# STM32Cube.AI 기능소개 및 시연

- STM32Cube.AI / IDE Setup & Evaluation
- Introduction of STM32 IoT node(B-L475E-IOT01A)
- Using of STM32Cube.AI at the sensor node

# 발표자 소개

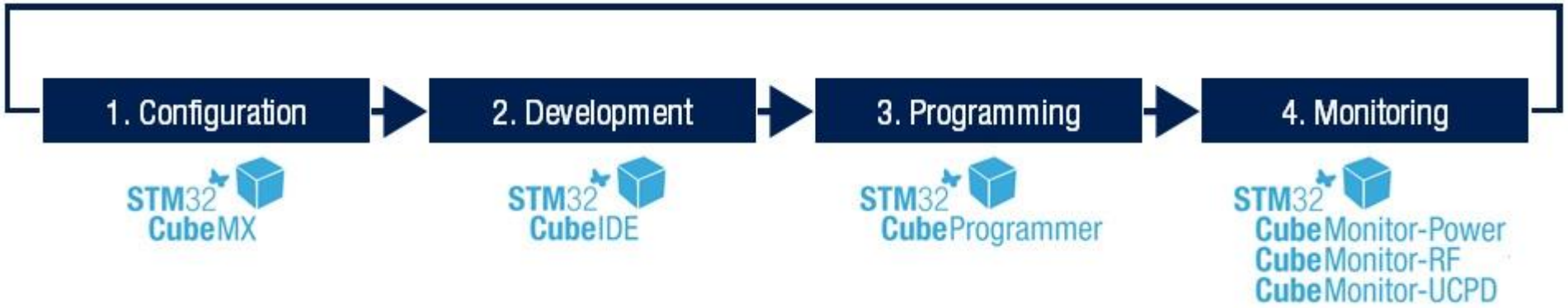


- 정재준 **(rgbi3307@nate.com)**
- 커널연구회(**www.kernel.bz**) 커뮤니티 대표
- 부천대학교 지능로봇과 겸임교수
- 연구 분야: 리눅스 커널, 자료구조 알고리즘, 머신러닝
- 집필 서적:
  - 리눅스 커널 자료구조 알고리즘 상세분석
  - 리눅스 커널 소스 해설 **[기초입문/RISC-V]**
  - **Device Tree** 상세분석, 리눅스 시스템 프로그래밍
  - 직접 코딩하면서 배우는 머신러닝 /딥러닝

# STM32Cube.AI / IDE



- STM32Cube ecosystem
- https://www.st.com/content/st_com/en/stm32cube-ecosystem.html




- https://www.st.com/en/development-tools/stm32cubeide.html

# STM32Cube.AI / IDE



- STM32CubeIDE is
  - C/C++ development platform
  - peripheral configuration, code generation, code compilation, and debug features
  - integrates all STM32CubeMX functionalities
  - Multi-OS support:
  - Windows®
  - Linux®
  - macOS®
  - 64-bit versions only

