Programming eMMC Flash Memories using Elnec Device programmers

Application note

July 2025
an_elnec_emmc, version 1.09

Disclaimer:

This application note describes how to program eMMC devices using Elnec device programmers. Before reading this document, user should be familiarized with eMMC devices. There are plentiful sources available through the web containing detailed informations about eMMC internal organization, features in eMMC, etc. Study, please, your device datasheet thoroughly, at least.

This application note is provided by our technical support department to help our customers and is provided "as-is", without warranty of any kind, either expressed or implied. We reserve the rights to make changes to the information available in this application note at any time and assume no liability for applications assistance, customer product design and any damages arising from the use of this application note.

This application note may refer to various products and brand names that may be claimed as property of their respective owners.

1. eMMC Partitions

The memory configuration initially consists of the User Area, two Boot Areas and RPMB Area partition. The sizes and attributes of Boot Areas and RPMB Area are defined by the memory manufacturer. The sizes and attributes of General Purpose Area Partitions 1-4(GP0-GP3) and Enhanced User Data Area can be programmed by the host setting the corresponding values in the Extended CSD registers.

Extended CSD bytes for configuring the parameters of General Purpose Area Partitions and Enhanced User Data Area:

GP_SIZES_MULT_GP0 – the size for Partition 1 area
GP_SIZES_MULT_GP1 – the size for Partition 2 area
GP_SIZES_MULT_GP2 – the size for Partition 3 area
GP_SIZES_MULT_GP3 – the size for Partition 4 area
ENH_SIZE_MULT – the size for Enhanced User Data area (pSLC)
ENH_START_ADDR – the device start address for Enhanced User Area
PARTITIONS_ATTRIBUTE – enhanced attribute (pSLC/MLC) for Partition1-3, Enhanced User Area
EXT_ PARTITIONS_ATTRIBUTE – extended attribute for Partition 1-4
PARTITION_SETTING_COMPLETED – mandatory for setting successfully completed

1.1. Example of configuration setting for Partitions 1-4 (MLC mode)

Requirements eMMC device configuration:

PARTITION 1(GP0) size = 1000000h (MLC)
PARTITION 2(GP1) size = 2000000h (MLC)
PARTITION 3(GP2) size = 6000000h (MLC)
PARTITION 4(GP3) size = 0000000h (MLC)
Enhanced UserArea size = 00h (none)
Enhanced UserArea Device Start/offset = 00h (none)

HC_WP_GRP_SIZE = (**value from the vendor datasheet**)
HC_ERASE_GRP_SIZE = (**value from the vendor datasheet**)
GP_SIZE_MULT_GPx = Partition size / (HC_WP_GRP_SIZE x HC_ERASE_GRP_SIZE x 80000h)

Example settings of Extended CSD register in the PG4UW/device:

GP_SIZE_MULT_GP0 [145-143] = 02h
GP_SIZE_MULT_GP1 [148-146] = 04h
GP_SIZE_MULT_GP2 [148-146] = 0Ch
GP_SIZE_MULT_GP3 [151-149] = 00h
ENH_SIZE_MULT [142:140] = 0000h (none)
ENH_START_ADDR [139:136] = 0000h (none)
PARTITIONS_ATTRIBUTE [156] = 00h (MLC mode setting)
PARTITIONS_SETTING_COMPLETE [155] = 01h (mandatory for configuration)

1.2. Example of configuration setting for UserArea (pSLC mode)

Requirements eMMC device configuration:

PARTITION 1(GP0) size = 00h (none)
PARTITION 2(GP1) size = 00h (none)
PARTITION 3(GP2) size = 00h (none)
PARTITION 4(GP3) size = 00h (none)
Enhanced UserArea size = configure device to Full pSLC mode

MAX_ENH_SIZE_MULT = **value from the vendor data sheet**

Example settings of Extended CSD register in the PG4UW/device:

GP_SIZE_MULT_GP0 [145-143] = 0000h
GP_SIZE_MULT_GP1 [148-146] = 0000h
GP_SIZE_MULT_GP2 [148-146] = 0000h
GP_SIZE_MULT_GP3 [151-149] = 0000h
ENH_SIZE_MULT [142:140] = MAX_ENH_SIZE_MULT
ENH_START_ADDR [139:136] = 0000h
PARTITIONS_ATTRIBUTE [156] = 01h (UserArea pSLC mode setting)
PARTITIONS_SETTING_COMPLETE [155] = 01h (mandatory for configuration)

2. Program Extended CSD register

To enable programming the Extended CSD register, open Device operation options (ALT+O) window and use check box “Extended CSD”, see figure 1. Setting this option is mandatory to allow programming selected bytes of Extended CSD register. For setting the individual bytes of Extended CSD register use the checkboxes in View/Edit Device settings (ALT+S) window, section "Extended CSD". During the "Program" operation only selected bytes of the Extended CSD register are programmed and verified. During operation of "Verify" all bytes of the Extended CSD register are verified.

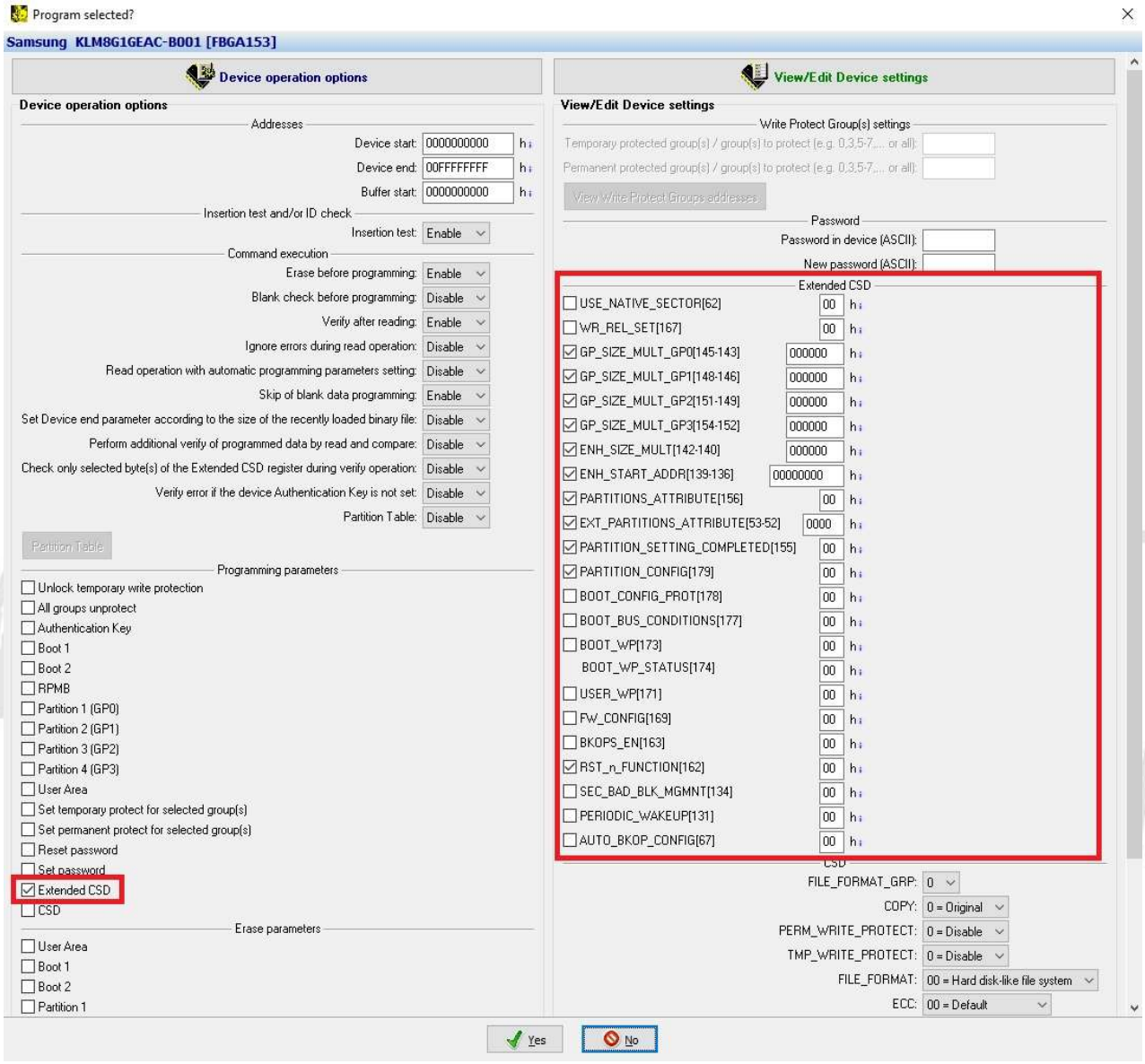

Figure 1. Enabling program Extended CSD register

2.1. Loading input data for Extended CSD register

For the loading Extended CSD register values you can use two ways. First, the user can view and edit the individual bytes of Extended CSD register via the View/Edit Device settings (Alt+S) window, section "Extended CSD", see figure 1. The Second way is import the data from the file(*.bin) to the buffer called “Extended CSD”, see figure 2. Both ways are equivalent and the setting of individual bytes is mirrored between the “Extend CSD” buffer and View/Edit Device settings (Alt+S) window.

