



life.augmented

# Getting Started with IoT Development Platform



# Agenda

1 Introduction

2 Case study

3 Hardware ecosystem

4 Software ecosystem

5 Live demo 1: GPIO Toggle using Keil IDE

6 Live demo 2: GPIO Toggle using STM32CubeIDE

7 Live demo 3: Using X-Cube Package

# Introduction

# What is an ecosystem?

All collaterals required to develop with an MCU

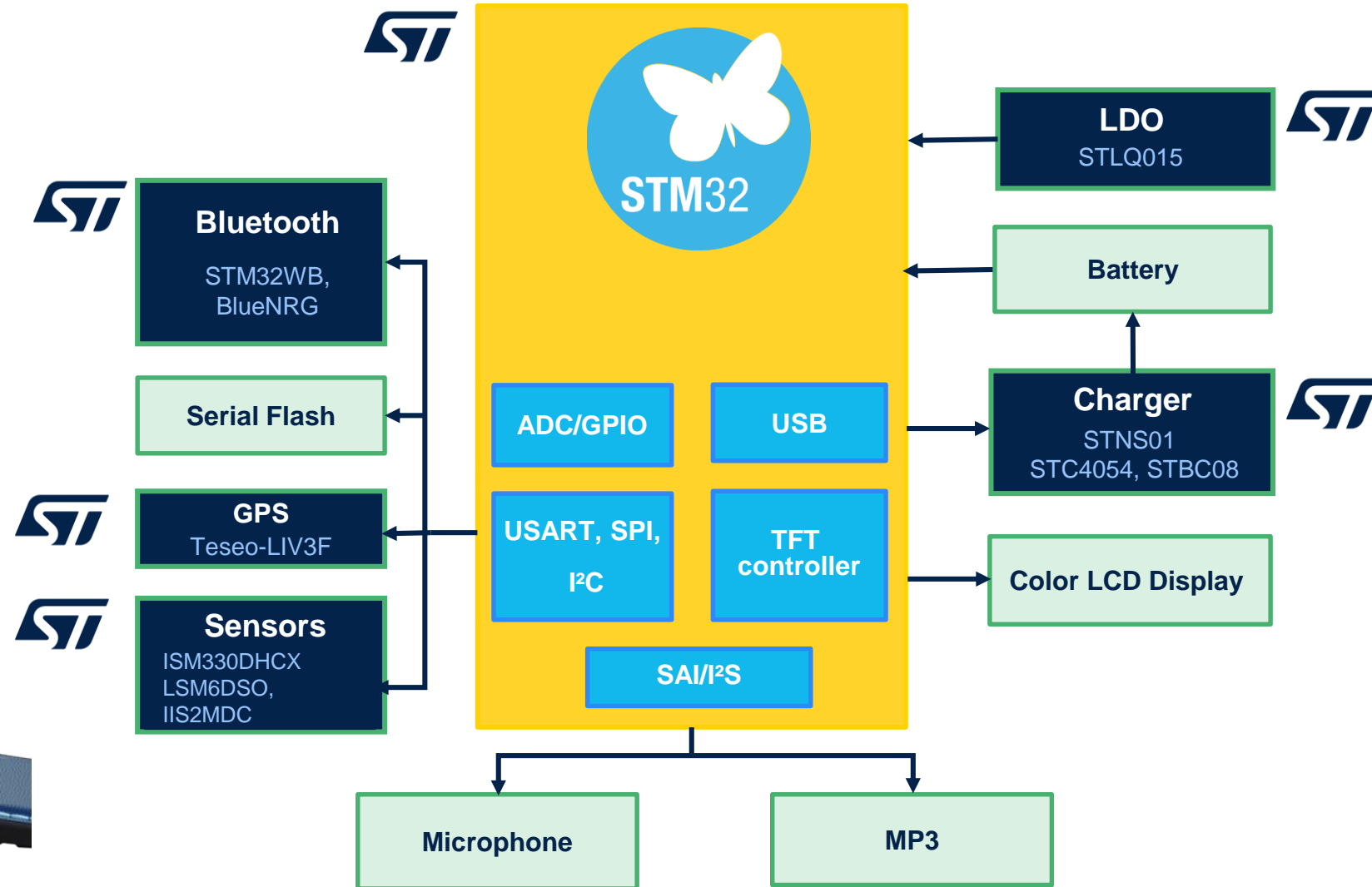




# Case study – smart watch

# Typical block diagram of smart watch

- Smart watches can contain:
  - Processing microcontroller
  - Connectivity over BLE
  - Sensors – Gyroscope, Accelerometer
  - Charger – USB Type C
  - LCD display with touch screen
  - Microphone
- Optional
  - NFC connectivity
  - Heart rate sensor



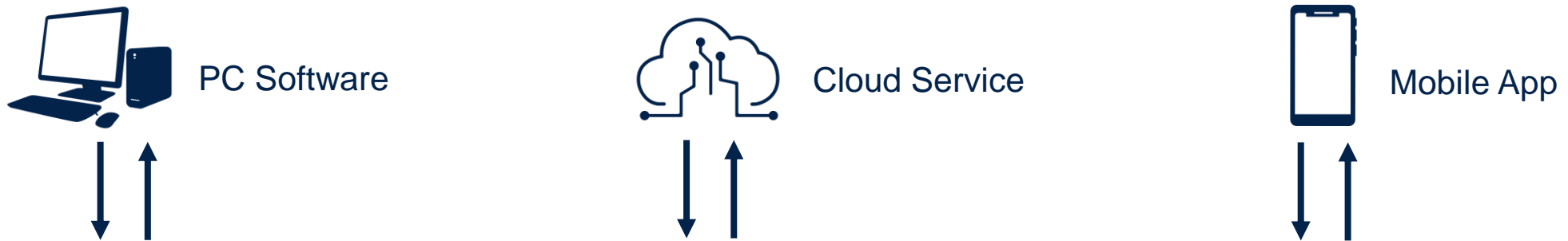
# The STM32 Open Development Environment (ODE)

The STM32 Open Development Environment (STM32 ODE) allows developers to verify their design assumptions and move quickly from ideas to a proof-of-concept.

STM32 ODE offers a comprehensive choice of hardware, firmware and software for the function required by the application.

It leverages the STM32 MCU hardware and software ecosystem.

# STM32 ODE synopsis



Product evaluation  
example



Function Pack with application examples



Middleware



Drivers and Hardware Abstraction Layer API



Hardware code optimized

Nucleo & X-Nucleo Boards



**64** STM32 Nucleo development boards  
Covering the broad portfolio of STM32 MCU families

**73** STM32 Nucleo expansion boards  
Offering peripheral functions

Solution Boards

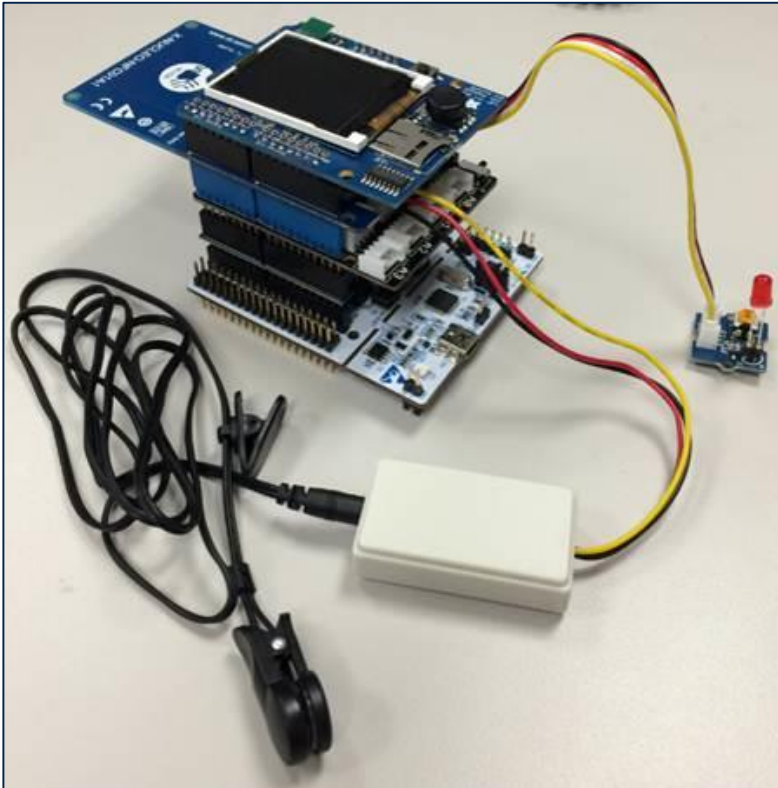


**11**

System boards  
Exploiting application functionalities and their form factor



# Rapid prototyping with ODE environment



← STM32 Nucleo64 board

M24SR NFC tag expansion board →



← Blue NRG BLE expansion board



X-NUCLEO-GFX01M2 Display expansion board →



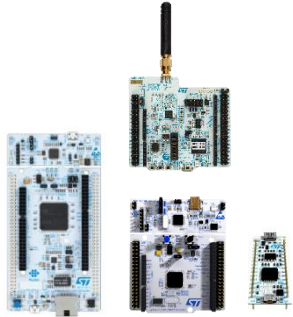
← Heart rate sensor



# HW Ecosystem



# Development tools overview



STM32 Nucleo

Flexible  
prototyping

[www.st.com/stm32nucleo](http://www.st.com/stm32nucleo)



Discovery kits

Key feature  
prototyping

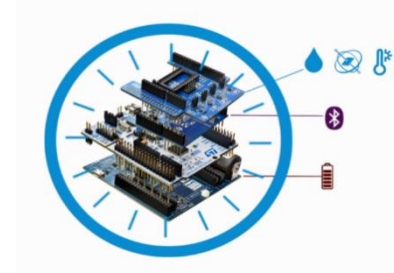
[www.st.com/stm32discovery](http://www.st.com/stm32discovery)



Evaluation  
boards

Full feature  
evaluation

[www.st.com/stm32evaltools](http://www.st.com/stm32evaltools)



STM32 Nucleo  
expansion

Functionality  
add-on

[www.st.com/x-nucleo](http://www.st.com/x-nucleo)



Third-party  
boards

From full  
evaluation to  
open hardware



Move Actuate



Connect



Power Drive



Sense



Translate



# STM32 Nucleo form choices

Ethernet

\*\*\*

USB

\*\*\*

ST Zio  
(Uno extended)

\*\*\*

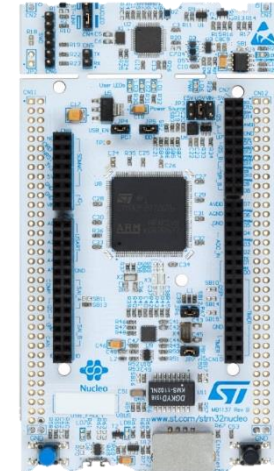
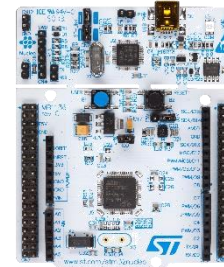
Arduino Uno  
ST morpho

\*\*

\*\*

Arduino Nano

\*



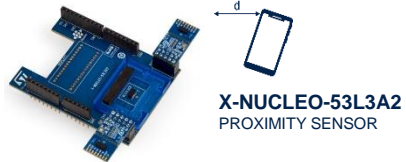
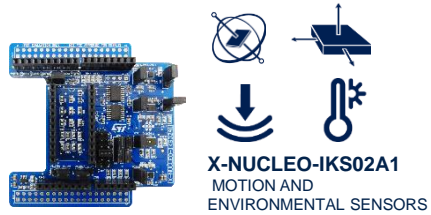
Nucleo-32  
32-pin MCU

Nucleo-64  
64-pin MCU

Nucleo-144  
144-pin MCU

# Nucleo expansion boards from ST

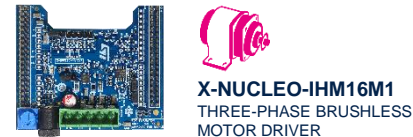
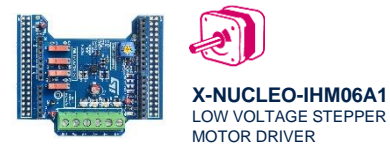
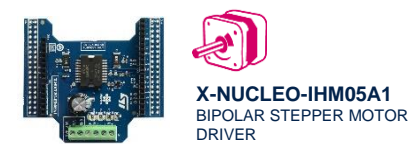
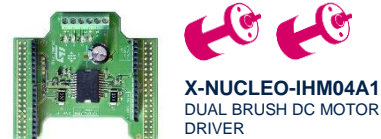
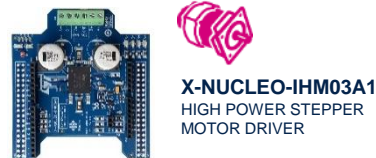
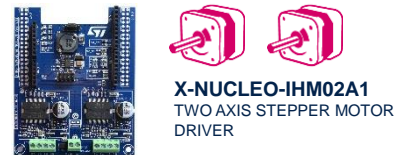
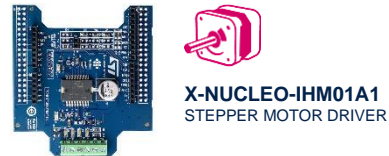
## Sensors and analog



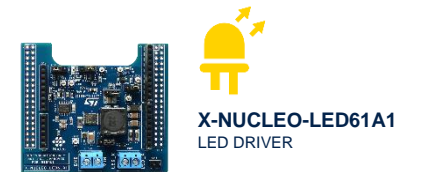
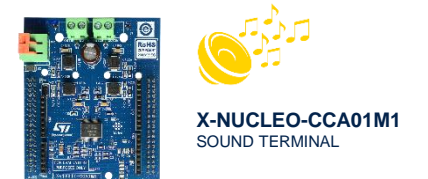
## Communication



## Motor drive

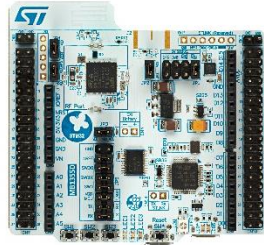


## Translate



# Nucleo boards for wireless communication

NUCLEO-WB55RG  
By ST



OPENTHREAD  
powered by Thread

**Multiprotocol RF transceiver**  
Bluetooth® LE 5.2 and IEEE 802.15.4-2011  
compliant

**LoRaWAN® node & gateway**  
LRWAN2 frequency > 800 MHz,  
LRWAN3 frequency < 500 MHz,  
LoRaWAN-compliant protocol stack



P-NUCLEO-LRWAN2  
By ST and USI



P-NUCLEO-LRWAN3  
By ST and RisingHF

NUCLEO-WL55JC  
By ST



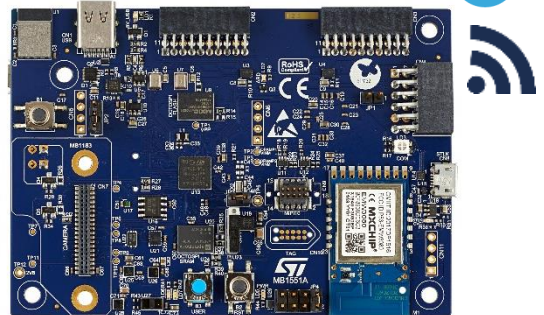
**Long-range wireless node**  
Frequency range 150-960 MHz  
LoRa, (G)FSK, (G)MSK, BPSK modulations  
LoRaWAN, Sigfox-compliant protocol stack



# Latest discovery kits



[STM32WB5MM-DK](#)



[B-U585I-IOT02A](#)



[B-L462E-CELL1](#)



[B-L072Z-LRWAN1](#)



[STM32G0316-DISCO](#)



[B-G474E-DPOW1](#)



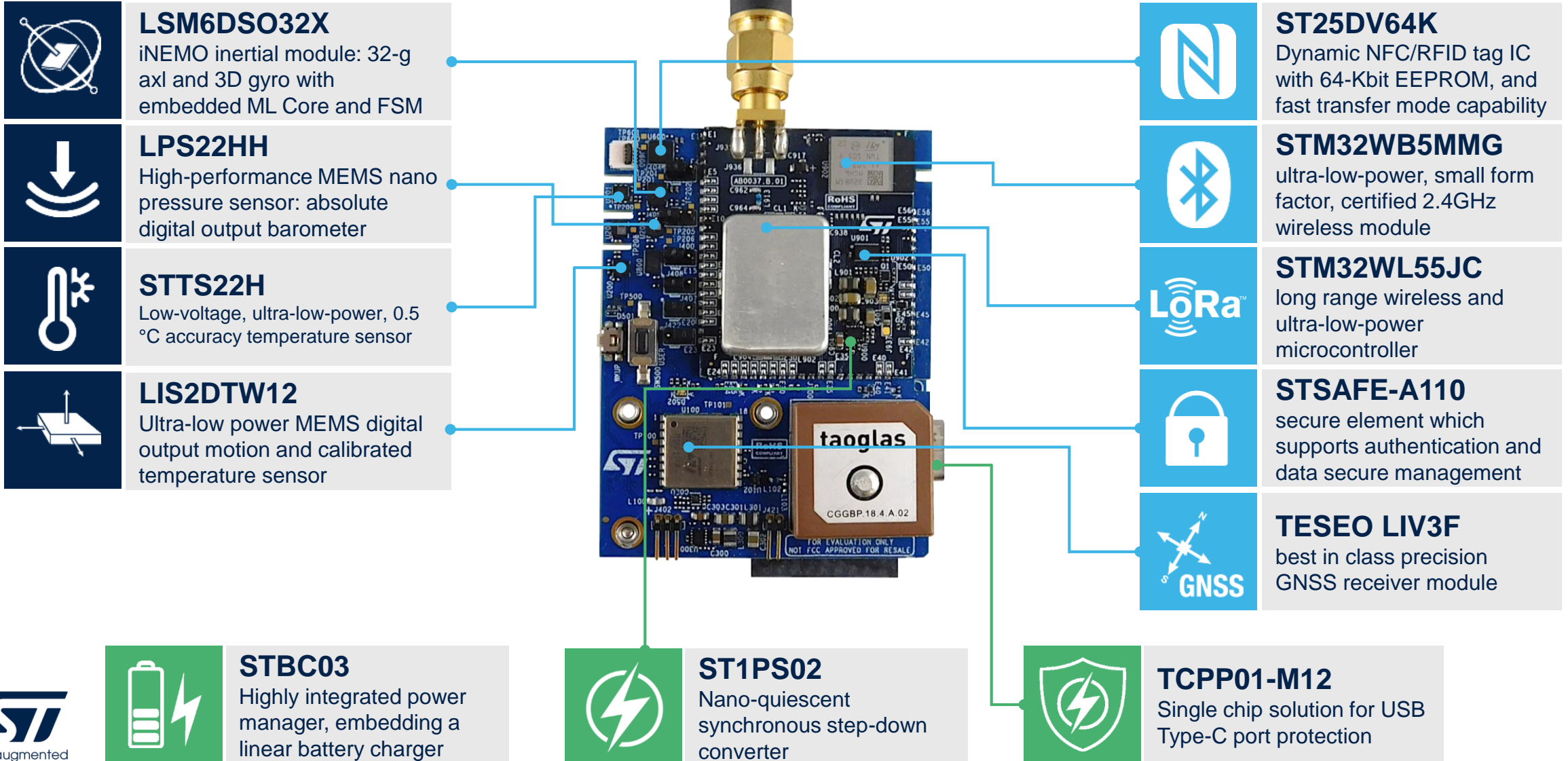
[STM32H735G-DK](#)



[STM32MP157F-DK2](#)

