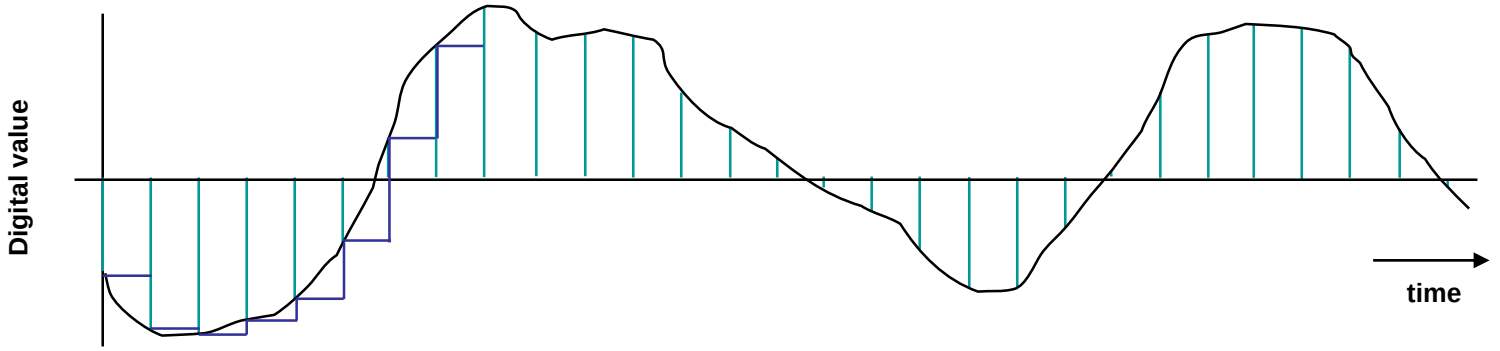
Lab 1

A/D and D/A Converter

Analog and Digital

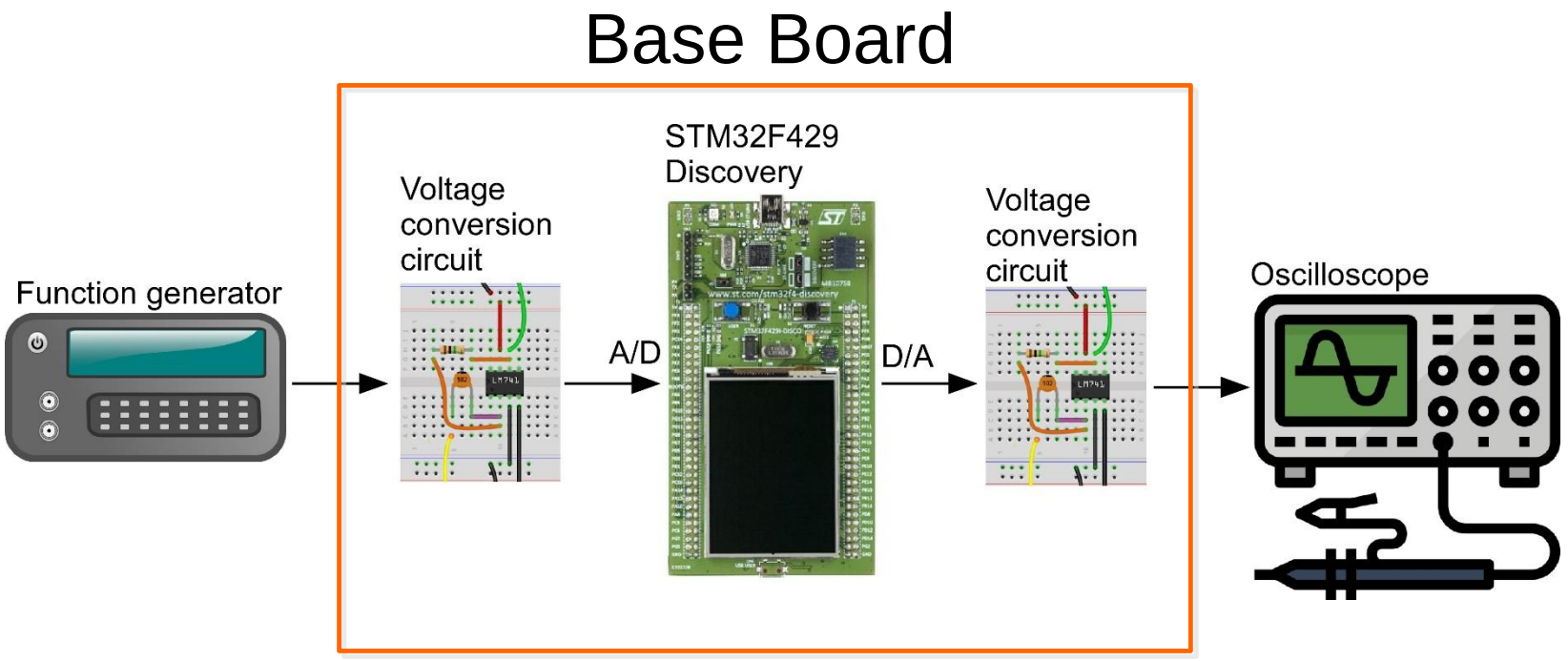
- CD digital audio encoding: 2-channel signed 16-bit Linear PCM sampled at 44,100 Hz.

Waveform Sampling and Quantization



- A waveform is **sampled** at a constant rate – every Δ_t
  - Each such sample represents the instantaneous amplitude at the instant of sampling
  - *"At 37 ms, the input is 1.913419145134514512343 11… V"*
  - Sampling converts a **continuous time** signal to a **discrete time** signal
- The sample can now be **quantized** (converted) into a digital value
  - Quantization represents a **continuous** (analog) value with the closest **discrete** (digital) value
  - *"The sampled input voltage of 1.913419145134514512343 11… V is best represented by the code 0x018, since it is in the range of 1.901 to 1.9980 V which corresponds to code 0x018."*

Lab Setup


Base Board
Function generator
Voltage conversion circuit
A/D
STM32F429 Discovery
D/A
Voltage conversion circuit
Oscilloscope

