Interfacing Ultrasonic sensors (US 100) to STM32 board

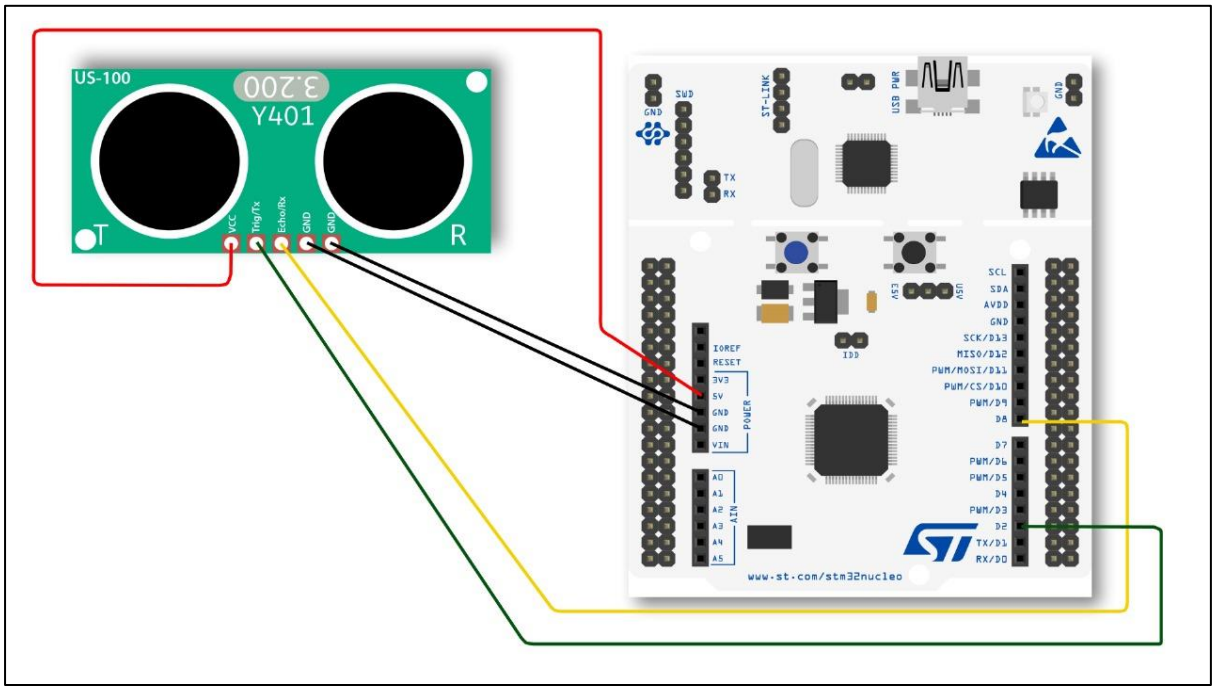
Objective:

The objectiveof this experiment is to interface an ultrasonic sensor (US 100) to an STM32 microcontroller.

Requirements:

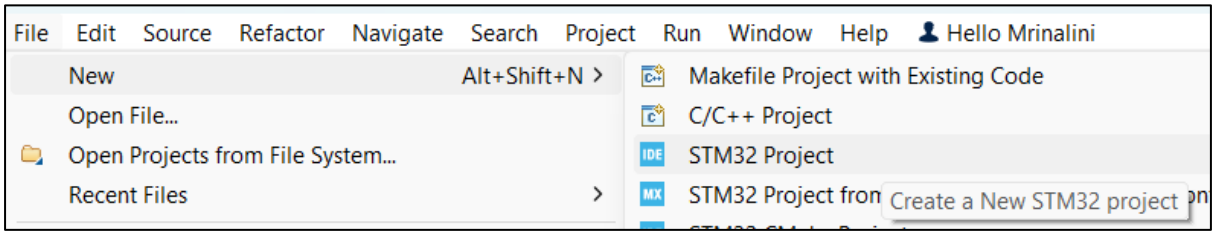
1. STM32 Cube IDE software.
2. Ultrasonic Sensor (US 100).
3. STM32 Microcontroller.
4. USB Cable for the microcontroller.
5. Jumper Wires.

