M.2/SerialMemory interface specification

Introduction

SPI serial memories are evolving to higher data rates using Quad-SPI, Octo-SPI, and Hexadeca-SPI interfaces, collectively called XSPI. These interfaces maximize data rate while minimizing pin usage, especially in memory-intensive embedded MCU applications such as those offering graphical displays.

Early in new projects, multisourcing and part selection for both MCUs and memories are critical. To simplify this, a nonsolder solution is preferred for validating hardware and software compatibility. Inspired by Sparkfun Micromod, which reuses the PCIe M.2 NGFF connector, here a similar interface reuses the M.2 Key A connector for XSPI serial memories, named M.2/SerialMemory.

This document provides the M.2/SerialMemory interface specification along with board-level design guidelines for both MCU and memory add-on boards. It aims to facilitate the integration and compatibility of serial memory extensions using the M.2 connector.

STMicroelectronics offers a serial memory pack, consisting of five serial memory boards that can be used with Nucleo-144 boards containing an M.2 Key A serial memory connector. The following boards are compatible with the B-M2MEM-PACK1 M.2 serial memory pack:

- NUCLEO-C5A3ZG
- NUCLEO-H5E5ZJ
- NUCLEO-U3C5ZI-Q

1 M.2 Key A serial memory connector

Figure 1. M.2 Key A serial memory connector schematic

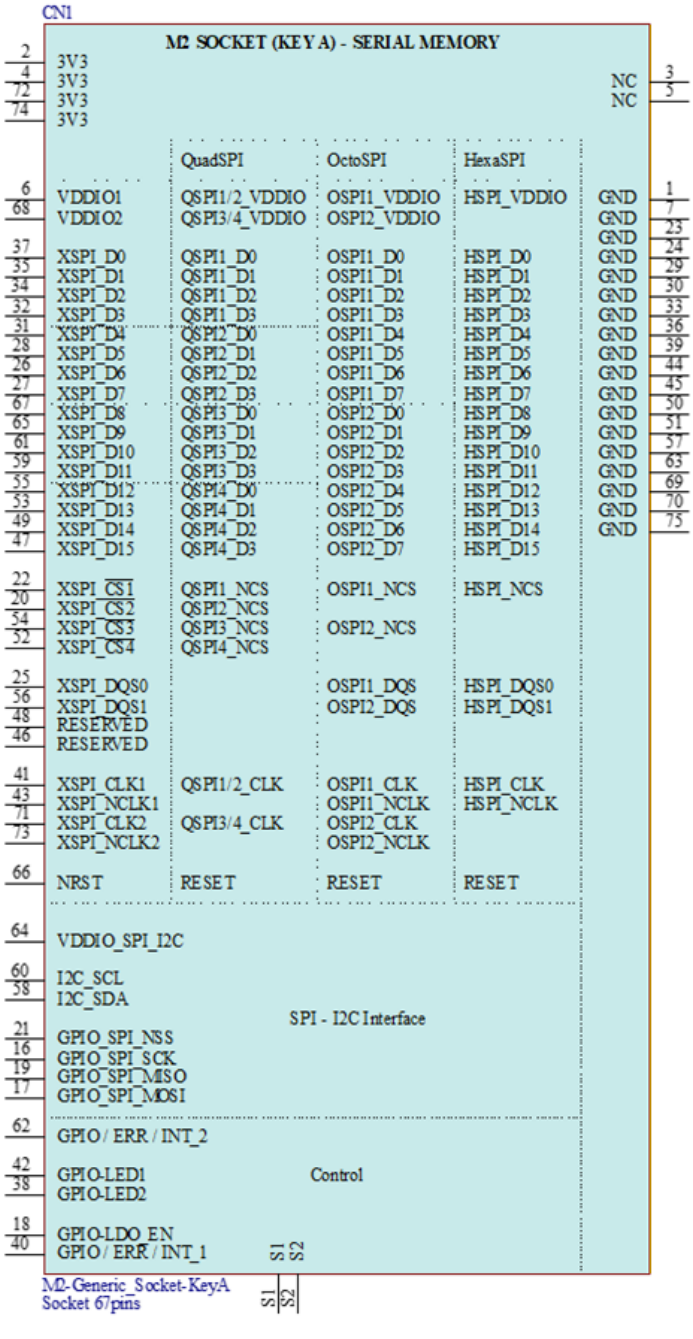


Table 1. **M.2 Key A serial memory connector pinout**

M.2 pin name	M.2 pin number	M.2 pin number	M.2 pin name
GND	1	2	3V3
NC	3	4	3V3
NC	5	6	VDDIO1
GND	7	8	KEY A
KEY A	9	10	KEY A
KEY A	11	12	KEY A
KEY A	13	14	KEY A
KEY A	15	16	GPIO_SPI_SCK
GPIO_SPI_MOSI	17	18	GPIO_LDO_EN
GPIO_SPI_MISO	19	20	XSPI_NCS2
GPIO_SPI_NSS	21	22	XSPI_NCS1
GND	23	24	GND
XSPI_DQS0	25	26	XSPI_D6
XSPI_D7	27	28	XSPI_D5
GND	29	30	GND
XSPI_D4	31	32	XSPI_D3
GND	33	34	XSPI_D2
XSPI_D1	35	36	GND
XSPI_D0	37	38	GPIO-LED2
GND	39	40	GPIO / ERR / INT_1
XSPI_CLK1	41	42	GPIO-LED1
XSPI_NCLK1	43	44	GND
GND	45	46	RESERVED
XSPI_D15	47	48	RESERVED
XSPI_D14	49	50	GND
GND	51	52	XSPI_NCS4
XSPI_D13	53	54	XSPI_NCS3
XSPI_D12	55	56	XSPI_DQS1
GND	57	58	I2C_SDA
XSPI_D11	59	60	I2C_SCL
XSPI_D10	61	62	GPIO / ERR / INT_2
GND	63	64	VDDIO_SPI_I2C
XSPI_D9	65	66	NRST
XSPI_D8	67	68	VDDIO2
GND	69	70	GND
XSPI_CLK2	71	72	3V3
XSPI_NCLK2	73	74	3V3
GND	75	-	

2 M.2/SerialMemory interface implementation

The list of pinout possibilities described here is not exhaustive but serves as the official proposal to ensure compatibility between boards with an M.2 Key A connector and external memory modules.

The M.2 Key A connector pinout can support, for example:

- 1 to 4 Quad-SPI memories
- 1 to 2 Octo-SPI memories
- 1 Hexadeca-SPI memory

Memory configuration support depends on the MCU capabilities embedded on the board.

The connector includes:

- An I^2C/I3C bus interface allows memory modules to include a serial EEPROM. The MCU software reads and configures the serial memory. This feature is mandatory.
- An SPI bus interface (optional)
- Two LED command signals (optional)
- An open-drain *LDO_Enable* MCU-controlled signal (mandatory)

All board design resources, including schematics, EDA databases, manufacturing files, and the bill of materials, are available from the B-M2MEM-PACK1 product page at *www.st.com*.

Section 5 details the power strategy. By default, examples assume a 3.3 V supply to the memory board. The memory board includes an LDO with an Enable control signal to clamp input voltage to the maximum operating voltage of the memory. This LDO control allows users to restart the memory during debugging by power-cycling it properly. Since the LDO acts as a bypass when the input voltage is below the regulated level and its output feeds back to the M.2 connector, various voltage scenarios and large operating voltage memories are supported. This also helps support memories without a reset control signal.

A memory add-on board must use a single-voltage source for all memories.

Table 2 presents the recommended pinout and sequence for connecting XSPI data memory, with the **minimum configuration required to support one memory interface** highlighted in bold.

Table 2. M.2 Key A serial memory connector XSPI data ordering

M.2 pinout	M.2 pin names	1 to 4 Quad-SPI	1 to 2 Octo-SPI	1 Hexadeca-SPI
37	XSPI_D0	**QSPI1-Data 0**	**OSPI1 Data 0**	**HSPI Data 0**
35	XSPI_D1	**QSPI1-Data 1**	**OSPI1 Data 1**	**HSPI Data 1**
34	XSPI_D2	**QSPI1-Data 2**	**OSPI1 Data 2**	**HSPI Data 2**
32	XSPI_D3	**QSPI1-Data 3**	**OSPI1 Data 3**	**HSPI Data 3**
31	XSPI_D4	QSPI2-Data 0	**OSPI1 Data 4**	**HSPI Data 4**
28	XSPI_D5	QSPI2-Data 1	**OSPI1 Data 5**	**HSPI Data 5**
26	XSPI_D6	QSPI2-Data 2	**OSPI1 Data 6**	**HSPI Data 6**
27	XSPI_D7	QSPI2-Data 3	**OSPI1 Data 7**	**HSPI Data 7**
67	XSPI_D8	QSPI3-Data 0	OSPI2 Data 0	**HSPI Data 8**
65	XSPI_D9	QSPI3-Data 1	OSPI2 Data 1	**HSPI Data 9**
61	XSPI_D10	QSPI3-Data 2	OSPI2 Data 2	**HSPI Data 10**
59	XSPI_D11	QSPI3-Data 3	OSPI2 Data 3	**HSPI Data 11**
55	XSPI_D12	QSPI4-Data 0	OSPI2 Data 4	**HSPI Data 12**
53	XSPI_D13	QSPI4-Data 1	OSPI2 Data 5	**HSPI Data 13**
49	XSPI_D14	QSPI4-Data 2	OSPI2 Data 6	**HSPI Data 14**
47	XSPI_D15	QSPI4-Data 3	OSPI2 Data 7	**HSPI Data 15**
22	XSPI_NCS1	**QSPI1_NCS**	**OSPI1_NCS**	**HSPI_NCS**
20	XSPI_NCS2	QSPI2_NCS(1)	-	-

