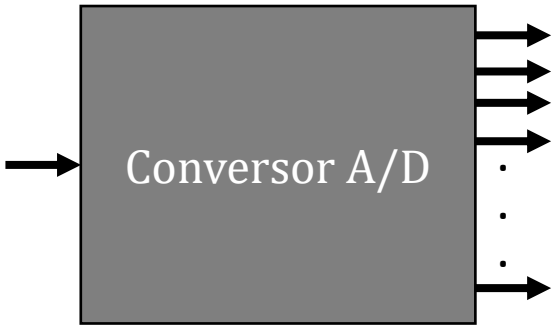


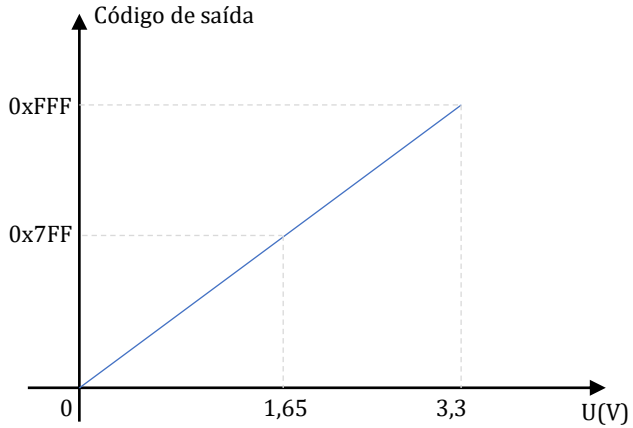
N – quantidade de bits da conversão (12 bits);

2^N – quantidade de códigos digitalizáveis (números codificados de 0 a 2^N-1) (4096 códigos);

Resolução = $\frac{faixa\ de\ tensão\ de\ entrada}{2^N - 1}$ (805 μV).

Conversor AD (Analógico-Digital)





relação entre tensão de entrada e código de saída no STM32F407

Conversor AD – tecnologias de implementação

Paralelo – flash;

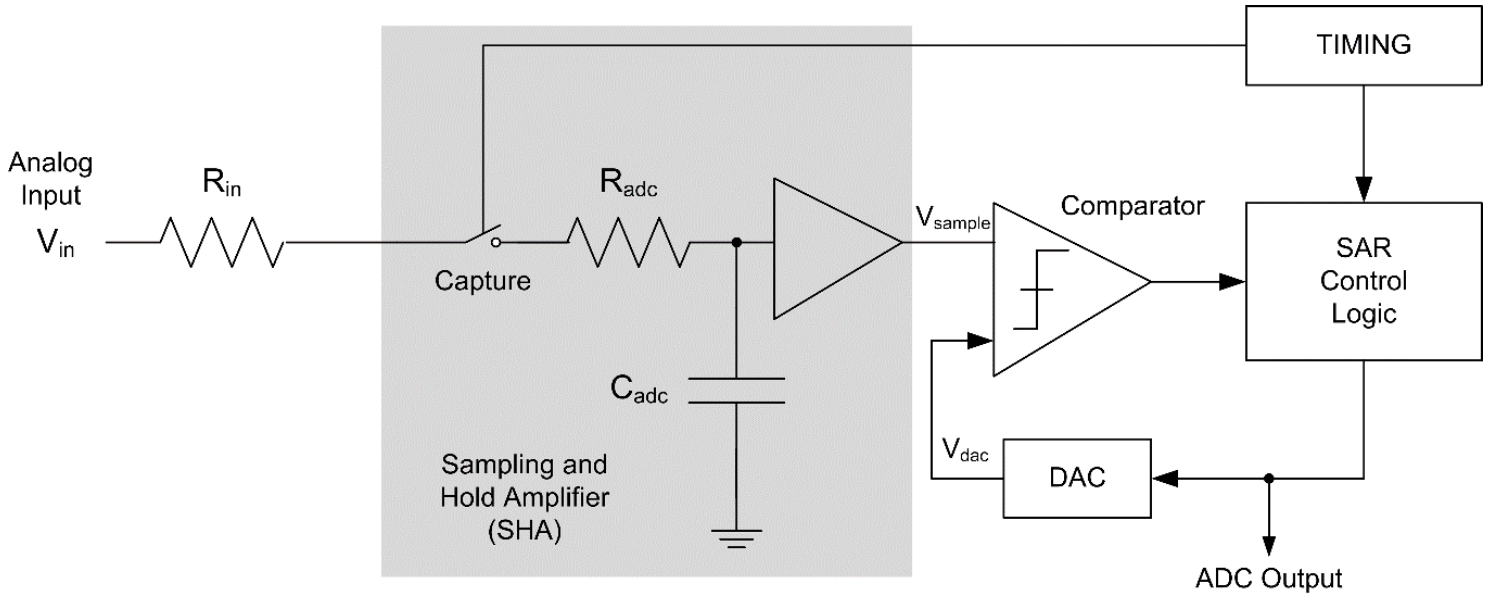
*Aproximações sucessivas;

Tipo contador;

Integrador simples e dupla rampa;

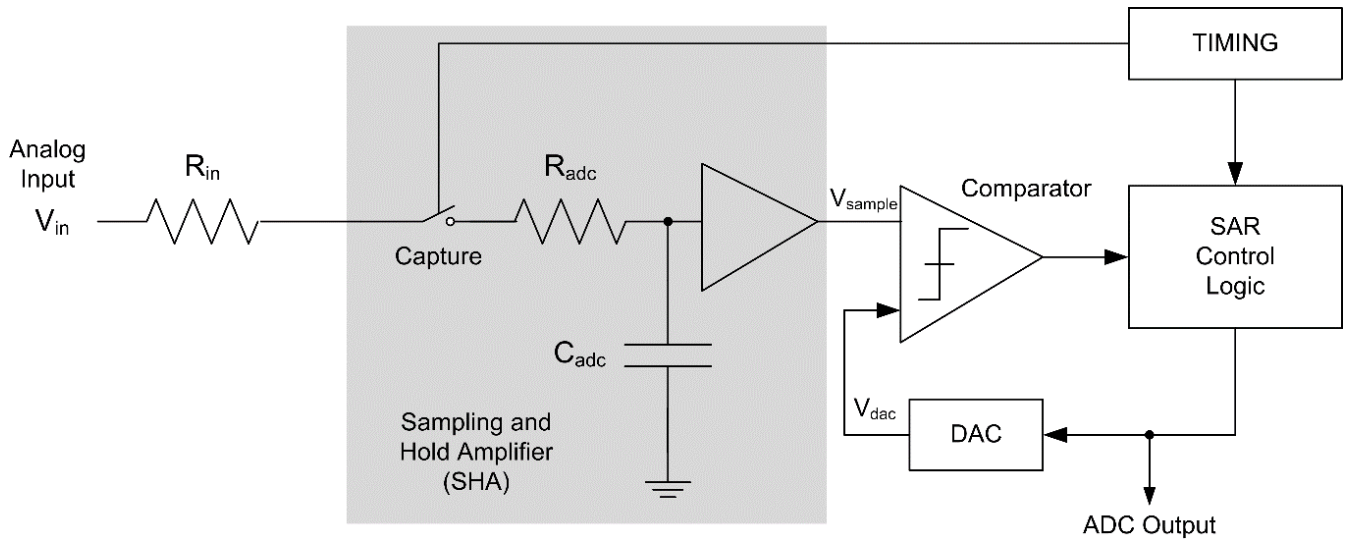
Redistribuição de cargas;

