Interfacing Ultrasonic sensors (US 100) to implement a data logger to send ultrasonic sensor data from STM32 to NanoEdge AI Studio

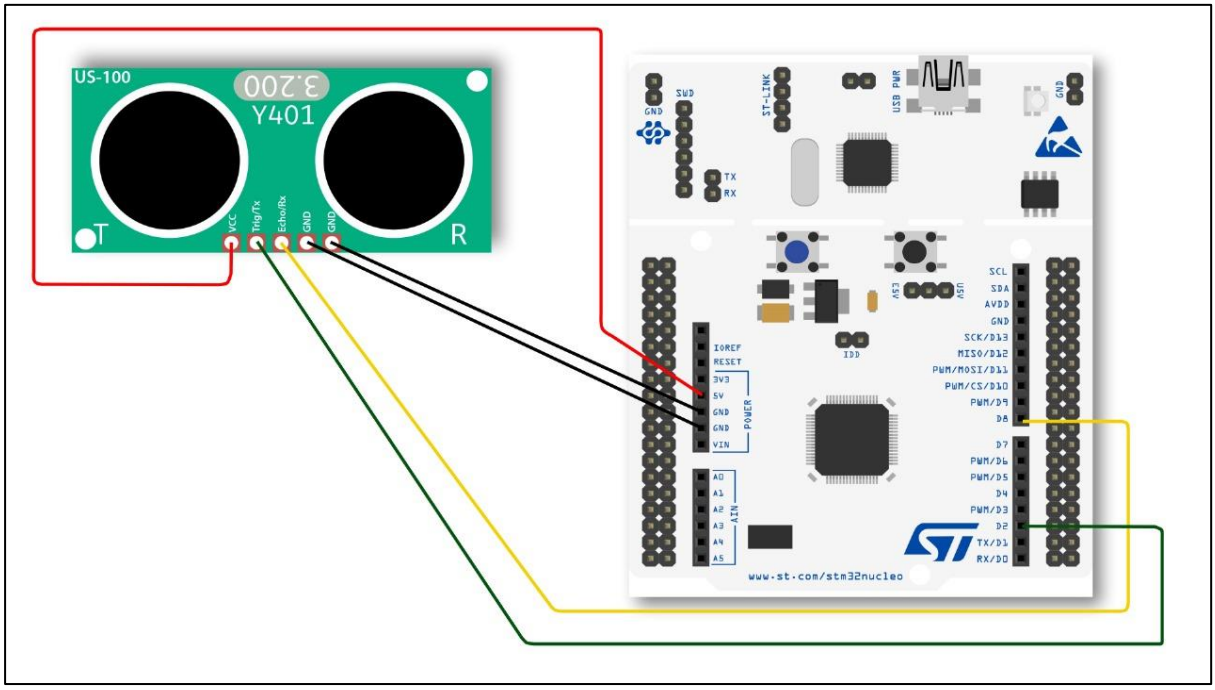
Objective:

The objective of this experiment is to interface an ultrasonic sensor to an STM32 microcontroller and create a datalogger code. This datalogger code will create a buffer where all the ultrasonic sensor data sample will be stored, using which we will be able to create datasets of different distance samples to build a machine learning model in the NanoEdge AI Studio.

Requirements:

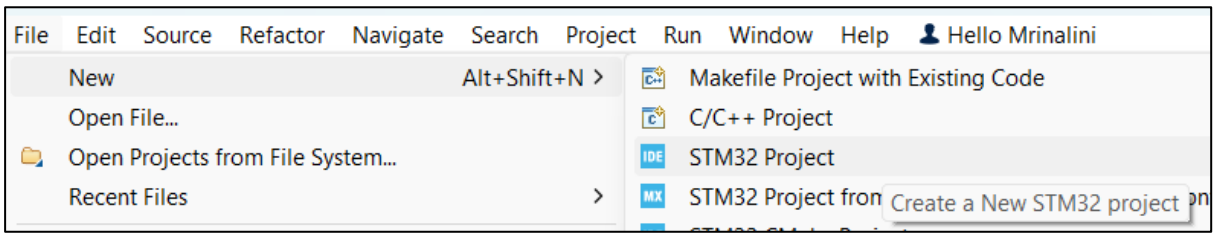
1. STM32 Cube IDE software.
2. Ultrasonic Sensor (US 100).
3. STM32 Microcontroller.
4. USB Cable for the microcontroller.
5. Jumper Wires.

