**Non-volatile Memory : EEPROM vs Flash**

March 6, 2014

In embedded system design, EEPROM and Serial Flash(e.g Adesto DataFlash AT45DB321E, or previously AT45DB321D from Atmel) are often use as non-volatile memory. Both devices has distinctive behavior and it is important to understand these behaviors to select the right device for the right applications.  In this post I will summarize key differences between serial EEPROM(refer as EEPROM below) and serial Flash(refer as Flash below)

**Capacity**

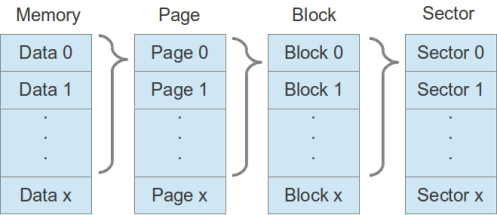
EEPROM have smaller capacity (typically in Kbit size) will Flash have a larger capacity (typically in Mbit size)

**Bus Type To Communicate With Master**

Dependant on device itself, both EEPROM and Flash can support on I2C or SPI bus. I2C bus is running at slower speed (standard of 100KHz/400KHz), it is commonly supported in EEPROM. While SPI bus is more commonly seen in Flash device due to high serial speed (in MHz range, e.g. 40/60MHz)

**Memory Layout**

EEPROM data memory is seen as a block of continuous memory, each data with respective address. Each data byte is access by its respective memory address. Fundamentally flash is having the same memory layout as EEPROM. But Flash has divided memory into three levels of granularity, namely: pages, blocks, sectors(This apply to Atmel DataFlash/Adesto DataFlash) . A different flash type may have different memory layout. A pages is consisted of a section of continuous data memory, e.g. 512 bytes data is located in the same pages. Multiple pages will be group as a single block and multiple block will be group as a single sectors.

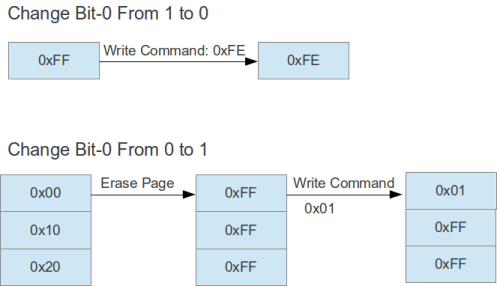
[](https://tkcheng.files.wordpress.com/2014/02/flashmemorylayout.png)

Flash Memory Layout

**Read/Write Process**

Reading data is similar between EEPROM and Flash.

However, there is a distinctive difference between writing behavior. First and foremost is Flash does not allow a bit change of ‘0’ to ‘1’ in a single write command. Instead a page erase command has to be issue, which in effect will erase a particular page of memory (into value of 0xff). User can change the bit from ‘1’ to ‘0’ in a single write command. Examples as shown below:

[](https://tkcheng.files.wordpress.com/2014/02/flashwrite.png)Flash Data Writing Process

As shown on diagram above, writing data 1 into existing data(with value 0) involve page erase, which will erase other data memory(in same page) into 0xFF as side effect. And in order to provide data integrity it is require to write other memory with original data (in this case 0x10, 0x20).  The additional erase stage as well as rewrite of data on other data (in same page) will have impact of the following:

* A difference code handling on writing value of 1 or 0, if optimization is require to reduce read/write.
* A longer write time as Page Erase process would take some time (in milliseconds)
* A chance of data corruption when power outage during:  page erase, or data writing
* Writing a individual bit may involve erase and re-write of whole page of memory

All the above will have great impact on product if it is not handle properly.

In additional to Page Erase, flash is also equip with ‘Block Erase’, ‘Sector Erase’ and ‘Chip Erase’ to cater for large memory erase. Erasing memory may take up to seconds as shown below

[FlashEraseTiming](https://tkcheng.files.wordpress.com/2014/02/flasherasetiming.png)

Duration of Memory Erase Samples

**Endurance**

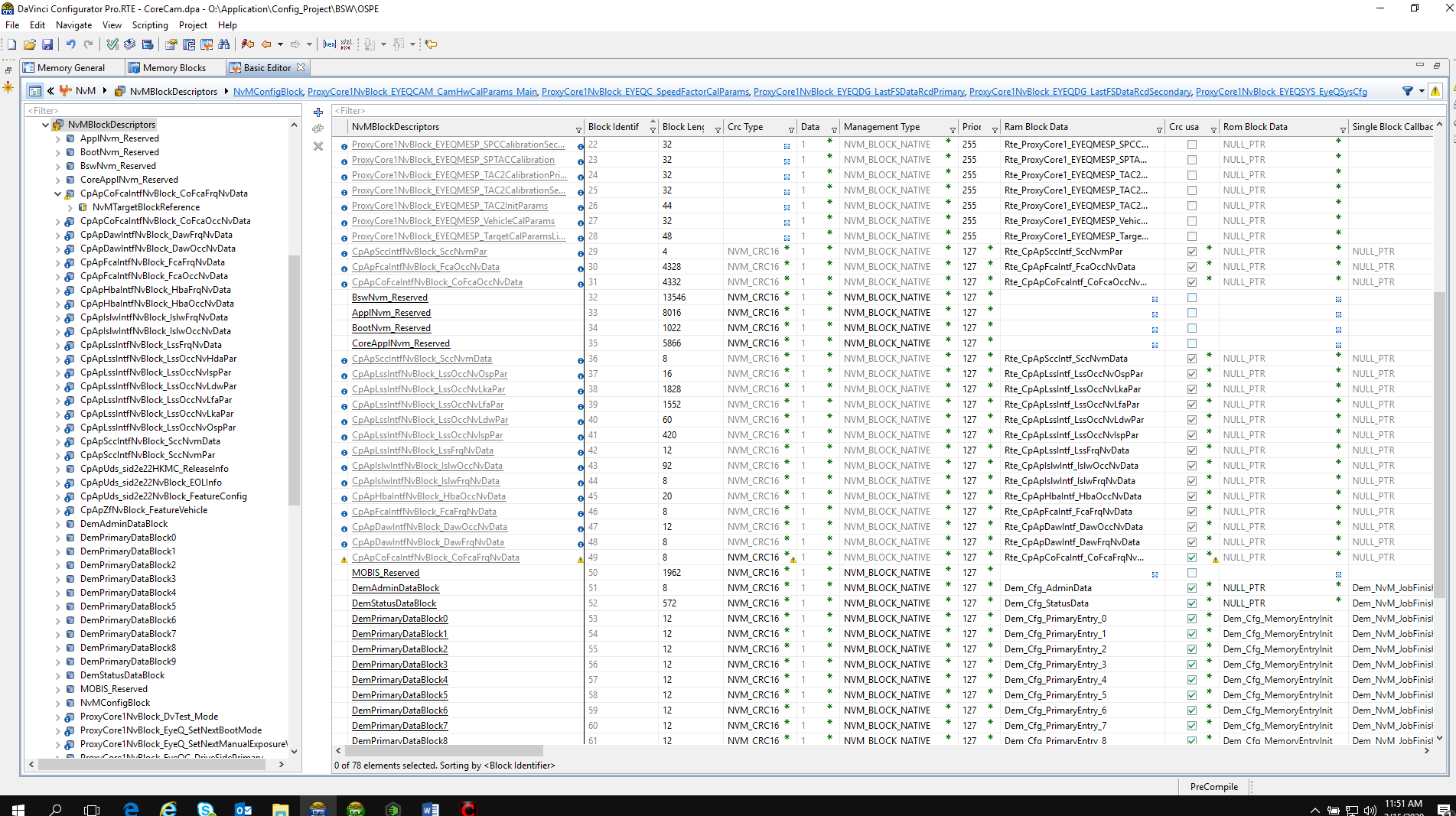
Typical EEPROM life cycle is written as 100K erase/writes. My interpretation of this would be each byte of data memory have an endurance of 100K cycles. This is commonly available across different vendors.

