G+ Code Packer User Guide

V1.0.8 – Oct 29, 2012

No.19, Industry E. Rd. IV, Hsinchu Science Park, Hsinchu City 30077, Taiwan, R.O.C.
Tel: +886-3-666-2118 Fax: +886-3-666-2117 Web: www.generalplus.com

Important Notice

GENERALPLUS TECHNOLOGY INC. reserves the right to change this documentation without prior notice. Information provided by GENERALPLUS TECHNOLOGY INC. is believed to be accurate and reliable. However, GENERALPLUS TECHNOLOGY INC. makes no warranty for any errors which may appear in this document. Contact GENERALPLUS TECHNOLOGY INC. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by GENERALPLUS TECHNOLOGY INC. for any infringement of patent or other rights of third parties which may result from its use. In addition, GENERALPLUS products are not authorized for use as critical components in life support devices/ systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Generalplus.

Table of Content

Revision History

Revision	Date	By	Remark
V1.0.0	2009/11/16	Dominant Yang Luowl Willie Wang	1. First edition for G+ Code Packer user guide.
V1.0.1	2010/6/11	Kenny	1. Update the user guide to G+CodePacker V1.0.2 and higher version.
V1.0.2	2010/8/23	Kenny	1. Add FAT Image generation function. 2. Add upload function for NAND flash.
V1.0.3	2010/12/8	Kenny	1. List out IC body support 2. Add APP spare size limitation
V1.0.4	2011/4/13	Kenny	1. Add GPL329XX body
V1.0.5	2011/6/15	Kenny	Add eMMC NVRAM support for GPL329XX
V1.0.6	2011/10/14	Willie	Update G+CodePacker function description.
V1.0.7	2012/6/4	Willie	Update for G+CodePacker V1.0.9
V1.0.8	2012/10/29	Willie	Add descriptions for FAT Image size limitation and the function of “Check for Update(s)”.

1 Introduction to G+ Code Packer

1.1 Tool Introduction

G+ Code Packer is a resource integration tool developed by Generalplus. This tool is able to gather all resource files such as boot code, user runtime code, and all other resource files into a single bin file (ROM Image). It supports several NVRAM types: SPI NOR Flash, NAND Flash, and SDCard.

In addition to integrate resource files, G+ Code Packer also features the following functions:

1. Performs DRAM Calibration which can be used for pre-set register.
2. Integrates FAT Image Tool to generate data image file with file system.
3. Integrates USB Mass Production Tool. After resources are completely integrated, program the bin file onto designated devices.
4. Export the BIN file with USB Mass Production Tool for mass production.

1.2 Applicable IC

G+ Code Packer only supports the following IC series.

GPL325XX-003A

GPL326XX, all versions

GPL329XX, all versions

GPL327XX, all versions

GPDV6XXX, all versions

2 Installation

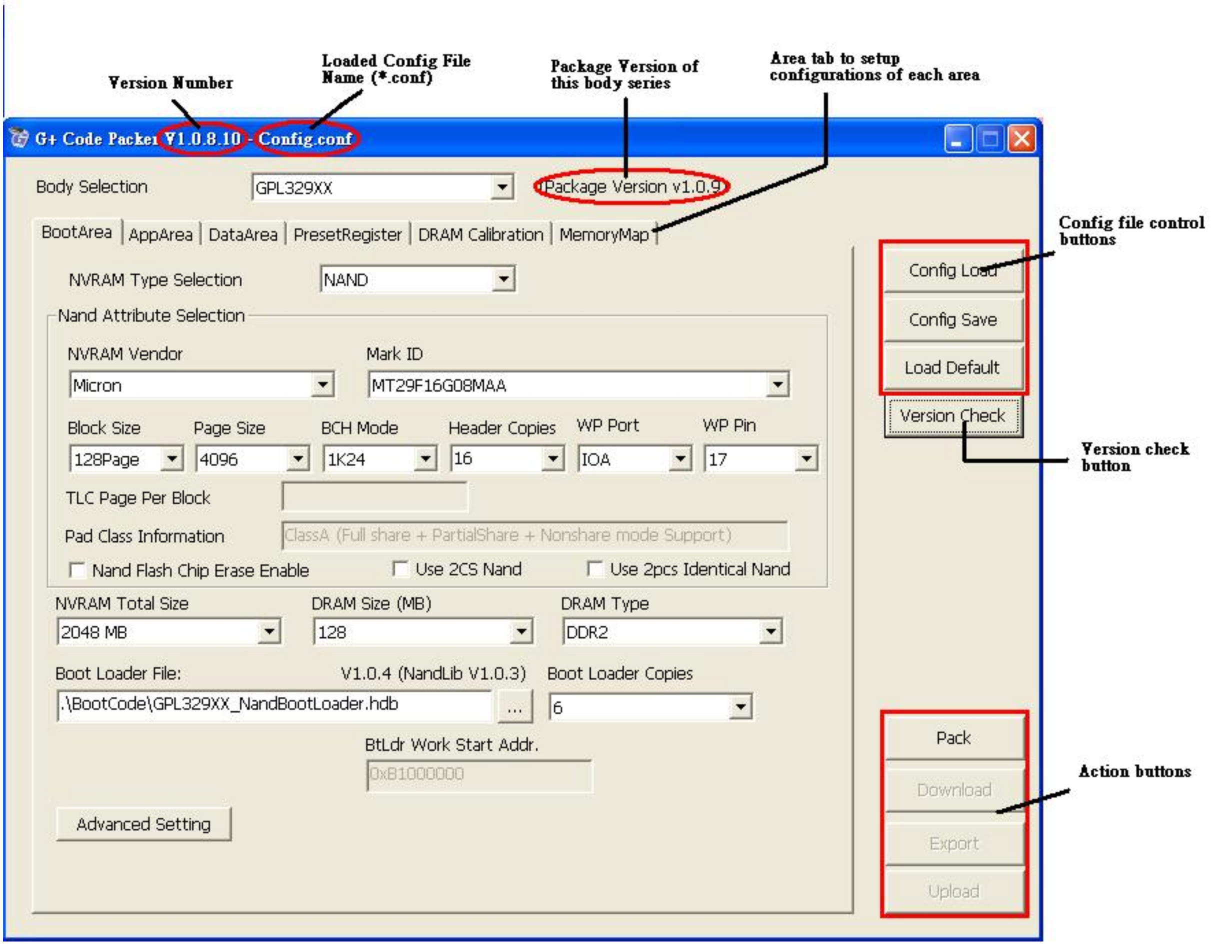
2.1 Install and Upgrade

1. G+ Code Packer is an installation-free tool. It only needs to be extracted to a destination from a zip file.
2. To upgrade the tool, simply overwrite the older file by an upgrade version.

3 G+ Code Packer Function

3.1 Main Menu

The G+ Code Packer main menu is depicted as follows:



Version Number

Loaded Config File Name (*.conf)

Package Version of this body series

Area tab to setup configurations of each area

Config file control buttons

Version check button

Action buttons

G+ Code Packer V1.0.8.10 - Config.conf

Body Selection: GPL329XX

Package Version v1.0.9

NVRAM Type Selection: NAND

Nand Attribute Selection

NVRAM Vendor: Micron

Mark ID: MT29F16G08MAA

Block Size: 128Page

Page Size: 4096

BCH Mode: 1K24

Header Copies: 16

WP Port: IOA

WP Pin: 17

TLC Page Per Block:

Pad Class Information: ClassA (Full share + PartialShare + Nonshare mode Support)

□ Nand Flash Chip Erase Enable

□ Use 2CS Nand

□ Use 2pcs Identical Nand

NVRAM Total Size: 2048 MB

DRAM Size (MB): 128

DRAM Type: DDR2

Boot Loader File: V1.0.4 (NandLib V1.0.3)

.\BootCode\GPL329XX_NandBootLoader.hdb

Boot Loader Copies: 6

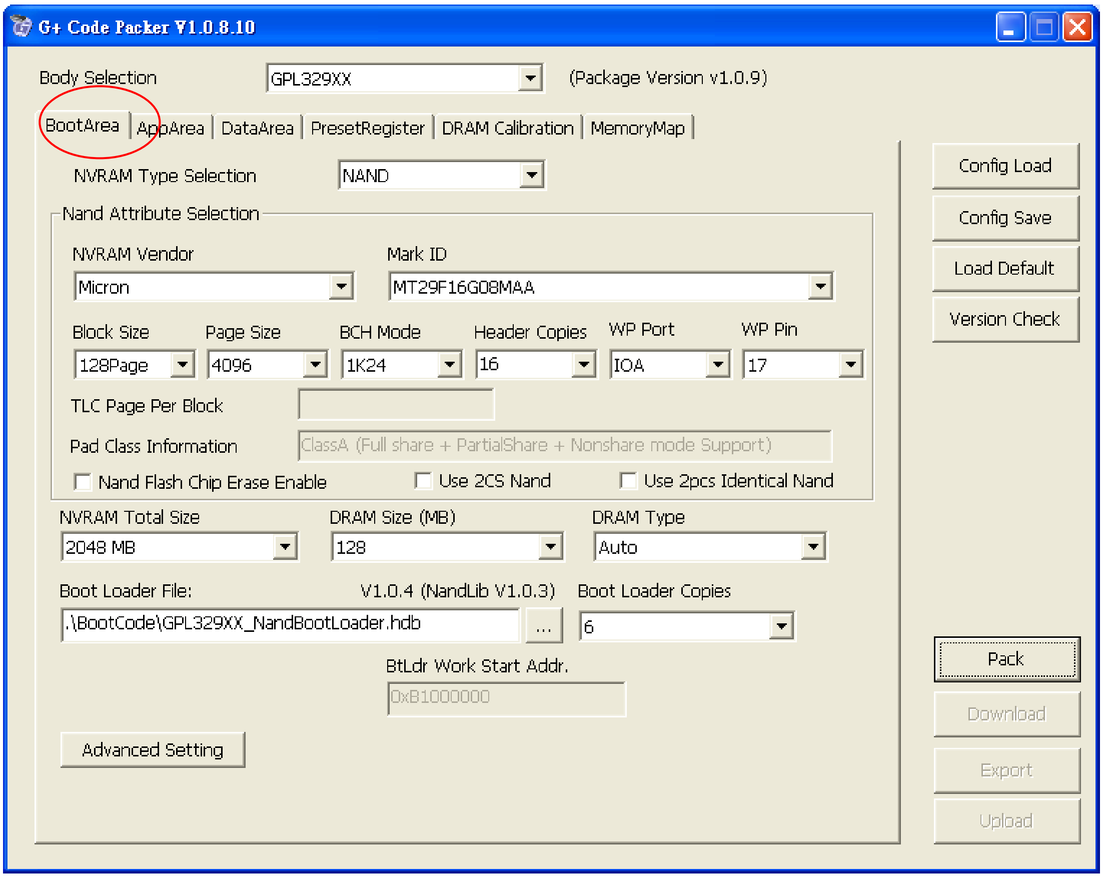
BtLdr Work Start Addr.: 0xB1000000

3.2 Interfaces

3.2.1 Boot Area

Boot area is the first area to be processed after the internal ROM Boot up (on chip). The main file in this area is the Boot Loader File. We recommend using the Boot Loader, provided by Generalplus, to achieve the best and correct result. If necessary, the user can also change the Boot Loader File's directory to load user's Boot Loader to meet special needs. In addition to initialize App Area, the Boot Loader also loads user's Runtime code to be executed on DRAM from NVRAM App Area.

Boot Area's user interface is as follows:


G+ Code Packer V1.0.8.10

Body Selection GPL329XX (Package Version v1.0.9)

NVRAM Type Selection: NAND

Nand Attribute Selection

NVRAM Vendor: Micron

Mark ID: MT29F16G08MAA

Block Size: 128Page

Page Size: 4096

BCH Mode: 1K24

Header Copies: 16

WP Port: IOA

WP Pin: 17

TLC Page Per Block

Pad Class Information: ClassA (Full share + PartialShare + Nonshare mode Support)

□ Nand Flash Chip Erase Enable □ Use 2CS Nand □ Use 2pcs Identical Nand

NVRAM Total Size: 2048 MB

DRAM Size (MB): 128

DRAM Type: Auto

Boot Loader File: .\BootCode\GPL329XX_NandBootLoader.hdb

V1.0.4 (NandLib V1.0.3)

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0xB1000000

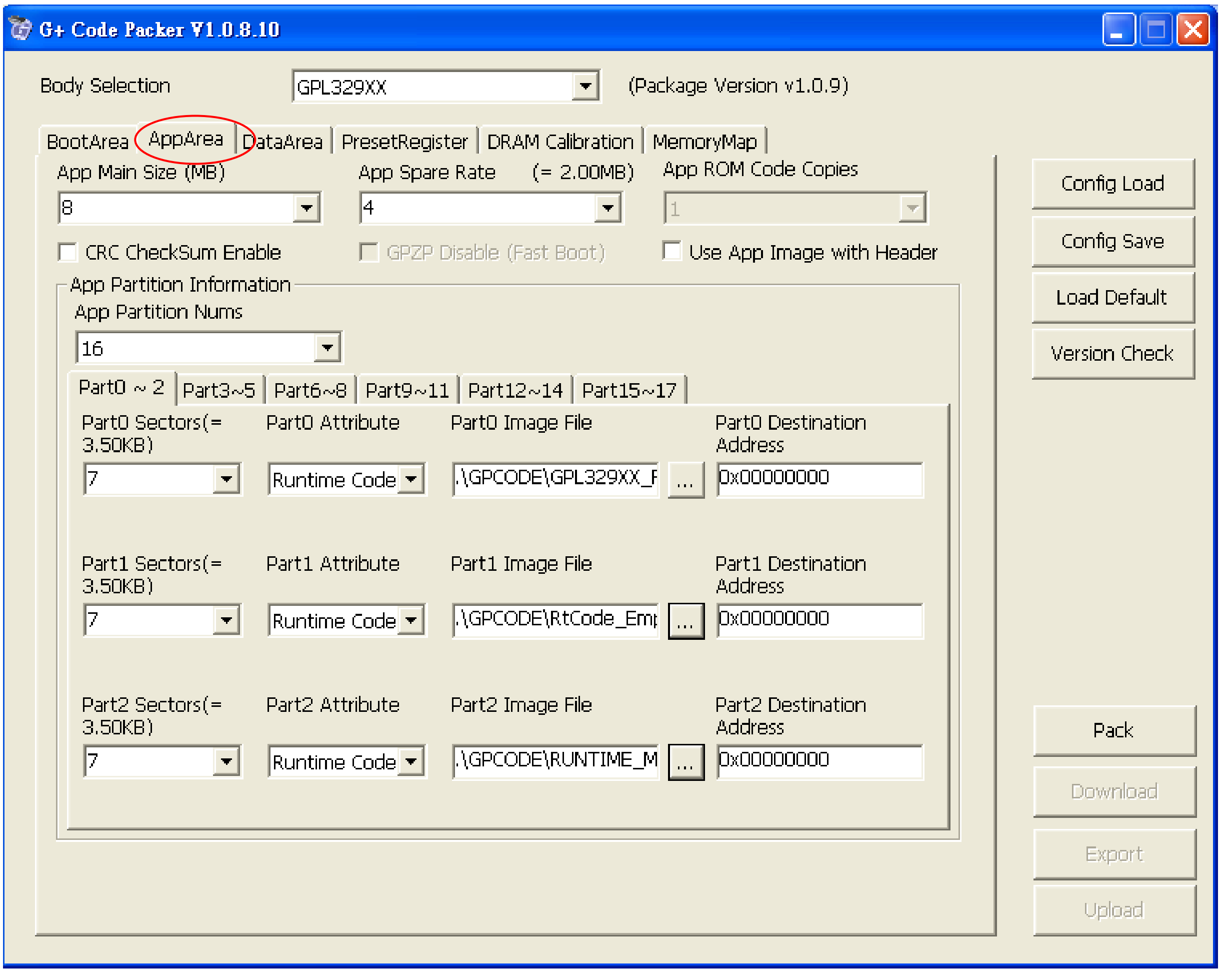
3.2.2 App Area

The main file in App Area is the user’s Runtime Code. App Area is initialized by Boot Loader. If Boot Loader is not detected by internal ROM boot, the App Area service becomes void. App Area design guideline is as follows:

a. App Area is read-only area

b. App Area is mainly used for storing user’s Runtime Code. Runtime code means the Runtime Process Image which can be an application such as DPF program, a game, a DV, a ELA, …,etc.