Connection Diagram:

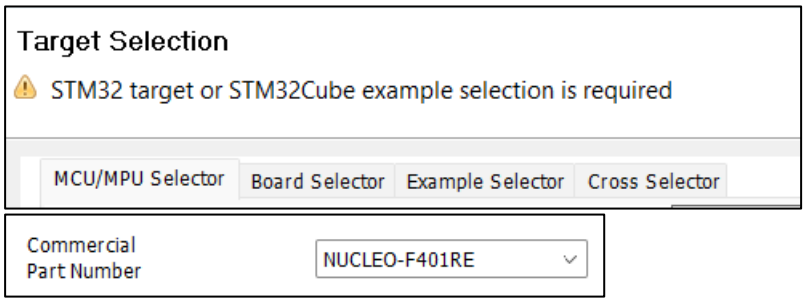


Procedure:

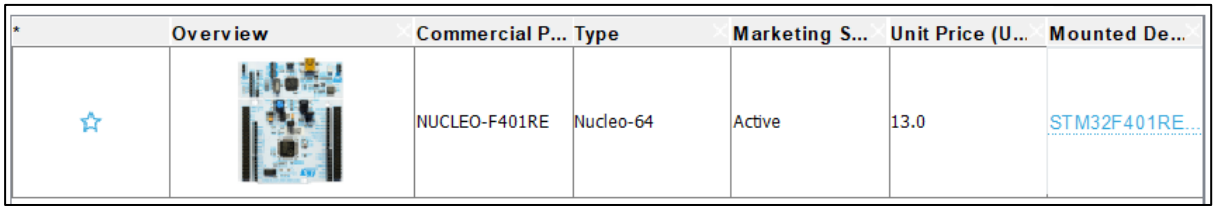
1. Click on **File →New →STM32 Project** to start your project on Cube IDE.



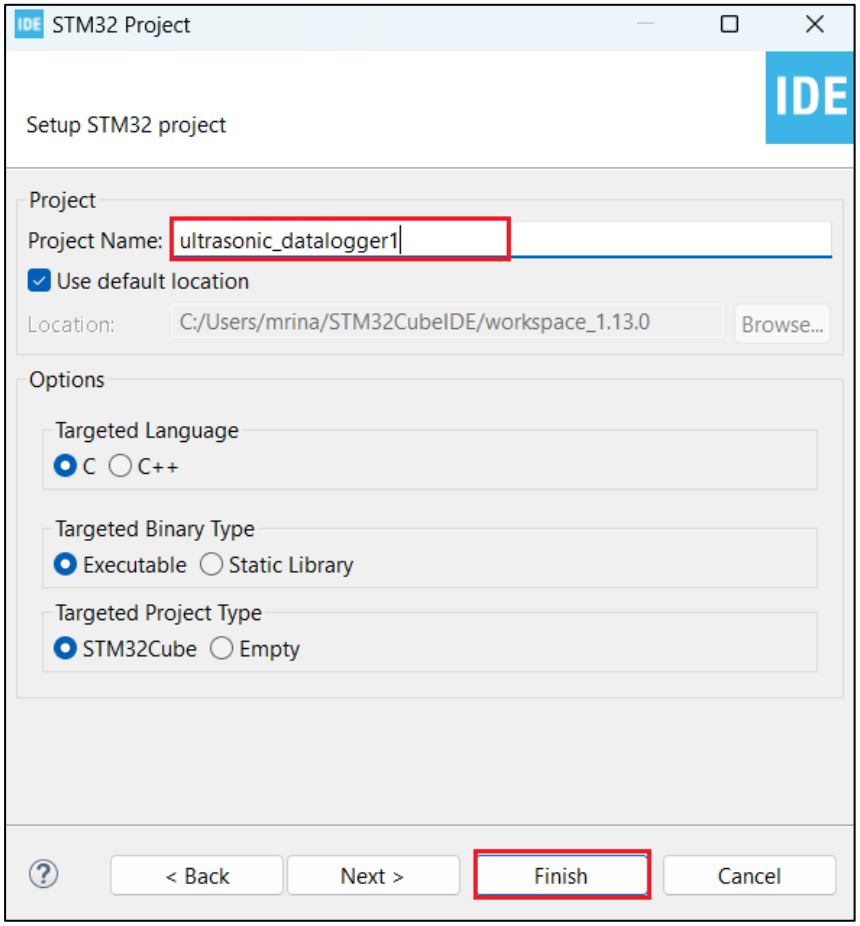
2. A **Target Selection** window will open. Click on **Board Selector**, where you need to select the microcontroller board you are working with (NUCLEO-F401RE/NUCLEO-F411RE).



