Course: Digital Signal Processing

**Session 3: Basic Setting up of Hardware/Software for CT to DT and Difference Equation**

Dept.: Electrical & Electronics Engineering V1.4

# 1 Objectives/Outcome:

1) Understand the blocks involved in DSP –A/D, Processing and D/A: Use Microcontroller IDE and setup A/D, timer, D/A.
2) Learn sampling and converting to discrete samples on microcontroller board
3) Plot relation between (analogue)continuous time input frequency to discrete time frequency on conversion
4) Learn how difference equations can realise low pass filters

# 2 Tasks

## *2.1 Task1: Generate the code from IDE using STM32CubeIDE and Update SW*

Follow the steps to setup as given in the last section.

Measure the sampling frequency using the GPIO pin: PC3.

Set the sampling frequency to 8KHz by playing with preset, timer count and clock frequency. Note the setting. **Input signal must be in the range 0 to 3V.**

## *2.2 Task2: Take plots of input and output*

Observe and take plots of input and output signals from oscilloscope for 6 scenarios of inputs with same sampling frequencies:

Scenarios of $F_{in} < 2F_s$: 1) $F_{in} = F_s$ 2) $F_{in} = 1.5\ F_s$

Scenarios of $F_{in} = \frac{F_s}{2}$

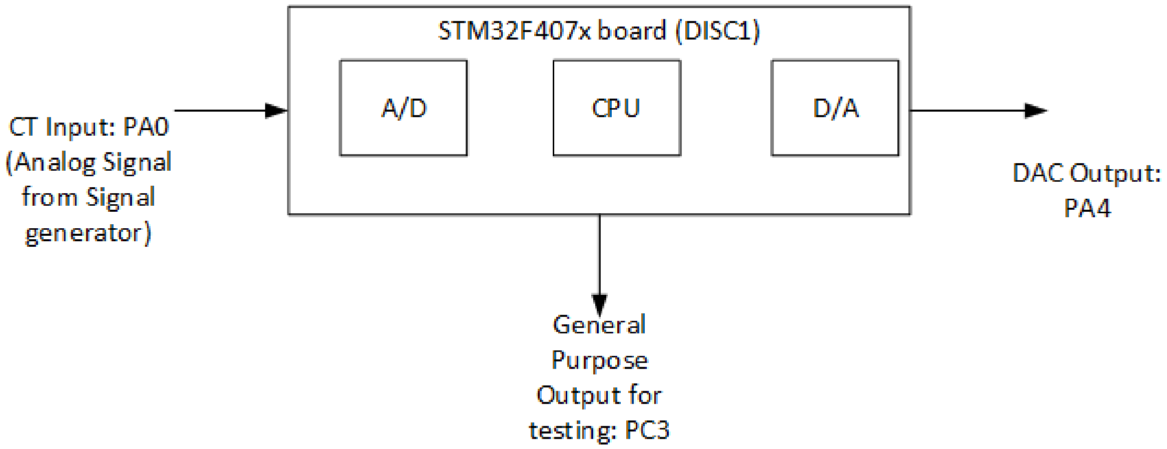
Scenarios of $F_{in} > 2F_s$: 1) $F_{in} = 3F_s$ 2) $F_{in} = 4\ F_s$ 3) $F_{in} = 5\ F_s$

## *2.3 Task3: Plot output frequency vs input frequency*

Go from 1Hz to 2Fs with different steps on input frequency and take readings of frequency of output.

| Sl. No | Freq Input (Hz) | Freq Output (Hz) |
|---|---|---|
| 1 | 100 | |
| 2 | 500 | |
| 3 | 1000 | |
| 4 | 2000 | |
| …. | …….. | …. |
| | 4000 | …. |
| …. | ….. | |
| | 16000 | |

Observe the repeated pattern and infer the same.


STM32F407x board (DISC1)
A/D
CPU
D/A
CT Input: PA0 (Analog Signal from Signal generator)
DAC Output: PA4
General Purpose Output for testing: PC3

## 2.4 *Task4: Realise moving average difference equation*

In the Timer ISR code, implement the moving average of last 2 input sequences as difference equation that is:

$$y[n] = \frac{x[n] + x[n-1]}{2}$$

Plot the frequency response for DAC output to ADC input by giving sinusoidal input to A/D and measuring the output for different frequency inputs.

Peak to Peak Input: (0 to 3V signal min and max levels): <3V (Record this)

| Sl. No | Frequency input | Output Peak to Peak (V) | Gain = Output/Input | Gain_dB = 20 log (Gain) |
|---|---|---|---|---|
| 1 | 1Hz | | | |
| 2 | 200Hz | | | |
| 3 | 500Hz | | | |
| 4 | | | | |
| …. | | …. | | |
| | | …. | | |
| …. | | | | |
| | | | | |

**Validate the frequency response against the system response plot from MATLAB.**

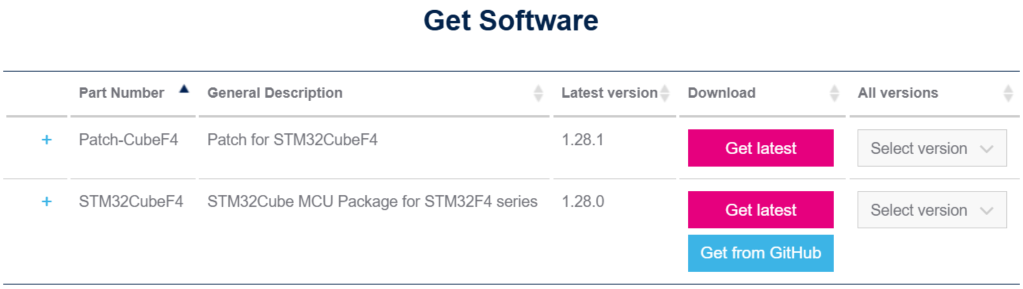
(Note it is good enough if you consider inputs from frequency 0 to $\frac{fs}{2}$ where $fs$ is sampling frequency.)

