Guidelines to migrate from serial NOR flash to page EEPROM

Introduction

This technical note is intended to help software developers to migrate their application firmware based on a serial NOR flash memory to the page EEPROM.

With its ultralow power consumption the page EEPROM is an ideal choice for battery-operated applications such as IoT trackers, wearables, monitoring, and healthcare devices or tiny camera modules.

Table 1. **Applicable products**

Series	Products
Standard serial page EEPROM	M95P32-E, M95P32-I
	M95P16-E, M95P16-I
	M95P08-E, M95P08-I

1 Reference documents

The table below contains the reference documents mention in this technical note.

Table 2. Reference documents

Reference	Document
AN5747	Page EEPROM memory architecture
AN5866	Guidelines for cycling endurance and data retention of page EEPROMs
TN1397	Page program operation with buffer load usage
DS12964	Ultra low power 32-Mbit serial SPI page EEPROM with dual and quad outputs
AN6076	How to save the application energy budget with page EEPROM family

2 Serial NOR flash in embedded systems

Serial NOR flash is used in embedded systems. Due to their very high communication speeds (ranging from 50 to 200 MHz), serial NOR flash memories are typically used to store firmware or boot code.

However, EEPROM, with its byte architecture, is much more suitable for data logging:

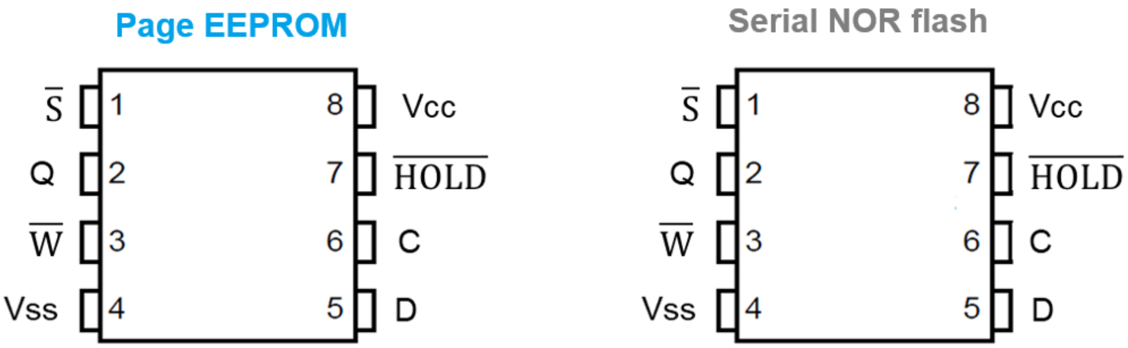
Page EEPROM has the flexibility to manage both firmware and data logging.

Available in 8, 16 and 32 Mbit options, the page EEPROM can replace small and medium (from 1Mbit to 128Mbit) serial NOR flash.

3 Hardware compatibility

The page EEPROM is based on an 8-pin package and supports single, dual and quad serial peripheral interface (SPI). The serial NOR flash with single, dual, or quad SPI interface are based on the same pinout as described below:

Figure 1. **Page EEPROM and serial NOR flash 8-pin package connections**



In standard packages (SO8N and DFN8), the page EEPROM is fully pin-to-pin compatible with serial NOR flash. This means that:

- the PCB does not need to be modified to integrate the page EEPROM.
- there is no additional cost at hardware level to migrate from flash to page EEEPROM.

In addition, with a wide supply voltage range (from 1.6 to 3.6 V) and temperature range (from -40 to +105 °C), the page EEPROM could be compatible with your system.

4 Firmware compatibility

4.1 Instruction set compatibility

The instruction set defines the operations that a memory can perform.

These instructions are represented in binary form. For example, a read status register can be represented as 0000 0101 (05h) in an 8-bit instruction set. When transitioning from one type of memory to another, it is essential to review datasheets and compare the differences in their instruction sets.

