CPE3500: Embedded Digital Signal Processing

<u>Lab Exercise 5: Digital to Analog Conversion through ADC and DMA</u>

<u>Objective:</u>

The purpose of this lab exercise is to introduce digital to analog converter (DAC), Timer and Direct Memory Access (DMA) module configurations in STM32 microcontrollers. The students will be acquiring analog signals generated from a benchtop signal generator and output these signals through DAC while monitoring on the oscilloscope.

<u>Introduction:</u>

Digital to analog conversion is the inverse operation of ADC. While an ADC converts analog voltage to digital data the DAC converts digital numbers to the analog voltage on the output pin. The theoretical background of digital to analog conversion is discussed in the class. Here, we will start with the basic setup to adjust the DAC module of the STM32L476RG microcontroller.

In this exercise students will:

- Learn how to setup DAC and DMA in STM32CubeIde.
- Create a simple application to capture analog signals from ADC and output them through DAC using HAL functions.
- Learn how to configure Timer and DMA for both ADCs and DACs.

<u>Required Equipment:</u>

Personal computer (Windows/Mac OS)

Nucleo-L476RG Board (64-pin)

USB Type-A (or Type-C) to Mini-B cable and Jumper Wires

Agilent 33220A Signal Generator

BNC to Alligator Clips Cable

<u>Procedure:</u>

1. Create a new Project in STM32CubeIDE

Create a blank project following the steps that we learned in previous labs.

2. Configure ADC, DMA and DAC Hardware

We will use PA0 port pin on the Nucleo board for the analog input.

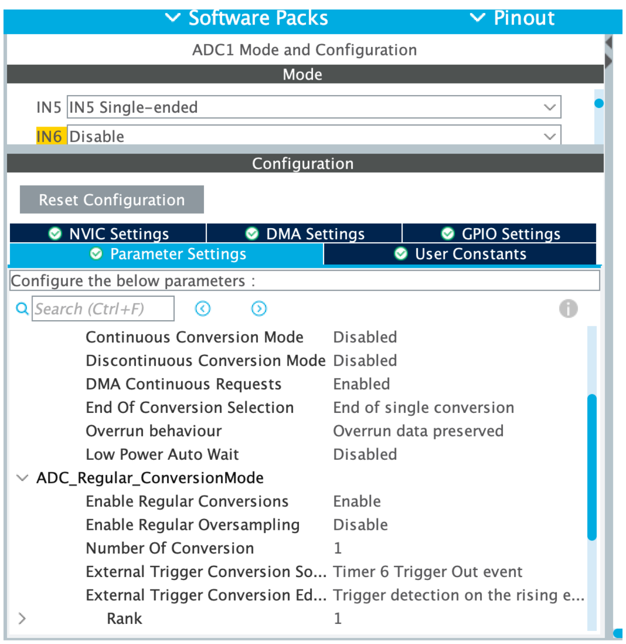
For signal generator settings, we will set Sine wave option with 10 Hz frequency, 3Vpp (3V peak to peak) amplitude and 1.65V offset.

Note: Check load impedance from Utility menu and set it to HighZ. Do not press Output button before making the configurations for the STM32 module!

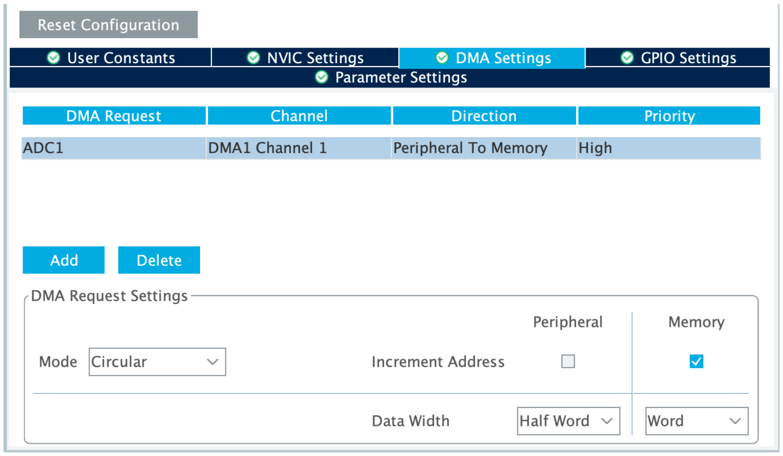
Open .ioc configuration file in the STM32Cubeide project. Find and click on PA0 port pin in STM32L476 pinout view. Configure PA0 pin as analog input by selecting ADC1_IN5 option.