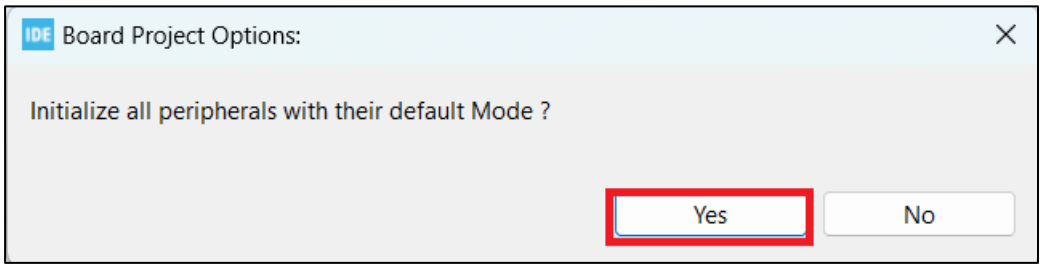
3. After this on the right-hand side of the window, under **Board List** you will see the board you have selected. Click on the board and then click on **Next.**



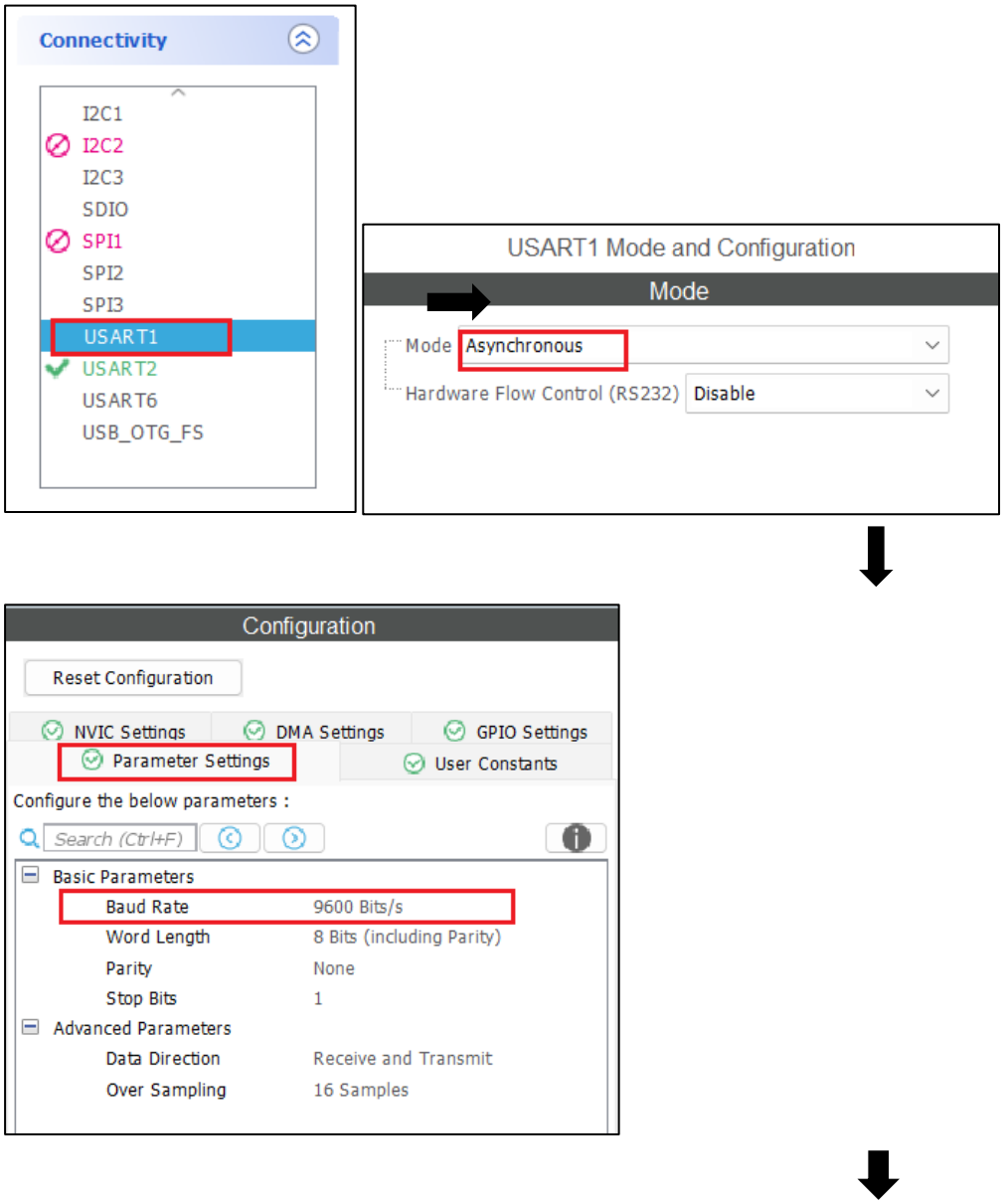
4. Give your project a name, rest of the things will remain by default as it is for now. Click on **FINISH.**