# STEVAL-ASTRA1B

## Asset tracking solution





# Software ecosystem

# Inside the STM32Cube ecosystem

Software Tools



Embedded Software

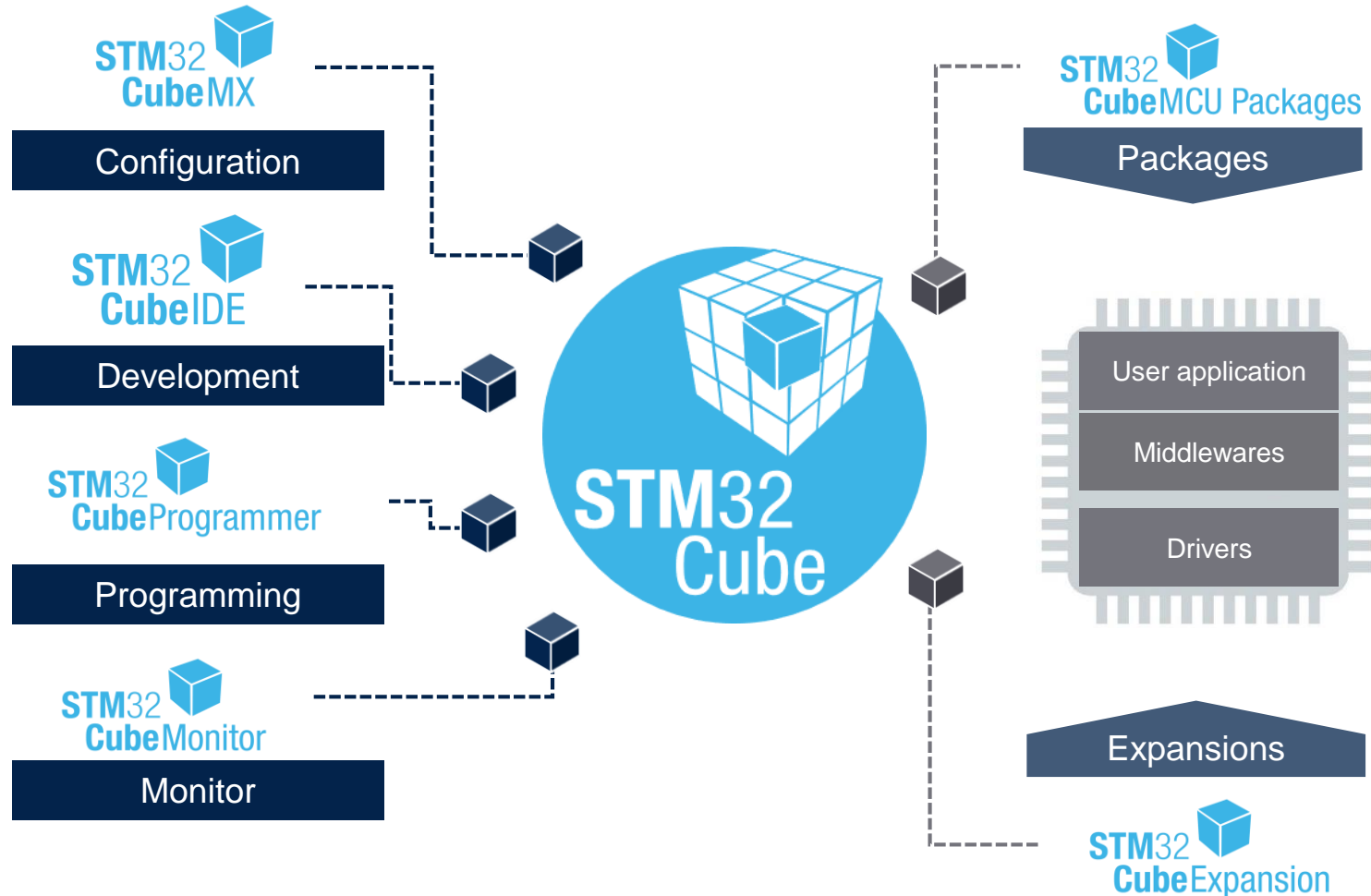


Save time

Save cost

Go beyond  
existing solutions

Easily monitor &  
debug applications



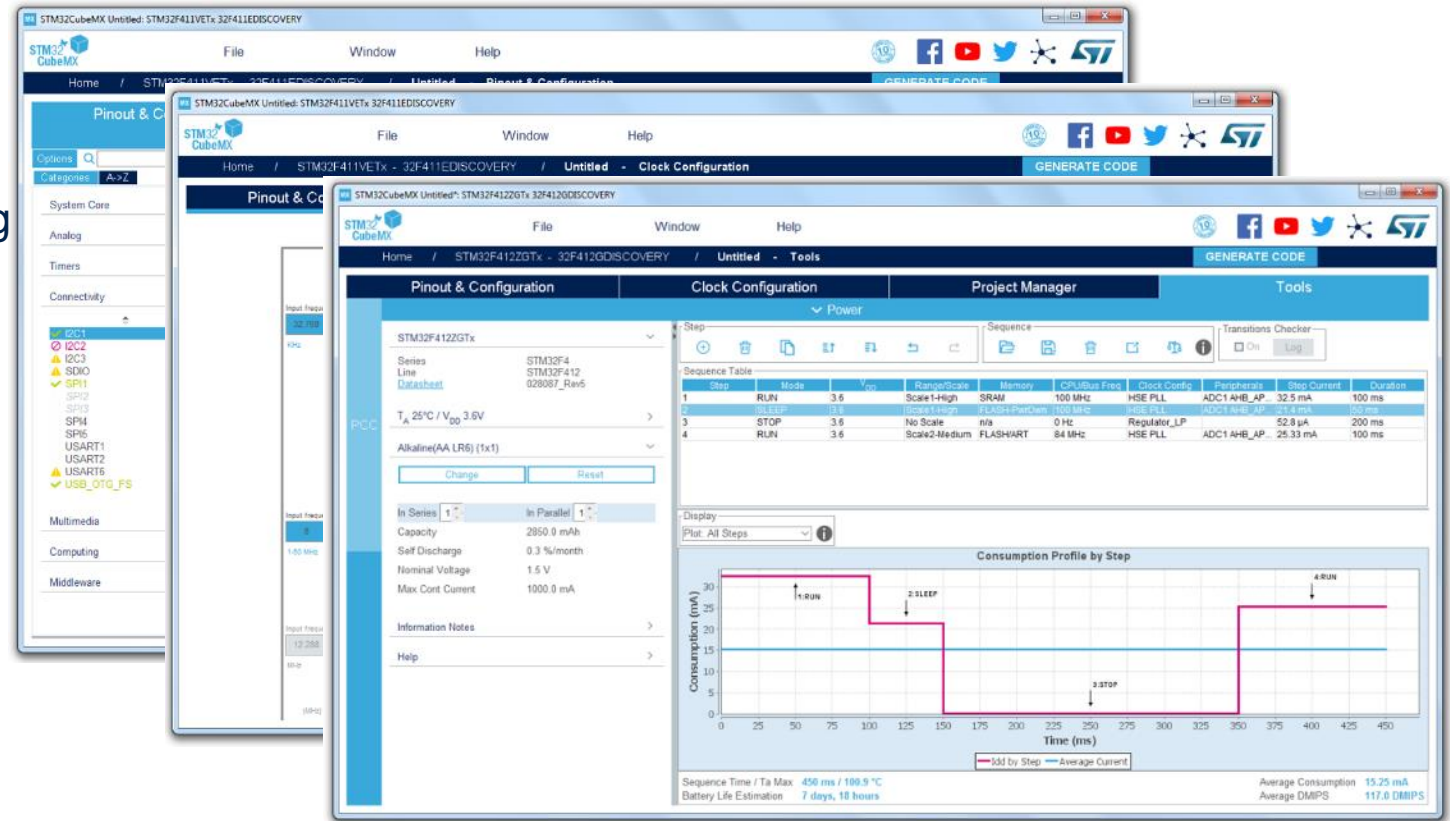
## Microcontroller configuration, step by step

### Step 1: Select the microcontroller

- Through easy filtering capabilities
- Through selection of ST board you are using

### Step 2: Configure the microcontroller

- Pinout wizard
- Peripherals and middleware wizard
- Clock tree wizard
- Power consumption wizard



## History

atollic  
TrueSTUDIO®

ST | atollic  
TrueSTUDIO® for STM32



STM32  
CubeMX



FREE



Eclipse/GCC Based

Free for Commercial Development

Multi-OS Support

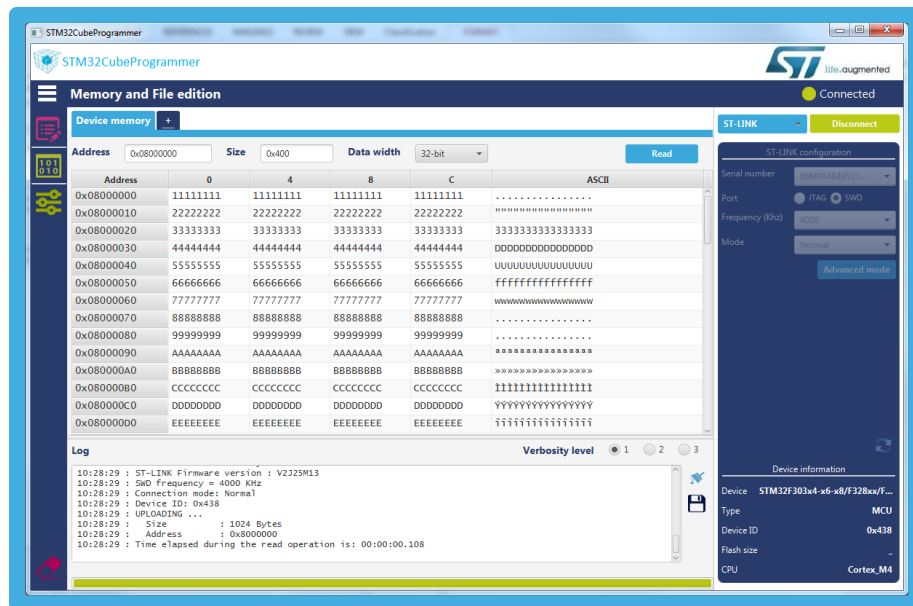
# All-in-one programming software tool



– debug interface of JTAG/SWD  
internal flash, ext. memory,  
option byte



– bootloader interface of UART  
– IAP, ISP



Windows  
macOS



– upgrade STLINK



– bootloader interface of USB  
– IAP, ISP





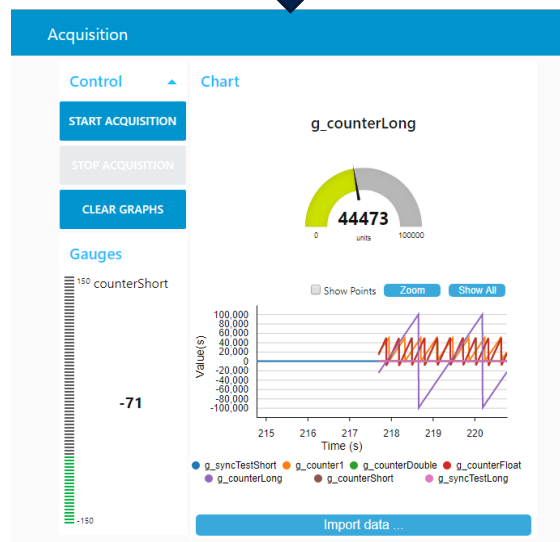
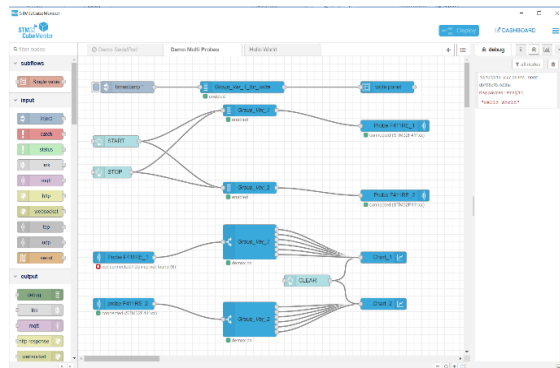
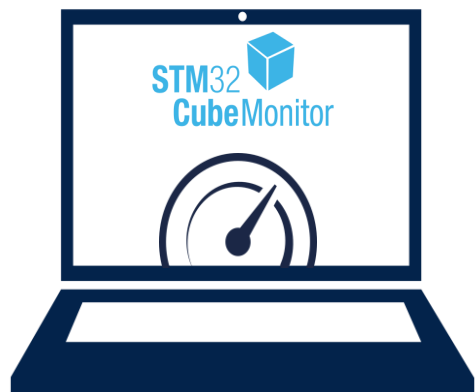
# From ST-Link V2 to STLink-V3

## ST-Link V2



## STLink-V3





# STM32CubeMonitor

## Monitoring application variables during runtime

- Non-intrusive tool to follow application behavior without interruption
- Real-time analysis to finetune application configuration

## Drag & drop creation of dashboard UI

- Large choice of graphical components (gauges, bar graphs, plots...)
- Customize settings. No need for programming.

## Graphical visualization on any display

- Multi-OS tool: direct support of PC, tablets and smartphones
- Remote monitoring

# STM32Cube MCU Packages

# One-stop-shop SW packages

Mainstream MCU	High Performance MCU	MPU	Ultra-Low Power MCU	Wireless MCU
STM32 CubeG4	STM32 CubeH7	STM32 CubeMP1	STM32 CubeL0	STM32 CubeWB
STM32 CubeF3	STM32 CubeF7		STM32 CubeL1	STM32 CubeWL
STM32 CubeF1	STM32 CubeF4		STM32 CubeL4	
STM32 CubeG0	STM32 CubeF2		STM32 CubeL5	
STM32 CubeF0				

## Peripheral drivers

### HAL API

Hardware Abstraction Layer, highly portable and easy to use

### LL APIs

Low-Layer APIs, light weight and highly optimized for runtime efficiency

## STM32Cube Middleware

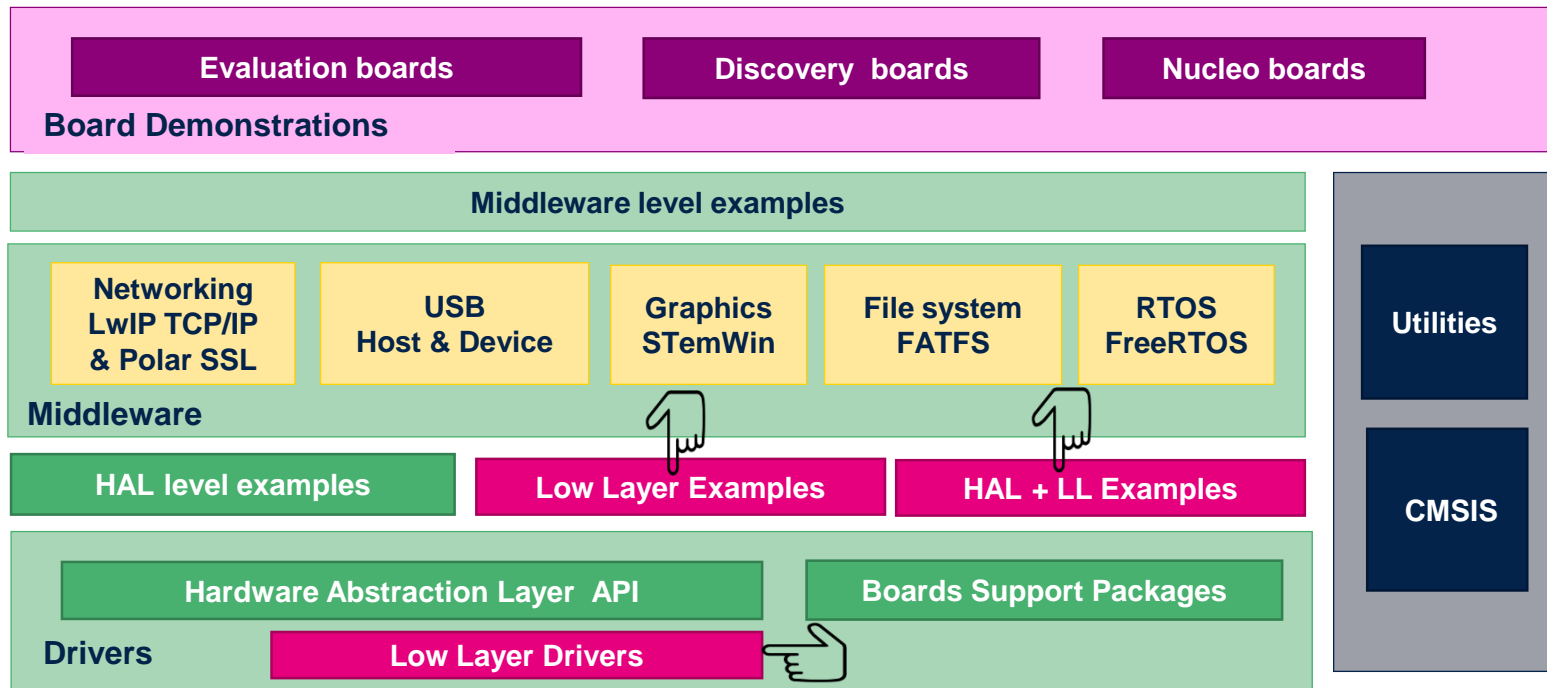
### Generic MW

- FreeRTOS
- FatFS file system
- LwIP TCP/IP stack
- mbedTLS and mbedCrypto
- Open Bootloader

### Dedicated MW

- ST Bluetooth 5 stack
- OpenThread stack
- ST 802.15.4 MAC
- Zigbee 3 stack
- STM32 WPAN
- LoRaWAN stack
- Sigfox stack
- Sub-GHz phy
- ST Key Management Services (KMS)
- TF-M
- ST USB Host & Device stacks
- STM32 Touch Sensing library
- STemWin graphics stack

# STM32Cube Package Overview



## Application benefits

- Single package
- Compatible with all STM32 series
- Source code with open-source BSD license



# Adding Azure RTOS in the STM32Cube ecosystem

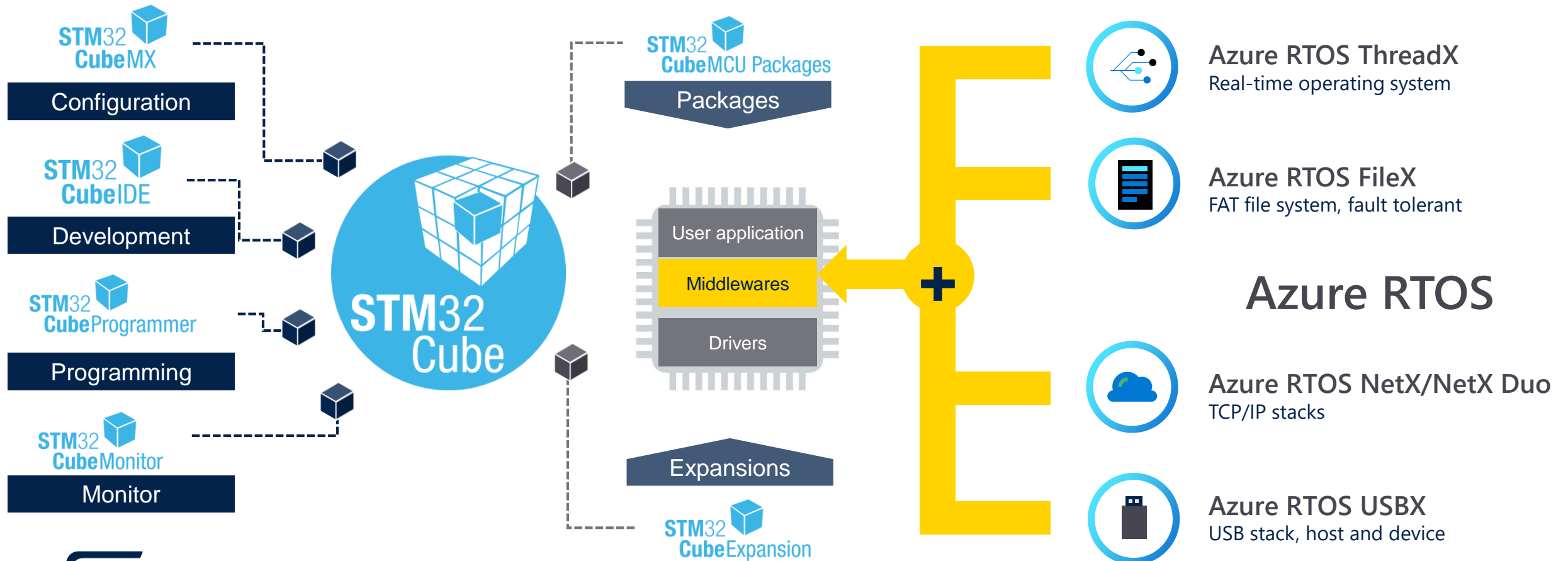
Software Tools



Embedded Software



Complemented with Microsoft Azure RTOS





# Expansions for various applications

## Audio



### X-CUBE

X-CUBE-AUDIO  
X-CUBE-VS4A  
X-CUBE-USB-AUDIO

## Bootloader/Secure Boot



### X-CUBE

X-CUBE-IAP-USART  
X-CUBE-IAP-SD  
X-CUBE-SBSFU

## Safety



### X-CUBE

X-CUBE-CLASSB  
X-CUBE-STL

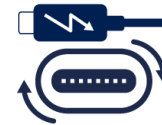
## Crypto



### X-CUBE

X-CUBE-CRYPTOLIB

## USB



### X-CUBE

X-CUBE-USB-PD

## SigFox



### X-CUBE

X-CUBE-SFOX

## LoRa



### I-CUBE

I-CUBE-LRWAN

## Sub-1G



### X-CUBE

X-CUBE-SUBG1

## BLE



### X-CUBE

X-CUBE-BLE1

Enhanced for  
STM32 Toolset

X-CUBE-BLE2

Enhanced for  
STM32 Toolset

## NFC



### X-CUBE

X-CUBE-NFC1

X-CUBE-NFC2

X-CUBE-NFC3

X-CUBE-NFC4

Enhanced for  
STM32 Toolset

X-CUBE-NFC5

# Expansions with Function Packs

## Cloud



### X-CUBE

X-CUBE-CLD-GEN1  
X-CUBE-AWS  
X-CUBE-AZURE  
X-CUBE-WATSON  
X-CUBE-GCP

### FP

FP-CLD-AWS1  
FP-CLD-AZURE1  
FP-CLD-WASTON1

## Motion



### X-CUBE

X-CUBE-6180XA1  
X-CUBE-IKA02A1  
X-CUBE-MEMS-XT1  
X-CUBE-MEMS1  
X-CUBE-MEMS1-V4

Enhanced for  
STM32 Toolset

### FP

FP-SNS-6LPNODE1  
FP-SNS-ALLMEMS1  
FP-SNS-FLIGHT1  
FP-SNS-MOTENV1

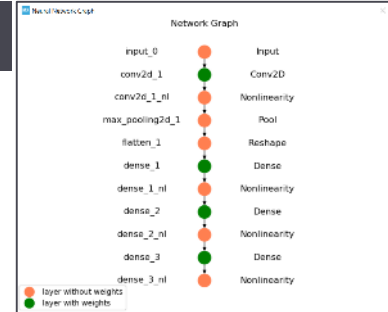
## AI



### X-CUBE

### X-CUBE-AI

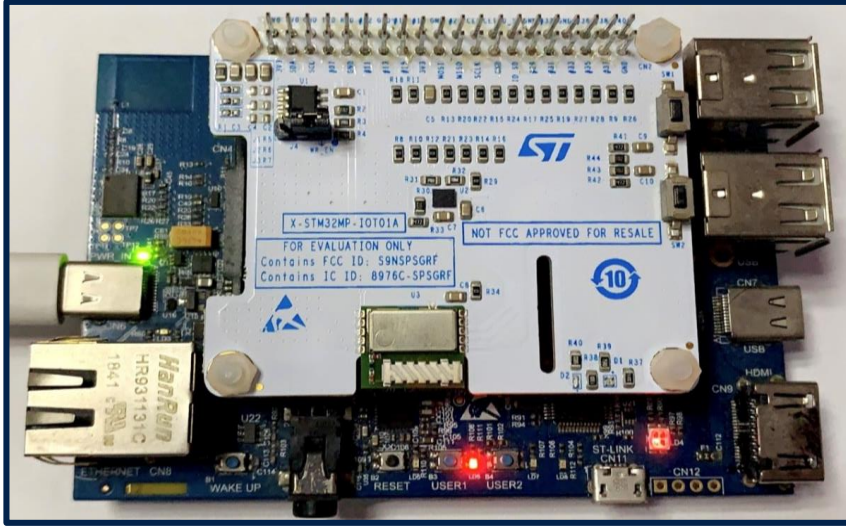
Enhanced for  
ST Toolset



### FP

FP-AI-SENSING1  
FP-AI-VISION1

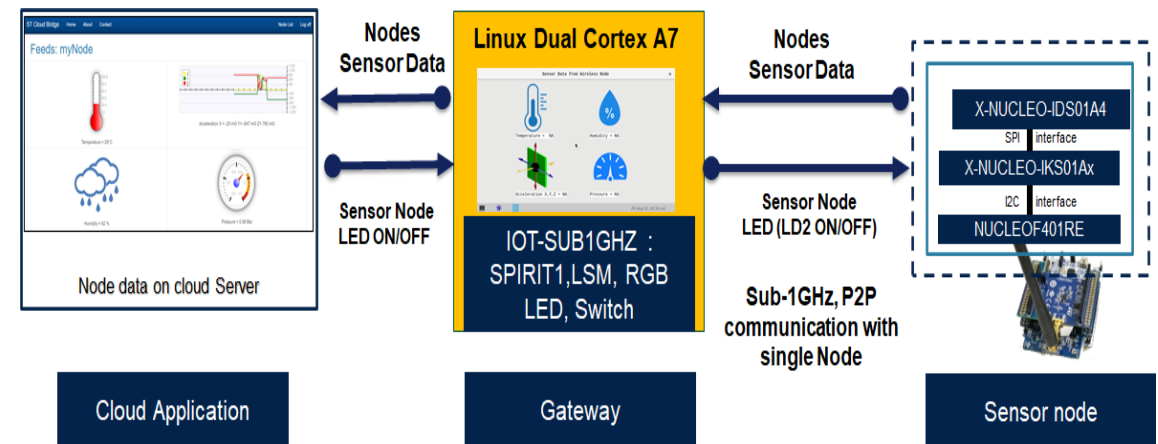
# X-STM32MP-IOT01A & X-LINUX-IOT01E/A



- [X-STM32MP-IOT01A](#) is the first 40-pin expansion board under the umbrella of Open Development Environment (ODE)
- The 40-pin expansion board is created for new MPU platform from ST (STM32MP157F-DK2) & inline with the specification from Raspberry Pi HAT

