

Introduction

This application note provides design guidelines and recommendations for designs using the EMxxLX and EMxxLXB family of **PERSYST** Industrial STT-MRAM.

Contents

Introduction	1
1. Pin Connection Guide	2
1.1 BGA (24 ball) package	2
1.2 DFN (5x6 & 6x8) package	3
2. Device Initialization	4
2.1 Example: Using single SPI mode to initialize EMxxLX MRAM. (64Mb or lower density)	5
2.2 Example: Using single SPI mode to initialize 128Mb EM128LX MRAM.	7
2.3 CS deselect time requirement.....	9
2.4 JEDEC reset (JESD252.01)	9
3. Printed Circuit Board (PCB) Footprint	10
3.1 Recommended PCB footprint for 24 ball BGA package	10
3.2 Recommended PCB footprint for 6x8mm DFN package	11
3.3 Recommended PCB footprint for 5x6mm DFN package	12
Revision History	13

1. Pin Connection Guide

1.1 BGA (24 ball) package

Pin #	Pin Name	Pin Connection Recommendation	Note
A1	No Ball		
A2	NC	No connection	User can connect to any node since the input is not connected within the device.
A3	RFU	Don't Connect, Leave Floating	Reserved for Future Use
A4	RESET#	Pull-up resistor $\sim 10K\Omega$	RESET# should track the rise of Vdd. A pull-up resistor can provide this behavior.
A5	INT#	Pull-up resistor $\sim 10K\Omega$	INT# is open-drain, no internal pull-up. If INT# is not used, no pull-up needed.
B1	RFU	Don't Connect, Leave Floating	Reserved for Future Use
B2	CK	Clock	
B3	Vss	GND	
B4	Vdd	1.8V	
B5	RFU	Don't Connect, Leave Floating	Reserved for Future Use
C1	Vss	GND	
C2	CS#	Pull-up resistor $\sim 10K\Omega$	CS# should track the rise of Vdd. A pull-up resistor can provide this behavior.
C3	DS	Data Strobe	
C4	IO2/WP#	Pull-up resistor $\sim 10K\Omega$	It is required that there be a way to make WP# high. Either drive or pull-up
C5	RFU	Don't Connect, Leave Floating	Reserved for Future Use
D1	Vdd	1.8V	
D2	IO1	IO1 / SO	SPI mode: SO / DSPI, QSPI, OSPI: IO1
D3	IO0	IO0 / SI	SPI mode: SI / DSPI, QSPI, OSPI: IO0
D4	IO3	IO3	
D5	IO4	IO4	
E1	IO7	IO7	
E2	IO6	IO6	
E3	IO5	IO5	
E4	Vdd	1.8V	
E5	Vss	GND	

1.2 DFN (5x6 & 6x8) package

Pin #	Pin Name	Pin Connection requirement	Note
1	CS#	Pull-up resistor $\approx 10K\Omega$	CS# should track the rise of Vdd. A pull-up resistor can provide this behavior.
2	IO1	IO1 / SO	SPI mode: SO / DSPI, QSPI: IO1
3	IO2/WP#	Pull-up resistor $\approx 10K\Omega$	SPI mode: WP# / DSPI, QSPI: IO2 It is required that there be a way to make WP# high. Either drive or pull-up
4	Vss	GND	
5	IO0	IO0 / SI	SPI mode: SI / DSPI, QSPI: IO0
6	CK	Clock	
7	IO3/RESET#	Pull-up resistor $\approx 10K\Omega$	SPI, DSPI mode: RESET# / QSPI: IO3 RESET# should track the rise of Vdd. A pull-up resistor can provide this behavior.
8	Vdd	1.8V	
X	Thermal Pad	GND or Floating	

2. Device Initialization

Full device initialization is required if an EMxxLX MRAM is exposed to:

- temperatures greater than 125°C for one hour or more.
- strong magnetic fields greater than HMax = 350 Oe.

For complete information refer to the following application notes on www.everspin.com:

- EST3000 Device Initialization, Reset and Recovery EMxxLX
- EST3004 Magnetic Immunity of EMxxLX.

During board assembly or repair reflow, an EMxxLX MRAM will be exposed to temperatures of 260 °C, possibly resulting in undefined status for some data or configuration values.

Device initialization and configuration is required after such high temperature reflow exposure.

Initialization overview.

