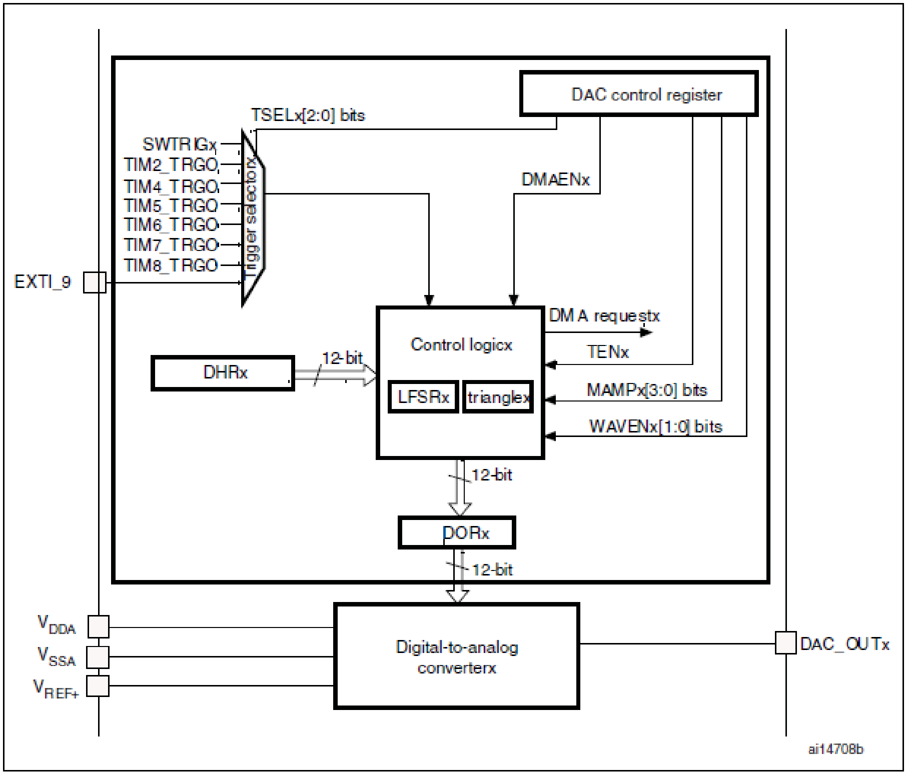
The main goal here is to explore noise/wave hardware generation capabilities of stm32f4 DAC.

DAC introduction

The DAC module is a 12-bit, voltage output digital-to-analog converter. The DAC can be configured in 8- or 12-bit mode and may be used in conjunction with the DMA controller. In 12-bit mode, the data could be left- or right-aligned. The DAC has two output channels, each with its own converter.

DAC main features:

- Left or right data alignment in 12-bit mode
- Synchronized update capability
- Noise-wave generation
- Triangular-wave generation
- Dual DAC channel for independent or simultaneous conversions
- DMA capability for each channel
- DMA underrun error detection
- External triggers for conversion
- Input voltage reference, VREF+



DAC channel block diagram

Noise generation

In order to generate a variable-amplitude pseudonoise, an LFSR (linear feedback shift register) is available. DAC noise generation is selected by setting **WAVEx[1:0]** to “**01**”. The preloaded value in LFSR is 0xAAA. This register is updated three APB1 clock cycles after each trigger event, following a specific calculation algorithm.

Triangle-wave generation

The STM32 DAC provides the user with a triangular waveform generator with a flexible offset, amplitude and frequency. It is possible to add a small-amplitude triangular waveform on a DC or slowly varying signal.