## Prerequistes: (INSTALLATION Details)

**Install STM32CubeIDE** from: https://www.st.com/en/development-tools/stm32cubeide.html

### Installation of Embedded Software Package

You can download the Zip file from https://www.st.com/en/embedded-software/stm32cubef4.html

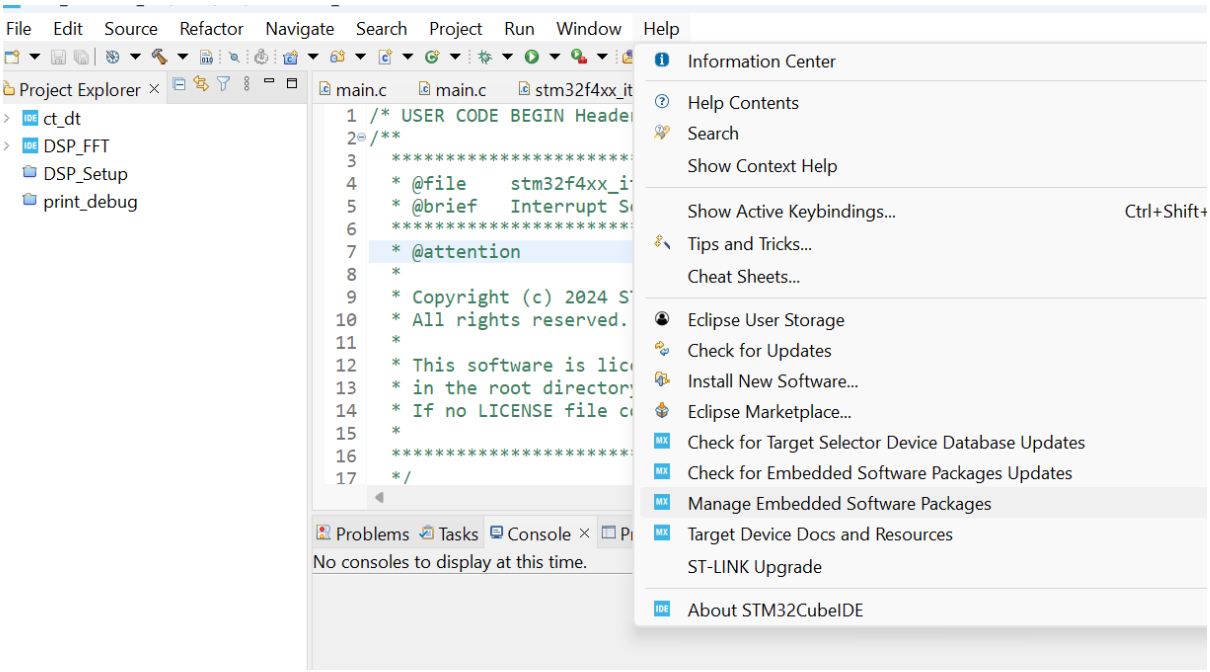


**Get Software**

| | Part Number ▲ | General Description | Latest version | Download | All versions |
|---|---|---|---|---|---|
| + | Patch-CubeF4 | Patch for STM32CubeF4 | 1.28.1 | Get latest | Select version |
| + | STM32CubeF4 | STM32Cube MCU Package for STM32F4 series | 1.28.0 | Get latest<br>Get from GitHub | Select version |

It is important that you install STM32F4xx package for the creation of the project. If you have installed propely you would find installations in the appropriate folder : **C:\Users\mjvrangan\STM32Cube\Repository (Under users, it will be your own name instead of “mjvrangan”)**

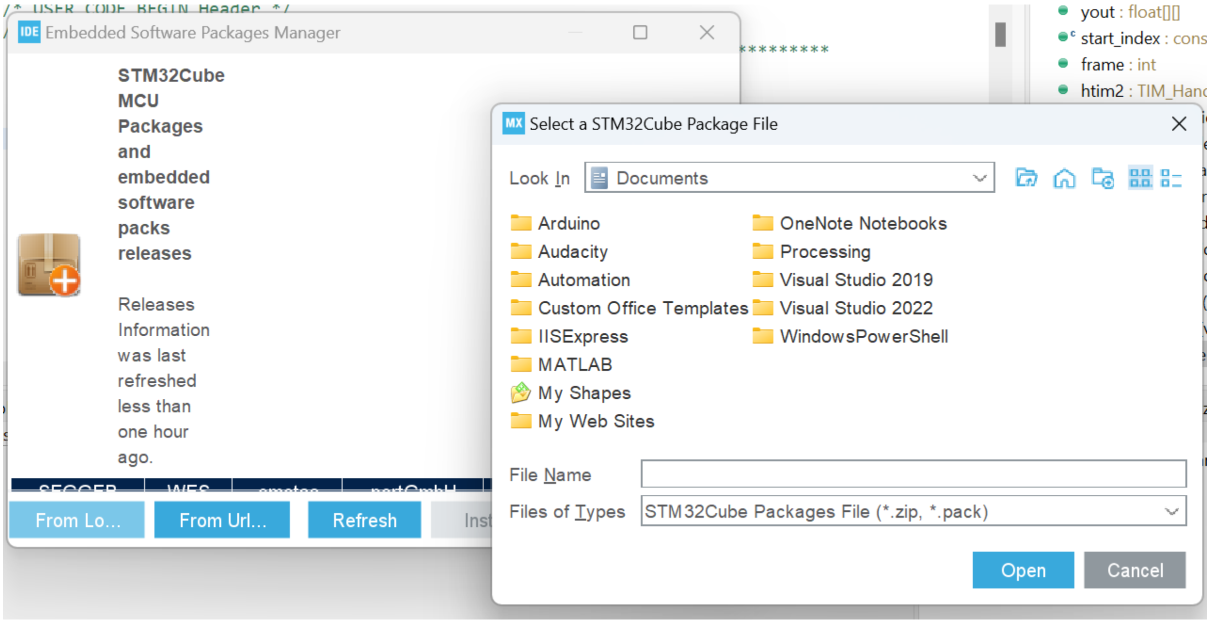
### If you click on STM32CubeIDE, you can still install the package via menu.



```
/* USER CODE BEGIN Heade
/**
  **********************
  * @file    stm32f4xx_i
  * @brief   Interrupt S
  **********************
  * @attention
  *
  * Copyright (c) 2024 S
  * All rights reserved.
  *
  * This software is lic
  * in the root director
  * If no LICENSE file c
  *
  **********************
  */
```

No consoles to display at this time.

