

# STM32 Expert Training – Edge AI based on **STM32N6**

ST KOREA

Blaine MOON

# Agenda

STM32N6 Overview

STM32N6 Architecture & Features

NUCLEO-N657Z0-Q

Hands-On: FSBL + LRUN

Convolution Neural Network

ST Neural-ART Accelerator™

Hands-On AI: Recognizing Handwritten Digits with MNIST



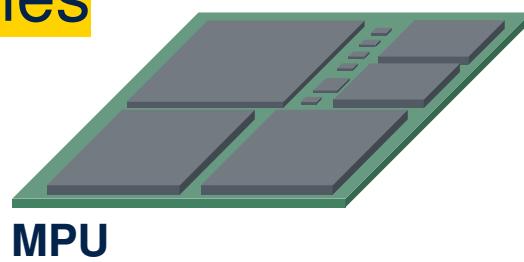


# The first high-performance STM32 MCU with AI acceleration

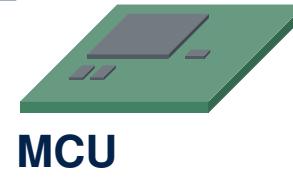
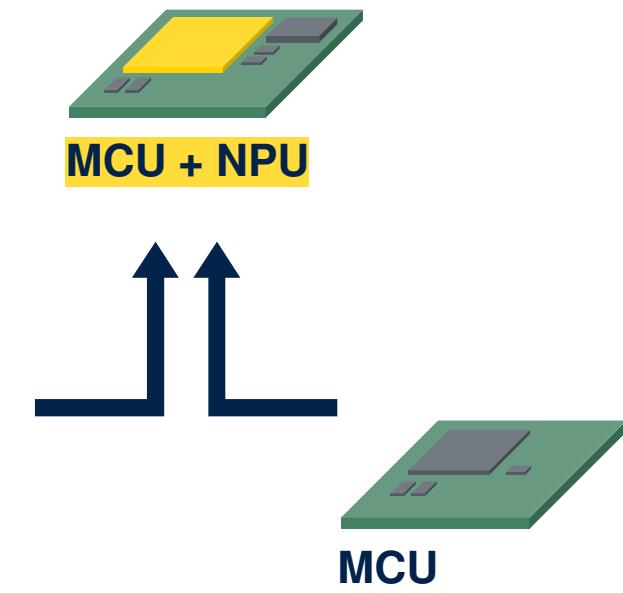
# Enabling unmatched edge AI performance on an MCU

Benefit from extended neural network computing capabilities while leveraging the advantages of an MCU.

High neural processing capabilities



MPU



MCU

Small footprint

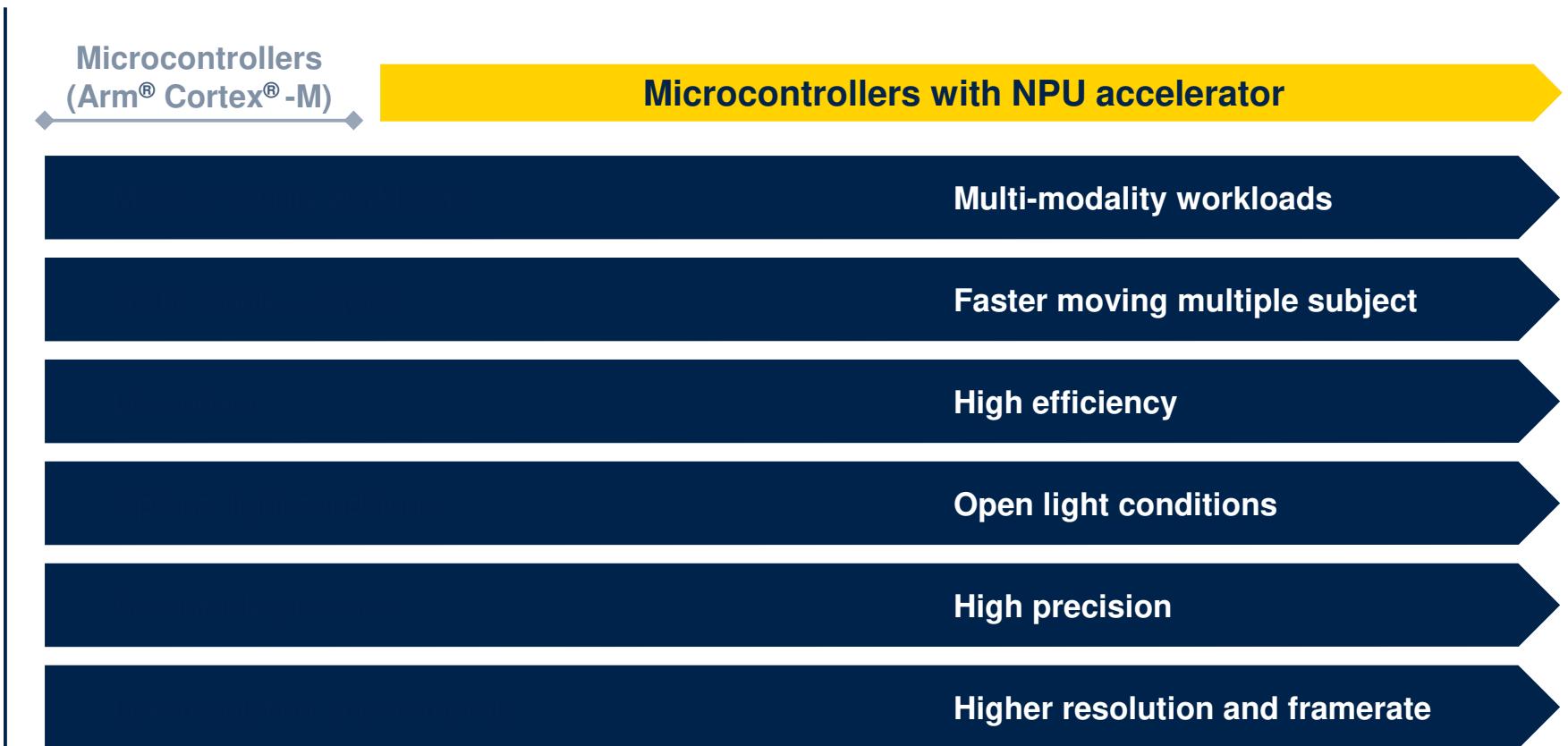
Lower power

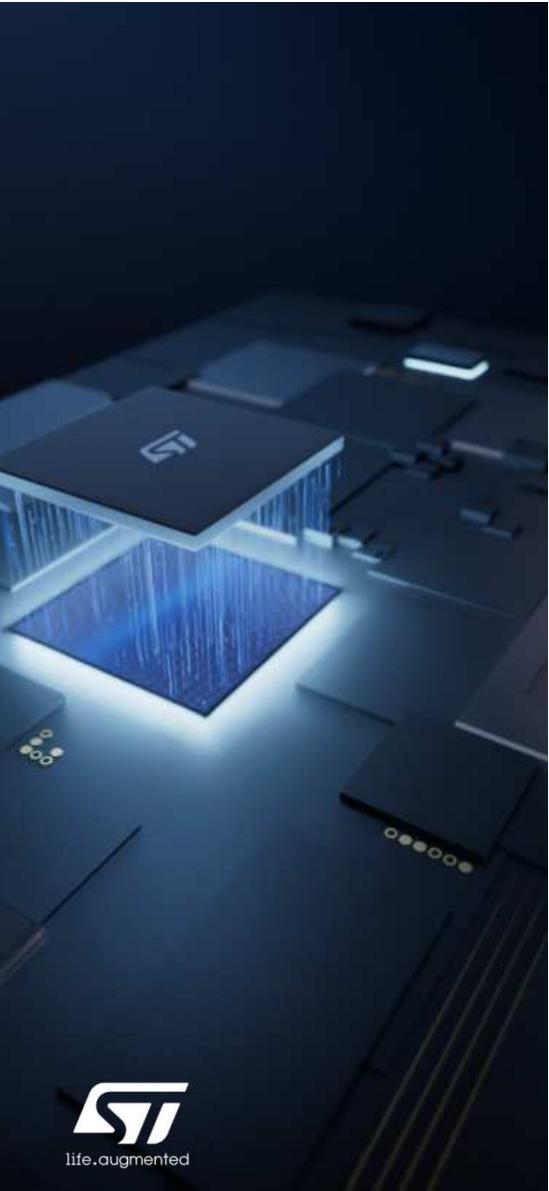
Lower cost

Lower BOM

Faster boot/wkup

# From DMIPS to TOPS, the paradigm shift Opening a new range of embedded AI applications





# Achieve new performance levels with the STM32N6

## Enabling high-performance edge AI on MCUs

- Embedded proprietary neural processing unit, ST Neural-ART accelerator.
- Arm® Cortex®-M55 core at 800 MHz, highest STM32 core frequency yet.

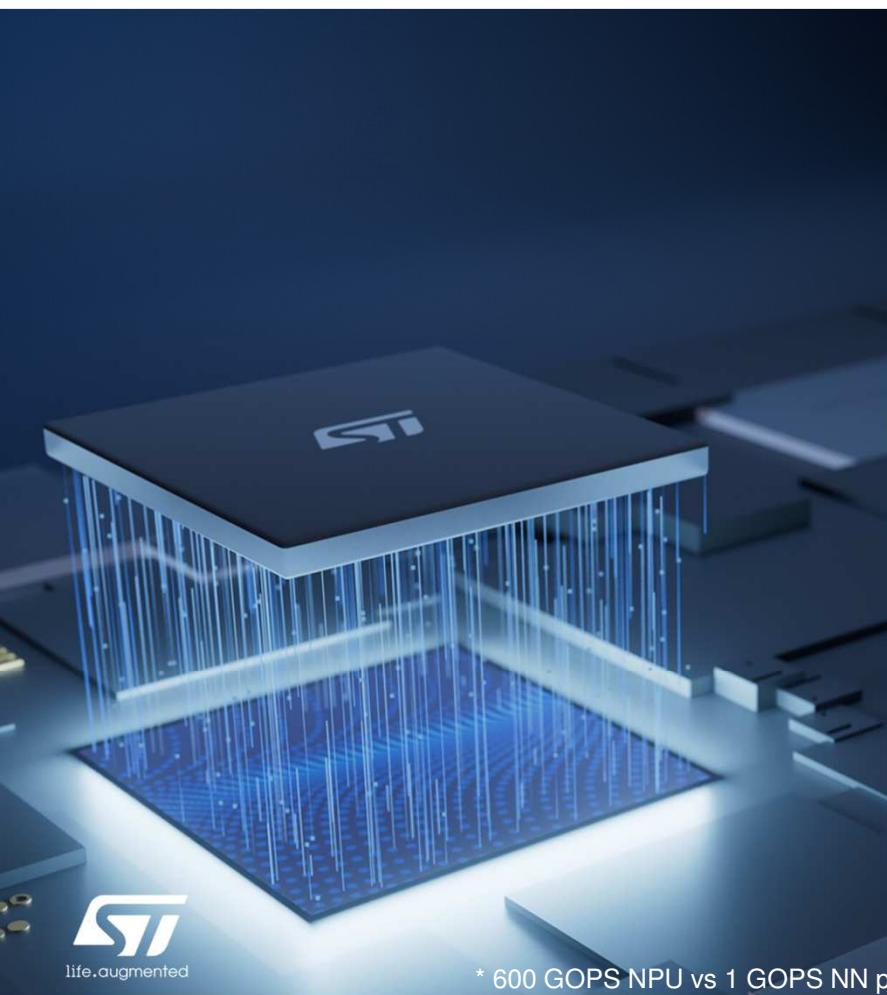
## Elevating graphics & multimedia experiences

- Computer vision pipeline.
- Multiple graphics accelerators.

## Seamlessly integrated in the STM32 ecosystem

- Supported by ST Edge AI Suite tools, resources, and case studies.
- Compatible with the TouchGFX packages for graphics.

# STM32N6 feature overview



## Dedicated embedded neural processing unit (NPU)

- 600 GOPS NPU
- 3 TOPS/W power consumption

## Arm® Cortex®- M55 core

- 1280 DMIPS / 3360 CoreMark
- New DSP extensions (MVE)

## Embedded RAM

- 4.2 Mbytes of embedded RAM for real-time data processing and multitasking

## Computer vision pipeline

- Parallel and MIPI CSI-2 camera module I/F
- Dedicated image processor (ISP)

## Extended multimedia capabilities

- 2.5D graphics accelerator
- H.264 encoder, JPEG encoder/decoder

## Extended security features

- Arm® TrustZone® for the Cortex®-M55 core and the NPU
- Target certifications SESIP3, PSA L3

\* 600 GOPS NPU vs 1 GOPS NN peak processing capabilities on STM32H7

# Embedding innovation across product segments



**Drones**  
Flying & landing



**Smart industry**  
Anomaly detection



**Smart homes**  
Event detection



**Smart farming**  
Animal well-being



**Personal healthcare**  
Body measurements



**White goods**  
Smart control



**Automotive**  
Environment sensing



**Smart buildings**  
Building automation



**Robots**  
Collision detection



**Personal electronics**  
Wearables



# Edge AI



...in a smart & efficient way



**Ultra-low latency**  
Real-time applications



**Privacy & security**  
No raw data sent to the cloud



**Improved accuracy**  
Adapt to local environment



**Sustainable on energy**  
Low-power consumption



**Reduced data transmission**  
Generate meaningful  
information



**Advanced experience**  
Personalized features

# The most powerful STM32 series ever made



Two lines sharing the same DNA

ST  
STM32N6

ST  
STM32N6

ST  
STM32N6

Artificial intelligence  
line

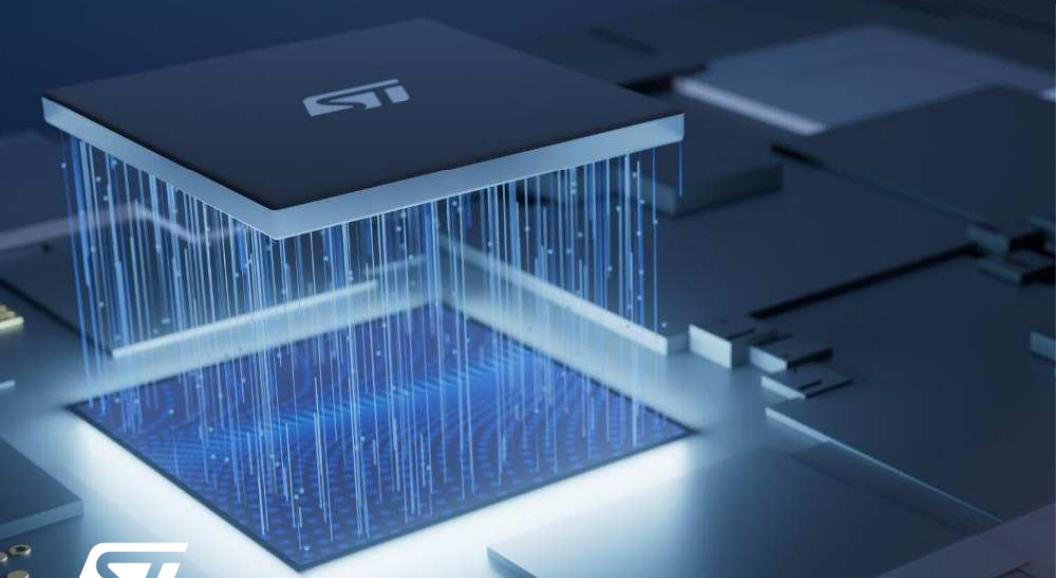
General-purpose  
line

STM32N6x7 MCUs

STM32N6x5 MCUs

# ST Neural-ART Accelerator

**600x**  
ML performance uplift\*

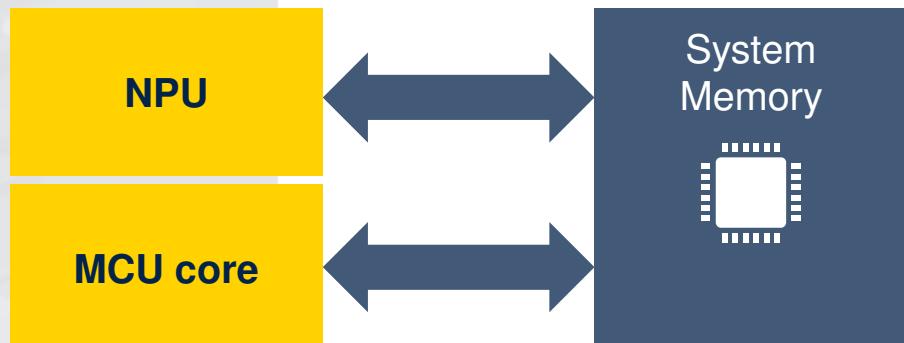


**ST**  
life.augmented

## Dedicated embedded neural processing unit

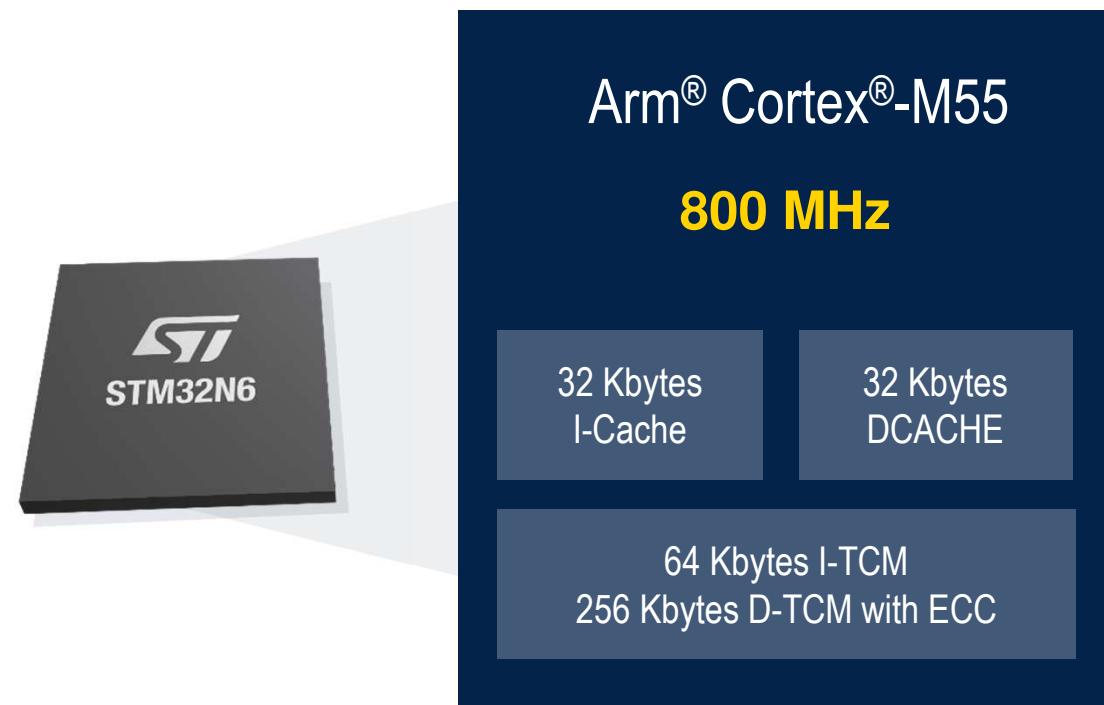
- 600 GOPS
- 3 TOPS/W power consumption
- Cache memory to optimize external memory access

Dataflow stream processing engine  
reduces MCU memory throughput  
requirements and power consumption



\* 600 GOPS NPU vs 1 GOPS NN peak processing capabilities on STM32H7

# A leading MCU Arm® core



## A powerful Arm® core

- 1280 DMIPS / 3360 CoreMark

## Energy-efficient digital signal processing

- Helium technology
- M-Profile Vector Extension (MVE)
- Ideal for signal pre- and post-processing

## Enhanced security with TrustZone®

- Isolate processes in the Arm® core
- Isolate processes in the neural processing unit

# Optimize your application with the large embedded memory

**Large embedded RAM**  
**4.2 Mbytes**



**Fast external memory I/F**

Hexa-SPI

Up to 800 Mbytes/s

Octo-SPI

Up to 400 Mbytes/s

FMC

Up to 664 Mbytes/s

## **Large contiguous embedded memory**

- Ideal for running neural networks or graphic applications
- External RAM becomes optional

## **Fast serial I/F for external memories**

- Allows the use of fast and cost-effective memory
- Hexa-SPI for fast access to RAM
- Octo-SPI for secured flash memory

## **Flashless configuration**

- Adaptability to application requirements
- Enabling cost flexibility

## **Flexible memory controller**

PSRAM, SDRAM, NOR, NAND

## **Improved security with on-the-fly encryption**

Hardware-accelerated crypto engine on all interfaces



# Elevating graphics performance

## Graphic accelerators

NeoChrom  
GPU

- 2.5D GUI acceleration
- Perspective correct texture mapping (scale, rotate, flip)

Chrom-ART  
Accelerator

Efficient 2D graphics sub-system

JPEG codec

MJPEG video coding & decoding

Chrom-GRC

Framebuffer optimization

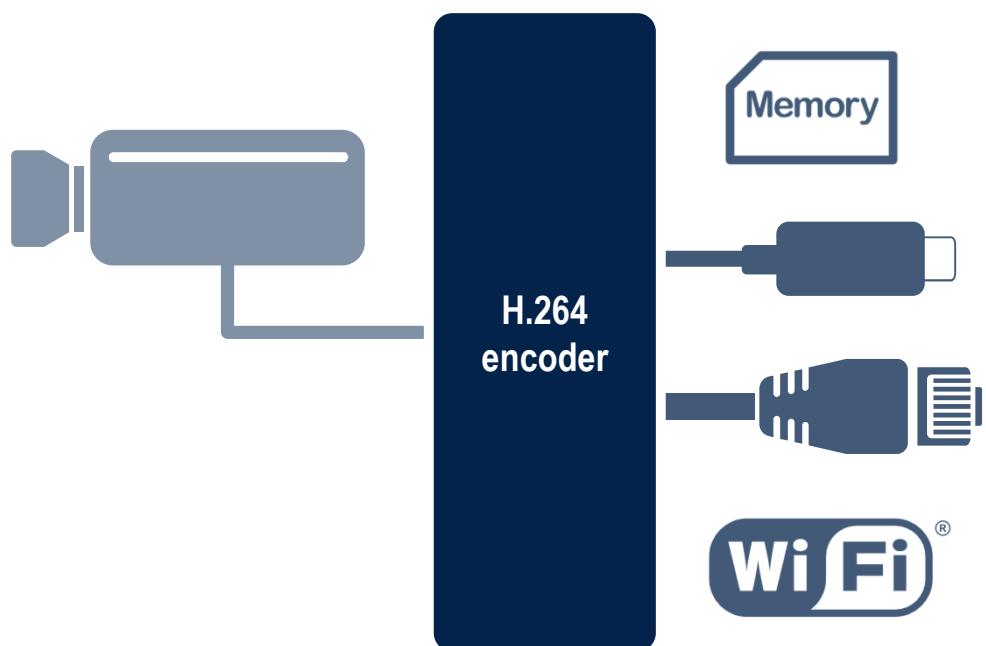


# Elevating multimedia experiences

## Multimedia unit

H.264  
encoder

- 1080p15 and 720p30
- Real-time streaming over USB, over Ethernet, Wi-Fi



# Geared for computer vision applications

Enabling fast & efficient image acquisition and processing thanks to a widely adopted camera interface and embedded ISP.

## CAMERA PIPELINE

HR image sensor



←→  
MIPI CSI-2  
Parallel Camera I/F



### Image signal processor (ISP)

- Dimensioned for 5 Mpixel camera at 30 FPS
- Generates 3 different outputs from the same input for sending to the multimedia encoder or to the NPU
- [ISP IQTune Software tool](#) to tune ISP for cost savings and design flexibility

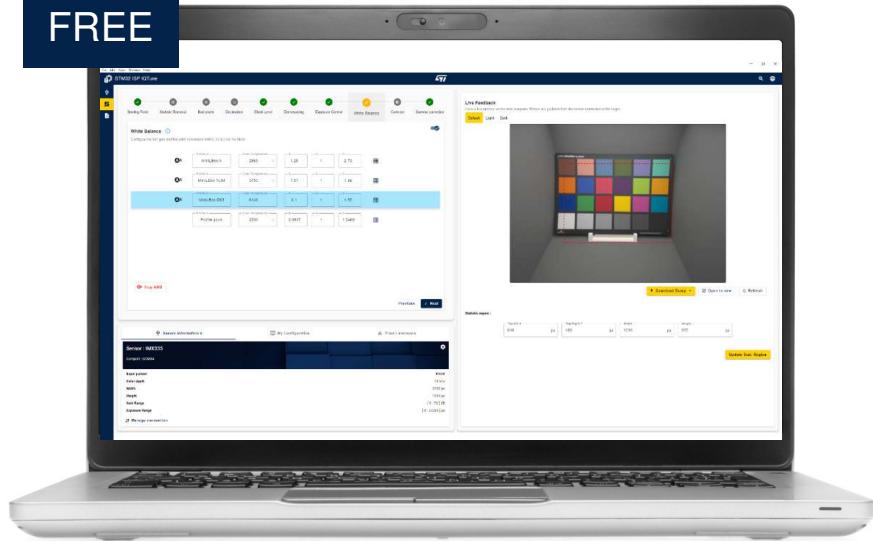
### Embedded firmware on Arm® Cortex® core

- 2A for auto white balance and auto exposure
- Image processing library



# Configure the image signal processor for free

FREE



**Industry-first software tool for ISP tuning on MCUs and MPUs.**

**Save ISP tuning cost and gain efficiency.**

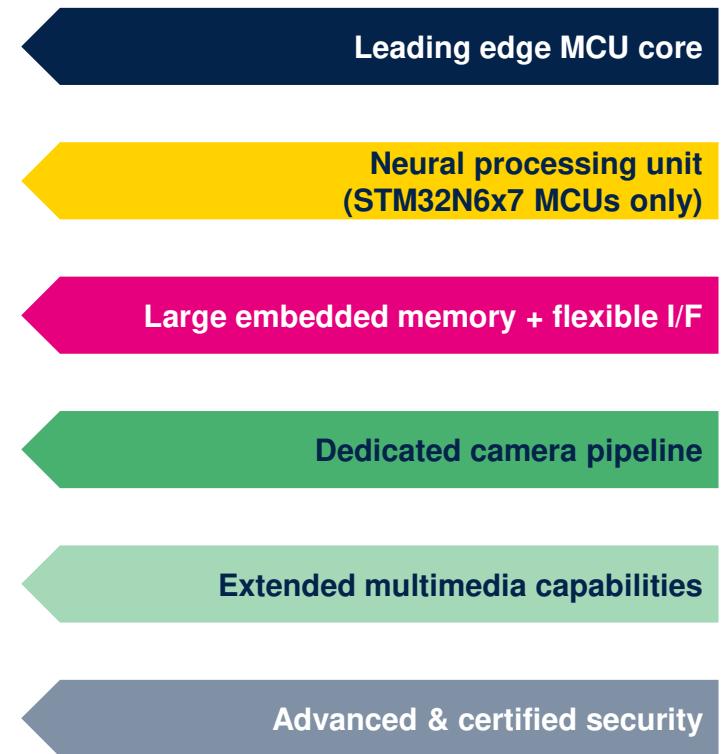
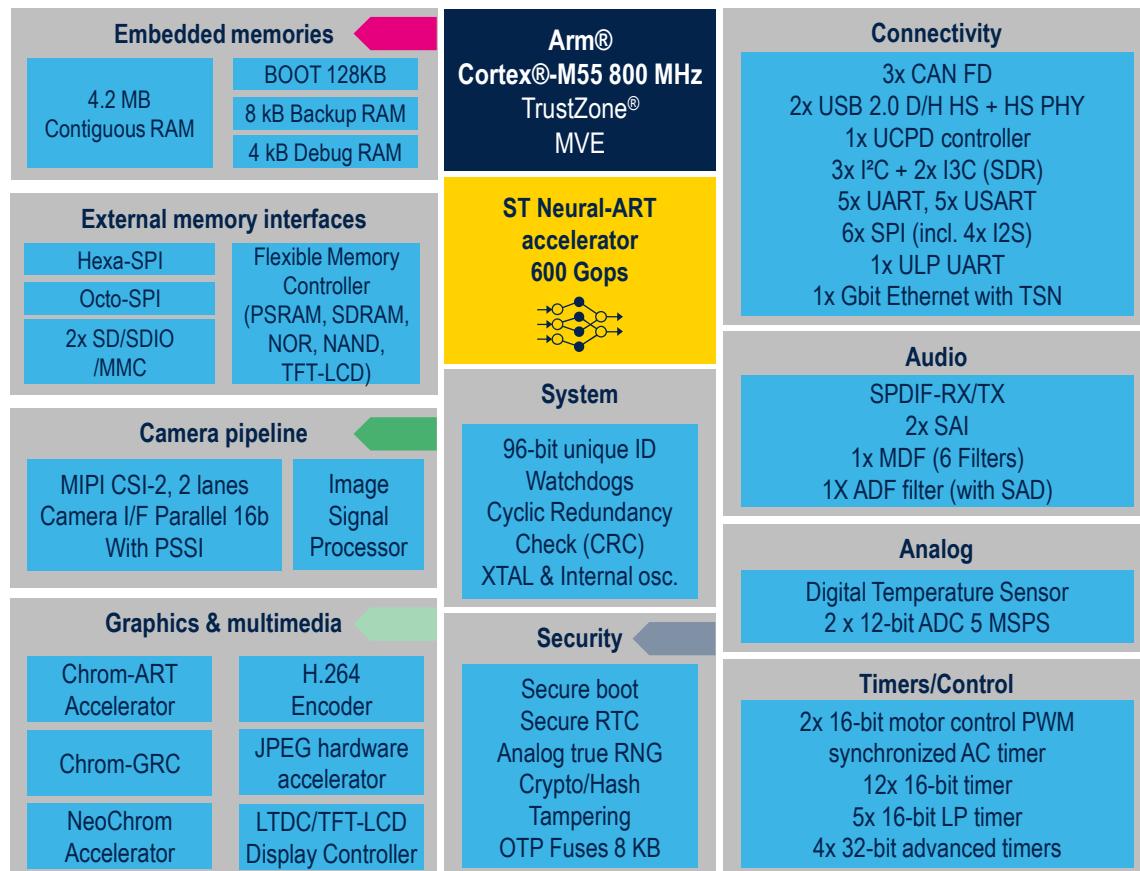
**Flexibility to configure the ISP to your application requirements.**



[Access now](#)



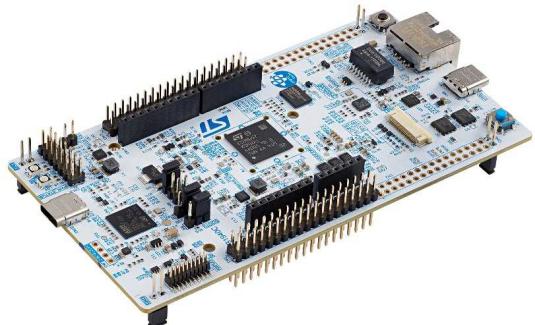
# STM32N6x7 and STM32N6x5 MCUs





# Development tools for STM32N6 series

**Jump-start your evaluation, prototyping, and design**



**NUCLEO-N657X0-Q**

**Affordable prototyping**

STLINK v3, ST morpho, ARDUINO®, MIPI  
CSI-2 connector, USB 2.0, 1GB Ethernet

Camera connector compatible with Raspberry.



**STM32N6570-DK**

**Advanced prototyping including AI**

STLINK v3, ST morpho, Arduino®, MIPI  
connector, USB 2.0, 1 Gbyte Ethernet, 32  
Mbytes HexaRAM, Audio Jack, SD card

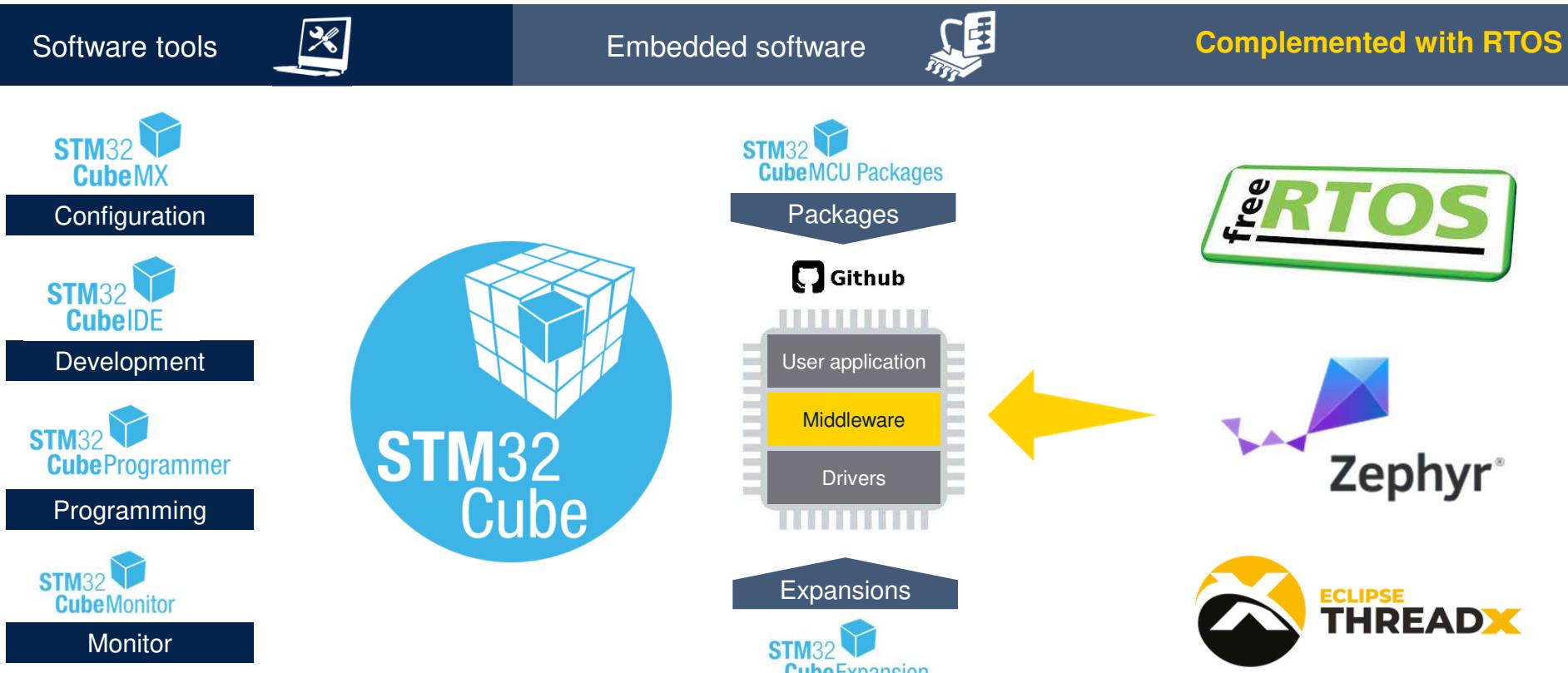


**B-CAMS-IMX expansion board**

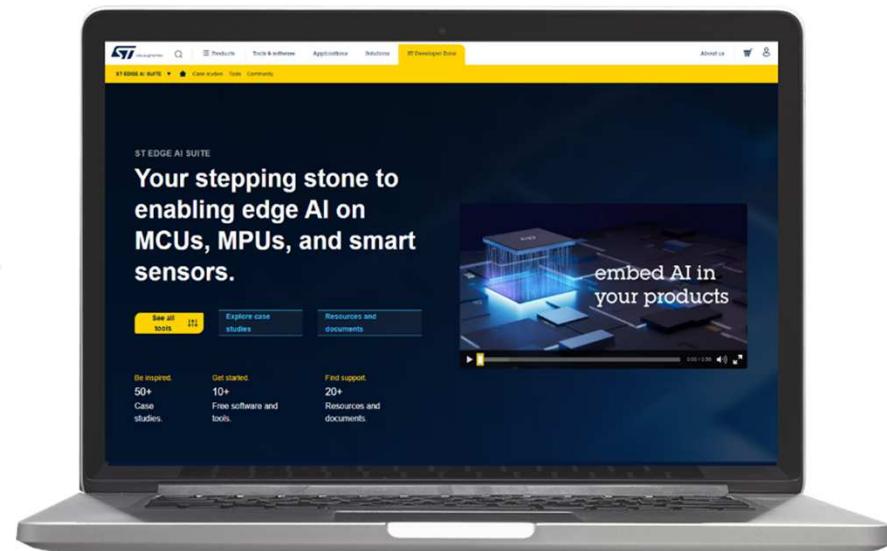
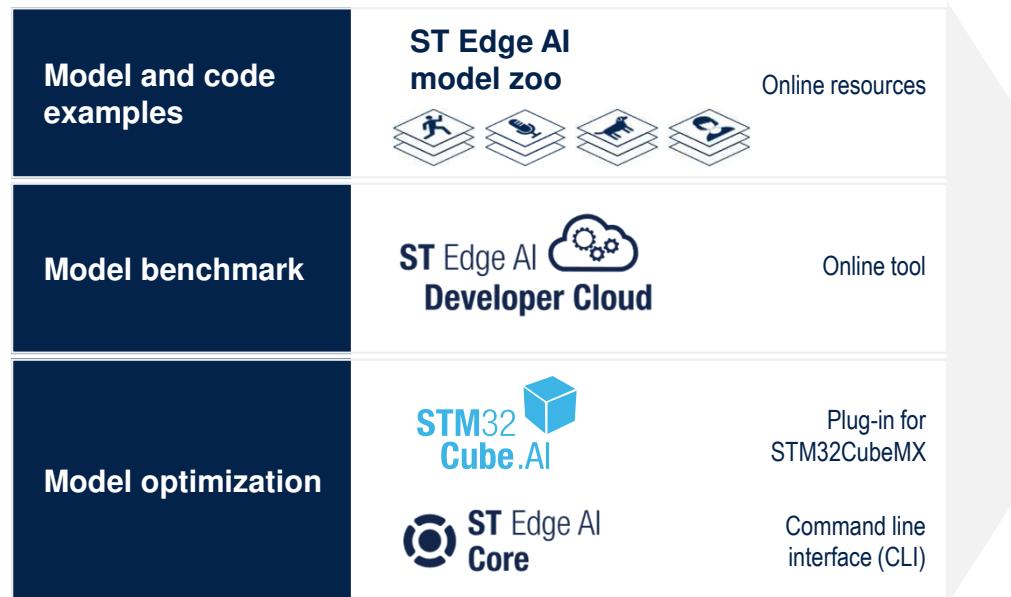
Rolling shutter camera, M12 removable lens,  
multizone direct Time-of-Flight sensor, inertial  
motion unit, Raspberry Pi compatible 22-pin  
connector.

*Included in discovery kit.*

# Leveraging STM32Cube software suite



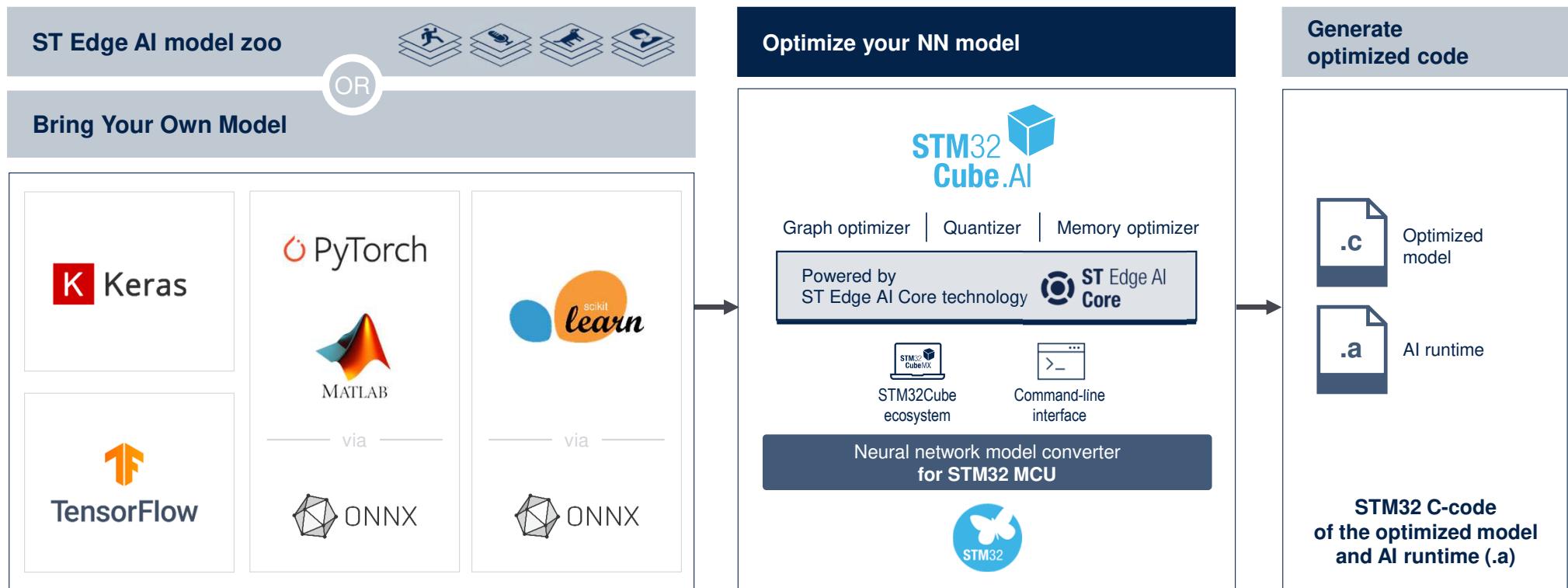
# Online development tools for edge AI development



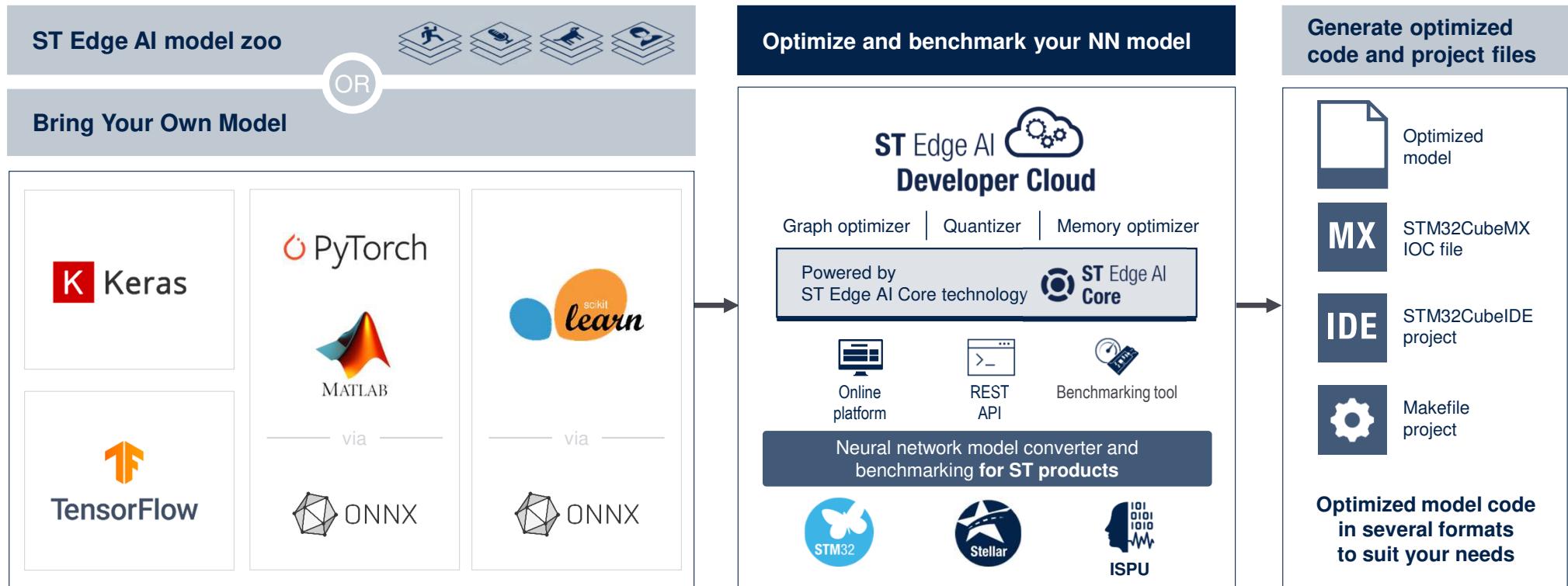
[st.com/st-edge-ai-suite](http://st.com/st-edge-ai-suite)



# STM32Cube.AI



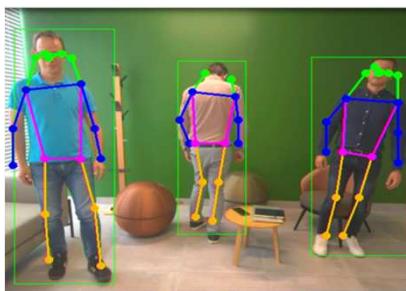
# ST Edge AI Developer Cloud



# Get started with edge AI examples



## People detection



## Multipose estimation



## Hand landmark detection



## H264 encoding / USB UVC streaming



## Power measurement

- Application example showing a people detection use case.
- Demonstrating typical AI computer vision application: camera capture, pre-processing, single model inference and post-processing.
- RTOS-based application example.

- Application like people detection but built around a multi-pose estimation use case.
- RTOS-based application example.

- Application example showing hand landmark detections.
- Demonstrating the execution of two NN models consecutively.
- RTOS-based application example.

- Demonstrating a more complete application involving several STM32N6 multimedia features: NPU to perform the inference, H264 encoding and USB video device class stream output data to a PC.
- RTOS-based application example.

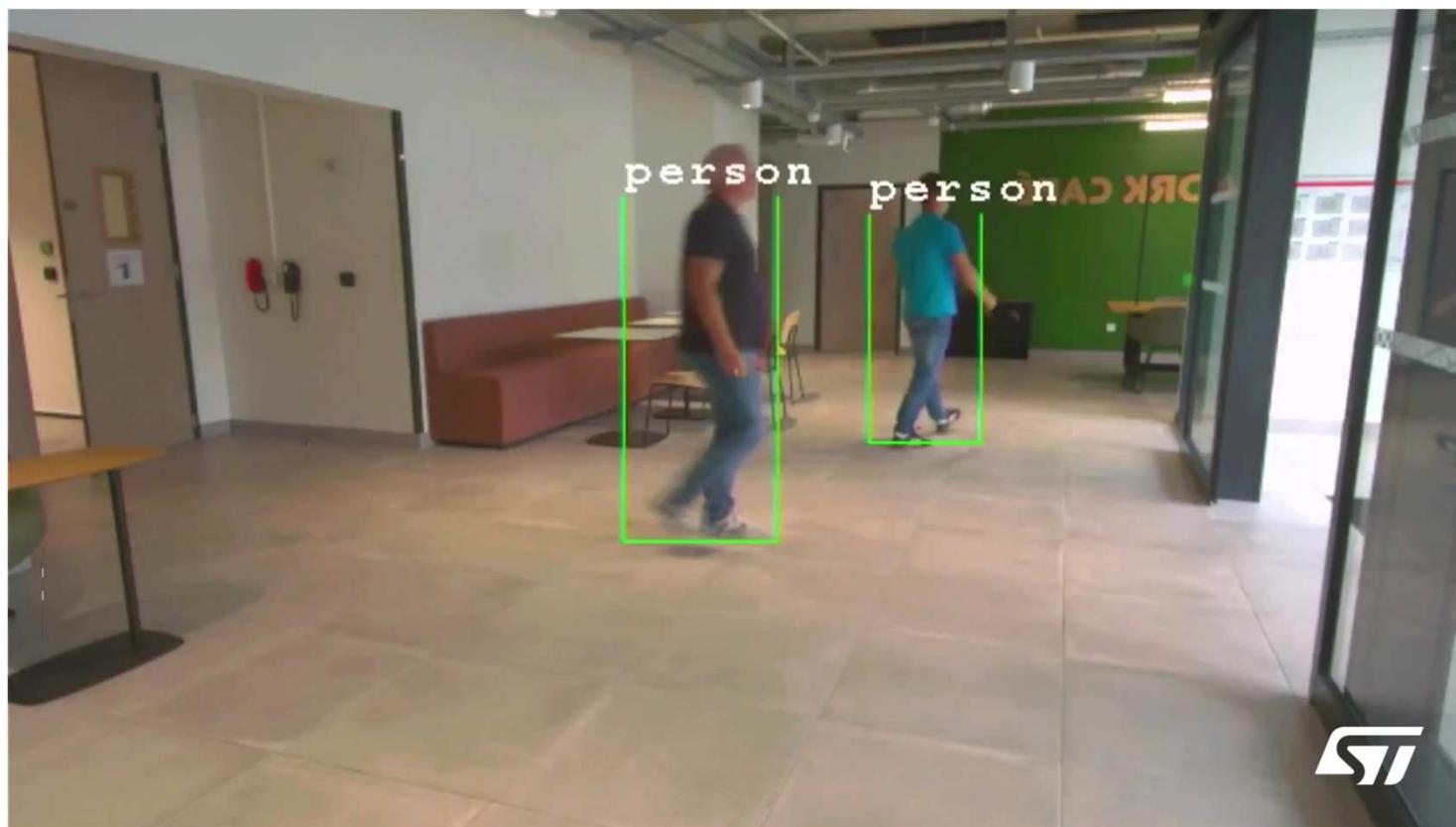
- Demonstrating low power optimizations.
- Enabling easy power measurement on STM32N6 discovery board.
- Bare-metal application example.



Access in the source code [here](#)



# High-accuracy people detection at a distance in varied ambient conditions



## KEY METRICS

### Yolo v8

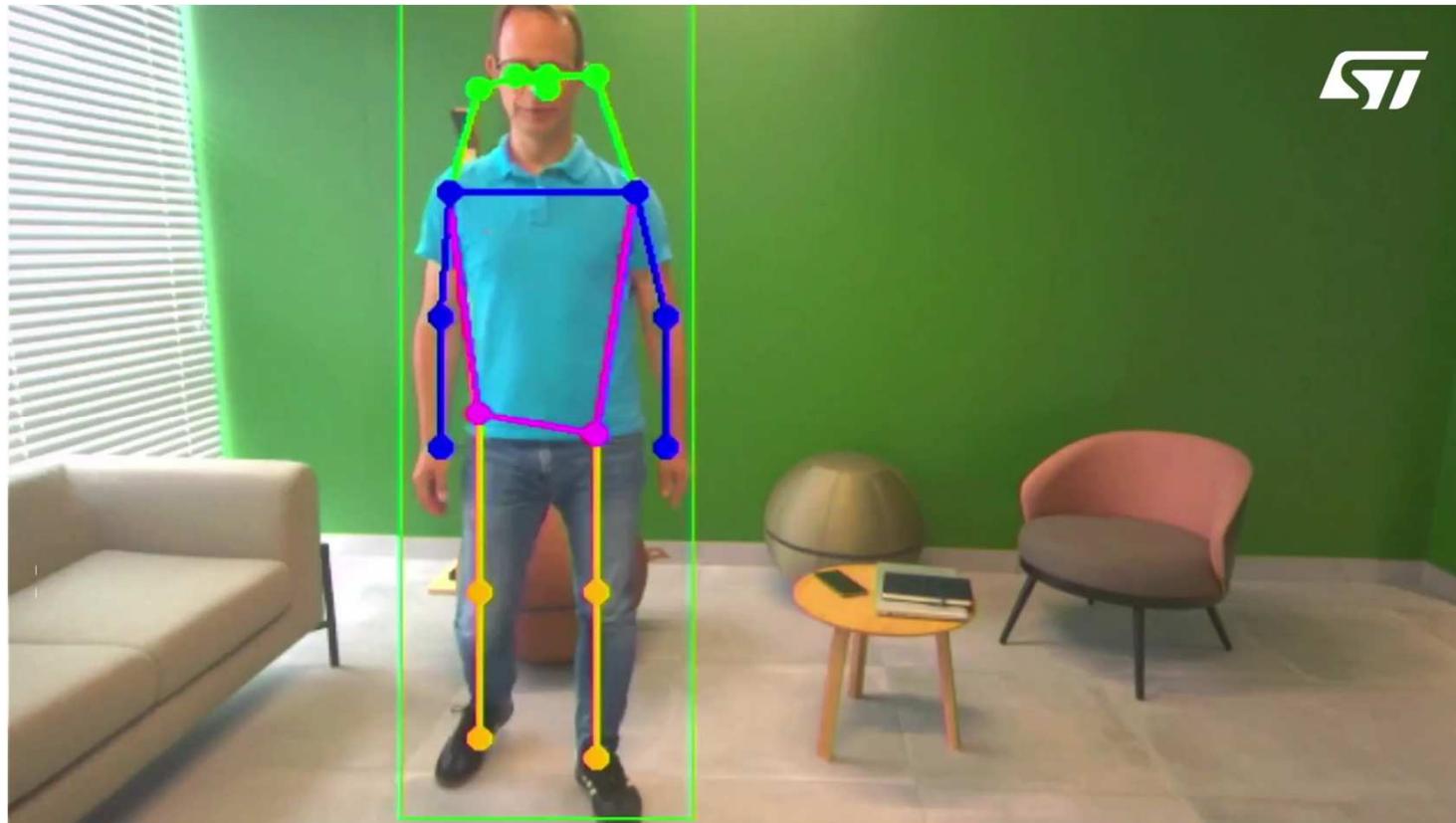
- 320 x 320
- 26 FPS
- 2.9 MB weights
- 1.6 MB activations

## KEY APPLICATIONS

- Smart doorbells
- Room occupancy
- Alarm systems



# High-accuracy multipose estimation



## KEY METRICS

### Yolo v8n

- 256 x 256
- 26 FPS
- 3.35 MB weights
- 2.59 MB activations

## KEY APPLICATIONS

- Behavior analysis
- Activity monitoring
- Fall detection



# Precise system control with hand landmark



The image is split into two main sections. On the left, a person's profile is shown looking at a screen. The screen displays the ST logo and the text "Advance your product offering with edge AI". Below this, there is a "ST Edge AI Suite" logo with a QR code and the text "Start your journey now". On the right, a close-up of a person's hand is shown with yellow dots connected by lines, forming a polygonal outline that tracks the hand's movement. The background is a light blue gradient.

**KEY METRICS  
(Two models in parallel)**

**Palm detector**

- 192 x 192
- 1.1 MB weights
- 1.1 MB activations

**Hand landmark**

- 224 x 224
- 3.2 MB weights
- 1 MB activations

**KEY APPLICATIONS**

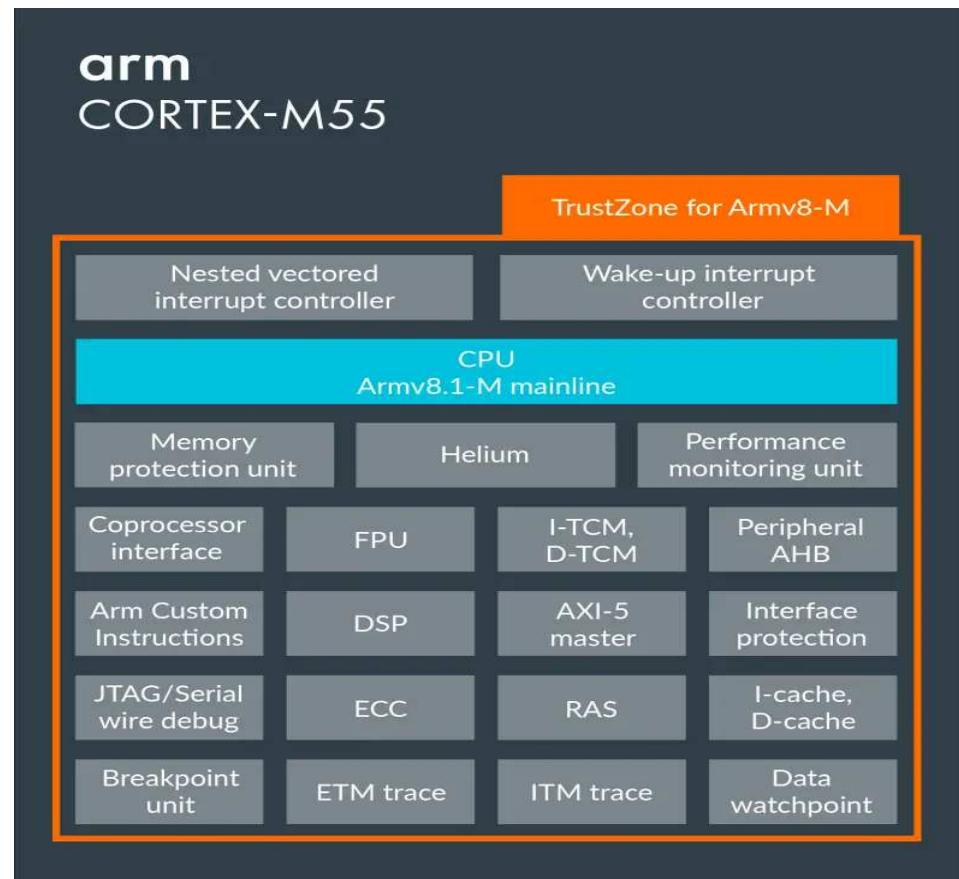
- Touchless HMIs
- Smart appliances
- Smart industry

# **STM32N6 Architecture & Features**

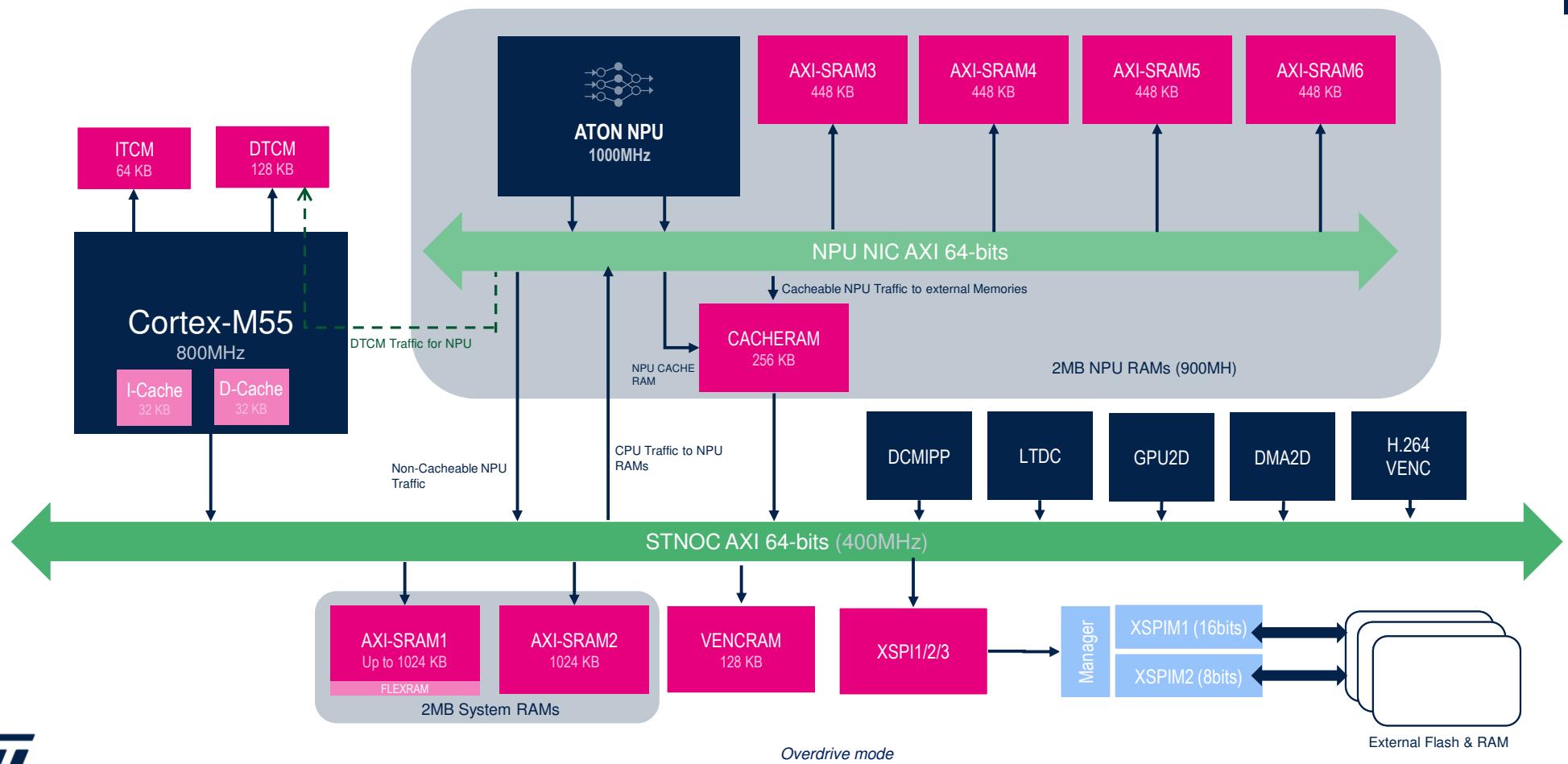


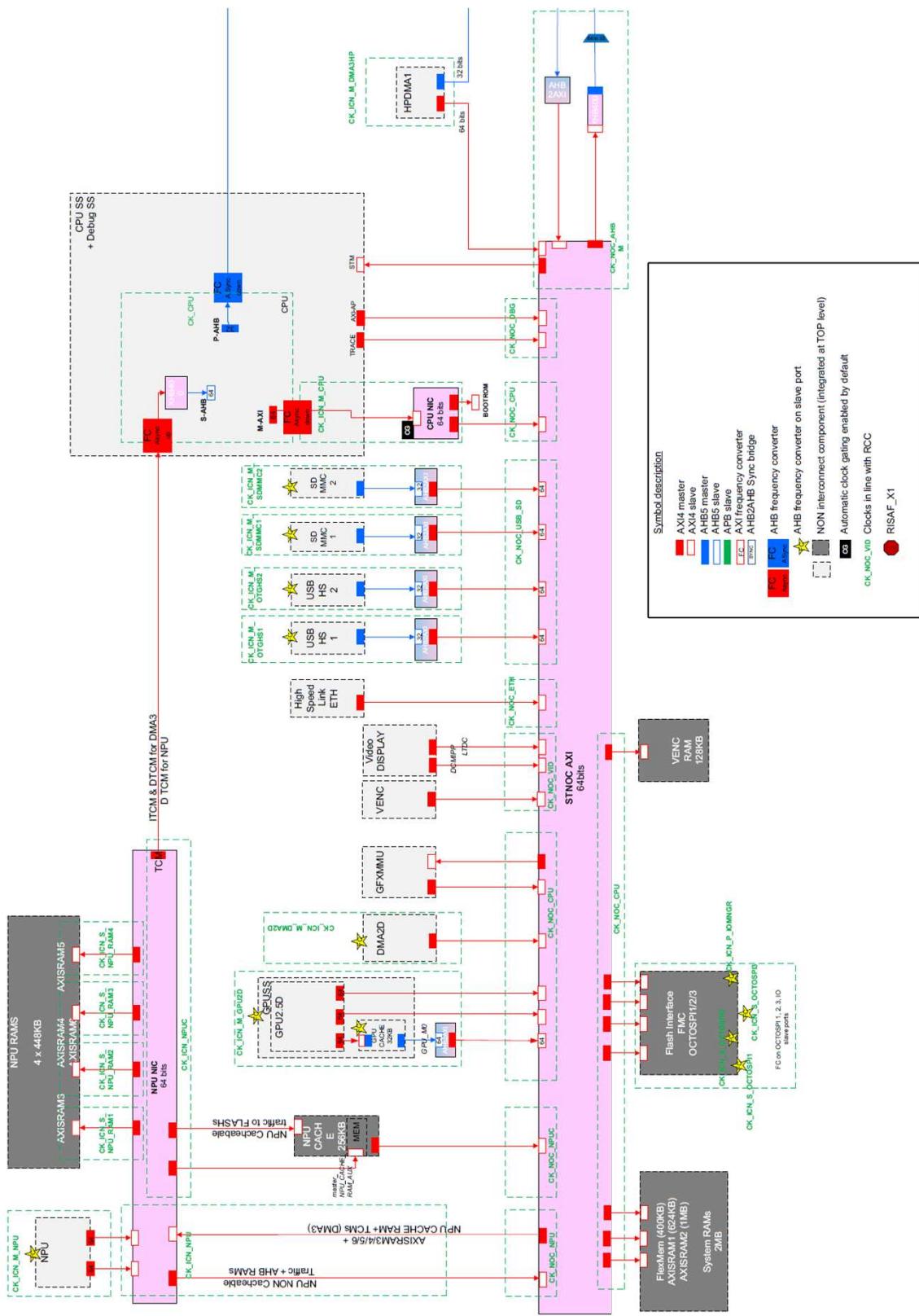
# Cortex®-M55 processor overview

- Armv8.1-M architecture
  - Floating-point Unit (FPU) with support for multiple floating-point formats
  - Based on a 4-stage integer pipeline design + Helium vector processing (increase the 5 total pipeline)
- Memory Protection
  - Advanced memory caches and tightly coupled memory (TCM) support
  - Security Attribution Unit (SAU)
  - Memory Protection Units (MPUs)
- Integrated Nested Vectored Interrupt Controller (NVIC) – handle all N6 Iips interrupts
- System security = TrustZone® for Armv8.1-M
- Invasive and non-invasive debug



# STM32N6 Block Diagram





# STM32N6 system RAM List

Memory	Frequency	Size	RIF Access Control	Retention	Erase on reset	Comments
Embedded memories	BOOTROM	400 MHz	128 KB	●	-	-
	AXISRAM1	400 MHz	624 KB	●	-	HW & SW
	AXISRAM2	600 MHz (TCM) 400 MHz (RAM)	1024 KB	●	-	HW & SW
	FLEXRAM	600 MHz	Up to 400 KB	●	(●) -Standby	SW  Only the 80 Kbytes that correspond to the first portion of the I-TCM extension can be retained
	AXISRAM3	800 MHz (NPU)	448 KB	●	-	SW
	AXISRAM4	800 MHz (NPU)	448 KB	●	-	SW
	AXISRAM5	800 MHz (NPU)	448 KB	●	-	SW
	AXISRAM6	800 MHz (NPU)	448 KB	●	-	SW
	AHBSRAM1	200 MHz	16 KB	●	-	SW
	AHBSRAM2	200 MHz	16 KB	●	-	HW & SW
	VENCRAM	?? MHz	128 KB	●	-	SW
	BKPSRAM	?? MHz	8 KB	●	● (Standby & Vbat)	● On VSW



# Memory Map

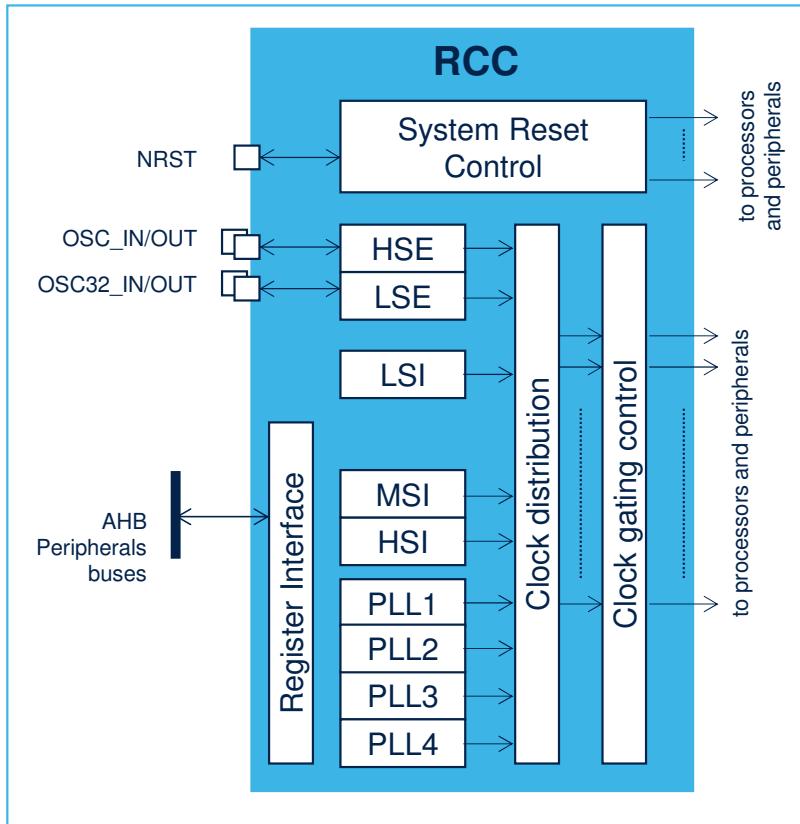
Description	Non-Secure	Secure	Size
ITCM – Base line	0x0000 0000	0x1000 0000	64KB
ITCM – FLEXMEM extension			Up to 192KB
<b>Boot ROM</b>	0x0800 0000	0x1000 0000	128KB
DTCM – Base line	0x2000 0000	0x3000 0000	128KB
DTCM – FLEXMEM extension	0x2002 0000	0x3002 0000	Up to 128KB
<b>AXI-SRAM1</b> (FLEXMEM extension bites on the lower end)	0x2400 0000	0x3400 0000	Up to 1MB
<b>AXI-SRAM2</b>	0x2410 0000	0x3410 0000	1MB
<b>AXI-SRAM3</b>	<b>0x2420 0000</b>	0x3420 0000	448KB
<b>AXI-SRAM4</b>	<b>0x2427 0000</b>	0x3427 0000	448KB
<b>AXI-SRAM5</b>	<b>0x242E 0000</b>	0x342E 0000	448KB
<b>AXI-SRAM6</b>	<b>0x2435 0000</b>	0x3435 0000	448KB
<b>NPU CACHE RAM</b>	0x243C 0000	0x343C 0000	256KB
<b>VENC RAM</b>	0x2440 0000	0x3440 0000	128KB
<b>AHB-SRAM1</b>	0x2800 0000	0x3800 0000	16KB
<b>AHB-SRAM2</b>	0x2800 4000	0x3800 4000	16KB
<b>NOR/SRAM</b> - Remap SDRAM 1 (through FMC)	0x6000 0000		Up to 256MB
<b>XSPI2</b>	0x7000 0000		Up to 256MB
<b>XSPI3</b>	0x8000 0000		Up to 256MB
<b>XSPI1</b>	0x9000 0000		Up to 256MB
<b>SDRAM 1</b> - Remap NOR/SRAM bank (through FMC)	0xC000 0000		Up to 256MB
<b>SDRAM 2</b> - Reserved (through FMC)	0xD000 0000		Up to 256MB

After reset, the Cortex-M55 starts in Boot ROM. This ROM code is used to authenticate and decrypt images loaded into the internal memory from an external flash memory, or through UART, USB, I2C, SPI, ...

Up to **4MB** contiguous RAM accessible by the NPU

Not connected to NPU NIC

# Reset and clocking (RCC)



- The Reset and Clock Controller (RCC) manages:
  - The generation of all the clocks,
    - up to 4 x PLLs, 3 x RC oscillators, 2 x Crystal oscillators...
  - The gating of all the clocks
    - Possibility to enable/disable clocks for each peripheral
  - The control of all the system and peripheral resets.

## Application benefits

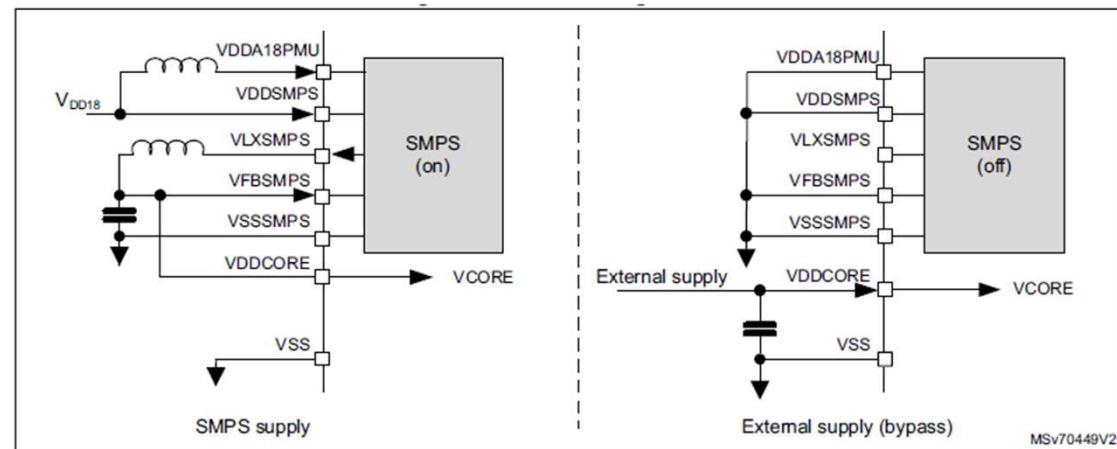
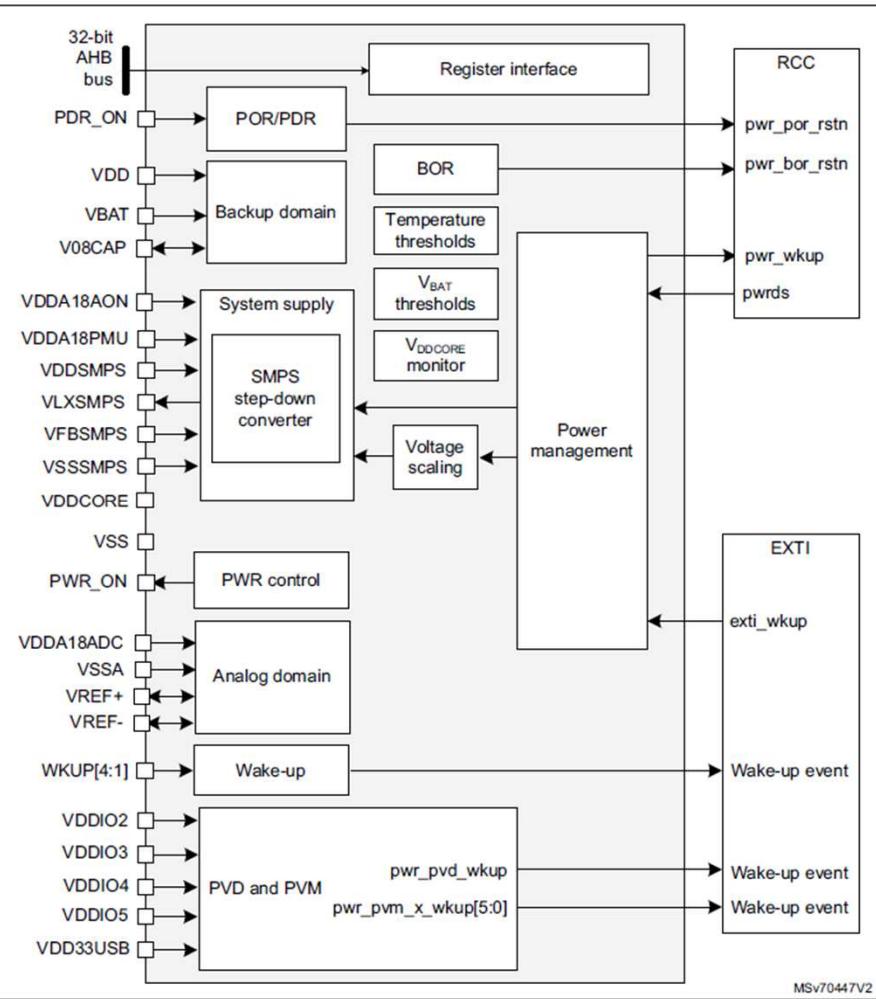
- High flexibility regarding the clock sources to meet consumption and accuracy requirements.
- Safe and flexible reset management

# STM32N6xx - Power supplies

Name	Typical or Range	Description
<b>VDD</b>	<b>1.8V / 3.3V</b>	Power supply for most I/Os
<b>VDDIO2/3/4/5</b>	<b>1.8V / 3.3V</b>	Independent power supply for IOs related to SDMMC1, SDMMC2, XSPIM1 and XSPIM2
<b>VDDA18AON</b>	<b>1.8V</b>	Analog power supply for system analog such as reset, power management, oscillators and OTP
<b>VDDCORE</b>	<b>0.8V (0.91V overdrive)</b>	Digital power supply for Core Cortex-M55 domain. Overdrive available on some part number to get higher CPU frequency
VDDSMPS, VLXSMPS, VFBSMPS, VSSSMPS		Step-down converter , SMPS connected
<b>VDDCSI</b>	<b>0.8V (0.91V overdrive)</b>	Power supply for CSI PHY digital supply input. usually be connected to VDDCORE
<b>VDDA18PLL</b>	<b>1.8V</b>	Analog Power supply for RCC PLLs
<b>VDDA18CSI, VDDA18USB</b>	<b>1.8V</b>	Analog power supply for various PHY
<b>VDDA18ADC</b>	<b>1.8V</b>	Analog Power supply for ADCs and VREFBUF
Vref+, Vref-		external reference voltage for ADCs, independent from any other supply
<b>VDDA18PMU</b>	<b>1.1V / 1.2V / 1.35V</b>	Power supply for DDR Physical Interface (PHY) and related IOs, respectively for LPDDR4, DDR4 and DDR3L
<b>VDD33USB</b>	<b>3.3V</b>	Power supply input for respectively USB PHY and USB Type-C Power Delivery PHY and related IOs
<b>VBAT</b>	<b>1.62 – 3.6V</b>	Backup supply, usually connected to a 3V coin-cell battery
<b>Internally generated Power Supplies</b>		
<b>V08CAP</b>	<b>0.8V</b>	Regulator decoupling for backup domain and Smart Run Domain. Internally supplied by either VDD or VBAT



# STM32N6xx - Power supplies



- Two configurations exist :
  - SMPS step-down converter supply on
  - SMPS step-down off (bypass mode – Internal converter is disable)



# Power controller (PWR)

System mode	Description	Wakeup
Run	CPU in Run mode	-
Sleep	Peripheral clock OFF (PERxLPEN bit, operate accordingly)	
Stop SVOS high	CPU clock OFF . System clock OFF. Peripheral clock OFF (if PERxLPEN bit, operate accordingly)	RTC/TAMP, EXTI, some peripherals, WKUP pins
Stop SVOS low	CPU clock OFF . Some platform supplies are powered down. VDDCORE voltage could be lowered. System & peripheral clock OFF	RTC/TAMP, EXTI, WKUP pins
Standby	Most platform supplies are powered down. VDDCORE supply is powered down.	RTC/TAMP, WKUP pins, external NRST, IWDG reset
OFF/VBAT	All Platform, Core and IOs supplies are powered down. VSW domain may be active  D3 domain could be wakeup (powered by VBAT) upon TAMP request	RTC/TAMP event could request PMIC to restore power supplies TAMP event could wake D3 domain



BOOT0	BOOT1	Boot Source	interface	word OTP11 Value
-	1	Development Boot		
0	0	Flash boot	XSPI serial NOR (in SPI mode, single)	word OTP11 [8:5] = 0x0011
			XSPI HyperFlash™ (8-bit)	word OTP11 [8:5] = 0x0101
			e.MMC™ SDMMC1	word OTP11 [8:5] = 0x0010
			e.MMC™ SDMMC2	word OTP11 [8:5] = 0x1000
			SD-Card SDMMC1 (up to SD standard v6.0)	word OTP11 [8:5] = 0x0001
			SD-Card SDMMC2 (up to SD standard v6.0)	word OTP11 [8:5] = 0x0111
1	0	Serial boot	USB boot: USB 2.0 OTG HS	word OTP11 [16:9] = 0x1111111x x= 0 USB enable , x=1 USB disable
			UART boot (Default mode)	word OTP11 [16:9] = 0x111111x1 x= 0 UART enable , x=1 UART disable

The boot mode is determined by BOOT0 and BOOT1 pins, and one OTP word.

The BOOT1 is a non dedicated boot pin.

If Flash boot selected, the flash device selection is done using OTP fuses

If Serial boot selected, Interfaces could be disabled by fuse

Note : Boot1 pin check has priority on Boot0 pin check. If Boot1 pin is not set, Boot0 is checked. If Boot1 pin is selected but not allowed in current lifecycle, Boot0 pin is checked

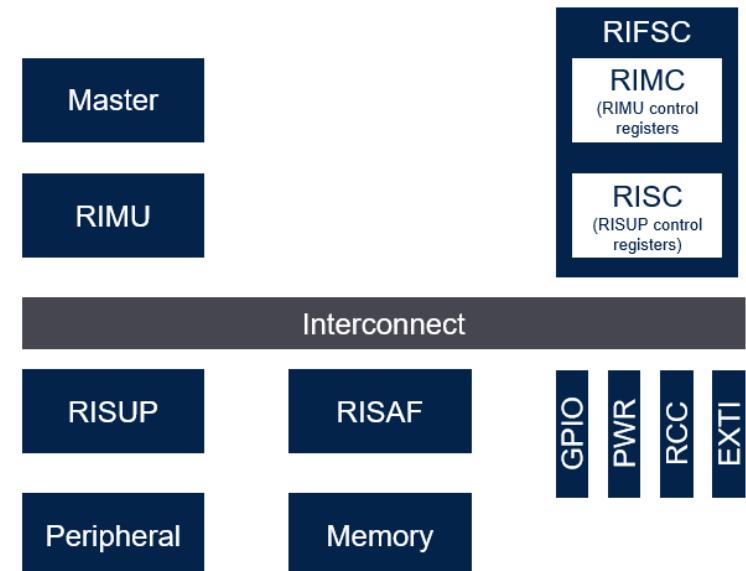
# Resource isolation

- Components

- RIMU: Placed on AXI bus initiators to determine which attributes are attached to bus accesses
  - S/NS, U/P, R/W, CID
- RISUP: Firewall placed in front of peripheral bus targets to filter control accesses.
- RIFSC : Centralized configuration of all the RIMUs and RISUPs
- RISAF : Firewall placed in front of memory target, configured locally
- IAC : Illegal access controller, recording which firewall has detected a violation.

- Compartments

- CID : Compartment Identifier: 3-bit field programmable in each RIMU
- Memory RISAF regions can be configured to filter on CID.
- N6 allows the OS to delegate a peripheral DMA to a compartment
  - Can ensure that the peripheral DMA can only access data belonging to that compartment.
  - CPU CID is always 1, DAP CID is always 7



# OTP fuses

- Total : 1.5 Kbyte fuses

- OTP Fuses are One Time Programming memory
  - Initial bits are '0' and are irreversibly programmed to '1'
  - Incremental programing of bits in a 32-bit word is possible on lower 4096 bits of the OTP
- Handled thru BSEC controller IP
  - Programming, reading, status and locking handled by BSEC
  - Lock mechanism to avoid read and/or program (32-bits granularity)
- OTP Content
  - Product configuration and Trimming values set by ST during production
  - Secrets and unique identification numbers set by ST during production
  - Device configuration set by OEM (e.g. MAC address, boot source, security mode, etc...)
  - Secrets set by OEM (e.g. for secure boot)
  - Total of 12288 bits, up to 3200 bits + 3328 bits available for respectively OEM data and OEM secrets



# OTP fuses

RM0486

## OTP mapping (OTP)

Table 18. OTP fuse description (lower OTP region) (continued)

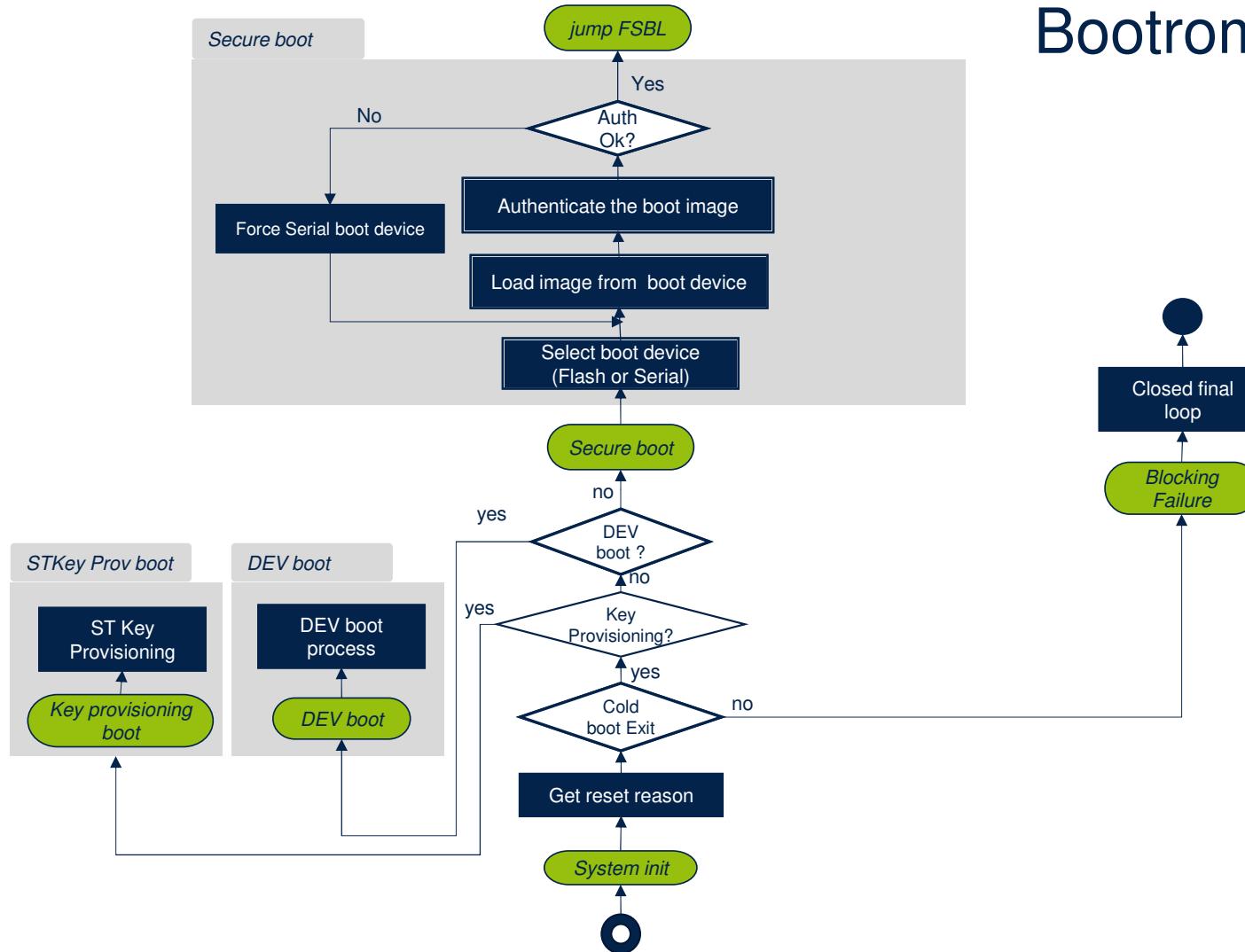
OTP word	OTP bits	Name or description	Detailed description
OTP96 to OTP99	-	Reserved	Reserved
OTP100 to OTP102	-	Reserved	Reserved
OTP103	-	Reserved	Reserved
OTP104 to OTP123	-	Reserved	Reserved
OTP124	-	HCONF1	-
	[0]	IWDG1_HW	IWDG1 start on reset
	[1]	IWDG1_FZ_STOP	IWDG1 freeze in Stop mode
	[2]	IWDG1_FZ_STANDBY	IWDG1 freeze in Standby mode
	[9:3]	Reserved	Reserved
	[10]	RST_STOP	Reset caused if the device is put in Stop mode
	[11]	RST_STDBY	Reset caused if the device is put in Standby mode
	[12]	SELINBORH	- 0: BOR disabled - 1: BOR = 2.7 V
	[13]	HSLV_VDDIO5	VDDIO5 I/O segment below 2.5 V for I/O mode. The I/O segment is used by SDMMC2 port.
	[14]	HSLV_VDDIO4	VDDIO4 I/O segment below 2.5 V for I/O mode (I/O segment used by SDMMC1 port)
	[15]	HSLV_VDDIO3	VDDIO3 I/O segment below 2.5 V for I/O mode (I/O segment used by XSPIM port 2)

Table 18. OTP fuse description (lower OTP region) (continued)

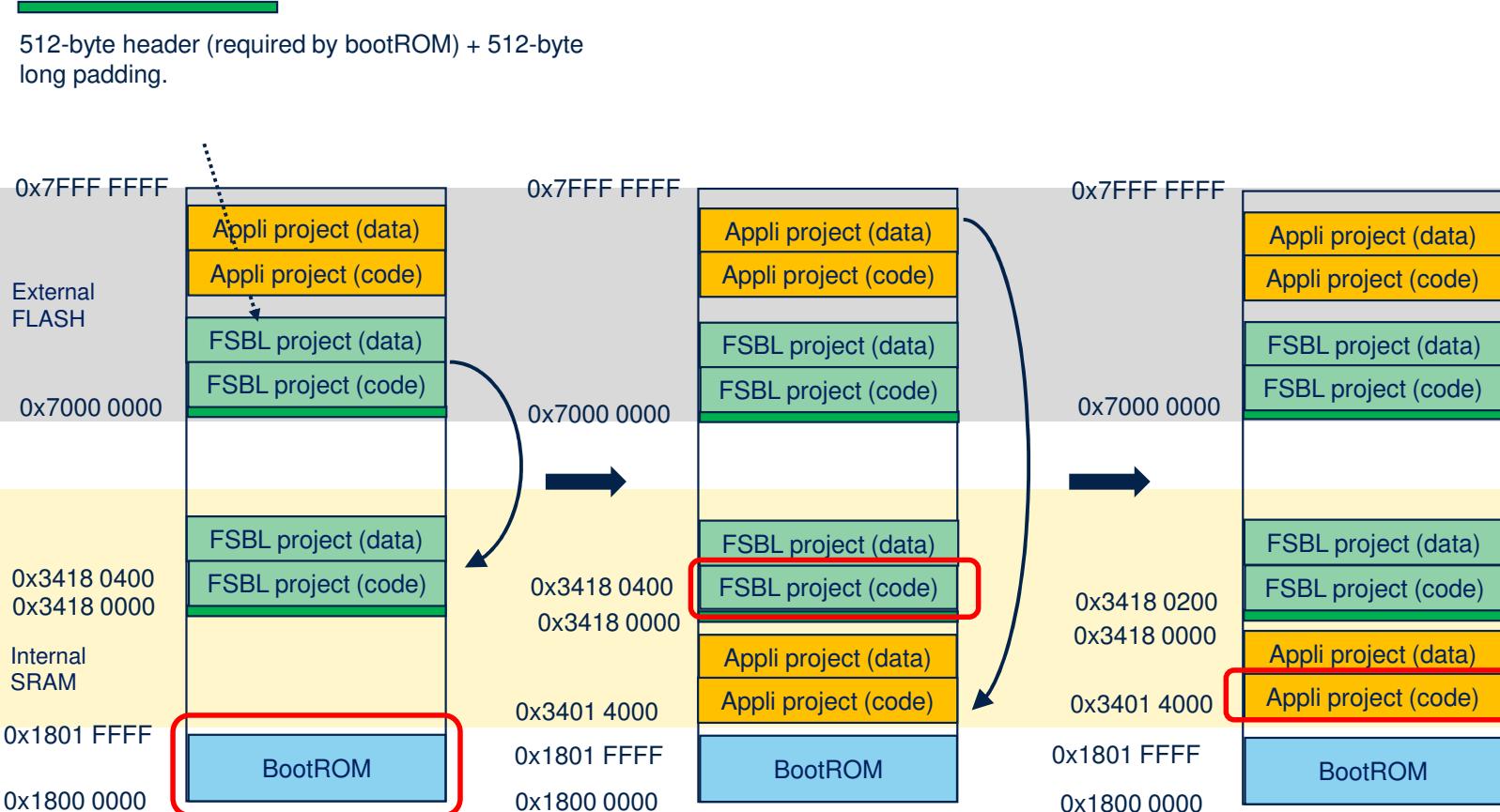
OTP word	OTP bits	Name or description	Detailed description
	[2]	sdmmc1_not_default_af	- 0 (no): SDMMC1 uses default hard coded AFmux - 1 (yes): SDMMC1 uses AFmux defined in OTP
	[3]	sdmmc2_not_default_af	- 0 (no): SDMMC2 uses default hard coded AFmux - 1 (yes): SDMMC2 uses AFmux defined in OTP
	[4]	Reserved	Reserved
	[8:5]	flash_boot_source	If different from 0, identifies the flash memory used to boot: - 1 (sdcard): SD-Card SDMMC1 - 2 (emmc): e.MMC SDMMC1 - 3 (snor): XSPI NOR - 4 (snand): XSPI NAND - 5 (hflash): XSPI HyperFlash - 6 (pnand): FMC pNAND - 7 (sdcard): SD-Card SDMMC2 - 8 (emmc): e.MMC SDMMC2 - others: invalid



# Bootrom overview

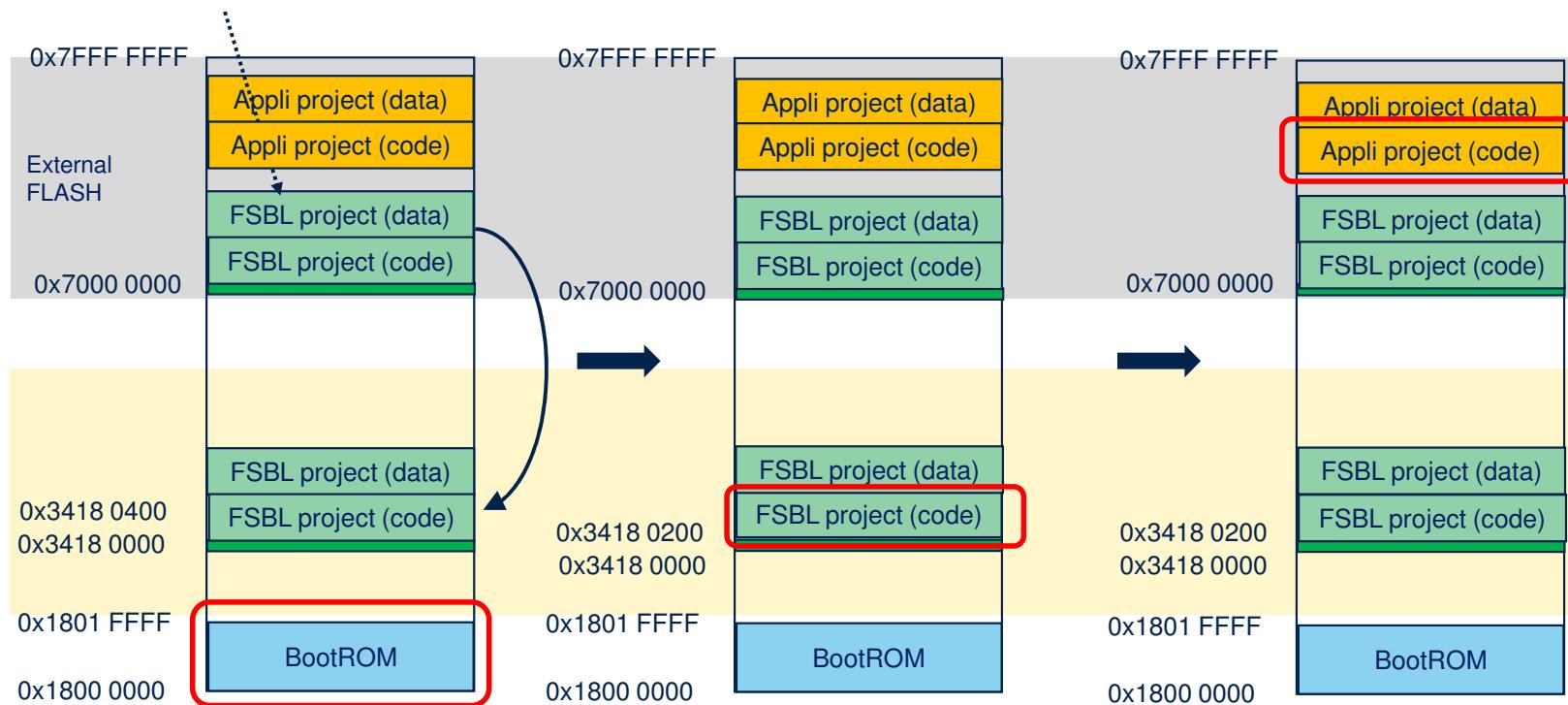


## Boot from external Flash. FSBL + Load & Run



# Boot from external Flash FSBL + XiP (Execute in Place)

512-byte header (required by bootROM) + 512-byte long padding.



