

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
DEPARTAMENTO DE ENGENHARIA ELETRÔNICA

Programação de Sistemas Embarcados

Professor: Ricardo de Oliveira Duarte

Autor: Victor Freitas Borges - [victorfborges75@gmail.com](mailto:victorfborges75@gmail.com)

Versão: 1.0

Licença: GNU 3

Data: 08/2021

Application Note: AN01-1.0 - DAC (Digital Analog Converter)

## DAC (Digital to Analog Converter)

Esta Application Note foi desenvolvida como trabalho da disciplina de Programação de Sistemas Embarcados da UFMG - Prof. Ricardo de Oliveira Duarte - Departamento de Engenharia Eletrônica.

## **Introdução:**

Essa application note aborda o funcionamento do periférico DAC - Digital to Analog Converter, presente nos microcontroladores STM32. O objetivo desse documento é explicitar o princípio de funcionamento do periférico e como configurá-lo. Para isso usaremos o microcontrolador presente no kit NUCLEO-H743.

**Sumário:**

Introdução.....	1
Figuras.....	3
Funcionamento	
Princípio de funcionamento e aplicação.....	4
Modos de operação.....	8
Parâmetros e configuração.....	10
Referências.....	12

## **Figuras:**

*Figura 1 - Diagrama esquemático que representa as ligações Retirado do Reference Manual[1] da família STM32H7.*

*Figura 2 - Tabela que mostra a descrição dos sinais presentes no diagrama da Figura 1. Retirado do Reference Manual[1] da família STM32H7.*

*Figura 3 - Tabela que mostra a origem das interrupções associadas ao DAC. Retirado do Reference Manual[1] da família STM32H7.*

*Figura 4 - Registrador de entrada do canal do DAC configurado como canal único, é possível ver que há 3 possibilidades, sendo a primeira aplicável quando o modo de operação é configurado para 8 bits e as duas em sequência para modos de 12 bits, com diferença para a forma de alinhamento. Observamos que o campo que armazena os dados do segundo conversor (bit 31 ao bit 15 estão vazios).*

*Figura 5 - Registrador de entrada do canal do DAC configurado como canal duplo, é possível ver que há 3 possibilidades, sendo a primeira aplicável quando o modo de operação é configurado para 8 bits e as duas em sequência para modos de 12 bits, com diferença para a forma de alinhamento. Observamos que o campo que armazena os dados para o primeiro modo usa somente metade do espaço total disponível, mantendo vazio os bits 31 ao 15. Já os dois modos em sequência utilizam a totalidade do registrado e se diferenciam somente por alinhamento à direita ou à esquerda.*