App Area’s user interface is as follows:



Body Selection: GPL329XX (Package Version v1.0.9)

App Main Size (MB): 8

App Spare Rate: 4 (= 2.00MB)

App ROM Code Copies: 1

□ CRC CheckSum Enable □ GPZP Disable (Fast Boot) □ Use App Image with Header

App Partition Information

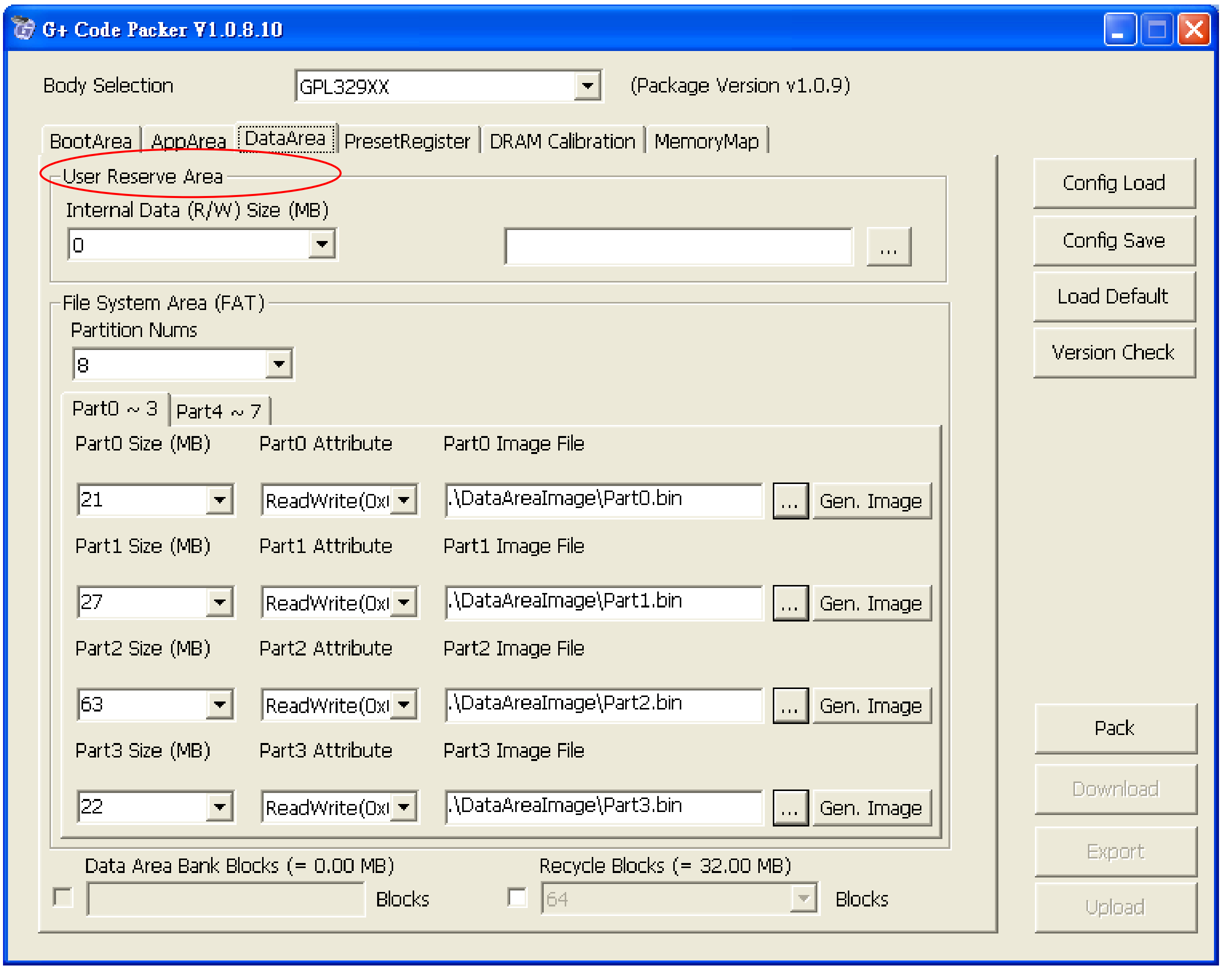
App Partition Nums: 16

Part0 Sectors(= 3.50KB)	Part0 Attribute	Part0 Image File	Part0 Destination Address
7	Runtime Code	.\GPCODE\GPL329XX_F	0x00000000
Part1 Sectors(= 3.50KB)	**Part1 Attribute**	**Part1 Image File**	**Part1 Destination Address**
7	Runtime Code	.\GPCODE\RtCode_Em	0x00000000
Part2 Sectors(= 3.50KB)	**Part2 Attribute**	**Part2 Image File**	**Part2 Destination Address**
7	Runtime Code	.\GPCODE\RUNTIME_M	0x00000000

3.2.3 Data Area

The data area is the area that NVRAM subtracts the Boot Area and App Area. If Generalplus solution is applied in this area, R/W area (most often access area, no specific format) and file system partition are available (FAT16/32 or SPI FAT12 is commonly used).

Data Area’s user interface is as follows:



Body Selection: GPL329XX (Package Version v1.0.9)

User Reserve Area

Internal Data (R/W) Size (MB): 0

File System Area (FAT)

Partition Nums: 8

Part0 ~ 3 | Part4 ~ 7

Size (MB)	Attribute	Image File
Part0 Size (MB): 21	Part0 Attribute: ReadWrite(0x	Part0 Image File: .\DataAreaImage\Part0.bin
Part1 Size (MB): 27	Part1 Attribute: ReadWrite(0x	Part1 Image File: .\DataAreaImage\Part1.bin
Part2 Size (MB): 63	Part2 Attribute: ReadWrite(0x	Part2 Image File: .\DataAreaImage\Part2.bin
Part3 Size (MB): 22	Part3 Attribute: ReadWrite(0x	Part3 Image File: .\DataAreaImage\Part3.bin

☐ Data Area Bank Blocks (= 0.00 MB) Blocks

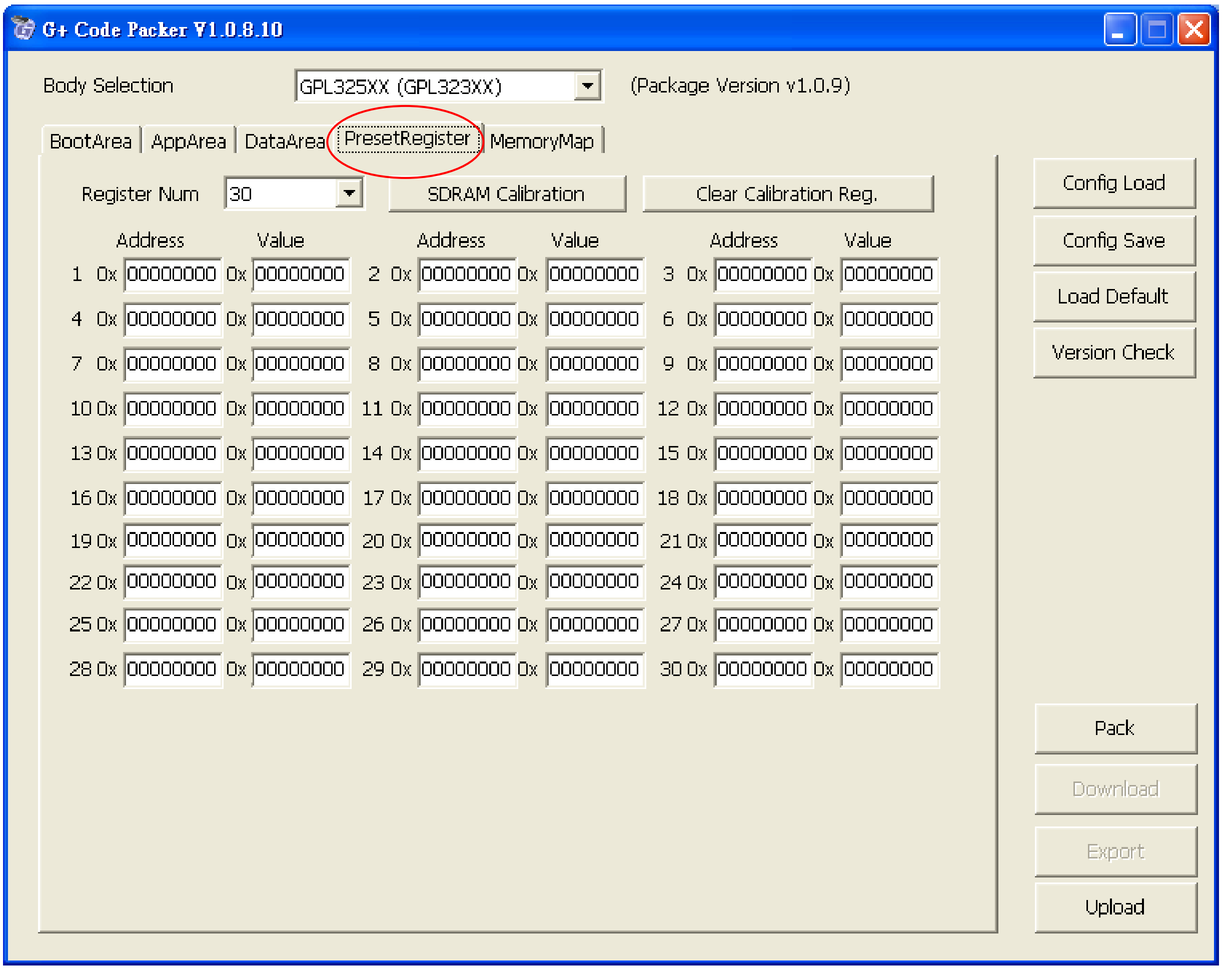
☐ Recycle Blocks (= 32.00 MB) 64 Blocks

3.2.4 Pre-set Register & DRAM Calibration

After the Internal MASK ROM code is run, the register can be set, and each parameter includes: register address and the content written into register.

Pre-set Register's interface differs dependent on the body chosen. Different NVRAM type determines the set number that register can use.

GPL325XX:



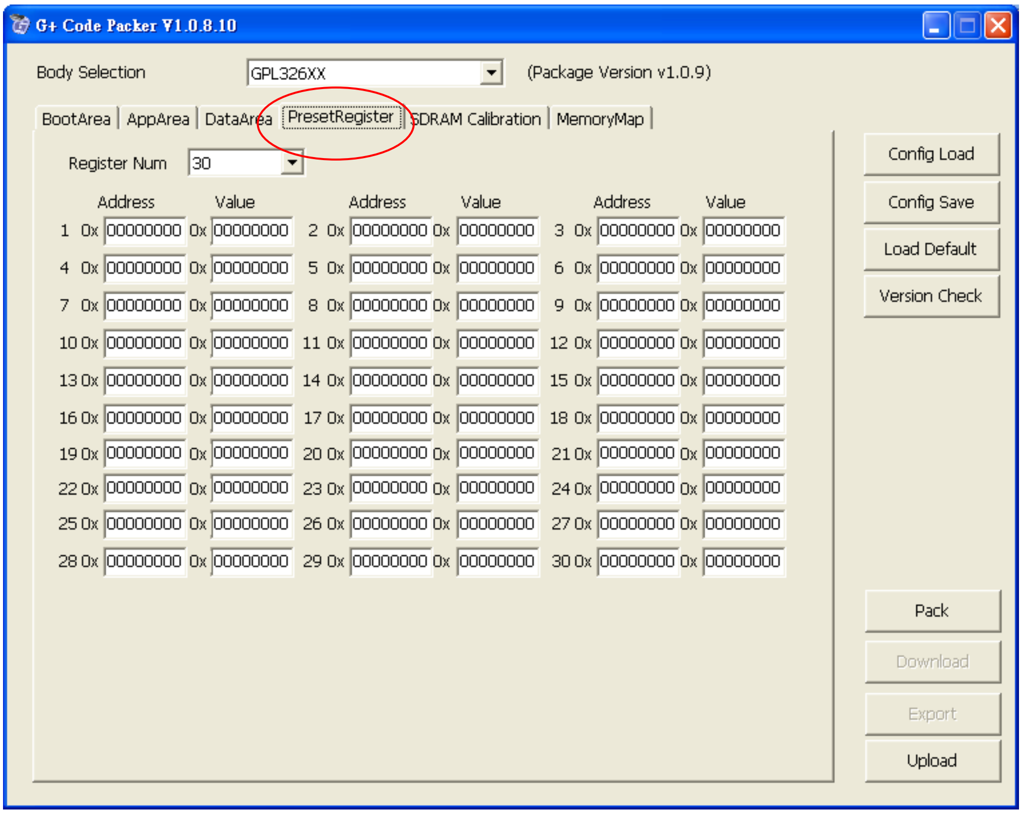
G+ Code Packer V1.0.8.10

Body Selection: GPL325XX (GPL323XX) (Package Version v1.0.9)