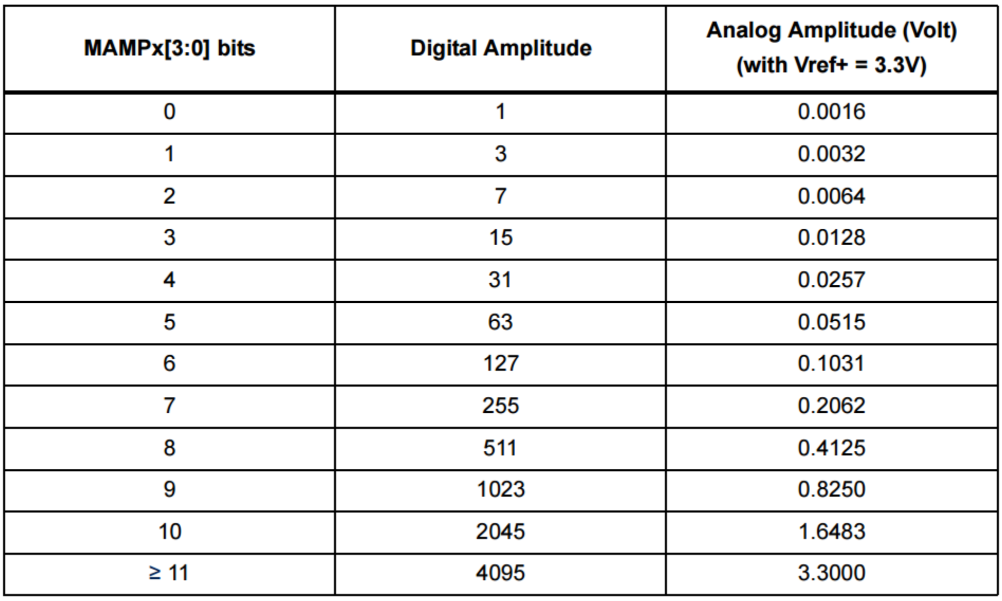
DAC triangle-wave generation is selected by setting **WAVEx[1:0]** to “**10**”. The amplitude is configured through the **MAMPx[3:0]** bits in the DAC_CR register. An internal triangle counter is incremented three APB1 clock cycles after each trigger event. The value of this counter is then added to the DAC_DHRx register without overflow and the sum is stored into the DAC_DORx register.

Application1 : Triangle wave generation

The amplitude of the triangular waveform can be fixed using the MAMPx bits in the

DAC_CR register.

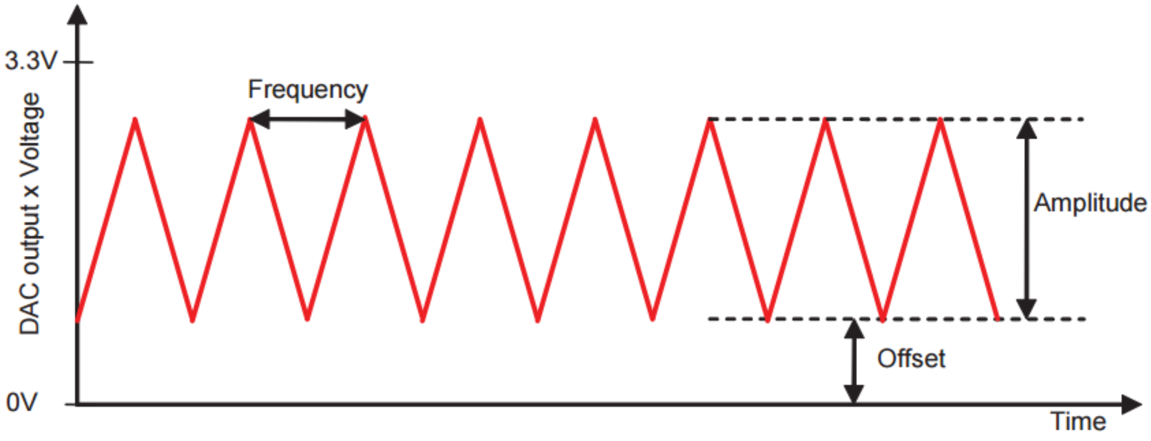
Preprogrammable triangular waveform amplitude values



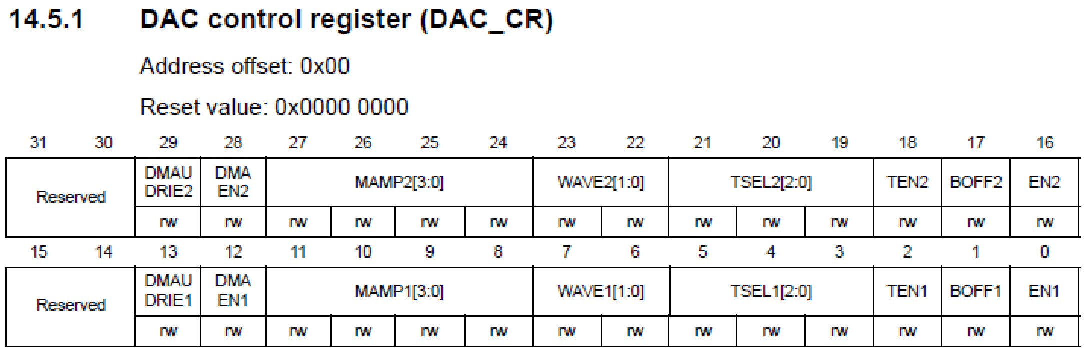
MAMPx[3:0] bits	Digital Amplitude	Analog Amplitude (Volt) (with Vref+ = 3.3V)
0	1	0.0016
1	3	0.0032
2	7	0.0064
3	15	0.0128
4	31	0.0257
5	63	0.0515
6	127	0.1031
7	255	0.2062
8	511	0.4125
9	1023	0.8250
10	2045	1.6483
≥ 11	4095	3.3000

The triangular waveform frequency is related to the frequency of the trigger source.