- Identify SPI protocol in use by the device or force the protocol to SPI x1, to ensure command reception.
- Make WP# high, send write enable command.
- Enter Device Factory Initialization Mode (DFIM)
- All the nonvolatile configuration bits must be written/programmed. (to “0” then to “1”)
- The OTP area must be written/programmed (to “1” then to “0” – excluding the OTP lock byte).
- All the memory cells must be written/programmed/erased. (to “0”, to “1”, to “0” then to “1”)
 - Note, erase commands may be configured to erase to “0” or to “1” and may be used to have the EMxxLX MRAM do all the work of writing all memory cells to “0” or “1” as one command operation, for each initialize write/program step in the initialization.

Initialization sequence summary.

1. Identify the device SPI protocol or provide JEDEC JESD252 reset to force SPI x1 1S-1S-1S protocol.
2. Set Write Enable. (WREN command) and make WP# high.
3. Set desired SPI protocol for device initialization and clear interrupt status.
4. **Enter DFIM mode. (Device Factory Initialization Mode)**
5. Initialize Status Register to clear block protect bits. (write FFh, write 00h, write FFh, write 00h)
6. Initialize all Nonvolatile Configuration Registers (NCR). (write registers to “00h”, to “FFh”, to “00h”, to “FFh”).
7. Initialize OTP. (unlock OTP, write OTP byte 0~256 to “FFh, to “00h, to “FFh, to “00h)
8. Initialize all MRAM cells. (erase/write all MRAM cells to “0”)
9. Initialize all MRAM cells. (erase/write all MRAM cells to “1”)
10. Initialize all MRAM cells. (erase/write all MRAM cells to “0”)
11. Initialize all MRAM cells. (erase/write all MRAM cells to “1”)
12. **Exit DFIM mode.**
13. Write all Nonvolatile Configuration Registers to their desired state, ready for the next reset or power on.

In the following example sections, for simplicity:

- JEDEC JESD252 reset is used to ensure that the EMxxLX MRAM device is forced to the SPI x1 1S-1S-1S protocol, so that the device is in a known protocol.
- Initialization / Recovery is not performance critical, so time outs are used for each step rather than status polling or interrupts, to know when to proceed to the next step
- Read verify is not done

2.1 Example: Using single SPI mode to initialize EMxxLX MRAM. (64Mb or lower density)

(64/32/16/8/4Mb devices. For 128Mb, see section 2.2)

		Set Host SPI mode to single SPI										
1		JEDEC reset										Force MRAM in single SPI mode. Please check 2.4 JEDEC reset.
2		Write Enable										
	SPI write	CS↓	0x06	CS↑(60ns)								WREN
3		Set IO Mode for Device Initialization : write volatile configuration										
3.1	SPI write	CS↓	0x81	0x00	0x00	0x00	0xFF	0x04	0xFF	0xFF	0xFF	Write VCR 0~8(SPI with DS mode, erase to "0")
			0xFF	0xFF	0xFF	0xFF	CS↑(1.5us)					
3.2	SPI write	CS↓	0x81	0x00	0x00	0x0F	0x00	CS↑(60ns)				Write Int. mask reg: mask interrupt
3.3	SPI write	CS↓	0x81	0x00	0x00	0x10	0x07	CS↑(60ns)				Write Int. status reg: clear
4		Enter DFIM mode										
	SPI write	CS↓	0x81	0x00	0x00	0x1E	0x6B	CS↑(60ns)				Write 0x6B to VCR 1E: enter DFIM
5		Initialize Status Register and clear Block Protect Bits										
5.1	SPI write	CS↓	0x01	0xFF	CS↑(1.5us)							Write 0xFF to status register
5.2	SPI write	CS↓	0x01	0x00	CS↑(1.5us)							Write 0x00 to status register
5.3	SPI write	CS↓	0x01	0xFF	CS↑(1.5us)							Write 0xFF to status register
5.4	SPI write	CS↓	0x01	0x00	CS↑(1.5us)							Write 0x00 to status register, clear block protect bits
6		Initialize Nonvolatile Configuration Registers										
6.1	SPI write	CS↓	0xB1	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	Write NVCR 0~12
			0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CS↑(1.5us)	
6.2	SPI write	CS↓	0xB1	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	Write NVCR 0~12
			0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	CS↑(1.5us)	
6.3	SPI write	CS↓	0xB1	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	Write NVCR 0~12
			0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	CS↑(1.5us)	
6.4	SPI write	CS↓	0xB1	0x00	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	Write NVCR 0~12
			0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	CS↑(1.5us)	
7		Initialize OTP area										
7.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0xF9	CS↑(60ns)				Write VCR 8: Disable OTP lock
7.2	SPI write	CS↓	0x42	0x00	0x00	0x00	0xFF	(0xFF)	0xFF	CS↑(1.5us)		Write OTP byte 0~256 to "1" (257 bytes)
	Byte count		1	2	3	4	5	6~260	261			
7.3	SPI write	CS↓	0x81	0x00	0x00	0x08	0xF9	CS↑(60ns)				Write VCR 8: Disable OTP lock
7.4	SPI write	CS↓	0x42	0x00	0x00	0x00	0x00	(0x00)	0x00	CS↑(1.5us)		Write OTP byte 0~256 to "0" (256 bytes)
	Byte count		1	2	3	4	5	6~260	260			
7.5	SPI write	CS↓	0x81	0x00	0x00	0x08	0xF9	CS↑(60ns)				Write VCR 8: Disable OTP lock