You can select “From Local” and the Zip file. The Zip file can be downloaded from:



Embedded Software Packages Manager

STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed less than one hour ago.

Select a STM32Cube Package File

Look In: Documents

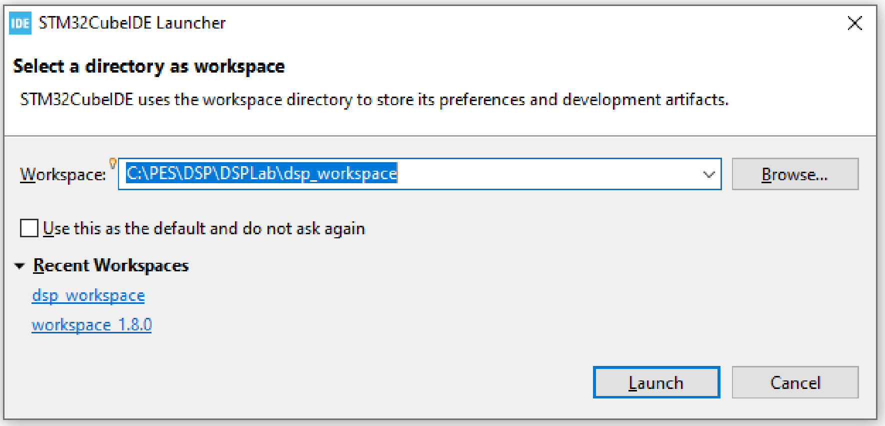
Arduino
Audacity
Automation
Custom Office Templates
IISExpress
MATLAB
My Shapes
My Web Sites
OneNote Notebooks
Processing
Visual Studio 2019
Visual Studio 2022
WindowsPowerShell

File Name:

Files of Types: STM32Cube Packages File (*.zip, *.pack)

## Steps for creating workspace and creating project

### Step1: Workspace creation



STM32CubeIDE Launcher

**Select a directory as workspace**

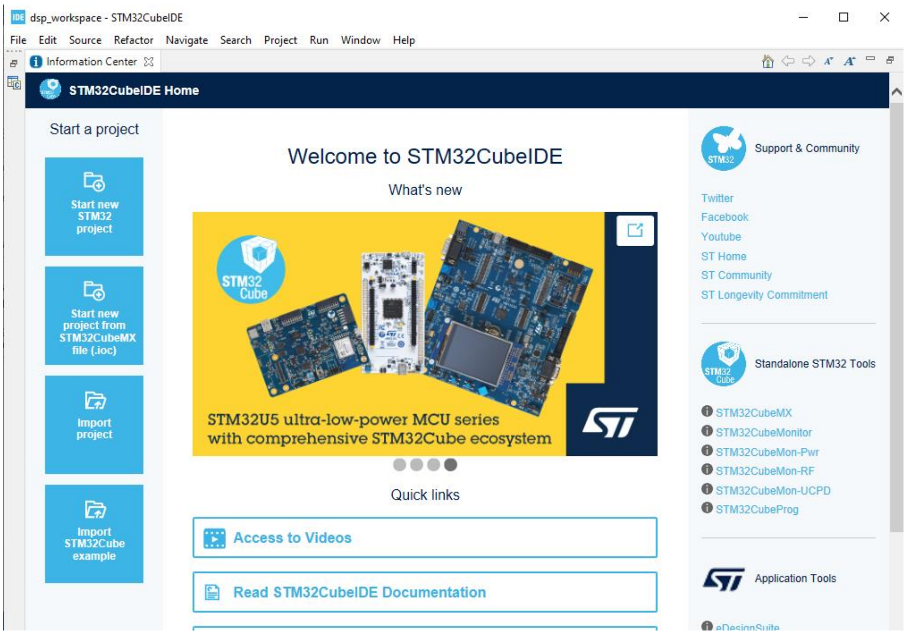
STM32CubeIDE uses the workspace directory to store its preferences and development artifacts.

Workspace: C:\PES\DSP\DSPLab\dsp_workspace

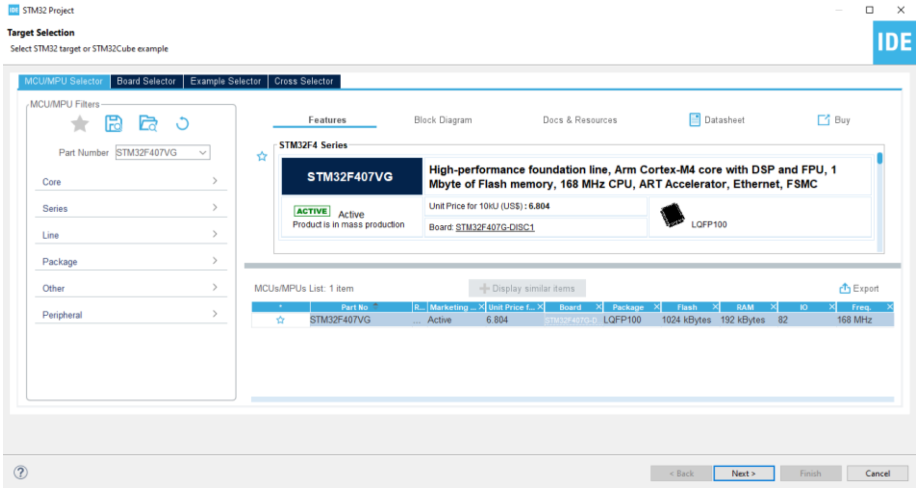
□ Use this as the default and do not ask again