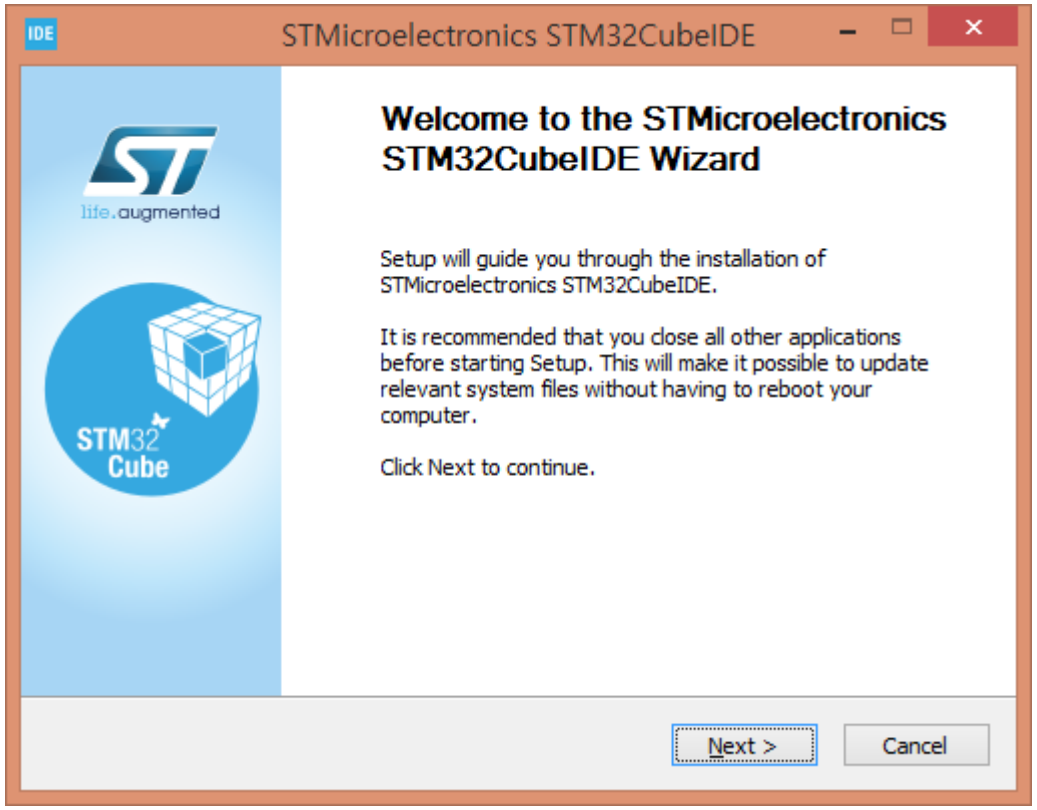
STM32CubeIDE

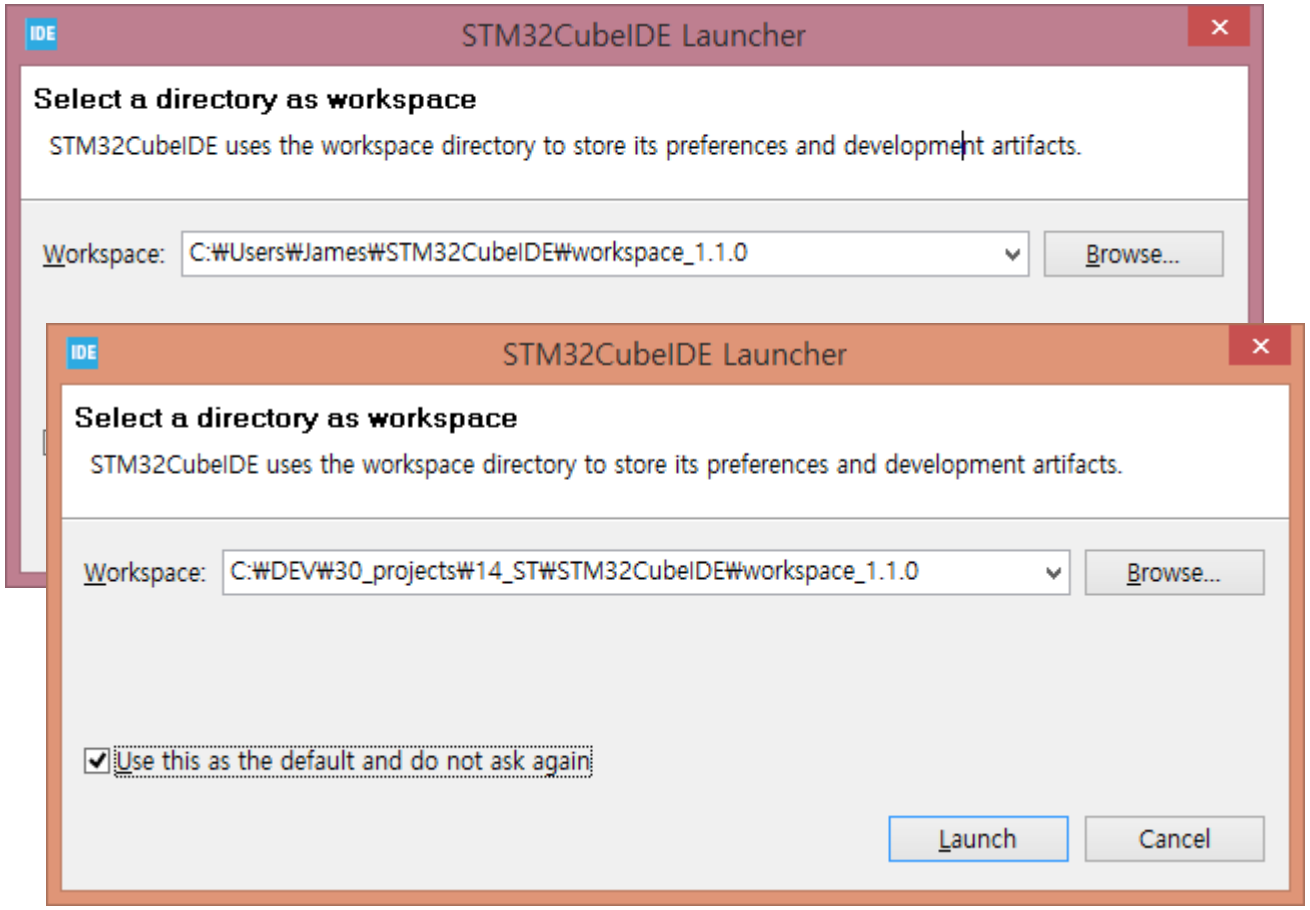
All-in-one STM32 development tool

# STM32Cube.AI / IDE



- Install at **C:\ST\STM32CubeIDE_1.1.0**

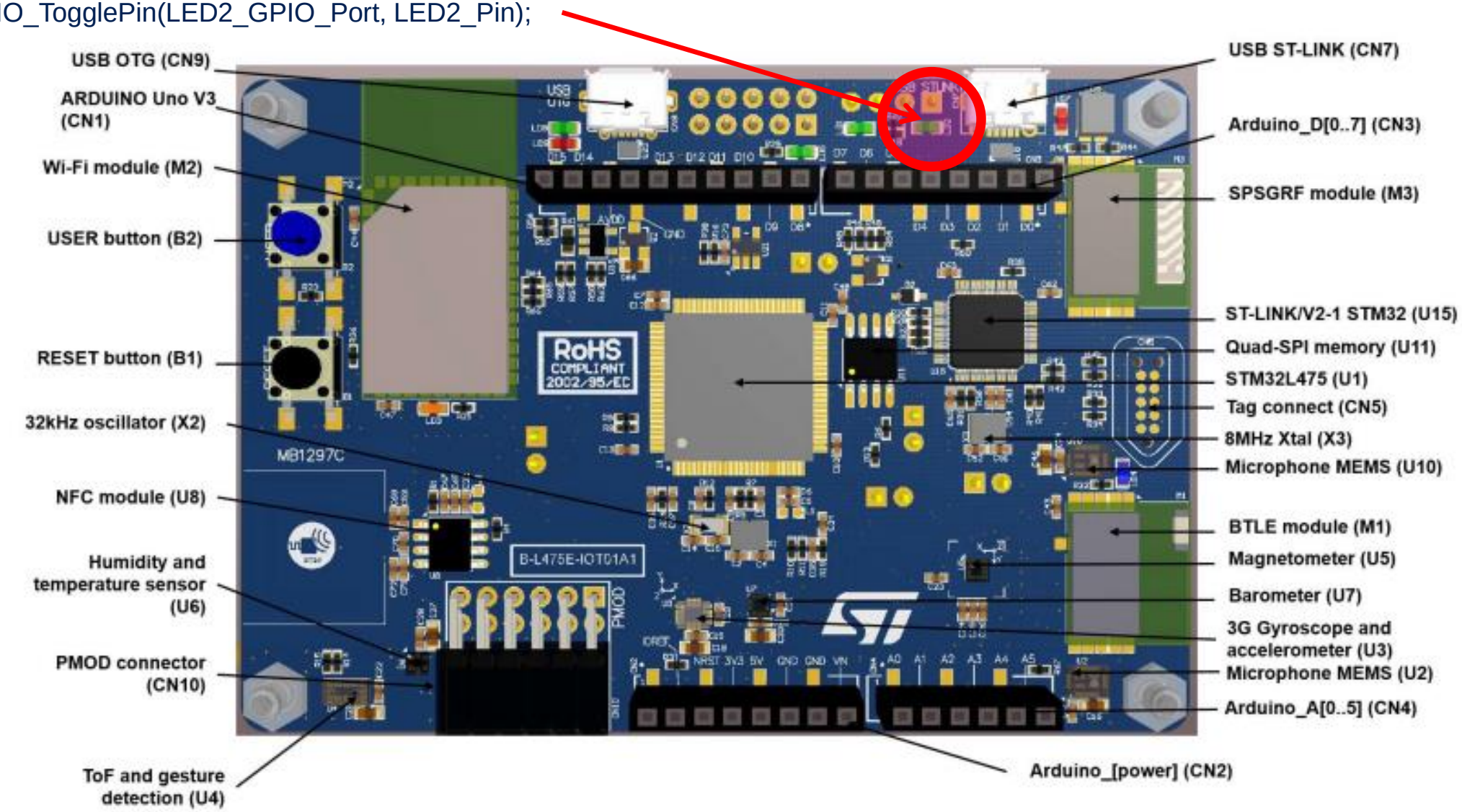


# STM32 IoT node(B-L475E-IOT01A)



```
while (1) {    HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
   HAL_Delay(1000);
}
```



# www.mouser.com (B-L475E-IOT01A)



MOUSER ELECTRONICS

All | Part # / Keyword

Products | Manufacturers | Services & Tools | Technical Resources | Help

All Products > Embedded Solutions > Engineering Tools > Embedded Processor Development Kits > Development Boards & Kits - ARM > STMicroelectronics B-L475E-IOT01A1

## B-L475E-IOT01A1

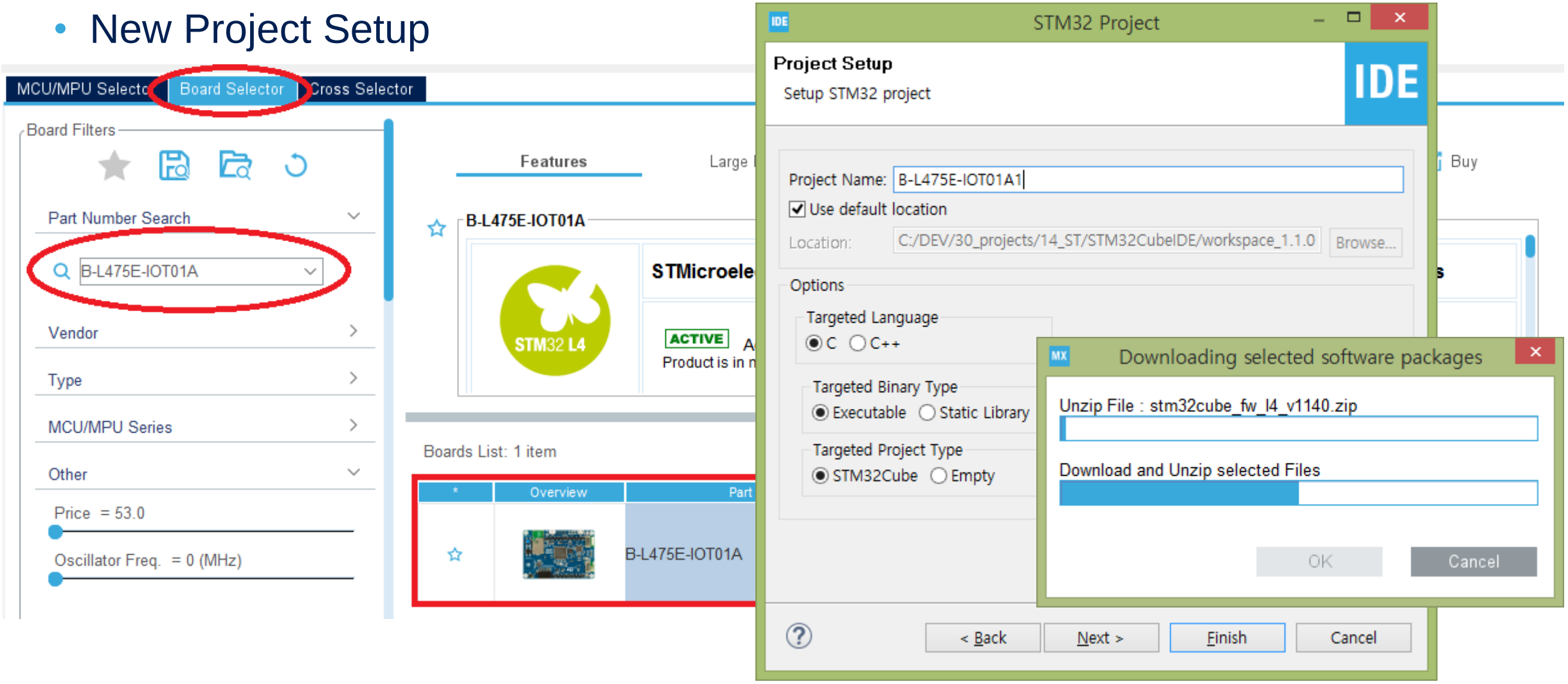
Enlarge

| | |
|---|---|
| **Mouser #:** | 511-B-L475E-IOT01A1 |
| **Mfr. #:** | B-L475E-IOT01A1 |
| **Mfr.:** | STMicroelectronics |
| **Customer #:** | Customer # |
| **Description:** | Development Boards & Kits - ARM 16/32-BITS MICROS |