The Table 3 compares the instructions of a typical serial NOR flash with those of a page EEPROM:

Table 3. **Instruction set comparison between serial NOR flash and page EEPROM**

Operation	Serial NOR flash	Page EEPROM
Array access		
Read	03h	03h
Fast read	0Bh	0Bh
Dual read	3Bh	3Bh
Quad read	6Bh	6Bh
Page program	**02h**	**0Ah**
Quad page program	38h	**NA**
Page write	**NA**	**02h**
Page erase (512 bytes)	**NA**	DBh
Sector erase (4 Kbytes)	20h	20h
Block erase (64 Kbytes)	D8	D8
Chip erase	C7h	C7h
Device operation		
Write enable	05h	05h
Write disable	06h	06h
Prog/Erase suspend	75h	**NA**
Prog/Erase resume	02h	**NA**
Deep power down enter	7Ah	B9h
Deep power down release	ABh	ABh
Enable reset	66h	66h
Software reset	99h	99h
Register access		
Read status register	05h	05h
Write status register	01h	01h
Read identification page	**NA**	83h
Write identification page	**NA**	82h
Read configuration and safety register	15h	15h
Read volatile register	**NA**	85h
Write volatile register	**NA**	81h
Read SFDP	5A	5A
JEDEC identification	9F	9F

Apart from the page program instruction, if an operation exists in both products, it uses the same instruction, ensuring complete opcode compatibility between the page EEPROM and the serial NOR flash.

The page program instruction differs between the page EEPROM and the serial NOR flash. This is because the page EEPROM has a page write operation, unlike the serial NOR flash, and the opcode 02h is reserved for this operation, similar to a standard EEPROM.

For more information about the page write operation, refer to AN5747.

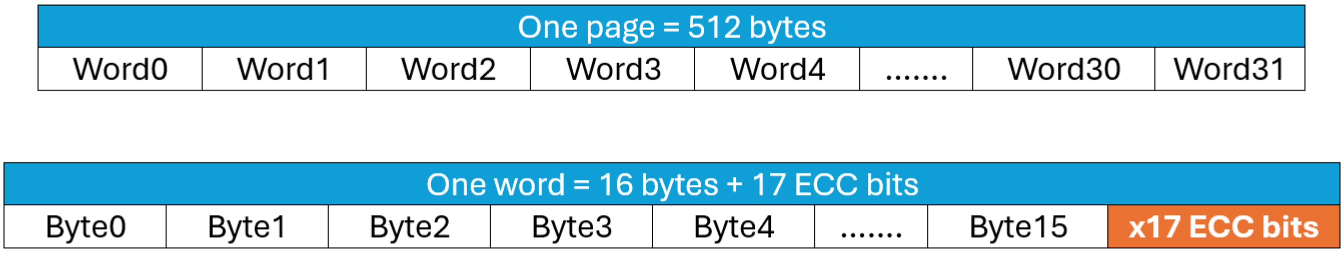
4.2 Error Correction Code (ECC) algorithm

The Error Correction Code (ECC) is a specific logic designed to correct or detect bits errors. This embedded algorithm is transparent for the user and increases the integrity and the longevity of stored data.

In the page EEPROM, the ECC can correct up to two-bits errors and detect three-bits errors over 16 bytes. In the safety register, ECC flag bits are updated according to bit correction or detection during the read operation.

Based on the ECC, the pages are divided into 32 words of 16 bytes each, as described in Figure 2

Figure 2. **Page architecture with the ECC**



One page = 512 bytes							
Word0	Word1	Word2	Word3	Word4	....	Word30	Word31

One word = 16 bytes + 17 ECC bits							
Byte0	Byte1	Byte2	Byte3	Byte4	....	Byte15	x17 ECC bits

4.3 Write, program and erase data in page EEPROM

4.3.1 Page program operation

With 512 bytes per page (compared to 256 bytes for serial NOR flash), the page EEPROM can program twice as much data as serial NOR flash in one operation. Therefore, the application firmware does not need to be updated but could be optimized in terms of loops or buffers.