7.6	SPI write	CS↓	0x42	0x00	0x00	0x00	0xFF	(0xFF)	0xFF	CS↑(1.5us)	Write OTP byte 0~256 to "1" (257 bytes)
		Byte count	1	2	3	4	5	6~260	261		
7.7	SPI write	CS↓	0x81	0x00	0x00	0x08	0xF9			CS↑(60ns)	Write VCR 8: Disable OTP lock
7.8	SPI write	CS↓	0x42	0x00	0x00	0x00	0x00	(0x00)	0x00	CS↑(1.5us)	Write OTP byte 0~256 to "0" (256 bytes)
		Byte count	1	2	3	4	5	6~260	260		
8	Bulk Erase (initialize MRAM cells, erase to "0")										
8.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0x7F			CS↑(60ns)	Write VCR 8: set erase to "0"
8.2	SPI write	CS↓	0xC7							CS↑(125ms)	Erase all MRAM cells to "0"
9	Bulk Erase (initialize MRAM cells, erase to "1")										
9.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0xFF			CS↑(60ns)	Write VCR 8: set erase to "1"
9.2	SPI write	CS↓	0xC7							CS↑(125ms)	Erase all MRAM cells to "1"
10	Bulk Erase (initialize MRAM cells, erase to "0")										
10.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0x7F			CS↑(60ns)	Write VCR 8: set erase to "0"
10.2	SPI write	CS↓	0xC7							CS↑(125ms)	Erase all MRAM cells to "0"
11	Bulk Erase (initialize MRAM cells, erase to "1")										
11.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0xFF			CS↑(60ns)	Write VCR 8: set erase to "1"
11.2	SPI write	CS↓	0xC7							CS↑(125ms)	Erase all MRAM cells to "1"
12	Exit DFIM mode										
	SPI write	CS↓	0x81	0x00	0x00	0x1E	0x00			CS↑(60ns)	Write 0x00 to VCR 0x1E: exit DFIM
13	Configure device for next power up. Write NVCRs.										
13.1	SPI write	CS↓	0xB1	0x00	0x00	0x00	0x			CS↑(1.5us)	Write NCR 0: IO mode, refer to datasheet section 5.5
13.2	SPI write	CS↓	0xB1	0x00	0x00	0x01	0x			CS↑(1.5us)	Write NVCR 1: read dummy cycle, refer to datasheet section 5.10
13.3	SPI write	CS↓	0xB1	0x00	0x00	0x04	0x03			CS↑(1.5us)	Write NVCR 4: DS delay 1.2ns (recommend setting)

2.2 Example: Using single SPI mode to initialize 128Mb EM128LX MRAM.

Note that 128Mb devices are constructed from two stacked dice and die select commands are needed to direct some commands to each die. Configuration register initialization is automatically written to both dice.