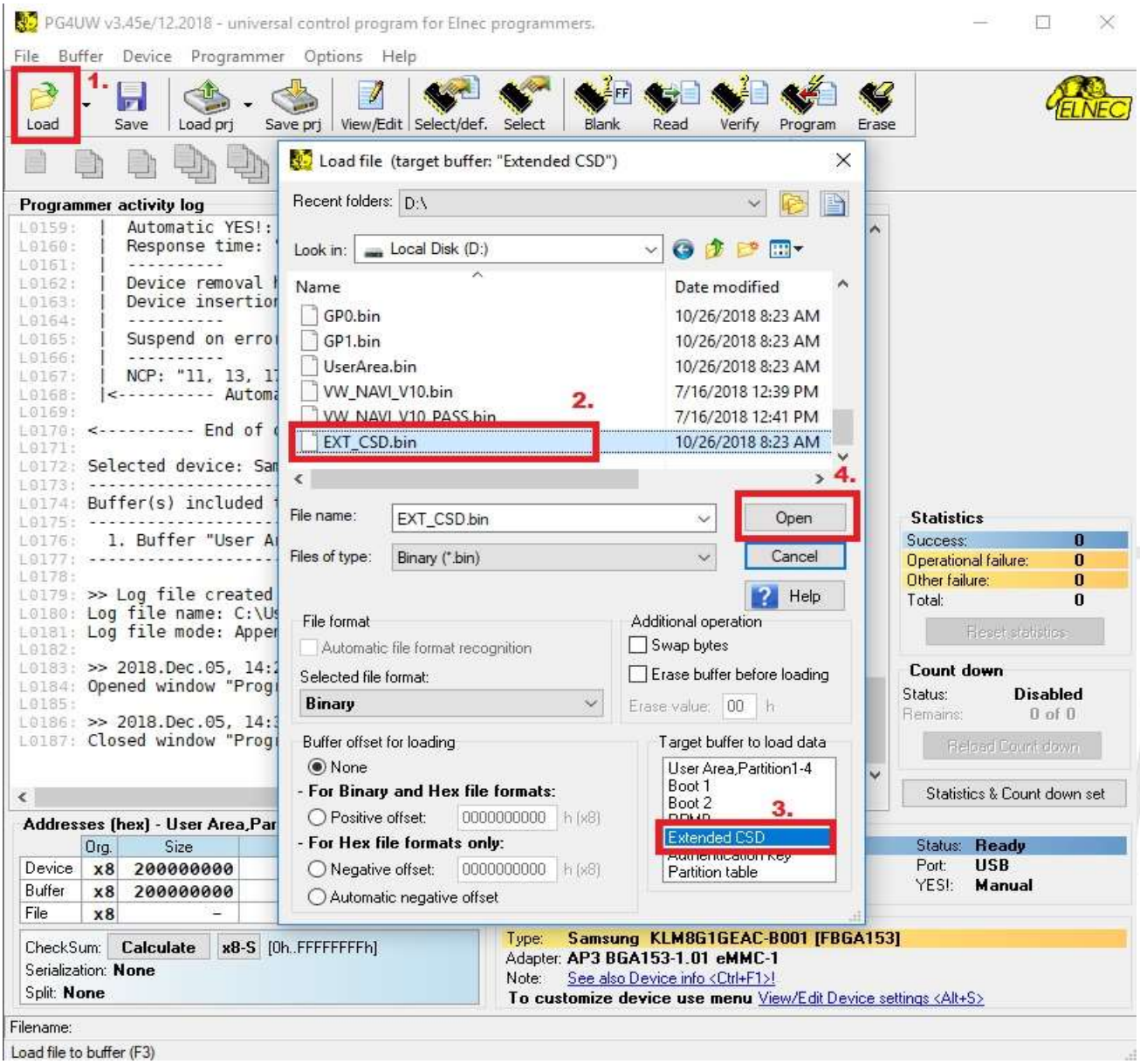

Figure 2. Load input file for Extended CSD register

3. Program Boot 1(or Boot 2)

To enable programming of Boot 1 (Boot 2) area, open the Device operation options (ALT+O) window and use check box “Boot 1”, see figure 3.

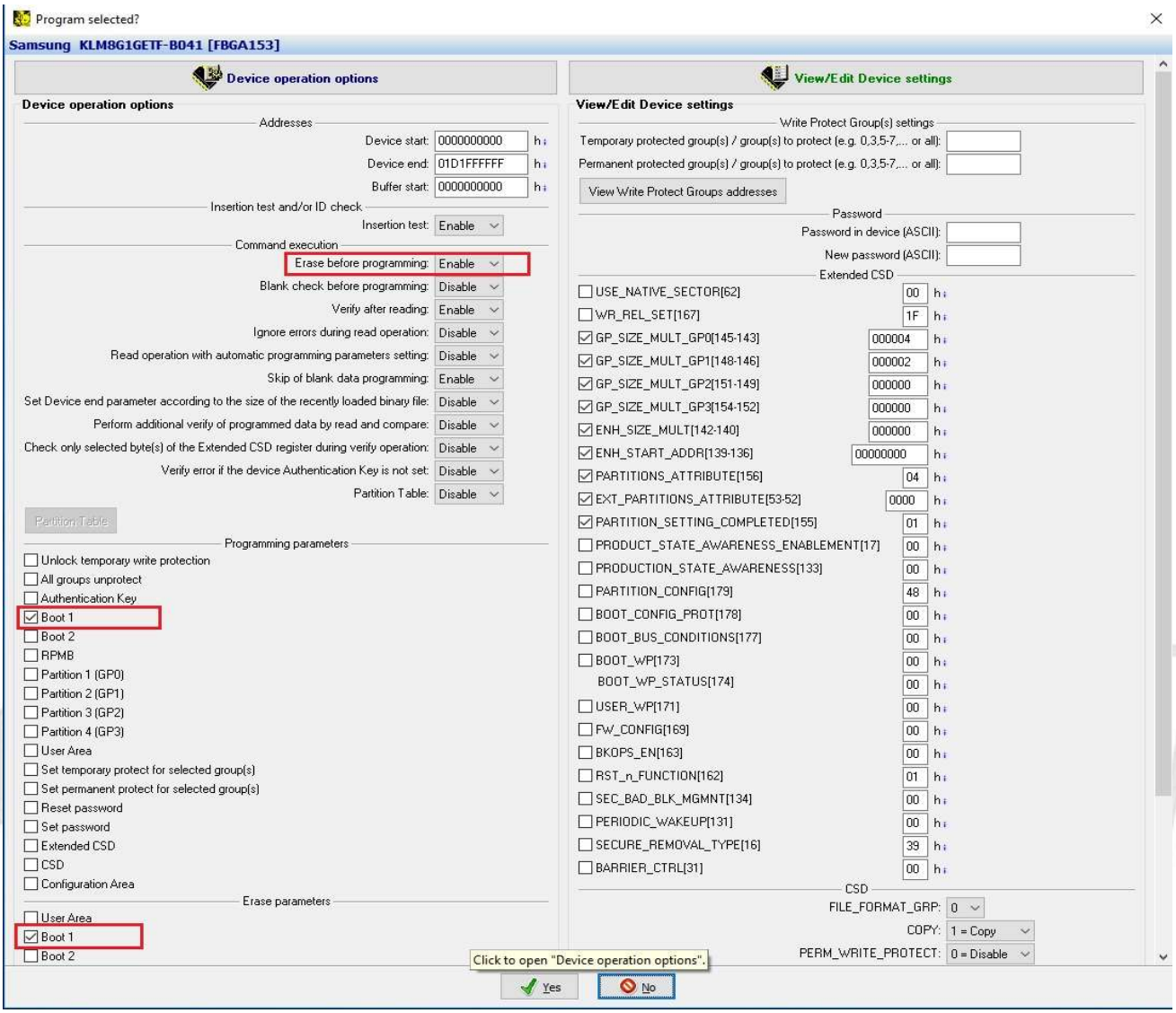

Figure 3. Enabling program Boot 1 area

3.1. Loading input data for Boot 1(or Boot 2)

The data for the Boot 1 (Boot 2) area must be stored in a separate *.bin file. For loading Boot 1 (Boot 2) data open the “Load file” window, select the appropriate target buffer to Boot 1 (Boot 2) and open *.bin file, see figure 4. The loaded data can be viewed in View/Edit Buffer window, Boot 1 (Boot 2) buffer tab.