# **NUCLEO-N657Z0-Q**



# STM32N6 Nucleo-144 boards



## Common features

- STM32 microcontroller in an LQFP144, TFBGA225, or VFBGA264 package
- 3 user LEDs
- 1 user push-button and 1 reset push-button
- 32.768 kHz crystal oscillator
- Board connectors:
  - SWD
  - ST morpho expansion connector
- Flexible power-supply options: ST-LINK USB VBUS, USB connector, or external sources
- Comprehensive free software libraries and examples available with the STM32Cube MCU Package
- Support of a wide choice of Integrated Development Environments (IDEs) including IAR Embedded Workbench®, MDK-ARM, and STM32CubeIDE

## Features specific to some of the boards

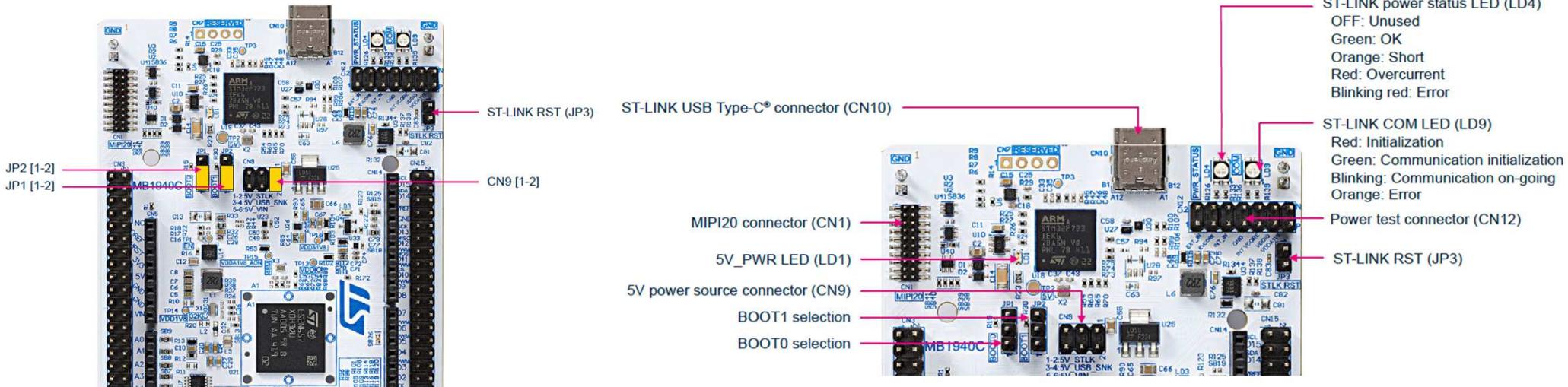
(refer to the ordering information section of the data brief for details)

- External or internal SMPS to generate  $V_{core}$  logic supply
- Ethernet compliant with IEEE-802.3-2002
- USB Device only, USB OTG full speed, or SNK/UFP (full-speed or high-speed mode)
- Board connectors:
  - ARDUINO® Uno V3 connector or ST Zio expansion connector including ARDUINO® Uno V3
  - Camera module FPC
  - MIPI20 compatible connector with trace signals
  - USB with Micro-AB or USB Type-C®
  - Ethernet RJ45
- On-board ST-LINK (STLINK/V2-1, STLINK-V3E, or STLINK-V3EC) debugger/programmer with USB re-enumeration capability: mass storage, Virtual COM port, and debug port

[https://www.st.com/resource/en/user\\_manual/um3417-stm32n6-nucleo144-board-mb1940-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um3417-stm32n6-nucleo144-board-mb1940-stmicroelectronics.pdf)

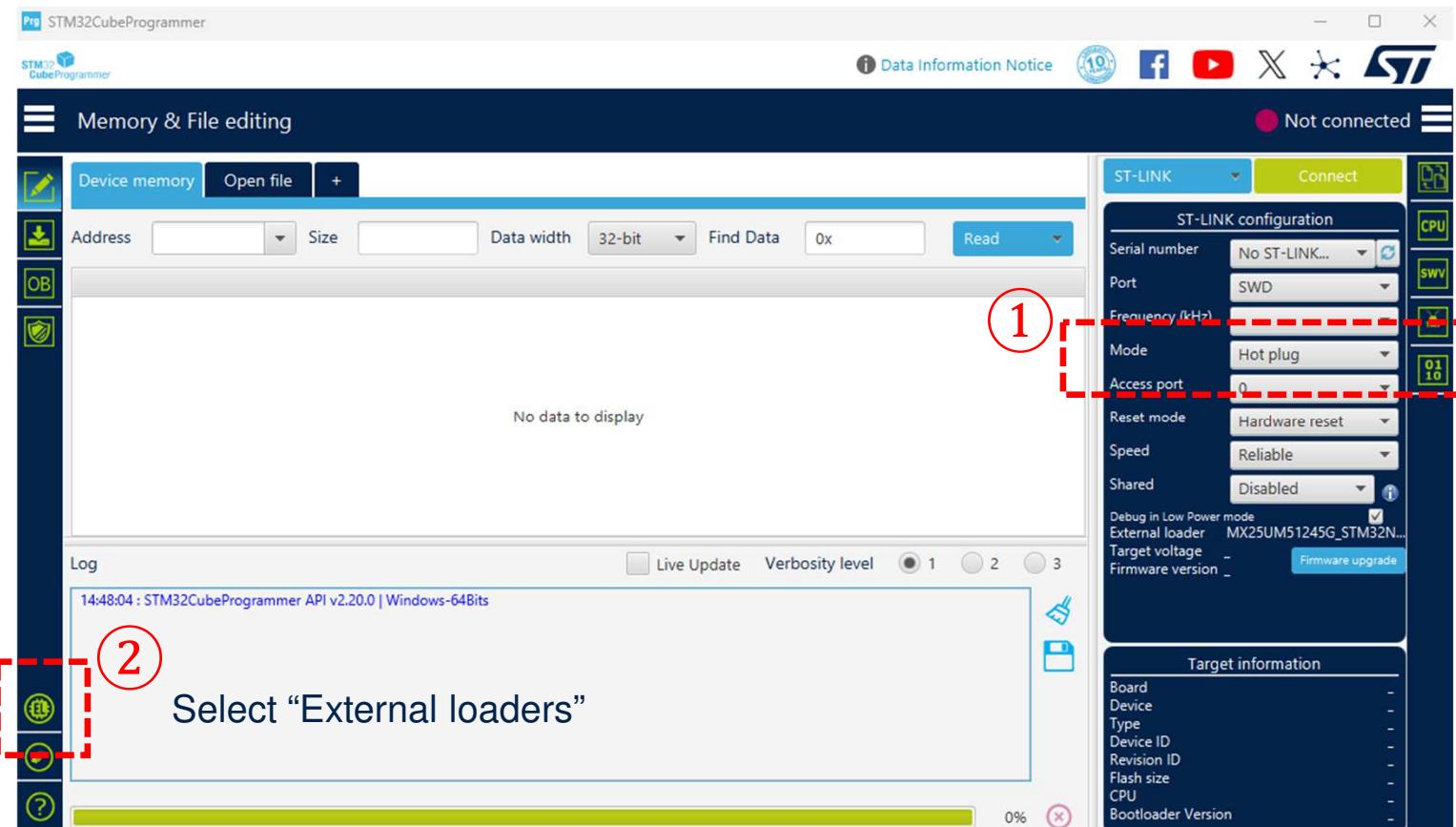


# Check Hardware Board



Jumper	Definition	Default position	Comment
CN9	5V power selection (user USB power source selection)	[1-2]	5V from STLINK-V3EC
JP1	BOOT0 selection	[1-2]	Boot pin flash/serial selection
JP2	BOOT1 selection	[1-2]	Boot in flash mode
JP3	STLK_RST	OFF	-

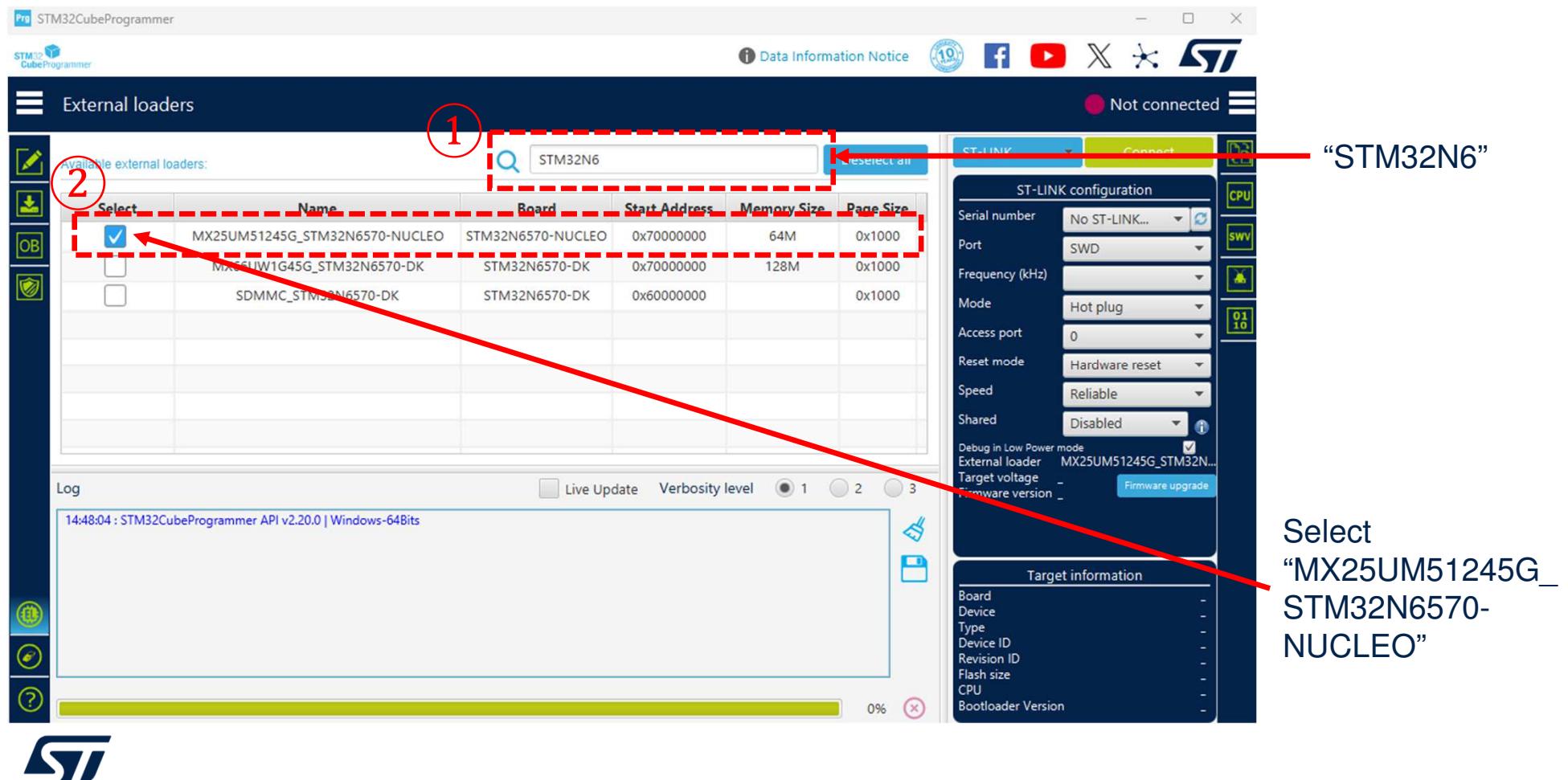
# Check Hardware Board



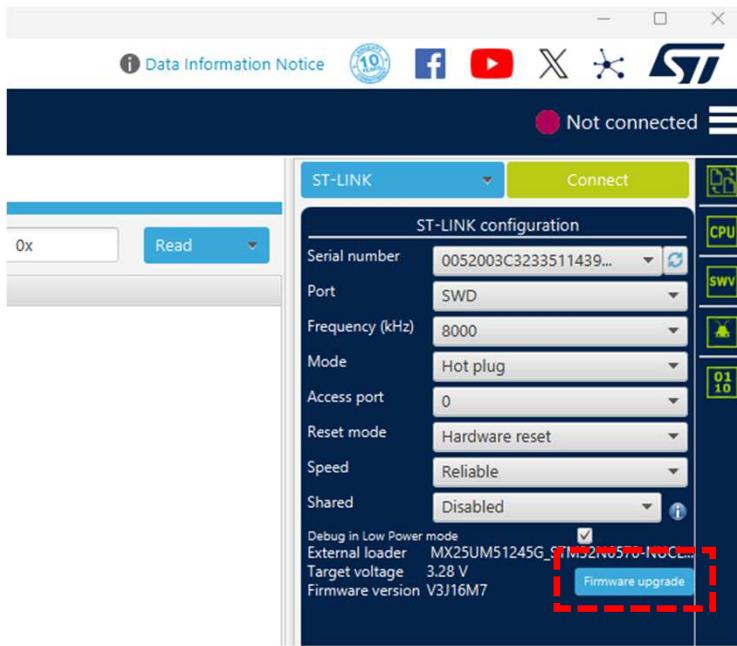
Select “Hot plug”

Select “External loaders”

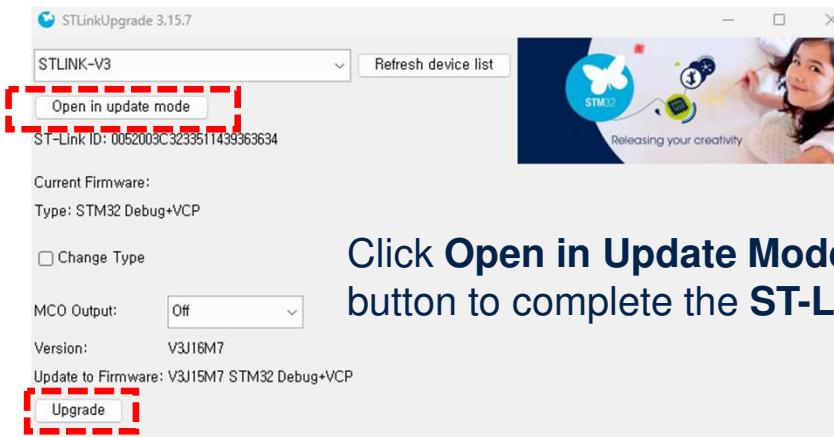
# Check Hardware Board



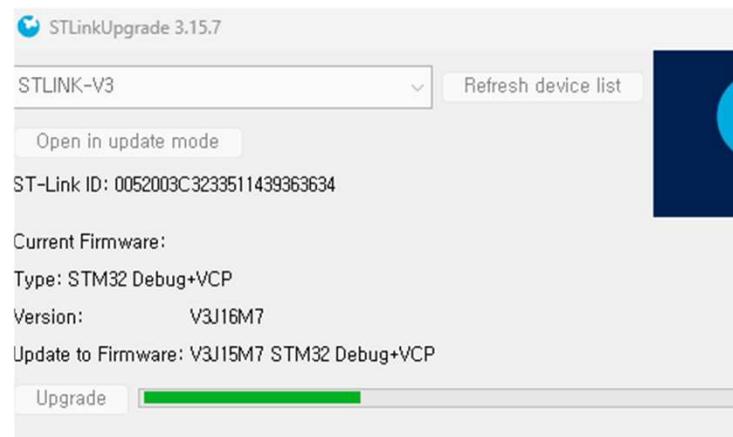
# Check Hardware Board



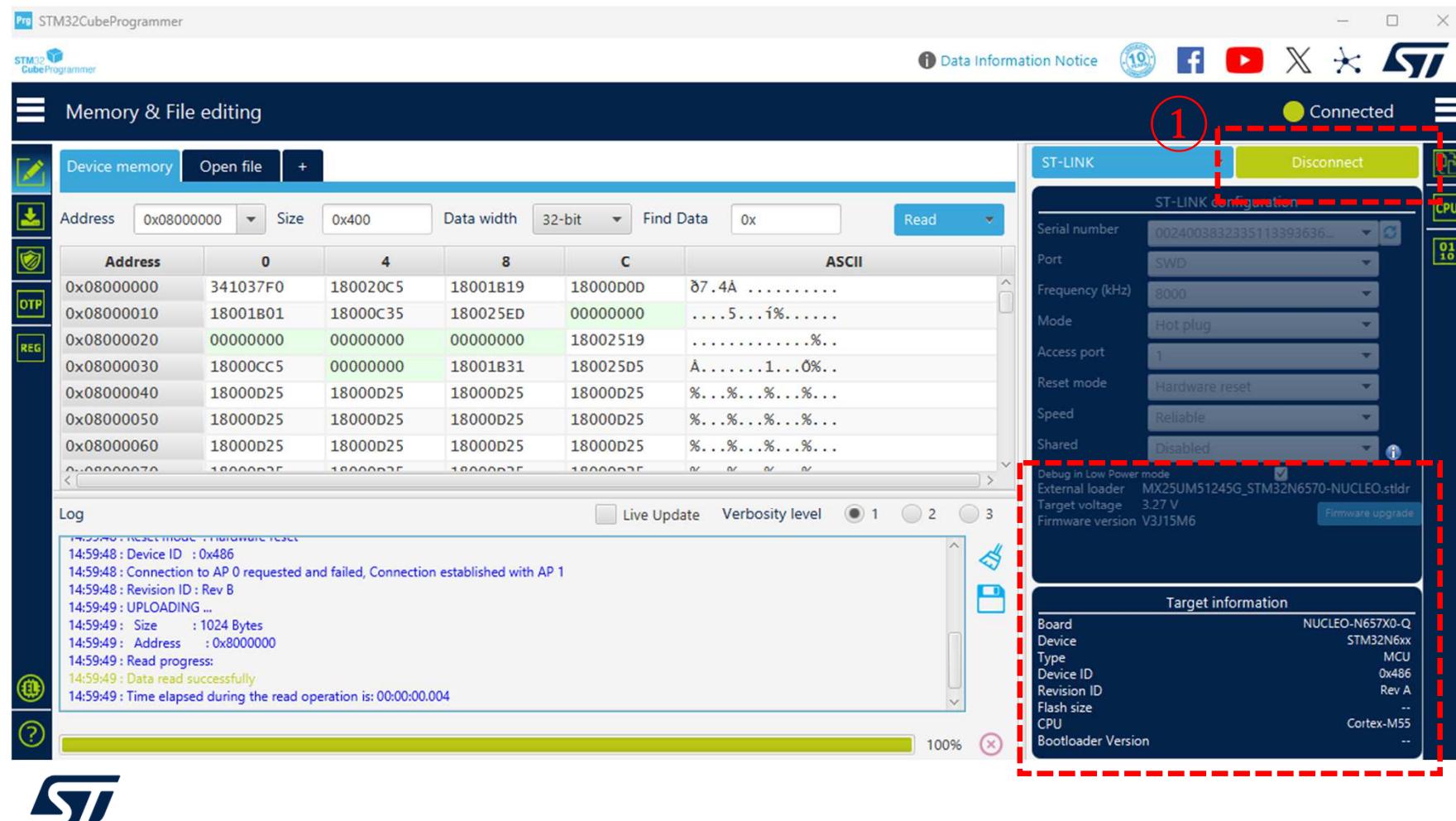
Click the **Firmware Upgrade** button.



Click **Open in Update Mode**, then click the **Upgrade** button to complete the **ST-Link firmware update**.



# Check Hardware Board



# Document URL & References

STM32N657X0 Product overview

<https://www.st.com/en/microcontrollers-microprocessors/stm32n657x0.html#overview>

STM32N6x5xx STM32N6x7xx Datasheet :

<https://www.st.com/resource/en/datasheet/stm32n657a0.pdf>

STM32N6x5xx STM32N6x7xx Reference manual :

[https://www.st.com/resource/en/reference\\_manual/rm0486-stm32n647657xx-armbased-32bit-mcus-stmicroelectronics.pdf](https://www.st.com/resource/en/reference_manual/rm0486-stm32n647657xx-armbased-32bit-mcus-stmicroelectronics.pdf)

STM32N6x5xx STM32N6x7xx boot ROM manual:

[https://www.st.com/resource/en/user\\_manual/um3234-how-to-proceed-with-boot-rom-on-stm32n6-mcus-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um3234-how-to-proceed-with-boot-rom-on-stm32n6-mcus-stmicroelectronics.pdf)

NUCLEO-N657X0-Q Product overview:

<https://www.st.com/en/evaluation-tools/nucleo-n657x0-q.html>

STM32N6 Nucleo-144 board User manual:

[https://www.st.com/resource/en/user\\_manual/um3417-stm32n6-nucleo144-board-mb1940-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um3417-stm32n6-nucleo144-board-mb1940-stmicroelectronics.pdf)

NUCLEO-N657X0-Q Board schematic:

[https://www.st.com/resource/en/schematic\\_pack/mb1940-n657x0q-c02-schematic.pdf](https://www.st.com/resource/en/schematic_pack/mb1940-n657x0q-c02-schematic.pdf)



# Installation & Download Guide (1/3)

STM32CubeIDE (This Hands-on uses version 1.19.0)

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

STM32CubeMX (This Hands-on uses version 6.15.0)

<https://www.st.com/en/development-tools/stm32cubemx.html>

STM32CubeProgrammer (This Hands-on uses version 2.20.0)

<https://www.st.com/en/development-tools/stm32cubeprog.html>

STEdgeAI-NPU (This Hands-on uses version 2.2.0)

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

X-CUBE-AI (This Hands-on uses version 10.2.0)

<https://www.st.com/en/embedded-software/x-cube-ai.html#get-software>

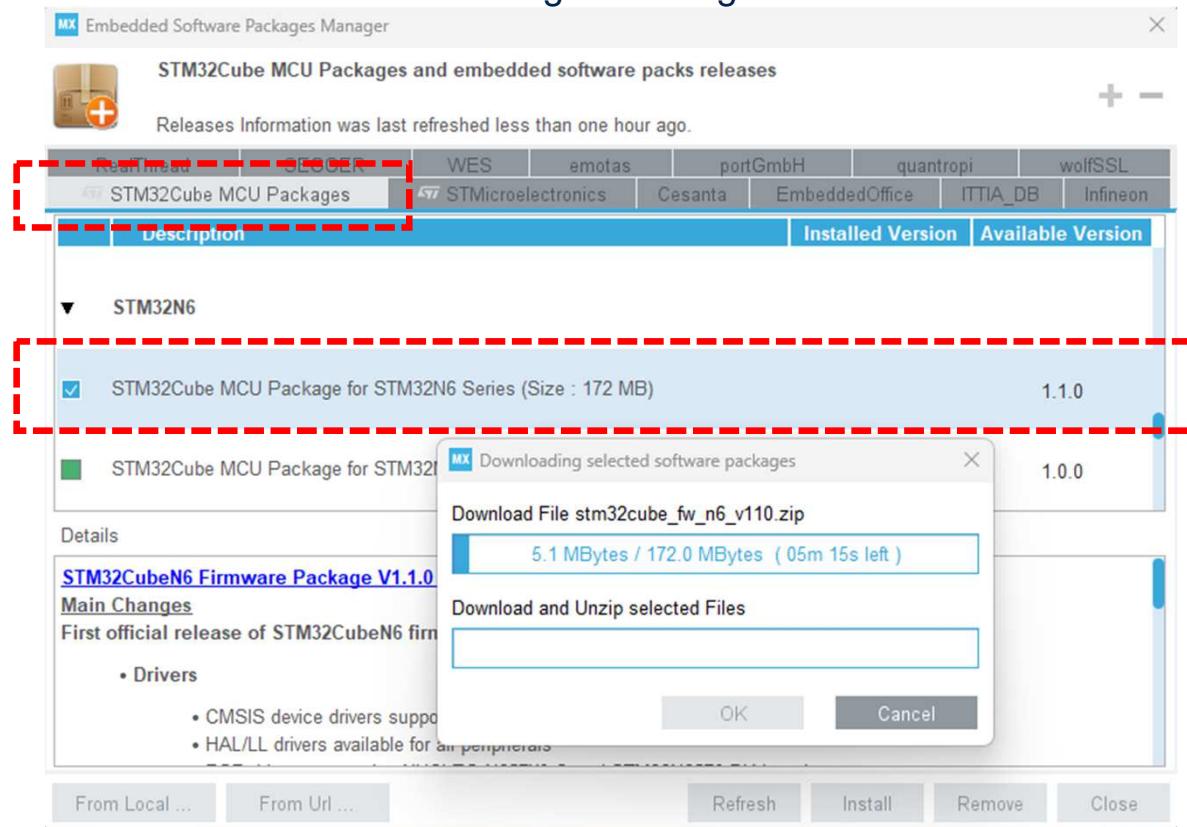
Hands On and Example code

[https://github.com/BlaineMoon30/STM32\\_Expert\\_Training\\_AI\\_STM32N6](https://github.com/BlaineMoon30/STM32_Expert_Training_AI_STM32N6)



# Installation & Download Guide (2/3)

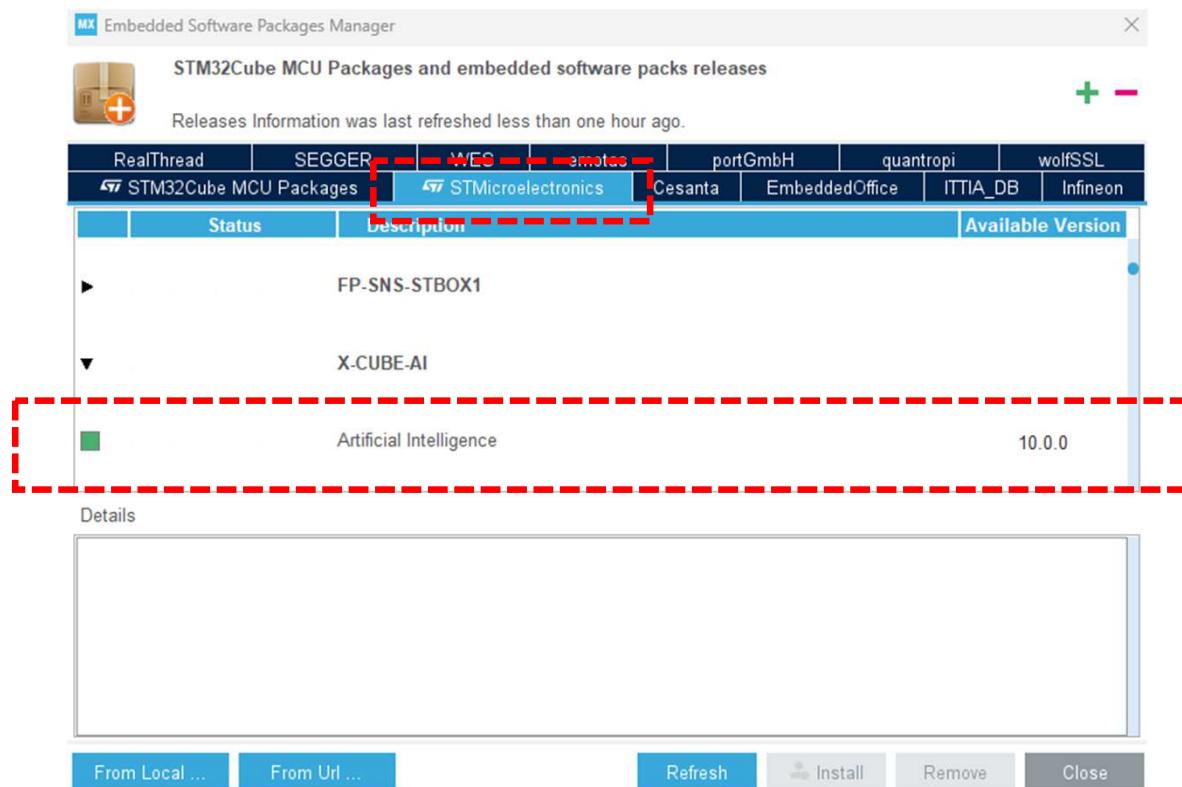
This hands-on uses the STM32CubeN6 Firmware Package V1.1.0.  
Install it from CubeMX's Embedded Software Packages Manager.



# Installation & Download Guide (3/3)

This hands-on uses X-CUBE-AI V10.0.0.

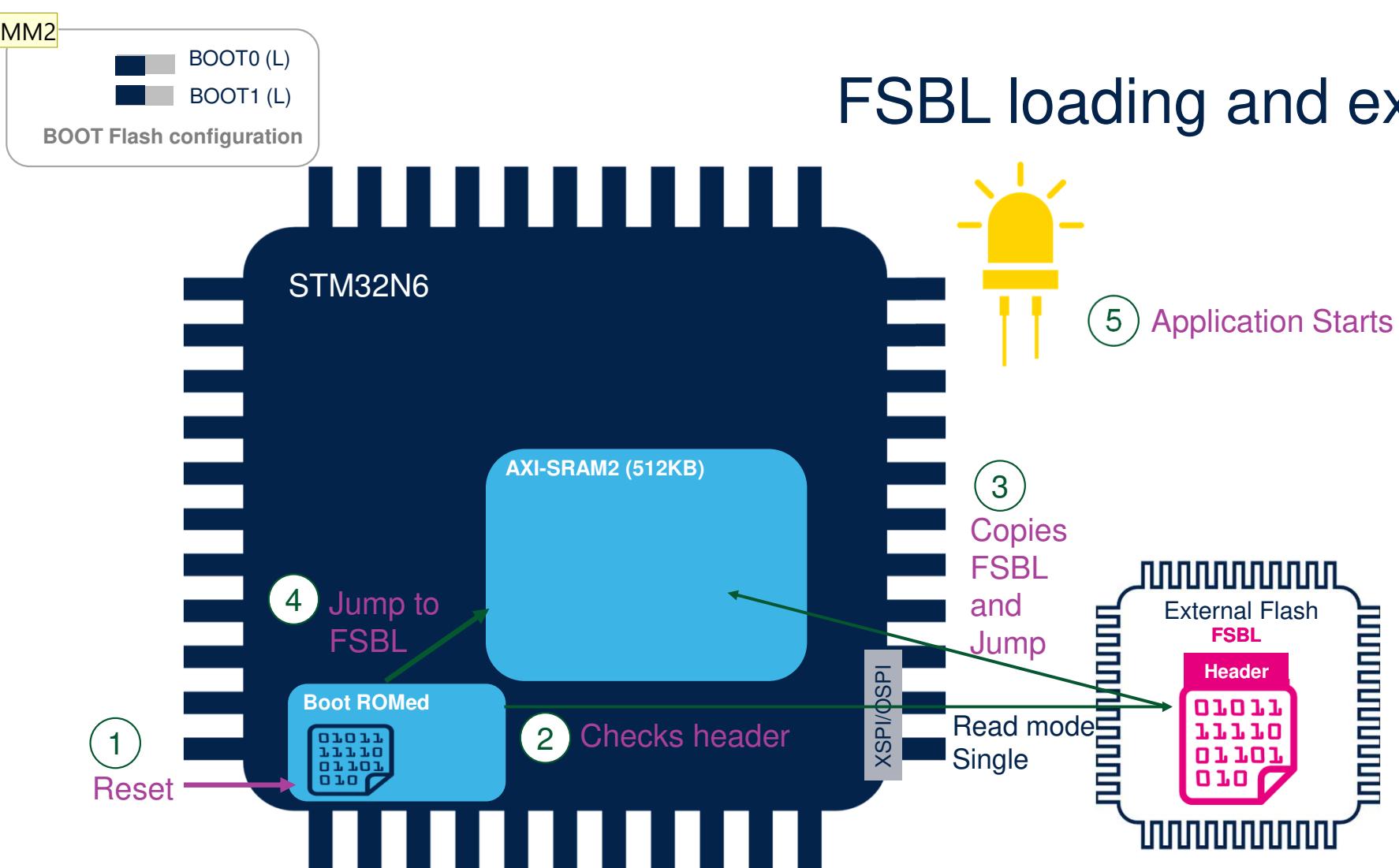
Install it from CubeMX's Embedded Software Packages Manager.



# **Hands-On: FSBL + LRUN**



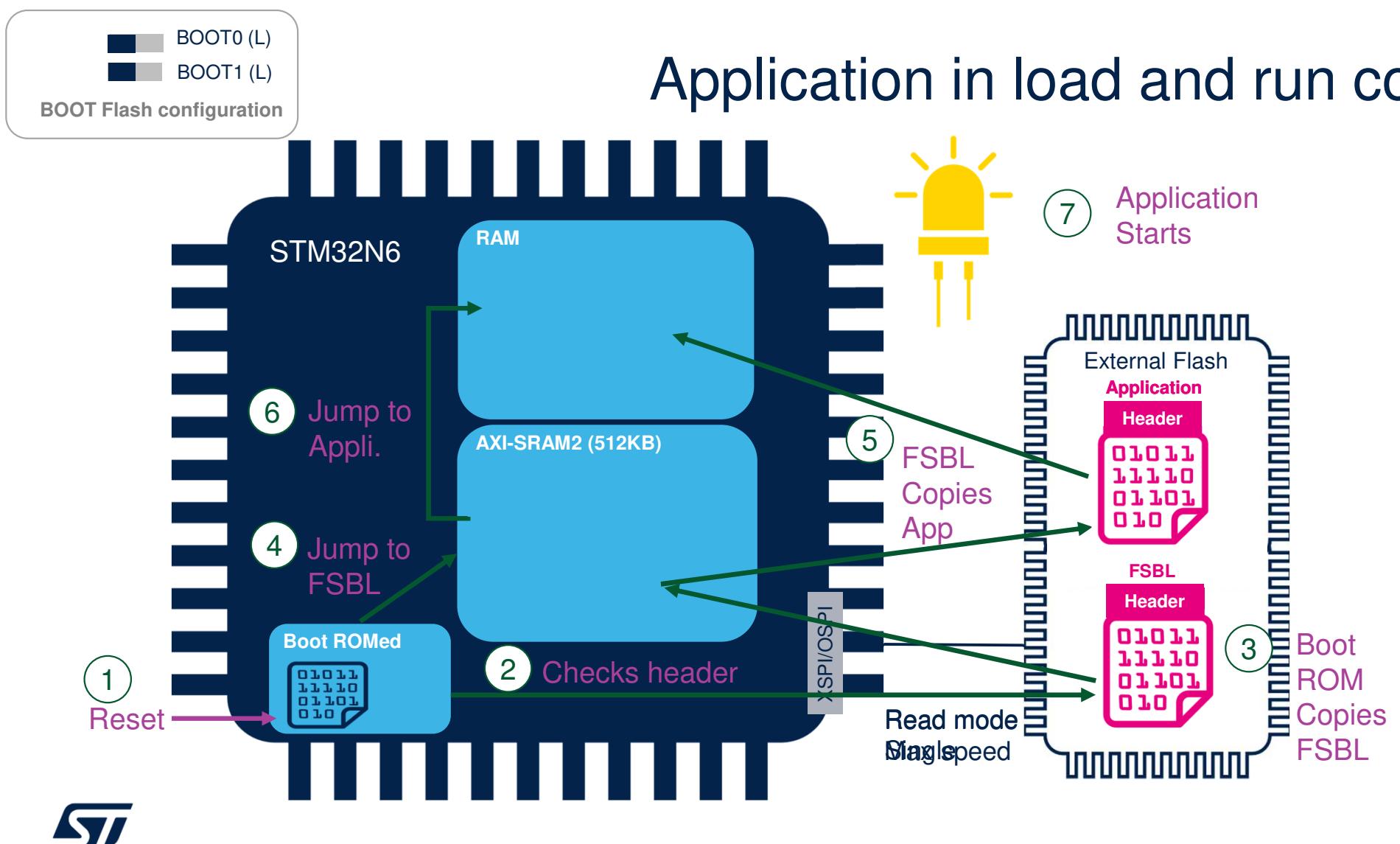
# FSBL loading and execution



**MM1** Remark: BootROM can be called verbally zero stage bootloader to simplify the concept for customers  
Manuel MARCIAS, 2025-03-06T15:45:05.682

**MM2** focus on explaining what is given by ST and what is user defined  
Manuel MARCIAS, 2025-03-06T15:50:23.255

# Application in load and run config



# CubeMX Configuration

The image shows the STM32CubeMX software interface for creating a new project. On the left, a 'New Project' wizard lists three options: 'Start My project from MCU', 'Start My project from ST Board', and 'Start My project from Example'. The first option is highlighted with a red circle labeled '1' and a red dashed box around the 'ACCESS TO MCU SELECTOR' button. A large red arrow points from this button to the 'MCU/MPU Selector' interface on the right. In the selector, a dropdown menu under 'Commercial Part Number' is highlighted with a red circle labeled '2' and a red dashed box. The selected part number is 'STM32N657X0H3Q'. Below the dropdown is a table titled 'MCUs/MPUs List: 1 item' showing the selected part. A red circle labeled '3' and a red dashed box highlight the row for 'STM32N657X0H...'. An arrow points from this row to the text 'Double Click' at the bottom right.

New Project

I need to :

Start My project from MCU  
1 ACCESS TO MCU SELECTOR

Start My project from ST Board  
ACCESS TO BOARD SELECTOR

Start My project from Example  
ACCESS TO EXAMPLE SELECTOR

STM32N657X0H3Q

MCU/MPU Selector   Board Selector   Example Selector   Cross Selector

MCU/MPU Filters

Commercial Part Number: STM32N657X0H3Q

PRODUCT INFO

Segment, Series, Line, Marketing Status, Price, Package, Core, Coprocessor

MEMORY

Flash = 0 (kBytes)

MCUs/MPUs List: 1 item

Commercial Part No.	Part No.	Reference	Marketing St...	Unit Price for
STM32N657X0H...	STM32N657X0	STM32N657X0...	Active	10.7254

Features   Block Diagram   Docs & Resources   CAD

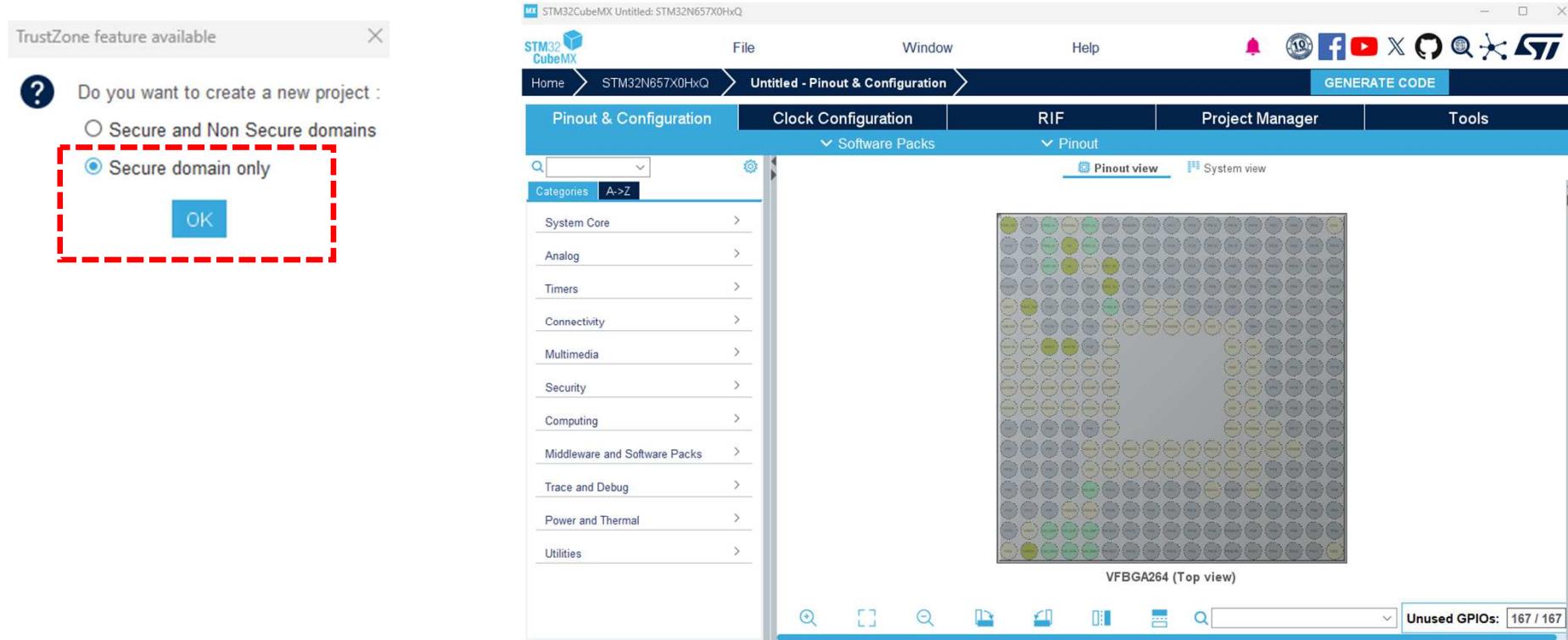
The first high-performance STM32 MCU with AI capabilities

Double Click



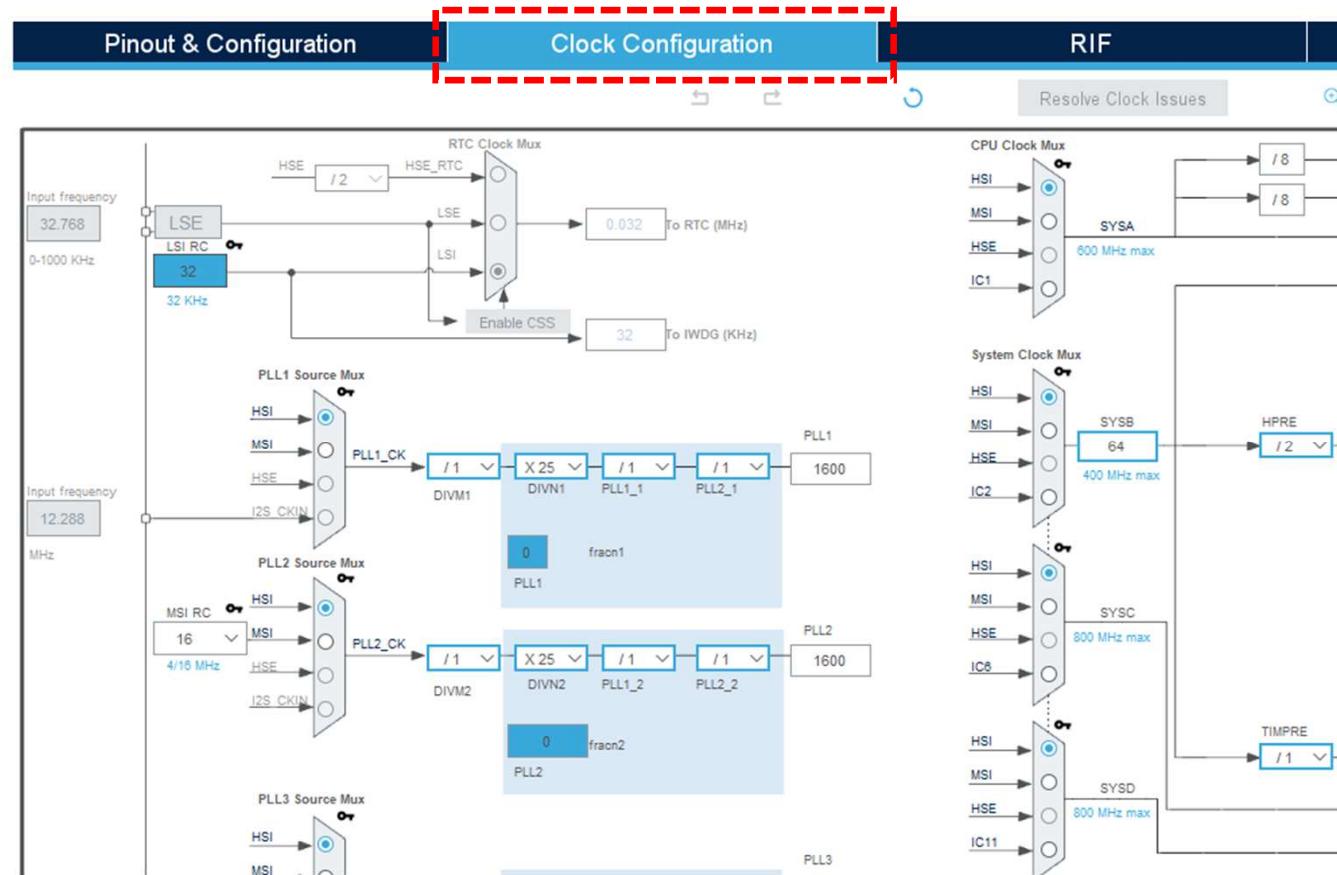
# CubeMX Configuration

In this hands-on, select **Secure domain only**.



# Clock Configuration

Click the **Clock Configuration** tab.



## Electrical characteristics

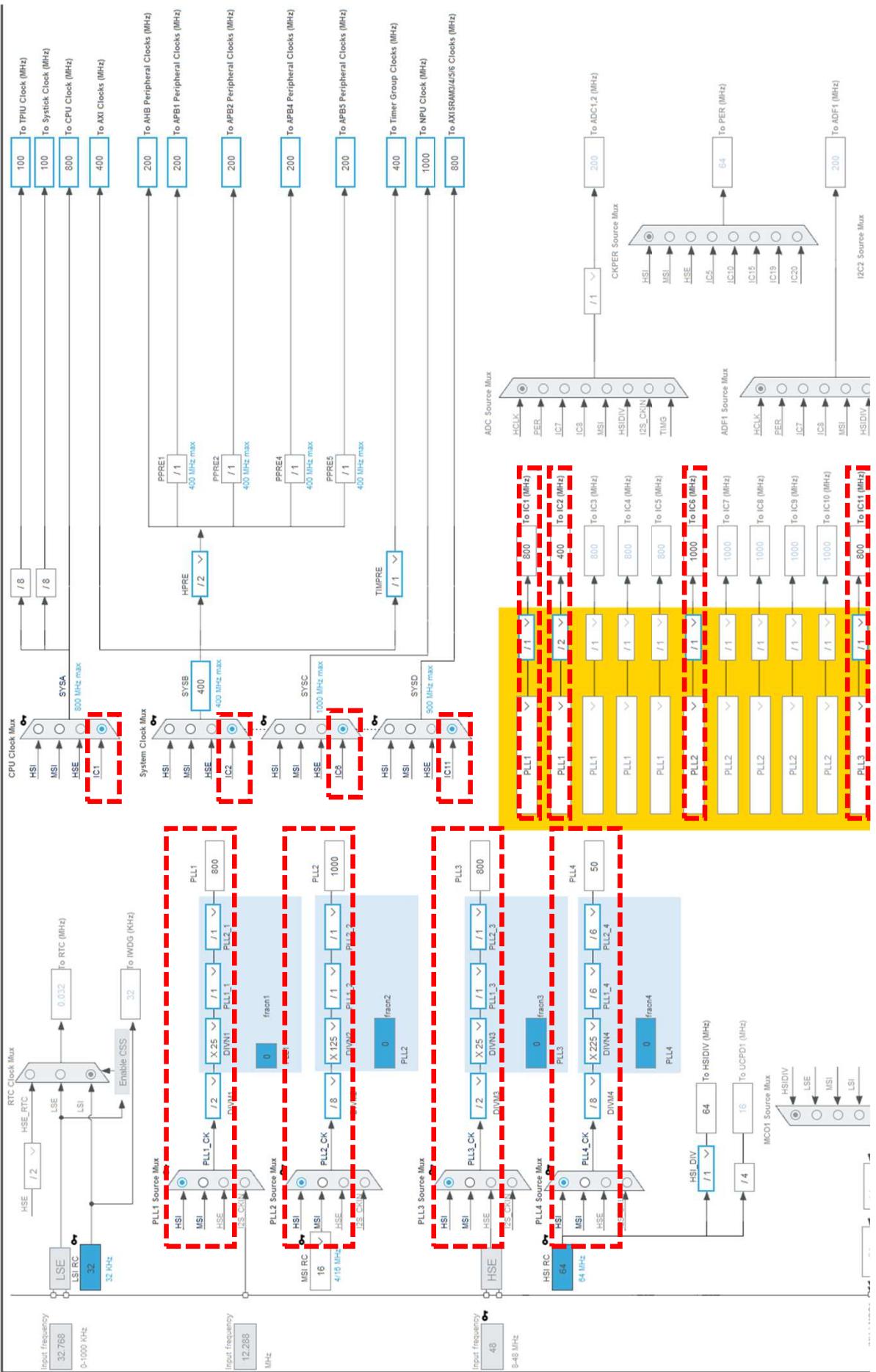
STM32N647xx STM32N657xx

### 5.3 Operating conditions

#### 5.3.1 General operating conditions

Table 22. General operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$F_{CPU}$	Clock frequency of Cortex-CM55 (VOS low)	Base-TCM with 0 wait states Flex-TCM with 1 wait state	0	-	600	MHz
$F_{CPU\ overdrive}$	Clock frequency of Cortex-CM55 in overdrive (VOS high)		0	-	800	
$F_{NPU}$	Clock frequency of NPU/CNN		0	-	800	
$F_{NPU\ overdrive}$	Clock frequency of NPU/CNN in overdrive		0	-	1000	
$F_{ck\_icn\_hsl}$	Clock frequency of USB, ETH buses	-	0	-	400	
$F_{HCLK}$	Clock frequency of AHB bus	-	0	-	200	
$F_{ck\_cpu\_axi}$	Clock frequency of AXI CPU bus	-	0	-	400	
$F_{PCLKx}$	Clock frequency of APB buses ( $x = 1, 2, 3, 4, 5$ )	-	0	-	$F_{HCLKx} / 4$	



# CORTEX\_M55\_FSBL Configuration

The screenshot shows two overlapping software interface windows. The left window, titled 'Pinout & Configuration', displays a table of system components and their configuration status across three modes: FSBL, Application, and ExtMemLoader. Components listed include CORTEX\_M55\_FSBL (selected), CORTEX\_M55\_S, GPDMA1, GPIO, HPDMA1, ICACHE, IWWDG, NVIC1\_S\_Application, NVIC\_FSBL, RAMCFG, RCC, SYS\_S, and WWDG. The CORTEX\_M55\_FSBL row is highlighted with a red dashed box. The right window, titled 'Clock Configuration', shows runtime contexts for the selected mode (First Stage Boot Loader) and configuration parameters for Cortex Interface Settings (CPU ICache, CPU DCache) and Cortex Memory Protection Unit Control Settings (MPU Control Mode). The 'CPU DCache' setting is also highlighted with a red dashed box.

Pinout & Configuration

Clock Configuration

RIF

Software Packs

Pinout

CORTEX\_M55\_FSBL Mode and Configuration

Mode

Runtime contexts:

First Stage Boot Loader	Application	External Memory Loader
<input checked="" type="checkbox"/>		

Configuration

Reset Configuration

Parameter Settings

User Constants

Configure the below parameters :

Search (Ctrl+F)

Cortex Interface Settings

CPU ICache	Enabled
CPU DCache	Enabled

Cortex Memory Protection Unit Control Settings

MPU Control Mode	MPU NOT USED
------------------	--------------



# RCC Configuration

The screenshot shows three tabs: Pinout & Configuration, Clock Configuration, and RIF.

**Pinout & Configuration:** This tab displays a hierarchical tree of system components. A red dashed box highlights the 'RCC' row under the 'System Core' section, which contains checkboxes for FSBL, Application, and ExtMemLoader.

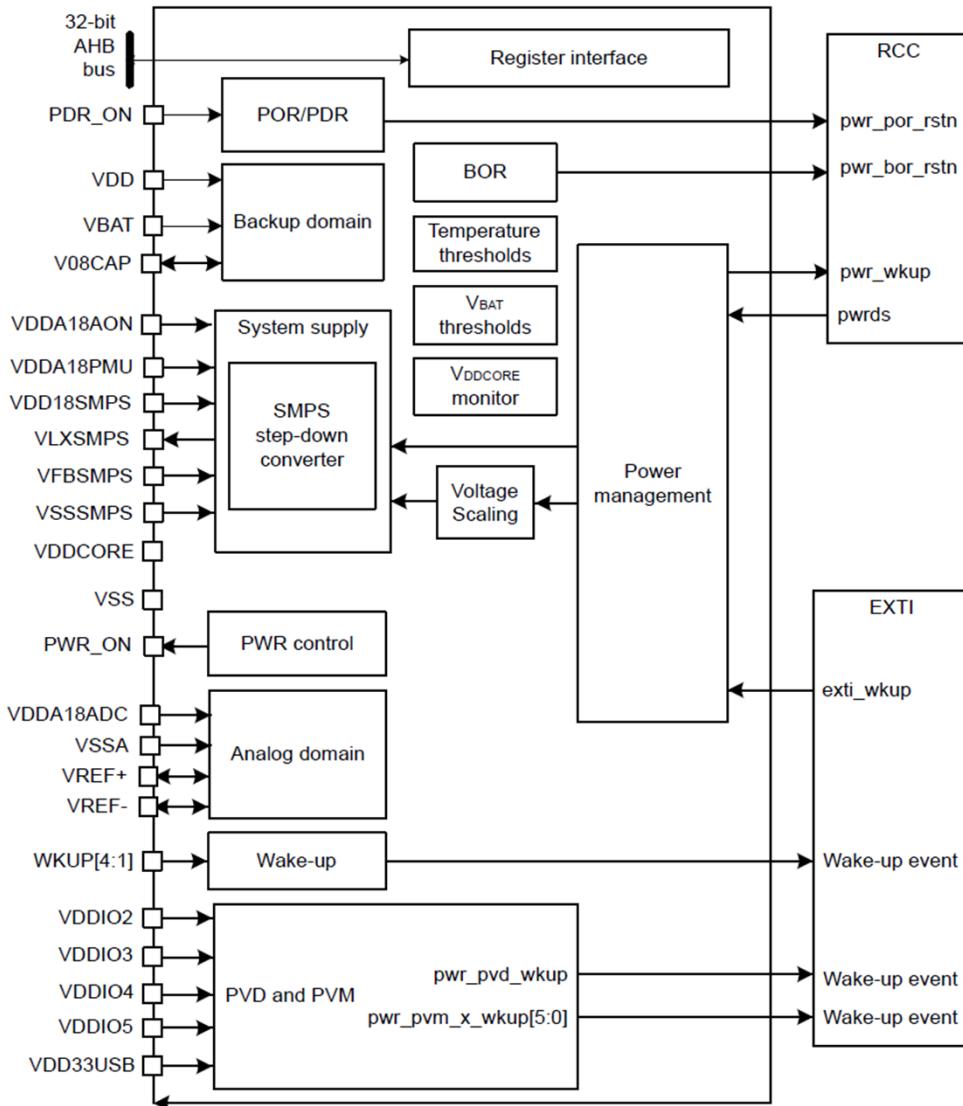
	FSBL	Application	ExtMemLoader
CORTEX_M55_FSBL	<input checked="" type="checkbox"/>		
CORTEX_M55_S		<input checked="" type="checkbox"/>	
GPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
GPIO			
HPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICACHE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NVIC1_S_Application		<input checked="" type="checkbox"/>	
NVIC_FSBL	<input checked="" type="checkbox"/>		
RAMFC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>RCC</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SYS_S	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
WWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Clock Configuration:** This tab shows RCC Mode and Configuration settings. A red dashed box highlights the 'Power Parameters' section, which includes fields for SupplySource and Power Regulator Voltage Scale.

Mode	First Stage Boot Loader	Application	External Memory Loader
High Speed Clock (HSE)	Disable		
Low Speed Clock (LSE)	Disable		
Master Clock Output 1			
Master Clock Output 2			
Audio Clock Input (I2S_CKIN)			

**RIF:** This tab is partially visible at the top right.

**Figure 9. Power control block diagram**



The boot ROM code uses multiple supplies for I/Os:

**VDDIO3:** Independent I/O supply 3 (PN[12:0]), configured for XSPI1M\_P2 (XSPI).

**VDDIO4:** Independent I/O supply 4 (PC[1], PC[12:6], and PH[2,9]), configured for eMMC.

**VDDIO5:** Independent I/O supply 5 (PC[0], PC[5:2], and PE[4]), configured for SD card.

**VDD33USB:** Supply input for USB HS PHYs and USB Type-C® PHY 3V3.

These supplies can be independent of VDD and can be monitored with peripheral voltage monitoring. The voltage range configuration is described in the PWR section of the reference manual (RM0486).

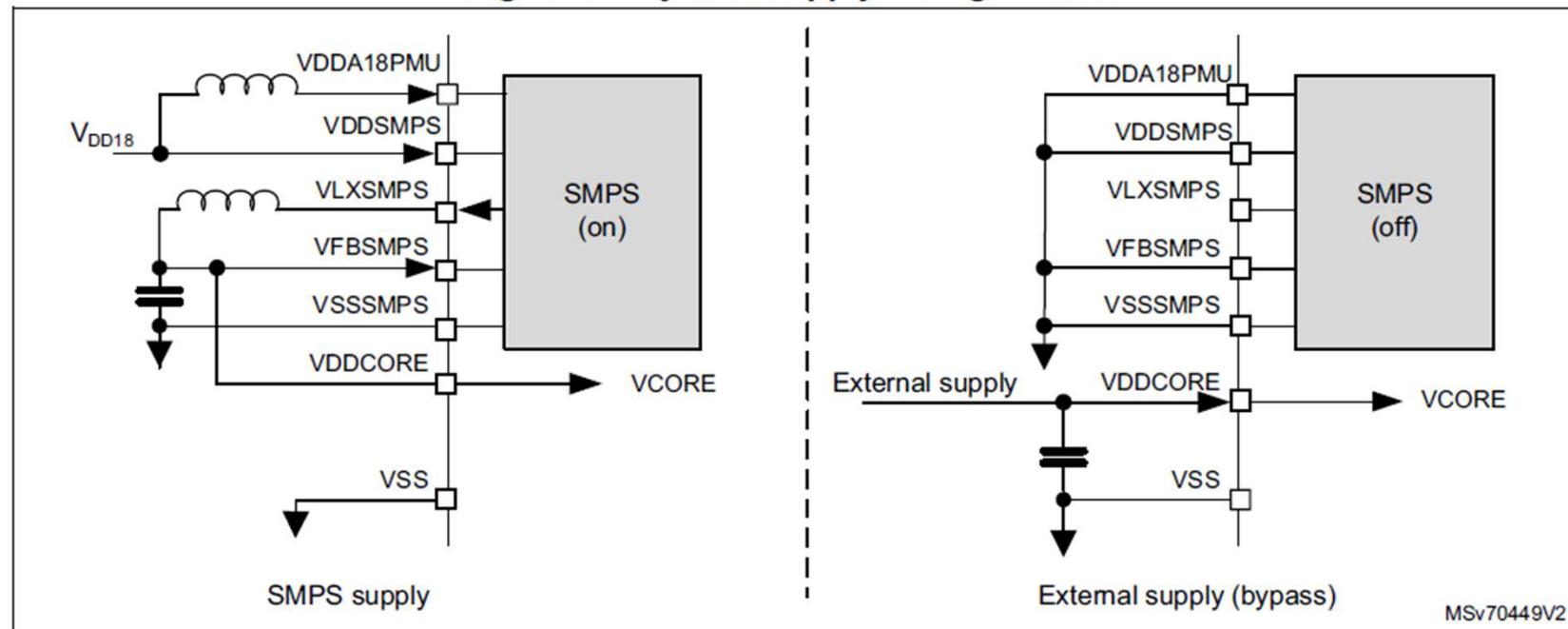
# Power supply

RM0486

Power control (PWR)

By configuring the SMPS step-down converter, the supply configurations shown in [Figure 16](#) are supported for the V<sub>CORE</sub> domain.

**Figure 16. System supply configurations**



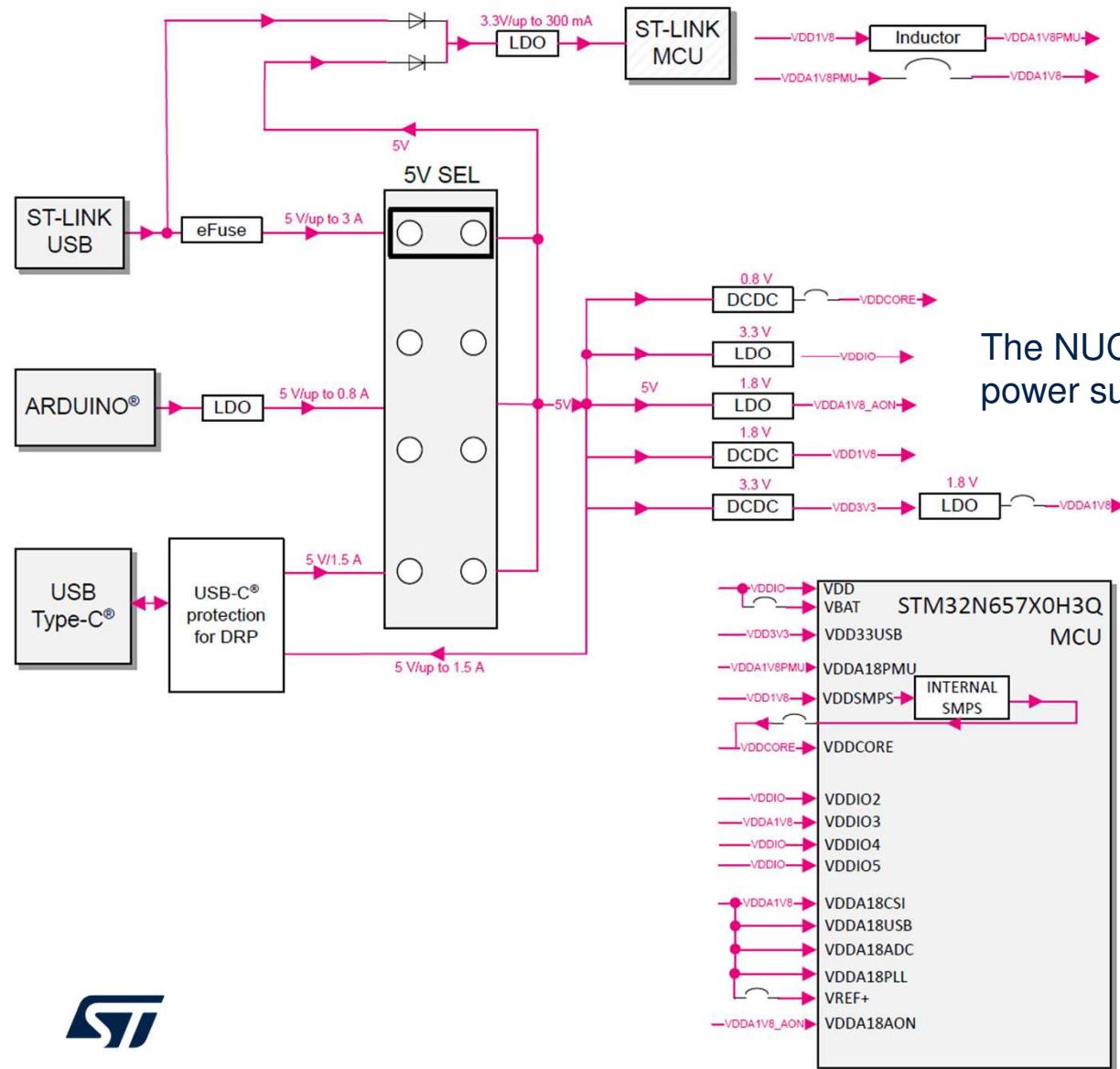
MSv70449V2

# Power supply

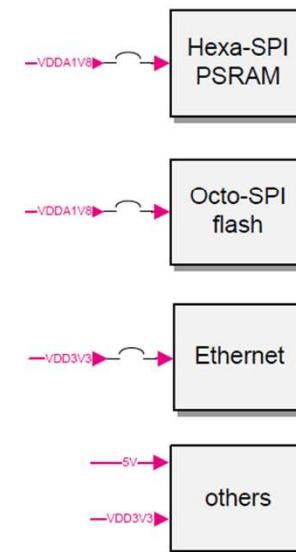
$V_{DD}^{(1)}$	I/Os supply voltage	1.8 V range	1.62	1.8	1.98	V
		3.3 V range	3.0	3.3	3.6	
$V_{DDA18ON}^{(1)}$	Internal analog supply voltage	-	1.71	1.8	1.935	
$V_{DDIOx}$	Specific I/Os supply voltage (x = 2, 3, 4, 5)	1.8 V range	1.71	1.8	1.935	
		3.3 V range	2.7	3.3	3.6	
$V_{DDCORE}$	Main digital logic supply voltage	SoC Run mode (VOS low)	0.782	0.81	0.842	
		SoC Run mode (VOS high)	0.858	0.89	0.921	
		Sleep mode (SoC Run mode, peripheral clock stopped, VOS low)	0.782	0.81	0.842	
		Sleep mode (SoC Run mode, peripheral clock stopped, VOS high)	0.858	0.89	0.921	
		Stop mode (SVOS low)	0.64	0.68	0.71	
		Stop mode (SVOS high)	0.782	0.81	0.842	
$V_{DDA18PLL}$	1.8 V analog supply for PLL	$F_{NPU}$ range	1.62	1.8	1.98	
$V_{DDA18CSI}$	1.8 V analog supply for CSI	$F_{NPU}$ overdrive range	1.746	1.8	1.98	
$V_{DDCSI}$	CSI operating voltage	-	0.784	0.81	0.842	
$V_{DD18USB}$	1.8 V analog supply for USBPHY	-	1.746	1.8	1.935	
$V_{DD18ADCx}$	ADC operating voltage (x = 1, 2)	-	1.62	1.8	1.98	
$V_{REF+}$	ADC reference voltage	-	1.1	-	$V_{DD18ADC}$	
$V_{BAT}$	Backup operating voltage	-	TBD	-	3.6	



# Power supply



The NUCLEO board uses **External DCDC** as the default power supply.

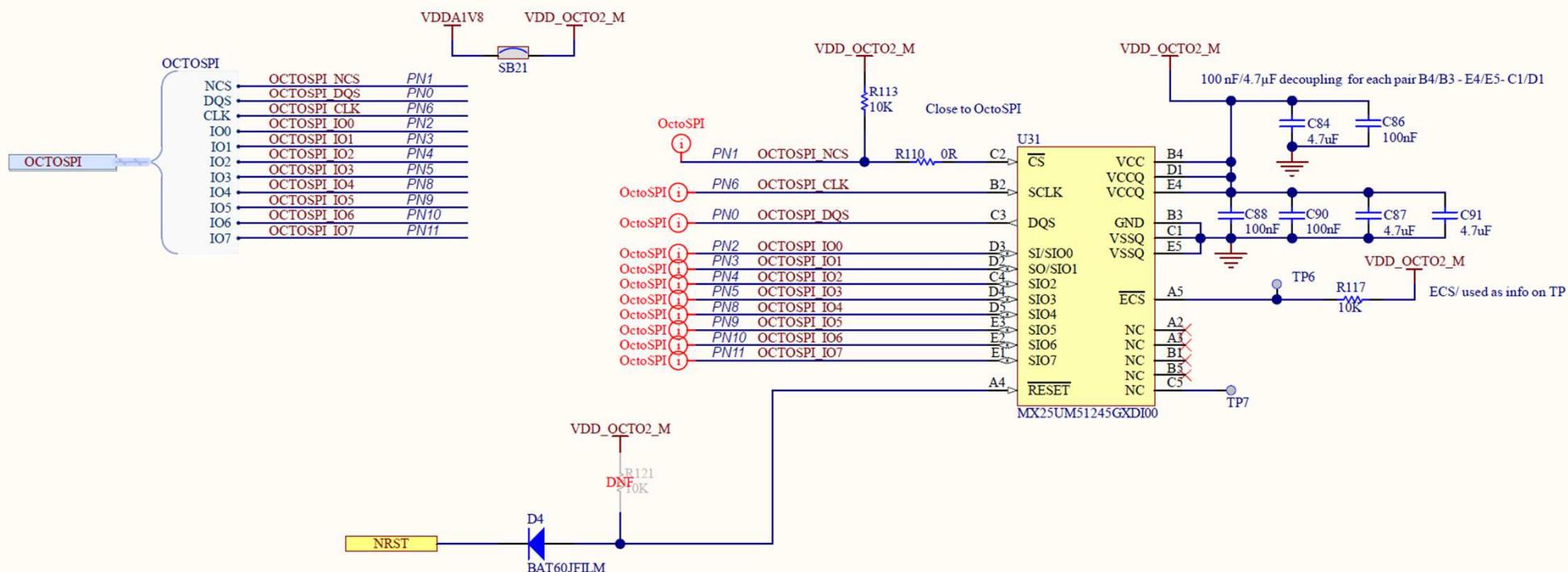


# Octo-SPI Flash Memory Configuration

## Octo-SPI flash memory

The Octo-SPI flash memory has the following characteristics: 512 Mbits, 1.8 V, 200 MHz, DTR, read while writing.

It is connected to the Octo-SPI interface of the STM32N657X0H3Q microcontroller. The embedded footprint is also compatible with many other references in the BGA24 package. Check the compatibility of the memory datasheet versus MB1940 schematics.





# Octo-SPI Flash Memory Configuration

Table 16. Pin description (continued)

Pin number								Pin name (function after reset)	Pin type	I/O structure	Notes	Alternate functions	Additional functions				
VFBGA142	VFBGA169	VFBGA178	VFBGA198	VFBGA223	VFBGA264	H15	J11	L13	M15	P19	R15	PN2	I/O	-	-	XSPIM_P2_IO0(boot), FMC_A23, EVENTOUT	-
						K15	H12	N13	M15	P19	R15	PN3	I/O	-	-	XSPIM_P2_IO1(boot), FMC_A22, EVENTOUT	-
						E14	L12	J13	J14	V19	N17	PN4	I/O	-	-	XSPIM_P2_IO2(boot), EVENTOUT	-
						F15	L13	K14	K13	U18	R16	PN5	I/O	-	-	XSPIM_P2_IO3(boot), EVENTOUT	-
						G15	K12	K12	K15	U19	P15	PN6	I/O	-	-	XSPIM_P2_CLK(boot), EVENTOUT	-
						F14	K11	K13	K14	R16	T16	PN7	I/O	-	-	XSPIM_P2_NCLK(boot), EVENTOUT	-
						E15	M13	J14	J15	V18	P16	PN8	I/O	-	-	XSPIM_P2_IO4(boot), EVENTOUT	-
						G14	K13	L14	L13	T18	T15	PN9	I/O	-	-	XSPIM_P2_IO5(boot), DCMIPP_D5/DCMI_D5/PSSI_D5, EVENTOUT	-
						H14	J12	M14	L15	R18	U15	PN10	I/O	-	-	XSPIM_P2_IO6(boot), LCD_B4, EVENTOUT	-
						J14	H11	N14	M14	P18	U16	PN11	I/O	-	-	XSPIM_P2_IO7(boot), LCD_B6, EVENTOUT	-
						K14	M12	J12	L12	W18	P14	PN12	I/O	-	-	XSPIM_P2_NCS2, EVENTOUT	-

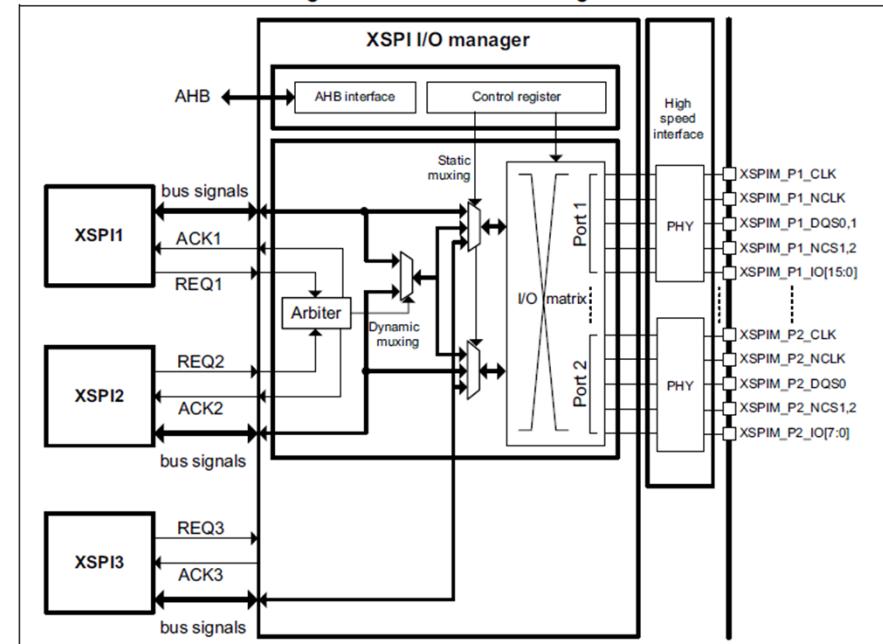
# Octo-SPI Flash Memory Configuration

The screenshot shows the Pinout & Configuration interface. In the left sidebar, under 'Categories' and 'A-Z', 'XSPIM' is selected. A red dashed box highlights the 'XSPIM' row. In the main area, the 'Clock Configuration' tab is active. Under 'Software Packs', the 'XSPIM Mode and Configuration' section is shown. The 'Mode' dropdown is set to 'Direct (XSPI1 to Port1; XSPI2 to Port2; XSPI3 not used)'. A red dashed box highlights this selection. Below it, other options are listed: 'Disable', 'Swapped (XSPI1 to Port2; XSPI2 to Port1; XSPI3 not used)', 'Multiplexed to Port1 (XSPI1 and XSPI2 to Port1; XSPI3 to Port2)', and 'Multiplexed to Port2 (XSPI1 and XSPI2 to Port2; XSPI3 to Port1)'.

Select **FSBL** as the **Runtime Context** for **XSPIM**.

Select **Direct** as the **Port Mode** for **XSPI**.

Figure 215. XSPIM block diagram



**Pinout & Configuration**

**Clock Configuration**

**RIF**

**Pinout**

**XSPI2 Mode and Configuration**

**Mode**

Runtime contexts:

- First Stage Boot Loader
- Application
- External Memory Loader

Mode Octo SPI

Port Port2\_Octo

HyperBus/T/M 1.8V Inverted Clock Disable

Chip Select Override NCS1 - Port2 -

**NVIC Settings**

**Parameter Settings**

**DMA Settings**

**GPIO Settings**

**User Constants**

Configure the below parameters :

Search (Ctrl+F)

Configure the below parameters :

Generic

Fifo Threshold 4

Memory Mode Disable

Memory Type Macromix

Memory Size 1 GBits

Chip Select High Time Cycle 1

Free Running Clock Disable

Clock Mode Low

Wrap Size Not Supported

Clock Prescaler 0

Sample Shifting None

Delay Hold Quarter Cycle Enable

Chip Select Boundary Disabled

Maximum Transfer 0

Refresh Rate 0

Memory Select NCS1

Switching Duration Clock Number 1

Connectivity

	FSBL	Application	ExtMemLoader
ETH1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FDCAN1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FDCAN2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FDCAN3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FMC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2C1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2C2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2C3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2C4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I3C1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I3C2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LPUART1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MDIOS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SDMMC1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SDMMC2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UCPDI	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USB1_OTG_HS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USB2_OTG_HS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XSPI2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ZSIF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NSPIM	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

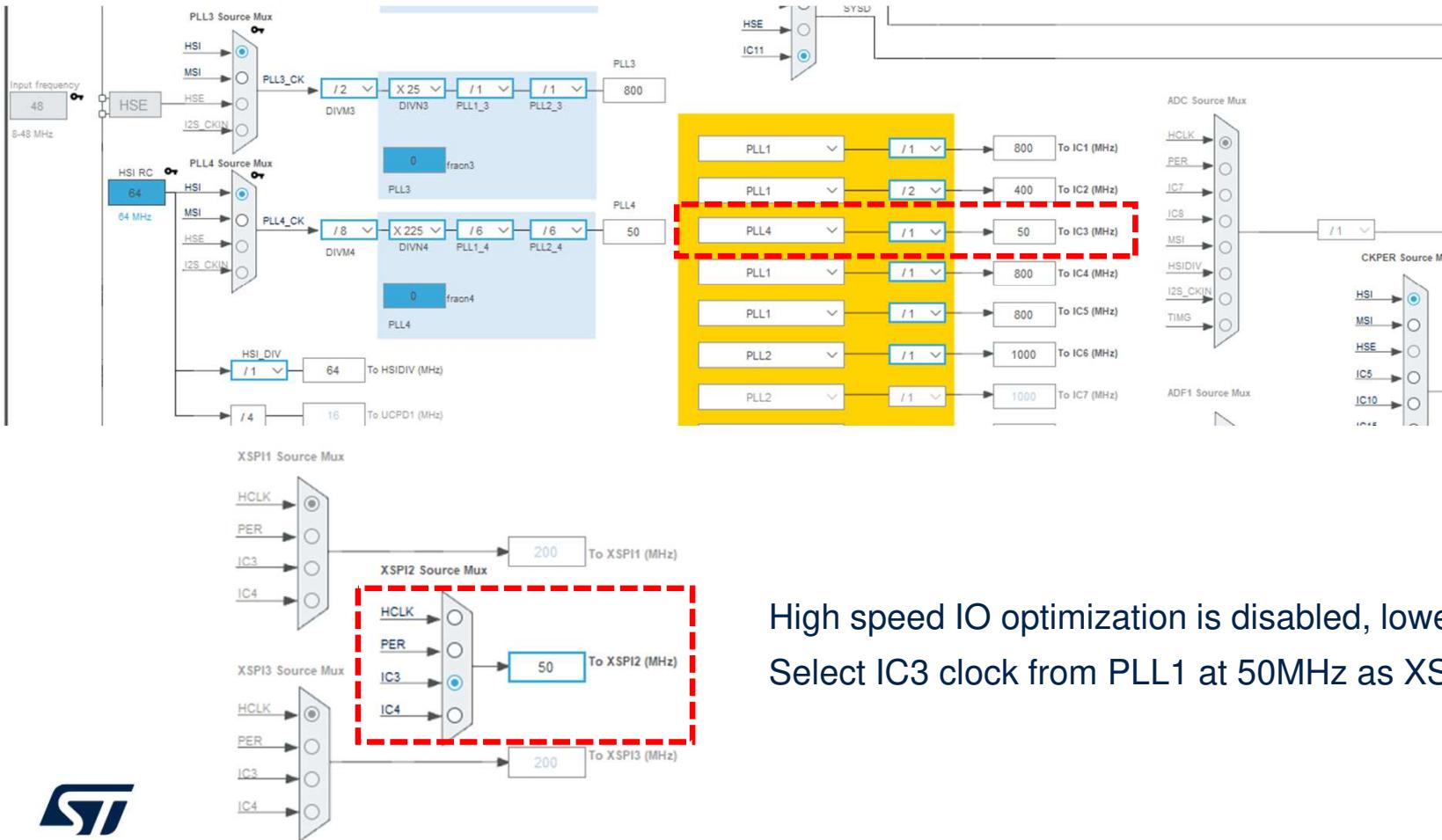


# Octo-SPI Flash Memory Configuration

OTP124	-	HCONF1	-
	[0]	IWDG1_HW	IWDG1 start on reset
	[1]	IWDG1_FZ_STOP	IWDG1 freeze in Stop mode
	[2]	IWDG1_FZ_STANDBY	IWDG1 freeze in Standby mode
	[9:3]	Reserved	Reserved
	[10]	RST_STOP	Reset caused if the device is put in Stop mode
	[11]	RST_STDBY	Reset caused if the device is put in Standby mode
	[12]	SELINBORH	- 0: BOR disabled - 1: BOR = 2.7 V
	[13]	HSLV_VDDIO5	VDDIO5 I/O segment below 2.5 V for I/O mode. The I/O segment is used by SDMMC2 port.
	[14]	HSLV_VDDIO4	VDDIO4 I/O segment below 2.5 V for I/O mode (I/O segment used by SDMMC1 port)
	[15]	HSLV_VDDIO3	VDDIO3 I/O segment below 2.5 V for I/O mode (I/O segment used by XSPIM port 2)



# Octo-SPI Flash Memory Configuration



High speed IO optimization is disabled, lower XSPI clock speed.  
Select IC3 clock from PLL1 at 50MHz as XSPI2 source.



# OTP Configuration (Optional)

The screenshot shows the STM32CubeProgrammer software interface for OTP configuration. The title bar reads "STM32CubeProgrammer". The main window is titled "OTP MPU". The left sidebar has icons for File, Project, STM32CubeProgrammer, and a lock icon labeled "OTP". The main area displays a table of OTP settings:

Name	Word	Value	Status	Lock	Description
▶ Reserved	OTP121	0x011D6C6D	0x40000000	<input checked="" type="checkbox"/>	Permanent write lock
▶ Reserved	OTP122	0x95C0077C	0x40000000	<input checked="" type="checkbox"/>	Permanent write lock
▶ Reserved	OTP123	0xAE087777	0x40000000	<input checked="" type="checkbox"/>	Permanent write lock
▼ HCONF1	OTP124	0x00018000	0x00000000	<input type="checkbox"/>	-
	IWDG1_HW	0x0			IWDG1 start on reset
	IWDG1_FZ_STOP	0x0			IWDG1 freeze in Stop mode
	IWDG1_FZ_STAN...	0x0			IWDG1 freeze in Standby mode
	RST_STOP	0x0			Reset caused if the device is put in Stop mode
	RST_STANDBY	0x0			Reset caused if the device is put in Standby mode
	SELINBORH	0x0			0: BOR disabled. 1: BOR = 2.7 V
	HSLV_VDDIO5	0x0			VDDIO5 I/O segment below 2.5 V for I/O mode. The I/O segment is used by SDMMC2 port.
	HSLV_VDDIO4	0x0			VDDIO4 I/O segment below 2.5 V for I/O mode (I/O segment used by SDMMC1 port)
	HSLV_VDDIO3	0x1			VDDIO3 I/O segment below 2.5 V for I/O mode (I/O segment used by XSPIM port 2)
	HSLV_VDDIO2	0x1			VDDIO2 I/O segment below 2.5 V for I/O mode (I/O segment used by XSPIM port 1)
	HSLV_VDD	0x0			Main I/O segment below 2.5 V for I/O mode

At the bottom, there are buttons for "Export", "Save OTP partition", "Lock all" (with a checkbox), "Apply", and "Read".

You can fuse OTP in **STM32CubeProgrammer**.

However, since the **OTP** area can no longer be written once fused, **it must be used with extreme caution**.

In this hands-on, **OTP settings will not be configured**.



# EXTMEM\_MANAGER Configuration

Middleware and Software Packs

	FSBL	Application	ExtMemLo...
AIROC-Wi-Fi-Bluetooth-STM32			
EXTMEM LOADER			
EXTMEM MANAGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
EXTMEM_MANAGER_APPLI			

Select **FSBL** as the Context for **EXTMEM\_MANAGER**.

