Lab 4: Analog Peripherals

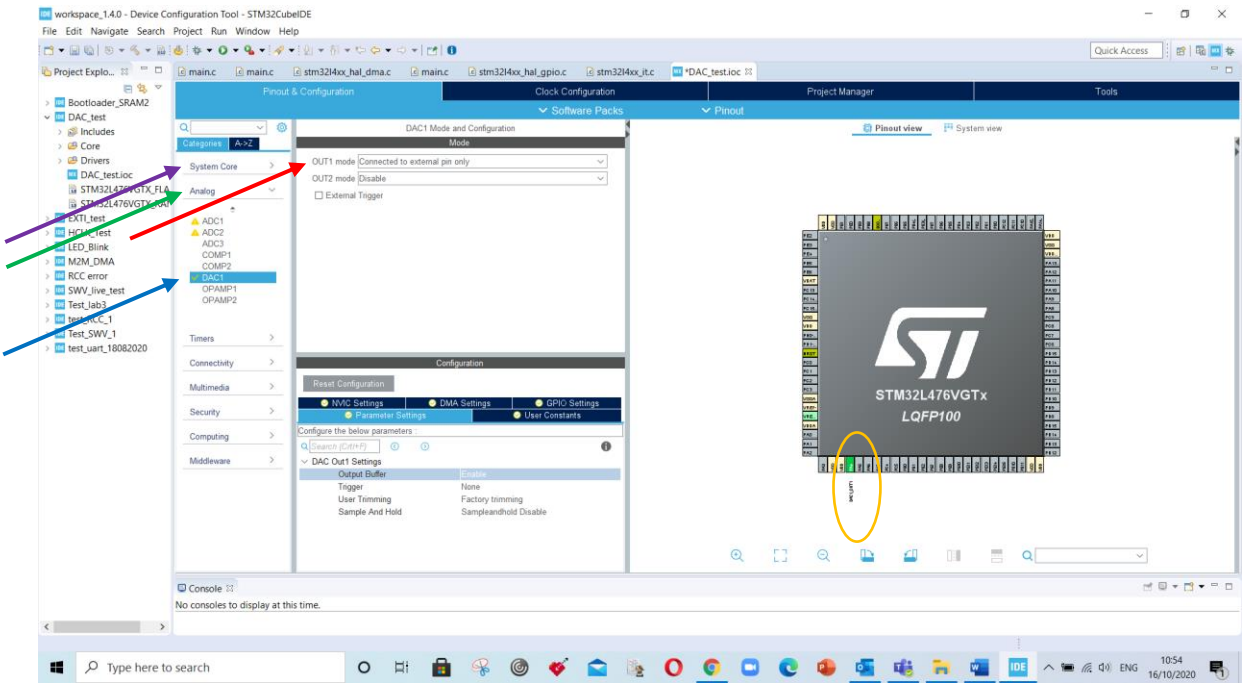
Activity 1: Digital to Analog Converter

Aim: Learn to how to configure DAC on STM32L4 DISCO in STM32CubeIDE.

Objectives:

1- Learn how to configure the DAC.
2- Generating a sawtooth wave using DAC.
3- Testing and validation the functionality by SWV Data Trace Timeline graph.

Step 1: Create a project in STM32CubeIDE. Go to 'System Core' menu and expand 'Analog' section then select 'DAC1' and in DAC1 ***Mode*** dialog box for "**OUT1 Connected to**" select 'only to external pin' for OUT1 Mode. Pin PA4 gets highlighted green in pinout view. Lastly set the HCLK to clock speed of 80 MHz in Clock Configuration Tab. Observe the Configuration options in Configuration Tab.