		Set Host SPI mode to single SPI										
1		JEDEC reset										Force MRAM in single SPI mode. Please check 2.4 JEDEC reset.
2		Write Enable										
	SPI write	CS↓	0x06	CS↑(60ns)							WREN	
3		Set IO Mode for Device Initialization : write volatile configuration										
3.1	SPI write	CS↓	0x81	0x00	0x00	0x00	0xFF	0x04	0xFF	0xFF	0xFF	Write VCR 0~8(SPI with DS mode, erase to "0")
				0xFF	0xFF	0xFF	0xFF	CS↑(1.5us)				
3.2	SPI write	CS↓	0x81	0x00	0x00	0x0F	0x00	CS↑(60ns)				Write Int. mask reg: mask interrupt
3.3	SPI write	CS↓	0x81	0x00	0x00	0x10	0x07	CS↑(60ns)				Write Int. status reg: clear
4		Enter DFIM mode										
	SPI write	CS↓	0x81	0x00	0x00	0x1E	0x6B	CS↑(60ns)				Write 0x6B to VCR 1E: enter DFIM
5		Initialize Status Register and clear Block Protect Bits										
5.1	SPI write	CS↓	0x01	0xFF	CS↑(1.5us)							Write 0xFF to status register
5.2	SPI write	CS↓	0x01	0x00	CS↑(1.5us)							Write 0x00 to status register
5.3	SPI write	CS↓	0x01	0xFF	CS↑(1.5us)							Write 0xFF to status register
5.4	SPI write	CS↓	0x01	0x00	CS↑(1.5us)							Write 0x00 to status register, clear block protect bits
6		Initialize Nonvolatile Configuration Registers										
6.1	SPI write	CS↓	0xB1	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	Write NVCR 0~12
				0x00	0x00	0x00	0x00	0x00	0x00	0x00	CS↑(1.5us)	
6.2	SPI write	CS↓	0xB1	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF		Write NVCR 0~12
				0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	CS↑(1.5us)	
6.3	SPI write	CS↓	0xB1	0x00	0x00	0x00	0x00	0x00	0x00	0x00		Write NVCR 0~12
				0x00	0x00	0x00	0x00	0x00	0x00	0x00	CS↑(1.5us)	
6.4	SPI write	CS↓	0xB1	0x00	0x00	0xFF	0xFF	0xFF	0xFF	0xFF		Write NVCR 0~12
				0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	CS↑(1.5us)	
7		Initialize OTP area										
7.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0xF9	CS↑(60ns)				Write VCR 8: Disable OTP lock
7.2	SPI write	CS↓	0x42	0x00	0x00	0x00	0xFF	(0xFF)	0xFF	CS↑(1.5us)		Write OTP byte 0~256 to "1" (257 bytes)
		Byte count	1	2	3	4	5	6~260	261			
7.3	SPI write	CS↓	0x81	0x00	0x00	0x08	0xF9	CS↑(60ns)				Write VCR 8: Disable OTP lock
7.4	SPI write	CS↓	0x42	0x00	0x00	0x00	0x00	(0x00)	0x00	CS↑(1.5us)		Write OTP byte 0~256 to "0" (256 bytes)
		Byte count	1	2	3	4	5	6~260	260			
7.5	SPI write	CS↓	0x81	0x00	0x00	0x08	0xF9	CS↑(60ns)				Write VCR 8: Disable OTP lock

7.6	SPI write	CS↓	0x42	0x00	0x00	0x00	0xFF (0xFF)	0xFF	CS↑(1.5us)	Write OTP byte 0~256 to "1" (257 bytes)
		Byte count	1	2	3	4	5	6~260	261	
7.7	SPI write	CS↓	0x81	0x00	0x00	0x08	0xF9		CS↑(60ns)	Write VCR 8: Disable OTP lock
7.8	SPI write	CS↓	0x42	0x00	0x00	0x00	0x00 (0x00)	0x00	CS↑(1.5us)	Write OTP byte 0~256 to "0" (256 bytes)
		Byte count	1	2	3	4	5	6~260	260	
8	Bulk Erase (initialize MRAM cells, erase to "0")									
8.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0x7F		CS↑(60ns)	Write VCR 8: set erase to "0"
8.2	SPI write	CS↓	0xC4	0x00					CS↑(60ns)	Select Die "0"
8.3	SPI write	CS↓	0xC7						CS↑(125ms)	Erase all MRAM cells to "0"
8.4	SPI write	CS↓	0xC4	0x01					CS↑(60ns)	Select Die "1"
8.5	SPI write	CS↓	0xC7						CS↑(125ms)	Erase all MRAM cells to "0"
9	Bulk Erase (initialize MRAM cells, erase to "1")									
9.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0xFF		CS↑(60ns)	Write VCR 8: set erase to "1"
9.2	SPI write	CS↓	0xC4	0x00					CS↑(60ns)	Select Die "0"
9.3	SPI write	CS↓	0xC7						CS↑(125ms)	Erase all MRAM cells to "1"
9.4	SPI write	CS↓	0xC4	0x01					CS↑(60ns)	Select Die "1"
9.5	SPI write	CS↓	0xC7						CS↑(125ms)	Erase all MRAM cells to "1"
10	Bulk Erase (initialize MRAM cells, erase to "0")									
10.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0x7F		CS↑(60ns)	Write VCR 8: set erase to "0"
10.2	SPI write	CS↓	0xC4	0x00					CS↑(60ns)	Select Die "0"
10.3	SPI write	CS↓	0xC7						CS↑(125ms)	Erase all MRAM cells to "0"
10.4	SPI write	CS↓	0xC4	0x01					CS↑(60ns)	Select Die "1"
10.5	SPI write	CS↓	0xC7						CS↑(125ms)	Erase all MRAM cells to "0"
11	Bulk Erase (initialize MRAM cells, erase to "1")									
11.1	SPI write	CS↓	0x81	0x00	0x00	0x08	0xFF		CS↑(60ns)	Write VCR 8: set erase to "1"
11.2	SPI write	CS↓	0xC4	0x00					CS↑(60ns)	Select Die "0"
11.3	SPI write	CS↓	0xC7						CS↑(125ms)	Erase all MRAM cells to "1"
11.4	SPI write	CS↓	0xC4	0x01					CS↑(60ns)	Select Die "1"
11.5	SPI write	CS↓	0xC7						CS↑(125ms)	Erase all MRAM cells to "1"
12	Exit DFIM mode									
	SPI write	CS↓	0x81	0x00	0x00	0x1E	0x00		CS↑(60ns)	Write 0x00 to VCR 0x1E: exit DFIM
13	Configure device for next power up. Write NVCRs.									
13.1	SPI write	CS↓	0xB1	0x00	0x00	0x00	0x		CS↑(1.5us)	Write NCR 0: IO mode, refer to datasheet section 5.5
13.2	SPI write	CS↓	0xB1	0x00	0x00	0x01	0x		CS↑(1.5us)	Write NVCR 1: read dummy cycle, refer to datasheet section 5.10
13.3	SPI write	CS↓	0xB1	0x00	0x00	0x04	0x03		CS↑(1.5us)	