Recent Workspaces
dsp_workspace
workspace_1.8.0

### Step2: Click on Start Project



## Step3: Type STM32F407VG in Part number and click NEXT



Target Selection

Select STM32 target or STM32Cube example

Part Number: STM32F407VG

STM32F4 Series

STM32F407VG — High-performance foundation line, Arm Cortex-M4 core with DSP and FPU, 1 Mbyte of Flash memory, 168 MHz CPU, ART Accelerator, Ethernet, FSMC

ACTIVE Active
Product is in mass production

Unit Price for 10kU (US$) : 6.804

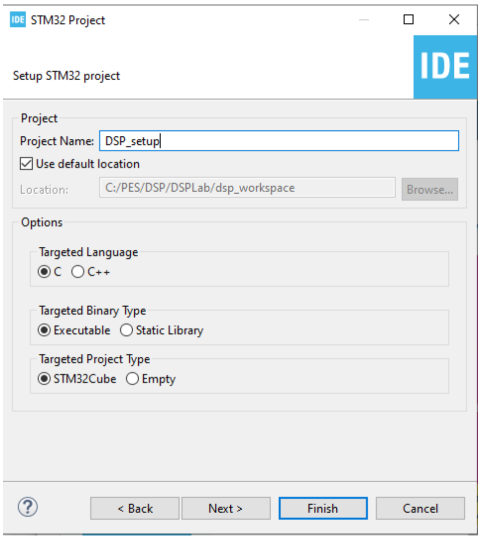
Board: STM32F407G-DISC1

LQFP100

MCUs/MPUs List: 1 item

| * | Part No | R... | Marketing ... | Unit Price f... | Board | Package | Flash | RAM | IO | Freq. |
|---|---|---|---|---|---|---|---|---|---|---|
| ☆ | STM32F407VG | ... | Active | 6.804 | STM32F407G-D... | LQFP100 | 1024 kBytes | 192 kBytes | 82 | 168 MHz |

## Step4: Selct Project name and click NEXT



Setup STM32 project

Project

Project Name: DSP_setup

☑ Use default location

Location: C:/PES/DSP/DSPLab/dsp_workspace

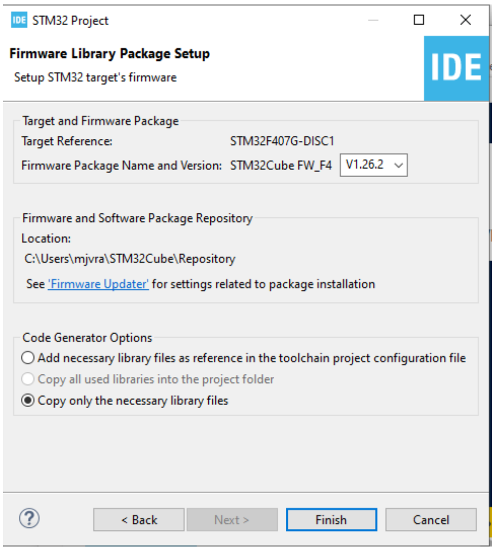
Options

Targeted Language
● C ○ C++

Targeted Binary Type
● Executable ○ Static Library

Targeted Project Type
● STM32Cube ○ Empty

## Step5: Selct Code generation option and click on Finish



STM32 Project

**Firmware Library Package Setup**

Setup STM32 target's firmware

Target and Firmware Package

Target Reference: STM32F407G-DISC1

Firmware Package Name and Version: STM32Cube FW_F4 V1.26.2

Firmware and Software Package Repository

Location:

C:\Users\mjvra\STM32Cube\Repository

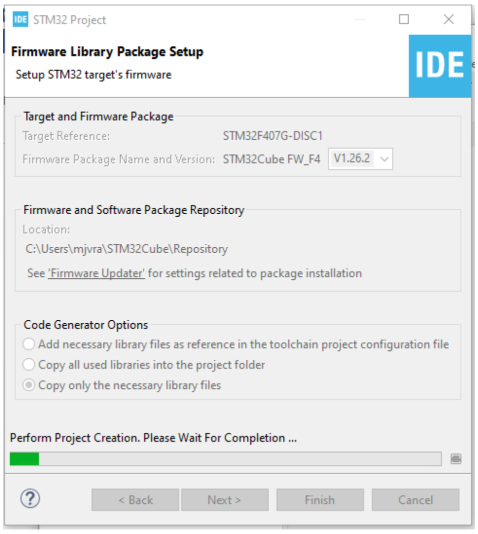
See 'Firmware Updater' for settings related to package installation

Code Generator Options

○ Add necessary library files as reference in the toolchain project configuration file

○ Copy all used libraries into the project folder

● Copy only the necessary library files



STM32 Project

**Firmware Library Package Setup**

Setup STM32 target's firmware

Target and Firmware Package

Target Reference: STM32F407G-DISC1

Firmware Package Name and Version: STM32Cube FW_F4 V1.26.2

Firmware and Software Package Repository

Location:

C:\Users\mjvra\STM32Cube\Repository

See 'Firmware Updater' for settings related to package installation

Code Generator Options

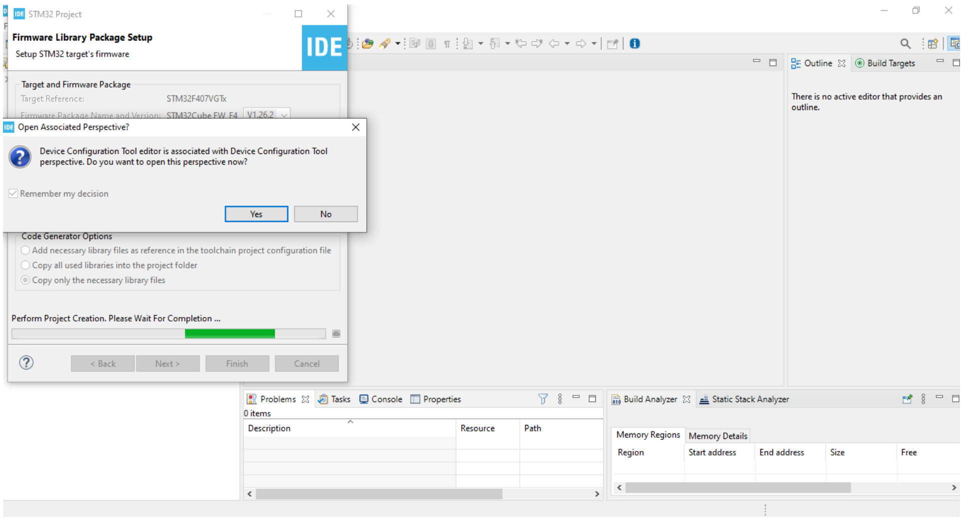
○ Add necessary library files as reference in the toolchain project configuration file

