SMART ARM-based Microcontrollers

AT03974: Read While Write EEPROM

APPLICATION NOTE

Description

The non-volatile memory of the SAM L21 and SAM D21 device variant B features a Read While Write EEPROM Emulation (RWWEE) section. This section is intended for EEPROM emulation and can be programmed at the same time as reading the main array.

This application note will highlight the benefits and show how to use the RWWEE section in the SAM L21 and SAM D21 non-volatile memory.

Features

- Benefits of RWWEE
- How to use RWWEE

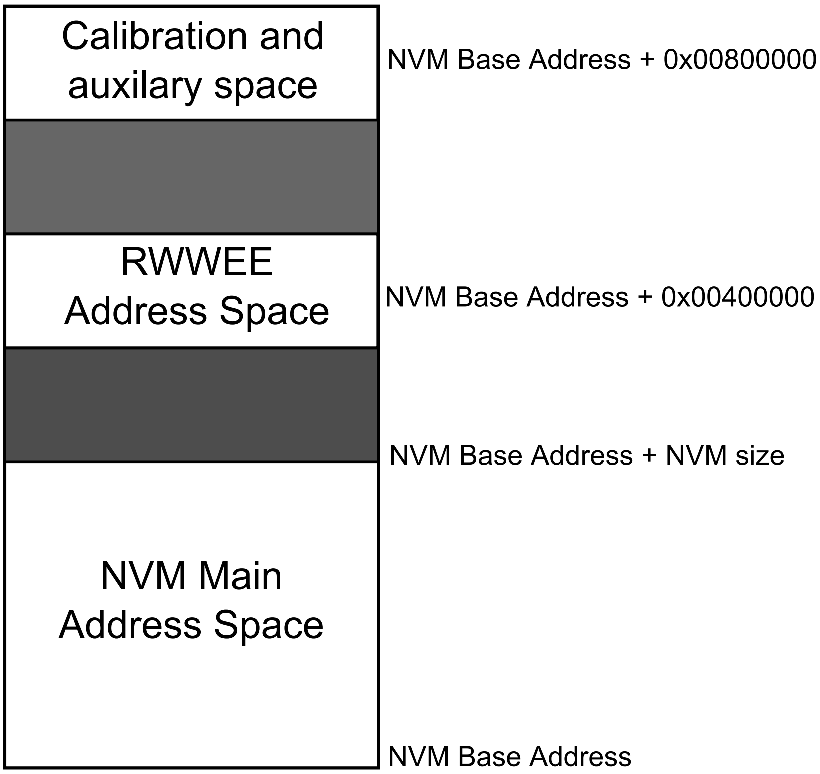
1. Introduction

This application note will introduce the read while write EEPROM (RWWEE) section featured in the SAM L21 and SAM D21 device variant B.

Depending on the device, the RWWEE section can be 1KB, 2KB, 4KB, or 8KB. This is a separate array intended for the RWW EEPROM emulation, and can be programmed while reading the main array.

Figure 1-1 NVM Memory Organization on page 2 shows the organization of the non-volatile memory of the SAM L21. The EEPROM emulation space is memory mapped in the same way as the main array. For the SAM D21 device variant B, the RWWEE address space is located at NVM Base Address + 0x000100000.

Figure 1-1 NVM Memory Organization


Calibration and auxilary space
NVM Base Address + 0x00800000
RWWEE Address Space
NVM Base Address + 0x00400000
NVM Base Address + NVM size
NVM Main Address Space
NVM Base Address

In legacy implementations of the NVM controller, the EEPROM emulation area was only available as a part of the main area, which meant that while this was being programmed, all code fetching and execution would stop. The RWWEE section featured in the SAM L21 and SAM D21 device variant B can be written at the same time as the main area can be read. In addition, it is still possible to use a part of the main area for EEPROM emulation. The size of this area can be configured with the EEPROM bits in the NVM User Row.

When reading from the RWWEE area, it is not possible to program the main array.

To use the RWWEE as an EEPROM, an emulated EEPROM scheme must be applied. There are many different algorithms that may be employed for EEPROM emulation, to tune the write and read latencies, RAM usage, wear leveling, and other characteristics. As a result, multiple different emulator schemes may be implemented, so that the most appropriate scheme for a specific application's requirements may be used. ASF features one implementation of an EEPROM emulation scheme, which is available both for the RWW EEPROM emulation section and EEPROM emulation in the main area. The user is free to implement any other EEPROM emulation scheme as needed.

2. Benefits

Reading the NVM main address space while a write or erase operation is ongoing on the NVM main array results in an AHB bus stall until the end of the operation. When using the RWWEE area for emulated EEPROM instead of the main address space, the processor is able to read the main area at the same time as the RWWEE area is written or erased. This allows for the processor to service interrupt requests or run linear code at the same time as programming the emulated EEPROM.