M.2 pinout	M.2 pin names	1 to 4 Quad-SPI	1 to 2 Octo-SPI	1 Hexadeca-SPI
54	XSPI_NCS3	QSPI3_NCS(1)	OSPI2_NCS	-
52	XSPI_NCS4	QSPI4_NCS(1)	-	-
25	XSPI_DQS0	-	**OSPI1 DQS**	**HSPI DQS0**
56	XSPI_DQS1	-	OSPI2 DQS	**HSPI DQS1**
48	RESERVED	-	-	-
46	RESERVED	-	-	-
41	XSPI_CLK1	**QSPI1/2 CLK**	**OSPI1_CLK**	**HSPI_CLK**
43	XSPI_NCLK1	-	OSPI1_NCLK	HSPI_NCLK
71	XSPI_CLK2	QSPI3/4 CLK	OSPI2_CLK	-
73	XSPI_NCLK2	-	OSPI2_NCLK	-
66	NRST	RESET	RESET	RESET

1. *For the chip-select function, it is recommended to use dedicated pins from the XSPI interface. Refer to the target MCU datasheet to determine the number of available chip-select pins.*

2.1 QSPI interface implementation

The section below details how to support one Quad-SPI memory using a QSPI interface. Other configurations can be supported, provided the default I/O strategy is followed.

2.1.1 One QSPI interface with one Quad-SPI memory

This use case describes how to implement a single QSPI interface to support one Quad-SPI memory. It presents the recommended pinout for modules supporting only one Quad-SPI memory.

This configuration is also the minimum and mandatory pinout for any board supporting an XSPI interface with an M.2 Key A serial memory connector.

Figure 2. One QSPI interface with one Quad-SPI memory - board schematic

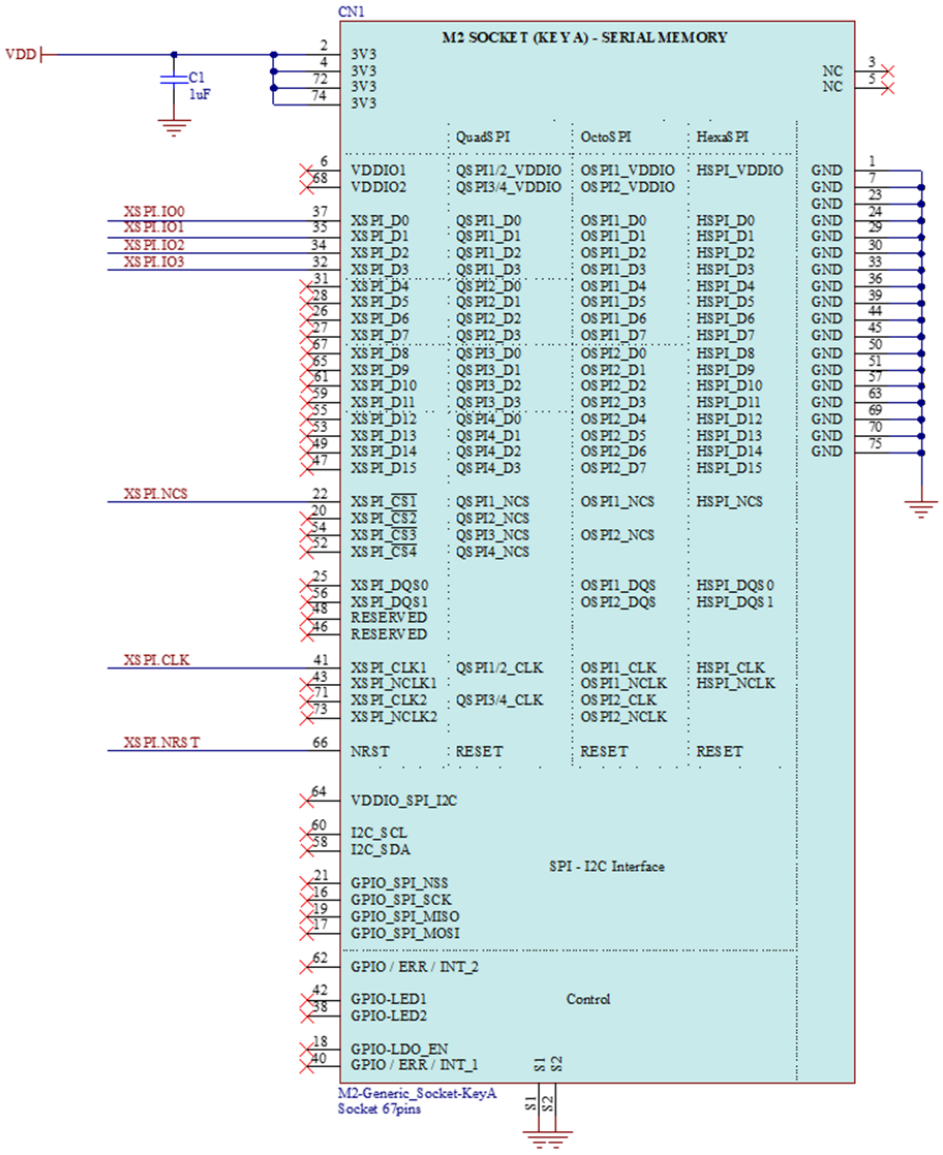


Table 3. One QSPI interface with one Quad-SPI memory - board pinout

M.2 pinout	M.2 pin name	Default implementation
37	XSPI_D0	QSPI Data 0
35	XSPI_D1	QSPI Data 1
34	XSPI_D2	QSPI Data 2
32	XSPI_D3	QSPI Data 3
22	XSPI_NCS1	Chip select for Quad-SPI memory
41	XSPI_CLK1	QSPI CLK
66	NRST	Dedicated GPIO to reset memory
2/4/72/74	3V3	3.3 V memory module power supply from the main board
1/7/23/24/29/30/33/ 36/39/44/45/50/51/ 57/63/69/70/75	GND	-

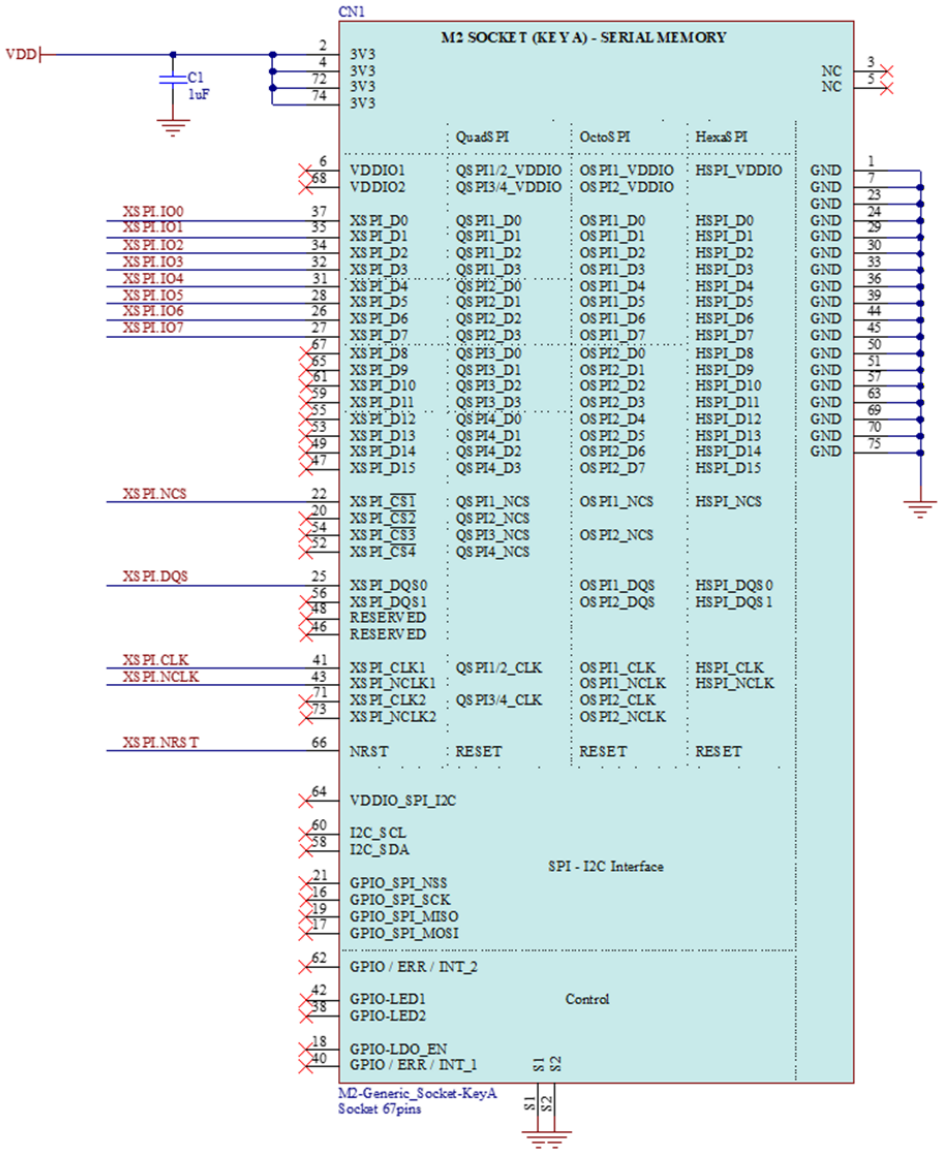
2.2 OSPI interface implementation

The section below details how to support one or two Octo-SPI memories using an OSPI interface. Other configurations can be supported, provided the default I/O strategy is followed.