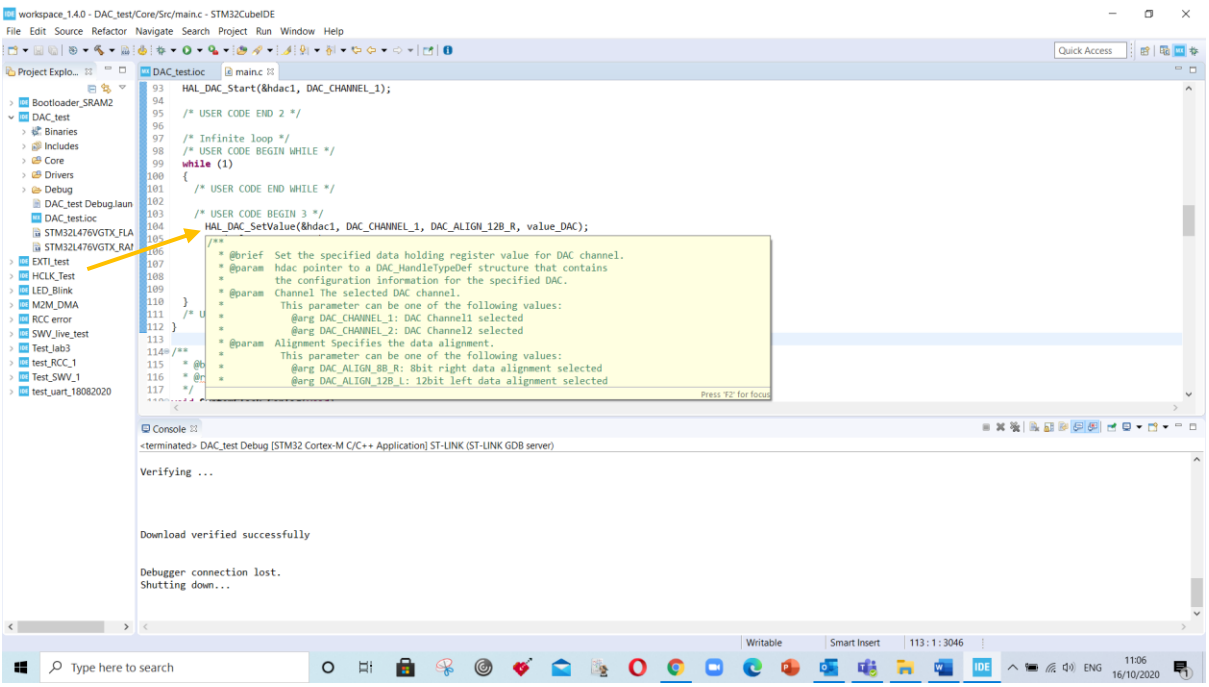
Step 2: Generate the code and add the code section given below in shown comments area:

```
  /* USER CODE BEGIN 1 */
uint16_t value_DAC = 0;    /* 16-bit unsigned integer variable to set the DAC
values */
uint32_t state_DAC = 0;    /* 32-bit unsigned integer variable to hold the status
of DAC for SWV*/
```

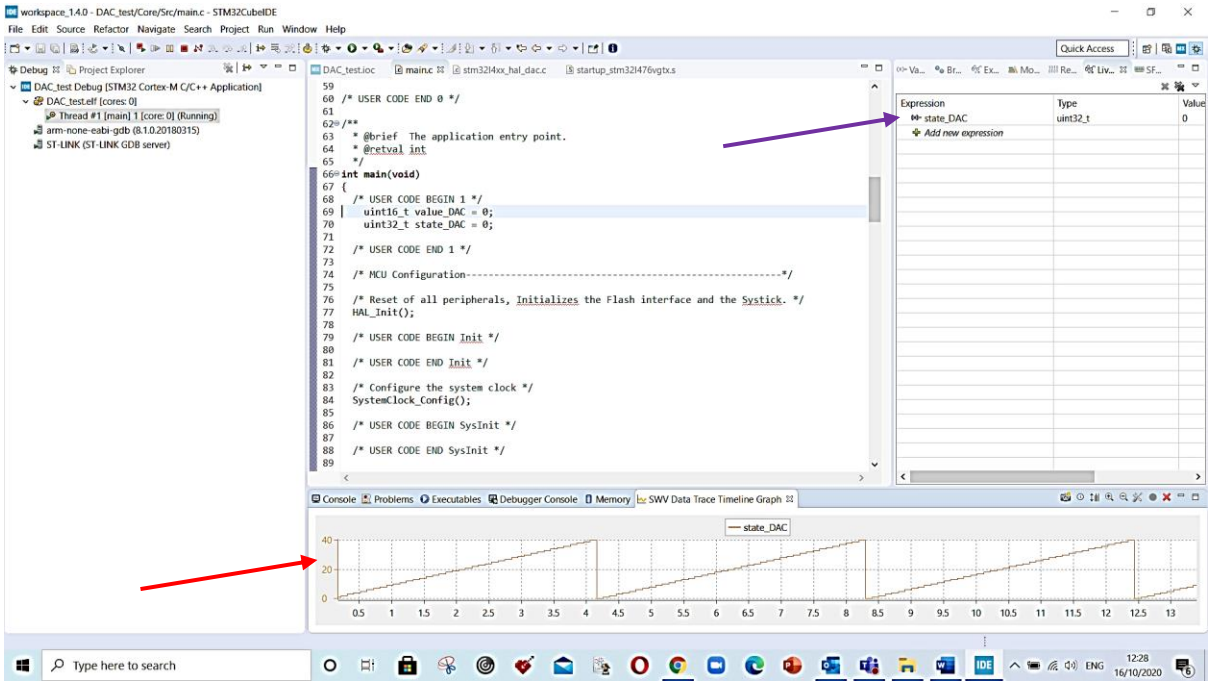
```
/* USER CODE BEGIN 2 */
  HAL_DAC_Start(&hdac1, DAC_CHANNEL_1); /* start the DAC channel 1*/
```

```
    /* USER CODE BEGIN 3 */
HAL_DAC_SetValue(&hdac1, DAC_CHANNEL_1, DAC_ALIGN_12B_R, value_DAC); /*setvalues*/
        if (value_DAC < 40)
              value_DAC++;
        else
              value_DAC=0;
        state_DAC= HAL_DAC_GetValue(&hdac1, DAC_CHANNEL_1);
        HAL_Delay(100);
```