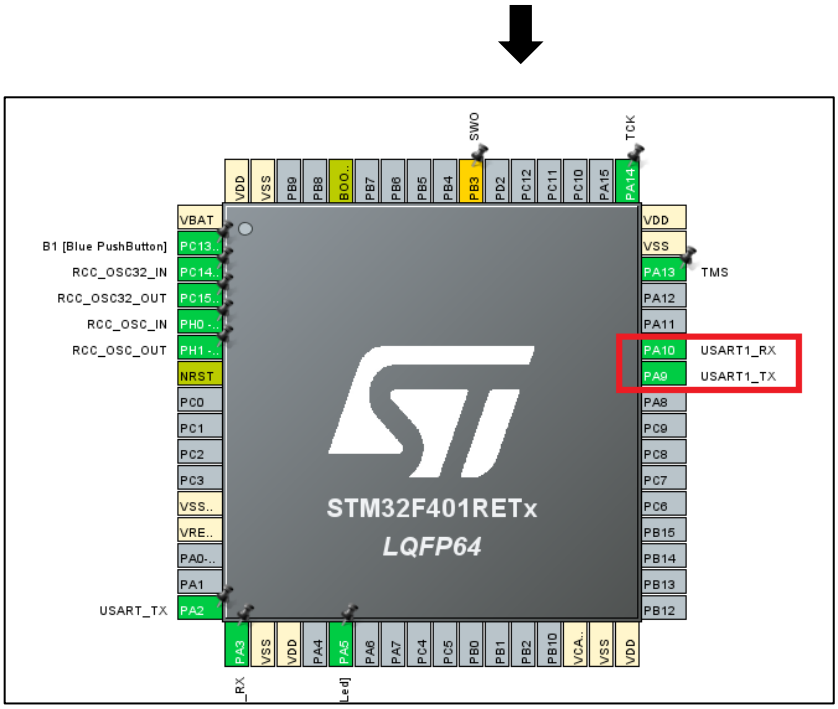


5. Cube IDE will ask if you want to initialize all peripherals with their default mode, click on **Yes.**

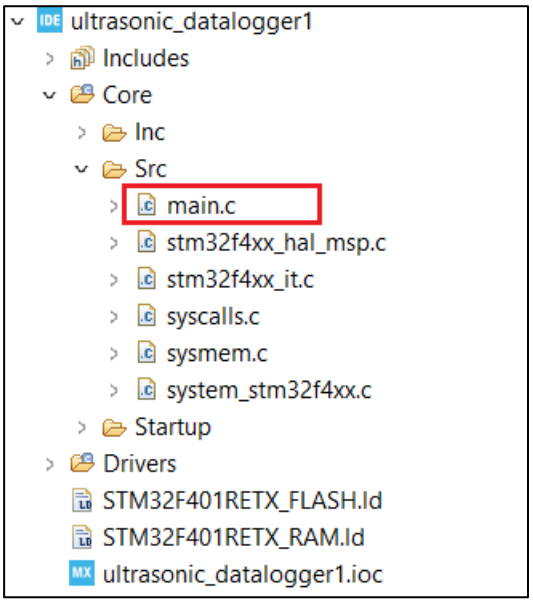


6. In left hand side **Categories→Connectivity**select **USART1**. Then in **USART1 mode & configuration** under mode select Asynchronous. Then go to configuration select **Parameter→settingBasic parameter**, change baud rate to **9600 Bits/s**. In right hand side, pinout view you can see that **PA10**&**PA9** as **USART1_Rx**&**USART1_Tx**respectively ishighlighted.





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 (ultrasonic_datalogger1) **ultrasonic_datalogger1→Core →Src→main.c (**double click to load the code).



8. Cube IDE automatically generates a code format based on the configurations you have done. Cube IDE uses HAL libraries. Below are the code snippets, please put your code in the appropriate places in the **main.c** file.

```
  */
/* USER CODE END Header */
/* Includes ------------------------------------
#include "main.h"

/* Private includes ----------------------------
/* USER CODE BEGIN Includes */
#include "stdio.h"
#include "stdlib.h"
#include "string.h"


/* USER CODE END Includes */

```

```
/* Private macro -------------------------------------
/* USER CODE BEGIN PM */
#define buffer 256
/* USER CODE END PM */

/* Private variables ---------------------------------
UART_HandleTypeDef huart1;
UART_HandleTypeDef huart2;

/* USER CODE BEGIN PV */
uint8_t txData = 0x55;
uint8_t rxData[2];
float distance;
float sonic_distance_buffer[buffer]={0};
/* USER CODE END PV */
```

```
/* Private function prototypes -----------------------
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_USART2_UART_Init(void);
/* USER CODE BEGIN PFP */
void fill_sonic_buffer();
//void distance_measure_buffer();
/* USER CODE END PFP */