Due to the page architecture and ECC, programming a single byte in the page EEPROM requires the entire word (16 bytes) to be erased. Refer to AN5747 for more information. If the application firmware is programming byte by byte, two solutions are possible:

- Use a buffer in the controller embedded memory to program 16 bytes (to optimize memory capacity) up to 512 bytes (to optimize program power consumption) at once.
- Use the page write operation (see below).

Note: *Unlike the page write or erase operations, the page program has no impact on cycling endurance.*

For programming a large amount of data (such as firmware), it is recommended to use the page program operation, as it is faster than the page write operation.

4.3.2 Page write operation

Serial NOR flash typically requires sector-level erasure before writing, whereas page EEPROM allows byte-level writing.

With the page write operation, the page EEPROM can write up to 512 bytes within the memory without erasing blocks of memory first. Refer to AN5747 for more information.

This feature is adapted for use cases such as data logging (recording data periodically, such as sensor readings) or configuration settings (updating the configuration parameters that change occasionally).

Each page write operation counts as one write cycle over the page.

4.3.3 Erase options

As serial NOR flash, page EEPROM offers sector, block and chip erase operations. However, in most serial NOR flash devices, the sector (4 Kbyte) is the smallest area that can be erased. In contrast, the page EEPROM architecture offers a page erase operation (512 bytes). This gives software developers more flexibility to erase only a small amount of data when required. For instance, the page erase might be used to prepare the erased area before an event recording.

Note: Event recording is a method used to store data of specific behaviors or events within a given time frame.

4.3.4 Cycling endurance and software emulation

The page EEPROM offers 500k cycles per page over the full temperature range (-40°C to +105°C). Only page write and erase operations are considered as write cycles. In contrast, serial NOR flash is less robust, offering only 10k to 100k cycles.

Note: For more information about cycling, refer to AN5866.

The serial NOR flash has limited cycling endurance and required block or sector erase for writing data. Therefore, when an application needs to update data frequently, the serial NOR flash devices use software emulation. This simulates EEPROM functionality by using a portion of the flash memory to store data with mechanisms for wear leveling and data integrity.

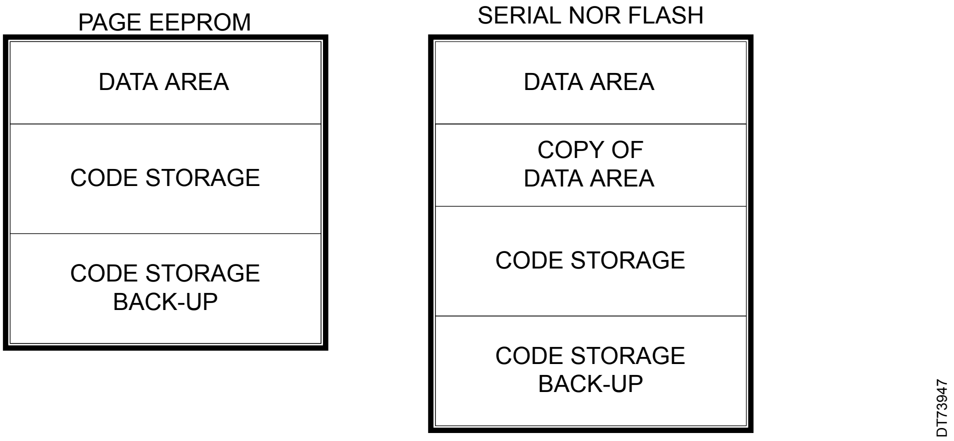
Implementing EEPROM emulation requires complex software algorithms to manage wear leveling, error correction, and data integrity. This adds to the development time and the complexity. Additionally, the emulation requires additional storage for metadata (status flags, wear leveling information, error correction codes, and other) and additional pages can be reserved to reduce wear, further reducing the available memory capacity.

With the page EEPROM you have access to the page write operation. Like a standard EEPROM, you can update any byte in the device. It drastically reduces the memory block reserved for datalogging.

If an application updates and stores 1000 parameters, and each parameter has the size of 1 byte.