○ Copy all used libraries into the project folder

● Copy only the necessary library files

Perform Project Creation. Please Wait For Completion ...

## Step6: Click "Yes" for Device Configuration



STM32 Project

**Firmware Library Package Setup**

Setup STM32 target's firmware

Target and Firmware Package

Target Reference: STM32F407VGTx

Firmware Package Name and Version: STM32Cube FW_F4 V1.26.2

Open Associated Perspective?

Device Configuration Tool editor is associated with Device Configuration Tool perspective. Do you want to open this perspective now?

☑ Remember my decision

Code Generator Options

○ Add necessary library files as reference in the toolchain project configuration file

○ Copy all used libraries into the project folder

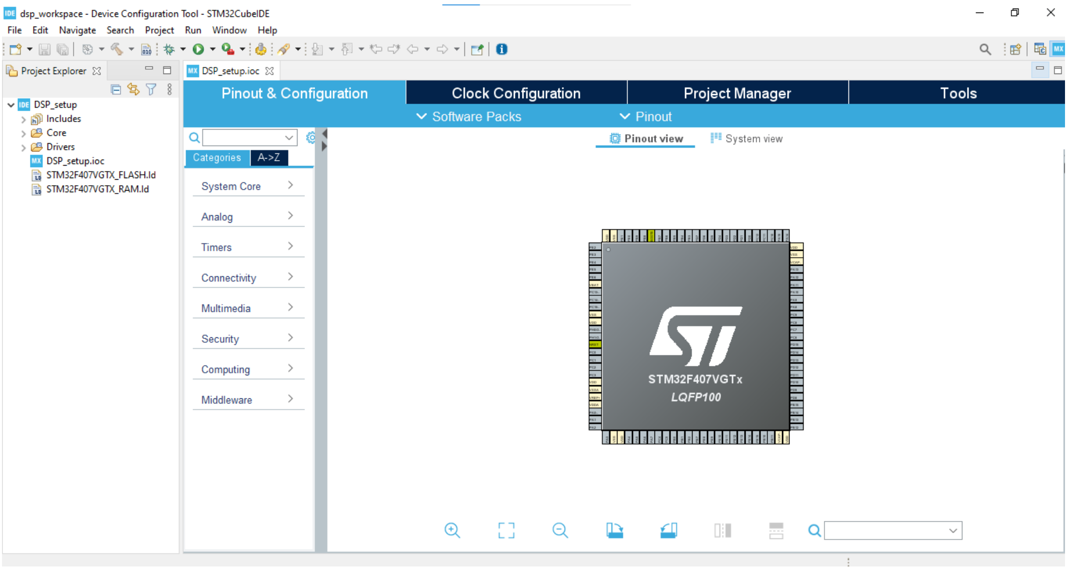
● Copy only the necessary library files

Perform Project Creation. Please Wait For Completion ...

There is no active editor that provides an outline.