```

```
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
      fill_sonic_buffer();
//    distance_measure_buffer();
    /* USER CODE END WHILE */
```

```c
/* USER CODE BEGIN 4 */

            /* Send trigger command */
void fill_sonic_buffer(){
            if(HAL_UART_Transmit(&huart1, &txData, 1, HAL_MAX_DELAY)!= HAL_OK){
                Error_Handler();
                HAL_Delay(50);
            }

          /* Receive and process data if available */
            if (HAL_UART_Receive(&huart1, rxData, 2, HAL_MAX_DELAY) == HAL_OK) {
                uint16_t highByte = rxData[0];
                uint16_t lowByte = rxData[1];
                distance = (highByte << 8) | lowByte;
            }

                if (distance > 1 && distance < 10000) {
                    for(int i=0;i<buffer;i++){
                            sonic_distance_buffer[i] = distance;
                            HAL_Delay(3);
                            printf("%.2f",sonic_distance_buffer[i]);
                            printf(" ");
                        }
                        printf("\r\n");
                }

            }
```

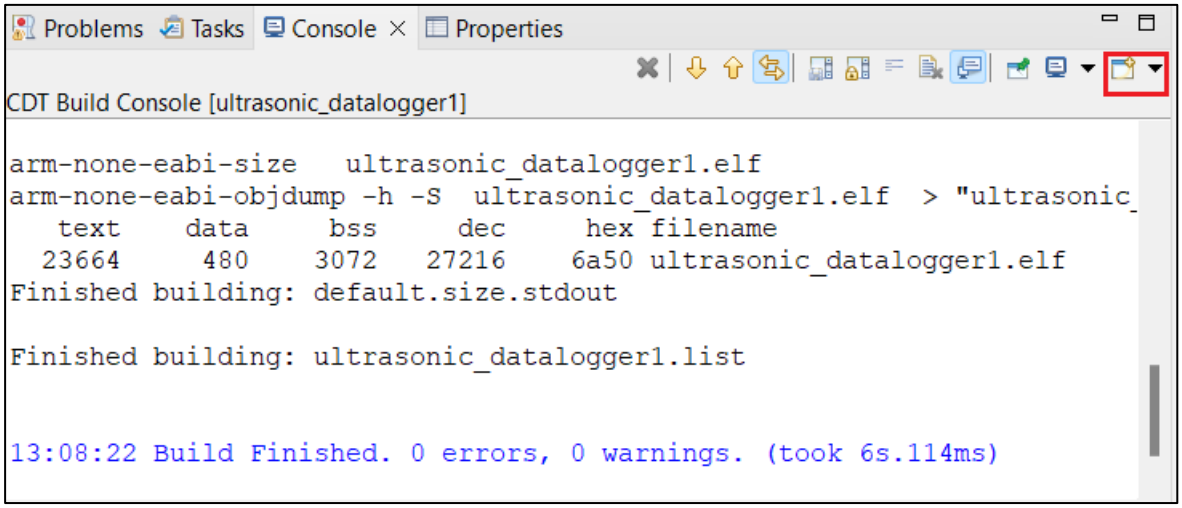
```c
int __io_putchar(int ch){
    HAL_UART_Transmit(&huart2, (uint8_t*)&ch, 1, HAL_MAX_DELAY);
    return ch;
}

/* USER CODE END 4 */
```

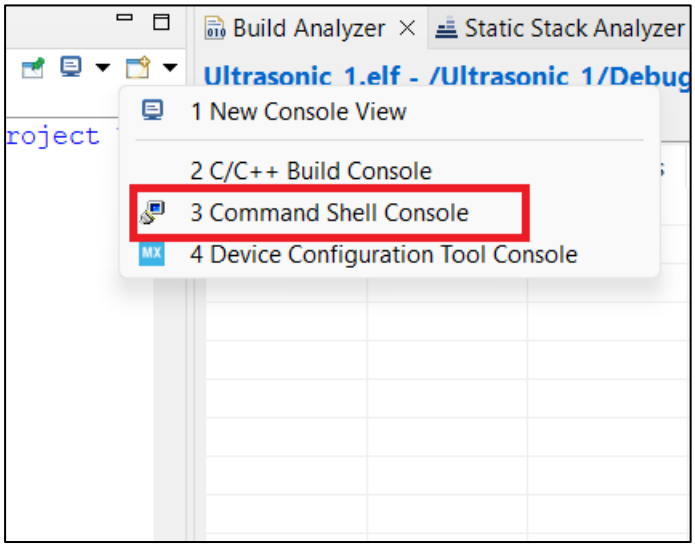
9. Now click on the build  symbol on the top left corner on your Cube IDE. If you have done everything correctly your code should be built without any errors.