STM32F429 Discovery Board

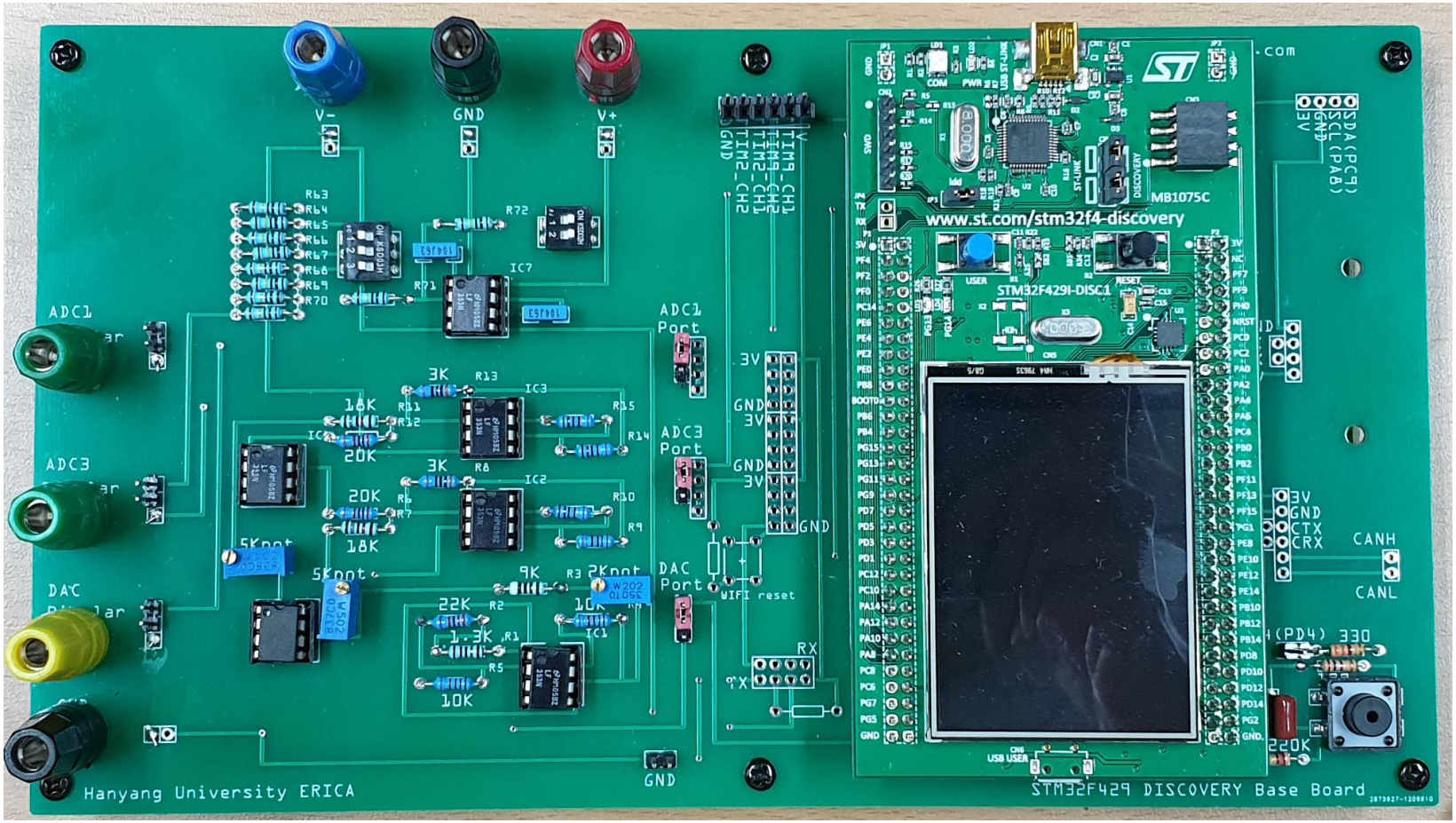
-15V GND +15V

Mini USB: To PC

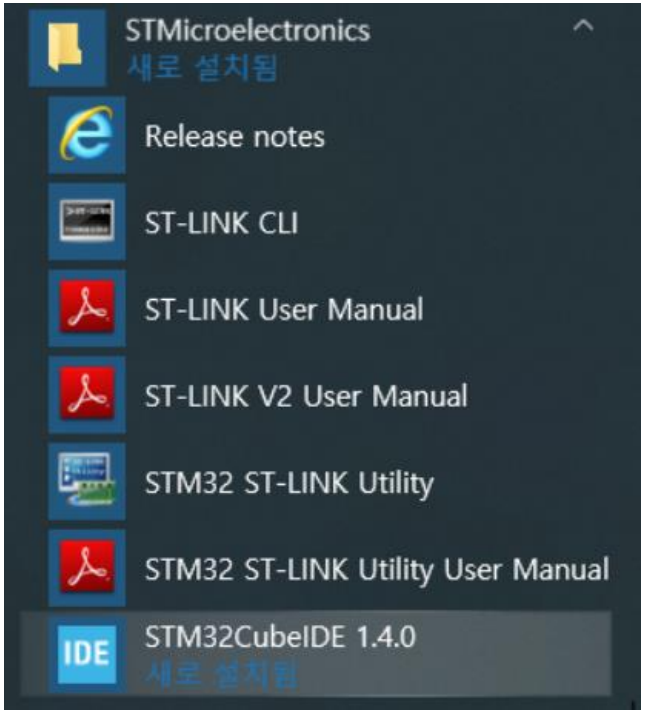
ADC input

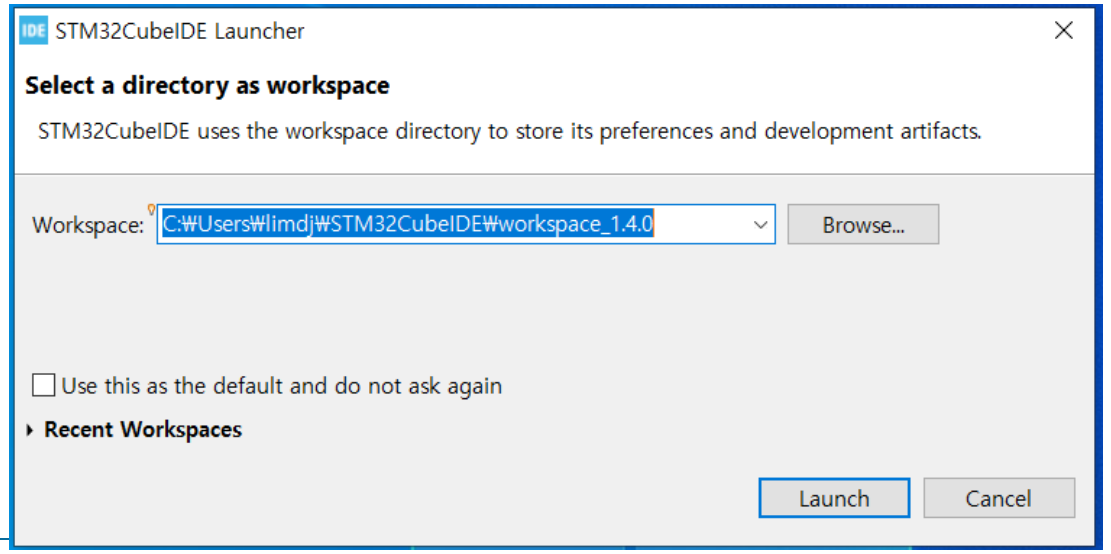
DAC output

GND

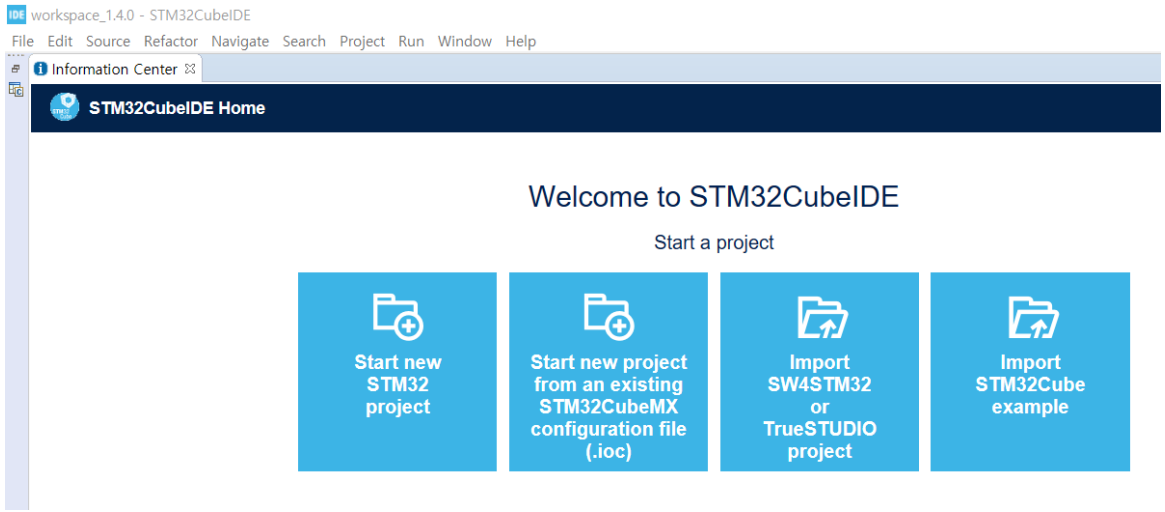


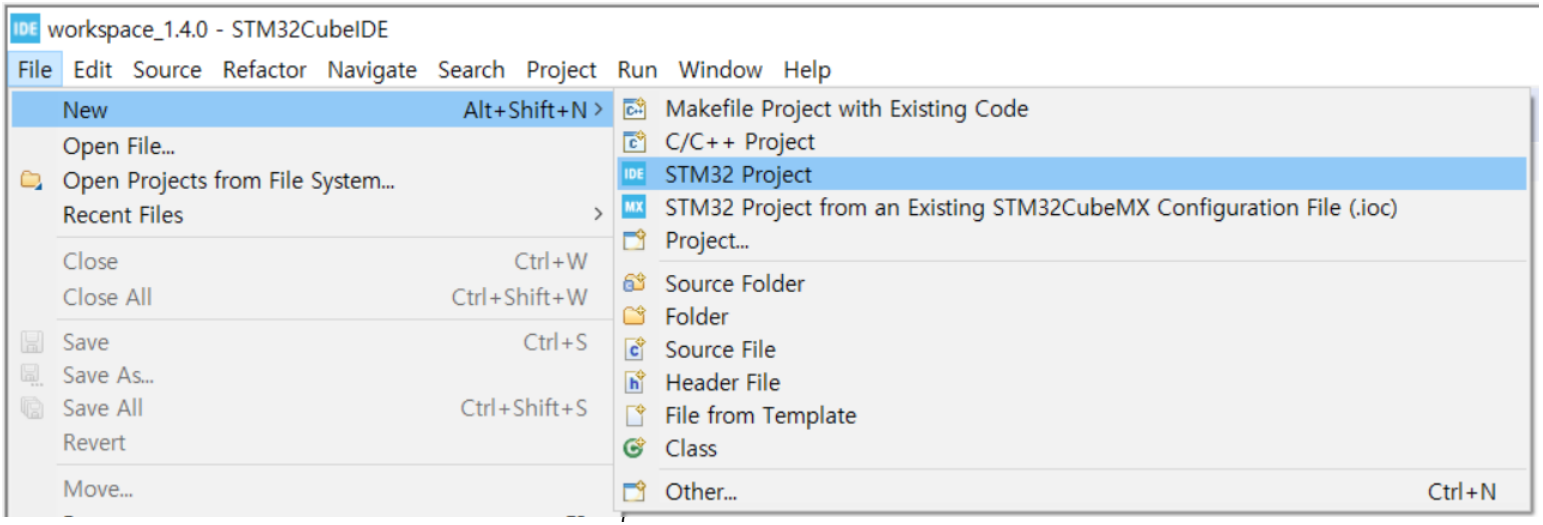
Start STM32CubeIDE





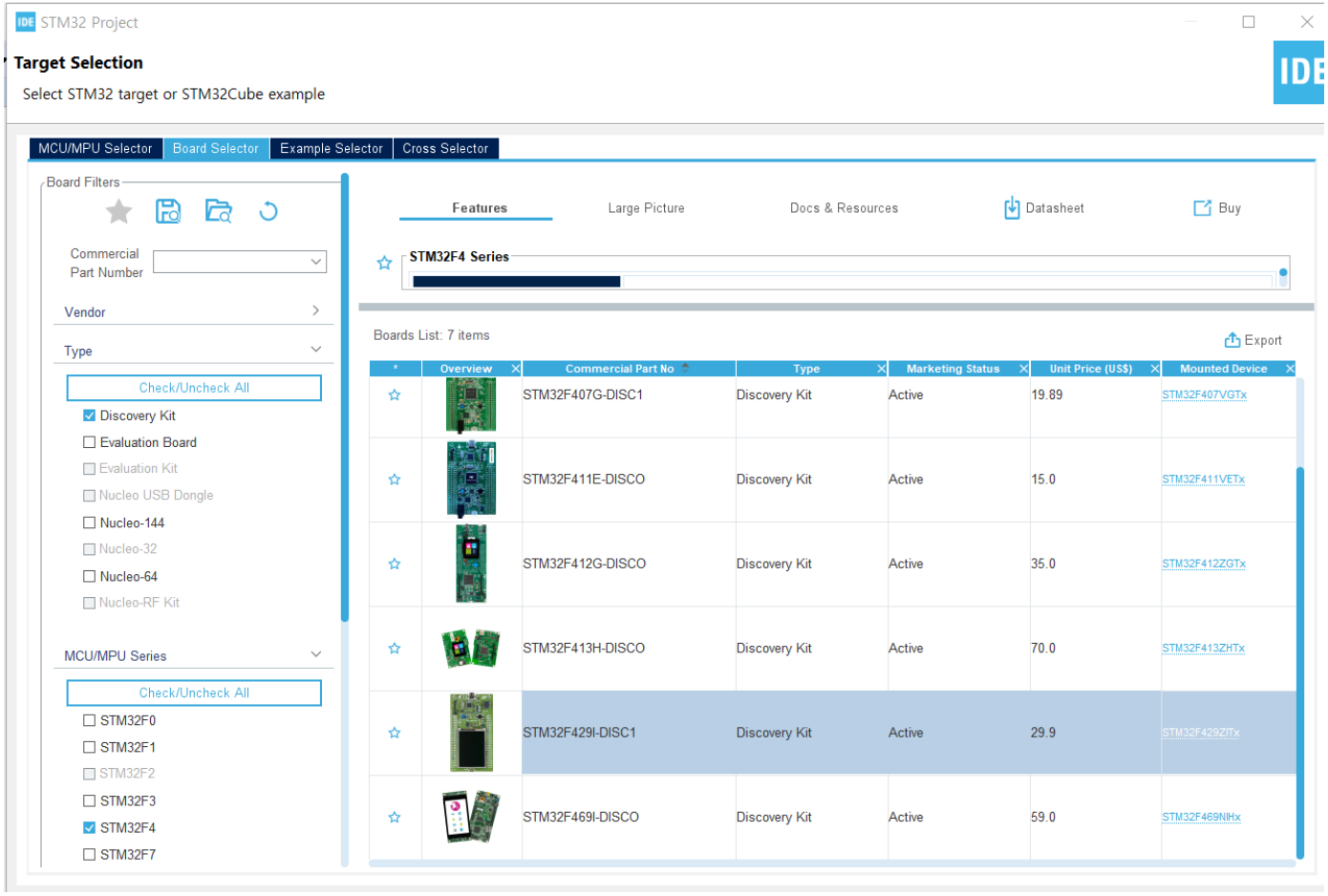
Start New STM32 Project





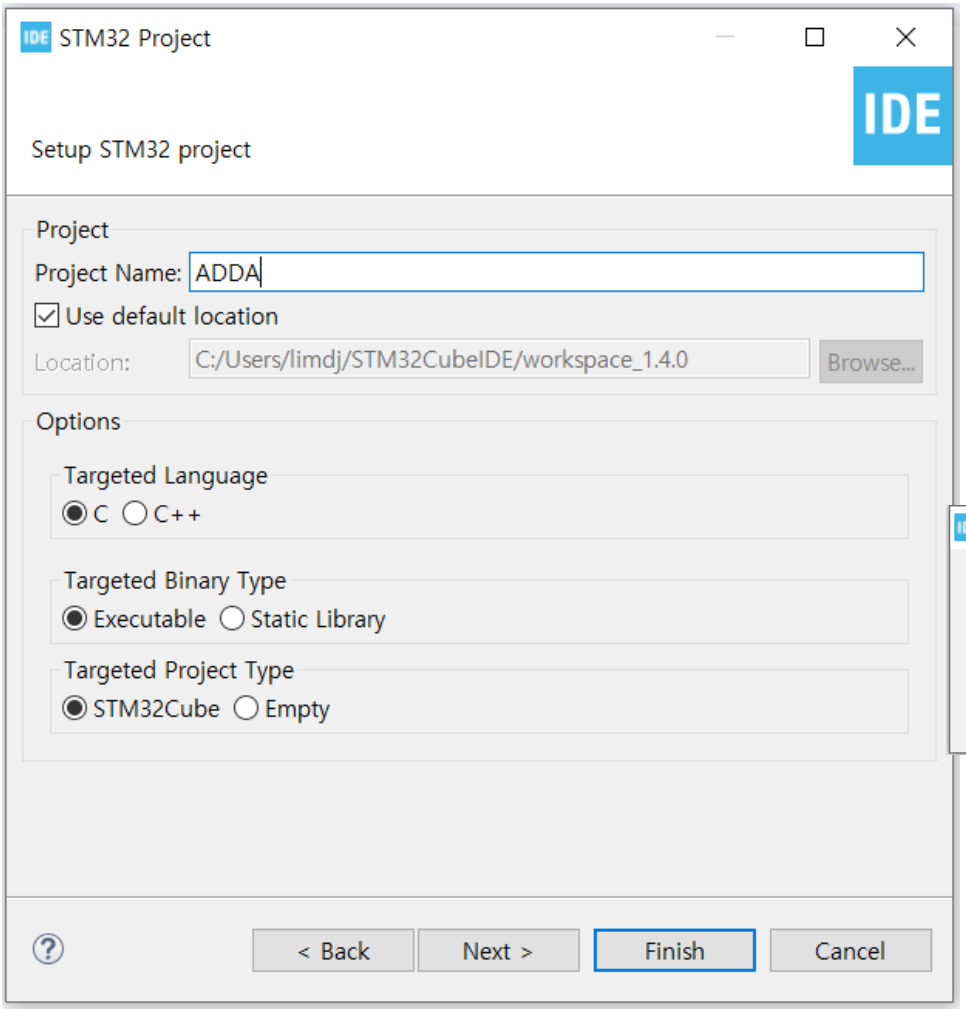
Board Selector

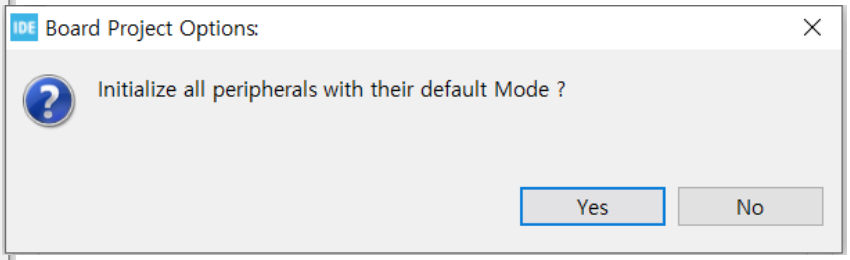
- Select STM32F-429I-DISC1
- Click Next

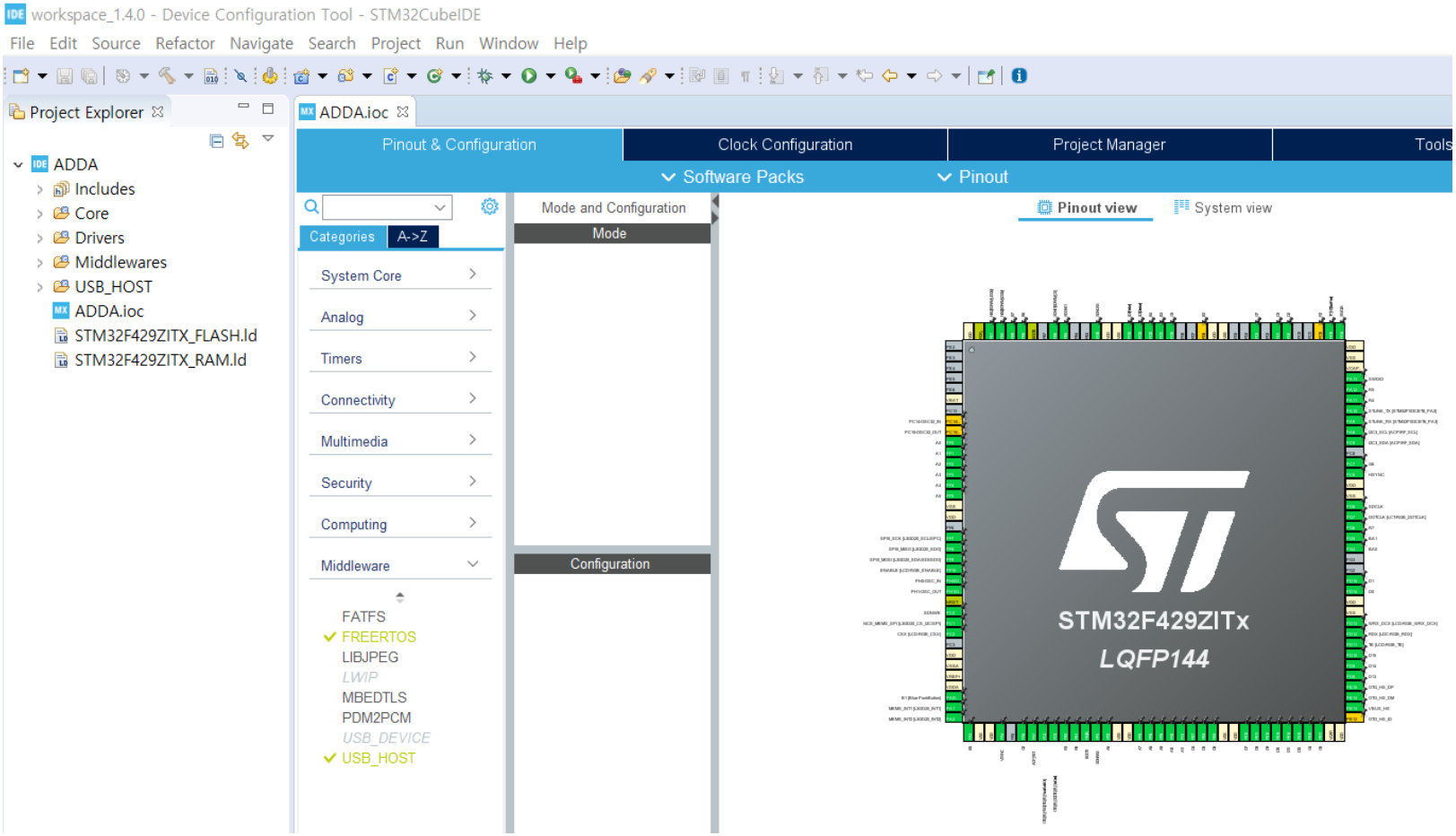


Setup Project

- Project Name:ADDA