Connection Diagram:

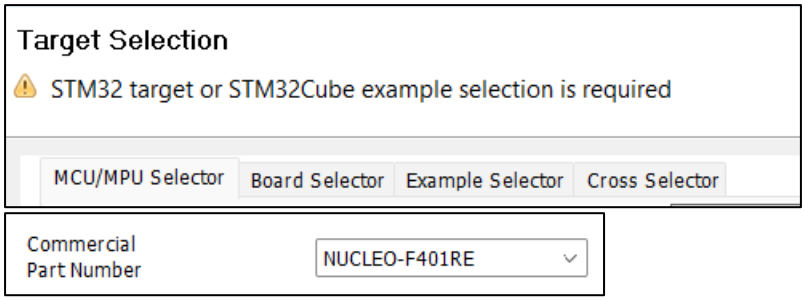


Procedure:

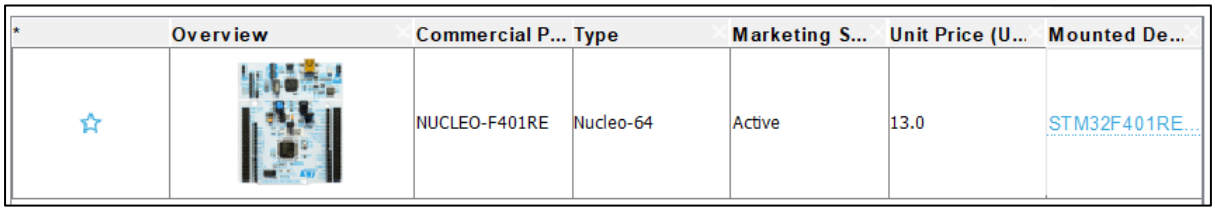
1. Click on **File →New →STM32 Project** to start your project on Cube IDE.



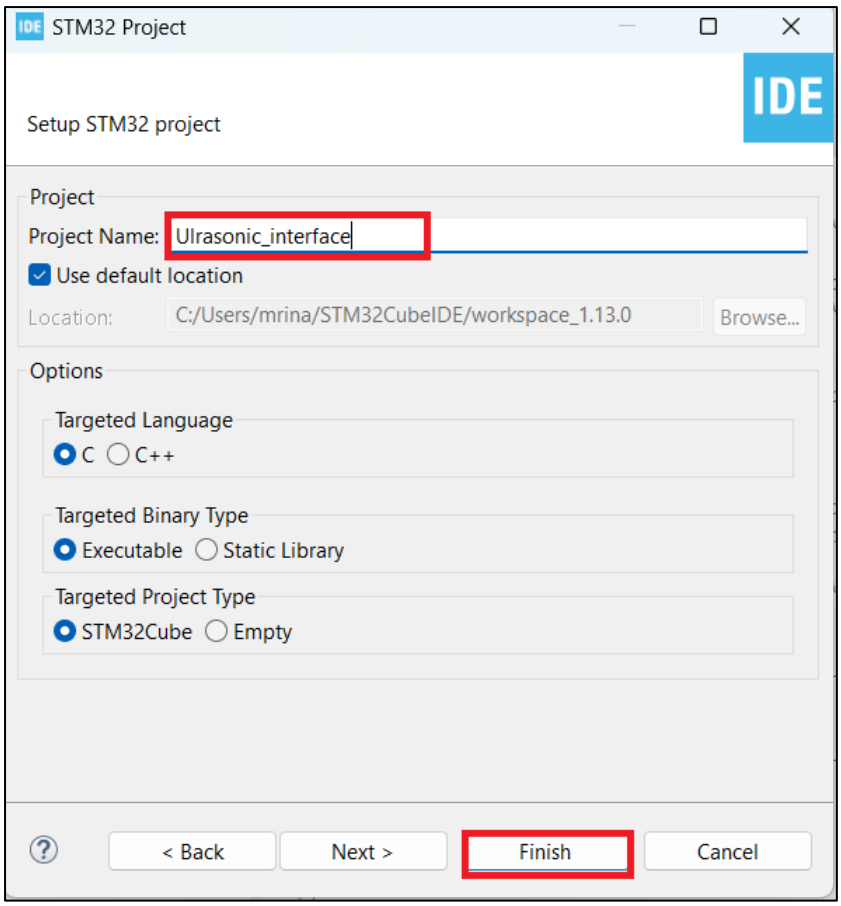
2. A **Target Selection** window will open. Click on **Board Selector**, where you need to select the microcontroller board you are working with (NUCLEO-F401RE/NUCLEO-F411RE).