Register Num: 30

	Address	Value		Address	Value		Address	Value
1	0x 00000000	0x 00000000	2	0x 00000000	0x 00000000	3	0x 00000000	0x 00000000
4	0x 00000000	0x 00000000	5	0x 00000000	0x 00000000	6	0x 00000000	0x 00000000
7	0x 00000000	0x 00000000	8	0x 00000000	0x 00000000	9	0x 00000000	0x 00000000
10	0x 00000000	0x 00000000	11	0x 00000000	0x 00000000	12	0x 00000000	0x 00000000
13	0x 00000000	0x 00000000	14	0x 00000000	0x 00000000	15	0x 00000000	0x 00000000
16	0x 00000000	0x 00000000	17	0x 00000000	0x 00000000	18	0x 00000000	0x 00000000
19	0x 00000000	0x 00000000	20	0x 00000000	0x 00000000	21	0x 00000000	0x 00000000
22	0x 00000000	0x 00000000	23	0x 00000000	0x 00000000	24	0x 00000000	0x 00000000
25	0x 00000000	0x 00000000	26	0x 00000000	0x 00000000	27	0x 00000000	0x 00000000
28	0x 00000000	0x 00000000	29	0x 00000000	0x 00000000	30	0x 00000000	0x 00000000

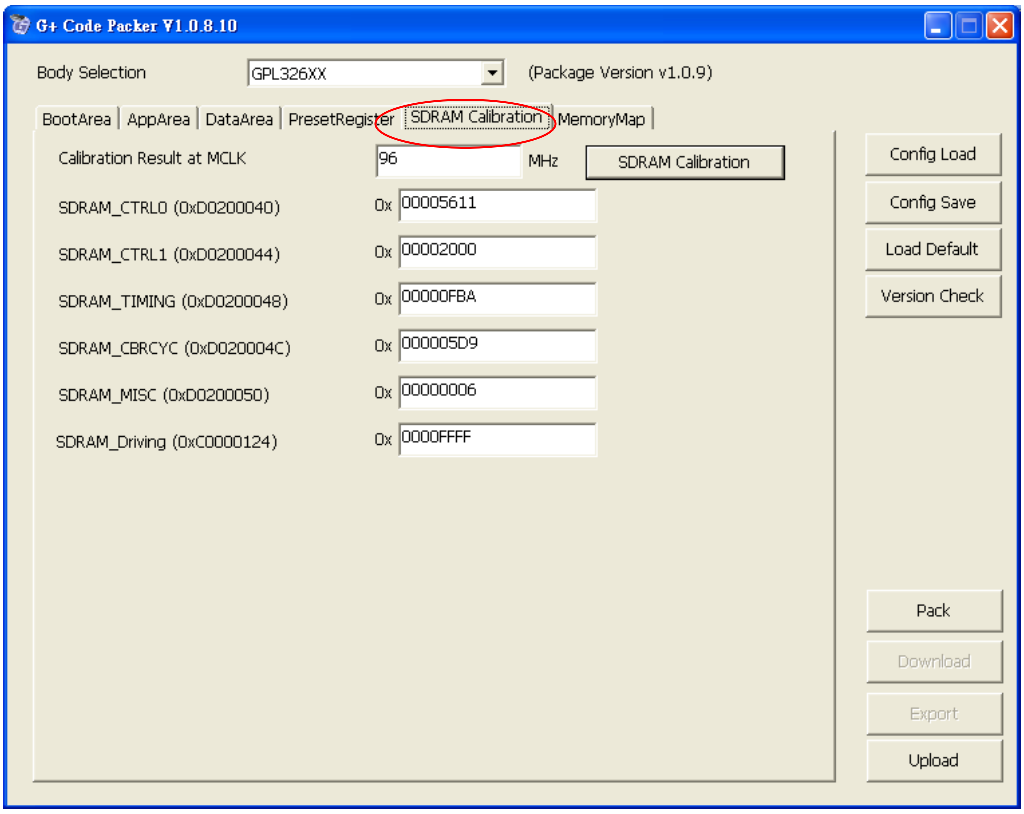
GPL326XX: This page does not include SDRAM Calibration button. The IC shows the exclusive SDRAM information in this page.



Body Selection GPL326XX (Package Version v1.0.9)

Register Num 30

	Address	Value		Address	Value		Address	Value
1	0x 00000000	0x 00000000	2	0x 00000000	0x 00000000	3	0x 00000000	0x 00000000
4	0x 00000000	0x 00000000	5	0x 00000000	0x 00000000	6	0x 00000000	0x 00000000
7	0x 00000000	0x 00000000	8	0x 00000000	0x 00000000	9	0x 00000000	0x 00000000
10	0x 00000000	0x 00000000	11	0x 00000000	0x 00000000	12	0x 00000000	0x 00000000
13	0x 00000000	0x 00000000	14	0x 00000000	0x 00000000	15	0x 00000000	0x 00000000
16	0x 00000000	0x 00000000	17	0x 00000000	0x 00000000	18	0x 00000000	0x 00000000
19	0x 00000000	0x 00000000	20	0x 00000000	0x 00000000	21	0x 00000000	0x 00000000
22	0x 00000000	0x 00000000	23	0x 00000000	0x 00000000	24	0x 00000000	0x 00000000
25	0x 00000000	0x 00000000	26	0x 00000000	0x 00000000	27	0x 00000000	0x 00000000
28	0x 00000000	0x 00000000	29	0x 00000000	0x 00000000	30	0x 00000000	0x 00000000



Body Selection GPL326XX (Package Version v1.0.9)

Calibration Result at MCLK 96 MHz

SDRAM_CTRL0 (0xD0200040) 0x 00005611

SDRAM_CTRL1 (0xD0200044) 0x 00002000

SDRAM_TIMING (0xD0200048) 0x 00000FBA

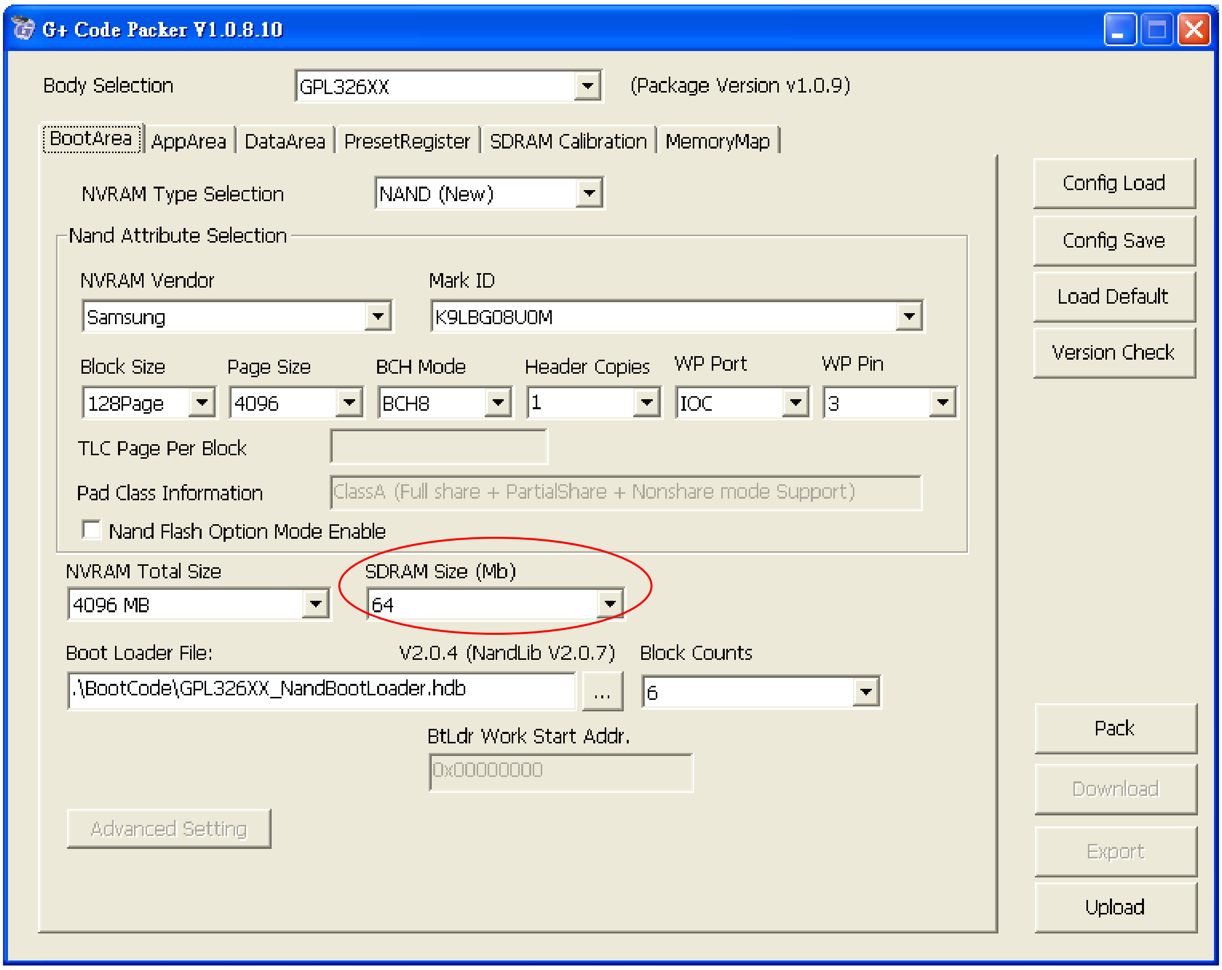
SDRAM_CBRCYC (0xD020004C) 0x 000005D9

SDRAM_MISC (0xD0200050) 0x 00000006

SDRAM_Driving (0xC0000124) 0x 0000FFFF

Be sure the App code is running on DRAM stably, we suggest using the DRAM calibration to set the DRAM parameter. Click on the DRAM Calibration to test the DRAM. It will obtain a set of good DRAM parameter which will be loaded into Pre-set Register.

Please note that DRAM size must be set correctly before executing DRAM calibration. Otherwise, the execution of DRAM calibration may fail.



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

NVRAM Type Selection: NAND (New)

Nand Attribute Selection

NVRAM Vendor: Samsung

Mark ID: K9LBG08U0M

Block Size	Page Size	BCH Mode	Header Copies	WP Port	WP Pin
128Page	4096	BCH8	1	IOC	3

TLC Page Per Block:

Pad Class Information: ClassA (Full share + PartialShare + Nonshare mode Support)

☐ Nand Flash Option Mode Enable

NVRAM Total Size: 4096 MB

SDRAM Size (Mb): 64

Boot Loader File: .\BootCode\GPL326XX_NandBootLoader.hdb

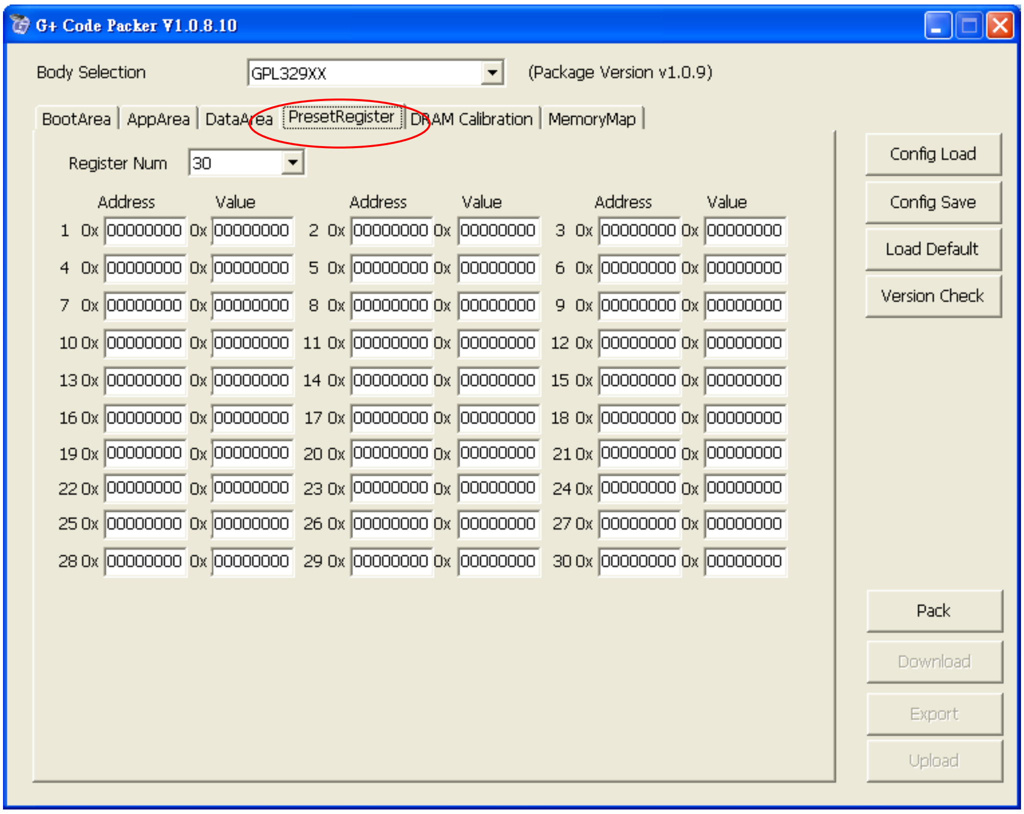
V2.0.4 (NandLib V2.0.7)

Block Counts: 6

BtLdr Work Start Addr.: 0x00000000

For more information about the Pre-set Register and DRAM calibration, please refer to Chapter 4.

