



POLITECNICO
DI MILANO

www.polimi.it



STM32 - First Trial

Dr. Federica Villa



Introduction

Objective of this first trial is to learn:

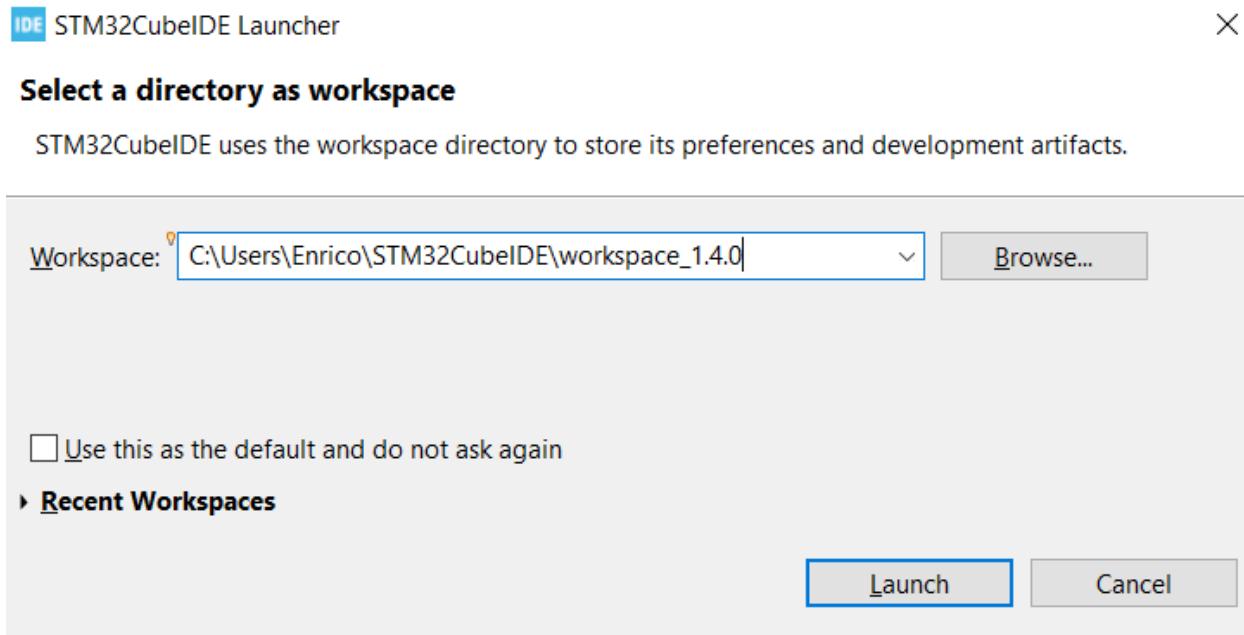
- how to create and configure a new project
- which is the structure of the Cube-generated code
- how to compile and debug a code

Last but not least it is useful to check if all the software and drivers have been correctly installed



New project with Cube

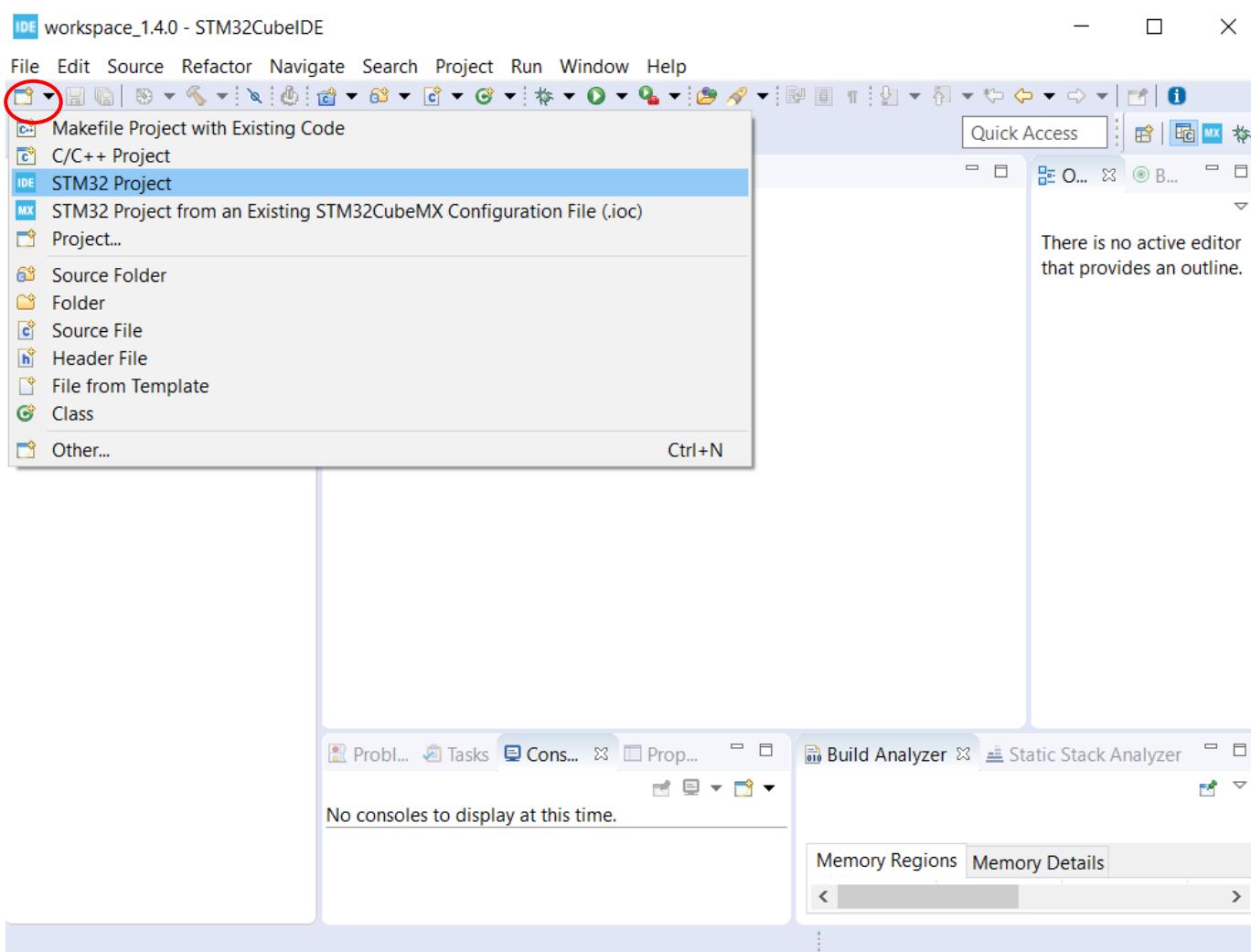
- Open **STM32CubeIDE**
- Select a folder for the workspace and click “Lauch”
- Accept to download any eventual new library / update
(the first time you use the SW it might prompt you to download libraries and updates, always accept and wait, they can take some minutes)





