

## Capítulo

## 8

# Conversor Digital-Analógico no STM32F407

## 8.1. Introdução

Dizemos que os fenômenos do mundo real são analógicos. Em contradição, a tecnologia de processamento digital está cada vez mais presente no nosso dia a dia. Assim, são necessárias interfaces que convertem informações analógicas – do mundo real – para dados digitais que podem ser processados e, em seguida, reconvertidos para a forma analógica. Um dos exemplos mais comuns dessas interfaces são os aparelhos que reproduzem áudio, como players de música, convertendo informações digitais (a música codificada digitalmente) para a forma analógica (o som produzido pelos alto-falantes).

Muitos componentes e equipamentos eletrônicos não entendem sinais digitais, necessitando-os na forma analógica. Em geral, processos de conversão de sinais elétricos para grandezas físicas exigem saídas analógicas. Um bom exemplo é o próprio alto-falante: Para poder reproduzir um sinal de áudio, o alto-falante precisa ser excitado por meio de um sinal analógico. É necessário então a transformação das cadeias de bits gerada nas saídas dos sistemas digitais em sinais analógicos. Esta é a função do conversor digital-analógicos, ou conversor DA, ou ainda simplesmente DAC (*Digital to Analog Converter*).

Um conversor digital para analógico transforma um número digital de entrada, com precisão finita de  $n$  bits, em uma tensão analógica de saída, linearmente proporcional ao código digital de entrada. Na Figura 1 é apresentado um diagrama ilustrativo da operação de um conversor DA, onde cada entrada binária é convertida em um valor analógico de tensão de saída.

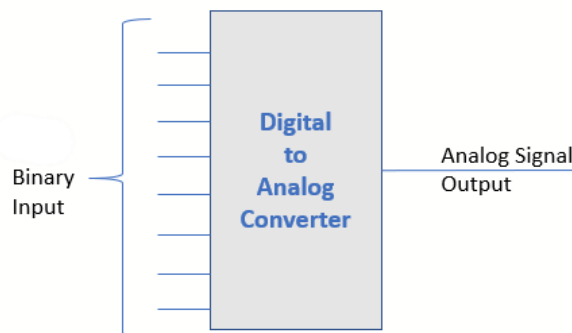


Figura 1 – Diagrama funcional de um conversor DA.

Existem diversas implementações práticas de circuitos para um conversor DA, sendo as redes resistivas ponderadas, redes resistivas R-2R, Sigma-Delta e técnicas de PWM as mais utilizadas. Na Figura 2 é mostrada uma implementação simples de um conversor DA de 4 bits de entrada construído a partir de um circuito somador inversor baseado em amplificador operacional, conhecida como rede resistiva ponderada.

Suponha que o valor digital a ser convertido em uma tensão analógica tenha 4 bits:  $D_3 D_2 D_1 D_0$ , com  $D_3$  sendo o bit mais significativo.  $D_i$  pode ser 0 ou 1, para  $i=0, 1, 2$  ou 3. Se  $D_i$  é 0, a chave correspondente no circuito da Figura 1 é aberta. No caso contrário, a chave correspondente é fechada. As chaves são dispositivos eletrônicos de chaveamento rápido, como transistores de efeito de campo (*FET – Field Effect Transistor*).

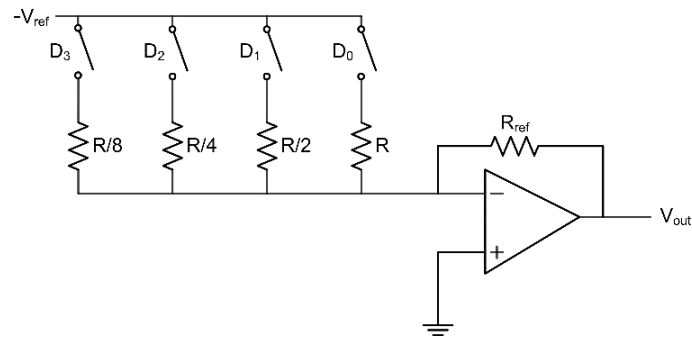


Figura 2 – Circuito básico de um DAC.