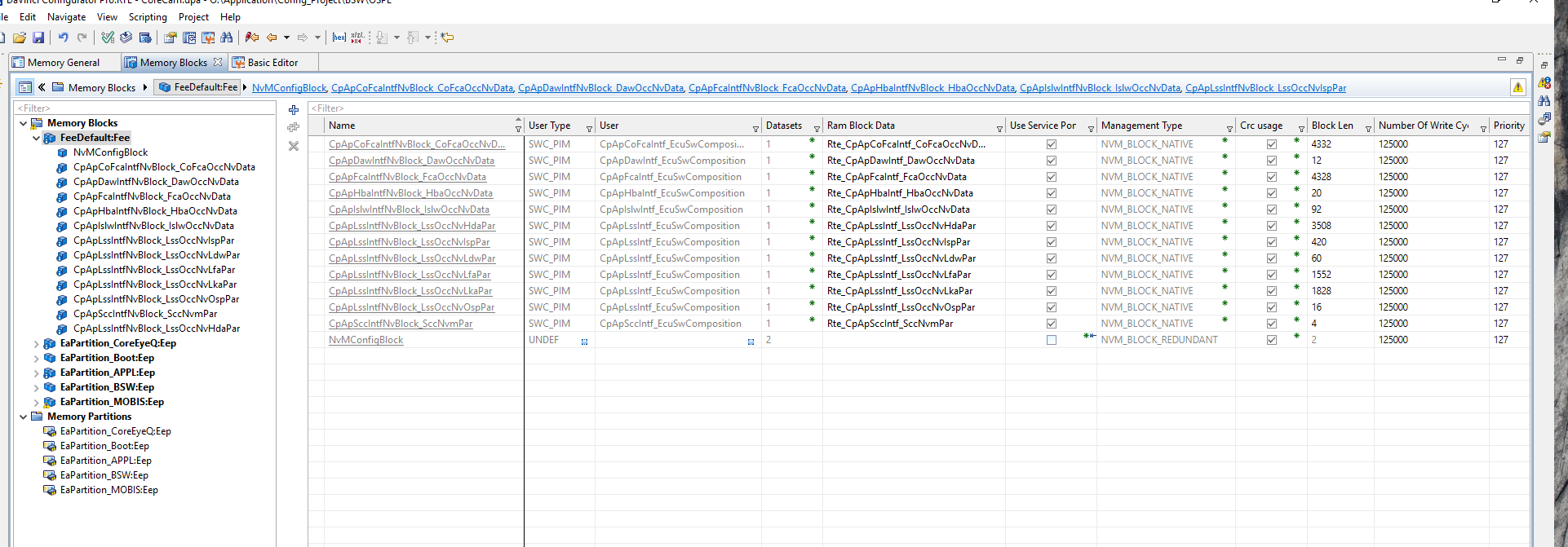
However when comes to Flash, different vendor will provide different interpretation on this. Some examples would be:

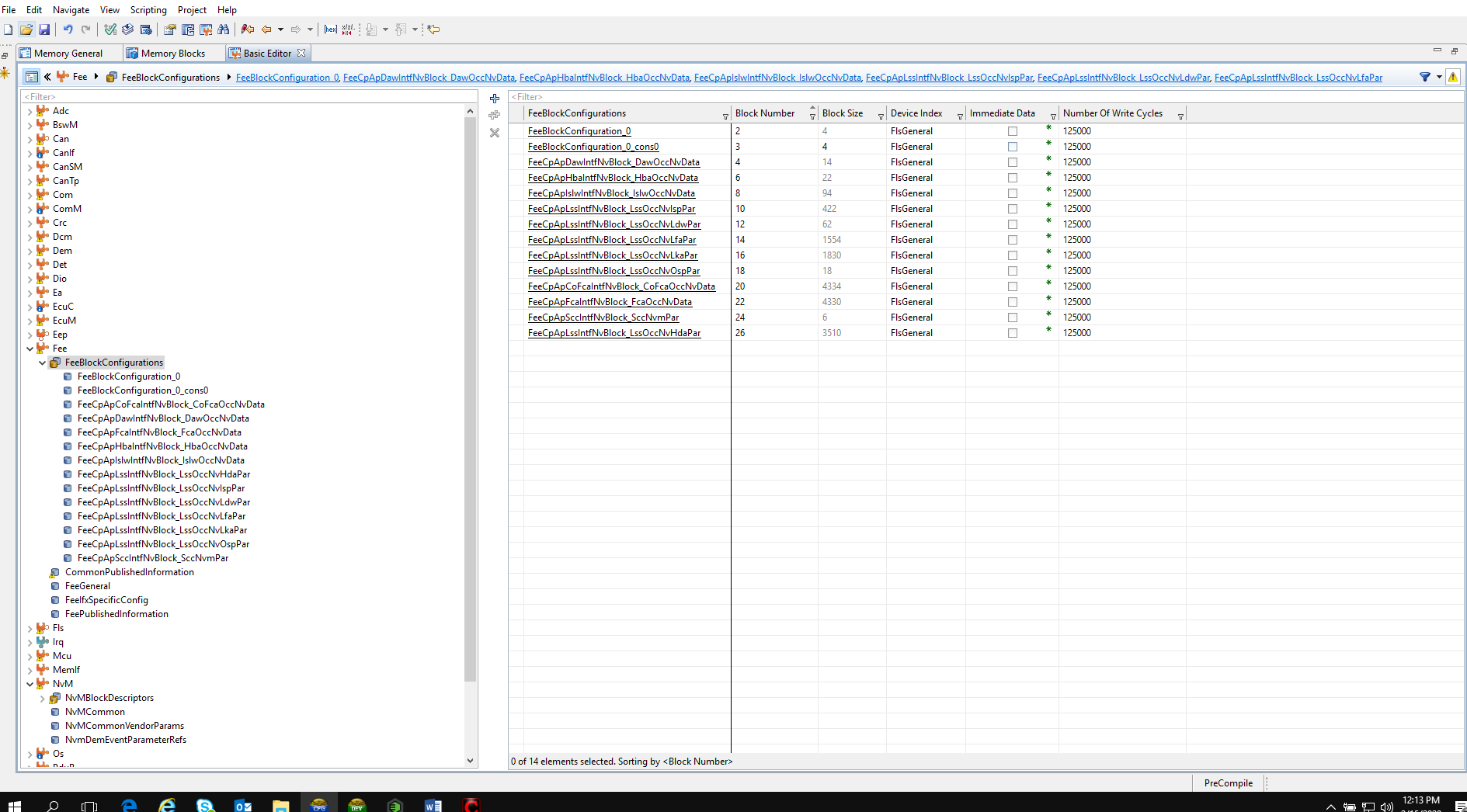
* Vendor A: 100,000 Program/Erase Cycles Per Page Minimum
* Vendor B: Endurance: 100,000 Cycles (typical)
* Vendor C: 100,000 Program-Erase Cycles on any sector typical