2.2.1 One OSPI interface with one Octo-SPI memory

This use case describes how to implement a single QSPI interface to support one Octo-SPI memory. It presents the recommended pinout for modules supporting only one Octo-SPI memory.

Figure 3. One OSPI interface with one Octo-SPI memory - board schematic



DT80243V1

Table 4. One OSPI interface with one Octo-SPI memory - board pinout

M.2 pinout	M.2 pin name	Default implementation
37	XSPI_D0	OSPI Data 0
35	XSPI_D1	OSPI Data 1
34	XSPI_D2	OSPI Data 2
32	XSPI_D3	OSPI Data 3
31	XSPI_D4	OSPI Data 4
28	XSPI_D5	OSPI Data 5
26	XSPI_D6	OSPI Data 6
27	XSPI_D7	OSPI Data 7
22	XSPI_NCS1	OSPI_NCS chip select for Octo-SPI memory
25	XSPI_DQS0	Data strobe signal for Octo-SPI memory: OSPI_DQS
41	XSPI_CLK1	OSPI_CLK
43	XSPI_NCLK1	OSPI_NCLK(1)
66	NRST	Dedicated GPIO to reset memory
2/4/72/74	3V3	3.3 V memory module power supply from the main board
1/7/23/24/29/30/33/ 36/39/44/45/50/51/ 57/63/69/70/75	GND	-

1. *The use of NCLK depends on the supported memory features, primarily for HyperBus™, operating at 1.8 V.*

2.2.2 One OSPI interface with two Quad-SPI memories for Octo-SPI emulation

This use case demonstrates the implementation of a single OSPI interface to support two Quad-SPI memories to emulate an Octo-SPI memory.

There is no DQS pin in this configuration because the supported memories are Quad-SPI memories, which do not use a DQS signal.

Figure 4. One OSPI interface with two Quad-SPI memories - board schematic

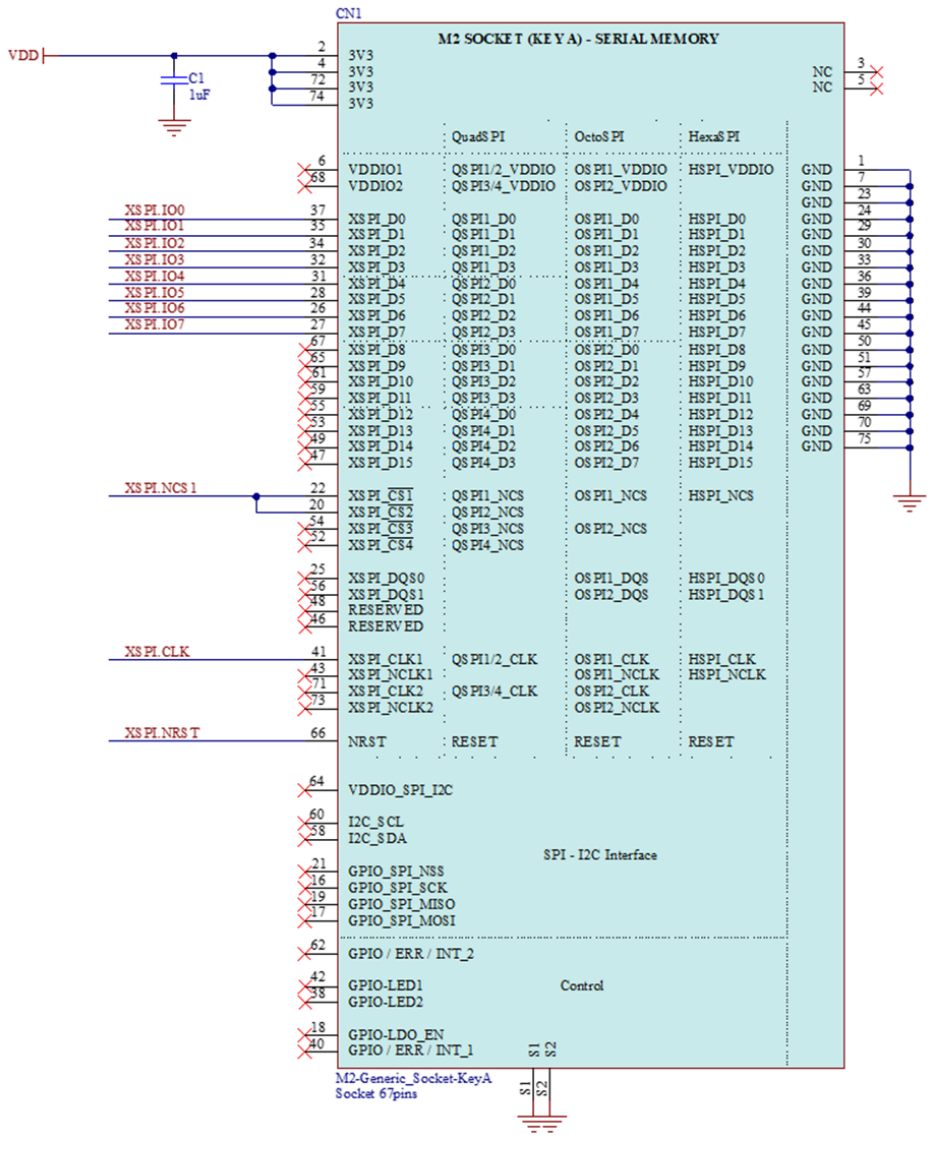


Table 5. One OSPI interface with two Quad-SPI memories - board pinout

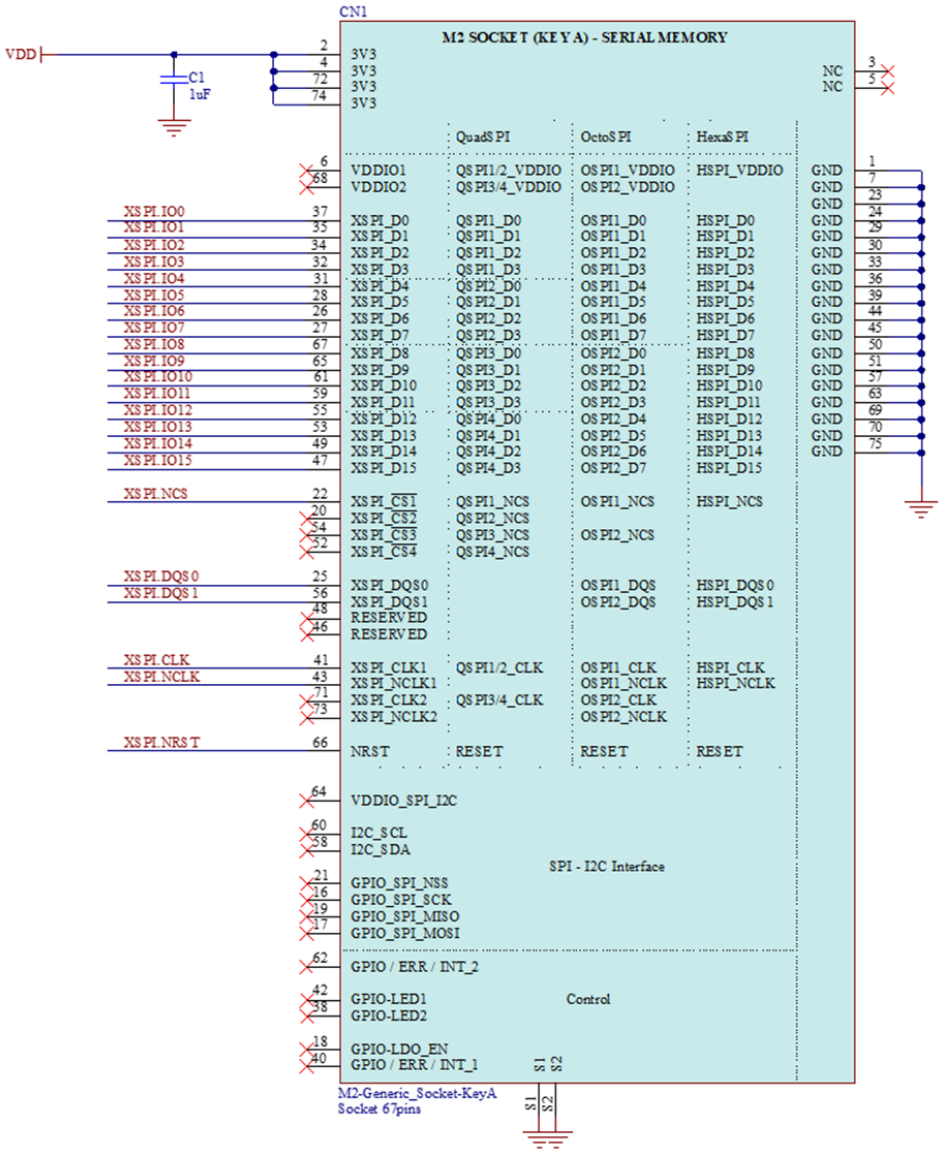
M.2 pinout	M.2 pin name	Default implementation
37	XSPI_D0	QSPI1 Data 0
35	XSPI_D1	QSPI1 Data 1
34	XSPI_D2	QSPI1 Data 2
32	XSPI_D3	QSPI1 Data 3
31	XSPI_D4	QSPI2 Data 0
28	XSPI_D5	QSPI2 Data 1
26	XSPI_D6	QSPI2 Data 2
27	XSPI_D7	QSPI2 Data 3
22	XSPI_NCS1	QSPI1_NCS chip select for memory 1. Both memories are connected to the same NCS.
20		QSPI2_NCS chip select for memory 2. Both memories are connected to the same NCS.
41	XSPI_CLK1	CLK shared with QSPI1_CLK and QSPI2 CLK.
66	NRST	Dedicated GPIO to reset memory.
2/4/72/74	3V3	3.3 V memory module power supply from the main board
1/7/23/24/29/30/33/36/39/44/45/50/51/57/63/69/70/75	GND	-

2.3 HSPI interface implementation

2.3.1 One HSPI interface with one Hexadeca-SPI memory

This use case describes how to implement a single HSPI interface to support one Hexadeca-SPI memory. It presents the recommended pinout for modules supporting only one Hexadeca-SPI memory.