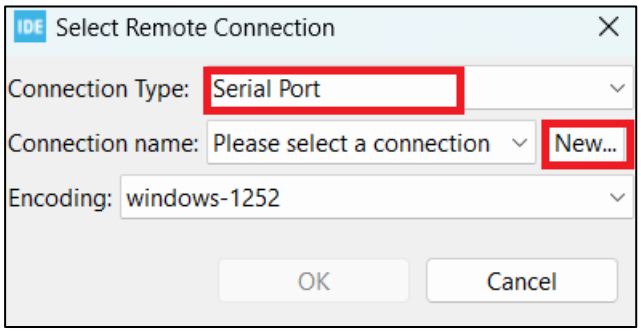
```
Problems  Tasks  Console ×  Properties
CDT Build Console [ultrasonic_datalogger1]

arm-none-eabi-size   ultrasonic_datalogger1.elf
arm-none-eabi-objdump -h -S  ultrasonic_datalogger1.elf  > "ultrasonic_datalogger1.list"
   text    data     bss     dec     hex filename
  23664     480    3072   27216    6a50 ultrasonic_datalogger1.elf
Finished building: default.size.stdout

Finished building: ultrasonic_datalogger1.list


13:08:22 Build Finished. 0 errors, 0 warnings. (took 6s.114ms)
```

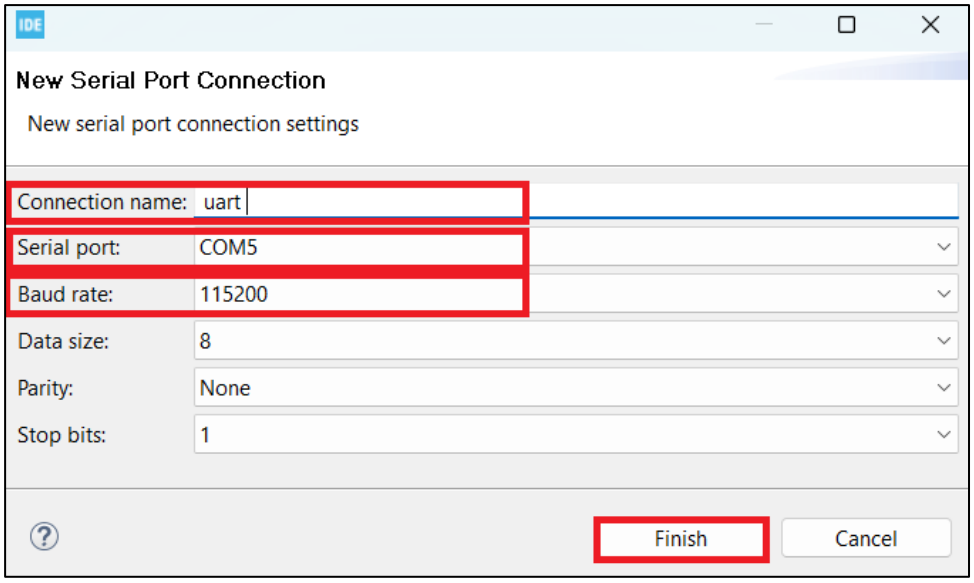
10. Next connect your STM32 board with your audio sensor connect to it to your PC and click on the **Debug**  icon to start the Debugging process. An **Edit Configuration** window will open, click on **OK,** without making any changes.