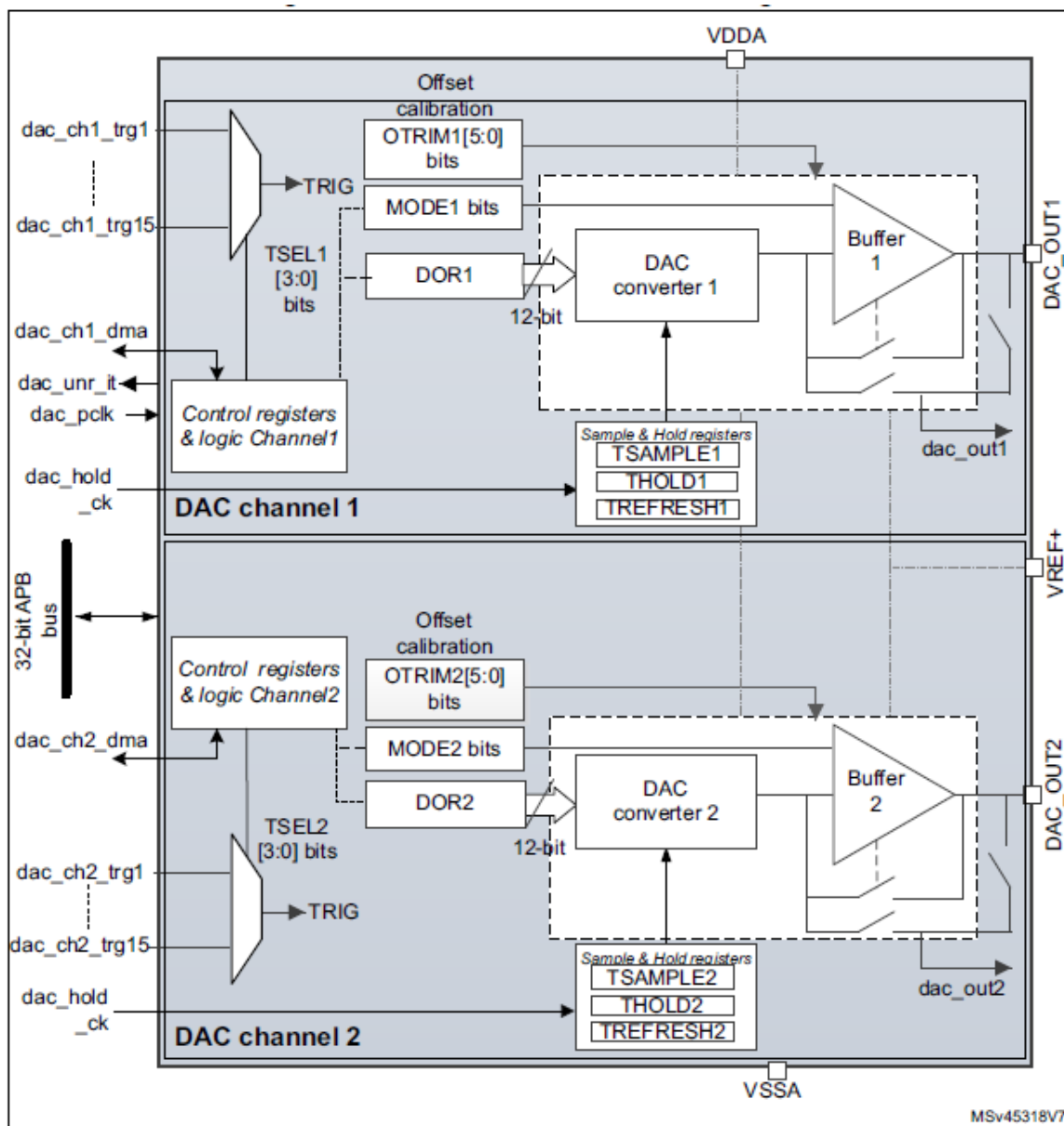
*Figura 6: Possibilidades de configuração dos canais do DAC.*

*Figura 7: Parâmetros de configuração do DAC para o modo de saída associado a um pino externo e interno.*

### Princípio de funcionamento e aplicação:

Um DAC (*Digital Analog Converter*) ou em português, conversor analógico digital é um periférico do microcontrolador que realiza interface entre um sinal digital e um sinal analógico. Essa ponte é feita por meio de um circuito que recebe um conjunto de bits e usa essa informação como referência para controlar a saída de um amplificador.

Os microcontroladores da família STM32H7 possuem dois canais do periférico DAC, que devem funcionar em conjunto com o periférico de DMA (*Direct Memory Access*), segundo o *reference manual* [1]. Esses canais podem ser representados na seguinte imagem:



## **AN01 - DAC (Digital Analog Converter)**

*Figura 1 - Diagrama esquemático que representa as ligações Retirado do Reference Manual[1] da família STM32H7.*

Cada DAC possui os próprios conjuntos de sinais de entrada e porta de saída, mas compartilham alimentação e referência. Podemos observar que essa instância também recebe a alimentação compartilhada com os demais periféricos que executam função analógica, com o objetivo de melhorar a resolução. Ademais, há a possibilidade de conectar internamente a saída analógica a outro periférico do microcontrolador.