EXTMEM\_MANAGER Mode and Configuration

Mode

Runtime contexts:	First Stage Boot Loader	Application	External Memory Loader
	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/> Activate External Memory Manager			

Boot usecase

Configure the below parameters :

Search (Ctrl+F)

Boot

Select boot code generation	<input checked="" type="checkbox"/>
Selection of the boot system	Load and Run
Header size	0x400

LRUN source

select the source memory	Memory 1
source address offset	0x00020000 hex
source code size	0x00010000 hex

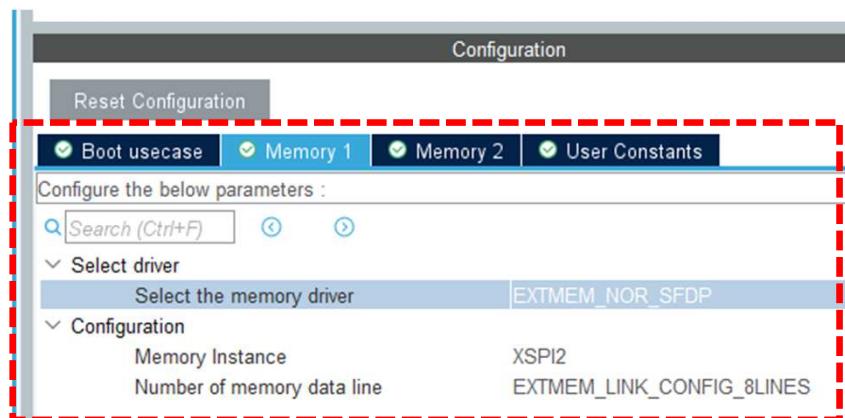
LRUN destination

selection of the memory	Internal Memory
destination address	0x34000000 hex

Set the **Boot System** to **Load and Run**.



# EXTMEM\_MANAGER Configuration



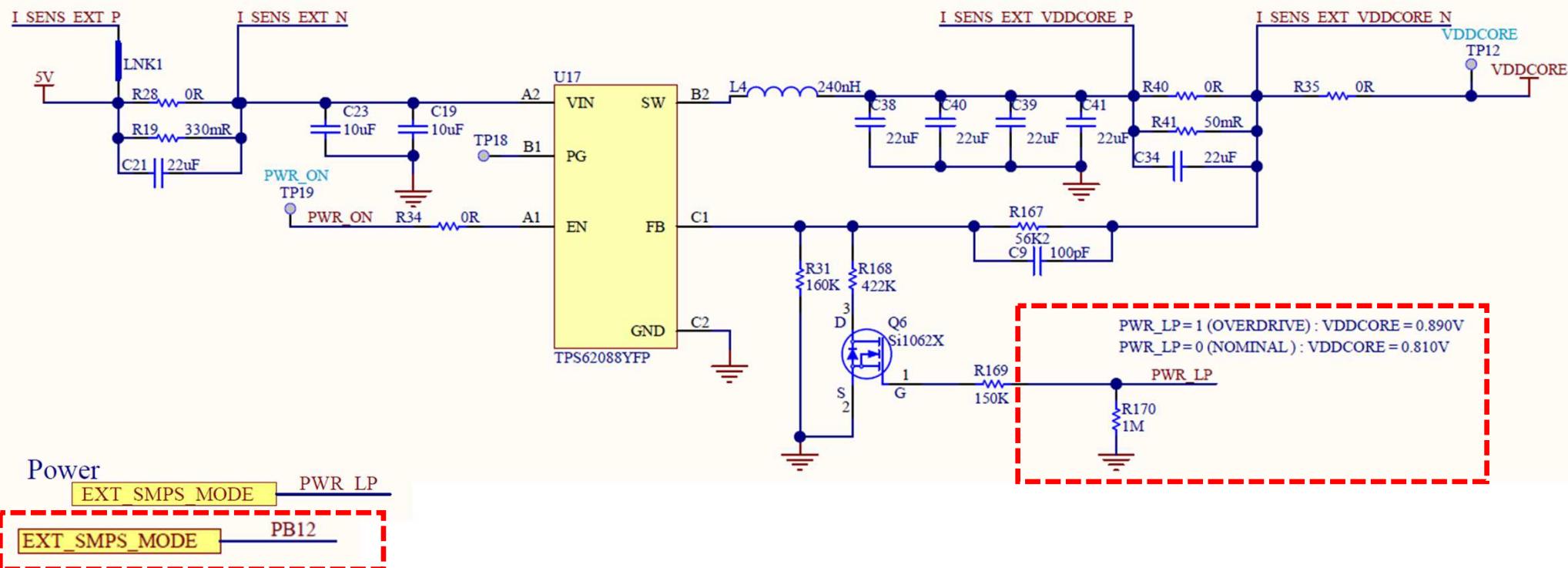
In this hands-on, FSBL loads the **Secure Application** into **AXISRAM1 (0x34000000)**.

Table 1. Memory map based on IDAU mapping (continued)

IDAU security type	Mapping	Description
Reserved SRAM/AXI bank secure	0x34270000	AXISRAM4
	0x34200000	AXISRAM3
	0x34100000	AXISRAM2
	0x34000000	AXISRAM1 (FLEXMEM extension bites on the lower end)



# PWR\_LP\_PIN Configuration



To use **Overdrive Mode**, set the **PWR\_LP(PB12)** pin to **High** to configure the **External DCDC** output voltage to **0.89V**.

Home > STM32N657X0HxQ > stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_test.ioc - Pinout & Configuration >

### Pinout & Configuration

Categories A-Z

	FSBL	Application	ExtMemLoader
CORTEX_M55_FSBL	<input checked="" type="checkbox"/>		
CORTEX_M55_S		<input checked="" type="checkbox"/>	
CDPDMA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
GPIO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
HDPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICACHE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NVIC1_S_Application		<input checked="" type="checkbox"/>	
NVIC_FSBL	<input checked="" type="checkbox"/>		
RAMCFG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RCC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SYS_S	<input type="checkbox"/>	<input type="checkbox"/>	
WWDG	<input type="checkbox"/>	<input type="checkbox"/>	

Analog >

Timers >

Multimedia >

Security >

Computing >

Middleware and Software Packs >

Trace and Debug >

	FSBL	Application	ExtMemLoader
DEBUG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HDP	<input checked="" type="checkbox"/>		

### Clock Configuration

Software Packs

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO XSPI Features

Search Signals Search (Ctrl+F)

Show only Modified Pins

Pin N...	Signal on...	Pin Conte...	Pin Privil...	GPIO out...	GPIO mode	GPIO Pul...	Maximu...	User Label	Modified
PB12	n/a	First Sta...	n/a	High	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG0	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG8	n/a	Free	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG10	n/a	Free	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>

Set the **Context of PB12 to First Stage Boot Loader** and configure the **GPIO output level to High**.

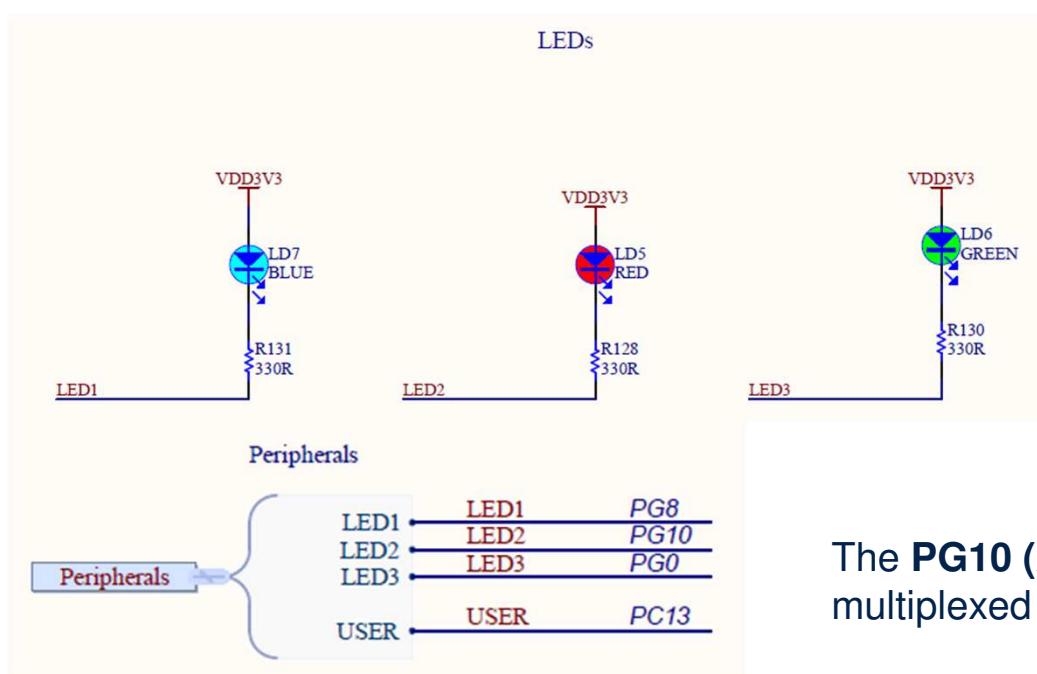
PB12 Configuration :

Pin Context Assignment	First Stage Boot Loader
GPIO output level	High
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	Low
User Label	



Set PB12 PIN as **GPIO\_Output**.

# LED\_PIN Configuration



The **PG10 (AF11)** is used as the **BootFailed** pin. This pin is multiplexed with **UART5\_TX** to send **UART status traces**.

Table 29. Pin configuration for UART5

UART5	
UART5_TX	PG10 (AF11)

It is a specific UART instance to retrieve data in case of blocking failure.



# LED\_PIN Configuration

Categories A-Z

	FSBL	Application	ExtMemLoader
CORTEX_M55_FSBL	<input checked="" type="checkbox"/>		
CORTEX_M55_S		<input checked="" type="checkbox"/>	
GPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>GPIO</b>	<b><input checked="" type="checkbox"/></b>	<b><input checked="" type="checkbox"/></b>	<b><input type="checkbox"/></b>
HPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICACHE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NVIC1_S_Application		<input checked="" type="checkbox"/>	
NVIC_FSB	<input checked="" type="checkbox"/>		
RAMCFG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RCC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SYS_S	<input type="checkbox"/>	<input type="checkbox"/>	
UART0	<input type="checkbox"/>	<input type="checkbox"/>	

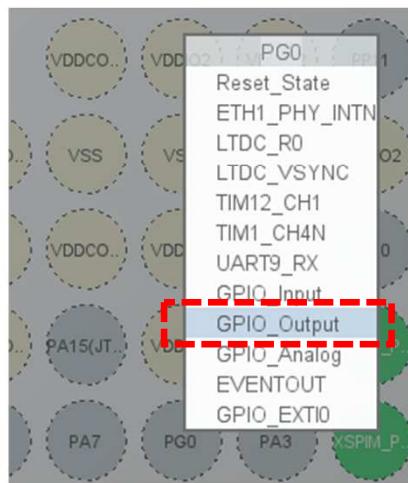
Configuration

Group By Peripherals

GPIO  XSPI  Features

Search Signals   Show only Modified Pins

Pin N...	Signal on...	Pin Conte...	Pin Privil...	GPIO out...	GPIO mode	GPIO Pul...	Maximu...	User Label	Modified
PB12	n/a	First Sta...	n/a	High	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG0	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG8	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG10	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>



PG0 Configuration :-

Pin Context Assignment  Application

GPIO output level

GPIO mode

GPIO Pull-up/Pull-down

Maximum output speed

User Label

# BSEC Configuration

The screenshot shows the Pinout & Configuration interface with two tabs: "Pinout & Configuration" and "Clock Configuration".

**Pinout & Configuration Tab:**

- Search bar:  ▾
- Categories: [Categories](#) [A-Z](#)
- System Core
- Analog
- Timers
- Connectivity
- Multimedia
- Security
- BSEC** (highlighted with a red dashed box)
- CRT1
- HASH
- MCE1
- MCE2
- MCE3

**Clock Configuration Tab:**

BSEC Mode and Configuration

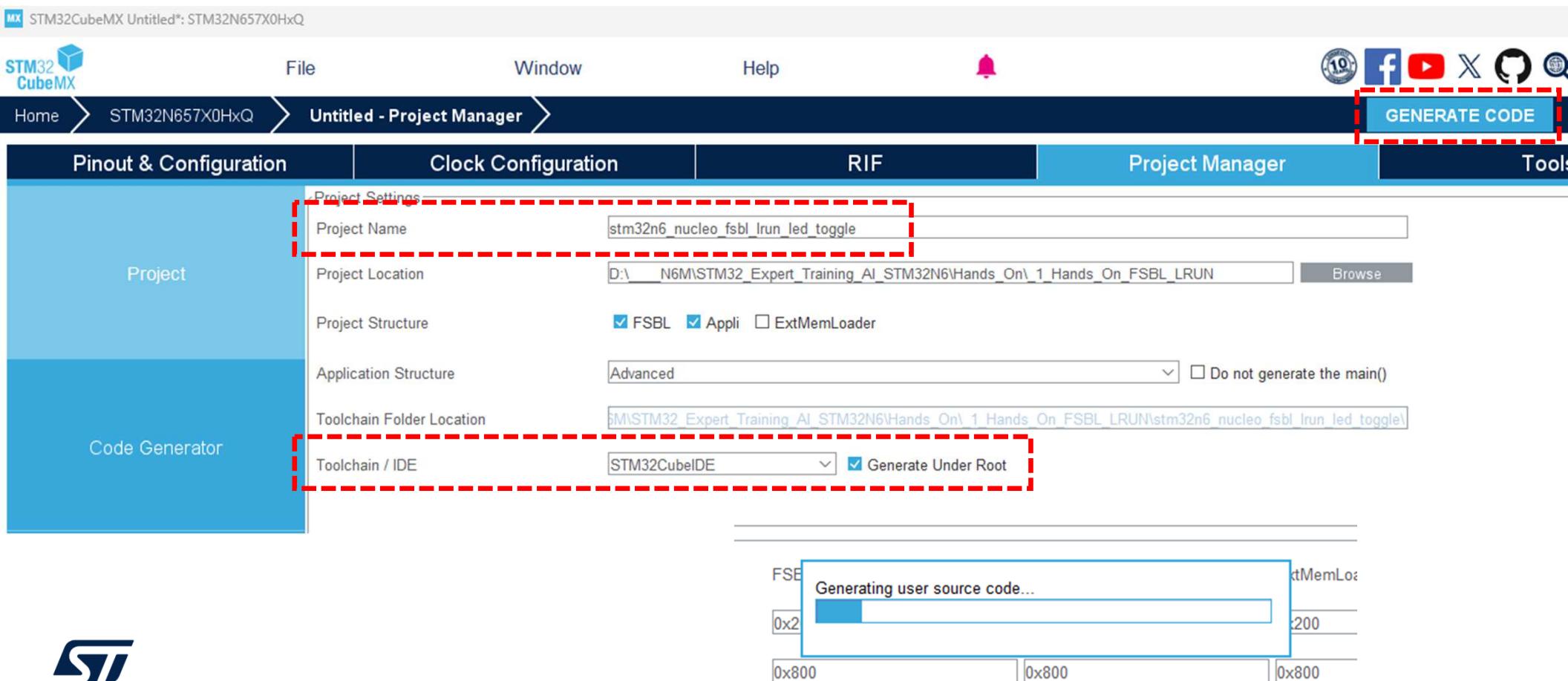
Mode			
Runtime contexts:	First Stage Boot Loader	Application	External Memory Loader
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Activated			

Configuration

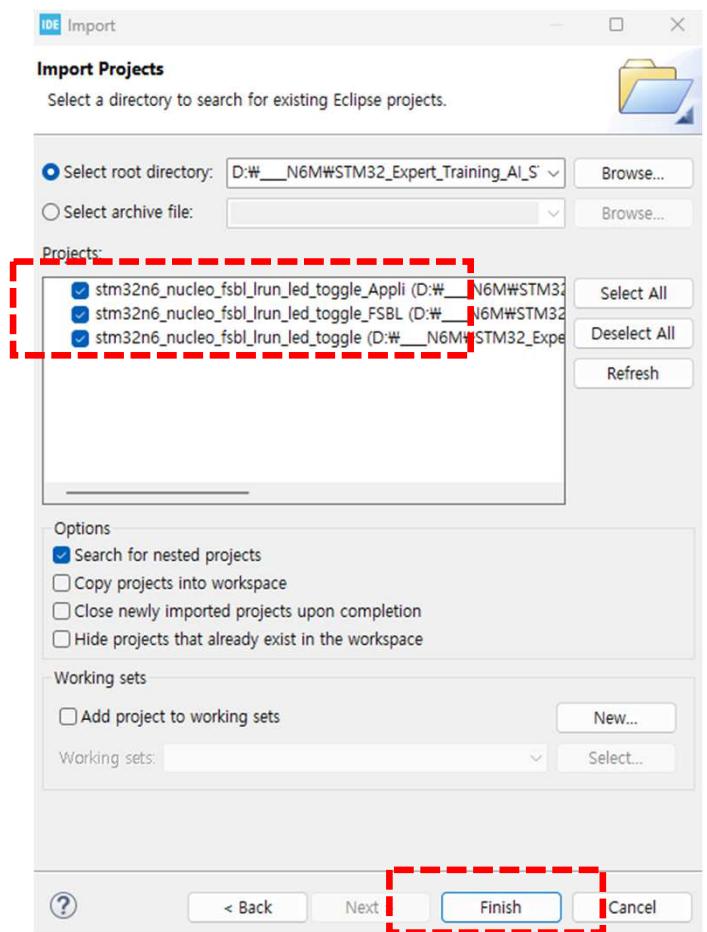
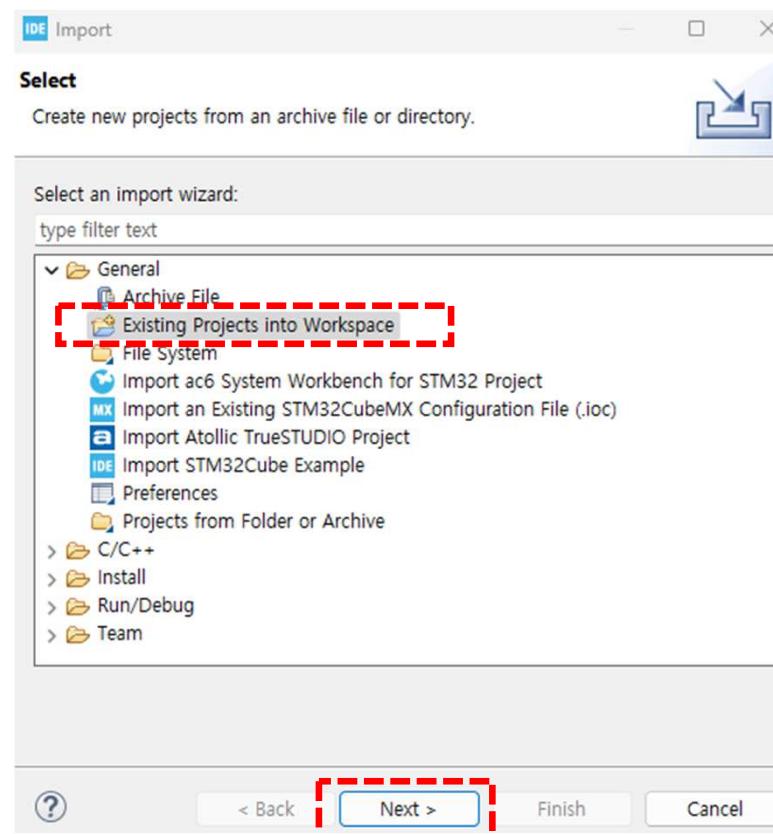
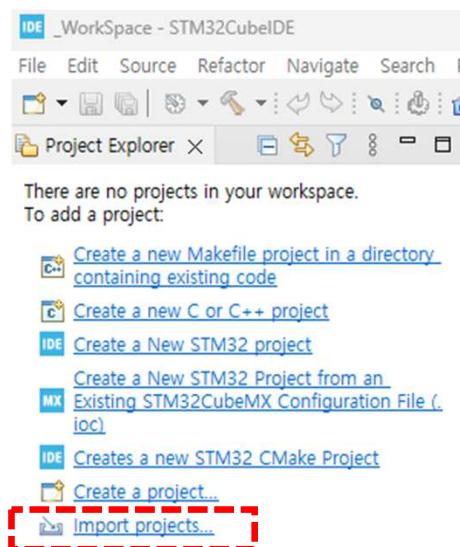
**Warning:** This peripheral has no parameters to be configured.



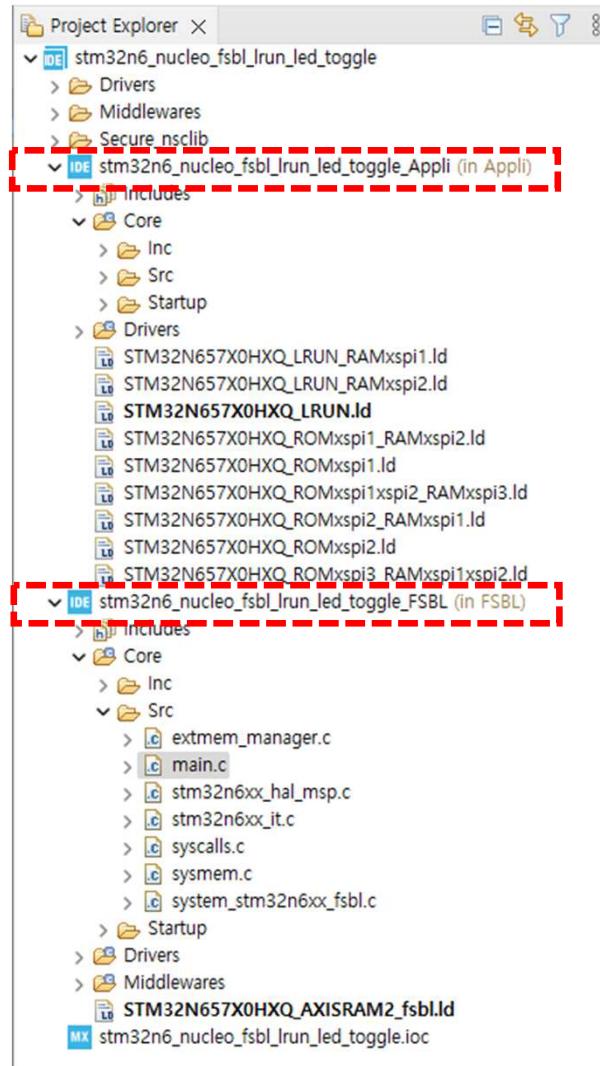
# GENERATE CODE



# STM32CubeIDE Import Project



# STM32CubeIDE Import Project



Two projects are created: **FSBL** and **Secure Application**.

# Add Code to FSBL

IDE \_WorkSpace - stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer X

stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle

- Drivers
- Middlewares
- Secure\_nsclib
- stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_Appli (in Appli)
- stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL (in FSBL)
- Includes
- Core
  - Inc
  - Src
    - extmem\_manager.c
    - main.c
    - stm32n6xx\_hal\_msp.c
    - stm32n6xx\_it.c
    - syscalls.c
    - sysmem.c
    - system\_stm32n6xx\_fsbl.c
  - Startup
- Drivers
- Middlewares

STM32N657X0HXQ\_AXIRAM2\_fsbl.ld

stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle.ioc

main.c X

```
76  /* Enable the CPU Cache */
77
78  /* Enable I-Cache-----*/
79  SCB_EnableICache();
80
81  /* Enable D-Cache-----*/
82  SCB_EnableDCache();
83
84  /* MCU Configuration-----*/
85  HAL_Init();
86
87  /* USER CODE BEGIN Init */
88
89  /* USER CODE END Init */
90
91  /* Configure the system clock */
92  SystemClock_Config();
93
94  /* USER CODE BEGIN SysInit */
95
96  /* USER CODE END SysInit */
97
98  /* Initialize all configured peripherals */
99  MX_GPIO_Init();
100  MX_BSEC_Init();
101  MX_XSPI2_Init();
102  MX_EXTMEM_MANAGER_Init();
103  /* USER CODE BEGIN 2 */
104
105  /* USER CODE END 2 */
```

main.c X

```
79  SCB_EnableICache();
80
81  /* Enable D-Cache-----*/
82  SCB_EnableDCache();
83
84  /* MCU Configuration-----*/
85  HAL_Init();
86
87  /* USER CODE BEGIN Init */
88
89  MX_GPIO_Init();
90  HAL_Delay(1);
91
92  /* USER CODE END Init */
93
94  /* Configure the system clock */
95  SystemClock_Config();
96
97  /* USER CODE BEGIN SysInit */
98
99  /* USER CODE END SysInit */
100
101 /* Initialize all configured peripherals */
102 MX_GPIO_Init();
103 MX_BSEC_Init();
104 MX_XSPI2_Init();
105 MX_EXTMEM_MANAGER_Init();
106 /* USER CODE BEGIN 2 */
107
108 /* USER CODE END 2 */
```

MX\_GPIO\_Init();  
HAL\_Delay(1);

Add code to set **PB12 High** before configuring the clock to enable **Overdrive Mode (Core Clock: 800MHz)** in the “main.c”

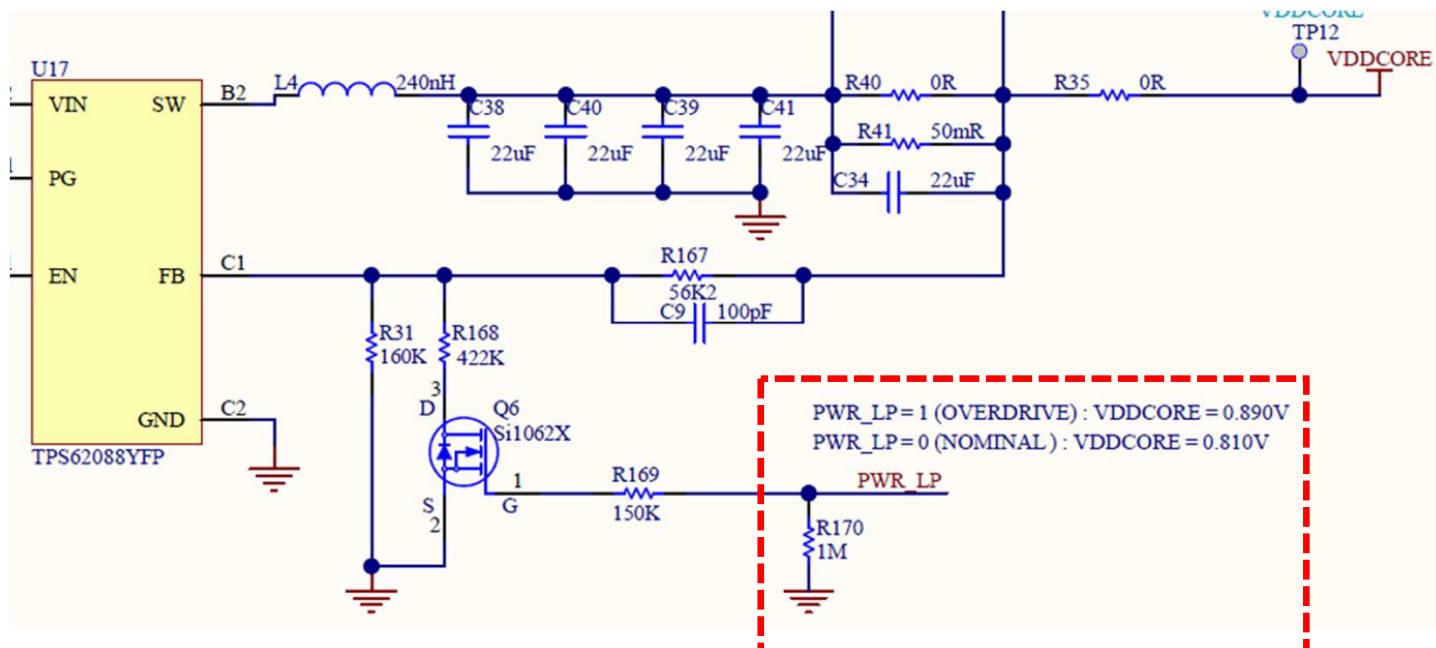


# Add Code to FSBL

```

306 /**
307 * @brief GPIO Initialization Function
308 * @param None
309 * @retval None
310 */
311 static void MX_GPIO_Init(void)
312 {
313     GPIO_InitTypeDef GPIO_InitStruct = {0};
314     /* USER CODE BEGIN MX_GPIO_Init_1 */
315
316     /* USER CODE END MX_GPIO_Init_1 */
317
318     /* GPIO Ports Clock Enable */
319     __HAL_RCC_GPIOA_CLK_ENABLE();
320     __HAL_RCC_GPIOB_CLK_ENABLE();
321
322     /*Configure GPIO pin Output Level*/
323     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, GPIO_PIN_SET);
324
325     /*Configure GPIO pin : PB12 */
326     GPIO_InitStruct.Pin = GPIO_PIN_12;
327     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
328     GPIO_InitStruct.Pull = GPIO_NOPULL;
329     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
330     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
331
332     /* USER CODE BEGIN MX_GPIO_Init_2 */
333
334     /* USER CODE END MX_GPIO_Init_2 */
335 }

```



# Add Code to FSBL

The screenshot shows the STM32CubeIDE interface. On the left, the Project Explorer displays a project structure for a Nucleo board. A red dashed box highlights the file `stm32n6xx_hal_msp.c` in the `Src` folder under the `Core` section. The right pane shows the code editor for this file. A red dashed box highlights the code block between `/* USER CODE BEGIN Define */` and `/* USER CODE END Define */`. The code consists of two define statements:

```
#define HSLV OTP 124
#define VDDIO3_HSLV_MASK (1<<15)
```

Add code between `/* USER CODE BEGIN Define */` and `/* USER CODE END Define */` in `stm32n6xx_hal_msp.c`.

`#define HSLV OTP 124`  
`#define VDDIO3_HSLV_MASK (1<<15)`



# Add Code to FSBL

The screenshot shows the STM32CubeIDE interface. On the left, the Project Explorer displays the project structure for "stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle". The file "stm32n6xx\_hal\_msp.c" is open in the code editor. A red dashed box highlights the code being added to the XSPI MspInit function. The code adds BSEC and SYSCFG clock enablements, reads fuse data, sets PWR and SYSCFG configurations, and enables the XSPI memory interface clock.

```
88 void HAL_XSPI_MspInit(XSPI_HandleTypeDef* hxsip)
89 {
90     GPIO_InitTypeDef GPIO_InitStruct = {0};
91     RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};
92     if(hxsip->Instance==XSPI2)
93     {
94         /* USER CODE BEGIN XSPI2_MspInit 0 */
95
96         BSEC_HandleTypeDef hbsec;
97         uint32_t fuse_data = 0;
98         /* Enable BSEC & SYSCFG clocks to ensure BSEC data accesses */
99         __HAL_RCC_BSEC_CLK_ENABLE();
100        __HAL_RCC_SYSCFG_CLK_ENABLE();
101
102        hbsec.Instance = BSEC;
103        if (HAL_BSEC OTP_Read(&hbsec, HSLV OTP, &fuse_data) != HAL_OK)
104        {
105            Error_Handler();
106        }
107
108        /* Set PWR configuration for IO speed optimization */
109        __HAL_RCC_PWR_CLK_ENABLE();
110        HAL_PWREx_EnableVddIO3();
111        HAL_PWREx_ConfigVddIORange(PWR_VDDIO3, PWR_VDDIO_RANGE_1V8);
112        /* Set SYSCFG configuration for IO speed optimization (clock already enabled) */
113        HAL_SYSCFG_EnableVDDIO3CompensationCell();
114
115        /* Enable the XSPI memory interface clock */
116        HAL_RCC_XSPI2_CLK_ENABLE();
117
118     /* USER CODE END XSPI2_MspInit 0 */
119 }
```



# Add Code to FSBL

```
BSEC_HandleTypeDef hbsec;
uint32_t fuse_data = 0;
/* Enable BSEC & SYSCFG clocks to ensure BSEC data accesses */
__HAL_RCC_BSEC_CLK_ENABLE();
__HAL_RCC_SYSCFG_CLK_ENABLE();

hbsec.Instance = BSEC;
if (HAL_BSEC OTP_Read(&hbsec, HSLV OTP, &fuse_data) != HAL_OK)
{
Error_Handler();
}

/* Set PWR configuration for IO speed optimization */
__HAL_RCC_PWR_CLK_ENABLE();
HAL_PWREx_EnableVddIO3();
HAL_PWREx_ConfigVddIORange(PWR_VDDIO3, PWR_VDDIO_RANGE_1V8);
/* Set SYSCFG configuration for IO speed optimization (clock already enabled)
*/
HAL_SYSCFG_EnableVDDIO3CompensationCell();

/* Enable the XSPI memory interface clock */
__HAL_RCC_XSPI2_CLK_ENABLE();
```



# Build FSBL

The screenshot shows the STM32CubeIDE interface with the following components:

- Project Explorer:** Displays the project structure under "stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle".
- STM32N657X0HXQ\_AXISRAM2\_fsbl.ld:** A linker script file containing memory and section definitions.
- CDT Build Console:** Shows the build log output.

**Linker Script Content (stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle.FSBL.lds):**

```
31 ****
32 */
33
34 /* Entry Point */
35 ENTRY(Reset_Handler)
36
37 /* Highest address of the user mode stack */
38 _estack = ORIGIN(RAM) + LENGTH(RAM); /* end of "RAM" Ram type memory */
39 _sstack = _estack - _Min_Stack_Size;
40
41 _Min_Heap_Size = 0x200; /* required amount of heap */
42 _Min_Stack_Size = 0x800; /* required amount of stack */
43
44 /* Memories definition */
45 MEMORY
46{
47    ROM   (xrw)      : ORIGIN = 0x34180400, LENGTH = 255K
48    RAM   (xrw)      : ORIGIN = 0x341C0000, LENGTH = 256K
49}
50
51 /* Sections */
52 SECTIONS
53{
54    /* The startup code into "RAM"
55    .isr_vector :
56    {
57        . = ALIGN(4);
58        KEEP(.isr_vector) /* Start of interrupt vector table */
59        . = ALIGN(4);
60    } >ROM
61}
```

**CDT Build Console Output:**

```
18:17:17 **** Incremental Build of configuration Debug for project stm32n6_nucleo_fsbl_lrun_led_toggle
make -j16 all
arm-none-eabi-gcc "../Core/Src/stm32n6xx_hal_msp.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDEBUG -DUSE_H
arm-none-eabi-gcc -o "stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf" @"objects.list" -mcpu=cortex-m55
Finished building target: stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf

arm-none-eabi-size stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf
arm-none-eabi-objdump -h -S stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf > "stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.h
text    data    bss    dec    hex filename
61020     12   3268   64300   fb2c stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf
arm-none-eabi-objcopy -O binary stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf "stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin
Finished building: default.size.stdout

Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin

Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.list

18:17:18 Build Finished. 0 errors, 0 warnings. (took 1s.912ms)
```

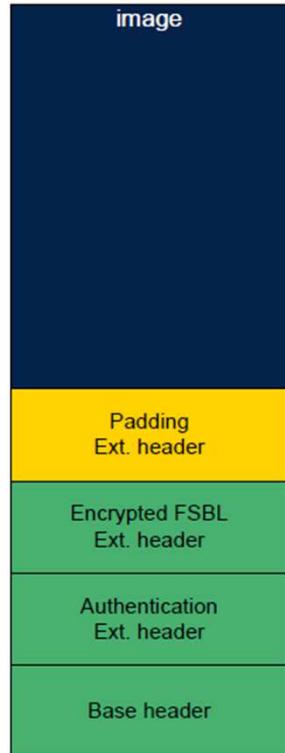
# Debug

The screenshot shows the STM32CubeIDE interface with the following details:

- Project Explorer:** Displays the project structure under "stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle".
- Code Editor:** Shows the main.c file with code related to the application entry point.
- Context Menu:** Opened over the main.c file, showing options like "Run As", "Debug As", and "Properties".
- Edit launch configuration properties Dialog:** Opened, showing the "Edit launch configuration properties" dialog with the following settings:
  - Name: stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL
  - Project: stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL
  - C/C++ Application: Debug/stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL.elf
  - Build Configuration: Select Automatically
  - Build (if required) before launching
  - Build Configuration dropdown: Select Automatically
  - Build Configuration buttons: Enable auto build (unchecked), Disable auto build (unchecked), Use workspace settings (checked)
  - Buttons: Revert, Apply, OK, Cancel



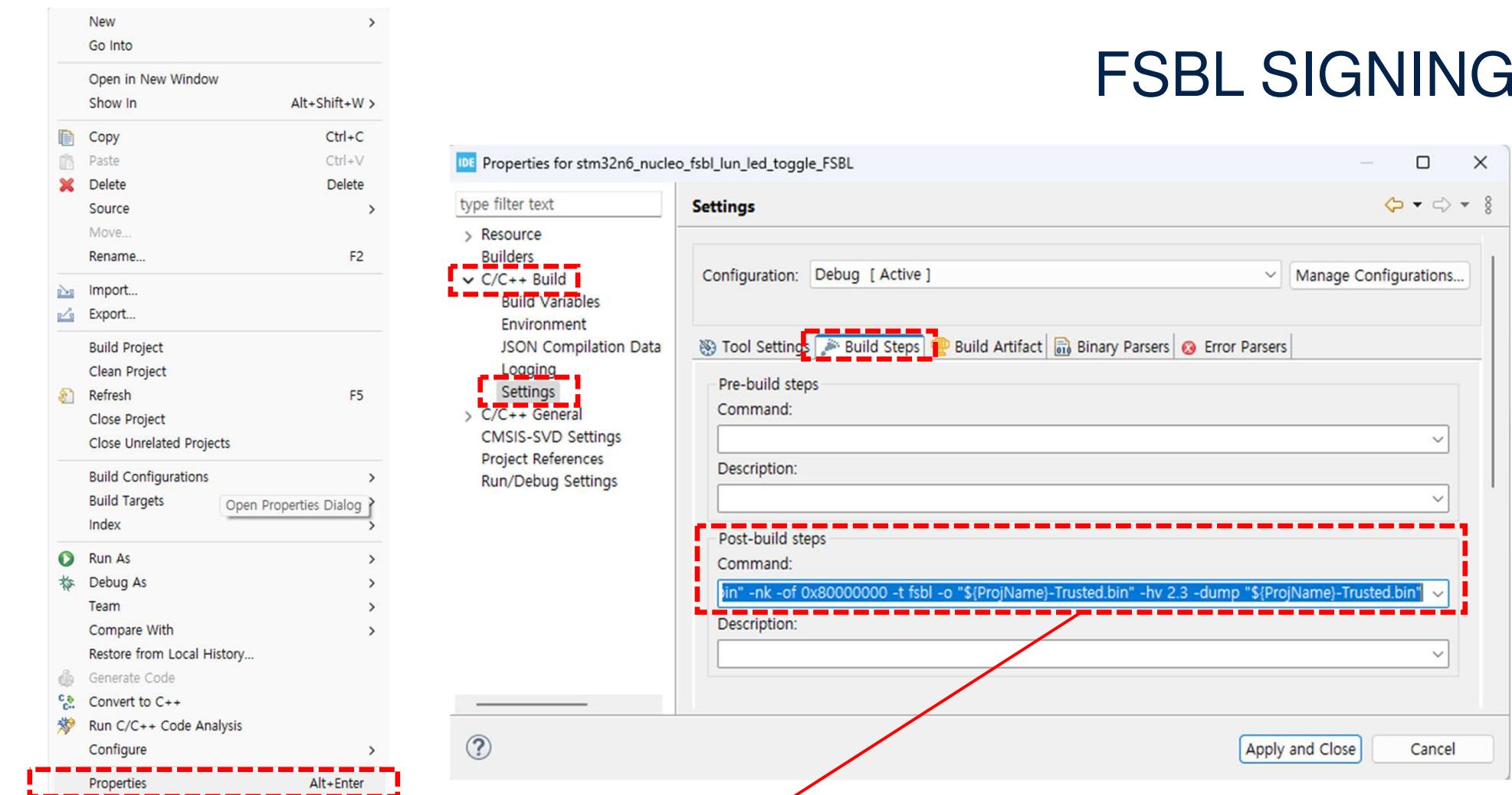
# FSBL Header



**Table 32. Base header**

Name	Length	Byte offset (Dec, hex)	Description	Part of signature
Magic number	32 bits	0, 0x0	'S'; 'T'; 'M'; 0x32	n
Image signature	768 bits	4, 0x4	ECDSA signature: calculated on header + image	n
Image checksum	32 bits	100, 0x64	Checksum of the secured payload	n
Header version	32 bits	104, 0x68	Header version v2.3 = 0x00MMmm00MM: major version = 0x02mm: minor version = 0x03	y
Image length	32 bits	108, 0x6C	Length of FSBL image in bytes	y
Image entry point	32 bits	112, 0x70	Entry point of image	y
Reserved1	32 bits	116, 0x74	Reserved (64-bits entry point)	y
Load address	32 bits	120, 0x78	Load address of image	y
Reserved2	32 bits	124, 0x7C	Reserved (64-bits load address)	y
Version number	32 bits	128, 0x80	Image version (monotonic number)	y
Extension flags	32 bits	132, 0x84	b0=1: Authentication extension header b1=1: FSBL encryption extension header b31=1: Padding extension header	y
Post header length	32 bits	136, 0x88	Length in bytes of all extension headers	y
Binary type	32 bits	140, 0x90	Used to check the binary type	y
PAD	64 bits	144, 0x94	Reserved padding bytes. Must all be set to 0	y
Nonsecure payload length	32 bits	152, 0x98	Length in bytes of optional nonsecured payload	n
Nonsecure payload hash	32 msb bits	156, 0x9C	32 msb bits of SHA256 of nonauthenticated payload	n

# FSBL SIGNING



```
cd "${ProjDirPath}/Debug" && echo y | "C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe" -bin "${ProjName}.bin" -nk -of 0x80000000 -t fsbl -o "${ProjName}-Trusted.bin" -hv 2.3 -dump "${ProjName}-Trusted.bin"
```

# FSBL SIGNING

```
STM32 Signing Tool v2.19.0

Adding head padding bytes for payload with header v2.3
Header version 2.3 preparation...
Extracting Entry point value from the input file...
Entry point value : 0x341813ad
The headed image file generated successfully: stm32n6_nucleo_fsbl_lun_led_toggle_FSBL-Trusted.bin

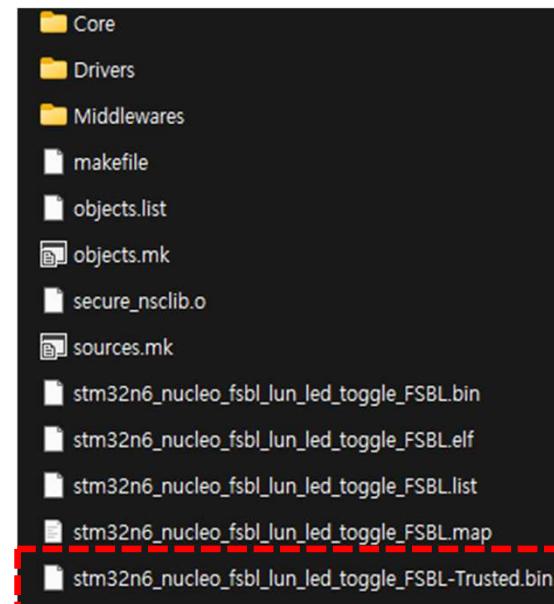
Header description:

Magic: 0x53544d32
Signature: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Checksum: 0x69e803
Header version: 0x20300
Size: 0xf1e0
Load address: 0xffffffff
Entry point: 0x341813ad
Image version: 0x0
Extension: 0x80000000

Pad header detected:
Type: 0x5354ffff
Size: 0x1a0
Padding values: 92 05 ca bb 60 ea 25 df e8 39 35 3b 5a c6 34 b4 9d ae c3 59 9a 26 eb ec d7 f0 7b 9b f0 4b 6b
28 4a dd fd 2f e4 58 80 ab 17 0b af 32 5d a6 17 82 7a 9a c7 8d 50 ef e0 85 3f ac 11 9a dd 72
a0 4d 8d 93 00 5d d1 cc 7e fa ce a0 cb 58 1e ba cc 59 f6 44 b0 d2 f6 01 32 36 21 38 12 de 9b
c3 75 31 44 4e 3b e7 fb 2e 19 4d d9 56 08 5f 89 00 5b d8 b9 32 1c dc 66 f5 ed 28 bb 5c 83 75
39 81 ac 91 38 e6 1c ce bf a9 32 57 c0 b8 87 33 c3 d6 08 53 35 95 1a 79 c6 ae d7 eb 4d e7 8c
0a c7 4a bf e9 a7 31 7a 8f f1 ea 93 26 ca a9 46 91 7f 07 22 46 6b 61 09 08 b3 91 e4 0e 5c 74
9f e5 4e 7a 62 0a c1 f6 9a 51 de e3 cc 61 9c 93 f1 08 fb 74 ae 25 50 b6 16 b6 18 6c 11 7a 7a
06 f0 b7 82 47 cb f5 67 b4 76 da 67 c1 4d de f5 57 e0 10 37 1e 2d 0e 60 74 b0 10 22 24 7f af
b0 6f ae a0 cd 2f bf 7a cd 4c ee 4c 8f ae e6 c9 91 b5 1b bb ed 57 d0 a2 62 57 80 4d 39 c0 5b
d9 aa 49 70 e5 07 cc c3 fa 5b 8a 5f 70 75 05 57 42 a7 e3 55 d1 d3 45 b3 01 c2 69 13 73 e7 ba
87 c9 ac 86 05 d7 4c c1 3f c9 59 83 79 2e 0e 26 1a ce f4 9b 73 a3 ca 6a 40 0e 23 81 85 0e 8a
97 19 e0 ac 25 e4 5e a9 eb 78 72 23 a2 2b b7 da b2 31 c9 e3 16 5a 58 af 6d f6 08 91 c5 96 f1
ad 22 9f 54 f4 2f 10 e5 00 00 02 00 51 38 78 dd a8 2a 00 00 e0 45 3c 9c a1 02 00 00 30 b7 47
9c a1 02 00 00

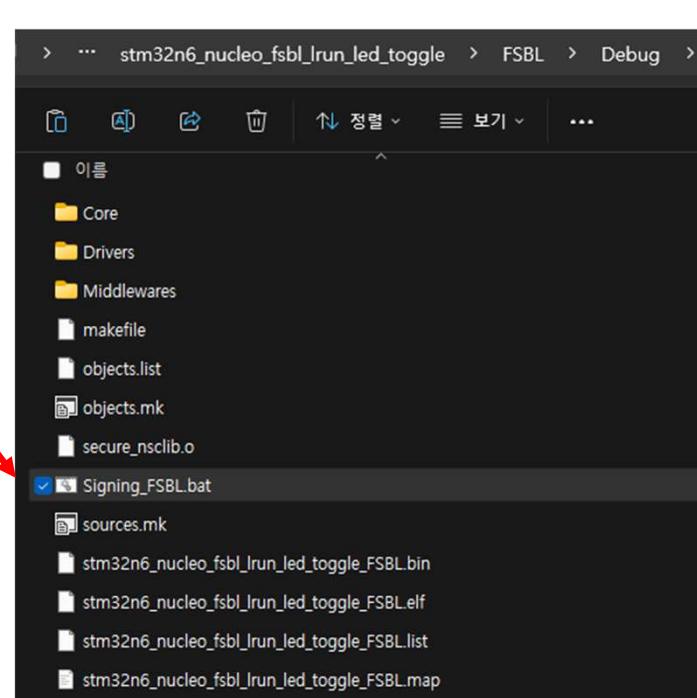
Post header 2.3 information :
Binary type: 0x10
Non authenticated payload length: 0x0
Non authenticated payload hash: 0x0
```

16:27:56 Build Finished. 0 errors, 0 warnings. (took 2s.383ms)



# FSBL SIGNING

It is also possible to use the Signing Tool externally.



Copy the **Signing\_FSB.bat** file from the **Siging\_Batch\_File** folder into the **FSBL\Debug** folder of the current project.



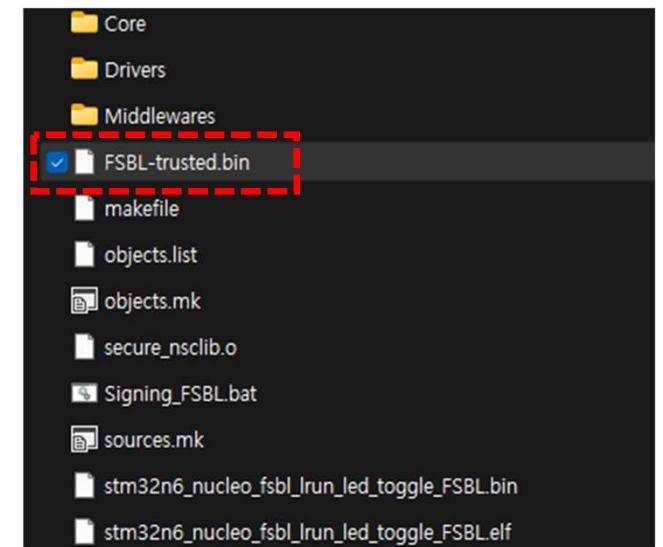
# FSBL SIGNING

```
 Signing_FSBL.bat
1 @echo off
2
3 REM STM32MP_SigningTool_CLI.exe -bin FSBL.bin -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
4
5 set SIGNING_TOOL_PATH="C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe"
6 set APP_NAME="stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin"
7
8 echo %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
9 %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
10
11 cmd /k
```

Paste the path of the **STM32\_SigningTool\_CLI.exe** file from **STM32CubeProgrammer**.

After building the **FSBL**, paste the name of the generated **binary file**.





# FSBL SIGNING

The signed file "**FSBL-trusted.bin**" is generated upon completing the signing process.

# Add Code to APP

The screenshot shows the STM32CubeIDE interface. On the left, the Project Explorer displays the project structure under 'stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle'. It includes sub-folders like Drivers, Middlewares, Secure\_nsclib, and Core. The Core folder contains Includes and Src, with Src being expanded to show files such as main.c, secure\_nscc.c, and various HAL and system files. On the right, the main.c editor window shows the C code for the application. The code includes initialization functions for peripherals and a main loop that toggles three LEDs connected to GPIO pins 0, 8, and 10. A blue selection bar highlights the section of code from line 88 to line 103.

```
72 HAL_Init();
73 /* USER CODE BEGIN Init */
74
75 /* USER CODE END Init */
76
77 /* USER CODE BEGIN SysInit */
78
79 /* USER CODE END SysInit */
80
81 /* Initialize all configured peripherals */
82 MX_GPIO_Init();
83 SystemIsolation_Config();
84 /* USER CODE BEGIN 2 */
85
86
87 /* USER CODE END 2 */
88
89 /* Infinite loop */
90 /* USER CODE BEGIN WHILE */
91 while (1)
92 {
93     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
94     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_8);
95     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_10);
96     HAL_Delay(200);
97     /* USER CODE END WHILE */
98
99     /* USER CODE BEGIN 3 */
100 }
101 /* USER CODE END 3 */
102 }
103
104 */
105 * @brief RIF Initialization Function
106 * @param None
107 * @retval None
108 */
```

```
HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_8);
HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_10);
HAL_Delay(200);
```

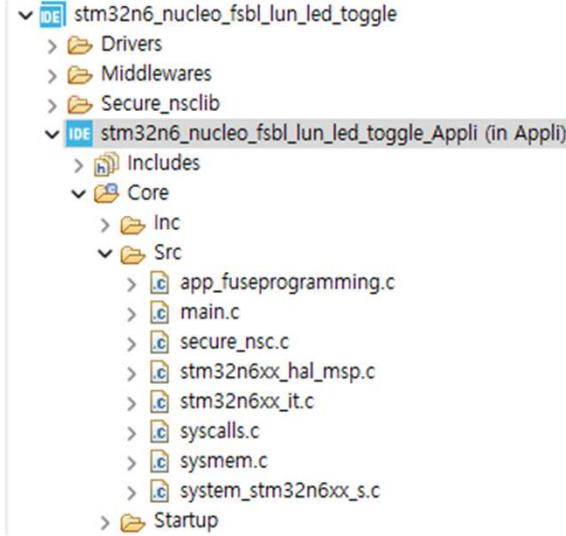
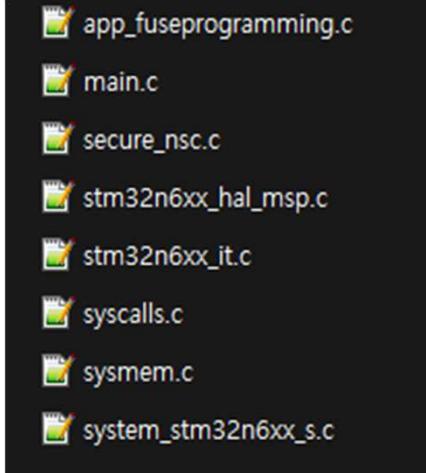
Insert code in **main.c** of the **Secure Application** to toggle the LEDs.



# Add Code to APP

Copy and paste the **app\_fuseprogramming.c** and **app\_fuseprogramming.h** files.

Hands\_On\1\_Hands\_On\_FSBL\_LRUN\stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\Appli\Core\Src\ app\_fuseprogramming.c  
Hands\_On\1\_Hands\_On\_FSBL\_LRUN\stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\Appli\Core\Inc\app\_fuseprogramming.h



# Add Code to APP

```
18  /* USER CODE END Header */
19  /* Includes -----
20  #include "main.h"
21
22  /* Private includes -----
23  /* USER CODE BEGIN Includes */
24  #include "app_fuseprogramming.h"
25  /* USER CODE END Includes */
26
```

```
#include "app_fuseprogramming.h"
```

```
83  /* Initialize all configured peripherals */
84  MX_GPIO_Init();
85  MX_BSEC_Init();
86  SystemIsolation_Config();
87  /* USER CODE BEGIN 2 */
88  Fuse_Programming();
89  /* USER CODE END 2 */
```

```
Fuse_Programming();
```



# Build APP

The screenshot shows the STM32CubeIDE interface with the following components:

- Project Explorer:** Shows the project structure under "stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle".
- Code Editor:** Displays the `main.c` file content.
- Build Console:** Shows the build logs for the target `stm32n6_nucleo_fsbl_lrun_led_toggle_Appli`.

**Project Explorer Content:**

- stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle (selected)
- Drivers
- Middlewares
- Secure\_nsclib
- stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_Appli (in Appli)

  - Includes
  - Core
    - Inc
    - Src
      - main.c
      - secure\_nsclib.c
      - stm32n6xx\_hal\_msp.c
      - stm32n6xx\_it.c
      - syscalls.c
      - sysmem.c
      - system\_stm32n6xx\_s.c
    - Startup
  - Drivers
    - STM32N657X0HXQ\_LRUN\_RAMxspi1.ld
    - STM32N657X0HXQ\_LRUN\_RAMxspi2.ld
    - STM32N657X0HXQ\_LRUN.ld**
    - STM32N657X0HXQ\_ROMxspi1\_RAMxspi2.ld
    - STM32N657X0HXQ\_ROMxspi1.ld
    - STM32N657X0HXQ\_ROMxspi1xspi2\_RAMxspi3.ld
    - STM32N657X0HXQ\_ROMxspi2\_RAMxspi1.ld
    - STM32N657X0HXQ\_ROMxspi2.ld
    - STM32N657X0HXQ\_ROMxspi3\_RAMxspi1xspi2.ld
  - stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSB (in FSB)
  - stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle.ioc

**Code Editor (main.c):**

```
28 ** in the root directory of this software component.
29 ** If no LICENSE file comes with this software, it is provided AS-IS.
30 **
31 ****
32 */
33
34 /* Entry Point */
35 ENTRY(Reset_Handler)
36
37 /* Highest address of the user mode stack */
38 _estack = ORIGIN(RAM) + LENGTH(RAM); /* end of "RAM" Ram type memory */
39 _sstack = _estack - _Min_Stack_Size;
40
41 _Min_Heap_Size = 0x200; /* required amount of heap */
42 _Min_Stack_Size = 0x800; /* required amount of stack */
43
44 /* Memories definition */
45 MEMORY
46 {
47     RAM      (xrw)      : ORIGIN = 0x34000400,    LENGTH = 2047K
48 }
49
50 /* Sections */
51 SECTIONS
52 {
53     /* The startup code into "RAM" Ram type memory */
54     .isr_vector :
55     {
56         . = ALIGN(4);
57         KEEP(*(.isr_vector)) /* Startup code */
58         . = ALIGN(4);
59     } >RAM
60
61     /* The program code and other data into "RAM" Ram type memory */
62     .text :
63     {
```

**Build Console Log:**

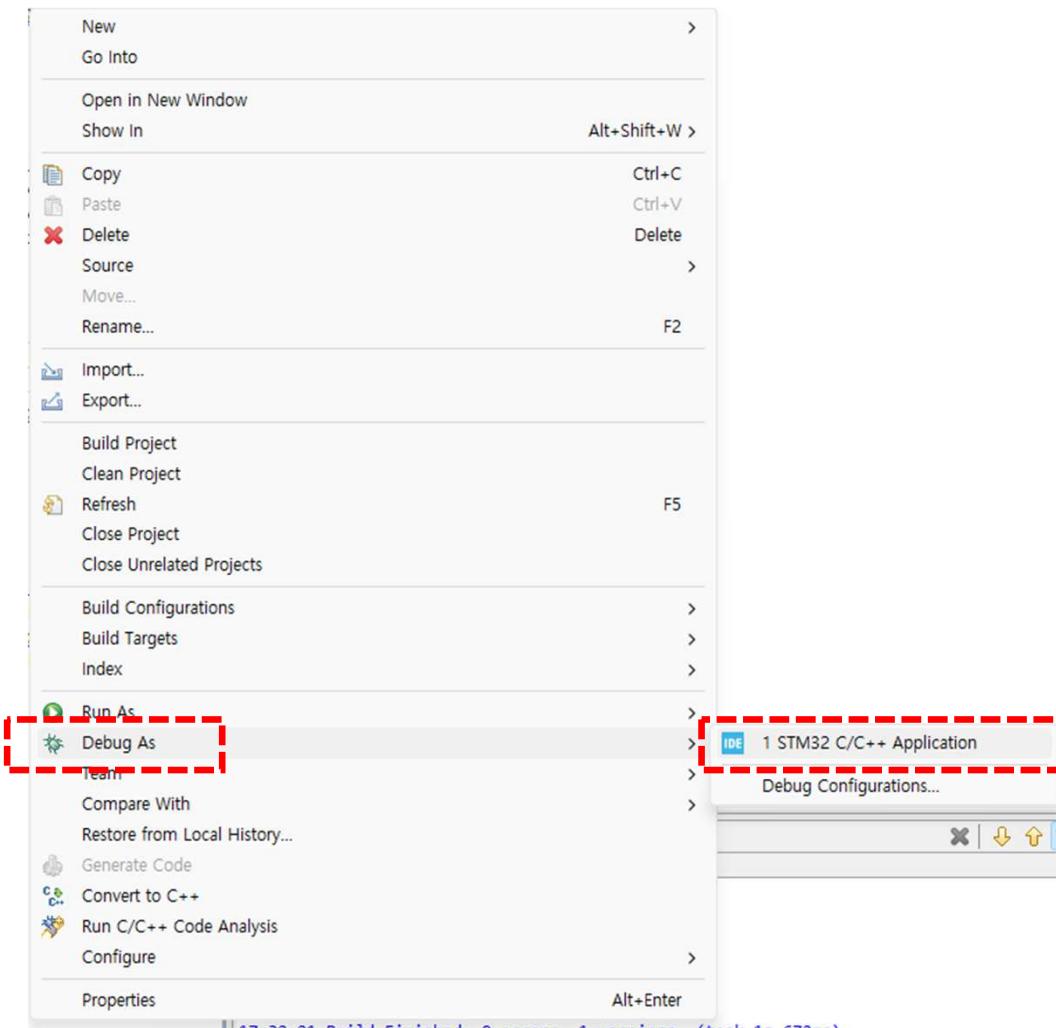
```
CDT Build Console [stm32n6_nucleo_fsbl_lrun_led_toggle_Appli]
arm-none-eabi-gcc ".../Core/Src/stm32n6xx_it.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDDEBUG
arm-none-eabi-gcc ".../Core/Src/syscalls.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDDEBUG
arm-none-eabi-gcc ".../Core/Src/sysmem.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDDEBUG
arm-none-eabi-gcc ".../Core/Src/system_stm32n6xx_s.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDDEBUG
arm-none-eabi-gcc -o "stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf" @"objects.list
C:/ST/STM32CubeIDE_1.18.0/STM32CubeIDE/plugins/com.st.stm32cube.ide.mcu.externaltargets/obj
Finished building target: stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf

arm-none-eabi-size stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf
arm-none-eabi-objdump -h -S stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf > "stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.h
4656           12       2604      7272      1c68 stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf
arm-none-eabi-objcopy -O binary stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf "stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.bin
Finished building: default.size.stdout

Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.list
Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.bin

18:33:42 Build Finished. 0 errors, 1 warnings. (took 3s.555ms)
```

# DEBUG APP



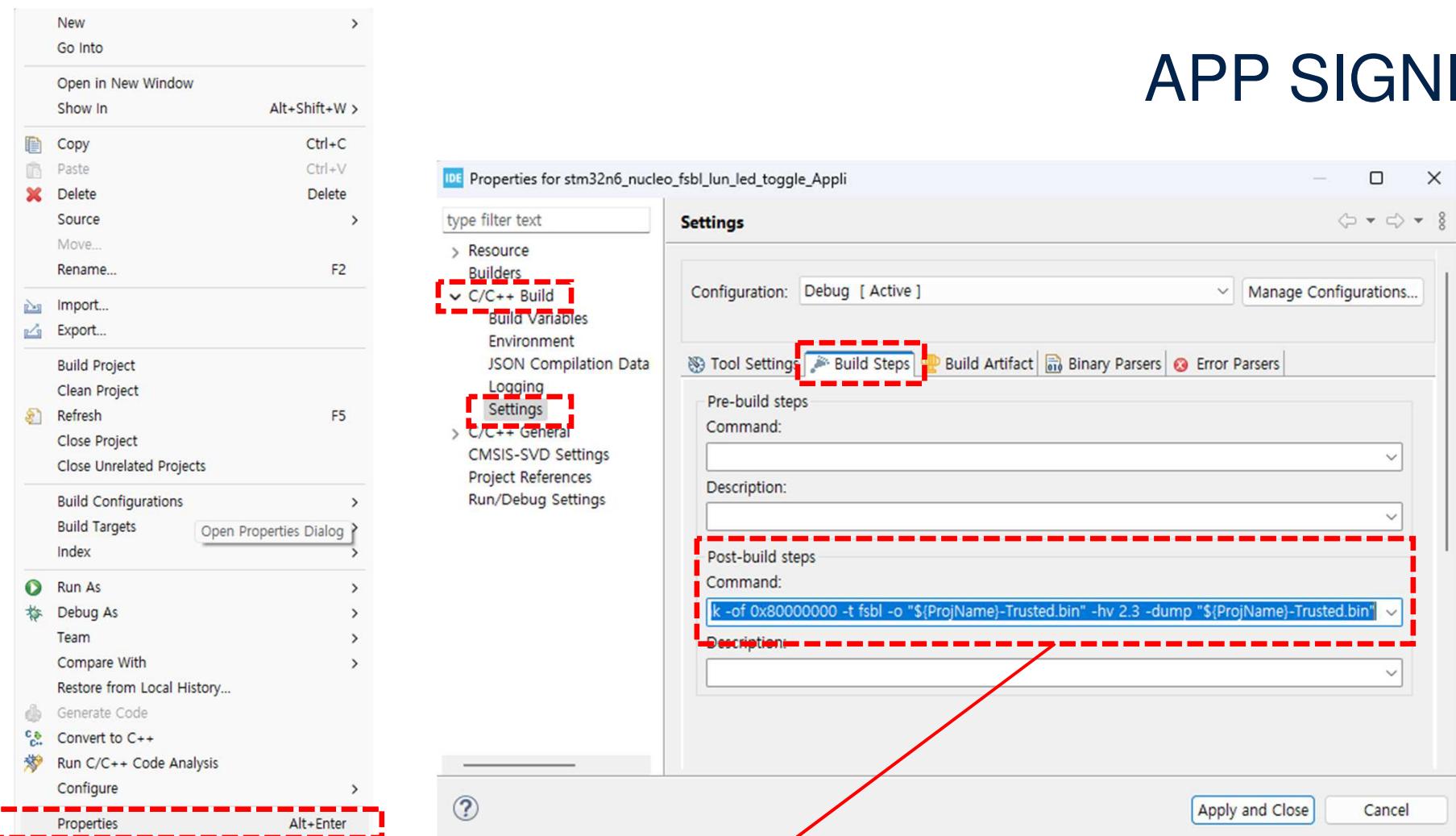
The screenshot shows the ST-Connect IDE during a debug session. The Project Explorer shows a project named "stm32n6\_nucleo\_fsbl\_lun\_led\_toggle\_App". The main.c file is open in the editor, showing the following code:

```
/* Enable the CPU Cache */
74 /* Enable I-Cache-----*/
75 SCB_EnableICache();
76 /* Enable D-Cache-----*/
77 SCB_EnableDCache();
78 /* MCU Configuration-----*/
79 HAL_Init();
80 /* USER CODE BEGIN Init */
81 /* USER CODE END Init */
82 /* USER CODE BEGIN SysInit */
83 /* USER CODE END SysInit */
84 /* Initialize all configured peripherals */
85 MX_GPIO_Init();
86 MX_BSEC_Init();
87 SystemIsolation_Config();
88 /* USER CODE BEGIN 2 */
89 /* Infinite loop */
90 /* USER CODE BEGIN WHILE */
91 while (1)
92 {
93     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
94     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_8);
95     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_10);
96     HAL_Delay(200);
97 }
98 /* USER CODE END 2 */
99
100
101
102
103
104
105
106
107
108
```

The debugger view shows a thread list with "Thread #1 [main] 1 [core: 1] (Suspended : Breakpoint)" and a stack trace at main.c:76. The GDB connection is shown as "arm-none-eabi-gdb (14.2.90.20240526)". The ST-LINK connection is shown as "ST-LINK (ST-LINK GDB server)". A red dashed box highlights the "Run As" and "Debug As" options in the context menu.



# APP SIGNING

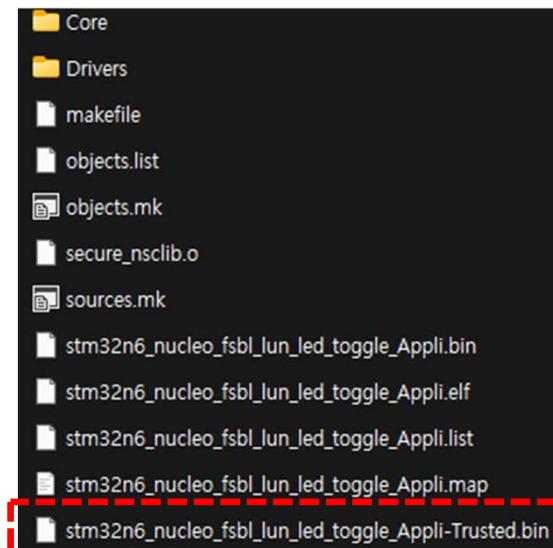


```
cd "${ProjDirPath}/Debug" && echo y | "C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe" -bin "${ProjName}.bin" -nk -of 0x80000000 -t fsbl -o "${ProjName}-Trusted.bin" -hv 2.3 -dump "${ProjName}-Trusted.bin"
```

# APP SIGNING

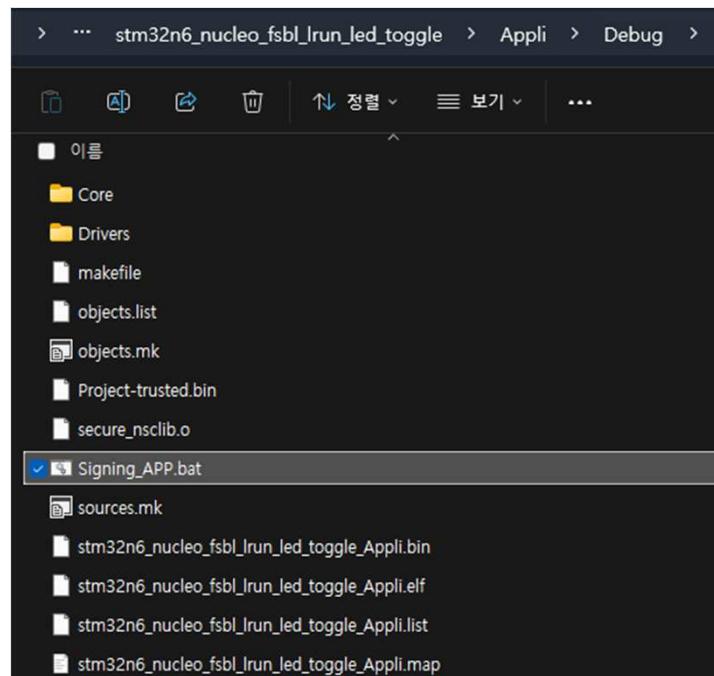
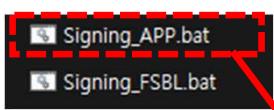
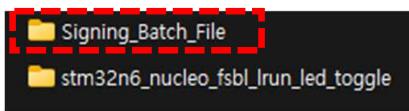
```
-----  
STM32 Signing Tool v2.19.0  
-----  
  
Adding head padding bytes for payload with header v2.3  
Header version 2.3 preparation...  
Extracting Entry point value from the input file...  
Entry point value : 0x34000d7d  
The headed image file generated successfully: stm32n6_nucleo_fsbl_lun_led_toggle_Appi-Trusted.bin  
  
Header description:  
  
Magic: 0x53544d32  
Signature: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
Checksum: 0x566b3  
Header version: 0x20300  
Size: 0x1000  
Load address: 0xffffffff  
Entry point: 0x34000d7d  
Image version: 0x0  
Extension: 0x80000000  
  
Pad header detected:  
Type: 0x5354ffff  
Size: 0x1a0  
Padding values: ca 86 c4 ed 21 c3 b7 63 29 6a 1e 21 c5 82 da 38 e9 09 68 83 48 b3 61 00 07 b4 27 b6 77 78 6d  
f9 28 db 6f 3d 32 3f 87 03 43 1f e9 17 11 ba 59 a7 4a 02 e7 6d 81 5d 42 c3 c9 8f 24 d6 c5 e7  
8f dc df 31 f3 ed 79 3f 00 77 d2 fd 9c 3d a7 4b bb 56 2a 9a 09 0b 83 01 7a f3 bf d4 7a 29 3c  
d1 2c f2 9d bb e7 d6 b1 fc d1 e4 13 87 0c 9f 6d 9c 77 a5 eb f6 b8 17 c8 c5 94 90 3d e7 e4 0c  
8f 82 ba 13 96 64 76 32 f2 3a f8 6a 4c 7f cb 69 9f 68 a9 e3 bb 9a 10 e5 e1 63 b5 91 ed c2 f2  
90 c1 08 1b e1 24 ab dc a6 a8 cd 82 ae 0f ba db 83 01 16 59 8b 69 85 60 27 70 2e 5a e4 44 b8  
90 3b 41 53 71 70 81 49 89 48 c0 54 c9 bb 93 05 39 a0 a7 fd 66 be 1e ad 2c 4c bd d2 da e3 b8  
df 2a 31 56 66 71 9e 42 b2 06 09 d8 18 aa 2f 48 c0 44 41 f4 0b 96 37 a9 83 60 0b d1 ff 98  
d6 45 5c c1 9c af 11 85 db e7 d7 1f 93 51 d8 b8 a3 05 b7 87 fd 40 4c f0 db 0b 31 ea f1 3d 99  
8e 93 51 57 2b 79 8f fb e0 2c cb 5f 99 a1 a5 1f 9d 64 da 3d 46 51 52 6c 04 31 de b1 c7 7d 6d  
95 79 e3 95 7d 5a 95 9d 7e 36 8e 4a 2d 89 48 82 a7 cf ca 9b 36 47 73 74 23 ae d4 f8 c4 b7 05  
39 23 8f 7f 25 c7 6c 1a e2 21 81 5f 0c 5d 62 45 79 84 1c 42 66 70 5e 54 88 b3 0c 40 80 11 4d  
a9 6c 9a f3 91 7c d8 10 00 00 02 00 fd f8 dd c9 e2 90 00 00 80 ba 0b e0 74 02 00 00 40 b7 0b  
e0 74 02 00 00  
  
Post header 2.3 information :  
Binary type: 0x10  
Non authenticated payload length: 0x0  
Non authenticated payload hash: 0x0
```

16:53:27 Build Finished. 0 errors, 1 warnings. (took 1s.616ms)



# APP SIGNING

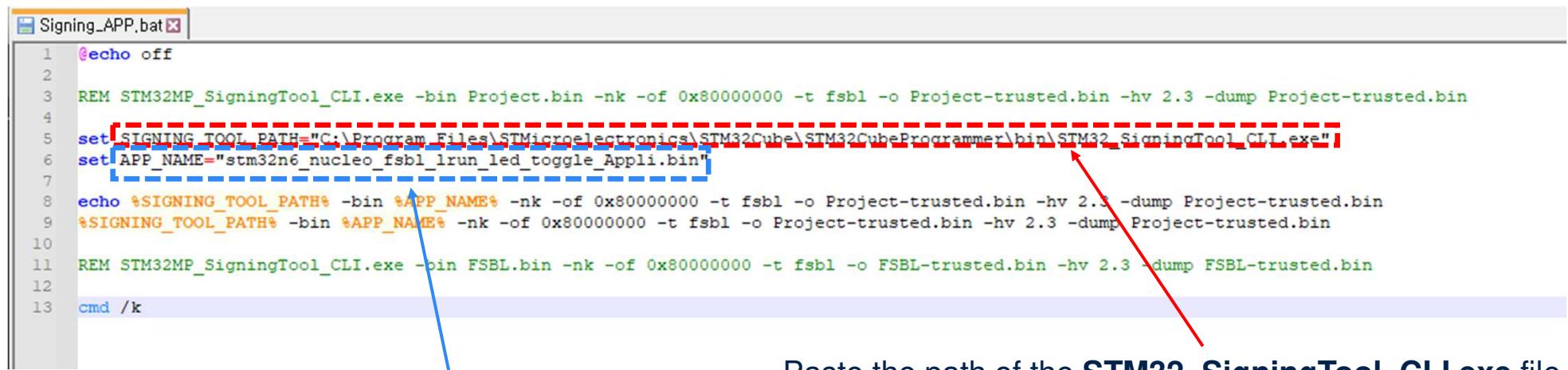
It is also possible to use the Signing Tool externally.



Copy the **Signing\_FSBL.bat** file from the **Siging\_Batch\_File** folder into the **FSBL\Debug** folder of the current project.



# APP SIGNING



```
1 @echo off
2
3 REM STM32MP_SigningTool_CLI.exe -bin Project.bin -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -dump Project-trusted.bin
4
5 set SIGNING_TOOL_PATH="C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe"
6 set APP_NAME="stm32n6_nucleo_fsbl_lrun_led_toggle_Applic.bin"
7
8 echo %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.1 -dump Project-trusted.bin
9 %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -dump Project-trusted.bin
10
11 REM STM32MP_SigningTool_CLI.exe -bin FSBL.bin -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
12
13 cmd /k
```

Paste the path of the **STM32\_SigningTool\_CLI.exe** file from **STM32CubeProgrammer**.

After building the **Secure Application**, paste the name of the generated **binary file**.



# APP SIGNING

```
C:\WINDOWS\system32\Wcar > + <
"C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe" -bin "stm32n6_nuc
toggle_App.lib.bin" -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -dump Project-trusted.bin

STM32 Signing Tool v2.19.0

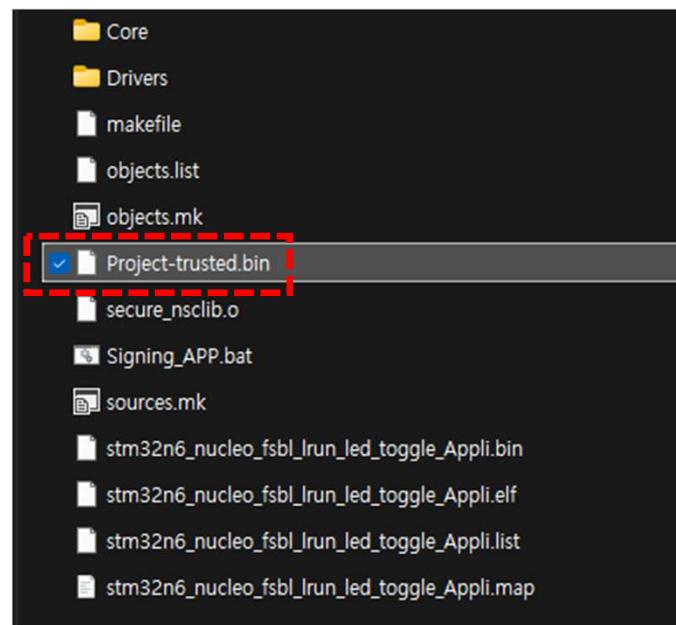
Adding head padding bytes for payload with header v2.3
Header version 2.3 preparation...
Extracting Entry point value from the input file...
Entry point value : 0x340000de9
The headed image file generated successfully: Project-trusted.bin

Header description:

Magic: 0x53544d32
Signature: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Checksum: 0x65e6a
Header version: 0x20300
Size: 0x1400
Load address: 0xffffffff
Entry point: 0x340000de9
Image version: 0x0
Extension: 0x80000000

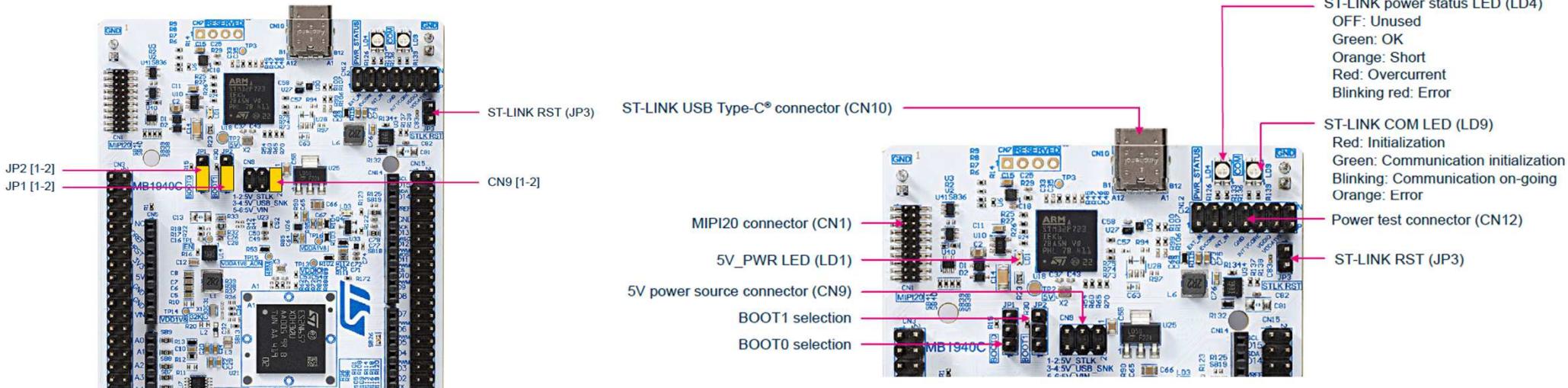
Pad header detected:
Type: 0x5354ffff
Size: 0x1a0
Padding values: c9 1d ea 24 56 a6 77 77 58 fd c2 fb 09 02 50 8e da 90 33 12 96 3e 38 74 9e 8e 16 06 3f 88 dc
72 f1 db df 8f 1d ba 27 4a e4 c3 7a c5 aa 11 43 9e bd 8f fe 21 45 b6 b7 ef 2b 2d 42 96
47 15 a8 4c 3d 60 4b 2a 11 d1 17 b3 c0 72 ad 9b 06 d0 a4 54 85 68 58 c6 86 6b 8d 40 83 0c 0d
0c 32 56 d2 c6 61 d0 16 e1 1d 89 c2 c3 91 f7 4f 5a 82 9e 53 74 56 2a 31 8e 30 bf 0a 87 a6 dc
ce 4c 17 94 61 79 cc 45 25 63 ce 43 21 2d 80 1b e4 f7 8d b2 a8 8a 25 60 92 1e bc bd db 64 42
1c 42 4b b3 5e be 90 19 97 97 8c 58 df 9e 37 21 68 a6 9c 4b 00 bf d3 d3 ff bb 69 fb de d7 68
0c ae d4 59 47 97 5f ed ef 2e 04 8e 09 31 78 17 8d ca 70 e4 14 27 c3 8e 3e 8b fc ed 1e 08 b3
59 bb 0c 12 ab 5e 41 7f 42 c4 89 ab 4d b9 f7 0d 78 83 d8 a0 89 3b e0 ab ef bf c6 66 3a cc 58
c8 86 e0 dd 5b 77 aa fd 25 e0 22 a8 e2 57 1c 4b 27 01 90 43 24 65 ad 99 fe b3 16 88 22 e8 d8
15 83 b4 10 98 86 d2 e6 28 0a b4 7a f1 36 90 00 08 d5 09 e8 32 4e 5a 47 8d 13 24 a0 37 4f 31
58 ce ac 7c d8 8a a4 04 9d 96 86 9c a4 ea c3 91 96 70 59 90 13 f3 1b 8b 02 ad d2 f4 8a 83 c2
9d a2 8c 9b 78 06 20 5f 37 7d 4a ba fe 53 e5 83 1e dc 3e d3 6f c3 23 9e b9 8c 26 df 19 a1 85
30 49 f3 bb f5 6c 1b 58 00 00 02 00 ca 19 32 cc 0b a5 00 00 60 dc c7 4d 7a 02 00 00 40 63 b7
4d 7a 02 00 00

Post header 2.3 information :
Binary type: 0x10
Non authenticated payload length: 0x0
Non authenticated payload hash: 0x0
```



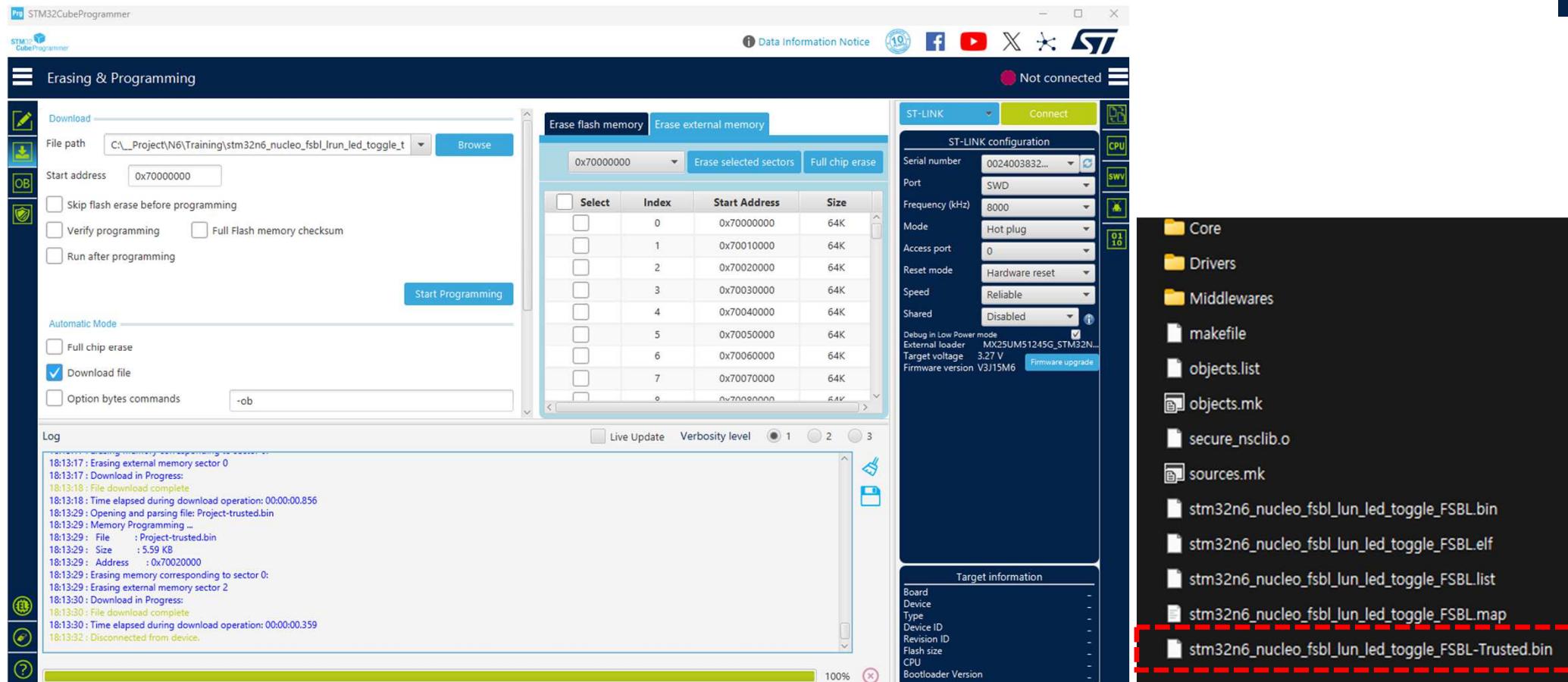
The signed file “**Project-trusted.bin**” is generated upon completing the signing process.

# FSBL and APP Download



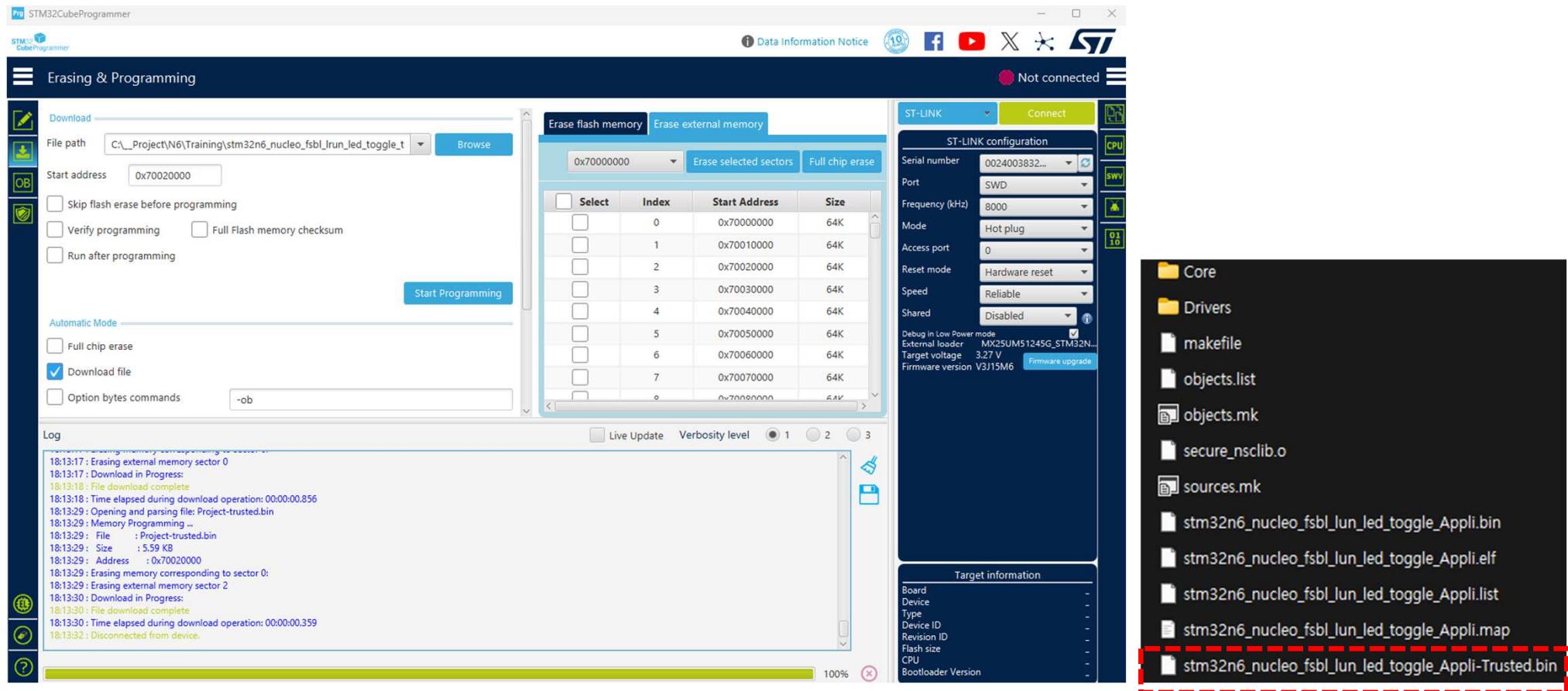
Jumper	Definition	Default position	Comment
CN9	5V power selection (user USB power source selection)	[1-2]	5V from STLINK-V3EC
JP1	BOOT0 selection	[1-2]	Boot pin flash/serial selection
JP2	BOOT1 selection	[1-2]	Boot in flash mode
JP3	STLK_RST	OFF	-

# FSBL Download



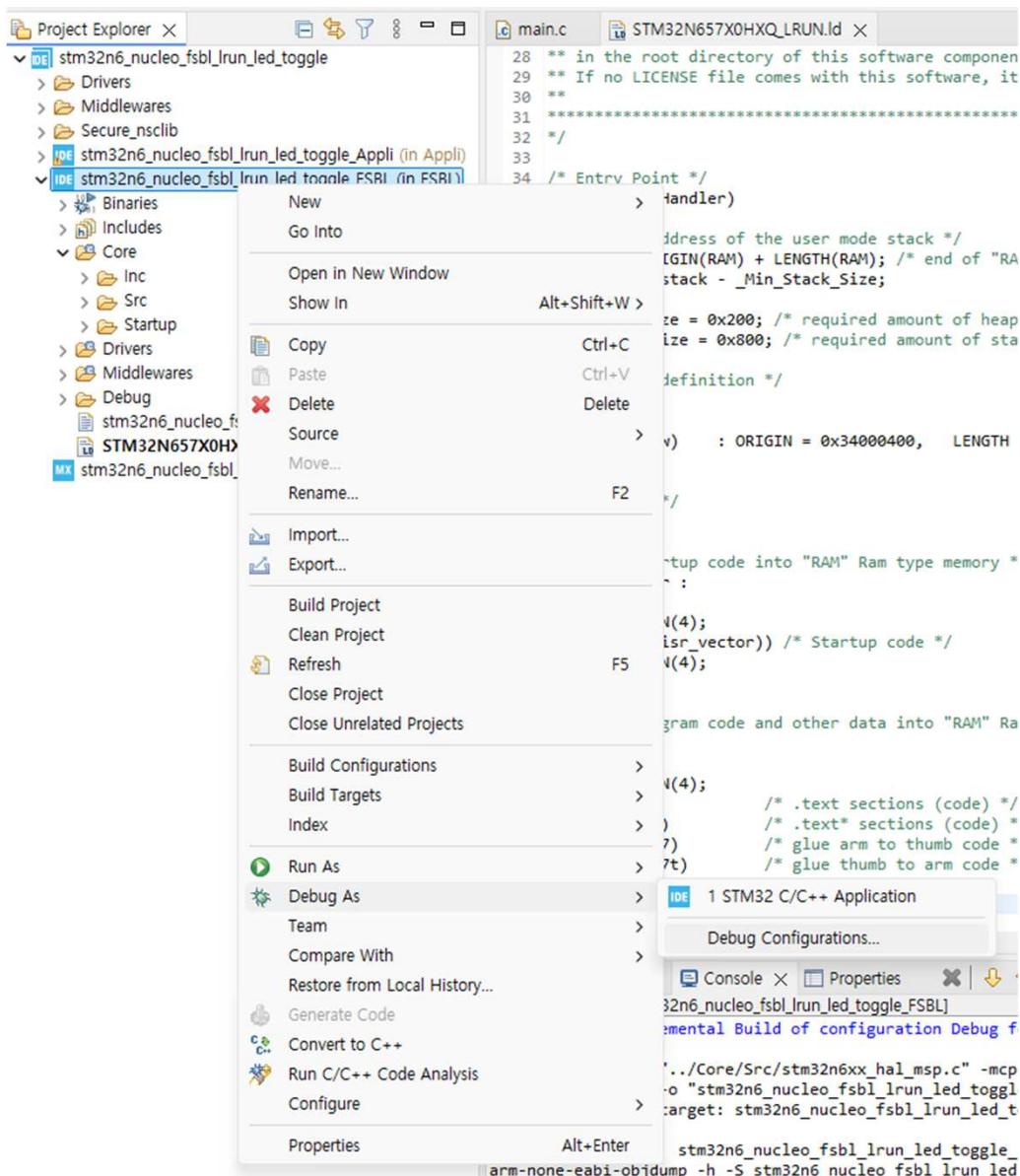
Write the **stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL-Trusted.bin** file to **0x70000000**.