Figure 5. One HSPI interface with one Hexadeca-SPI memory - board schematic



DT80245V1

Table 6. One HSPI interface with one Hexadeca-SPI memory - board pinout

M.2 pinout	M.2 pin names	Default implementation
37	XSPI_D0	HSPI1 Data 0
35	XSPI_D1	HSPI1 Data 1
34	XSPI_D2	HSPI1 Data 2
32	XSPI_D3	HSPI1 Data 3
31	XSPI_D4	HSPI1 Data 4
28	XSPI_D5	HSPI1 Data 5
26	XSPI_D6	HSPI1 Data 6
27	XSPI_D7	HSPI1 Data 7
67	XSPI_D8	HSPI1 Data 8
65	XSPI_D9	HSPI1 Data 9
61	XSPI_D10	HSPI1 Data 10
59	XSPI_D11	HSPI1 Data 11
55	XSPI_D12	HSPI1 Data 12
53	XSPI_D13	HSPI1 Data 13
49	XSPI_D14	HSPI1 Data 14
47	XSPI_D15	HSPI1 Data 15
22	XSPI_NCS1	HSPI_NCS chip select for HSPI memory
25	XSPI_DQS0	Data strobe signal for HSPI_DQS0 memory
56	XSPI_DQS1	Data strobe signal for HSPI_DQS1 memory
41	XSPI_CLK1	HSPI_CLK for HSPI_CLK memory
43	XSPI_NCLK1	HSPI_NCLK(1) for HSPI_CLK memory
66	NRST	Dedicated GPIO to reset memory
2/4/72/74	3V3	3.3 V memory module power supply from the main board
1/7/23/24/29/30/33/ 36/39/44/45/50/51/ 57/63/69/70/75	GND	-

1. *The use of NCLK depends on the supported memory features, primarily for HyperBus™, operating at 1.8 V.*

2.3.2 One HSPI interface with two Octo-SPI memories for Hexadeca-SPI emulation

This use case demonstrates the implementation of a single HSPI interface to support two Octo-SPI memories to emulate an Hexadeca-SPI memory.

Figure 6. One HSPI interface with two Octo-SPI memories - board schematic

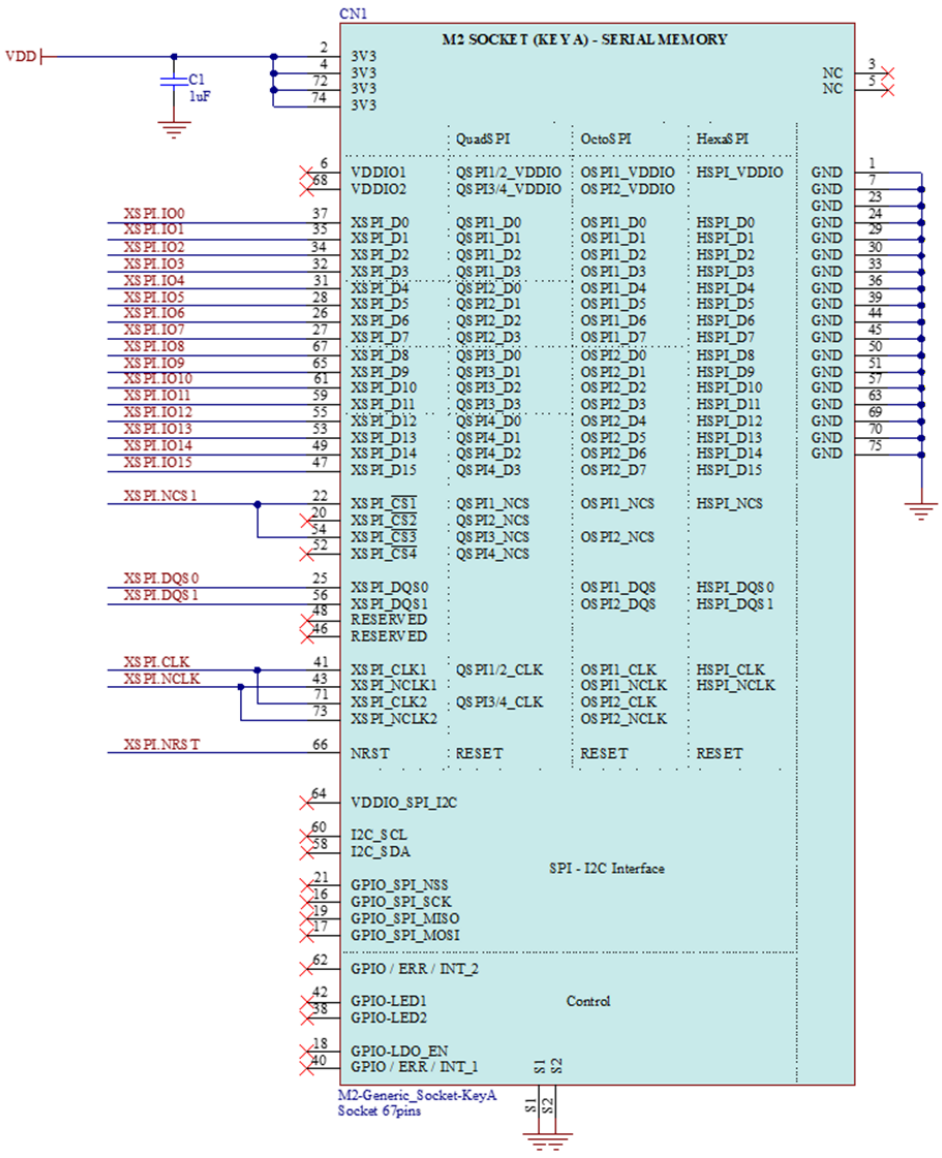


Table 7. One HSPI interface with two Octo-SPI memories - board pinout

M.2 pinout	M.2 pin name	Default implementation
37	XSPI_D0	OSPI1 Data 0
35	XSPI_D1	OSPI1 Data 1
34	XSPI_D2	OSPI1 Data 2
32	XSPI_D3	OSPI1 Data 3
31	XSPI_D4	OSPI1 Data 4
28	XSPI_D5	OSPI1 Data 5
26	XSPI_D6	OSPI1 Data 6
27	XSPI_D7	OSPI1 Data 7
67	XSPI_D8	OSPI2 Data 0
65	XSPI_D9	OSPI2 Data 1
61	XSPI_D10	OSPI2 Data 2
59	XSPI_D11	OSPI2 Data 3
55	XSPI_D12	OSPI2 Data 4
53	XSPI_D13	OSPI2 Data 5
49	XSPI_D14	OSPI2 Data 6
47	XSPI_D15	OSPI2 Data 7
22 54	XSPI_NCS1 XSPI_NCS3	Same processor chip select for memories 1 and 2. Both memories are connected to the same NCS.
25	XSPI_DQS0	Data strobe signal for OSPI_DQS memory 1.
56	XSPI_DQS1	Data strobe signal for OSPI_DQS memory 2.
41 71	XSPI_CLK1 XSPI_CLK2	Same processor CLK shared with OSPI1_CLK and OSPI2 CLK.
43 73	XSPI_NCLK1 XSPI_NCLK2	Same processor NCLK shared with OSPI1_NCLK and OSPI2_NCLK.(1)
66	NRST	Dedicated GPIO to reset memory
2/4/72/74	3V3	3.3 V memory module power supply from the main board
1/7/23/24/29/30/33/ 36/39/44/45/50/51/ 57/63/69/70/75	GND	-

1. *The use of NCLK depends on the supported memory features, primarily for HyperBus™, operating at 1.8 V.*

3 XSPI interface layout guidelines

This section provides guidelines for implementing the XSPI interface layout based on the target maximum frequency. Two requirements must be fulfilled:

- Route signals using single-ended traces with a characteristic impedance of 50 Ω ± 5%.
- Maintain consistent trace lengths for the XSPI buses to prevent timing mismatches.

Table 8 contain links to detailed information on calculations and methodologies for high-speed signal layouts:

Table 8. High-speed XSPI interface layout resources

Link	Author
How to Calculate Trace Length for High-speed Signals(1)	Zuken(1)
PCB Trace Width Conversion Calculator(1)	DigiKey(1)

1. *This URL belongs to a third party. It is active at document publication, however STMicroelectronics shall not be liable for any change, move, or inactivation of the URL or the referenced material.*

To comply with JEDEC specifications, the maximum skew among all signals must be less than 2.5% of the clock period driven by the memory controller. All memory signals are directly or indirectly referenced to this clock.