# STM32Cube.AI / IDE



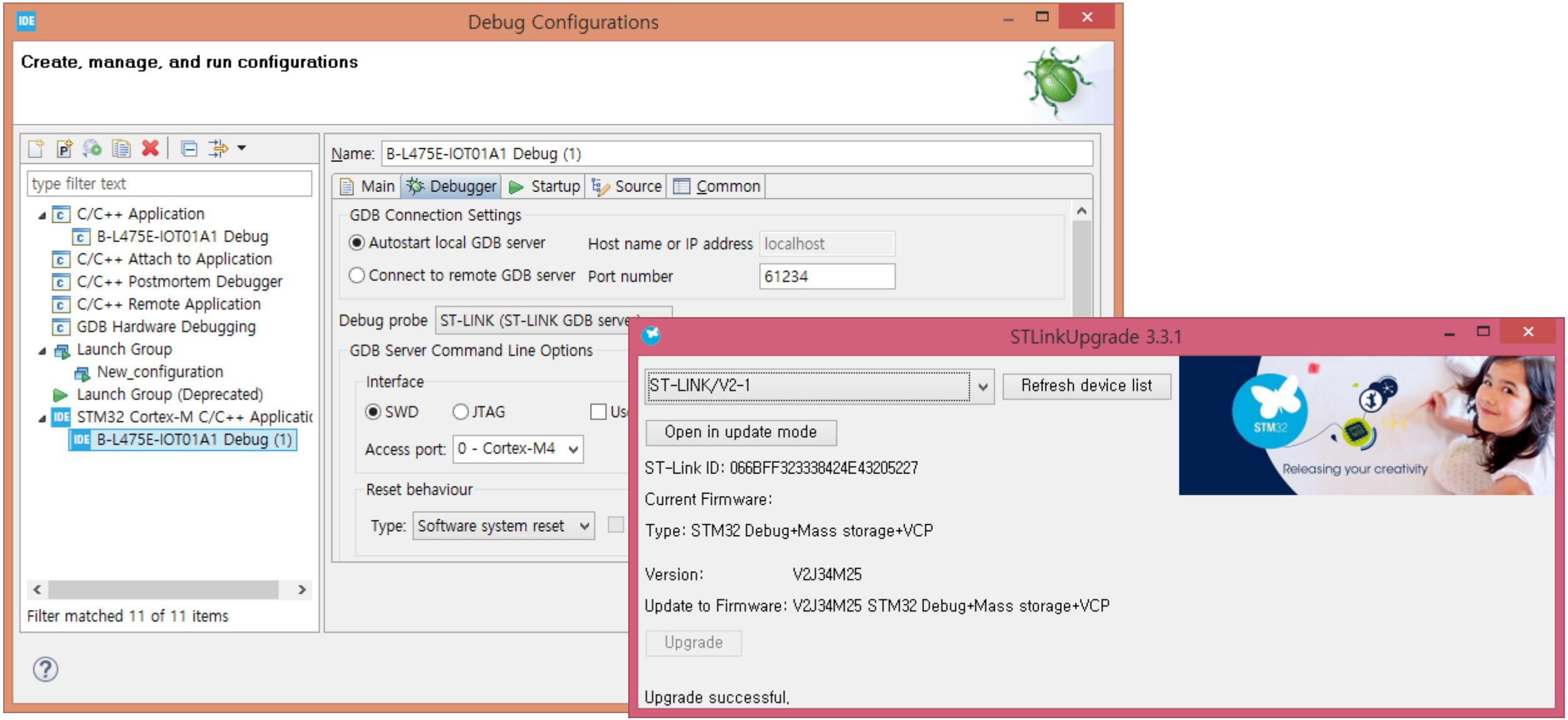
- New Project Setup



# STM32Cube.AI / IDE



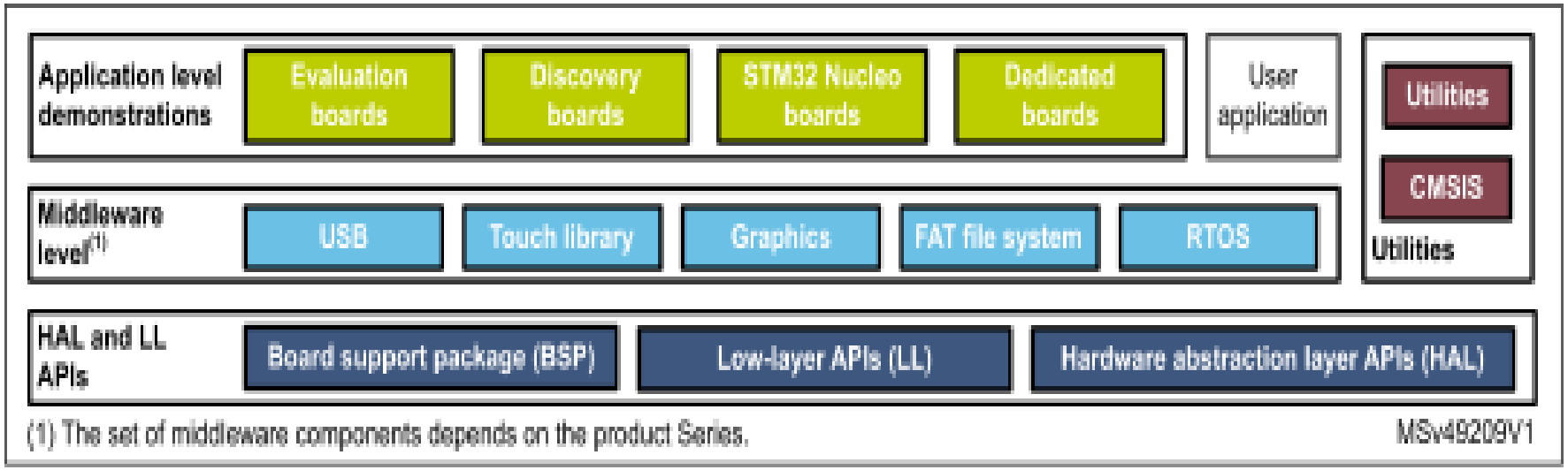
- Running & Debugging



# STM32Cube.AI / IDE



**Figure 1. STM32Cube MCU Package components**



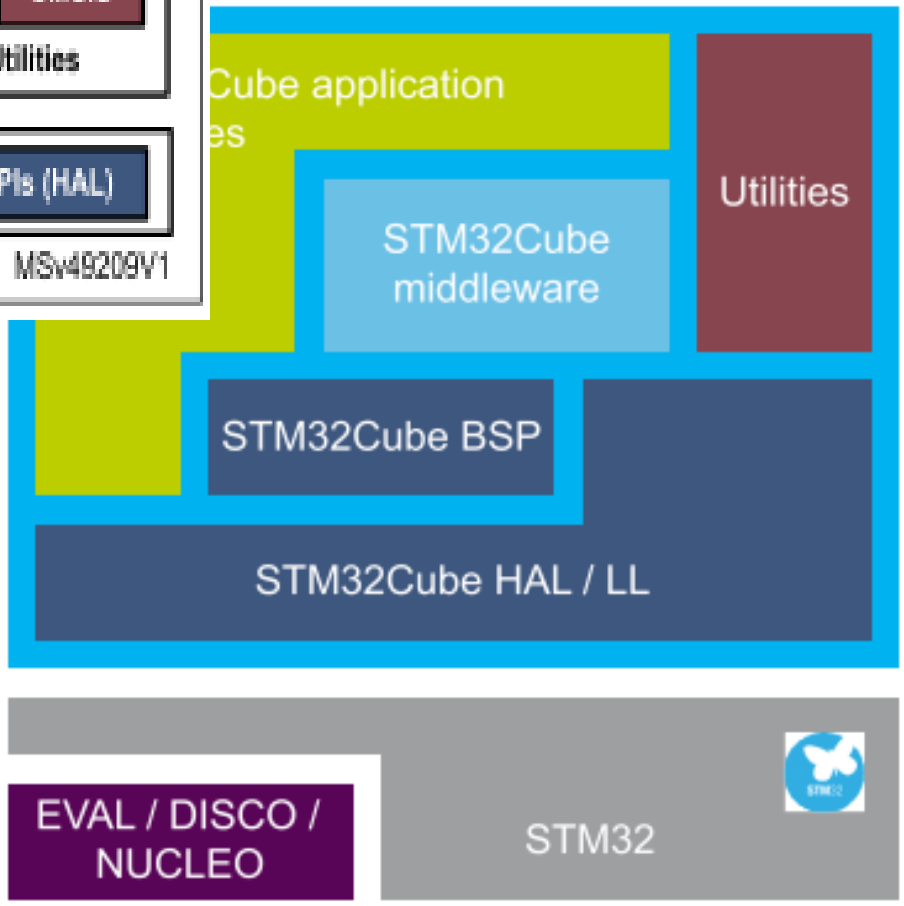
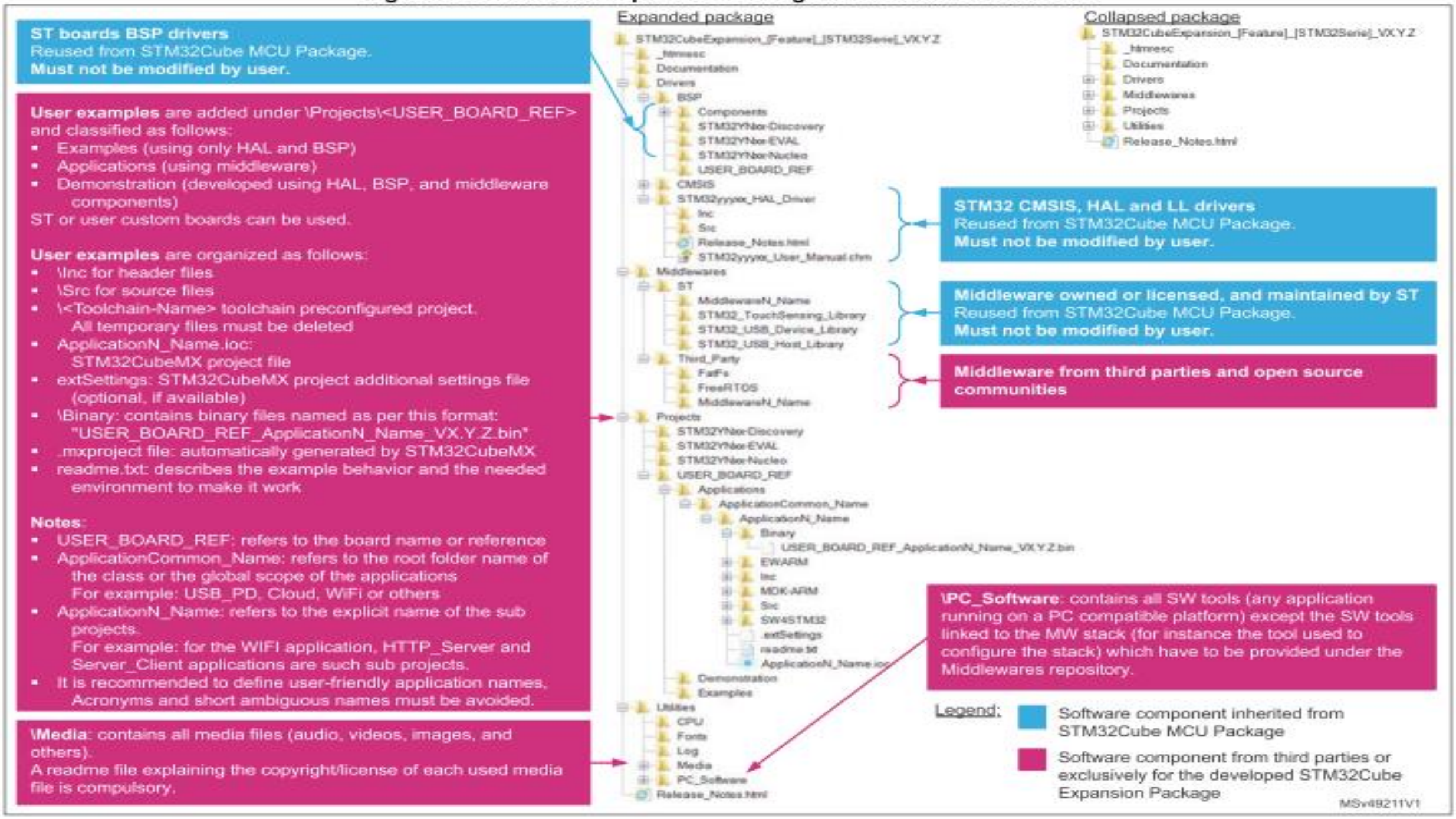
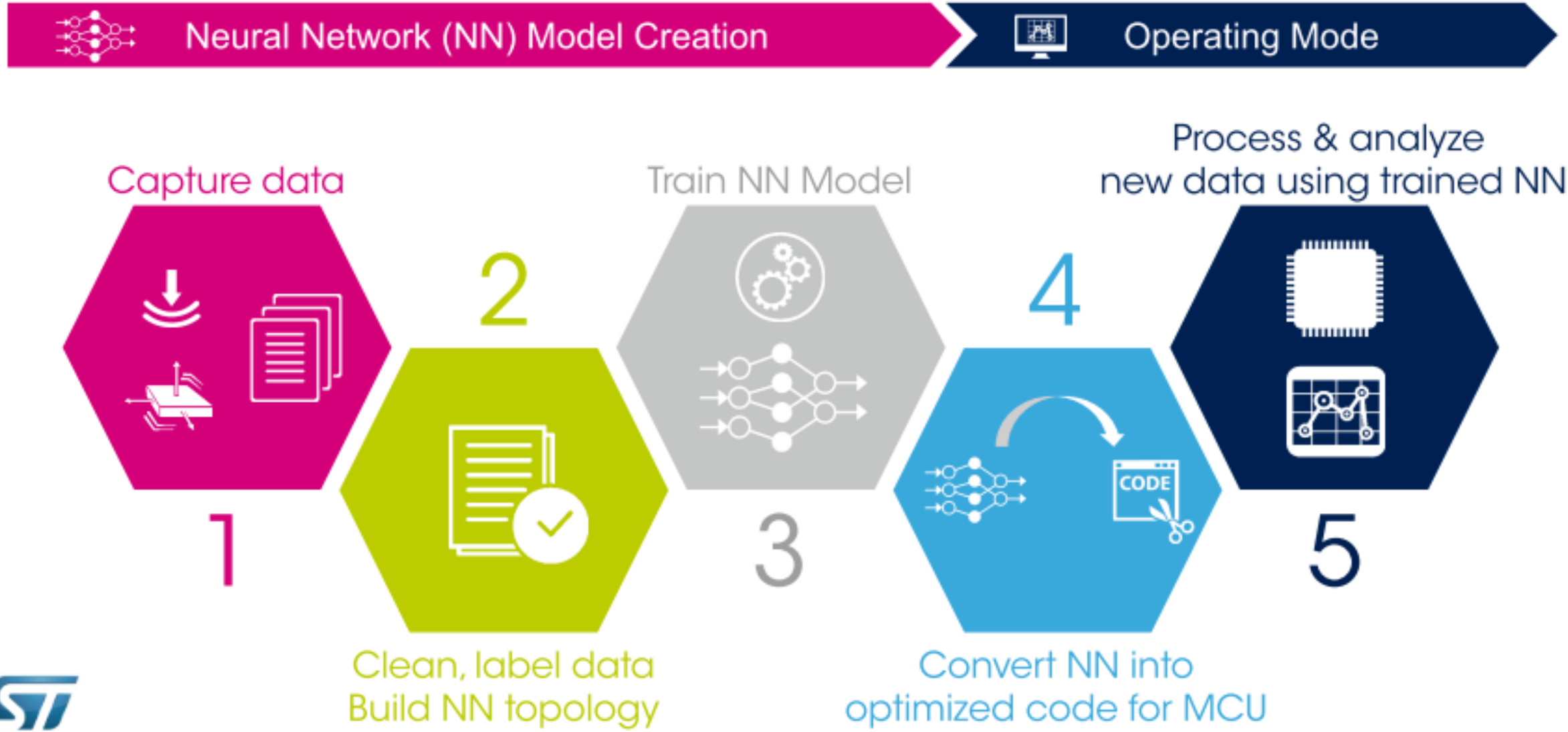



