AAE DSPA LAB 3 instruction: DAC

November 2021

1 Introduction

1.1 Digital to analog conversion

The process of transforming the signal from the digital domain to the analog domain. It can be characterized by two major parameters: resolution and sampling frequency. This process is frequently required in a digital system used to control some external analogue circuitry. The Digital to Analog Converter (DAC) gives mostly a controlled analogue output voltage or current, whose value corresponds to an input digital word. Most DAC are based upon a precision resistor network containing a network of standard resistor values each of which can be switched into or out of circuit according to which bits are set in the input binary word. In the popular 'binary weighted', the resistors have values of R, 2R, 4R, 8R, and so on, in multiples of powers of 2, and each resistor is switched into circuit as its own associated bit is set equal to 1. Another popular solution is R-2R ladder, what requires only two values of resistance. The resolution of such a DAC is defined as the smallest output increment possible, divided by the difference between the maximum and minimum output values. The accuracy or linearity of the converter is defined as the difference between the actual output and the expected output value, measured with any specified digital input value [1].

1.2 Direct digital synthesis

Direct digital synthesis (DDS) is a method employed by frequency synthesizers used for creating arbitrary waveforms from fixed-frequency reference clock. It uses partial phase accumulator to achieve precision desired output frequency of signal.

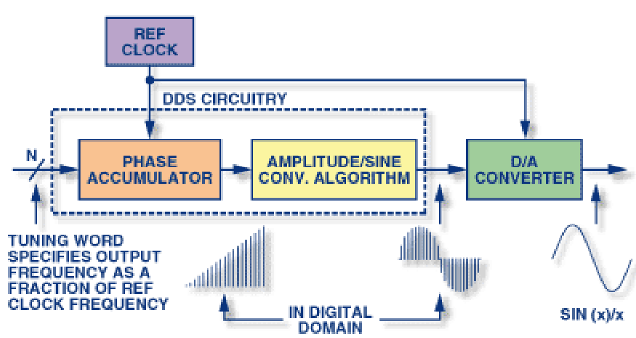

Figure 1: DDS [2].

2 Tasks

2.1 DAC configuration

- Configure DAC channel 2. Enable Buffer, no trigger.

- Use DAC to generate signal achieved from ADC after oversampling. Use Analog Discovery to generate 100 Hz, 1 kHz, 10 kHz and 40 kHz to drive ADC input.
- Measure both generated signal and DAC out signal at oscilloscope. Save plots.
- Explain the behavior of measured signals.

2.2 Sine function generator

- Create 1024 values look-up table with one period sine signal. You can use online generators [3] or Octave, Matlab, Python, etc.
- Configure DMA to transfer data from look-up table to the DAC.
- Configure TIM2 to generate event at 100 kHz and trigger the DAC.
- Measure DAC out signal with using oscilloscope.
- Explain the limitation of achievable frequencies.

2.3 DDS

- Create DDS algorithm to drive DAC. Use look-up table with sine signal.
- Make DDS function configurable - use global variables for frequency and amplitude settings or setup function (example: $ddsSet(freq, amp)$).
- Configure TIM2 to generate event at 100 kHz. Use its interrupt to update DDS function in precision time interval. Do not trigger DAC directly.
- Setup DDS to generate 1 kHz, 1.37 kHz, 10 kHz, and 30 kHz sine signal. Measure the signal with using oscilloscope and spectrum analyzer.
- Connect RC low pass filter to DAC output and repeat measurement at 30 kHz.
- Explain the difference between results with and without RC filter.

2.4 Extra task - sinc correction

- Add sinc ($sin(x)/x$) corrector to DDS amplitude.
- Explain the principle of its operation.

3 Questions

1. What is the advantages of DDS technique? Is it possible to perform DDS using only DMA with DAC in STM32F4?

4 Appendix - CubeIDE peripherals configuration

4.1 Example 1: DAC configuration and usage.

DAC Out2 with using DMA and triggered by TIM2 configuration is presented on Figure 2. To start DAC using HAL, one of the following function should be used:

```
//start DAC channel 2
HAL_DAC_Start(&hdac, DAC_CHANNEL_2);
//start DAC channel 2 with using DMA
HAL_DAC_Start_DMA(&hdac, DAC_CHANNEL_2, (uint32_t*)buf, BUF_SIZE, DAC_ALIGN_12B_R);
//data align to right or left: DAC_ALIGN_12B_R, DAC_ALIGN_12B_L
```