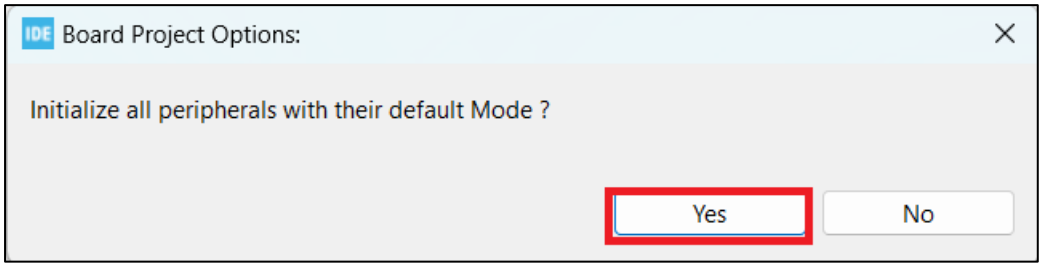
3. After this on the right-hand side of the window, under **Board List** you will see the board you have selected. Click on the board and then click on **Next.**



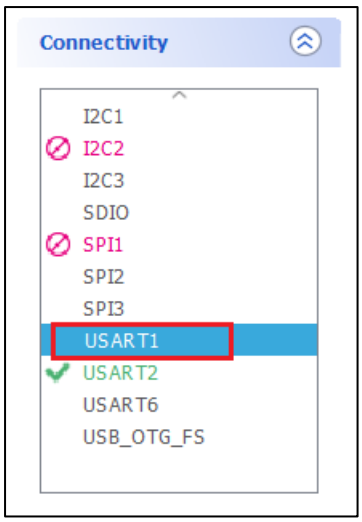
4. Give your project a name, rest of the things will remain by default as it is for now. Click on **FINISH.**

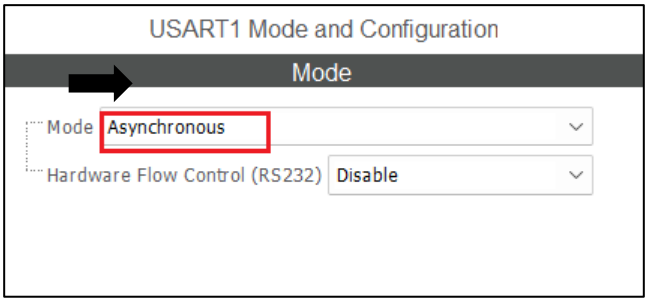


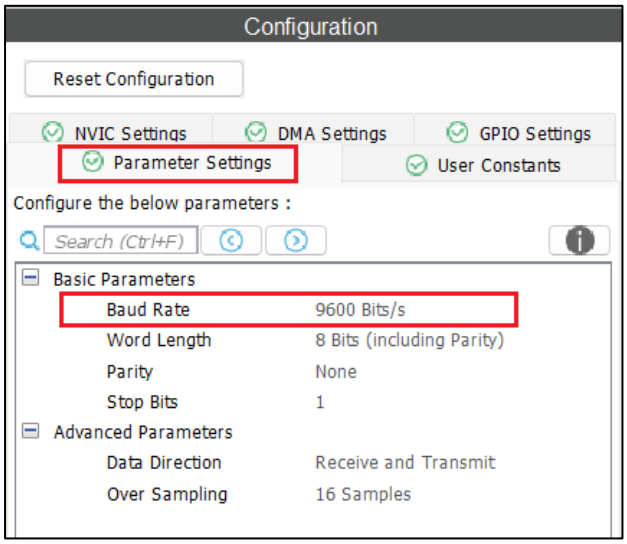
5. Cube IDE will ask if you want to initialize all peripherals with their default mode, click on **Yes.**