New project with Cube

- Click on the downwards arrow next to “New” → STM32 Project [Setup from existing STM32CubeMX projects can be imported by choosing “STM32 Project from an Existing STM32CubeMX Configuration file (.ioc)’’]





Board Selection

NUCLEO-F401RE

IDE STM32 Project

Target Selection

⚠ STM32 target or STM32Cube example selection is required

MCU/MPU Selector Board Selector Example Selector Cross Selector

Board Filters

Commercial Part Number F401RE NUCLEO-F401RE

Vendor

Type

MCU/MPU Series

Other

Peripheral

Features Large Picture Docs & Resources Datasheet Export

Boards List: 1 item

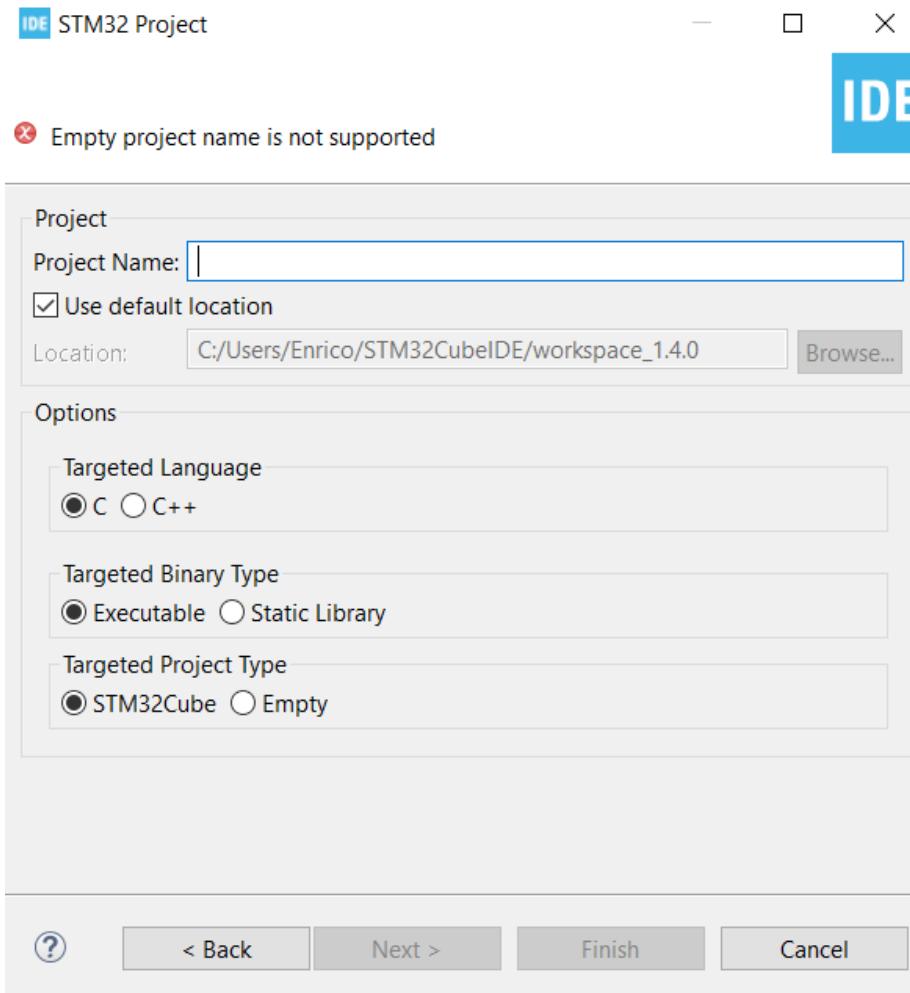
	Overview	Com	Type	Marketing Stat...	Unit Price (US\$)	Mounted Device
*		NUCLEO-F401RE Nucleo-64	Active	13.0	STM32F401RETx	

< Back Next > Finish Cancel

- Select **NUCLEO-F401RE**
- If you need, download **datasheet** and other documentation
- Click “**Next**”



