

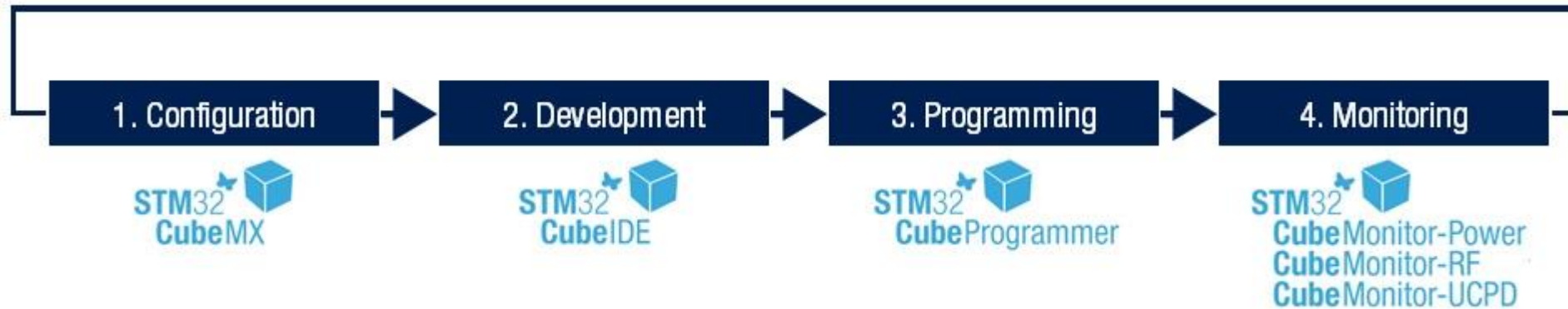
STM32Cube.AI 기능소개 및 시연

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

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



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



- 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**

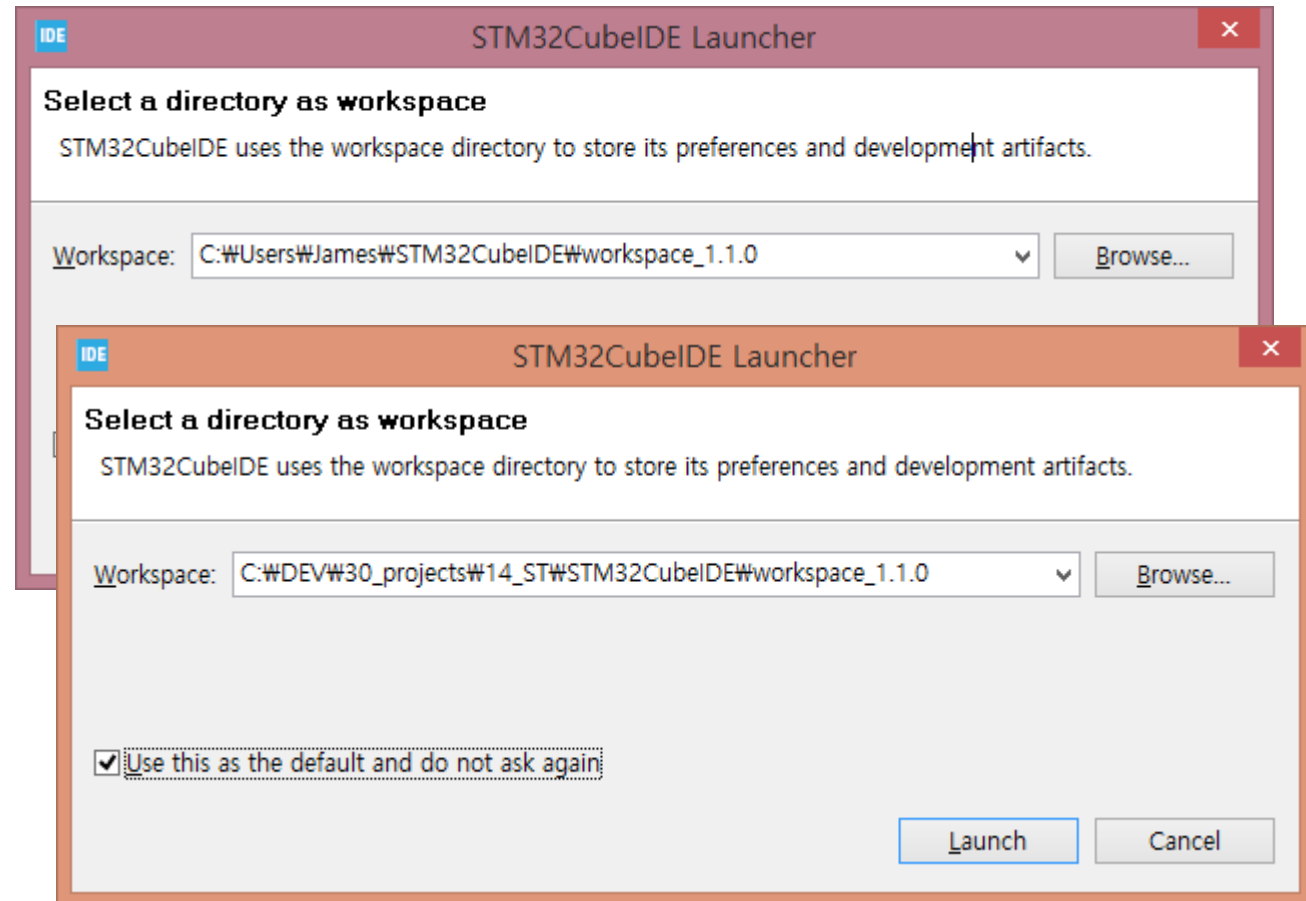
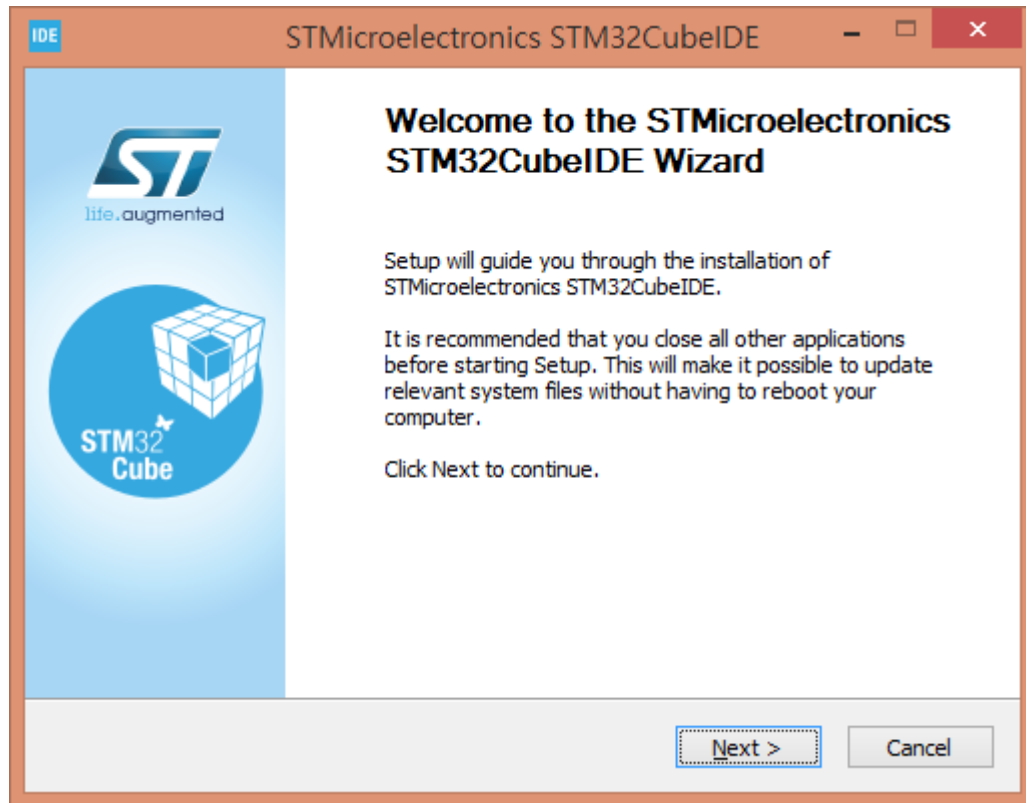
STM32CubeIDE