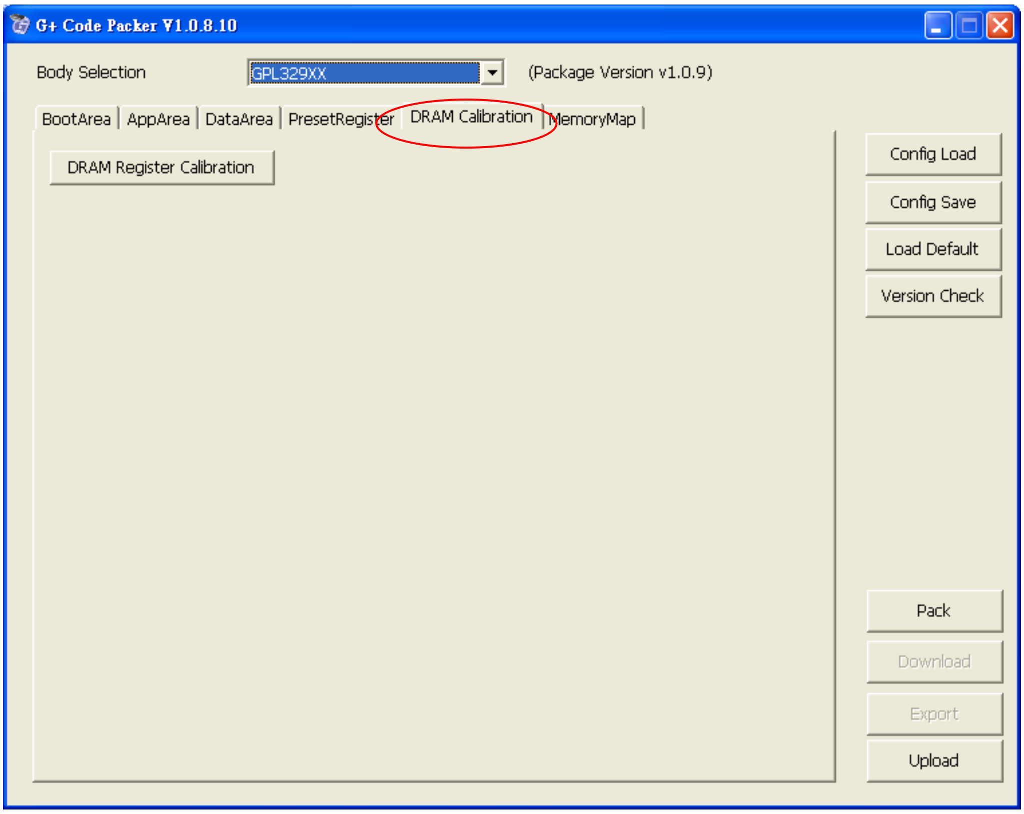
GPL329XX/GPL327XX/GPDV6XXX: These ICs will separate the Pre-Register and DRAM Calibration to different pages.



Body Selection GPL329XX (Package Version v1.0.9)

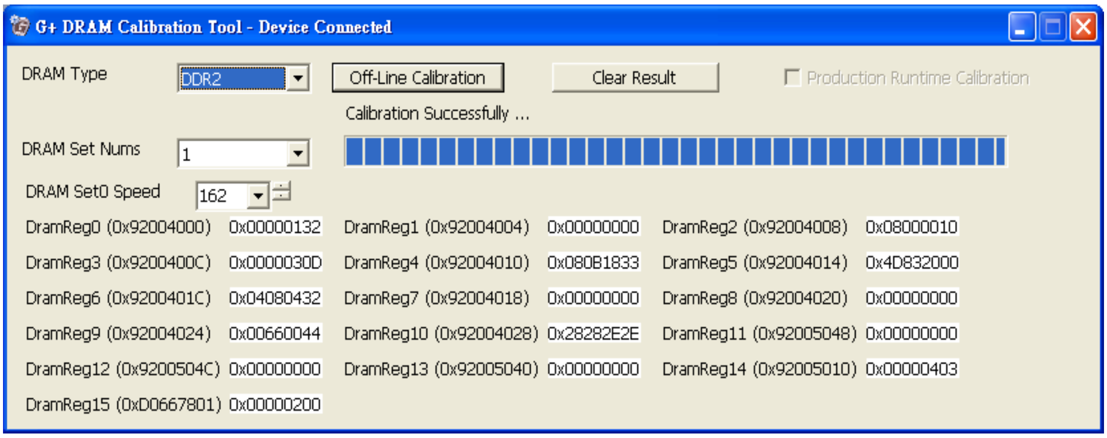
Register Num 30

	Address	Value		Address	Value		Address	Value
1	0x 00000000	0x 00000000	2	0x 00000000	0x 00000000	3	0x 00000000	0x 00000000
4	0x 00000000	0x 00000000	5	0x 00000000	0x 00000000	6	0x 00000000	0x 00000000
7	0x 00000000	0x 00000000	8	0x 00000000	0x 00000000	9	0x 00000000	0x 00000000
10	0x 00000000	0x 00000000	11	0x 00000000	0x 00000000	12	0x 00000000	0x 00000000
13	0x 00000000	0x 00000000	14	0x 00000000	0x 00000000	15	0x 00000000	0x 00000000
16	0x 00000000	0x 00000000	17	0x 00000000	0x 00000000	18	0x 00000000	0x 00000000
19	0x 00000000	0x 00000000	20	0x 00000000	0x 00000000	21	0x 00000000	0x 00000000
22	0x 00000000	0x 00000000	23	0x 00000000	0x 00000000	24	0x 00000000	0x 00000000
25	0x 00000000	0x 00000000	26	0x 00000000	0x 00000000	27	0x 00000000	0x 00000000
28	0x 00000000	0x 00000000	29	0x 00000000	0x 00000000	30	0x 00000000	0x 00000000



Body Selection GPL329XX (Package Version v1.0.9)

Press DRAM Register Calibration button to call DRAM Calibration Tool for Calibration.


G+ DRAM Calibration Tool - Device Connected
Off-Line Calibration
Clear Result
□ Production Runtime Calibration

DRAM Type: DDR2

Calibration Successfully ...

DRAM Set Nums: 1

DRAM Set0 Speed: 162

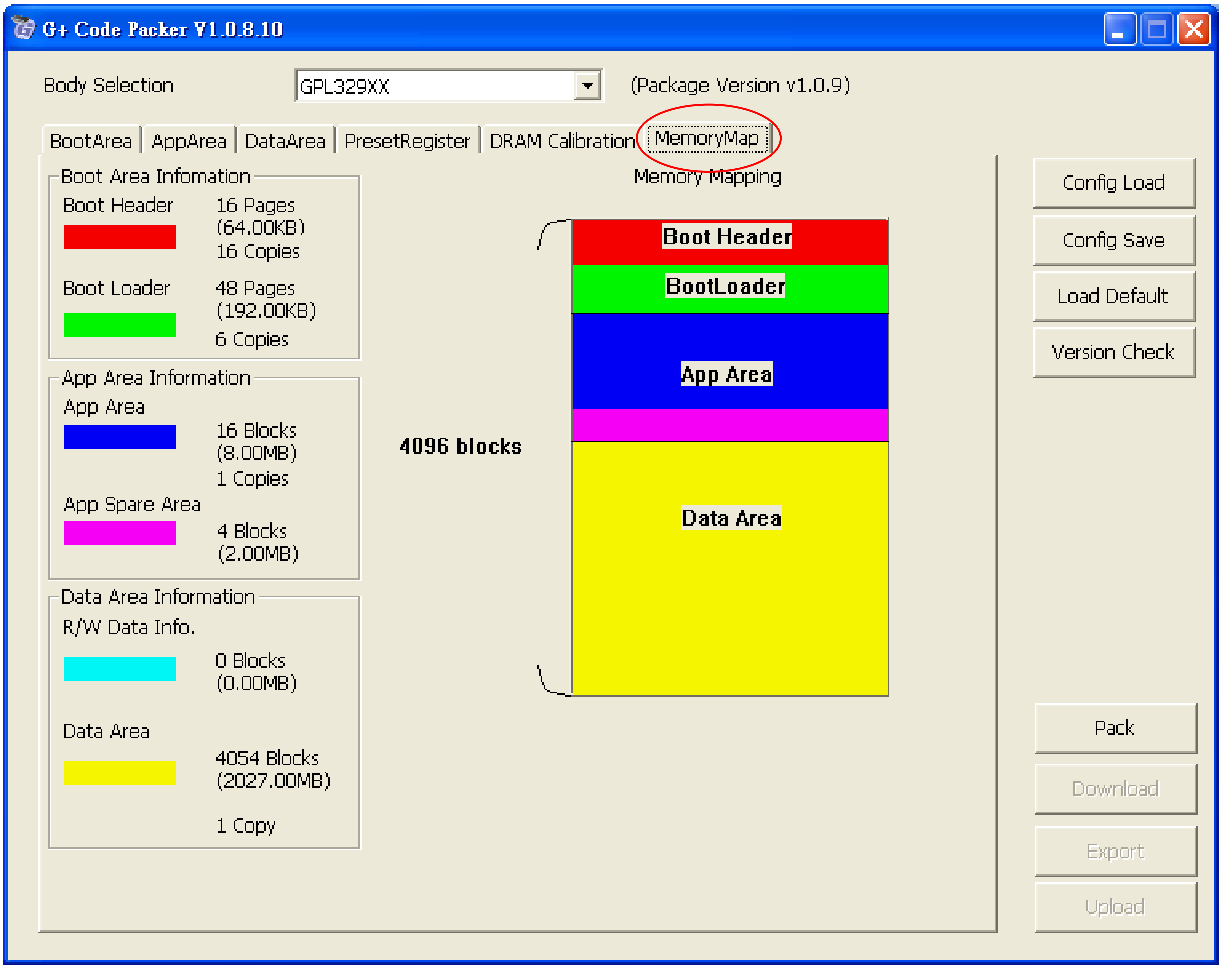
DramReg0 (0x92004000)	0x00000132	DramReg1 (0x92004004)	0x00000000	DramReg2 (0x92004008)	0x08000010
DramReg3 (0x9200400C)	0x0000030D	DramReg4 (0x92004010)	0x080B1833	DramReg5 (0x92004014)	0x4D832000
DramReg6 (0x9200401C)	0x04080432	DramReg7 (0x92004018)	0x00000000	DramReg8 (0x92004020)	0x00000000
DramReg9 (0x92004024)	0x00660044	DramReg10 (0x92004028)	0x28282E2E	DramReg11 (0x92005048)	0x00000000
DramReg12 (0x9200504C)	0x00000000	DramReg13 (0x92005040)	0x00000000	DramReg14 (0x92005010)	0x00000403
DramReg15 (0xD0667801)	0x00000200				

For more information about Pre-set Register & DRAM Calibration, please refer to Chapter 5.

3.2.5 Memory Mapping

Memory Map provides graphical picture to show the user’s memory plan about the NVRAM usage. It clearly expressed the size and how the NVRAM is used for mass production in the future.

Memory Map’s user interface is as follows:



Body Selection: GPL329XX (Package Version v1.0.9)

Boot Area Infomation
- Boot Header 🟥 16 Pages (64.00KB) 16 Copies
- Boot Loader 🟩 48 Pages (192.00KB) 6 Copies

App Area Information
- App Area 🟦 16 Blocks (8.00MB) 1 Copies
- App Spare Area 🟪 4 Blocks (2.00MB)

Data Area Information
- R/W Data Info. 0 Blocks (0.00MB)
- Data Area 🟨 4054 Blocks (2027.00MB) 1 Copy

a. Boot Area Information: The unit is per block or per page and will be converted to MB or KB. The unit of a block size varies depending on the NVRAM type or model.

b. App Area Information: The unit is per block. It depicts the size of read only area. This area shouldn’t be too tight in order to keep the flexibility of code update.

c. Data Area Information: R/W Data Info. (User Reserve Area) and Data Area (File System Area Size -- Part0…Part3).

3.2.6 Configuration Tool Buttons

a. Config Save: Save all settings and designate the config file.

b. Config Load: Recall the stored Config setting via Config Load.

c. Load Default: Reload the tool’s default setting.

3.2.7 Action Buttons

a. Pack: Pack the current ROM image.

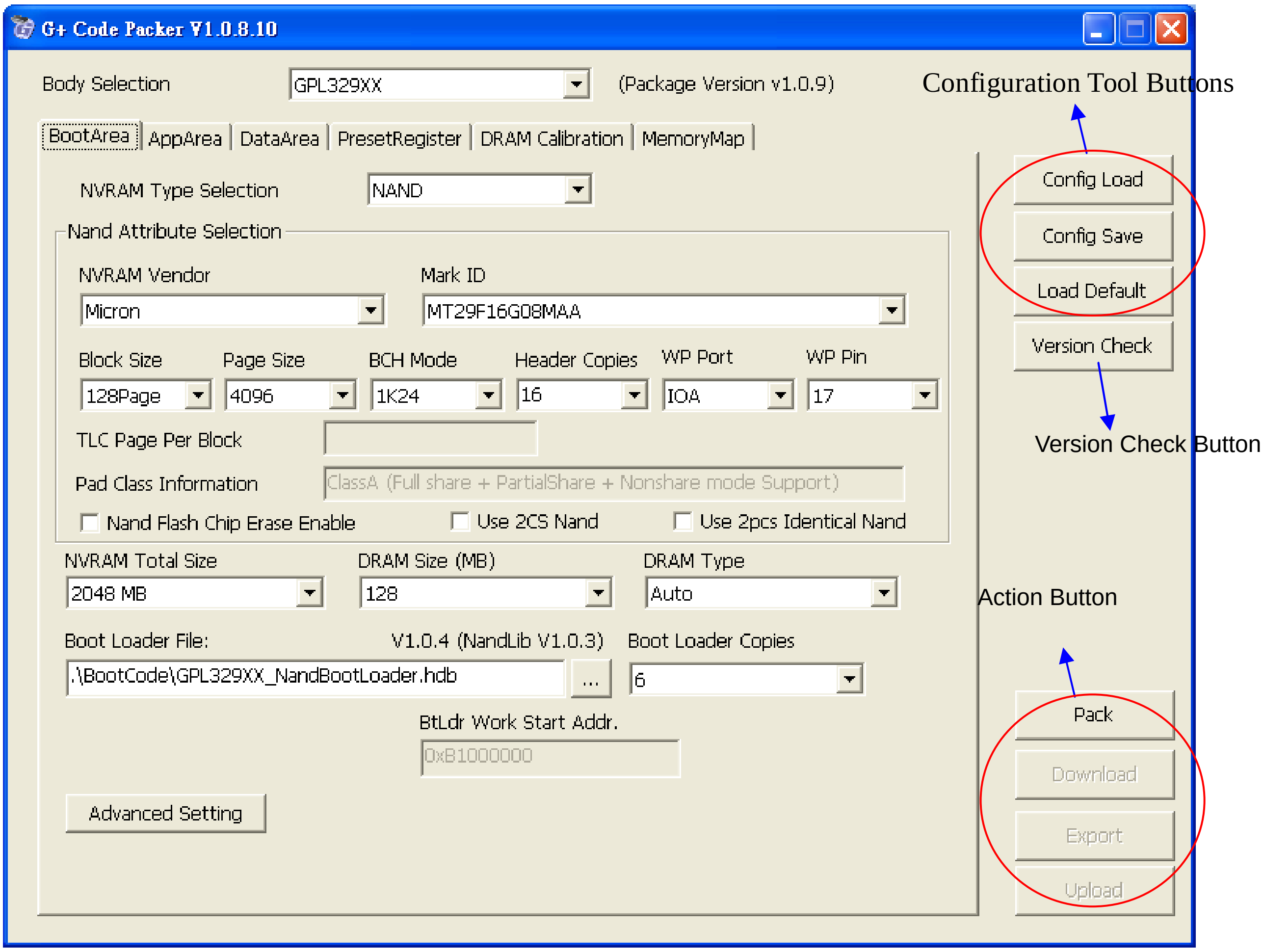
b. Download: Call USM Mass Tool from G+ CodePacker. Download the Packed code on NVRAM via USB.

c. Export: Export the files that downloading needs to the designated folder for mass production.

d. Upload: Read back NAND Flash Image Data and write it to destined folder.