# APP Download

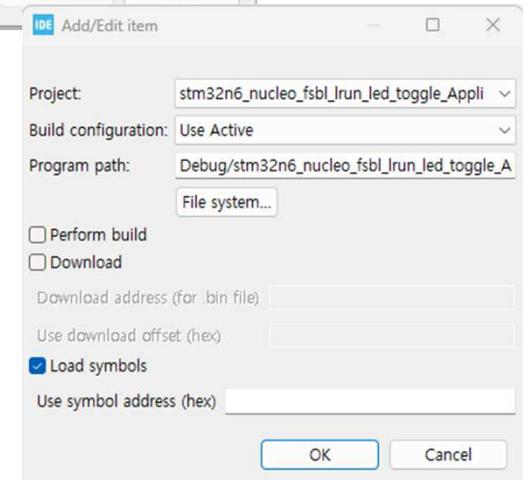
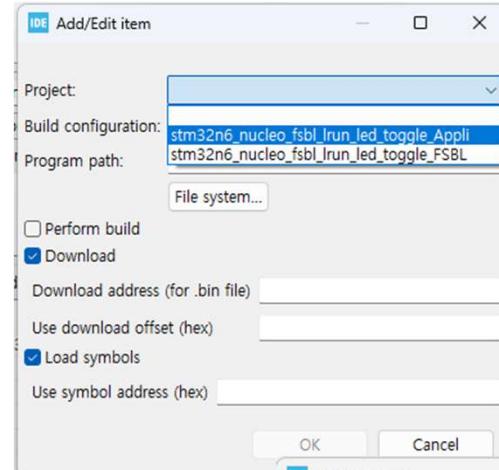
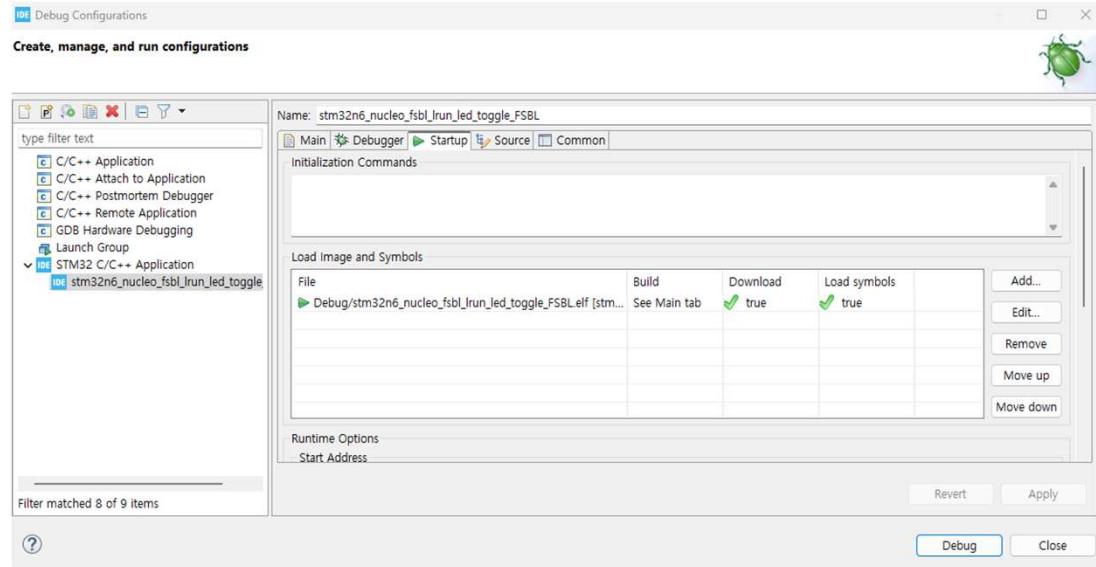


Write the **stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_Appi-Trusted.bin** file to **0x70020000**.

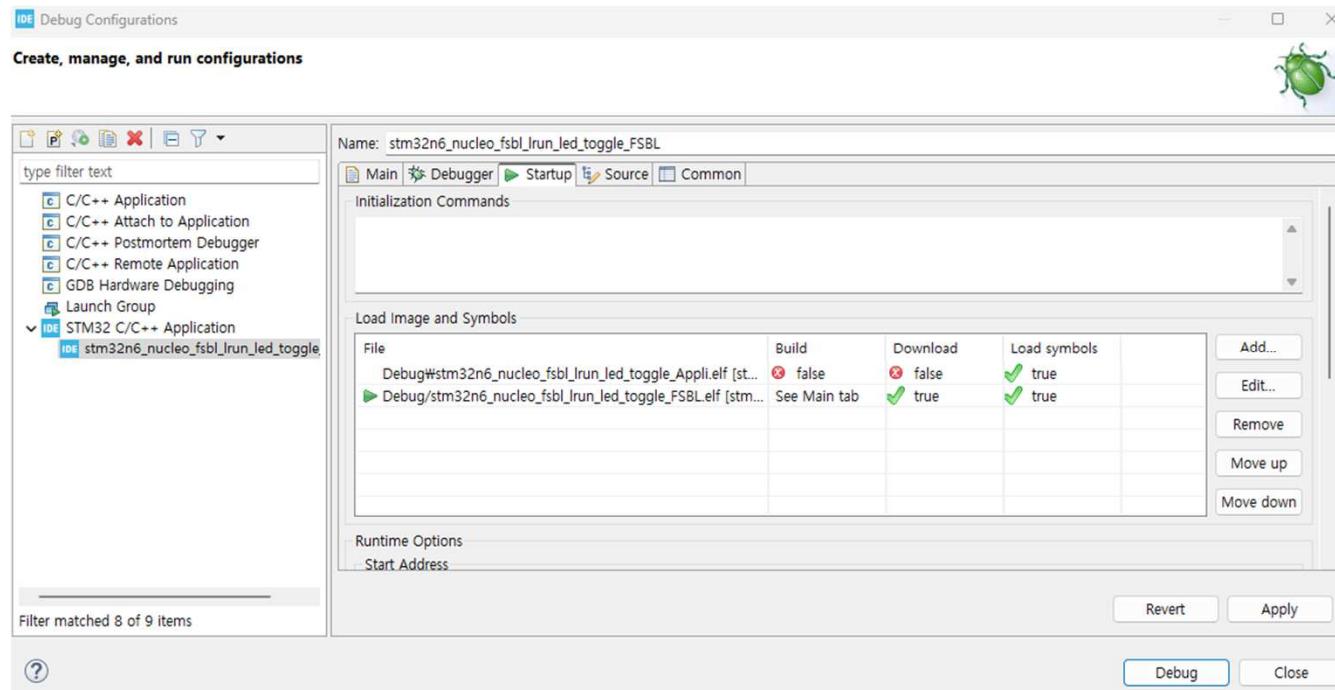
# LRUN Debug



# LRUN Debug



# LRUN Debug



# LRUN Debug

The screenshot shows the ST-Connect IDE interface during a debug session. The Project Explorer on the left lists the project and its components. The main window displays the source code of `main.c` with line numbers and assembly code on the right. The assembly code shows the execution flow, including calls to HAL functions like `HAL_GPIO_TogglePin`. The assembly code is color-coded to match the source code, highlighting the same lines.

```
File Edit Source Refactor Navigate Search Project Run Window Help
Debug X Project Explorer
stm32n6_nucleo_fsb_lrun_led_toggle_FSB [STM32 C/C++ Application]
  Thread #1 [main] 1 [core: 1] (Suspended : Signal : SIGINT:Interrupt)
    HAL_GetTick() at stm32n6xx_hal.c:312 0x34000f04
    HAL_Delay() at stm32n6xx_hal.c:389 0x34000f48
    main() at main.c:96 0x34000816
    Reset_Handler() at startup_stm32n657x0hxq.s:99 0x34000e24
arm-none-eabi-gdb (14.2.90.20240526)
ST-LINK (ST-LINK GDB server)

main.c x 0x18003a1a main.c stm32n6xx_hal.c »1
  /* USER CODE BEGIN 1 */
  ...
  /* USER CODE END 1 */
  /* MCU Configuration----*/
  HAL_Init();
  ...
  /* USER CODE BEGIN Init */
  ...
  /* USER CODE END Init */
  ...
  /* USER CODE BEGIN SysInit */
  ...
  /* USER CODE END SysInit */
  ...
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  SystemIsolation_Config();
  /* USER CODE BEGIN 2 */
  ...
  /* USER CODE END 2 */
  ...
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
    HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_8);
    HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_10);
    HAL_Delay(200);
    /* USER CODE END WHILE */
    ...
    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
  ...
34000801: ldr r0, [pc, #24] @ (0x3400081c <main+56
34000803: bl 0x340014c4 <HAL_GPIO_TogglePin>
34000807: mov.w r1, #1024 @ 0x400
3400080b: ldr r0, [pc, #16] @ (0x3400081c <main+56
3400080d: bl 0x340014c4 <HAL_GPIO_TogglePin>
34000811: movs r0, #200 @ 0xc8
34000813: bl 0x34000f1c <HAL_Delay>
93      HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
34000816: nop
34000818: b.n 0x340007f4 <main+16>
3400081a: nop
3400081c: adds r0, r0, r0
3400081e: ldrsb r2, [r0, r0]
110     {
SystemIsolation_Config:
34000820: push {r7, lr}
34000822: add r7, sp, #0
117      __HAL_RCC_RIFSC_CLK_ENABLE();
34000824: mov.w r0, #512 @ 0x200
34000828: bl 0x3400078c <LL_AHB3_GRP1_EnableClock>
122      HAL_GPIO_ConfigPinAttributes(GPIOB,GPIO_PIN_12
3400082c: movw r2, #769 @ 0x301
34000830: mov.w r1, #4096 @ 0x1000
34000834: ldr r0, [pc, #48] @ (0x34000868 <SystemI:
34000836: bl 0x340014f8 <HAL_GPIO_ConfigPinAttribut:
123      HAL_GPIO_ConfigPinAttributes(GPIOG,GPIO_PIN_0,
3400083a: movw r2, #769 @ 0x301
3400083e: movs r1, #1
34000840: ldr r0, [pc, #40] @ (0x3400086c <SystemI:
34000842: bl 0x340014f8 <HAL_GPIO_ConfigPinAttribut:
124      HAL_GPIO_ConfigPinAttributes(GPIOG,GPIO_PIN_8,
34000846: movw r2, #769 @ 0x301
3400084a: mov.w r1, #256 @ 0x100
3400084e: ldr r0, [pc, #28] @ (0x3400086c <SystemI:
34000850: bl 0x340014f8 <HAL_GPIO_ConfigPinAttribut:
125      HAL_GPIO_ConfigPinAttributes(GPIOG,GPIO_PIN_10
...
```

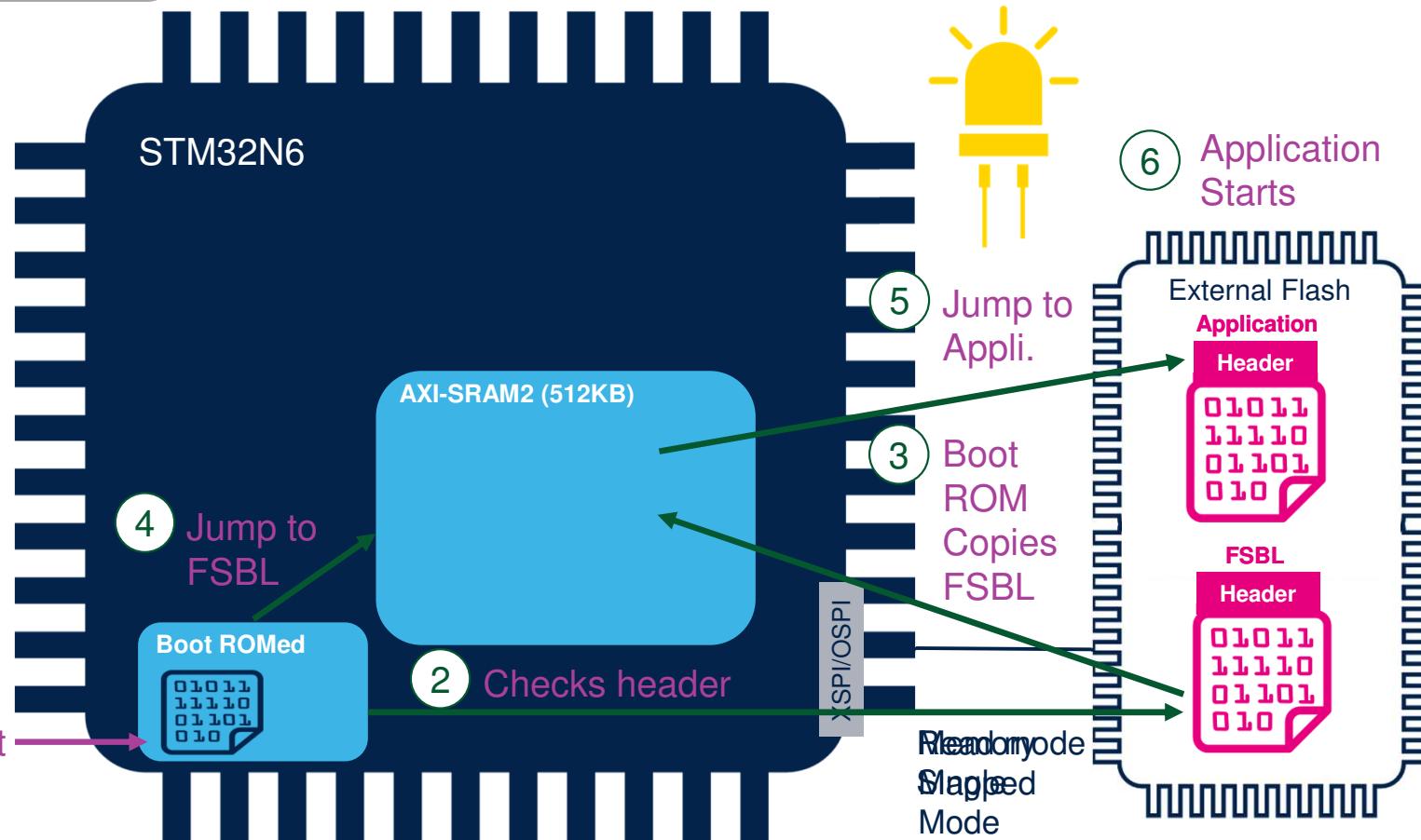


# **Hands-On: FSBL + XIP**





# Application in execution in place config



# CubeMX Configuration

The image shows the STM32CubeMX software interface for creating a new project. On the left, a 'New Project' wizard lists three options: 'Start My project from MCU', 'Start My project from ST Board', and 'Start My project from Example'. The first option is highlighted with a red circle labeled '1' and a red dashed box around the 'ACCESS TO MCU SELECTOR' button. A large red arrow points from this button to the 'MCU/MPU Selector' interface on the right. In the selector, a dropdown menu under 'Commercial Part Number' is highlighted with a red circle labeled '2' and a red dashed box. The selected part number is 'STM32N657X0H3Q'. Below the dropdown is a table titled 'MCUs/MPUs List: 1 item' showing the selected part. A red circle labeled '3' and a red dashed box highlight the row for 'STM32N657X0H...'. An arrow points from this row to the text 'Double Click' at the bottom right.

New Project

I need to :

Start My project from MCU  
1 ACCESS TO MCU SELECTOR

Start My project from ST Board  
ACCESS TO BOARD SELECTOR

Start My project from Example  
ACCESS TO EXAMPLE SELECTOR

STM32N657X0H3Q

MCU/MPU Selector   Board Selector   Example Selector   Cross Selector

MCU/MPU Filters

Commercial Part Number: STM32N657X0H3Q

PRODUCT INFO

Segment, Series, Line, Marketing Status, Price, Package, Core, Coprocessor

MEMORY

Flash = 0 (kBytes)

MCUs/MPUs List: 1 item

Commercial Part No.	Part No.	Reference	Marketing St...	Unit Price for
STM32N657X0H...	STM32N657X0	STM32N657X0...	Active	10.7254

Features   Block Diagram   Docs & Resources   CAD

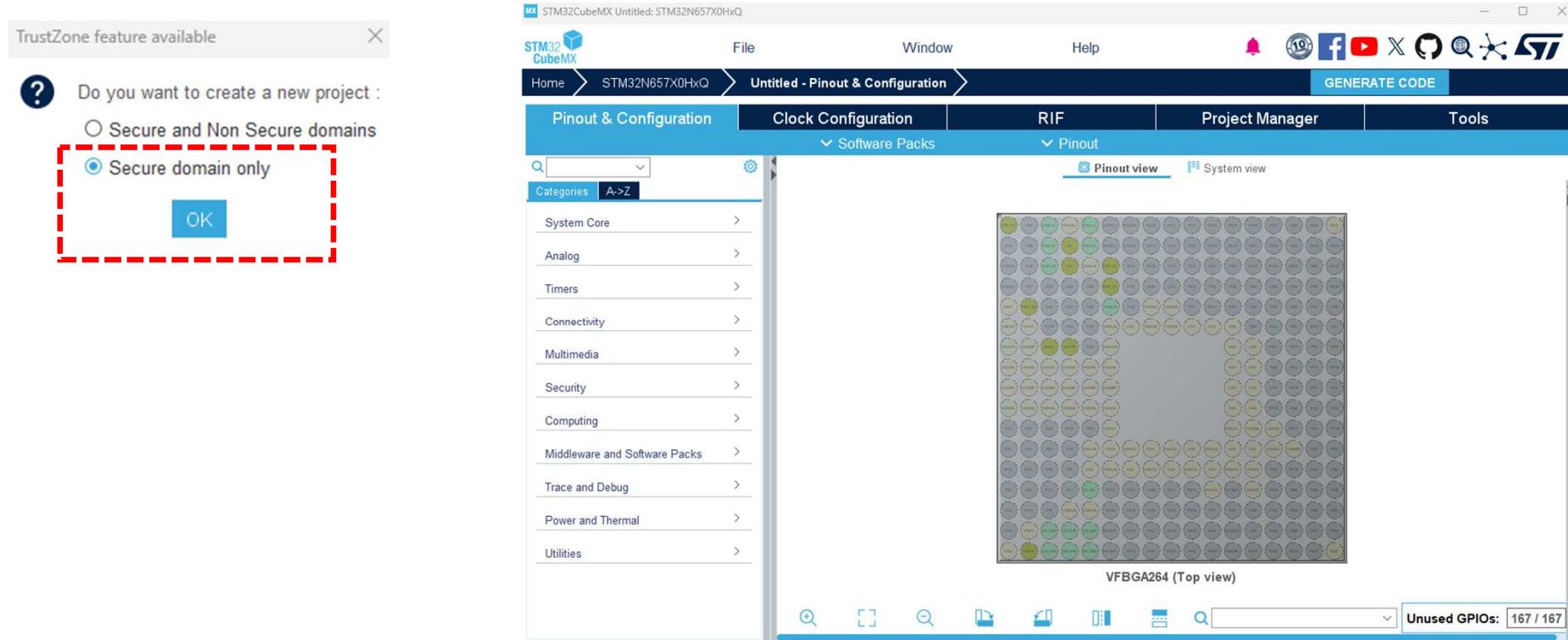
The first high-performance STM32 MCU with AI capabilities

Double Click



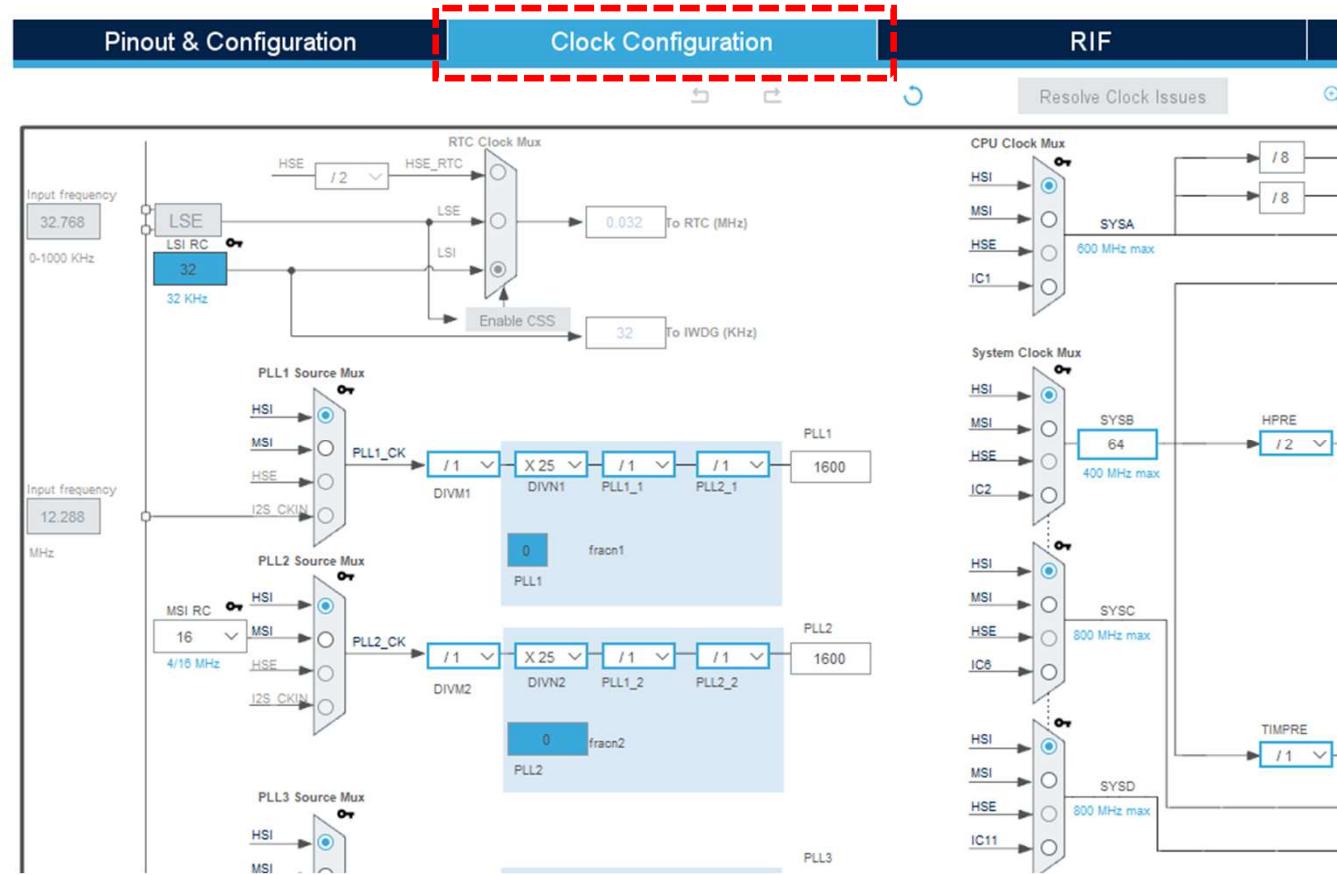
# CubeMX Configuration

In this hands-on, select **Secure domain only**.



# Clock Configuration

Click the **Clock Configuration** tab.



## Electrical characteristics

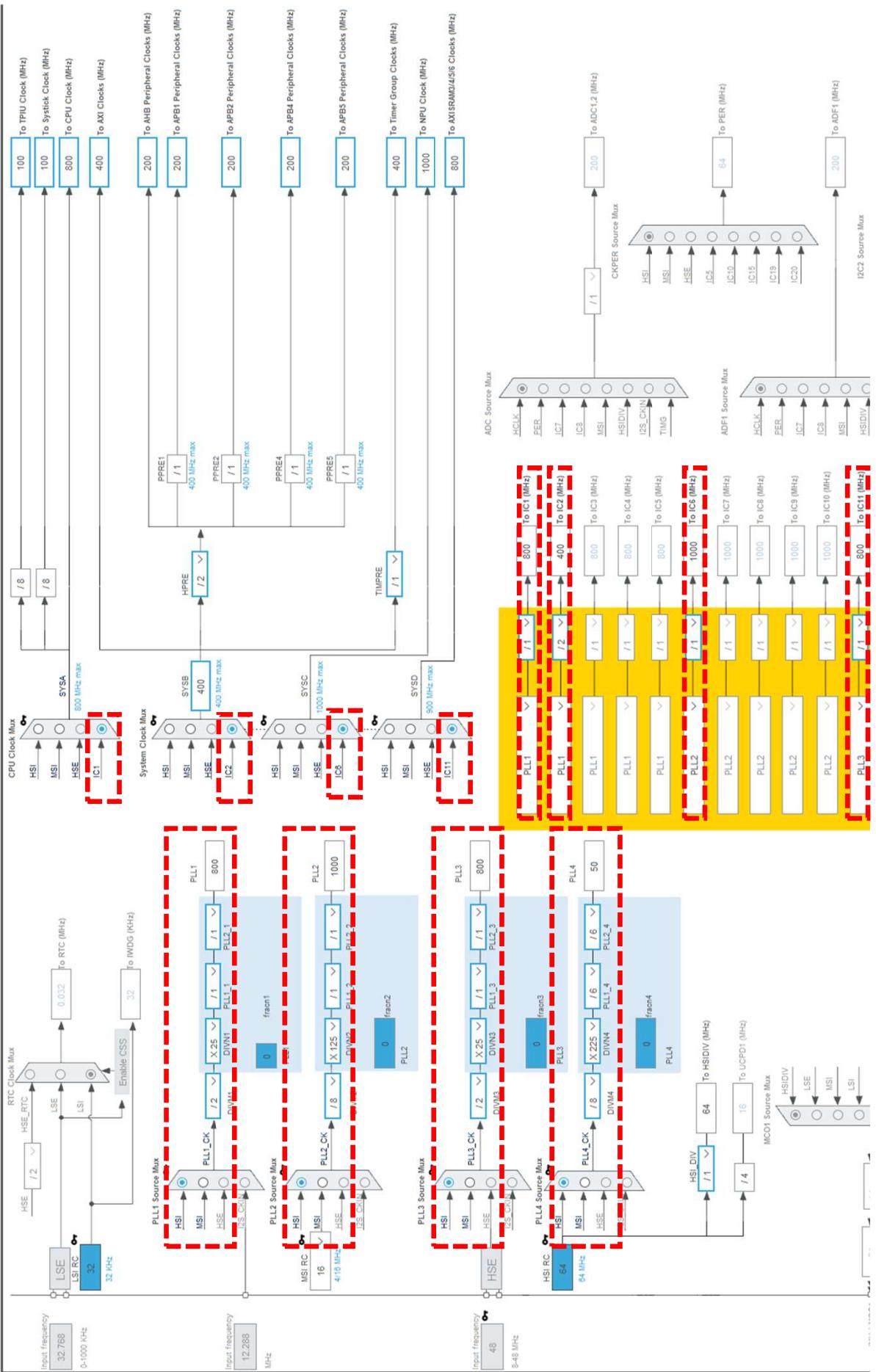
STM32N647xx STM32N657xx

### 5.3 Operating conditions

#### 5.3.1 General operating conditions

Table 22. General operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$F_{CPU}$	Clock frequency of Cortex-CM55 (VOS low)	Base-TCM with 0 wait states Flex-TCM with 1 wait state	0	-	600	MHz
$F_{CPU\ overdrive}$	Clock frequency of Cortex-CM55 in overdrive (VOS high)		0	-	800	
$F_{NPU}$	Clock frequency of NPU/CNN		0	-	800	
$F_{NPU\ overdrive}$	Clock frequency of NPU/CNN in overdrive		0	-	1000	
$F_{ck\_icn\_hsl}$	Clock frequency of USB, ETH buses	-	0	-	400	
$F_{HCLK}$	Clock frequency of AHB bus	-	0	-	200	
$F_{ck\_cpu\_axi}$	Clock frequency of AXI CPU bus	-	0	-	400	
$F_{PCLKx}$	Clock frequency of APB buses ( $x = 1, 2, 3, 4, 5$ )	-	0	-	$F_{HCLKx} / 4$	



# CORTEX\_M55\_FSBL Configuration

Pinout & Configuration      Clock Configuration      RIF

Software Packs      Pinout

CORTEX\_M55\_FSBL Mode and Configuration

Mode

Runtime contexts:

First Stage Boot Loader	Application	External Memory Loader
<input checked="" type="checkbox"/>		

Configuration

Reset Configuration

Parameter Settings      User Constants

Configure the below parameters :

Search (Ctrl+F)     

**Cortex Interface Settings**

CPU ICACHE	Enabled
CPU DCACHE	Enabled

**Cortex Memory Protection Unit Control Settings**

MPU Control Mode	MPU NOT USED
------------------	--------------

Analog >

The screenshot shows the configuration interface for the CORTEX\_M55\_FSBL. The left panel, 'Pinout & Configuration', displays a table of system components and their status across FSBL, Application, and ExtMemLoader modes. The right panel, 'Clock Configuration', shows mode and configuration settings for runtime contexts. A red dashed box highlights the 'Cortex Interface Settings' section under 'Configure the below parameters'.



# RCC Configuration

The screenshot shows three tabs: Pinout & Configuration, Clock Configuration, and RIF.

**Pinout & Configuration:** This tab displays a hierarchical tree of system components. A red dashed box highlights the 'RCC' row under the 'System Core' section, which contains checkboxes for FSBL, Application, and ExtMemLoader.

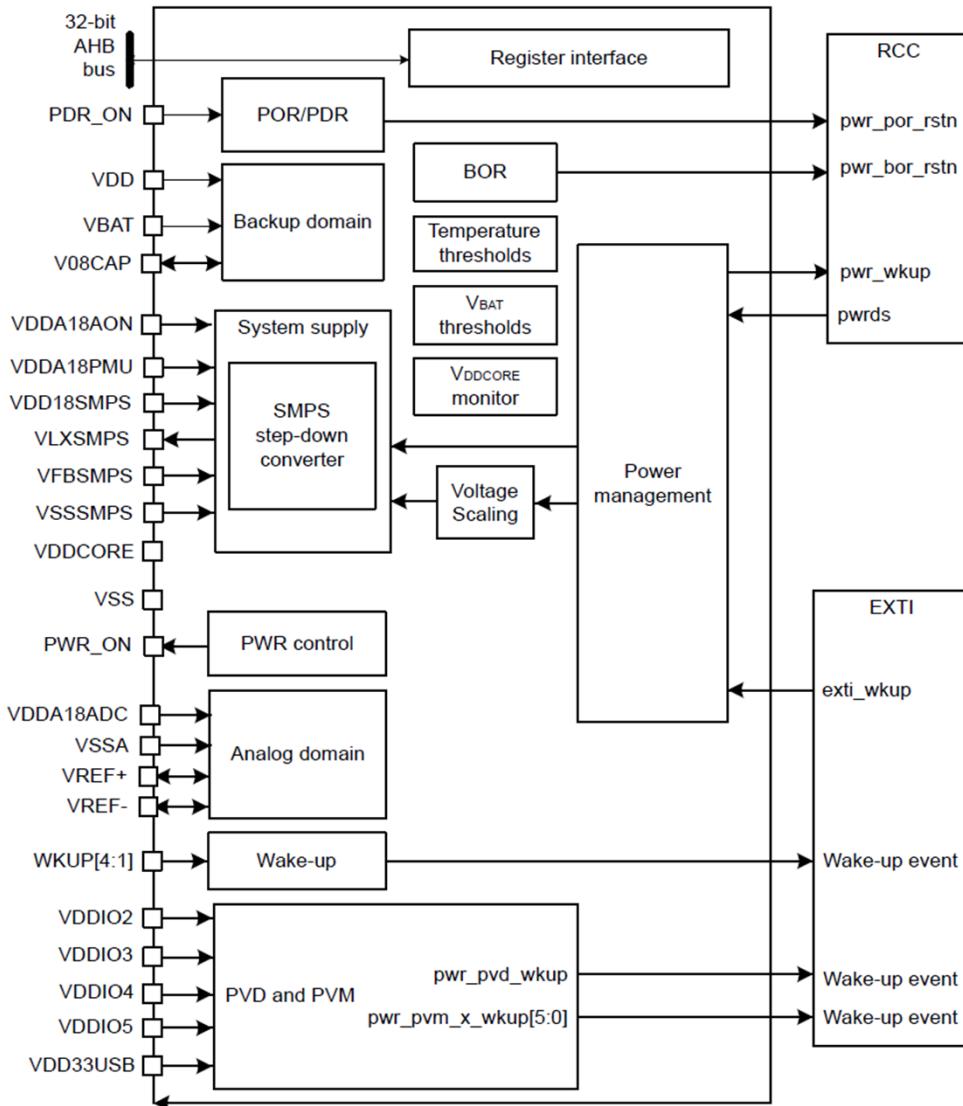
	FSBL	Application	ExtMemLoader
CORTEX_M55_FSBL	<input checked="" type="checkbox"/>		
CORTEX_M55_S		<input checked="" type="checkbox"/>	
GPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
GPIO			
HPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICACHE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NVIC1_S_Application		<input checked="" type="checkbox"/>	
NVIC_FSBL	<input checked="" type="checkbox"/>		
RAMFC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>RCC</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SYS_S	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
WWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Clock Configuration:** This tab shows RCC Mode and Configuration settings. A red dashed box highlights the 'Power Parameters' section, which includes fields for SupplySource and Power Regulator Voltage Scale.

Mode	First Stage Boot Loader	Application	External Memory Loader
High Speed Clock (HSE)	Disable		
Low Speed Clock (LSE)	Disable		
Master Clock Output 1			
Master Clock Output 2			
Audio Clock Input (I2S_CKIN)			

**RIF:** This tab is partially visible at the top right.

**Figure 9. Power control block diagram**



The boot ROM code uses multiple supplies for I/Os:

**VDDIO3:** Independent I/O supply 3 (PN[12:0]), configured for XSPI1M\_P2 (XSPI).

**VDDIO4:** Independent I/O supply 4 (PC[1], PC[12:6], and PH[2,9]), configured for eMMC.

**VDDIO5:** Independent I/O supply 5 (PC[0], PC[5:2], and PE[4]), configured for SD card.

**VDD33USB:** Supply input for USB HS PHYs and USB Type-C® PHY 3V3.

These supplies can be independent of VDD and can be monitored with peripheral voltage monitoring. The voltage range configuration is described in the PWR section of the reference manual (RM0486).

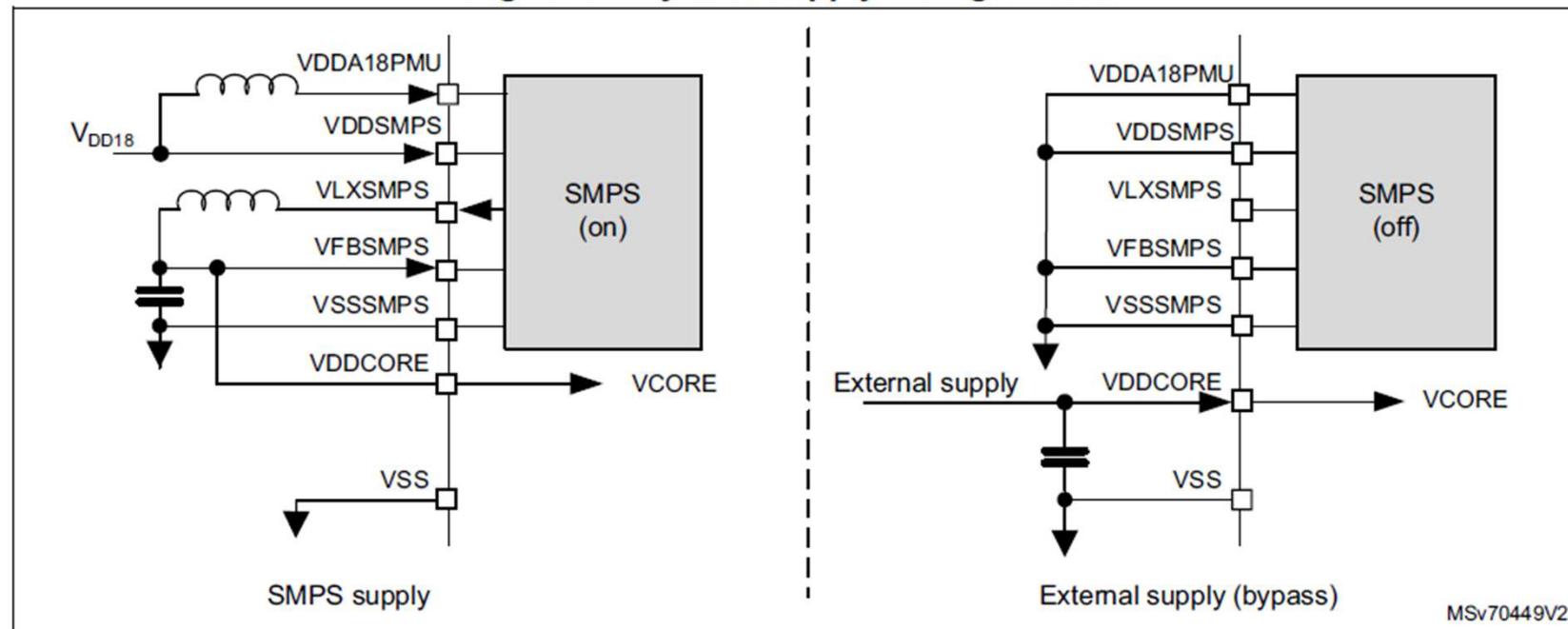
# Power supply

RM0486

Power control (PWR)

By configuring the SMPS step-down converter, the supply configurations shown in [Figure 16](#) are supported for the V<sub>CORE</sub> domain.

**Figure 16. System supply configurations**

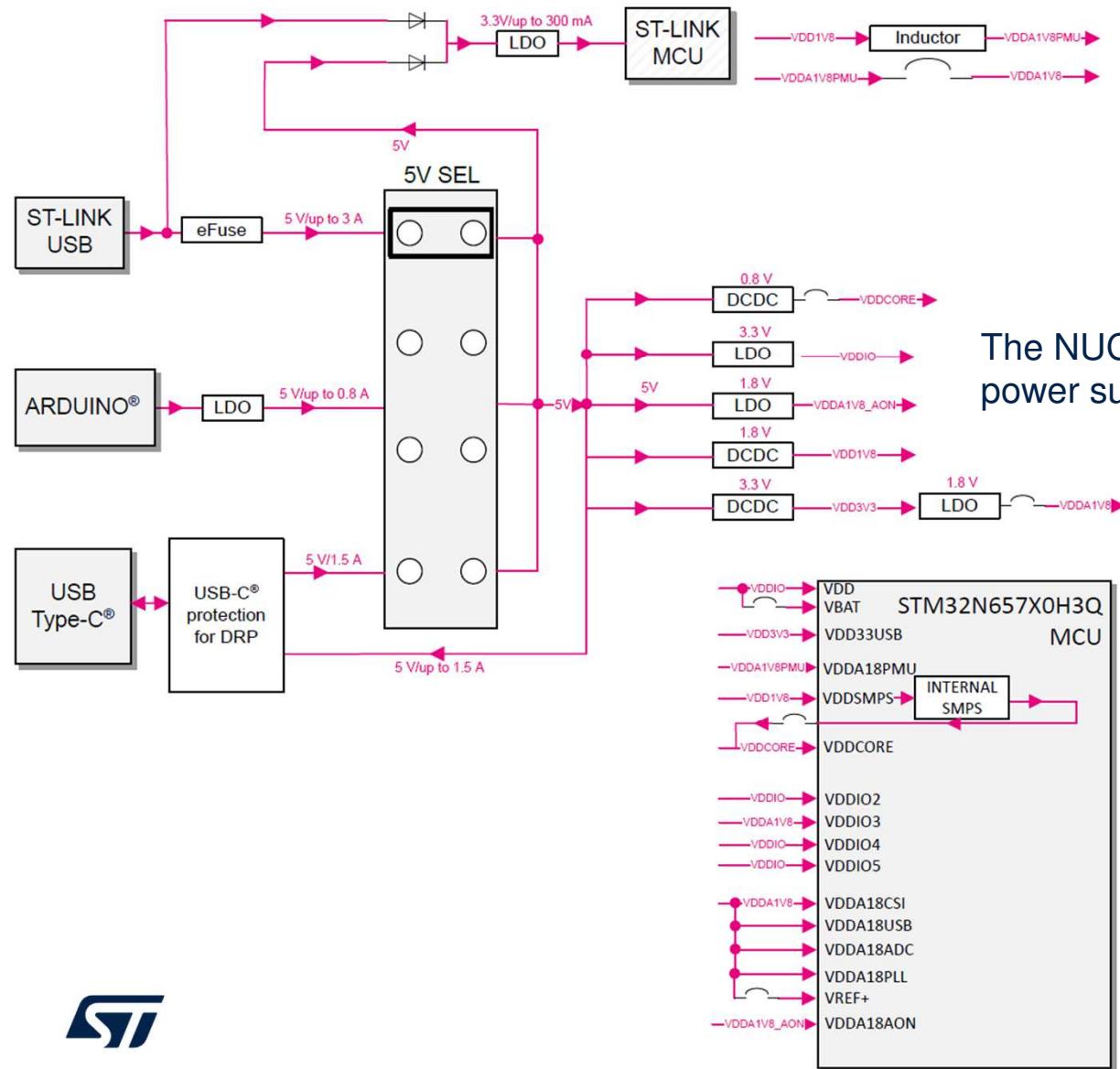


# Power supply

$V_{DD}^{(1)}$	I/Os supply voltage	1.8 V range	1.62	1.8	1.98	V
		3.3 V range	3.0	3.3	3.6	
$V_{DDA18ON}^{(1)}$	Internal analog supply voltage	-	1.71	1.8	1.935	
$V_{DDIOx}$	Specific I/Os supply voltage (x = 2, 3, 4, 5)	1.8 V range	1.71	1.8	1.935	
		3.3 V range	2.7	3.3	3.6	
$V_{DDCORE}$	Main digital logic supply voltage	SoC Run mode (VOS low)	0.782	0.81	0.842	
		SoC Run mode (VOS high)	0.858	0.89	0.921	
		Sleep mode (SoC Run mode, peripheral clock stopped, VOS low)	0.782	0.81	0.842	
		Sleep mode (SoC Run mode, peripheral clock stopped, VOS high)	0.858	0.89	0.921	
		Stop mode (SVOS low)	0.64	0.68	0.71	
		Stop mode (SVOS high)	0.782	0.81	0.842	
$V_{DDA18PLL}$	1.8 V analog supply for PLL	$F_{NPU}$ range	1.62	1.8	1.98	
$V_{DDA18CSI}$	1.8 V analog supply for CSI	$F_{NPU}$ overdrive range	1.746	1.8	1.98	
$V_{DDCSI}$	CSI operating voltage	-	0.784	0.81	0.842	
$V_{DD18USB}$	1.8 V analog supply for USBPHY	-	1.746	1.8	1.935	
$V_{DD18ADCx}$	ADC operating voltage (x = 1, 2)	-	1.62	1.8	1.98	
$V_{REF+}$	ADC reference voltage	-	1.1	-	$V_{DD18ADC}$	
$V_{BAT}$	Backup operating voltage	-	TBD	-	3.6	



# Power supply



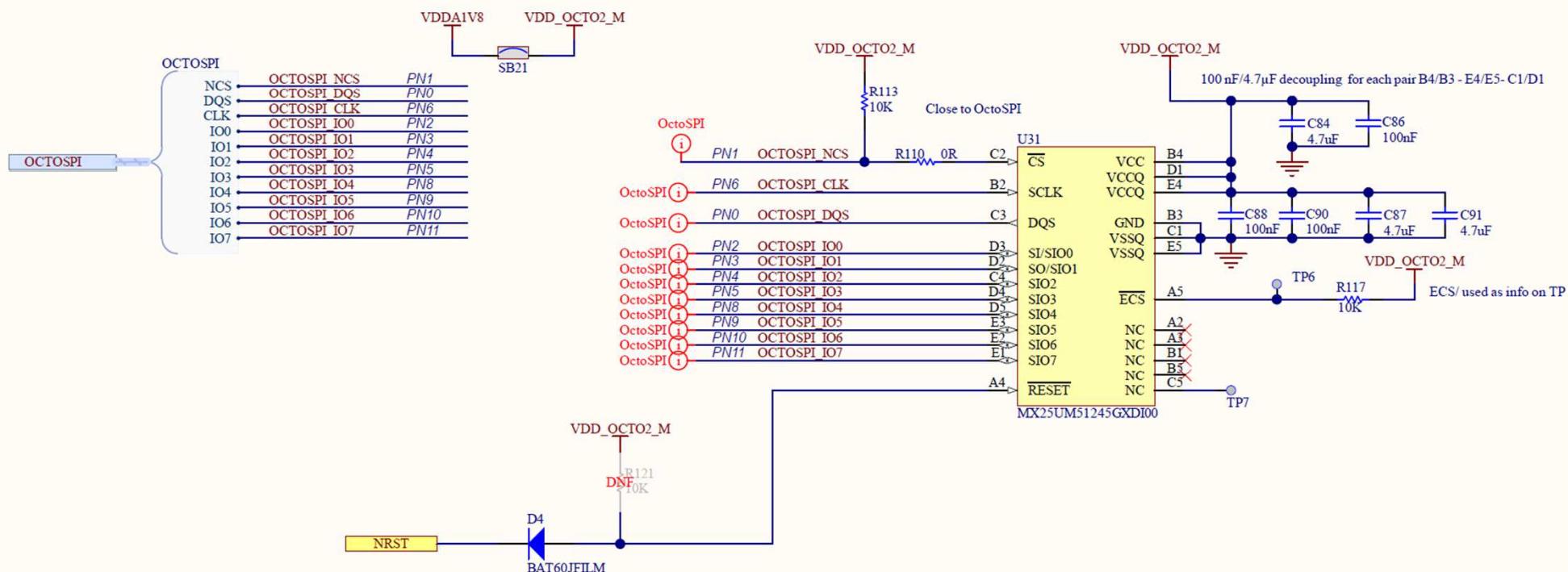
The NUCLEO board uses **External DCDC** as the default power supply.



# Octo-SPI Flash Memory Configuration

## Octo-SPI flash memory

The Octo-SPI flash memory has the following characteristics: 512 Mbits, 1.8 V, 200 MHz, DTR, read while writing. It is connected to the Octo-SPI interface of the STM32N657X0H3Q microcontroller. The embedded footprint is also compatible with many other references in the BGA24 package. Check the compatibility of the memory datasheet versus MB1940 schematics.





# Octo-SPI Flash Memory Configuration

Table 16. Pin description (continued)

Pin number								Pin name (function after reset)	Pin type	I/O structure	Notes	Alternate functions	Additional functions				
VFBGA142	VFBGA169	VFBGA178	VFBGA198	VFBGA223	VFBGA264	H15	J11	L13	M15	P19	R15	PN2	I/O	-	-	XSPIM_P2_IO0(boot), FMC_A23, EVENTOUT	-
						K15	H12	N13	M15	P19	R15	PN3	I/O	-	-	XSPIM_P2_IO1(boot), FMC_A22, EVENTOUT	-
						E14	L12	J13	J14	V19	N17	PN4	I/O	-	-	XSPIM_P2_IO2(boot), EVENTOUT	-
						F15	L13	K14	K13	U18	R16	PN5	I/O	-	-	XSPIM_P2_IO3(boot), EVENTOUT	-
						G15	K12	K12	K15	U19	P15	PN6	I/O	-	-	XSPIM_P2_CLK(boot), EVENTOUT	-
						F14	K11	K13	K14	R16	T16	PN7	I/O	-	-	XSPIM_P2_NCLK(boot), EVENTOUT	-
						E15	M13	J14	J15	V18	P16	PN8	I/O	-	-	XSPIM_P2_IO4(boot), EVENTOUT	-
						G14	K13	L14	L13	T18	T15	PN9	I/O	-	-	XSPIM_P2_IO5(boot), DCMIPP_D5/DCMI_D5/PSSI_D5, EVENTOUT	-
						H14	J12	M14	L15	R18	U15	PN10	I/O	-	-	XSPIM_P2_IO6(boot), LCD_B4, EVENTOUT	-
						J14	H11	N14	M14	P18	U16	PN11	I/O	-	-	XSPIM_P2_IO7(boot), LCD_B6, EVENTOUT	-
						K14	M12	J12	L12	W18	P14	PN12	I/O	-	-	XSPIM_P2_NCS2, EVENTOUT	-

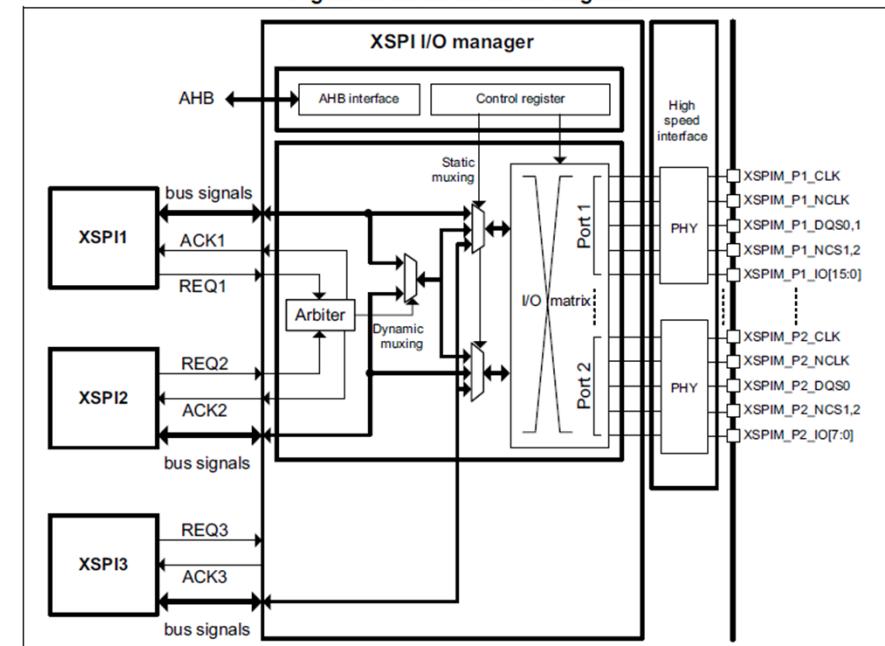
# Octo-SPI Flash Memory Configuration

The screenshot shows the Pinout & Configuration interface. In the left sidebar, under 'Categories' and 'A-Z', 'XSPIM' is selected. A red dashed box highlights the 'XSPIM' row. In the main area, the 'Clock Configuration' tab is active. Under 'Software Packs', the 'XSPIM Mode and Configuration' section is shown. The 'Mode' dropdown is set to 'Direct (XSPI1 to Port1; XSPI2 to Port2; XSPI3 not used)'. A red dashed box highlights this mode selection. Below it, other options like 'Disable', 'Swapped', and 'Multiplexed' are listed.

Select **FSBL** as the **Runtime Context** for **XSPIM**.

Select **Direct** as the **Port Mode** for **XSPI**.

Figure 215. XSPIM block diagram



**Pinout & Configuration**

**Clock Configuration**

**RIF**

**Pinout**

**XSPI2 Mode and Configuration**

**Mode**

Runtime contexts:

- First Stage Boot Loader
- Application
- External Memory Loader

Mode Octo SPI

Port Port2\_Octo

HyperBus/T/M 1.8V Inverted Clock Disable

Chip Select Override NCS1 - Port2 -

**NVIC Settings**

**Parameter Settings**

**DMA Settings**

**GPIO Settings**

**User Constants**

Configure the below parameters :

Search (Ctrl+F)

Configure the below parameters :

Generic

Fifo Threshold 4

Memory Mode Disable

Memory Type Macromix

Memory Size 1 GBits

Chip Select High Time Cycle 1

Free Running Clock Disable

Clock Mode Low

Wrap Size Not Supported

Clock Prescaler 0

Sample Shifting None

Delay Hold Quarter Cycle Enable

Chip Select Boundary Disabled

Maximum Transfer 0

Refresh Rate 0

Memory Select NCS1

Switching Duration Clock Number 1

Connectivity

	FSBL	Application	ExtMemLoader
ETH1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FDCAN1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FDCAN2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FDCAN3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FMC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2C1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2C2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2C3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I2C4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I3C1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I3C2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LPUART1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MDIOS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SDMMC1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SDMMC2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SPI6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UCPDI	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USART10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USB1_OTG_HS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USB2_OTG_HS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XSPI2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ZSIF	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NSPIM	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

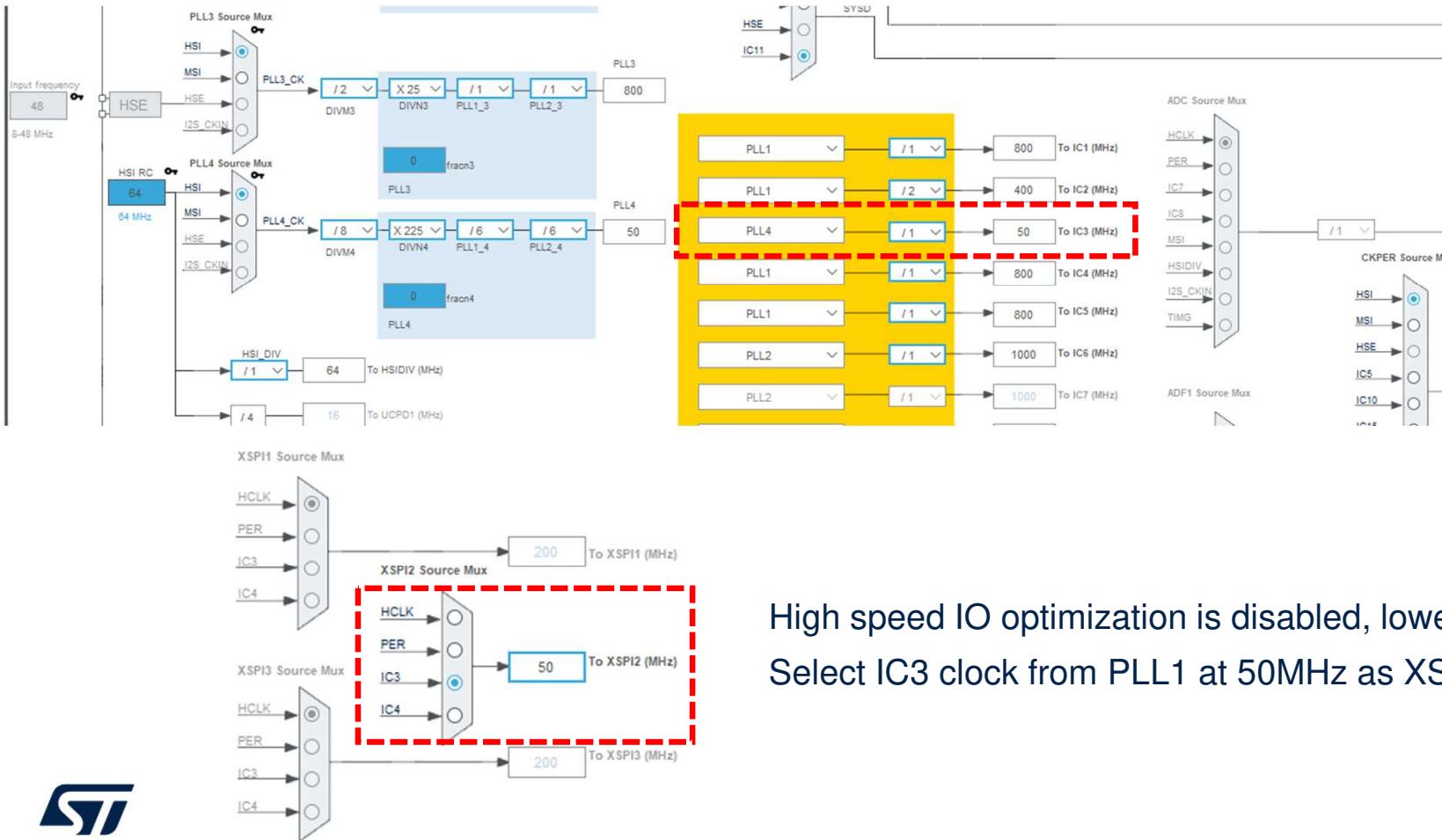


# Octo-SPI Flash Memory Configuration

OTP124	-	HCONF1	-
	[0]	IWDG1_HW	IWDG1 start on reset
	[1]	IWDG1_FZ_STOP	IWDG1 freeze in Stop mode
	[2]	IWDG1_FZ_STANDBY	IWDG1 freeze in Standby mode
	[9:3]	Reserved	Reserved
	[10]	RST_STOP	Reset caused if the device is put in Stop mode
	[11]	RST_STDBY	Reset caused if the device is put in Standby mode
	[12]	SELINBORH	- 0: BOR disabled - 1: BOR = 2.7 V
	[13]	HSLV_VDDIO5	VDDIO5 I/O segment below 2.5 V for I/O mode. The I/O segment is used by SDMMC2 port.
	[14]	HSLV_VDDIO4	VDDIO4 I/O segment below 2.5 V for I/O mode (I/O segment used by SDMMC1 port)
	[15]	HSLV_VDDIO3	VDDIO3 I/O segment below 2.5 V for I/O mode (I/O segment used by XSPIM port 2)



# Octo-SPI Flash Memory Configuration



High speed IO optimization is disabled, lower XSPI clock speed.  
Select IC3 clock from PLL1 at 50MHz as XSPI2 source.



# OTP Configuration (Optional)

The screenshot shows the STM32CubeProgrammer software interface for managing OTP memory. The title bar reads "STM32CubeProgrammer". The main window is titled "OTP MPU". The left sidebar has icons for File, Project, STM32CubeProgrammer, and a lock icon labeled "OTP". The main area displays a table of OTP settings:

Name	Word	Value	Status	Lock	Description
▶ Reserved	OTP121	0x011D6C6D	0x40000000	<input checked="" type="checkbox"/>	Permanent write lock
▶ Reserved	OTP122	0x95C0077C	0x40000000	<input checked="" type="checkbox"/>	Permanent write lock
▶ Reserved	OTP123	0xAE087777	0x40000000	<input checked="" type="checkbox"/>	Permanent write lock
▼ HCONF1	OTP124	0x00018000	0x00000000	<input type="checkbox"/>	-
	IWDG1_HW	0x0			IWDG1 start on reset
	IWDG1_FZ_STOP	0x0			IWDG1 freeze in Stop mode
	IWDG1_FZ_STAN...	0x0			IWDG1 freeze in Standby mode
	RST_STOP	0x0			Reset caused if the device is put in Stop mode
	RST_STANDBY	0x0			Reset caused if the device is put in Standby mode
	SELINBORH	0x0			0: BOR disabled. 1: BOR = 2.7 V
	HSLV_VDDIO5	0x0			VDDIO5 I/O segment below 2.5 V for I/O mode. The I/O segment is used by SDMMC2 port.
	HSLV_VDDIO4	0x0			VDDIO4 I/O segment below 2.5 V for I/O mode (I/O segment used by SDMMC1 port)
	HSLV_VDDIO3	0x1			VDDIO3 I/O segment below 2.5 V for I/O mode (I/O segment used by XSPIM port 2)
	HSLV_VDDIO2	0x1			VDDIO2 I/O segment below 2.5 V for I/O mode (I/O segment used by XSPIM port 1)
	HSLV_VDD	0x0			Main I/O segment below 2.5 V for I/O mode

At the bottom, there are buttons for "Export", "Save OTP partition", "Lock all" (with a checkbox), "Apply", and "Read".

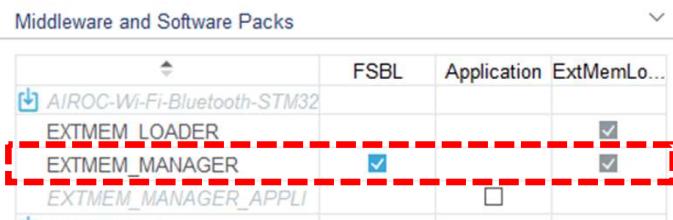
You can fuse OTP in **STM32CubeProgrammer**.

However, since the **OTP** area can no longer be written once fused, **it must be used with extreme caution**.

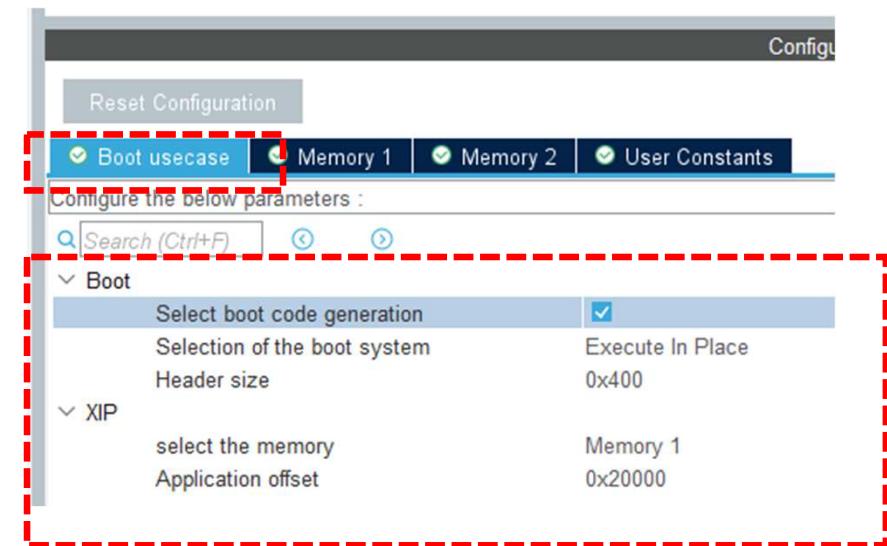
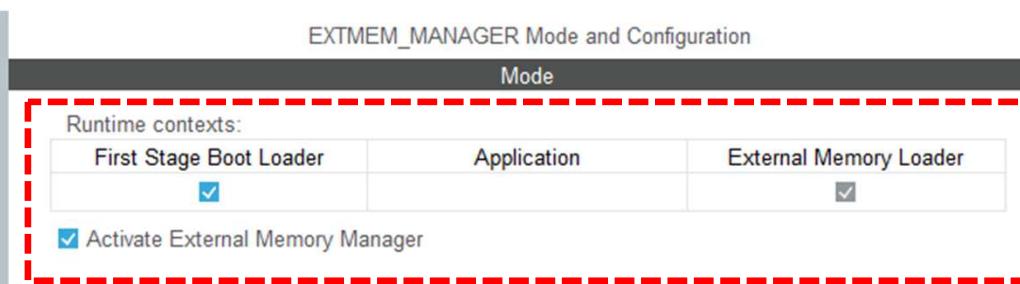
In this hands-on, **OTP settings will not be configured**.



# EXTMEM\_MANAGER Configuration



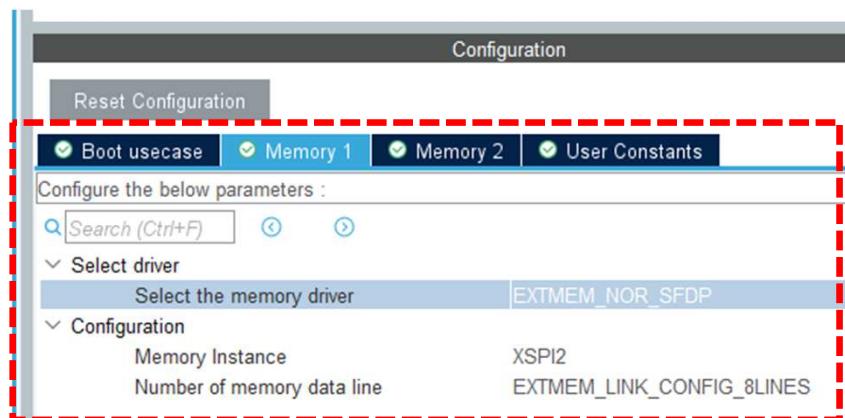
Select **FSBL** as the **Context** for **EXTMEM\_MANAGER**.



Set the **Boot System** to **Excute in Place**.



# EXTMEM\_MANAGER Configuration



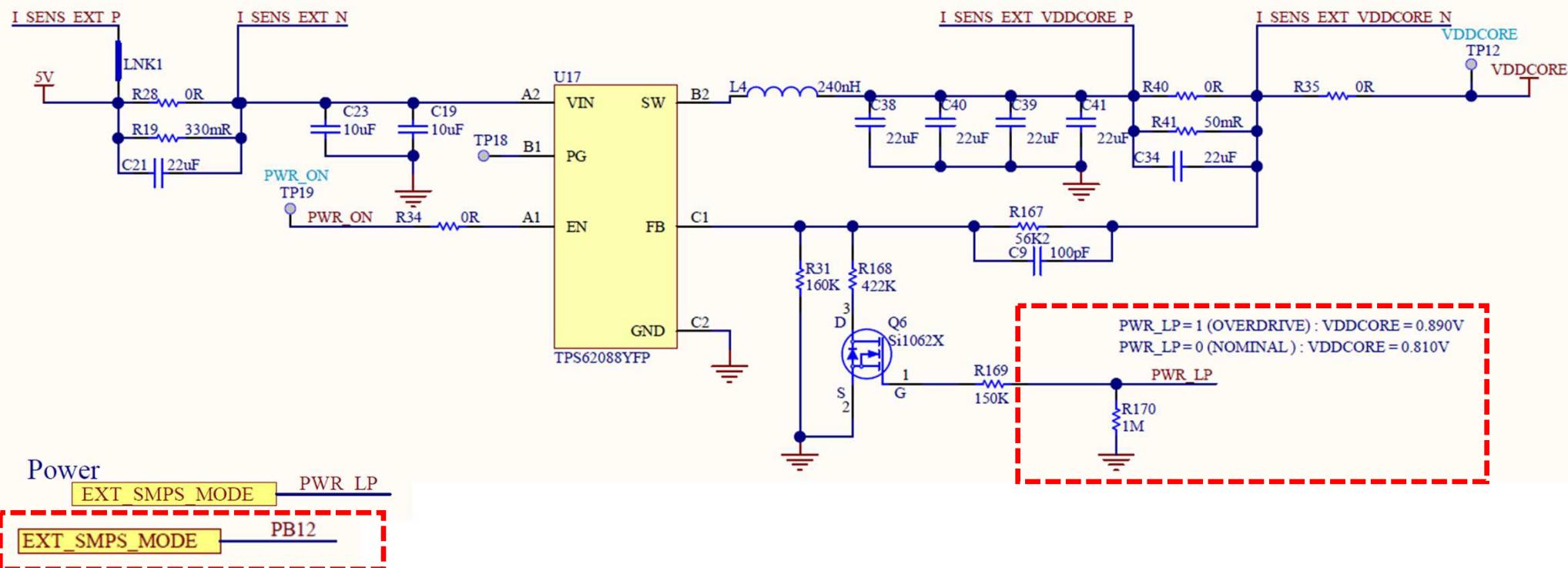
In this hands-on, FSBL loads the **Secure Application** into **AXISRAM1 (0x34000000)**.

Table 1. Memory map based on IDAU mapping (continued)

IDAU security type	Mapping	Description
Reserved SRAM/AXI bank secure	0x34270000	AXISRAM4
	0x34200000	AXISRAM3
	0x34100000	AXISRAM2
	0x34000000	AXISRAM1 (FLEXMEM extension bites on the lower end)



# PWR\_LP\_PIN Configuration



To use **Overdrive Mode**, set the **PWR\_LP(PB12)** pin to **High** to configure the **External DCDC** output voltage to **0.89V**.

Home > STM32N657X0HxQ > stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_test.ioc - Pinout & Configuration >

### Pinout & Configuration

Categories A-Z

	FSBL	Application	ExtMemLoader
CORTEX_M55_FSBL	<input checked="" type="checkbox"/>		
CORTEX_M55_S		<input checked="" type="checkbox"/>	
CDPDMA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
GPIO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
HDPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICACHE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NVIC1_S_Application		<input checked="" type="checkbox"/>	
NVIC_FSBL	<input checked="" type="checkbox"/>		
RAMCFG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RCC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SYS_S	<input type="checkbox"/>	<input type="checkbox"/>	
WWDG	<input type="checkbox"/>	<input type="checkbox"/>	

Analog >

Timers >

Multimedia >

Security >

Computing >

Middleware and Software Packs >

Trace and Debug >

	FSBL	Application	ExtMemLoader
DEBUG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HDP	<input checked="" type="checkbox"/>		

### Clock Configuration

Software Packs

GPIO Mode and Configuration

Configuration

Group By Peripherals

GPIO XSPI Features

Search Signals Search (Ctrl+F)

Show only Modified Pins

Pin N...	Signal on...	Pin Conte...	Pin Privil...	GPIO out...	GPIO mode	GPIO Pul...	Maximu...	User Label	Modified
PB12	n/a	First Sta...	n/a	High	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG0	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG8	n/a	Free	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG10	n/a	Free	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>

Set the **Context of PB12 to First Stage Boot Loader** and configure the **GPIO output level to High**.

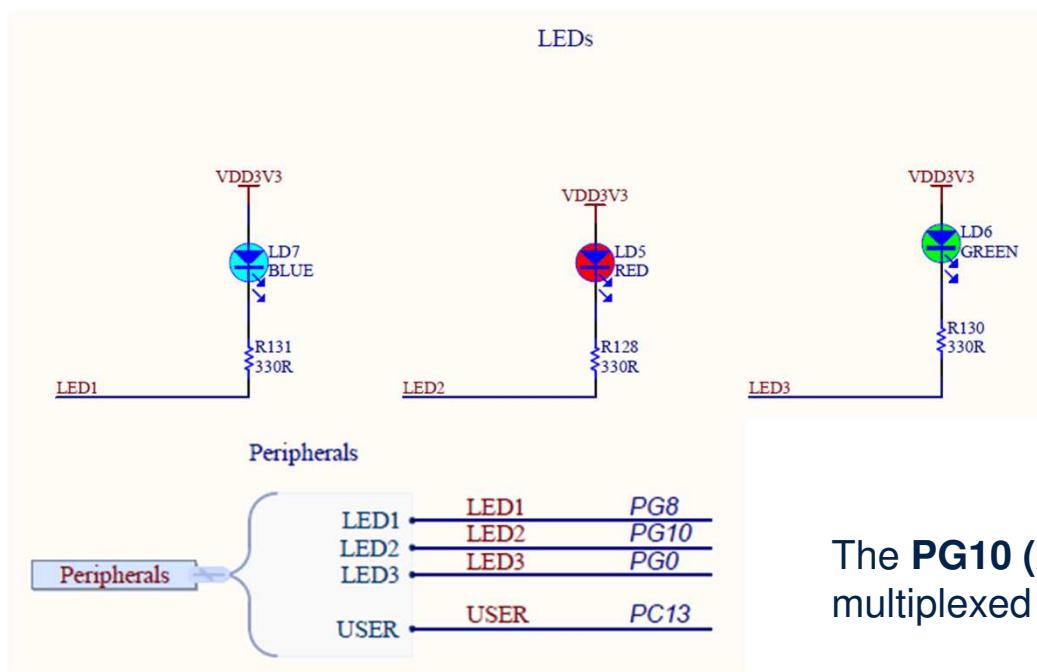
PB12 Configuration :

Pin Context Assignment	First Stage Boot Loader
GPIO output level	High
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	Low
User Label	



Set PB12 PIN as **GPIO\_Output**.

# LED\_PIN Configuration



The **PG10 (AF11)** is used as the **BootFailed** pin. This pin is multiplexed with **UART5\_TX** to send **UART status traces**.

Table 29. Pin configuration for UART5

UART5	
UART5_TX	PG10 (AF11)

It is a specific UART instance to retrieve data in case of blocking failure.



# LED\_PIN Configuration

Categories A-Z

	FSBL	Application	ExtMemLoader
CORTEX_M55_FSBL	<input checked="" type="checkbox"/>		
CORTEX_M55_S		<input checked="" type="checkbox"/>	
GPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>GPIO</b>	<b><input checked="" type="checkbox"/></b>	<b><input checked="" type="checkbox"/></b>	<b><input type="checkbox"/></b>
HPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICACHE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NVIC1_S_Application		<input checked="" type="checkbox"/>	
NVIC_FSB	<input checked="" type="checkbox"/>		
RAMCFG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RCC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SYS_S	<input type="checkbox"/>	<input type="checkbox"/>	
UART0	<input type="checkbox"/>	<input type="checkbox"/>	

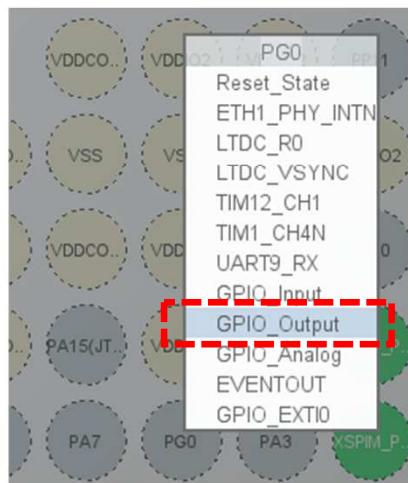
Configuration

Group By Peripherals

GPIO  XSPI  Features

Search Signals   Show only Modified Pins

Pin N...	Signal on...	Pin Conte...	Pin Privil...	GPIO out...	GPIO mode	GPIO Pul...	Maximu...	User Label	Modified
PB12	n/a	First Sta...	n/a	High	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG0	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG8	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG10	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>



PG0 Configuration :-

Pin Context Assignment  Application

GPIO output level

GPIO mode

GPIO Pull-up/Pull-down

Maximum output speed

User Label

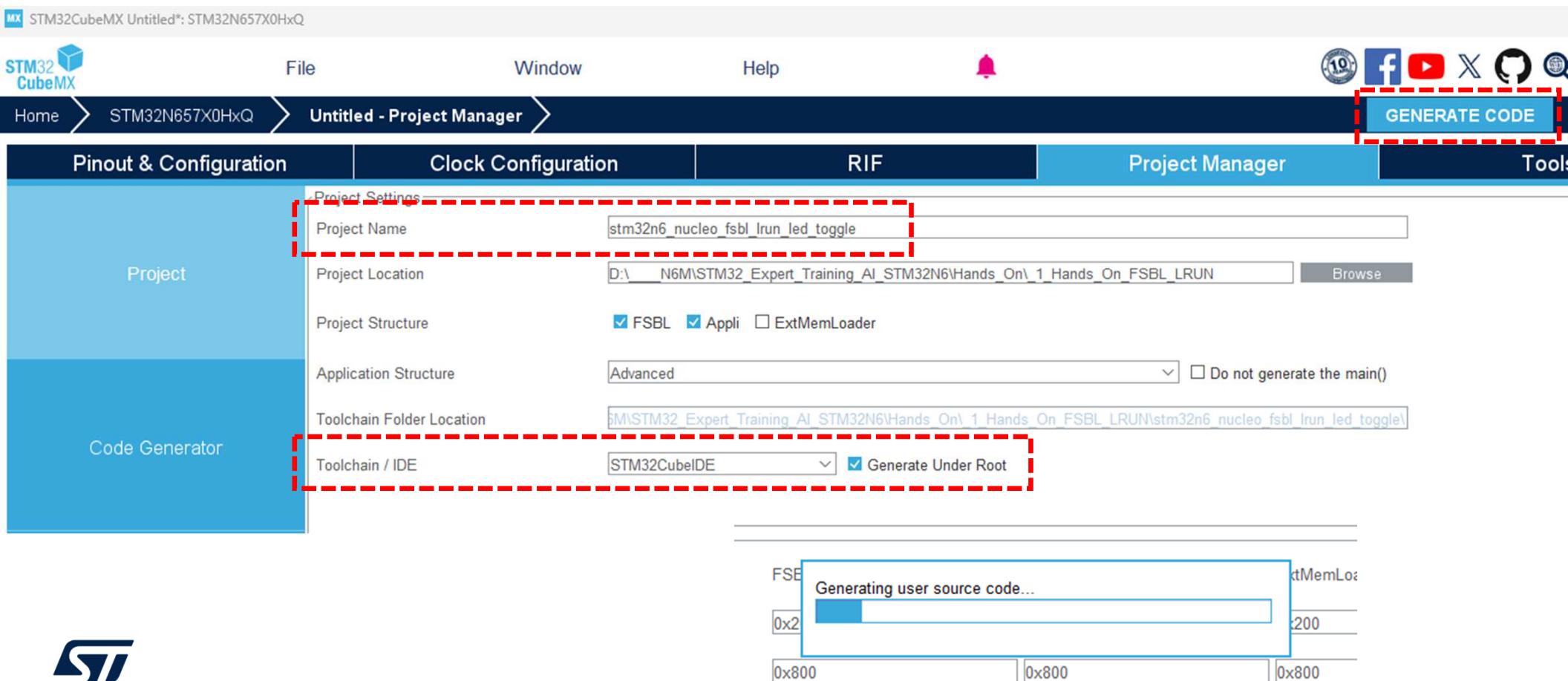


# BSEC Configuration

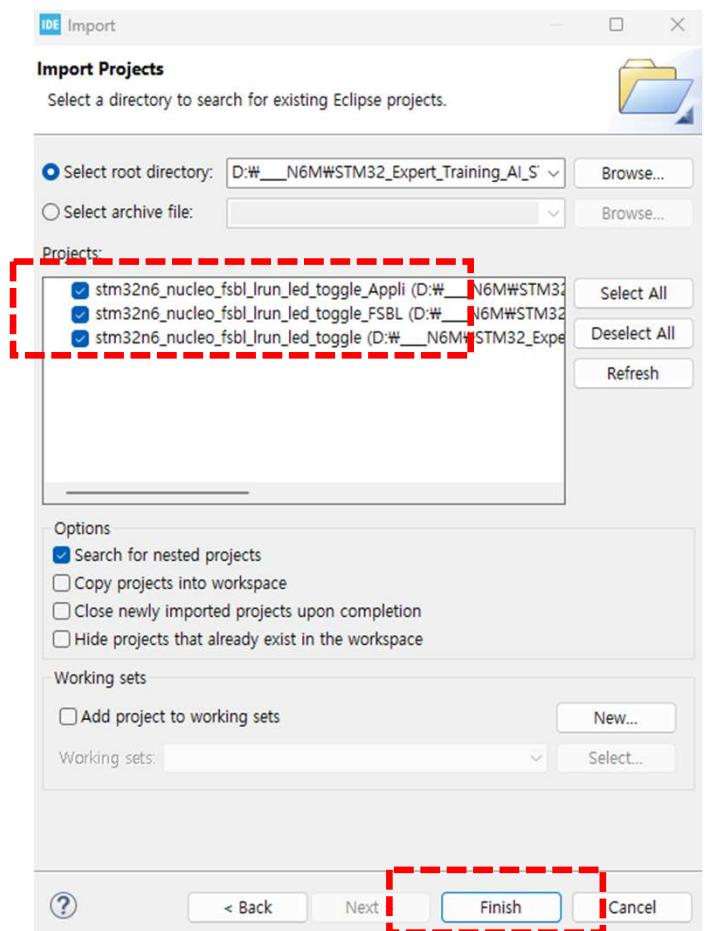
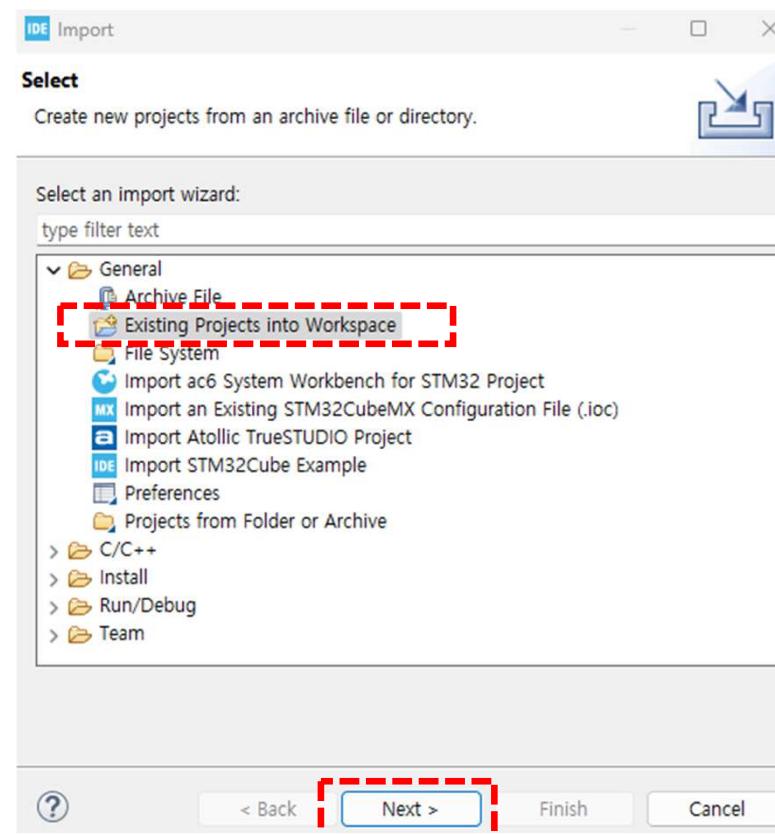
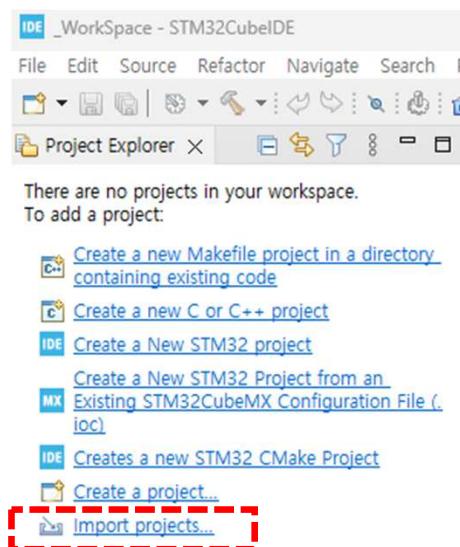
The screenshot shows the Pinout & Configuration interface. On the left, there's a sidebar with categories like System Core, Analog, Timers, Connectivity, Multimedia, and Security. Under Security, there's a table for peripherals. The first row, BSEC, has a checked checkbox in the FSBI column and a checked checkbox in the RIF column. A red dashed box highlights this row. On the right, the Clock Configuration tab is active, showing the BSEC Mode and Configuration section. It includes a table for runtime contexts: First Stage Boot Loader (checked), Application (unchecked), and External Memory Loader (unchecked). A checked checkbox labeled "Activated" is also present. A warning message at the bottom states: "⚠ Warning: This peripheral has no parameters to be configured."

	FSBI	Application	ExtMemLo
BSEC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CRYP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HASH	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MCE1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MCE2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MCE3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MCE4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PKA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RIF	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
RNG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SAES	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

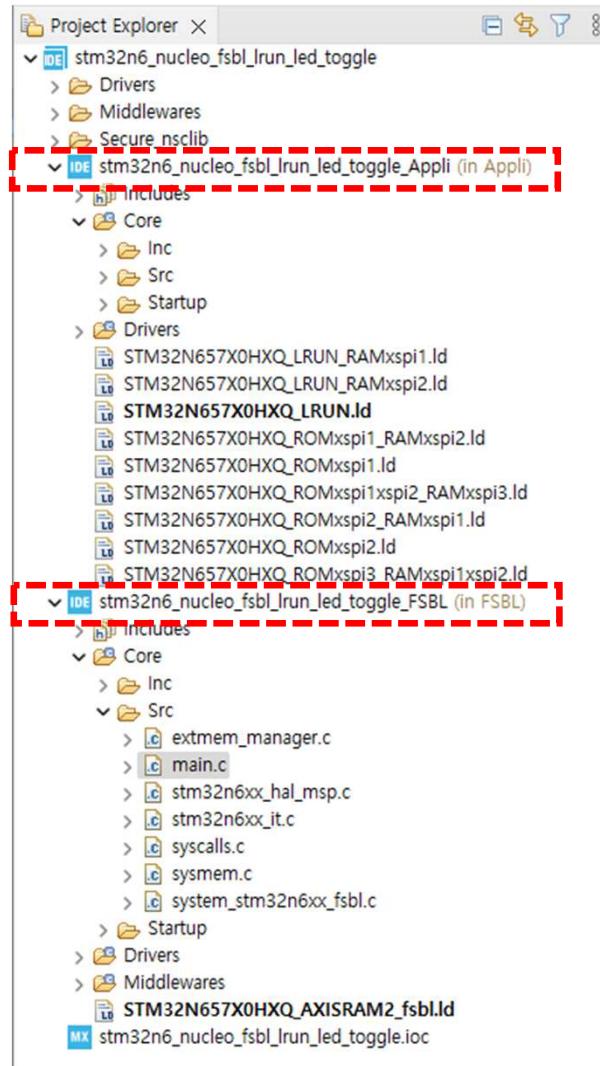
# GENERATE CODE



# STM32CubeIDE Import Project



# STM32CubeIDE Import Project



Two projects are created: **FSBL** and **Secure Application**.

