DigiToad Technologies

ADC using interrupt in CUBE IDE (Audio Sensor)

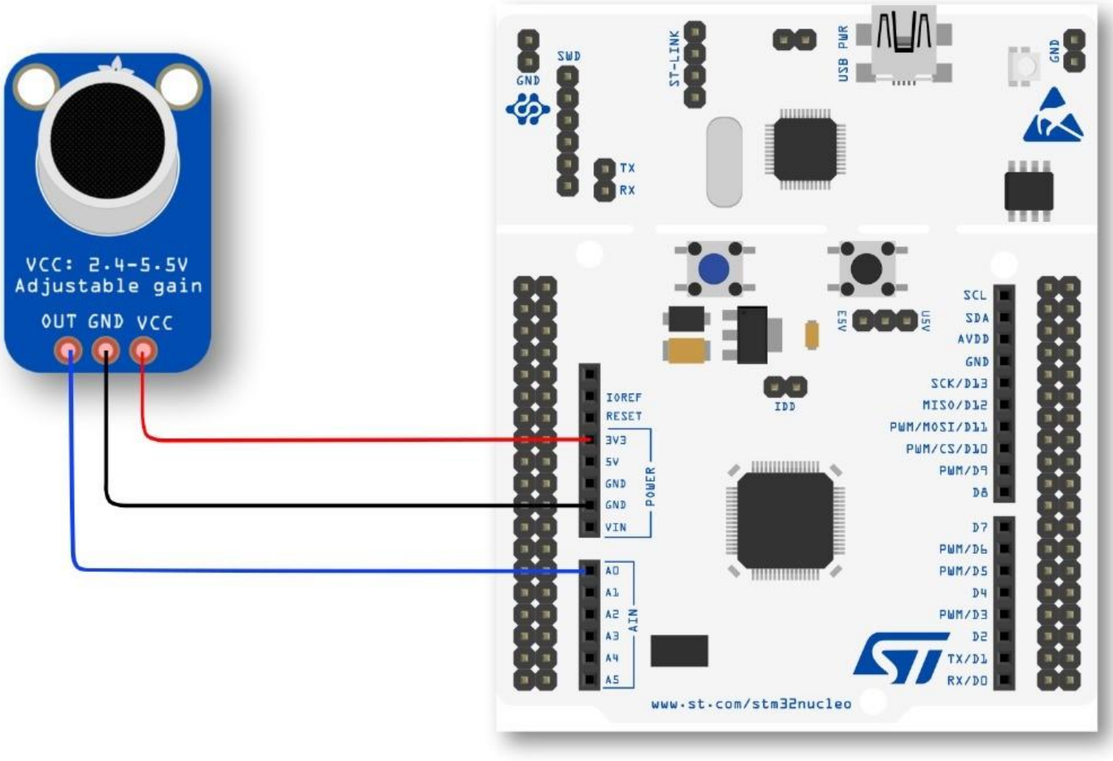
Objective:

The objective of this experiment is to interface an audio sensor to an STM32 microcontroller and configure the ADC in interrupt mode. Every time the ADC completes a conversion, the ADC will generate an internal interrupt. Once the interrupt occurs, the converted value will be collected from the ADC data registers and stored into a variable.

Requirements:

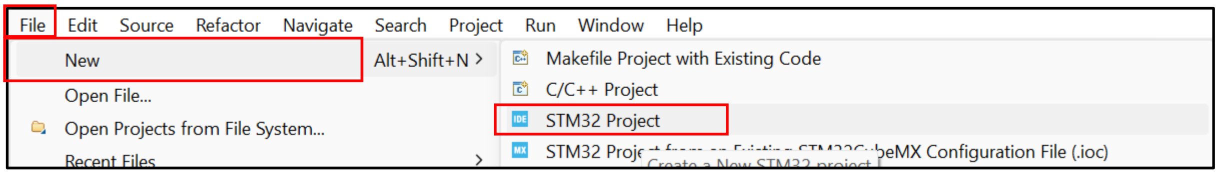
1. STM32 Cube IDE software.
2. Audio Sensor (Analog).
3. STM32 Microcontroller.
4. USB Cable for the microcontroller.
5. Jumper Wires.
6. PC or Laptop.