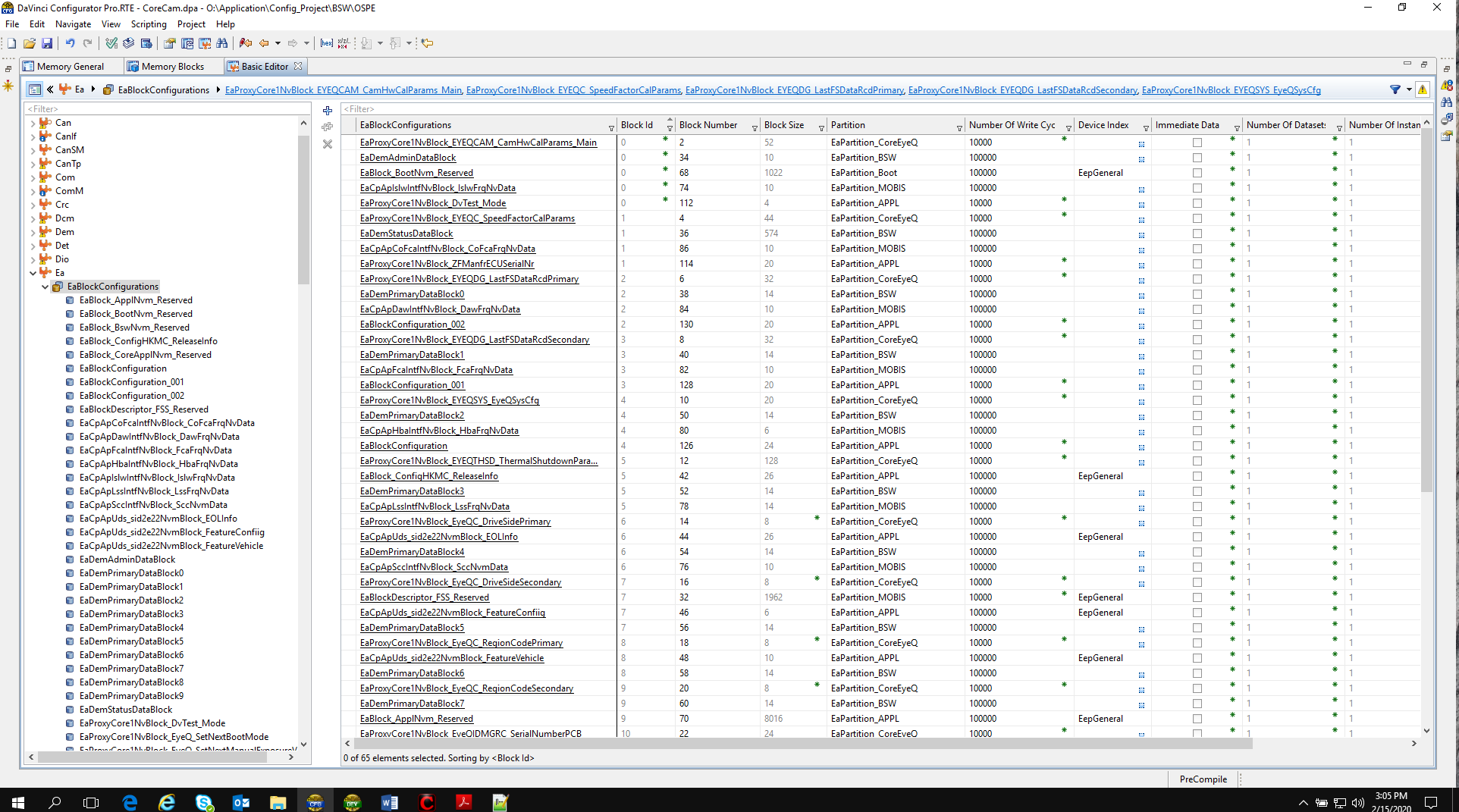
At first glance, it seems that both EEPROM and Flash is offering 100K program/erase cycles. But the real problem of Flash lie on the data writing describe aboves. Lets take Vendor A as example, with 100K program/erase cycles on each pages. Referring to the datasheet, each sector contains 512Bytes. Dividing this to 100Kbytes, each data memory is actually have an endurance of **200** read/write cycles.