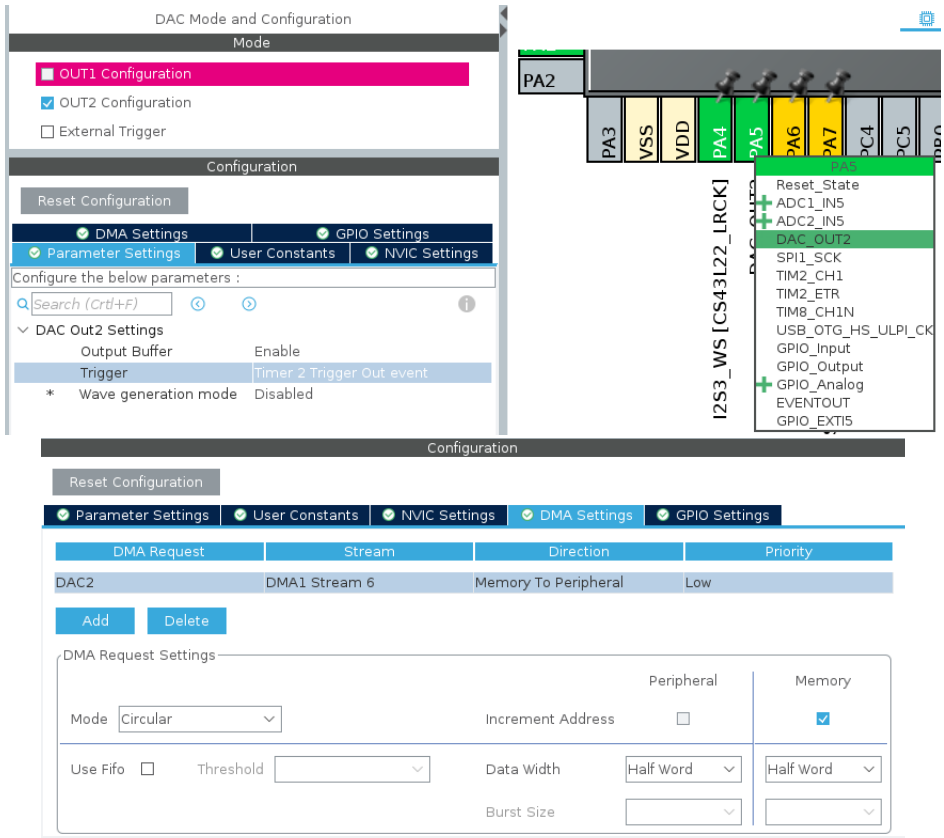

Figure 2: DAC configuration with DMA.

If DMA is not used to send samples to DAC, following function should be used to set DAC value:

```
HAL_DAC_SetValue(&hdac, DAC_CHANNEL_2, DAC_ALIGN_12B_R, value);
```

4.2 Example 2: TIM2 configuration to generate interrupt.

Timer can be used directly to generate an interrupt. Timer should be set as a standard up counter clocked internally. Prescaler and period depends on desired interrupt frequency. Figure 3 shows this example configuration. Remember to start TIM2 in code initialization.

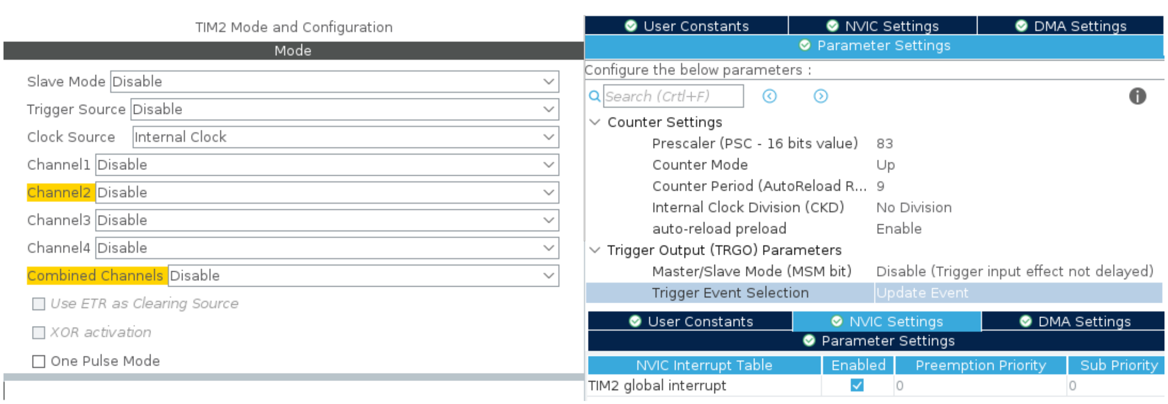

Figure 3: DAC configuration with DMA.

Timer interrupt callback is following:

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
  if(htim == &htim2)
  {
    //TIM2 interrupt
  }
}
```

HAL library uses one concentrated interrupt for all timers, so additional condition has to be written to check if the interrupt is from TIM2.

References

[1] B. Holdsworth and R. Woods, *DAC - https://www.sciencedirect.com/topics/engineering/digital-to-analog-conversion*. Science Direct, 2002.

[2] E. Murphy and C. Slattery, *DDS - https://www.analog.com/en/analog-dialogue/articles/all-about-direct-digital-synthesis.html*. Analog Devices, 2020.

[3] *https://www.daycounter.com/Calculators/Sine-Generator-Calculator.phtml*. Daycounter.