All-in-one STM32 development tool

TrueSTUDIO[®] for STM32

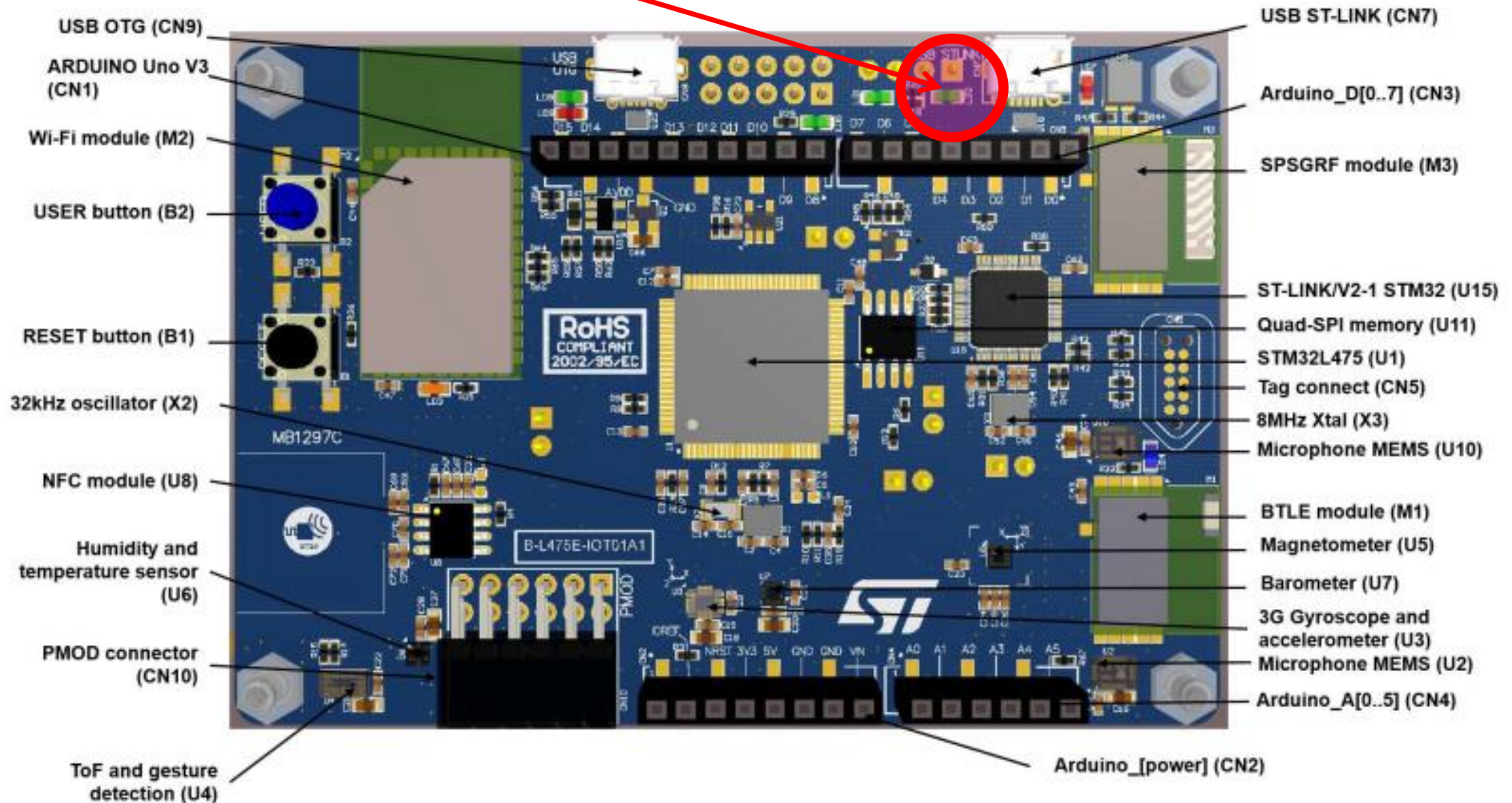


- Install at **C:\ST\STM32CubeIDE_1.1.0**



STM32 IoT node(B-L475E-IOT01A)