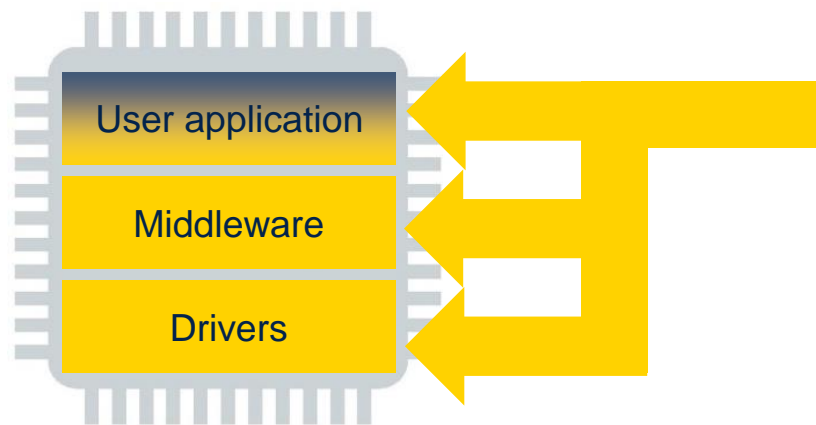
**X-STM32MP-IOT01A**

- [X-LINUX-IOT01E/A](#) is a software package on for X-STM32MP-IOT01A expansion board on Yocto Platform
- An IOT Application where X-STM32MP-IOT01A expansion plugged on Discovery Kit act as gateway. The gateway fetches data from sensor node on Sub1-Ghz and send to the Cloud.



**X-LINUX-IOT01E/A**

# Leading Graphics Performance



STM32G0	STM32H725	STM32U5
up to 480x272	to 640x480	up to 1024x768
<b>New NeoChrom GPU</b>		

## TouchGFX DESIGNER

Drag-and-drop  
PC GUI-builder & -simulator



# STM32 MCU Developer Zone

Everything for STM32 developers, in one place



MCU portfolio and  
selection



Hardware  
evaluation and  
development tools



Software  
development kit



Artificial  
Intelligence for  
STM32



Wireless  
Connectivity



STM32Trust  
security framework



Solutions



Developer  
resources



Community &  
support





# STM32 Education and Support

## Wide support for developer communities



**FAE - Worldwide  
Customer Support**



**STM32 Online Training**



**community.st.com**



**Massive Open Online  
Courses**



**Wiki.st.com/stm32mpu  
wiki.st.com/stm32mcu**



**Textbooks**



**Partner training courses**



**github.com/STMicroelectronics**



life.augmented



[www.st.com/content/st\\_com/en/support/learning/stm32-education.html](http://www.st.com/content/st_com/en/support/learning/stm32-education.html)



# Live demo 1: GPIO Toggling using Keil IDE

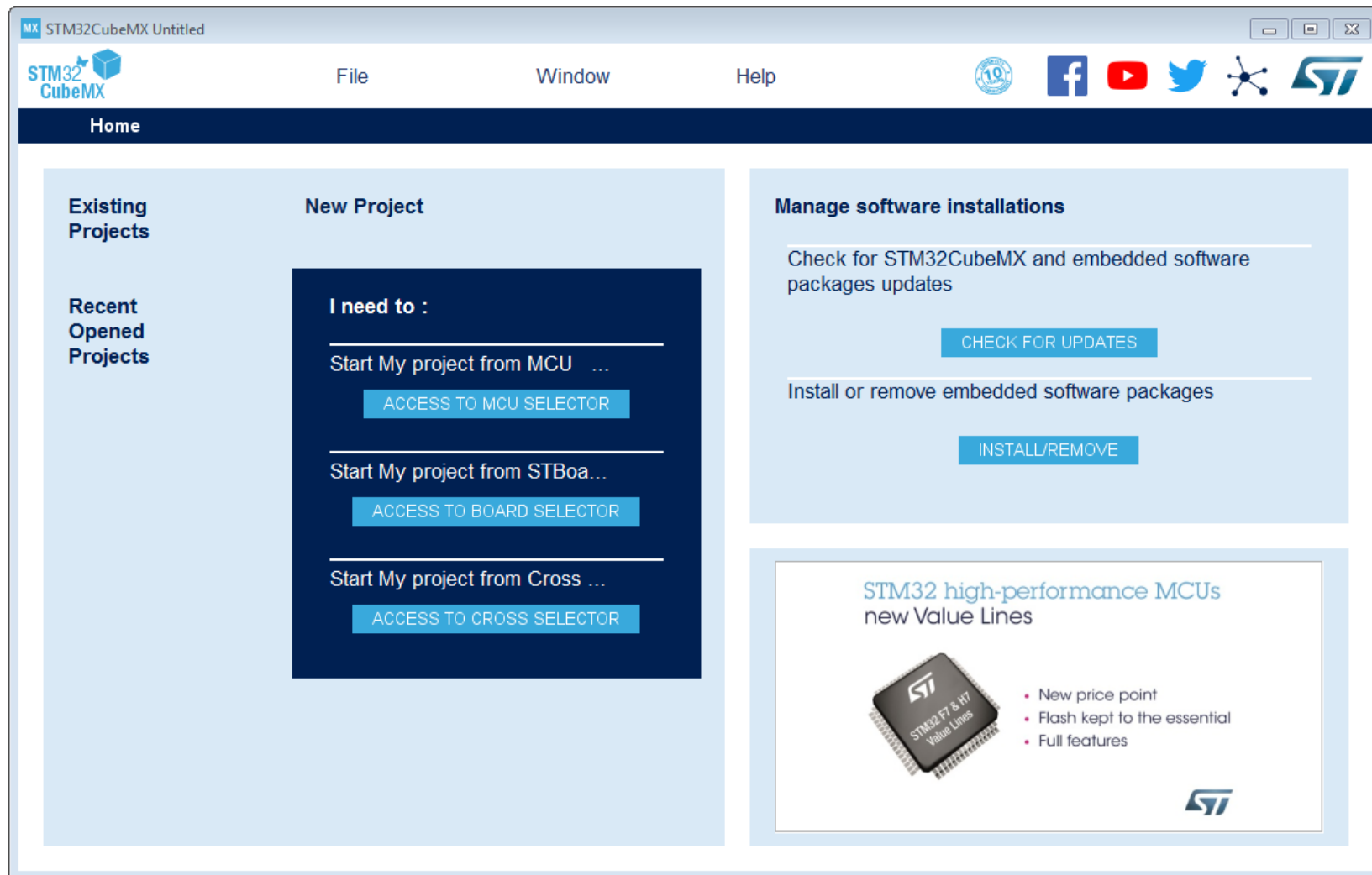
# STM32CubeMX installation

- STM32CubeMX, STM32CubeIDE & STM32Cube package is downloadable from [www.st.com/stm32cube](http://www.st.com/stm32cube)
- STM32CubeMX needs Java RE
  - Check release notes of the particular version for additional requirements.
  - Multiplatform tool runs on Windows, Linux and macOS
  - Download from [www.java.com/en/download/](http://www.java.com/en/download/)

# STM32CubeMX home page



1. Launch STM32CubeMX tool
2. The home page allows you to select the existing or recent projects or to create new projects.
3. Quick access to package installation and updates also available.



# Prerequisites: updater settings

- After the software package is launched, click on Help > Updater Settings (Alt+S)
- Configure the Updater Settings.
  - Default repository folder is predefined but is user configurable.
  - Enable/Disable the automatic check for updates in tab “Updater Settings”

MX Updater Settings

Updater Settings Connection Parameters

Firmware Repository

Repository Folder

C:/Users/ekkachai muangrodpai/STM32Cube/Repository/ Browse

Check and Update Settings

☒ Manual Check

☐ Automatic Check Interval between two Checks (days) 5

Data Auto-Refresh

☒ No Auto-Refresh at Application start

☐ Auto-Refresh Data-only at Application start

☐ Auto-Refresh Data and Docs at Application start

Interval between two data-refreshes (days) 5

MX Updater Settings

Updater Settings Connection Parameters

Firmware Repository

Repository Folder

C:/Users/john doe/STM32Cube/Repository/ Browse

Check and Update Settings

☐ Manual Check

☒ Automatic Check Interval between two Checks (days) 5

Data Auto-Refresh

☐ No Auto-Refresh at Application start

☒ Auto-Refresh Data-only at Application start

☐ Auto-Refresh Data and Docs at Application start

Interval between two data-refreshes (days) 3

OK Cancel

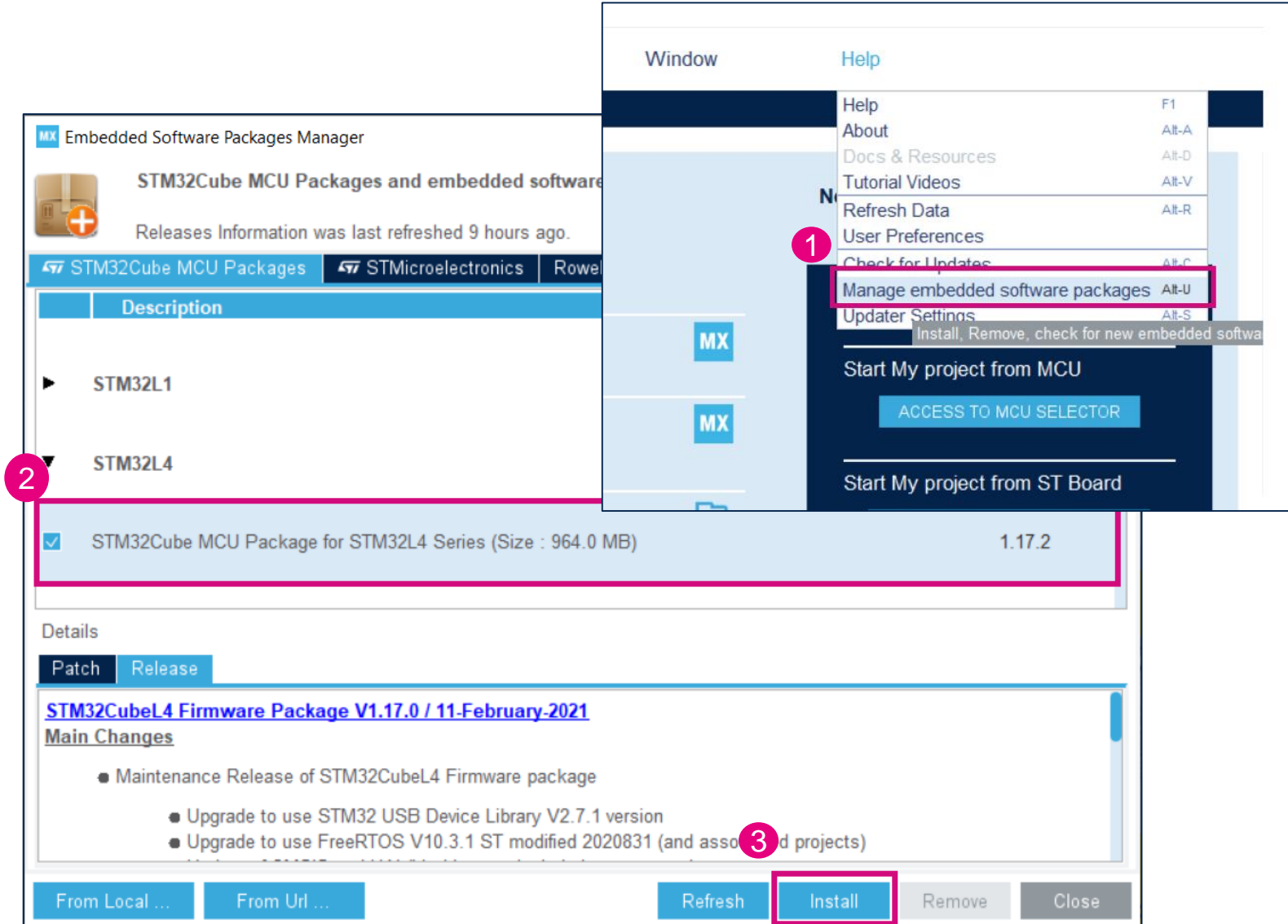
# Prerequisites: Updater Settings

- Configure the Connection Parameters
  - By default, there is no Proxy need. If your computer needs to set a Proxy you can do it at Connection Parameter
  - Specify the proxy server settings that are appropriate for your network configuration by selecting a proxy type among the following possibilities

The image displays two screenshots of the 'MX Updater Settings' dialog box, specifically the 'Connection Parameters' tab. The top screenshot shows the default configuration where 'No Proxy' is selected under 'Proxy Server Type'. The bottom screenshot shows the configuration for 'Manual Configuration of Proxy Server', where 'Manual Configuration of Proxy Server' is selected. In this configuration, the 'Proxy HTTP' field is set to 'myproxy.mycompany.com' and the 'Port' is '8080'. Under the 'Authentication' section, both 'Require Authentication' and 'Remember my Credentials' are checked. The 'User Login' field contains 'John Doe' and the 'Password' field is masked with dots. A 'Check Connection' button is visible in the bottom right of the dialog, along with 'OK' and 'Cancel' buttons.

# Prerequisites: Installing STM32Cube MCU packages

- To download new STM32 MCU packages, follow the steps below:
  - Select ❶ « [Help > Manage embedded software packages](#) » to open the Embedded Software Packages Manager
  - Click the ❷ checkbox to select STM32L4 package then select ❸ « [Install](#) » to start the download and installation





# Board Selection

- Select ❶ « Board Selector » tab search for ❷ « NUCLEO-L496ZG » in Commercial Part Number
- Select ❸ « NUCLEO-L496ZG » kit and click on ❹ « Start Project »

The screenshot shows the STM32CubeIDE Board Selector interface. The 'Board Selector' tab is active. In the 'Commercial Part Number' field, 'NUCLEO-L496ZG' is entered. The 'PRODUCT INFO' section shows details for the NUCLEO-L496ZG board. The 'MEMORY' section shows configuration options for Ext. Flash, Ext. EEPROM, and Ext. RAM. The 'FEATURES' section shows options for Embedded Sensor and User Button. The 'Boards List' table at the bottom shows two items: NUCLEO-L496ZG and NUCLEO-L496ZG-P. The 'Start Project' button is highlighted in the top right corner.

1. Board Selector tab

2. Commercial Part Number field

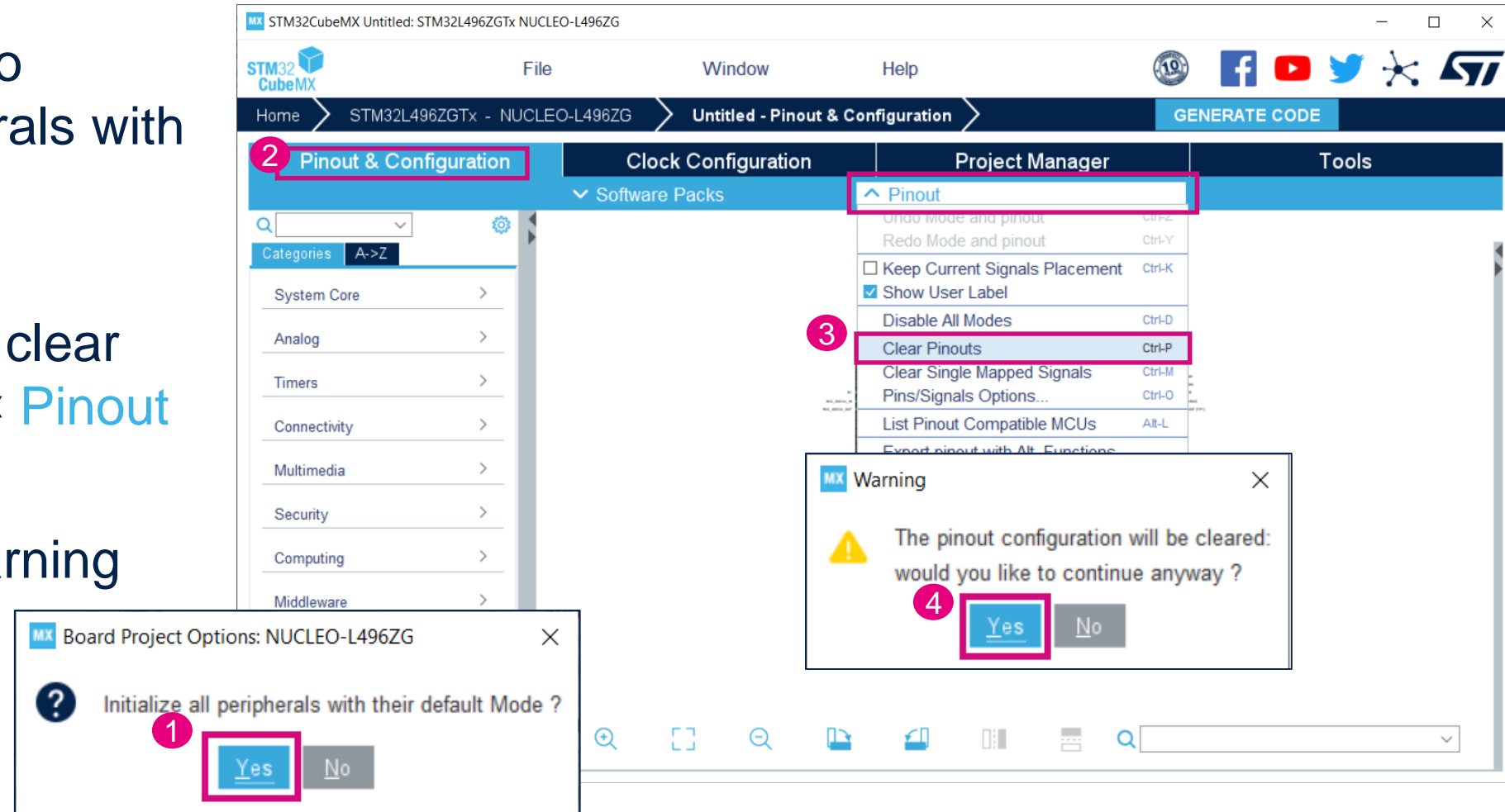
3. Boards List table

4. Start Project button

	Overview	Commercial Part No	Type	Marketing Stat...	Unit Price (US\$)	Mounted Device
☆		NUCLEO-L496ZG	Nucleo-144	Active	19.0	STM32L496ZGT3
☆		NUCLEO-L496ZG-P	Nucleo-144	Active	20.0	STM32L496ZGT6P