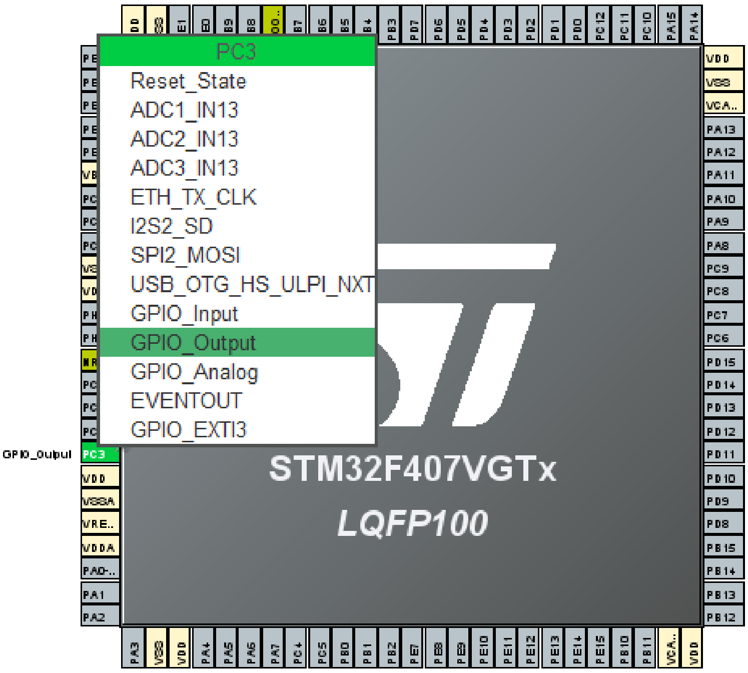
0 items

## YOU WILL GET THIS SCREEN

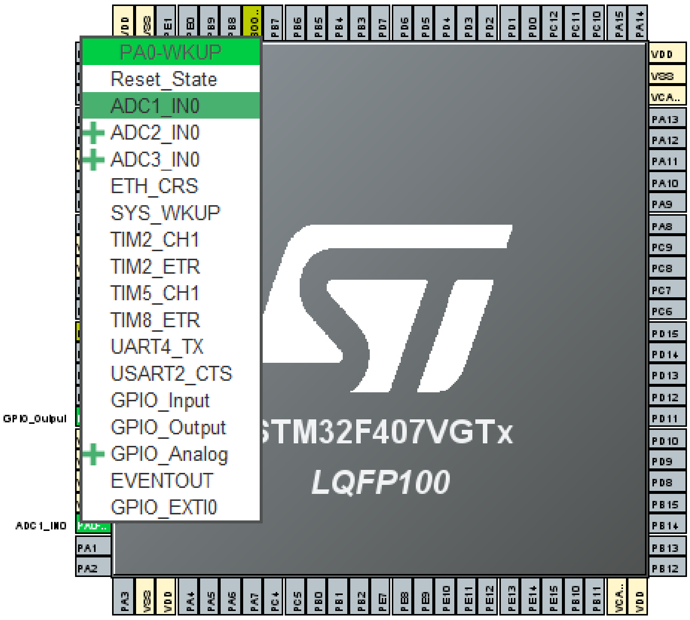


dsp_workspace - Device Configuration Tool - STM32CubeIDE

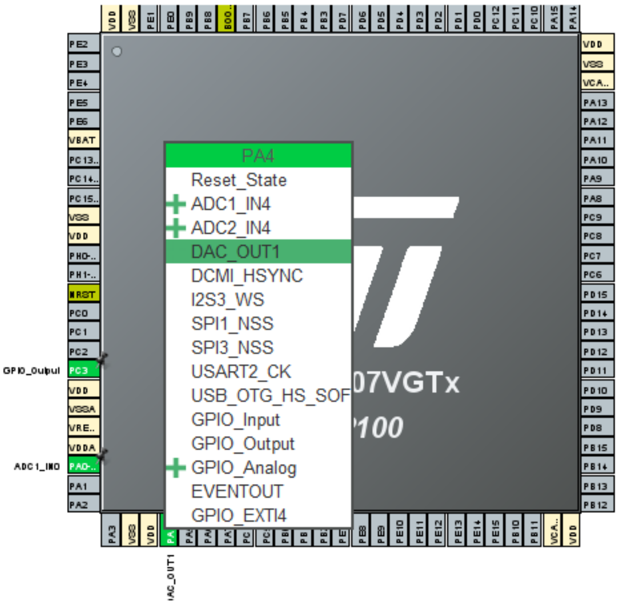
**Step6: Click “PC3” to select it as GPIO_OUTPUT**



**Step7: Click “PA0” to select it as ADC_IN0**

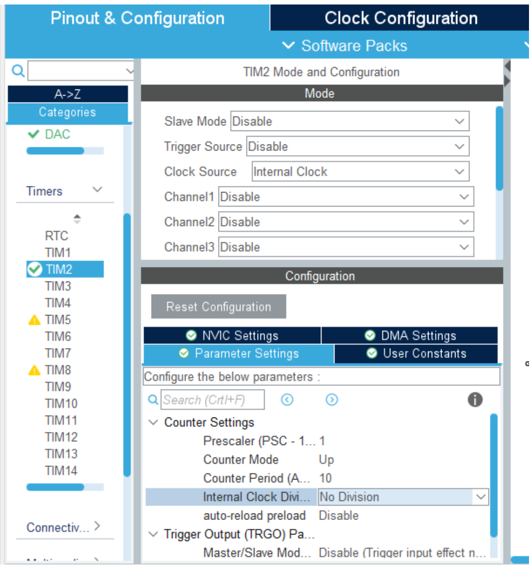


**Step8: Click “PA4” to select it as DAC_OUT1**



**Step9: Click "Pinout and Configuration" and Select TIM2 with the following settings:**

Clock Source: Internal Clock, Prescale=1, Counter Mode=Up, CounterPeriod=10. **This can be adjusted according to the sampling frequency needed.**



Pinout & Configuration | Clock Configuration | Software Packs

TIM2 Mode and Configuration

Mode

Slave Mode: Disable
Trigger Source: Disable
Clock Source: Internal Clock
Channel1: Disable
Channel2: Disable
Channel3: Disable

Configuration

Reset Configuration

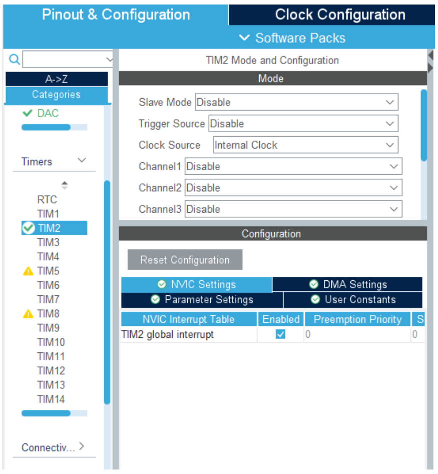
NVIC Settings | DMA Settings | Parameter Settings | User Constants

Configure the below parameters :

Counter Settings
- Prescaler (PSC - 1... 1
- Counter Mode: Up
- Counter Period (A... 10
- Internal Clock Divi... No Division
- auto-reload preload: Disable

Trigger Output (TRGO) Pa...
- Master/Slave Mod... Disable (Trigger input effect n...

**Step10: Enable Timer Interrupt by selecting NVIC Setting under TIM2**



Pinout & Configuration | Clock Configuration | Software Packs

TIM2 Mode and Configuration

Mode

Slave Mode: Disable
Trigger Source: Disable
Clock Source: Internal Clock
Channel1: Disable
Channel2: Disable
Channel3: Disable