Connection Diagram:

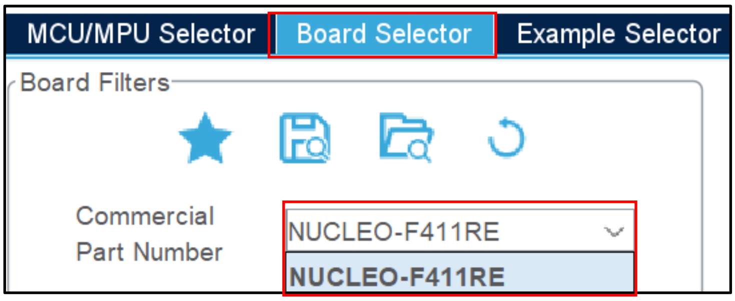


Procedure:

1. Click on **File→New→STM32 Project**to start your project on Cube IDE.



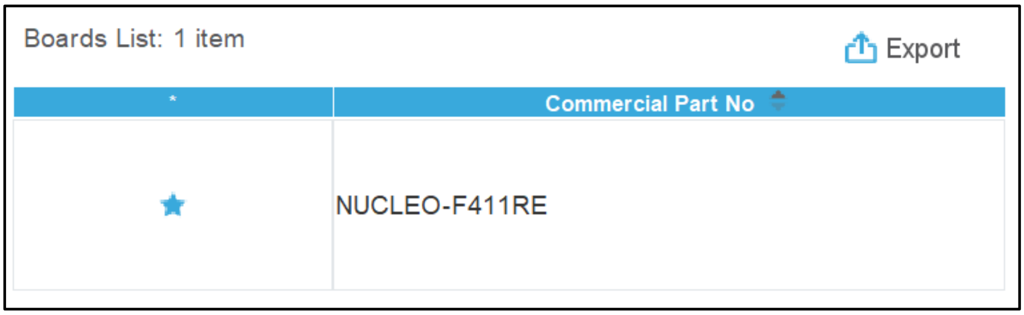
2. A **Target Selection** window will open. Click on **Board Selector**, where you need to select the microcontroller board you are working with.



Commercial Part Number: NUCLEO-F411RE

NUCLEO-F411RE

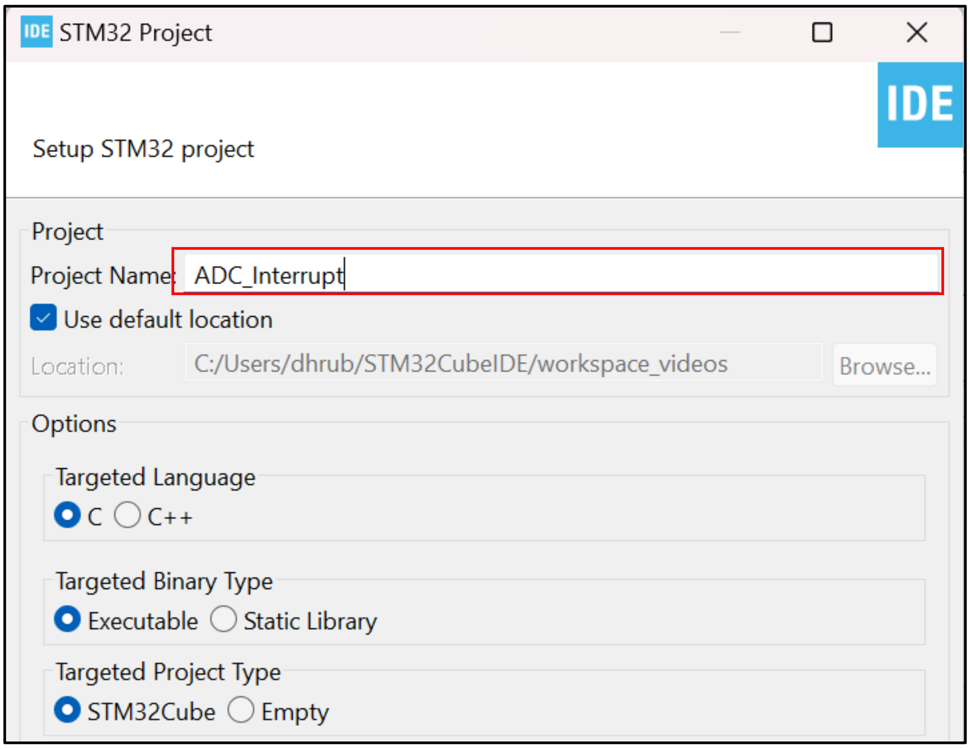
3. After this on the right-hand side of the window, under **Board List** you will see the board you have selected. Click on the board and then click on **Next.**



Boards List: 1 item

*	Commercial Part No
★	NUCLEO-F411RE

4. In the next window you need to give your project a name, rest of the things will remain by default as it is for now. Click on **Next.**