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





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

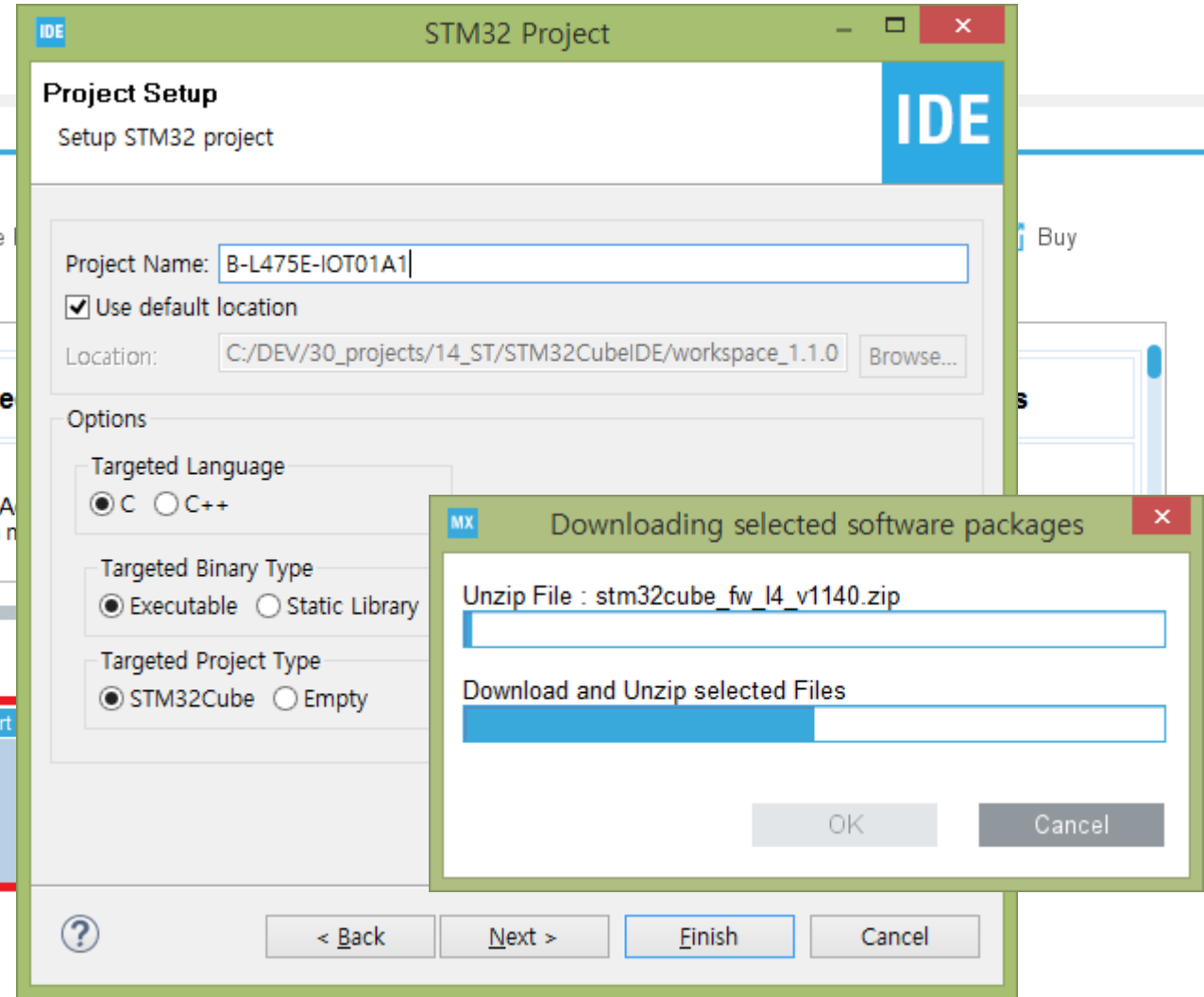
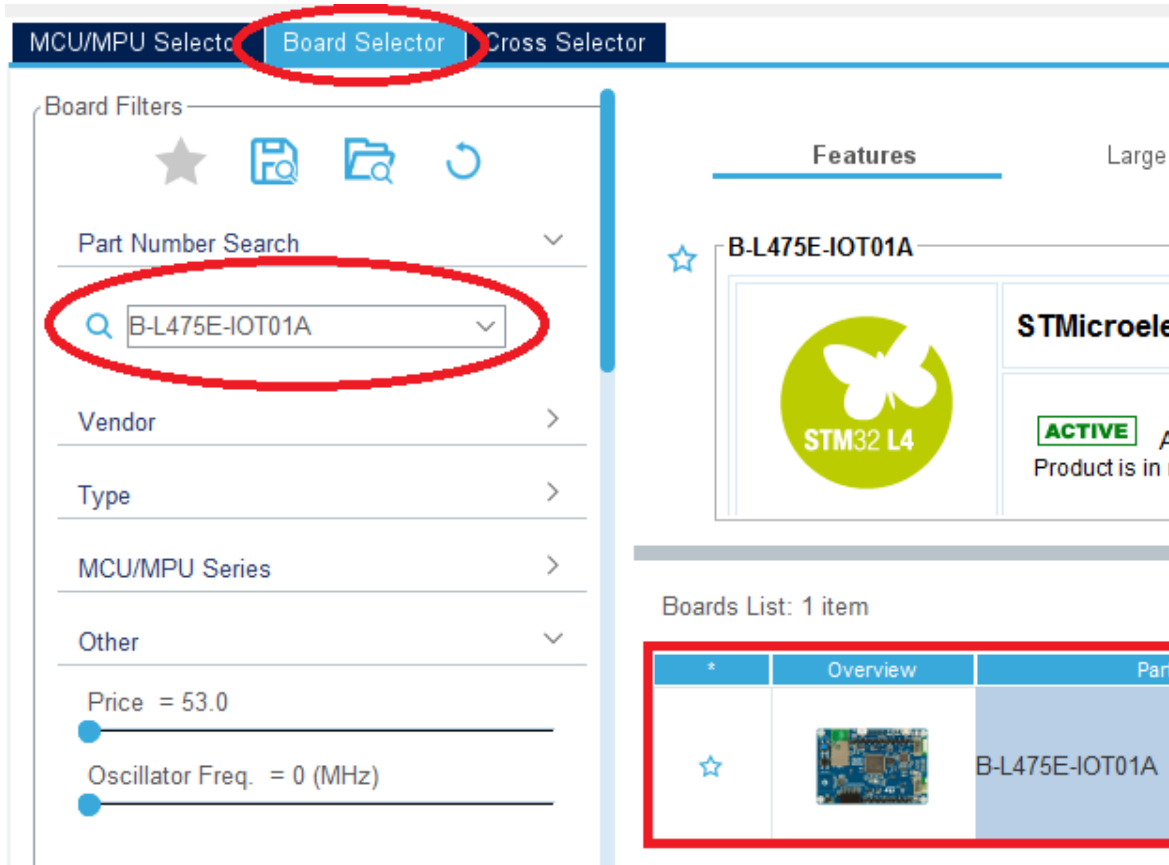
Mfr. #: B-L475E-IOT01A1

Mfr.: [STMicroelectronics](#)

Customer #:

Description: Development Boards & Kits - ARM 16/32-BITS MICROS

- New Project Setup



- Running & Debugging

IDE Debug Configurations

Create, manage, and run configurations

Name: B-L475E-IOT01A1 Debug (1)

Main Debugger Startup Source Common

GDB Connection Settings

Autostart local GDB server Host name or IP address localhost

Connect to remote GDB server Port number 61234

Debug probe ST-LINK (ST-LINK GDB server)

GDB Server Command Line Options

Interface

SWD JTAG Use JTAG

Access port: 0 - Cortex-M4

Reset behaviour

Type: Software system reset

STLinkUpgrade 3.3.1

ST-LINK/V2-1 Refresh device list

Open in update mode

ST-Link ID: 066BFF323338424E43205227

Current Firmware:

Type: STM32 Debug+Mass storage+VCP

Version: V2J34M25

Update to Firmware: V2J34M25 STM32 Debug+Mass storage+VCP

Upgrade

Upgrade successful.

