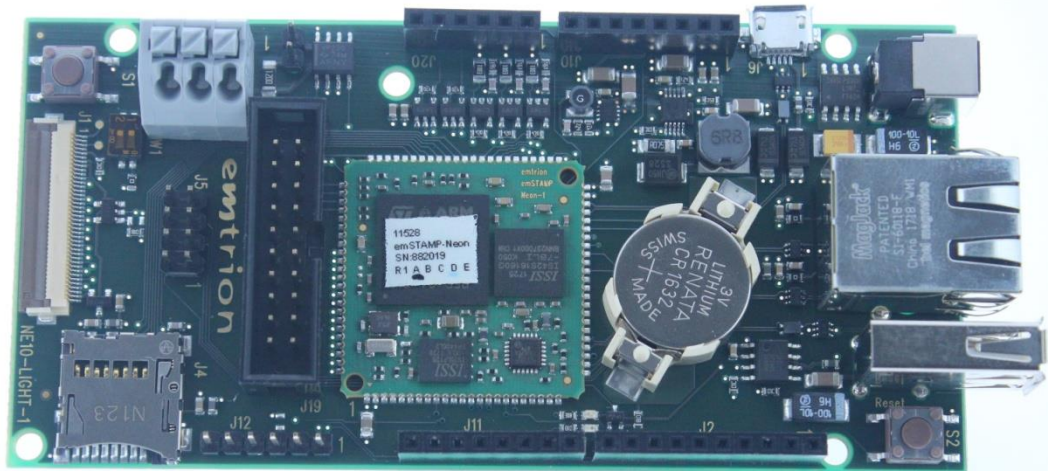


# emSTAMP Neon Developer Kit

## Software Manual

Rev04 / 13.05.2020



© Copyright 2020 **emtrion GmbH**

All rights reserved. This documentation may not be photocopied or recorded on any electronic media without written approval. The information contained in this documentation is subject to change without prior notice. We assume no liability for erroneous information or its consequences. Trademarks used from other companies refer exclusively to the products of those companies.

Revision: **04 / 13.05.2020**

Rev	Date/Signature	Changes
<b>1</b>	01.08.2018/Ha	Initial release
<b>2</b>	27.09.2018/Ha	Minor change, fix typo
<b>3</b>	23.03.2020/Ha	Chang for CM479 + STM32CubeIDE
<b>4</b>	13.05.2020/Ha	Add Link to the IOC file

## 1 Table of contents

Revision: <b>04 / 13.05.2020</b> .....	2
2 Introduction.....	4
3 Hardware requirement .....	4
3.1 JTAG debugger/programmer .....	4
4 Workstation Software installation .....	5
4.1 STM32CubeMX initialization code generator.....	5
4.2 STM32 ST-LINK utility.....	5
4.3 STM32CubeIDE for development.....	5
4.4 The IOC file .....	5
5 Testing your in-circuit Debugger/programmer on your emStamp-Neon.....	6
6 Working with STM32CubeIDE to develop your application.....	7

## 2 Introduction

Welcome to emSTAMP-Neon documentation. This manual will give you a startup software guideline of our developer kit. It will describe how to use the different free software to program your developer kit.

It is assumed that users of emtrion developer kits are already familiar with software development. Programming knowledge are out of the scope of this document. emtrion will gladly assist you in acquiring this knowledge. If you are interested in training courses or getting support, please contact the emtrion sales department.

The examples in this manual are demonstrated on specific hardware but if not mentioned otherwise they all work on all supported emtrion devices.

Please refer to the "Hardware Description" of to emSTAMP-Neon available on the emtrion support website (<http://support.emtrion.de>) for more detailed info of the capability of the product.

## 3 Hardware requirement

### 3.1 JTAG debugger/programmer

In this manual, the in-circuit debugger/programmer used is the ST-LINK/V2 (<https://www.st.com/en/development-tools/st-link-v2.html>).