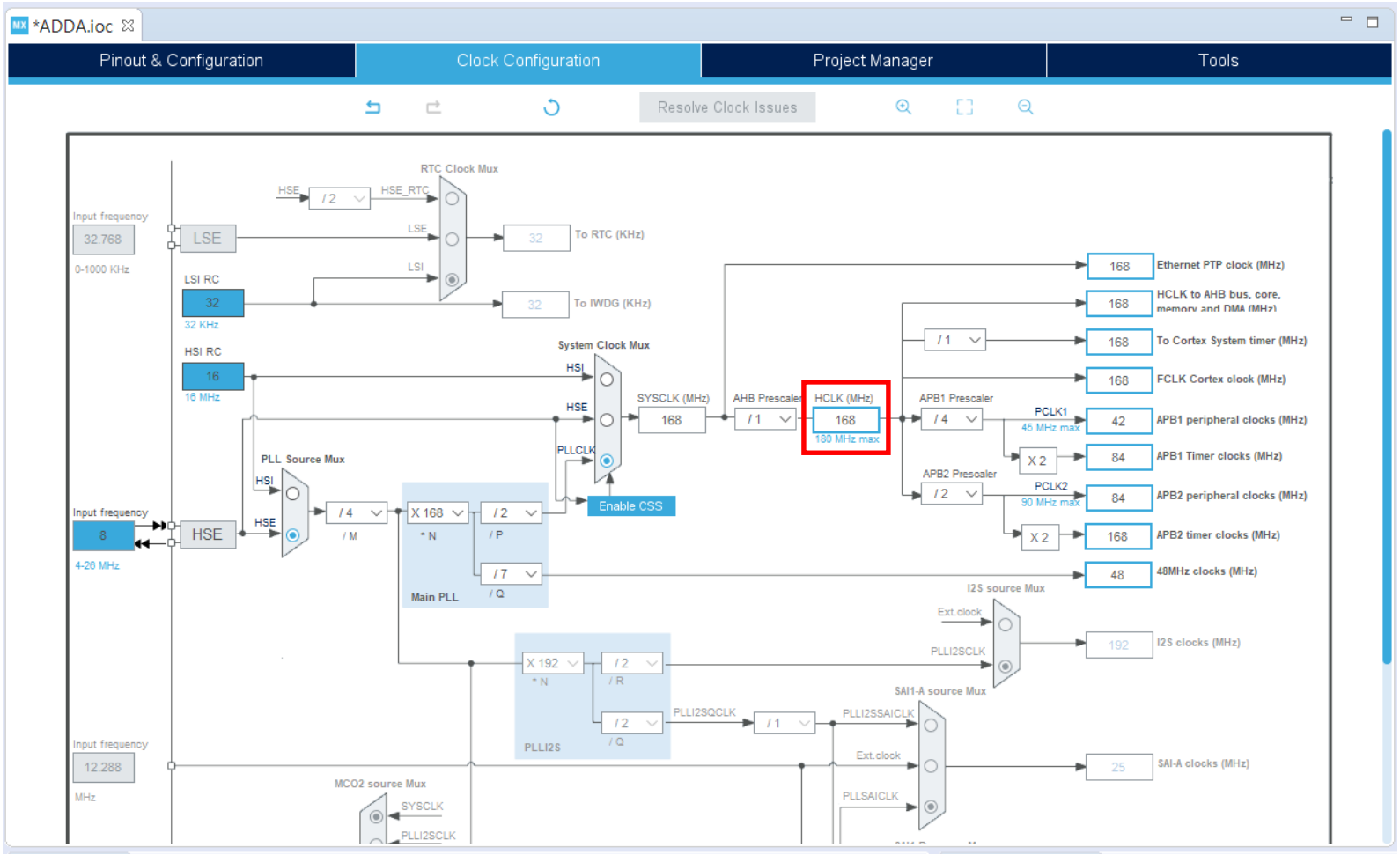




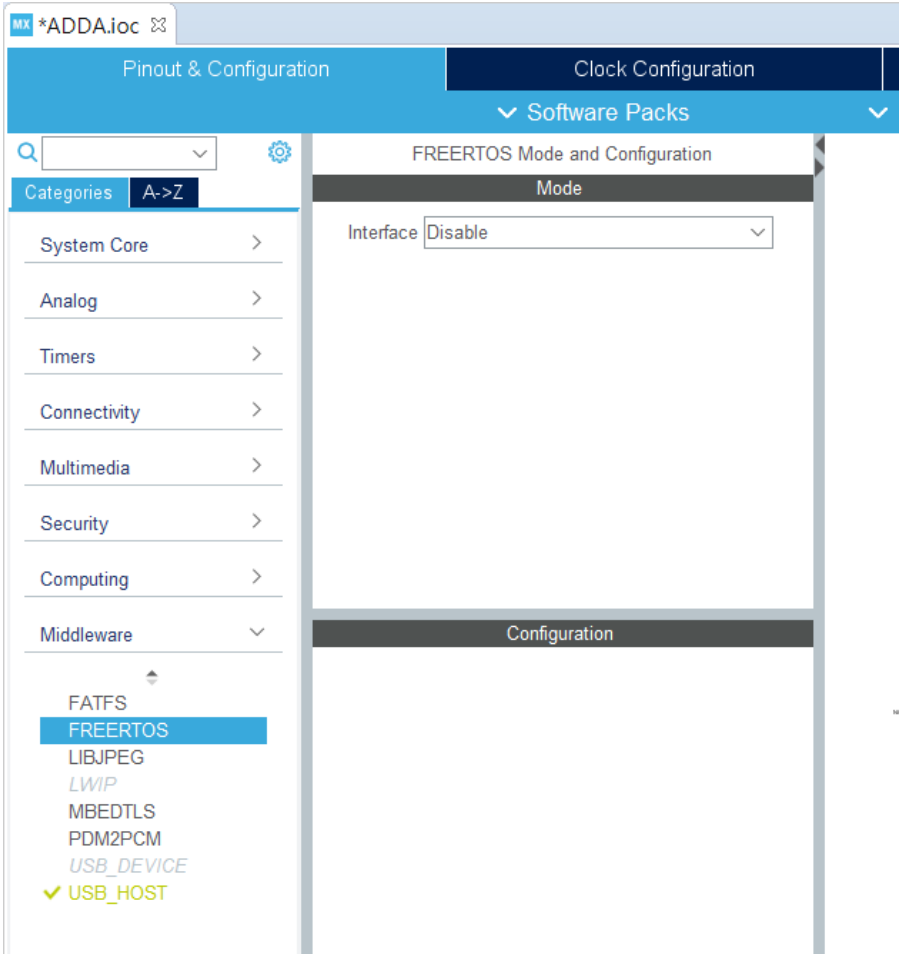

workspace_1.4.0 - Device Configuration Tool - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ADDA
Includes
Core
Drivers
Middlewares
USB_HOST
ADDA.ioc
STM32F429ZITX_FLASH.ld
STM32F429ZITX_RAM.ld
ADDA.ioc
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Software Packs
Pinout
Categories
A->Z
System Core
Analog
Timers
Connectivity
Multimedia
Security
Computing
Middleware
FATFS
FREERTOS
LIBJPEG
LWIP
MBEDTLS
PDM2PCM
USB_DEVICE
USB_HOST
Mode and Configuration
Mode
Configuration
Pinout view
System view
STM32F429ZITx
LQFP144