Figure 1. STM32Cube MCU Package components

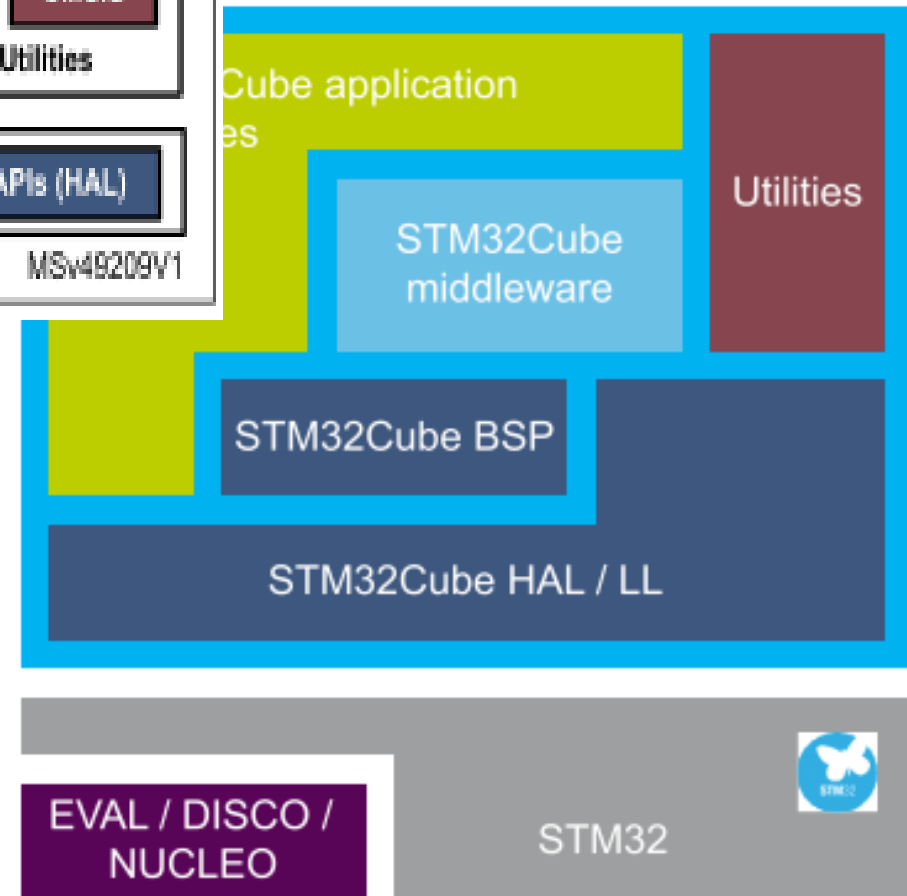
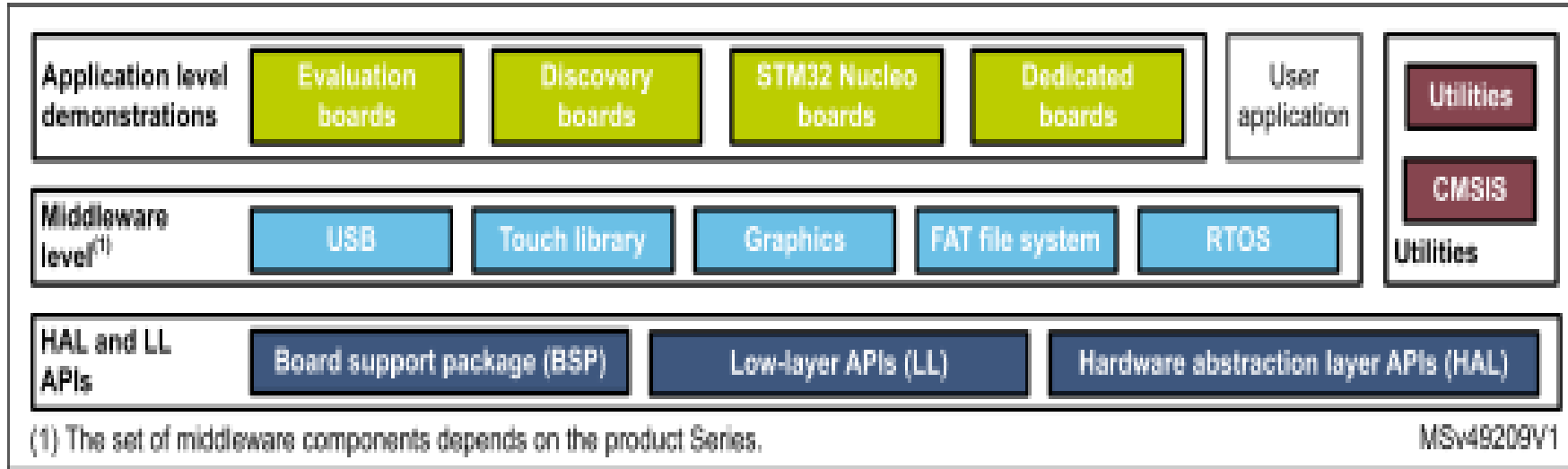


Figure 3. STM32Cube Expansion Package architecture and content

ST boards BSP drivers
 Reused from STM32Cube MCU Package.
 Must not be modified by user.

User examples are added under \Projects\<>USER_BOARD_REF<> and classified as follows:

- Examples (using only HAL and BSP)
- Applications (using middlewares)
- Demonstration (developed using HAL, BSP, and middleware components)

ST or user custom boards can be used.