As an example, the tables below show calculations for 133 MHz and 200 MHz, illustrating the maximum bus length differences allowed to avoid timing mismatches.

Table 9. 133 MHz layout guideline example

Operation frequency	133 MHz	Note
Period	7.5 ns	= 1 / 133 MHz
Skew from length difference	< 187 ps	= 7.5 ns x 2.5% (period x 2.5%)
Length difference	< 28 mm	= 187 ps / 6.7 ps (propagation time: 1mm = 6.7 ps)

Table 10. 200 MHz layout guideline example

Operation frequency	200 MHz	Note
Period	5 ns	= 1 / 200 MHz
Skew from length difference	< 125 ps	= 5 ns x 2.5% (period x 2.5%)
Length difference	< 18.6 mm	= 125 ps / 6.7 ps (propagation time: 1mm = 6.7 ps)

4 I²C, SPI, and spare I/O implementation proposal

This chapter explains the optional connections for spare interfaces such as I²C, SPI, and additional general purpose input/outputs. Two spare pins are available to support optional features like interrupts (INT_1/2), ECC error detection (ERR), and GPIOs.

These interfaces are considered optional and may be shared with embedded STM32 board interfaces like ARDUINO® or ST morpho, depending on I/O multiplexing availability.

A dedicated power pin (pin 64, VDDIO_SPI_I2C) supplies the SPI and I²C interfaces and spare I/Os at the same voltage level as the STM32 interface. I²C devices on the memory board must support a wide operating voltage range (1.8 to 3.3 V), with 3.3 V being mandatory.

Figure 7. I²C, SPI, and spare I/O implementation proposal schematic

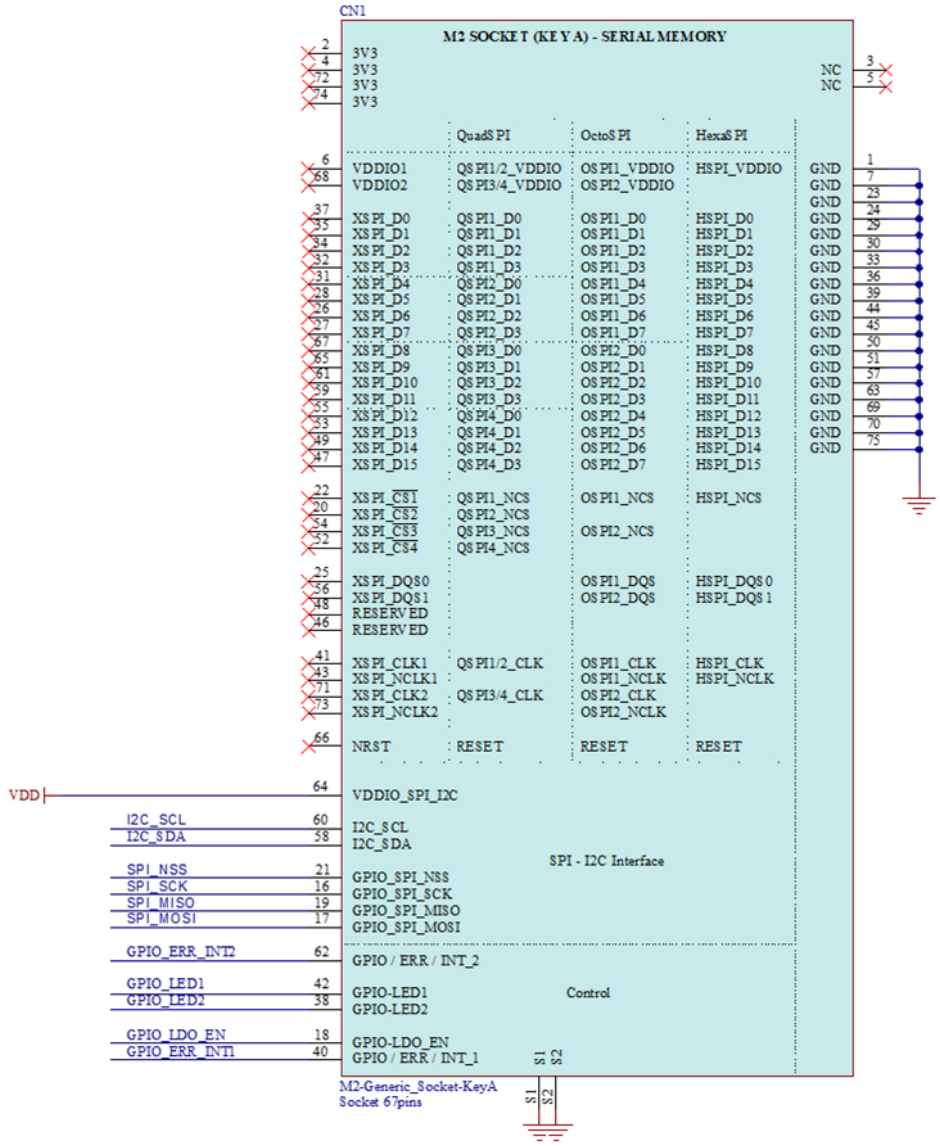


Table 11. I²C, SPI, and spare I/O implementation proposal pinout

M.2 pin number	Pin name	Default implementation
16	GPIO_SPI_SCK	Optional
17	GPIO_SPI_MOSI	Optional
18	GPIO_LDO_EN	Optional
19	GPIO_SPI_MISO	Optional
21	GPIO_SPI_NSS	Optional
38	GPIO_LED2	Optional
40	GPIO/ERR/INT_1	Optional; ERR function is a link to the OSPI 1 memory supported on Data[7:0].
42	GPIO-LED1	Optional
58	I2C_SDA	Optional
60	I2C_SCL	Optional
62	GPIO/ERR/INT_2	Optional; ERR function is a link to the OSPI 2 memory supported on Data[15:8].
64	VDDIO_SPI_I2C	VDD power supply from the STM32 board.
1/7/23/24/29/30/33/ 36/39/44/45/50/51/ 57/63/69/70/75	GND	-

5 M.2 power strategy

This chapter explains the various options for powering the external module. The list of use cases is not exhaustive.

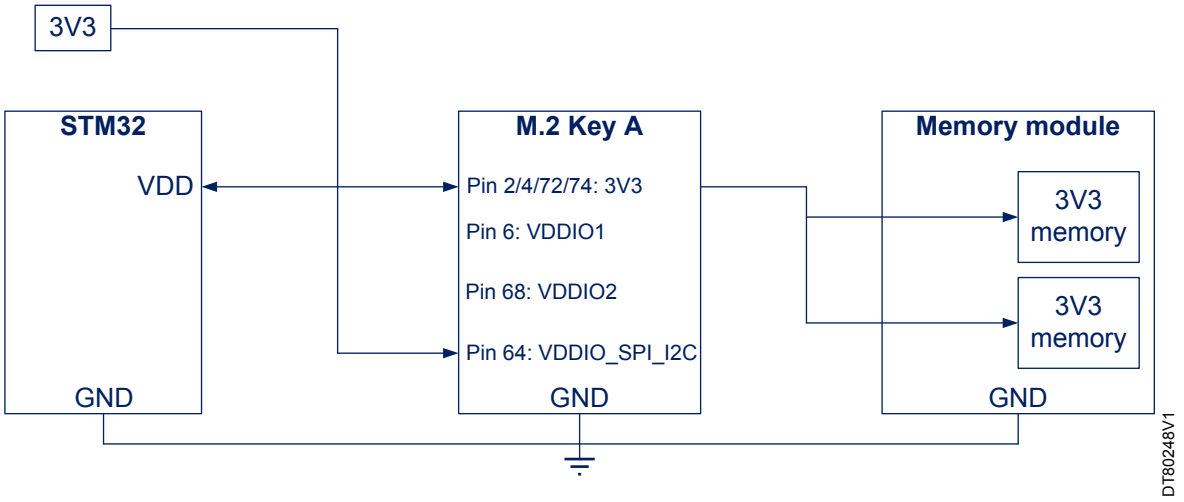
The identified use cases include:

- Use case 1: the STM32 board and the memory module share the same voltage level at 3.3 V.
- Use case 2: the STM32 board provides dedicated power pins for the XSPI interface.
- Use case 3: the STM32 board and the memory module operate at different voltage levels, and the MCU lacks a dedicated XSPI power pin. In this case, the main board offers a jumper for voltage selection.

Use case 1: the STM32 board and the memory module share the same voltage level

In this case, the M.2 power pins (2, 4, 72, and 74) are connected to the STM32 board 3.3 V. The M.2 VDDIO1/2 power pins are not used.