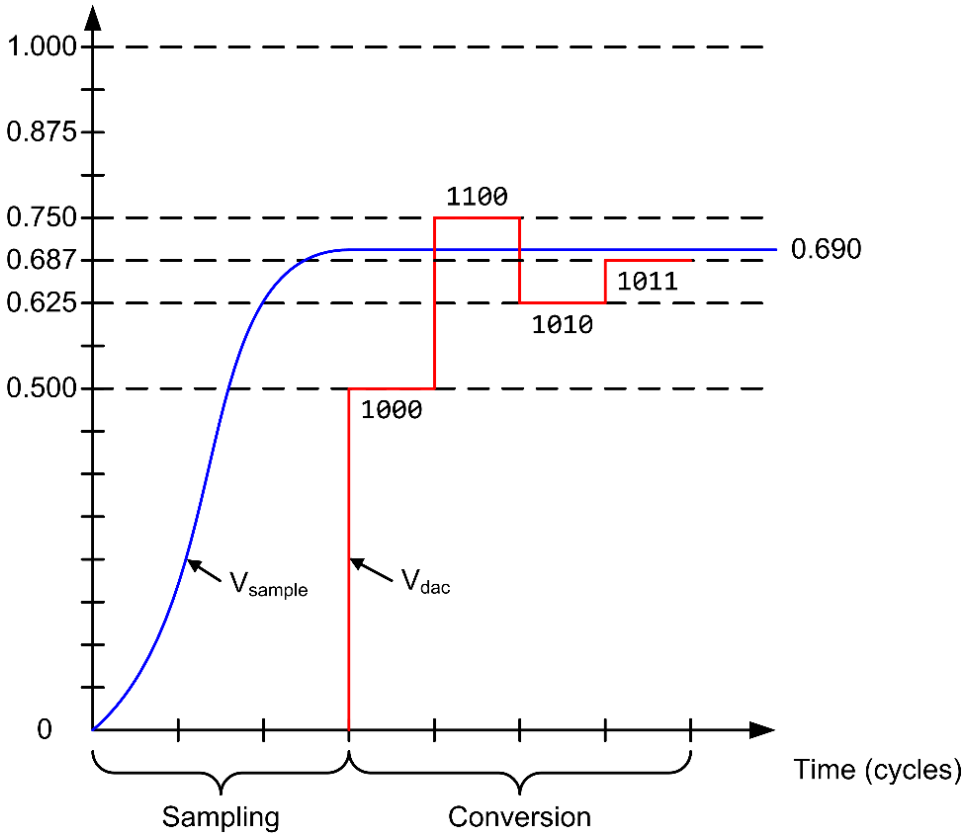
Sigma-delta.



Conversor AD – tecnologias de implementação

Exemplo de funcionamento durante a conversão





Conversor AD no STM32F407

* 3 conversores de 12 bits (ADC1, ADC2 e ADC3)
* 18 canais (16 fontes externas, 1 sensor de temperatura interna e o canal VBAT)
* Resolução configurável de 12, 10, 8 ou 6 bits
* Geração de interrupção no final da conversão
* Modos de conversão simples e contínuo;
* Modo de varredura para conversão automática do canal 0 até o canal N
* Tempo de amostragem programável por canal
* Modo de conversão simultânea em todos os conversores
* Atraso configurável entre conversões no modo simultâneo
* Gatilhos por software ou sinais oriundos, por exemplo, de temporizadores internos ou de pinos externos.

Conversor AD no STM32F407

* Para ligar o ADC, basta setar o bit ADON no registrador CR2 do módulo ADC. É possível colocar o ADC no modo de desligamento resetando o bit ADON. Nesse modo, o ADC consome quase nenhuma energia;
* Para iniciar uma conversão, basta setar o bit SWSTART no registrador CR2.

Esquemas de clock

* Clock para o circuito analógico, ADCCLK, comum a todos os ADCs. Este sinal é gerado a partir do clock do barramento APB2 dividido por um prescaler programável que permite que o ADC trabalhe em f_{APB2} /2, /4, /6 ou /8. O valor máximo de ADCCLK é de 36 MHz.
* Clock para a interface digital (para acesso aos registradores), igual ao clock do barramento APB2. O clock da interface digital pode ser ativado/desativado individualmente para cada ADC por meio dos bits ADC1EN, ADC2EN e ADC3EN no registrador APB2ENR do módulo RCC.

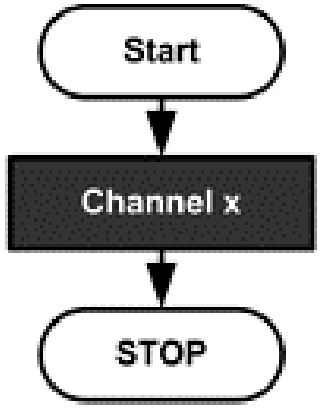
Conversor AD no STM32F407

Tabela 1. Definição dos pinos de entrada do ADC

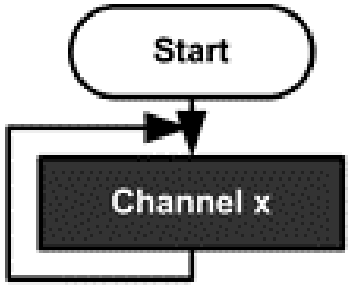
Canal de entrada analógico	Pino	Canal de entrada analógico	Pino
ADC123_IN0	PA0	ADC12_IN8	PB0
ADC123_IN1	PA1	ADC12_IN9	PB1
ADC123_IN2	PA2	ADC123_IN10	PC0
ADC123_IN3	PA3	ADC123_IN11	PC1
ADC12_IN4	PA4	ADC123_IN12	PC2
ADC12_IN5	PA5	ADC123_IN13	PC3
ADC12_IN6	PA6	ADC12_14	PC4
ADC12_IN7	PA7	ADC12_15	PC5