# Add Code to FSBL

IDE \_WorkSpace - stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer X

stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle

- Drivers
- Middlewares
- Secure\_nsclib
- stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_Appli (in Appli)
- stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL (in FSBL)
- Includes
- Core
  - Inc
  - Src
    - extmem\_manager.c
    - main.c
    - stm32n6xx\_hal\_msp.c
    - stm32n6xx\_it.c
    - syscalls.c
    - sysmem.c
    - system\_stm32n6xx\_fsbl.c
  - Startup
- Drivers
- Middlewares

STM32N657X0HXQ\_AXIRAM2\_fsbl.ld

stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle.ioc

main.c X

```
76  /* Enable the CPU Cache */
77
78  /* Enable I-Cache-----*/
79  SCB_EnableICache();
80
81  /* Enable D-Cache-----*/
82  SCB_EnableDCache();
83
84  /* MCU Configuration-----*/
85  HAL_Init();
86
87  /* USER CODE BEGIN Init */
88
89  /* USER CODE END Init */
90
91  /* Configure the system clock */
92  SystemClock_Config();
93
94  /* USER CODE BEGIN SysInit */
95
96  /* USER CODE END SysInit */
97
98  /* Initialize all configured peripherals */
99  MX_GPIO_Init();
100  MX_BSEC_Init();
101  MX_XSPI2_Init();
102  MX_EXTMEM_MANAGER_Init();
103  /* USER CODE BEGIN 2 */
104
105  /* USER CODE END 2 */
```

main.c X

```
79  SCB_EnableICache();
80
81  /* Enable D-Cache-----*/
82  SCB_EnableDCache();
83
84  /* MCU Configuration-----*/
85  HAL_Init();
86
87  /* USER CODE BEGIN Init */
88
89  MX_GPIO_Init();
90  HAL_Delay(1);
91
92  /* USER CODE END Init */
93
94  /* Configure the system clock */
95  SystemClock_Config();
96
97  /* USER CODE BEGIN SysInit */
98
99  /* USER CODE END SysInit */
100
101 /* Initialize all configured peripherals */
102 MX_GPIO_Init();
103 MX_BSEC_Init();
104 MX_XSPI2_Init();
105 MX_EXTMEM_MANAGER_Init();
106 /* USER CODE BEGIN 2 */
107
108 /* USER CODE END 2 */
```

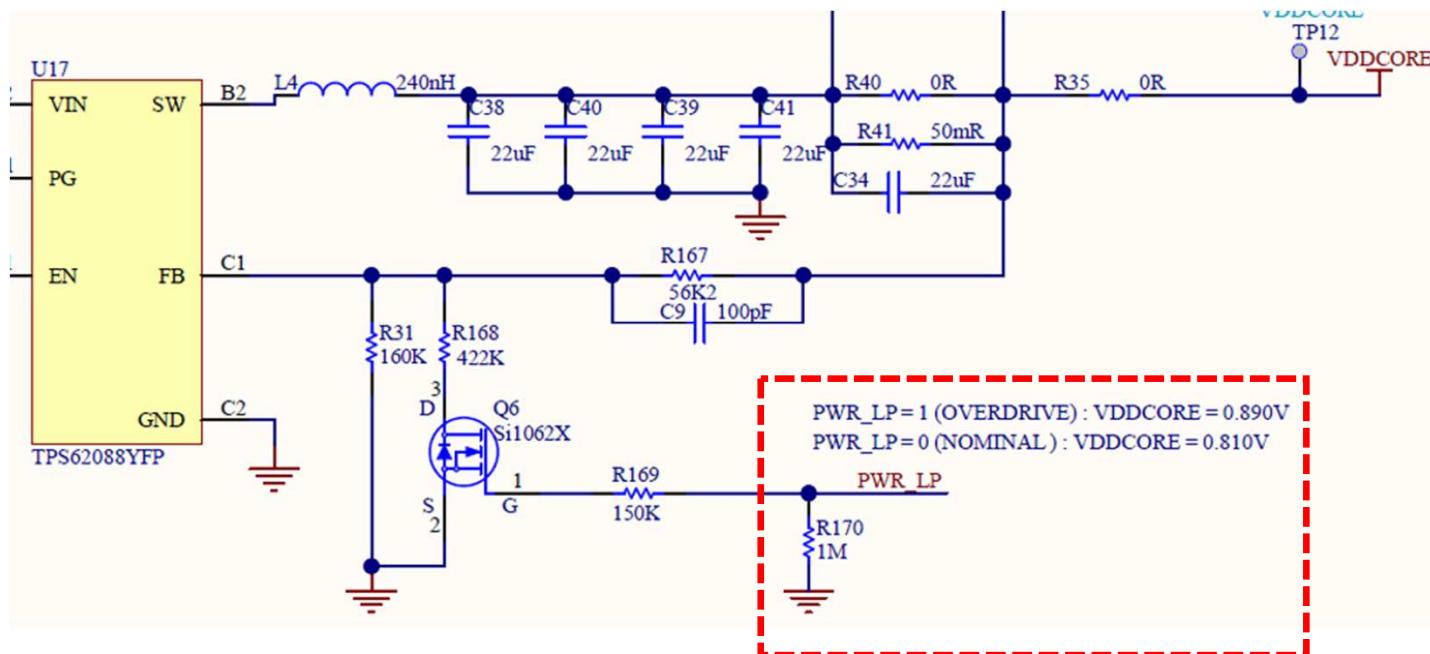
MX\_GPIO\_Init();  
HAL\_Delay(1);

Add code to set **PB12 High** before configuring the clock to enable **Overdrive Mode (Core Clock: 800MHz)** in the “main.c”



# Add Code to FSBL

```
306 */  
307 * @brief GPIO Initialization Function  
308 * @param None  
309 * @retval None  
310 */  
311 static void MX_GPIO_Init(void)  
312 {  
313     GPIO_InitTypeDef GPIO_InitStruct = {0};  
314     /* USER CODE BEGIN MX_GPIO_Init_1 */  
315  
316     /* USER CODE END MX_GPIO_Init_1 */  
317  
318     /* GPIO Ports Clock Enable */  
319     __HAL_RCC_GPIOA_CLK_ENABLE();  
320     __HAL_RCC_GPIOB_CLK_ENABLE();  
321  
322     /*Configure GPIO pin Output Level */  
323     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, GPIO_PIN_SET);  
324  
325     /*Configure GPIO pin : PB12 */  
326     GPIO_InitStruct.Pin = GPIO_PIN_12;  
327     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;  
328     GPIO_InitStruct.Pull = GPIO_NOPULL;  
329     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;  
330     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);  
331  
332     /* USER CODE BEGIN MX_GPIO_Init_2 */  
333  
334     /* USER CODE END MX_GPIO_Init_2 */  
335 }
```



# Add Code to FSBL

The screenshot shows the STM32CubeIDE interface. On the left, the Project Explorer displays a project structure for a Nucleo board. A red dashed box highlights the file `stm32n6xx_hal_msp.c` in the `Src` folder under the `Core` section. The right pane shows the code editor for this file. A red dashed box highlights the code block between `/* USER CODE BEGIN Define */` and `/* USER CODE END Define */`. The code consists of two define statements:

```
#define HSLV OTP 124
#define VDDIO3_HSLV_MASK (1<<15)
```

Add code between `/* USER CODE BEGIN Define */` and `/* USER CODE END Define */` in `stm32n6xx_hal_msp.c`.

`#define HSLV OTP 124`  
`#define VDDIO3_HSLV_MASK (1<<15)`



# Add Code to FSBL

The screenshot shows the STM32CubeIDE interface. On the left, the Project Explorer displays the project structure for "stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle". The file "stm32n6xx\_hal\_msp.c" is open in the code editor. A red dashed box highlights the code being added to the XSPI MspInit function. The code adds BSEC and SYSCFG clock enablements, reads fuse data, sets PWR and SYSCFG configurations, and enables the XSPI memory interface clock.

```
88 void HAL_XSPI_MspInit(XSPI_HandleTypeDef* hxsip)
89 {
90     GPIO_InitTypeDef GPIO_InitStruct = {0};
91     RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};
92     if(hxsip->Instance==XSPI2)
93     {
94         /* USER CODE BEGIN XSPI2_MspInit 0 */
95
96         BSEC_HandleTypeDef hbsec;
97         uint32_t fuse_data = 0;
98         /* Enable BSEC & SYSCFG clocks to ensure BSEC data accesses */
99         __HAL_RCC_BSEC_CLK_ENABLE();
100        __HAL_RCC_SYSCFG_CLK_ENABLE();
101
102        hbsec.Instance = BSEC;
103        if (HAL_BSEC OTP_Read(&hbsec, HSLV OTP, &fuse_data) != HAL_OK)
104        {
105            Error_Handler();
106        }
107
108        /* Set PWR configuration for IO speed optimization */
109        __HAL_RCC_PWR_CLK_ENABLE();
110        HAL_PWREx_EnableVddIO3();
111        HAL_PWREx_ConfigVddIORange(PWR_VDDIO3, PWR_VDDIO_RANGE_1V8);
112        /* Set SYSCFG configuration for IO speed optimization (clock already enabled) */
113        HAL_SYSCFG_EnableVDDIO3CompensationCell();
114
115        /* Enable the XSPI memory interface clock */
116        HAL_RCC_XSPI2_CLK_ENABLE();
117
118     /* USER CODE END XSPI2_MspInit 0 */
119 }
```



# Add Code to FSBL

```
BSEC_HandleTypeDef hbsec;
uint32_t fuse_data = 0;
/* Enable BSEC & SYSCFG clocks to ensure BSEC data accesses */
__HAL_RCC_BSEC_CLK_ENABLE();
__HAL_RCC_SYSCFG_CLK_ENABLE();

hbsec.Instance = BSEC;
if (HAL_BSEC OTP_Read(&hbsec, HSLV OTP, &fuse_data) != HAL_OK)
{
Error_Handler();
}

/* Set PWR configuration for IO speed optimization */
__HAL_RCC_PWR_CLK_ENABLE();
HAL_PWREx_EnableVddIO3();
HAL_PWREx_ConfigVddIORange(PWR_VDDIO3, PWR_VDDIO_RANGE_1V8);
/* Set SYSCFG configuration for IO speed optimization (clock already enabled)
*/
HAL_SYSCFG_EnableVDDIO3CompensationCell();

/* Enable the XSPI memory interface clock */
__HAL_RCC_XSPI2_CLK_ENABLE();
```



# Build FSBL

The screenshot shows the STM32CubeIDE interface with the following details:

- Project Explorer:** Displays the project structure:
  - stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle
  - ↳ Drivers
  - ↳ Middlewares
  - ↳ Secure\_nsclib
  - ↳ stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_Appli (in Appli)
  - ↳ stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL (in FSBL)
    - ↳ Binaries
    - ↳ Includes
    - ↳ Core
      - ↳ Inc
      - ↳ Src
        - extmem\_manager.c
        - main.c
        - stm32n6xx\_hal\_msp.c
        - stm32n6xx\_it.c
        - syscalls.c
        - sysmem.c
        - system\_stm32n6xx\_fsbl.c
    - Startup
  - STM32N657X0HXQ\_AXISRAM2\_fsbl.ld
  - stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle.ioc
- STM32N657X0HXQ\_AXISRAM2\_fsbl.ld:** Content of the Linker Script file (ld script). The script defines memory regions, sections, and startup code.
- CDT Build Console:** Shows the build log output.

```
18:17:17 **** Incremental Build of configuration Debug for project stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL
make -j16 all
arm-none-eabi-gcc ".../Core/Src/stm32n6xx_hal_msp.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDEBUG -DUSE_H
arm-none-eabi-gcc -o "stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf" "@objects.list" -mcpu=cortex-m55
Finished building target: stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf

arm-none-eabi-size stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf
arm-none-eabi-objdump -h -S stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf > "stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.h
text      data      bss      dec      hex filename
61020       12     3268    64300    fb2c stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf
arm-none-eabi-objcopy -O binary stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf "stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin
Finished building: default.size.stdout

Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin

Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.list

18:17:18 Build Finished. 0 errors, 0 warnings. (took 1s.912ms)
```

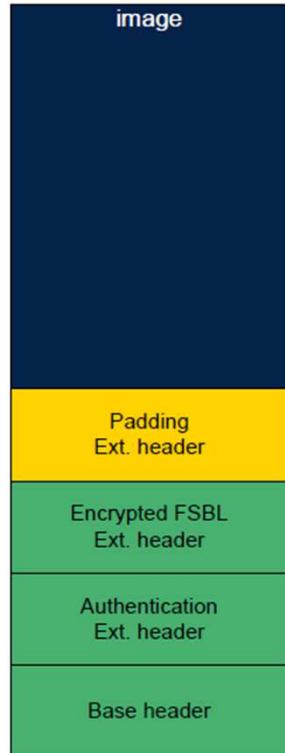
# Debug

The screenshot shows the STM32CubeIDE interface with the following details:

- Project Explorer:** Displays the project structure under "stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle".
- Code Editor:** Shows the main.c file with C code.
- Context Menu:** Opened over the main.c code editor, showing options like "Run As", "Debug As", and "Properties".
- Edit launch configuration properties Dialog:** Opened, showing the "Edit launch configuration properties" dialog with the following settings:
  - Name: stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL
  - Project: stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL
  - C/C++ Application: Debug/stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSBL.elf
  - Build Configuration: Select Automatically
  - Build (if required) before launching
  - Build Configuration dropdown: Select Automatically
  - Build Configuration buttons: Enable auto build (unchecked), Use workspace settings (checked), Disable auto build (unchecked)
  - Buttons: Revert, Apply, OK, Cancel



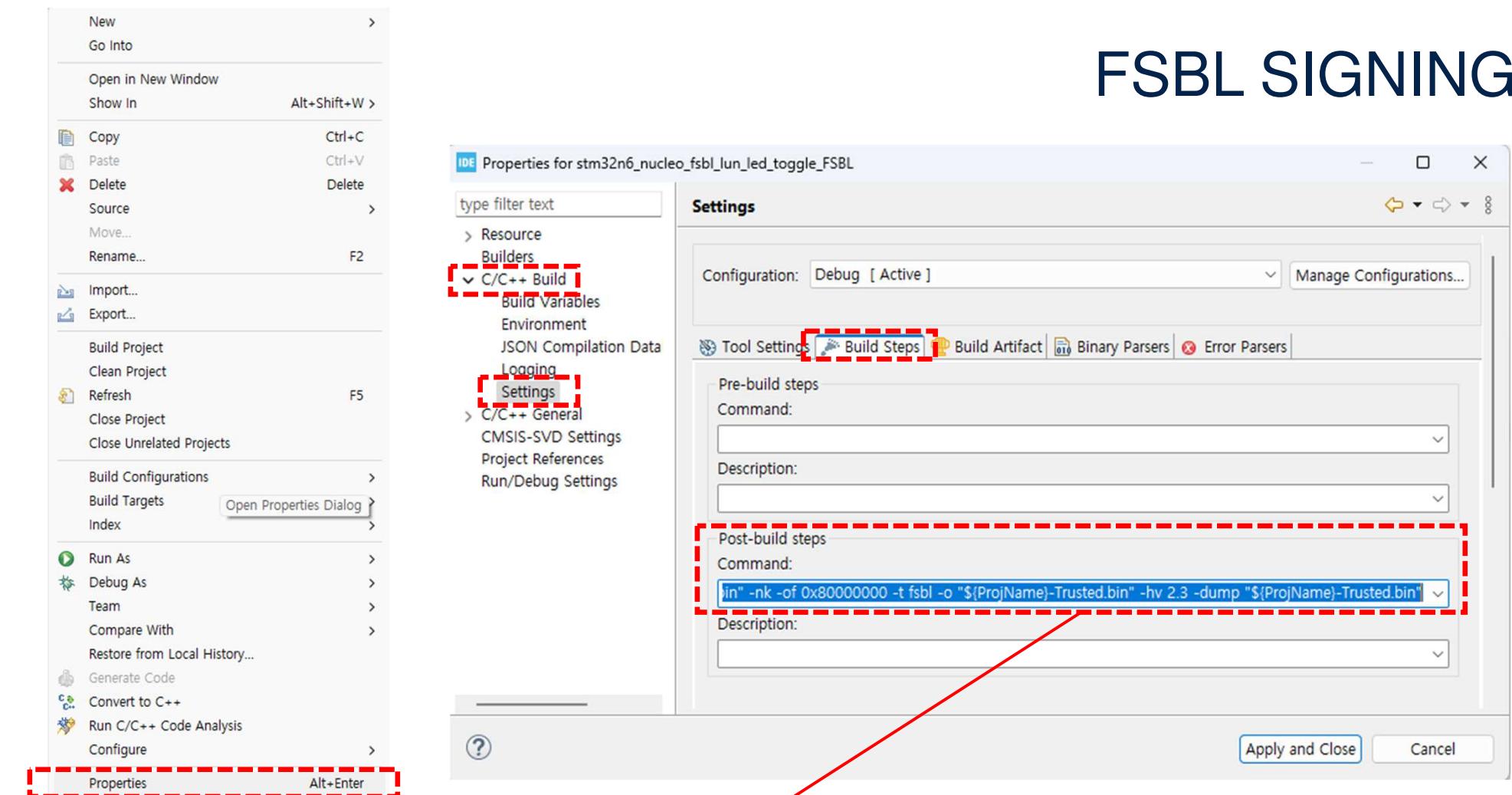
# FSBL Header



**Table 32. Base header**

Name	Length	Byte offset (Dec, hex)	Description	Part of signature
Magic number	32 bits	0, 0x0	'S'; 'T'; 'M'; 0x32	n
Image signature	768 bits	4, 0x4	ECDSA signature: calculated on header + image	n
Image checksum	32 bits	100, 0x64	Checksum of the secured payload	n
Header version	32 bits	104, 0x68	Header version v2.3 = 0x00MMmm00MM: major version = 0x02mm: minor version = 0x03	y
Image length	32 bits	108, 0x6C	Length of FSBL image in bytes	y
Image entry point	32 bits	112, 0x70	Entry point of image	y
Reserved1	32 bits	116, 0x74	Reserved (64-bits entry point)	y
Load address	32 bits	120, 0x78	Load address of image	y
Reserved2	32 bits	124, 0x7C	Reserved (64-bits load address)	y
Version number	32 bits	128, 0x80	Image version (monotonic number)	y
Extension flags	32 bits	132, 0x84	b0=1: Authentication extension header b1=1: FSBL encryption extension header b31=1: Padding extension header	y
Post header length	32 bits	136, 0x88	Length in bytes of all extension headers	y
Binary type	32 bits	140, 0x90	Used to check the binary type	y
PAD	64 bits	144, 0x94	Reserved padding bytes. Must all be set to 0	y
Nonsecure payload length	32 bits	152, 0x98	Length in bytes of optional nonsecured payload	n
Nonsecure payload hash	32 msb bits	156, 0x9C	32 msb bits of SHA256 of nonauthenticated payload	n

# FSBL SIGNING



```
cd "${ProjDirPath}/Debug" && echo y | "C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe" -bin "${ProjName}.bin" -nk -of 0x80000000 -t fsbl -o "${ProjName}-Trusted.bin" -hv 2.3 -dump "${ProjName}-Trusted.bin"
```

# FSBL SIGNING

```
STM32 Signing Tool v2.19.0

Adding head padding bytes for payload with header v2.3
Header version 2.3 preparation...
Extracting Entry point value from the input file...
Entry point value : 0x341813ad
The headed image file generated successfully: stm32n6_nucleo_fsbl_lun_led_toggle_FSBL-Trusted.bin

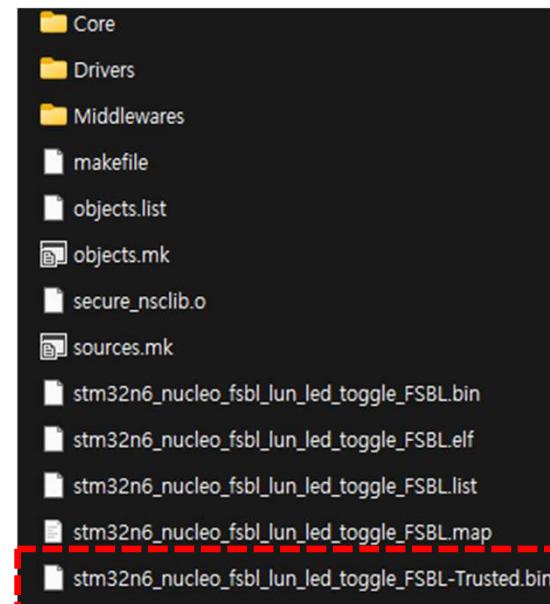
Header description:

Magic: 0x53544d32
Signature: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Checksum: 0x69e803
Header version: 0x20300
Size: 0xf1e0
Load address: 0xffffffff
Entry point: 0x341813ad
Image version: 0x0
Extension: 0x80000000

Pad header detected:
Type: 0x5354ffff
Size: 0x1a0
Padding values: 92 05 ca bb 60 ea 25 df e8 39 35 3b 5a c6 34 b4 9d ae c3 59 9a 26 eb ec d7 f0 7b 9b f0 4b 6b
28 4a dd fd 2f e4 58 80 ab 17 0b af 32 5d a6 17 82 7a 9a c7 8d 50 ef e0 85 3f ac 11 9a dd 72
a0 4d 8d 93 00 5d d1 cc 7e fa ce a0 cb 58 1e ba cc 59 f6 44 b0 d2 f6 01 32 36 21 38 12 de 9b
c3 75 31 44 4e 3b e7 fb 2e 19 4d d9 56 08 5f 89 00 5b d8 b9 32 1c dc 66 f5 ed 28 bb 5c 83 75
39 81 ac 91 38 e6 1c ce bf a9 32 57 c0 b8 87 33 c3 d6 08 53 35 95 1a 79 c6 ae d7 eb 4d e7 8c
0a c7 4a bf e9 a7 31 7a 8f f1 ea 93 26 ca a9 46 91 7f 07 22 46 6b 61 09 08 b3 91 e4 0e 5c 74
9f e5 4e 7a 62 0a c1 f6 9a 51 de e3 cc 61 9c 93 f1 08 fb 74 ae 25 50 b6 16 b6 18 6c 11 7a 7a
06 f0 b7 82 47 cb f5 67 b4 76 da 67 c1 4d de f5 57 e0 10 37 1e 2d 0e 60 74 b0 10 22 24 7f af
b0 6f ae a0 cd 2f bf 7a cd 4c ee 4c 8f ae e6 c9 91 b5 1b bb ed 57 d0 a2 62 57 80 4d 39 c0 5b
d9 aa 49 70 e5 07 cc c3 fa 5b 8a 5f 70 75 05 57 42 a7 e3 55 d1 d3 45 b3 01 c2 69 13 73 e7 ba
87 c9 ac 86 05 d7 4c c1 3f c9 59 83 79 2e 0e 26 1a ce f4 9b 73 a3 ca 6a 40 0e 23 81 85 0e 8a
97 19 e0 ac 25 e4 5e a9 eb 78 72 23 a2 2b b7 da b2 31 c9 e3 16 5a 58 af 6d f6 08 91 c5 96 f1
ad 22 9f 54 f4 2f 10 e5 00 00 02 00 51 38 78 dd a8 2a 00 00 e0 45 3c 9c a1 02 00 00 30 b7 47
9c a1 02 00 00

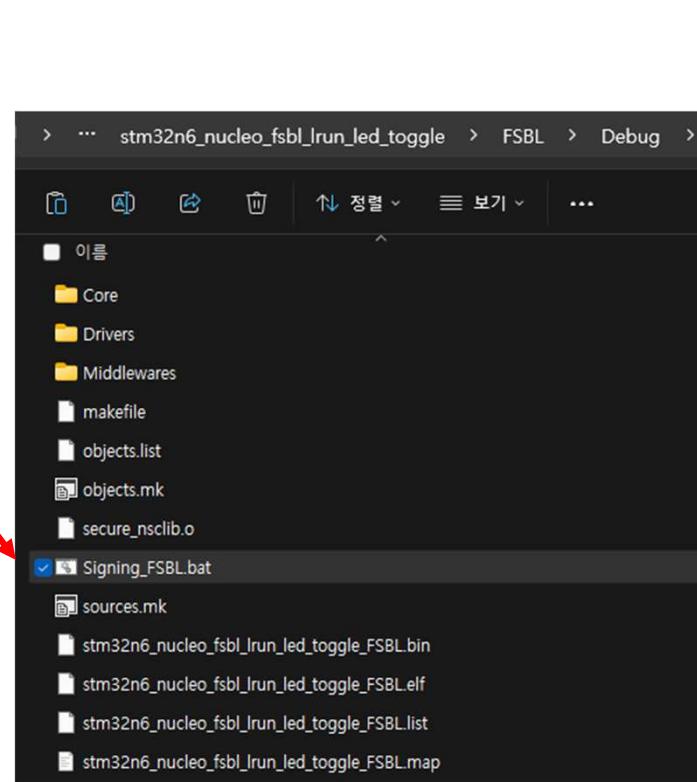
Post header 2.3 information :
Binary type: 0x10
Non authenticated payload length: 0x0
Non authenticated payload hash: 0x0
```

16:27:56 Build Finished. 0 errors, 0 warnings. (took 2s.383ms)



# FSBL SIGNING

It is also possible to use the Signing Tool externally.



Copy the **Signing\_FSBLS.bat** file from the **Siging\_Batch\_File** folder into the **FSBL\Debug** folder of the current project.



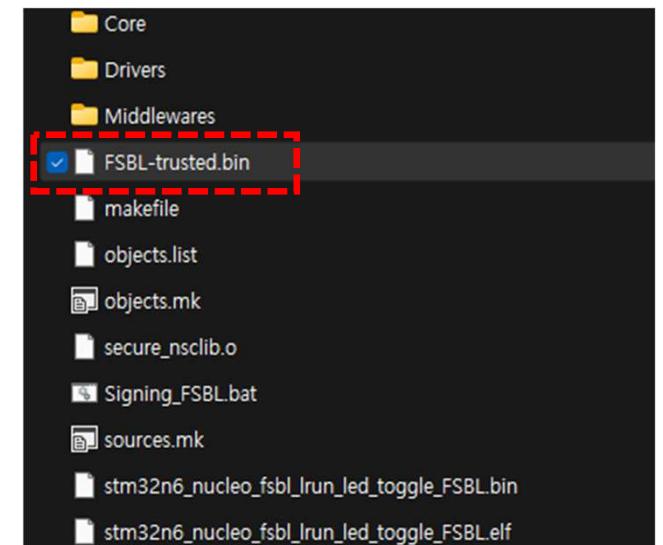
# FSBL SIGNING

```
 Signing_FSBL.bat
1 @echo off
2
3 REM STM32MP_SigningTool_CLI.exe -bin FSBL.bin -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
4
5 set SIGNING_TOOL_PATH="C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe"
6 set APP_NAME="stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin"
7
8 echo %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
9 %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
10
11 cmd /k
```

Paste the path of the **STM32\_SigningTool\_CLI.exe** file from **STM32CubeProgrammer**.

After building the **FSBL**, paste the name of the generated **binary file**.





# FSBL SIGNING

The signed file "**FSBL-trusted.bin**" is generated upon completing the signing process.

# Add Code to APP

The screenshot shows the STM32CubeIDE interface. On the left, the Project Explorer displays the project structure for 'stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle'. It includes subfolders for Drivers, Middlewares, Secure\_nsclib, and the main application folder 'stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_App'. Within the application folder, there are Core, Includes, and Src subfolders. The Src folder contains files like main.c, secure\_nsclib.c, and various header files. On the right, the main.c file is open in the code editor. The code is a template for a Secure Application, showing initialization of peripherals and an infinite loop for toggling LEDs. A blue selection bar highlights the section from line 88 to line 104, which corresponds to the code to be inserted.

```
72 HAL_Init();
73 /* USER CODE BEGIN Init */
74
75 /* USER CODE END Init */
76
77 /* USER CODE BEGIN SysInit */
78
79 /* USER CODE END SysInit */
80
81 /* Initialize all configured peripherals */
82 MX_GPIO_Init();
83 SystemIsolation_Config();
84 /* USER CODE BEGIN 2 */
85
86
87 /* USER CODE END 2 */
88
89 /* Infinite loop */
90 /* USER CODE BEGIN WHILE */
91 while (1)
92 {
93     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
94     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_8);
95     HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_10);
96     HAL_Delay(200);
97     /* USER CODE END WHILE */
98
99     /* USER CODE BEGIN 3 */
100 }
101 /* USER CODE END 3 */
102 }
103
104 */
105 * @brief RIF Initialization Function
106 * @param None
107 * @retval None
108 */
```

```
HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_8);
HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_10);
HAL_Delay(200);
```

Insert code in **main.c** of the **Secure Application** to toggle the LEDs.



# Add Code to APP

In this example, the Application offset is set to 0x70020000. If the **ROM section** of the STM32N657X0HXQ\_ROMxspi2.1d file has not been updated accordingly, you need to modify it manually.

The screenshot shows a software interface for memory configuration and assembly code generation. At the top, there are tabs for 'Boot usecase', 'Memory 1', 'Memory 2', and 'User Constants'. Below the tabs, a search bar and some buttons are visible. The main area is divided into sections for 'Boot' and 'XIP'.

**Boot Section:**

- Select boot code generation (checkbox checked)
- Selection of the boot system
- Header size: 0x400
- Execute In Place (checkbox checked)

**XIP Section:**

- select the memory: Memory 1 (0x20000)
- Application offset: 0x20000

**Assembly Code:**

```
34 /* Entry Point */
35 ENTRY(Reset_Handler)
36 _Min_Heap_Size = 0x200; /* required amount of heap */
37 _Min_Stack_Size = 0x800; /* required amount of stack */
38
39 /* Highest address of the user mode stack */
40 _estack = ORIGIN(RAM) + LENGTH(RAM); /* end of "RAM" Ram type memory */
41 _sstack = _estack - _Min_Stack_Size;
42
43 /* Memories definition */
44 MEMORY
45{
46 ROM (xrw) : ORIGIN = 0x70100400, LENGTH = 511K
47 RAM (xrw) : ORIGIN = 0x34000000, LENGTH = 2048K
48 }
```

**Memory Definition:**

```
43 /* Memories definition */
44 MEMORY
45{
46 ROM (xrw) : ORIGIN = 0x70020400, LENGTH = 511K
47 RAM (xrw) : ORIGIN = 0x34000000, LENGTH = 2048K
48 }
```

A red dashed box highlights the 'ROM' and 'RAM' definitions in both the assembly code and the memory definition section. A red arrow points from the highlighted code in the assembly section to the highlighted definitions in the memory section.



# Build APP

The screenshot shows the STM32CubeIDE interface with the following components:

- Project Explorer:** Shows the project structure under "stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle".
- Code Editor:** Displays the `main.c` file content.
- Build Console:** Shows the build logs for the target `stm32n6_nucleo_fsbl_lrun_led_toggle_Appli`.

**Project Explorer Content:**

- stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle (selected)
- Drivers
- Middlewares
- Secure\_nsclib
- stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_Appli (in Appli)

  - Includes
  - Core
    - Inc
    - Src
      - main.c
      - secure\_nsclib.c
      - stm32n6xx\_hal\_msp.c
      - stm32n6xx\_it.c
      - syscalls.c
      - sysmem.c
      - system\_stm32n6xx\_s.c
    - Startup
  - Drivers
    - STM32N657X0HXQ\_LRUN\_RAMxspi1.ld
    - STM32N657X0HXQ\_LRUN\_RAMxspi2.ld
    - STM32N657X0HXQ\_LRUN.ld**
    - STM32N657X0HXQ\_ROMxspi1\_RAMxspi2.ld
    - STM32N657X0HXQ\_ROMxspi1.ld
    - STM32N657X0HXQ\_ROMxspi1xspi2\_RAMxspi3.ld
    - STM32N657X0HXQ\_ROMxspi2\_RAMxspi1.ld
    - STM32N657X0HXQ\_ROMxspi2.ld
    - STM32N657X0HXQ\_ROMxspi3\_RAMxspi1xspi2.ld
  - stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle\_FSB (in FSB)
  - stm32n6\_nucleo\_fsbl\_lrun\_led\_toggle.ioc

**Code Editor (main.c):**

```
28 ** in the root directory of this software component.
29 ** If no LICENSE file comes with this software, it is provided AS-IS.
30 **
31 ****
32 */
33
34 /* Entry Point */
35 ENTRY(Reset_Handler)
36
37 /* Highest address of the user mode stack */
38 _estack = ORIGIN(RAM) + LENGTH(RAM); /* end of "RAM" Ram type memory */
39 _sstack = _estack - _Min_Stack_Size;
40
41 _Min_Heap_Size = 0x200; /* required amount of heap */
42 _Min_Stack_Size = 0x800; /* required amount of stack */
43
44 /* Memories definition */
45 MEMORY
46 {
47     RAM      (xrw)      : ORIGIN = 0x34000400,    LENGTH = 2047K
48 }
49
50 /* Sections */
51 SECTIONS
52 {
53     /* The startup code into "RAM" Ram type memory */
54     .isr_vector :
55     {
56         . = ALIGN(4);
57         KEEP(*(.isr_vector)) /* Startup code */
58         . = ALIGN(4);
59     } >RAM
60
61     /* The program code and other data into "RAM" Ram type memory */
62     .text :
63     {
```

**Build Console Log:**

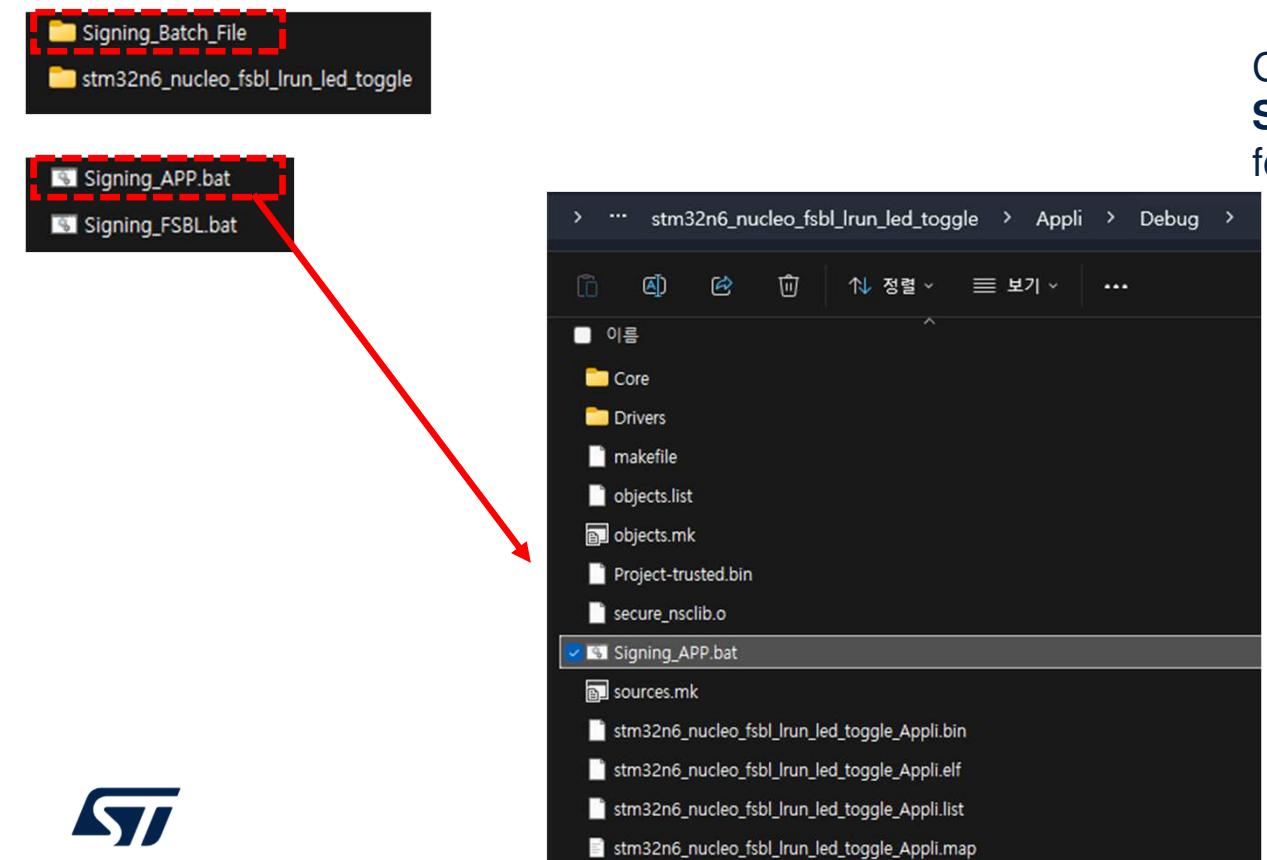
```
CDT Build Console [stm32n6_nucleo_fsbl_lrun_led_toggle_Appli]
arm-none-eabi-gcc ".../Core/Src/stm32n6xx_it.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDDEBUG
arm-none-eabi-gcc ".../Core/Src/syscalls.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDDEBUG
arm-none-eabi-gcc ".../Core/Src/sysmem.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDDEBUG
arm-none-eabi-gcc ".../Core/Src/system_stm32n6xx_s.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDDEBUG
arm-none-eabi-gcc -o "stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf" @"objects.list"
C:/ST/STM32CubeIDE_1.18.0/STM32CubeIDE/plugins/com.st.stm32cube.ide.mcu.externaltargets/obj
Finished building target: stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf

arm-none-eabi-size stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf
arm-none-eabi-objdump -h -S stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf > "stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.h"
        text      data      bss      dec      hex filename
        4656       12     2604     7272   1c68 stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf
arm-none-eabi-objcopy -O binary stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf "stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.bin"
Finished building: default.size.stdout

Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.list
Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.bin

18:33:42 Build Finished. 0 errors, 1 warnings. (took 3s.555ms)
```

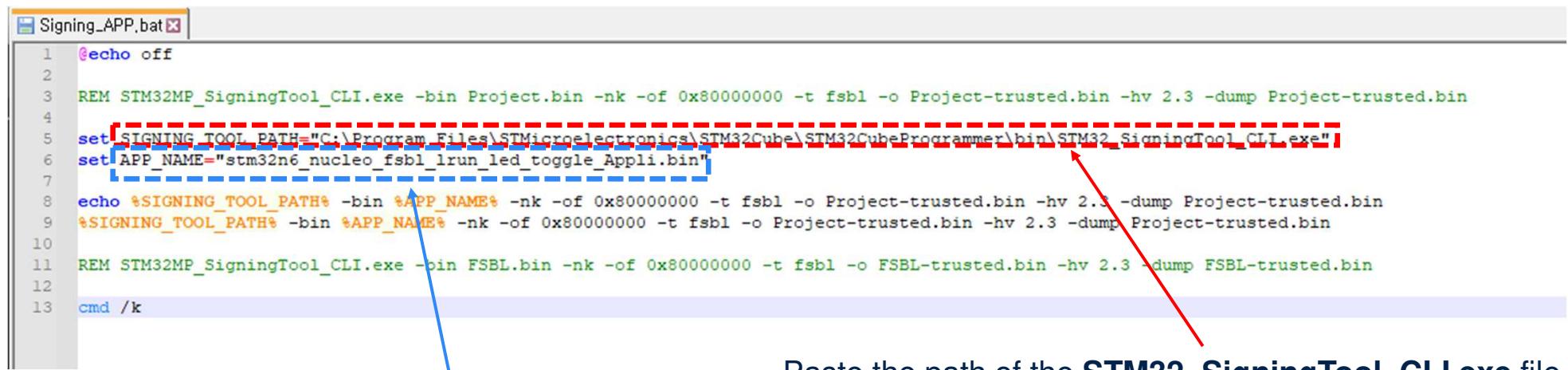
# APP SIGNING



Copy the **Signing\_FSB.bat** file from the **Siging\_Batch\_File** folder into the **FSBL\Debug** folder of the current project.



# APP SIGNING



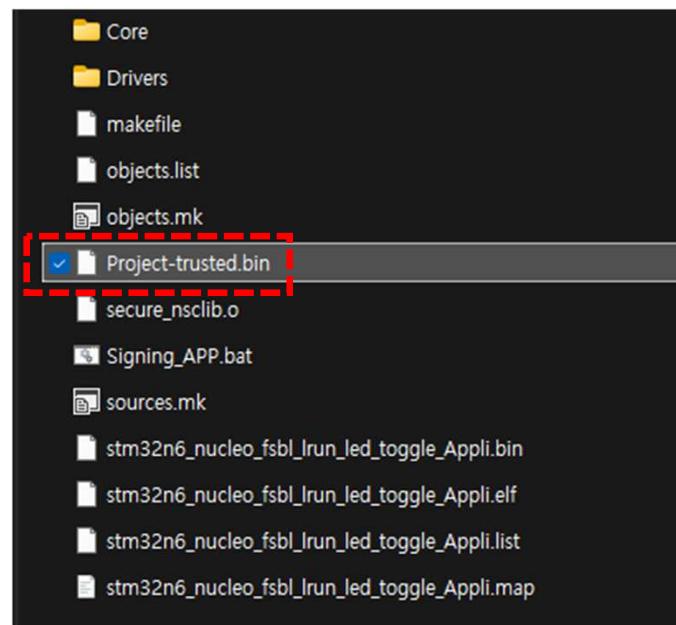
```
1 @echo off
2
3 REM STM32MP_SigningTool_CLI.exe -bin Project.bin -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -dump Project-trusted.bin
4
5 set SIGNING_TOOL_PATH="C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe"
6 set APP_NAME="stm32n6_nucleo_fsbl_lrun_led_toggle_Applic.bin"
7
8 echo %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.1 -dump Project-trusted.bin
9 %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -dump Project-trusted.bin
10
11 REM STM32MP_SigningTool_CLI.exe -bin FSBL.bin -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
12
13 cmd /k
```

Paste the path of the **STM32\_SigningTool\_CLI.exe** file from **STM32CubeProgrammer**.

After building the **Secure Application**, paste the name of the generated **binary file**.

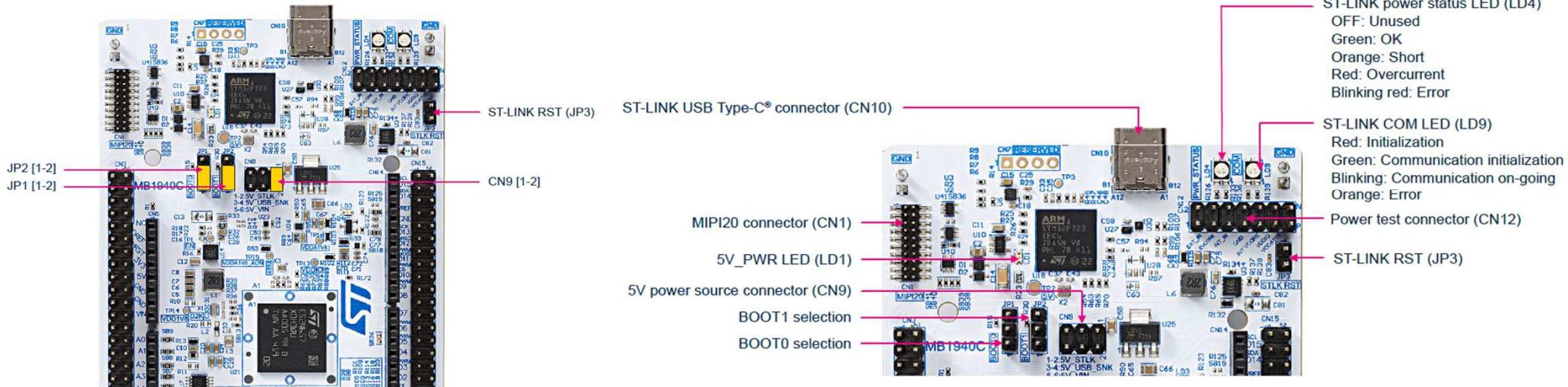


# APP SIGNING



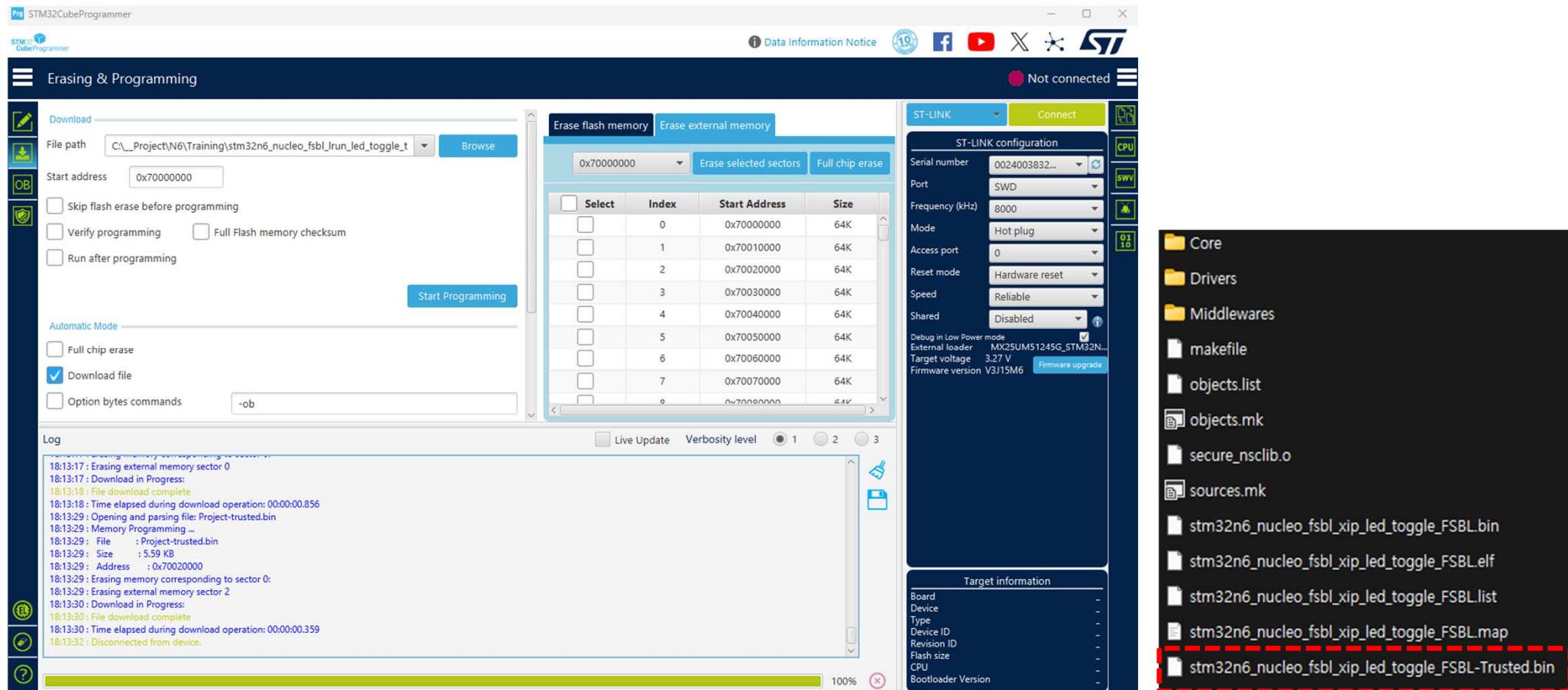
The signed file “**Project-trusted.bin**” is generated upon completing the signing process.

# FSBL and APP Download



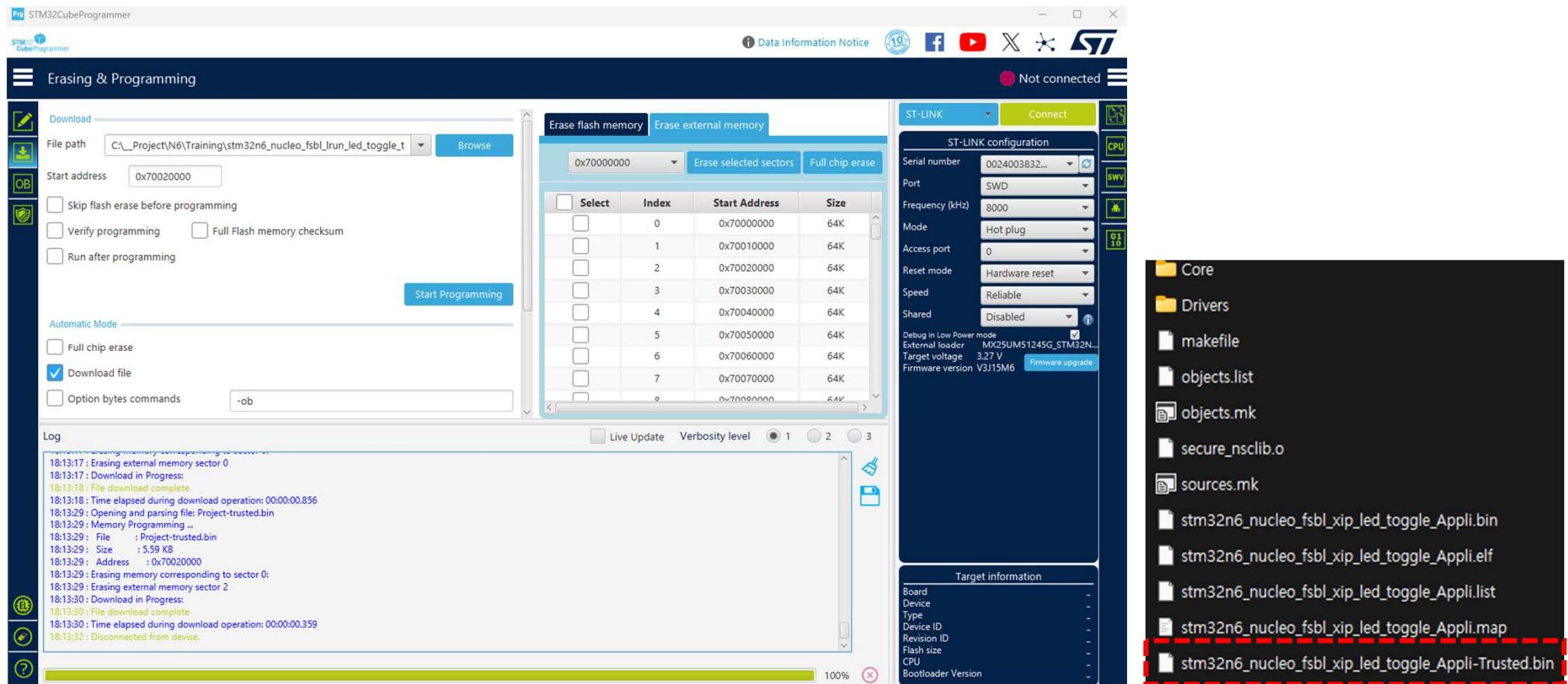
Jumper	Definition	Default position	Comment
CN9	5V power selection (user USB power source selection)	[1-2]	5V from STLINK-V3EC
JP1	BOOT0 selection	[1-2]	Boot pin flash/serial selection
JP2	BOOT1 selection	[1-2]	Boot in flash mode
JP3	STLK_RST	OFF	-

# FSBL Download



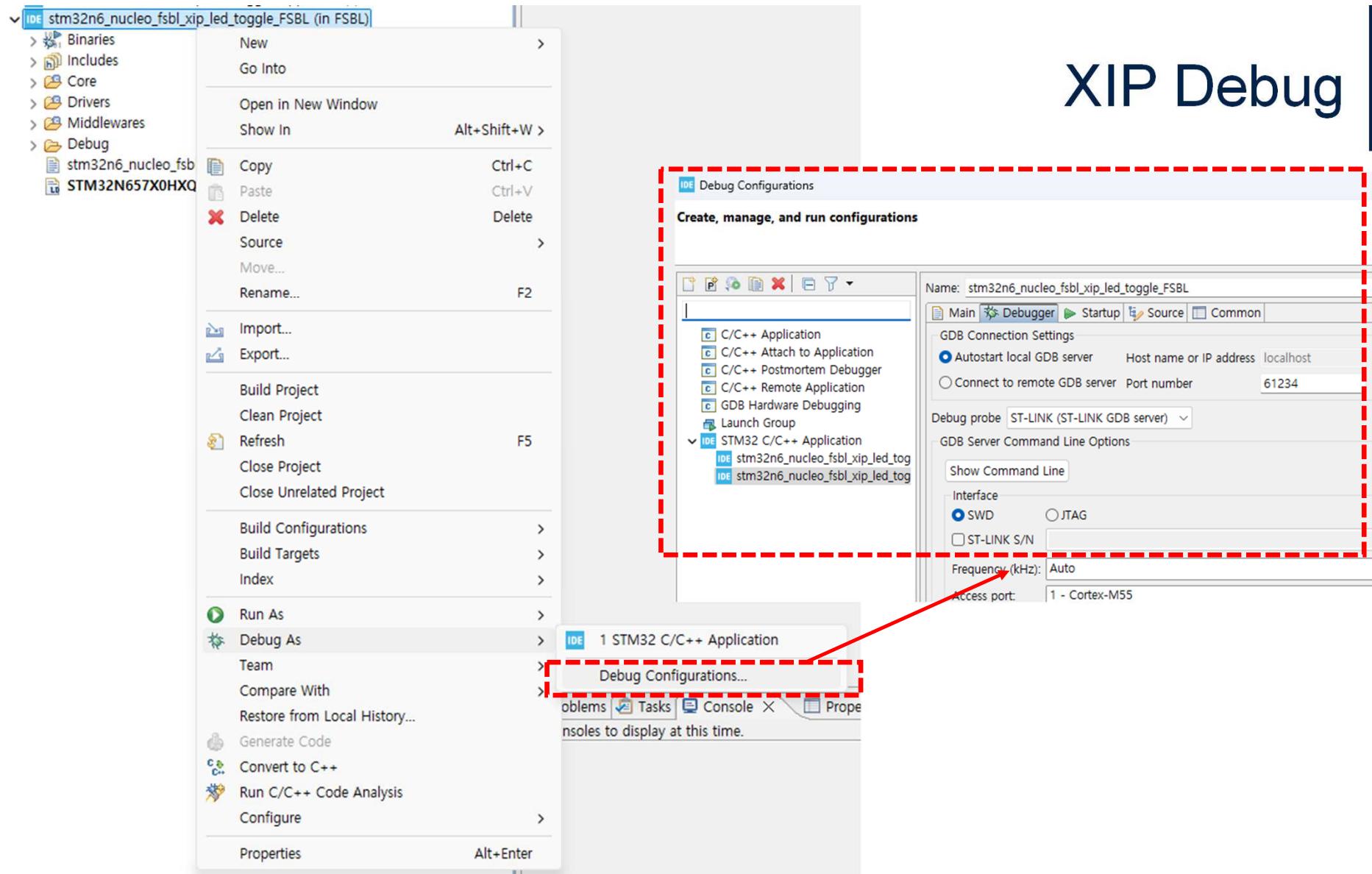
Write the **stm32n6\_nucleo\_fsbl\_xip\_led\_toggle\_FSBL-Trusted**.bin file to 0x70000000.

# APP Download

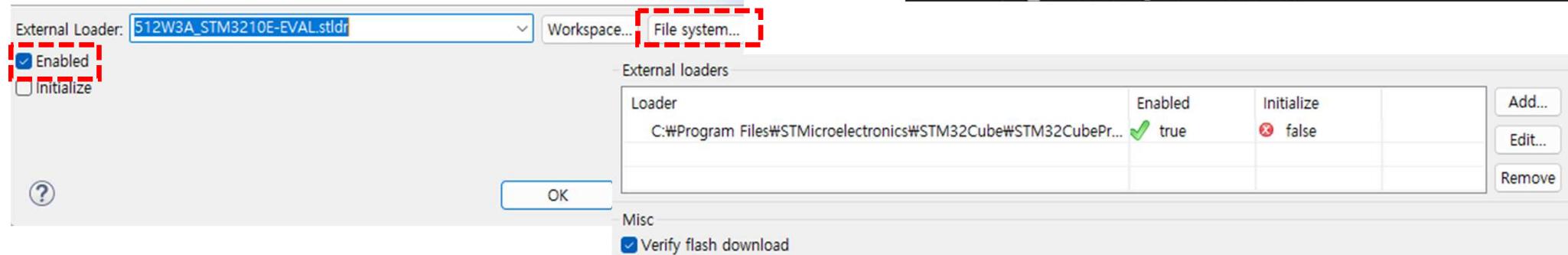
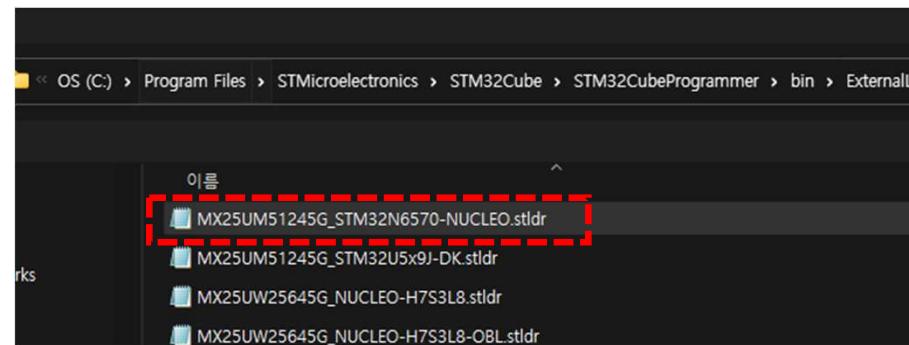
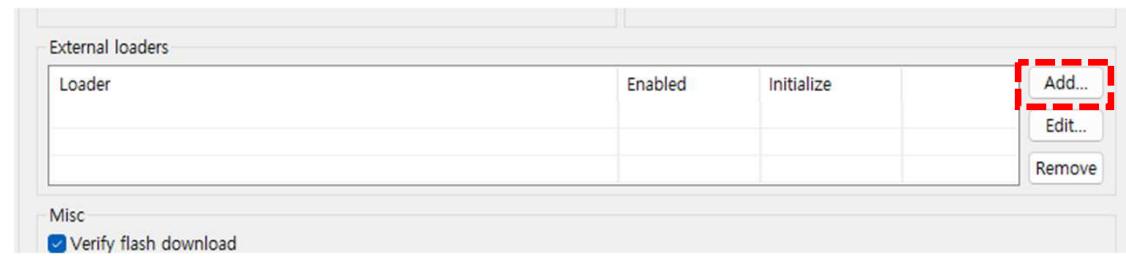


Write the **stm32n6\_nucleo\_fsbl\_xip\_led\_toggle\_Appi-Trusted**file to **0x70020000**.

# XIP Debug



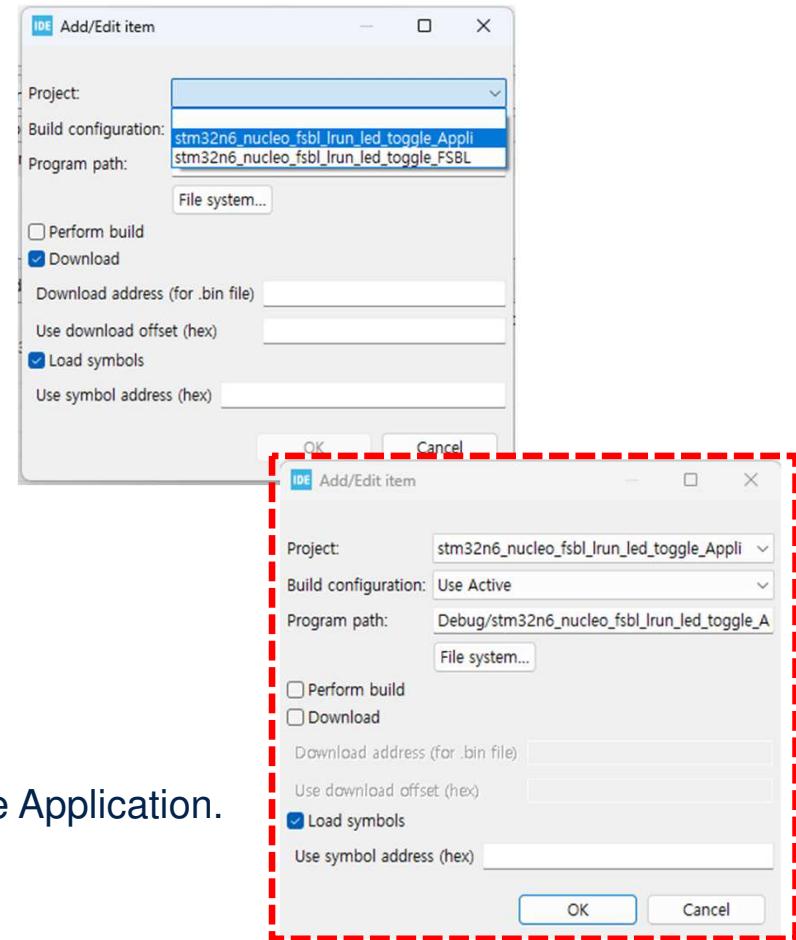
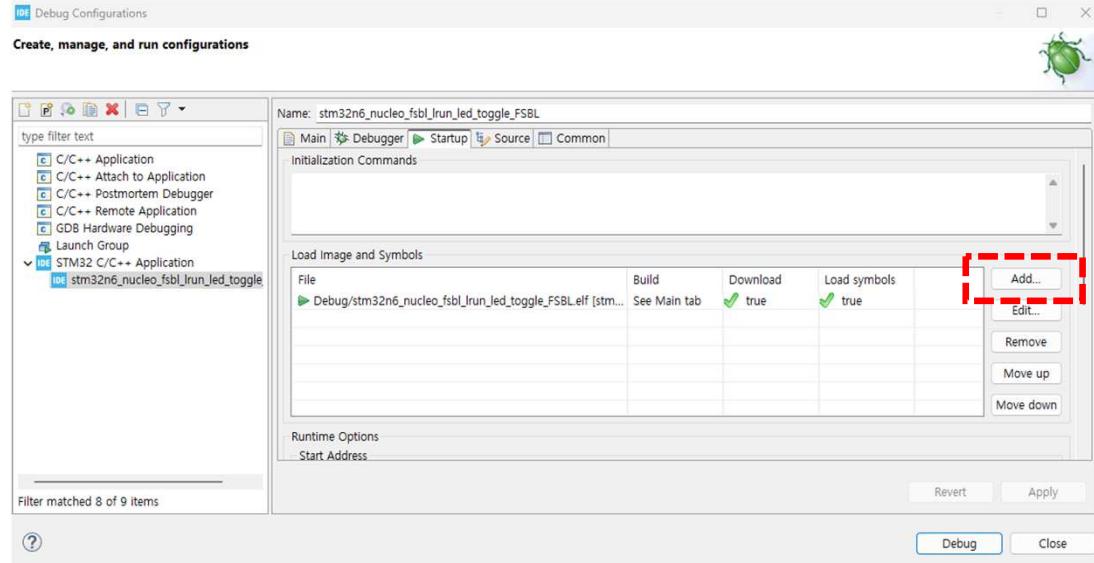
# XIP Debug



C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\ExternalLoader



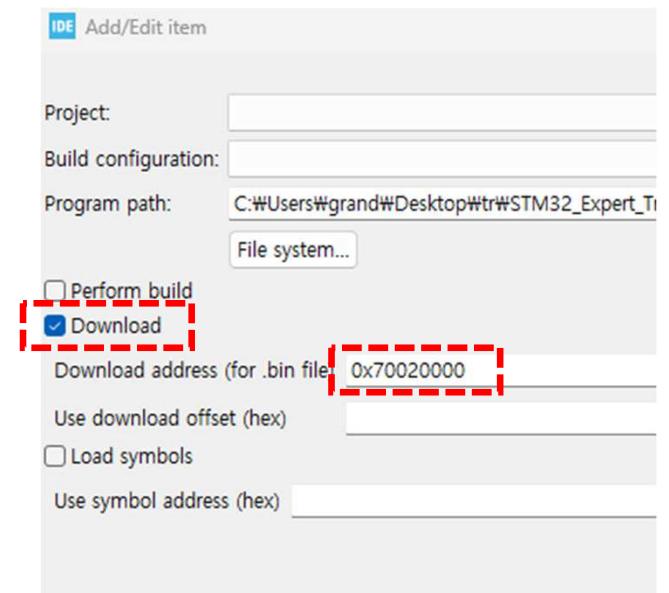
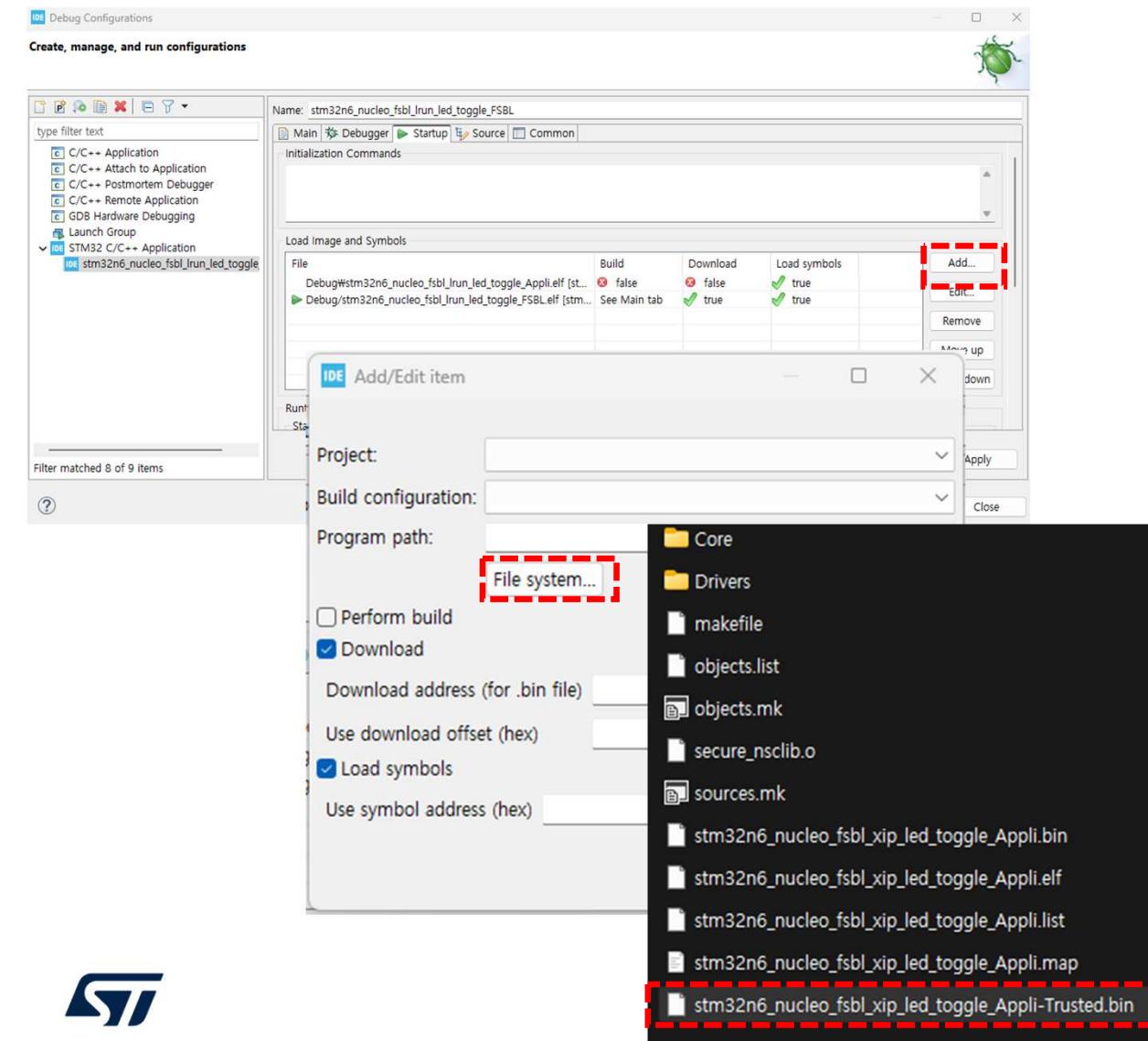
# XIP Debug



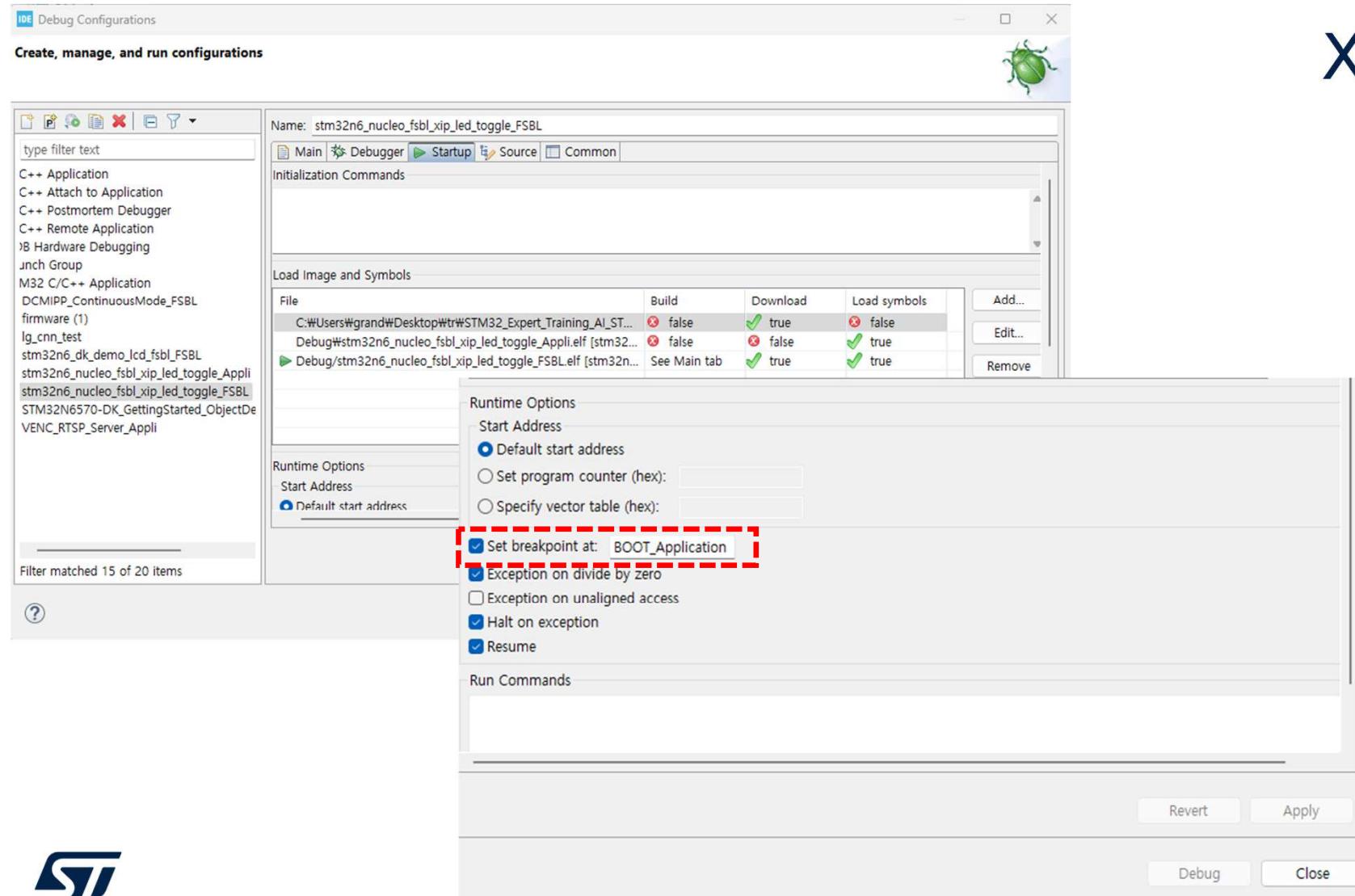
Only add the symbols of the Application.



# XIP Debug



# XIP Debug



# XIP Debug

The screenshot shows a debugger interface with the following components:

- Project Explorer:** Shows the project "stm32n6\_nucleo\_fsb1\_lrun\_led\_toggle\_FSB1 [STM32 C/C++ Application]" with a suspended thread.
- Code Editor:** Displays the main.c file with C code. The assembly code is visible at the bottom of the editor window.
- Registers:** A panel showing CPU registers.
- Stack:** A panel showing the stack.
- Variables:** A panel showing variable values.
- Breakpoints:** A panel showing current breakpoints.
- Expressions:** A panel showing evaluated expressions.
- Disassembly:** A panel showing the assembly code corresponding to the C code in the editor.

The assembly code in the Disassembly panel is as follows:

```
34000801: ldr    r0, [pc, #24]  @ (0x3400081c <main+56
34000803: bl     0x340014c4 <HAL_GPIO_TogglePin>
34000807: mov.w  r1, #1024   @ 0x400
3400080b: ldr    r0, [pc, #16]  @ (0x3400081c <main+56
3400080d: bl     0x340014c4 <HAL_GPIO_TogglePin>
34000811: movs   r0, #200    @ 0xc8
34000813: bl     0x34000f1c <HAL_Delay>
34000816: HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
34000818: nop
3400081a: b.n   0x340007f4 <main+16>
3400081c: adds   r0, r0, r0
3400081e: ldrsb  r2, [r0, r0]
110 {
SystemIsolation_Config:
34000820: push   {r7, lr}
34000822: add    r7, sp, #0
117 __HAL_RCC_RIFSC_CLK_ENABLE();
34000824: mov.w  r0, #512   @ 0x200
34000828: bl     0x3400078c <LL_AHB3_GRP1_EnableClock>
122 HAL_GPIO_ConfigPinAttributes(GPIOB,GPIO_PIN_12,
3400082c: movw   r2, #769   @ 0x301
34000830: mov.w  r0, #4096  @ 0x1000
34000834: ldr    r0, [pc, #48] @ (0x34000868 <SystemI:
34000836: bl     0x340014f8 <HAL_GPIO_ConfigPinAttribut:
123 HAL_GPIO_ConfigPinAttributes(GPIOG,GPIO_PIN_0,
3400083a: movw   r2, #769   @ 0x301
3400083e: movs   r1, #1
34000840: ldr    r0, [pc, #40] @ (0x3400086c <SystemI:
34000842: bl     0x340014f8 <HAL_GPIO_ConfigPinAttribut:
124 HAL_GPIO_ConfigPinAttributes(GPIOG,GPIO_PIN_8,
34000846: movw   r2, #769   @ 0x301
3400084a: mov.w  r1, #256   @ 0x100
3400084e: ldr    r0, [pc, #28] @ (0x3400086c <SystemI:
34000850: bl     0x340014f8 <HAL_GPIO_ConfigPinAttribut:
125 HAL_GPIO_ConfigPinAttributes(GPIOA,GPIO_PIN_10,
```



# **Convolution Neural Network**



# What is AI?

## The evolution of AI

### Artificial intelligence (AI)

Early artificial intelligence stirs excitement



### Machine learning (ML)

Machine learning begins to develop



### Deep learning

Deep learning breakthroughs drive AI boom



Any technique that enables computer to mimic **human behavior**

**Subset of AI.** Algorithms and methodologies that improve over time through **learning from data**

**Subset of ML.** Learning algorithms that derive meaning from **massive amounts of data** by using a hierarchy of multiple layers that **mimic the neural networks of the human brain**

1950

1960

1970

1980

1990

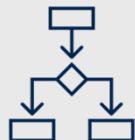
2000

2010

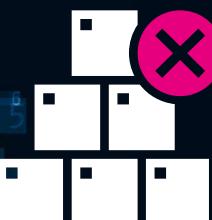
2020



# AI offers the best approach to process growing amounts of data



Algorithms and **predefined models** to analyze data and make predictions or decisions

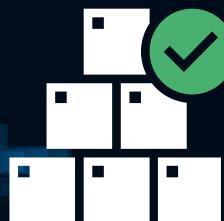


**Traditional approaches show their limitations:**

- when dealing with **large datasets**
- when the **phenomena are too complex**



Machine learning algorithms to automatically **learn** patterns and relationships **from the data**

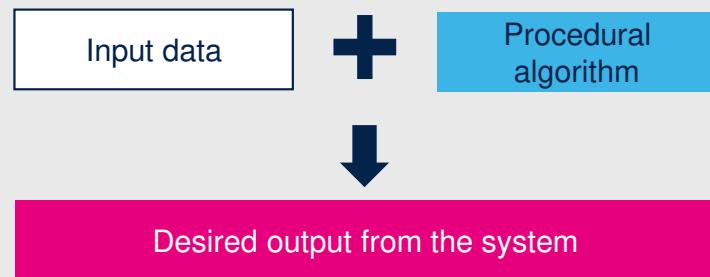


**AI-based data processing offers a more flexible and powerful approach** to analyzing and making decisions from large data collection

# A new way to add environment awareness to your products

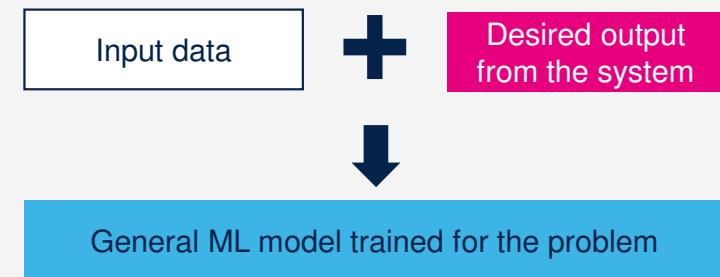
Create more robust software using machine learning on STM32

Standard programming  
Handcrafted rules based on experience



- Requires domain expertise to code
- Need to rewrite if environment evolves

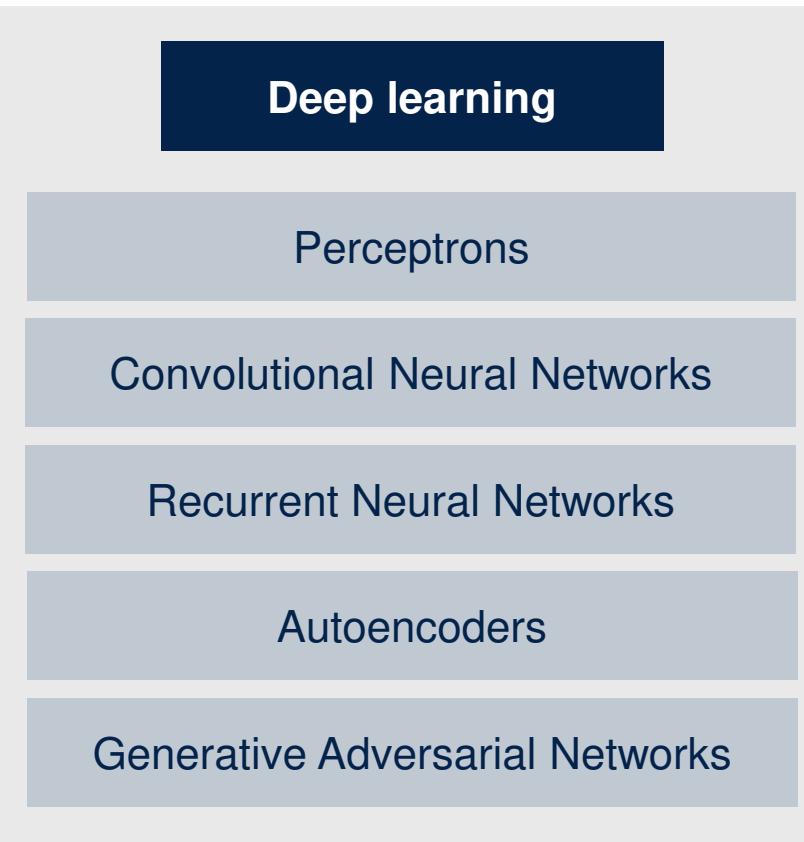
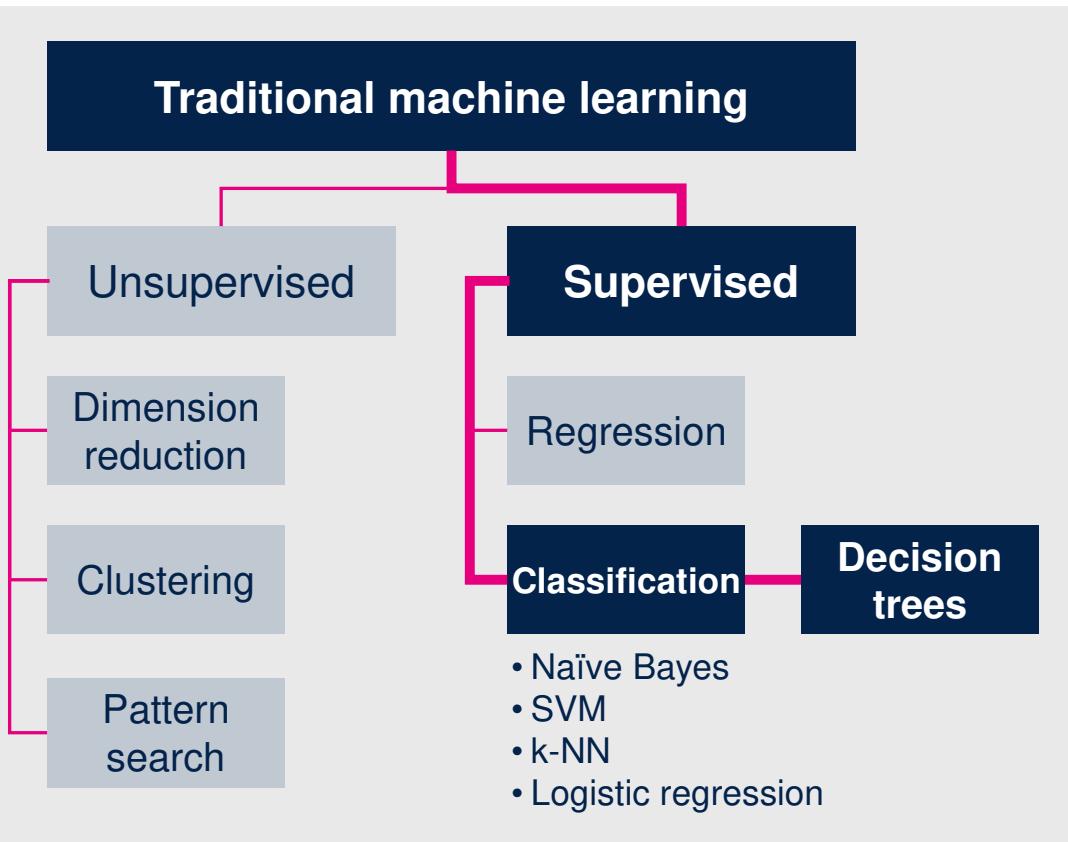
Machine learning  
Rules learned from real-world data



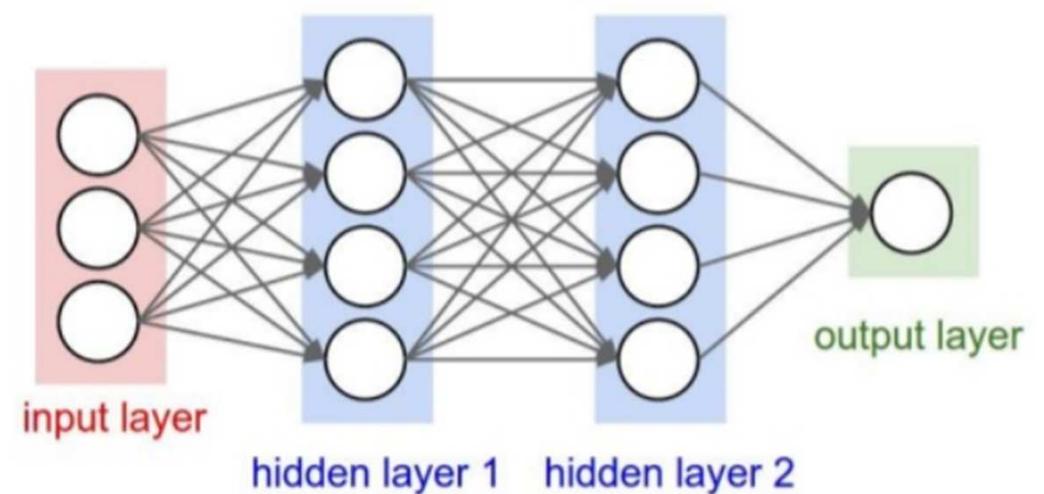
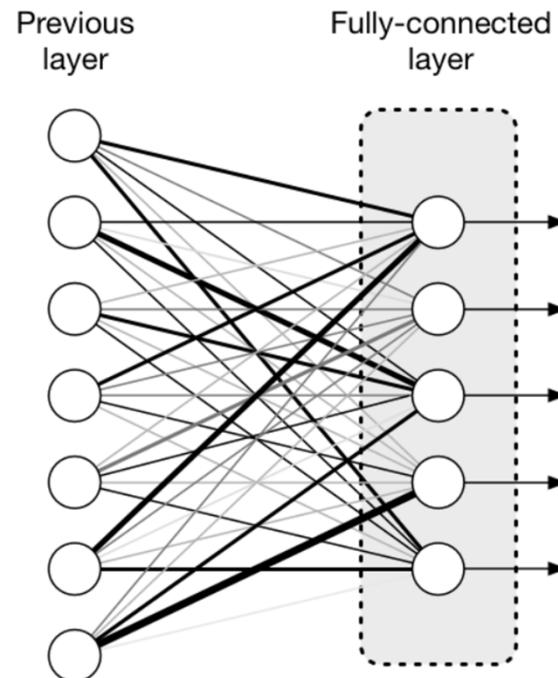
- Generate code from real-world observations
- Relearn from data if environment evolves