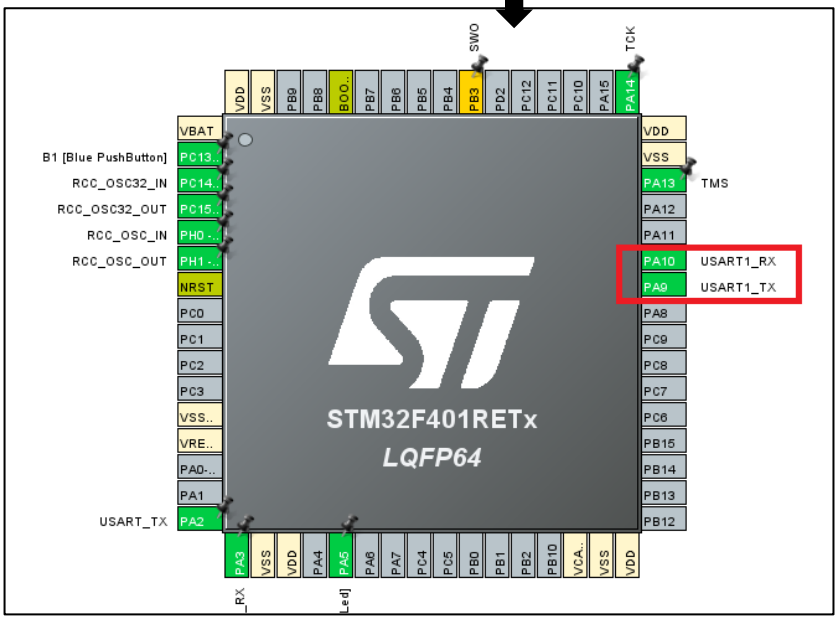


6. In left hand side **Categories→Connectivity**select **USART1**. Then in **USART1 mode & configuration** under mode select Asynchronous. Then go to configuration select **Parameter settings→Basic parameter**, change baud rate to **9600 Bits/s**. In right hand side, pinout view you can see that **PA10**&**PA9**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

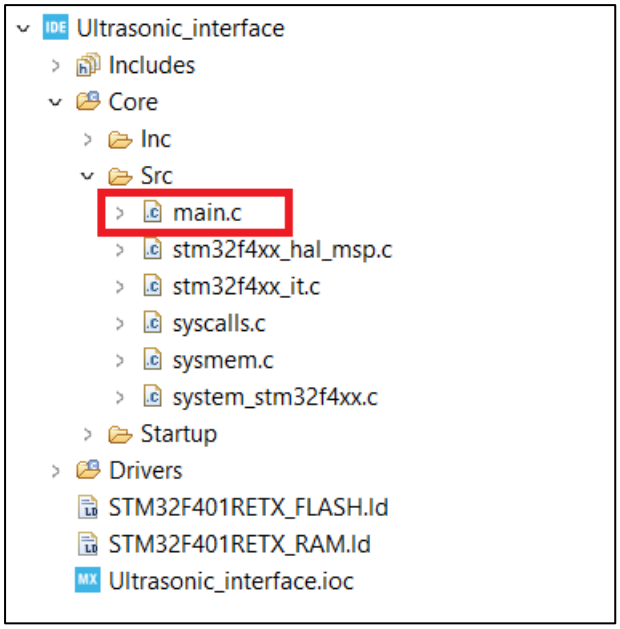

Connectivity
I2C1
I2C2
I2C3
SDIO
SPI1
SPI2
SPI3
USART1
USART2
USART6
USB_OTG_FS


USART1 Mode and Configuration
Mode
Mode
Asynchronous
Hardware Flow Control (RS232)
Disable