**Figure 3. STM32Cube Expansion Package architecture and content**



# Introduction of STM32Cube.AI






life.augmented

# Introduction of STM32Cube.AI

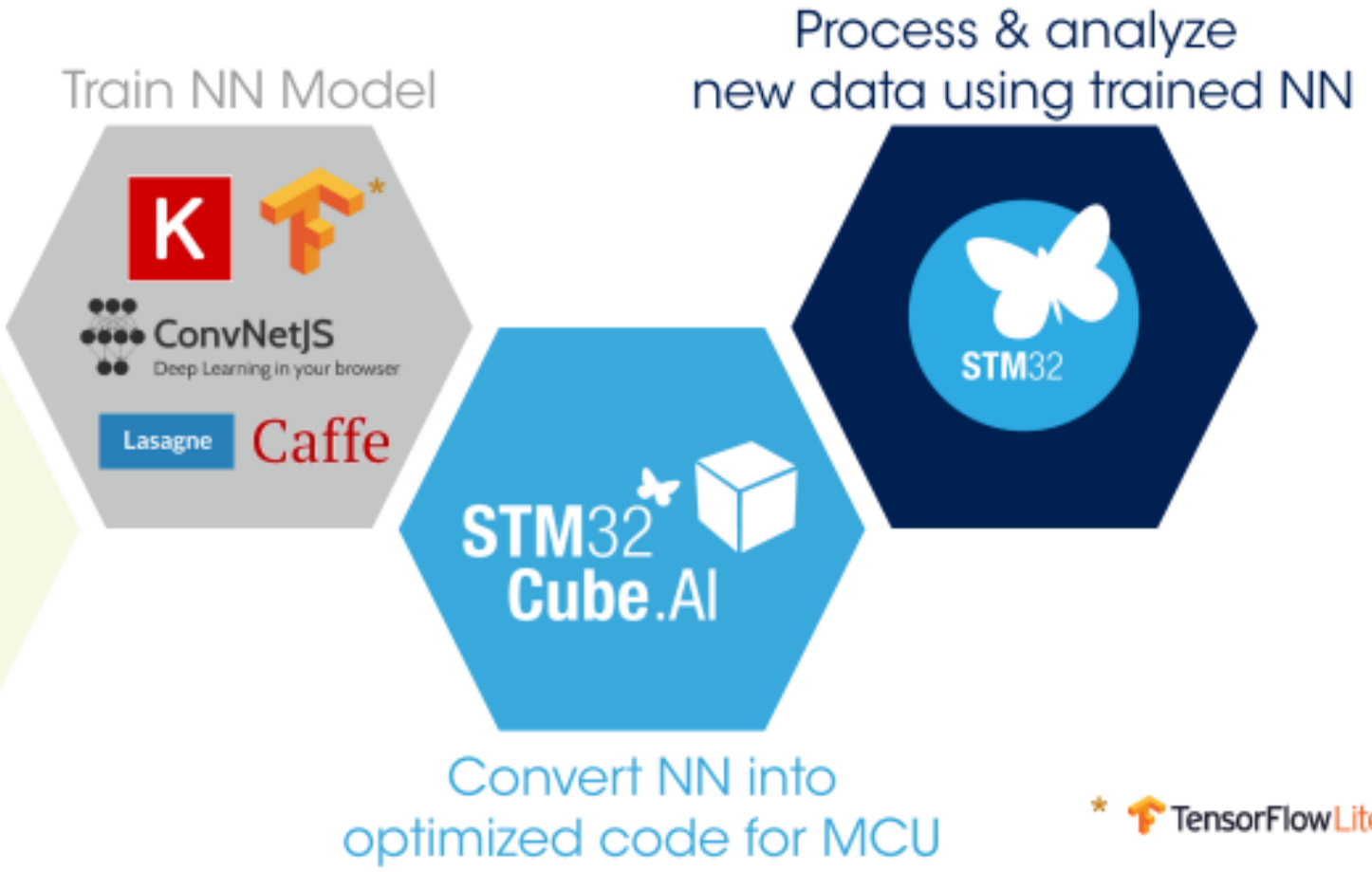


- AI Conversion Tool

Input your framework-dependent, pre-trained Neural Network into the **STM32Cube**.AI conversion tool