Figure 8. STM32 board and M.2 memory module at 3.3 V

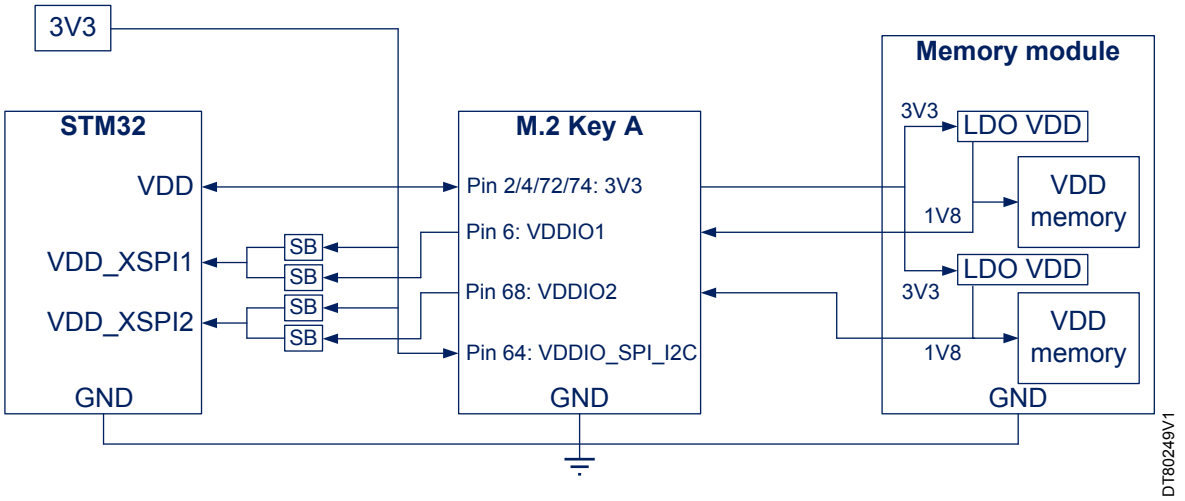


Use case 2: the STM32 board has dedicated power pins for the XSPI interface (VDD_XSPI or STM32_VDDIO2).

The following points describe the power connections and configuration requirements:

- The M.2 power pins 2, 4, 72, and 74 are connected to the 3.3 V supply of the STM32 board.
- VDD_XSPI and STM32_VDDIO2 can be connected either to the 3.3 V of the STM32 board or to a VDDIOx pin to match the memory module voltage level.
- Users must configure the correct solder bridge (SB) on the STM32 board based on the supported memory voltage (3.3 or 1.8 V). The recommended default configuration is 3.3 V to avoid leaving the VDD_XSPI input power pins unconnected.

Figure 9. **STM32 board with dedicated XSPI power domains**

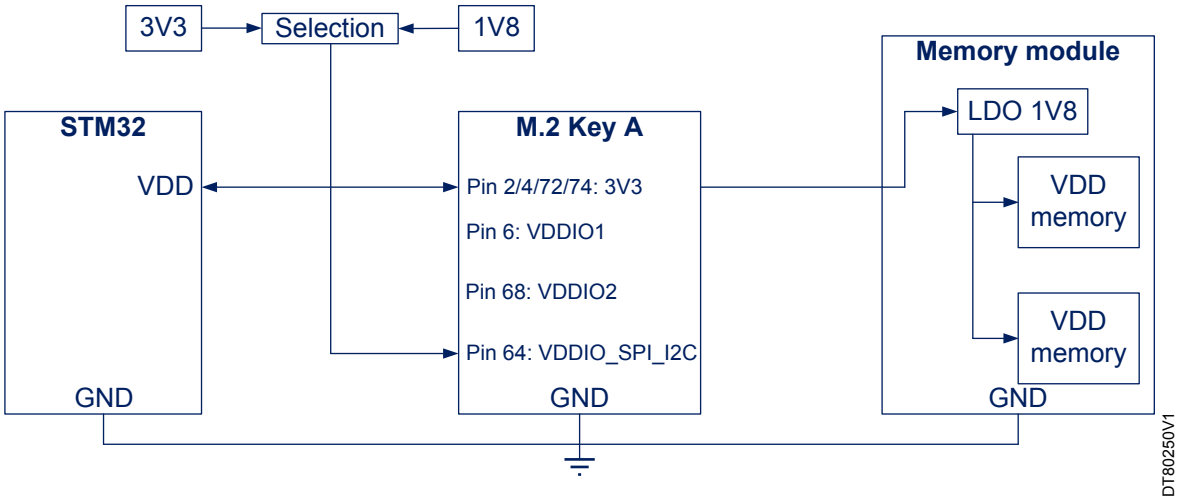


Use case 3: different voltage levels for the STM32 board and memory module and no dedicated XSPI power pin

When the STM32 board and memory module do not share a common voltage level and the STM32 board lacks a dedicated XSPI power input, the following conditions apply:

- The memory module can only be used with STM32 boards that provide a jumper for voltage selection. The user must set the jumper to the correct voltage level.
- The M.2 power pins 2, 4, 72, and 74 are connected to the STM32_VDD_MCU voltage, which is compatible with the memory voltage.
- The M.2 VDDIOx power pins are not used since there is no dedicated XSPI power input on the STM32 board.
- The 3.3 to 1.8 V LDO can be bypassed or used as needed.

Figure 10. **Configuration with a jumper for VDD_MCU voltage selection**



Appendix A MCU board implementation example using STM32 Nucleo-144

STMicroelectronics offers three types of STM32 promotion boards: Nucleo, Discovery, and Eval boards.

Discovery kits mainly demonstrate product performance. **Nucleo boards** expose most MCU pins on ST morpho headers for software development and debugging. They come in three sizes: Nucleo-32 (header pins), Nucleo-64, and Nucleo-144.

The M.2/SerialMemory interface is available on selected Nucleo-144 boards. By default, all M.2/SerialMemory signals are routed to the ST morpho headers, with solder bridges (SB) to disconnect connections for maximum serial memory interface performance.

Power supply rails for M.2/SerialMemory

By default, the memory board receives a 3.3 V supply. Some Nucleo boards feature an MCU_Vdd jumper to select 3.3 or 1.8 V, while others can optionally use MCU_Vddio2 on serial memory pins.

Memory bus support

Most memory add-on boards support four Quad-SPI, two Octo-SPI, or one Hexadeca-SPI memory configuration. Currently, most Nucleo boards currently in design or production support an 8-bit bus. To maximize flexibility, the Nucleo memory serial data bus bits [0:7] and [8:15] are shorted together, allowing the MCU to:

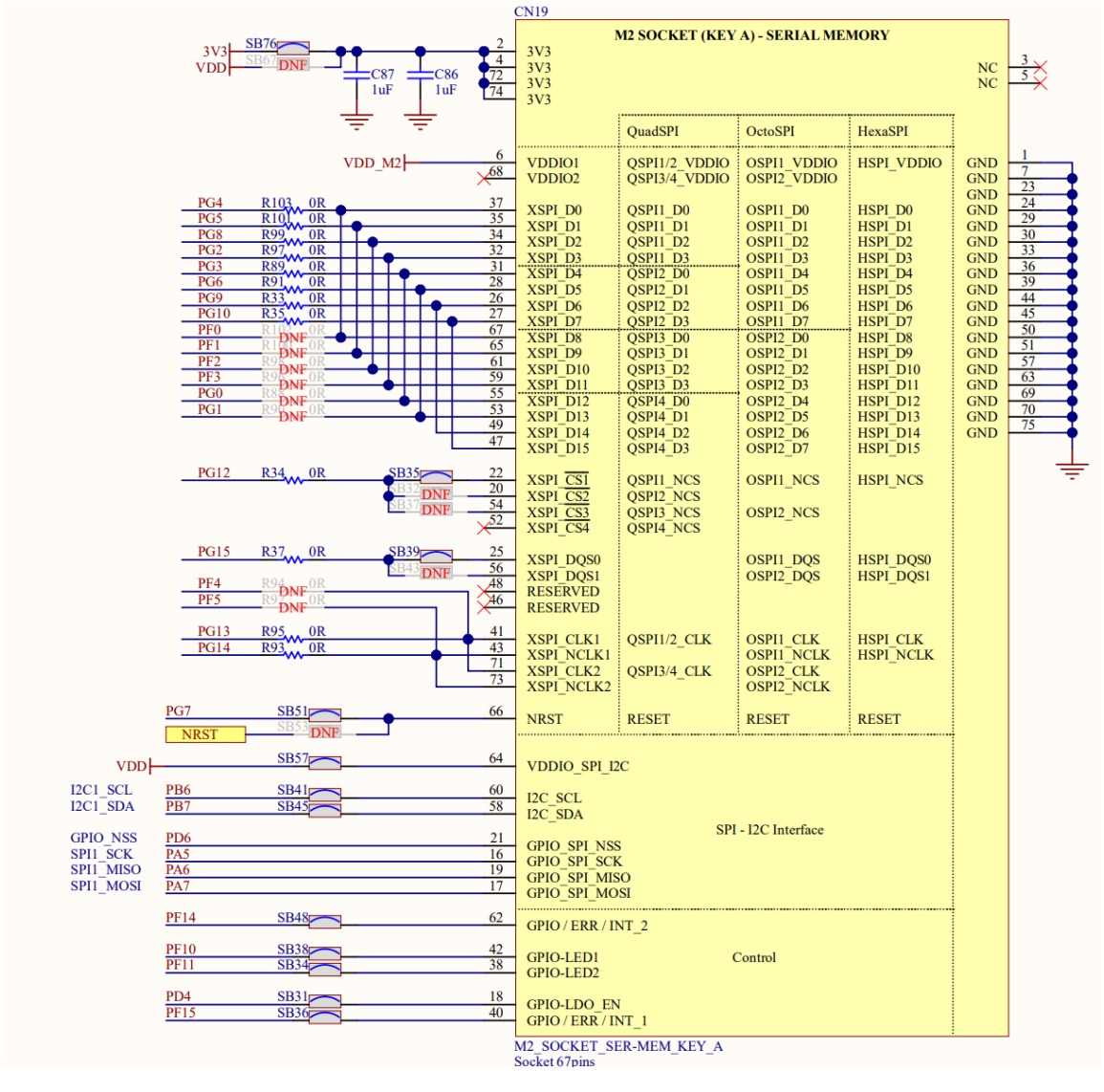
- Drive a 4-bit serial bus to one or two Quad-SPI memories (statically or dynamically selected)
- Drive an 8-bit serial bus to two Quad-SPI memories emulating an Octo-SPI memory
- Drive an 8-bit serial bus to one or two Octo-SPI memories (statically or dynamically selected)