Conversor AD no STM32F407

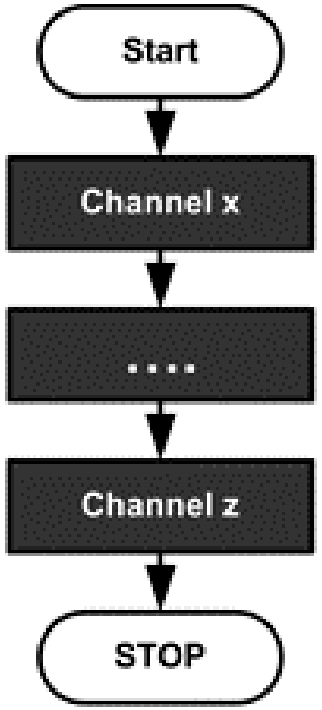
Modos de conversão



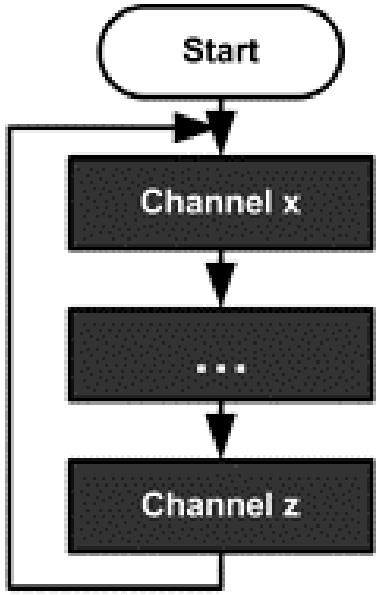
Single Channel, Single Conversion Mode



Single Channel, Continuous Conversion Mode



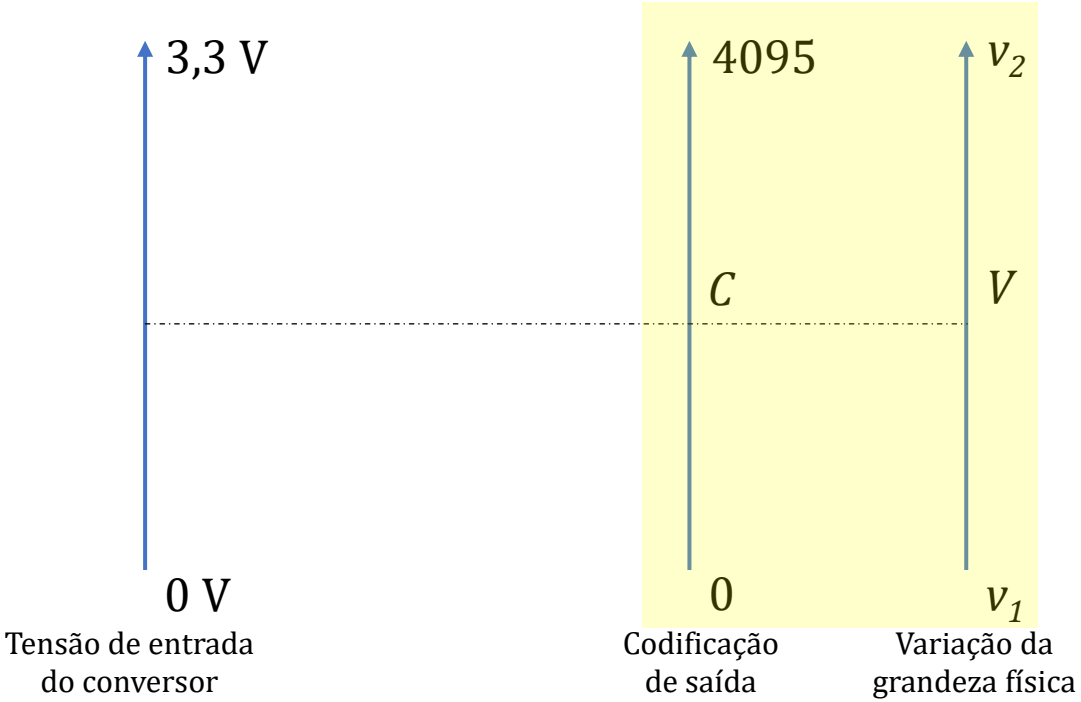
Scan Mode with Single Conversion



Scan Mode with Continuous Conversion

Conversor AD no STM32F407

Escalonamento das grandezas



$$V = \frac{C\ (v_2 - v_1)}{4095} + v_1$$

Conversor AD no STM32F407

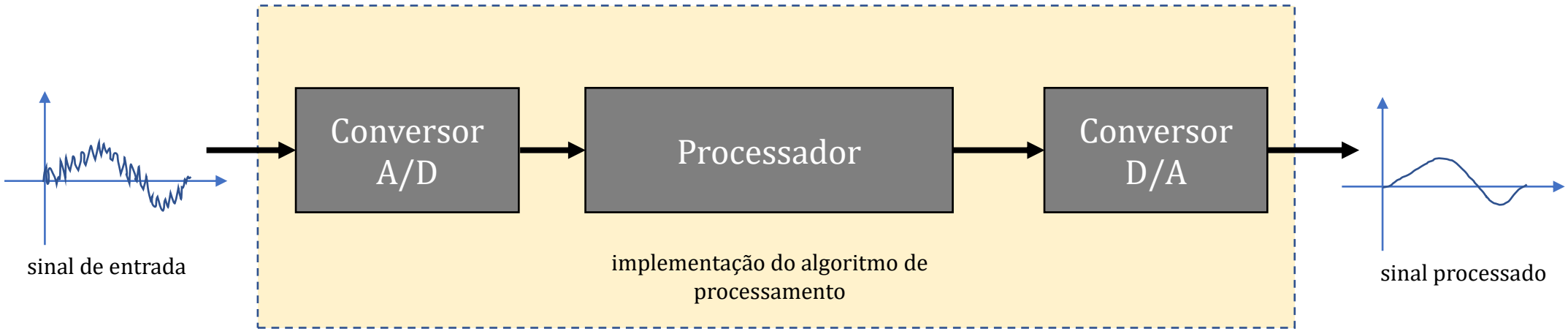
Exemplo de programação

```
void Configure_ADC()
{
  RCC->AHB1ENR |= 1;              //habilita o clock do GPIOA
  GPIOA->MODER |= 0b11;           //pino PA0 como entrada analógica

  RCC->APB2ENR |= 1 << 8;         //liga o clock da interface digital do ADC1

  ADC->CCR |= 0b01 << 16;         //prescaler /4 (f_ADC=21MHz)
  ADC1->SQR1 &= ~(0xF << 20);     //conversão de apenas um canal
  ADC1->SQR3 &= ~(0x1F);          //seleção do canal a ser convertido (IN_0)
  ADC1->CR2 |= 1;                 //liga o conversor AD
}
```

```
ADC1->CR2 |= 1 << 30;             //inicia a conversão
while(!(ADC1->SR & 0x02));        //aguarda o fim da conversão
uint16_t leitura = ADC1->DR;      //faz a leitura do valor convertido
```