Automatic and fast generation of an STM32-optimized library

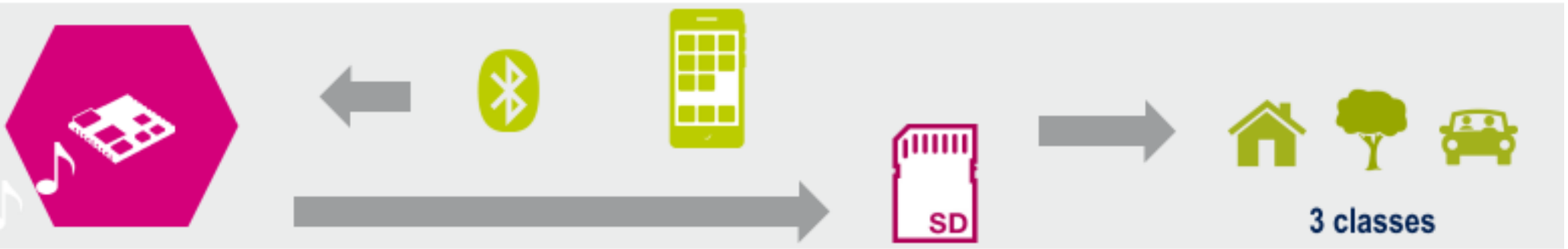
**STM32Cube**.AI offers interoperability with state-of-the-art Deep Learning design frameworks




life.augmented


# Audio Scene Classification (ASC)

## Audio Example in FP-AI-SENSING1 Package

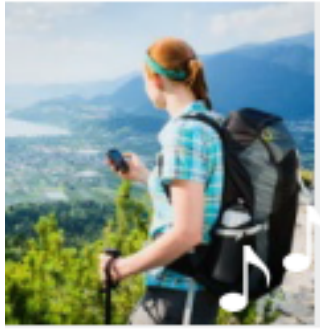


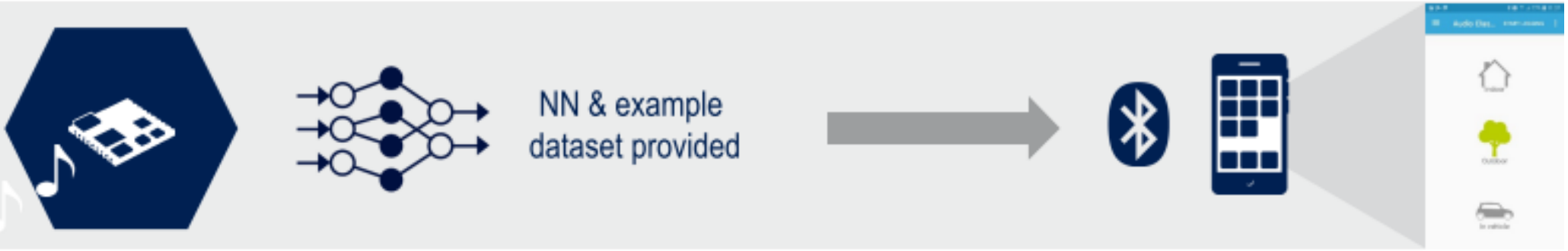

Audio **Data capture**

**Labelling** controlled by smartphone application

Data stored on the device SD card for future **learning**

Indoor, Outdoor, In vehicle **labelling**






Embedded **audio** pre-processing

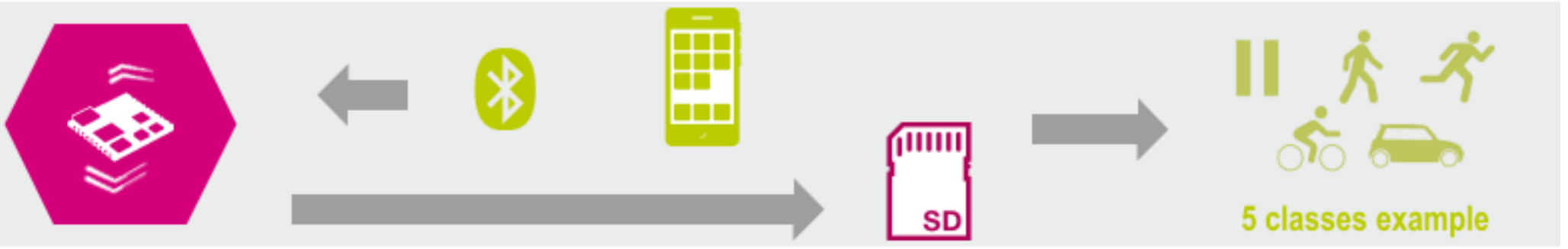
**Inferences** running on the microcontroller

**Inference result** displayed on mobile app

life.augmented


# Human Activity Recognition (HAR)

## Motion Example in FP-AI-SENSING1 Package

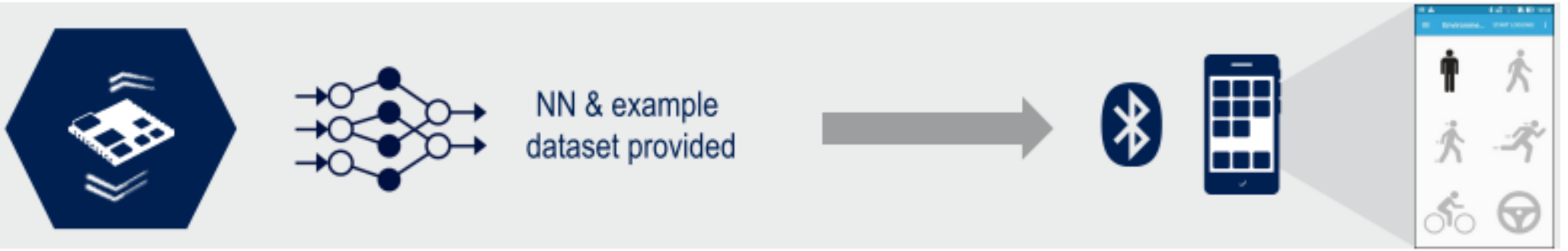



Motion **Data Capture**

**Labelling** controlled by smartphone application

Data stored on the device SD card for future **learning**

Stationary, walking, running, biking, driving **labelling**




Embedded **motion** pre-processing

**Inferences** running on the microcontroller

**Inference result** displayed on mobile app

life.augmented


# Image Classification

## Vision Example in FP-AI-VISION1 Package



Enjoy the food classification demo

- Default demo based on 18 classes (224x224 RGB pictures)
- Several camera image output size possible

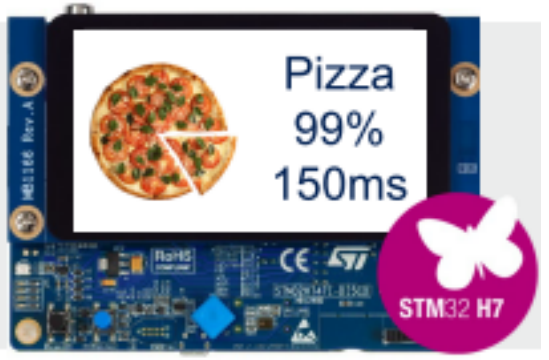
Full end-to-end optimized software example

- from camera acquisition to image pre-processing before feeding the NN
- Multiple memory mapping possibilities to optimise and test impact on performances
- Retrain this NN with your own dataset
- Quantize your trained network to optimized inference time and memory usage