Memory selection based on MCU capability

The method for selecting memory depends on the number and configuration of the MCU chip-select pins, as described below:

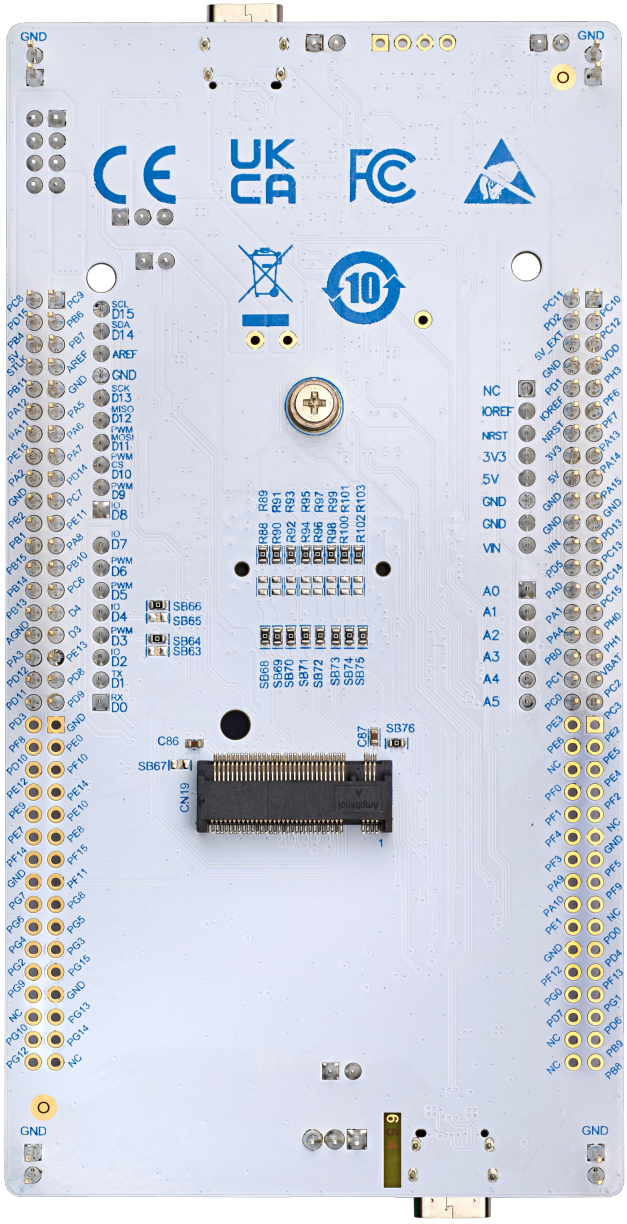
- If only one MCU NCS pin is available, the solder bridge enables static memory selection.
- If the same MCU NCS pin is available on two different MCU pins, static selection is managed by software.
- If the MCU has NCS1 and NCS2 pins, dynamic bus time sharing is enabled.

Figure 11. Example of a Nucleo-144 schematic



In Figure 11, the MCU exposes an 8-bit serial memory bus with a single NCS memory select signal. Zero-ohm serial resistors have been added to the XSPI data path to maximize serial performance, and can be fine-tuned by the user. Solder bridges route the Octo-SPI signals to the unpopulated ST morpho connector, along with the VDDIO1 option.

Figure 12. **NUCLEO-U3C5ZI-Q, supporting M.2/SerialMemory**



Picture is not contractual

Appendix B Memory board implementation example

The reference design for the memory board is a dual-octo configuration using the widely adopted JEDEC BGA24 package.

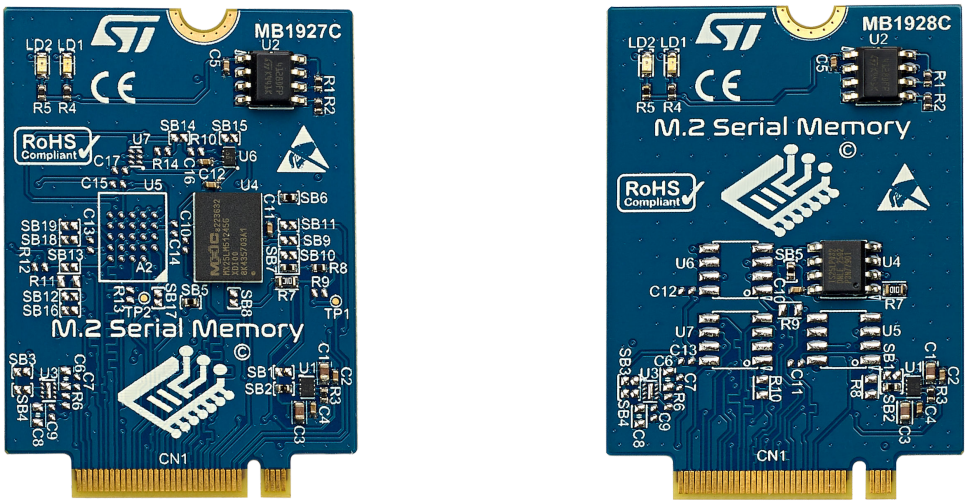
The memory board includes:

- 2 BGA24-compatible memories or 4 SO8L-compatible memories
- Up to 2 identical LDOs with debug bypass solder bridges (LDO footprints are available in various voltages)
- Test points for unused memory BGA balls
- 2 user LEDs (red and green)
- 1 I^2C M24128 EEPROM containing preprogrammed configuration data for MCU software
- M.2 screw location offset from the center to prevent improper mounting on CPU boards

All board design resources, including schematics, EDA databases, manufacturing files, and the bill of materials, are available from the B-M2MEM-PACK1 product page at *www.st.com*.

Figure 13 shows the memory module placements for the different serial memory boards of the B-M2MEM-PACK1 M.2 serial memory pack. MB1927 boards (left) contain two Octo-SPI and MB1928 (right) four Quad-SPI memory module placements.

Figure 13. MB1927 and MB1928 memory module placements



Additional notes

- Voltage level shifting for the NRST pin is assumed to be handled on the add-on board (for BGA packages only), as the NRST voltage may not be compatible with some memories.
- During debugging, either a GPIO or the NRST of the MCU can control the memory NRST signal. Keep in mind that if the MCU software is halted and restarted (for instance, after a breakpoint), the memory must be reset from 8-bit mode back to 1-bit reset mode.
- The INT signal is typically configured as open-drain to alert the MCU and can be used to detect ECC events.

Memory add-on board EEPROM mapping and configuration

Each memory add-on board includes an I²C M24128D identification EEPROM with 16 Kbytes of data storage and, optionally, a 64-byte ID page. Table 12 shows the memory mapping for the 16-Kbyte contiguous memory segment.

Table 12. 16-Kbyte contiguous memory segment memory mapping

Address ange	Size (in Kbytes)	Description	C label
0x0000-1FFF	8	Free to user	-
0x2000-23FF	1	Add-on board information	M2SM_INFO_BASE
0x2400-27FF	1	Memory A info (D0 - D3)	M2SM_MEMA_BASE
0x2800-2BFF	1	Memory B info (D4 - D7)	M2SM_MEMB_BASE
0x2C00-2FFF	1	Memory C info (D8 - D11)	M2SM_MEMC_BASE
0x3000-37FF	1	Memory D info (D12- D15)	M2SM_MEMD_BASE
0x3400-37FF	1	SPI information	M2SM_SPI_BASE
0x3800-3BFF	1	I²C information	M2SM_I2C_BASE
0x3C00-3FFF	1	Reserved	M2SM_RESERVED_BASE

Currently, the code downloads each segment into an MCU internal RAM C structure describing the board and onboard memories. It does not yet include a universal description of memory commands. This feature may be added in future versions of the specification.

The software applies the correct command set for MCU-memory interaction based on the JEDEC ID and memory part number.

Each add-on board memory uses an onboard LDO regulator, which can power down or up the memories through the LDO_EN signal. This method provides software control to reset memories regardless of their current state. It reduces the need for a dedicated reset signal, which is especially useful for memories in small pin-count packages that may lack a reset pin.

Template schematics and C reference source code, particularly for EEPROM memory mapping using a 32-bit Arm® Cortex®-M compiler and linker data structure, are provided to facilitate the development of new memory or MCU boards. This document is open for reuse by any user.

arm

Note: Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries or its affiliates) in the US and/or elsewhere.

Appendix C Board reference storage in M.2 memory board EEPROM

The code snippet below demonstrates at which EEPROM addresses the Board reference information is stored in the M.2 serial memory board.

```
typedef enum {
M2SM_TESTINFO_BASE = 0x0000, // 0x0000 Test Information (number and status) base
M2SM_INFO_BASE = 0x2000, // 0x2000 Board information
M2SM_MEMA_BASE = 0x2400, // 0x2400 Memory A information (data 0..3+)
M2SM_MEMB_BASE = 0x2800, // 0x2800 Memory B information (data 4..7+)
M2SM_MEMC_BASE = 0x2C00, // 0x2A00 Memory C information (data 8..11+)
M2SM_MEMD_BASE = 0x3000, // 0x3000 Memory D information (data 12..15)
M2SM_SPI_BASE = 0x3400,  // 0x3400 SPI devices information
M2SM_I2C_BASE = 0x3800,  // 0x3800 I2C devices information
M2SM_RESERVED_BASE = 0x3C00, // 0x7C00 Reserved
} EEP_Info_Bases_t;
```

The relevation information is stored from here: `M2SM_INFO_BASE = 0x2000, // 0x2000 Board information`. The first half of the EEPROM remains user-configurable.