Para o circuito da Figura 2, a tensão analógica de saída  $V_{out}$  pode ser calculada como:

$$V_{out} = V_{ref} * \frac{R_{ref}}{R/8} D_3 + V_{ref} * \frac{R_{ref}}{R/4} D_2 + V_{ref} * \frac{R_{ref}}{R/2} D_1 + V_{ref} * \frac{R_{ref}}{R} D_0$$

$$V_{out} = V_{ref} * R_{ref} \left( \frac{D_3}{R/8} + \frac{D_2}{R/4} + \frac{D_1}{R/2} + \frac{D_0}{R} \right),$$

que pode ser reescrita da seguinte forma:

$$V_{out} = V_{ref} * \frac{R_{ref}}{R} (2^3 D_3 + 2^2 D_2 + 2 D_1 + 2^0 D_0)$$

Fazendo  $R_{ref} = R/(2^n - 1)$  na equação acima, com  $n$  igual à quantidade de bits, temos que:

$$V_{out} = \frac{V_{ref}}{15} * (8D_3 + 4D_2 + 2D_1 + 1D_0),$$

o que mostra que a tensão de saída é linearmente proporcional ao valor digital a ser convertido e pode variar de 0 a  $V_{ref}$ .

Um registrador  $D_{in}$  é usado para armazenar a entrada digital do DAC e assegurar que sua tensão de saída se mantenha fixa até que o conversor receba outra entrada digital. O registrador pode ser externo, mas normalmente faz parte do circuito do DAC. Para um conversor de 12 bits, por exemplo, a tensão de saída  $DAC_{output}$  pode ser calculada diretamente a partir do código fornecido na entrada  $D_{in}$  como:

$$DAC_{output} = V_{ref} * \frac{D_{in}}{4095}$$

Um conversor DA é sempre avaliado por três parâmetros: sua resolução, tempo de acomodação (*settling time*) e transitórios (*glitches*).

A resolução é a menor mudança que pode ocorrer na tensão analógica de saída quando a entrada digital varia apenas o seu bit menos significativo. Por simplicidade, também podemos usar o número de bits na entrada do DAC para representar sua resolução. Para um DAC de  $n$  bits, o número total de possíveis níveis de saída é  $2^n$  e a sua resolução é definida como:

$$Resolução = \frac{range\ de\ saída}{2^n - 1}$$

Por exemplo, se a tensão de saída varia de 0 a 3,3V (range de saída), então a resolução de um conversor de 12 bits é:

$$Resolução = \frac{range\ de\ saída}{2^n - 1} = \frac{3,3}{2^{12} - 1} \cong 805,86\ \mu V$$

O tempo de acomodação é o intervalo de tempo entre uma atualização nas entradas digitais do DAC até o instante em que a saída do conversor se torna estável.

Os transitórios (*glitches*) é o primeiro pico transitório que aparece na saída do conversor. Idealmente, quando a entrada muda, a saída do DAC deveria mudar monotonicamente para o novo valor. Na prática, a saída pode reduzir ou ultrapassar o novo valor devido às capacitâncias parasitas do circuito, ou devido ao tempo de fechamento das chaves da Figura 2. Por exemplo, algumas chaves podem operar mais rápidas do que outras, resultando em transitórios na tensão de saída.

Na figura 3 é mostrado como se comporta a saída de um conversor DA de 4 bits quando as entradas digitais são fornecidas periodicamente e seus valores (amostras) são obtidos a partir de uma função senoidal com valor médio igual a 7. Em vermelho, temos a função analógica senoidal ideal. A saída do conversor DA possui uma forma de onda em degraus na qual os valores são mantidos constantes durante o período de amostragem, e, por isso, podemos dizer que um conversor DA é um sistema de retenção de ordem zero. Para suavizar a forma de onda em degraus e aproximá-la à função original, podemos conectar na saída do conversor DA filtros analógicos do tipo passa-baixas.

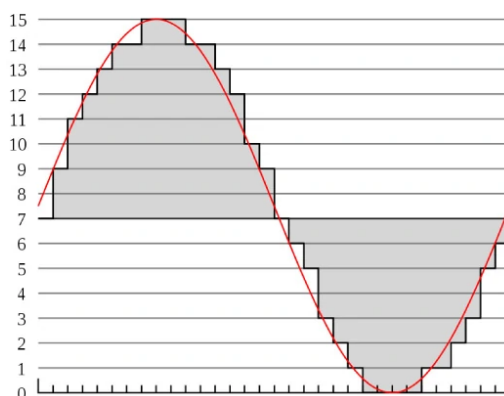


Figura 3 – Sinal de saída de um DAC quando as entradas são obtidas a partir de uma função senoidal de média 7.