Embedded **image** pre-processing (SW) on the STM32H747

NN & example dataset provided

**Inferences** running on the microcontroller



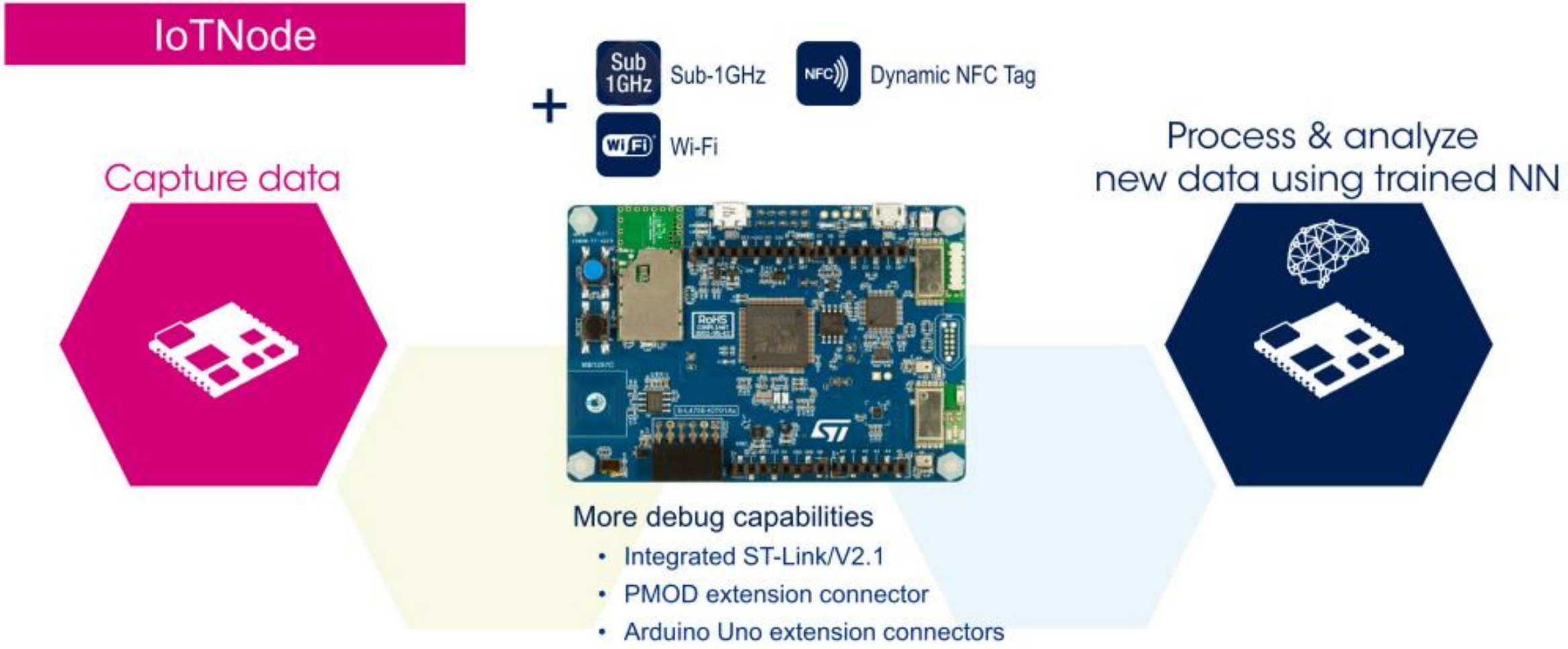

**Inference result** displayed on STM32H747 Discovery board LCD display

life.augmented

# Using of STM32Cube.AI



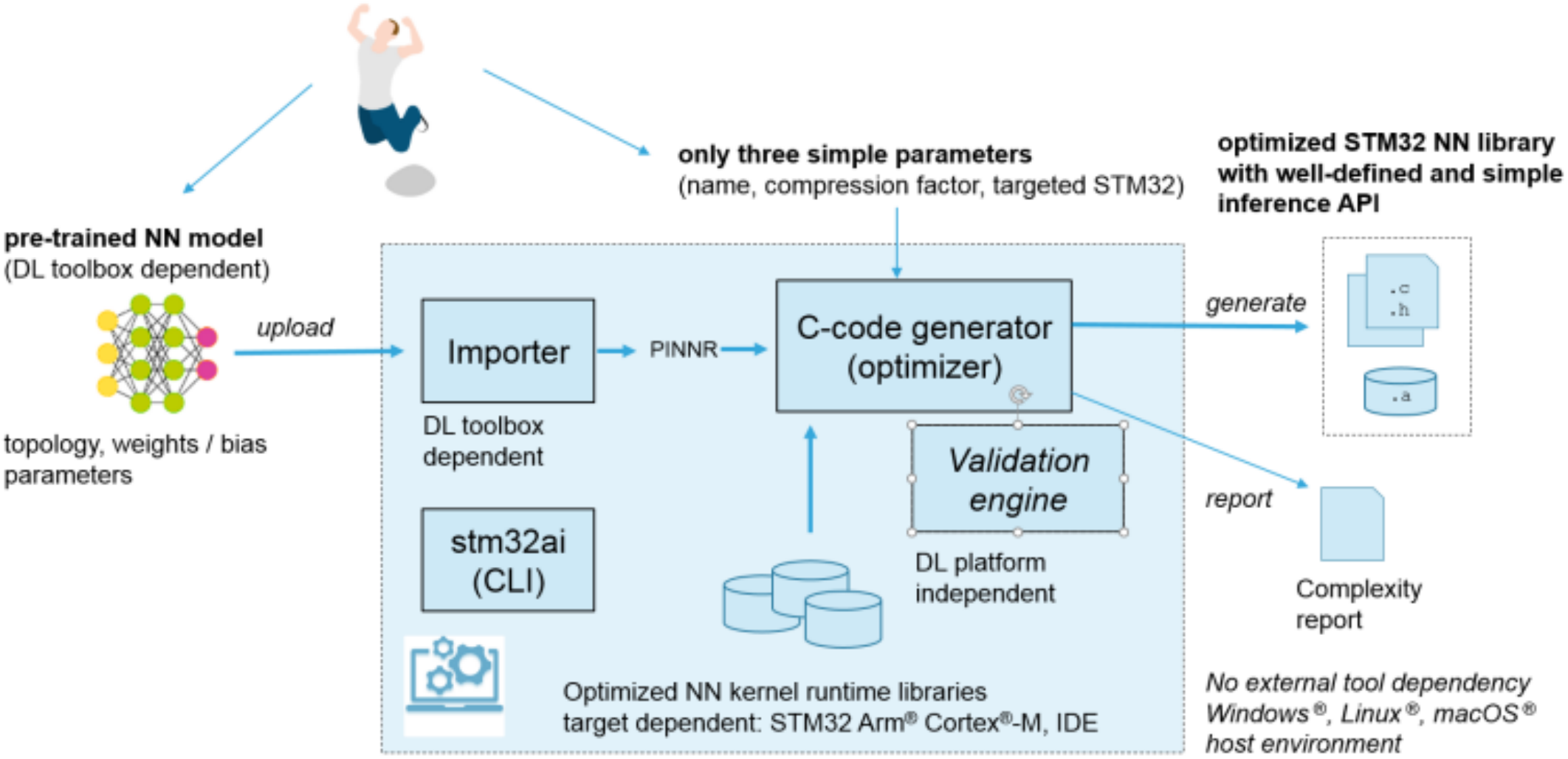
- https://www.st.com/IoTnode (B-L475E-IOT01A1)