STM32 Project

Setup STM32 project

Project

Project Name: ADC_Interrupt

☑ Use default location

Location: C:/Users/dhrub/STM32CubeIDE/workspace_videos

Options

Targeted Language

● C ○ C++

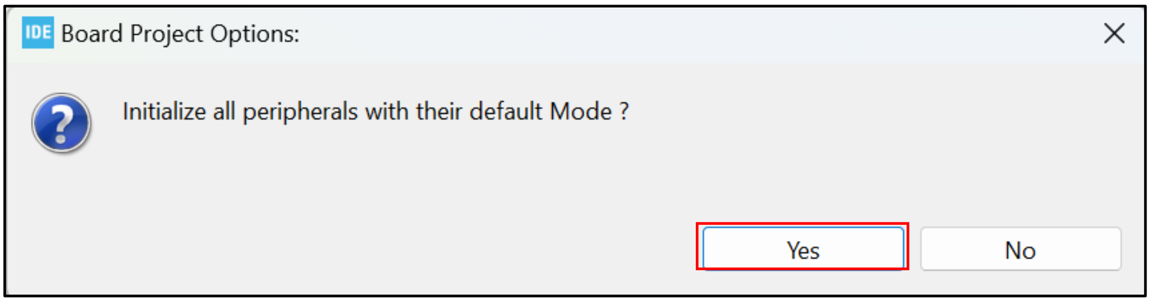
Targeted Binary Type

● Executable ○ Static Library

Targeted Project Type

● STM32Cube ○ Empty

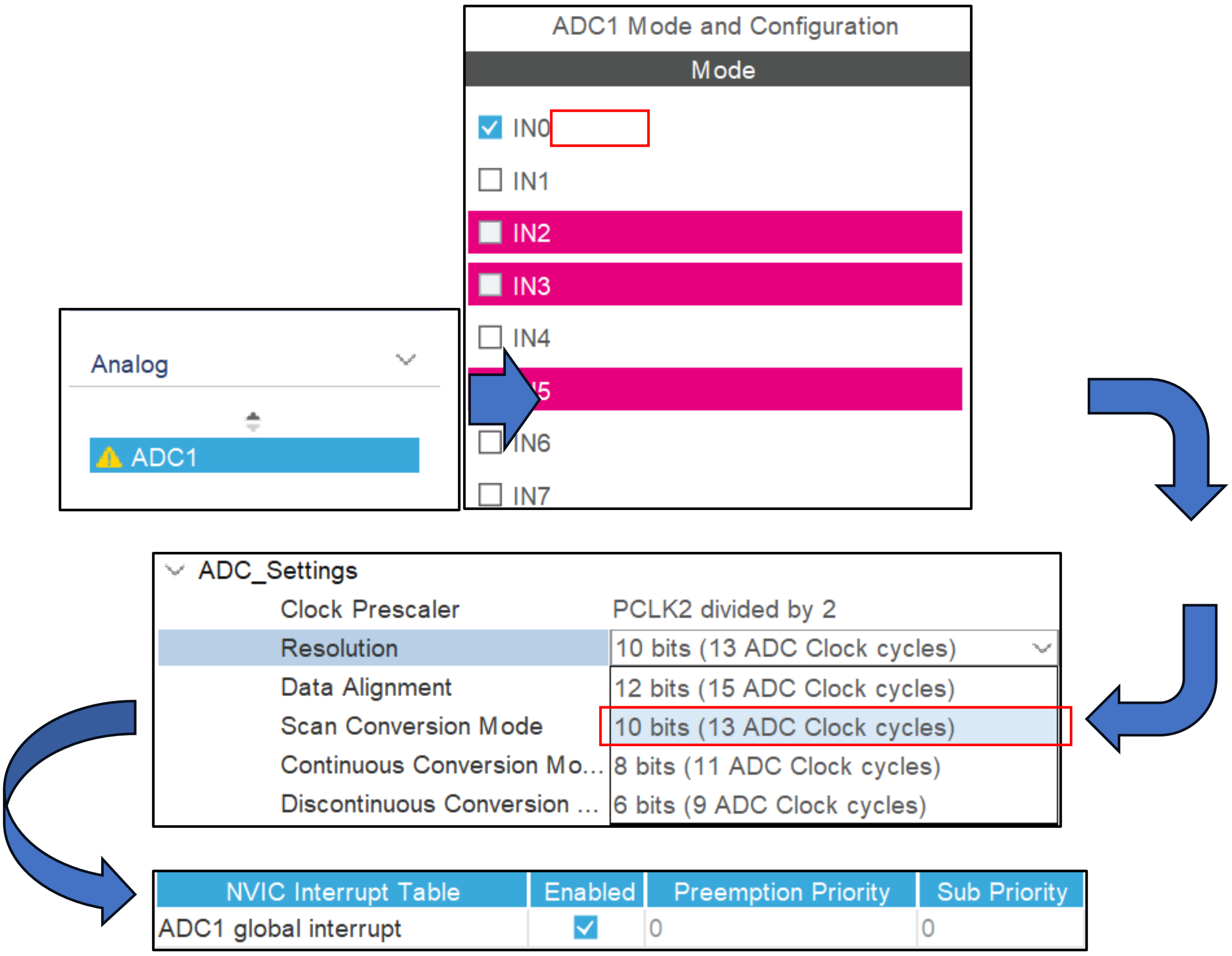
5. Cube IDE will ask if you want to initialize all peripherals with their default mode, click on **Yes.**