Place the cursor on the fuction to know the argument details. (use ‘F3’ Key to open declaration)



Step 3: Run to build the project and debug the code. Add the 'state_DAC' variable to '**live expression window**' and view the trace 'SWV Data Trace Timeline Graph'



Optional To do: Use the DAC with DMA and generate a sawtooth waveform.

Activity 2: Analog to Digital Converter

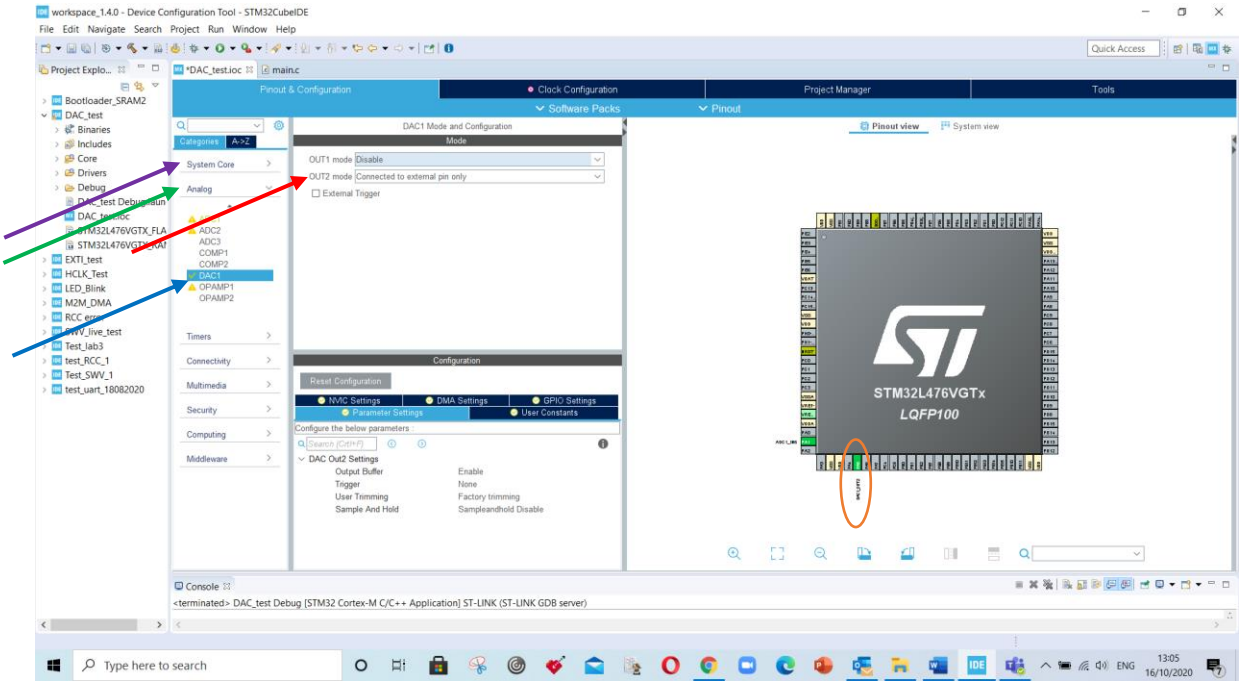
Aim: Learn to how to configure ADC.

Objectives:

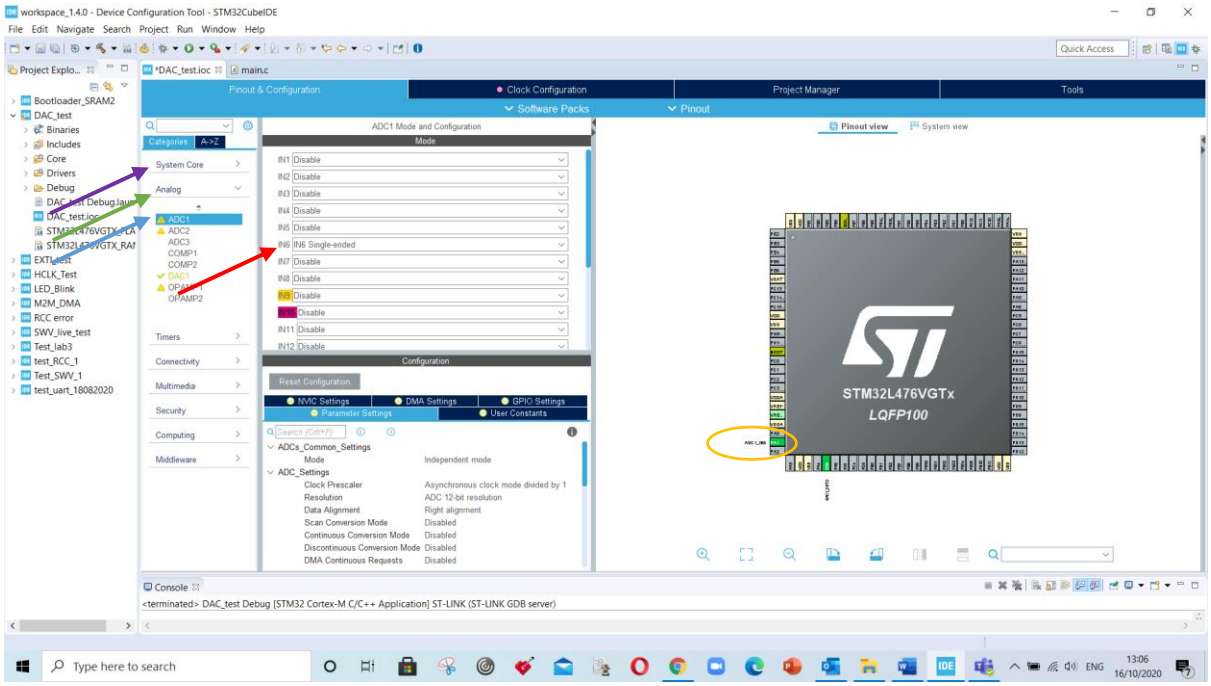
1- Learn how to configure the ADC with DMA
2- Capturing the sawtooth waveform generated by DAC
3- Testing the functionality by SWV Data Trace Timeline graph

---

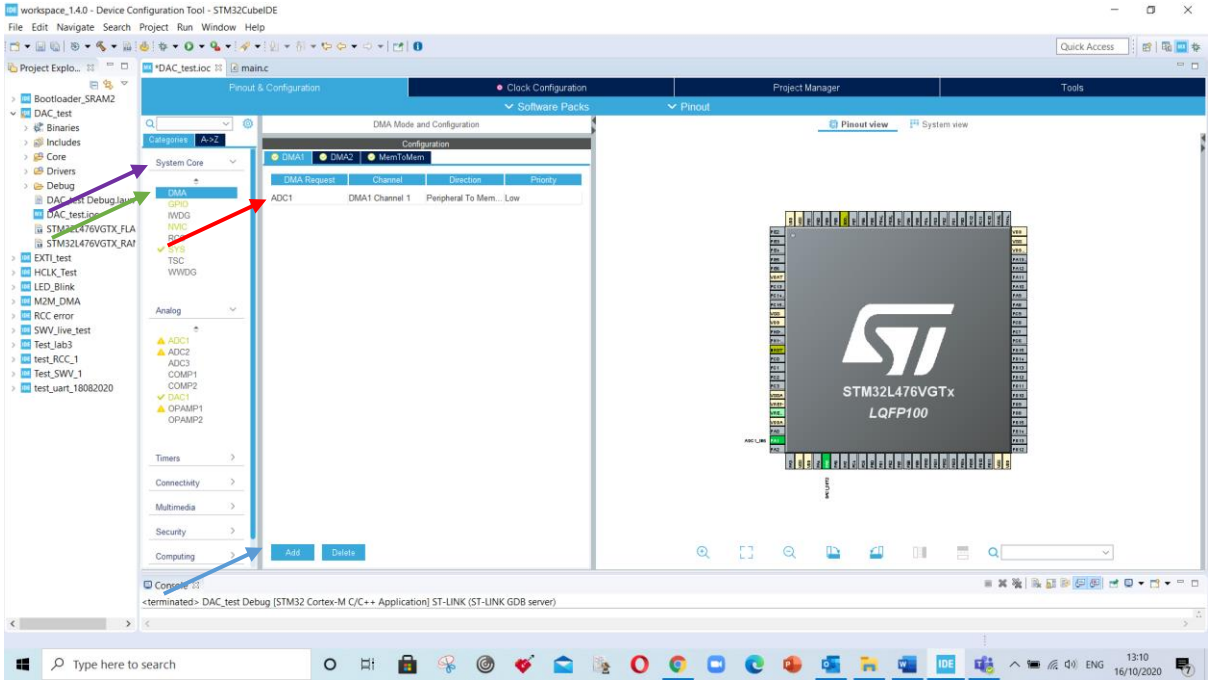
Step 1: create a new project, Go to 'System Core' menu and select 'Analog' then select 'DAC1' and in DAC1 Mode dialog box select **OUT2 connected to** option of 'only to external pin'. Pin PA5 gets highlighted green in pinout view.