User examples are organized as follows:

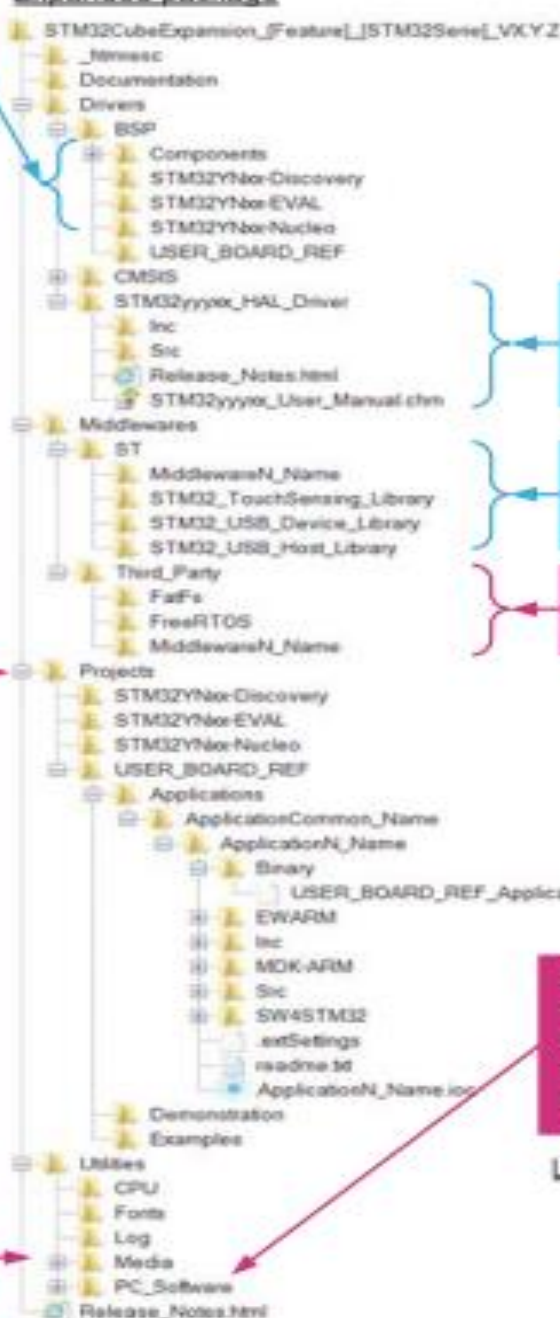
- \Inc for header files
- \Src for source files
- \<Toolchain-Name> toolchain preconfigured project. All temporary files must be deleted
- ApplicationN_Name.loc: STM32CubeMX project file
- extSettings: STM32CubeMX project additional settings file (optional, if available)
- \Binary: contains binary files named as per this format: "USER_BOARD_REF_ApplicationN_Name_VX.Y.Z.bin"
- .mxproject file: automatically generated by STM32CubeMX
- readme.txt: describes the example behavior and the needed environment to make it work

Notes:

- USER_BOARD_REF: refers to the board name or reference
- ApplicationCommon_Name: refers to the root folder name of the class or the global scope of the applications
 For example: USB_PD, Cloud, WiFi or others
- ApplicationN_Name: refers to the explicit name of the sub projects.
 For example: for the WIFI application, HTTP_Server and Server_Client applications are such sub projects.
- It is recommended to define user-friendly application names. Acronyms and short ambiguous names must be avoided.

Media: contains all media files (audio, videos, images, and others).
 A readme file explaining the copyright/license of each used media file is compulsory.

Expanded package



Collapsed package



STM32 CMSIS, HAL and LL drivers
 Reused from STM32Cube MCU Package.
 Must not be modified by user.

Middleware owned or licensed, and maintained by ST
 Reused from STM32Cube MCU Package.
 Must not be modified by user.

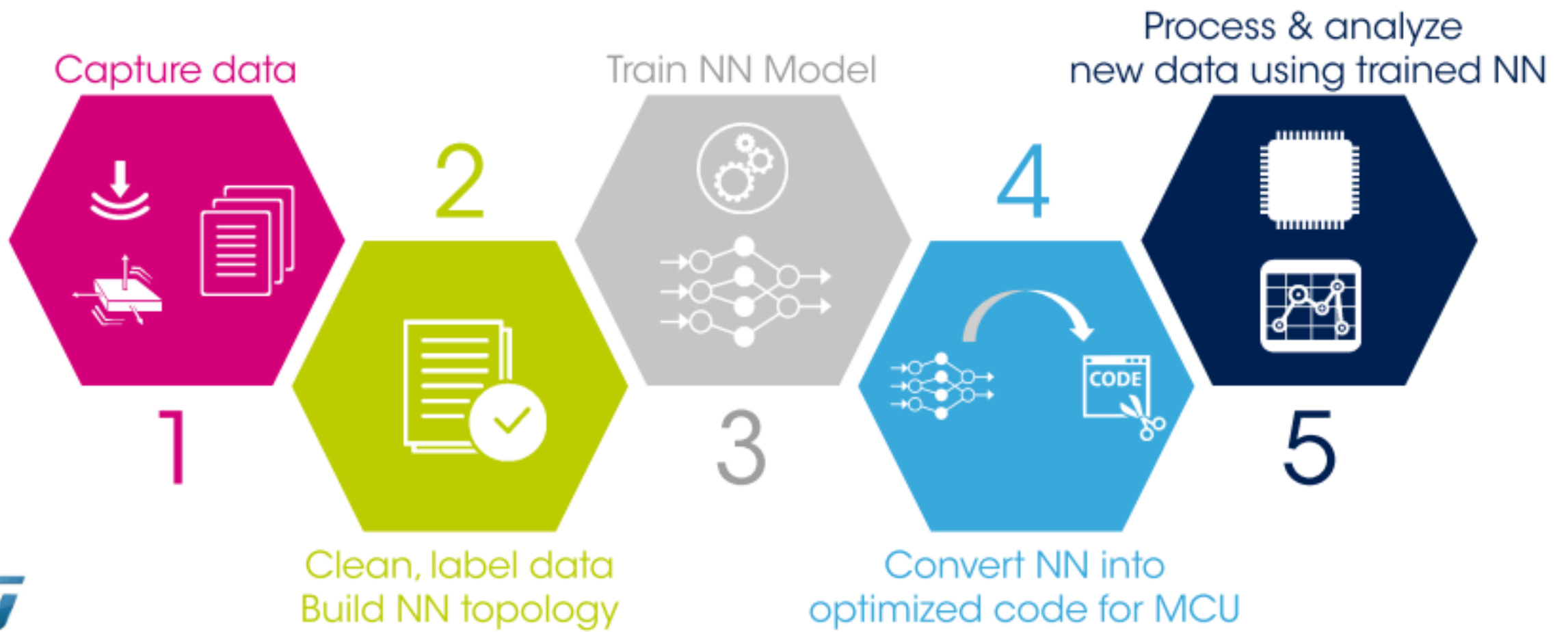
Middleware from third parties and open source communities

IPC_Software: contains all SW tools (any application running on a PC compatible platform) except the SW tools linked to the MW stack (for instance the tool used to configure the stack) which have to be provided under the Middlewares repository.

