Simplifying Motion MEMS and Environmental Sensors Design Using the STM32CubeMX and the X-CUBE-MEMS1 Software Pack

(Hands-On Workshop)

Andrea Vitali

life.augmented

Technology Tour 2019

Vancouver, BC | September 24

Required HW

- **Laptop with USB type A**
- **64 bit OS** Win (7, 10), MacOS, Linux (Generic, Debian, RPM)
  - Virtual Machine is also OK (example: Parallel, VM Fusion…)
  - **Administration rights** are required for SW tools installation

Developing with Sensors Made Simple

Hands-On

- USB Flash drive with relevant material for the hands-on
- **Please copy** the content of the USB drive on your laptop

life.augmented

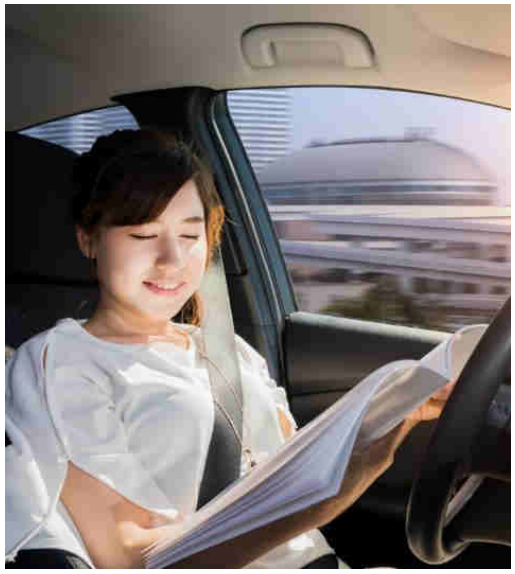
Agenda

- Introduction
- Products & Ecosystem
- Hands-on Training Session
  - Installation Process
    - STM32CubeMX libraries installation
    - STM32CubeIDE installation
    - Tera Term installation
  - Hardware Description
  - Lab Examples:
    - Lab 0 - Read all sensors in polling mode
    - Lab 1 - LIS2DW12 accelerometer orientation change detection
    - Lab 2 - LPS22HH barometer read from internal FIFO
    - Lab 3 - LSM6DSO accelerometer embedded step counter
    - Lab 4 - LIS2DW12 accelerometer acceleration detection
    - Lab 5 - LSM6DSO accelerometer single/double tap detection
    - Lab 6 - LSM6DSO accelerometer 6.6 kHz data rate read at 100 Hz

life.augmented

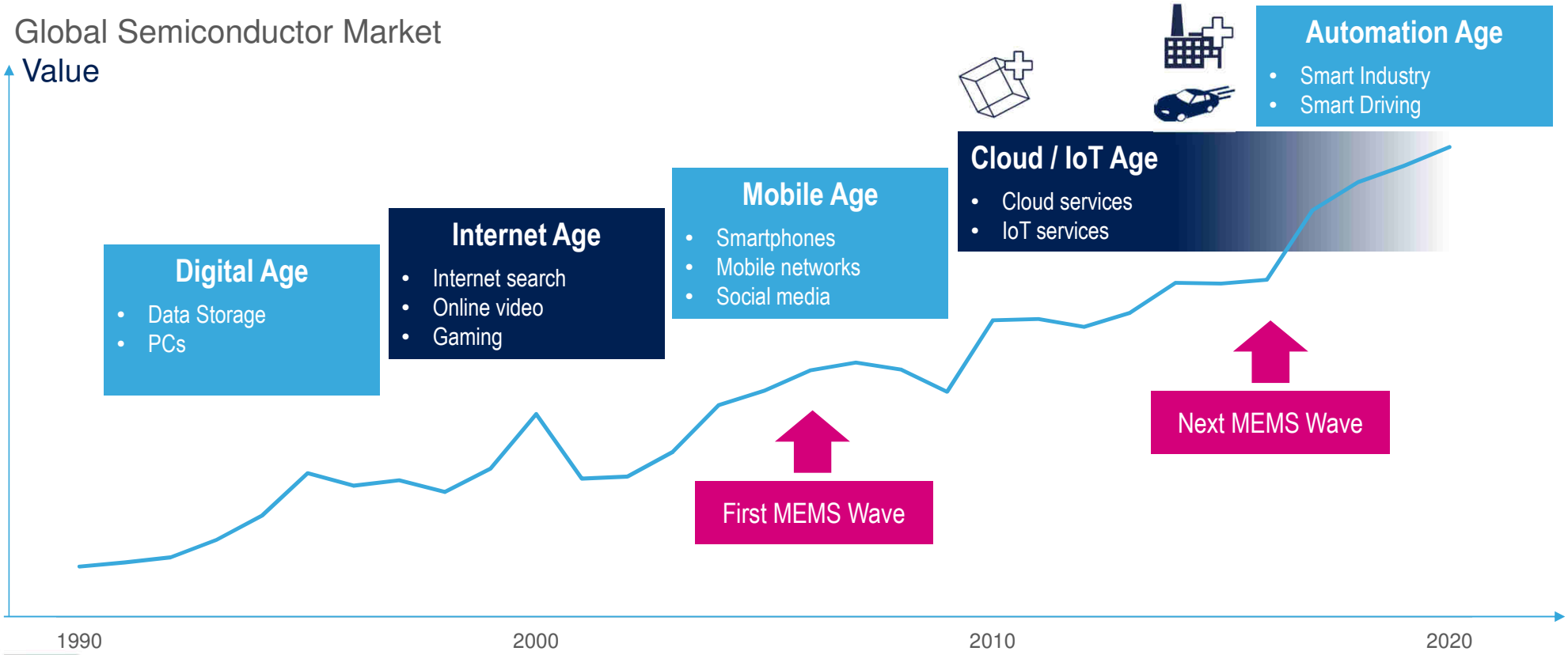
ST Addresses Four End Markets

Automotive	Industrial	Personal Electronics	Communications Equipment, Computers & Peripherals



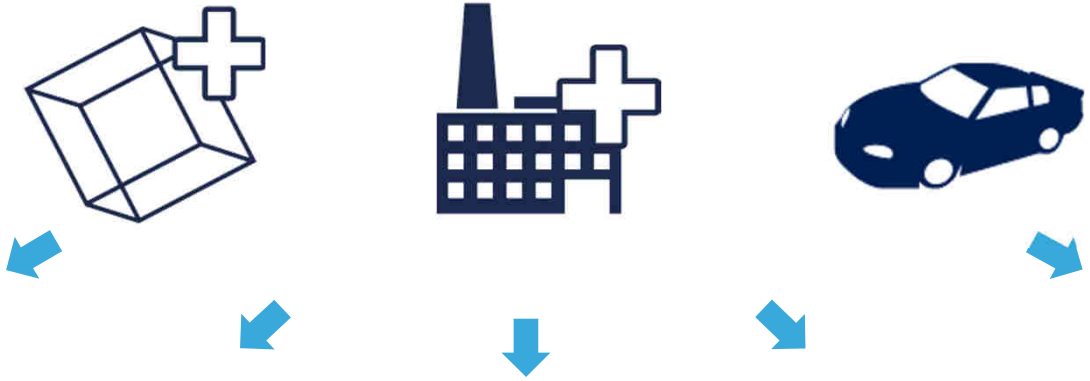
life.augmented

Sensors Trends



life.augmented

Sensors Evolution



Technologies
Manufacturing

Accuracy
Stability

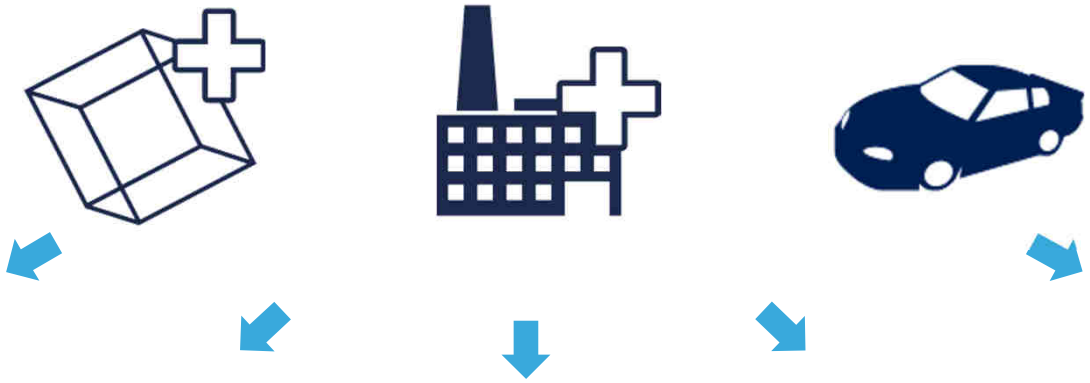
Multi Sensors
Integration

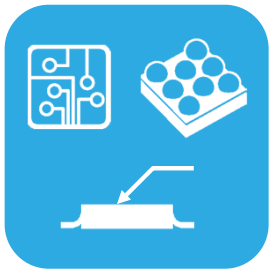
Embedded
Smart Functions

Low Power
Always ON

life.augmented

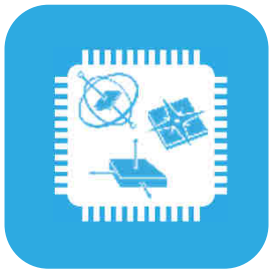
Sensors Evolution





17 Billion Units
Shipped

+83%
Temp. Stability



-70%
size reduction

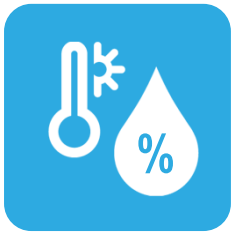
Embedded
FSM and Dec. Tree

-91%
Current Consumption

life.augmented

Broad Sensors Portfolio

Motion



Environment

Interactivity

Micro-Actuators

Optical

life.augmented

ST: Products & Ecosystem

Connect

Translate

Move
Actuate

Power

Secure

Process

Sense

Sensor Tile

ProfiMEMS

STM32 Nucleo Development
& Expansion boards

Pre-integrated SW for vertical applications

Smart Things

Smart Home

Smart City

Smart Industry

Development Ecosystem

Code generators

Prototyping
software

IDE
Development
environments

Debug
solutions

Simulation
and analysis tools

On-line
design tools

ST Community

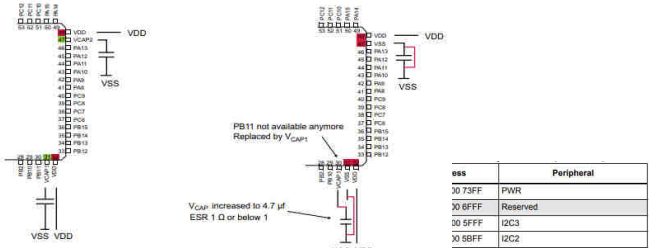
life.augmented

Partner
Program

life.augmented

Developing with Sensors

Complicated



ess	Peripheral
00 73FF	PWR
00 6FFF	Reserved
00 5FFF	I2C3
00 5BFF	I2C2
00 57FF	I2C1
0x4000 4800 - 0x4000 53FF	Reserved
0x4000 4400 - 0x4000 47FF	USART2
0x4000 4000 - 0x4000 43FF	I2S3ext
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2
0x4000 3400 - 0x4000 37FF	I2S2ext
0x4000 3000 - 0x4000 33FF	IWDG
0x4000 2C00 - 0x4000 2FFF	WWDG
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers
0x4000 1000 - 0x4000 27FF	Reserved
0x4000 0C00 - 0x4000 0FFF	TIM5
0x4000 0800 - 0x4000 0BFF	TIM4
0x4000 0400 - 0x4000 07FF	TIM3

APB1

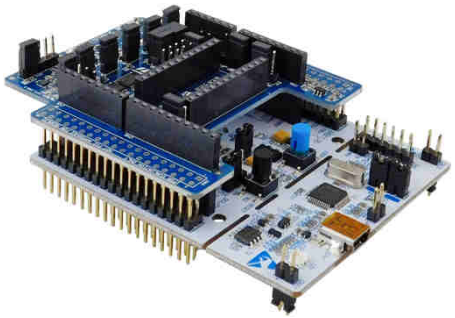
Table 42. **CTRL1_XL register**

ODR_XL3	ODR_XL2	ODR_XL1	ODR_XL0	FS1_XL	FS0_XL	LPF2_XL_EN	0

Table 43. **CTRL1_XL register description**

ODR_XL[3:0]	Accelerometer ODR selection (see Table 44)
FS[1:0]_XL	Accelerometer full-scale selection (see Table 45)
LPF2_XL_EN	Accelerometer high-resolution selection (0: output from first stage digital filtering selected (default); 1: output from LPF2 second filtering stage selected)

Simple

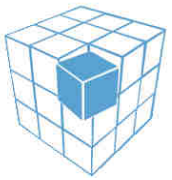


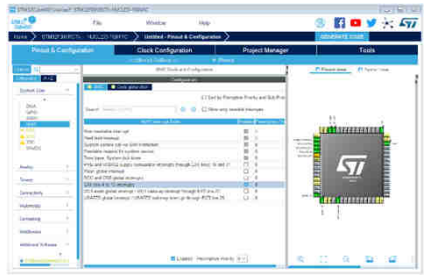
HW

Nucleo + X-Nucleo

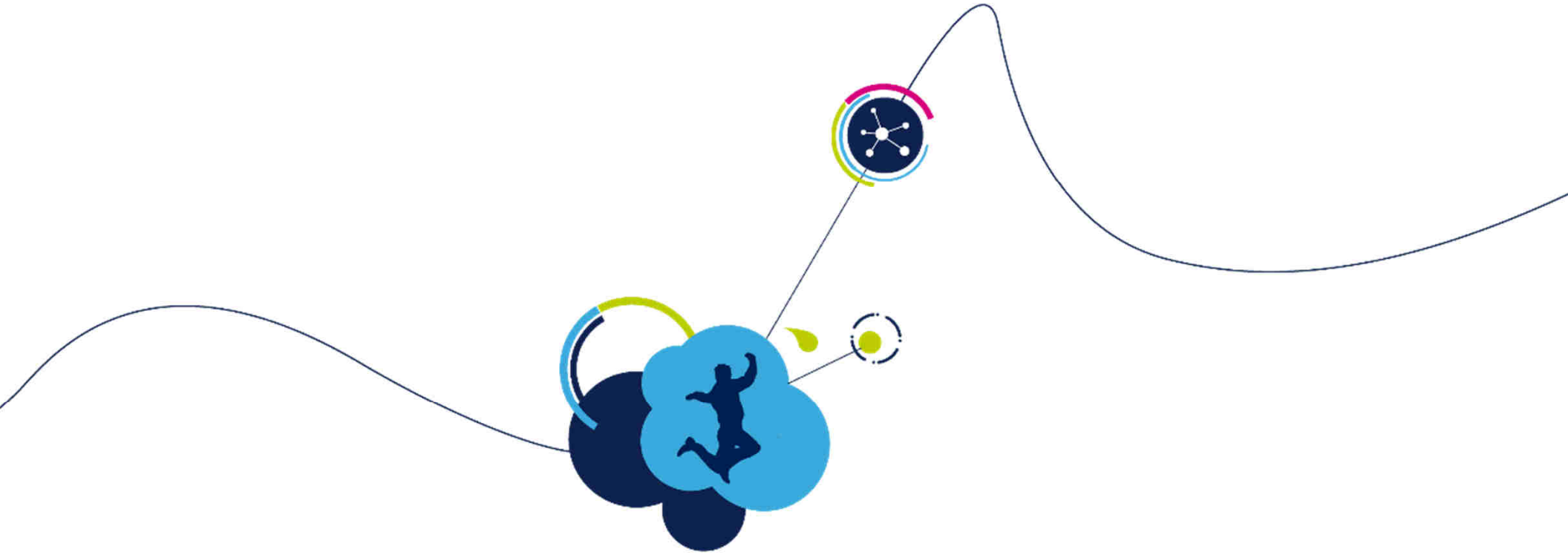
SW

STM32CubeIDE + STM32CUBEMX + X-CUBE-MEMS1





life.augmented

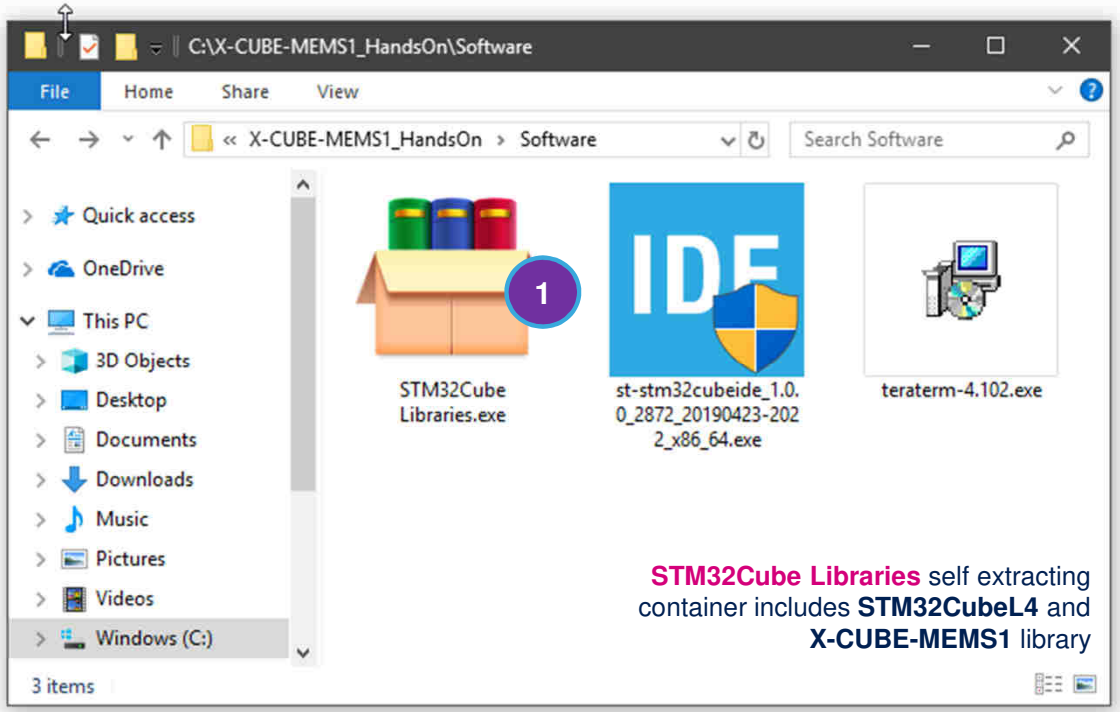


STM32CubeMX Libraries Installation

life.augmented

STM32CubeMX Libraries Installation

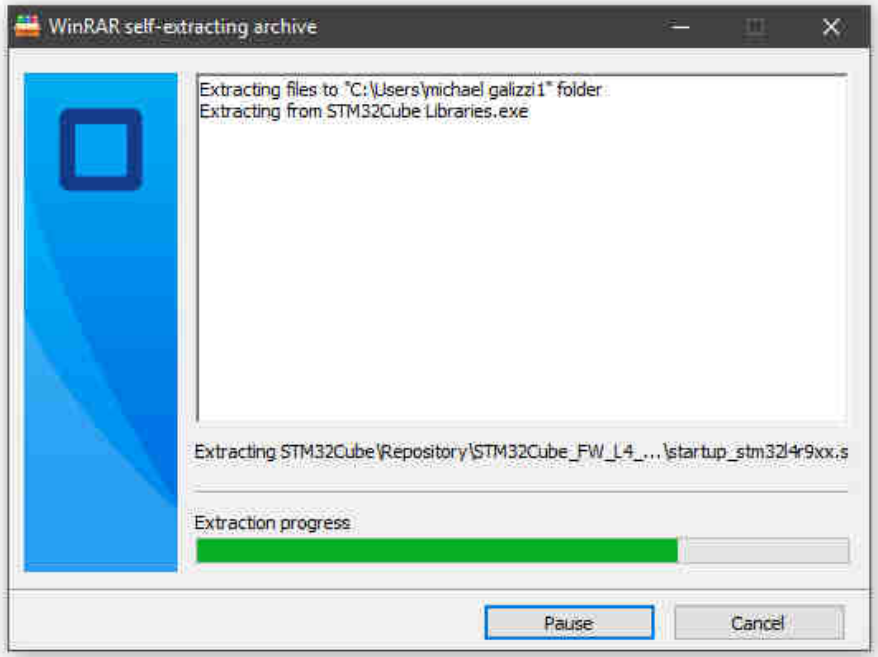
1. Run the **STM32Cube Libraries.exe** self-extracting archive located in **X-CUBE-MEMS1_HandsOn\Software**



life.augmented

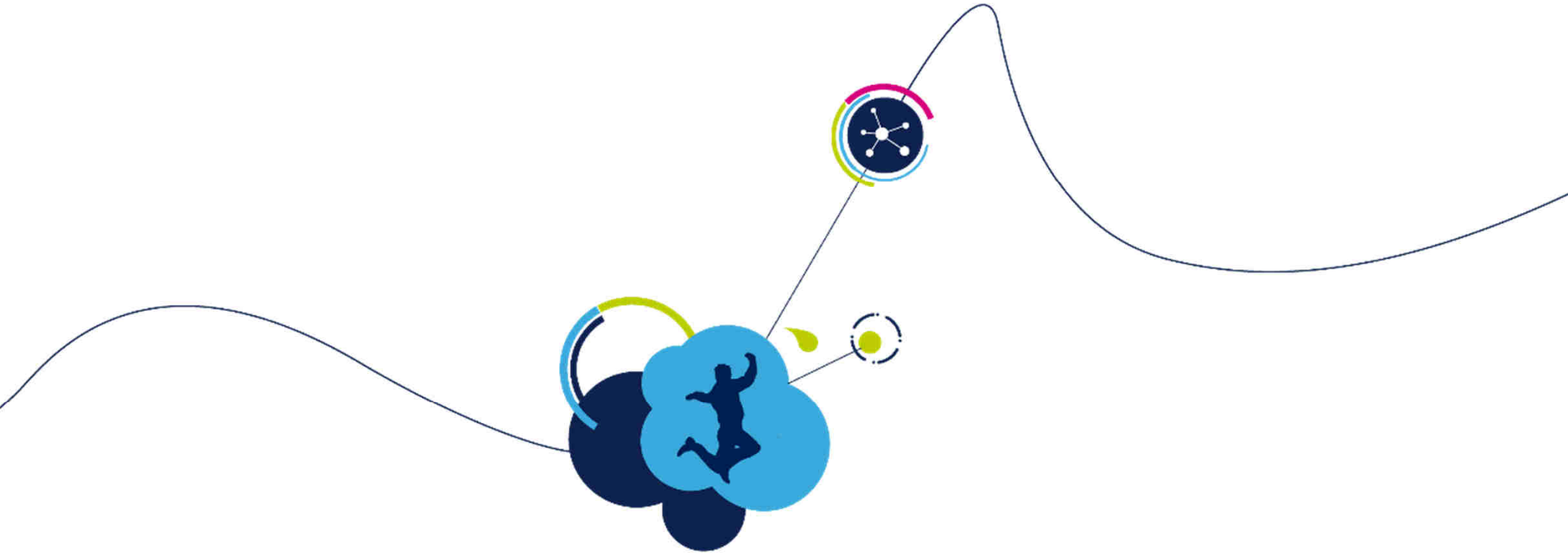
STM32CubeMX Libraries Installation

1. Library extraction may take few minutes.
   Please wait the library to be fully extracted before proceed any further.



Meantime we'll proceed to install and configure STM32CubeIDE

life.augmented

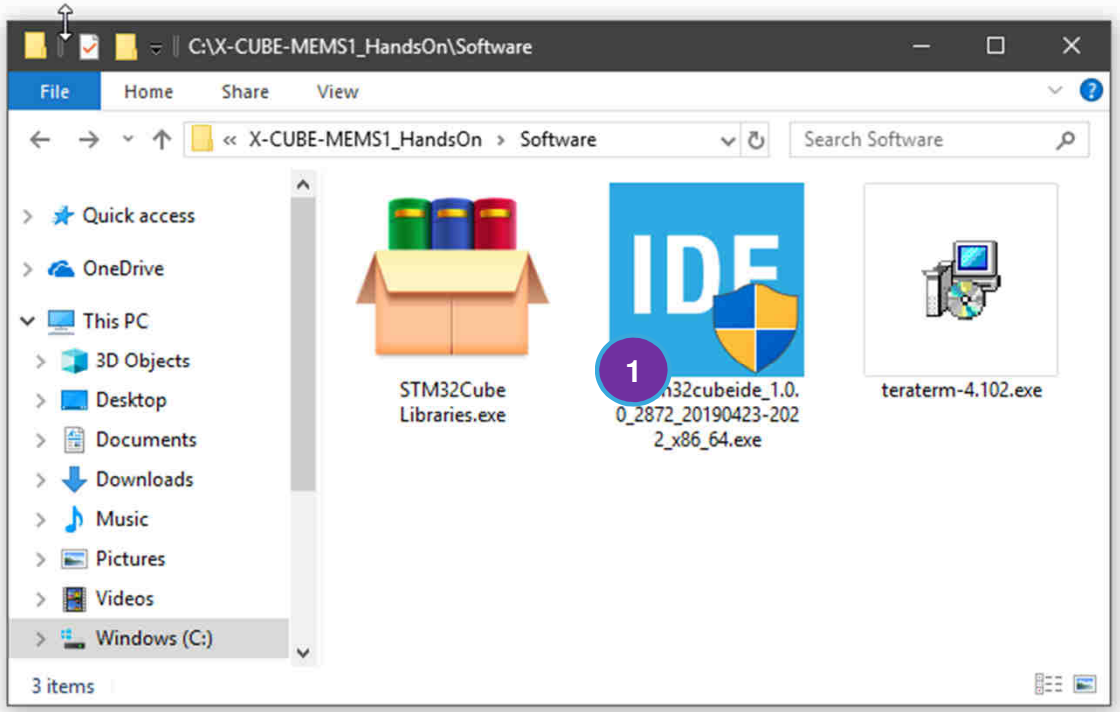


STM32CubeIDE Installation

life.augmented

STM32CubeIDE Installation

1. Run the **STM32CubeIDE** installer located in **X-CUBE-MEMS1_HandsOn\Software**



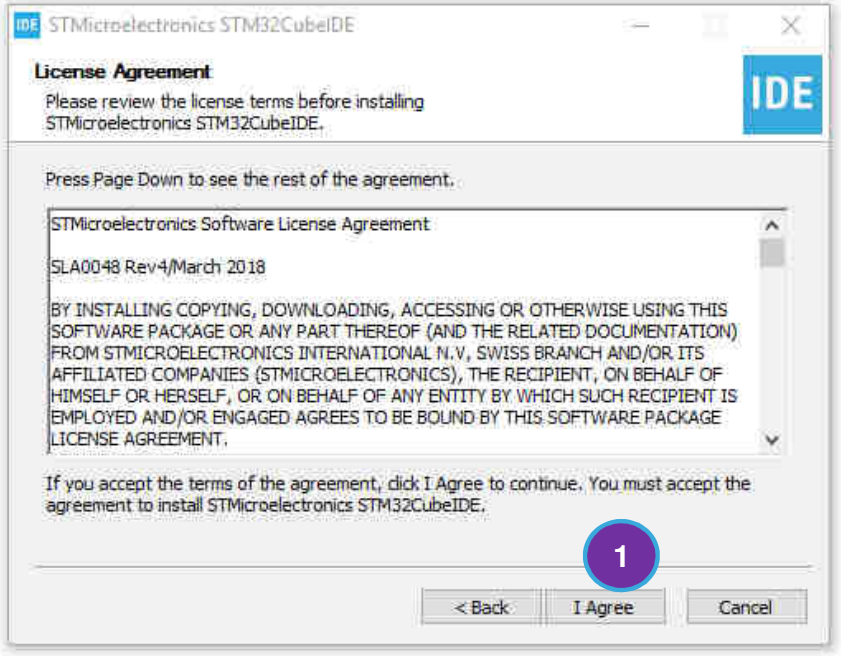
life.augmented

STM32CubeIDE Installation

1. Click **Next >**

STM32CubeIDE Installation

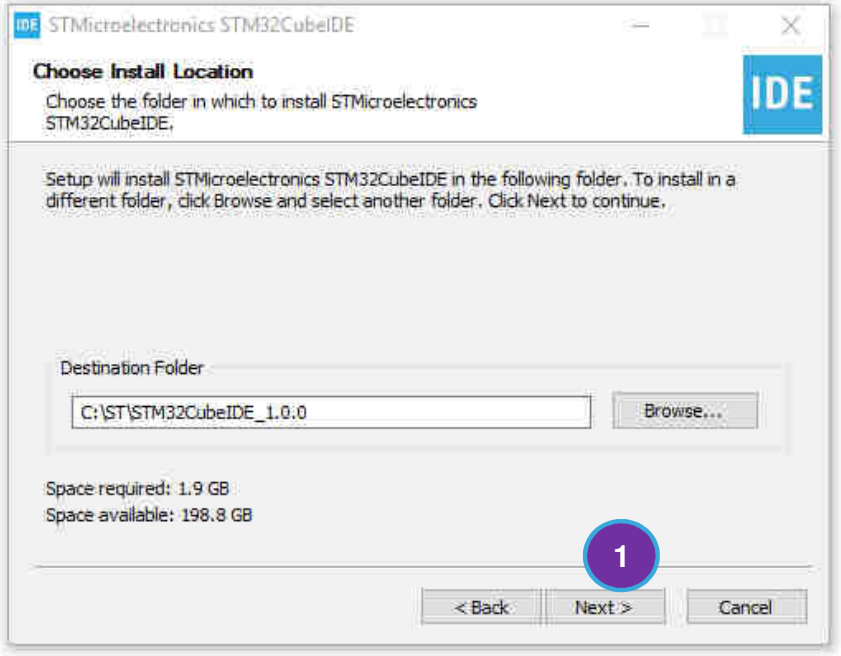
1. Click **I Agree**



life.augmented

STM32CubeIDE Installation

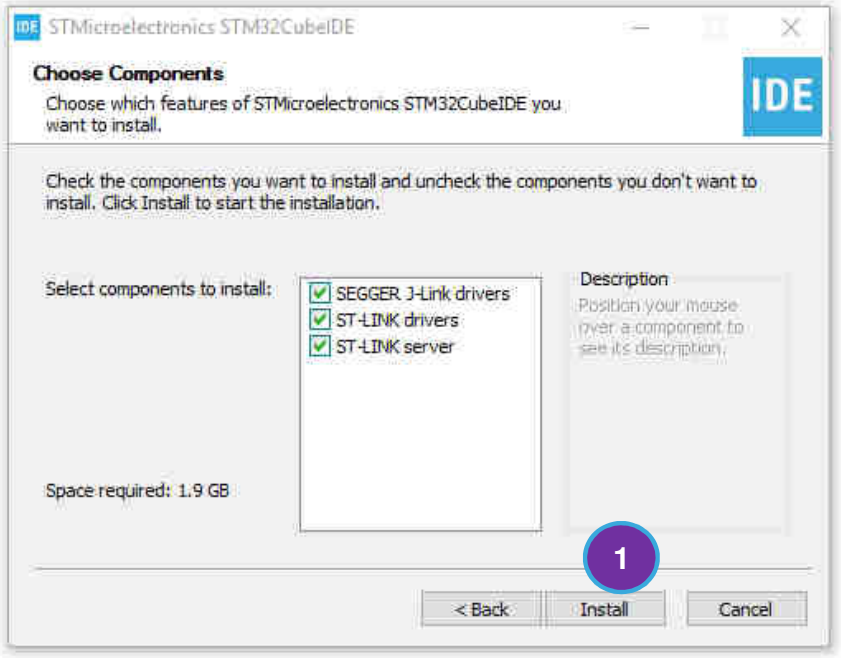
1. Click **Next >**



life.augmented

STM32CubeIDE Installation

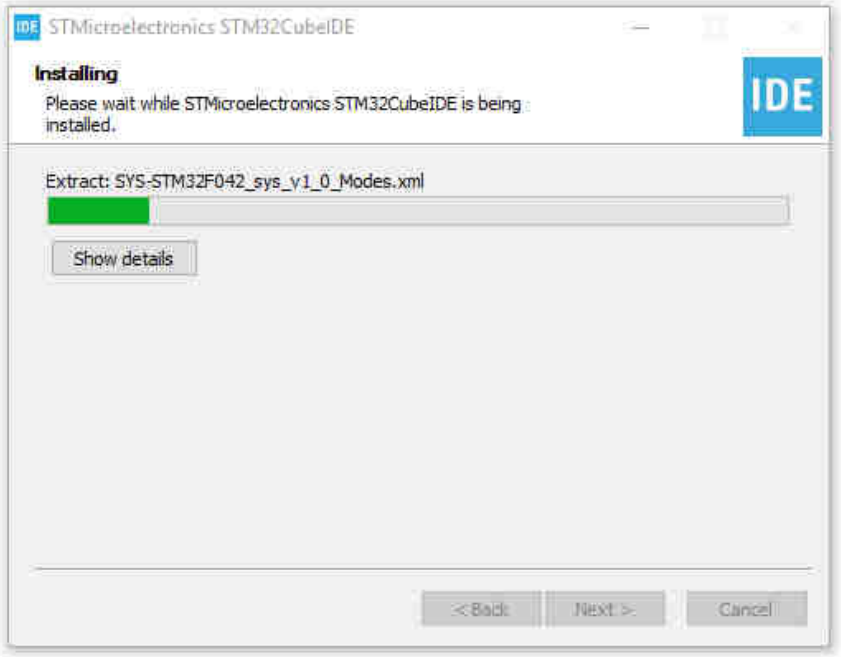
1. Click **Install**

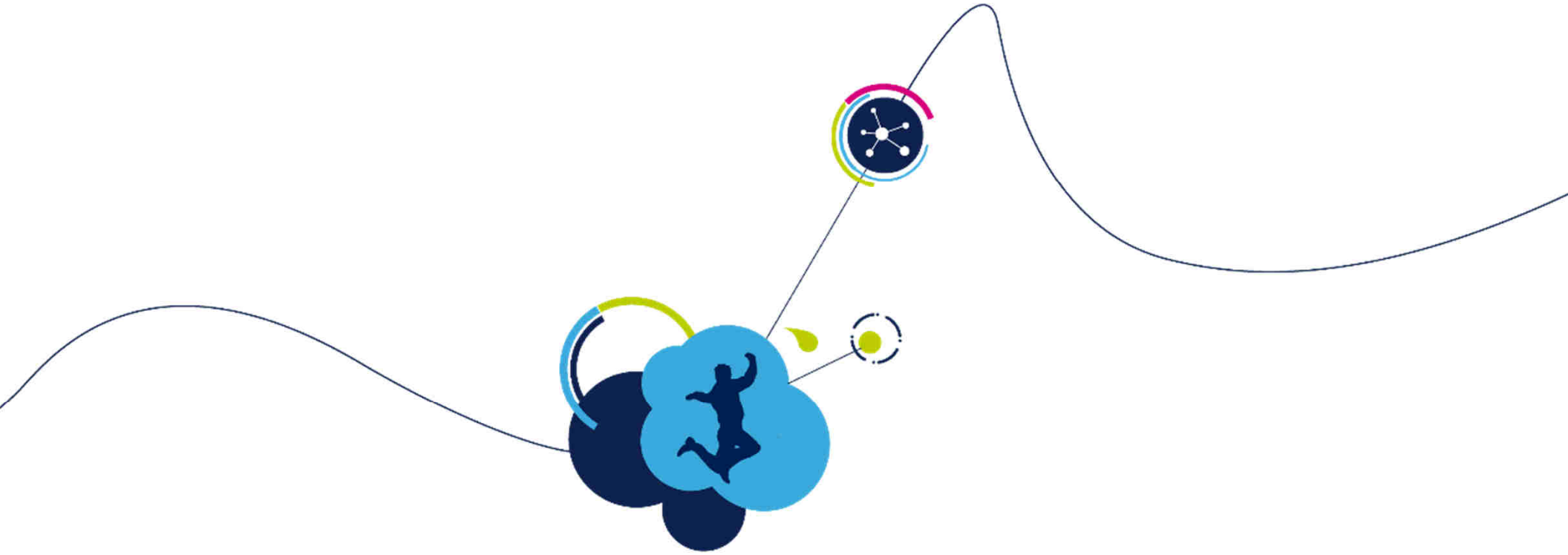


life.augmented

STM32CubeIDE Installation…

1. Wait until installation is completed





STM32 Nucleo Expansion

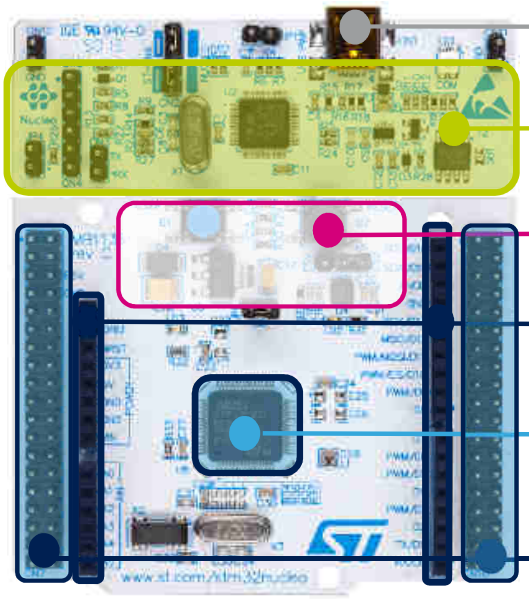
life.augmented

HW #1: STM32 NUCLEO Board

Microcontroller

2

1



Flexible board power supply :
through USB or external source

Integrated ST-Link/V2-1:
mass storage device flash programming

2 push buttons, 2 color Leds

Arduino extension connectors :
easy access to add-ons

STM32 MCU (NUCLEO-L476RG)

Morpho extension headers :
direct acces to all MCU I/Os

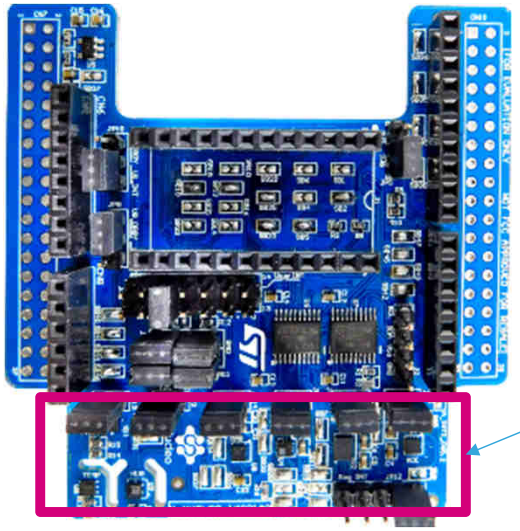
life.augmented

HW #2: X-NUCLEO-IKS01A3 Board

Motion MEMS and environmental sensor

2

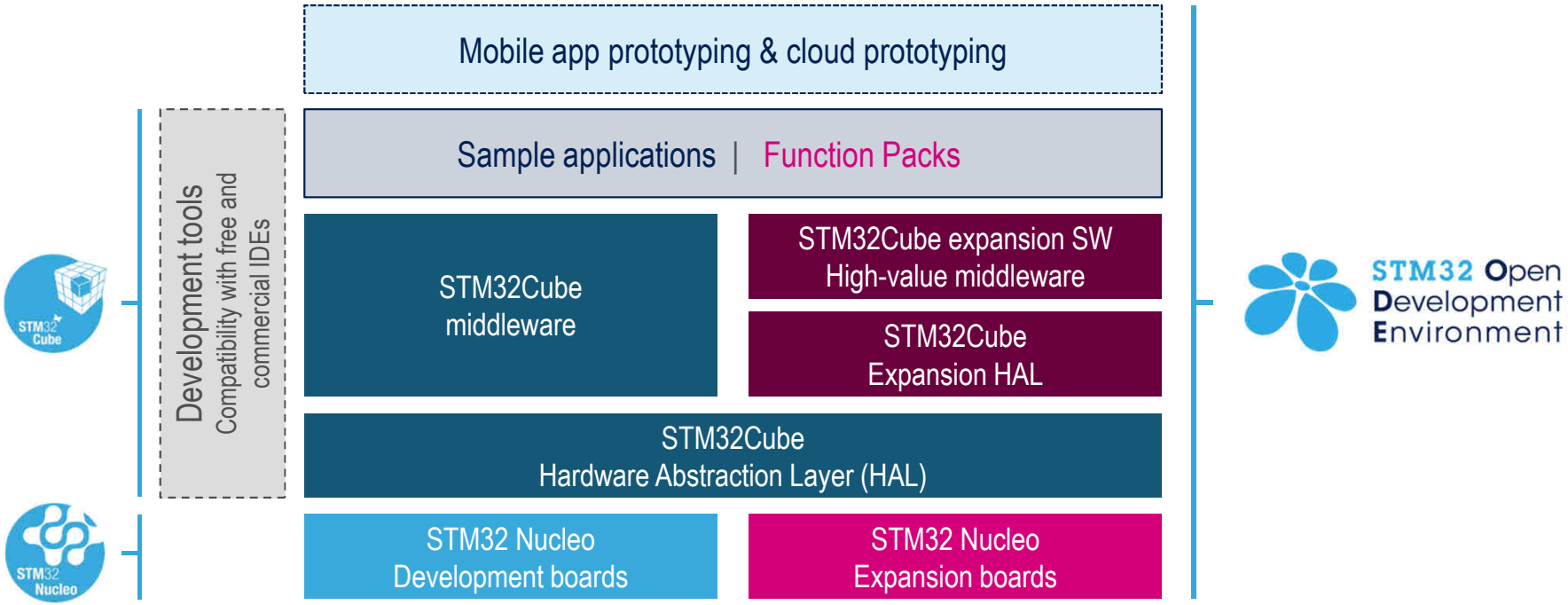
1



- Accelerometer: LIS2DW12
- Accel. + Gyro: LSM6DSO
- Pressure: LPS22HH
- Temperature: STTS751
- Humidity: HTS221
- Magnetometer: LIS2MDL

life.augmented

Development Software Architecture



STM32 Supported IDEs

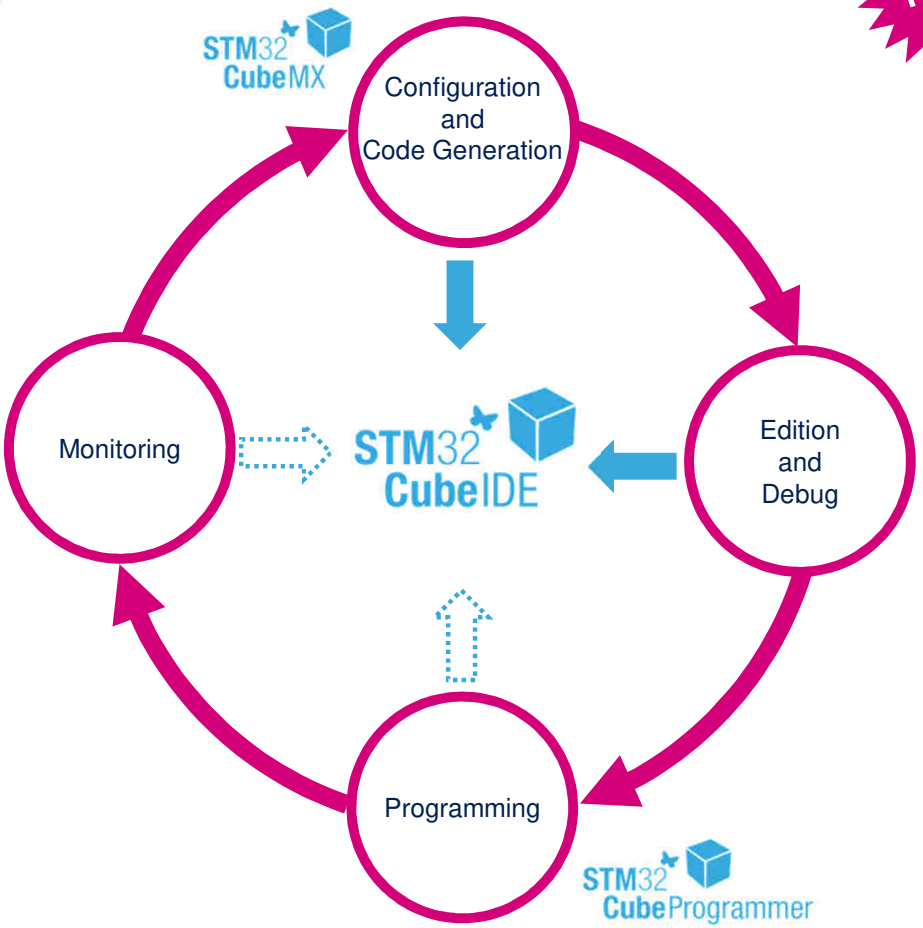
STM32 IDEs

STM32CubeIDE

a[illegible]n-1 Development Tool

FREE

- Complete multi platform family tools
- Cover full development cycle
- Integrated solution



mac OS

life.augmented

Software Development Tools

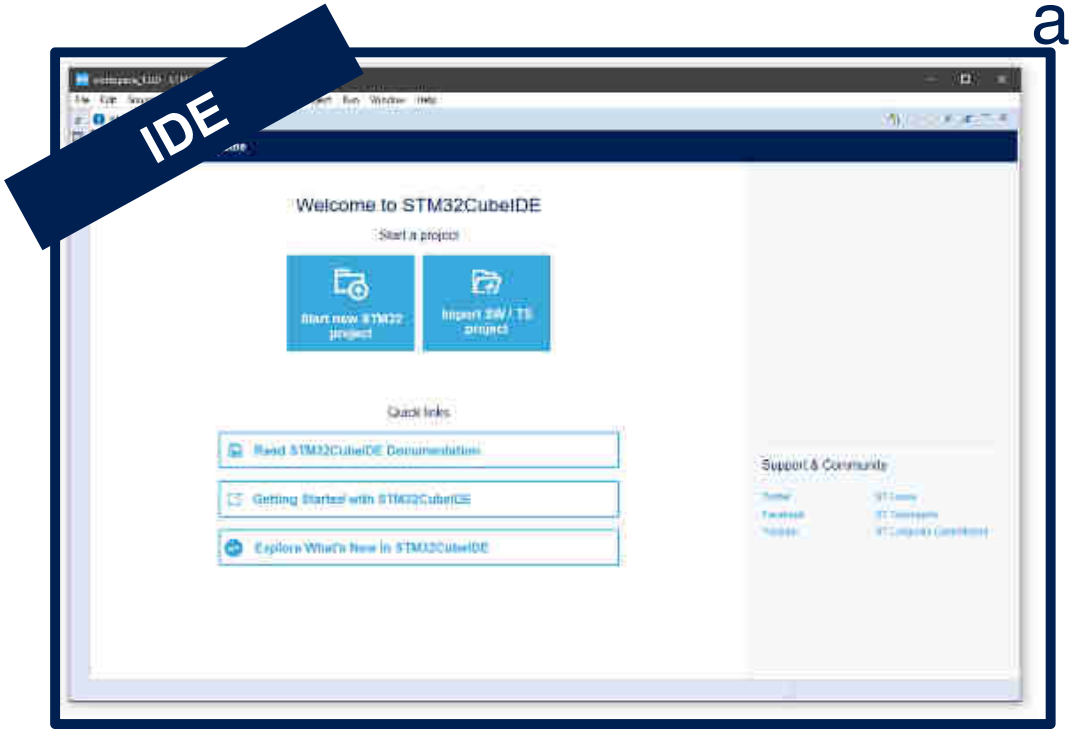
“STM32CubeIDE”
development environment

↳ **“STM32CubeMX”**
initialization code generator

↳ **“X-CUBE-MEMS1”**
application code and drivers for sensors

life.augmented

STM32CubeIDE
an All-in-1 Development Tool



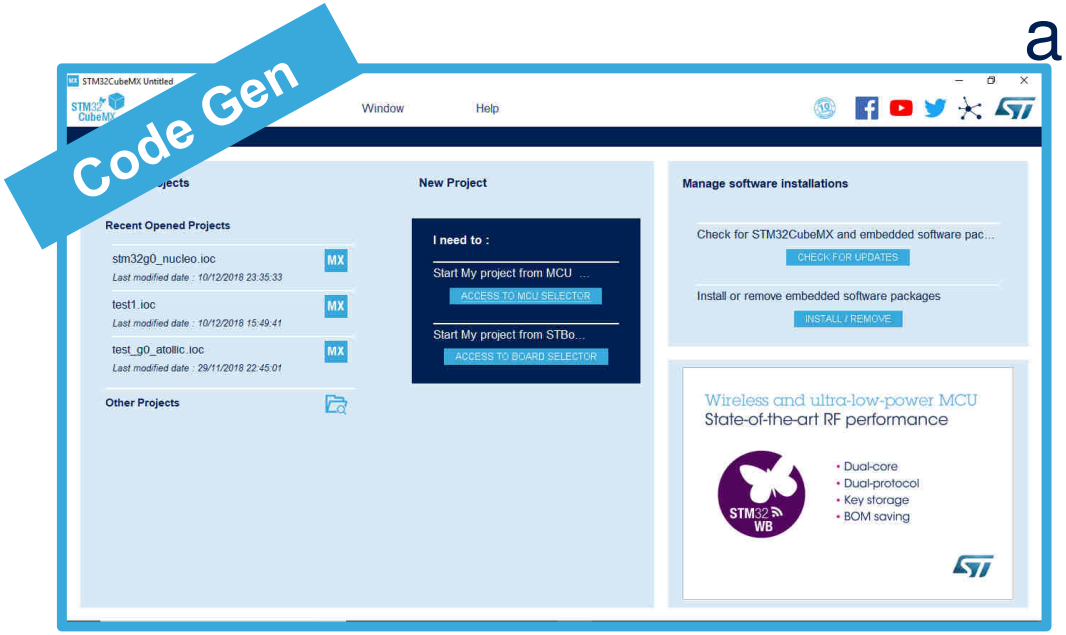
Dedicated IDE for all STM32 microcontrollers

STM32 Cube**MX** is natively integrated as code generator

Based on Eclipse (Text editor), GCC (compiler) and GDB (debugger)

life.augmented

STM32CubeMX
an All-in-1 Development Tool



STM32CubeL0

STM32CubeL1

STM32CubeL4

STM32CubeF1

STM32CubeF3

STM32CubeF2

STM32CubeF4

STM32CubeG0

STM32CubeF0

STM32CubeF7

STM32CubeH7

life.augmented

Very powerful configuration and code generation

Support all STM32 MCU and MPU, with integrated powerful Finder

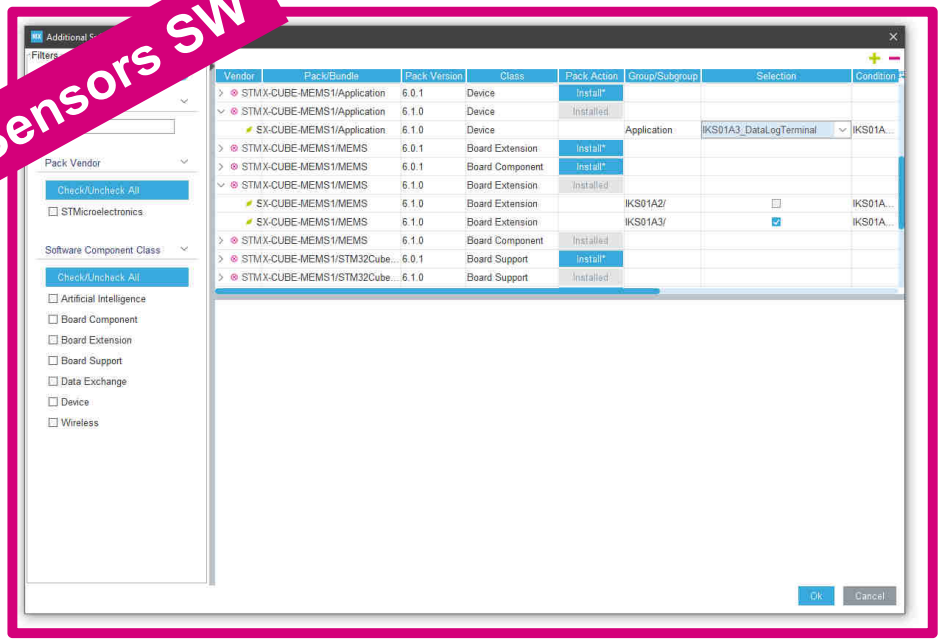
Pinouts, clock tree, peripherals and middleware configuration.

Expandable to support wireless connectivity, sensors and much more!

X-CUBE-MEMS1

an All-in-1 Development Tool

Sensors SW



Accelerometer | Inertial module | Magnetometer | Gyroscope | Pressure Sensor | Temperature Sensor | Humidity sensor

- Drivers for sensors
- Board Support Package firmware architecture
- Include Sample Applications

life.augmented

X-CUBE-MEMS1
an All-in-1 Development Tool

Sample Applications

- Data Logging on Terminal All Sensors


Accelerometer

6D Orientation, Wake Up, Self Test


Magnetometer

Self Test


Pressure Sensor

FIFO Mode


Inertial module

6D, FIFO Modes, FreeFall, Pedometer, Tap, Tilt, Wake Up, Self Test

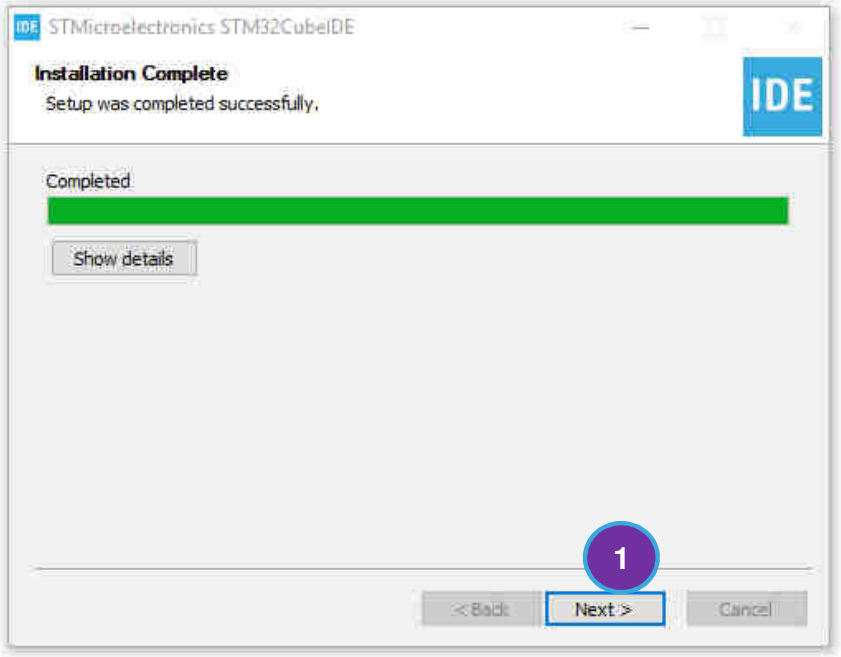

Temperature Sensor

Temperature Limit

life.augmented

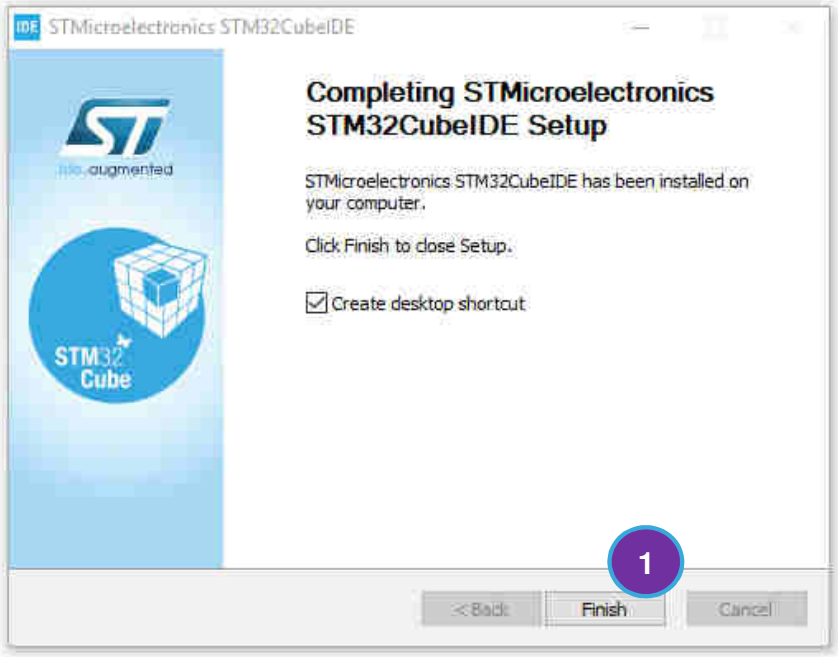
…STM32CubeIDE Installation

1. Click **Next >**



STM32CubeIDE Installation

1. Click **Finish**

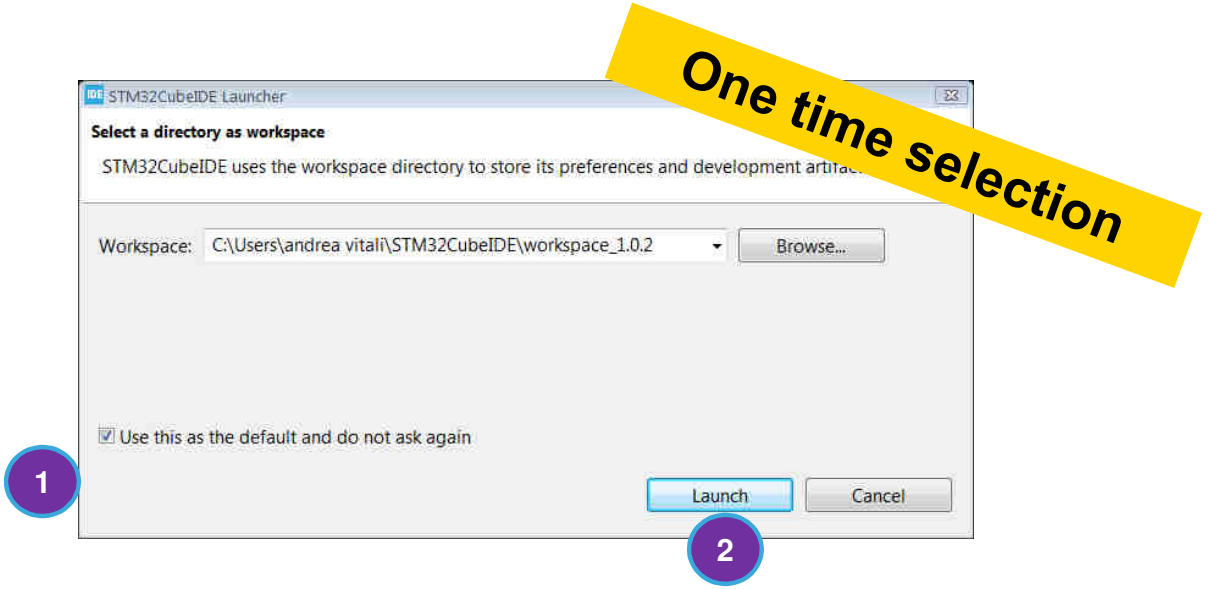


STM32CubeIDE Configuration

1. Open **STM32CubeIDE** by double clicking the icon on your desktop
2. Wait until **STM32CubeIDE** is loading

STM32CubeIDE Configuration

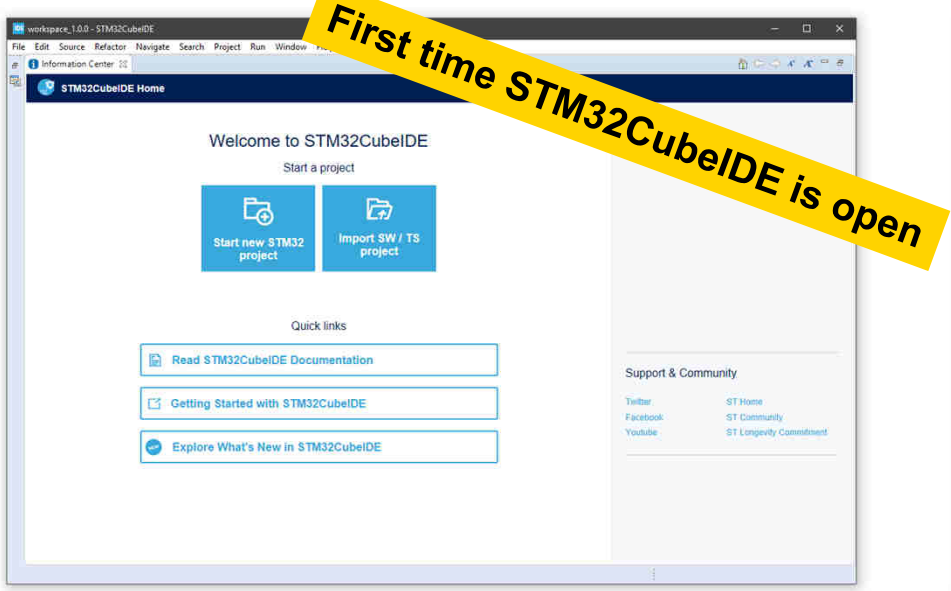
1. Check **Use this as the default and do not ask again**
2. Click **Launch**

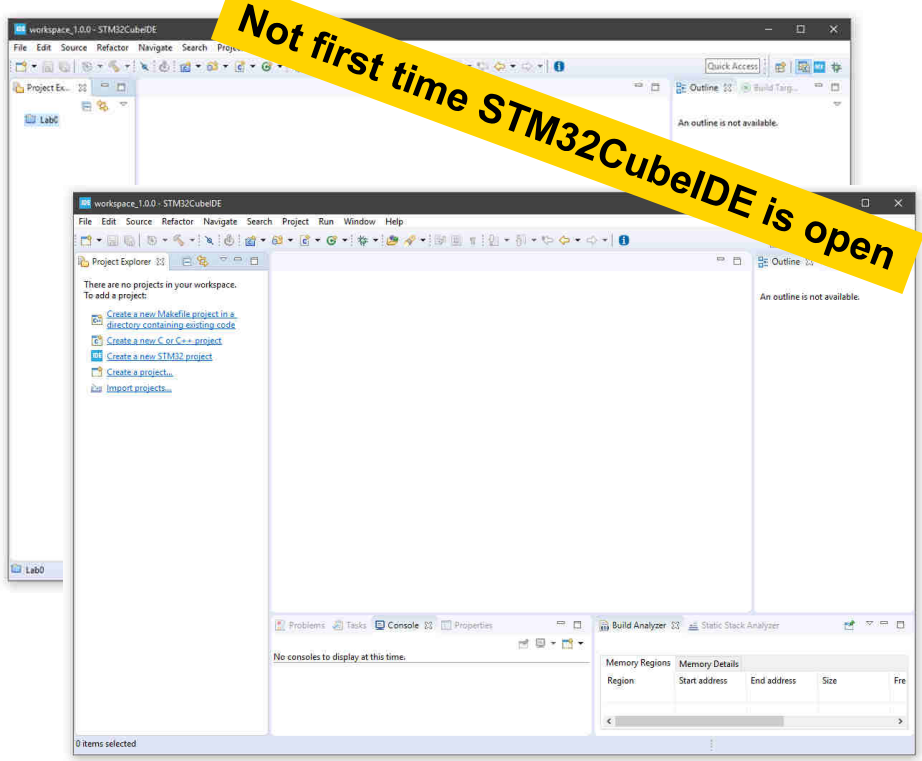


life.augmented

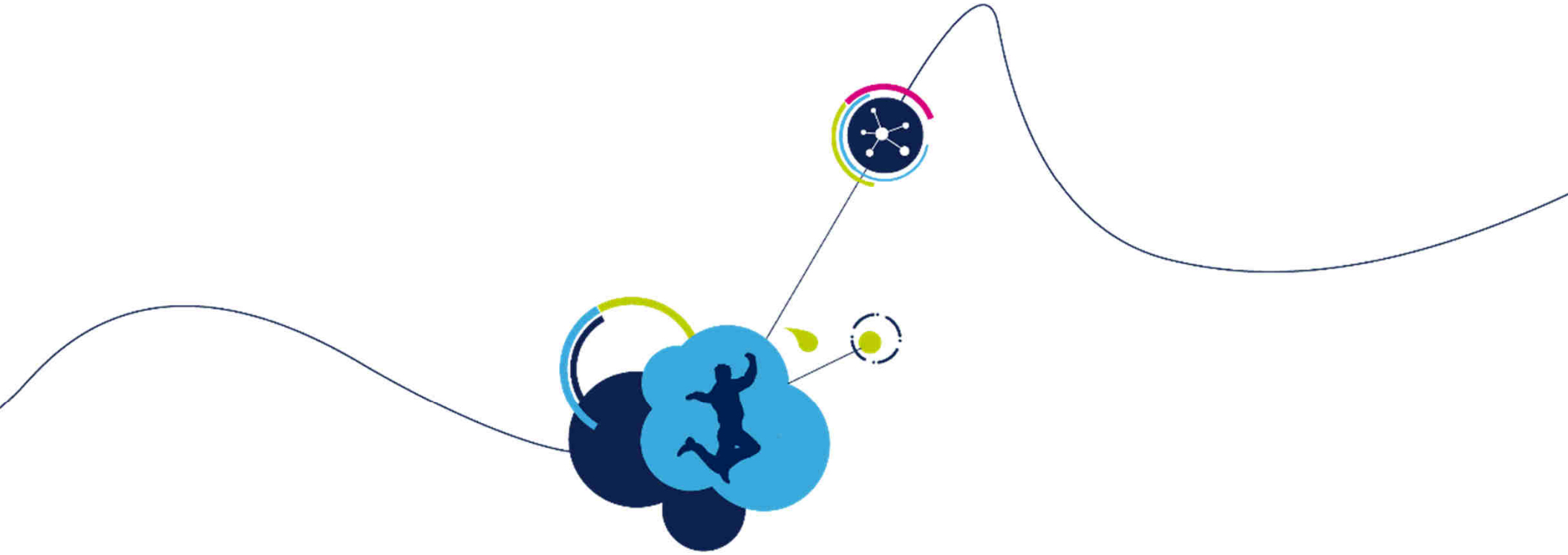
STM32CubeIDE Configuration

The main interface of STM32CubeIDE will appear differently, depending if it is the first time it has been run or not:





life.augmented

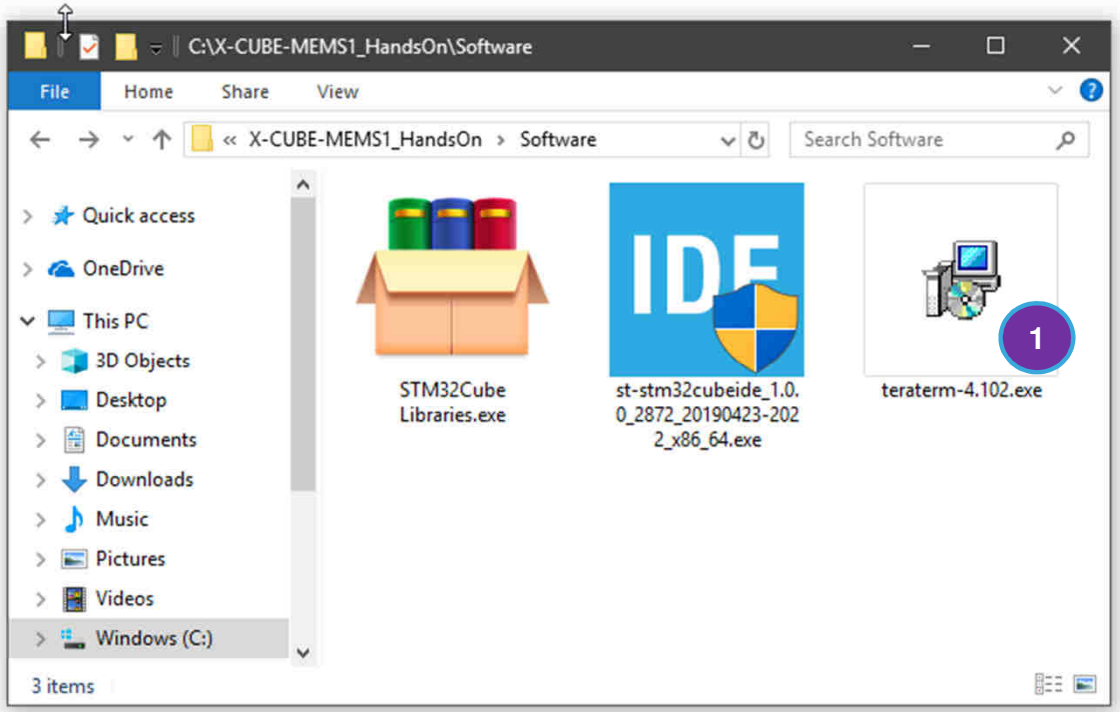


Tera Term Installation

life.augmented

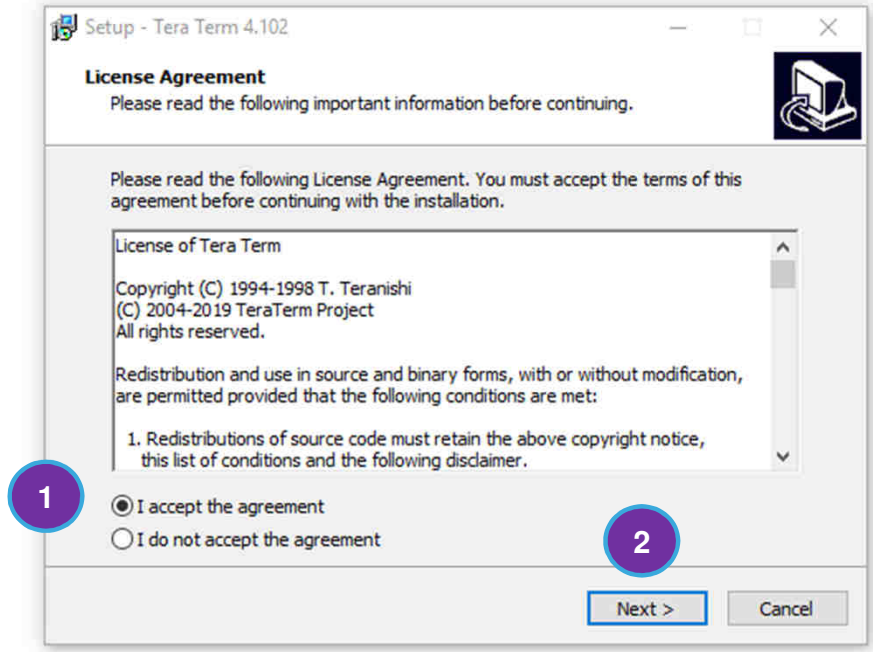
Tera Term Setup 1/9

1. Lunch the ***teraterm-4.102.exe*** installer inside the **X-CUBE-MEMS1_HandsOn\Software** directory



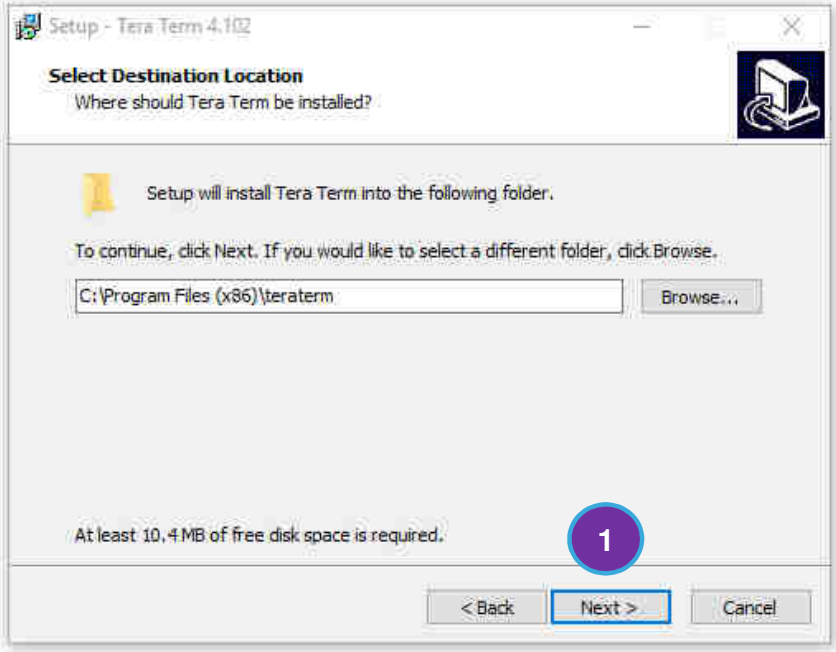
Tera Term Setup 2/9

1. **Accept the agreement**
2. Click on **Next**



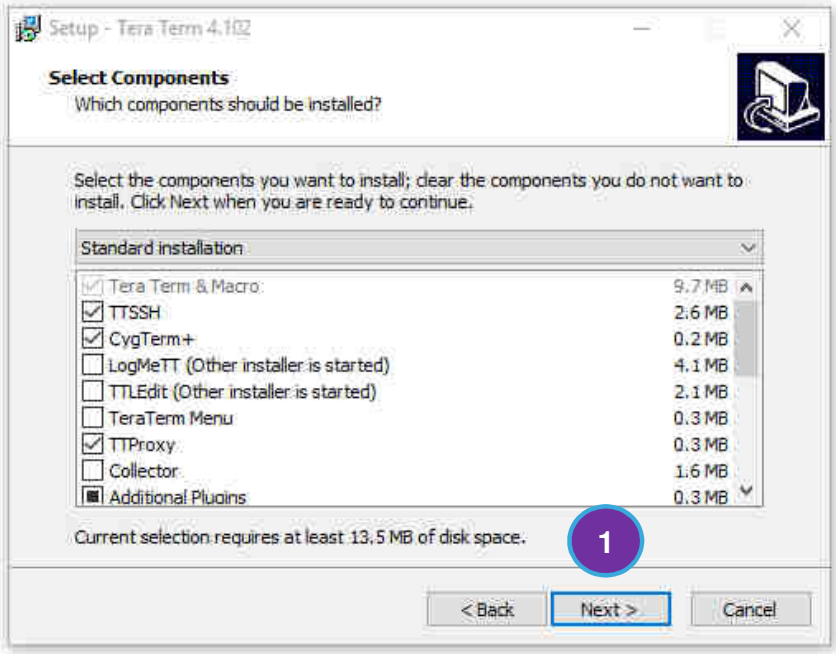
Tera Term Setup 3/9

1. Click on **Next**



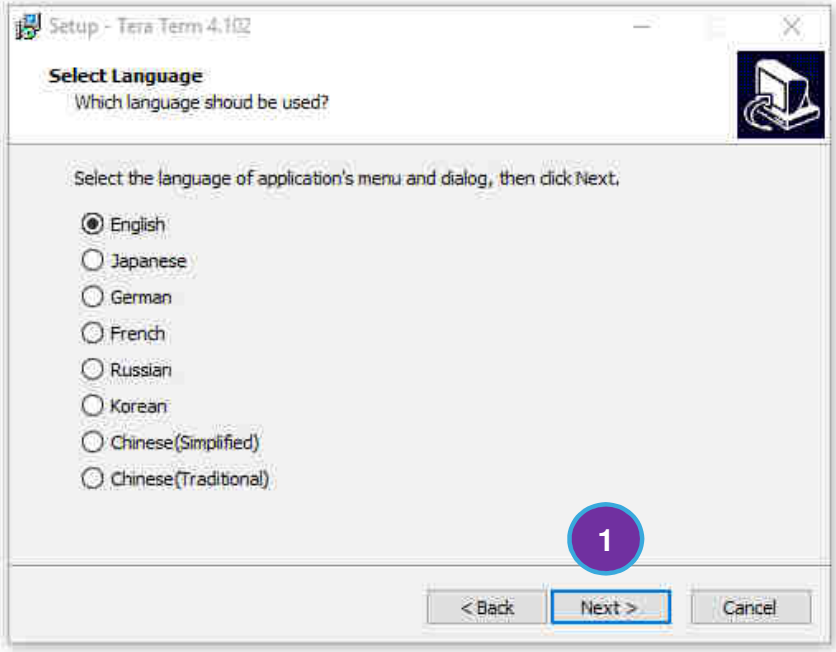
Tera Term Setup 4/9

1. Click on **Next**



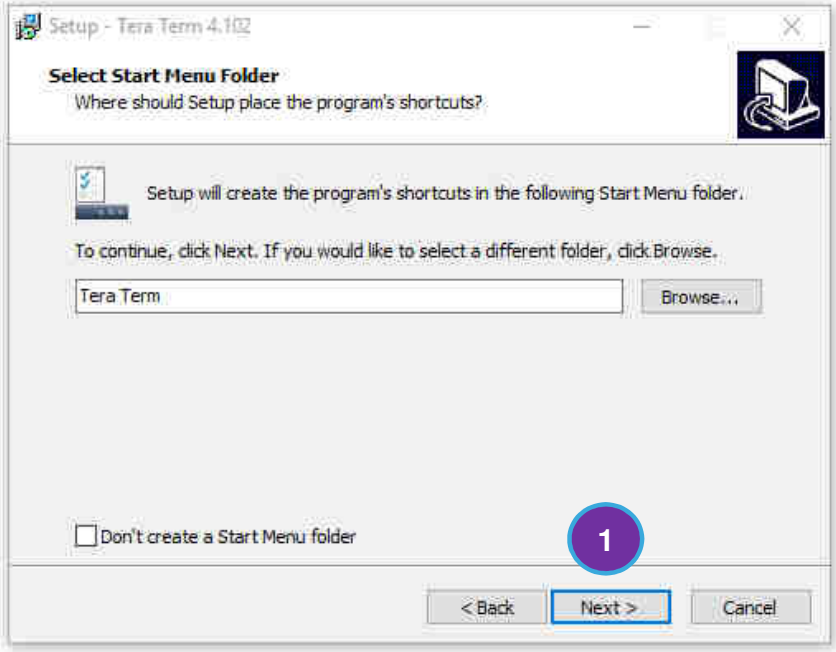
Tera Term Setup 5/9

1. Click on **Next**



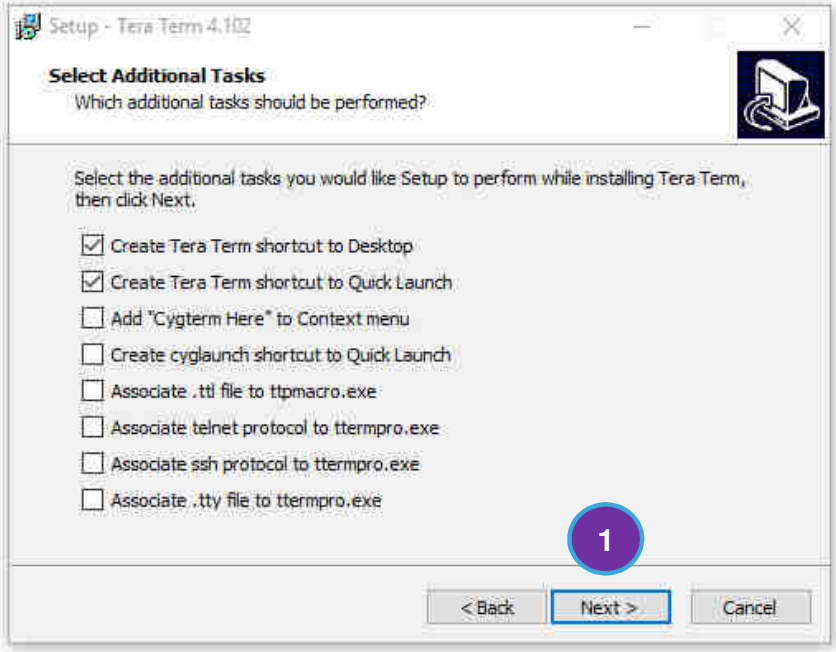
Tera Term Setup 6/9

1. Click on **Next**



Tera Term Setup 7/9

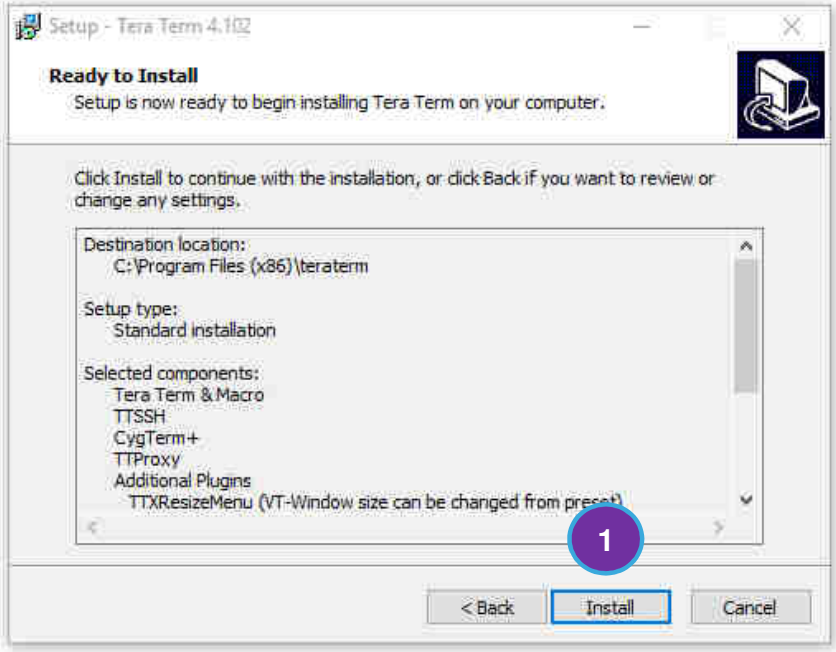
1. Click on **Next**



life.augmented

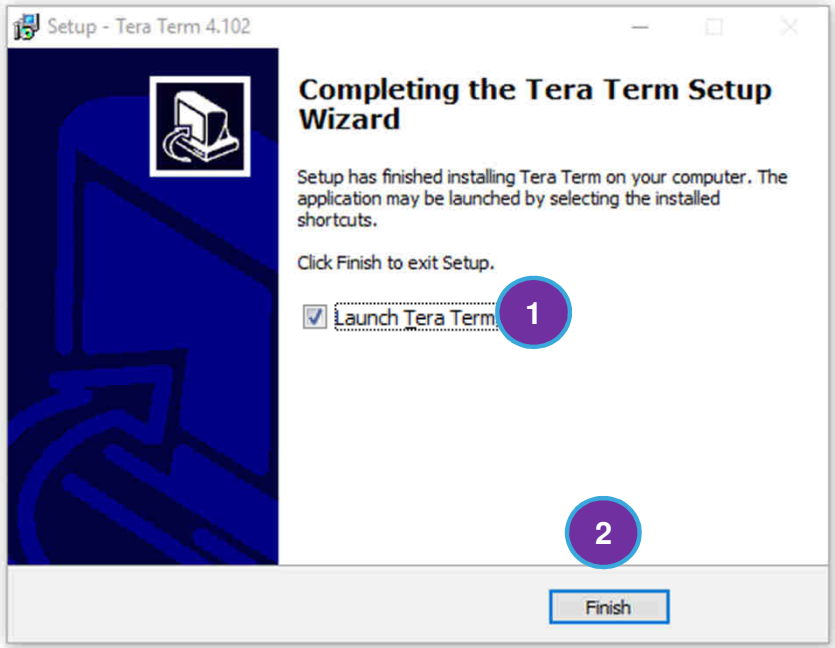
Tera Term Setup 8/9

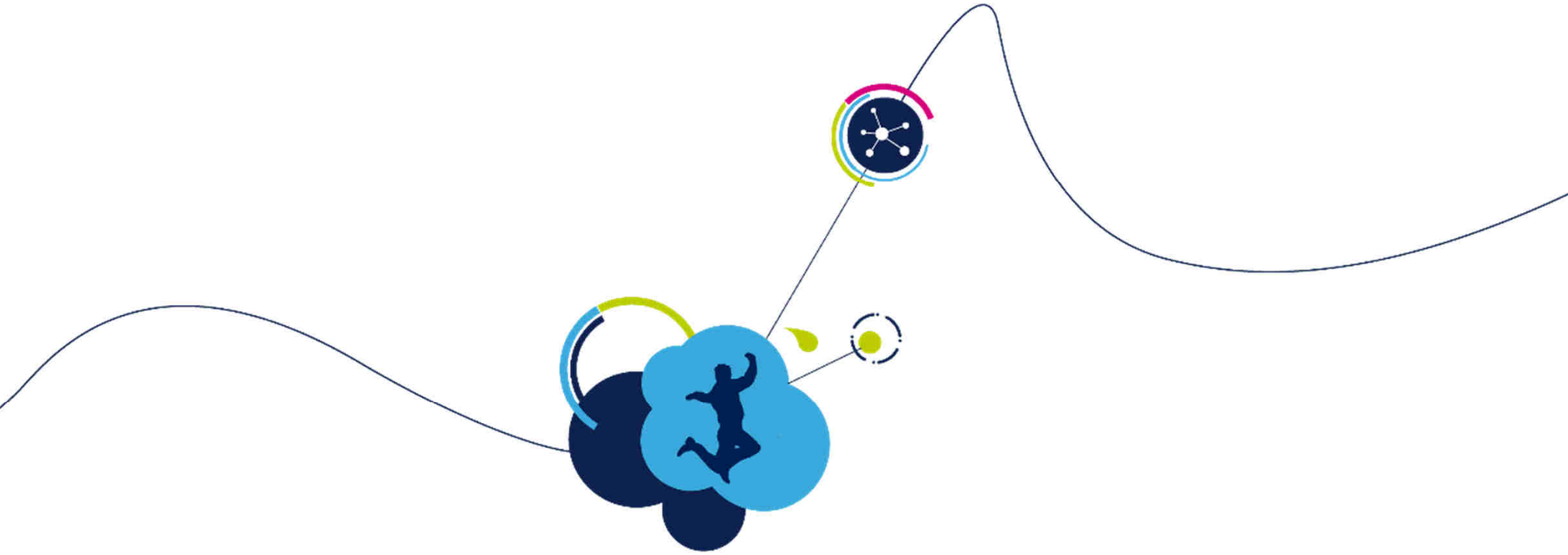
1. Click on **Install**



Tera Term Setup 9/9

1. Select **Launch Tera Term**
2. Click on **Finish**





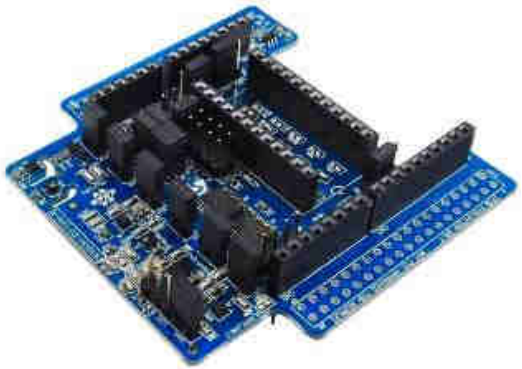
Preparing the Hardware

life.augmented

In Your Hands

NUCLEO-L476RG

Development board with STM32L476RG MCU, supports Arduino and ST morpho connectivity



X-NUCLEO-IKS01A3

Motion MEMS and environmental sensor evaluation board system

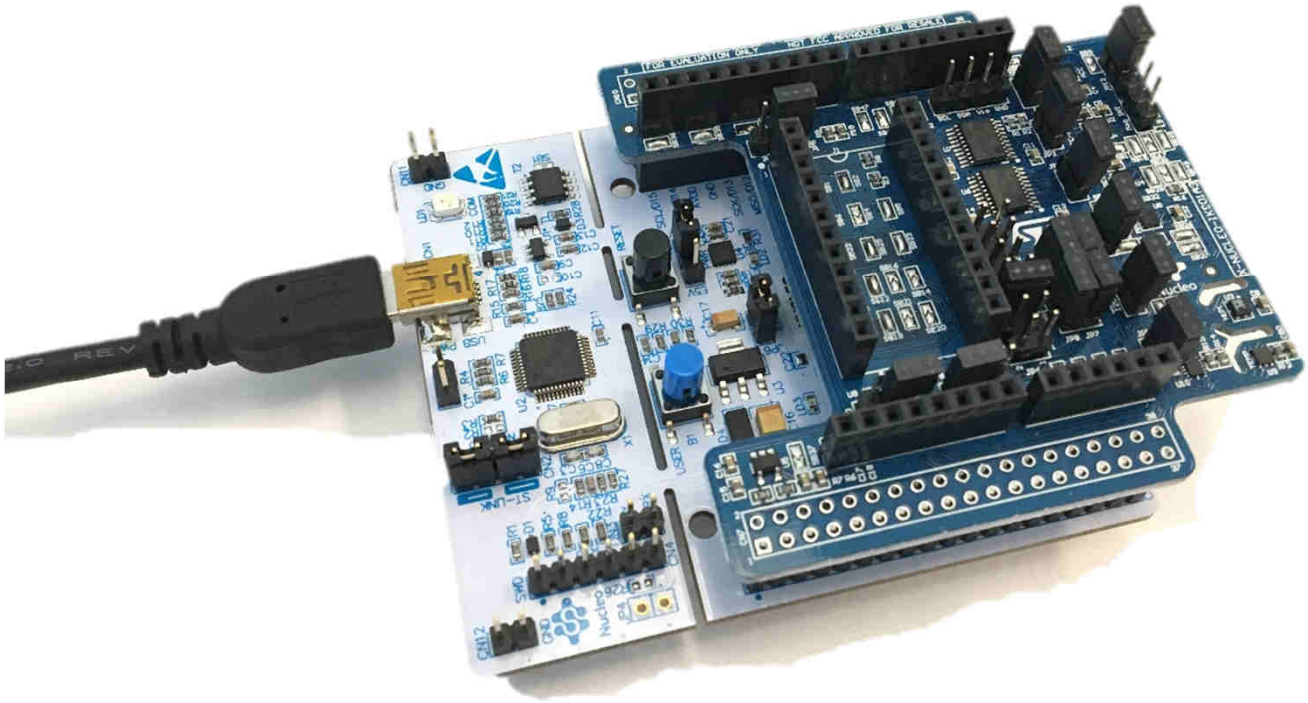
MINI USB CABLE

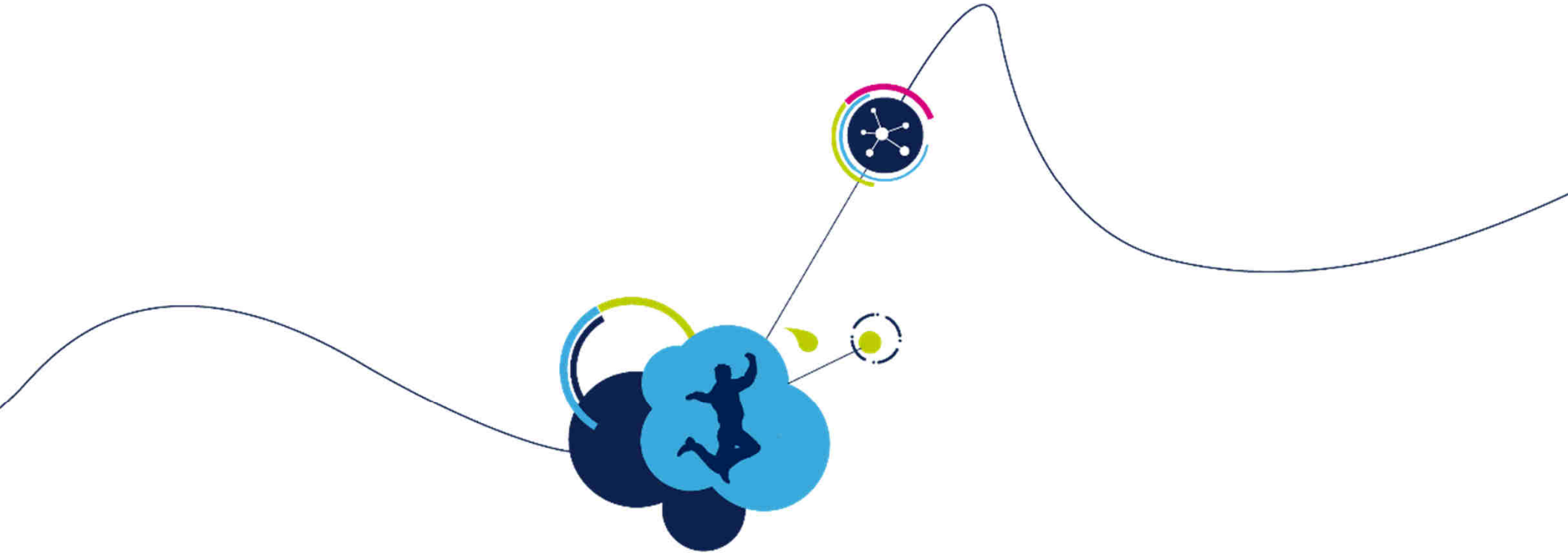
Enable USB communication to the laptop

life.augmented

Preparing the Hardware

1. Stack **NUCLEO-L476RG** and **X-NUCLEO-IKS001A3** and connect it to the PC





Tera Term Configuration

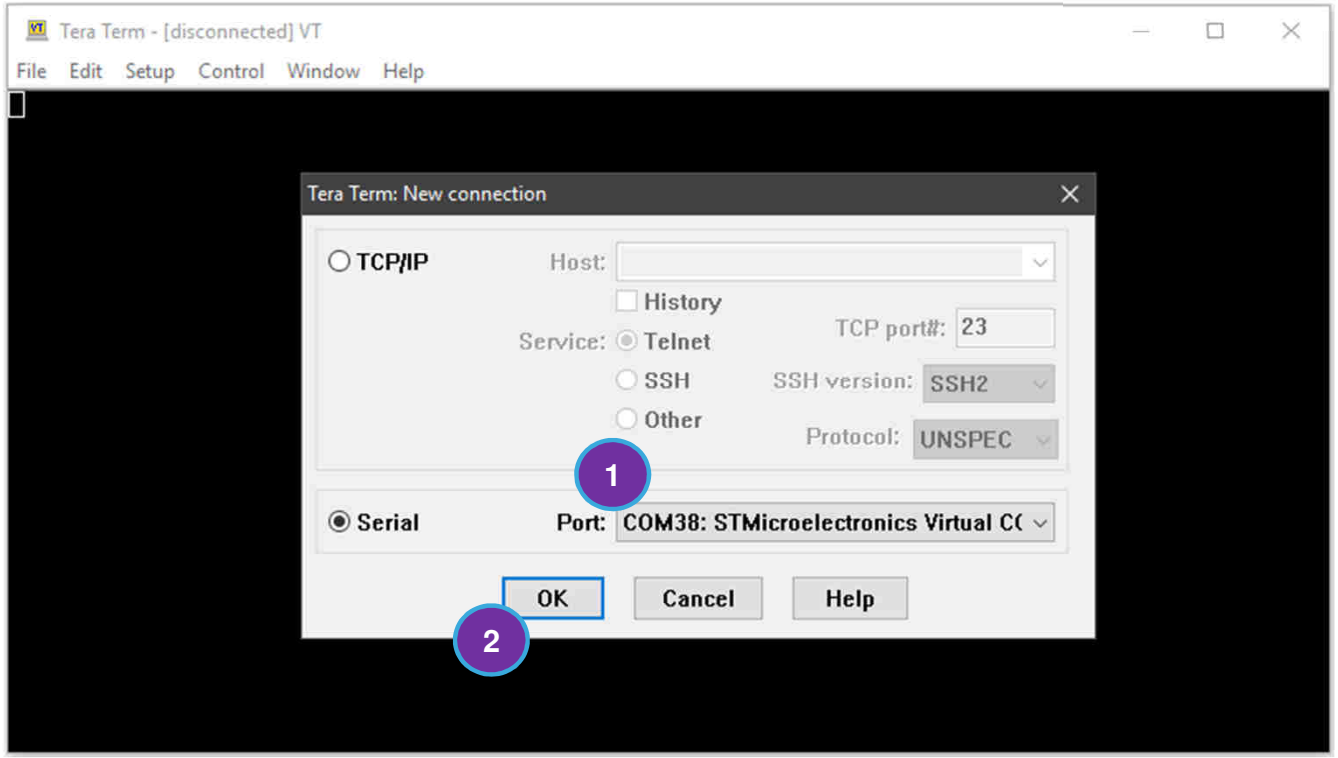
This can only be done when the board is connected via USB

life.augmented

Tera Term Configuration 1/5

Plug the board to the PC using the micro USB cable provided

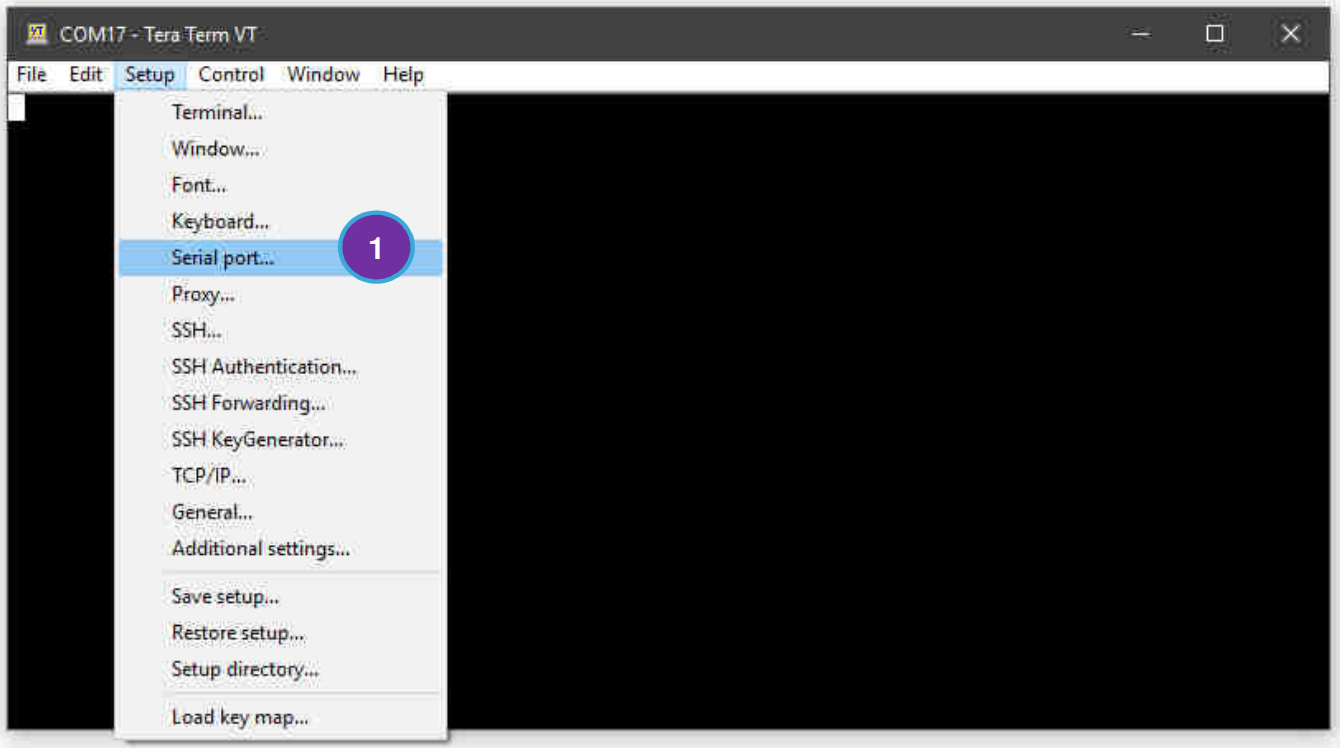
1. Select the **STMicroelectronics Virtual COM Port**
2. Click **OK**



life.augmented

Tera Term Configuration 2/5

1. Click **Setup** -> **Serial port…**

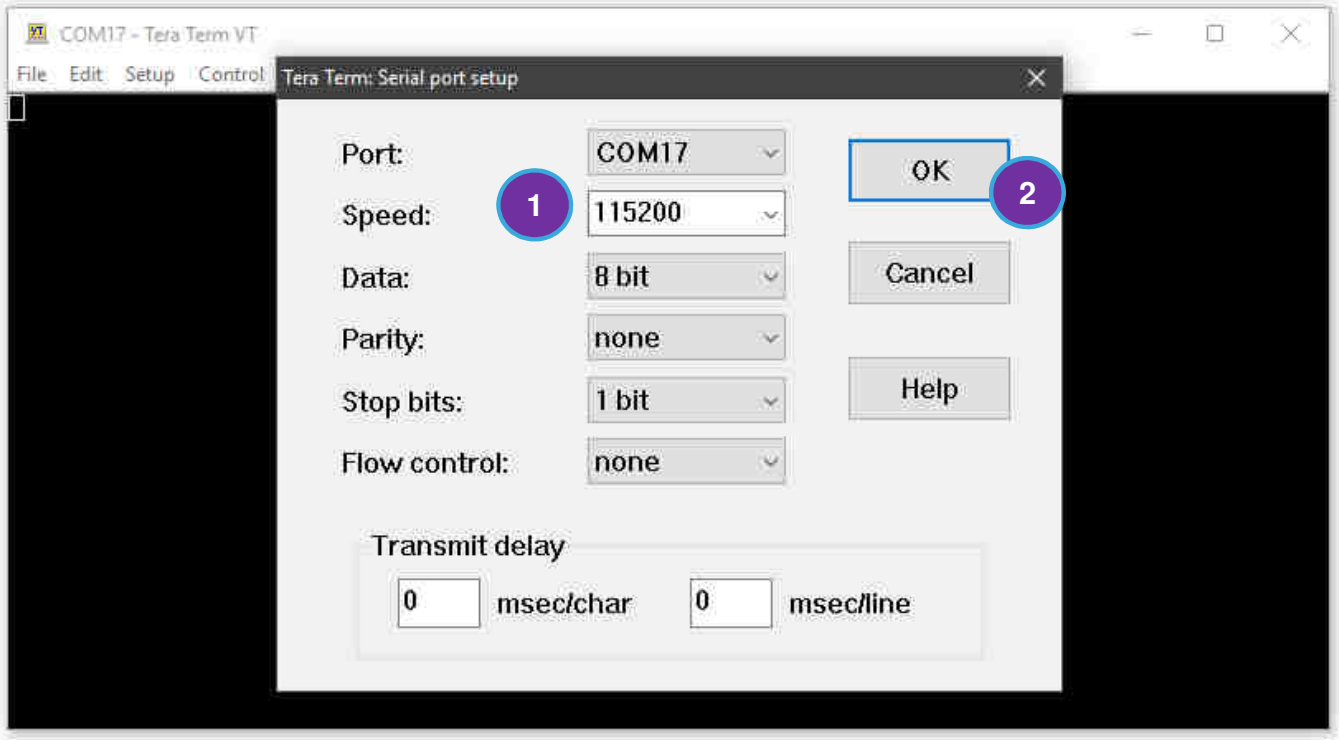


life.augmented

Tera Term Configuration 3/5

1. Set the following:
   Baud Rate: **115200**
   Data: **8 bit**
   Parity: **none**
   Stop: **1 bit**
   Flow control: **none**

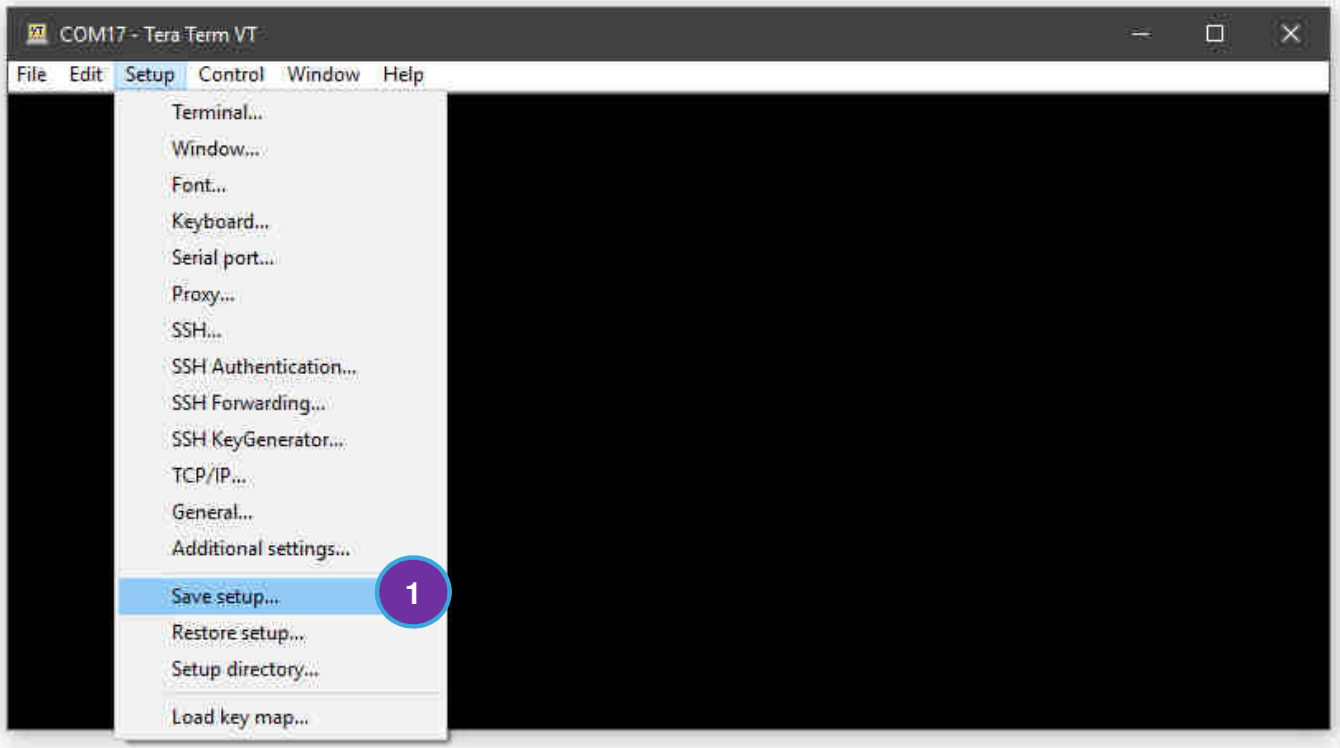
2. Click **OK**



life.augmented

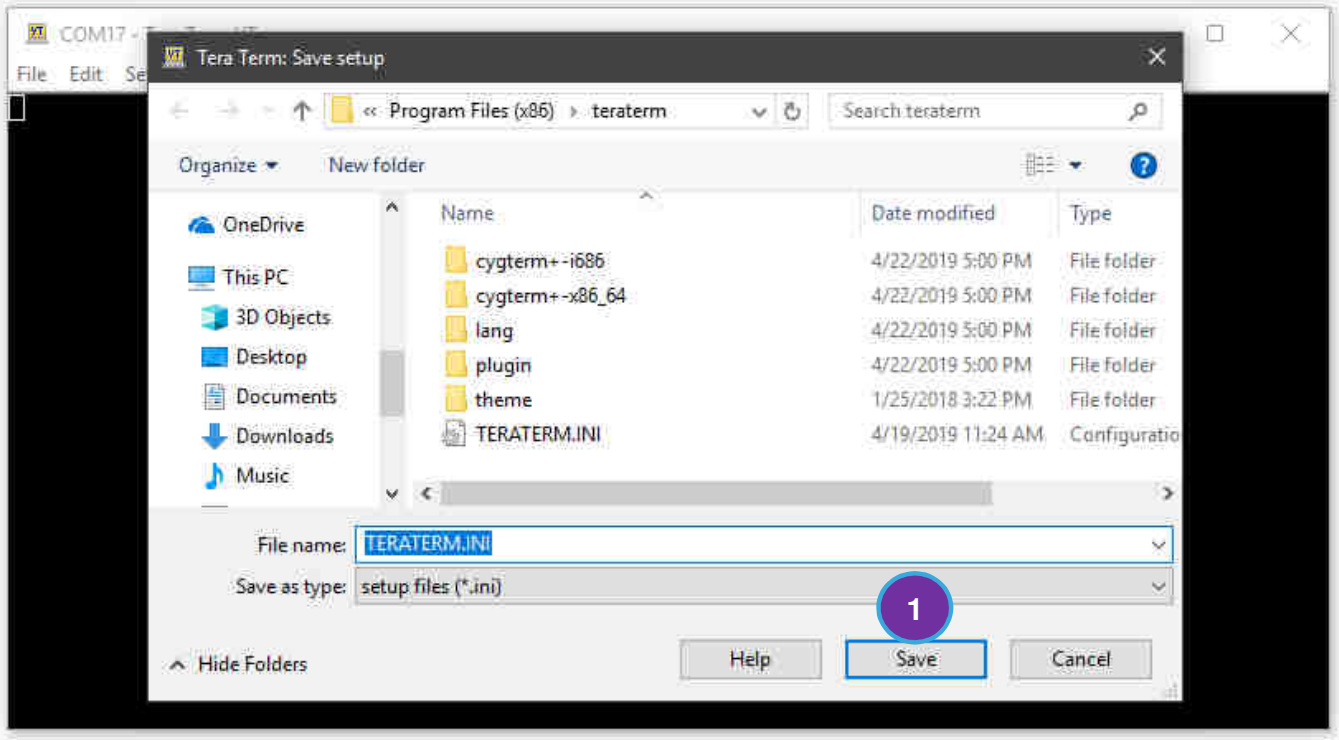
Tera Term Configuration 4/5

1. Click **Setup** -> **Save setup…**



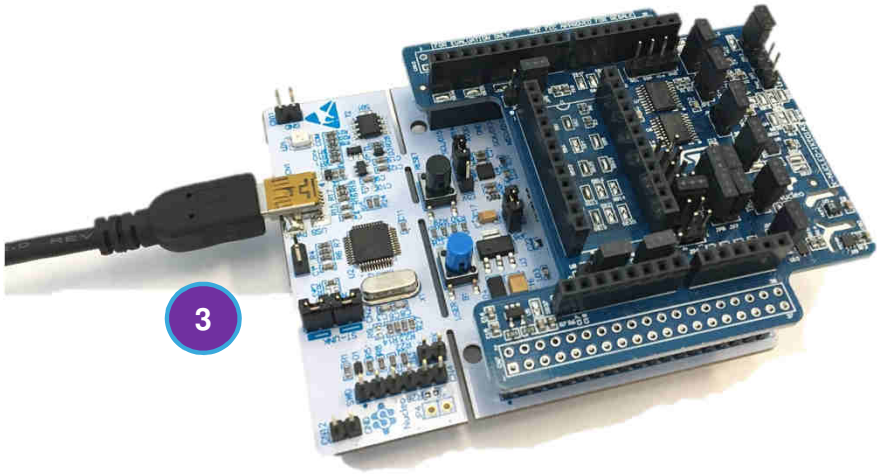
Tera Term Configuration 5/5

1. Click **Save**



life.augmented

Hardware and Software Check



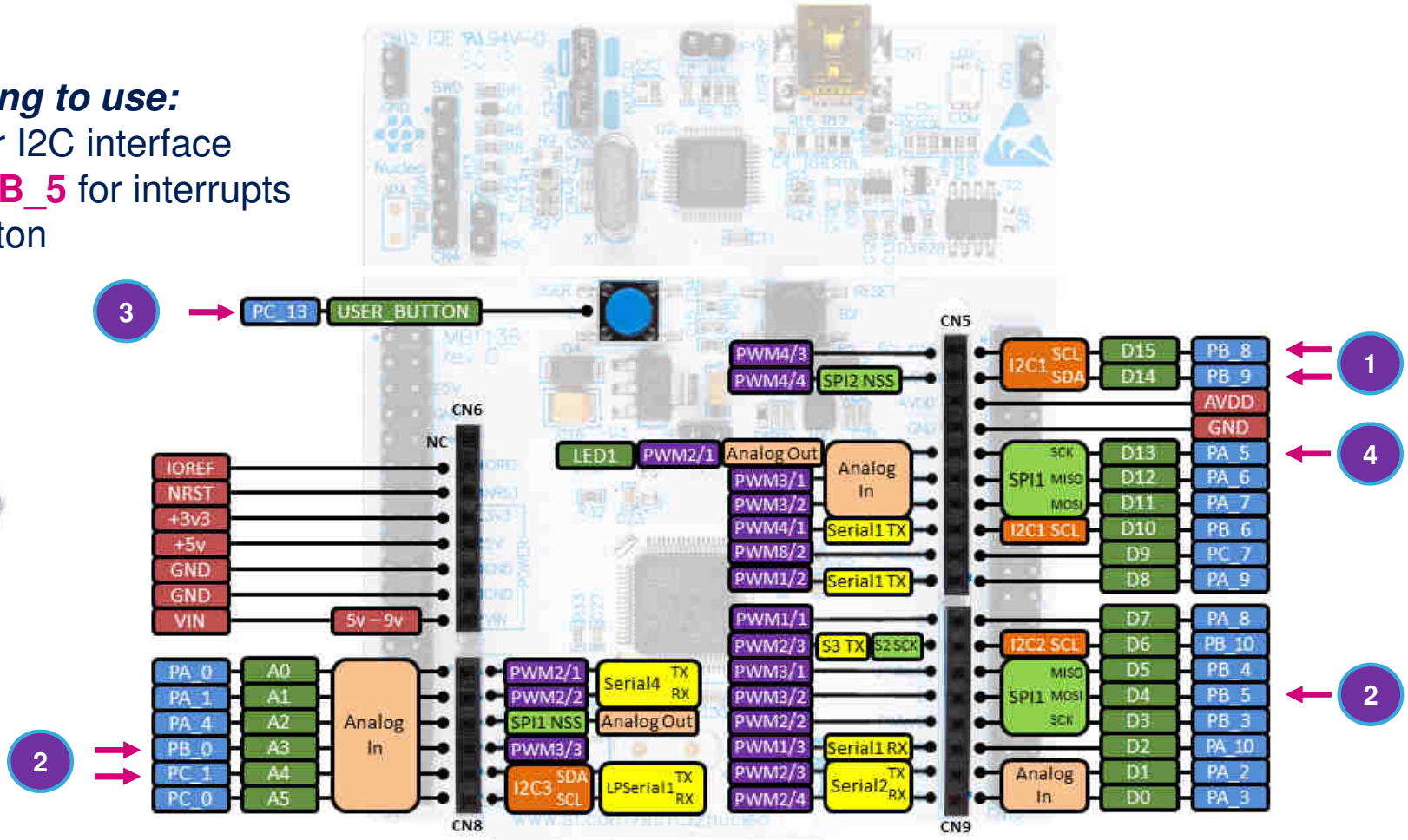
- At this point you should all have:
  1. *Installed* STM32Cube libraries
  2. *Installed* STM32CubeIDE
  3. *Stacked* the Nucleo-L476 with X-Nucleo-IKS01A3 and plug it to the PC
  4. *Installed and configured* Tera Term

life.augmented

NUCLEO-L476RG Header Pinout

In our labs we are going to use:

1. **PB_8** and **PB_9** for I2C interface
2. **PB_0**, **PC_1** and **PB_5** for interrupts
3. **PC_13** for user button
4. **PA_5** for the LED

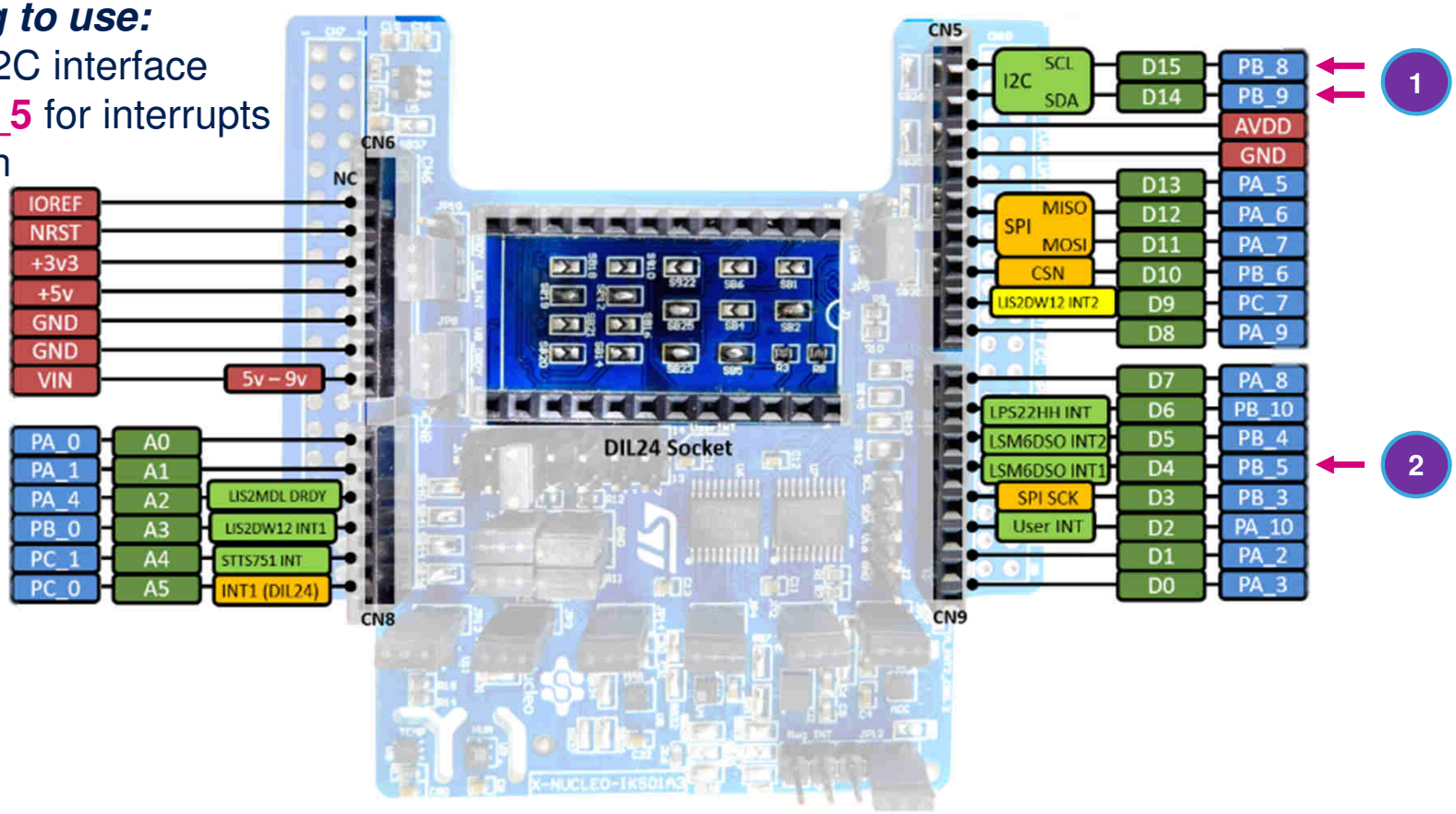


life.augmented

X-NUCLEO-IKS01A3 Header Pinout

In our labs we are going to use:

1. **PB_8** and **PB_9** for I2C interface
2. **PB_0**, **PC_1** and **PB_5** for interrupts
3. **PC_13** for user button
4. **PA_5** for the LED



life.augmented

LAB0

Goals:

- Configure a new project using X-CUBE-MEMS1
- Configure all sensors and read data in polling mode
- Visualizing sensor data on Tera Term

life.augmented

Leading Sensors for IoT

Accelerometer & Gyroscope **6-Axis IMU**

LSM6DSO

Wearables, Smartphones, IoT, AR/VR

- Best in class for Power/Noise
- Embedded Digital Features including: New Pedo, FSM
- High speed I3C interface
- Real dual core

Ultra Low Power **Accelerometer**

LIS2DW12

IoT, wearable, anti-tampering, security

- Flexibility Power Consumption vs. Noise
- Ecosystem (SW, libraries, ref design, Nucleo boards …)

High Accuracy **Pressure Sensor**

LPS22HH

Compact, low power, water resistant

- LPS22HH: high accuracy,
- Low Power Consumption
- Skip One Point Cal. post Soldering

Magnetometer **Compass**

LIS2MDL

Accuracy, Temperature Stability

- Power Consumption
- Thermal Stability & Precision
- The LIS2MDL has a magnetic field dynamic range of ±50 gauss.