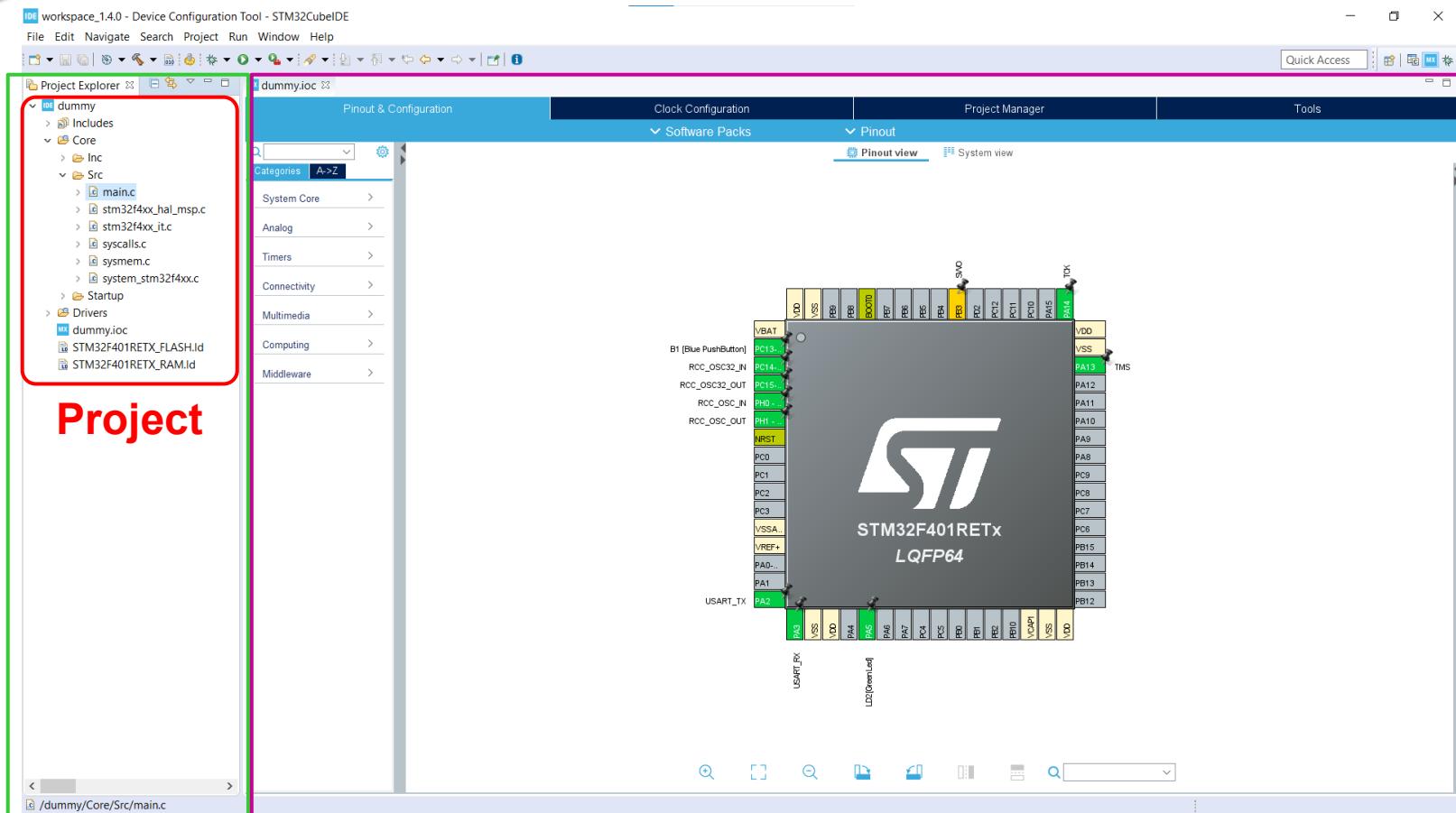
Project generation



- Choose a **Project Name**
- Click "**Finish**"
- Click "**Yes**" on the prompt "**Initialize all peripherals with their default mode?**"
- Let the code generator run