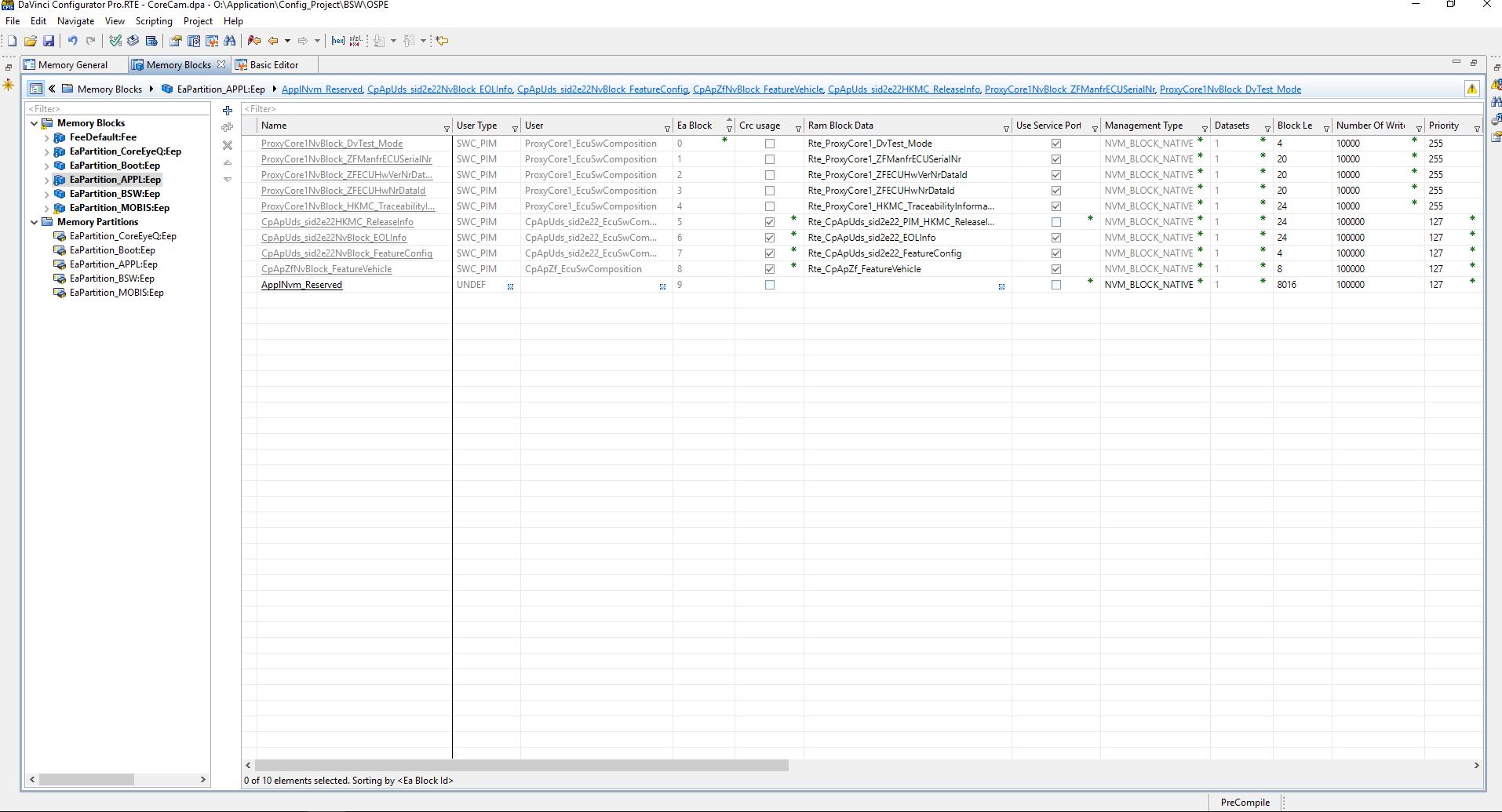
In addition, different flash Vendor will have different interpretation on device endurance as shown above, some is using ‘per page’ as unit will some is ‘any sector’.