The data are stored in structures as shown below:

```
typedef struct {
  uint8_t VerRev[2];
  char  CPN[16]; // board order code
  char  FG[16]; // board version code
  uint16_t Vmin_Vin_x100; // incoming power supply acceptable voltage
  uint16_t Vmax_Vin_x100;
  uint16_t Vout_x100; // Vout value if 0 = no regulator, vout = vin
  uint8_t PopulatedMemoryCount; // 0..4 max from D0+ this tells how many valid xspi memory slots
  uint8_t MaxMemoryCount; // 0..4 max how many xspi memories can be populated
  uint8_t JEDEC_MemA[4];
  uint8_t JEDEC_MemB[4];
  uint8_t JEDEC_MemC[4];
  uint8_t JEDEC_MemD[4];
  uint8_t Clock_MHz_Min;
  uint8_t Clock_MHz_Max; // 56 bytes used
  uint32_t Reserved1;
  uint32_t Reserved2;
} M2SM_ID_INFO_t;
```

The board reference is stored after a 2-byte table, so at the third byte (M2SM_INFO_BASE + 2 = 0x2002).

The software is coded as follows to fill the EEPROM data:

```
CopyFromToByteSize (tM2SM_INFO[MemoryType], EEP_Data1kbyte, sizeof(M2SM_ID_INFO_t));
  M2SM_EEP_Write1kbyte(pEEP,M2SM_INFO_BASE);
```

Where:

- `tM2SM_INFO` is a table with the instantiation of each memory detail.
- `M2SM_ID_INFO_t` is its type.
- `EEP_Data1kbyte` is a temporary buffer.
- `pEEP` is a structure containing the EEPROM IDs.
- `M2SM_INFO_BASE` is the information base address.

The code block below shows the instantiations:

```
#define CPN_WB_3V3_STRING "MB1928-33LB-C01"
#define CPN_MX_3V3_STRING "MB1927-33BA-C01"
#define CPN_MX_1V8_STRING "MB1927-18BA-C01"
#define CPN_ST_1V8_STRING "MB1928-18LA-C01"
#define CPN_IS_3V3_STRING "MB1928-33LA-C01"

M2SM_ID_INFO_t M2SM_INFO_W25Q16JV =  {
  .VerRev = { 0x01, 0x00 },
  .CPN = CPN_WB_3V3_STRING, // board order code WB 3V3 SO 1 populated
  .FG =  "V0.1", // board version code
  .Vmin_Vin_x100 = 270, // incoming power supply acceptable voltage
  .Vmax_Vin_x100 = 360,
  .PopulatedMemoryCount = 1, // 0..4 max from D0+ this tells how many valid xspi memory slots
  .MaxMemoryCount = 2, // 0..4 max how many xspi memories can be populated
  .JEDEC_MemA =   {0xEF, 0x70, 0x15, 0x00 }, // this correspond to D0..D3 The last byte meaning is TBD
  .JEDEC_MemB = { 0, 0, 0, 0 }, // this correspond to D4..D7
  .JEDEC_MemC = { 0, 0, 0, 0 }, // this correspond to D8..D11
  .JEDEC_MemD = { 0, 0, 0, 0 }, // this correspond to D12..D15
  .Clock_MHz_Min = 25,
  .Clock_MHz_Max = 133,
  .Reserved1 = 0,
  .Reserved2 = 0,
  // we will fill later the rest
} ;

M2SM_ID_INFO_t M2SM_INFO_MX25LM51245G =  {
  .VerRev = { 0x01, 0x00 },
  .CPN = CPN_MX_3V3_STRING, // board order code MX25 1V8 BGA 1 populated
  .FG =  "V0.1", // board version code
  .Vmin_Vin_x100 = 270, // incoming power supply acceptable voltage
  .Vmax_Vin_x100 = 360,
  .PopulatedMemoryCount = 1, // 0..4 max from D0+ this tells how many valid xspi memory slots
  .MaxMemoryCount = 2, // 0..4 max how many xspi memories can be populated
  .JEDEC_MemA =  {0xC2, 0x85, 0x3A, 0x00 }, // this correspond to D0..D3 The last byte meaning is TBD
  .JEDEC_MemB = { 0, 0, 0, 0 }, // this correspond to D4..D7
  .JEDEC_MemC = { 0, 0, 0, 0 }, // this correspond to D8..D11
  .JEDEC_MemD = { 0, 0, 0, 0 }, // this correspond to D12..D15
  .Clock_MHz_Min = 25,
  .Clock_MHz_Max = 133,
  .Reserved1 = 0,
  .Reserved2 = 0,
  // we will fill later the rest
} ;

M2SM_ID_INFO_t M2SM_INFO_MX25UM51245G =  {
  .VerRev = { 0x01, 0x00 },
  .CPN = CPN_MX_1V8_STRING, // board order code MX25 3V3 BGA 1 populated
  .FG =  "V0.1", // board version code
  .Vmin_Vin_x100 = 270, // incoming power supply acceptable voltage
  .Vmax_Vin_x100 = 360,
  .PopulatedMemoryCount = 1, // 0..4 max from D0+ this tells how many valid xspi memory slots
  .MaxMemoryCount = 2, // 0..4 max how many xspi memories can be populated
  .JEDEC_MemA =  {0xC2, 0x85, 0x3A, 0x00 }, // this correspond to D0..D3 The last byte meaning is TBD
  .JEDEC_MemB = { 0, 0, 0, 0 }, // this correspond to D4..D7
  .JEDEC_MemC = { 0, 0, 0, 0 }, // this correspond to D8..D11
  .JEDEC_MemD = { 0, 0, 0, 0 }, // this correspond to D12..D15
  .Clock_MHz_Min = 25,
  .Clock_MHz_Max = 133,
  .Reserved1 = 0,
  .Reserved2 = 0,
  // we will fill later the rest
} ;

M2SM_ID_INFO_t M2SM_INFO_M95P32 =  {
  .VerRev = { 0x01, 0x00 },
  .CPN = CPN_ST_1V8_STRING, // board order code ST 1V8 1 populated
  .FG =  "V0.1", // board version code
  .Vmin_Vin_x100 = 270, // incoming power supply acceptable voltage
  .Vmax_Vin_x100 = 360,
  .PopulatedMemoryCount = 1, // 0..4 max from D0+ this tells how many valid xspi memory slots
  .MaxMemoryCount = 2, // 0..4 max how many xspi memories can be populated
  .JEDEC_MemA =  {0x00, 0x00, 0x00, 0x00 }, // this correspond to D0..D3 The last byte meaning is TBD
  .JEDEC_MemB = { 0, 0, 0, 0 }, // this correspond to D4..D7
  .JEDEC_MemC = { 0, 0, 0, 0 }, // this correspond to D8..D11
  .JEDEC_MemD = { 0, 0, 0, 0 }, // this correspond to D12..D15
  .Clock_MHz_Min = 25,
```

```
  .Clock_MHz_Max = 133,
  .Reserved1 = 0,
  .Reserved2 = 0,

} ;

M2SM_ID_INFO_t M2SM_INFO_IS25LP032D =  {
  .VerRev = { 0x01, 0x00 },
  .CPN = CPN_IS_3V3_STRING, // board order IS 3V3 BGA 1 populated
  .FG =  "V0.1", // board version code
  .Vmin_Vin_x100 = 270, // incoming power supply acceptable voltage
  .Vmax_Vin_x100 = 360,
  .PopulatedMemoryCount = 1, // 0..4 max from D0+ this tells how many valid xspi memory slots
  .MaxMemoryCount = 2, // 0..4 max how many xspi memories can be populated
  .JEDEC_MemA =  {0x00, 0x00, 0x00, 0x00 }, // this correspond to D0..D3 The last byte meaning is TBD
  .JEDEC_MemB = { 0, 0, 0, 0 }, // this correspond to D4..D7
  .JEDEC_MemC = { 0, 0, 0, 0 }, // this correspond to D8..D11
  .JEDEC_MemD = { 0, 0, 0, 0 }, // this correspond to D12..D15
  .Clock_MHz_Min = 25,
  .Clock_MHz_Max = 133,
  .Reserved1 = 0,
  .Reserved2 = 0,
} ;

int8_t *tCPN[] =             {CPN_WB_3V3_STRING,   CPN_MX_3V3_STRING,       CPN_MX_1V8_STRING,       CP
N_ST_1V8_STRING, CPN_IS_3V3_STRING};
M2SM_ID_INFO_t *tM2SM_INFO[]={&M2SM_INFO_W25Q16JV, &M2SM_INFO_MX25LM51245G, &M2SM_INFO_MX25UM51245G, &M
2SM_INFO_M95P32, &M2SM_INFO_IS25LP032D};
#define NUMBER_OF_MEMORY_SUPPORTED sizeof(tCPN)/sizeof(*tCPN)
```

Revision history

Table 13. **Document revision history**

Date	Revision	Changes
13-Jan-2026	1	Initial release.

Contents

List of tables

List of figures