Board Project Options:

Initialize all peripherals with their default Mode ?

6. Next on the left-hand side**Categories →Analog**select **ADC1**then select **IN0**under **Mode.** Under **Configuration** select **NVIC** and enable the **ADC global interrupt.**Otherwise, you can also go to **System Core →NVIC** and enable the same.



ADC1 Mode and Configuration

Mode

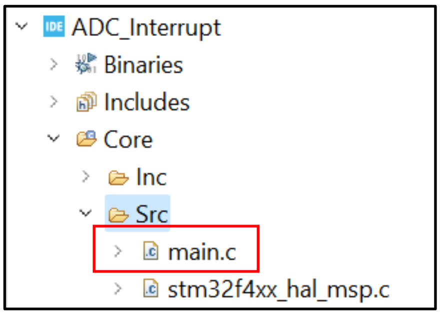
☑ IN0
□ IN1
□ IN2
□ IN3
□ IN4
IN5
□ IN6
□ IN7

ADC_Settings

Clock Prescaler	PCLK2 divided by 2
Resolution	10 bits (13 ADC Clock cycles)
Data Alignment	12 bits (15 ADC Clock cycles)
Scan Conversion Mode	10 bits (13 ADC Clock cycles)
Continuous Conversion Mo...	8 bits (11 ADC Clock cycles)
Discontinuous Conversion ...	6 bits (9 ADC Clock cycles)

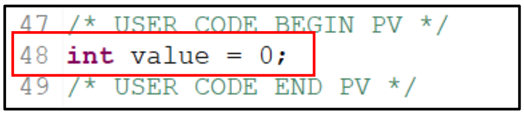
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
ADC1 global interrupt	☑	0	0

7. Press C**trl+S**to generate your code. On the left-hand side of the Cube IDE, under **Project Explorer** go to the project you have created (For example I have named my project as (ADC_Interrupt)**ADC_Interrupt→Core →Src→main.c (**double click to load the code).

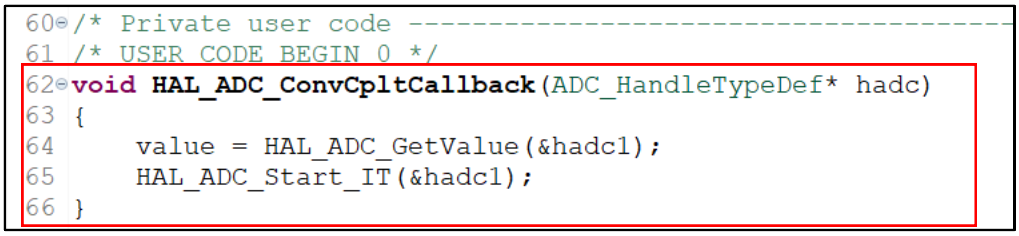


```
ADC_Interrupt
  Binaries
  Includes
  Core
    Inc
    Src
      main.c
      stm32f4xx_hal_msp.c
```

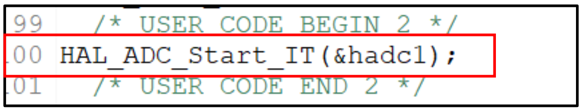
8. Cube IDE automatically generates theinitialization codebased on the configurations you have done. Cube IDE used HAL libraries. Below is the code snippets, please put your code in the appropriate places in the **main.c** file.



```
/* USER CODE BEGIN PV */
int value = 0;
/* USER CODE END PV */
```

