

STM32 Expert Training – Edge AI based on STM32N6 v1.2

ST KOREA
Blaine MOON

Agenda

STM32N6 Overview

STM32N6 Architecture & Features

NUCLEO-N657Z0-Q

Hands-On: FSBL + LRUN

Hands-On: FSBL + LRUN

Convolution Neural Network

ST Neural-ART Accelerator™

Hands-On AI: Recognizing Handwritten Digits with MNIST



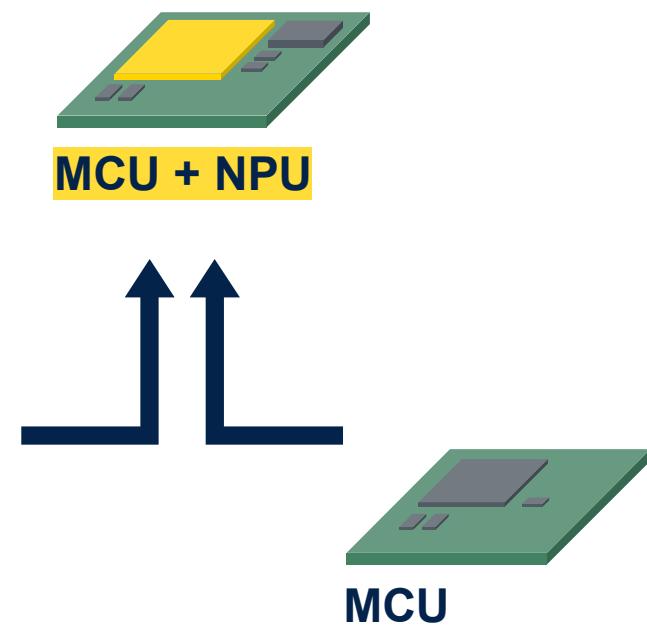
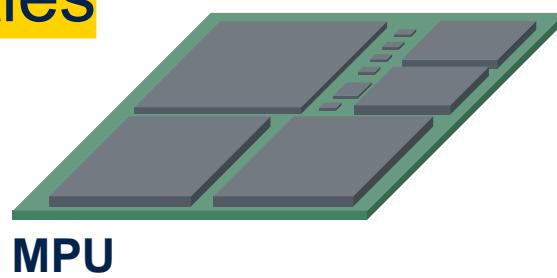


The first high-performance STM32 MCU with AI acceleration

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



Small footprint

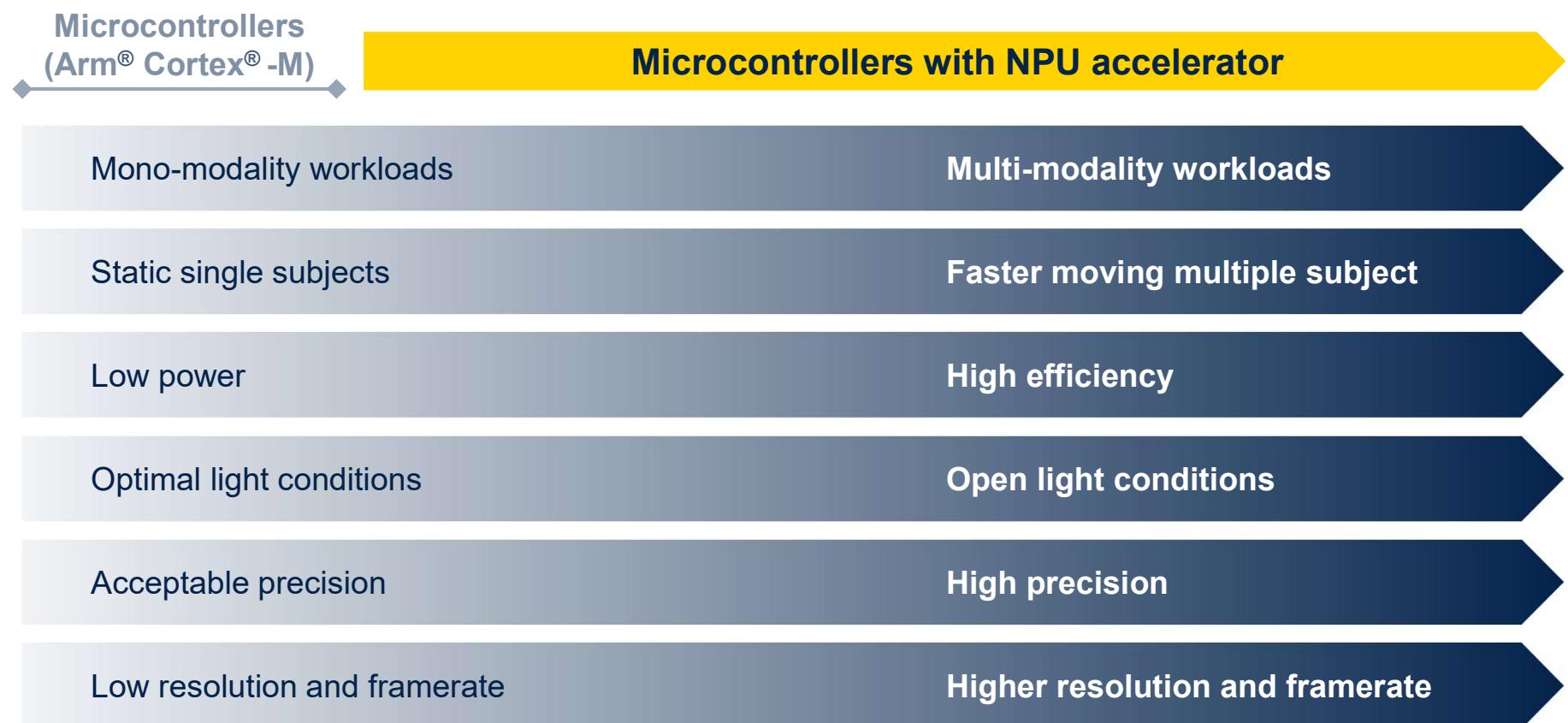
Lower power

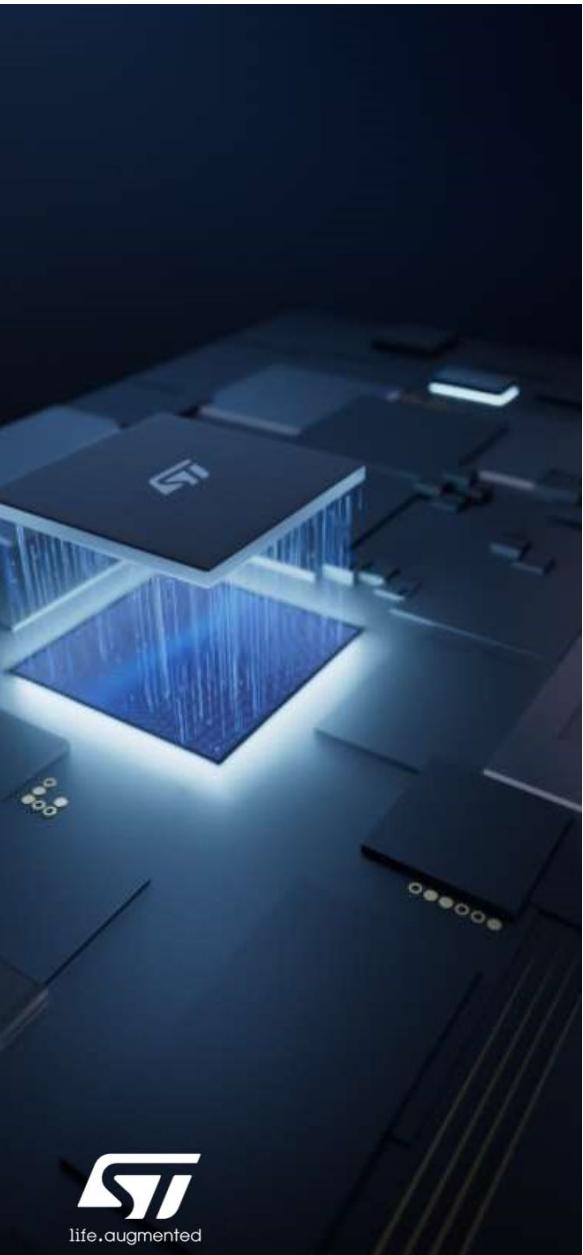
Lower cost

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.

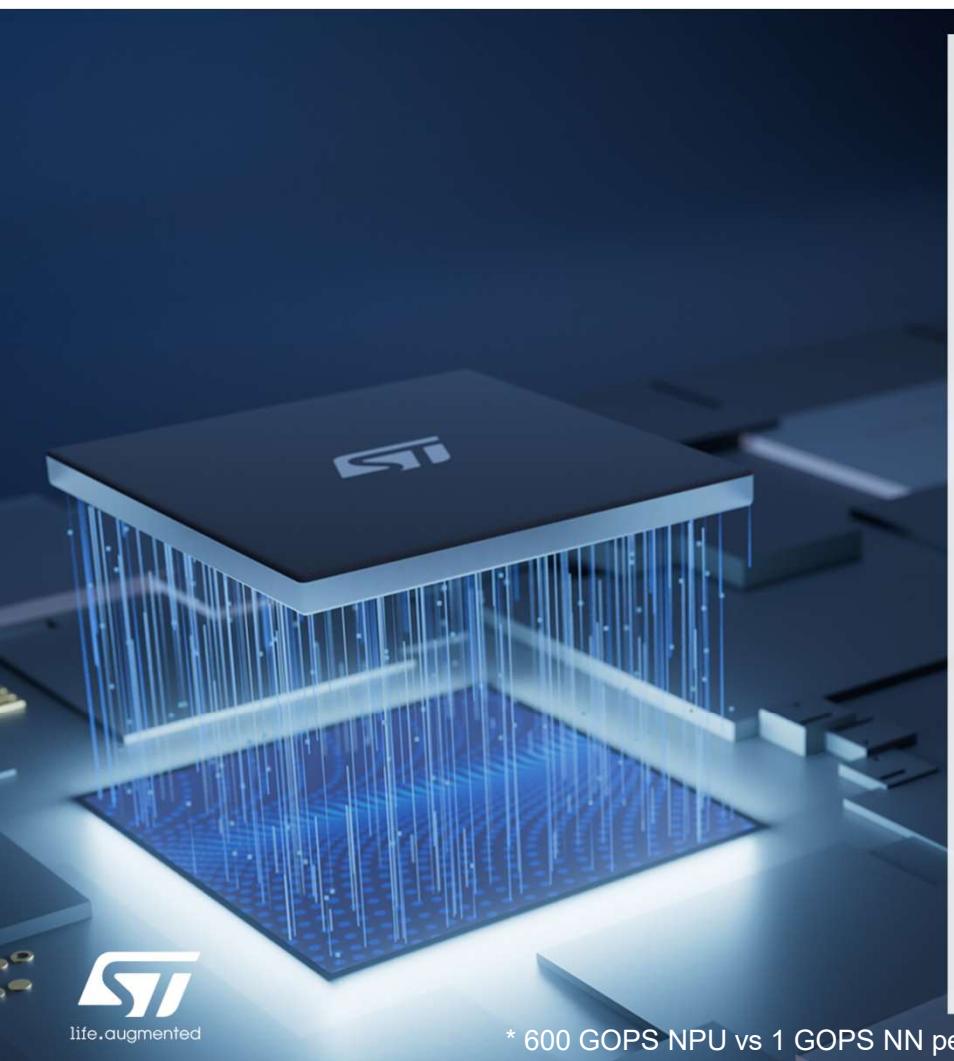
Elevating graphics & multimedia experiences

- 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



Improved accuracy
Adapt to local environment



Reduced data transmission
Generate meaningful
information



Privacy & security
No raw data sent to the cloud



Sustainable on energy
Low-power consumption



Advanced experience
Personalized features



The most powerful STM32 series ever made



Two lines sharing the same DNA



Artificial intelligence
line

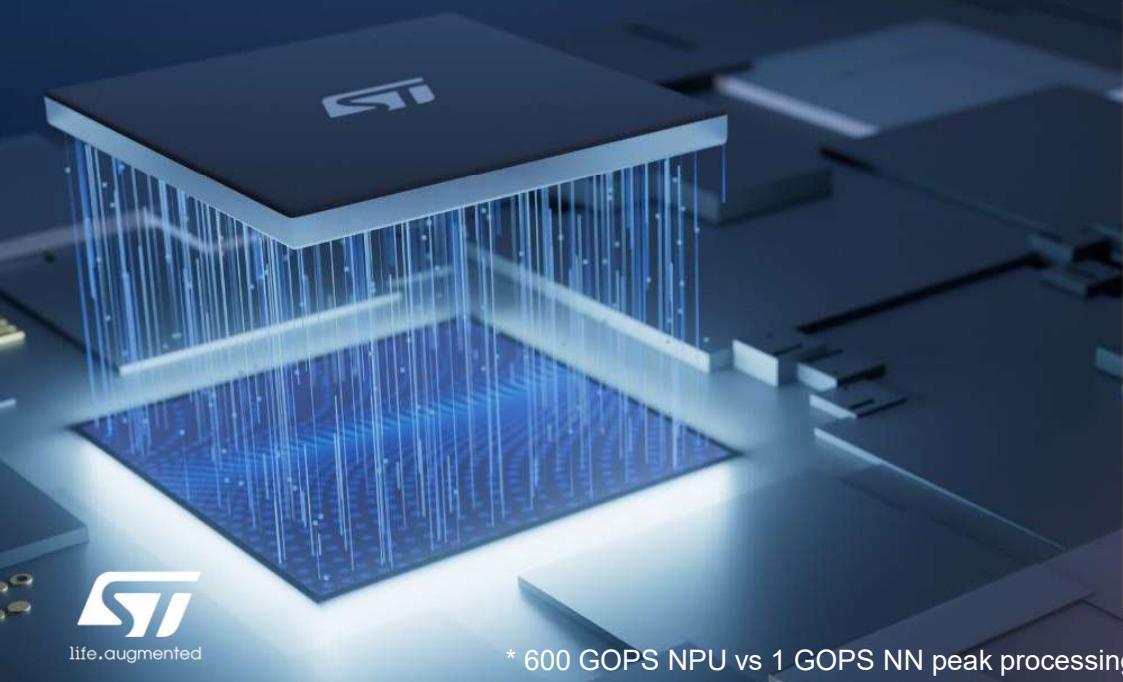
STM32N6x7 MCUs

General-purpose
line

STM32N6x5 MCUs

ST Neural-ART Accelerator

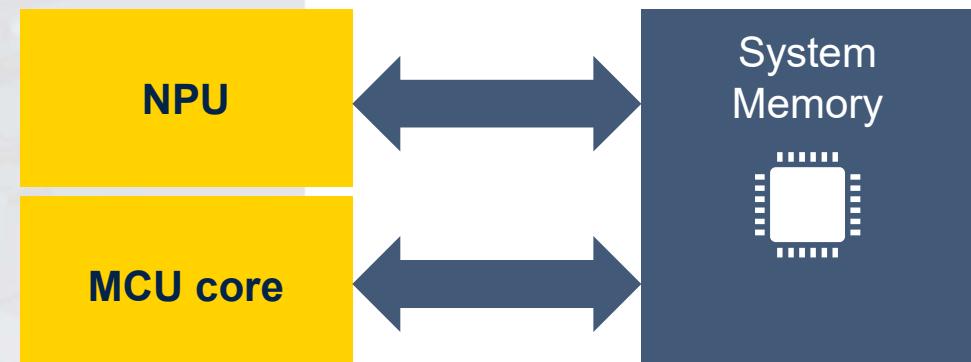
600x
ML performance uplift*



Dedicated embedded neural processing unit

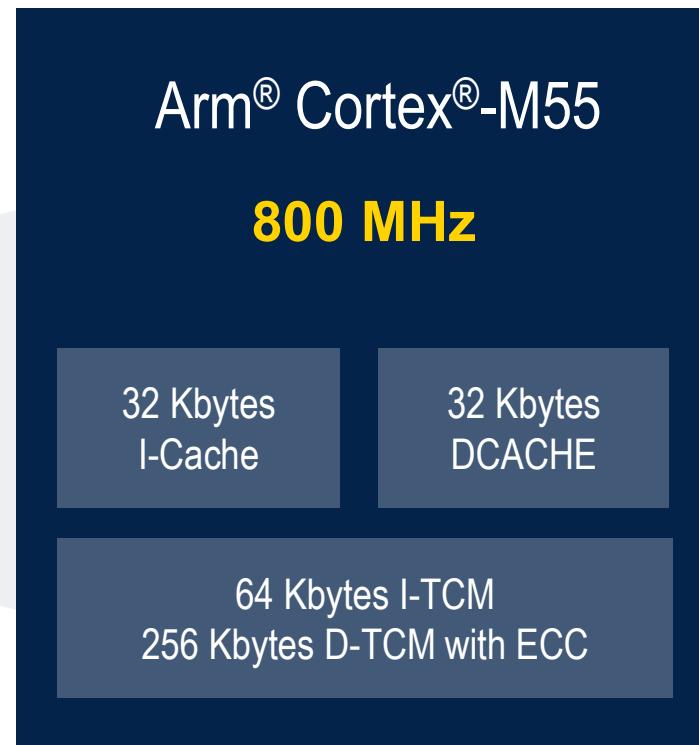
- 600 GOPS
- 3 TOPS/W power consumption
- Cache memory to optimize external memory access

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	Octo-SPI	FMC
Up to 800 Mbytes/s	Up to 400 Mbytes/s	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

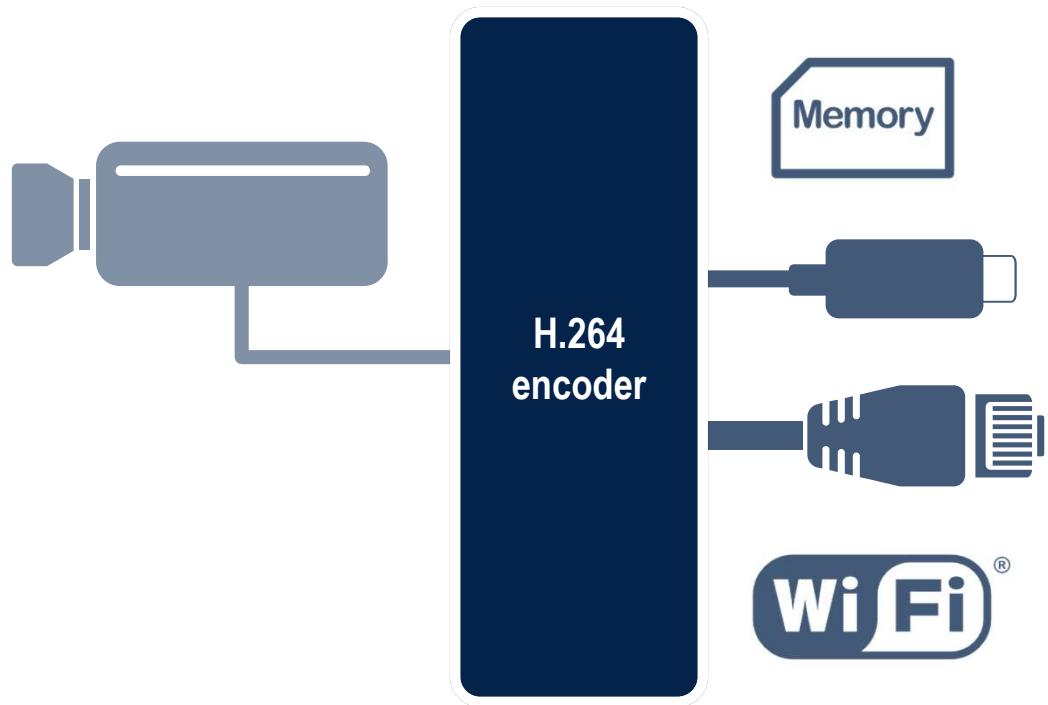


Elevating multimedia experiences

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



Embedded firmware
2A algorithms

Image signal processor (ISP)

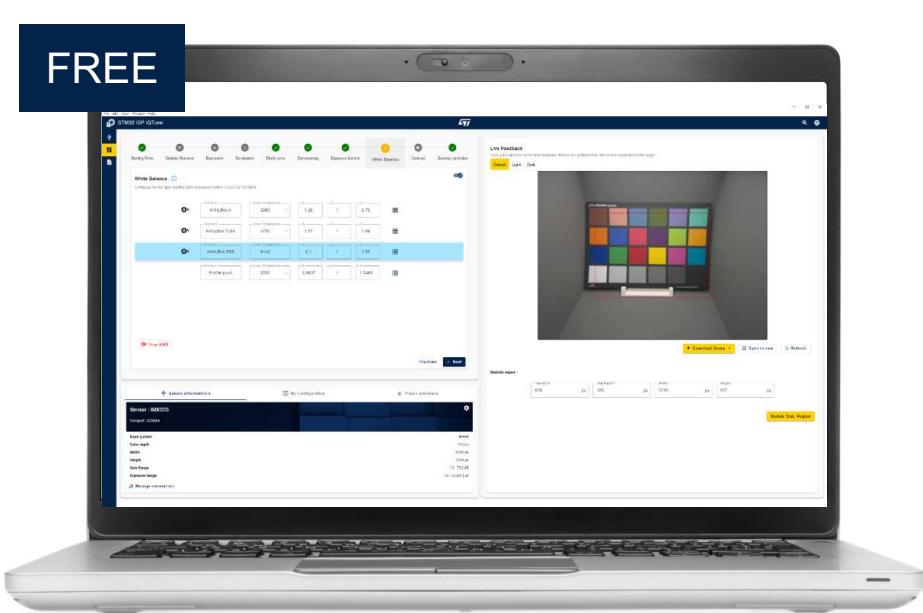
- Dimensioned for 5 Mpixel camera at 30 FPS
- Generates 3 different outputs from the same input for sending to the multimedia encoder or to the NPU
- [ISP IQTune Software tool](#) to tune ISP for cost savings and design flexibility

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.

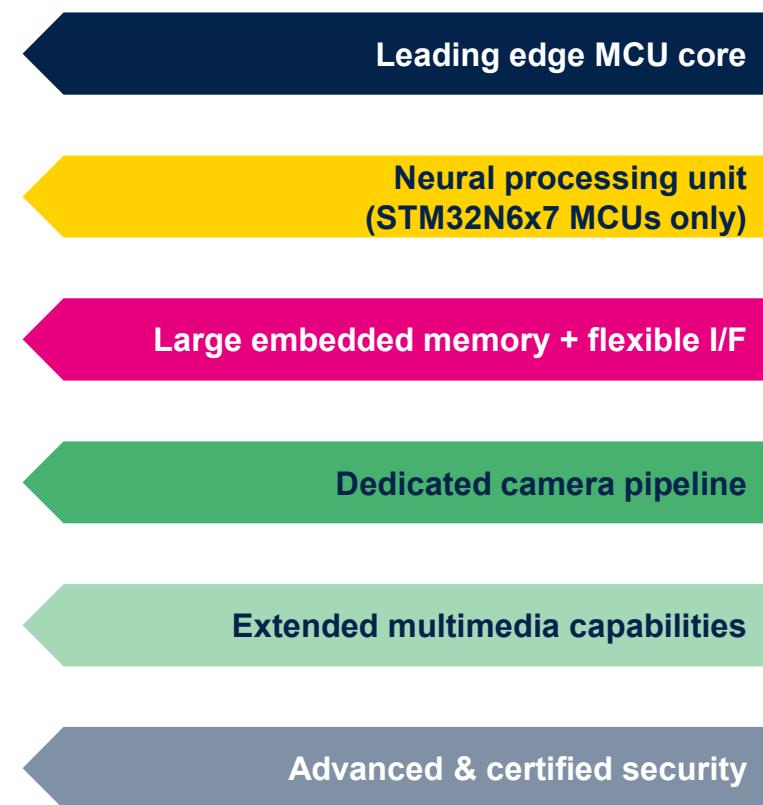
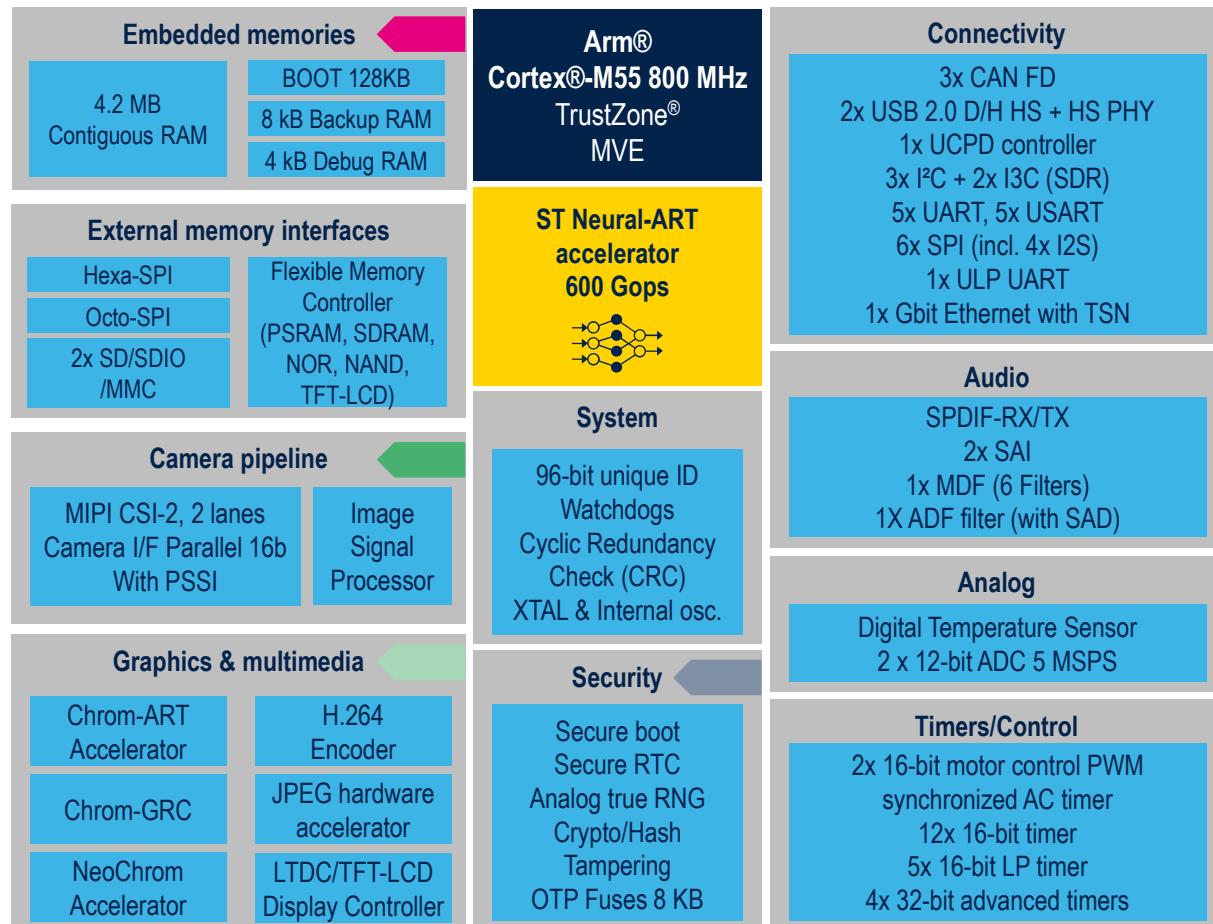
Flexibility to configure the ISP to your application requirements.



[Access now](#)



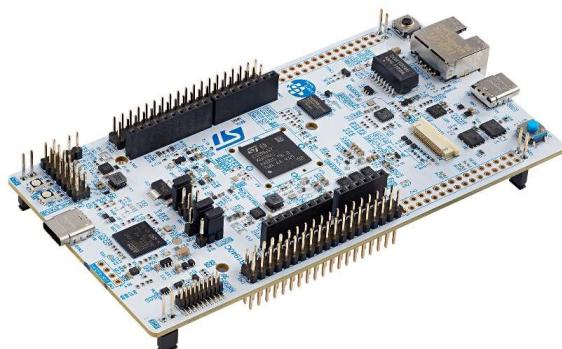
STM32N6x7 and STM32N6x5 MCUs





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

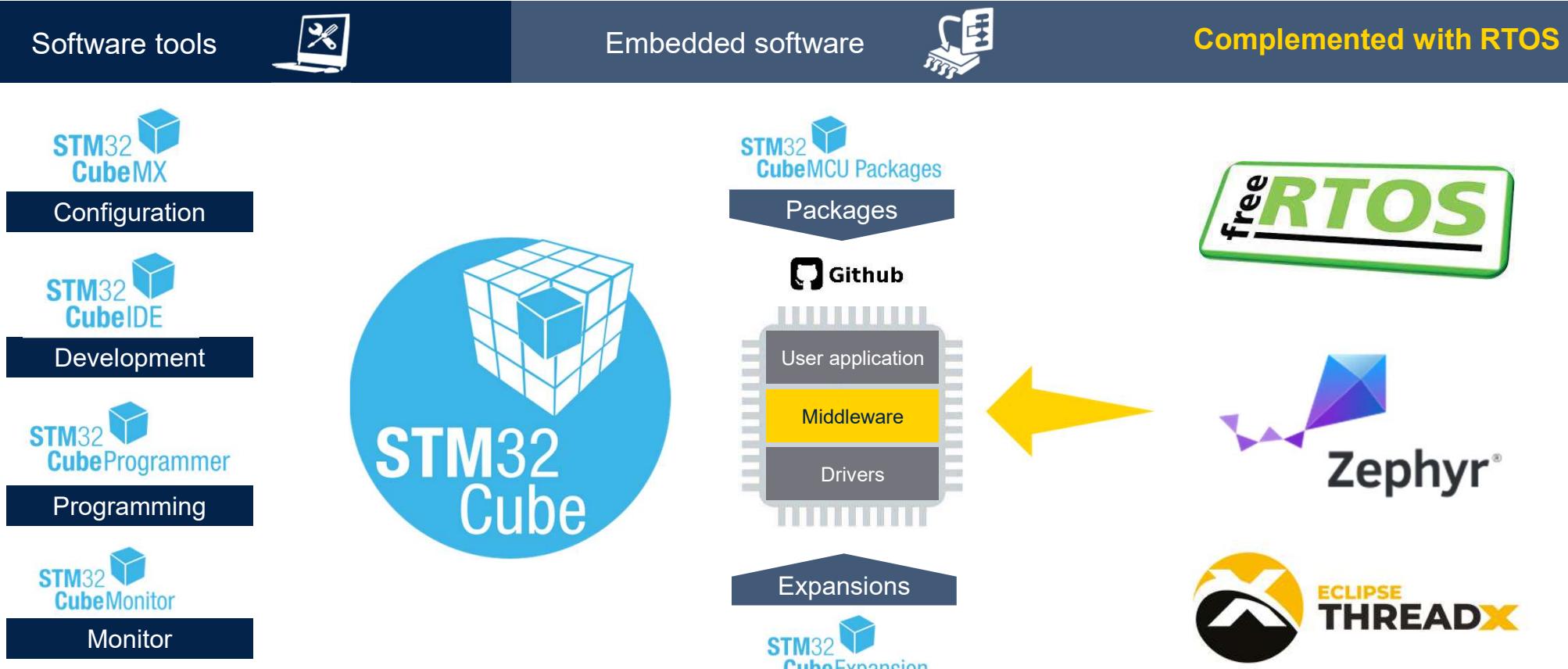


B-CAMS-IMX expansion board

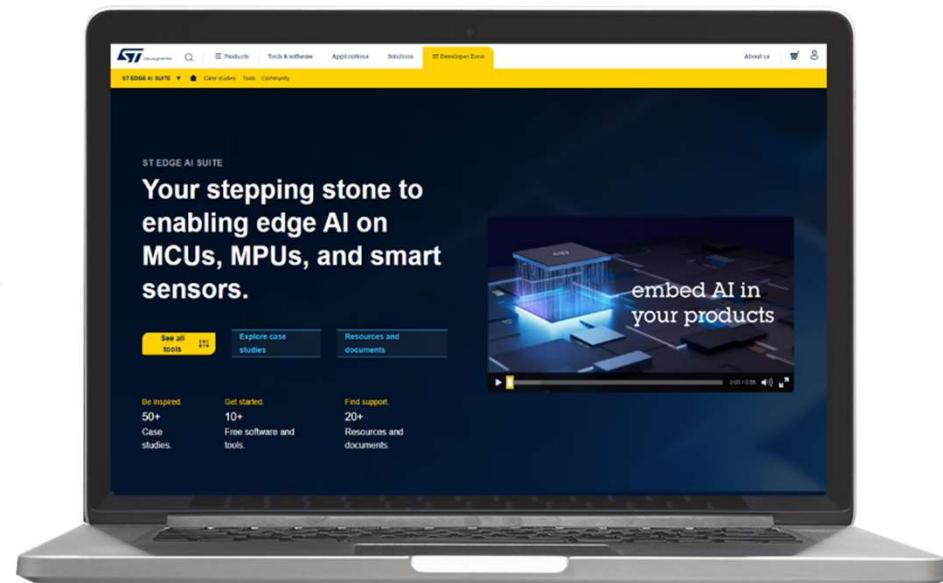
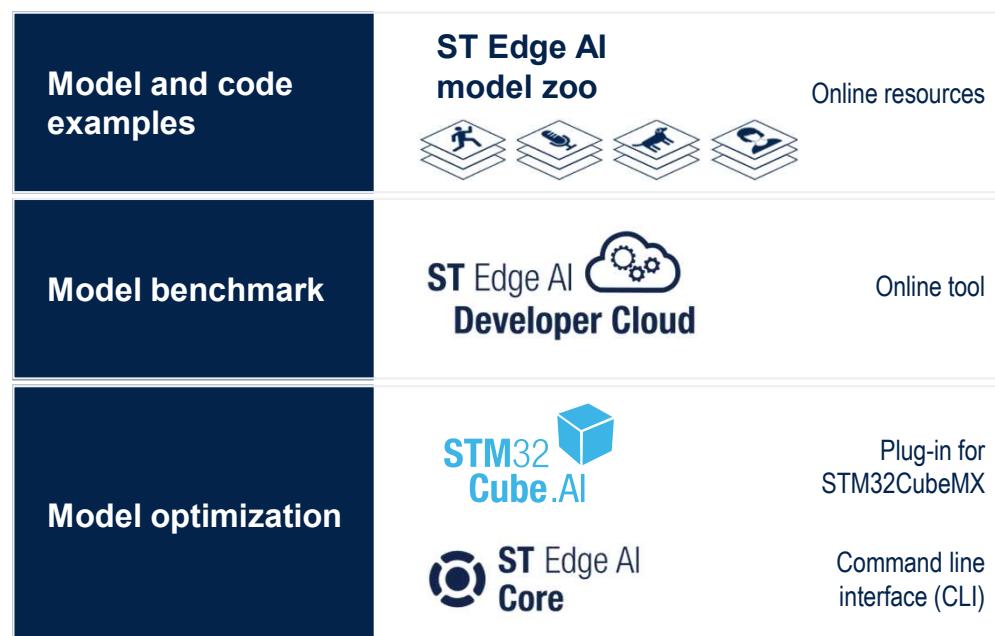
Rolling shutter camera, M12 removable lens, multizone direct Time-of-Flight sensor, inertial motion unit, Raspberry Pi compatible 22-pin connector.

Included in discovery kit.

Leveraging STM32Cube software suite



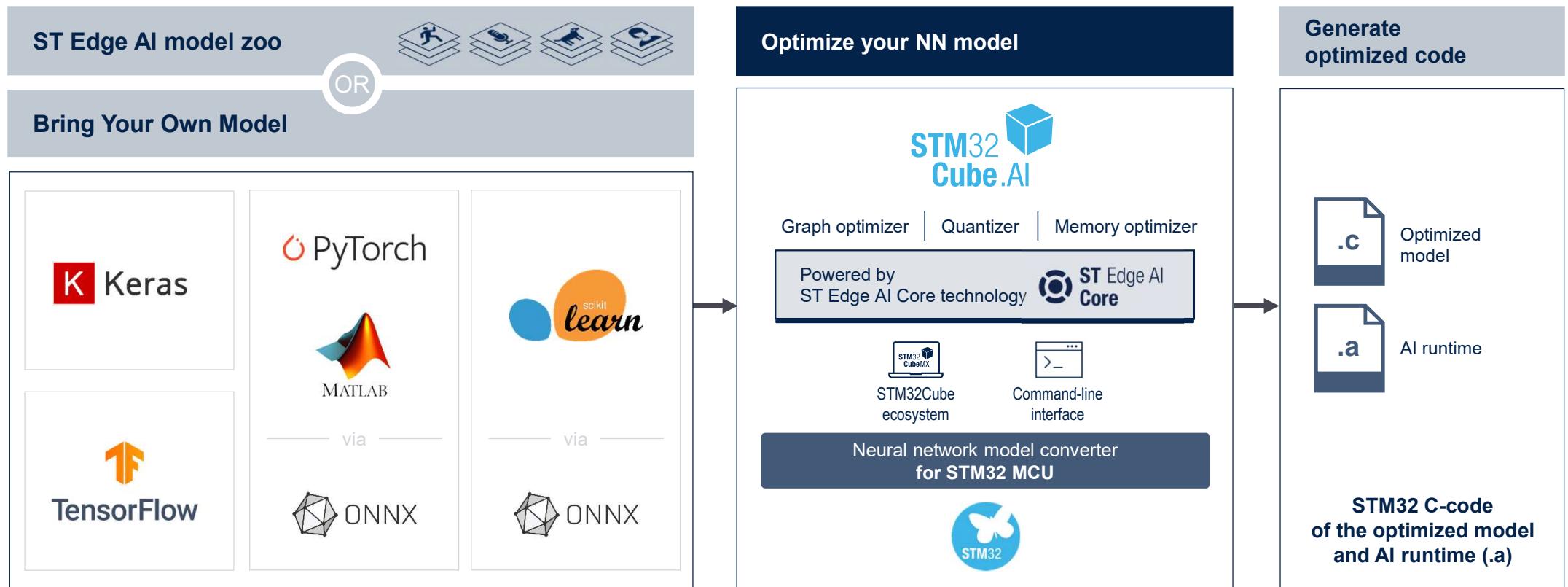
Online development tools for edge AI development



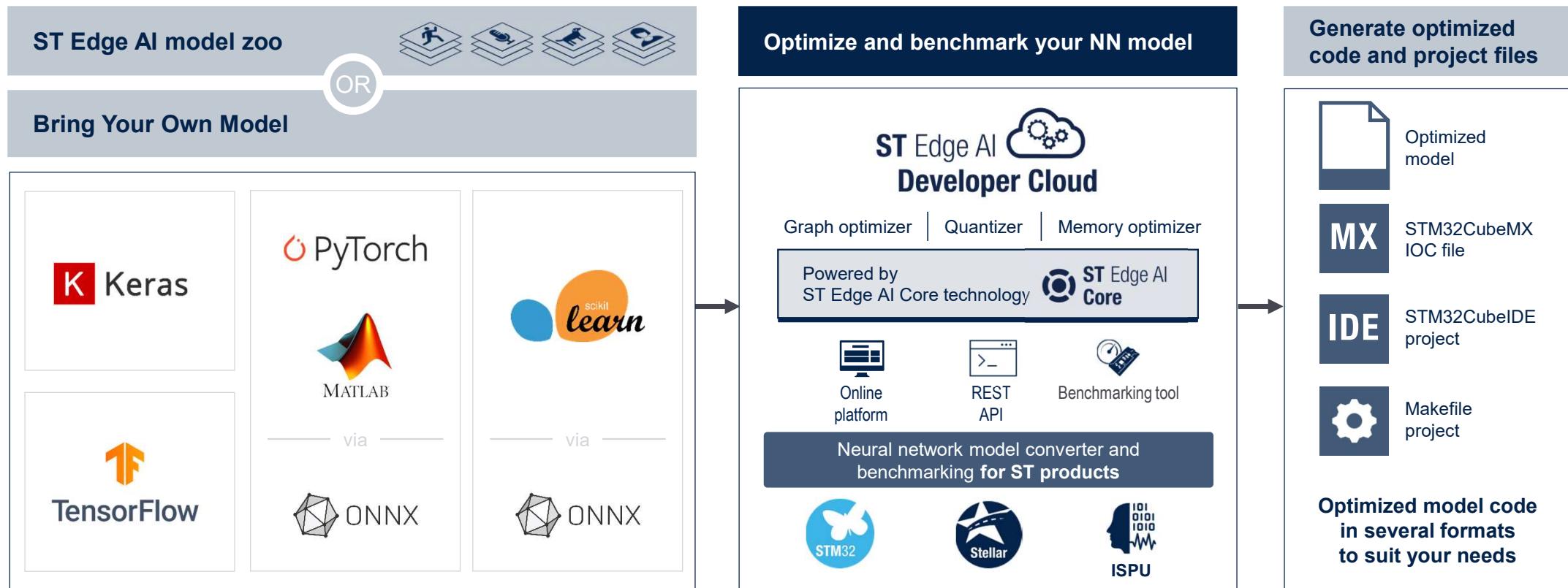
st.com/st-edge-ai-suite



STM32Cube.AI



ST Edge AI Developer Cloud

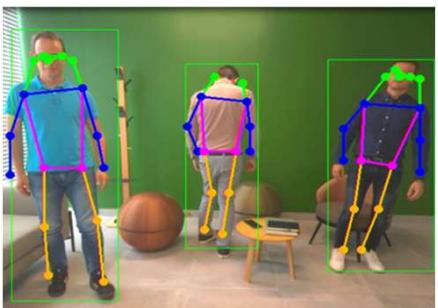


Get started with edge AI examples



People detection

- 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.



Multipose estimation

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



Hand landmark detection

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



H264 encoding / USB UVC streaming

- 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.



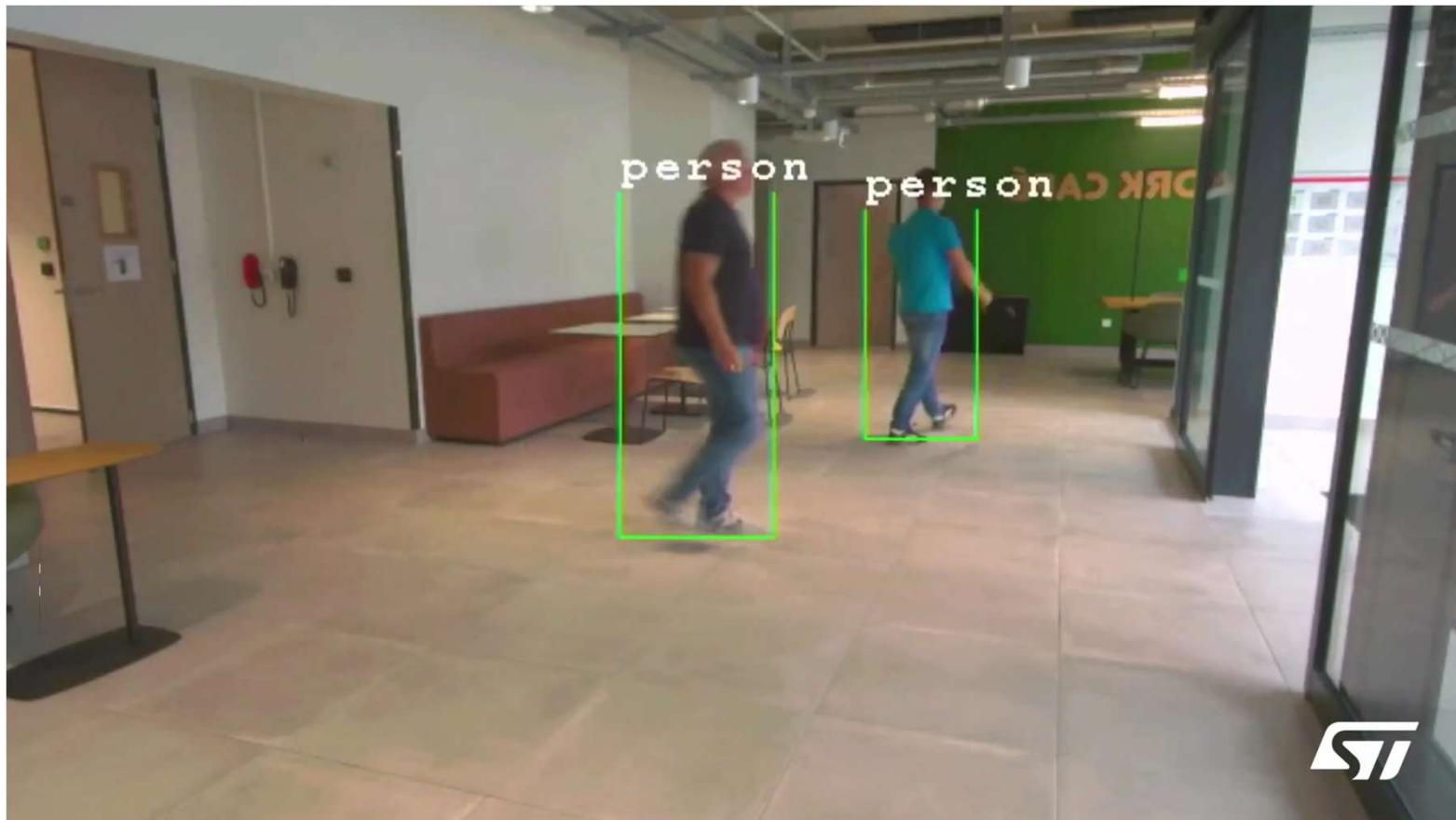
Power measurement

- 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

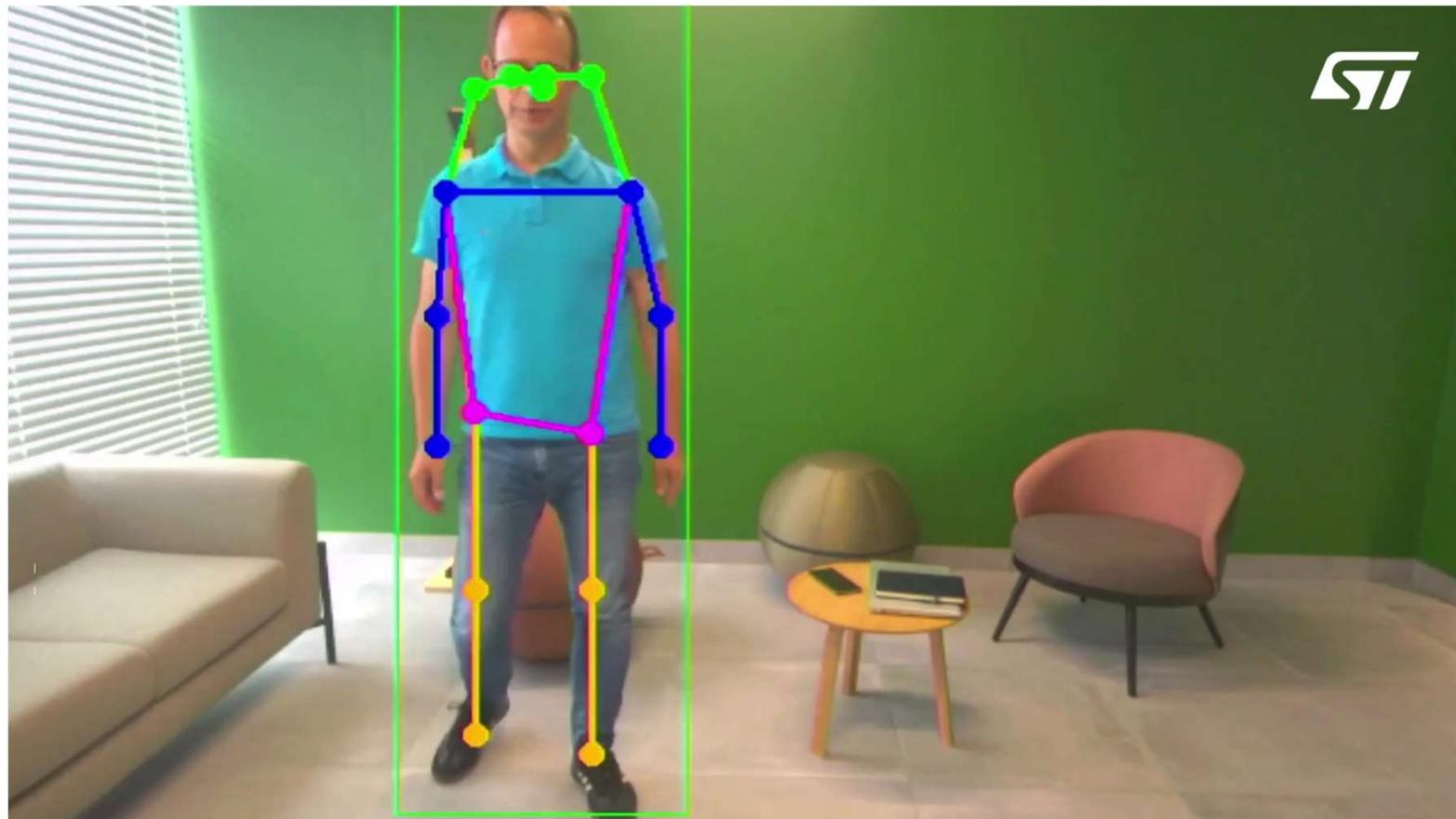
- 320 x 320
- 26 FPS
- 2.9 MB weights
- 1.6 MB activations

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 a composite of three parts. On the left, a dark blue display board features the ST logo at the top, followed by the text "Advance your product offering with edge AI" in white. Below this, there's a "ST Edge AI Suite" logo with four white squares and the text "Start your journey now" next to a QR code. A man's profile is visible at the bottom left. In the center, a hand is shown from the front with yellow circular landmarks placed on the palm and fingers, connected by black lines to form a polygonal shape. On the right, a dark blue sidebar contains the ST logo at the top. Below it, the text "KEY METRICS (Two models in parallel)" is displayed. Under "Palm detector", there is a bulleted list: • 192 x 192 • 1.1 MB weights • 1.1 MB activations. Under "Hand landmark", there is another bulleted list: • 224 x 224 • 3.2 MB weights • 1 MB activations. At the bottom of the sidebar, the text "KEY APPLICATIONS" is followed by a bulleted list: • Touchless HMIs • Smart appliances • Smart industry.

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

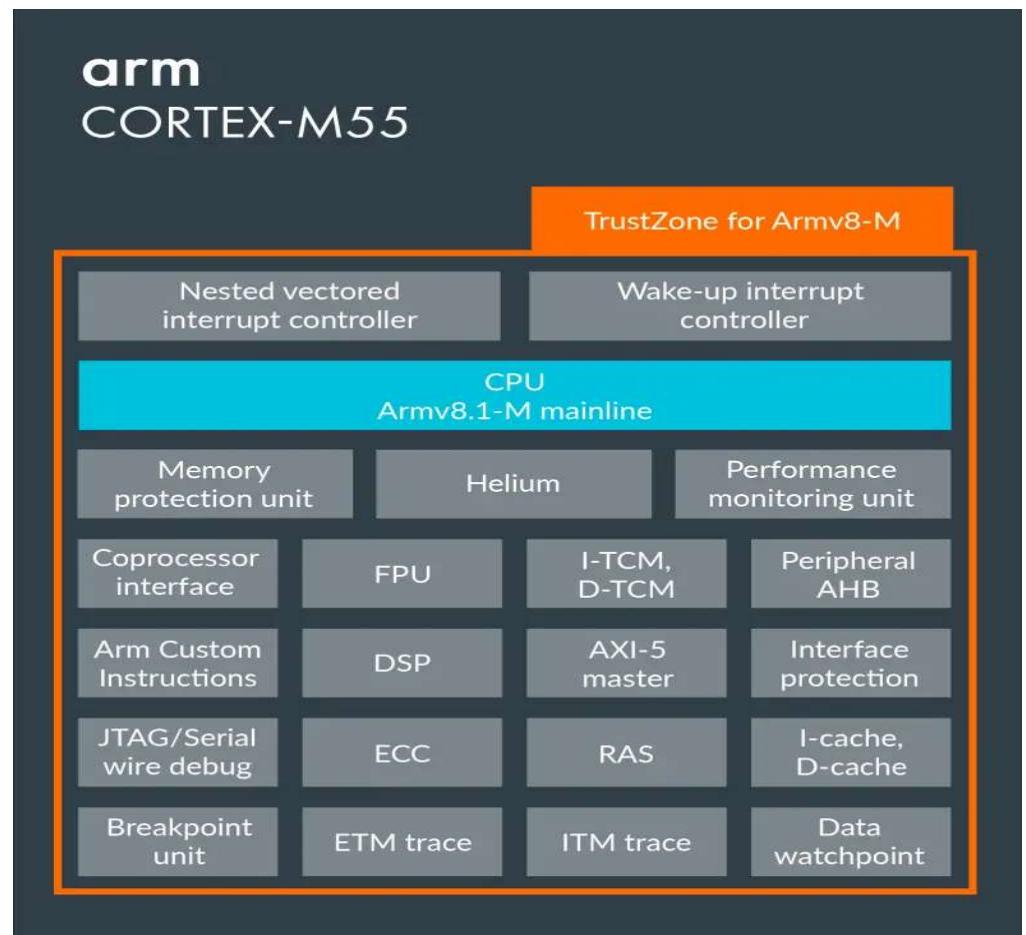
- Touchless HMIs
- Smart appliances
- Smart industry

STM32N6 Architecture & Features

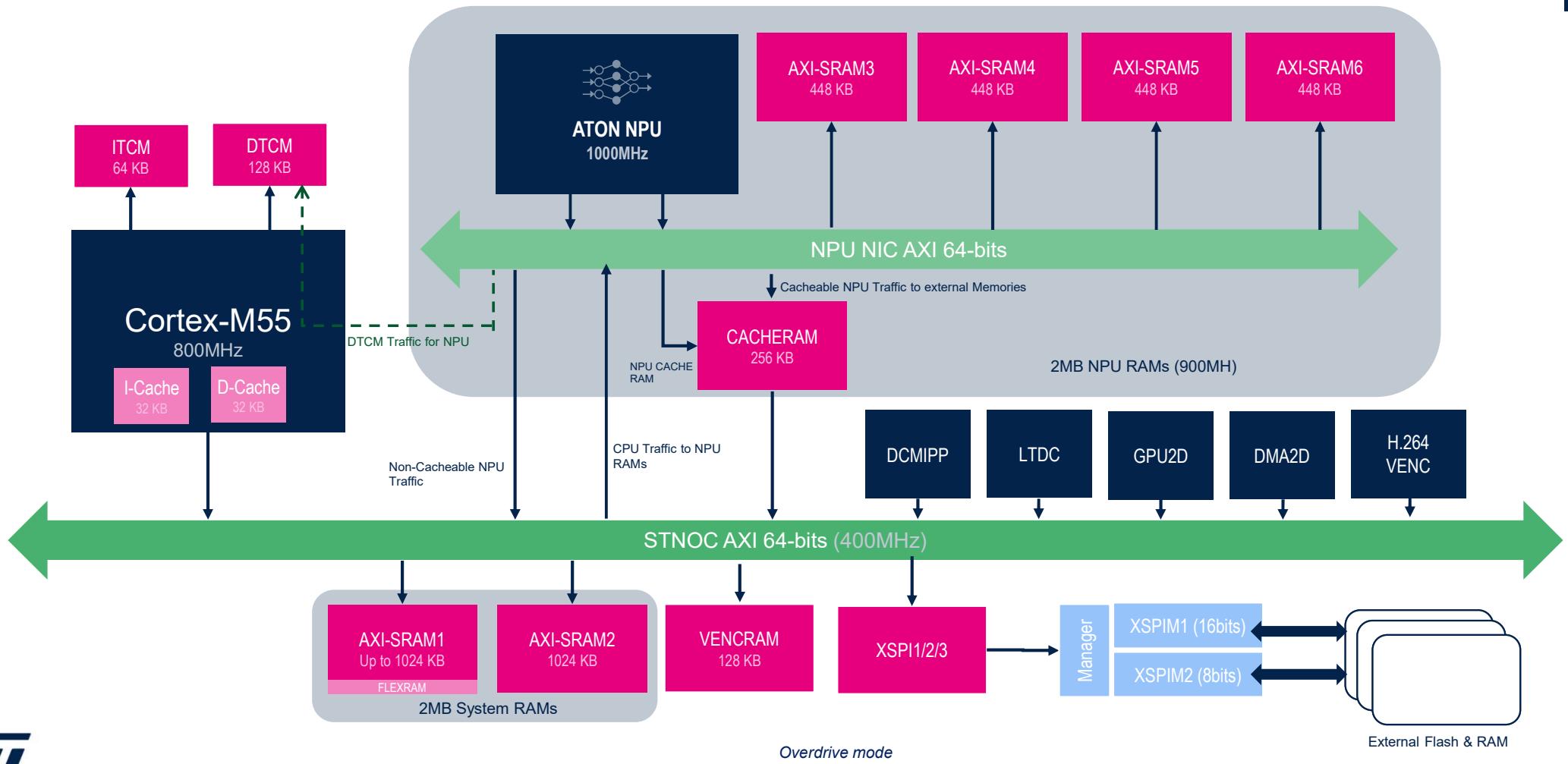


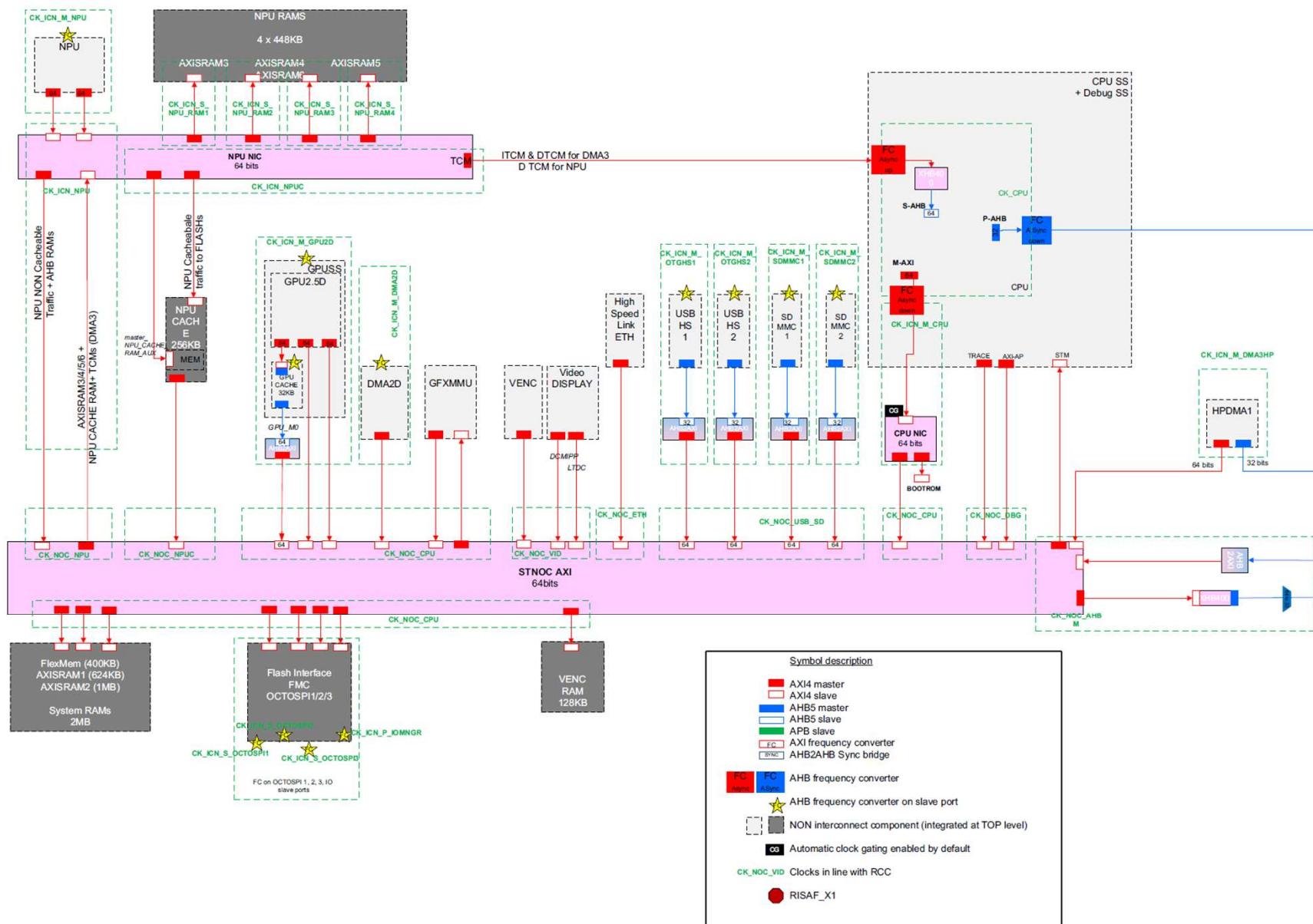
Cortex®-M55 processor overview

- 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 N61ps 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
	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

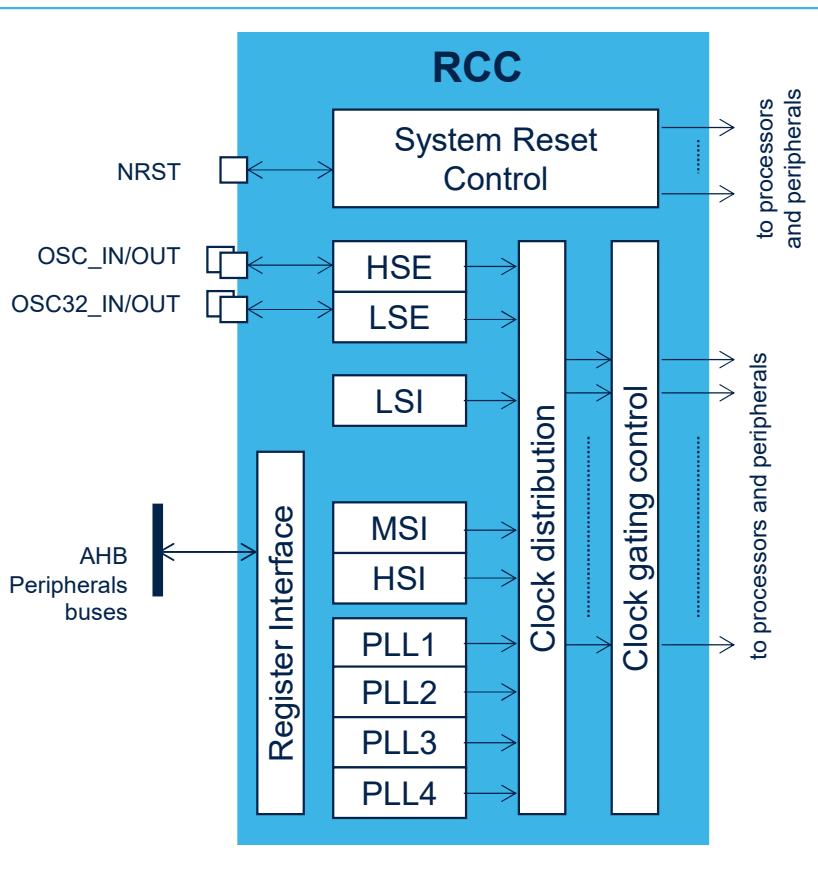
Description	Non-Secure	Secure	Size
ITCM – Base line	0x0000 0000	0x1000 0000	64KB
ITCM – FLEXMEM extension			Up to 192KB
Boot ROM	0x0800 0000	0x1000 0000	128KB
DTCM – Base line	0x2000 0000	0x3000 0000	128KB
DTCM – FLEXMEM extension	0x2002 0000	0x3002 0000	Up to 128KB
AXI-SRAM1 (FLEXMEM extension bites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

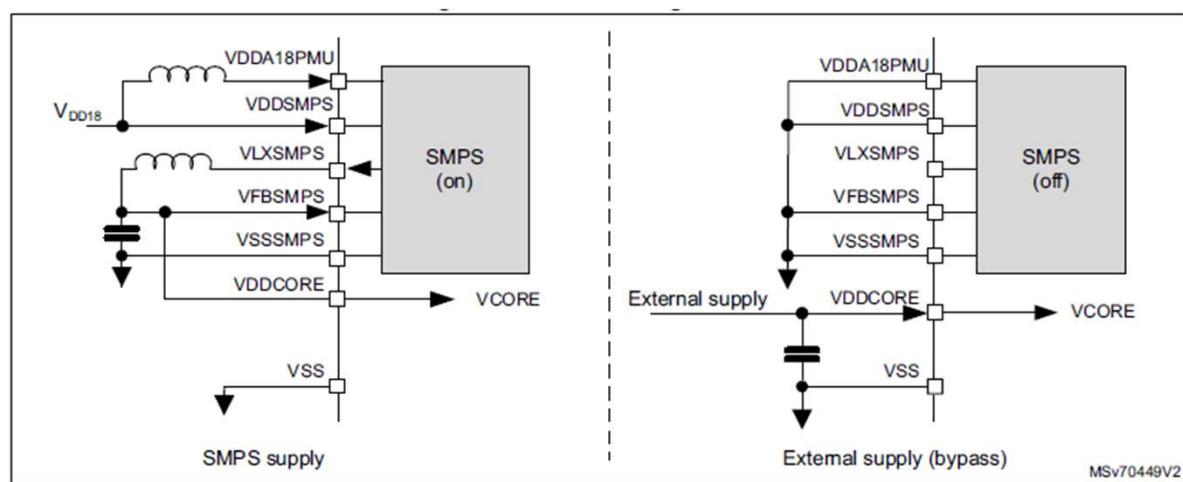
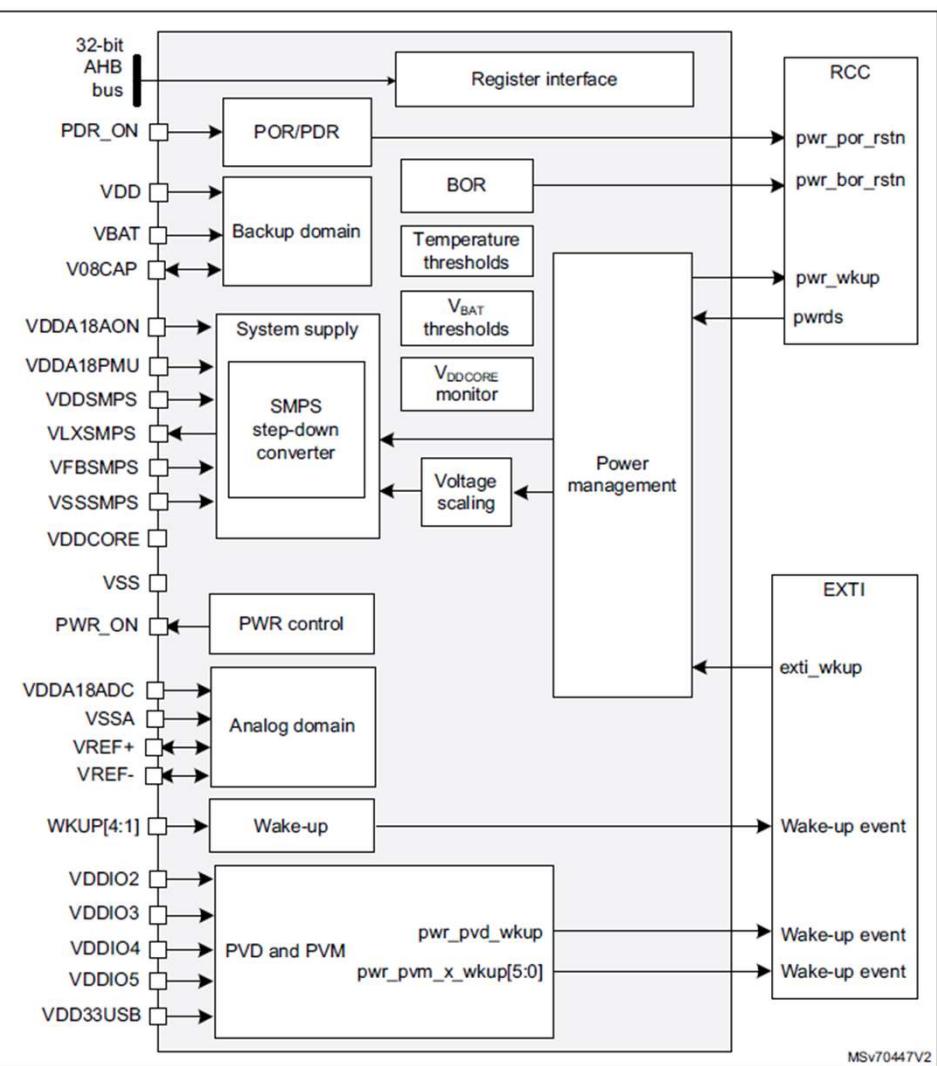
- High flexibility regarding the clock sources to meet consumption and accuracy requirements.
- Safe and flexible reset management

STM32N6xx - Power supplies

Name	Typical or Range	Description
VDD	1.8V / 3.3V	Power supply for most I/Os
VDDIO2/3/4/5	1.8V / 3.3V	Independent power supply for IOs related to SDMMC1, SDMMC2, XSPIM1 and XSPIM2
VDDA18AON	1.8V	Analog power supply for system analog such as reset, power management, oscillators and OTP
VDDCORE	0.81V (0.89V overdrive)	Digital power supply for Core Cortex-M55 domain. Overdrive available on some part number to get higher CPU frequency
VDDSMPS, VLXSMPS, VFBSMPS, VSSMPS		Step-down converter , SMPS connected
VDDCSI	0.81V (0.89V overdrive)	Power supply for CSI PHY digital supply input. usually be connected to VDDCORE
VDDA18PLL	1.8V	Analog Power supply for RCC PLLs
VDDA18CSI, VDDA18USB	1.8V	Analog power supply for various PHY
VDDA18ADC	1.8V	Analog Power supply for ADCs and VREFBUF
Vref+ , Vref-		external reference voltage for ADCs, independent from any other supply
VDDA18PMU	1.1V / 1.2V / 1.35V	Power supply for DDR Physical Interface (PHY) and related IOs, respectively for LPDDR4, DDR4 and DDR3L
VDD33USB	3.3V	Power supply input for respectively USB PHY and USB Type-C Power Delivery PHY and related IOs
VBAT	1.62 – 3.6V	Backup supply, usually connected to a 3V coin-cell battery
Internally generated Power Supplies		
V08CAP	0.8V	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 the 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

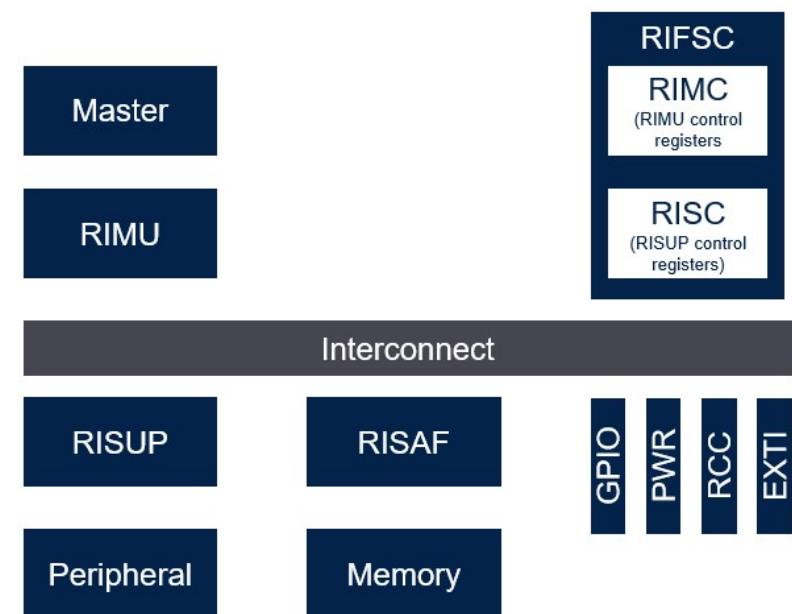
Resource isolation

- 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)

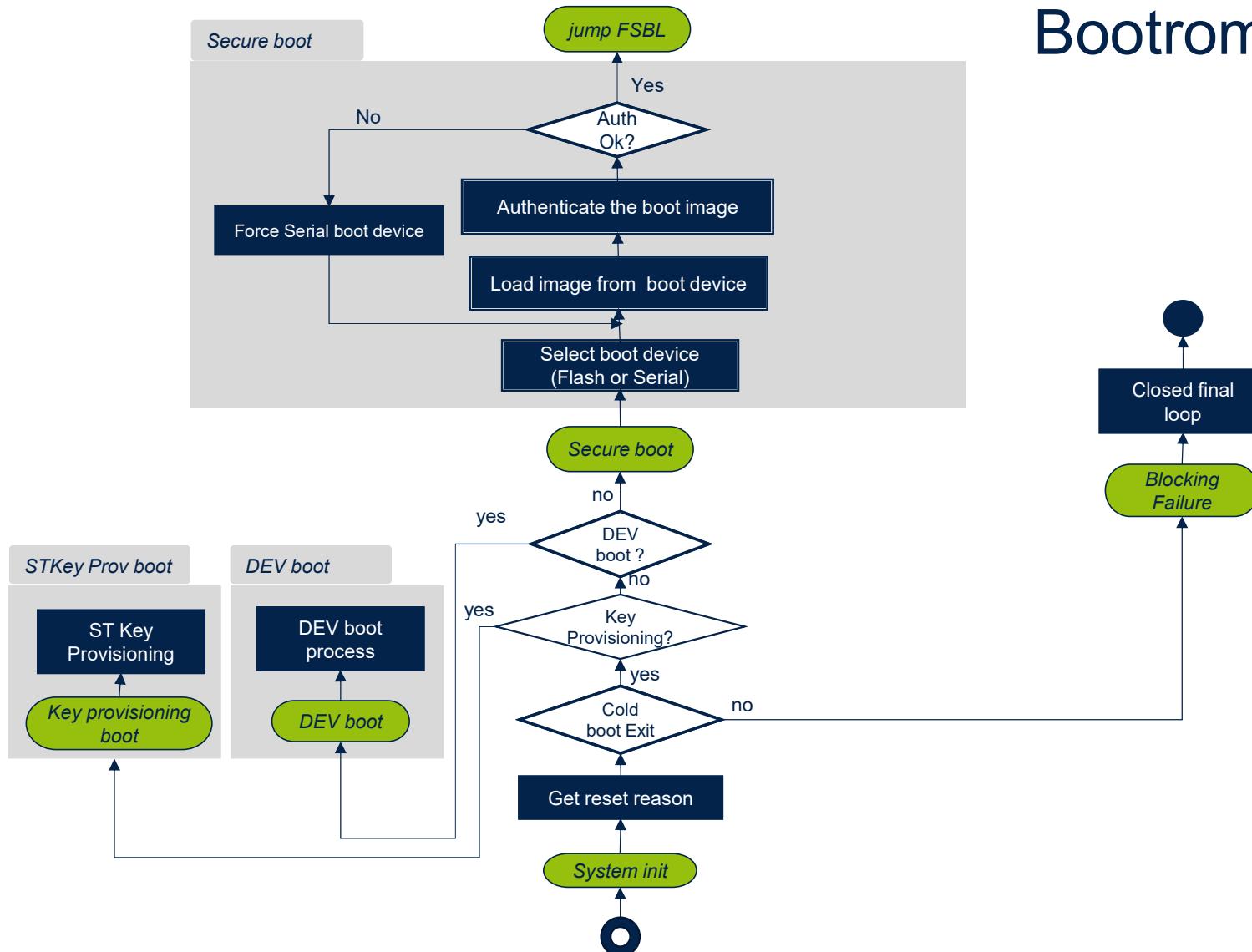
OTP word	OTP bits	Name or description	Detailed description
OTP96 to OTP99	-	Reserved	Reserved
OTP100 to OTP102	-	Reserved	Reserved
OTP103	-	Reserved	Reserved
OTP104 to OTP123	-	Reserved	Reserved
OTP124	-	HCONF1	-
	[0]	IWDG1_HW	IWDG1 start on reset
	[1]	IWDG1_FZ_STOP	IWDG1 freeze in Stop mode
	[2]	IWDG1_FZ_STANDBY	IWDG1 freeze in Standby mode
	[9:3]	Reserved	Reserved
	[10]	RST_STOP	Reset caused if the device is put in Stop mode
	[11]	RST_STANDBY	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

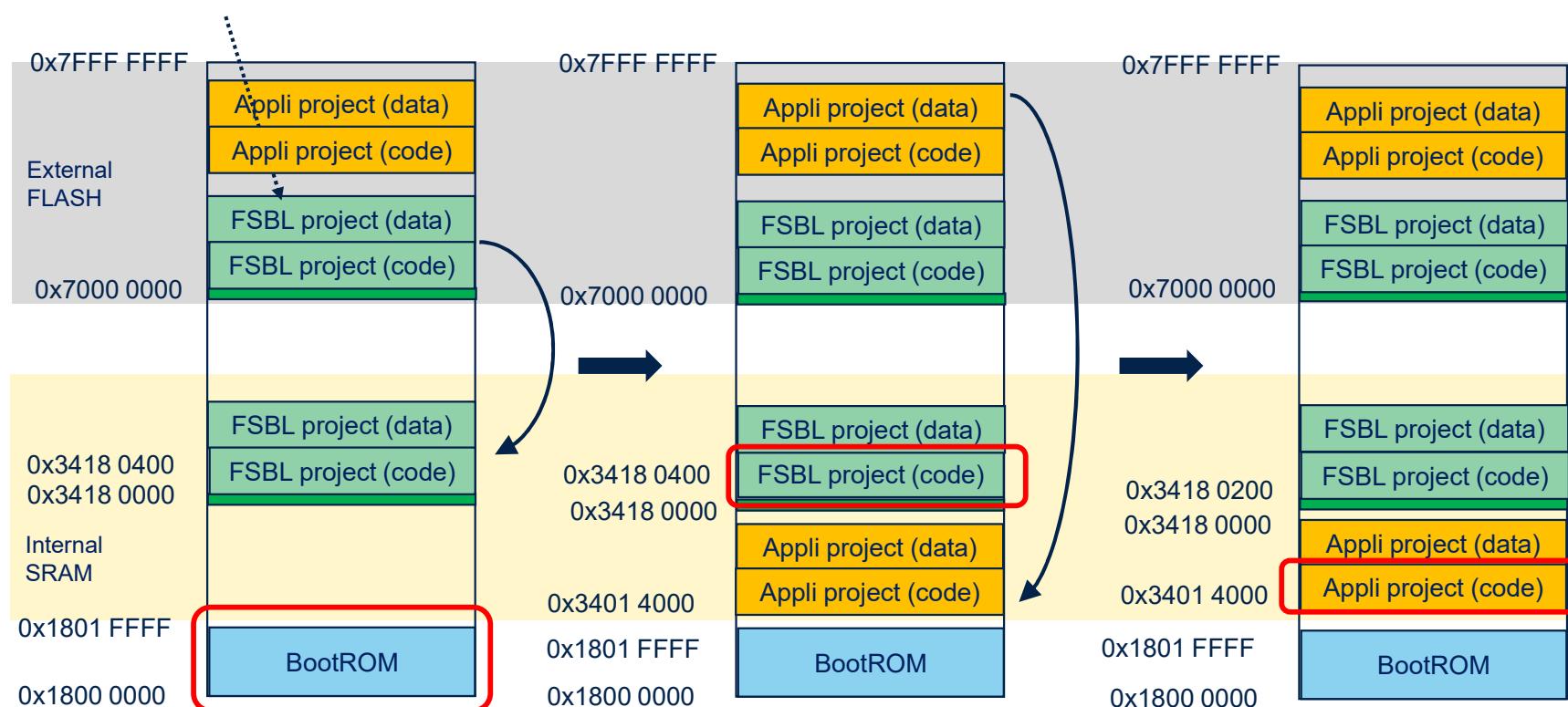


Bootrom overview



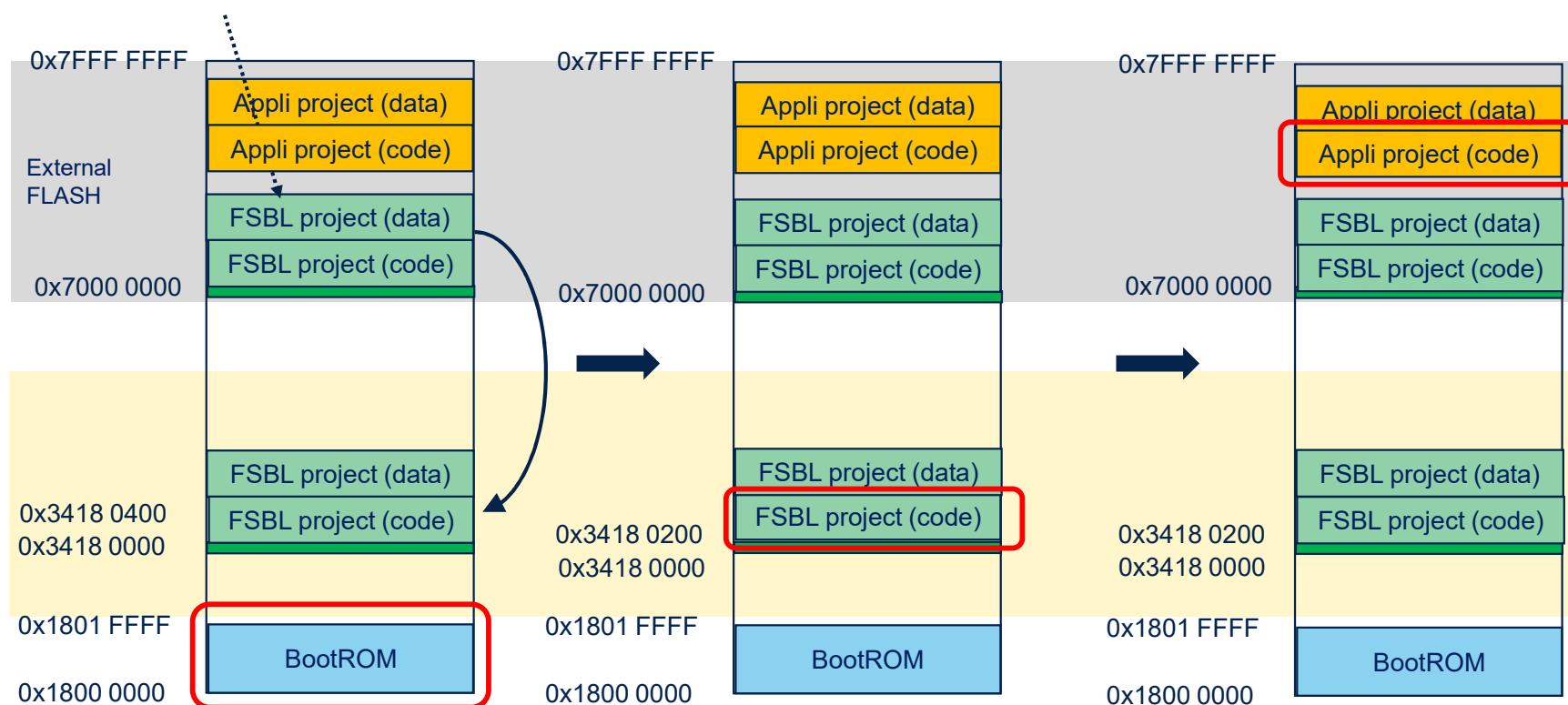
Boot from external Flash. FSBL + Load & Run

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



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

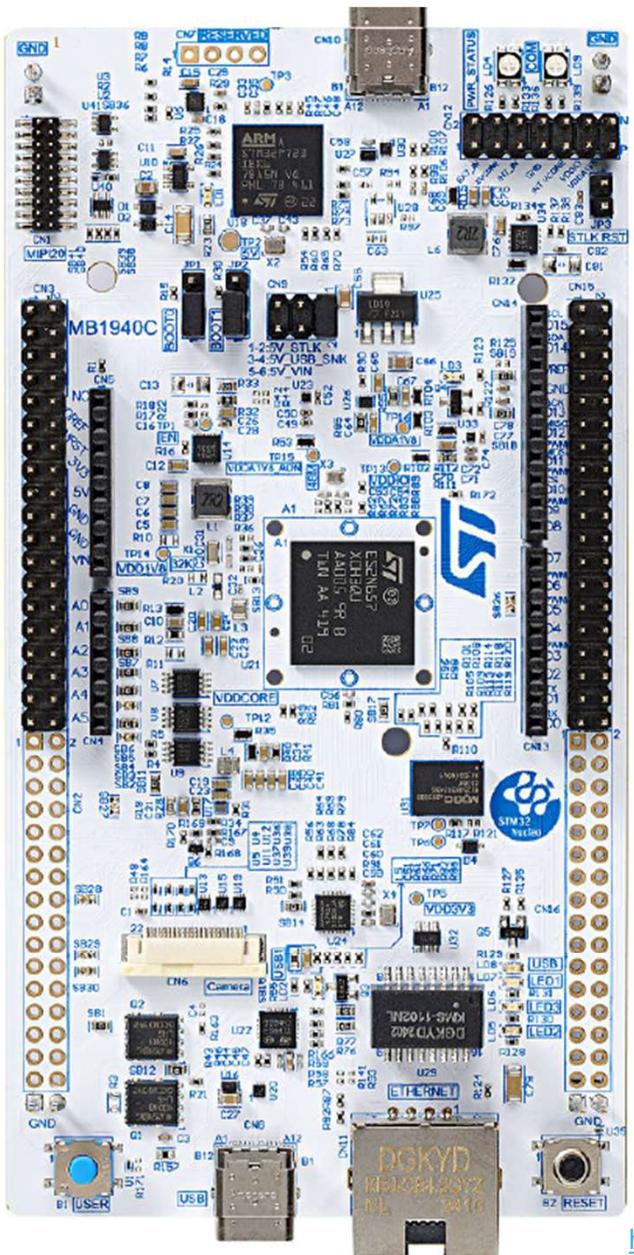
512-byte header (required by bootROM) + 512-byte long padding.



NUCLEO-N657Z0-Q



STM32N6 Nucleo-144 boards



Common features

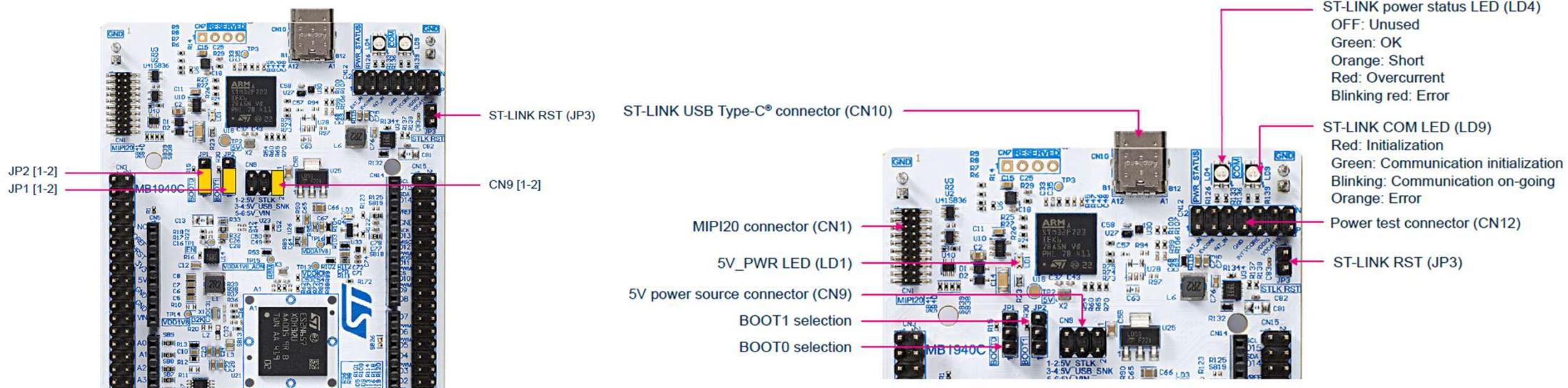
- STM32 microcontroller in an LQFP144, TFBGA225, or VFBGA264 package
- 3 user LEDs
- 1 user push-button and 1 reset push-button
- 32.768 kHz crystal oscillator
- Board connectors:
 - SWD
 - ST morpho expansion connector
- Flexible power-supply options: ST-LINK USB V_{BUS} , USB connector, or external 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

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	-

Supported boot configuration

Table 8. Supported boot configurations

Boot Config.	Attached boot	Applied boot memory setup	Description
0	Dev boot	-	Dev boot mode selected in CLOSED-UNLOCKED life cycle
1	Serial boot	-	Serial boot monitors in parallel USB and USART links
2	SD device	SD1	SD device, connected to SDMMC1
3	SD device	SD2	SD device, connected to SDMMC2
4	eMMC device	eMMC1	eMMC device, connected to SDMMC1
5	eMMC device	eMMC2	eMMC device, connected to SDMMC2
6	sNOR device	XSPI NOR	sNOR device, connected to XSPIM_P2
7	HyperFlash™	XSPI HYPER	HyperFlash™ device, connected to XSPIM_P2

The boot ROM code evaluates the register bits BOOTS[0:1] to determine the applied boot configuration. The register bits BOOTS[0:1] reflect the level of the external boot pins as latched at reset.

Allocated pins for BOOTS[0:1]:

- **BOOTS[0]:** Boot0 pin (dedicated pin)
- **BOOTS[1]:** Boot1 pin (nondedicated pin, PA6)

The selected Boot1 pin, PA6, can be overwritten by fuses, BOOTROM_CONFIG_10[24:21] (port ID) and BOOTROM_CONFIG_10[28:25] (pin ID). The Boot1 pin check has priority over the Boot0 pin check. If the Boot1 pin is not set, the Boot0 pin is checked. If the Boot1 pin is selected but not allowed in the current life cycle, the Boot0 pin is then checked.

The flash boot configuration is checked using fuses in BOOTROM_CONFIG_2[8:5], OTP_WORD11. The boot configuration coding rules are defined in [Table 9](#).

Table 9. Boot configuration coding

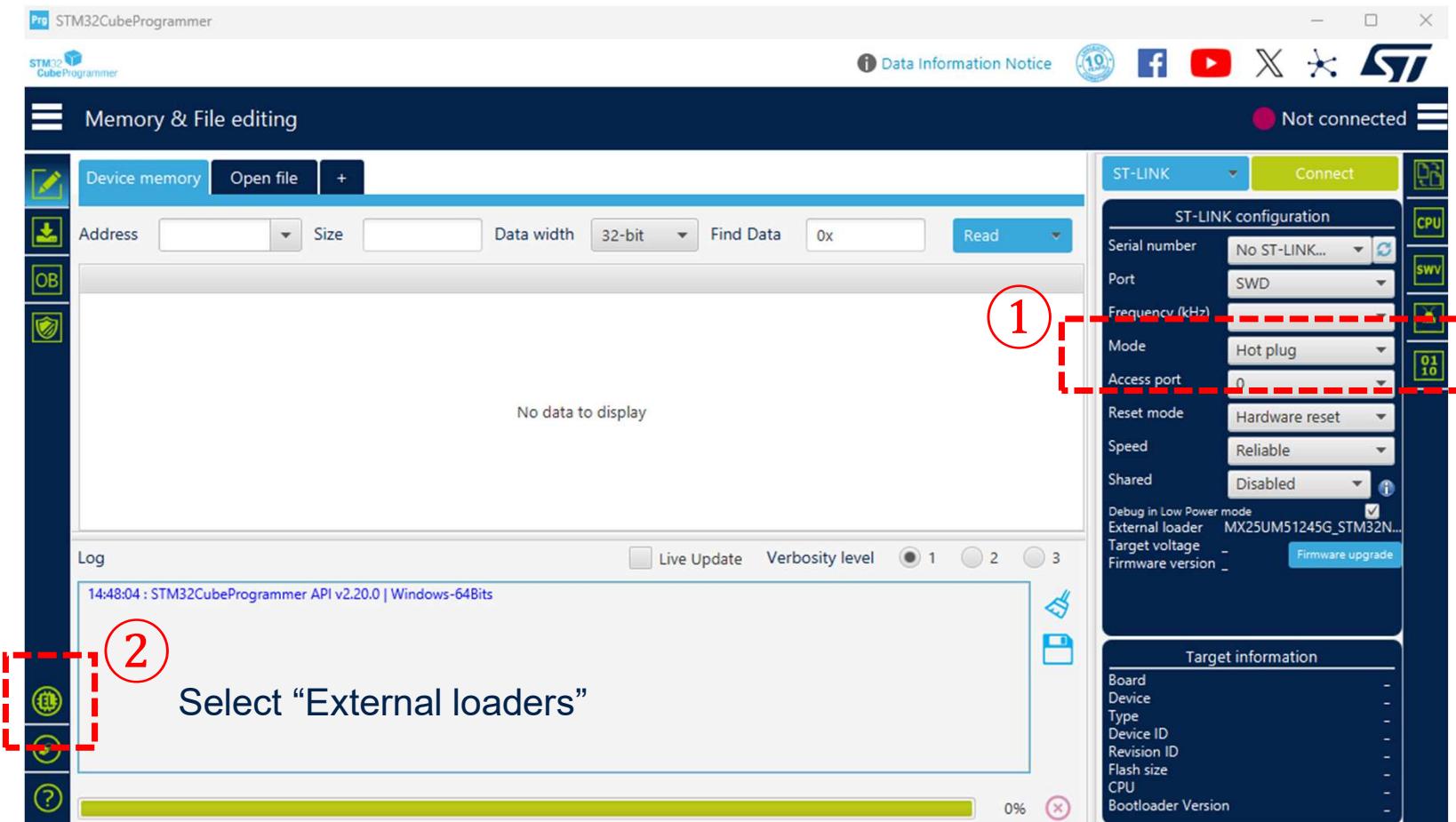
BOOTS[1:0]	BOOTROM_CONFIG_2[8:5]	Applied boot configuration
0	1	Boot config. 2
0	2	Boot config. 4
0	3	Boot config. 6
0	5	Boot config. 7
0	7	Boot config. 3
0	8	Boot config. 5
0	0	Boot config. 6 (default configuration)
1	X	Boot config. 1
2	X	Boot config. 0

Table 10. Hardware blocks required per boot category

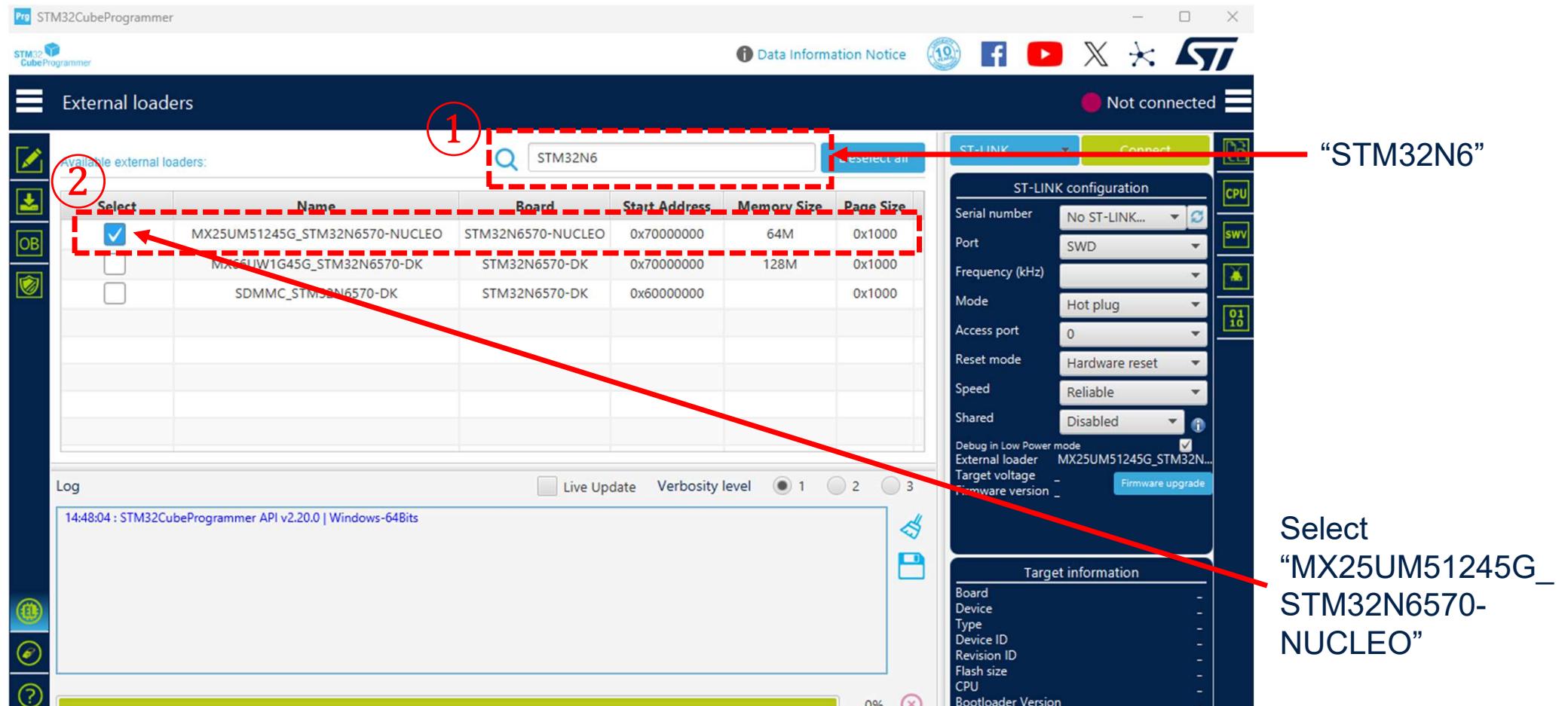
Boot category	Required hardware blocks
Serial boot	USB OTG1, USART1, USART2, UART4
SD1	SDMMC1
eMMC1	SDMMC1
SD2	SDMMC2
eMMC2	SDMMC2
XSPI NOR	XSPIM_P2, XSPI1
XSPI HyperFlash™	XSPIM_P2, XSPI1



Check Hardware Board

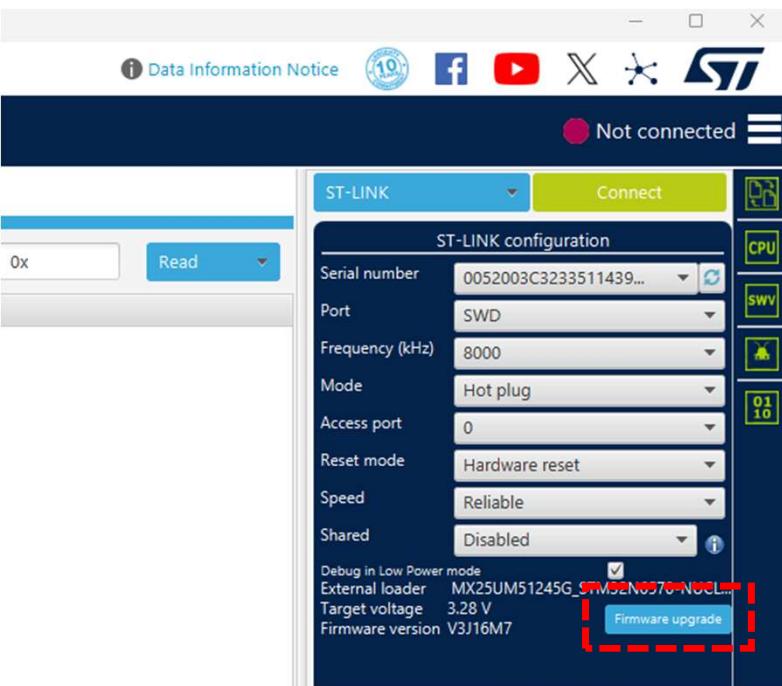


Check Hardware Board

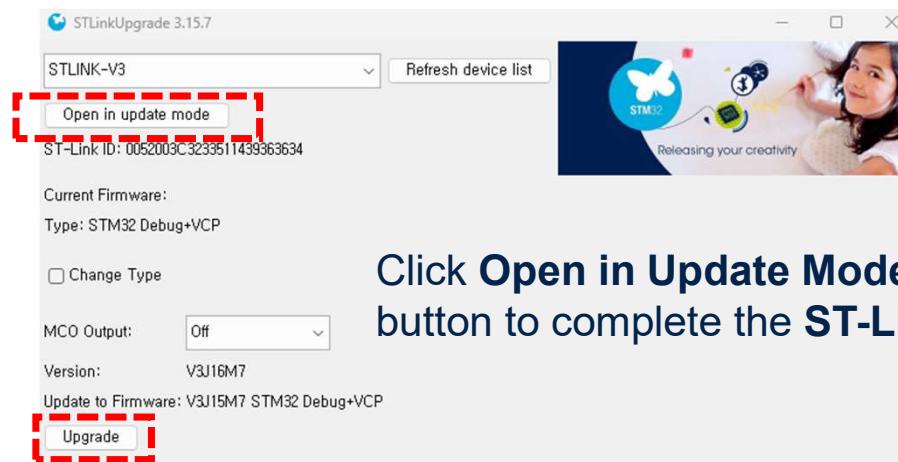


Select
"MX25UM51245G_
STM32N6570-
NUCLEO"

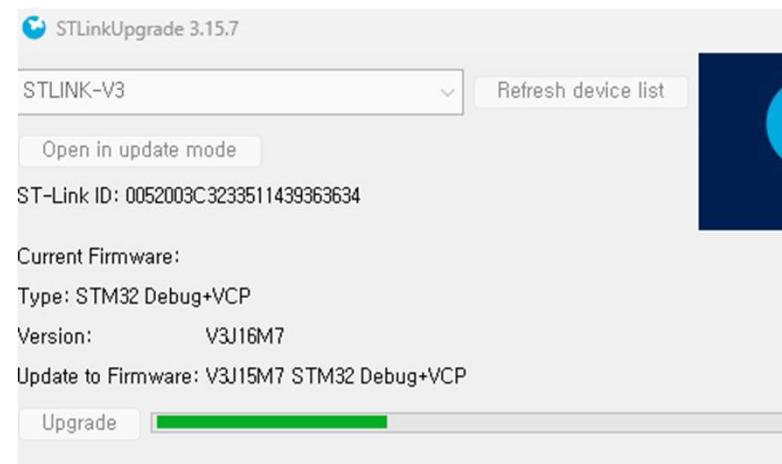
Check Hardware Board



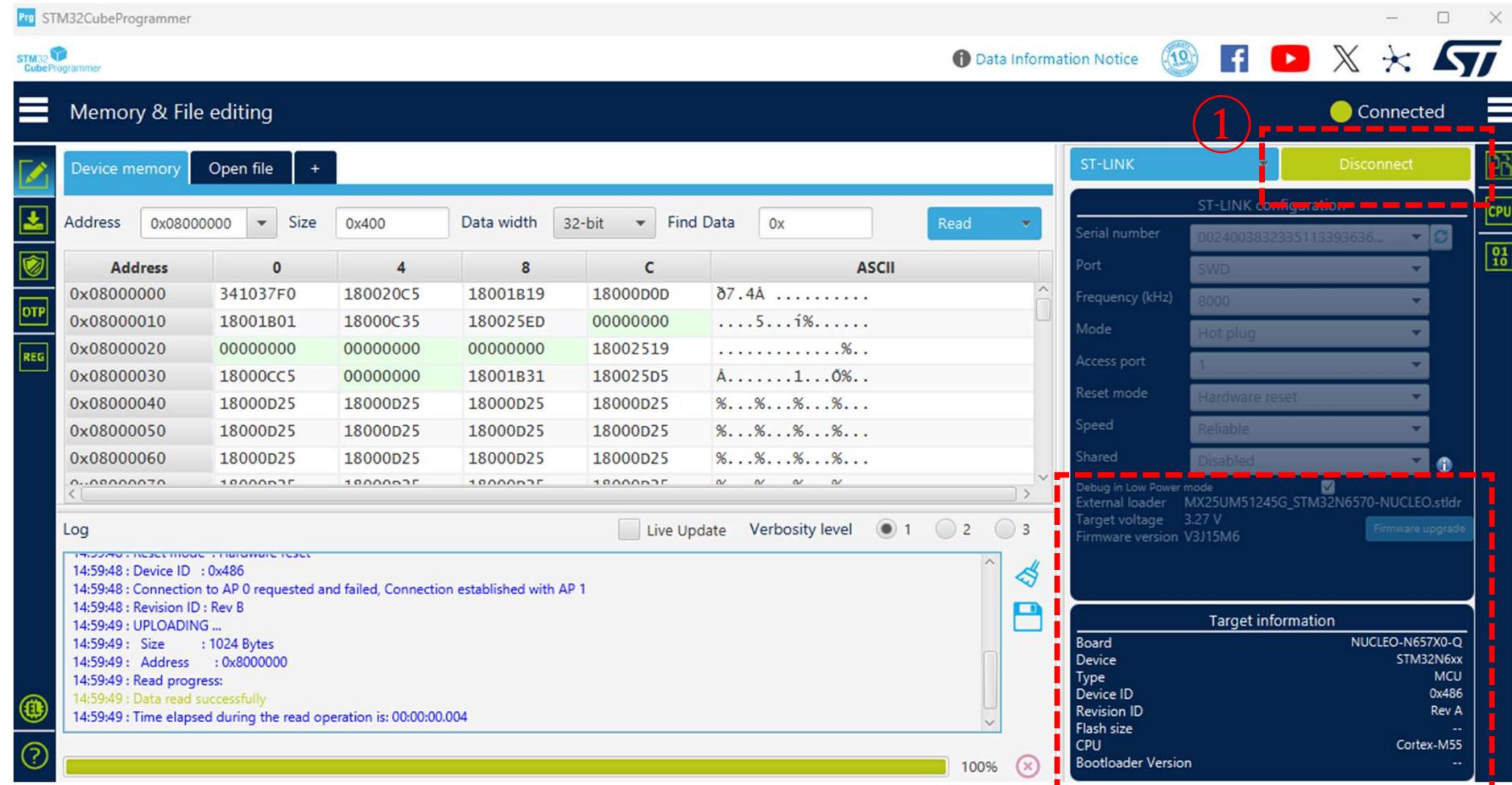
Click the **Firmware Upgrade** button.



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

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

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

NUCLEO-N657X0-Q Board schematic:

https://www.st.com/resource/en/schematic_pack/mb1940-n657x0q-c02-schematic.pdf



Installation & Download Guide (1/3)

STM32CubeIDE (This Hands-on uses version 1.18.0)

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

STM32CubeMX (This Hands-on uses version 6.14.0)

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

STM32CubeProgrammer (This Hands-on uses version 2.19.0)

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

STEdgeAI-NPU (This Hands-on uses version 10.0.0)

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

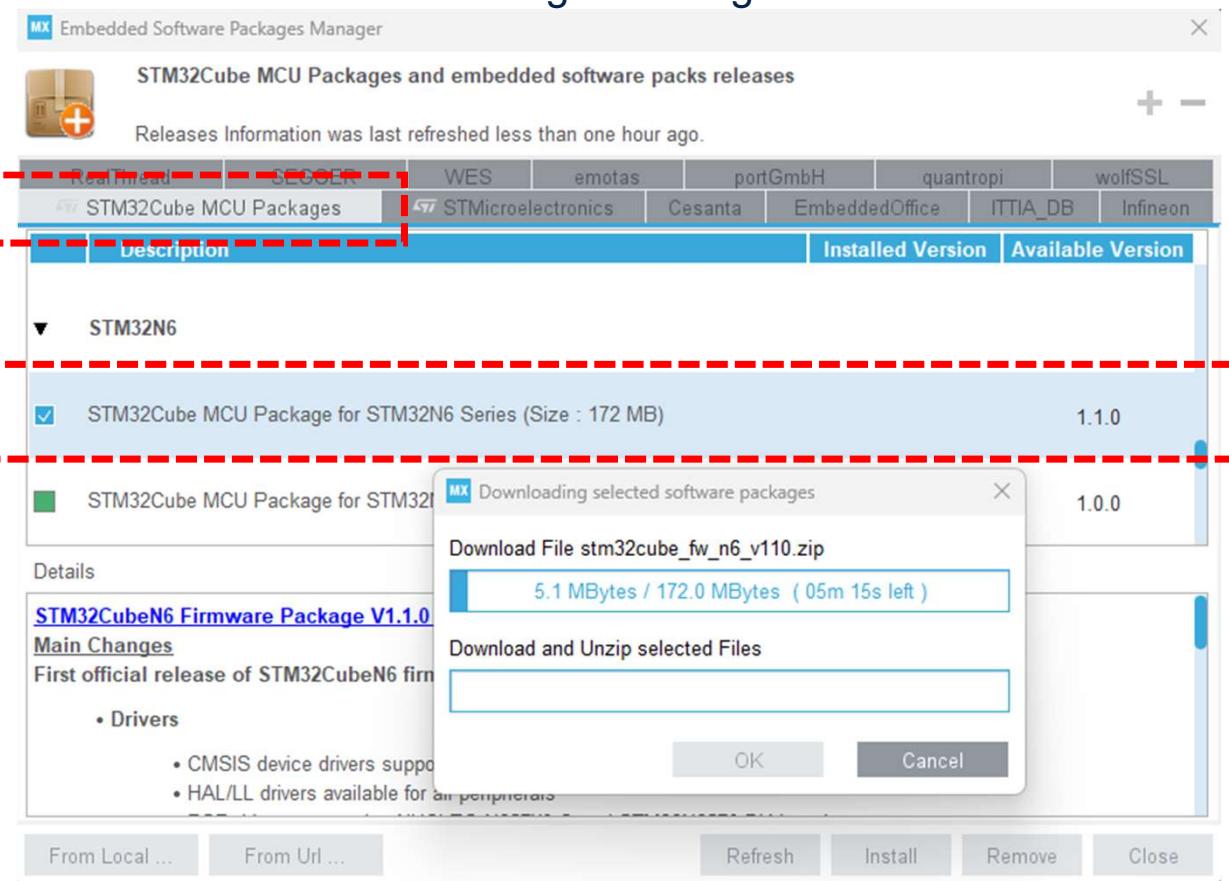
Hands On and Example code

https://github.com/BlaineMoon30/STM32_Expert_Training_AI_STM32N6



Installation & Download Guide (2/3)

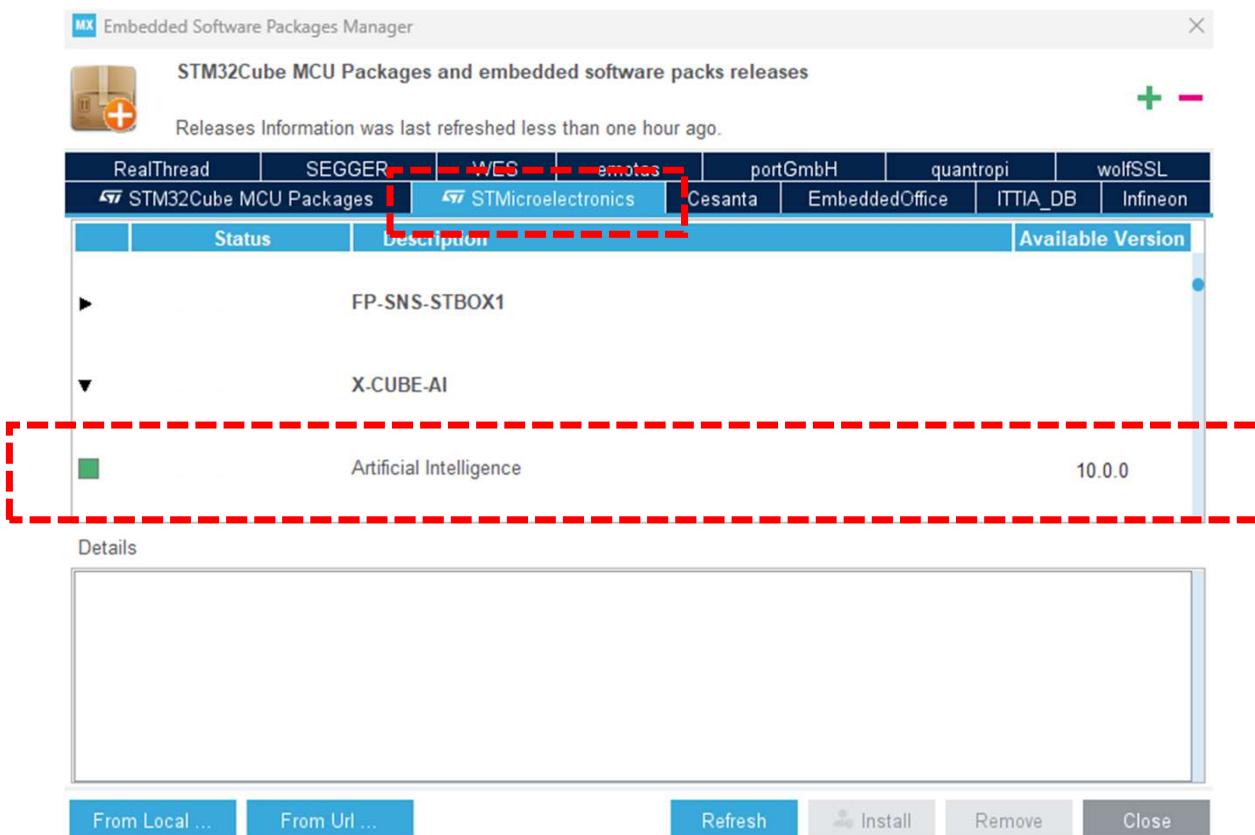
This hands-on uses the STM32CubeN6 Firmware Package V1.1.0.
Install it from CubeMX's Embedded Software Packages Manager.



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



CubeMX Configuration

The screenshot shows the 'New Project' screen of the CubeMX software. On the left, a sidebar lists three options: 'Start My project from MCU', 'Start My project from ST Board', and 'Start My project from Example'. Each option has a blue button labeled 'ACCESS TO [SELECTOR]'. A red arrow points from the 'STM32N657X0H3Q' text at the bottom left towards the 'MCU/MPU Selector' tab in the top navigation bar. The 'MCU/MPU Selector' tab is highlighted with a red circle labeled '1'. Inside the 'MCU/MPU Selector' tab, there is a 'Commercial Part Number' dropdown menu. A red circle labeled '2' highlights the dropdown menu, which contains the text 'STM32N657X0H3Q'. A red dashed box surrounds the dropdown menu. Below the dropdown is a search bar and a '+'/- button. To the right of the dropdown, there is a 'PRODUCT INFO' section with expandable items: Segment, Series, Line, Marketing Status, Price, Package, Core, and Coprocessor. At the bottom of this section is a 'MEMORY' section with a slider set to '0'. On the right side of the interface, there is a large yellow advertisement for the STM32N6 MCU, featuring the ST logo and the text 'The first high-performance STM32 MCU with AI capabilities'. Below the advertisement is a table titled 'MCUs/MPUs List: 1 item'. The table has columns for Commercial Part No., Part No., Reference, Marketing Status, and Unit Price. A red circle labeled '3' highlights the 'Commercial Part No.' column, and a red dashed box surrounds the entire row. A red arrow points from the 'Double Click' text at the bottom right towards the 'Commercial Part No.' column.

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

②

③

Double Click

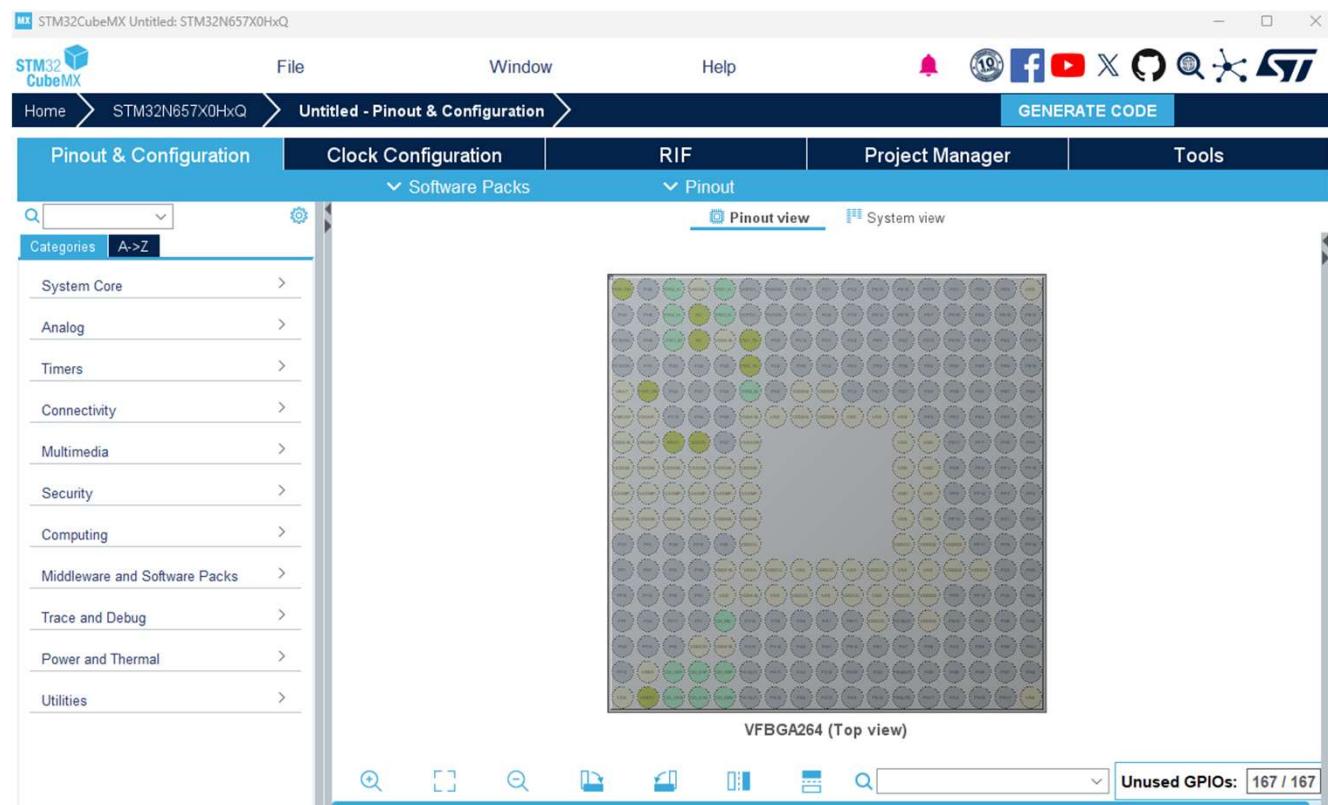
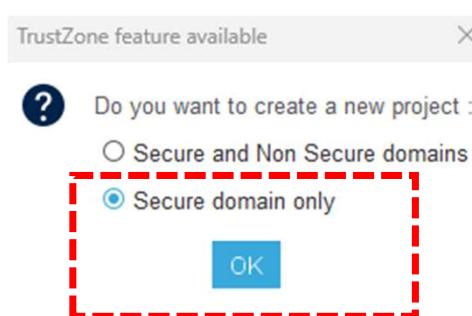
Commercial Part No.	Part No.	Reference	Marketing Status	Unit Price
STM32N657X0H3Q	STM32N657X0	STM32N657X0...	Active	10.7254

https://github.com/BlaineMoon30/STM32_Expert_Training_AI_STM32/tree/master/Hands_On/_1_Hands_On_FSBL_LRUN



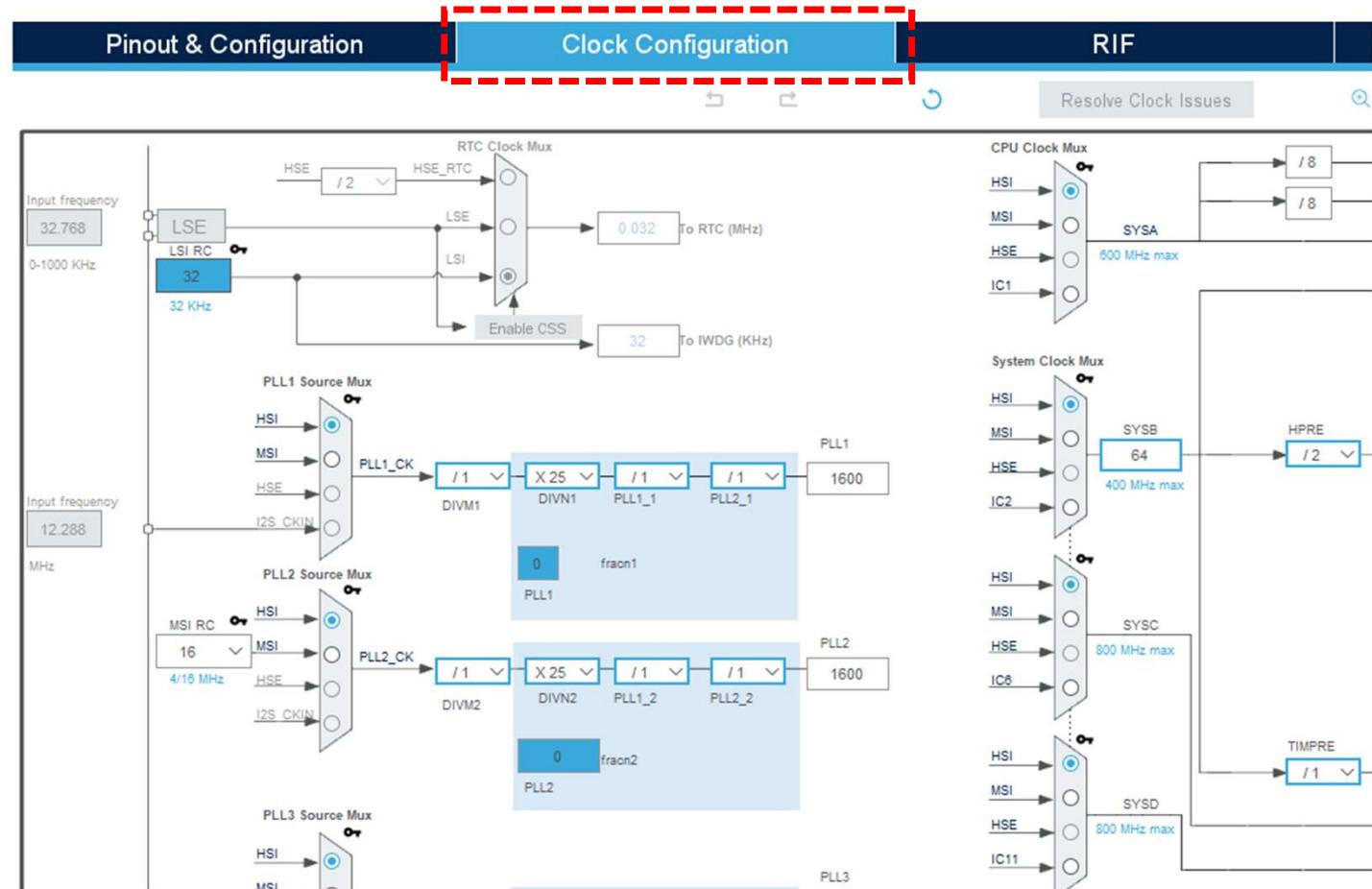
CubeMX Configuration

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



Clock Configuration

Click the **Clock Configuration** tab.



Electrical characteristics

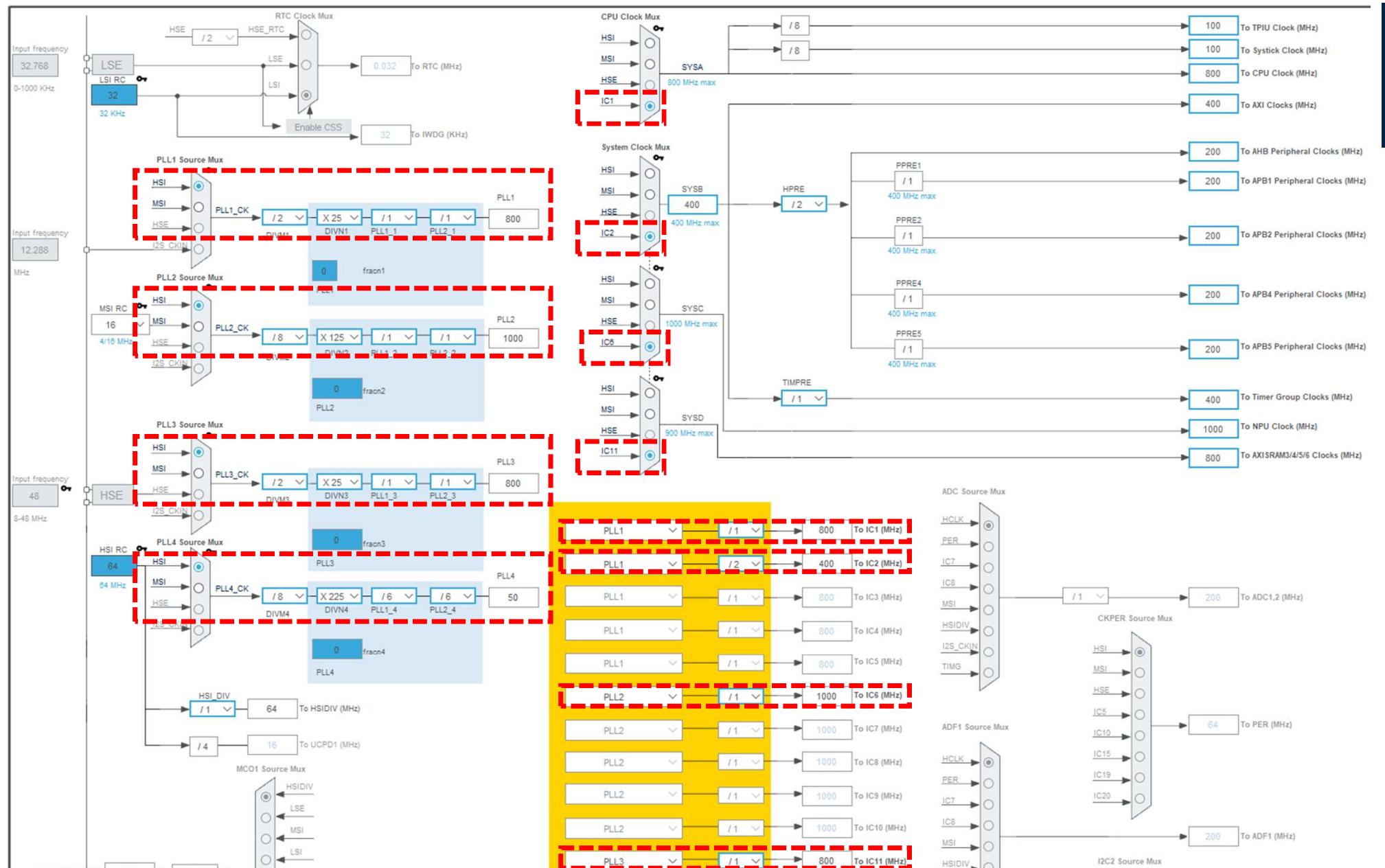
STM32N647xx STM32N657xx

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
$F_{CPU \text{ overdrive}}$	Clock frequency of Cortex-CM55 in overdrive (VOS high)		0	-	800	
F_{NPU}	Clock frequency of NPU/CNN		0	-	800	
$F_{NPU \text{ overdrive}}$	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} ($x = 1, 2, 3, 4, 5$)	Clock frequency of APB buses	-	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 >

System Core	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"/>		
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"/>	<input type="checkbox"/>



RCC Configuration

Pinout & Configuration Clock Configuration RIF

Categories A-Z

System Core

	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"/>		
RCC	<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"/>

Analog

Timers

Connectivity

Multimedia

Software Packs

Pinout

RCC 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"/>

High Speed Clock (HSE) Disable

Low Speed Clock (LSE) Disable

Master Clock Output 1

Master Clock Output 2

Audio Clock Input (I2S_CKIN)

Configuration

Reset Configuration

Parameter Settings User Constants Features

Configure the below parameters :

Search (Ctrl+F)

Power Parameters

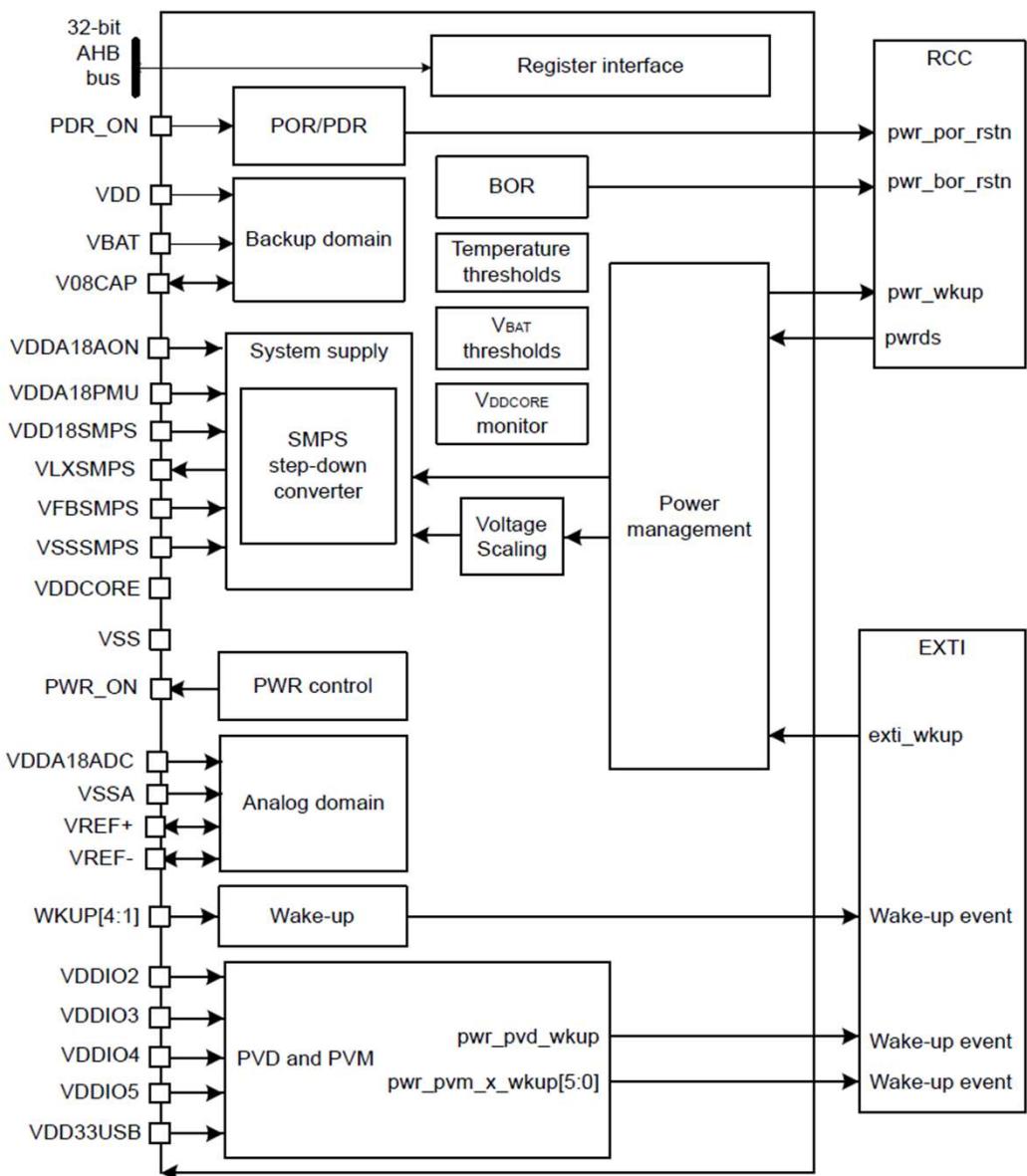
SupplySource	PWR_EXTERNAL_SOURCE_SUPPLY
Power Regulator Voltage Scale	Power Regulator Voltage Scale 0

Spread spectrum mode

PLL1 CSG mode	Disabled
PLL2 CSG mode	Disabled
PLL3 CSG mode	Disabled



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).

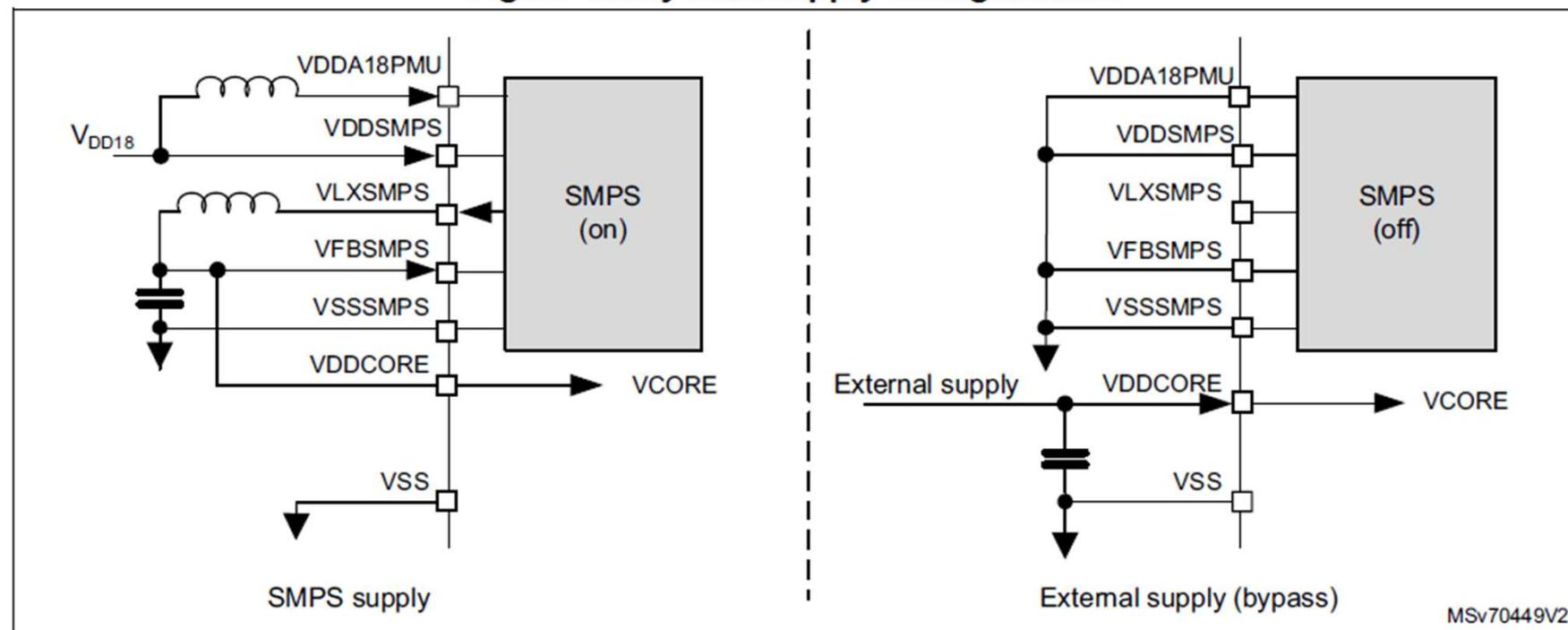
Power supply

RM0486

Power control (PWR)

By configuring the SMPS step-down converter, the supply configurations shown in [Figure 16](#) are supported for the V_{CORE} domain.

Figure 16. System supply configurations

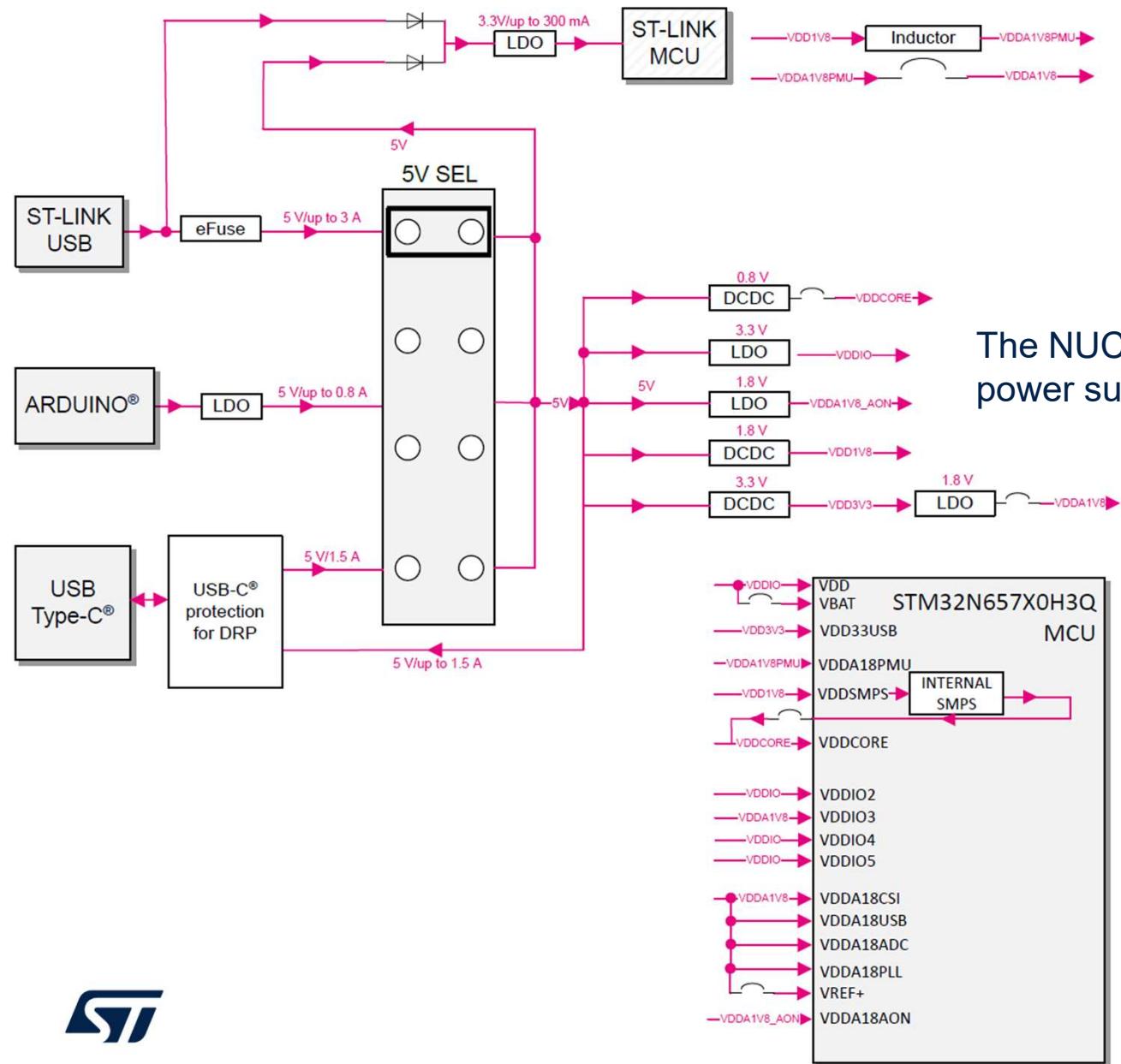


Power supply

$V_{DD}^{(1)}$	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.

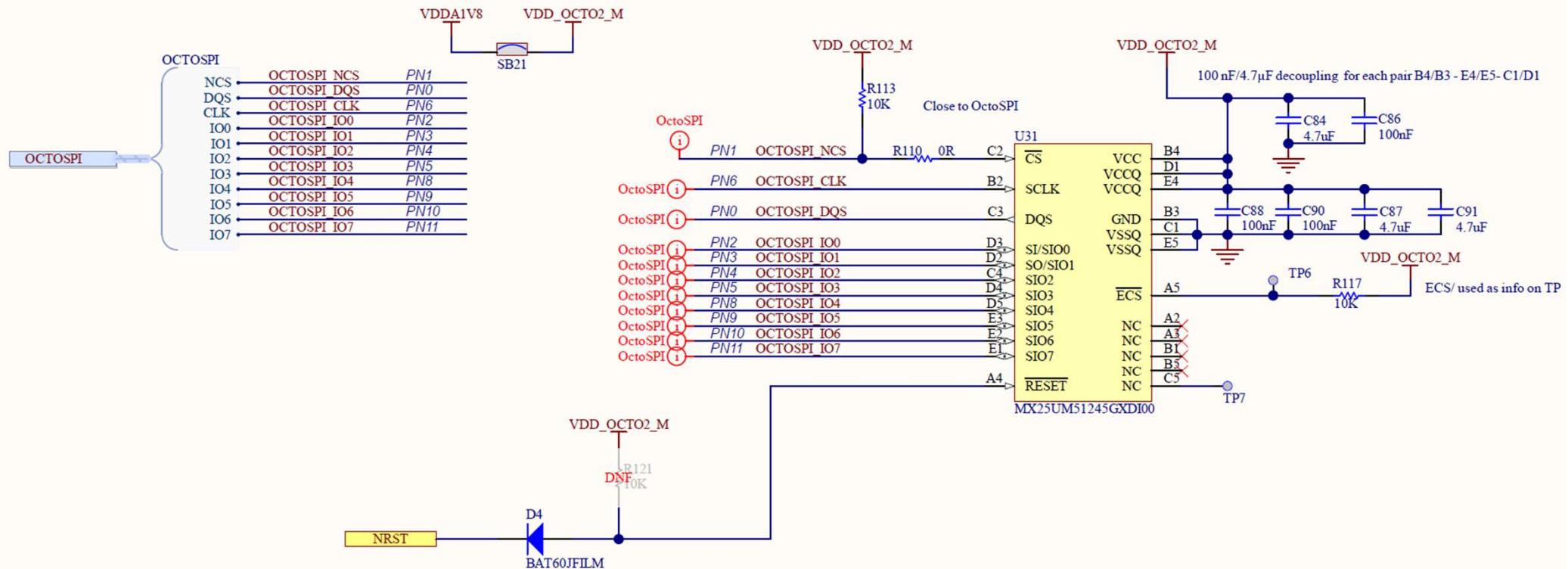


Octo-SPI Flash Memory Configuration

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	L14	T19	T17	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	-	-

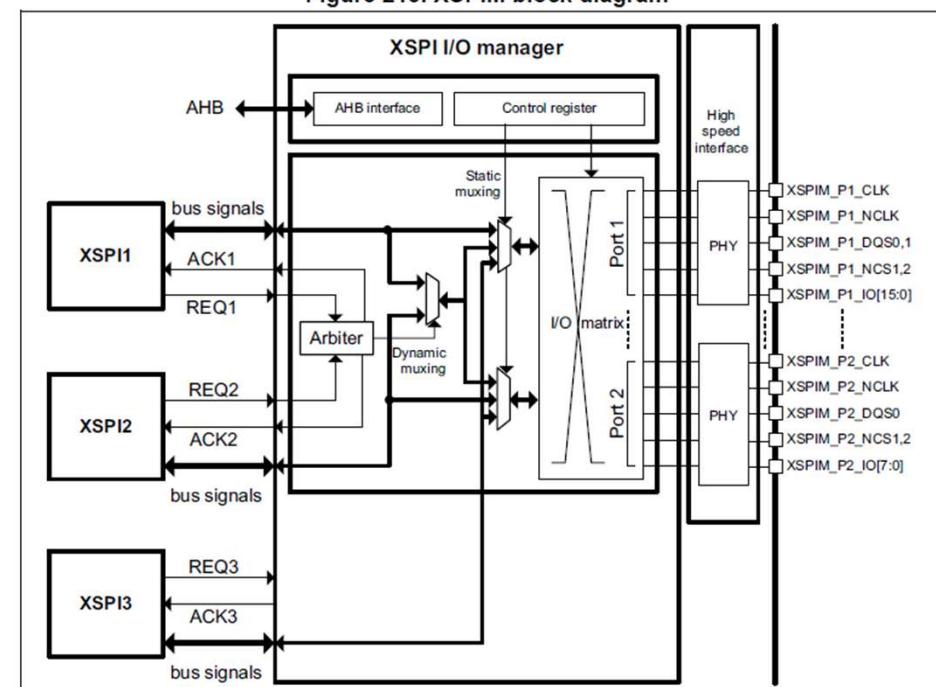
Octo-SPI Flash Memory Configuration

The screenshot shows two overlapping software windows. The top window is titled "Pinout & Configuration" and contains a search bar, a categories dropdown, and a table of pins for various peripherals like UCPD1, USART1-10, USB1_OTG, USB2_OTG, XSPI1-2, and YPD1. A row for "XSPIM" is highlighted with a red dashed box and checked in the first column. The bottom window is titled "Clock Configuration" and shows "XSPIM Mode and Configuration". It includes sections for "Runtime contexts:" (First Stage Boot Loader, Application, External Memory Loader) and a dropdown menu for "Mode" with options: Direct (XSPI1 to Port1; XSPI2 to Port2; XSPI3 not used), 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). The "Direct" option is selected and highlighted with a red dashed box.

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

Software Packs Pinout

XSPI2 Mode and Configuration

Mode

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

Runtime contexts:

Mode Octo SPI
Port Port2 Octo
HyperBus(TM) 1.8V Inverted Clock Disable
Chip Select Override NCS1 -- Port2 --

NVIC Settings DMA Settings GPIO Settings

Parameter Settings User Constants

Configure the below parameters :

Search (Ctrl+F)

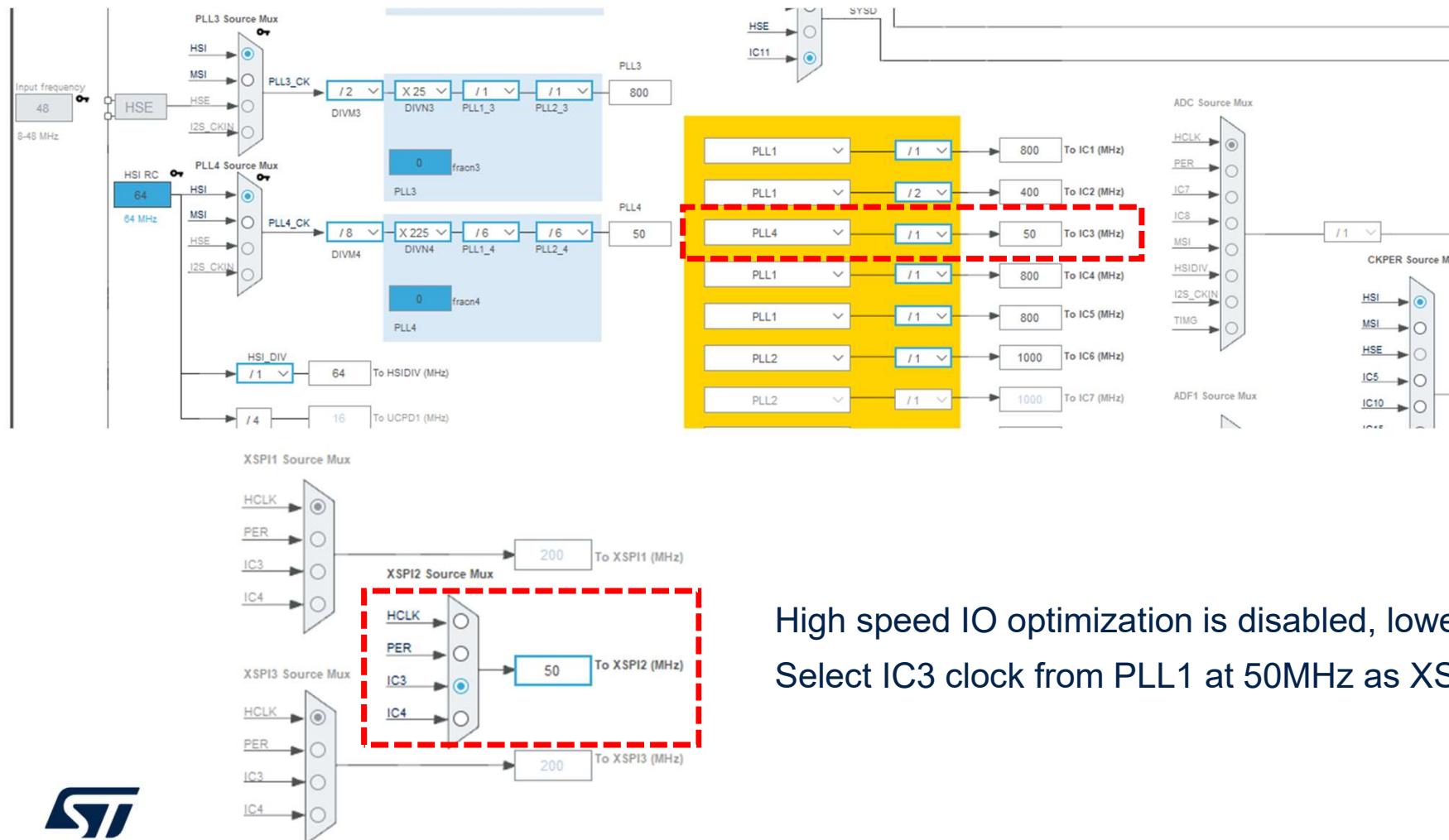
Generic

Fifo Threshold	4
Memory Mode	Disable
Memory Type	Macronix
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

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 interface with the 'OTP MPU' tab selected. The main area displays a table of OTP memory locations. The columns are: Name, Word, Value, Status, Lock, and Description. The 'Lock' column contains checkboxes, with most entries having checked boxes and one entry (OTP124) having an unchecked box. The 'Description' column provides details for each location, such as 'Permanent write lock' for OTP121, OTP122, and OTP123, and various reset and voltage-related descriptions for the other entries.

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

Buttons at the bottom include 'Export', 'Save OTP partition', 'Lock all' (unchecked), '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

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

Select **FSBL** as the Context for **EXTMEM_MANAGER**.

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

Configuration

Reset Configuration

Boot usecase: Memory 1 Memory 2 User Constants

Configure the below parameters :

Search (Ctrl+F) ⏪ ⏩

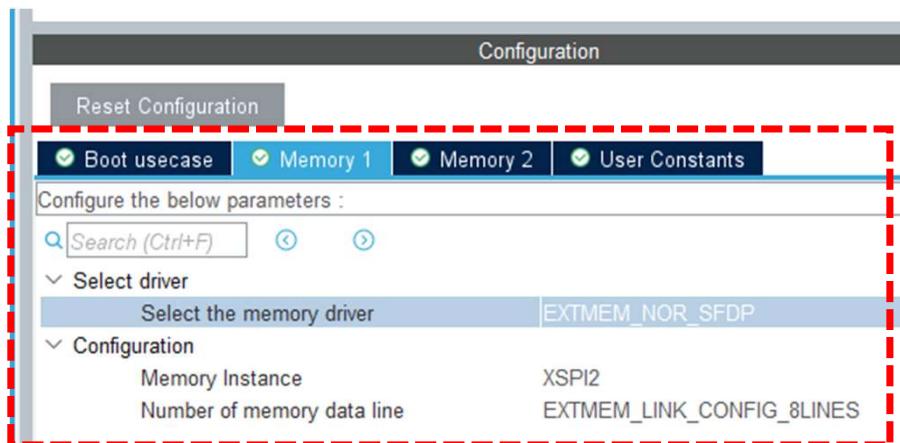
Boot

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

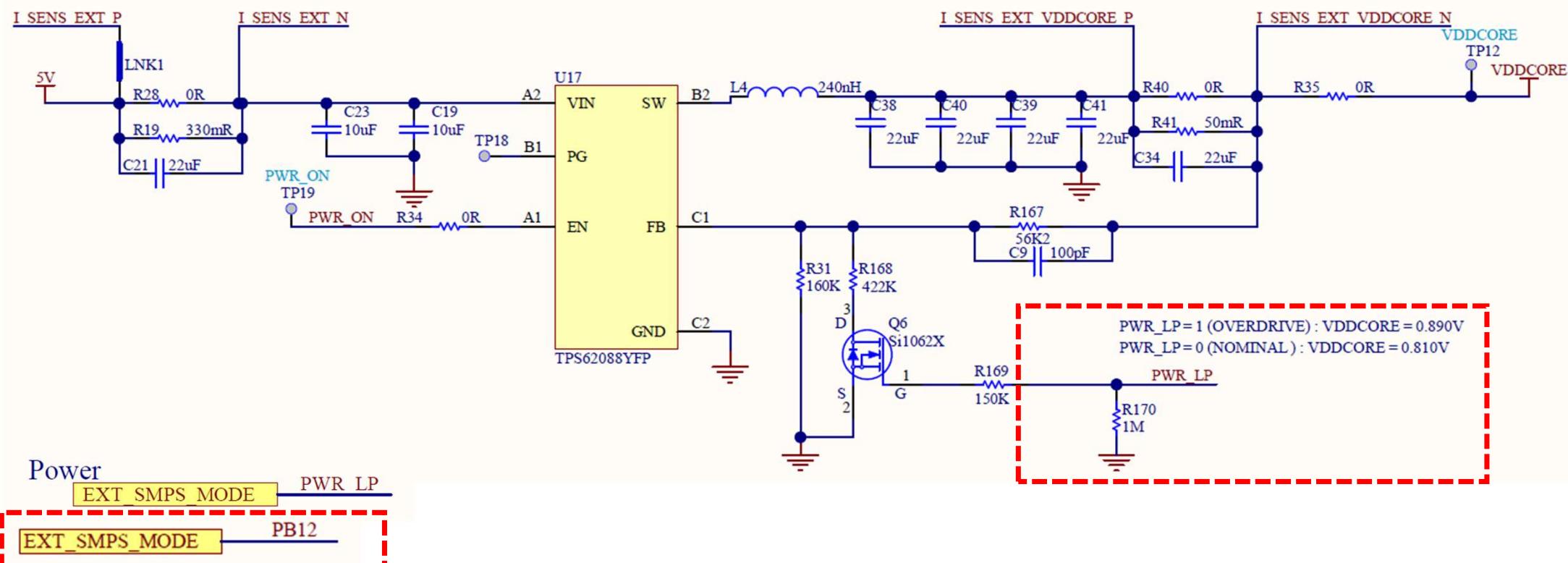


In this hands-on, FSBL loads the **Secure Application** into **AXISRAM1 (0x34000000)**.

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**.



Set PB12 PIN as GPIO_Output.

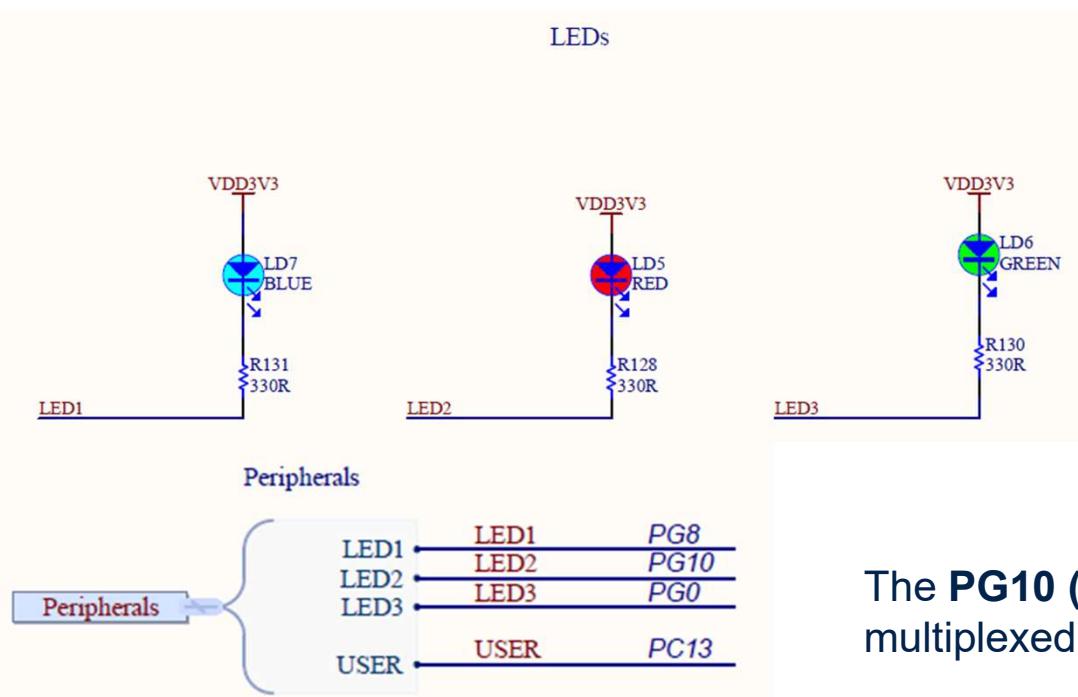
Pin N...	Signal on...	Pin Conte...	Pin Privil...	GPIO out...	GPIO mode	GPIO Pul...	Maximu...	User Label	Modified
PB12	n/a	First Sta...	n/a	High	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG0	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>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

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

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"/>
GPIO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
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"/>		
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"/>	
WWDDCO	<input type="checkbox"/>	<input type="checkbox"/>	

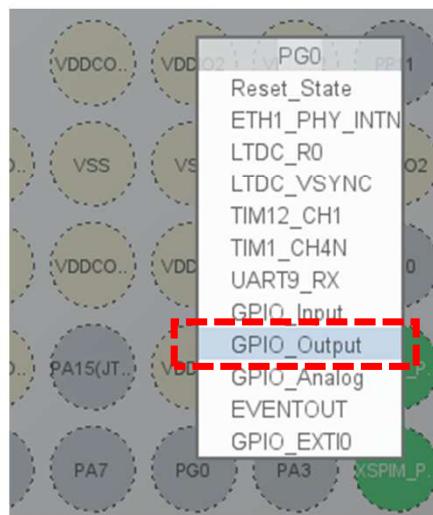
Configuration

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	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	<input type="text" value="Application"/>
GPIO output level	<input type="text" value="Low"/>
GPIO mode	<input type="text" value="Output Push Pull"/>
GPIO Pull-up/Pull-down	<input type="text" value="No pull-up and no pull-down"/>
Maximum output speed	<input type="text" value="Low"/>
User Label	<input type="text"/>



BSEC Configuration

The screenshot shows the Pinout & Configuration interface with the following details:

Pinout & Configuration Tab:

- Search bar:
- Categories: System Core, Analog, Timers, Connectivity, Multimedia, Security.
- Table Headers: FSBL, Application, ExtMemLo.
- Table Rows:
 - BSEC** (highlighted with a red dashed box): FSBL checked, Application checked, ExtMemLo checked.
 - CRYP: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - HASH: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - MCE1: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - MCE2: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - MCE3: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - MCE4: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - PKA: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - RIF: FSBL unchecked, Application checked, ExtMemLo unchecked.
 - RNG: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - SAES: FSBL unchecked, Application unchecked, ExtMemLo unchecked.

Clock Configuration Tab:

BSEC Mode and Configuration

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

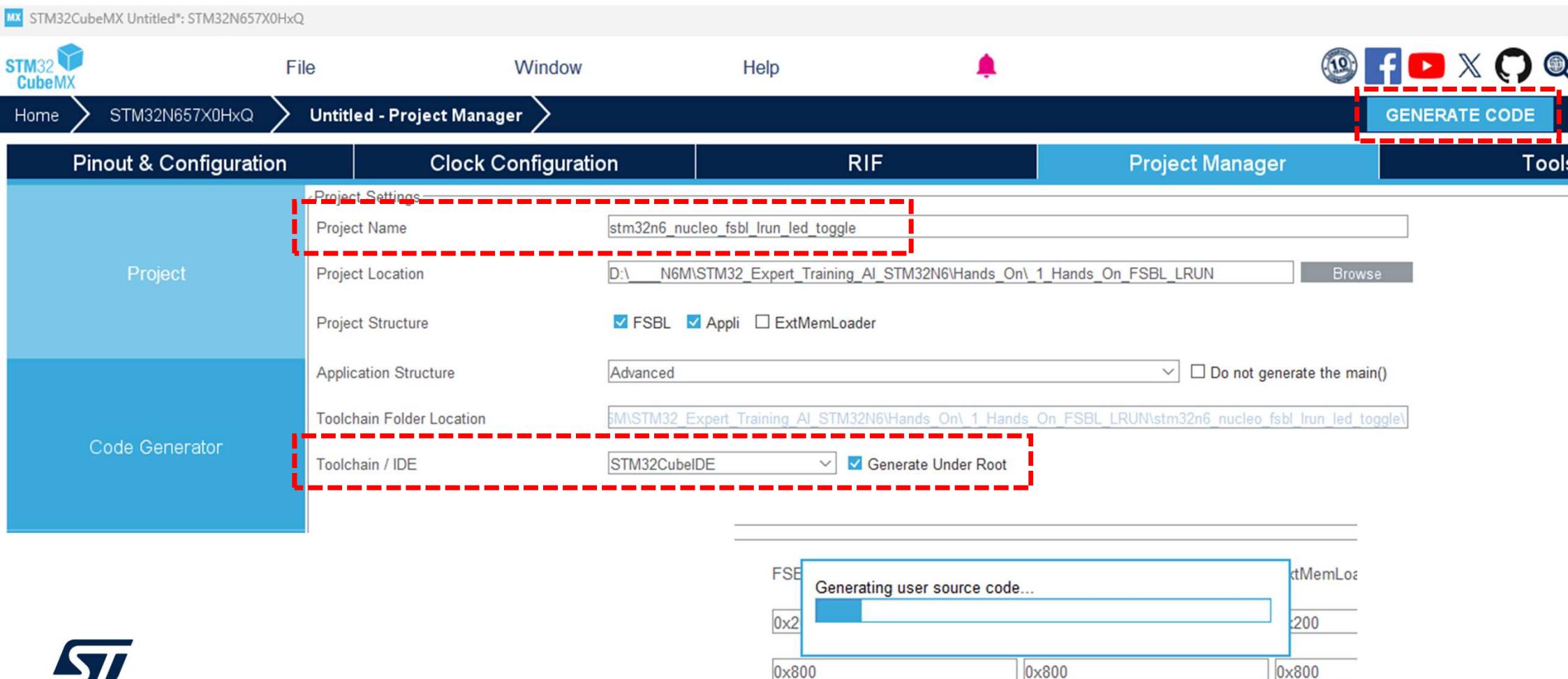
Runtime contexts:

- First Stage Boot Loader (checked)
- Application
- External Memory Loader

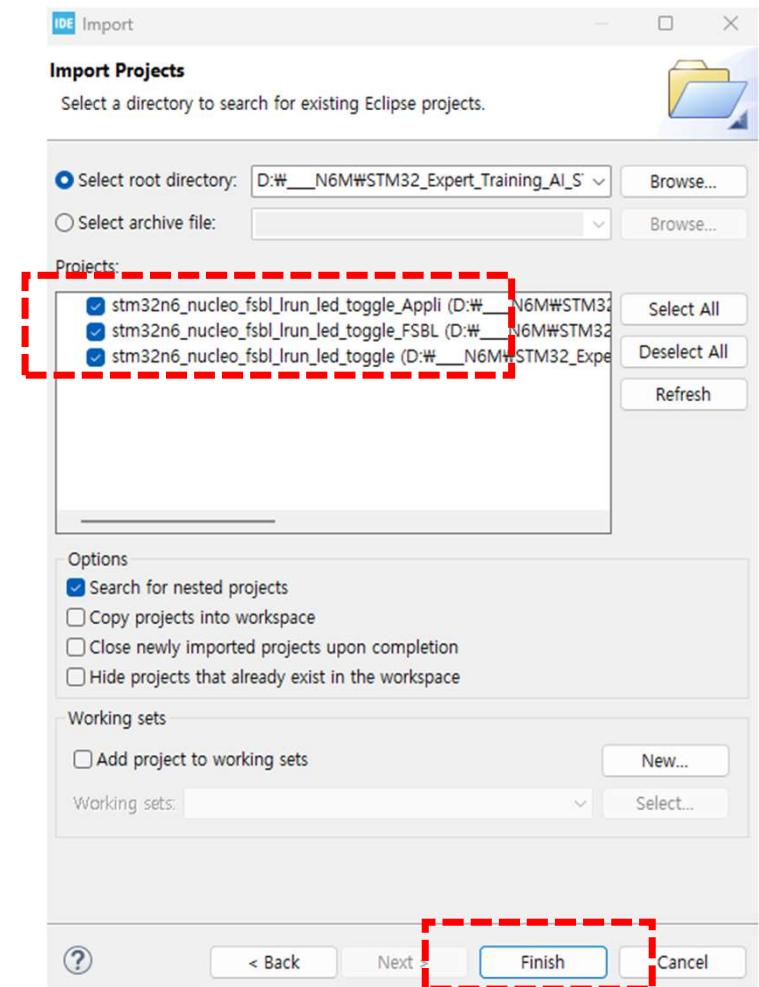
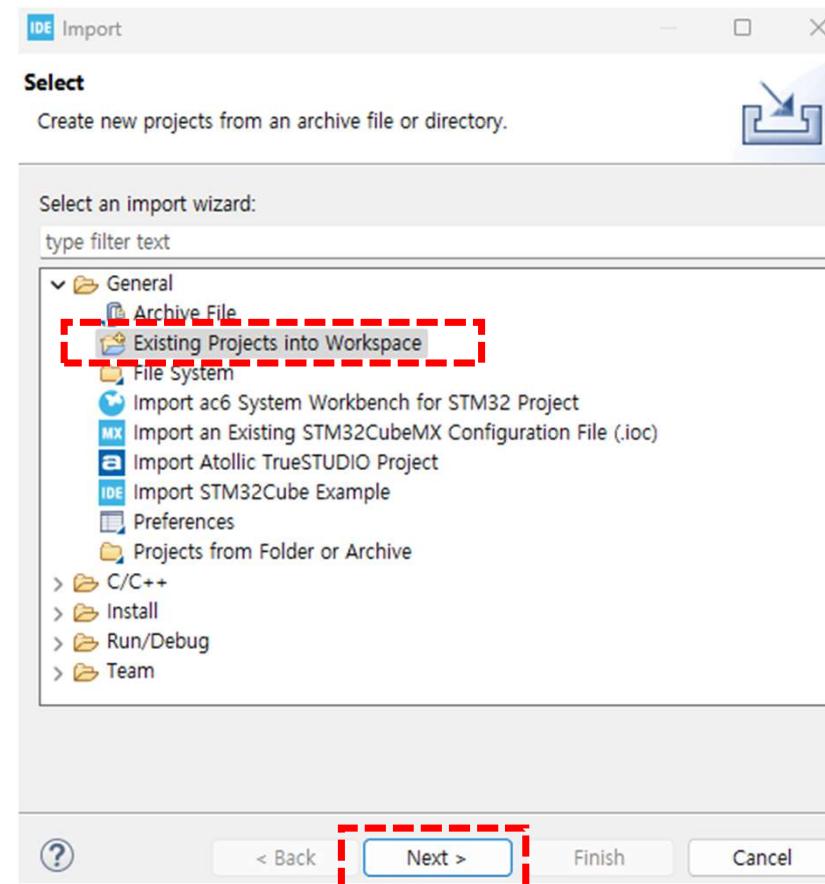
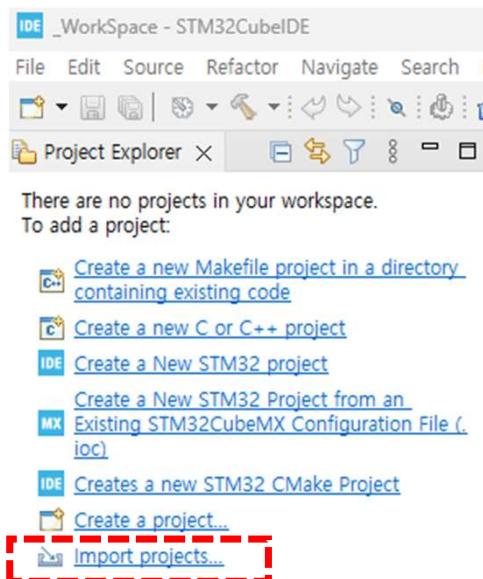
Configuration

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

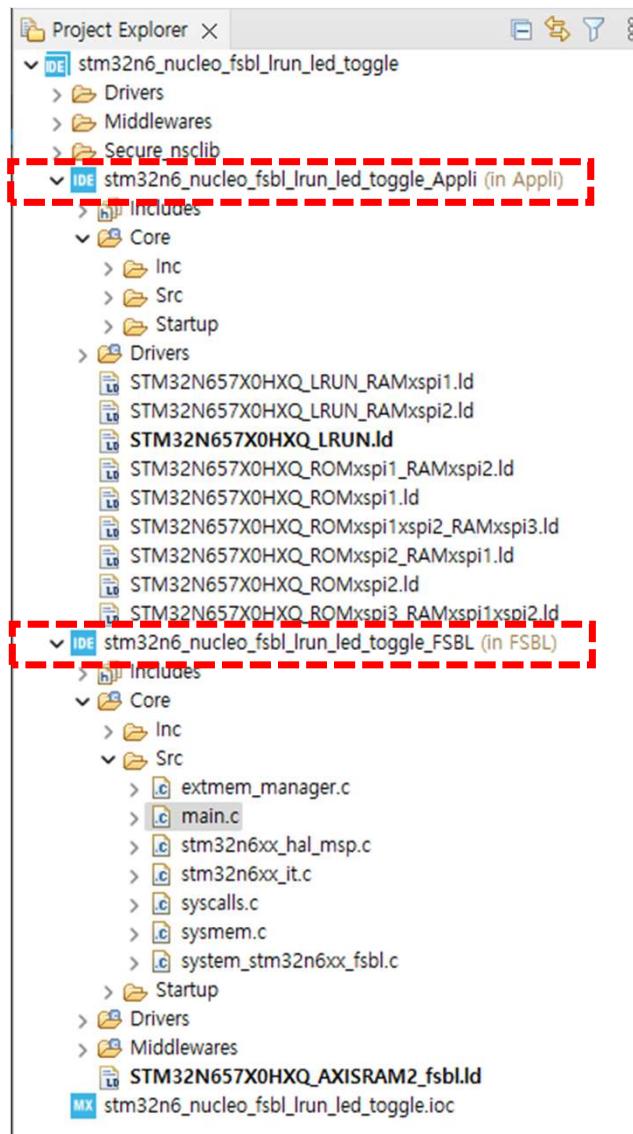
GENERATE CODE



STM32CubeIDE Import Project



STM32CubeIDE Import Project



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



Add Code to FSBL

The screenshot shows the STM32CubeIDE interface. The Project Explorer on the left lists the project structure, including subfolders like Drivers, Middlewares, and Core, along with source files such as extmem_manager.c, main.c, and system_stm32n6xx_fsbl.c. The main.c file is open in the editor on the right, displaying initialization code. A red box highlights the line `MX_USART1_Init();`, which is part of the peripheral initialization block. The code also includes comments for enabling CPU cache, I-Cache, D-Cache, and MCU configuration, as well as sections for USER CODE BEGIN and END.

```
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_USART1_Init();
101 MX_USART1_Init();
102 MX_XSPI2_Init();
103 MX_EXTMEM_MANAGER_Init();
104
105 /* USER CODE BEGIN 2 */
106
107 /* USER CODE END 2 */
```

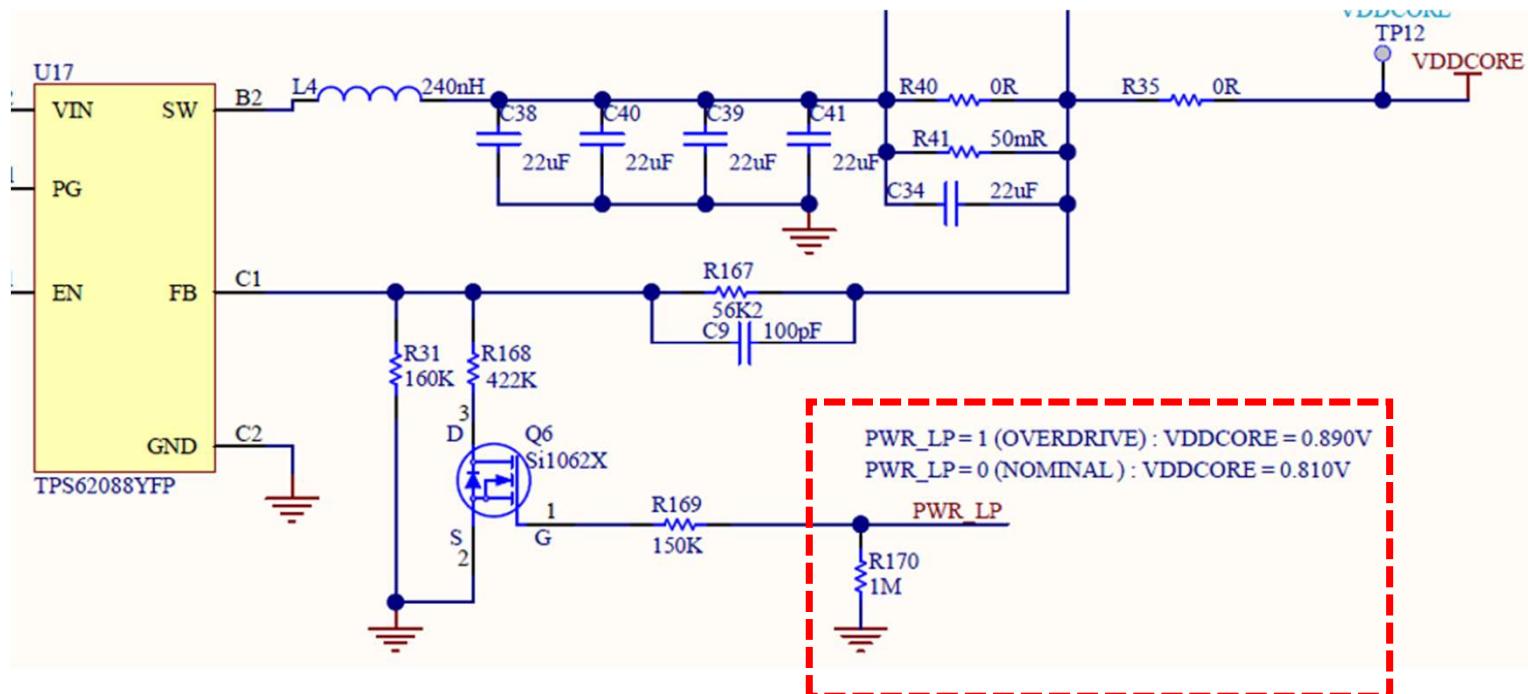
```
79 SCB_EnableICache();
80
81 /* Enable D-Cache-----*/
82 SCB_EnableDCache();
83
84 /* MCU Configuration-----*/
85 HAL_Init();
86
87 /* USER CODE BEGIN Init */
88
89 MX_GPIO_Init();
90 HAL_Delay(1);
91
92 /* USER CODE END Init */
93
94 /* Configure the system clock */
95 SystemClock_Config();
96
97 /* 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 */
109
```

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 */  
311static 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_GPION_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 the project structure under 'stm32n6_nucleo_fsbl_lrun_led_toggle'. A red dashed box highlights the file 'stm32n6xx_hal_msp.c' in the 'Src' folder. The main window shows the code for 'stm32n6xx_hal_msp.c'. The code includes comments for user-defined code sections and defines two specific macros:

```
19 /* USER CODE END Header */
20
21 /* Includes -----*/
22 #include "main.h"
23 /* USER CODE BEGIN Includes */
24
25 /* USER CODE END Includes */
26
27/* Private typedef -----*/
28 /* USER CODE BEGIN TD */
29
30 /* USER CODE END TD */
31
32/* Private define -----*/
33 /* USER CODE BEGIN Define */
34
35 #define HSLV OTP 124
36 #define VDDIO3_HSLV_MASK (1<<15)
37
38 /* USER CODE END Define */
39
40/* Private macro -----*/
41 /* USER CODE BEGIN Macro */
42
43 /* USER CODE END Macro */
44
45/* Private variables -----*/
46 /* USER CODE BEGIN PV */
47
48 /* USER CODE END PV */
49
50/* Private function prototypes -----*/
```

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 STM32 Nucleo IDE interface. On the left, the Project Explorer displays the project structure for 'stm32n6_nucleo_fsbl_lrun_led_toggle'. The 'Core' folder contains source files like 'main.c', 'extmem_manager.c', 'stm32n6xx_hal_msp.c' (which is currently selected and highlighted with a red dashed box), 'stm32n6xx_it.c', 'syscalls.c', 'sysmem.c', and 'system_stm32n6xx_fsbl.c'. Other folders include 'Drivers', 'Middlewares', and 'Debug'.

The code editor on the right shows the content of 'stm32n6xx_hal_msp.c'. A red dashed box highlights the main body of the function 'HAL_XSPI_MspInit'. The code initializes GPIO and RCC, reads BSEC fuse data, configures PWR and SYSCFG, and enables the XSPI2 clock. The entire function body is enclosed in a red dashed box.

```
88@ void HAL_XSPI_MspInit(XSPI_HandleTypeDef* hxspi)
89 {
90     GPIO_InitTypeDef GPIO_InitStruct = {0};
91     RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};
92     if(hxspi->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".
- Editor:** Shows the file `STM32N657X0HXQ_AXISRAM2_fsbl.id` containing assembly code for the FSBL.
- CDT Build Console:** Displays the build log output.

CDT Build Console Output:

```
18:17:17 **** Incremental Build of configuration Debug for project stm32n6_nucleo_fsbl_lrun_led_toggle_FSB
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_FSB.elf" "@objects.list" -mcpu=cortex-
Finished building target: stm32n6_nucleo_fsbl_lrun_led_toggle_FSB.elf

arm-none-eabi-size stm32n6_nucleo_fsbl_lrun_led_toggle_FSB.elf
arm-none-eabi-objdump -h -S stm32n6_nucleo_fsbl_lrun_led_toggle_FSB.elf > "stm32n6_nucleo_fsbl_lr
text      data      bss      dec      hex filename
61020      12    3268   64300    fb2c stm32n6_nucleo_fsbl_lrun_led_toggle_FSB.elf
arm-none-eabi-objcopy -O binary stm32n6_nucleo_fsbl_lrun_led_toggle_FSB.elf "stm32n6_nucleo_fsbl_
Finished building: default.size.stdout

Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_FSB.elf

Finished building: stm32n6_nucleo_fsbl_lrun_led_toggle_FSB.list

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



Debug



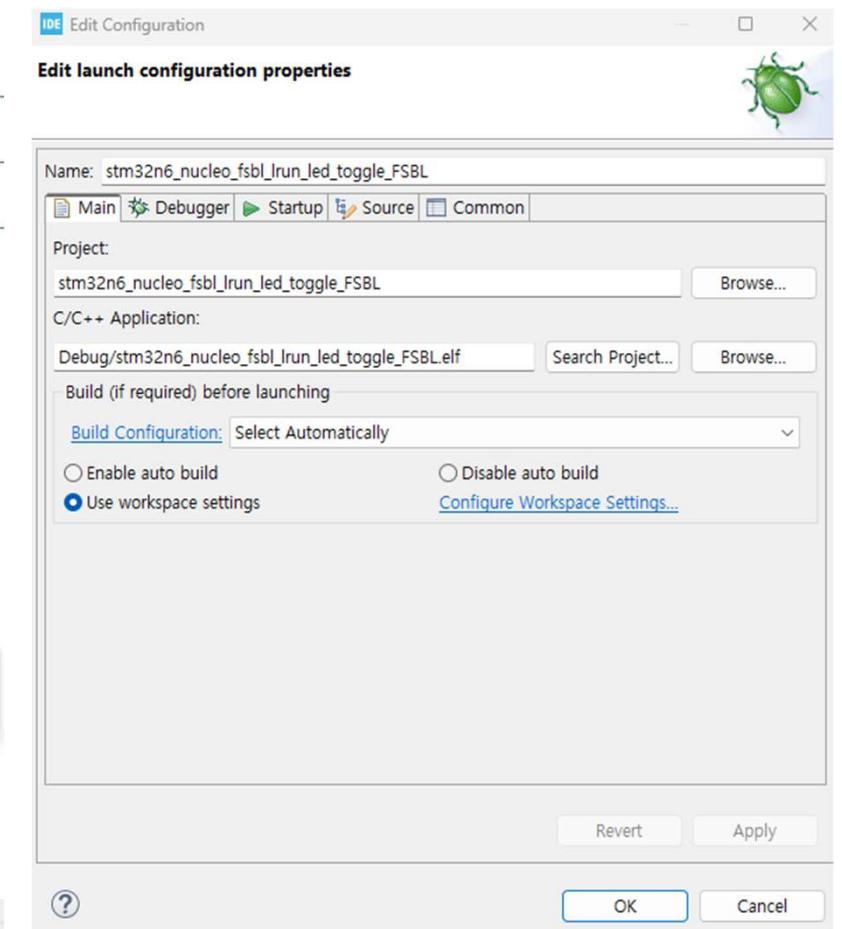
Project Explorer X

- IDE stm32n6_nucleo_fsbl_lrun_led_toggle
 - Drivers
 - Middlewares
 - Secure_nsclib
 - IDE stm32n6_nucleo_fsbl_lrun_led_toggle_App (in Appli)
 - IDE stm32n6_nucleo_fsbl_lrun_led_t
 - Includes
 - Core
 - Inc
 - Src
 - extmem_manager.c
 - main.c
 - stm32n6xx_hal_msp.c
 - stm32n6xx_it.c
 - syscalls.c
 - sysmem.c
 - system_stm32n6xx_fsb
 - Startup
 - Drivers
 - Middlewares
 - STM32N657X0HXQ_AXIISR
 - stm32n6_nucleo_fsbl_lrun_led_t

main.c X

```
63 /* USER CODE END 0 */  
64  
65/**  
66 * @brief  The application entry point.  
67 * @retval int  
68 */
```

- New
- Go Into
- Open in New Window
- Show In Alt+Shift+W >
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Delete
- Source
- Move...
- Rename... F2
- Import...
- Export...
- Build Project
- Clean Project
- Refresh F5
- Close Project
- Close Unrelated Projects
- Build Configurations
- Build Targets
- Index
- Run As
- Debug As
 - Team
 - Compare With
 - Restore from Local History...
- Generate Code
- Convert to C++
- Run C/C++ Code Analysis
- Configure
- Properties Alt+Enter

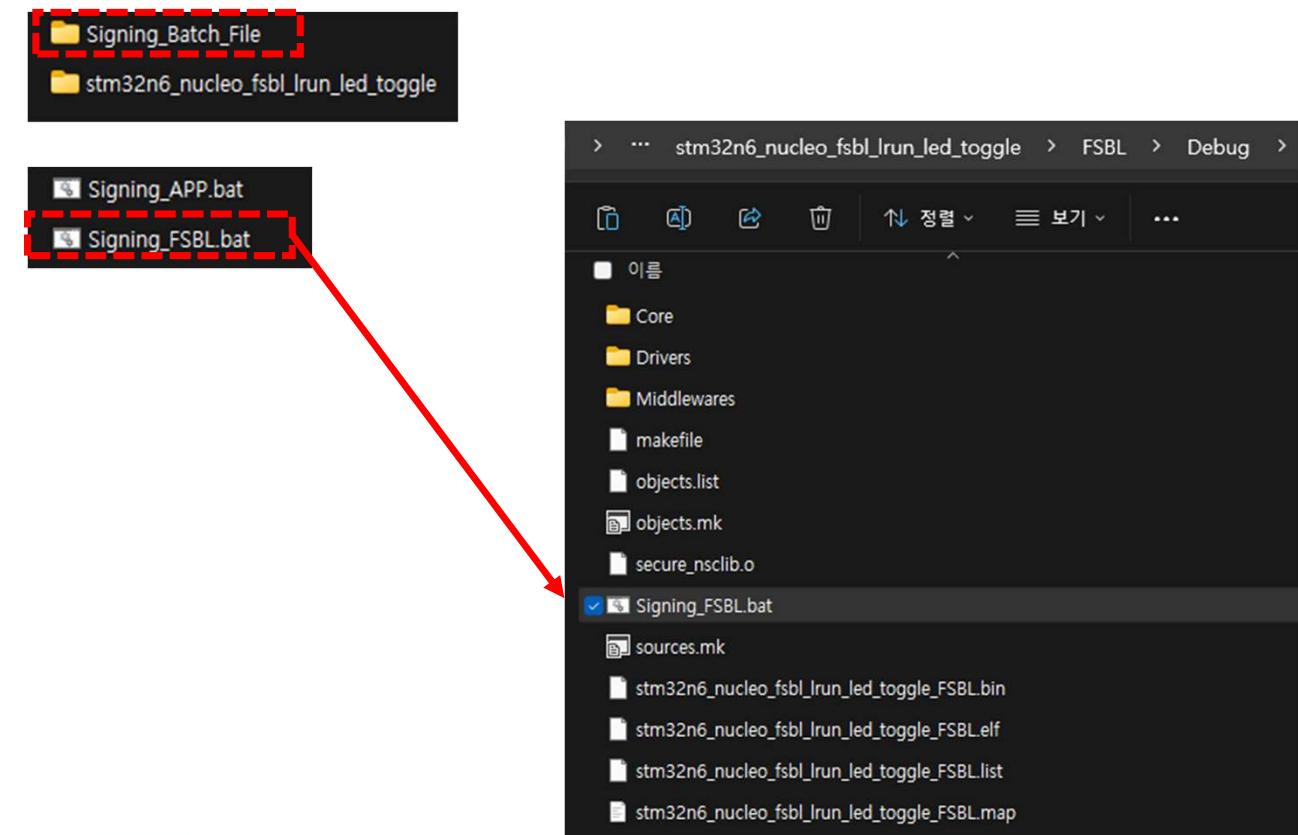


FSBL Header



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

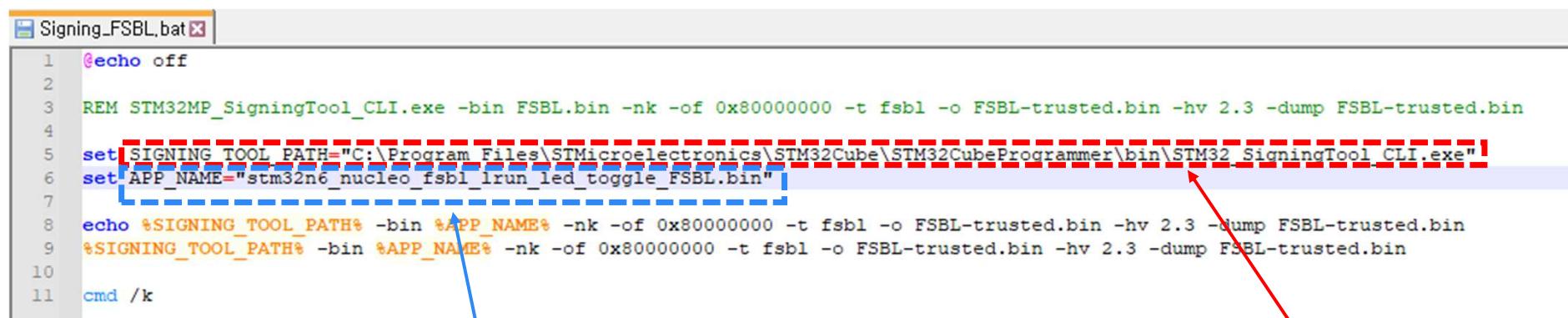
FSBL SIGNING



Copy the **Signing_FSBL.bat** file from the **Siging_Batch_File** folder into the **FSBL\Debug** folder of the current project.



FSBL SIGNING



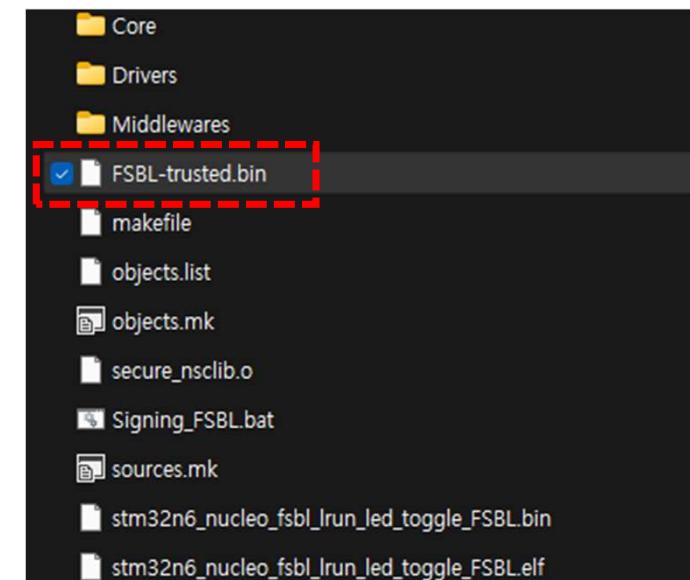
```
1 @echo off
2
3 REM STM32MP_SigningTool_CLI.exe -bin FSBL.bin -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
4
5 set SIGNING_TOOL_PATH="C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe"
6 set APP_NAME="stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin"
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'. Inside the application folder, there are 'Includes', 'Core' (containing 'Inc'), and 'Src' (containing 'main.c', 'secure_nsclib.c', 'stm32n6xx_hal_msp.c', 'stm32n6xx_it.c', 'syscalls.c', 'sysmem.c', 'system_stm32n6xx_s.c', and 'Startup'). The 'Drivers' folder contains several ICD files for different memory configurations. At the bottom, there are entries for 'stm32n6_nucleo_fsbl_lrun_led_toggle_FSB' and 'stm32n6_nucleo_fsbl_lrun_led_toggle.ioc'. The right side of the interface shows the code editor for 'main.c'. The code is a template for a secure application, starting with HAL_Init() and ending with HAL_Delay(200). It includes sections for USER CODE BEGIN and END for Init, SysInit, and a WHILE loop. A blue selection bar highlights the code from line 88 to line 103, which defines the infinite loop body.

```
72     HAL_Init();
73
74     /* USER CODE BEGIN Init */
75
76     /* USER CODE END Init */
77
78     /* USER CODE BEGIN SysInit */
79
80     /* USER CODE END SysInit */
81
82     /* Initialize all configured peripherals */
83     MX_GPIO_Init();
84     SystemIsolation_Config();
85     /* USER CODE BEGIN 2 */
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.



Build APP

The screenshot shows the STCubeIDE interface during the build process of an STM32 application. The left pane displays the Project Explorer with several source files and linker scripts under the main application project. The central pane shows the code editor for the `main.c` file, which includes memory definitions and startup code sections. The right pane shows the build console output, detailing the compilation command, assembly statistics, and the successful completion of the build.

Project Explorer:

- stm32n6_nucleo_fsbl_lrun_led_toggle
- 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_FSBL (in FSBL)
- 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:

```
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 -DD
arm-none-eabi-gcc ".../Core/Src/syscalls.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDEBUG
arm-none-eabi-gcc ".../Core/Src/sysmem.c" -mcpu=cortex-m55 -std=gnu11 -g3 -DDEBUG
arm-none-eabi-gcc ".../Core/Src/system_stm32n6xx_s.c" -mcpu=cortex-m55 -std=gnu11 -
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.externalto
Finished building target: stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf

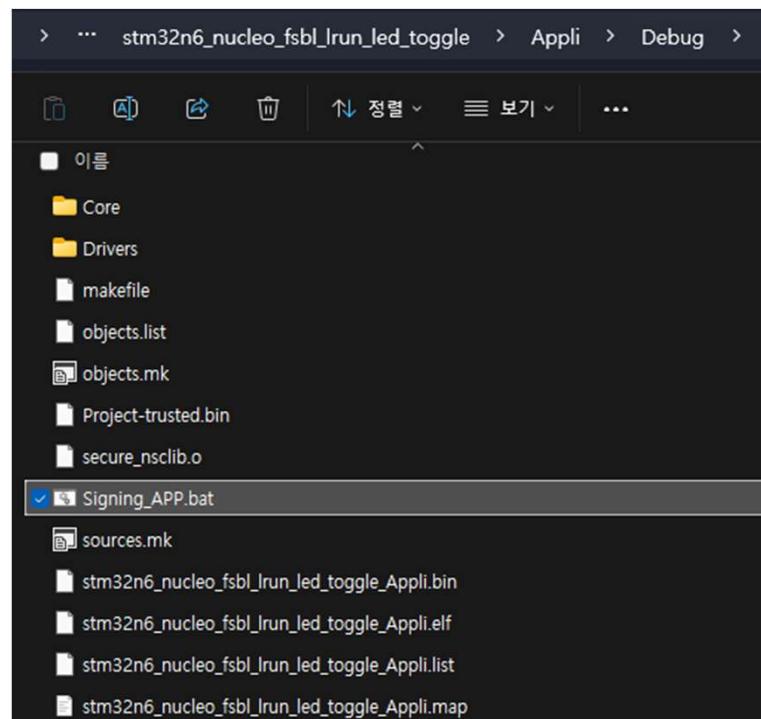
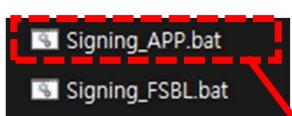
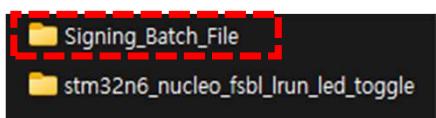
arm-none-eabi-size stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf
arm-none-eabi-objdump -h -S stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf > "stm3
    text      data      bss      dec      hex filename
        4656       12     2604     7272   1c68 stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.
arm-none-eabi-objcopy -O binary stm32n6_nucleo_fsbl_lrun_led_toggle_Appli.elf "s
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_FSBBL.bat** file from the **Siging_Batch_File** folder into the **FSBL\Debug** folder of the current project.



APP SIGNING

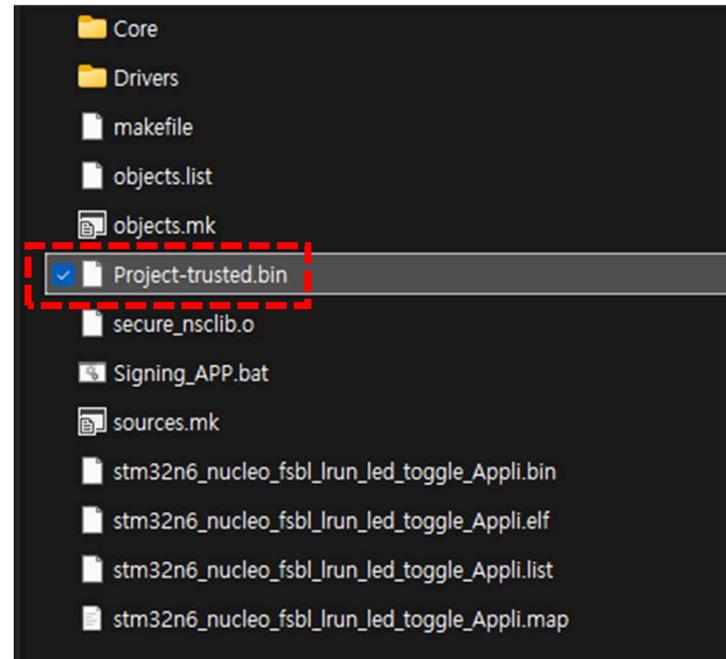
```
 Signing_APP.bat
1  @echo off
2
3  REM STM32MP_SigningTool_CLI.exe -bin Project.bin -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -dump Project-trusted.bin
4
5  set SIGNED_TOOLS_PATH="C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe"
6  set APP_NAME="stm32n6_nucleo_fsbl_lrun_led_toggle_Applic.bin"
7
8  echo %SIGNED_TOOLS_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -dump Project-trusted.bin
9  %SIGNED_TOOLS_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**.



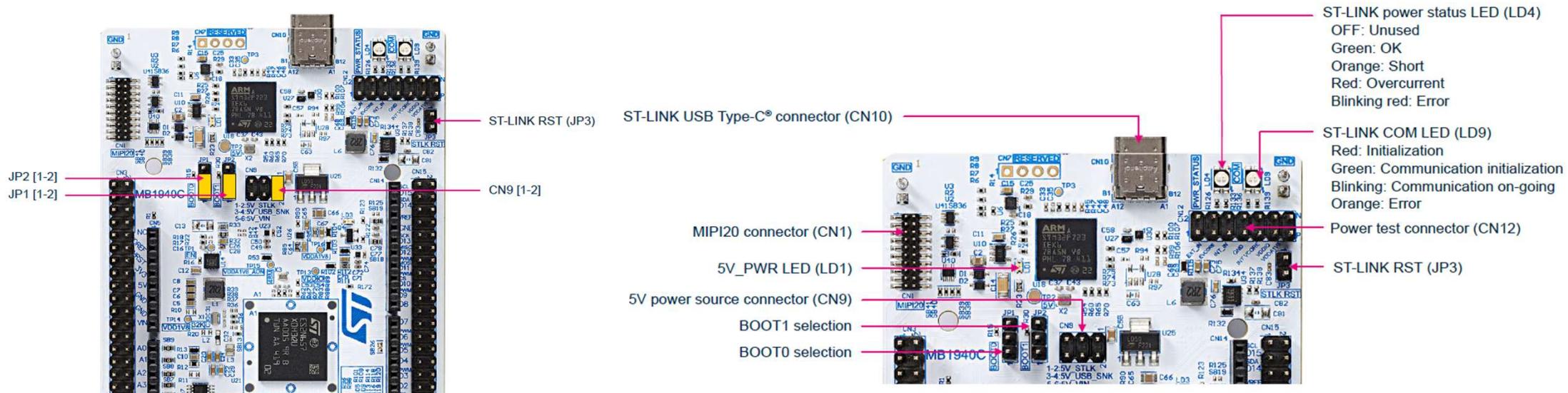
APP SIGNING



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

The screenshot shows the STM32CubeProgrammer interface for an 'Erasing & Programming' session. The left panel includes fields for 'File path' (C:_Project\N6\Training\stm32n6_nucleo_fsbl_lrun_led_toggle_t), 'Start address' (0x70000000), and various checkboxes for programming options. The center panel displays an 'Erase flash memory' dialog with a table of sectors to be erased, starting at address 0x70000000. The right panel shows 'ST-LINK configuration' (Serial number: 0024003832...) and a file browser window. The file browser lists several files under 'Middlewares': 'FSBL-trusted.bin' (selected with a red dashed box), 'makefile', 'objects.list', 'objects.mk', 'secure_nsclib.o', 'Signing_FSBL.bat', 'sources.mk', 'stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin', and 'stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf'. The log window at the bottom shows the command-line output of the download process.

STM32CubeProgrammer

Erasing & Programming

Download

File path: C:_Project\N6\Training\stm32n6_nucleo_fsbl_lrun_led_toggle_t

Start address: 0x70000000

Skip flash erase before programming

Verify programming

Full Flash memory checksum

Run after programming

Automatic Mode

Full chip erase

Download file (checked)

Option bytes commands

-ob

Log

```
18:13:17 : Erasing external memory sector 0
18:13:17 : Download in Progress:
18:13:18 : File download complete
18:13:18 : Time elapsed during download operation: 00:00:00.856
18:13:29 : Opening and parsing file: Project-trusted.bin
18:13:29 : Memory Programming ...
18:13:29 : File : Project-trusted.bin
18:13:29 : Size : 5.59 KB
18:13:29 : Address : 0x70020000
18:13:29 : Erasing memory corresponding to sector 0:
18:13:29 : Erasing external memory sector 2
18:13:30 : Download in Progress:
18:13:30 : File download complete
18:13:30 : Time elapsed during download operation: 00:00:00.359
18:13:32 : Disconnected from device.
```

100%

ST-LINK Connect

ST-LINK configuration

- Serial number: 0024003832...
- Port: SWD
- Frequency (kHz): 8000
- Mode: Hot plug
- Access port: 0
- Reset mode: Hardware reset
- Speed: Reliable
- Shared: Disabled

Debug in Low Power mode: MX25UM51245G_STMR2N

External loader: MX25UM51245G_STMR2N

Target voltage: 3.27 V

Firmware version: V3.15M6

Core

Drivers

Middlewares

FSBL-trusted.bin (checked)

makefile

objects.list

objects.mk

secure_nsclib.o

Signing_FSBL.bat

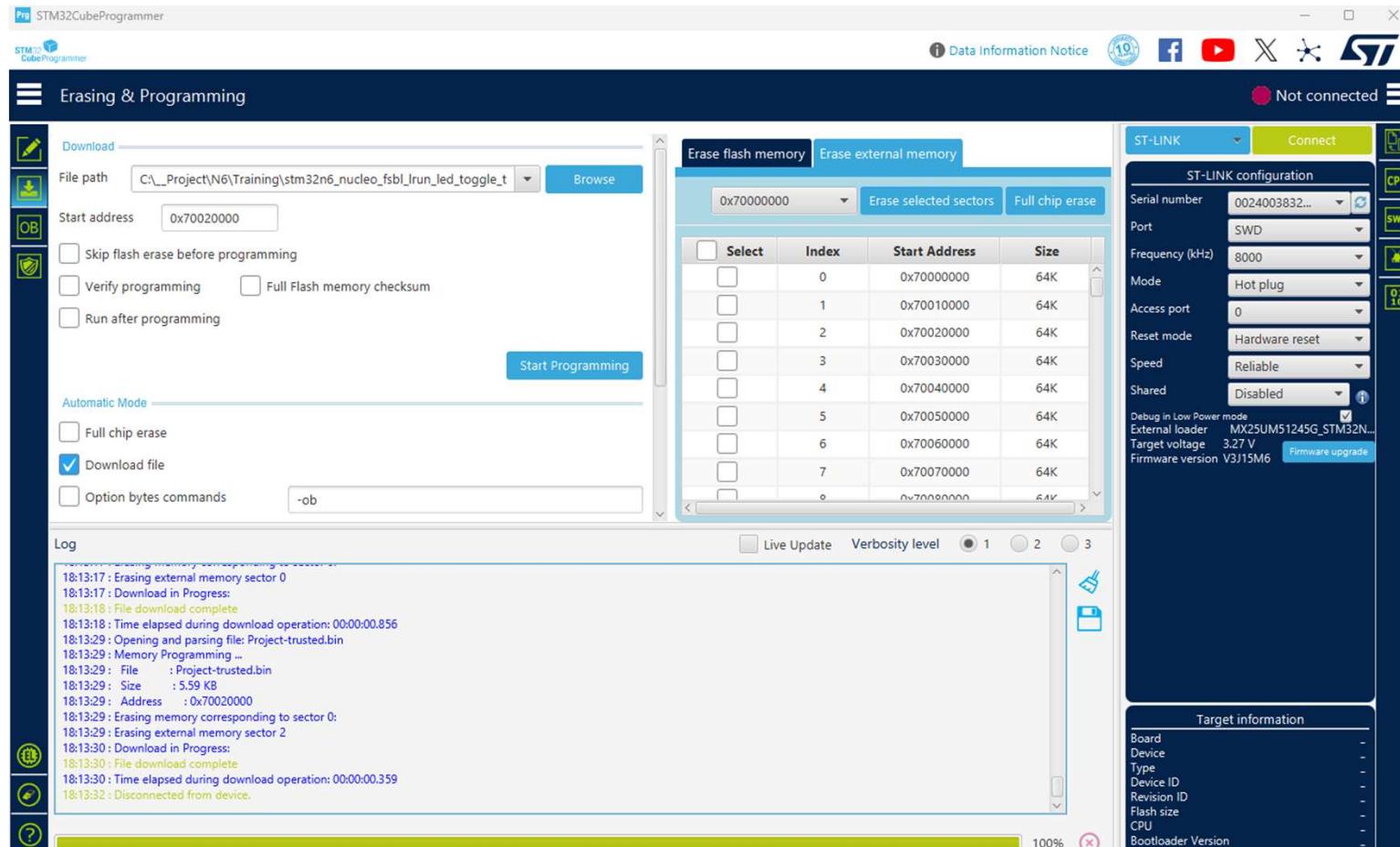
sources.mk

stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.bin

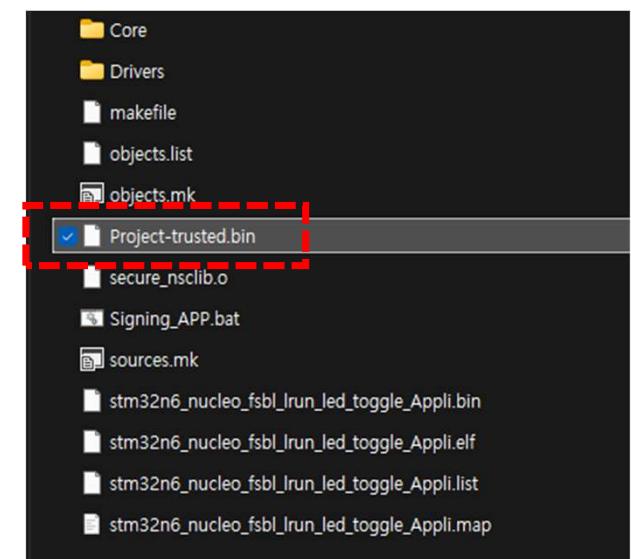
stm32n6_nucleo_fsbl_lrun_led_toggle_FSBL.elf

Write the **FSBL-trusted.bin** file to **0x70000000**.

APP Download



Write the **Project-trusted.bin** file to **0x70020000**.



LRUN Debug

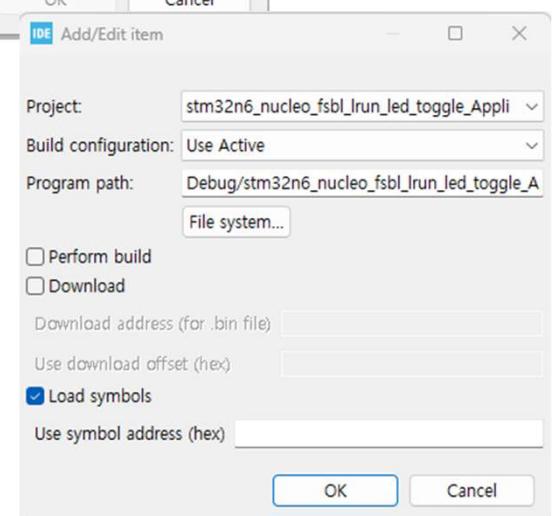
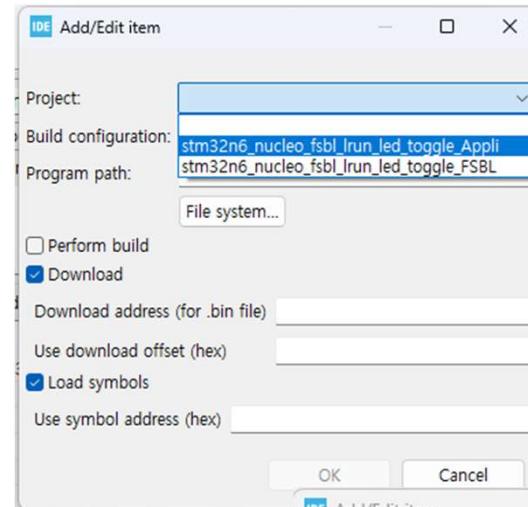
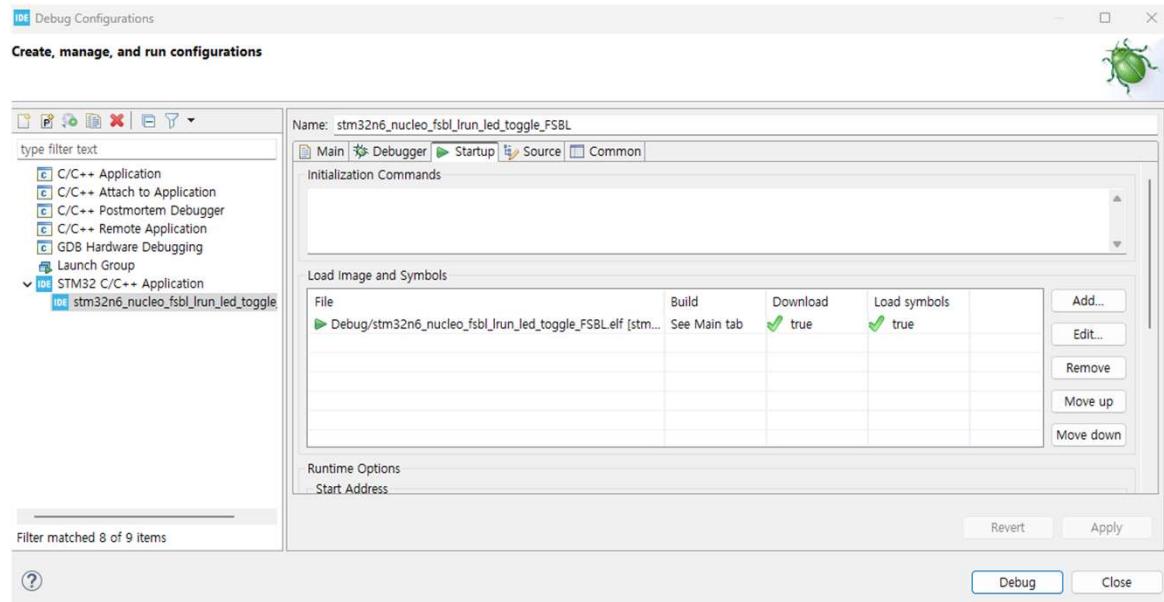
The screenshot shows the STM32CubeIDE interface. On the left is the Project Explorer with a tree view of the project structure. The main window displays the file `main.c`. A context menu is open over the project node, specifically under the "Run As" section. The menu items include:

- New
- Go Into
- Open in New Window
- Show In Alt+Shift+W >
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Delete
- Source > v) : ORIGIN = 0x34000400, LENGTH
- Move... F2
- Rename... F2
- Import...
- Export...
- Build Project
- Clean Project
- Refresh F5
- Close Project
- Close Unrelated Projects
- Build Configurations
- Build Targets
- Index
- Run As
- Debug As > IDE 1 STM32 C/C++ Application
- Team
- Compare With
- Restore from Local History...
- Generate Code
- Convert to C++
- Run C/C++ Code Analysis
- Configure
- Properties Alt+Enter

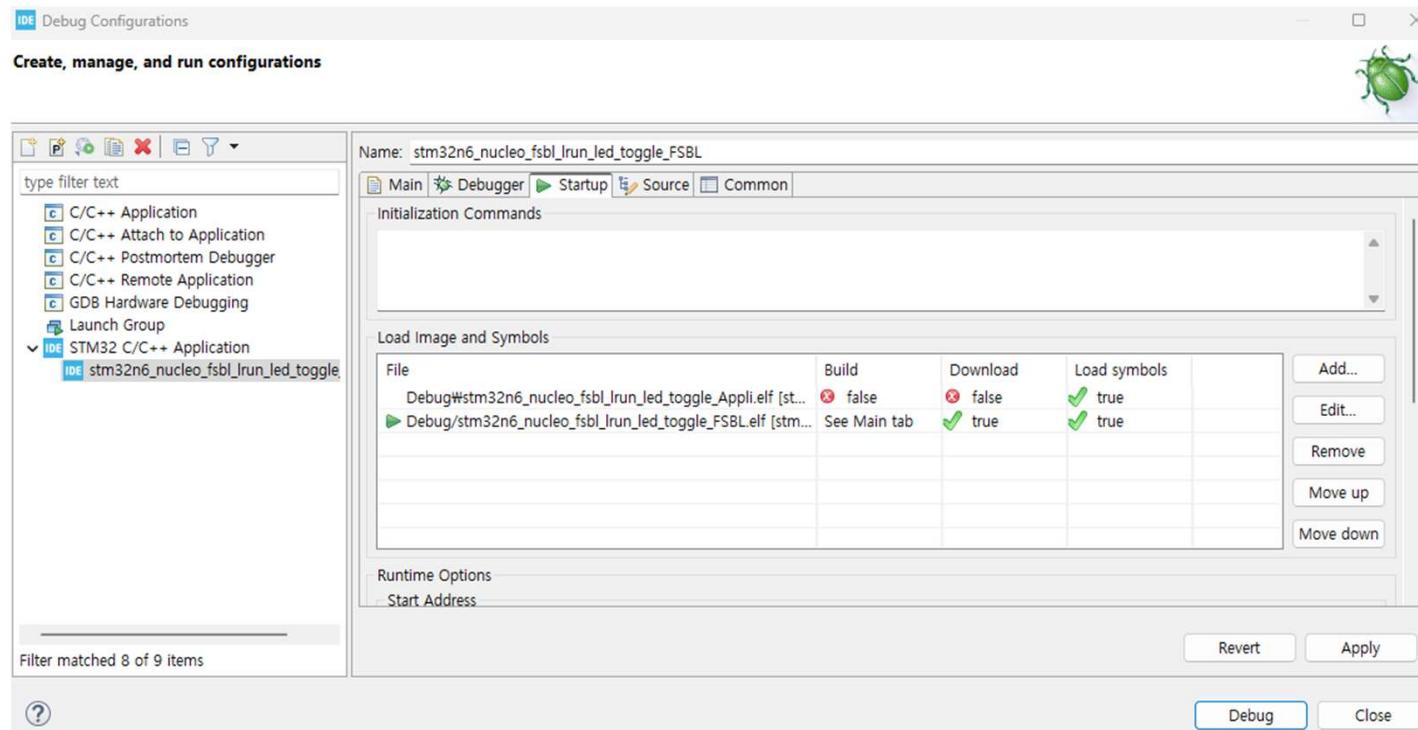
The "Debug As" submenu is expanded, showing "IDE 1 STM32 C/C++ Application" selected. The main code editor shows the beginning of the `main.c` file, including the startup code and memory definitions.



LRUN Debug



LRUN Debug



LRUN Debug

The screenshot shows a debugger interface with the title "LRUN Debug". The left pane displays the "Project Explorer" with a project named "stm32n6_nucleo_fsb1_lrunt_led_toggle_FSBL [STM32 C/C++ Application]" containing files like main.c and stm32n6xx_hal.c. The "main.c" file is open, showing C code for a GPIO toggle application. The right pane shows the "Disassembly" view, mapping the C code to assembly instructions. A specific assembly instruction at address 0x34000816, "nop", is highlighted in green, indicating it is the current instruction being executed.

```
File Edit Source Refactor Navigate Search Project Run Window Help
Debug X Project Explorer main.c 0x18003a1a main.c stm32n6xx_hal.c
IDE stm32n6_nucleo_fsb1_lrunt_led_toggle_FSBL [STM32 C/C++ Application]
  stm32n6_nucleo_fsb1_lrunt_led_toggle_FSBL.elf [cores: 1]
    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
67  /* USER CODE BEGIN 1 */
68
69  /* USER CODE END 1 */
70
71  /* MCU Configuration-----*/
72  HAL_Init();
73
74  /* USER CODE BEGIN Init */
75
76  /* USER CODE END Init */
77
78  /* USER CODE BEGIN SysInit */
79
80  /* USER CODE END SysInit */
81
82  /* Initialize all configured peripherals */
83  MX_GPIO_Init();
84  SystemIsolation_Config();
85  /* USER CODE BEGIN 2 */
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 }

Disassembly X 0x18003a1a
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(GPIOB,GPIO_PIN_10,
```



Hands-On: FSBL + XIP



CubeMX Configuration

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

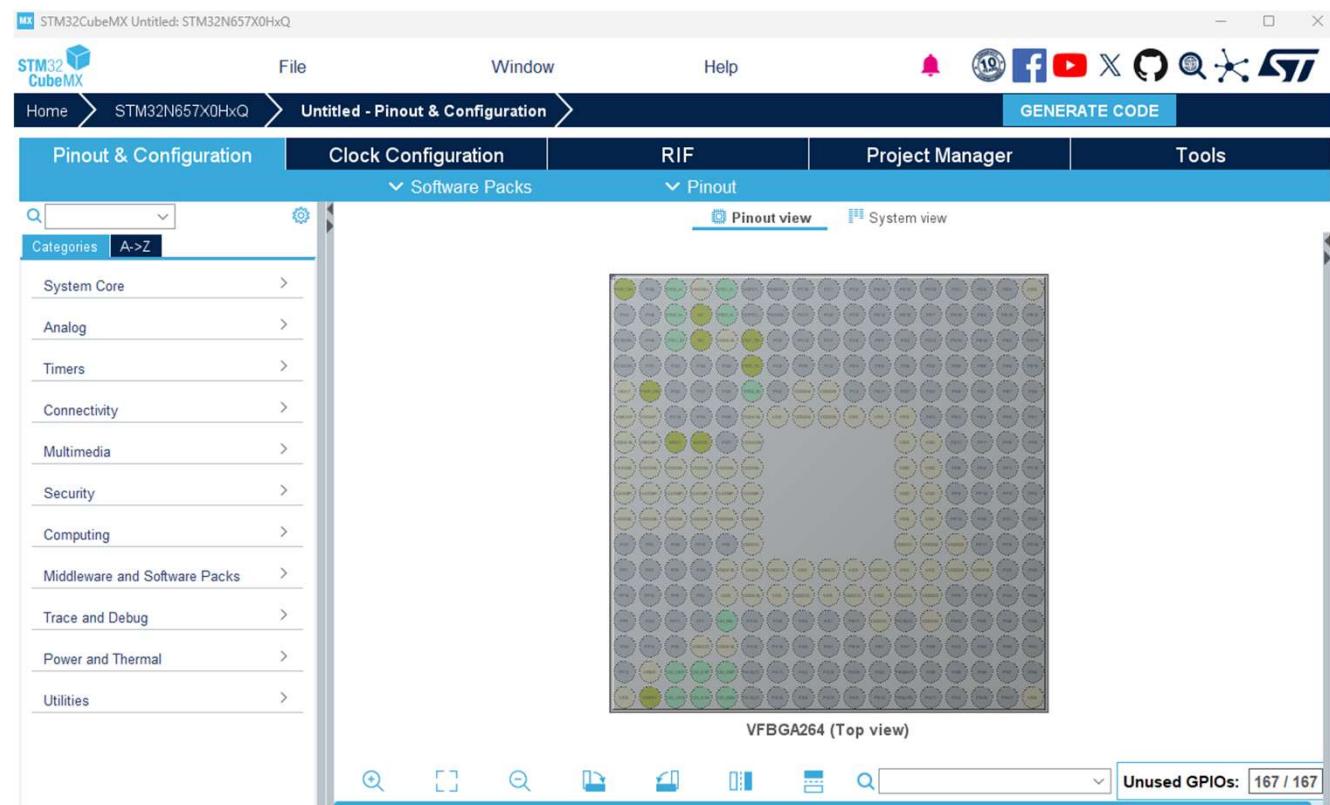
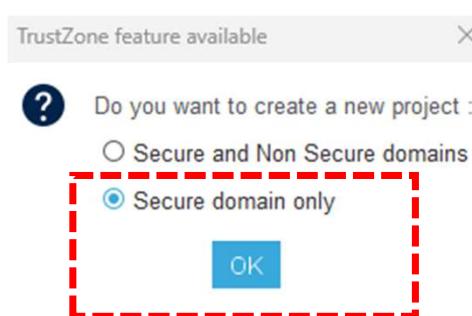
Commercial P...	Part No	Reference	Marketing St...	Unit Price for
STM32N657X0H...	STM32N657X0	STM32N657X0...	Active	10.7254



https://github.com/BlaineMoon30/STM32_Expert_Training_AI_STM32N6/tree/master/Hands_On/_3_Hands_On_FSBL_XIP

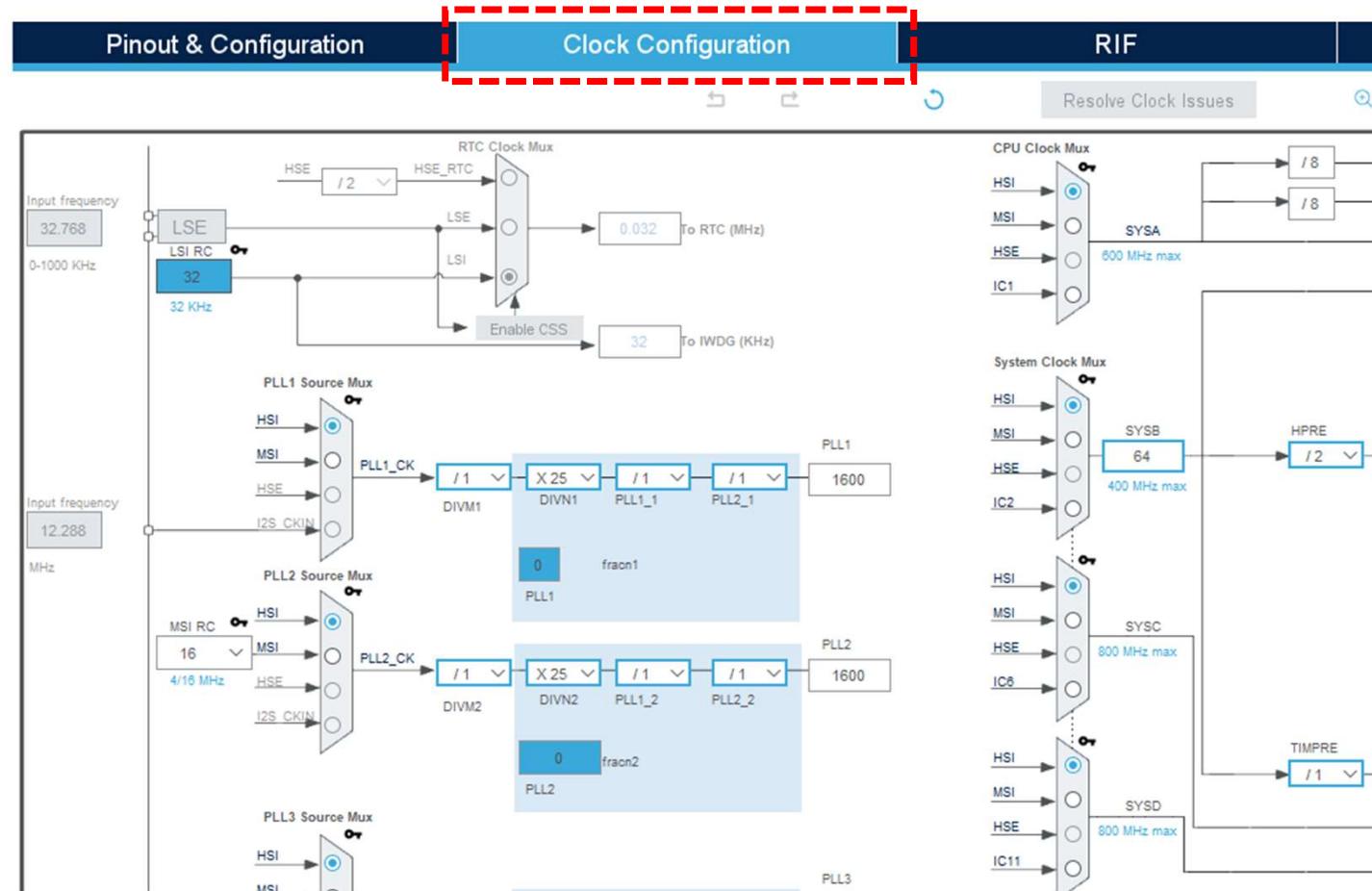
CubeMX Configuration

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



Clock Configuration

Click the **Clock Configuration** tab.



Electrical characteristics

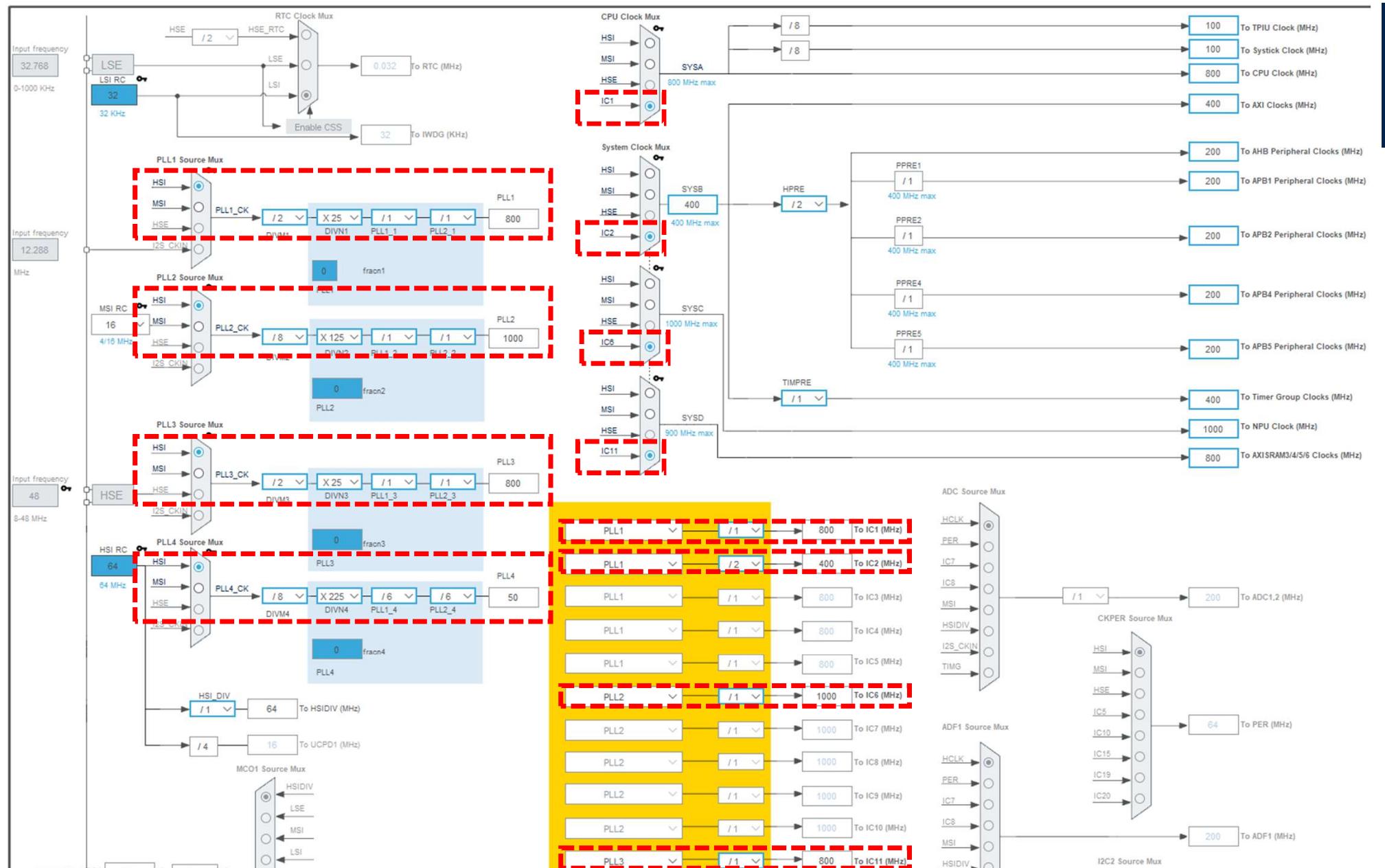
STM32N647xx STM32N657xx

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
$F_{CPU \text{ overdrive}}$	Clock frequency of Cortex-CM55 in overdrive (VOS high)		0	-	800	
F_{NPU}	Clock frequency of NPU/CNN		0	-	800	
$F_{NPU \text{ overdrive}}$	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} ($x = 1, 2, 3, 4, 5$)	Clock frequency of APB buses	-	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 >

System Core	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"/>		
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"/>	<input type="checkbox"/>

RCC Configuration

Pinout & Configuration Clock Configuration RIF

Categories A-Z

System Core

	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"/>		
RCC	<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"/>

Analog

Timers

Connectivity

Multimedia

Software Packs Pinout

RCC 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"/>

High Speed Clock (HSE) Disable

Low Speed Clock (LSE) Disable

Master Clock Output 1

Master Clock Output 2

Audio Clock Input (I2S_CKIN)

Configuration

Reset Configuration

Parameter Settings User Constants Features

Configure the below parameters :

Search (Ctrl+F)

Power Parameters

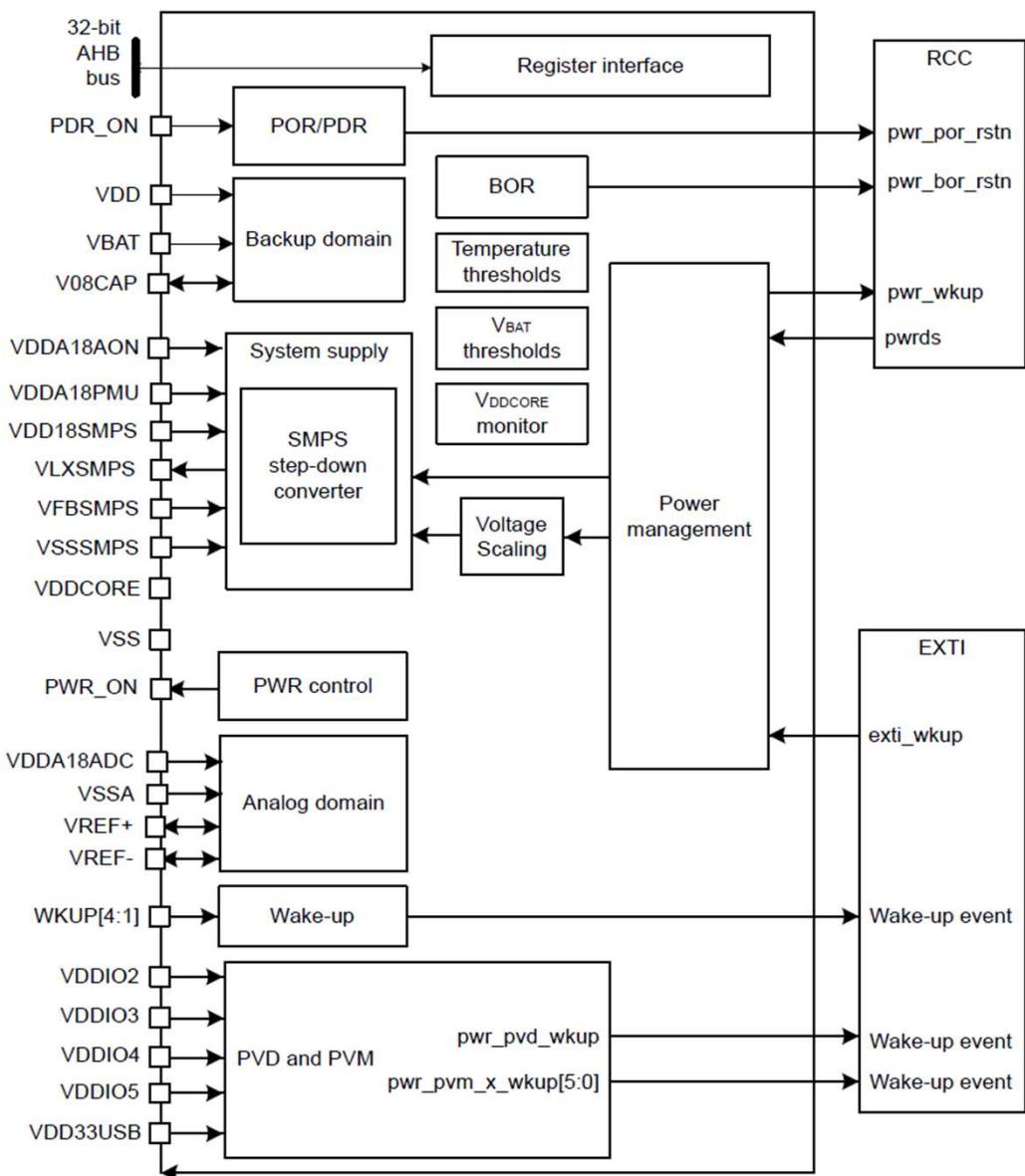
SupplySource	PWR_EXTERNAL_SOURCE_SUPPLY
Power Regulator Voltage Scale	Power Regulator Voltage Scale 0

Spread spectrum mode

PLL1 CSG mode	Disabled
PLL2 CSG mode	Disabled
PLL3 CSG mode	Disabled



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).

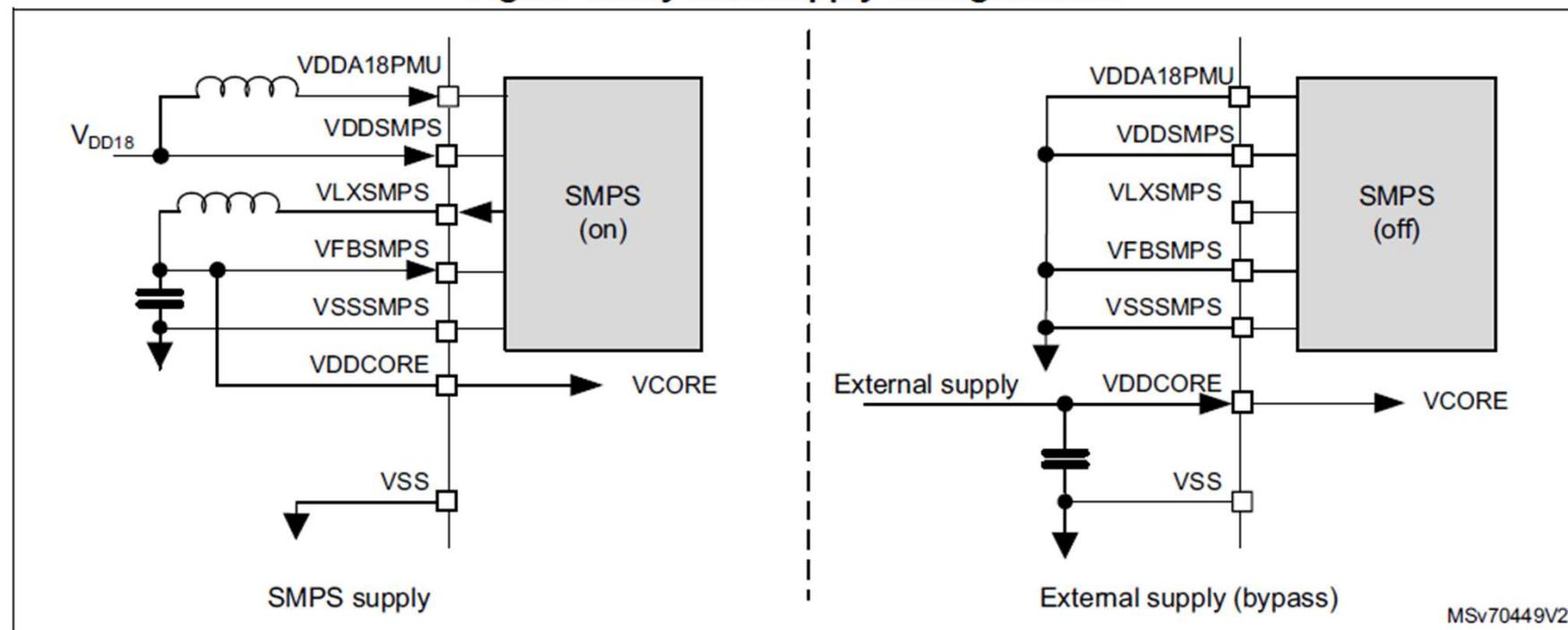
Power supply

RM0486

Power control (PWR)

By configuring the SMPS step-down converter, the supply configurations shown in [Figure 16](#) are supported for the V_{CORE} domain.

Figure 16. System supply configurations



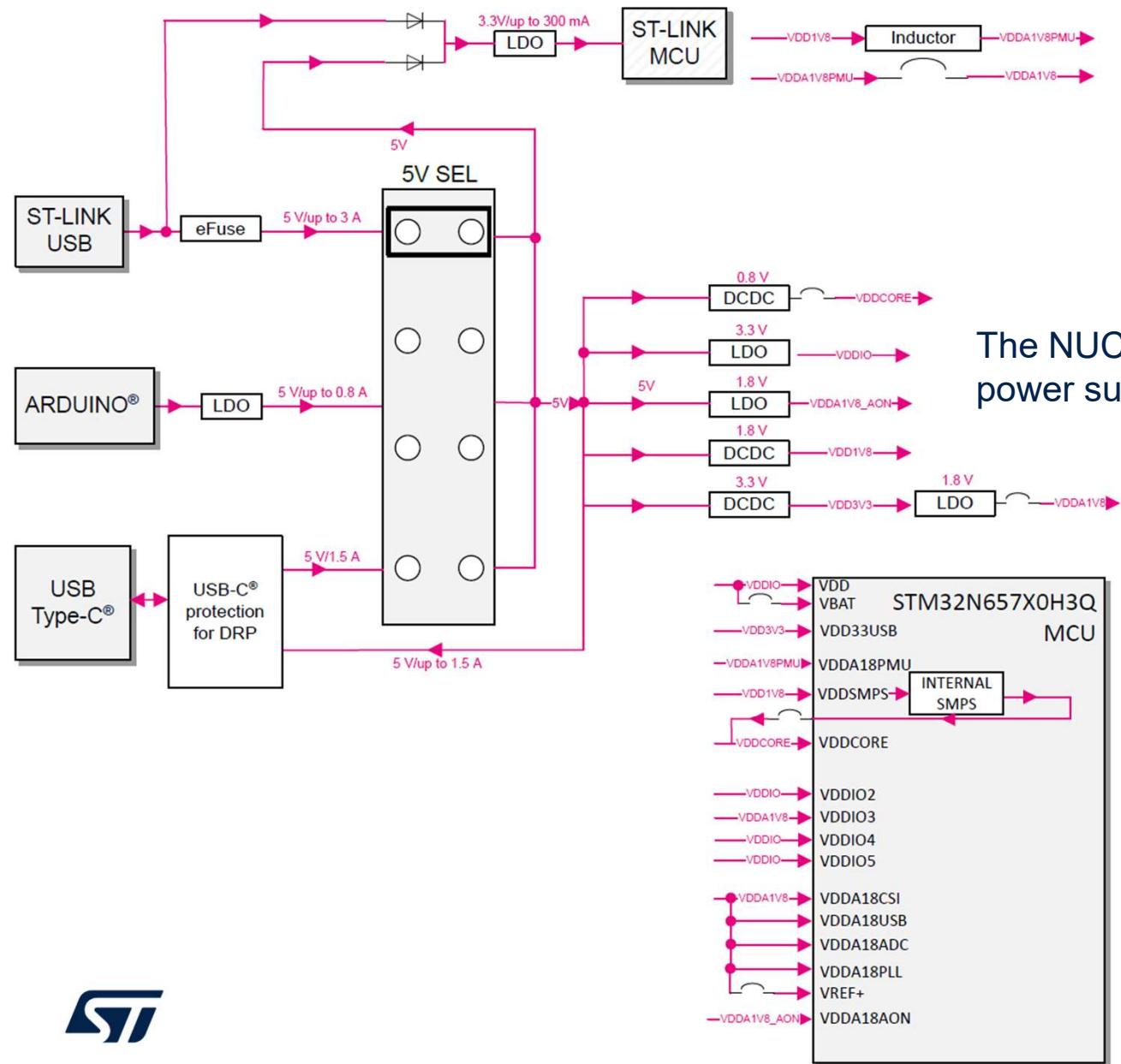
MSv70449V2

Power supply

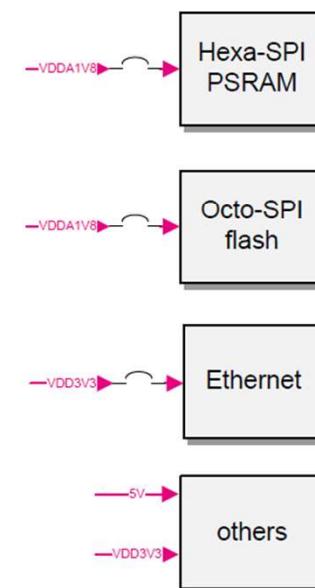
$V_{DD}^{(1)}$	I/Os supply voltage	1.8 V range	1.62	1.8	1.98	V
		3.3 V range	3.0	3.3	3.6	
$V_{DDA18ON}^{(1)}$	Internal analog supply voltage	-	1.71	1.8	1.935	
V_{DDIOx}	Specific I/Os supply voltage (x = 2, 3, 4, 5)	1.8 V range	1.71	1.8	1.935	
		3.3 V range	2.7	3.3	3.6	
V_{DDCORE}	Main digital logic supply 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.

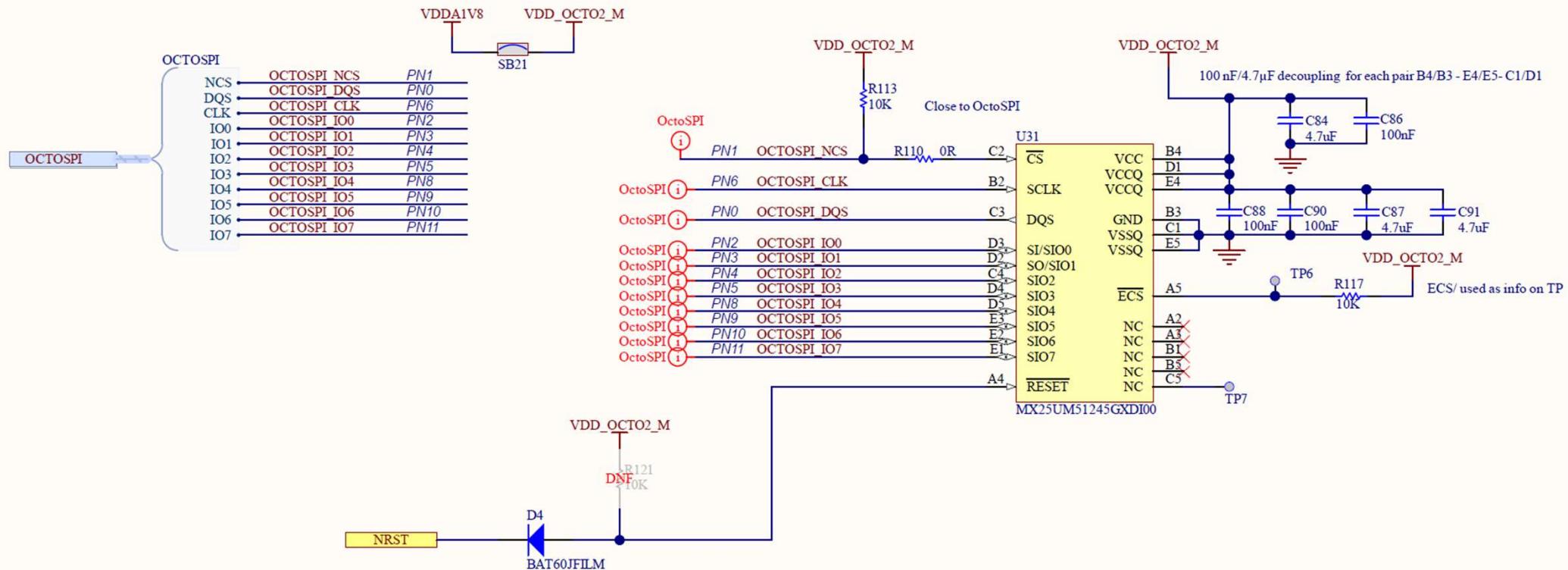


Octo-SPI Flash Memory Configuration

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	L14	T19	T17	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	-	-

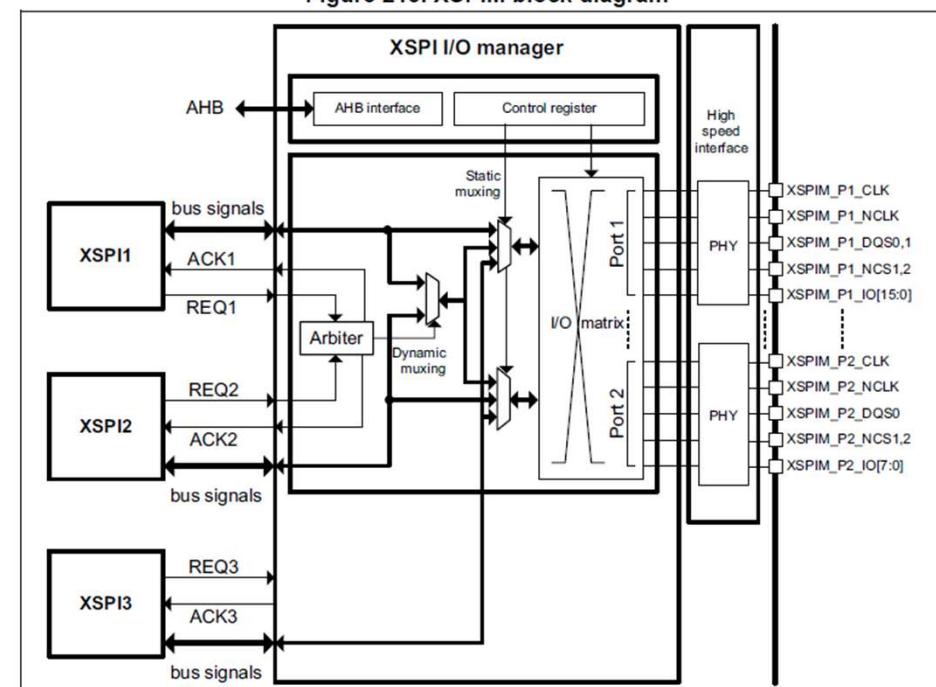
Octo-SPI Flash Memory Configuration

The screenshot shows two overlapping software windows. The top window is titled "Pinout & Configuration" and contains a search bar, a categories dropdown, and a table of pins for various peripherals like UCPD1, USART1-10, USB1_OTG, USB2_OTG, XSPI1-2, and YPD1-2. The bottom window is titled "Clock Configuration" and shows "XSPIM Mode and Configuration". It includes sections for "Runtime contexts:" (First Stage Boot Loader checked, Application, External Memory Loader), "Mode" (Direct selected from a dropdown menu), and a list of mode options: Direct (XSPI1 to Port1; XSPI2 to Port2; XSPI3 not used), 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). A red dashed box highlights the "XSPIM" row in the pinout table and the "Direct" mode selection in the configuration window.

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

Software Packs Pinout

XSPI2 Mode and Configuration

Mode

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

Runtime contexts:

Mode Octo SPI
Port Port2 Octo
HyperBus(TM) 1.8V Inverted Clock Disable
Chip Select Override NCS1 -- Port2 --

NVIC Settings DMA Settings GPIO Settings

Parameter Settings User Constants

Configure the below parameters :

Search (Ctrl+F)

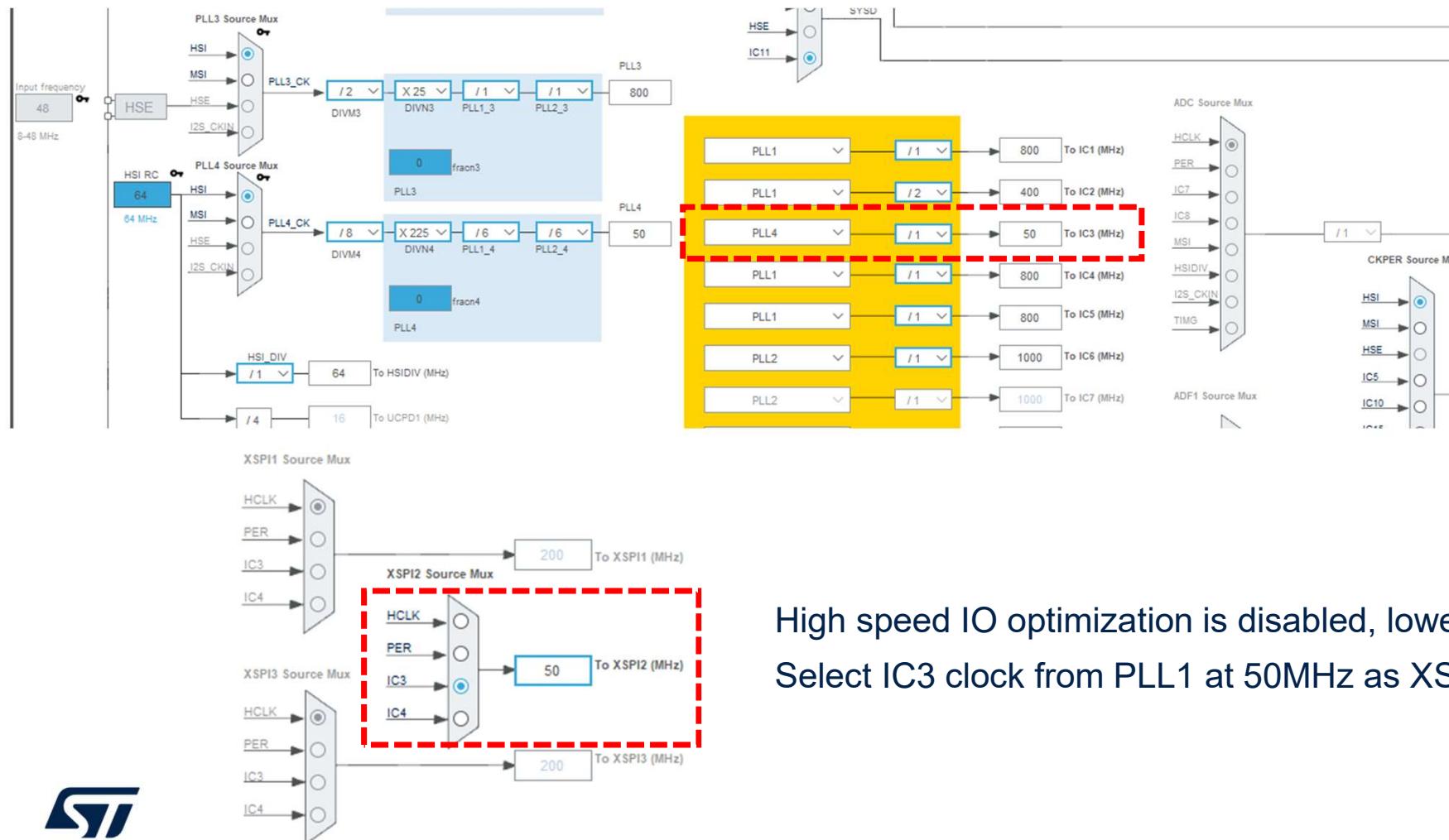
Generic

Fifo Threshold	4
Memory Mode	Disable
Memory Type	Macronix
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

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 interface with the 'OTP MPU' tab selected. The main area displays a table of OTP memory locations. The columns are: Name, Word, Value, Status, Lock, and Description. The 'Lock' column contains checkboxes, with most entries having checked boxes and one entry (OTP124) having an unchecked box. The 'Description' column provides details for each location, such as 'Permanent write lock' for OTP121, OTP122, and OTP123, and various reset and voltage-related descriptions for the other entries.

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

Buttons at the bottom include 'Export', 'Save OTP partition', 'Lock all' (unchecked), 'Apply', and 'Read'.

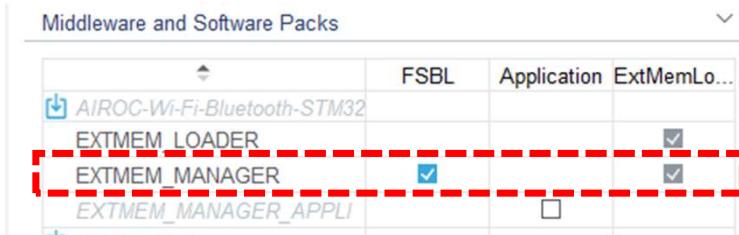
You can fuse **OTP** in **STM32CubeProgrammer**.

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

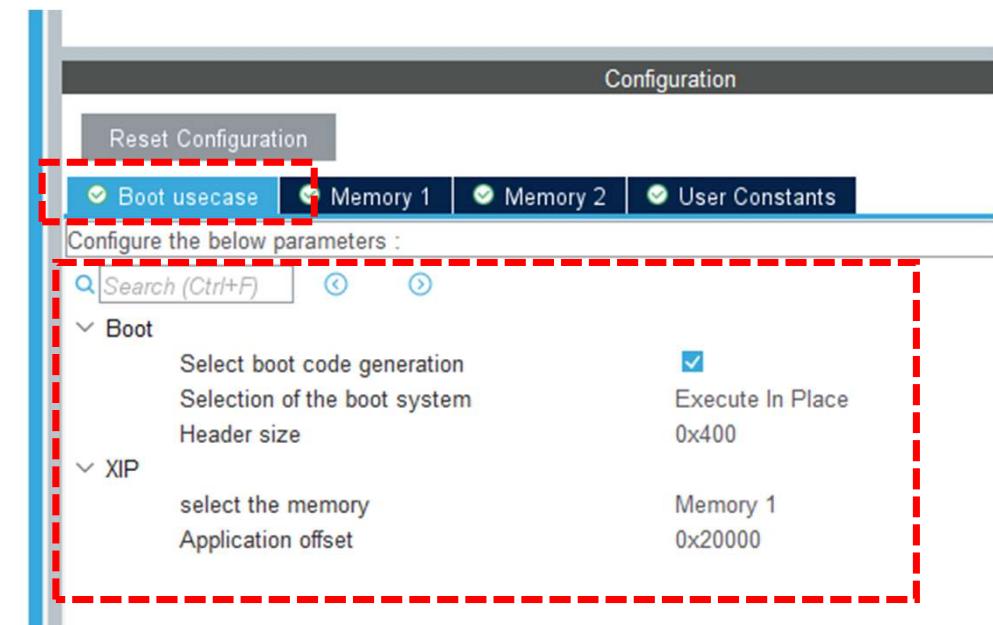
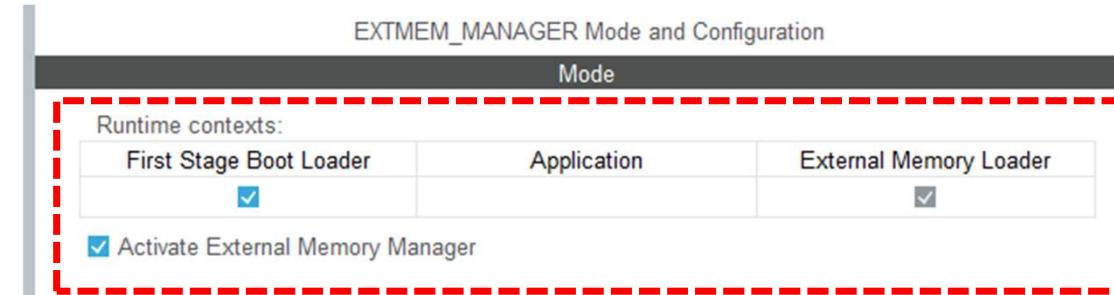
In this hands-on, **OTP settings will not be configured**.



EXTMEM_MANAGER Configuration



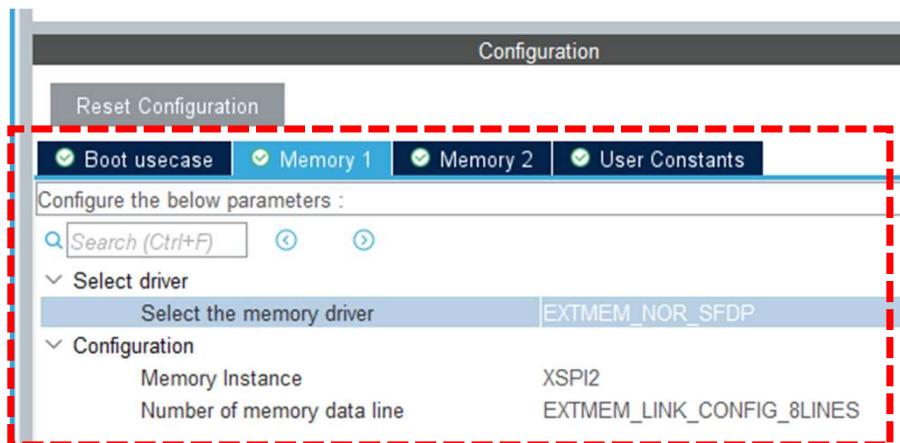
Select **FSBL** as the Context for **EXTMEM_MANAGER**.



Set the **Boot System** to **Excute In Place**.
Application offset : 0x20000



EXTMEM_MANAGER Configuration

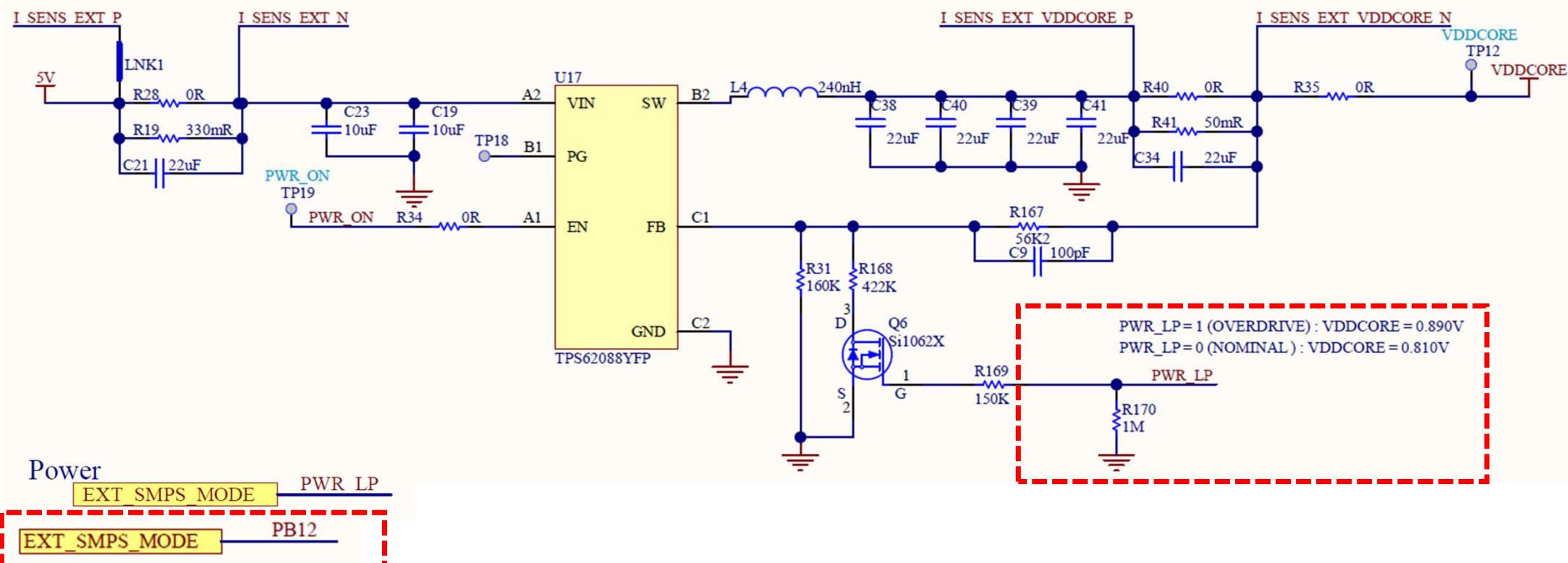


In this hands-on, FSBL loads the **Secure Application** into **AXISRAM1 (0x34000000)**.

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**.



Set PB12 PIN as GPIO_Output.

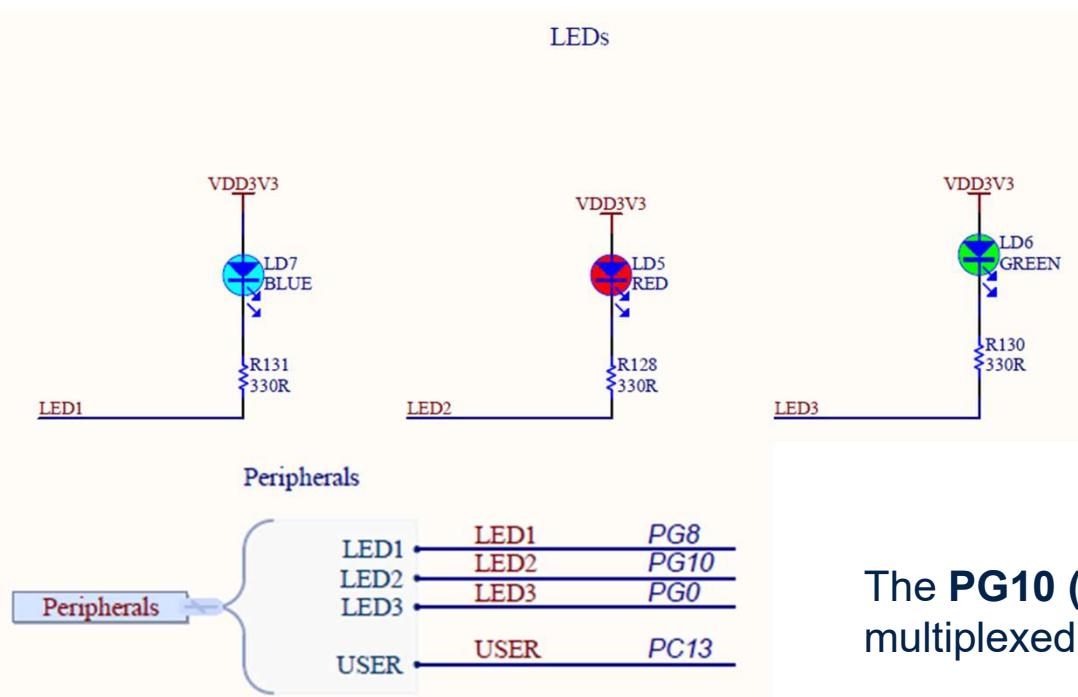
Pin N...	Signal on...	Pin Conte...	Pin Privil...	GPIO out...	GPIO mode	GPIO Pul...	Maximu...	User Label	Modified
PB12	n/a	First Sta...	n/a	High	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>
PG0	n/a	Application	n/a	Low	Output P...	No pull-u...	Low		<input checked="" type="checkbox"/>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

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

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"/>
GPIO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
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"/>		
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"/>	
WWIO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

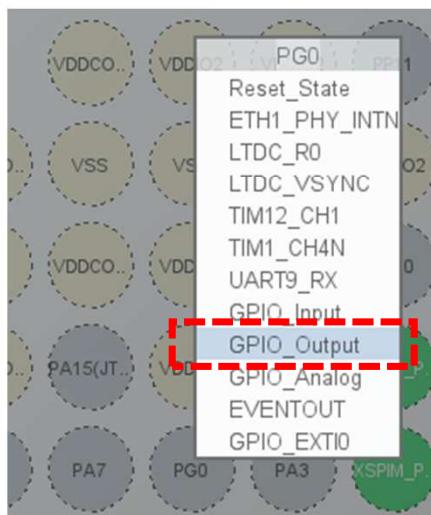
Configuration

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	Low
GPIO mode	Output Push Pull
GPIO Pull-up/Pull-down	No pull-up and no pull-down
Maximum output speed	Low
User Label	



BSEC Configuration

The screenshot shows the Pinout & Configuration interface with the following details:

Pinout & Configuration Tab:

- Search bar:
- Categories: System Core, Analog, Timers, Connectivity, Multimedia, Security.
- Table Headers: FSBL, Application, ExtMemLo.
- Table Rows:
 - BSEC** (highlighted with a red dashed box): FSBL checked, Application checked, ExtMemLo checked.
 - CRYP: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - HASH: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - MCE1: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - MCE2: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - MCE3: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - MCE4: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - PKA: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - RIF: FSBL unchecked, Application checked, ExtMemLo unchecked.
 - RNG: FSBL unchecked, Application unchecked, ExtMemLo unchecked.
 - SAES: FSBL unchecked, Application unchecked, ExtMemLo unchecked.

Clock Configuration Tab:

BSEC Mode and Configuration

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

Runtime contexts:

- First Stage Boot Loader (checked)
- Application
- External Memory Loader

Configuration

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

GENERATE CODE

STM32N657X0HxQ > stm32n6_nucleo_fsbl_xip_led_toggle.ioc - Project Manager > GENERATE CODE

Pinout & Configuration | Clock Configuration | RIF | Project Manager | Tools

Project

Project Settings

Project Name: stm32n6_nucleo_fsbl_xip_led_toggle

Project Location: C:\STM32_Expert_Training_AI_STM32N6\Hands_On_3_Hands_On_FSBL_XIP | Browse

Project Structure: FSBL Appli ExtMemLoader

Application Structure: Advanced | Do not generate the main()

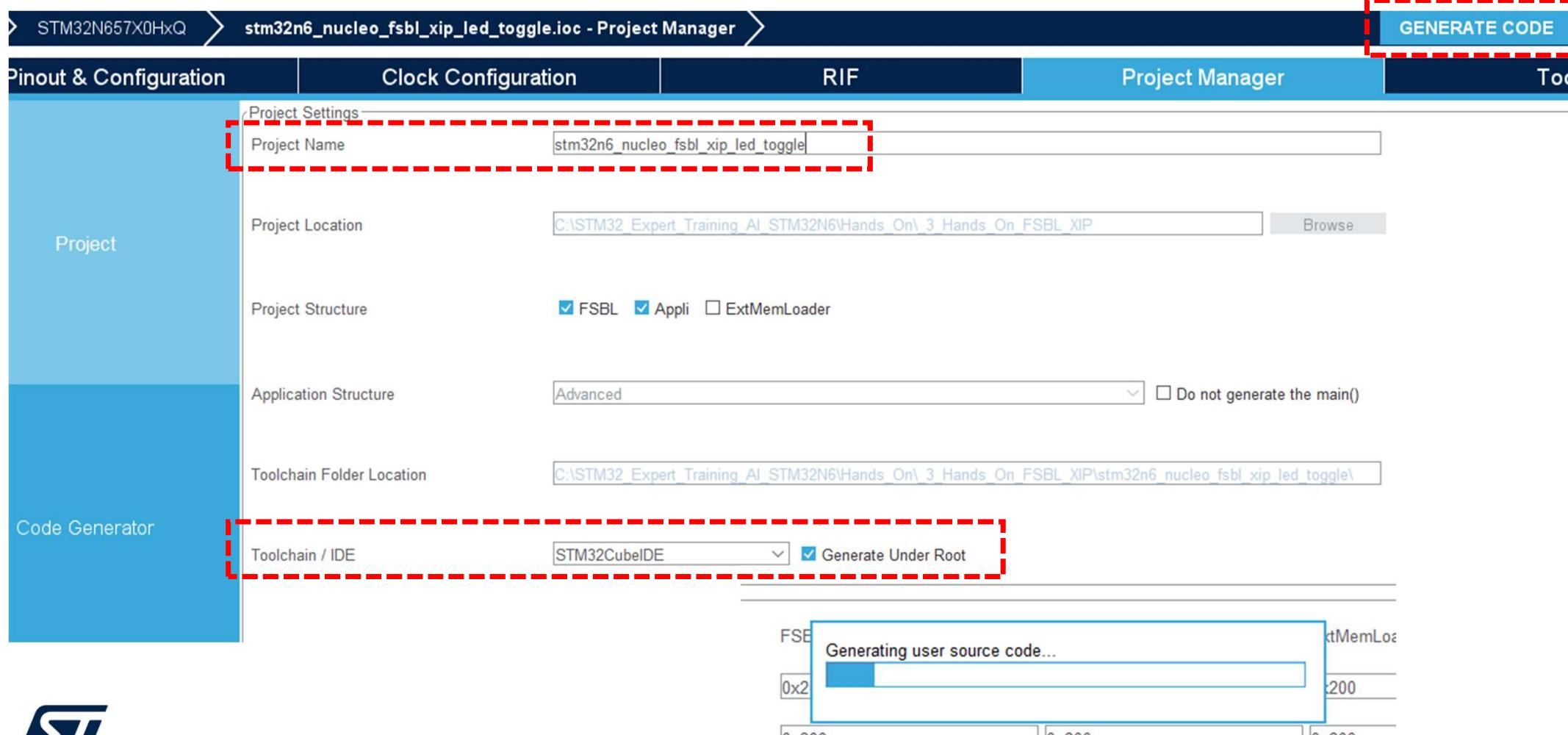
Toolchain Folder Location: C:\STM32_Expert_Training_AI_STM32N6\Hands_On_3_Hands_On_FSBL_XIP\stm32n6_nucleo_fsbl_xip_led_toggle\

Code Generator

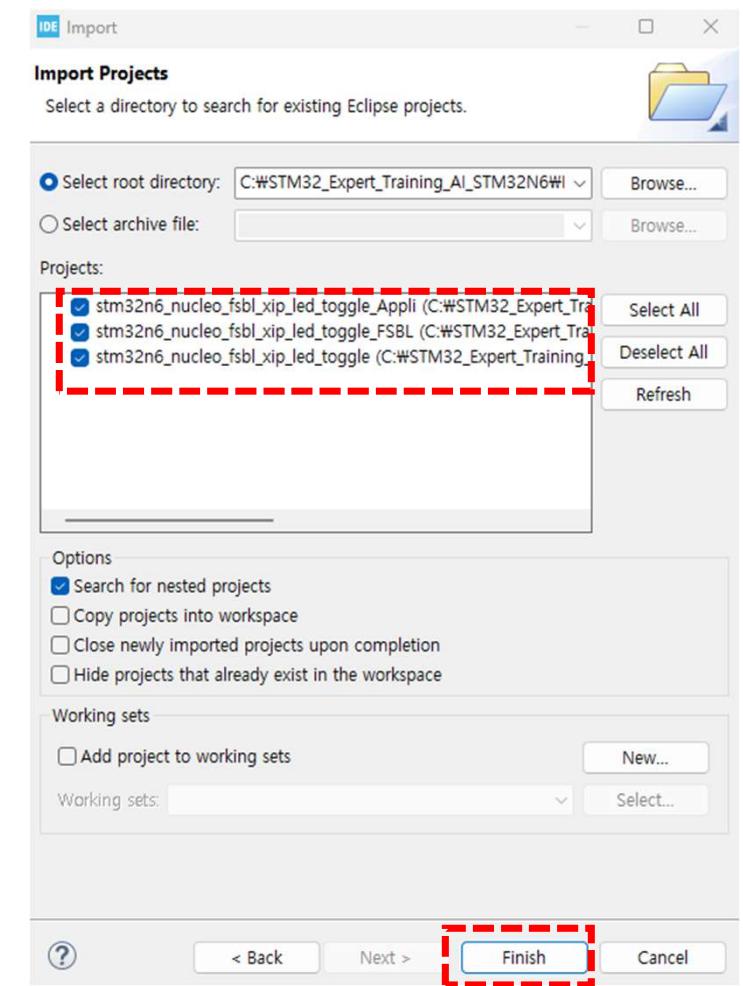
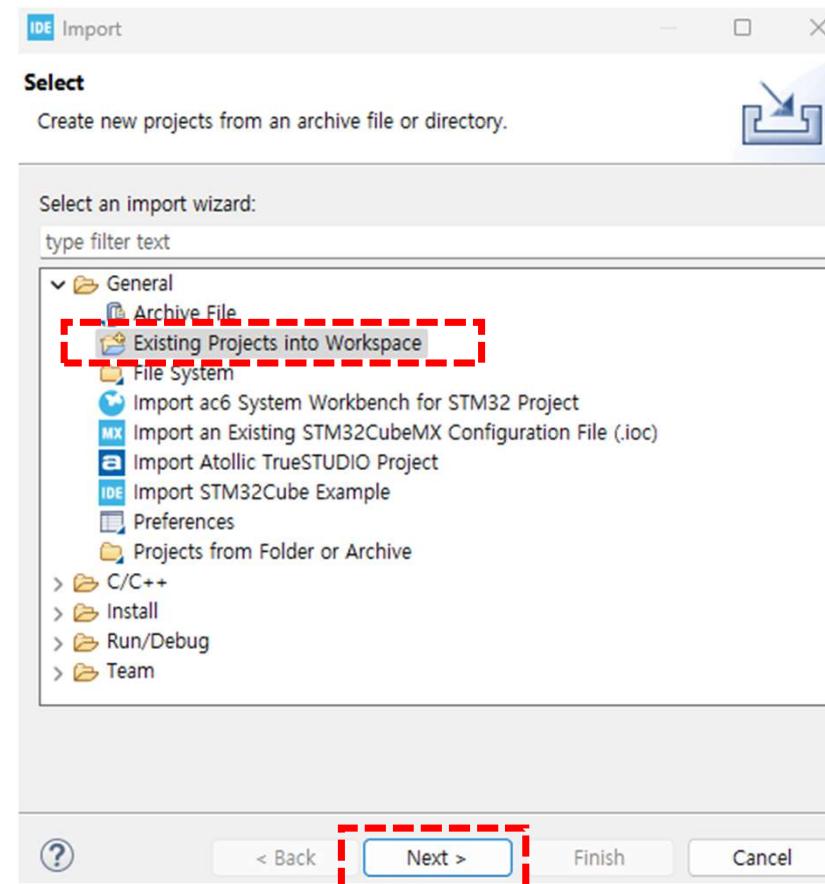
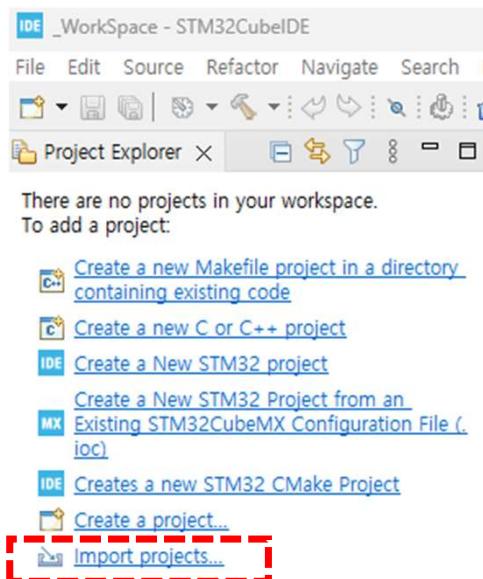
Toolchain / IDE: STM32CubeIDE | Generate Under Root

FSE Generating user source code... 0x200
0x800 0x800 0x800

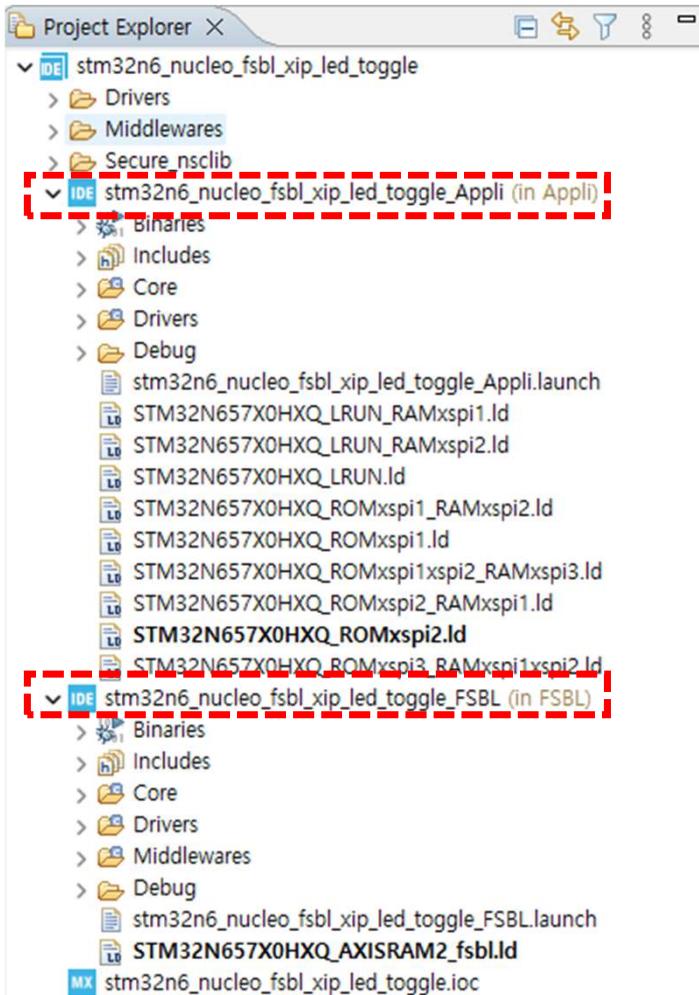
xtMemLoa 200



STM32CubeIDE Import Project



STM32CubeIDE Import Project



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

Add Code to FSBL

The screenshot shows the STM32CubeIDE interface with the Project Explorer open. The main.c file is selected in the STM32n6_nucleo_fsbl_xip_led_toggle_FSBLe project. The code in main.c includes cache enable, MCU configuration, and system clock setup. It also contains sections for user code initialization and system startup.

```
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 */
106
107 /* Launch the application */
108 if (BOOT_OK != BOOT_Application())
109 {
110     Error_Handler();
111 }
112 /* Infinite loop */
113 /* USER CODE BEGIN WHILE */
114 while (1)
115 {
116     /* USER CODE END WHILE */
117
118     /* USER CODE BEGIN 3 */
119 }
120 /* USER CODE END 3 */
121
122 /* USER CODE BEGIN CLK 1 */
123 /* USER CODE END CLK 1 */
124
```

The screenshot shows the main.c file in the STM32n6_nucleo_fsbl_xip_led_toggle_FSBLe project. A red box highlights the addition of MX_GPIO_Init() and HAL_Delay(1) in the user code section. The code now includes these two lines between the user code begin and end markers.

```
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 */
109
```

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”

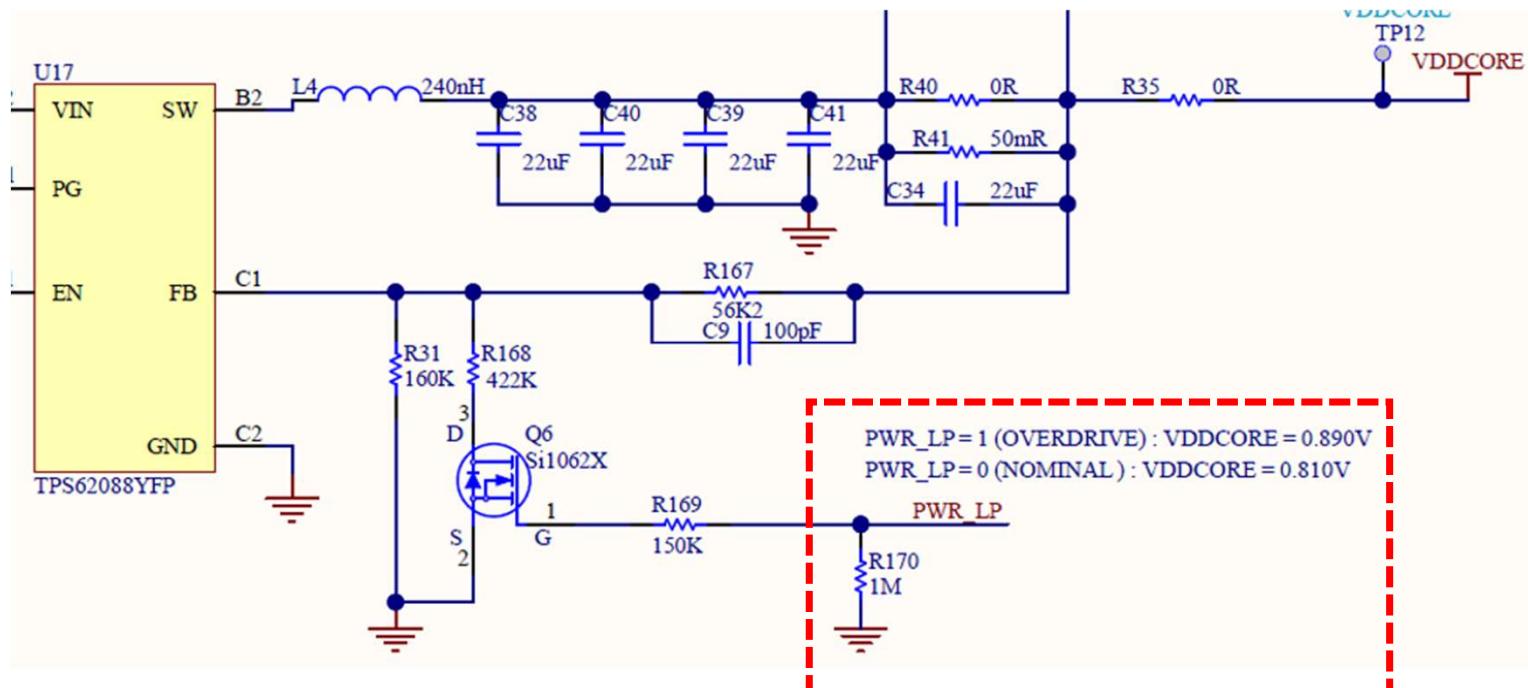


Add Code to FSBL

```

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_GPION_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 ST-IDE Project Explorer with the following structure:

- stm32n6_nucleo_fsbl_xip_led_toggle
 - Drivers
 - Middlewares
 - Secure_nsclib
- stm32n6_nucleo_fsbl_xip_led_toggle_Appli (in Appli)
 - Binaries
 - Includes
 - Core
 - Drivers
 - Debug
 - stm32n6_nucleo_fsbl_xip_led_toggle_Appli.launch
 - STM32N657X0HXQ_LRUN_RAMxspi1.ld
 - STM32N657X0HXQ_LRUN_RAMxspi2.ld
 - STM32N657X0HXQ_LRUN.Id
 - STM32N657X0HXQ_ROMxspi1_RAMxspi2.Id
 - STM32N657X0HXQ_ROMxspi1.Id
 - STM32N657X0HXQ_ROMxspi1xspi2_RAMxspi3.Id
 - STM32N657X0HXQ_ROMxspi2_RAMxspi1.Id
 - STM32N657X0HXQ_ROMxspi2.Id
 - STM32N657X0HXQ_ROMxspi3_RAMxspi1xspi2.Id
- stm32n6_nucleo_fsbl_xip_led_toggle_FSBLaunch (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
 - Drivers
 - Middlewares
 - Debug
 - stm32n6_nucleo_fsbl_xip_led_toggle_FSBLaunch.launch
 - STM32N657X0HXQ_AXISRAM2_fsbl.ld
 - stm32n6_nucleo_fsbl_xip_led_toggle.ioc

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 under 'stm32n6_nucleo_fsbl_xip_led_toggle_FSB'. The 'Core' folder contains 'Inc' and 'Src' subfolders. The 'Src' folder includes files like 'extmem_manager.c', 'main.c' (highlighted with a red dashed box), 'stm32n6xx_hal_msp.c' (also highlighted with a red dashed box), 'stm32h6xx_it.c', 'syscalls.c', 'sysmem.c', and 'system_stm32n6xx_fsbl.c'. Other project files shown include 'stm32n6_nucleo_fsbl_xip_led_toggle_FSB.launch', 'STM32N657X0HXQ_AXISRAM2_fsbl.id', and 'stm32n6_nucleo_fsbl_xip_led_toggle.ioc'. On the right, the code editor shows the 'stm32n6xx_hal_msp.c' file. A red dashed box highlights the user code block from line 94 to line 118. The code performs various initializations, including enabling BSEC and SYSCFG clocks, setting PWR and SYSCFG configurations for IO speed optimization, enabling the XSPI memory interface clock, and initializing peripheral clocks. The code ends with a red box at line 120.

```
/* USER CODE BEGIN XSPI2_MspInit_0 */  
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();  
  
/* USER CODE END XSPI2_MspInit_0 */  
  
/* Initializes the peripherals clock */  
PeriphClkInitStruct.PeriphClockSelection = RCC_PERIPHCLK_XSPI2;  
PeriphClkInitStruct.Xspi2ClockSelection = RCC_XSPI2CLKSOURCE_IC3;  
PeriphClkInitStruct.ICSelection[RCC_IC3].ClockSelection = RCC_ICCLKSOURCE_PLL4;  
PeriphClkInitStruct.ICSelection[RCC_IC3].ClockDivider = 1;  
if (HAL_RCCE_PeriphCLKConfig(&PeriphClkInitStruct) != HAL_OK)  
{  
    Error_Handler();  
}  
  
/* Peripheral clock enable */  
HAL_RCC_XCDTM_Clk_ENABLE();
```



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();

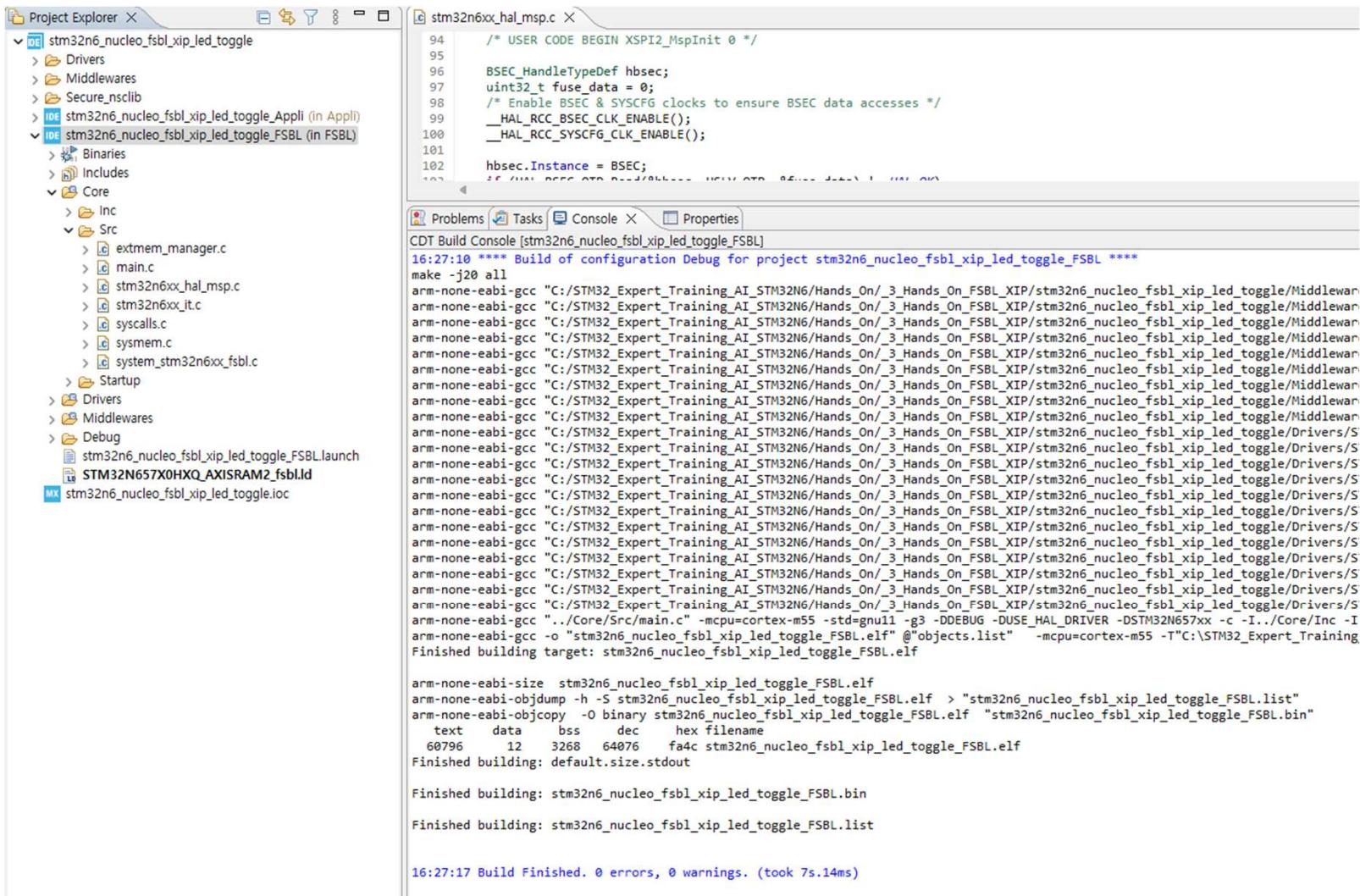
hbsec.Instance = BSEC;
if (HAL_BSEC OTP_Read(&hbsec, HSLV OTP, &fuse_data) != HAL_OK)
{
Error_Handler();
}

/* 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

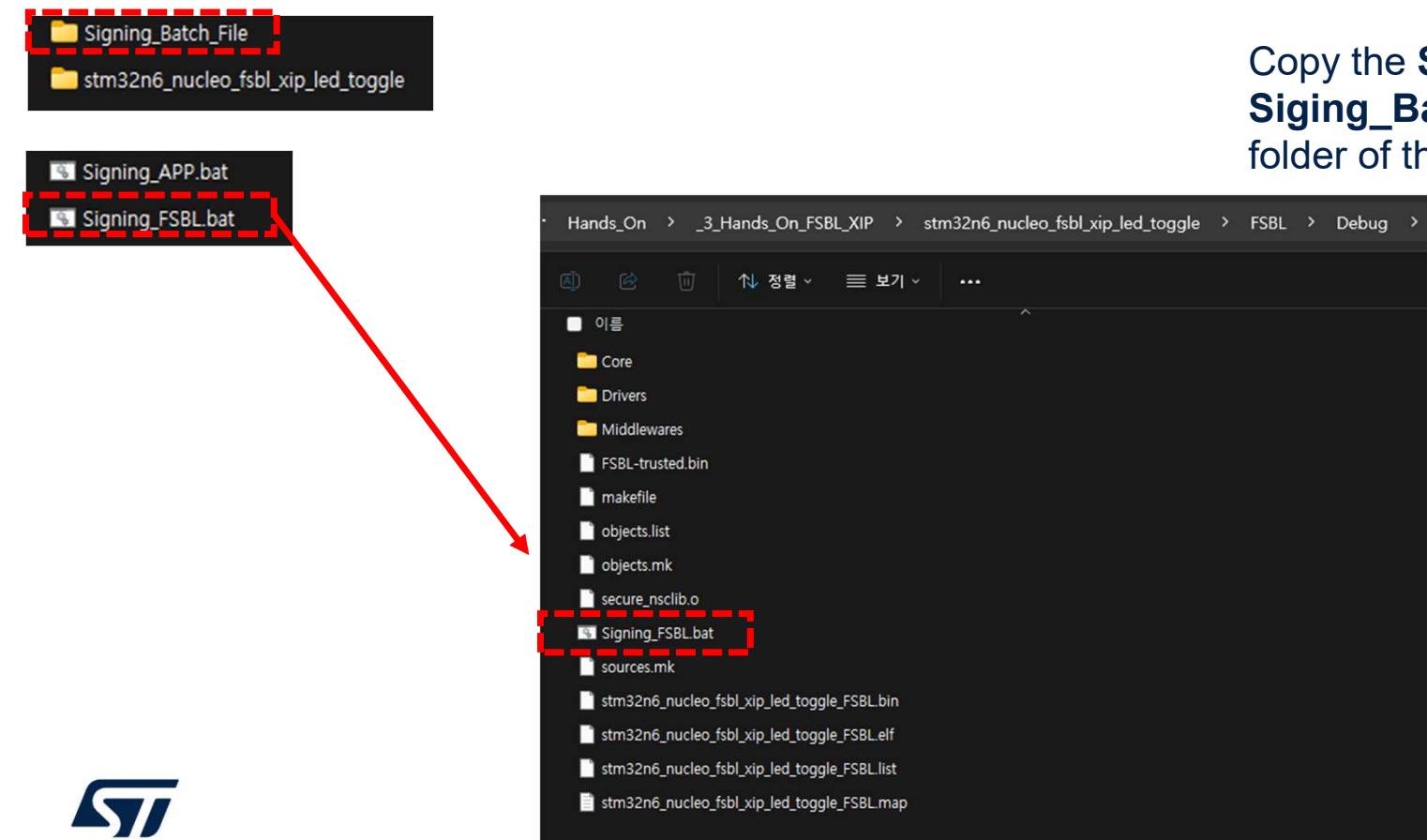


FSBL Header



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



Copy the **Signing_FSBL.bat** file from the **Siging_Batch_File** folder into the **FSBL\Debug** folder of the current project.



FSBL SIGNING

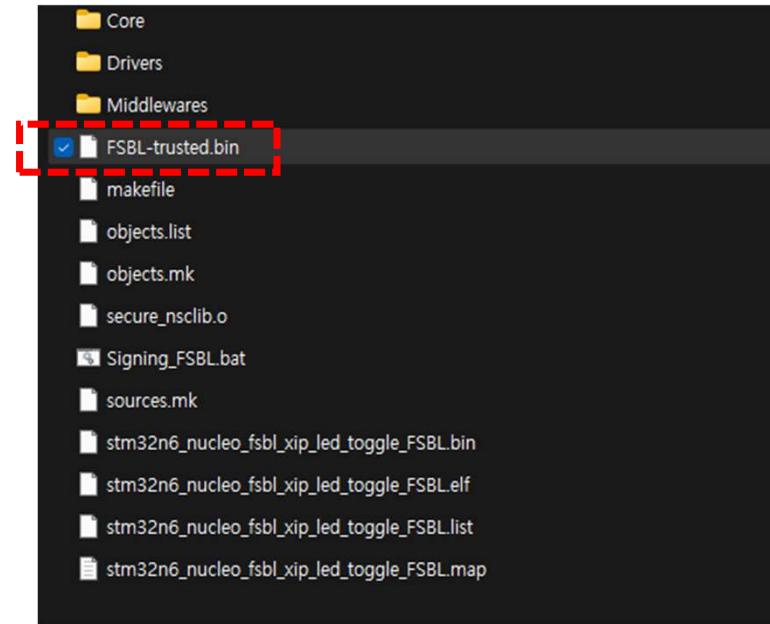
```
1 @echo off
2
3 REM STM32MP_SigningTool_CLI.exe -bin FSBL.bin -nk -of 0x80000000 -t fsbl -o FSBL-trusted.bin -hv 2.3 -dump FSBL-trusted.bin
4
5 set SIGNING_TOOL_PATH="C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe"
6 set APP_NAME="stm32n6_nucleo_fsbl_xip_led_toggle_FSBL.bin"
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_xip_led_toggle'. It includes sub-folders like Drivers, Middlewares, Secure_nsclib, and Core, which contains Inc and Src. The Src folder contains files such as main.c, secure_ns.c, stm32n6xx_hal_msp.c, stm32n6xx_it.c, syscalls.c, sysmem.c, and system_stm32n6xx_s.c. Below the project structure are launch files: STM32N657X0HXQ_LRUN_RAMxspi1.ld, STM32N657X0HXQ_LRUN_RAMxspi2.ld, STM32N657X0HXQ_LRUN.ld, STM32N657X0HXQ_ROMxspi1_ROMxspi2.ld, STM32N657X0HXQ_ROMxspi1.ld, STM32N657X0HXQ_ROMxspi2_ROMxspi3.ld, STM32N657X0HXQ_ROMxspi2_ROMxspi1.ld, STM32N657X0HXQ_ROMxspi2.ld, and STM32N657X0HXQ_ROMxspi3_ROMxspi1xspi2.ld. At the bottom, there are FSBL files: stm32n6_nucleo_fsbl_xip_led_toggle_FSBL (in FSBL) and stm32n6_nucleo_fsbl_xip_led_toggle.ioc. On the right, the main.c file is open in the code editor. A red dashed box highlights the following code block:

```
    HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_0);
    HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_8);
    HAL_GPIO_TogglePin(GPIOG,GPIO_PIN_10);
    HAL_Delay(1000);
```

```
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.



Build APP

```
> C sysmem.c
> C system_stm32n6xx_s.c
> Startup
> Drivers
> Debug
  stm32n6_nucleo_fsbl_xip_led_toggle_Appi.launch
  STM32N657X0HXQ_LRUN_RAMxspi1.ld
  STM32N657X0HXQ_LRUN_RAMxspi2.ld
  STM32N657X0HXQ_LRUN.ld
  STM32N657X0HXQ_ROMxspi1_RAMxspi2.ld
  STM32N657X0HXQ_ROMxspi1.ld
  STM32N657X0HXQ_ROMxspi2_RAMxspi3.ld
  STM32N657X0HXQ_ROMxspi2_RAMxspi1.ld
  STM32N657X0HXQ_ROMxspi3_RAMxspi2.ld
  STM32N657X0HXQ_ROMxspi3_RAMxspi2.ld
> IDE stm32n6_nucleo_fsbl_xip_led_toggle_FSB (in FSBL)
MX stm32n6_nucleo_fsbl_xip_led_toggle.ioc
```

```
31 ****
32 */
33
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 = 0x70020400, LENGTH = 511K
47   RAM  (xrw)    : ORIGIN = 0x34000000, LENGTH = 2048K
48 }
49
50 /* Sections */
51 SECTIONS
52 {
53   /* The startup code into "RAM" Ram type memory */
```

In this exercise, the address offset of the application is **0x20000**. The ROM in the linker script file is set to **0x70020400**.

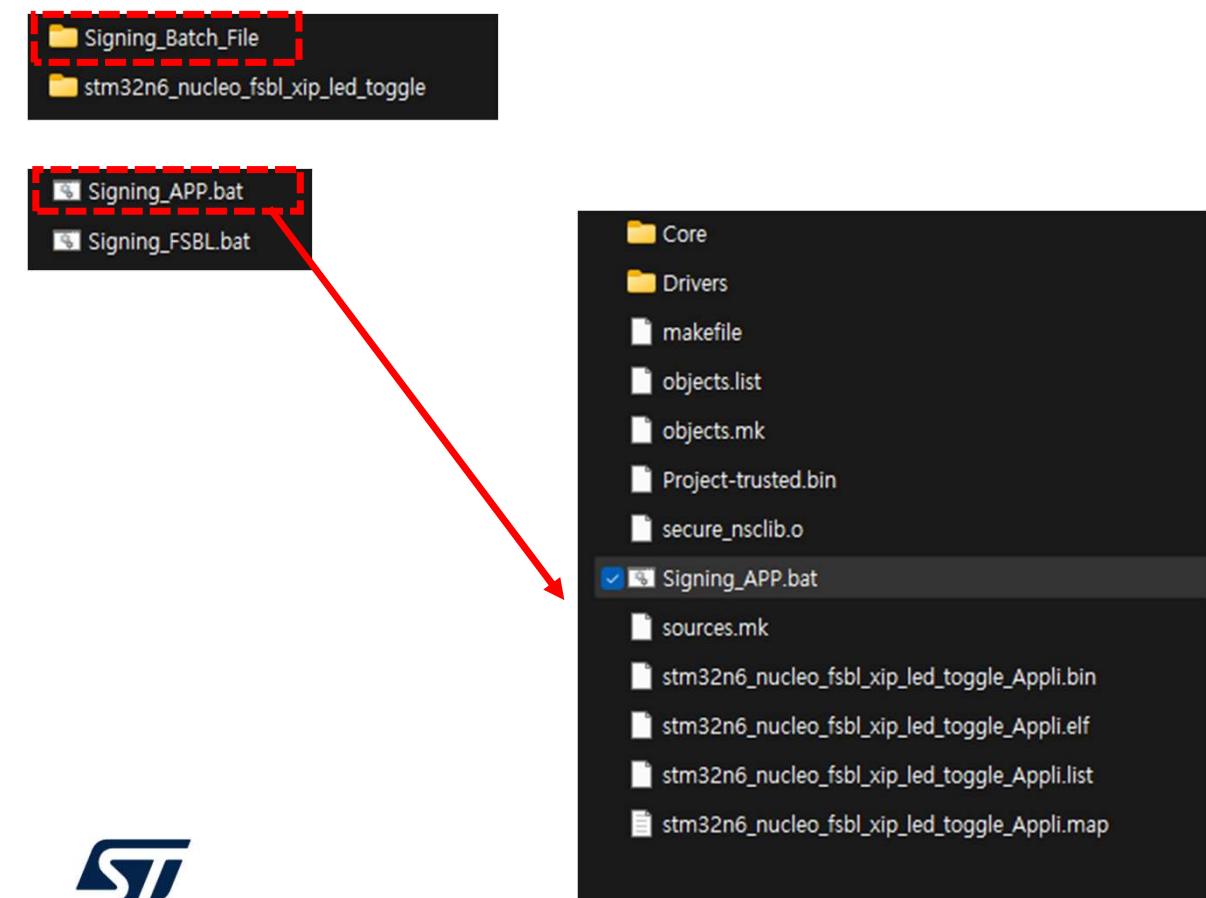
```
arm-none-eabi-size stm32n6_nucleo_fsbl_xip_led_toggle_Appi.elf
arm-none-eabi-objdump -h -S stm32n6_nucleo_fsbl_xip_led_toggle_Appi.elf > "stm32n6_nucleo_fsbl_xip_led_toggle_Appi.list"
arm-none-eabi-objcopy -O binary stm32n6_nucleo_fsbl_xip_led_toggle_Appi.elf "stm32n6_nucleo_fsbl_xip_led_toggle_Appi.bin"
      text   data   bss   dec   hex filename
        4656     12   2604   7272   1c68 stm32n6_nucleo_fsbl_xip_led_toggle_Appi.elf
Finished building: default.size.stdout

Finished building: stm32n6_nucleo_fsbl_xip_led_toggle_Appi.bin
Finished building: stm32n6_nucleo_fsbl_xip_led_toggle_Appi.list
```

16:41:41 Build Finished. 0 errors, 0 warnings. (took 3s.686ms)



APP SIGNING



Copy the **Signing_FSBL.bat** file from the **Siging_Batch_File** folder into the **FSBL\Debug** folder of the current project.

APP SIGNING

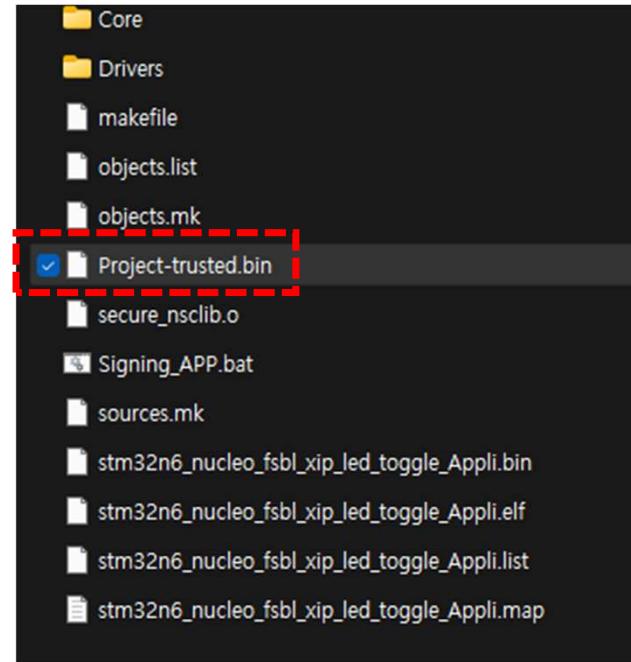
```
1 @echo off
2
3 REM STM32MP_SigningTool_CLI.exe -bin Project.bin -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -dump Project-trusted.bin
4
5 set SIGNING_TOOL_PATH="C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\STM32_SigningTool_CLI.exe"
6 set APP_NAME="stm32n6_nucleo_fsbl_xip_led_toggle_Appli.bin"
7
8 echo %SIGNING_TOOL_PATH% -bin %APP_NAME% -nk -of 0x80000000 -t fsbl -o Project-trusted.bin -hv 2.3 -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**.



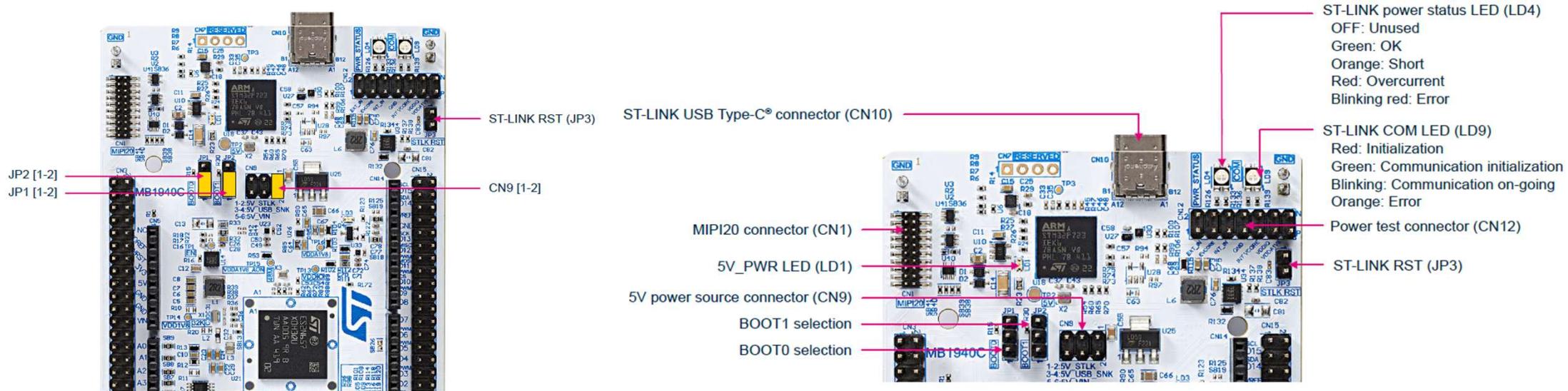
APP SIGNING



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

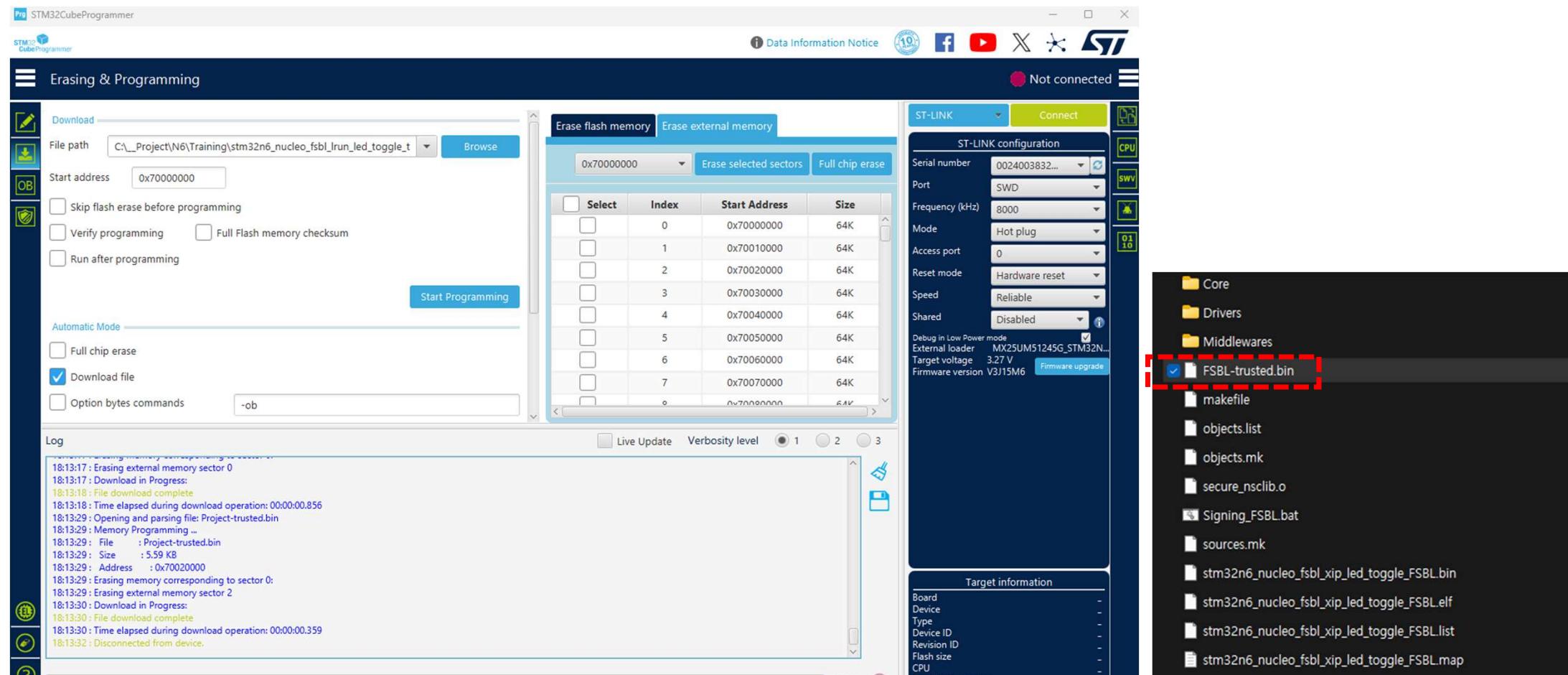


FSBL and APP Download



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 **FSBL-trusted.bin** file to **0x70000000**.

APP Download

The screenshot shows the STM32CubeProgrammer software interface for programming an STM32 device. The main window is titled "Erasing & Programming".

Left Panel (Download):

- File path: C:_Project\N6\Training\stm32n6_nucleo_fsbl_xip_led_toggle.t
- Start address: 0x70020000
- Checkboxes:
 - Skip flash erase before programming
 - Verify programming
 - Full Flash memory checksum
 - Run after programming
- Automatic Mode:
 - Full chip erase
 - Download file (checked)
 - Option bytes commands: -ob
- Log window showing the following text:

```
18:13:17 : Erasing external memory sector 0
18:13:17 : Download in Progress:
18:13:18 : File download complete
18:13:18 : Time elapsed during download operation: 00:00:00.856
18:13:29 : Opening and parsing file: Project-trusted.bin
18:13:29 : Memory Programming ...
18:13:29 : File : Project-trusted.bin
18:13:29 : Size : 5.59 KB
18:13:29 : Address : 0x70020000
18:13:29 : Erasing memory corresponding to sector 0:
18:13:29 : Erasing external memory sector 2
18:13:30 : Download in Progress:
18:13:30 : File download complete
18:13:30 : Time elapsed during download operation: 00:00:00.359
18:13:32 : Disconnected from device.
```

Center Panel (Erase flash memory):

- Address: 0x70000000
- Buttons: Erase selected sectors, Full chip erase
- Table of sectors:

Select	Index	Start Address	Size
	0	0x70000000	64K
	1	0x70010000	64K
	2	0x70020000	64K
	3	0x70030000	64K
	4	0x70040000	64K
	5	0x70050000	64K
	6	0x70060000	64K
	7	0x70070000	64K
	...	0x70080000	64K

Right Panel (ST-LINK):

- ST-LINK configuration:
 - Serial number: 0024003832...
 - Port: SWD
 - Frequency (kHz): 8000
 - Mode: Hot plug
 - Access port: 0
 - Reset mode: Hardware reset
 - Speed: Reliable
 - Shared: Disabled
- Debug in Low Power mode: checked
- External loader: MX25UM51245G_STM32N...
- Target voltage: 3.27 V
- Firmware version: V3J15M6
- Buttons: Firmware upgrade
- Target information:
 - Board
 - Device
 - Type
 - Device ID
 - Revision ID
 - Flash size
 - CPU
 - Bootloader Version

File Browser:

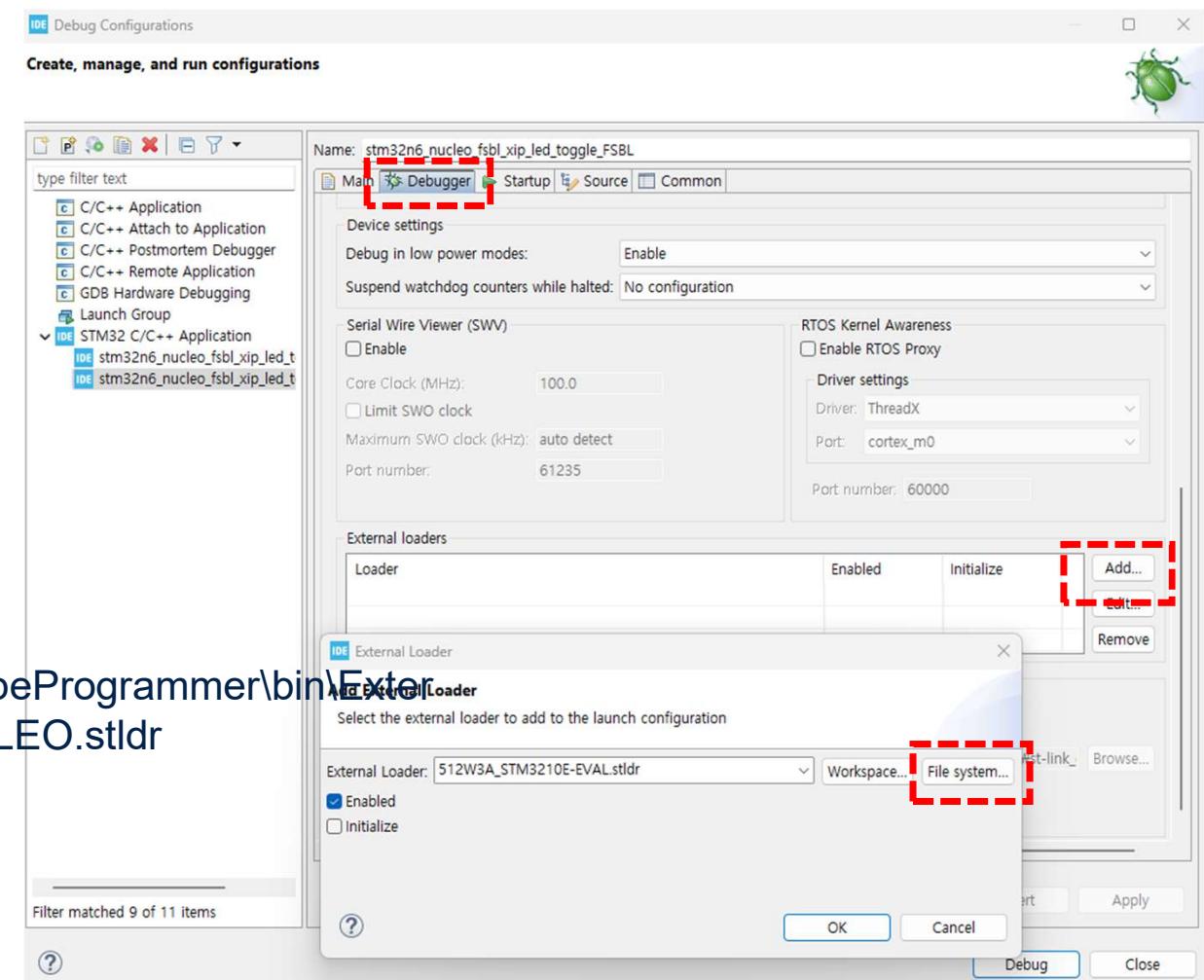
- Core
- Drivers
- makefile
- objects.list
- objects.mk
- Project-trusted.bin (highlighted with a red dashed box)
- secure_nsclib.o
- Signing_APP.bat
- sources.mk
- stm32n6_nucleo_fsbl_xip_led_toggle_Appi.bin
- stm32n6_nucleo_fsbl_xip_led_toggle_Appi.elf
- stm32n6_nucleo_fsbl_xip_led_toggle_Appi.list
- stm32n6_nucleo_fsbl_xip_led_toggle_Appi.map



Write the **Project-trusted.bin** file to **0x70020000**.

XIP Debug

The screenshot shows the STM32CubeMX software interface. On the left, there's a tree view of the project structure under 'stm32n6_nucleo_fsbl_xip_led_toggle'. The main area is a code editor displaying C code for the application. A context menu is open over the code, with the 'Debug Configurations...' option highlighted.

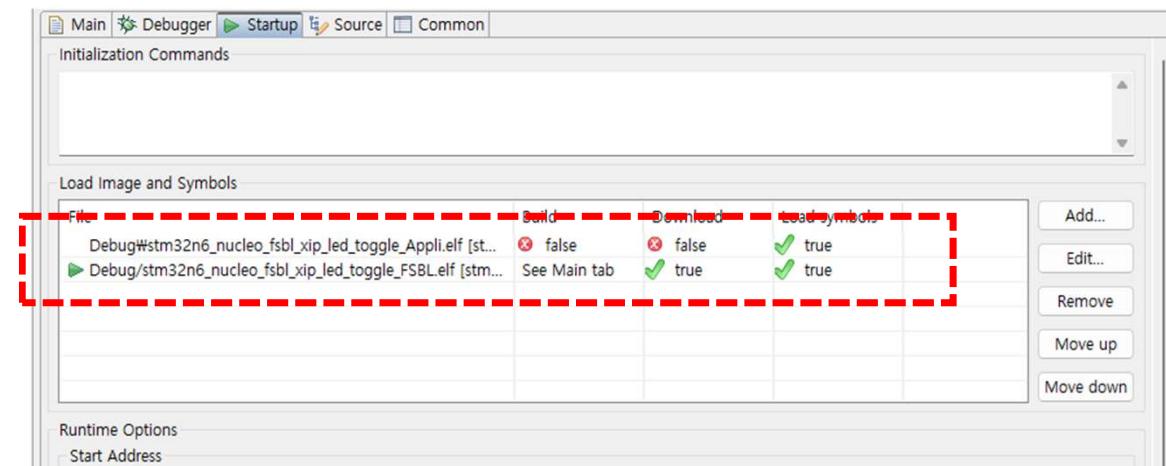
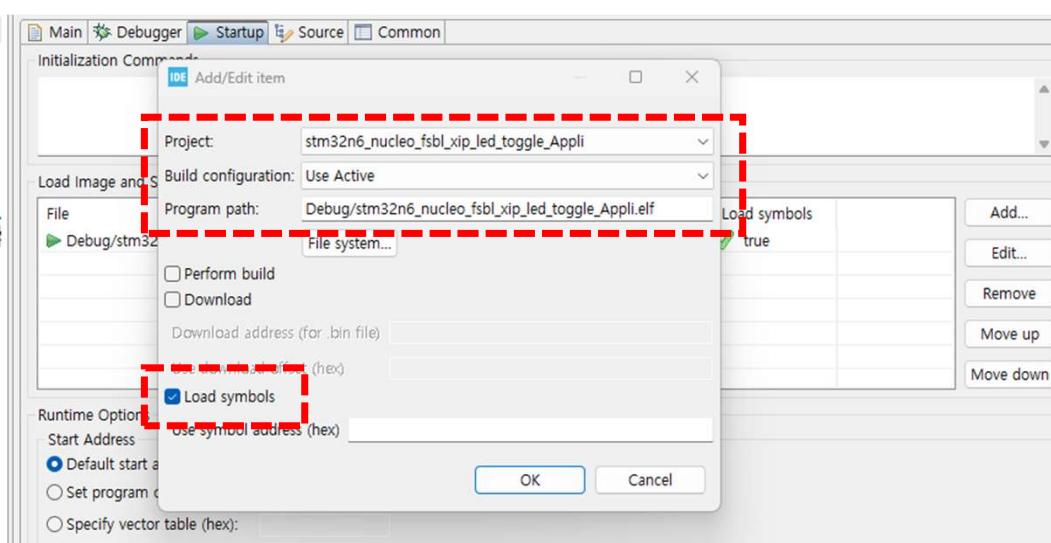
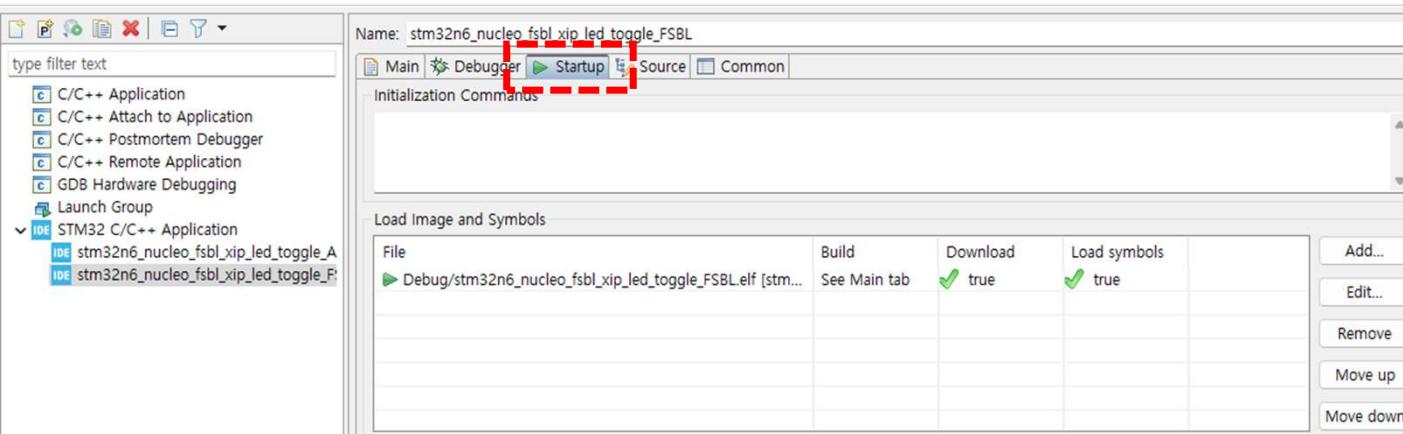


Select the External Loader.

C:\Program
Files\STMicroelectronics\STM32Cube\STM32CubeProgrammer\bin\External
Loader\MX25UM51245G_STM32N6570-NUCLEO.stldr



XIP Debug

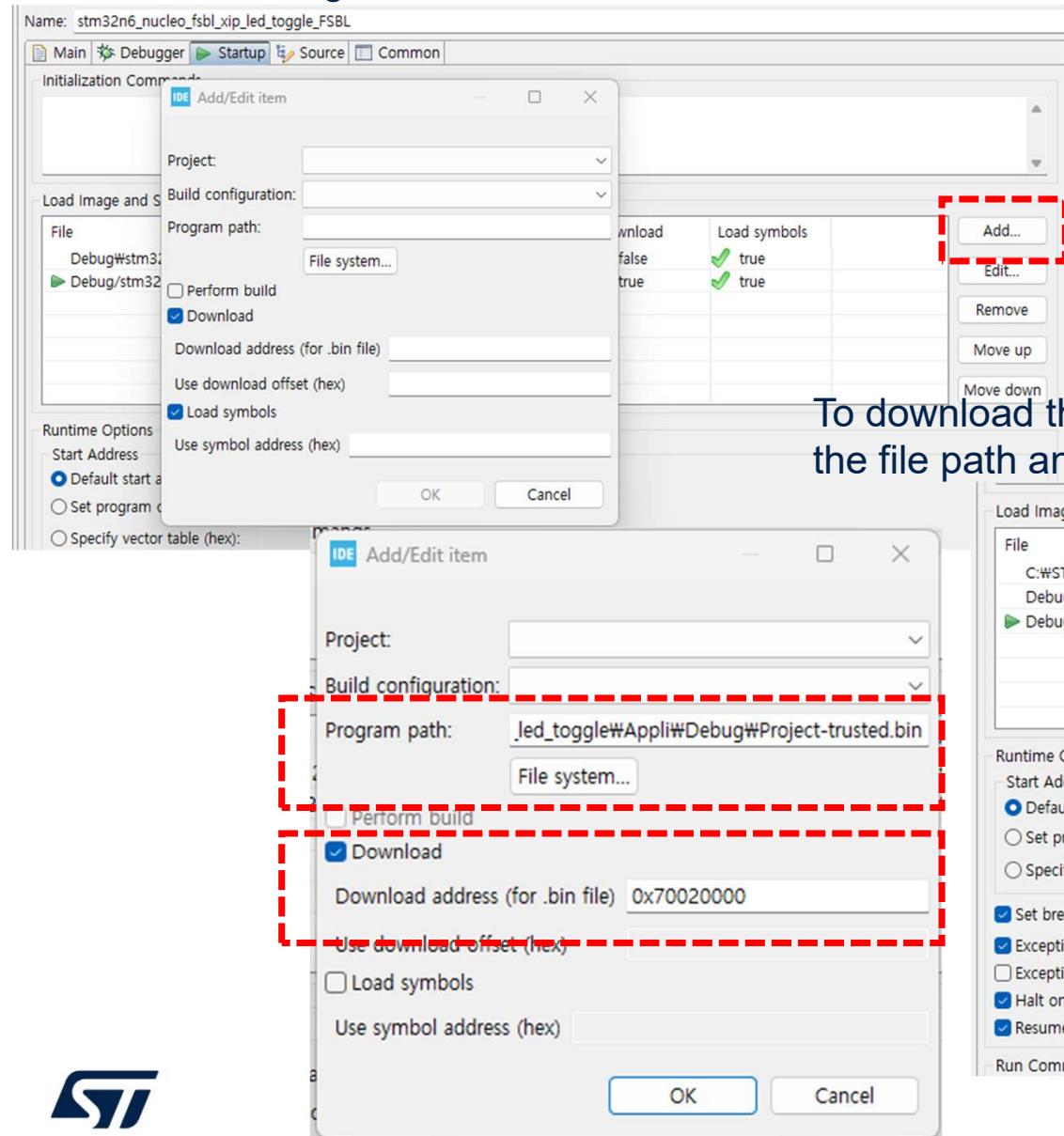


Since you need to enter debug mode through FSBL,
configure FSBL to load the application's symbols.



XIP Debug

To download the signed Project-trusted.bin file, configure the file path and download address.



To download the signed Project-trusted.bin file, configure the file path and download address.

Load Image and Symbols			
File	Build	Download	Load symbols
C:\STM32_Expert_Training_AI_STM32N6\Hands_On\3_Hands...	false	true	false
Debug\stm32n6_nucleo_fsbl_xip_led_toggle_Appli.elf [stm32n6...	false	false	true
Debug\stm32n6_nucleo_fsbl_xip_led_toggle_FSB.elf [stm32n6...	See Main tab	true	true

Runtime Options

Start Address
Default start address
Set program counter (hex)
Specify vector table (hex)

Set breakpoint at: BOOT_Application
 Exception on divide by zero
 Exception on unaligned access
 Halt on exception
 Resume

Run Commands



XIP Debug

IDE workspace_1.18.0 - stm32n6_nucleo_fsbl_xip_led_toggle_FSBL/Middlewares/ST/STM32_ExtMem_Manager/stm32_boot_xip.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug X Project Explorer

IDE stm32n6_nucleo_fsbl_xip_led_toggle_FSBL [STM32 C/C++ Application]

stm32n6_nucleo_fsbl_xip_led_toggle_FSBL.elf [cores: 1]

Thread #1 [main] 1 [core: 1] (Suspended : Breakpoint)

BOOT_Application() at stm32_boot_xip.c:61 0x3418c682

main() at main.c:112 0x341808ee

arm-none-eabi-gdb (14.2.90.20240526)

ST-LINK (ST-LINK GDB server)

stm32_boot_xip.c

```
55
56 BOOTStatus_TypeDef BOOT_Application(void)
57 {
58     BOOTStatus_TypeDef retr;
59
60     /* mount the memory */
61     retr = MapMemory();
62     if (BOOT_OK == retr)
63     {
64         /* jump on the application */
65         retr = JumpToApplication();
66     }
67     return retr;
68 }
69
70 /**
71 * @}
72 */
```

IDE workspace_1.18.0 - C:_Project\N6\Training\stm32n6_nucleo_fsbl_xip_led_toggle\Appli\Core\Src\main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug X Project Explorer

IDE stm32n6_nucleo_fsbl_xip_led_toggle_FSBL [STM32 C/C++ Application]

stm32n6_nucleo_fsbl_xip_led_toggle_FSBL.elf [cores: 1]

Thread #1 [main] 1 [core: 1] (Suspended : Breakpoint)

main() at main.c:96 0x70020810

Reset_Handler() at startup_stm32n657x0hxq.s:99 0x70020e24

arm-none-eabi-gdb (14.2.90.20240526)

ST-LINK (ST-LINK GDB server)

stm32_boot_xip.c main.c stm32n6xx_hal.c main.c

```
84 SystemIsolation_Config();
85 /* USER CODE BEGIN 2 */
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(1000);
97
98     /* USER CODE END WHILE */
99
100    /* USER CODE BEGIN 3 */
101 }
102
103 /* USER CODE END 3 */
```

Enter FSBL in debug mode, jump to the application address, and then debug.



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

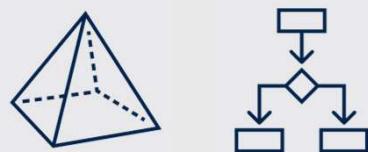


Any technique that enables computer to mimic **human behavior**

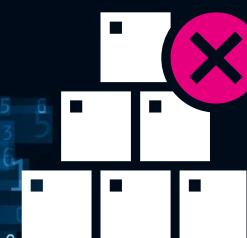
Subset of AI. Algorithms and methodologies that improve over time through **learning from data**

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**

AI offers the best approach to process growing amounts of data



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

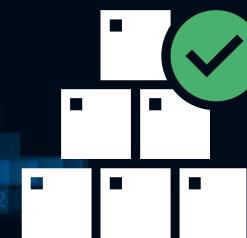


Traditional approaches show their limitations:

- 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



Desired output from the system

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

Machine learning

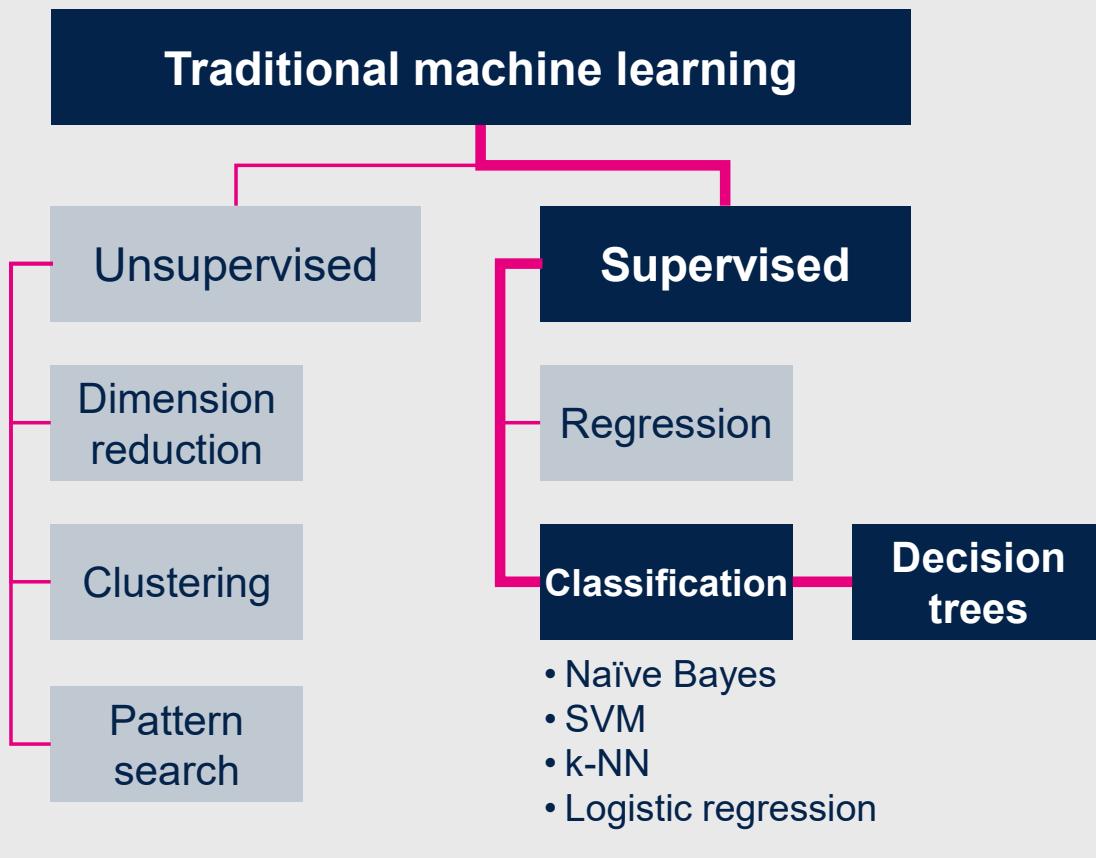
Rules learned from real-world data



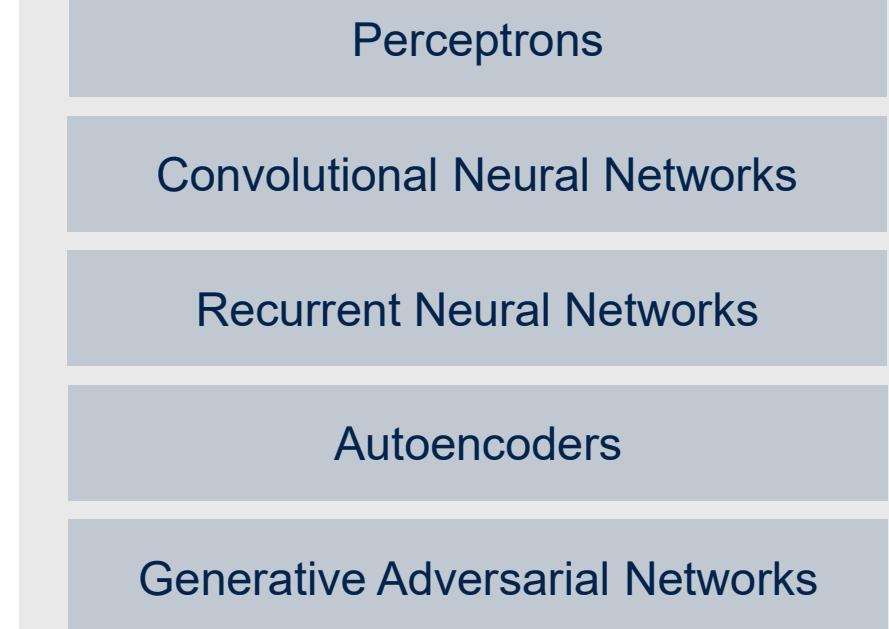
General ML model trained for the problem

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

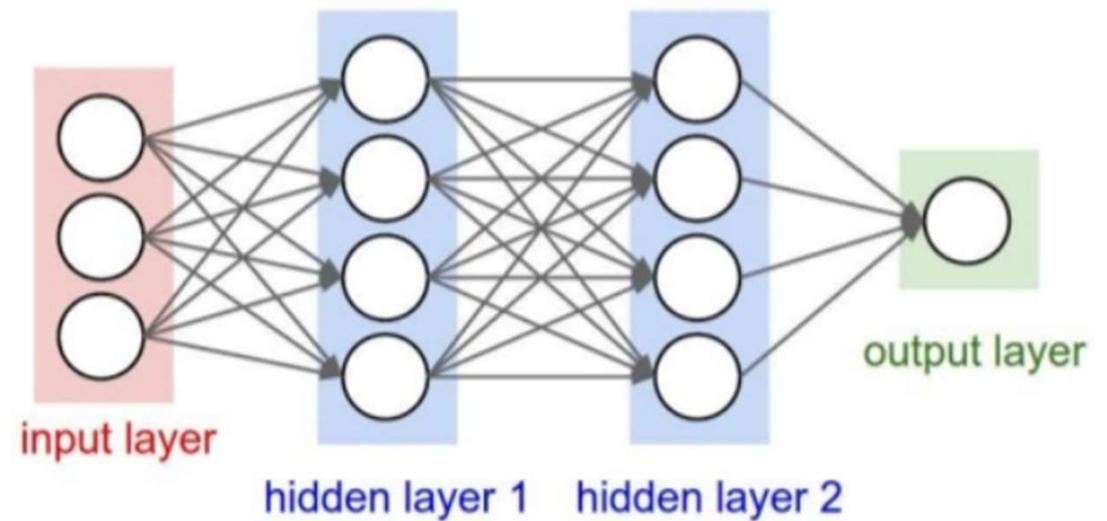
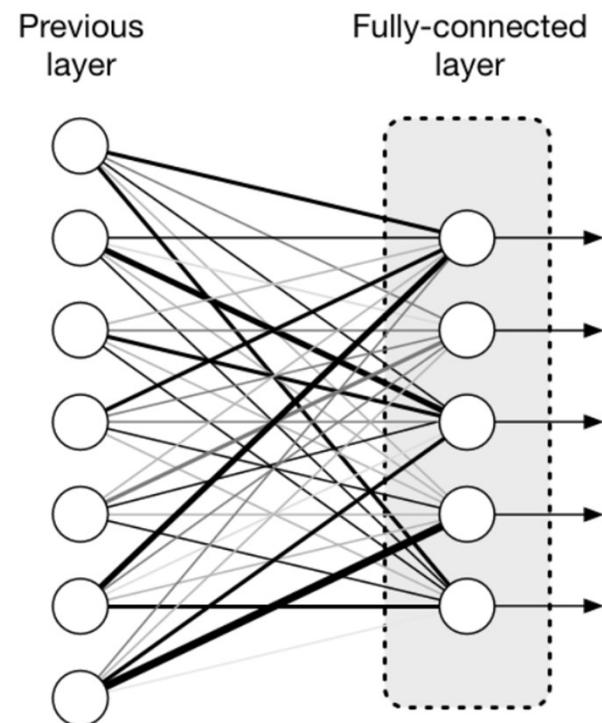
Machine learning



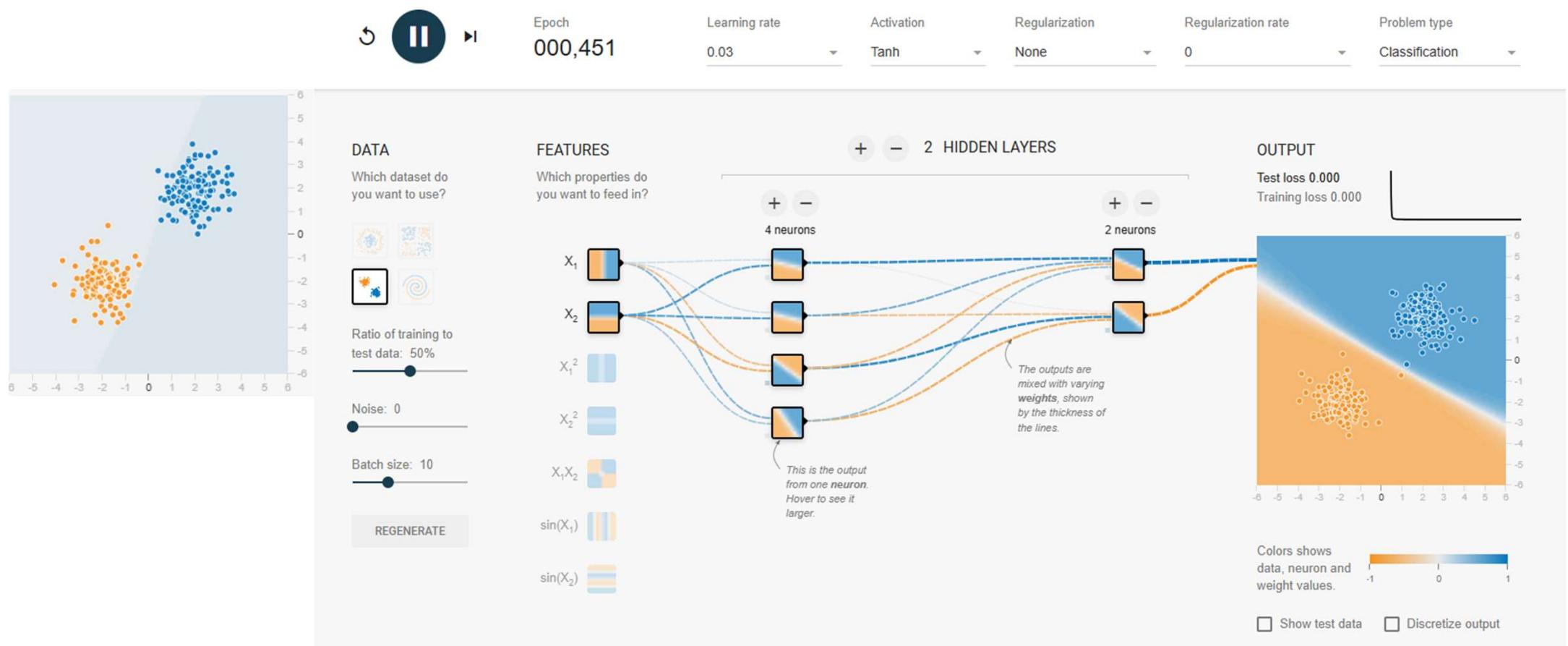
Deep learning



What is a Neural Network?

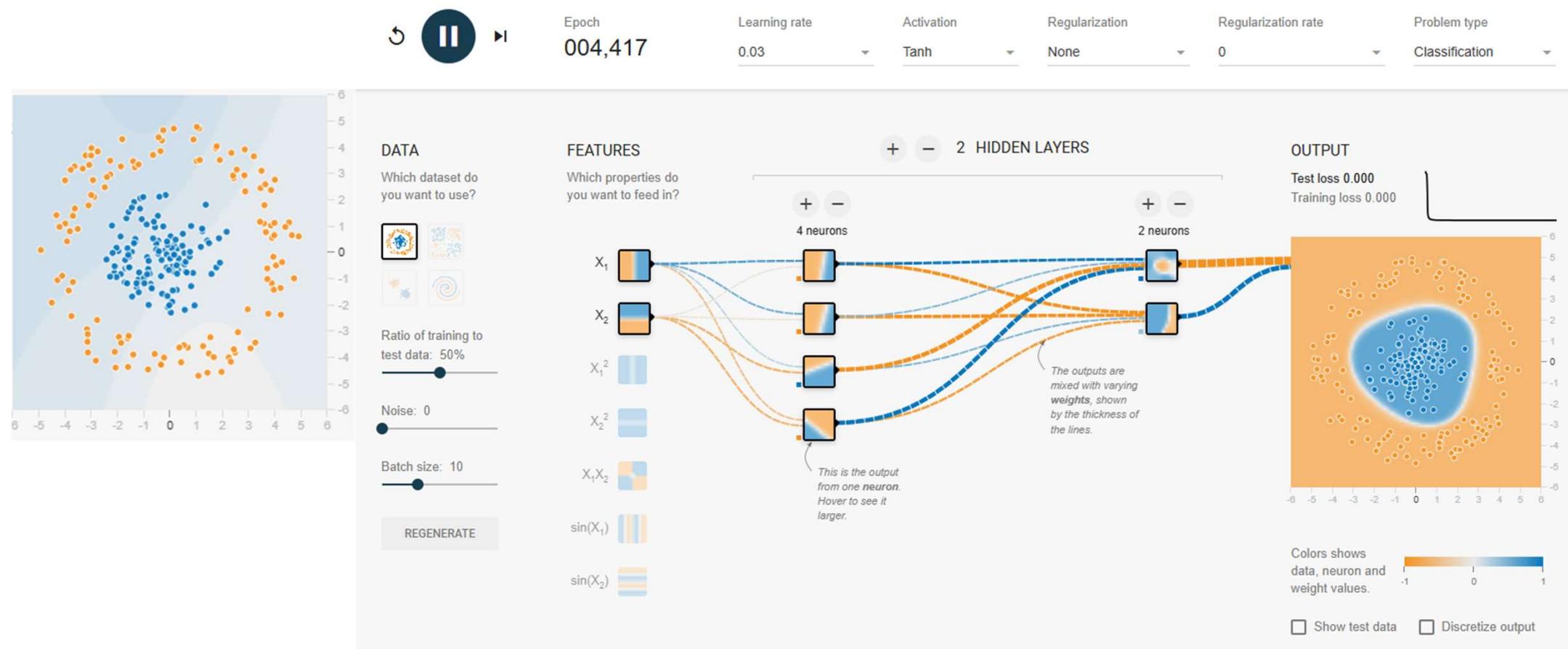


What is a Neural Network?



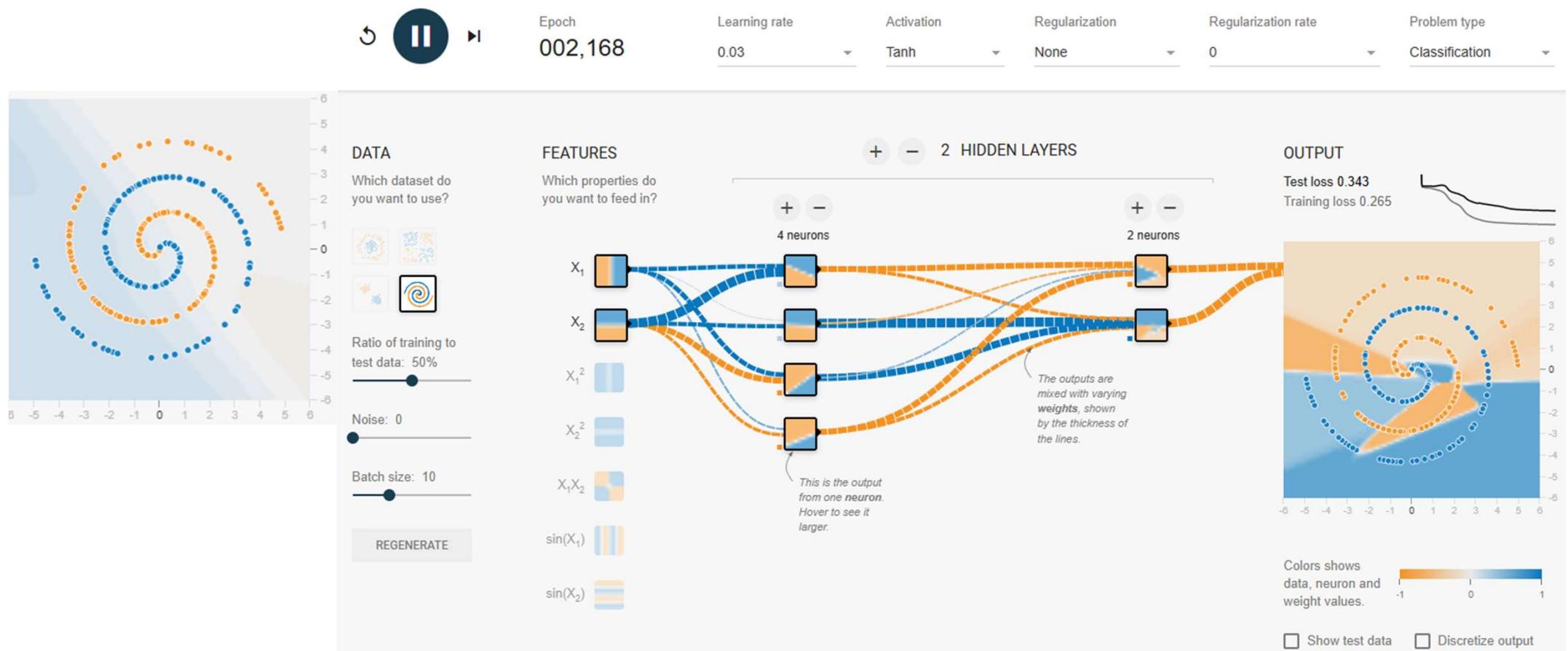
<https://playground.tensorflow.org>

What is a Neural Network?



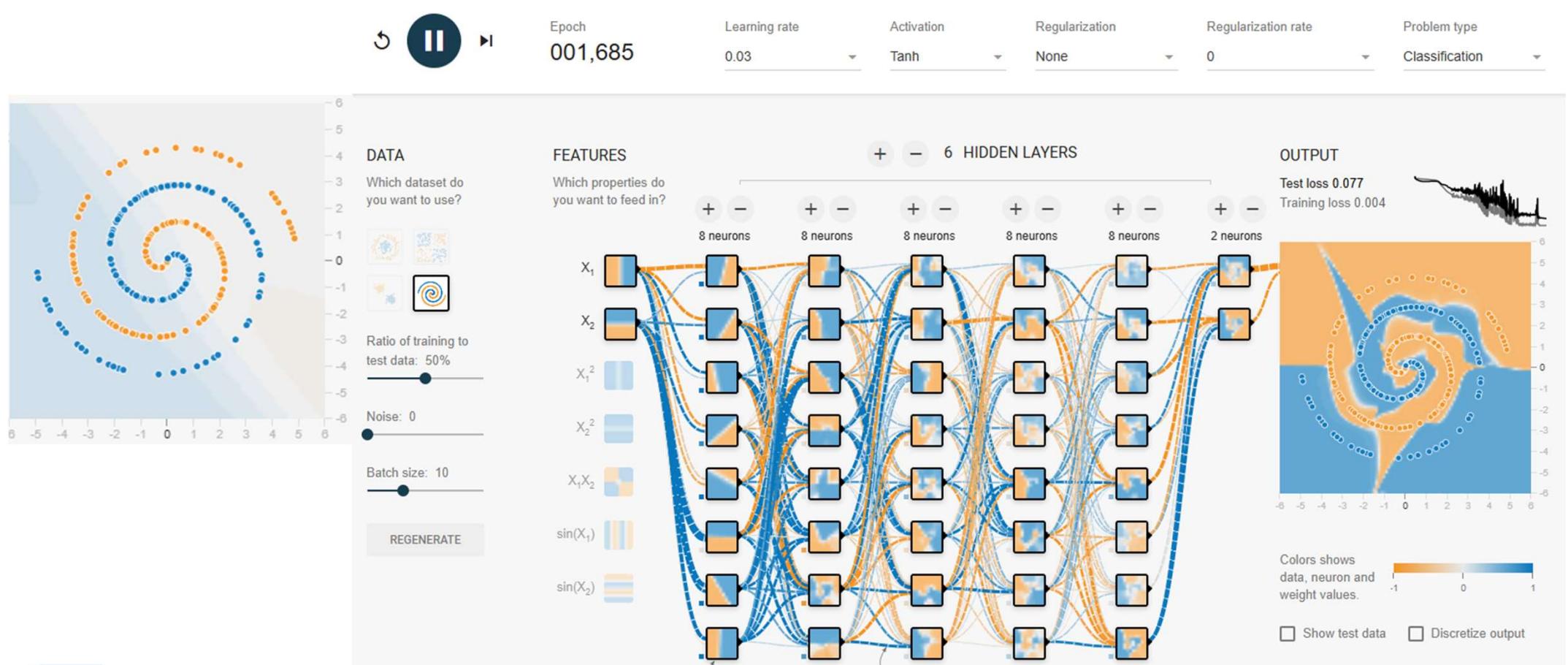
<https://playground.tensorflow.org>

What is a Neural Network?



<https://playground.tensorflow.org>

What is a Neural Network?



<https://playground.tensorflow.org>

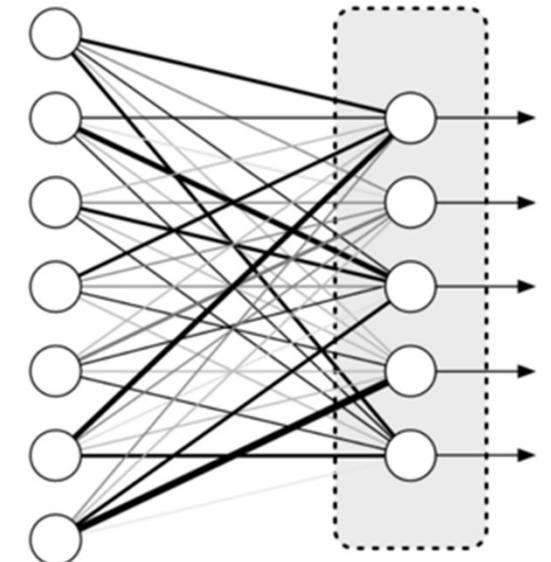
Convolution Neural Network

Hands on



Convolution Neural Network

Hands on



Convolution Neural Network

Channel

RED Channel



Green Channel



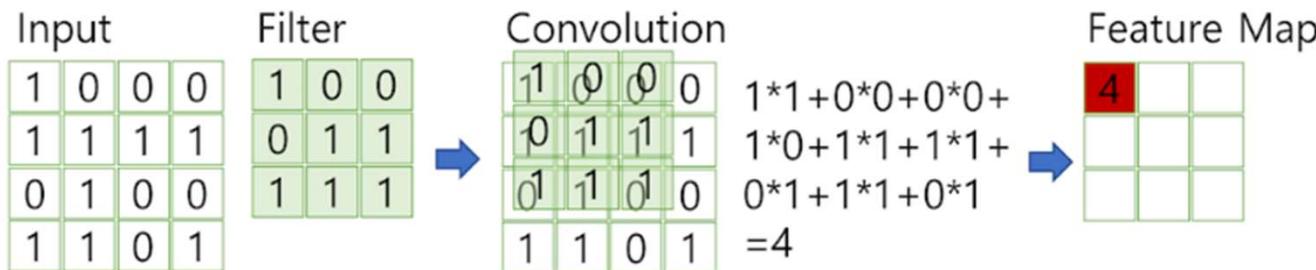
Blue Channel



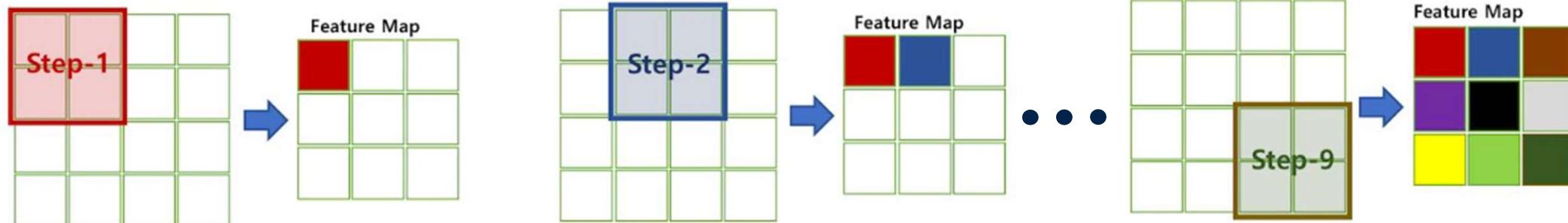
[https://en.wikipedia.org/wiki/Channel_\(digital_image\)](https://en.wikipedia.org/wiki/Channel_(digital_image))

Convolution Neural Network

Filter and Stride



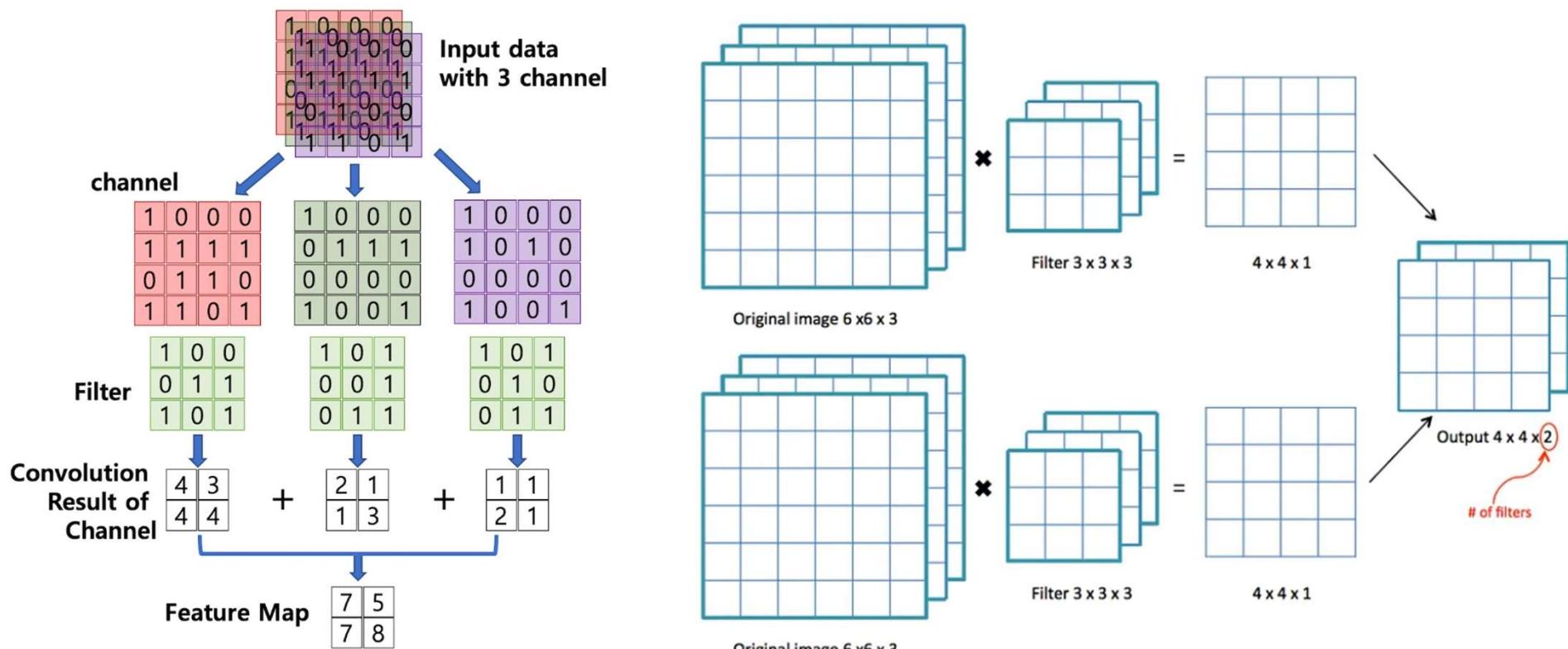
<http://taewan.kim>



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

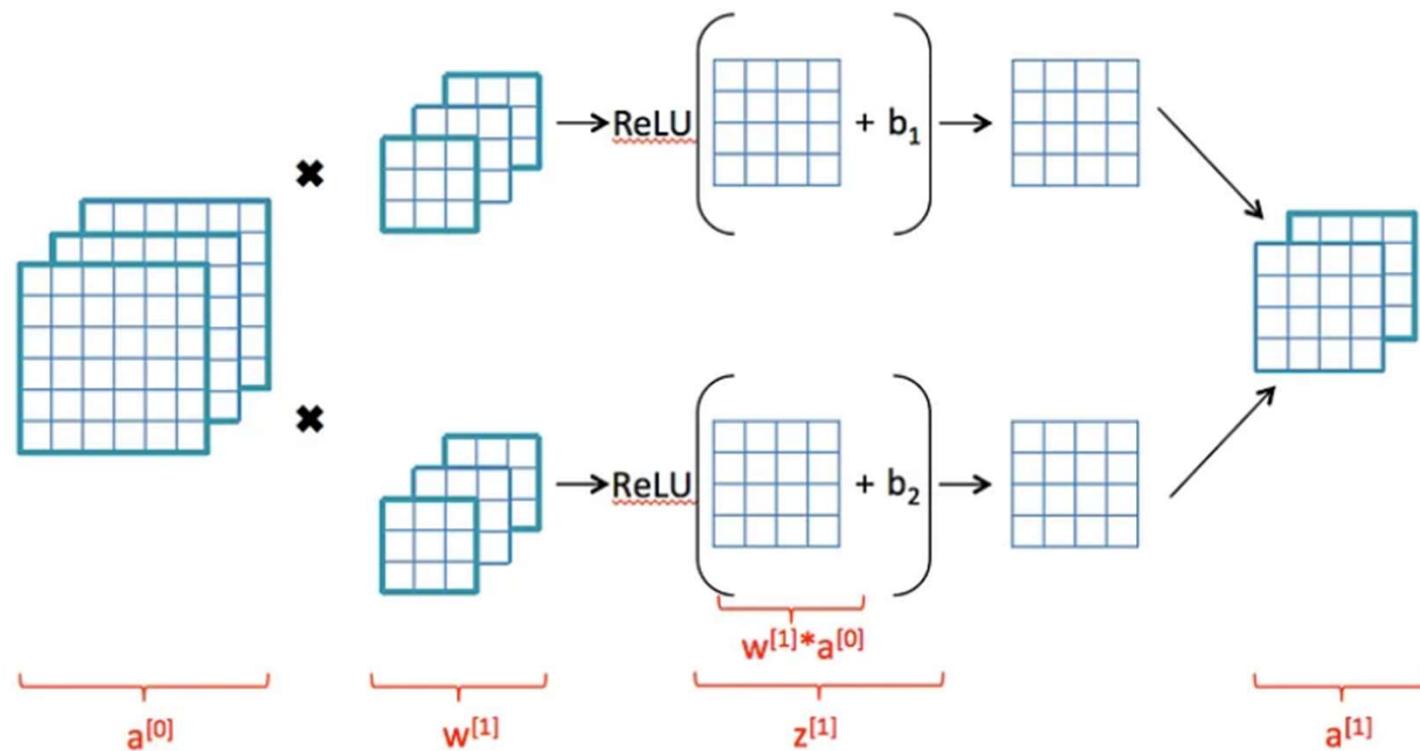
Convolution Neural Network

Hands on



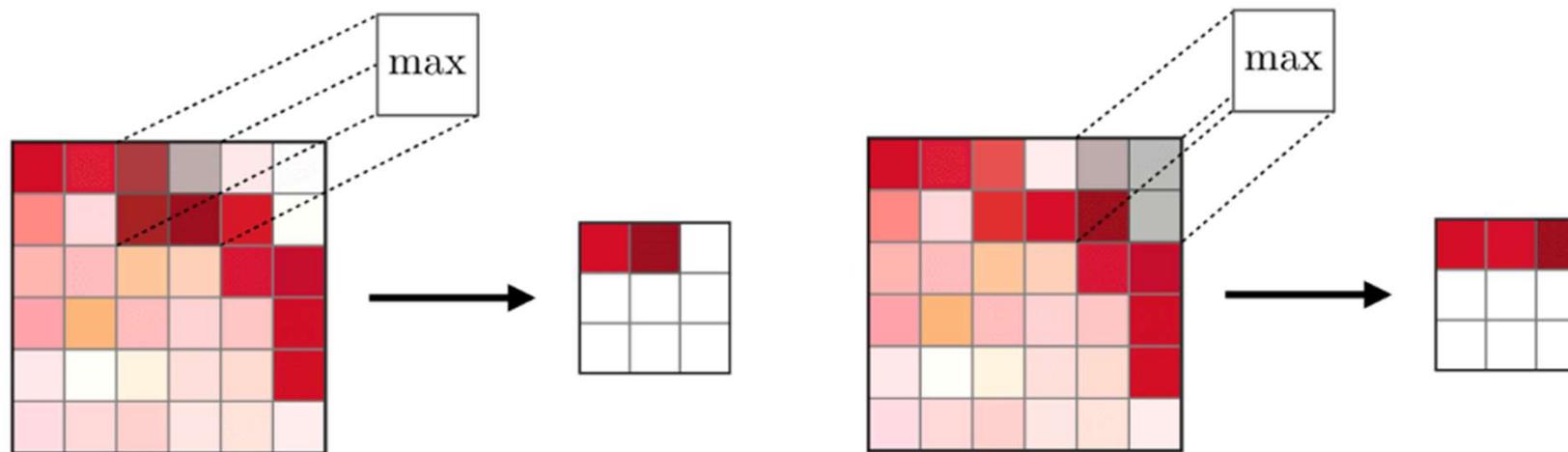
Convolution Neural Network

Hands on



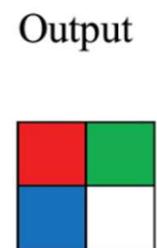
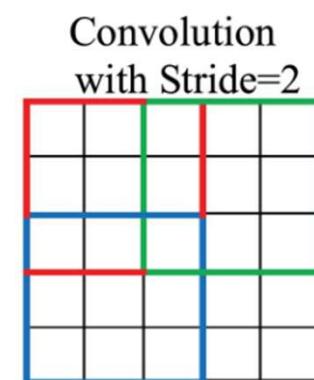
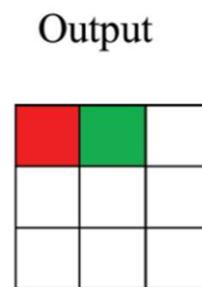
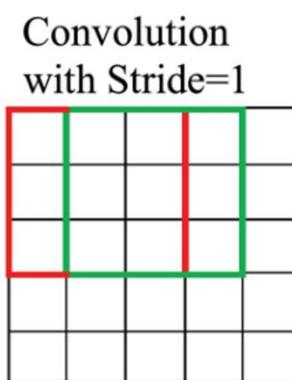
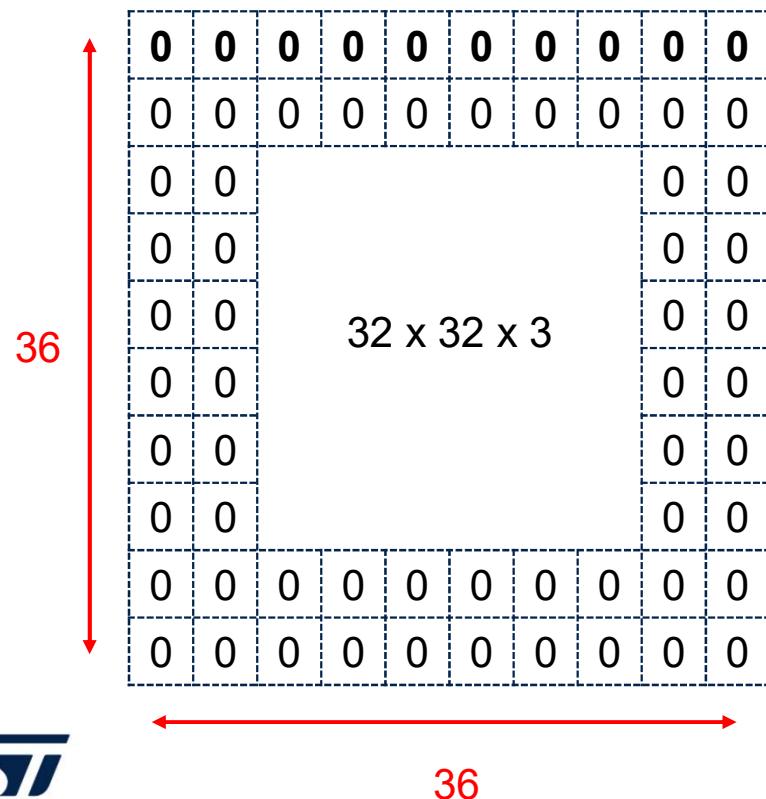
Convolution Neural Network

Pooling



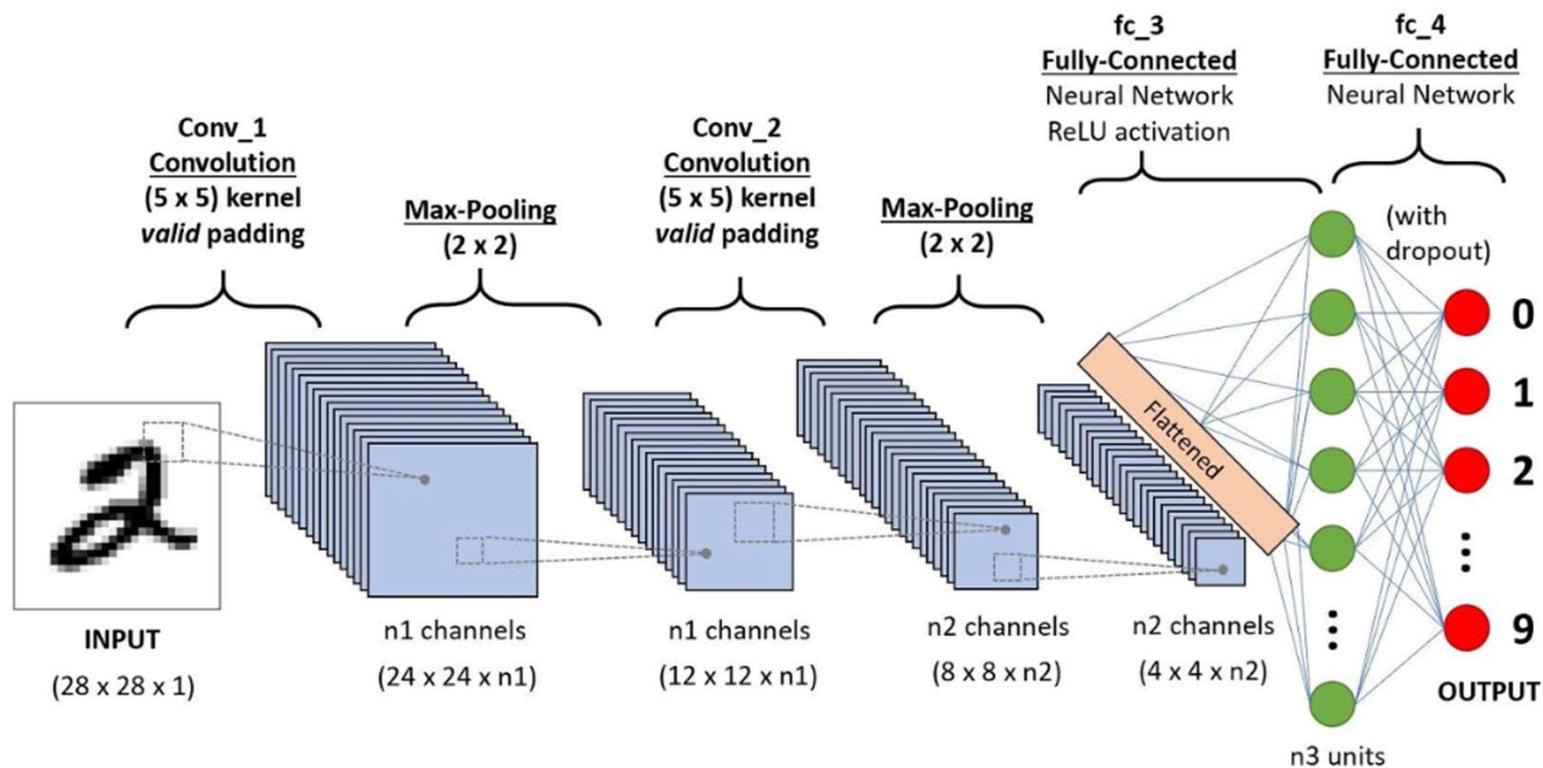
Convolution Neural Network

Padding and Stride



Convolution Neural Network

Hands on



Convolution Neural Network

Hands on

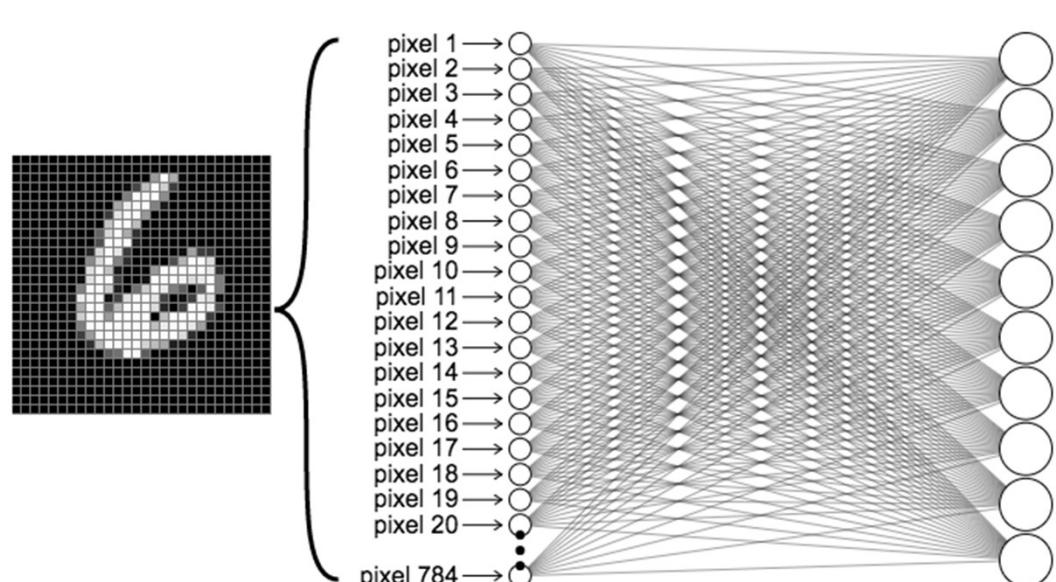
		Activation Shape	Activation Size	#Parameters
1	Input Layer :	(32, 32, 3)	3072	0
2	CONV1 (f=5, s=1)	(28, 28, 8)	6272	608
3	POOL1	(14, 14, 8)	1568	0
4	CONV2 (f=5, s=1)	(10, 10, 16)	1600	3216
5	POOL2	(5, 5, 16)	400	0
6	FC3	(120, 1)	120	48120
7	FC4	(84, 1)	84	10164
8	Softmax	(10, 1)	10	850



Convolution Neural Network

Hands on – mnist

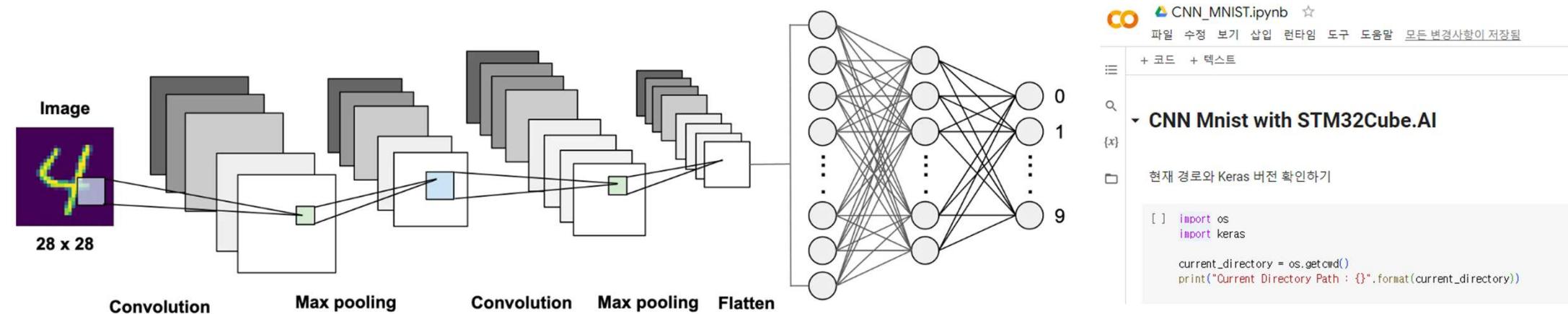
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9



https://ml4a.github.io/ml4a/looking_inside_neural_nets/

Convolution Neural Network

Hands on – mnist

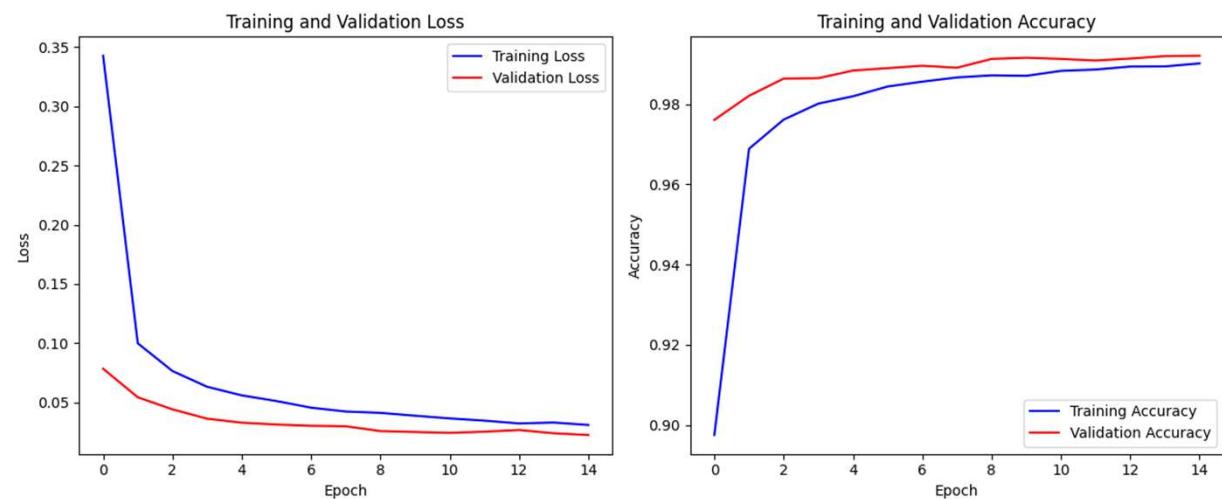
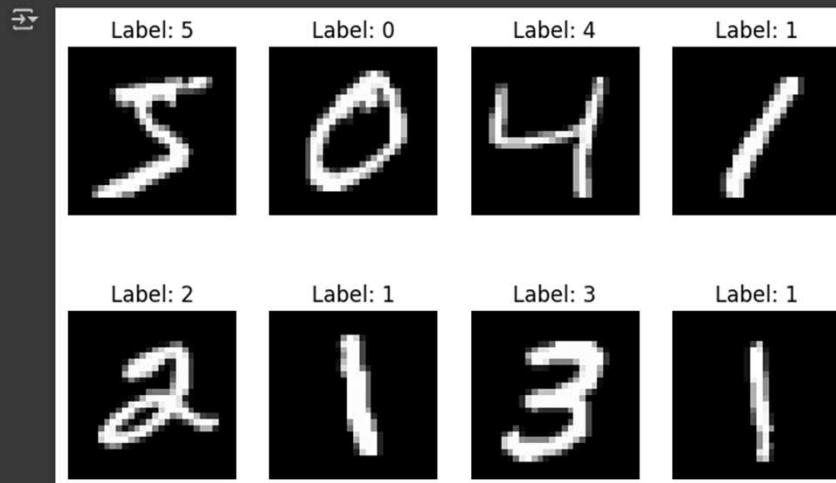


Training

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.

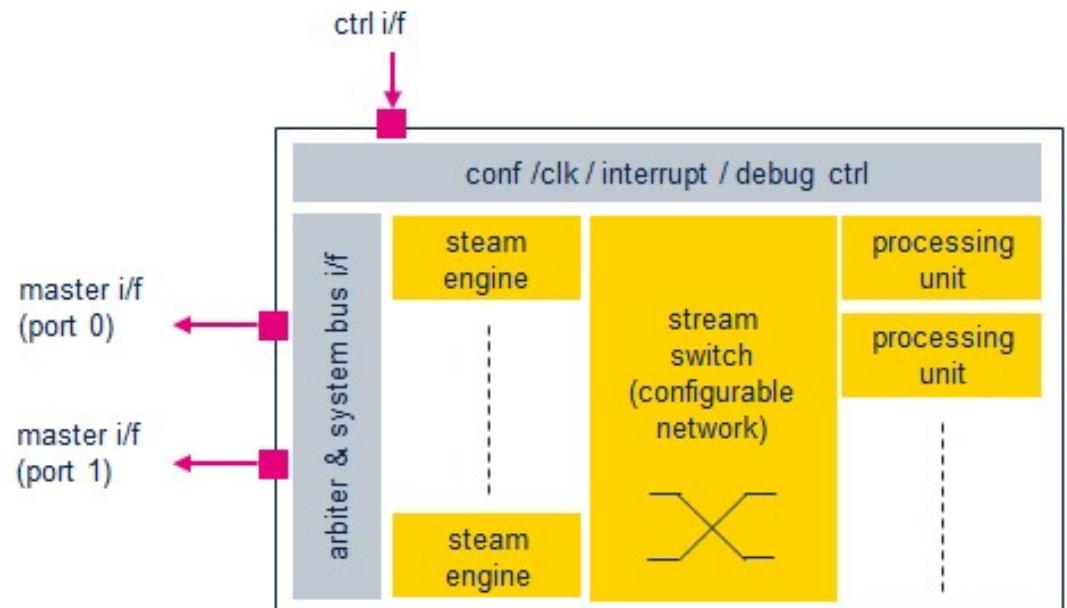
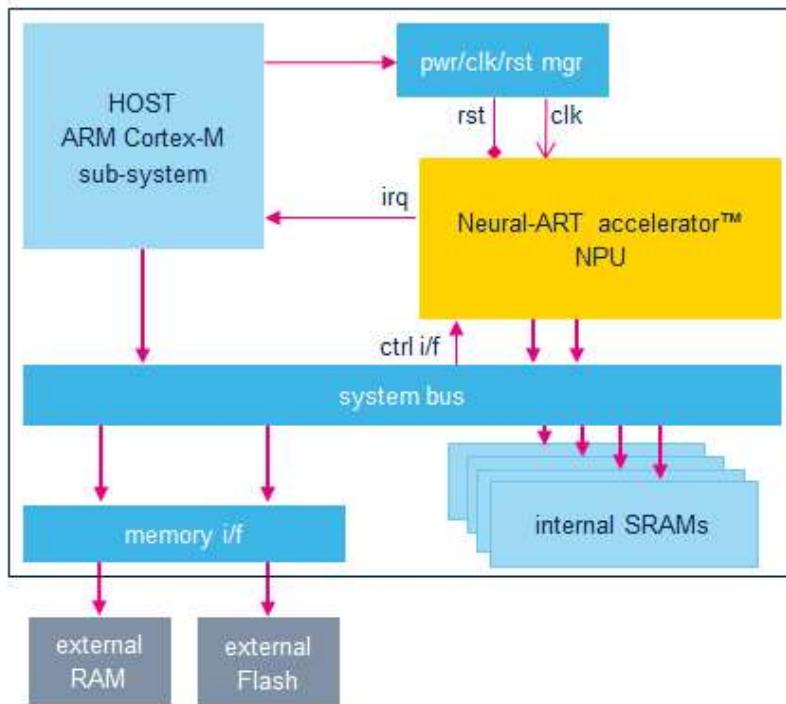


https://github.com/BlaineMoon30/STM32_Expert_Training_AI_STM32N6/tree/master/Hands_On_2_Hands_On_NPU_Mnist/Mnist_Model/Script

ST Neural-ART Accelerator™



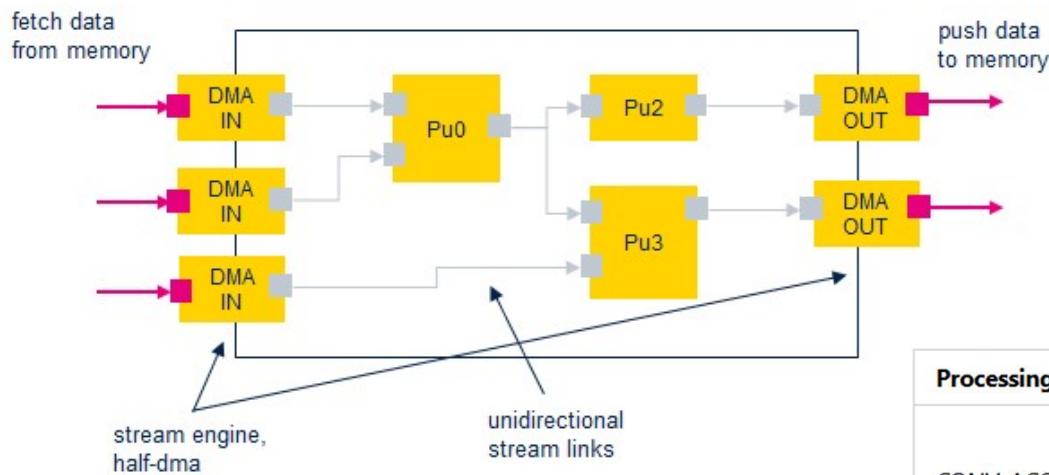
ST Neural-ART Accelerator™



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



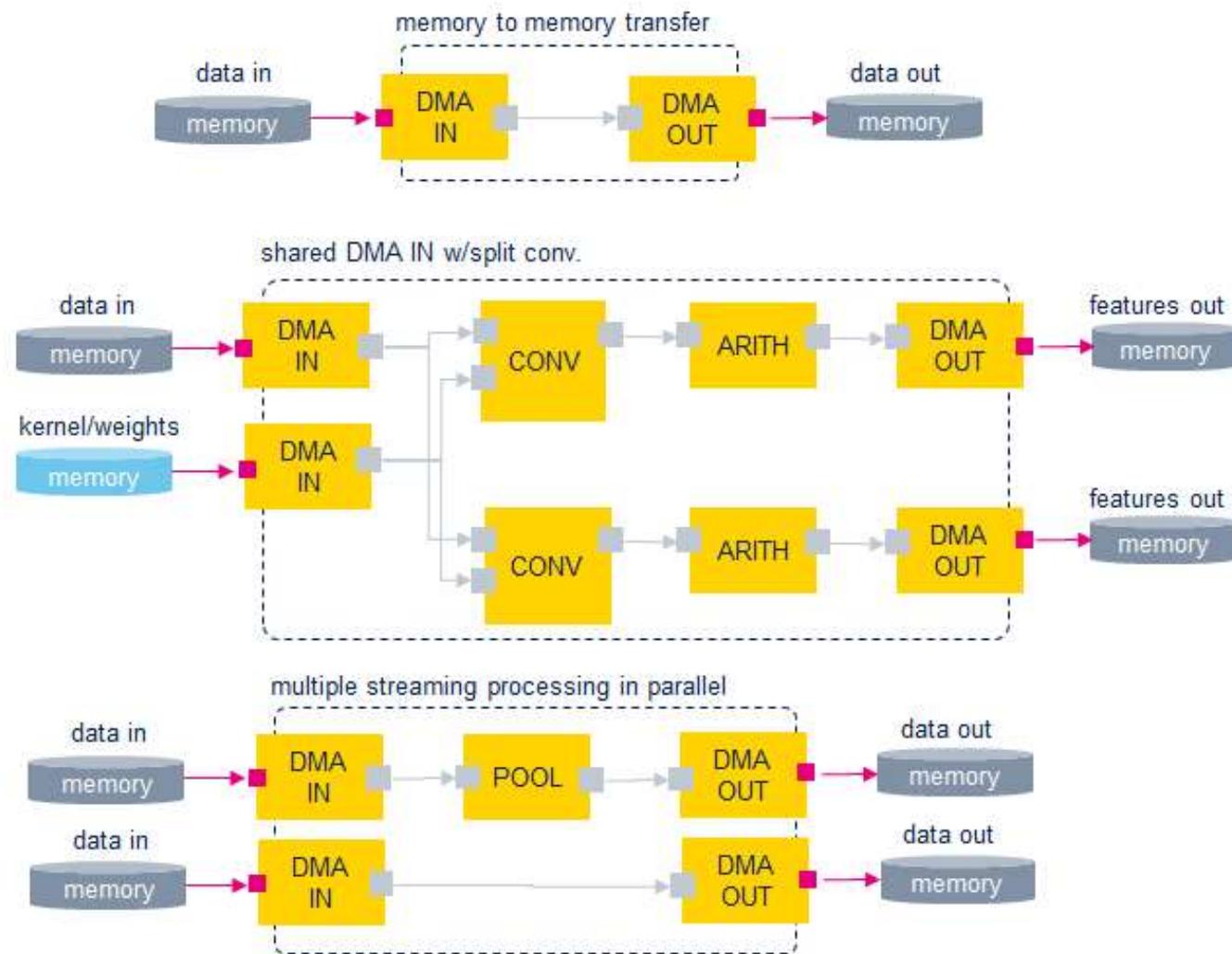
Streaming-based architecture



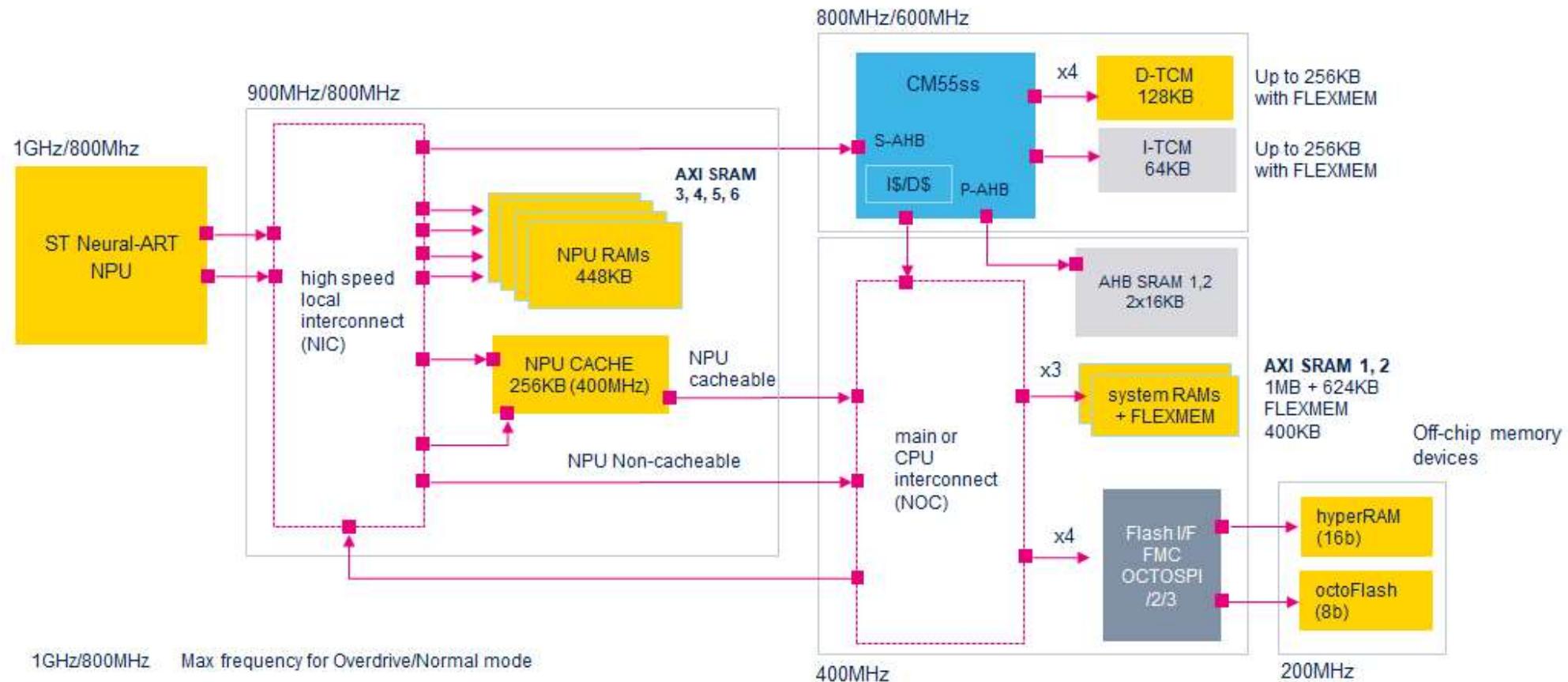
The stream engine units function as intelligent half-DMAs, capable of reading and writing data to and from external memory.

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 600+ GOPs including the operations from the ACTIV/ARITH/POOL units.
POOL_ACC (x2)	Perform the pooling operations like local 2D windowed (NxN), 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.

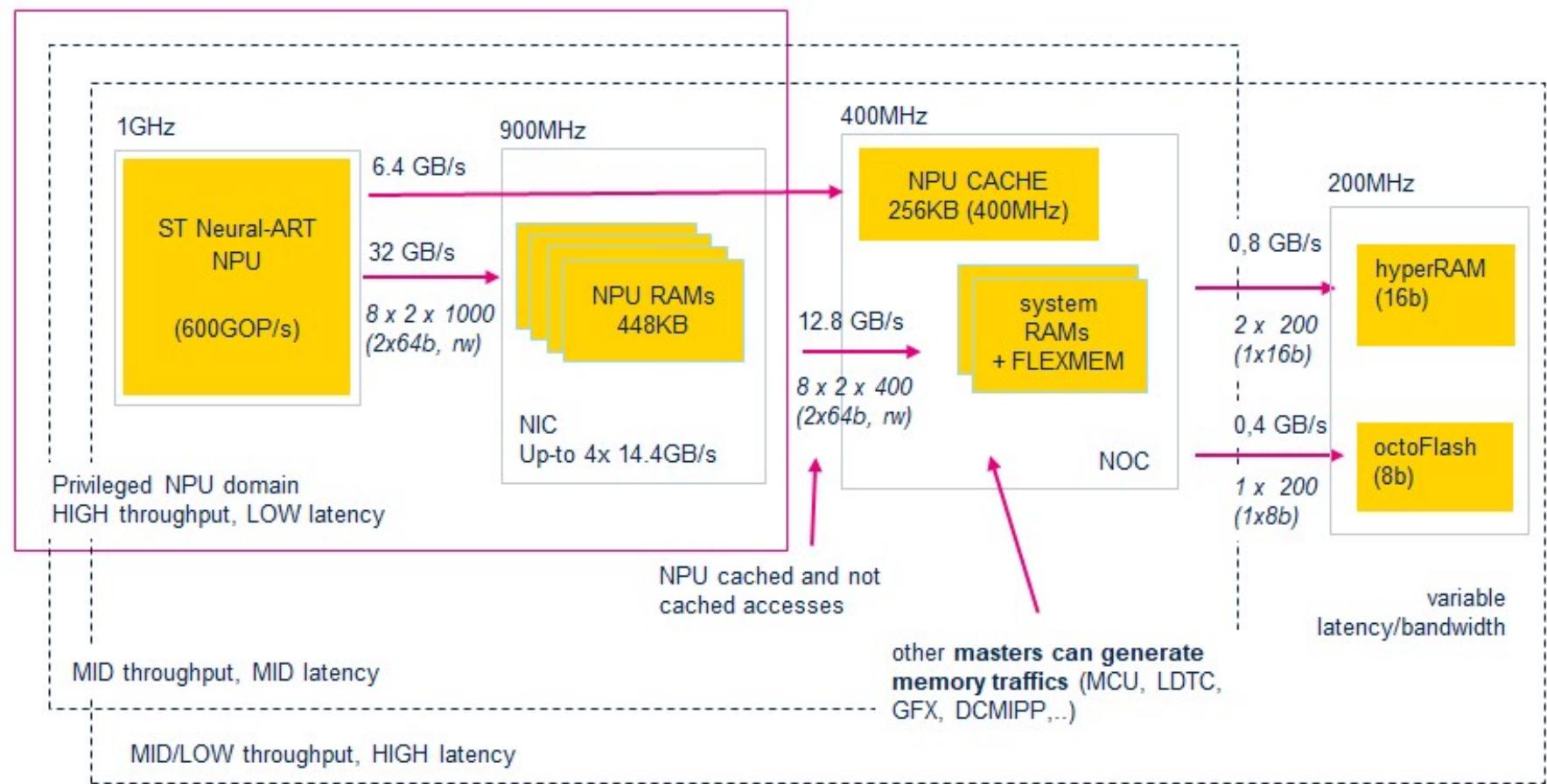
Data processing paths



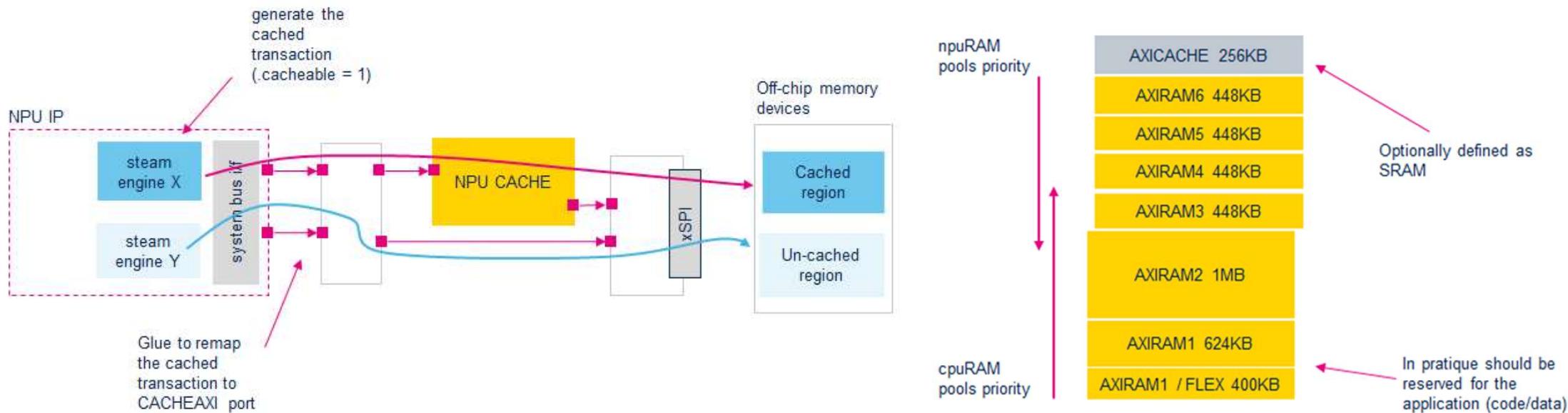
NPU Memory System



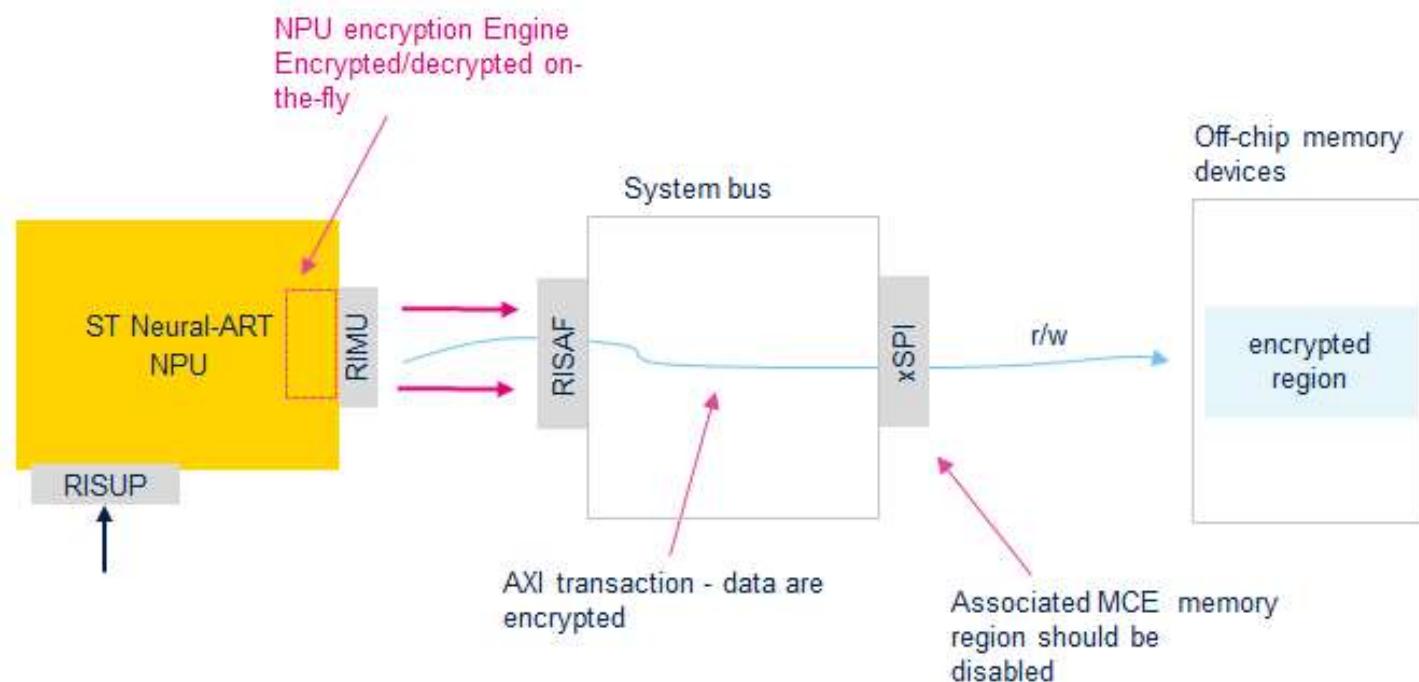
Processing-Bound vs Memory-Bound Operation



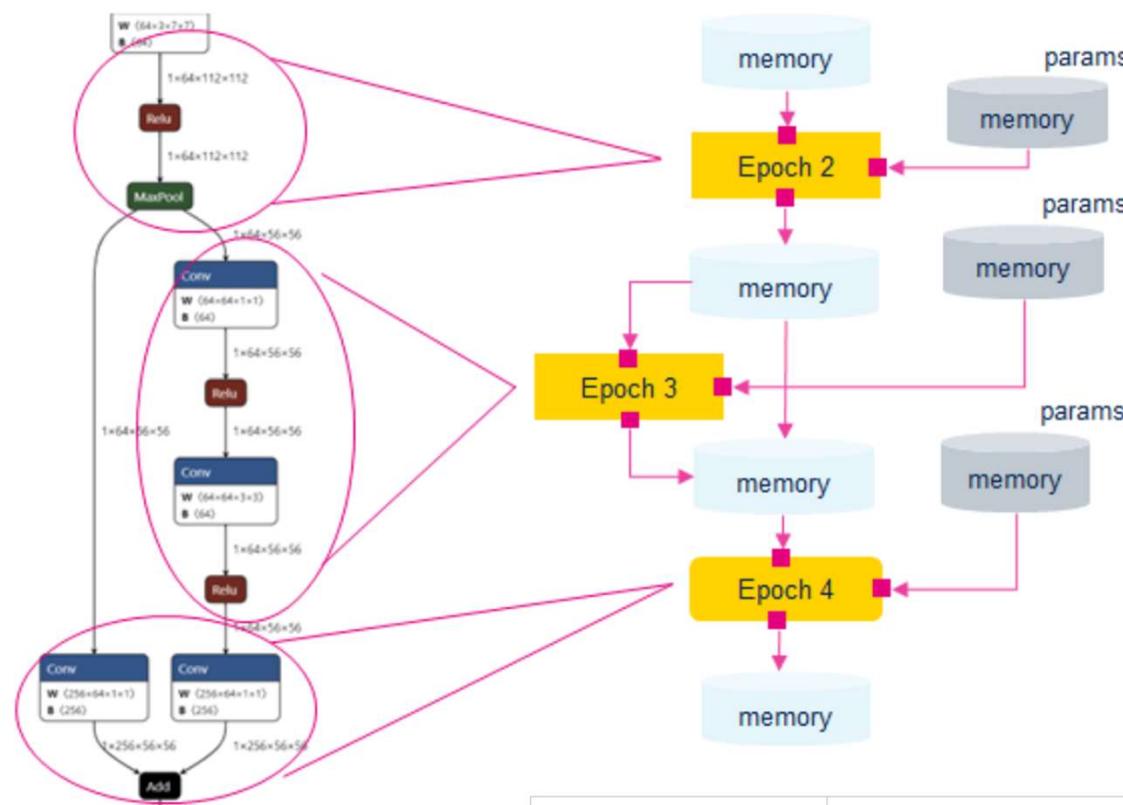
NPU Cache and Virtual memory pool



Security considerations



Programming model



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

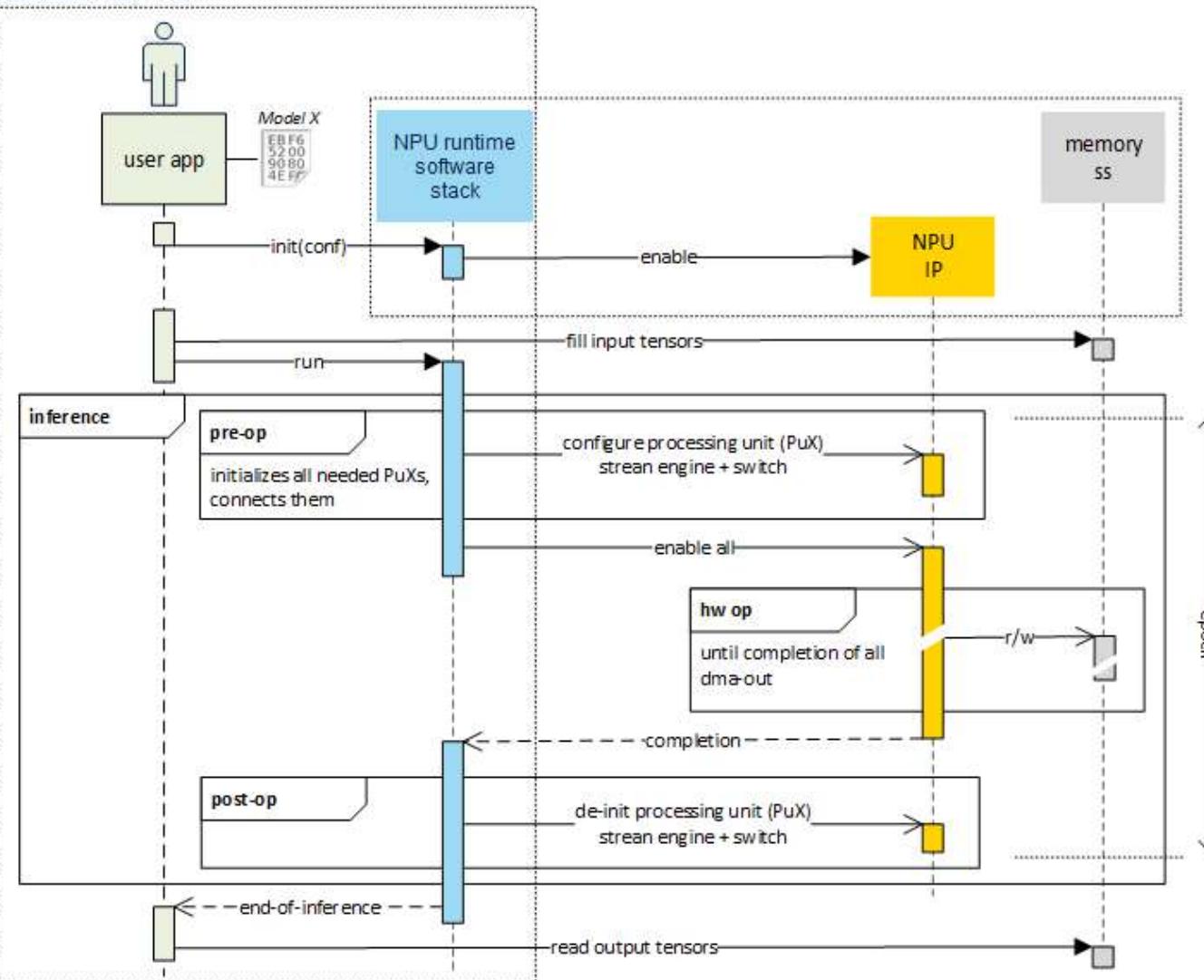
It must be split into elementary subsets, named *NPU epochs* (or 'epoch'), which fit the NPU's available resources.

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 epoch controller unit .

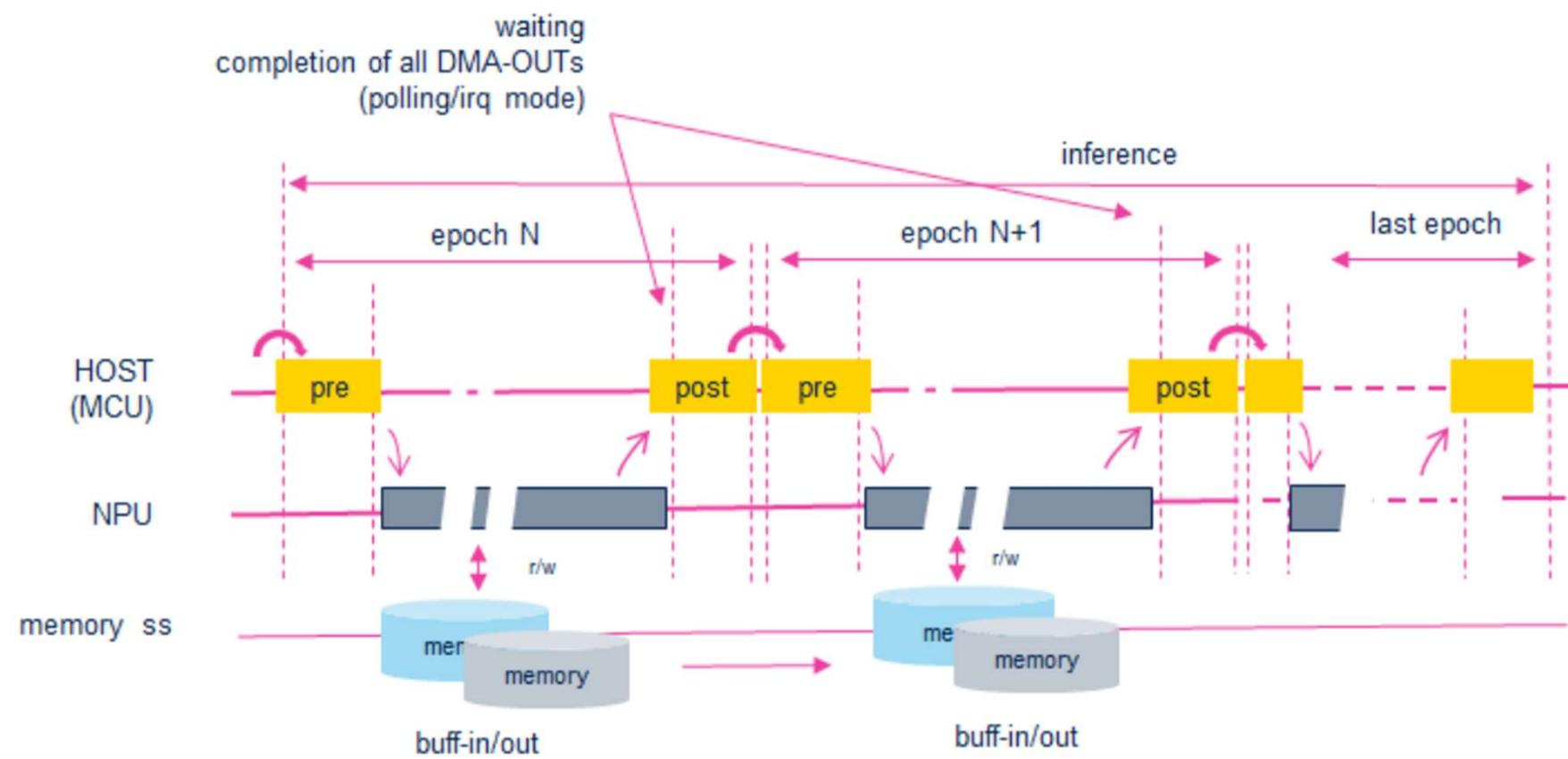
Scheduling

Executed on the HOST

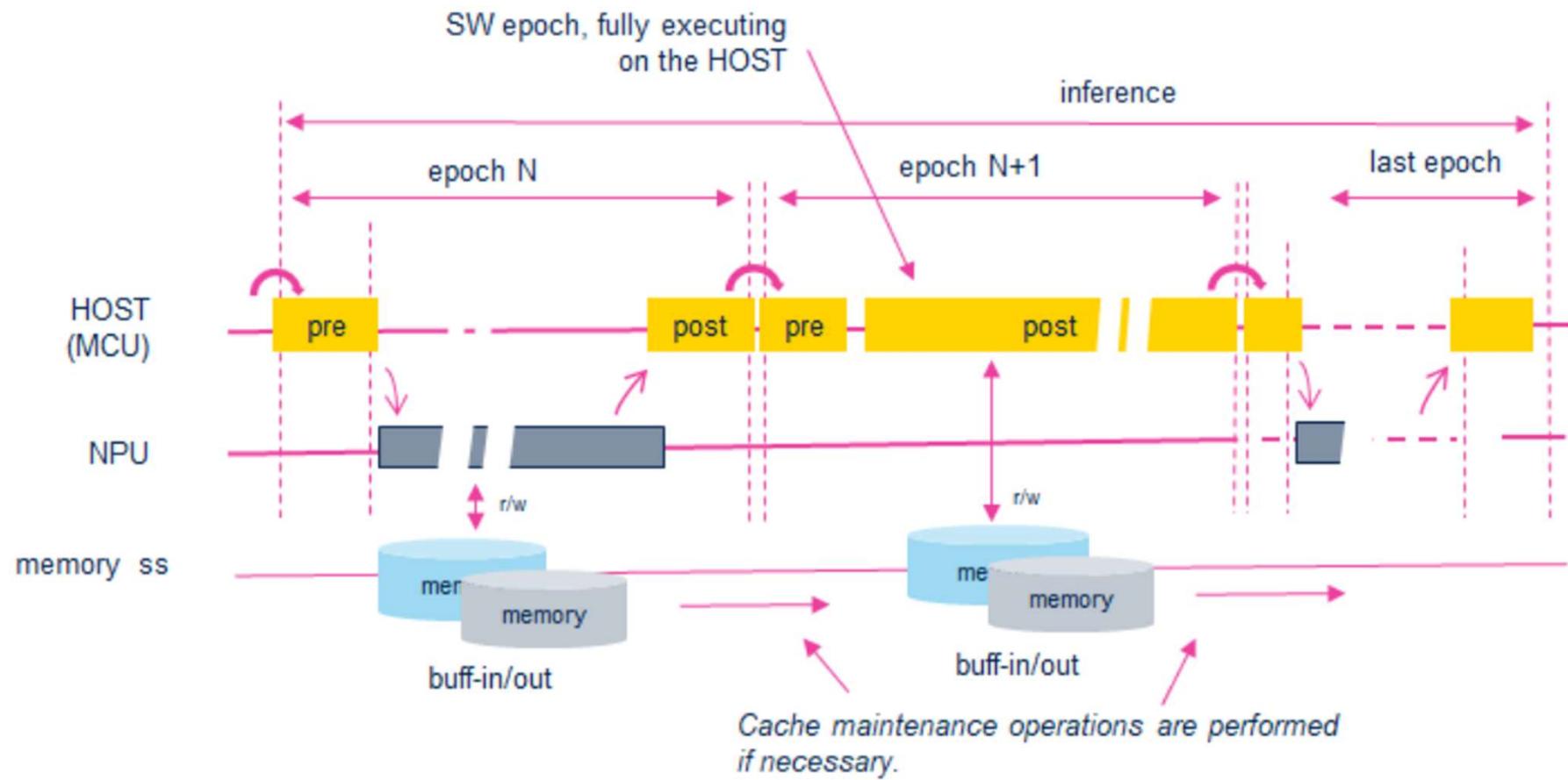


As part of the [NPU runtime software stack](#), a lightweight scheduler engine is responsible for executing the list of different epochs. Each epoch is considered an *atomic operation*, and its execution is ordered and fixed to ensure data dependency across the entire computational graph.

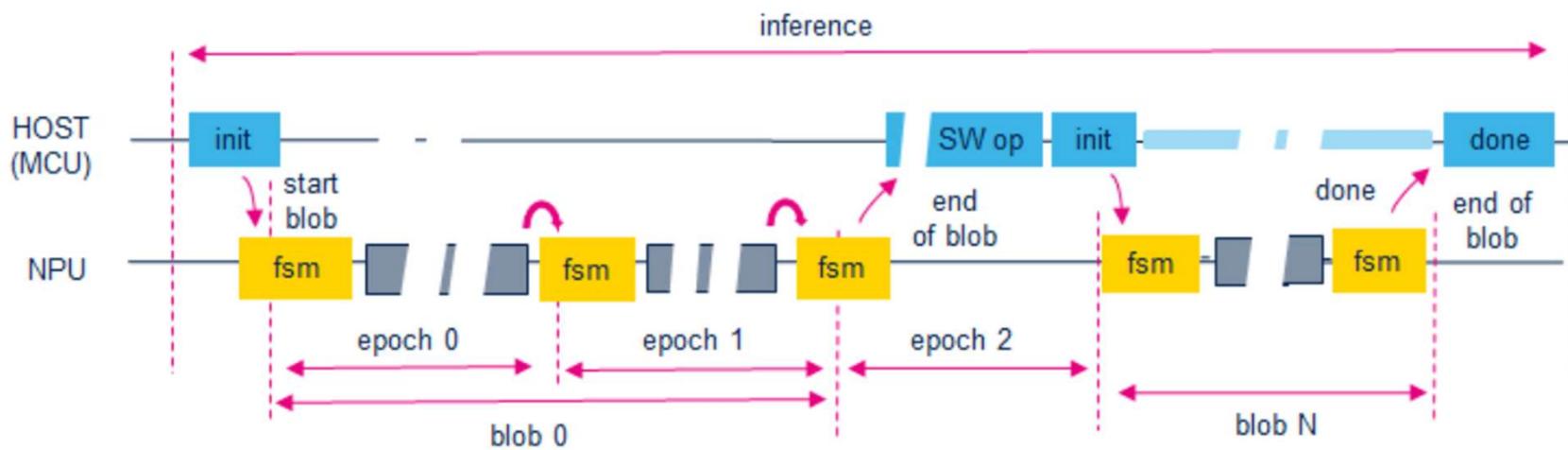
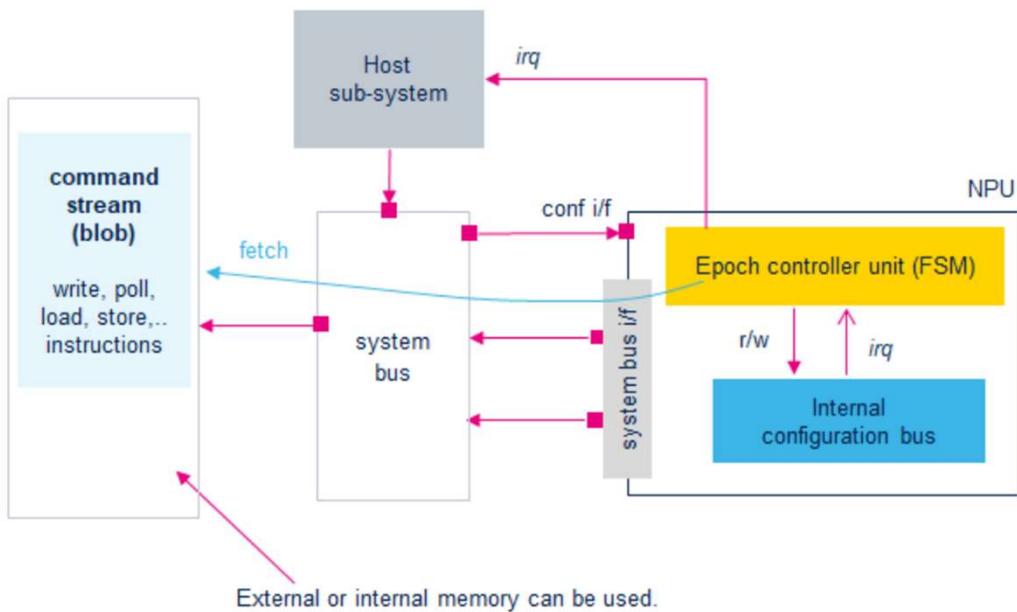
HW epochs only



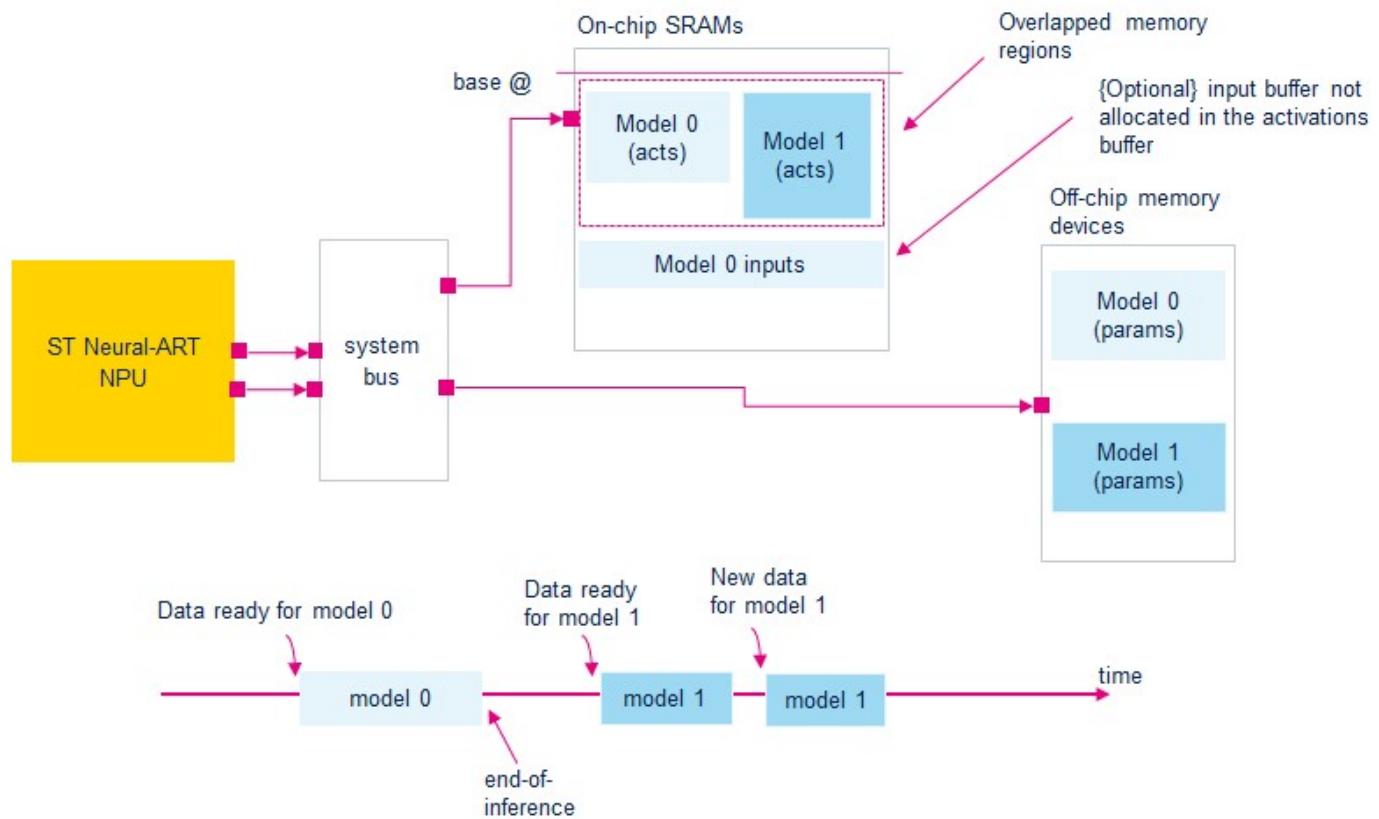
Including SW epoch



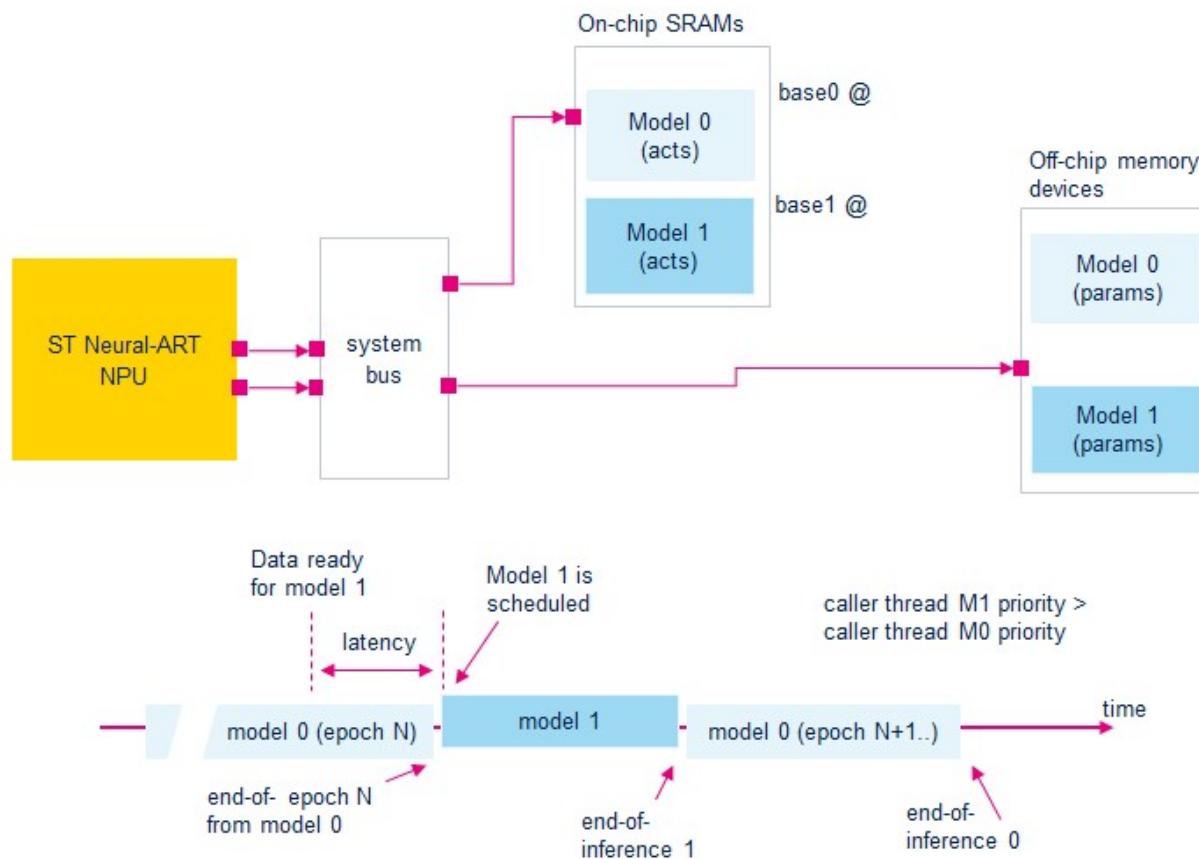
Epoch controller mode



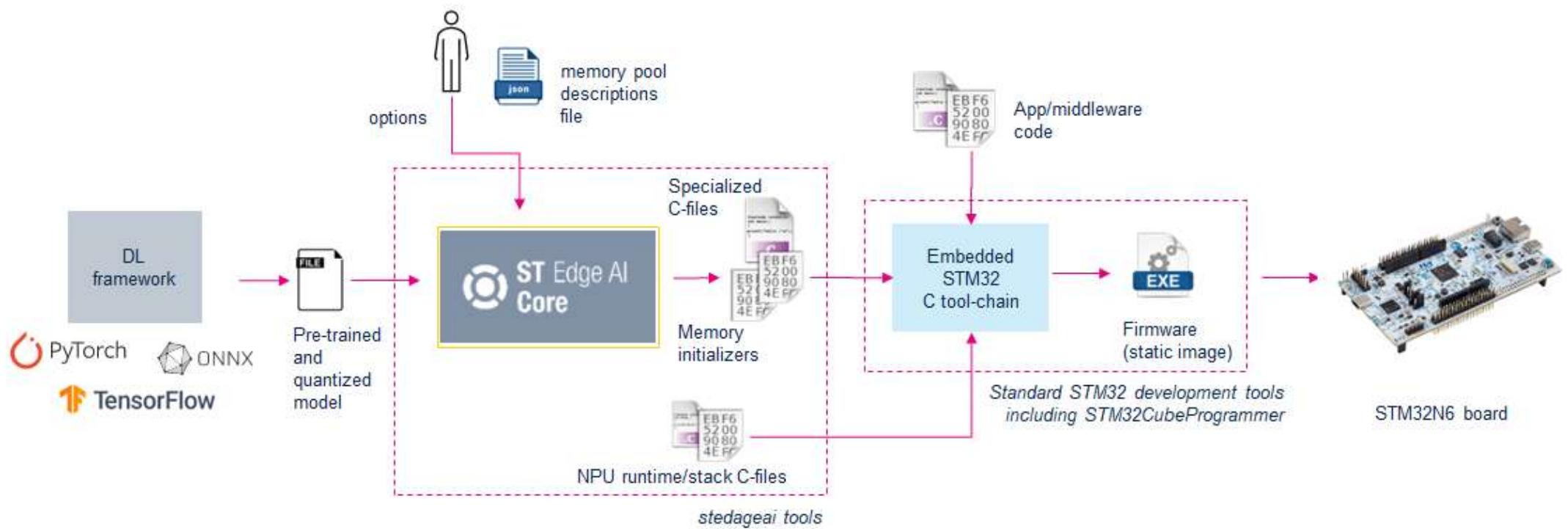
Multiple models support – Serial mode



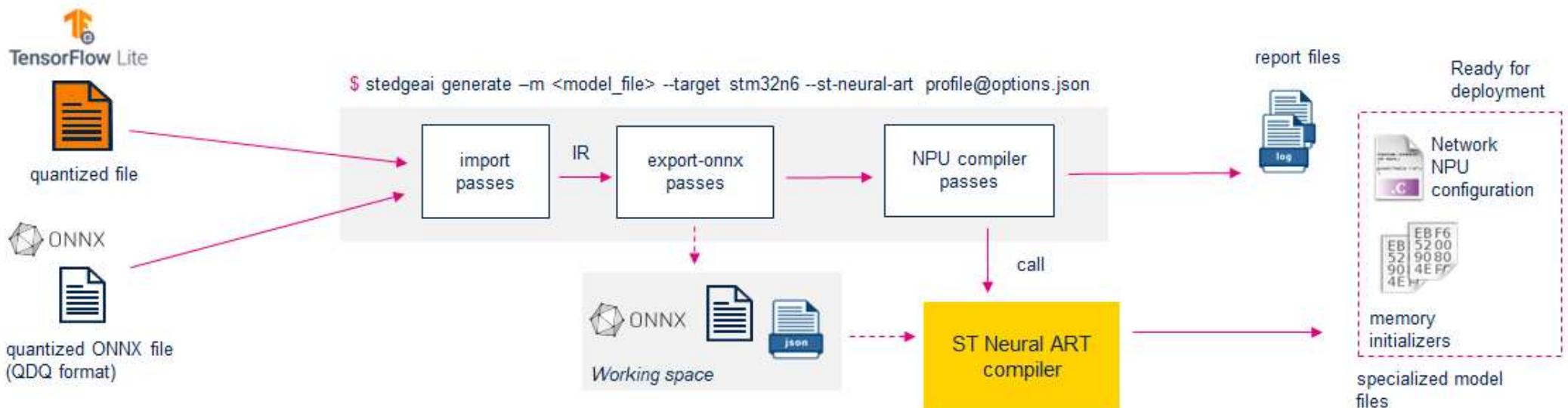
Multiple models support - Epoch level mode



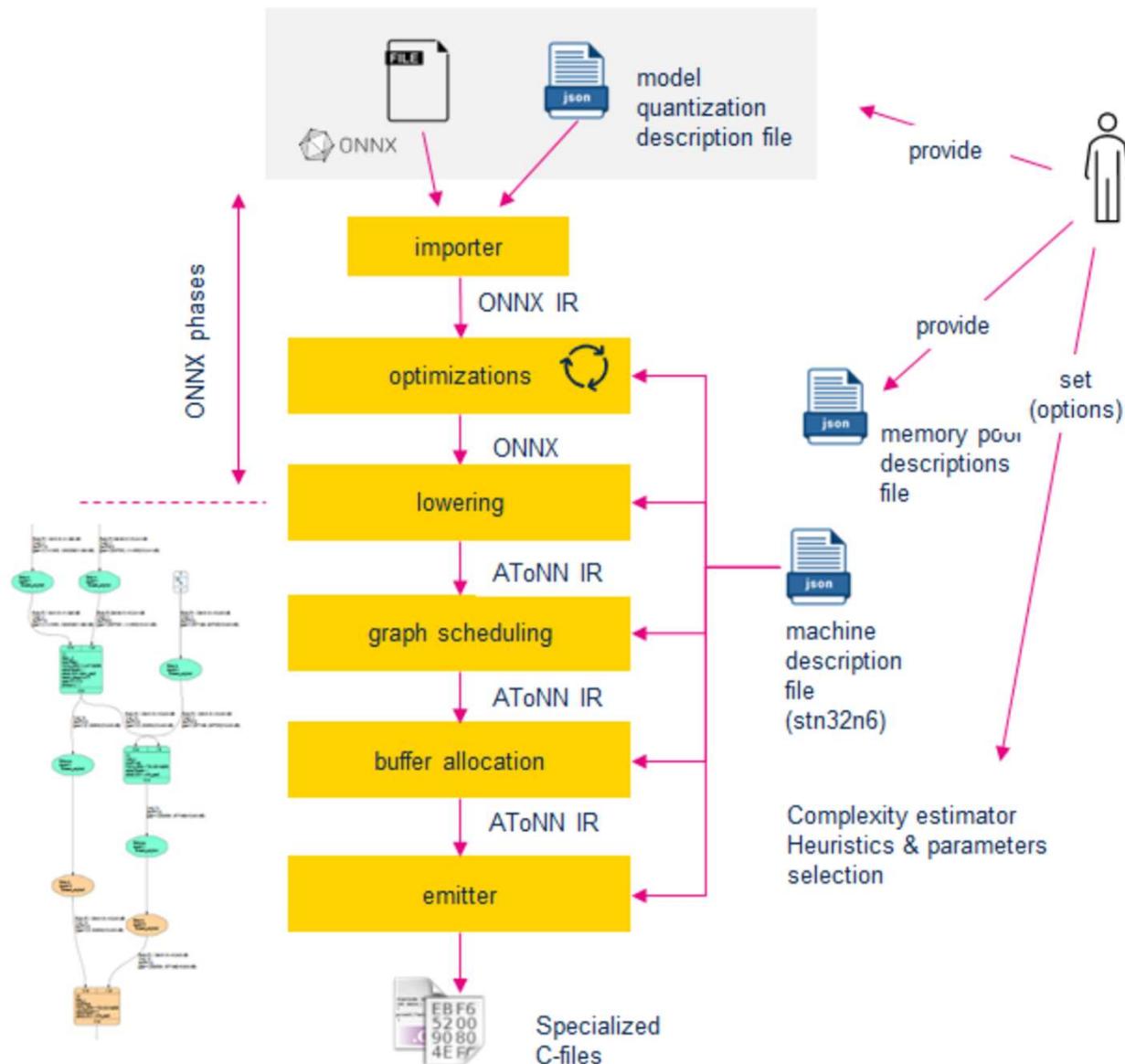
Ahead-Of-Time (AOT) flow



Intermediate files



NPU Compiler



\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_mdescs/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_mdescs/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_mdescs/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_mdescs/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-O1": {  
        "memory_pool": "./my_mpools/stm32n6.mpool",  
        "memory_desc": "./my_mdescs/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-O2": {  
        "memory_pool": "./my_mpools/stm32n6.mpool",  
        "memory_desc": "./my_mdescs/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-O3": {  
        "memory_pool": "./my_mpools/stm32n6.mpool",  
        "memory_desc": "./my_mdescs/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_mdescs/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", "magnitude": "BYTES" },
        "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": "0x70000000", "magnitude": "BYTES" },
        "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**.



Hands-On AI: Recognizing Handwritten Digits with MNIST



Running CubeMX

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 'Commercial Part Number' field in the central selector window. The selector window has tabs for 'MCU/MPU Selector', 'Board Selector', 'Example Selector', and 'Cross Selector'. The 'MCU/MPU Selector' tab is active. It displays 'MCU/MPU Filters' with a dropdown set to 'STM32N657X0H3Q' (circled with '2'). Below are sections for 'PRODUCT INFO' (Segment, Series, Line, Marketing Status, Price, Package, Core, Coprocessor) and 'MEMORY' (Flash = 0 (kBytes)). 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 labeled '3' highlights the first column of the table. A red arrow labeled 'Double Click' points to the table's header.

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

②

③

Double Click

MX New Project from a MCU/MPU

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

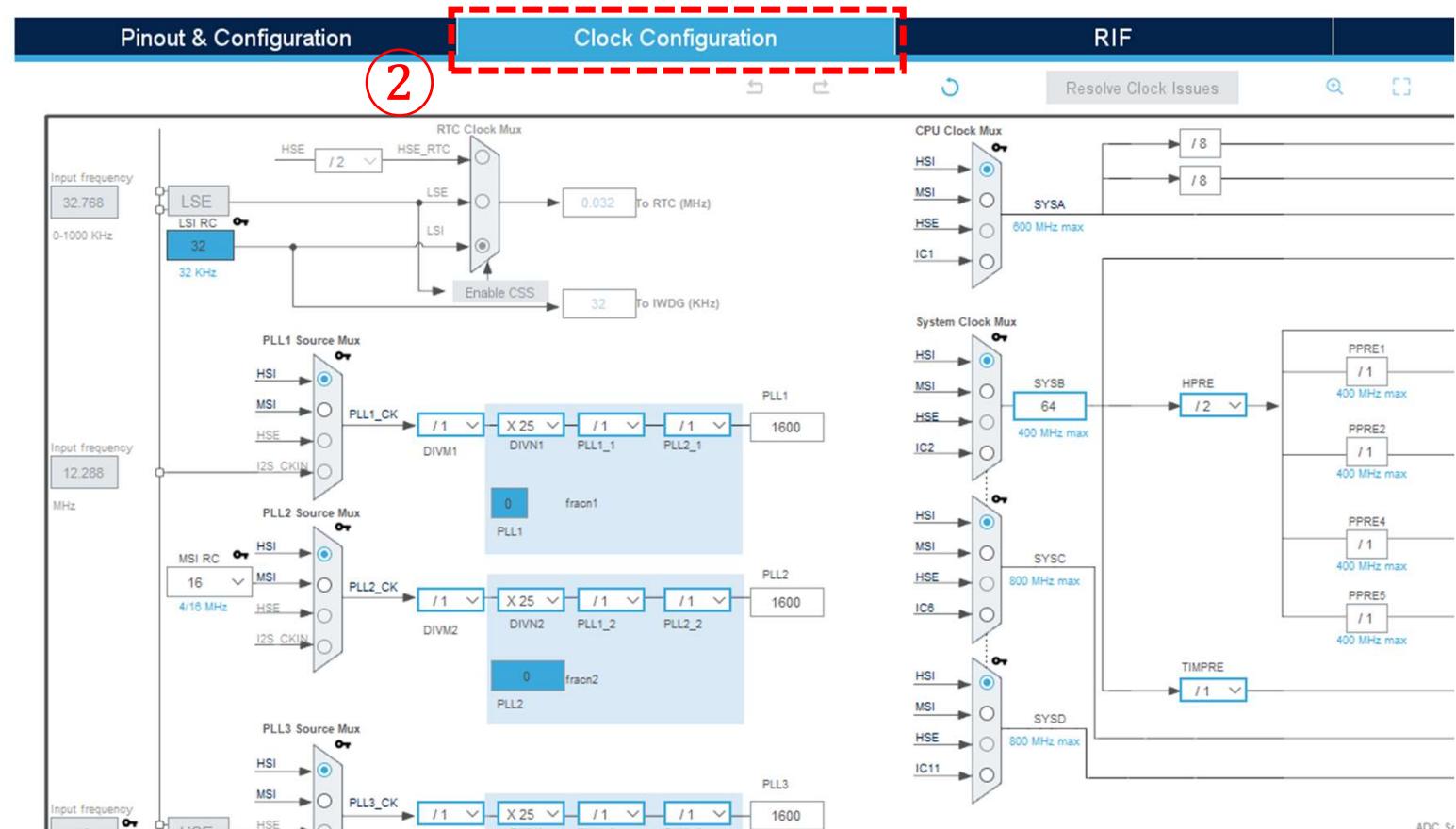


https://github.com/BlaineMoon30/STM32_Expert_Training_AI_STM32/tree/master/Hands_On/_2_Hands_On_NPU_Mnist

Clock Configuration

TrustZone feature available X

? Do you want to create a new project :
① Secure and Non Secure domains
② 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 the configuration interface for the CORTEX_M55_S. The left pane displays the 'Pinout & Configuration' tab, listing various system components like System Core, GPDMA1, GPIO, HPDMA1, ICACHE, IWDG, NVIC1_S_Application, NVIC_FSBL, RAMCFG, RCC, SYS_S, and WWDG. The 'CORTEX_M55_S' component is highlighted with a red box and circled with a red number '1'. The right pane displays the 'Clock Configuration' tab, which includes sections for Software Packs and Pinout. The 'Software Packs' section shows 'CORTEX_M55_S Mode and Configuration' with 'Mode' set to 'Runtime contexts: First Stage Boot Loader, Application, External Memory Loader'. The 'Pinout' section shows 'CORTEX_M55_S Mode and Configuration' with 'Mode' set to 'Runtime contexts: First Stage Boot Loader, Application, External Memory Loader'. The 'Configuration' section contains 'Reset Configuration' and tabs for 'Parameter Settings' (selected) and 'User Constants'. Under 'Parameter Settings', there is a search bar and a section for 'Cortex Interface Settings' where 'CPU ICache' and 'CPU DCache' are both set to 'Enabled'. A red box highlights the 'Cortex Interface Settings' section and a red number '2' is circled around it.

"CPU ICACHE and CPU DCACHE are enabled."



RAMCFG Configuration

The screenshot shows two software interfaces side-by-side for configuring a microcontroller's peripherals.

Pinout & Configuration: This interface allows selecting components by category or letter. The "System Core" section is expanded, showing various peripherals like CORTEX_M55_FSB, GPIO, and RAMCFG. The RAMCFG row is highlighted with a red dashed box, indicating it is selected for configuration.

	FSBL	Application	ExtMemLoa...
CORTEX_M55_FSB	<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_FSB	<input checked="" type="checkbox"/>		
RAMCFG	<input type="checkbox"/>	<input checked="" 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"/>	<input type="checkbox"/>

Clock Configuration: This interface shows "Software Packs" for "RAMCFG Mode and Configuration". The "Mode" tab is selected, displaying configuration options for RAMCFG AXISRAM1 through AXISRAM6. The same RAMCFG row from the Pinout & Configuration interface is also highlighted with a red dashed box here.

RAMCFG AXISRAM1
<input type="checkbox"/>
<input type="checkbox"/>
<input checked="" type="checkbox"/> RAMCFG AXISRAM3
<input checked="" type="checkbox"/> RAMCFG AXISRAM4
<input checked="" type="checkbox"/> RAMCFG AXISRAM5
<input checked="" type="checkbox"/> RAMCFG AXISRAM6
<input type="checkbox"/> RAMCFG AHBSRAM1
<input type="checkbox"/> RAMCFG AHBSRAM2

Configuration: This section contains a "Reset Configuration" button and tabs for "Parameter Settings" and "User Constants". It lists parameters for each AXISRAM module, all currently set to "Disable".

Configure the below parameters :	
<input type="checkbox"/> Search (Ctrl+F)	
<input checked="" type="checkbox"/> Parameter Settings	
<input checked="" type="checkbox"/> User Constants	
AXISRAM3	
Software Erase	Disable
AXISRAM4	
Software Erase	Disable
AXISRAM5	
Software Erase	Disable
AXISRAM6	
Software Erase	Disable



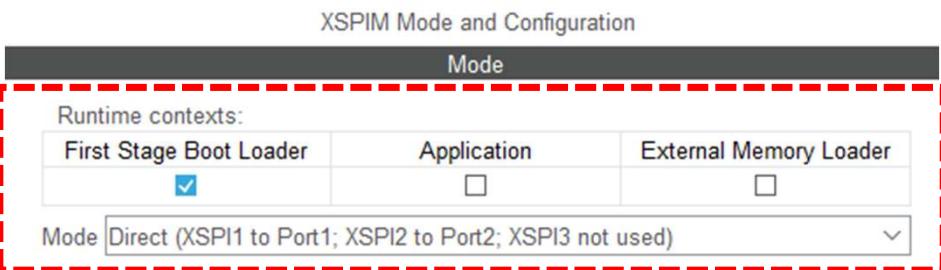
XSPIM Configuration

A screenshot of the CubeMX pinout matrix configuration. The columns are labeled 'Pin' and 'Port'. The rows include 'USB1_OTG_HS', 'USB2_OTG_HS', 'XSPI1', 'XSPI2', 'XSPI3', and 'XSPIM'. The 'XSPIM' row has a red dashed box around it, and the first column contains a checked checkbox under the 'XSPIM' heading.

Pin	Port	USB1_OTG_HS	USB2_OTG_HS	XSPI1	XSPI2	XSPI3	XSPIM
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

1

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



2



XSPI2 Configuration

The screenshot shows the XSPI2 configuration interface in CubeMX. It is divided into two main sections:

- Step 1: XSPI2 Mode Selection**
A grid of checkboxes for various SPI modes (XSPI1, XSPI2, XSPIM) across different pins (MOSI, MISO, SCK, CS). The XSPI2 row is highlighted with a red dashed border, and the XSPI2 checkbox is checked.
- Step 2: XSPI2 Mode and Configuration**
A detailed configuration panel for XSPI2:
 - Runtime contexts:** Application is selected (checked).
 - Mode:** Octo SPI
 - Port:** Port2 Octo
 - HyperBus(TM) 1.8V Inverted Clock:** Disable
 - Chip Select Override:** NCS1 -- Port2 --A red dashed border surrounds the entire configuration panel.

The screenshot shows the XSPI2 configuration parameters in the CubeMX configuration tab:

Configuration

Reset Configuration

Checkboxes for enabling settings:
NVIC Settings, DMA Settings, GPIO Settings, Parameter Settings, User Constants

Configure the below parameters :

Search (Ctrl+F)
i		
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"/>
MCA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2

BSEC Mode and Configuration

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



GPIOB Configuration

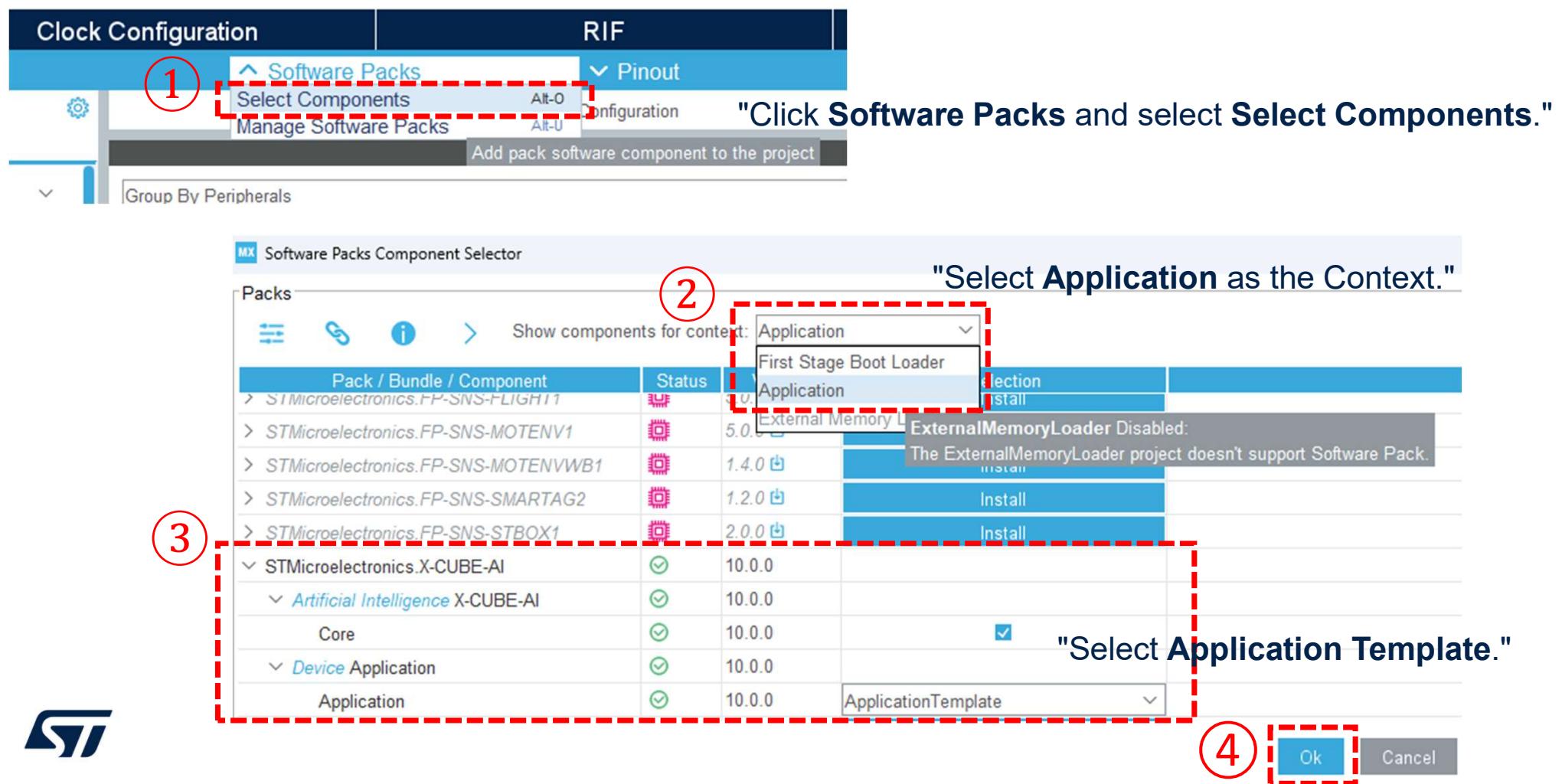
The image shows three panels from the STM32CubeMX software:

- Pinout View (Left):** Shows the physical pin layout. Pin PB12 is highlighted with a red circle labeled 1. A red dashed box labeled 2 highlights the row for GPIO.
- System Core Configuration (Middle):** Shows the configuration for various peripherals. The GPIO row is highlighted with a red dashed box labeled 2. The "Application" column for PB12 is checked, indicating it is configured as an output. Other peripherals listed include CORTEX_M55_FSBL, CORTEX_M55_S, CDDMA4, ICACHE, IWDG, NVIC1_S_Application, NVIC_FSBL, RAMCFG, RCC, SYS_S, and WWDG.
- Configuration Panel (Right):** Provides detailed configuration for the selected pin. It includes sections for "Pin Context Assignment" (Application, High), "GPIO output level" (High), "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."

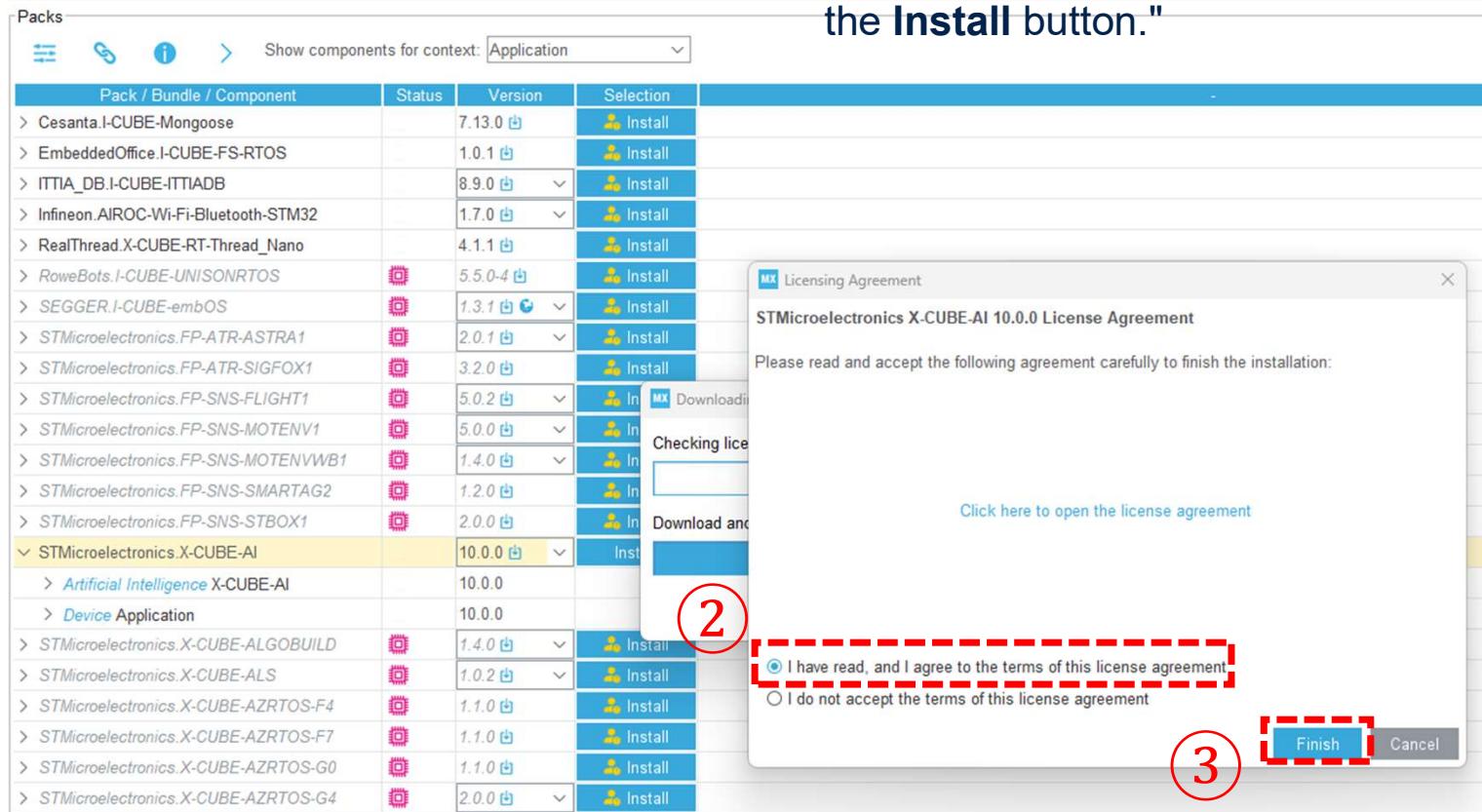
ST EdgeAI Core Configuration - 1



ST EdgeAI Core Configuration -

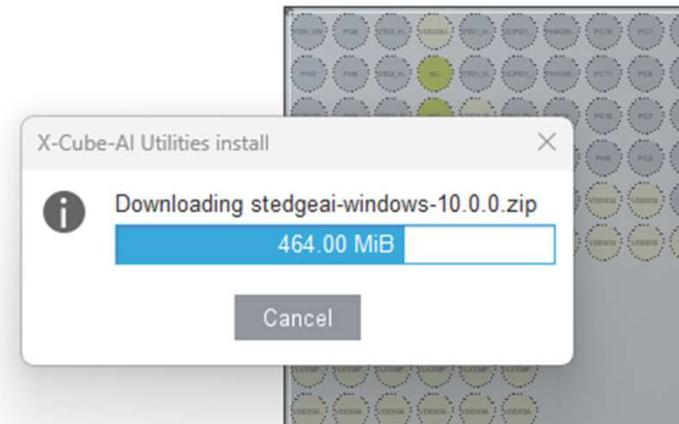
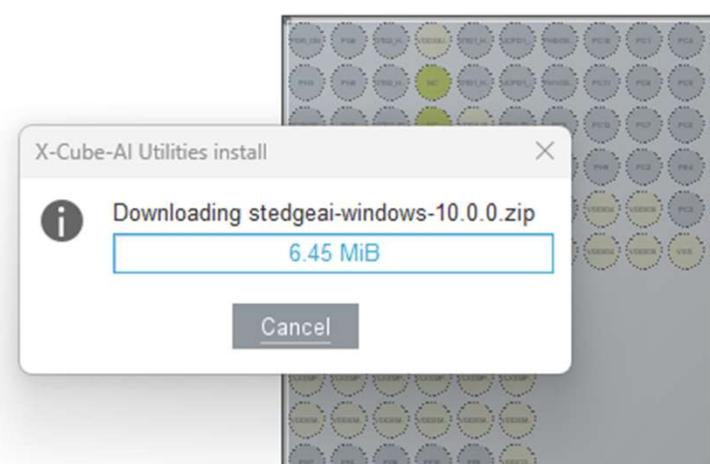


"If X-CUBE-AI is not installed, you can install it by clicking the **Install** button."

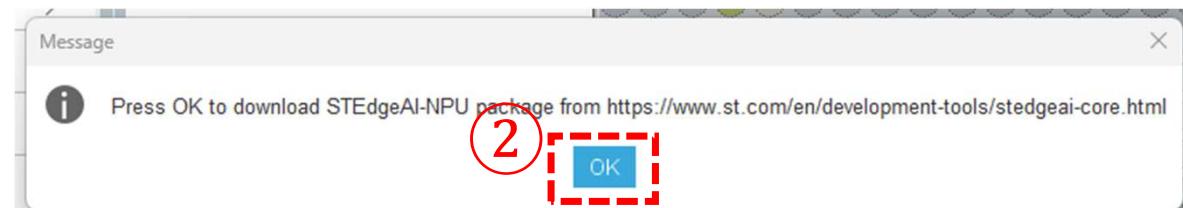
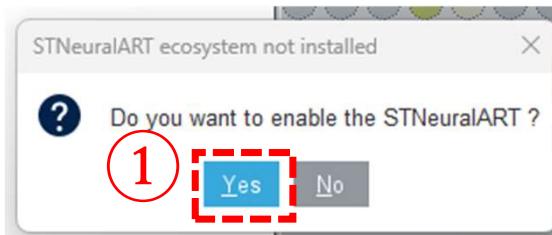


ST EdgeAI Core Configuration -

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



ST EdgeAI Core Configuration -

A screenshot of the ST website at <https://www.st.com/en/development-tools/stedgeai-core.html>. The page lists several software packages under the heading "ST Edge Ai Core SW Package". The "STEdgeAI-NPU" package is highlighted with a red circle and a red dashed box around its "Get latest" button.

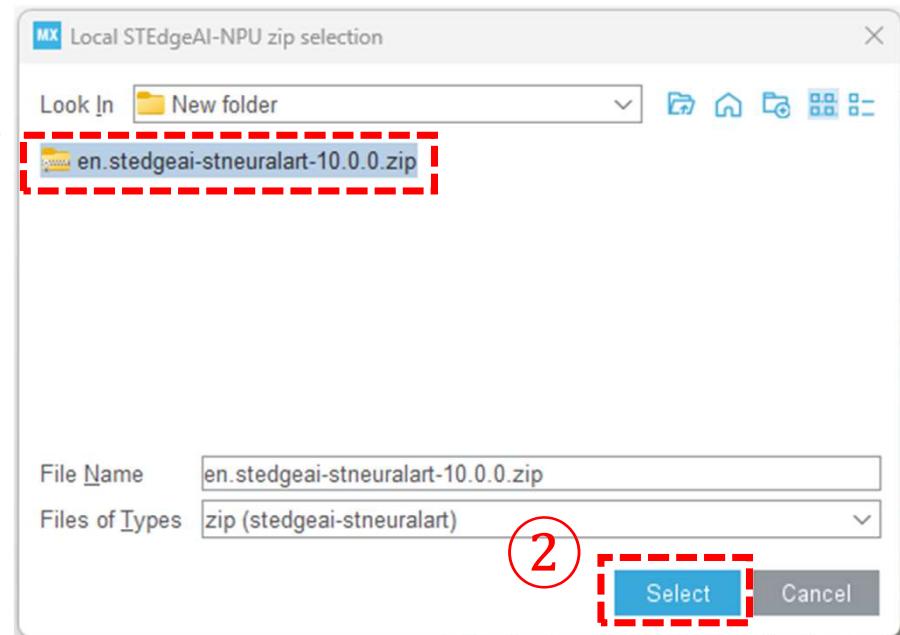
Package	Description	Part Number	Action
STEAICore-Linux	ST Edge Ai Core SW Package for Linux	3D991	Get latest
STEAICore-Mac	ST Edge Ai Core SW Package for Mac	3D991	Get latest
STEAICore-MacArm	ST Edge Ai Core SW Package for Mac ARM	3D991	Get latest
STEAICore-Win	ST Edge Ai Core SW Package for Windows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

"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."

ST EdgeAI Core Configuration -

	FSBL	Application	ExtMemL...
AIROC-Wi-Fi-Bluetooth-ST...			<input checked="" type="checkbox"/>
EXTMEM_LOADER	<input type="checkbox"/>		<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 type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

①

Software Packs Pinout

STMicroelectronics.X-CUBE-AI.10.0.0_Appl Mode and Configuration

Mode

Runtime contexts:

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

②

Artificial Intelligence X-CUBE-AI
Device Application

Non optimal boards parameters

Clock and Peripherals parameters not set for best performance, automatically fix ?

Yes No

"If the popup window appears, click YES."



ST EdgeAI Core Configuration -

The image shows two screenshots of the ST EdgeAI Core Configuration interface. Both screenshots have a header with 'Configuration' and buttons for 'Reset Configuration', 'Developer Cloud', 'Add network', and 'Delete network'.

The top screenshot shows a 'Model manager' section with a table:

Name	Used Ram	Used Flash	Complexity
Library	.00 B	.00 B	-
Total (0)	.00 B	.00 B	-

The bottom screenshot shows a 'Model inputs' section with a table:

network	Runtime
Keras	STM32Cube.AI MCU runtime

Below the table are 'Browse...' buttons. At the bottom of the configuration screen are dropdowns for 'Compression' (None), 'Optimization' (Balanced), and 'Validation inputs' (Random numbers). There is also a 'Show graph' button and a gear icon.

A red dashed box highlights the 'TFLite' entry in the top configuration's 'Model inputs' table. A red dashed box highlights the 'Keras' entry in the bottom configuration's 'Model inputs' table. A red arrow points from the 'TFLite' entry in the top configuration to the 'STM32Cube.AI Neural-ART™ runtime' entry in the bottom configuration's 'Model inputs' table.

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



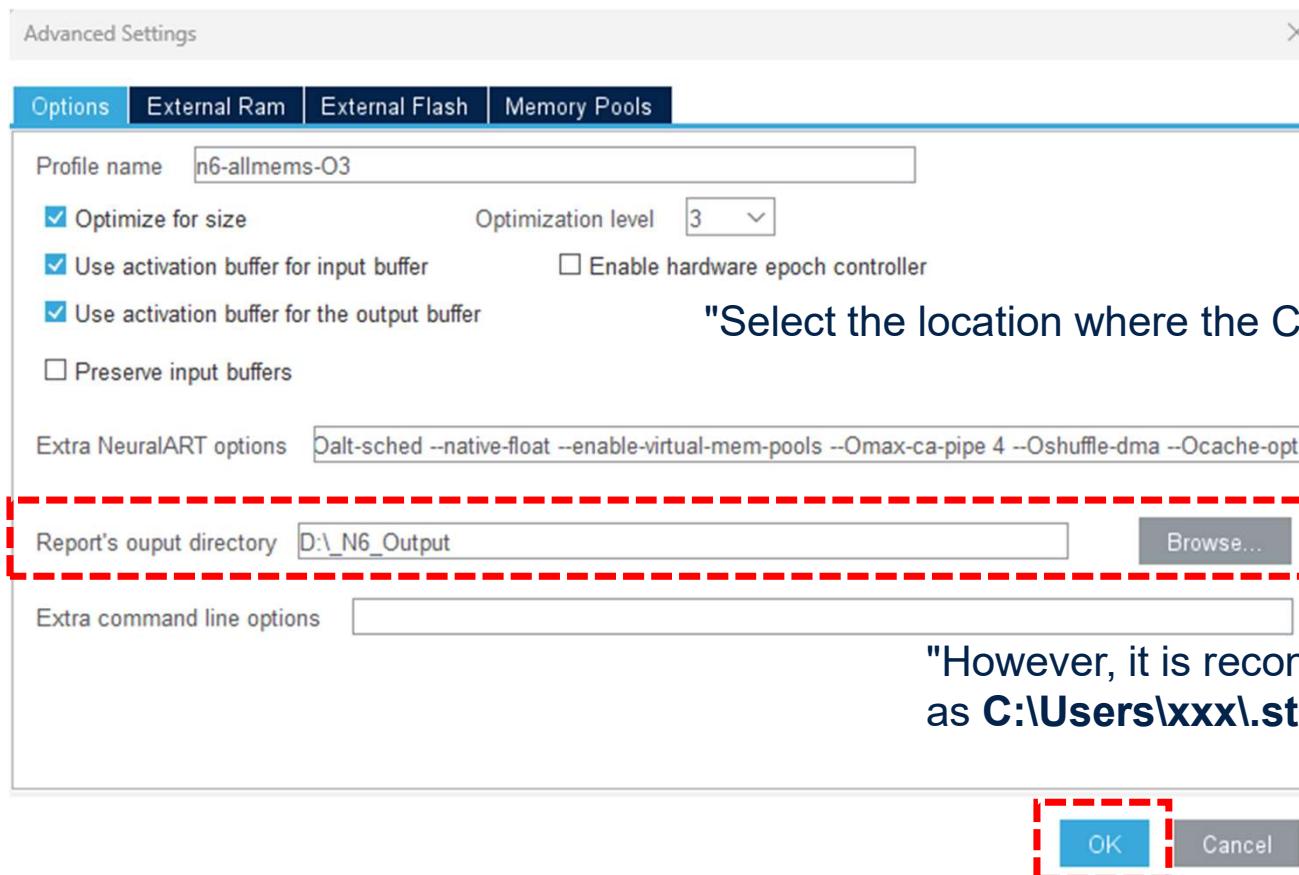
ST EdgeAI Core Configuration -

The screenshot shows the ST EdgeAI Core Configuration software interface. At the top, there are tabs for 'Reset Configuration' (disabled), 'Developer Cloud' (selected), 'Add network', and a search bar. Below the tabs, there are sections for 'Main' (selected), 'network' (disabled), and a '+' button. The 'Model inputs' section contains a 'network' dropdown set to 'TFLite' and a 'Profile' dropdown set to 'n6-allmems-O3'. The 'Validation' section includes dropdowns for 'Validation inputs' (set to 'Random numbers') and 'Validation outputs' (set to 'None'). On the right, there are buttons for 'Show graph', 'Analyze' (highlighted with a red box), 'Validate on desktop', and 'Validate on target'. A large central area displays a 'Select' dialog box. The dialog has a 'Look In' dropdown set to 'Mnist_Model' and a file list containing 'network_output', 'Script', and 'cnn_mnist_quantized.tflite'. The 'cnn_mnist_quantized.tflite' file is highlighted with a red dashed box. Below the list are fields for 'File Name' (set to 'cnn_mnist_quantized.tflite') and 'Files of Types' (set to '.tflite'), both with red dashed boxes around them. At the bottom of the dialog are 'Select' and '취소' (Cancel) buttons, with the 'Select' button also highlighted with a red dashed box.



Load the **cnn_mnist_quantized.tflite** model from the
_2_Hands_On_NPU_Mnist\Nist_Model folder.

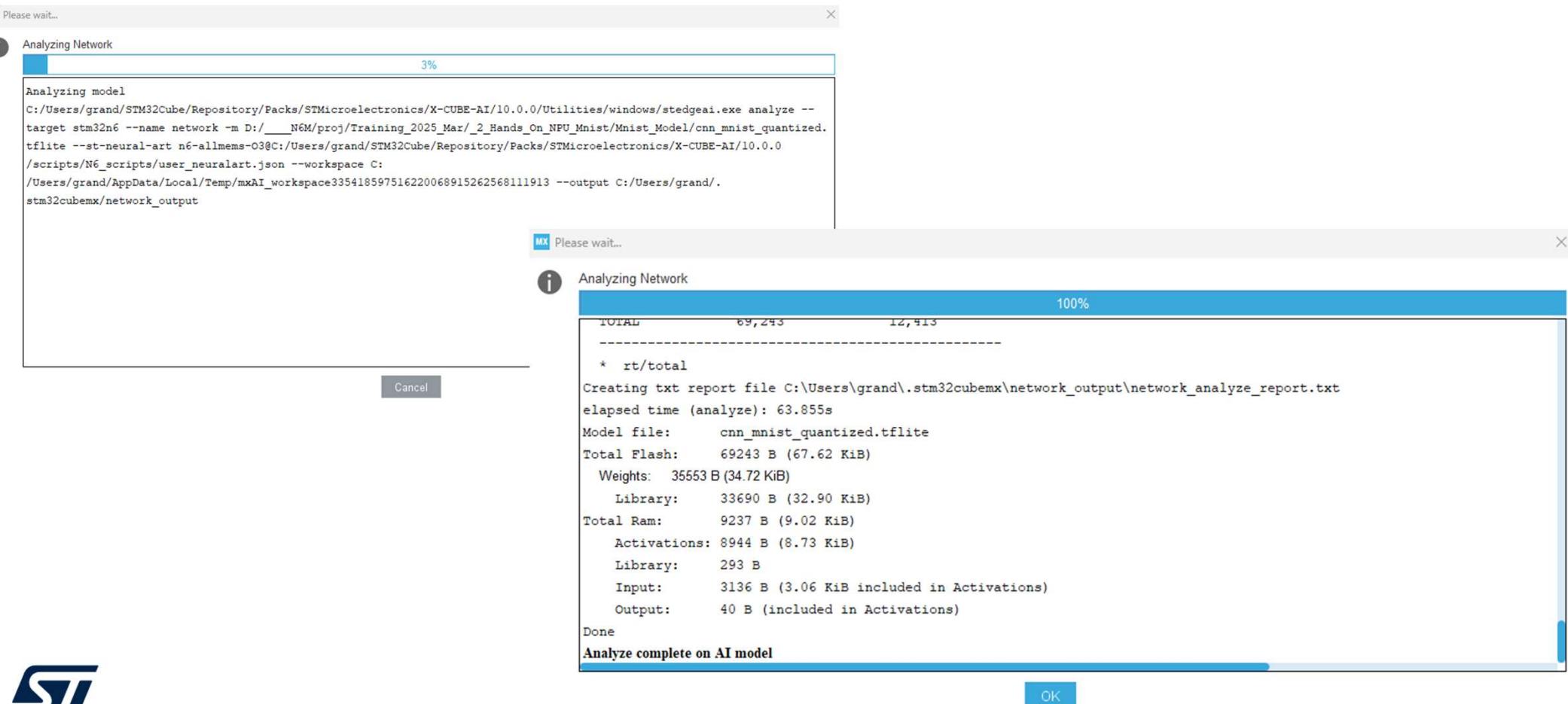
ST EdgeAI Core Configuration -



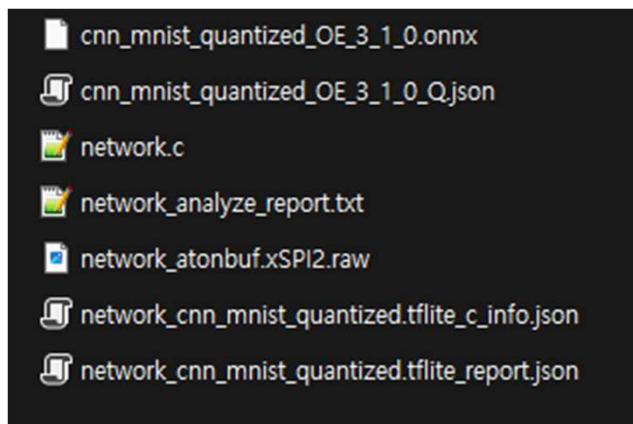
"Click **OK** to automatically start the network analysis."



ST EdgeAI Core Configuration -



ST EdgeAI Core Configuration -

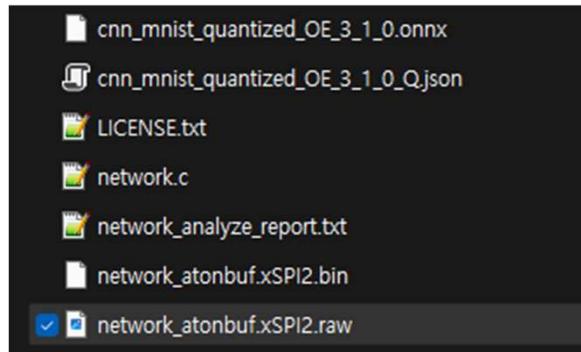


"The **network_output** folder will contain the generated report and code files."

```
network_analyze_report.txt
O| 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
-----
61
62 Used memory ranges
63 -----
64   npuRAM5 [0x342E0000 - 0x34350000]: 0x342E0000-0x342E22F0
65   octoFlash [0x70000000 - 0x73FFFFFF8]: 0x70000000-0x70008AF0
66 -----
68 Epochs details
69 -----
70 Total number of epochs: 9 of which 3 implemented in software
71 -----
72 epoch ID HW/SW/EC Operation (SW only)
73 epoch 1 HW
74 epoch 2 -SW- ( QuantizeLinear )
75 epoch 3 HW
76 epoch 4 HW
77 epoch 5 HW
78 epoch 6 HW
79 epoch 7 HW
80 epoch 8 -SW- ( Softmax )
81 epoch 9 -SW- ( DequantizeLinear )
82 -----
83 Requested memory size by section - "stm32n6npu" target
84 -----
```



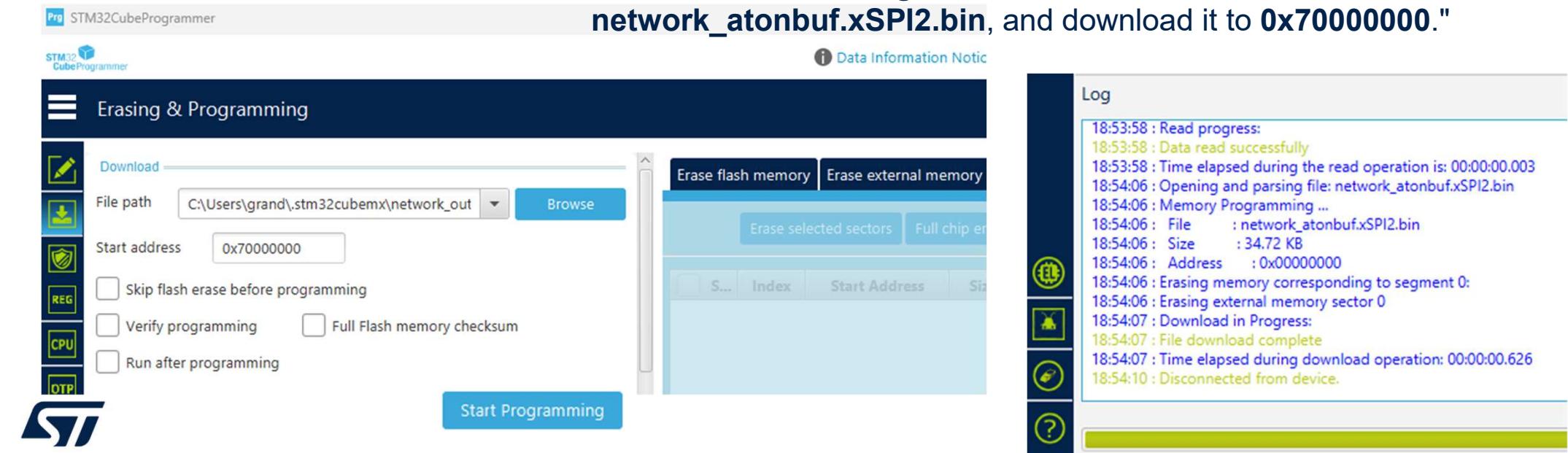
ST EdgeAI Core Configuration -



"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 **0x70000000**."

"Run **STM32CubeProgrammer**, load **network_atonbuf.xSPI2.bin**, and download it to **0x70000000**."



The screenshot shows the STM32CubeProgrammer interface with the following details:

- Download:** File path is set to `C:\Users\grand\stm32cubemx\network_out`.
- Start address:** Set to `0x70000000`.
- Programming Options:** Includes checkboxes for "Skip flash erase before programming", "Verify programming", "Full Flash memory checksum", and "Run after programming".
- Start Programming:** A large blue button.
- Central Panel:** Shows memory erase operations with buttons for "Erase flash memory" and "Erase external memory".
- Log Window:** Displays the following log entries:

```
18:53:58 : Read progress:  
18:53:58 : Data read successfully  
18:53:58 : Time elapsed during the read operation is: 00:00:00.003  
18:54:06 : Opening and parsing file: network_atonbuf.xSPI2.bin  
18:54:06 : Memory Programming ...  
18:54:06 : File : network_atonbuf.xSPI2.bin  
18:54:06 : Size : 34.72 KB  
18:54:06 : Address : 0x00000000  
18:54:06 : Erasing memory corresponding to segment 0:  
18:54:06 : Erasing external memory sector 0  
18:54:07 : Download in Progress:  
18:54:07 : File download complete  
18:54:07 : Time elapsed during download operation: 00:00:00.626  
18:54:10 : Disconnected from device.
```

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 7 nl_7 (SOFTMAX) [b:1,c:10]  
30 -----  
31 8 conversion_8 (DEQUANTIZE) [b:1,c:10]  
32 -----  
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)		<p>Global lock : OFF</p> <table><thead><tr><th>RIMU IP</th><th>RIMU ID</th><th>MASTER CID</th><th>SECURE</th></tr></thead><tbody><tr><td>DCMIPP</td><td>9</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>DMA2D</td><td>8</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>ETH1</td><td>6</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>ETR</td><td>0</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>GPU</td><td>7</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>LTDC_L1</td><td>10</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>LTDC_L2</td><td>11</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>NPU</td><td>1</td><td>1</td><td><input checked="" type="checkbox"/></td></tr><tr><td>OTG1</td><td>4</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>OTG2</td><td>5</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>SDMMC1</td><td>2</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>SDMMC2</td><td>3</td><td>0</td><td><input type="checkbox"/></td></tr><tr><td>VENC</td><td>12</td><td>0</td><td><input type="checkbox"/></td></tr></tbody></table>	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"/>	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"/>	
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"/>																																																								
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"/>																																																								
Domains (RIMU)																																																											

Users can configure the RIF as needed.



Project Configuration

Pinout & Configuration		Clock Configuration	RIF	Project Manager
Project	Project Settings			
	Project Name	<input type="text" value="stm32n6_nucleo_npu_nn_mnist"/>		
	Project Location	<input type="text" value="D:\N6M\proj\Training_2025_Mar\2_Hands_On_NPU_Mnist"/> <input type="button" value="Browse"/>		
Code Generator	Project Structure	<input checked="" type="checkbox"/> FSBL <input checked="" type="checkbox"/> Appli <input type="checkbox"/> ExtMemLoader		
	Application Structure	<input type="text" value="Advanced"/> <input type="checkbox"/> Do not generate the main()		
	Toolchain Folder Location	<input type="text" value="D:\N6M\proj\Training_2025_Mar\2_Hands_On_NPU_Mnist\stm32n6_nucleo_npu_nn_mnist\"/> <input type="checkbox"/>		
Toolchain / IDE	<input type="text" value="STM32CubelDE"/> <input checked="" type="checkbox"/> Generate Under Root			



Project Configuration

Advanced Settings

Generate Code	Rank	Function Name	Run
<input checked="" type="checkbox"/>	1	SystemClock_Config	Run

"In this example, uncheck the checkbox to **disable automatic code generation.**"

Generated Function Calls Application

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

Generated Function Calls ExtMemLoader

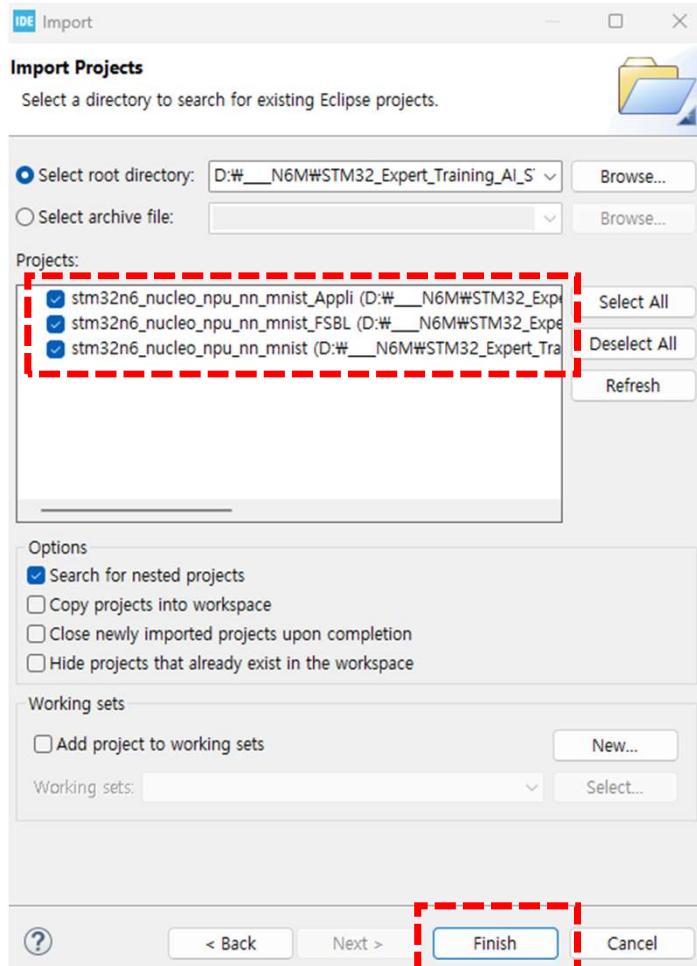
Generate Code	Rank	Function Name	Run



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

GENERATE CODE

STM32CubeIDE Import Project



The Project Explorer view shows the imported projects:

- stm32n6_nucleo_fsbl_lrnn_led_toggle
- stm32n6_nucleo_npu_nn_mnist
- stm32n6_nucleo_npu_nn_mnist_Appli (in Appli)
- stm32n6_nucleo_npu_nn_mnist_FSBBL (in FSBBL)
- stm32n6_nucleo_nn_mnist
- stm32n6_nucleo_nn_mnist_FSBBL

The main.c file content is as follows:

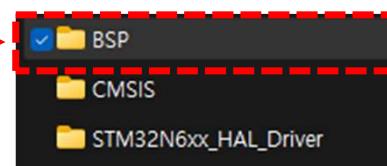
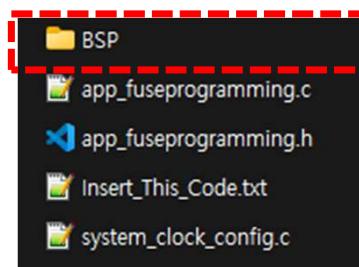
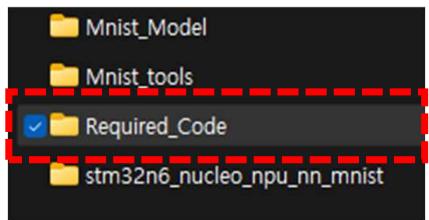
```
1 /* USER CODE BEGIN Header */
2 /**
3  * @file           : main.c
4  * @brief          : Main program body
5  * @attention
6  * 
7  * Copyright (c) 2025 STMicroelectronics
8  * All rights reserved.
9  * 
10 * This software is licensed under terms
11 * in the root directory of this package.
12 * If no LICENSE file comes with this file,
13 * please read "LICENSE.md".
14 * 
15 */
16 
17 /*
18 * USER CODE END Header
19 */
20 /* Includes -----
21 #include "main.h"
22 #include "app_x-cube-ai.h"
23 */
24 /* Private includes -----
25 /* USER CODE BEGIN Includes */
26 */
27 /* USER CODE END Includes */
28 */
29 /* Private typedef -----
30 /* USER CODE BEGIN PTD */
31 */
32 /* USER CODE END PTD */
33 */
34 /* Private define -----
35 /* USER CODE BEGIN PD */
36 */
37 /* USER CODE END PD */
38 */
39 /* Private macro -----
```

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

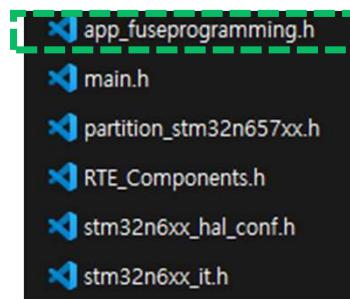
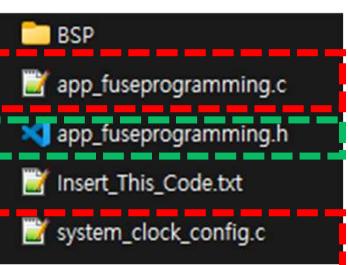
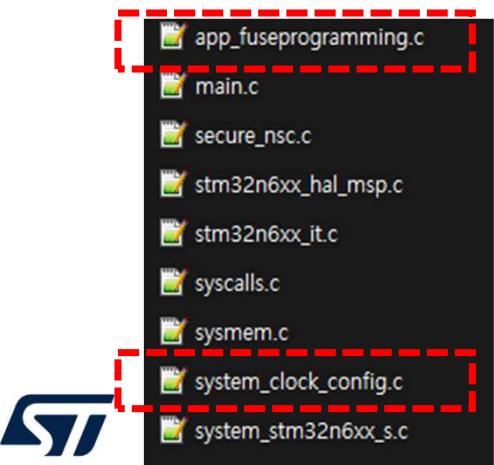


STM32CubeIDE

"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 **AppliCore** folder."

"Copy the **app_fuseprogramming.h** file into the **AppliInc** folder."

STM32CubeIDE

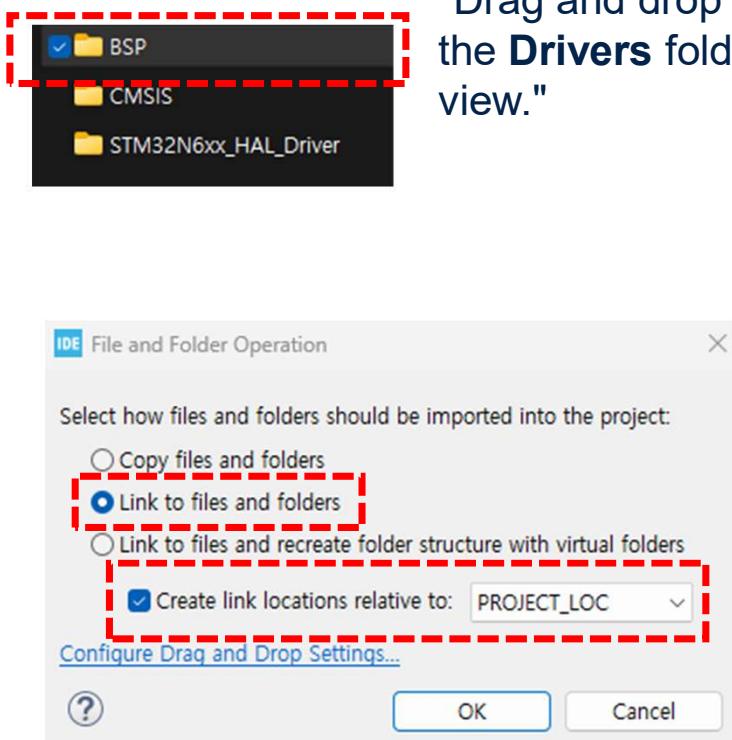
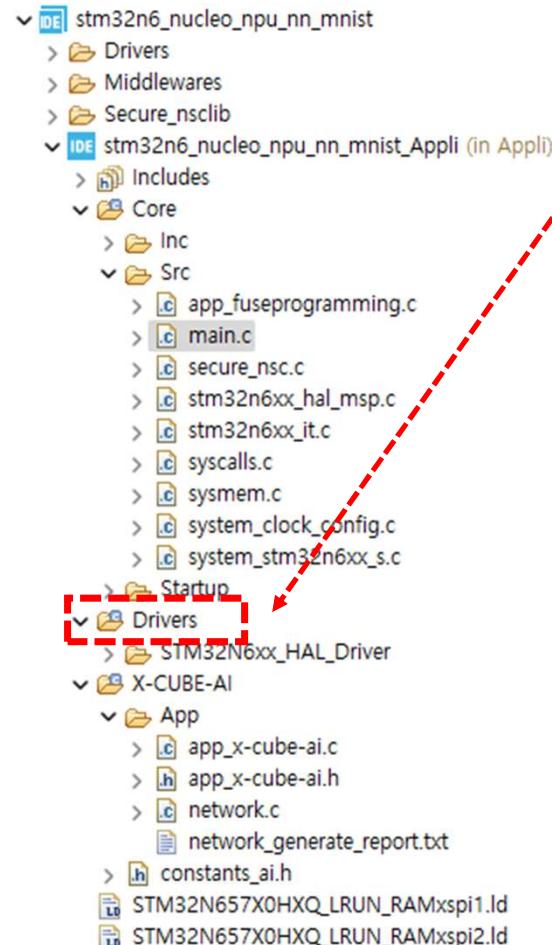
"In this example, **MX_X_CUBE_AI_Process()** is not used."

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

- stm32n6_nucleo_npu_nn_mnist (selected)
- Drivers
- Middlewares
- Secure_nsclib
- stm32n6_nucleo_npu_nn_mnist_App (in Appli)
 - Includes
 - Core
 - Inc
 - Src
 - app_fuseprogramming.c
 - main.c
 - secure_nsclib.c
 - stm32n6xx_hal_msp.c
 - stm32n6xx_it.c
 - syscalls.c
 - sysmem.c
 - system_clock_config.c
 - system_stm32n6xx_s.c
 - Startup
 - Drivers
 - STM32N6xx_HAL_Driver
 - X-CUBE-AI
 - App

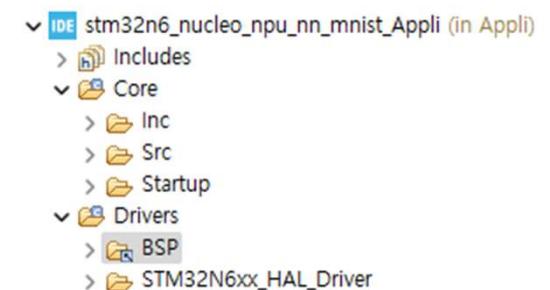


STM32CubeIDE



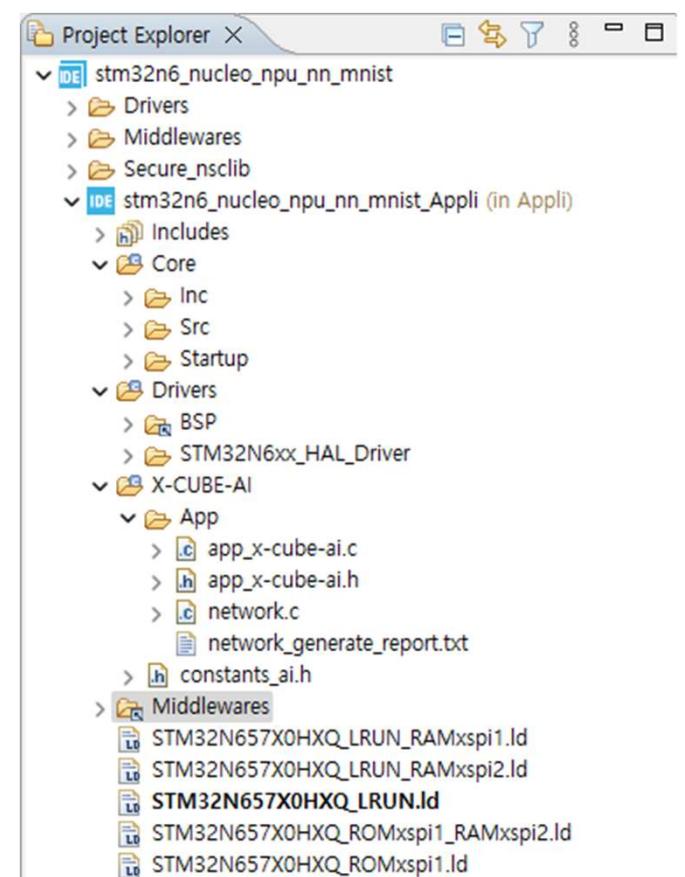
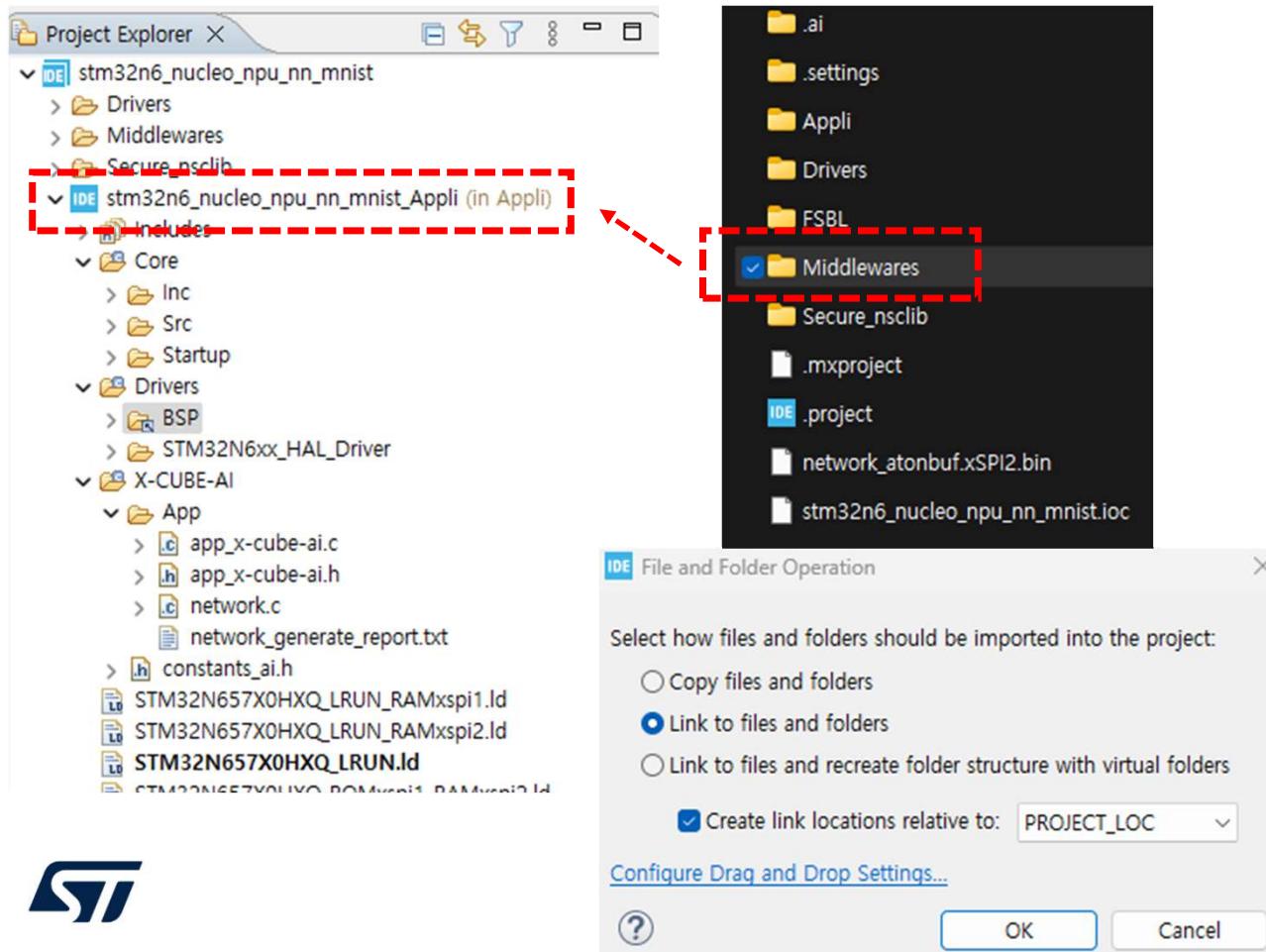
"Drag and drop the **BSP** folder from the **Drivers** folder into the **Drivers** folder in STM32CubeIDE's **Project Explorer** view."

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

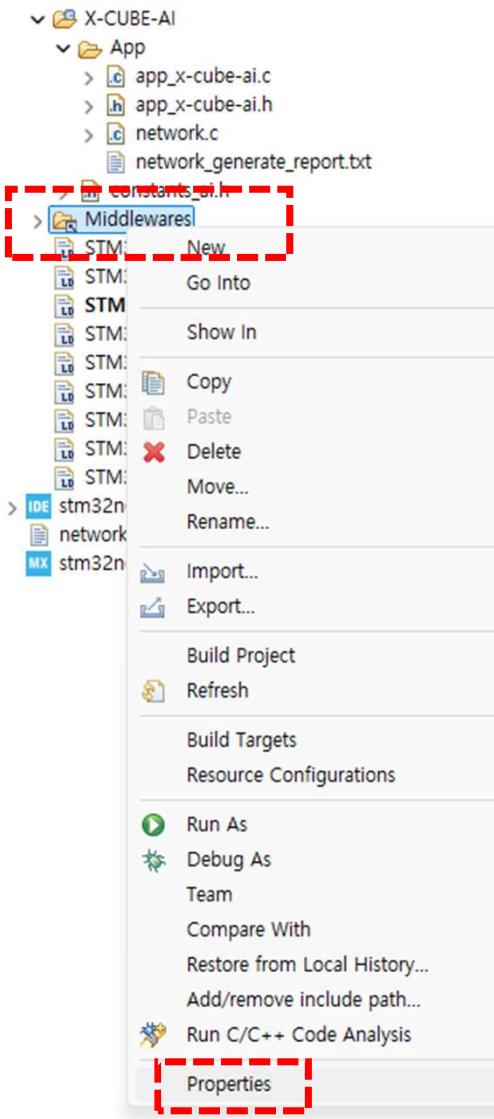


STM32CubeIDE

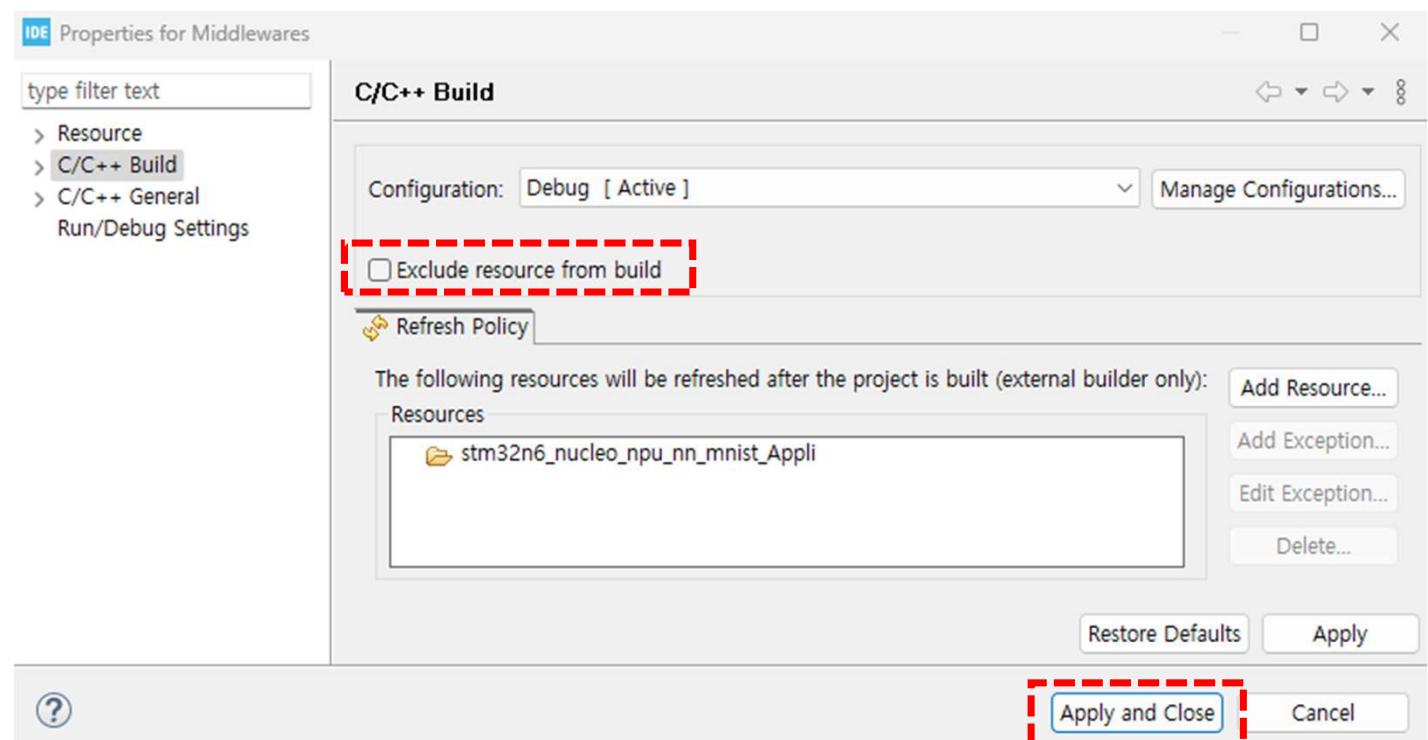
"Drag and drop the **Middlewares** folder to create a link."



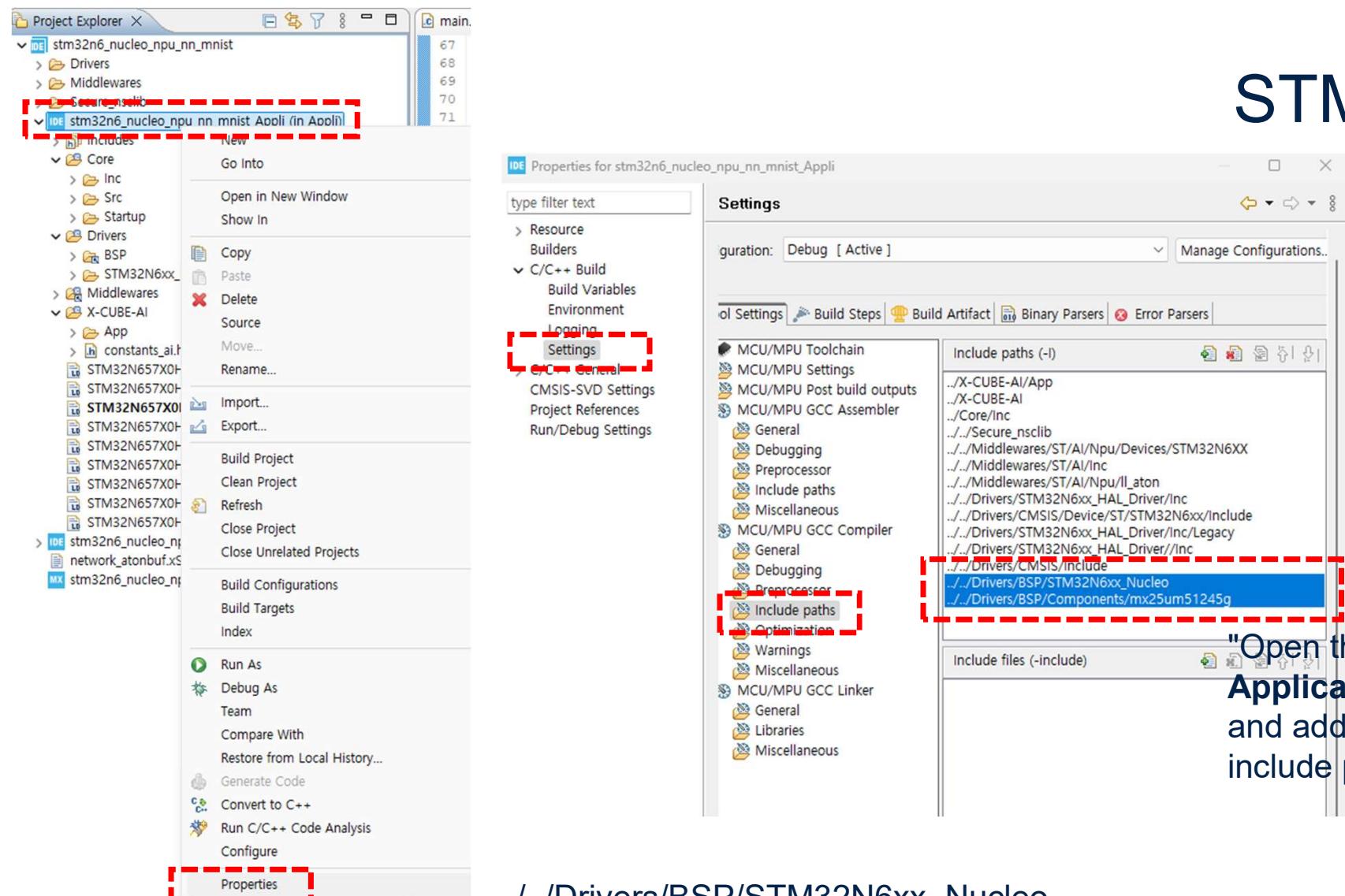
STM32CubeIDE



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



STM32CubeIDE



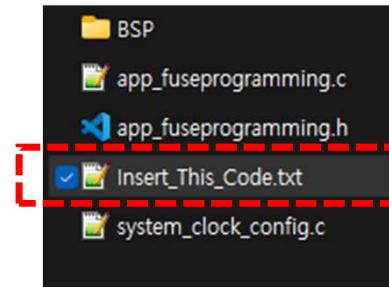
"Open the Properties of Application, go to Include paths, and add the BSP folder to the include paths."

```
.../Drivers/BSP/STM32N6xx_Nucleo  
.../Drivers/BSP/Components/mx25um51245g
```

STM32CubeIDE

The screenshot shows the STM32CubeIDE interface. On the left is the Project Explorer, displaying a project structure for 'stm32n6_nucleo_npu_nn_mnist'. The 'Core' folder contains several source files: app_fuseprogramming.c, main.c, secure_nsclib.c, stm32n6xx_hal_msp.c, stm32n6xx_it.c, syscalls.c, sysmem.c, system_clock_config.c, and system_stm32n6xx_s.c. The 'main.c' file is open in the center editor window. The code includes standard headers like stdio.h, stdlib.h, and string.h, along with application-specific headers like app_fuseprogramming.h, ll_aton_runtime.h, main.h, and app_x-cube-ai.h. A block of code starting with '#include "app_x-cube-ai.h"' is highlighted in blue.

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

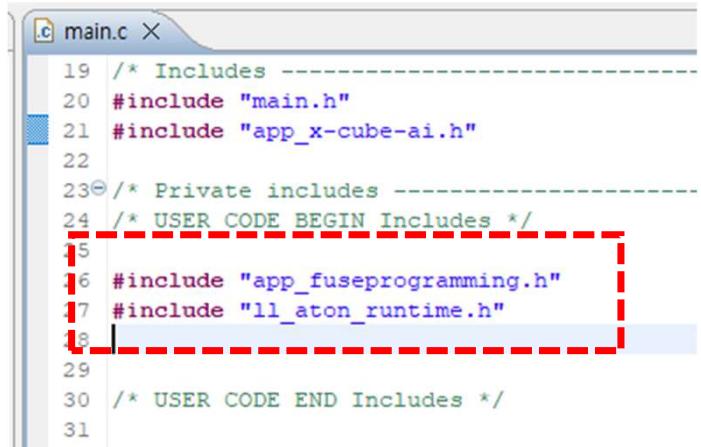


The code editor window shows the contents of 'Insert_This_Code.txt'. The code includes comments for USER CODE BEGIN and END sections, and defines for LL_ATON_DECLARE_NAMED_NN_INSTANCE_AND_INTERFACE. It also includes declarations for time_in and time_out functions, and variables for duration_us, duration_dwt, t_init, t_out, clock_Hz, and cpuclk.

```
1  /* USER CODE BEGIN Includes */
2
3  #include "app_fuseprogramming.h"
4  #include "ll_aton_runtime.h"
5
6  /* USER CODE END Includes */
7
8  /* USER CODE BEGIN PD */
9
10 LL_ATON_DECLARE_NAMED_NN_INSTANCE_AND_INTERFACE(Default) // Defines
11
12 /* USER CODE END PD */
13
14 /* USER CODE BEGIN O */
15
16  *****
17  * TIME MEASUREMENT
18  *****
19  void time_in(void);
20  uint32_t time_out(void);
21
22  uint32_t duration_us;
23  uint32_t duration_dwt;
24  static uint32_t t_init;
25  static uint32_t t_out;
26  float_t clock_Hz;
27  uint32_t cpuclk;
28
29  *****
30
```



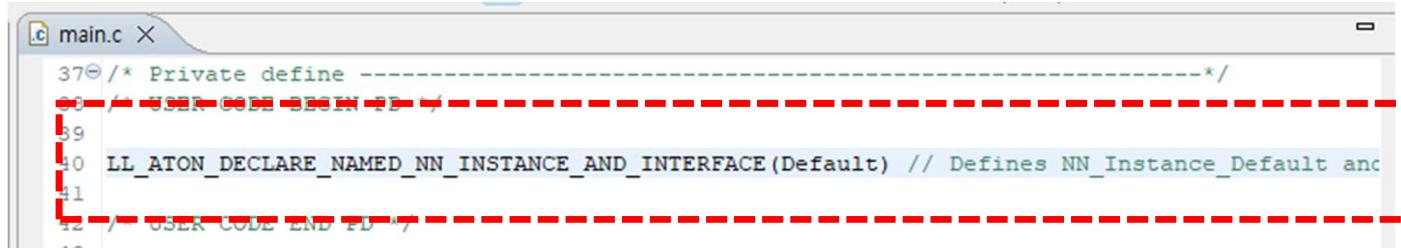
STM32CubeIDE



main.c X

```
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 part of the 'USER CODE BEGIN' section. This section contains two include statements: #include "app_fuseprogramming.h" and #include "ll_aton_runtime.h". The rest of the file consists of standard C code for includes and comments.



main.c X

```
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 part of the 'USER CODE BEGIN' section. This section contains a single macro definition: LL_ATON_DECLARE_NAMED_NN_INSTANCE_AND_INTERFACE(Default). The rest of the file consists of standard C code for defines and comments.



STM32CubeIDE

```
main.c X
59 /* USER CODE END PFP */
60
61/* Prints a message
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 cpuclk;
76
77/******************* DWT INITIALIZATION
78 /*
79 /*****
80void init_dwt()
81 {
82     /* Enable Trace */
83     CoreDebug->DEMCR |= CoreDebug_DEMCR_TRCENA_Msk;
84
85     /* Reset Cycle Counter and Event Counters */
```

```
main.c X
168 /* USER CODE BEGIN Init */
169
170 MX_GPIO_Init();
171 Fuse_Programming();
172 SystemClock_Config_HSI_overdrive();
173 init_external_memories();
174
175 NPURam_enable();
176
177 NPUCache_enable();
178
179 /* Set all required IPs as secure privileged */
180 Security_Config();
181
182
183 /*****
184 /*
185 /*****
186 LL_ATON_RT_RetValues_t ll_aton_rt_ret = LL_ATON_RT_DONE;
187 const EpochBlock_HandleTypeDef *eb_list = LL_ATON_EpochBlockItems_Default();
188
189 /*
190 /* Retrive the start address of the input and output buffer
191 /* reserved in the activation buffer */
192 const LL_Buffer_InfoTypeDef *ibuffersInfos = NN_Interface_Default.input_buffers;
193 const LL_Buffer_InfoTypeDef *obuffersInfos = NN_Interface_Default.output_buffers;
194 buffer_in = (float *)LL_Buffer_addr_start(&ibuffersInfos[0]);
195 buffer_out_p = (float *)LL_Buffer_addr_start(&obuffersInfos[0]);
196
197 LL_ATON_RT_RuntimeInit();
198
199 /*****
```



STM32CubeIDE

```
328
329     /* Infinite 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 }
```

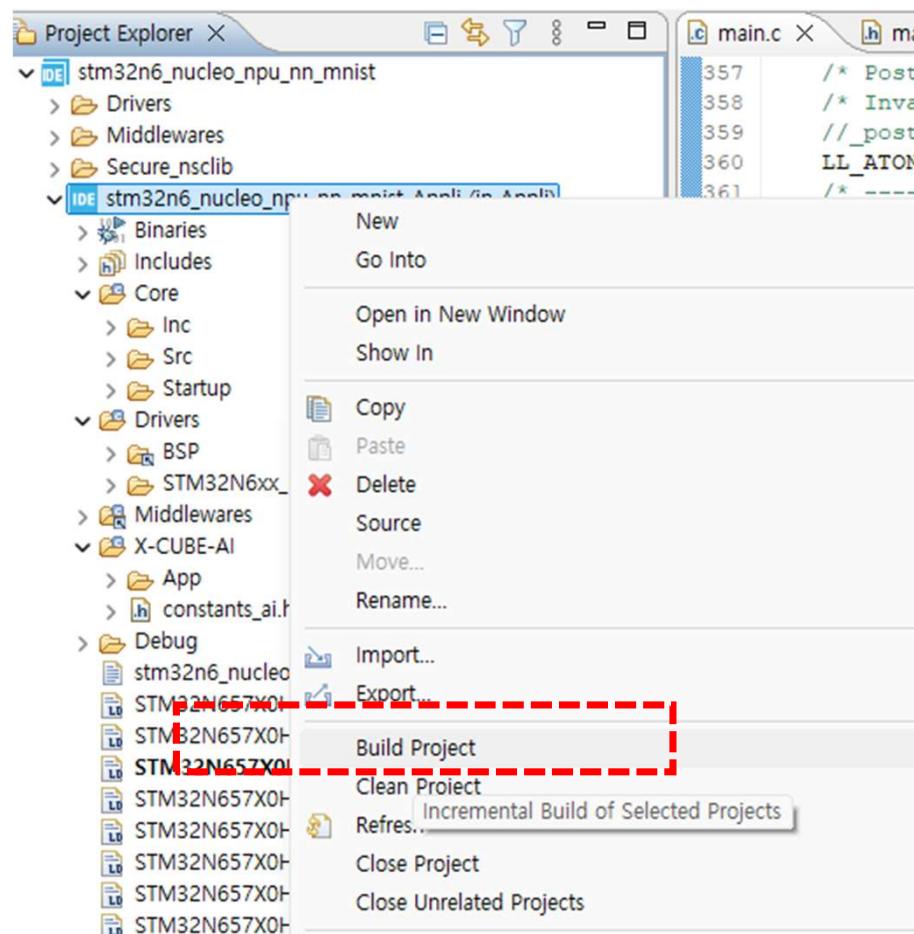
```
#include "stm32n6xx_nucleo.h"  
#include "stm32n6xx_nucleo_xspi.h"
```

```
main.c
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 typedef void CMSE_NS_CALL (*funcptr)(void);
52 #endif
```



STM32CubeIDE



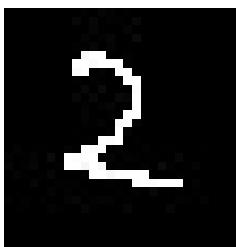
The screenshot shows the STM32CubeIDE Console tab. It displays the build logs for the target "stm32n6_nucleo_npu_nn_mnist_Appli". The logs show the compilation process, including assembly dump and object copy steps, and finally the completion message: "20:15:51 Build Finished. 0 errors, 19 warnings. (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



Screenshot of STM32CubeIDE showing a debugging session. The project is 'stm32n6_nucleo_npu_nn_mnist_Appli'. A breakpoint is set at line 378. The code snippet shows the inference loop and a softmax calculation:

```
357     /* Post-process the output buffer */
358     /* Invalidate the associated CPU cache region if requested */
359     //_post_process(buffer_out);
360     LL_ATON_RT_DeInit_Network(&NN_Instance_Default);
361     /* -----
362     /* - End of Inference - */
363     /* ----- */
364 #else
365     LL_ATON_RT_Main(&NN_Instance_Default);
366 #endif
367
368     duration_dwt = time_out();
369     duration_us = (uint32_t)((float_t)duration_dwt * 10000
370 //printf("Inference: %d us (%d) cycles)\r\n",duration_u
371
372     sum = 0;
373     for(int i = 0 ; i < 10; i++)
374     {
375         nn_buffer_out[i] = *((float *)nn_buffer_out_p+i);
376         sum += nn_buffer_out[i];
377     }
378     HAL_Delay(100);
379     /* USER CODE END WHILE */
380
381     //MX_X_CUBE_AI_Process();
```

The Variables view shows the current state of variables:

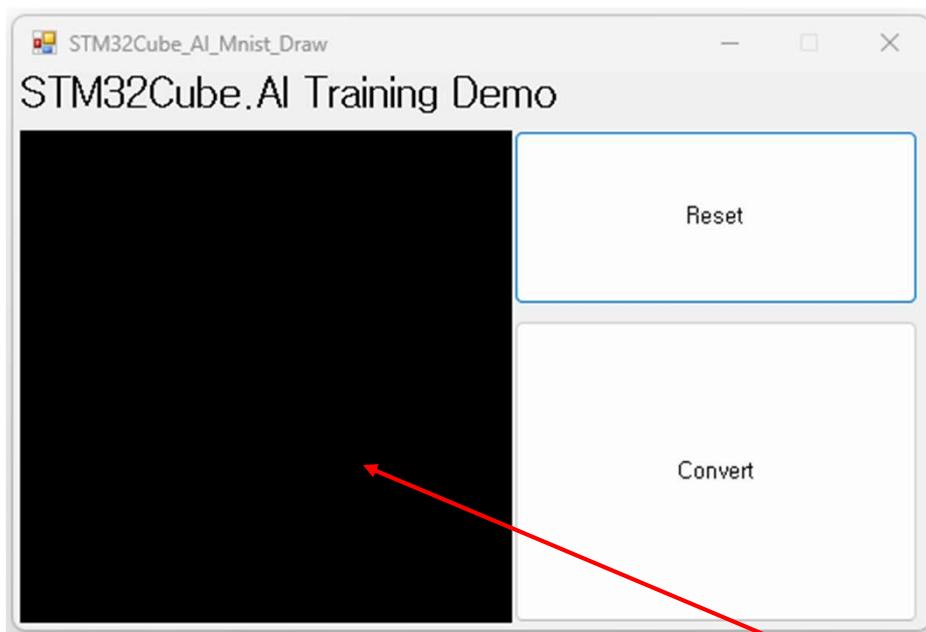
Expression	Type	Value
nn_sum	float	1
nn_buffer_out	float [10]	0x3401c058 <nn_buffer_out>
(nn_buffer_out[0])	float	0
(nn_buffer_out[1])	float	0
(nn_buffer_out[2])	float	0.98828125
(nn_buffer_out[3])	float	0.0078125
(nn_buffer_out[4])	float	0
(nn_buffer_out[5])	float	0
(nn_buffer_out[6])	float	0
(nn_buffer_out[7])	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."

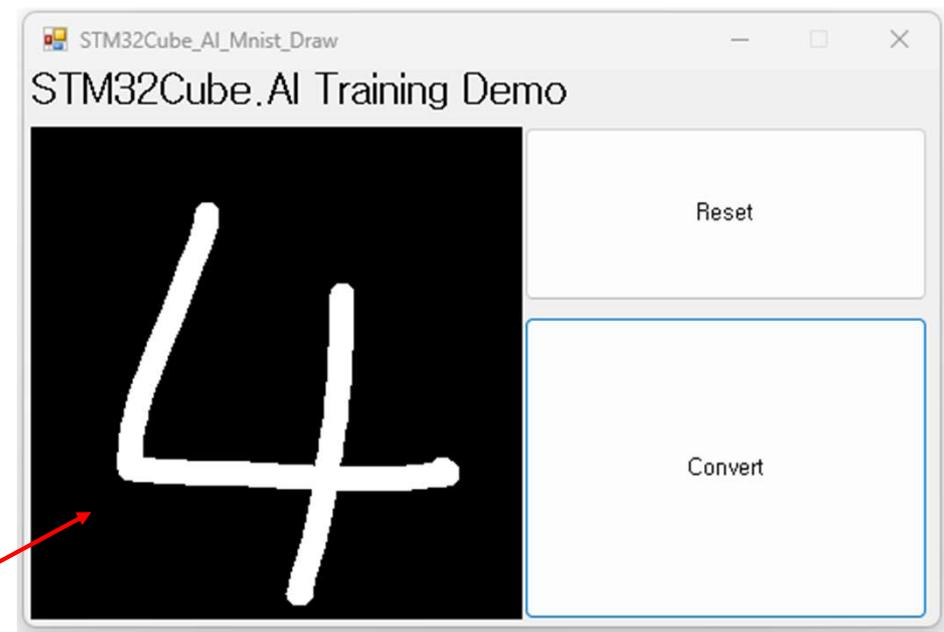


Input Data Generation

Run to the “STM32Cube_AI_Mnist_example.exe”

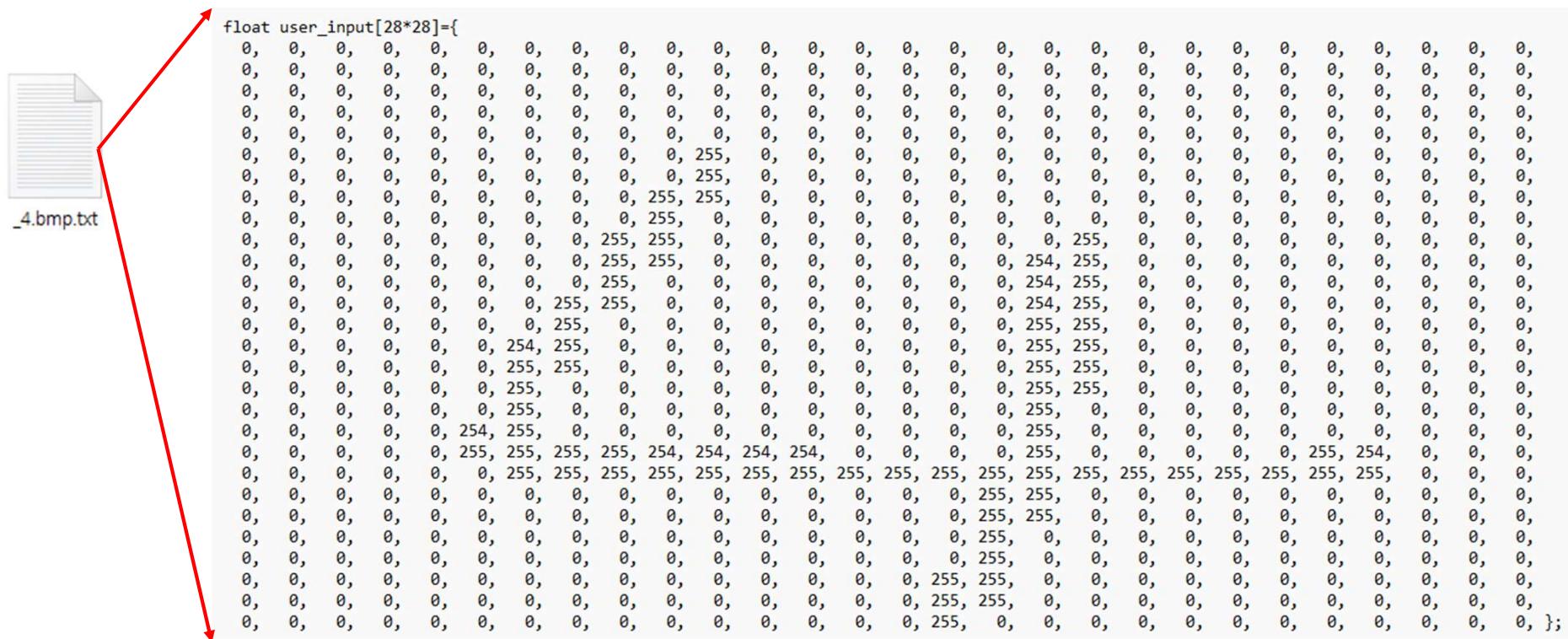


Convert the entered number



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. The page lists six software packages for the STM32N6 with Neural-ART accelerator, each with a 'Request Software' button.

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



<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.

The image shows two screenshots of a GitHub repository. The left screenshot shows the main directory structure of the repository, which contains sub-directories for various machine learning models: audio_event_detection, doc, hand_posture, human_activity_recognition, image_classification, instance_segmentation, object_detection, pose_estimation, semantic_segmentation, speech_enhancement, and CODE_OF_CONDUCT.md. The right screenshot shows the object_detection sub-directory, which contains sub-directories for different models: ssd_mobilenet_v2_fpnlite, st_ssd_mobilenet_v1, st_yolo_lc_v1, st_yolo_x, tiny_yolo_v2, yolov8n, LICENSE.md, README.md, pose_estimation, semantic_segmentation, and speech_enhancement. It also includes README.md and Object detection STM32 model zoo sections.



<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 page for 'stm32ai-modelzoo/object_det'. The main content area displays the README.md file, which includes sections for 'Yolov8n object detection quantized', 'Use case : Object detection', and 'Model description'. The 'Model description' section states: 'Yolov8n is a lightweight and efficient object detection model designed for instance segmentation tasks. Once family of models, known for their real-time object detection capabilities. The "n" is optimized for speed and resource efficiency, making it suitable for deployment on devices and embedded systems.' Below the README, there is a 'Network information' section. The left sidebar shows a tree view of the repository's files and folders, including 'main', 'audio_event_detection', 'doc', 'hand_posture', 'human_activity_recognition', 'image_classification', 'instance_segmentation', 'object_detection' (which is expanded to show 'ssd_mobilenet_v2_tflite', 'st_ssd_mobilenet_v1', 'st_yolo_x', 'tiny_yolo_v2', 'yolov8n' (expanded to show 'LICENSE.md', 'README.md', 'yolov8n_256_config.yaml', 'pose_estimation'), and 'pose_estimation').



<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: 'Reference NPU memory footprint based on COCO Person dataset' and 'Reference NPU inference time based on COCO Person dataset'. Both tables compare four different model configurations (YOLOv8n per channel) across various resolutions, datasets, and boards.

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

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 'main' branch, which is 8 commits ahead of and 1214 commits behind the 'ultralytics/ultralytics:main' branch. The commits are listed in descending order of age, with the most recent at the top.

Commit	Author	Message	Date
342ac37 · 4 days ago	MCHSTM	add per tensor yolov8 person models	11,110 Commits
Run --slow tests for Raspberry Pi (ultralytics#10132)			11 months ago
ultralytics 8.2.1 adopt v8.2 GitHub assets URL (ultralytics#...)			11 months ago
ultralytics 8.2.2 replace COCO128 with COCO8 (ultralytics...)			11 months ago
add per tensor yolov8 person models			4 days ago
Run --slow tests for Raspberry Pi (ultralytics#10132)			11 months ago
Normalize yolov8 pose output			9 months ago
STEdgeAI: introduce STEdgeAI object detection models			9 months ago
YAML reformat (ultralytics#7669)			last year
Update CITATION.cff with cifnnt corrections (ultralytics#8658)			last year
ultralytics 8.0.224 Counting and Heatmaps updates (ultral...			2 years ago
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



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 yellow button with 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". It describes the service as a free online service for developing AI on ST devices, mentioning support for STM32MP1 and STM32MP2, and integrates with PyTorch. A yellow "Open online platform" button is located at the bottom of this section.

The screenshot shows the ST Edge AI Developer Cloud home page with a white background. At the top, there's a navigation bar with links for Home, Help, and Log in. The main heading is "OPTIMIZE YOUR TRAINED NEURAL NETWORK". Below it, a sub-headline reads: "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". A yellow "START NOW" button is located on the right. The page features several sections: "ST Edge AI model zoo" (with icons for STM32MP1, STM32MP2, STM32MP3, and STM32MP4), "Bring Your Own Model" (with a PyTorch icon), "Optimize and benchmark your NN model" (with icons for Graph optimizer, Quantizer, and Memory optimizer), and a "Generate code and download" section (with icons for C, C++, and MX). The overall layout is clean and modern, emphasizing the tools available for AI development on ST hardware.



<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.

The image shows two browser tabs. The left tab is the login page for 'my.st.com', prompting for email and password. The right tab is the 'ST Edge AI Developer Cloud' dashboard, which includes a 'Your model library' section for uploading local models and a 'Pick a model from ST Model Zoo' section for selecting pre-trained models.

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

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 MB

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...
Content Length: 31.09 kB
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

K Import

CNN2D_ST_HandPosture_8classes_hand_posture_ST_VL53L8CX_handpostur...
Content Length: 72.52 kB
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

K Import

cnn_8x8x8_ispu_wand_lsmfdso16is_ISPU_reference_dataset.h5
Content Length: 90.02 kB
Description: CNN1D for air writing recognition that detects the letters I, S, P, and U
Use case: ispu_wand
Dataset used: lsmfdso16is_ISPU_reference_dataset

K Import



ST Edge AI Developer Cloud

The screenshot shows the ST Edge AI Developer Cloud interface. At the top, there's a navigation bar with icons for back, forward, search, and user profile. The main title is "ST Edge AI Developer Cloud". Below the title, a sub-header reads "OPTIMIZE YOUR TRAINED NEURAL NETWORK". A sub-sub-header says "Optimize and measure performance of your Artificial Intelligence library for STMicroelectronics microcontrollers, microprocessors and smart-sensors". A note below states "This free online tool allows you to generate and test optimized AI libraries based on your trained Neural Networks".

The interface is divided into two main sections:

- Your model library:** On the left, there's a dashed box containing a "New model" button with an upward arrow icon. Below it, instructions say "Drop your model or click here to open a file browser. Supported models are Keras, ONNX and TFLite (.h5, .hdf5, .onnx, .tflite)". A note specifies "Max file size: 256 MiB". Below this are buttons for "Date", "Name", and "Size".
- Pick a model from ST Model Zoo:** This section has dropdown menus for "Target" (2 selected), "Use case" (1 selected), and "Neural Network" (17 selected). It lists several 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.tfliteEach item includes a preview thumbnail, a description, and an "Import" button. The "tiny_yolo_v2_224_int8_object_detection_ST_Person.tflite" entry is highlighted with a red dashed rectangle.

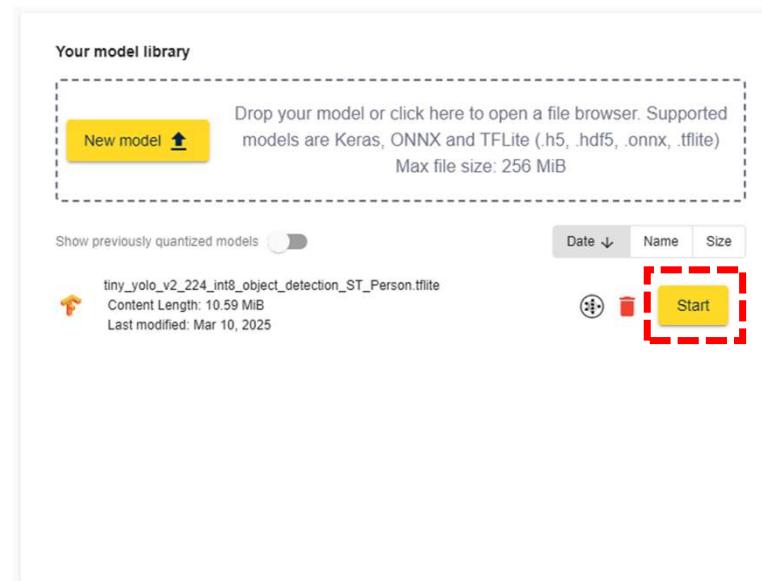
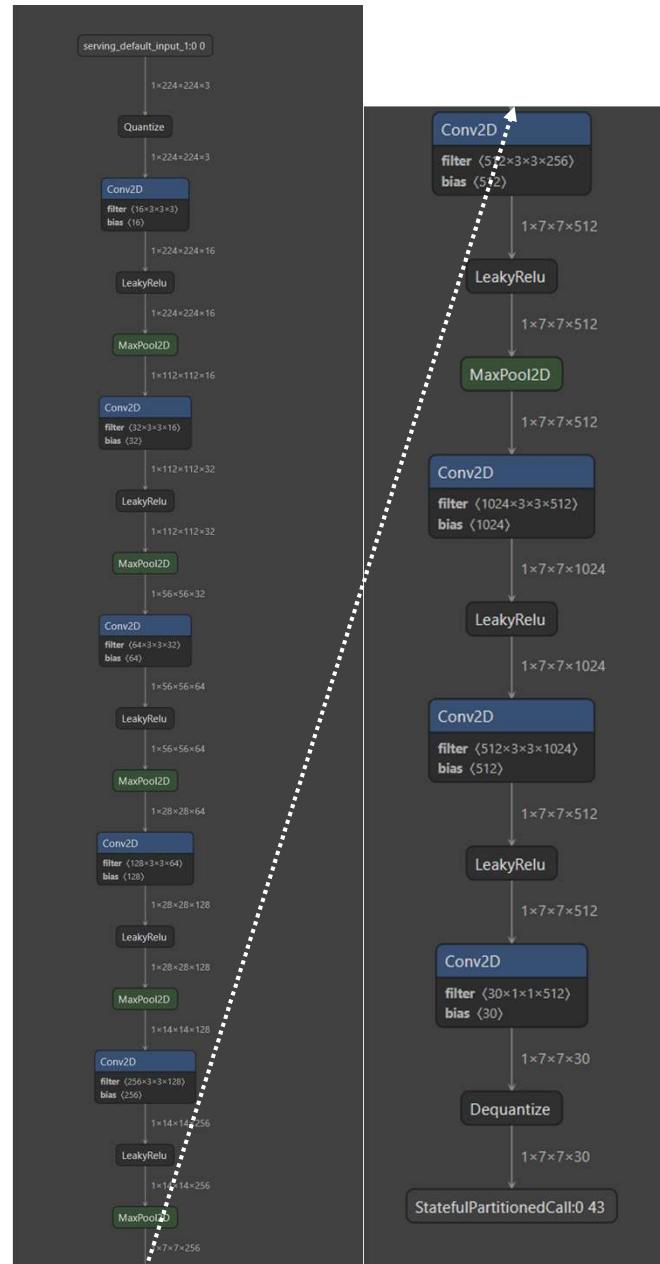
To the right, there's a larger panel titled "Your model library" which displays the imported model details:

- A dashed box contains the "New model" button and the import note about supported file types and max size.
- Below the box are buttons for "Date", "Name", and "Size".
- On the right side of the panel are three small icons: a red square with a white circle, a red 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 icons for Quantize (Optional), Optimize, Benchmark, Results, and Generate. Below the navigation bar, a red dashed box highlights the "Select a platform" tab, which is currently active. The main content area has a title "Select a platform" and a subtitle "Select which platform is most suitable for your use-case".

Underneath, there is a section titled "Model currently selected" showing "TINY_YOLO_V2_224_INT8_OBJECT_DETECTION_ST_PERSON.TFLITE". A red dashed box highlights the "INPUT" and "OUTPUT" fields, which show "224x224x3 (8 bits)" and "7x7x30 (32 bits)" respectively, along with a "MODEL TYPE" field showing "STAI_FORMAT_S32".

Below this, there is a "Select another model" button and a "Show Graph" button.

The main content area also includes a "Select ST Edge AI Core Version" section showing "ST Edge AI Core 2.0.0".

At the bottom, there are three sections:

- STM32 MCUs**: Shows two green boards (STM32 Discovery Kits and Nucleos). A red dashed box highlights the "Select" button.
- STM32 MCU with Neural-ART™**: Shows a blue board (STM32 Nucleo with NPU). A red dashed box highlights the "Select" button.
- STM32 MPUs**: Shows a blue board (STM32MP157C-DK). A red dashed box highlights the "Select" button.

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

[Show terminal](#) [Launch quantization ▶](#)



Optimize

The screenshot shows the 'Model optimization' step in the workflow. A red dashed box highlights the 'Optimize' button. Below it, the title 'Model optimization' and the subtitle 'Use the different options to optimize your neural network' are visible.

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 Latest	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 Default	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-b3
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 ST Neural-ART Model Benchmarking interface. At the top, a horizontal progress bar indicates 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

Start Benchmark ▶️ 🔍 ⚙️ ⚡ ...

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

Circular progress bar with icons: checkmark, gear, play/pause, export, settings, ellipsis.

Schedule a benchmark



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

Measured inference time 30.57 ms

Circular progress bar with icons: checkmark, gear, play/pause, export, settings, ellipsis.

Results of current benchmarks session												Show performance summary	Export as CSV	Delete	⋮
<input type="checkbox"/>	Neural-ART Accelerator™	Platform	MACC	Duration (ms)	Weights Size	Activation Size	Flash size	RAM size	Device	Clock	Use External Flash	Use External RAM			
<input type="checkbox"/>	Art	STM32 NPU	0	30.57	10.55 MiB	343 KiB	10.77 MiB	343.01 KiB	STM32N6570-DK	800MHz	undefined	undefined			
<input type="checkbox"/>	rt	STM32 NPU	0	30.57	10.55 MiB	343 KiB	10.77 MiB	343.01 KiB	STM32N6570-DK	800MHz	undefined	undefined			



Result

The screenshot shows the ST Neural-Art tool interface for model optimization and benchmarking. The top navigation bar features 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 steps, the 'Benchmarking results' section displays the following information:

Model currently selected: TINY_YOLO_V2_224_INT8_OBJECT_DETECTION_ST_PERSON.TFLITE

Current parameters:

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

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 at the bottom include 'Show performance summary', 'Export as CSV', 'Refresh', 'Delete', and a dropdown for 'Items per page'.



Generate

Generate your project

Generate all what you need to start your project

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

STM32N6570-DK

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

© 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.

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