O sinais do periférico podem ser visualizados na Figura 2 e Figura 3:

<b>Internal signal name</b>	<b>Signal type</b>	<b>Description</b>
dac_ch1_dma	Bidirectional	DAC channel1 DMA request/acknowledge
dac_ch2_dma	Bidirectional	DAC channel2 DMA request/acknowledge
dac_ch1_trgx (x = 1 to 15)	Inputs	DAC channel1 trigger inputs
dac_ch2_trgx (x = 1 to 15)	Inputs	DAC channel2 trigger inputs
dac_unr_it	Output	DAC underrun interrupt
dac_pclk	Input	DAC peripheral clock
dac_hold_ck	Input	DAC low-power clock used in Sample and hold mode
dac_out1	Analog output	DAC channel1 output for on-chip peripherals
dac_out2	Analog output	DAC channel2 output for on-chip peripherals

*Figura 2 - Tabela que mostra a descrição dos sinais presentes no diagrama da Figura 1. Retirado do Reference Manual[1] da família STM32H7.*

## **AN01 - DAC (Digital Analog Converter)**

Signal name	Source	Source type
dac_hold_ck	ck_lsi (selected in the RCC)	LSI clock selected in the RCC
dac_chx_trg1 (x = 1, 2)	tim1_trgo	Internal signal from on-chip timers
dac_chx_trg2 (x = 1, 2)	tim2_trgo	Internal signal from on-chip timers
dac_chx_trg3 (x = 1, 2)	tim4_trgo	Internal signal from on-chip timers
dac_chx_trg4 (x = 1, 2)	tim5_trgo	Internal signal from on-chip timers
dac_chx_trg5 (x = 1, 2)	tim6_trgo	Internal signal from on-chip timers
dac_chx_trg6 (x = 1, 2)	tim7_trgo	Internal signal from on-chip timers
dac_chx_trg7 (x = 1, 2)	tim8_trgo	Internal signal from on-chip timers
dac_chx_trg8 (x = 1, 2)	tim15_trgo	Internal signal from on-chip timers
dac_chx_trg9 (x = 1, 2)	hrtim1_dactrg1	Internal signal from on-chip timers
dac_chx_trg10 (x = 1, 2)	hrtim1_dactrg2	Internal signal from on-chip timers
dac_chx_trg11 (x = 1, 2)	lptim1_out	Internal signal from on-chip timers
dac_chx_trg12 (x = 1, 2)	lptim2_out	Internal signal from on-chip timers
dac_chx_trg13 (x = 1, 2)	exti9	External pin

*Figura 3 - Tabela que mostra a origem das interrupções associadas ao DAC. Retirado do Reference Manual[1] da família STM32H7.*

Nesse contexto podemos observar que esse periférico possui uma conexão bidirecional associada a cada um dos conversores, uma saída analógica para cada conversor, um conjunto de 15 entradas de gatilho para cada conversor, uma saída de feedback para cada conversor, um barramento de clock para cada conversor e uma entrada dedicada para o clock quando operado no modo de baixo consumo. Para compreender o funcionamento desse tipo de conversor da família do STM32 é necessário compreender quais os significados de cada um desses sinais.

### **Conexão Bidirecional:**

Conexão utilizada para fazer a requisição e o reconhecimento da leitura. Essa troca de informações é feita entre o DMA e o controlador lógico de cada um dos conversores de forma independente. É por esse caminho que vem a informação de qual valor deve ser colocado na saída. O formato dessa informação tem variações de acordo com o modo em que o DAC é configurado. É importante mencionar que os dados vindos da memória são escritos nos registradores do DAC antes de serem usados para a operação.

### **Entradas de gatilho:**

## **AN01 - DAC (Digital Analog Converter)**

As conversões são realizadas condicionadas a um evento de gatilho. Ou seja, o valor da saída só é atualizado quando se recebe algum sinal de gatilho indicando para que esse processo seja executado. Nesse caso, podemos ver na Figura 3 que os gatilhos podem vir tanto de instâncias internas, quanto externas do microcontrolador.

### **Saída analógica:**

A saída analógica é o sinal de interesse, o qual possui um range que varia de 0 a  $V_{ref+}$ . E o valor de saída é dado pela equação  $DAC_{output} = V_{ref} * DOR / 4096$ . Em que DOR é a informação digital que corresponde a parte dos dados trocados entre o DMA e o DAC. É possível configurar a saída analógica para ser associada, separadamente, a um pino externo do microcontrolador, a um periférico interno ou aos dois simultaneamente.