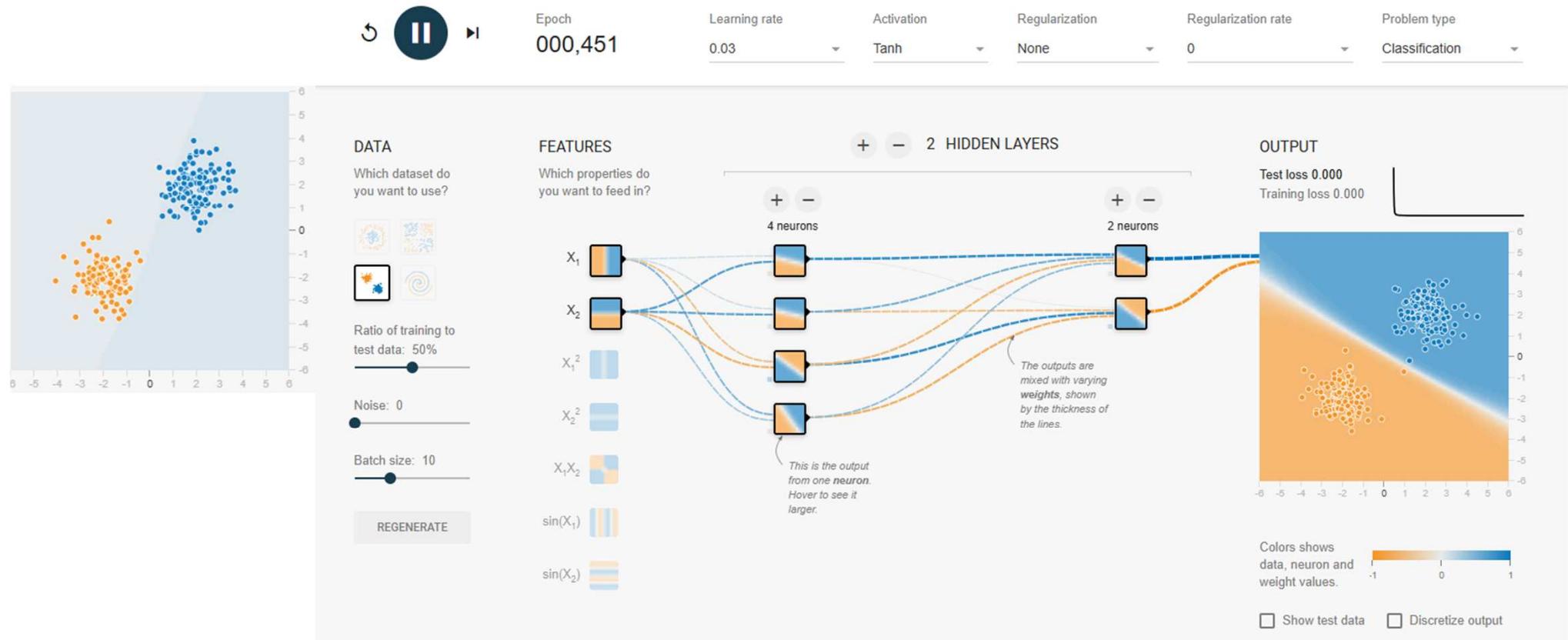
# Machine learning



# What is a Neural Network?

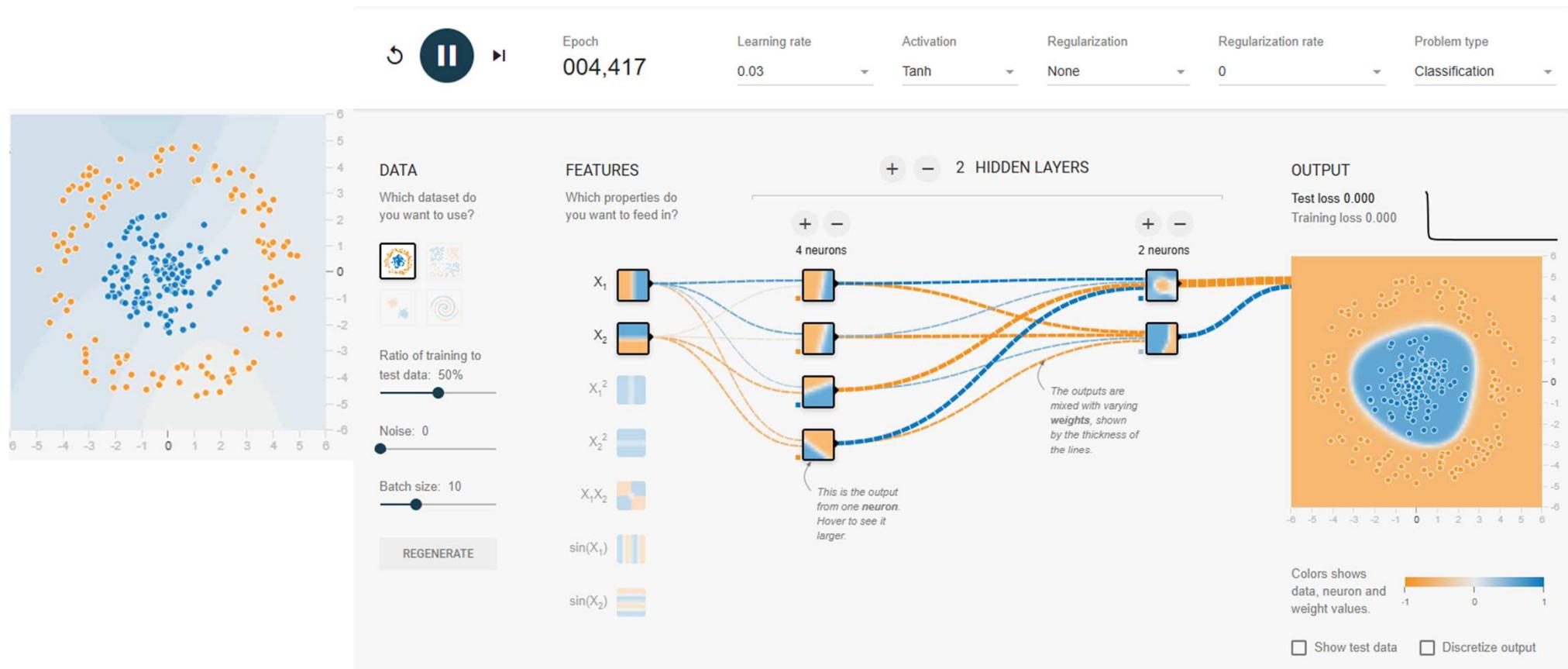


# What is a Neural Network?



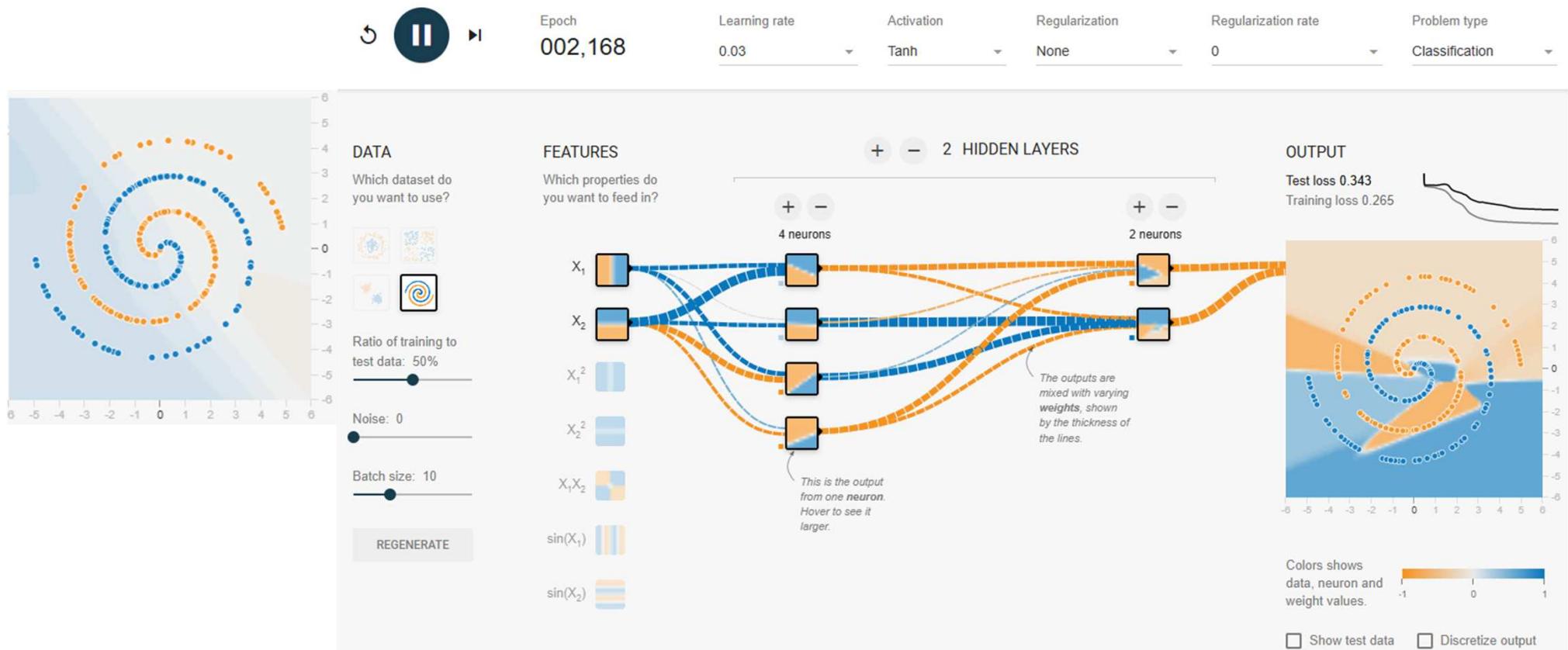
<https://playground.tensorflow.org>

# What is a Neural Network?



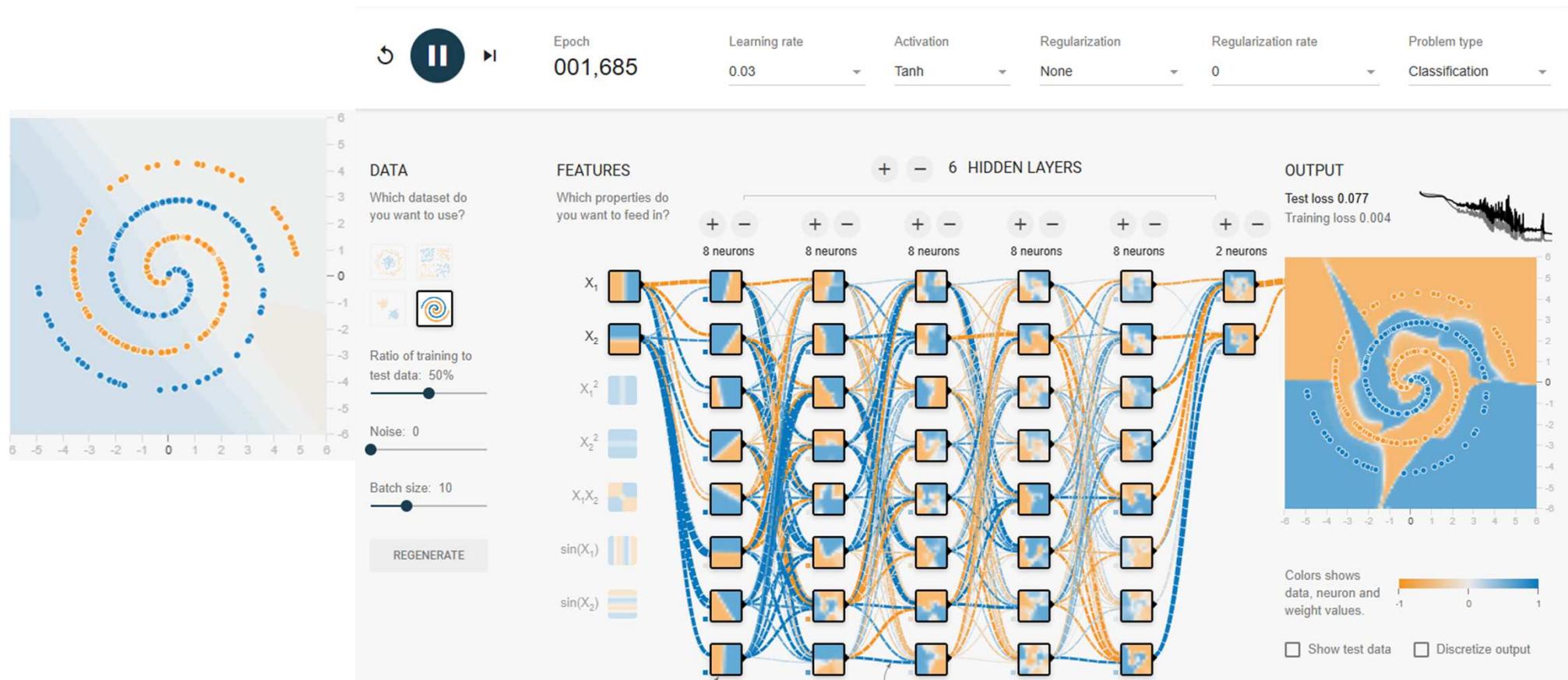
<https://playground.tensorflow.org>

# What is a Neural Network?



<https://playground.tensorflow.org>

# What is a Neural Network?



<https://playground.tensorflow.org>



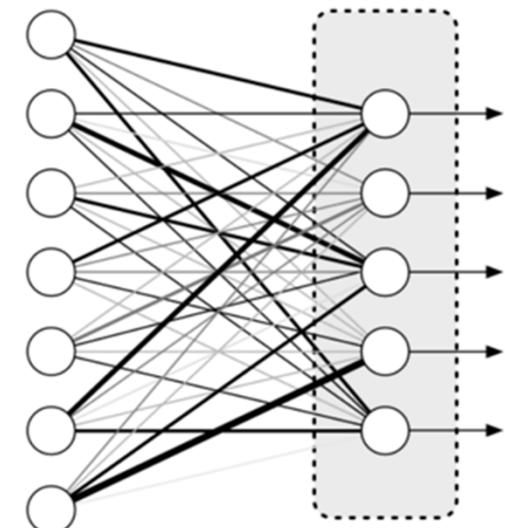
# Convolution Neural Network

## Hands on



# Convolution Neural Network

## Hands on



# Convolution Neural Network

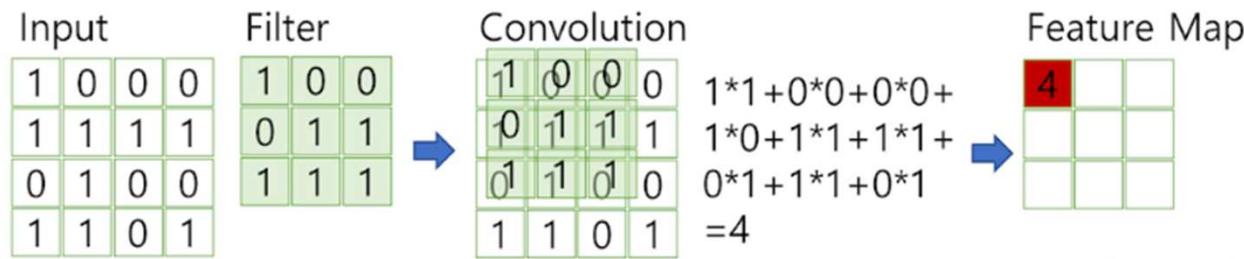
Channel



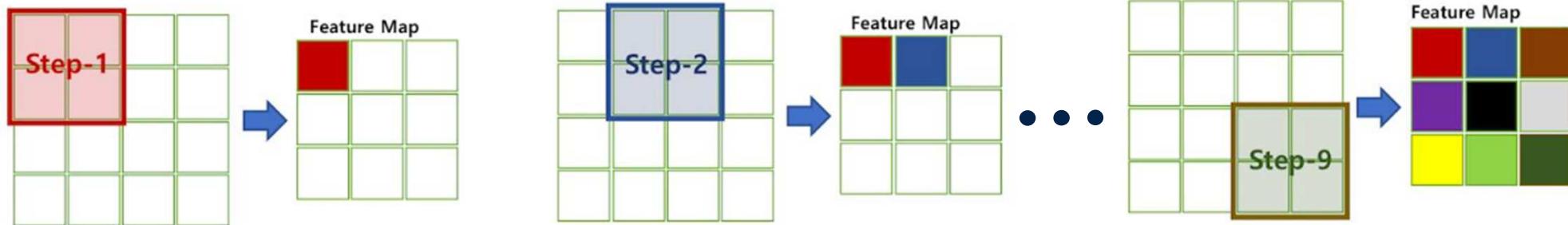
[https://en.wikipedia.org/wiki/Channel\\_\(digital\\_image\)](https://en.wikipedia.org/wiki/Channel_(digital_image))

# Convolution Neural Network

## Filter and Stride



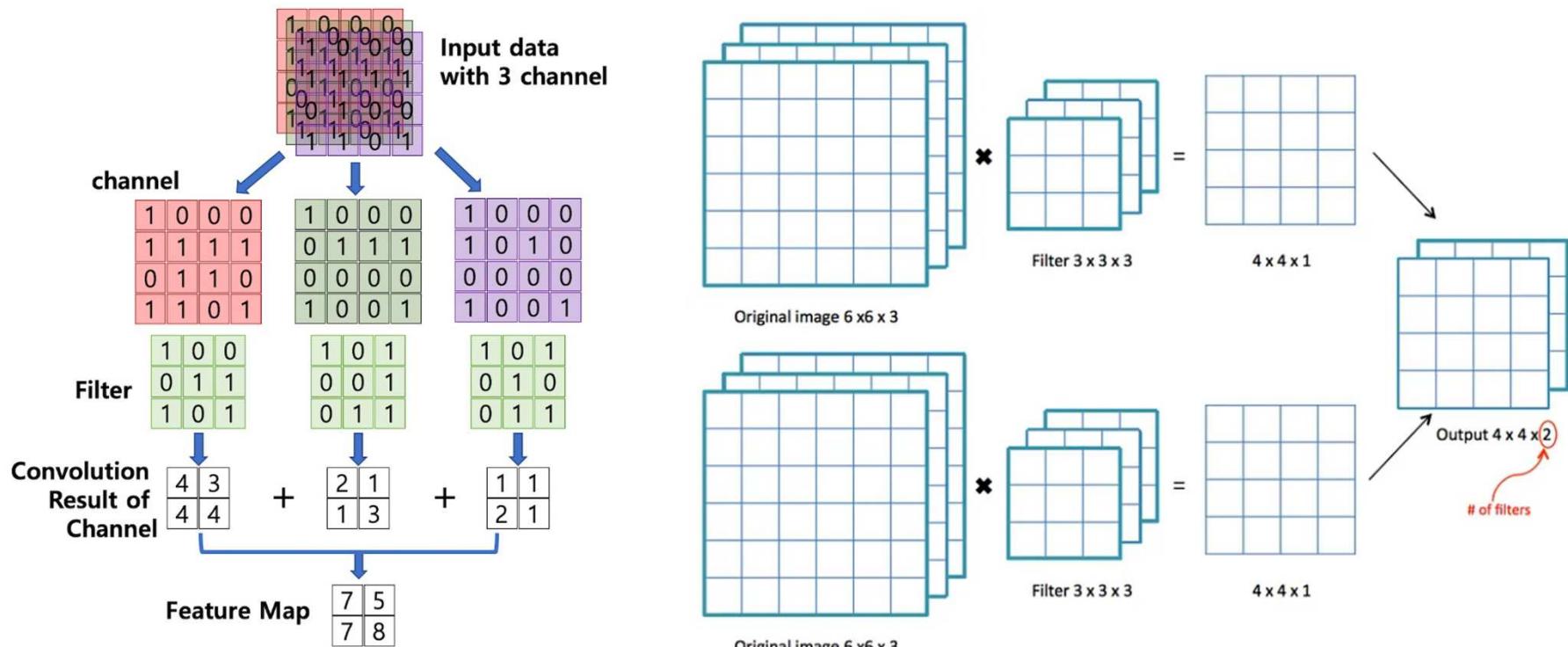
<http://taewan.kim>



<http://taewan.kim/post/cnn/>

# Convolution Neural Network

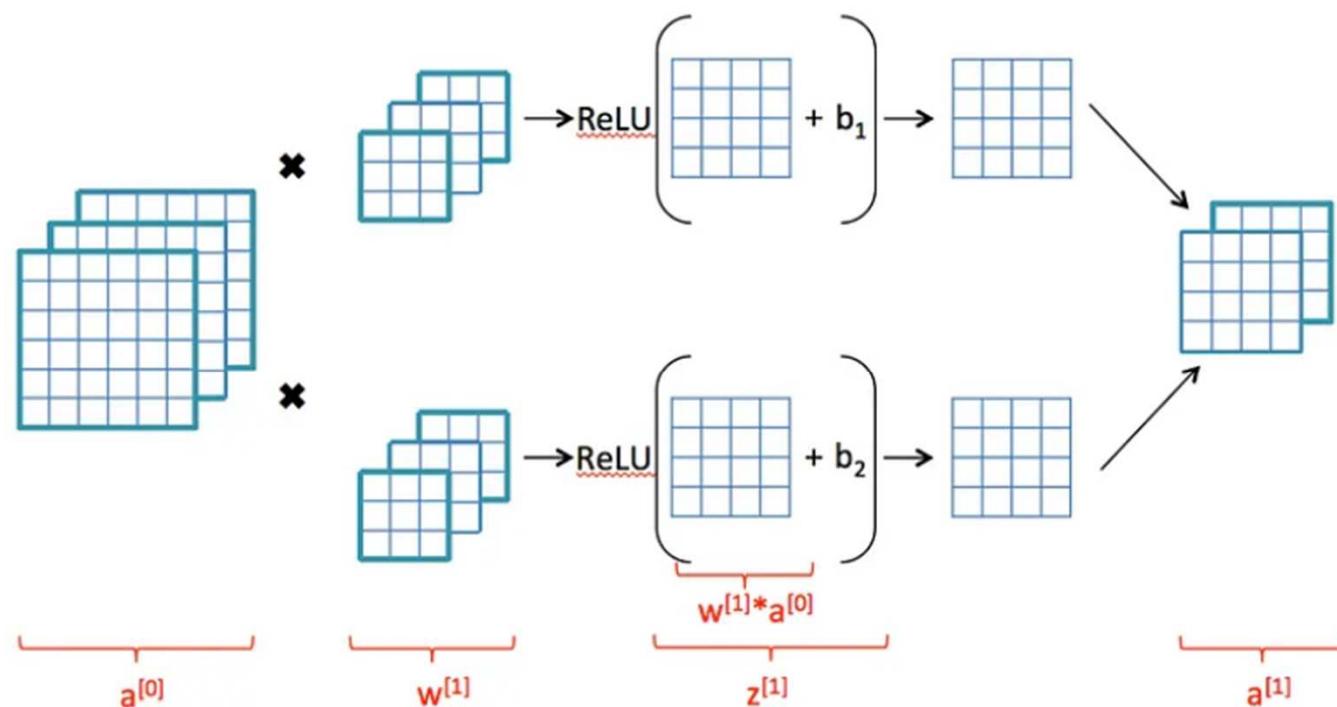
## Hands on



<http://taewan.kim/post/cnn/> , <https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524>

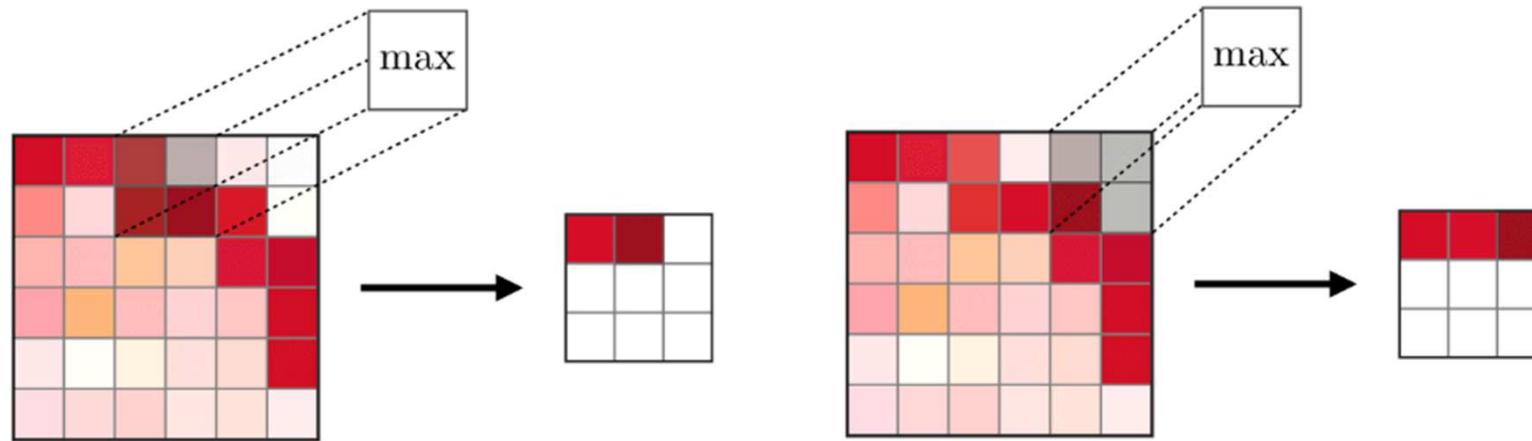
# Convolution Neural Network

Hands on



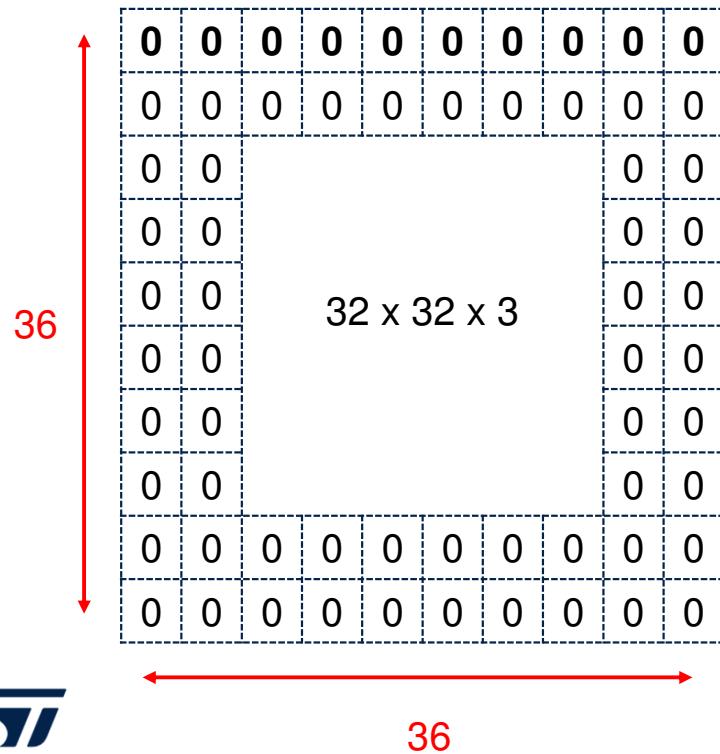
# Convolution Neural Network

## Pooling

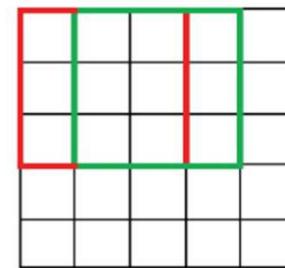


# Convolution Neural Network

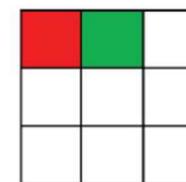
## Padding and Stride



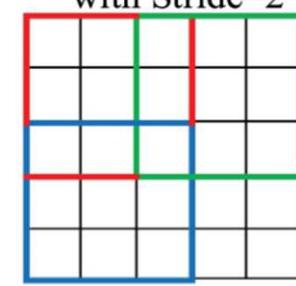
Convolution  
with Stride=1



Output



Convolution  
with Stride=2

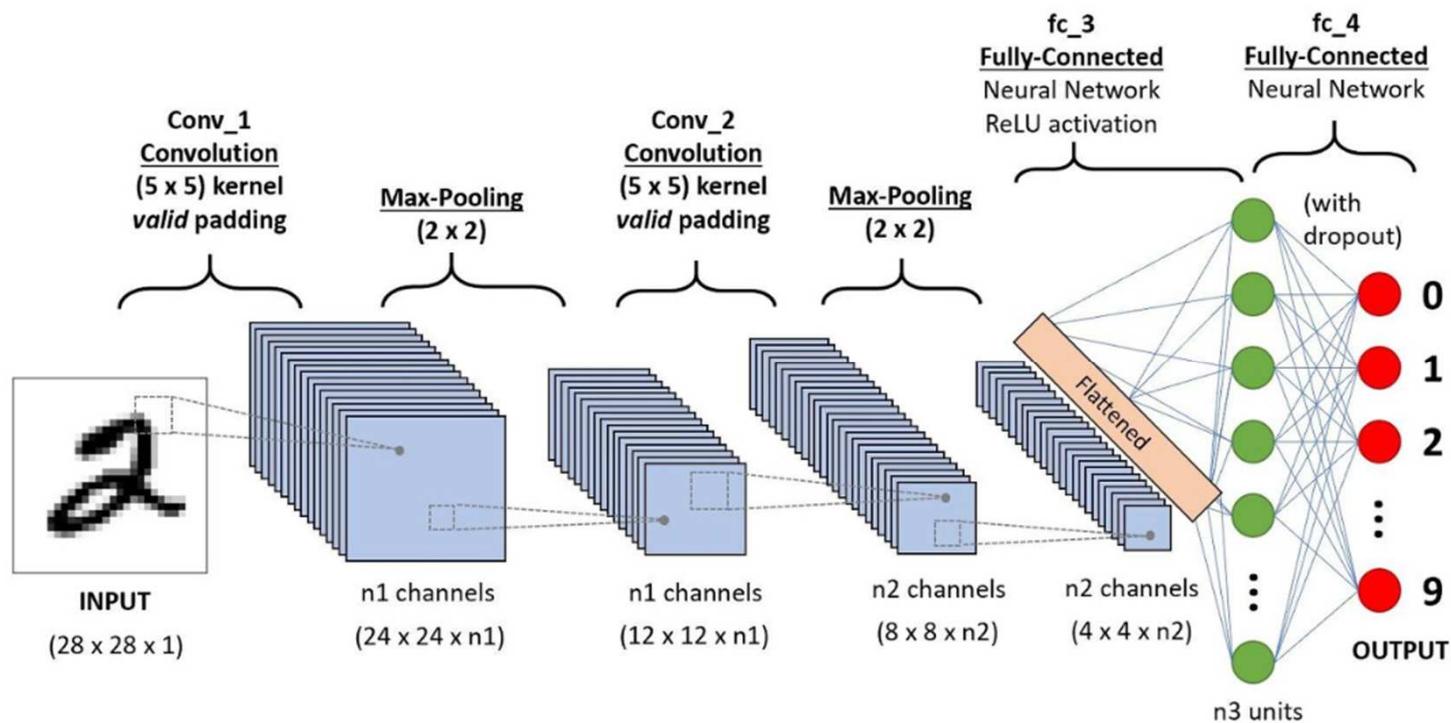


Output



# Convolution Neural Network

Hands on



# Convolution Neural Network

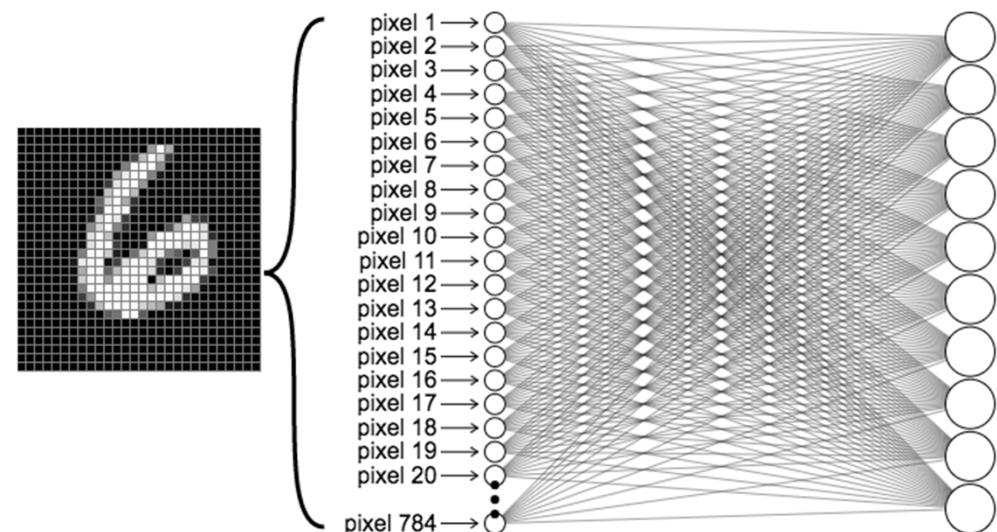
## Hands on

		Activation Shape	Activation Size	#Parameters
1	Input Layer :	(32, 32, 3)	3072	0
2	CONV1 (f=5, s=1)	(28, 28, 8)	6272	608
3	POOL1	(14, 14, 8)	1568	0
4	CONV2 (f=5, s=1)	(10, 10, 16)	1600	3216
5	POOL2	(5, 5, 16)	400	0
6	FC3	(120, 1)	120	48120
7	FC4	(84, 1)	84	10164
8	Softmax	(10, 1)	10	850



# Convolution Neural Network

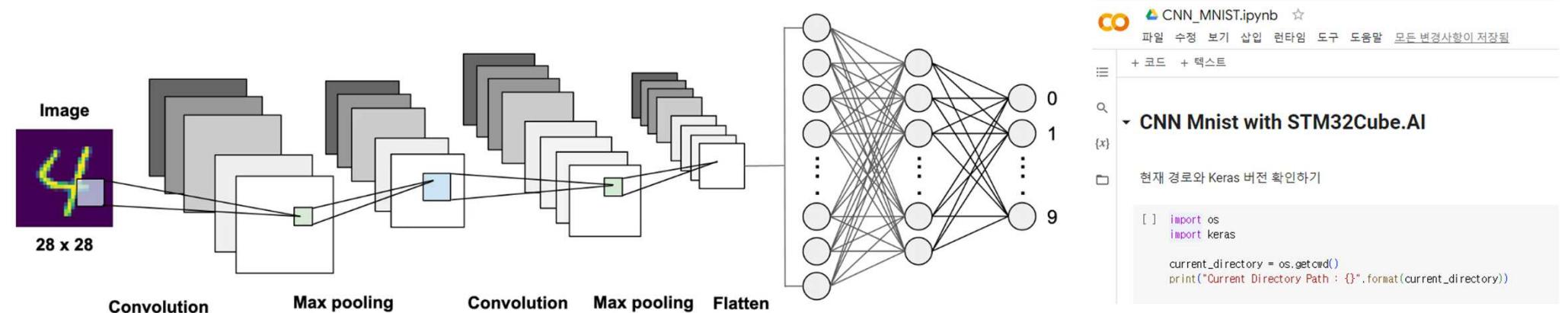
## Hands on – mnist

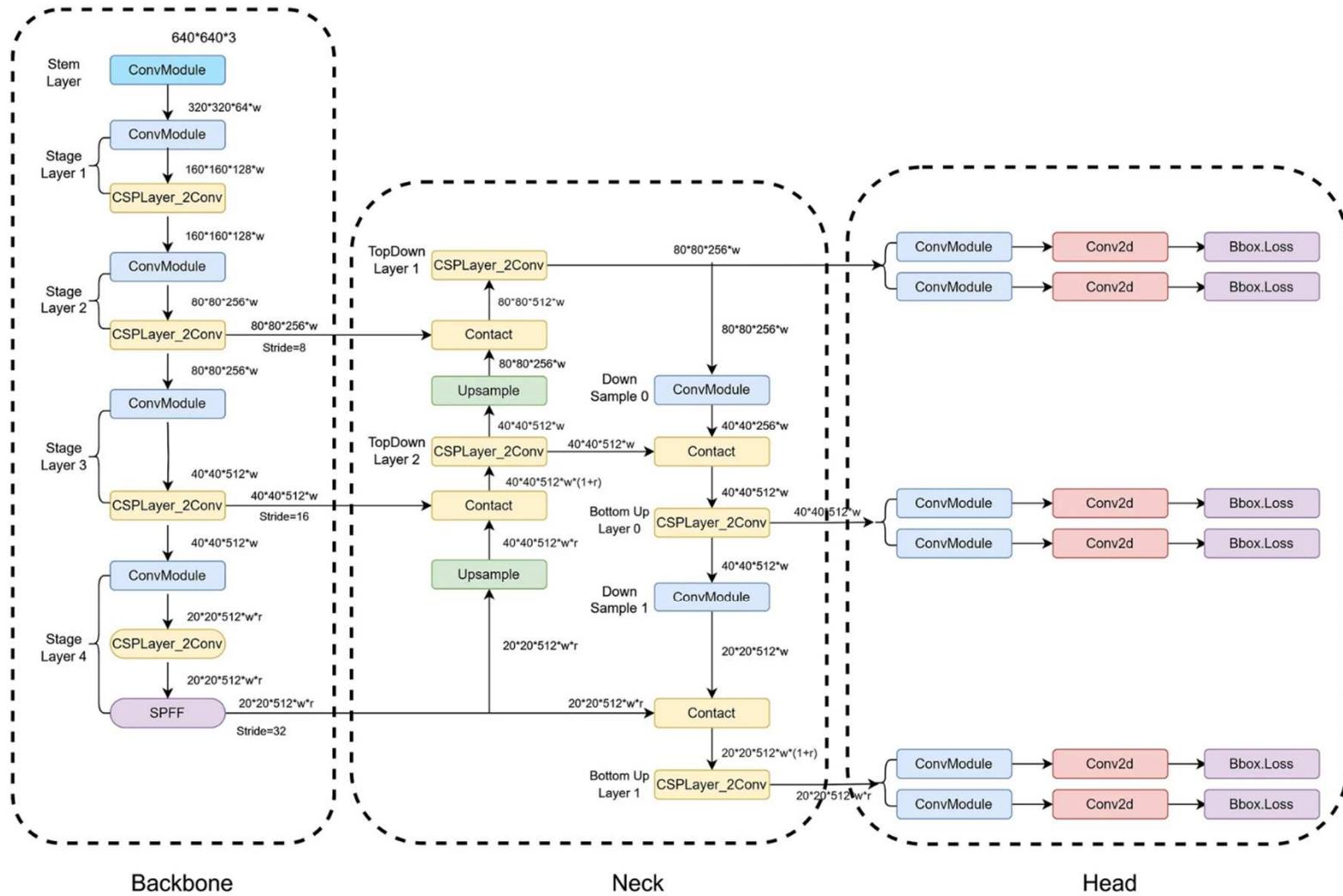


[https://ml4a.github.io/ml4a/looking\\_inside\\_neural\\_nets/](https://ml4a.github.io/ml4a/looking_inside_neural_nets/)

# Convolution Neural Network

## Hands on – mnist





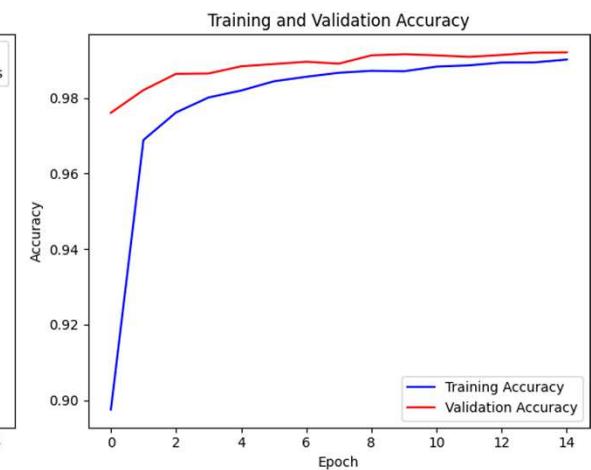
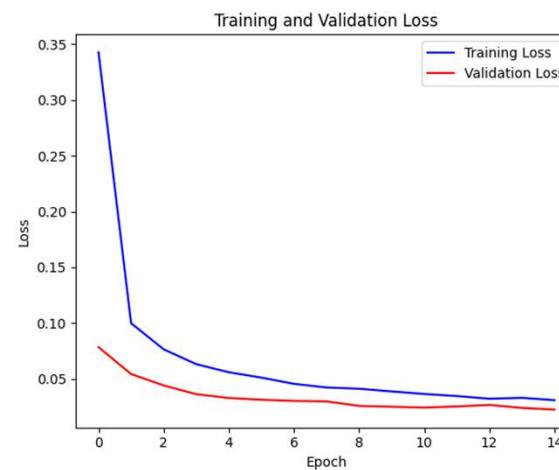
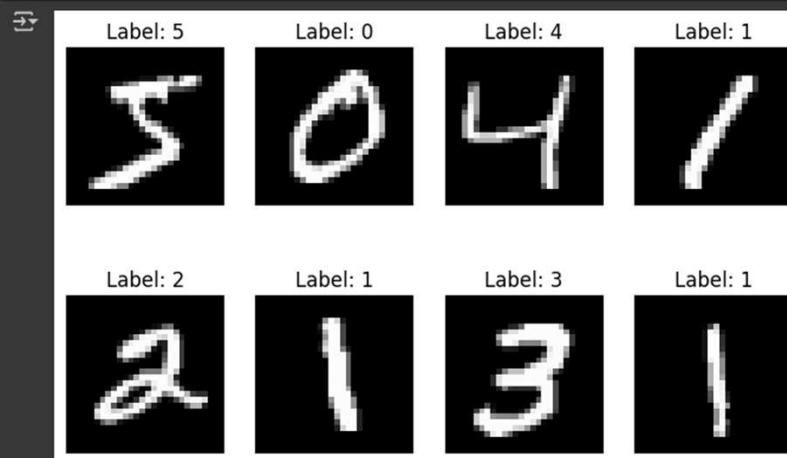
<https://www.nature.com/articles/s40494-024-01144-1/figures/1>

# Training

Mnist 이미지 그려보기

```
[ ] # Function to draw the images
def draw_images(images, labels):
    plt.figure(figsize=(10, 5))
    for i in range(10): # Display the first 10 images
        plt.subplot(2, 5, i + 1)
        plt.imshow(images[i], cmap='gray')
        plt.title("Label: " + str(labels[i]))
        plt.axis('off')
    plt.show()

draw_images(x_train,y_train)
```



You can practice training and quantization based on the MNIST handwritten digit data using example code.

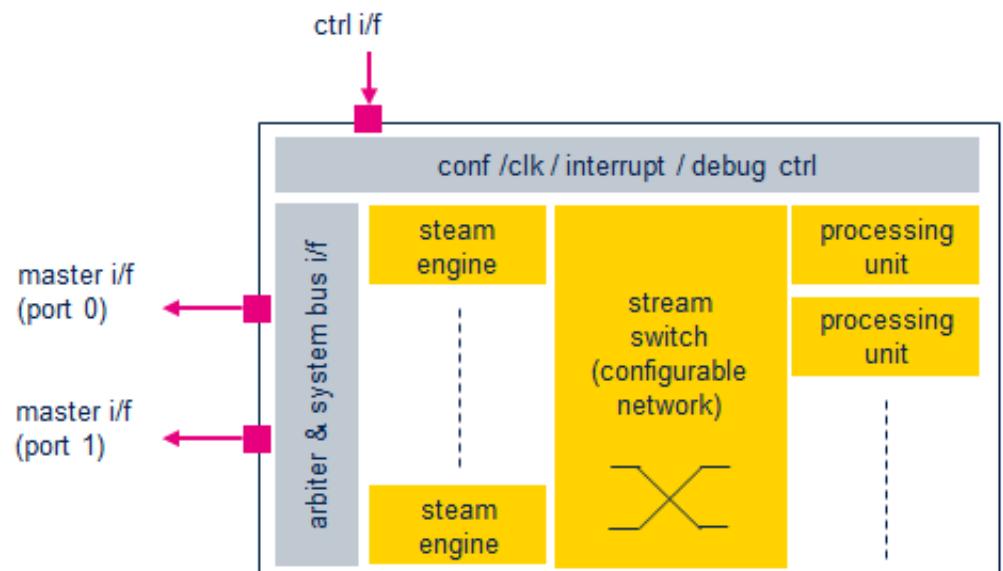
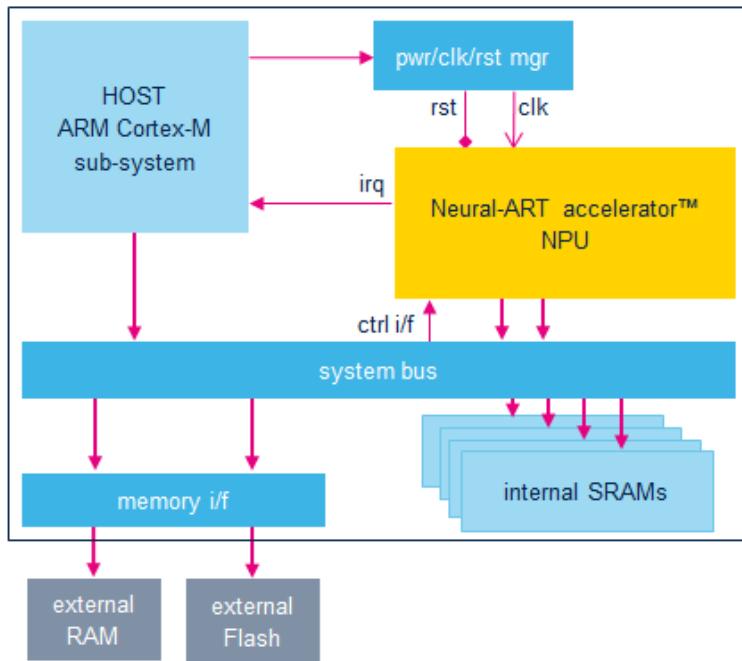


[https://github.com/BlaineMoon30/STM32\\_Expert\\_Training\\_AI\\_STM32N6/tree/master/Hands\\_On\\_ 2\\_Hands\\_On\\_NPU\\_Mnist/Mnist\\_Model/Script](https://github.com/BlaineMoon30/STM32_Expert_Training_AI_STM32N6/tree/master/Hands_On_ 2_Hands_On_NPU_Mnist/Mnist_Model/Script)

# **ST Neural-ART Accelerator™**



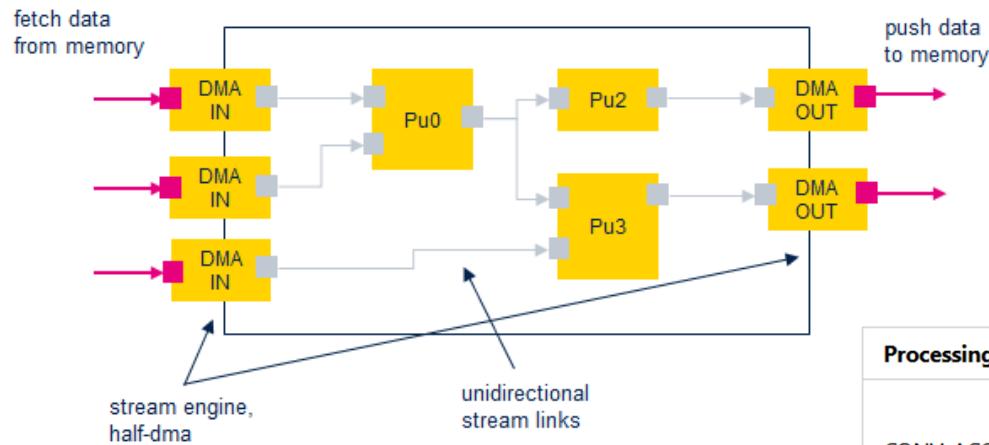
# ST Neural-ART Accelerator™



Neural-Art Accelerator™ is a branded family of design-time parametric and runtime reconfigurable neural processing unit (NPU) cores.



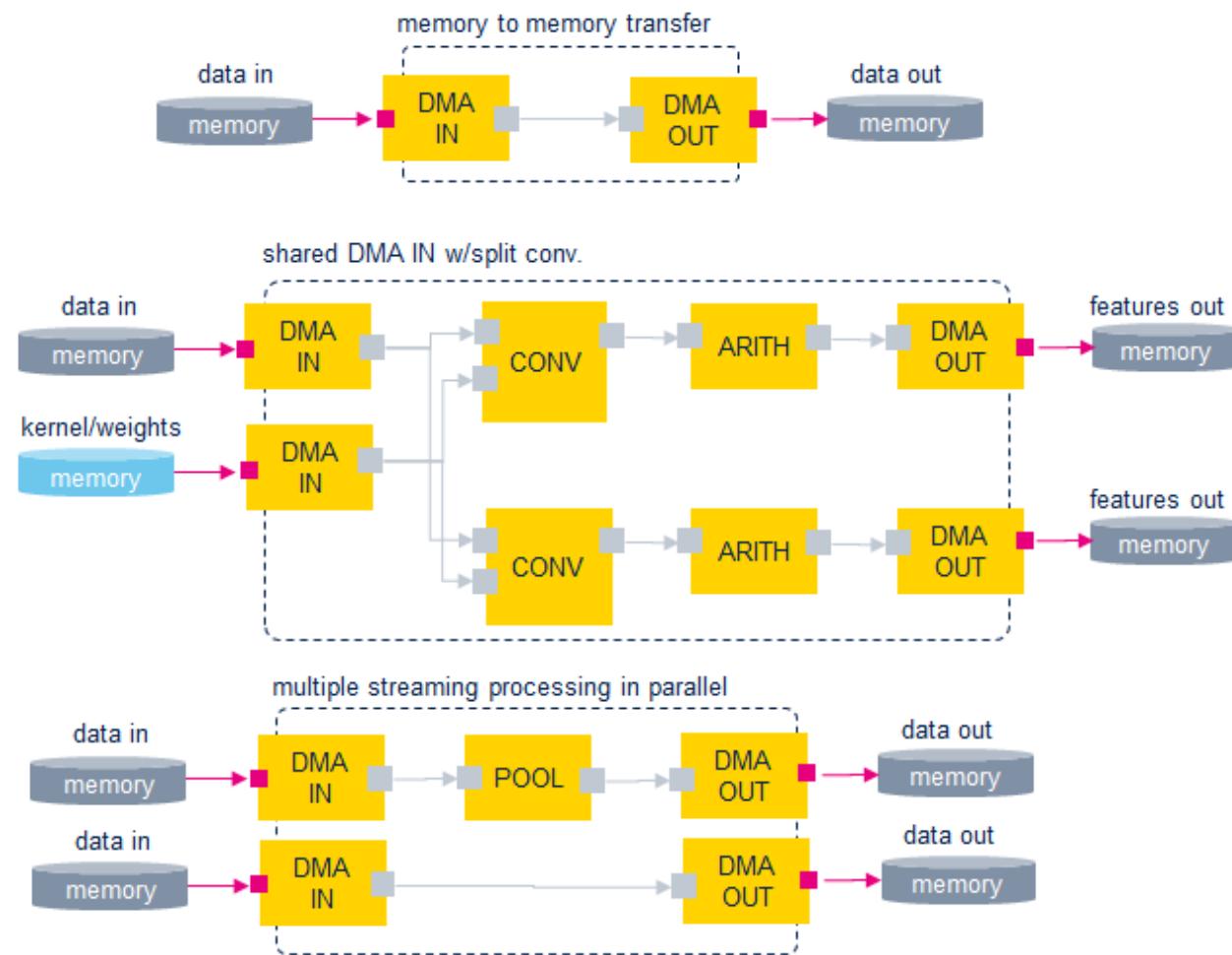
# Streaming-based architecture



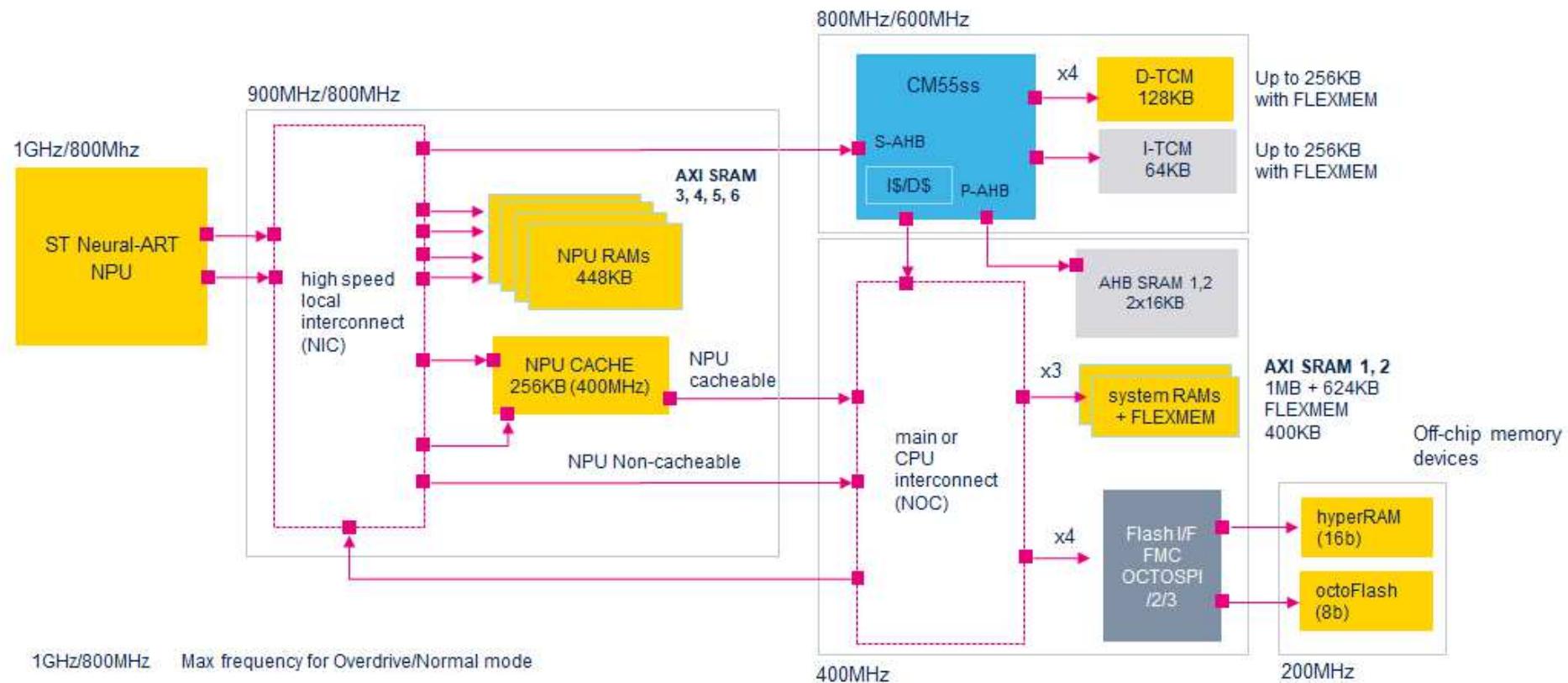
The stream engine units function as intelligent half-DMAs, capable of reading and writing data to and from external memory.

Processing unit	description
CONV_ACC (x4)	The main processing unit performing the convolution operations with up to 72 8x8 multiply accumulate operations per cycle (or 18 16x16 macc/cycle). This allows a theoretical peak processing of $72 \times 4 \times 2$ (addition and multiplication) = 576+ GOPs @1GHz or <b>600+ GOPs</b> including the operations from the ACTIV/ARITH/POOL units.
POOL_ACC (x2)	Perform the pooling operations like local 2D windowed ( $N \times N$ ), min, max, average pooling as well as global max, min, or average pooling.
ACTIV_ACC (x2)	Perform the activation functions associated with the convolutional neural networks: Logistic, TanH, ReLU, PReLU, etc.
ARITH_ACC (x4)	Perform the arithmetic operations: element-wise addition/subtraction/multiplication or any other affine operation.
STREAM_ENG (2x5)	5 per port, Key unit (smart half-DMA engine) to fetch/push data to/from memory subsystem.

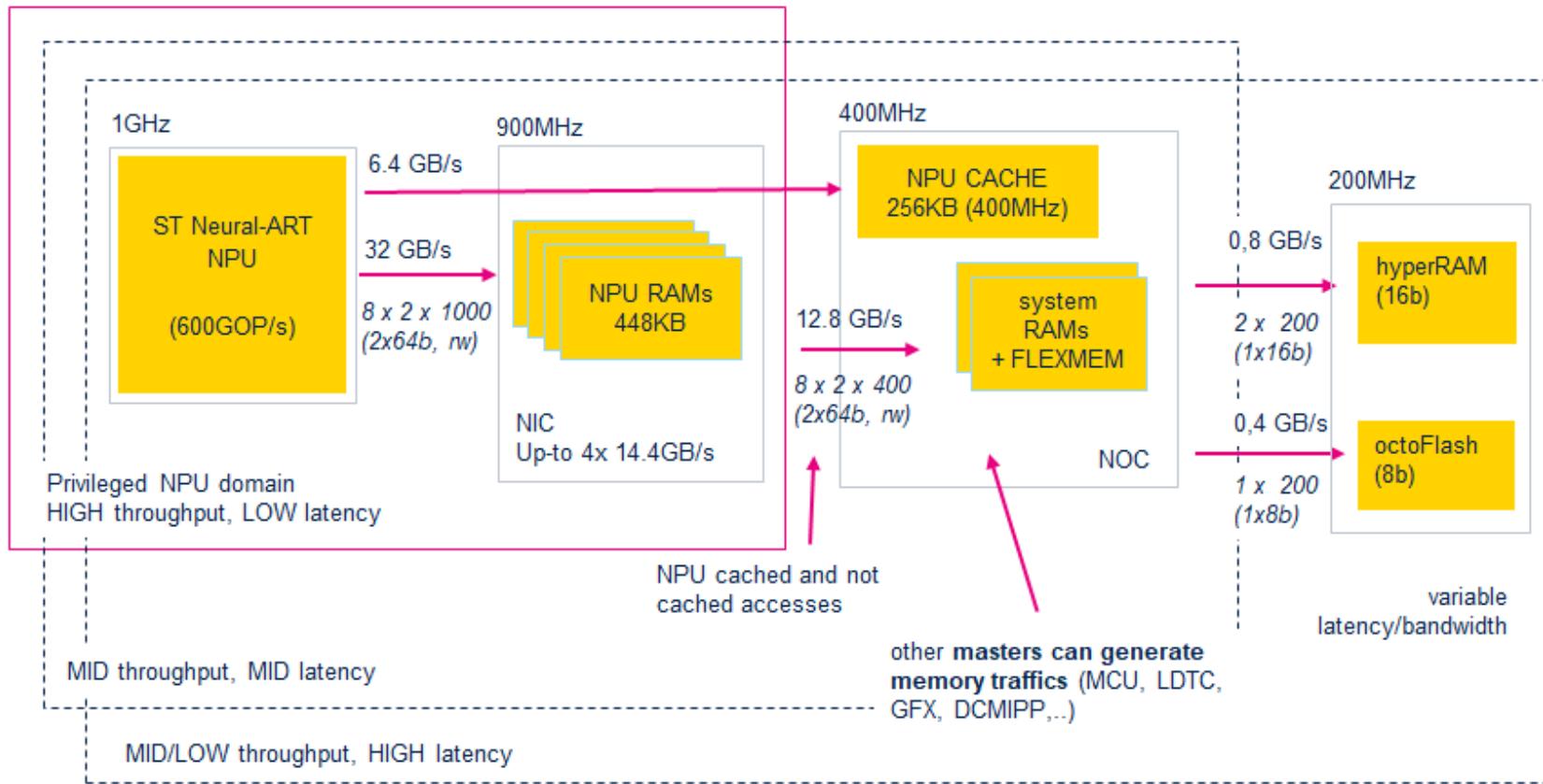
# Data processing paths



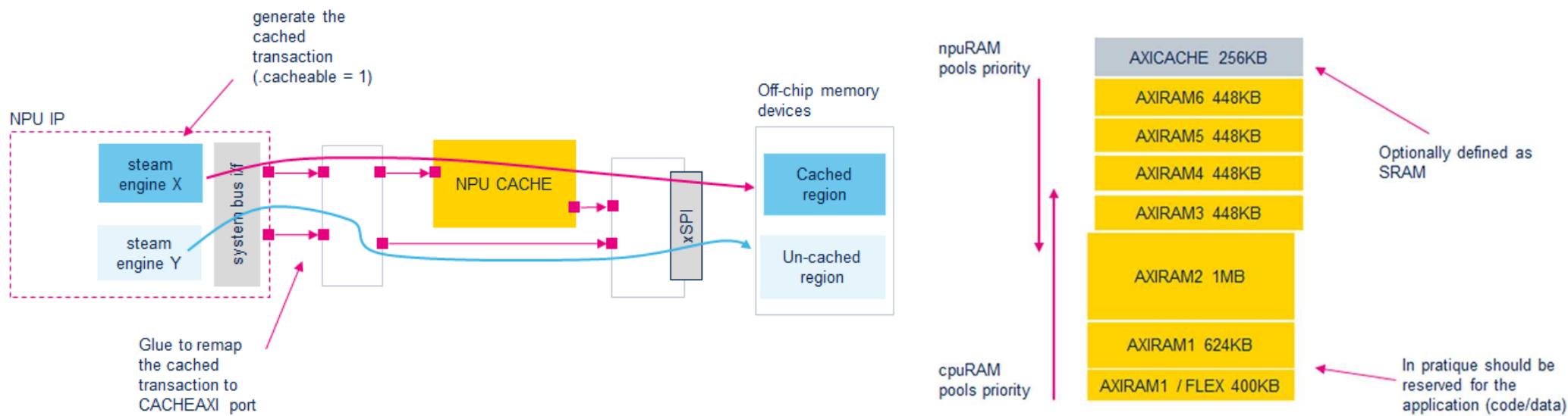
# NPU Memory System



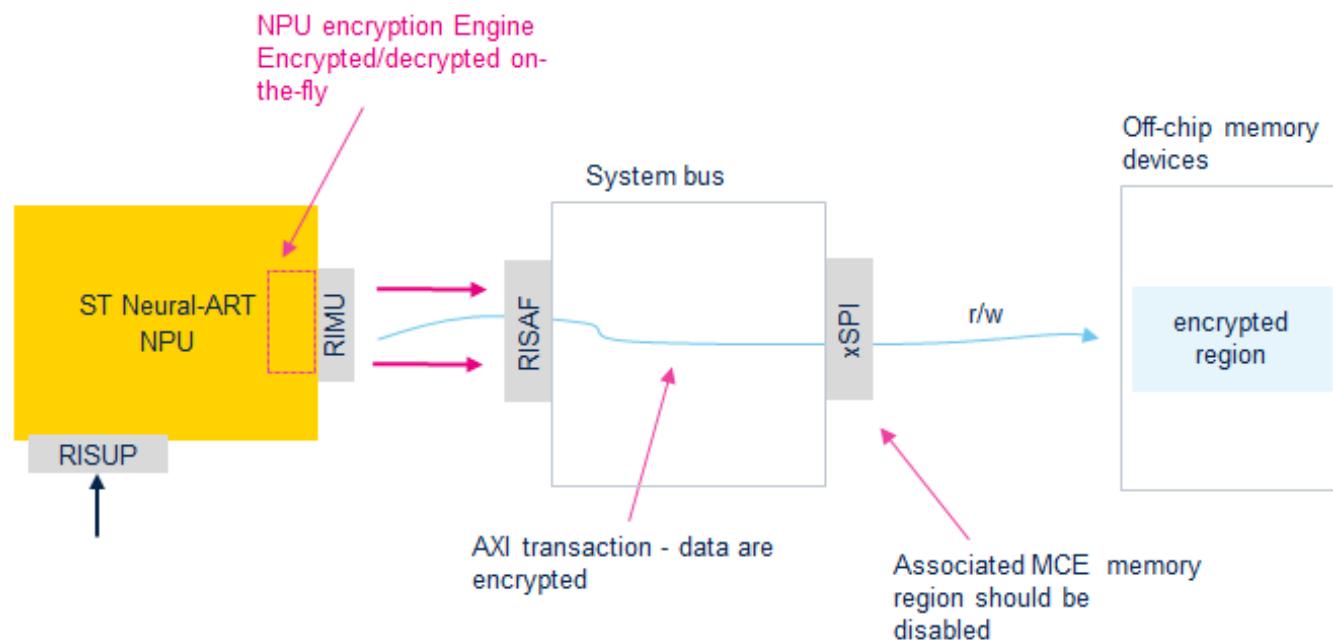
# Processing-Bound vs Memory-Bound Operation



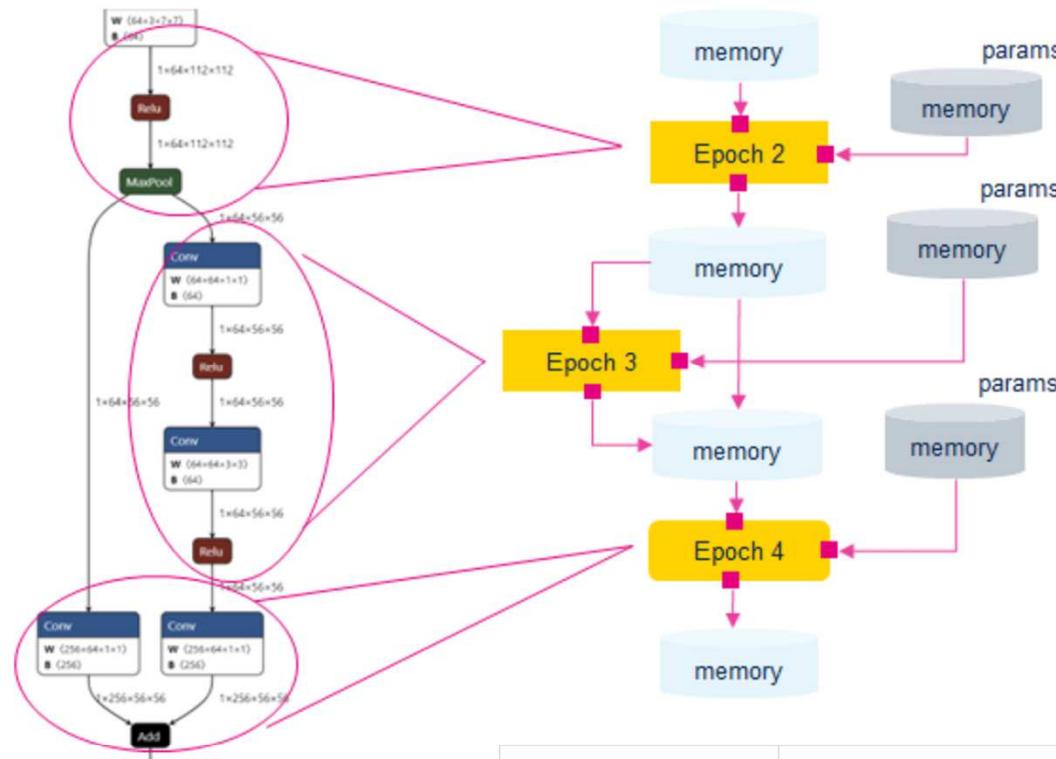
# NPU Cache and Virtual memory pool



# Security considerations



# Programming model



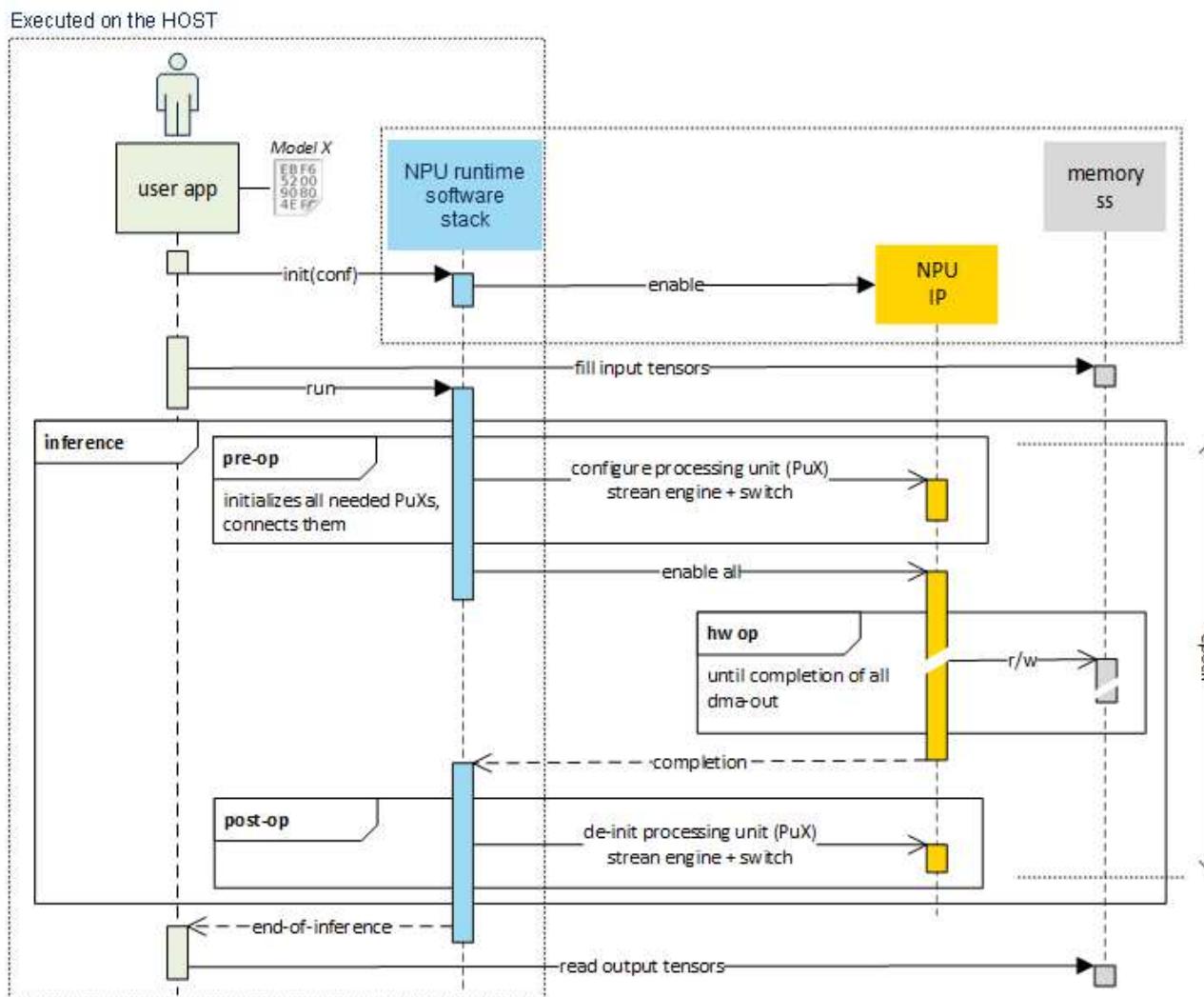
Generally, an entire model cannot fit on the available NPU hardware resources.

It must be split into elementary subsets, named *NPU epochs* (or 'epoch'), which fit the NPU's available resources.

The model is compiled offline, producing the settings for the different epochs needed to execute the whole model.

Name	description
HW epoch	designates a case where the operations related to a part of the model are fully mapped on the NPU HW resources.
SW epoch	designates a case where the operation is delegated on the HOST. It is no-hardware accelerated.
Hybrid epoch	designates a specific case where a part of the operation is executed in software with a support of the predefined HW epochs.
Meta epoch	designates a set of HW epochs controlled by a command stream thanks to the <a href="#">epoch controller unit</a> .

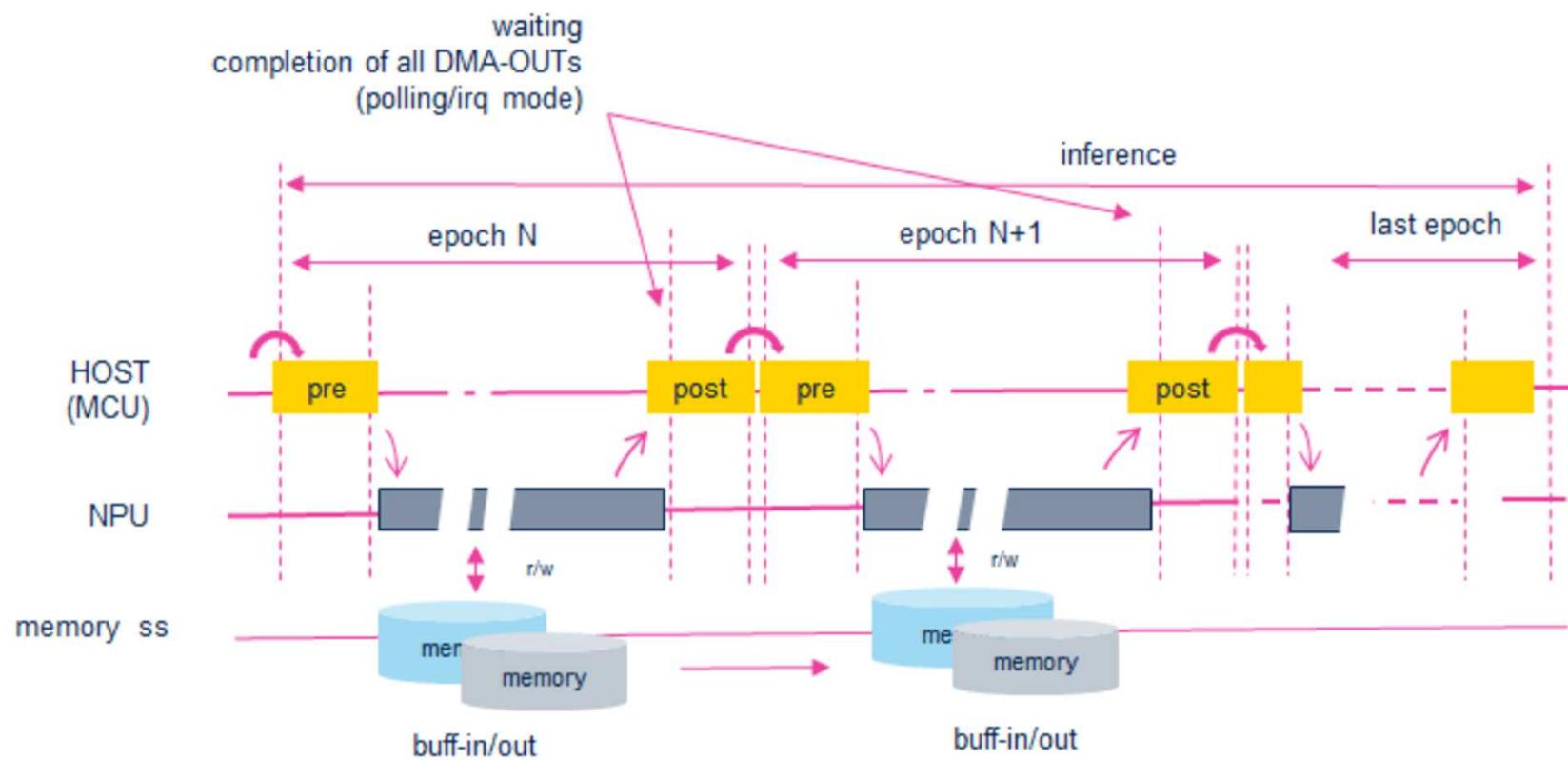
# Scheduling



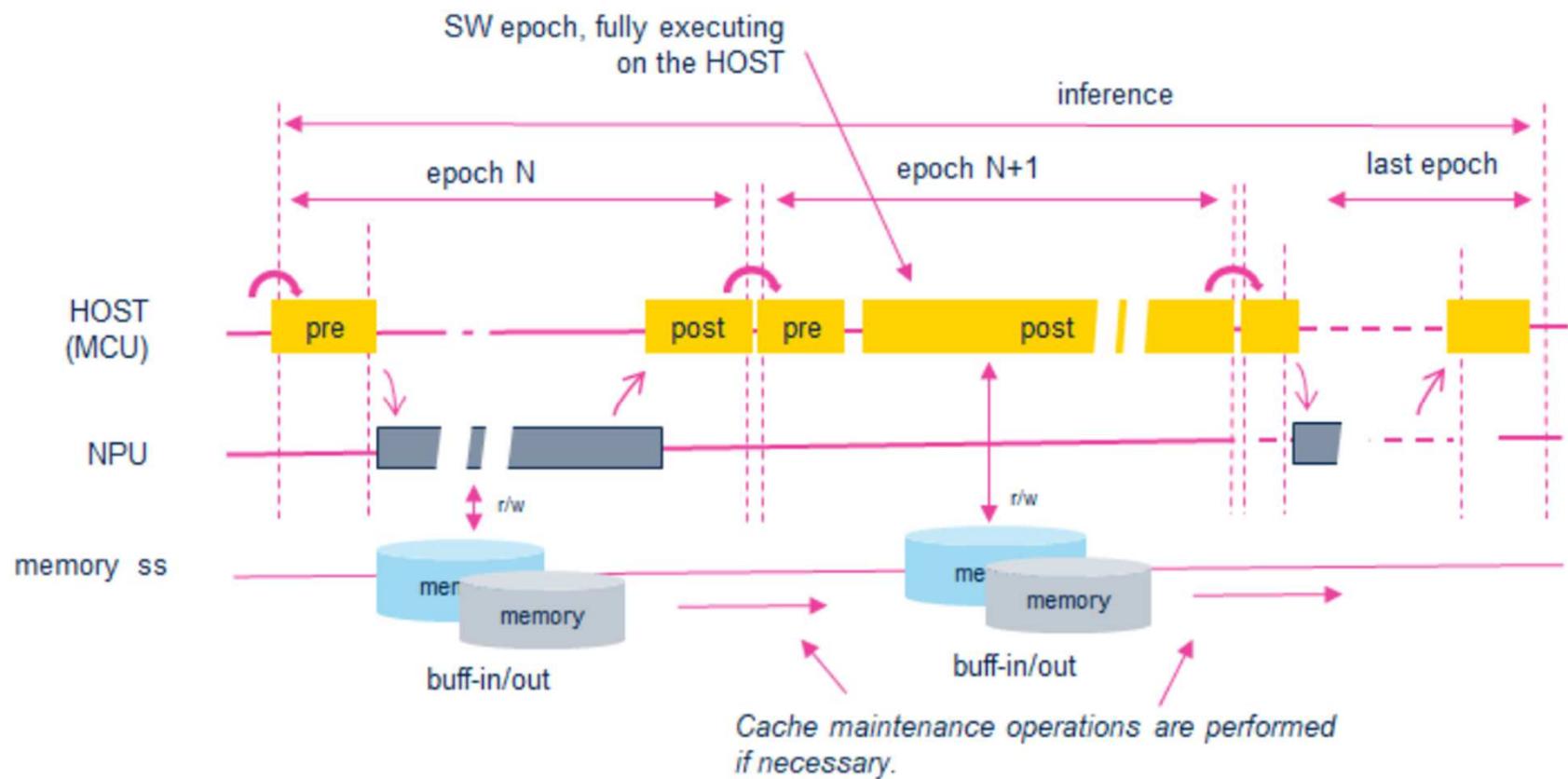
As part of the [NPU runtime software stack](#), a lightweight scheduler engine is responsible for executing the list of different epochs. Each epoch is considered an *atomic operation*, and its execution is ordered and fixed to ensure data dependency across the entire computational graph.



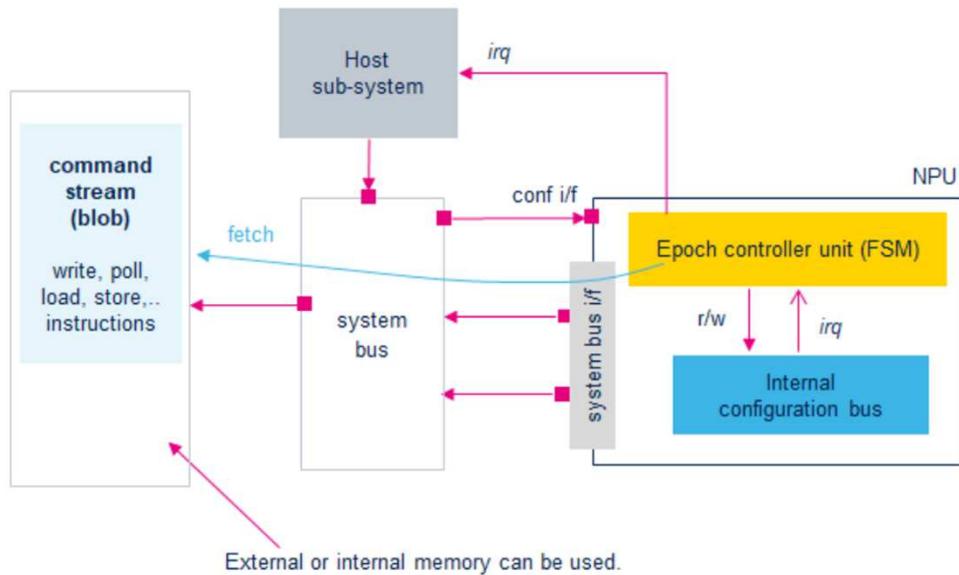
# HW epochs only



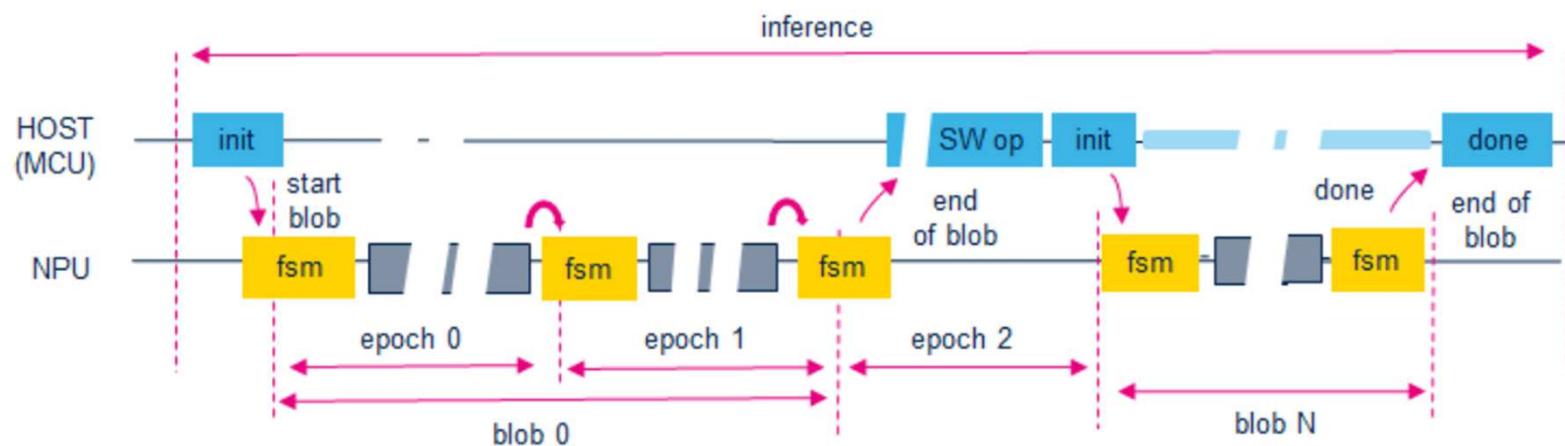
# Including SW epoch



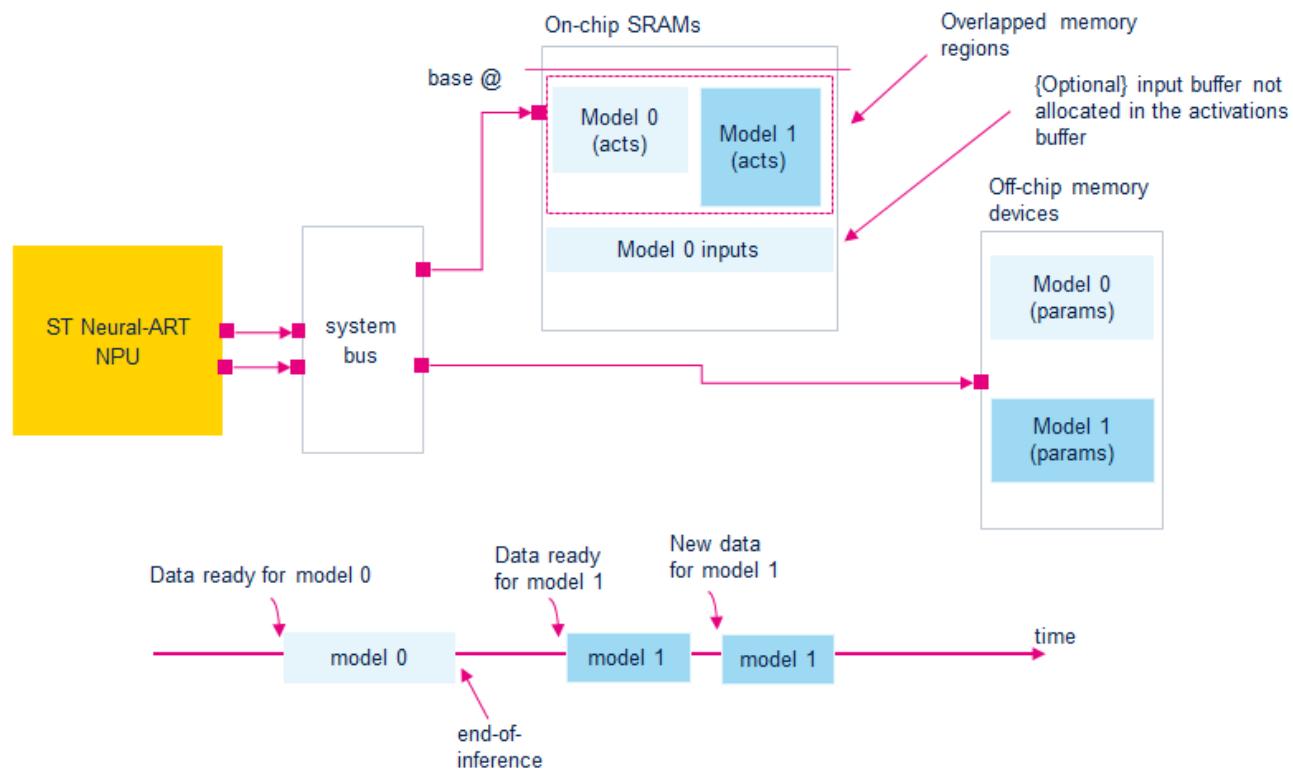
# Epoch controller mode



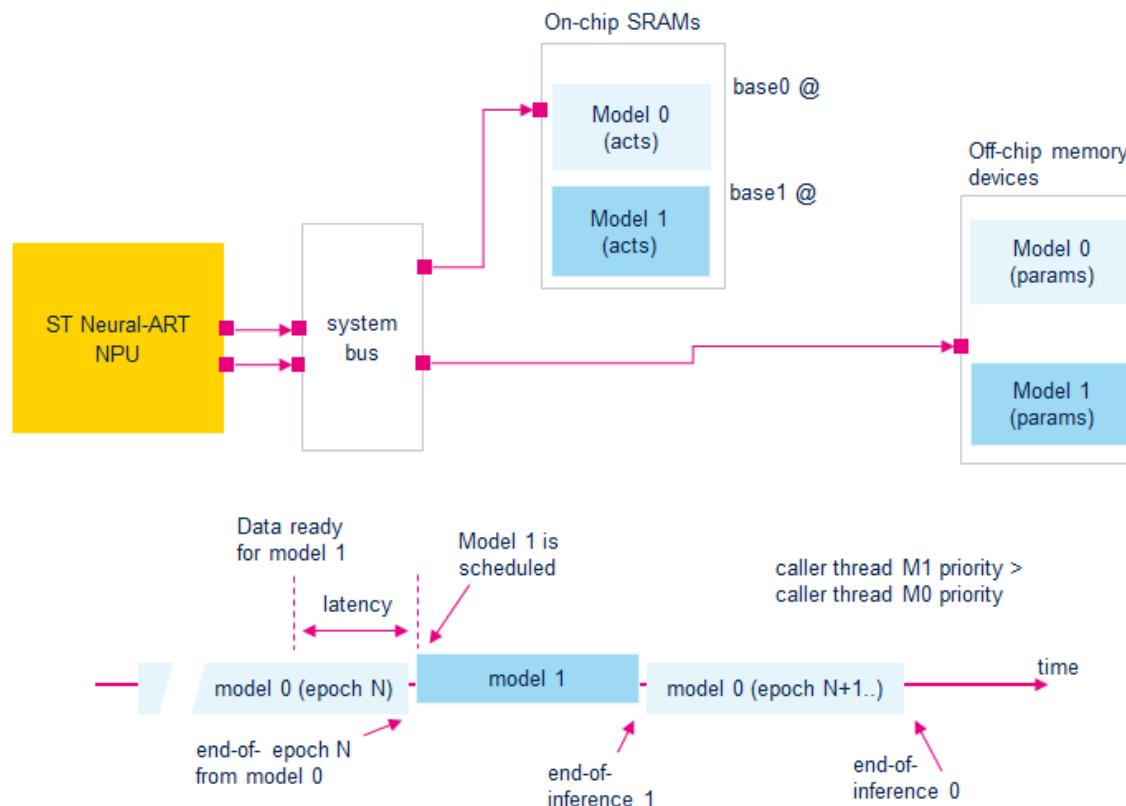
External or internal memory can be used.