3.2.8 Version Check Button

The version check button is to check the version of pre-set binary file (BootLoader, UsbBin, …) and the NAND library version information. Generalplus strongly suggests the version of NAND library used by tool and development project to be the same to avoid un-predictable problem. If you have question about the version you are using, click this button to get the latest version information of this tool or contact your Generalplus FAE for assistance.



G+ Code Packer V1.0.8.10

Body Selection: GPL329XX (Package Version v1.0.9)

NVRAM Type Selection: NAND

Nand Attribute Selection

NVRAM Vendor: Micron

Mark ID: MT29F16G08MAA

Block Size: 128Page

Page Size: 4096

BCH Mode: 1K24

Header Copies: 16

WP Port: IOA

WP Pin: 17

TLC Page Per Block:

Pad Class Information: ClassA (Full share + PartialShare + Nonshare mode Support)

□ Nand Flash Chip Erase Enable □ Use 2CS Nand □ Use 2pcs Identical Nand

NVRAM Total Size: 2048 MB

DRAM Size (MB): 128

DRAM Type: Auto

Boot Loader File: V1.0.4 (NandLib V1.0.3)

.\BootCode\GPL329XX_NandBootLoader.hdb

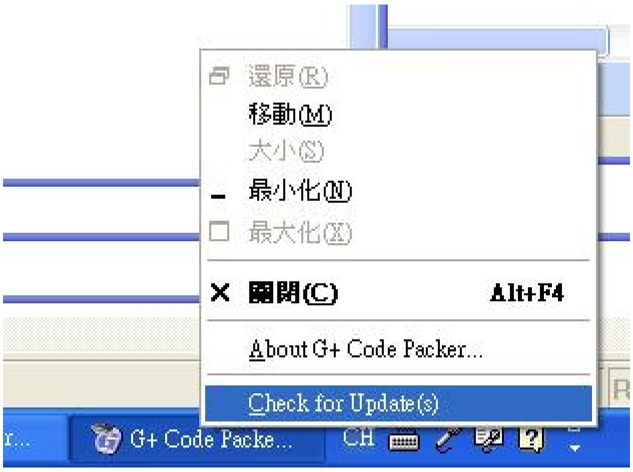
Boot Loader Copies: 6

BtLdr Work Start Addr.: 0xB1000000

3.2.9 Check for Update(s)

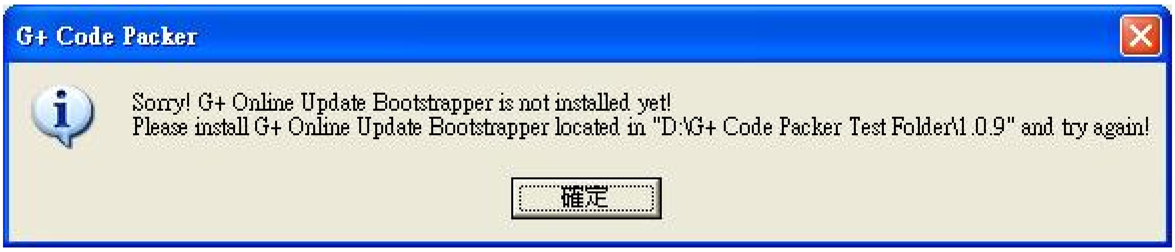
The “Check for Update(s)” button will be seen when user click right button on the G+ Code Packer icon in taskbar. The function is used to check the tool version is newest or not. It also allows the user which has A-Key to download the newest version directly. The operation of the function is described below：

1. The “Check for Update(s)” button will be seen when user click right button on the G+ Code Packer icon in taskbar.



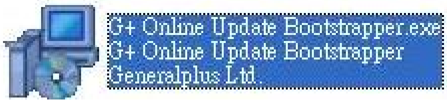
2. Click button to execute the function of “Check for Update(s)”.

It will pop up following message if the tool “SW Tool Online Updater” is not installed.

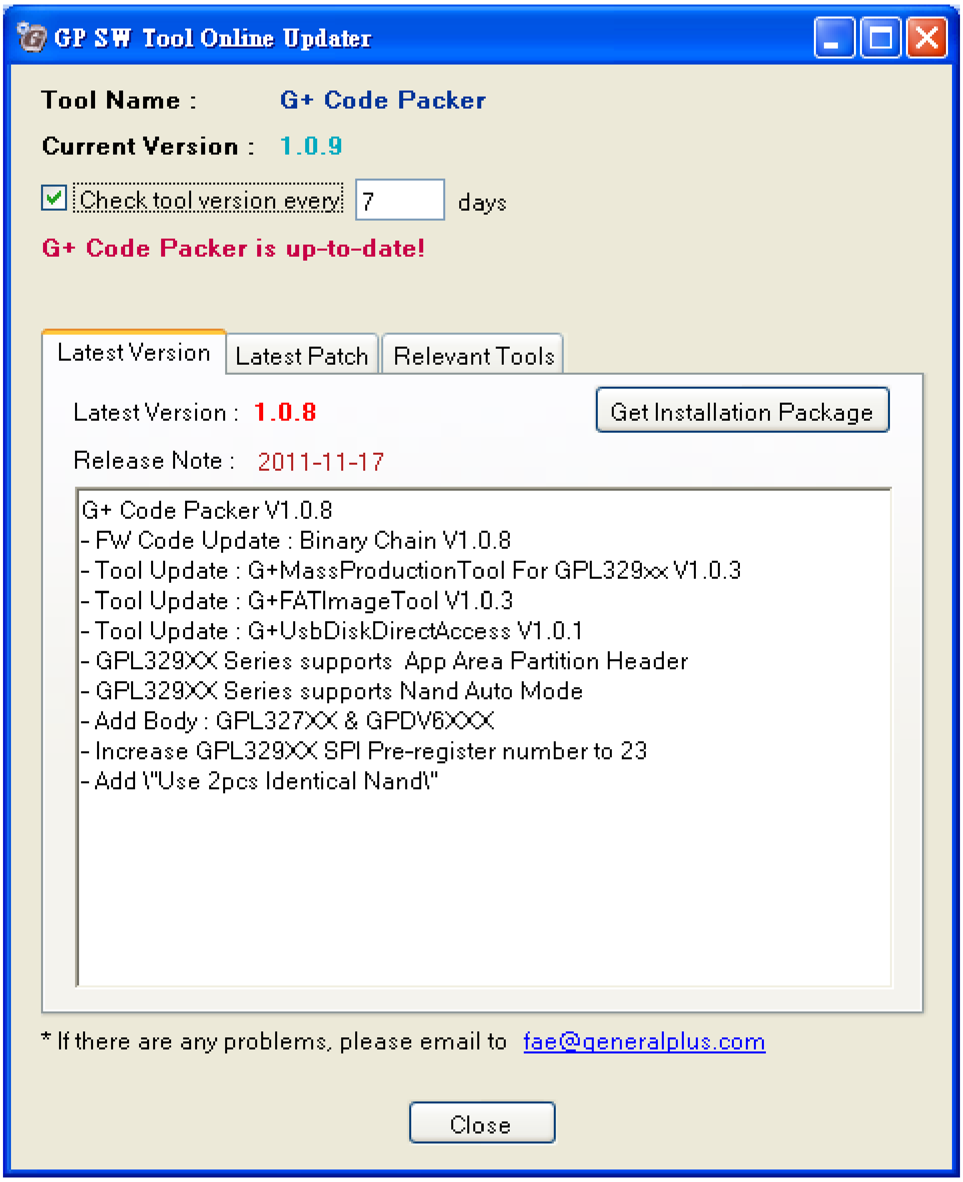

G+ Code Packer

Sorry! G+ Online Update Bootstrapper is not installed yet!
Please install G+ Online Update Bootstrapper located in "D:\G+ Code Packer Test Folder\1.0.9" and try again!

You can execute “G+ Online Update Bootstrapper.exe” in the folder mentioned in the message to install the tool “SW Tool Online Updater”.


G+ Online Update Bootstrapper.exe
G+ Online Update Bootstrapper
Generalplus Ltd.

It will pop up the user interface of “SW Tool Online Updater” if the tool is installed. You can get the information of current tool version、latest version、latest release date … and so on. You can also set the auto check version period. The user which has A-Key can press “Get Installation Package” button to download the newest version when the current tool version is out of date.



Tool Name : G+ Code Packer

Current Version : 1.0.9

☑ Check tool version every 7 days

G+ Code Packer is up-to-date!

Latest Version : 1.0.8

Release Note : 2011-11-17

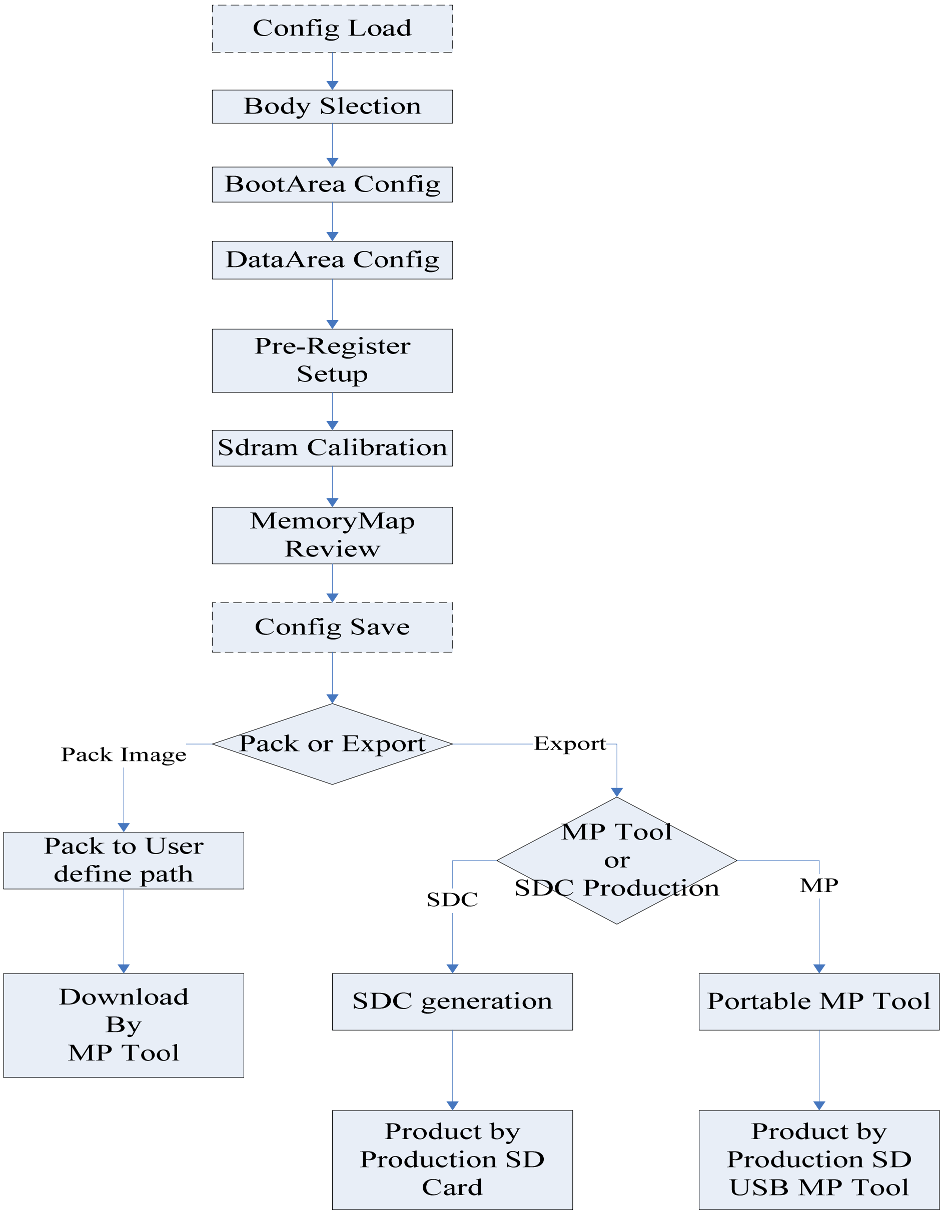
```
G+ Code Packer V1.0.8
- FW Code Update : Binary Chain V1.0.8
- Tool Update : G+MassProductionTool For GPL329xx V1.0.3
- Tool Update : G+FATImageTool V1.0.3
- Tool Update : G+UsbDiskDirectAccess V1.0.1
- GPL329XX Series supports  App Area Partition Header
- GPL329XX Series supports Nand Auto Mode
- Add Body : GPL327XX & GPDV6XXX
- Increase GPL329XX SPI Pre-register number to 23
- Add \"Use 2pcs Identical Nand\"
```

* If there are any problems, please email to fae@generalplus.com

4 NAND Flash Boot Operation (GPL325XX/GPL326XX)

G+ Code Packer divides the NAND Flash into three areas: Boot Area, App Area, and data Area. The resources in the three areas can be set up individual in order to integrate into BIN file (ROM image). In addition, tool supports pre-set register and use SDRAM calibration tool to test SDRAM parameters and load them into the programming file.