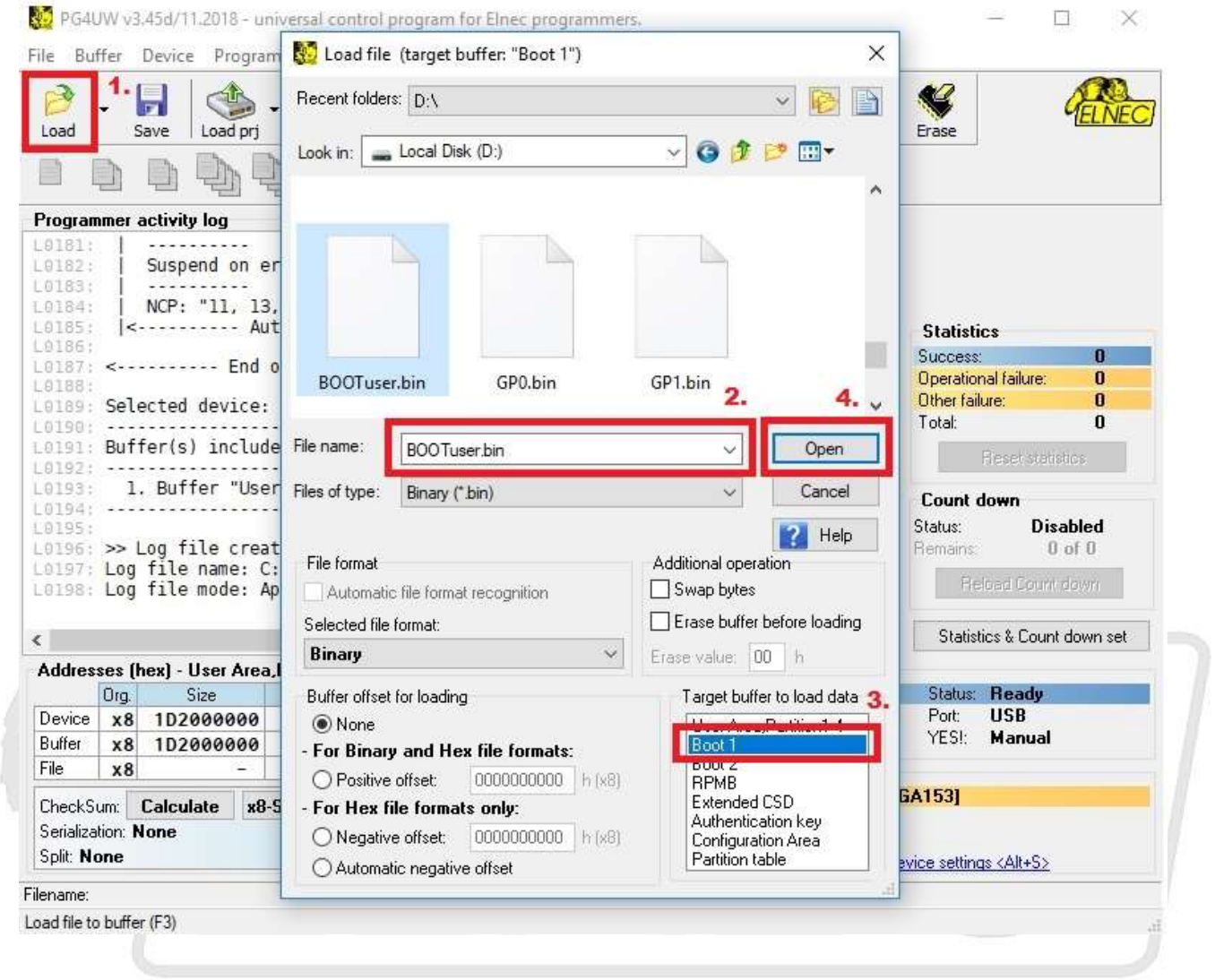

Figure 4. Load input file for BOOT 1 area

4. Program User Area

To enable programming of the User Area, open the Device operation options (ALT+O) window and tick the “User Area” check box. If the status of Command execution option "Partition table" is disable then address range for the User Area can be set in the Operation options window, section "Addresses", see figure 5. If the option "Partition table" is enabled then the address range of User Area is defined by the Partition table.

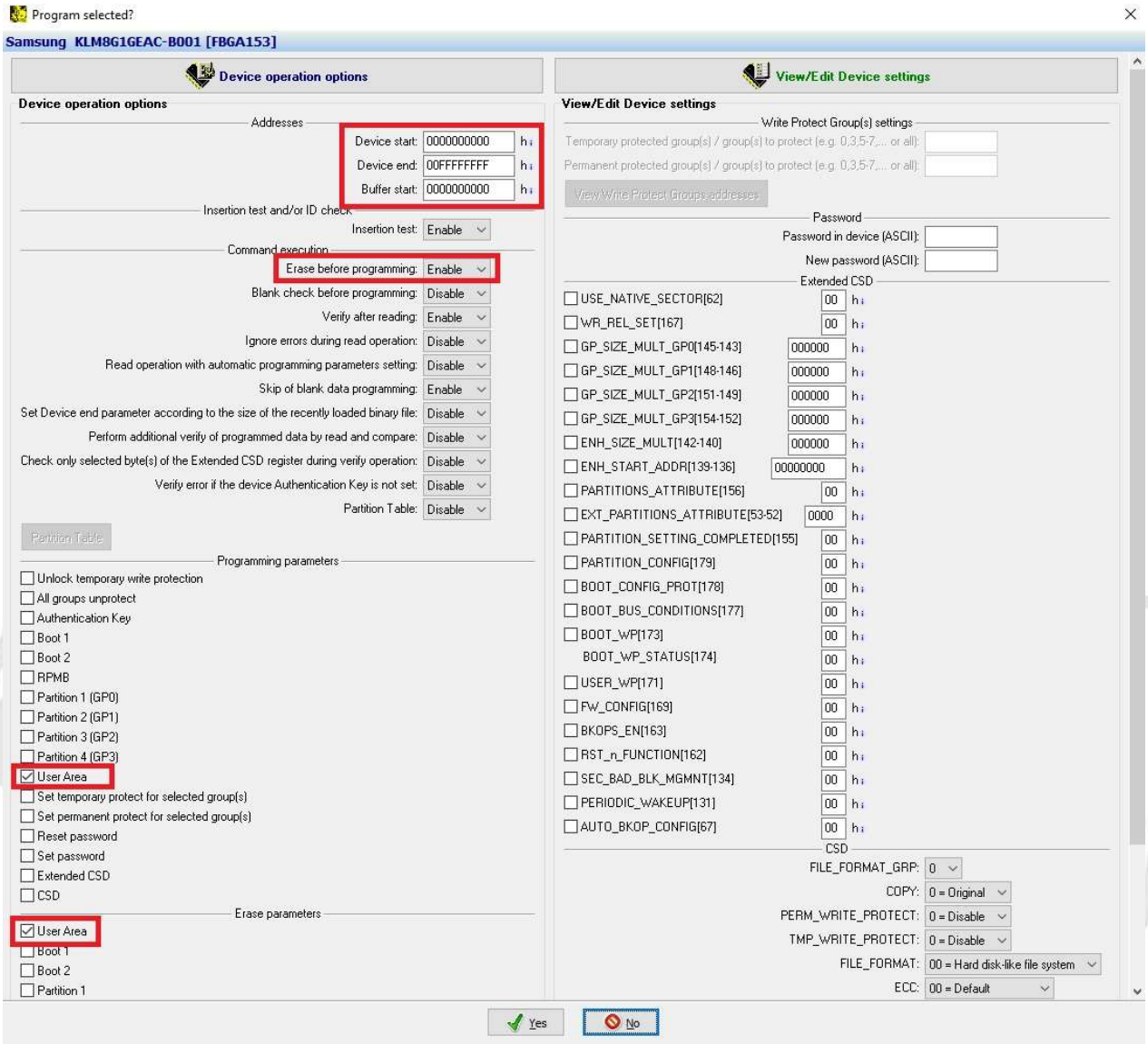

Figure 5. Enabling program User Area

4.1. Loading input data for User Area

The data for the User Area area must be stored in a separate *.bin file. For loading data of the User Area, open “Load file” window, select the appropriate target buffer to User Area, Partition1-4 and open *.bin file, see figure 6. The loaded data can be viewed in View/Edit Buffer window, “User Area,Partition1-4” buffer tab.

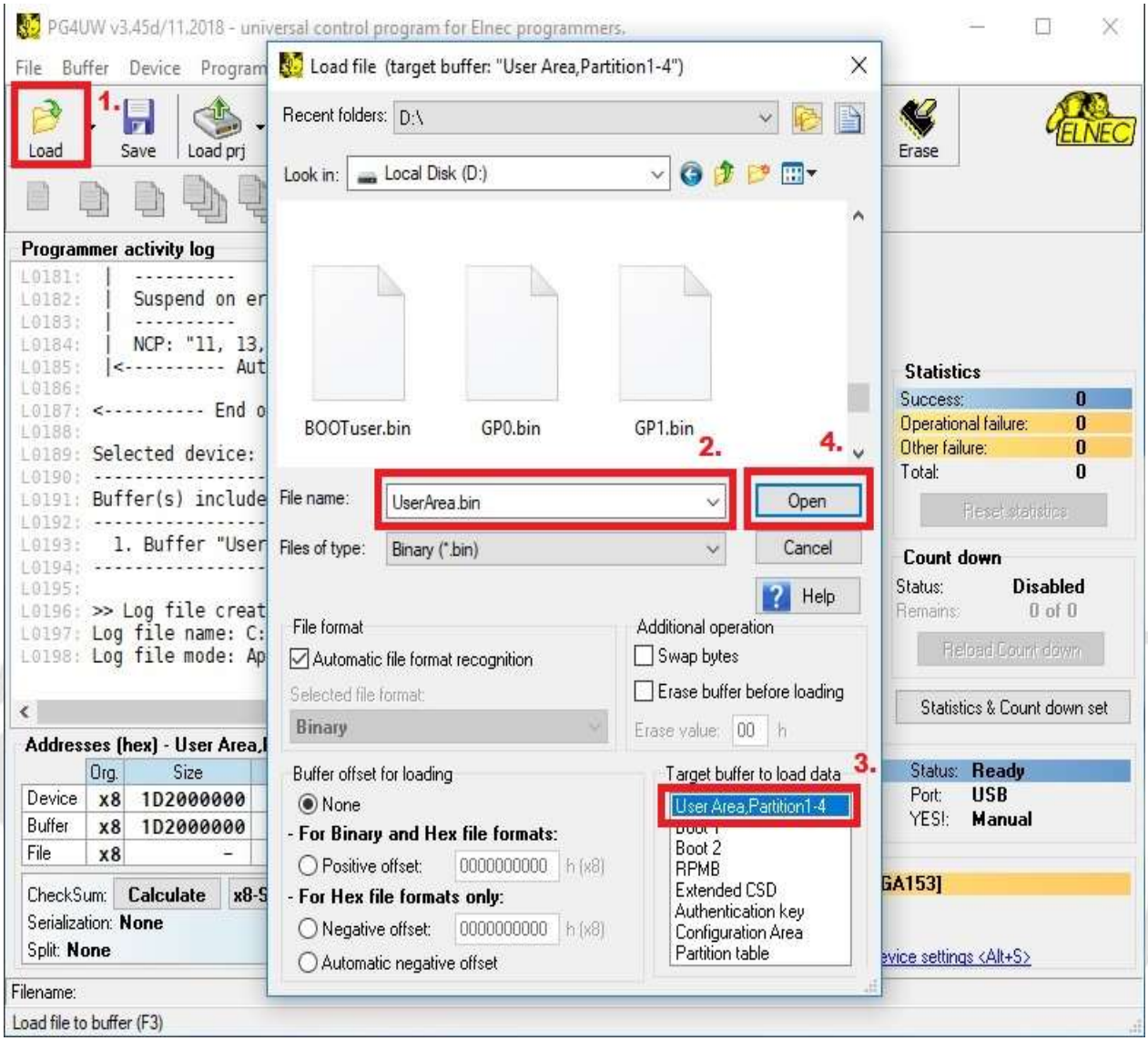

Figure 6. Load input file for User Area

4.2. Loading input data for Enhanced User Area

The “Enhanced User Data Area” occupy address range within the “User data area” (there is no obligatory address gap between the “Enhanced User Data Area” and the rest of the “User data area”).The start address(sector address) of the “Enhanced User Data Area” is defined by the ENH_START_ADDR[139-136] location in the Extended CSD register (see Alt+S window)