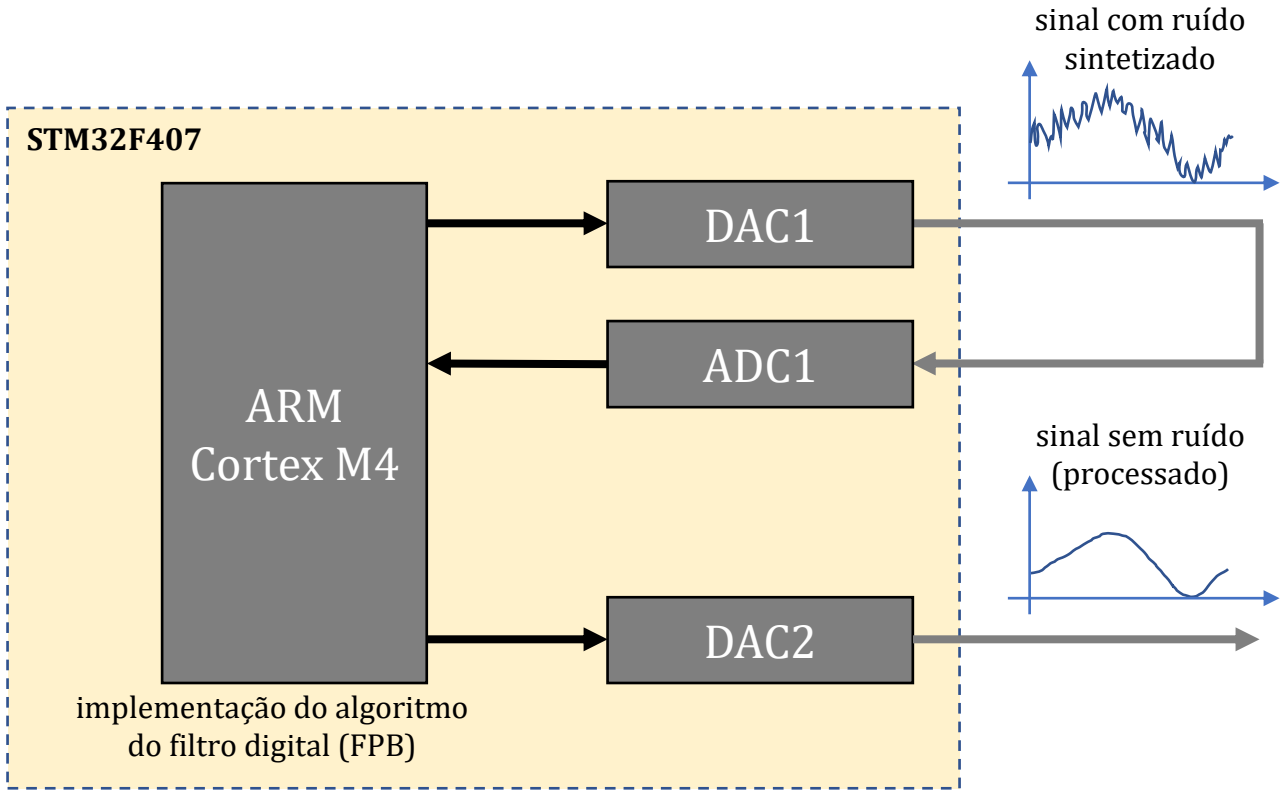
Conversor AD no STM32F407

Sistema de processamento digital de sinais



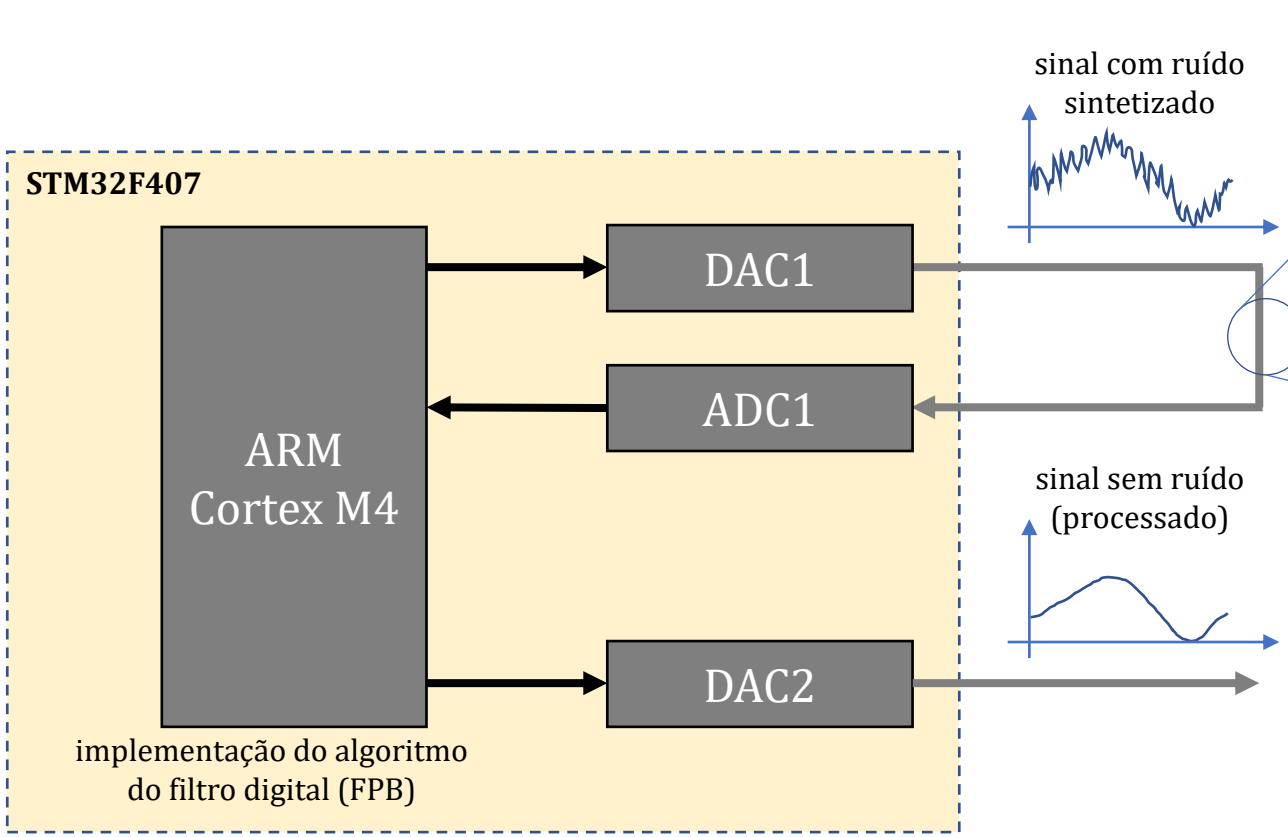
Conversor AD no STM32F407

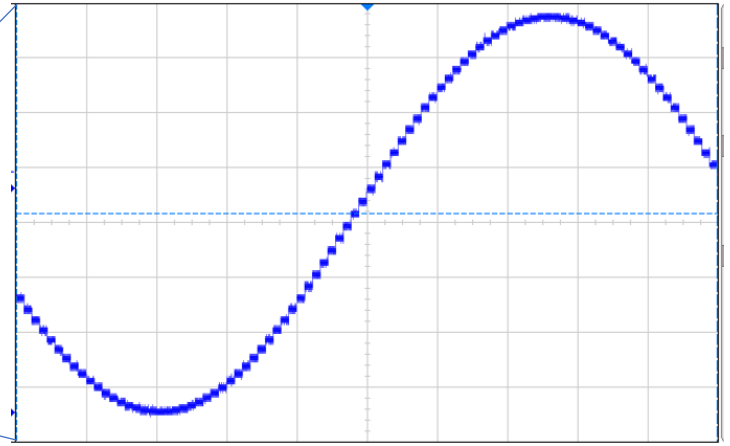
Exemplo 1: filtragem digital



Conversor AD no STM32F407

Exemplo 1: filtragem digital

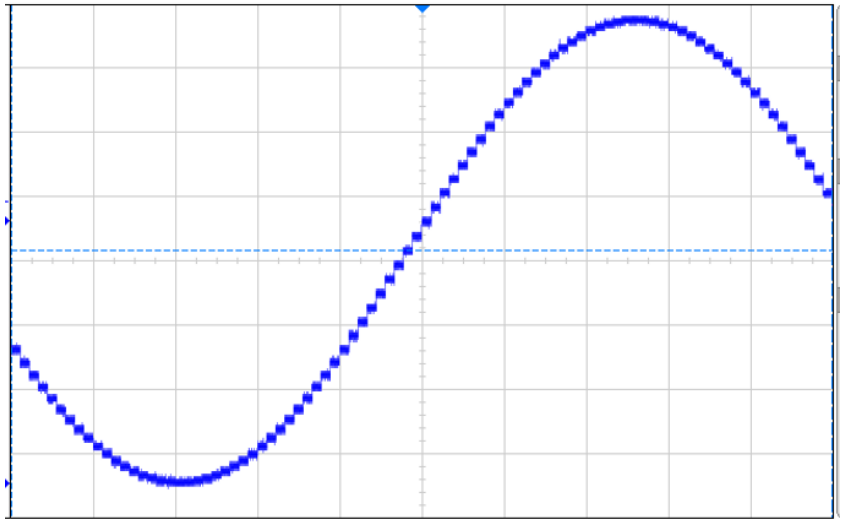