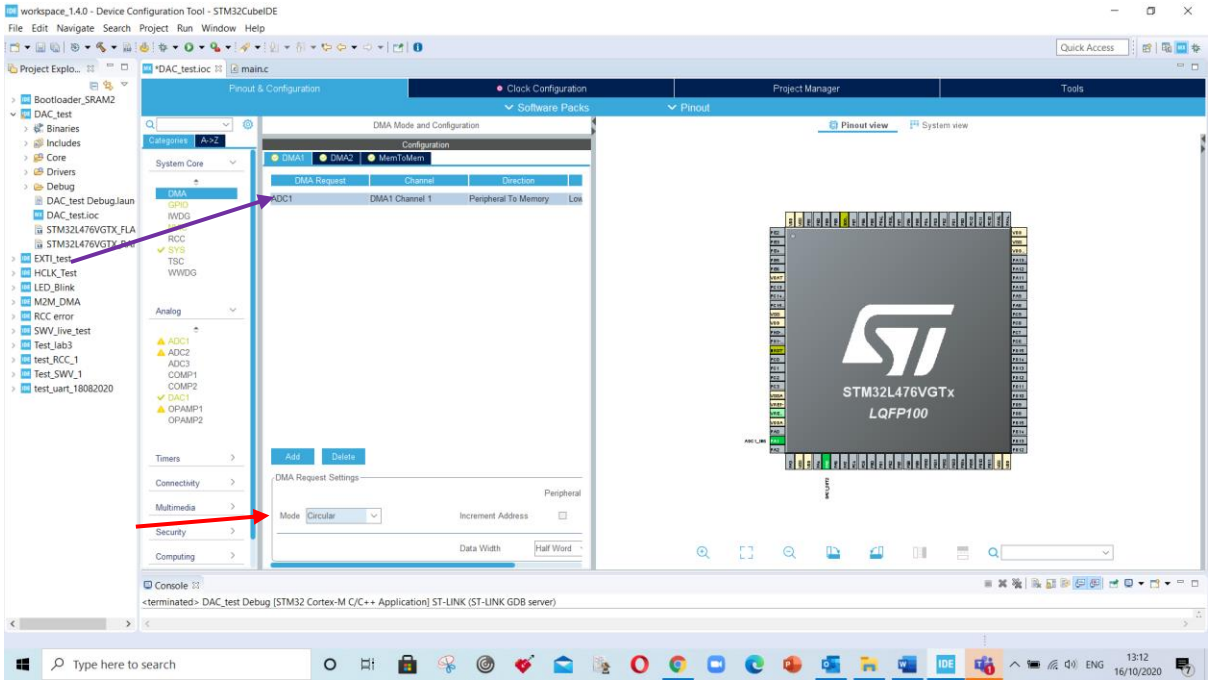
Step 2: Go to ‘System Core’ menu and expand ‘Analog’ then select ‘ADC1’ and select ’IN6 single ended’ in ADC1 Mode and configuration dialog box. Pin PA1 gets highlighted in green in pinout view.



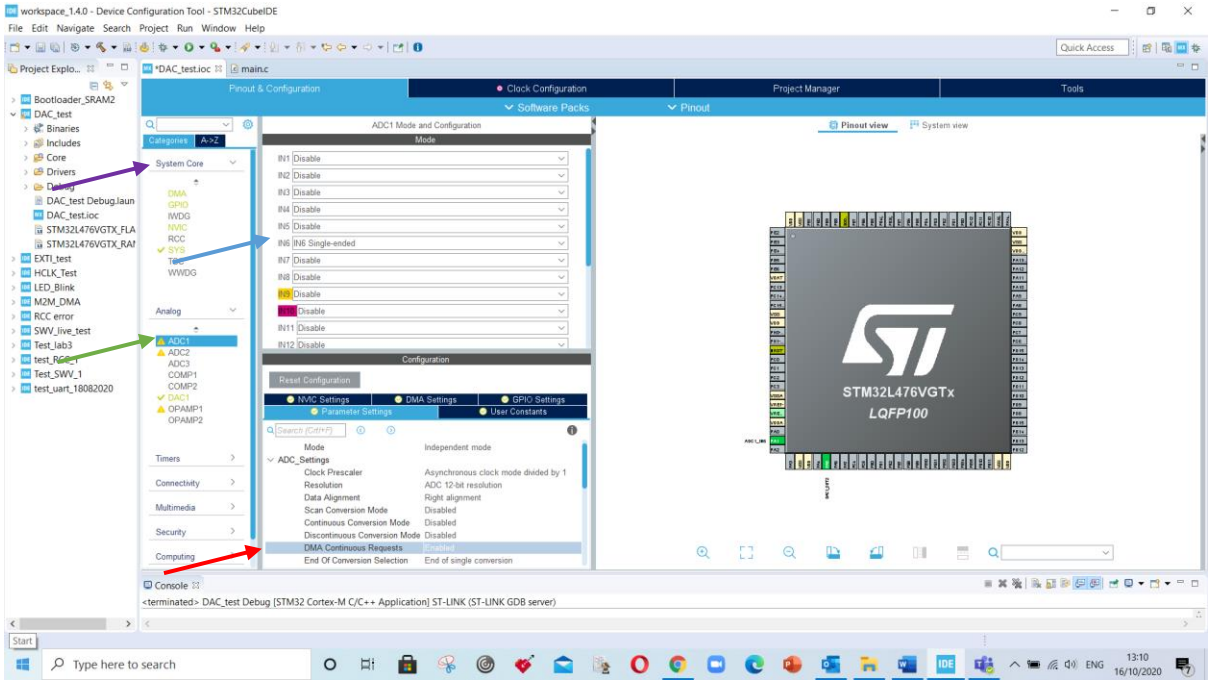
Step 3: Go to ‘System Core’ menu and select ‘DMA’ then ‘Add’ and select ‘ADC1’ in DMA Mode and configuration dialog box.