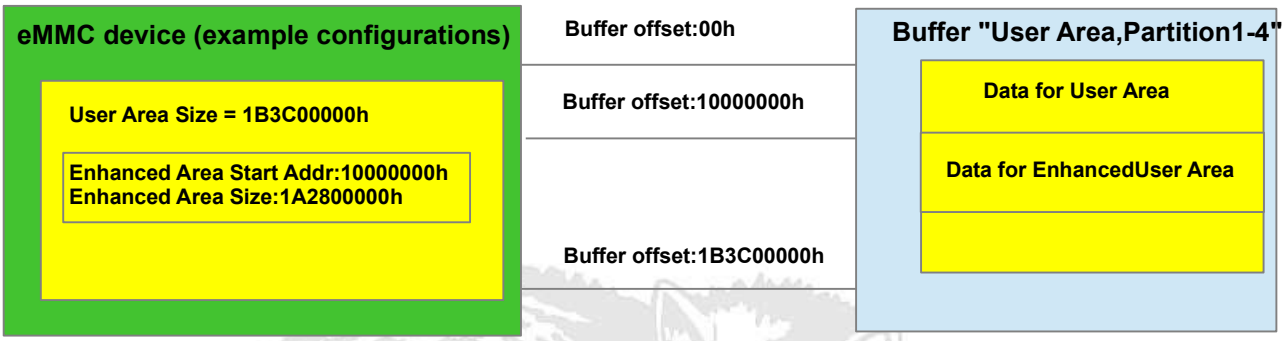

Figure 7.Example configuration or Enhanced User Area

Note: The “Enhanced User Data Area” start address is in byte units, i.e. start address(in bytes) = ENH_START_ADDR[139-136] x SECTOR_SIZE(200h).

For loading data to the Enhanced User Area, open Load file window, set buffer offset, choose the “User Area,Partition1-4” target buffer and open specified *.bin file (see figure 8). It is necessary to enter the correct **positive offset** and then load the *.bin file. The loaded data can be viewed in View/Edit Buffer window, “User Area,Partition1-4” buffer tab.

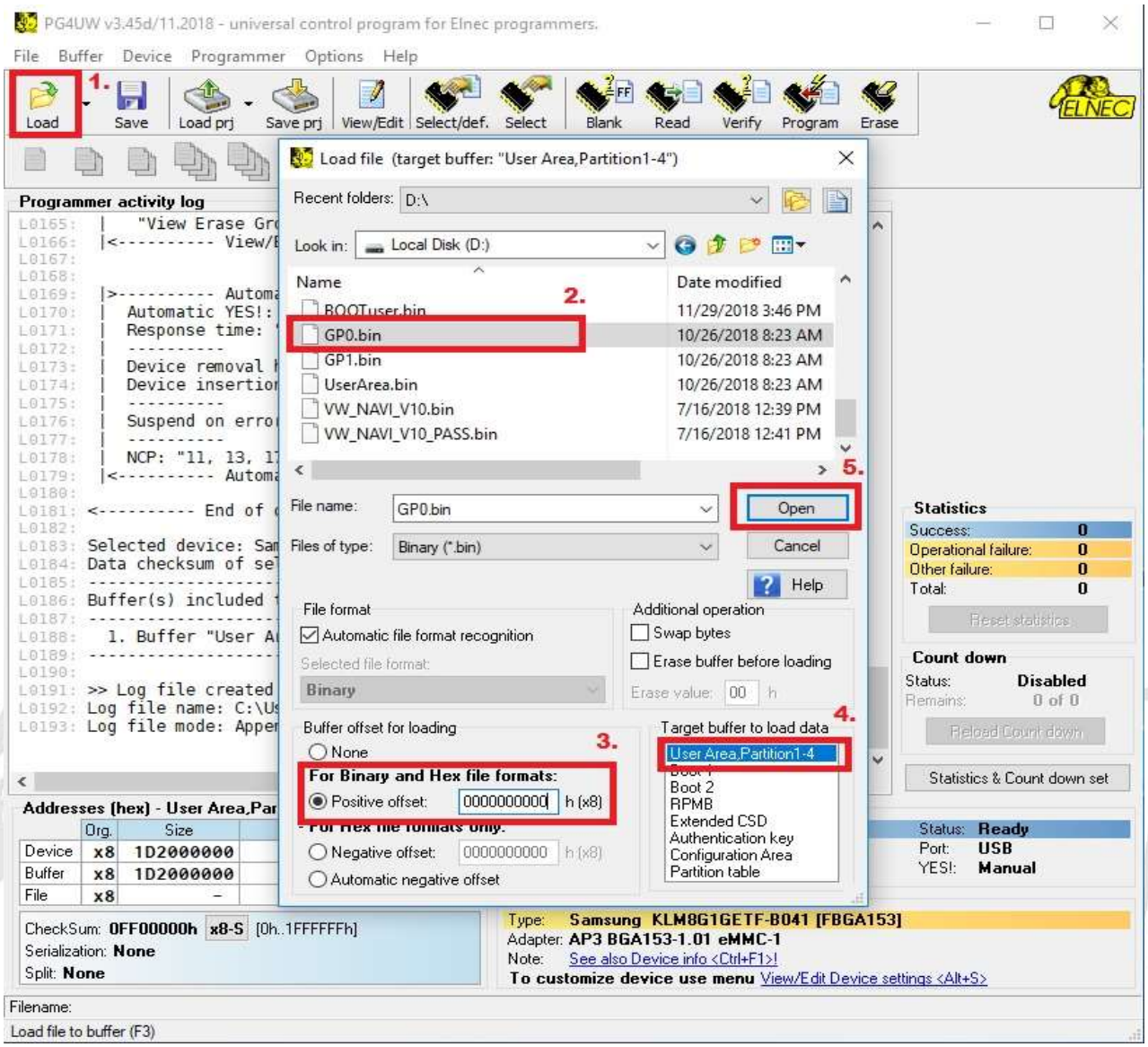

Figure 8.Load input file for Enhanced User Area

Note: When load the *bin file for “Enhanced User Data Area” use positive offset for start address in bytes, i.e. ENH_START_ADDR[139-136] x SECTOR_SIZE(200h).

5. Program Partition 1-4

To enable programming of Partition 1-4, open the Device operation options (ALT+O) window and tick the “Partition 1-4” check box.

Data for the Partition 1-4 must be loaded in the "User Area,Partition1-4" buffer. For loading data to the Partition 1-4, open Load file window, set buffer offset, choose the “User Area,Partition1-4” target buffer and open specified *.bin file(see figure 9). It is necessary to enter the correct **positive offset** and then load the *.bin file. The loaded data can be viewed in View/Edit Buffer window, “User Area,Partition1-4” buffer tab.

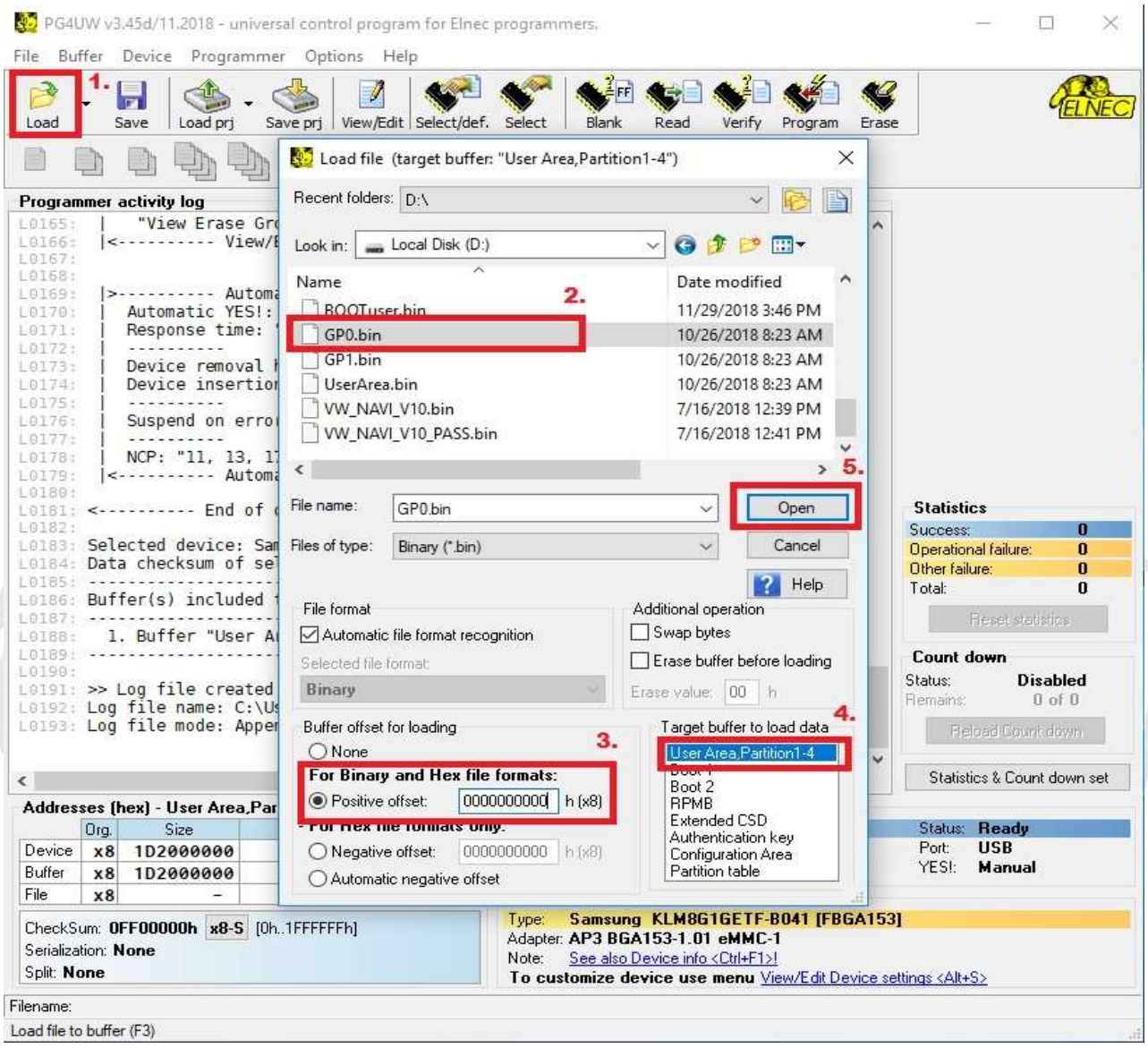

Figure 9. Load input file for Partition 1-4

Note: Positive offset = for more information see section 5.1, 5.2

5.1 Buffer map in case of disabled "Partition table" option

If the "Partition table" option (Device operation options windows(Alt+O), section Command execution) is disabled then is not possible to edit the Partition1-4 buffer map. SW will automatically change the buffer map in accordance with setting in the Extended CSD register(View/edit Device settings window Alt+S, section Extend CSD). Address ranges of the data for Partitions1-4(buffer "User Area, Partition1-4") are showing in the programmer activity log window during Read operation or suring programming these partitions. The loaded data for the Partition 1-4 must be placed in the "User Area,Partition1-4" buffer in accordance with example case in figure 10.