Local **Temperature** sensor

STTS751

Low Voltage, Low Power

- Ultra low power with one shot mode for superior power savings
- SMBus 2.0 compliant supporting ALERT command
- Very small package

Humidity and Temperature Combo

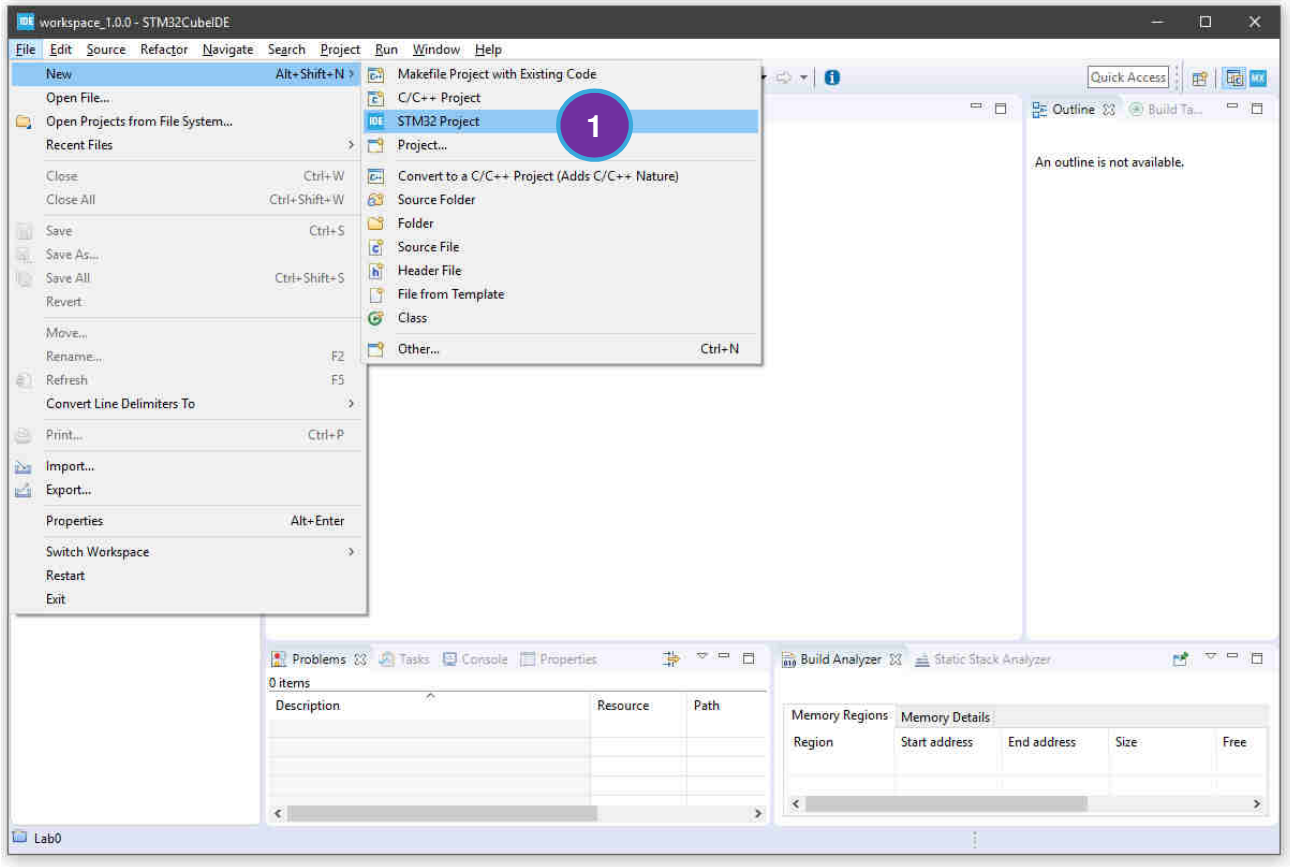
HTS221

Compact Low Power

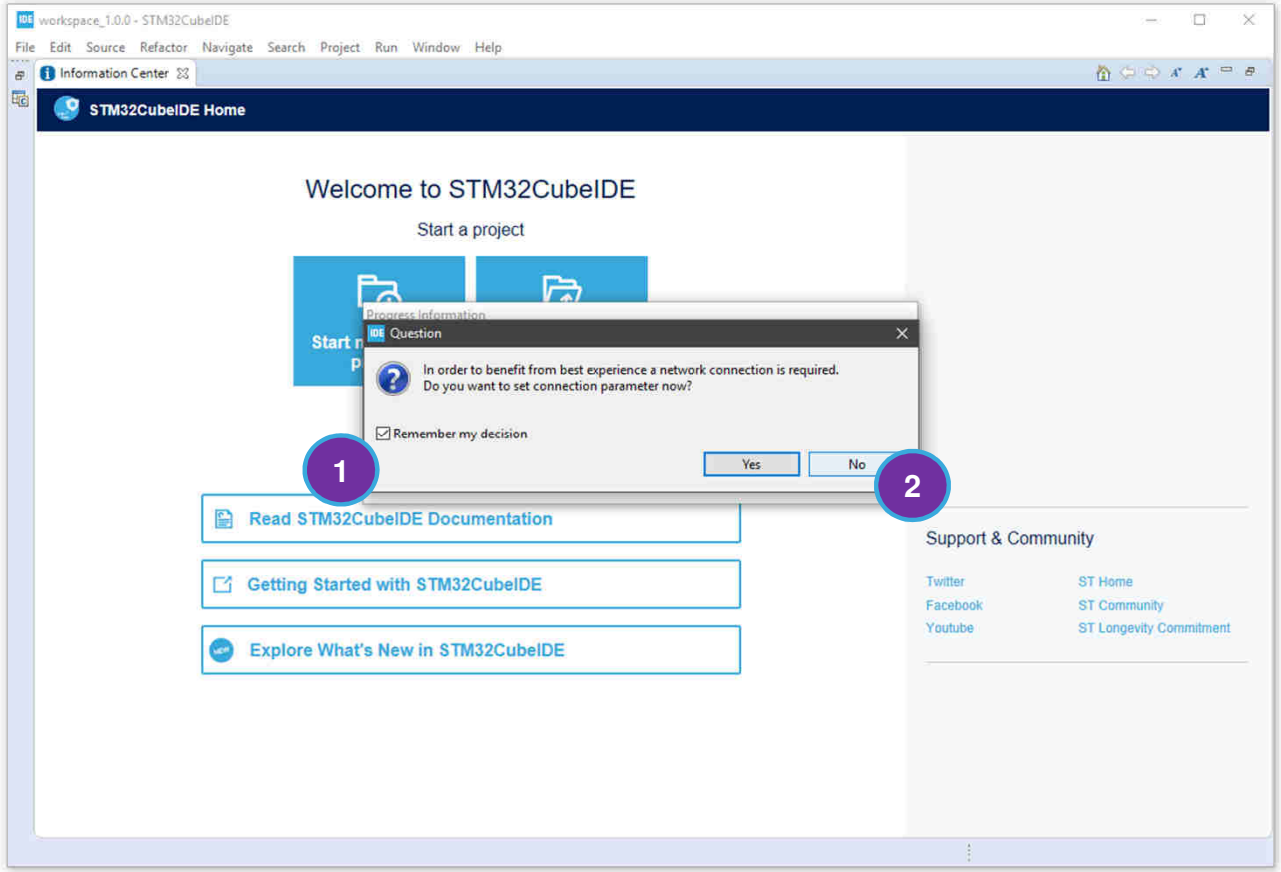
- Direct H and T data readout
- Low current consumption (3.5uA)
- Industrial temperature operating range

life.augmented

1. Click on **File > New > STM32 Project**



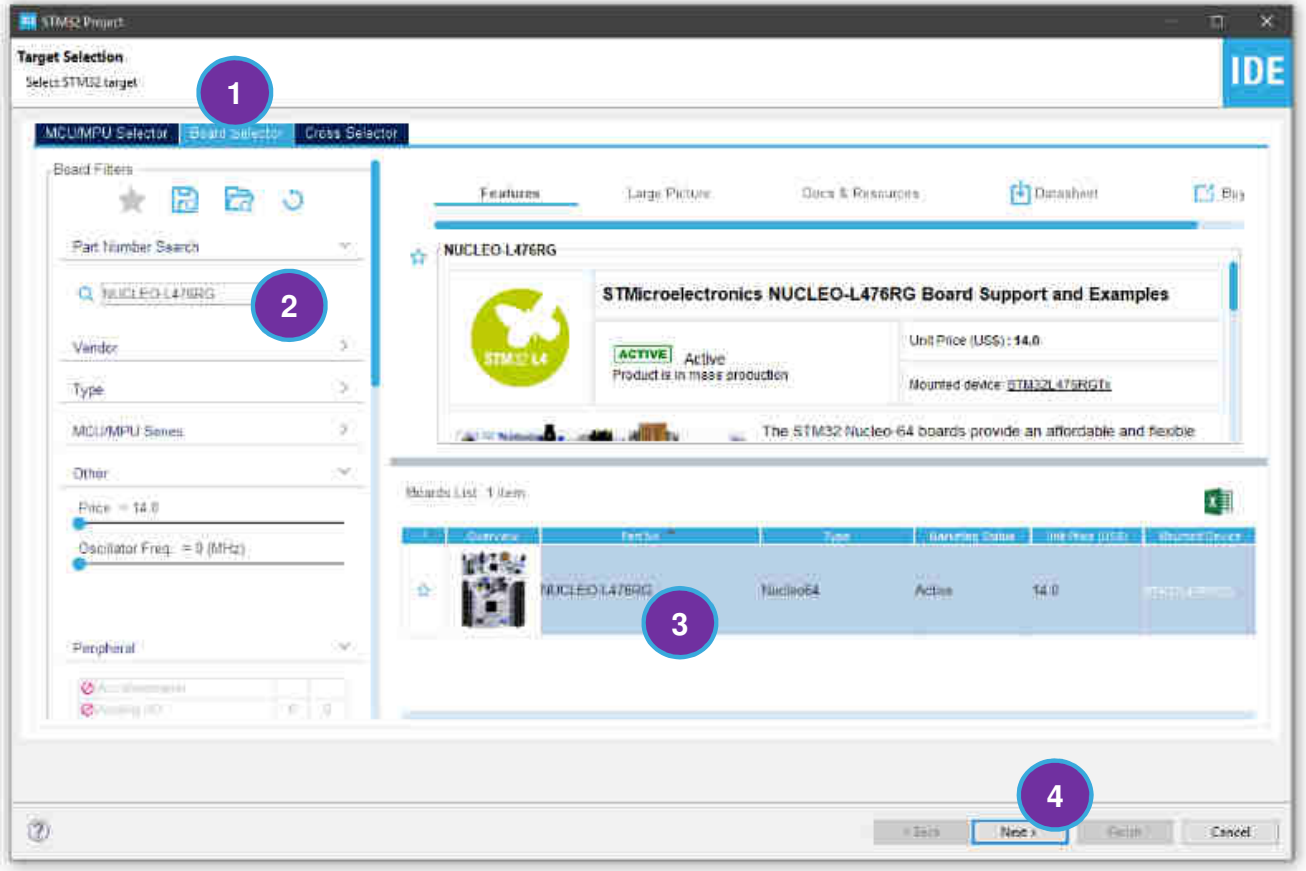
1. Check **Remember my decision***
2. Click on **No**



* It is always possible to reconfigure internet connection parameter in **Window > Preferences** then in **General > Network Connections**

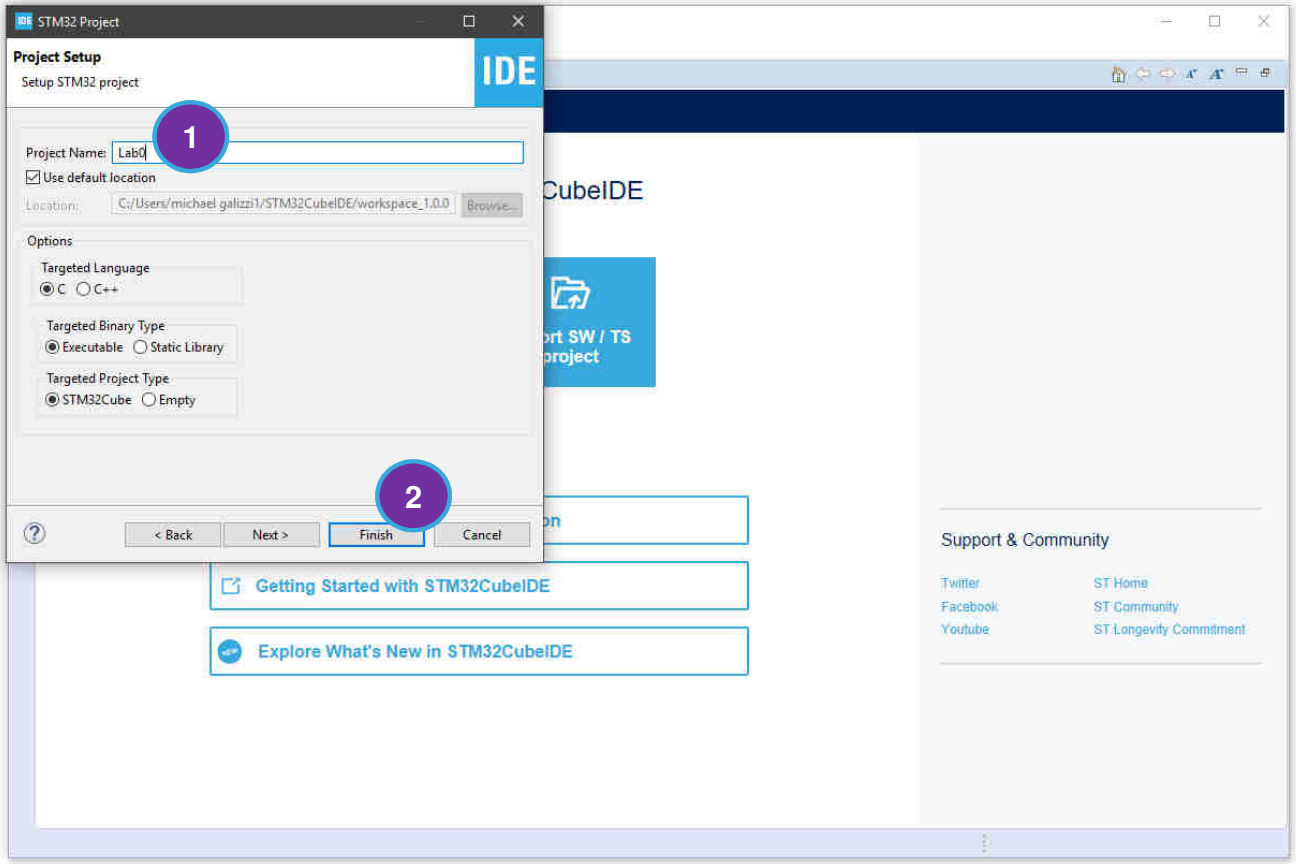
life.augmented

1. Click on **Board Selector**
2. Type **NUCLEO-L476RG**
3. Click on the board
4. Click **Next >**



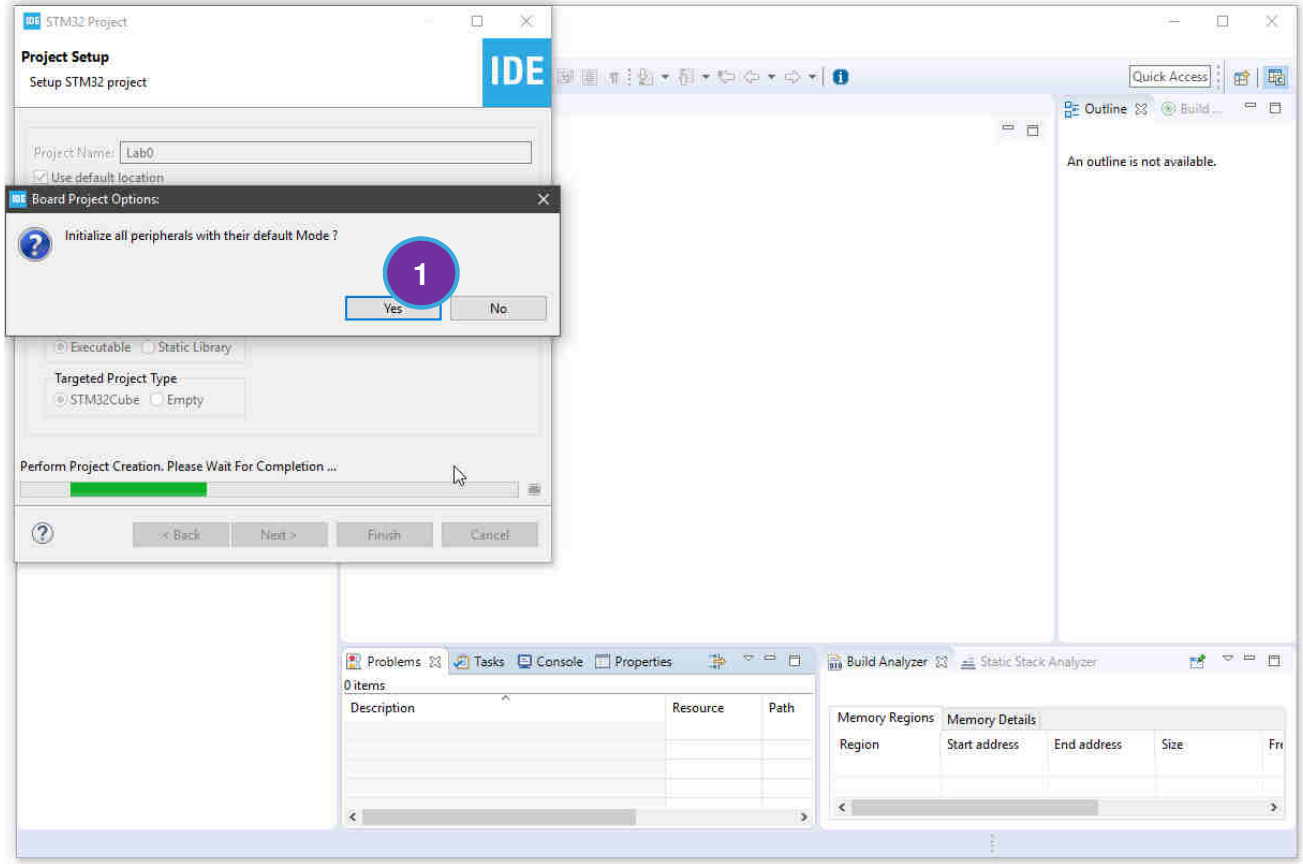
life.augmented

1. Project Name **Lab0**
2. Click **Finish**



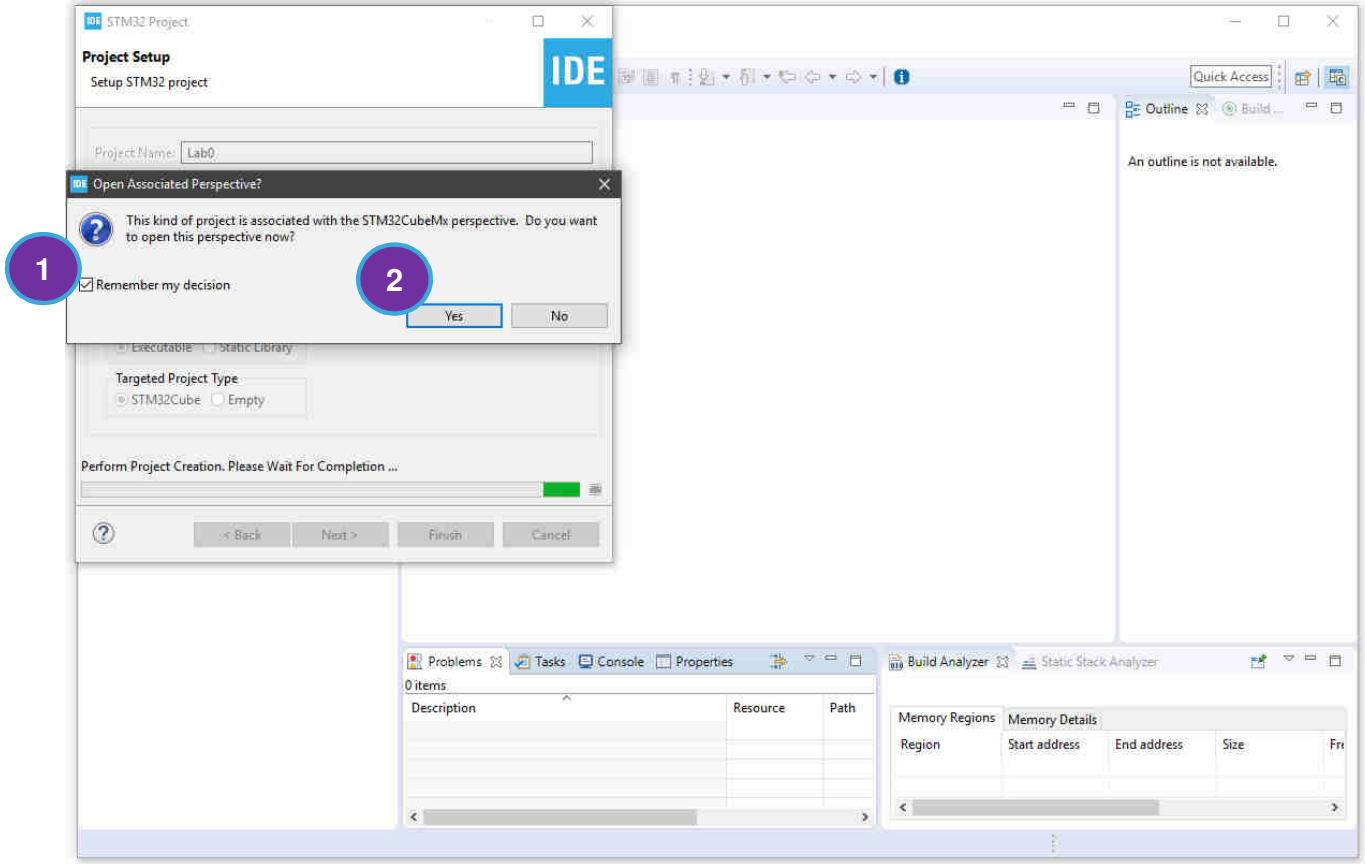
life.augmented

1. Click **Yes** to init peripherals in default mode



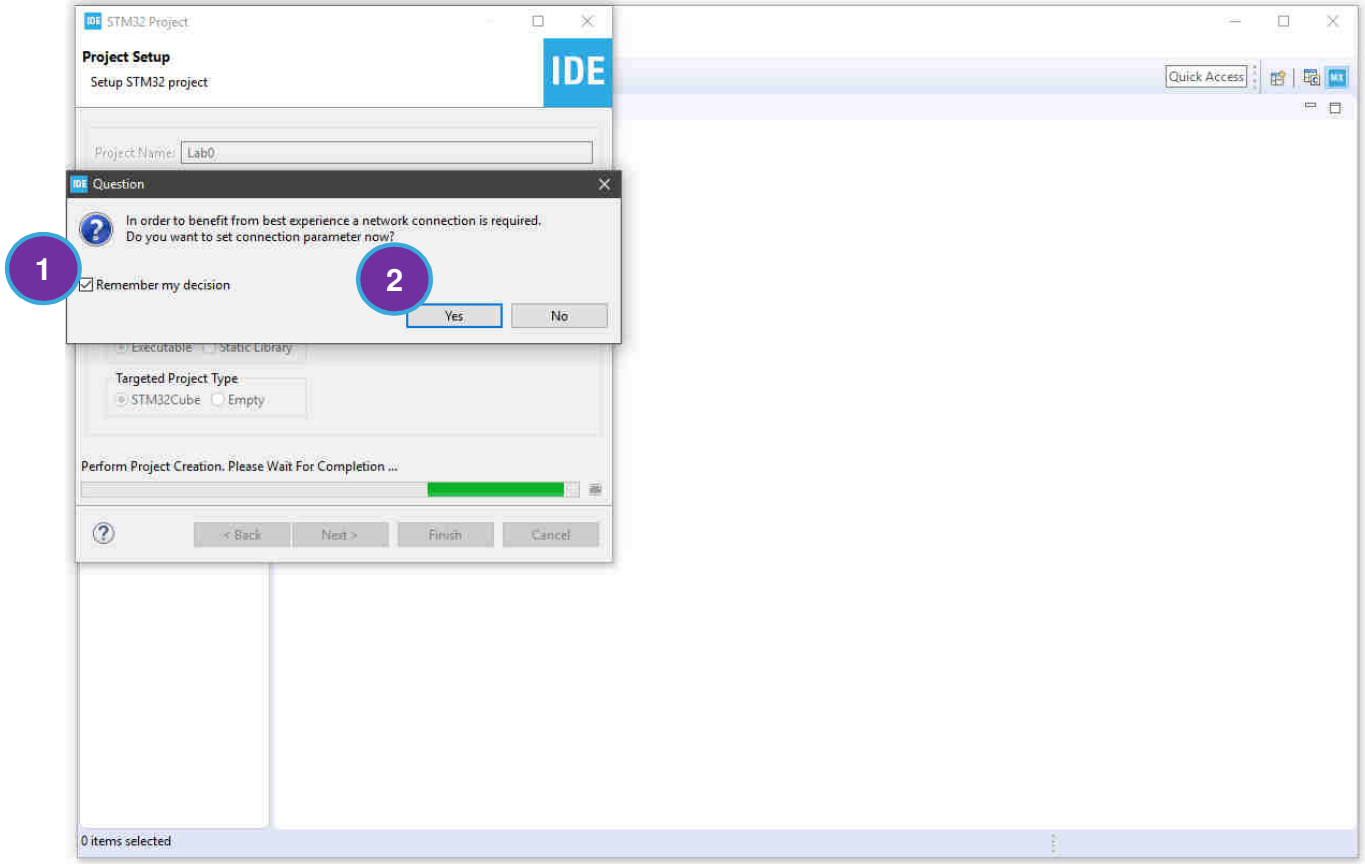
life.augmented

1. Check **Remember my decision**
2. Click **Yes**



life.augmented

1. Check **Remember my decision**
2. Click **No** *

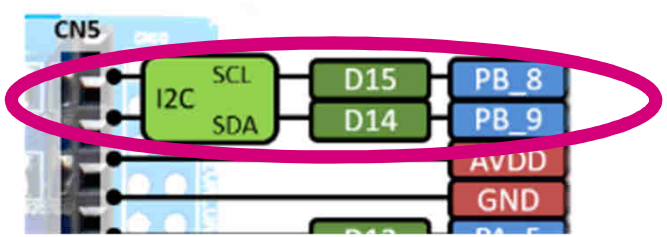


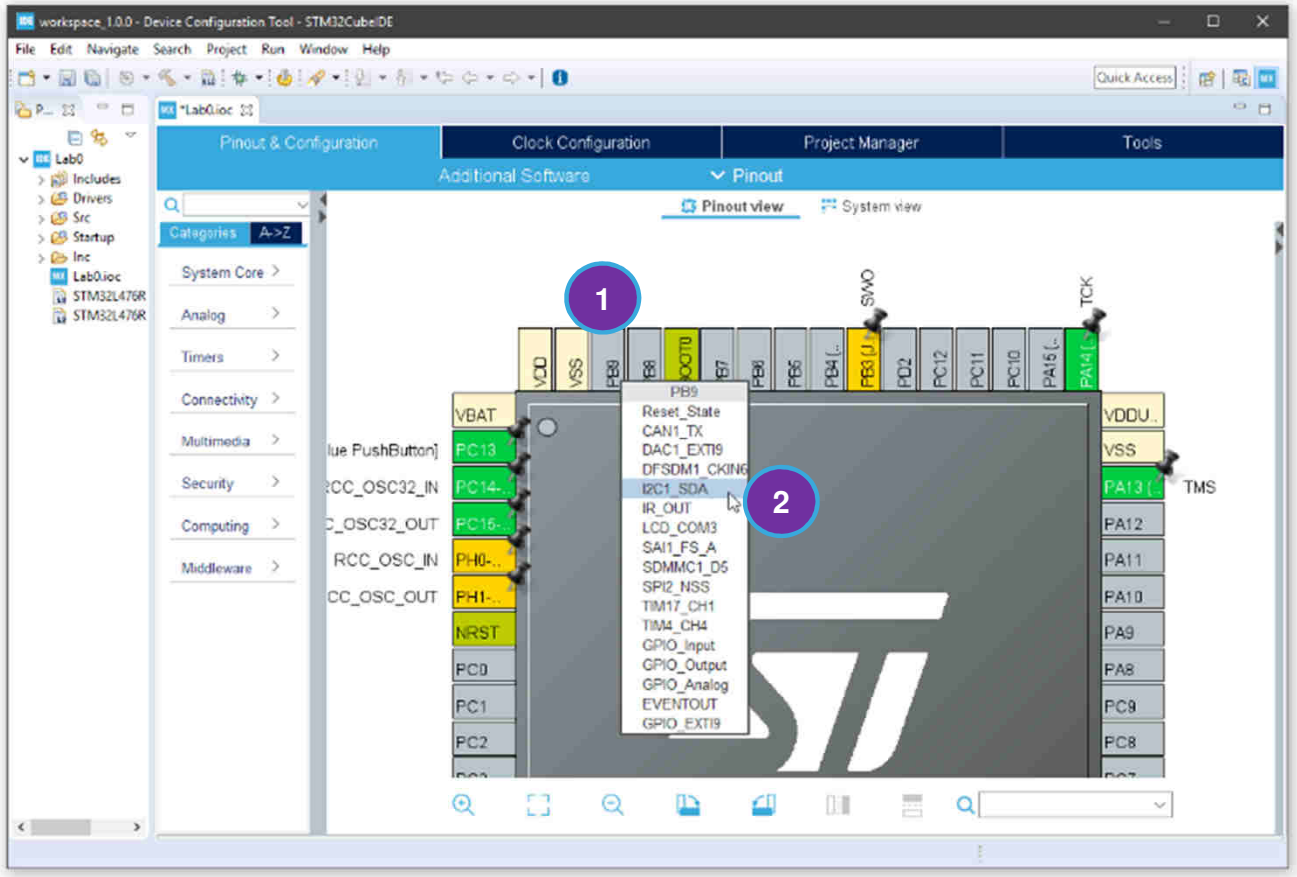
* We are using all the tools in offline mode, so that internet connection is not strictly needed

life.augmented

Lab0 – Configure the I2C Bus

1. Left Click on **PB9 pin**
2. Select **I2C1_SDA**

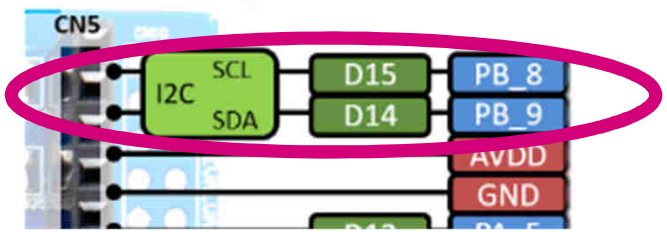


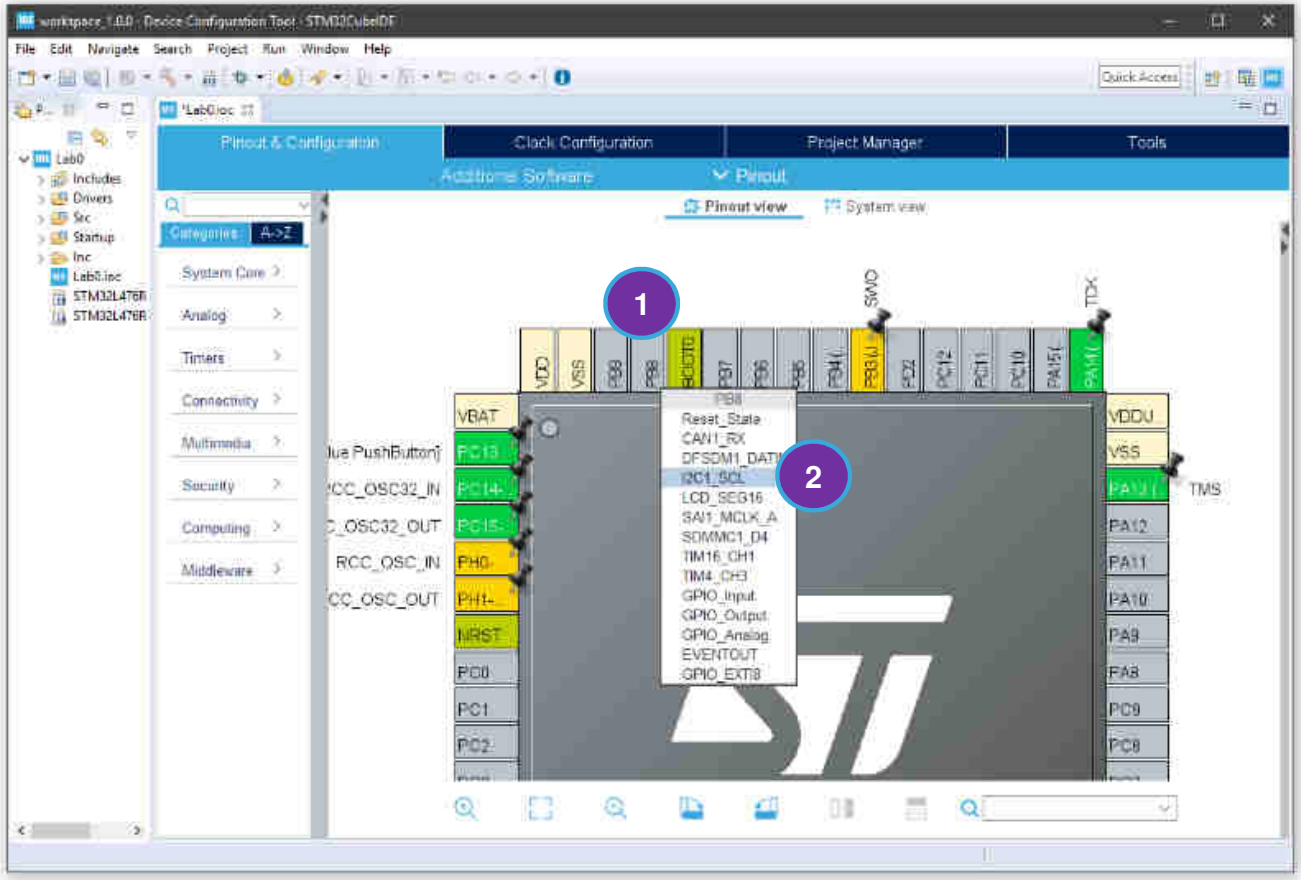


life.augmented

Lab0 – Configure the I2C Bus

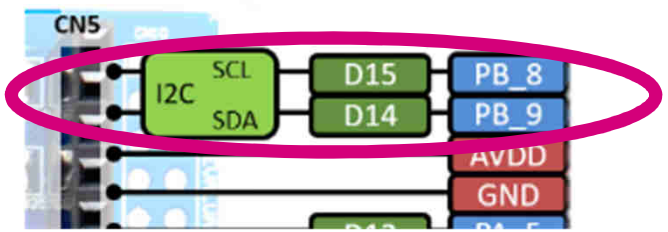
1. Left Click on **PB8 pin**
2. Select **I2C1_SCL**

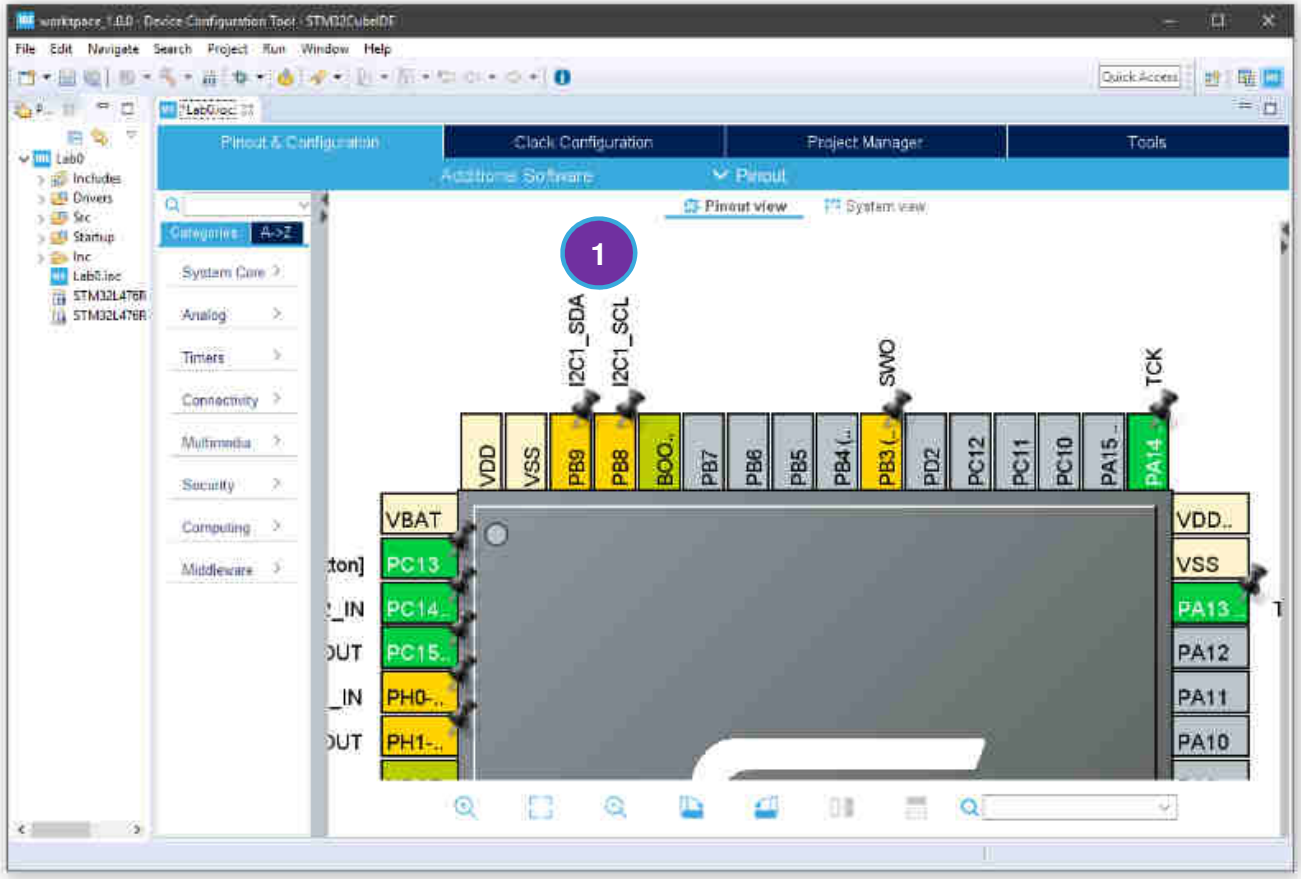




Lab0 – Configure the I2C Bus

1. At this point I2C1 pins PB9 and PB8 should appear in yellow (selected but not configured)

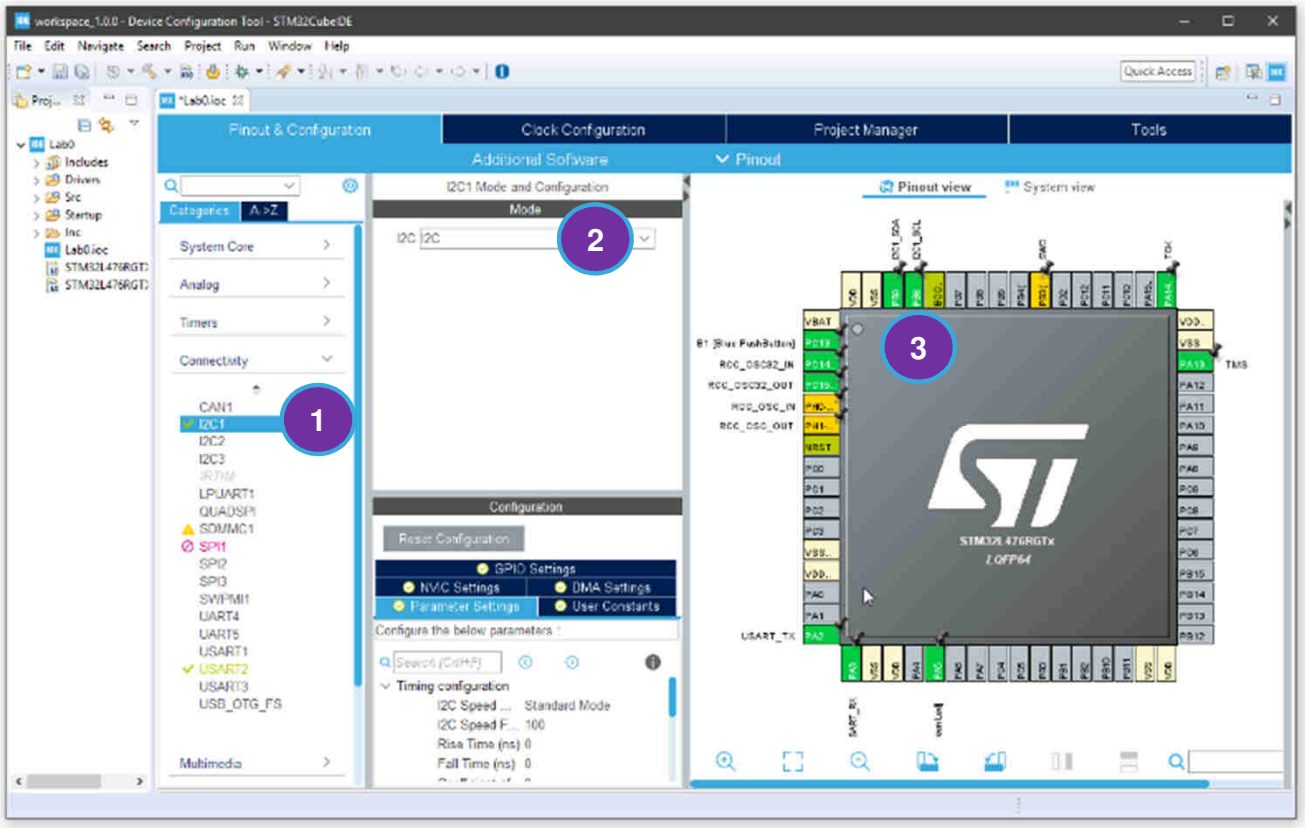




life.augmented

Lab0 – Configure the I2C Bus

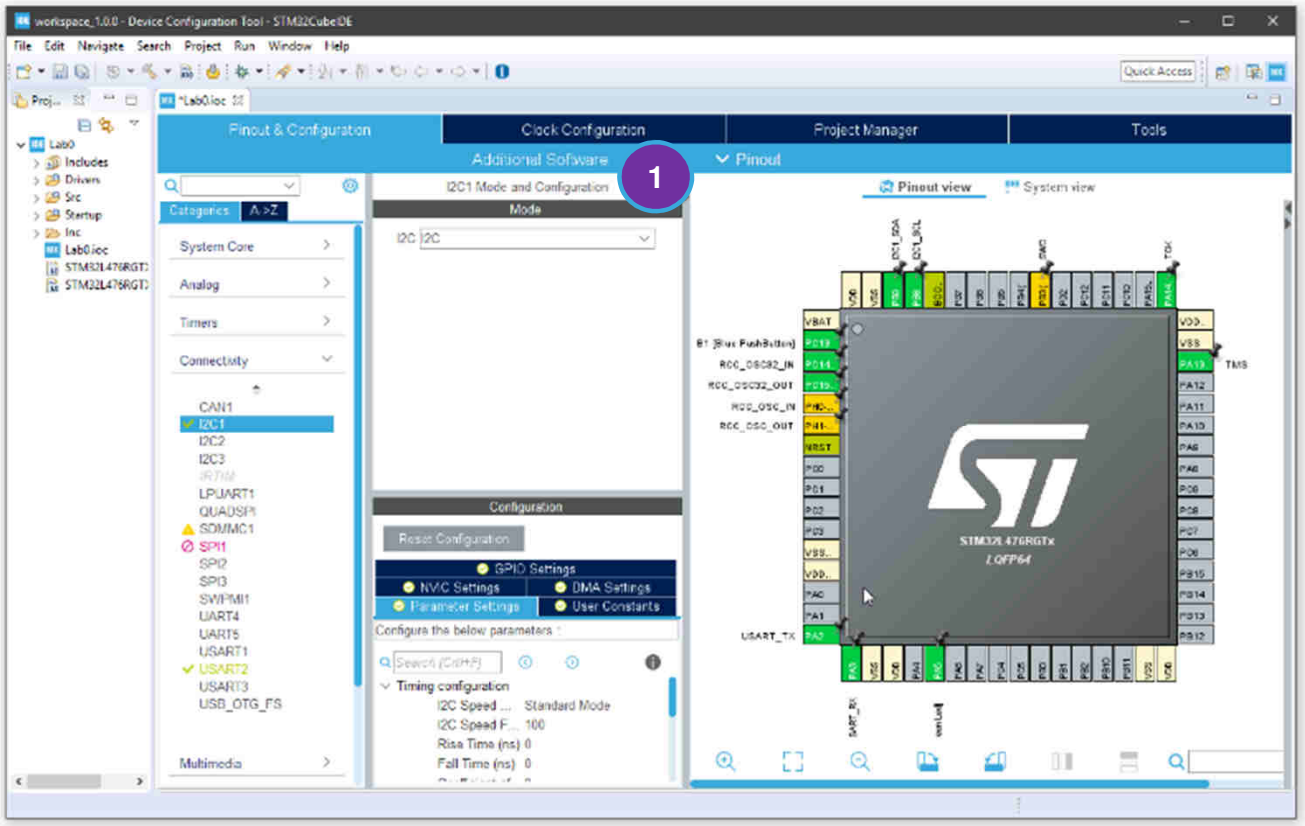
1. Expand *Connectivity* tab and check **I2C1**
2. Select **I2C** in *I2C1 Mode and Configuration*
3. PB8 and PB9 should now become green



life.augmented

Lab0 – Select the MEMS Library

1. Click on
   Additional Software



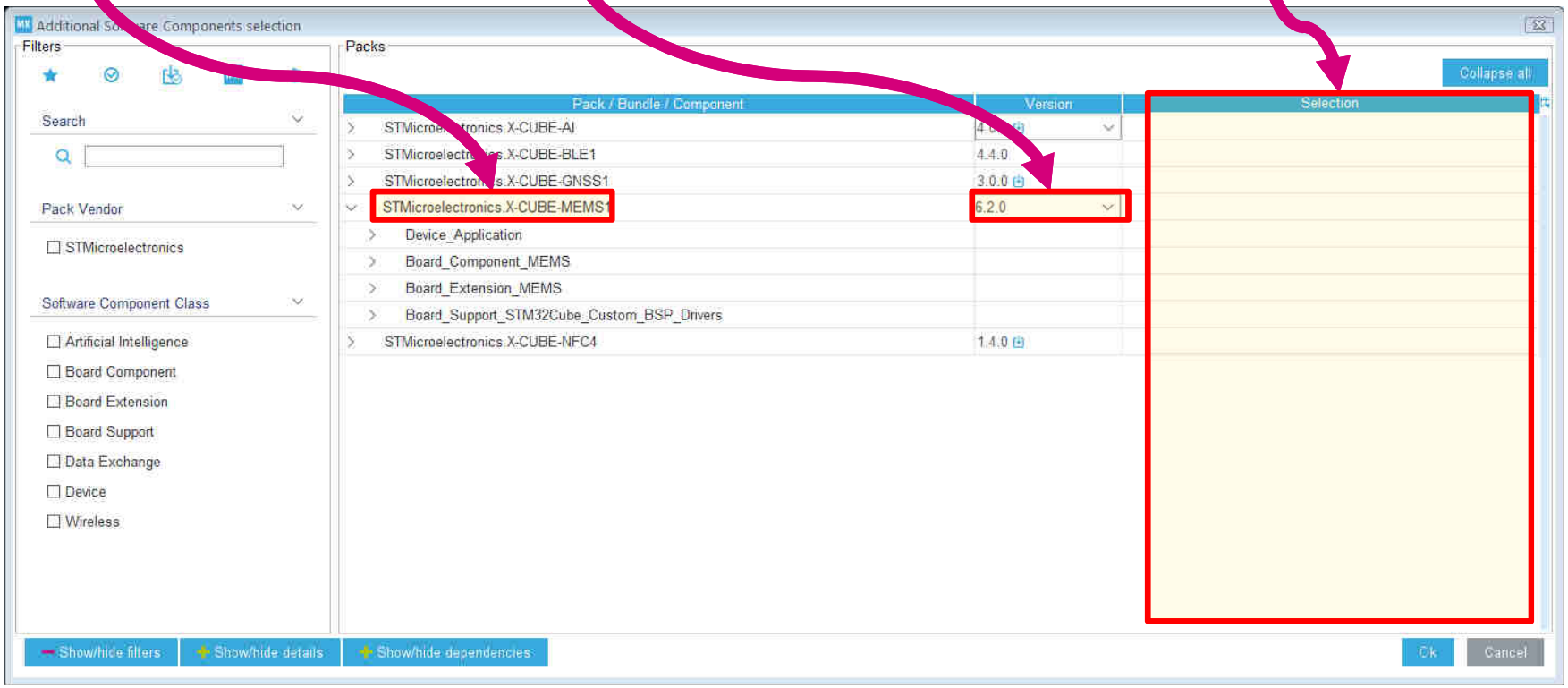
life.augmented

Additional Software Intro

X-CUBE-MEMS1/… will be the Pack/Bundle we are going to use in this hands-on

If multiple X-Cube packs, use latest version: here version **6.2.0**

If needed, enlarge the column **Selection** to see the options available



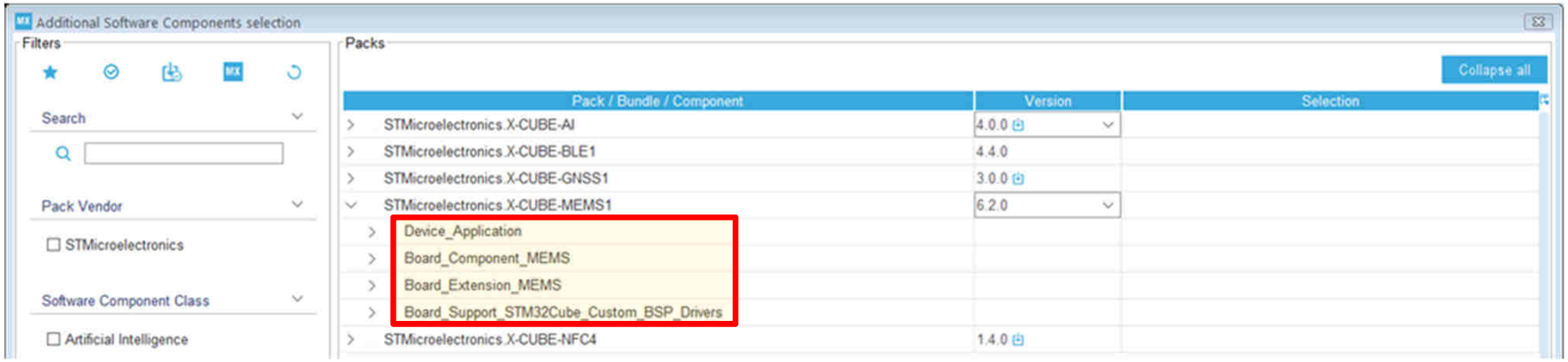
life.augmented

Additional Software Intro

There are two ways to use the X-CUBE-MEMS1 Pack/Bundle:

- If you are using the X-NUCLEO from STMicroelectronics
  1. Select the high level **Device** application. Example: IKS01A3_DataLogTerminal
  2. Select the **Board Extension** that you are using. Example: IKS01A3

- If you are using a custom expansion board
  1. Select the high level **Device** application. Example: DataLogTerminal
  2. Select the middle level **Board Support Pack (BSP)**. Example: Custom/MOTION_SENSOR
  3. Select the low level **Board Components**. Example: AccGyr/LSM6DSO on I2C



life.augmented

Additional Software Intro

There are two ways to use the X-CUBE-MEMS1 Pack/Bundle:

Lab0 - Lab 5

- If you are using the X-NUCLEO from STMicroelectronics
  1. Select the high level **Device** application. Example: IKS01A3_DataLogTerminal
  2. Select the **Board Extension** that you are using. Example: IKS01A3

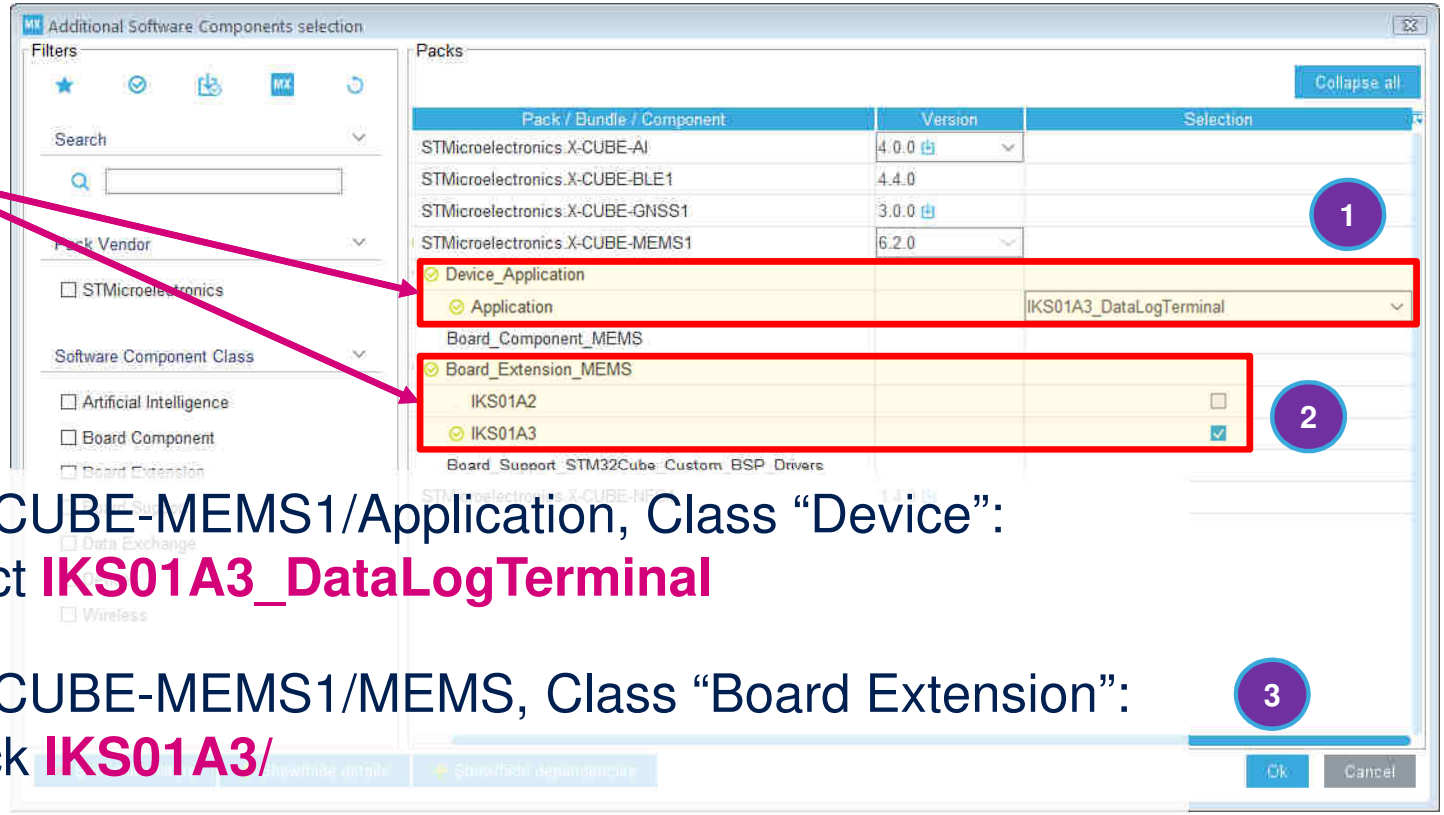
Lab6

- If you are using a custom expansion board
  1. Select the high level **Device** application. Example: DataLogTerminal
  2. Select the middle level **Board Support Pack (BSP)**. Example: Custom/MOTION_SENSOR
  3. Select the low level **Board Components**. Example: AccGyr/LSM6DSO on I2C

life.augmented

Lab0 – Select the MEMS Library

Click to expand tree



1. In X-CUBE-MEMS1/Application, Class “Device”: Select **IKS01A3_DataLogTerminal**
2. In X-CUBE-MEMS1/MEMS, Class “Board Extension”: Check **IKS01A3/**
3. Click **OK**

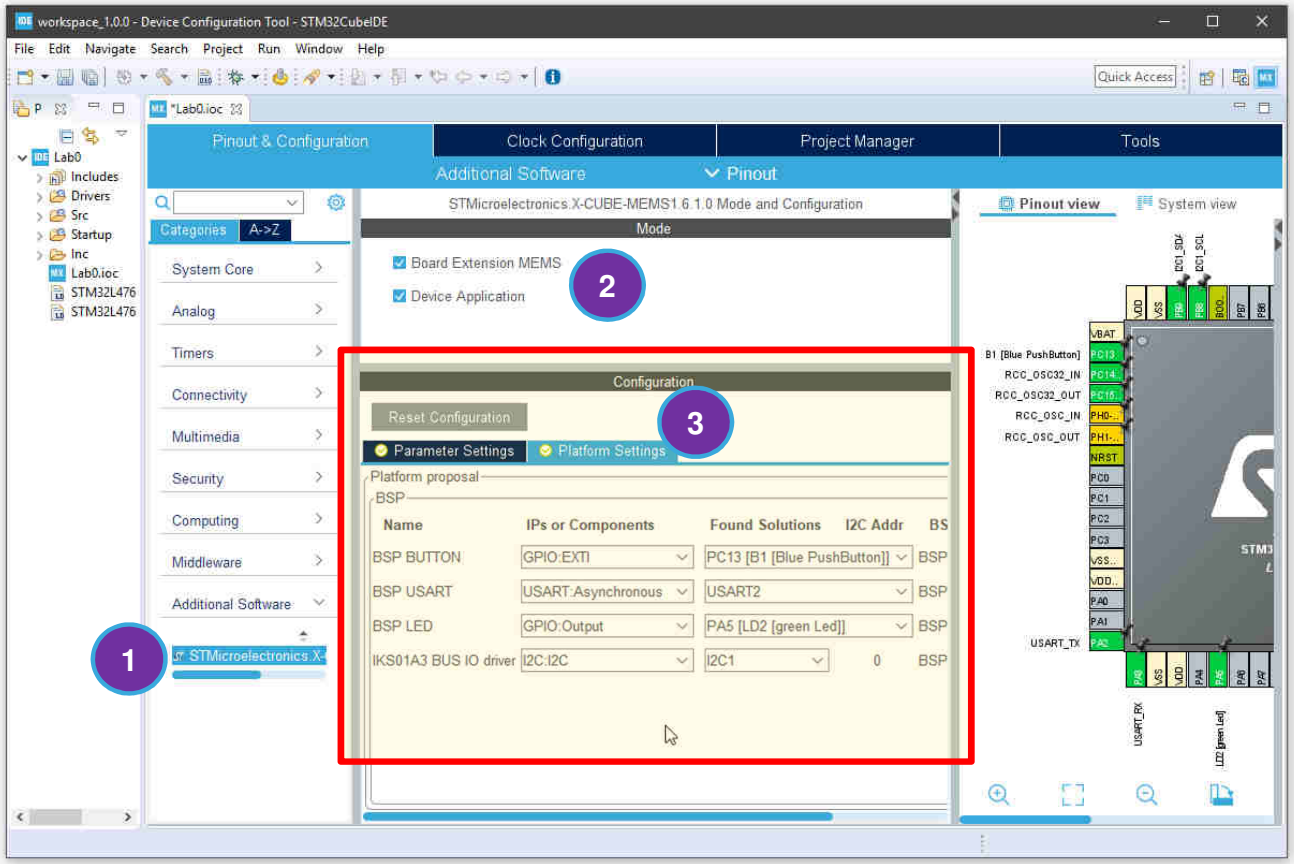
life.augmented

Lab0 – Configure the MEMS Library

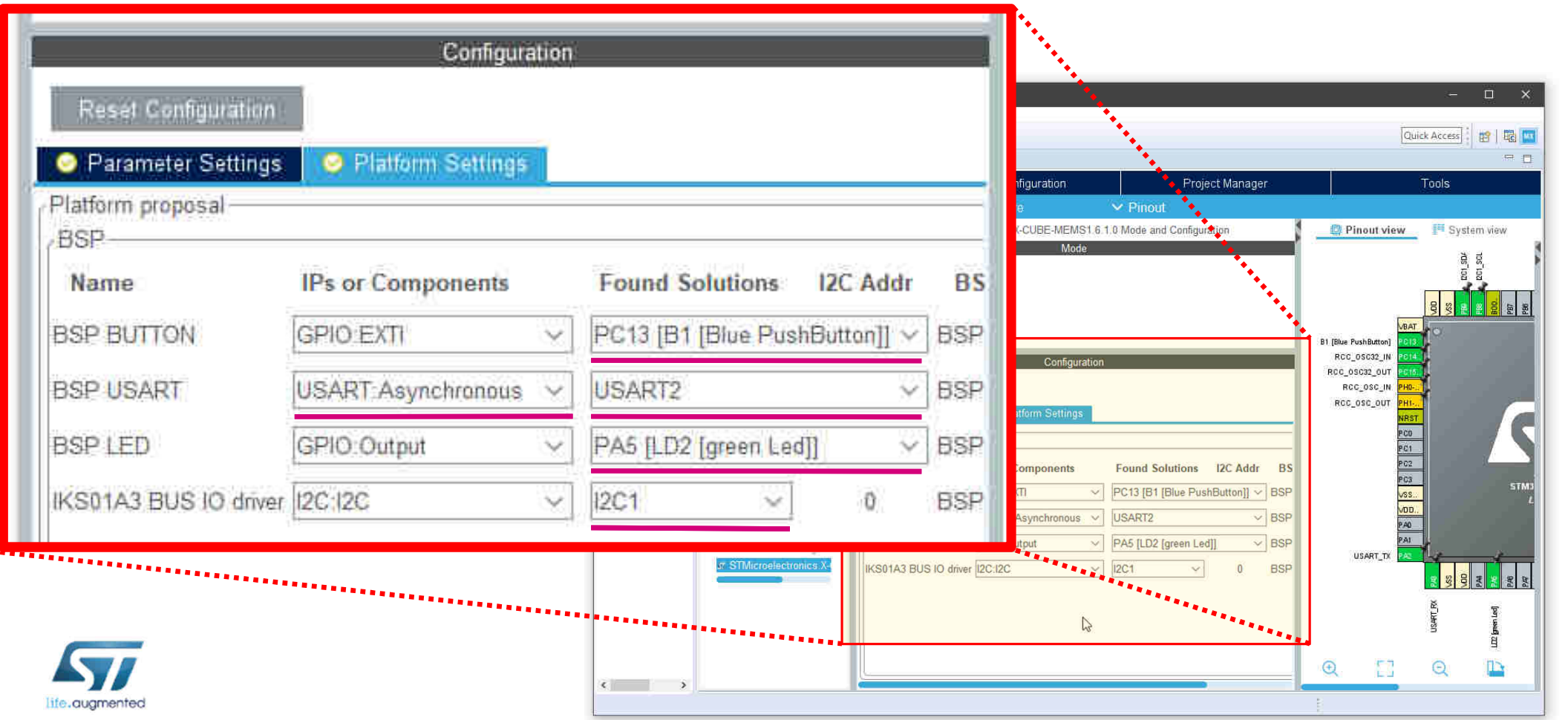
1. Expand Additional Software and select the X-CUBE-MEMS1

2. Check both:
   Board Extension MEMS
   Device Application

3. Configure Platform Settings as in picture (details in next page)



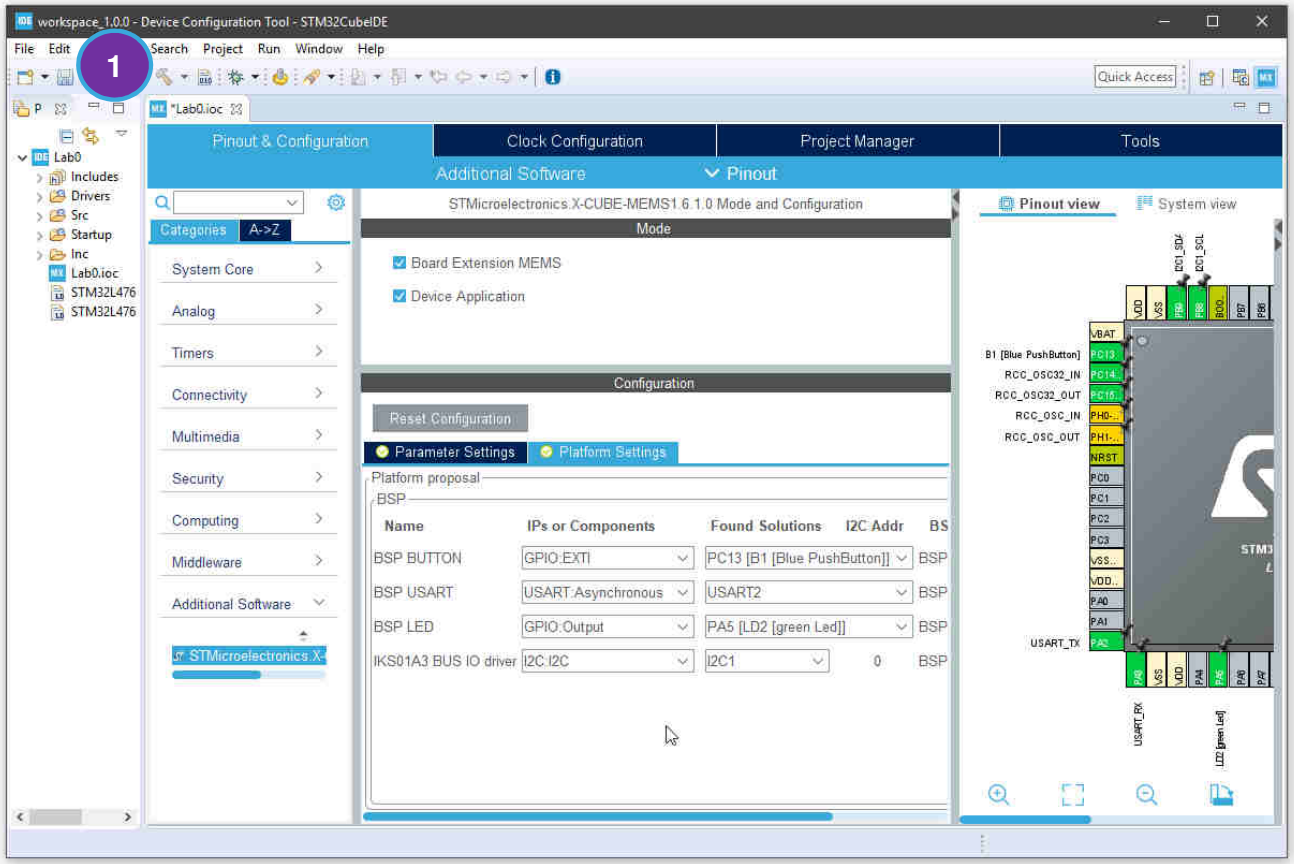
Lab0 – Configure the MEMS Library



life.augmented

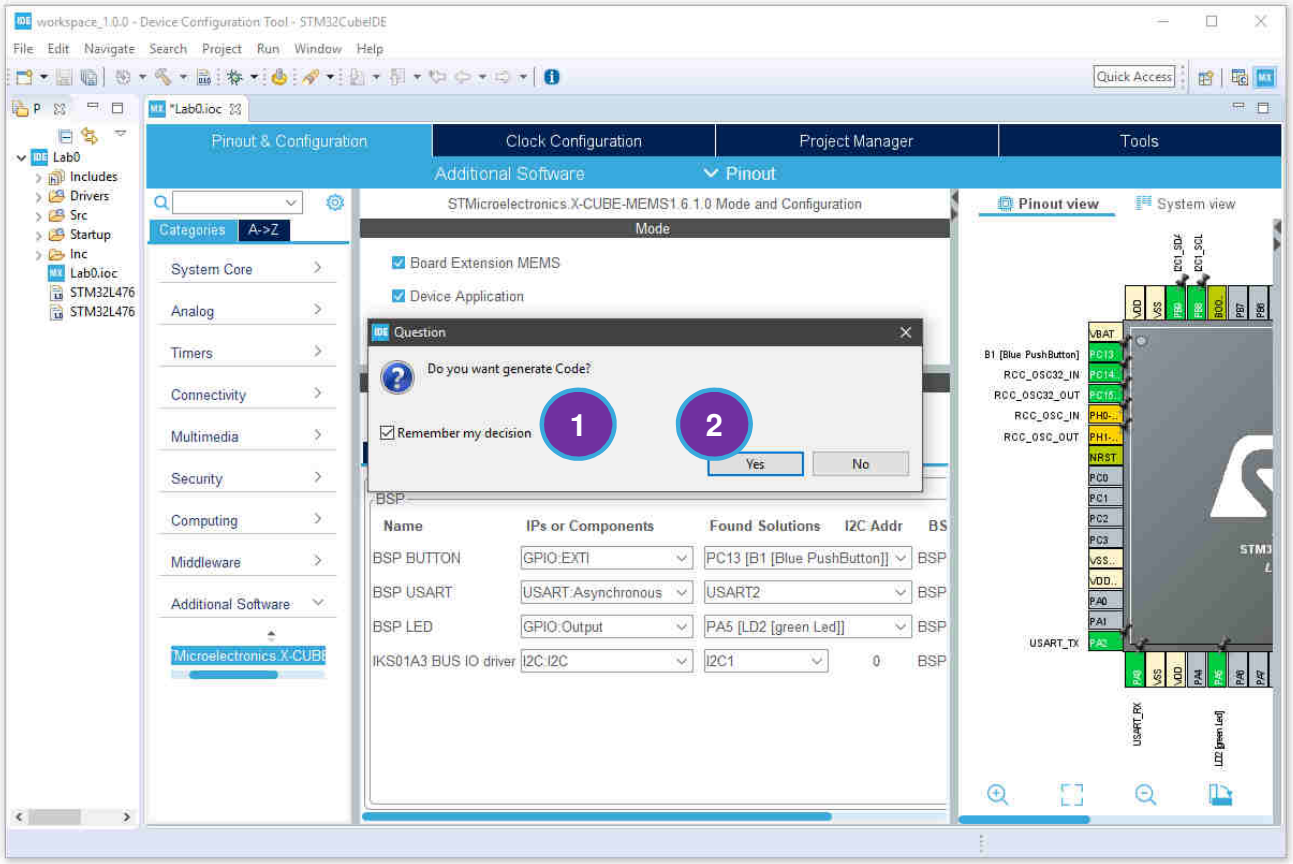
Lab0 – Save the Project

1. Click the save button 



Lab0 – Code Generation

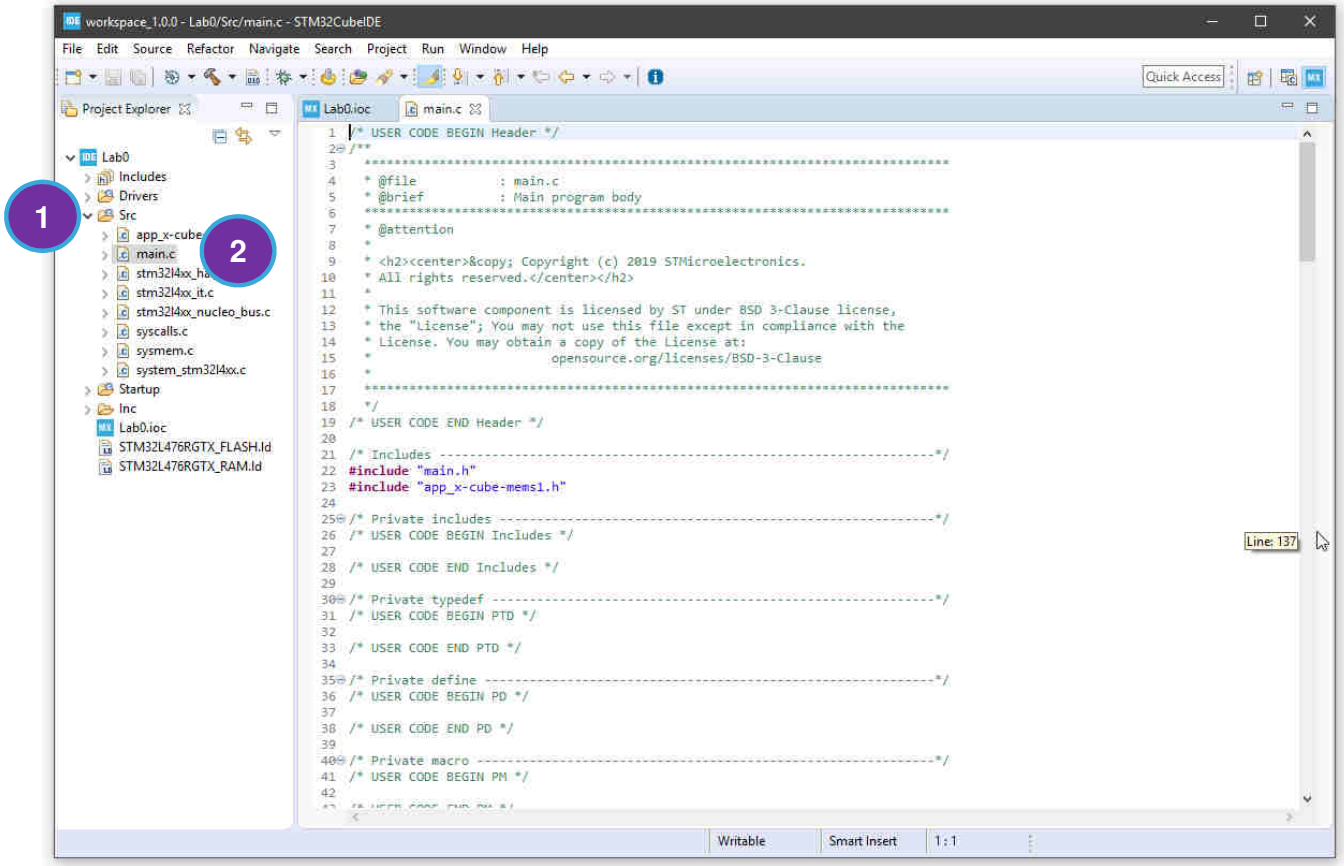
1. Check **Remember my decision**
2. Click **Yes**



life.augmented

Lab0 – Code Editing

1. Expand **Src**
2. Double click on **main.c**

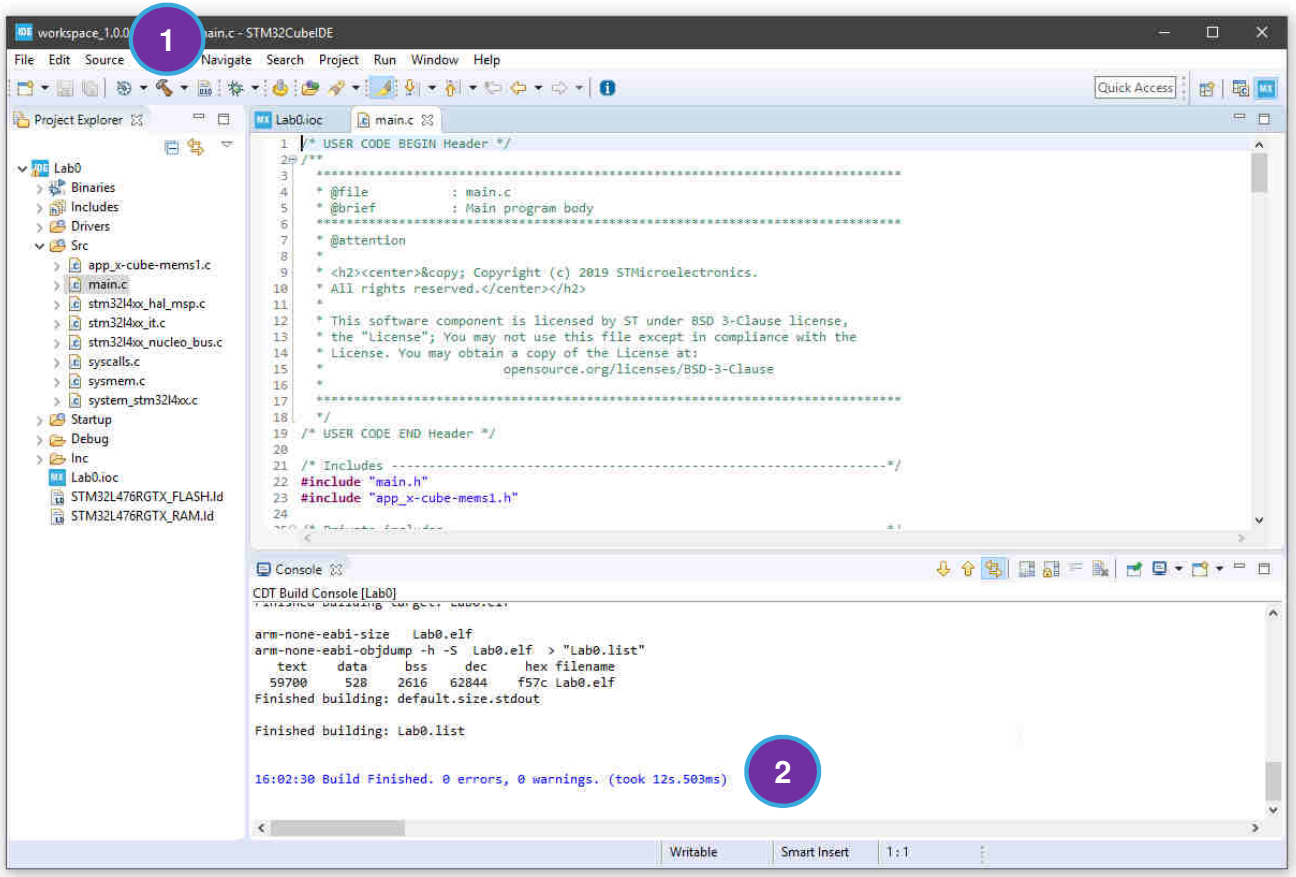


life.augmented

Lab0 - Compiling

1. Click on the hammer  to begin compilation, or press **CTRL+B**

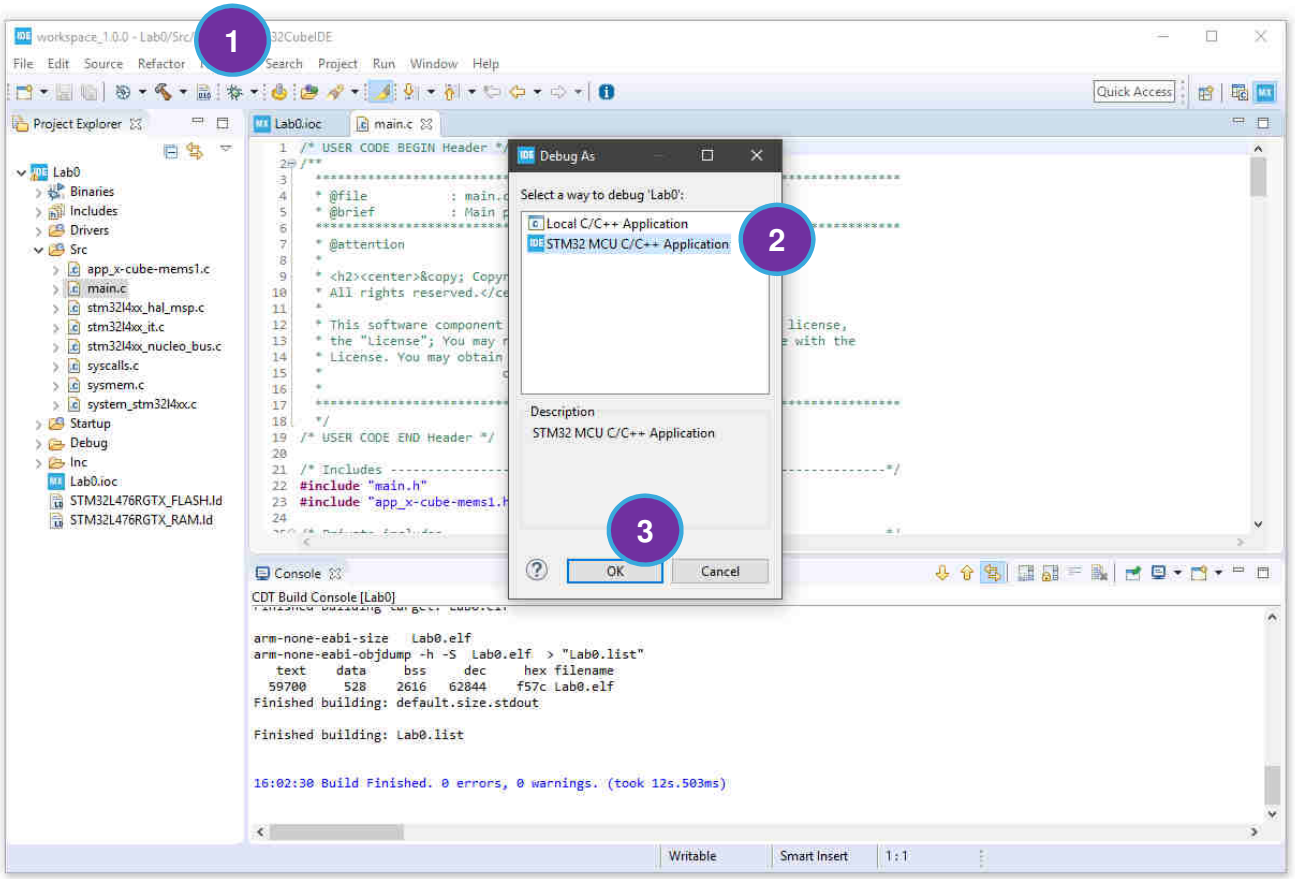
2. Compilation should terminate with 0 errors and 0 warning



life.augmented

Lab0 - Debugging

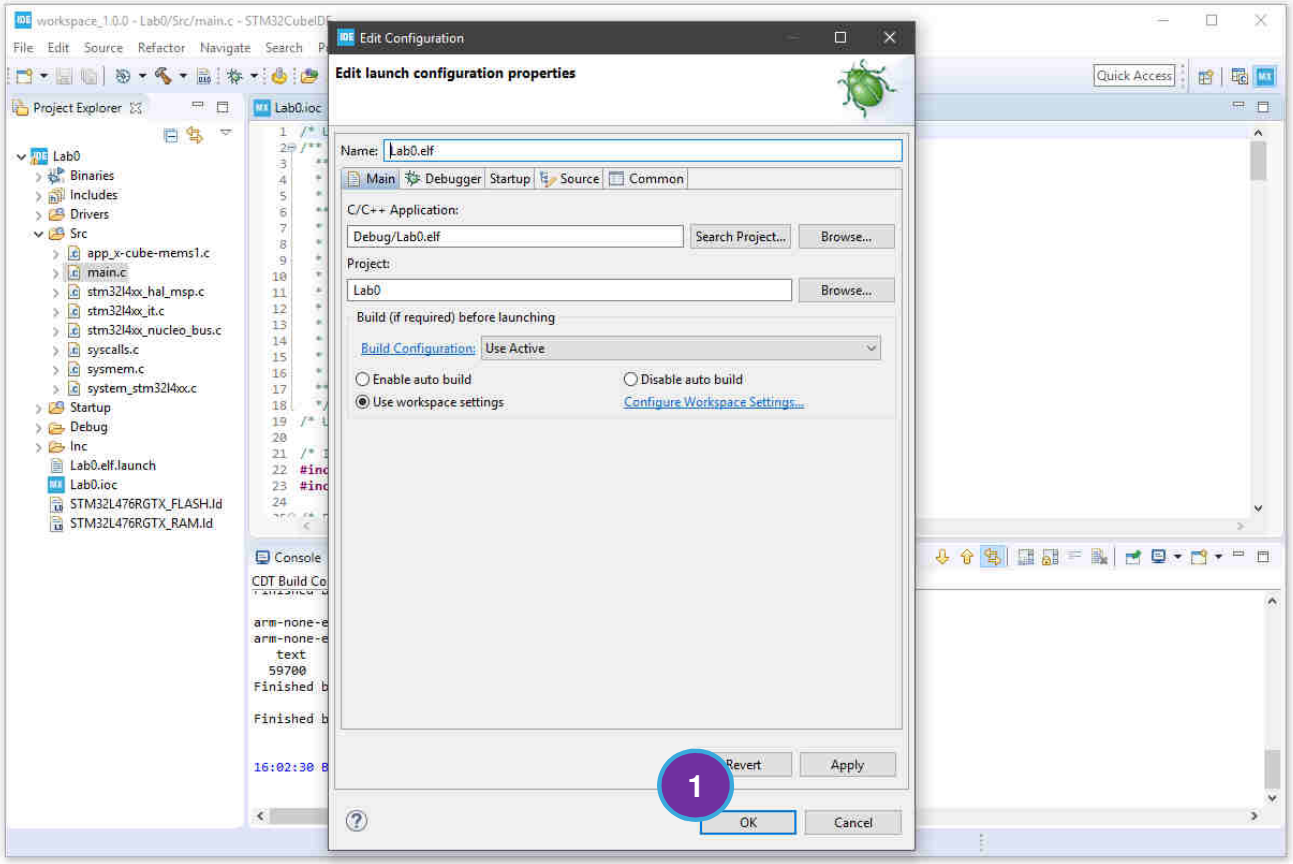
1. Click on the bug  to begin debugging
2. Select **STM32 MCU C/C++ App**
3. Click **OK**



life.augmented

Lab0 - Debugging

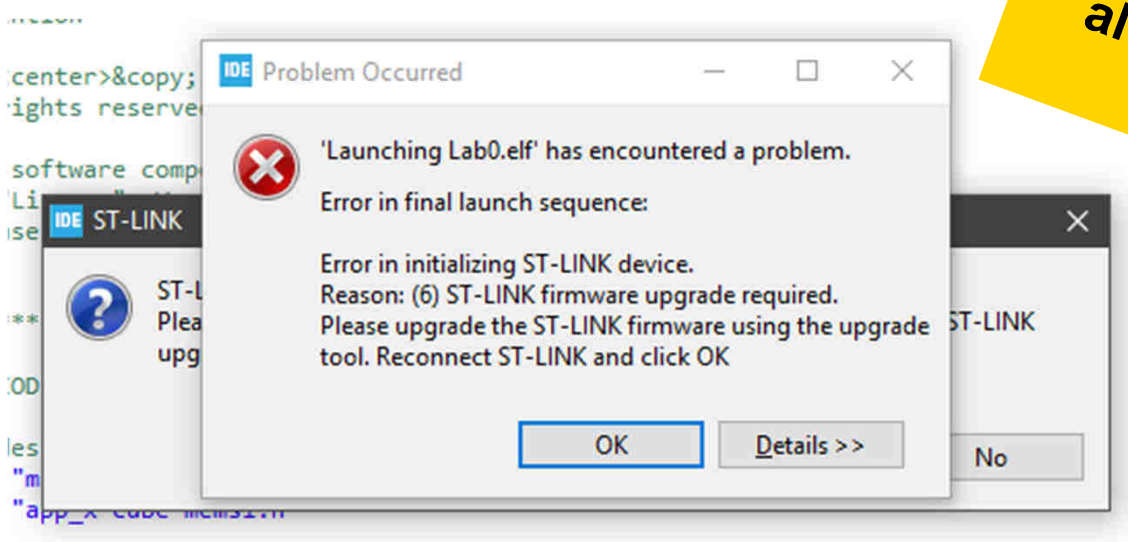
1. Click **OK**



life.augmented

Lab0 - Debugging

- *The first time you debug the board by pressing*  , a **Problem Occurred** because ST-LINK needs to be upgraded

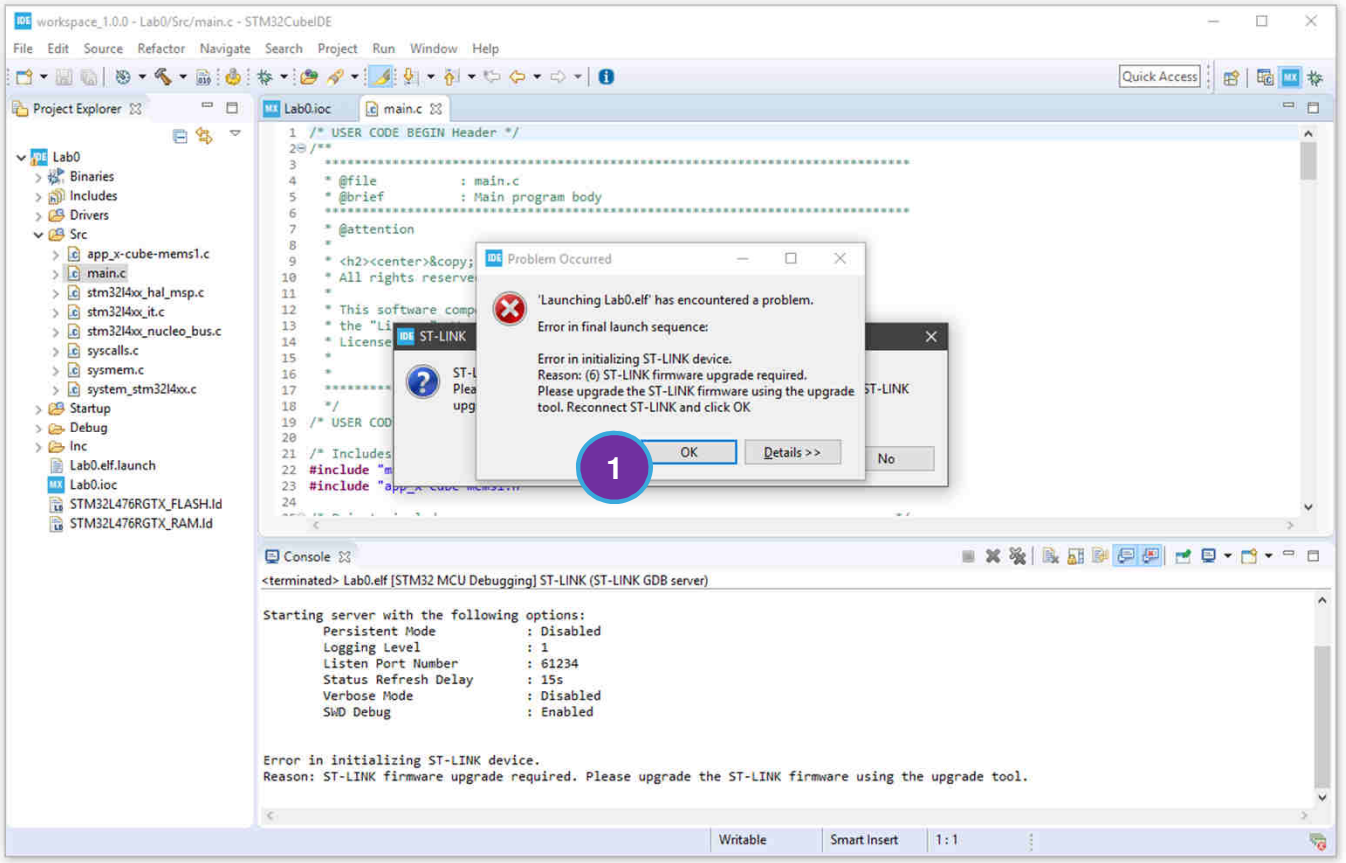


New NUCLEO boards always need to be upgraded

life.augmented

Lab0 – Updating ST-LINK

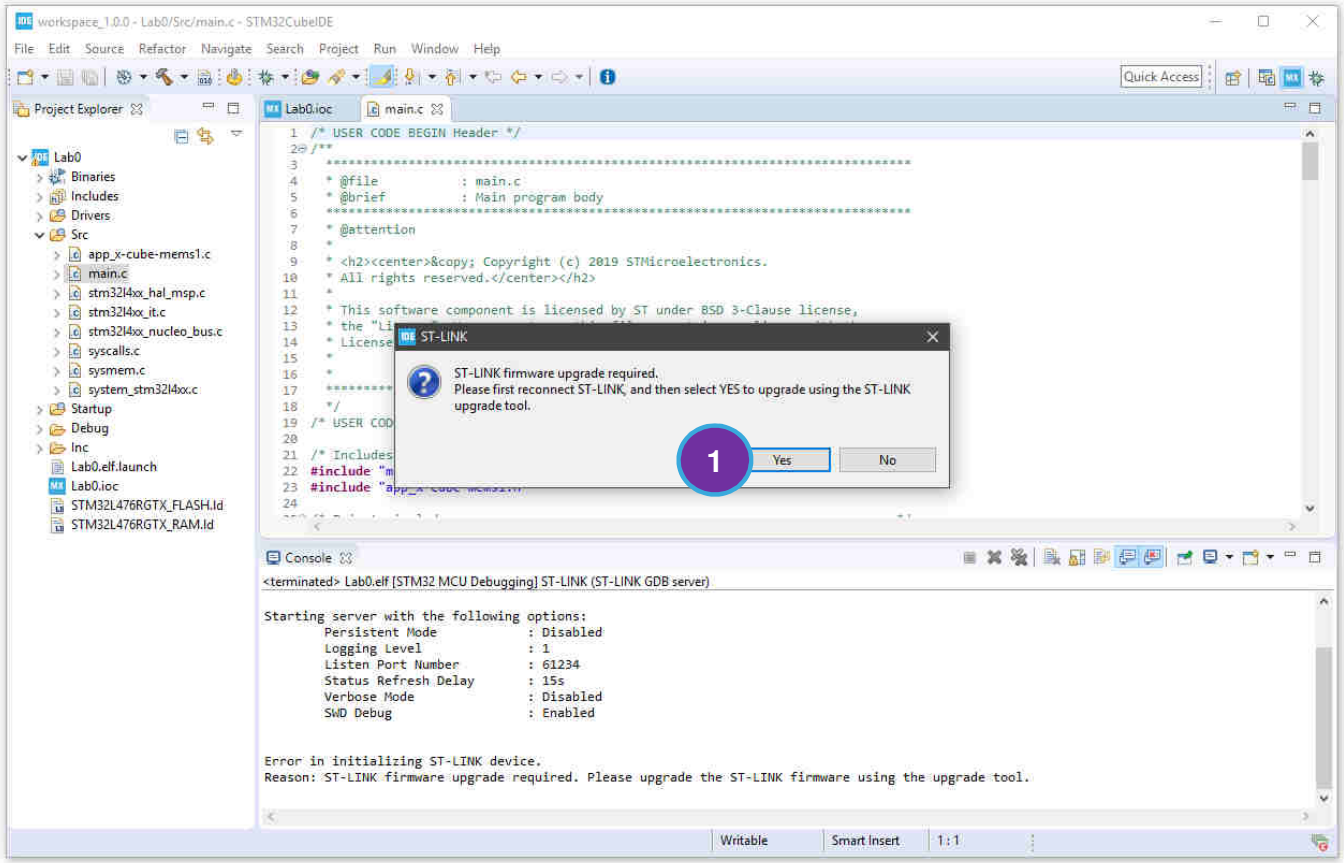
1. Click **OK** to run upgrade to latest firmware



life.augmented

Lab0 – Upgrading ST-LINK

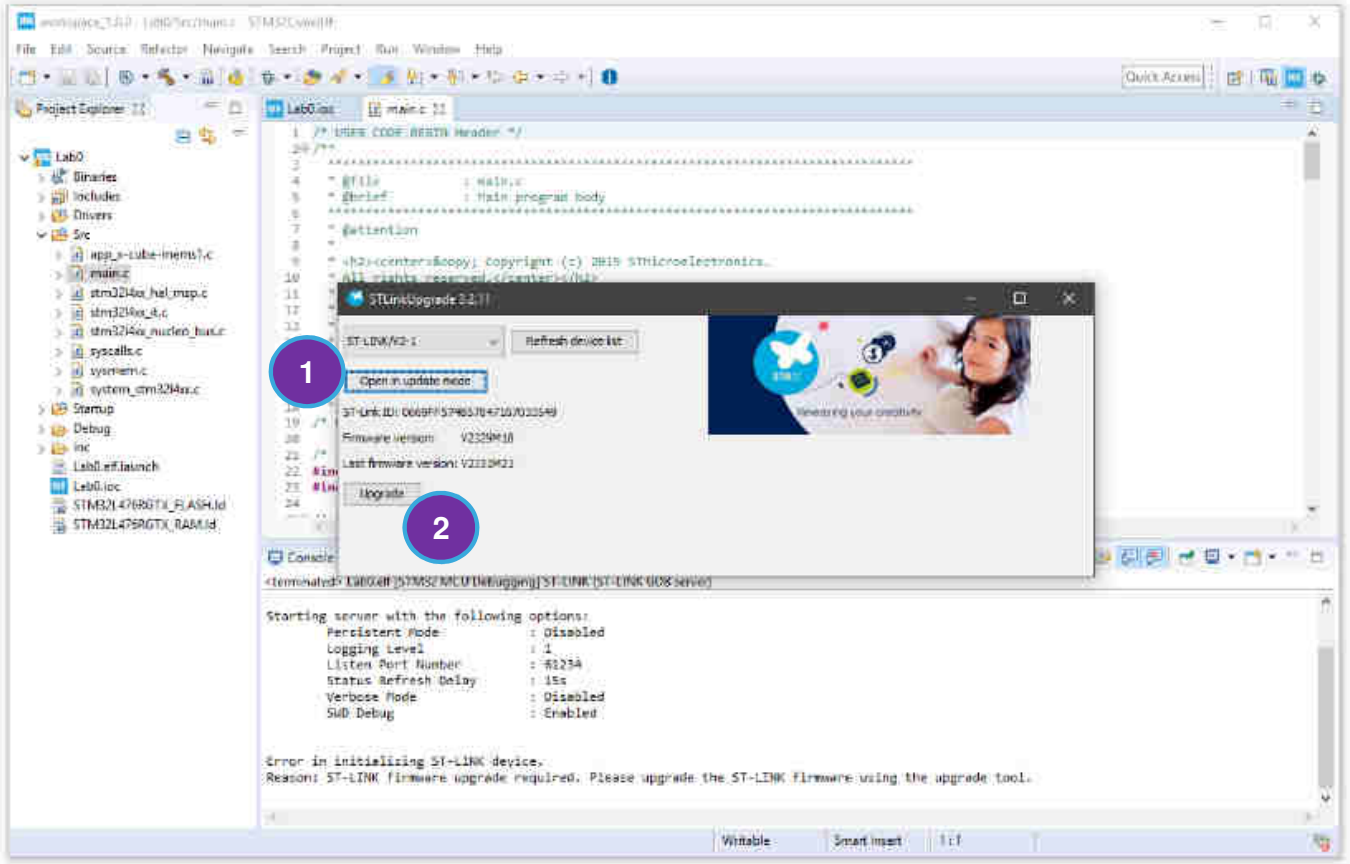
1. Click **Yes** when is asked to upgrade the ST-LINK



life.augmented

Lab0 – Upgrading ST-LINK

1. Click **Open in update mode** to force ST-LINK
2. Click on **Upgrade**

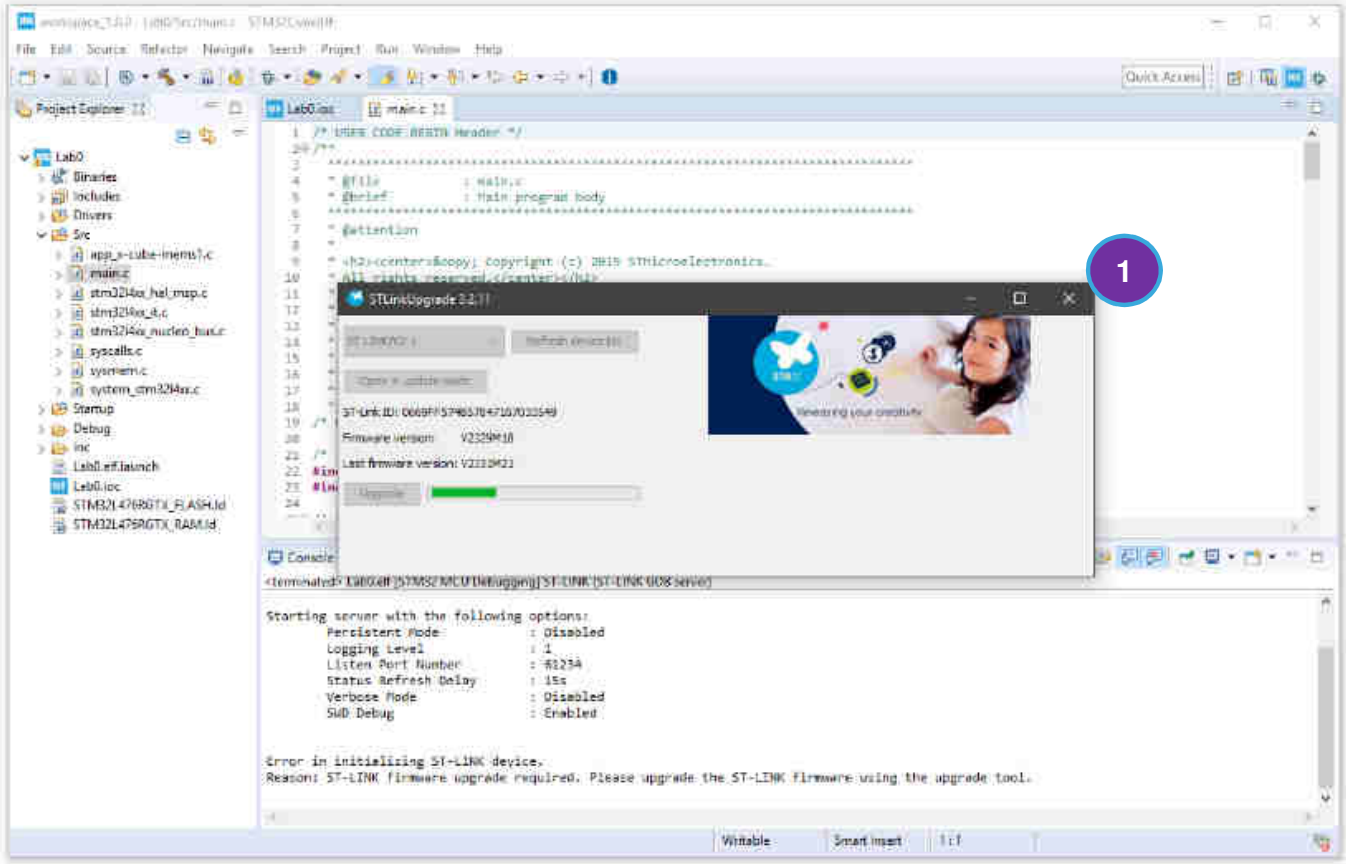


life.augmented

Lab0 – Upgrading ST-LINK

Wait until update is finished and then close the window.

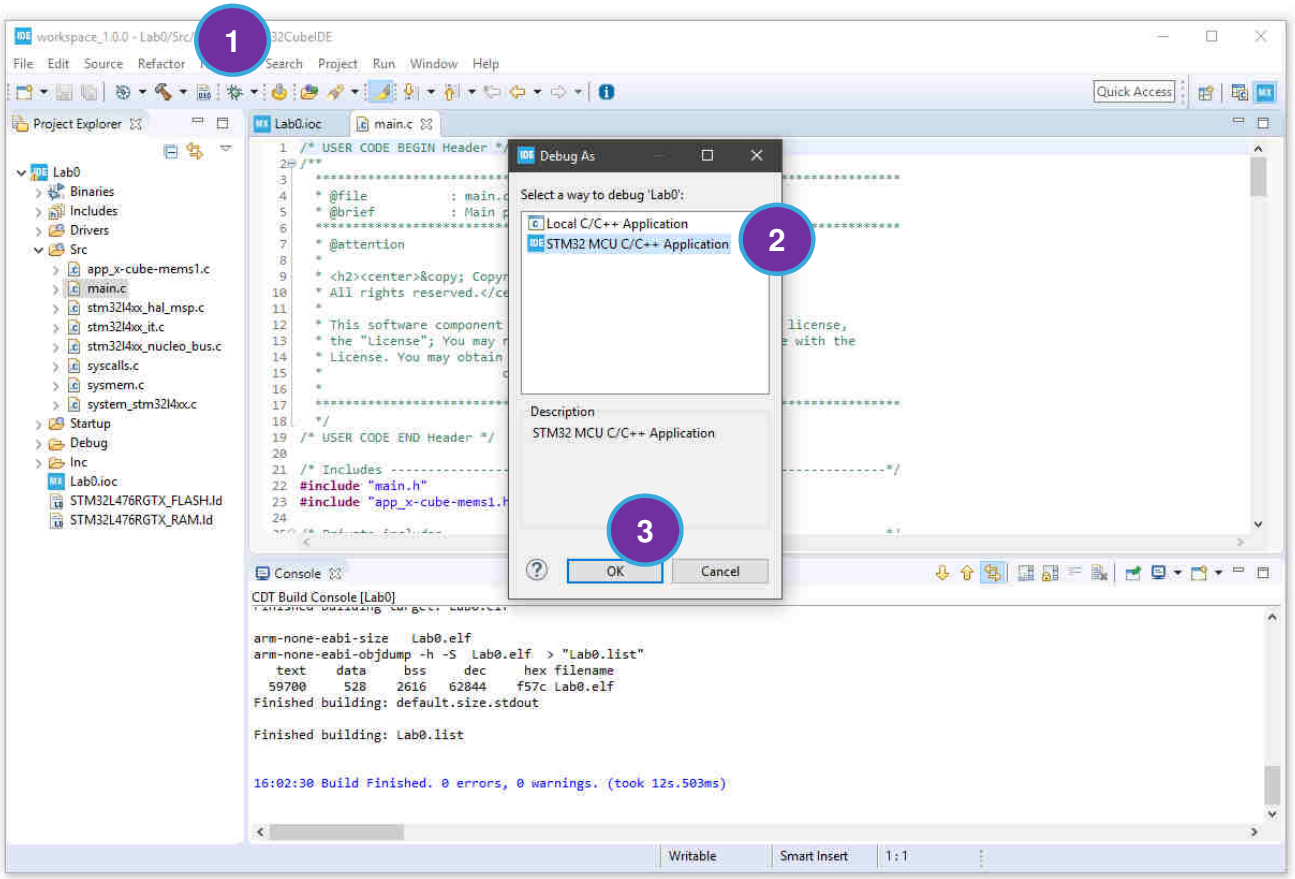
1. Click on **X** when finished



life.augmented

Lab0 - Debugging

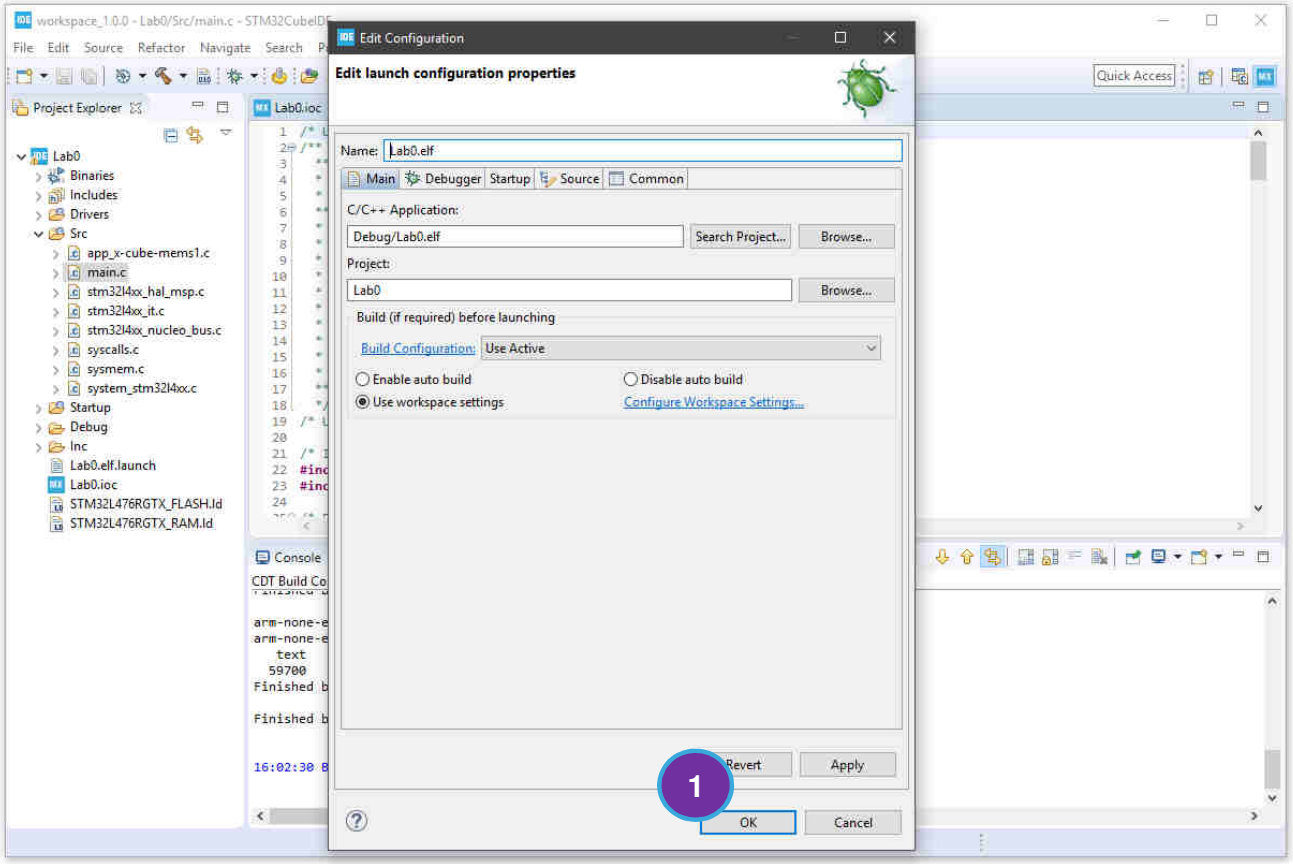
1. Click again on the bug  to begin debugging
2. Select **STM32 MCU C/C++ App**
3. Click **OK**



life.augmented

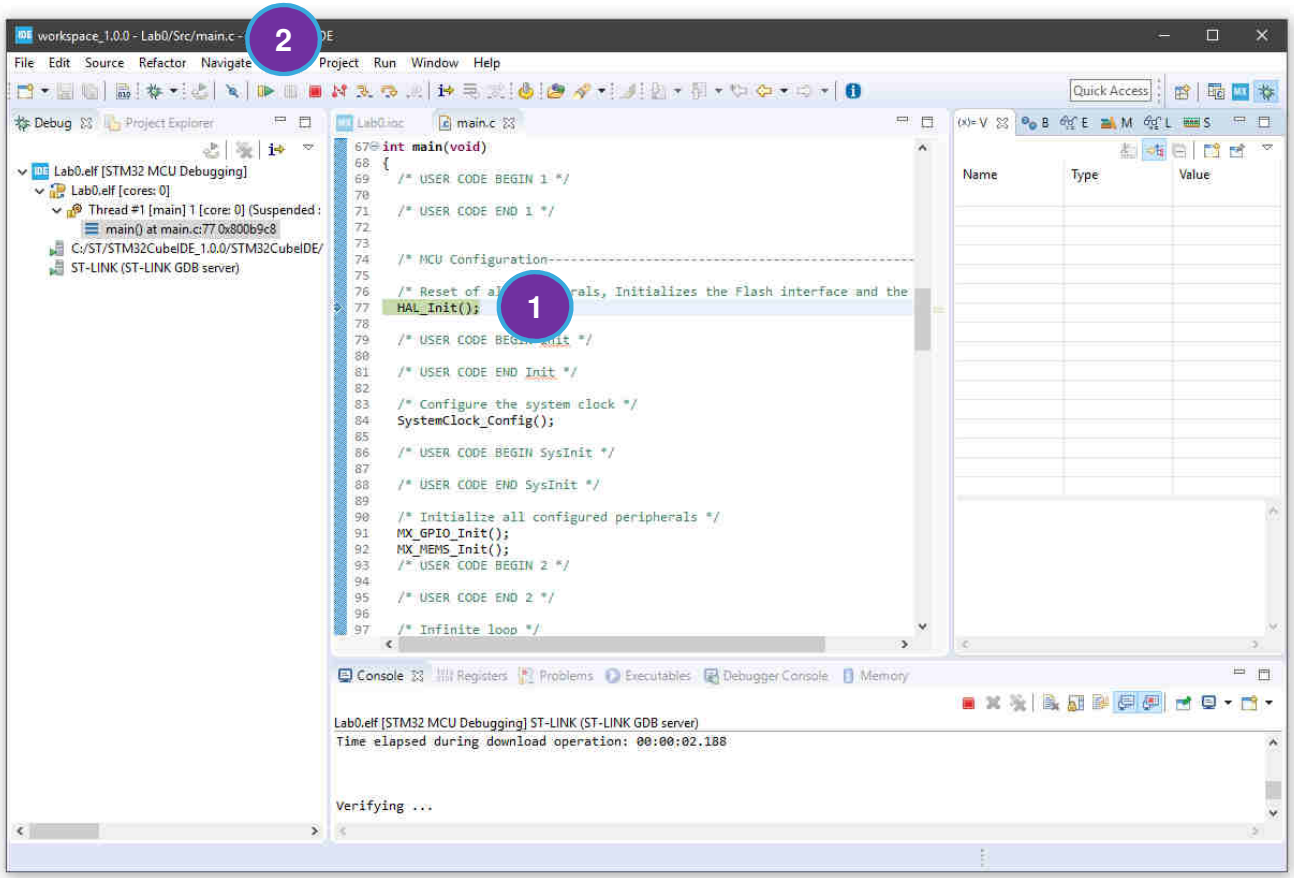
Lab0 - Debugging

1. Click **OK**



Lab0 - Debugging

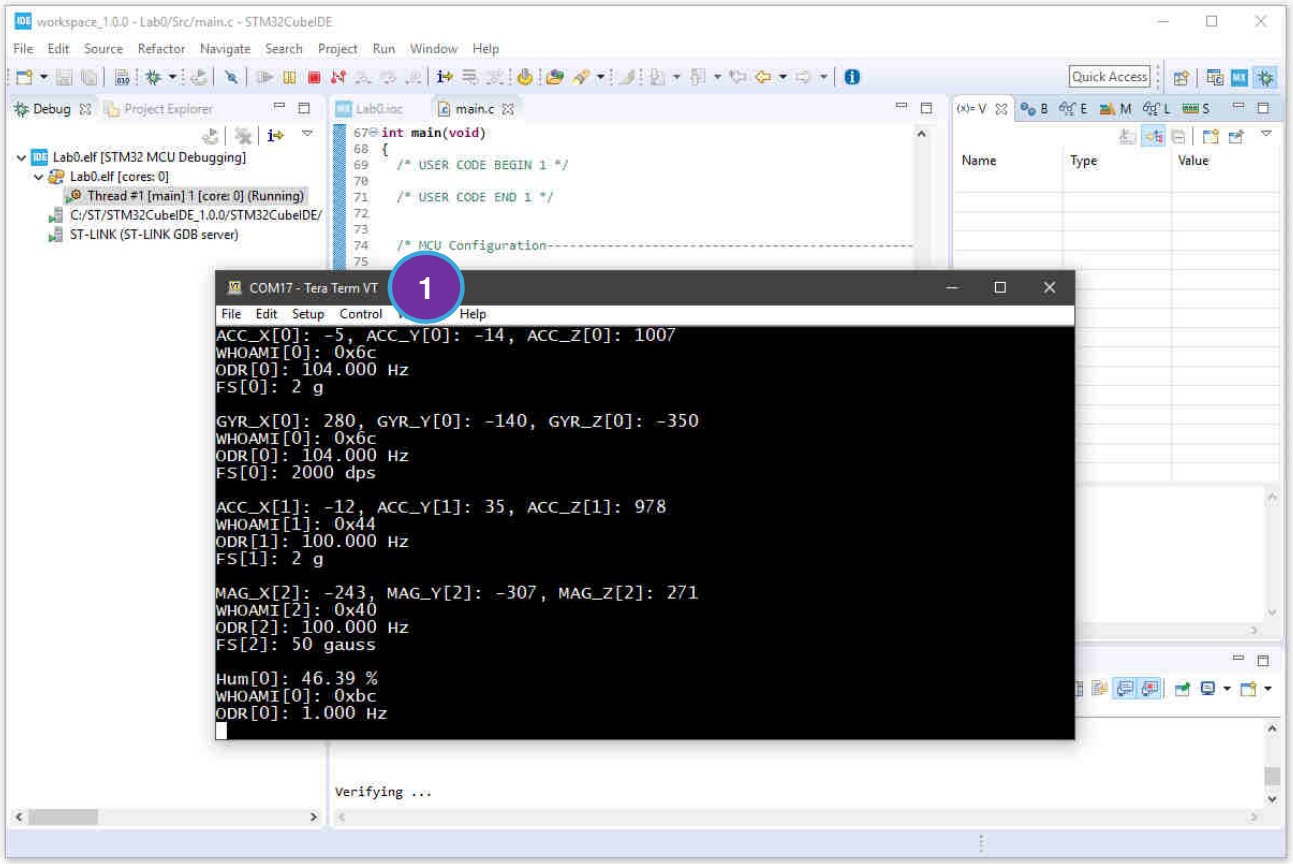
1. Code start at the first line of the main function
2. Click play  button to run the code



life.augmented

Lab0 - Debugging

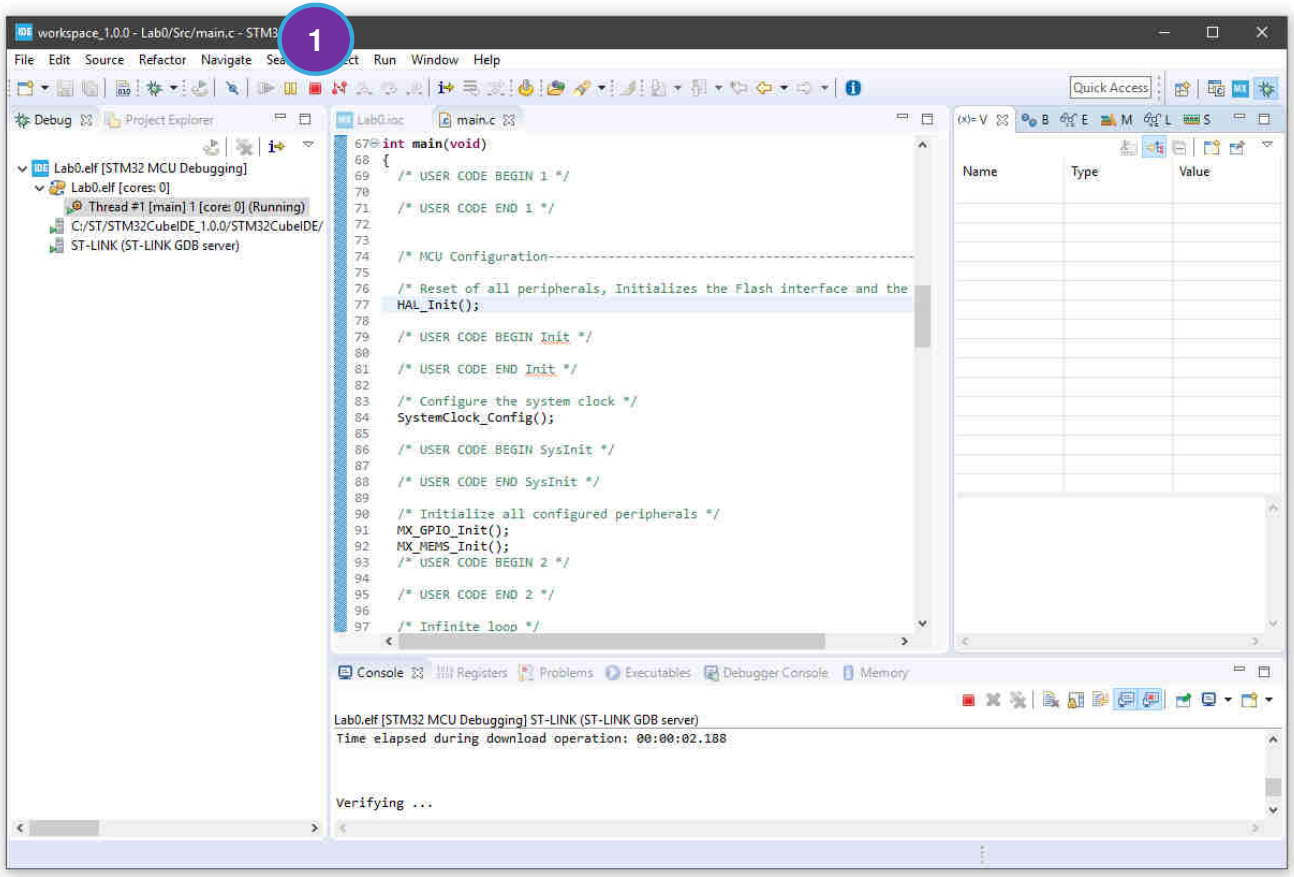
1. Open Tera Term to view the output



life.augmented

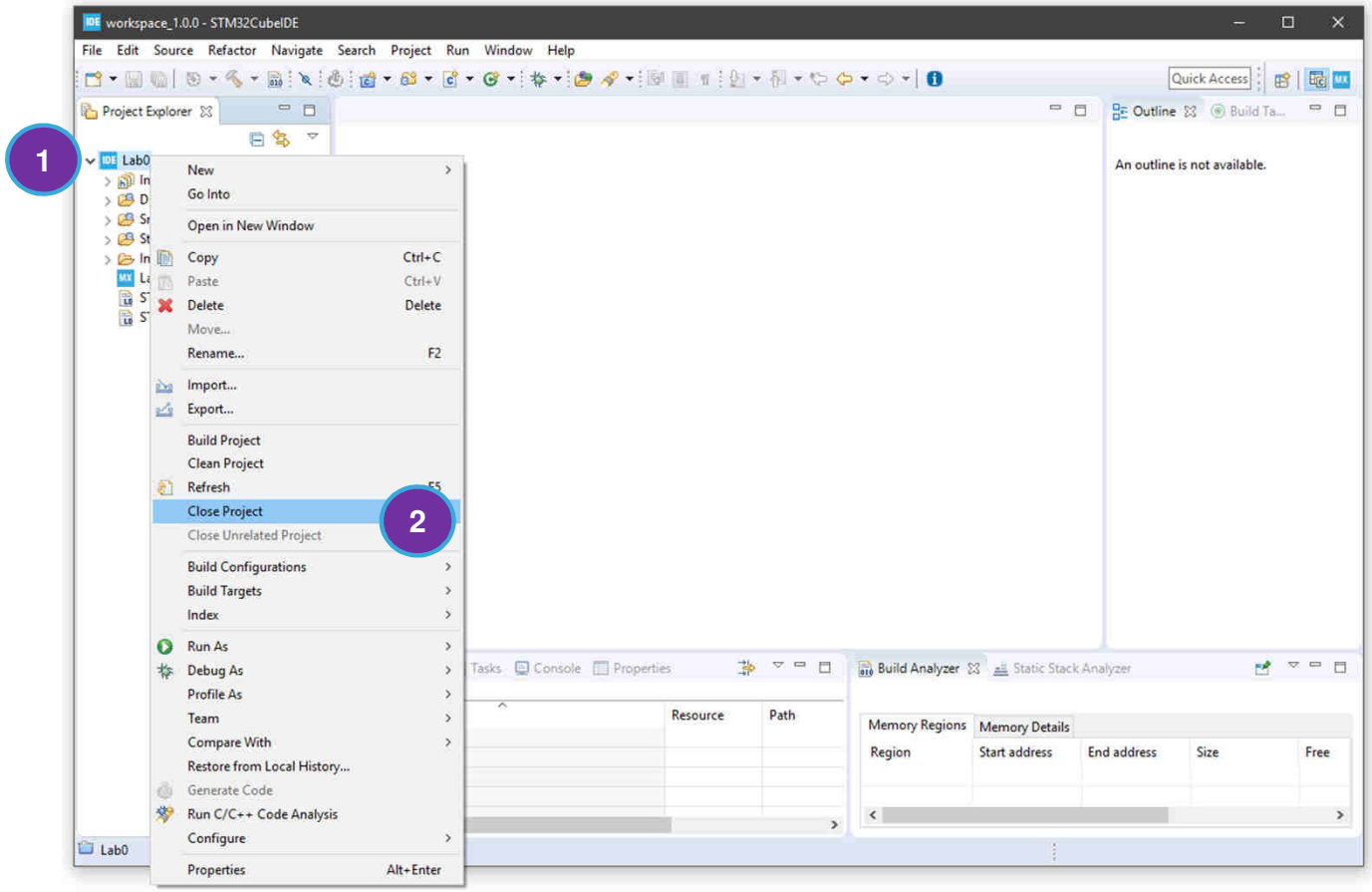
Lab0 - Debugging

1. Click stop  button to interrupt the debugging



Lab0 – Closing the Project

1. Right-Click on **Lab0** project
2. Click on **Close Project**



life.augmented

LAB1

Goals:

- Configure a new project using X-CUBE-MEMS1
- Configure LIS2DW12 accelerometer in order to generate an interrupt at every orientation change
- Enable interrupts in STM32CubeIDE
- Visualizing the new orientation on Tera Term

life.augmented

LIS2DW12

Power saving and flexible Accelerometer

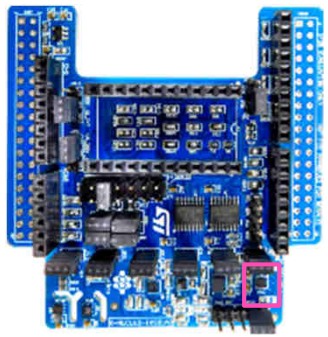
- Down to 0.38 µA power consumption (1.6Hz ODR)
- High Perf. mode: up to 1600Hz with noise density 90 µg/√Hz
- 5 Power Modes + 2 Noise Modes
- 32-level FIFO buffer
- Digital Features
  - Free fall
  - Wake-up
  - 6D / 4D
  - Stationary/Motion detection
  - Double Tap

Enabling battery saving

Accuracy

High Flexibility

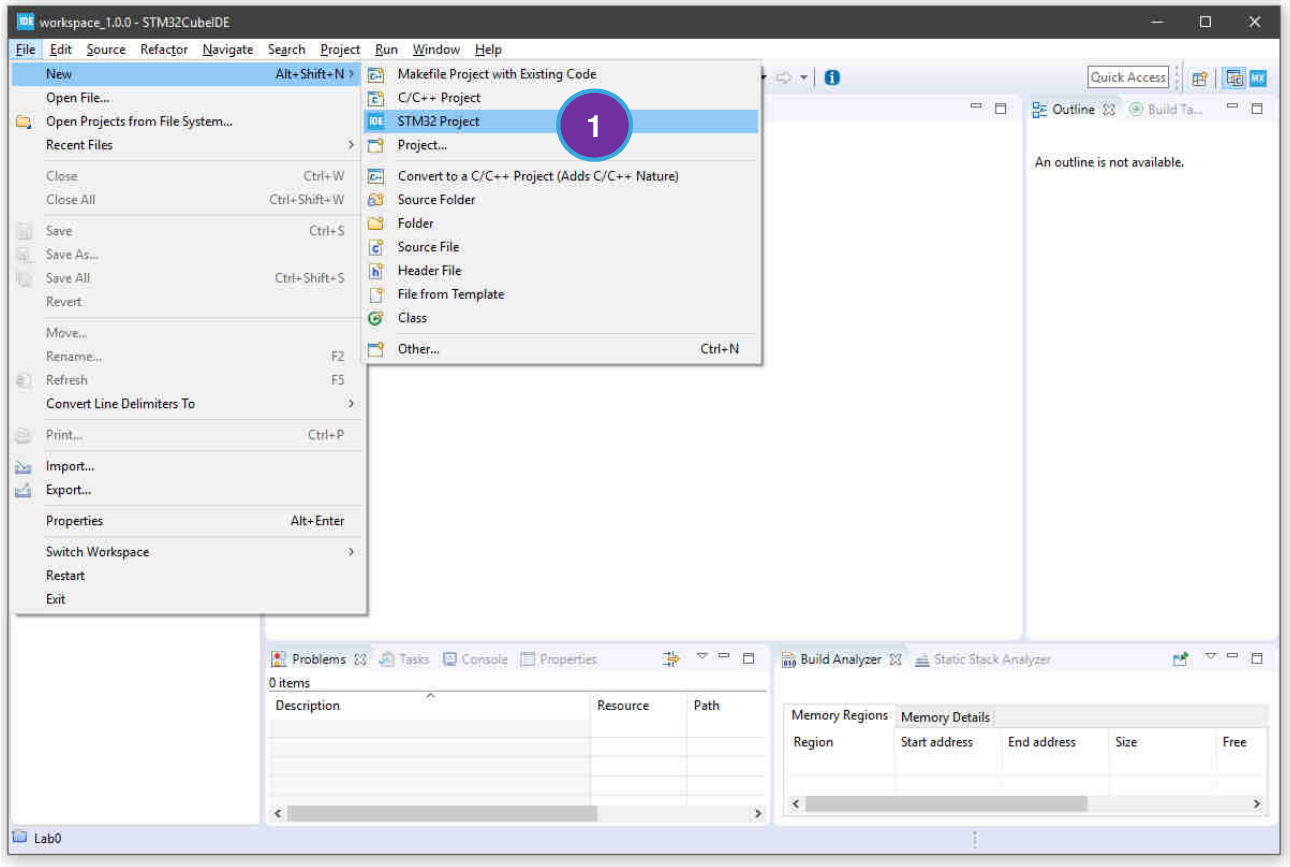
System Power Saving & Smart Functions



LGA-12L
(2.0 x 2.0 x 0.7 mm)

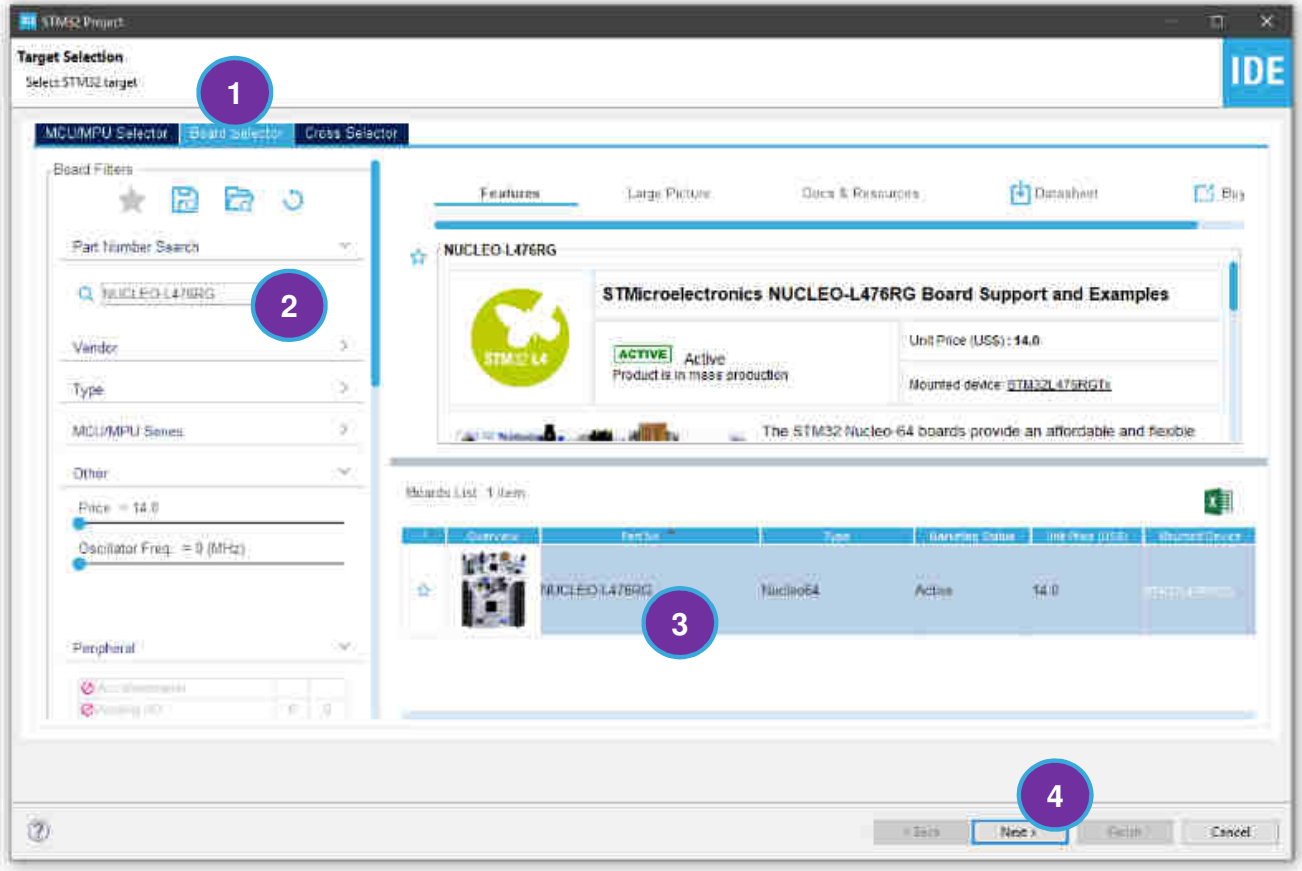
life.augmented

1. Click on **File > New > STM32 Project**



life.augmented

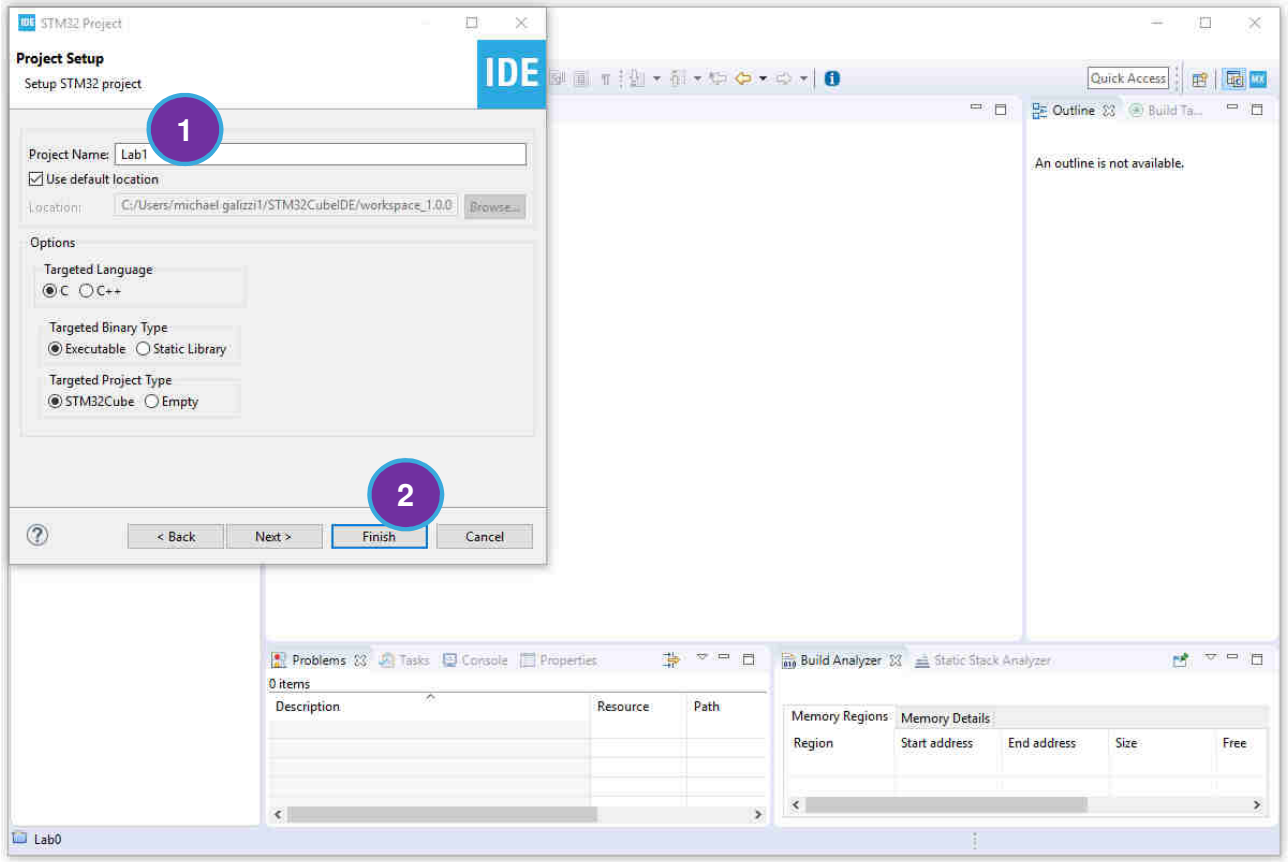
1. Click on **Board Selector**
2. Type **NUCLEO-L476RG**
3. Click on the board
4. Click **Next >**



life.augmented

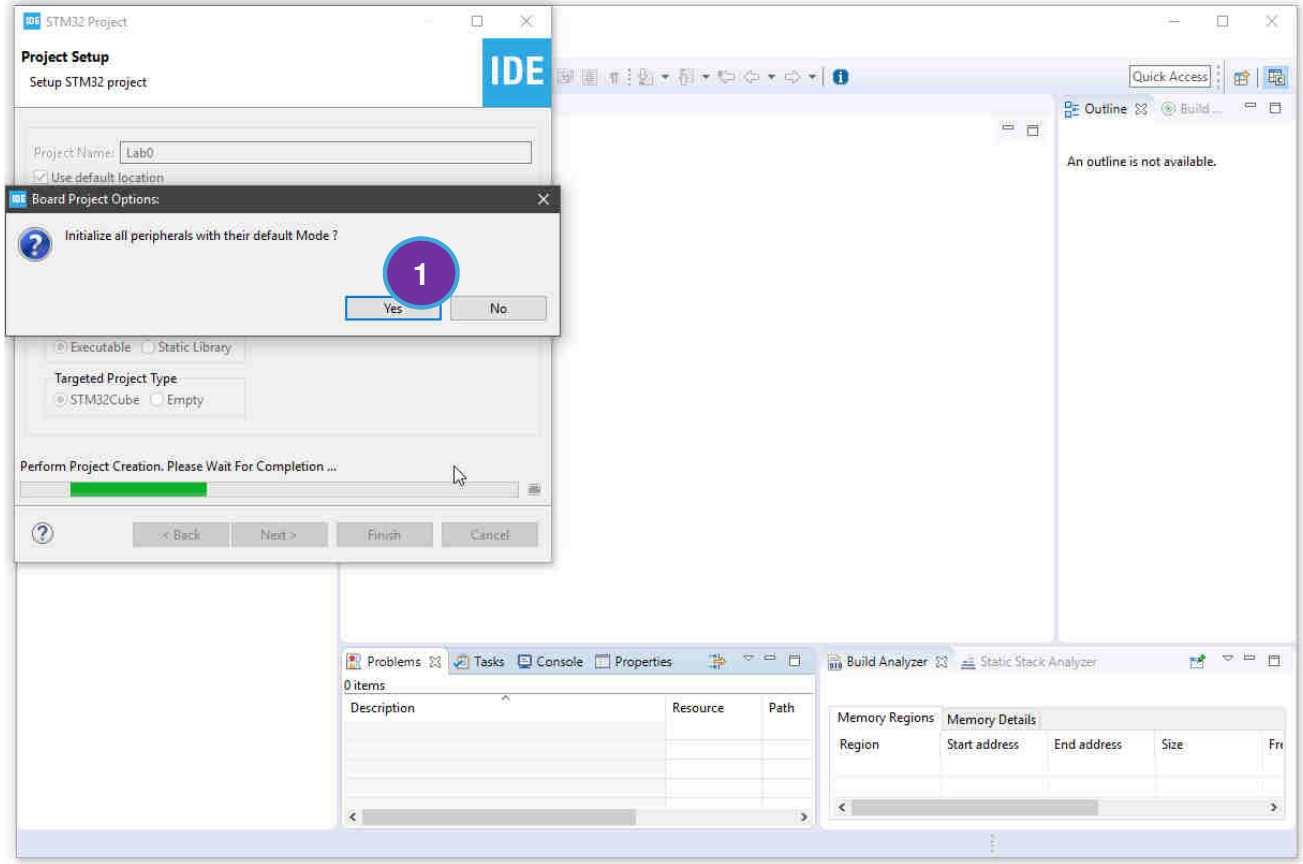
Lab1

1. Project Name **Lab1**
2. Click **Finish**



life.augmented

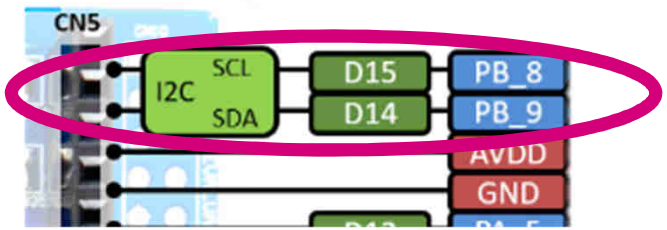
1. Click **Yes** to init peripherals in default mode

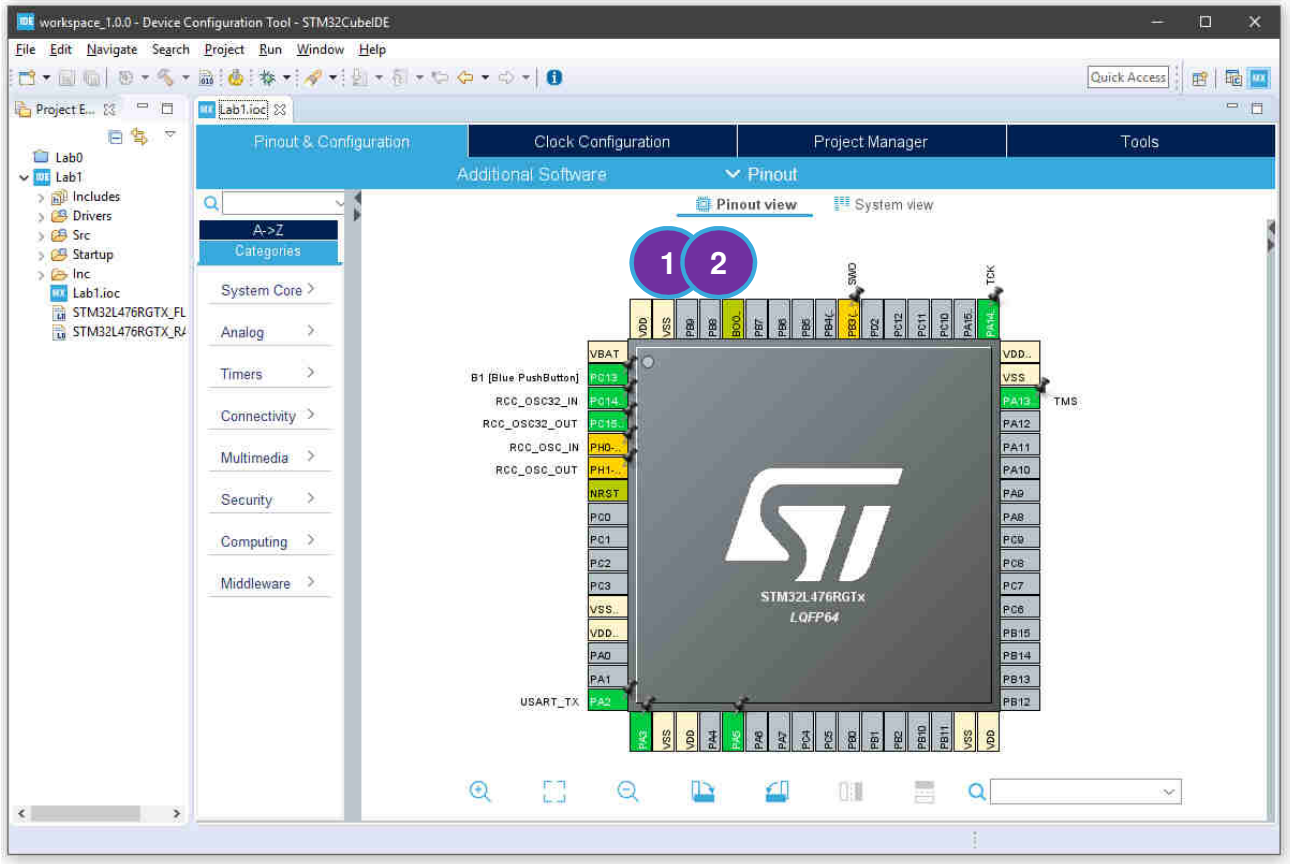


life.augmented

Lab1 – Configure the I2C Bus

1. Left Click on **PB9** and select I2C1_SDA
2. Left Click on **PB8** and select I2C1_SCL

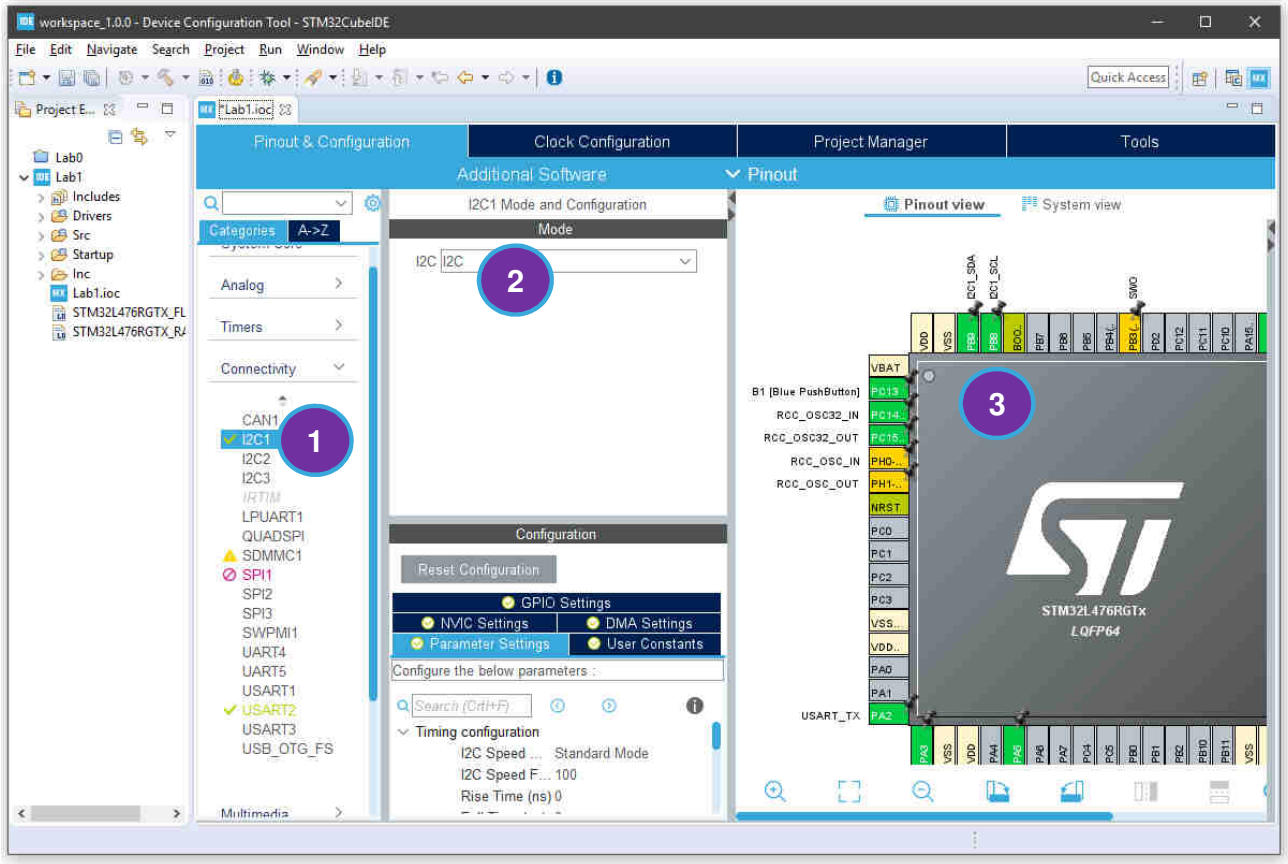




life.augmented

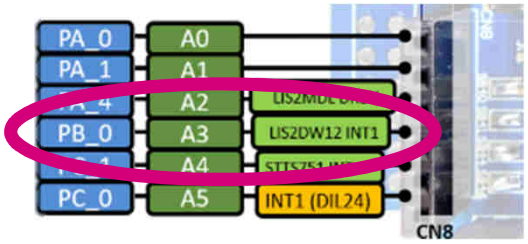
Lab1 – Configure the I2C Bus

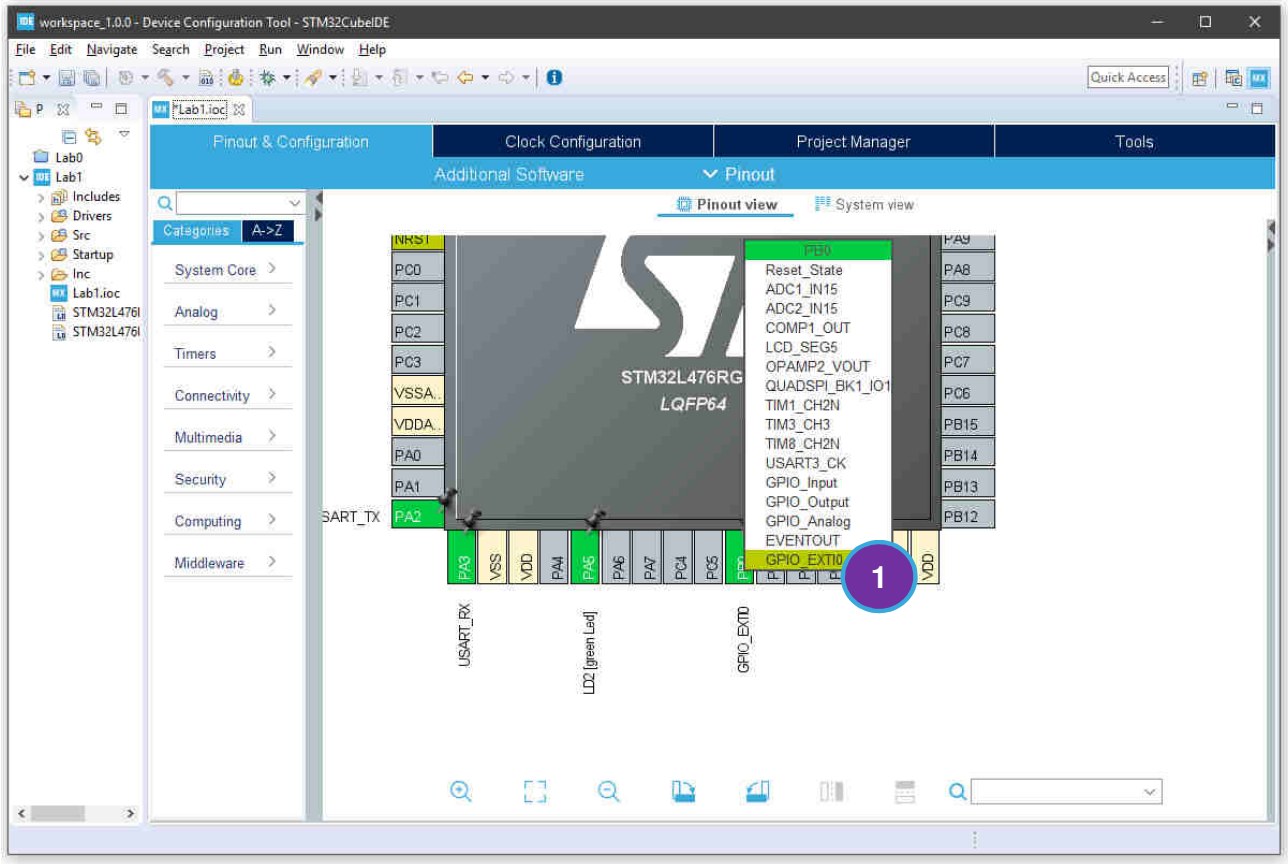
1. Expand *Connectivity* tab and check **I2C1**
2. Select **I2C** in *I2C1 Mode and Configuration*
3. PB8 and PB9 should now become green



Lab1 – Configure LIS2DW12 Interrupt

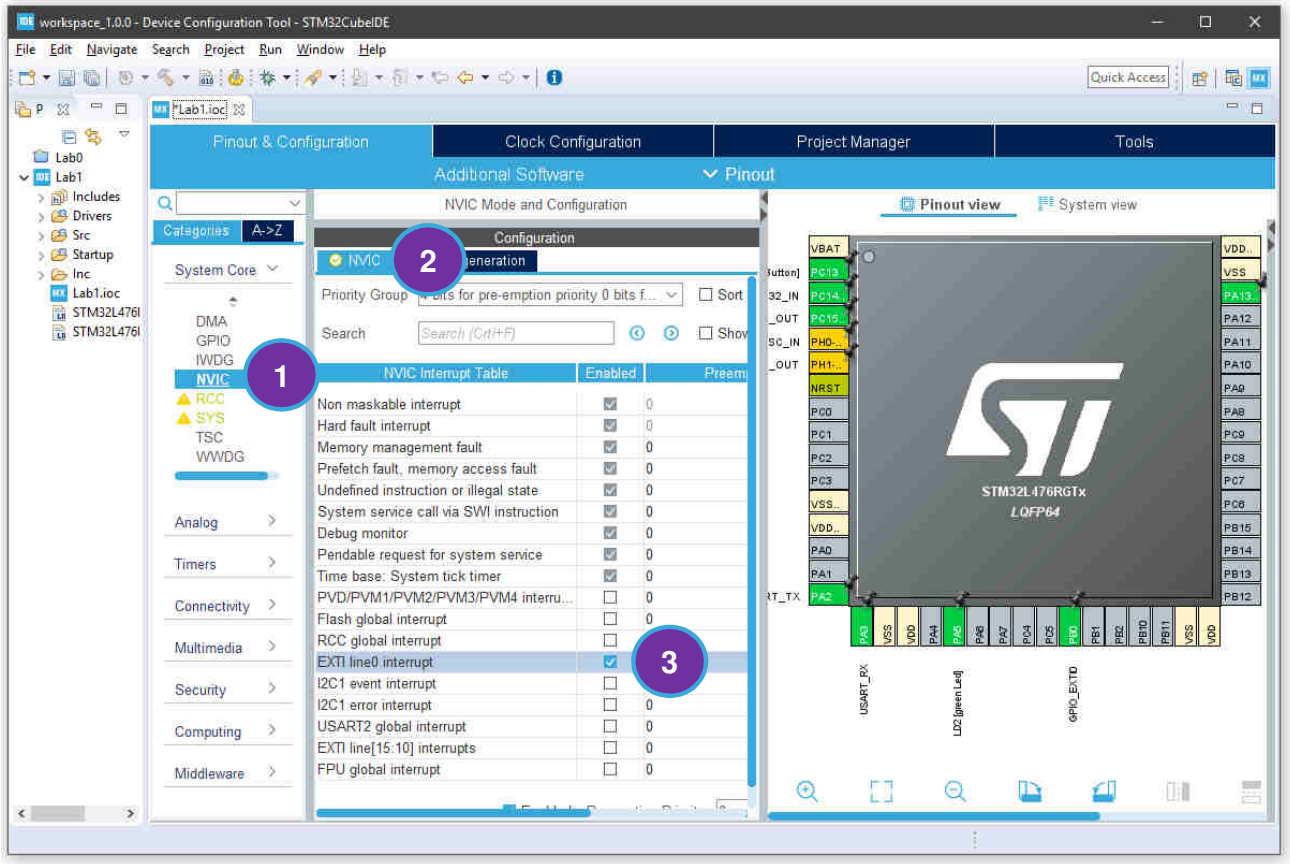
1. Left Click on **PB0** and select **GPIO_EXTI0**





Lab1 – Configure LIS2DW12 Interrupt

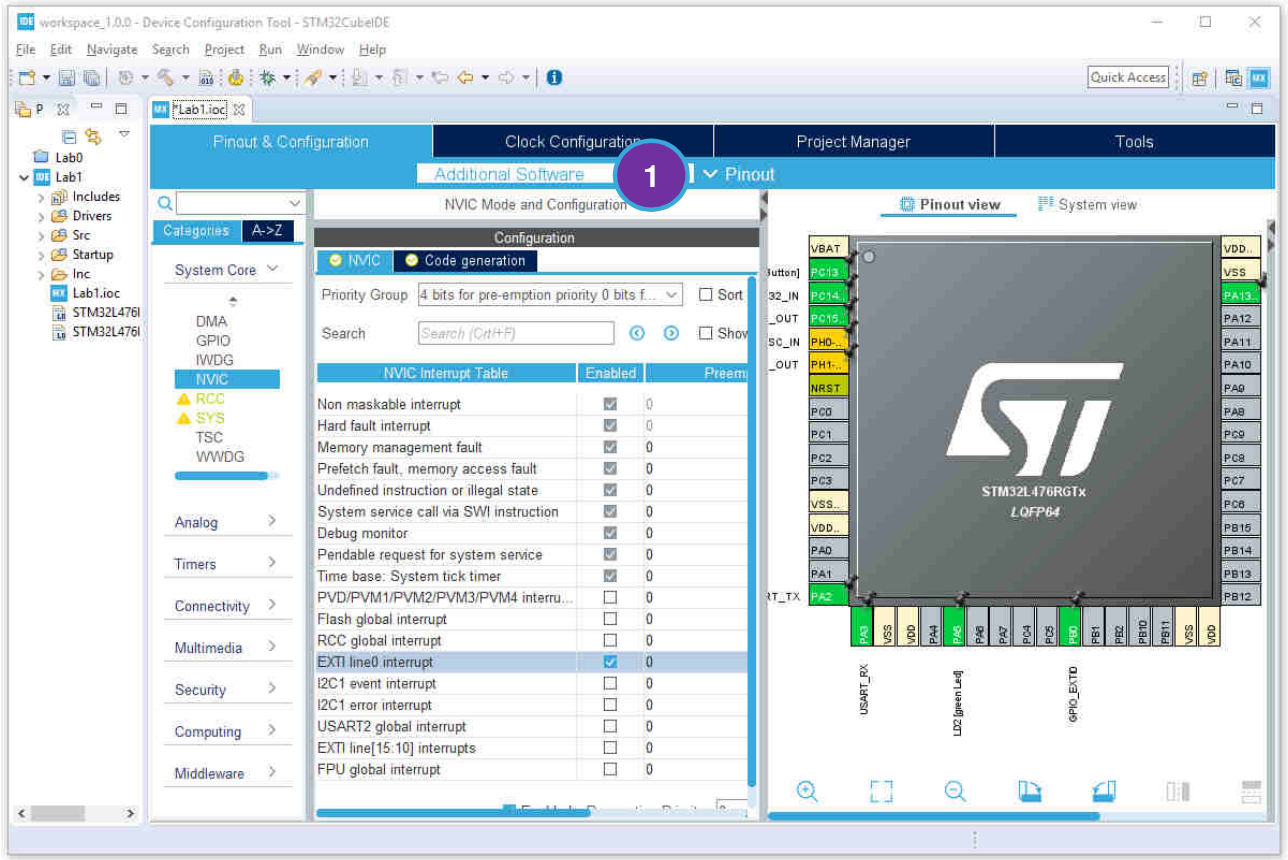
1. Check **NVIC** in tab System Core
2. Select **NVIC** in NVIC Mode and Configuration
3. Enable **EXTI line0 interrupt**



life.augmented

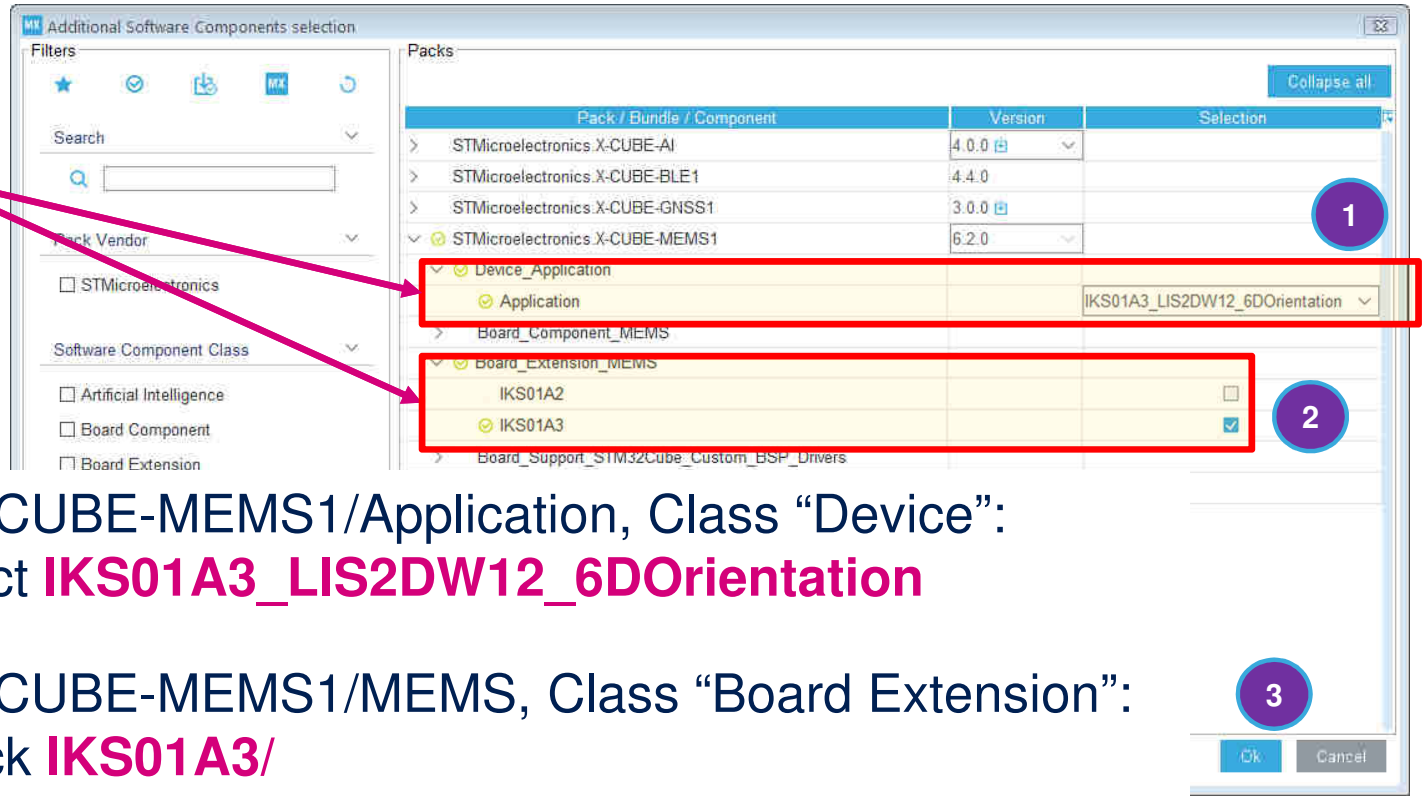
Lab1 – Select the MEMS Library

1. Click on **Additional Software**



Lab1 – Select the MEMS Library

Click to expand tree



1. In X-CUBE-MEMS1/Application, Class “Device”: Select **IKS01A3_LIS2DW12_6DOrientation**

2. In X-CUBE-MEMS1/MEMS, Class “Board Extension”: Check **IKS01A3/**

3. Click **OK**

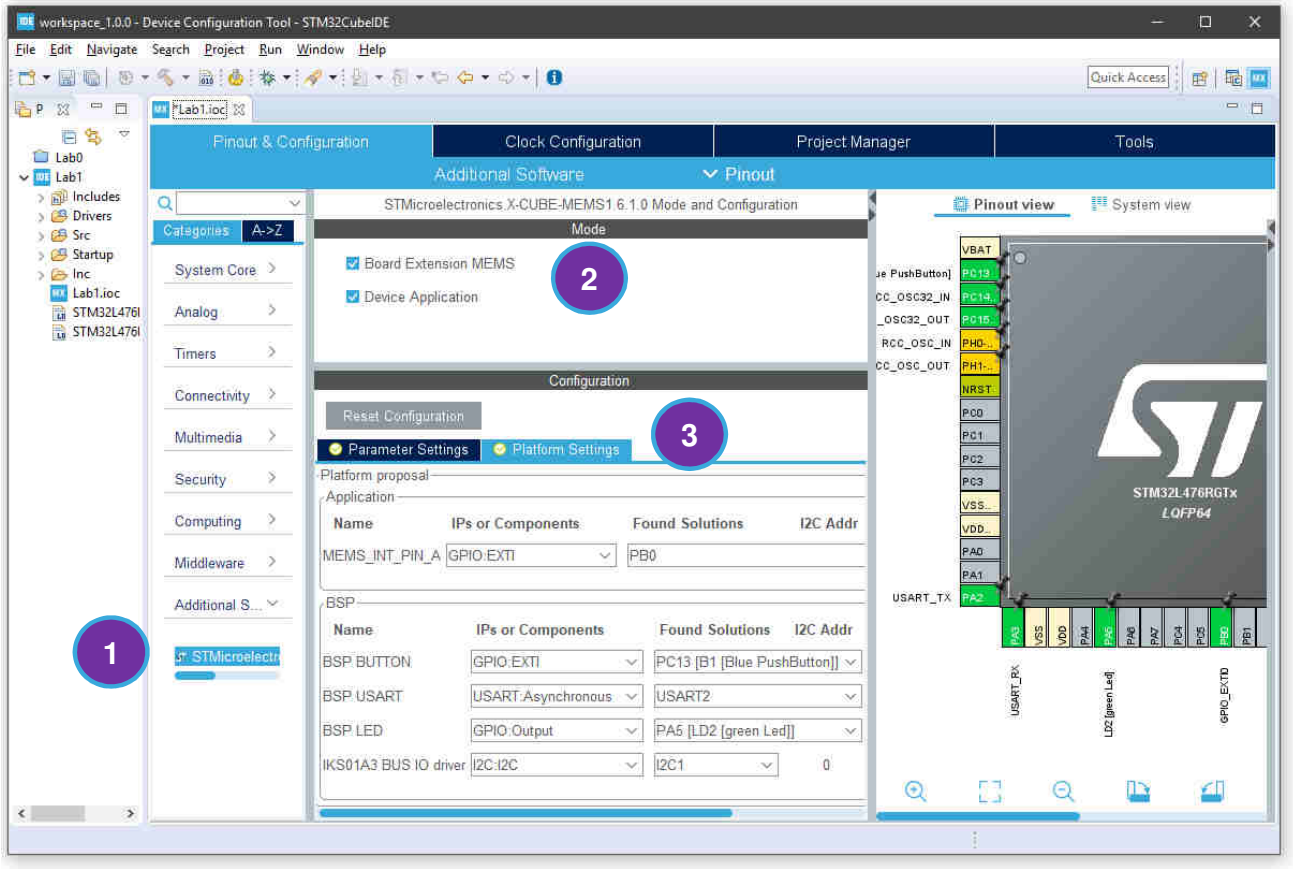
life.augmented

Lab1 – Configure the MEMS Library

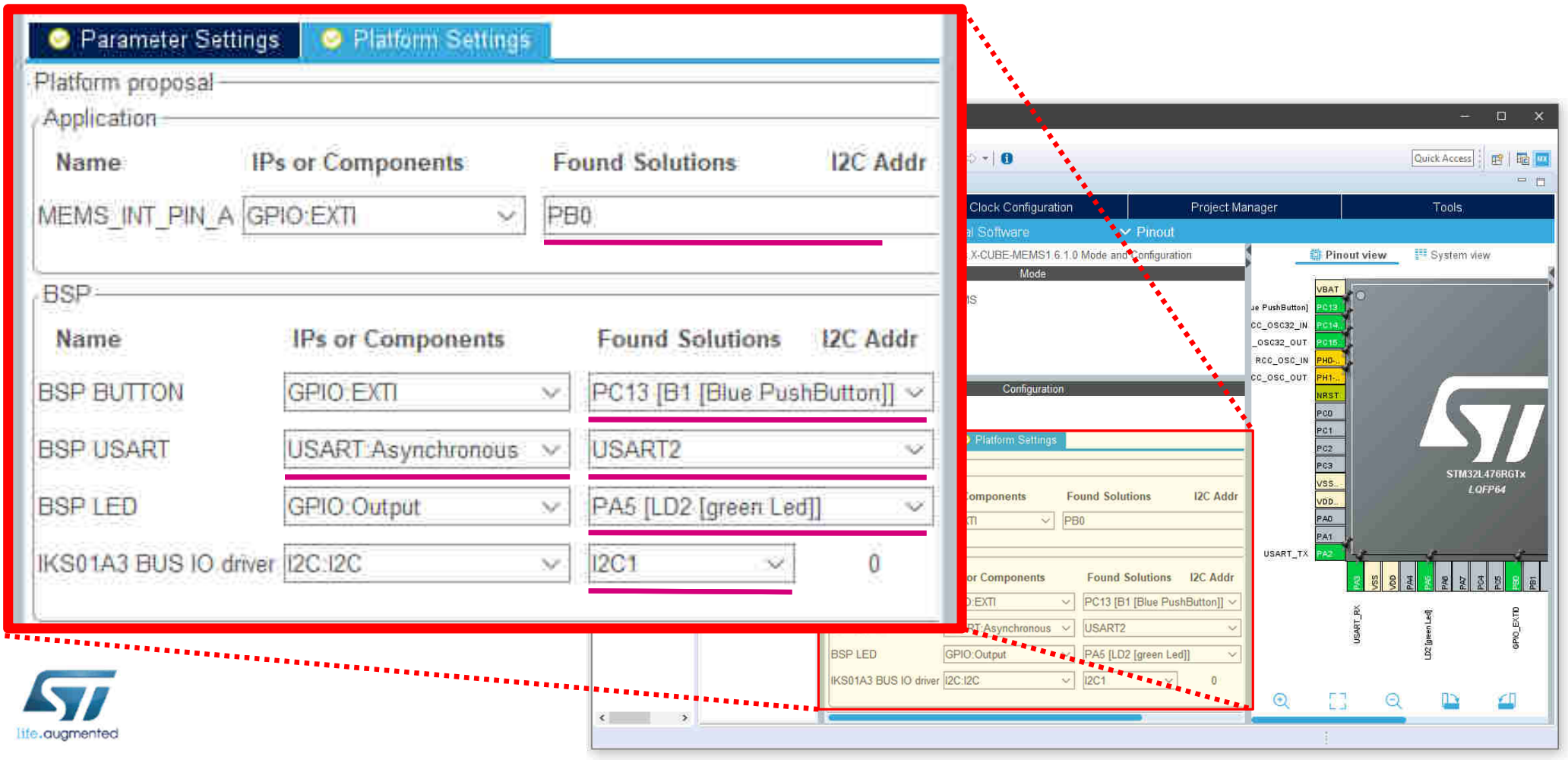
1. Expand Additional Software and select the X-CUBE-MEMS1

2. Check both:
   Board Extension MEMS
   Device Application

3. Configure Platform Settings as in picture



Lab1 – Configure the MEMS Library

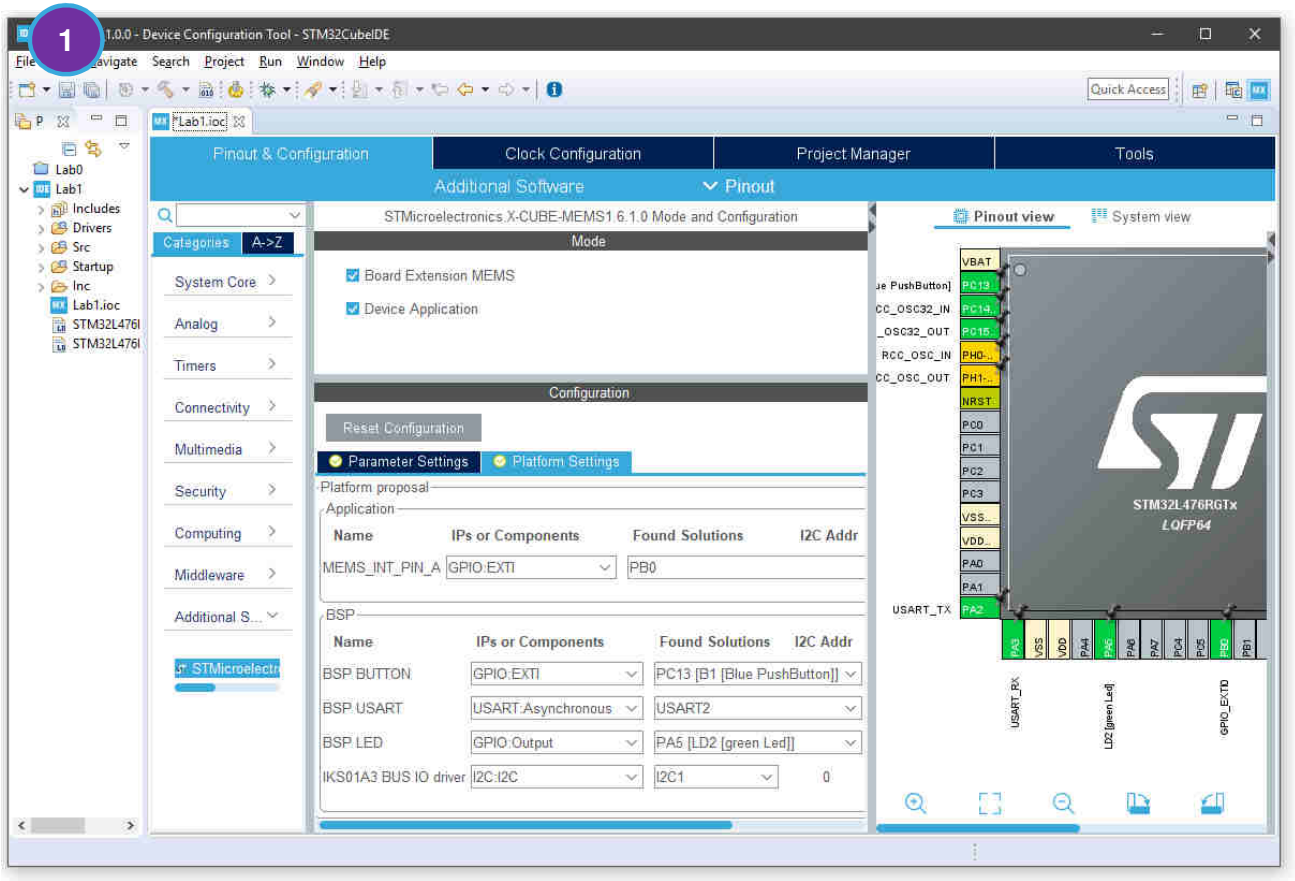


life.augmented

Lab1 – Save the Project

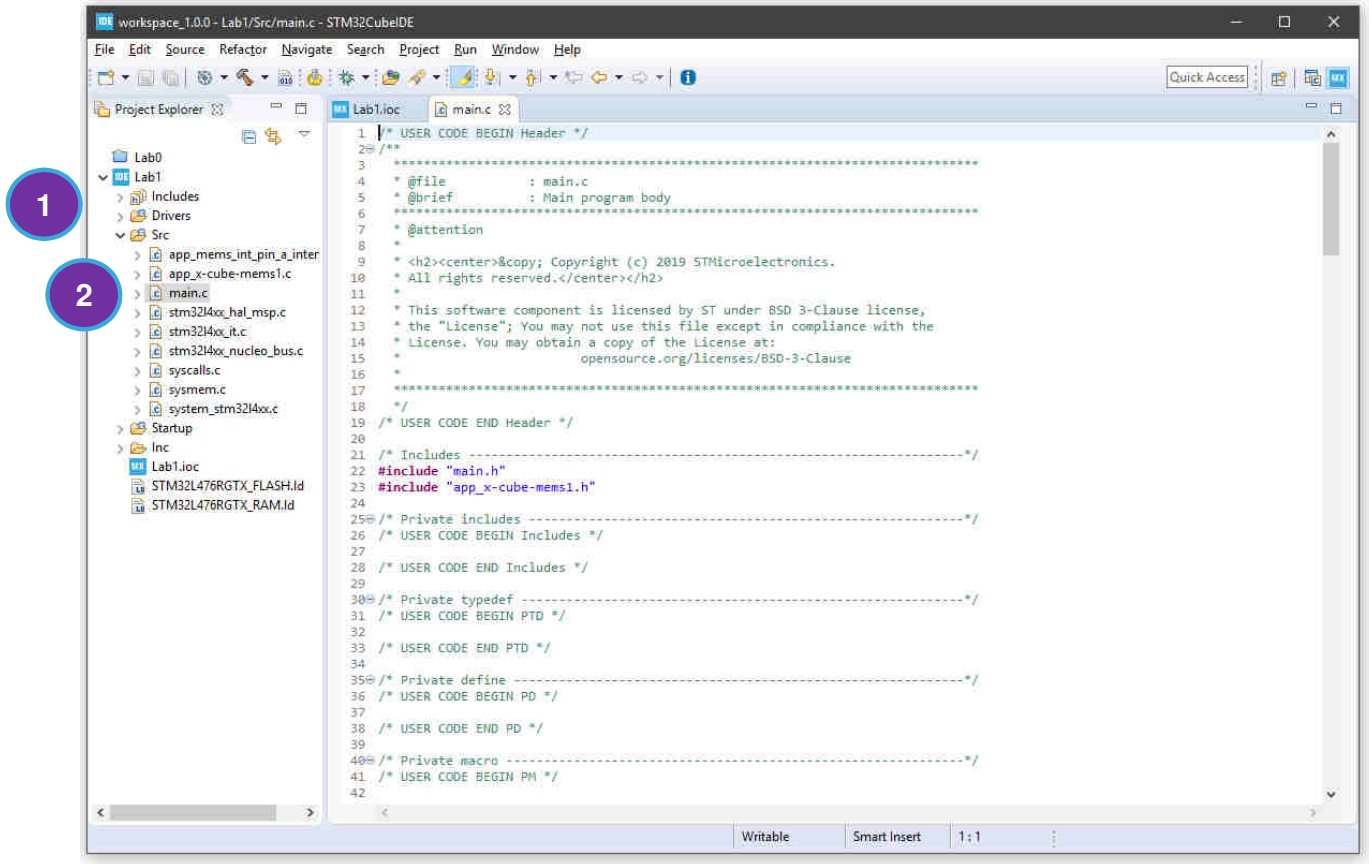
1. Click the save button 

   This action will generate the source code of this lab



Lab1 – Code Editing

1. Expand **Src** in folder **Lab1**
2. Double click on **main.c**

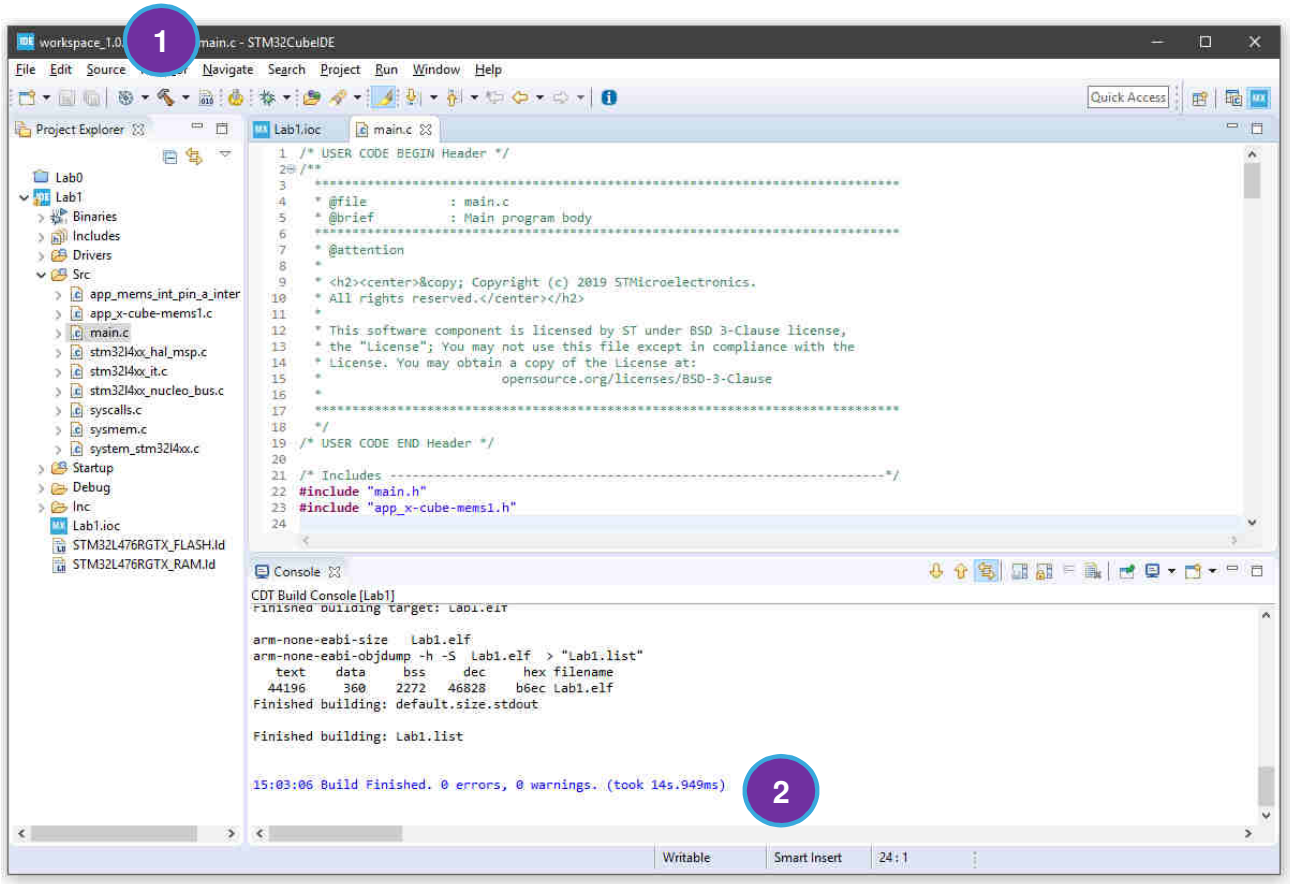


life.augmented

Lab1 - Compiling

1. Click on the hammer  to begin compilation, or press **CTRL+B**

2. Compilation should terminate with 0 errors and 0 warning



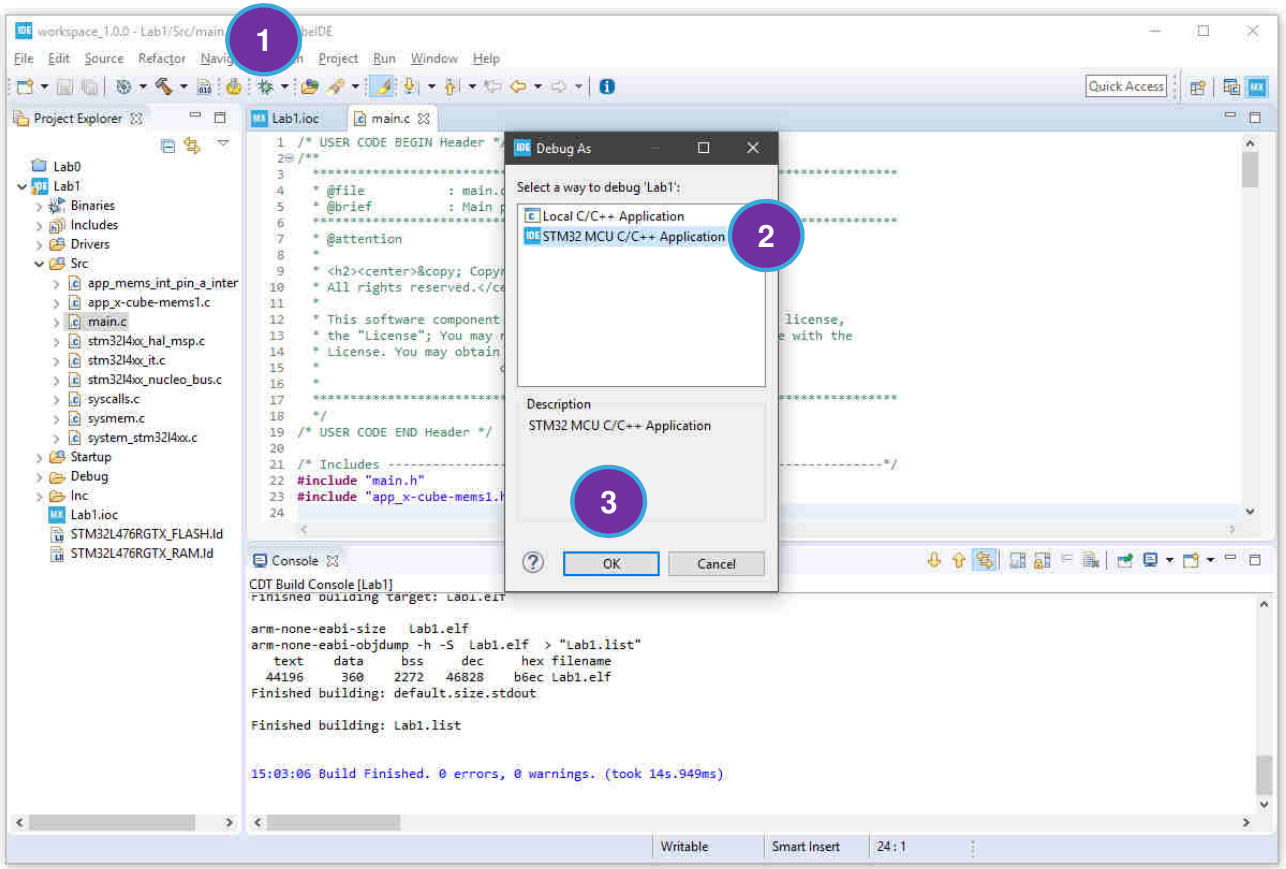
life.augmented

Lab1 - Debugging

1. Click on the bug  to begin debugging

2. Select
   STM32 MCU C/C++ App

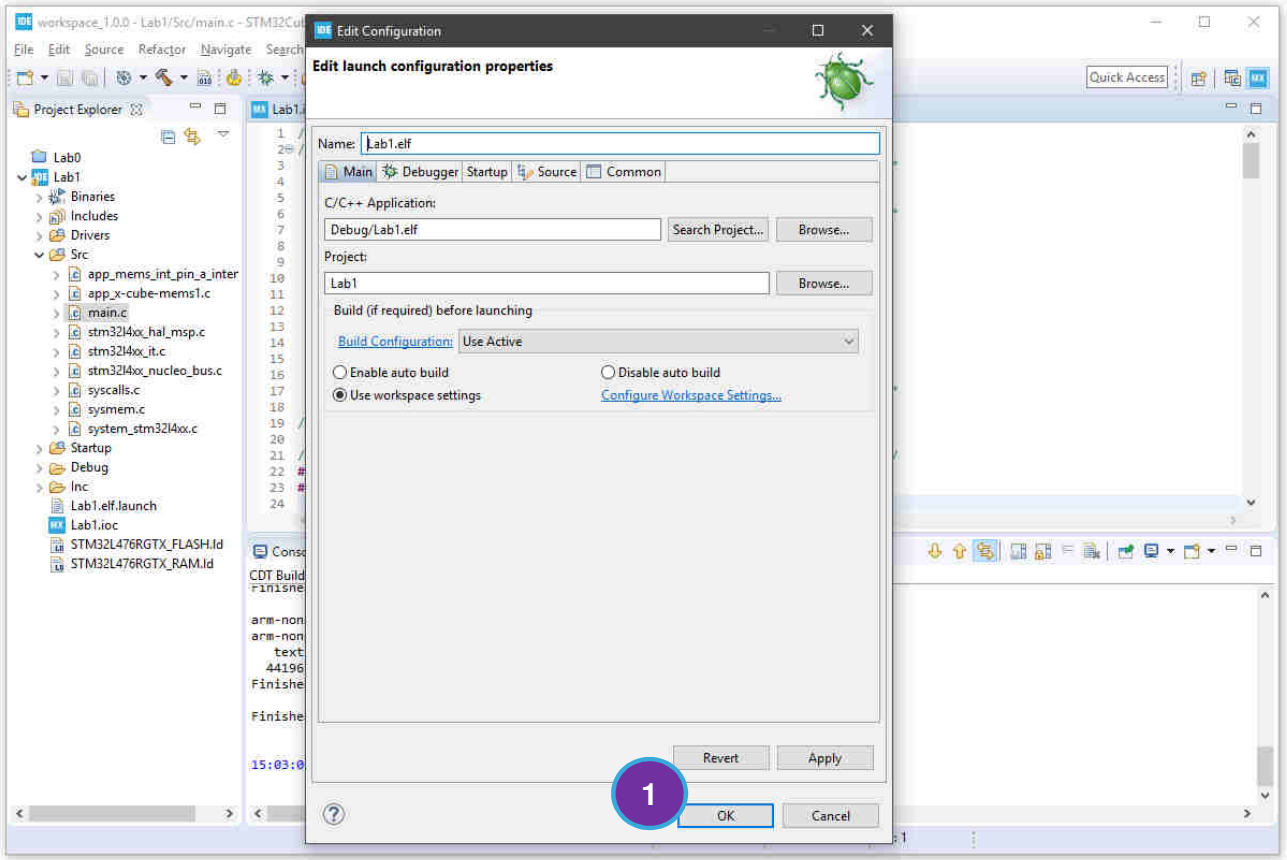
3. Click **OK**



life.augmented

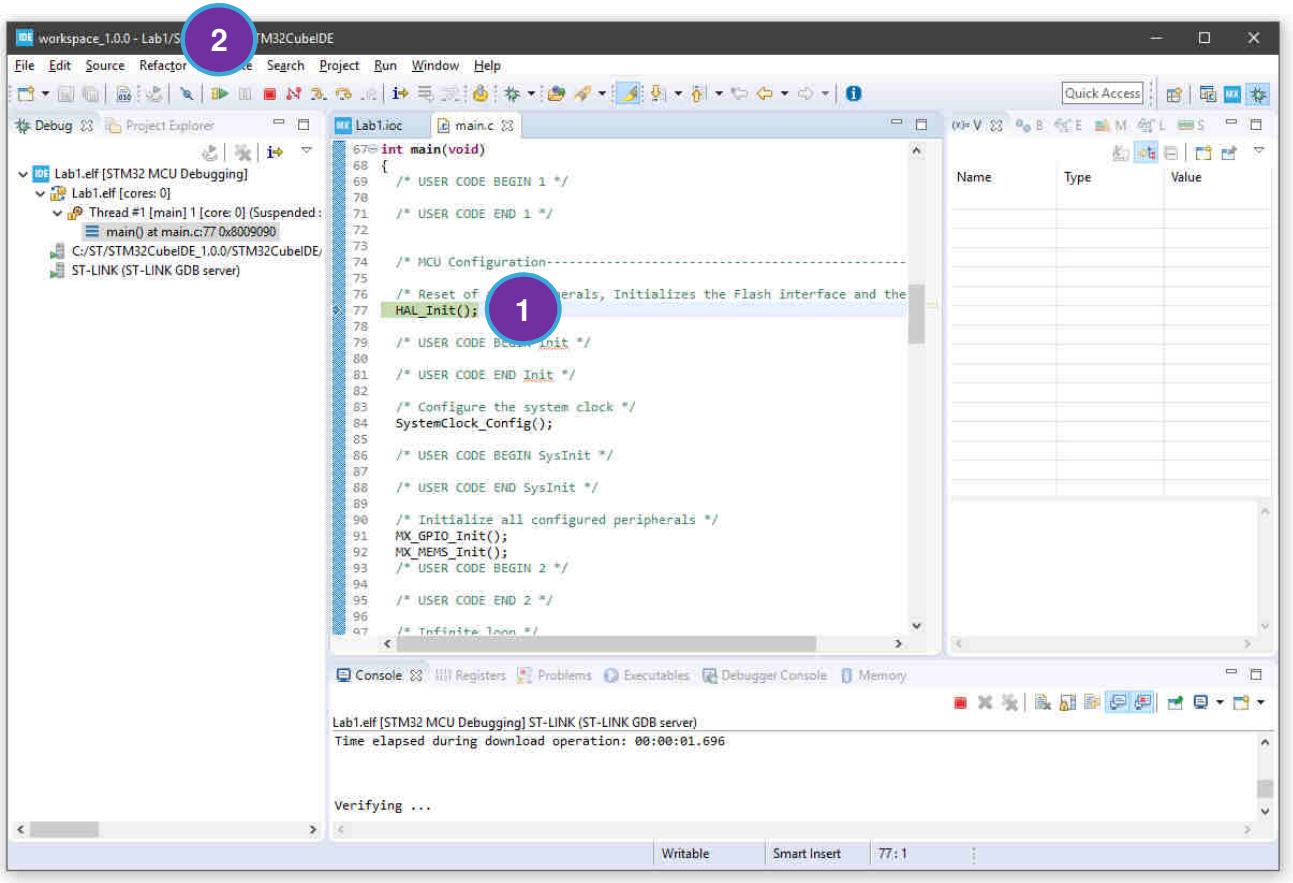
Lab1 - Debugging

1. Click **OK**



Lab1 - Debugging

1. Code start at the first line of the main function
2. Click play  button to run the code



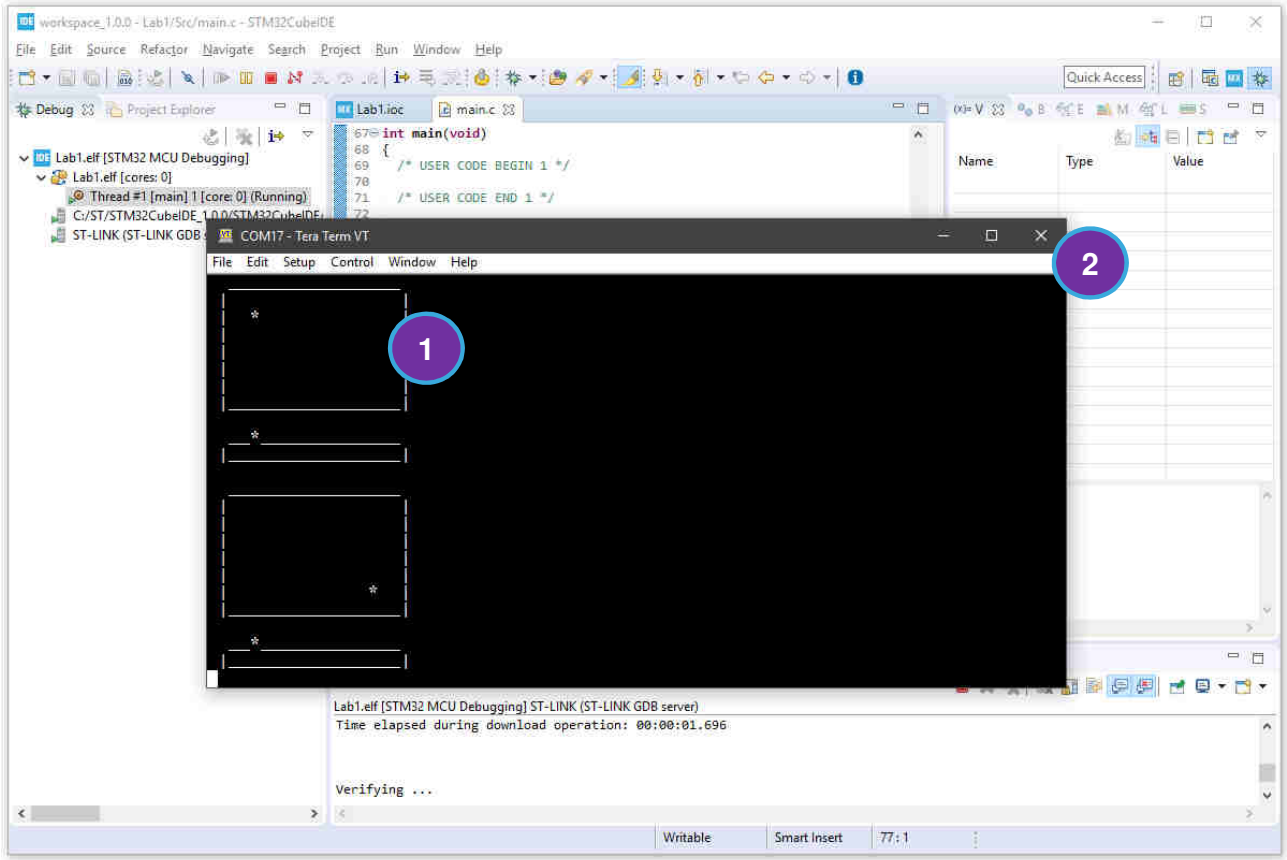
life.augmented

Lab1 - Testing

1. Open Tera Term to view the output

Rotate the board to see the output changing

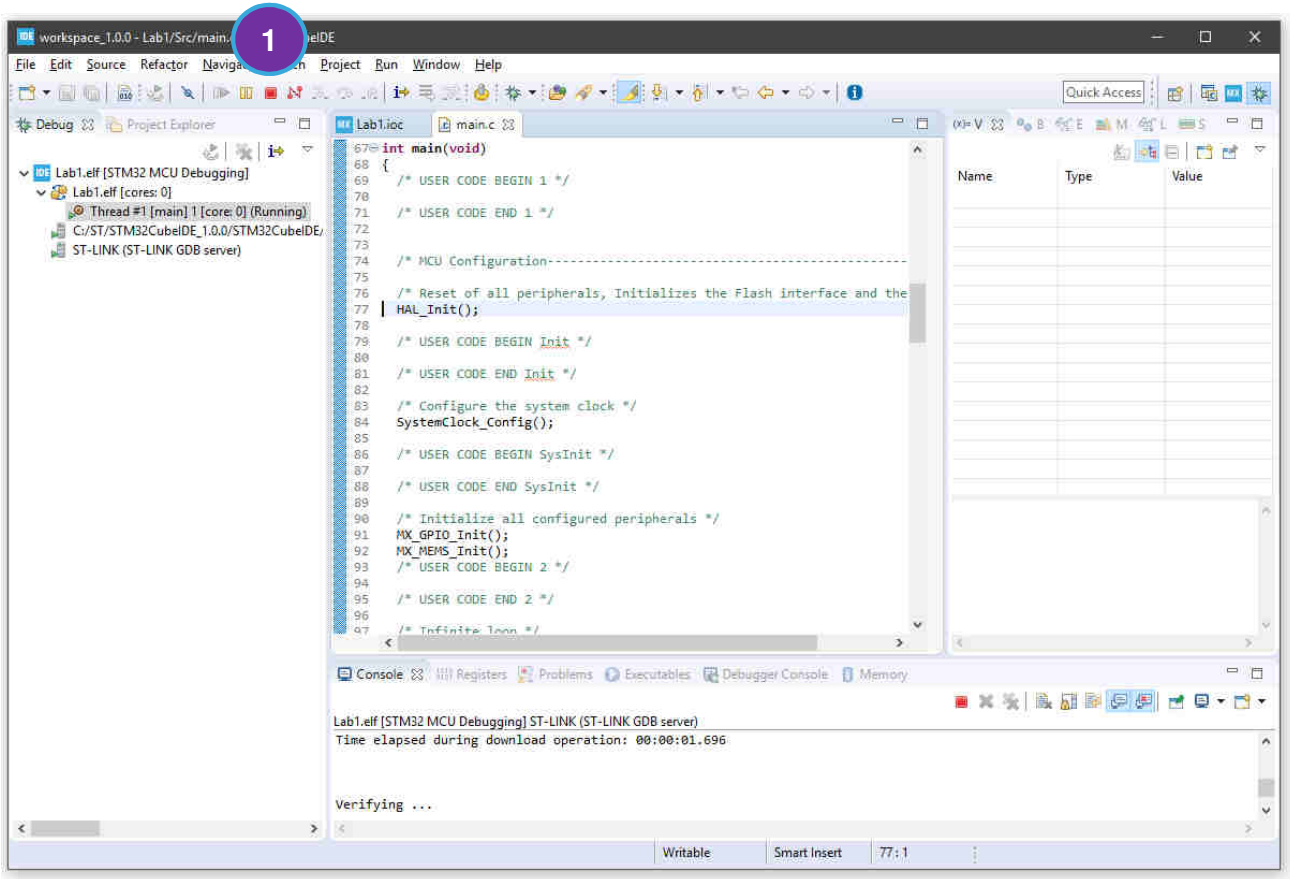
2. After testing close Tera Term by clicking **X**



life.augmented

Lab1 - Debugging

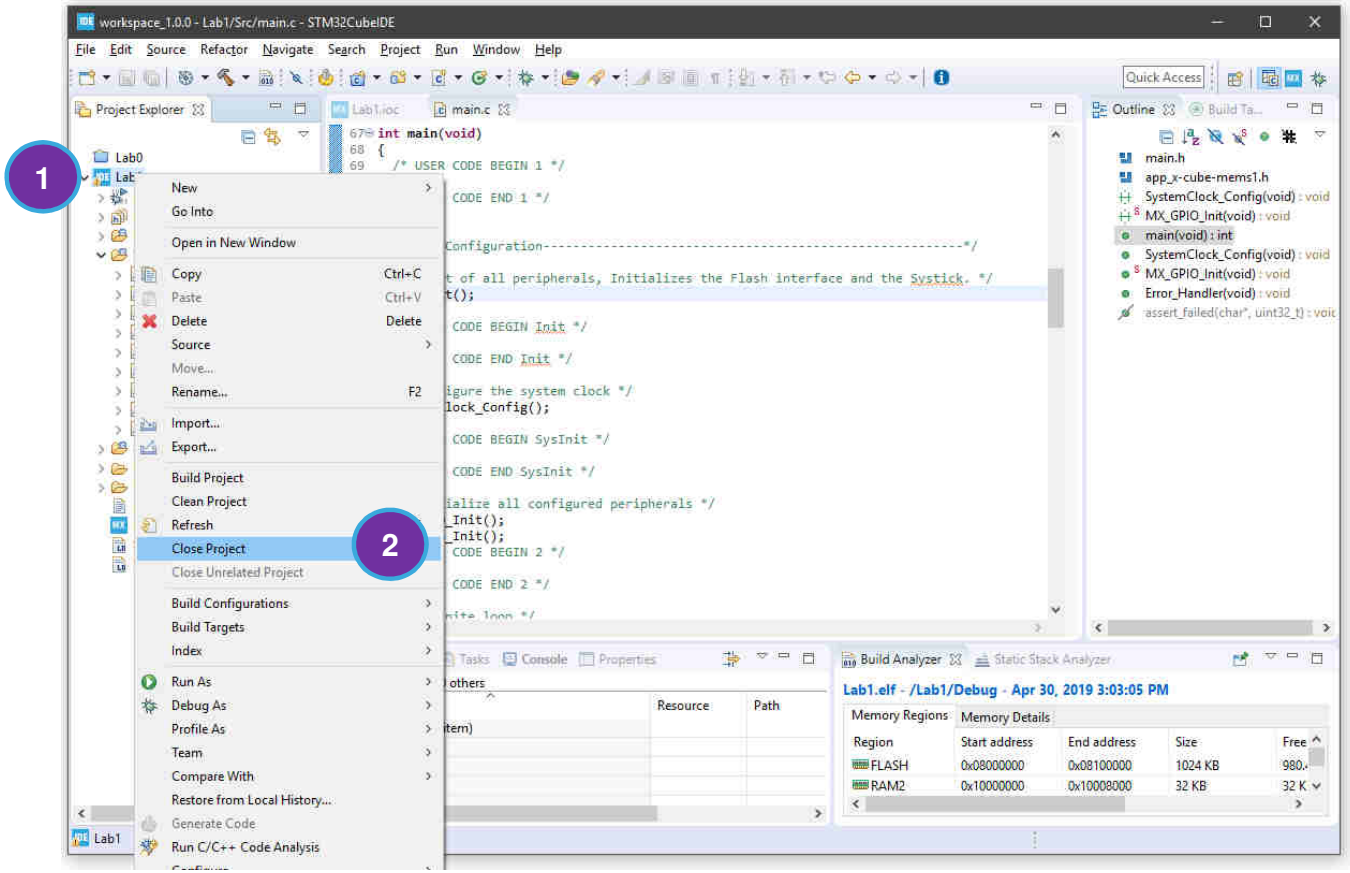
1. Click stop  button to interrupt the debugging



life.augmented

Lab1 – Closing the Project

1. Right-Click on **Lab1** project
2. Click on **Close Project**



life.augmented

LAB2

Goals:

- Configure a new project using X-CUBE-MEMS1
- Configure LPS22HH pressure and temperature sensor to acquire data using internal FIFO
- Enable interrupts in STM32CubeIDE

life.augmented

LPS22HH

High-performance, high-ODR Barometer / Altimeter

- Absolute Accuracy: 0.5hPa
- Relative Accuracy: ±0.025hPa
- RMS Noise: ±0.65Pa (0.0065hPa)
- ODR up to 200Hz
- Embedded Temperature compensation
- Unique Full Molded Package

High Resolution - ~5cm

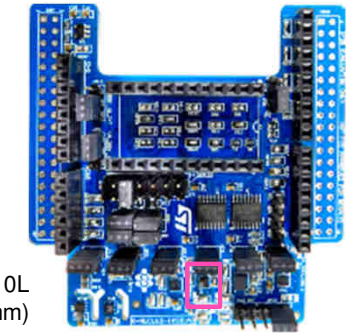
High Performance: excellent noise figure

Fast Response

System Power Saving

Robustness

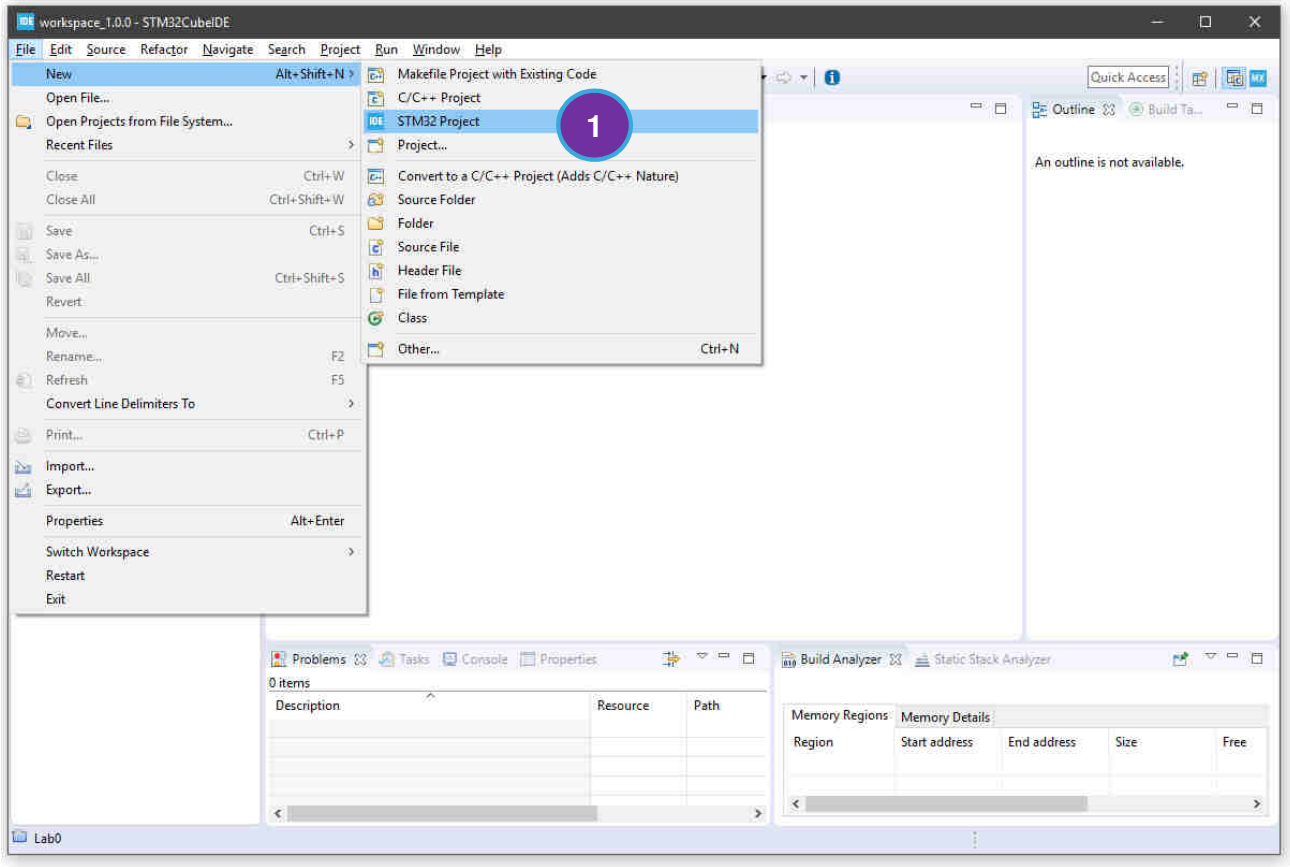
Skip One Point Cal.
post Soldering
→ Cost Saving



HLGA-10L
(2 x 2 x 0.73 mm)

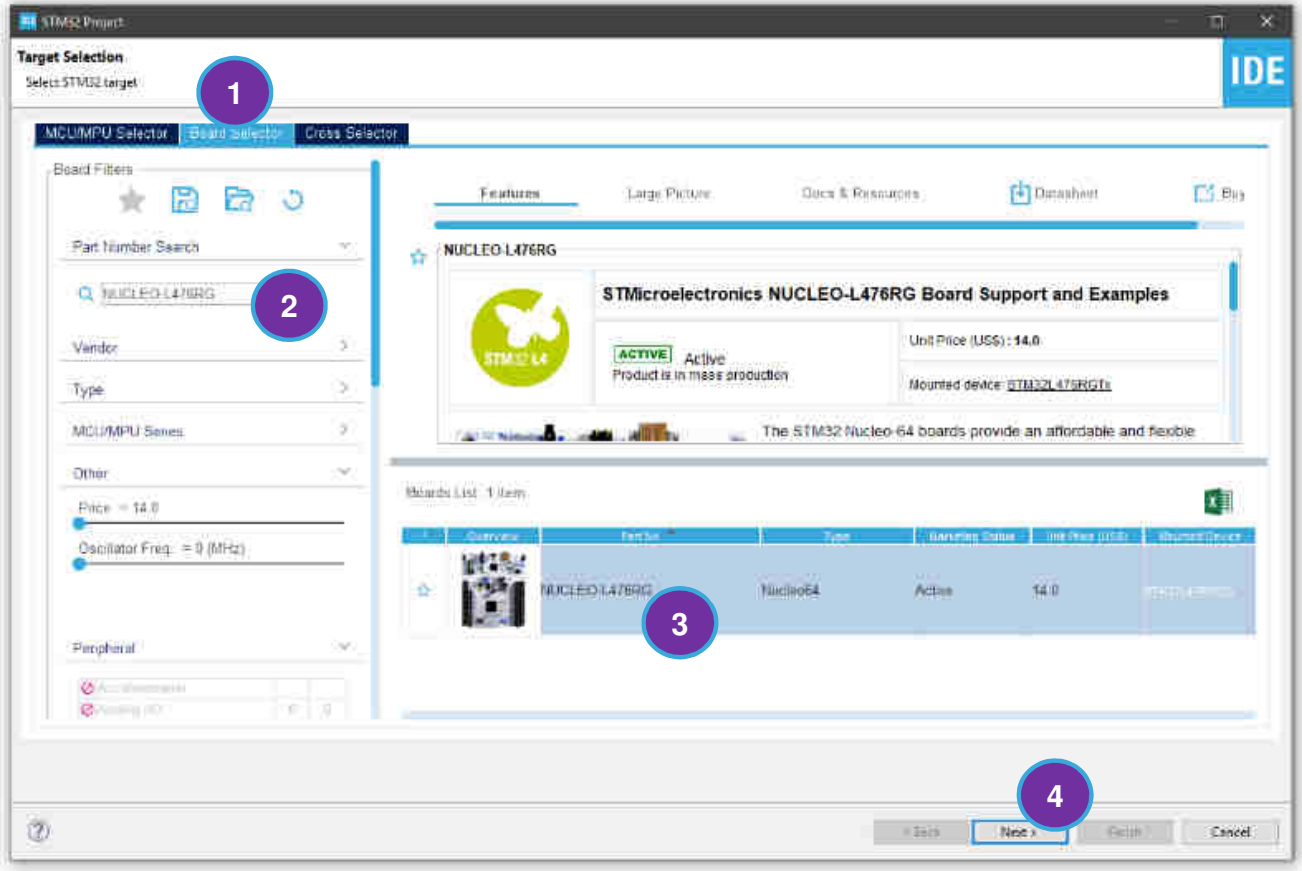
life.augmented

1. Click on **File > New > STM32 Project**



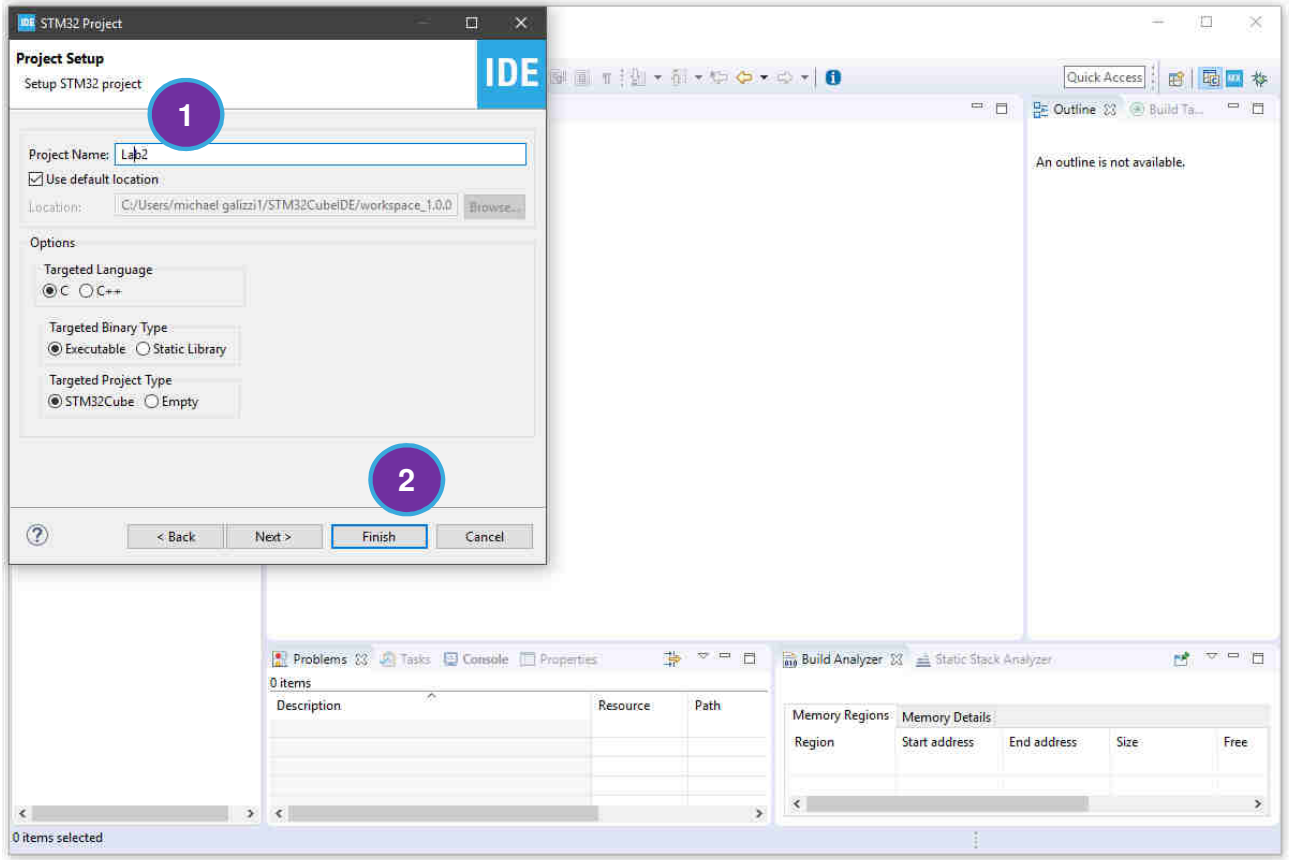
life.augmented

1. Click on **Board Selector**
2. Type **NUCLEO-L476RG**
3. Click on the board
4. Click **Next >**

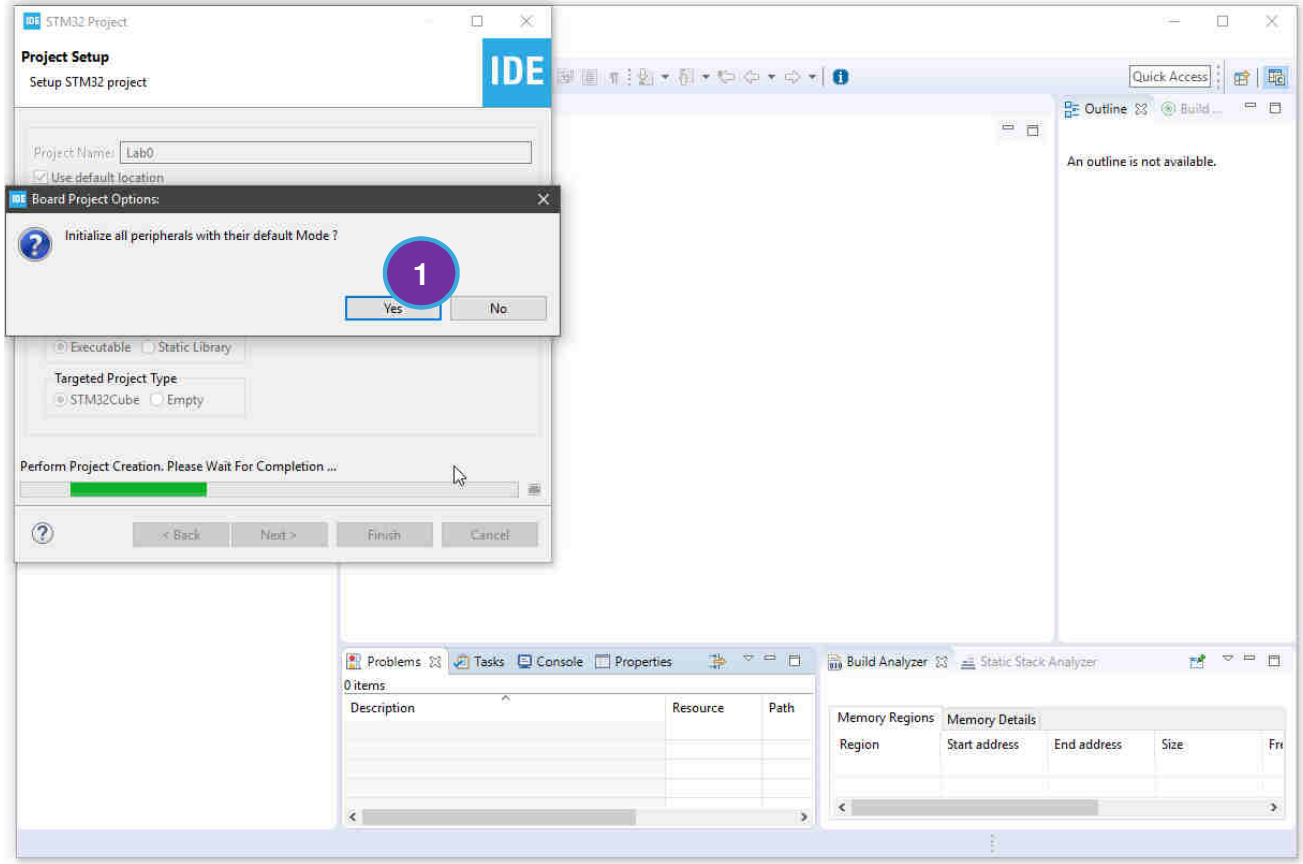


life.augmented

1. Project Name **Lab2**
2. Click **Finish**



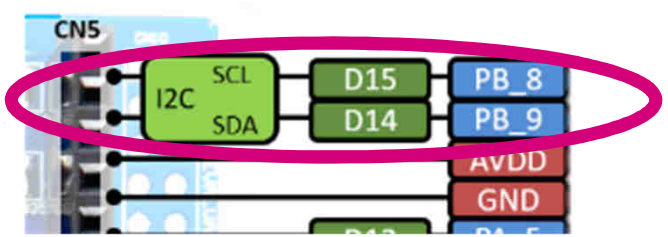
1. Click **Yes** to init peripherals in default mode

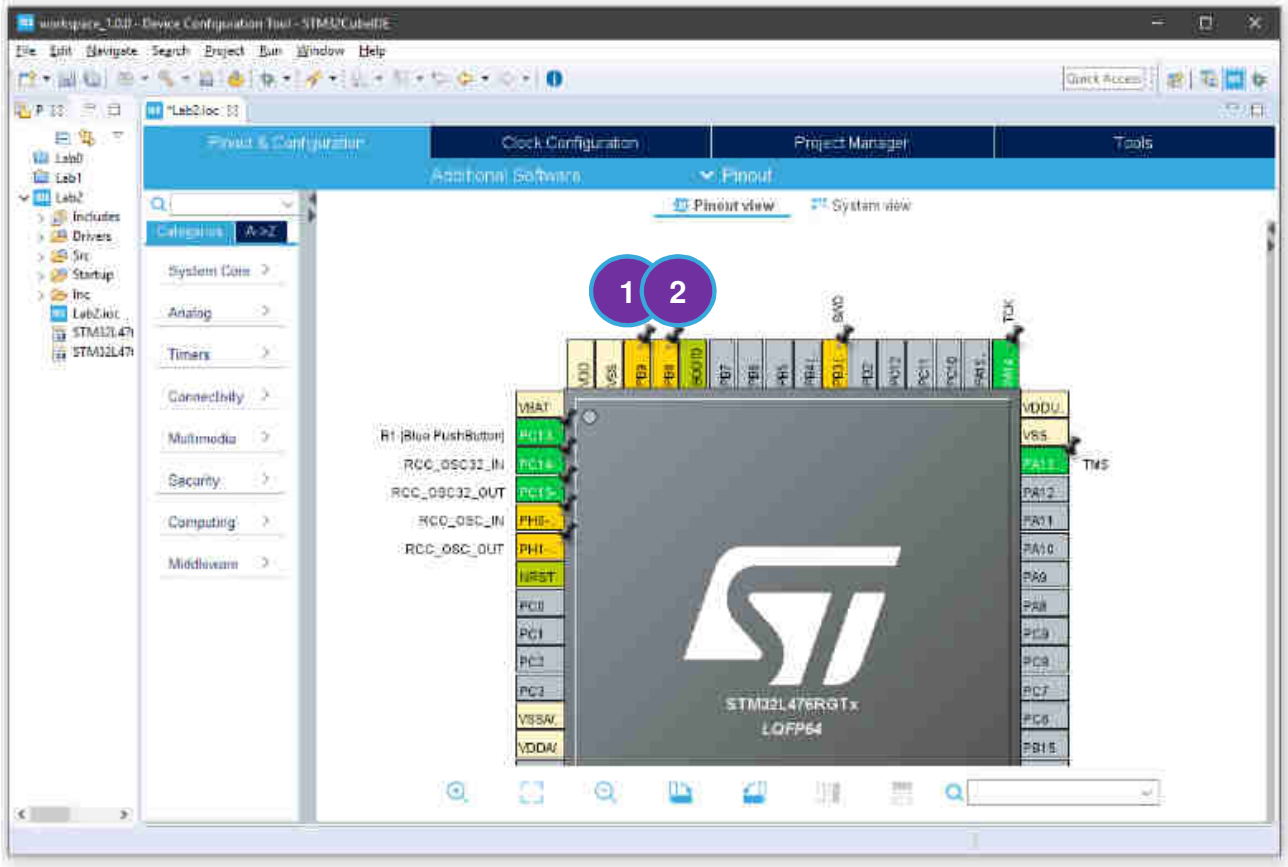


life.augmented

Lab2 – Configure the I2C Bus

1. Left Click on **PB9** and select I2C1_SDA
2. Left Click on **PB8** and select I2C1_SCL

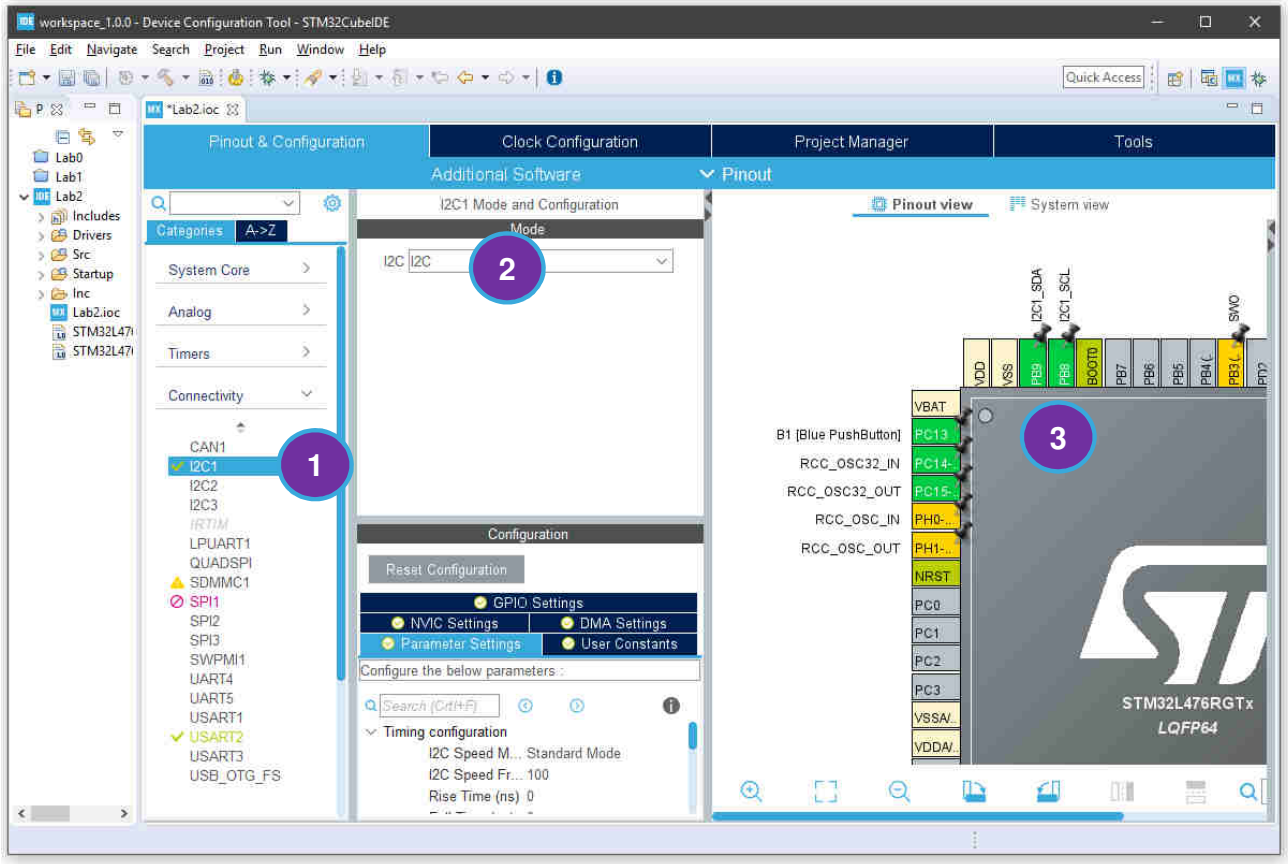




life.augmented

Lab2 – Configure the I2C Bus

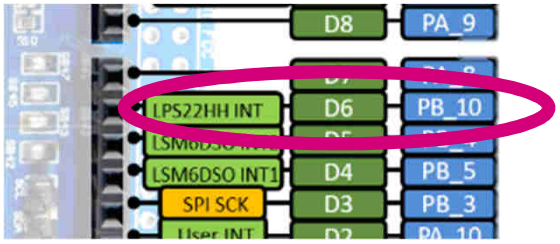
1. Expand *Connectivity* tab and check **I2C1**
2. Select **I2C** in *I2C1 Mode and Configuration*
3. PB8 and PB9 should now become green

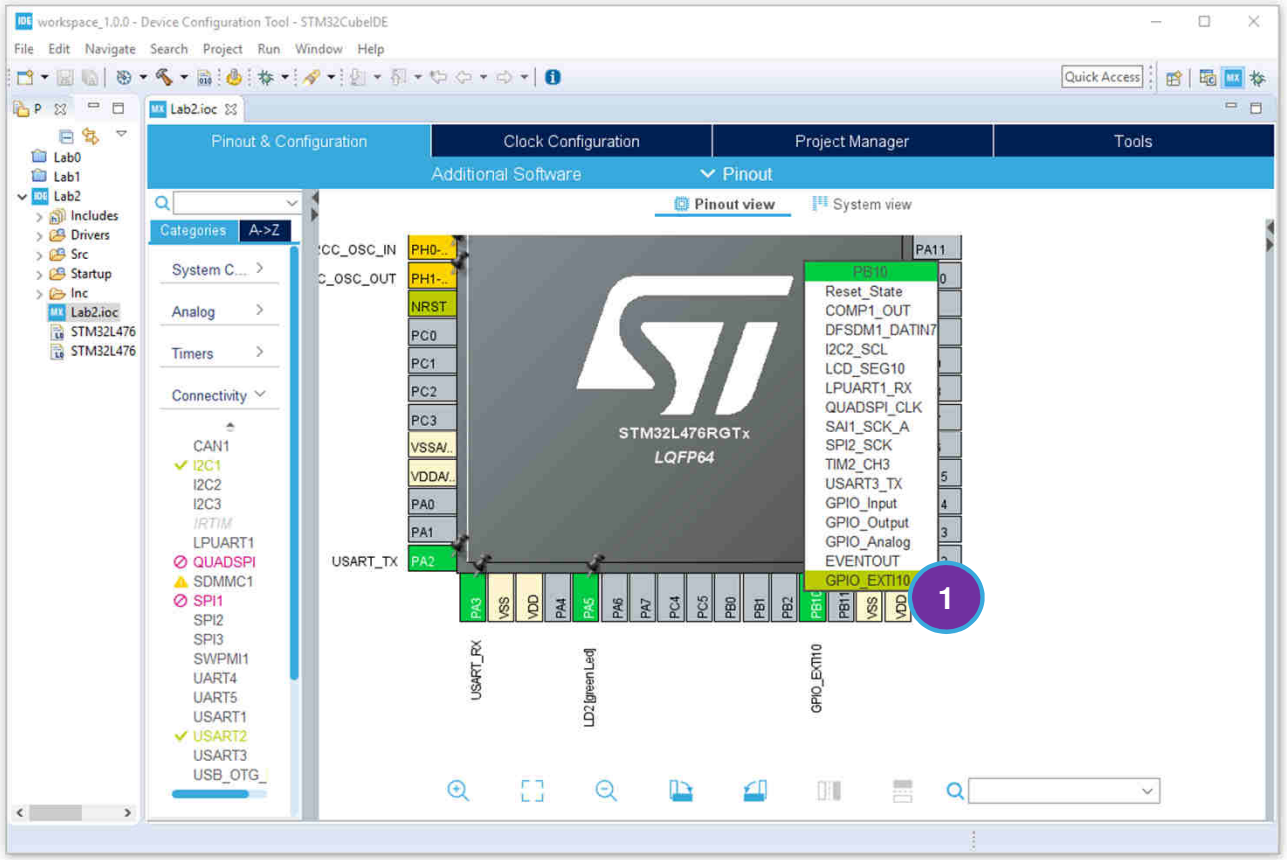


life.augmented

Lab2 – Configure LPS22HH Interrupt

1. Left Click on **PB10** and select **GPIO_EXTI10**

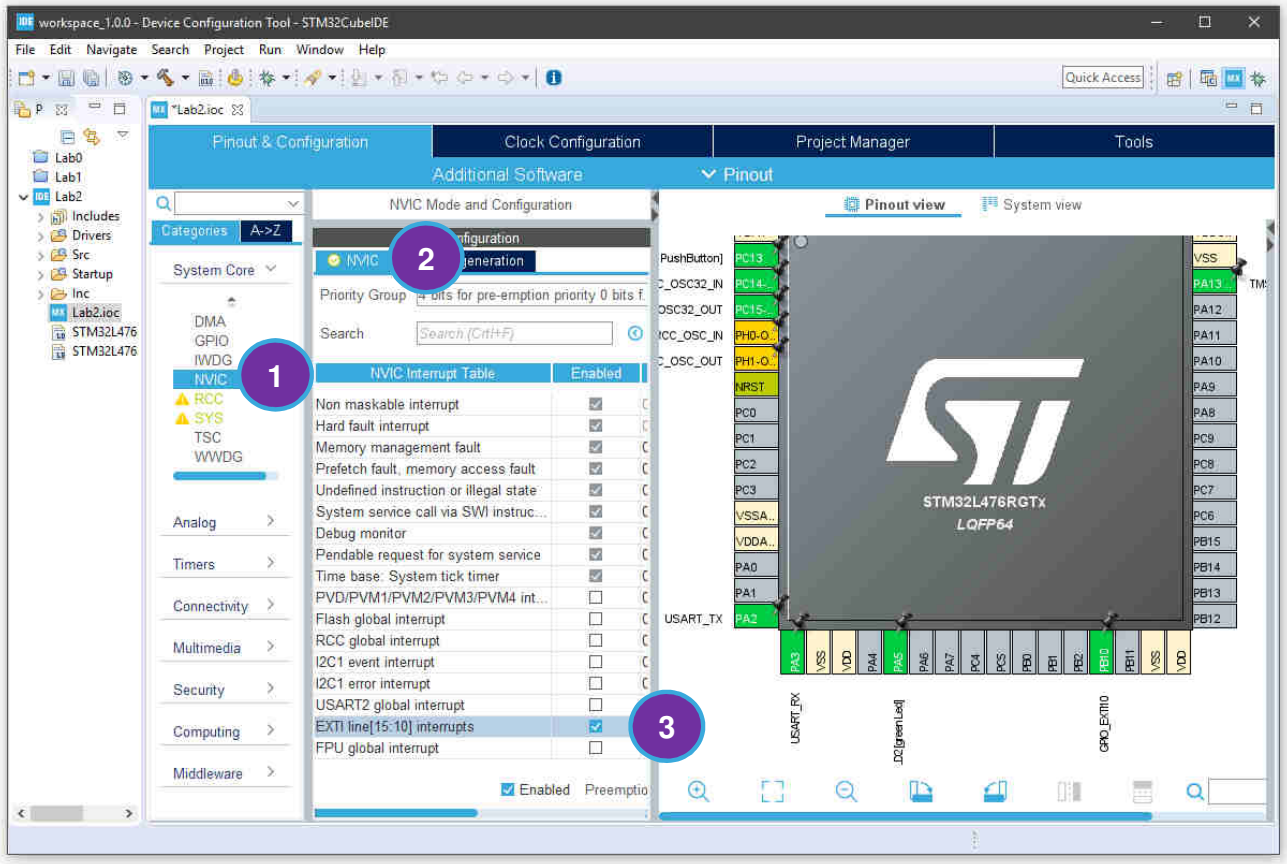




life.augmented

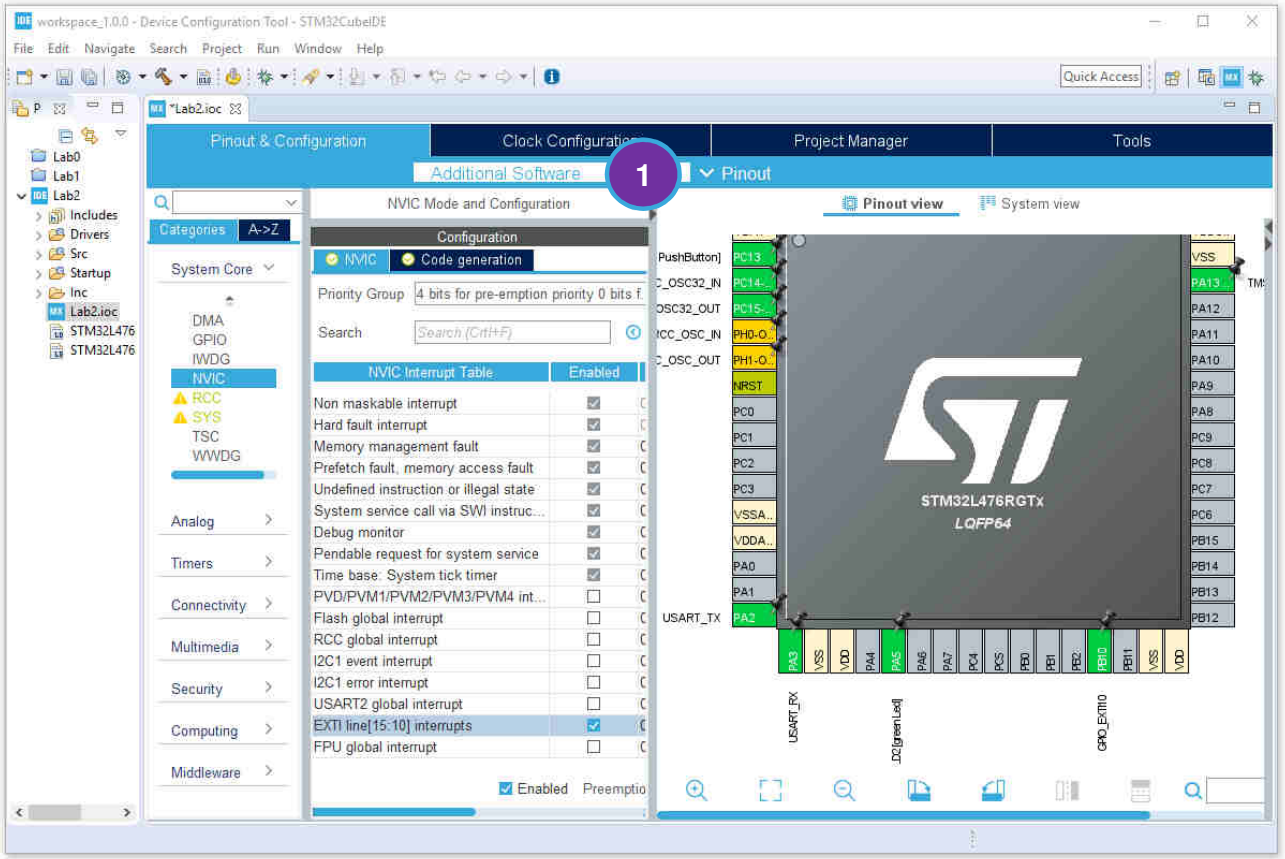
Lab2 – Configure LPS22HH Interrupt

1. Check **NVIC** in tab System Core
2. Select **NVIC** in NVIC Mode and Configuration
3. Enable **EXTI line[15:10] interrupt**