The following chapters will introduce Boot Area, App Area, Data Area, and Pre-set Register sets, SDRAM Calibration, Pack and Download flow.



G+ Code Packer Basic Operation Flow

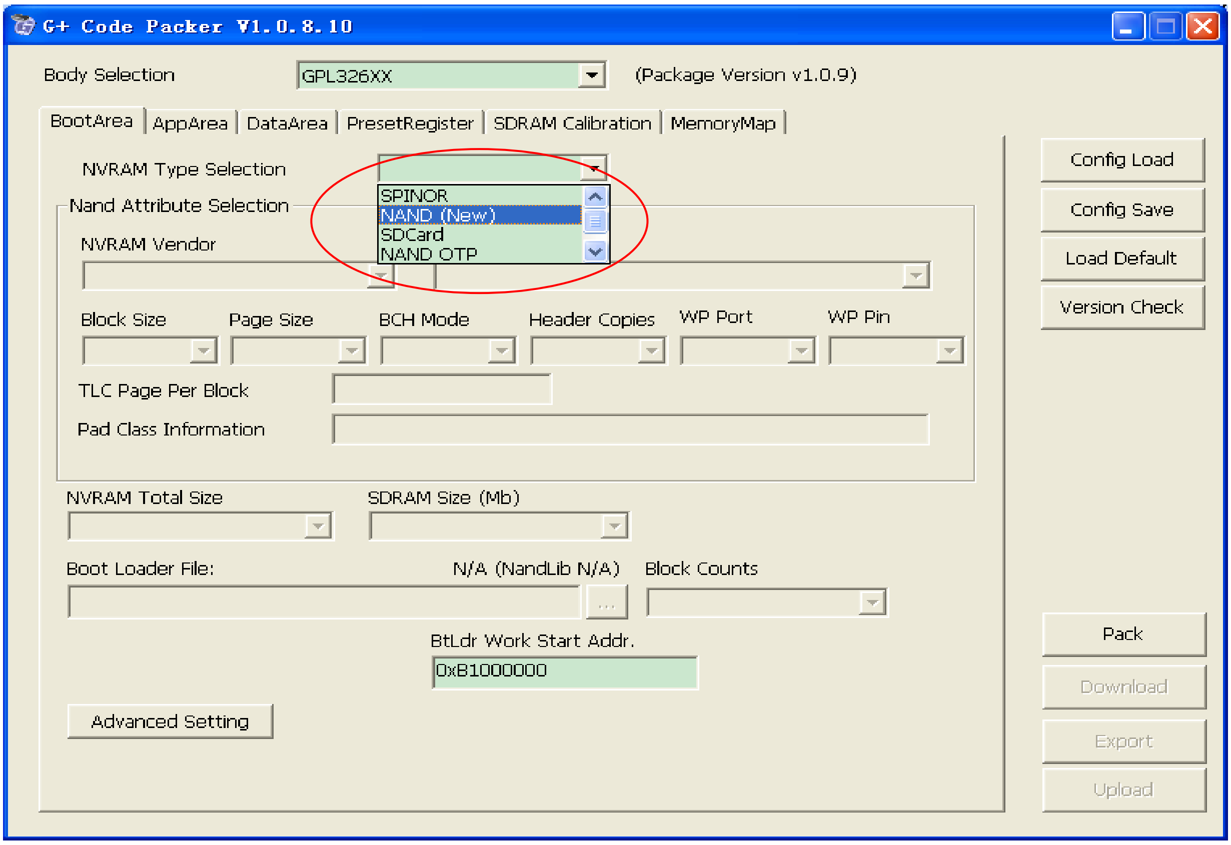
4.1 Load Config & Config Save

If Config Save is saved, those settings and parameters can be reloaded via Config Load. Generalplus highly recommends storing all resource files under the G+CodePacker's root directory. In such way, it enhances the portability and transferability from user to associate partner.

4.2 Boot Area Setup

Step1: NVRAM Type Selection

Select NVRAM(Non-Volatile RAM) type: supports SPINOR, NAND or SD Card (only for GPL326XX).



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

BtLdr Work Start Addr.: 0xB1000000

Step2: NAND Attribute Selection

If SPINOR is selected in step 1, ignore the step 2. If NAND is chosen in step1, fill in the NAND Attribute in step 2. These include NAND Vendor ID (manufacture ID), NAND Mark ID (Product ID), Block Size, Page Size, Boot Area's BCH mode, and number of copies for Boot Area's BCH Table.

Most of NAND flash on the market of which vendor's ID can be found according to the Mark ID. If any NAND is not in the list, user needs to manually fill in the block size and page size based on the NAND flash specification (please select manual mode for NVRAM vendor). Besides the manual mode, G+CodePacker will list the appropriate total NVRAM size for you through NAND mark ID. This cell can not be modified. For SDRAM size, user needs to fill in the proper NVRAM size. This is very important for SDRAM calibration for later use.

Step3：Boot Header BackUp NAND Block Numbers

After mark ID and SDRAM size are filled in, it is possible to change the number of header copy if it is a good NAND (for those not re-used NAND flash, we recommend keeping it as manufacture's default setting).

Step4：Write Protect Pin Assign (depended on HW design)

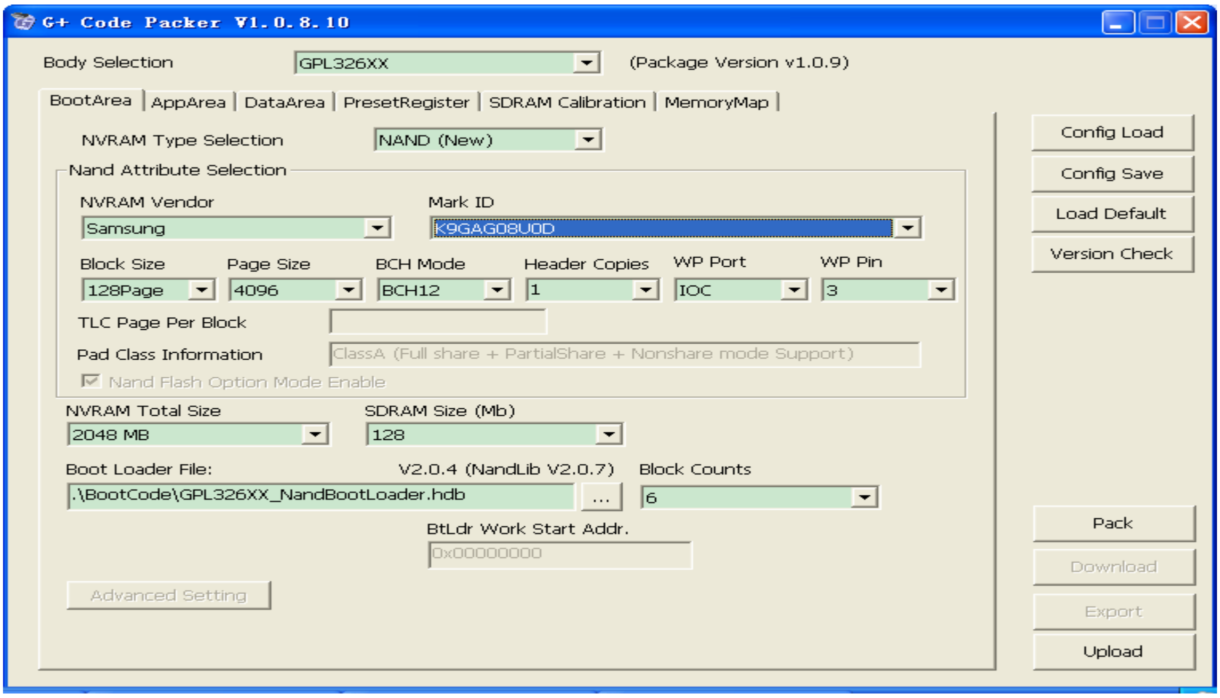
Write Protect Pin is also an important parameter. In most development boards, NAND write protect pin is pulled high. In this case, the parameter can be set NULL. Others are set based on how the circuitry is designed. IOC3 is the Generalplus default setting. The G+CodePacker also provides the others IO setting and user must set it up appropriately.

Step5: Boot Loader File Choice

We highly recommend applying the default Boot Loader File provided by Generalplus. For special use, unique Boot Loader may be provided by user. The Loader file, the first program in GPL32XXX ROM code, will be brought into Generalplus Management area, including user's Runtime code or the APP initialization of read only resource. The FAT initialization is also executed here. This procedure is very important to the users who use the Generalplus standard platform. For best result in stability and correctness, the Generalplus standard NAND Driver Library is coincident with the NAND library used in the Runtime code.

Step 6: Boot Loader File Backup Count (base on NAND Block nums)

Boot Loader Block Count is the number of copied blocks for Boot Loader. We suggest make several copied blocks, unit per NAND Block Size. Suppose Block size is 512KB, and NANDBootLoader is 35KB, a copied block can store 512 / 35 = 14 (bin counts/Block). Add or delete may be applied if necessary. (BootLoader is protected by BCH)



Body Selection: GPL326XX (Package Version v1.0.9)

NVRAM Type Selection: NAND (New)

Nand Attribute Selection

NVRAM Vendor: Samsung

Mark ID: K9GAG08U0D

Block Size: 128Page

Page Size: 4096

BCH Mode: BCH12

Header Copies: 1

WP Port: IOC

WP Pin: 3

TLC Page Per Block:

Pad Class Information: ClassA (Full share + PartialShare + Nonshare mode Support)

☑ Nand Flash Option Mode Enable

NVRAM Total Size: 2048 MB

SDRAM Size (Mb): 128

Boot Loader File: V2.0.4 (NandLib V2.0.7)

.\BootCode\GPL326XX_NandBootLoader.hdb ...

Block Counts: 6

BtLdr Work Start Addr.: 0x00000000

4.3 App Area Setup

Step1: App Area parameter setup

When NVRAM Type is NAND, user needs to set up the APP parameters. If NVRAM Type is SPINOR, skip Step 1. The App parameters include: App Man Size, App Spare Rate, App ROM Code Copies, and CRC Check Sum.

App Main Size

App Main’s Size must be greater than the sum of ROM Code’s Size and Resource File’s Size.

App Spare Rate

App Area’s block spare rate. To prevent App Area from void block or data loss, a spare rate is featured. The formula is as follows:

$$AppSpareSize = \frac{AppMainSize}{AppSpareRate}$$ (unit: NAND Blocks)

Example:

Block Size = 64, Page Size=2048, App Main Size = 16MB (=128 blocks), App Spare Rate = 5; then, App Spare Size = 128 blocks / 5 = 25.6 ≒ 26 blocks = 3.25MB

The limitation of max. and min. App Spare Sizes (unit in block) are:

Max. = Page Size / 4, Min. = 4.

Thus, legal values must fulfill the following two conditions:

1. 4 ≦ App Spare Size ≦ Page Size / 4, unit in block.
2. App Spare Size ≦ App Main Size

In the same example as above, to fulfill these two conditions, unit in block:

4 ≦ App Spare Size ≦ 2048 / 4

App Spare Size ≦ 128

Thus, the legal range of App Spare Size is:

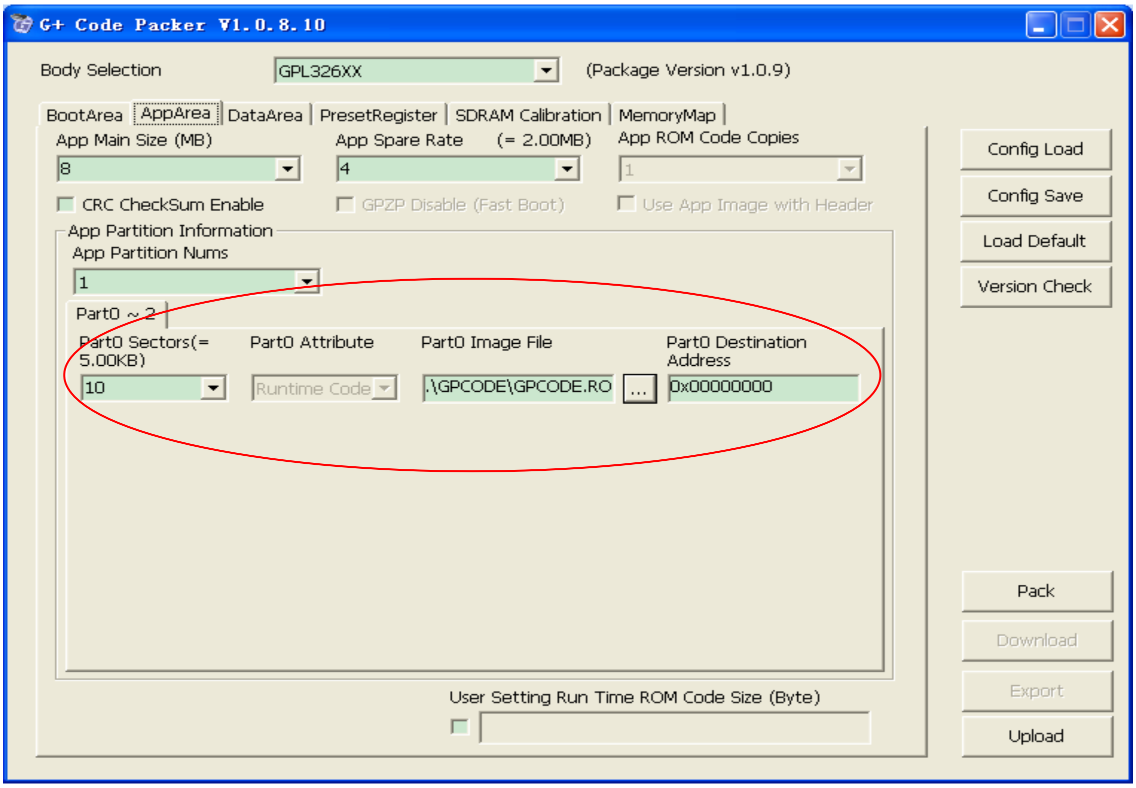
4(=0.5MB) ≦ App Spare Size ≦ 128(=16MB), meaning 1 ≦ App Spare Rate ≦ 32

Therefore, the legal range of App Spare Rate is 1 ~ 32; the corresponding App Spare Size is 4 ~ 128 blocks (=0.5 ~ 16MB).