## 4 Workstation Software installation

Before starting you need to prepare your workstation. In order to build an image that runs on the target, you need to install the following set of free software available online on the ST microelectronics, the microcontroller manufacturer ([www.st.com](http://www.st.com)).

### 4.1 STM32CubeMX initialization code generator

STM32CubeMX is a graphical tool that allows a very easy configuration of STM32 microcontrollers and the generation of the corresponding initialization C code through a step-by-step process. (<https://www.st.com/en/development-tools/stm32cubemx.html> )

Note: STM32CubeMX is a stand alone application but it is also part of the STM32CubeIDE (see below)

### 4.2 STM32 ST-LINK utility

STM32 ST-LINK Utility is a full-featured software interface for programming STM32 microcontrollers (<https://www.st.com/en/development-tools/stsw-link004.html>)

The tool offers a wide range of features to program STM32 internal memories (Flash, RAM, OTP and others), external memories, to verify the programming content (checksum, verify during and after programming, compare with file) and to automate STM32 programming. STM32 ST-LINK Utility is delivered as a graphical user interface (GUI) with a command line interface (CLI).

### 4.3 STM32CubeIDE for development

STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors. It is based on the ECLIPSE™/CDT framework and GCC toolchain for the development, and GDB for the debugging. (<https://www.st.com/en/development-tools/stm32cubeide.html> )

### 4.4 The IOC file

emtrion is providing a ioc file that is compliant with the developer kit and let you generate the MCU peripheral (GPIO, USART, Pin MUX ...) and the middleware (USB, TCP/IP, FreeRTOS...). The output will be a bundle of initialization file in C code and all project files that allow you to create your main project.

Link to download the IOC File:

<https://cloud.emtrion.de/index.php/s/EONS1xzUWNYVXON/download>

## 5 Testing your in-circuit Debugger/programmer on your emStamp-Neon

It is strongly advice to test the communication between your workstation and your target using the ST-LINK utility tool before starting any debugging and/or programming with SW4STM32.

The output of a good communication should look like this:

The screenshot shows the STM32 ST-LINK Utility window. The top menu includes File, Edit, View, Target, ST-LINK, External Loader, and Help. Below the menu is a toolbar with icons for file operations and connection. The main area is divided into several sections:

- Memory display:** Address: 0x90000000, Size: 0x2000, Data Width: 16 bits.
- Device Information:**
  - Device: STM32F469x/F479x
  - Device ID: 0x434
  - Revision ID: Rev A
  - Flash size: 2MBytes
- Target memory, Address range: [0x90000000 0x90002000]**
- Memory Table:**

Address	0	2	4	6	8	A	C	E	ASCII
0x90000000	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000010	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000020	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000030	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000040	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000050	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000060	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000070	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000080	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x90000090	0000	0000	0000	0000	0000	0000	0000	0000	.....
- Log Output:**

```

16:38:37 : ST-LINK SN : 31FF6F064D4E353849390943
16:38:37 : ST-LINK Firmware version : V2J29S7
16:38:37 : Connected via JTAG.
16:38:37 : JTAG Frequency = 9 MHz.
16:38:37 : Connection mode : Connect Under Reset.
16:38:37 : Debug in Low Power mode enabled.
16:38:38 : Device ID:0x434
16:38:38 : Device flash Size : 2MBytes
16:38:38 : Device family :STM32F469x/F479x

```
- Status Bar:**
  - Debug in Low Power mode enabled.
  - Device ID:0x434
  - Core State : Live Update Disabled