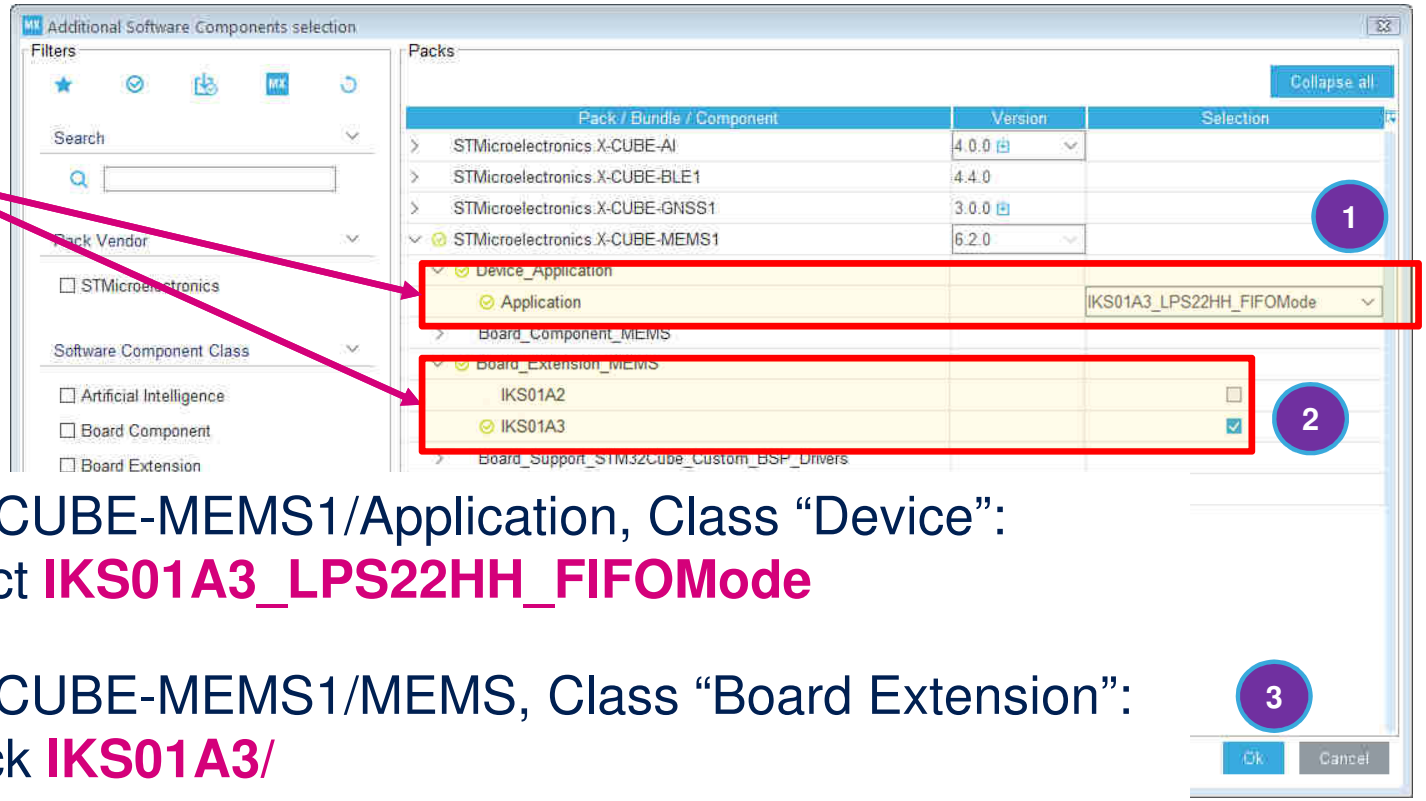
Lab2 – Select the MEMS Library

1. Click on
   Additional Software



Lab2 – Select the MEMS Library

Click to expand tree



1. In X-CUBE-MEMS1/Application, Class “Device”: Select **IKS01A3_LPS22HH_FIFOMode**

2. In X-CUBE-MEMS1/MEMS, Class “Board Extension”: Check **IKS01A3/**

3. Click **OK**

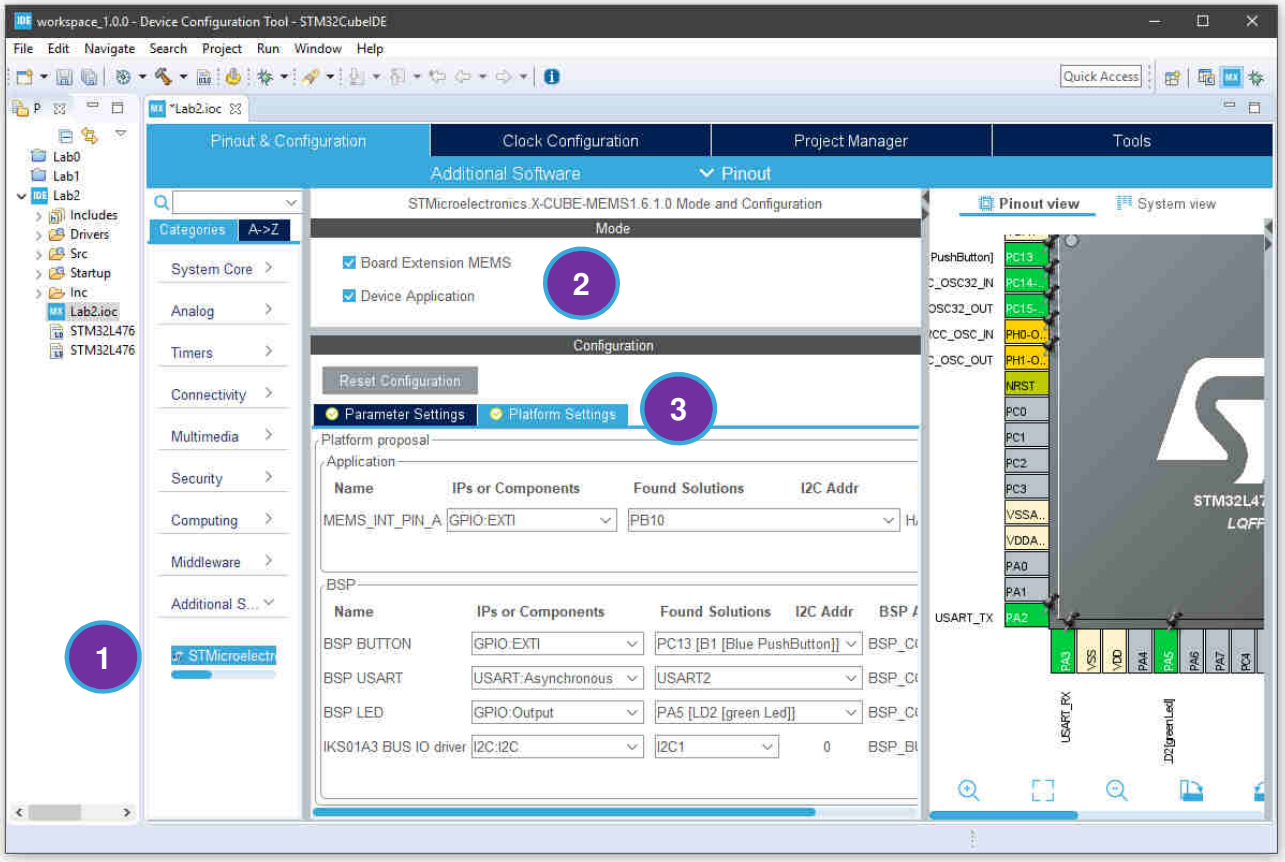
life.augmented

Lab2 – Configure the MEMS Library

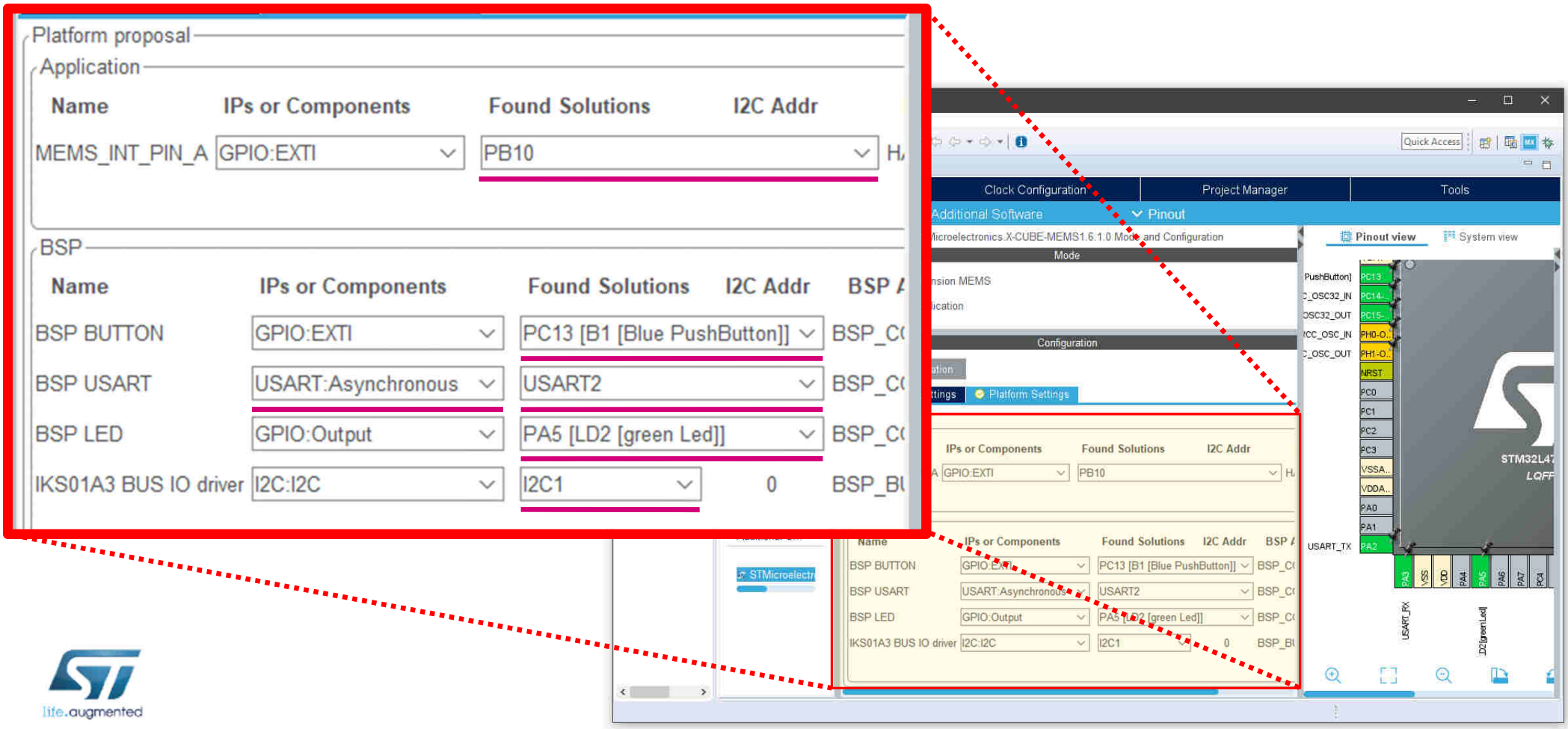
1. Expand Additional Software and select the X-CUBE-MEMS1

2. Check both:
   Board Extension MEMS
   Device Application

3. Configure Platform Settings as in picture



Lab2 – Configure the MEMS Library

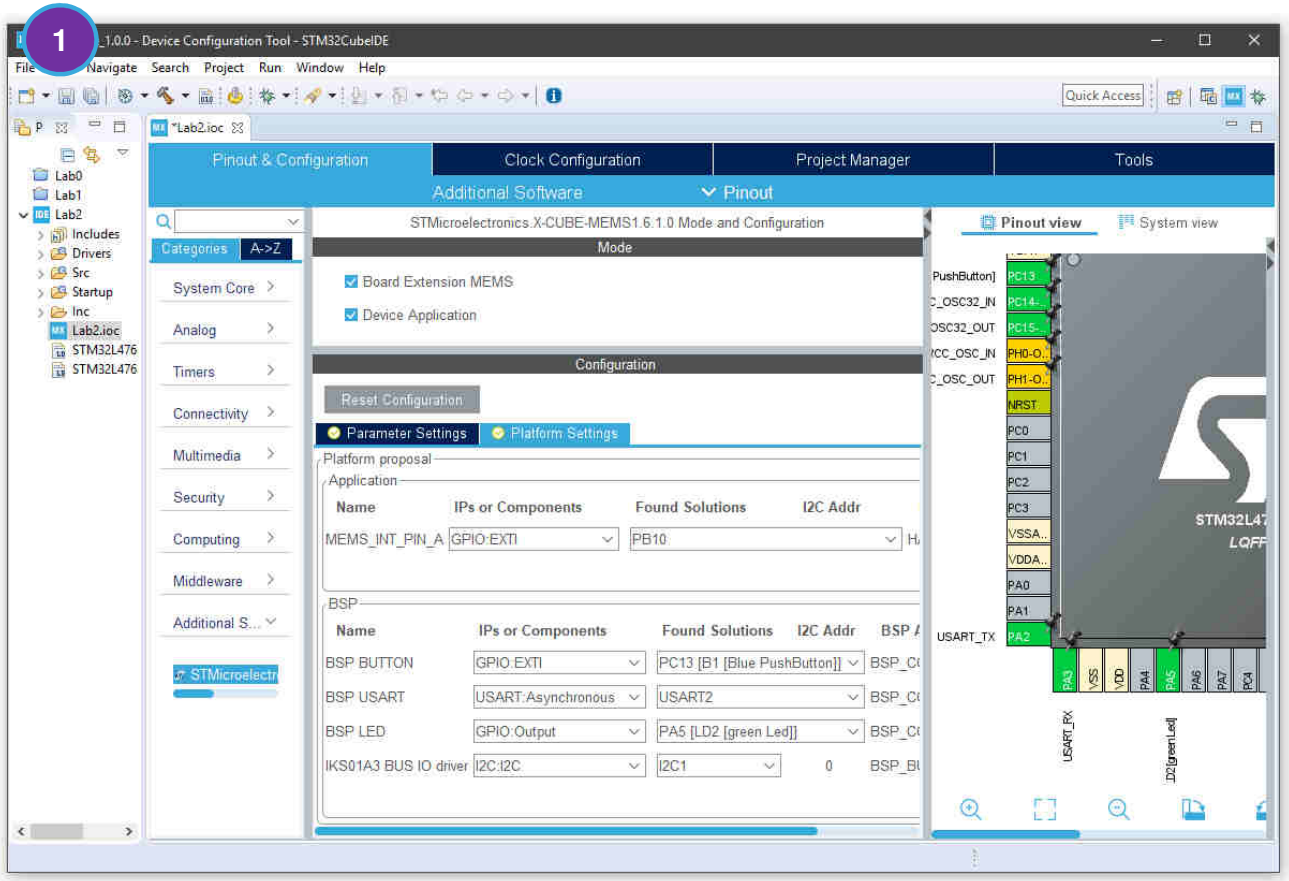


life.augmented

Lab2 – Save the Project

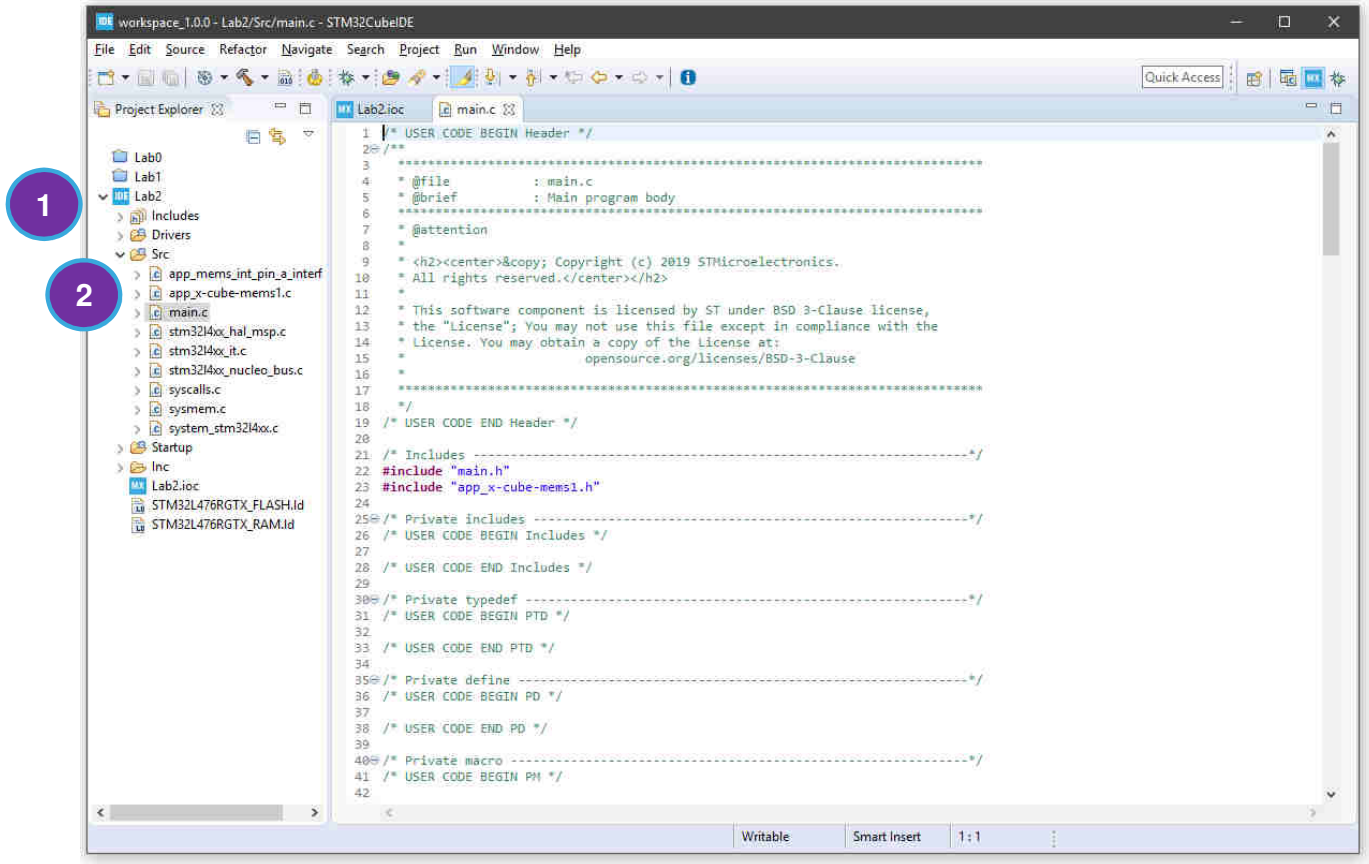
1. Click the save button 

   This action will generate the source code of this lab



Lab2 – Code Editing

1. Expand **Src** in folder **Lab2**
2. Double click on **main.c**

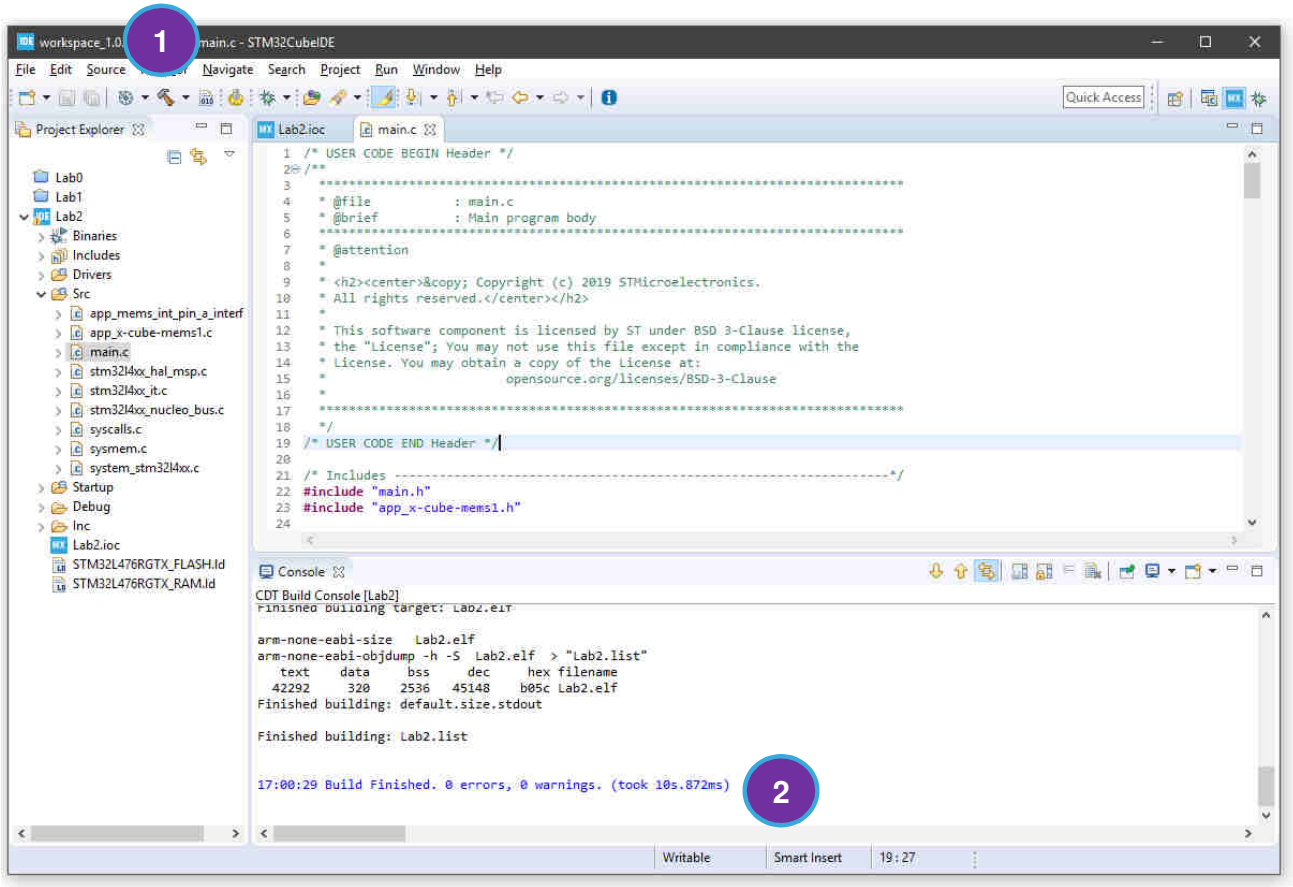


life.augmented

Lab2 - Compiling

1. Click on the hammer  to begin compilation, or press **CTRL+B**

2. Compilation should terminate with 0 errors and 0 warning



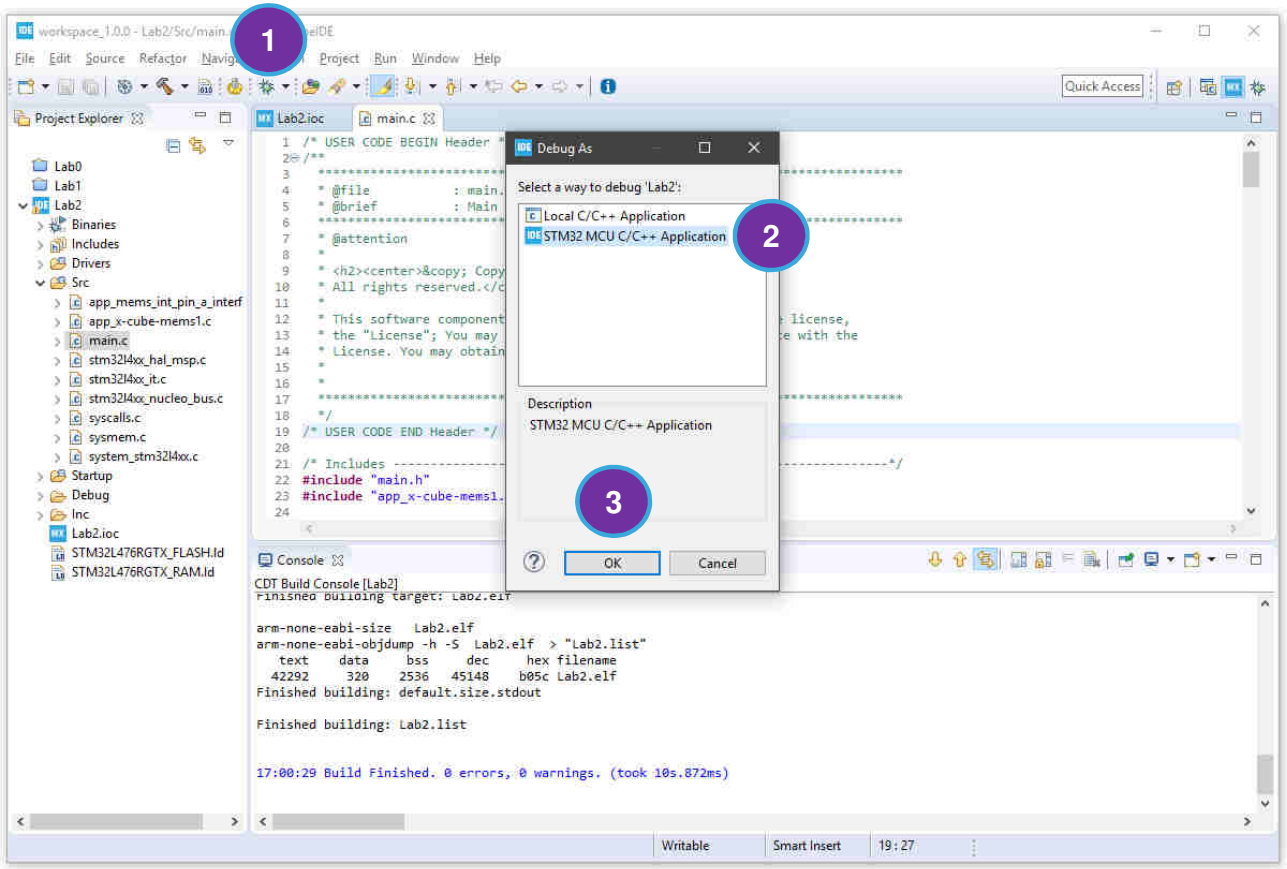
life.augmented

Lab2 - Debugging

1. Click on the bug  to begin debugging

2. Select
   STM32 MCU C/C++ App

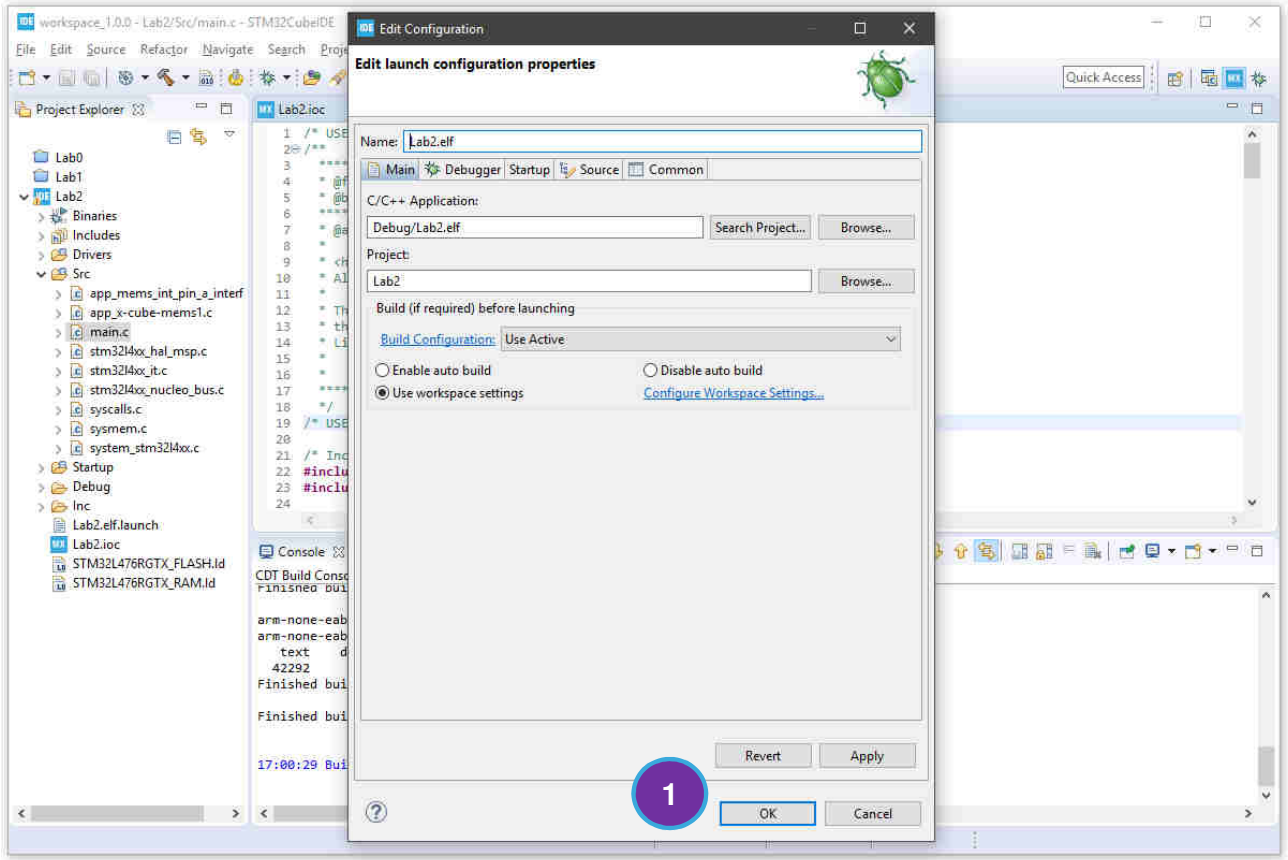
3. Click **OK**



life.augmented

Lab2 - Debugging

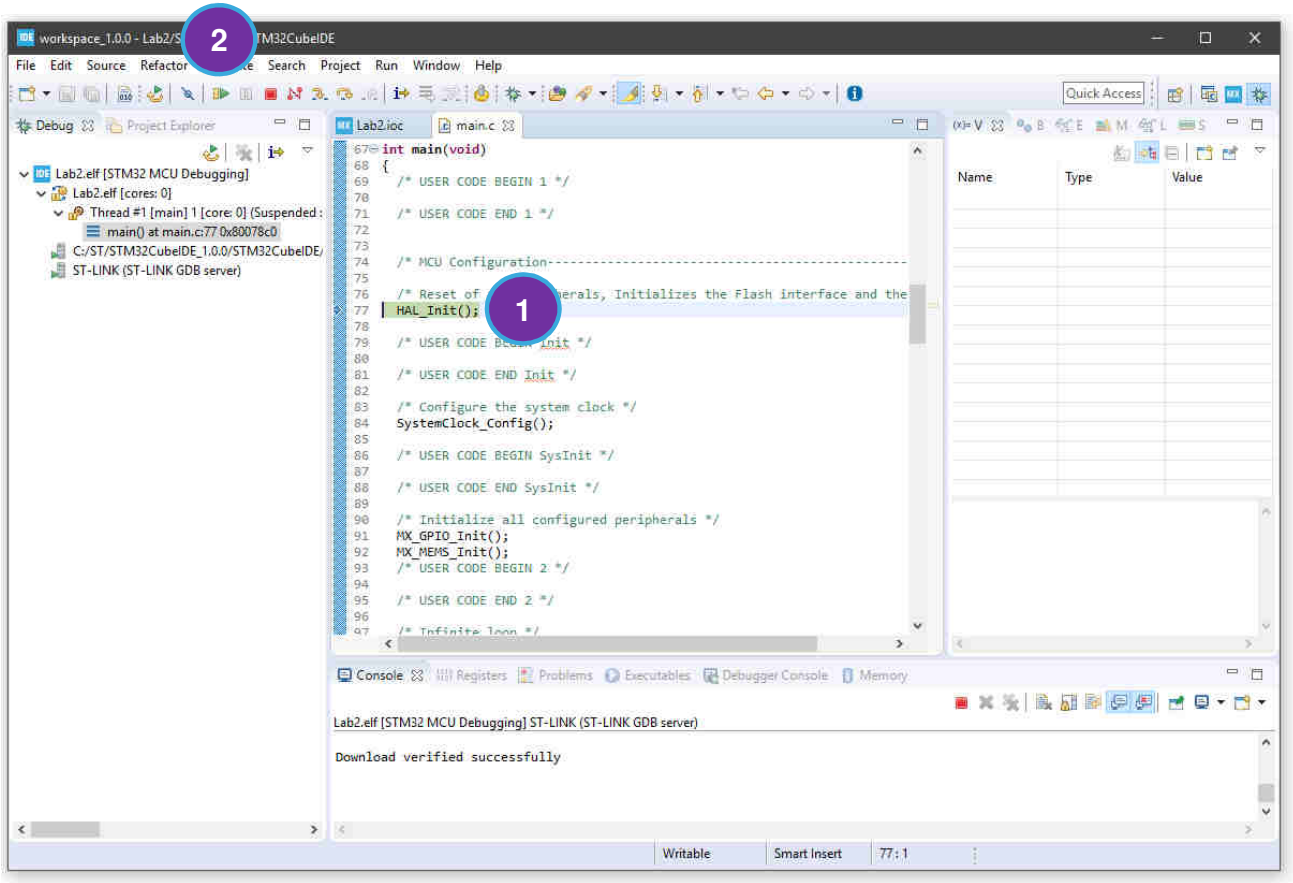
1. Click **OK**



life.augmented

Lab2 - Debugging

1. Code start at the first line of the main function
2. Click play  button to run the code



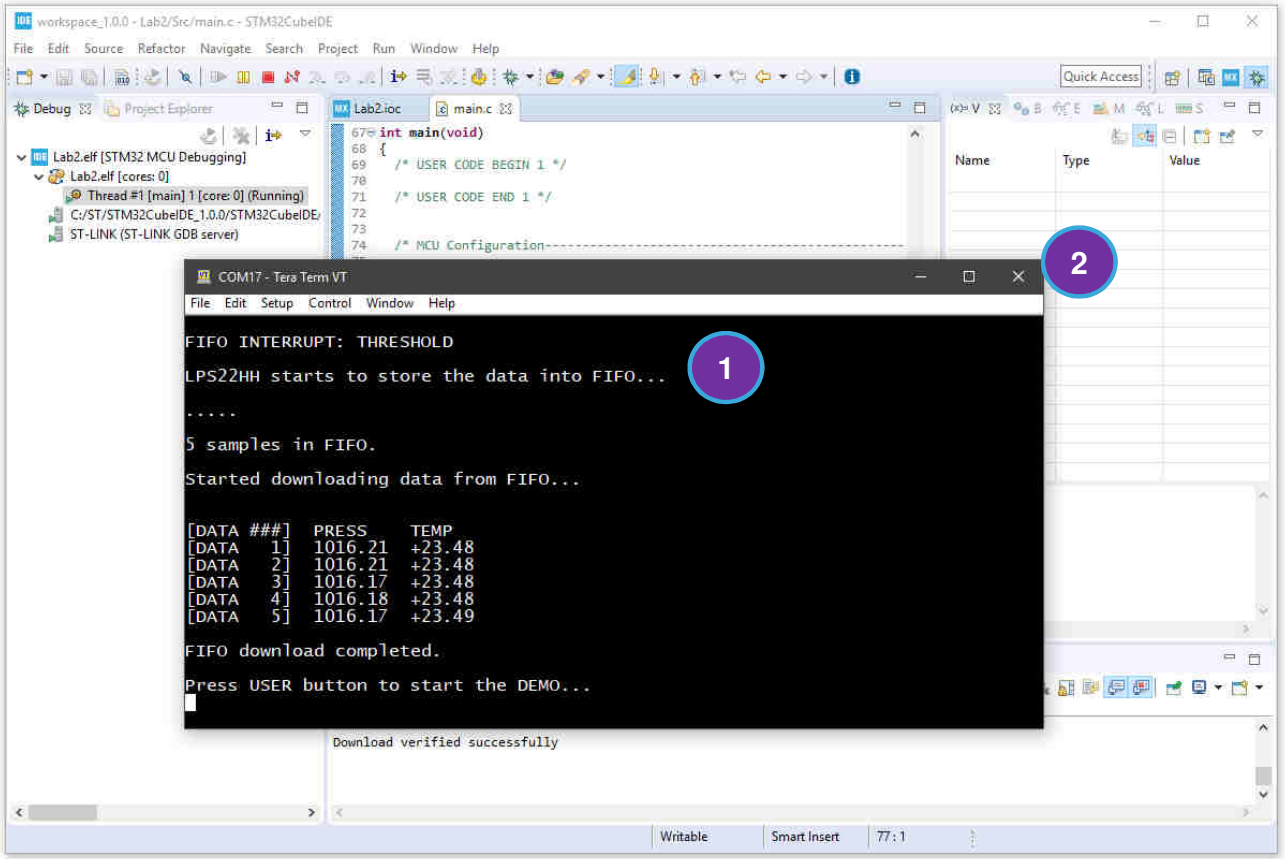
life.augmented

Lab2 - Testing

1. Open Tera Term to view the output

Press User button (blue) on Nucleo board

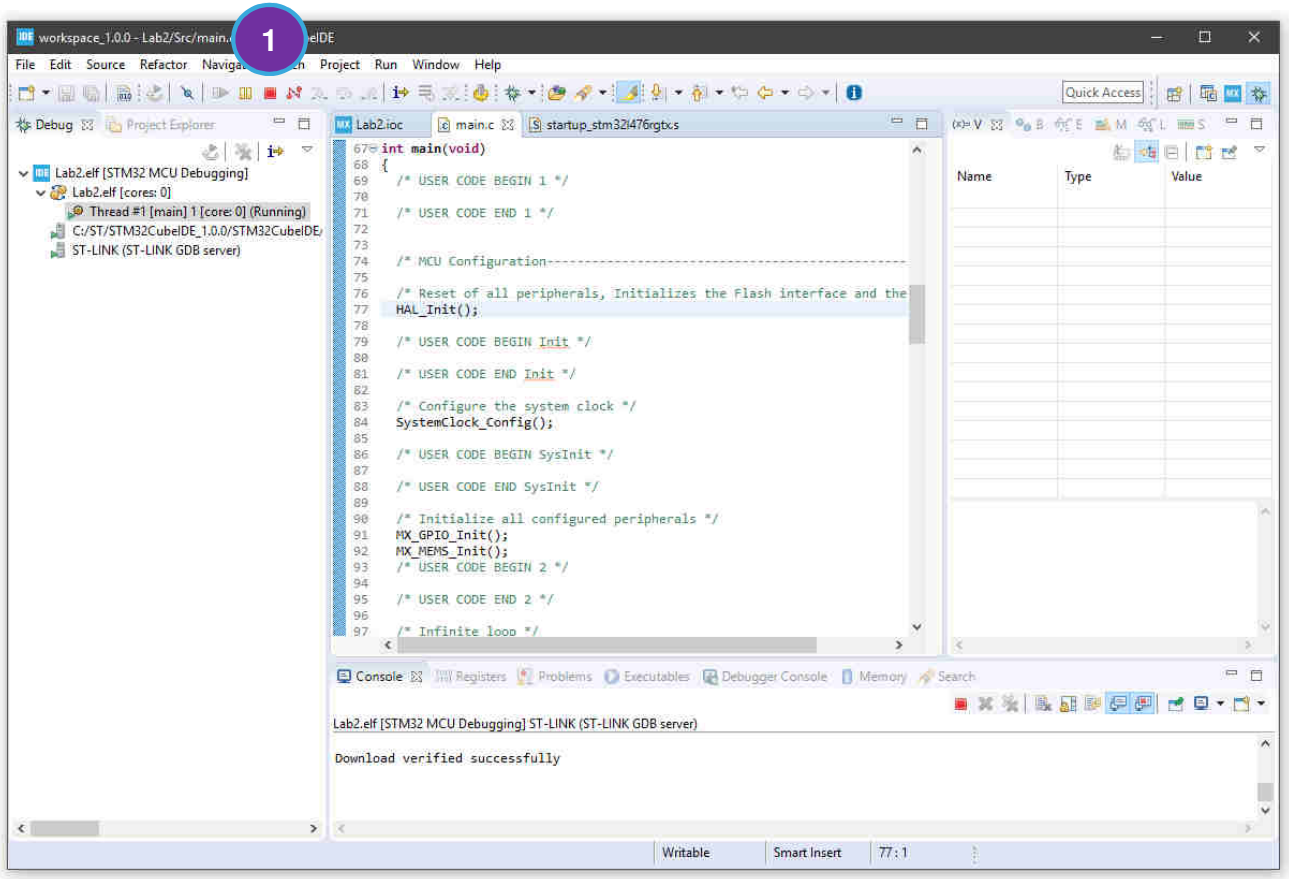
2. After testing close Tera Term by clicking **X**



life.augmented

Lab2 - Debugging

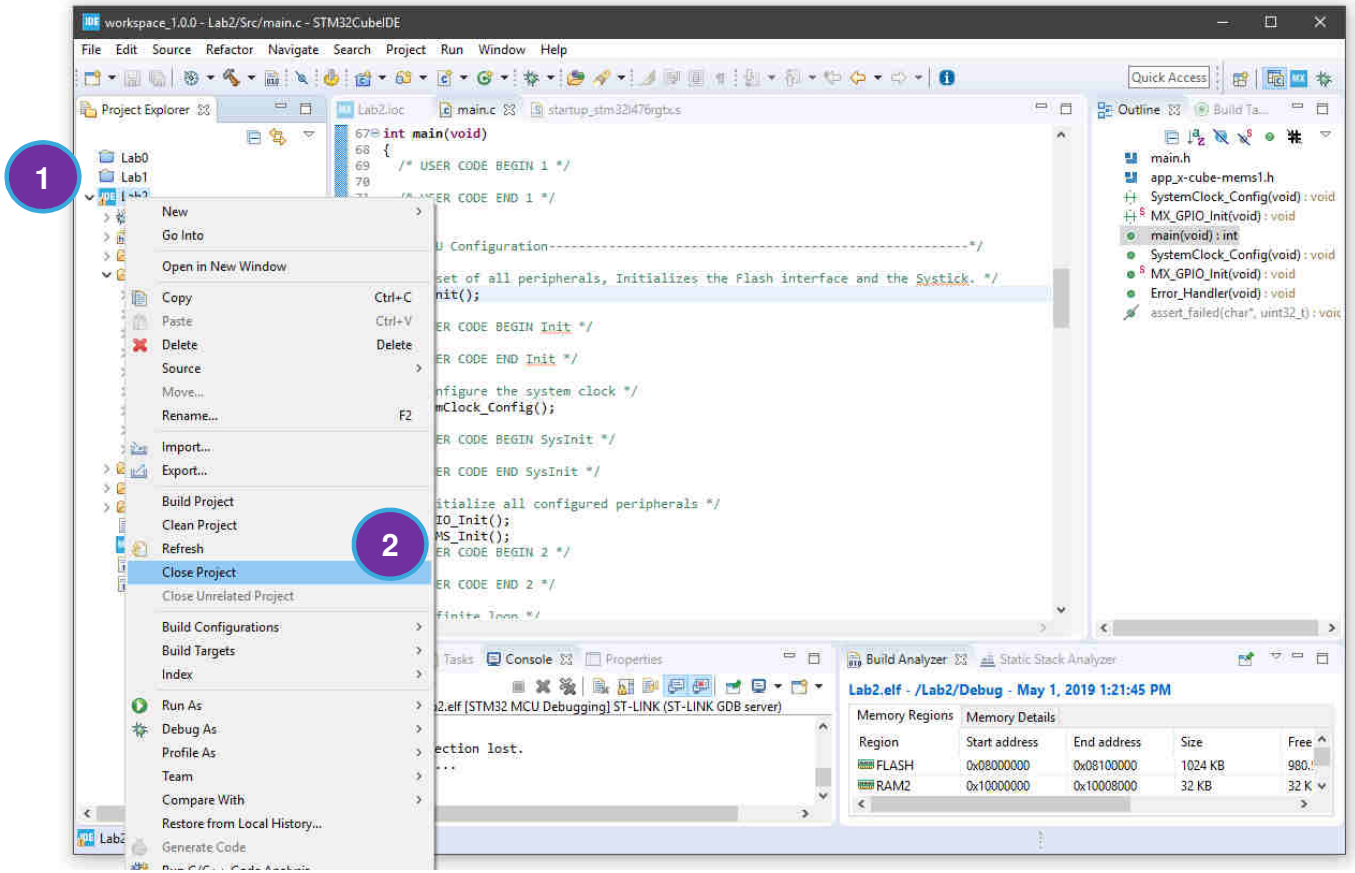
1. Click stop  button to interrupt the debugging



life.augmented

Lab2 – Closing the Project

1. Right-Click on **Lab2** project
2. Click on **Close Project**



life.augmented

LAB3

Goals:

- Configure a new project using X-CUBE-MEMS1
- Configure LSM6DSO accelerometer embedded step counter
- Enable interrupts in STM32CubeIDE

life.augmented

LSM6DSO

Accelerometer + Gyroscope Inertial Measurement Unit

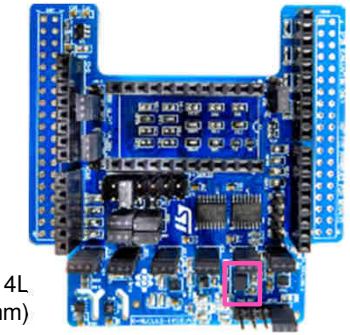
- 0.45mA power consumption (normal mode)
  - Accelerometer: 0.05mA, Gyroscope: 0.40mA
- Auxiliary SPI typically used for OIS / EIS or closed loop control; I3C Interface
- 9kB equivalent FIFO Memory for local data storage
- Finite State Machines (up to 16)
- Digital Features
  - Free fall
  - Pedometer 2.0
  - 6D / 4D
  - Tilt detection
  - Tap/ Double Tap

Lowest power consumption IMU → battery saving

Design Flexibility and cost optimization

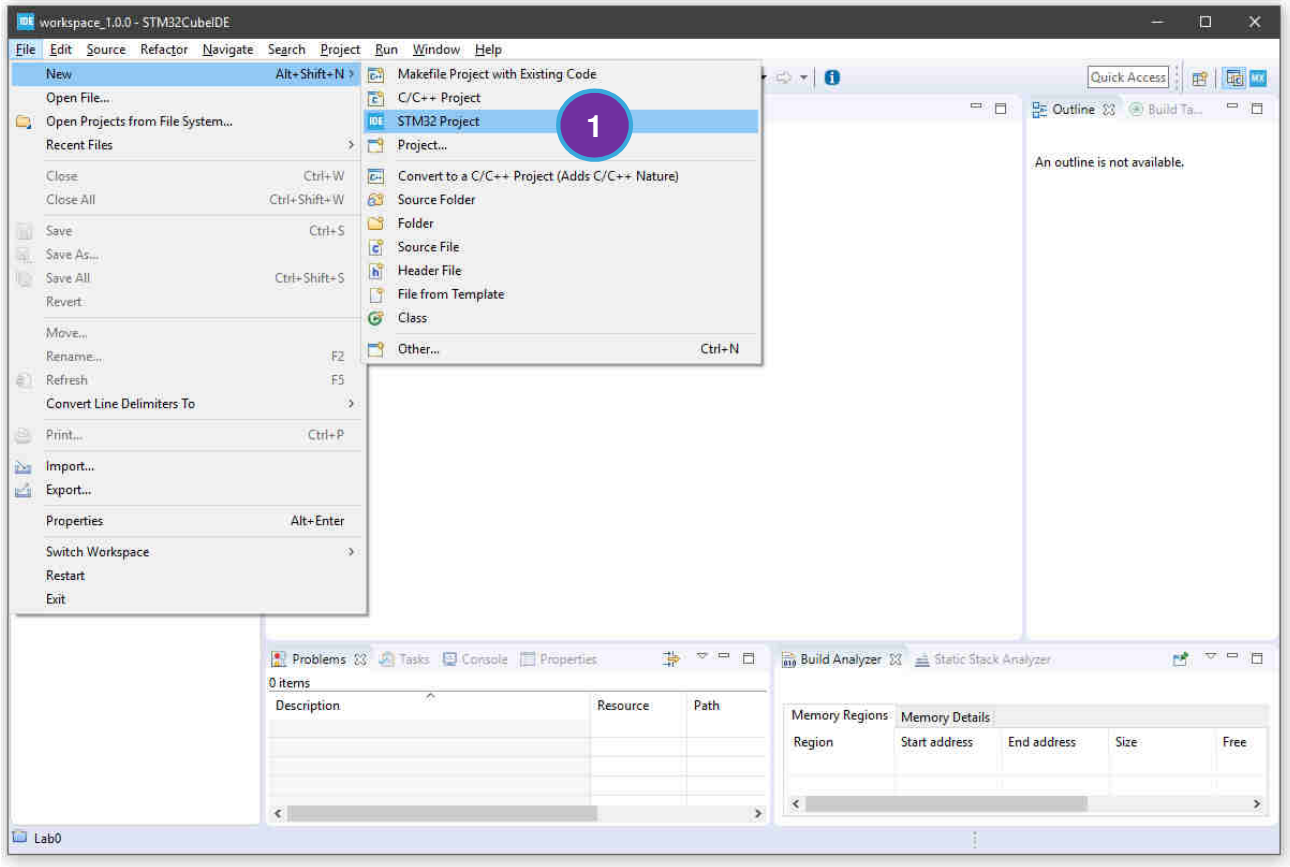
System power saving

Smart Functions with High Flexibility



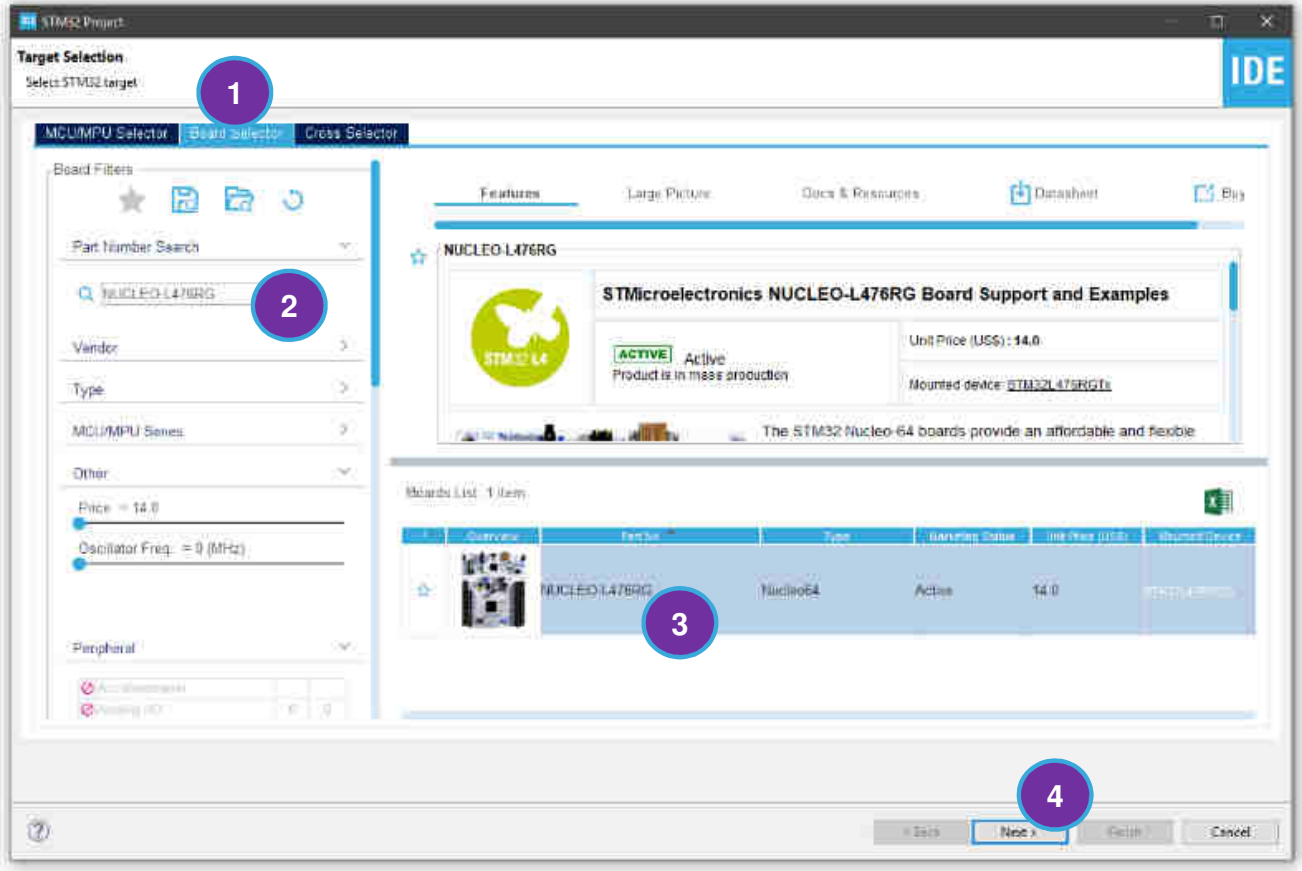
LGA-14L
(2.5 x 3 x 0.83 mm)

1. Click on **File > New > STM32 Project**



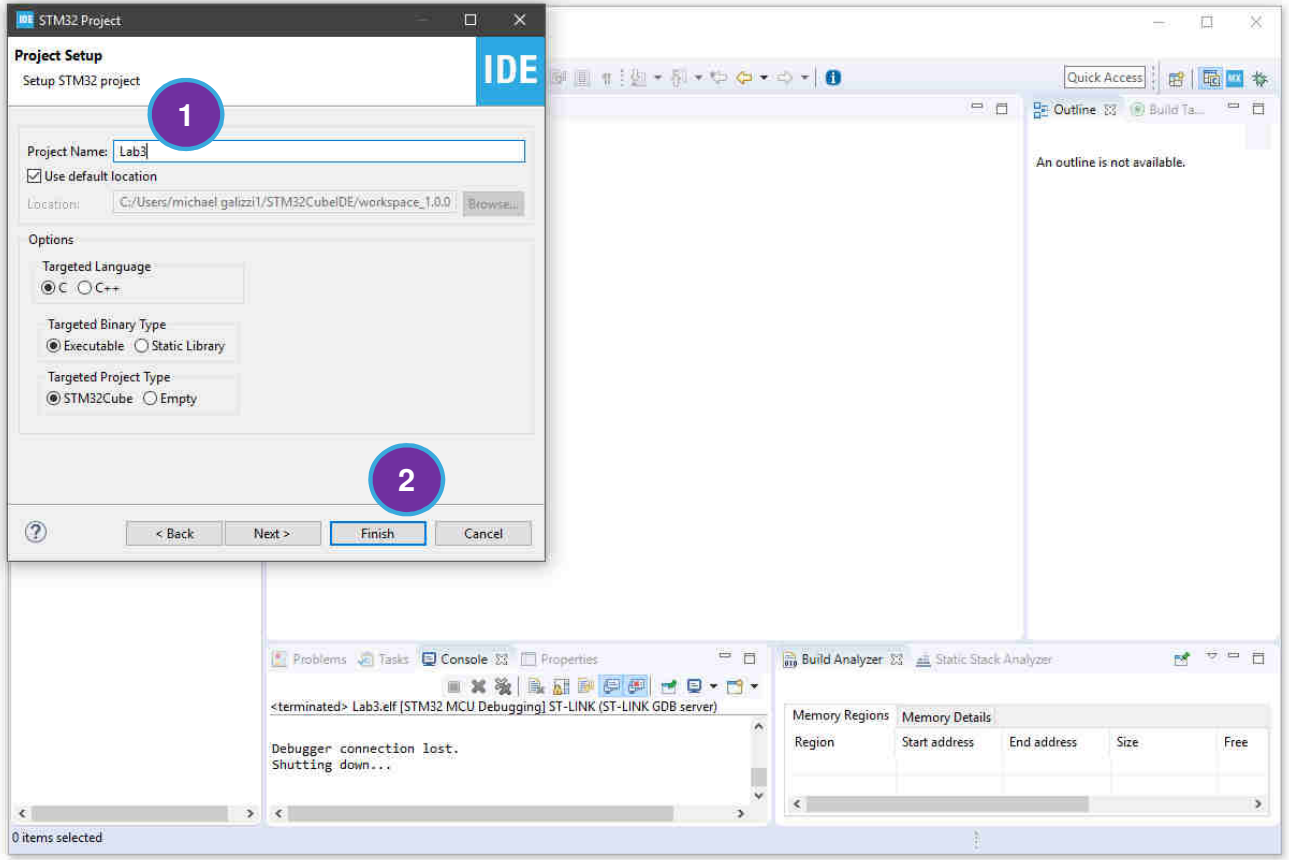
life.augmented

1. Click on **Board Selector**
2. Type **NUCLEO-L476RG**
3. Click on the board
4. Click **Next >**



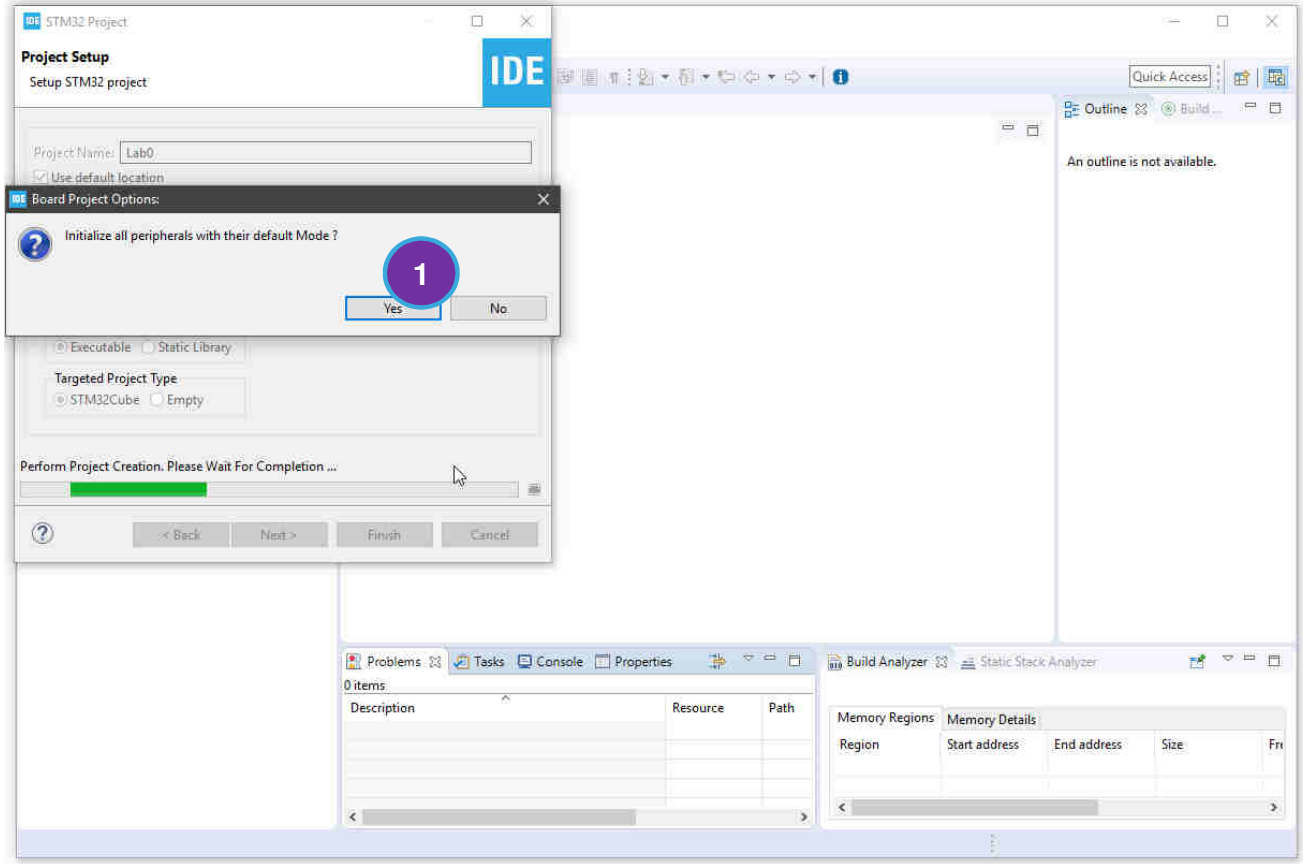
Lab3

1. Project Name **Lab3**
2. Click **Finish**



life.augmented

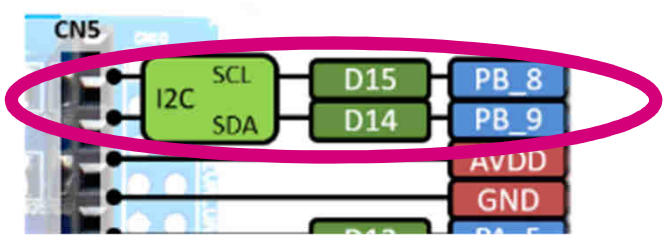
1. Click **Yes** to init peripherals in default mode

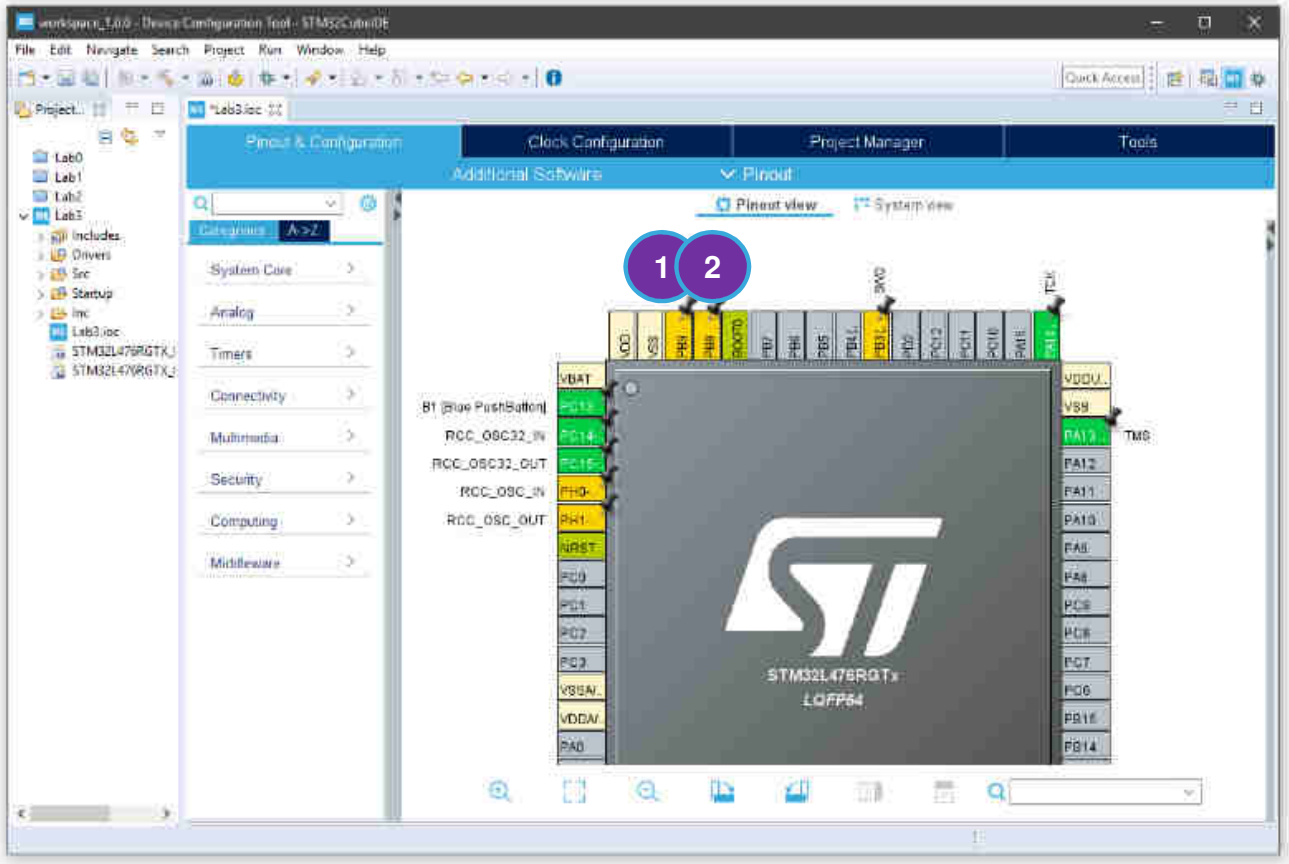


life.augmented

Lab3 – Configure the I2C Bus

1. Left Click on **PB9** and select I2C1_SDA
2. Left Click on **PB8** and select I2C1_SCL

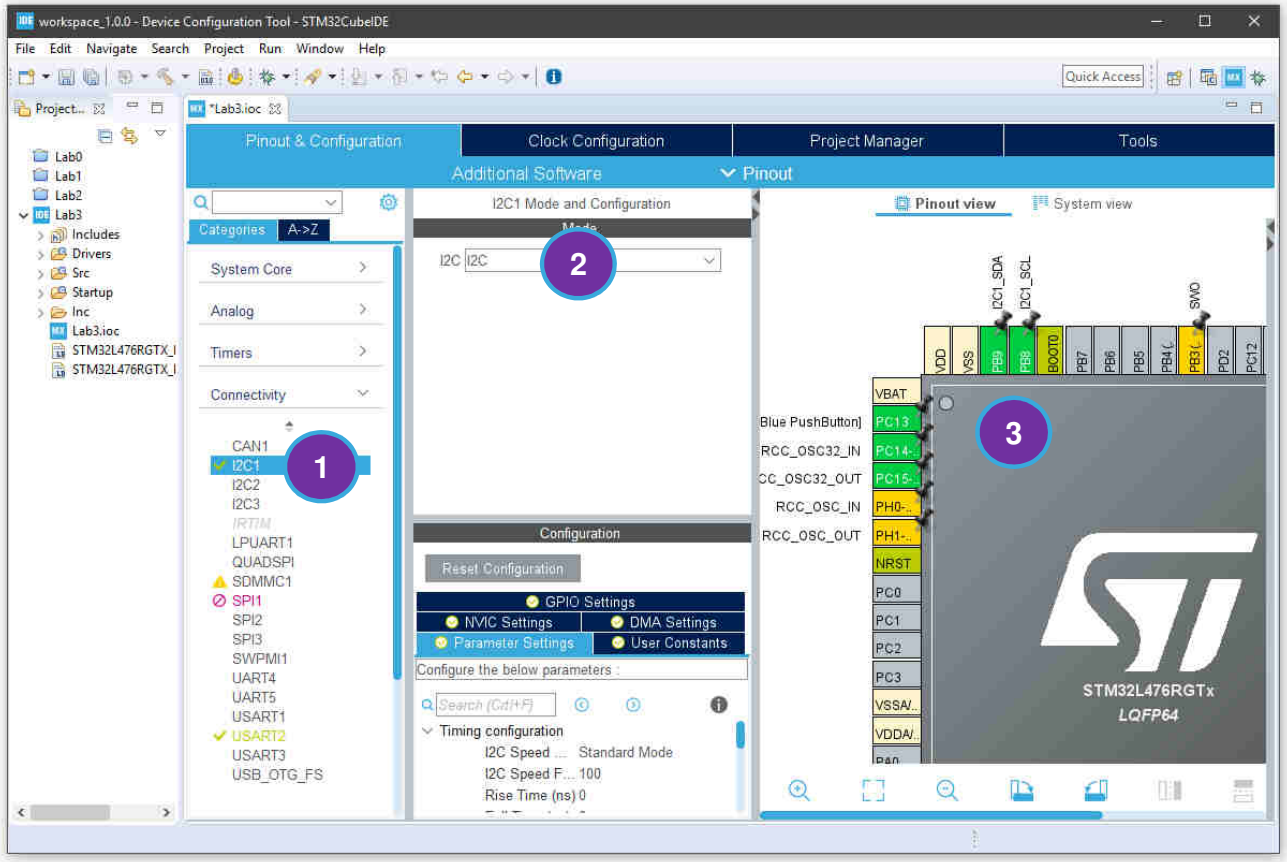




life.augmented

Lab3 – Configure the I2C Bus

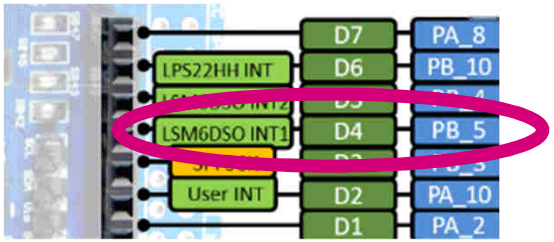
1. Expand *Connectivity* tab and check **I2C1**
2. Select **I2C** in *I2C1 Mode and Configuration*
3. PB8 and PB9 should now become green

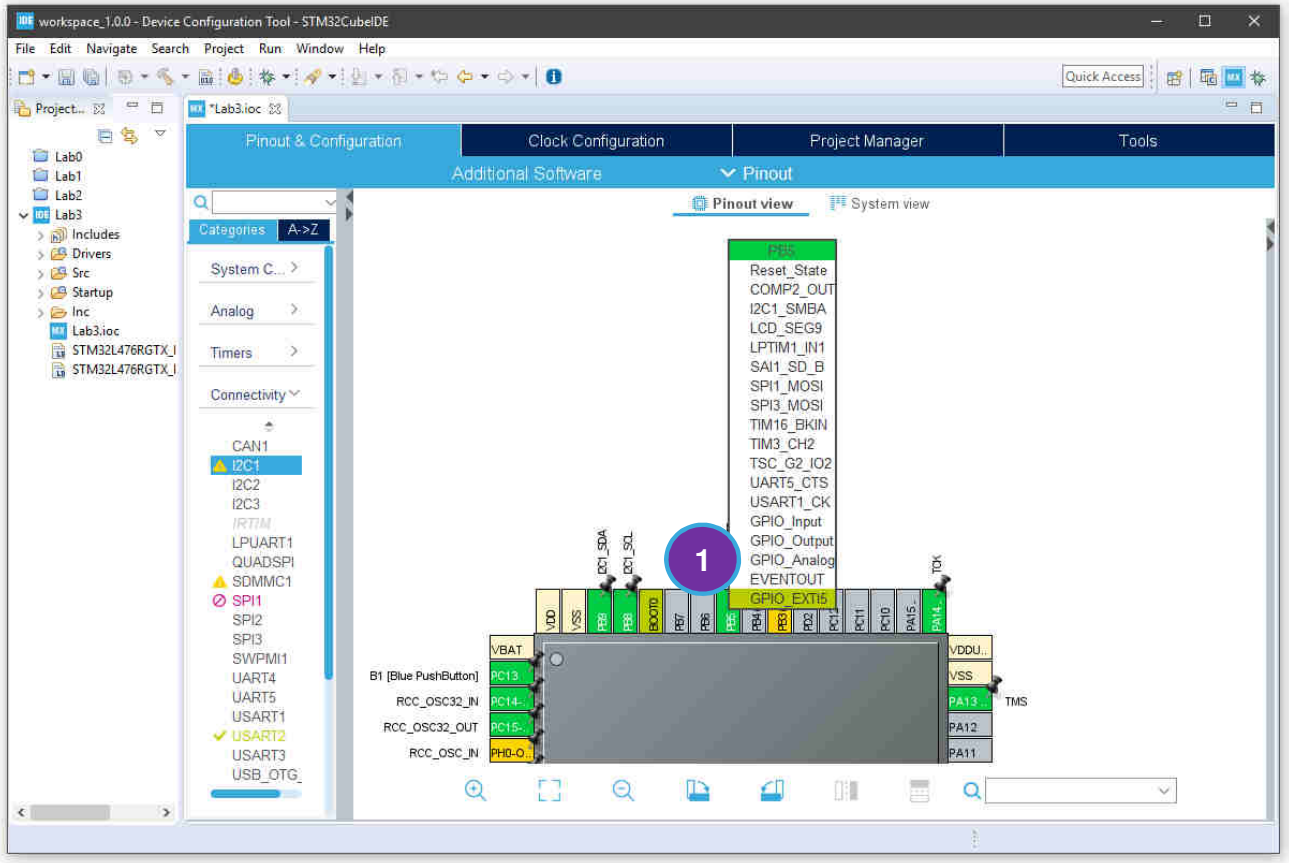


life.augmented

Lab3 – Configure LSM6DSO Interrupt

1. Left Click on **PB5** and select **GPIO_EXTI5**

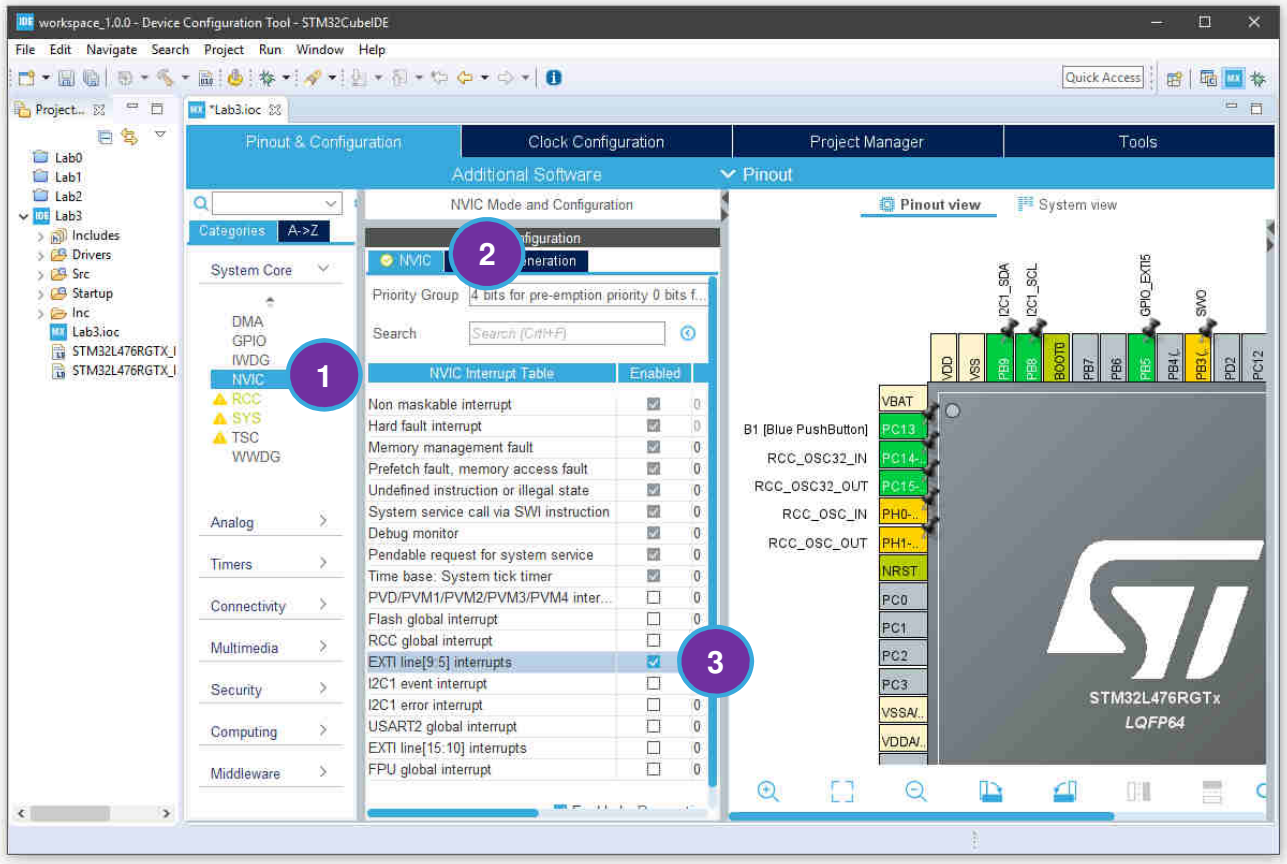




life.augmented

Lab3 – Configure LSM6DSO Interrupt

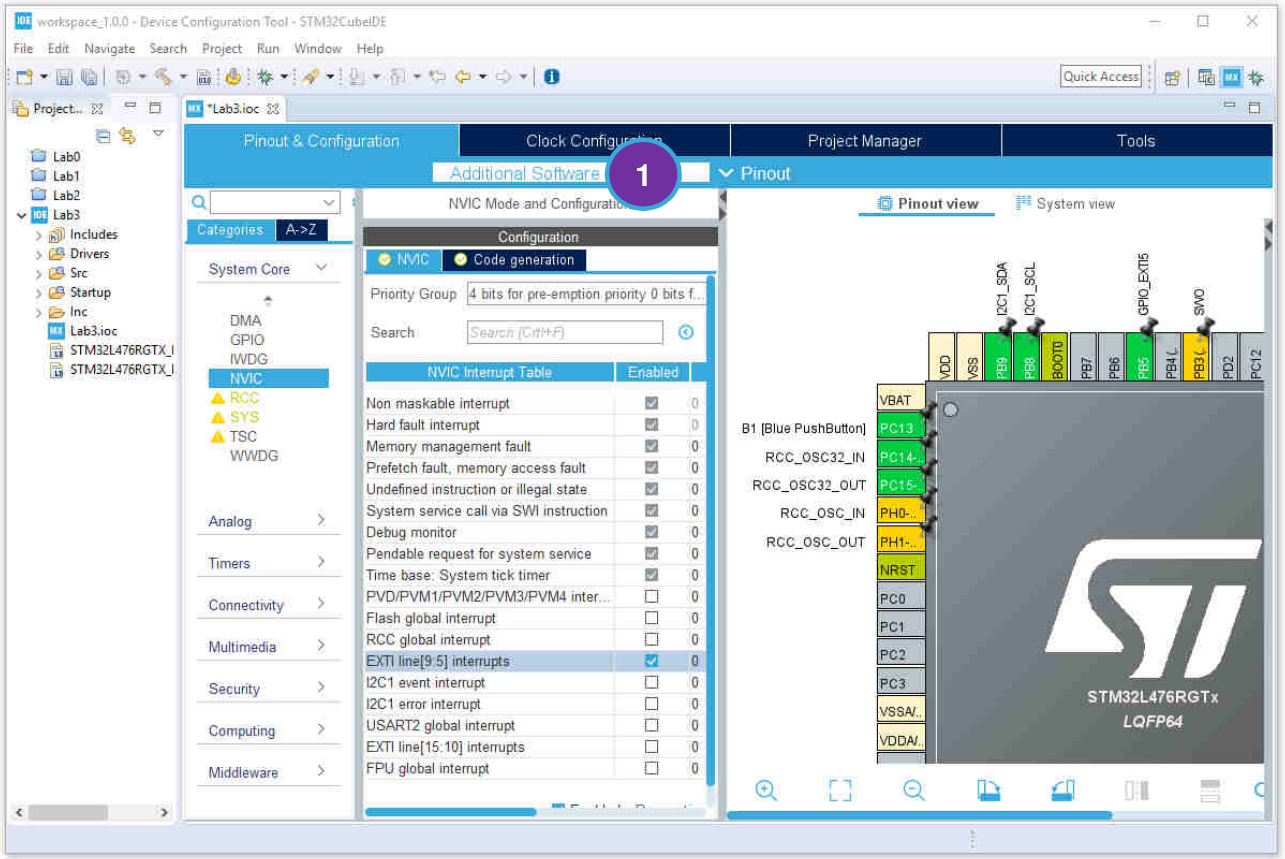
1. Check **NVIC** in tab System Core
2. Select **NVIC** in NVIC Mode and Configuration
3. Enable **EXTI line[9:5] interrupt**



life.augmented

Lab3 – Select the MEMS Library

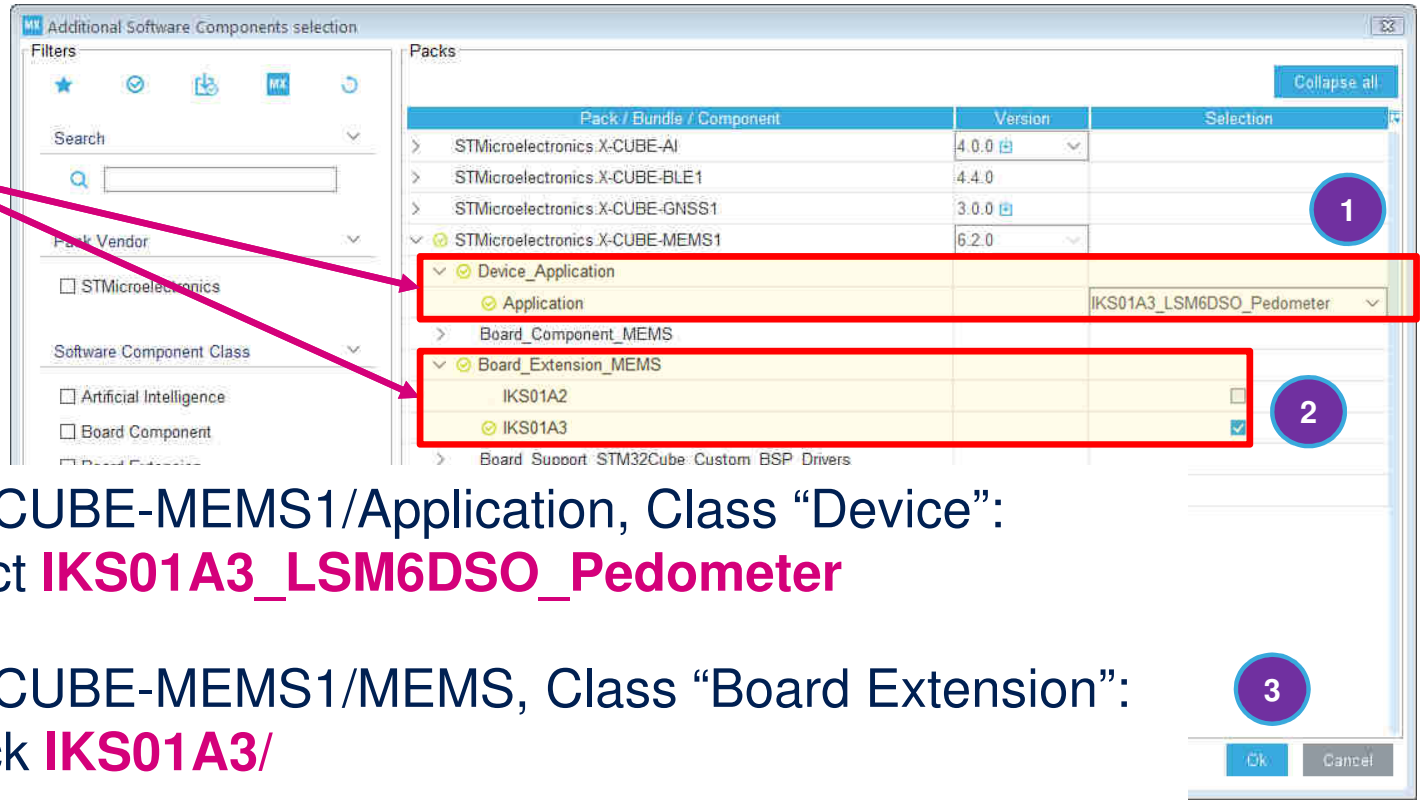
1. Click on **Additional Software**



life.augmented

Lab3 – Select the MEMS Library

Click to expand tree



1. In X-CUBE-MEMS1/Application, Class “Device”: Select **IKS01A3_LSM6DSO_Pedometer**

2. In X-CUBE-MEMS1/MEMS, Class “Board Extension”: Check **IKS01A3/**

3. Click **OK**

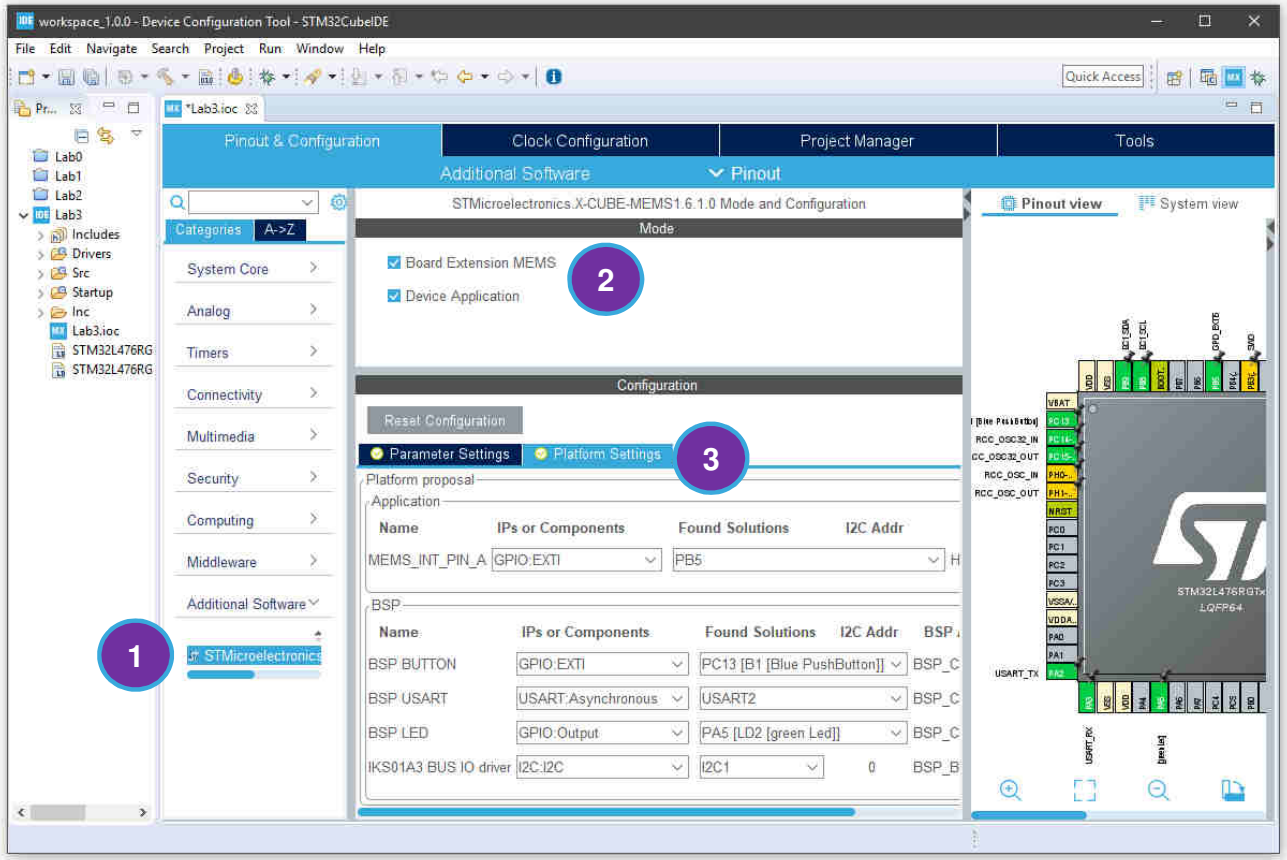
life.augmented

Lab3 – Configure the MEMS Library

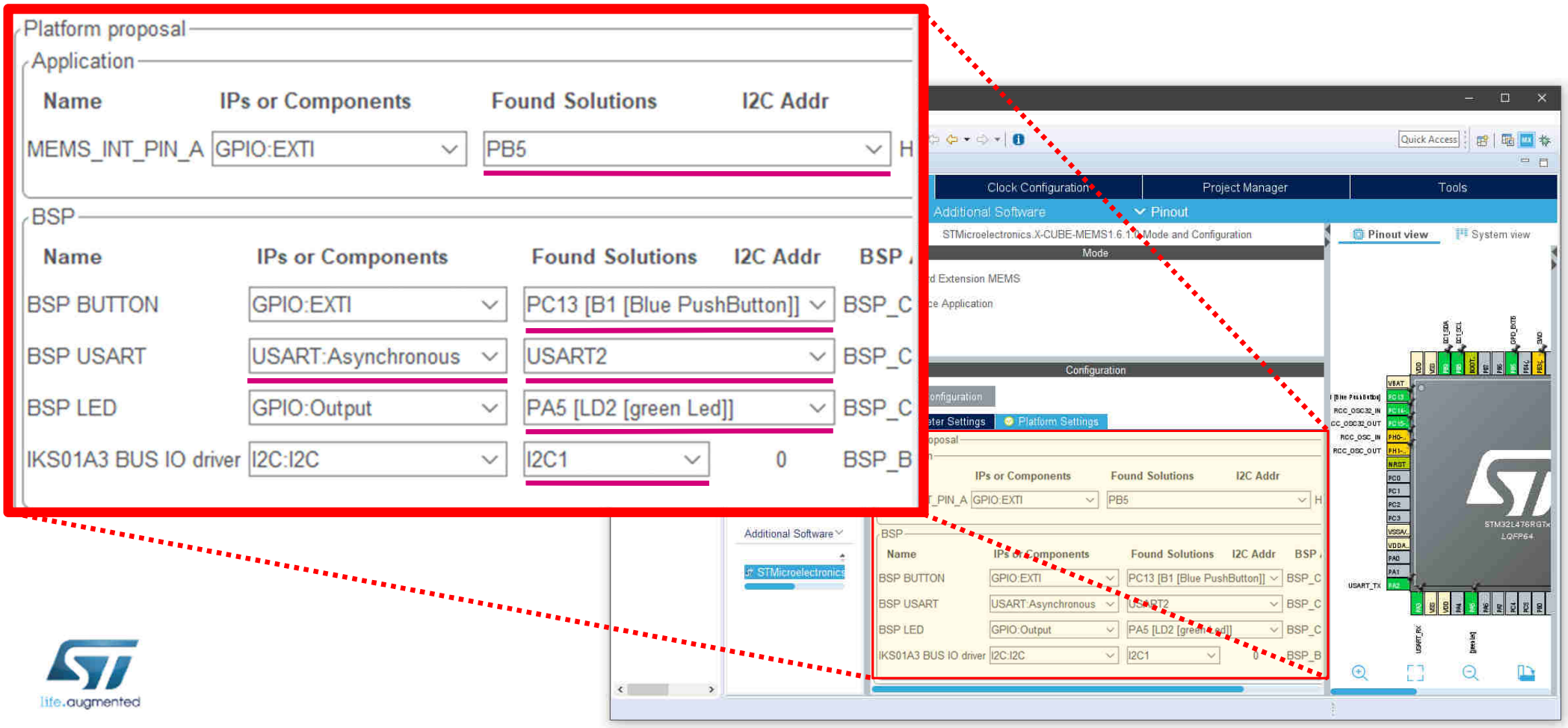
1. Expand Additional Software and select the X-CUBE-MEMS1

2. Check both:
   Board Extension MEMS
   Device Application

3. Configure Platform Settings as in picture



Lab3 – Configure the MEMS Library

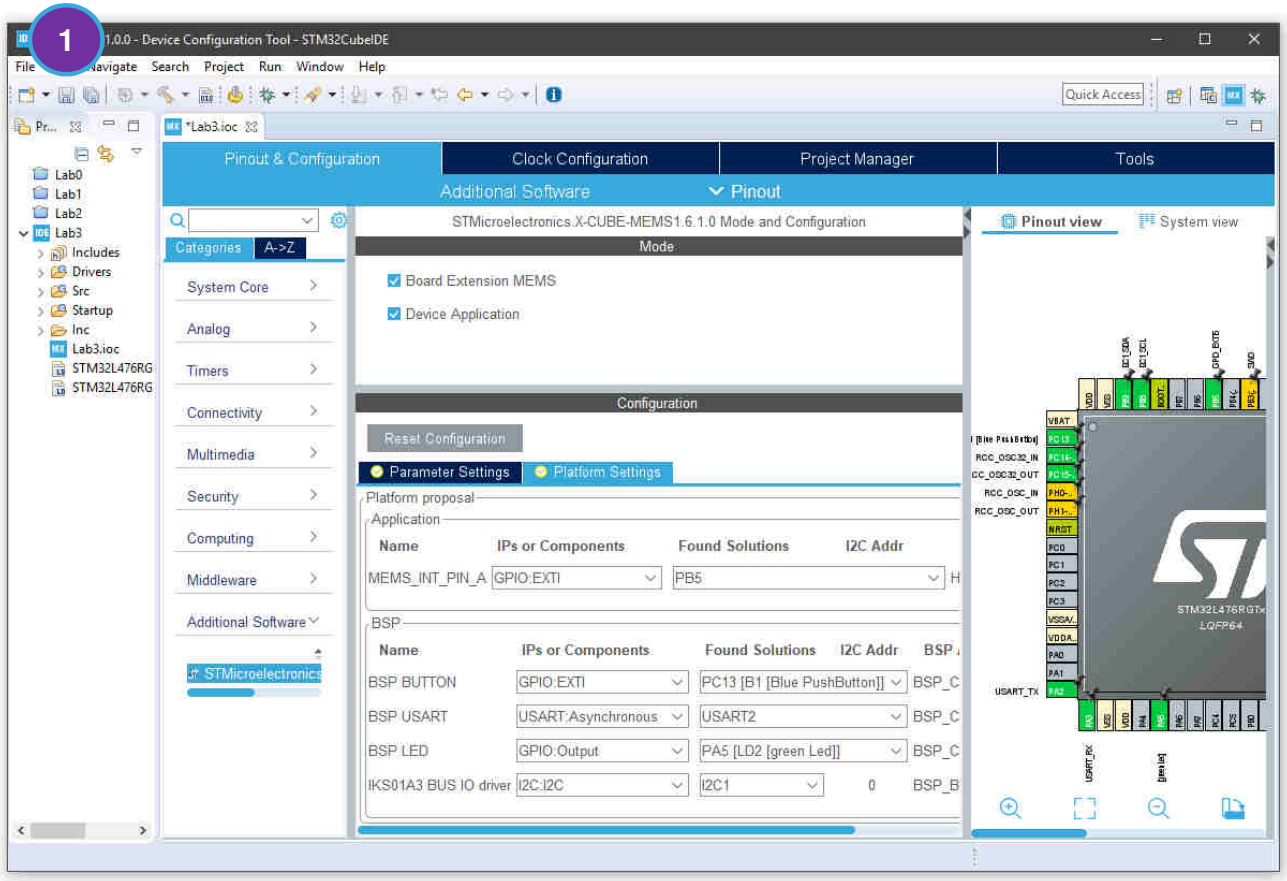


life.augmented

Lab3 – Save the Project

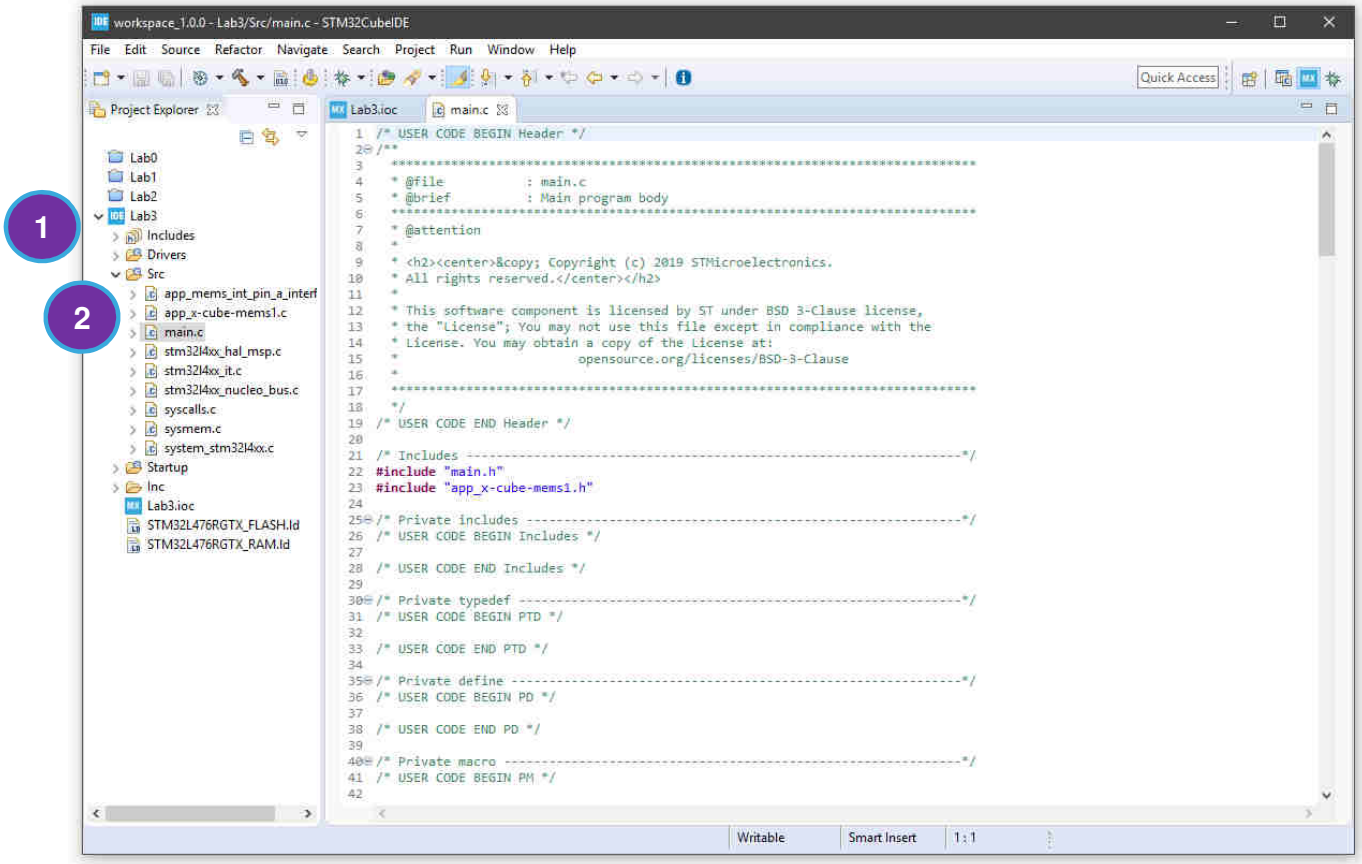
1. Click the save button 

   This action will generate the source code of this lab



Lab3 – Code Editing

1. Expand **Src** in folder **Lab3**
2. Double click on **main.c**

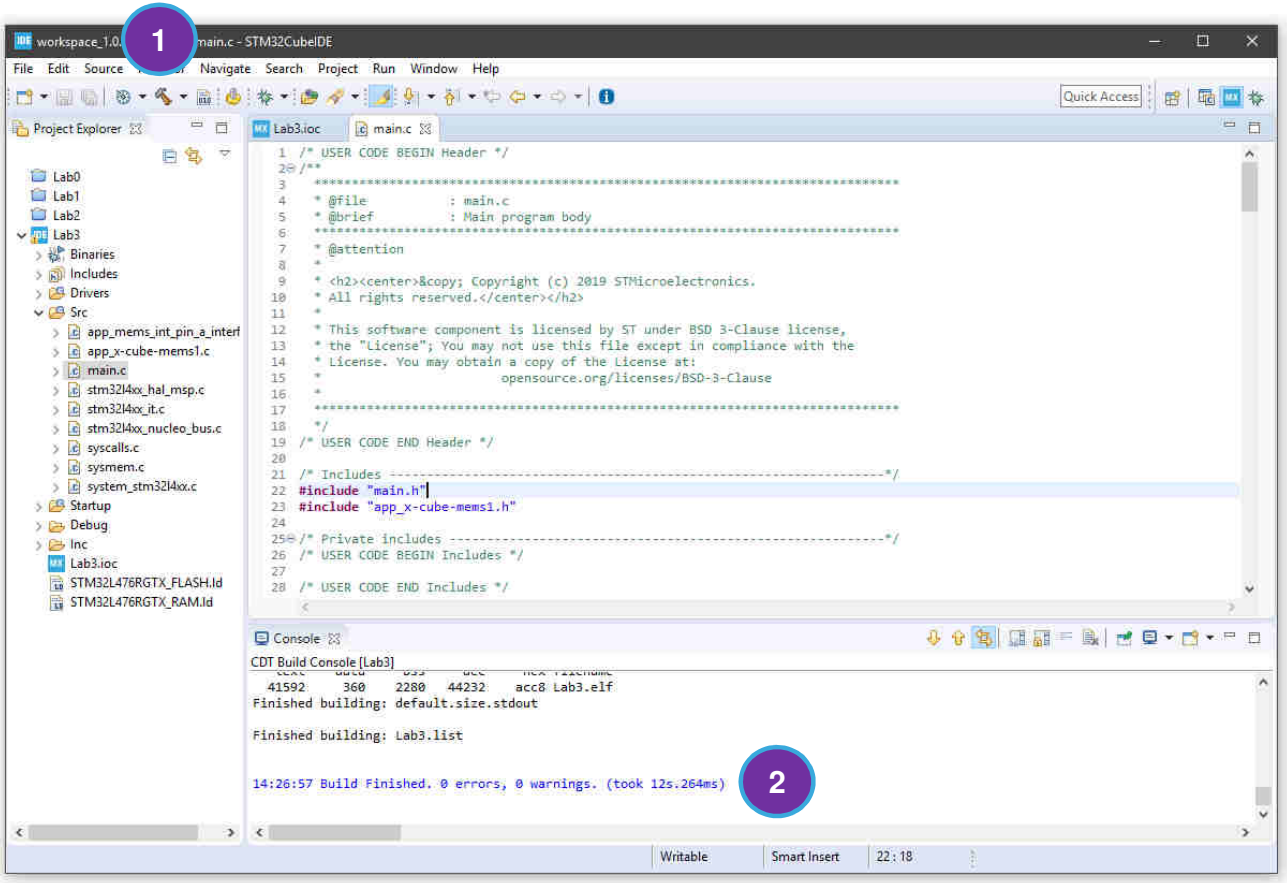


life.augmented

Lab3 - Compiling

1. Click on the hammer  to begin compilation, or press **CTRL+B**

2. Compilation should terminate with 0 errors and 0 warning



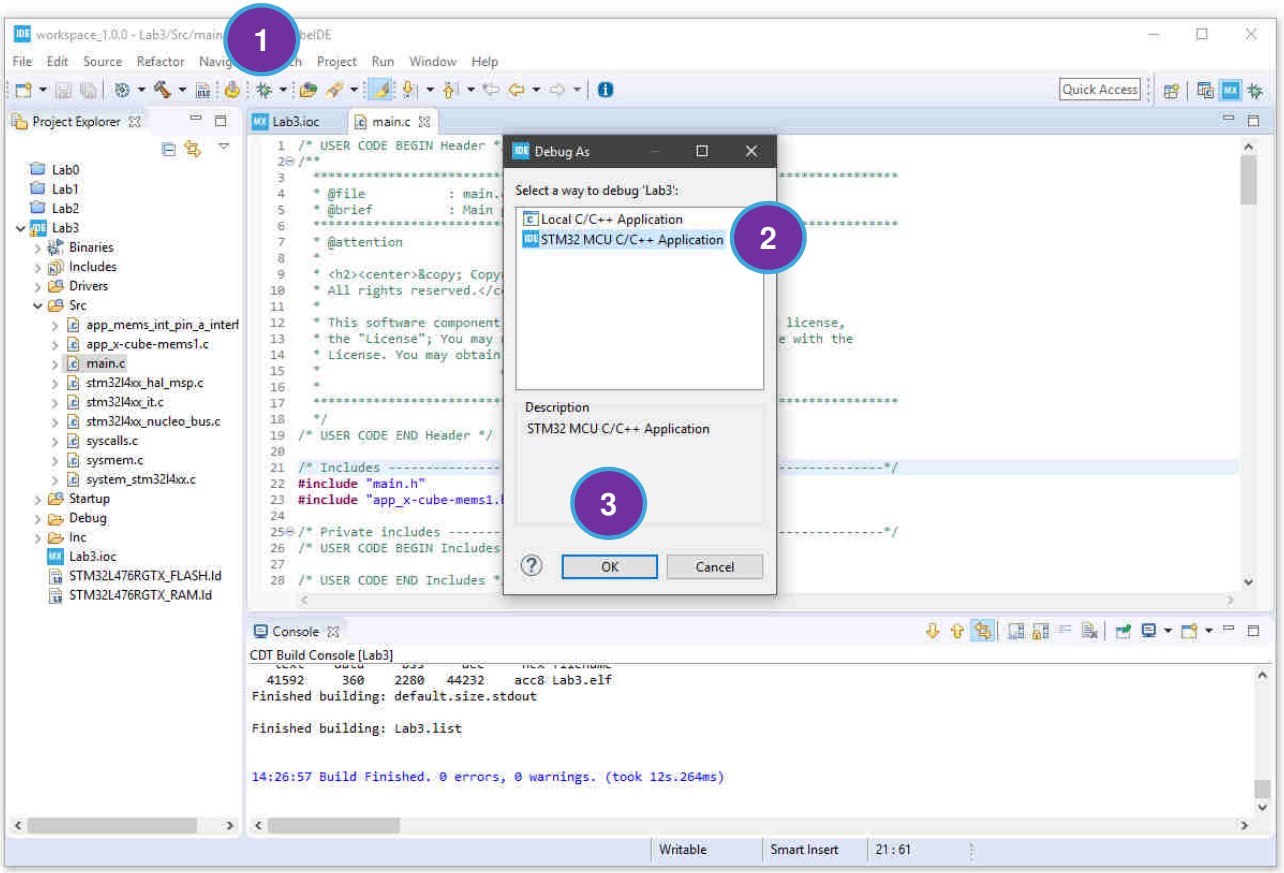
life.augmented

Lab3 - Debugging

1. Click on the bug  to begin debugging

2. Select
   STM32 MCU C/C++ App

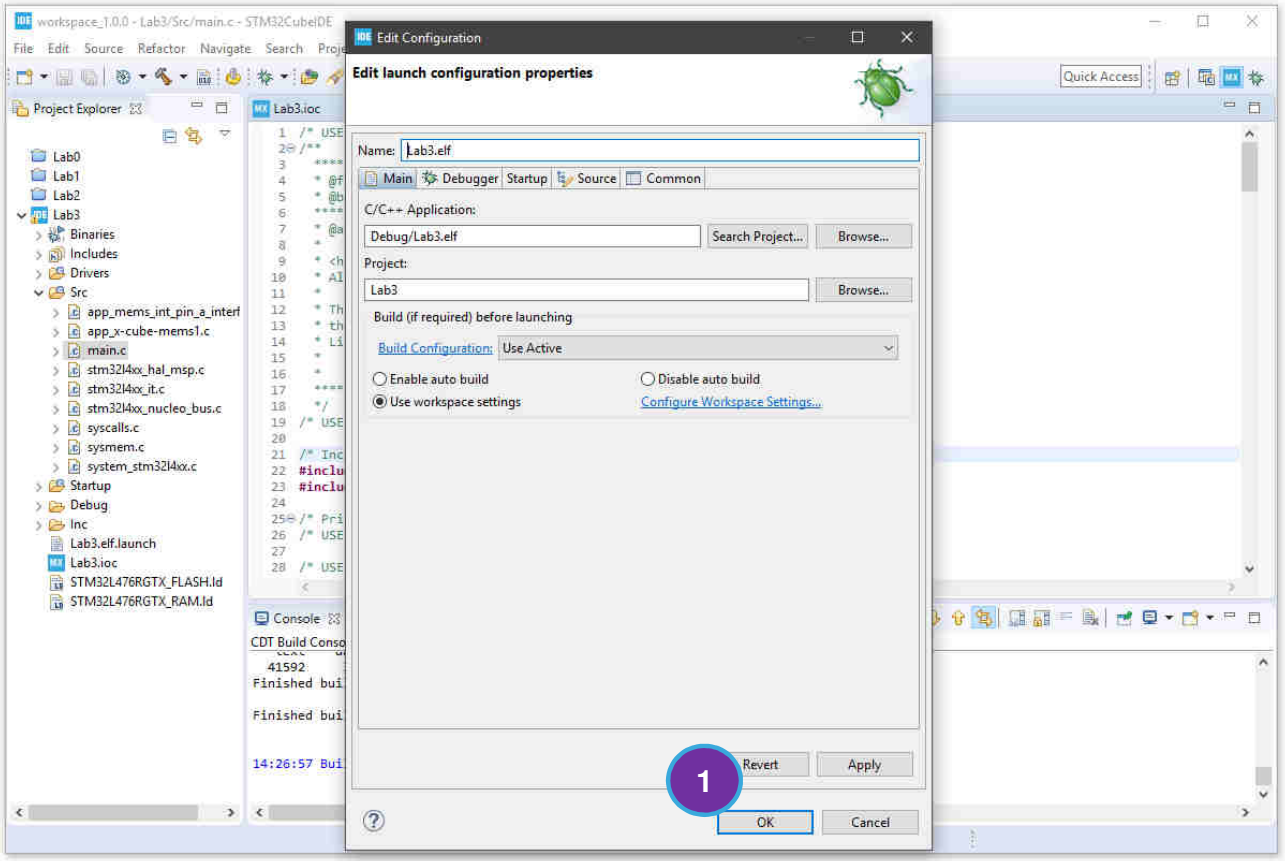
3. Click **OK**



life.augmented

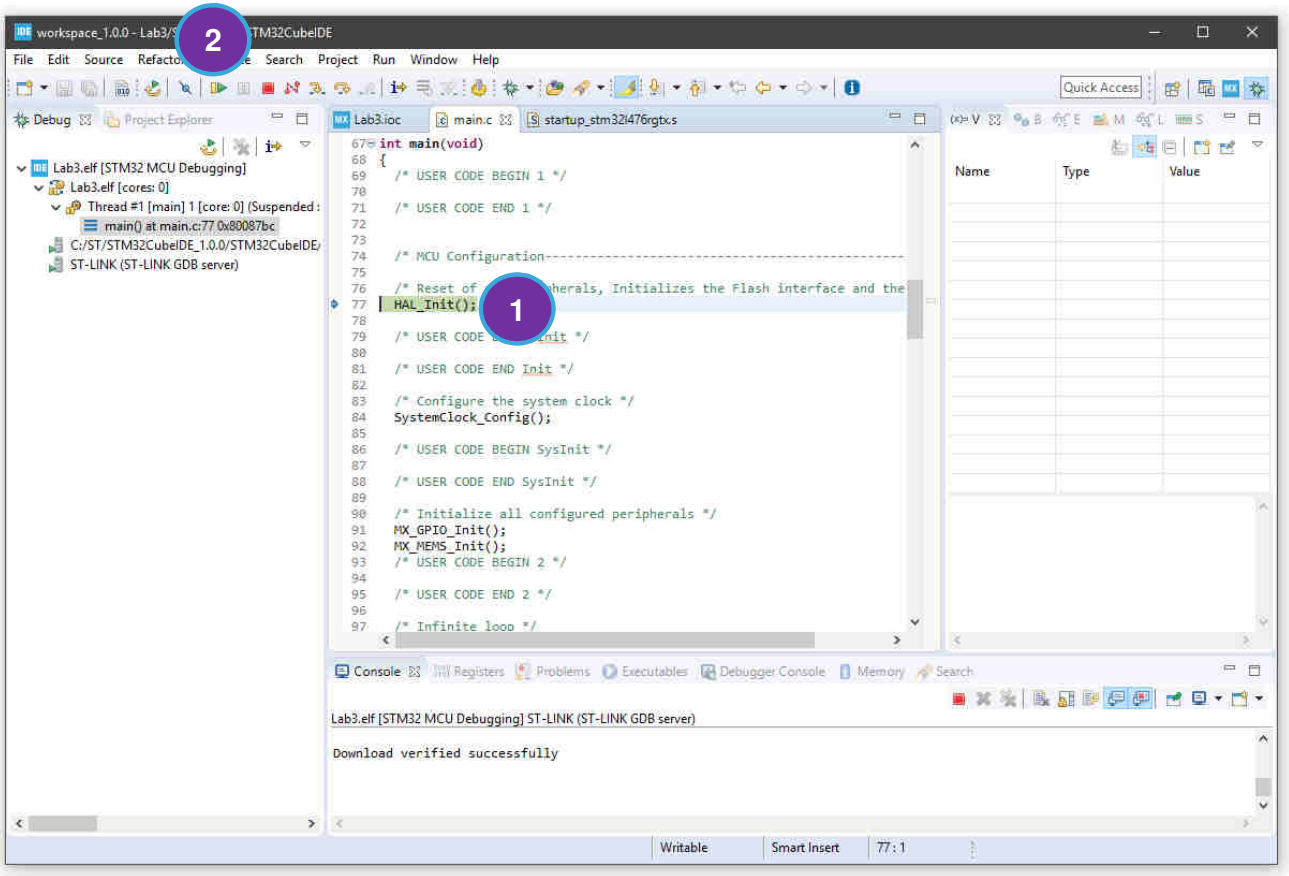
Lab3 - Debugging

1. Click **OK**



Lab3 - Debugging

1. Code start at the first line of the main function
2. Click play  button to run the code



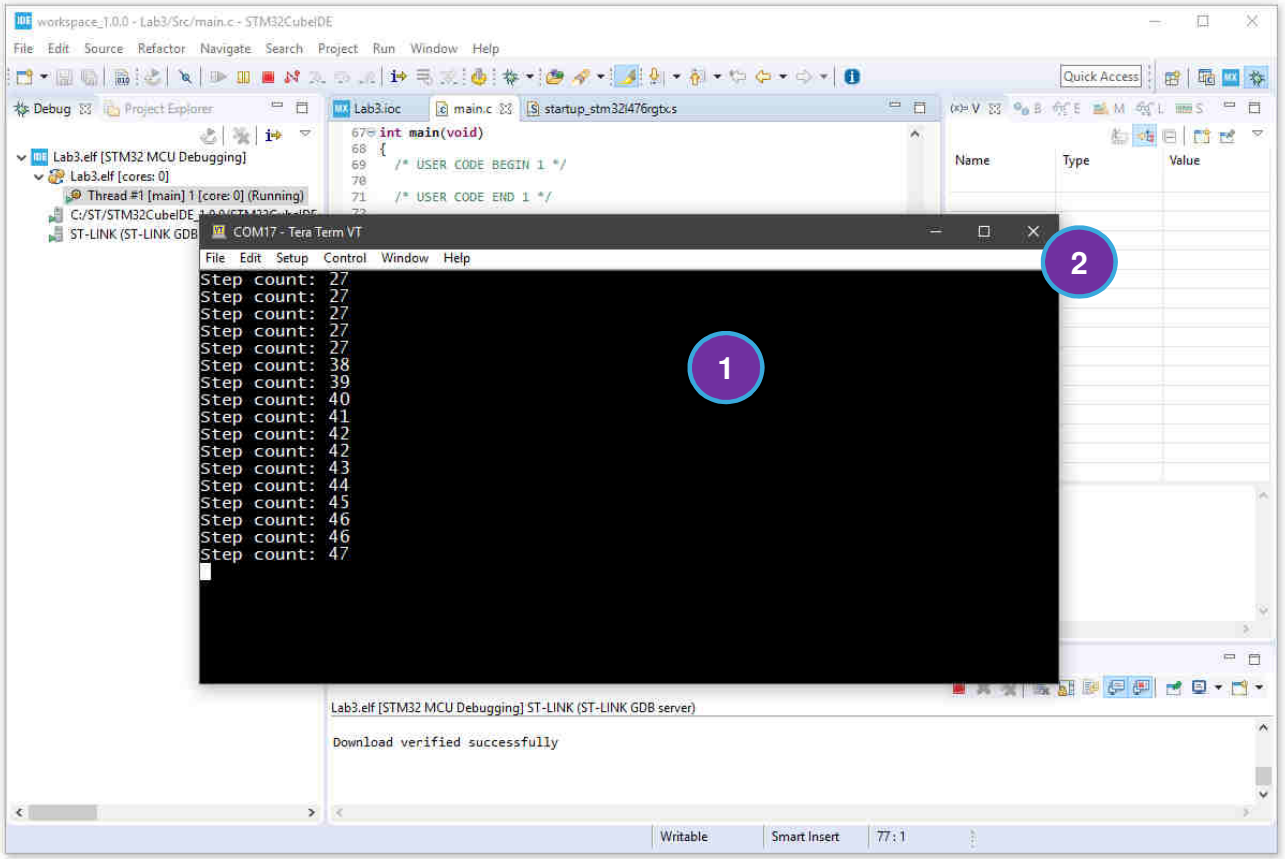
life.augmented

Lab3 - Testing

1. Open Tera Term to view the output

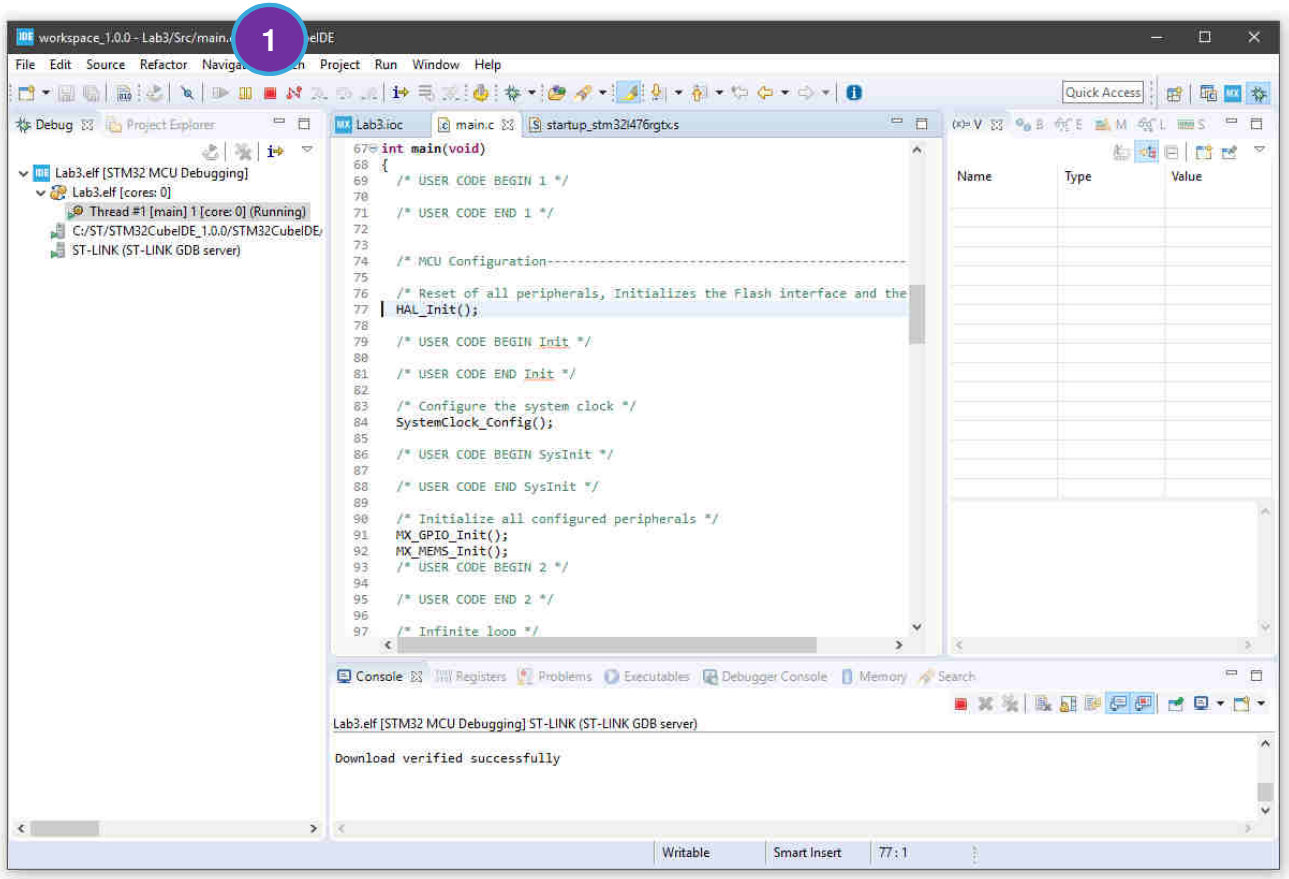
Simulate a walk by giving up/down movement to the board. Steps will be updated after about 10 steps

2. After testing close Tera Term by clicking **X**



Lab3 - Debugging

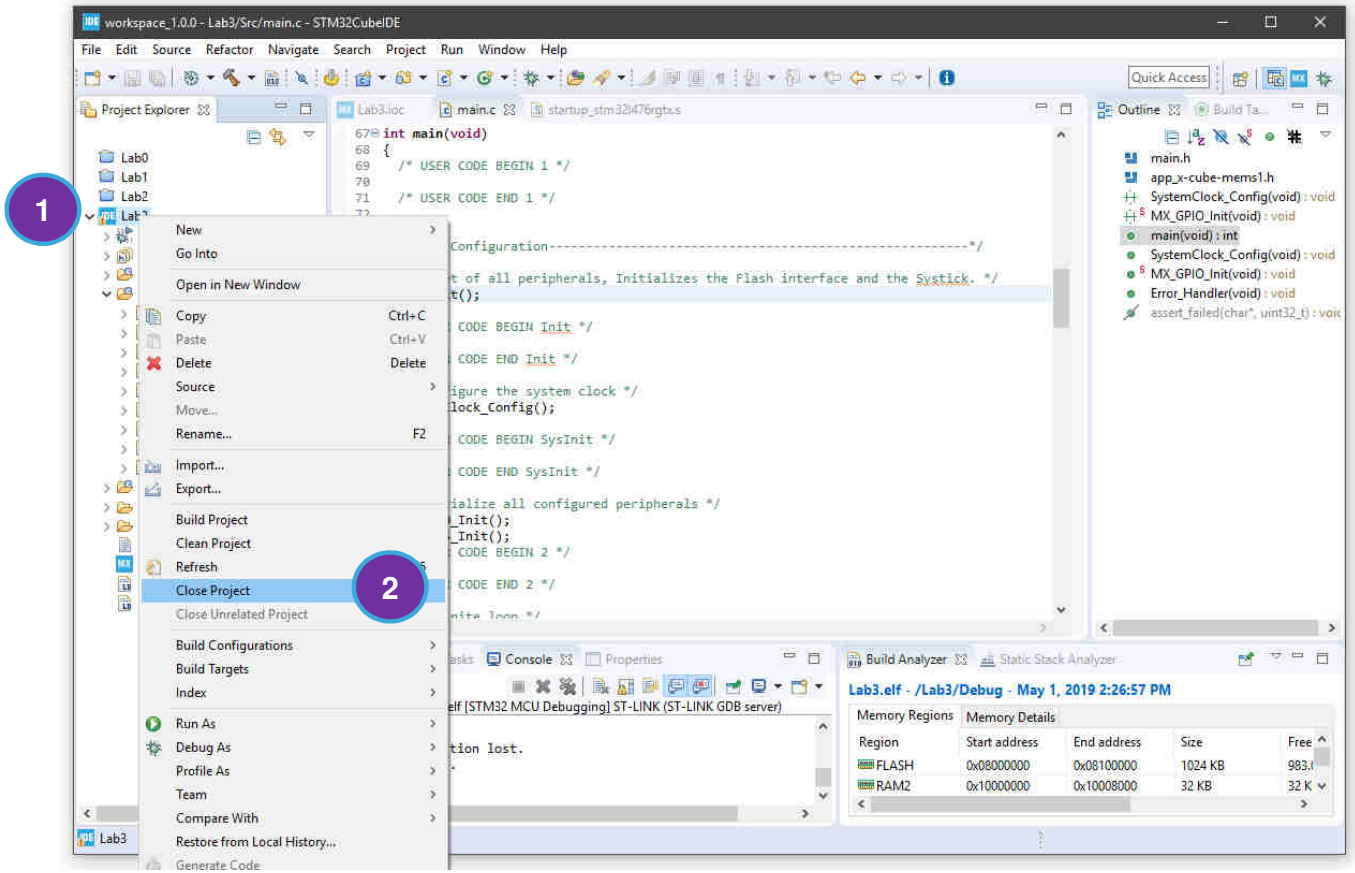
1. Click stop  button to interrupt the debugging



life.augmented

Lab3 – Closing the Project

1. Right-Click on **Lab3** project
2. Click on **Close Project**



life.augmented

LAB4

Goals:

- Configure a new project using X-CUBE-MEMS1
- Configure LIS2DW12 accelerometer in order to generate an interrupt when an acceleration is detected
- Enable interrupts in STM32CubeIDE
- Turn the led ON at wake-up

life.augmented

LIS2DW12

Power saving and flexible Accelerometer

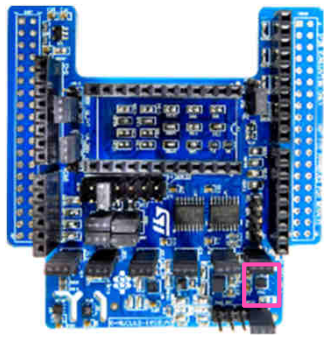
- Down to 0.38 µA power consumption (1.6Hz ODR)
- High Perf. mode: up to 1600Hz with noise density 90 µg/√Hz
- 5 Power Modes + 2 Noise Modes
- 32-level FIFO buffer
- Digital Features
  - Free fall
  - Wake-up
  - 6D / 4D
  - Stationary/Motion detection
  - Double Tap

Enabling battery saving

Accuracy

High Flexibility

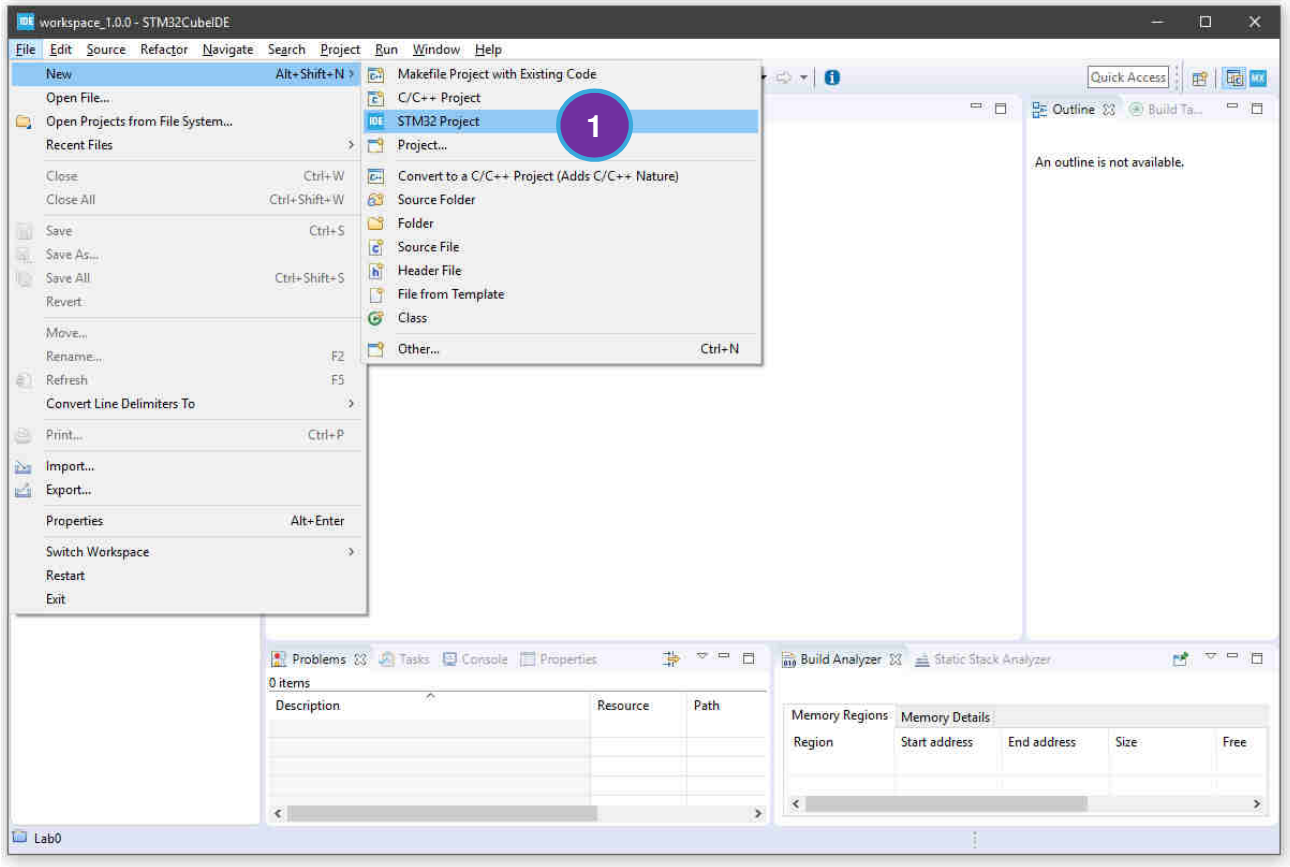
System Power Saving & Smart Functions



LGA-12L
(2.0 x 2.0 x 0.7 mm)

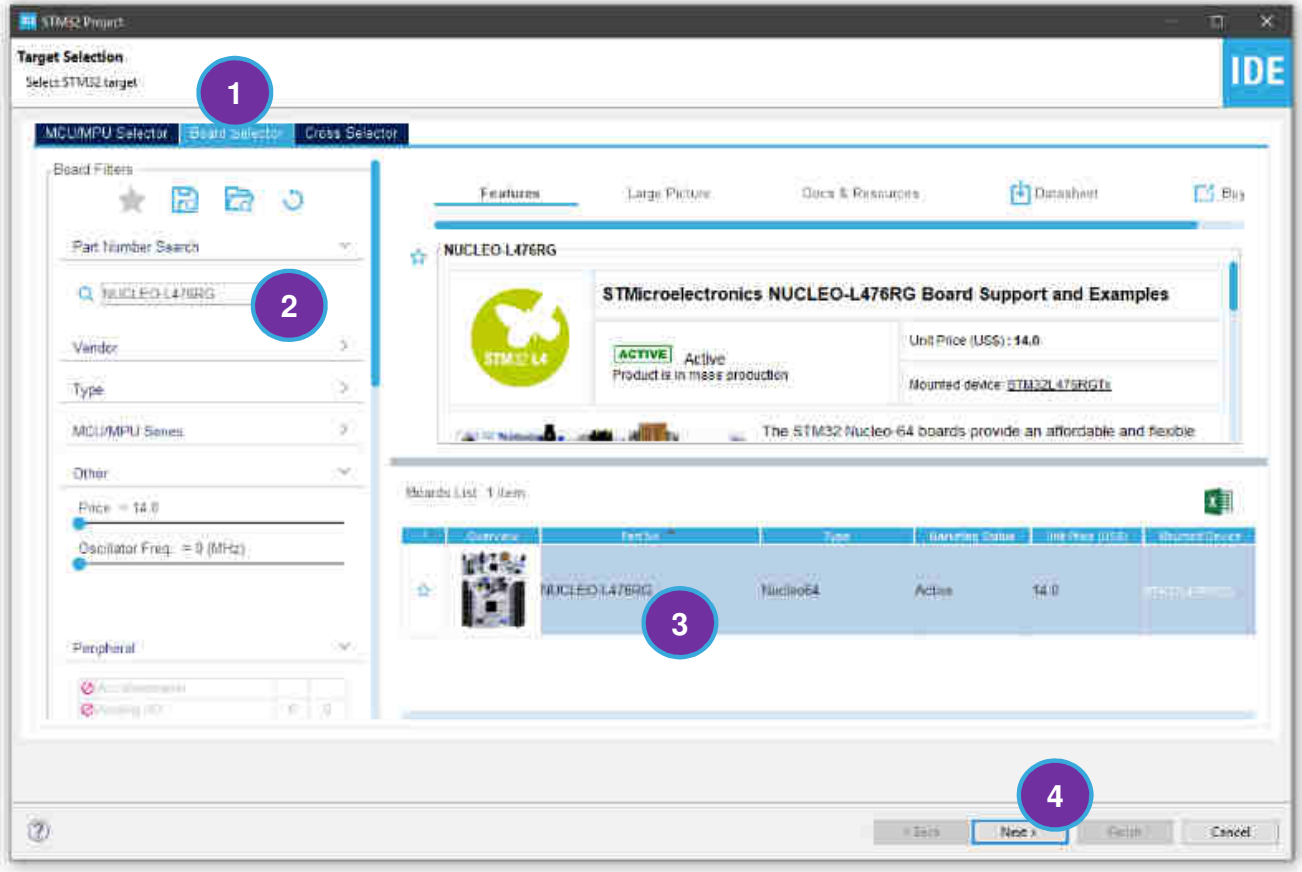
life.augmented

1. Click on **File > New > STM32 Project**

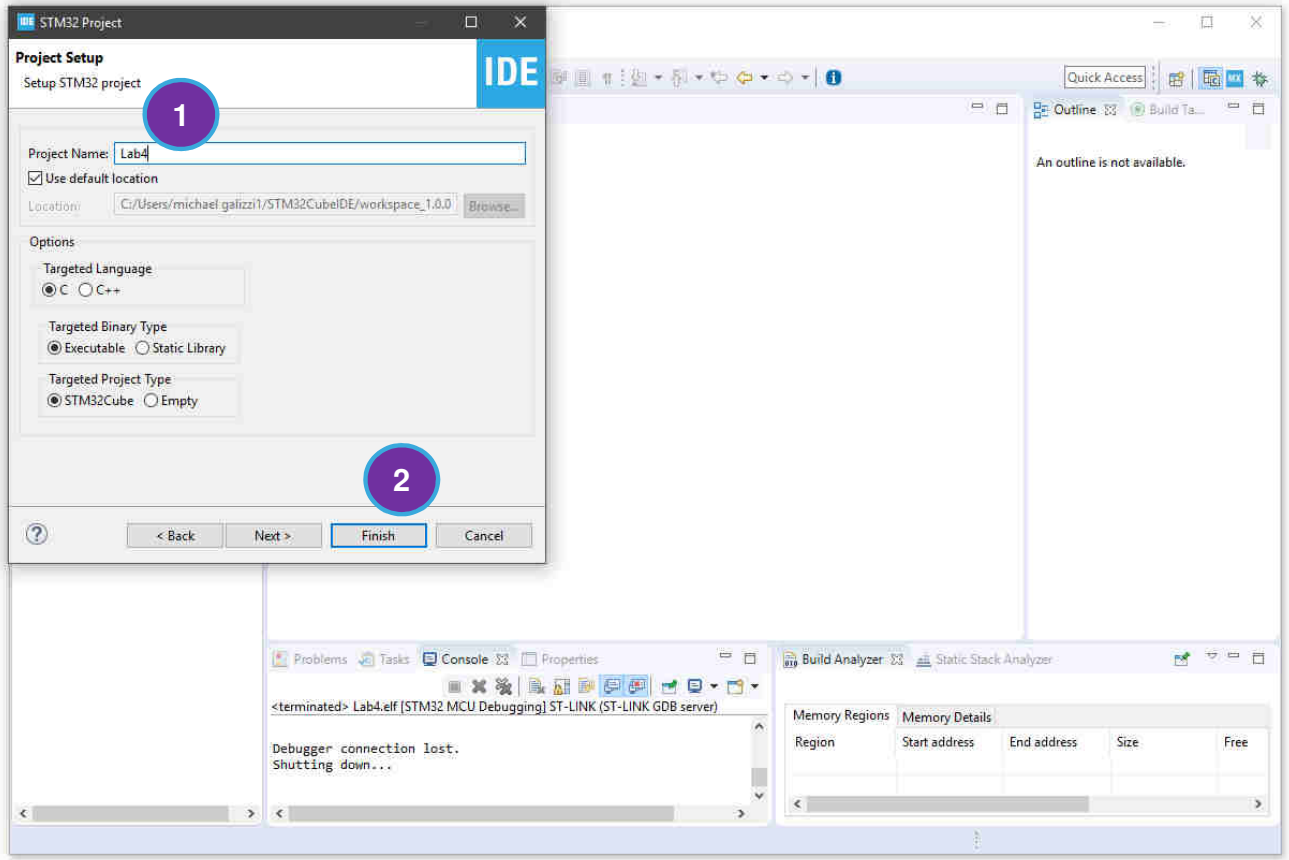


life.augmented

1. Click on **Board Selector**
2. Type **NUCLEO-L476RG**
3. Click on the board
4. Click **Next >**

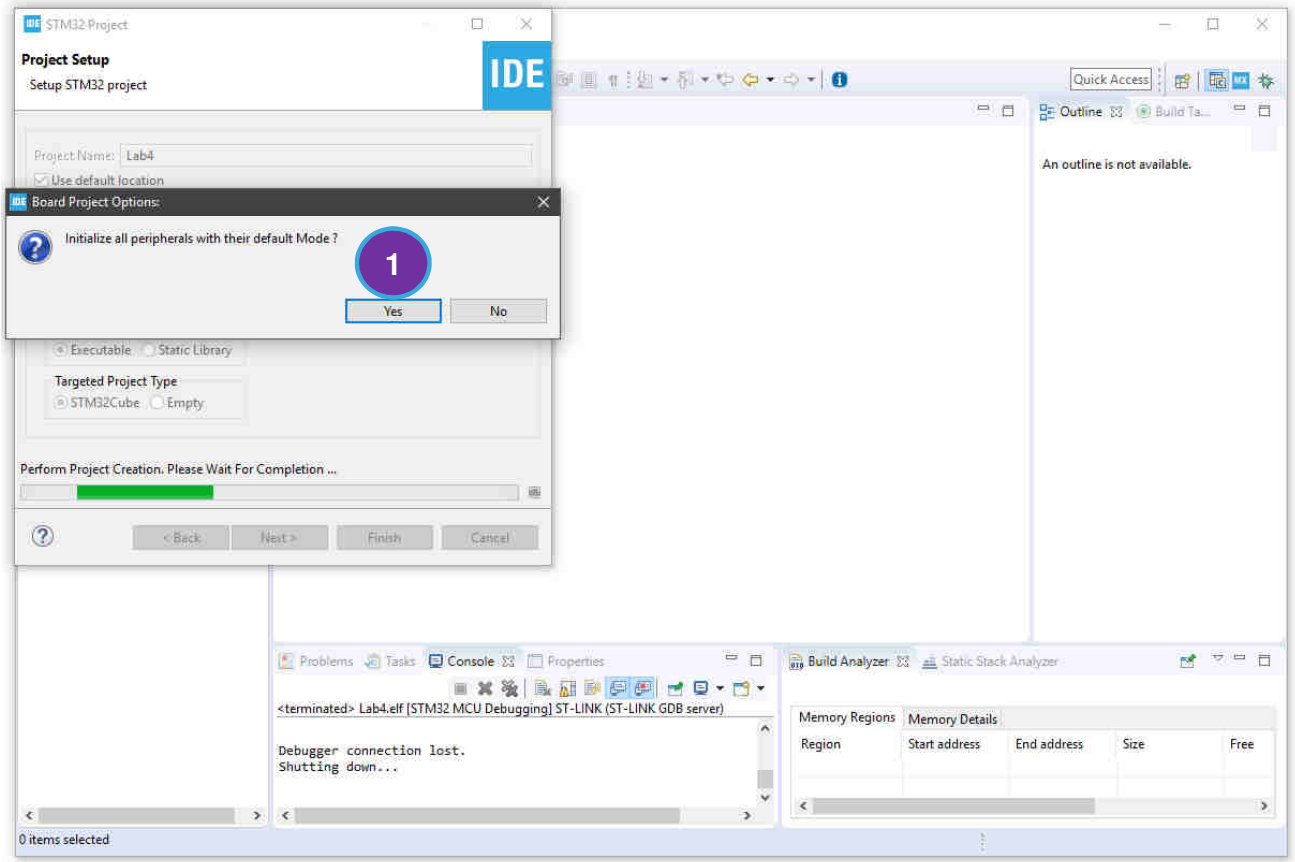


1. Project Name **Lab4**
2. Click **Finish**



life.augmented

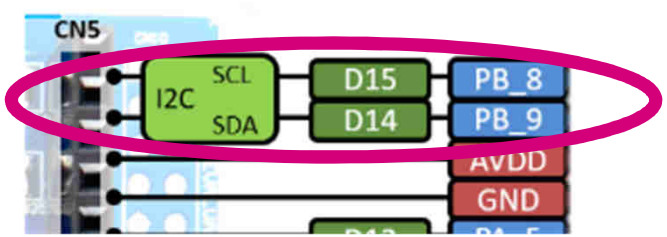
1. Click **Yes** to init peripherals in default mode

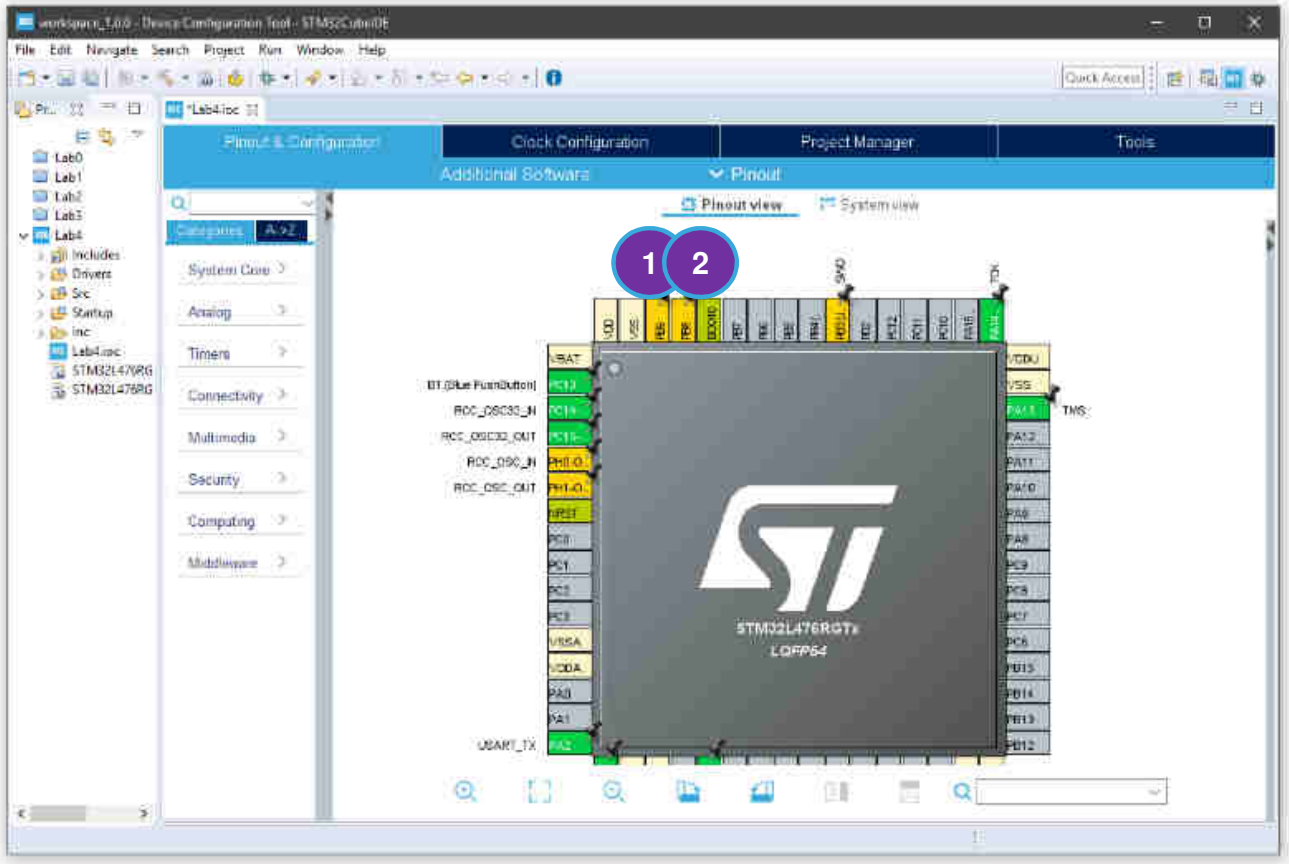


life.augmented

Lab4 – Configure the I2C Bus

1. Left Click on **PB9** and select I2C1_SDA
2. Left Click on **PB8** and select I2C1_SCL

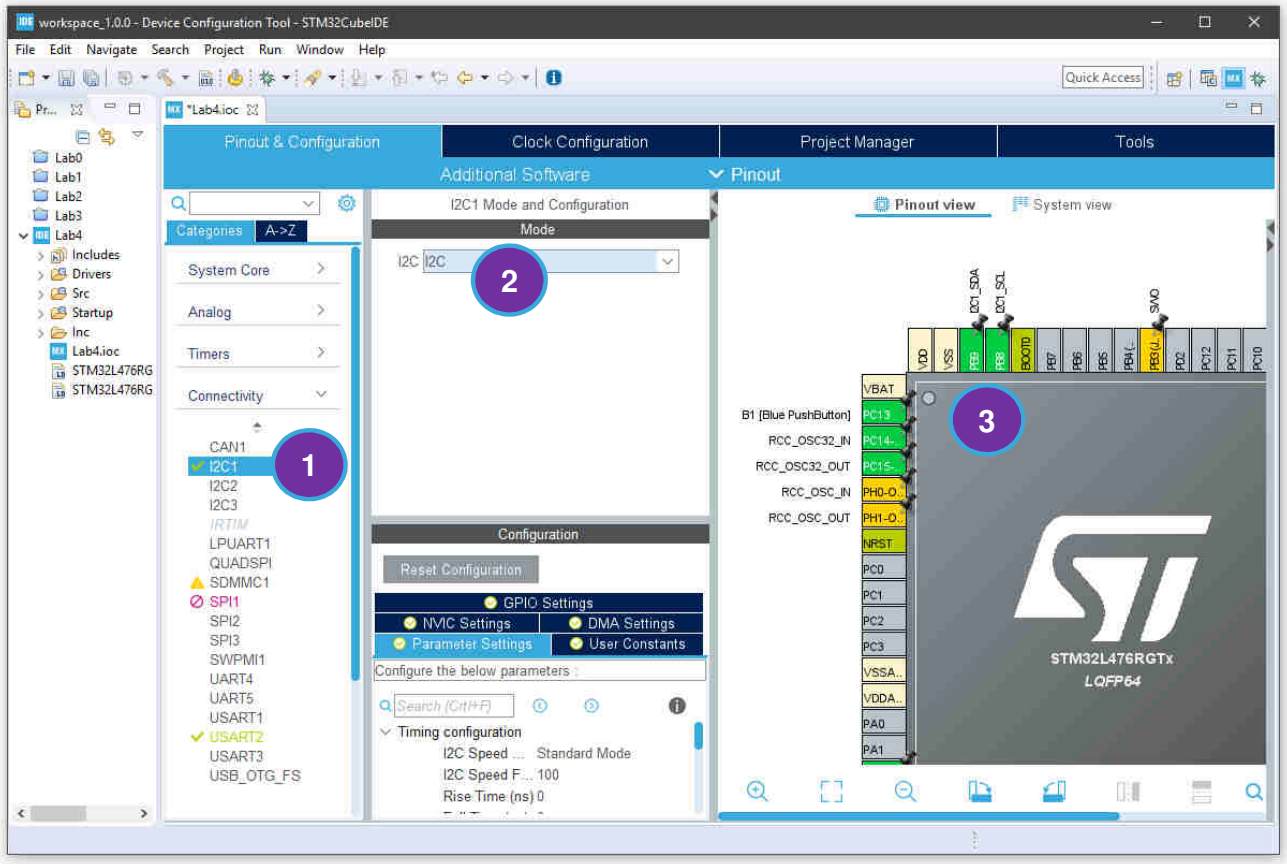




life.augmented

Lab4 – Configure the I2C Bus

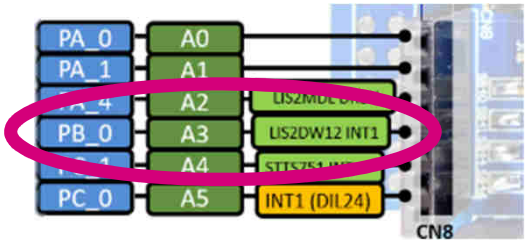
1. Expand *Connectivity* tab and check **I2C1**
2. Select **I2C** in *I2C1 Mode and Configuration*
3. PB8 and PB9 should now become green

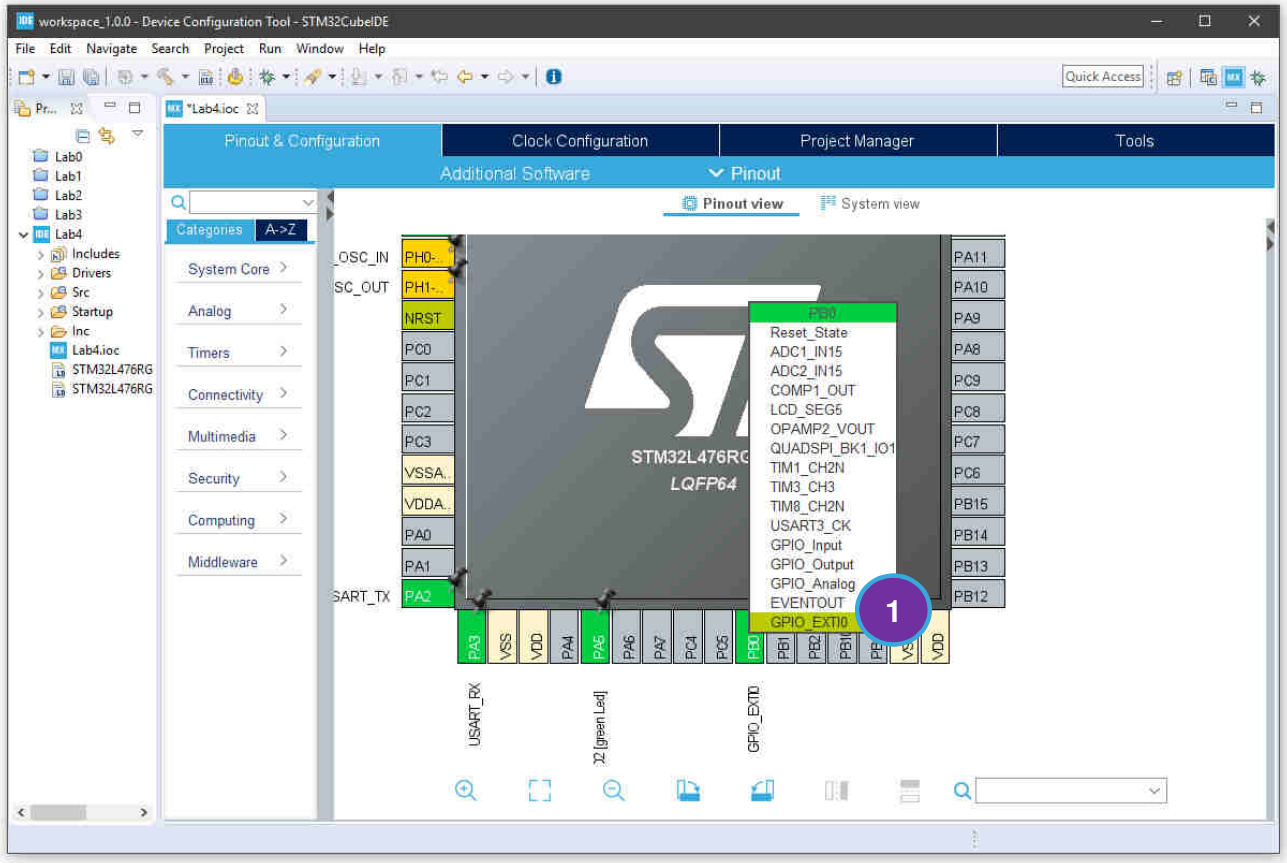


life.augmented

Lab4 – Configure LIS2DW12 Interrupt

1. Left Click on **PB0** and select **GPIO_EXTI0**





life.augmented

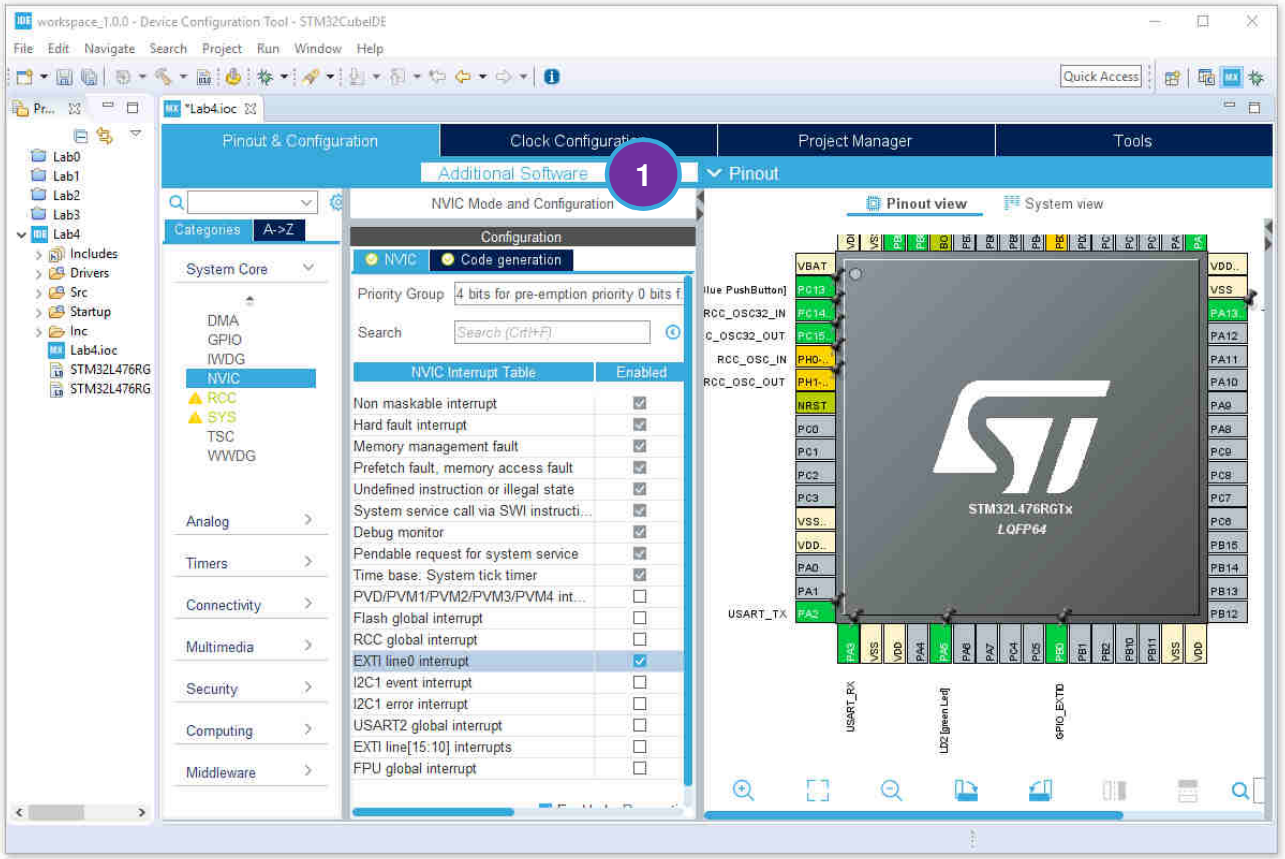
Lab4 – Configure LIS2DW12 Interrupt

1. Check **NVIC** in tab System Core
2. Select **NVIC** in NVIC Mode and Configuration
3. Enable **EXTI line0 interrupt**

life.augmented

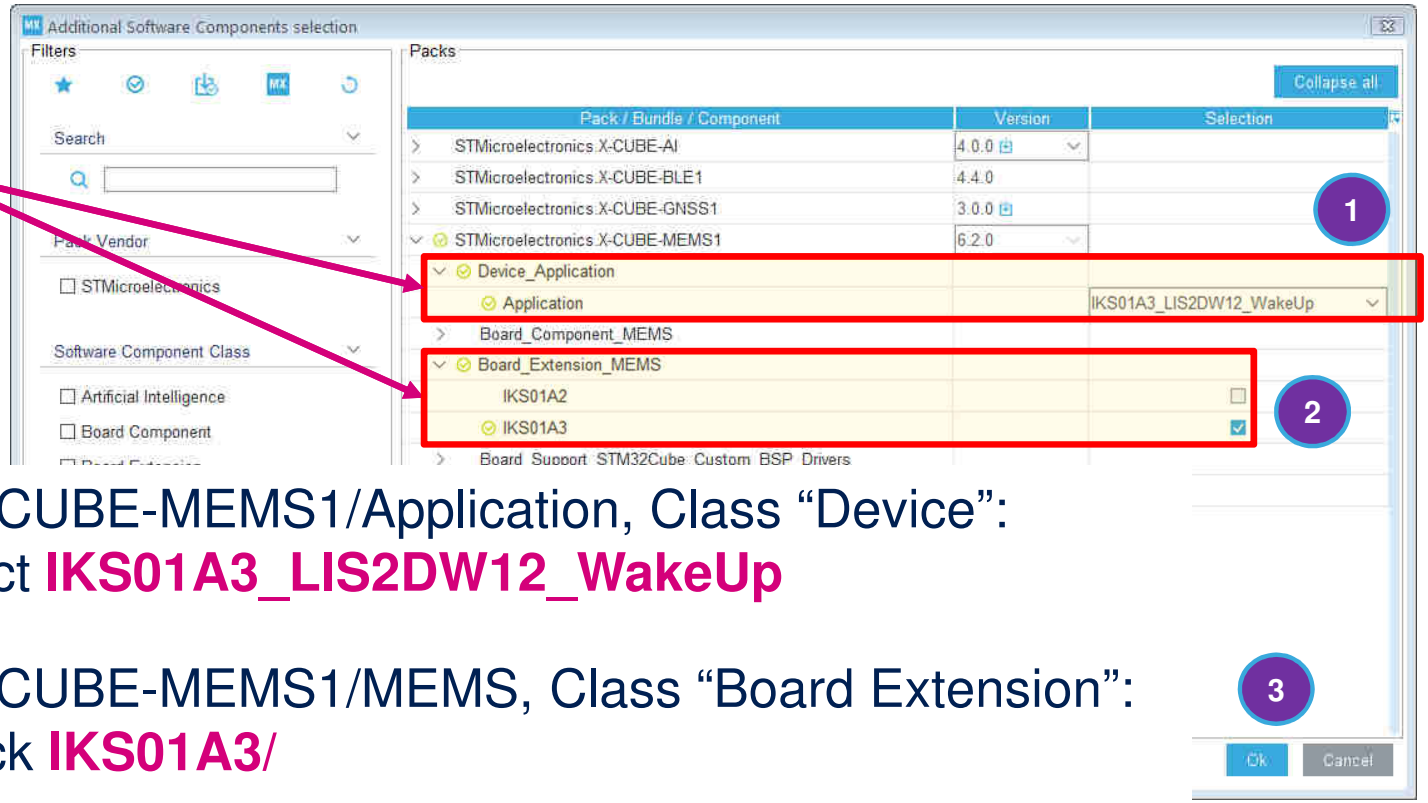
Lab4 – Select the MEMS Library

1. Click on **Additional Software**



Lab4 – Select the MEMS Library

Click to expand tree



1. In X-CUBE-MEMS1/Application, Class “Device”: Select **IKS01A3_LIS2DW12_WakeUp**

2. In X-CUBE-MEMS1/MEMS, Class “Board Extension”: Check **IKS01A3/**

3. Click **OK**

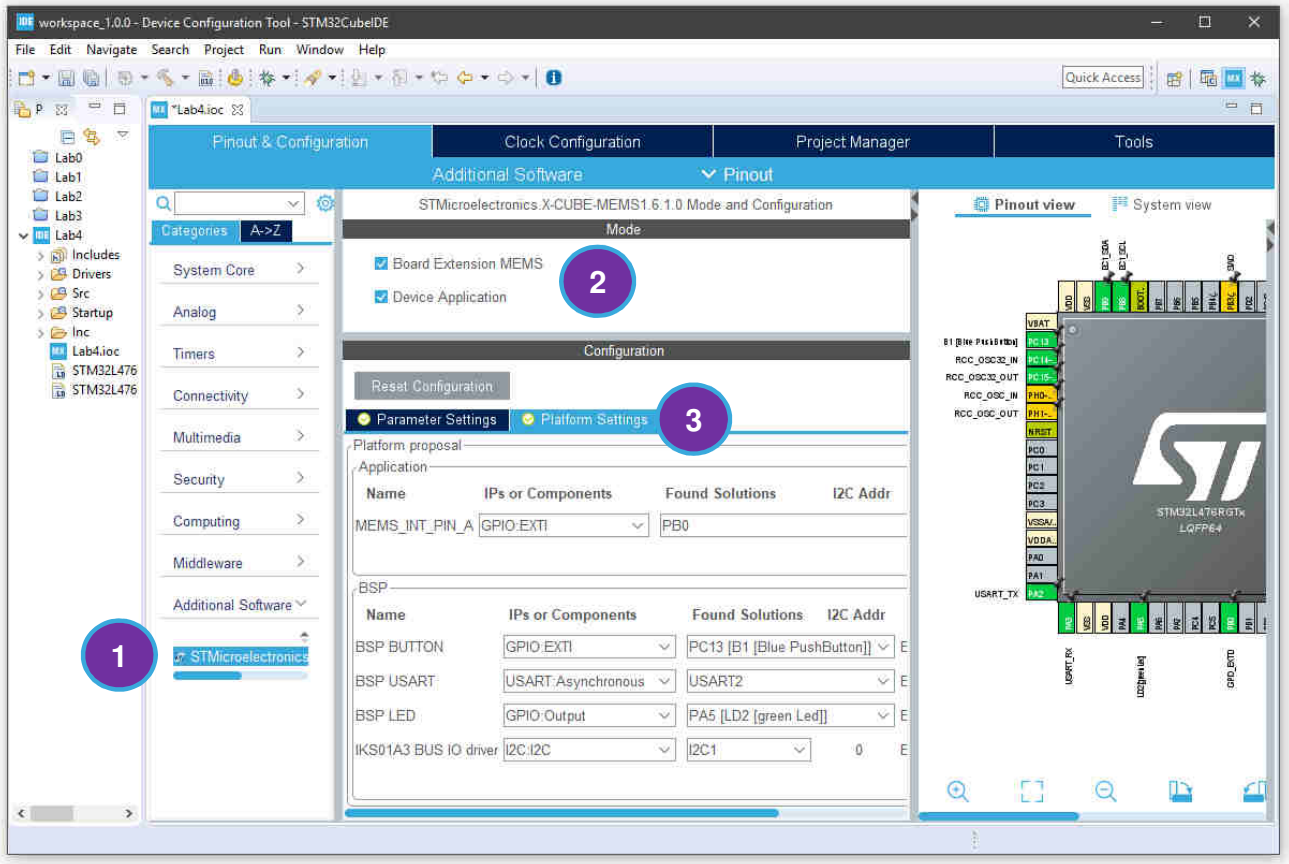
life.augmented

Lab4 – Configure the MEMS Library

1. Expand Additional Software and select the X-CUBE-MEMS1

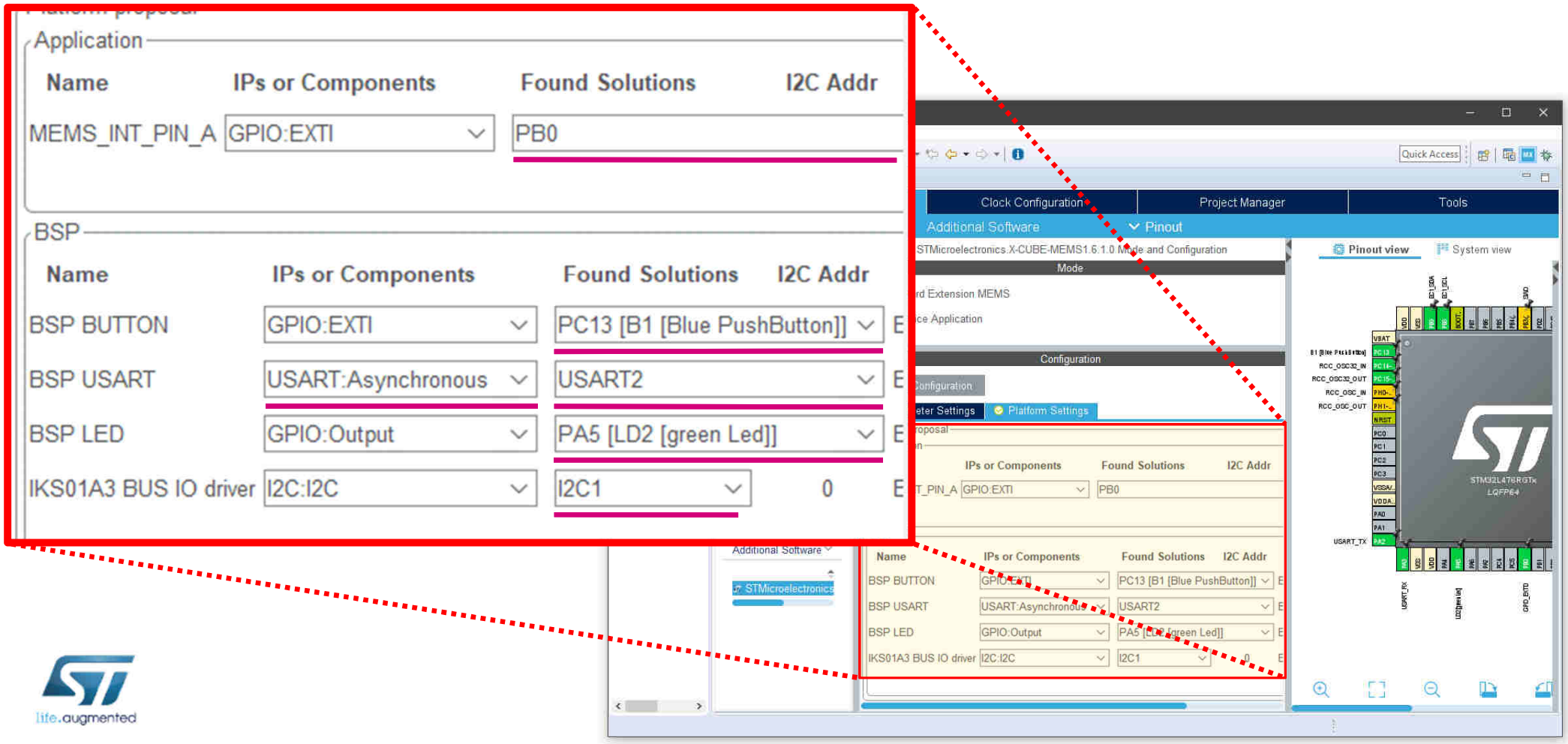
2. Check both:
   Board Extension MEMS
   Device Application

3. Configure Platform Settings as in picture



life.augmented

Lab4 – Configure the MEMS Library

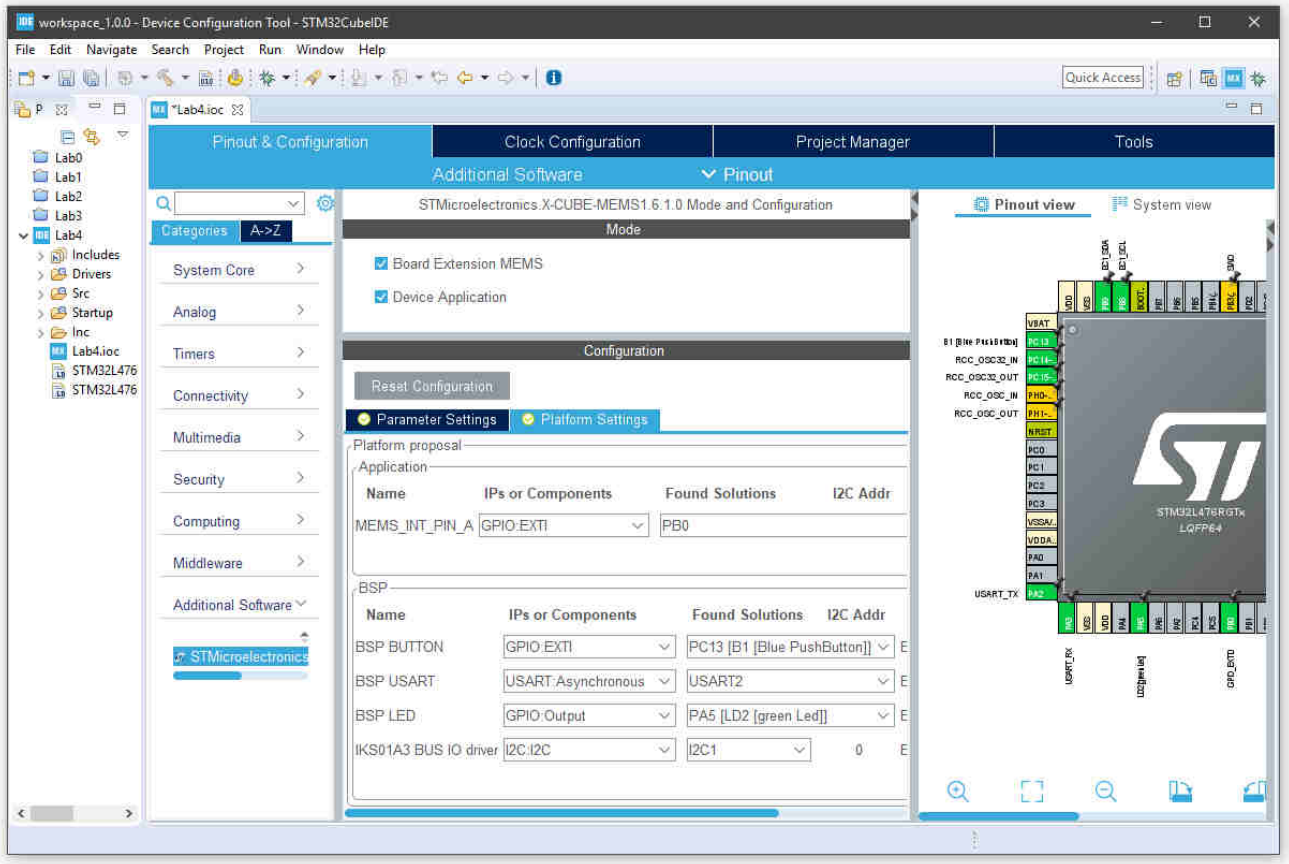


life.augmented

Lab4 – Save the Project

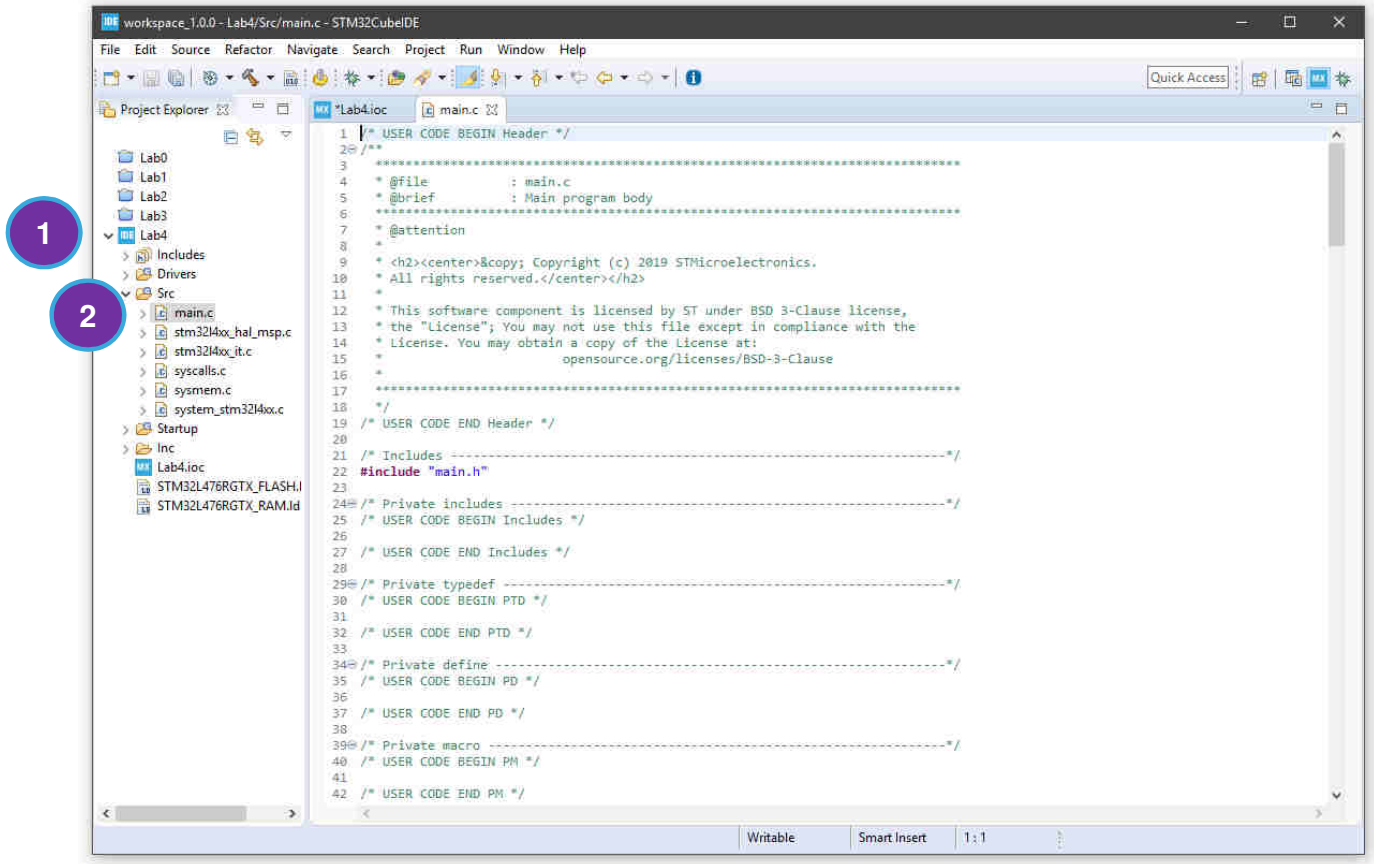
1. Click the save button 

   This action will generate the source code of this lab



Lab4 – Code Editing

1. Expand **Src** in folder **Lab4**
2. Double click on **main.c**

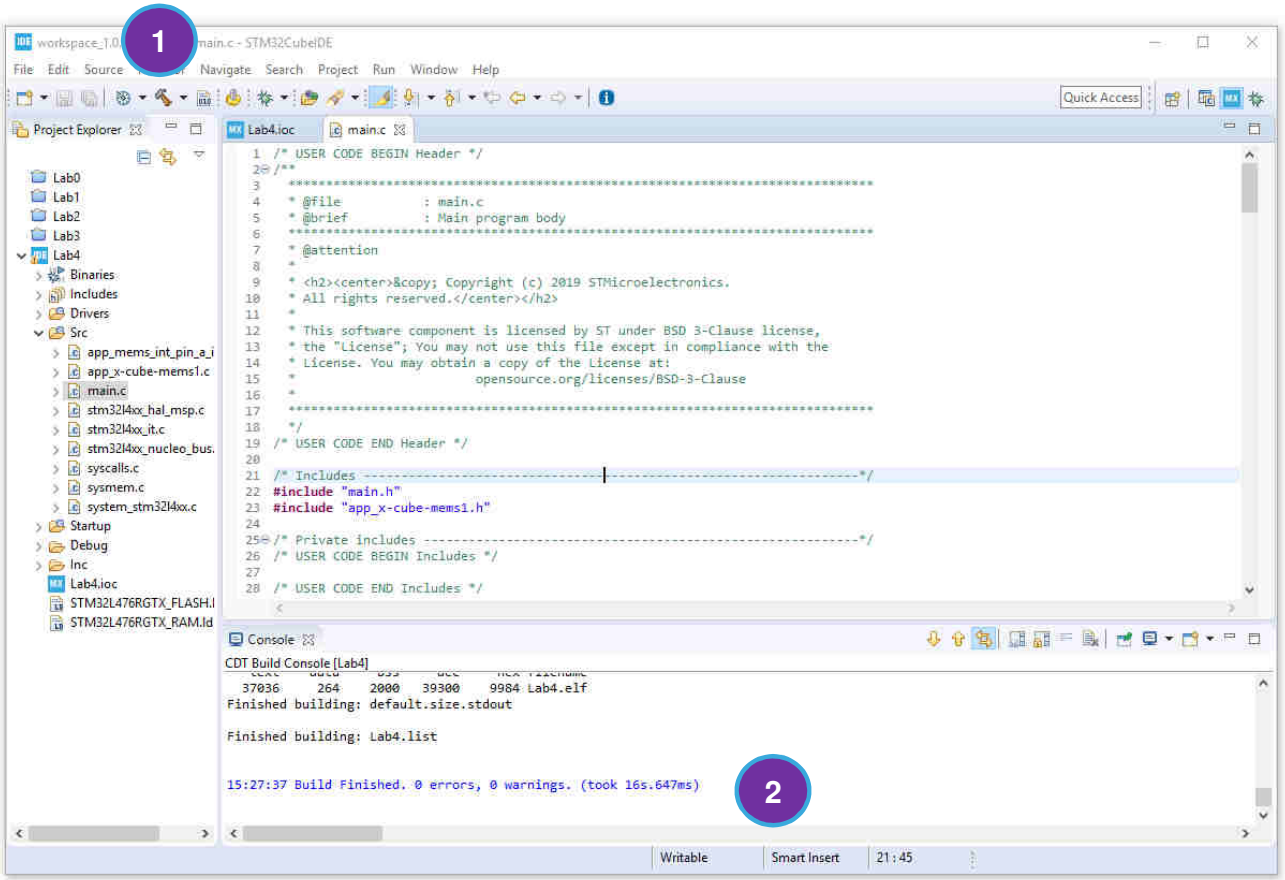


life.augmented

Lab4 - Compiling

1. Click on the hammer  to begin compilation, or press **CTRL+B**

2. Compilation should terminate with 0 errors and 0 warning



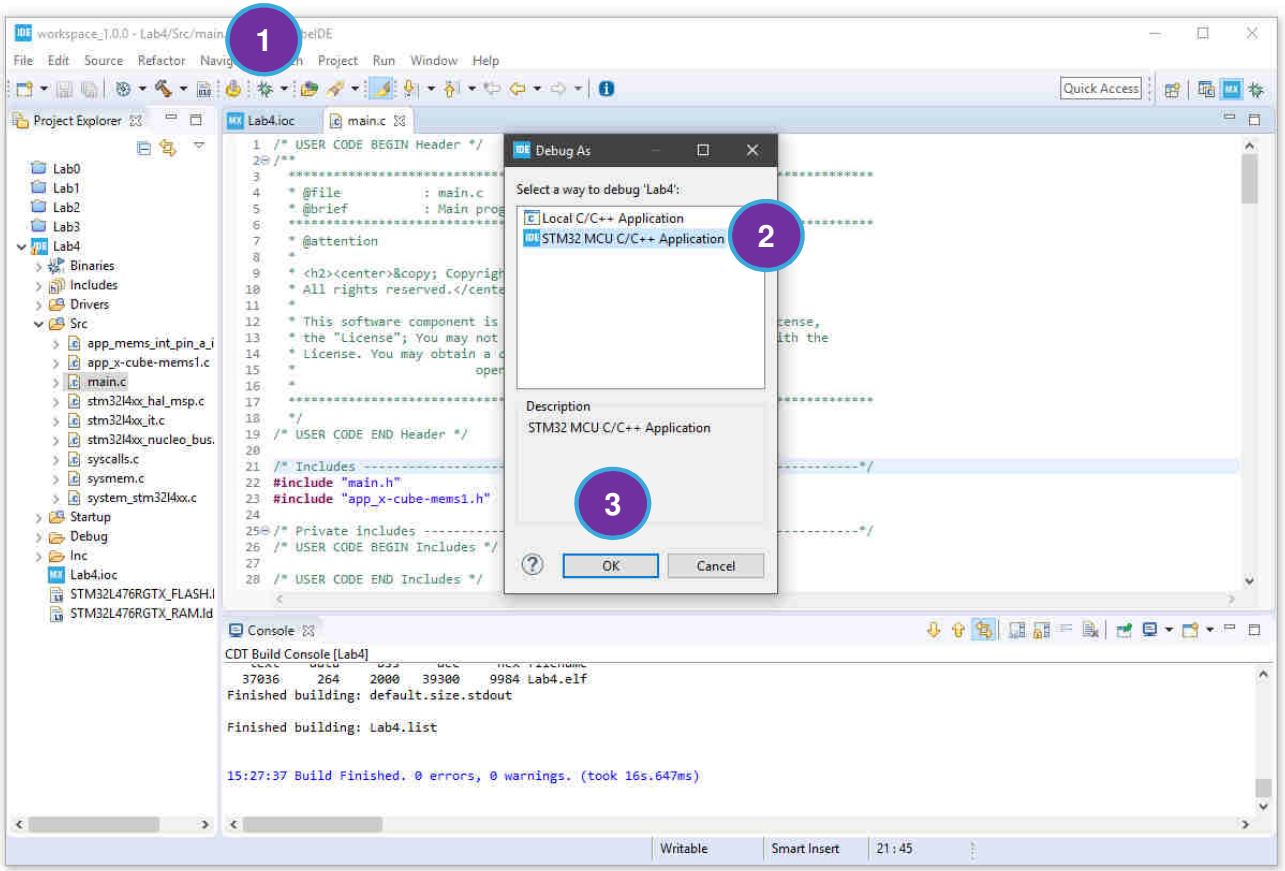
life.augmented

Lab4 - Debugging

1. Click on the bug  to begin debugging

2. Select
   STM32 MCU C/C++ App

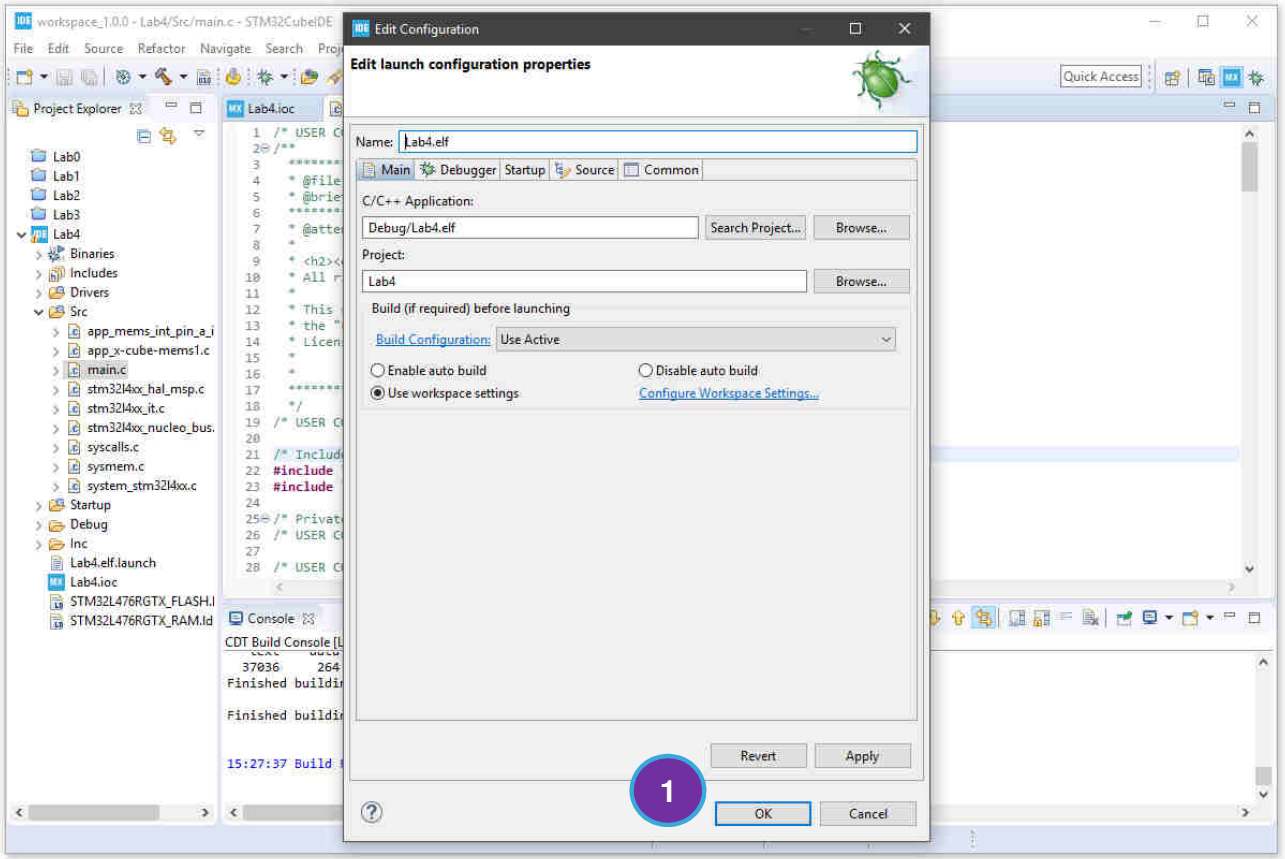
3. Click **OK**



life.augmented

Lab4 - Debugging

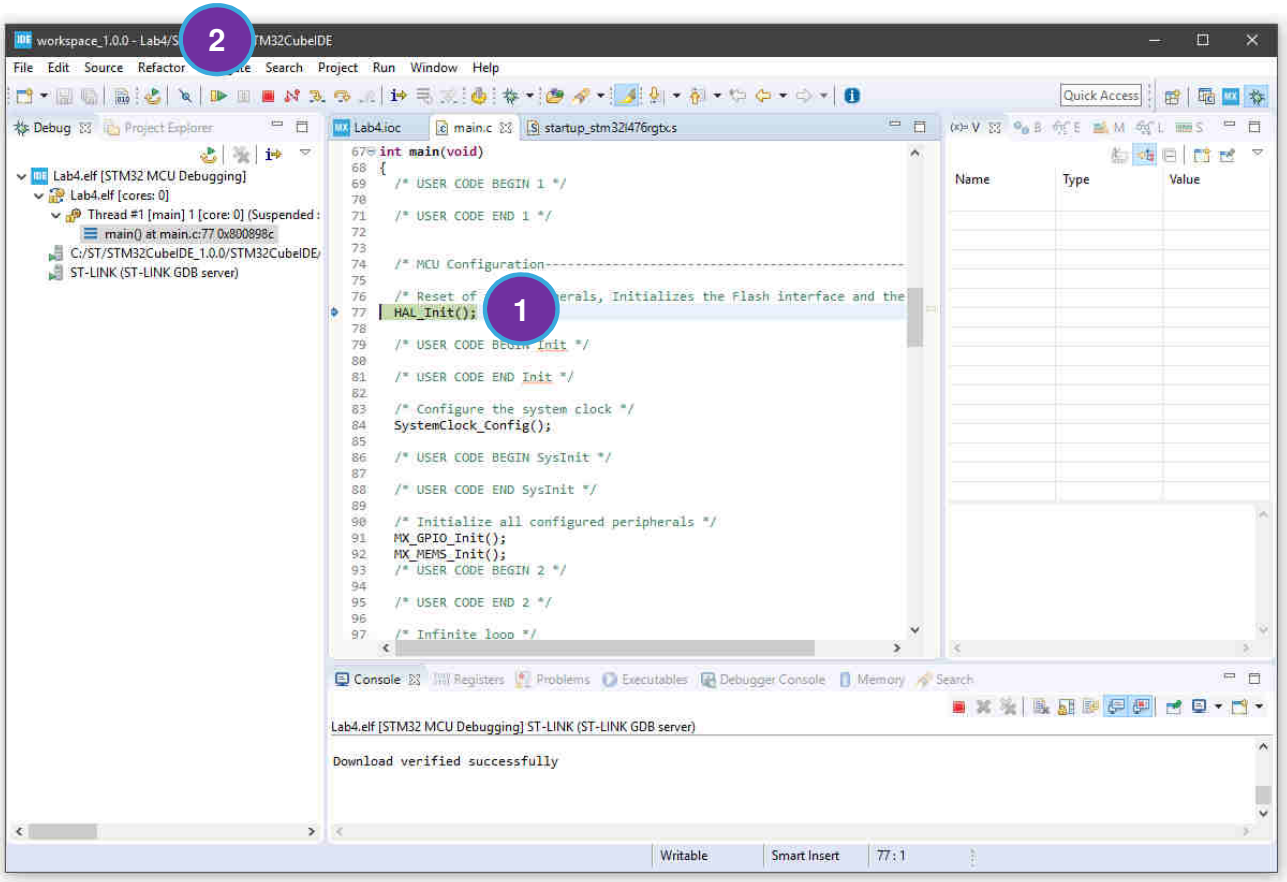
1. Click **OK**



life.augmented

Lab4 - Debugging

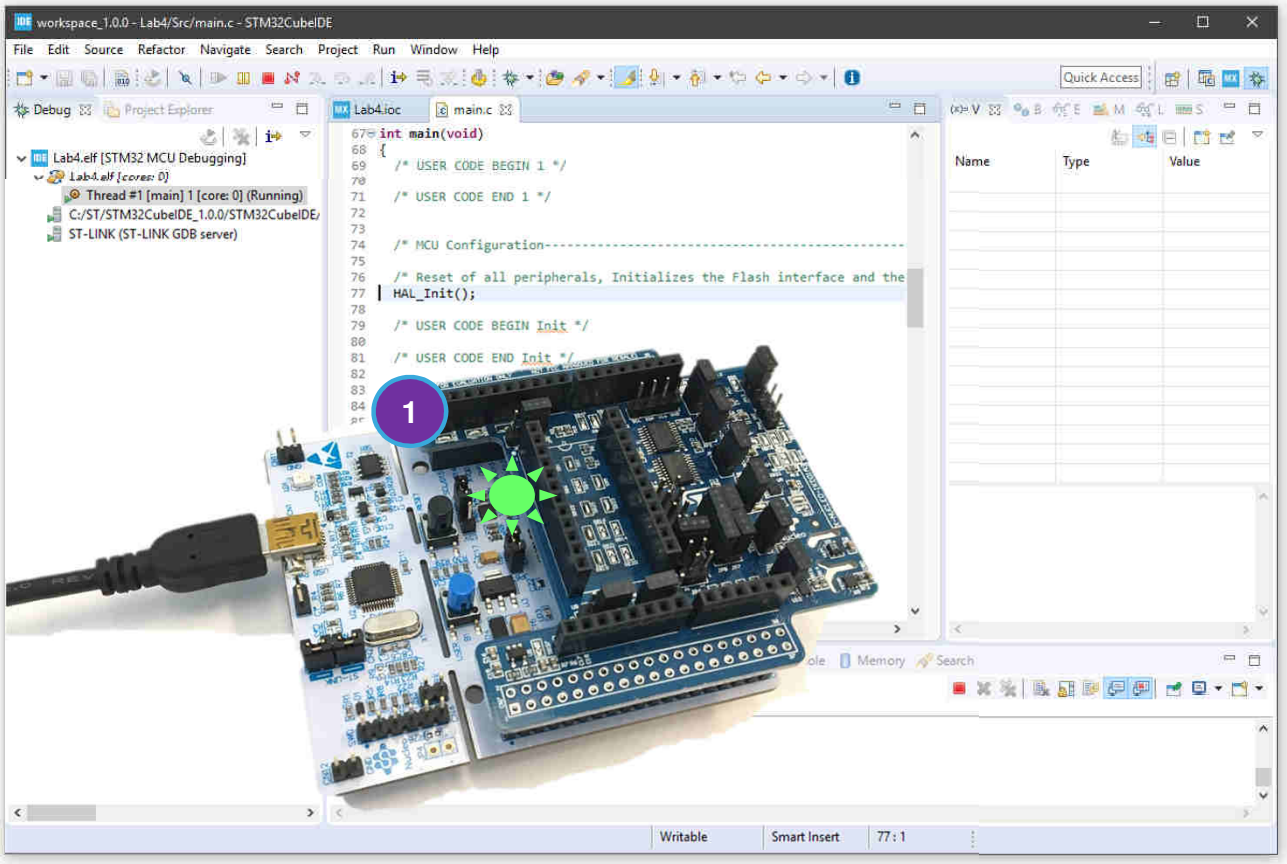
1. Code start at the first line of the main function
2. Click play  button to run the code



life.augmented

Lab4 - Testing

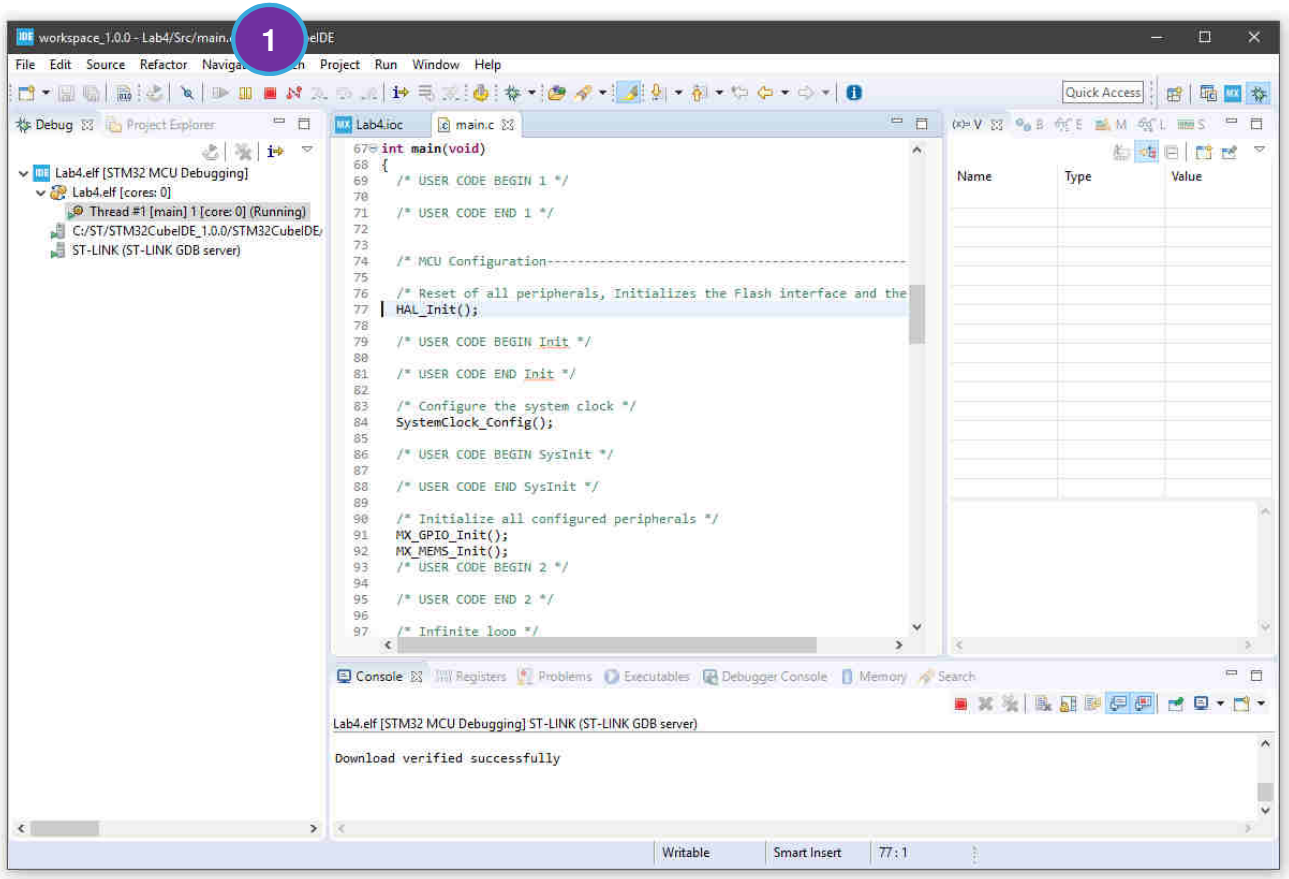
1. Shake the board and the GREEN led will turn ON



life.augmented

Lab4 - Debugging

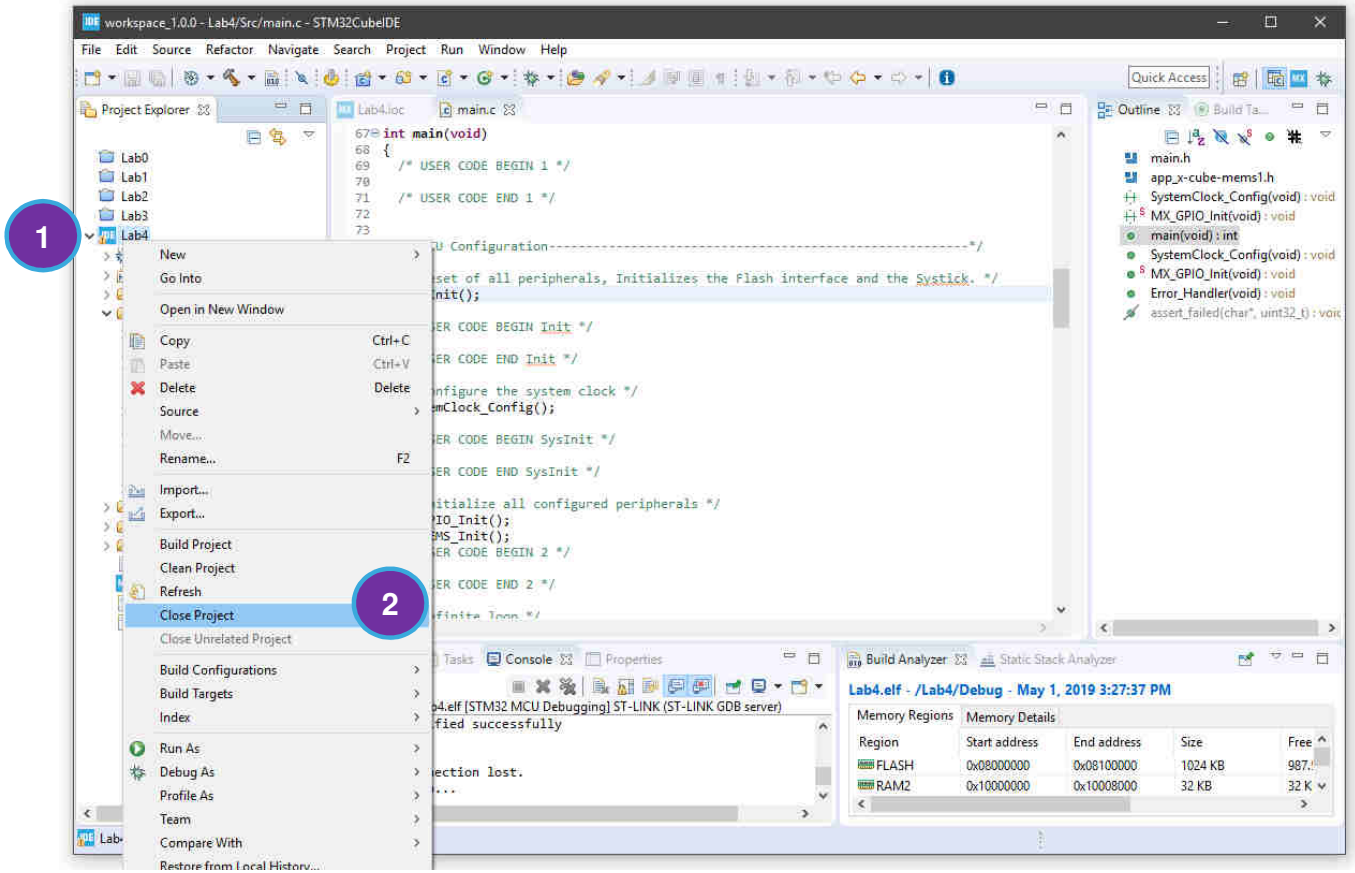
1. Click stop button to interrupt the debugging



life.augmented

Lab4 – Closing the Project

1. Right-Click on **Lab4** project
2. Click on **Close Project**



life.augmented

LAB5

Goals:

- Configure a new project using X-CUBE-MEMS1
- Configure LSM6DSO single and double tap detection
- Enable interrupts in STM32CubeIDE
- Change tap threshold to increase sensitivity

life.augmented

LSM6DSO

Accelerometer + Gyroscope Inertial Measurement Unit

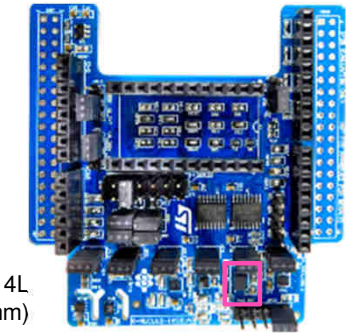
- 0.45mA power consumption (normal mode)
  - Accelerometer: 0.05mA, Gyroscope: 0.40mA
- Auxiliary SPI typically used for OIS / EIS or closed loop control; I3C Interface
- 9kB equivalent FIFO Memory for local data storage
- Finite State Machines (up to 16)
- Digital Features
  - Free fall
  - Pedometer 2.0
  - 6D / 4D
  - Tilt detection
  - Tap/ Double Tap

Lowest power consumption IMU → battery saving

Design Flexibility and cost optimization

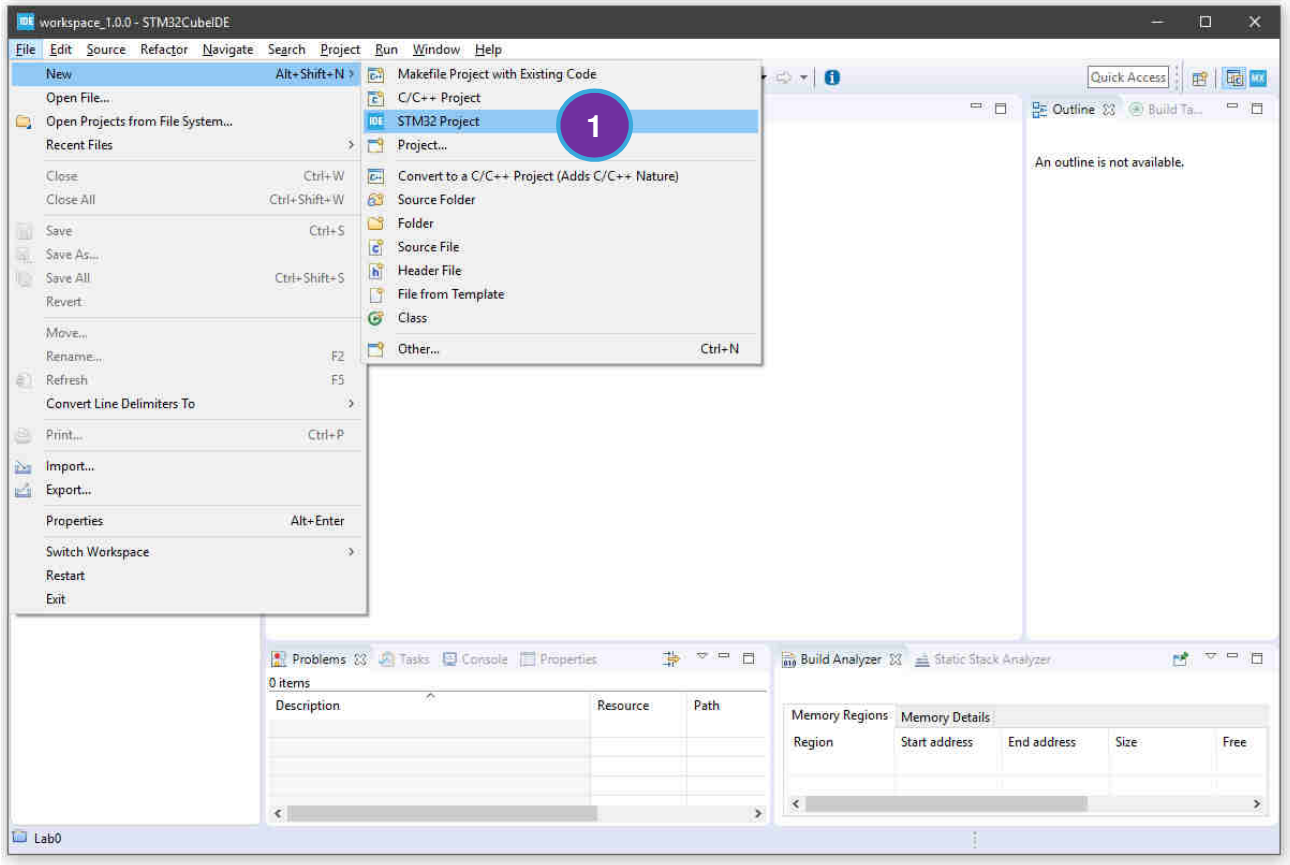
System power saving

Smart Functions with High Flexibility



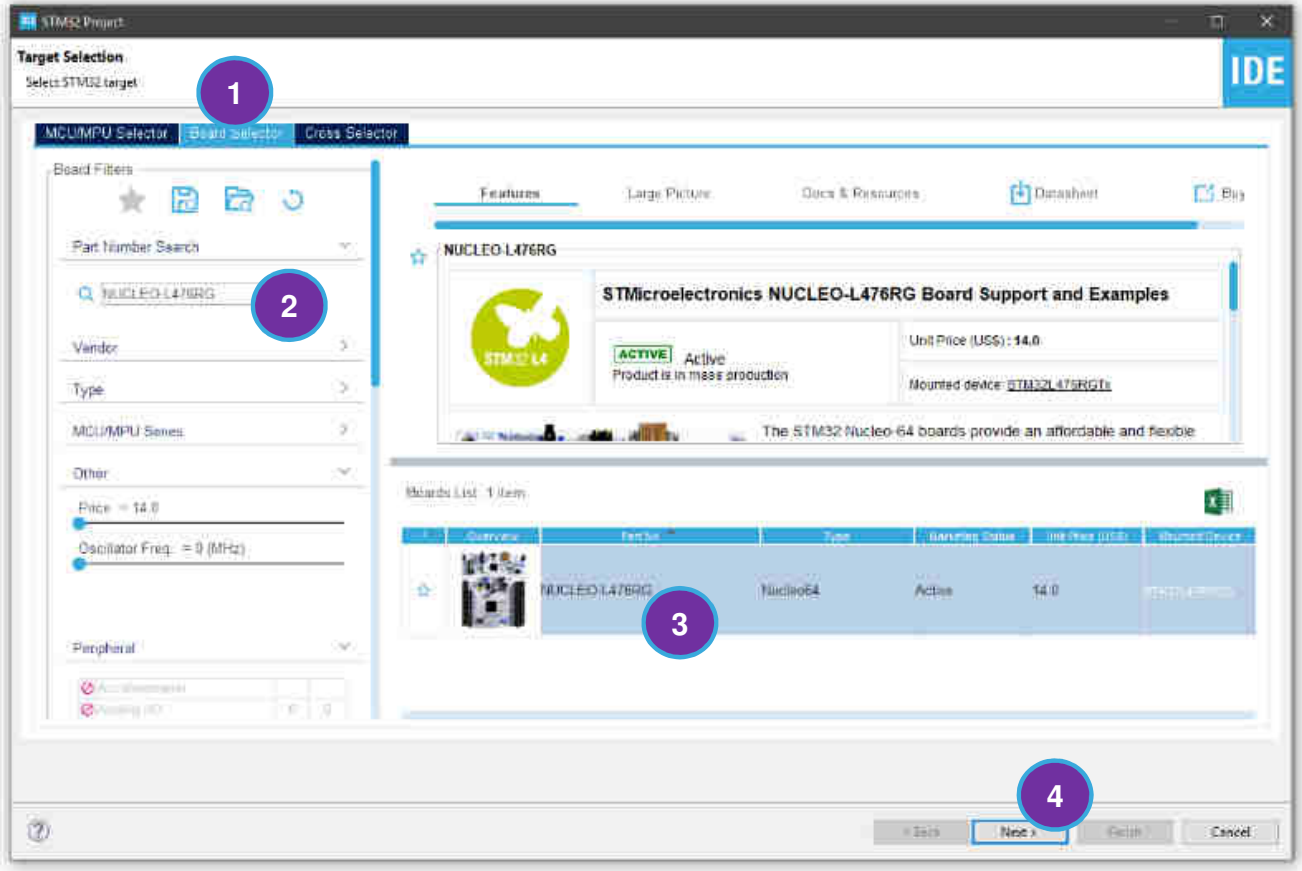
LGA-14L
(2.5 x 3 x 0.83 mm)

1. Click on **File > New > STM32 Project**



life.augmented

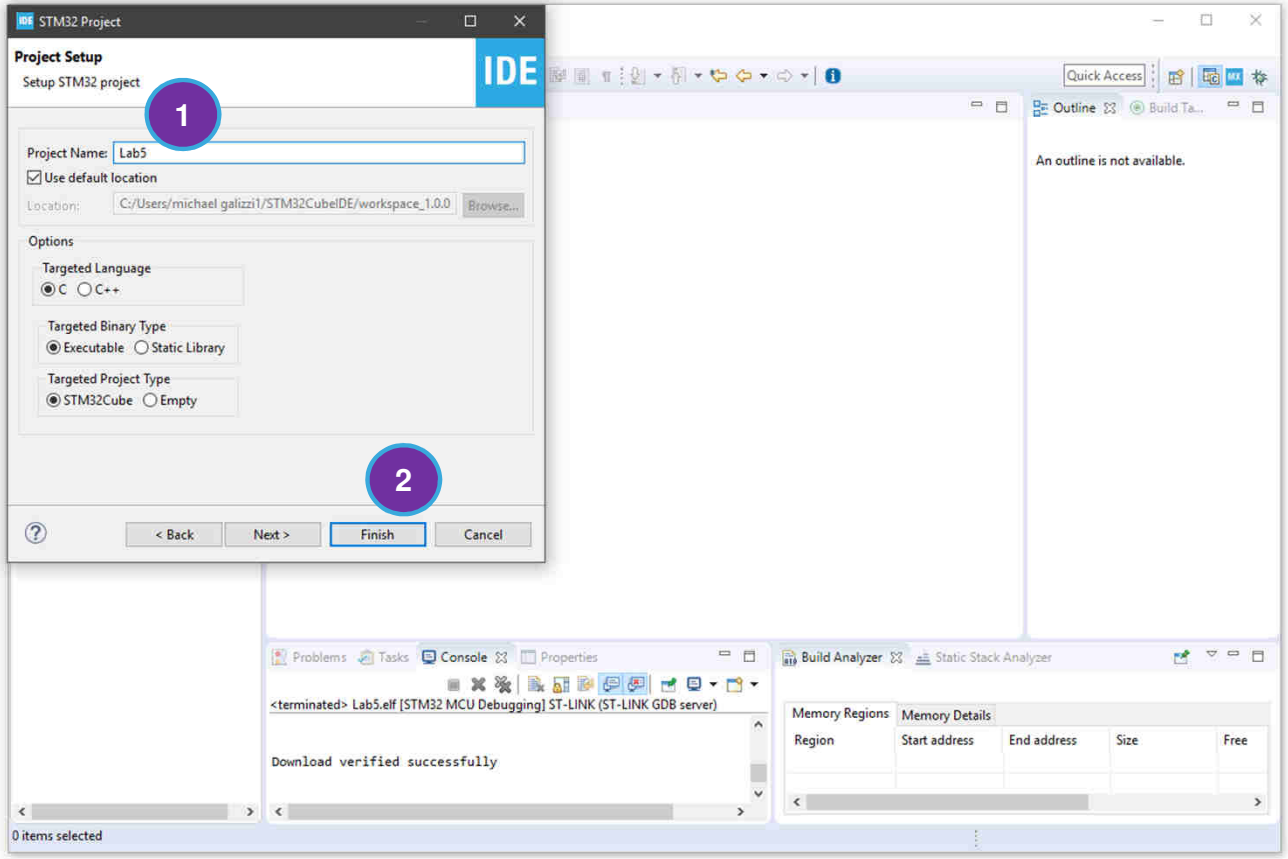
1. Click on **Board Selector**
2. Type **NUCLEO-L476RG**
3. Click on the board
4. Click **Next >**



life.augmented

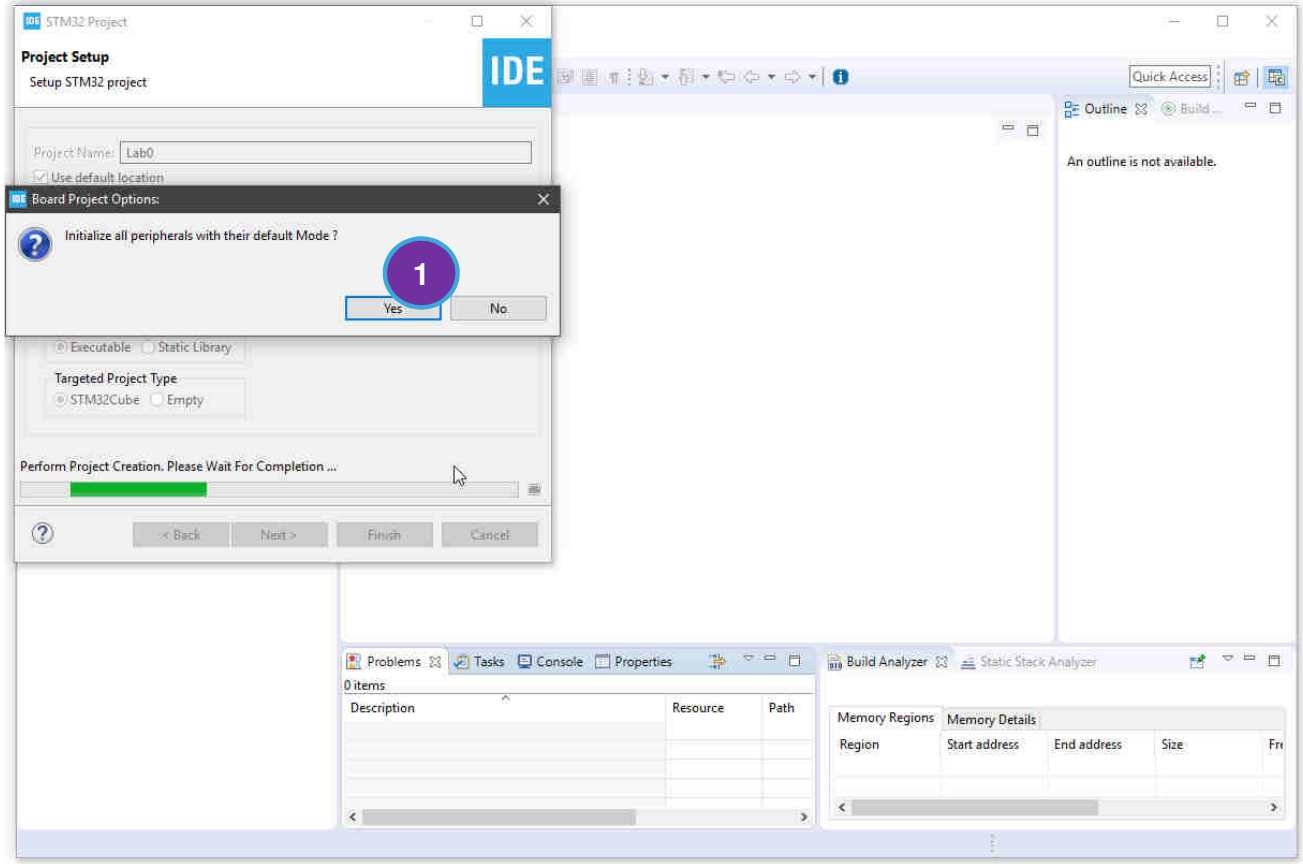
Lab5

1. Project Name **Lab5**
2. Click **Finish**



life.augmented

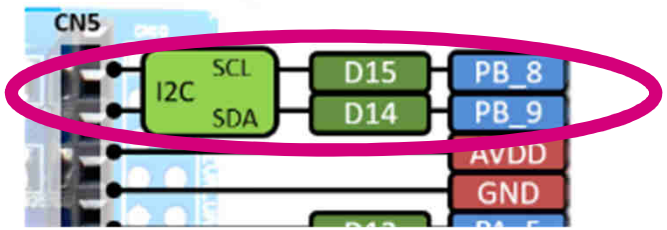
1. Click **Yes** to init peripherals in default mode

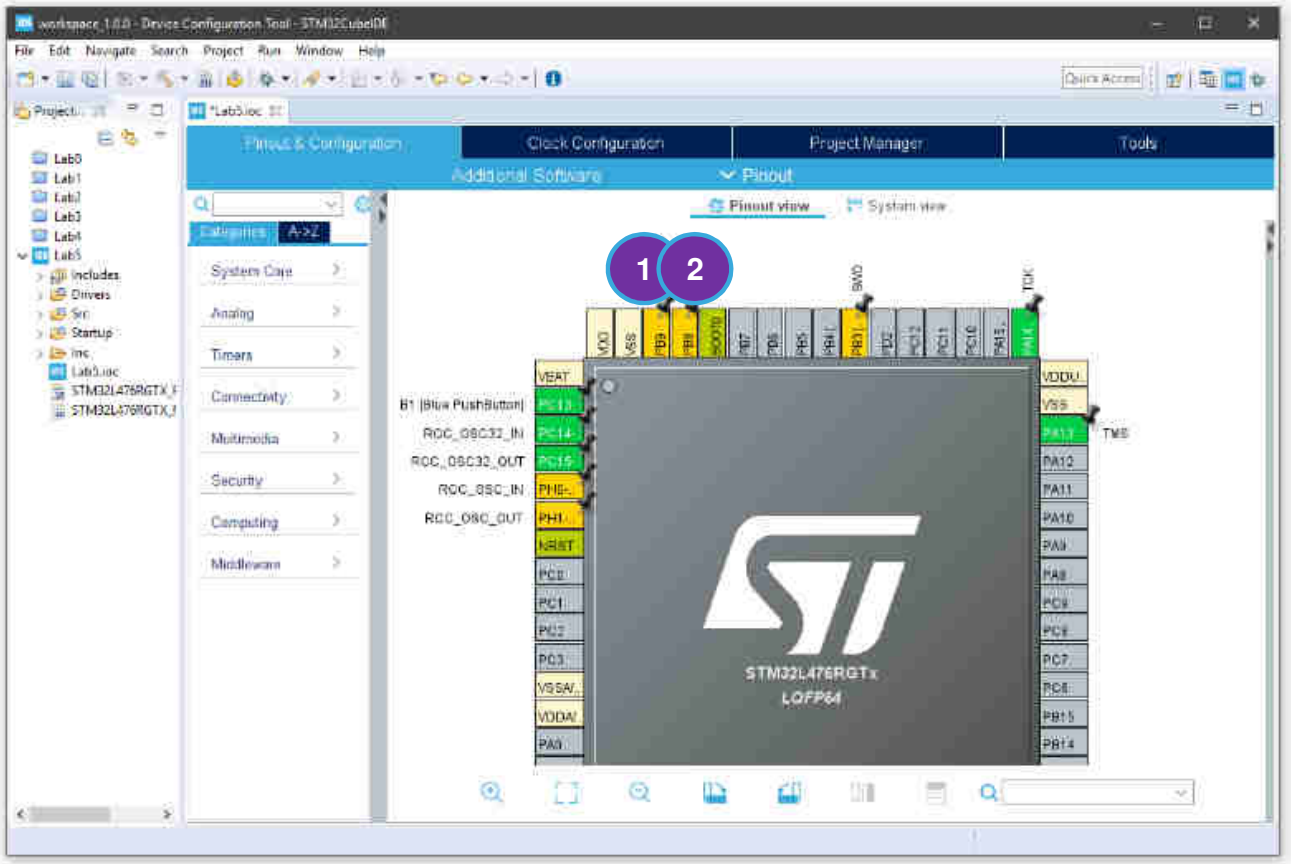


life.augmented

Lab5 – Configure the I2C Bus

1. Left Click on **PB9** and select I2C1_SDA
2. Left Click on **PB8** and select I2C1_SCL

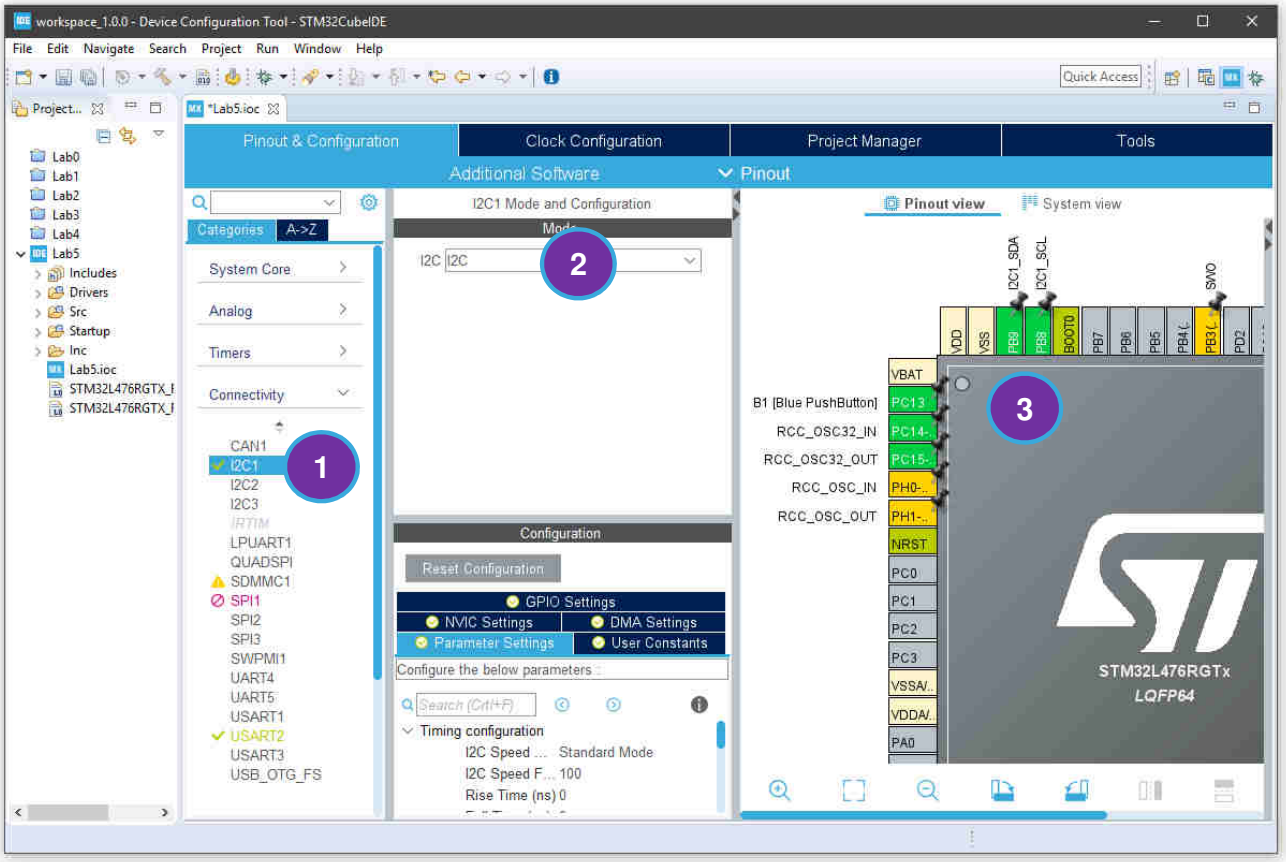




life.augmented

Lab5 – Configure the I2C Bus

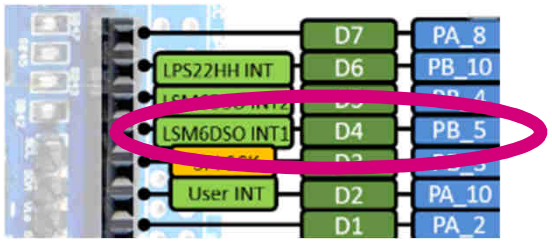
1. Expand *Connectivity* tab and check **I2C1**
2. Select **I2C** in *I2C1 Mode and Configuration*
3. PB8 and PB9 should now become green

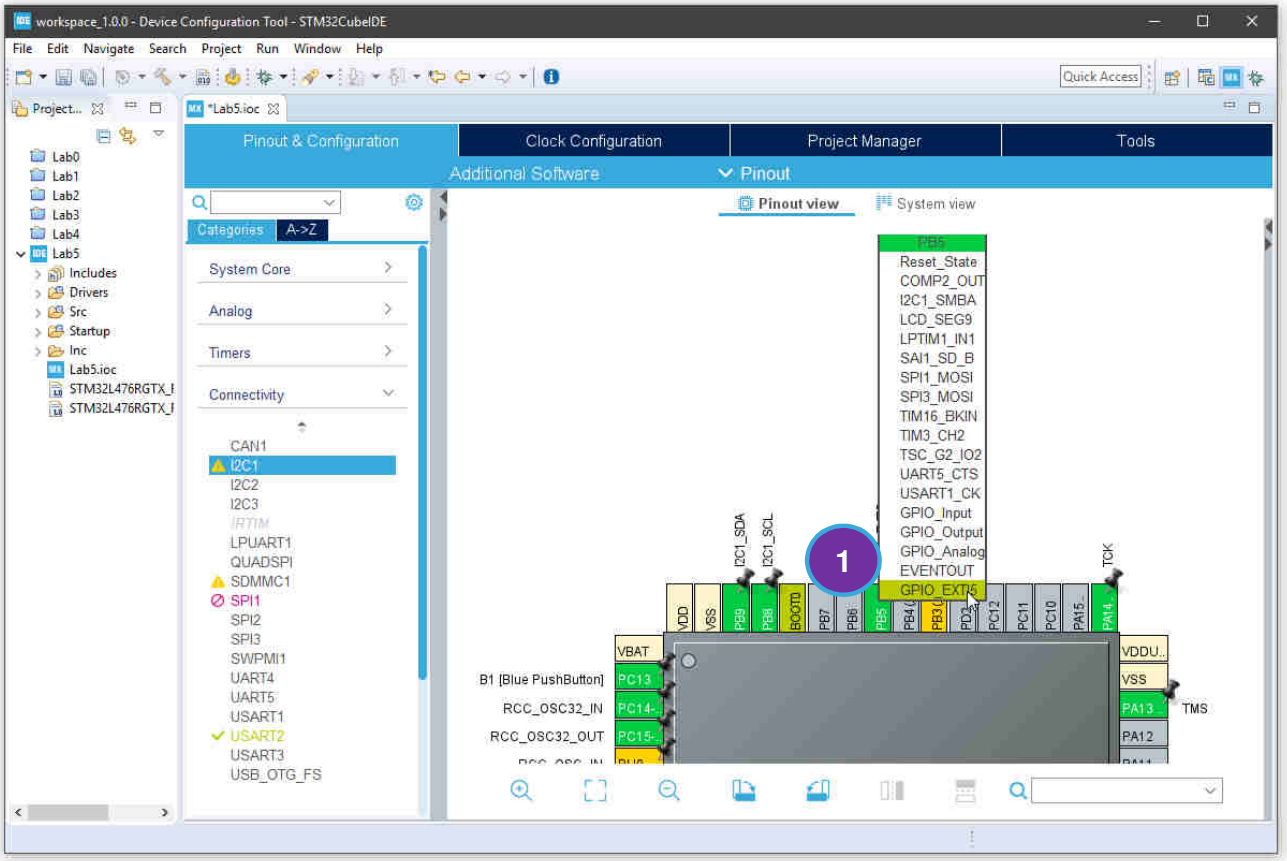


life.augmented

Lab5 – Configure LSM6DSO Interrupt

1. Left Click on **PB5** and select **GPIO_EXTI5**

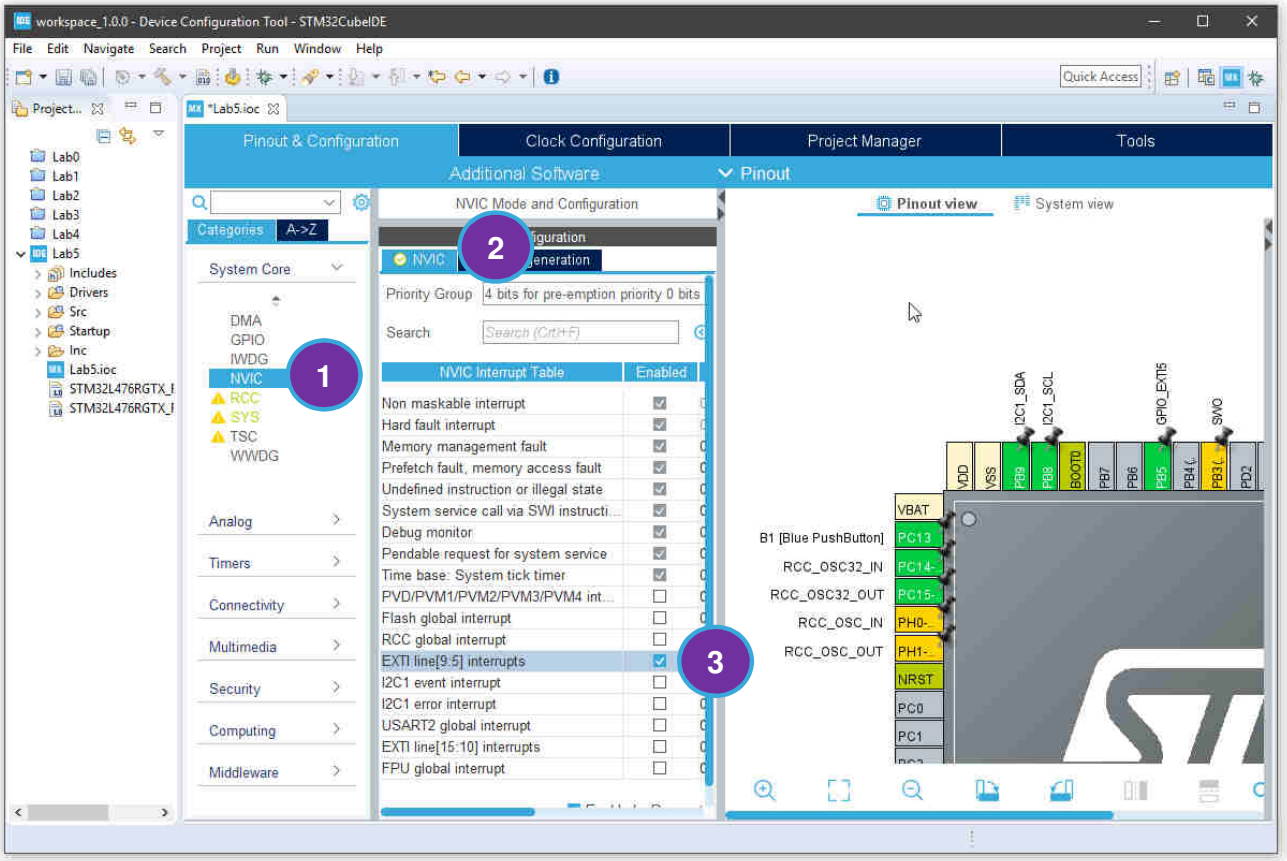




life.augmented

Lab5 – Configure LSM6DSO Interrupt

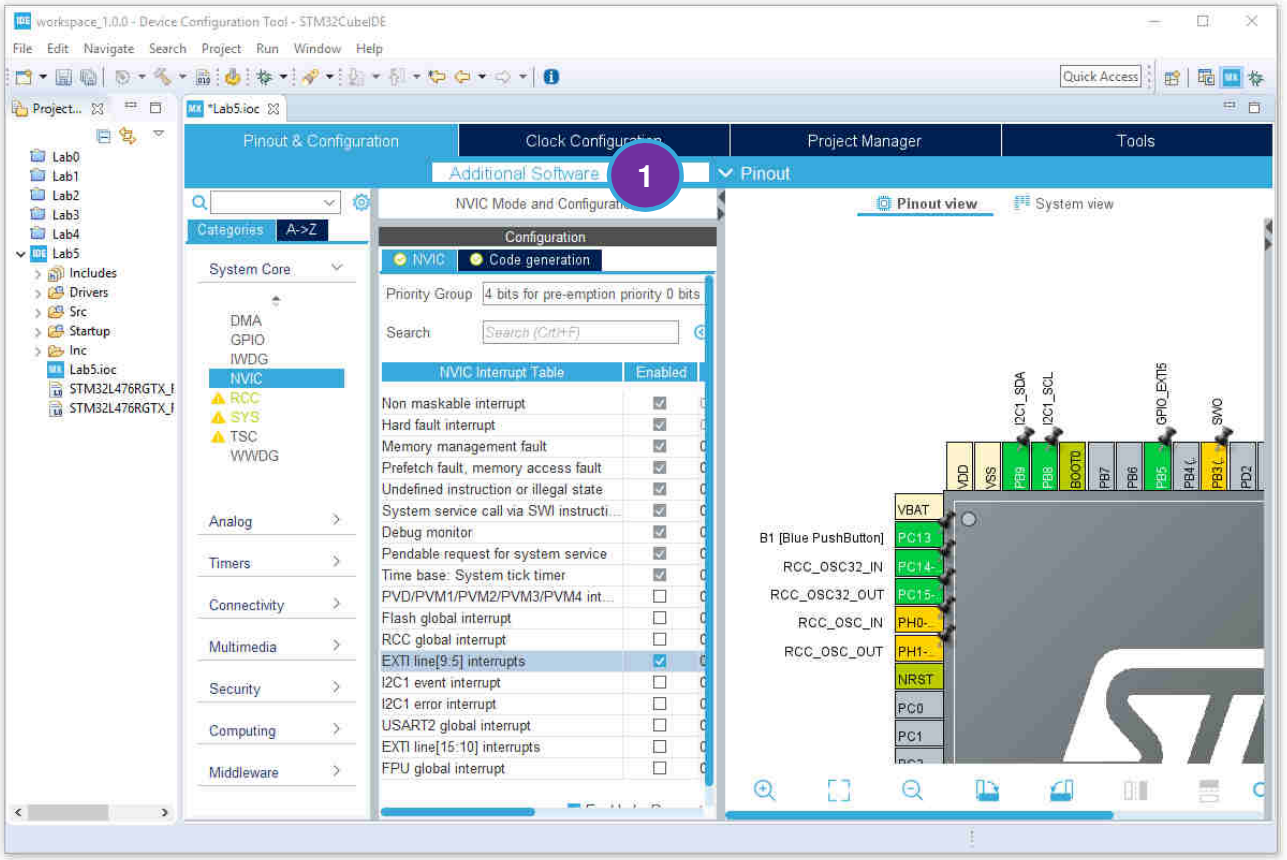
1. Check **NVIC** in tab System Core
2. Select **NVIC** in NVIC Mode and Configuration
3. Enable **EXTI line[9:5] interrupt**



life.augmented

Lab5 – Select the MEMS Library

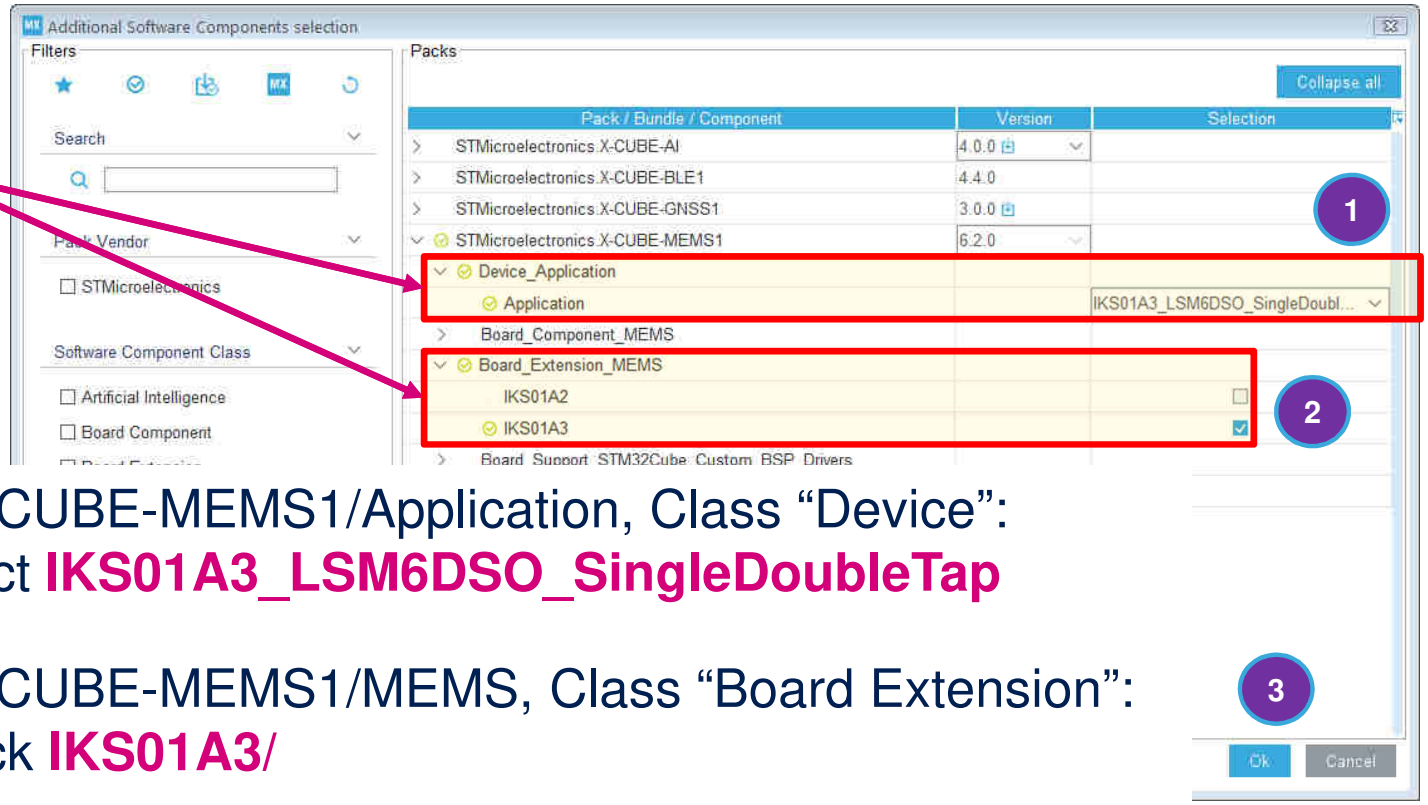
1. Click on **Additional Software**



life.augmented

Lab5 – Select the MEMS Library

Click to expand tree



1. In X-CUBE-MEMS1/Application, Class “Device”: Select **IKS01A3_LSM6DSO_SingleDoubleTap**

2. In X-CUBE-MEMS1/MEMS, Class “Board Extension”: Check **IKS01A3/**

3. Click **OK**

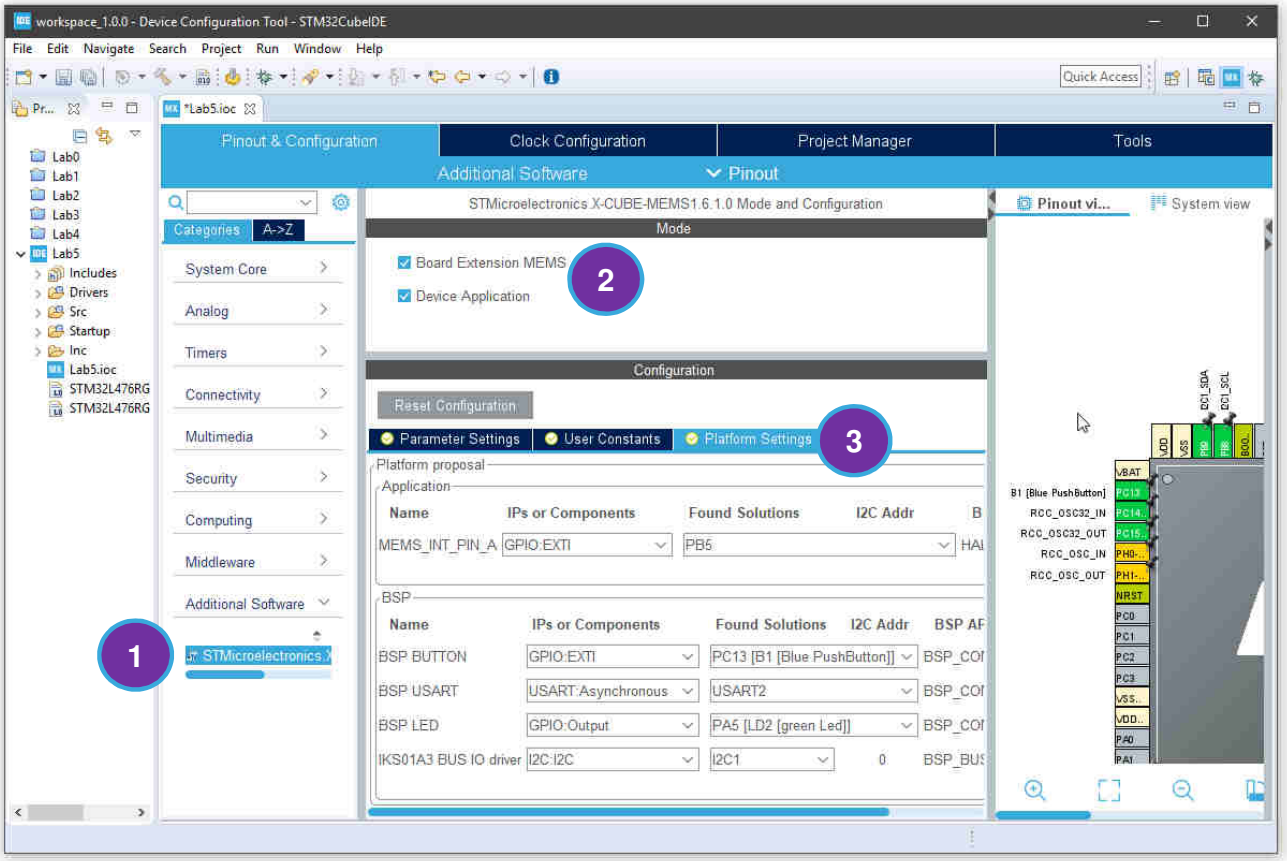
life.augmented

Lab5 – Configure the MEMS Library

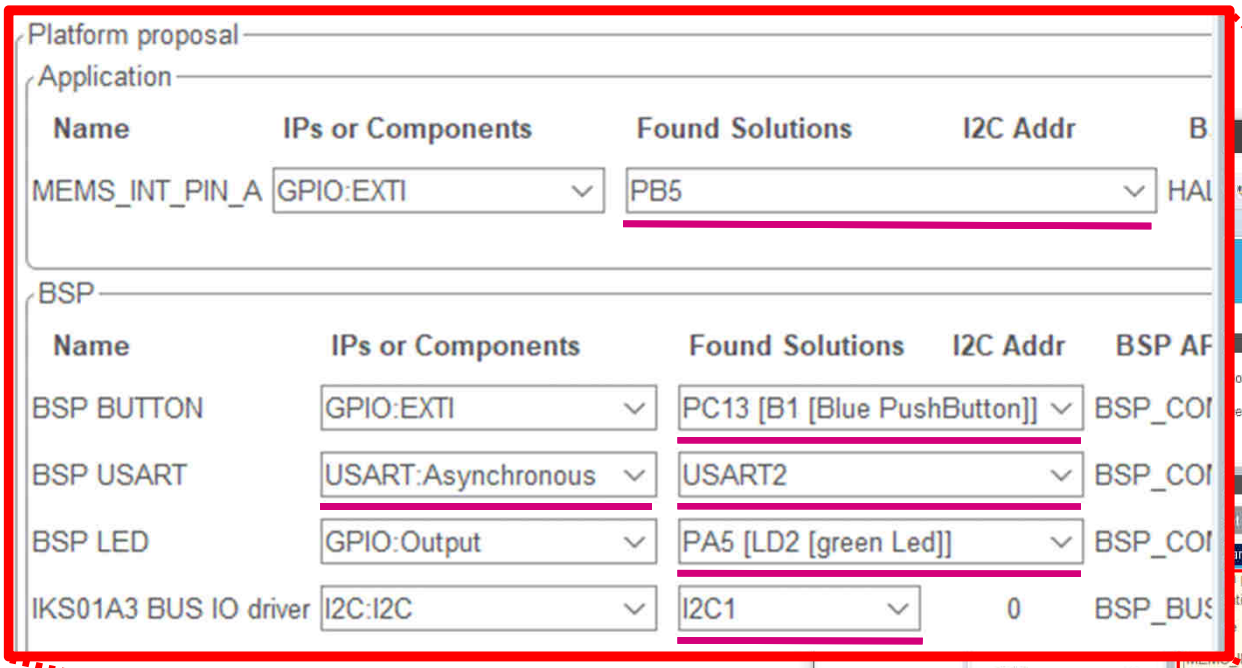
1. Expand Additional Software and select the X-CUBE-MEMS1