Step2: Setting Run Time Rom Code Size and Run Time ROM Code

After Boot Loader is executed, it will copy the Run Time ROM code (chosen by user) into SDRAM and starts running. If "Setting Run Time ROM Code Size" is not checked off, the copy size will be determined by the tool according to the actual size of Run Time ROM Code. If "Setting Run Time ROM Code Size" is checked off, the copy size is equal to the given size. (unit: Byte)

Caution: If size is given incorrectly, program will not be able to execute.


G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

App Main Size (MB): 8

App Spare Rate (= 2.00MB): 4

App ROM Code Copies: 1

☐ CRC CheckSum Enable

☐ GPZP Disable (Fast Boot)

☐ Use App Image with Header

App Partition Information

App Partition Nums: 1

Part0 ~ 2

Part0 Sectors(= 5.00KB): 10

Part0 Attribute: Runtime Code

Part0 Image File: .\GPCODE\GPCODE.RO ...

Part0 Destination Address: 0x00000000

User Setting Run Time ROM Code Size (Byte): ☐

As indicated in the diagram above, this page is the configuration about App Partition. One is the number of areas and the other is its corresponding parameters about each area.

The message parameters for each area are as follows:

- App Partition Num: This option means the number of sub-area for App area. It has two areas, ranged 1 ~ 2, meaning 2 sub-areas available at most.
- Partition N Size: N ranges 0 ~ 1, representing the size of sub-area, unit in sector. Usually, after we select the imaging file, tool will automatically fill in the corresponding file size to this field. Under one circumstance, when user does not select file, this field will be set as auto. If user intends to specify the size of the sub-area, simply key in the value.
- Partition N Attribute: Similarly, N ranges 0~1, representing the type of current imaging file. In present, Tool offers 2 types of imaging file: Runtime Code, Resource Part. Also, it is in disable state, meaning user is temporary not able to change it. The first sub-area is for storing Runtime Code and second area is for the resource files for Runtime code to use. It is important, especially when First bootloader is moving the code from App area.
- Partition N Image file: this parameter is used for selecting the sub-area’s imaging file on the disk.
- Partition N Destination Address: This parameter is to specify SDRAM location for the current sub-area’s file to be placed.。

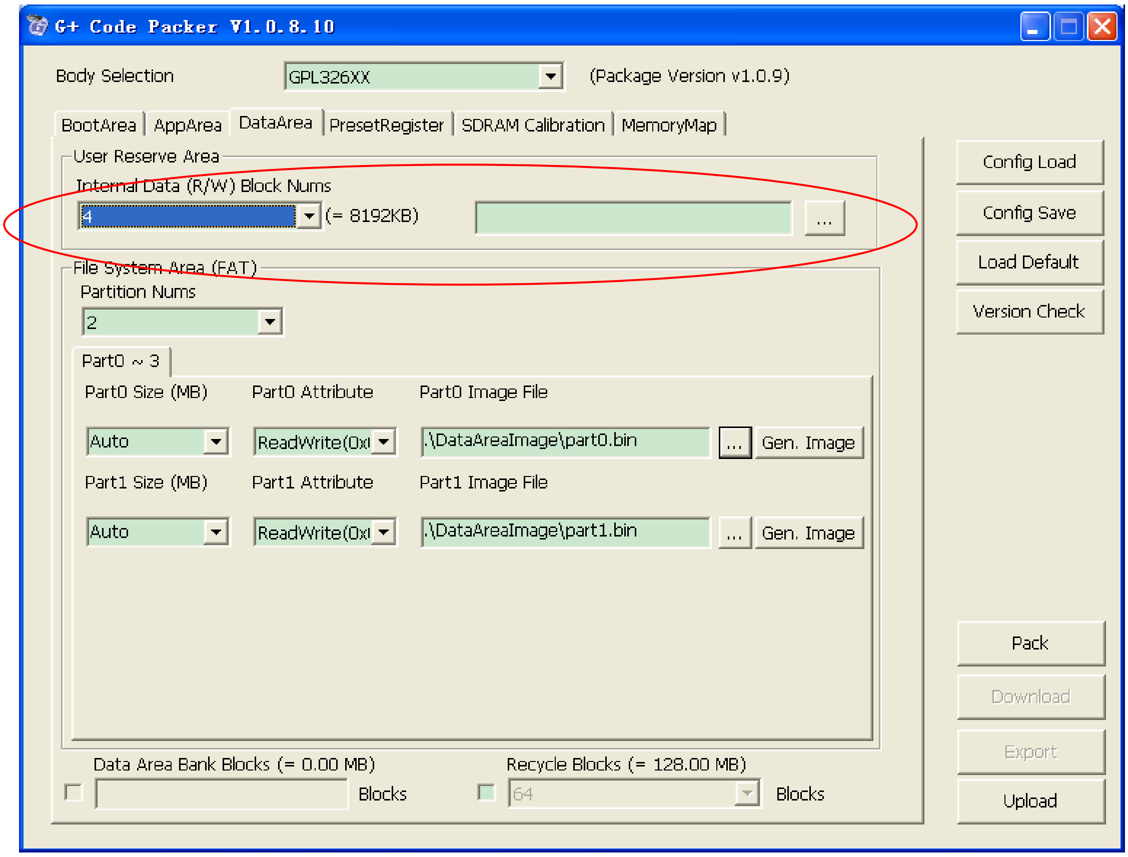
Based on this method, configure and set up all the sub-areas until all areas’ messages are configured completely.

4.4 Data Area Setup and Usage of FAT Image Tool

4.4.1 Data Area Setup

Step1: User Reserve Area Size and Image

User Reservation Area is to store App ROM code R/W data. The unit size of User Reservation Area is per Block. After the choice is made, calculation of reservation area size (unit: Kbytes) is shown immediately after the choice made. Therefore, user can easily get the actual size used. R/W area accepts the pre-set data (pre-load image is required by user). The pre-set data can be read/write by the API in APP Area provided by Generalplus.



G+ Code Packer V1.0.8.10

Body Selection GPL326XX (Package Version v1.0.9)

User Reserve Area

Internal Data (R/W) Block Nums 4 (= 8192KB)

File System Area (FAT)

Partition Nums 2

Part0 Size (MB) Auto

Part0 Attribute ReadWrite(0x

Part0 Image File .\DataAreaImage\part0.bin

Part1 Size (MB) Auto

Part1 Attribute ReadWrite(0x

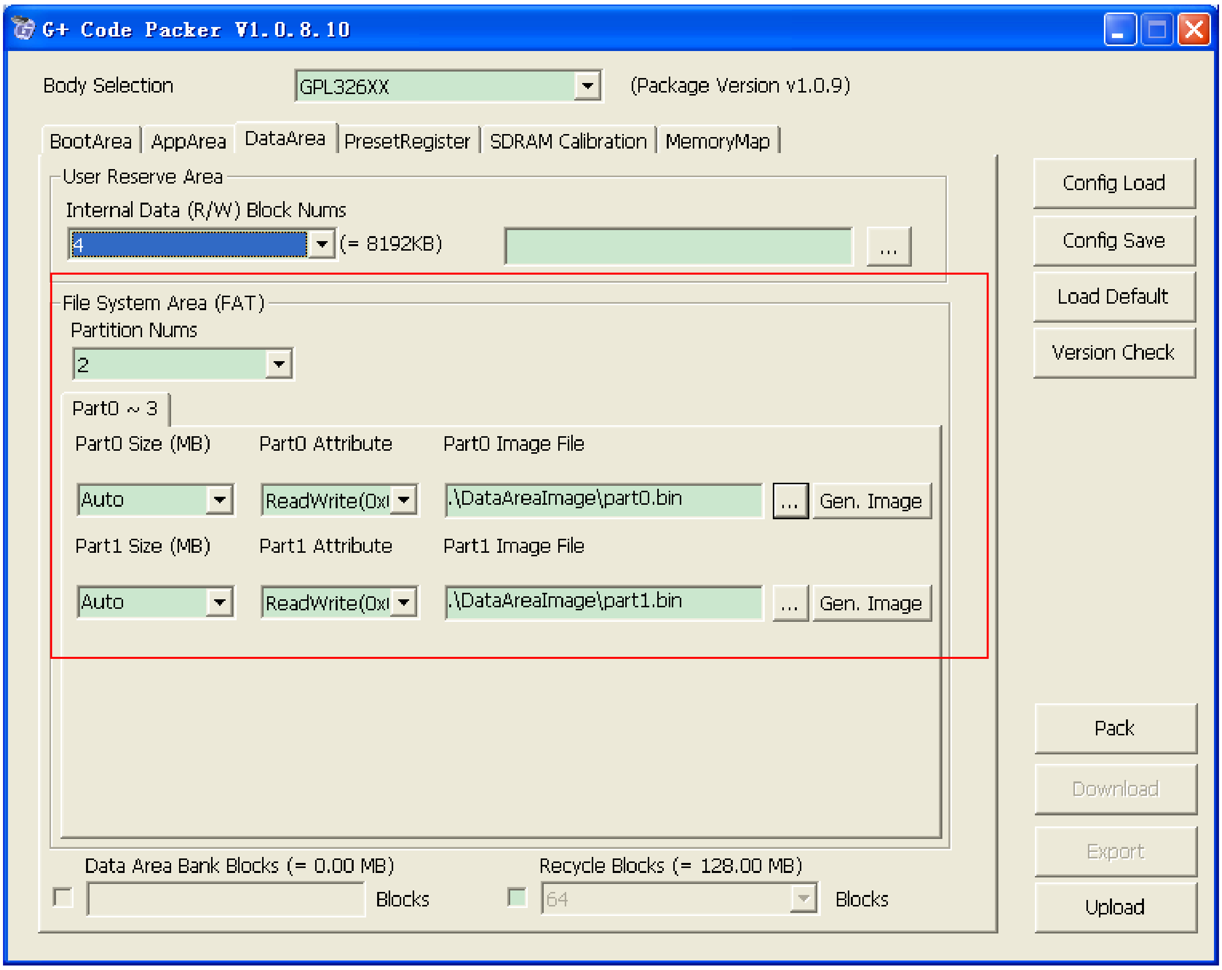
Part1 Image File .\DataAreaImage\part1.bin

Data Area Bank Blocks (= 0.00 MB) □ Blocks

Recycle Blocks (= 128.00 MB) □ 64 Blocks

Step2: File System Area Setup

File System Area defines the size and corresponding image files for each area. SPINOR only supports one File System area. The NAND Flash supports up to two File System areas. User can plan these areas based on their needs. The size for each area can be determined based on needs. The basic unit is per Mbytes. If Auto is selected, tool will determine the area size based on the image file sizes. In Attribute, three options are: Read Only, Write Only, and Read/Write.



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

User Reserve Area
Internal Data (R/W) Block Nums: 4 (= 8192KB)

File System Area (FAT)
Partition Nums: 2

Part0 Size (MB): Auto
Part0 Attribute: ReadWrite(0x(
Part0 Image File: .\DataAreaImage\part0.bin

Part1 Size (MB): Auto
Part1 Attribute: ReadWrite(0x(
Part1 Image File: .\DataAreaImage\part1.bin

Data Area Bank Blocks (= 0.00 MB) Blocks
Recycle Blocks (= 128.00 MB) 64 Blocks

Step3: Advanced Recycle Blocks (Option Choice) Setup

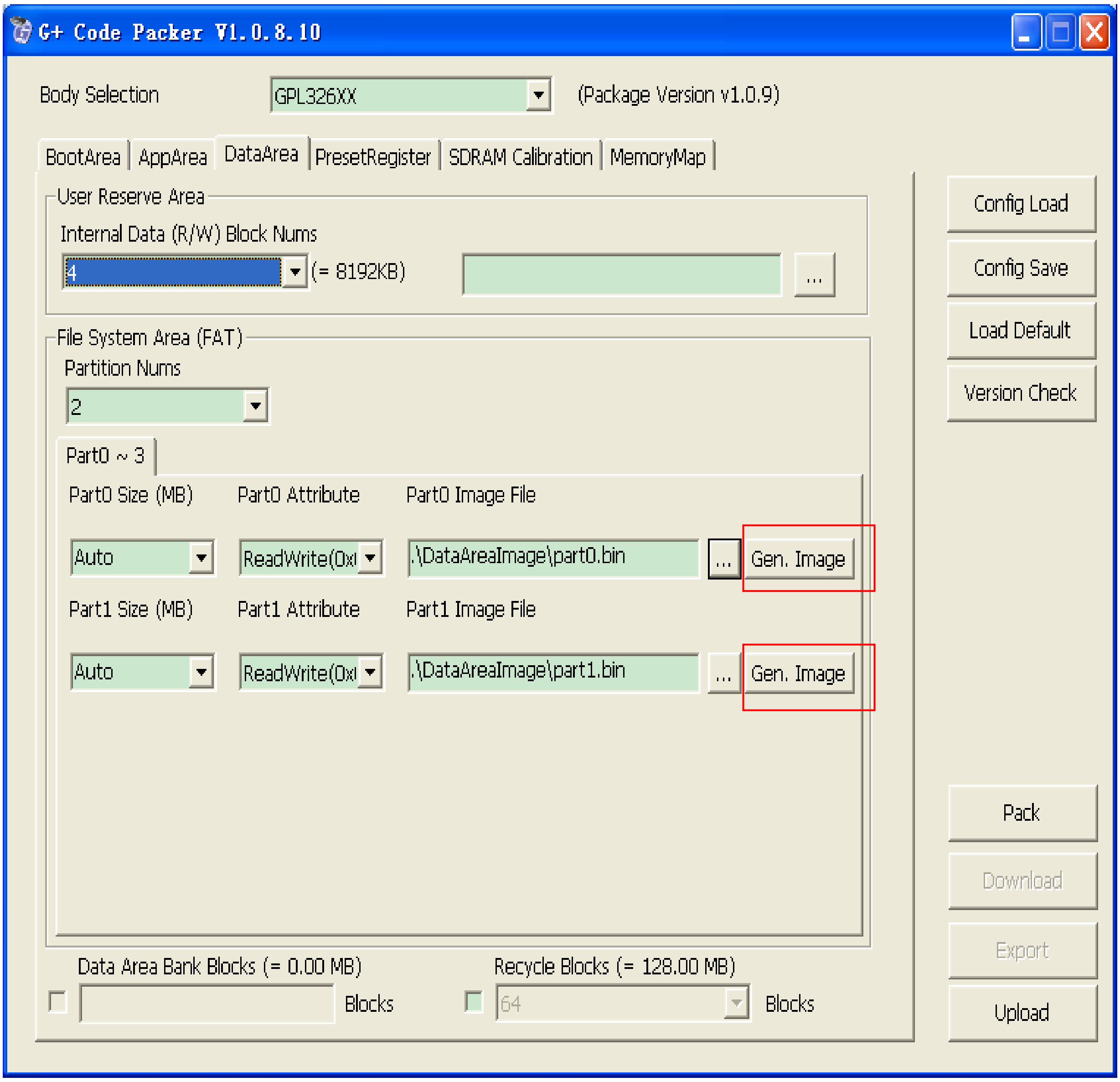
Generalplus highly recommends using the its pre-set (default) values for your setting. This is the advanced settings for Data Area exchange Block number **of each Bank**. The default bank size is 512 blocks which is not allowed to be changed at this moment.