As the amount of time used to program a page or erase a row is in the order of several milliseconds, the RWWEE will allow for much more deterministic and efficient code when using this section for storing data compared to using the EEPROM section in the main array.

3. How to use the RWWEE Section

3.1. Command and Data Interface

The NVM Controller is addressable from the APB bus, while the NVM main address space is addressable from the AHB bus. Read and automatic page write operations are performed by addressing the NVM main address space or the RWWEE address space directly, while other operations such as manual page writes and row erases must be performed by issuing commands through the NVM Controller.

To issue a command, the CTRLA.CMD bits must be written along with the CTRLA.CMDEX value. When a command is issued, INTFLAG.READY will be cleared until the command has completed. Any commands written while INTFLAG.READY is low will be ignored.

3.2. NVM Write

The NVM Controller requires that an erase must be done before programming. The entire NVM main address space and the RWWEE address space can be erased by a debugger Chip Erase command. Alternatively, rows can be individually erased by the Erase Row command or the RWWEE Erase Row command to erase the NVM main address space or the RWWEE address space, respectively.

Data to be written to the NVM block are first written to and stored in an internal buffer called the *page buffer*. The page buffer contains the same number of bytes as an NVM page. Writes to the page buffer must be 16 or 32 bits. 8-bit writes to the page buffer are not allowed and will cause a system exception.

Both the NVM main array and the RWWEE array share the same page buffer. Writing to the NVM block via the AHB bus is performed by a load operation to the page buffer. For each AHB bus write, the address is stored in the ADDR register. After the page buffer has been loaded with the required number of bytes, the page can be written to the NVM main array or the RWWEE array by setting CTRLA.CMD to 'Write Page' or 'RWWEE Write Page', respectively, and setting the key value to CMDEX. The LOAD bit in the STATUS register indicates whether the page buffer has been loaded or not. Before writing the page to memory, the accessed row must be erased.

3.3. Erase Row

Before a page can be written, the row containing that page must be erased. The Erase Row command can be used to erase the desired row in the NVM main address space. The RWWEE Erase Row can be used to erase the desired row in the RWWEE array. Erasing the row sets all bits to '1'. If the row resides in a region that is locked, the erase will not be performed and the Lock Error bit in the Status register (STATUS.LOCKE) will be set.

3.4. RWWEE Read

Reading from the RWW EEPROM address space is performed via the AHB bus by addressing the RWWEE address space directly.

Read timings are similar to regular NVM read timings when access size is Byte or half-Word. The AHB data phase is twice as long in case of full-Word-size access.

It is not possible to read the RWWEE area while the NVM main array is being written or erased, whereas the RWWEE area can be written or erased while the main array is being read.

The RWWEE address space is not cached, therefore it is recommended to limit access to this area for performance and power consumption considerations.

3.5. Examples

To erase a row, the 16-bit hardware address of the row to be erased must first be loaded into the ADDR register. Then, the erase command must be issued together with the command execute key.

```
NVMCTRL->ADDR.reg = NVMCTRL_RWW_EEPROM_ADDR/2;
NVMCTRL->CTRLA.reg = NVMCTRL_CTRLA_CMD_RWWEEER | NVMCTRL_CTRLA_CMDEX_KEY;
```

By default, automatic page writes are enabled. This means that when the last location of a page is written, the page buffer will be automatically written to memory.

If manual page writes are enabled, the address must be loaded to the ADDR register before issuing the write with the RWWEE Write Page command. This will write the contents of the page buffer to the page addressed by ADDR. ADDR is automatically updated while writing to the page buffer, so it is not necessary to write the ADDR manually unless a different page in memory is to be written.

```
NVMCTRL->ADDR.reg = NVMCTRL_RWW_EEPROM_ADDR/2;
NVMCTRL->CTRLA.reg = NVMCTRL_CTRLA_CMD_RWWEEWP | NVMCTRL_CTRLA_CMDEX_KEY;
```

After issuing a command, the NVM Ready bit in the INTFLAG register will be set to 0. This bit will be set to 1 when the programming is complete. The user can either wait for this by polling for this bit to be set, or enable the NVM Ready interrupt to get an interrupt when the NVM Controller is ready to accept the next command. When implementing an interrupt-driven scheme for writing the RWWEE, the SAM L21 can continue to execute other parts of the code while the emulated EEPROM is being programmed.

Regardless of polling or using interrupts for programming the RWWEE, the device is able to service any incoming interrupts at the same time as the RWWEE area is being programmed.

4. Revision History

Doc. Rev.	Date	Description
42413A	02/2015	Initial document release.

Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

DISCLAIMER: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (“Safety-Critical Applications”) without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.