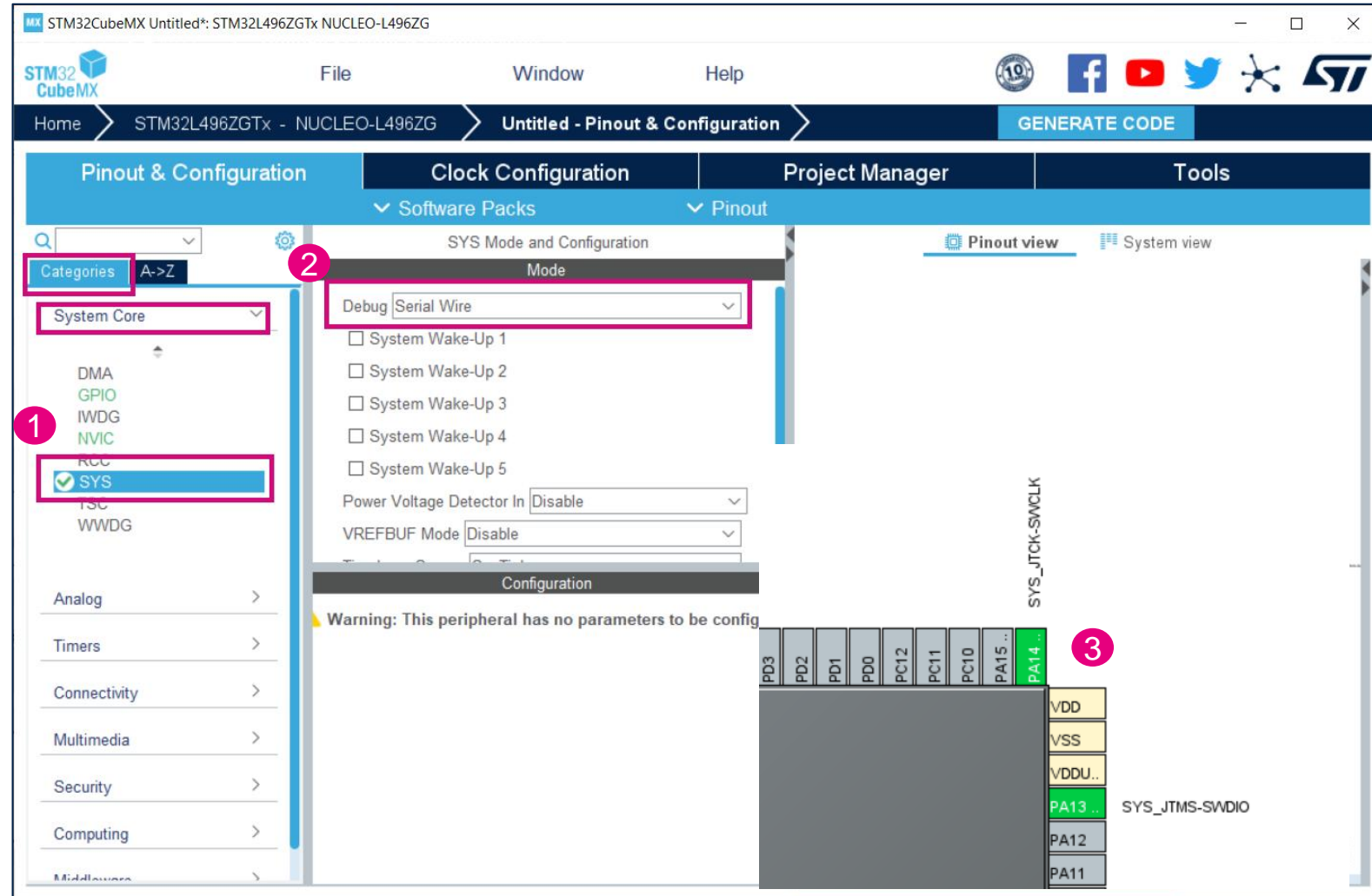
# Set Up Clear Pinouts

- Click on ❶ « Yes » to initialize all peripherals with their default Mode
- In ❷ « Pinout and Configuration » tab clear the pinout using ❸ « Pinout > Clear Pinouts »
- Click ❹ « Yes » to warning

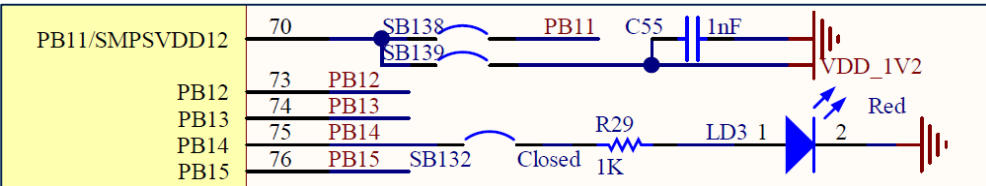


# Enable Debug Pins

- Click ❶ « Categories > System Core > SYS »
- Enable ❷ « Serial Wire » in Debug
- ❸ PA13, and PA14 will turn to GREEN.

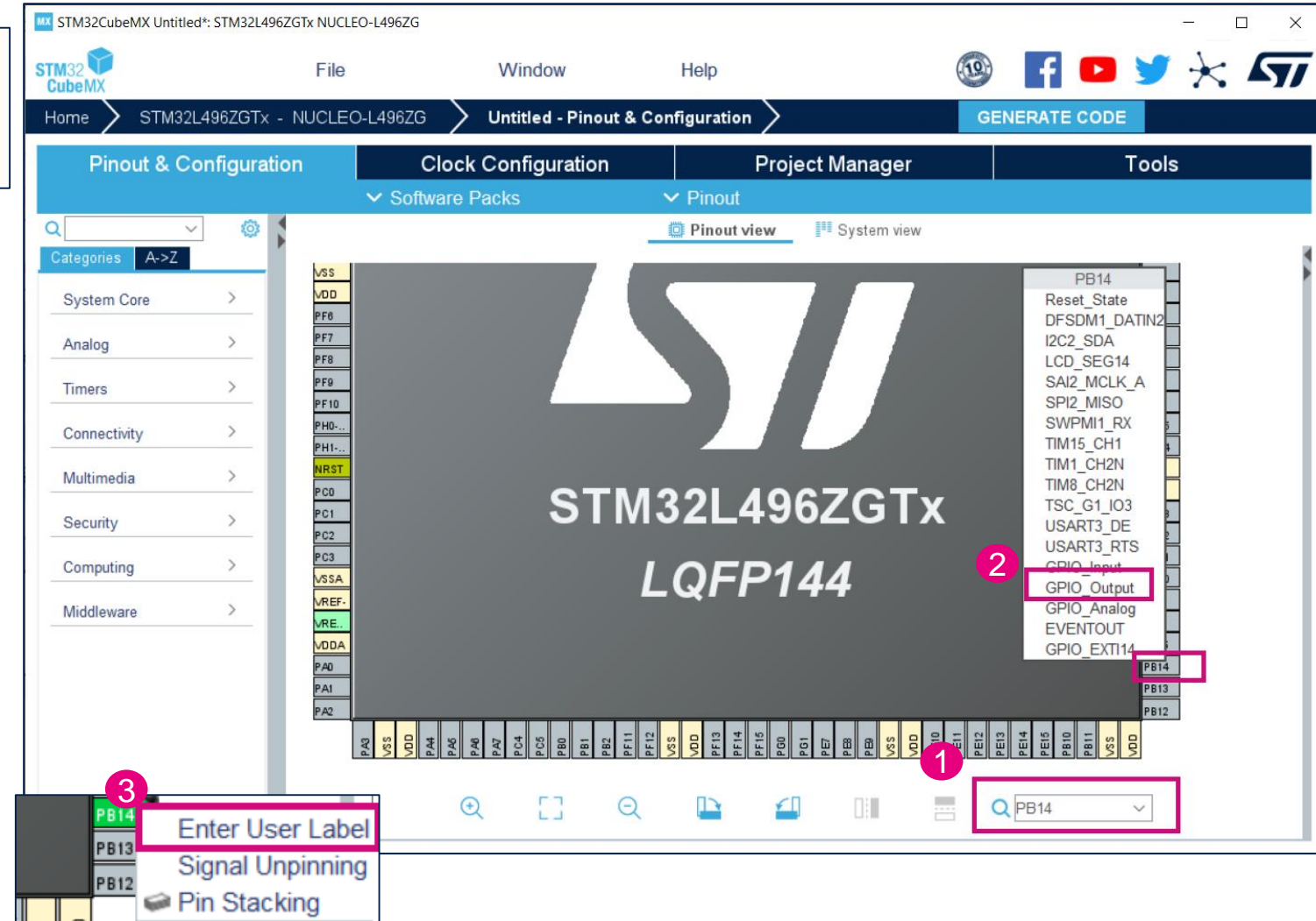


# Enable GPIO Pin



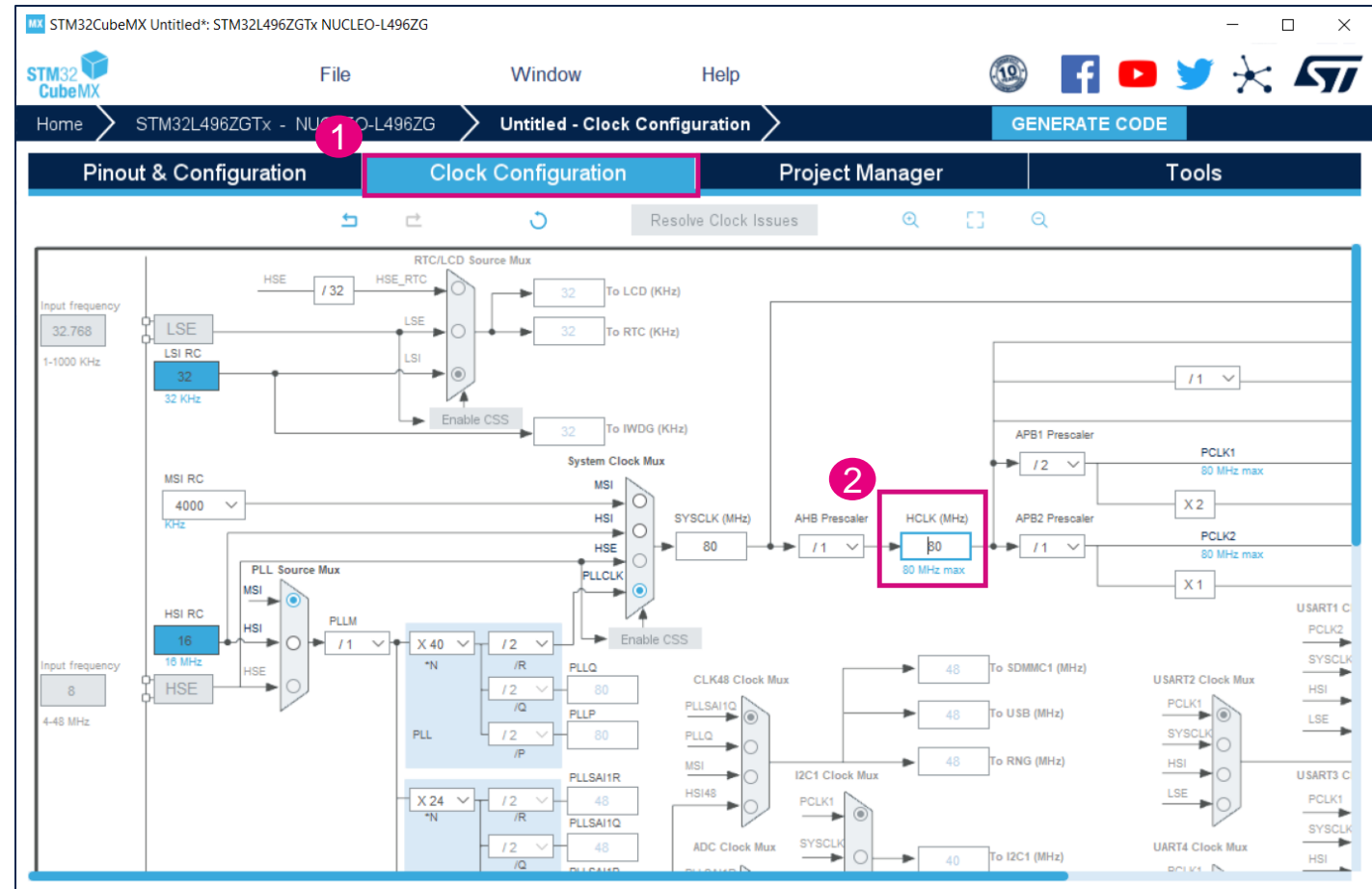
Schematic

- In Pinout view search ❶ « **PB14** » and then Left-click on the highlighted pin and set as ❷ « **GPIO\_Output** » to drive the LED
- Right-click pin **PB14** and Enter User Label & give name ❸ « **LD3** » as on the board.



# Clock configuration

- In ❶ « Clock Configuration » tab Set HCLK (MHz) to ❷ « 80 », then press Enter.
- Click OK when Clock Wizard message pop out to automatically find the correct clock sources
- The appropriate clock source and PLL values will be set automatically





# Code generation project settings

- In Project tab **1** « Name project » e.g GPIO\_Toggle
- Browse for project location
- Select **2** « MDK-ARM » toolchain
- **3** Review the exact MCU type and library version

The screenshot shows the STM32CubeMX Project Manager interface. The 'Project Manager' tab is selected, and the 'Project' section on the left sidebar is highlighted. The settings are as follows:

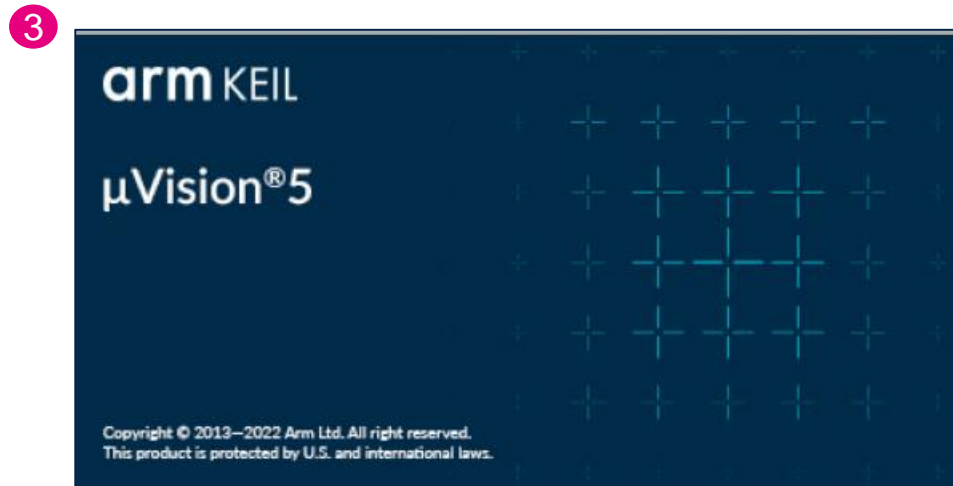
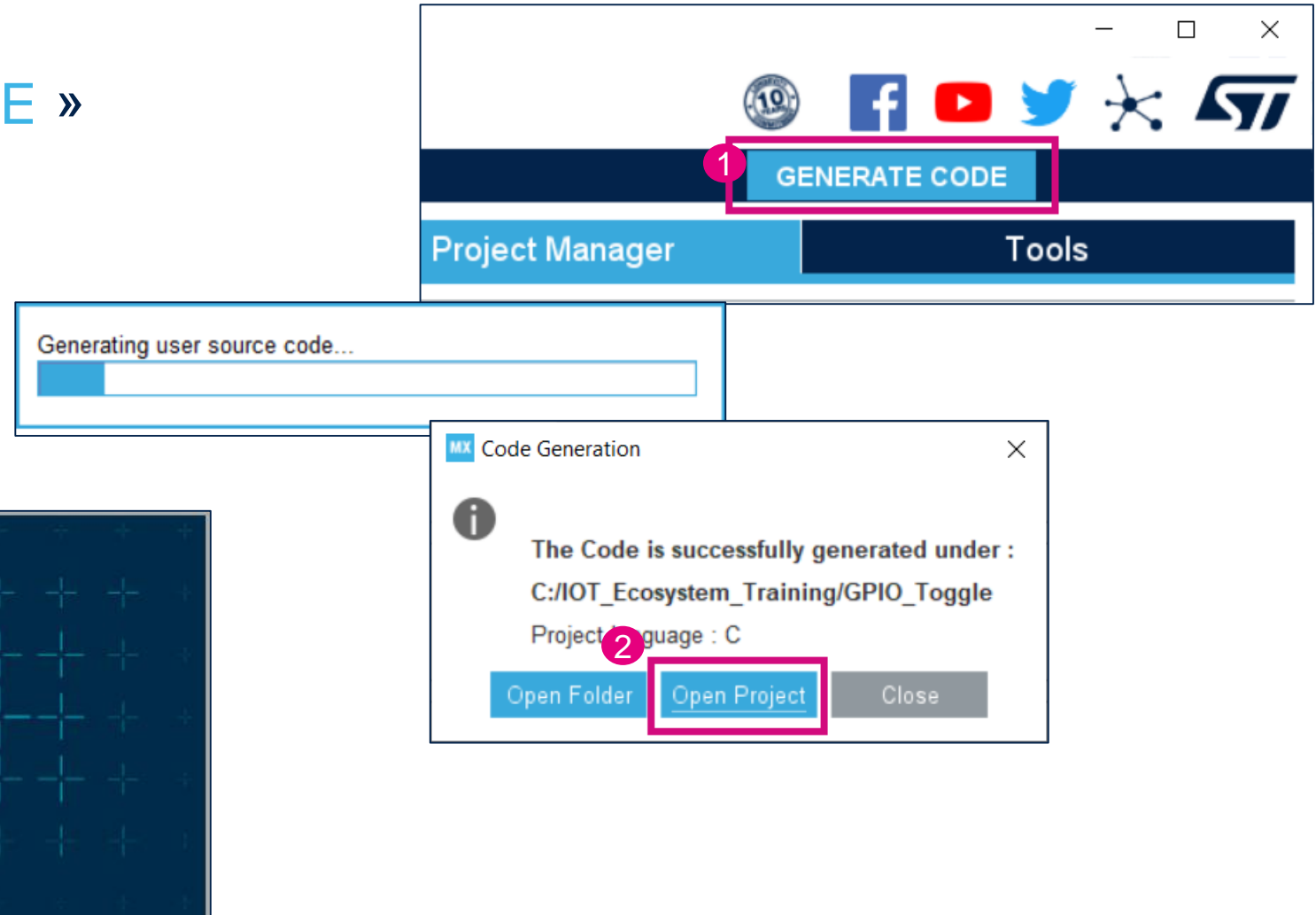
Section	Setting	Value
Project Settings	Project Name	GPIO_Toggle
	Project Location	C:\VOT_Ecosystem_Training
	Application Structure	Advanced
	Toolchain / IDE	MDK-ARM
Linker Settings	Minimum Heap Size	0x200
	Minimum Stack Size	0x400
Thread-safe Settings	Thread-safe Locking Strategy	Default – Mapping suitable strategy depending on RTOS selection.
	Thread-safe Settings	<input type="checkbox"/> Enable multi-threaded support
MCU and Firmware Package	Mcu Reference	STM32L496ZGTx
	Firmware Package Name and Version	STM32Cube FW_L4 V1.17.2
	Firmware Relative Path	C:/Users/s boharapi/STM32Cube/Repository/STM32Cube_FW_L4_V1.17.2

Additional options visible: ☐ Do not generate code, ☒ Use latest, ☒ Use Default Firmware Location.