# Using of STM32Cube.AI



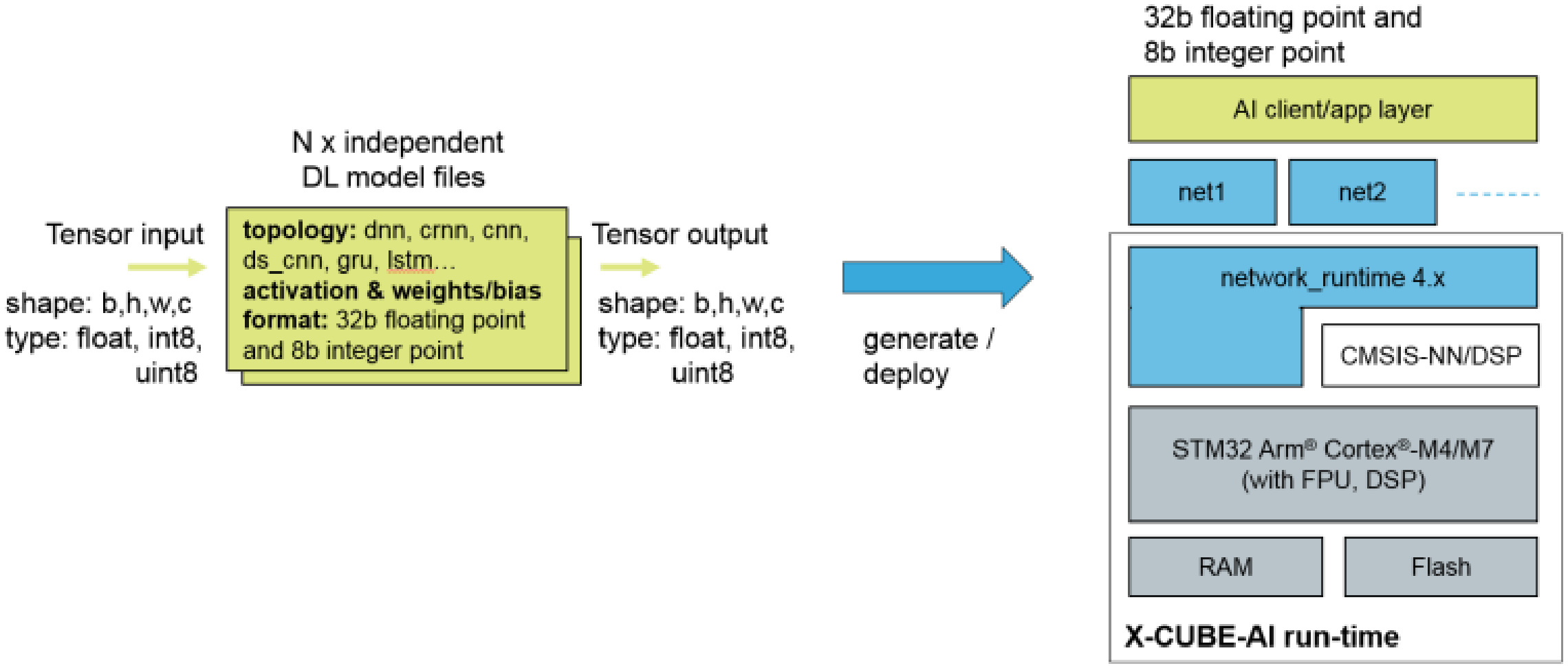
- X-CUBE-AI core engine



# Using of STM32Cube.AI



- X-CUBE-AI overview



# Using of STM32Cube.AI



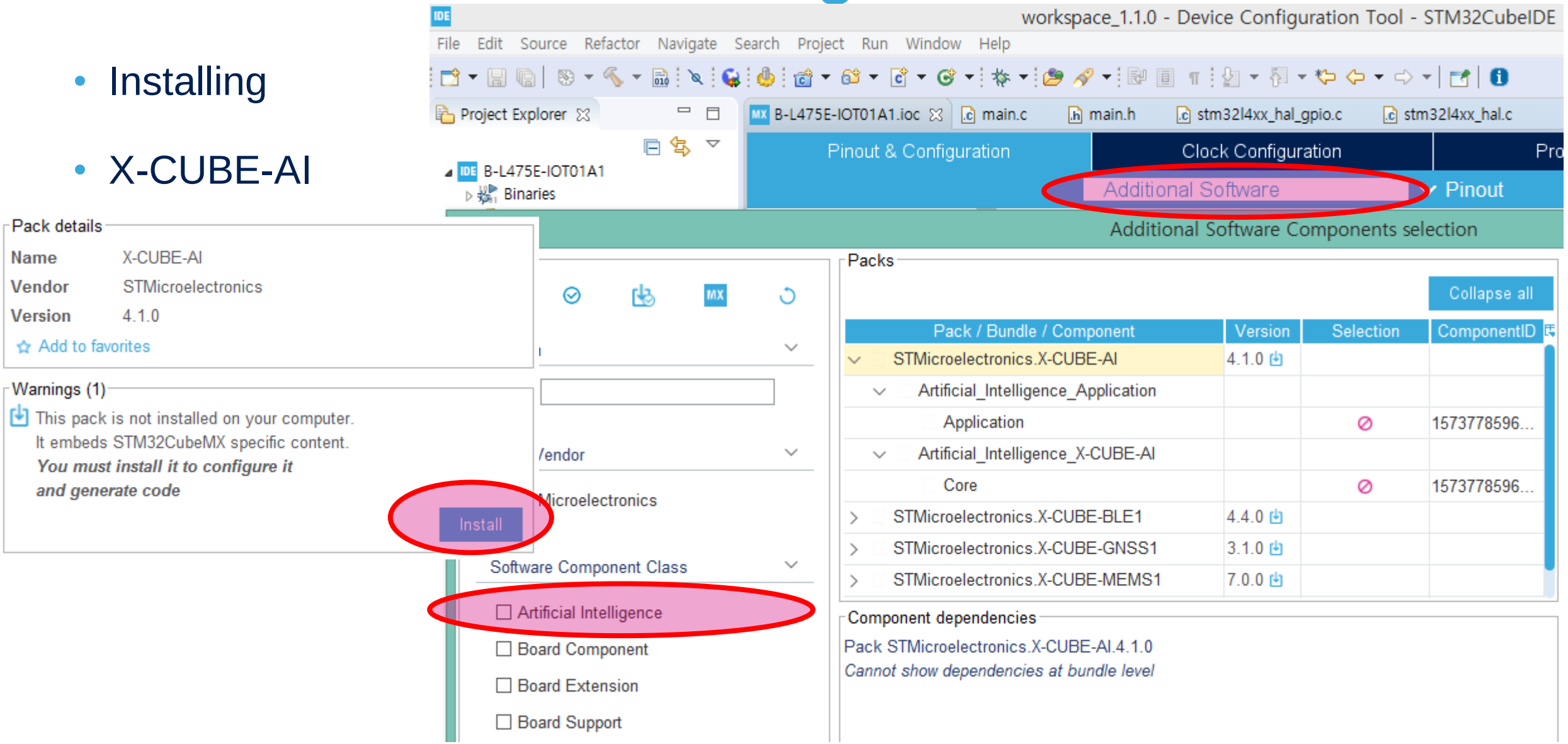
- Library source tree view

```
<project_name>
    |- Inc
    |      |- app_x-cube-ai.h /* entry points - MX_X_CUBE_AI_xx() fcts */
    |      |- bsp_ai.h        /* BSP AI adapt. for AI validation/systemperf application */
    |      |- constants_ai.h  /* BSP constant AI definition */
    |      |- <name_1>.h      /* specialized NN files */
    |      |- <name_1>_data.h
    |      |- <name_2>.h
    |      \- ...
    |- Src
    |      |- app_x-cube-ai.c
    |      |- <name_1>.c      /* specialized NN files */
    |      |- <name_1>_data.c
    |      \- ...
    | ...
    \--Middlewares
        \- ST/AI
             |-- include
             |    \- *.h                /* Internal/private AI headers */
             |-- lib
             |    \- network_runtime.a /* generic run-time library */
             \-- Application
                  \- SystemPerformance       /* generic sample application */
                       |- Inc
                       |   \- aiSystemPerformance.h
                       \- Src
                            \- aiSystemPerformance.c
```