Clock Configuration

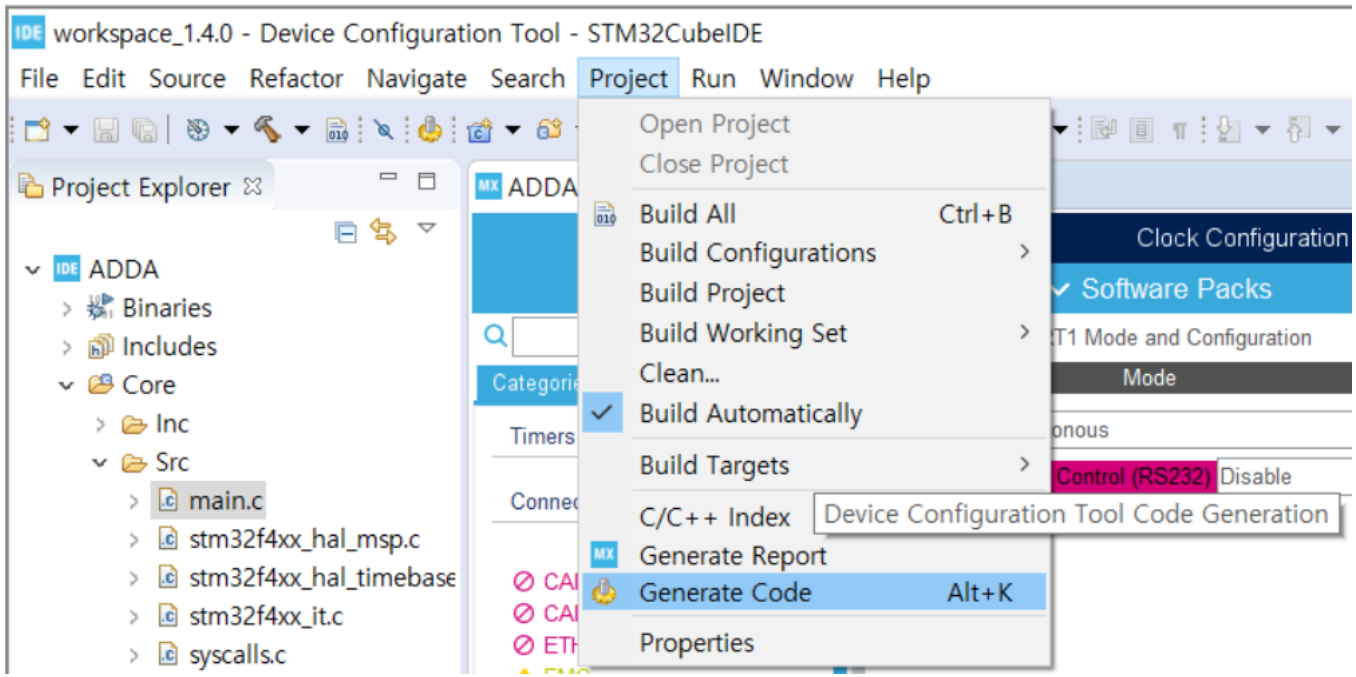
- HCLK (MHZ): 168

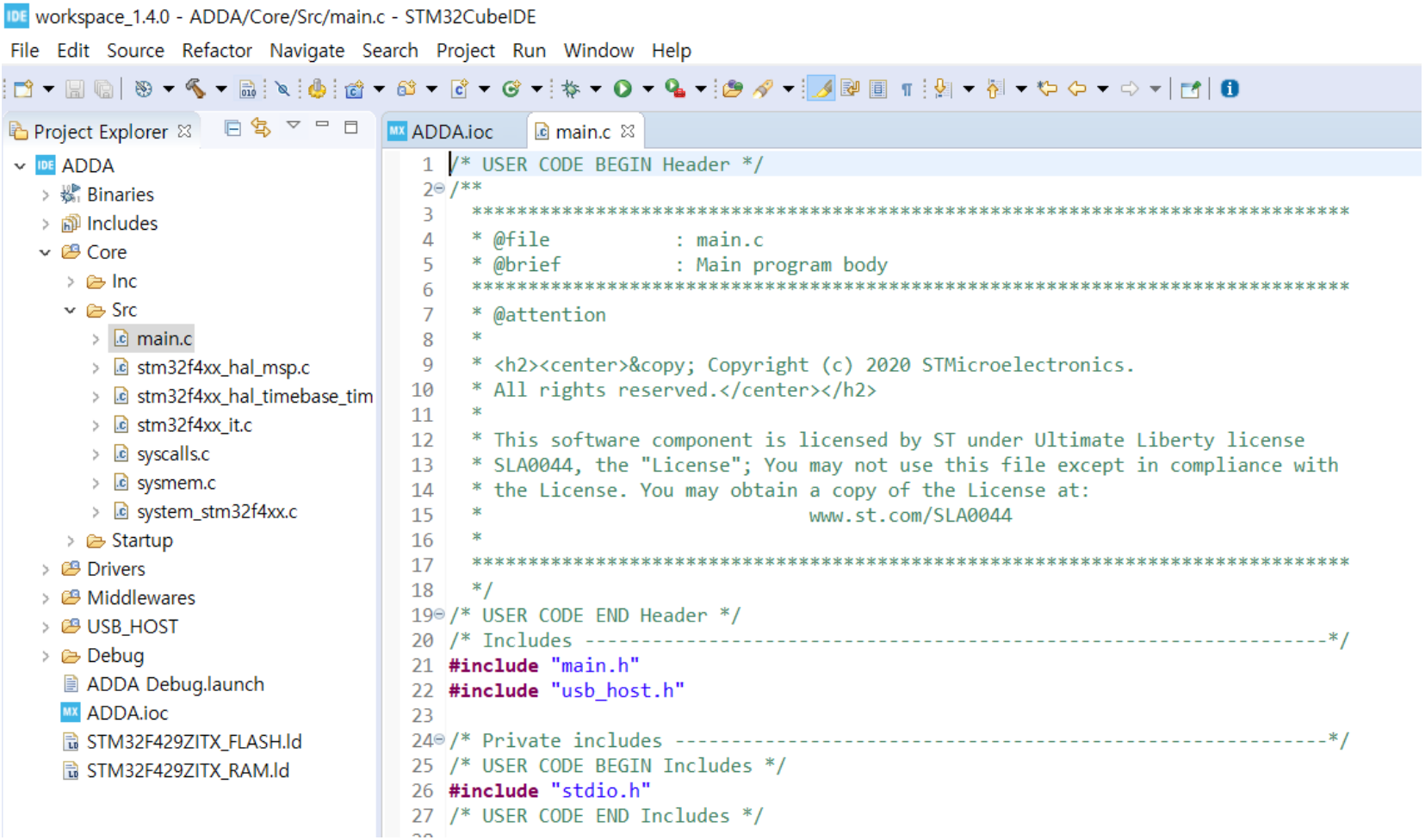


Disable FREERTOS



Generate Code

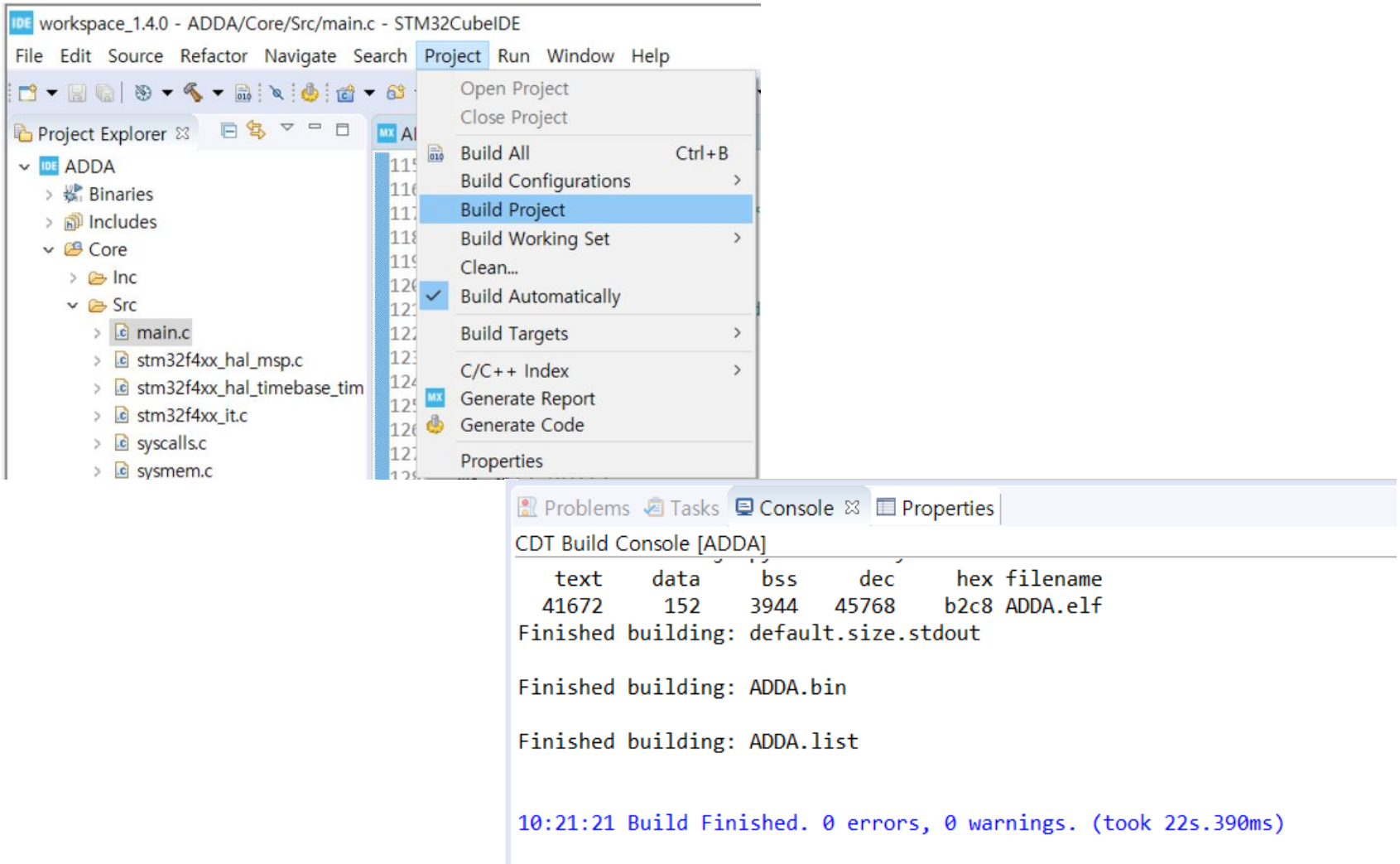



workspace_1.4.0 - ADDA/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ADDA
Binaries
Includes
Core
Inc
Src
main.c
stm32f4xx_hal_msp.c
stm32f4xx_hal_timebase_tim
stm32f4xx_it.c
syscalls.c
sysmem.c
system_stm32f4xx.c
Startup
Drivers
Middlewares
USB_HOST
Debug
ADDA Debug.launch
ADDA.ioc
STM32F429ZITX_FLASH.ld
STM32F429ZITX_RAM.ld
ADDA.ioc
main.c
/* USER CODE BEGIN Header */
/**
* @file : main.c
* @brief : Main program body
* @attention
* <h2> © Copyright (c) 2020 STMicroelectronics.
* All rights reserved. </h2>
* This software component is licensed by ST under Ultimate Liberty license
* SLA0044, the "License"; You may not use this file except in compliance with
* the License. You may obtain a copy of the License at:
* www.st.com/SLA0044
*/
/* USER CODE END Header */
/* Includes */
#include "main.h"
#include "usb_host.h"
/* Private includes */
/* USER CODE BEGIN Includes */
#include "stdio.h"
/* USER CODE END Includes */