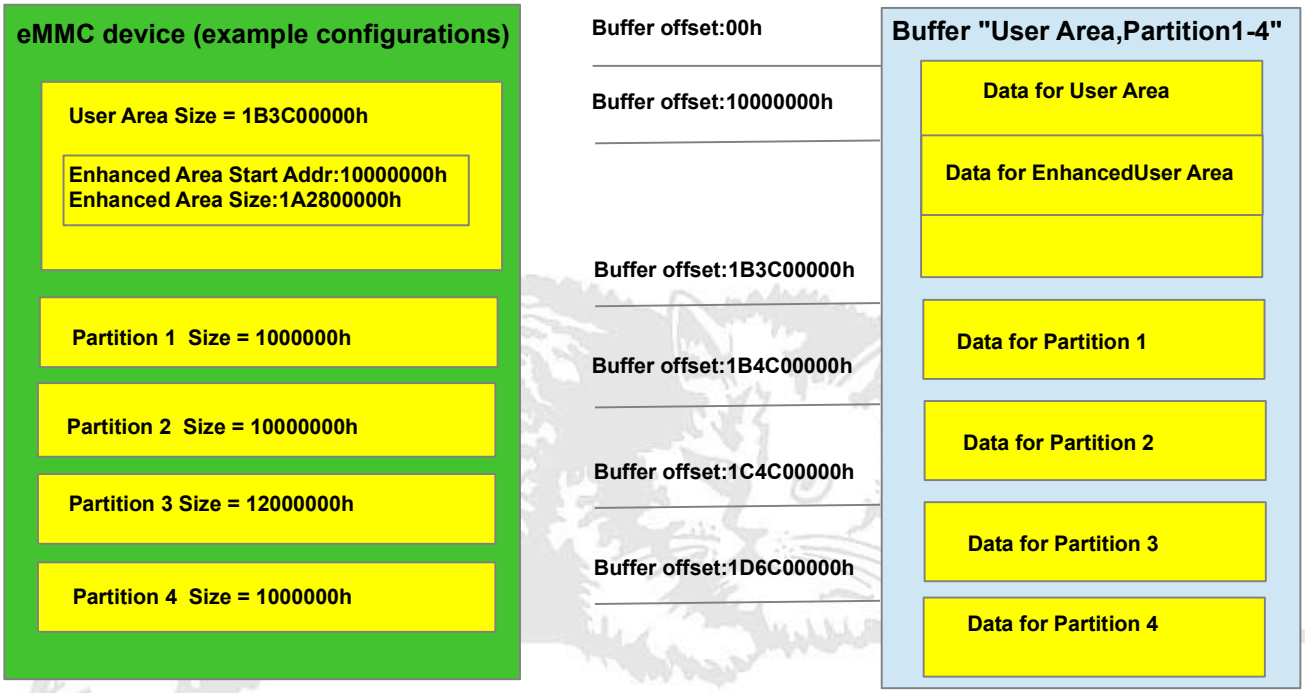

Figure 10. Example of "User Area,Partition1-4" buffer map when Partition table = DISABLE

5.1.1 Example of loading several *.bin files when the option "Partition table" is disabled

eMMC device configuration:
UserArea default size = 37A400000h
Enhanced UserArea start address = 10000000h
Enhanced UserArea Size = 1A2800000h(pSLC)
Partition1/GP0 size = 1000000h(pMLC)
Partition2/GP1 size = 10000000h(pMLC)
Partition3/GP2 size = 12000000h(pMLC)
Partition4/GP3 size = 1000000h(pMLC)
UserArea size = 37A400000h–1A2800000h-1000000h–10000000h-12000000h-1000000h = **1B3C00000h**

Programming files and destination:
1.DTIW381.bin to UserArea address 1 = 10000000h
2.TT_D_De.bin to UserArea address 2 = 10011000h
3.FSOS_Ex.bin to UserArea address 3 = 14011000h
4.DIW3870.bin to GP0 address = 00h
5.SC60401.bin to GP1 address = 00h
6.DIW3871.bin to GP3 address = 00h

Calculation of the buffer offset for loading files:
1.DTIW381.bin
Buffer **offset1** = UserArea address 1
= **10000000h**
2.TT_D_De.bin
Buffer **offset2** = UserArea address 2
= **10011000h**
3.FSOS_Ex.bin
Buffer **offset3** = UserArea address 3
= **14011000h**
4.DIW3870.bin
Buffer **offset4** = UserArea size + GP0 address
= 1B3C00000h + 00h
= **1B3C00000h**
5.SC60401.bin
Buffer **offset5** = UserArea size + GP0 size + GP1 address
= 1B3C00000h + 1000000h + 00h
= **1B4C00000h**
6.DIW3871.bin
Buffer **offset6** = UserArea size + GP0 size + GP1 size + GP2 size+ GP3 address
= 1B3C00000h + 1000000h + 10000000h + 12000000h + 00h
= **1D6C00000h**

Loading files sequence:
1.Clear buffer “User Area,Partition1-4”
2.Open Load file window, choose the “User Area,Partition1-4” target buffer,
set positive buffer **offset1** = **10000000h** and then load the **DTIW381.bin** file.
3.Open Load file window, choose the “User Area,Partition1-4” target buffer,
set positive buffer **offset2** = **10011000h** and then load the **TT_D_De.bin** file.
4.Open Load file window, choose the “User Area,Partition1-4” target buffer,
set positive buffer **offset3** = **14011000h** and then load the **FSOS_Ex.bin** file
5.Open Load file window, choose the “User Area,Partition1-4” target buffer,
set positive buffer **offset4** = **1B3C00000h** and then load the **DIW3870.bin** file.
6.Open Load file window, choose the “User Area,Partition1-4” target buffer,
set positive buffer **offset5** = **1B4C00000h** and then load the **SC60401.bin** file.
7.Open Load file window, choose the “User Area,Partition1-4” target buffer,
set positive buffer **offset6** = **1D6C00000h** and then load the **DIW3871.bin** file.
8.Save project

5.2 Buffer map in case of enabled "Partition table" option

If the option "Partition table" is enabled then the address ranges(buffer map) for Partition1-4 can be viewed and edited in the Partition Table window(see section 5.3 Partition table). The user load appropriate *bin files to the “User Area, Partition 1-4” buffer and fill the records in the Partition table window. The loaded *.bin files must be 512-byte sector aligned. For more information see section “5.2.1 Partition table”.

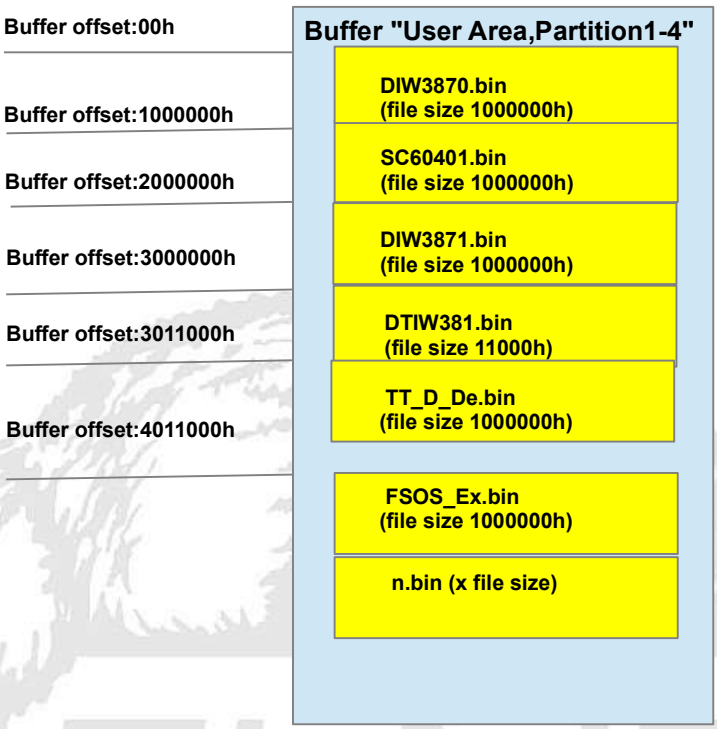

Figure 11. Example of "User Area,Partition1-4" buffer map when Partition table = ENABLE

Note: Loading the *.bin files without filling the Partition table records cause the omit of processing(skipping the operation of program/read/etc) “User Area, Partition 1-4” areas.

5.2.1 Partition table

Partition table records allow user to define address ranges for User Area and Partition 1-4. Each partition record is specified by four items Device_Start_Sec, Buffer_Start_Sec, Data_Size_Sec and Device_Location (4 bytes/32bits in size for each item). Every partition record have size of 16bytes. The data in the buffer "Partition Table" is stored using Little Endian byte order. The user can view and edit the individual records of Partition table using the “Partition table” window, see figure 12.

Partition table item meaning:

Device_Start_Sec – indicates start of the sector location for partition within the device.
Buffer_Start_Sec – indicates start of the sector location for partition within the buffer "User Area".
Data_Size_Sec – indicates the number of bytes for processing within the specified partition record
Device_Location – indicates the location of specified partition record. The lowest byte indicate wich physical partition it belongs to. The higher 3 bytes are not used.

