

TEMA 6: CONVERSIÓN DIGITAL / ANALÓGICA

Sistemas Digitales basados en Microprocesador
Grado en Ingeniería Telemática

© Raúl Sánchez Reillo

1





ÍNDICE

- Conceptos Previos
- Conversor D/A y Funcionamiento
- DAC: Registros de Control
- DAC: Registros de Datos
- DAC: Registros de Estado
- Ejemplo de Conversión
- Ejercicios



CONCEPTOS PREVIOS

- El conversor DAC realiza la función opuesta al ADC, por lo que pondrá una tensión analógica en la salida, equivalente al valor digital procesado en el programa.
- Se puede utilizar para crear cualquier tipo de señal analógica, siempre que su evolución sea mucho más lenta que el tiempo de conversión del DAC

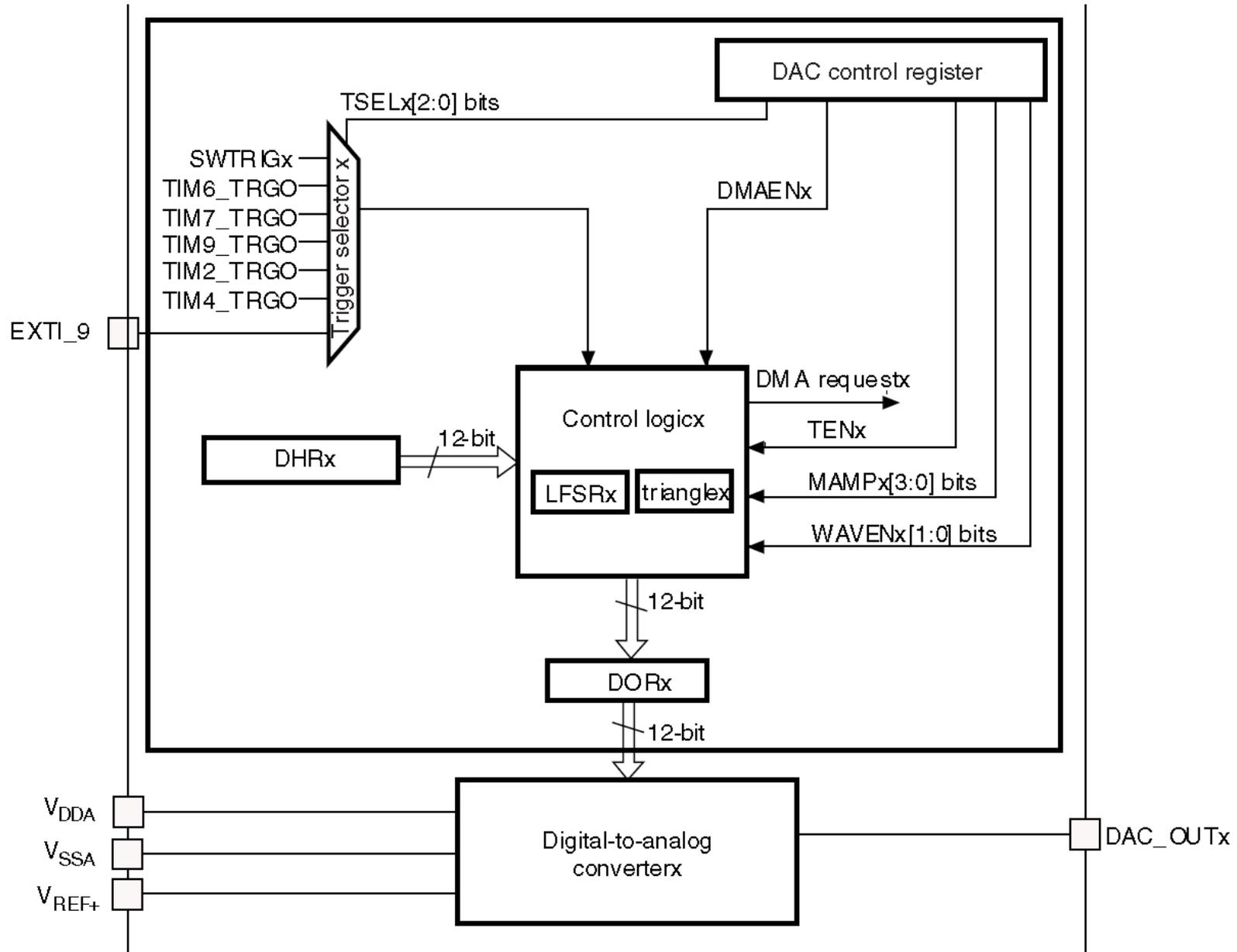


CONVERSOR DAC Y FUNCIONAMIENTO

- El STM32L152RB tiene dos DACs de 12 bits, cada uno con su propia salida
- Sus características principales son:
 - Resolución configurable a 12 u 8 bits
 - Posibilidad de gestionar datos digitales alineados a la derecha o a la izquierda
 - Las tensiones a obtener dependen de los valores de dos pines de entrada (V_{ref+} y masa)
 - En nuestro caso será entre 3,3V y 0V
- El DAC está conectado al APB1
- Para convertir, lo único que hay que hacer es activar el conversor, y poner un valor digital en el registro DHR escogido
 - El de 12-bits alineado a la derecha
 - El de 12-bits alineado a la izquierda
 - El de 8-bits alineado a la derecha



DAC Y FUNCIONAMIENTO





DAC: REGISTROS DE CONTROL

DAC → CR – Control Register:

- Registro de 32 bits con los siguientes bits de configuración:
 - DMAUDRIE2, DMAEN2, MAMP2[3:0], WAVE2[1:0], TSEL2[2:0], TEN2 – todos a '0'
 - BOFF2** – Deshabilitación del buffer de salida de canal 2 del DAC