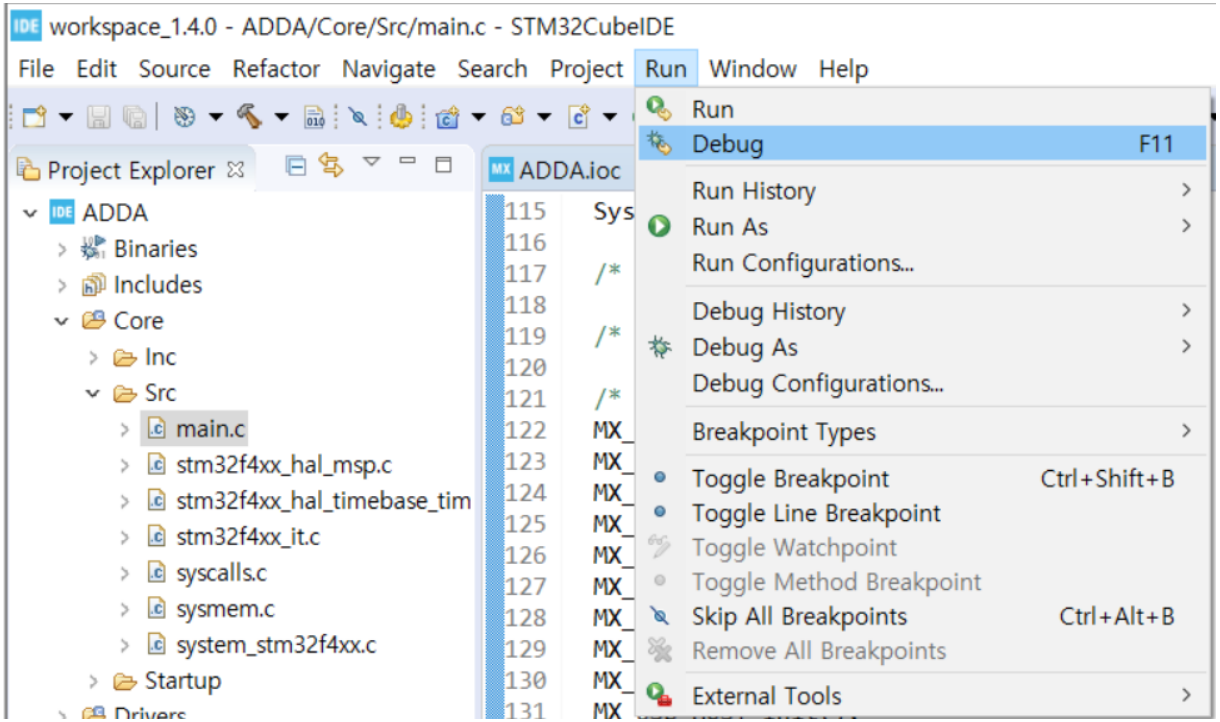
printf

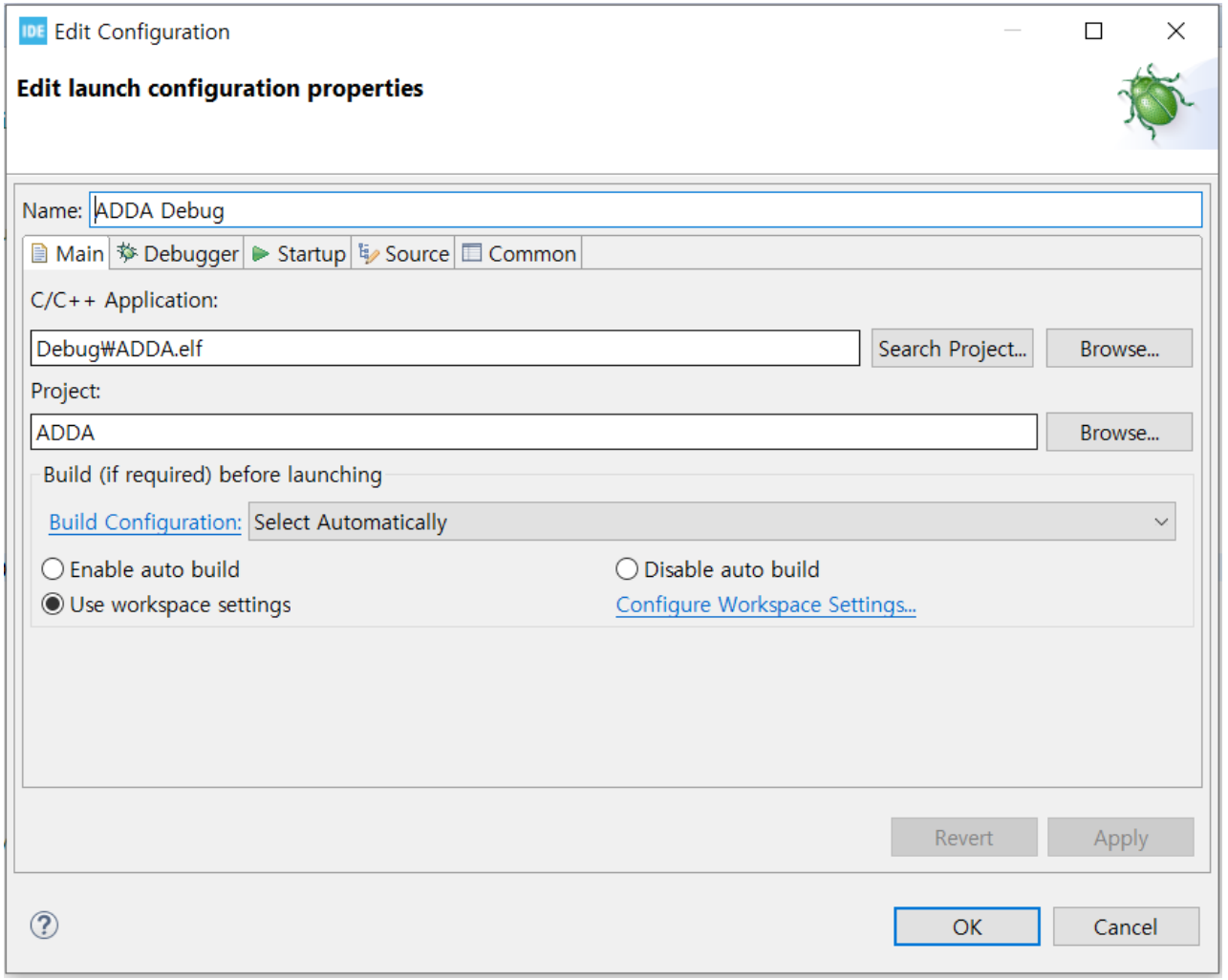
```
/* USER CODE BEGIN Includes */
#include "stdio.h"
/* USER CODE END Includes */

/* USER CODE BEGIN 0 */
#ifdef __GNUC__
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#else
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
#endif /* __GNUC__ */
PUTCHAR_PROTOTYPE
{
 HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
return ch;
}
/* USER CODE END 0 */

/* USER CODE BEGIN 2 */
printf("Hello World\r\n");
/* USER CODE END 2 */
```

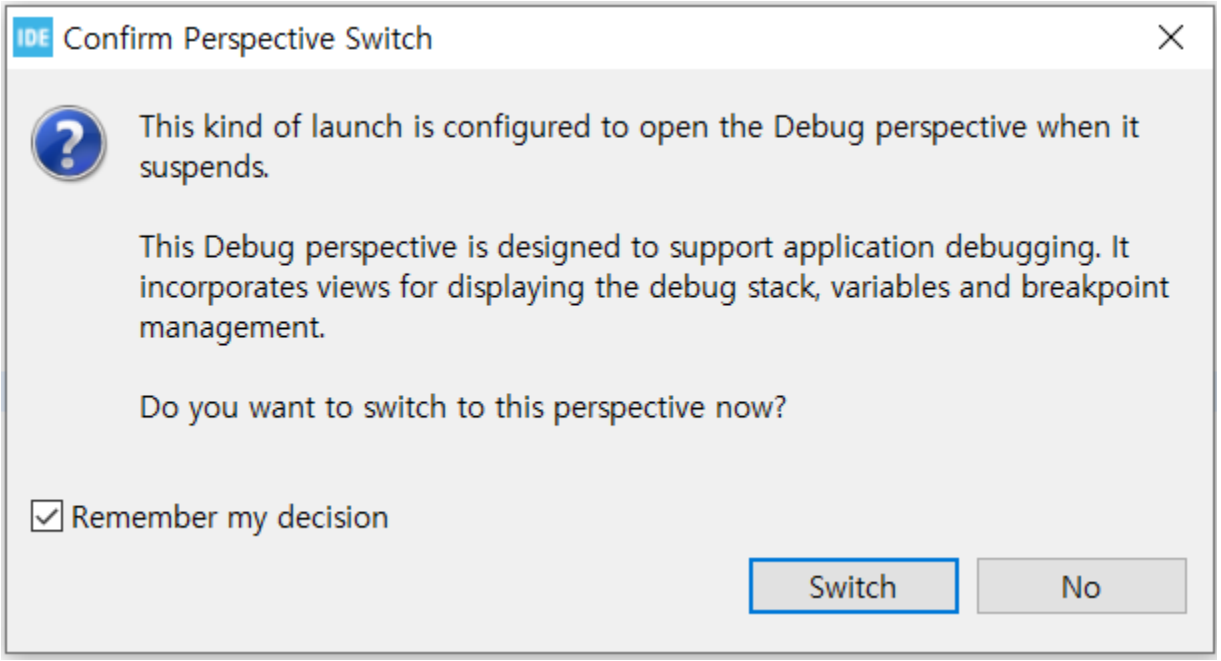
Build Project



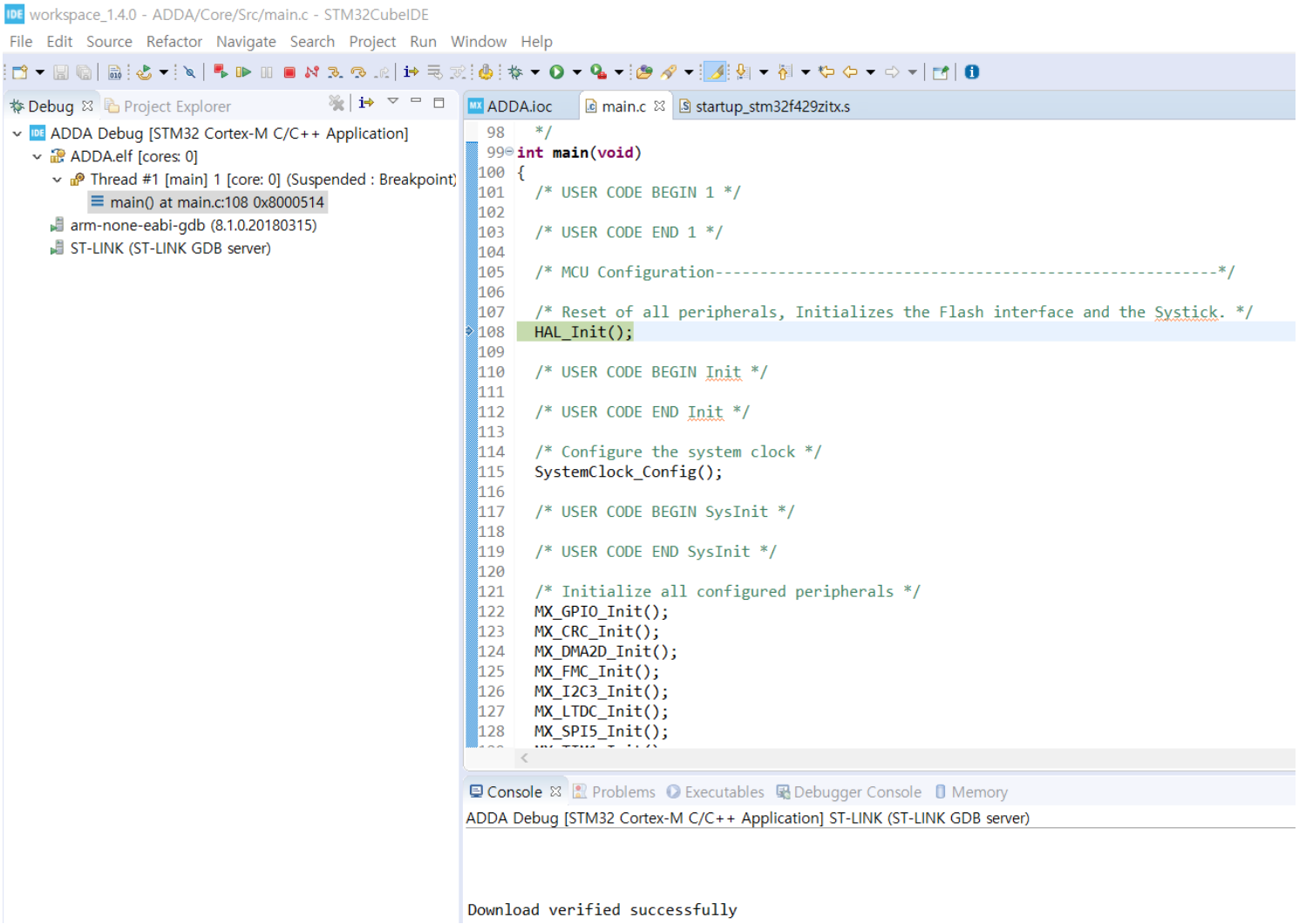
Start Debug


workspace_1.4.0 - ADDA/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
ADDA
Binaries
Includes
Core
Inc
Src
main.c
stm32f4xx_hal_msp.c
stm32f4xx_hal_timebase_tim
stm32f4xx_it.c
syscalls.c
sysmem.c
system_stm32f4xx.c
Startup
ADDA.ioc
Run
Debug F11
Run History
Run As
Run Configurations...
Debug History
Debug As
Debug Configurations...
Breakpoint Types
Toggle Breakpoint Ctrl+Shift+B
Toggle Line Breakpoint
Toggle Watchpoint
Toggle Method Breakpoint
Skip All Breakpoints Ctrl+Alt+B
Remove All Breakpoints
External Tools