## 8.2. Conversor DA no STM32F407

O módulo DAC no STM32F407 consiste em dois conversores digital-analógico independentes de 12 bits de resolução máxima com range de tensão de saída de 0 a 3,3V conectado ao barramento APB1. Os dois canais do DAC podem ser configurados no modo de resolução de 8 ou 12 bits.

No modo de operação duplo, as conversões podem ser feitas independentemente ou simultaneamente quando os dois canais são agrupados para operações de atualização síncrona.

O conversor também pode ser usado em conjunto com o periférico de acesso direto à memória DMA. Nesse modo, os dados a serem convertidos são enviados da memória ao conversor diretamente pelo DMA, sem o intermédio do processador, que pode ficar livre para execução de outras tarefas.

As principais características do módulo DAC do STM32F407 são:

- Alinhamento de dados, no registrador de saída de dados, à esquerda ou à direita no modo de 12 bits
- Geração automática de formas de ondas de ruído e triangulares
- Gatilhos internos e externos para conversão

É mostrado na Tabela 1 o roteamento entre as saídas do DAC e os pinos de I/O do chip.

Tabela 1. Definição dos pinos de saída do DAC

Canal de saída analógico	Pino
DAC1	PA4
DAC2	PA5

Para usar um canal do módulo DAC, o pino de saída correspondente, PA4 ou PA5, deve inicialmente ser configurado no modo analógico. Quando um canal DAC é ativado, o pino GPIO correspondente é diretamente conectado à saída do conversor.

Cada canal pode ser ligado setando o bit EN correspondente no registrador CR (*Control Register*) do módulo DAC. O canal DAC é ativado depois de um tempo de inicialização  $t_{WAKEUP}$ . O bit EN ativa apenas a macro célula analógica do DAC correspondente. A interface digital do DAC, onde estão os registradores de controle e de dados, é ativada ao ligar o clock do módulo DAC por meio do bit DACEN do registrador APB1ENR do módulo RCC, independentemente do estado do bit EN.

O módulo DAC possui dois *buffers* analógicos, um para cada saída, que podem ser usados para reduzir a impedância de saída e alimentar cargas diretamente sem a necessidade de adicionar externamente um *buffer* baseado em amplificador operacional. Cada *buffer* de saída pode ser ativado ou desativado usando o bit BOFF correspondente no registrador CR do módulo DAC.

Para converter um valor digital para analógico é necessário apenas escrever o dado em um registrador de saída. Cada canal tem três registradores de dados de saída (DHR - *Data Holding Register*). Dependendo do modo de operação selecionado, o software precisa carregar os dados no registrador específico, conforme descrito abaixo:

- Para utilização de apenas um único canal, existem três possibilidades, conforme mostrado na Figura 5:

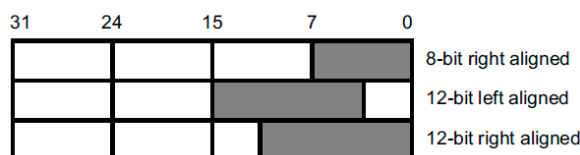


Figura 5 – Alinhamento de dados com um único canal do DAC.

- Alinhamento à direita de 8 bits: carregar dados no registrador DHR8Rx[7:0] do módulo DAC;
- Alinhamento à esquerda de 12 bits: carregar dados no registrador DHR12Lx[15:4] do módulo DAC;
- Alinhamento à direita de 12 bits: carregar dados no registrador DHR12Rx[11:0] do módulo DAC.

Dependendo do registrador DHRyyyx carregado, os dados gravados pelo usuário são deslocados e armazenados no registrador DHR correspondente, que são registradores internos não mapeados na memória. O registrador DHR é então carregado no registrador DOR (*Data Output Register*) automaticamente, por gatilho de software ou por um gatilho externo.

- Para utilização dos dois canais DAC, existem três possibilidades, conforme mostrado na Figura 6:

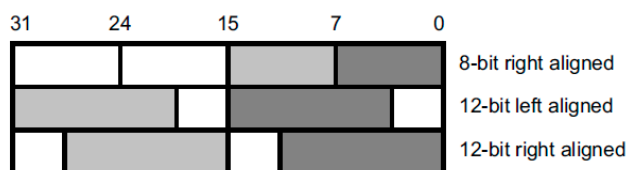


Figura 6 – Alinhamento de dados com canais duplos do DAC.