2.3 CS deselect time requirement

CS needs to keep high 1.5us(min) after 0xB1 command (write nonvolatile configuration register).

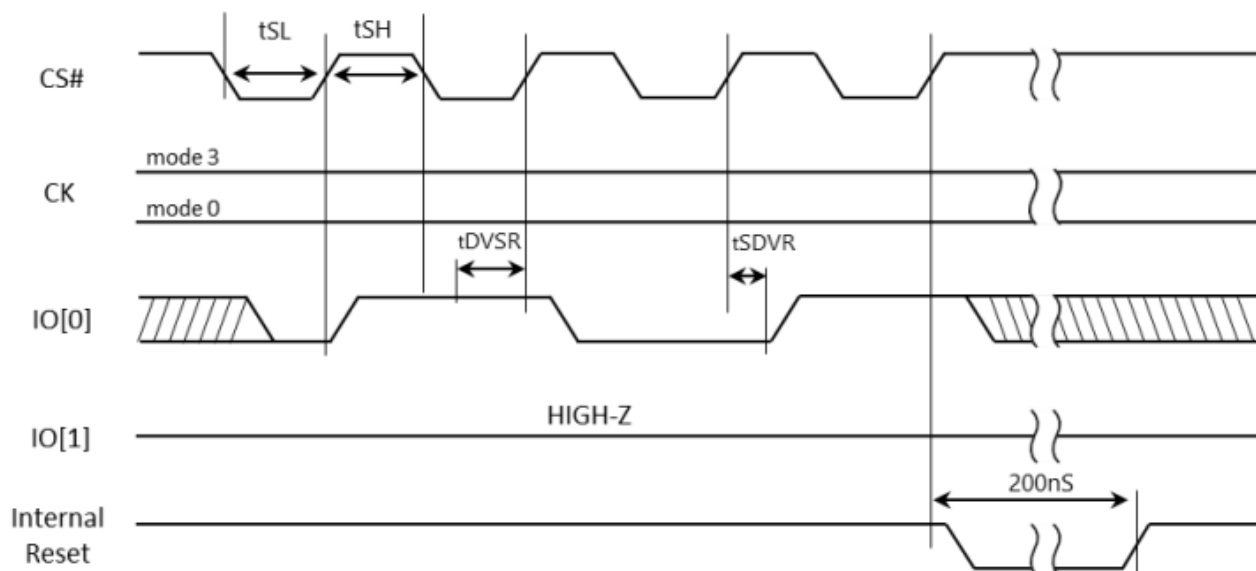
CS needs to keep high 1.5us(min) after 0x01 command (write status register).

CS needs to keep high 125ms(min) after 0x01 command (Bulk Erase).

Parameter	Symbol	Data Transfer Rate	Min	Typ	Max	Unit
CS# deselect time after a non-READ command	t_{SHSL2}	STR/DTR	60	-	-	ns
WRITE STATUS REGISTER cycle time	t_W	STR/DTR	-	-	1.5	μs
WRITE NONVOLATILE CONFIGURATION REGISTER cycle time (per Address)	t_{WNVCR}	STR/DTR	-	-	1.5	μs
PROGRAM OTP cycle time	t_{POTP}	STR/DTR	-	-	1.5	μs
64Mb bulk erase time	t_{BE}	STR/DTR	-	-	125	ms

2.4 JEDEC reset (JESD252.01)

For the details, please refer to datasheet 18.3 and JESD 252.01.