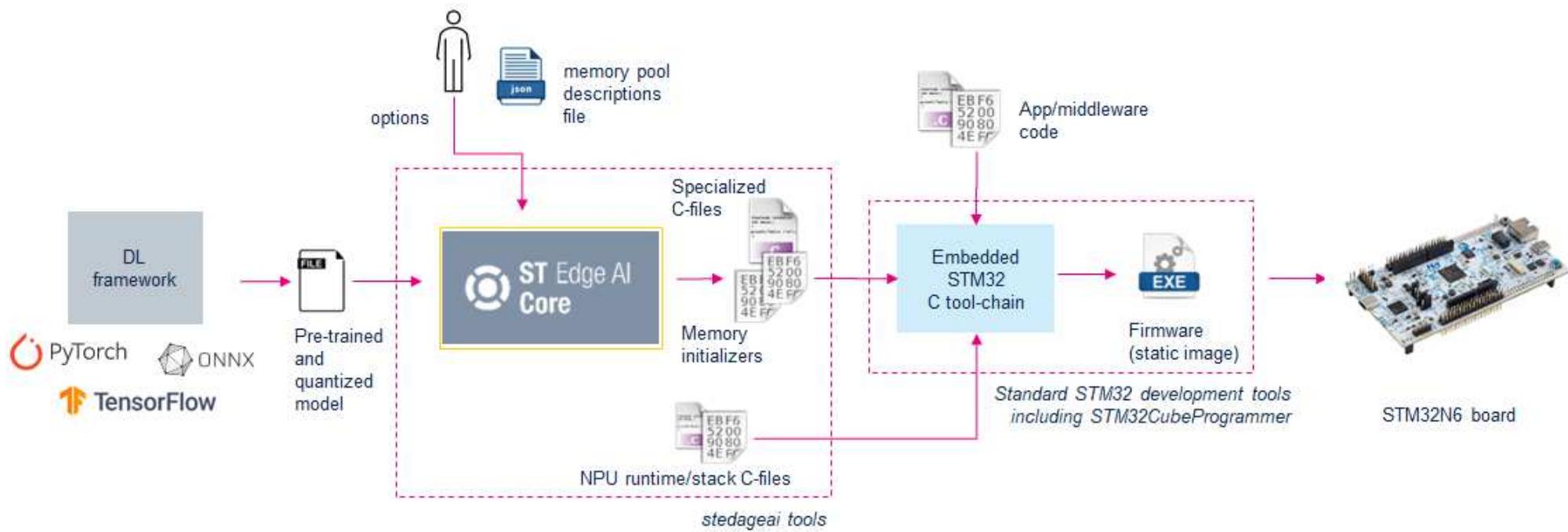
# Multiple models support – Serial mode



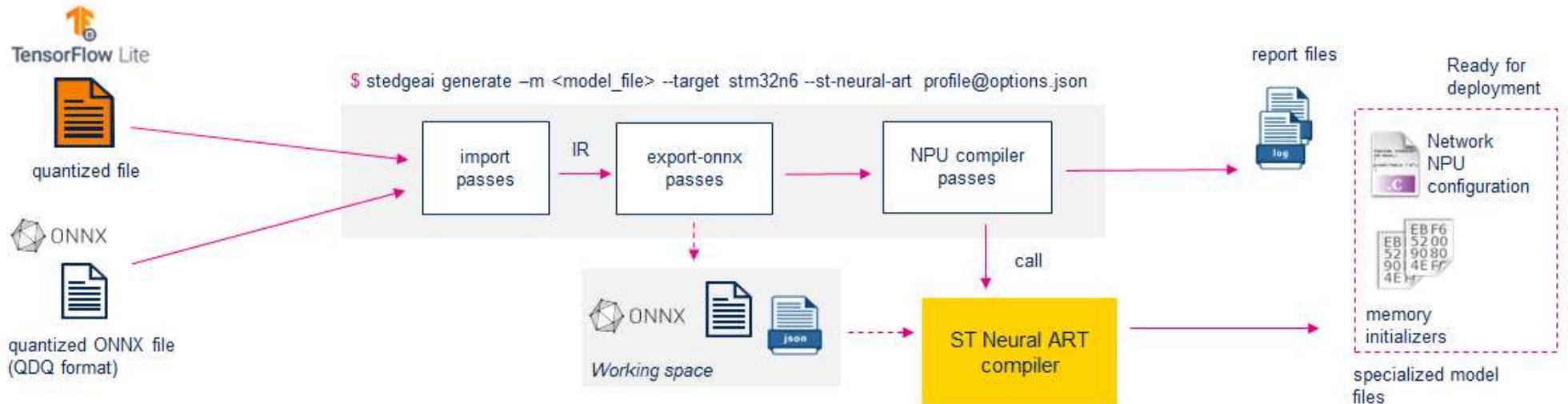
# Multiple models support - Epoch level mode



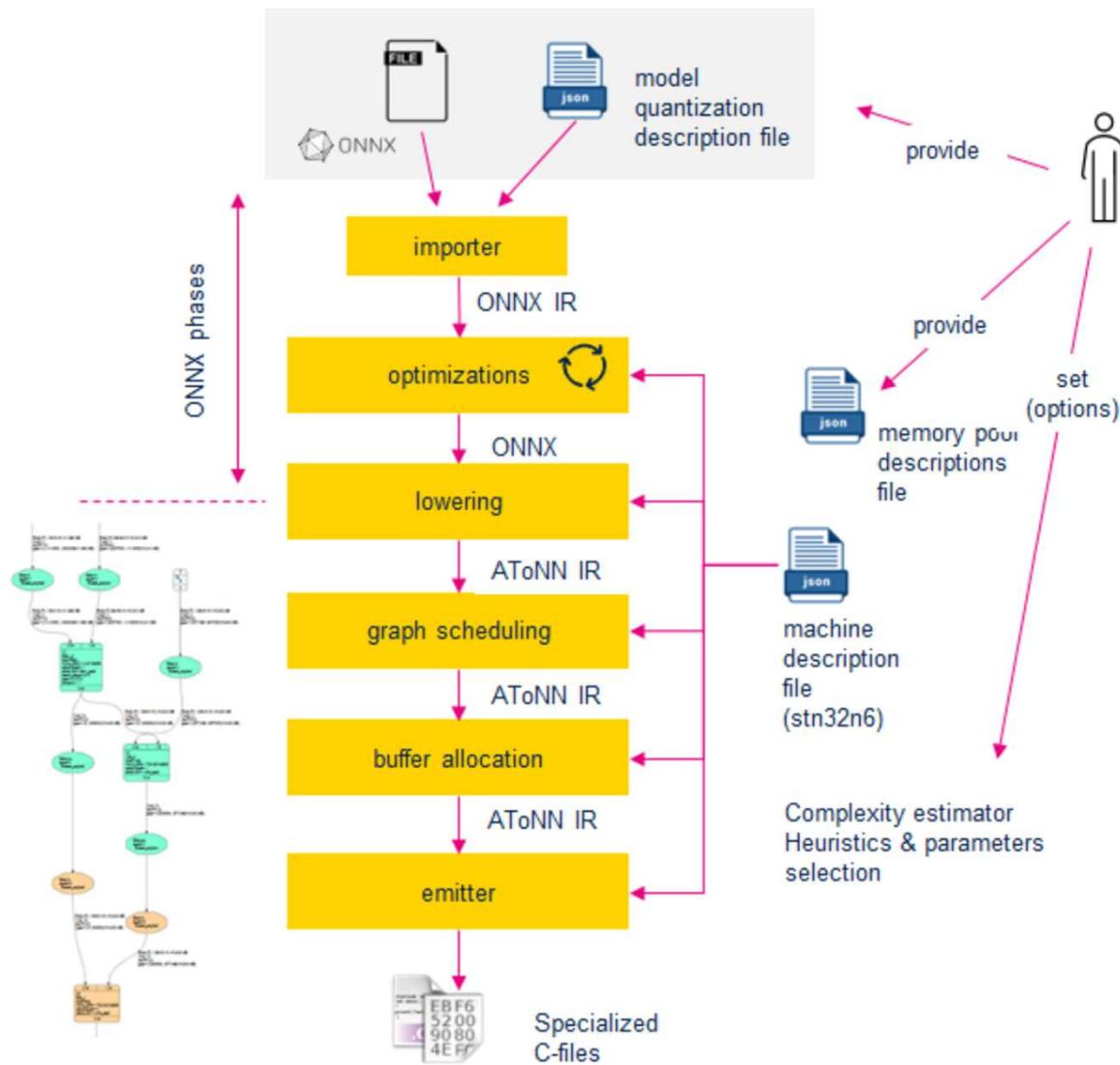
# Ahead-Of-Time (AOT) flow



# Intermediate files



# NPU Compiler



json

## \Repository\Packs\STMicroelectronics\X-CUBE-AI\10.0.0\scripts\N6\_scripts\user\_neuralart.json

```
"Profiles": {
    "n6-extram": {
        "memory_pool": "./my_mpools/stm32n6_extRam.mpool",
        "memory_desc": "./my_mdscs/stm32n6.mdesc",
        "options" : "--optimization 3 --all-buffers-info --mvei --no-hw-sw-parallelism --cache-maintenance --Oalt-sched --native-float --enable-virtual-mem-pools --Omax-ca-pipe 4 --Oshuffle-dma --Ocache-opt --Os"
    },
    "n6-extflash": {
        "memory_pool": "./my_mpools/stm32n6_extFlash.mpool",
        "memory_desc": "./my_mdscs/stm32n6.mdesc",
        "options" : "--optimization 3 --all-buffers-info --mvei --no-hw-sw-parallelism --cache-maintenance --Oalt-sched --native-float --enable-virtual-mem-pools --Omax-ca-pipe 4 --Oshuffle-dma --Ocache-opt --Os"
    },
    "n6-noextmem": {
        "memory_pool": "./my_mpools/stm32n6_noextmem.mpool",
        "memory_desc": "./my_mdscs/stm32n6.mdesc",
        "options" : "--optimization 3 --all-buffers-info --mvei --no-hw-sw-parallelism --cache-maintenance --Oalt-sched --native-float --enable-virtual-mem-pools --Omax-ca-pipe 4 --Oshuffle-dma --Ocache-opt --Os"
    },
    "n6-nointmem": {
        "memory_pool": "./my_mpools/stm32n6_nointmem.mpool",
        "memory_desc": "./my_mdscs/stm32n6.mdesc",
        "options" : "--optimization 3 --all-buffers-info --mvei --no-hw-sw-parallelism --cache-maintenance --Oalt-sched --native-float --enable-virtual-mem-pools --Omax-ca-pipe 4 --Oshuffle-dma --Ocache-opt --Os"
    },
    "n6-allmems-01": {
        "memory_pool": "./my_mpools/stm32n6.mpool",
        "memory_desc": "./my_mdscs/stm32n6.mdesc",
        "options" : "--optimization 1 --all-buffers-info --mvei --no-hw-sw-parallelism --cache-maintenance --Oalt-sched --native-float --enable-virtual-mem-pools --Omax-ca-pipe 4 --Oshuffle-dma --Ocache-opt --Os"
    },
    "n6-allmems-02": {
        "memory_pool": "./my_mpools/stm32n6.mpool",
        "memory_desc": "./my_mdscs/stm32n6.mdesc",
        "options" : "--optimization 2 --all-buffers-info --mvei --no-hw-sw-parallelism --cache-maintenance --Oalt-sched --native-float --enable-virtual-mem-pools --Omax-ca-pipe 4 --Oshuffle-dma --Ocache-opt --Os"
    },
    "n6-allmems-03": {
        "memory_pool": "./my_mpools/stm32n6.mpool",
        "memory_desc": "./my_mdscs/stm32n6.mdesc",
        "options" : "--optimization 3 --all-buffers-info --mvei --no-hw-sw-parallelism --cache-maintenance --Oalt-sched --native-float --enable-virtual-mem-pools --Omax-ca-pipe 4 --Oshuffle-dma --Ocache-opt --Os"
    },
    "n6-allmems-Oauto": {
        "memory_pool": "./my_mpools/stm32n6.mpool",
        "memory_desc": "./my_mdscs/stm32n6.mdesc",
        "options" : "--Oauto --all-buffers-info --mvei --no-hw-sw-parallelism --cache-maintenance --Oalt-sched --native-float --enable-virtual-mem-pools --Omax-ca-pipe 4 --Oshuffle-dma --Ocache-opt --Os"
    }
},
```



## \Repository\Packs\STMicroelectronics\X-CUBE-AI\10.0.0\scripts\N6\_scripts\my\_mdescs\stm32n6.mdesc

```
1  {
2    "externals": [
3      | "mpool_name" : "stm32n6"
4    },
5    "machine": {
6      "name": "stm32n6",
7      "version": "0.0",
8      "fully_connected": "true",
9      "nominal_frequency": 1000,
10     "nominal_vdd": 0.8,
11     "frequency": 1000,
12     "vdd": 0.8,
13     "switches": [
14       { "index": 0, "num": 1, "type": "STRSWITCH", "version": "1.6" }
15     ],
16     "accelerators": [
17       { "index": 0, "num": 5, "type": "STRENG", "version": "5.0", "power": 5.331, "inPorts": [ 0 ], "outPorts": [ 0 ], "params": [ { "na
18       { "index": 5, "num": 5, "type": "STRENG", "version": "5.0", "power": 5.331, "inPorts": [ 0 ], "outPorts": [ 0 ], "params": [ { "na
19       { "index": 0, "num": 4, "type": "CONVACC", "version": "5.1", "power": 30.132, "inPorts": [ 0, 0, 0 ], "outPorts": [ 0 ] },
20       { "index": 0, "num": 2, "type": "DECUN", "version": "1.10", "power": 5.57, "inPorts": [ 0, 0 ], "outPorts": [ 0 ] },
21       { "index": 0, "num": 2, "type": "ACTIV", "version": "1.1", "power": 5.30, "inPorts": [ 0 ], "outPorts": [ 0 ] },
22       { "index": 0, "num": 4, "type": "ARITH", "version": "1.0", "power": 5.877, "inPorts": [ 0, 0 ], "outPorts": [ 0 ] },
23       { "index": 0, "num": 2, "type": "POOL", "version": "1.1", "power": 9.09, "inPorts": [ 0 ], "outPorts": [ 0 ] },
24       { "index": 0, "num": 1, "type": "RECBUF", "version": "3.2", "power": 5.57, "inPorts": [ 0, 0, 0 ], "outPorts": [ 0, 0, 0 ] },
25       { "index": 0, "num": 1, "type": "EPOCHCTRL", "version": "1.1", "power": 0.0 },
26       { "index": 0, "num": 1, "type": "DEBUG_TRACE", "version": "1.1", "power": 0.0 },
27       { "index": 0, "num": 1, "type": "PROCESSOR", "version": "0.0", "power": 0.0 }
28     ]
29   }
30 }
31 }
```

## \Repository\Packs\STMicroelectronics\X-CUBE-AI\10.0.0\Utilities\windows\targets\stm32\resources\mpools\stm32n6.mpool

```
"fname": "AXISRAM3",
"name": "npuRAM3",
"fformat": "FORMAT_RAW",
"prop": { "rights": "ACC_WRITE", "throughput": "HIGH", "latency": "LOW", "byteWidth": 8, "freqRatio": 1.25, "read_power": 18.531, "write_power": 16.201 },
"offset": { "value": "0x34200000", "magnitude": "BYTES" },
"size": { "value": "448", "magnitude": "KBYTES" }
},
{
"fname": "AXISRAM4",
"name": "npuRAM4",
"fformat": "FORMAT_RAW",
"prop": { "rights": "ACC_WRITE", "throughput": "HIGH", "latency": "LOW", "byteWidth": 8, "freqRatio": 1.25, "read_power": 18.531, "write_power": 16.201 },
"offset": { "value": "0x34270000", "magnitude": "BYTES" },
"size": { "value": "448", "magnitude": "KBYTES" }
},
{
"fname": "AXISRAM5",
"name": "npuRAM5",
"fformat": "FORMAT_RAW",
"prop": { "rights": "ACC_WRITE", "throughput": "HIGH", "latency": "LOW", "byteWidth": 8, "freqRatio": 1.25, "read_power": 18.531, "write_power": 16.201 },
"offset": { "value": "0x342e0000", "magnitude": "BYTES" },
"size": { "value": "448", "magnitude": "KBYTES" }
},
{
"fname": "AXISRAM6",
"name": "npuRAM6",
"fformat": "FORMAT_RAW",
"prop": { "rights": "ACC_WRITE", "throughput": "HIGH", "latency": "LOW", "byteWidth": 8, "freqRatio": 1.25, "read_power": 19.006, "write_power": 15.790 },
"offset": { "value": "0x34350000", "magnitude": "BYTES" },
"size": { "value": "448", "magnitude": "KBYTES" }
},
{
"fname": "xSPI1",
"name": "hyperRAM",
"fformat": "FORMAT_RAW",
"prop": { "rights": "ACC_WRITE", "throughput": "MID", "latency": "HIGH", "byteWidth": 2, "freqRatio": 5.00, "cacheable": "CACHEABLE_ON", "read_power": 380, "write_power": 340.0, "constant": "0x00000000" },
"offset": { "value": "0x90000000", "magnitude": "BYTES" },
"size": { "value": "32", "magnitude": "MBYTES" }
},
{
"fname": "xSPI2",
"name": "octoFlash",
"fformat": "FORMAT_RAW",
"prop": { "rights": "ACC_READ", "throughput": "MID", "latency": "HIGH", "byteWidth": 1, "freqRatio": 6.00, "cacheable": "CACHEABLE_ON", "read_power": 110, "write_power": 400.0, "constant": "0x00000000" },
"offset": { "value": "0x70000000", "magnitude": "BYTES" },
"size": { "value": "64", "magnitude": "MBYTES" }
}
}
```

# Documentation

이름	수정한 날짜	유형	크기
quantization.html	2025-02-03 오후 5:45	Chrome HTML D...	2,138KB
release_note.html	2025-02-03 오후 5:45	Chrome HTML D...	1,786KB
setting_env.html	2025-02-03 오후 5:45	Chrome HTML D...	1,734KB
stellar_command_line_interface.html	2025-02-03 오후 5:45	Chrome HTML D...	1,961KB
stellar_how_to_upgrade_a_project.html	2025-02-03 오후 5:45	Chrome HTML D...	1,772KB
stellar_on_target_validation.html	2025-02-03 오후 5:45	Chrome HTML D...	1,716KB
stm32_command_line_interface.html	2025-02-03 오후 5:45	Chrome HTML D...	1,825KB
stm32_how_to_run_a_model_locally.html	2025-02-03 오후 5:45	Chrome HTML D...	1,751KB
stm32_how_to_upgrade_a_project.html	2025-02-03 오후 5:45	Chrome HTML D...	1,755KB
stm32_how_to_use_usb_cdc_for_validation.html	2025-02-03 오후 5:45	Chrome HTML D...	2,279KB
stm32_on_target_validation.html	2025-02-03 오후 5:45	Chrome HTML D...	1,722KB
stm32_relocatable_mode.html	2025-02-03 오후 5:45	Chrome HTML D...	2,005KB
stm32_tflite_micro_support.html	2025-02-03 오후 5:45	Chrome HTML D...	2,013KB
stm32mpu_command_line_interface.html	2025-02-03 오후 5:45	Chrome HTML D...	2,301KB
stneuralart_api_and_stack.html	2025-02-03 오후 5:45	Chrome HTML D...	1,810KB
stneuralart_aton_operator_support.html	2025-02-03 오후 5:45	Chrome HTML D...	1,747KB
stneuralart_faqs.html	2025-02-03 오후 5:45	Chrome HTML D...	1,732KB
stneuralart_getting_started.html	2025-02-03 오후 5:45	Chrome HTML D...	2,802KB
stneuralart_model_evaluation.html	2025-02-03 오후 5:45	Chrome HTML D...	2,692KB
stneuralart_neural_art_compiler.html	2025-02-03 오후 5:45	Chrome HTML D...	1,875KB
stneuralart_operator_support.html	2025-02-03 오후 5:45	Chrome HTML D...	1,750KB
stneuralart_programming_model.html	2025-02-03 오후 5:45	Chrome HTML D...	2,475KB
stneuralart_stm32n6_projects.html	2025-02-03 오후 5:45	Chrome HTML D...	1,764KB
supported_ops_keras.html	2025-02-03 오후 5:45	Chrome HTML D...	1,817KB

STM32Cube\Repository\Packs\STMicroelectronics\X-CUBE-AI\xx.x.x\Documentation

You can refer to documents related to **ST Neural ART** and **ST Edge AI**.



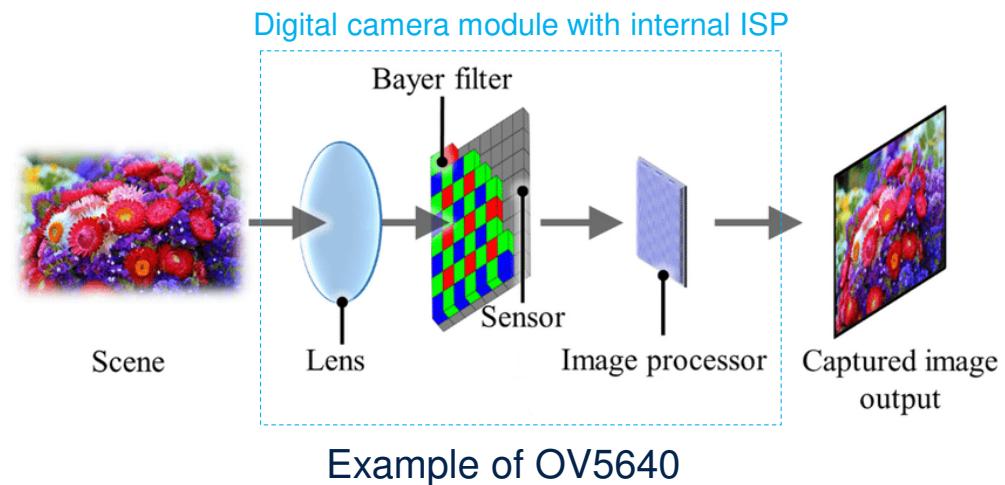
# **DCMIPP ISP and VENC**



# Camera modules ...



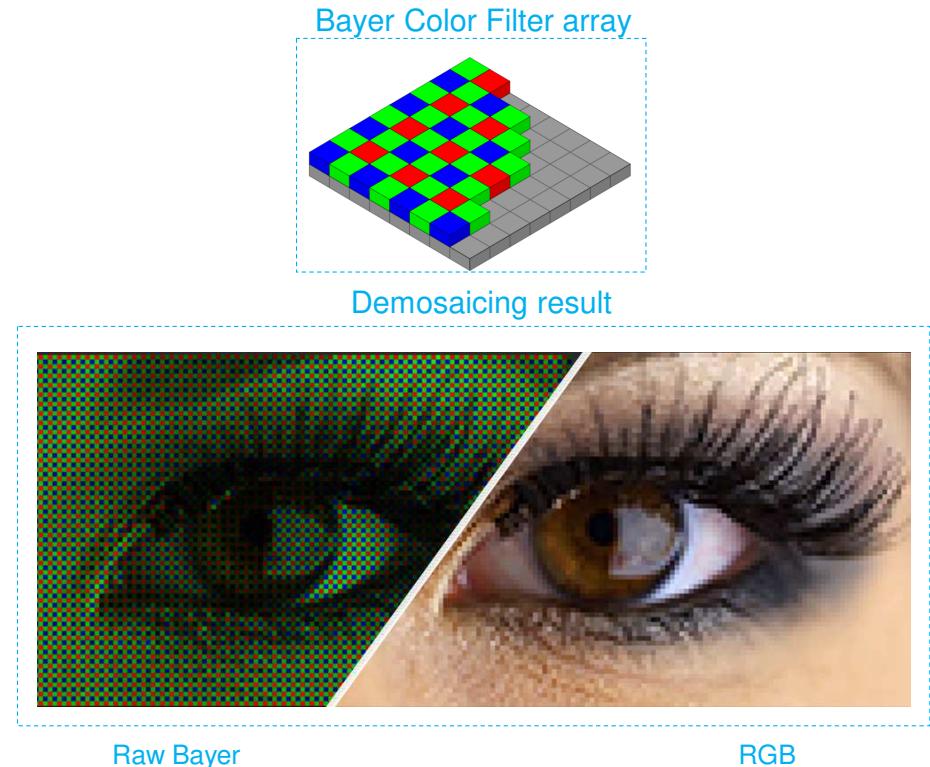
- The camera module embeds
  - The lens
  - The Bayer matrix and filter
  - The Image Signal processor
- Common output format of digital cameras:
  - YUV
  - RGB
  - RAW Bayer



## ... Vs Sensor module



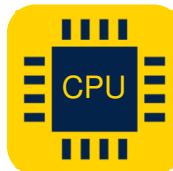
- The sensor module embeds:
  - The lens
  - The Bayer matrix and filter
- The output format of sensor modules:
  - RAW Bayer (8, 10, 12, 14 bits)



# Key benefits to integrate an ISP



**More versatility** to change image sensor  
(ISP to tune once)



**Totally offload CPU** from multi-channel application scenarios  
(H264+NPU/NPU+CPU/NPU+DISPLAY)



**Cost saving** on image sensor



Support **MIPI CSI2 image sensors** (higher bandwidth)

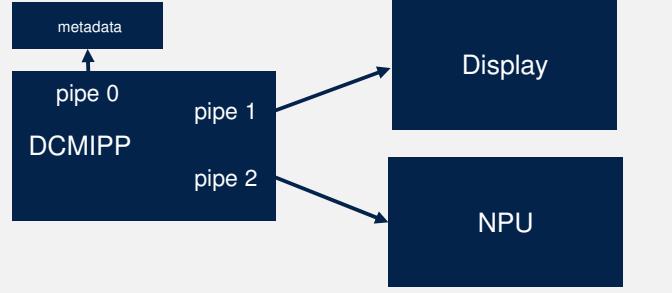
# Computer vision possible pipeline

## Offload CPU for various use cases

### AI + DISPLAY



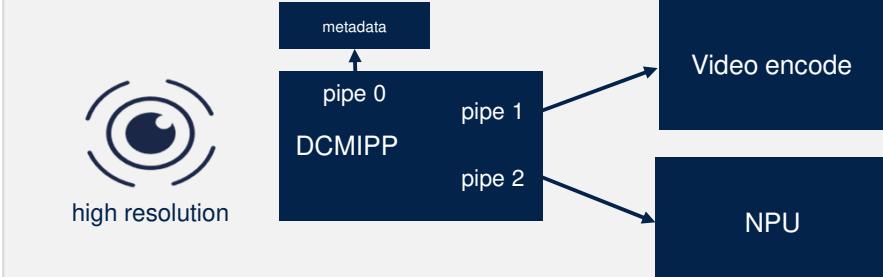
high resolution



### AI + H264 encode



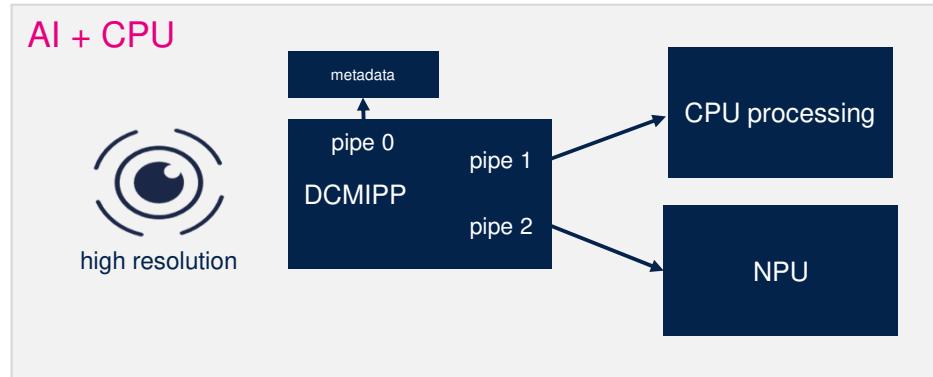
high resolution



### AI + CPU



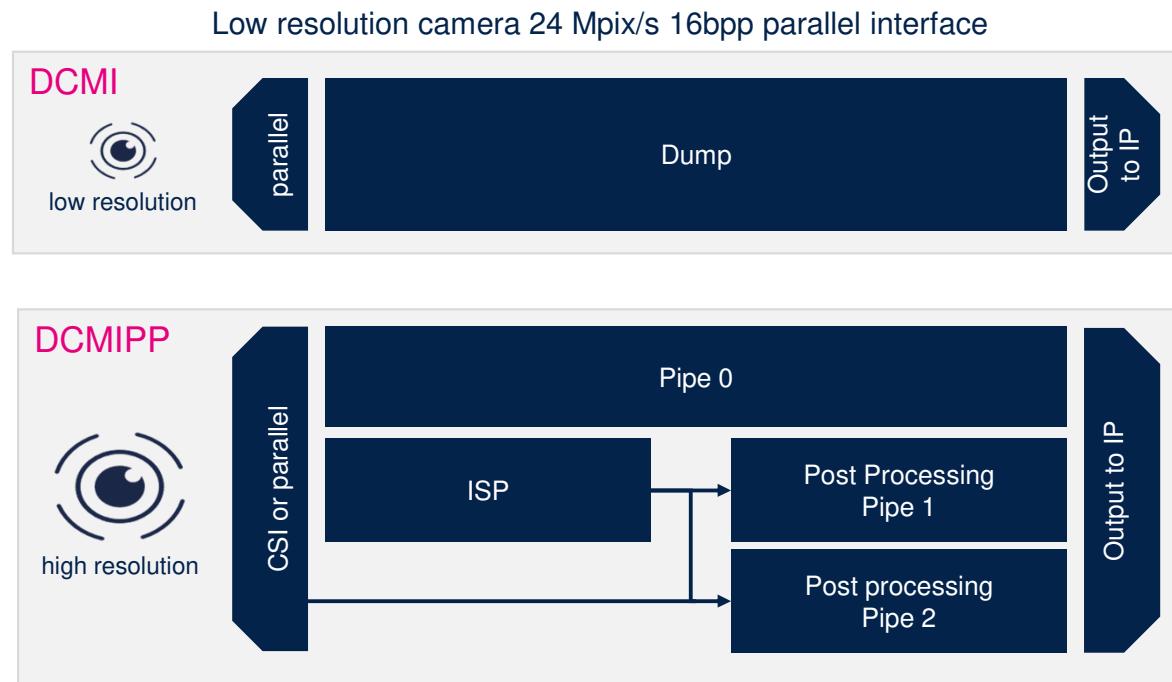
high resolution



# Camera subsystem

## Efficient image acquisition and processing to multiple output

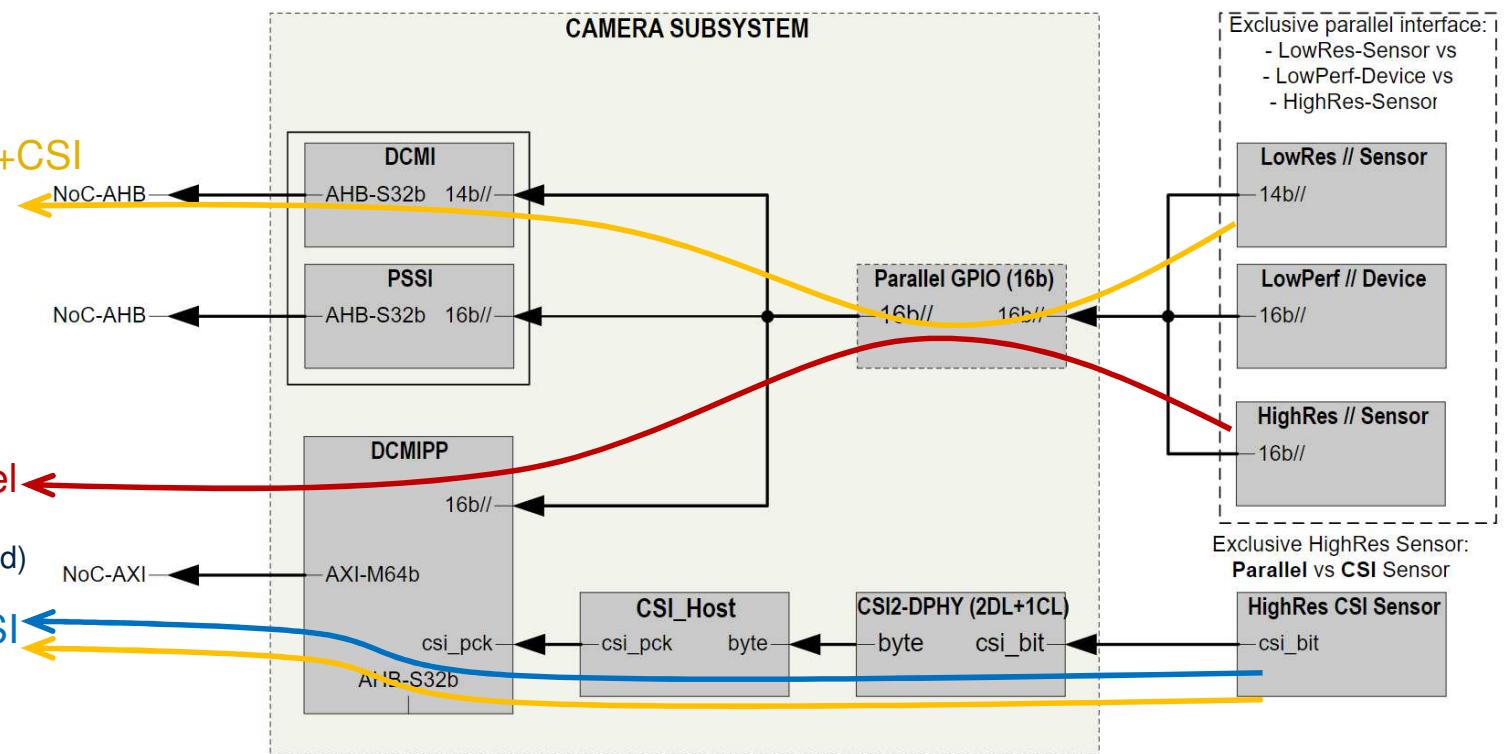
- MiPi CSI and parallel inputs
- Offload MCU of all image acquisition via 3 camera pipelines
- Lite-ISP features
  - Bad pixel, Decimation, Black level, Exposure, Demosaic, Column conversion, Contrast
- Post processing features
  - crop, downsize, ROI, Gamma, YUV conv, Pixel packer



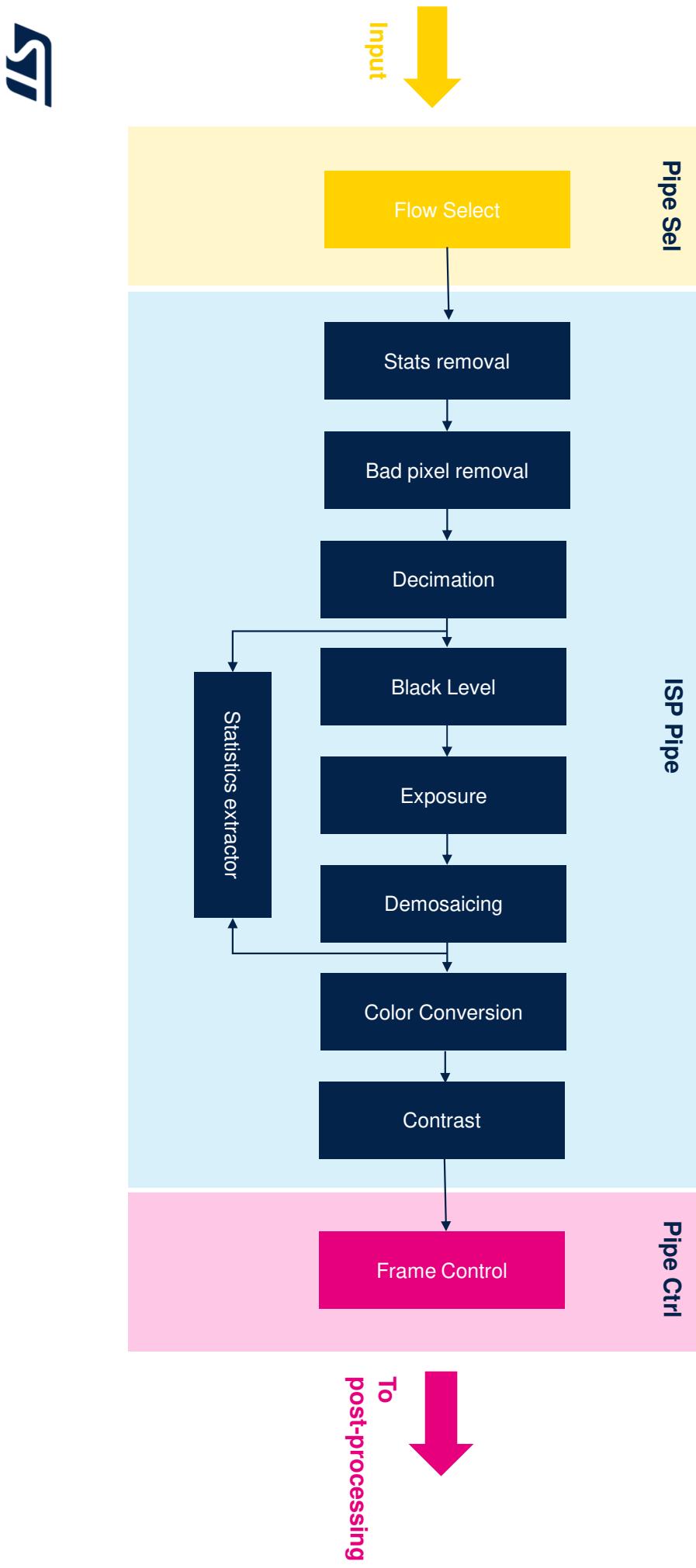
# Camera SubSystem

## UC3: Double Camera, Parallel+CSI

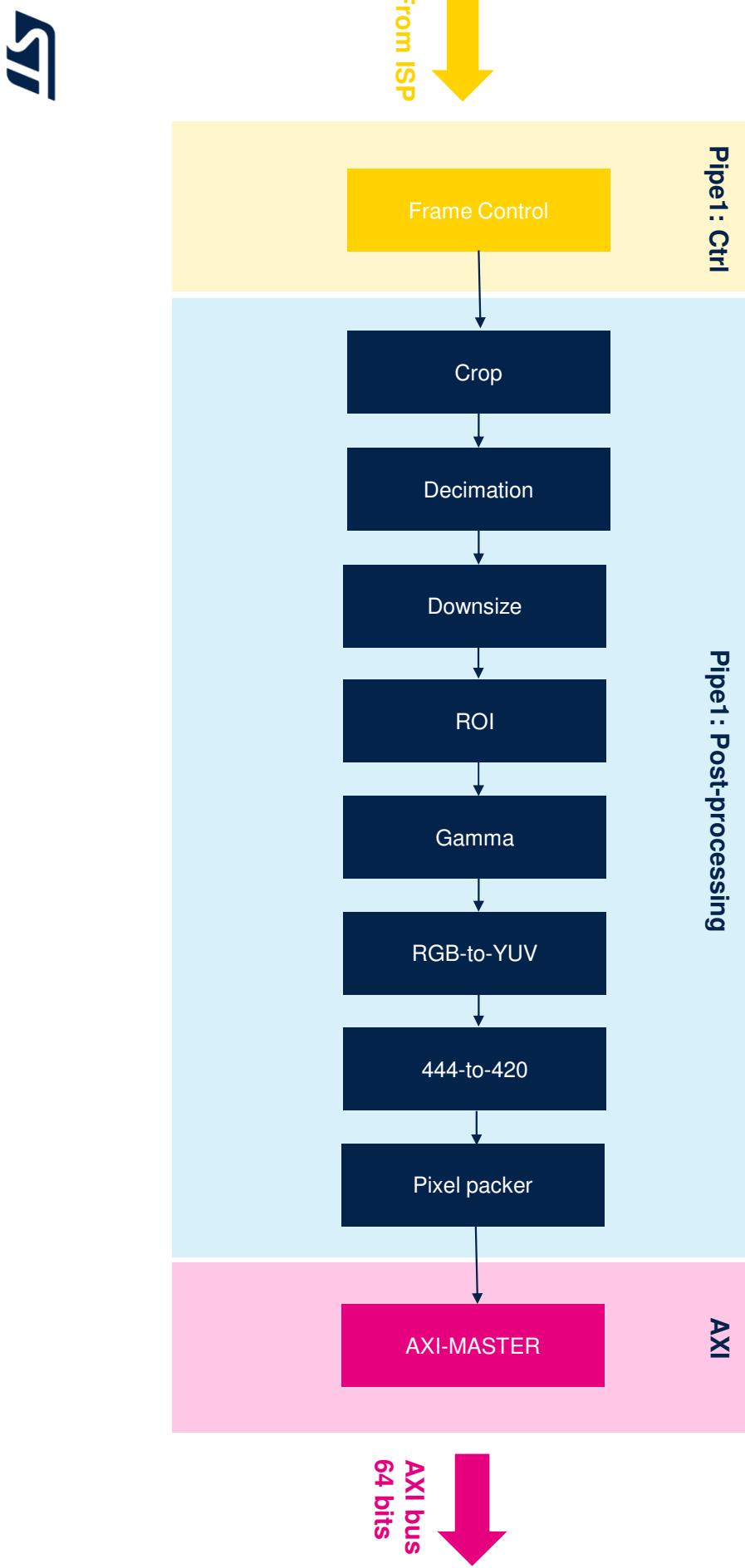
- Parallel on DCMI
- CSI on DCMIPP



# ISP details



# Post processing details



# STM32 ISP IQTune desktop application



- To take control of the DCMIPP ISP
  - The goal of the ISP IQ tuning tool is to ease the tuning of the user selected sensor by providing an easy access to ISP registers and parameters
- The STM32 ISP IQTune desktop application supports:
  - Windows OS, Ubuntu OS and MacOS



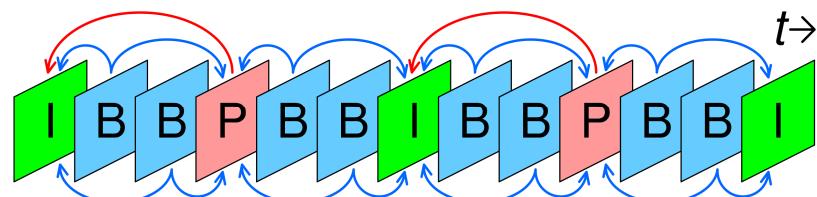
# Why encode video

- Raw video takes a lot of space :
  - CIF (352x288 pixels) : 148 kB per frame
  - 720p HD : 1.3 MB per frame
  - 1080p Full HD : 3MB per frame
  - 2160p UHD : 12MB per frame
- Average internet connection in France : 122Mb/s (~12MB/s including protocol overhead)
- Video needs to be compressed to be stored and streamed



# Vocabulary : frame types

- MJPEG: each frame is independently encoded in JPEG format

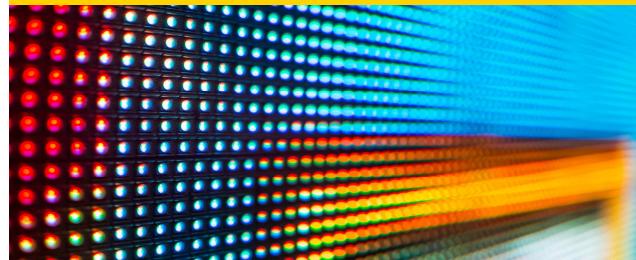


Example of dependencies between encoded frames

- Intra-Frame (I) : Coded using the current frame only
- Inter-Frame (P or B) : Coded using other reference frames
  - Predicted (P) frames are coded using information from previous I-frame
  - Bidirectional (B) frames are coded using data from any previous or following frame

# STM32N6 – Video Encoder – path SD Use Case

## HW VENC IP



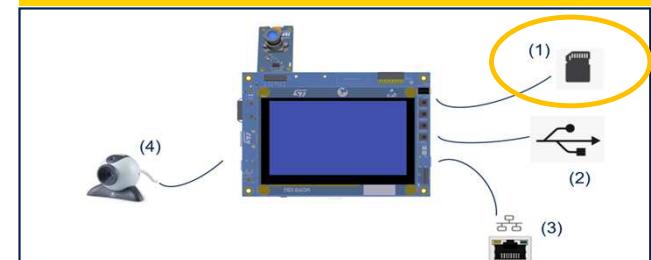
- IP provided by **Verisilicon** 
- Targeted **Performance**
  - 1080p30 (1920 x 1080 pixels @ 30 fps)
  - 720p60 (1280 x 720 pixels @ 60 fps)

## FW SYSTEM INTEGRATION



- Stack for full h264 control - **Verisilicon**
- EWL API (**Verisilicon**) for memory mgt and hardware access → wrapper **ST**
- LL for init and register access → **ST**

## APPLICATIONS



- 4 video flow use cases defined by Mktg
  1. **SD card**
  2. **USB**
  3. **Ethernet**
  4. **USB WebCam (not VENC)**

1. Using **Discovery** board: 800 x 480 @ 30 fps - record 600 frames, encode and store in SD

Camera

CSI2  
Interface

DCMIPP

VENC

DMA

SD card

2. Using **PC**: read SD, add mp4 header (no file system ready) and play file in VLC

Expected Compression : from 500MB raw to ~12MB encoded



# RTSP Server

1. Using **Discovery** board: 960x720p @ 30 fps encoded using h.264 at ~1Mbps



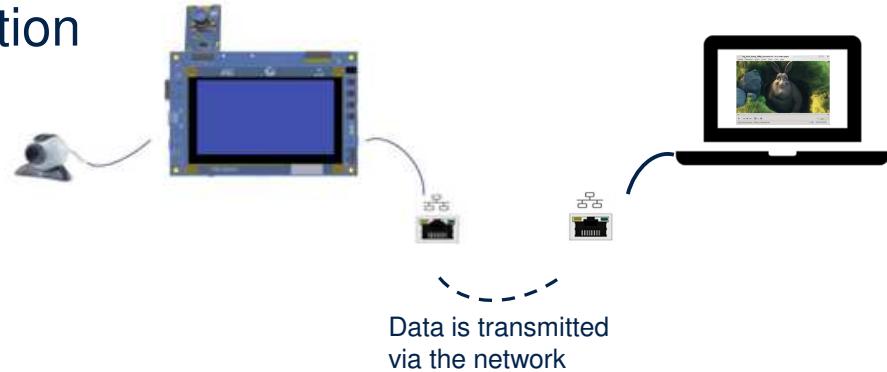
2. Using **PC**: Connect to the RTSP server and watch using VLC

Streaming mode between DCMIPP and VENC :

- Less CPU load related to VENC operation
- No full input buffer required

STM32N6 DK captures,  
encodes and acts as an  
RTSP server

The client receives and  
decodes the stream



# Hands-On AI: Recognizing Handwritten Digits with MNIST



# Running CubeMX

The screenshot shows the STM32CubeMX software interface. At the top, there's a navigation bar with tabs: 'MCU/MPU Selector' (highlighted in blue), 'Board Selector', 'Example Selector', and 'Cross Selector'. Below the navigation bar is a search bar and a 'MCU/MPU Filters' section. The 'Commercial Part Number' dropdown is set to 'STM32N657X0H3Q' and is circled with a red number 2. A red arrow points from the 'ACCESS TO MCU SELECTOR' button in the project setup guide to this dropdown. The 'MCU/MPU Filters' section includes dropdowns for Segment, Series, Line, Marketing Status, Price, Package, Core, and Coprocessor. Below these is a 'MEMORY' section with a slider for Flash memory at 0 kBytes. On the right side of the interface, there's a large yellow advertisement for the STM32N657 MCU, featuring the ST logo and the text 'The first high-perform STM32 MCU with AI c'. At the bottom, a table titled 'MCUs/MPUs List: 1 item' shows one entry: 'STM32N657X0H...', 'Part No: STM32N657X0', 'Reference: STM32N657X0...', 'Marketing St...: Active', and 'Unit Price for: 10.7254'. A red circle with a red number 3 is around the table header, and another red arrow points from the 'Double Click' text in the project setup guide to the table.

New Project

I need to :

Start My project from MCU  
① ACCESS TO MCU SELECTOR

Start My project from ST Board  
ACCESS TO BOARD SELECTOR

Start My project from Example  
ACCESS TO EXAMPLE SELECTOR

STM32N657X0H3Q

MCU/MPU Selector Board Selector Example Selector Cross Selector

MCU/MPU Filters

Commercial Part Number: STM32N657X0H3Q

PRODUCT INFO

Segment Series Line Marketing Status Price Package Core Coprocessor

MEMORY

Flash = 0 (kBytes)

MCUs/MPUs List: 1 item

Commercial Part No...	Part No	Reference	Marketing St...	Unit Price for
STM32N657X0H...	STM32N657X0	STM32N657X0...	Active	10.7254

Features Block Diagram Docs & Resources CAD

The first high-perform STM32 MCU with AI c

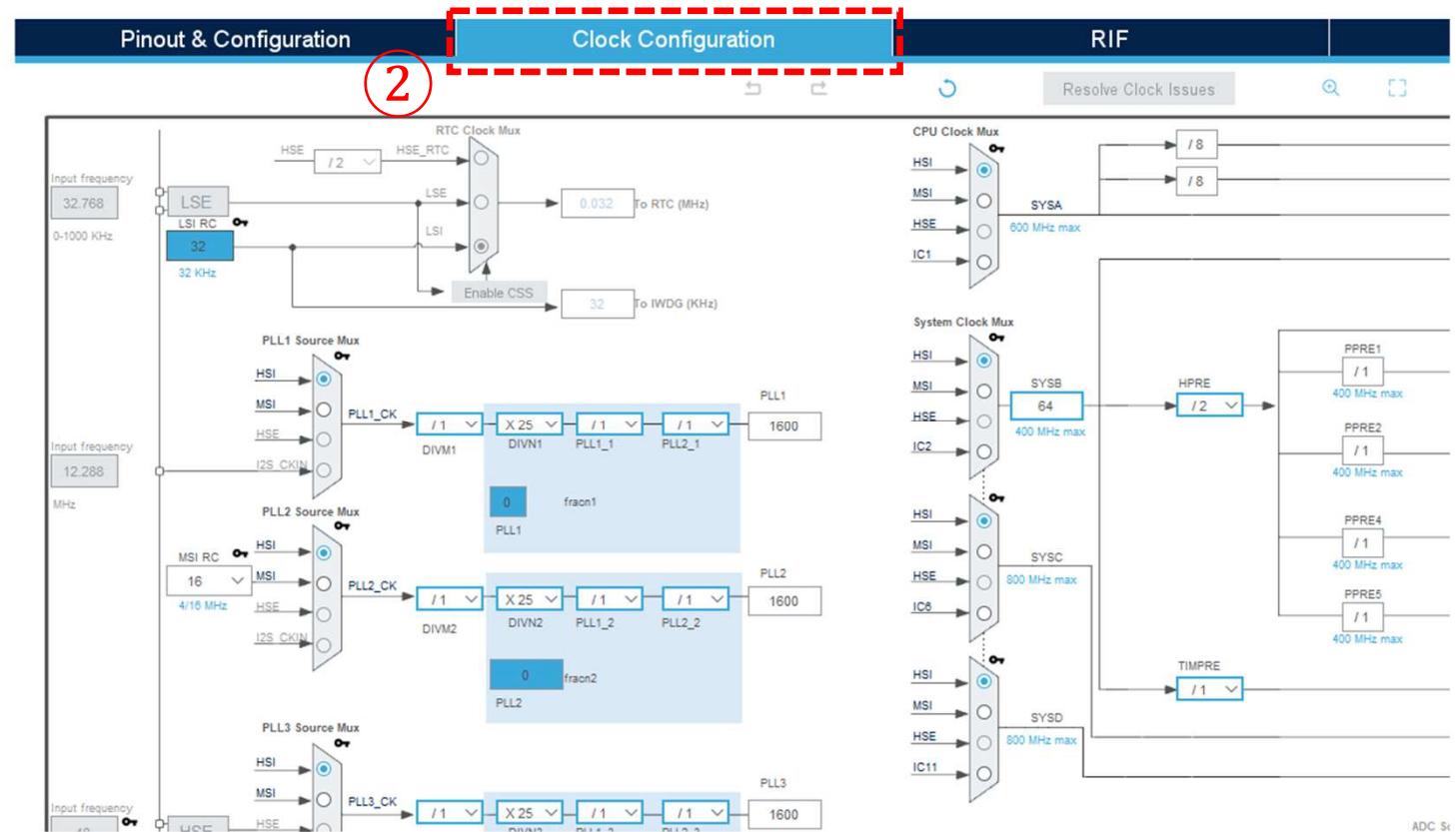
Double Click



# Clock Configuration

TrustZone feature available X

- ① Do you want to create a new project :
- Secure and Non Secure domains
  - Secure domain only
- OK



"In this example, the clock configuration is set directly in the code, so you can leave it as default in CubeMX."



# CORTEX\_M55\_S Configuration

"In this example, FSBL is not used; only the Secure Application is used."

The screenshot shows two tabs of a configuration tool: "Pinout & Configuration" and "Clock Configuration".

**Pinout & Configuration:** This tab displays a table of system components and their pinout configurations. A red circle labeled "1" highlights the row for "CORTEX\_M55\_S". The table has columns: FSBL, Application, and ExtMemLoader. The "Application" column for "CORTEX\_M55\_S" contains a checked checkbox.

	FSBL	Application	ExtMemLoader
CORTEX_M55_FSB	<input checked="" type="checkbox"/>		
<b>CORTEX_M55_S</b>		<input checked="" type="checkbox"/>	
GPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
GPIO			
HPDMA1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ICACHE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IWDG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NVIC1_S_Application		<input checked="" type="checkbox"/>	
NVIC_FSB	<input checked="" type="checkbox"/>		
RAMCFG	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RCC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SYS_S	<input type="checkbox"/>	<input type="checkbox"/>	
WWDG	<input type="checkbox"/>	<input type="checkbox"/>	

**Clock Configuration:** This tab displays a table of software packs and their configurations. A red circle labeled "2" highlights the "Cortex Interface Settings" row under "Parameter Settings". The "CPU ICACHE" and "CPU DCACHE" columns both have "Enabled" checked.

Software Packs	Pinout
CORTEX_M55_S Mode and Configuration	
Mode	
Runtime contexts:	
First Stage Boot Loader	<input checked="" type="checkbox"/>
Application	<input checked="" type="checkbox"/>
External Memory Loader	
Configuration	
Reset Configuration	
Parameter Settings	User Constants
Configure the below parameters :	
Search (Ctrl+F)	
Cortex Interface Settings	
CPU ICACHE	<input checked="" type="checkbox"/> Enabled
CPU DCACHE	<input checked="" type="checkbox"/> Enabled
Cortex Memory Protection Unit Control ...	
MPU Control Mode	MPU NOT USED

"CPU ICACHE and CPU DCACHE are enabled."



# RAMCFG Configuration

The screenshot shows the STM32CubeMX software interface with two main tabs: "Pinout & Configuration" and "Clock Configuration".

**Pinout & Configuration Tab:**

- Categories:** A-Z
- System Core:** A table showing various peripherals and their configuration status. The row for "RAMCFG" is highlighted with a red dashed box.
- Other Sections:** Analog, Timers, Connectivity, Multimedia, Security.

**Clock Configuration Tab:**

- Software Packs:** RAMCFG Mode and Configuration
- Mode:** A list of RAMCFG configurations:
  - RAMCFG AXISRAM1
  - RAMCFG AXISRAM2
  - RAMCFG AXISRAM3** (Selected)
  - RAMCFG AXISRAM4
  - RAMCFG AXISRAM5
  - RAMCFG AXISRAM6
  - RAMCFG AHBSRAM1
  - RAMCFG AHBSRAM2
- Configuration:** A section for configuring parameters. It includes a "Reset Configuration" button and tabs for "Parameter Settings" and "User Constants".
  - Configure the below parameters :**
  - Search (Ctrl+F)**
  - AXISRAM3:** Software Erase: Disable
  - AXISRAM4:** Software Erase: Disable
  - AXISRAM5:** Software Erase: Disable
  - AXISRAM6:** Software Erase: Disable



# XSPIM Configuration

1

	Port 1	Port 2	Port 3	Port 4
USB1_OTG_HS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
USB2_OTG_HS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XSPI1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XSPI2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
XSPI3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

"In this example, the XSPIM configuration is set directly in the code, so you can leave it as default in CubeMX."

2

XSPIM Mode and Configuration

Mode

Runtime contexts:

First Stage Boot Loader	Application	External Memory Loader
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Mode Direct (XSPI1 to Port1; XSPI2 to Port2; XSPI3 not used)



# XSPI2 Configuration

①

②

Configuration

Reset Configuration

NVIC Settings DMA Settings GPIO Settings

Parameter Settings User Constants

Configure the below parameters :

Search (Ctrl+F)

Generic

Fifo Threshold	1
Memory Mode	Disable
Memory Type	Micron
Memory Size	16 Bits
Chip Select High Time Cycle	1
Free Running Clock	Disable

"In this example, the XSPI2 configuration is set directly in the code, so you can leave it as default in CubeMX."



# BSEC and CACHEAXI Configuration

1

Security			
	FSBL	Application	ExtMemLoader
<input checked="" type="checkbox"/> BSEC	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CRYP	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HASH	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MCAE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2

BSEC Mode and Configuration		
Mode		
Runtime contexts:	First Stage Boot Loader	Application
	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Activated		

3

System Core			
	FSBL	Application	ExtMemLo...
<input checked="" type="checkbox"/> CACHEAXI	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CORTEX_M55_FSBL	<input checked="" type="checkbox"/>		
CORTEX_M55_S		<input checked="" type="checkbox"/>	

4

CACHEAXI Mode and Configuration		
Mode		
Runtime contexts:	First Stage Boot Loader	Application
	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Activated		



# GPIOB Configuration

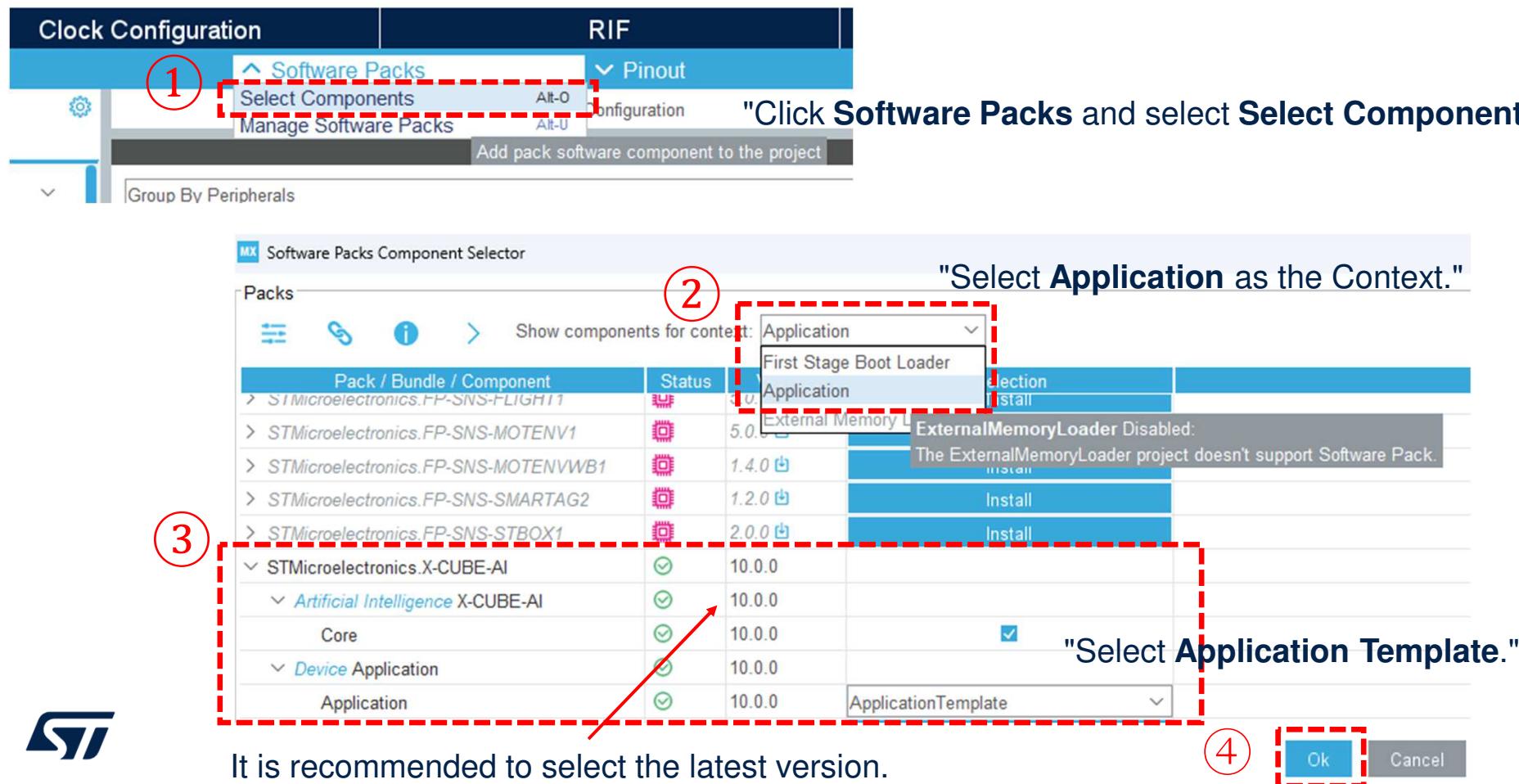
The screenshot illustrates the process of configuring the PB12 pin as an output in STM32CubeMX.

- Pin Selection:** On the left, the pinout diagram shows the PB12 pin highlighted with a red circle (1). The pin is labeled "PB12" and "GPIO\_Output".
- Peripheral Selection:** In the "System Core" table (2), the "GPIO" row is selected and highlighted with a red dashed box. The table rows include CORTEX\_M55\_FSBL, CORTEX\_M55\_S, GPIO, HPDMAT, ICACHE, IWDG, NVIC1\_S\_Application, NVIC\_FSB, RAMCFG, RCC, SYS\_S, and WWDG.
- Signal Configuration:** In the "Configuration" panel (3), the "Pin Context Assignment" dropdown is set to "Application".
- Output Settings:** In the "Configuration" panel (4), the "GPIO output level" is set to "High". Other settings shown include "GPIO mode: Output Push Pull", "GPIO Pull-up/Pull-down: No pull-up and no pull-down", and "Maximum output speed: Low".

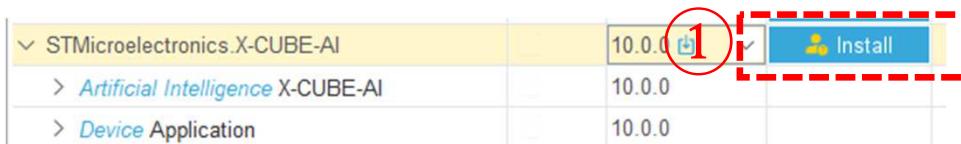


"To use Overdrive Mode (CPU: 800MHz, NPU: 1GHz), set the PB12 pin to High as Output Push-Pull."

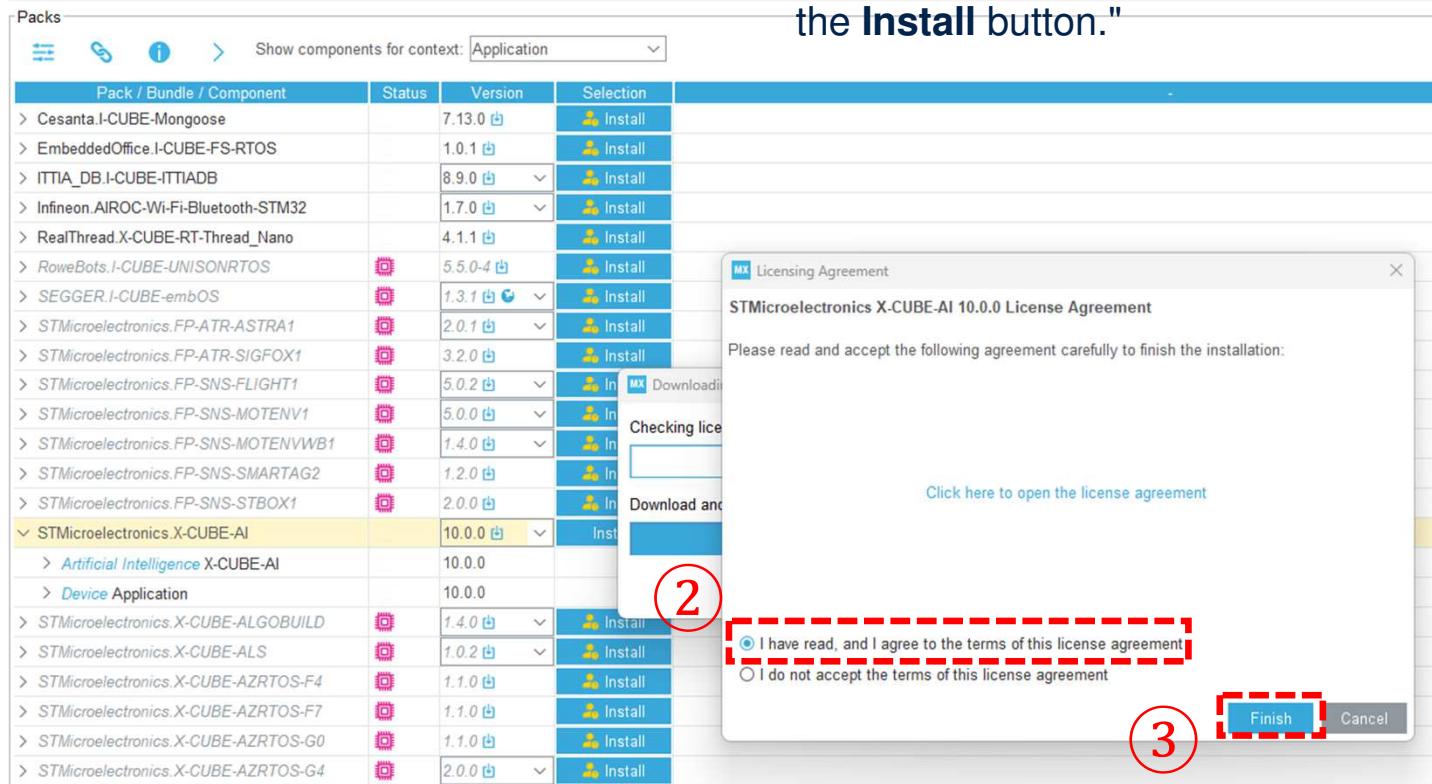
# ST EdgeAI Core Configuration



# ST EdgeAI Core Configuration

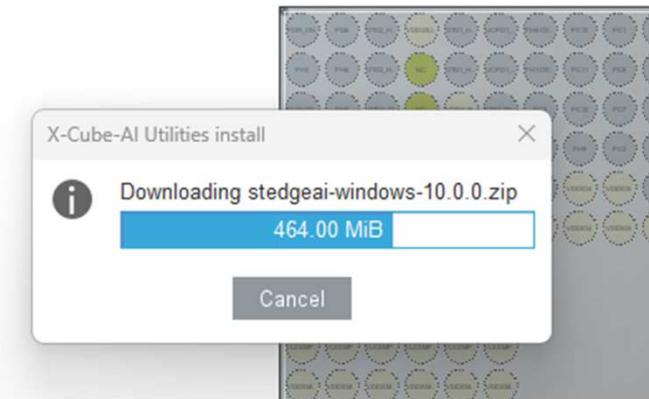
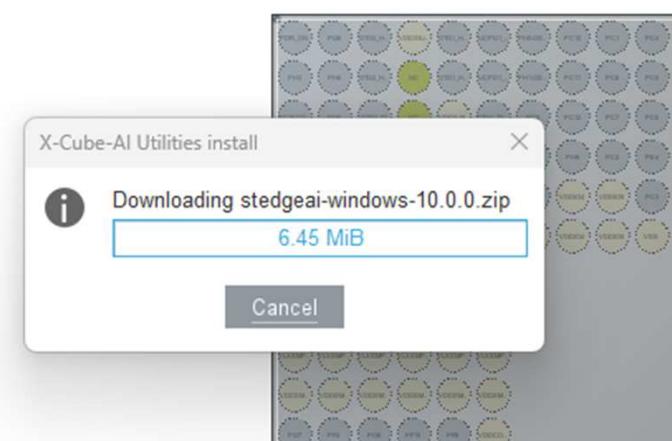


"If X-CUBE-AI is not installed, you can install it by clicking the **Install** button."

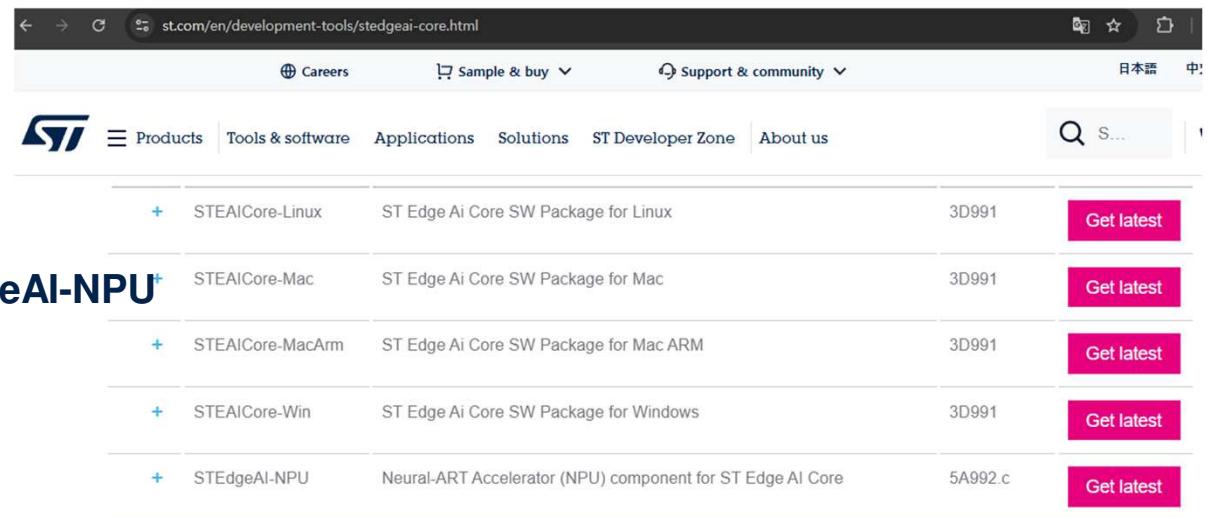
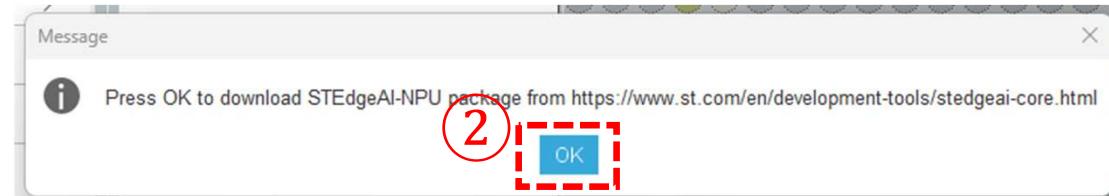
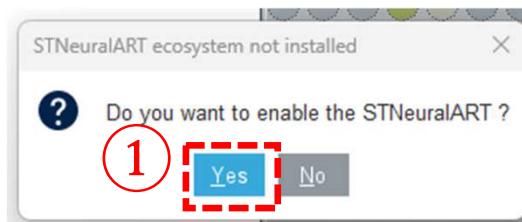


# ST EdgeAI Core Configuration

"The installation is in progress and may take a few minutes."



# ST EdgeAI Core Configuration



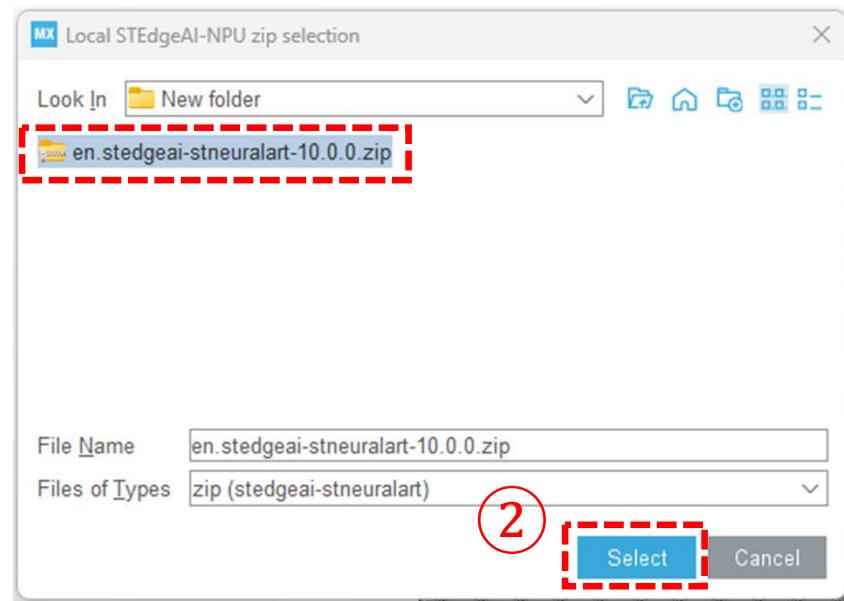
Product	Description	Part Number	Action
STEAICore-Linux	ST Edge Ai Core SW Package for Linux	3D991	<a href="#">Get latest</a>
STEAICore-Mac	ST Edge Ai Core SW Package for Mac	3D991	<a href="#">Get latest</a>
STEAICore-MacArm	ST Edge Ai Core SW Package for Mac ARM	3D991	<a href="#">Get latest</a>
STEAICore-Win	ST Edge Ai Core SW Package for Windows	3D991	<a href="#">Get latest</a>
STEdgeAI-NPU	Neural-ART Accelerator (NPU) component for ST Edge AI Core	5A992.c	<a href="#">Get latest</a>

"In addition, you need to install the **STEdgeAI-NPU Package.**"



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

# ST EdgeAI Core Configuration



"Once the **STEdgeAI-NPU Package** download is complete, load the **en.stedgeai-stneuralart-10.0.0.zip** file and complete the installation."

You may also download the latest version.



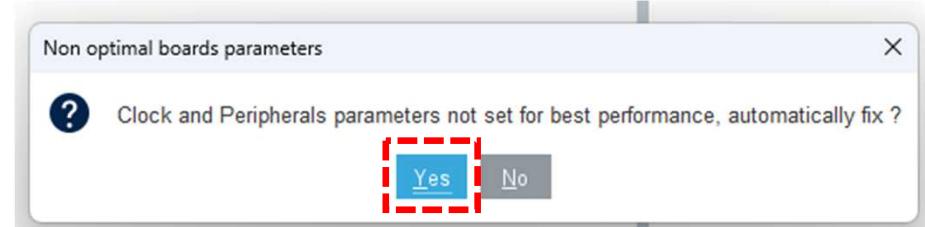
# ST EdgeAI Core Configuration

	FSBL	Application	ExtMemL...
AIROC-Wi-Fi-Bluetooth-ST...			
EXTMEM_LOADER			<input checked="" type="checkbox"/>
EXTMEM_MANAGER	<input type="checkbox"/>		<input checked="" type="checkbox"/>
EXTMEM_MANAGER_APPLI		<input type="checkbox"/>	
I-CUBE-Cesium			
I-CUBE-FS-RTOS			
I-CUBE-ITTIADB			
I-CUBE-Mongoose			
I-CUBE-wolfMQTT			
I-CUBE-wolfSSH			
I-CUBE-wolfSSL			
I-CUBE-wolfTPM			
I-Cube-SoM-uGOAL			
THREADX	<input type="checkbox"/>	<input type="checkbox"/>	
X-CUBE-AI	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

①

Software Packs			Pinout			
STMicroelectronics.X-CUBE-AI.10.0.0_Appl Mode and Configuration						
Mode						
Runtime contexts:						
First Stage Boot Loader	<input type="checkbox"/>	Application	<input checked="" type="checkbox"/>	External Memory Loader		
②						
<input checked="" type="checkbox"/> Artificial Intelligence X-CUBE-AI						
<input checked="" type="checkbox"/> Device Application						

②



"If the popup window appears, click YES."



# ST EdgeAI Core Configuration

The image displays two screenshots of the ST EdgeAI Core Configuration software interface. Both screenshots show a 'Configuration' header bar with 'Reset Configuration', 'Developer Cloud', 'Add network', and 'Delete network' buttons.

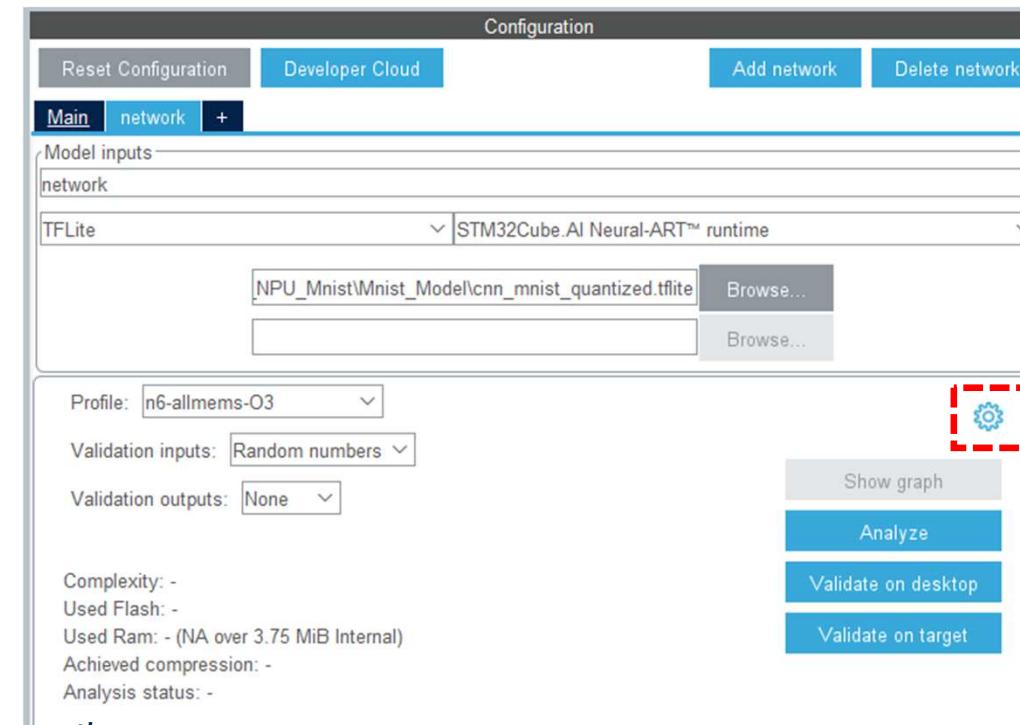
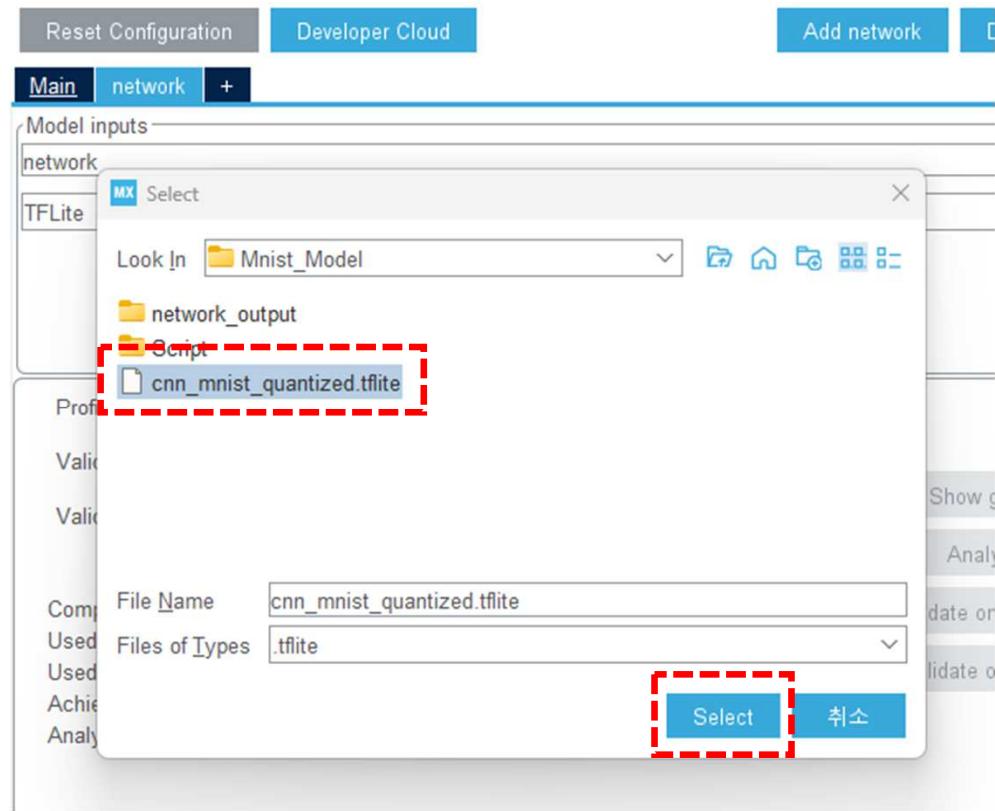
**Left Screenshot:** Shows the 'Main' tab selected. Below it, the 'network' tab is visible. A red dashed box highlights the 'Main' tab. Another red dashed box highlights the 'network' tab area, specifically the 'Model inputs' section where 'network' is listed under 'TFLite'. There are 'Browse...' buttons next to the dropdown menus.

**Right Screenshot:** Shows the 'Main' tab selected. Below it, the 'network' tab is visible. A red dashed box highlights the 'Main' tab. Another red dashed box highlights the 'network' tab area, specifically the 'Model inputs' section where 'TFLite' is selected in the dropdown menu. A red arrow points from the 'TFLite' dropdown in the left screenshot to the 'TFLite' dropdown in the right screenshot. There are 'Browse...' buttons next to the dropdown menus. Below the tabs, there are sections for 'Profile' (set to 'n6-allmems-O3'), 'Validation inputs' (set to 'Random numbers'), and 'Validation outputs' (set to 'None'). A 'Show graph' button is located in the bottom right corner.

Change the **TFLite** network to **STM32Cube.AI Neural-ART runtime**.

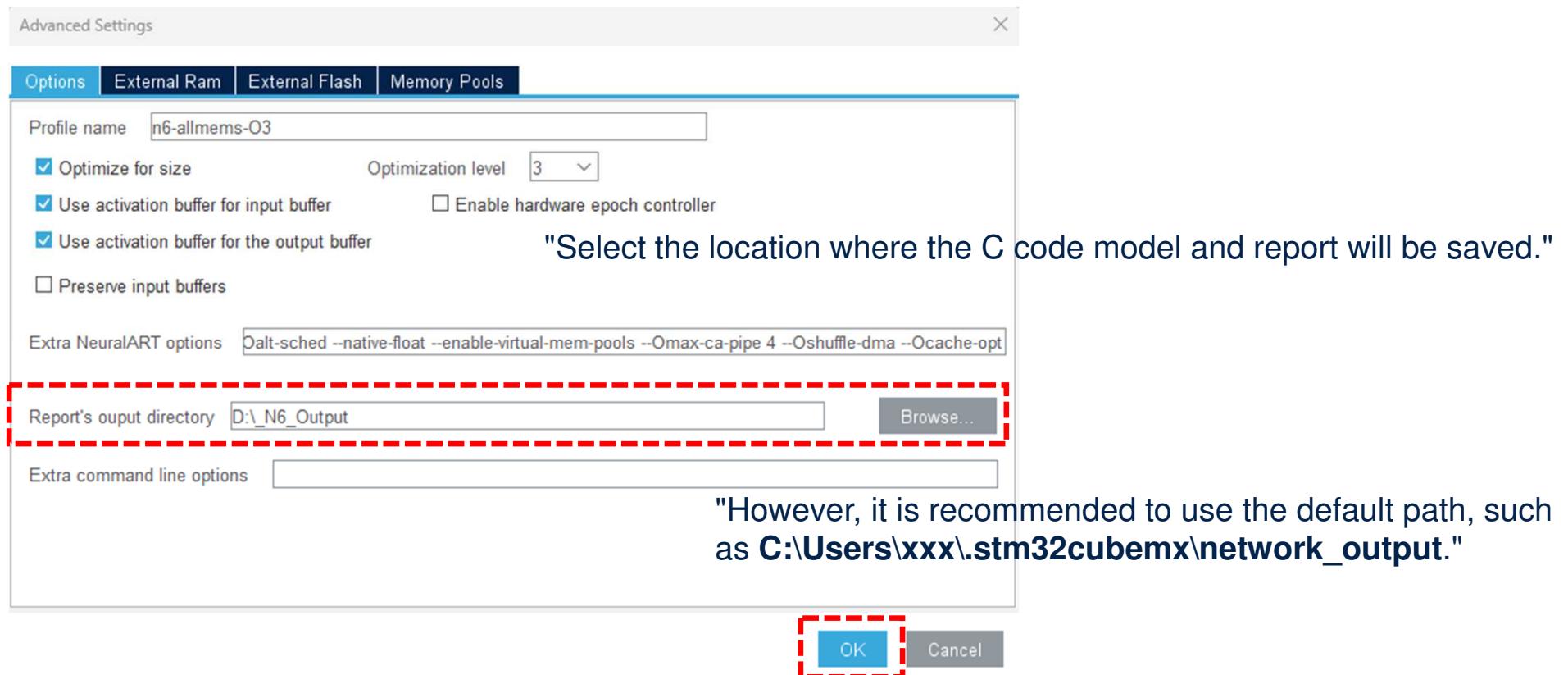


# ST EdgeAI Core Configuration



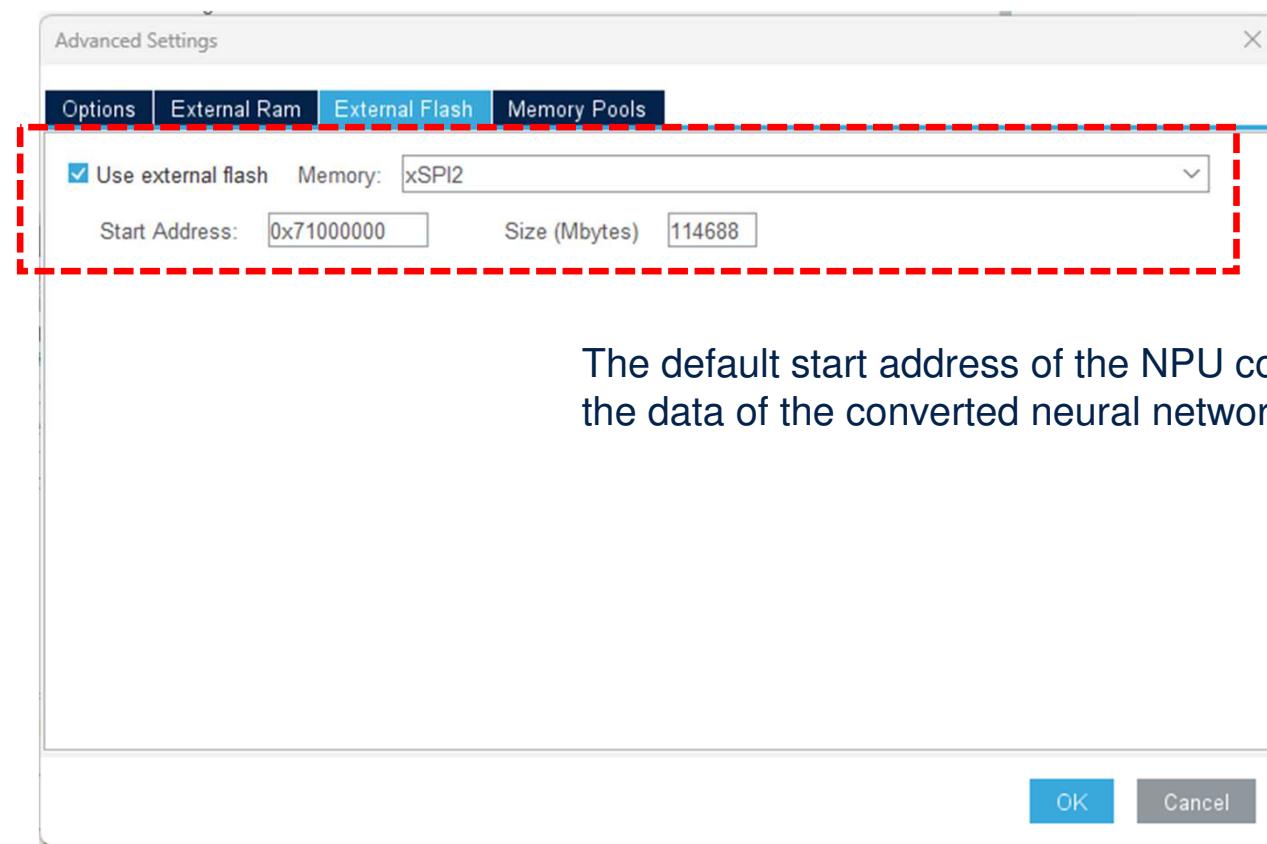
Load the **cnn\_mnist\_quantized.tflite** model from the  
**\_2\_Hands\_On\_NPU\_Mnist\Nist\Nist\Model** folder.

# ST EdgeAI Core Configuration



"Click **OK** to automatically start the network analysis."

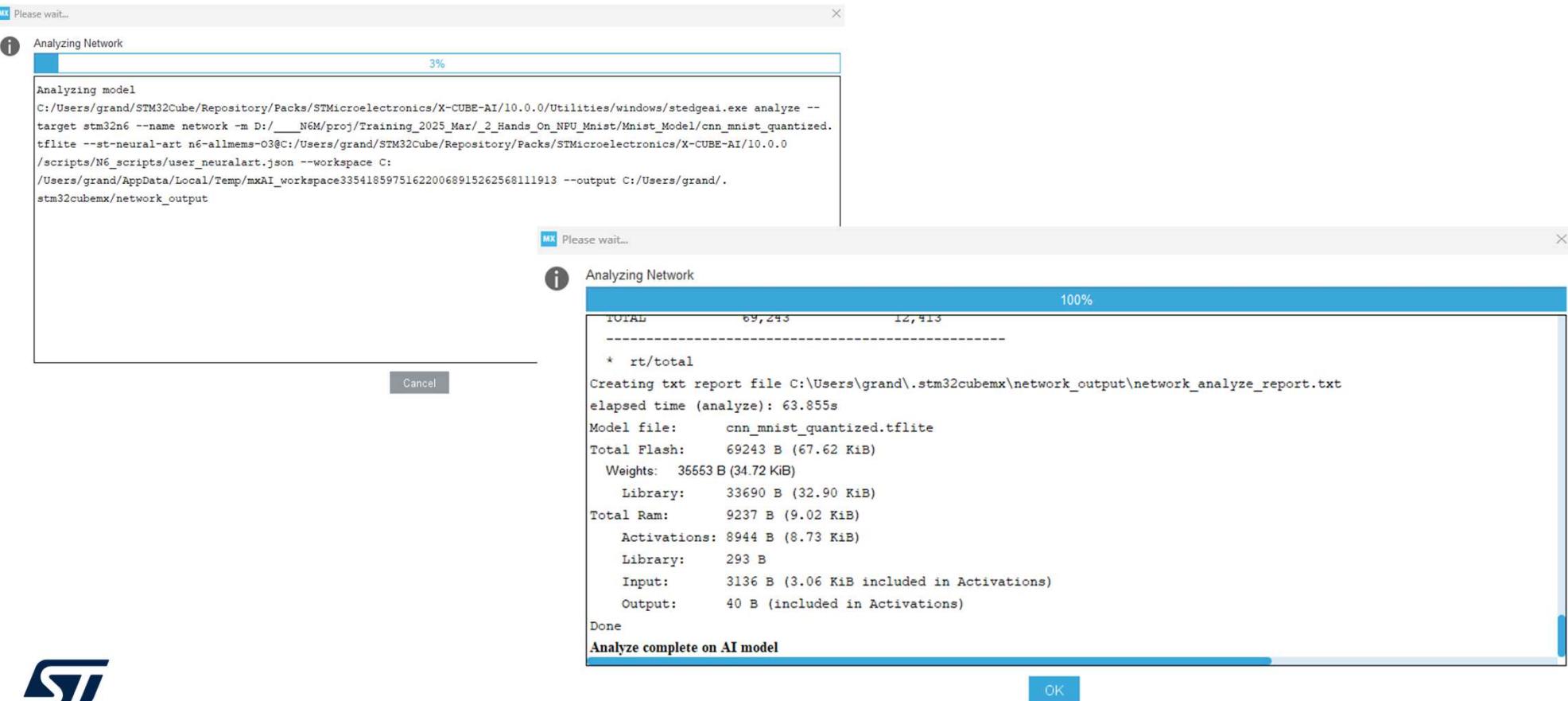
# ST EdgeAI Core Configuration



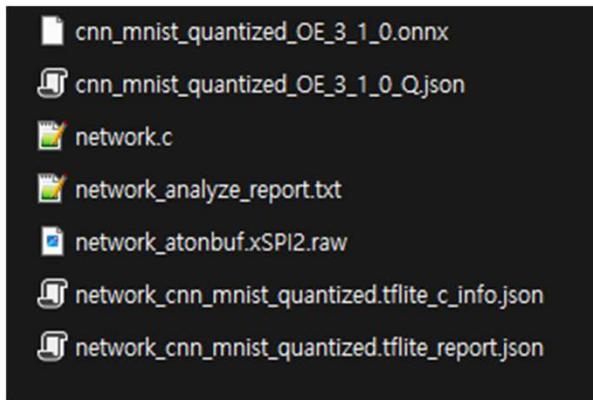
The default start address of the NPU compiler is **0x71000000**. At this address, the data of the converted neural network model will be stored.



# ST EdgeAI Core Configuration



# ST EdgeAI Core Configuration

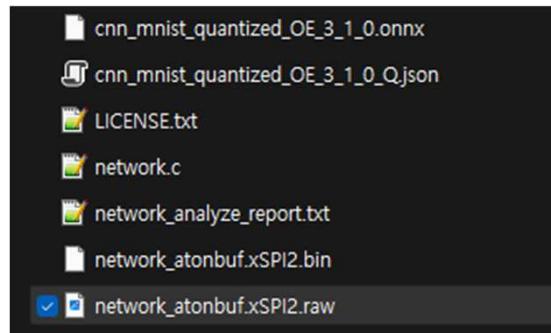


"The **network\_output** folder will contain the generated **report** and **code** files."

```
network_analyze_report.txt
PC에 저장됨
er arguments: -i C:\Users\grand\stm32cubemx\network_output\cnn_mnist_quar
=====
44
45
46 Memory usage information
47 -----
48     flexMEM    [0x34000000 - 0x34000000]:          0 B /          0 B (  0.00
49     cpuRAM1   [0x34064000 - 0x34064000]:          0 B /          0 B (  0.00
50     cpuRAM2   [0x34100000 - 0x34200000]:          0 B /          1.000 MB (  0.00
51     npuRAM3   [0x34200000 - 0x34270000]:          0 B /          448.000 kB (  0.00
52     npuRAM4   [0x34270000 - 0x342E0000]:          0 B /          448.000 kB (  0.00
53     npuRAM5   [0x342E0000 - 0x34350000]:          8.734 kB /          448.000 kB (  1.95
54     npuRAM6   [0x34350000 - 0x343BFFFF8]:          0 B /          447.992 kB (  0.00
55     octoFlash [0x70000000 - 0x73FFFFFF8]:          34.720 kB /          64.000 MB (  0.05
56     hyperRAM  [0x90000000 - 0x91FFFFFF8]:          0 B /          32.000 MB (  0.00
57 -----
58 Total:                               43.454 kB
-----
60
61 Used memory ranges
62 -----
63     npuRAM5   [0x342E0000 - 0x34350000]: 0x342E0000-0x342E22F0
64     octoFlash [0x70000000 - 0x73FFFFFF8]: 0x70000000-0x70008AF0
65 -----
66 Epochs details
67 -----
68 Total number of epochs: 9 of which 3 implemented in software
69 -----
70 epoch ID  HW/SW/EC Operation (SW only)
71 epoch 1    HW
72 epoch 2    -SW- ( QuantizeLinear )
73 epoch 3    HW
74 epoch 4    HW
75 epoch 5    HW
76 epoch 6    HW
77 epoch 7    HW
78 epoch 8    -SW- ( Softmax )
79 epoch 9    -SW- ( DequantizeLinear )
80 -----
81 Requested memory size by section - "stm32n6npu" target
82 -----
83
84 -----
```



# ST EdgeAI Core Configuration -



"Rename the file **network\_atonbuf.xSPI2.raw** to **network\_atonbuf.xSPI2.bin**, changing the extension from **.raw** to **.bin**."

"In this example, the network binary is stored at address **0x71000000**."

"Run **STM32CubeProgrammer**, load **network\_atonbuf.xSPI2.bin**, and download it to **0x71000000**."

A screenshot of the STM32CubeProgrammer software interface. The main window title is "Prg STM32CubeProgrammer". The left sidebar has icons for Erasing &amp; Programming, Download, DTP, and REG. The "Erasing &amp; Programming" tab is active. In the "Download" section, the "File path" is set to "C:\Users\grand\stm32cubemx\network\_out", the "Start address" is "0x71000000", and several checkboxes are checked: "Skip flash erase before programming", "Verify programming", "Full Flash memory checksum", and "Run after programming". A "Start Programming" button is located at the bottom of this section. To the right, there is a "Data" panel showing a table for "Erase flash memory" with five entries (Index 0 to 4, Address 0x7) and a "Log" panel displaying a timeline of events during the programming process.



# ST EdgeAI Core Configuration -

```
[ ] model = keras.Sequential(  
[  
    keras.Input(shape=input_shape),  
    layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),  
    layers.MaxPooling2D(pool_size=(2, 2)),  
    layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),  
    layers.MaxPooling2D(pool_size=(2, 2)),  
    layers.Flatten(),  
    layers.Dropout(0.5),  
    layers.Dense(num_classes, activation="softmax"),  
]  
)  
  
model.summary()
```

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)  
converter.optimizations = [tf.lite.Optimize.DEFAULT]  
converter.representative_dataset = representative_dataset  
converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS_INT8]  
converter.inference_input_type = tf.float32  
converter.inference_output_type = tf.float32
```

"The NN used in this example is a **quantized model** with both **input and output in float32 format**."



```
network_analyze_report.txt  
-----  
4  
5 Model name - cnn_mnist_quantized  
6 -----  
7 m_id layer (original) oshape  
8 -----  
9 0 serving_default_input_10 () [b:1,h:28,w:28,c:1]  
10 | conversion_0 (QUANTIZE) [b:1,h:28,w:28,c:1]  
11 -----  
12 1 conv2d_1 (CONV_2D) [b:1,h:26,w:26,c:32]  
13 | nl_1_nl (CONV_2D) [b:1,h:26,w:26,c:32]  
14 -----  
15 2 pool_2 (MAX_POOL_2D) [b:1,h:13,w:13,c:32]  
16 -----  
17 3 conv2d_3 (CONV_2D) [b:1,h:11,w:11,c:64]  
18 | nl_3_nl (CONV_2D) [b:1,h:11,w:11,c:64]  
19 -----  
20 4 pool_4 (MAX_POOL_2D) [b:1,h:5,w:5,c:64]  
21 -----  
22 5 reshape_5 (RESHAPE) [b:1,c:1600]  
23 -----  
24 6 sequential_dense_MatMul () [b:10,c:1600]  
25 | sequential_dense_B..VariableOp () [b:10]  
26 | gemm_6 (FULLY_CONNECTED) [b:1,c:10]  
27 -----  
28 -----  
29 -----  
30 7 nl_7 (SOFTMAX) [b:1,c:10]  
31 -----  
32 8 conversion_8 (DEQUANTIZE) [b:1,c:10]  
33 -----  
34 model: macc=2,500,212 weights=35,144 activations=-- io=--  
35 -----  
36 -----  
Epochs details  
-----  
Total number of epochs: 9 of which 3 implemented in software  
---  
epoch ID HW/SW/EC Operation (SW only)  
epoch 1 HW  
epoch 2 -SW- ( QuantizeLinear )  
epoch 3 HW  
epoch 4 HW  
epoch 5 HW  
epoch 6 HW  
epoch 7 HW  
epoch 8 -SW- ( Softmax )  
epoch 9 -SW- ( DequantizeLinear )  
=====
```

# RIF Configuration

Home > STM32N657X0HxQ > **stm32n6\_nucleo\_npu\_nn\_mnist.ioc - RIF** >

	Pinout & Configuration	Clock Configuration	RIF	
Peripherals (RISUP)	Global lock : OFF			
	RIMU IP	RIMU ID	MASTER CID	SECURE
	DCMIPP	9	0	<input type="checkbox"/>
	DMA2D	8	0	<input type="checkbox"/>
	ETH1	6	0	<input type="checkbox"/>
	ETR	0	0	<input type="checkbox"/>
Domains (RIMU)	GPU	7	0	<input type="checkbox"/>
	LTDC_L1	10	0	<input type="checkbox"/>
	LTDC_L2	11	0	<input type="checkbox"/>
	NPU	1	1	<input checked="" type="checkbox"/>
	OTG1	4	0	<input type="checkbox"/>
	OTG2	5	0	<input type="checkbox"/>
	SDMMC1	2	0	<input type="checkbox"/>
	SDMMC2	3	0	<input type="checkbox"/>
	VENC	12	0	<input type="checkbox"/>

Users can configure the RIF as needed.



# Project Configuration

Pinout & Configuration		Clock Configuration	RIF	Project Manager
Project	Project Settings			
	Project Name	stm32n6_nucleo_npu_nn_mnist		
	Project Location	D:\N6M\proj\Training_2025_Mar\2_Hands_On_NPU_Mnist		
Code Generator	Project Structure	<input checked="" type="checkbox"/> FSBL <input checked="" type="checkbox"/> Appli <input type="checkbox"/> ExtMemLoader		
	Application Structure	Advanced		
	Toolchain Folder Location	D:\N6M\proj\Training_2025_Mar\2_Hands_On_NPU_Mnist\stm32n6_nucleo_npu_nn_mnist\		
	Toolchain / IDE	STM32CubeIDE <input type="checkbox"/> Generate Under Root		



# Project Configuration

The screenshot shows the Project Configuration interface with three tables:

- Generated Function Calls Application**: A table with columns "Generate Code", "Rank", and "Function Name". The first row has "Generate Code" checked and "Rank" 1, with "Function Name" SystemClock\_Config. A note says: "In this example, uncheck the checkbox to **disable automatic code generation**."
- Generated Function Calls ExtMemLoader**: A table with columns "Generate Code", "Rank", and "Function Name". The first row has "Generate Code" checked and "Rank" 1, with "Function Name" MX\_GPIO\_Init. A red dashed box highlights the "Generate Code" column for rows 1 through 10.
- Generated Function Calls ExtMemLoader**: A table with columns "Generate Code", "Rank", and "Function Name". The first row has "Generate Code" checked and "Rank" 1, with "Function Name" MX\_GPIO\_Init. This table is identical to the one above it.

"Finally, click **GENERATE CODE**."

Generate Code	Rank	Function Name
<input checked="" type="checkbox"/>	1	SystemClock_Config
<input type="checkbox"/>	2	MX_GPIO_Init
<input type="checkbox"/>	3	MX_BSEC_Init
<input type="checkbox"/>	4	MX_CACHEAXI_Init
<input type="checkbox"/>	5	MX_XSPI2_Init
<input type="checkbox"/>	8	MX_XSPIM_Init
<input type="checkbox"/>	10	MX_X_CUBE_AI_Init
<input type="checkbox"/>		MX_RIF_Init

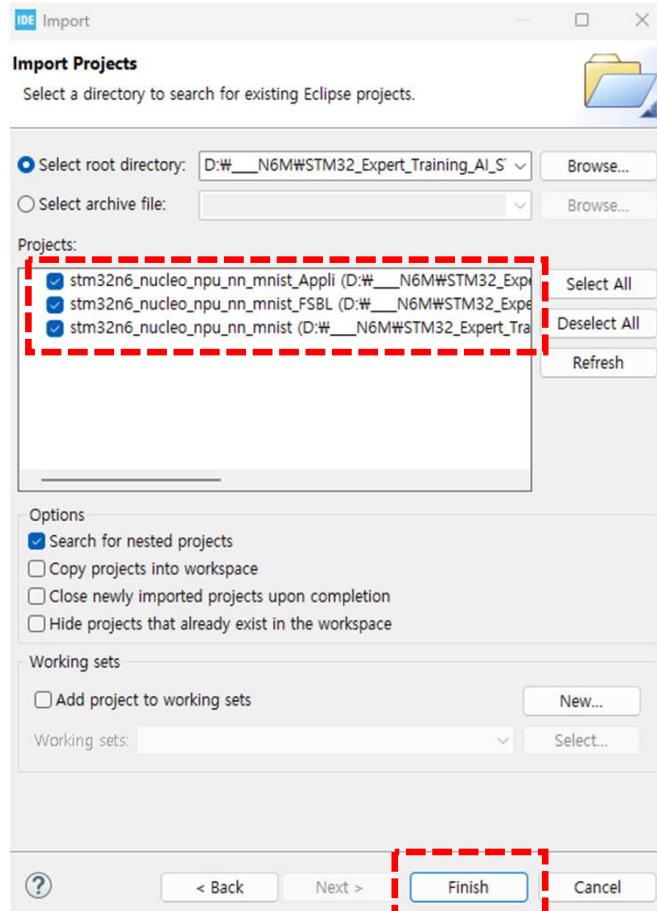
Generate Code	Rank	Function Name
<input checked="" type="checkbox"/>	1	MX_GPIO_Init
<input type="checkbox"/>	2	MX_BSEC_Init
<input type="checkbox"/>	3	MX_CACHEAXI_Init
<input type="checkbox"/>	4	MX_XSPI2_Init
<input type="checkbox"/>	5	MX_XSPIM_Init
<input type="checkbox"/>	8	MX_X_CUBE_AI_Init
<input type="checkbox"/>	10	MX_RIF_Init



"Finally, click **GENERATE CODE**."

**GENERATE CODE**

# STM32CubeIDE Import Project



The screenshot shows the STM32CubeIDE interface with the following components:

- Project Explorer**: Shows the imported project structure:
  - stm32n6\_nucleo\_fsb\_lrun\_led\_toggle
  - stm32n6\_nucleo\_npu\_nn\_mnist
    - Drivers
    - Middlewares
    - Secure nsclib
    - stm32n6\_nucleo\_npu\_nn\_mnist\_Appli (in Appli)
      - Includes
      - Core
      - Inc
      - Src
        - main.c
        - secure\_nsclib.c
        - stm32n6xx\_hal\_msp.c
        - stm32n6xx\_it.c
        - syscalls.c
        - systemmem.c
        - system\_stm32n6xx\_s.c
      - Startup
      - Drivers
      - X-CUBE-AI
        - STM32N657X0HXQ\_LRUN\_RAMxspi1.ld
        - STM32N657X0HXQ\_LRUN\_RAMxspi2.ld
        - STM32N657X0HXQ\_LRUN.ld
        - STM32N657X0HXQ\_ROMxspi1\_RAMxspi1.ld
        - STM32N657X0HXQ\_ROMxspi1.ld
        - STM32N657X0HXQ\_ROMxspi1xspi2\_RAMxspi3.ld
        - STM32N657X0HXQ\_ROMxspi2\_RAMxspi1.ld
        - STM32N657X0HXQ\_ROMxspi3\_RAMxspi1xspi2.ld
        - STM32N657X0HXQ\_ROMxspi3.ioc
  - Code Editor**: The 'main.c' file content is displayed:

```
1 /* USER CODE BEGIN Header */
2 /**
3  * @file           : main.c
4  * @brief          : Main program body
5  * @attention
6  *
7  * Copyright (c) 2025 STMicroelectronics
8  * All rights reserved.
9  *
10 * This software is licensed under terms
11 * in the root directory of this software package.
12 * If no LICENSE file comes with this software,
13 * one must be obtained from
14 * STMicroelectronics.
15 */
16 /**
17 */
18 /* USER CODE END Header */
19 /* Includes -----*/
20 #include "main.h"
21 #include "app_x-cube-ai.h"
22
23 /* Private includes -----*/
24 /* USER CODE BEGIN Includes */
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
36 /* USER CODE END PD */
37
38 /* Private macro -----*/
39
```

In this hands-on, **FSBL** is not used; only the **Secure Application** is used.



# STM32CubeIDE

"Copy the files from **\_2\_Hands\_On\_NPU\_Mnist/Required\_Code** into the newly created project."



"Copy the **BSP** folder into the **Drivers** folder of the newly created project."

"Copy the **app\_fuseprogramming.c** and **system\_clock\_config.c** files into the **App\src** folder."

"Copy the **app\_fuseprogramming.h** file into the **App\inc** folder."

# STM32CubeIDE

"In this example, **MX\_X\_CUBE\_AI\_Process()** is not used."

The screenshot shows the STM32CubeIDE interface with two code editors displaying the same `main.c` file. The left editor shows the original code, and the right editor shows the code with a specific line commented out.

**Left Editor (Original Code):**

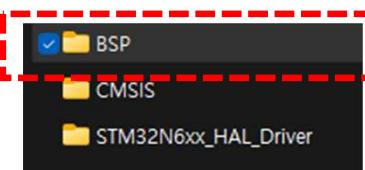
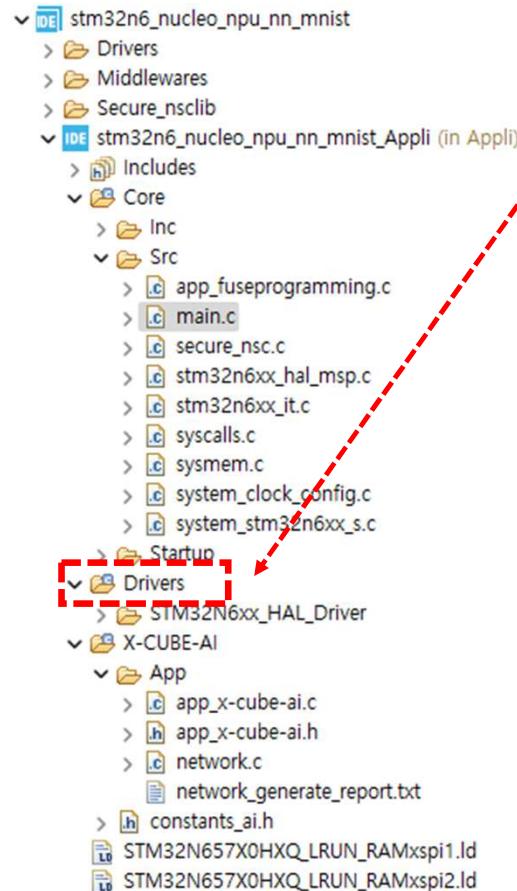
```
94  /* USER CODE END SysInit */  
95  
96  /* Initialize all configured peripherals */  
97  MX_GPIO_Init();  
98  /* USER CODE BEGIN 2 */  
99  
100 /* USER CODE END 2 */  
101  
102 /* Infinite loop */  
103 /* USER CODE BEGIN WHILE */  
104 while (1)  
105 {  
106     /* USER CODE END WHILE */  
107  
108     MX_X_CUBE_AI_Process();  
109     /* USER CODE BEGIN 3 */  
110 }  
111 /* USER CODE END 3 */  
112 }  
113  
114/**  
115 * @brief GPIO Initialization Function  
116 * @param None  
117 * @retval None  
118 */
```

**Right Editor (Modified Code):**

```
94  /* USER CODE END SysInit */  
95  
96  /* Initialize all configured peripherals */  
97  MX_GPIO_Init();  
98  /* USER CODE BEGIN 2 */  
99  
100 /* USER CODE END 2 */  
101  
102 /* Infinite loop */  
103 /* USER CODE BEGIN WHILE */  
104 while (1)  
105 {  
106     /* USER CODE END WHILE */  
107  
108 //MX_X_CUBE_AI_Process();  
109     /* USER CODE BEGIN 3 */  
110 }  
111 /* USER CODE END 3 */  
112 }  
113
```

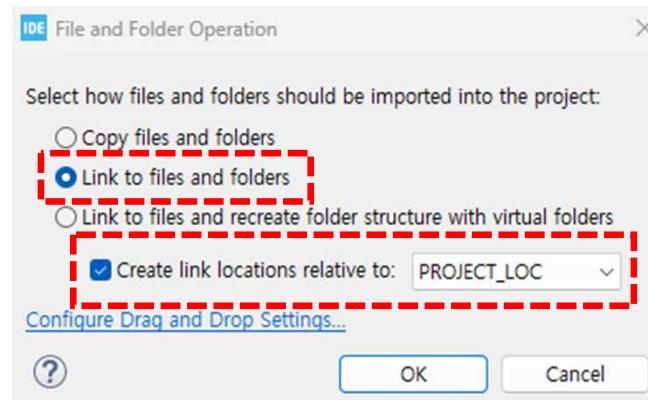


# STM32CubeIDE

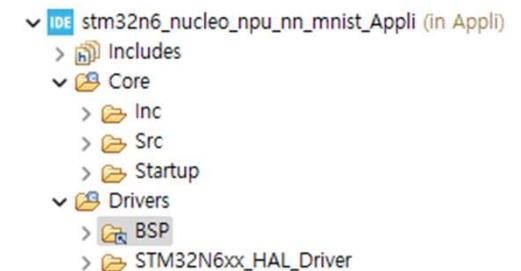


"Drag and drop the **BSP** folder from the **Drivers** folder into the **Drivers** folder in STM32CubeIDE's **Project Explorer** view."

If the **Middleware** folder is also not generated automatically, proceed as follows.

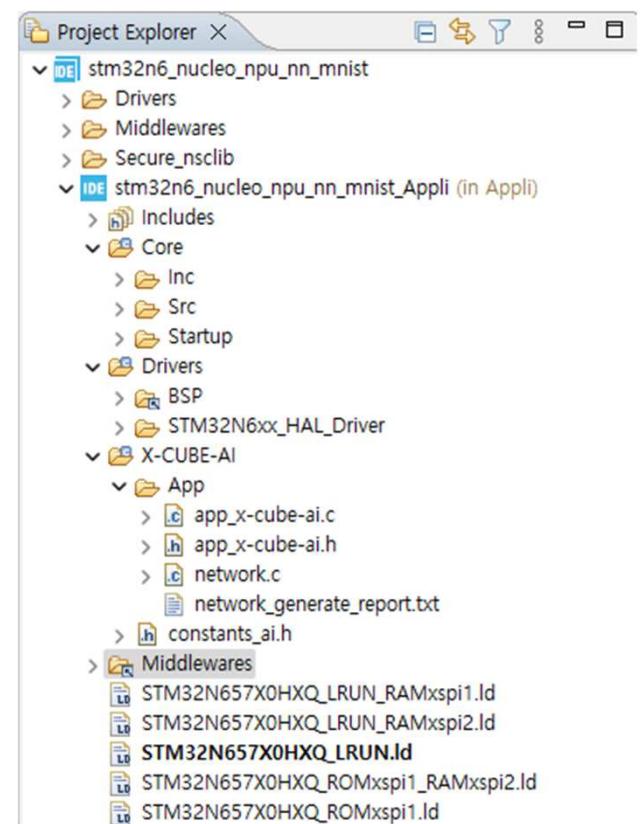
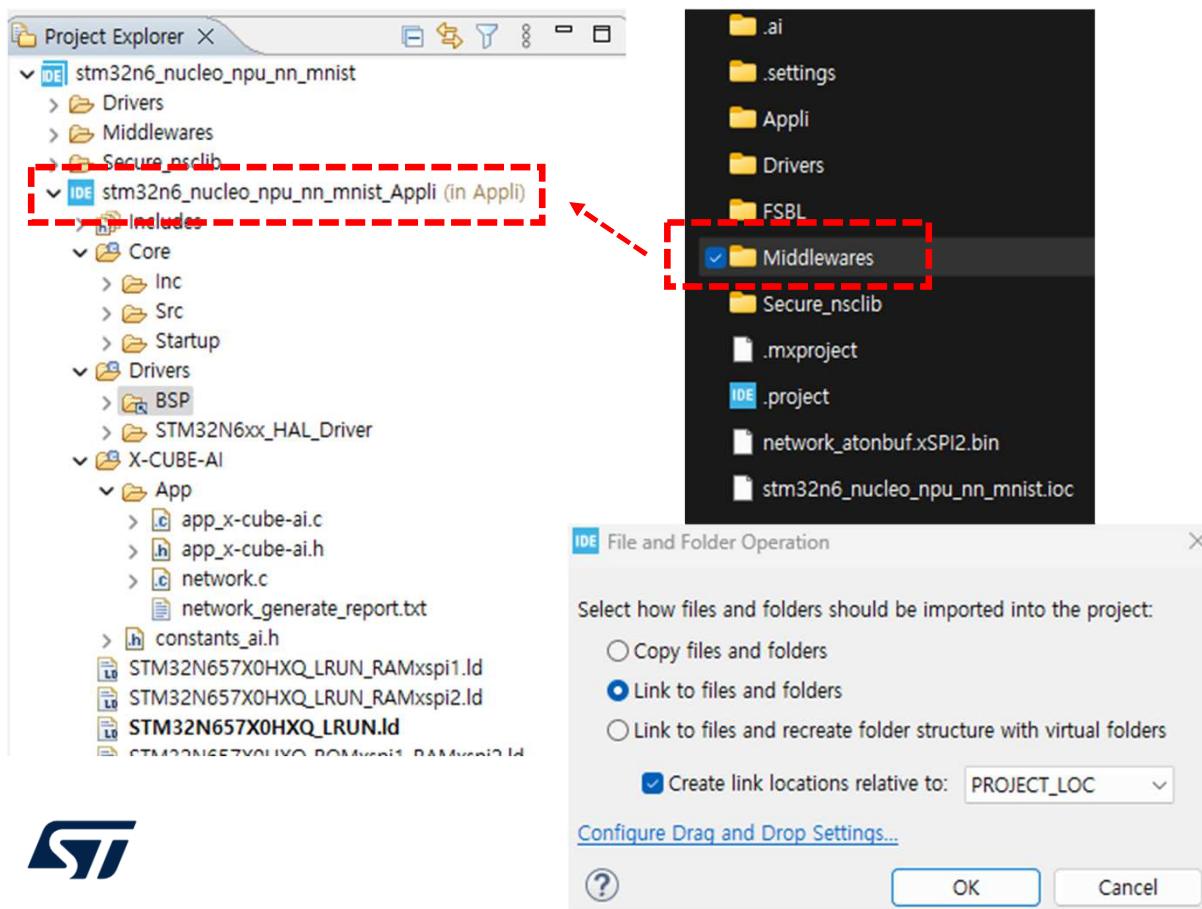


"Select '**Link to files and folders**', then choose '**Create Link locations relative to: PROJECT\_LOC**'."

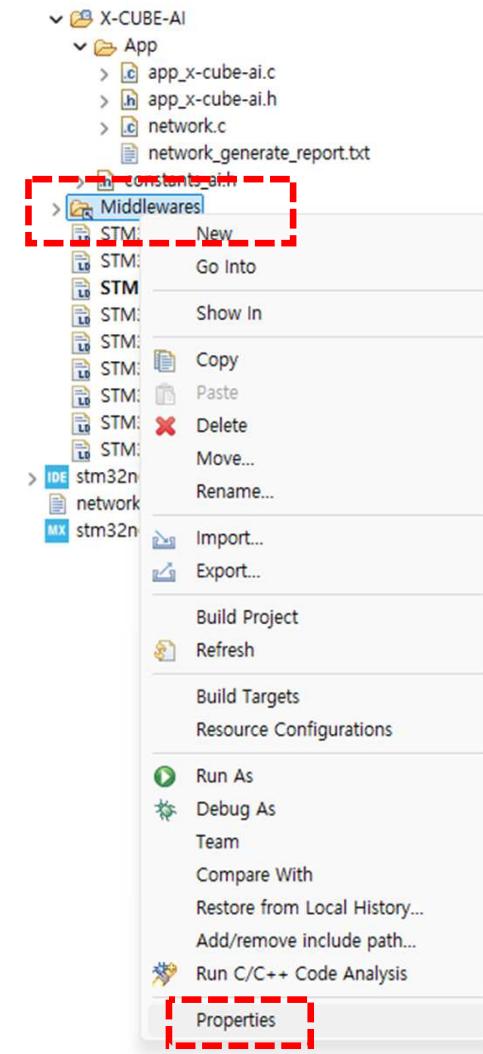


# STM32CubeIDE

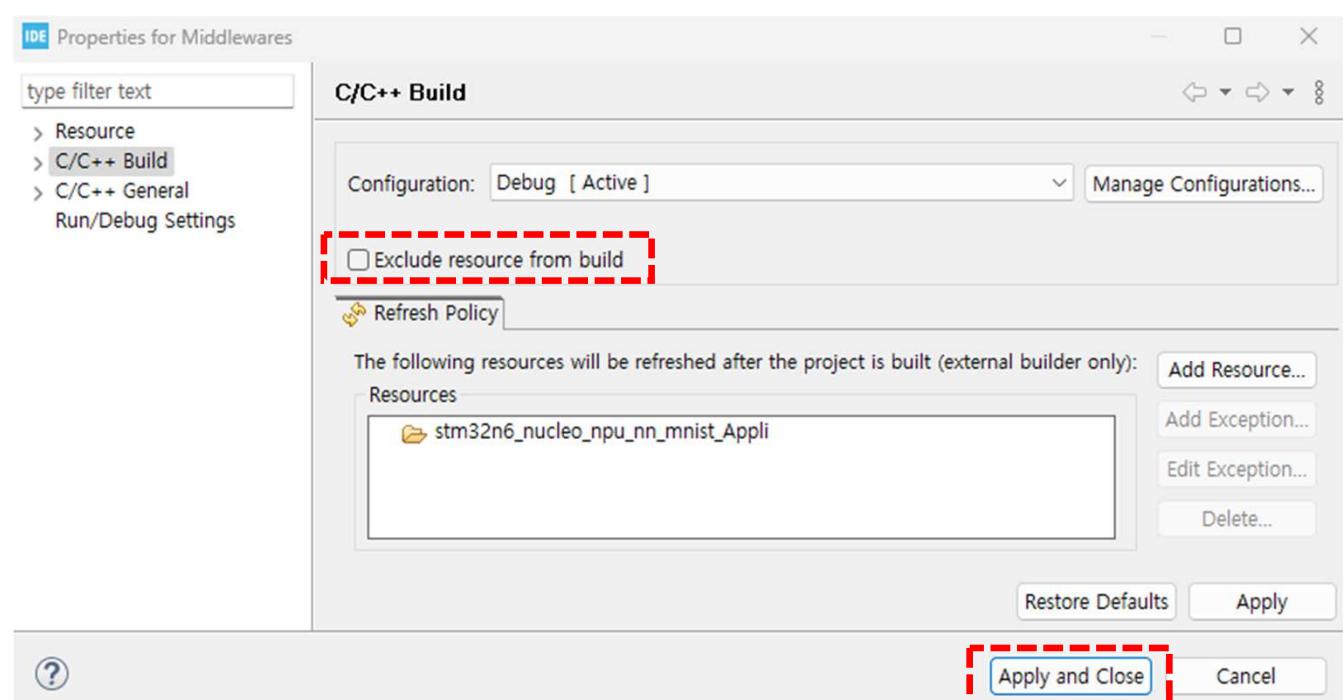
"Drag and drop the **Middlewares** folder to create a link."



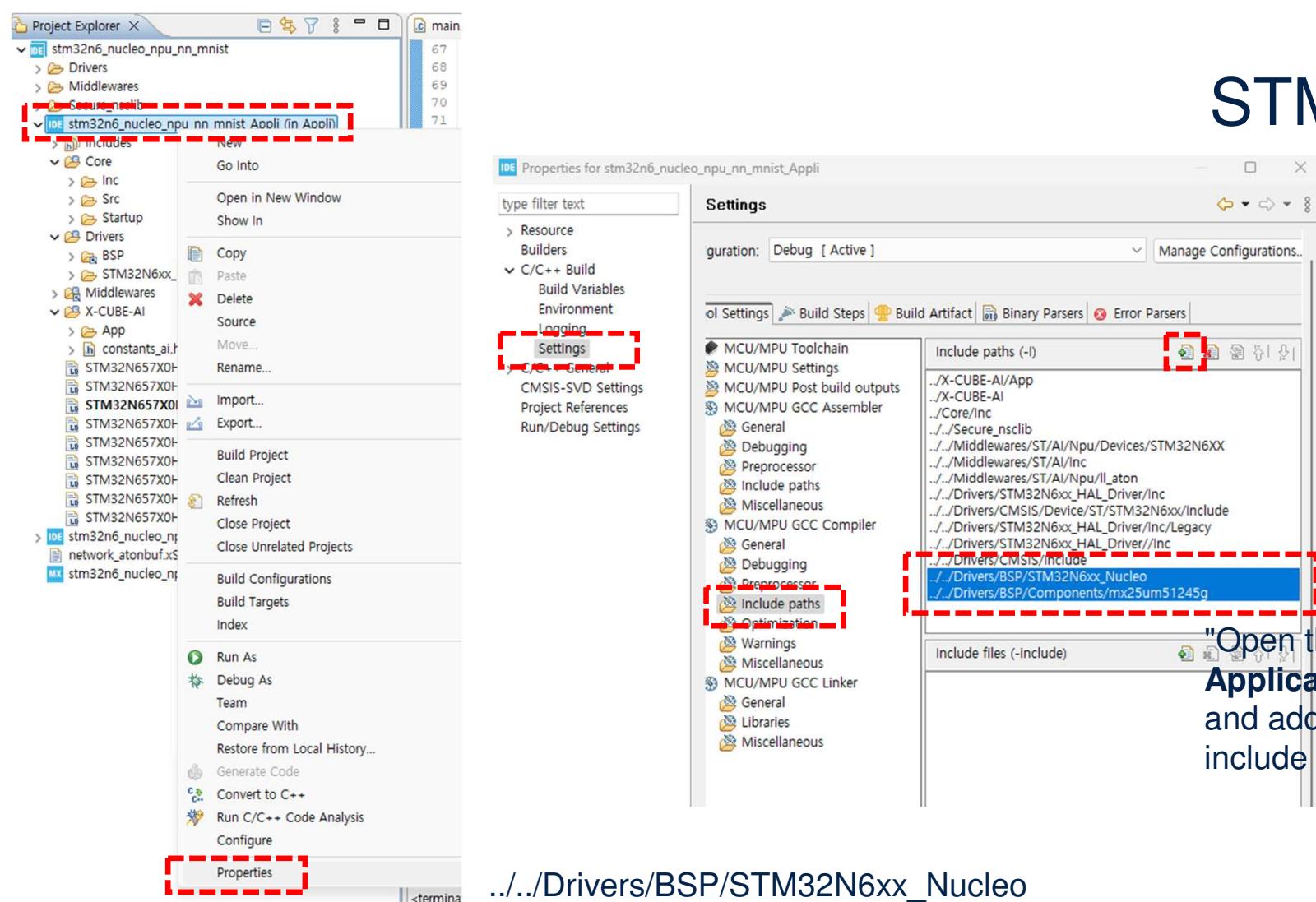
# STM32CubeIDE



"Right-click on the **Middlewares** folder, select **Properties**, uncheck **Exclude resource from build**, then click **Apply and Close**."



# STM32CubeIDE

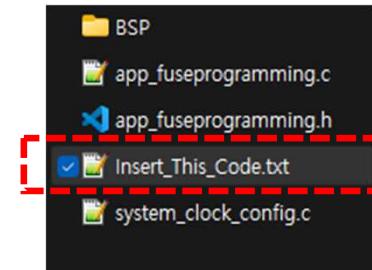


.../Drivers/BSP/STM32N6xx\_Nucleo  
.../Drivers/BSP/Components/mx25um51245g

# STM32CubeIDE

The screenshot shows the STM32CubeIDE interface. On the left is the Project Explorer, displaying the project structure:

- stm32n6\_nucleo\_npu\_nn\_mnist
- Drivers
- Middlewares
- Secure\_nsclib
- stm32n6\_nucleo\_npu\_nn\_mnist\_Appli (in Appli)
  - Binaries
  - Includes
  - Core
    - Inc
    - Src
      - app\_fuseprogramming.c
      - main.c
      - secure\_nscc.c
      - stm32n6xx\_hal\_msp.c
      - stm32n6xx\_it.c
      - syscalls.c
      - sysmem.c
      - system\_clock\_config.c
      - system\_stm32n6xx.c
    - Startup
  - Drivers



A screenshot of the code editor showing the contents of the file `Insert_This_Code.txt`:

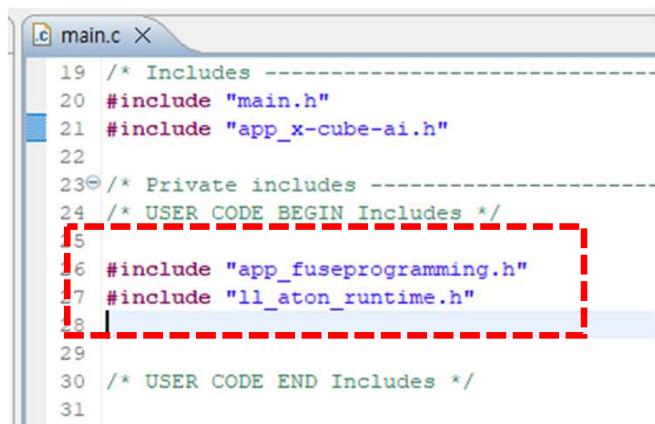
```
/* USER CODE BEGIN Includes */
#include "app_fuseprogramming.h"
#include "ll_aton_runtime.h"
/* USER CODE END Includes */
/* USER CODE BEGIN PD */
LL_ATON_DECLARE_NAMED_NN_INSTANCE_AND_INTERFACE(Default) // Defines
/* USER CODE END PD */
/* USER CODE BEGIN O */
/* USER CODE END O */
/* ***** TIME MEASUREMENT *****
void time_in(void);
uint32_t time_out(void);
uint32_t duration_us;
uint32_t duration_dwt;
static uint32_t t_init;
static uint32_t t_out;
float_t clock_Hz;
uint32_t cpclk;
***** */


```

"Prepare to copy the code from `Insert_This_Code.txt` into the `main.c` file of the current project."

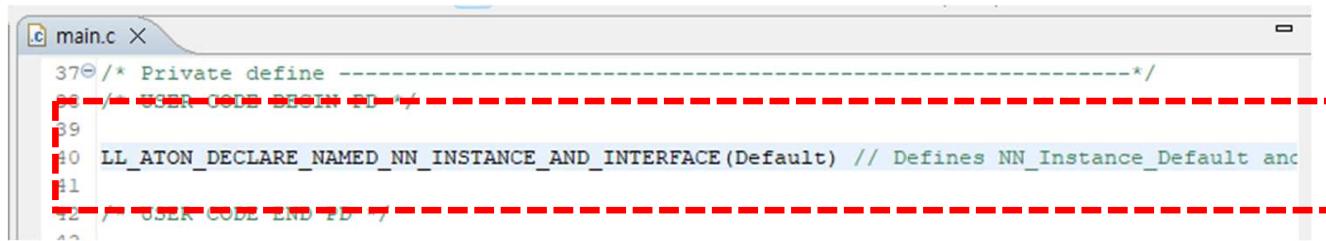


# STM32CubeIDE



main.c

```
19 /* Includes -----  
20 #include "main.h"  
21 #include "app_x-cube-ai.h"  
22  
23/* Private includes -----  
24 /* USER CODE BEGIN Includes */  
25  
26 #include "app_fuseprogramming.h"  
27 #include "ll_aton_runtime.h"  
28  
29  
30 /* USER CODE END Includes */  
31
```

A screenshot of the STM32CubeIDE interface showing the main.c file. The code editor displays C code with syntax highlighting. A red dashed box highlights the code block starting at line 24, which is preceded by a multi-line comment block. This indicates a user-defined code section.

main.c

```
37/* Private define -----*/  
38/* USER CODE BEGIN PD */  
39  
40 LL_ATON_DECLARE_NAMED_NN_INSTANCE_AND_INTERFACE(Default) // Defines NN_Instance_Default and  
41  
42 /* USER CODE END PD */
```

A screenshot of the STM32CubeIDE interface showing the main.c file. The code editor displays C code with syntax highlighting. A red dashed box highlights the code block starting at line 38, which is preceded by a multi-line comment block. This indicates a user-defined code section.

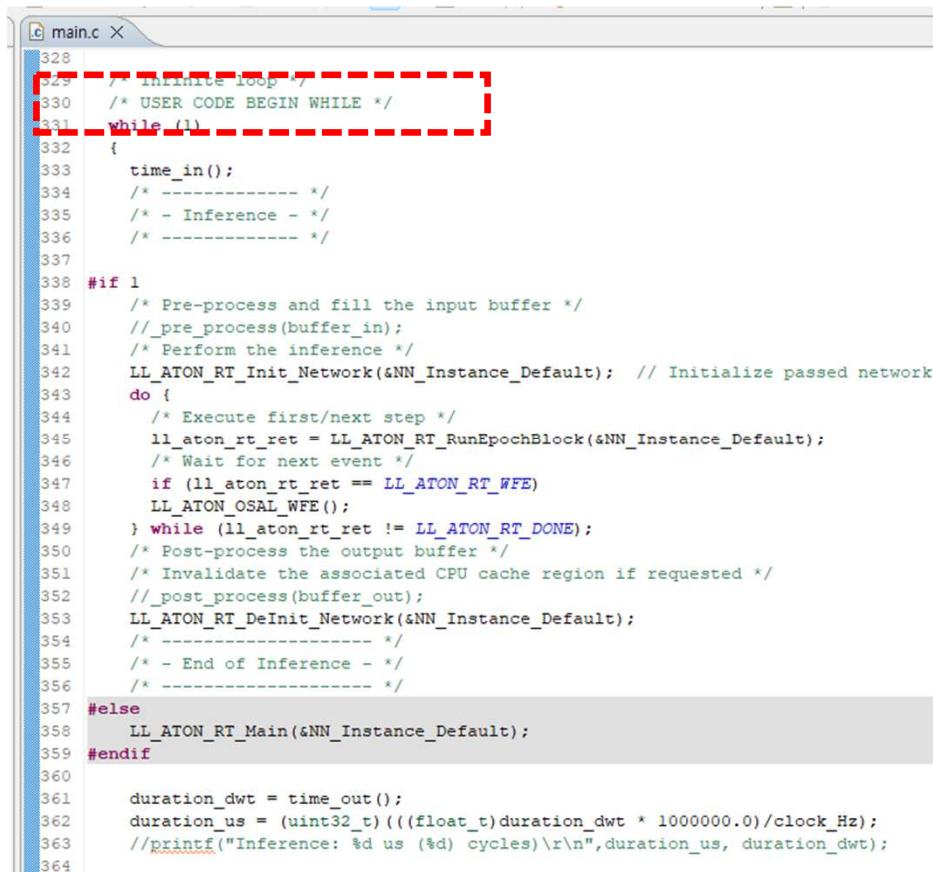
# STM32CubeIDE

```
main.c X
59 /* USER CODE END PFP */
60
61 // Private User Code
62 /* USER CODE BEGIN 0 */
63
64 //***** TIME MEASUREMENT *****
65 /*
66 //***** *****
67 void time_in(void);
68 uint32_t time_out(void);
69
70 uint32_t duration_us;
71 uint32_t duration_dwt;
72 static uint32_t t_init;
73 static uint32_t t_out;
74 float_t clock_Hz;
75 uint32_t cpufreq;
76
77 //***** DWT INITIALIZATION *****
78 /*
79 //***** *****
80 void init_dwt()
81 {
82     /* Enable Trace */
83     CoreDebug->DEMCR |= CoreDebug_DEMCR_TRCENA_Msk;
84
85     /* Reset Cycle Counter and Event Counters */
```

```
main.c X
68 /* USER CODE BEGIN Init */
69
70 MX_GPIO_Init();
71 Fuse_Programming();
72 SystemClock_Config_HSI_overdrive();
73 init_external_memories();
74
75 NPURam_enable();
76
77 NPUCache_enable();
78
79 /* Set all required IPs as secure privileged */
80 Security_Config();
81
82
83 //***** NN INIT *****
84 /*
85 //***** *****
86 LL_ATON_RT_RetValues_t ll_aton_rt_ret = LL_ATON_RT_DONE;
87 const EpochBlock_HandleTypeDef *eb_list = LL_ATON_EpochBlockItems_Default();
88
89 /* Retrieve the start address of the input and output buffer
90 (reserved in the activation buffer) */
91 const LL_Buffer_InfoTypeDef *ibuffersInfos = NN_Interface_Default.input_buffers_
92 const LL_Buffer_InfoTypeDef *obuffersInfos = NN_Interface_Default.output_buffers_
93 buffer_in = (float *)LL_Buffer_addr_start(&ibuffersInfos[0]);
94 buffer_out_p = (float *)LL_Buffer_addr_start(&obuffersInfos[0]);
95
96 LL_ATON_RT_RuntimeInit();
97
98 //***** *****/
```



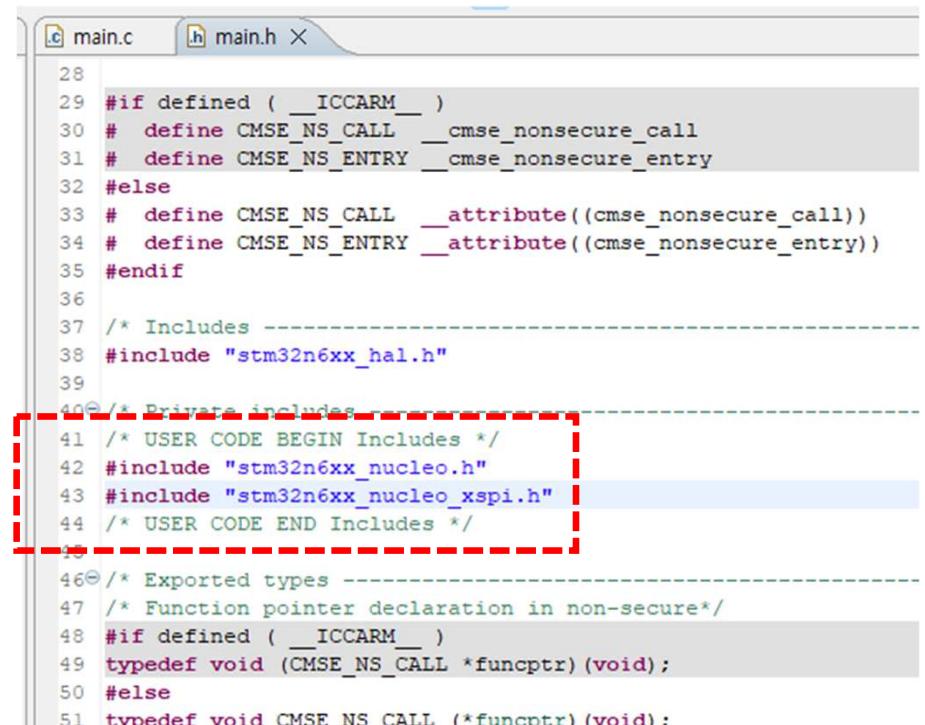
# STM32CubeIDE



```
main.c X
328
329 /* Infrinite loop */
330 /* USER CODE BEGIN WHILE */
331 while (1)
332 {
333     time_in();
334     /* ----- */
335     /* - Inference - */
336     /* ----- */
337
338 #if 1
339     /* Pre-process and fill the input buffer */
340     // _pre_process(buffer_in);
341     /* Perform the inference */
342     LL_ATON_RT_Init_Network(&NN_Instance_Default); // Initialize passed network
343     do {
344         /* Execute first/next step */
345         ll_aton_rt_ret = LL_ATON_RT_RunEpochBlock(&NN_Instance_Default);
346         /* Wait for next event */
347         if (ll_aton_rt_ret == LL_ATON_RT_WFE)
348             LL_ATON_OSAL_WFE();
349     } while (ll_aton_rt_ret != LL_ATON_RT_DONE);
350     /* Post-process the output buffer */
351     /* Invalidate the associated CPU cache region if requested */
352     // _post_process(buffer_out);
353     LL_ATON_RT_DeInit_Network(&NN_Instance_Default);
354     /* ----- */
355     /* - End of Inference - */
356     /* ----- */
357 #else
358     LL_ATON_RT_Main(&NN_Instance_Default);
359 #endif
360
361     duration_dwt = time_out();
362     duration_us = (uint32_t)((float_t)duration_dwt * 1000000.0)/clock_Hz;
363     //printf("Inference: %d us (%d) cycles)\r\n",duration_us, duration_dwt);
364 }
```

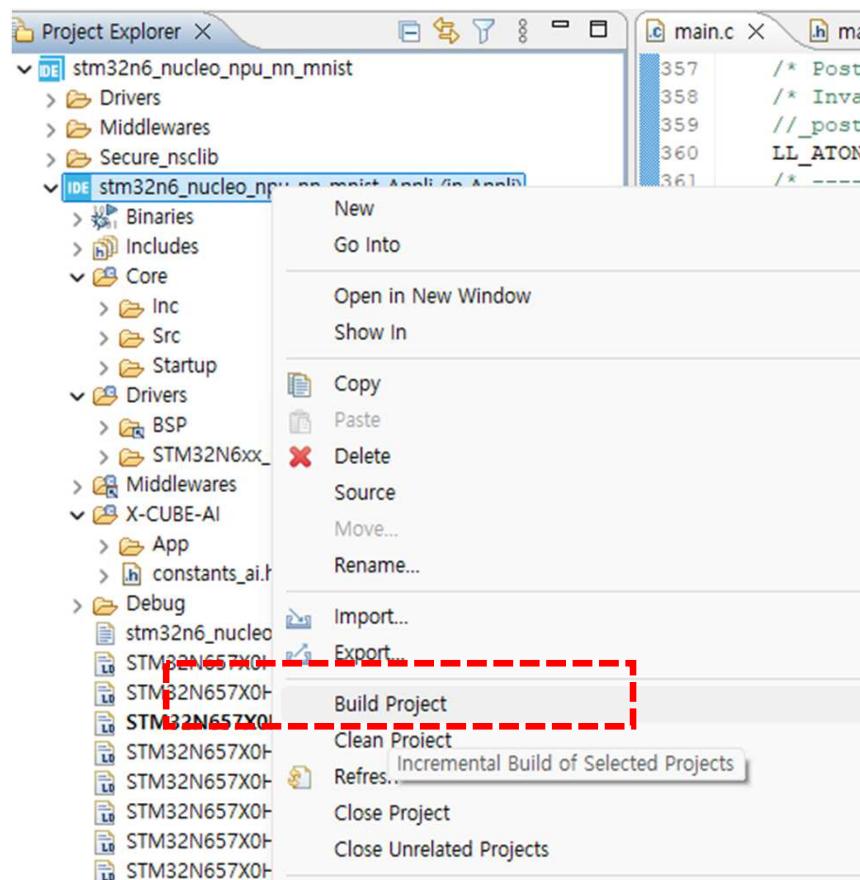


```
#include "stm32n6xx_nucleo.h"
#include "stm32n6xx_nucleo_xspi.h"
```



```
main.c X h main.h X
28
29 #if defined ( __ICCARM__ )
30 # define CMSE_NS_CALL __cmse_nonsecure_call
31 # define CMSE_NS_ENTRY __cmse_nonsecure_entry
32 #else
33 # define CMSE_NS_CALL __attribute__((cmse_nonsecure_call))
34 # define CMSE_NS_ENTRY __attribute__((cmse_nonsecure_entry))
35 #endif
36
37 /* Includes -----
38 #include "stm32n6xx_hal.h"
39
40 /* Private includes -----
41 /* USER CODE BEGIN Includes */
42 #include "stm32n6xx_nucleo.h"
43 #include "stm32n6xx_nucleo_xspi.h"
44 /* USER CODE END Includes */
45
46 /* Exported types -----
47 /* Function pointer declaration in non-secure*/
48 #if defined ( __ICCARM__ )
49 typedef void (CMSE_NS_CALL *funcptr)(void);
50 #else
51 tvbedef void CMSE NS CALL (*funcptr)(void);
52
53 /* Exported functions -----
54 /* Function declarations in non-secure */
55 #if defined ( __ICCARM__ )
56 CMSE_NS_CALL void main(void)
57 {
58     /* Main function body */
59 }
60 #else
61 CMSE_NS_CALL void main(void)
62 {
63     /* Main function body */
64 }
65 #endif
66
67 /* ----- */
68 /* End of main function */
69 }
```

# STM32CubeIDE



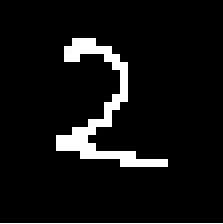
The screenshot shows the STM32CubeIDE Console tab. It displays the build logs for the target 'stm32n6\_nucleo\_npu\_nn\_mnist\_Appli'. The logs show the compilation process using arm-none-eabi-size, objdump, and objcopy tools, followed by the creation of binary and list files. A red dashed box highlights the final message: '20:15:51 Build Finished. 0 errors, 19 warnings. (took 8s.295ms)'.

```
C:/ST/STM32CubeIDE_1.17.0/STM32CubeIDE/plugins/com.st.stm32cube.ide.mcu.external  
Finished building target: stm32n6_nucleo_npu_nn_mnist_Appli.elf  
  
arm-none-eabi-size stm32n6_nucleo_npu_nn_mnist_Appli.elf  
arm-none-eabi-objdump -h -S stm32n6_nucleo_npu_nn_mnist_Appli.elf > "stm32n6_nu  
text data bss dec hex filename  
110052 3552 4744 118348 1ce4c stm32n6_nucleo_npu_nn_mnist_Appli.elf  
arm-none-eabi-objcopy -O binary stm32n6_nucleo_npu_nn_mnist_Appli.elf "stm32n6_n  
Finished building: default.size.stdout  
  
Finished building: stm32n6_nucleo_npu_nn_mnist_Appli.bin  
  
Finished building: stm32n6_nucleo_npu_nn_mnist_Appli.list  
  
20:15:51 Build Finished. 0 errors, 19 warnings. (took 8s.295ms)
```

"Build the project."



# STM32CubeIDE



The screenshot shows the STM32CubeIDE interface during a debugging session. The Project Explorer on the left shows a project named "stm32n6\_nucleo\_npu\_nn\_mnist\_Appli". The main window displays the "main.c" file with code related to AI inference. A breakpoint is set at line 378, where a call to "HAL\_Delay(100);" is located. The Variables tab in the debugger shows the state of the "nn\_buffer\_out" array, which contains 10 float values representing softmax probabilities. The value at index 0 is 1.0, indicating the model has correctly identified the digit '2'. The "Registers" and "Live Exp" tabs are also visible in the debugger.

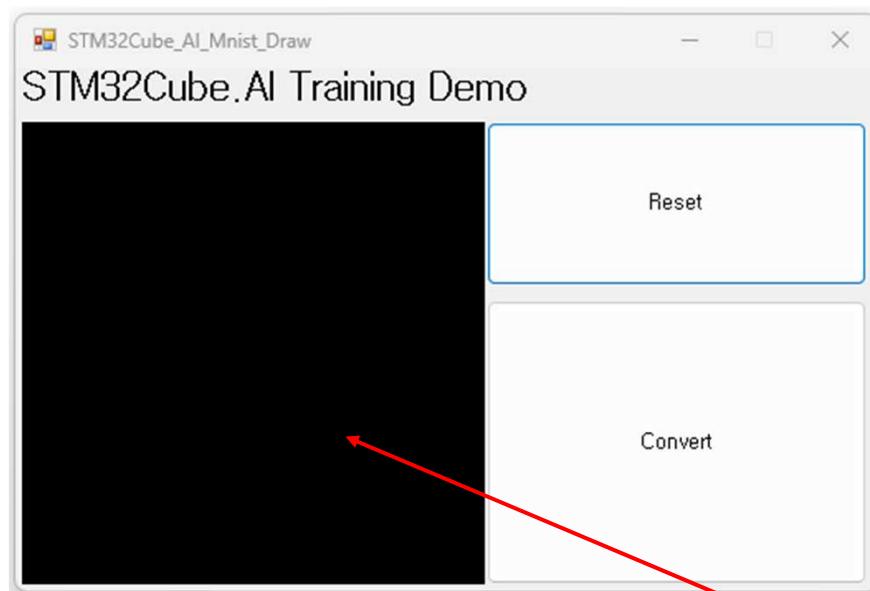
Expression	Type	Value
nn_buffer_out[0]	float	1
nn_buffer_out[1]	float	0
nn_buffer_out[2]	float	0.98828125
nn_buffer_out[3]	float	0.0078125
nn_buffer_out[4]	float	0
nn_buffer_out[5]	float	0
nn_buffer_out[6]	float	0
nn_buffer_out[7]	float	0.00390625
nn_buffer_out[8]	float	0
nn_buffer_out[9]	float	0

"Set a **BreakPoint** and verify the handwritten digit inference result using the **Softmax** values."

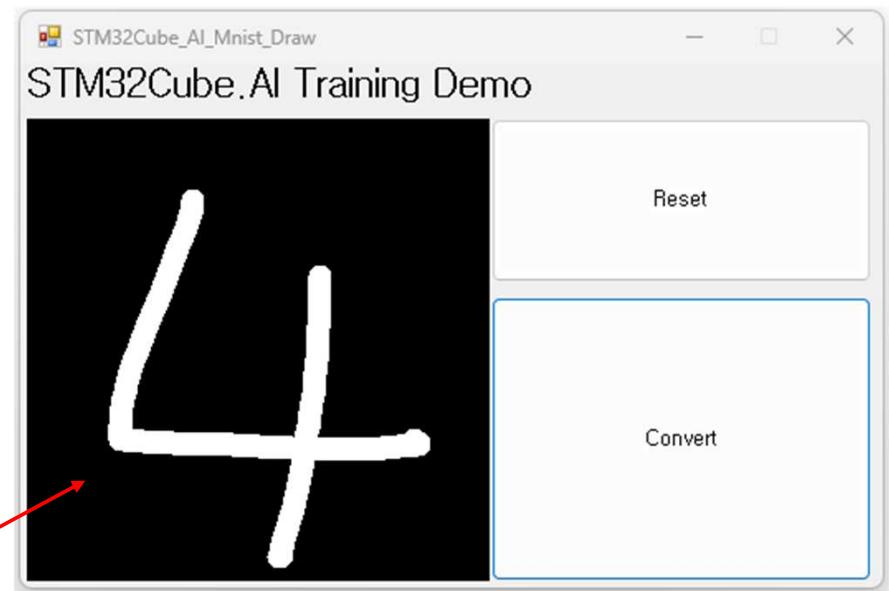


# Input Data Generation

Run to the “STM32Cube\_AI\_Mnist\_example.exe”



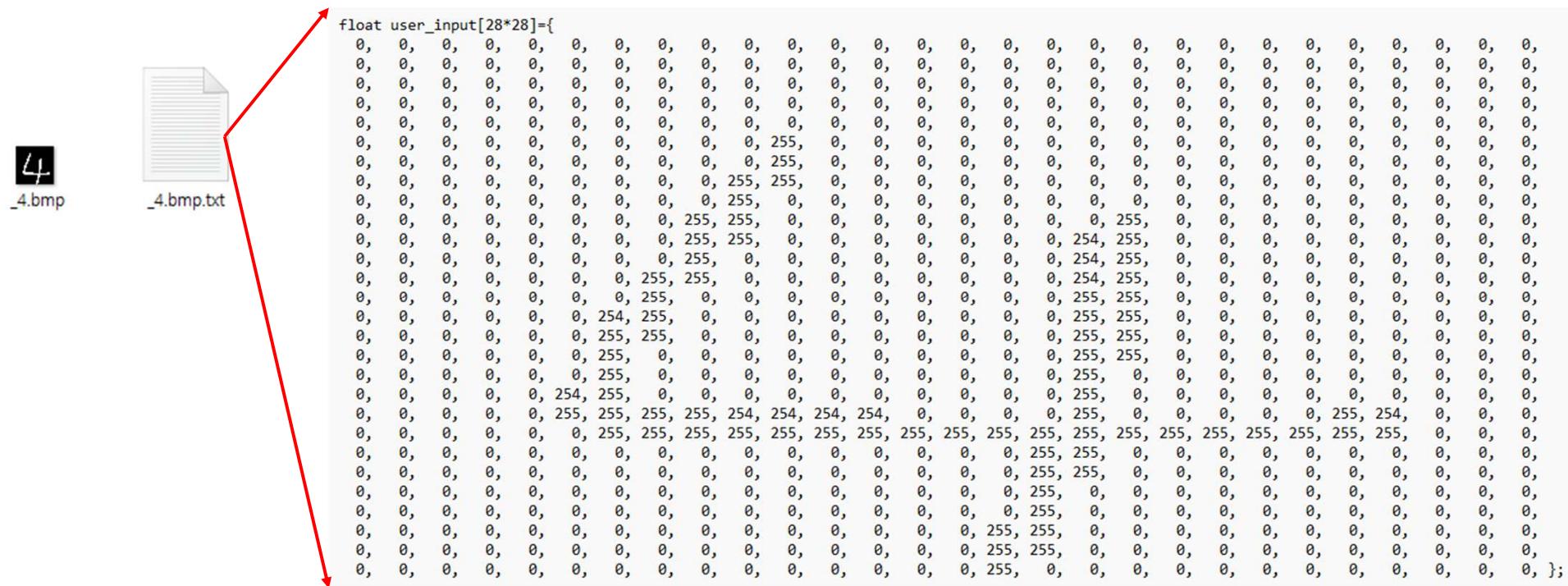
Convert the entered number



Drawing numbers



# Input Data Generation



"You can perform additional tests using the newly generated input data array."

# AI Software ecosystem for STM32N6

AI software ecosystem for STM32N6 with Neural-ART accelerator

The screenshot shows a web browser displaying the ST website at [st.com/en/development-tools/stm32n6-ai.html](https://www.st.com/en/development-tools/stm32n6-ai.html). The page lists several AI software components for the STM32N6 with Neural-ART accelerator, each with a 'Request Software' button.

Part Number	ECCN (US)	Download
n6-ai-getstarted	5D002.a.1	<a href="#">Request Software</a>
n6-ai-h264-uvc	5D002.a.1	<a href="#">Request Software</a>
n6-ai-hand-land	5D002.a.1	<a href="#">Request Software</a>
n6-ai-pdetect	5A002.a.1	<a href="#">Request Software</a>
n6-ai-pose-estim	5D002.a.1	<a href="#">Request Software</a>
n6-ai-pwr-meas	5D002.a.1	<a href="#">Request Software</a>



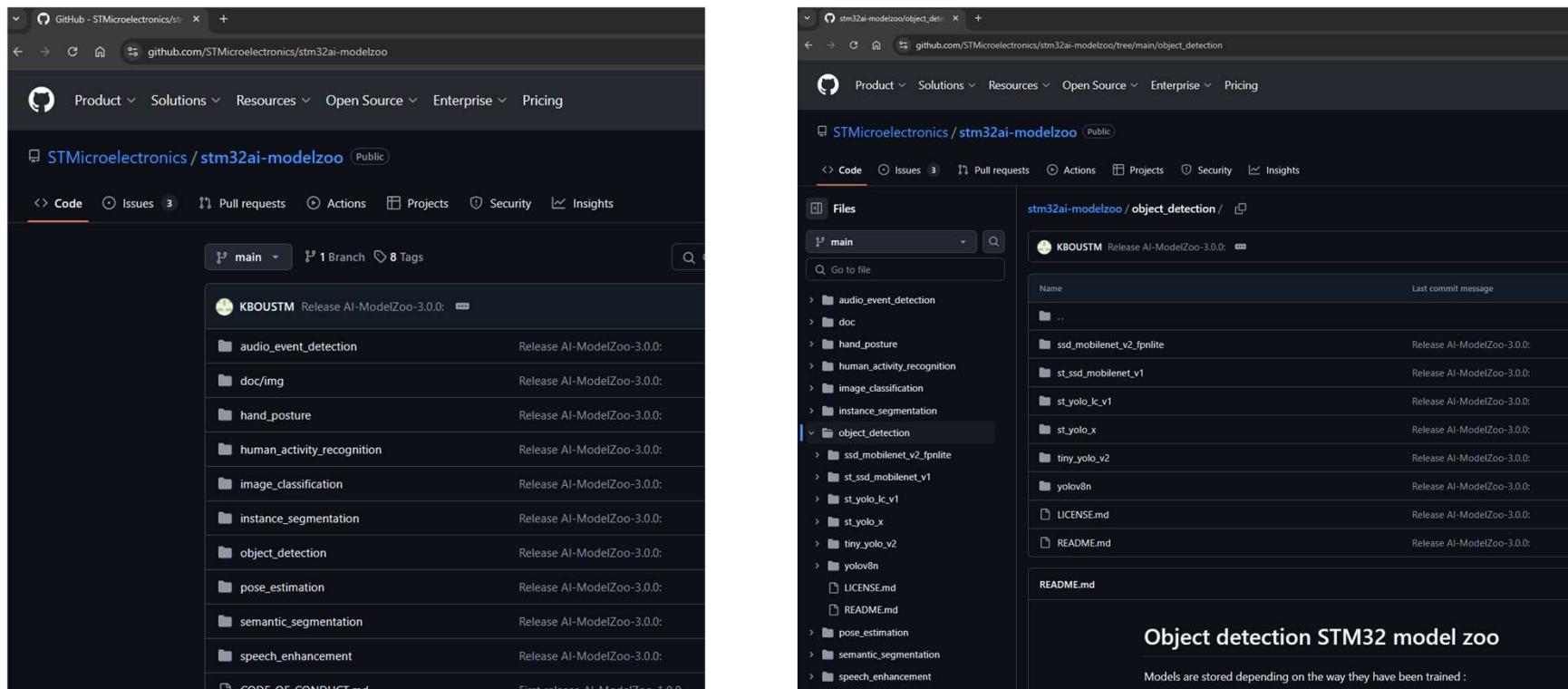
<https://www.st.com/en/development-tools/stm32n6-ai.html>

# **ST Edge AI Developer Cloud and STM32 AI-Model Zoo**



# STM32AI-Model Zoo

The STM32 AI model zoo is a collection of pre-trained machine learning models that are optimized to run on STM32 microcontrollers.

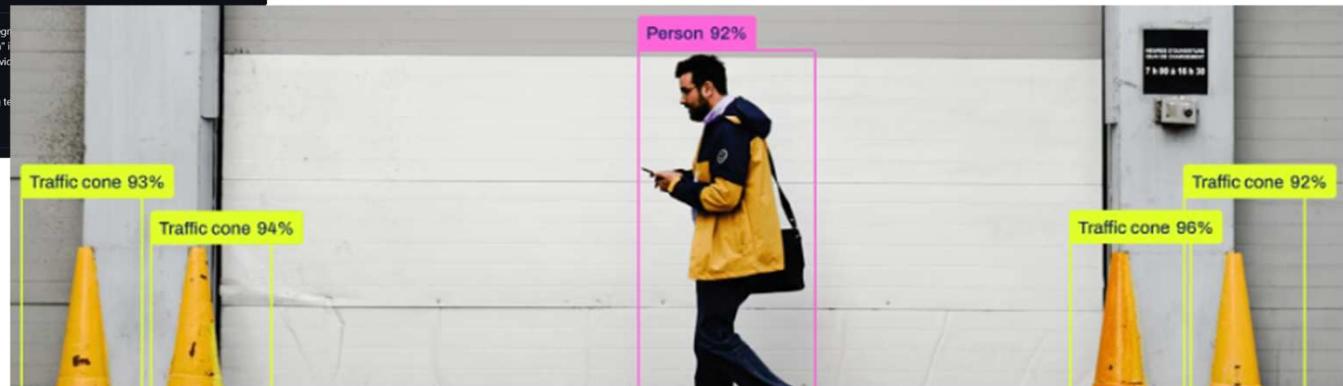


<https://github.com/STMicroelectronics/stm32ai-modelzoo>

# Yolov8n

Yolov8n is a lightweight and efficient object detection model designed for instance segmentation tasks.

The screenshot shows a GitHub repository interface for 'stm32ai-modelzoo/object\_detection'. The 'main' branch is selected. The repository contains several sub-directories and files related to AI models like audio\_event\_detection, doc, hand\_posture, human\_activity\_recognition, image\_classification, instance\_segmentation, object\_detection, ssd\_mobilenet\_v2\_tflite, st\_sd\_mobilenet\_v1, st\_yolo\_lc\_v1, st\_yolo\_x, tiny\_yolo\_v2, and the main yolov8n directory. The yolov8n directory contains LICENSE.md, README.md, and yolov8n\_256\_config.yaml. A README.md file within the directory provides a brief description of the Yolov8n object detection quantized model, mentioning it's a lightweight and efficient model for instance segmentation tasks. It also notes its implementation in Pytorch by Ultralytics and quantization in int8 format.



<https://docs.ultralytics.com/ko/tasks/detect/>

# Yolov8n

Measures are done with default STM32Cube.AI configuration with enabled input / output allocated option.

The screenshot shows the GitHub repository [stm32ai-modelzoo/object\\_detection/yolov8n](https://github.com/STMicroelectronics/stm32ai-modelzoo/tree/main/object_detection/yolov8n). It displays two tables of performance metrics for the YOLOv8n model on the STM32N6 series.

**Reference NPU memory footprint based on COCO Person dataset (see Accuracy for details on dataset)**

Model	Dataset	Format	Resolution	Series	Internal RAM	External RAM	Weights Flash	STM32Cube.AI version	STEdgeAI Core version
YOLOv8n per channel	COCO-Person	Int8	192x192x3	STM32N6	697.5	0.0	2965.61	10.0.0	2.0.0
YOLOv8n per channel	COCO-Person	Int8	256x256x3	STM32N6	1626	0.0	2970.13	10.0.0	2.0.0
YOLOv8n per channel	COCO-Person	Int8	320x320x3	STM32N6	2162.5	0.0	2975.99	10.0.0	2.0.0
YOLOv8n per channel	COCO-Person	Int8	416x416x3	STM32N6	2704	0.0	2987.52	10.0.0	2.0.0

**Reference NPU inference time based on COCO Person dataset (see Accuracy for details on dataset)**

Model	Dataset	Format	Resolution	Board	Execution Engine	Inference time (ms)	Inf / sec	STM32Cube.AI version	STEdgeAI Core version
YOLOv8n per channel	COCO-Person	Int8	192x192x3	STM32N6570-DK	NPU/MCU	18.91	52.89	10.0.0	2.0.0
YOLOv8n per channel	COCO-Person	Int8	256x256x3	STM32N6570-DK	NPU/MCU	28.6	34.97	10.0.0	2.0.0
YOLOv8n per channel	COCO-Person	Int8	320x320x3	STM32N6570-DK	NPU/MCU	38.32	26.09	10.0.0	2.0.0
YOLOv8n per channel	COCO-Person	Int8	416x416x3	STM32N6570-DK	NPU/MCU	63.03	15.86	10.0.0	2.0.0

The screenshot shows the GitHub repository [stm32-hotspot/ultralytics](https://github.com/stm32-hotspot/ultralytics). It displays the commit history for the `ultralytics` branch.

This branch is 8 commits ahead of `ultralytics/ultralytics:main`.

- MCHSTM add per tensor yolov8 person models (342ac37 · 4 days ago) 1,110 Commits
- .github Run --slow tests for Raspberry Pi (ultralytics#10132) (11 months ago)
- docker ultralytics 8.2.1 adopt v8.0 GitHub assets URL (ultralytics#... (11 months ago)
- docs ultralytics 8.2.2 replace COCO128 with COCO8 (ultralytics#... (11 months ago)
- examples add per tensor yolov8 person models (4 days ago)
- tests Run --slow tests for Raspberry Pi (ultralytics#10132) (11 months ago)
- ultralytics Normalize yolov8 pose output (9 months ago)
- .gitignore STEdgeAI: introduce STEdgeAI object detection models (9 months ago)
- .pre-commit-config.yaml YAML reformat (ultralytics#7669) (last year)
- CITATION.cff Update CITATION.cff with ccifit corrections (ultralytics#8658) (last year)
- CONTRIBUTING.md ultralytics 8.0.224 Counting and Heatmaps updates (ultral... (2 years ago)
- LICENSE Update LICENSE to AGPL-3.0 (ultralytics#2031) (2 years ago)

<https://github.com/stm32-hotspot/ultralytics>

[https://github.com/STMicroelectronics/stm32ai-modelzoo/tree/main/object\\_detection/yolov8n](https://github.com/STMicroelectronics/stm32ai-modelzoo/tree/main/object_detection/yolov8n)



# ST Edge AI Developer Cloud

ST Edge AI Developer Cloud is a free online service for developing AI on ST devices, offering tools for creation, optimization, and benchmarking.

The screenshot shows the STM32AI website with a dark blue header. The top navigation bar includes links for STM32 AI Solutions, Case studies, Products, Model zoo, Docs, ST Community, and English. Below the header, a large button with a right-pointing arrow and the text "► ST Edge AI Developer Cloud (online)" is prominently displayed. The main content area features a section titled "Online tool for fast AI optimization and benchmark".

## Online tool for fast AI optimization and benchmark

ST Edge AI Developer Cloud is a free online service for developing AI on ST devices, offering tools for creation, optimization, and benchmarking. It leverages ST Edge AI Core technology and supports AI hardware acceleration (NPU) on compatible devices.

This tool integrates what was previously available in STM32Cube.AI Developer Cloud.

[Open online platform](#)

The screenshot shows the ST Edge AI Developer Cloud home page with a dark blue header. The top navigation bar includes links for Home, Case studies, Products, Model zoo, Docs, ST Community, and English. The main content area features a section titled "OPTIMIZE YOUR TRAINED NEURAL NETWORK" with a sub-section "Optimize and measure performance of your Artificial Intelligence library for STMicroelectronics microcontrollers, microprocessors and smart-sensors". A "START NOW" button is visible. The page also includes sections for "ST Edge AI model zoo", "Bring Your Own Model", "Optimize and benchmark your NN model", and "Generate code and download".



<https://stm32ai.st.com/st-edge-ai-developer-cloud/>

# ST Edge AI Developer Cloud

To use the ST Edge AI Developer Cloud, you need to log in to [www.st.com](http://www.st.com).

The image shows two browser windows side-by-side. The left window displays the ST login page at [my.st.com/cas/login?service=https://my.st.com/cas-idpwebssologin%3Fresume%3D%2Fas%2F](https://my.st.com/cas/login?service=https://my.st.com/cas-idpwebssologin%3Fresume%3D%2Fas%2F). It features an ST logo and fields for 'E-mail address' and 'Password', along with a 'Remember me on this computer.' checkbox and a 'Login' button. The right window shows the ST Edge AI Developer Cloud home page at [stedgeai-dc.st.com/home](https://stedgeai-dc.st.com/home). It has an ST logo and a title 'OPTIMIZE YOUR TRAINED NEURAL NETWORK'. Below it, a sub-section titled 'Your model library' includes a 'New model' button and a file upload area. To the right, a 'Pick a model from ST Model Zoo' section lists several neural network models with details like target, use case, and import buttons.

**OPTIMIZE YOUR TRAINED NEURAL NETWORK**

Optimize and measure performance of your Artificial Intelligence library for STMicroelectronics microcontrollers, microprocessors and smart-sensors  
This free online tool allows you to generate and test optimized AI libraries based on your trained Neural Networks

**Your model library**

New model

Drop your model or click here to open a file browser. Supported models are Keras, ONNX and TFLite (.h5, .hdf5, .onnx, .tflite)  
Max file size: 256 MiB

Show previously quantized models

Date  Name  Size

**Pick a model from ST Model Zoo**

Target	Use case	Neural Network
2 selected	9 selected	61 selected

**CNN2D\_ST\_HandPosture\_8classes\_hand\_posture\_ST\_VL53L5CX\_handpostur...**

Description: CNN2D\_ST\_HandPosture 8 postures trained on ST\_VL53L5CX\_handposture\_dataset with 8 classes  
Use case: hand\_posture  
Dataset used: ST\_VL53L5CX\_handposture\_dataset

**CNN2D\_ST\_HandPosture\_8classes\_hand\_posture\_ST\_VL53L8CX\_handpostur...**

Description: CNN2D\_ST\_HandPosture 8 postures trained on ST\_VL53L8CX\_handposture\_dataset with 8 classes  
Use case: hand\_posture  
Dataset used: ST\_VL53L8CX\_handposture\_dataset

**cnn\_8x8\_ispu\_wand\_lsm6ds016is\_ISPU\_reference\_dataset.h5**

Description: CNN1D for air writing recognition that detects the letters I, S, P, and U  
Use case: ispu\_wand  
Dataset used: lsm6ds016is\_ISPU\_reference\_dataset



# ST Edge AI Developer Cloud

The screenshot shows the 'OPTIMIZE YOUR TRAINED NEURAL NETWORK' page. On the left, there's a section for 'Your model library' with a 'New model' button and a file upload area. On the right, there's a 'Pick a model from ST Model Zoo' section. This section includes dropdown menus for 'Target' (2 selected), 'Use case' (1 selected), and 'Neural Network' (17 selected). It lists three pre-trained models:

- st\_yolo\_x\_nano\_416\_0\_33\_0\_25\_int8\_object\_detection\_COCO\_2017\_Person.tflite
- tiny\_yolo\_v2\_224\_int8\_object\_detection\_ST\_Person.tflite
- tiny\_yolo\_v2\_416\_int8\_object\_detection\_COCO\_2017\_Person.tflite

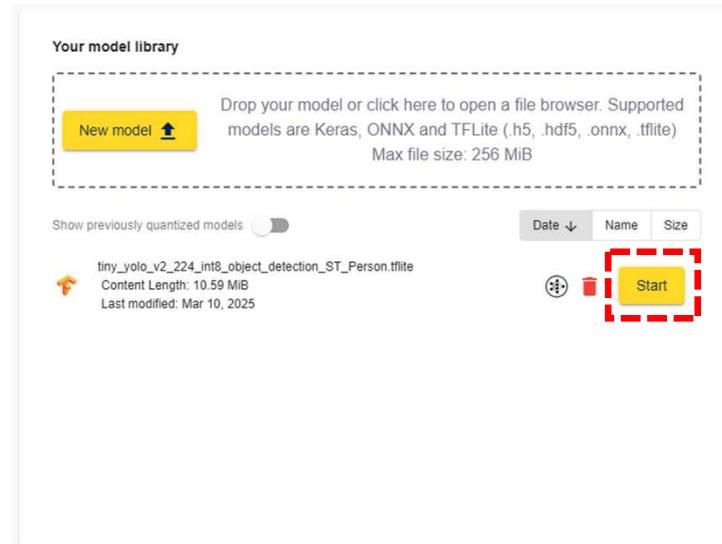
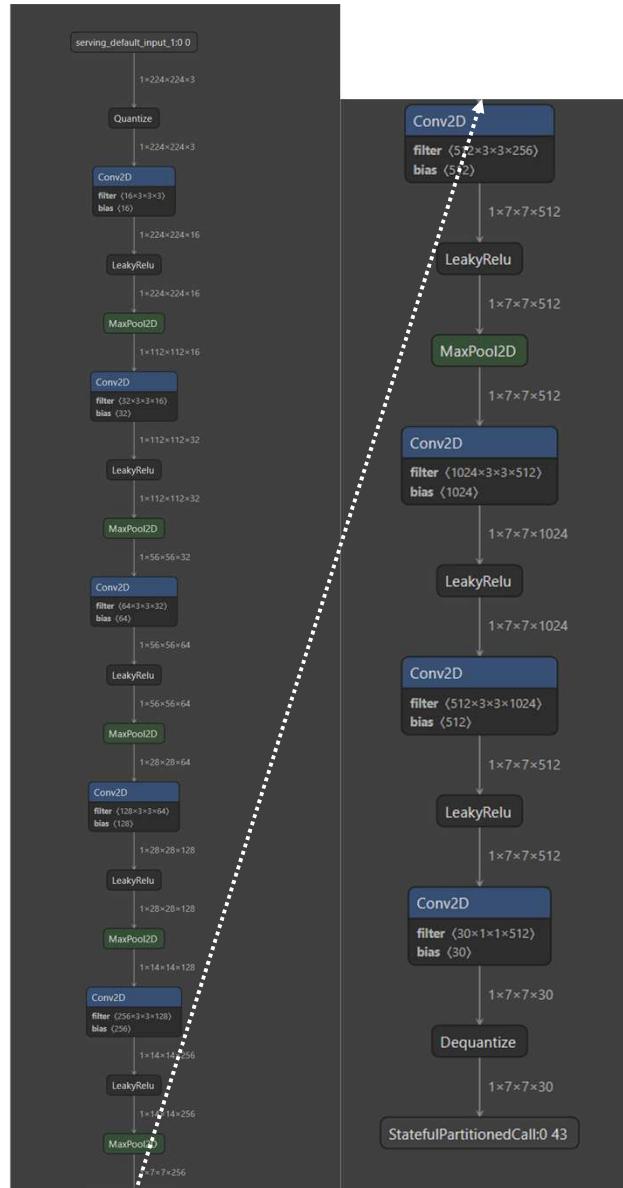
Each entry has an 'Import' button. A red dashed box highlights the second item, 'tiny\_yolo\_v2\_224\_int8\_object\_detection\_ST\_Person.tflite'.

The screenshot shows the 'Your model library' page. It features a 'New model' button and a file upload area. Below it, a list of imported models is shown, starting with 'tiny\_yolo\_v2\_224\_int8\_object\_detection\_ST\_Person.tflite'. To the right of the list are icons for a circular progress bar, a trash can, and a yellow 'Start' button.

You can select a pre-trained NN model from the ST Model Zoo or load your own pre-trained NN model.



# ST Edge AI Developer Cloud



You can also visualize the loaded pre-trained NN model in a graph format.

# Select a platform

The screenshot shows the ST Edge AI Platform interface. At the top, there is a navigation bar with tabs: Quantize (Optional), Optimize, Benchmark, Results, and Generate. The 'Quantize' tab is currently selected. Below the navigation bar, a red dashed box highlights the first tab, 'Select a platform'. The main content area has a title 'Select a platform' and a subtitle 'Select which platform is most suitable for your use-case'. A dropdown menu shows 'Model currently selected: TINY\_YOLO\_V2\_224\_INT8\_OBJECT\_DETECTION\_ST\_PERSON.TFLITE'. Below the model selection, there are three tabs: INPUT (224x224x3 (8 bits)), OUTPUT (7x7x30 (32 bits)), and MODEL TYPE (STAI\_FORMAT\_S32). Another red dashed box highlights the 'INPUT' tab. A 'Show Graph' button is also present. In the bottom section, there are three categories: 'STM32 MCUs' (with two green boards shown), 'STM32 MCU with Neural-ART™' (with a board featuring a screen shown), and 'STM32 MPUs' (with a blue board shown). Each category has a 'Select' button at the bottom. The 'STM32 MCU with Neural-ART™' category is highlighted with a red dashed box.

Model currently selected  
TINY\_YOLO\_V2\_224\_INT8\_OBJECT\_DETECTION\_ST\_PERSON.TFLITE

INPUT  
224x224x3 (8 bits)

OUTPUT  
7x7x30 (32 bits)

MODEL TYPE  
STAI\_FORMAT\_S32

Select another model

Show Graph

Select ST Edge AI Core Version  
ST Edge AI Core 2.0.0

**STM32 MCUs**  
STM32  
Microcontroller units  
Tool version: 10.0.0

Start with General Purpose STM32 Discovery Kits and Nucleos

Select

**STM32 MCU with Neural-ART™**  
STM32  
Microcontrollers embedding Neural Processing Unit (NPU)  
Tool version: 10.0.0

Start with STM32 MCUs including Neural-ART™ to accelerate your AI applications

Select

**STM32 MPUs**  
STM32  
Microprocessors Units (MPU)  
Tool version: 2.0.0

Start with STM32 Microprocessors embedding Cortex-A loaded with X-LINUX-AI

Select

In the **Select a platform** tab, select **STM32 MCU with Neural-ART**.



# Quantize

The screenshot shows the ST Model Optimizer workflow interface. The top navigation bar includes steps: Select a platform, Quantize (highlighted with a red dashed box), Optimize, Benchmark, Results, and Generate.

**Model quantization**  
Reduce the computational and memory costs of your neural network

**Model currently selected:** CNN2D\_ST\_HANDPOSTURE\_8CLASSES\_HAND\_POSTURE\_ST\_VL53L8CX\_HANDPOSTURE\_DATASET.H5

**INPUT:** 8x8x2 (32 bits)    **OUTPUT:** 8 (32 bits)    **MODEL TYPE:** STAI\_FORMAT\_FLOAT

[Select another model](#)    [Show Graph](#)    [Go next >](#)

**Apply post-training quantization**  
Powered by TensorFlow Lite Converter

For better performance on STM32 NPU, we recommend a per-channel quantization.

**Input Type:**  int8     uint8     float32

**Output Type:**  int8     uint8     float32

Disable per channel quantization

Load a dataset to check the accuracy obtained after quantization  
[Load file \(.npz\)](#)    [Show terminal](#)    [Launch quantization](#)

If no quantization file is provided, quantization will occur with random values



# Optimize

The screenshot shows the 'Model optimization' step in a workflow, highlighted with a red dashed box. The workflow steps are: Select a platform, Quantize Optional, Optimize, Benchmark, Results, and Generate.

**Model optimization**  
Use the different options to optimize your neural network

**Model currently selected:** TINY\_YOLO\_V2\_224\_INT8\_OBJECT\_DETECTION\_ST\_PERSON.TFLITE

**INPUT:** 224x224x3 (8 bits)    **OUTPUT:** 7x7x30 (32 bits)    **MODEL TYPE:** STAI\_FORMAT\_S32

**Select another model** | **Show Graph**

**Select your model optimization options**

Optimization	Memory pool settings	Epoch controller
<input checked="" type="radio"/> Default Configuration	<input checked="" type="radio"/> Internal and external memories (1MB reserved for code)	<input checked="" type="checkbox"/> Enable epoch controller (--enable-epoch-controller)
<input type="radio"/> Automatic Configuration	<input type="radio"/> Internal memories	
	<input type="radio"/> Manual	

**Extra Command Line Interface (CLI) Arguments**

**Optimize ▶**



# Optimize

Select your model optimization options

Optimization      Memory pool settings      Epoch controller

Default Configuration     Internal and external memories (1MB reserved for code)     Enable epoch controller (-enable-epoch-controller)

Automatic Configuration     Internal memories     Manual

Extra Command Line Interface (CLI) Arguments

Optimize ➔

History of optimization results (inference time will be measured in benchmark step)

Show terminal

Date	Version	Platform	Optimization	Allocate Inputs	Allocate Outputs	MACC	Flash size	RAM size
3/10/25, 1:47 PM <span style="color: orange;">Latest</span>	2.0.0-20049	STM32 NPU	-	-	-	0 ↘	11143097 ↗ Total: ≈10.6 MiB (+0.454%) Weights: 10.55 MiB Library (estimation) ≈75 KiB	507653 ↗ Total: ≈496 KiB (+120%) Activations: 343 KiB Library (estimation) ≈13 B Input/Output: 147 KiB / 5.74 KiB
3/10/25, 12:34 PM <span style="color: blue;">Default</span>	2.0.0-20049	STM32 MCU	balanced	true	true	777482634 → 11092691 →	231168 →	

Show graph   Delete

Compare with default   Go to benchmark

Items per page: 5   1 – 2 of 2   | < < > > |

Terminal

```
>>> stedgeai analyze --model tiny_yolo_v2_224_int8_object_detection_ST_Person.tflite --st-neural-art custom@/tmp/stm32ai_service/21fd4861-d57c-4da7-b377-93a516f72731/output/tiny_yolo_v2_224_int8_object_detection_ST_Person_OE_3_1_0.onnx" --json-q
ST Edge AI Core v2.0.0-20049
>>> EXECUTING NEURAL ART COMPILER
atonn -i "/tmp/stm32ai_service/21fd4861-d57c-4da7-b377-93a516f72731/output/tiny_yolo_v2_224_int8_object_detection_ST_Person_OE_3_1_0.onnx" --json-q
<<< DONE EXECUTING NEURAL ART COMPILER
>>> EXECUTING EPOCH_CONTROLLER POST_PROCESSING
<<< DONE EXECUTING EPOCH_CONTROLLER POST_PROCESSING
Exec/report summary (analyze)
```



# Benchmark

The screenshot shows the workflow steps: Select a platform, Quantize Optional, Optimize, **Benchmark**, Results, and Generate. The 'Benchmark' step is highlighted with a red dashed box.

**Model benchmarking**  
Run your model on different boards

**Model currently selected:** TINY\_YOLO\_V2\_224\_INT8\_OBJECT\_DETECTION\_ST\_PERSON.TFLITE

**Current parameters:**

- CURRENT PLATFORM: STM32 NPU
- VERSION: ST Edge AI Core 2.0.0
- ST -NEURAL -ART: Default Configuration

Buttons: Change parameters, Change platform, Go next >

**Schedule a benchmark:**

STM32N6570-DK Neural-ART  
Arm Cortex-M55 + ST Neural-ART Accelerator | 800 MHz

Buttons: Start Benchmark ▶, Settings, File, More, Measured inference time - ms



# Benchmark

Schedule a benchmark

STM32N6570-DK Neural-ART  
Arm Cortex-M55 + ST Neural-ART Accelerator | 800 MHz

Measured inference time - ms

Schedule a benchmark

STM32N6570-DK Neural-ART  
Arm Cortex-M55 + ST Neural-ART Accelerator | 800 MHz

Measured inference time 30.57 ms

Results of current benchmarks session

Platform	MACC	Duration (ms)	Weights Size	Activation Size	Flash size	RAM size	Device	Clock	Use External Flash	Use External RAM
STM32 NPU	0	30.57	10.55 MiB	343 KiB	10.77 MiB	343.01 KiB	STM32N6570-DK	800MHz	undefined	undefined

Show performance summary

Export as CSV

Delete

Items per page: 5 | 1 – 1 of 1 | < < > >|

**ST**

# Result

The screenshot shows the workflow interface of the ST Neural-Accelerator tool. At the top, a progress bar indicates five steps: 'Select a platform' (green checkmark), 'Quantize Optional' (green checkmark), 'Optimize' (green checkmark), 'Benchmark' (green checkmark), and 'Results' (highlighted with a red dashed box). Below the progress bar, the title 'Benchmarking results' is displayed, followed by the sub-instruction: 'Here are all the benchmark results for STMicroelectronics boards and smart-sensors'.

**Model currently selected:** TINY\_YOLO\_V2\_224\_INT8\_OBJECT\_DETECTION\_ST\_PERSON.TFLITE

**Current parameters:**

- CURRENT PLATFORM: STM32 NPU
- VERSION: ST Edge AI Core 2.0.0
- ST-NEURAL -ART: Default Configuration

Buttons for navigation and configuration include: 'Select another model', 'Show Graph', 'Change parameters', 'Change platform', and 'Go next >'.

**History of benchmark results:**

Date	Model Name	Model Type	Version	Optimization	Use Neural-Art	Platform	MACC	Duration (ms)	Weights Size	Activation Size
3/10/2025, 1:49:17 PM	tiny_yolo_v2_224_int8_object_detection_ST_Person.tflite	tflite	2.0.0-20049	-	default	STM32 NPU	0	30.57	10.55 MiB	343 KiB
2/27/2025, 8:35:41 PM	model_quantized1.tflite	tflite	2.0.0-20049	-	default	STM32 NPU	0	0.02890	577 B	1.25 KiB

Buttons for managing the results table include: 'Show performance summary', 'Export as CSV', 'Refresh', 'Delete', and a three-dot menu. A red dashed box highlights the first two rows of the results table.

Page navigation controls at the bottom include: 'Items per page: 5', '1 – 2 of 2', and arrows for navigating through the pages.



# Generate

Generate your project  
Generate all what you need to start your project

Select a platform   Quantize Optional   Optimize   Benchmark   Results   **Generate**

**Model currently selected**  
TINY\_YOLO\_V2\_224\_INT8\_OBJECT\_DETECTION\_ST\_PERSON.TFLITE

**INPUT** 224x224x3 (8 bits)   **OUTPUT** 7x7x30 (32 bits)   **MODEL TYPE** STAI\_FORMAT\_S32

[Select another model](#)   [Show Graph](#)

**Current parameters**

**CURRENT PLATFORM** STM32 NPU   **VERSION** ST Edge AI Core 2.0.0   **ST -NEURAL -ART** Default Configuration

[Change parameters](#)   [Change platform](#)

**CPU/Series filters**  
Arm Cortex-M55 + ST Neural-ART Accelerator   Ex: STM32F4

**Boards**  
STM32N6570-DK

**STM32N6570-DK**  
Arm Cortex-M55 + ST Neural-ART Accelerator |  
STM32N6 Series | 800 MHz  
RAM: 0 KB (Max: 1024 KB)  
External RAM: 32 MB  
External Flash: 64 MB

**STM32N6**

**Download C Code**

Generates optimized C Code associated with your pre-trained Neural Network

**Download**



# 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.