Configuration

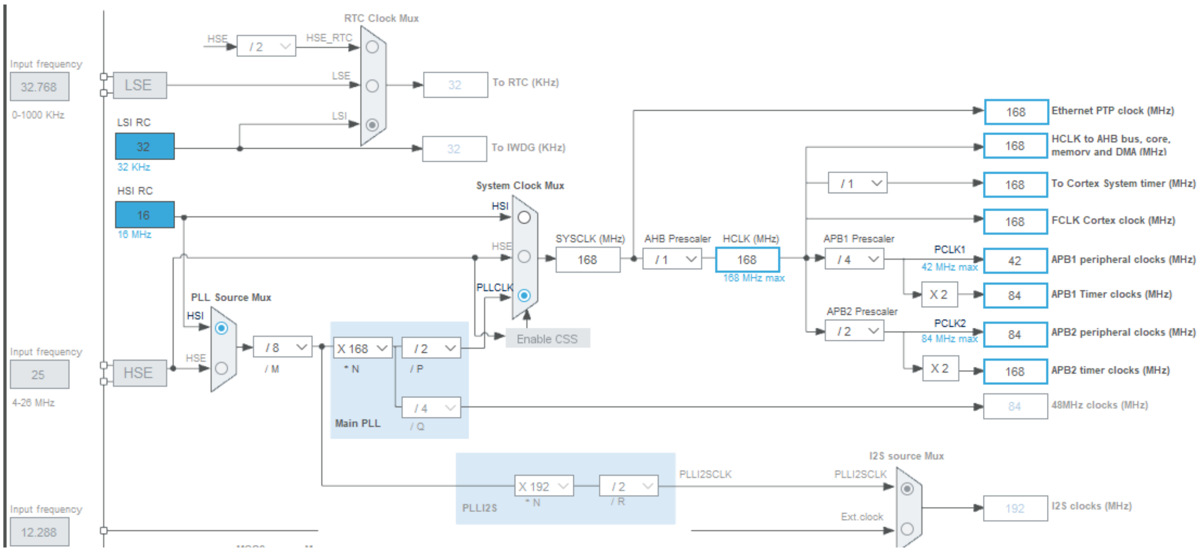
Reset Configuration

NVIC Settings | DMA Settings | Parameter Settings | User Constants

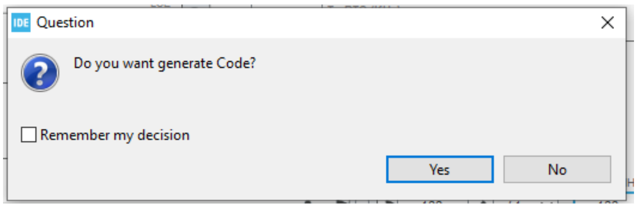
| NVIC Interrupt Table | Enabled | Preemption Priority | S |
|---|---|---|---|
| TIM2 global interrupt | ☑ | 0 | 0 |

**Step10: Click on "Clock Configuration" and select PLL and**

**get APB1 Timer Clock value**



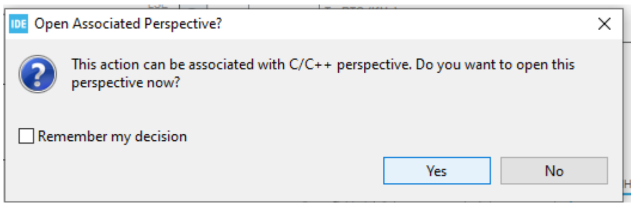
**Step11: Click Save All and it will pop up to generate code and open C/C++perspective. Click Yes for both pop-ups**



Question

Do you want generate Code?

☐ Remember my decision



Open Associated Perspective?

This action can be associated with C/C++ perspective. Do you want to open this perspective now?

☐ Remember my decision

**Step12: Add in main() code for starting DAC, ADC and Timer. (File: main.c)**

```
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_ADC1_Init();
  MX_TIM2_Init();
  MX_DAC_Init();
  /* USER CODE BEGIN 2 */
  HAL_DAC_Start(&hdac, DAC_CHANNEL_1);
  HAL_ADC_Start(&hadc1);
  HAL_TIM_Base_Start_IT(&htim2);
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
```

**Step13: Check if initialisation of ADC, DAC and TIM handles as global. (File: main.c)**

```
/* USER CODE END PM */

/* Private variables ---------------------------------------------------------*/
ADC_HandleTypeDef hadc1;
```

```
DAC_HandleTypeDef hdac;
TIM_HandleTypeDef htim2;
/* USER CODE BEGIN PV */
```

**Step14: Add the code in Timer Interrupt handler (File: stm32f4xx_it.c)**

```
void TIM2_IRQHandler(void)
{
  /* USER CODE BEGIN TIM2_IRQn 0 */

  /* USER CODE END TIM2_IRQn 0 */
  HAL_TIM_IRQHandler(&htim2);

  /* USER CODE BEGIN TIM2_IRQn 1 */
  HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_3);

  // Get the input sample as a digital value
  int xn_d = HAL_ADC_GetValue(&hadc1);

  // Output the same to DAC
  int yn_d = xn_d;
  HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R, yn_d);
  HAL_ADC_Start(&hadc1);   //Start a new conversion

  /* USER CODE END TIM2_IRQn 1 */
}
```

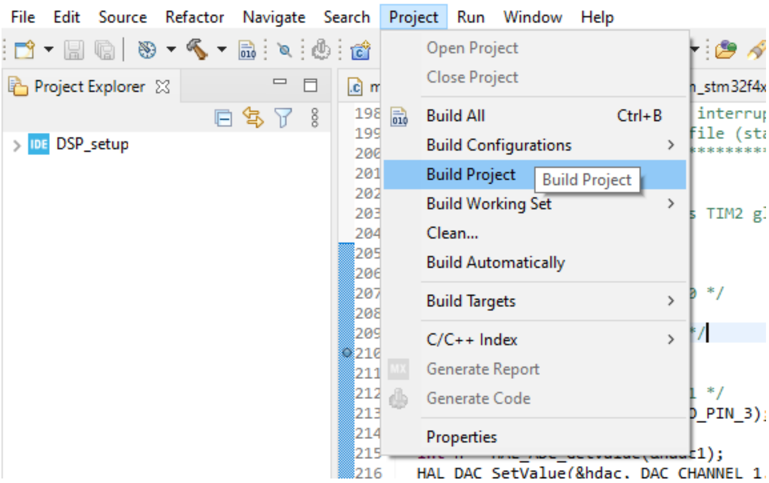
Step15: Add declaration of hadc1 and hdac in addition to generated htim2 already generated. (File: stm32f4xx_it.c)

```
/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
extern ADC_HandleTypeDef hadc1;
extern DAC_HandleTypeDef hdac;
/* USER CODE END 0 */

/* External variables --------------------------------------------------------*/
extern TIM_HandleTypeDef htim2;
/* USER CODE BEGIN EV */

/* USER CODE END EV */
```