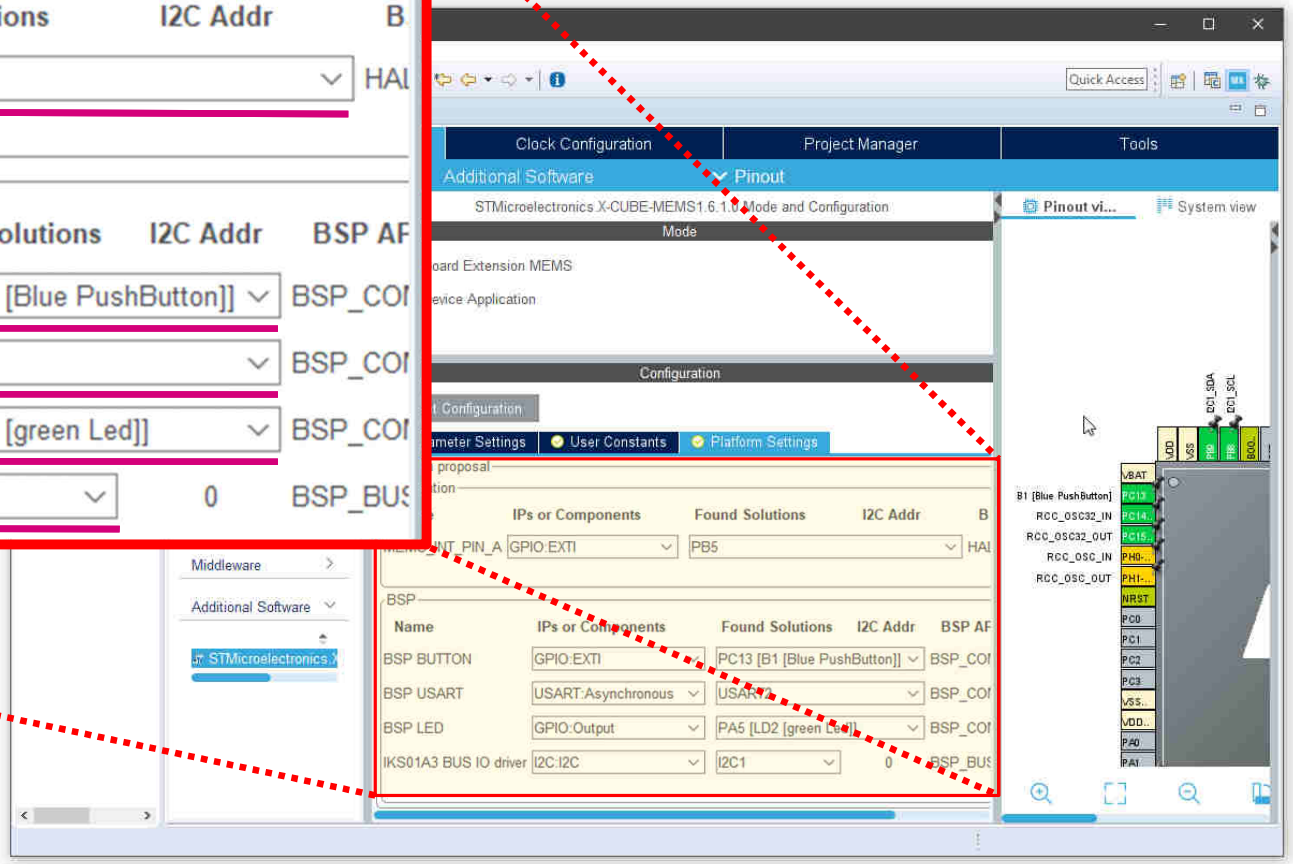
2. Check both:
   Board Extension MEMS
   Device Application

3. Configure Platform Settings as in picture



Lab5 – Configure the MEMS Library



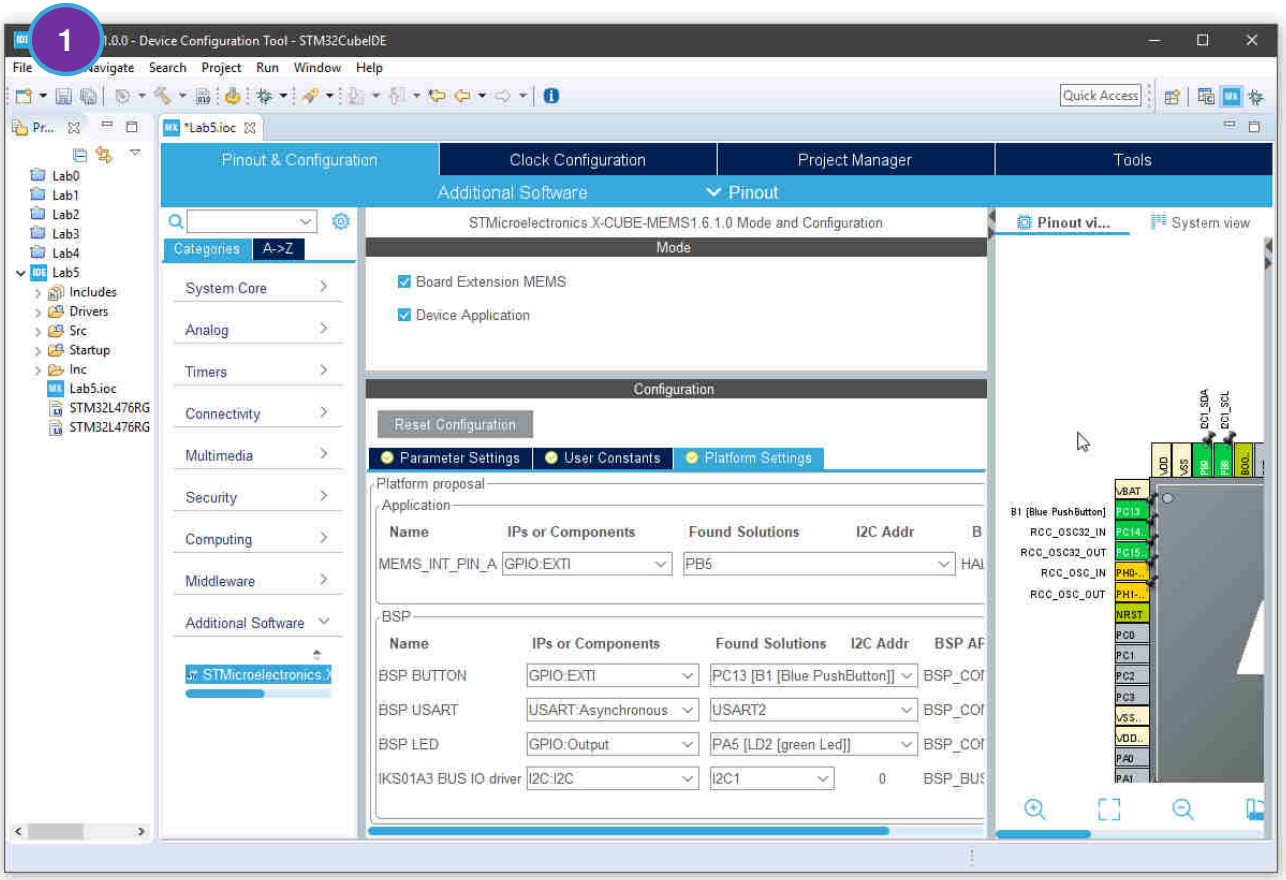


life.augmented

Lab5 – Save the Project

1. Click the save button 

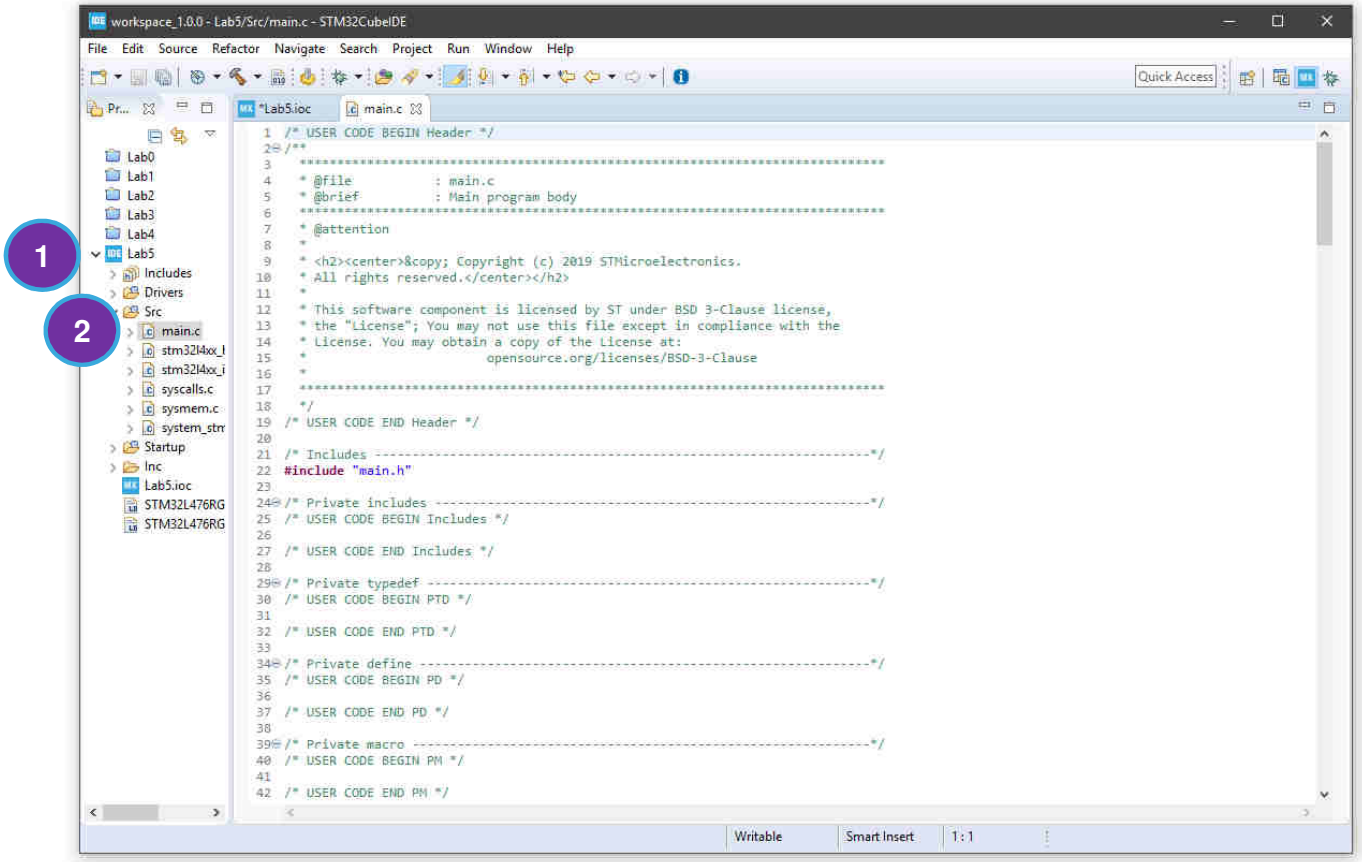
   This action will generate the source code of this lab



life.augmented

Lab5 – Code Editing

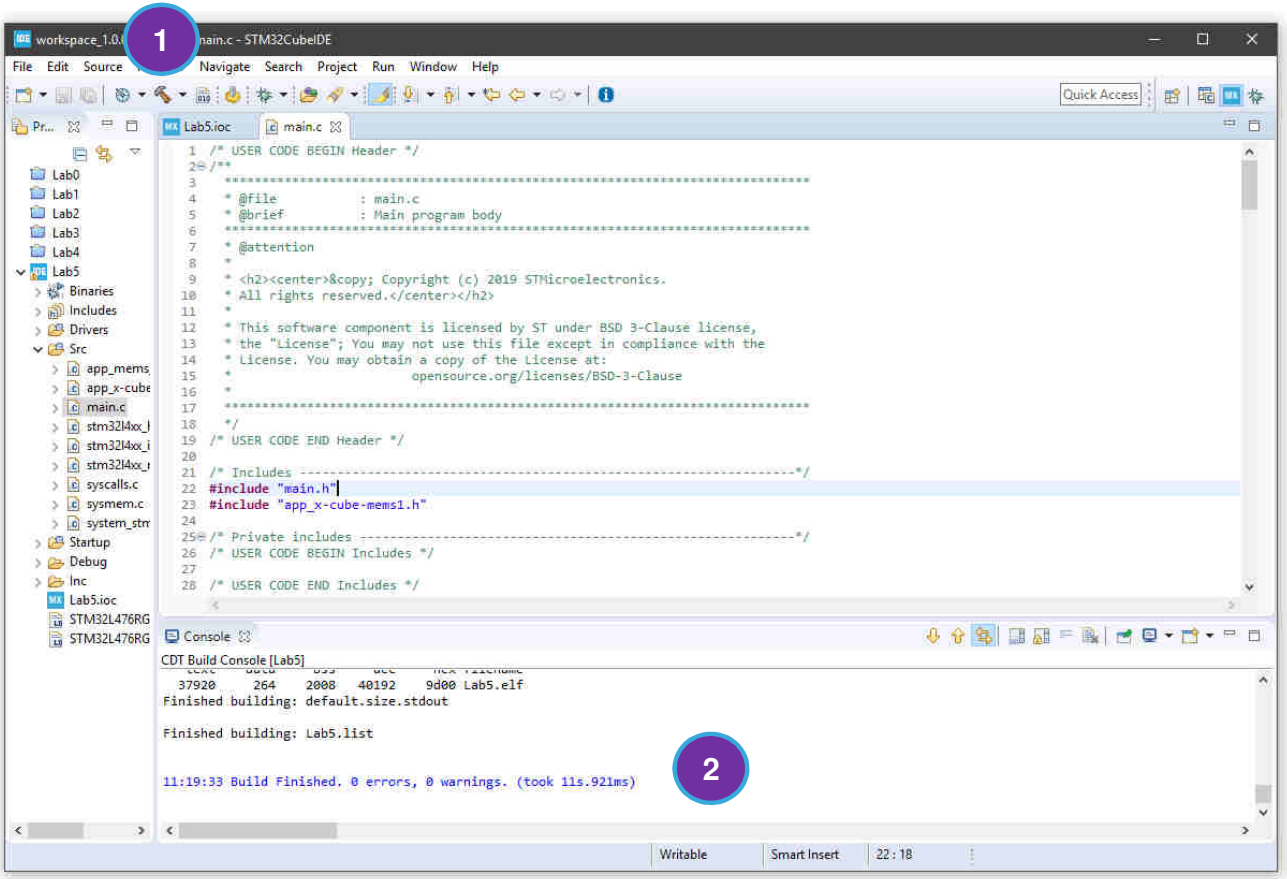
1. Expand **Src** in folder **Lab5**
2. Double click on **main.c**



Lab5 - Compiling

1. Click on the hammer  to begin compilation, or press **CTRL+B**

2. Compilation should terminate with 0 errors and 0 warning



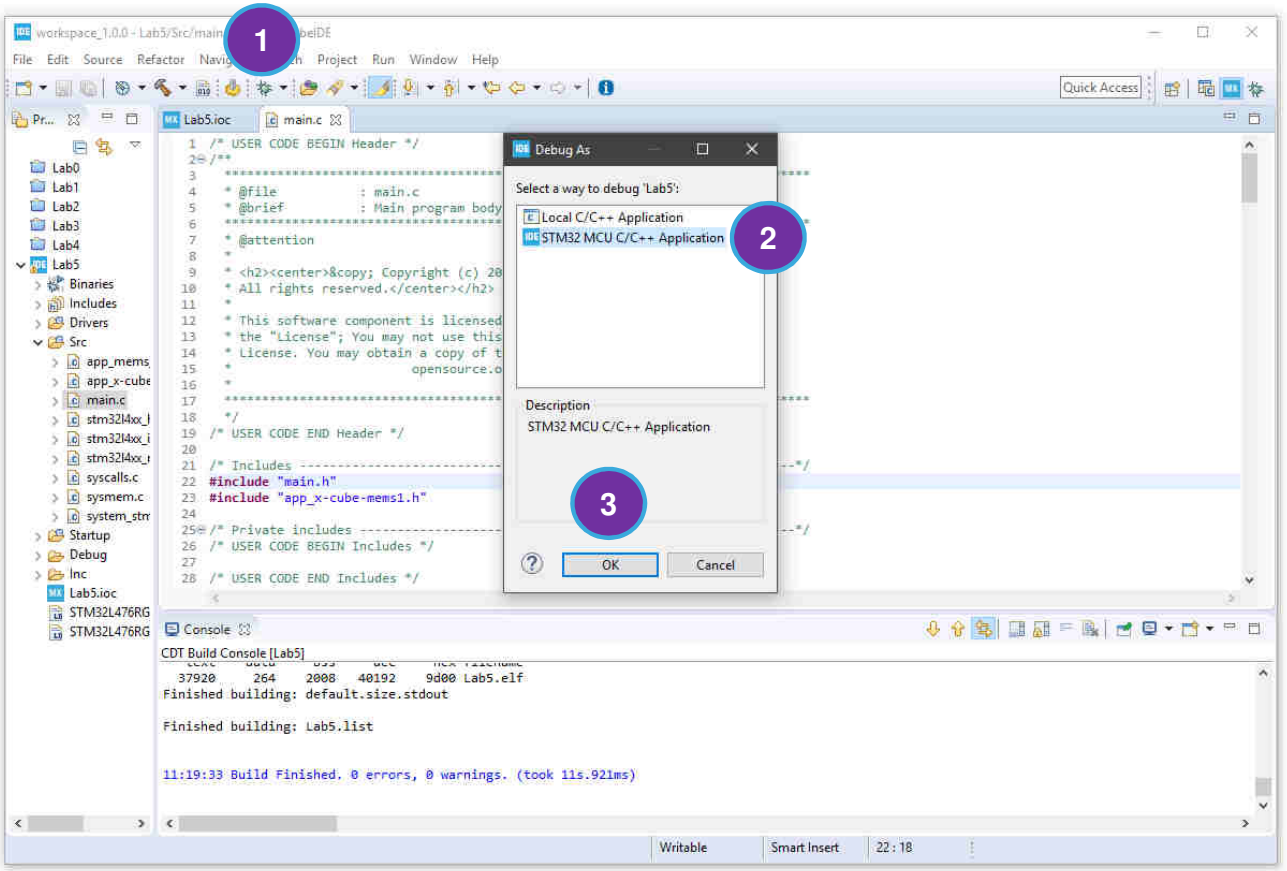
life.augmented

Lab5 - Debugging

1. Click on the bug  to begin debugging

2. Select
   STM32 MCU C/C++ App

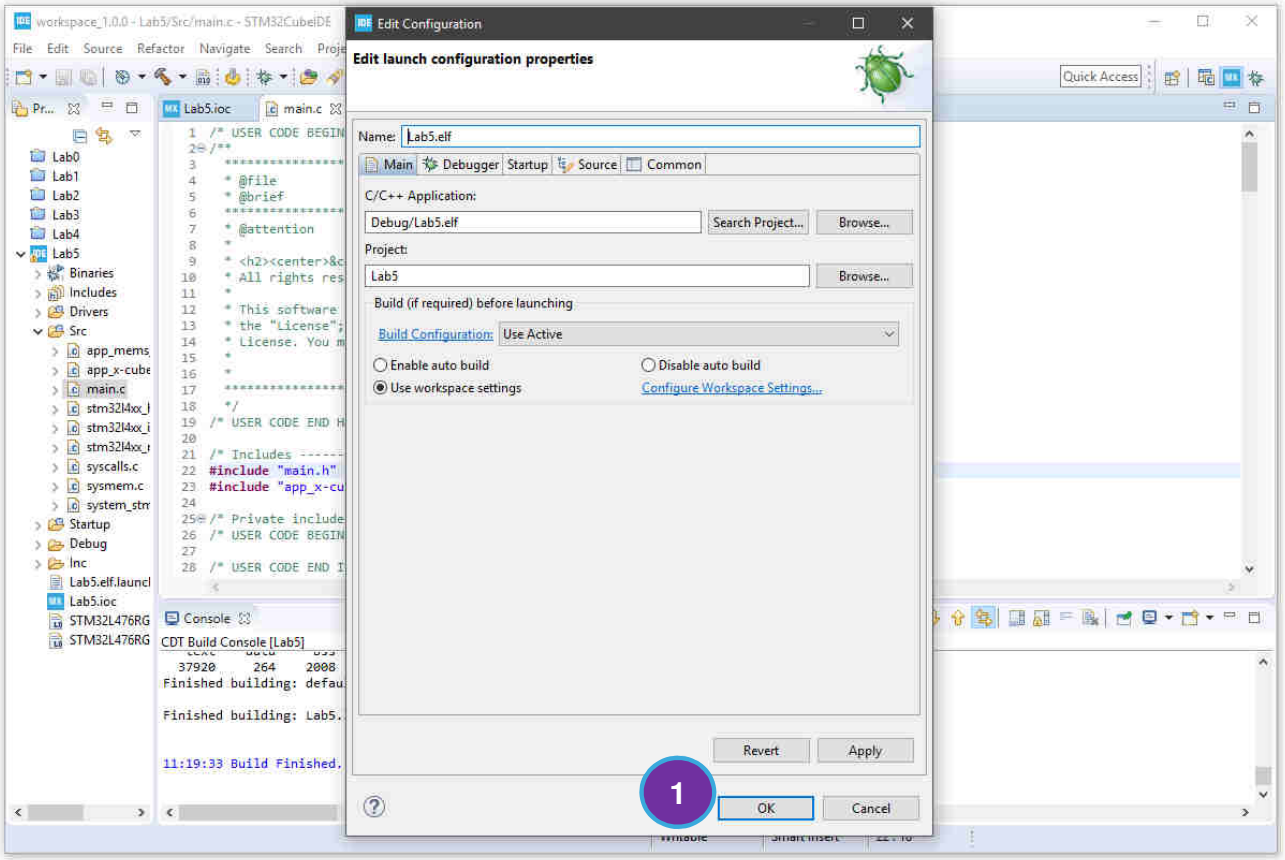
3. Click **OK**



life.augmented

Lab5 - Debugging

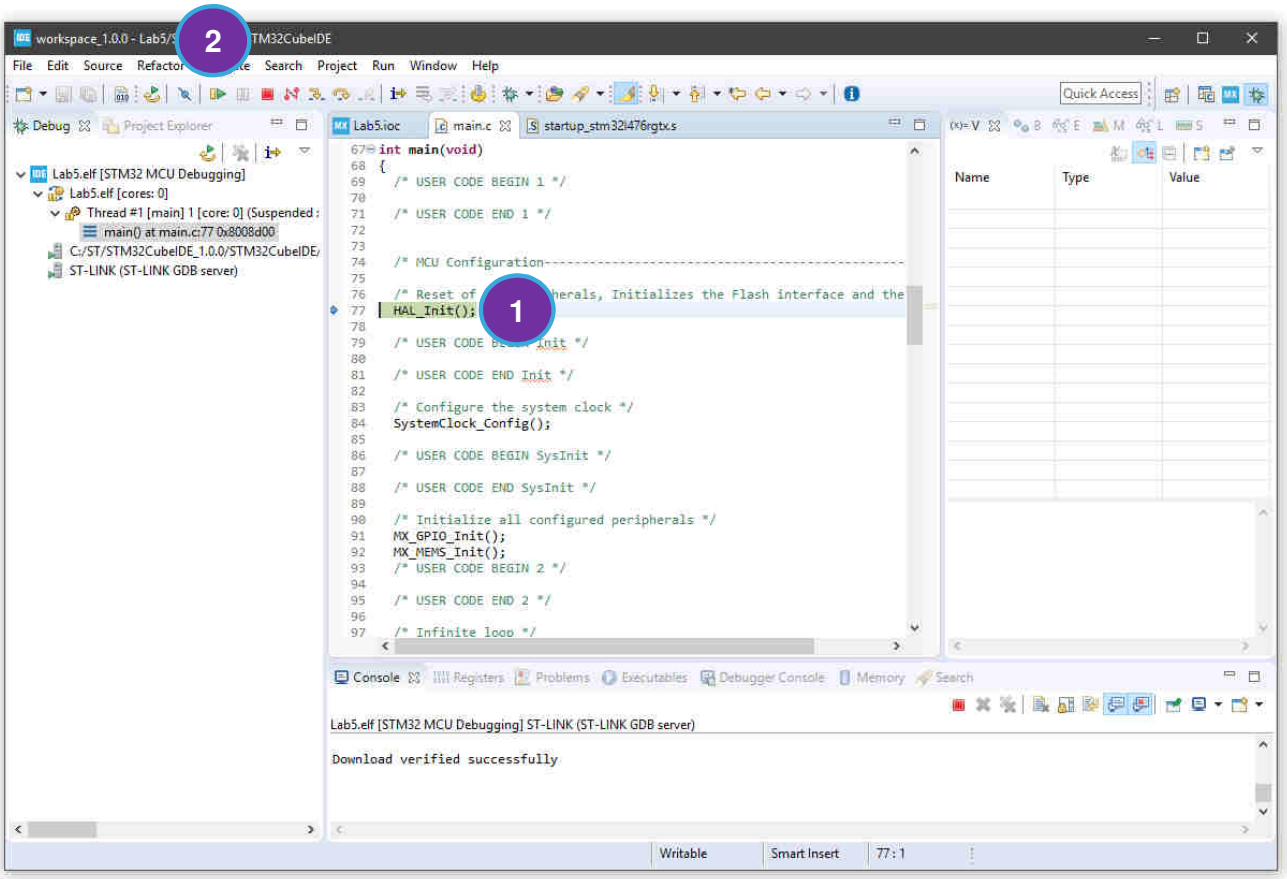
1. Click **OK**



life.augmented

Lab5 - Debugging

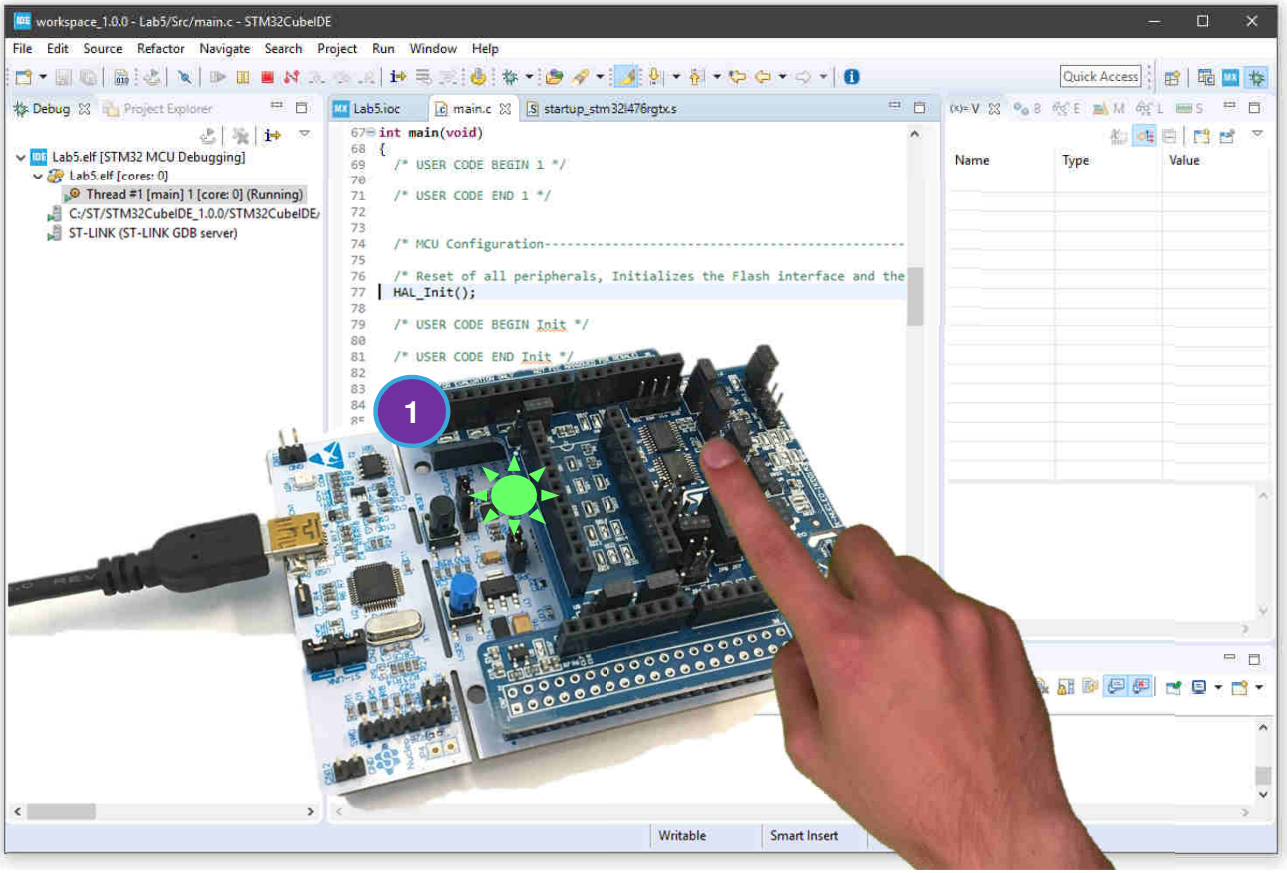
1. Code start at the first line of the main function
2. Click play  button to run the code



life.augmented

Lab5 - Testing

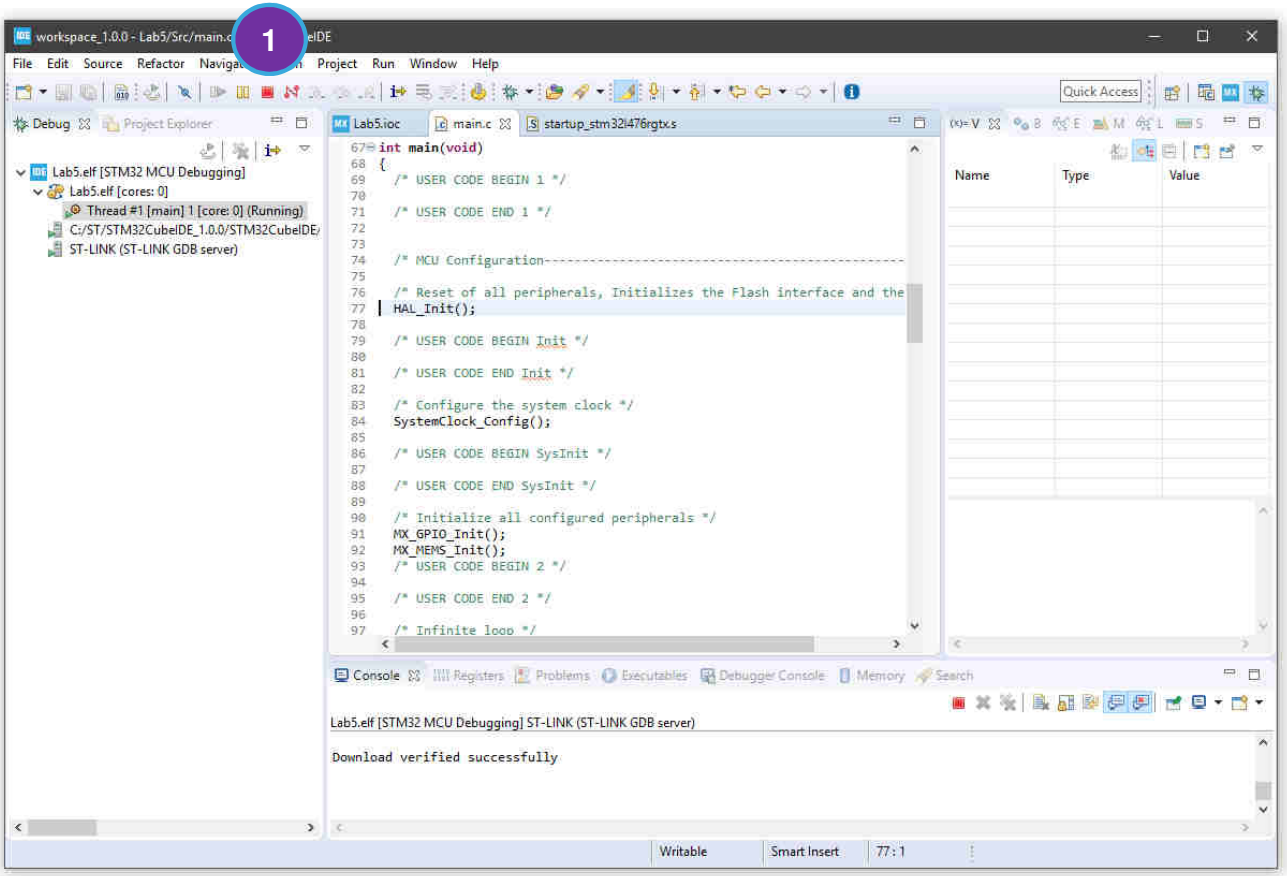
1. **TAP** or **DOUBLE TAP** the board and the GREEN led will turn ON



life.augmented

Lab5 - Debugging

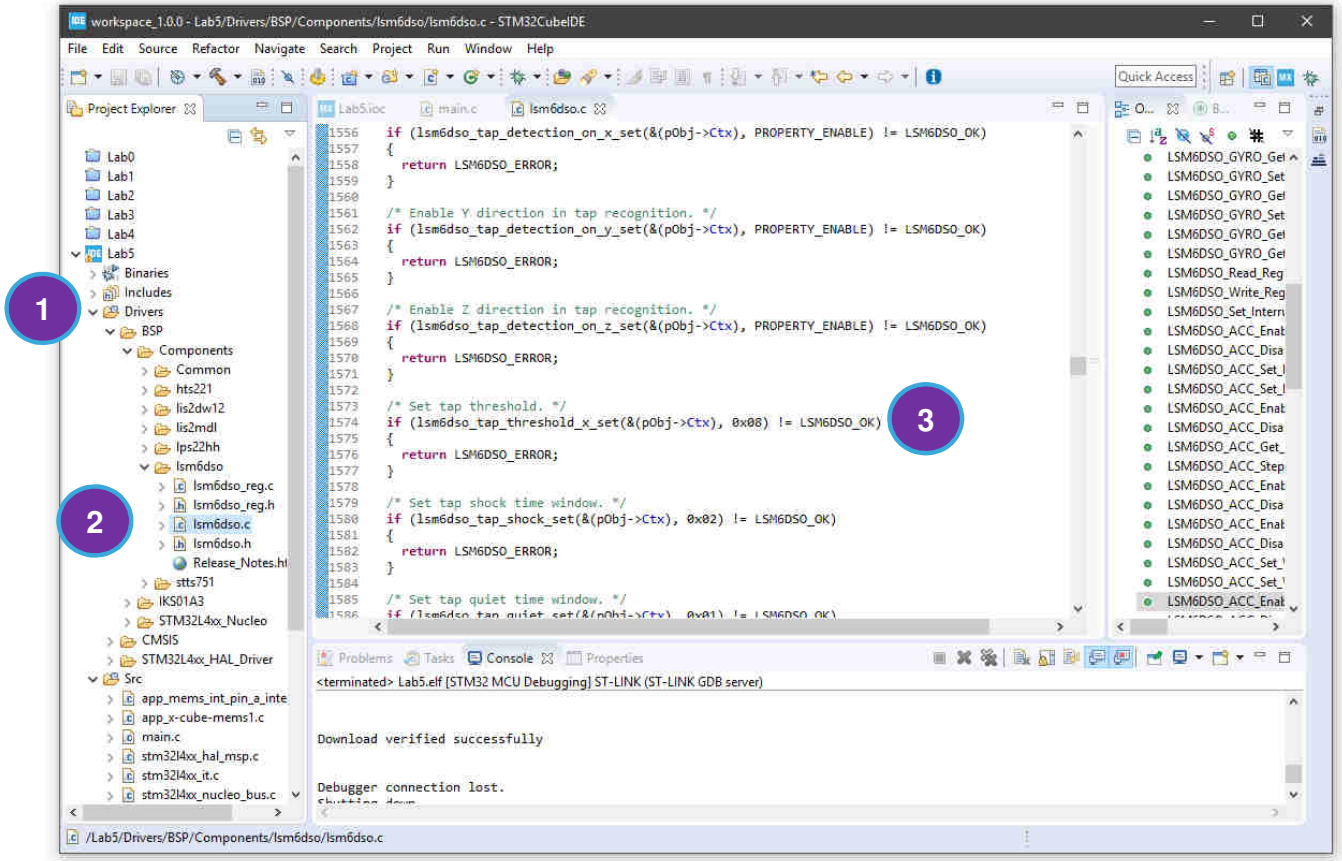
1. Click stop  button to interrupt the debugging



life.augmented

Lab5 – Code Editing

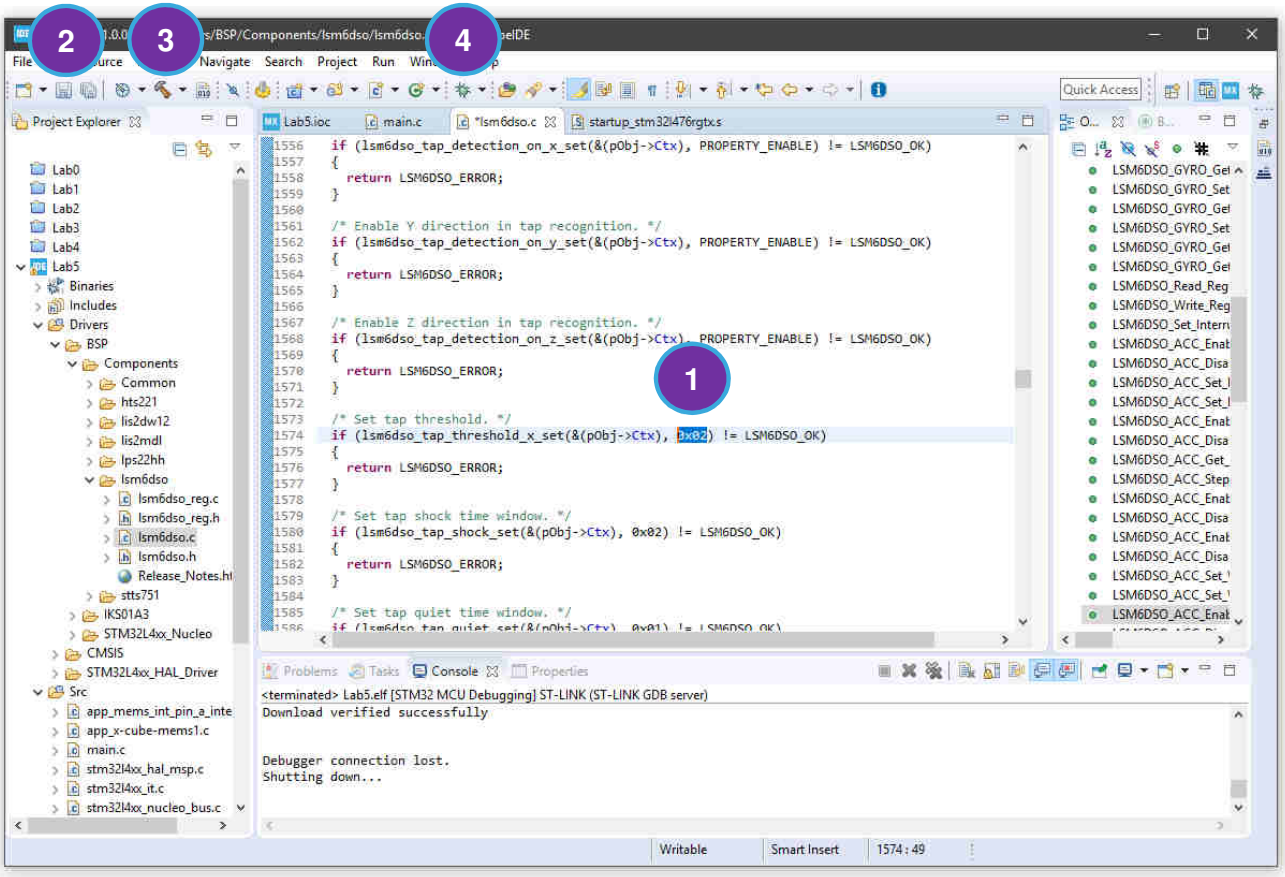
1. Expand folder **Drivers > BSP > Components > lsm6dso**
2. Open file **lsm6dso.c**
3. Go to **line #1574**



life.augmented

Lab5 – Code Editing

1. In **line #1574**, edit the threshold **from value 0x08 to value 0x02***
2. Save modification 
3. Recompile 
4. Launch debug 

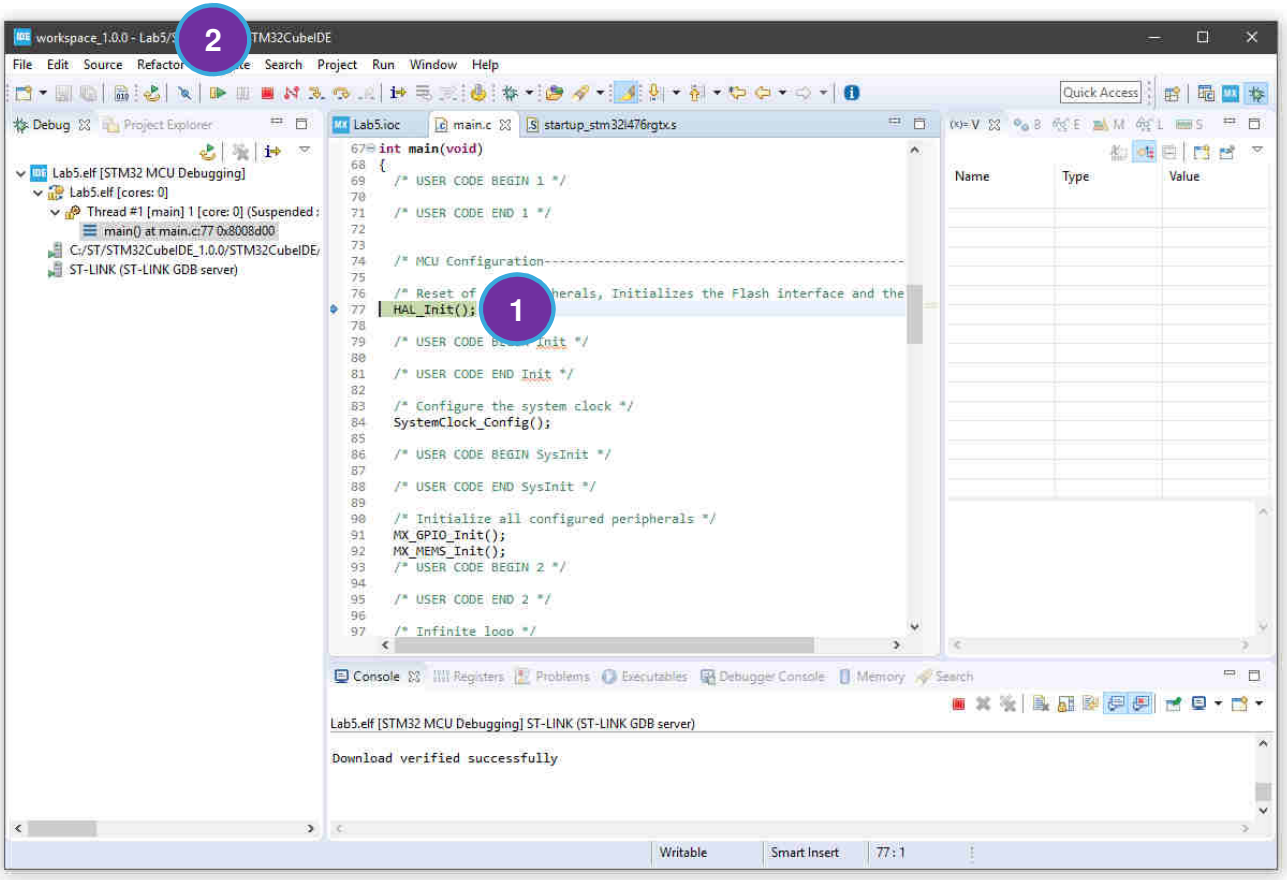


* This modification will reduce the threshold of the tap detection, increasing sensitivity of the recognized tap. For further details please refer to **Application Note AN5192 Section 5.5.1**

life.augmented

Lab5 - Debugging

1. Code start at the first line of the main function
2. Click play  button to run the code

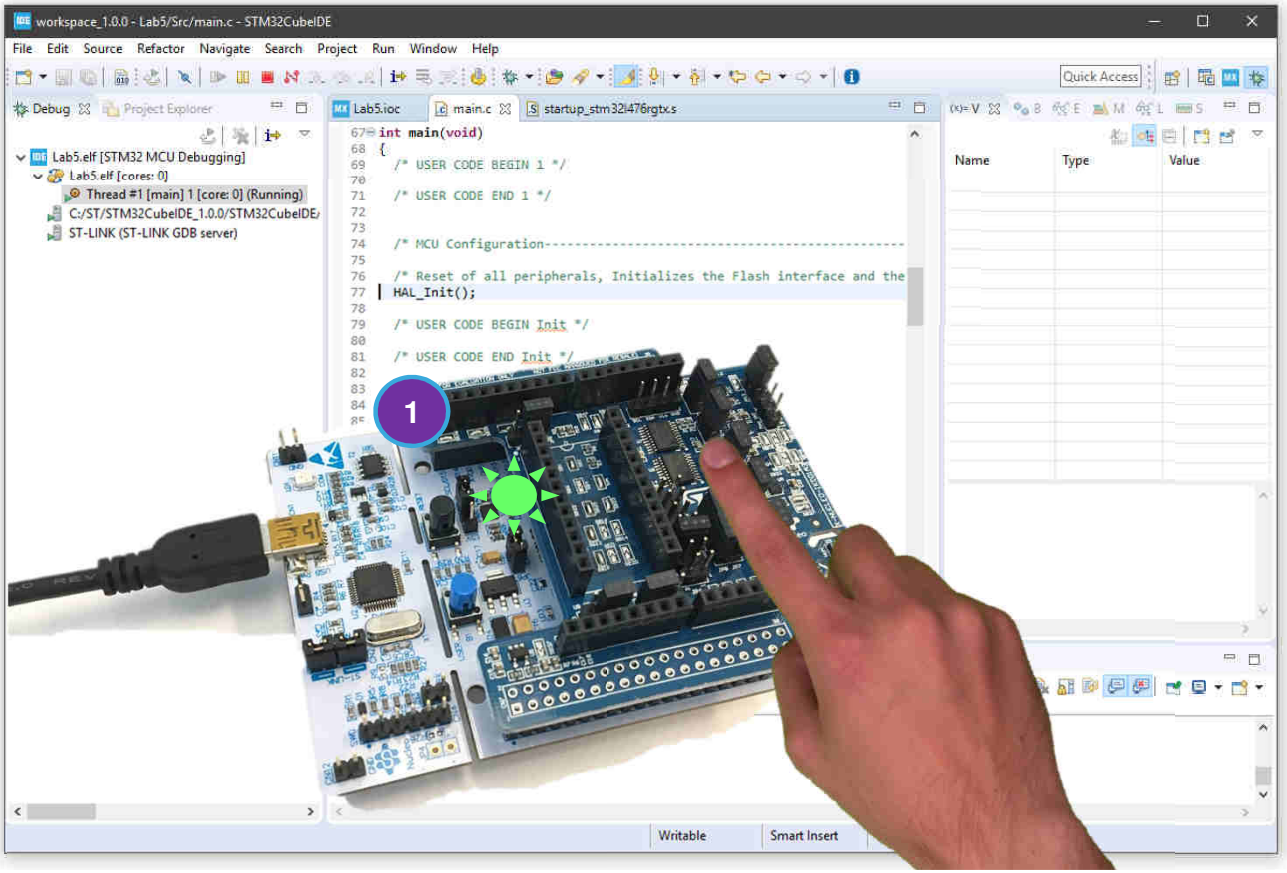


life.augmented

Lab5 - Testing

1. **TAP** or **DOUBLE TAP** the board and the GREEN led will turn ON

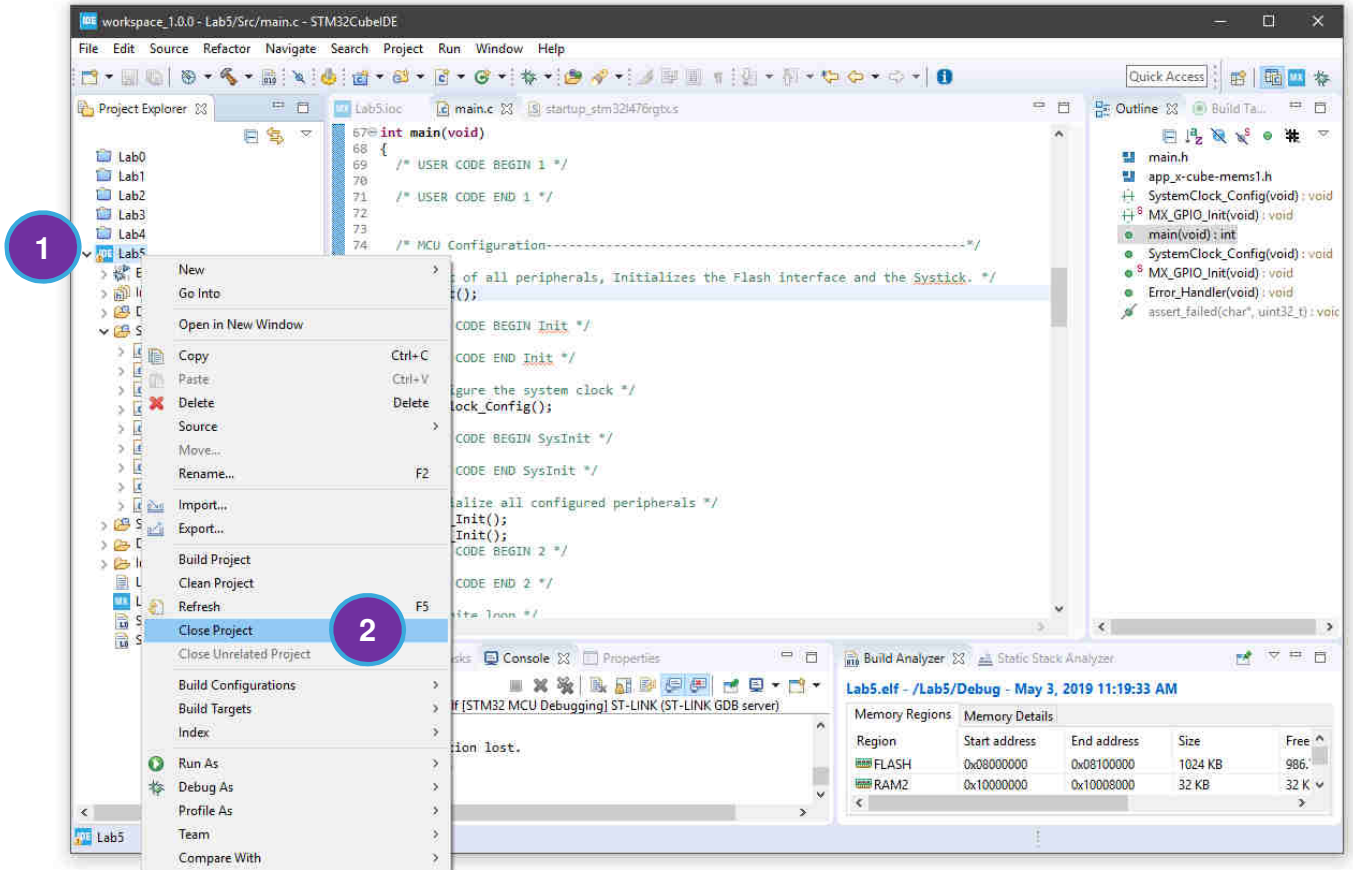
You may notice that the sensitivity of the TAP recognition has been increased



life.augmented

Lab5 – Closing the Project

1. Right-Click on **Lab5** project
2. Click on **Close Project**



New: In this LAB the “X-NUCLEO-IKS0A3” will be used as a “Custom board”

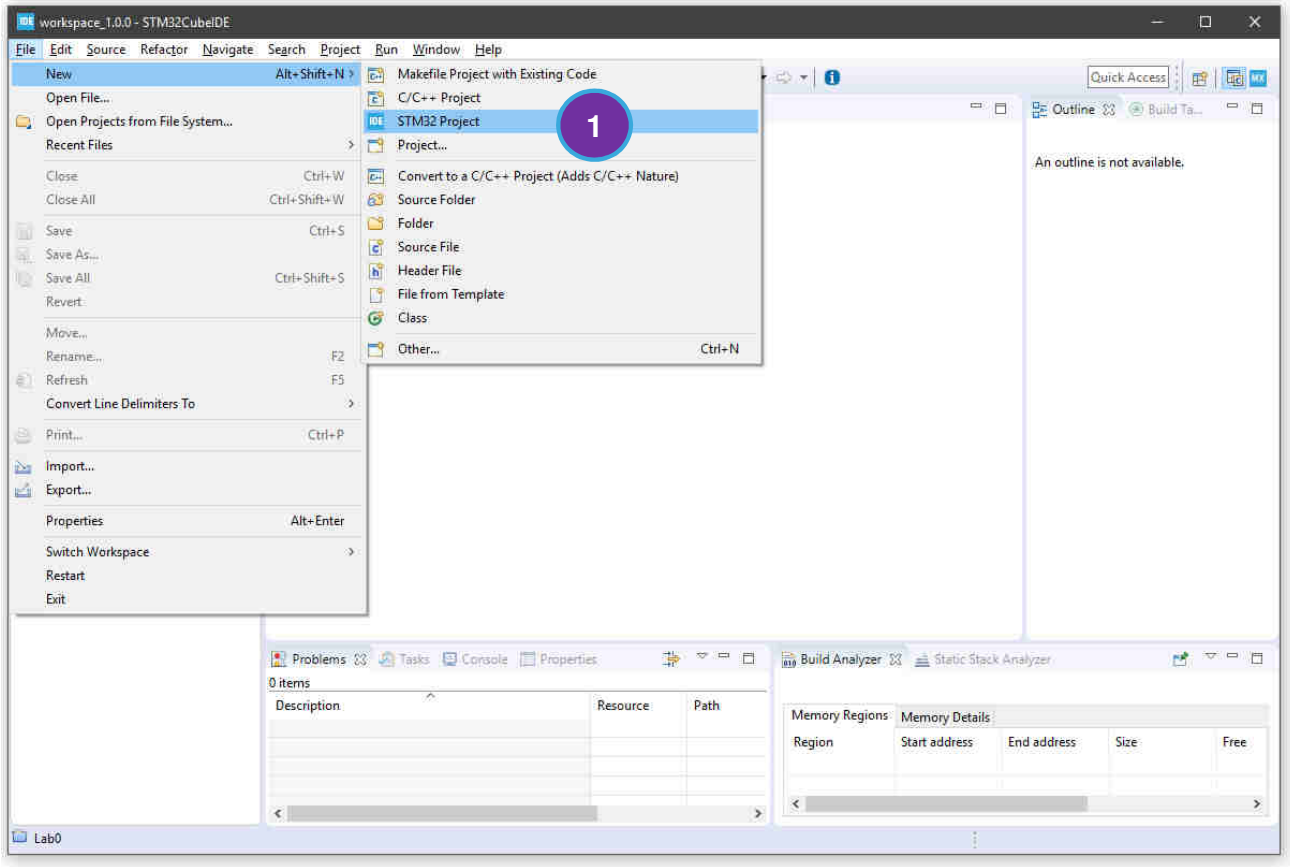
LAB6

Goals:

- Configure a new project using X-CUBE-MEMS1
- Configure LSM6DSO only for USB data logging
- Change Output Data Rate to log accelerometer and gyroscope at higher acquisition speed

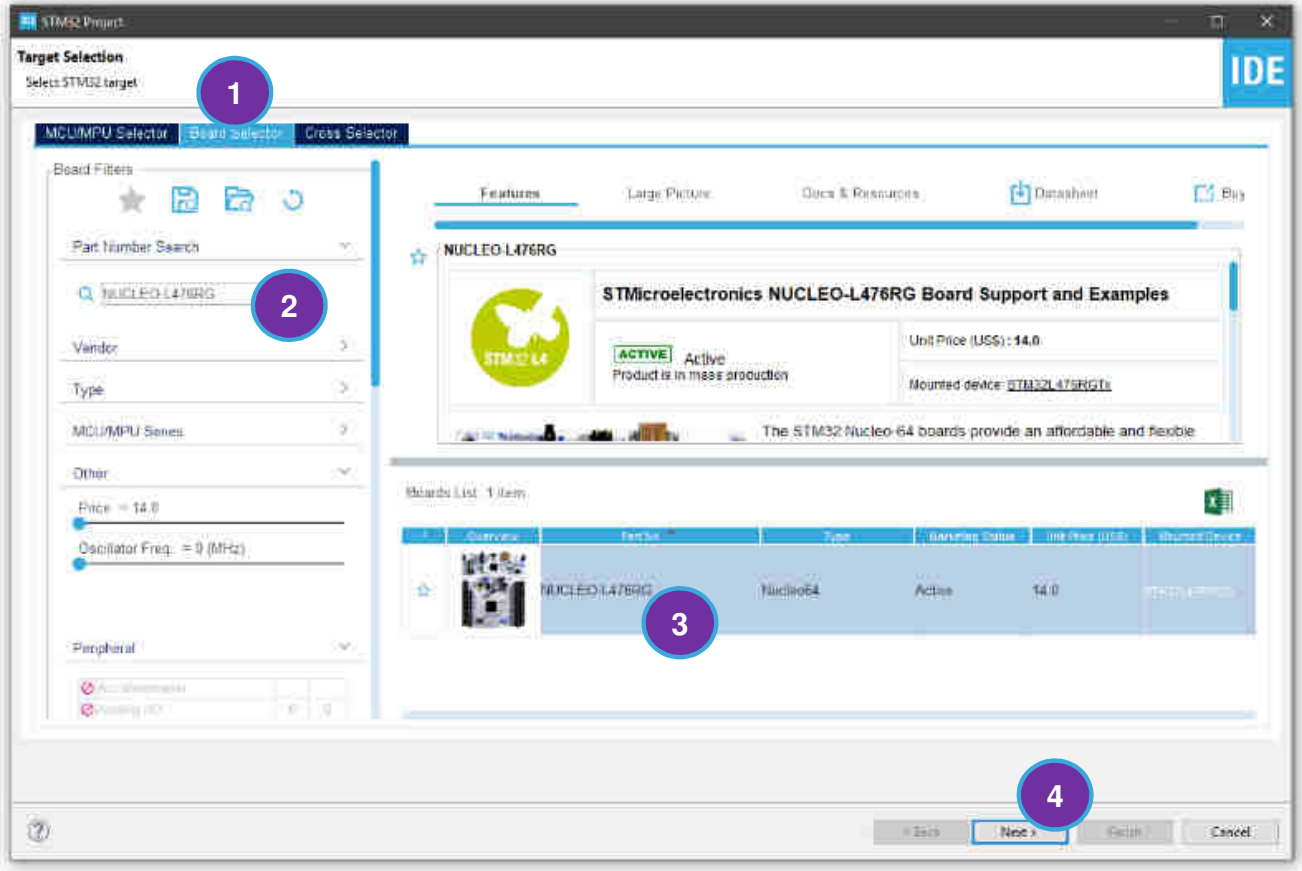
life.augmented

1. Click on **File > New > STM32 Project**

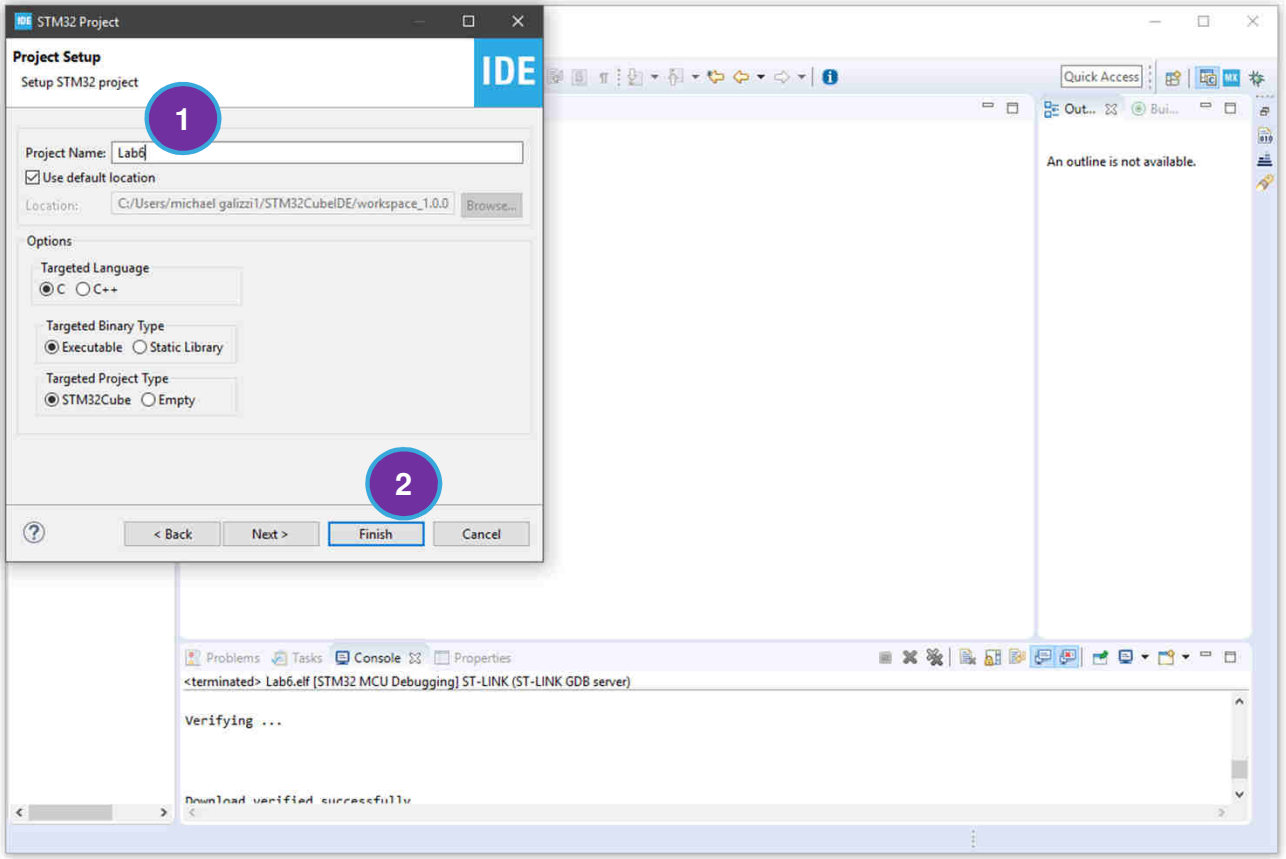


life.augmented

1. Click on **Board Selector**
2. Type **NUCLEO-L476RG**
3. Click on the board
4. Click **Next >**

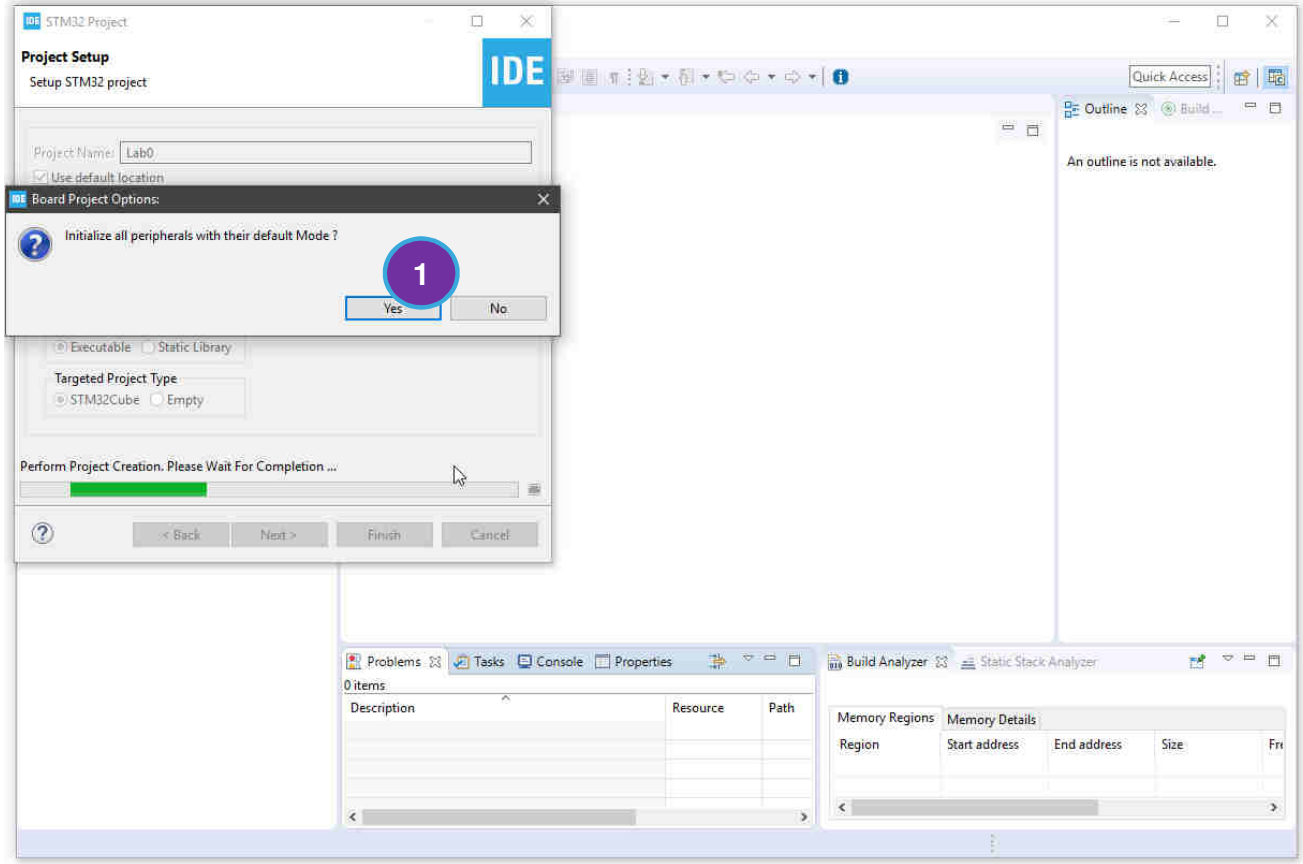


1. Project Name **Lab6**
2. Click **Finish**



life.augmented

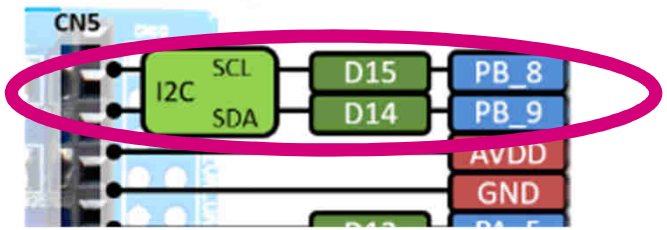
1. Click **Yes** to init peripherals in default mode

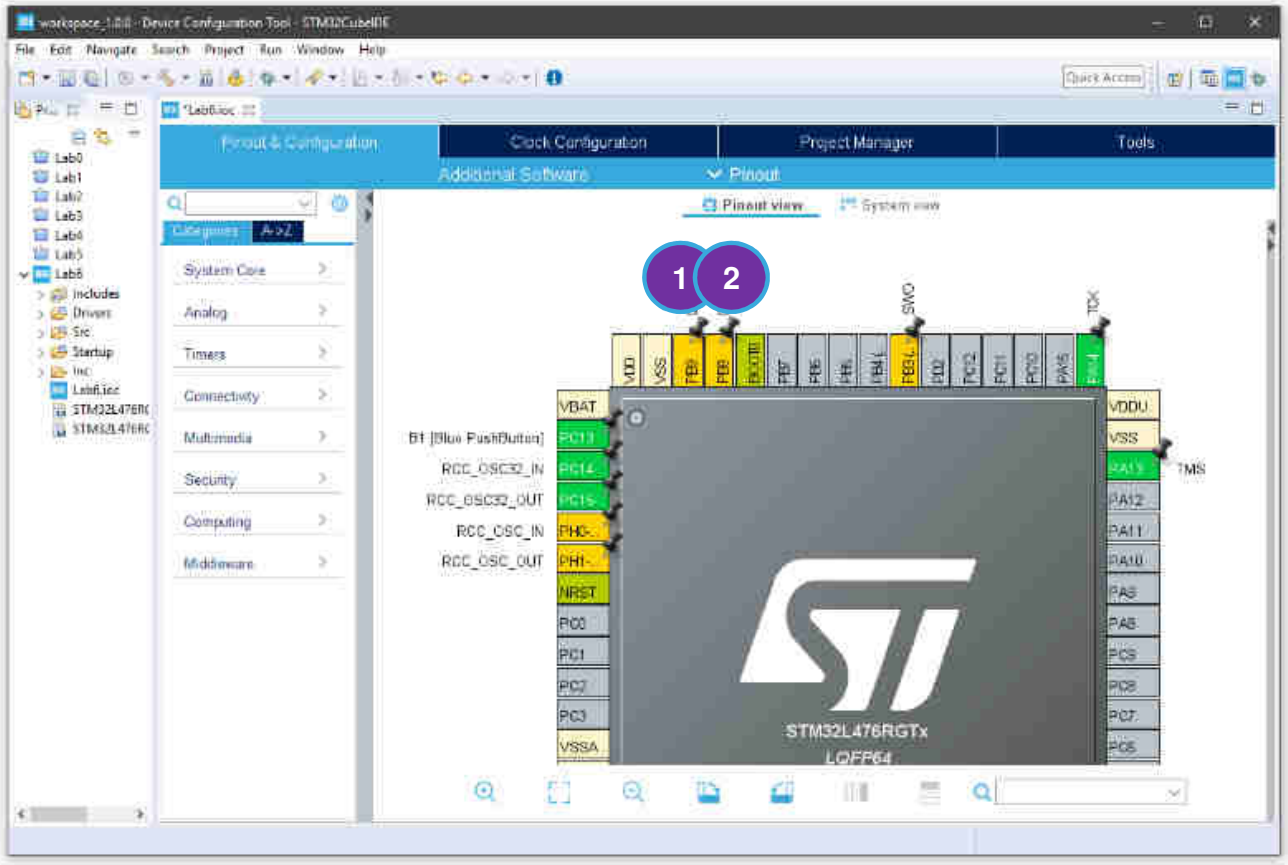


life.augmented

Lab6 – Configure the I2C Bus

1. Left Click on **PB9** and select I2C1_SDA
2. Left Click on **PB8** and select I2C1_SCL





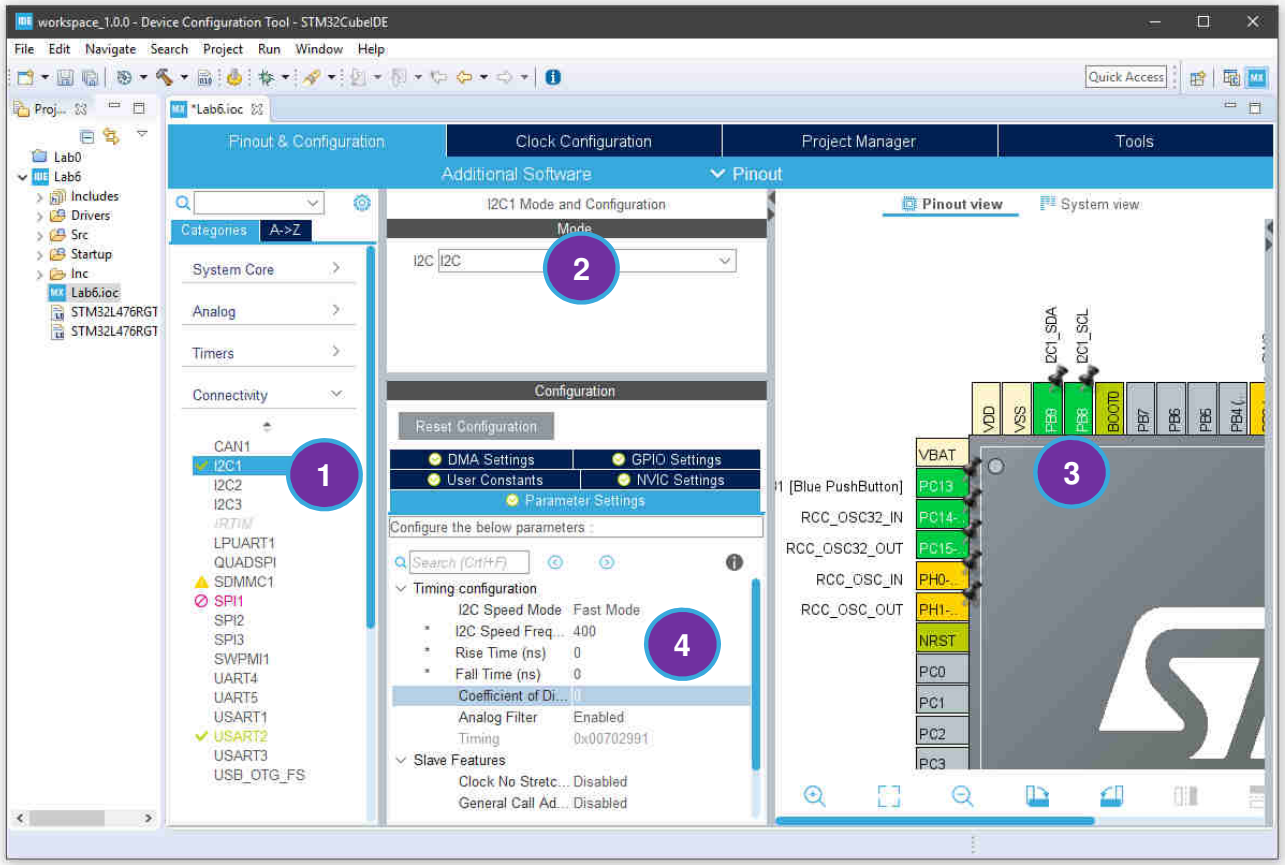
life.augmented

Lab6 – Configure the I2C Bus

1. Expand *Connectivity* tab and check **I2C1**
2. Select **I2C** in *I2C1 Mode and Configuration*
3. PB8 and PB9 should now become green

new

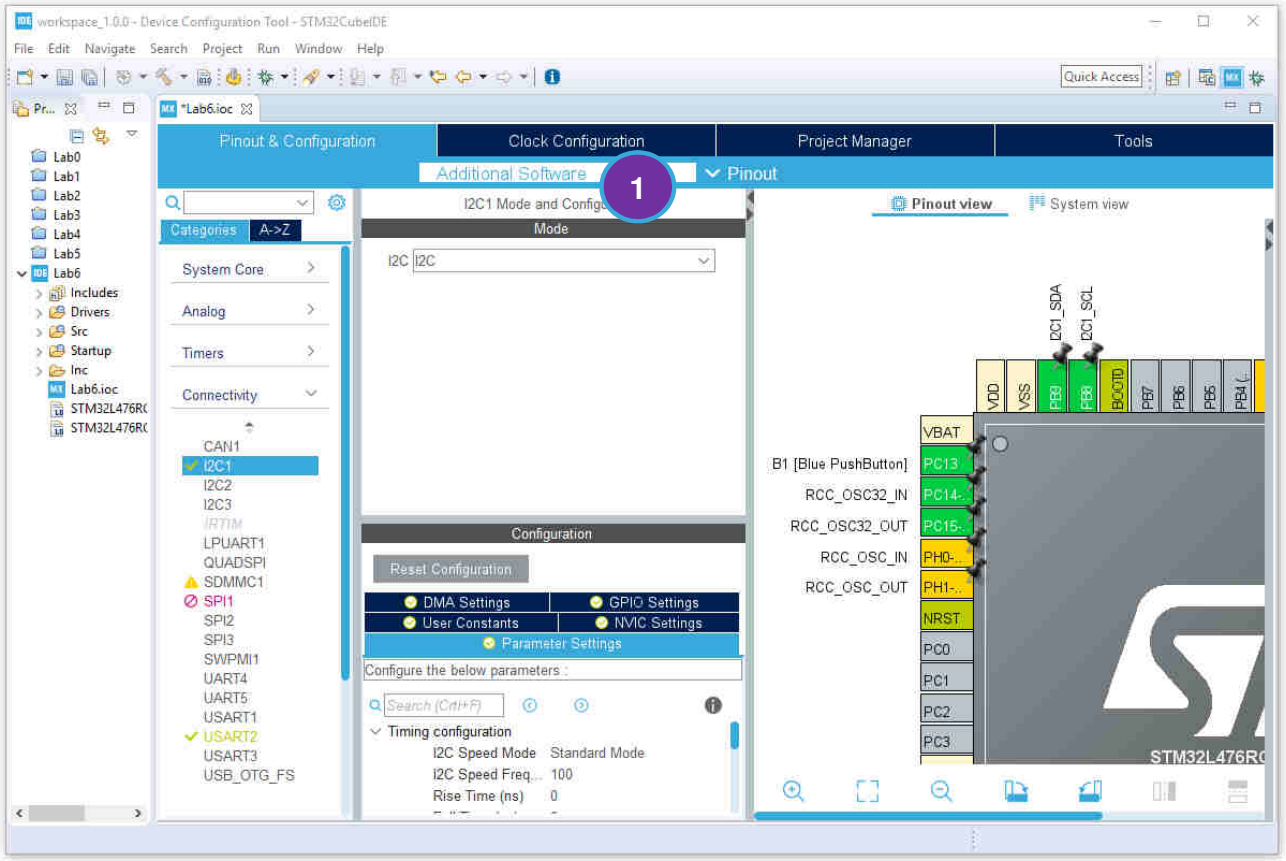
4. Setup **Fast Mode** in **Parameter Settings**



life.augmented

Lab6 – Select the MEMS Library

1. Click on **Additional Software**

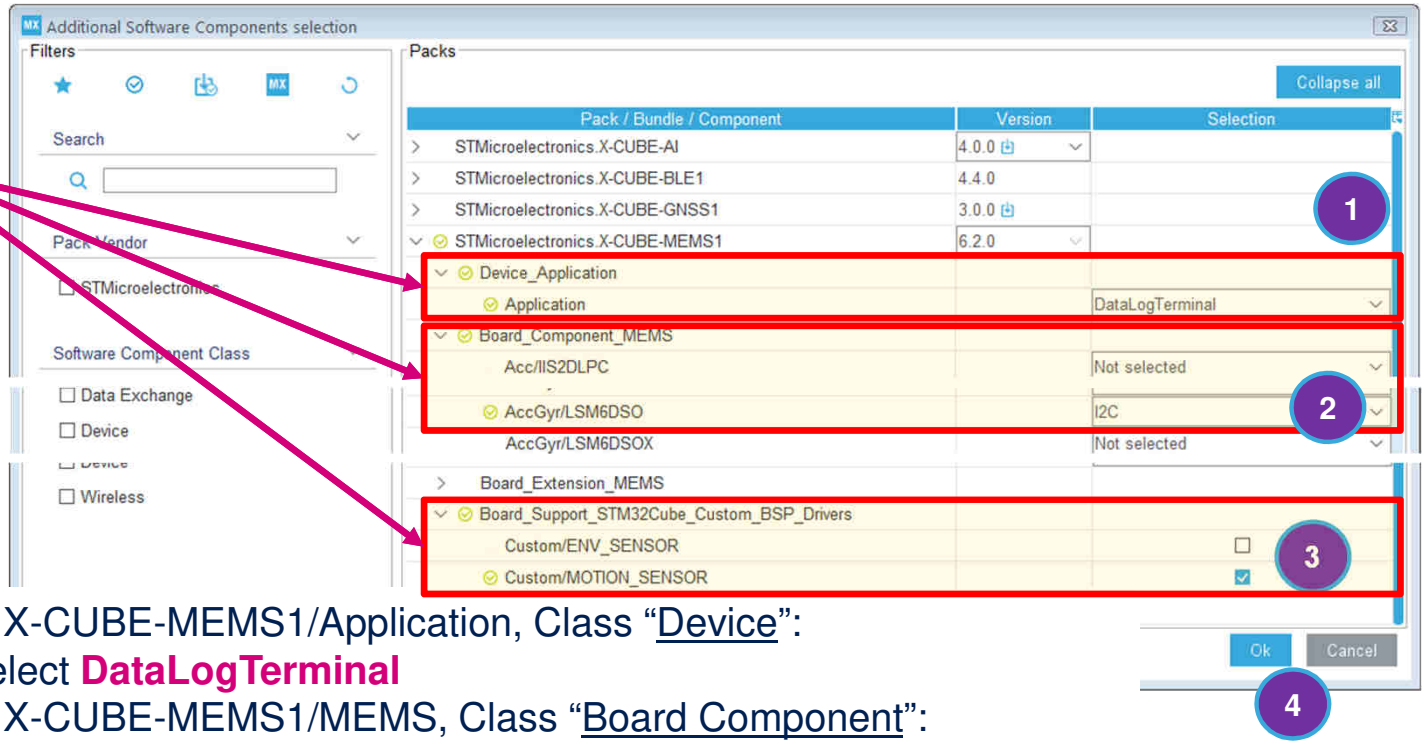


life.augmented

Lab6 – Select the MEMS Library

3 selections To be done!

Click to expand tree

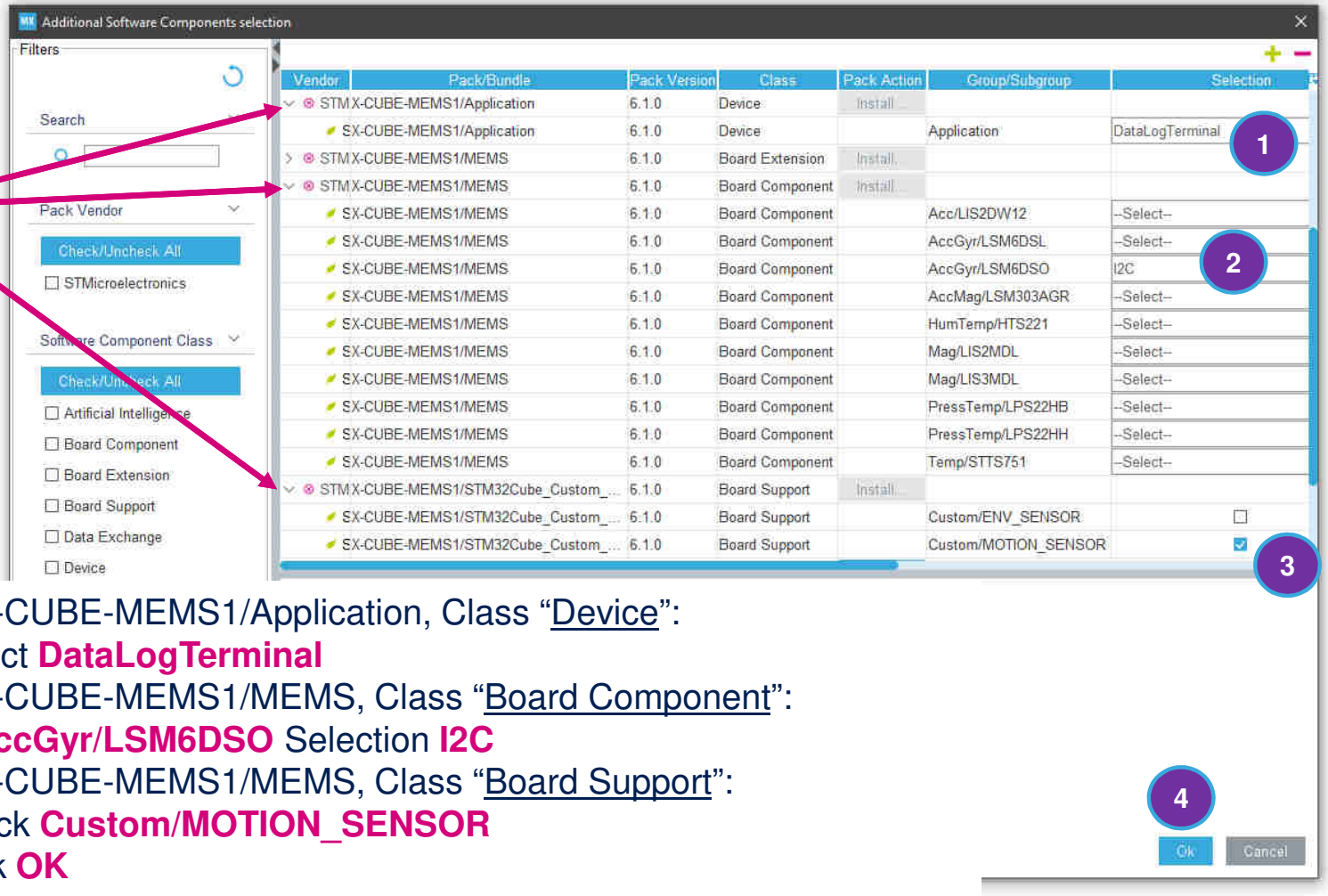


1. In X-CUBE-MEMS1/Application, Class "Device":
   Select **DataLogTerminal**
2. In X-CUBE-MEMS1/MEMS, Class "Board Component":
   In **AccGyr/LSM6DSO** Selection **I2C**
3. In X-CUBE-MEMS1/MEMS, Class "Board Support":
   Check **Custom/MOTION_SENSOR**
4. Click **OK**

life.augmented

Lab6 – Select the MEMS Library

Click to expand tree



1. In X-CUBE-MEMS1/Application, Class “Device”:
   Select **DataLogTerminal**
2. In X-CUBE-MEMS1/MEMS, Class “Board Component”:
   In **AccGyr/LSM6DSO** Selection **I2C**
3. In X-CUBE-MEMS1/MEMS, Class “Board Support”:
   Check **Custom/MOTION_SENSOR**
4. Click **OK**

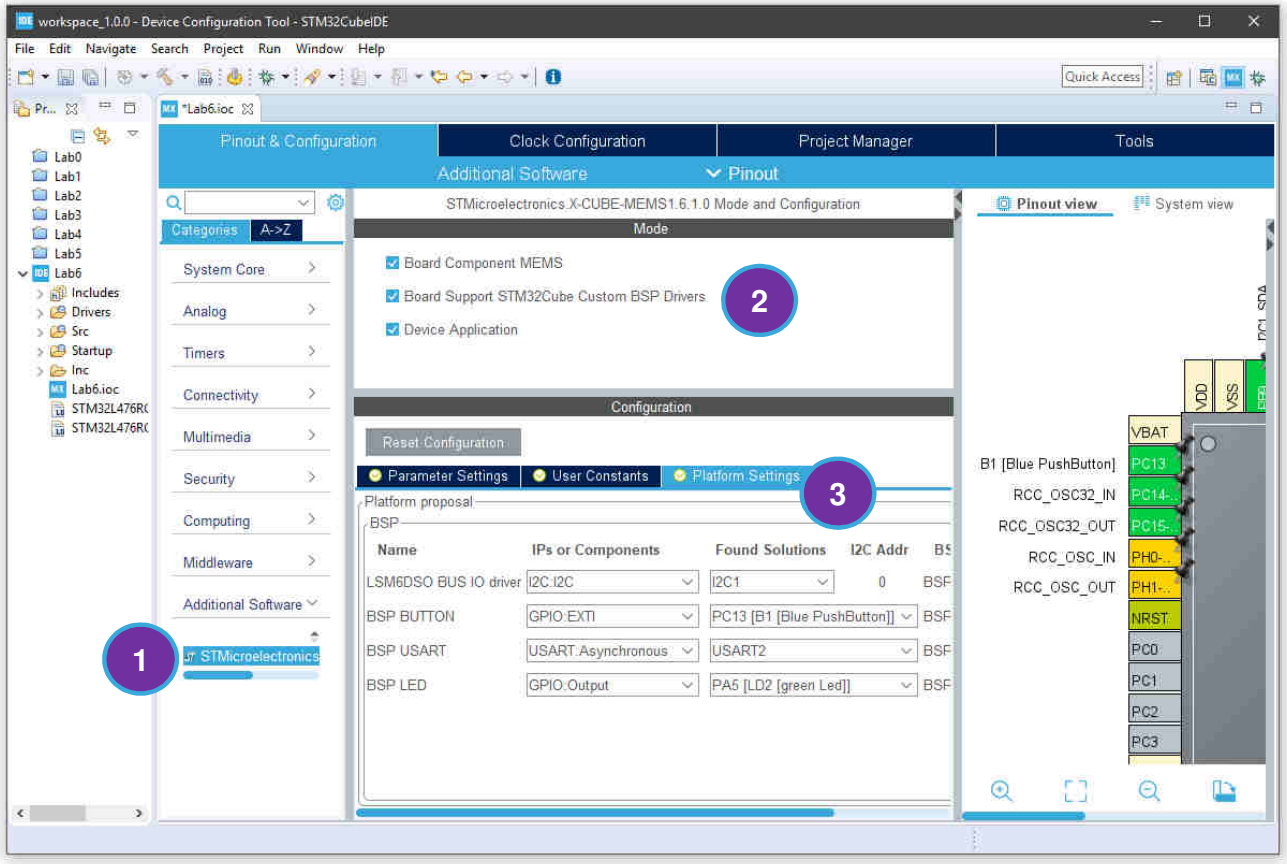
life.augmented

Lab6 – Configure the MEMS Library

1. Expand Additional Software and select the X-CUBE-MEMS1

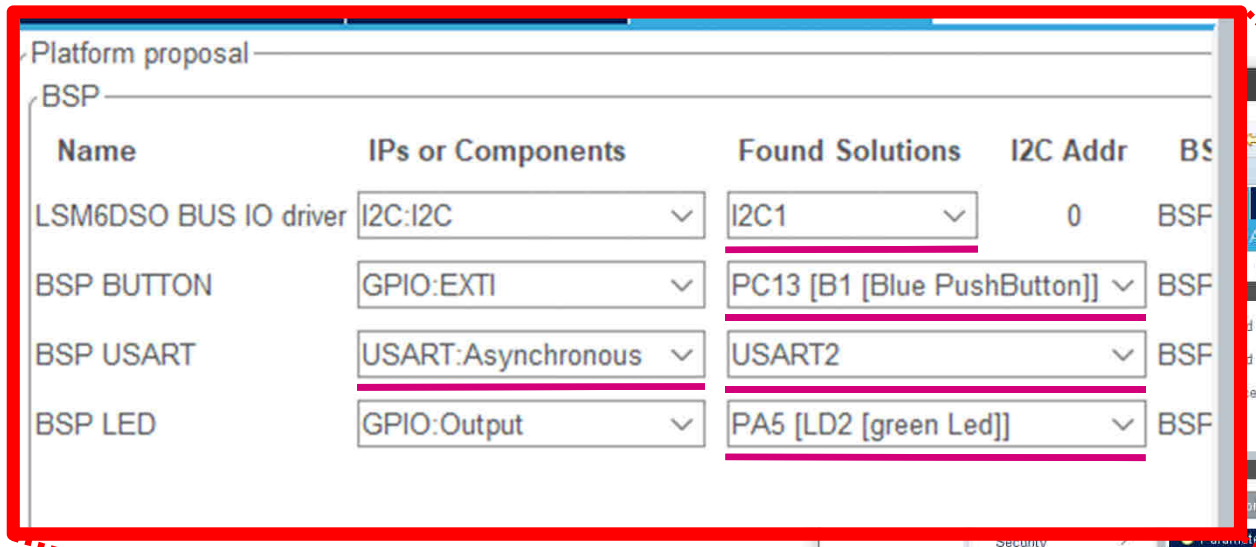
2. Check both:
   Board Component MEMS
   Board Support STM32…
   Device Application

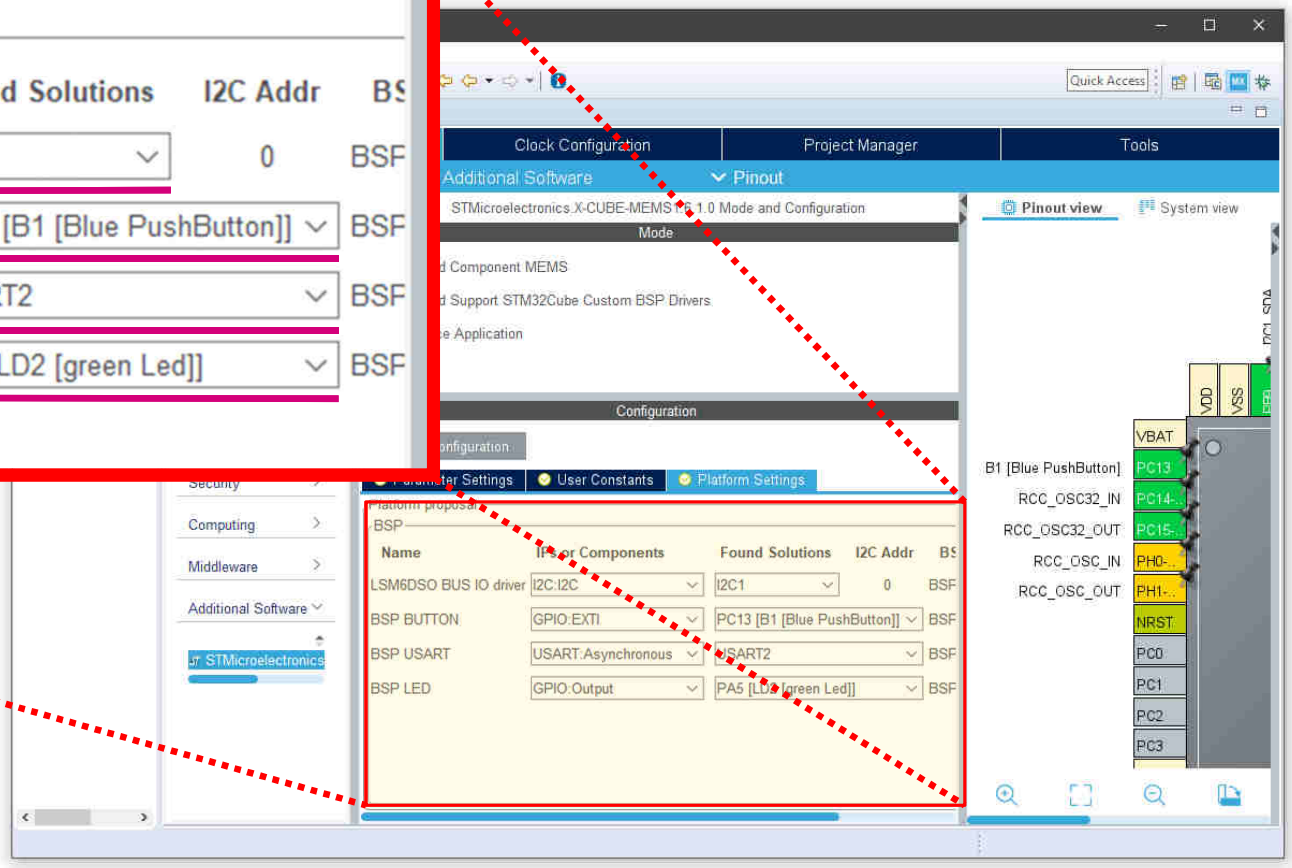
3. Configure ***Platform Settings*** as in picture



life.augmented

Lab6 – Configure the MEMS Library



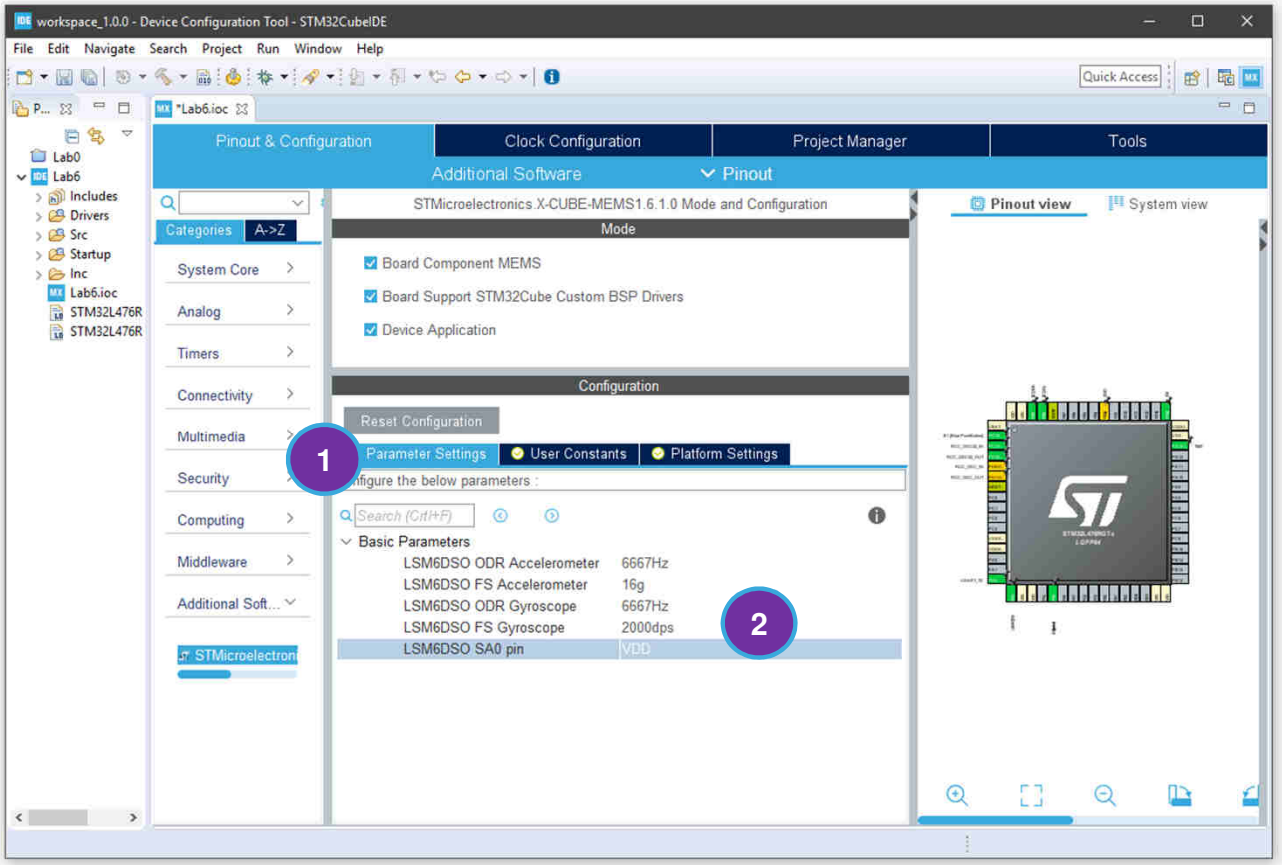


life.augmented

Lab6 – Configure the MEMS Library

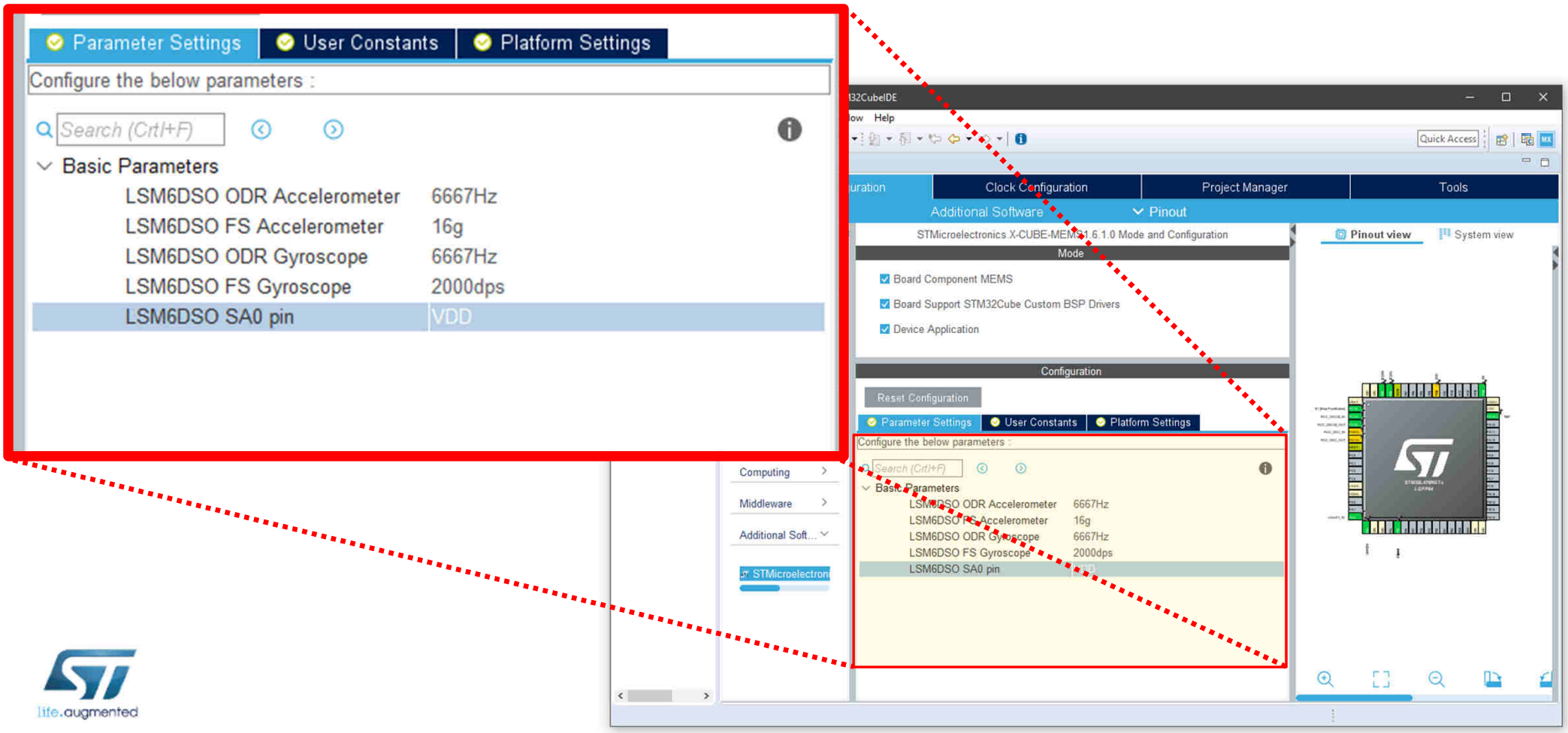
1. Configure ***Parameter Settings*** as in picture

2. ODR_Accelerometer: 6667Hz
   FS_Accelerometer: 16g
   ODR_Gyroscope: 6667Hz
   FS_Gyroscope: 2000dps



life.augmented

Lab6 – Configure the MEMS Library

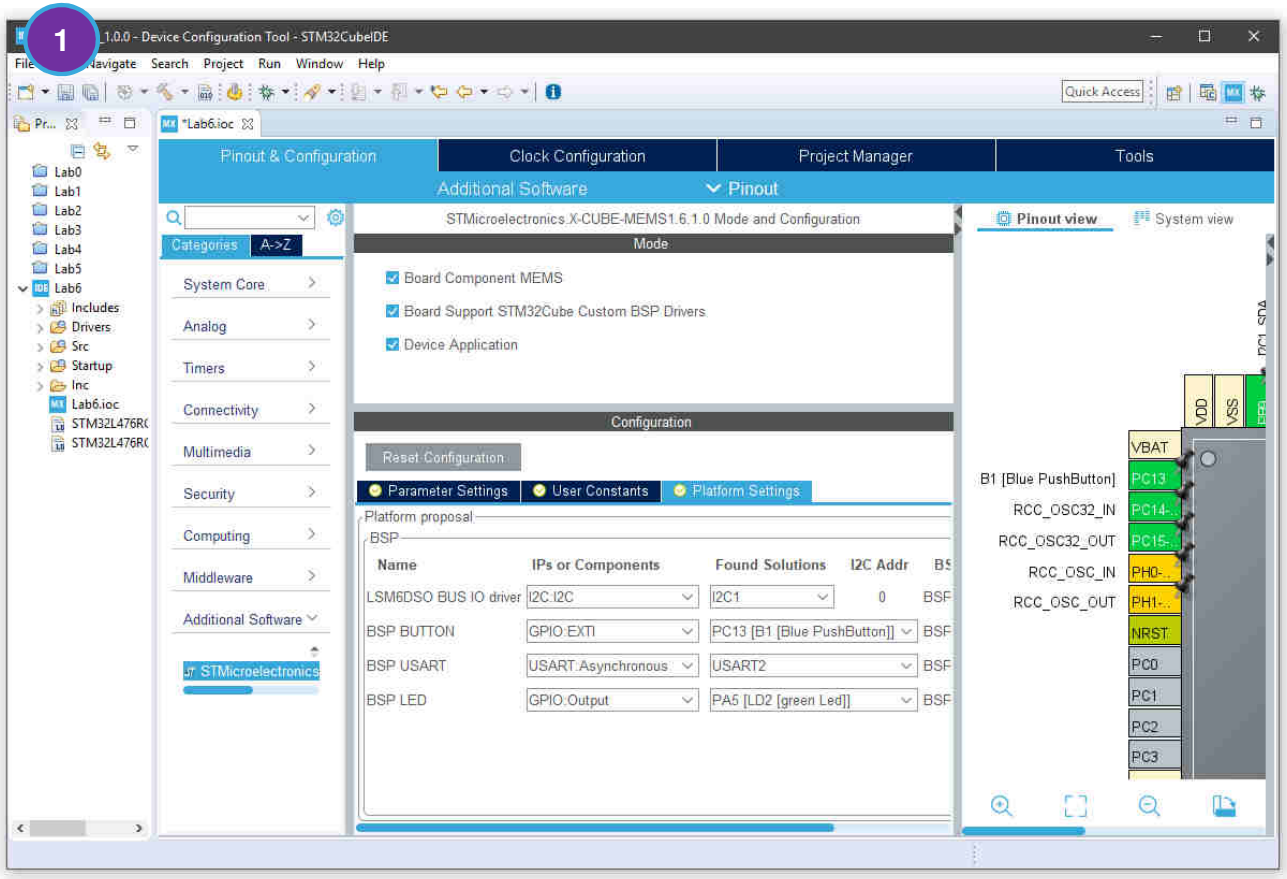


life.augmented

Lab6 – Save the Project

1. Click the save button 

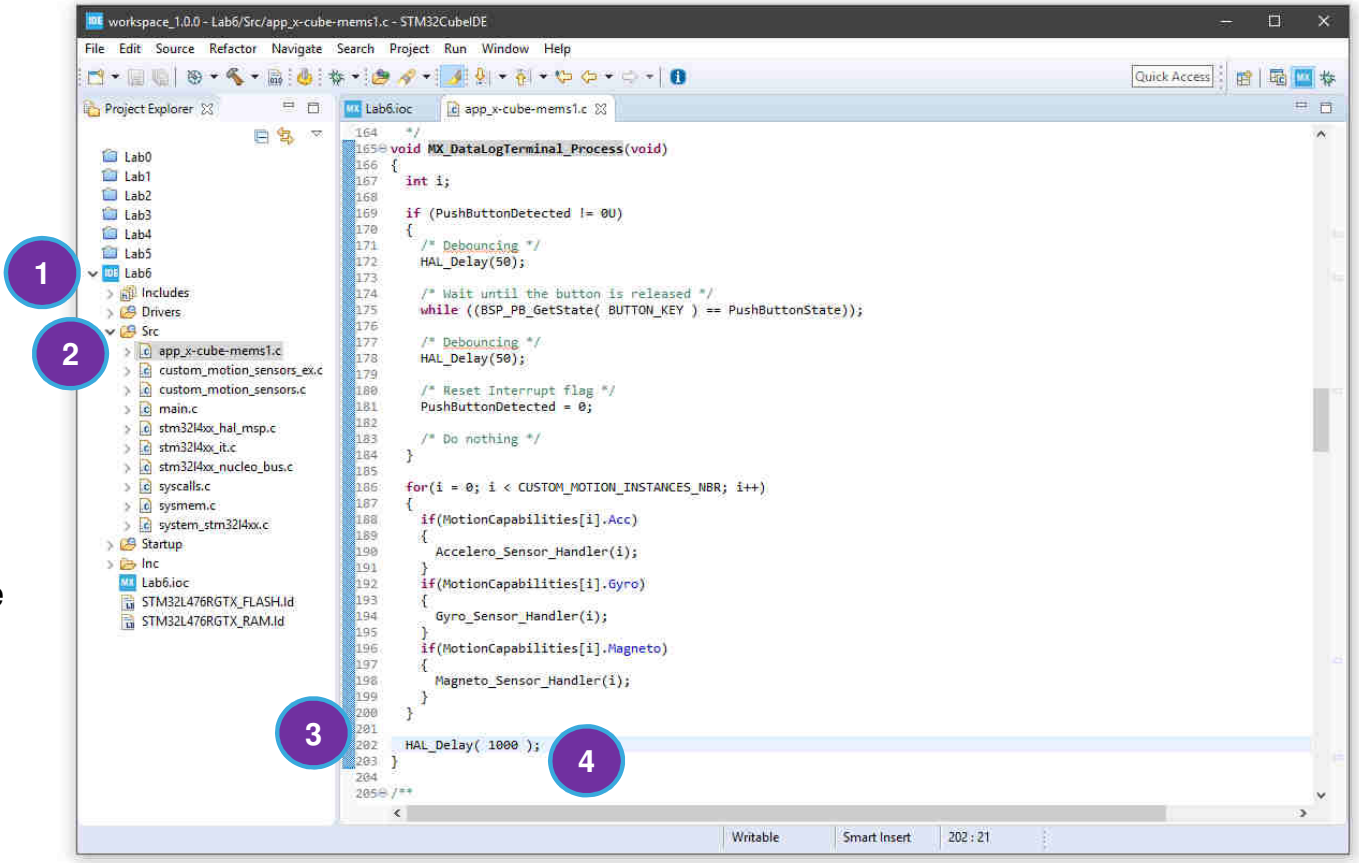
   This action will generate the source code of this lab



Lab6 – Code Editing 1/2

1. Expand **Src** in folder **Lab6**
2. Double click on **app_x-cube-mems1.c**
3. Go to line **#202**
4. Change it from **HAL_Delay(1000);** to **HAL_Delay(50);**

Note: Sensor ODR is 6667 Hz, meanwhile output on serial terminal is updated every 50 ms (20 Hz)