Sinal com forma de onda em degraus

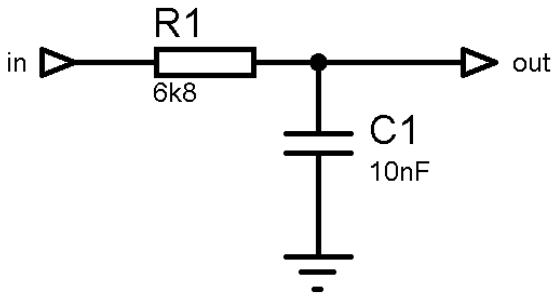
Conversor AD no STM32F407

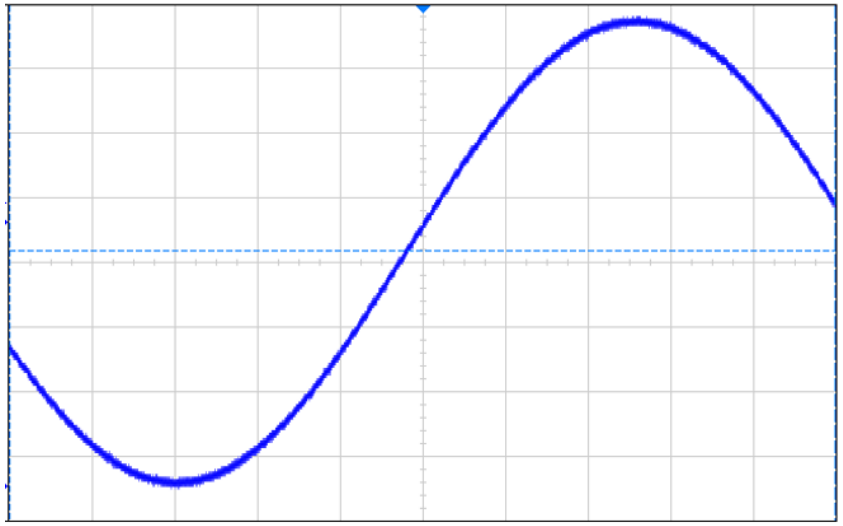
Exemplo 1: filtragem digital



Sinal com forma de onda em degraus

Filtro passa-baixas analógico

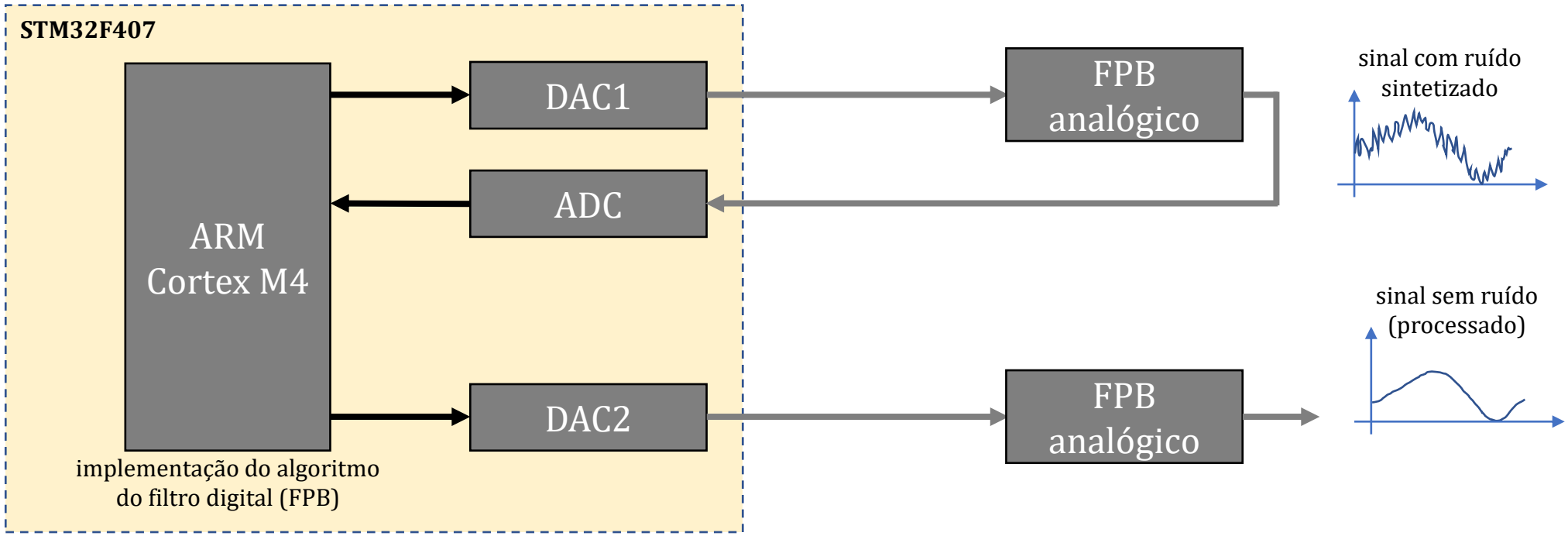




Sinal com forma de onda suavizada

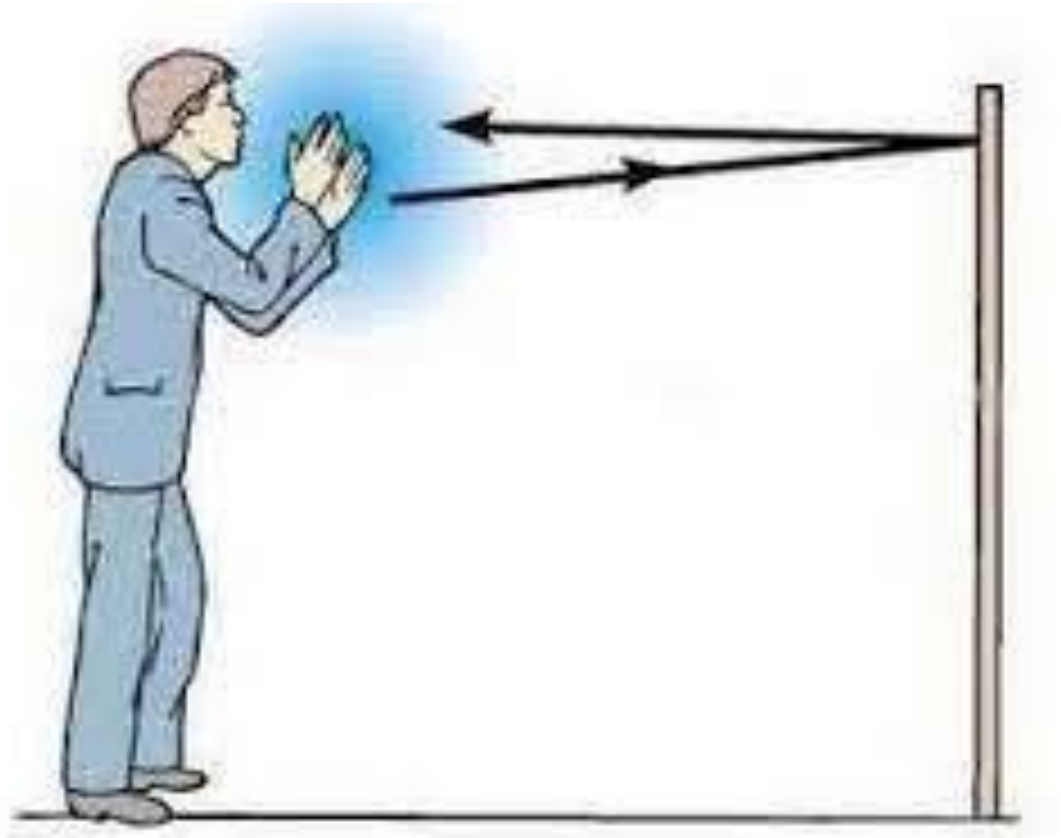
Conversor AD no STM32F407