### **Saída de interna:**

Essa conexão permite ligar o valor analógico da saída internamente a outros periféricos do microcontrolador. Essa saída pode ser conectada a uma entrada ADC ou bloco comparador caso estejam disponíveis.

### **Saída de status:**

Esse periférico possui uma saída chamada de underrun, a qual tem como função informar que o DAC está operando na conversão de um dado. Essa saída é importante, pois caso um novo dado seja enviado para o registrador do DAC e gatilho seja acionado durante o processo de conversão a tensão que vai aparecer na saída é referente ao dado que está sendo processado no momento que o gatilho foi acionado e não o dado que acabou de entrar no registrador, o que gera uma saída errada.

### **Entrada de clock:**

As operações realizadas no âmbito digital são dependentes de um sinal de relógio. Essa lógica continua sendo válida para esse periférico, de modo que o sinal dessa entrada é a referência para que para o bloco de controle dos registradores e controle da lógica funcione.



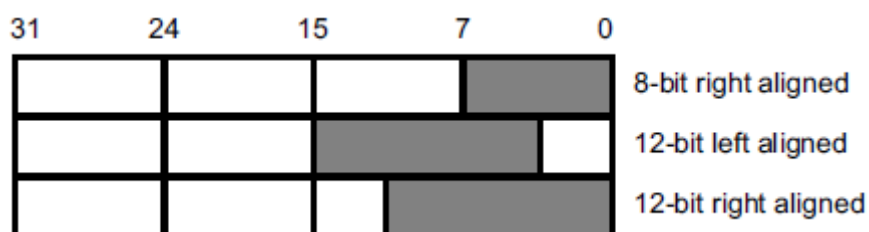
## **AN01 - DAC (Digital Analog Converter)**

### Entrada de clock de baixo consumo:

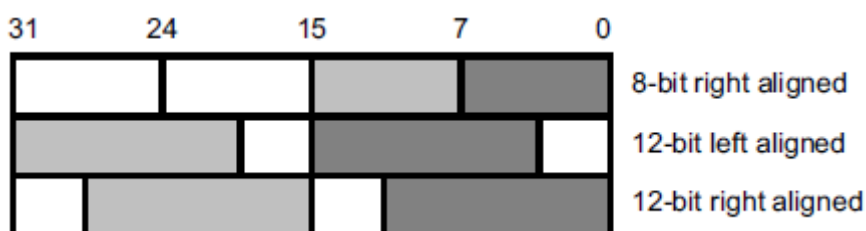
Esse sinal de relógio é dedicado ao controle no modo de operação Sample and Hold (o qual é explicado no tópico em sequência), sendo que a fonte para esse sinal é um sinal de clock de baixa frequência, logo menor consumo comparado com a entrada mencionada anteriormente. Esse recurso tem aplicação para o modo de operação de baixo consumo.

### **Modos de operação:**

Para o caso da família STM32H7 temos a possibilidade de operar o DAC de diferentes formas, sendo o modo de canal único, em que só um dos canais está ativo e o modo de canal duplo, em que os dois canais estão ativos. Além disso, é possível configurar para que o dado de entrada seja de 8 ou 12 bits. Essas possibilidades de configuração fazem com que os registradores que recebem os dados via DMA possam assumir as configurações vistas na Figura 4 (modo de operação canal único) e na Figura 5 (modo de operação de canal duplo).