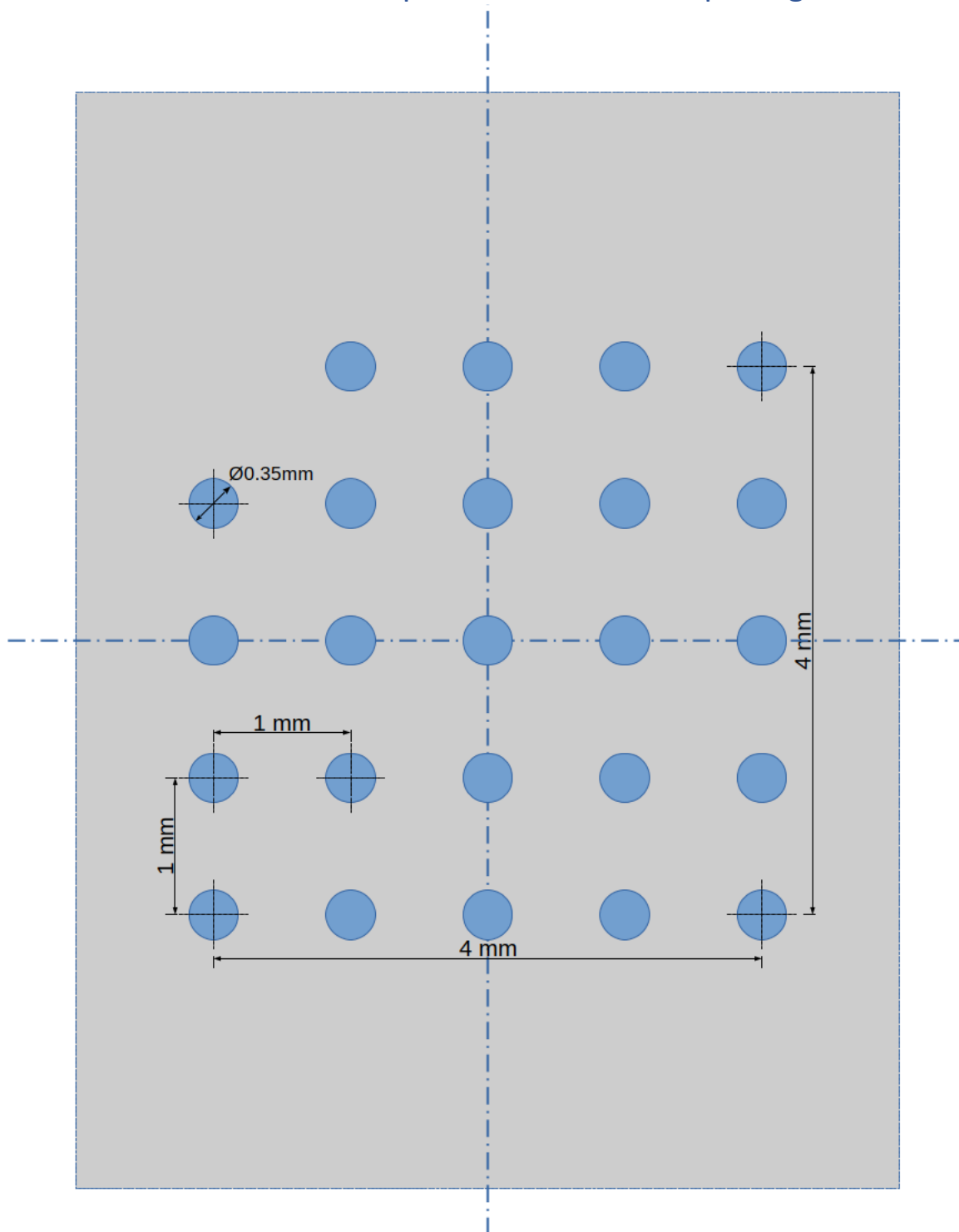


Note: CK should be kept stable high (mode 3) or low (mode 0) to prevent any confusion with commands