- Alinhamento à direita de 8 bits: carregar dados para o canal DAC1 no registrador DHR8RD[7:0] e dados para o canal DAC2 no registrador DHR8RD[15:8] do módulo DAC;
- Alinhamento à esquerda de 12 bits: carregar dados do canal DAC1 no registrador DAC\_DHR12LD[15:4] e dados para o canal DAC2 no registrador DHR12LD[31:20] do módulo DAC;
- Alinhamento à direita de 12 bits: carregar dados para o canal DAC1 no registrador DHR12RD[11:0] e dados para o canal DAC2 no registrador DHR12LD[27:16] do módulo DAC.

Dependendo do registrador DHRyyyD carregado, os dados gravados pelo usuário são deslocados e armazenados em DHR1 e DHR2, que são registradores internos não mapeados na memória. Os registradores DHR1 e DHR2 são então carregados nos registradores DOR1 e DOR2, respectivamente, automaticamente, por gatilho de software ou por um gatilho externo.

Os registradores DOR não podem ser gravados diretamente e qualquer transferência de dados para o canal DAC deve ser realizada carregando um dos registradores DHR. Os dados armazenados no registrador DHR são transferidos automaticamente para o registrador DOR após um ciclo de clock do barramento APB1, se nenhum gatilho de hardware for selecionado. No entanto, quando um gatilho de hardware é selecionado e um gatilho ocorre, a transferência é realizada após três ciclos de clock APB1. O controle de habilitação de gatilhos de hardware é feito pelo bit TEN e a fonte de gatilho é selecionada pelos bits TSEL[2:0] do registrador CR.

Quando DOR é carregado com o conteúdo DHR, a tensão de saída analógica fica disponível após o tempo de acomodação que depende da tensão da fonte de alimentação e da carga de saída. As entradas digitais são convertidas em tensão de saída em uma escala linear entre 0 e VREF+ (3,3V). Considerando uma resolução de 12 bits, as tensões de saída analógica em cada canal DAC são determinadas pela seguinte equação:

$$DAC_{output} = 3,3 * \frac{DOR}{4095}$$

## 8.4 Exemplo de configuração do conversor DA

Abaixo é mostrada a configuração de software para usar o pino PA4 como pino de saída do canal 1 do DAC no modo de conversão simples de um canal sem gatilhamento de hardware.

```
void Configure_DAC1()
{
    RCC->AHB1ENR |= 1;           //habilita o clock do GPIOA
    GPIOA->MODER |= 0b11 << 8;   //inicialização do pino PA4 no modo analógico
    RCC->APB1ENR |= 1 << 29;     //habilita o clock da interface digital do DAC
    DAC->CR |= 1;                //habilita o canal 1 do DAC
}
```

As linhas de código abaixo escrevem um valor digital de 12 bits alinhado à direita, armazenado na variável *valor*, para ser convertido pelo canal 1 do DAC.

```
uint16_t valor;
DAC->DHR12R1 = valor;
```

O código a seguir é um programa completo que gera uma forma de onda senoidal com uma frequência de 50 Hz no pino PA4 por meio de uma *look up table*. Para isso, o array `samples[100]` contém 100 amostras de um sinal senoidal que varia de 0 a 4095 que compreende o intervalo de um ciclo do sinal. No loop principal, cada amostra é fornecida ao conversor DAC em intervalos de 200  $\mu$ s.

```
//array com amostras do sinal
const uint16_t samples[100]={
    2048, 2176, 2304, 2431, 2557, 2680, 2801, 2919, 3034, 3145,
    3251, 3353, 3449, 3540, 3625, 3704, 3776, 3842, 3900, 3951,
    3995, 4031, 4059, 4079, 4091, 4095, 4091, 4079, 4059, 4031,
    3995, 3951, 3900, 3842, 3776, 3704, 3625, 3540, 3449, 3353,
    3251, 3145, 3034, 2919, 2801, 2680, 2557, 2431, 2304, 2176,
    2048, 1919, 1791, 1664, 1538, 1415, 1294, 1176, 1061, 950,
    844, 742, 646, 555, 470, 391, 319, 253, 195, 144,
    100, 64, 36, 16, 4, 0, 4, 16, 36, 64,
    100, 144, 195, 253, 319, 391, 470, 555, 646, 742,
    844, 950, 1061, 1176, 1294, 1415, 1538, 1664, 1791, 1919};

//Programa principal
void main()
{
    Configure_Clock();           //configura o sistema de clock
    Delay_Start();               //inicializa funções de Delay

    RCC->AHB1ENR |= 1;           //habilita o clock do GPIOA
    GPIOA->MODER |= 0b11 << 8;   //inicialização do pino PA4 no modo analógico

    RCC->APB1ENR |= 1 << 29;     //habilita o clock da interface digital do DAC
    DAC->CR |= 1;                //habilita o canal 1 do DAC

    uint8_t contador = 0;        //indexador do array de amostras do sinal
    while(1)
    {
        DAC->DHR12R1 = samples[contador]; //escreve no DAC1
        ++contador;                 //atualiza o indexador
        if(contador == 100) contador = 0; //verifica se chegou ao final do array
        Delay_us(200);              //aguarda para a próxima amostra
    }
}
```