STM32CubeIDE interface



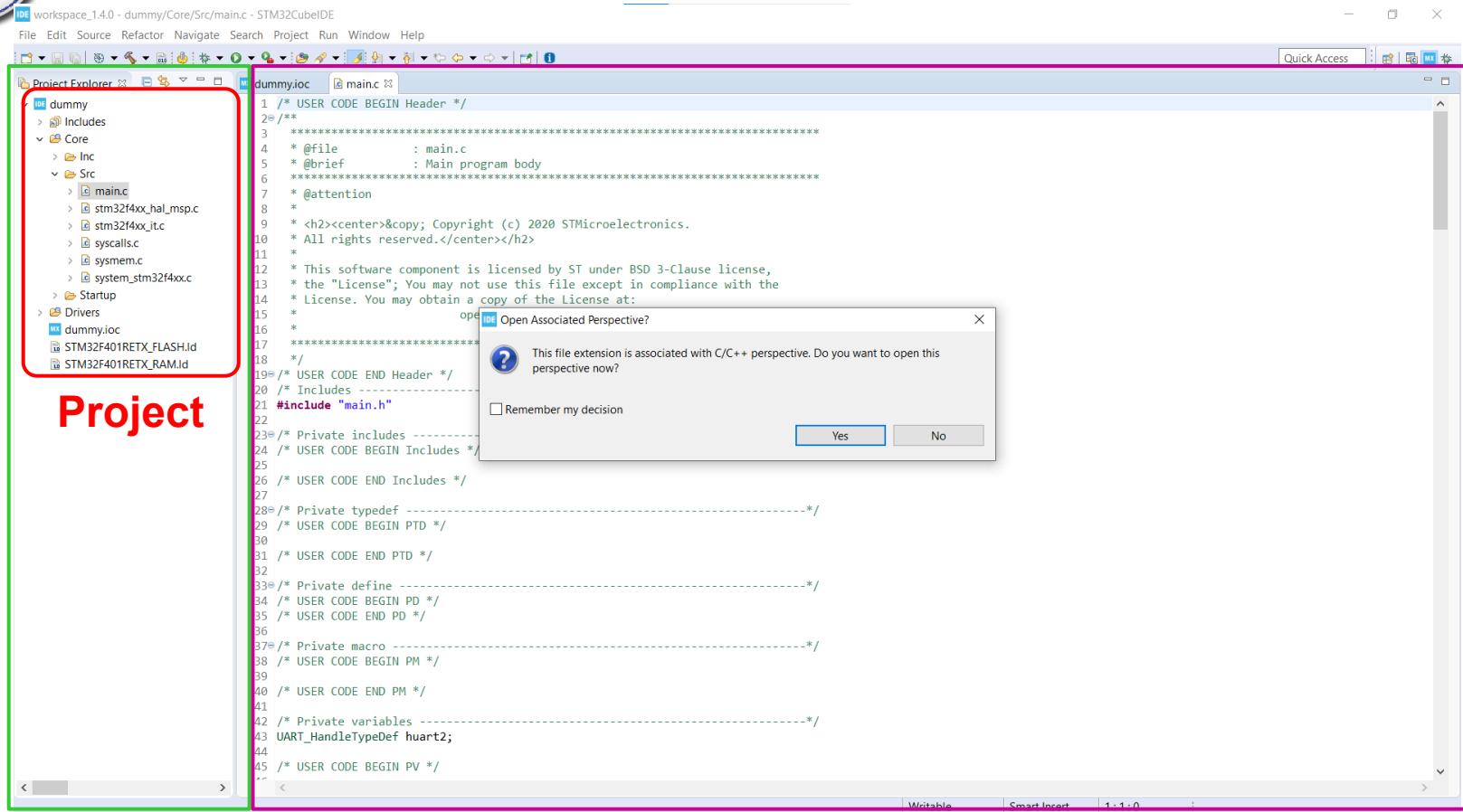
Workspace

- Double click “main.c”

Device configurator (STM32CubeMX)



STM32CubeIDE interface



Workspace

The IDE layout dynamically changes depending on the action being performed (coding, debugging, etc...)
Answer "Yes" when prompted to change perspective



Project code

The screenshot shows the STM32CubeIDE interface. The left sidebar displays the project structure under 'dummy'. The main area shows the 'main.c' file with the following code:

```
1 /* USER CODE BEGIN Header */
2 /**
3  * @file      : main.c
4  * @brief     : Main program body
5  * @attention
6  *
7  * <h2><center>&copy; Copyright (c) 2020 STMicroelectronics
8  * All rights reserved.</center></h2>
9  *
10 * This software component is licensed by ST under BSD 3
11 * the "License"; You may not use this file except in co
12 * License. You may obtain a copy of the License at:
13 *          opensource.org/licenses/BSD-3-
14 *
15 *          opensource.org/licenses/BSD-3-
16 *
17 *          opensource.org/licenses/BSD-3-
18 */
19 /* USER CODE END Header */
20 /* Includes -
```

The right sidebar shows the generated files: main.h, huart2, SystemClock_Config, MX_GPIO_Init, MX_USART2_UART_Init, main, SystemClock_Config, MX_USART2_UART_Init, MX_GPIO_Init, and Error_Handler.

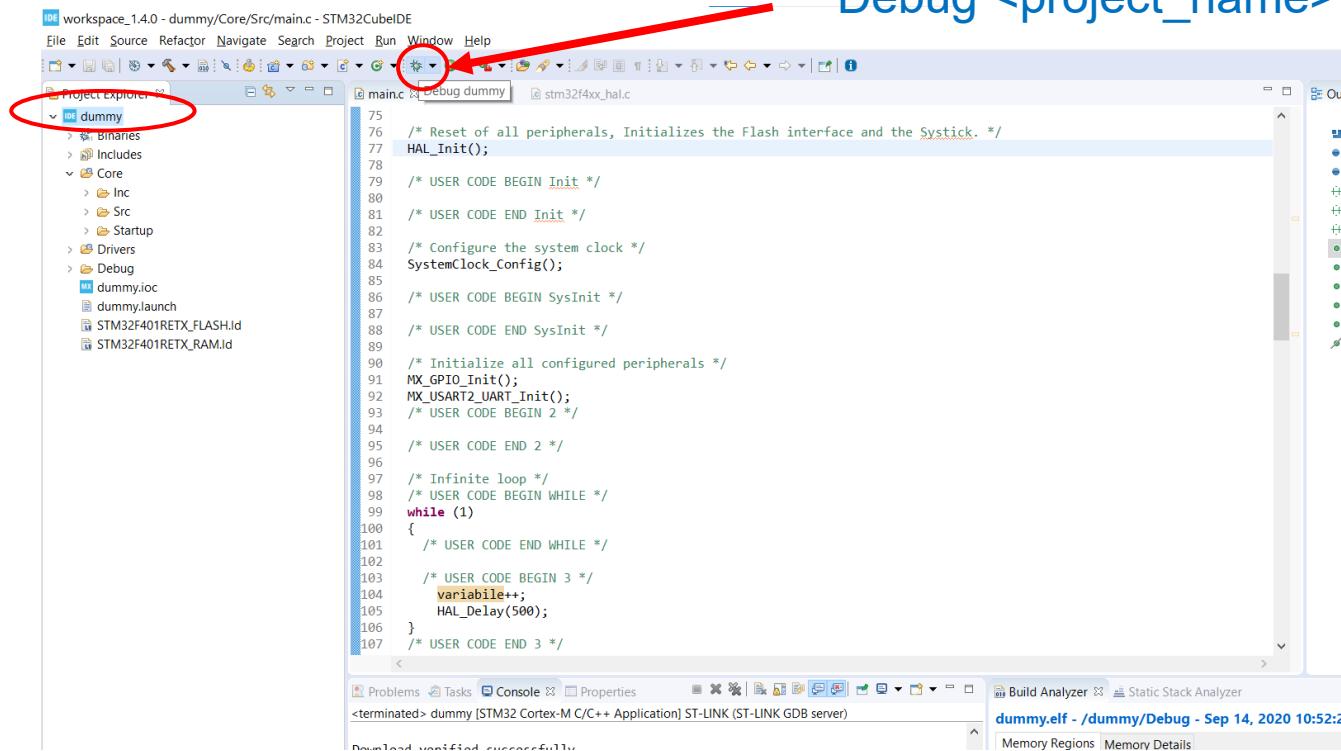
The bottom console window shows the build results:

```
CDT Build Console [dummy]
09:58:50 Build Finished. 0 errors, 0 warn:
```

- Some parts of the code are already written (based on user settings selected with Cube)
- Between **/*USER CODE BEGIN** and **/*USER CODE END** it is possible to add code, which will be not modified even in case the settings in Cube are modified and code is generated again.
- Click **Build All**
- Check in the **Console** if the code has been built with **"0 Error(s)"**