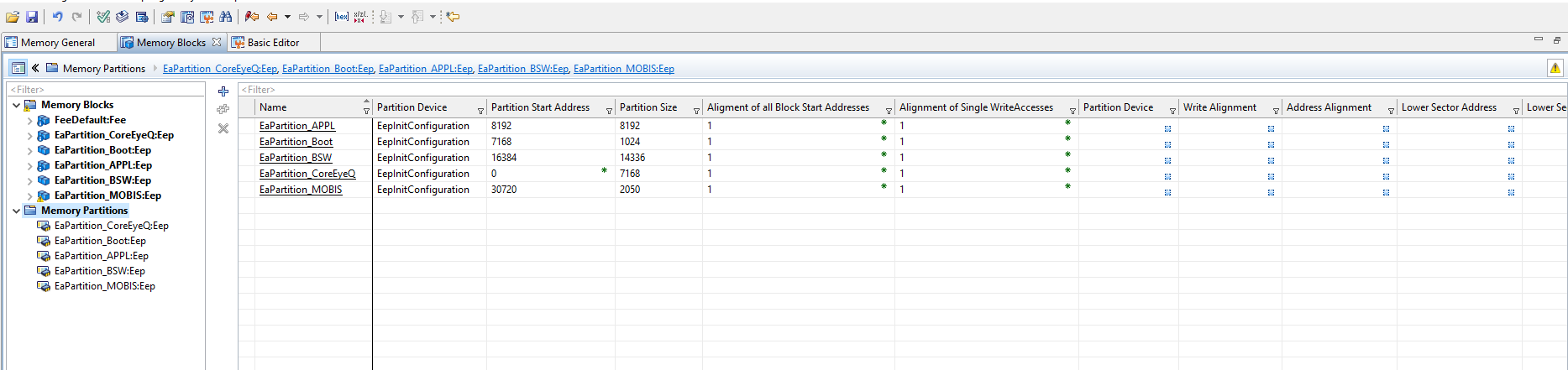




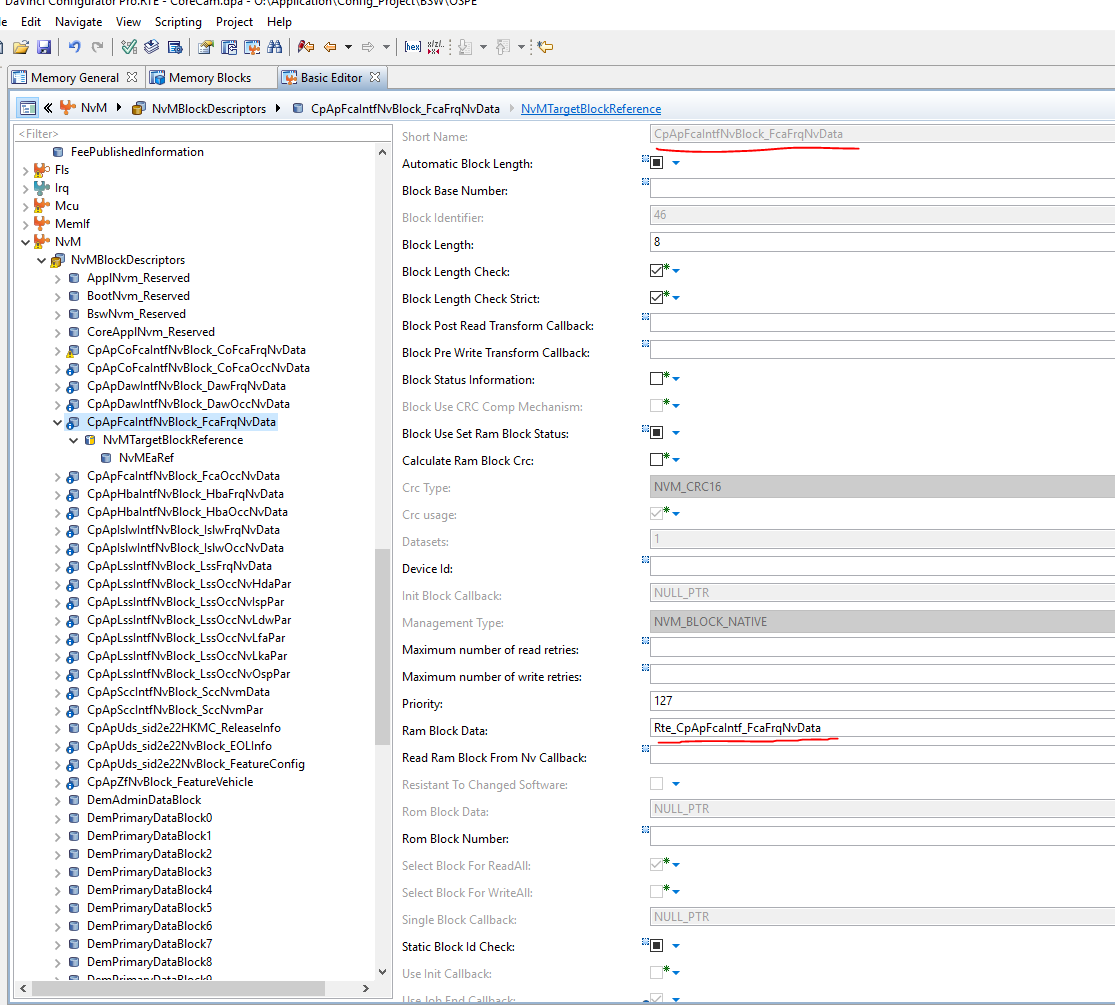


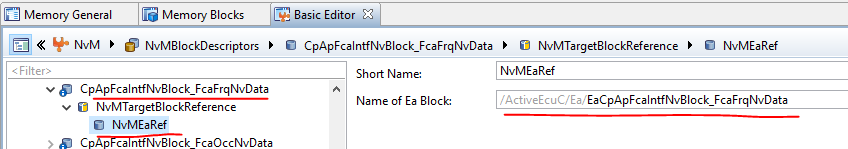




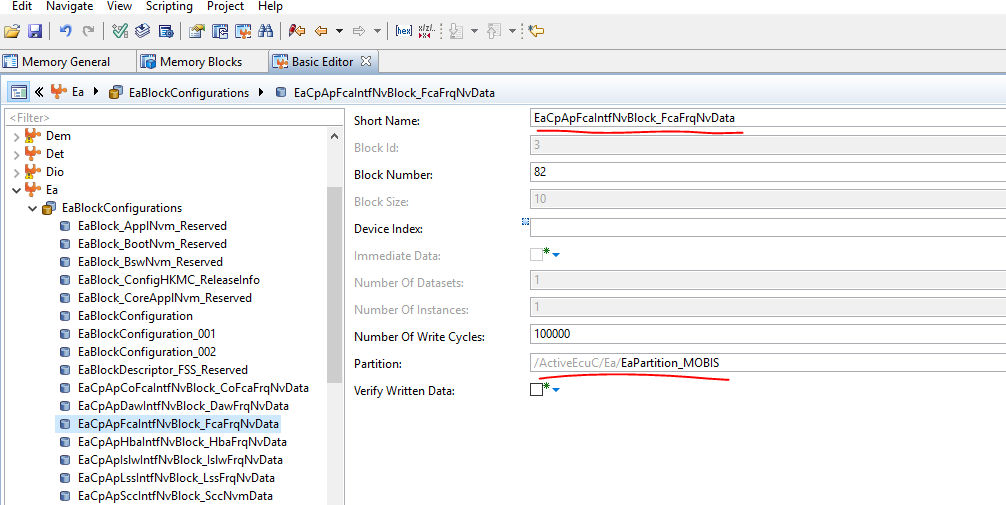


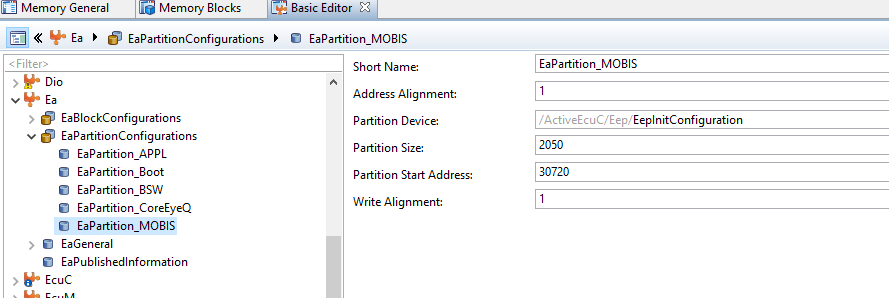
1. NVM-EA Mapping