Triangular waveform



This example describes step by step how to generate a sine waveform.



14.5.1 DAC control register (DAC_CR)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DMAU DRIE2	DMA EN2	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		DMAU DRIE1	DMA EN1	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

Case 1: Software trigger

TSEL1[2:0]: DAC channel1 trigger selection
These bits select the external event used to trigger DAC channel1.

000: Timer 6 TRGO event
001: Timer 8 TRGO event
010: Timer 7 TRGO event
011: Timer 5 TRGO event
100: Timer 2 TRGO event
101: Timer 4 TRGO event
110: External line9
111: Software trigger

If the software trigger is selected, the conversion starts once the SWTRIG bit is set.

SWTRIG is reset by hardware once the DAC_DORx register has been loaded with the DAC_DHRx register contents.

```
//Trigger Selection :Software Trigger

        DAC->CR|=((0x7)<<3);  // Channel 1

        DAC->CR|=((0x7)<<19); // Channel 2
```

WAVE1[1:0]: DAC channel1 noise/triangle wave generation enable These bits are set and cleared by software.

00: wave generation disabled
01: Noise wave generation enabled
1x: Triangle wave generation enabled
Note: Only used if bit TEN1 = 1 (DAC channel1 trigger enabled).

```
//WAVE 1 et WAVE 2 Config : Triangle Wave

	DAC->CR|=((0x3)<<6)+((0x3)<<22); // Triangle wave channel 1 and 2
```

MAMP1[3:0]: DAC channel1 mask/amplitude selector

These bits are written by software to select mask in wave generation mode or amplitude in triangle generation mode.

0000: Unmask bit0 of LFSR/ triangle amplitude equal to 1

0001: Unmask bits[1:0] of LFSR/ triangle amplitude equal to 3

0010: Unmask bits[2:0] of LFSR/ triangle amplitude equal to 7

0011: Unmask bits[3:0] of LFSR/ triangle amplitude equal to 15

0100: Unmask bits[4:0] of LFSR/ triangle amplitude equal to 31

0101: Unmask bits[5:0] of LFSR/ triangle amplitude equal to 63

0110: Unmask bits[6:0] of LFSR/ triangle amplitude equal to 127

0111: Unmask bits[7:0] of LFSR/ triangle amplitude equal to 255

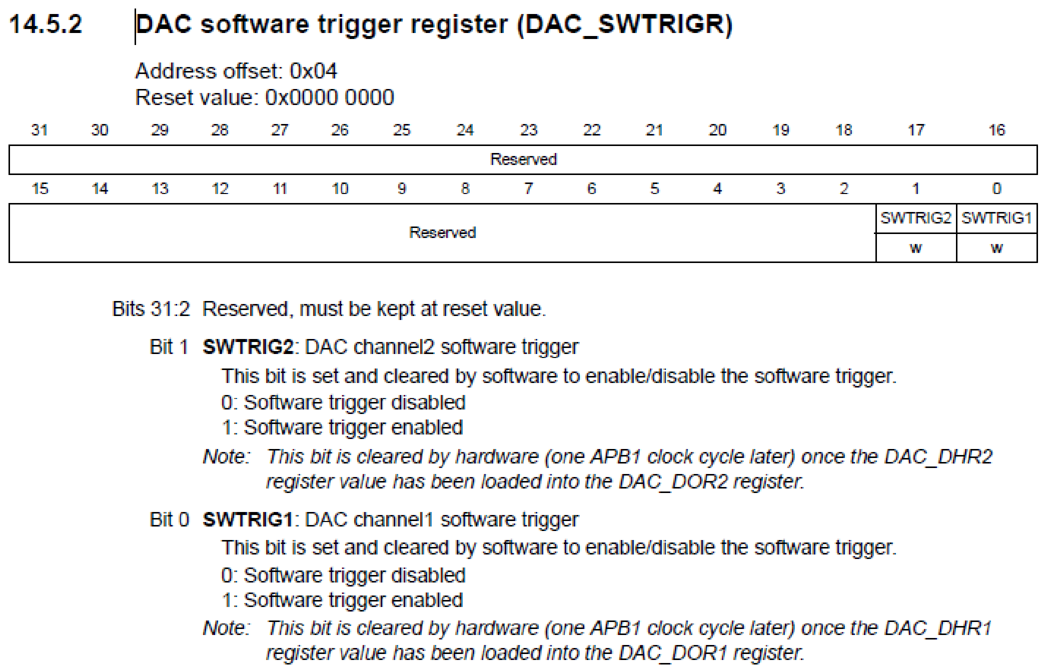
1000: Unmask bits[8:0] of LFSR/ triangle amplitude equal to 511