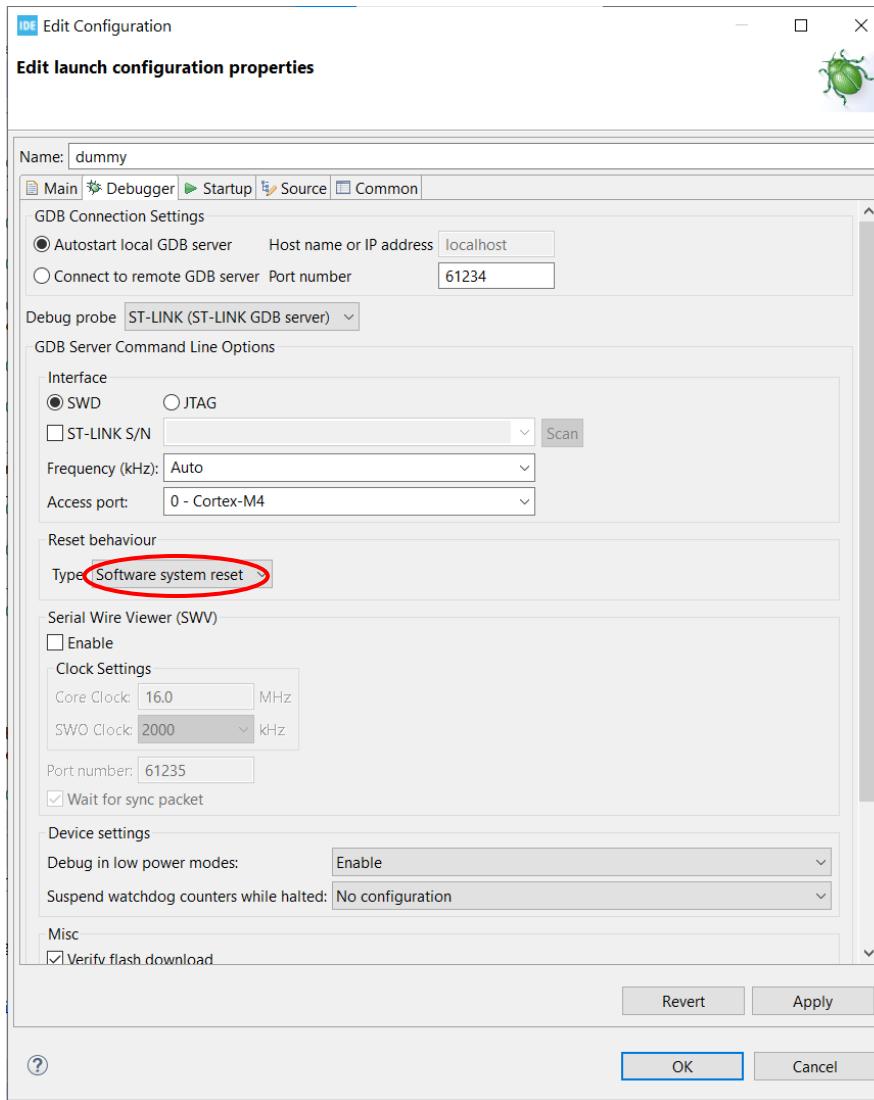
Start debug session

Debug <project_name>



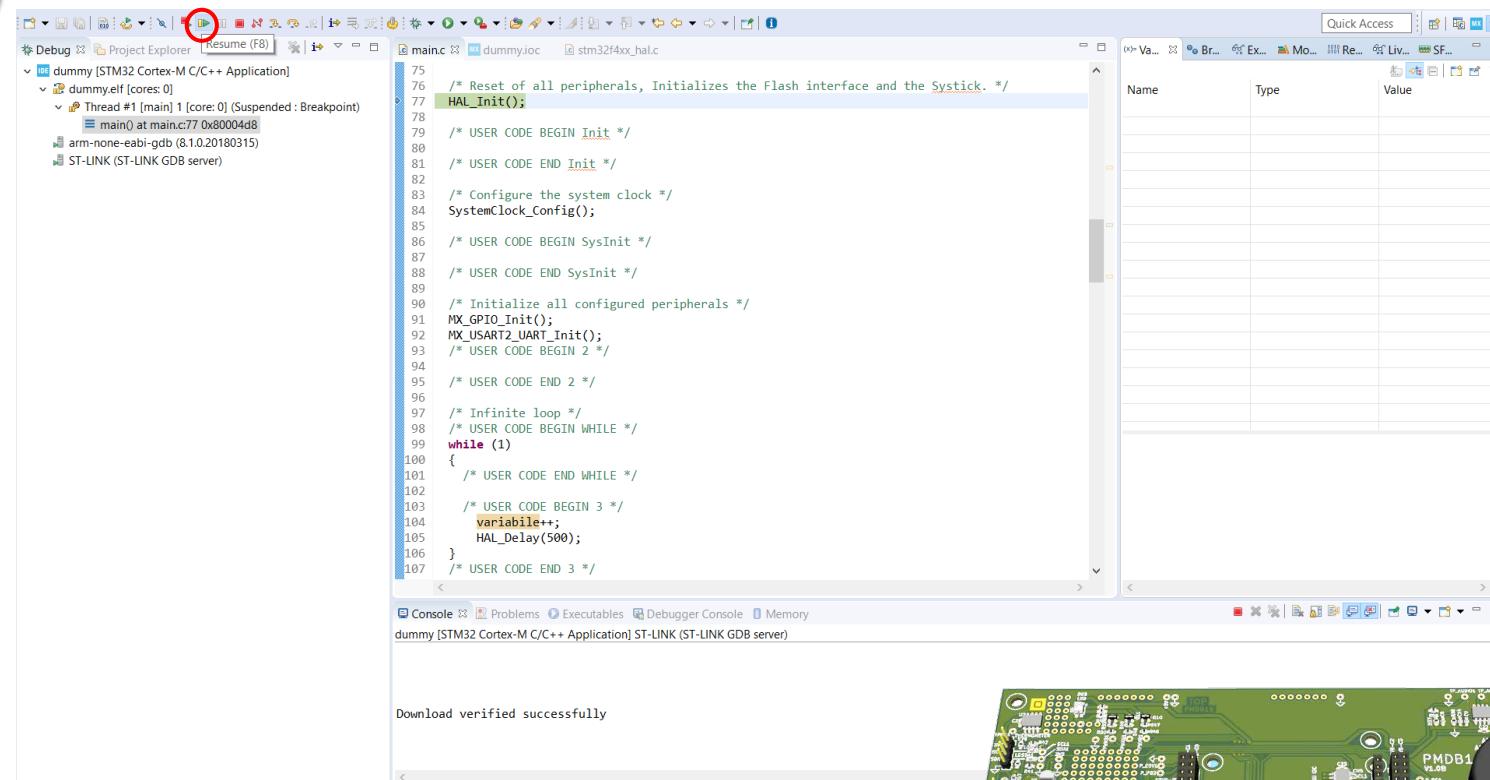
- Connect the Nucleo board to the PC through the USB link
- Select the project name on the left panel
- Click on “Debug <project_name>”
- If prompted, update the ST-Link firmware, wait for completion, reconnect the USB port and re-launch the debug session