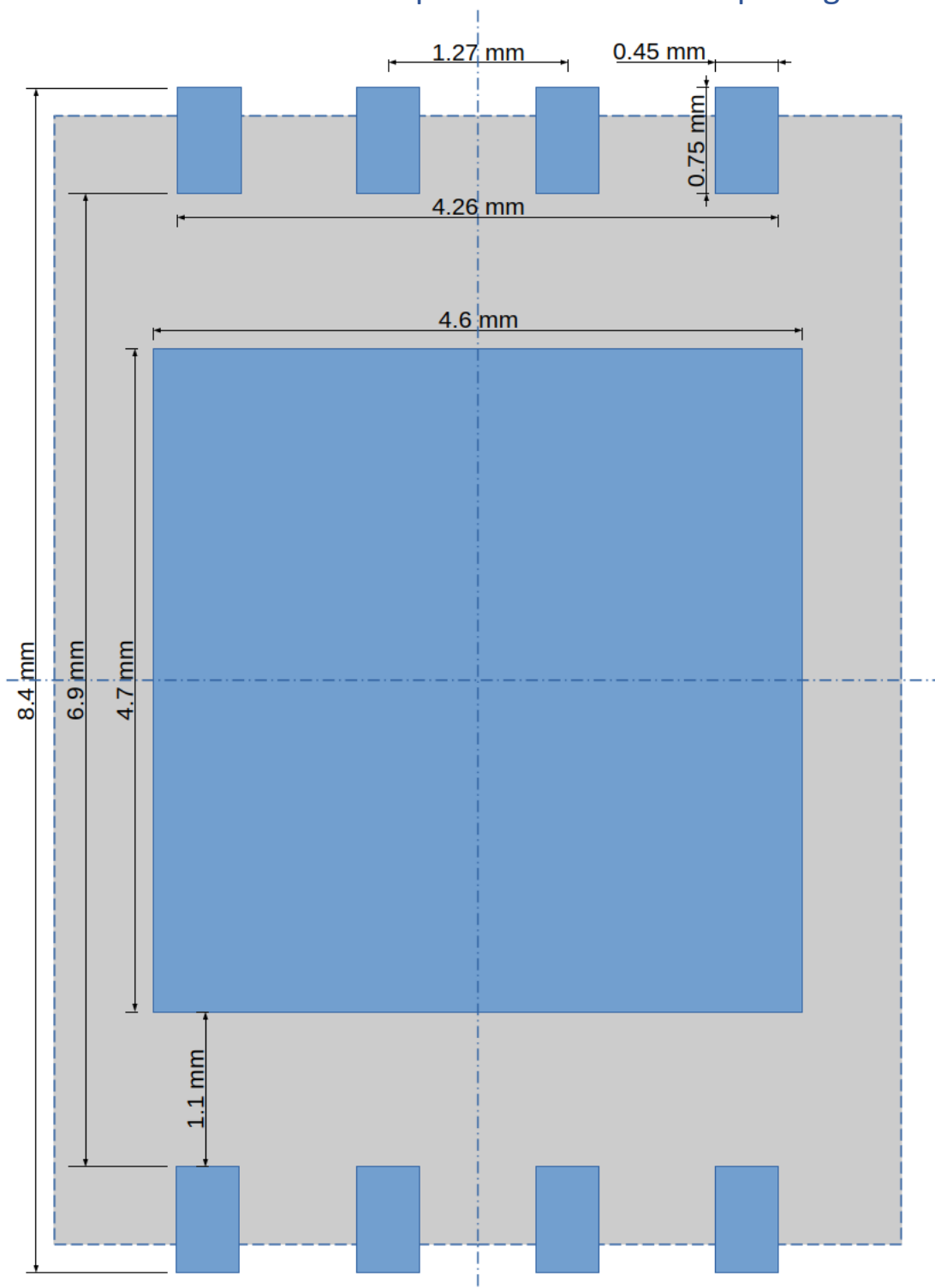
Parameter	Symbol	Min	Max	Units
CS# low time	t_{SL}	500	-	ns
CS# high time	t_{SH}	500	-	ns
Setup time data to CS# for Reset	t_{DVSR}	5	-	ns
Hold time data to CS# for Reset	t_{SDVR}	5	-	ns