Exemplo 1: filtragem digital



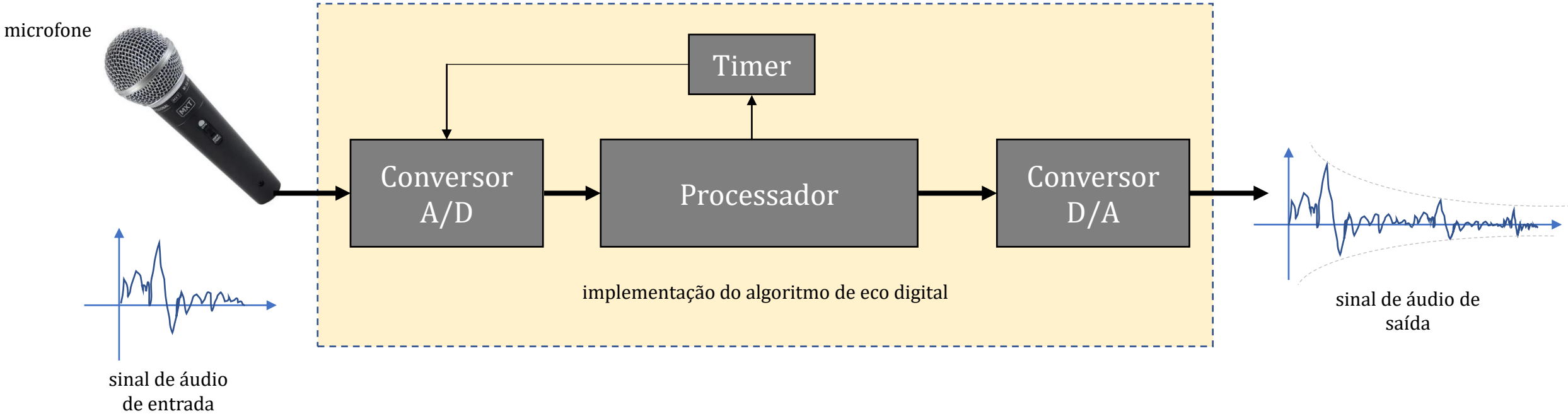
Conversor AD no STM32F407

Exemplo 2: Eco digital



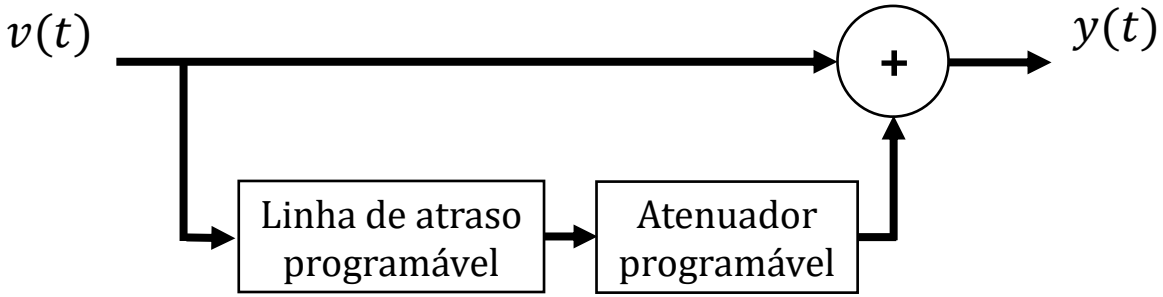
Conversor AD no STM32F407

Exemplo 2: Câmara de eco digital

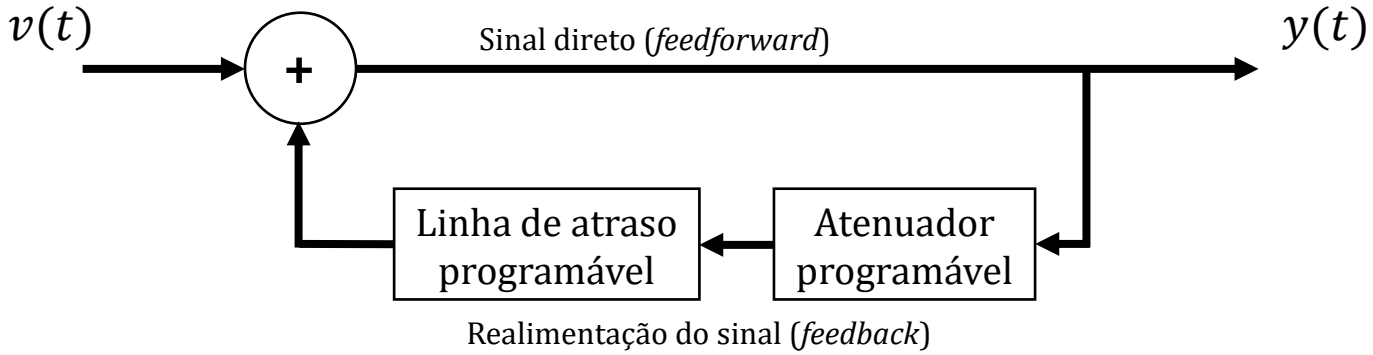


Conversor AD no STM32F407

Exemplo 2: Câmara de eco digital (algoritmos simplificados)