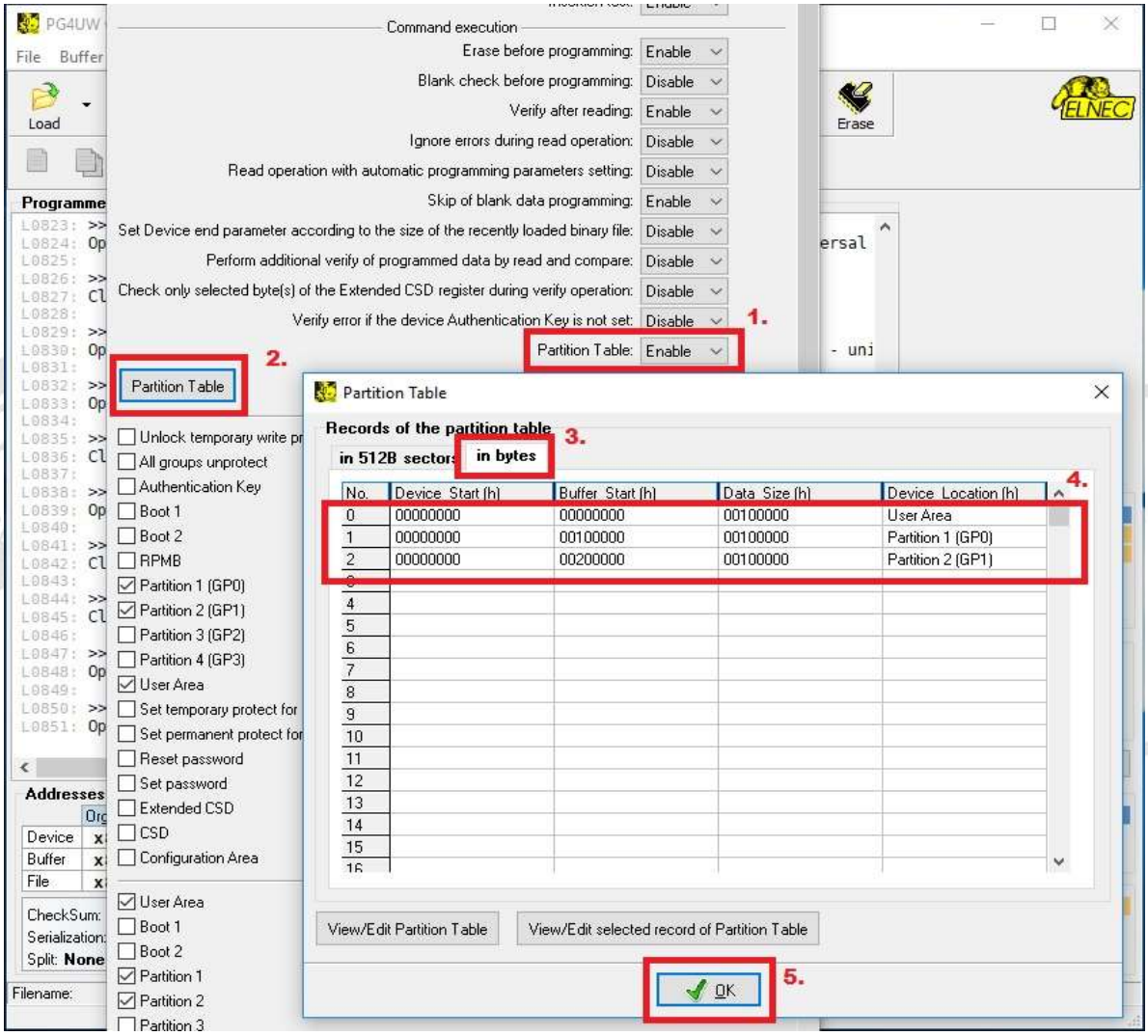

Figure 12. Partition table

The second way for import the data is loading the data from the file(*.bin) to the buffer called “Partition table”, see figure 13. The loaded data can be viewed in View/Edit Buffer window, “Partition table” buffer tab. Both ways are equivalent and the settings of individual records are mirrored between the “Partition table” buffer and the Partition table window

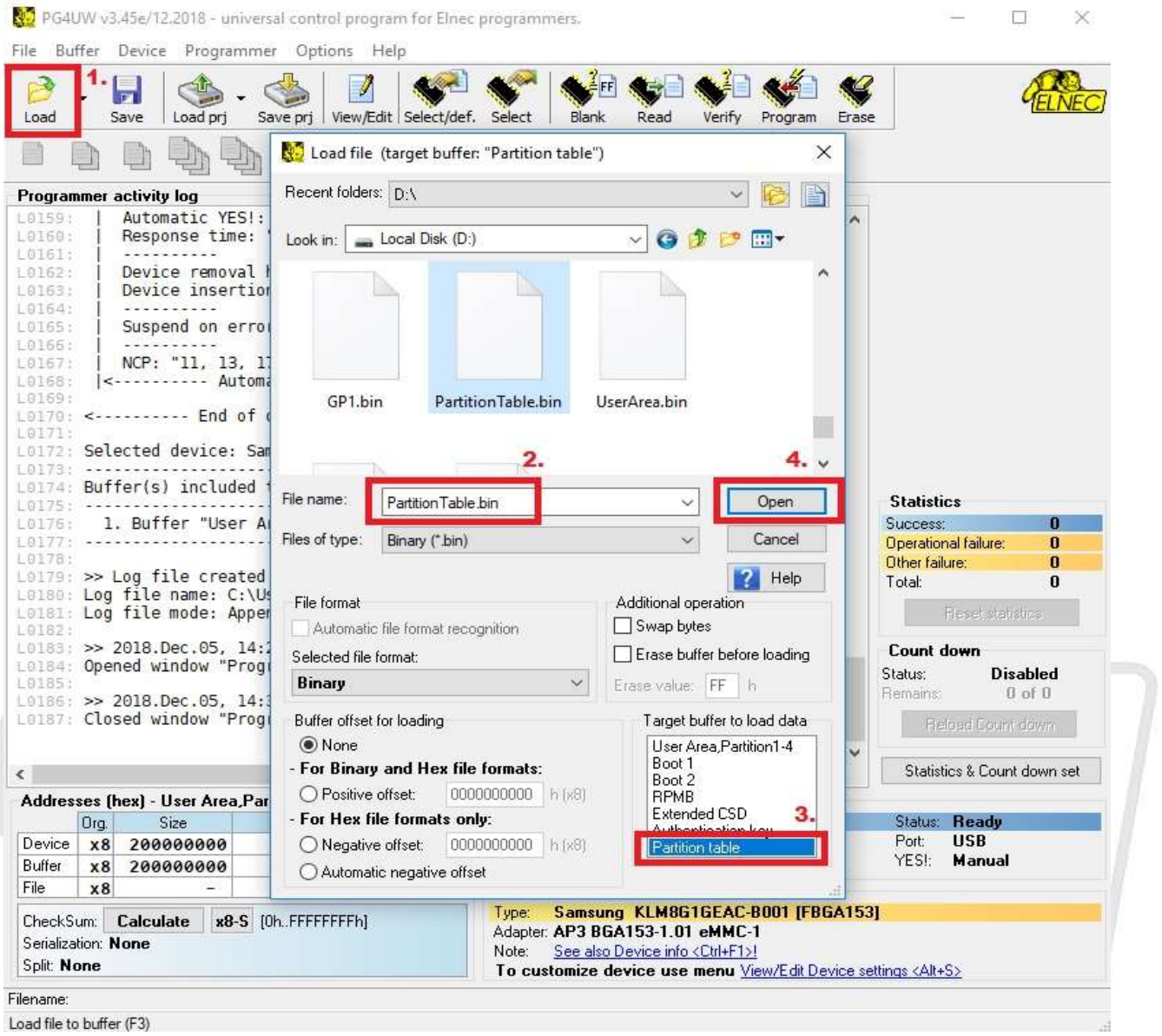

Figure 13. Load input file for Partition table

5.2.2 Example of loading several *.bin files when the option "Partition table" is enabled

UserArea default size = 37A400000h

eMMC device configuration:
Enhanced UserArea start address = 10000000h
Enhanced UserArea Size = 1A2800000h(pSLC)
Partition1/GP0 size = 1000000h(pMLC)
Partition2/GP1 size = 10000000h(pMLC)
Partition3/GP2 size = 12000000h(pMLC)
Partition4/GP3 size = 1000000h(pMLC)
UserArea Size = 37A400000h–1A2800000h-1000000h–10000000h-12000000h-1000000h = **1B3C00000h**

Programming files and destination:
1.DIW3870.bin(file size 1000000h) to GP0 address = 00h
2.SC60401.bin(file size 1000000h) to GP1 address = 00h
3.DIW3871.bin(file size 1000000h) to GP3 address = 00h
4.DTIW381.bin(file size 11000h) to UserArea address 1 = 10000000h
5.TT_D_De.bin(file size 1000000h) to UserArea address 2 = 10011000h
6.FSOS_Ex.bin(file size 1000000h) to UserArea address 3 = 14011000h

Calculation of the buffer offset for loading files:
1.DIW3870.bin:
Buffer **offset1** = **00h**
2.SC60401.bin:
Buffer **offset2** = (DIW3870.bin) size
= **1000000h**
3.DIW3871.bin:
Buffer **offset3** = (DIW3870.bin+SC60401.bin) size
= 1000000h+1000000h
= **2000000h**
4.DTIW381.bin:
Buffer **offset4** = (DIW3870.bin+SC60401.bin+DIW3871.bin) size
= 1000000h+1000000h+1000000h
= **3000000h**
5.TT_D_De.bin:
Buffer **offset5** = (DIW3870.bin+SC60401.bin+DIW3871.bin+DTIW381.bin) size
= 1000000h+1000000h+1000000h+11000h
= **3011000h**
6.FSOS_Ex.bin:
Buffer **offset6** = (DIW3870.bin+SC60401.bin+DIW3871.bin+DTIW381.bin+TT_D_De.bin) size
= 1000000h+1000000h+1000000h+11000h+1000000h
= 4011000h