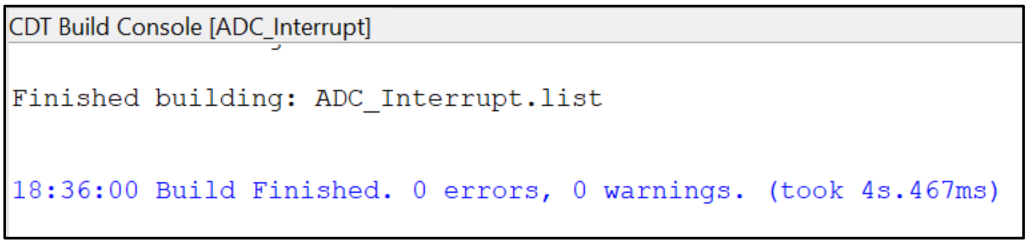


```
60⊖/* Private user code ---------------------------------------
61 /* USER CODE BEGIN 0 */
62⊖void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
63 {
64     value = HAL_ADC_GetValue(&hadc1);
65     HAL_ADC_Start_IT(&hadc1);
66 }
```



```
99    /* USER CODE BEGIN 2 */
00 HAL_ADC_Start_IT(&hadc1);
01    /* USER CODE END 2 */
```

9. Now click on the build symbol on the top left corner on your Cube IDE. If you have done everything correctly your code should be built without any errors.

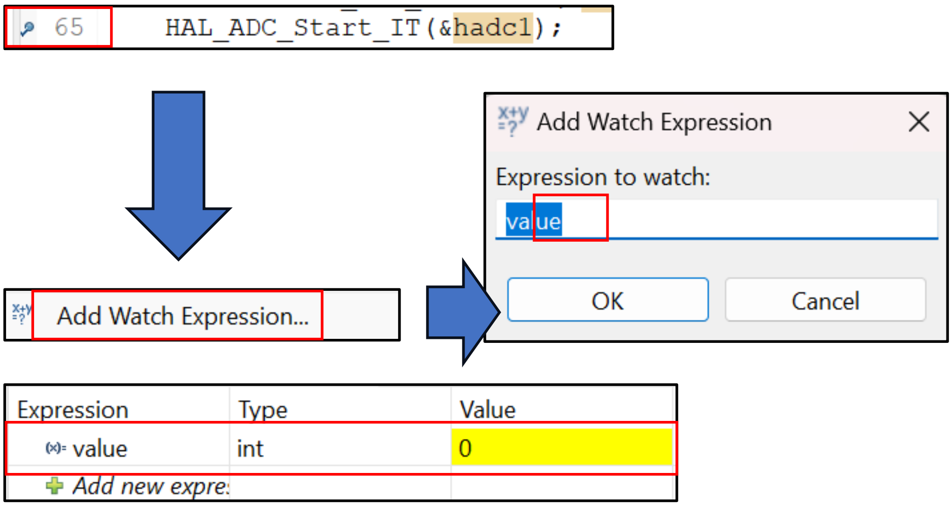


```
CDT Build Console [ADC_Interrupt]

Finished building: ADC_Interrupt.list

18:36:00 Build Finished. 0 errors, 0 warnings. (took 4s.467ms)
```

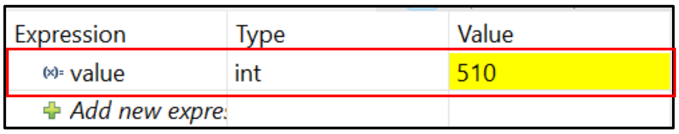
10. Next connect your STM32 board with your audio sensor connect to it to your PC and click on the **Debug** icon to start the Debugging process. An**Edit Configuration** window will open, click on **OK,** without making any changes.
11. In debugging mode, double click on line number 65 to add a breakpoint into the code. Next, select the **value**in line 64 and right click on it. Select **Add Watch Expression.**In the new pop-up window select **Ok**.On the right-hand side,a window will open where the **value**variable can be watched as it changes.



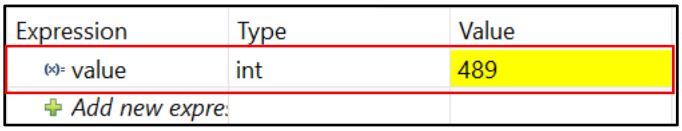
```
65     HAL_ADC_Start_IT(&hadc1);
```

Expression	Type	Value
value	int	0
Add new expre:		

12. Click on the **Resume** icon to run your code. You should be able to see the value of**value** changing every time you click on **Resume.**



Expression	Type	Value
(x)= value	int	510
✚ *Add new expre:*		



Expression	Type	Value
(x)= value	int	489
✚ *Add new expre:*		

13. To move out of the debugging mode, click on the **Terminate** icon . You will be moved out of the debugging mode.

Note: All important steps and parts are highlighted with a red colour box for the proper understanding of the user. This document is for the use of education purpose only.