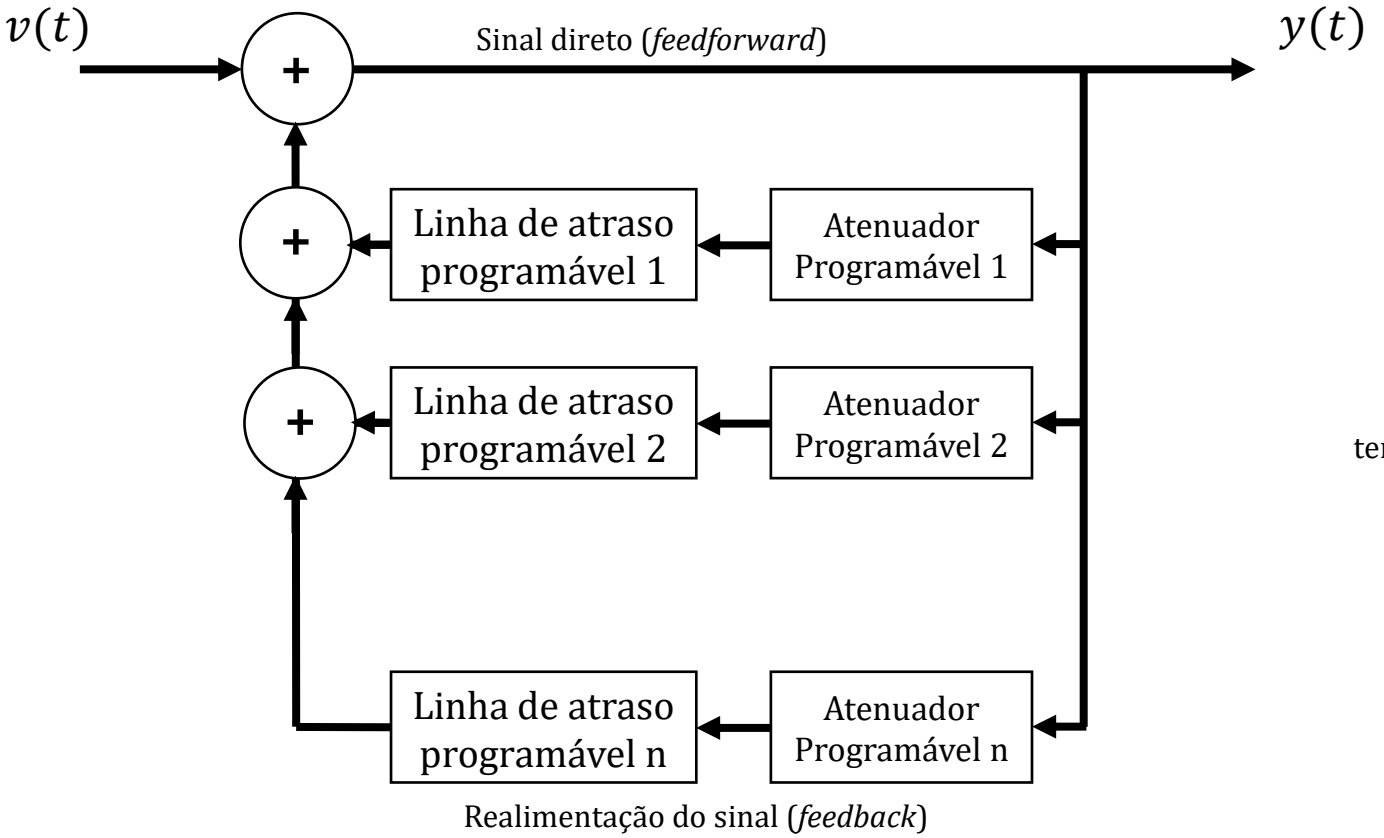
Reflexão única



Múltiplas reflexões

Conversor AD no STM32F407

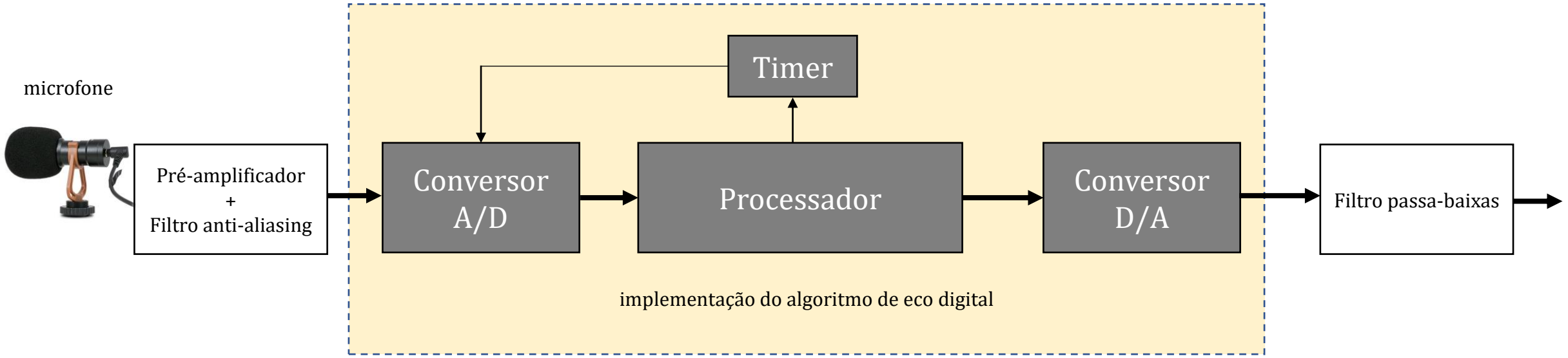
Exemplo 2: Câmara de eco digital (algoritmos simplificados)



Múltiplas reflexões com
tempos e taxas de atenuação distintas
(Reverb digital)

Conversor AD no STM32F407

Exemplo 2: Montagem



Conversor AD no STM32F407

Sensor de temperatura interno

* Faixa de operação de –40 a +105 °C com uma precisão de ± 1,5 °C.
* A tensão de saída do sensor varia linearmente com a temperatura.

Para usar o sensor de temperatura, é necessário:

1. Selecionar o canal de entrada ADC1_IN16;
2. Selecionar um tempo de amostragem maior que o tempo mínimo de amostragem especificado no datasheet do STM32F407 (10 μs);
3. Setar o bit TSVREFE no registrador CCR para ligar o sensor;
4. Iniciar a conversão do ADC setando o bit SWSTART (ou por gatilho externo);
5. Ler os dados resultantes no registrador de dados DR do módulo ADC;
6. Calcular a temperatura usando a seguinte fórmula:

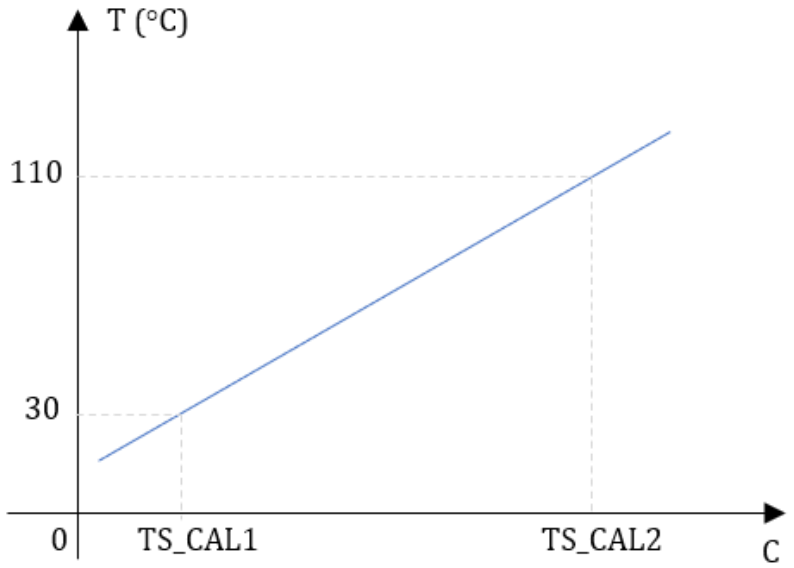
$$Temperatura\ (em\ ^\circ C) = \left\{\frac{(VSENSE - V_{25})}{Avg\_Slope}\right\} + 25$$

Conversor AD no STM32F407

Sensor de temperatura interno (calibração)

Tabela 2. Valores de calibração do sensor de temperatura

Símbolo	Parâmetro	Endereço de memória
TS_CAL1	Valor cru do ADC obtido na temperatura de 30 °C com V_{ref} a 3,3V	0x1FFF7A2C - 0x1FFF7A2D
TS_CAL2	Valor cru do ADC obtido na temperatura de 110 °C com V_{ref} a 3,3V	0x1FFF7A2E - 0x1FFF7A2F



$$T = \frac{80(C - TS\_CAL1)}{(TS\_CAL2 - TS\_CAL1)} + 30$$

Conversor AD no STM32F407

Sensor de temperatura interno

```
void main()
{