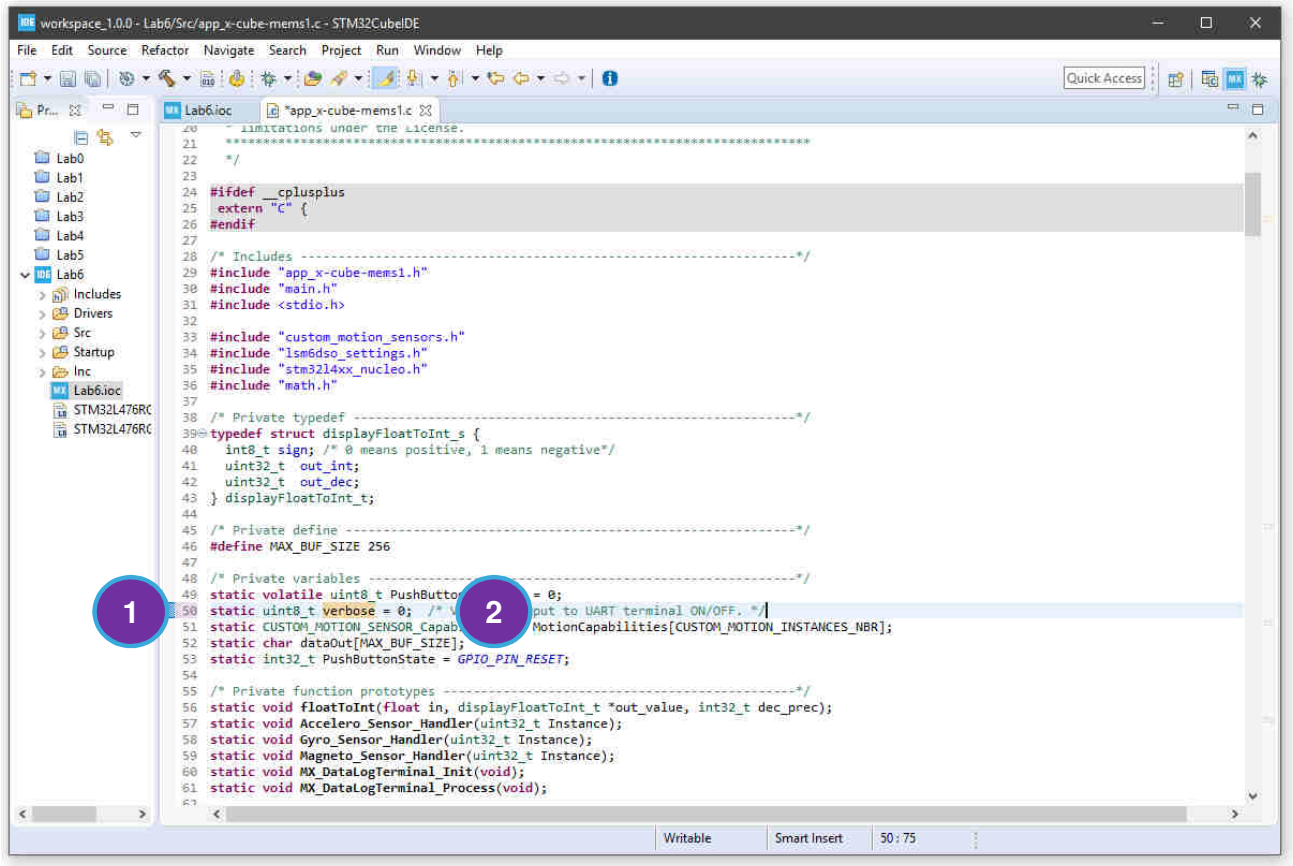


life.augmented

Lab6 – Code Editing 2/2

1. Go to line **#50** of file **app_x-cube-mems1.c**

2. Change it from
   static uint8_t verbose = 1;
   to
   static uint8_t verbose = 0;



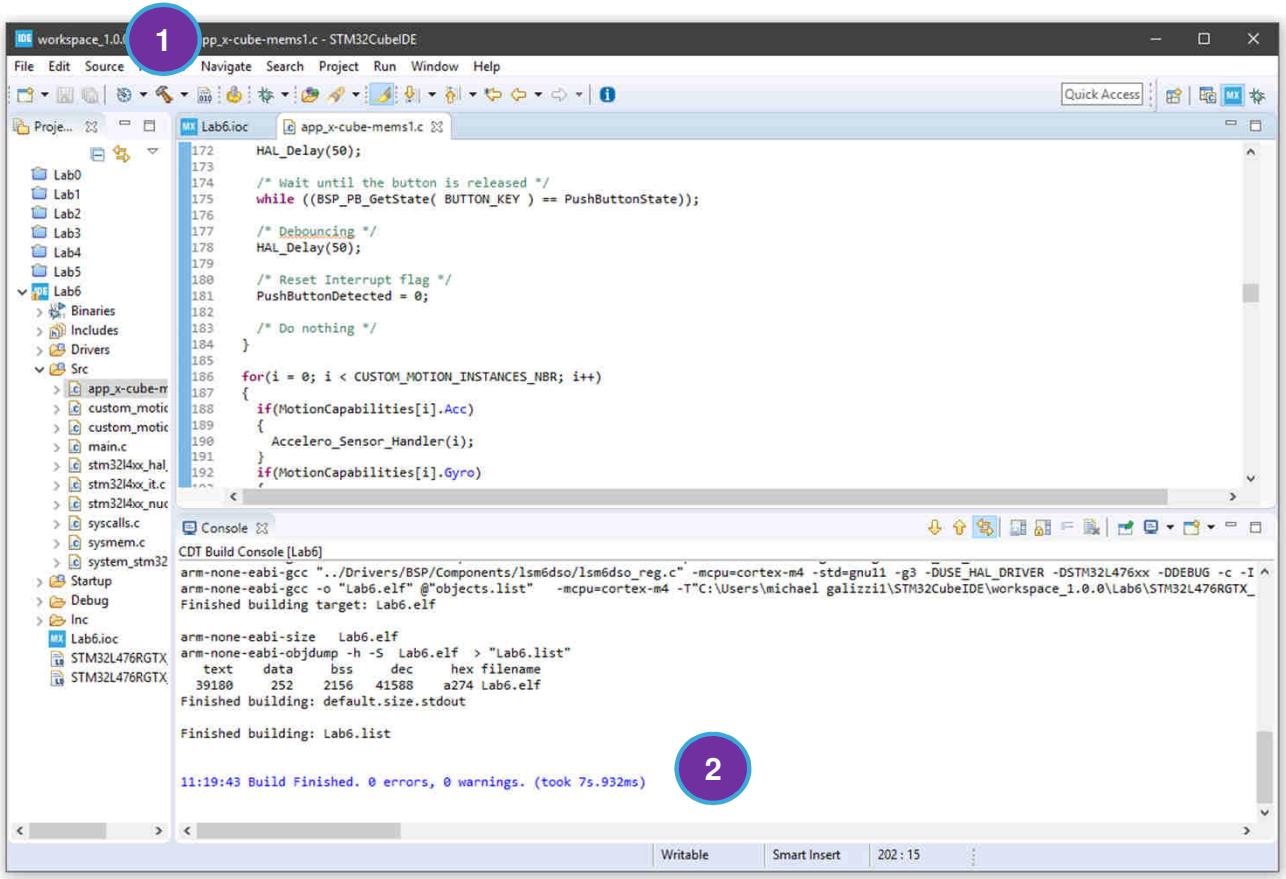
This modification will reduce the serial terminal output info to only RAW accelerometer and gyroscope data

life.augmented

Lab6 - Compiling

1. Click on the hammer  to begin compilation, or press **CTRL+B**

2. Compilation should terminate with 0 errors and 0 warning



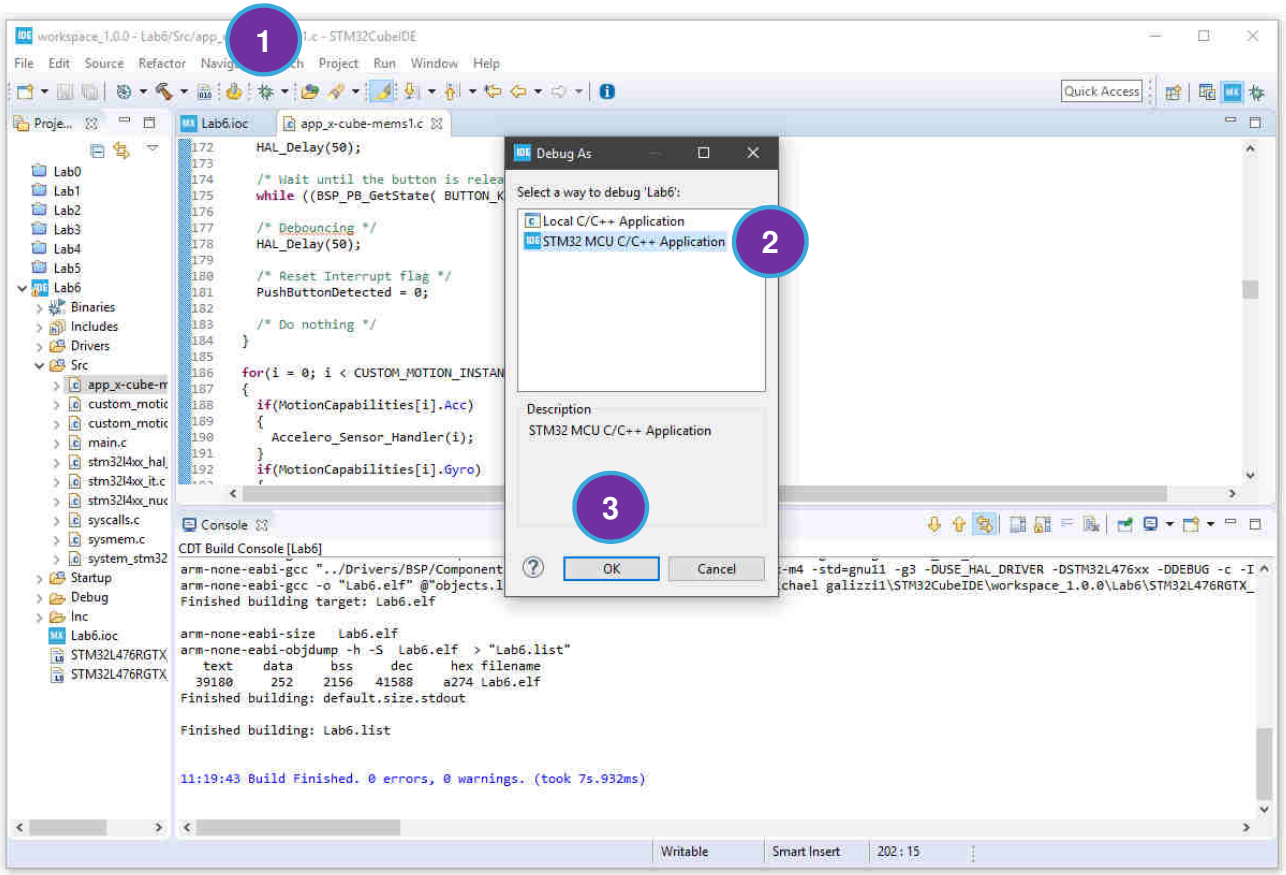
life.augmented

Lab6 - Debugging

1. Click on the bug  to begin debugging

2. Select
   STM32 MCU C/C++ App

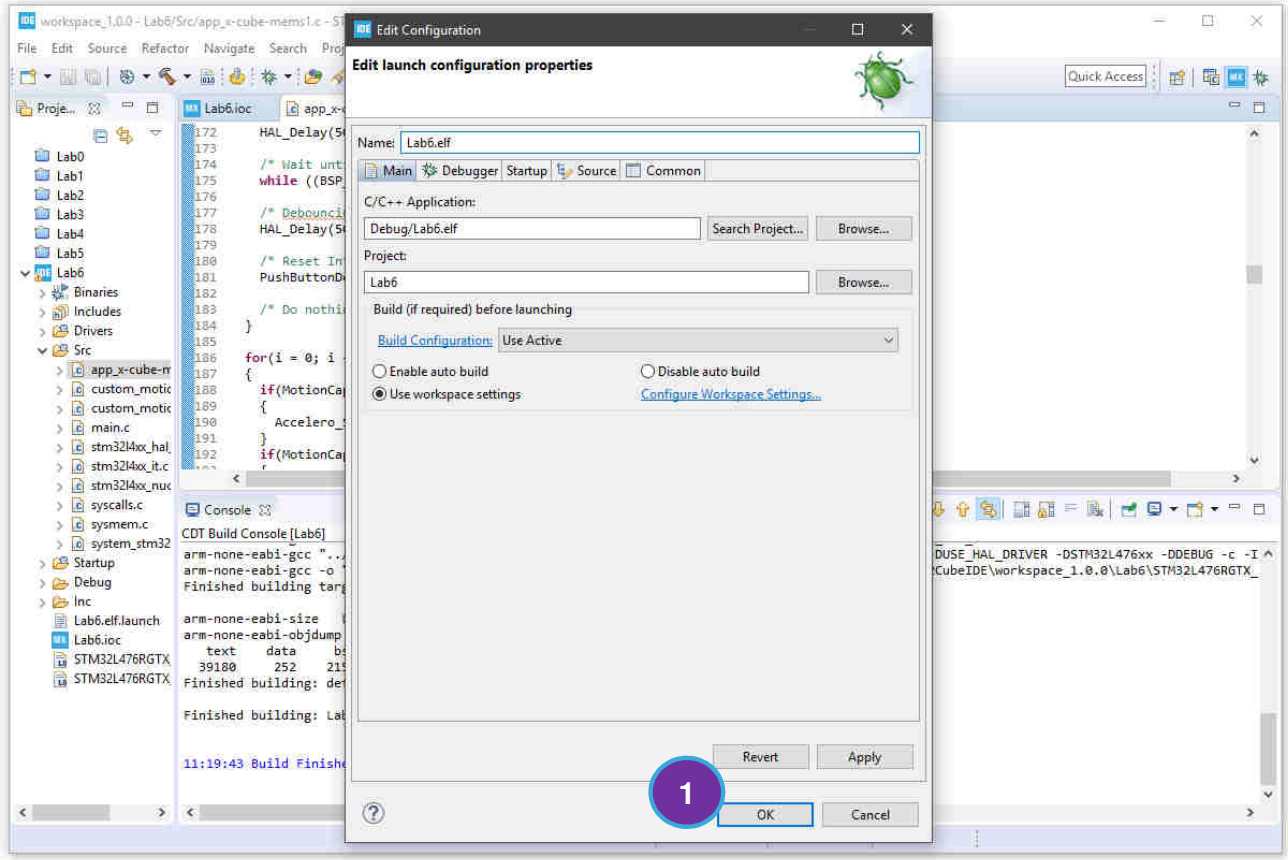
3. Click **OK**



life.augmented

Lab6 - Debugging

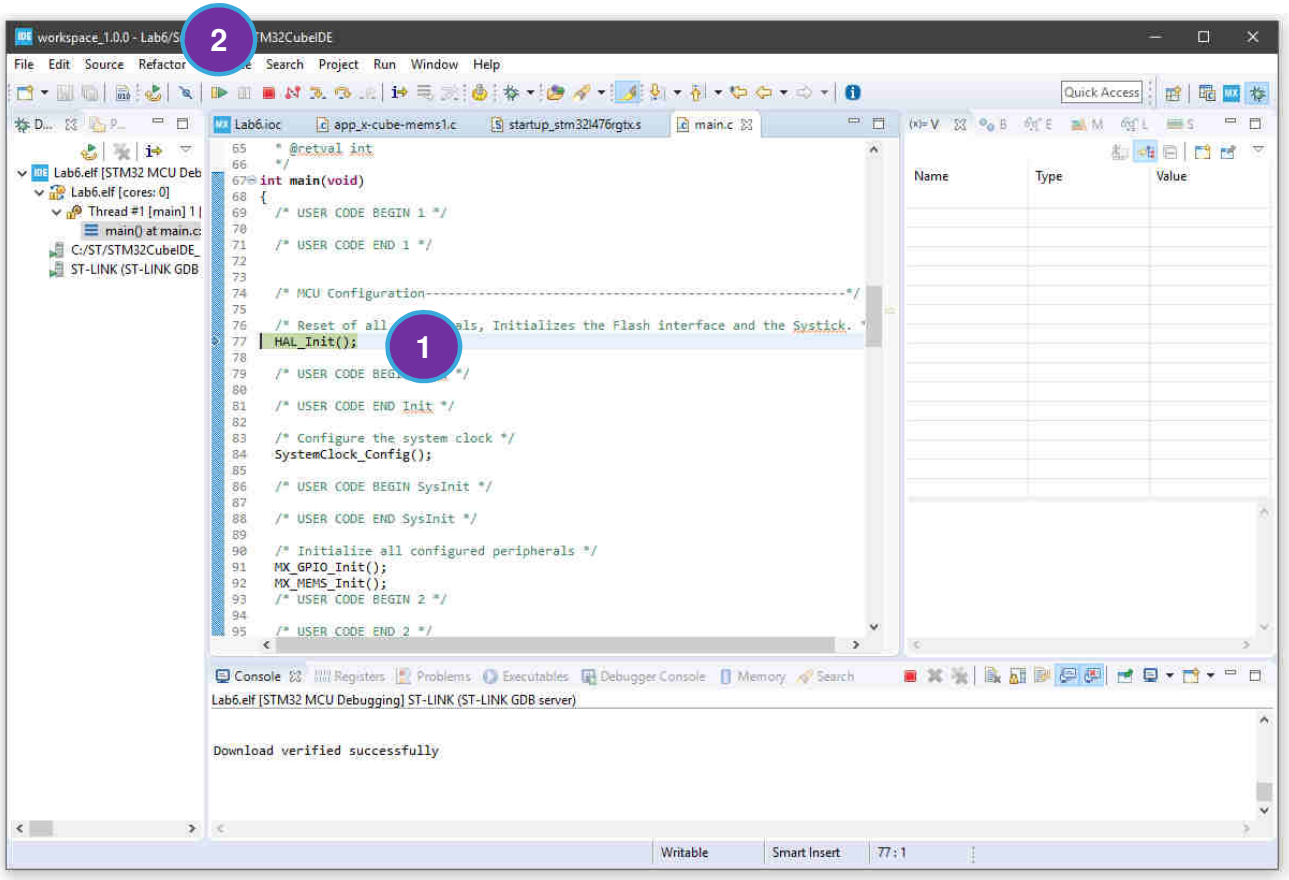
1. Click **OK**



life.augmented

Lab6 - Debugging

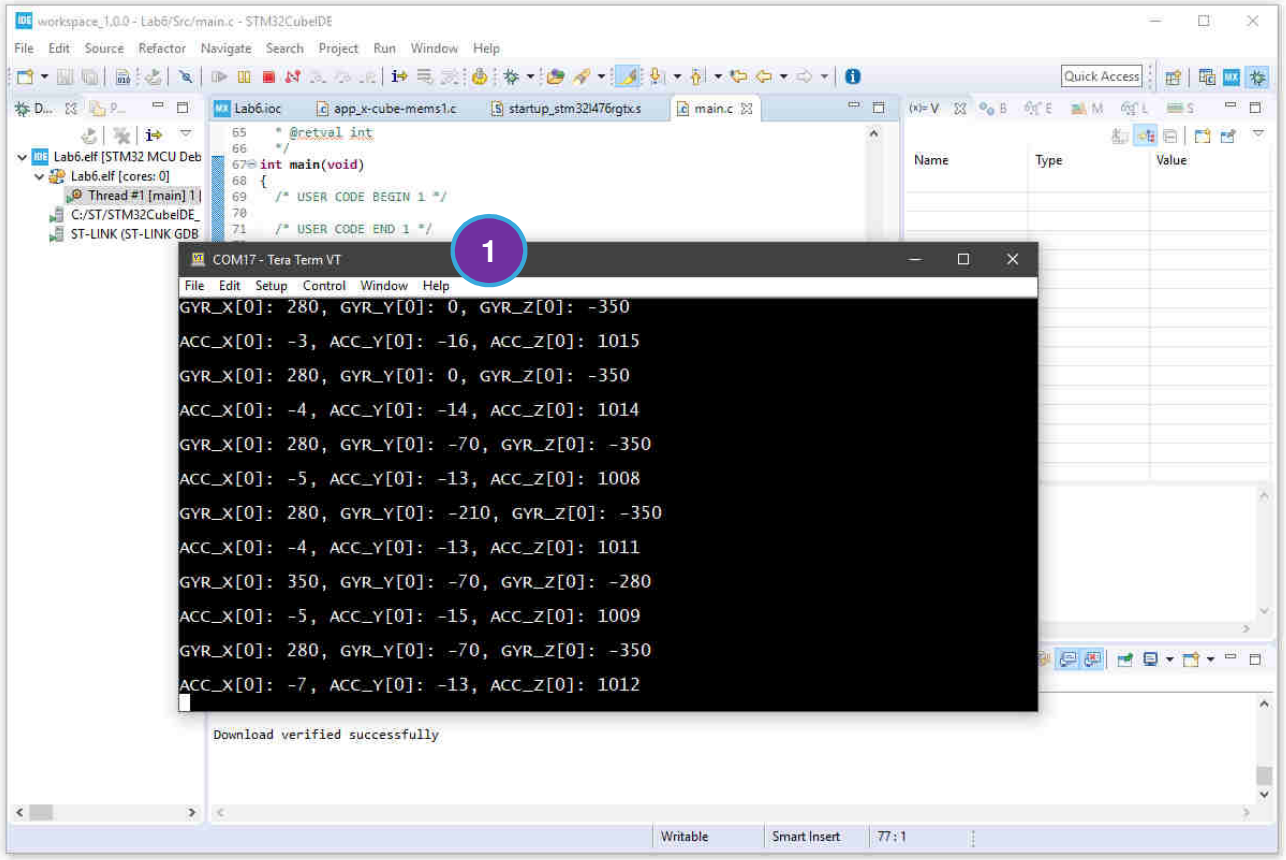
1. Code start at the first line of the main function
2. Click play  button to run the code



life.augmented

Lab6 - Debugging

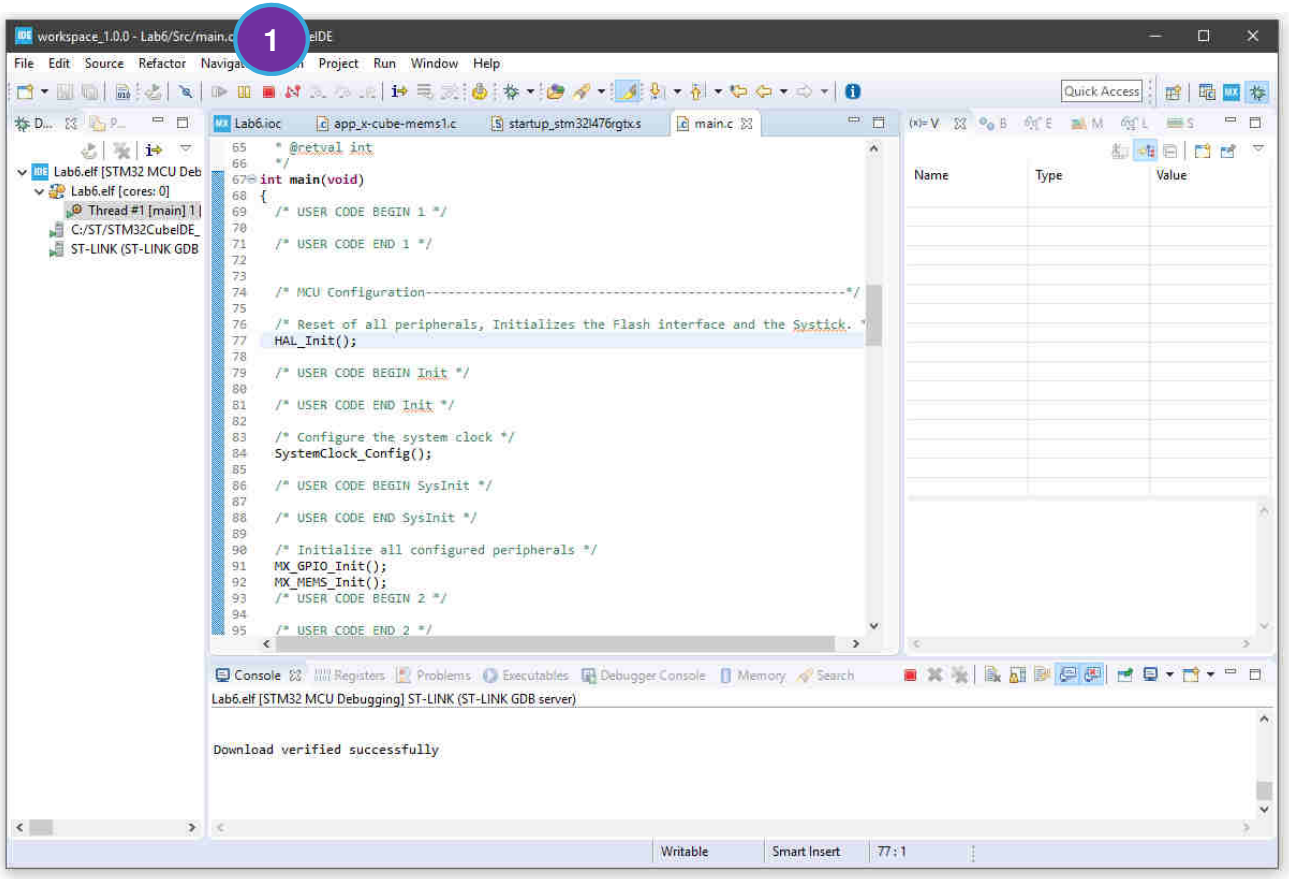
1. Open Tera Term to view the output



life.augmented

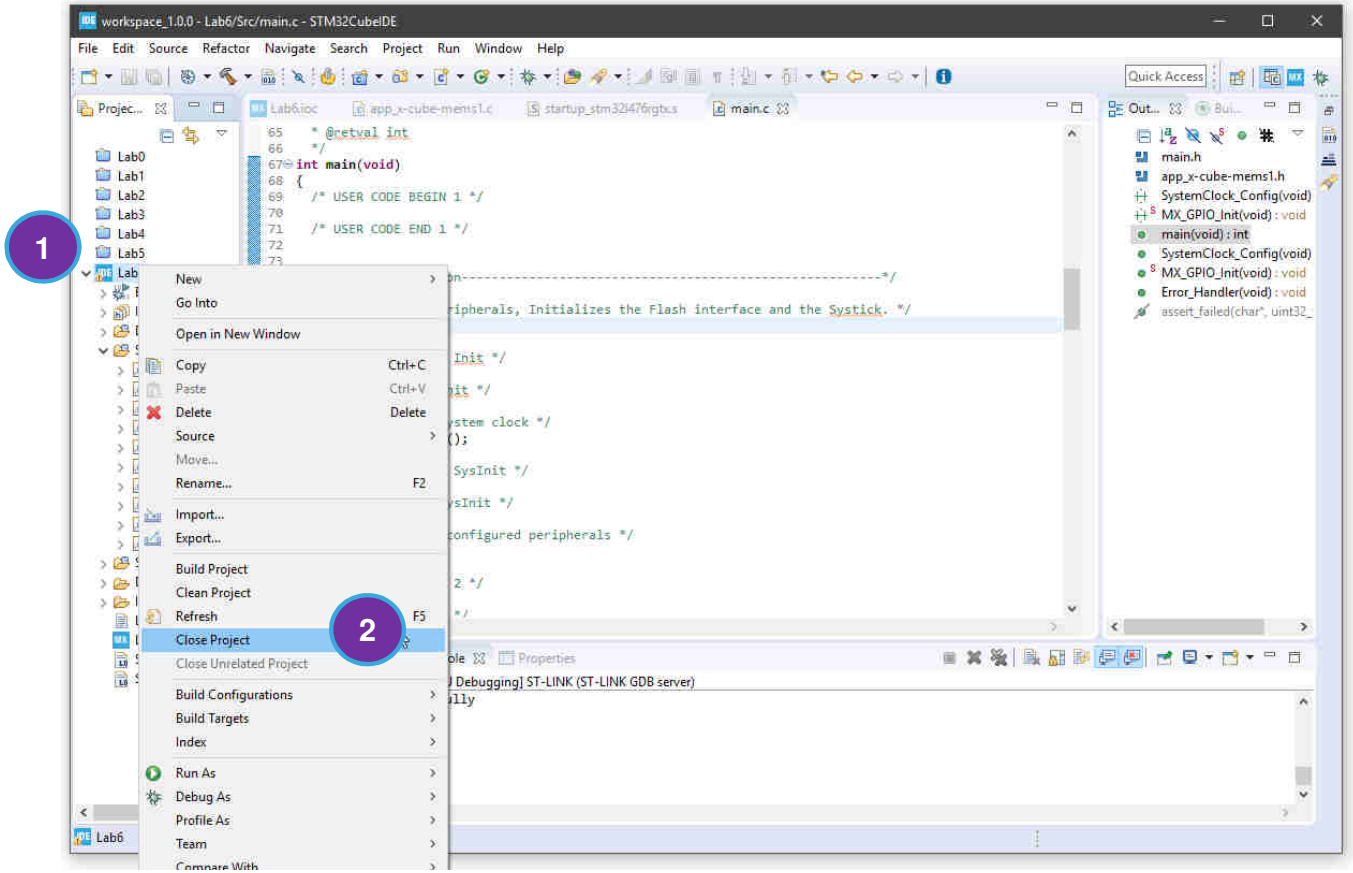
Lab6 - Debugging

1. Click stop  button to interrupt the debugging



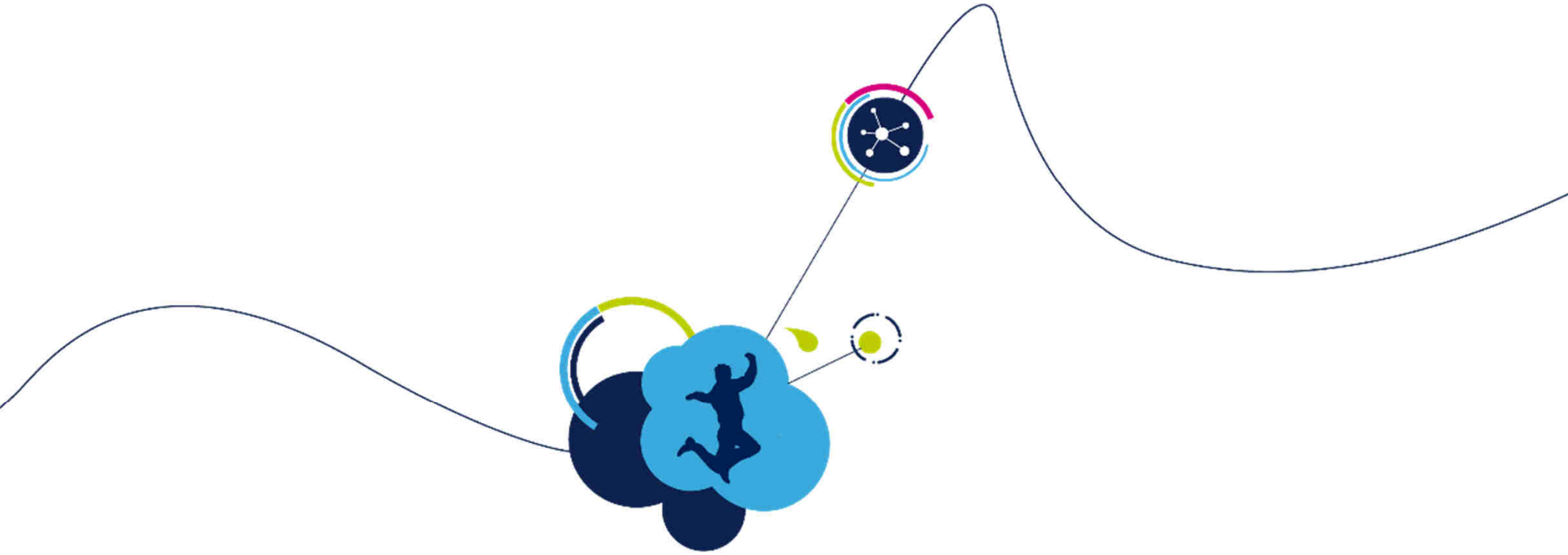
Lab6 – Closing the Project

1. Right-Click on **Lab6** project
2. Click on **Close Project**



Thank You!

life.augmented

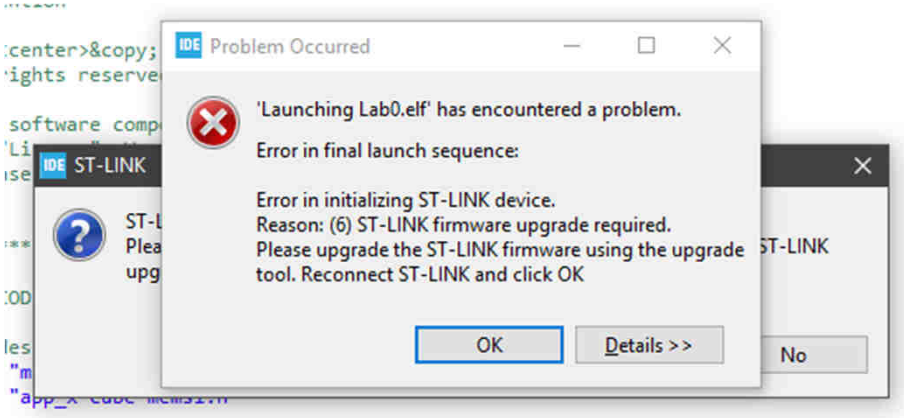


Troubleshooting

life.augmented

Common Issue #1

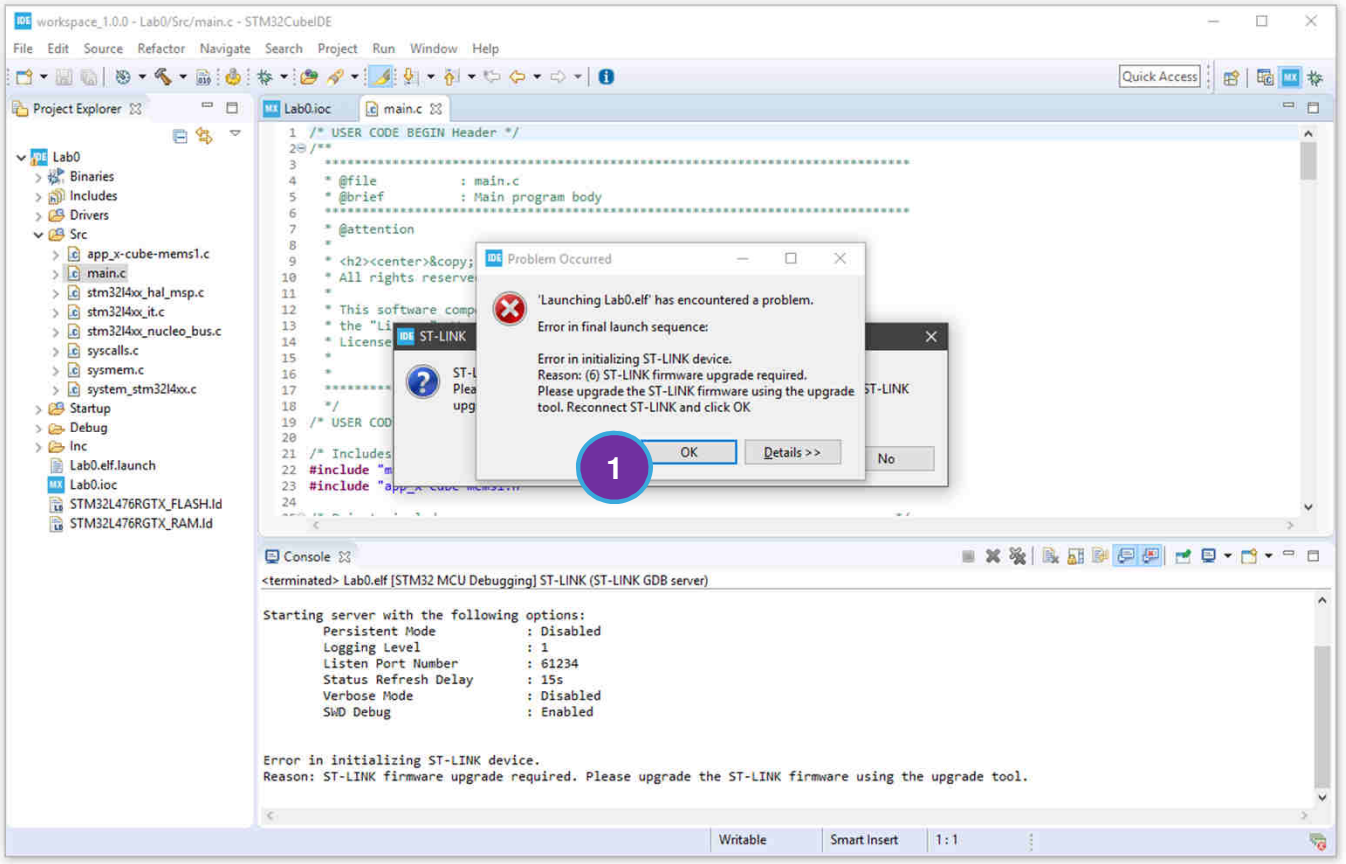
- *Description*: when debugging is launched by pressing  , a **Problem Occurred** because ST-LINK need to be updated



life.augmented

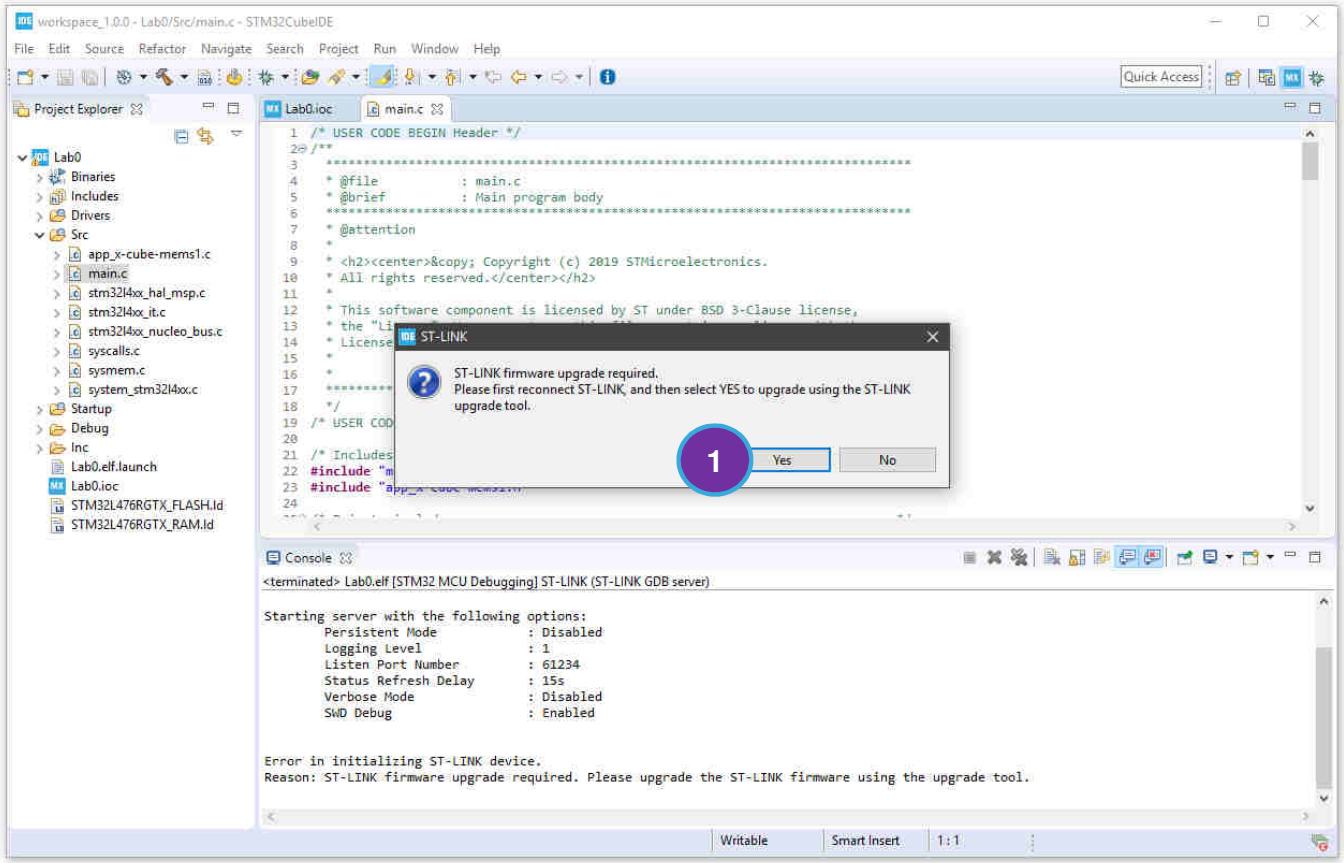
Common Issue #1: Solution

1. Click **OK** to run upgrade to latest firmware



Common Issue #1: Solution

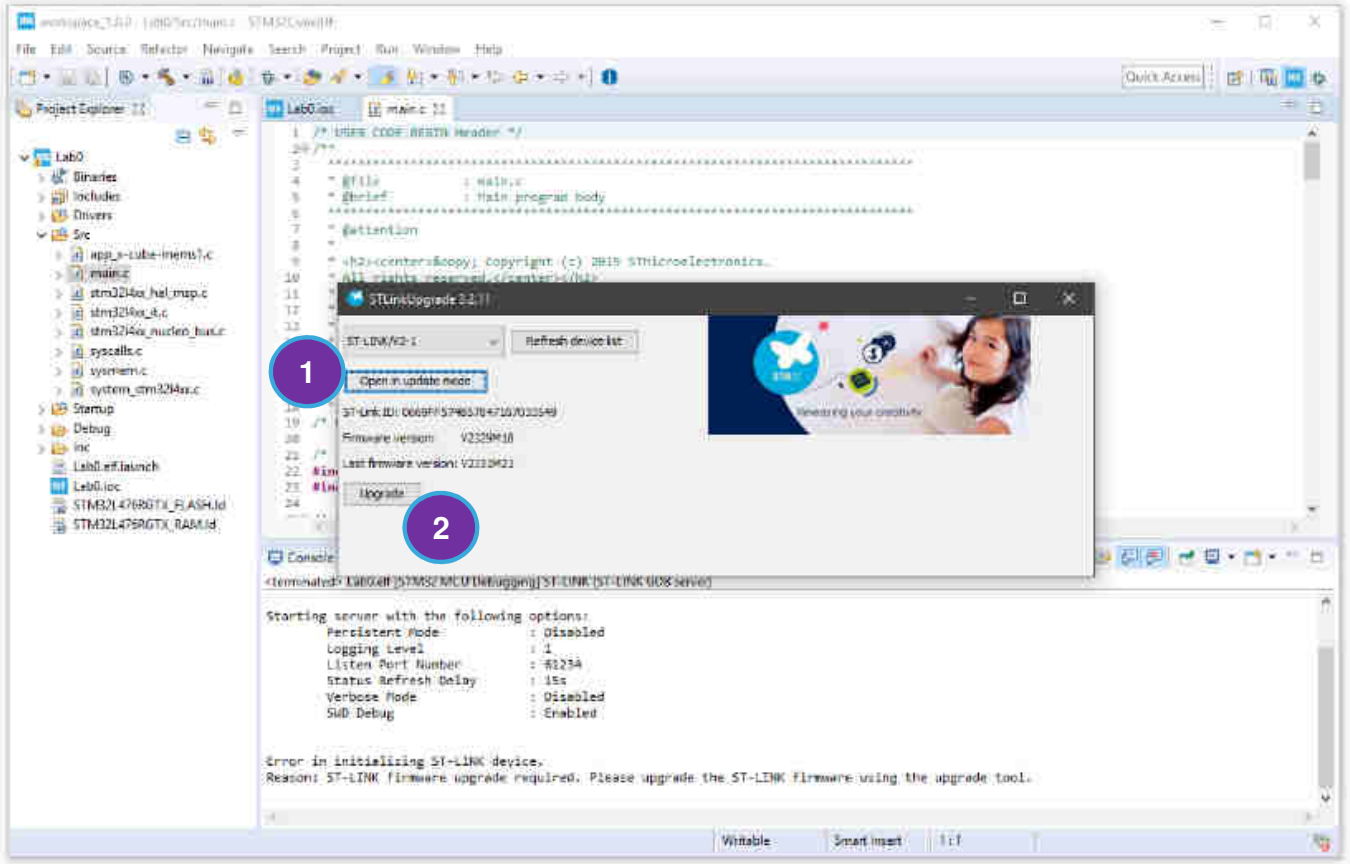
1. Click **Yes** when is asked to upgrade the ST-LINK



life.augmented

Common Issue #1: Solution

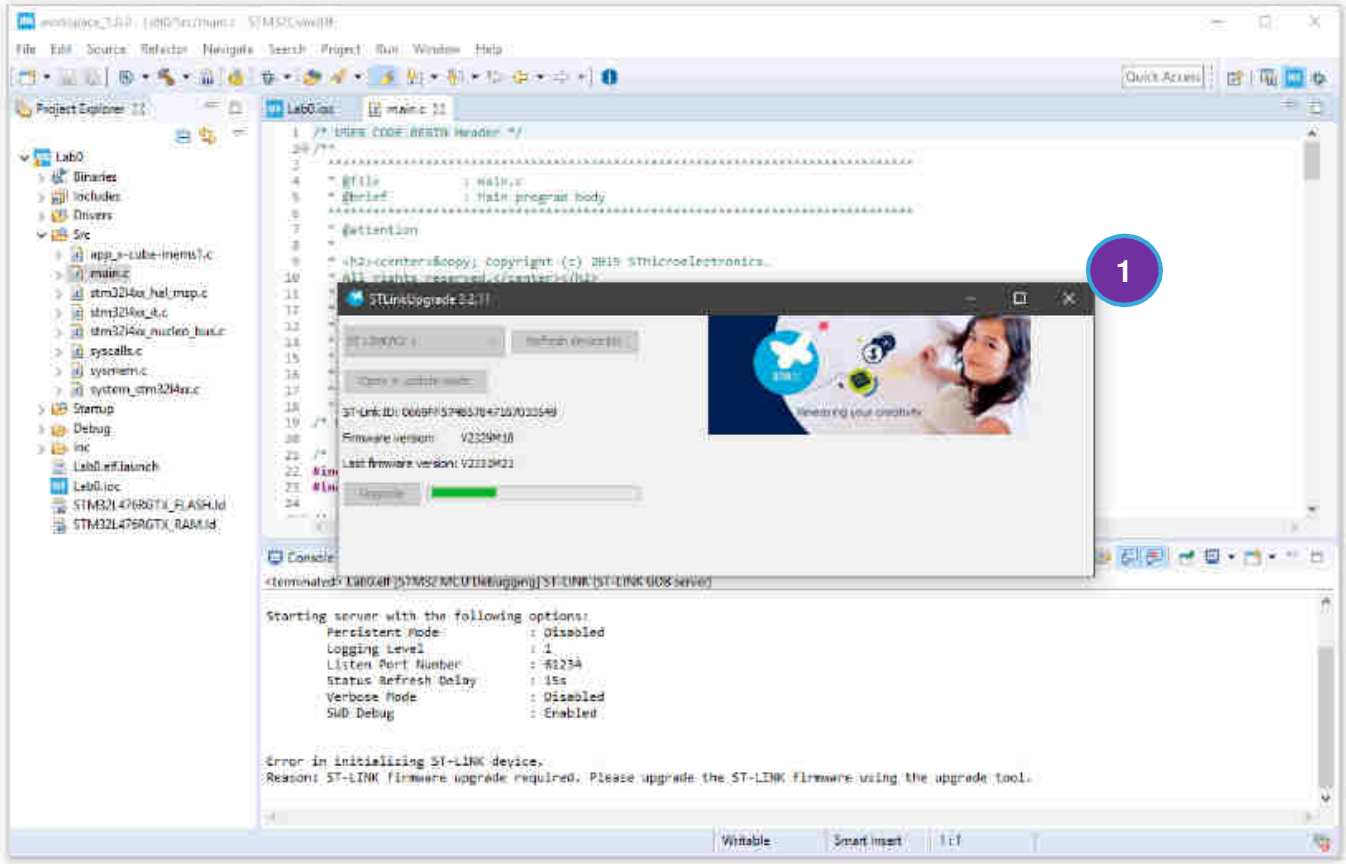
1. Click **Open in update mode** to force ST-LINK
2. Click on **Upgrade**



Common Issue #1: Solution

Wait until update is finished and then close the window.

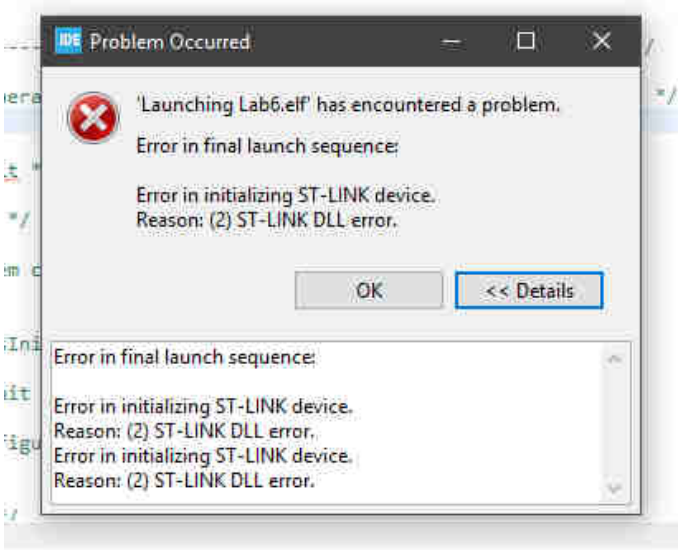
1. Click on **X** when finished



life.augmented

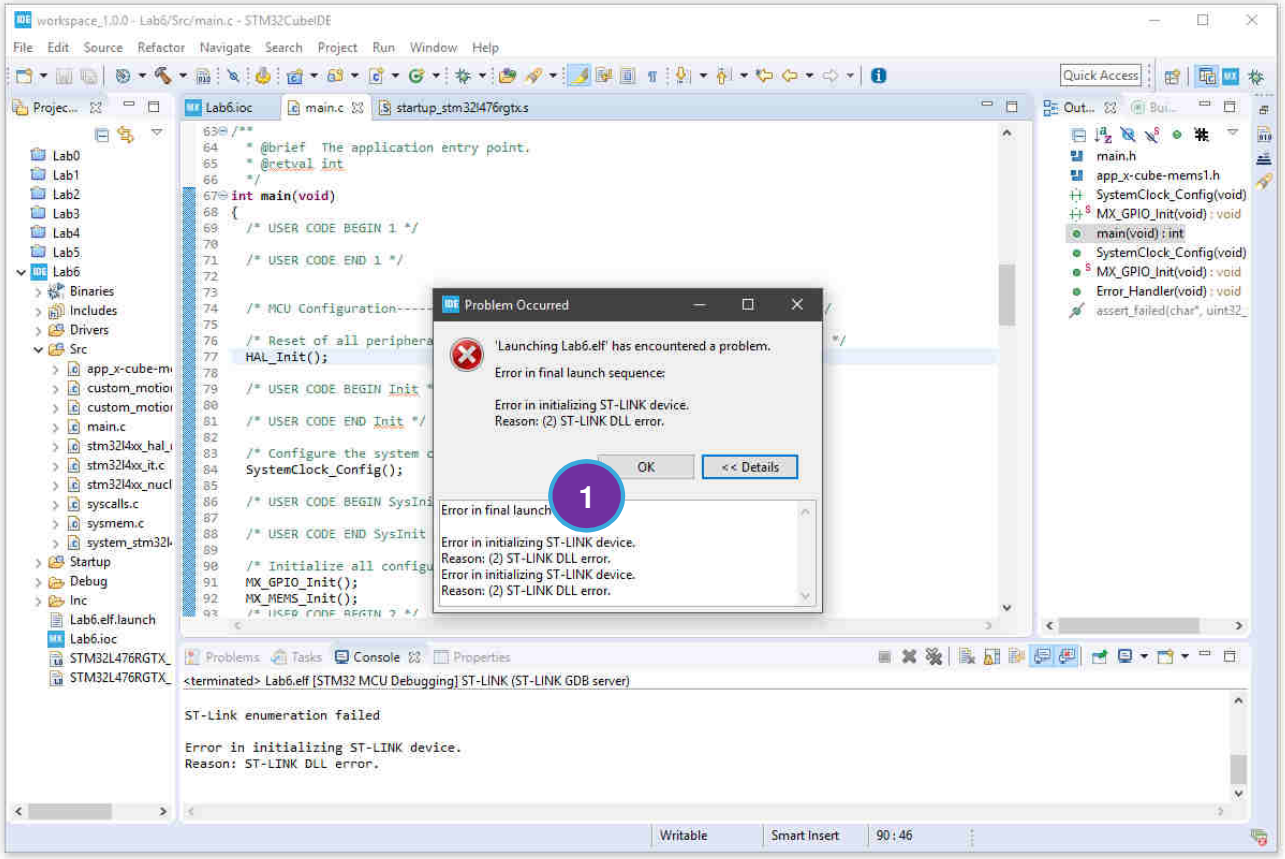
Common Issue #2

- *Description*: when debugging is launched by pressing  , a **Problem Occurred** because ST-LINK is not detected even if it is plugged correctly to the PC



Common Issue #2: Solution

1. Click **OK** and proceed to install ST-LINK driver

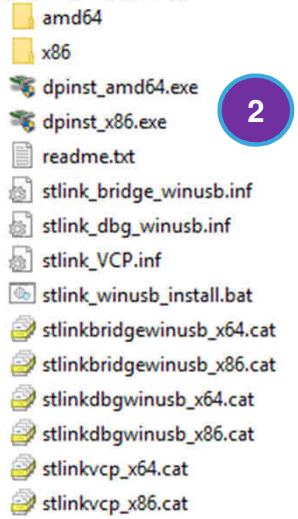


life.augmented

Common Issue #2: Solution

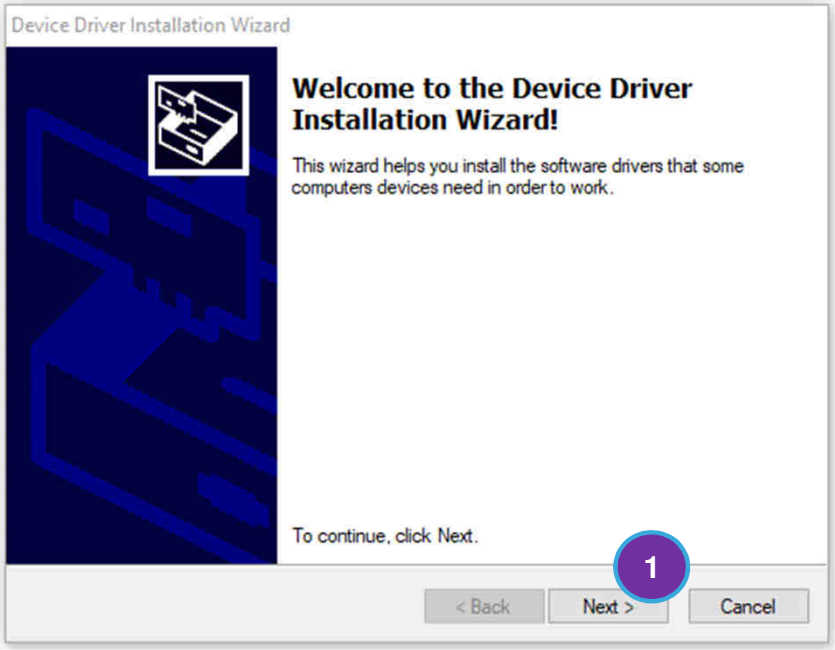
1. Extract the archive **en.stsw-link009.zip** located in **C:\X-CUBE-MEMS1_HandsOn\Drivers**

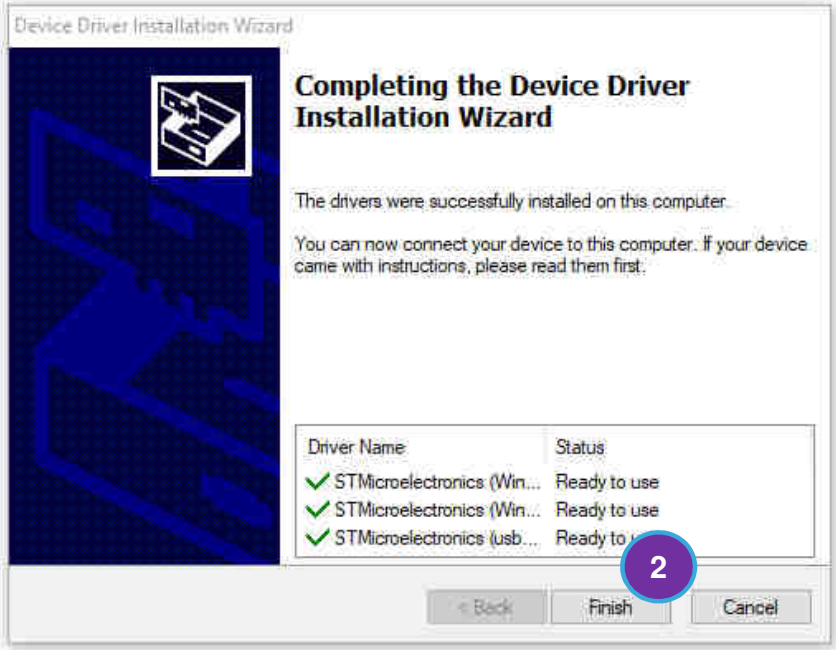
2. Run installer for your PC architecture (32bit or 64bit)



Common Issue #2: Solution

1. Click **Next >**
2. Click **Finish**

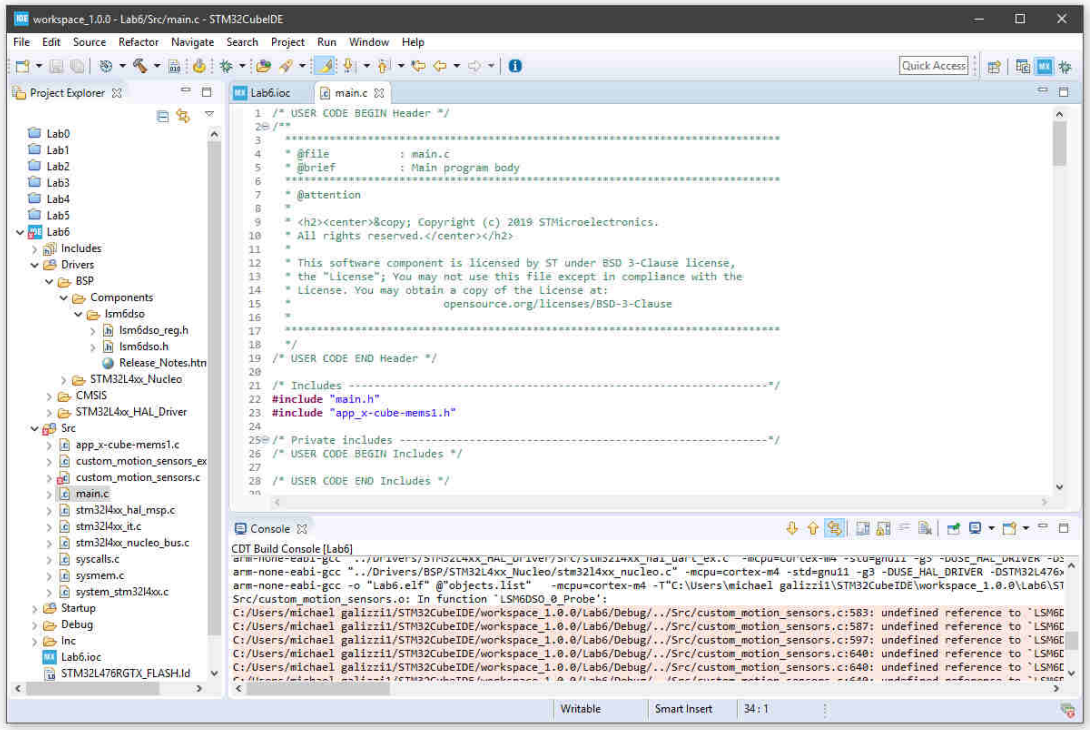




life.augmented

Common Issue #3

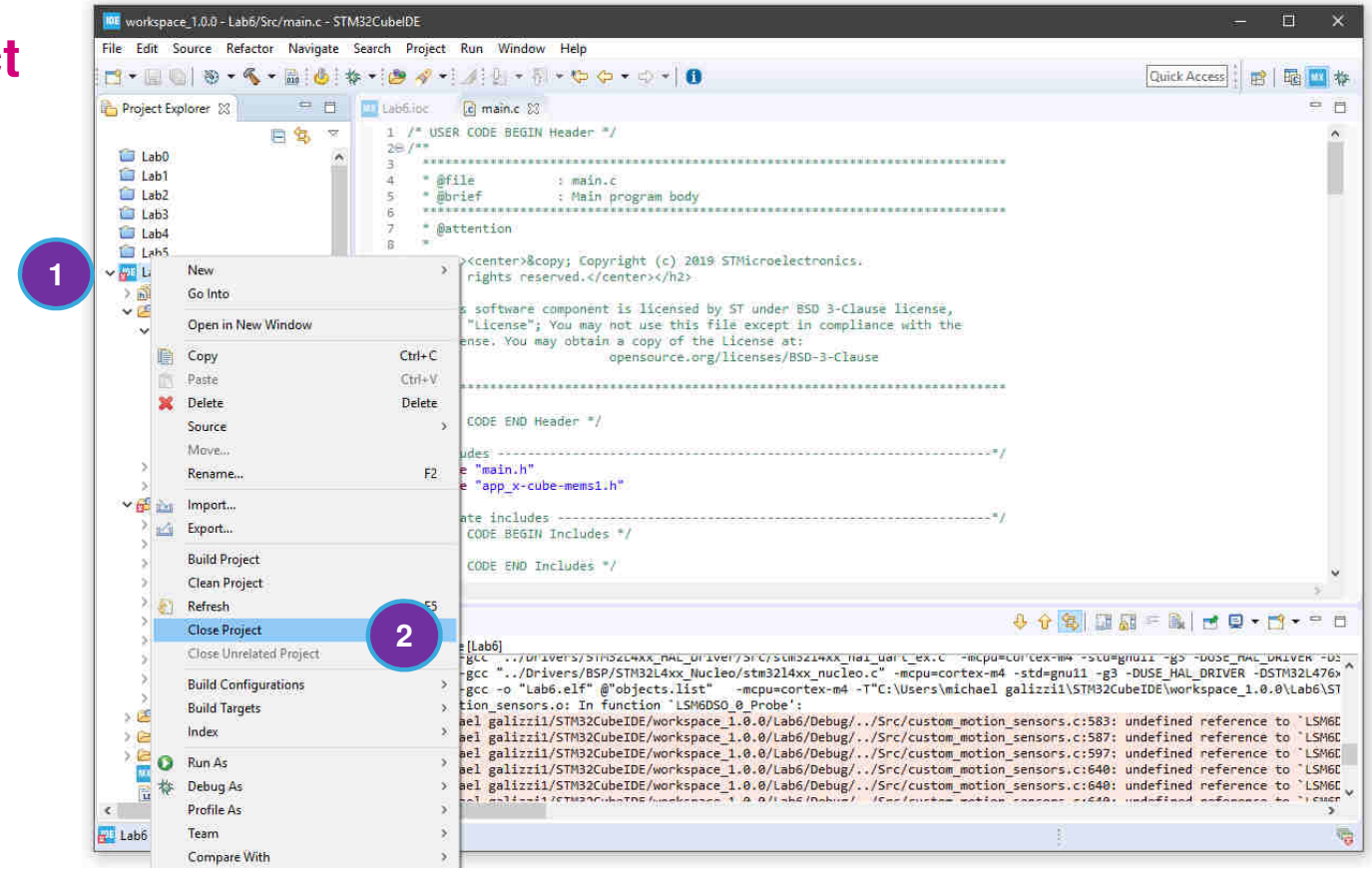
- *Description*: regenerating the code by saving the project  after modify the .ioc file may lead to some missed source file inclusion and compilation will fails.



life.augmented

Common Issue #3: Solution

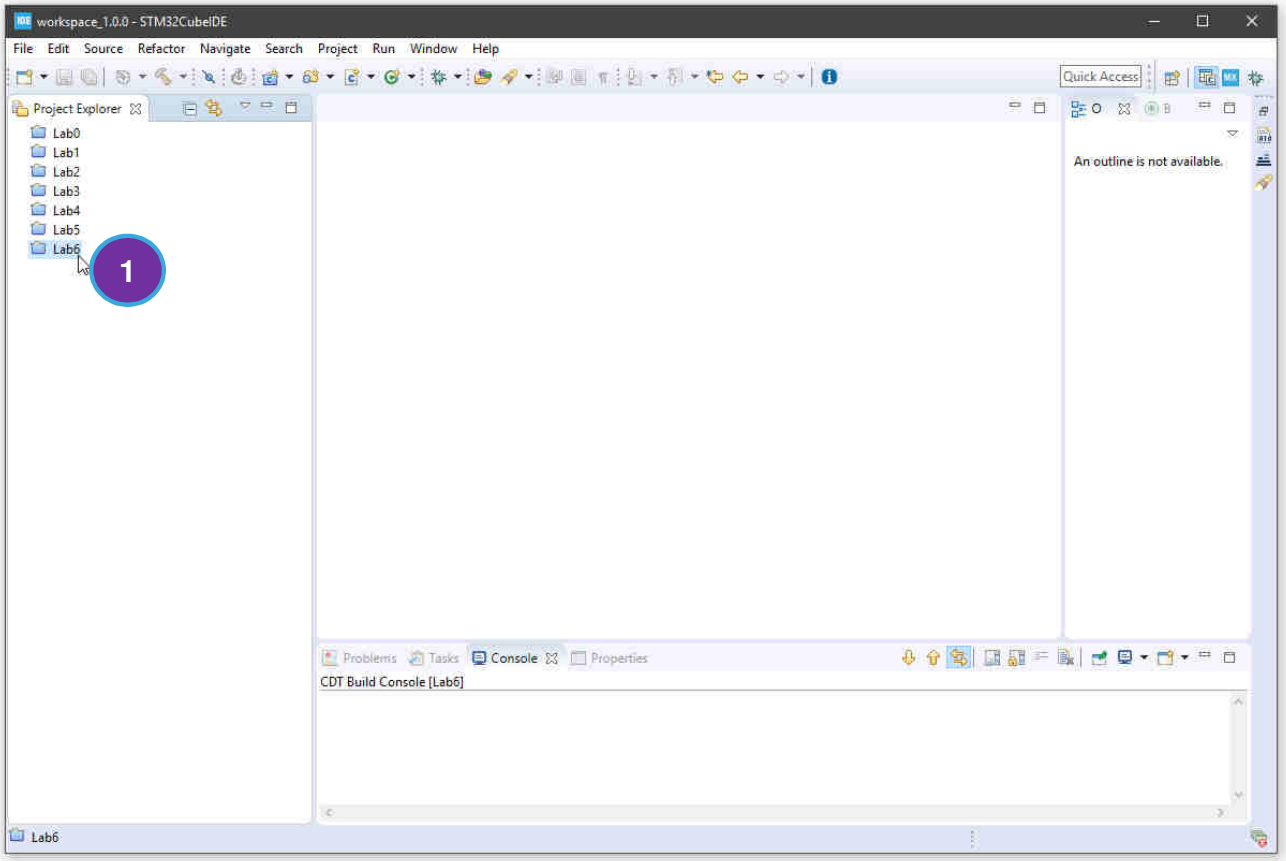
1. Right click on the project
2. Click on **Close Project**



life.augmented

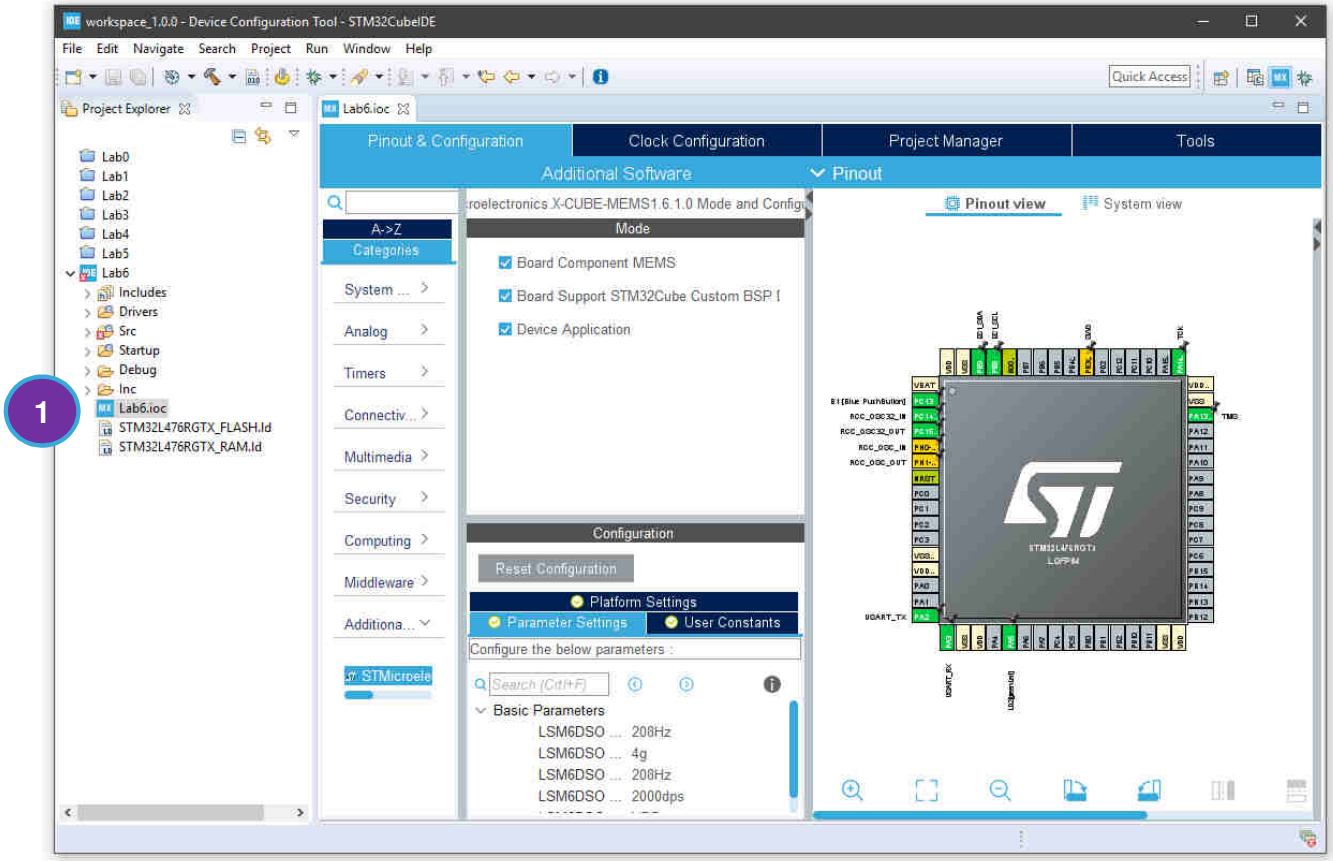
Common Issue #3: Solution

1. Double click the project to open it back



Common Issue #3: Solution

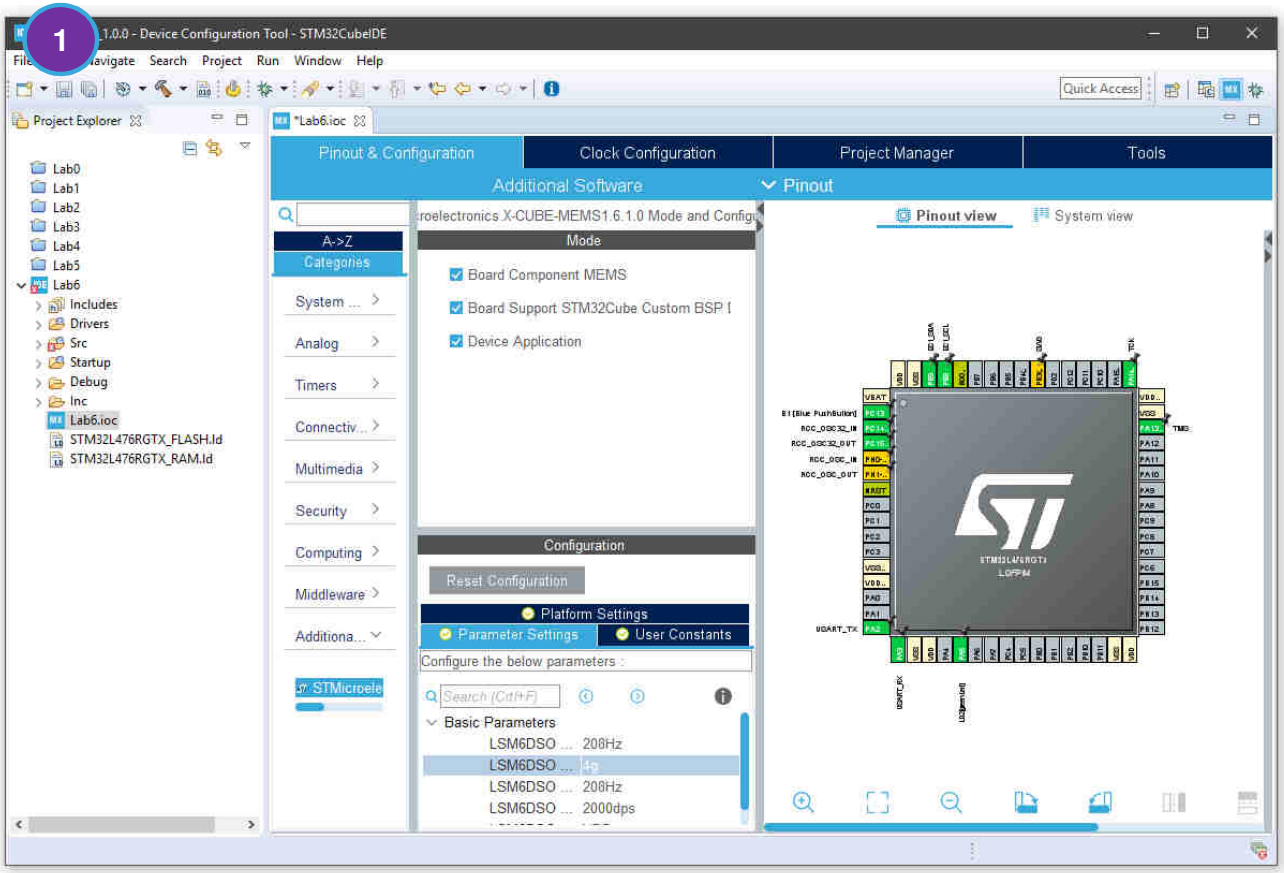
1. **Open the .ioc** file and modify something in the project in order that is possible to save it again



life.augmented

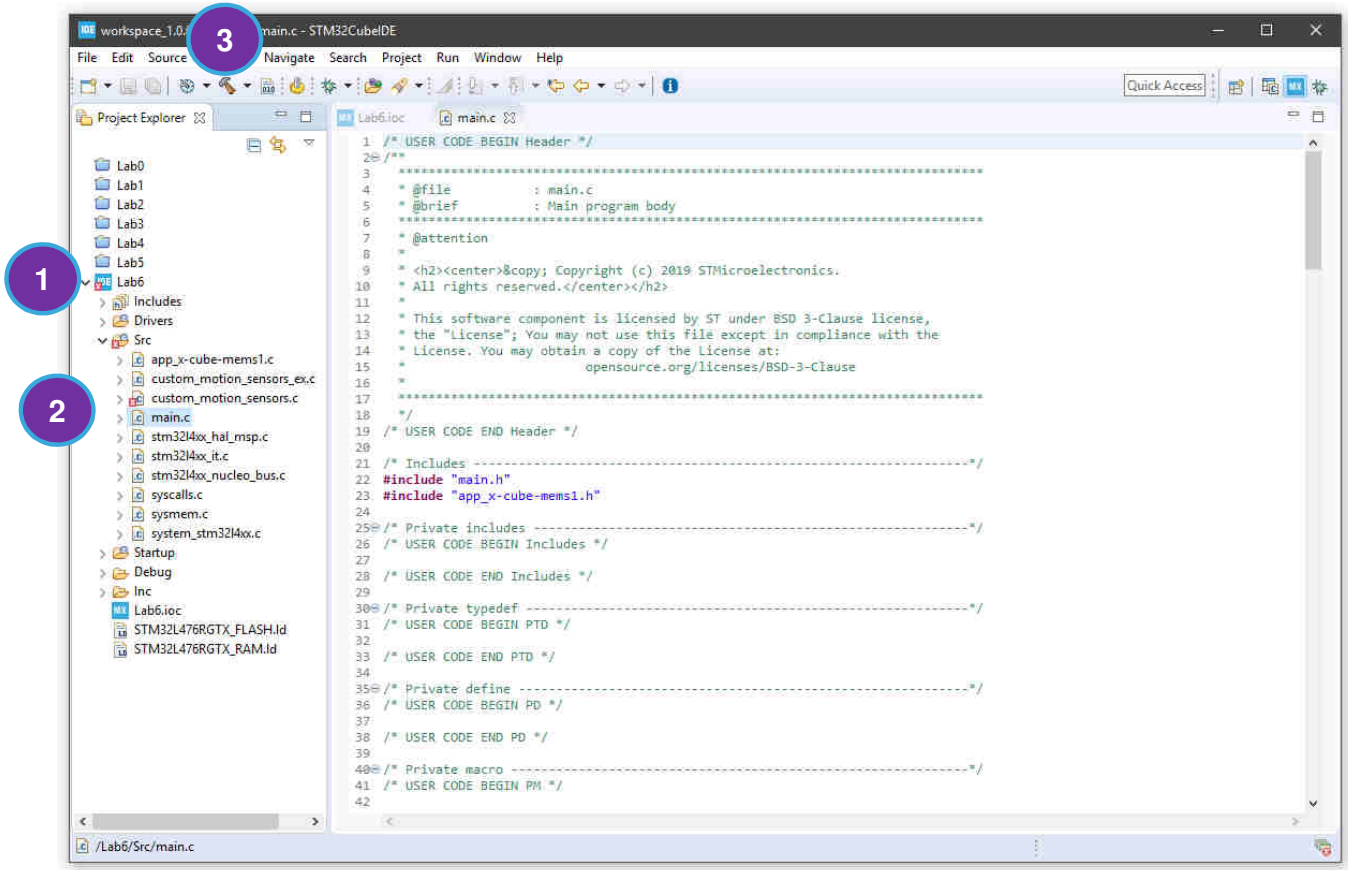
Common Issue #3: Solution

1. Save the project by clicking on 
   At this point included files will be restored correctly



Common Issue #3: Solution

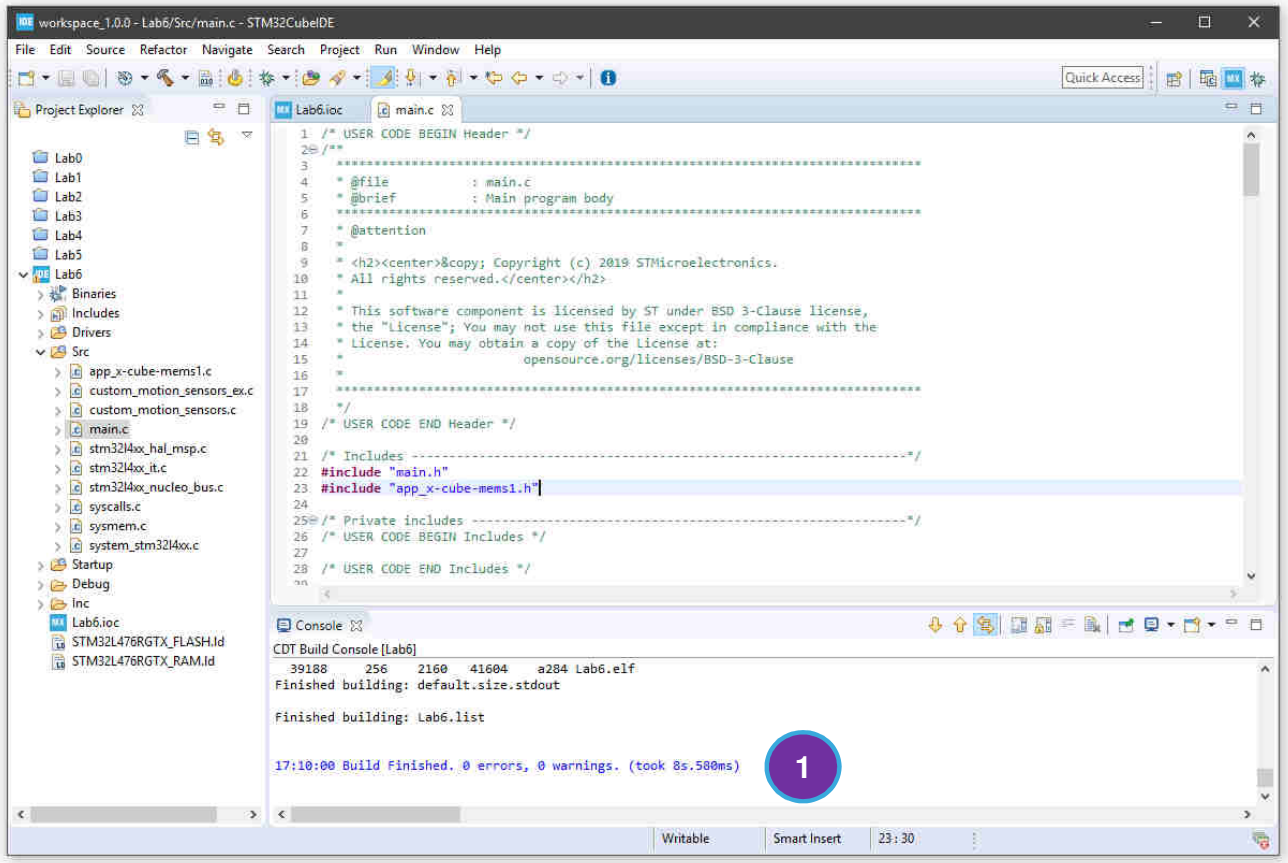
1. Expand **Src** folder and open **main.c**
2. Compile by clicking on 



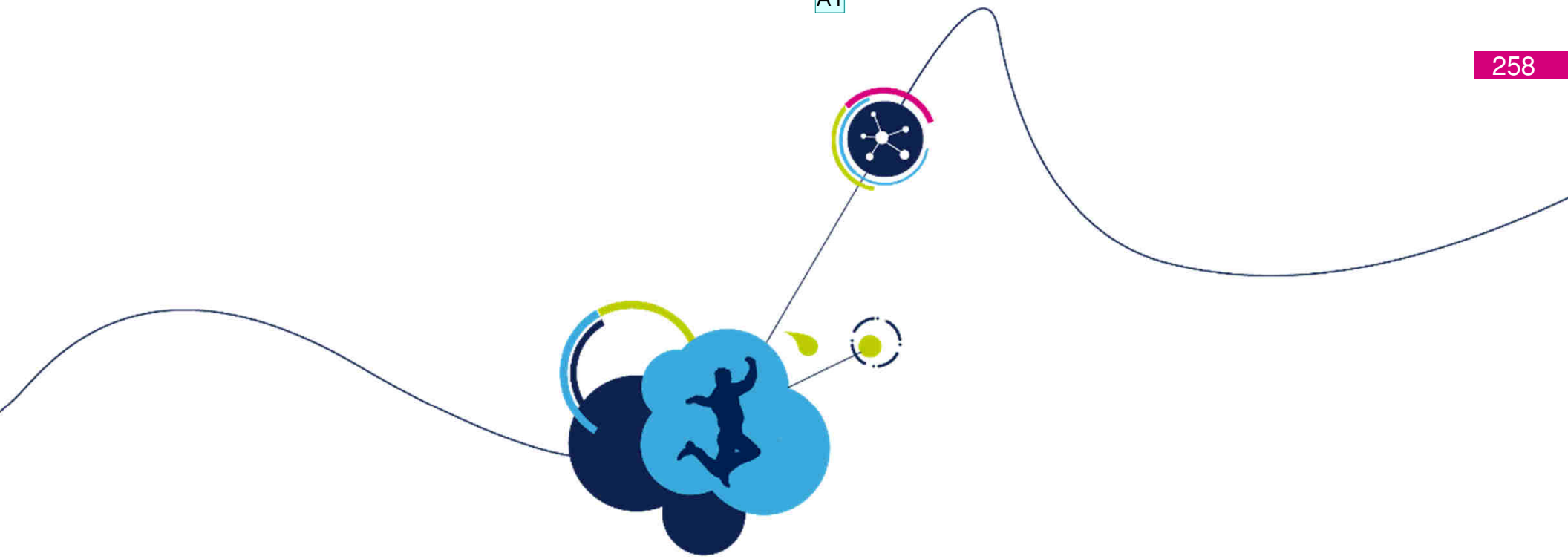
life.augmented

Common Issue #3: Solution

1. Compilation should now terminate without warnings/errors



A1



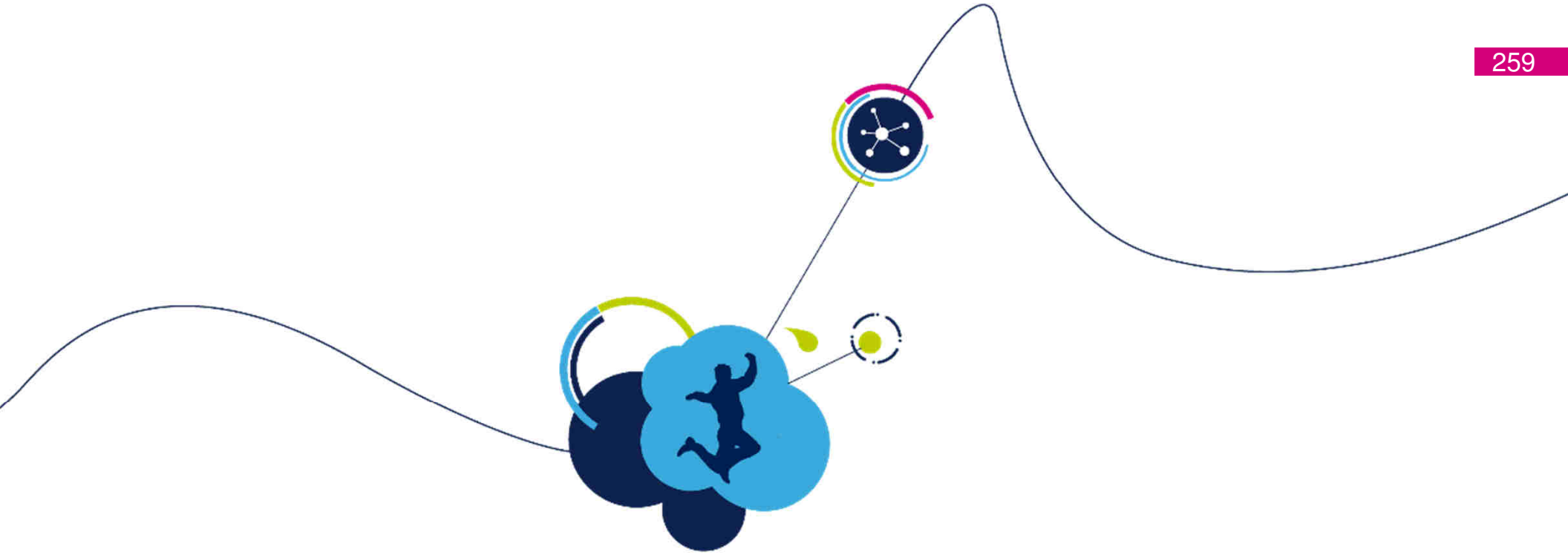
Backup: How to Download and Install sw

Tools from the Web

life.augmented

Slide 258

A1 Title Length issue:
1- Shorten title or use 2 lines
2- Call the macro: B_TOOL_FixTitle_slide()
Author, 8/27/2019

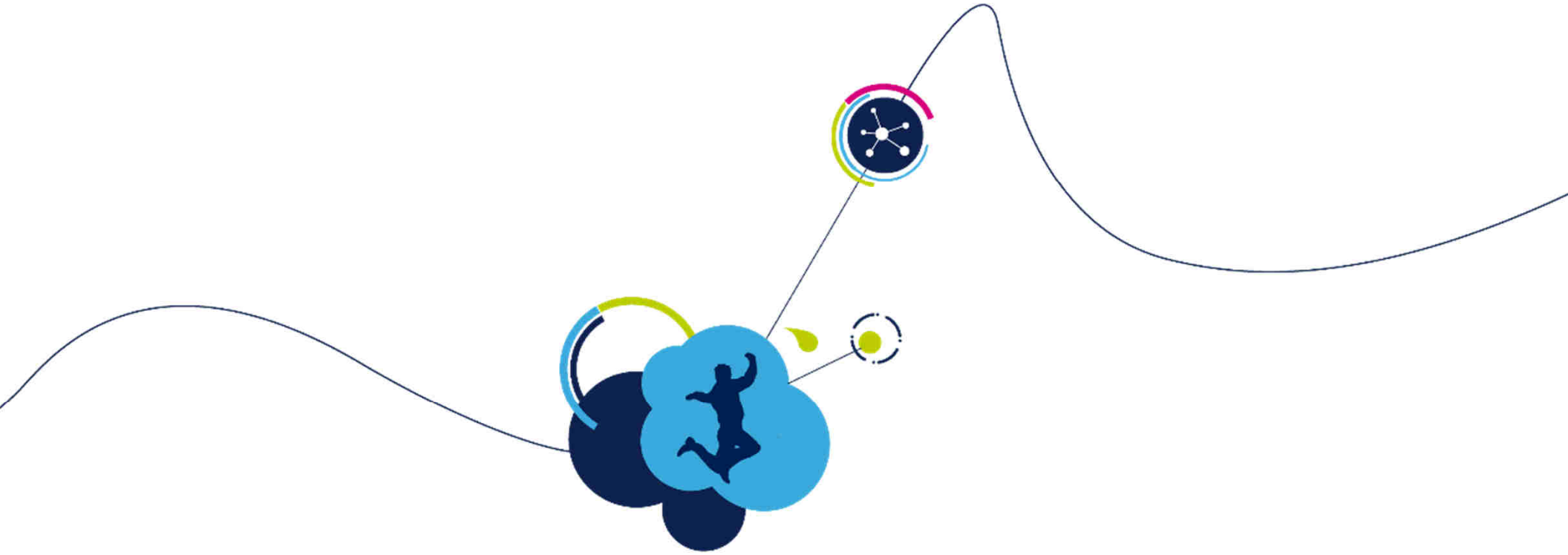


Required HW & SW

life.augmented

Required HW & SW

- **Laptop with USB type A**
- **64 bit OS** Win (7, 10), MacOS, Linux (Generic, Debian, RPM)
  - Virtual Machine is also OK (example: Parallel, VM Fusion…)
  - **Administration rights** are required for SW tools installation
- **Terminal emulator**
  - **TeraTerm** under Windows is OK, download from https://ttssh2.osdn.jp/index.html.en
  - Serial COM port settings: 115200 bps, 8N1, no flow control
- **SW tools from STMicroelectronics installed**
  - **STM32CubeIDE** 1.0.2 integrated development environment, download from www.st.com
    You will need to create a **my.st.com** account.
  - **X-Cube-MEMS1** 6.2.0 software package extension for MEMS sensors, download from STM32CubeIDE after installation (see next slides)

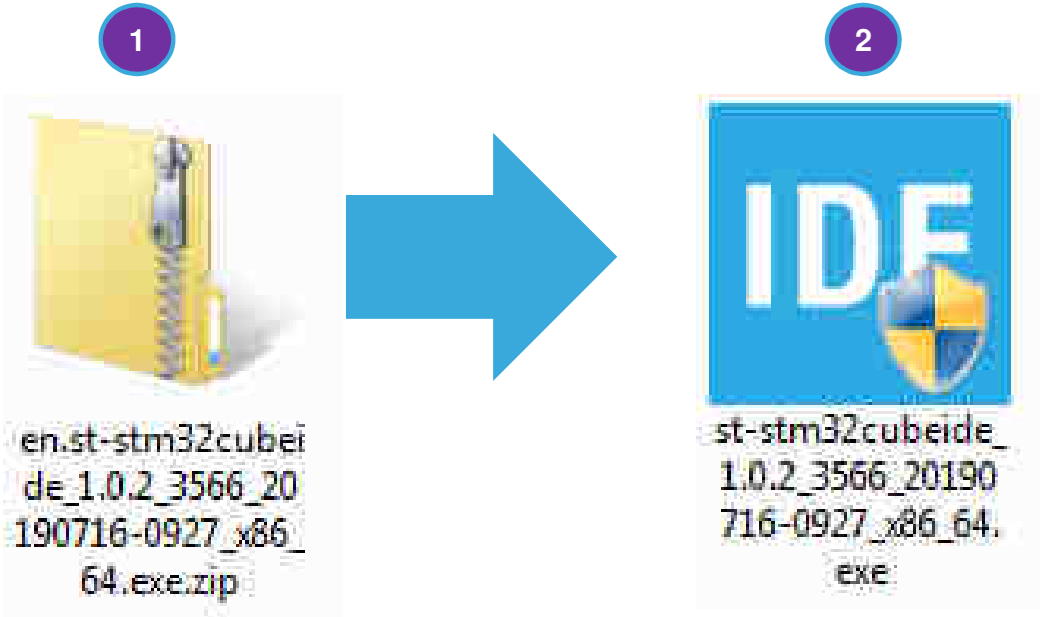


STM32CubeIDE Installation

life.augmented

STM32CubeIDE Installation

1. Go to www.st.com and download the zip package **STM32CubeIDE**
2. Unzip and run the installer



life.augmented

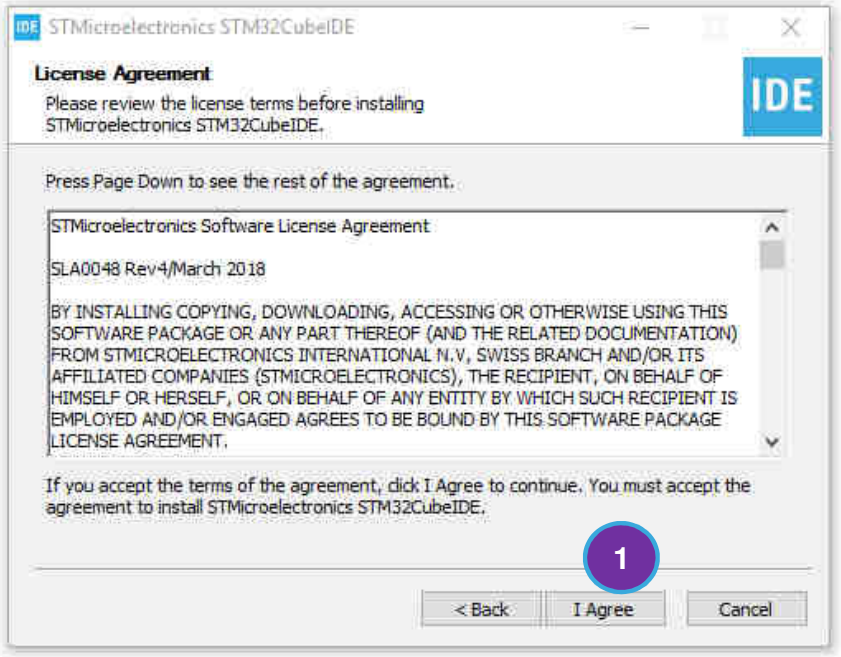
64 bit required

STM32CubeIDE Installation

1. Click **Next >**

STM32CubeIDE Installation

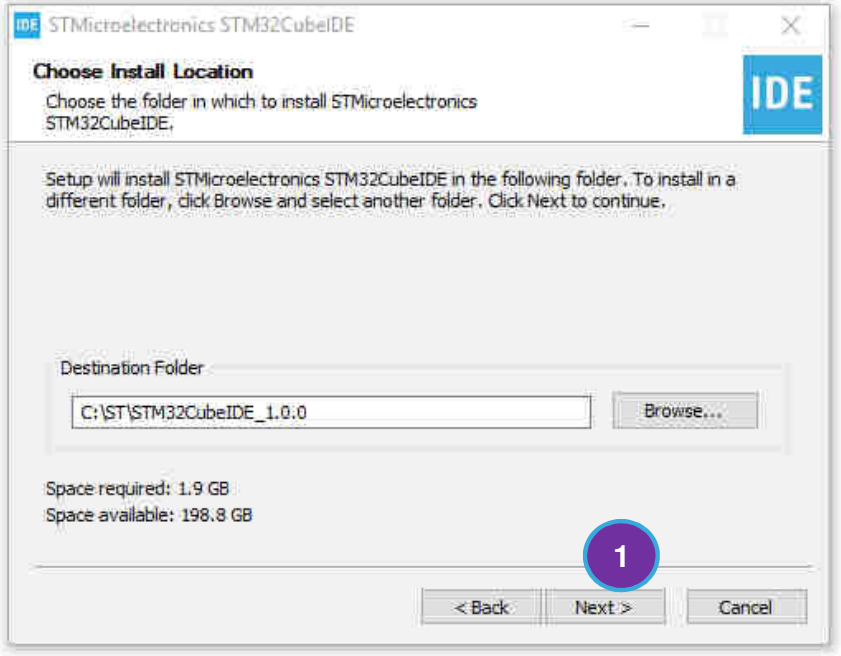
1. Click **I Agree**



life.augmented

STM32CubeIDE Installation

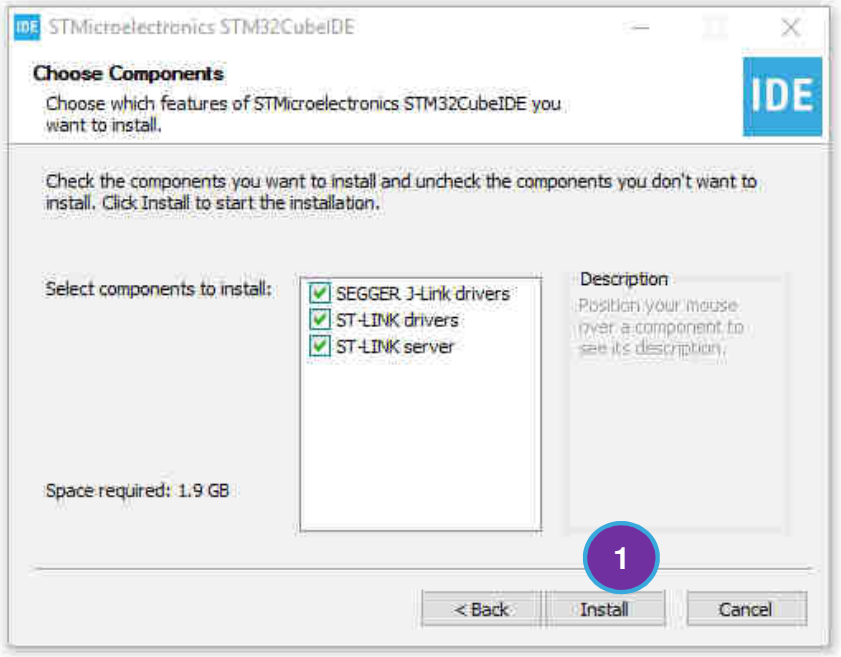
1. Click **Next >**



life.augmented

STM32CubeIDE Installation

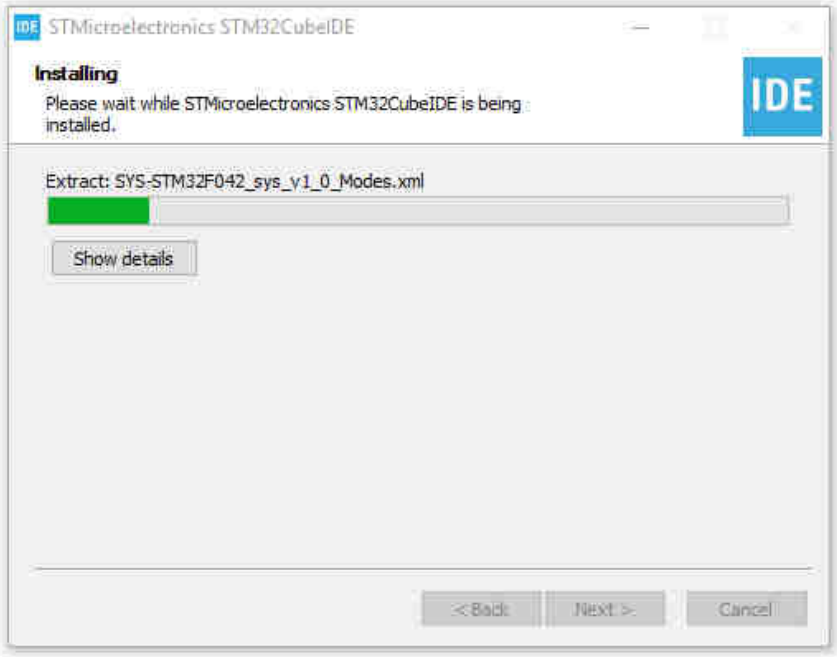
1. Click **Install**



life.augmented

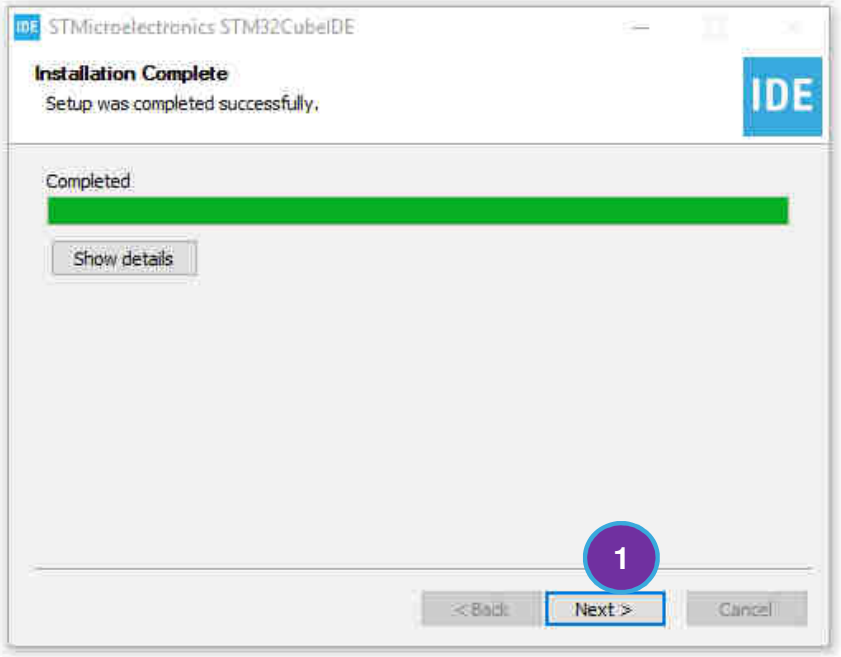
STM32CubeIDE Installation…

1. Wait until installation is completed



…STM32CubeIDE Installation

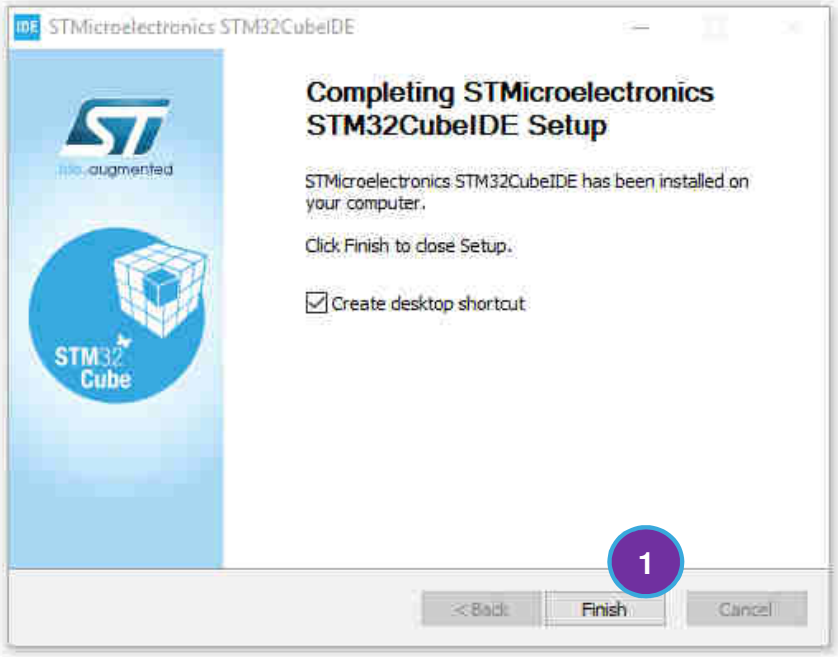
1. Click **Next >**



life.augmented

STM32CubeIDE Installation

1. Click **Finish**

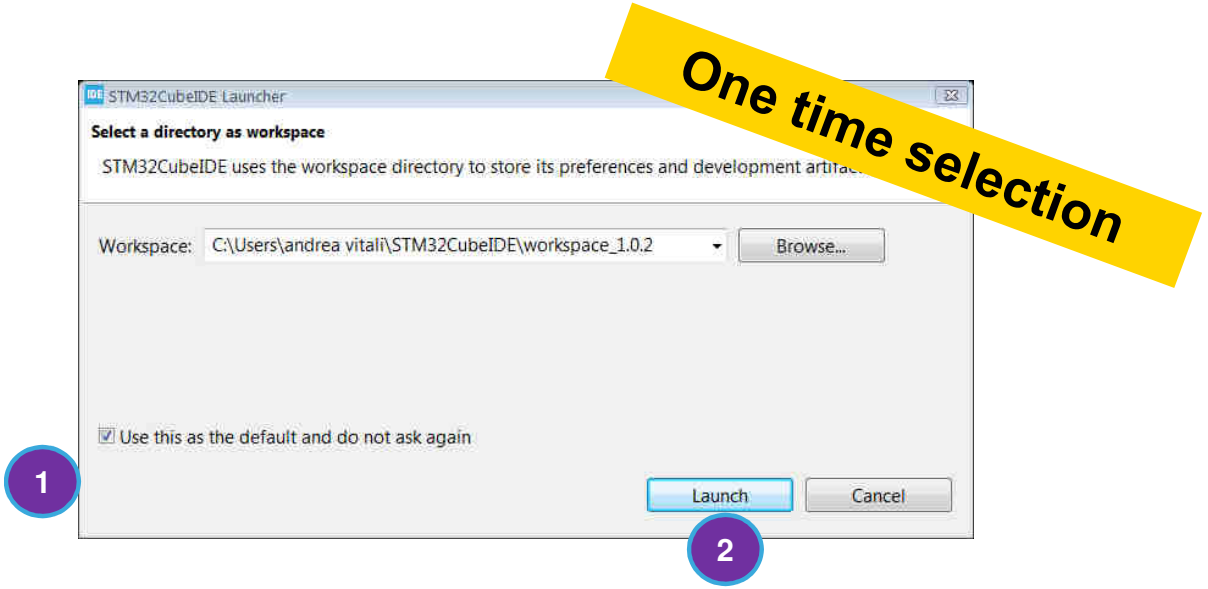


STM32CubeIDE Configuration

1. Open **STM32CubeIDE** by double clicking the icon on your desktop
2. Wait until **STM32CubeIDE** is loading

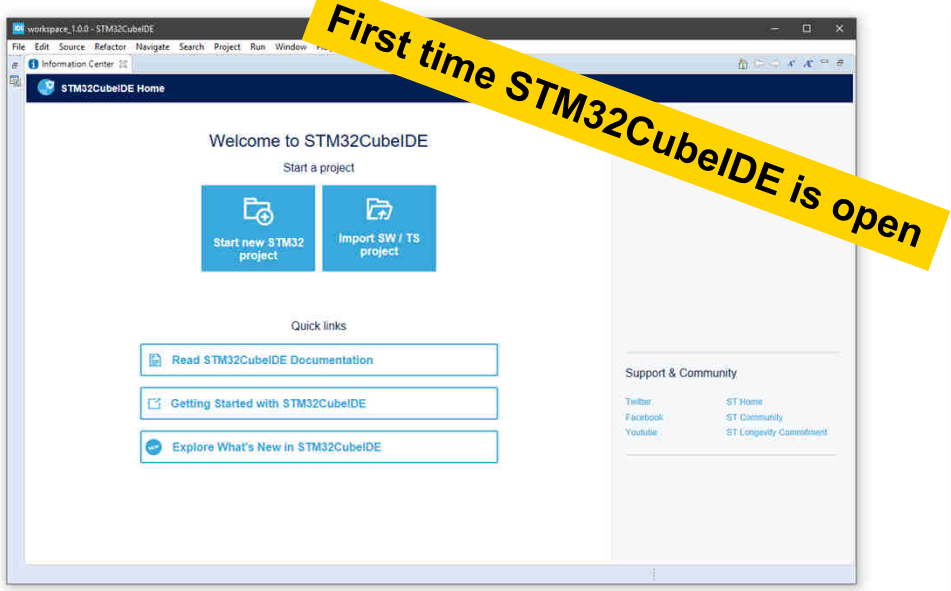
STM32CubeIDE Configuration

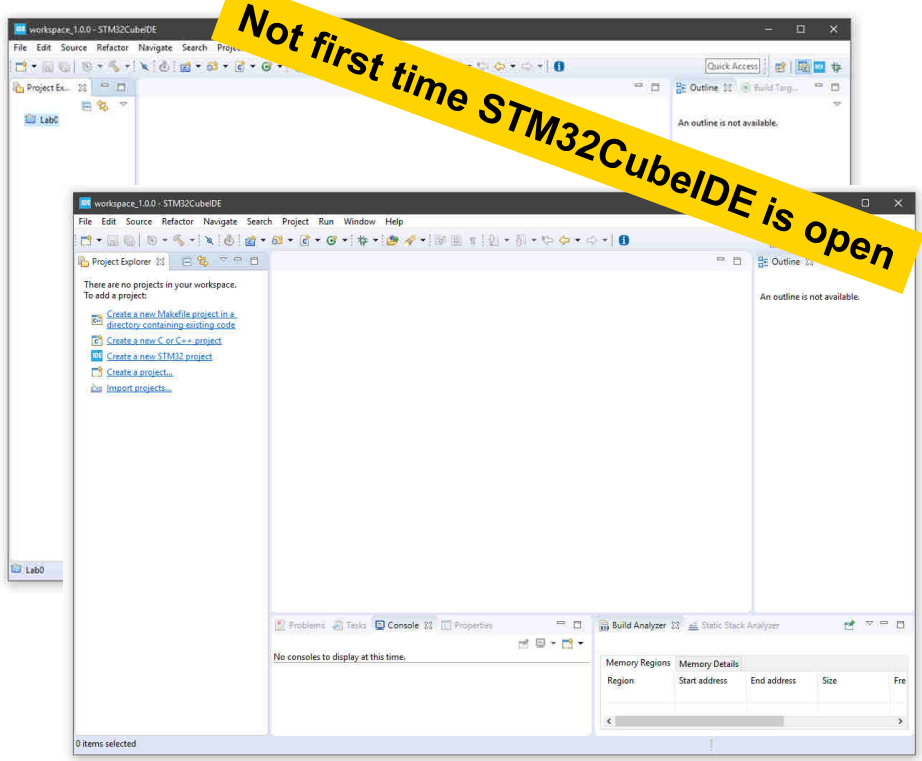
1. Check **Use this as the default and do not ask again**
2. Click **Launch**



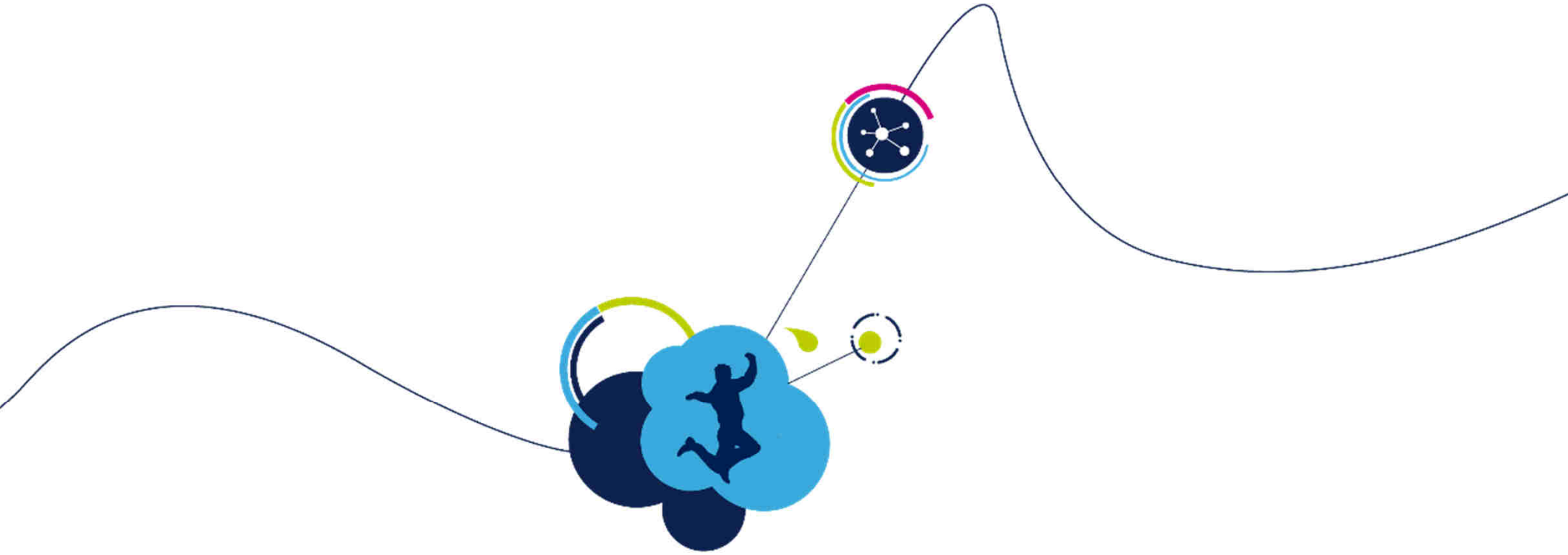
STM32CubeIDE Configuration

The main interface of STM32CubeIDE will appear differently, depending if it is the first time it has been run or not:





life.augmented



X-Cube-MEMS1 Installation

life.augmented

X-Cube-MEMS1 Installation

X-Cube-MEMS1 can be downloaded from STM32CubeIDE

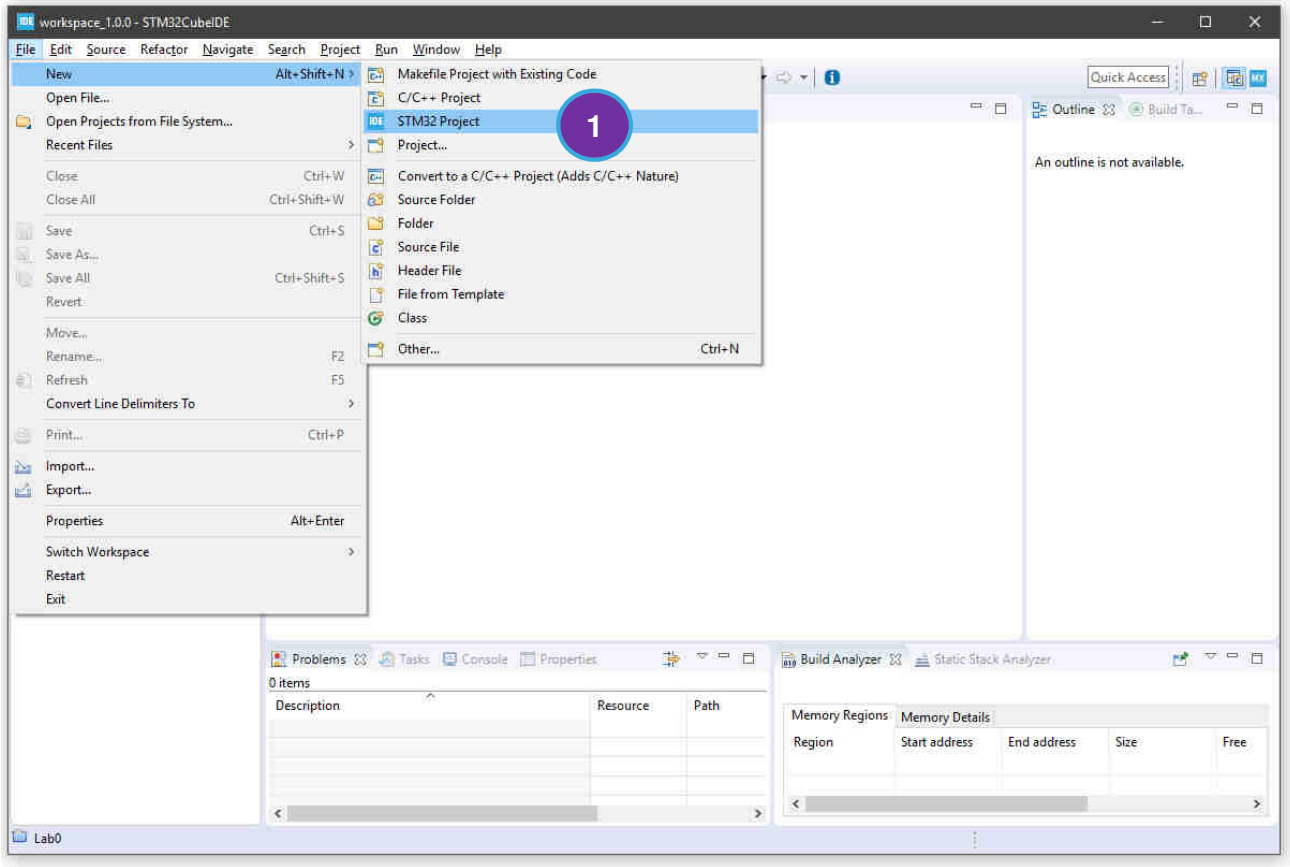
- X-Cube downloads can be managed **only while a project is opened**
  - We will create a **temporary project** which will be deleted at the end of the installation, this is also an opportunity to test the STM32CubeIDE installation
- Cube downloads may be triggered if updates are available and deemed necessary
  - Currently STM32CubeIDE 1.0.2 needs to download **STM32Cube_FW_I4_v1140.zip** package

- **NOTE**: If you are in a VPN and/or behind a proxy that requires authentication, you may need to change the network connection setting. You will find details the relevant information in the section “Network Setting Configuration”
  - It is always possible to configure network connection settings in Window > Preferences then in General > Network Connections

life.augmented

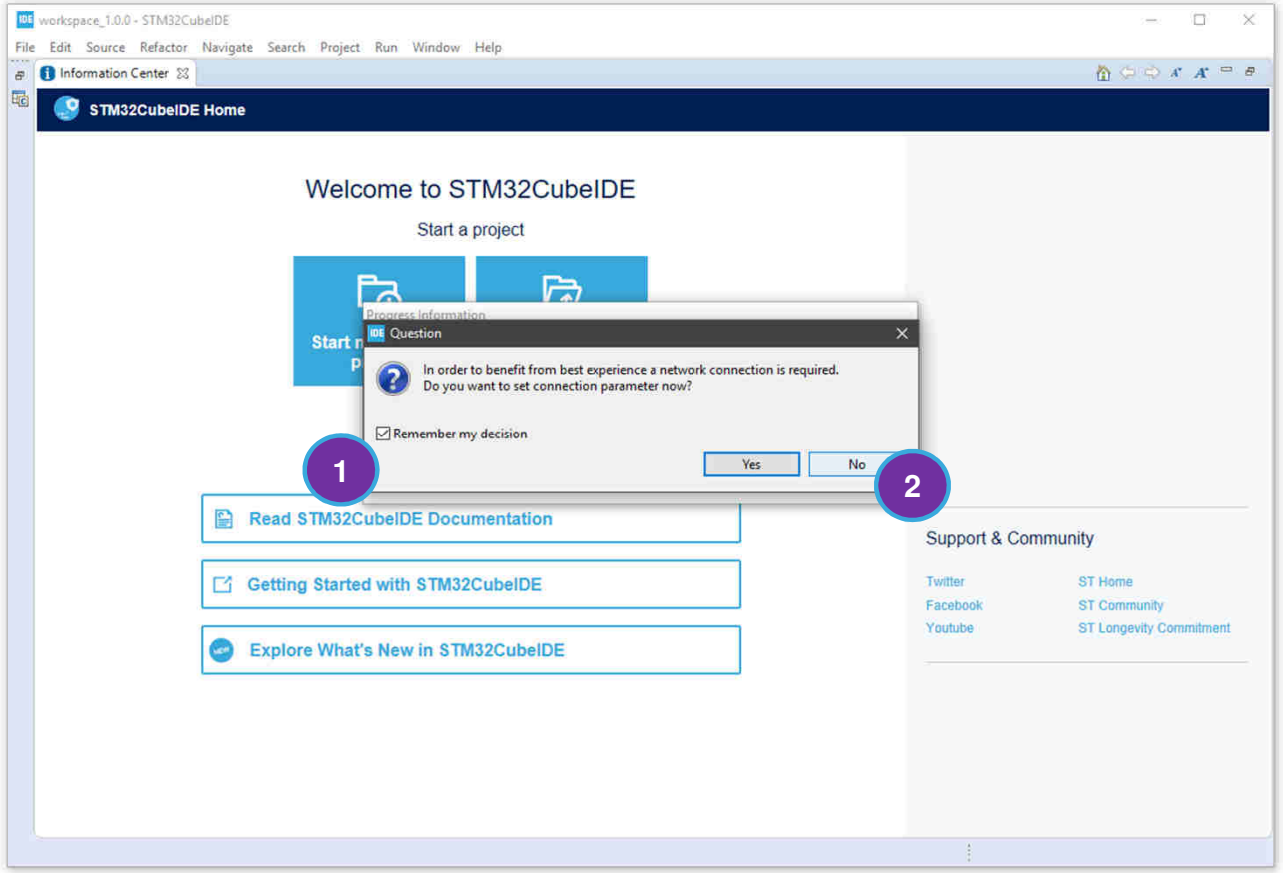
X-Cube-MEMS1 Installation

1. Click on **File > New > STM32 Project**



X-Cube-MEMS1 Installation

1. Check **Remember my decision***
2. Click **No***

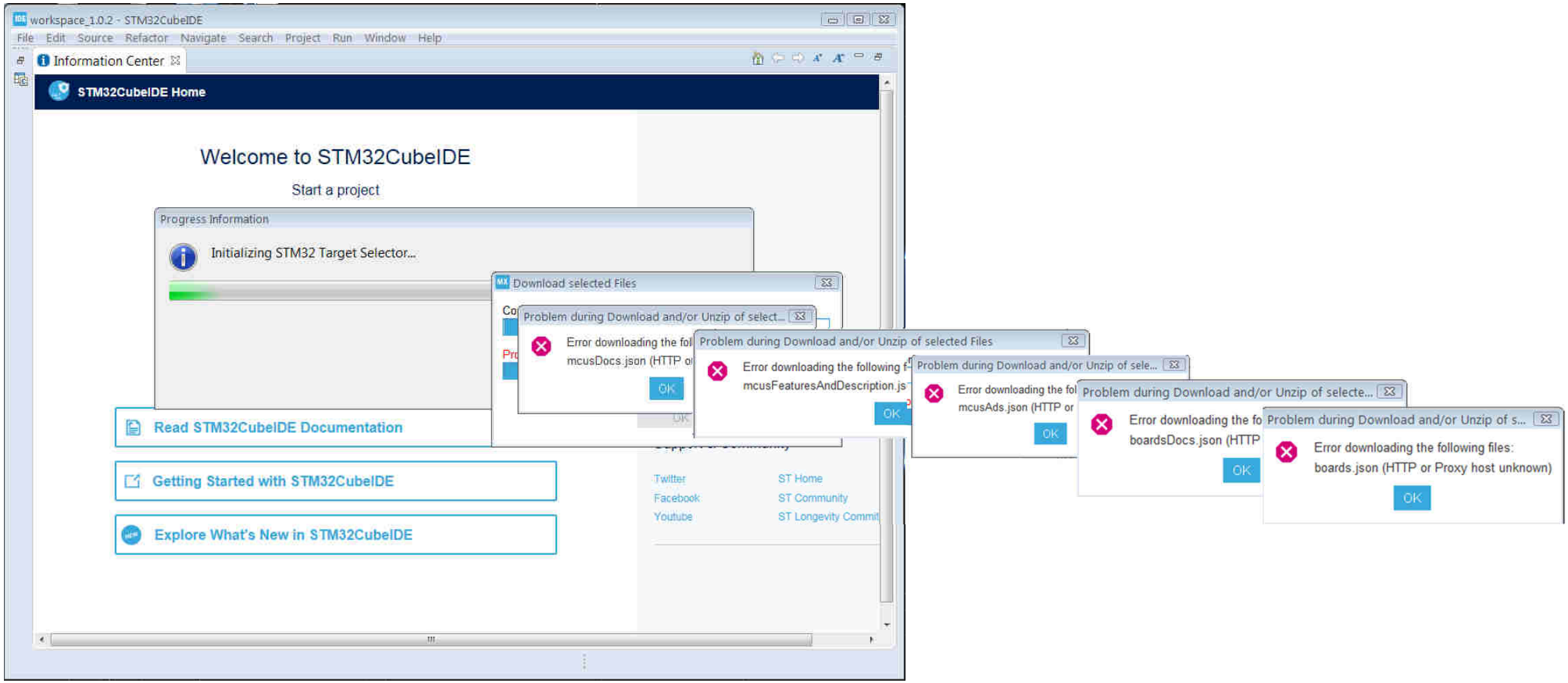


* It is always possible to reconfigure internet connection parameter in **Window > Preferences** then in **General > Network Connections**

life.augmented

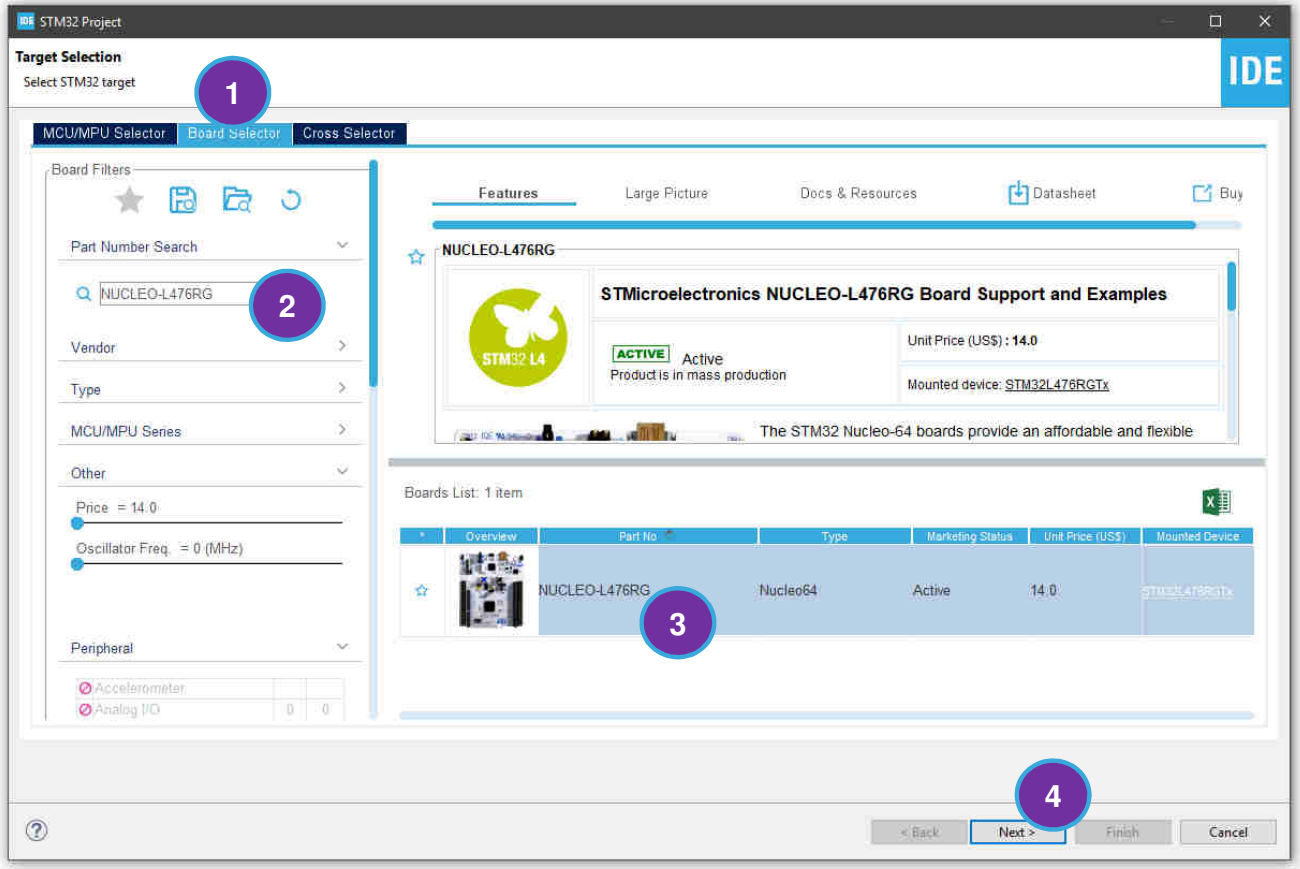
X-Cube-MEMS1 Installation

If you answer Yes and network settings are wrong, required downloads will not be completed. Just click **Ok** and ignore it. We can fix this later.



