F2837xS Flash API documentation is not available yet. TI will provide the documentation soon. Until then, SPNU595 can be used to get the Flash API usage guidelines. F2837xS devices need F021_API_F2837xS_FPU32.lib provided in Controlsuite to program/erase the Flash of F2837xS devices.

Below are some notes that user should know along with the information provided in SPNU595 to use the Flash API for F2837xS devices:

1. EALLOW has to be executed before using Flash API functions for C28x devices.

2. Flash bank-width in F2837xS is 128 data-bits+ 16 ECC-bits.

3. Fapi_CalculateEcc function needs a byte address (left shift the C28x Flash address by 1 position before passing it to this function).

4. A note regarding Fapi_AutoEccGeneration mode usage: Fapi_AutoEccGeneration mode will program the supplied data portion in Flash along with automatically generated ECC. The ECC is calculated for the data width of the bank and data not supplied is treated as 0xFFFF. Note that there are practical implications of this when writing a custom programming utility that streams in the output file of a code project and programs the individual sections one at a time into flash. If a 64-bit word spans more than one section (i.e., contains the end of one section, and the start of another), values of 0xFFFF cannot be assumed for the missing data in the 64-bit word when programming the first section. When you go to program the second section, you will not be able to program the ECC for the first 64-bit word since it was already (incorrectly) computed and programmed using assumed 0xFFFF for the missing values. One

way to avoid this problem is to align all sections linked to flash on a 64-bit boundary in the linker command file for your code project.

For example, like this:

```
SECTIONS
{
.text : > FLASH, PAGE = 0, ALIGN(4)
.cinit : > FLASH, PAGE = 0, ALIGN(4)
.const : > FLASH, PAGE = 0, ALIGN(4)
.econst : > FLASH, PAGE = 0, ALIGN(4)
.pinit : > FLASH, PAGE = 0, ALIGN(4)
.switch : > FLASH, PAGE = 0, ALIGN(4)
}
```

If you do not align the sections in flash, you would need to track incomplete 64-bit words in a section and combine with the words in other sections that complete the 64-bit word. This will be difficult to do so it is recommended to align your sections on 64-bit boundaries.

5. Fapi_UserDefinedFunctions.c file at *C:\ti\controlSUITE\device_support\F2837xS\vx\F2837xS_examples_Cpu1\flash_programming\cpu01* is updated (compared to that of F2837xD and F28M35x/36x APIs). Two of the functions in this file are now moved to the library itself. Hence, users will find only one user-definable function in the latest file.

6. Fapi_getDeviceInfo() and Fapi_getBankSectors() functions are not supported anymore (compared to that of F2837xD and F28M35x/36x APIs).

7. *Helpers.h* file is removed from *FlashAPI* headers folder intentionally (compared to that of F2837xD and F28M35x/36x APIs). It is not needed.

8. Switch the PUMP SEMAPHORE between Bank0 and Bank1 for Bank0 and Bank1 Flash program/erase operations respectively.

9. Call Fapi_initializeAPI(F021_CPU0_W0_BASE_ADDRESS, User_configured_frequency) and Fapi_setActiveFlashBank(Fapi_FlashBank0) functions before doing Flash operations on Bank0.

10. Call Fapi_initializeAPI(F021_CPU0_W1_BASE_ADDRESS, User_configured_frequency) and Fapi_setActiveFlashBank(Fapi_FlashBank1) functions before doing Flash operations on Bank1.