Then, click on ADC tab on the left menu and ADC1 to open mode and configurations window for ADC1. Select "Single-Ended" option for IN5 channel.

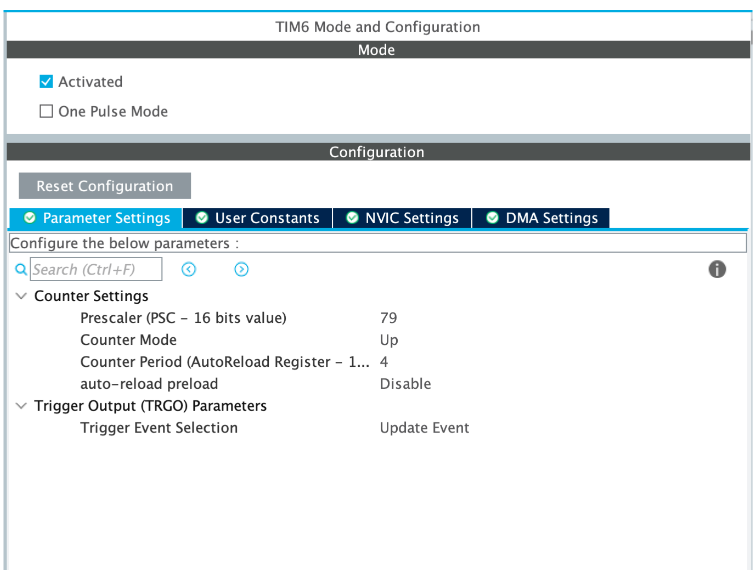
In the Parameter Settings tab, enable *DMA Continuous Request* and select Timer 6 Trigger Out Event in *External Trigger Conversion Source* setting.



In the configuration menu, click on DMA settings and add DMA. Select ADC1 and make Mode as Circular. Change the Data Width into Word.



Enter Timer 6 Mode and Configuration window by clicking on TIM6 in Timers menu. Activate TIM6 and set 79 for Prescaler value for 1MHz timer clock. Also set Counter Period (ARR value) to 4 to get 5 usec (f_{CLK}*(ARR+1))timer trigger period and select Update Event option on *Trigger Event Selection*.

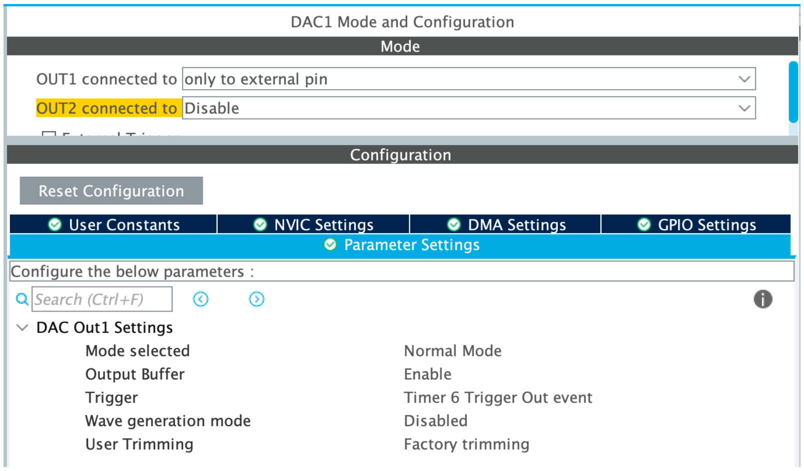


For DAC configuration, we will use PA4 port pin (A2 label on Nucleo board). Connect two jumper wires on PA4 port pin header and GND which will be used for Oscilloscope probing.