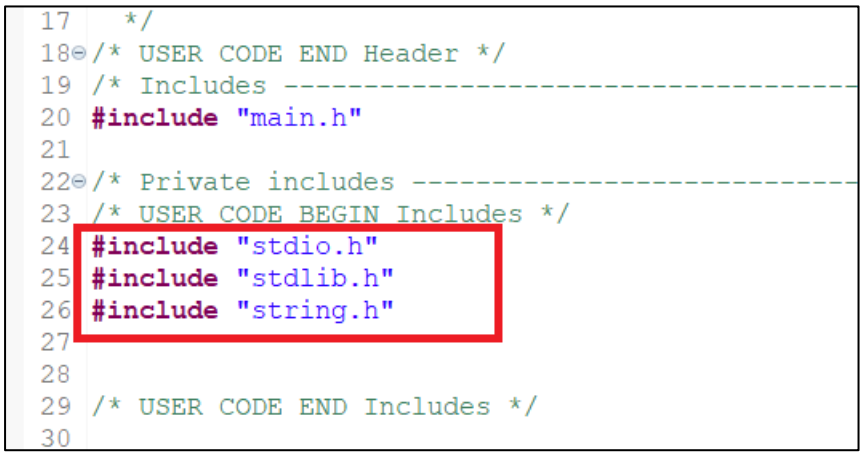

Configuration
Reset Configuration
NVIC Settings
DMA Settings
GPIO Settings
Parameter Settings
User Constants
Configure the below parameters :
Search (Ctrl+F)
Basic Parameters
Baud Rate
9600 Bits/s
Word Length
8 Bits (including Parity)
Parity
None
Stop Bits
1
Advanced Parameters
Data Direction
Receive and Transmit
Over Sampling
16 Samples


SWO
TCK
VDD
VSS
PB9
PB8
BOO..
PB7
PB6
PB5
PB4
PB3
PD2
PC12
PC11
PC10
PA15
PA14
VBAT
B1 [Blue PushButton]
PC13..
RCC_OSC32_IN
PC14..
RCC_OSC32_OUT
PC15..
RCC_OSC_IN
PH0 -..
RCC_OSC_OUT
PH1 -..
NRST
PC0
PC1
PC2
PC3
VSS..
VRE..
PA0-..
PA1
USART_TX
PA2
VDD
VSS
PA13
TMS
PA12
PA11
PA10
USART1_RX
PA9
USART1_TX
PA8
PC9
PC8
PC7
PC6
PB15
PB14
PB13
PB12
STM32F401RETx
LQFP64
PA3
VSS
VDD
PA4
PA5
PA6
PA7
PC4
PC5
PB0
PB1
PB2
PB10
VCA..
VSS
VDD
_RX
Led]

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_interface) **Ultrasonic_interface→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.