3. Printed Circuit Board (PCB) Footprint

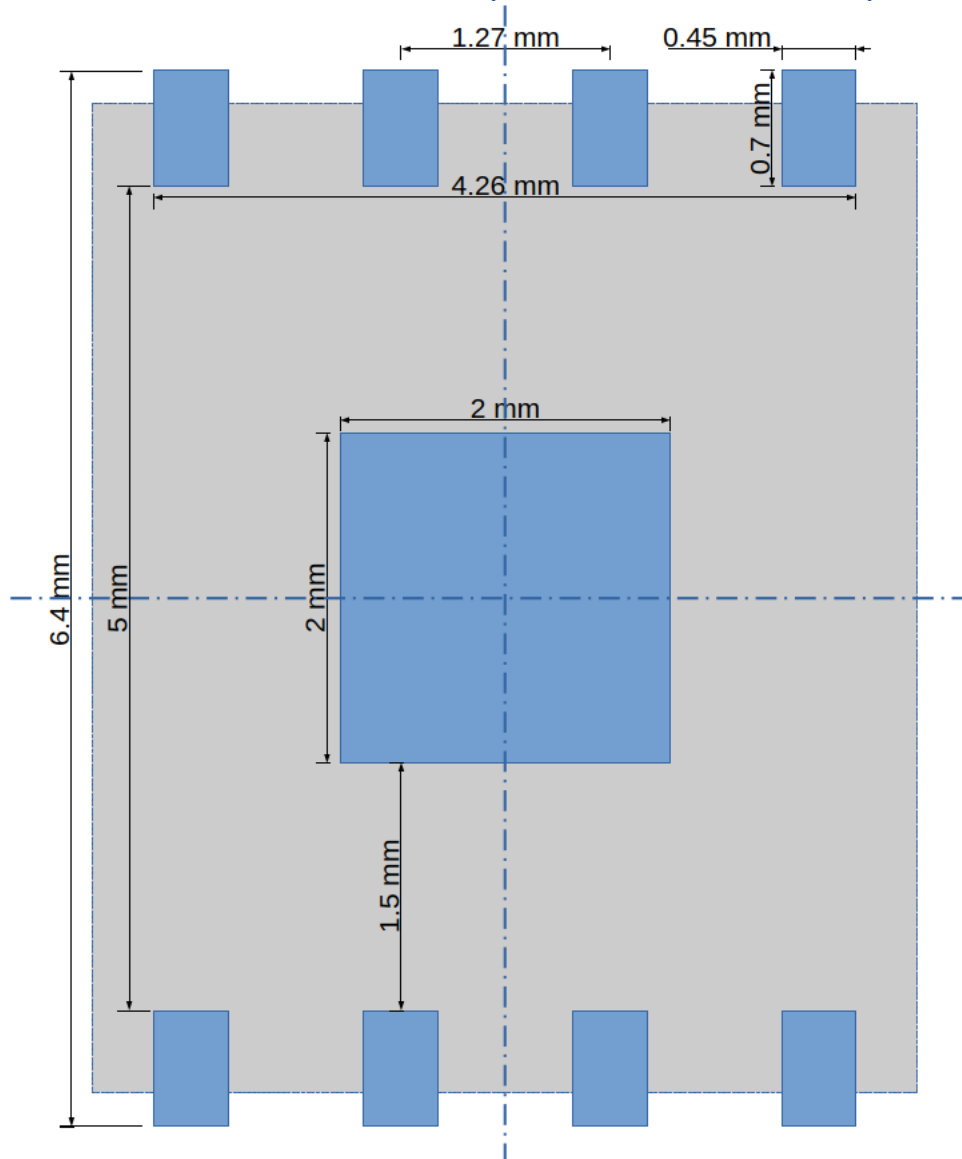
3.1 Recommended PCB footprint for 24 ball BGA package



3.2 Recommended PCB footprint for 6x8mm DFN package



3.3 Recommended PCB footprint for 5x6mm DFN package



Revision History

Revision	Date	Description
1.0	Oct., 2024	Initial Release
1.1	Dec., 2024	Modify device initialization sequence base on EST3000 application note V2.1

Contact Information:

Author: Wes Chang

FAE APAC & Japan

How to Reach Us:www.everspin.com**E-Mail:**support@everspin.comorders@everspin.comsales@everspin.com**USA/Canada/South and Central America**

Everspin Technologies

5670 W. Chandler Road, Suite 130

Chandler, Arizona 85226

+1-877-347-MRAM (6726)

+1-480-347-1111

Europe, Middle East and Africasupport.europe@everspin.com**Japan**support.japan@everspin.com**Asia Pacific**support.asia@everspin.com**Everspin Technologies, Inc.**

Information in this document is provided solely to enable system and software implementers to use Everspin Technologies products. There are no express or implied licenses granted hereunder to design or fabricate any integrated circuit or circuits based on the information in this document. Everspin Technologies reserves the right to make changes without further notice to any products herein. Everspin makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Everspin Technologies assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters, which may be provided in Everspin Technologies data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters including "Typical" must be validated for each customer application by customer's technical experts. Everspin Technologies does not convey any license under its patent rights nor the rights of others. Everspin Technologies products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Everspin Technologies product could create a situation where personal injury or death may occur. Should Buyer purchase or use Everspin Technologies products for any such unintended or unauthorized application, Buyer shall indemnify and hold Everspin Technologies and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Everspin Technologies was negligent regarding the design or manufacture of the part. Everspin™ and the Everspin logo are trademarks of Everspin Technologies, Inc. All other product or service names are the property of their respective owners.