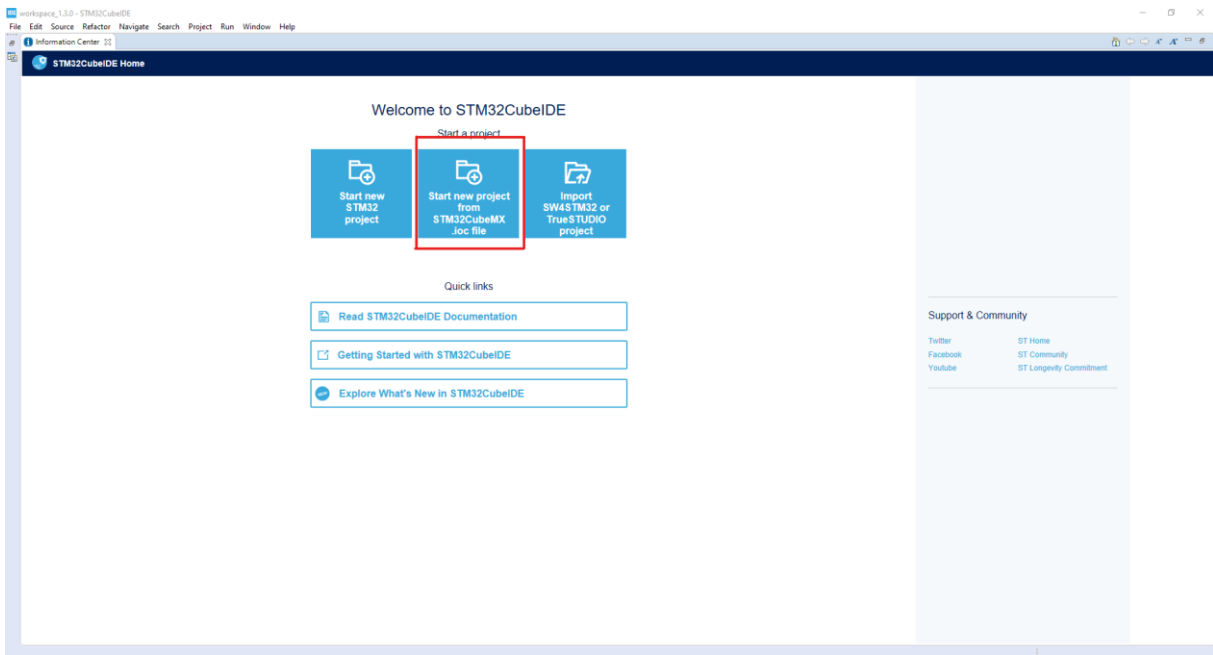
## 6 Working with STM32CubeIDE to develop your application

Emtrion is providing the configuration file that gives you the possibility to load the entire pin muxing, clock configuration and middleware of the MCU STM32F479NIHx used in the emStamp-Neon. To proceed to the generation, load the project file called *emSBC-Neon-2-CM479-Pinmux-V08-Release-emSTAMP-Neon-edi5937-v2en.ioc*