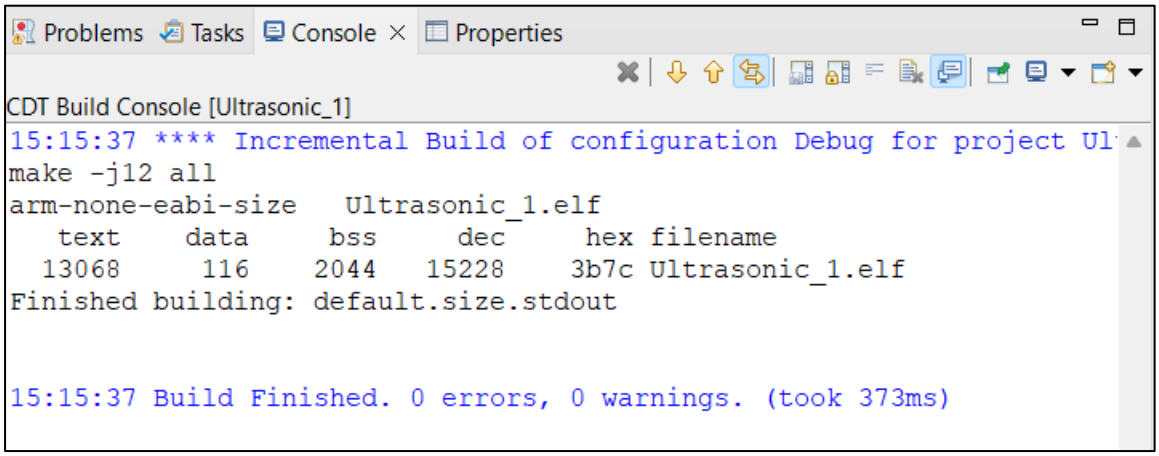


```
35
36⊖/* Private define ------------------------------------
37 /* USER CODE BEGIN PD */
38 uint8_t txData = 0x55;
39 uint8_t rxData[2];
40 uint16_t distance;
41
42 /* USER CODE END PD */
43
44⊖/* Private macro -------------------------------------
45 /* USER CODE BEGIN PM */
46
47 /* USER CODE END PM */
48
49 /* Private variables ---------------------------------
50 UART_HandleTypeDef huart1;
51 UART_HandleTypeDef huart2;
52
53⊖/* USER CODE BEGIN PV */
```

```
106   /* Infinite loop */
107   /* USER CODE BEGIN WHILE */
108   while (1)
109   {
110⊖//    /* Clear the RX buffer before sending trigger command */
111 //          HAL_UART_Receive(&huart1, rxData, 2, HAL_MAX_DELAY);
112
113             /* Send trigger command */
114             if(HAL_UART_Transmit(&huart1, &txData, 1, HAL_MAX_DELAY)!= HAL_OK){
115                 Error_Handler();
116                 HAL_Delay(50);
117             }
118
119
120             /* Receive and process data if available */
121             if (HAL_UART_Receive(&huart1, rxData, 2, HAL_MAX_DELAY) == HAL_OK) {
122                 uint16_t highByte = rxData[0];
123                 uint16_t lowByte = rxData[1];
124                 x = ((highByte << 8) | lowByte);
125                 distance = x/10;
126
127                 if (distance >= 1 && distance < 10000) {
128                     printf("Distance: %d cm\r\n", distance);
129                 }
130             }
131                 HAL_Delay(300);
132     /* USER CODE END WHILE */
```

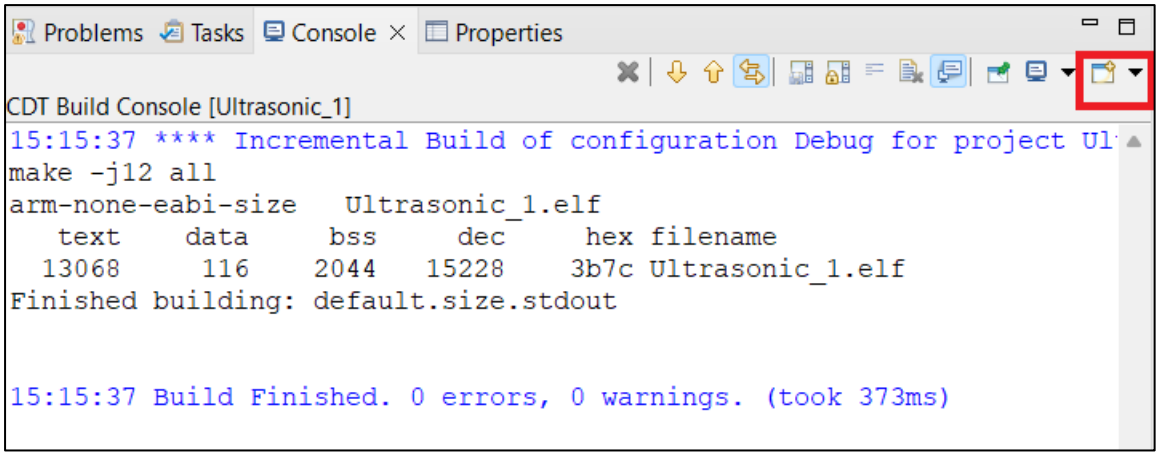
```
286
287 /* USER CODE BEGIN 4 */
288 int __io_putchar(int ch){
289     HAL_UART_Transmit(&huart2, (uint8_t *)&ch, 1, HAL_MAX_DELAY);
290     return ch;
291 }
292 /* USER CODE END 4 */
293
294 /**
295   * @brief  This function is executed in case of error occurrence.
296   * @retval None
297   */
298 void Error_Handler(void)
299 {
300   /* USER CODE BEGIN Error_Handler_Debug */
301   /* User can add his own implementation to report the HAL error return state */
302 //    disable_irq();
303     printf("Transmit Error!");
304   while (1)
305   {
306
```