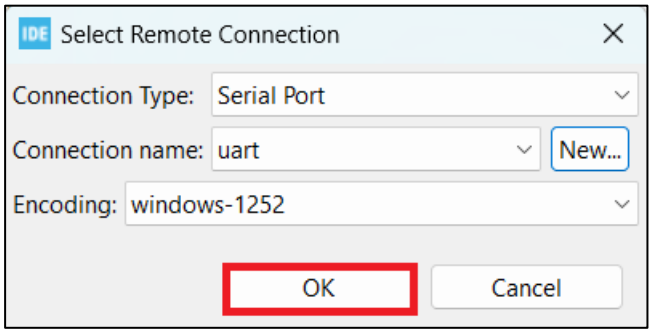
11. In the debug mode, go to the bottom right hand side corner, click on open console. Select the **Connection Type** as **Serial Port**, then click on **New.** In the new window, in **Connection name** give some name to your new connection, select the **Serial port** correctly, and **baud rate** should be **115200**. Then click on **Finish** and then **Ok.** A console with the given name will be opened at the bottom of your screen.

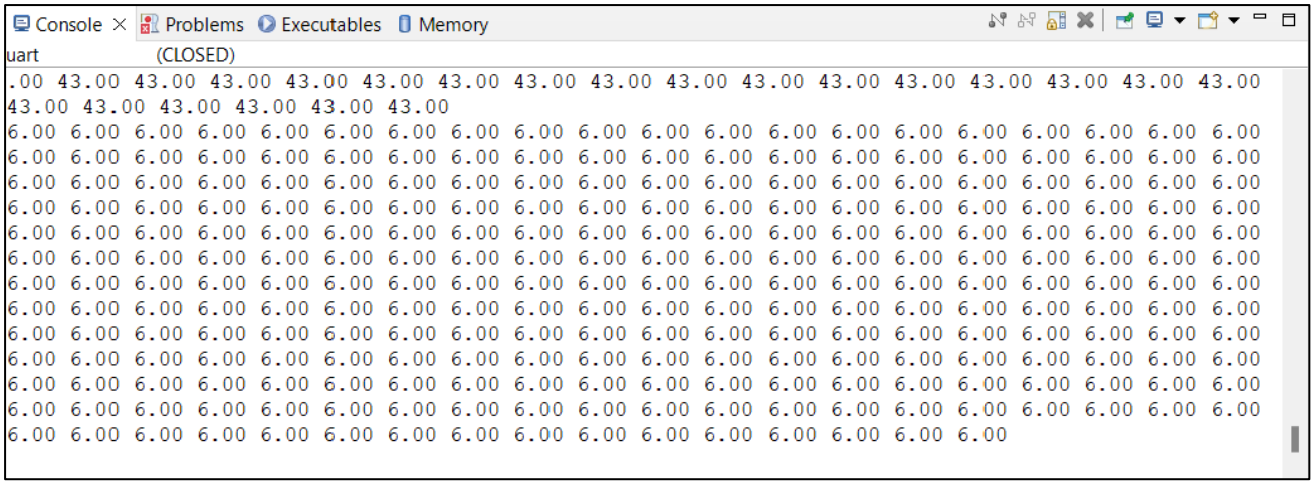





Select Remote Connection
Connection Type:
Serial Port
Connection name:
Please select a connection
New...
Encoding:
windows-1252
OK
Cancel


New Serial Port Connection
New serial port connection settings
Connection name:
uart
Serial port:
COM5
Baud rate:
115200
Data size:
8
Parity:
None
Stop bits:
1
Finish
Cancel


Select Remote Connection
Connection Type:
Serial Port
Connection name:
uart
New...
Encoding:
windows-1252
OK
Cancel



12. Click on the **Resume** icon to run your code. You should be able to see the value ofultrasonic sensor in the form of buffer containing 256 samples.

uart (CLOSED)

```
.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00 43.00
43.00 43.00 43.00 43.00 43.00 43.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00 6.00
```

13. Before moving out of the debugging mode, click on Disconnect and close the console then 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.