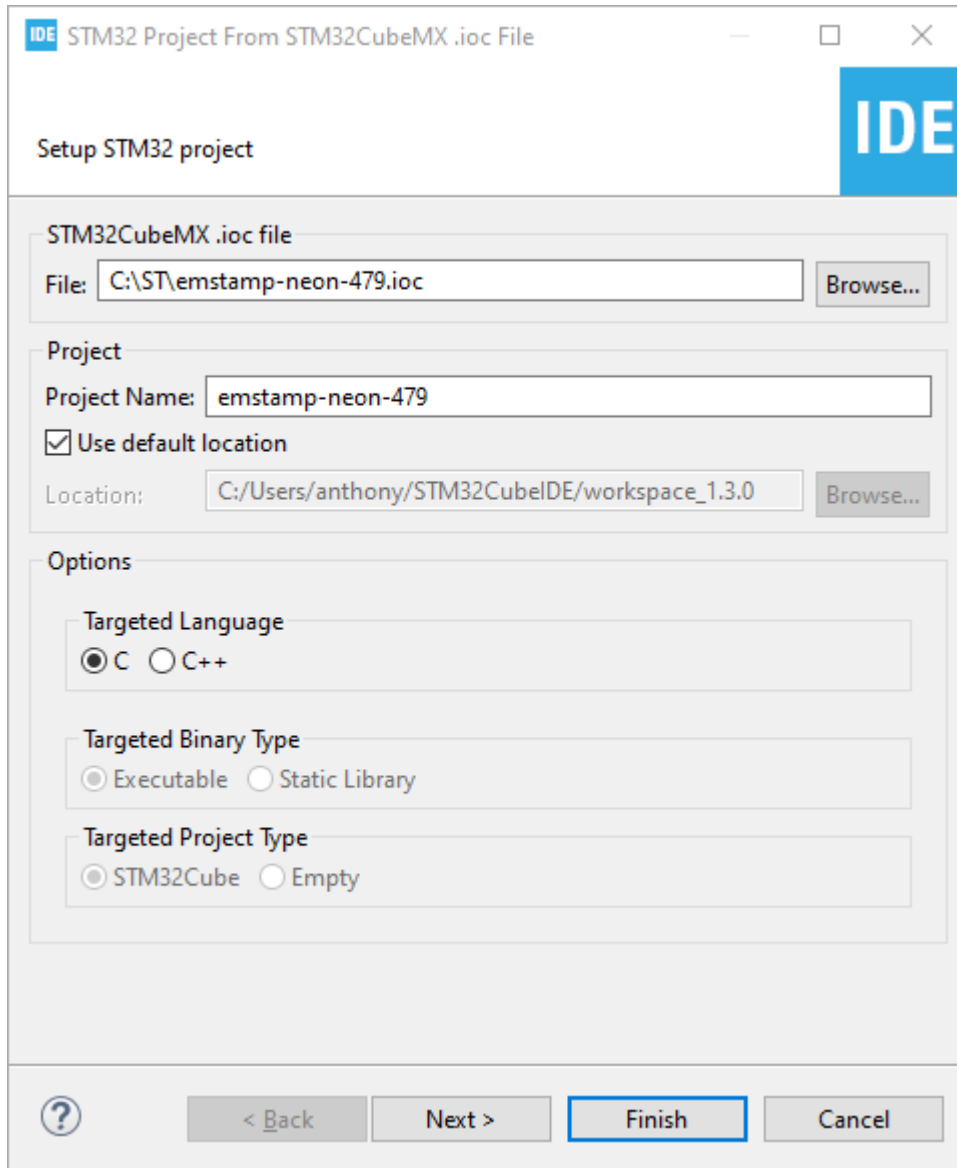
To import a project, open STM32CubeIDE:



Then click on the following icon:

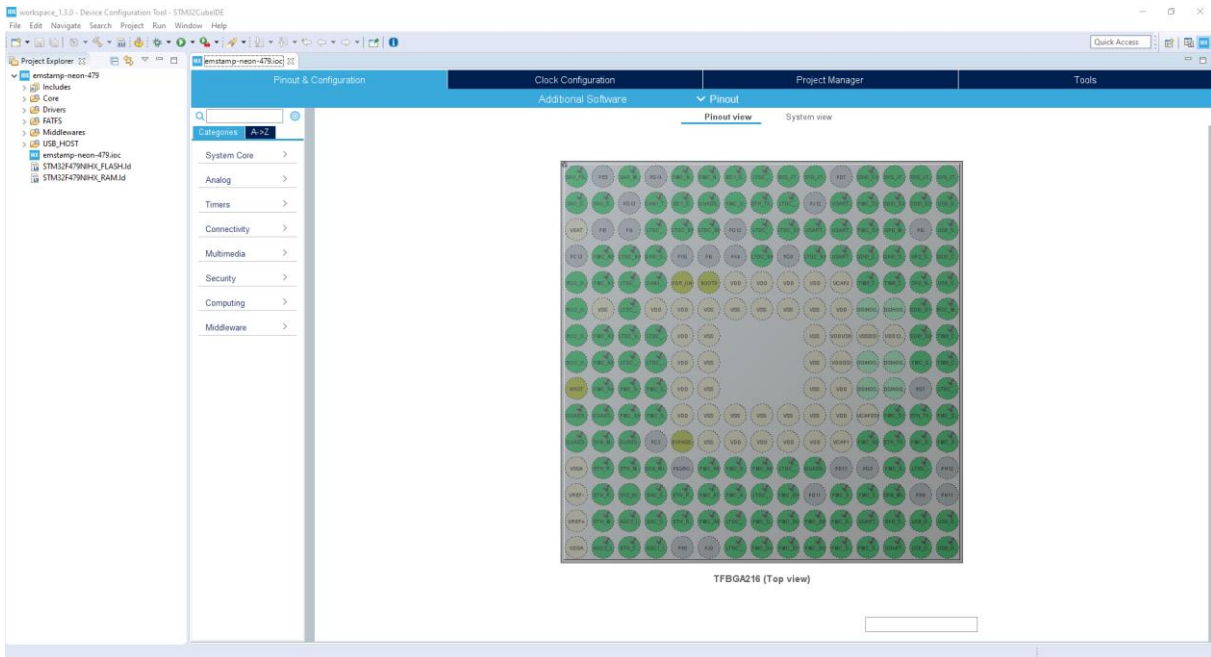


And select the provided ioc file. Then click on finish and wait for the files generations to finish.



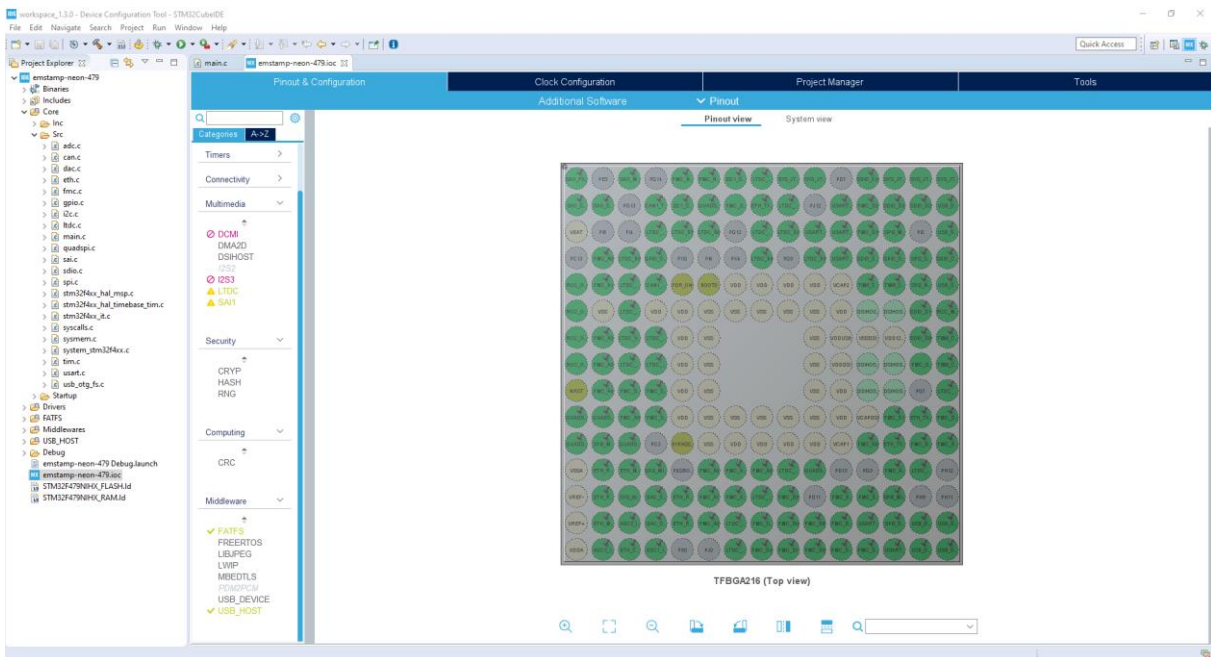


Once imported you should see the IDE like this:

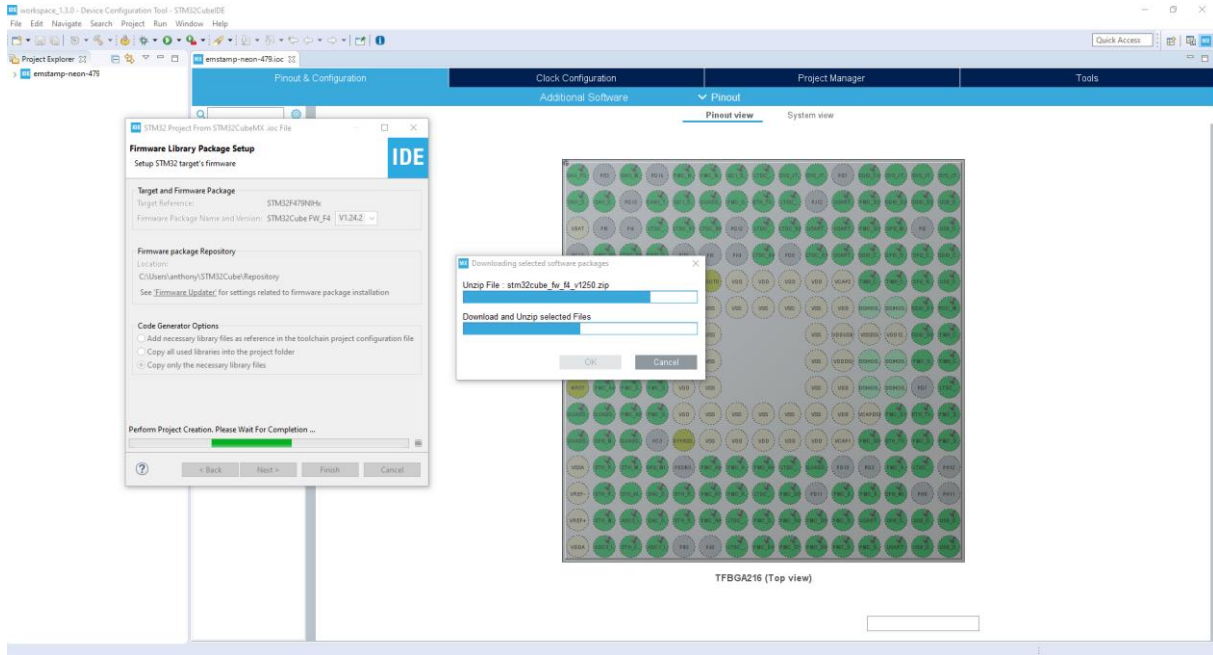


From here you can modify the source code and/or the configuration of the microcontroller. Using the integrated STM32CubeMX, you can modify and regenerate the configuration like this:

Double click on the ioc file:

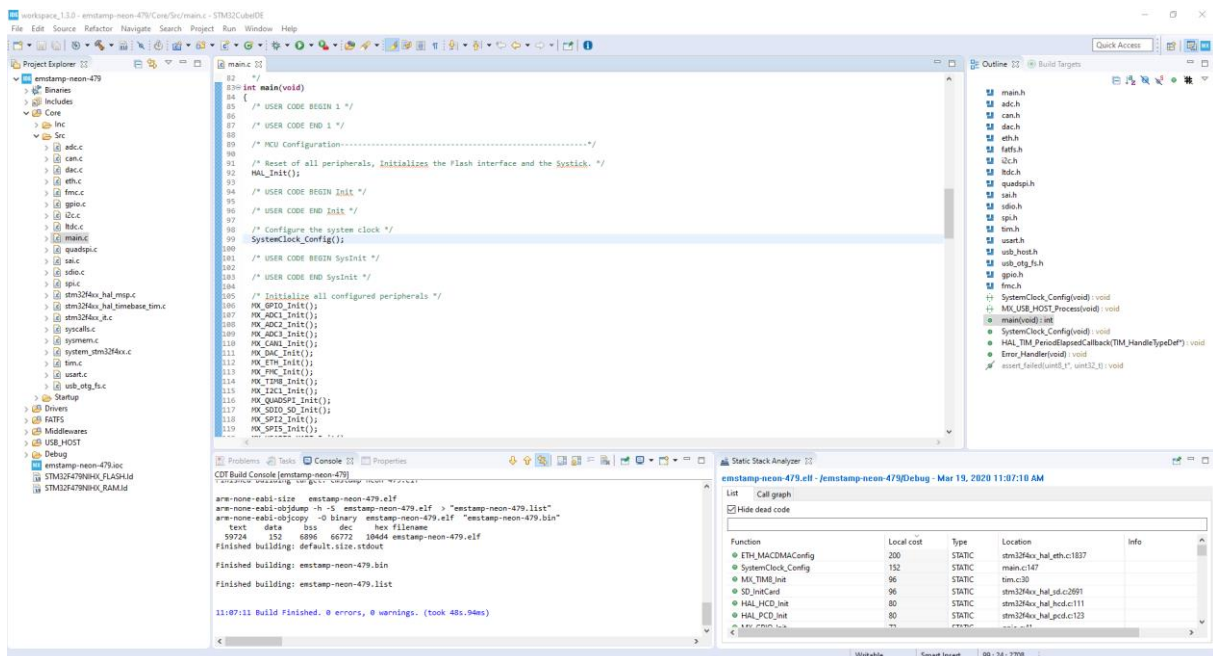


Then modify and regenerate:



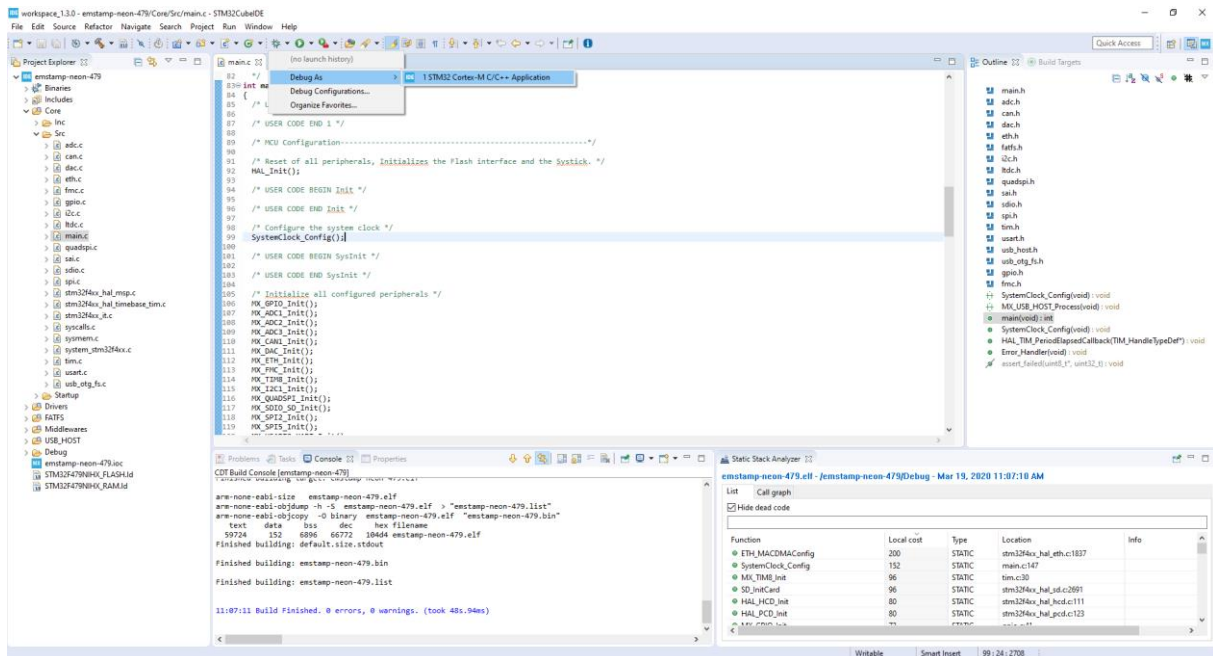
Note: This might take a long time if you haven't already downloaded the firmware package. The IDE do it itself.