Loading files sequence:
1. Clear buffer “User Area,Partition1-4”
2. Open Load file window, choose the “User Area,Partition1-4” target buffer, set positive buffer **offset1** = **00h** and then load the **DIW3870.bin** file.
3. Open Load file window, choose the “User Area,Partition1-4” target buffer, set positive buffer **offset2** = **1000000h** and then load the **SC60401.bin** file.
4. Open Load file window, choose the “User Area,Partition1-4” target buffer, set positive buffer **offset3** = **2000000h** and then load the **DIW3871.bin** file.
5. Open Load file window, choose the “User Area,Partition1-4” target buffer, set positive buffer **offset4** = **3000000h** and then load the **DTIW381.bin** file.
6. Open Load file window, choose the “User Area,Partition1-4” target buffer, set positive buffer **offset5** = **3011000h** and then load the **TT_D_De.bin** file.
7. Open Load file window, choose the “User Area,Partition1-4” target buffer, set positive buffer **offset6** = **4011000h** and then load the **FSOS_Ex.bin** file
8. Define partition table.(see figure 14)
9. Save project

Settings of the Partion table (in bytes):

Record 0: Device_start(h) = GP0 address = 00h
Buffer_start(h) = DIW3870.bin buffer offset1 = 00h
Data_size(h) = (DIW3870.bin)size = 1000000h
Device_Location(h) = Partition1/GP0 = 03h

Record 1: Device_start(h) = GP1 address = 00h
Buffer_start(h) = SC60401.bin buffer offset2 = 1000000h
Data_size(h) = (SC60401 .bin)size = 1000000h
Device_Location(h) = Partition2/GP1 = 04h

Record 2: Device_start(h) = GP3 address = 00h
Buffer_start(h) = DIW3871.bin buffer offset3 = 2000000h
Data_size(h) = (DIW3871 .bin)size = 1000000h
Device_Location(h) = Partition4/GP3 = 06h

Record 3: Device_start(h) = UserArea address 1 = 10000000h
Buffer_start(h) = DTIW381.bin buffer offset4 = 3000000h
Data_size(h) = (DTIW381.bin)size = 11000h
Device_Location(h) = User Area = 00h

Record 4: Device_start(h) = UserArea address 2 = 10011000h
Buffer_start(h) = TT_D_De.bin buffer offset5 = 3011000h
Data_size(h) = (TT_D_De .bin)size = 1000000h
Device_Location(h) = User Area = 00h

Record 5: Device_start(h) = UserArea address 3 = 14011000h
Buffer_start(h) = FSOS_Ex.bin buffer offset6 = 4011000h
Data_size(h) = (FSOS_Ex.bin)size = 1000000h
Device_Location(h) = User Area = 00h

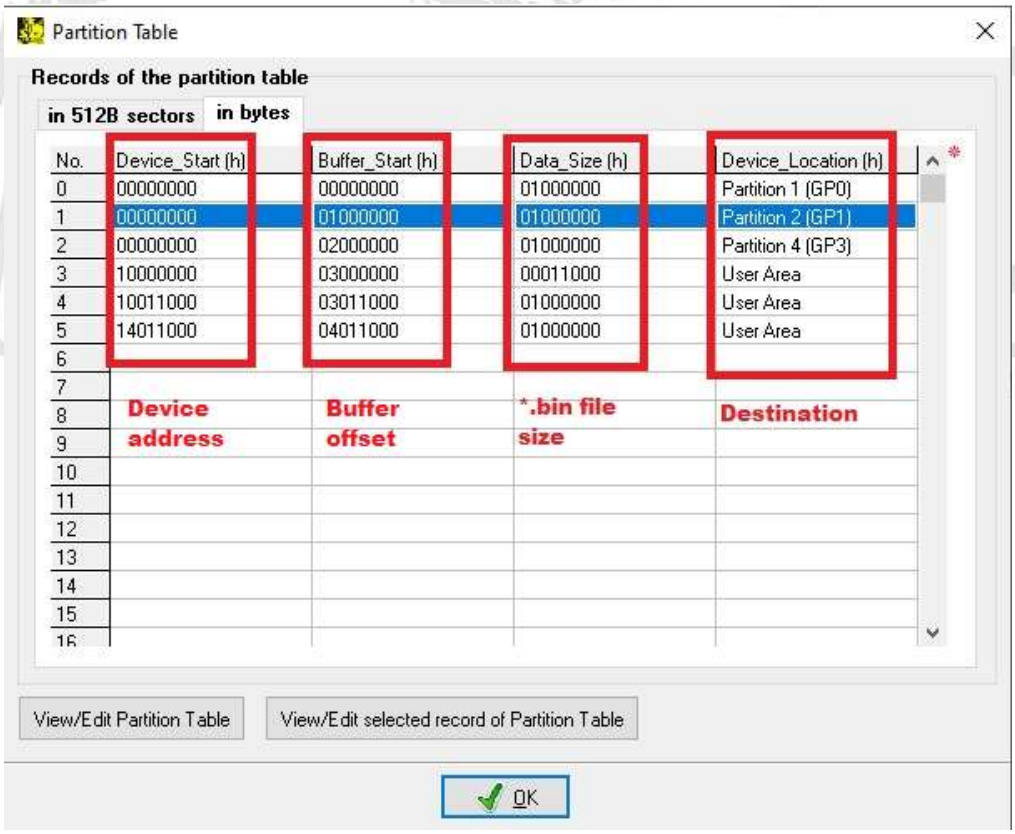

Figure 14. Example of partition table

6. Creating of copies from a Master device

The device settings option “Read operation with automatic programming parameters setting” analyze and set a whole group of device programming parameters(device end address, Extended CSD register, etc) based on the values readed from a Master device settings.

Working procedure:

1. Insert Master device to adapter ZIF socket

2. Enable the option “Read operation with automatic programming parameters setting”

3. Perform operation “READ”

4. Insert Blank device to adapter ZIF socket

5. Perform operation “Program”

6. Perform operation “Verify”

7. Save project

For working with a locked Master device it is necessary to enter the correct password to Device settings, section "Password", item "Password in device". For working with a Master device that contains "Authentication Key", it is necessary to enter the correct "Authentication Key" to the buffer. Settings "Production State Awareness" must be set manually by the user.

7. Erase operation

The PG4UW defines the address of the ranges using the ERASE_GROUP_START(CMD35), ERASE_GROUP_END(CMD36) commands and finally it starts the erase process by issuing the ERASE (CMD38) command with argument bits set to zero. The PG4UW performs the Discard operation before the Erase operation automatically for the devices eMMC version 4.5 or newer. The Trim operation is not supported.

8. Production State Awareness (PSA)

The PG4UW supports two methods to manage the device PSA: auto mode and manual mode. It is necessary to enter the correct PSA registers values and set the device addresses range option(i.e. User Area start, User Area end or Partition table). Please enable option "Erase before programming" in Device operation options.

8.1. Auto mode setting

Select programming the PRODUCT_STATE_AWARENESS_ENABLEMENT register and set the value to **30h.** The PG4UW sets register PRODUCT_STATE_AWARENESS to the 03h value automatically before the program data operation therefore it is not necessary to do this manually. The PRE_LOADING_DATA_SIZE register field value must be set to an appropriate image size, i.e. to the amount of programmed data to all normal partitions. Data written in BOOT, RPMB, Enhanced GPP and Enhanced User Area partitions must not be included when calculating the PRE_LOADING_DATA_SIZE register field value to set. (JEDEC Standard Specification No. JESD84-B51). The PG4UW (4.04 or newer) calculates the value of PRE_LOADING_DATA_SIZE before the program operation. If the PRE_LOADING_DATA_SIZE register is not set according to the data size, you can enable the automatic setting.(see section 8.1.1.)

8.1.1. Example of configuration settings for Auto mode

1. Set PRODUCT_STATE_AWARENESS_ENABLEMENT[17] = 30h (70h for Micron eMMC TLC (Pearl))
2. Set PRE_LOADING_DATA_SIZE [25:22] = (see Note)
3. Perform operation “Program”
4. Save project

Note: If the PRE_LOADING_DATA_SIZE register is not set according to the data size, select automatic setting during first program operation. If you found some problems, let us know.