*Figura 4 - Registrador de entrada do canal do DAC configurado como canal único, é possível ver que há 3 possibilidades, sendo a primeira aplicável quando o modo de operação é configurado para 8 bits e as duas em sequência para modos de 12 bits, com diferença para a forma de alinhamento. Observamos que o campo que armazena os dados do segundo conversor (bit 31 ao bit 15 estão vazios).*



*Figura 5 - Registrador de entrada do canal do DAC configurado como canal duplo, é possível ver que há 3 possibilidades, sendo a primeira aplicável quando o modo de*

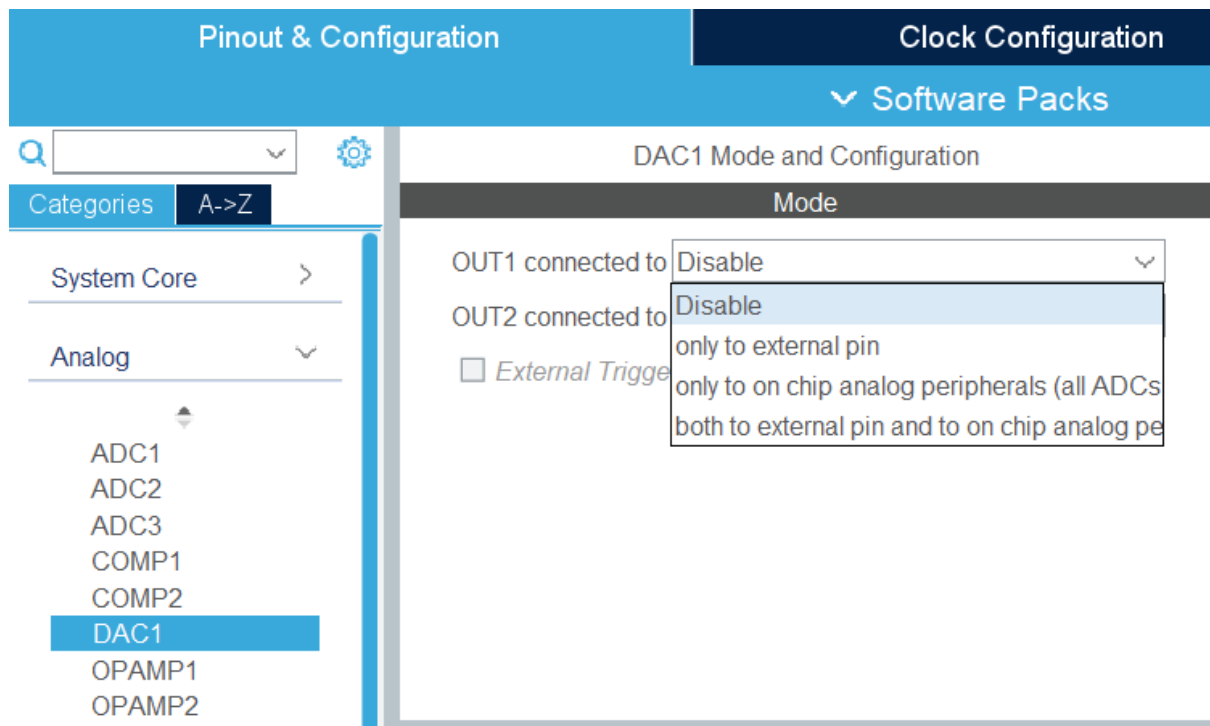
## **AN01 - DAC (Digital Analog Converter)**

*operação é configurado para 8 bits e as duas em sequência para modos de 12 bits, com diferença para a forma de alinhamento. Observamos que o campo que armazena os dados para o primeiro modo usa somente metade do espaço total disponível, mantendo vazio os bits 31 ao 15. Já os dois modos em sequência utilizam a totalidade do registrado e se diferenciam somente por alinhamento à direita ou à esquerda.*

Por fim, é possível configurar ainda o periférico para que ele atue como *Normal Mode* ou *Sample and Hold Mode*. O funcionamento do primeiro modo se diferencia do segundo pelo fato do núcleo de controle de conversão estar sempre ativo e controlando a tensão que é gerada na saída do conversor, ao passo que no segundo a tensão de saída é carregada em um capacitor, de forma a possibilitar que o núcleo de controle seja desativado por um determinado período de tempo e que a tensão de saída seja mantida. Esse último modo é utilizado em aplicações que exigem baixo consumo, já que durante determinado período de tempo parte do circuito fica desativado. Em ambos os modos é possível configurar se a saída analógica vai ser associada ou não a um circuito de *buffer* interno, o que serve para aumentar a capacidade de corrente da saída.