X-Cube-MEMS1 Installation

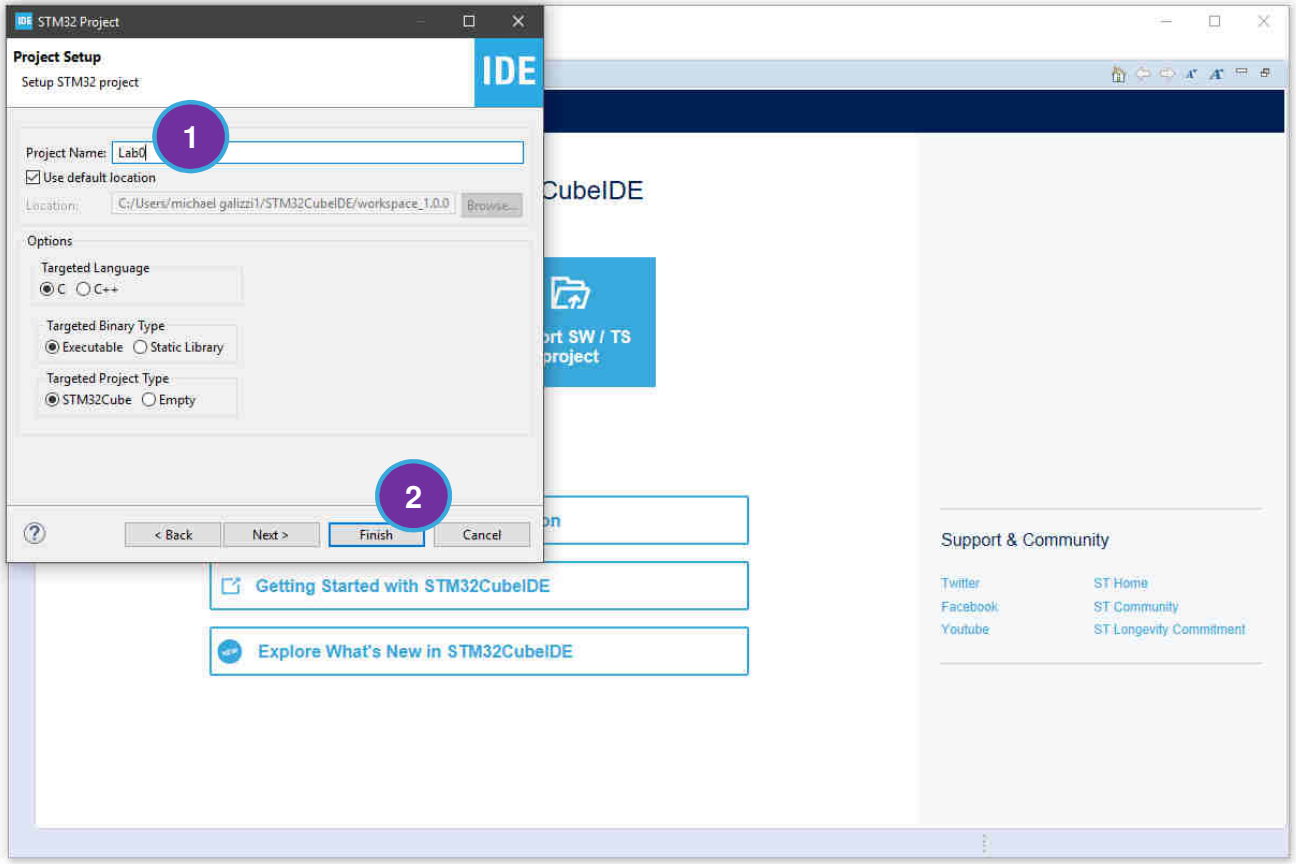
1. Click on **Board Selector**
2. Type **NUCLEO-L476RG**
3. Click on the board
4. Click **Next >**



life.augmented

X-Cube-MEMS1 Installation

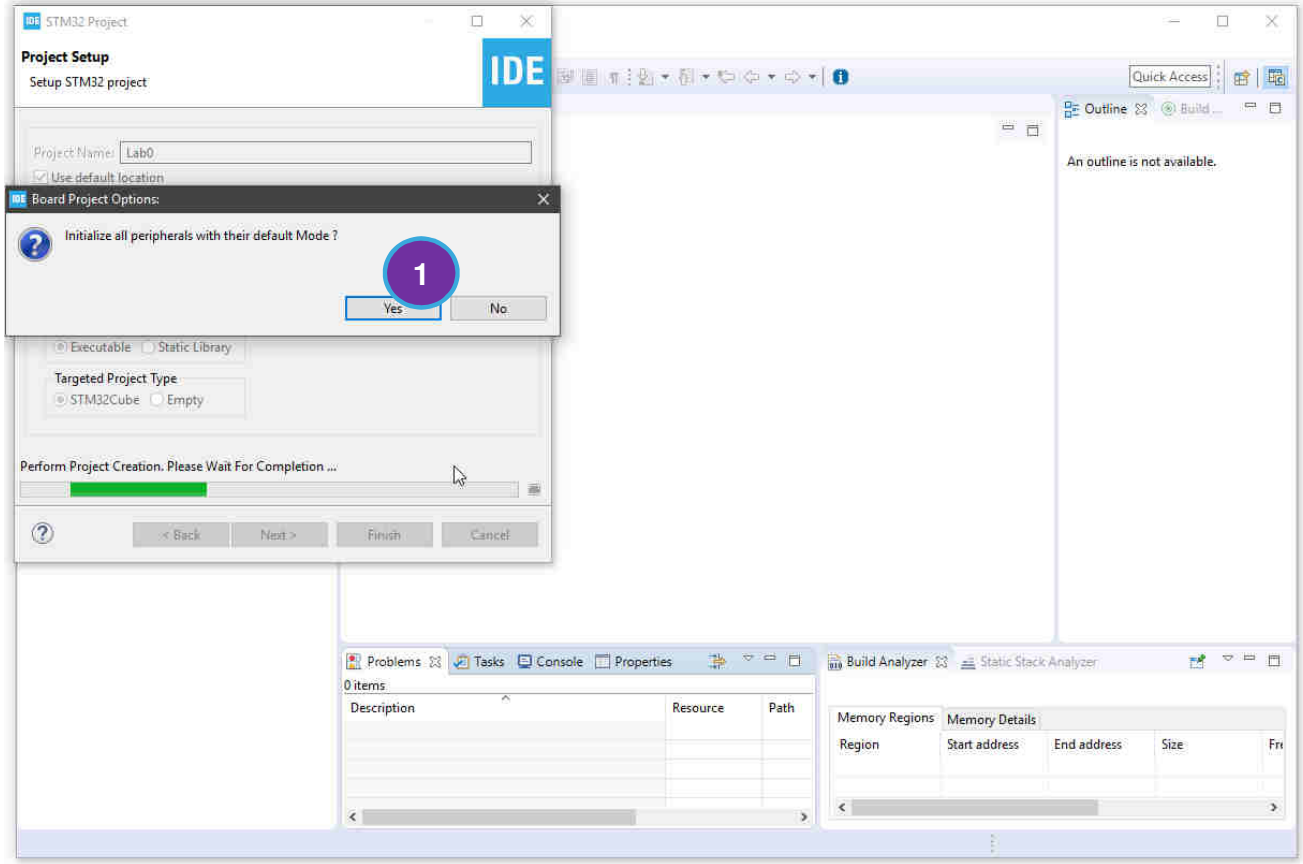
1. Project Name **Lab0** (any name is OK)
2. Click **Finish**



life.augmented

X-Cube-MEMS1 Installation

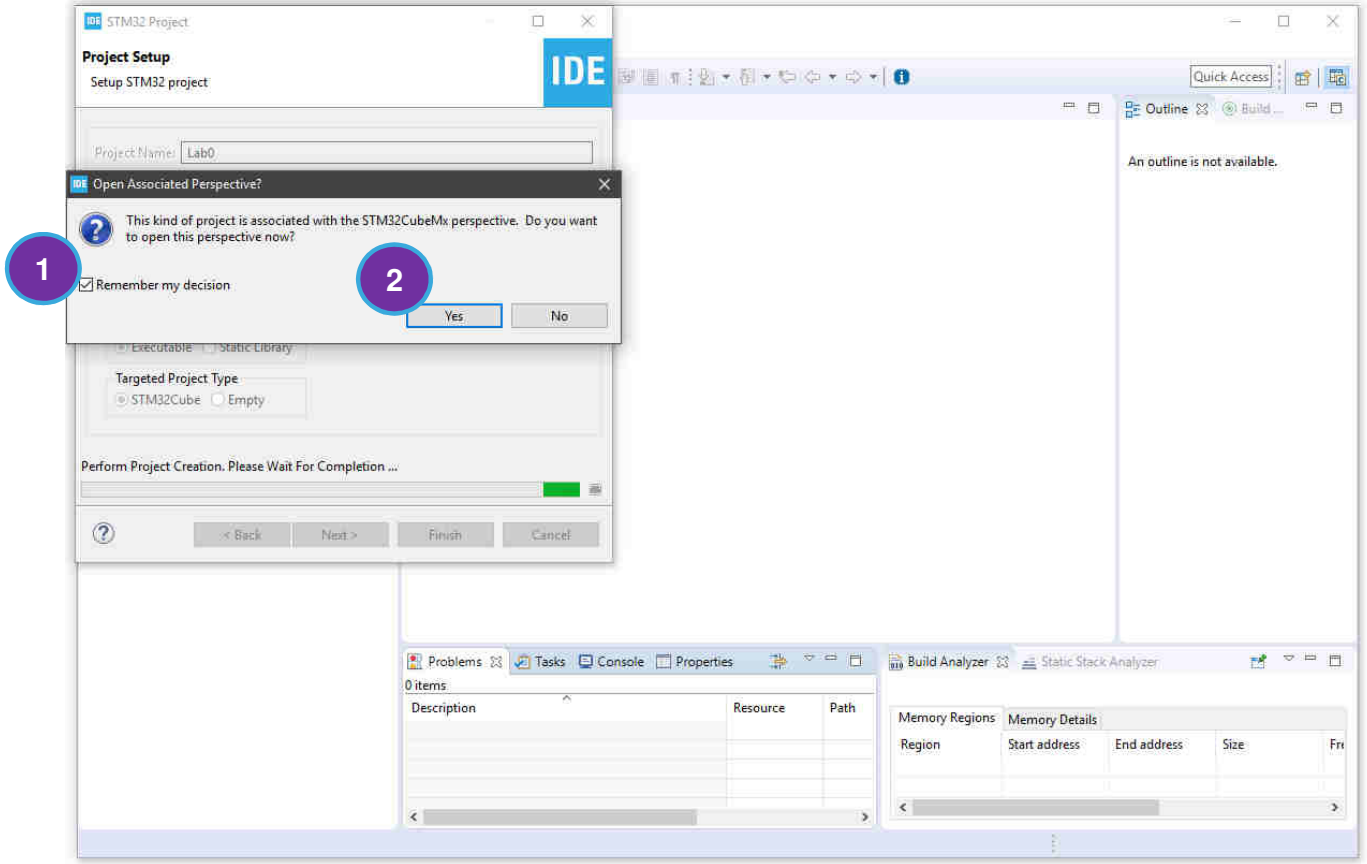
1. Click **Yes** to init peripherals in default mode



life.augmented

X-Cube-MEMS1 Installation

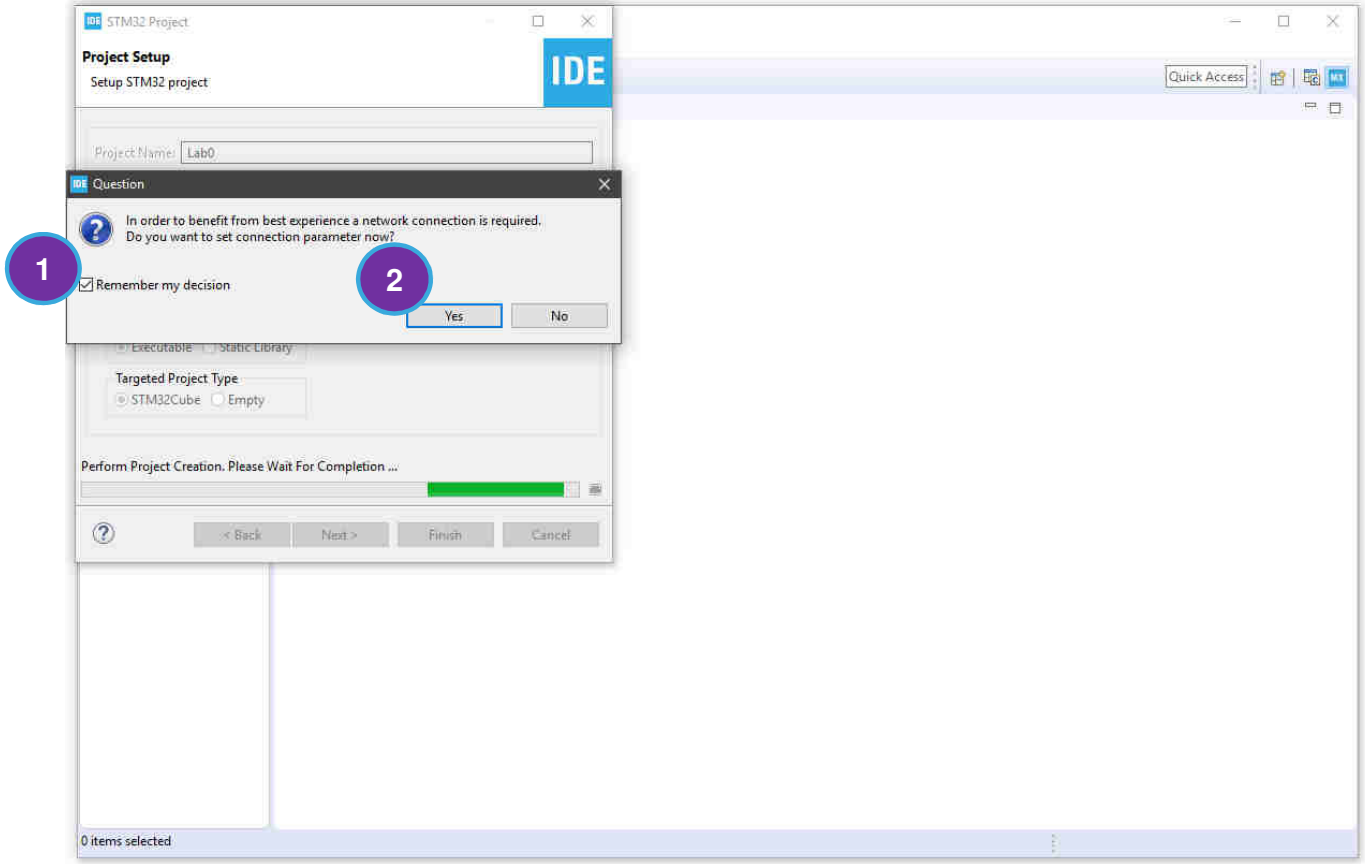
1. Check **Remember my decision**
2. Click **Yes** to open the project perspective



life.augmented

X-Cube-MEMS1 Installation

1. Check **Remember my decision**
2. Click **No***

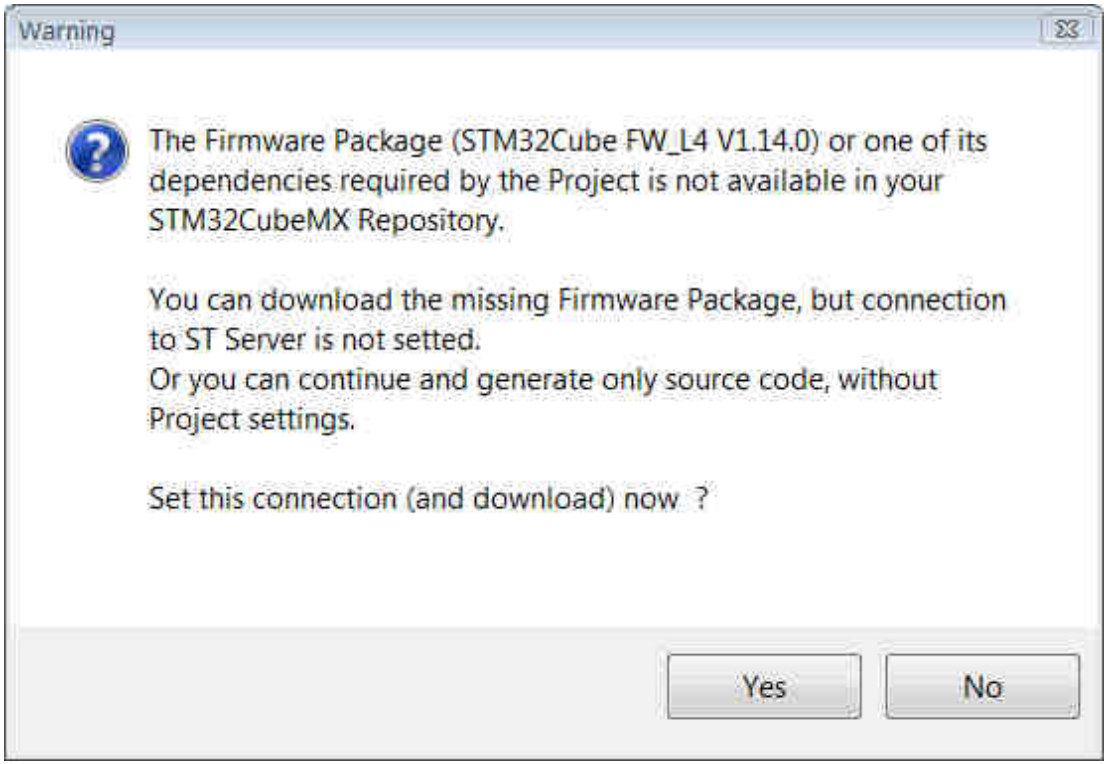


* It is always possible to reconfigure internet connection parameter in **Window > Preferences** then in **General > Network Connections**

life.augmented

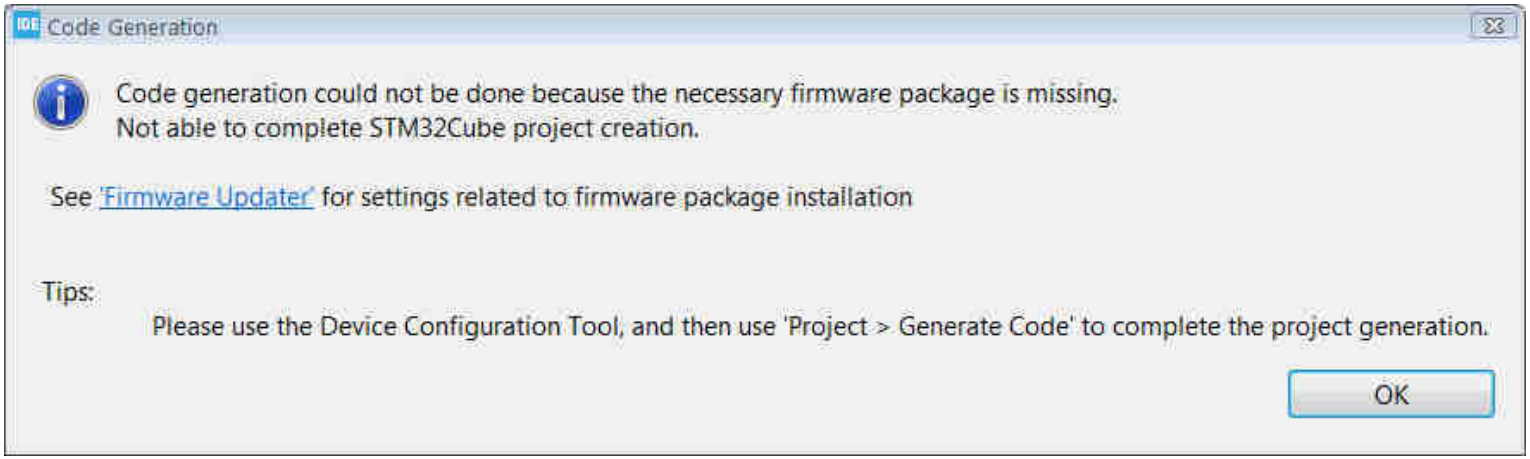
STM32CubeIDE Configuration

1. Click **No*** (we will download it later)



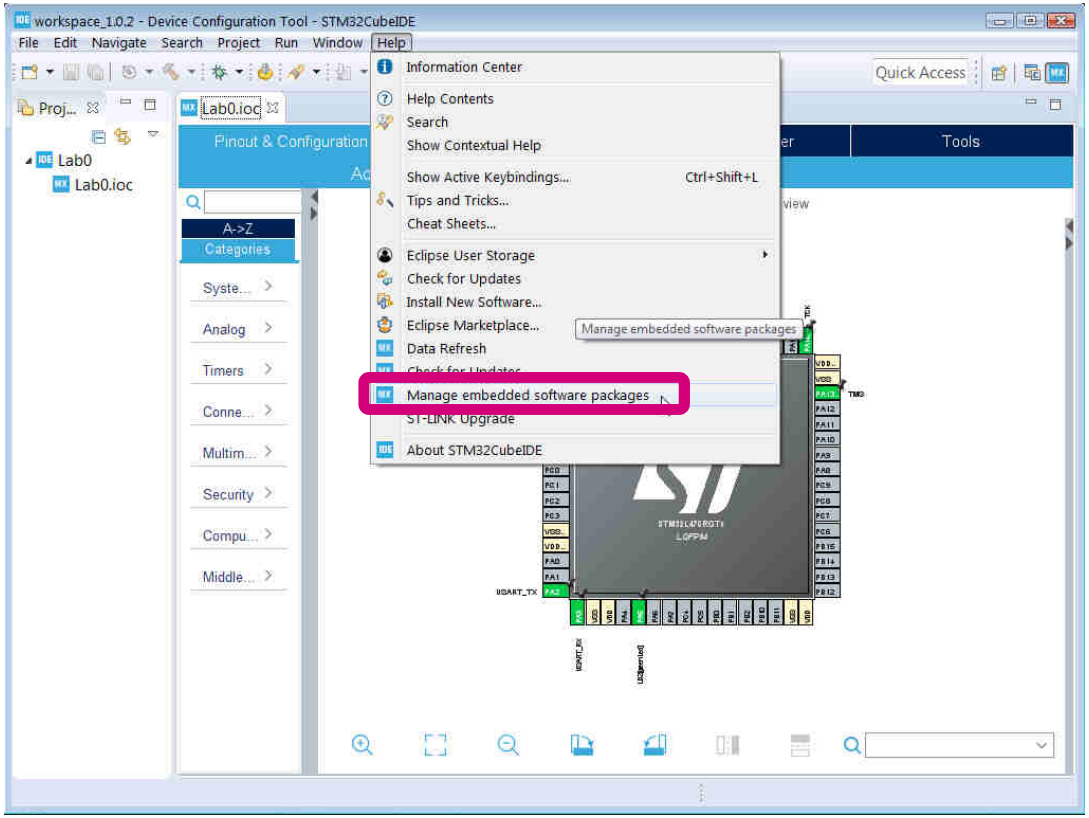
STM32CubeIDE Configuration

1. Click **Ok**



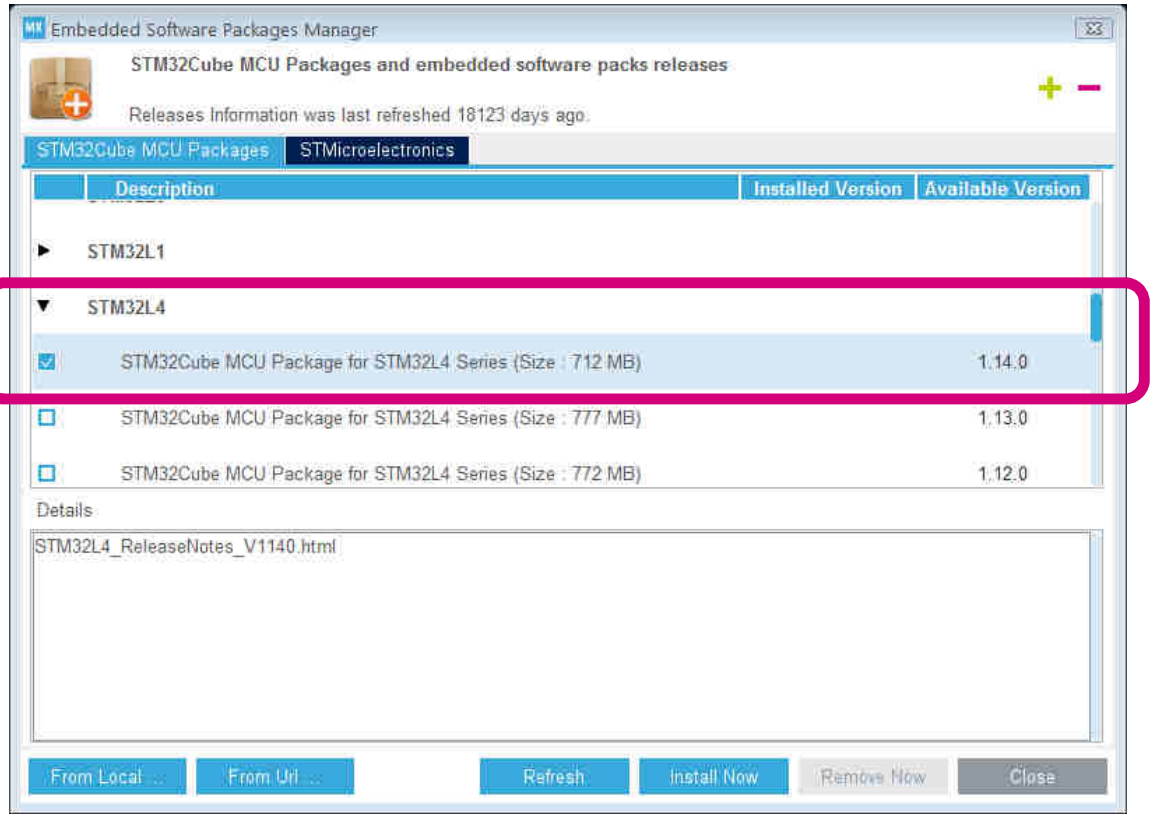
STM32CubeIDE Configuration

1. Click on **Help > Manage embedded software packages**



STM32CubeIDE Configuration

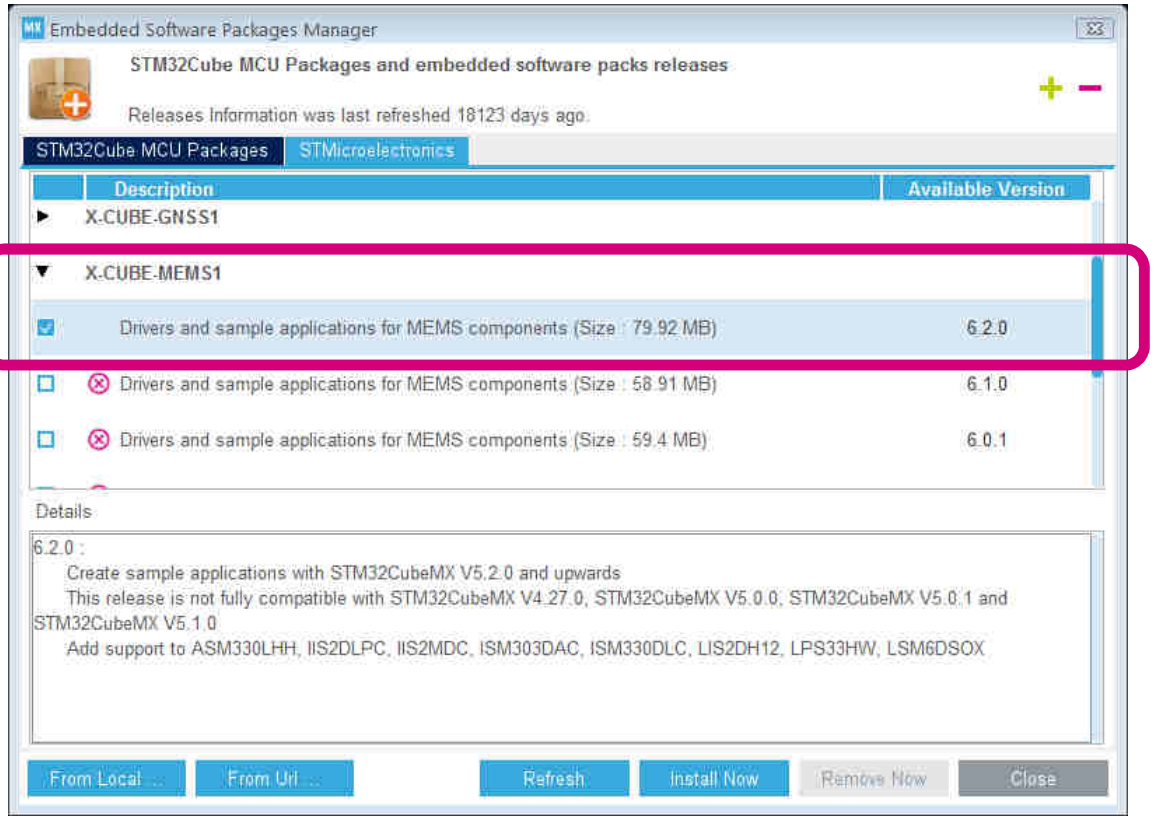
1. Scroll down to **STM32L4**, click to expand tree, select latest package **1.14.0**



life.augmented

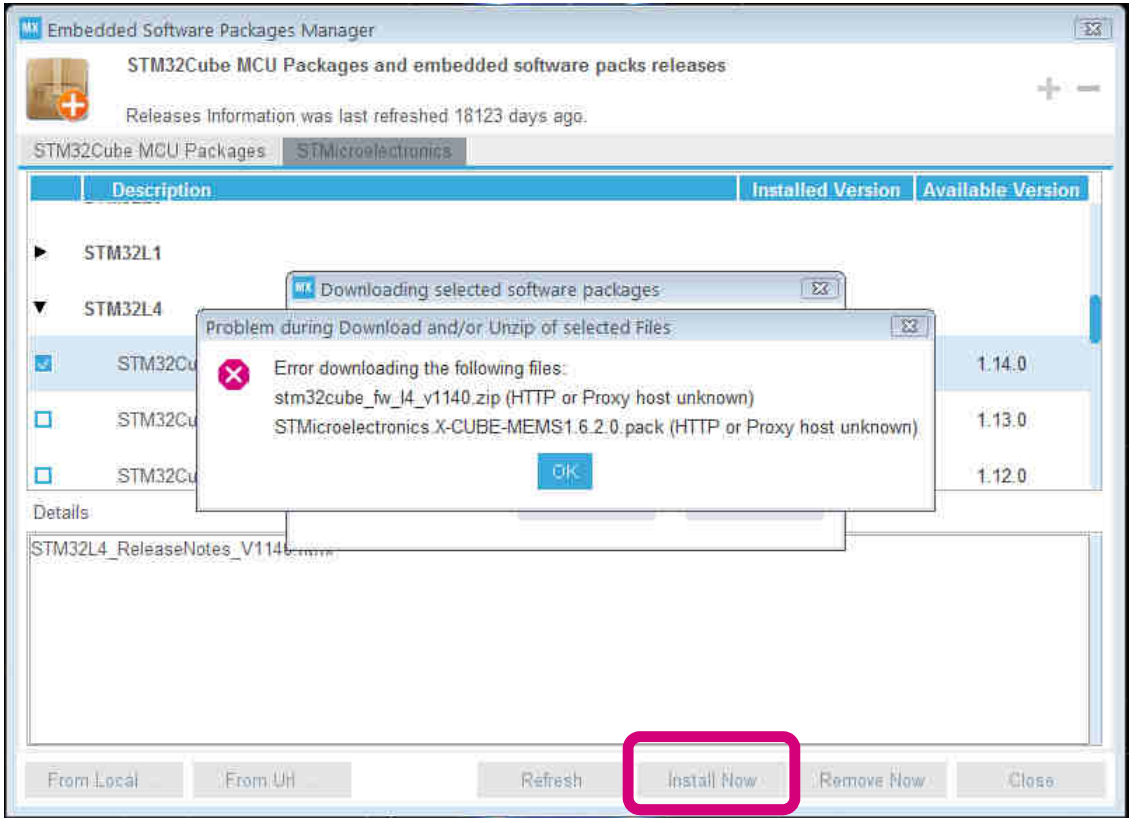
STM32CubeIDE Configuration

1. Scroll down to **X-CUBE-MEMS1**, click to expand, select latest package **6.2.0** (older packages are marked as not compatible with this IDE version)

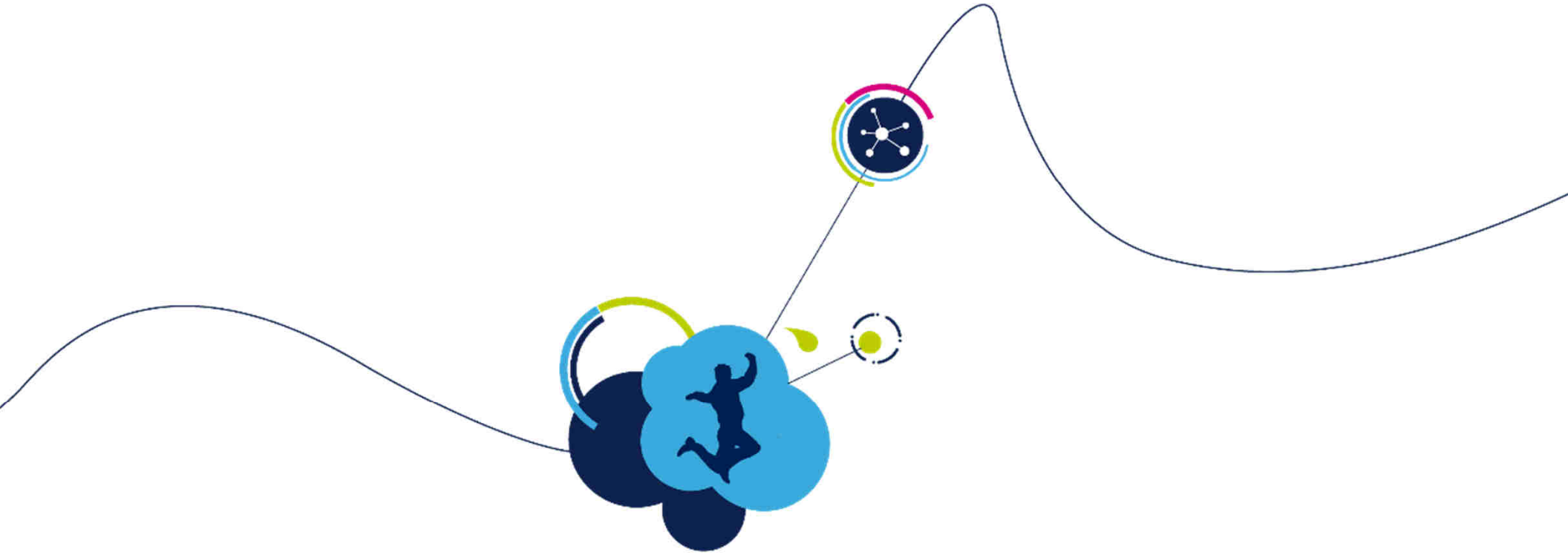


STM32CubeIDE Configuration

1. Click **Install Now**. If you get an error you must configure the network settings (see next slides).



life.augmented



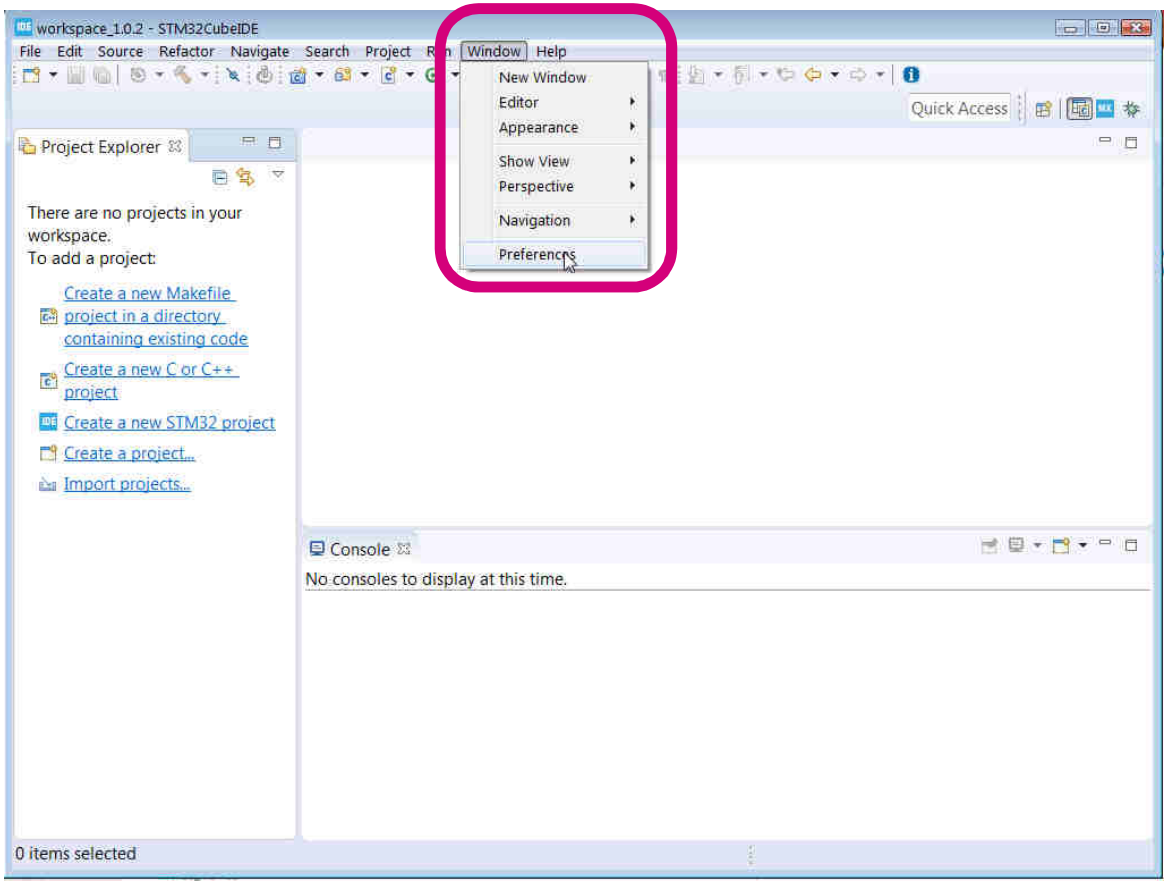
Network Setting Configuration

NOTE: this section applies only if you are in a VPN and/or behind a proxy that requires authentication. Otherwise you can go directly to the following section.

life.augmented

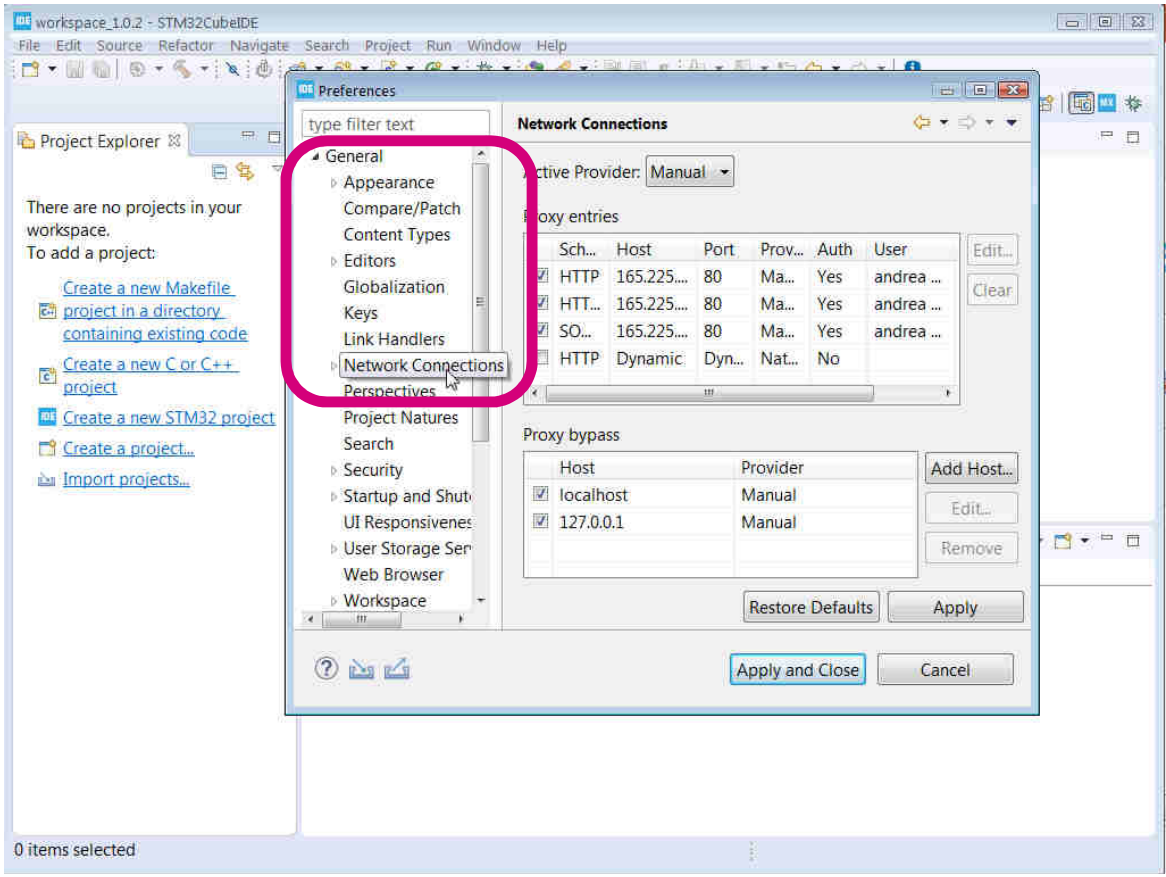
Network Setting Configuration

1. Click on **Window > Preferences**



Network Setting Configuration

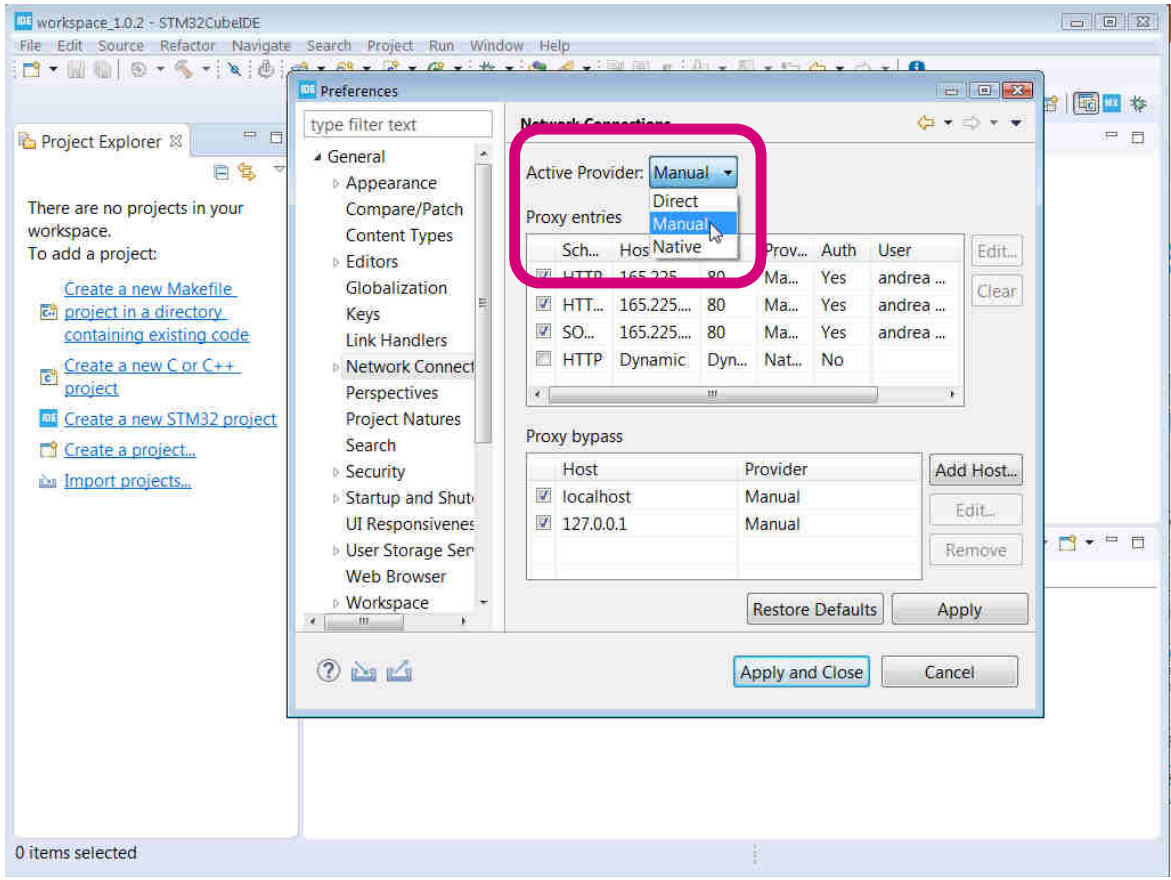
1. Click on **General > Network Connection**



life.augmented

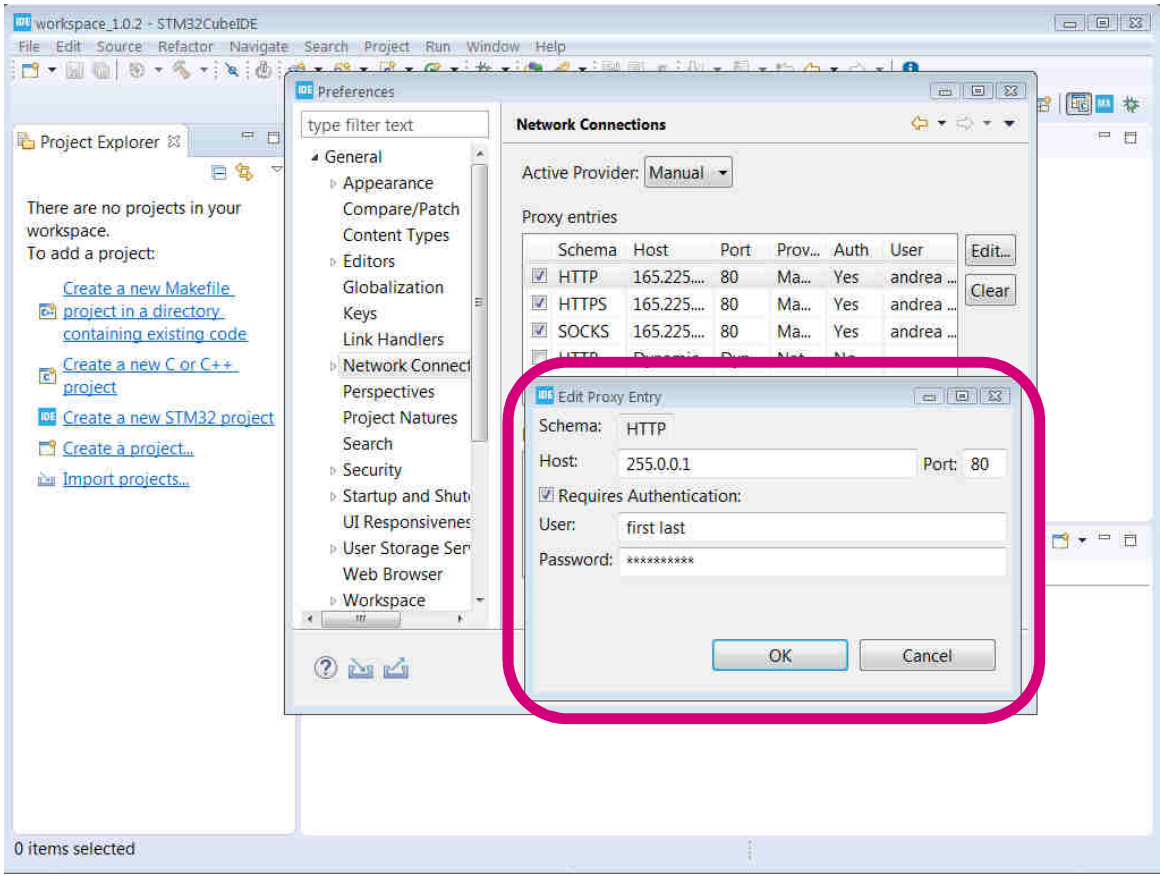
Network Setting Configuration

1. Change Active Provider to **Manual**

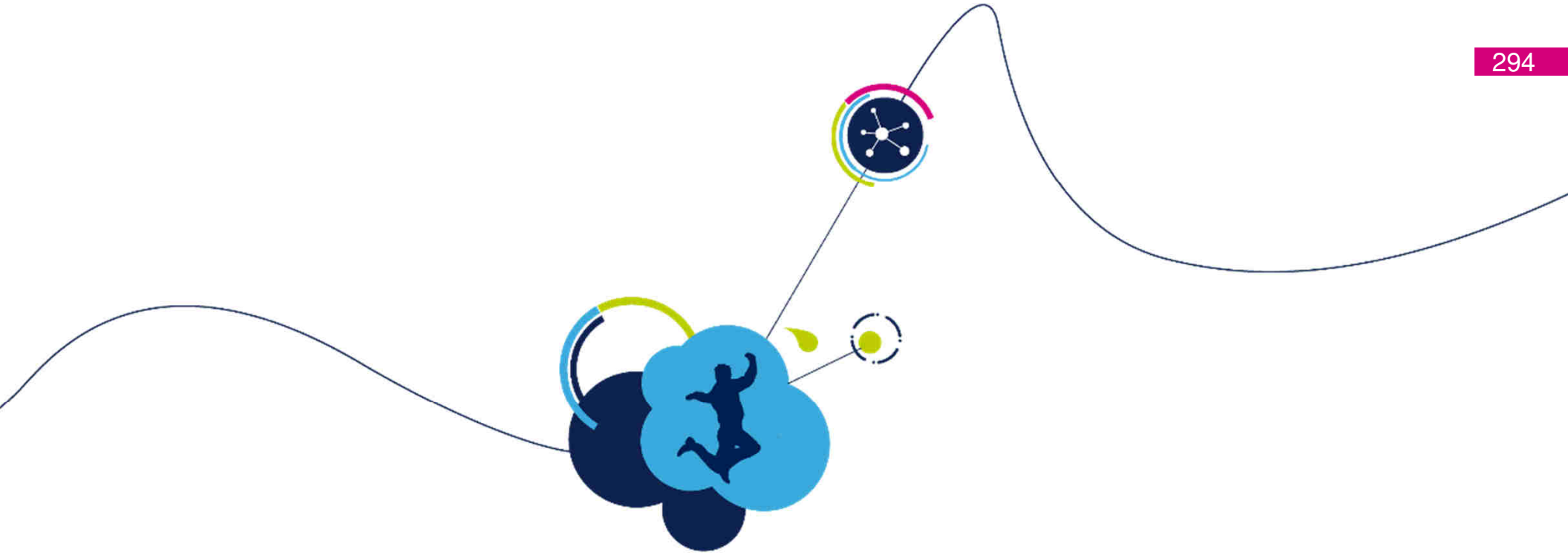


Network Setting Configuration

1. Select the protocol (**repeat for each protocol**, HTTP, HTTPS and SOCKS)
2. Press **Edit**
3. Insert proxy IP address and port number
4. Check “Requires Authentication” if needed, and insert user name and password
5. Click **OK**



life.augmented

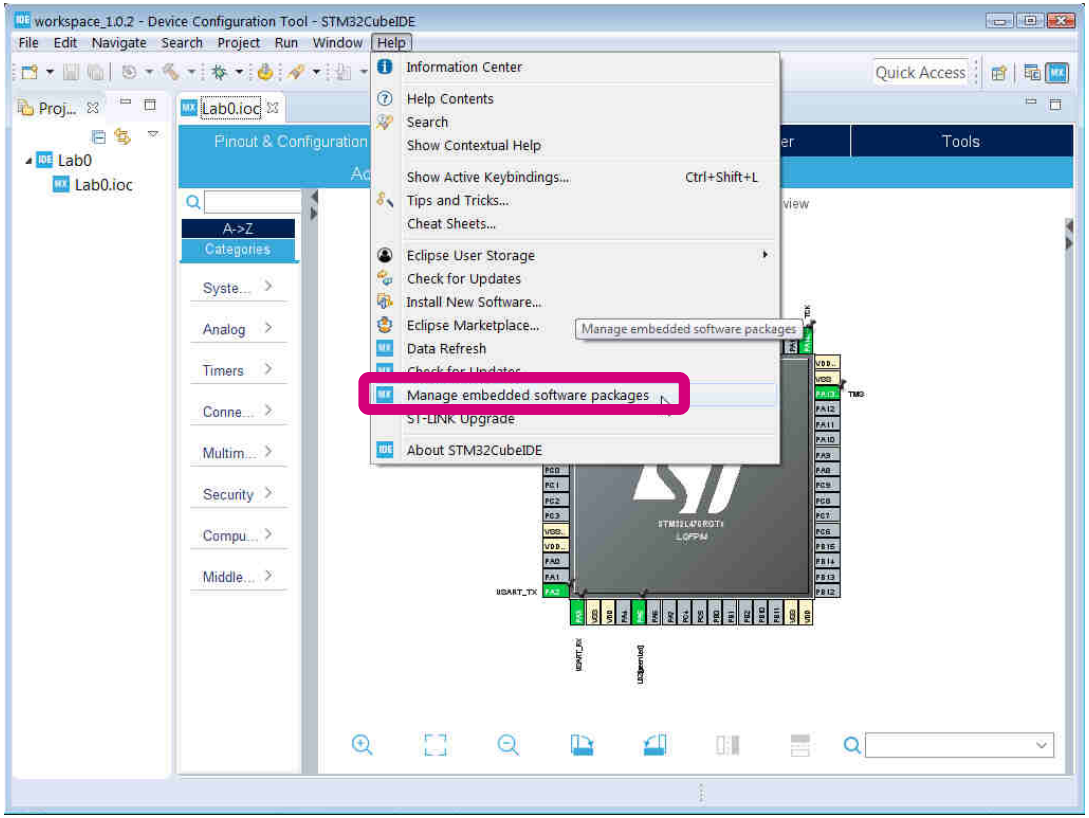


STM32CubeIDE Configuration

life.augmented

STM32CubeIDE Configuration

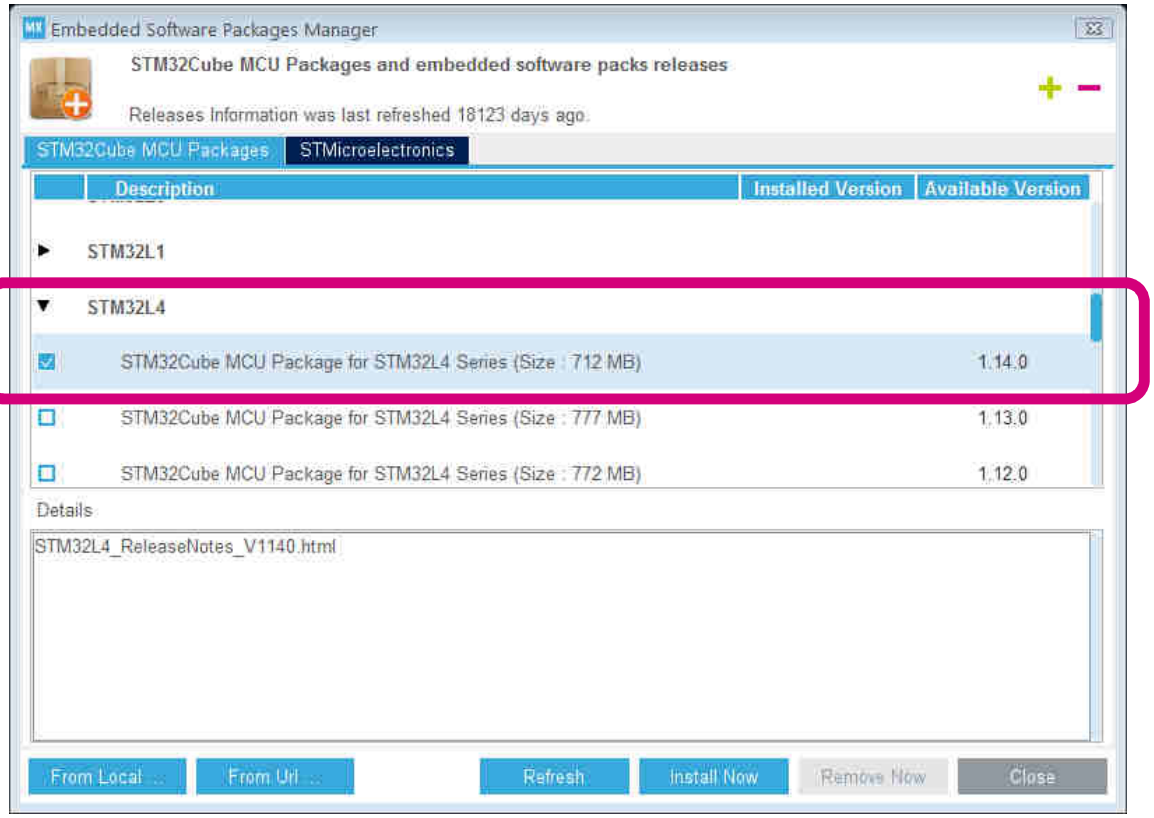
1. Click on **Help > Manage embedded software packages**



life.augmented

STM32CubeIDE Configuration

1. Scroll down to **STM32L4**, click to expand tree, select latest package **1.14.0**

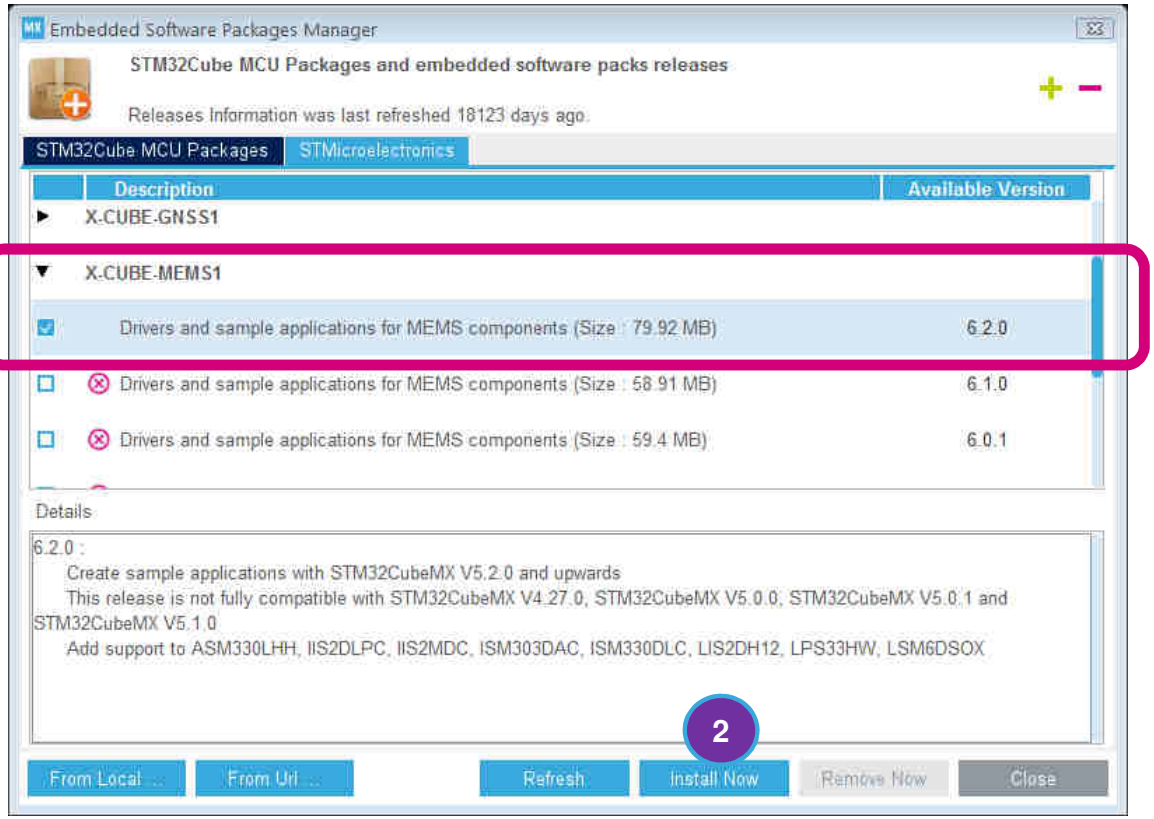


life.augmented

STM32CubeIDE Configuration

1. Scroll down to **X-CUBE-MEMS1**, click to expand, select latest package **6.2.0** (older packages are marked as not compatible with this IDE version)

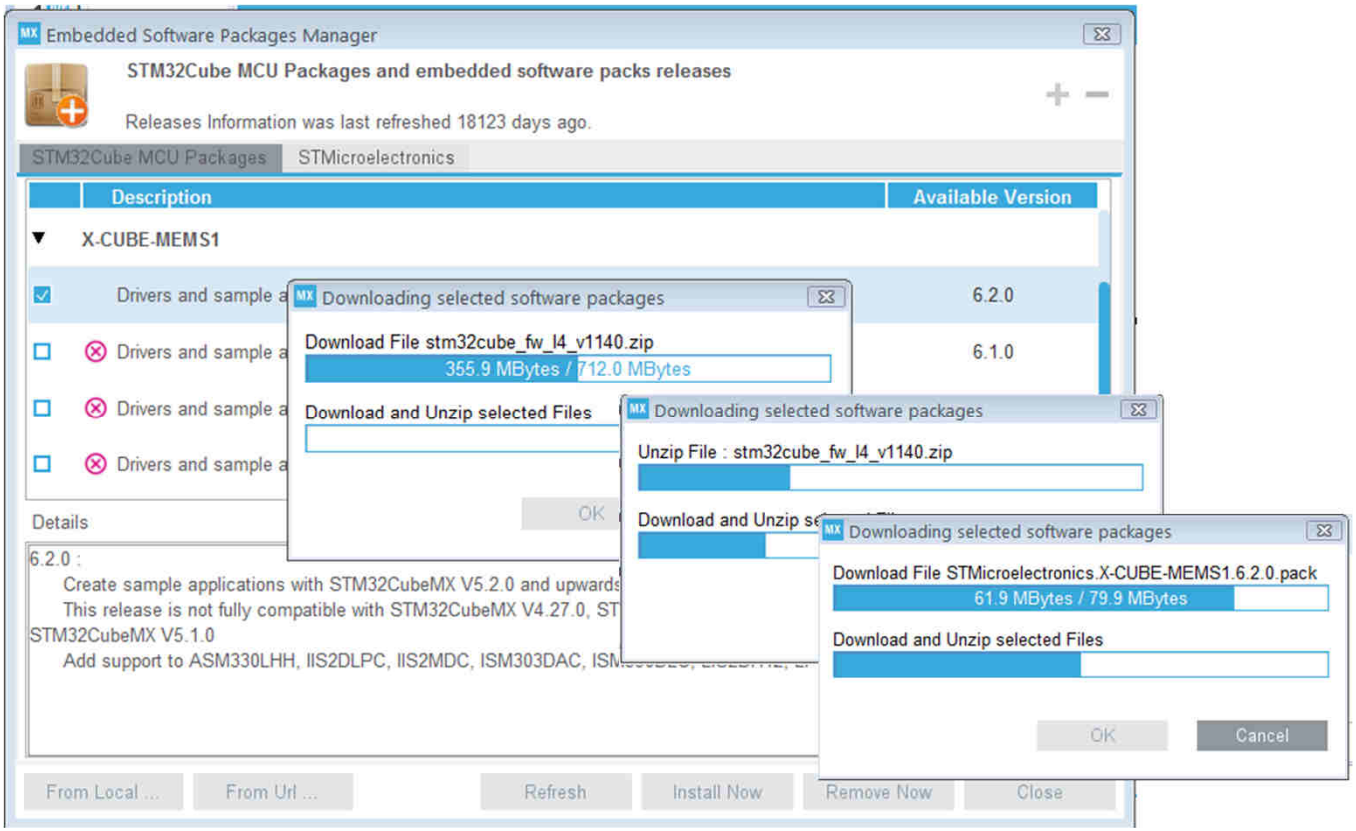
2. Click **Install Now**

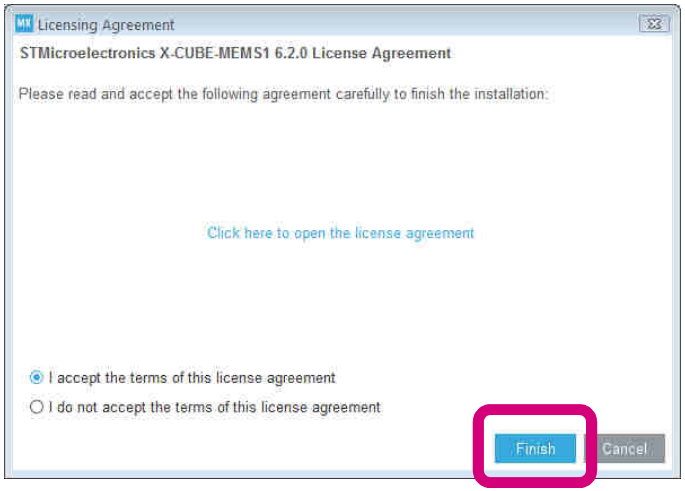


life.augmented

STM32CubeIDE Configuration

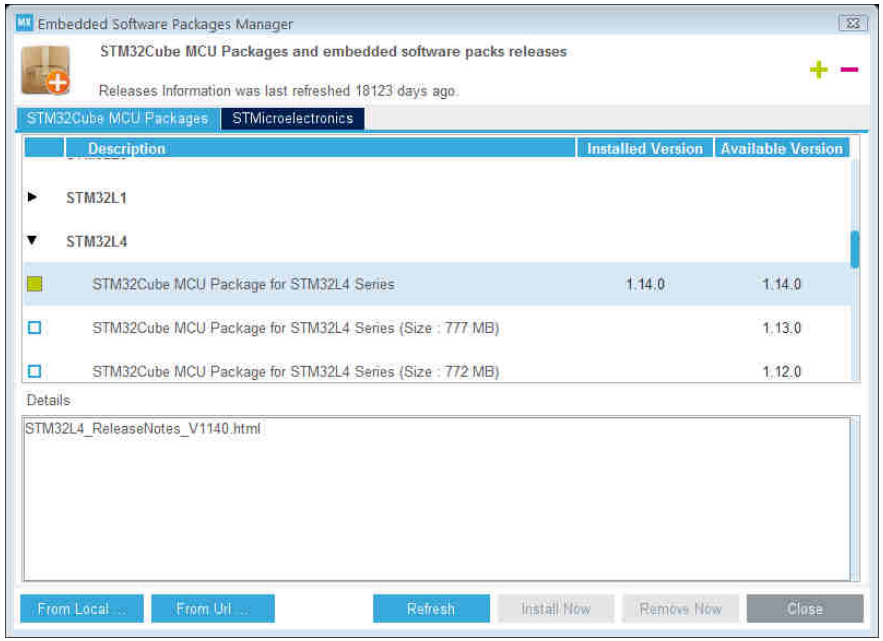
1. Wait for the download and unzip to be completed. Accept terms and conditions.

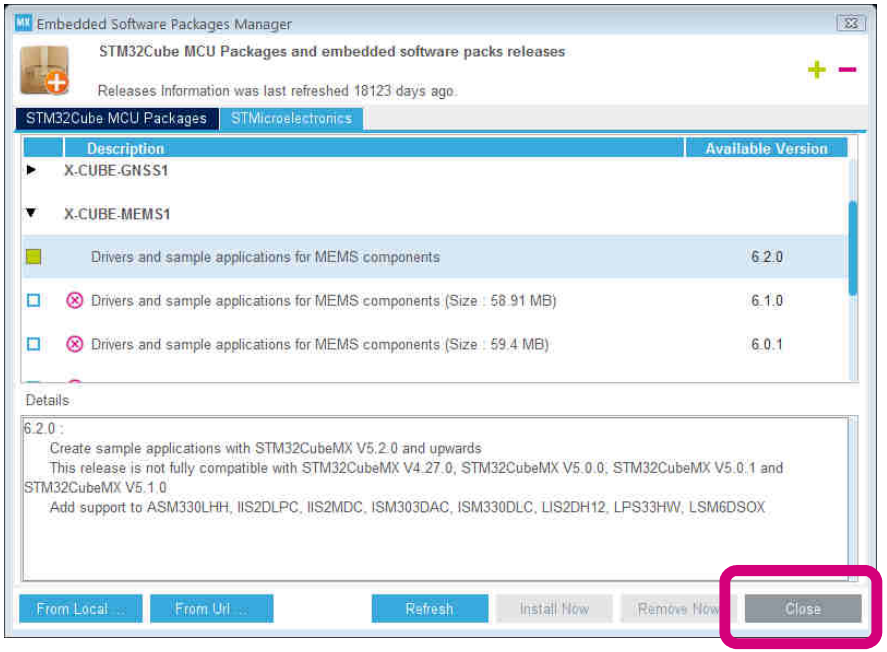




STM32CubeIDE Configuration

1. When STM32CUBE-L4 package is updated and X-CUBE-MEMS1 package is installed the corresponding checkboxs become green. **Close** window.

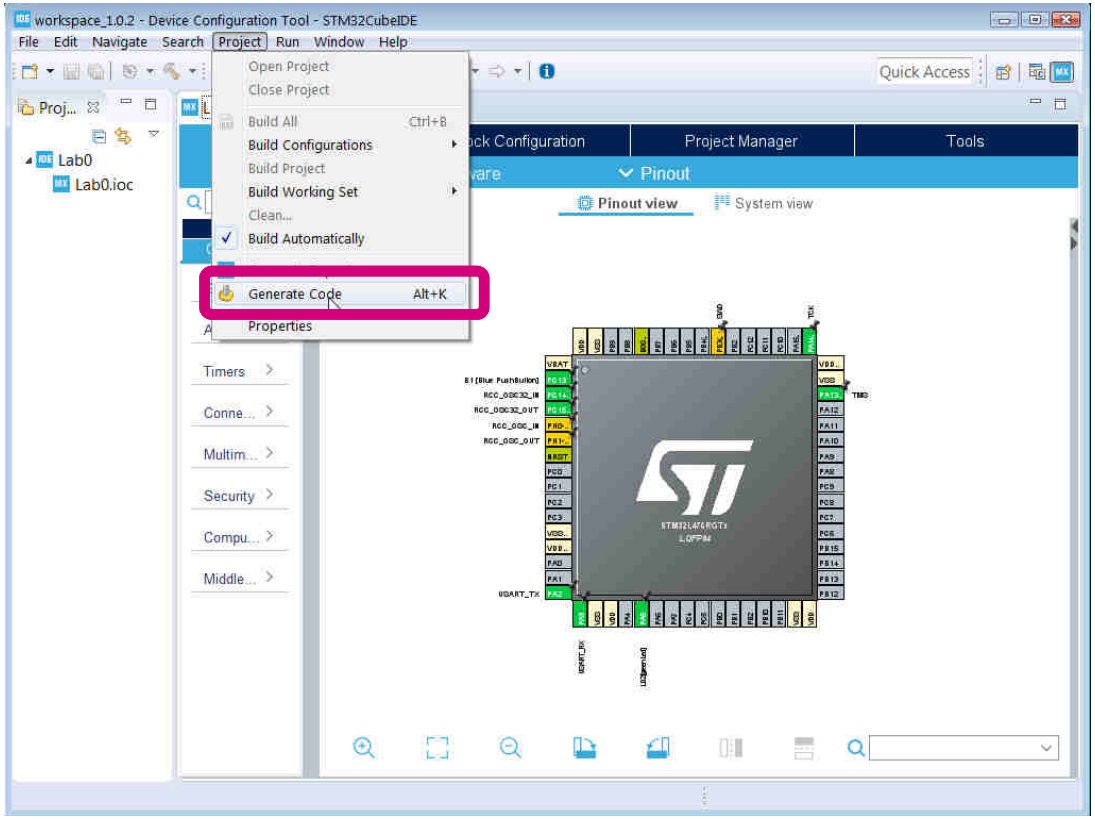




life.augmented

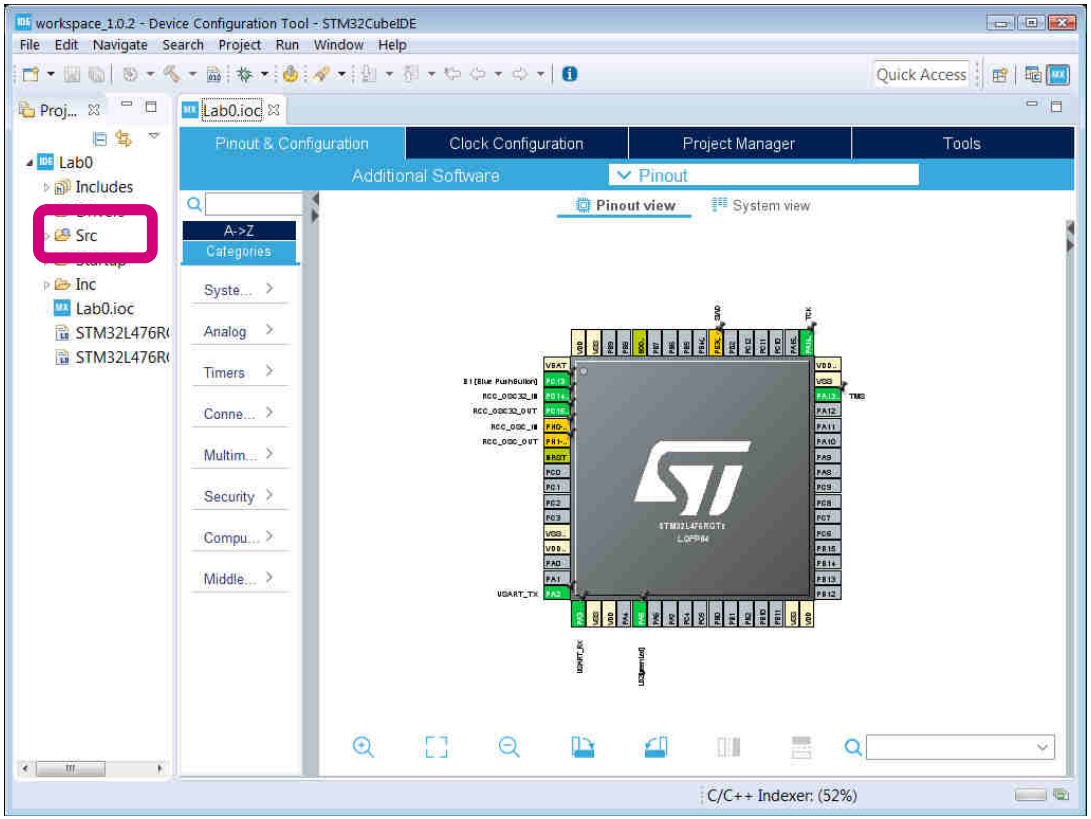
STM32CubeIDE Configuration

1. Let's check the code generator: **Project > Generate Code**



STM32CubeIDE Configuration

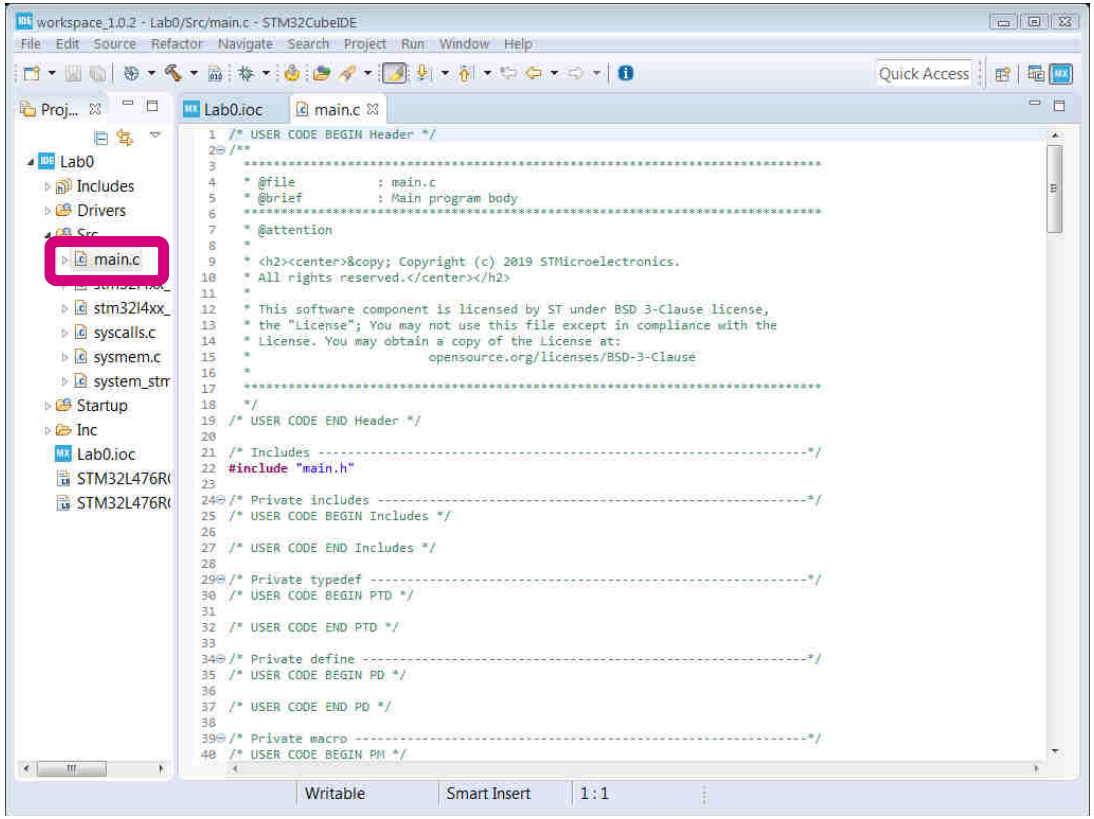
1. Let's have a look at the generated code: expand the **Src** folder.



life.augmented

STM32CubeIDE Configuration

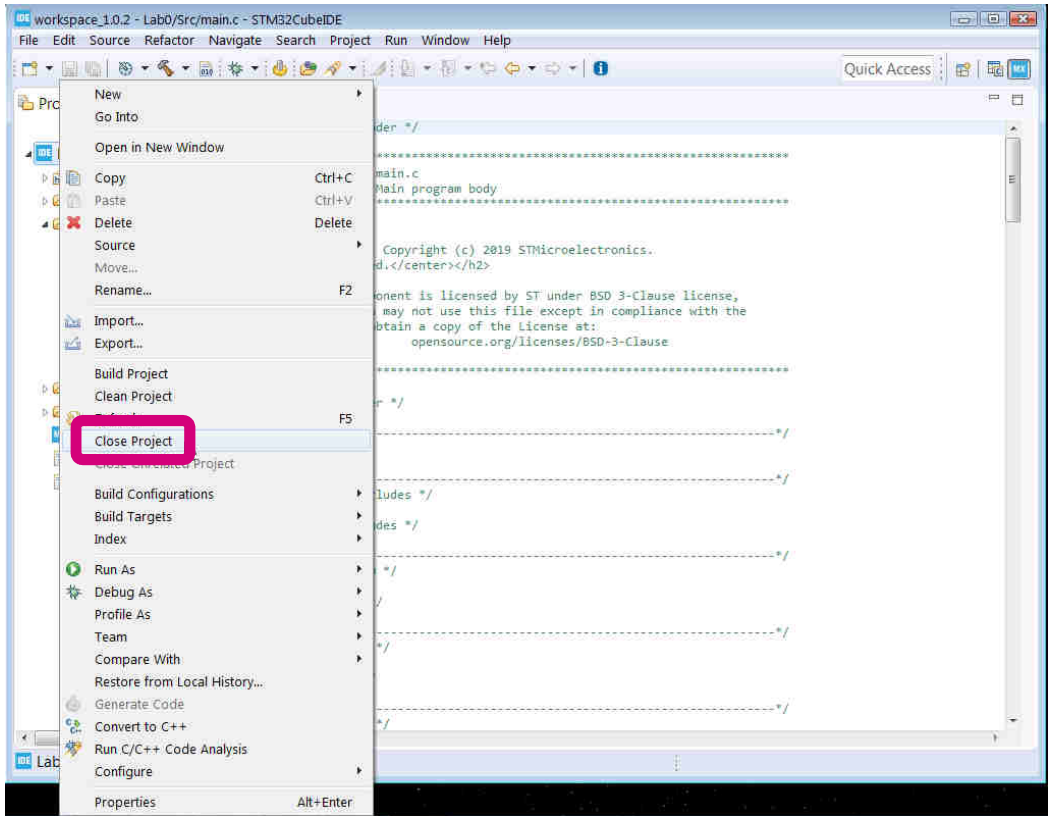
1. Let's have a look at the generated code: double click on **main.c** to open it.



life.augmented

STM32CubeIDE Configuration

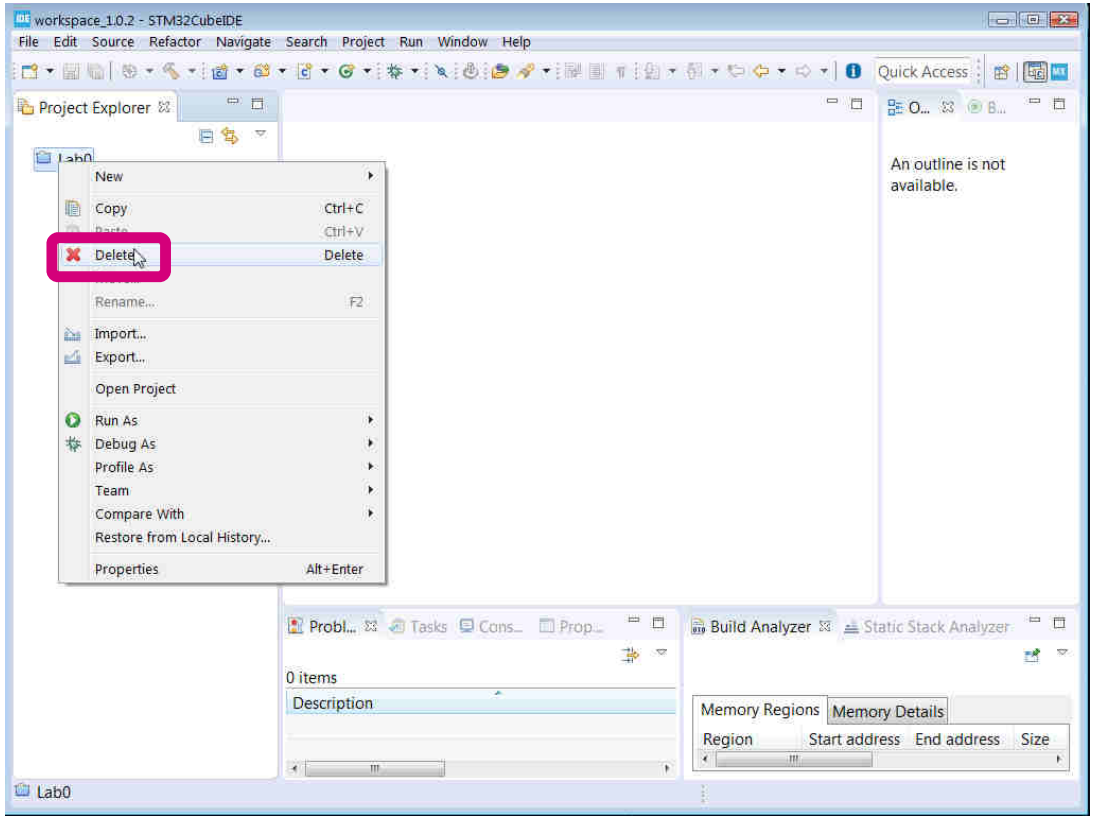
1. Right click on the project name and select **Close Project**



life.augmented

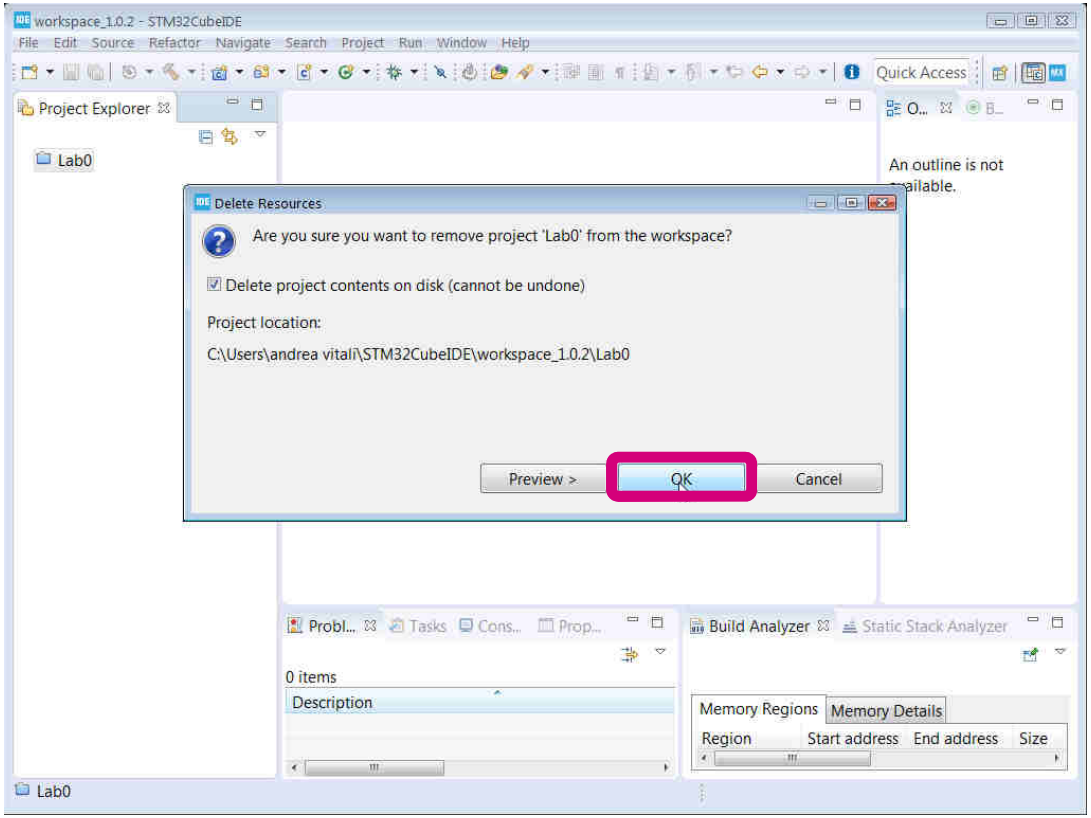
STM32CubeIDE Configuration

1. Right click on the project name and select **Delete**



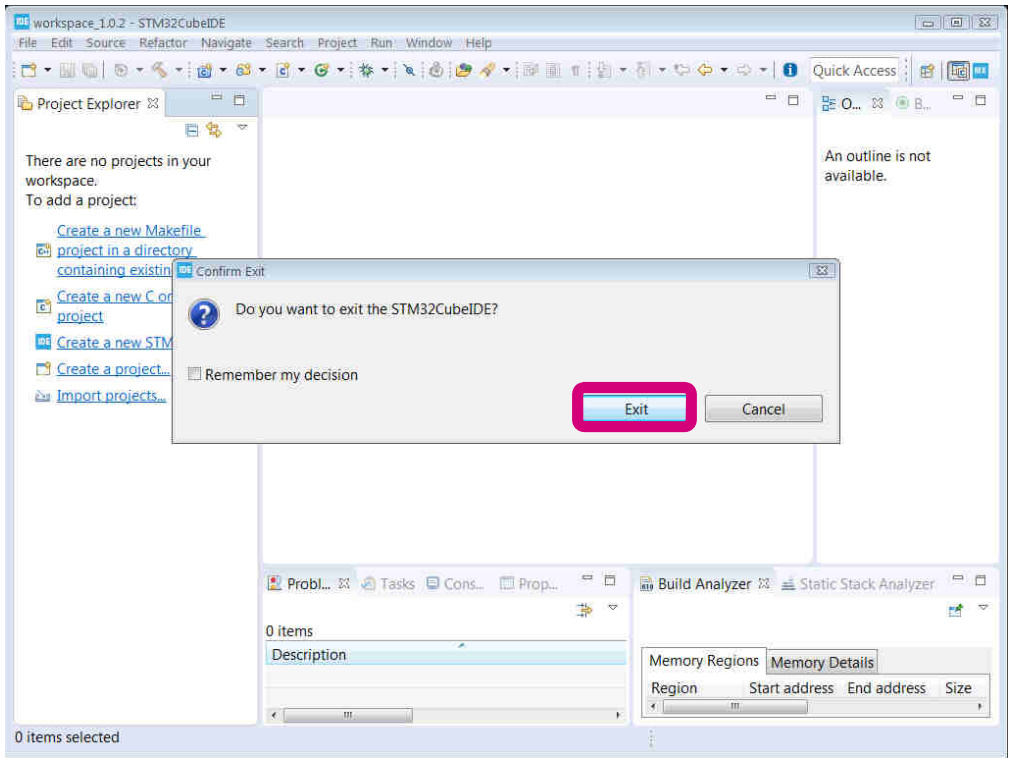
STM32CubeIDE Configuration

1. Check to delete files on disk and then click **Ok**

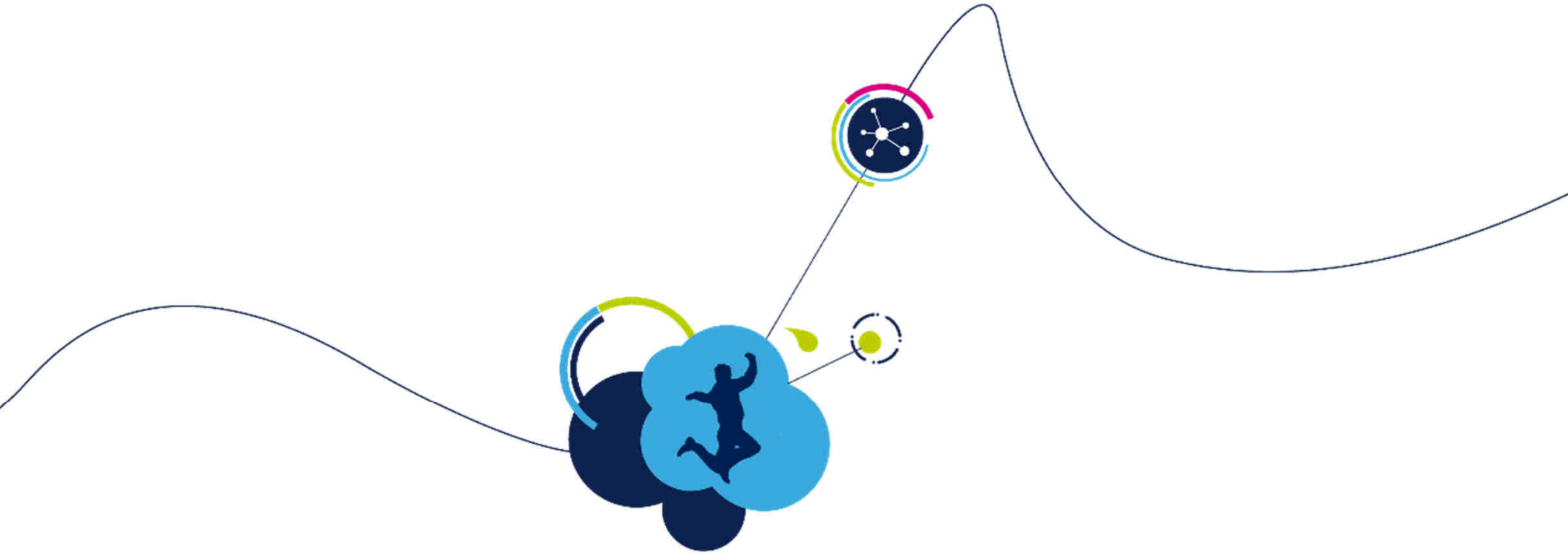


STM32CubeIDE Configuration

1. Close the window and confirm **Exit**



life.augmented

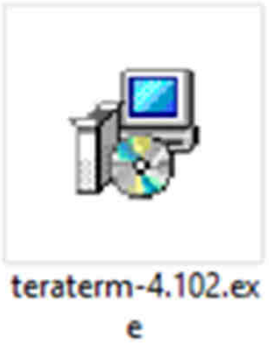


Tera Term Installation

life.augmented

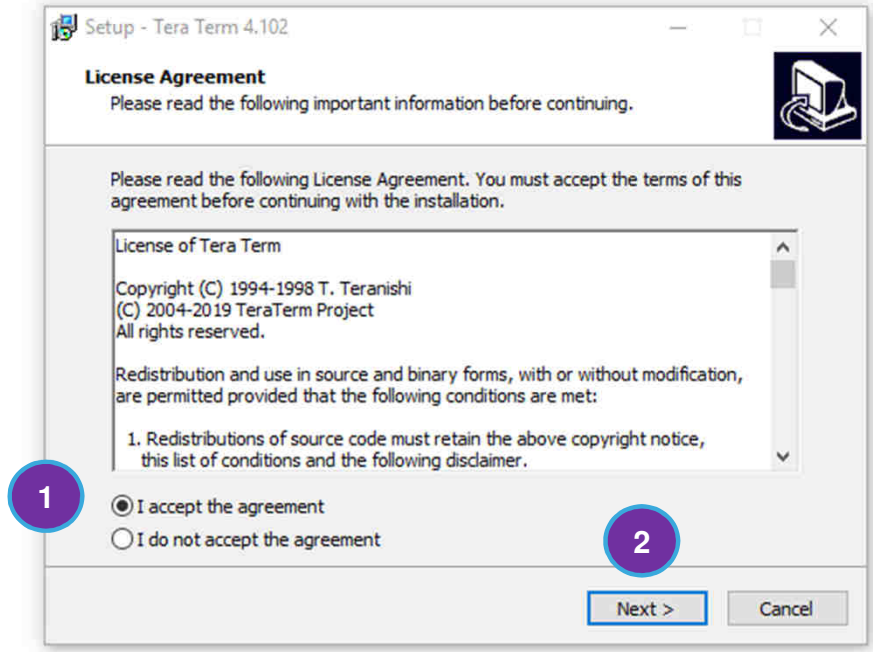
Tera Term Setup 1/9

1. Lunch the ***teraterm-4.102.exe*** installer



Tera Term Setup 2/9

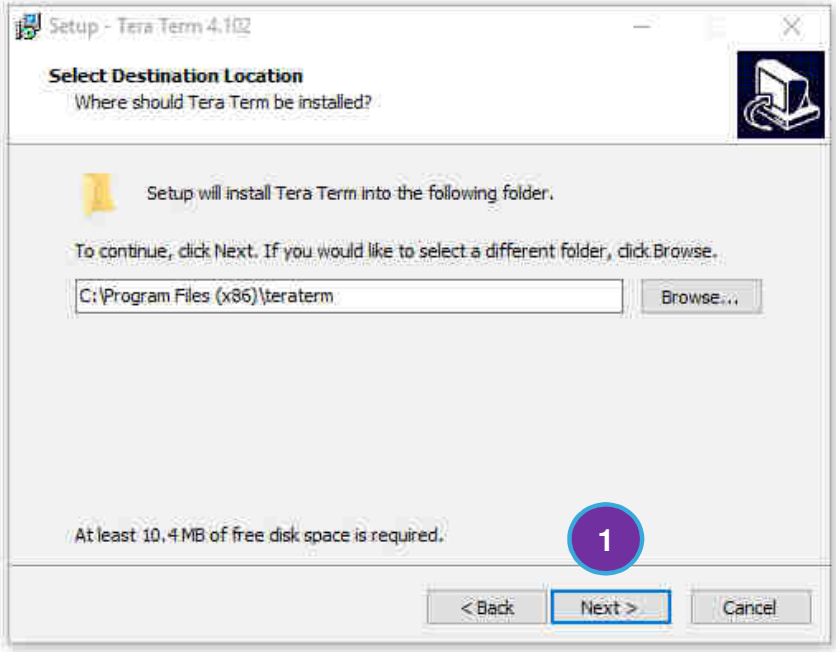
1. **Accept the agreement**
2. Click on **Next**



life.augmented

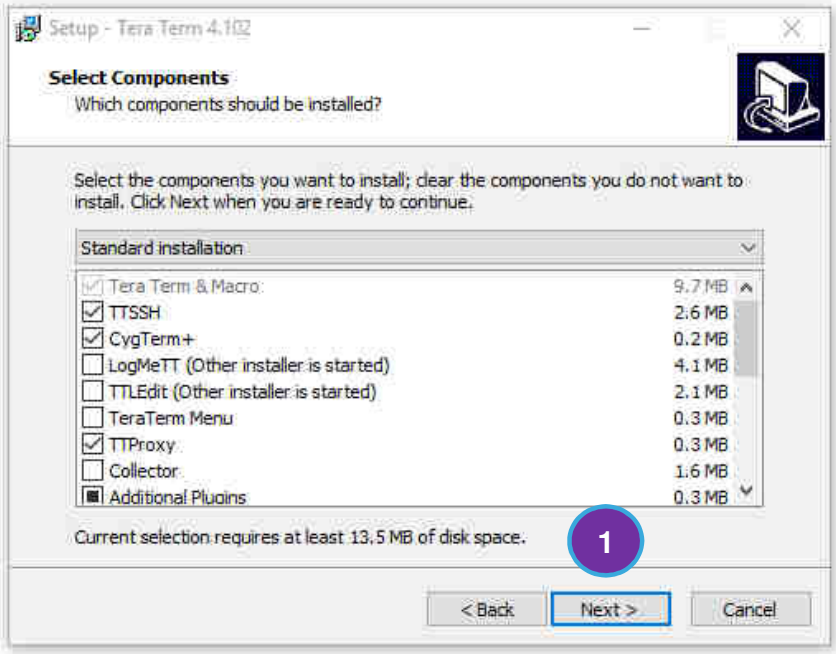
Tera Term Setup 3/9

1. Click on **Next**



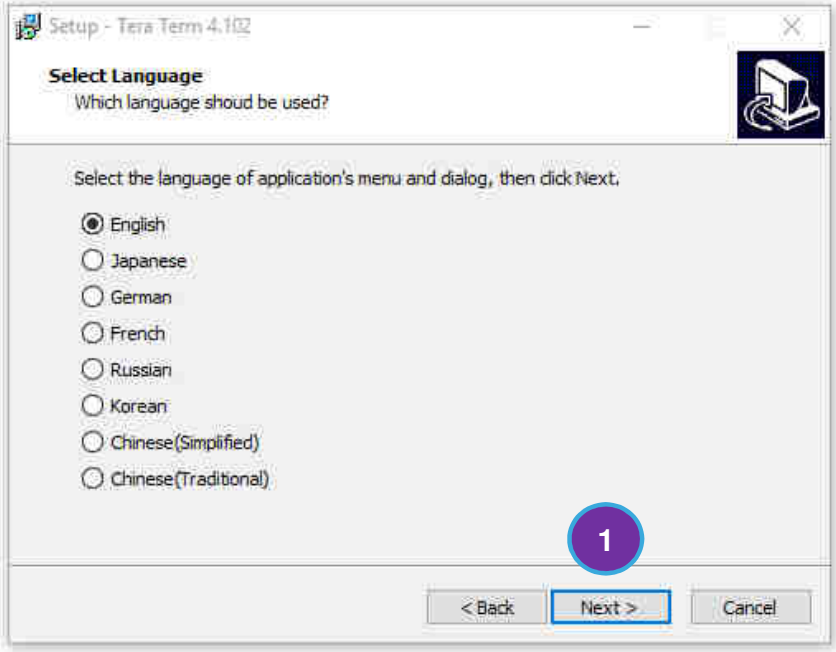
Tera Term Setup 4/9

1. Click on **Next**



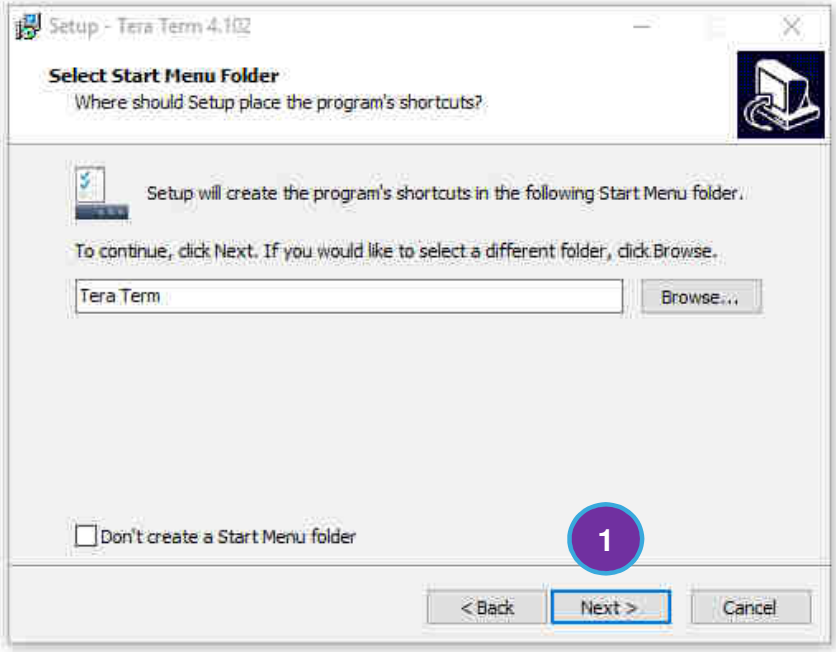
Tera Term Setup 5/9

1. Click on **Next**



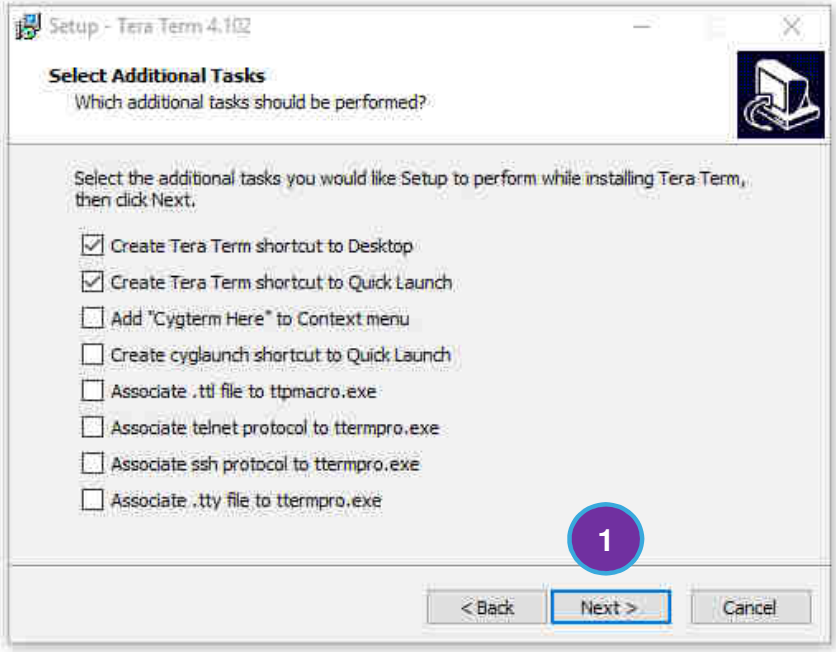
Tera Term Setup 6/9

1. Click on **Next**



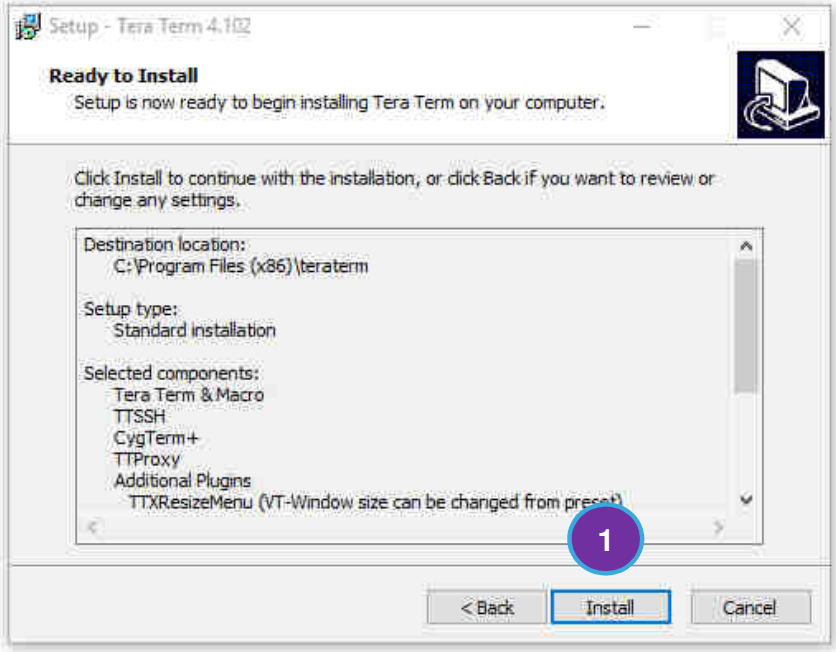
Tera Term Setup 7/9

1. Click on **Next**



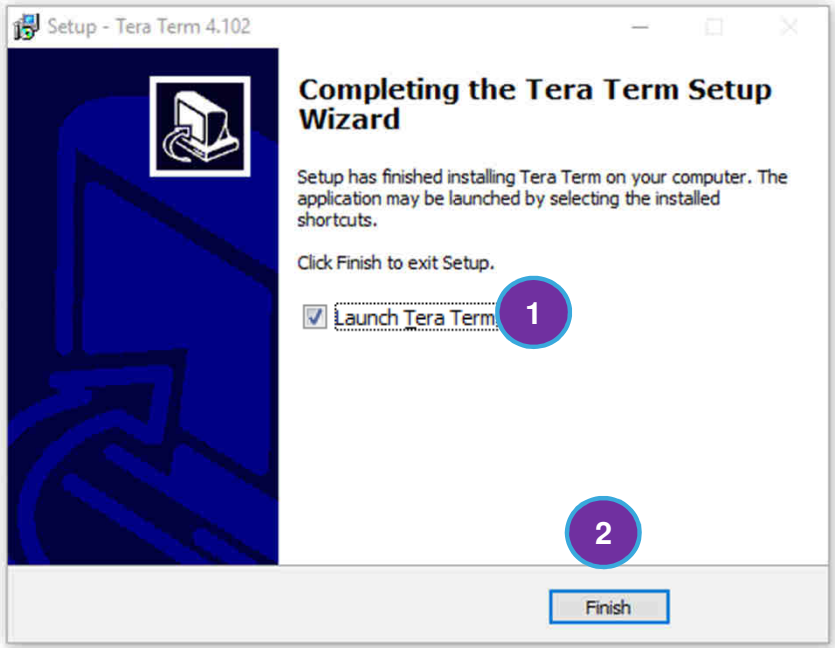
Tera Term Setup 8/9

1. Click on **Install**

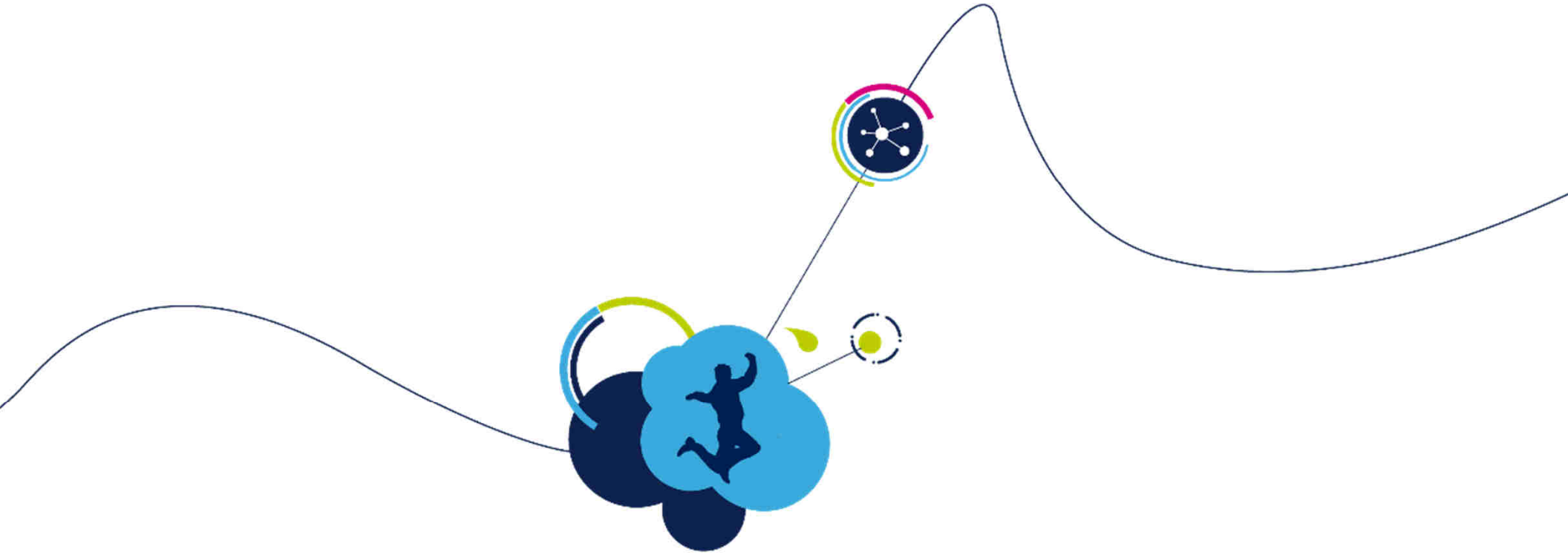


Tera Term Setup 9/9

1. Select **Launch Tera Term**
2. Click on **Finish**



life.augmented



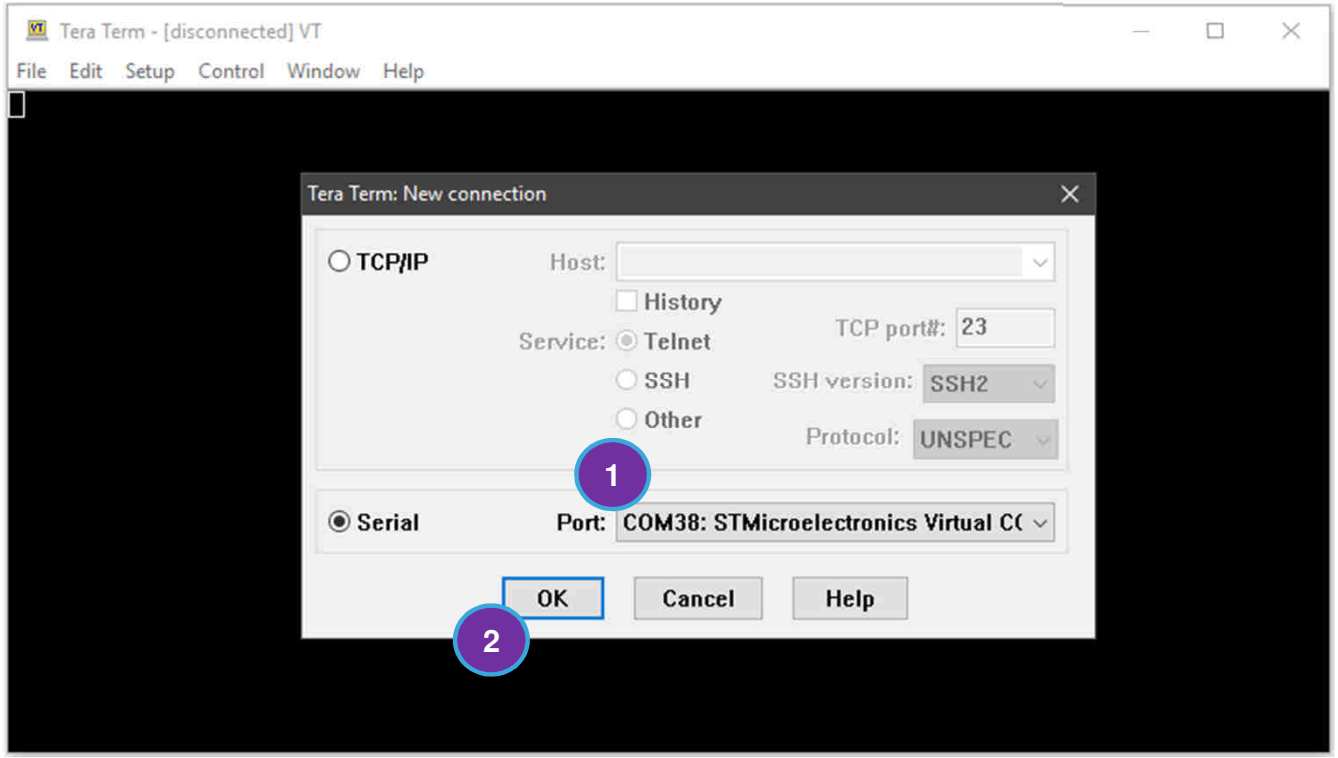
Tera Term Configuration

life.augmented

Tera Term Configuration 1/5

Plug the board to the PC using the micro USB cable provided

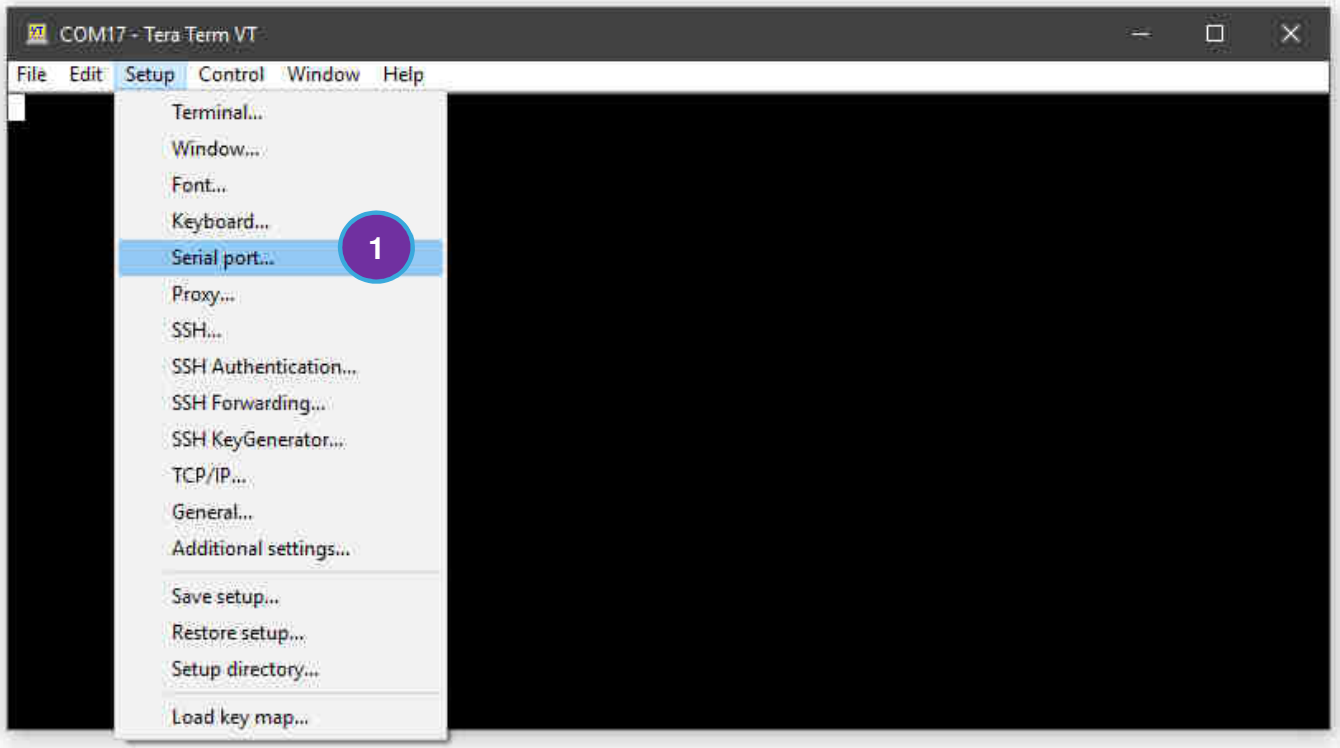
1. Select the **STMicroelectronics Virtual COM Port**
2. Click **OK**



life.augmented

Tera Term Configuration 2/5

1. Click **Setup** -> **Serial port…**

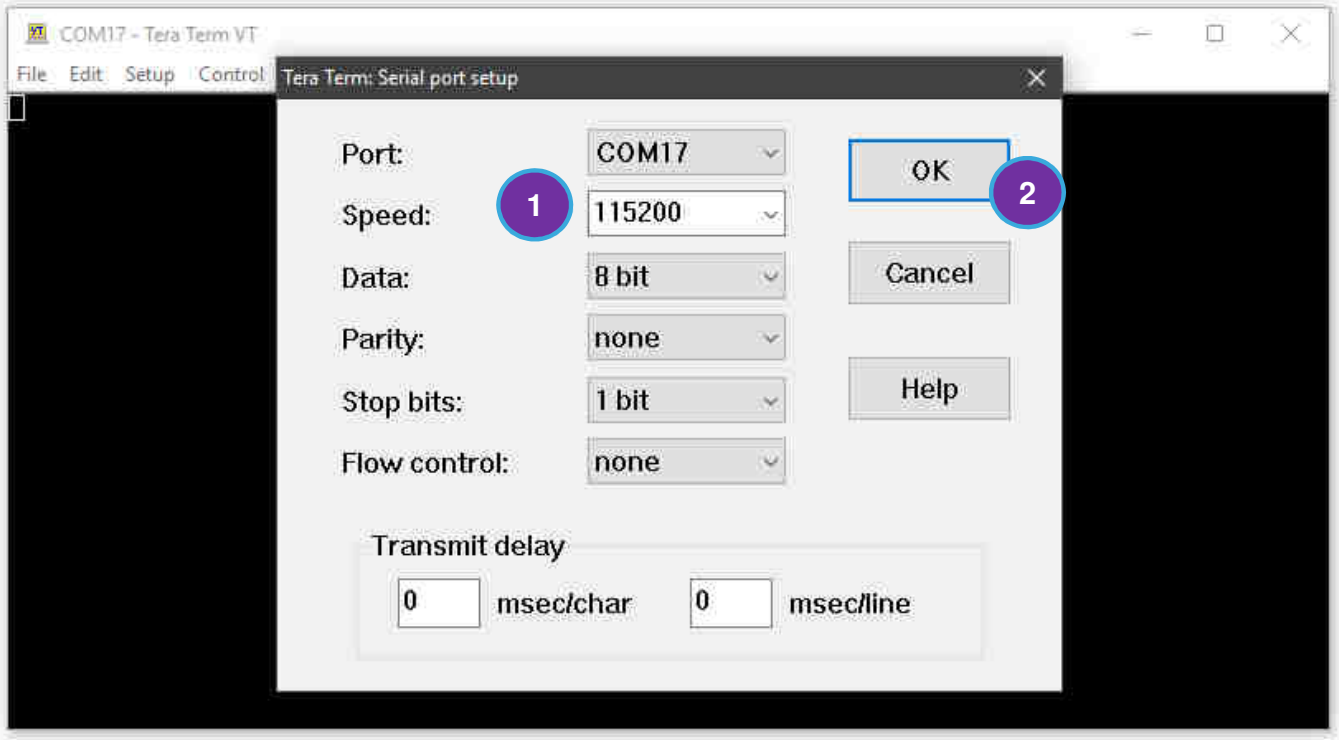


life.augmented

Tera Term Configuration 3/5

1. Set the following:
   Baud Rate: **115200**
   Data: **8 bit**
   Parity: **none**
   Stop: **1 bit**
   Flow control: **none**

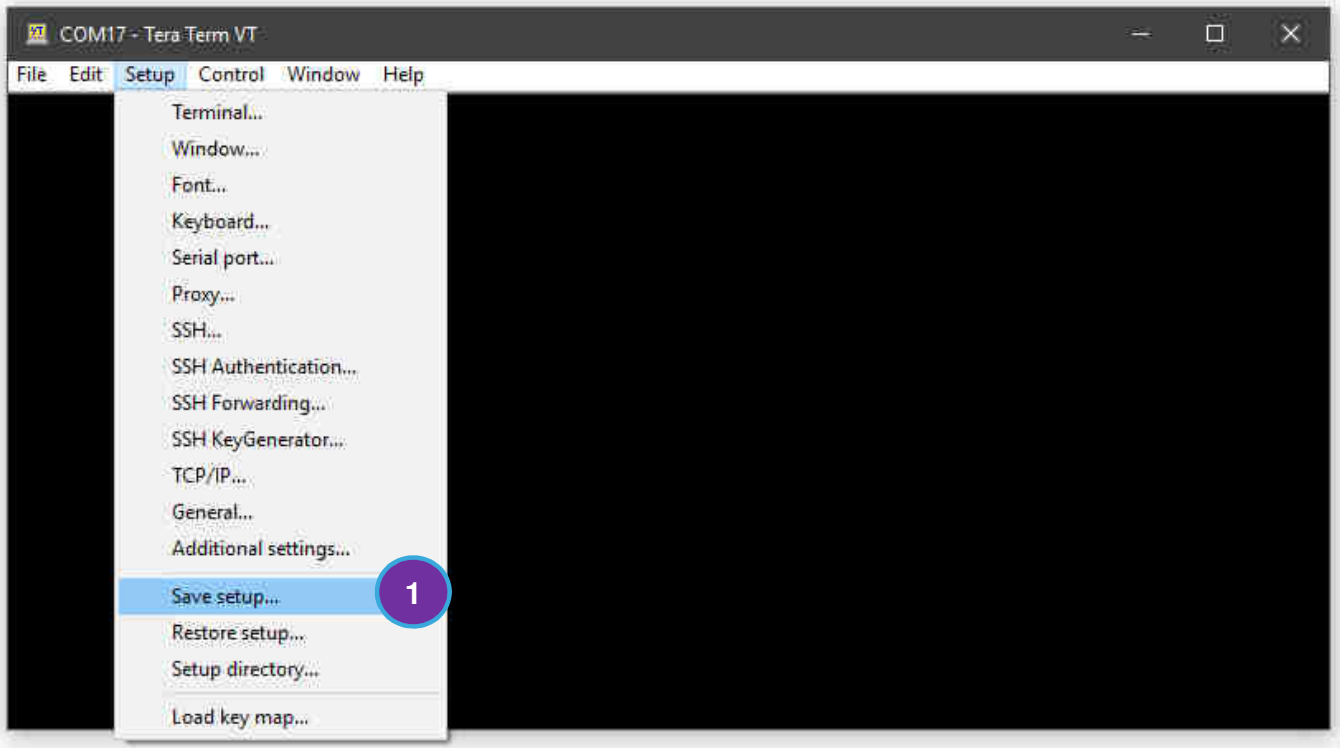
2. Click **OK**



life.augmented

Tera Term Configuration 4/5

1. Click **Setup** -> **Save setup…**



life.augmented

Tera Term Configuration 5/5

1. Click **Save**