Legend:

- Software component inherited from STM32Cube MCU Package
- Software component from third parties or exclusively for the developed STM32Cube Expansion Package

Introduction of STM32Cube.AI



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

Train NN Model



Process & analyze new data using trained NN

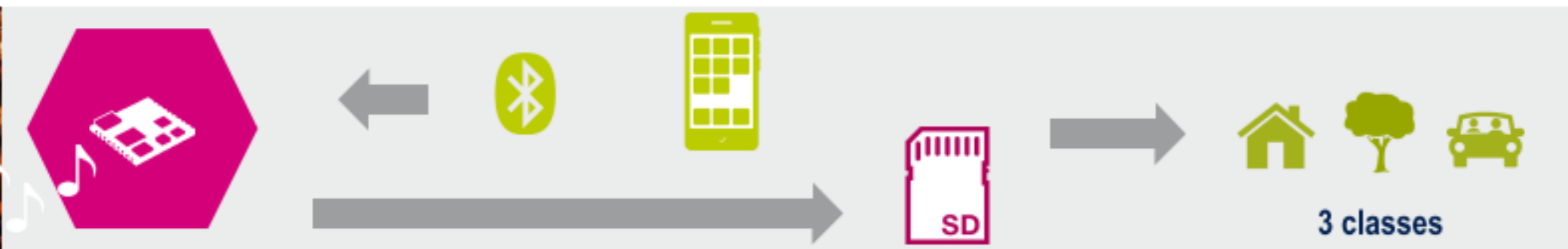


Convert NN into optimized code for MCU



Audio Scene Classification (ASC)

Audio Example in FP-AI-SENSING1 Package

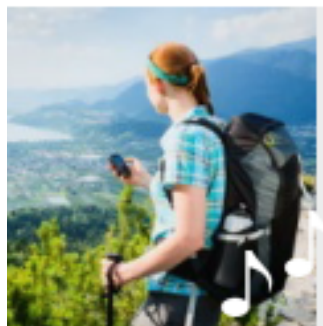


Audio Data capture

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



Human Activity Recognition (HAR)

Motion Example in FP-AI-SENSING1 Package

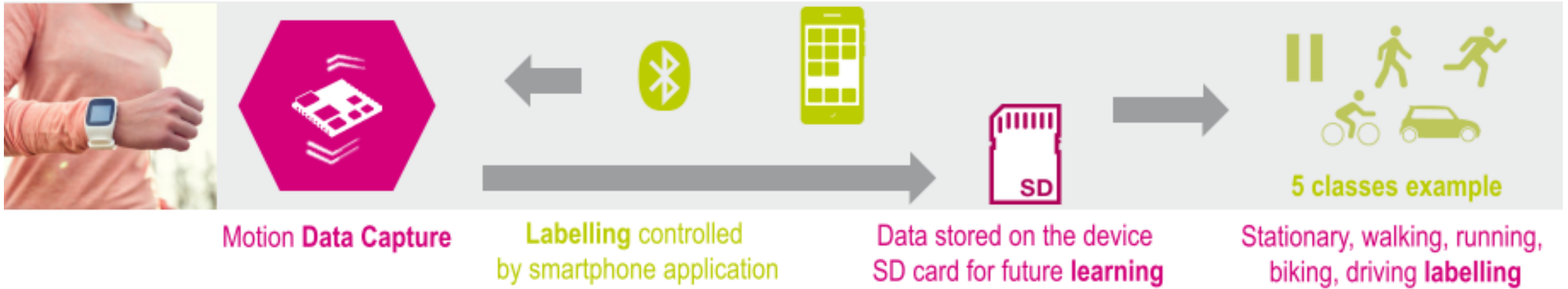




Image Classification

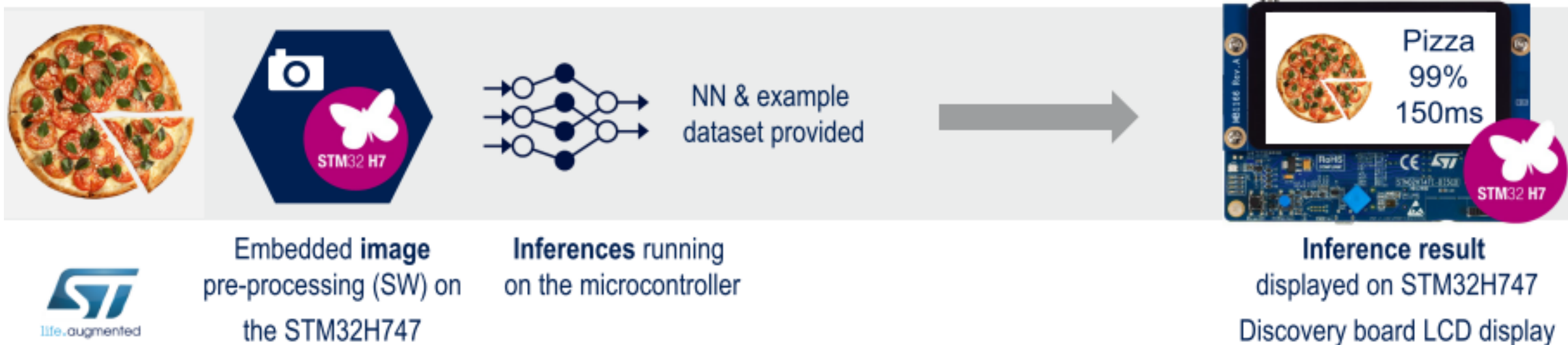
Vision Example in FP-AI-VISION1 Package

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



Using of STM32Cube.AI

- <https://www.st.com/loTnode> (B-L475E-IOT01A1)