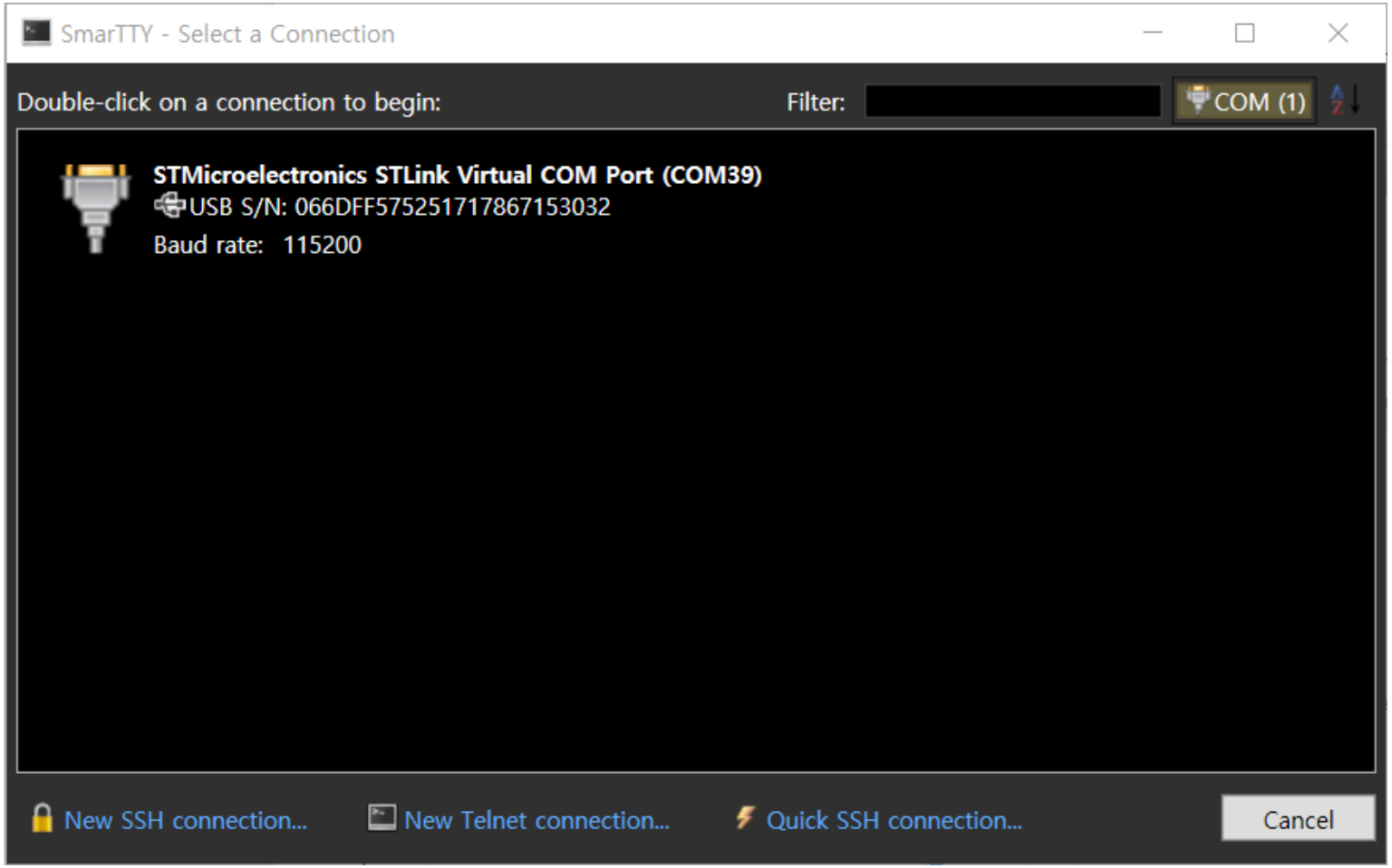

Edit Configuration
Edit launch configuration properties
Name: ADDA Debug
Main
Debugger
Startup
Source
Common
C/C++ Application:
Debug₩ADDA.elf
Search Project...
Browse...
Project:
ADDA
Browse...
Build (if required) before launching
Build Configuration:
Select Automatically
Enable auto build
Disable auto build
Use workspace settings
Configure Workspace Settings...
Revert
Apply
OK
Cancel


Confirm Perspective Switch
This kind of launch is configured to open the Debug perspective when it suspends.
This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.
Do you want to switch to this perspective now?
Remember my decision
Switch
No