1001: Unmask bits[9:0] of LFSR/ triangle amplitude equal to 1023

1010: Unmask bits[10:0] of LFSR/ triangle amplitude equal to 2047

≥ 1011: Unmask bits[11:0] of LFSR/ triangle amplitude equal to 4095

```
	//Config Amplitude

	DAC->CR|=((0xA)<<8);  // digital Amplitude 2045; Analog amplitude 1.6483 V

	DAC->CR|=((0xD)<<24); // digital Amplitude 4095; Analog amplitude 3.3 V
```



14.5.2 DAC software trigger register (DAC_SWTRIGR)

Address offset: 0x04
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SWTRIG2	SWTRIG1
														w	w

Bits 31:2 Reserved, must be kept at reset value.

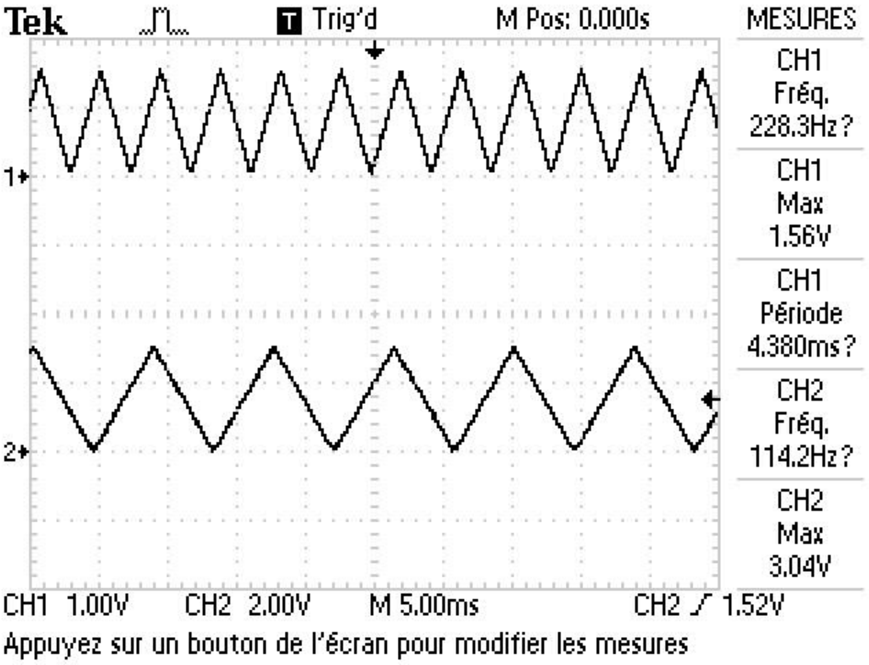
Bit 1 **SWTRIG2**: DAC channel2 software trigger
This bit is set and cleared by software to enable/disable the software trigger.
0: Software trigger disabled
1: Software trigger enabled
Note: This bit is cleared by hardware (one APB1 clock cycle later) once the DAC_DHR2 register value has been loaded into the DAC_DOR2 register.

Bit 0 **SWTRIG1**: DAC channel1 software trigger
This bit is set and cleared by software to enable/disable the software trigger.
0: Software trigger disabled
1: Software trigger enabled
Note: This bit is cleared by hardware (one APB1 clock cycle later) once the DAC_DHR1 register value has been loaded into the DAC_DOR1 register.

```
DAC->SWTRIGR|=0x3;
```

Triangle wave generation by using software trigger

With amplitude 1.6483 V and 3.3V



MESURES
CH1 Fréq. 228.3Hz ?
CH1 Max 1.56V
CH1 Période 4.380ms ?
CH2 Fréq. 114.2Hz ?
CH2 Max 3.04V

CH1 1.00V CH2 2.00V M 5.00ms CH2 ⁄ 1.52V
Appuyez sur un bouton de l'écran pour modifier les mesures

Case 2: Timer 2 as TRGO event

TSEL1[2:0]: DAC channel1 trigger selection
These bits select the external event used to trigger DAC channel1.