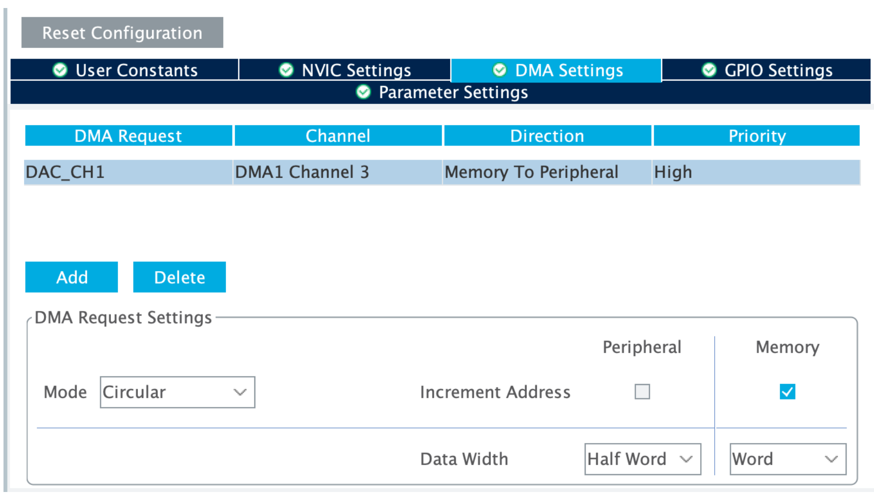
Click on PA4 port pin on the pinout view and select DAC1_OUT1 option. Go to DAC1 mode and configuration menu in the Analog section.

Set "*OUT1 connected to"* mode to only to external pin.

In the parameter tab, make the below settings:



In the DAM settings tab add DMA. Select DAC1_CH1 and make Mode as Circular. Change the Data Width into Word as show below.



Save or generate the code and switch to C perspective to edit the code.

We will create two buffers for ADC and DAC.

Inside /* USER CODE BEGIN PD */ section, define two constants for buffer half size and full size as follows:

#define BUFFER_HALFSIZE 500

#define BUFFER_SIZE 1000

Inside /* USER CODE BEGIN PV */ section, create the buffers:

```
uint32_t adc_buffer[BUFFER_SIZE];
uint32_t dac_buffer[BUFFER_SIZE];
```

Inside /* USER CODE BEGIN 2 */ section, start TIM, ADC and DAC:

```
HAL_TIM_Base_Start(&htim6);
HAL_ADC_Start_DMA(&hadc1, adc_buffer, BUFFER_SIZE);
HAL_DAC_Start_DMA(&hdac1, DAC_CHANNEL_1, dac_buffer, BUFFER_SIZE,
DAC_ALIGN_12B_R);
```

Inside /* USER CODE BEGIN 4 */, define two callback functions for ADC half and full buffer complete events as below:

```
void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef *hadc)
{
	for (int n=0; n<BUFFER_HALFSIZE; n++)
	{
		dac_buffer[n] = adc_buffer[n];
	}
}

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc)
{
	for (int n=BUFFER_HALFSIZE; n<BUFFER_SIZE; n++)
	{
		dac_buffer[n] = adc_buffer[n];
	}
}
```

This code will copy the ADC buffer to DAC buffer to prepare the DMA for analog output signal generation through the DAC after successful completion of each ADC operation.

Start debugging the code. Turn on the signal generator and oscilloscope. Connect oscilloscope channel 1 probe to the signal generator output signal (A0 and GND on Nucleo) and channel 2 probe to DAC output port (A2 and GND on Nucleo).

Task-1:

Take a picture of the oscilloscope display while the ADC input (10 Hz sine wave) and DAC output are observed together with the same reference GND level.

Task-2:

Although it is noticeable that there is a delay between ADC input and DMA output, it is difficult to measure on sine waveforms. Change the waveform into Square wave with the same parameters.

Zoom in to the rising edges of the square waves and take a picture of this view.

Using Oscilloscope vertical cursors, measure the time delay. Comment on the cause of this delay between ADC input and DAC output.

Task-3:

Switch back to Sine wave mode with 100 Hz frequency. Change the DAC output amplitude to half of the input sine amplitude by dividing the adc_buffer values before assigning them to dac_buffer inside the callback functions.

Get these new signals on the scope and take a picture. Comment on the DC value differences of two signals.

Task-4:

Undo the amplitude change in Task 3 and modify the callback functions (by determining the digital equivalent of the midpoint of the sine wave) to reverse the negative cycle sine wave into positive cycle.

Take a picture of the Oscilloscope view.

Lab Exercise 5 Report:

Prepare a lab report (as single pdf file) consisting of the followings and upload it to the D2L dropbox.

- Cover Sheet
- Task-1: Insert the picture of the oscilloscope display with label and explanation.
- Task-2: Insert the picture of the oscilloscope display with label and explanation. Report the measured delay and comment on the result.
- Task-3: Insert the picture of the new signals. Comment on the DC value differences.
- Task-4: Include the modified code in callback functions. Insert the picture of the oscilloscope display.
- Conclusion (1-paragraph)