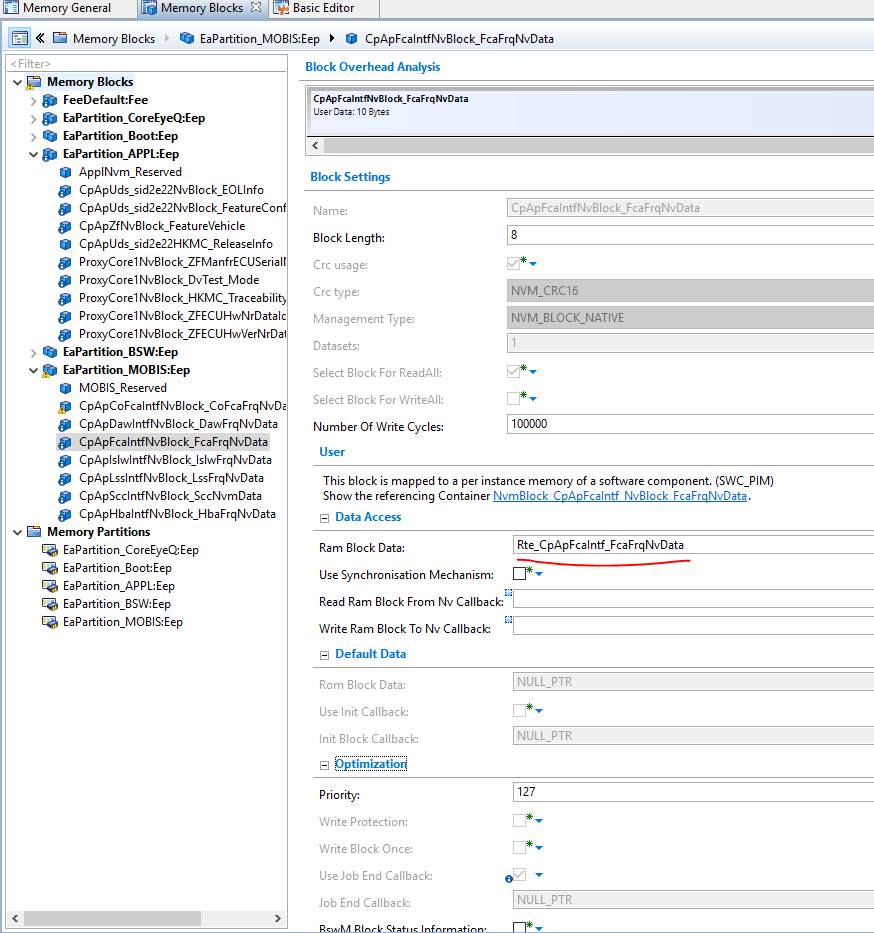




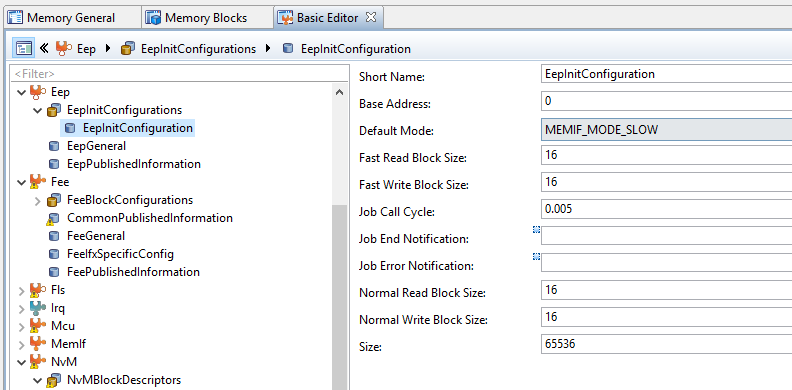
EA-EEP Mapping



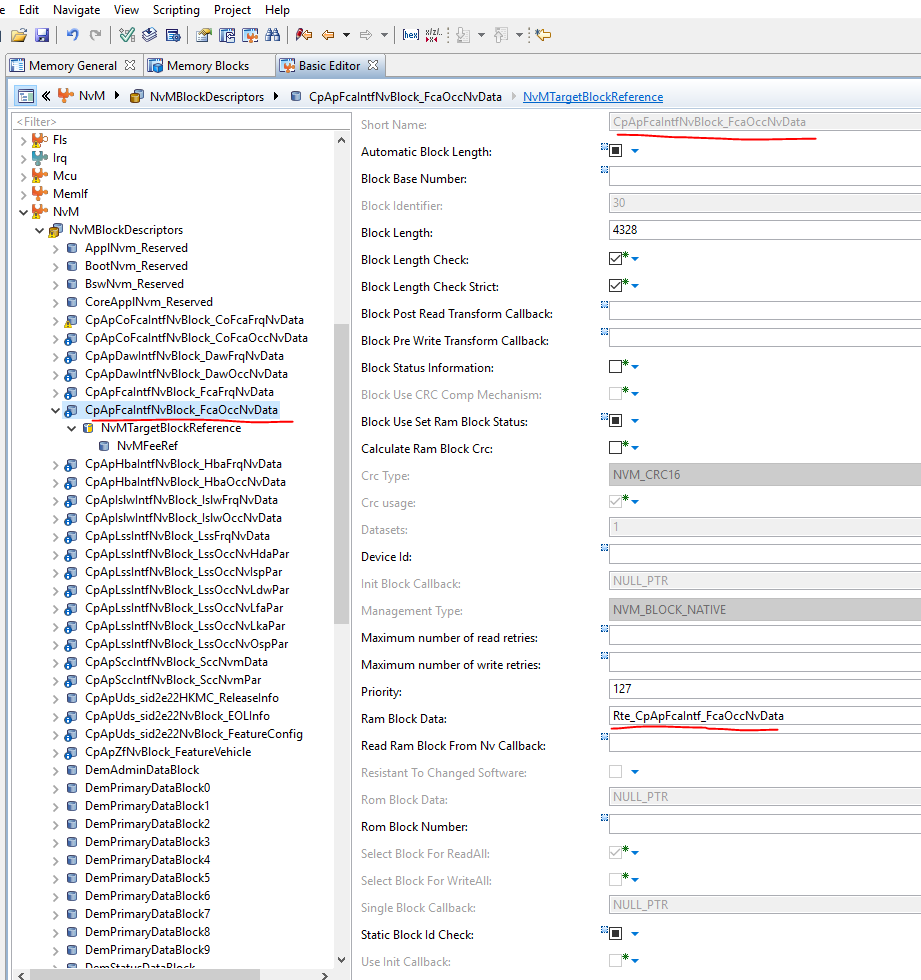


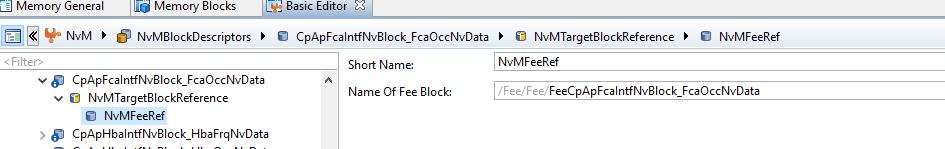


EEP

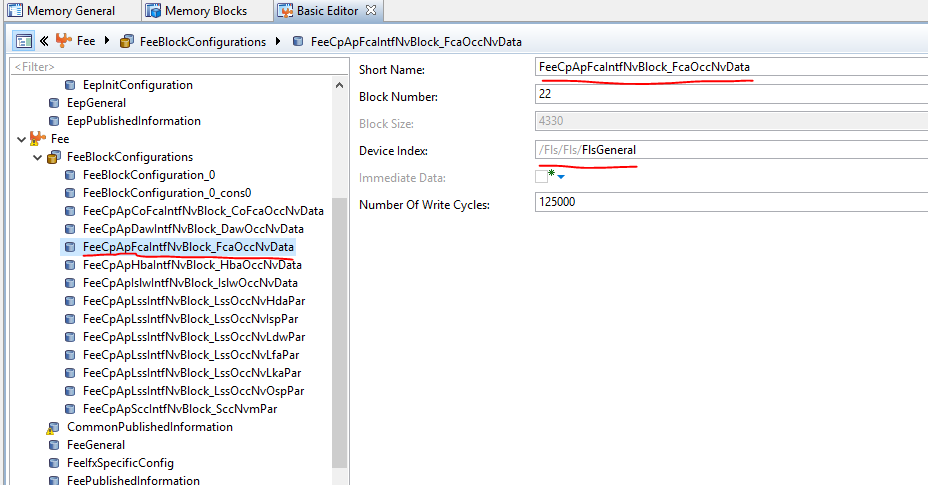