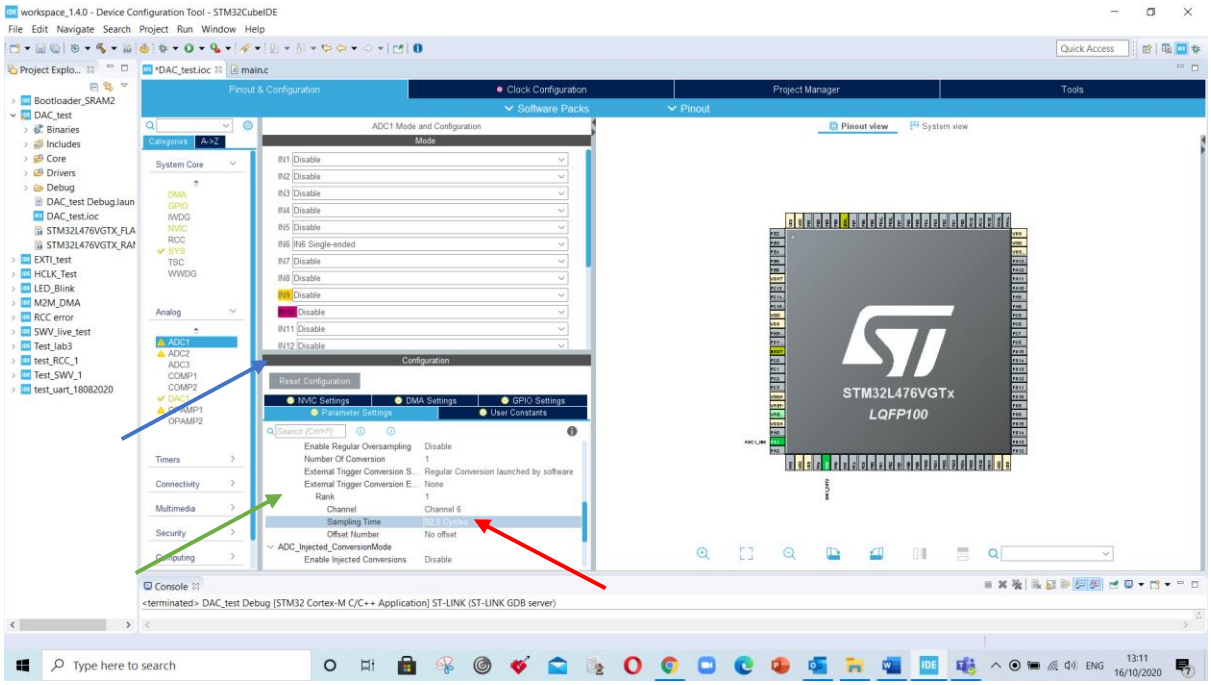
Step 4: Go to ‘ADC1’ setting and select ‘Circular’ mode.