- **Page EEPROM**:
  In a page EEPROM, storing 1000 parameters of 1 byte each would require 1000 bytes.
- **Flash memory with EEPROM emulation**:
  The EEPROM emulation in the flash memory typically requires additional overhead for wear leveling and metadata:
  - **Metadata**: Each parameter might require additional bytes for metadata (such as address or status). Assuming 2 bytes of metadata are needed per parameter, 1000 parameters would need 3000 bytes (1000 for the values, and 2000 for the metadata).
  - **Wear Leveling**: To ensure longevity and reliability, extra space is needed. A common practice is to allocate 2-3 times the actual data size to accommodate wear leveling. For 1000 parameters, 6000 to 9000 bytes (3000 bytes of metadata x 2 or 3) are needed, compared to the 1000 bytes required by the page EEPROM.

Figure 3. **Firmware size for datalogging and code storage**



In addition to the software complexity required to manage EEPROM emulation, the area reserved for data logging is much larger with the serial NOR flash, compared to the page EEPROM. Therefore, with equivalent densities, you can store more data with the page EEPROM, making your application more efficient and accurate.

4.4 Page EEPROM advanced features

This section lists the advanced features that are now available when migrating from serial NOR flash to page EEPROM.

Byte access level

With its smart architecture, the page EEPROM offers the write operation like the standard EEPROM. This means that you can overwrite 1 to 512 bytes without an erase operation beforehand. This simplifies the firmware development. Software emulation required for serial NOR flash to mirror the standard EEPROM behavior in datalogging use case is no longer needed.

Page program with buffer load

Page program instruction decoding and buffering is available while a previous program instruction is executing. It effectively hides the SPI communication and can save up to 50% of time. For more information, refer to TN1397.

Output driver strength

The page EEPROM determines the output driver strength for the read operations. With 4 (or 2 depending on memory density) buffer strengths (default, low, medium, high), it offers the possibility to improve the memory access time or to balance performance with energy efficiency. For more information, refer to the datasheet DS14962, *section 5.2.1 Configuration register*.

Safety register

This register monitors the proper functioning of page EEPROM operations. Offering 8 bits that provide information about power-up, ECC, or modify operations, it enhances the safety and robustness of the application firmware. For more information, refer to the datasheet DS14962, section 5.2.2 Safety Register.

Ultra-low power

The page EEPROM has been designed to combine high performance and energy saving. This memory can fit applications powered by tiny batteries as the current peaks are controlled to go below than 4mA. It is also possible to drive the VCC of page EEPROM with the GPIO of a microcontroller to chase µA even when the memory is in deep power down mode. For more information, refer to the AN6076.

Ultra-fast erase

The page EEPROM’s erase operations are very fast. For example, the block erase operation is only 4 ms whereas serial NOR flash can take up to 400ms. When it comes to erasing a large amount of data, such as FOTA (Firmware Over the Air) updates, the page EEPROM saves time and reduces downtimes.

Stability of operations among parts

The page EEPROM is reliable for critical time operations such as FOTA (Firmware Over The Air) updates or boot loading. Additionally, since the performance (timing, power consumption) varies very little from one chip to another, the overall stability of the final application is greatly improved.

4.5 Serial NOR flash features not available in page EEPROM

These features are not embedded in page EEPROM products:

- Double data rate (DDR)
- Continuous Read Mode
- Burst Read
- Suspend/Resume
- Quad Page Program

Note: This list is not exhaustive.

5 Conclusion

Migrating from a serial NOR flash to a page EEPROM is easy. The application PCB does not need to be adapted as there is pin-to-pin full compatibility for standard packages (SO8N and DFN8).

In terms of firmware, only a few adjustments are needed as most of the opcodes are similar. Additionally, with the byte-access write capability of the page EEPROM, firmware based on serial NOR flash can be simplified by removing EEPROM emulation and wear leveling.

Moreover, switching from flash memory to page EEPROM offers advanced features in terms of speed, ultra-low power consumption, and reliability.

Revision history

Table 4. **Document revision history**

Date	Version	Changes
06-Mar-2025	1	Initial release.

Contents

List of tables

List of figures

IMPORTANT NOTICE – READ CAREFULLY