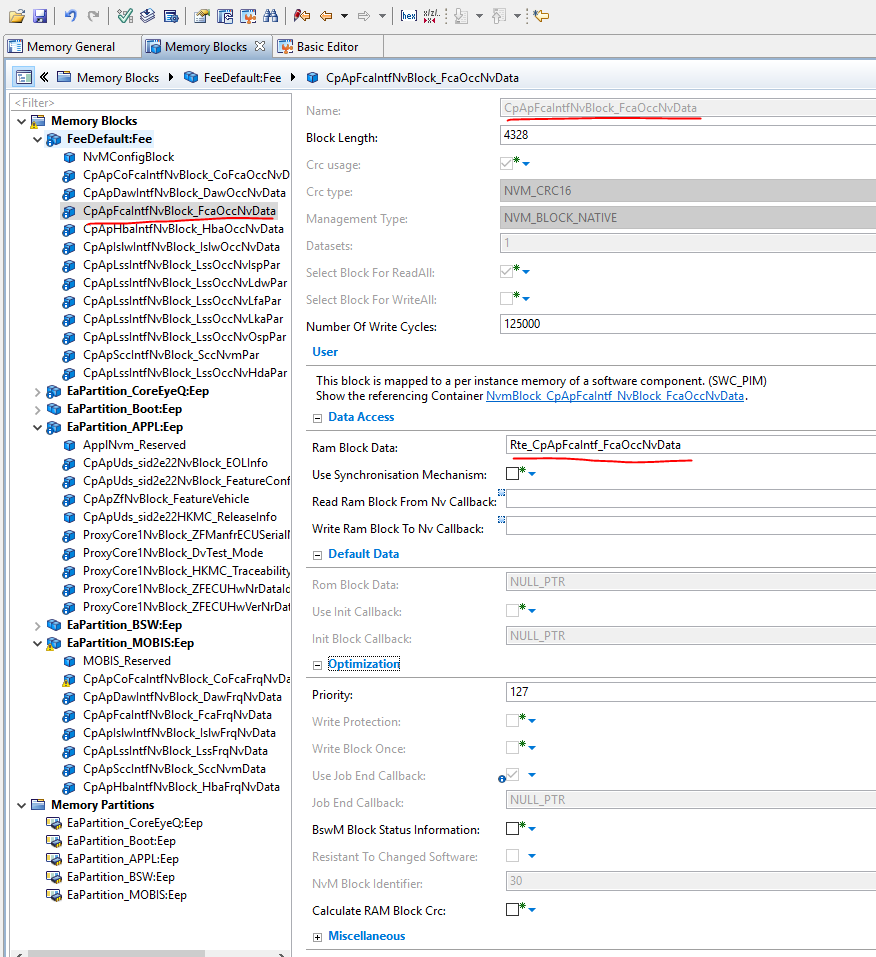
1. NVM-FEE Mapping



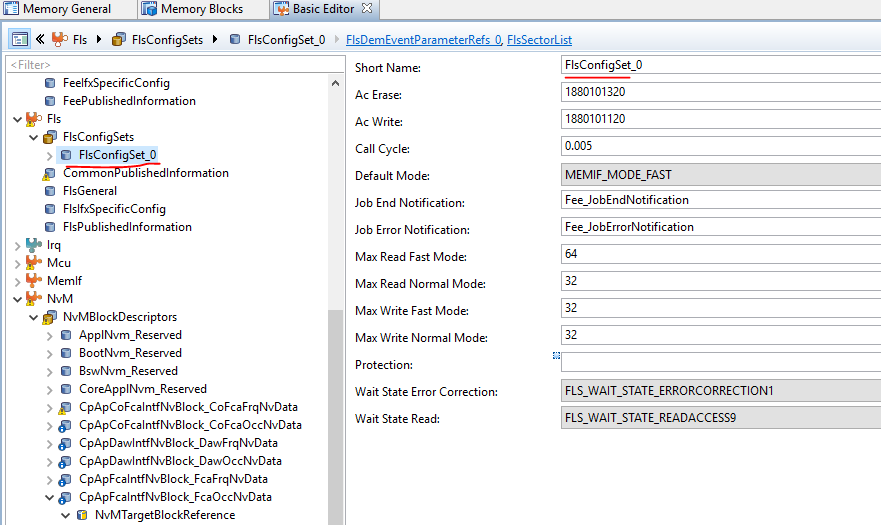


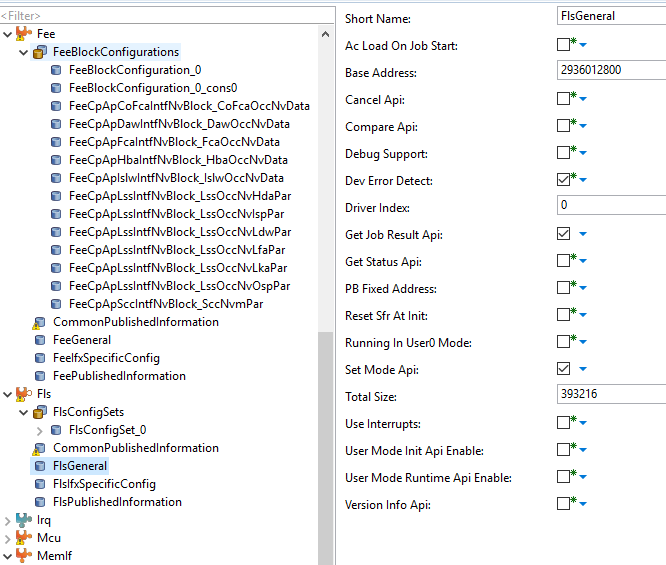
FEE-FLS Mapping

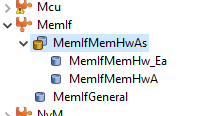
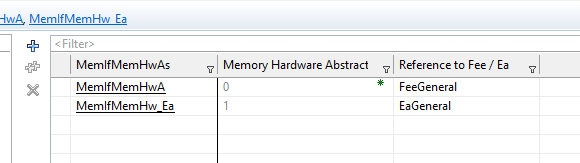