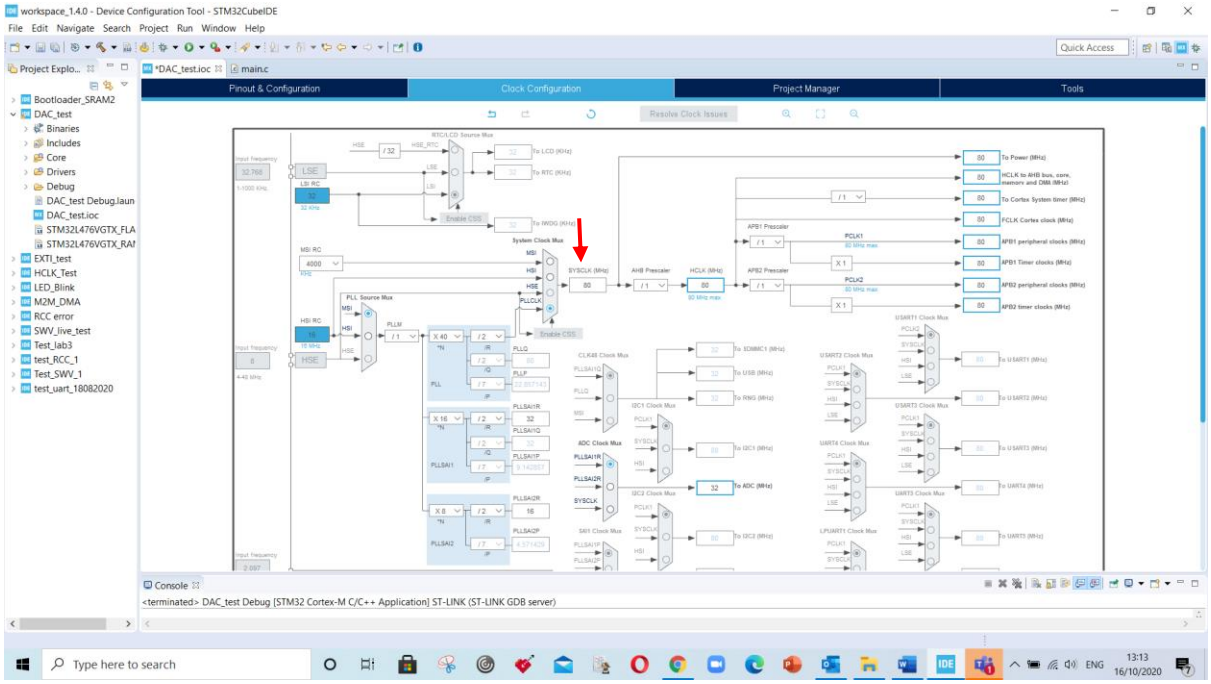
Step 5: Go to ‘System Core’ menu and select ‘Analog’ then select ‘IN6’ setting and Enable ’DMA Continuous Requests’ in ADC1 configuration dialog box.



Step 6: In 'IN6' configuration expand 'Rank' option and set the sampling time of '92.5 cycles'.



Step 7: Set the HCLK to '80MHz'.



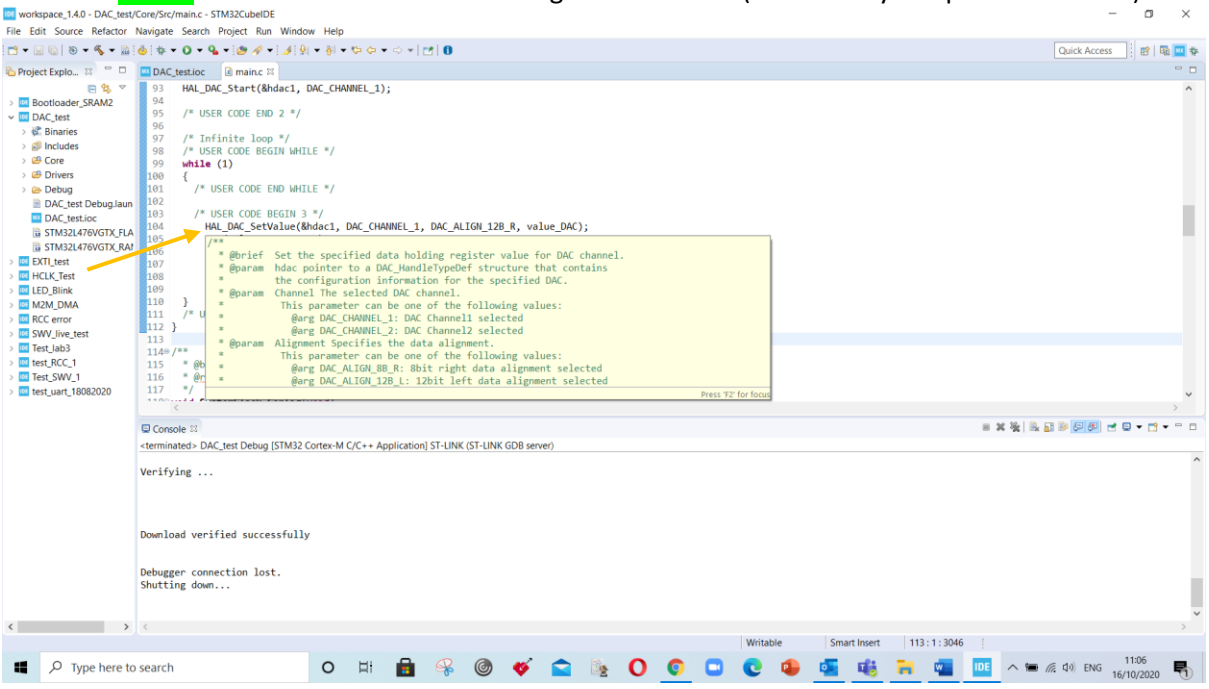
Step 8: Generate the code and add the code section given below:

```
/* USER CODE BEGIN PV */
uint16_t value_DAC = 0; // Variable for DAC value setting
uint16_t value_ADC =0;  // Variable for ADC reading
uint32_t state_DAC = 0; // Variable to plot the values in SWV window
```

```
  /* USER CODE BEGIN 2 */
HAL_ADCEx_Calibration_Start(&hadc1, ADC_SINGLE_ENDED); // Calibrate ADC
HAL_ADC_Start_DMA(&hadc1, (uint32_t *) &value_ADC, 1); // Start ADC with DMA
HAL_DAC_Start(&hdac1, DAC_CHANNEL_2); // Start the DAC
```