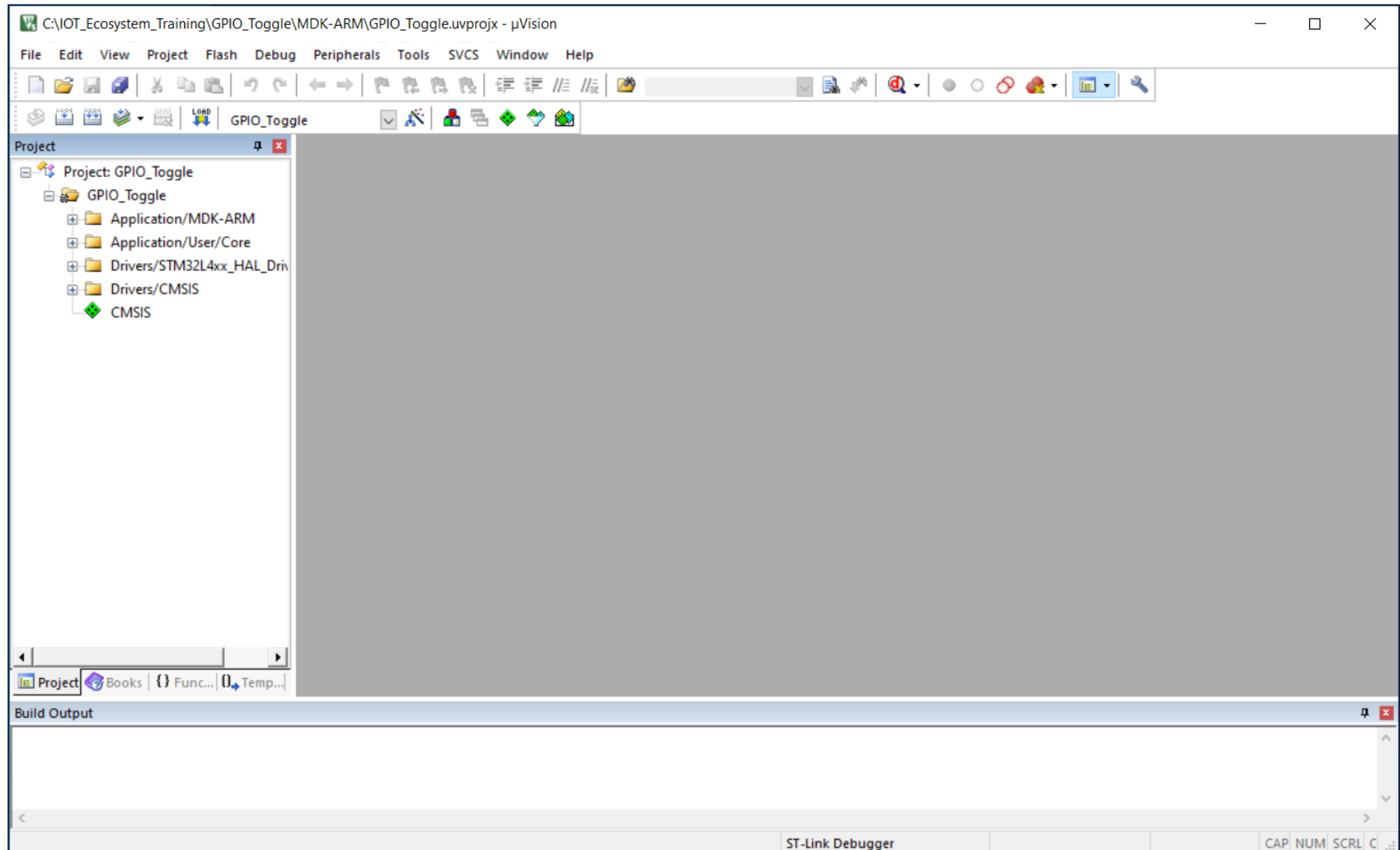


# Generate Code

- Click on ❶ « **GENERATE CODE** » at the top right corner.
- Select ❷ « **Open Project** »
- Then ❸ « **arm KEIL** » will open

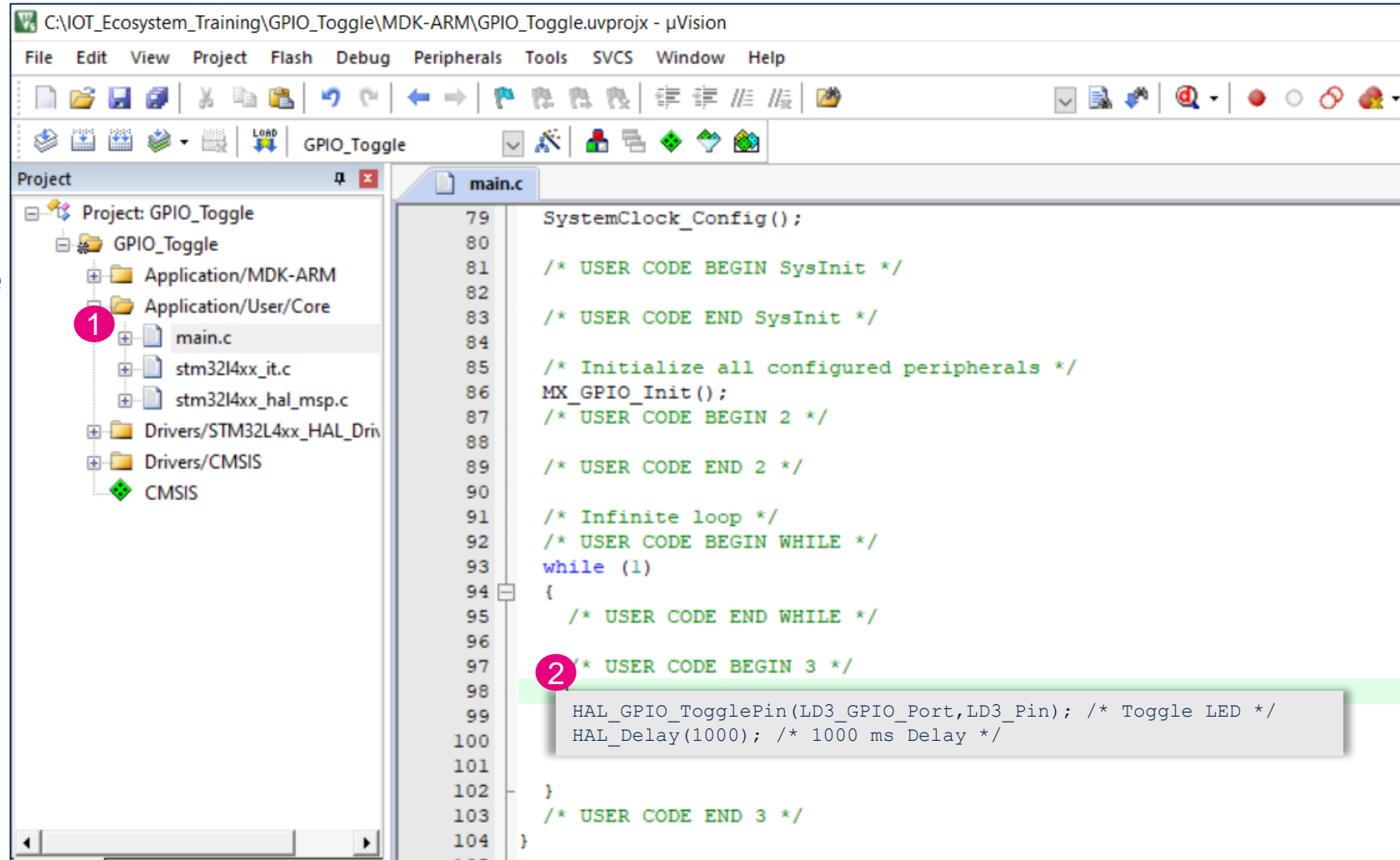


# Keil Project Window



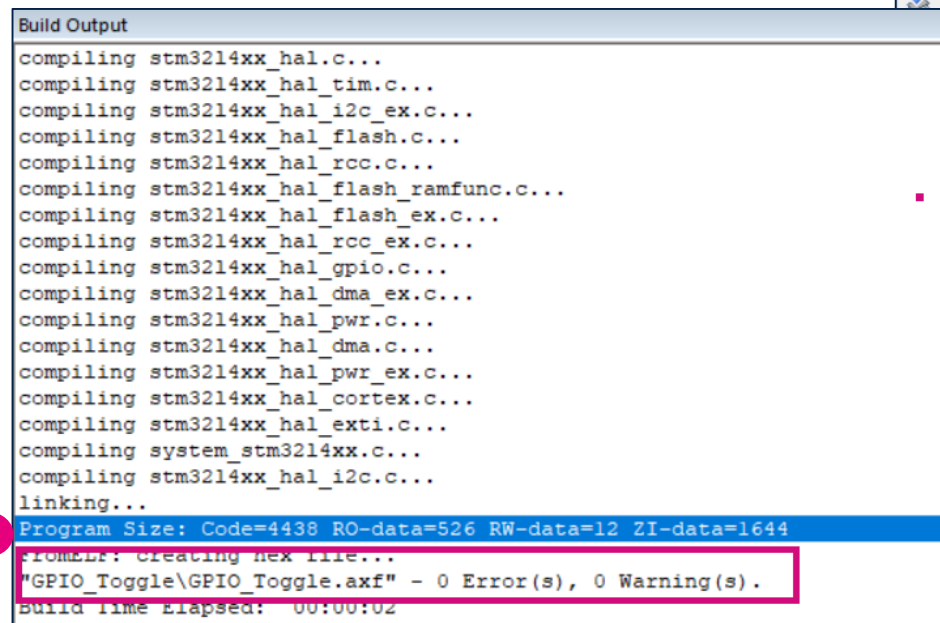
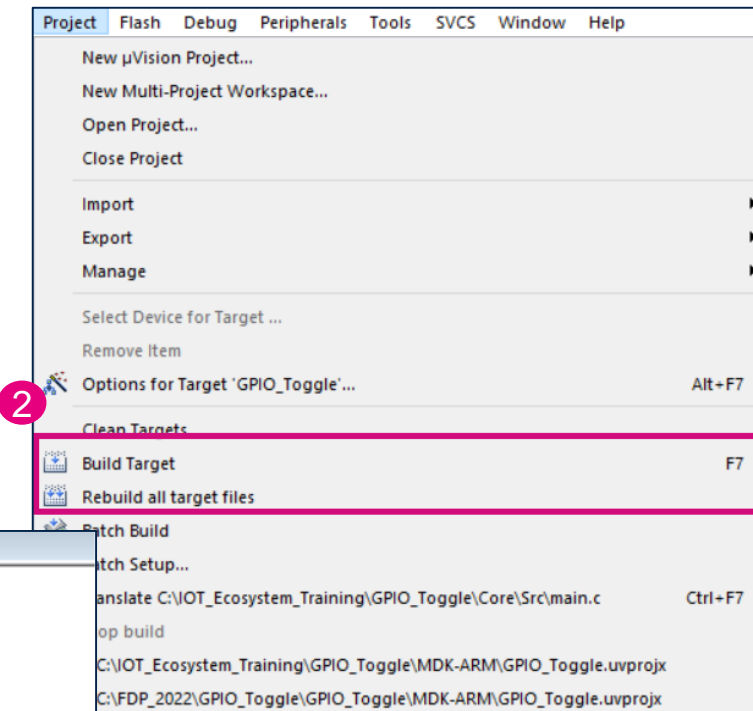
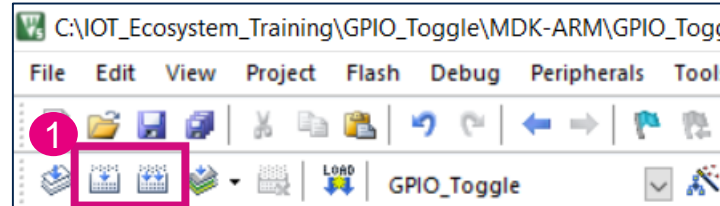
# Add User Code

- Browse the ❶ « `main.c` » file in Project
- Add the user code to toggle GPIOs driving the LED to the specific ❷ « `USER CODE` » BEGIN\END section in the `main.c`
- Ensure you save project

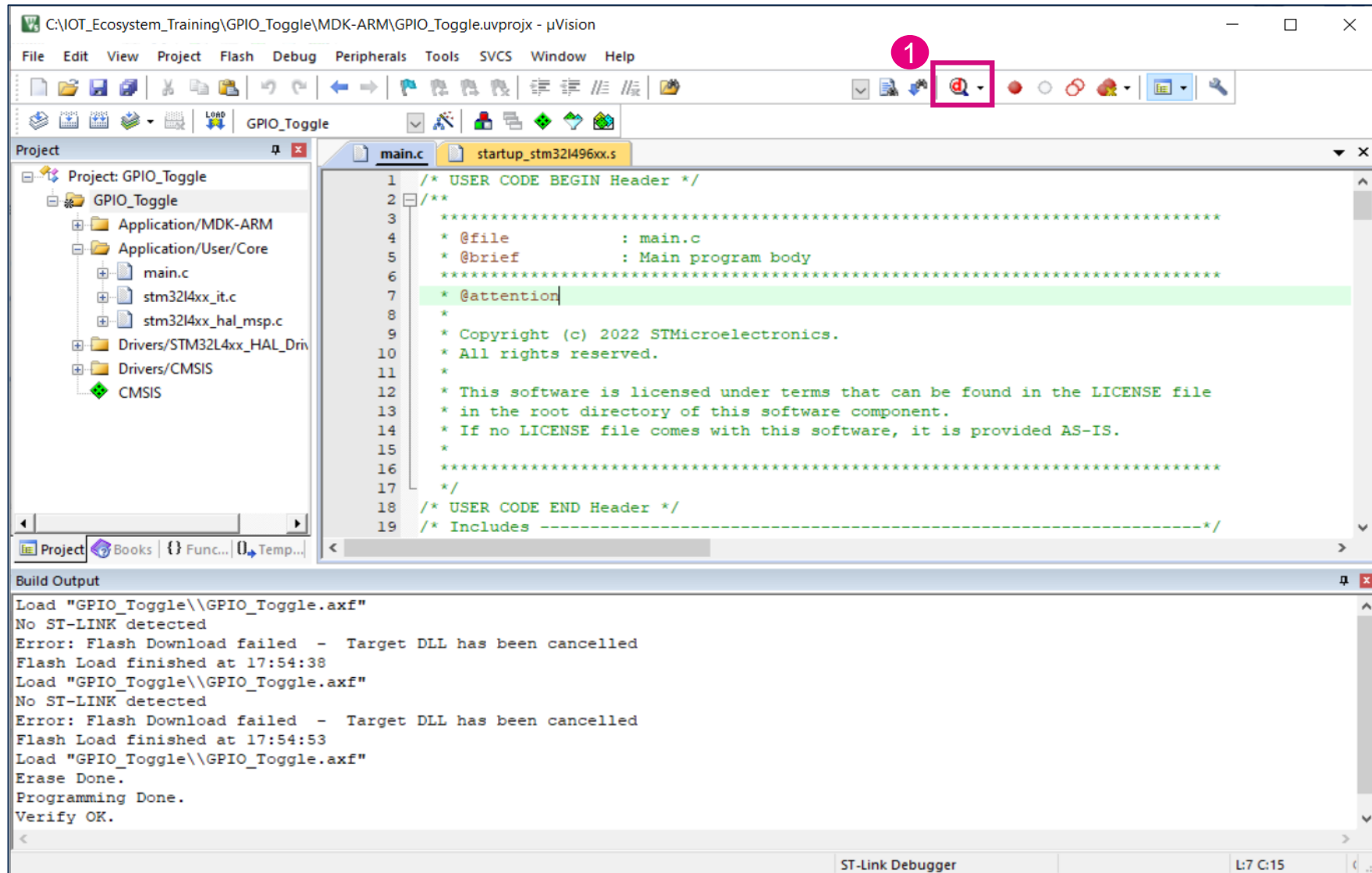


# Build Project

- Click ❶ « Build/Rebuild » icon or ❷ « Project > Build Target / Rebuild all target files »
- Build status (Error and warning) is display in « ❸ Build Output » window
- Ensure that no errors and warning



# Starting the Debugger

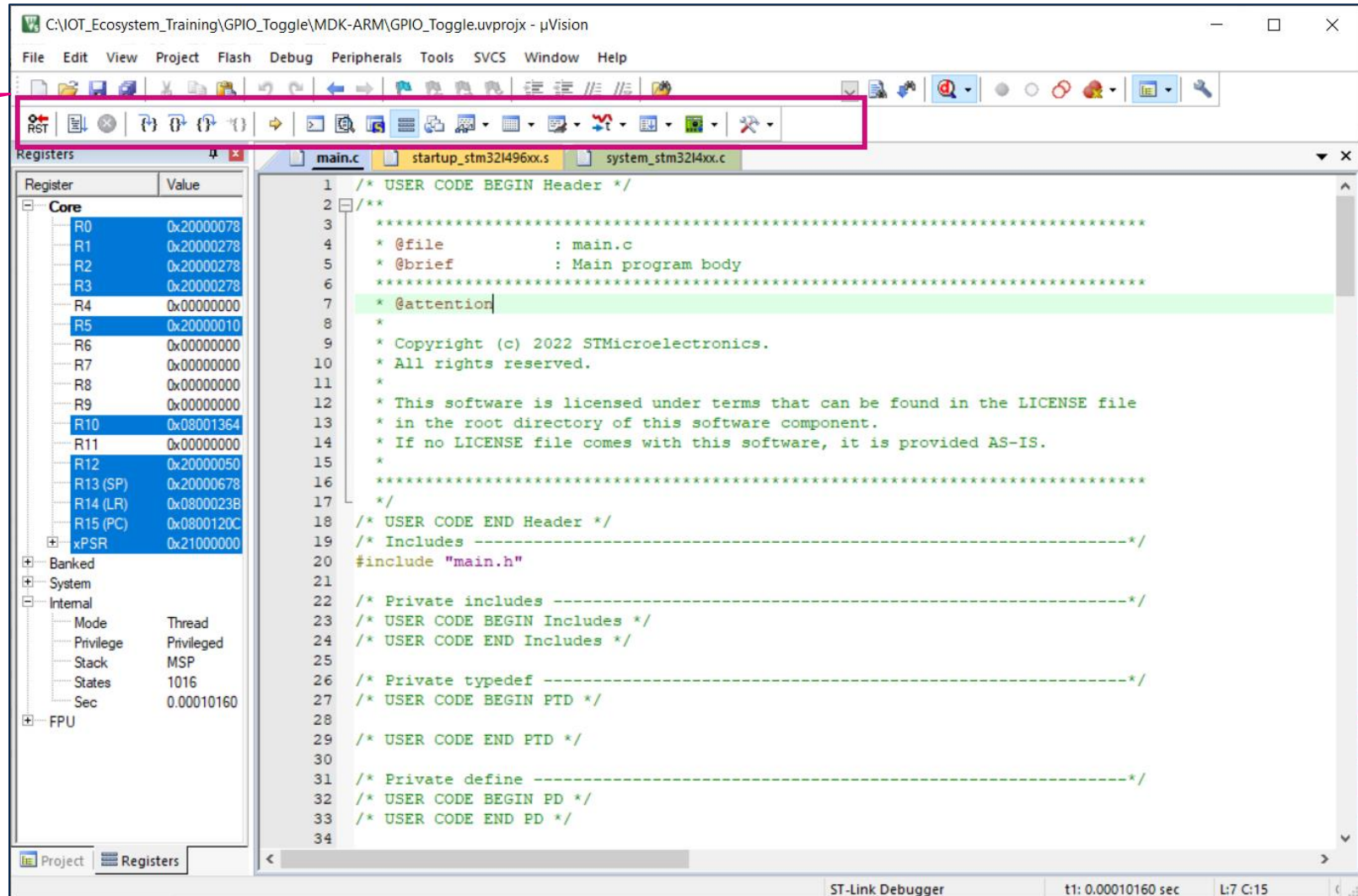


# Run Application

Debug Toolbar to reset, run/stop, step in, step out option



- Click to Run (F5) the application and see output on LD3 of NUCLEO board.





# Live demo 2: GPIO Toggling using STM32CubeIDE

# Generating Code for STM32CubeIDE

- Open STM32CubeMX (.IOC file) of same project created in last project.
- Navigate to Project Manager tab

The screenshot shows the STM32CubeMX Project Manager tab. The top panel lists files in a table:

File	Modified	Type	Size
Core	7/15/2022 12:09 PM	File folder	
Drivers	7/15/2022 12:09 PM	File folder	
MDK-ARM	7/15/2022 2:05 PM	File folder	
.mxproject	7/15/2022 1:47 PM	MXPROJECT File	9 KB
GPIO_Toggle.ioc	7/15/2022 1:47 PM	STM32CubeMX	8 KB

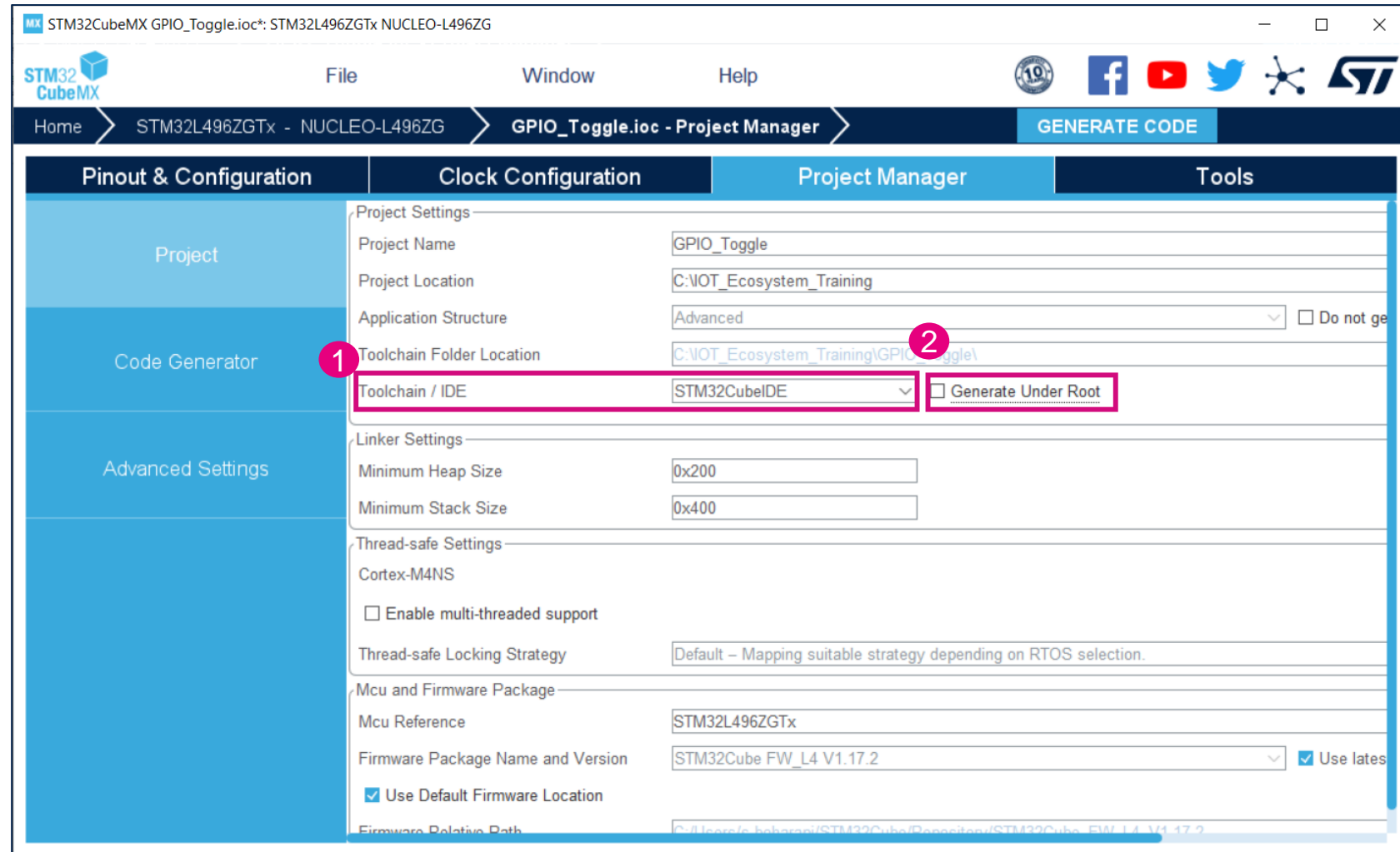
Below the table, the Project Manager tab is active, showing the following settings:

- Project Settings**
  - Project Name: GPIO\_Toggle
  - Project Location: C:\NOT\_Ecosystem\_Training
  - Application Structure: Advanced
  - Toolchain Folder Location: C:\NOT\_Ecosystem\_Training\GPIO\_Toggle\
  - Toolchain / IDE: MDK-ARM
  - Min Version: V5.27
- Linker Settings**
  - Minimum Heap Size: 0x200
  - Minimum Stack Size: 0x400
- Thread-safe Settings**
  - Cortex-M4NS
    - ☐ Enable multi-threaded support
  - Thread-safe Locking Strategy: Default – Mapping suitable strategy depending on RTOS selection.
- Mcu and Firmware Package**
  - Mcu Reference: STM32L496ZGTx
  - Firmware Package Name and Version: STM32Cube\_FW\_L4 V1.17.2
  - ☒ Use Default Firmware Location
  - Firmware Relative Path: Full to code: backslash\STM32Cube\Drivers\STM32Cube\_FW\_L4 V1.17.2

# Change project setting for STM32CubeIDE

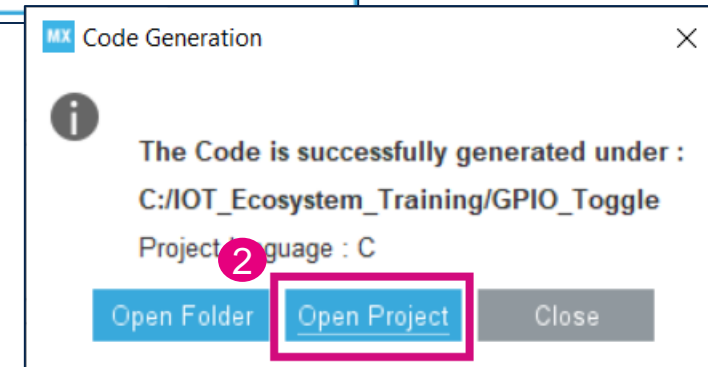
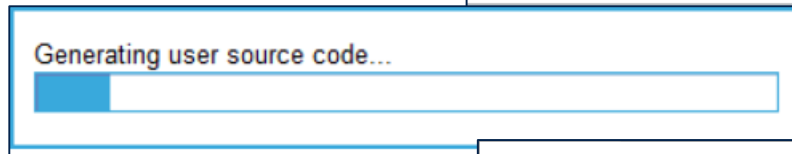
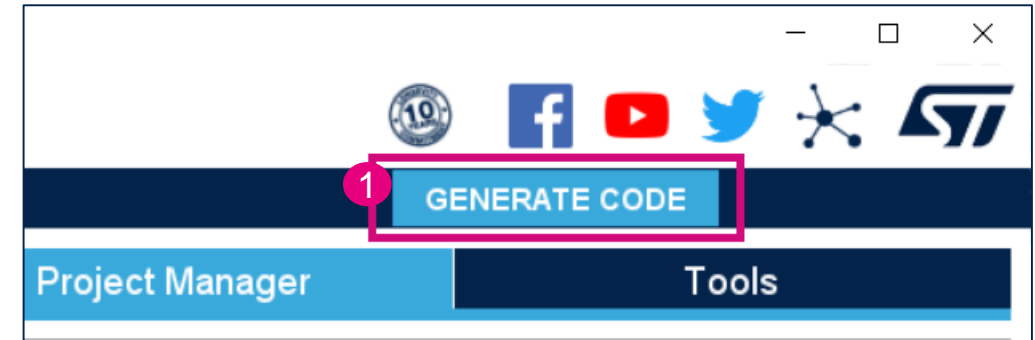
- Change Toolchain/IDE MDK-ARM to ❶ « STM32CubeIDE »
- Uncheck ❷ « Generate Under root » checkbox
- Keep other setting as it is

**Note:** If you generate the code to generate Under Root may corrupt old generated project



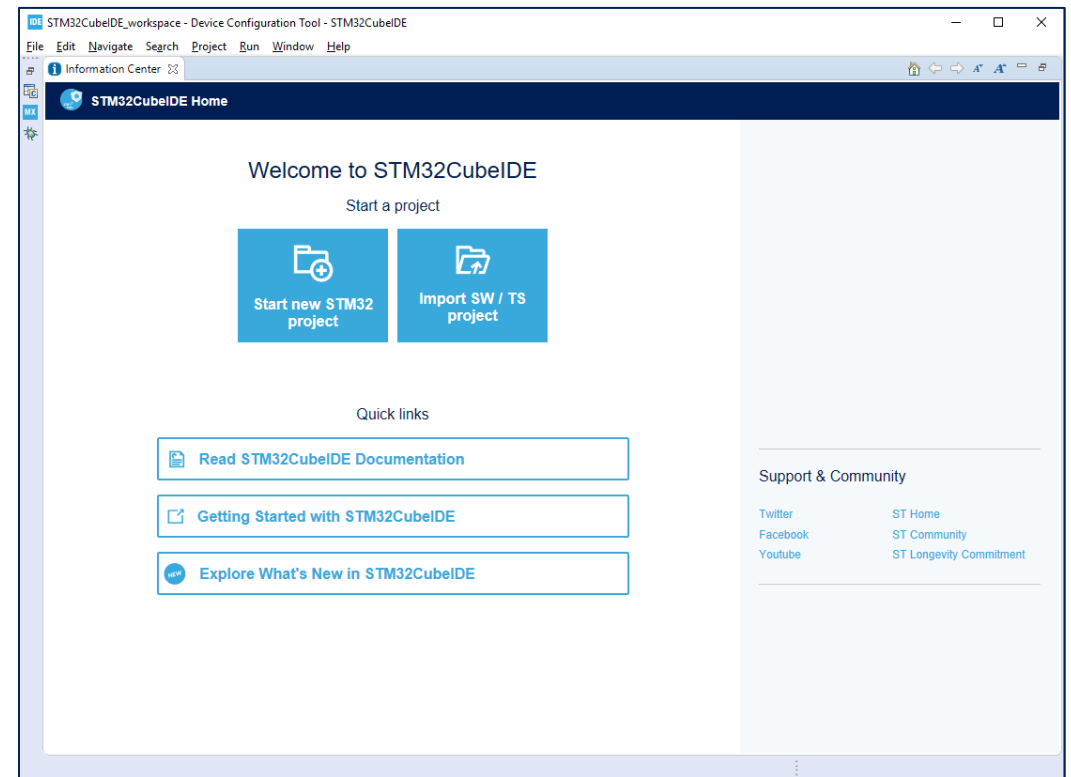
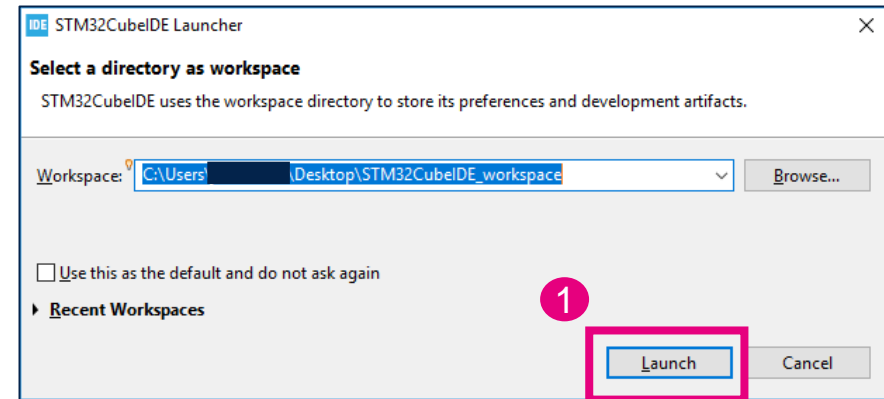
# Generate Code

- Click on ❶ « **GENERATE CODE** » at the top right corner.
- Select ❷ « **Open Project** »
- Then ❸ « **STM32CubeIDE** » will open



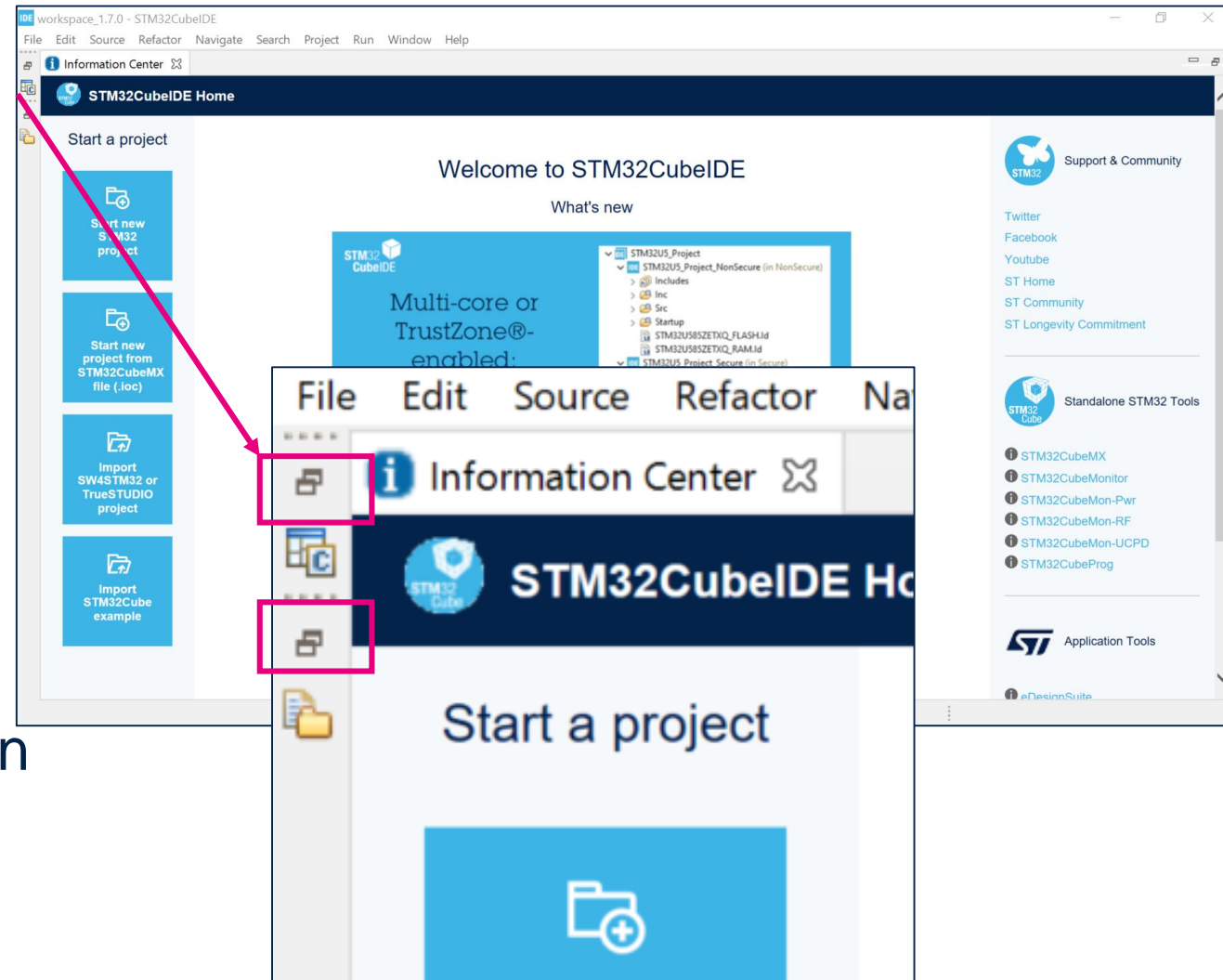
# Launch STM32CubeIDE

- Set the desired workspace folder for all the projects to be created by the STM32CubeIDE
- Click on ❶ « **Launch** »
- STM32CubeIDE Welcome Page
- The Welcome Page allows the user to quickly reach information regarding the product, and how to use it
- Can be reached any time via Help > Information Center



# First STM32CubeIDE start

- [Information Center] panel contains links to various information related to STM32CubeIDE.
  - Close the [Information Center] panel
  - C/C++ perspective will appear
- Please note that the content from the [Information Center] panel might take some time to appear.
- Click on the highlighted button to open the Project Explorer.





# C/C++ Perspective

The screenshot displays the STM32CubeIDE interface with four views highlighted by pink callouts:

- Editor view:** The central area showing the C code in `main.c`. The code includes comments for initialization and a while loop for GPIO toggling.
- Project Explorer view:** The left sidebar showing the project structure, including folders like `GPIO`, `Includes`, `Core`, and files like `GPIO.ioc`, `GPIO.Debug.launch`, and `STM32L476RGTX_FLASH.Id`.
- Outline view:** The right sidebar showing the project's symbol table, listing functions like `main.h`, `SystemClock_Config(void)`, `MX_GPIO_Init(void)`, and `main(void)`.
- Build analyzer view:** The bottom right pane showing the `Build Analyzer` for `GPIO.elf`. It displays a table of memory regions.

The bottom status bar shows the current line and column: `106 : 19 : 2830`.

Region	Start address	End address	Size	Usage
RAM	0x20000000	0x20018000	96 KB	94

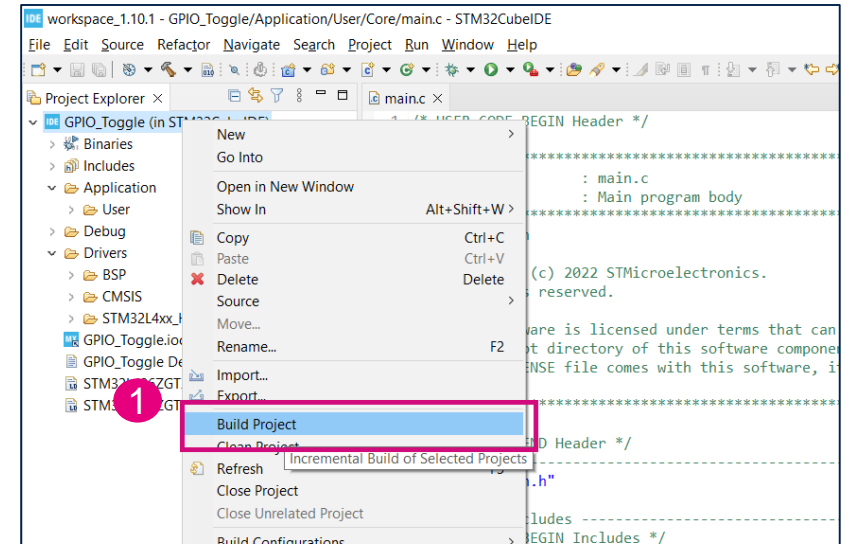
# Build Project

- Right click on project and click on ❶ « Build Project »
- ❷ Build status is displayed in the Console window
- ❸ Compile errors are displayed in the Problems window.

❸

Problems × Tasks Console Build Analyzer Static Stack Analyzer		
0 items		
Description	Resource	Path

❷



```
Problems × Tasks Console × Build Analyzer Static Stack Analyzer
CDT Build Console [GPIO_Toggle]
16:17:27 **** Incremental Build of configuration Debug for project GPIO_Toggle ****
make -j8 all
arm-none-eabi-gcc "C:/IOT_Ecosystem_Training/GPIO_Toggle/Core/Src/main.c" -mcpu=cortex-m4
arm-none-eabi-gcc -o "GPIO_Toggle.elf" @"objects.list" -mcpu=cortex-m4 -T"C:\IOT_Ecosyst
Finished building target: GPIO_Toggle.elf

arm-none-eabi-size GPIO_Toggle.elf
arm-none-eabi-objdump -h -S GPIO_Toggle.elf > "GPIO_Toggle.list"
text data bss dec hex filename
6216 20 1572 7808 1e80 GPIO_Toggle.elf
Finished building: default.size.stdout

Finished building: GPIO_Toggle.list
```

# Debug Perspective

## Debug toolbar

1 2 3 4 5 6 7 8 9

1. Restart
2. Resume
3. Suspend
4. Terminate
5. Terminate & Relaunch
6. Step Into
7. Step Over
8. Step Return
9. Instruction Stepping mode


Breakpoint view  
Existing breakpoints are  
shown or configured here

SFR view  
Peripheral registers  
contents are  
displayed here

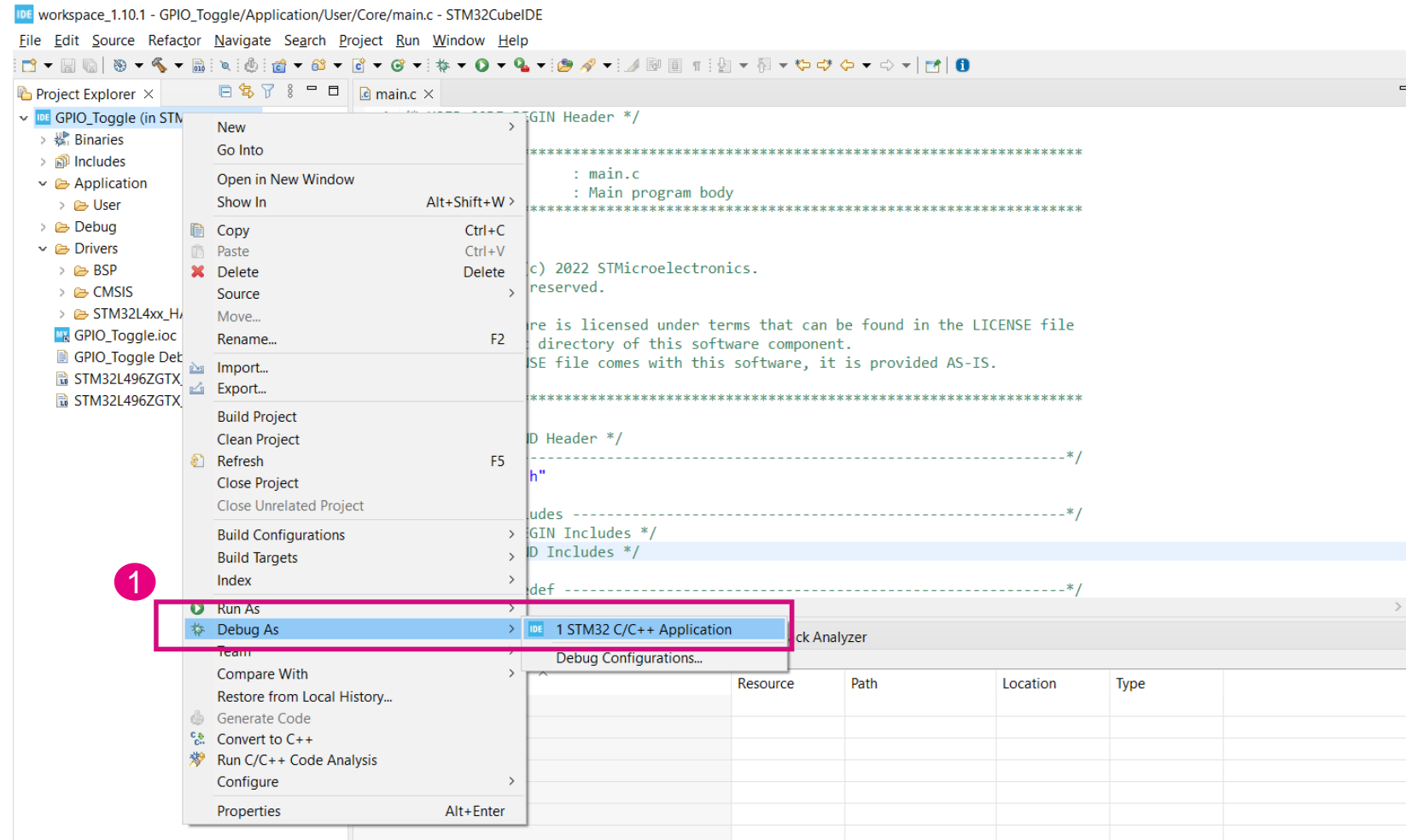
Fault Analyzer view  
Display information if  
a system fault occurs

Terminal view  
Serial port output  
can be directed  
to this view

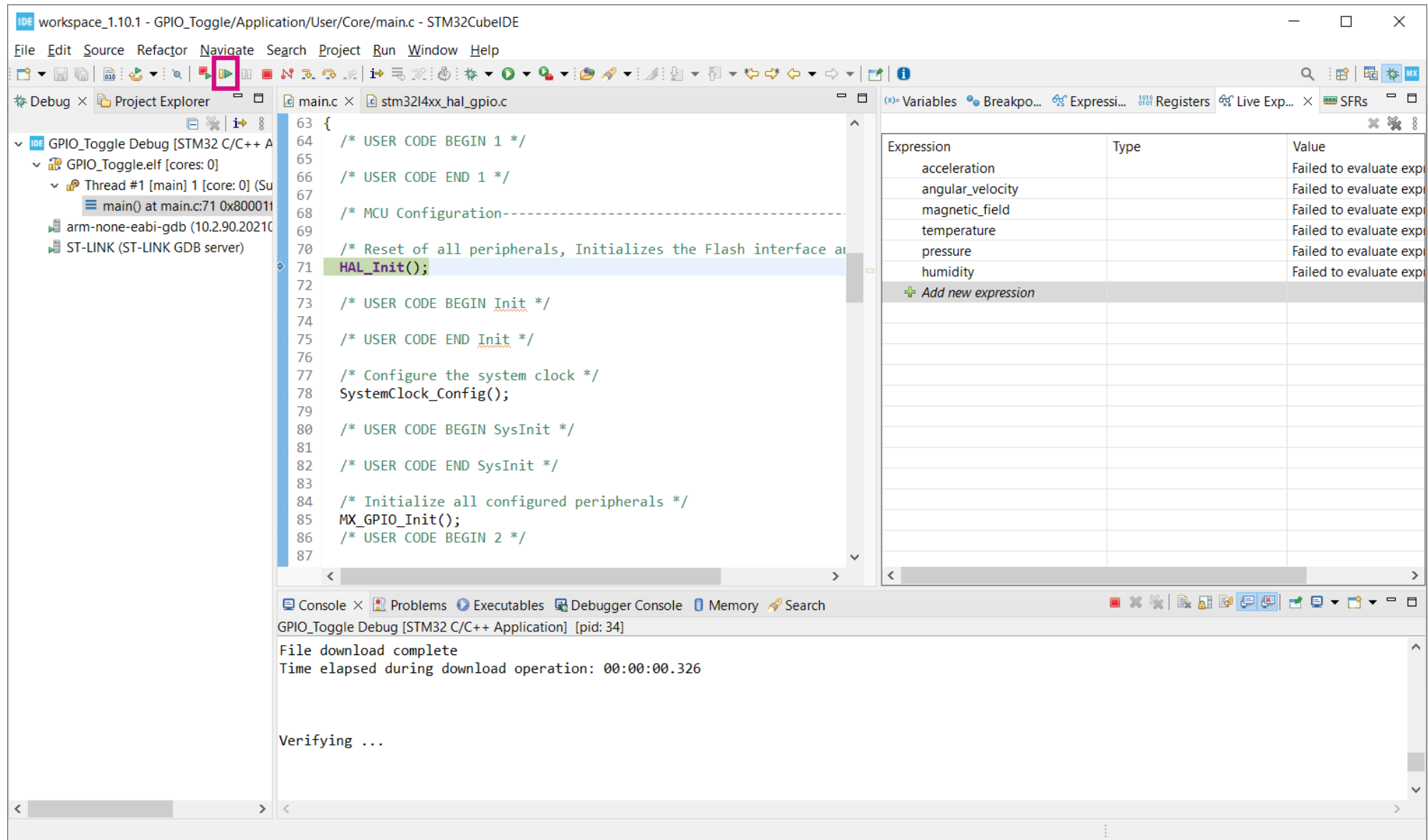
# Starting the Debugger

- In Project Explorer right click on project and click on **1**  **Debug as STM32 C/C++ Application** » to start the debugging

Note: STM32CubeIDE can detect if the current ST-Link firmware on Nucleo board is old. In this case, follow the instructions to update the ST-Link firmware. Once done, start the debug session again.