Debug session



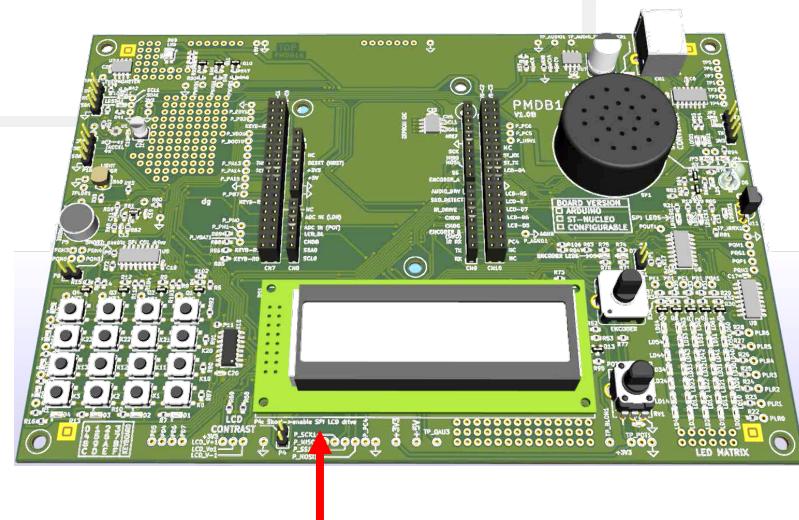
- Configure the debugger as shown on the left
- Most configuration are at their default value, except “Reset behaviour”

Debug software execution



Download verified successfully

- Start code execution by pressing “Resume” [F8]
- Global variables can be monitored in real time by adding them to the “Live expression” pane on the right



Red LED



End

Congratulations!

Your tools have been
correctly installed!



STM32CubeMX + Keil workflow



Introduction

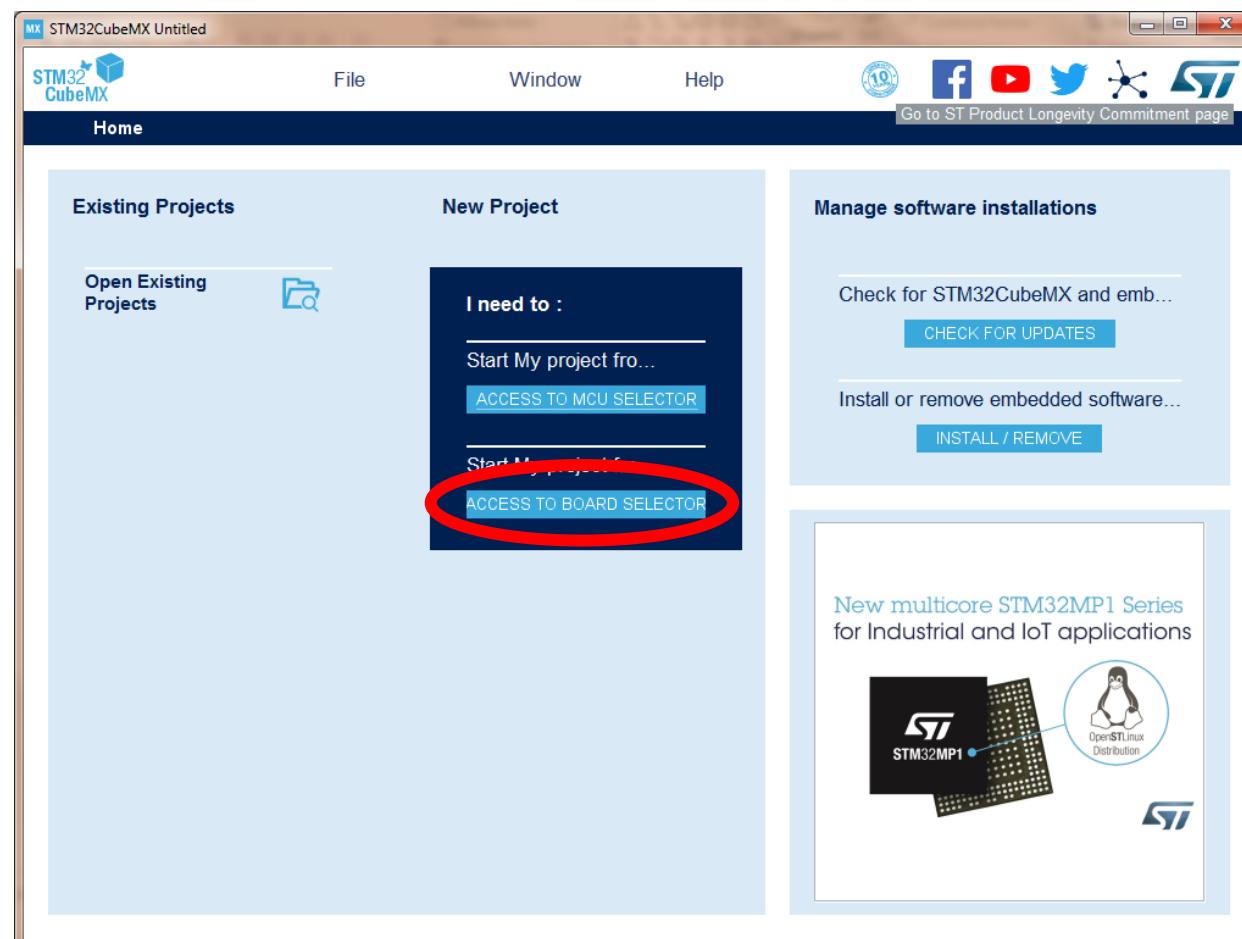
Objective of this first trial is to learn:

- how to create a new project with Cube
- how to generate a code for Keil
- which is the structure of the Cube-generated Keil code
- how to compile and debug a code

Last but not least it is useful to check if all the software and drivers have been correctly installed



New project with Cube



- Open **STM32CubeMX**
- **New Project**, select **“ACCESS TO BOARD SELECTOR”**
- Accept to download any eventual new library / update
(the first time you use Cube it will prompt you to download libraries and updates, always accept and wait, they can take some minutes)



Board Selection

NUCLEO-F401RE

The screenshot shows the 'MCU Selector' interface. A red arrow points from the 'Board Selector' tab at the top left to the search bar where 'NUCLEO-F401RE' is typed. Another red circle highlights the 'Start Project' button at the top right of the main content area. The central area displays information about the STM32MP1 series, featuring an STM32MP1 chip and a Linux penguin icon representing the OpenSTLinux distribution. Below this is a 'Boards List' section showing one item: NUCLEO-F401RE.

	Overview	Part No	Type	Marketing Status	Unit Price (US\$)	Mounted Device
★		NUCLEO-F401RE	Nucleo64	Active	13.0	STM32F401RETx

- Select **NUCLEO-F401RE**
- If you need, download datasheet and other documentation
- **Start Project**
- Initialize peripherals with their default Mode? **YES**

Home

/ STM32F401RETx - NUCLEO-F401RE

/ Untitled - Project Manager

GENERATE CODE

Pinout & Configuration

Clock Configuration

Project Manager

Tools

Project

Code Generator

Advanced Settings

Project Settings

Project Name

first_trial

Project Location

C:\Users\MiSPLA\Dropbox (DEIB)

Browse

Application Structure

Basic

 Do not generate the ma...

Toolchain Folder Location

C:\Users\MiSPLA\Dropbox (DEIB)\first_trial\

Toolchain / IDE

MDK-ARM V5

 Generate Under Root

Linker Settings

Minimum Heap Size 0x200

Minimum Stack Size 0x400

Mcu and Firmware Package

Mcu Reference

STM32F401RETx

Firmware Package Name and Version

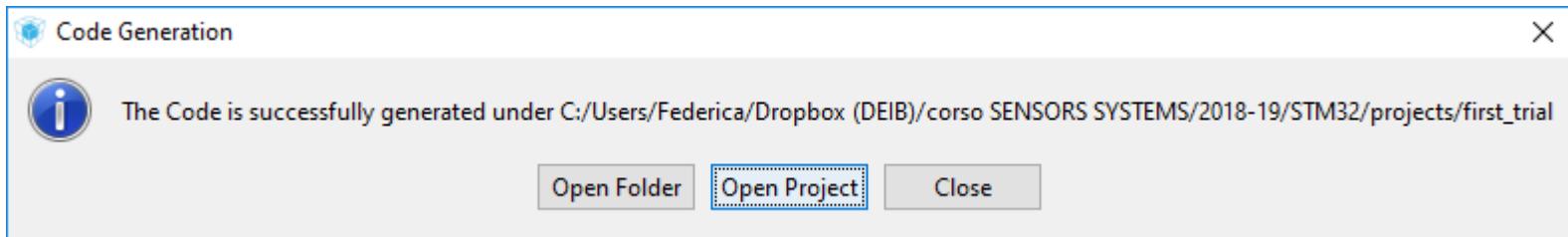
STM32Cube FW_F4 V1.23.0

 Use latest available version Use Default Firmware Location

- In the Project manager tab:
 - project name
 - project location
 - **MDK-ARM V5**
- **GENERATE CODE**, the code will be generated (the first time you generate it requires to install libraries through Pack-Installer).



Open Keil project



- Select **Open Project**
- In the future if you need to generate the code again and you already have the project open in Keil, select **Close**, and in Keil accept to reload the project



Project code

Rebuild

The screenshot shows the MDK-ARM IDE interface. The Project window on the left displays a project named 'first_trial' with various source files like main.c, stm32f4xx_it.c, and startup_stm32f401xe.s. The main code editor shows the main.c file with comments for user code sections and HAL_Init() calls. The Build Output window at the bottom shows the compilation process and a successful build with 0 errors and 0 warnings.

```
C:\Users\Federica\Dropbox (DEI)\corso SENSORS SYSTEMS\2018-19\STM32\projects\first_trial\MDK-ARM\first_trial.uvprojx - μVision  
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help  
Project first_trial  
first_trial Application/MDK-ARM Application/User main.c stm32f4xx_it.c stm32f4xx_hal_msp.c Drivers/STM32F4xx_Hal_Driver CMSIS  
main.c /* USER CODE BEGIN 1 */  
int main(void)  
{  
    /* USER CODE END 1 */  
  
    /* MCU Configuration-----*/  
  
    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */  
    HAL_Init();  
  
    /* USER CODE BEGIN Init */  
  
    /* USER CODE END Init */  
  
    /* Configure the system clock */  
    SystemClock_Config();  
  
    /* USER CODE BEGIN SysInit */  
  
    /* USER CODE END SysInit */  
  
    /* Initialize all configured peripherals */  
    MX_GPIO_Init();  
    MX_USART2_UART_Init();  
    /* USER CODE BEGIN 2 */  
  
    /* USER CODE END 2 */  
  
    /* Infinite loop */  
    /* USER CODE BEGIN WHILE */  
    /* USER CODE END WHILE */  
  
compiling stm32f4xx_hal_uart.c...  
compiling stm32f4xx_hal_rcc_ex.c...  
compiling stm32f4xx_hal_flash.c...  
compiling stm32f4xx_hal_rcc.c...  
compiling stm32f4xx_hal_flash_ex.c...  
compiling stm32f4xx_hal_flash_ramfunc.c...  
compiling stm32f4xx_hal_gpio.c...  
compiling stm32f4xx_hal_pwr.c...  
compiling stm32f4xx_hal_dma.c...  
compiling stm32f4xx_hal_dma_ex.c...  
compiling stm32f4xx_hal_cortex.c...  
compiling stm32f4xx_hal_pwr_ex.c...  
compiling stm32f4xx_hal.c...  
compiling system_stm32f4xx.c...  
linking...  
Program Size: Code=3760 RO-data=460 RW-data=16 ZI-data=108  
FromELF: creating hex file.  
"first_trial\first_trial.axf" - 0 Error(s), 0 Warning(s).  
Build Time Elapsed: 00:00:23
```