  Configure_Clock();                          //configura o sistema de clock
  USART1_Init();                              //inicializa a USART1
  Delay_Start();                              //inicializa funções de Delay

  RCC->APB2ENR |= 1 << 8;                     //liga o clock da interface digital do ADC1

  ADC->CCR |= 0b01 << 16;                     //prescaler /4
  ADC1->SQR1 &= ~(0xF << 20);                 //conversão de apenas um canal
  ADC1->SQR3 |= 16;                           //seleção do canal a ser convertido (IN_16)
  ADC1->SMPR1 |= (7 << 18);                   //tempo de amostragem igual a 480 ciclos de ADCCLK
  ADC->CCR |= (1 << 23);                      //liga o sensor de temperatura
  ADC1->CR2 |= 1;                             //liga o conversor AD

  uint32_t *p = (uint32_t *) 0x1FFF7A2C;      //cria ponteiro para uma posição de memória
  uint32_t Word = p[0];                       //lê o conteúdo da memória
  uint16_t TS_CAL1 = (Word & 0x0000FFFF);     //lê o valor de TS_CAL1
  uint16_t TS_CAL2 = (Word & 0xFFFF0000) >> 16; //lê o valor de TS_CAL2

  while(1)
  {
    ADC1->CR2 |= 1 << 30;                     //inicia a conversão
    while(!(ADC1->SR & 0x02));                //aguarda o fim da conversão

    //calcula a temperatura
    uint8_t temperatura = ((80*(int)(ADC1->DR - TS_CAL1))/(TS_CAL2-TS_CAL1))+30;
    printf("Temperatura = %d °C\n", temperatura);          //imprime a temperatura
    Delay_ms(500);                                         //aguarda 500ms para fazer a nova leitura
 }
}
```

Conversor analógico-digital (AD) no STM32F407

Prof. Fagner de Araujo Pereira

fagner.pereira@ifpb.edu.br

fagnereng@gmail.com