# Click Resume to run application



The screenshot displays the STM32CubeIDE interface. The main editor shows the file `main.c` with the following code:

```
63 {
64     /* USER CODE BEGIN 1 */
65
66     /* USER CODE END 1 */
67
68     /* MCU Configuration-----*/
69
70     /* Reset of all peripherals, Initializes the Flash interface and
71        the other components. */
72     HAL_Init();
73
74     /* USER CODE BEGIN Init */
75
76     /* USER CODE END Init */
77
78     /* Configure the system clock */
79     SystemClock_Config();
80
81     /* USER CODE BEGIN SysInit */
82
83     /* USER CODE END SysInit */
84
85     /* Initialize all configured peripherals */
86     MX_GPIO_Init();
87 }
```

The Project Explorer on the left shows the project structure:

- GPIO\_Toggle Debug [STM32 C/C++ Application]
- GPIO\_Toggle.elf [cores: 0]
- Thread #1 [main] 1 [core: 0] (Stopped at main.c:71 0x800011)
- arm-none-eabi-gdb (10.2.90.20210)
- ST-LINK (ST-LINK GDB server)

The right-hand side of the interface shows the Variables, Breakpoints, Expressions, Registers, Live Expressions, and SFRs panels. The Variables panel is currently empty, showing a table with columns for Expression, Type, and Value.

Expression	Type	Value
acceleration		Failed to evaluate expression
angular_velocity		Failed to evaluate expression
magnetic_field		Failed to evaluate expression
temperature		Failed to evaluate expression
pressure		Failed to evaluate expression
humidity		Failed to evaluate expression
+ Add new expression		

The bottom panel shows the Console output:

```
GPIO_Toggle Debug [STM32 C/C++ Application] [pid: 34]
File download complete
Time elapsed during download operation: 00:00:00.326

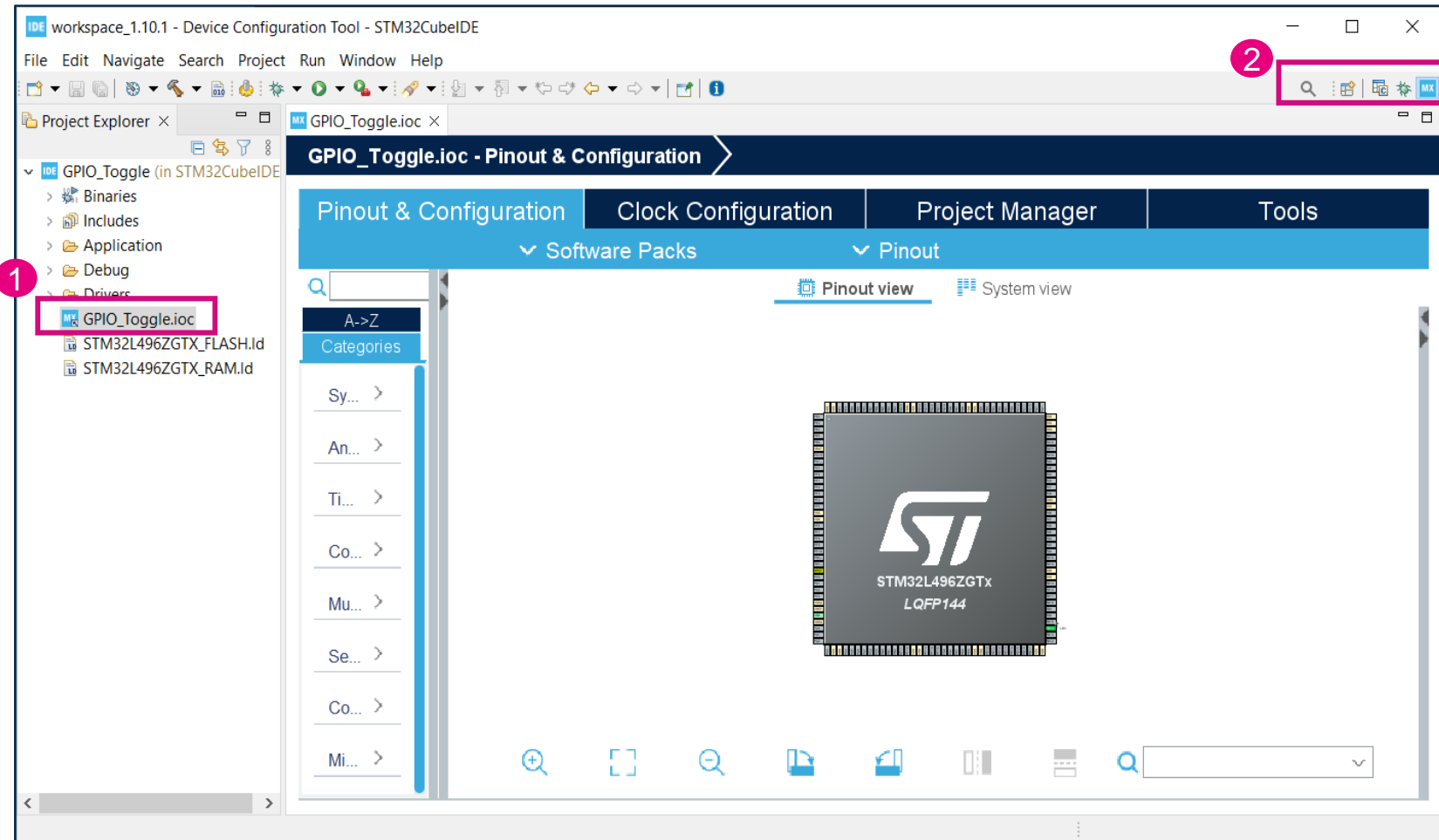
Verifying ...
```

# Live Demo 3: Using X-Cube Package



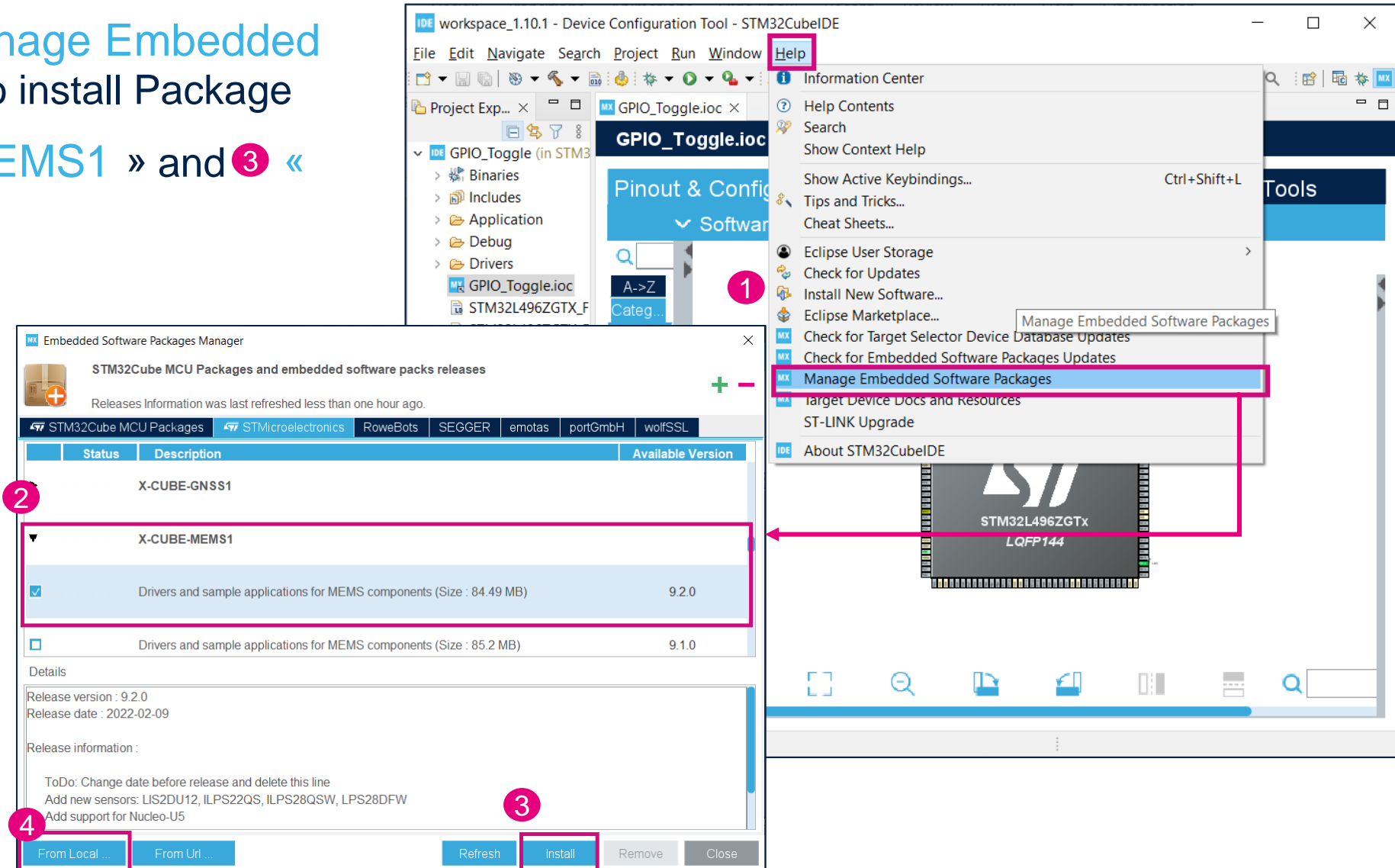
# Opening STM32CubeMX in STM32CubeIDE

- Double Click ❶ « GPIO.IOC » file
- This will open the STM32CubeMx perspective
- ❷ Switch between the perspective with their icons on top right corner



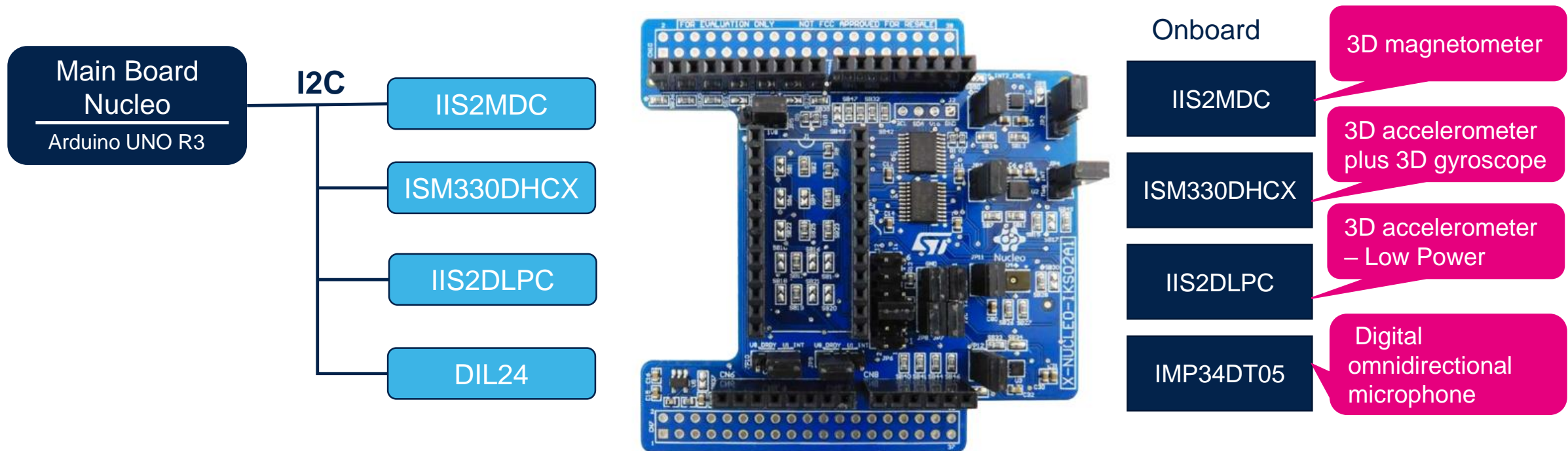
# Download and Install X-CUBE Packages

- Click to ❶ « **Help > Manage Embedded Software Package** » to install Package
- Search ❷ « **X-Cube-MEMS1** » and ❸ « **Install** » it
- ❹ Note: If you downloaded package form ST site you can install from click From Local tab



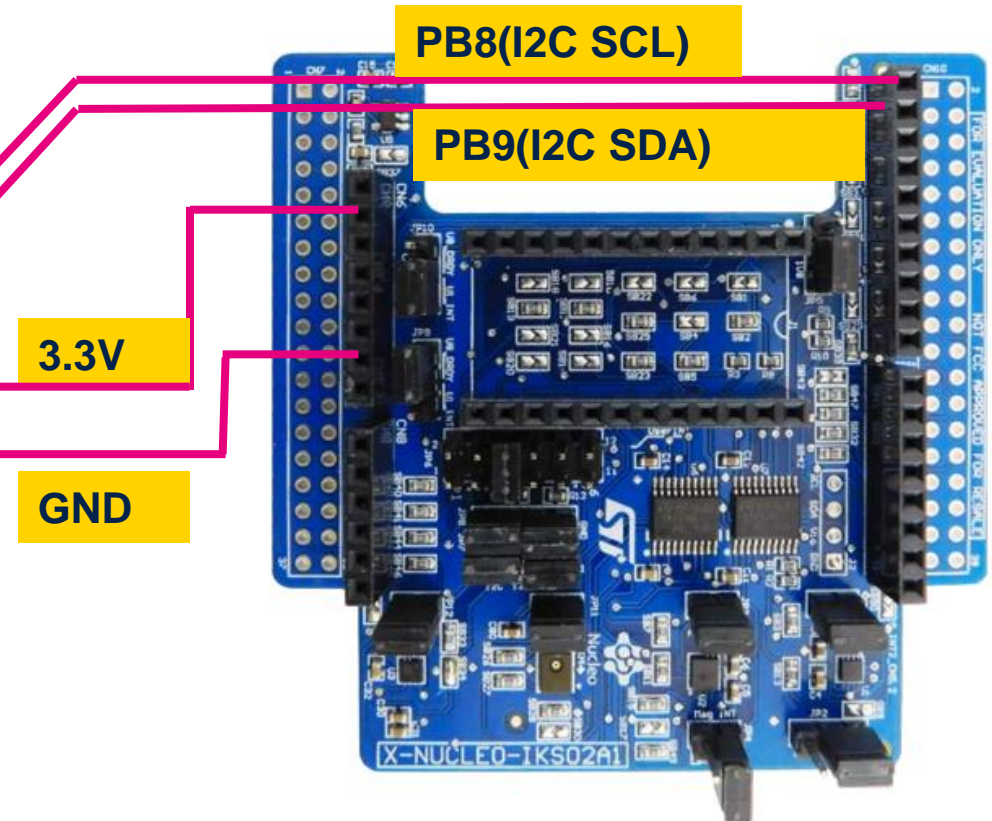
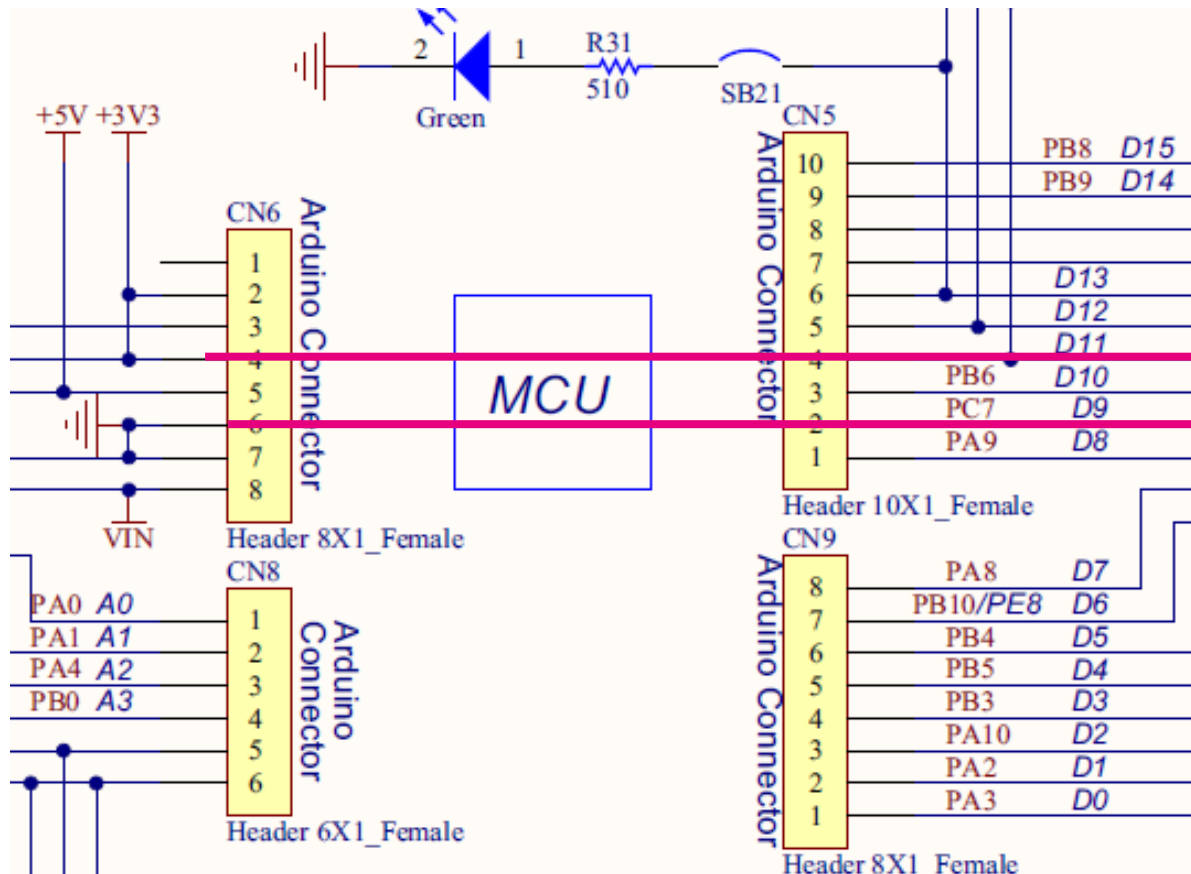
# Sensor IKS02A1

- Diagram of « IKS02A1 board »