Let's compile the project by click on the hammer icon, select Debug:

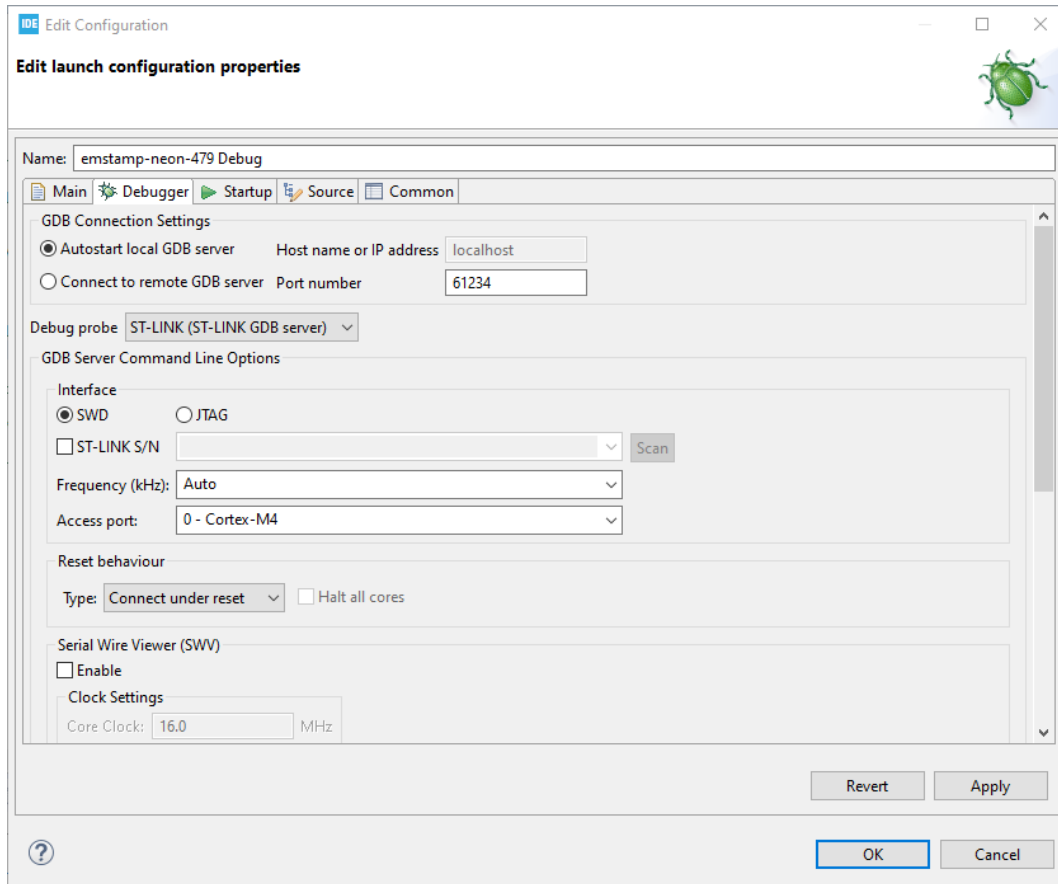


You are now ready to debug your application.

Make sure your target is power up and that the ST-LINK is connected properly and hit the bug icon to debug your project.



Choose the following configuration:



Once done you should get the IDE ready to debug the code on the target like this:

