SMH Technologies®

04/03/2025

Driver v. 5.07

Moreno Ortolan

Interfacing FlashRunner 2.0 with NXP IMX

SMH Technologies S.r.l.

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

NXP IMX Introduction

NXP i.MX Applications Processors

Multicore solutions for multimedia and display applications with high-performance and low-power capabilities that are scalable, safe and secure.
i.MX applications processors are part of the EdgeVerse™ edge computing platform built on a foundation of scalability, energy efficiency, security, machine learning and connectivity.

i.MX Family Products

i.MX RT Series
Arm® Cortex®-M7/M4/M33
Crossover MCUs with real-time functionality and MCU usability for next-generation consumer and industrial IoT applications.

i.MX 9 Series
Higher performance applications cores, an independent MCU-like real-time domain, EdgeLock® secure enclave and ML acceleration for the edge.

i.MX 8 Series
Cortex-A53, Cortex-A72, Cortex-A35 + Cortex-M4
Powerful solutions with advanced neural network processing, graphics, machine vision, video, audio and voice with many supporting safety critical applications.

i.MX 7 Series
Cortex-A7 + Cortex-M4
Low-power solutions for secure, wearable and portable IoT applications.

i.MX 6 Series
Cortex-A9, Cortex-A9 + Cortex-M4, or Cortex-A7
General-purpose solutions with balanced features, performance and scalability for automotive, consumer and industrial applications.

i.MX28 Series
Arm9
Power management and connectivity features for general-purpose automotive, consumer and industrial applications.

i.MX RT Crossover MCUs

i.MX RT Crossover MCUs feature the high-performance Arm® Cortex®-M core and Zephyr RTOS functionality in a real-time microcontroller.
NXP i.MX RT Crossover MCUs are optimized for real-time Ethernet protocols in industrial IoT and automotive applications.

Product	CPU	Package	Memory	Graphics Acceleration	Display Interfaces	Camera Interfaces	Audio	USB with PHY	Ethernet	CAN
i.MX RT1180 [1]	Arm Cortex-M7 @800 MHz + Arm Cortex-M33 @240 MHz	289 BGA 144 BGA	1.5 MB SRAM	-	-	-	4 x I²S, S/PDIF, DMIC	2	1 x independent 1 Gbit/s TSN MAC end point 5-port (4 external + 1 internal) TSN Switch with 1 Gbit/s TSN MAC, EtherCAT, OPC UA	3 x CANFD
i.MX RT1170	Arm Cortex-M7 @1 GHz + Arm Cortex-M4 @400 MHz	289 BGA	2 MB SRAM	2D GPU, P x P	Parallel, MIPI	Parallel, MIPI	4 x I²S, S/PDIF, DMIC	2	2 x Gbit/s, 1 x 10/100	3 x CANFD
i.MX RT1160	Arm Cortex-M7 @600 MHz + Arm Cortex-M4 @240 MHz	289 BGA	1 MB SRAM	2D GPU, P x P	Parallel, MIPI	Parallel, MIPI	4 x I²S, S/PDIF, DMIC	2	1 x Gbit/s, 1 x 10/100	3 x CANFD
i.MX RT1064	Arm Cortex-M7 @600 MHz	196 BGA	1 MB SRAM, 4 MB Flash	P x P	Parallel	Parallel	3 x I²S, S/PDIF	2	2 x 10/100	2 x FlexCAN, 1 x CANFD
i.MX RT1060	Arm Cortex-M7 @600 MHz	196 BGA 225 BGA	1 MB SRAM	P x P	Parallel	Parallel	3 x I²S, S/PDIF	2	2 x 10/100	2 x FlexCAN, 1 x CANFD
i.MX RT1050	Arm Cortex-M7 @600 MHz	196 BGA	512 kB SRAM	P x P	Parallel	Parallel	3 x I²S, S/PDIF	2	1 x 10/100	2 x FlexCAN
i.MX RT1040	Arm Cortex-M7 @600 MHz	169 BGA	512 kB SRAM	P x P	Parallel	-	3 x I²S, S/PDIF	1	1 x 10/100	2 x FlexCAN 1 x CANFD

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

i.MX RT1040	Arm Cortex-M7 @600 MHz	169 BGA	512 kB SRAM	P x P	Parallel	-	3 x I²S, S/PDIF	1	1 x 10/100	2 x FlexCAN 1 x CANFD
i.MX RT1024	Arm Cortex-M7 @500 MHz	144 LQFP	256 kB SRAM, 4 MB Flash	-	-	-	3 x I²S, S/PDIF	1	1 x 10/100	2 x FlexCAN
i.MX RT1020	Arm Cortex-M7 @500 MHz	100 LQFP, 144 LQFP	256 kB SRAM	-	-	-	3 x I²S, S/PDIF	1	1 x 10/100	2 x FlexCAN
i.MX RT1015	Arm Cortex-M7 @500 MHz	100 LQFP	128 kB SRAM	-	-	-	3 x I²S, S/PDIF	1	-	-
i.MX RT1010	Arm Cortex-M7 @500 MHz	80 LQFP	128 kB SRAM	-	-	-	2 x I²S, S/PDIF	1	-	-
i.MX RT600	Arm Cortex-M33 @300 MHz + Cadence® Tensilica® HiFi 4 @600 MHz	176 BGA, 249 FOWLP, 114 CSP	4.5 MB SRAM	-	-	-	8 x I²S 8-ch DMIC	1	-	-
i.MX RT500	Arm Cortex-M33 @275 MHz + Cadence® Tensilica® Fusion F1 @275 MHz	249 FOWLP	5 MB SRAM	2D GPU	Parallel, MIPI	Parallel	12 x I²S 8-ch DMIC	1	-	-

i.MX 9 Applications Processors

Building on the market-proven i.MX 6 and i.MX 8 series, i.MX 9 series applications processors bring together higher performance applications cores, an independent MCU-like real-time domain, Energy Flex architecture, state-of-the-art security with EdgeLock® secure enclave and dedicated multi-sensory data processing engines (graphics, image, display, audio and voice).
The i.MX 9 series, part of the EdgeVerse™ edge computing platform, integrates hardware neural processing units across many members of the series for acceleration of machine learning applications at the edge.

Product Family	Cortex-A55	Cortex-M33	Cortex-M7	DSP	NPU	GPU	ISP	Display Resolution and Interfaces	Camera Resolution and Interfaces	PCIe	USB 2.0	USB 3.0	Ethernet	External Memory	CAN-FD
i.MX 91[1] Secure, Energy-Efficient Applications Processor Family Brings Essential Linux Capabilities to Thousands of Edge Applications	1	-	-	-	-	-	-	24 bit-per-pixel parallel RGB/YUV Display	8-bit parallel RGB/YUV Camera	-	2x USB 2.0 (Dual mode, w/Type C)	-	2x 1GbE (1 w/TSN)	3x SD/SDIO3.0 /eMMC5.1, 1x Octal SPI	2
i.MX 93[1] ML Acceleration, Power Efficient MPU for Automotive, Consumer and Industrial IoT	2	1	-	-	1	2D	-	1080p60 MIPI DSI (4-lane), 720p60 LVDS (4-lane), 24-bit parallel RGB	1080p60 MIPI CSI (2-lane), 8-bit parallel YUV/RGB	-	2	-	2x 1GbE with 1 w/TSN	3x SD/SDIO3.0 /eMMC5.1, 1x Octal SPI	2
i.MX 95[1] Safe, Secure, Connected Applications Processor for Automotive, Industrial and IoT Edge	Up to 6	1	1	Immersiv3D™ Audio Framework	1	3D	1	4K30P, 3840x1440P60 MIPI-DSI (4-lane) Up to 1080P LVDS (2x 4-lane or 1x 8-lane)	2x4kp30, 4x1080p60, 8x1080p30 MIPI-CSI (2x 4-lane)	2 Gen 3.0 (1-lane)	1	1	10 GbE + 2x 1 GbE with 1x TSN	3x SD/SDIO3.0 /eMMC5.1, 1x Octal SPI	5

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

SMH Technologies®

i.MX 8 Series Applications Processors Multicore Arm® Cortex® Processors

The i.MX 8 series of applications processors, part of the EdgeVerse™ edge computing platform, is a feature- and performance-scalable multicore platform that includes single-, dual- and quad-core families based on the Arm® Cortex® architecture—including combined Cortex-A72 + Cortex-A53, Cortex-A35, Cortex-M4 and Cortex M7-based solutions for advanced graphics, imaging, machine vision, audio, voice, video and safety-critical applications.

	CPU, GPU and DSP									HMI and Multimedia				GPU Libraries and Extensions				Interfaces			Memory Types			
Product Family	Cortex-A72	Cortex-A53	Cortex-A35	Cortex-M4F	Cortex-M33	Cortex-M7	DSP	NPU	GPU	**Display Resolution and Interfaces**	**Camera Interfaces**	Video Decode Resolution (Top Codecs)	Video Encode Resolution (Top Codecs)	OpenVX™ (vision)	OpenGL® ES	OpenCL™	Vulkan™	PCIe	Gigabit Ethernet	10/100 Ethernet	LPDDR4	DDR4	DDR3L	ECC option
i.MX 8 Advanced Graphics, Performance and Virtualization	2	4	-	2	-	-	1	-	2	2x LVDS 2x MIPI DSI 1x HDMI TX/RX	2x MIPI-CSI	4K (h.265, h.264)	1080p30 (h.264)	✓	3.1	1.2	✓	2	2x	-	✓	-	-	-
i.MX 8M Advanced Audio, Voice and Video	-	4	-	1	-	-	-	-	1	1x MIPI DSI 1x HDMI	2x MIPI-CSI	4Kp60 with High Dynamic Range (h.265, VP9); 4Kp30 (h.264, VP8)	1080p30 (h.264) - SW	-	3.1	1.2	✓	2	1x	-	✓	✓	✓	-
i.MX 8M Mini Embedded Consumer and Industrial Applications	-	4	-	1	-	-	-	-	1	1x MIPI DSI	1x MIPI-CSI	1080p60 (h.265, VP9, h.264, VP8)	1080p60 (h.264)	-	2.0	-	-	1	1x	-	✓	✓	✓	-
i.MX 8M Nano Embedded Consumer and Industrial Applications	-	4	-	-	-	1	-	-	1	1x MIPI DSI	1x MIPI-CSI	-	-	-	3.1	1.2	✓	-	1x	-	✓	✓	✓	-
i.MX 8M Plus Machine Learning, Vision, Multimedia and Industrial IoT	-	4	-	-	-	1	1	1	1	1x MIPI DSI 1x HDMI 1x LVDS	2x MIPI-CSI Dual Camera ISP	1080p60 (h.265, h.264, VP9, VP8)	1080p60 (h.265, h.264)	-	3.1	1.2 FP	✓	1	2x (1 x/ TSN)	-	✓	✓	-	✓
i.MX 8ULP[1] Industrial, Mobile and Smart Home Applications	-	-	2	-	1	-	2	-	2	1x MIPI DSI 24-bit RGB EPD	1x MIPI-CSI	-	-	-	3.1	1.2	✓	-	-	1	✓	-	-	-
i.MX 8X Advanced Graphics and Efficient Performance	-	-	4	1	-	-	1	-	1	1x Parallel 2x LVDS/MIPI DSI Combo	1x MIPI-CSI 1x Parallel	4K (h.265); 1080p60 (h.264, VP8)	1080p30 (h.264)	-	3.1	1.2 EP	✓	1	2x	-	✓	-	✓	✓
i.MX 8XLite Secure Telematics and Connected Industrial Control	-	-	2	1	-	-	-	-	-	1x Parallel Display	-	-	-	-	-	-	-	1	2x (TSN+AVB)	-	✓	-	✓	✓

SMH Technologies S.r.l.

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

SMH Technologies®

i.MX 7 Series Applications Processors: Multicore Arm® Cortex®-A7, Cortex-M4

The i.MX 7 series, part of the EdgeVerse™ edge computing platform, offers highly-integrated multimarket applications processors designed to enable secure and portable applications within the Internet of Things.

				Graphics		Comms			Interfaces		Security	
Product	CPU	Memory	Packages	2D Acceleration	3D Acceleration	MMC/SDIO	Gigabit Ethernet	HS USB OTG	PCIe	Display I/F	Advanced Security	IPM
i.MX 7ULP Ultra Low Power	Single Cortex-A7 @800 MHz + Cortex-M4	96 kB ROM 512 kB SRAM 256KB L2 Cache	MAPBGA	✓	✓	2	0	1	-	MIPI	✓	✓
i.MX 7Solo Heterogeneous Processing	Single Cortex-A7 @800 MHz + Cortex-M4 @200 MHz	96 kB ROM 256 kB SRAM 512KB L2 Cache	MAPBGA	-	-	3	1	1	-	MIPI, Parallel	✓	✓
i.MX 7Dual High Performance	Dual Cortex-A7 @ up to 1.2 GHz + Cortex-M4 @200 MHz	96 kB ROM 256 kB SRAM 512KB L2 Cache	MAPBGA	-	-	3	2	2	✓	EPD, MIPI, Parallel	✓	✓

i.MX 6 Series Applications Processors: Multicore, Arm® Cortex®-A7 Core, Cortex-A9 Core, Cortex-M4 Core

The i.MX 6 series of applications processors, part of the EdgeVerse™ edge computing platform, offers a feature- and performance-scalable multicore platform that includes single-, dual- and quad-core families based on the Cortex architecture—including Cortex-A9, combined Cortex-A9 + Cortex-M4 and Cortex-A7 based solutions.

		Packages	HMI and Multimedia				Communications			Interfaces				Security	Automotive
Product	CPU	BGA Package Size, Pitch (mm)	Graphics Acc	EPDC (e-Ink)	Display Interfaces	Camera Interfaces	Ethernet	USB Controller	CAN	PCIe lanes	MMC/SDIO	SCIM	ADC Channels	Advanced Security	Automotive
i.MX 6ULZ Ultra Low Cost	Cortex-A7 @900 MHz	14 x 14, 0.8	-	-	-	-	-	2 / 0	-	-	2	-	-	-	-
i.MX 6ULL Cost Optimized	Cortex-A7 @900 MHz	9 x 9, 0.5 14 x 14, 0.8	-	-	✓	✓	2 x 10/100 Mbit/s	2 / 0 **	2	-	2	✓	20	✓	-
i.MX 6UltraLite Efficient and Secure	Cortex-A7 @696 MHz	9 x 9, 0.5 14 x 14, 0.8	-	-	✓	✓	2 x 10/100 Mbit/s	2 / 0 **	2	-	2	✓	20	✓	✓
i.MX 6SLL Efficient Multimedia	Cortex-A9 @800M/1GHz	13x13,0.5 14x14, 0.65	✓	✓	✓	✓	-	2 / 0	-	-	3	-	-	✓	-
i.MX 6SoloLite Multimedia/eReader	Cortex-A9 @1 GHz	13 x 13, 0.5	✓	✓	✓	✓	10/100 Mbit/s	2 / 1	-	-	4	-	-	✓	-
i.MX 6SoloX Secure Processing	Cortex-A9 @1 GHz + Cortex-M4 @227 MHz	14 x 14, 0.65 17 x 17, 0.8 19 x 19, 0.8	✓	-	✓	✓	2 x 1 Gbit/s	2 / 1	2	1	4	-	8	✓	✓

SMH Technologies S.r.l.

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

i.MX 6Solo Multimedia	Cortex-A9 @1 GHz	21 x 21, 0.8	✓	✓	✓	✓	1 Gbit/s	2 / 1	2	1	4	-	-	✓	✓
i.MX 6DualLite 3D Graphics	2 x Cortex-A9 @1 GHz	21 x 21, 0.8	✓	✓	✓	✓	1 Gbit/s	2 / 2	2	1	4	-	-	✓	✓
i.MX 6Dual Advanced 3D Graphics	2 x Cortex-A9 @1.2 GHz	12 x 12, 0.4 (PoP) 21 x 21, 0.8	✓	-	✓	✓	1 Gbit/s	2 / 2	2	1	4	-	-	✓	✓
i.MX 6DualPlus Extreme 3D Graphics	2 x Cortex-A9 @1.2 GHz	21 x 21, 0.8	✓	-	✓	✓	1 Gbit/s	2 / 2	2	1	4	-	-	✓	✓
i.MX 6Quad High Performance	4 x Cortex-A9 @1.2 GHz	12 x 12, 0.4 (PoP) 21 x 21, 0.8	✓	-	✓	✓	1 Gbit/s	2 / 2	2	1	4	-	-	✓	✓
i.MX 6QuadPlus Ultimate Performance	4 x Cortex-A9 @1.2 GHz	21 x 21, 0.8	✓	-	✓	✓	1 Gbit/s	2 / 2	2	1	4	-	-	✓	✓

i.MX28 Applications Processors: Integrated Power Management Unit (PMU), Arm9™ Core

The i.MX28 family integrates display, power management and connectivity features to provide broad levels of integration in Arm9™-based devices.

i.MX28 for Industrial and Consumer Applications

The i.MX28 family of multimedia applications processors is the latest extension of our Arm9 product portfolio. The i.MX28 family reduces system complexity for cost-sensitive applications.

i.MX280
Low-Power, High-Performance

i.MX283
High-Performance, Low-Power

i.MX286
Dual CAN, High Performance, Low Power

i.MX287
Dual Ethernet, Dual CAN, LCD Touch Screen

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

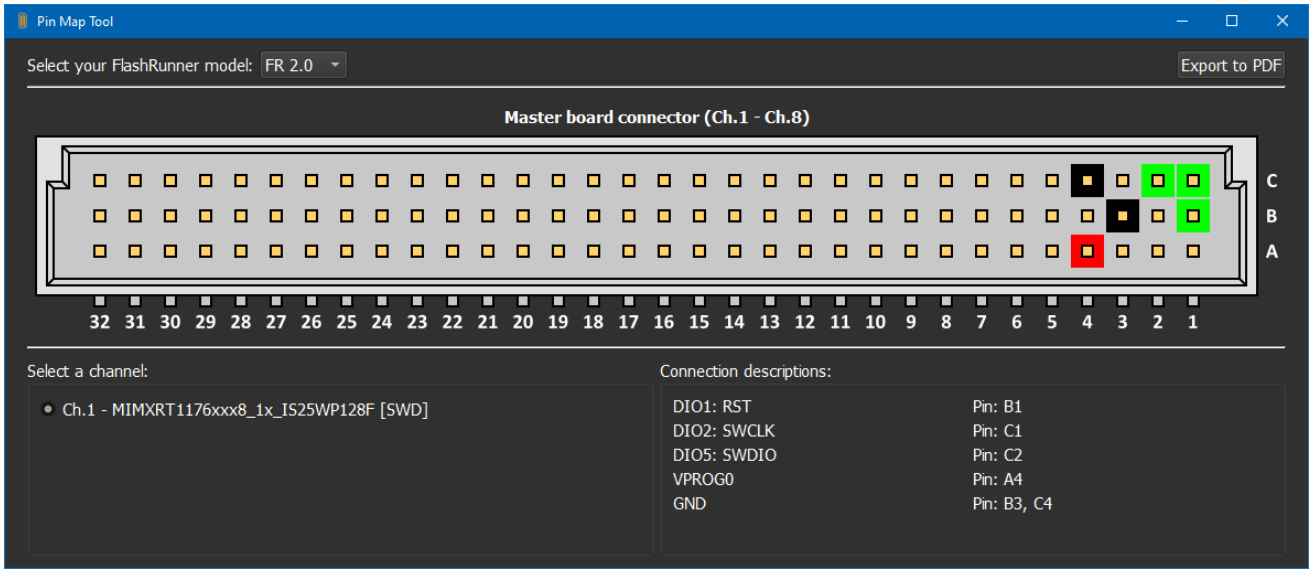
→smh-tech.com
info@smh-tech.com

NXP IMX Protocol and PIN map

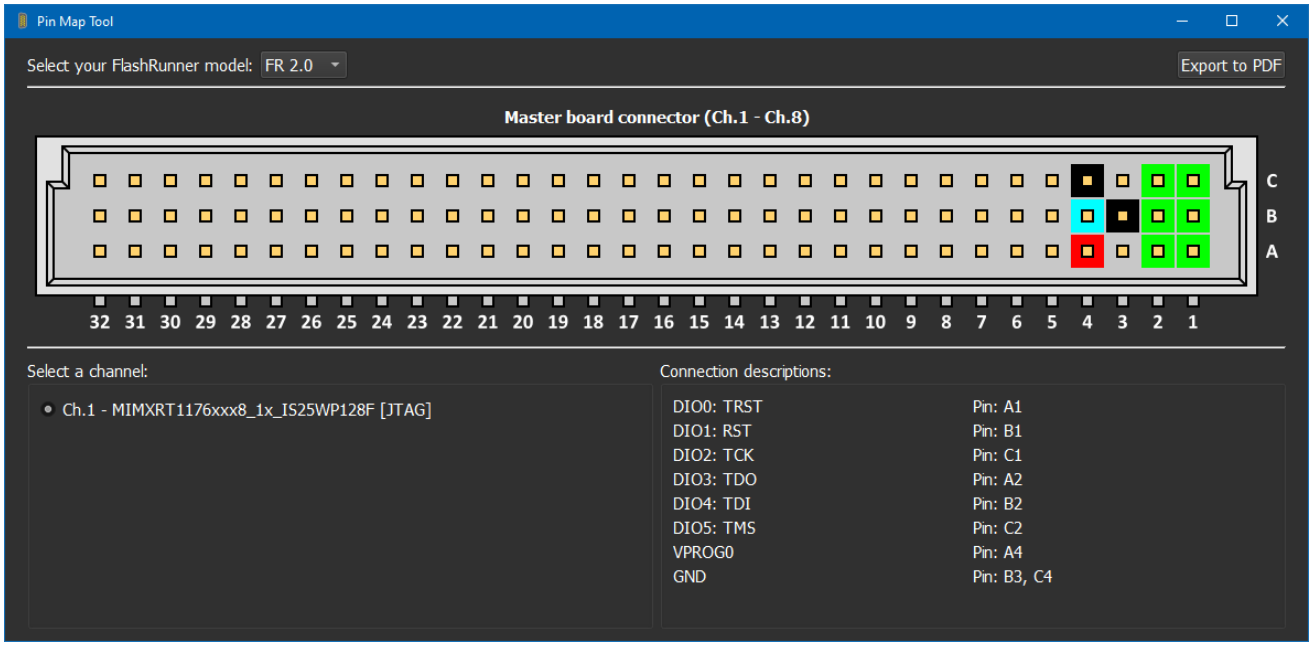
IMX devices support the SWD and JTAG protocol.

```
#TCSETPAR CMODE <SWD/JTAG>
```

NXP IMX SWD PIN MAP



NXP IMX JTAG PIN MAP



HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

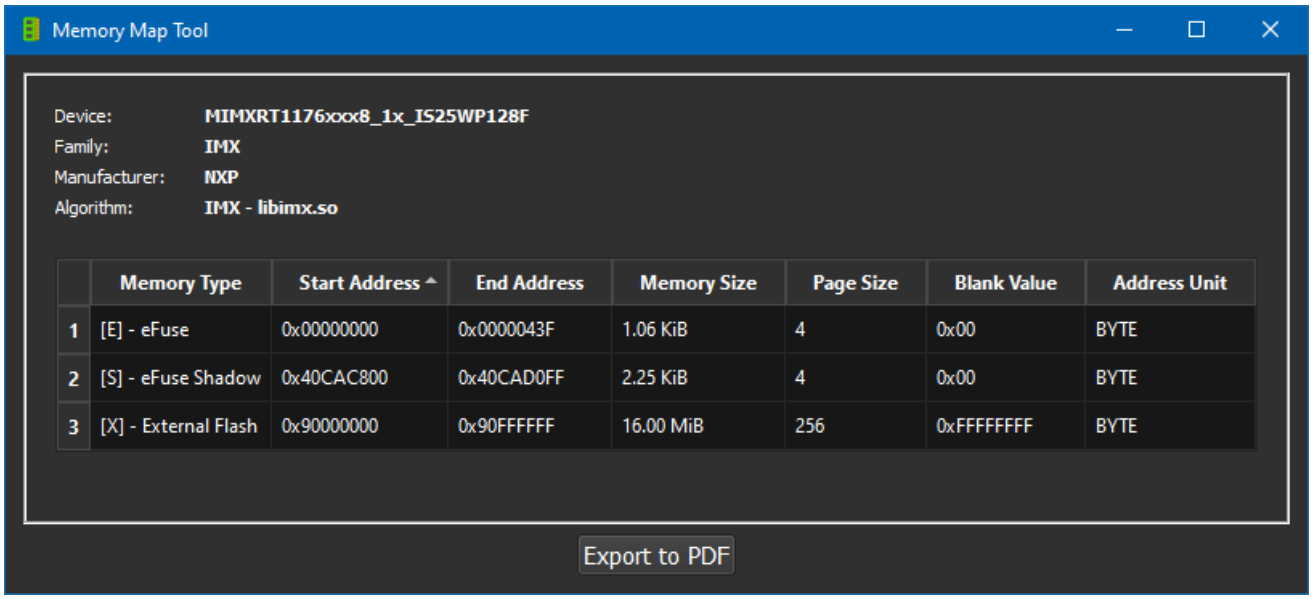
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

NXP IMX Memory Map

Memory Type	Start Address	End Address	Memory Size	Page Size	Blank Value	Address Unit
[E] - eFUSE	0x00000000	0x0000043F	1.06 KiB	4	0x00000000	BYTE
[S] - eFUSE Shadow	0x40CAC800	0x40CAD0FF	2.25 KiB	4	0x00000000	BYTE
[X] - External Flash	0x90000000	0x90FFFFFF	16.00 MiB	256	0xFFFFFFFF	BYTE



NXP IMX eFUSE

NXP IMX eFUSE Introduction

EFUSE are OTP registers and Shadow Registers are a duplicate non-OTP of the EFUSE.

There are some points to keep in mind:

- For some devices the Shadow Registers don't match the whole EFUSE map, some EFUSE must to be read directly.
- There could be a difference between EFUSE and Shadow Registers values.
  For example: iMXRT1170 there are some EFUSE with Redundancy as method for Error Correction.
  For these EFUSE the high part [31:16] of the register is a copy of the low part [15:0]. In the Shadow Registers the high bits are the inverted version of the low bits.
  This is not true for the direct read of the EFUSE.
- It is possible to access the EFUSE with two different addressing:
  - Direct addressing: the address is the actual number of the register. For example: iMXRT1170 has 0x110 EFUSE. A value between 0x00 and 0x10F can be used. The value can be calculated removing the start address of the EFUSE map and dividing by 0x10. Example: EFUSE map starts at 0x800. The EFUSE at 0x9A0 is: (0x9A0-0x800)/0x10 = 0x1A
  - Shadow addressing: the address is in the shadow memory. It is possible to use it for example for a write. The driver will automatically convert it to the direct address.
- The memory map for the EFUSE (E) is byte-oriented ad it is used only if a program using an FRB is performed.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale
Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021
→smh-tech.com
info@smh-tech.com

NXP IMX eFUSE Specific Commands

Here below the dedicated commands for the EFUSE, check also the standard commands.

#TPCMD EFUSE_OVERVIEW

Syntax: `#TPCMD EFUSE_OVERVIEW <Mode>`

`<Mode>` Accepted value is BASIC and ADVANCED

Prerequisites: none

Description: Print on the log the EFUSE content
With BASIC parameter only some basic information is showed, instead with ADVANCED a more complete overview is provided.

Note: This command prints into Real Time Log

#TPCMD EFUSE_RELOAD

Syntax: `#TPCMD EFUSE_RELOAD`

Prerequisites: none

Description: Force the reload of the Shadow Registers with the content of the EFUSE without issuing a reset.

#TPCMD EFUSE_WRITE

Syntax: `#TPCMD EFUSE_WRITE <Address> <Value>`

`<Address>` eFUSE Address
`<Value>` eFUSE value

Prerequisites: none

Description: Write of the selected EFUSE (start address) with the content given (value)
The address can be expressed respect to the Shadow Registers map or passing the exact position in the Fuse Map of the EFUSE
The Shadow Registers are automatically reloaded after a write

#TPCMD EFUSE_COMPARE

Syntax: `#TPCMD EFUSE_COMPARE <Address> <Value>`

`<Address>` eFUSE Address
`<Value>` eFUSE value

Prerequisites: none

Description: Compare of the selected EFUSE (start address) with the content given (value)
The address can be expressed respect to the Shadow Registers map or passing the exact position in the Fuse Map of the EFUSE

#TPCMD EFUSE_LOCK

Syntax: `#TPCMD EFUSE_LOCK <Address> <Value>`

`<Address>` eFUSE Address
`<Value>` eFUSE value

Prerequisites: none

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

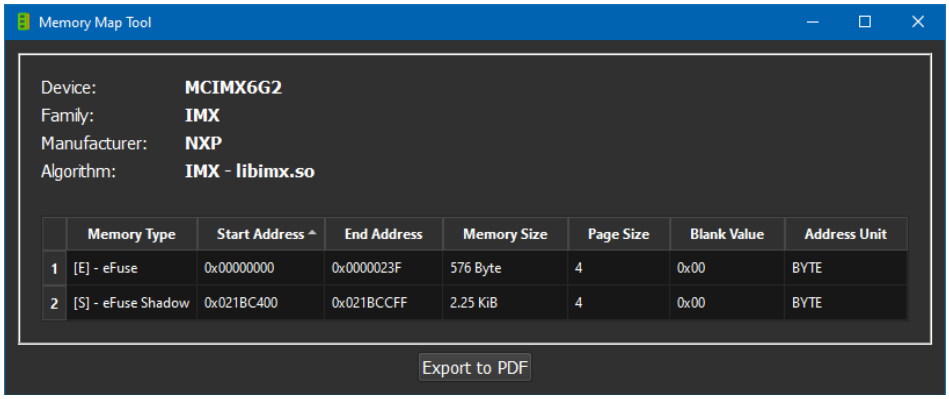
Description: Lock of the selected EFUSE from start address for the size give
This command is not available for all devices and can be used to lock the EFUSE which use Redundancy method as Error Correction
The address can be expressed respect to the Shadow Registers map or passing the exact position in the Fuse Map of the EFUSE.

NXP IMX eFUSE Management

This chapter describes in detail how to manage eFuse of IMX devices using the eFuse memory map or the Shadow eFuse memory map.

NXP IMX eFUSE Introduction

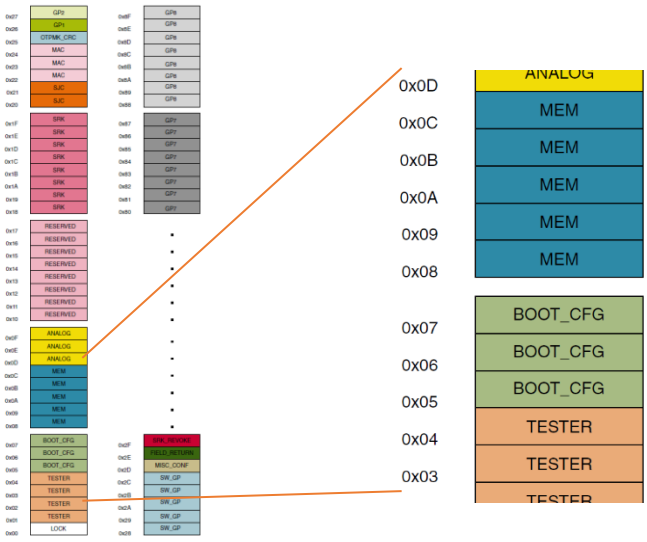
Let's start from the memory map of a generic IMX device:



As you can see the [E] eFuse memory (576Byte) has a different size than the [S] eFuse Shadow memory (2.25KiB).
Later we will understand why these two memories have different sizes.

Before speaking about these two memories in depth we analyze for a moment how the eFuse are physically arranged inside this IMX device.

Here you can see the eFuse memory:



As you can see, the reference manual lists eFuses with addresses that increment 1 by 1.
For example, the BOOT_CFG eFuse at address 0x7 is followed by the MEM eFuse at address 0x8.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale
Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255
D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021
→smh-tech.com
info@smh-tech.com

However, each of these eFUSE is made up of 32 bits (1 word of 4 bytes) so it means that every time the address is increased by 1, we are talking about moving forward by 4 bytes (1 Word).

This means that if there are 0x90 eFuse as in this case then considering 1 word per eFuse, the total size will be:
Total size = 0x90 * 4byte = 0x240 byte.

This explains why [E] eFUSE memory starts at 0x00000000 and ends at 0x00000023F.

[E] - eFuse	0x00000000	0x0000023F	576 Byte	4	0x00	BYTE

Now let's always analyze the eFuse Shadows from the reference manual.

21B_C450	Value of OTP Bank0 Word5 (Configuration and Manufacturing Info.) (OCOTP_CFG4)	32	R/W	0000_0000h	35.5.16/ 2190
21B_C460	Value of OTP Bank0 Word6 (Configuration and Manufacturing Info.) (OCOTP_CFG5)	32	R/W	0000_0000h	35.5.17/ 2190
21B_C470	Value of OTP Bank0 Word7 (Configuration and Manufacturing Info.) (OCOTP_CFG6)	32	R/W	0000_0000h	35.5.18/ 2191
21B_C480	Value of OTP Bank1 Word0 (Memory Related Info.) (OCOTP_MEM0)	32	R/W	0000_0000h	35.5.19/ 2191
21B_C490	Value of OTP Bank1 Word1 (Memory Related Info.) (OCOTP_MEM1)	32	R/W	0000_0000h	35.5.20/ 2192
21B_C4A0	Value of OTP Bank1 Word2 (Memory Related Info.) (OCOTP_MEM2)	32	R/W	0000_0000h	35.5.21/ 2192
21B_C4B0	Value of OTP Bank1 Word3 (Memory Related Info.) (OCOTP_MEM3)	32	R/W	0000_0000h	35.5.22/ 2193

The eFuse Shadow registers have addresses that are always incremented by 0x10 from base address 0x021BC400.

Obviously the eFuse Shadow is a copy of the original eFuse, which is 4 bytes, so in truth only the first 4 bytes of the shadow part really represent the eFuse.

This means that, as you can see in the figure below, the first 4 bytes of the eFuse Shadow are the copy of the original eFuse. The remaining 12 in themselves are not useful.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15
Address 0x21BC470	**eFuse Shadow copy of original eFuse**				Not relevant											

So, for this reason the eFuse Shadow memory map register is the following:

[S] - eFuse Shadow	0x021BC400	0x021BCCFF	2.25 KiB	4	0x00	BYTE

NXP IMX eFUSE Flashing

There are two different methods to program and test eFuse.

Via `#TPCMD EFUSE_WRITE` - `#TPCMD EFUSE_COMPARE` or via commands `#TPCMD PROGRAM` - `#TPCMD VERIFY` commands.
There are important differences between the two methods.

NXP IMX eFUSE Flashing with eFuse Write

Let's start with the first of the two, that is `#TPCMD EFUSE_WRITE` - `#TPCMD EFUSE_COMPARE`:

These two commands are used to program and verify directly the eFuse, so you need to insert the "Reference Manual" eFuse address.
For example, if you want to write the eFuse MEM at address 0x8 you need to use the following commands:

```
#TPCMD EFUSE_WRITE 0x8 <eFuse value>
#TPCMD EFUSE_COMPARE 0x8 <eFuse value>
```

Instead, if you want to use the same commands through eFuse Shadow registers you need to calculate the correct address.
To do that, it's very simple, you need to start from the base address 0x021BC400 an then you need to and then you need to add 0x8 times 0x10.

Address = 0x021BC400 * (0x8 * 0x10) = 0x021BC400 + 0x80 = 0x021BC480

UNIVERSAL PRODUCTION IN-SYSTEM PROGRAMMING

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

21B_C480	Value of OTP Bank1 Word0 (Memory Related Info.) (OCOTP_MEM0)	32	R/W	0000_0000h	35.5.19/ 2191

```
#TPCMD EFUSE_WRITE 0x021BC480 <eFuse value>
#TPCMD EFUSE_COMPARE 0x021BC480 <eFuse value>
```

NXP IMX eFUSE Flashing with eFuse Program

Now we can describe the second method, that is `#TPCMD PROGRAM` - `#TPCMD VERIFY`:

These two commands are used to program and verify indirectly the eFuse, so you need to insert the eFuse memory map address.

For example, if you want to program the eFuse MEM at address 0x8 you need to use and FRB file or the dynamic memory and the following commands:

```
#DYNMEMCLEAR
#DYNMEMSET2 0x20 4 xxyyzzkk
#TPSETSRC DYNMEM

#TPCMD PROGRAM E
#TPCMD VERIFY E R
```

Now let's see in detail how the commands just entered are formed:

0x20 is the address of the eFuse.
It is calculated starting from the physical address 0x8 and multiplying it by 4 bytes (each eFuse is made up of 4 bytes).
So 0x8 * 4 = 0x20.

```
xxyyzzkk
```

```
xx -> 1st eFuse byte
yy -> 2nd eFuse byte
zz -> 3rd eFuse byte
kk -> 4th eFuse byte
```

Instead, if you want to use the same commands through eFuse Shadow registers you need to calculate the correct address.
To do that, it's very simple, you need to start from the base address 0x021BC400 an then you need to and then you need to add 0x8 times 0x10.

Address = 0x021BC400 * (0x8 * 0x10) = 0x021BC400 + 0x80 = 0x021BC480

21B_C480	Value of OTP Bank1 Word0 (Memory Related Info.) (OCOTP_MEM0)	32	R/W	0000_0000h	35.5.19/ 2191

```
#DYNMEMCLEAR
#DYNMEMSET2 0x021BC480 4 xxyyzzkk
#TPSETSRC DYNMEM

#TPCMD PROGRAM E
#TPCMD VERIFY E R
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

NXP IMX Driver Parameters

The standard parameters are used to configure some specific options inside IMX driver.

#TCSETPAR ENTRY_CLOCK

Syntax: `#TCSETPAR ENTRY_CLOCK <Frequency>`

`<Frequency>` Accepted parameters 4000000, 2000000, 1000000, 500000, 100000 Hz

Description: Set the JTAG/SWD frequency used in the Connect procedure before raising the PLL of the device, if the device PLL is available

Note: Default value 4.00 MHz

#TCSETPAR SAMPLING_POINT

Syntax: `#TCSETPAR SAMPLING_POINT <Value>`

`<Value>` Accepted values are in the range 1-15

Description: Use this parameter to permanently set the sampling point of the FPGA
It is recommended to leave this parameter with the default value

Note: Default value 17

#TCSETPAR QSPI_CLOCK

Syntax: `#TCSETPAR QSPI_CLOCK <Value>`

`<Value>` Accepted values are 120, 96, 80, 60, 48, 40, 30, 24, 20, 15, 10, 5MHz (i.e., 25000000)

Description: Set the SPI or QUAD-SPI communication frequency between the IMX and the External Memory

#TCSETPAR QSPI_PROTOCOL

Syntax: `#TCSETPAR QSPI_PROTOCOL <Value>`

`<Value>` Accepted values are SPI, QUAD-SPI or OCTO-SPI

Description: Select SPI, QUAD-SPI or OCTO-SPI communication between the IMX and the External Memory

#TCSETPAR FLEXSPIx_y

Syntax: `#TCSETPAR FLEXSPIx_y <Value>`

`<Value>` Accepted values are in the range 1-15

Description: Select the FlexSPI port of the IMX used. Example: FLEXSPI1_A1, FLEXSPI1_A2, FLEXSPI1_B1, FLEXSPI1_B2, FLEXSPI2_A1, FLEXSPI2_A2, FLEXSPI2_B1, FLEXSPI2_B2

NXP IMX Driver Commands

Here you can find the complete list of all available commands for IMX driver.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

```
E → eFUSE
S → Shadow Registers
F → Flash (internal to the SoC)
X → External SPI Flash
```

#TPCMD CONNECT

#TPCMD CONNECT
This function performs the entry and is the first command to be executed when starting the communication with the device.

```
---#TPCMD CONNECT
Protocol selected SWD.
Entry Clock is 4.00 MHz.
Trying Hot Plug connect procedure.
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 4.00 MHz.
JTAG-SWD Debug Port enabled.
Scanning AP map to find all APs.
AP[0] IDR: 0x84770001, Type: AMBA AHB3 bus.
AP[1] IDR: 0x24770011, Type: AMBA AHB3 bus.
AP[2] IDR: 0x54770002, Type: AMBA APB2 or APB3 bus.
AP[0] ROM table base address 0xE00FD000.
CPUID: 0x411FC272.
Implementer Code: 0x41 - [ARM].
Found Cortex M7 revision r1p2.
Cortex M7 Core halted [0.002 s].
Internal Watchdogs are disabled.
M7 core PLL set to 800 MHz
Cache disabled.
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 4-6].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Time for Connect: 0.126 s.
>|
```

#TPCMD MASSERASE

#TPCMD MASSERASE <F|X>
This function performs a masserase for Main Flash or External Memory.

#TPCMD ERASE

#TPCMD ERASE <F|X>
This function performs a sector erase for the Main Flash or External Memory.

#TPCMD ERASE <F|X> <start address> <size>
This function performs a sector erase for the Main Flash or External Memory.
Enter the Start Address and Size in hexadecimal format.

#TPCMD BLANKCHECK

#TPCMD BLANKCHECK <F|X>
Blankcheck is available for Main Flash and External Memory.
Verify if all memory is erased.

#TPCMD BLANKCHECK <F|X> <start address> <size>
Blankcheck is available for Main Flash and External Memory.
Verify if selected part of memory is erased.
Enter the Start Address and Size in hexadecimal format.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

#TPCMD PROGRAM

#TPCMD PROGRAM <F|E|S|X>
Program available for Main Flash, eFUSE, Shadow Registers and External Memory.
Shadow Registers addressing (S) can be used as alternative to program the EFUSE.
Be aware that the OTP EFUSE are programmed with S memory addressing.
Programs all memory of the selected type based on the data in the FRB file.

#TPCMD PROGRAM <F|E|S|X> <start address> <size>
Program available for Main Flash, eFUSE, Shadow Registers and External Memory.
Shadow Registers addressing (S) can be used as alternative to program the EFUSE.
Be aware that the OTP EFUSE are programmed with S memory addressing.
Programs selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

#TPCMD VERIFY

#TPCMD VERIFY <F|E|S|X> <R>
R: Readout Mode.
Verify Readout available for Main Flash, eFUSE, Shadow Registers and External Memory.
Verify all memory of the selected type based on the data in the FRB file.
Shadow Registers addressing (S) in this case verify the Shadow Registers and not the EFUSE.

#TPCMD VERIFY <F|E|S|X> <R> <start address> <size>
R: Readout Mode.
Verify Readout available for Main Flash, eFUSE, Shadow Registers and External Memory.
Verify selected part of memory of the selected type based on the data in the FRB file.
Shadow Registers addressing (S) in this case verify the Shadow Registers and not the EFUSE.
Enter the Start Address and Size in hexadecimal format.
#TPCMD VERIFY <F|X> <S>
S: Checksum 32 Bit Mode.
Verify Checksum available for Main Flash and External Memory.
Verify all memory of the selected type based on the data in the FRB file.

#TPCMD VERIFY <F|X> <S> <start address> <size>
S: Checksum 32 Bit Mode.
Verify Checksum available for Main Flash and External Memory.
Verify selected part of memory based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

#TPCMD READ

#TPCMD READ <F|E|S|X>
#TPCMD READ <F|E|S|X> <start address> <size>
Read function for Main Flash, eFUSE, Shadow Registers and External Memory.
The result of the read command will be visible into the Terminal.

#TPCMD DUMP

#TPCMD DUMP <F|E|S|P|X>
#TPCMD DUMP <F|E|S|P|X> <start address> <size>
Dump command for Main Flash, eFUSE, Shadow Registers and External Memory.
The result of the dump command will be stored in the FlashRunner 2.0 internal memory.

#TPCMD RUN

Syntax: **#TPCMD RUN** <Time [s]>

<Time [s]>	Time in seconds (i.e., 2 s). This time is an optional parameter.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Prerequisites: none

Description: Move the Reset line up and down quickly if no parameter <Time [s]> is inserted.
#TPCMD RUN <Time [s]> instead moves the Reset line down and high, waits for the entered time.
This command typically can be used to execute the firmware programmed in the device.

#TPCMD EXECUTE

Syntax:
#TPCMD EXECUTE <Program Counter>
#TPCMD EXECUTE <Program Counter> <Wait Time [ms]>
#TPCMD EXECUTE <Program Counter> <Check Address> <Expected Value> <Timeout [ms]>

<Program Counter>	Program Counter in HEX format (i.e., 0x20004000)
<Wait Time [ms]>	Wait time in decimal format (i.e., 2000 ms -> 2 s)
<Check Address>	Check Address in HEX format (i.e., 0x20004000)
<Expected Value>	Expected Value in HEX format (i.e., 0xBADABADA)
<Timeout [ms]>	Timeout in decimal format (i.e., 2000 ms -> 2 s)

Prerequisites: none

Description: This command can be used in three different modes:

#TPCMD EXECUTE <Program Counter>

Set the Program Counter PC at inserted address and try to start the Cortex core.
If the core starts correctly, the command returns PASS and leaves the core in a RUN state.

#TPCMD EXECUTE <Program Counter> <Wait Time [ms]>

Set the Program Counter PC at inserted address and try to start the Cortex core.
If core starts correctly, wait for inserted Time [ms], halt the core before continue with other commands.

#TPCMD EXECUTE <Program Counter> <Check Address> <Expected Value> <Timeout [ms]>

Set the Program Counter PC at inserted address and try to start the Cortex core.
If core starts correctly, wait until Expected Value is present inside Check Address.
If the Timeout value expire before Expected Value is found, the command fails.

Note: This command is available from driver version **5.07**

#TPCMD READ_MEM32

Syntax:
#TPCMD READ_MEM32 <Address> <32-bit Word Count>

<Address>	Address in HEX format (i.e., 0x52002020)
<32-bit Word Count>	32-bit Word count in decimal format (i.e., 2 -> two 32-bit words)

Prerequisites: none

Description: Read memory 32-bit word per 32-bit word from target IMX device

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
---#TPCMD READ_MEM32 0x52002020 2
Read[0x52002020]: 0x1416AAF0
Read[0x52002024]: 0x00000000
Time for Read Mem: 0.002 s
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

SMH Technologies®

#TPCMD DISCONNECT

#TPCMD DISCONNECT

Disconnect function. Power off and exit.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

NXP IMX Driver Examples

Here you can see a complete example of NXP IMX projects.

1 – NXP IMX 16 MB example Commands

IS25WP128F memory with 16MB size through MIMXRT1176xxx8

```
#TCSETPAR FLEXSPIx_y FLEXSPI1_A1
#TCSETPAR PROTCLK 37500000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR QSPI_CLOCK 40000000
#TCSETPAR QSPI_PROTOCOL QUAD-SPI
#TCSETPAR RSTDOWN 100
#TCSETPAR RSTDRV OPENDRAIN
#TCSETPAR RSTUP 100
#TCSETPAR VPROG0 3300
#TCSETPAR CMODE SWD
#TPSETSRC 16MB.frb
#TPSTART
#TPCMD CONNECT
#TPCMD MASSERASE X
#TPCMD BLANKCHECK X
#TPCMD PROGRAM X
#TPCMD VERIFY X R
#TPCMD VERIFY X S
#TPCMD DISCONNECT
#TPEND
```

1 – NXP IMX 16 MB example Real Time Log

```
---#TPCMD CONNECT
Protocol selected SWD.
Entry Clock is 4.00 MHz.
Trying Hot Plug connect procedure.
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 4.00 MHz.
JTAG-SWD Debug Port enabled.
Scanning AP map to find all APs.
AP[0] IDR: 0x84770001, Type: AMBA AHB3 bus.
AP[1] IDR: 0x24770011, Type: AMBA AHB3 bus.
AP[2] IDR: 0x54770002, Type: AMBA APB2 or APB3 bus.
AP[0] ROM table base address 0xE00FD000.
CPUID: 0x411FC272.
Implementer Code: 0x41 - [ARM].
Found Cortex M7 revision r1p2.
Cortex M7 Core halted [0.002 s].
Internal Watchdogs are disabled.
M7 core PLL set to 800 MHz
Cache disabled.
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 4-6].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Time for Connect: 0.126 s.
>|
---#TPCMD MASSERASE X
Initialize FLEXSPI1 peripheral.
 * Initialize QSPI GPIO pins.
 * Initialize QSPI Clock.
 * Initialize FLEXSPI LUT.
> FLEXSPI1 peripheral initialized.
Configure external memory.
 * Check external memory ID Code.
 * Check Status Register.
  ** Status register 0x40.
  ** Clean memory status register.
  ** Status register 0x00.
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

```
 * Check external memory properties.
  ** SFDP table supported.
  ** Flash size check passed: 16MiB.
 * Configure external memory properties.
  ** Switching from SPI to QUAD-SPI protocol.
  ** Status register 0x00.
  ** Clean memory status register.
  ** Status register 0x40.
  ** Set eight dummy cycles.
  ** Use 3-Byte address mode operation.
> External memory configured.
Time for Masserase X: 25.748 s.
>|
---#TPCMD BLANKCHECK X
Time for Blankcheck X: 1.029 s.
>|
---#TPCMD PROGRAM X
Time for Program X: 17.270 s.
>|
---#TPCMD VERIFY X R
Time for Verify Readout X: 6.654 s.
>|
---#TPCMD VERIFY X S
Time for Verify Checksum 32bit X: 0.844 s.
>|
---#TPCMD DISCONNECT
>|
```

1 – NXP IMX 16 MB example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.126 s
Masserase External Flash	25.748 s
Blankcheck External Flash	1.029 s
Program External Flash	17.270 s
Verify Readout External Flash	6.654 s
Verify Checksum External Flash	0.844 s
Cycle Time	**00:51.865 s**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

NXP IMX Driver Changelog

Info about driver versions prior to 4.00
All driver versions prior to 4.00 are to be considered obsolete, please update your driver to the latest version.

Info about driver version 4.00 - 16/07/2022
Supported i.MXRT1064 (with "Internal" Flash) and i.MXRT1170 EFUSE.

Info about driver version 5.00 - 25/08/2022
Added FPGA for new FlashRunner 2.0 models.
Support of MIMXRT1176xxx8_1x_IS25WP128F: external flash through iMX.

Info about driver version 5.01 - 12/12/2022
Added **rstUp** and **rstDown** to reset impulse connect procedure.

Info about driver version 5.02 - 10/11/2023
Internal driver update.

Info about driver version 5.03 - 29/04/2024
Support of MIMXRT1171xxx8_1x_MX25UW6445G: external flash through iMX.

Info about driver version 5.04 - 29/04/2024
Updated Read and Dump method for external memories through iMX device.

Info about driver version 5.05 - 27/06/2024
Support of MIMXRT106xxxxx_1x_IS25xxxxx: external flash through iMX.

Info about driver version 5.06 - 27/09/2024
Supported IMX 6 Ultra Lite series.

Info about driver version 5.07 - 04/03/2025
Added RAM memory [X] for MIMXRT117x devices.
Added commands:

- `#TPCMD PROGRAM X`
- `#TPCMD VERIFY X R`
- `#TPCMD READ X`
- `#TPCMD DUMP X`
- `#TPCMD EXECUTE` <Program Counter> <Check Address> <Expected Value> <Timeout [ms]>

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com