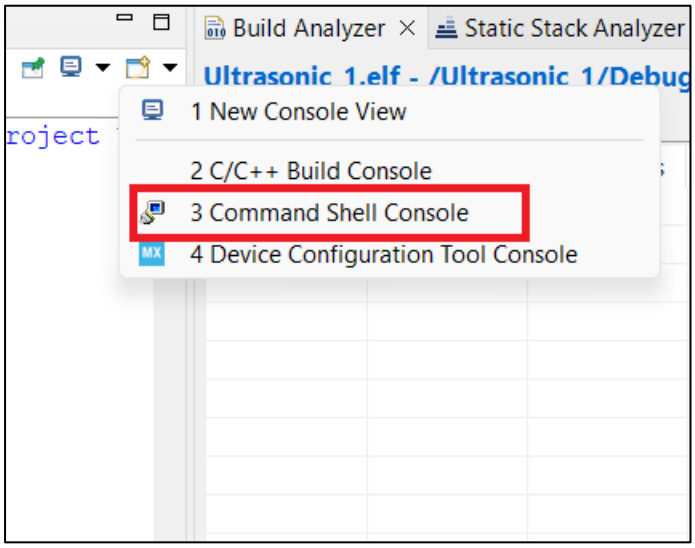
9. Now click on the build  symbol on the top left corner on your Cube IDE. If you have done everything correctly your code should be built without any errors.

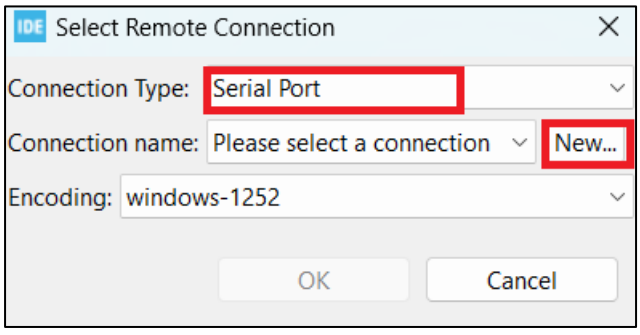


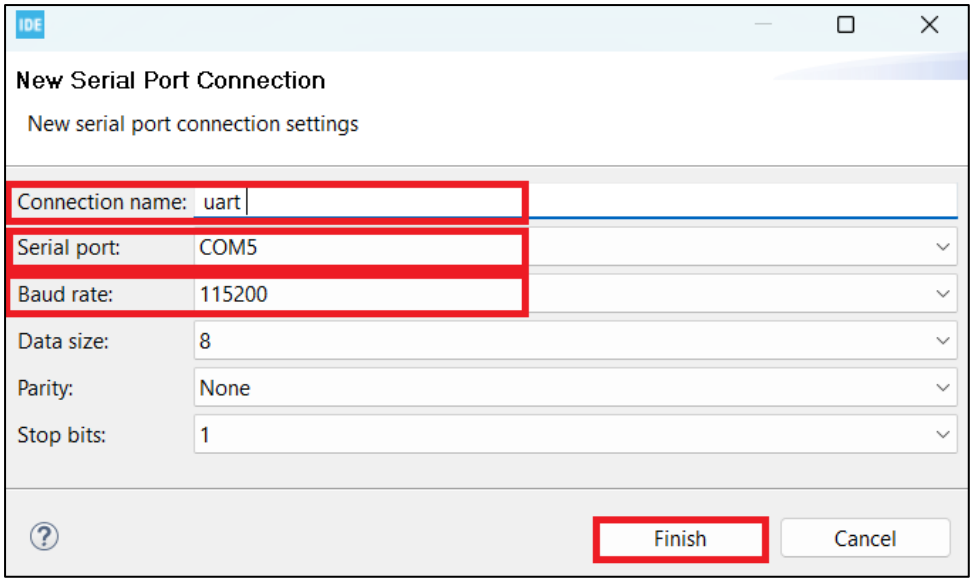
10. Next connect your STM32 board with your audio sensor connect to it to your PC and click on the **Debug**  icon to start the Debugging process. An **Edit Configuration** window will open, click on **OK,** without making any changes.