- Open **main.c** from the *Project window* (on the left)
- Some parts of the code are already written (based on user settings selected with Cube)
- Between **/*USER CODE BEGIN** and **/*USER CODE END** it is possible to add code, which will be not modified even in case the settings in Cube are modified and code is generated again.
- Click **Rebuild**
- Check in the *Build Output* window if the code has been build with **0 –Error(s)**



Start debug session

The screenshot shows the uVision IDE interface. The main window displays the `main.c` file with the following code:

```
74 int main(void)
75 {
76     /* USER CODE BEGIN 1 */
77
78     /* USER CODE END 1 */
79
80     /* MCU Configuration-- */
81
82     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
83     HAL_Init();
84
85     /* USER CODE BEGIN Init */
86
87     /* USER CODE END Init */
88
89     /* Configure the system clock */
90     SystemClock_Config();
91
92     /* USER CODE BEGIN SysInit */
93
94     /* USER CODE END SysInit */
95
96     /* Initialize all configured peripherals */
97     MX_GPIO_Init();
98     MX_USART2_UART_Init();
99     /* USER CODE BEGIN 2 */
100
101    /* USER CODE END 2 */
102
103    /* Infinite loop */
104    /* USER CODE BEGIN WHILE */
105    /* USER CODE END WHILE */
106}
```

A red arrow points to the "Start/Stop debug session" button in the toolbar. The text "Start/Stop debug session" is overlaid on the code area.

The bottom pane shows the "Build Output" with the following log:

```
compiling stm32f4xx_hal_uart.c...
compiling stm32f4xx_hal_rcc_ex.c...
compiling stm32f4xx_hal_flash.c...
compiling stm32f4xx_hal_rcc.c...
compiling stm32f4xx_hal_flash_ex.c...
compiling stm32f4xx_hal_flash_ramfunc.c...
compiling stm32f4xx_hal_gpio.c...
compiling stm32f4xx_hal_pwr.c...
compiling stm32f4xx_hal_dma.c...
compiling stm32f4xx_hal_dma_ex.c...
compiling stm32f4xx_hal_cortex.c...
compiling stm32f4xx_hal_pwr_ex.c...
compiling stm32f4xx_hal.c...
compiling system_stm32f4xx.c...
linking...
Program Size: Code=3760 RO-data=460 RW-data=16 ZI-data=1088
FromELF: creating hex file...
"first_trial\first_trial.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:27
```

The status bar at the bottom right says "ST-Link Debugger".

- Connect the Nucleo board to the PC through the USB link
- Click **Start/Stop debug session**
- If asked, update the ST-link drivers
- Evaluation Mode → OK



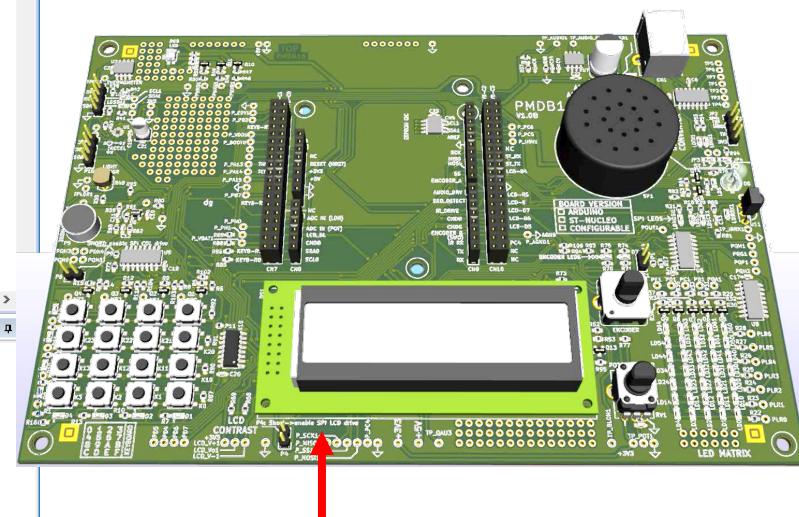
Debug session

Run (F5)

Stop

The screenshot shows the MDK-ARM uVision IDE interface. The top menu bar includes File, Edit, View, Project, Flash, Debug, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations, project management, and debugging. The Registers window shows the core registers (R0-R13, R14-LR, R15-PC, xPSR) with their current values. The Disassembly window displays assembly code for the main.c and startup_stm32f401xe.s files. The Call Stack + Locals window shows the main function with its location value at 0x00000000 and type int f(). The bottom status bar indicates the project is running with a code size limit of 32K and the file first_trial\first_trial.axf loaded. The ST-Link Debugger status shows t1: 0.0001783.

- Click **Run (F5)** to run the code
- In this first trial a red LED on the Sensors Board should switch on
- **Stop** and click on **Start/Stop debug session** to come back to the code.



Red LED



End

Congratulations!

Your tools have been
correctly installed!