 - Con un '0' está habilitado, con un '1' se deshabilita
 - EN2** – Habilitación del canal 2
 - Con un '0' se deshabilita, con un '1' se habilita
 - DMAUDRIE1, DMAEN1, MAMP1[3:0], WAVE1[1:0], TSEL1[2:0], TEN1 – todos a '0'
 - BOFF1** – Deshabilitación del buffer de salida de canal 1 del DAC
 - Con un '0' está habilitado, con un '1' se deshabilita
 - EN1** – Habilitación del canal 1
 - Con un '0' se deshabilita, con un '1' se habilita

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DMAUDRIE2	DMAEN2	MAMP2[3:0]			WAVE2[1:0]		TSEL2[2:0]			TEN2	BOFF2	EN2	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DMAUDRIE1	DMAEN1	MAMP1[3:0]			WAVE1[1:0]		TSEL1[2:0]			TEN1	BOFF1	EN1	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



DAC: REGISTROS DE DATOS

- **DAC→DHR12R1** – DAC Data Register con 12 bits y alineado a la derecha (canal 1):
 - Se escribe en los bits [11:0] el dato a convertir
- **DAC→DHR12L1** – DAC Data Register con 12 bits alineado a la izquierda (canal 1):
 - Se escribe en los bits [15:4] el dato a convertir
- **DAC→DHR8R1** – DAC Data Register con 8 bits alineado a la derecha (canal 1):
 - Se escribe en los bits [7:0] el dato a convertir
- Y de la misma forma existen, para el canal 2:
 - **DAC→DHR12R2**
 - **DAC→DHR12L2**
 - **DAC→DHR8R2**



DAC: REGISTROS DE ESTADO

○ DAC→SR – Status Register

- Registro de 32 bits con sólo 2 bits disponibles, pero ninguno útil para el curso.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DMAUDR2	Reserved												
		rc_w1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DMAUDR1	Reserved												
		rc_w1													



EJEMPLO DE USO DE GENERACIÓN DE ONDA

- El siguiente ejemplo genera una onda de variación no sinusoidal, usando valores de 8 bits, por el canal 2 (conectado al PA5)

```
#include "stm3211xx.h"
#include "Biblioteca_SDM.h"

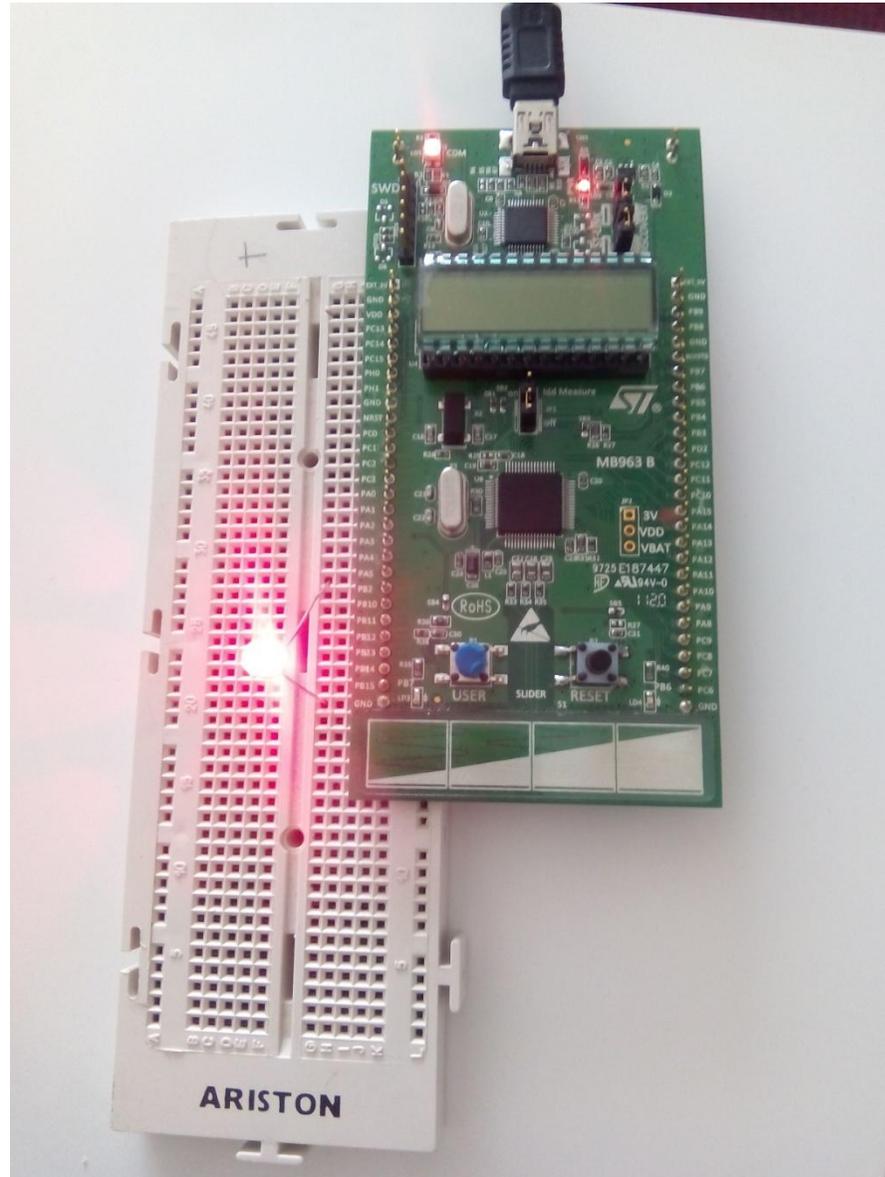
int main(void) {

    unsigned short i = 0;
    unsigned short onda[16] = {0, 2, 4, 8, 16, 32, 64, 128, 255, 128, 64, 32, 16, 8, 4, 2};
    Init_SDM();

    // Configuración del DAC
    GPIOA->MODER |= 0x00000C00;           // PA5 como señal analógica
    DAC->CR = 0x00010000;                 // Configuración y habilitación del canal 2

    while (1) {
        for (i=0; i<16; i++) {           // Saco los 16 valores uno a uno esperando un segundo
            DAC->DHR8R2 = onda[i];
            espera(5000000);
        }
    }
}
```

PRUEBA DEL EJEMPLO EXPLICADO





EJERCICIOS

- 1) Análisis del ejemplo: Realice el diagrama de flujo del ejemplo, cree un proyecto y escriba el código fuente comentando cada línea y/o grupo funcional. Finalmente ejecútelo y pruébelo con el depurador.
 - ¿Por qué es necesaria la función espera(x)?
 - ¿En qué influye el valor de x de la función espera(x)?
- 2) Cambie la resolución del conversor a 12 bits.
 - ¿Qué diferencias aprecia al ejecutarlo?
- 3) Con la resolución que desee, intente modificar el programa para generar una señal sinusoidal de 1KHz de frecuencia.