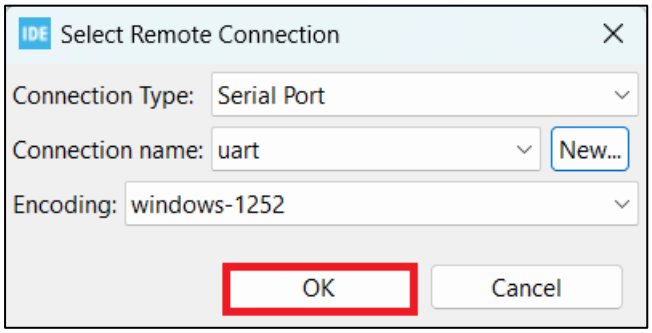
11. In the debug mode, go to the bottom right hand side corner, click on open console. Select the **Connection Type** as **Serial Port**, then click on **New.** In the new window, in **Connection name** give some name to your new connection, select the **Serial port** correctly, and **baud rate** should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


Problems
Tasks
Console
Properties
CDT Build Console [Ultrasonic_1]
15:15:37 **** Incremental Build of configuration Debug for project Ul
make -j12 all
arm-none-eabi-size Ultrasonic_1.elf
text data bss dec hex filename
13068 116 2044 15228 3b7c Ultrasonic_1.elf
Finished building: default.size.stdout
15:15:37 Build Finished. 0 errors, 0 warnings. (took 373ms)

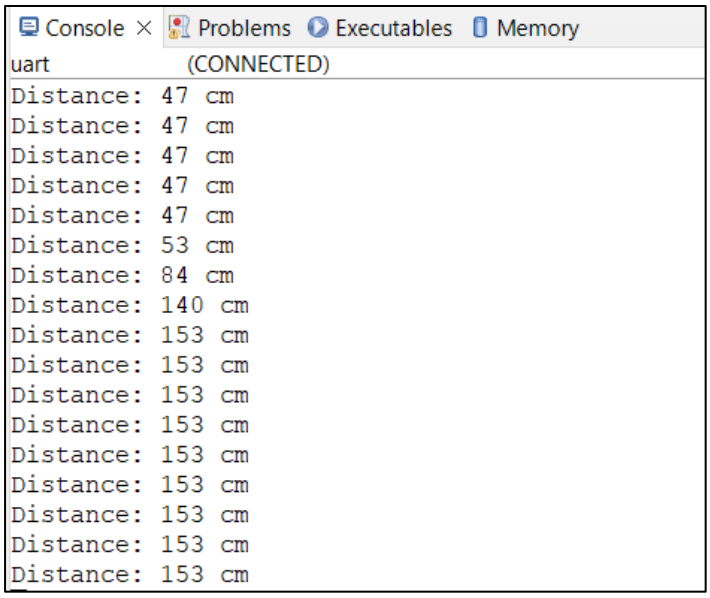

Build Analyzer
Static Stack Analyzer
Ultrasonic_1.elf - /Ultrasonic_1/Debug
1 New Console View
2 C/C++ Build Console
3 Command Shell Console
4 Device Configuration Tool Console


Select Remote Connection
Connection Type: Serial Port
Connection name: Please select a connection
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 distance in cm.



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.