**Sep16: Click Build Project to build the project successfully to create binary files.**



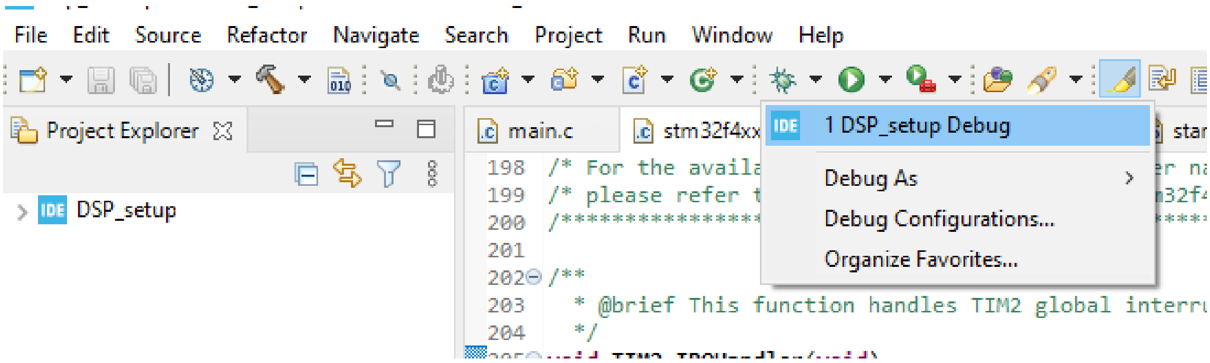
…………………………..

```
arm-none-eabi-gcc "../Core/Src/syscalls.c" -mcpu=cortex-m4 -std=gnu11 -g3 -DDEBUG
-DUSE_HAL_DRIVER -DSTM32F407xx -c -I../Core/Inc -
I../Drivers/STM32F4xx_HAL_Driver/Inc -I../Drivers/STM32F4xx_HAL_Driver/Inc/Legacy
-I../Drivers/CMSIS/Device/ST/STM32F4xx/Include -I../Drivers/CMSIS/Include -O0 -
ffunction-sections -fdata-sections -Wall -fstack-usage -MMD -MP -
MF"Core/Src/syscalls.d" -MT"Core/Src/syscalls.o" --specs=nano.specs -mfpu=fpv4-sp-
d16 -mfloat-abi=hard -mthumb -o "Core/Src/syscalls.o"
arm-none-eabi-gcc "../Core/Src/sysmem.c" -mcpu=cortex-m4 -std=gnu11 -g3 -DDEBUG -
DUSE_HAL_DRIVER -DSTM32F407xx -c -I../Core/Inc -
I../Drivers/STM32F4xx_HAL_Driver/Inc -I../Drivers/STM32F4xx_HAL_Driver/Inc/Legacy
-I../Drivers/CMSIS/Device/ST/STM32F4xx/Include -I../Drivers/CMSIS/Include -O0 -
ffunction-sections -fdata-sections -Wall -fstack-usage -MMD -MP -
MF"Core/Src/sysmem.d" -MT"Core/Src/sysmem.o" --specs=nano.specs -mfpu=fpv4-sp-d16
-mfloat-abi=hard -mthumb -o "Core/Src/sysmem.o"
arm-none-eabi-gcc "../Core/Src/system_stm32f4xx.c" -mcpu=cortex-m4 -std=gnu11 -g3
-DDEBUG -DUSE_HAL_DRIVER -DSTM32F407xx -c -I../Core/Inc -
I../Drivers/STM32F4xx_HAL_Driver/Inc -I../Drivers/STM32F4xx_HAL_Driver/Inc/Legacy
-I../Drivers/CMSIS/Device/ST/STM32F4xx/Include -I../Drivers/CMSIS/Include -O0 -
ffunction-sections -fdata-sections -Wall -fstack-usage -MMD -MP -
MF"Core/Src/system_stm32f4xx.d" -MT"Core/Src/system_stm32f4xx.o" --
specs=nano.specs -mfpu=fpv4-sp-d16 -mfloat-abi=hard -mthumb -o
"Core/Src/system_stm32f4xx.o"
arm-none-eabi-gcc -o "DSP_setup.elf" @"objects.list"   -mcpu=cortex-m4 -
T"C:\PES\DSP\DSPLab\dsp_workspace\DSP_setup\STM32F407VGTX_FLASH.ld" --
specs=nosys.specs -Wl,-Map="DSP_setup.map" -Wl,--gc-sections -static --
specs=nano.specs -mfpu=fpv4-sp-d16 -mfloat-abi=hard -mthumb -Wl,--start-group -lc
-lm -Wl,--end-group
Finished building target: DSP_setup.elf

arm-none-eabi-size   DSP_setup.elf
arm-none-eabi-objdump -h -S  DSP_setup.elf  > "DSP_setup.list"
arm-none-eabi-objcopy  -O binary  DSP_setup.elf  "DSP_setup.bin"
   text	   data	    bss	    dec	    hex	filename
  11128	     20	   1732	  12880	   3250	DSP_setup.elf
Finished building: default.size.stdout
Finished building: DSP_setup.bin
Finished building: DSP_setup.list

23:51:20 Build Finished. 0 errors, 0 warnings. (took 5s.800ms)
```

**Sep17: You can Debug and Run on the board now.**



```
/* For the availa
/* please refer t
/***************

/**
  * @brief This function handles TIM2 global interru
  */
```

- Check if Timerhandler is getting executed repeatedly by inserting breakpoint on first line
- Observe PC3 ( GPIO_Output ) if the toggling is happening properly and also check the frequency which must give sampling frequency. This needs oscilloscope