## Parâmetros e configuração:

Para ilustrar as possibilidades de configuração do DAC temos a seguinte sequência de figuras que foram retiradas do ambiente de configuração de pinagem do STM32H743ZI:



*Figura 6: Possibilidades de configuração dos canais do DAC.*

## AN01 - DAC (Digital Analog Converter)

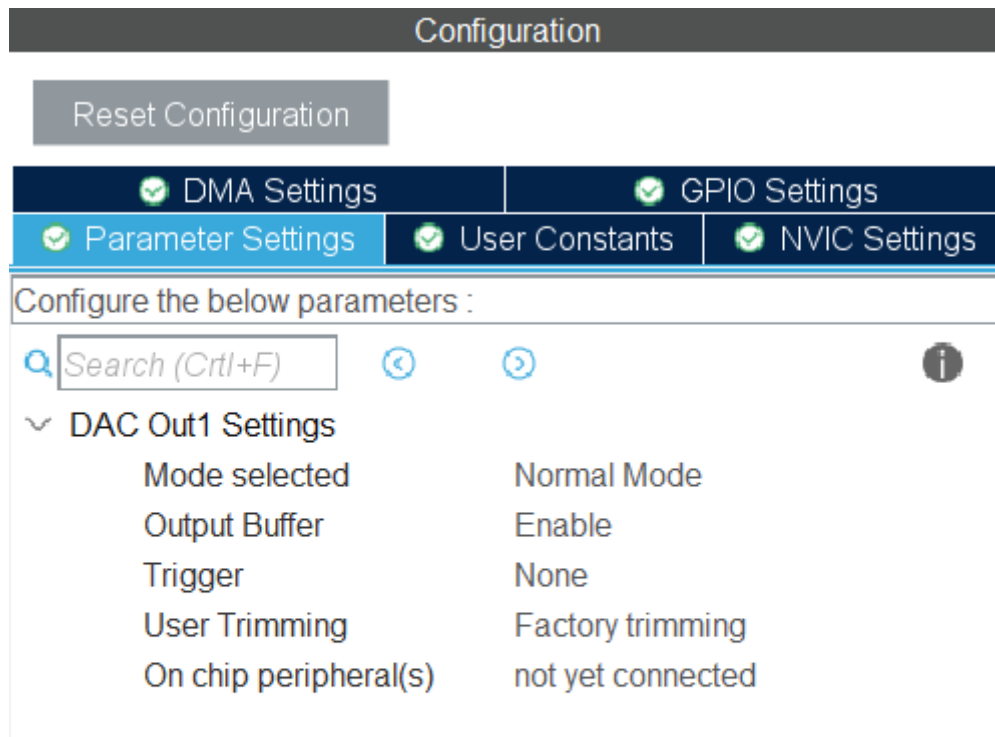


Figura 7: Parâmetros de configuração do DAC para o modo de saída associado a um pino externo e interno.

Na Figura 6 podemos observar os 3 possíveis modos de configuração da saída analógica, sendo possível associá-la a um pino externo, a um periférico interno ou aos dois simultaneamente. Para fins de exemplificação optamos pela configuração de saída compartilhada para periférico interno e pino externo. Com isso podemos configurar os parâmetros ilustrados na Figura 7. Sendo eles o modo de operação que varia de *Normal Mode* para *Sample and Hold Mode*. Em seguida podemos configurar a presença ou não de um *buffer* para a saída, a fonte do gatilho externo, que pode ser escolhida dentre as opções mostradas na Figura 3, o *user trimming* e por fim o periférico interno que essa saída vai ser associada.

Após realizar as configurações dos parâmetros da forma desejada, é possível utilizar as funções do HAL para implementar uma API que utilize esse periférico. O documento utilizado para visualizar as funções da HAL foi o *Description of STM32H7 HAL*[2].

## **AN01 - DAC (Digital Analog Converter)**

### **Referências:**

*[\[1\] Reference Manual STM32H7](#)*

*[\[2\] Description of STM32H7 HAL](#)*