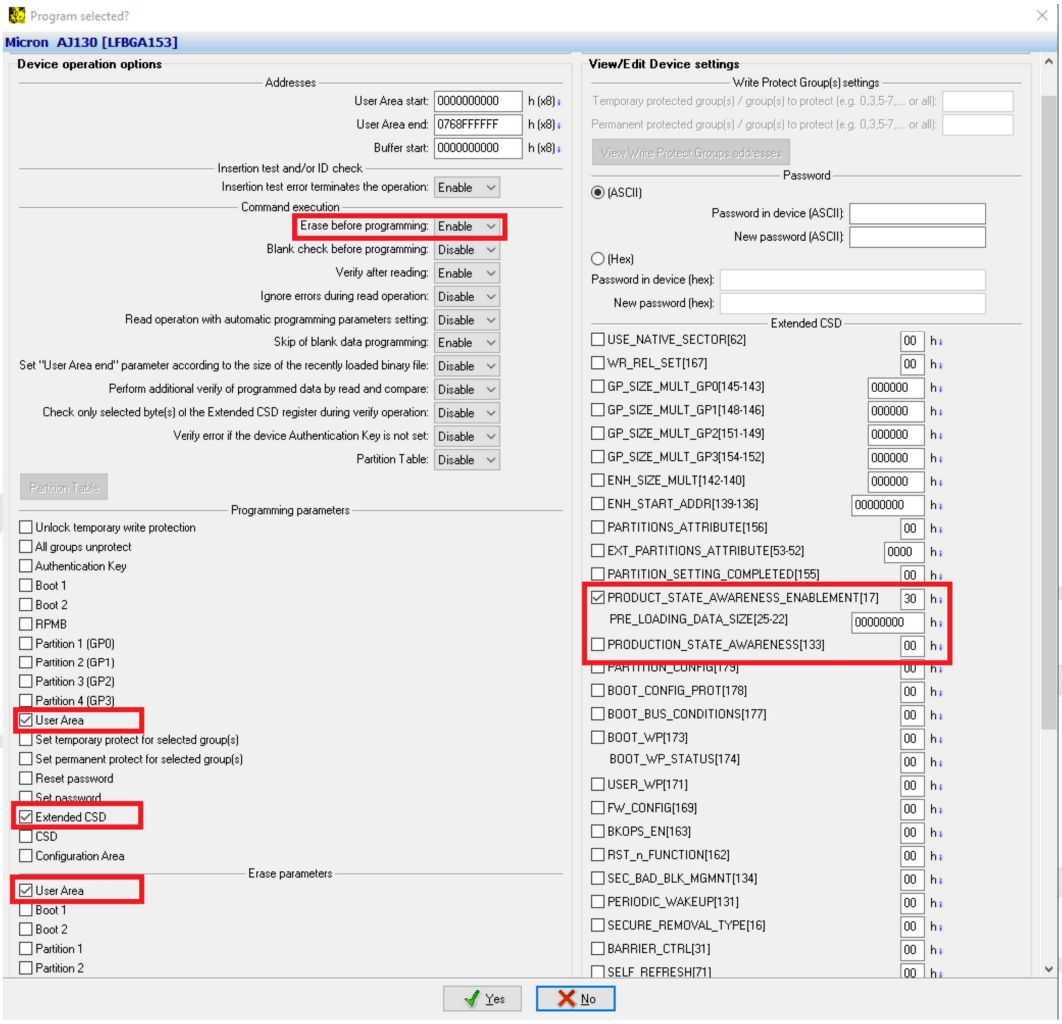

Figure 15. Example of configuration settings for Auto mode

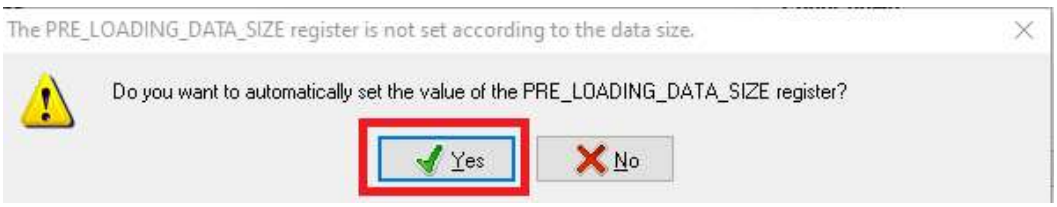

Figure 15a. Set automatically the value of the PRE_LOADING_DATA_SIZE register

8.2. Manual mode setting

Select programming the PRODUCT_STATE_AWARENESS_ENABLEMENT register and set the value to **10h**. Select programming the PRODUCT_STATE_AWARENESS register and set the value to 02h. The PG4UW sets register PRODUCT_STATE_AWARENESS to the 01h value automatically before the program data operation therefore it is not necessary to do this manually. The PRE_LOADING_DATA_SIZE register field value must be set to an appropriate image size, i.e. to the amount of programmed data to all normal partitions. Data written in BOOT, RPMB, Enhanced GPP and Enhanced User Area partitions must not be included when calculating the PRE_LOADING_DATA_SIZE register field value to set. (JEDEC Standard Specification No. JESD84-B51). The PG4UW (4.04 or newer) calculates the value of PRE_LOADING_DATA_SIZE before the program operation. If the PRE_LOADING_DATA_SIZE register is not set according to the data size, you can enable the automatic setting.(see section 8.1.2.)

8.2.1. Example of configuration settings for Manual mode

1. Set PRODUCT_STATE_AWARENESS_ENABLEMENT[17] = 10h (50h for Micron eMMC TLC (Pearl))
2. Set PRE_LOADING_DATA_SIZE [25:22] = (see Note)
3. Set PRODUCT_STATE_AWARENESS[133] = 02h
4. Perform operation “Program”
5. Save project

Note: If the PRE_LOADING_DATA_SIZE register is not set according to the data size, select automatic setting during first program operation. If you found some problems, let us know.

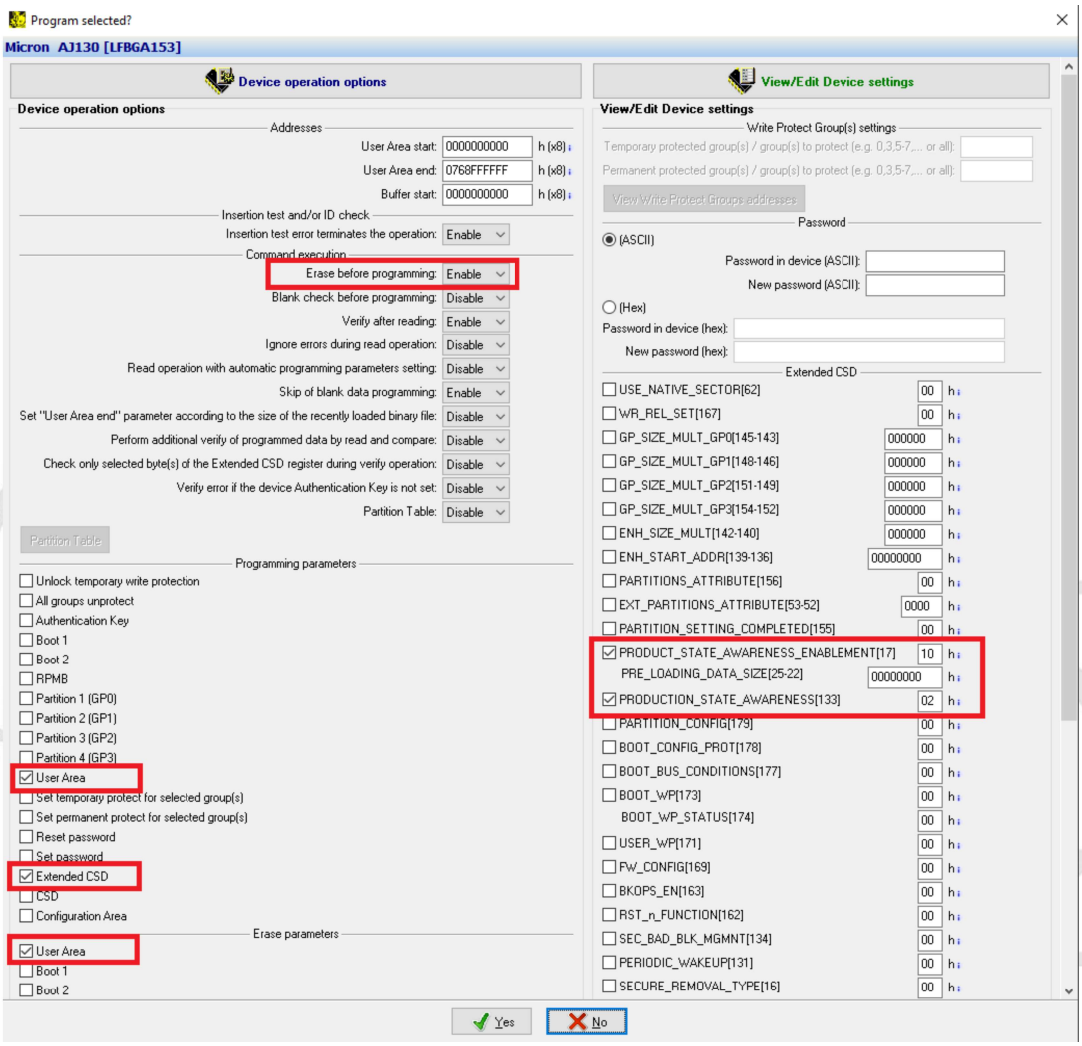

Figure 16. Example of configuration settings for Manual mode

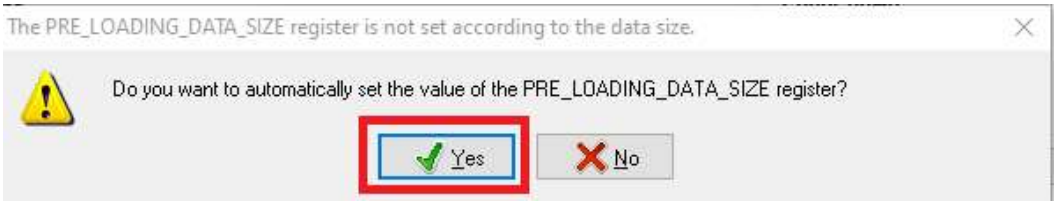

Figure 16a. Set automatically the value of the PRE_LOADING_DATA_SIZE register

9. Program RPMB (Authentication Key)

Access to RPMB area is not allowed/possible before the Authentication Key is programmed. If the Authentication Key has been programmed to the device, then the correct Authentication Key must be entered into the PG4UW buffer(the same Authentication Key as programmed in device) for the success operations with the RPMB area. The Authentication key can be programmed only once and cannot be read from the device.

The content in RPMB area doesn't provide the data confidentiality, i.e. everyone can read them. It means that read operation itself reads the data from the RPMB area also in the case that incorrect Authentication Key is provided. The correct Authentication Key and RPMB content must be programmed for the proper operation of the device in the user application. Detail description is in JEDEC Standard Specification No. JESD84-B451.

To enable programming of the RPMB area or the Authentication Key, open the Device operation options (ALT+O) window and use the checkbox “RPMB” or “Authentication Key”. The erase operation of RPMB area is not available (and therefore do not implemented). The RPMB area can be re-programmed without needing of erase procedure.

For loading or saving data to the RPMB area select RPMB tab in View/Edit Buffer window and use icons Load or Save. For loading or saving Authentication Key data select Authentication Key tab in View/Edit Buffer window and use icons Load or Save.

Versions history

Revision	Date	Comment
1.00	December 5, 2018	• initial version
1.01	January 15, 2020	• added “Creating of copies from a Master device”
1.02	July 15, 2020	• added example
1.03	July 20, 2021	• added section 1.1 • updated section 5.1
1.04	January 11, 2022	• changed section 1.1 • added section 1.2
1.05	May 25, 2023	• added section 7
1.06	August 10, 2023	• added section 8
1.07	March 13, 2025	• added section 9
1.08	April 3, 2025	• added Disclaimer
1.09	July 10, 2025	• updated section 8