Debugging Screen

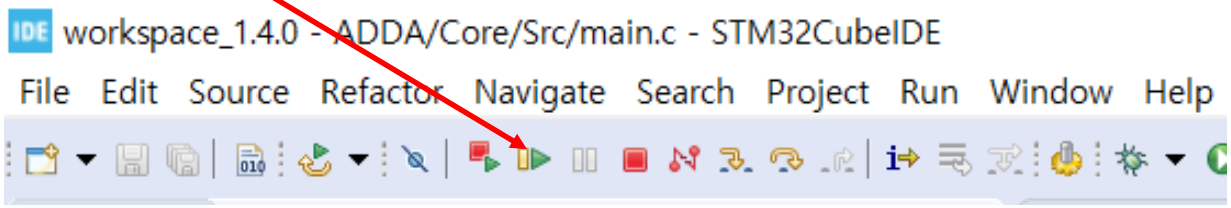


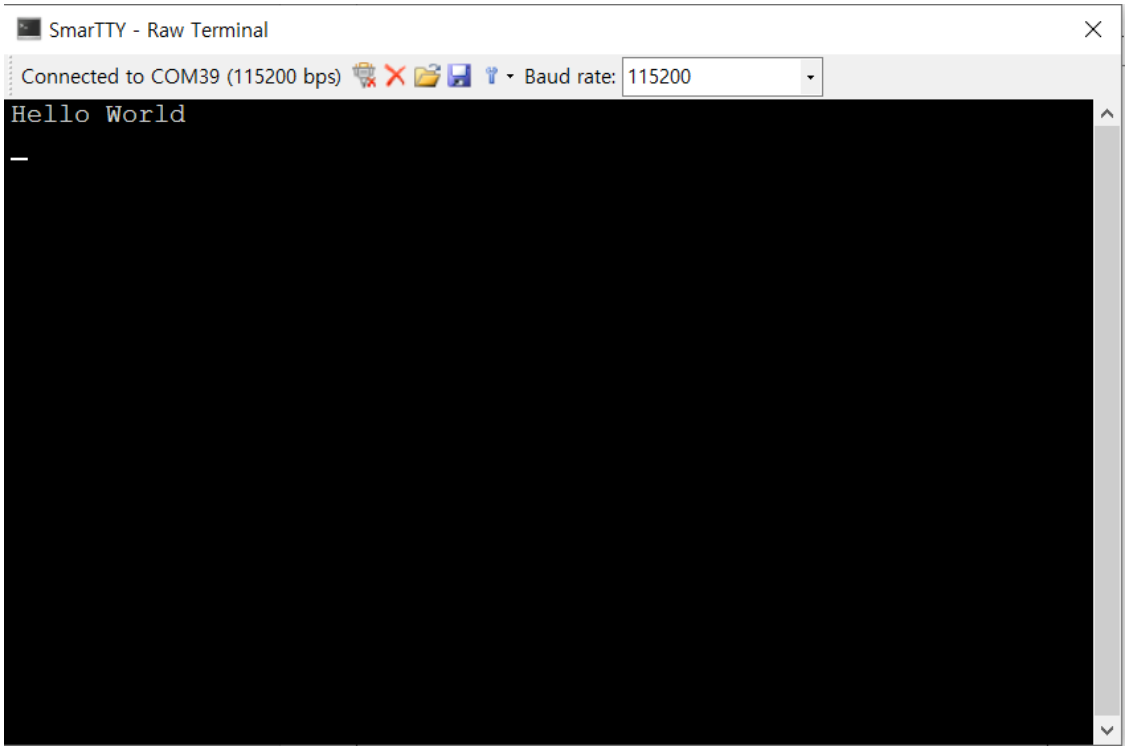
Open Terminal Program: SmarTTY



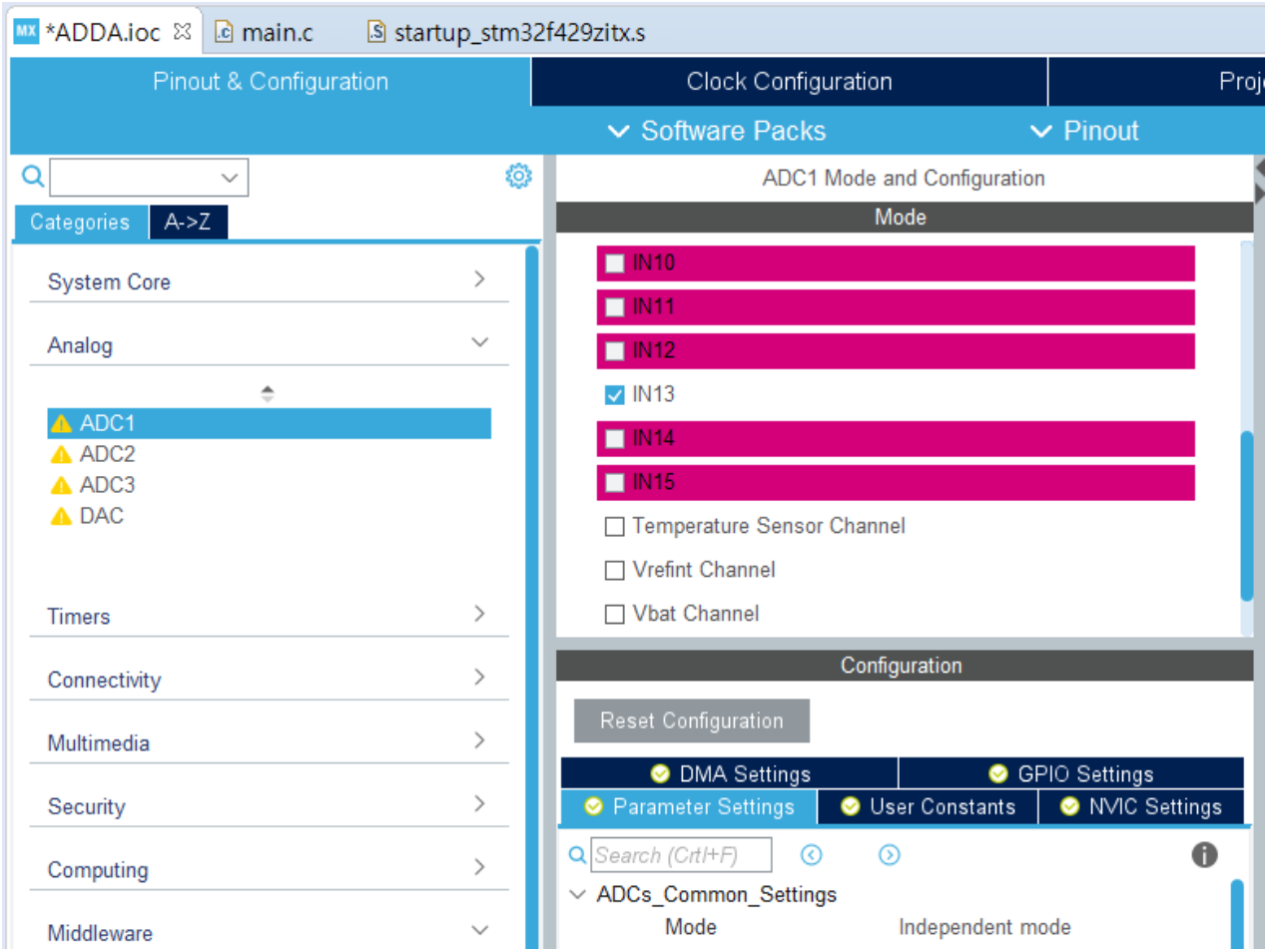
Resume

- Green Arrow: Resume
- Red Square: Stop Debugging

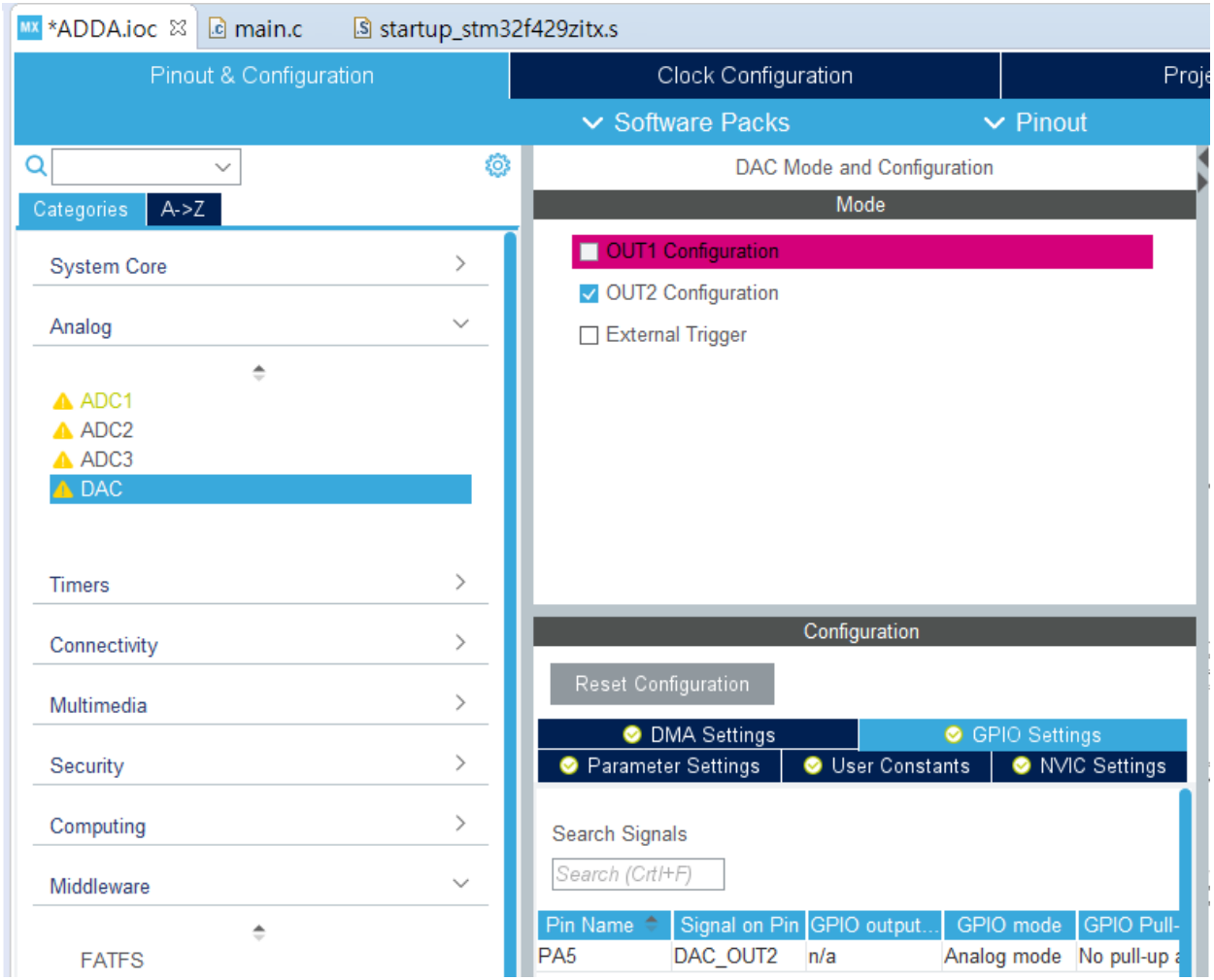




ADC1

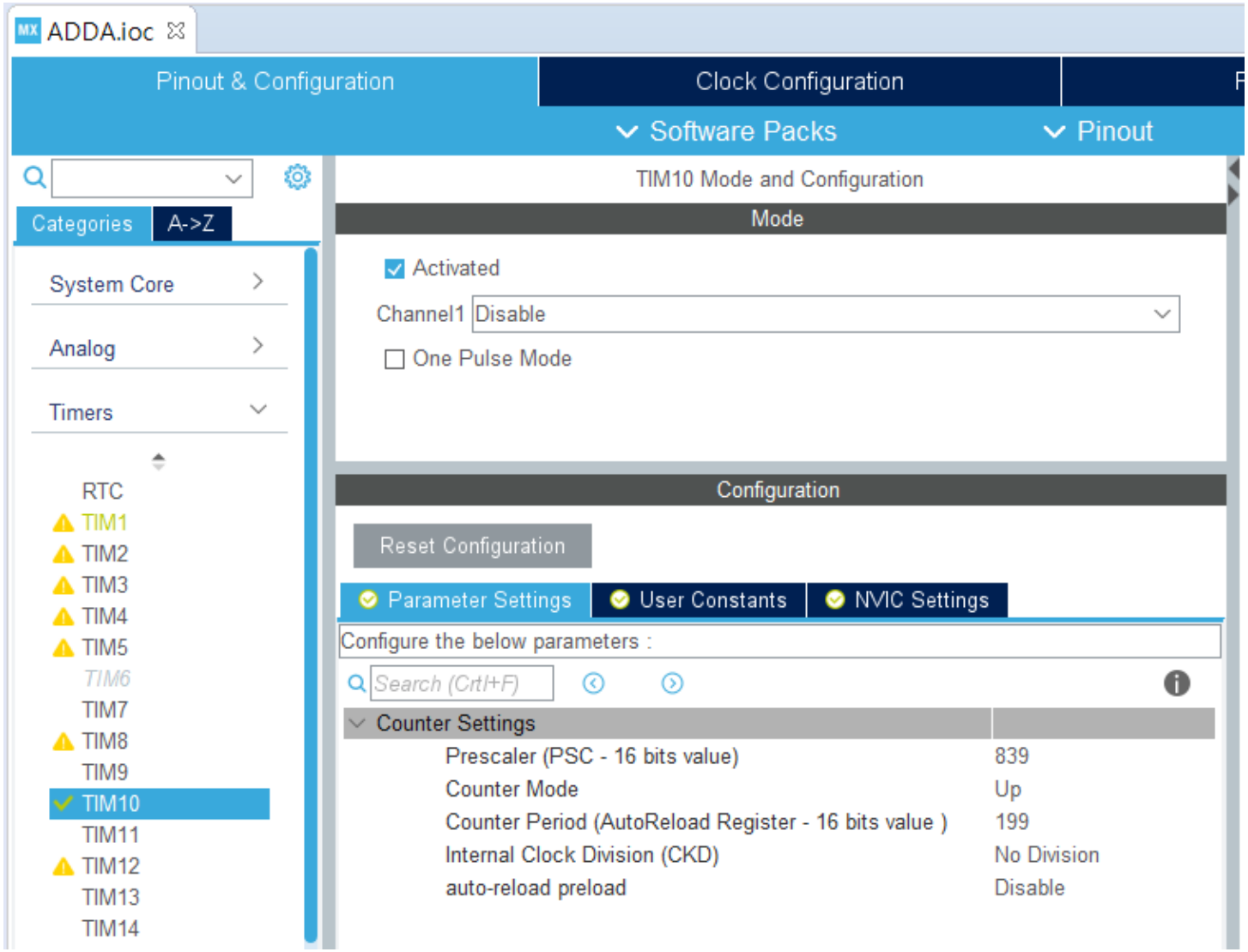

*ADDA.ioc
main.c
startup_stm32f429zitx.s
Pinout & Configuration
Clock Configuration
Software Packs
Pinout
Categories
A->Z
System Core
Analog
ADC1
ADC2
ADC3
DAC
Timers
Connectivity
Multimedia
Security
Computing
Middleware
ADC1 Mode and Configuration
Mode
IN10
IN11
IN12
IN13
IN14
IN15
Temperature Sensor Channel
Vrefint Channel
Vbat Channel
Configuration
Reset Configuration
DMA Settings
GPIO Settings
Parameter Settings
User Constants
NVIC Settings
Search (Crtl+F)
ADCs_Common_Settings
Mode
Independent mode

DAC

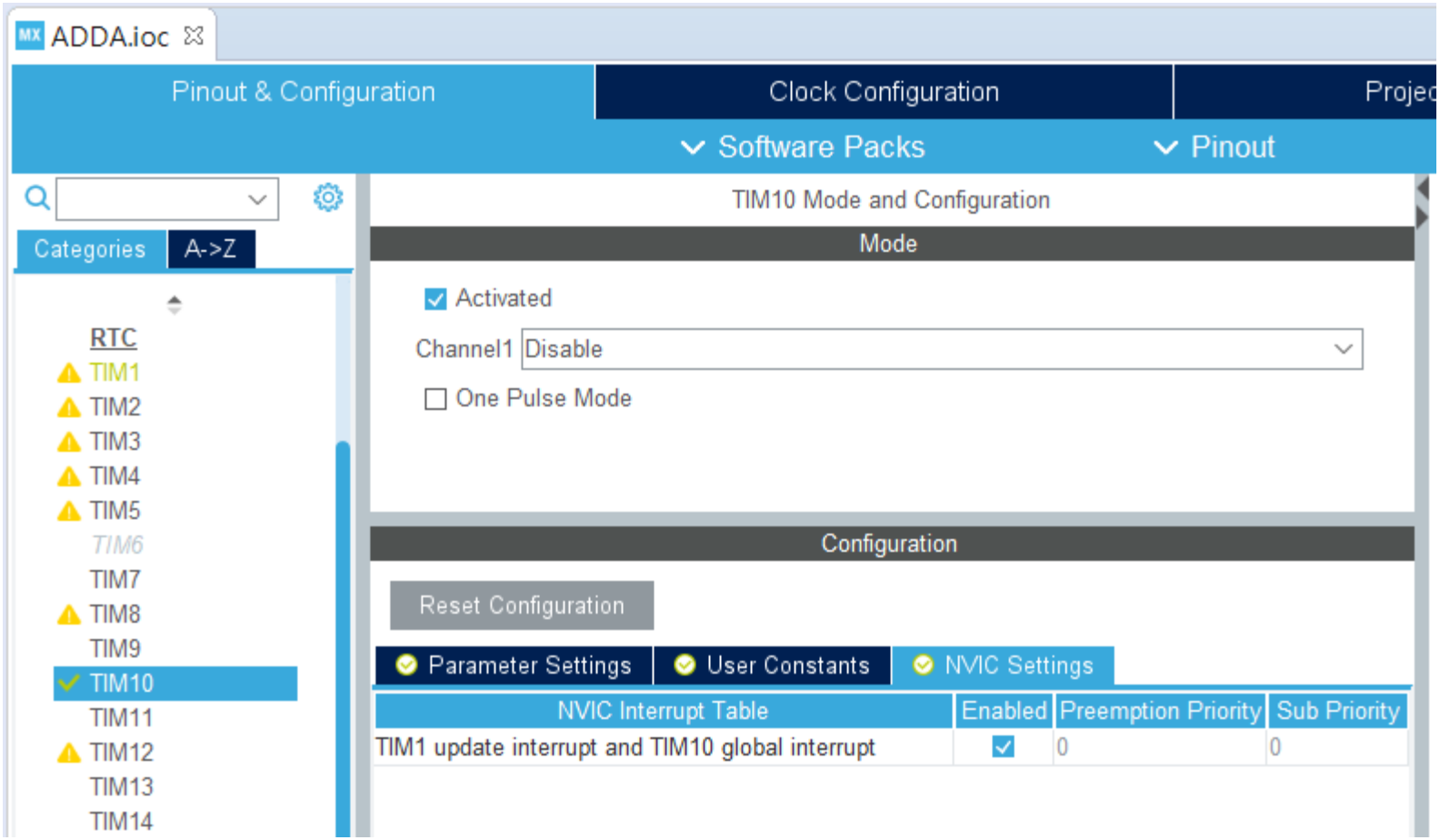

*ADDA.ioc
main.c
startup_stm32f429zitx.s
Pinout & Configuration
Clock Configuration
Software Packs
Pinout
Categories
A->Z
System Core
Analog
ADC1
ADC2
ADC3
DAC
Timers
Connectivity
Multimedia
Security
Computing
Middleware
FATFS
DAC Mode and Configuration
Mode
OUT1 Configuration
OUT2 Configuration
External Trigger
Configuration
Reset Configuration
DMA Settings
GPIO Settings
Parameter Settings
User Constants
NVIC Settings
Search Signals
Search (Crtl+F)
Pin Name
Signal on Pin
GPIO output...
GPIO mode
GPIO Pull-
PA5
DAC_OUT2
n/a
Analog mode