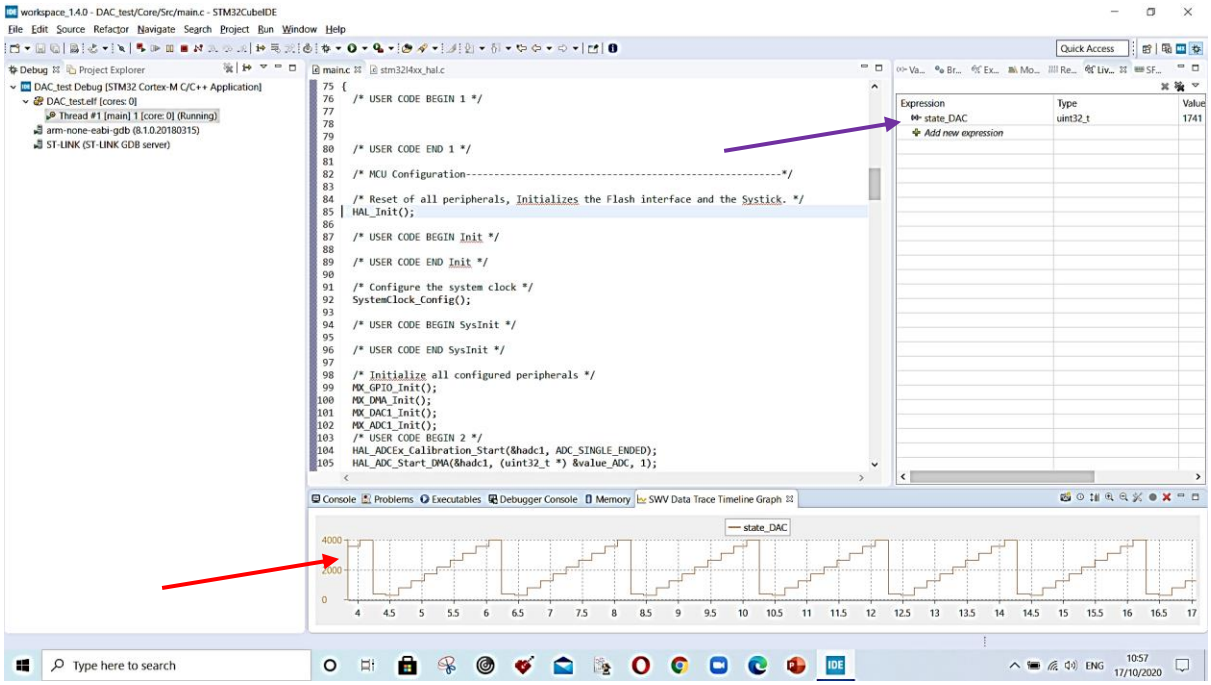
```
/* USER CODE BEGIN 3 */
HAL_DAC_SetValue(&hdac1, DAC_CHANNEL_2, DAC_ALIGN_12B_R, value_DAC); // Set value
if (value_DAC < 4096)
      value_DAC+=500;
      else
      value_DAC=0;
HAL_Delay(200);
HAL_ADC_Start(&hadc1); // Start ADC to read the value
state_DAC = (uint32_t)value_ADC; // Use typecasting to assign the value
```

Place the cursor on the fuction to know the argument details. (use 'F3' Key to open declaration)



Step 9: Place the jumper on pin PA1 and PA5 in STM32DISCO board to read the value of DAC through ADC. (use the jumper available on GND pins)

Step 10: Run and build the project and debug the code. Add the 'state_DAC' variable to 'live expression window' and view the trace 'SWV Data Trace Timeline Graph'.



To do: Experiment the different values of sampling cycle in ADC and explain effect of different sampling time.

Appendix A: Declaration of Serial Wire interface (if required)

Go to 'System Core' menu and select 'SYS' then select 'Serial Wire' in SYS mode and configuration menu. This step speeds up the Debug process as this mode is not turned on by default.