Na Figura 7 é mostrada o sinal gerado no pino PA4 a partir da captura da tela de um osciloscópio e na Figura 8 é exibido em detalhe a forma de onda em degraus do sinal de saída (a) e após a suavização desse sinal por um filtro analógico do tipo passa-baixas (b). ■

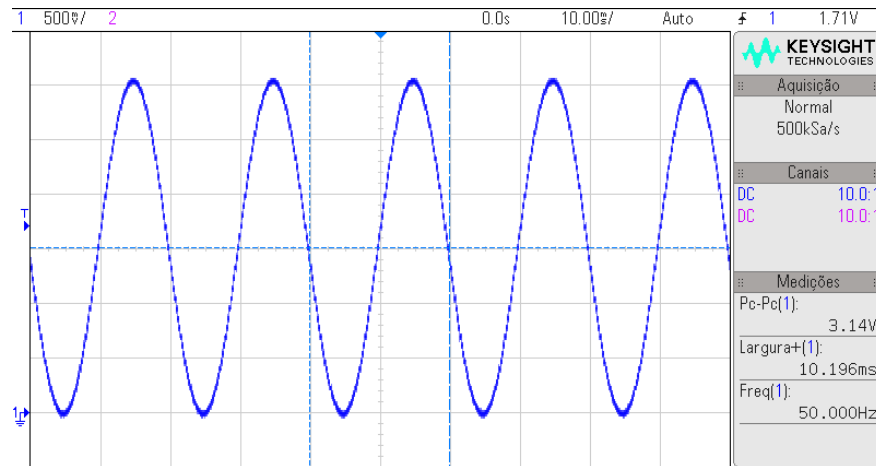


Figura 7 – Forma de onda capturada na saída do canal 1 do DAC.

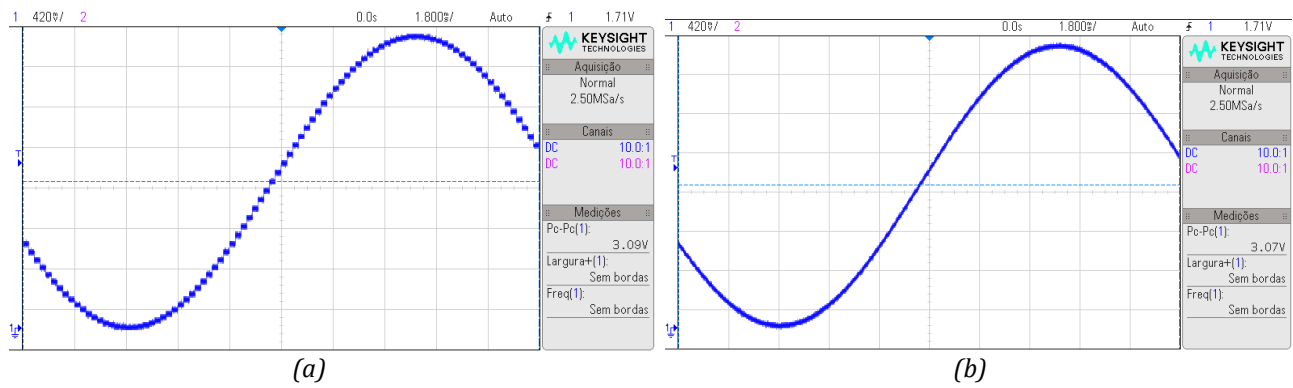


Figura 8 – Detalhe da forma de onda em degraus na saída do canal 1 do DAC (a) e após a suavização com filtro passa baixas (b).