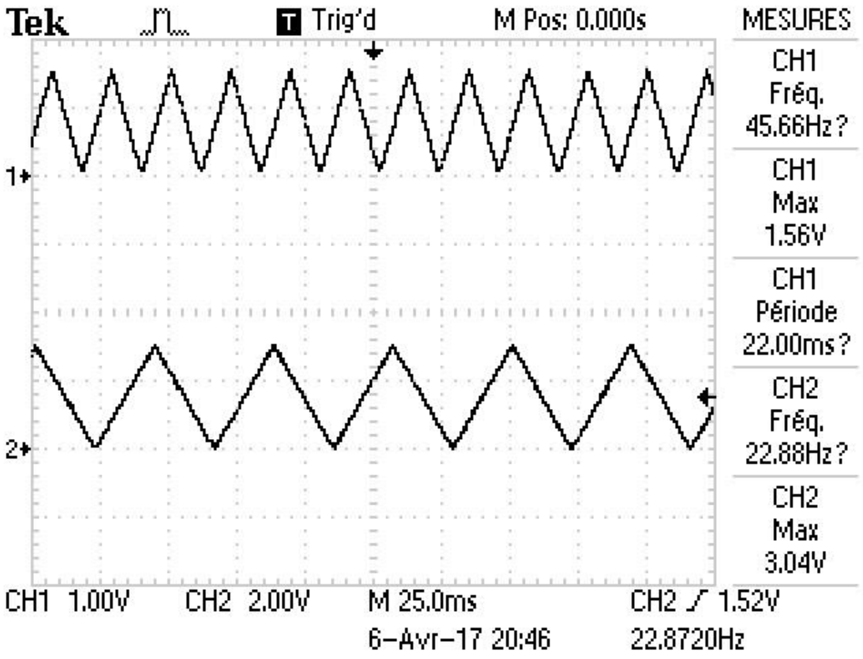
100: Timer 2 TRGO event

```
//Trigger Selection :Timer 2 Trigger

DAC->CR|=((0x4)<<3);

DAC->CR|=((0x4)<<19);
```

Timer 2 initialization :

```
void Timer2_init(void) {

  RCC->APB1ENR |= 0x0001;      // Enable clock for Timer 2

  TIM2->ARR = 84;        // Auto Reload value: 84

  TIM2->CR2    |= 0x0020;      // select TRGO to be update event (UE)

  TIM2->CR1    |= 0x0001;      // Enable Counting

}
```

Triangle wave generation by using Timer 2 as Trgo event

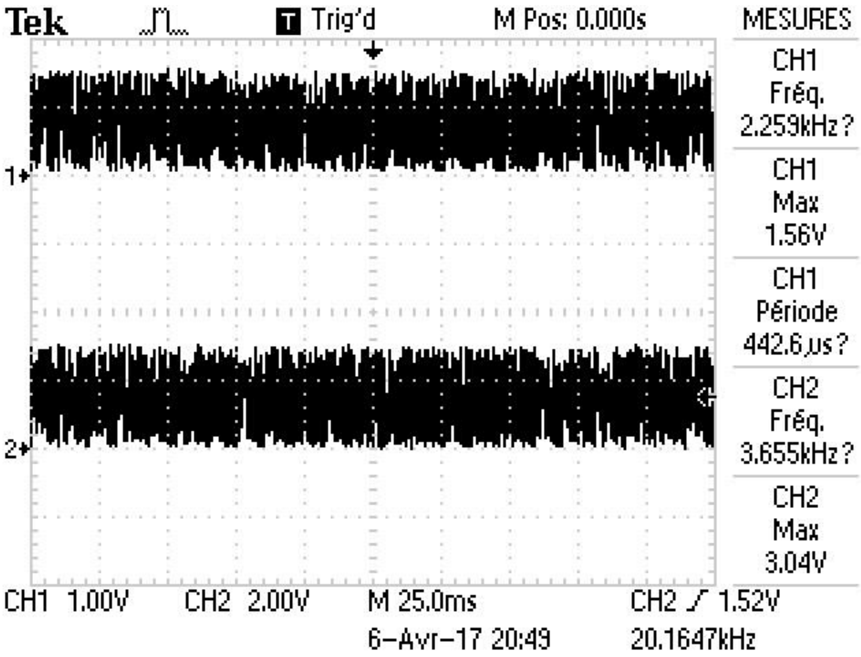


Application2 : noise wave generation

//WAVE 1 et WAVE 2 Config : noise Wave

```
DAC->CR|=((0x1)<<6)+((0x1)<<22);
```

Noise wave generation



References:

RM0090 Reference manual
AN3126 Application note