Timer10

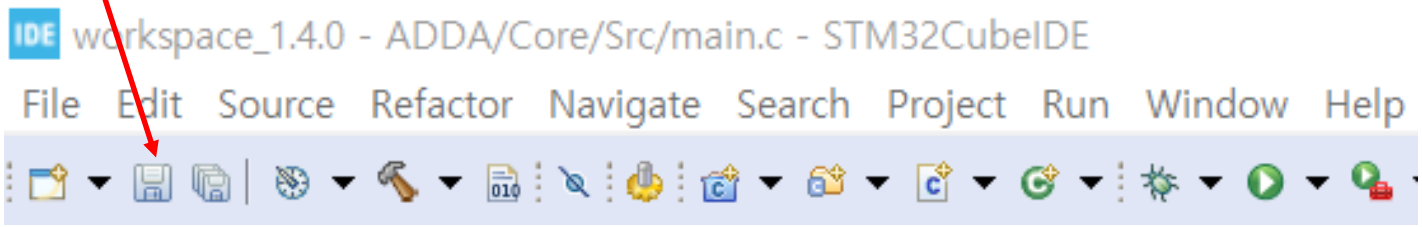
- 168,000,000/(840*200)=1000

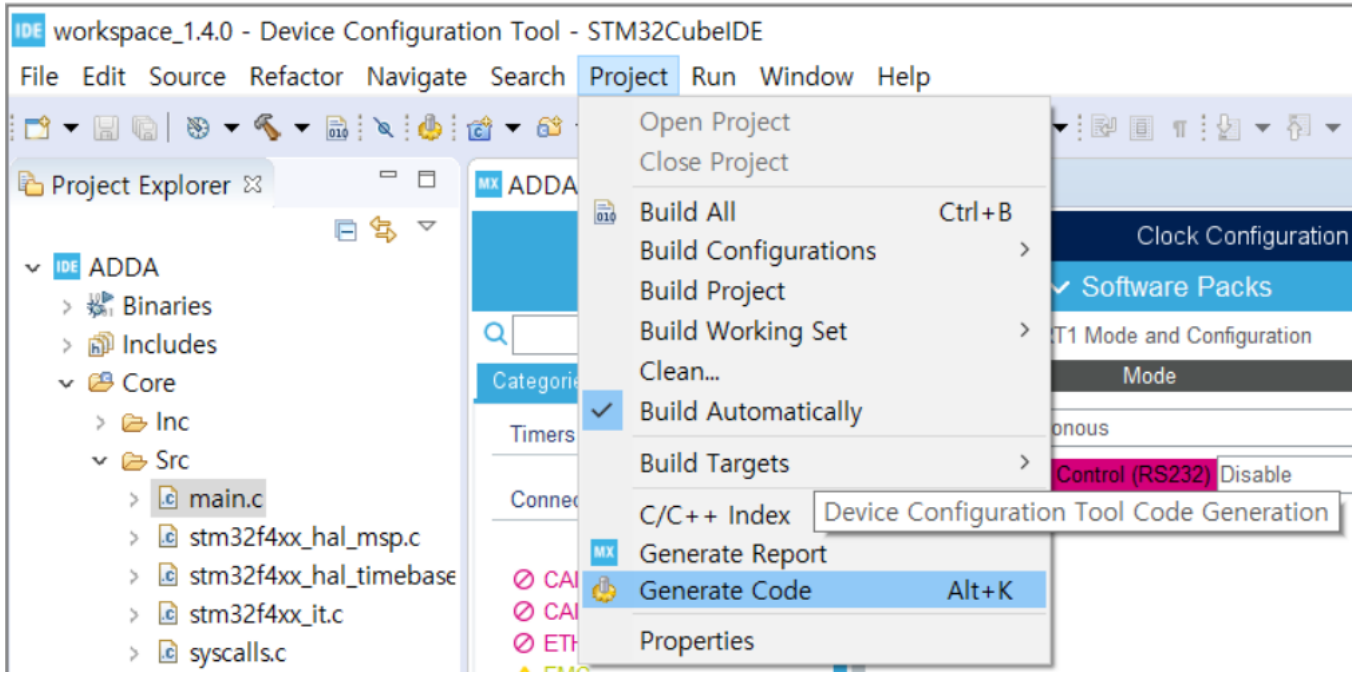


Timer10 Interrupt



Save and Generate Code





A/D D/A Code

```
/* USER CODE BEGIN Includes */
#include "stdio.h"
/* USER CODE END Includes */

/* USER CODE BEGIN 0 */
#ifdef __GNUC__
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#else
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
#endif /* __GNUC__ */
PUTCHAR_PROTOTYPE
{
 HAL_UART_Transmit(&huart1, (uint8_t *)&ch, 1, 0xFFFF);
return ch;
}
/* USER CODE END 0 */

/* USER CODE BEGIN 2 */
printf("Hello World\r\n");
HAL_TIM_Base_Start_IT(&htim10);
HAL_DAC_Start(&hdac, DAC_CHANNEL_2);
/* USER CODE END 2 */
```

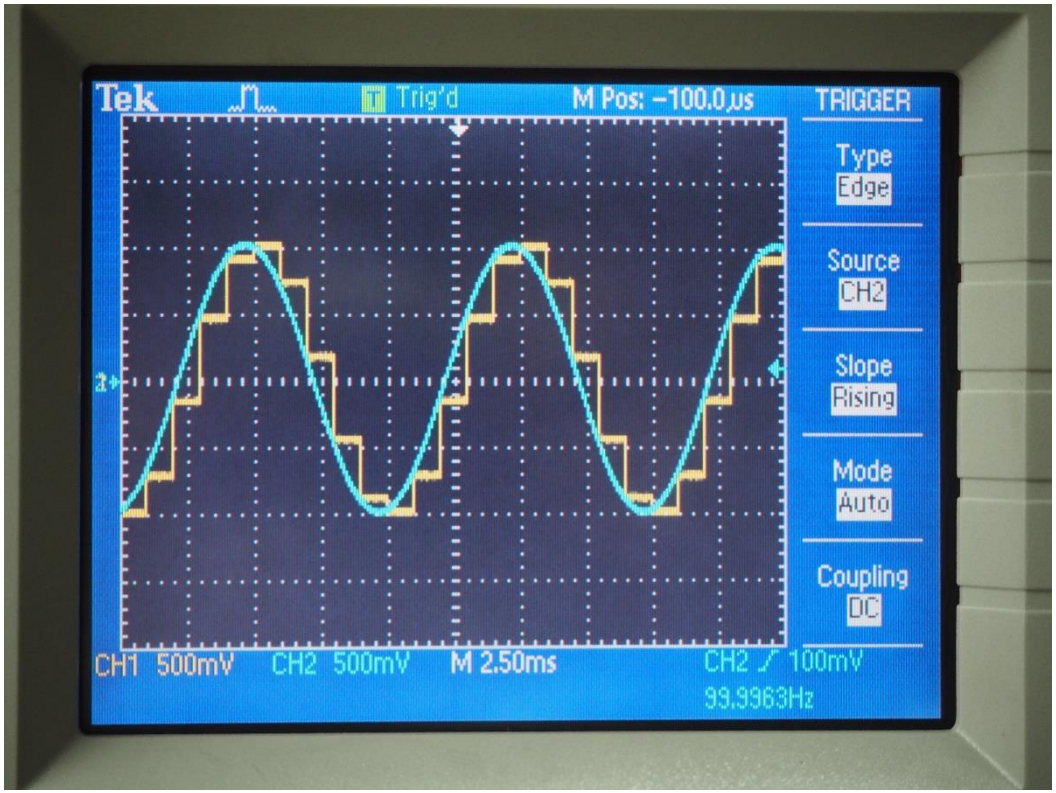
A/D D/A Code

```
/* USER CODE BEGIN Callback 0 */
int da_value,ad_value;
if (htim->Instance == TIM10) {
  HAL_ADC_Start(&hadc1);
  if (HAL_ADC_PollForConversion(&hadc1, 10000) == HAL_OK) {
    ad_value = HAL_ADC_GetValue(&hadc1);
  }
  da_value = ad_value;
  HAL_DAC_SetValue(&hdac, DAC_CHANNEL_2, DAC_ALIGN_12B_R, (uint32_t)(da_value));
}
/* USER CODE END Callback 0 */
```

A/D Sampling

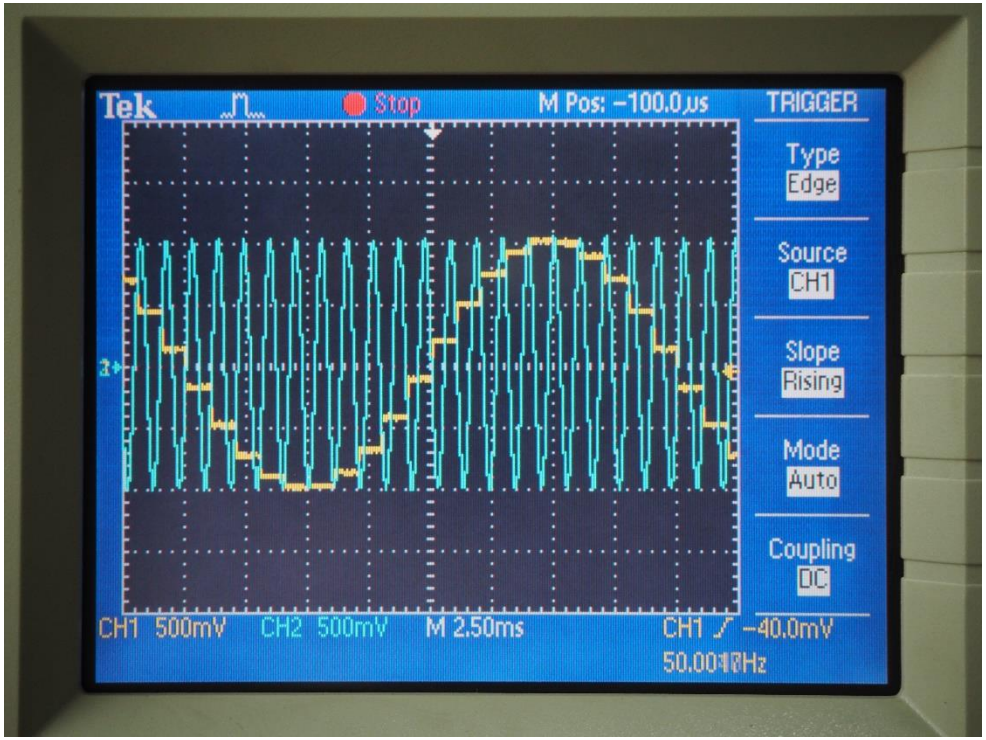
- Build and Run

- Function Generator
  - Sine wave
  - 2V peak-to-peak
  - 100 Hz



Aliasing

- Change function generator frequency to:
  - 1KHz=1000Hz
  - 1020Hz, 1030Hz, 1040Hz, 1050Hz
  - 980Hz, 970,Hz,960Hz,950Hz
- Find the frequencies of the DA output signal (Trigger source DA output signal)



Aliasing

- Change function generator frequency to:
  - 2KHz=2000Hz
  - 2020Hz, 2030Hz, 2040Hz, 2050Hz
  - 1980Hz, 1970,Hz,1960Hz,1950Hz
- Find the frequencies of the DA output signal (Trigger source DA output signal)

Nyquist Shannon Sampling Theorem

- Anti-aliasing filter