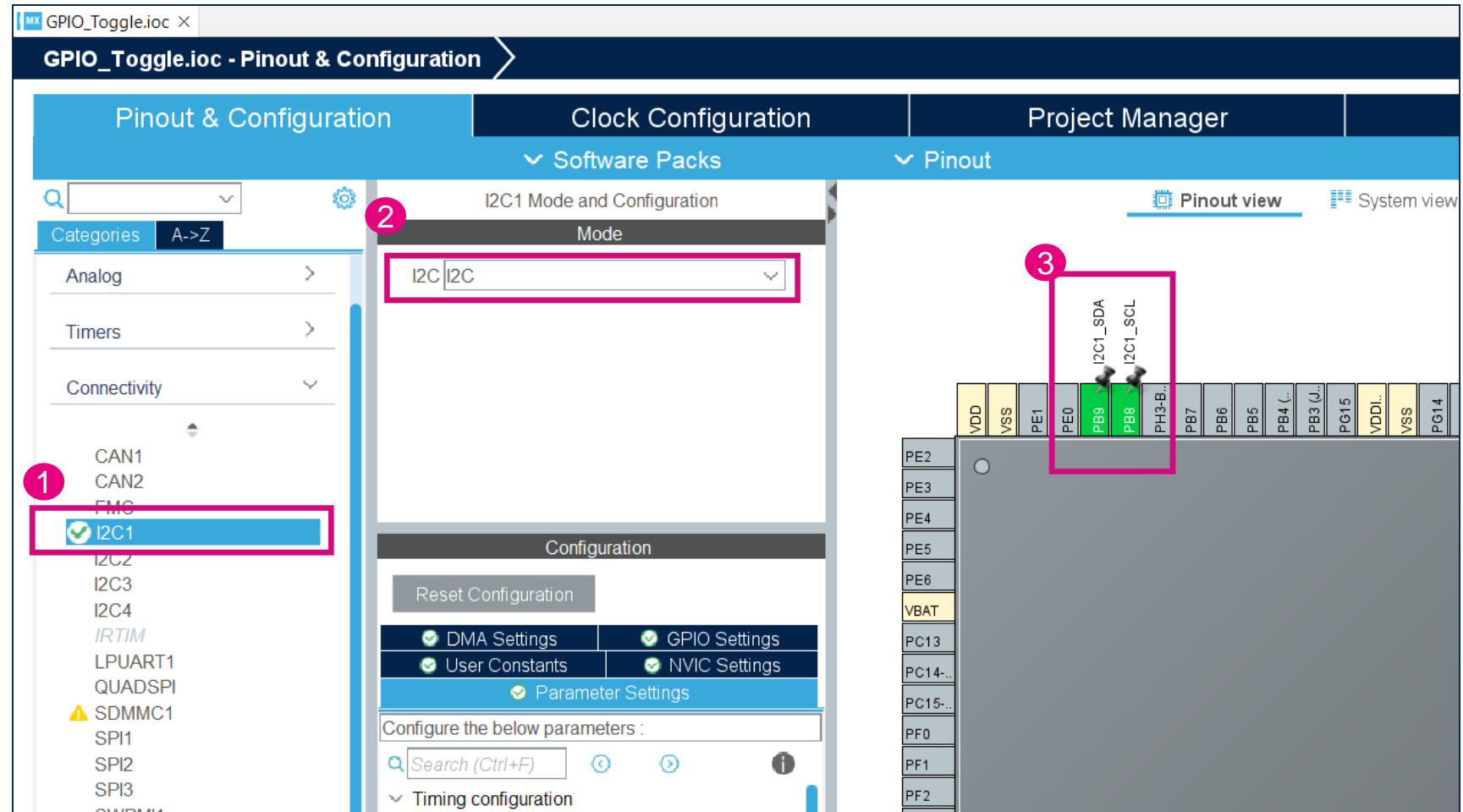
# Pin connection

## Nucleo-L496ZG and X-NUCLEO-IKS02A1



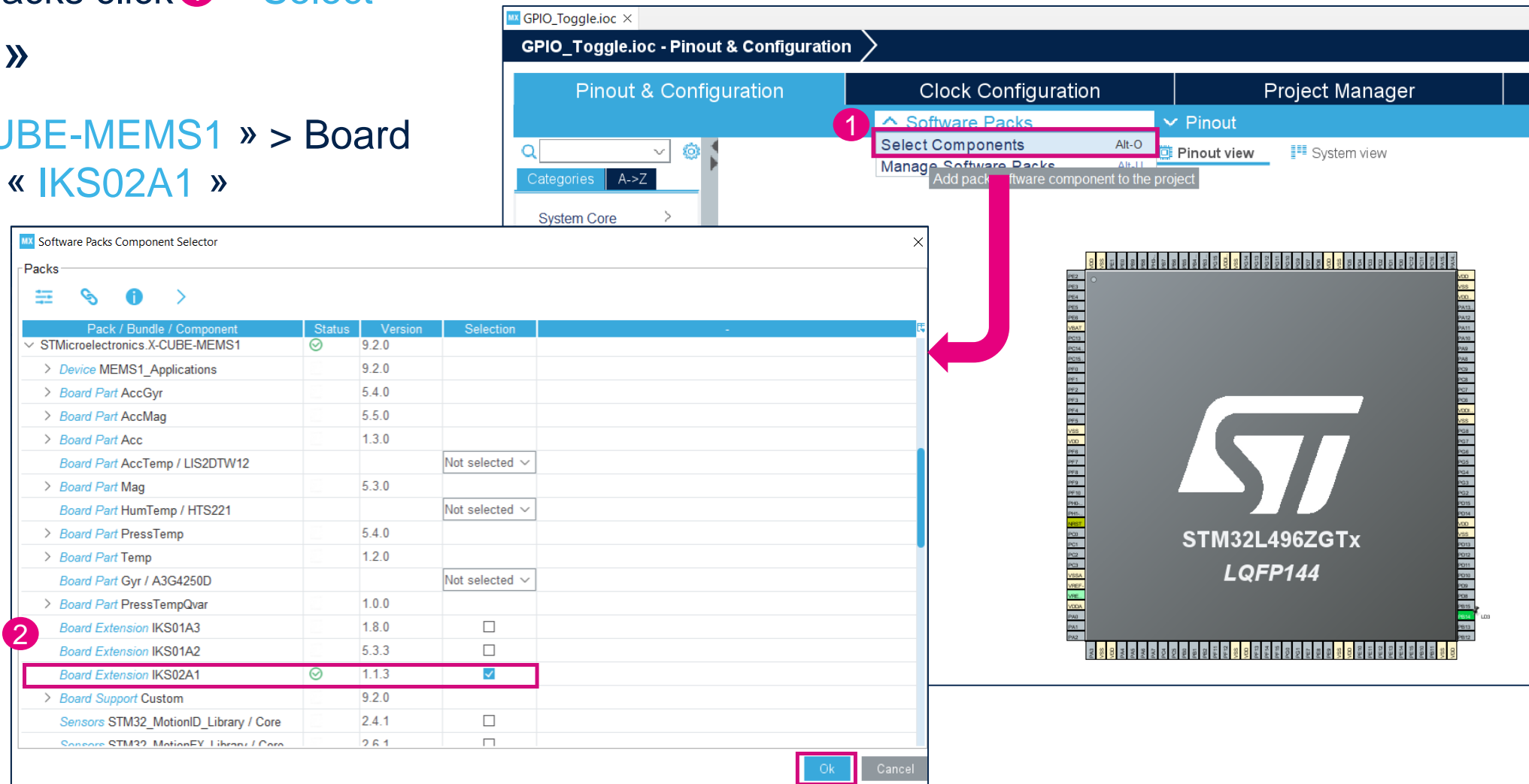
# Configure communication

- Select ❶ « I2C » in ❷ « Connectivity » & Mode « I2C »
- ❸ Manually select PB8 as I2C1\_CLK  
PB9 as I2C1\_SDA
- No change: use default (Standard Mode/100kHz)



# Adding Software Pack

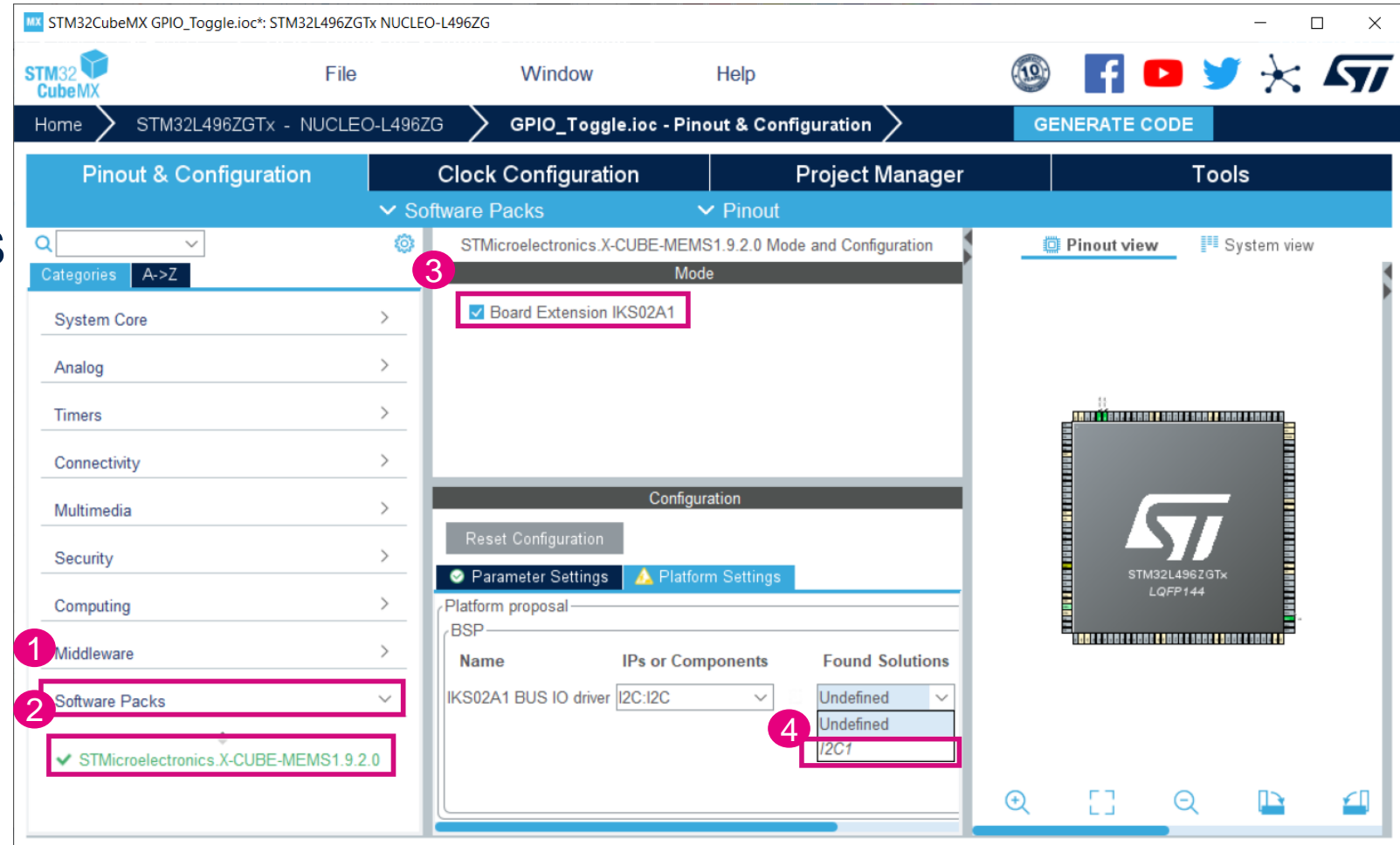
- In Software Packs click ❶ « Select Components »
- Select « X-CUBE-MEMS1 » > Board Expansion ❷ « IKS02A1 »





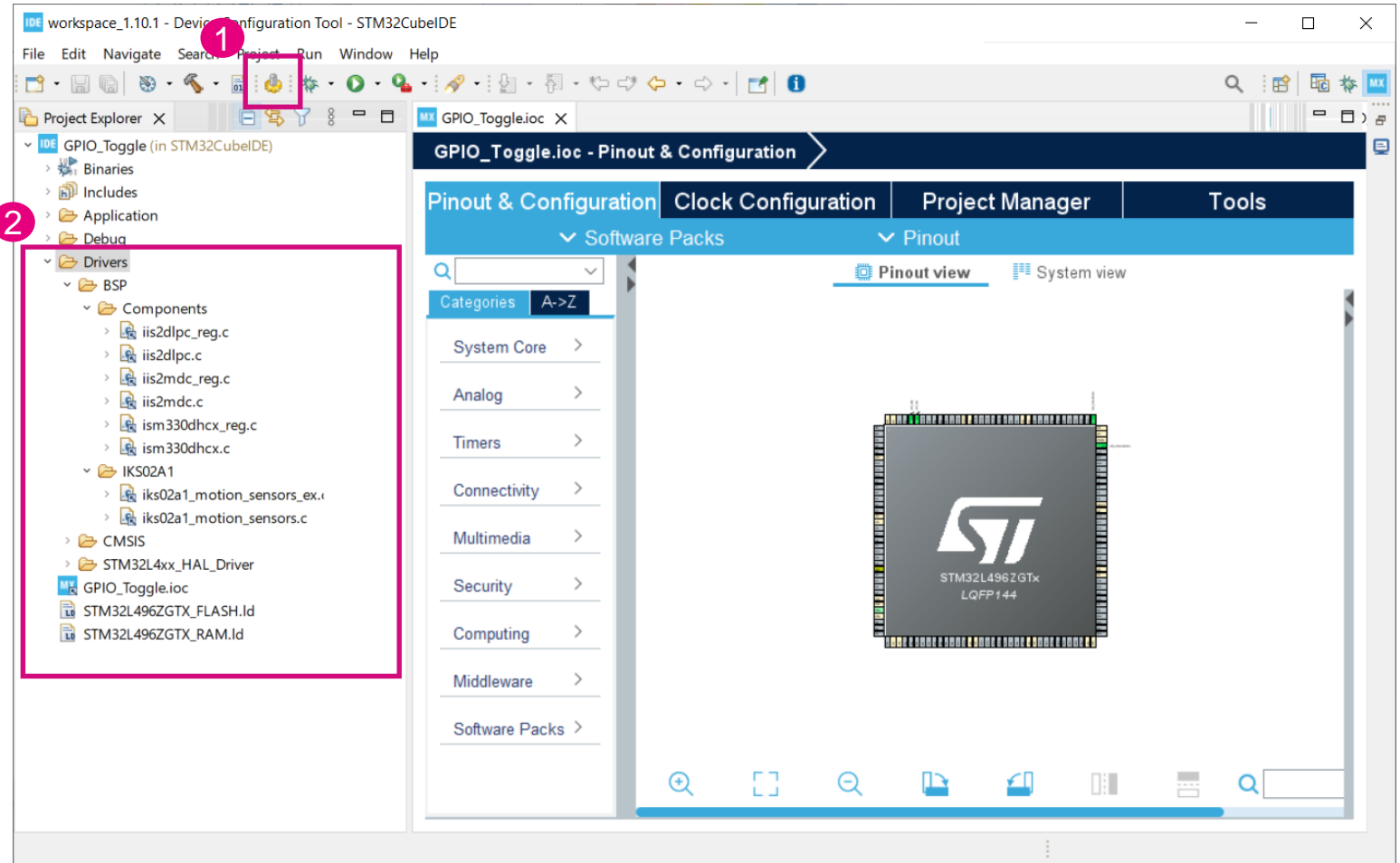
# Activate Board Extension MEMs

- 1 Select Software Packs
- 2 Select STMicroelectronics X-CUBE-MEMS1
- 3 Activate Board Extension MEMS
- 4 Configure Platform Setting : Select I2C1



# Generate Code using STM32CubeIDE

- 1 Generate code by clicking on «Generate code icon»
- 2 All related sensor drivers will be added to your workspace



# Adding code (main.c)

- ① Add header files
- ② Add sensor variable

```
GPIO_Toggle.ioc  *main.c X
16 *****
17 */
18 /* USER CODE END Header */
19 /* Includes -----*/
20 #include "main.h"
21
22 ① /* Private includes -----*/
23 /* USER CODE BEGIN Includes */
24 #include "iks02a1_motion_sensors.h"
25
26 /* USER CODE END Includes */
27
28 /* Private typedef -----*/
29 /* USER CODE BEGIN PTD */
30
31 /* USER CODE END PTD */
32
33 /* Private define -----*/
34 /* USER CODE BEGIN PD */
35 /* USER CODE END PD */
36
37 /* Private macro -----*/
38 /* USER CODE BEGIN PM */
39
40 /* USER CODE END PM */
41
42 /* Private variables -----*/
43
44 ② /* USER CODE BEGIN PV */
45 IKS02A1_MOTION_SENSOR_Axes_t acceleration;
46 IKS02A1_MOTION_SENSOR_Axes_t angular_velocity;
47 IKS02A1_MOTION_SENSOR_Axes_t magnetic_field;
48
49 /* USER CODE END PV */
50
```

# Adding code (main.c)

- ① Add sensor Initialization code
- ② Add sensor read code

```
MX GPIO_Toggle.ioc  *main.c X
82  /* Configure the system clock */
83  SystemClock_Config();
84
85  /* USER CODE BEGIN SysInit */
86
87  /* USER CODE END SysInit */
88
89  /* Initialize all configured peripherals */
90  ① MX_GPIO_Init();
91  /* USER CODE BEGIN 2 */
92
93  IKS02A1_MOTION_SENSOR_Init(IKS02A1_IIS2DLPC_0, MOTION_ACCELERO );
94  IKS02A1_MOTION_SENSOR_Init(IKS02A1_ISM330DHCX_0, MOTION_GYRO);
95  IKS02A1_MOTION_SENSOR_Init(IKS02A1_IIS2MDC_0, MOTION_MAGNETO);
96
97  /* USER CODE END 2 */
98
99  /* Infinite loop */
100 /* USER CODE BEGIN WHILE */
101 while (1)
102 {
103     /* USER CODE END WHILE */
104
105     ② /* USER CODE BEGIN 3 */
106     IKS02A1_MOTION_SENSOR_GetAxes(IKS02A1_IIS2DLPC_0, MOTION_ACCELERO, &acceleration);
107     IKS02A1_MOTION_SENSOR_GetAxes(IKS02A1_ISM330DHCX_0, MOTION_GYRO, &angular_velocity);
108     IKS02A1_MOTION_SENSOR_GetAxes(IKS02A1_IIS2MDC_0, MOTION_MAGNETO, &magnetic_field);
109 }
110 /* USER CODE END 3 */
```

# Download and debug

The screenshot shows the STM32CubeIDE interface with three numbered steps:

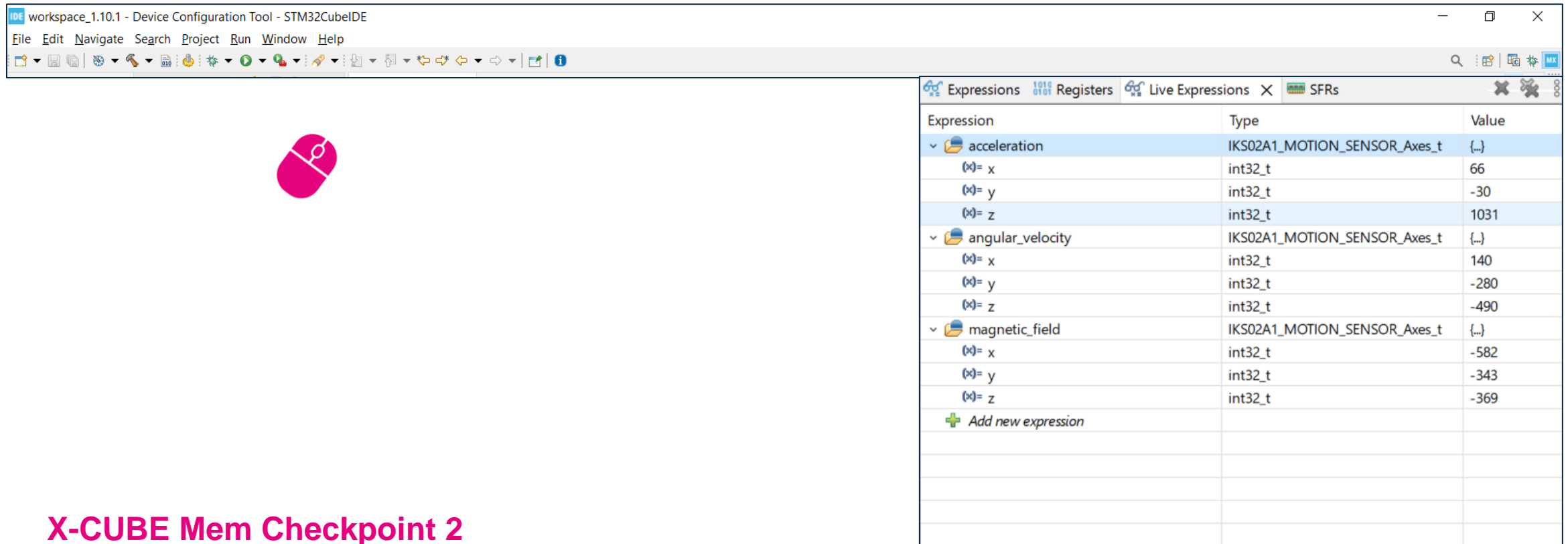
- 1 Right Click**: A right-click context menu is open over the 'GPIO\_Toggle (in STM32CubeIDE)' project in the Explorer. The 'Debug As' option is highlighted.
- 2 Debug application**: The 'Debug As' submenu is open, showing '1 STM32 C/C++ Application' selected.
- 3 Add variable to Live Expression**: The 'Live Expressions' window is open, showing a table of variables. A yellow callout box says 'Add Variables in Live Expressions'.

Expression	Type	Value
acceleration	IKS02A1_MOTION_SENSOR_Axes_t	{...}
(x)= x	int32_t	0
(x)= y	int32_t	0
(x)= z	int32_t	0
angular_velocity	IKS02A1_MOTION_SENSOR_Axes_t	{...}
(x)= x	int32_t	0
(x)= y	int32_t	0
(x)= z	int32_t	0
magnetic_field	IKS02A1_MOTION_SENSOR_Axes_t	{...}
(x)= x	int32_t	0
(x)= y	int32_t	0
(x)= z	int32_t	0
+ Add new expression		

- 1 Right Click
- 2 Debug application
- 3 Add variable to Live Expression

# Monitor sensor values

- Click on the GO icon to Run project: 



The screenshot shows the STM32CubeIDE interface. The top menu bar includes File, Edit, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations, debugging, and monitoring. The main window is divided into two panes. The left pane is empty. The right pane is titled 'Expressions' and contains a table of sensor values.

Expression	Type	Value
acceleration	IKS02A1_MOTION_SENSOR_Axes_t	{...}
x	int32_t	66
y	int32_t	-30
z	int32_t	1031
angular_velocity	IKS02A1_MOTION_SENSOR_Axes_t	{...}
x	int32_t	140
y	int32_t	-280
z	int32_t	-490
magnetic_field	IKS02A1_MOTION_SENSOR_Axes_t	{...}
x	int32_t	-582
y	int32_t	-343
z	int32_t	-369
+ Add new expression		

X-CUBE Mem Checkpoint 2



# Our technology starts with You



Find out more at [www.st.com](http://www.st.com)

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks).

All other product or service names are the property of their respective owners.



life.augmented