IoTNode



Capture data



Process & analyze new data using trained NN

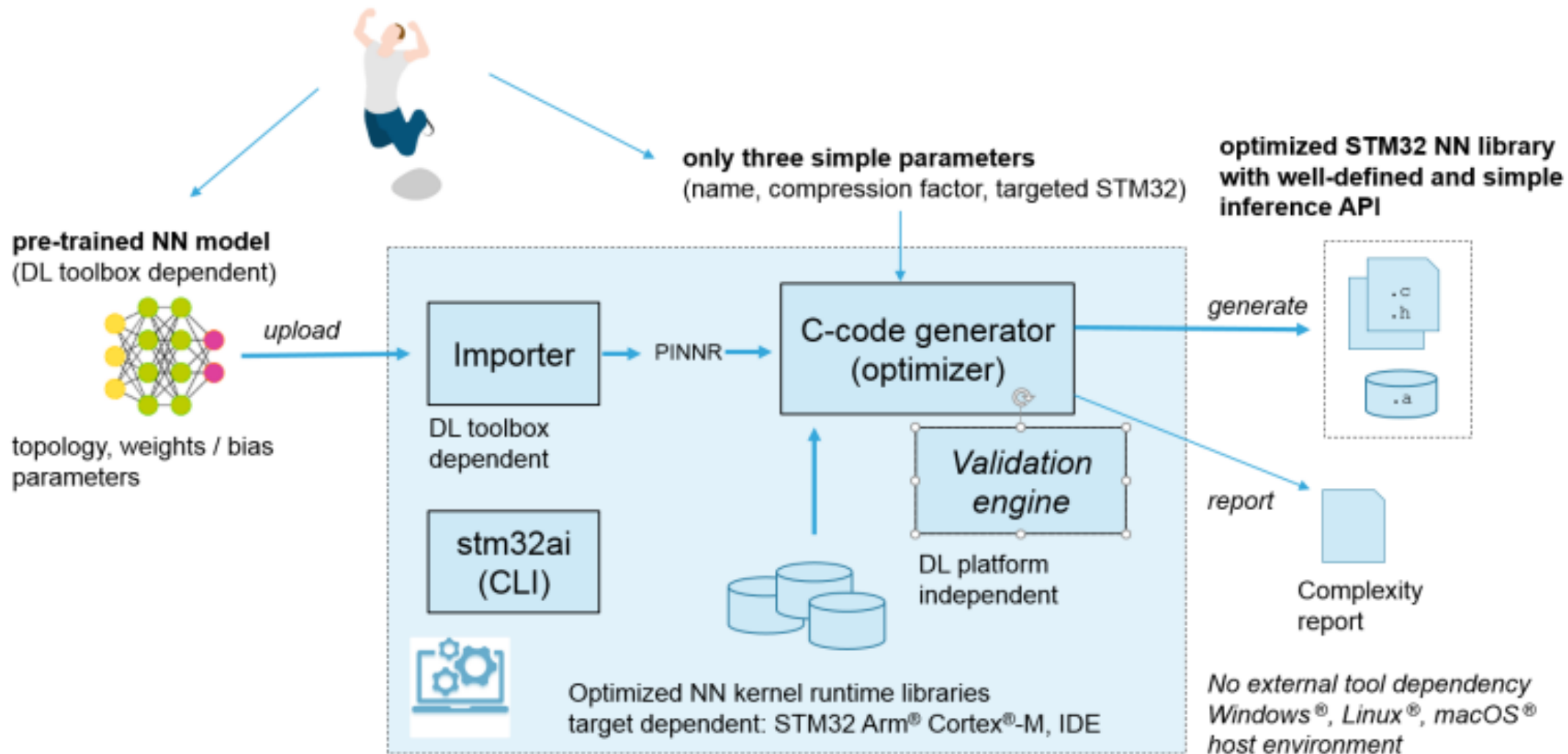


More debug capabilities

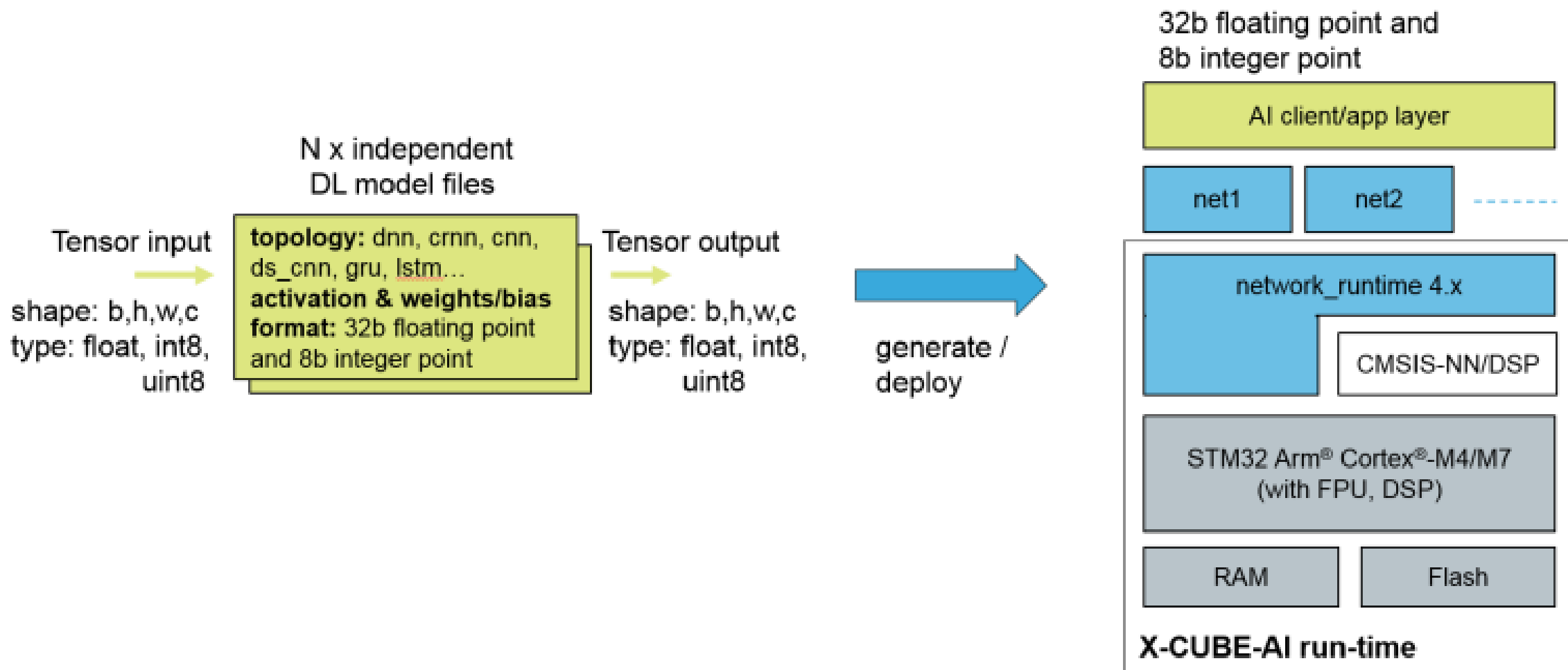
- Integrated ST-Link/V2.1
- PMOD extension connector
- Arduino Uno extension connectors

Using of STM32Cube.AI

- X-CUBE-AI core engine



- 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

21

- Installing
- X-CUBE-AI

workspace_1.1.0 - Device Configuration Tool - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer B-L475E-IOT01A1 B-L475E-IOT01A1.ioc main.c main.h stm32l4xx_hal_gpio.c stm32l4xx_hal.c

Pinout & Configuration Clock Configuration Additional Software Pinout

Additional Software Components selection

Pack details

Name X-CUBE-AI

Vendor STMicroelectronics

Version 4.1.0

Add to favorites

Warnings (1)

This pack is not installed on your computer. It embeds STM32CubeMX specific content. You must install it to configure it and generate code

Install

Artificial Intelligence

Board Component

Board Extension

Board Support

Packs

Pack / Bundle / Component	Version	Selection	ComponentID
STMicroelectronics.X-CUBE-AI	4.1.0		
Artificial_Intelligence_Application			
Application			1573778596...
Artificial_Intelligence_X-CUBE-AI			
Core			1573778596...
STMicroelectronics.X-CUBE-BLE1	4.4.0		
STMicroelectronics.X-CUBE-GNSS1	3.1.0		
STMicroelectronics.X-CUBE-MEMS1	7.0.0		