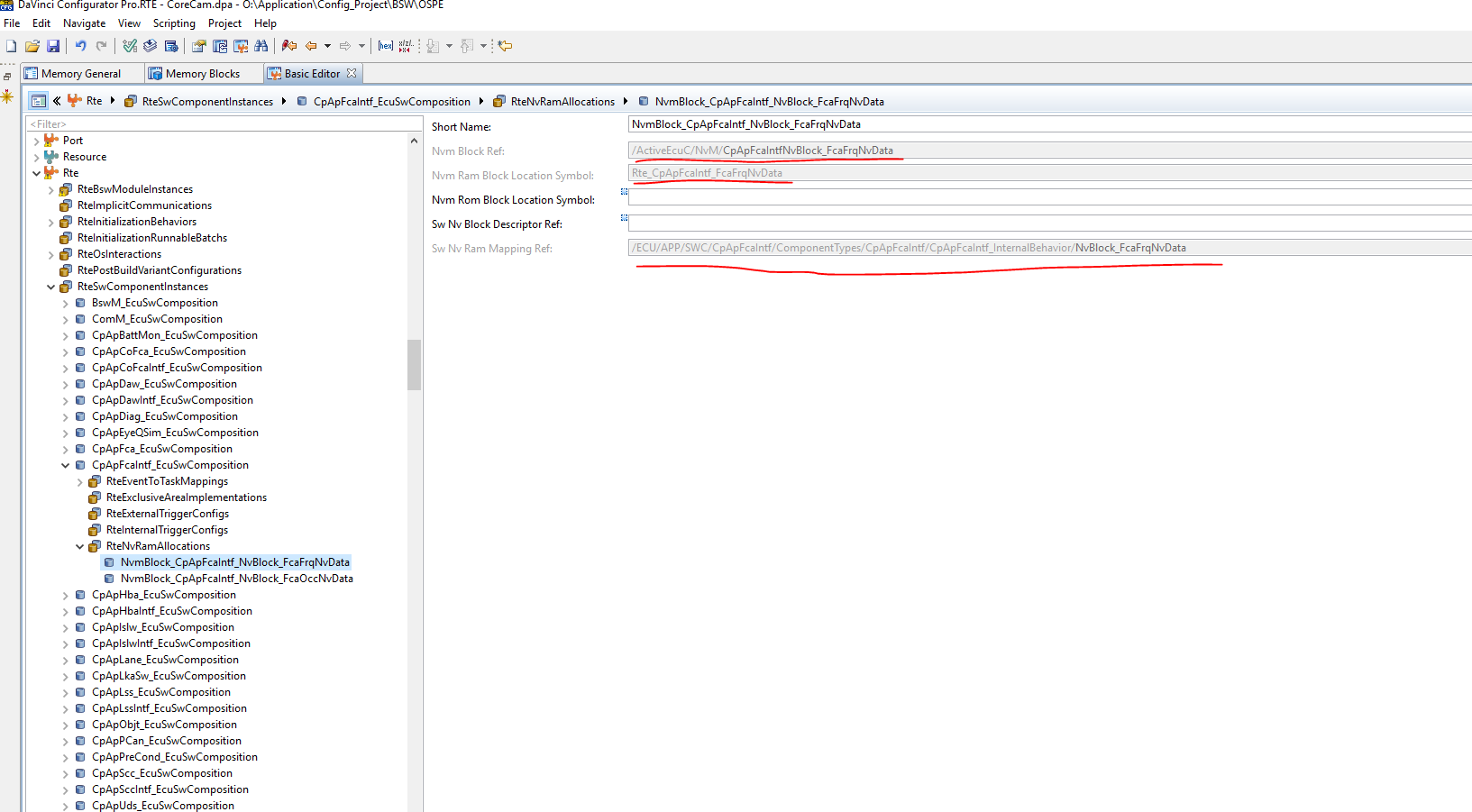
FLS

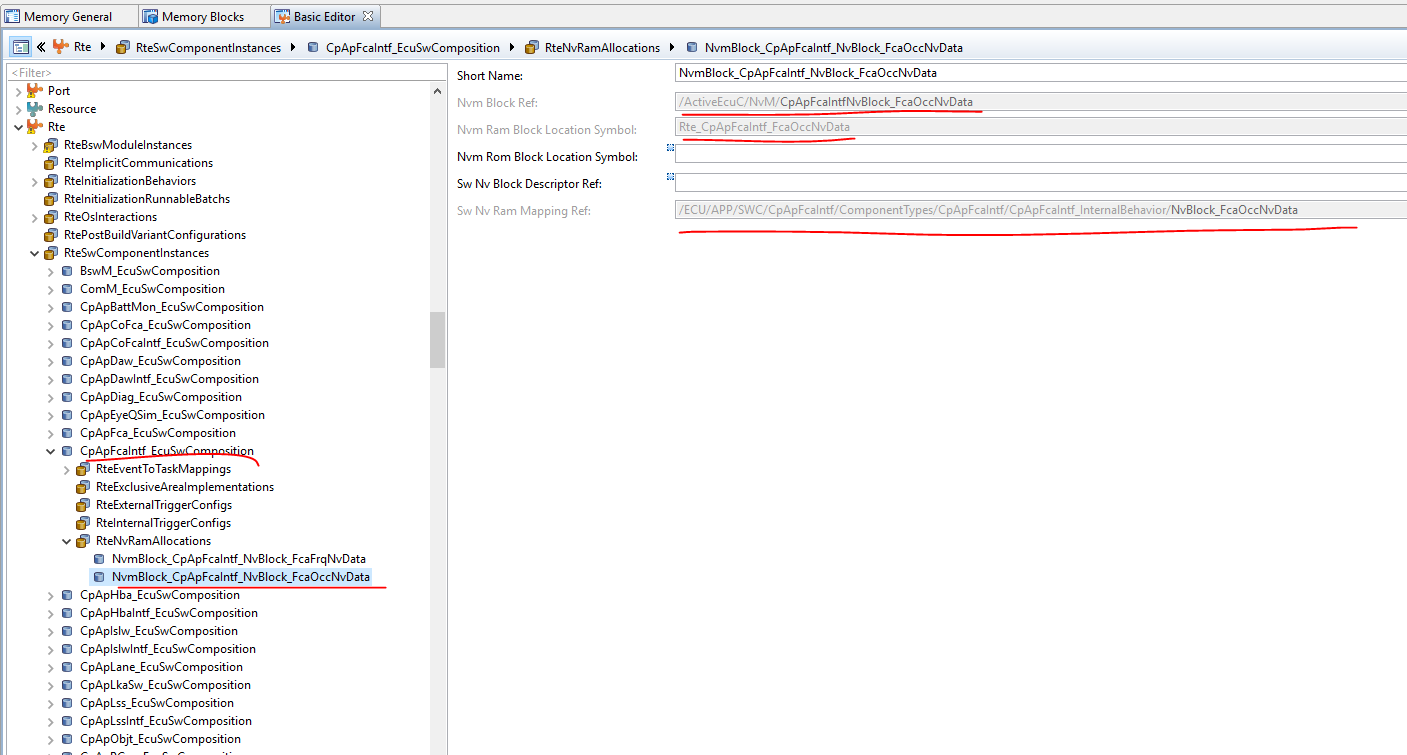




SWC-RTE-NVM Mapping





SW Component Mapping