# Using of STM32Cube.AI

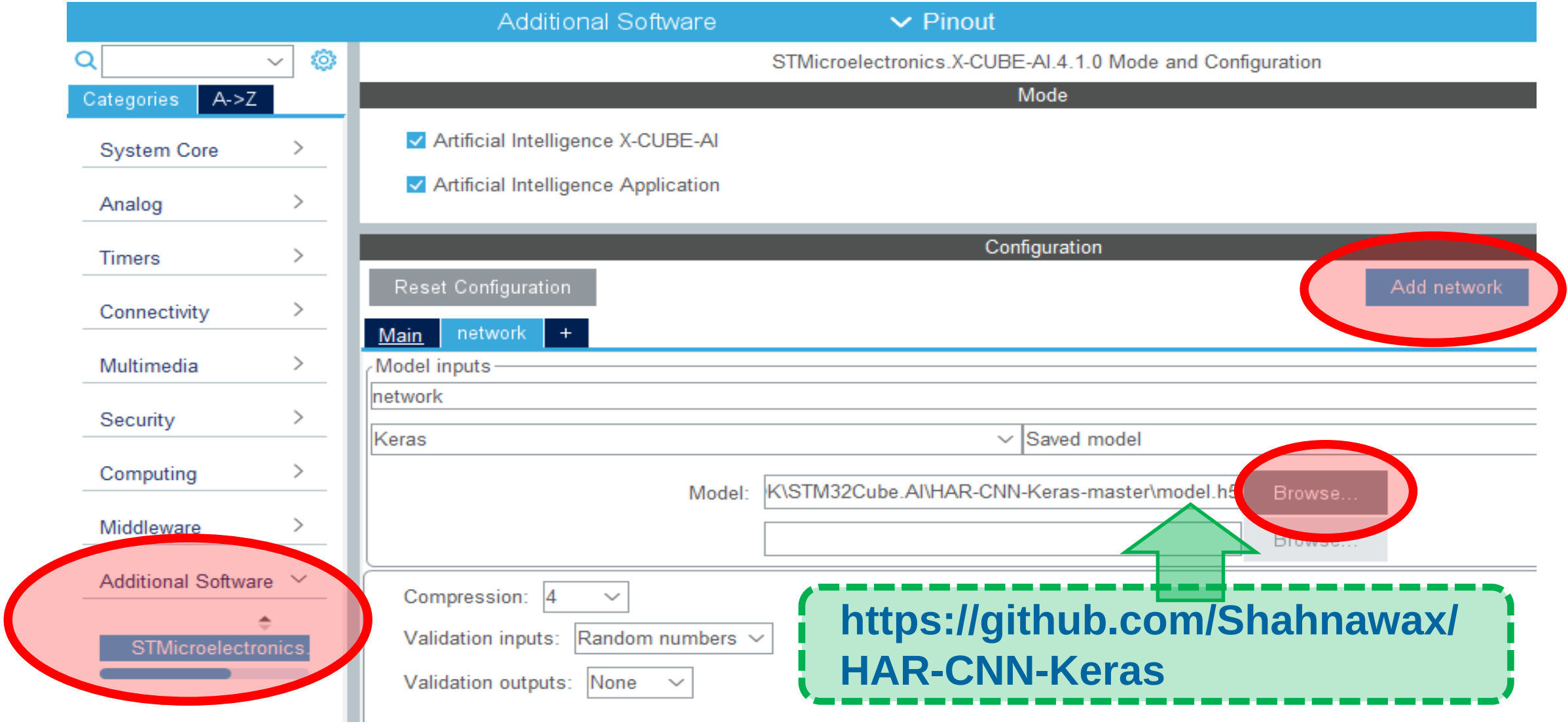


- Installing
- X-CUBE-AI



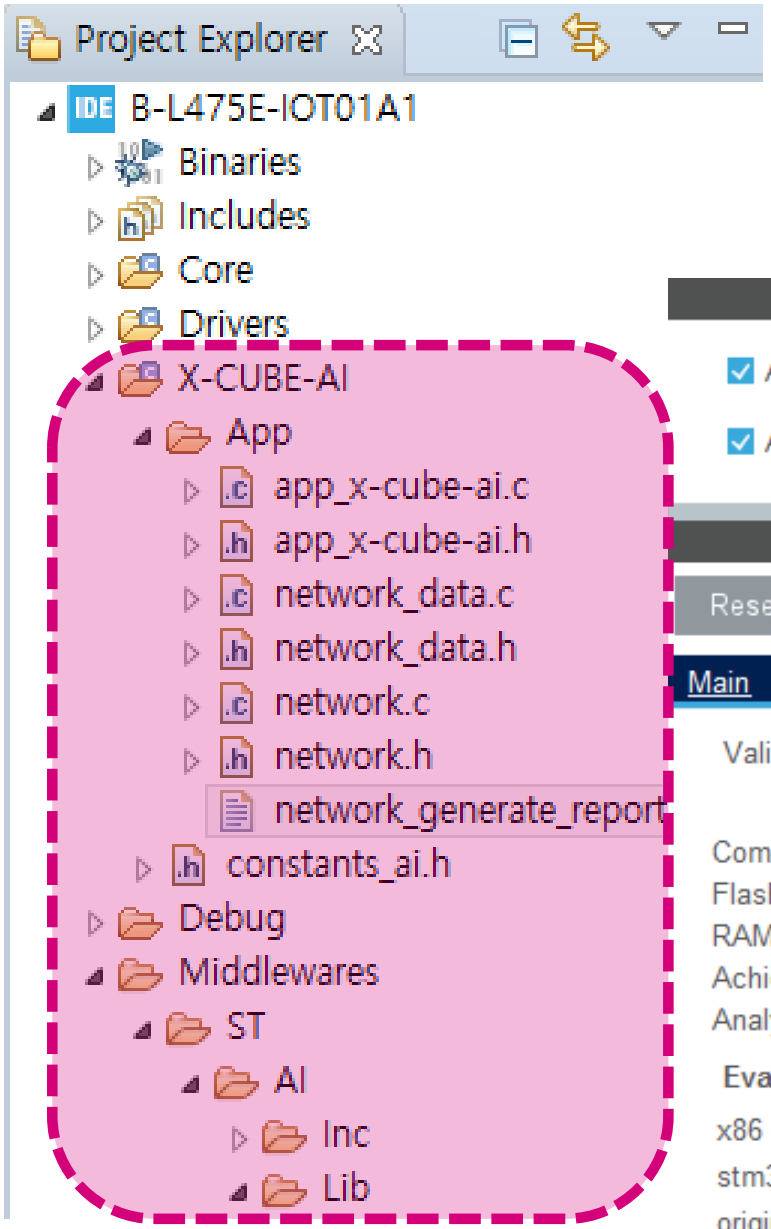
# Using of STM32Cube.AI

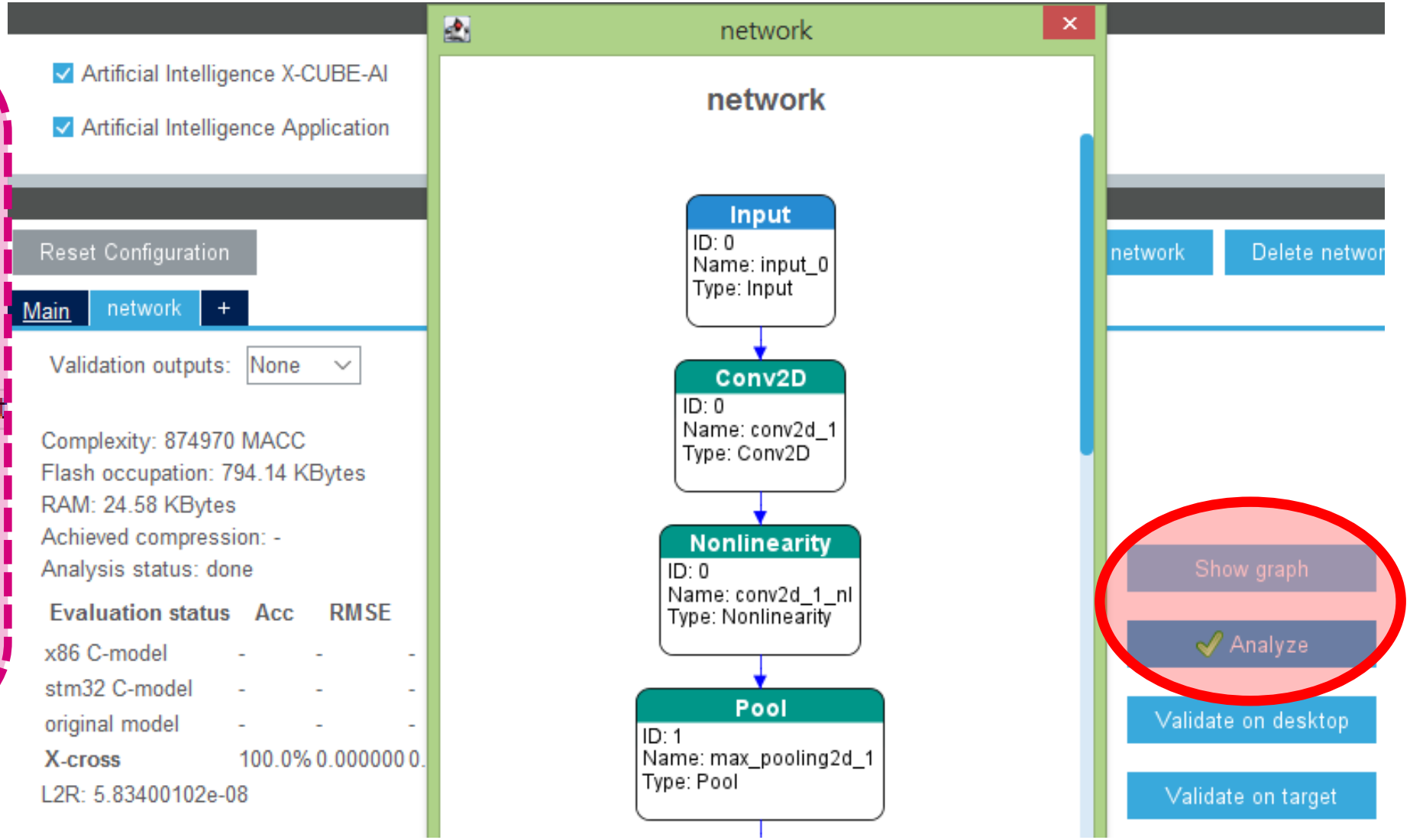




# Using of STM32Cube.AI





# Using of STM32Cube.AI



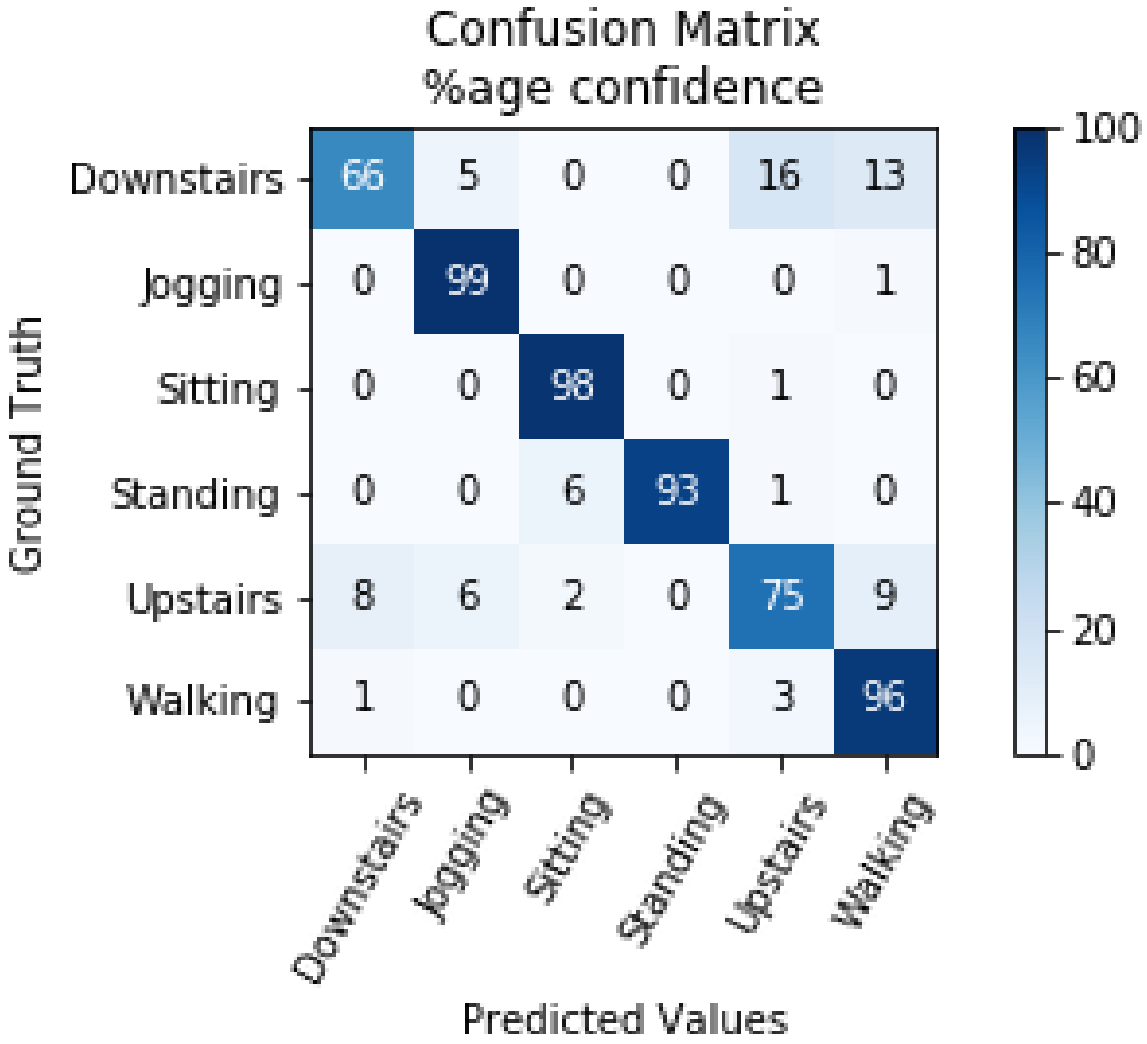
- Coding

```
//app_x-cube-ai.c
void MX_X_CUBE_AI_Init(void)
{
    AI_ALIGNED(4)
    static ai_u8 activations[AI_NETWORK_DATA_ACTIVATIONS_SIZE];
    aiInit(activations);
}

void MX_X_CUBE_AI_Process(void)
{
     /* USER CODE */
}

//main.c
int main(void)
{
    MX_X_CUBE_AI_Init();
    MX_X_CUBE_AI_Process();
    while (1)  {
        HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
        HAL_Delay(1000);
    }
}
```

# Using of STM32Cube.AI



- Output
- https://github.com/Shahnawax/HAR-CNN-Keras



# Using of STM32Cube.AI



- Running
- Q & A
- 감사합니다.

```
loop[8] answer:
-----------------------------------------------
  31,   97,   77,   50, -121,   60, : Jogging
 -26,   47,  -58,   -1,  127,   63, : Upstairs
 -29,   77,  -47,   49,   92,    1, : Upstairs
  10,   51,   37,   44,  100,   54, : Upstairs

loop[9] answer:
-----------------------------------------------
 119,  -59,  -84,   49,  -77,  -56, : Downstairs
  14,   47,  -22,   -1,  127,   63, : Upstairs
  27,   61,   15,   49,   29,  -57, : Jogging
  56,   50,  -90,  -18,  -78,   53, : Downstairs

loop[10] answer:
-----------------------------------------------
  51,   19,  -19,   48,   60,   -1, : Upstairs
-118,   46,  -16,   -1,  127,   63, : Upstairs
 -49,   88,  -69,   48,  -50,  116, : Walking
 -96,   49, -108,   53, -127,   53, : Standing


MX_X_CUBE_AI_Process() end.
```