Component dependencies

Pack STMicroelectronics.X-CUBE-AI.4.1.0

Cannot show dependencies at bundle level

Using of STM32Cube.AI

The screenshot displays the STM32Cube.AI software interface. The top navigation bar includes 'Additional Software' and 'Pinout'. The main window title is 'STMicroelectronics.X-CUBE-AI.4.1.0 Mode and Configuration'. The interface is divided into 'Mode' and 'Configuration' sections. In the 'Mode' section, 'Artificial Intelligence X-CUBE-AI' and 'Artificial Intelligence Application' are checked. The 'Configuration' section includes a 'Reset Configuration' button, a 'Main network +' tab, and a 'Model inputs' field containing 'network'. Below this, a 'Keras' dropdown is set to 'Saved model', and the 'Model' field contains the path 'K:\STM32Cube.AI\HAR-CNN-Keras-master\model.h5'. A 'Browse...' button is circled in red. At the bottom, 'Compression' is set to 4, 'Validation inputs' is 'Random numbers', and 'Validation outputs' is 'None'. On the left sidebar, the 'Additional Software' menu is circled in red, showing 'STMicroelectronics' selected. A green dashed box at the bottom right contains the URL <https://github.com/Shahnawax/HAR-CNN-Keras>, with a green arrow pointing to the 'Browse...' button.

Using of STM32Cube.AI

Project Explorer

- IDE B-L475E-IOT01A1
 - Binaries
 - Includes
 - Core
 - Drivers
 - X-CUBE-AI
 - App
 - app_x-cube-ai.c
 - app_x-cube-ai.h
 - network_data.c
 - network_data.h
 - network.c
 - network.h
 - network_generate_report
 - constants_ai.h
 - Debug
 - Middlewares
 - ST
 - AI
 - Inc
 - Lib

network

Artificial Intelligence X-CUBE-AI

Artificial Intelligence Application

Reset Configuration

Main network +

Validation outputs: None

Complexity: 874970 MACC
Flash occupation: 794.14 KBytes
RAM: 24.58 KBytes
Achieved compression: -
Analysis status: done

Evaluation status	Acc	RMSE
x86 C-model	-	-
stm32 C-model	-	-
original model	-	-
X-cross	100.0%	0.0000000

L2R: 5.83400102e-08

network

network

network Delete network

Show graph

Analyze

Validate on desktop

Validate on target

- Coding

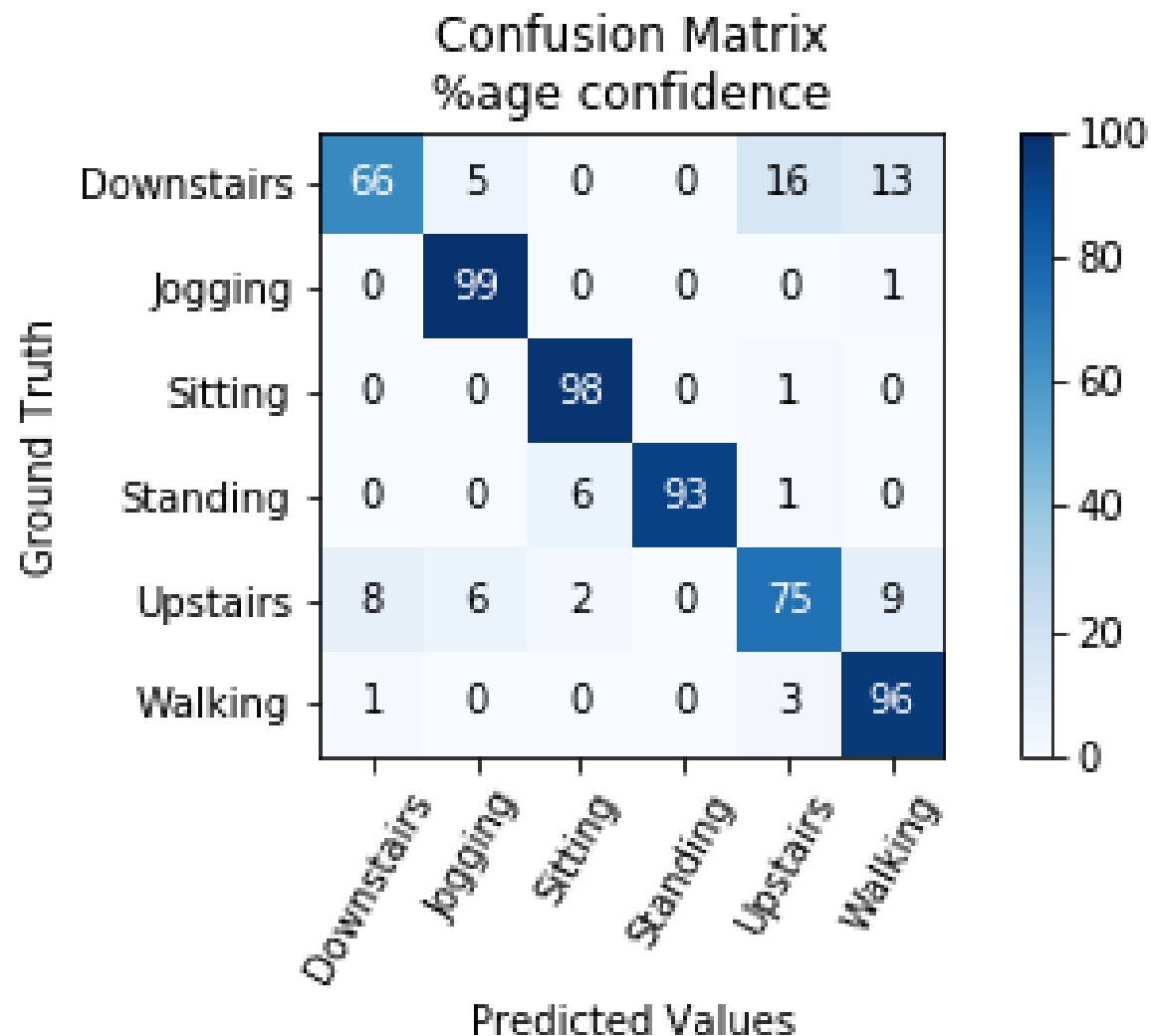
```
//app_x-cube-ai.c
void MX_X_CUBE_AI_Init(void)
{
    AI_ALIGNED(4)
    static ai_u8 activations[AI_NETWORK_DATA_ACTIVATIONS_SIZE];
    aiInit(activations);
}

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>



- 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.
```