4.4.2 How to use FAT Image Tool

In addition to File System Area has option to select image file, it also integrates useful tools for image production. Flows are as follows:

Step1: Run FAT Image Tool

The image made by FAT Image Tool will feedback its filename and Partition Size back to the tool. Thus, we should first select the necessary Partition Number, click the “Gen. Image” button after partition, run FAT Image Tool and then proceed the image production.



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

User Reserve Area

Internal Data (R/W) Block Nums: 4 (= 8192KB)

File System Area (FAT)

Partition Nums: 2

Part0 Size (MB): Auto | Part0 Attribute: ReadWrite(0x(| Part0 Image File: .\DataAreaImage\part0.bin

Part1 Size (MB): Auto | Part1 Attribute: ReadWrite(0x(| Part1 Image File: .\DataAreaImage\part1.bin

Data Area Bank Blocks (= 0.00 MB) □ Blocks

Recycle Blocks (= 128.00 MB) □ 64 Blocks

Step2：FAT Image Tool setup and making Image file

After running FAT Image Tool, set up the follows: FAT Disk Size, Cluster Size, FAT Image Source Folder, and FAT Image Output Path.　Please be aware of the followings:

3. **FAT Disk Size**

   The partition size after produced image is set up.　The size must be greater than the size of source folder selected.　Or it will be failed during the image production.

4. **Cluster Size**

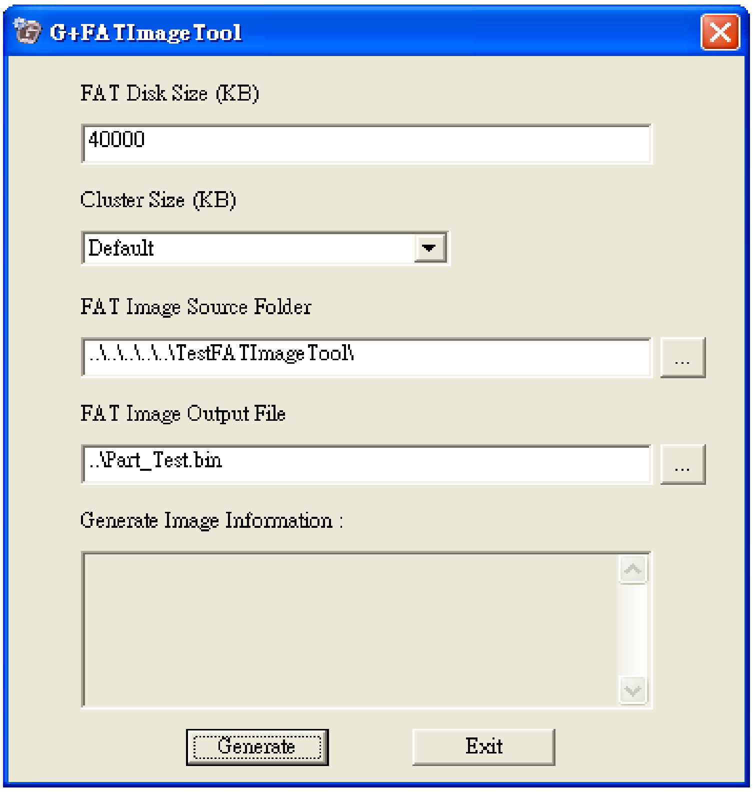
   Set up Cluster Size.　We recommend using the default value for this option.　If size is incorrect given, it may cause failure during image production or even a image file is successfully produced but it cannot be used or recognized by system.

5. **FAT Image Source Folder**

   The source folder for image produced.　Tool will make the folder to an image file, including all the files under the folder, structure of folder and the files under sub-folder.

6. **FAT Image Output Path**

   Set up the produced image's file path and filename.



G+FATImageTool

FAT Disk Size (KB)
40000

Cluster Size (KB)
Default

FAT Image Source Folder
..\..\..\..\..\TestFATImageTool\

FAT Image Output File
..\Part_Test.bin

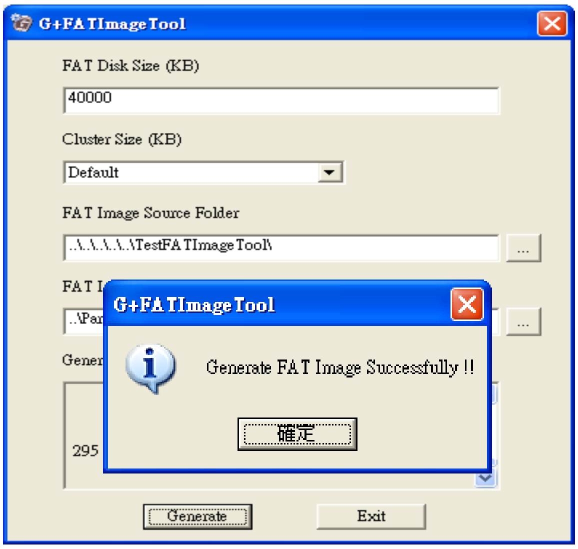
Generate Image Information :

Click "Generate" button after completing configuration and starts to make image file.　During image production, we can view the progress, to which folder or file through "Generate Image Information" window.

Note: It is not allowed having Simplified Chinese filename or folder name under the Traditional Chinese Operating System and vice versa; otherwise the image file production will be failed.

Step3：Bring information into Code Packer automatically

After image file is completed, the "Generate FAT Image Successfully" window pops out and click OK to close FAT Image Tool. Relevant information will be auto-loaded into Code Packer and ends the image production.



G+FATImageTool

FAT Disk Size (KB): 40000

Cluster Size (KB): Default

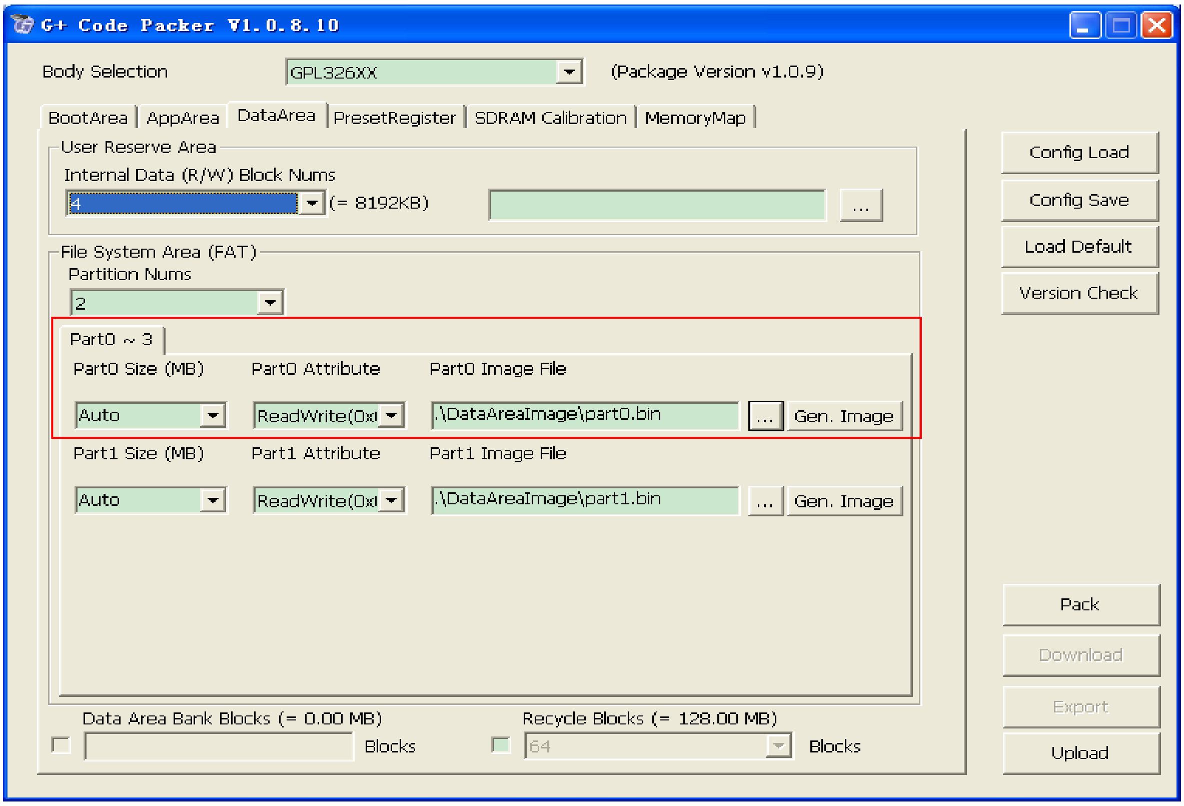
FAT Image Source Folder: ..\..\..\..\..\TestFATImageTool\

FAT I ..\Par

Gener 295

G+FATImageTool

Generate FAT Image Successfully !!



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

User Reserve Area

Internal Data (R/W) Block Nums: 4 (= 8192KB)

File System Area (FAT)

Partition Nums: 2

Part0 ~ 3

Part0 Size (MB)	Part0 Attribute	Part0 Image File
Auto	ReadWrite(0x	.\DataAreaImage\part0.bin

Part1 Size (MB)	Part1 Attribute	Part1 Image File
Auto	ReadWrite(0x	.\DataAreaImage\part1.bin

□ Data Area Bank Blocks (= 0.00 MB) Blocks

□ Recycle Blocks (= 128.00 MB) 64 Blocks

NOTE:

There is a special case when user sets the disk size smaller than 17MB in FAT Image Tool. The special case will be described below：

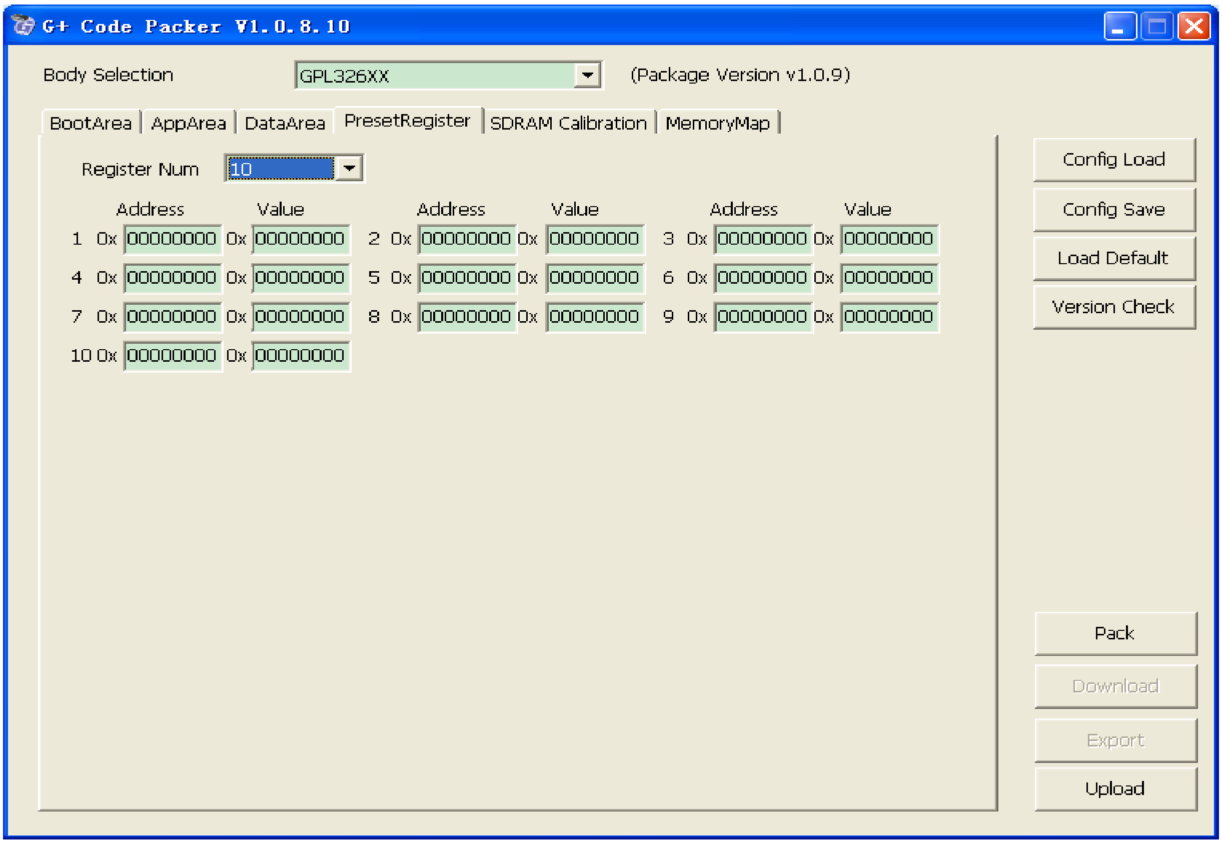
1. **The tool will simulate the disk size to 17MB and format to FAT16 when user sets the disk size smaller than 17MB. The real size of the image is still the size setting by user.**
2. **The tool will also simulate the disk size to 17MB and format to FAT16 when user sets the disk size smaller than 17MB and check the MBR option.**

The special case will cause the size shown in Data Area user interface is always 17MB when the image is generated by FAT Image Tool and the disk size is set smaller than 17MB.

4.5 Pre-set Register and SDRAM Calibration Setup

4.5.1 Pre-set Register setup

Users are able to set up registers based on their needs. SDCard and SPI Flash only support up to 10 setups for registers. NAND Flash supports up to 30 setups. Each setup requires Register address and Register value.



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

Register Num: 10

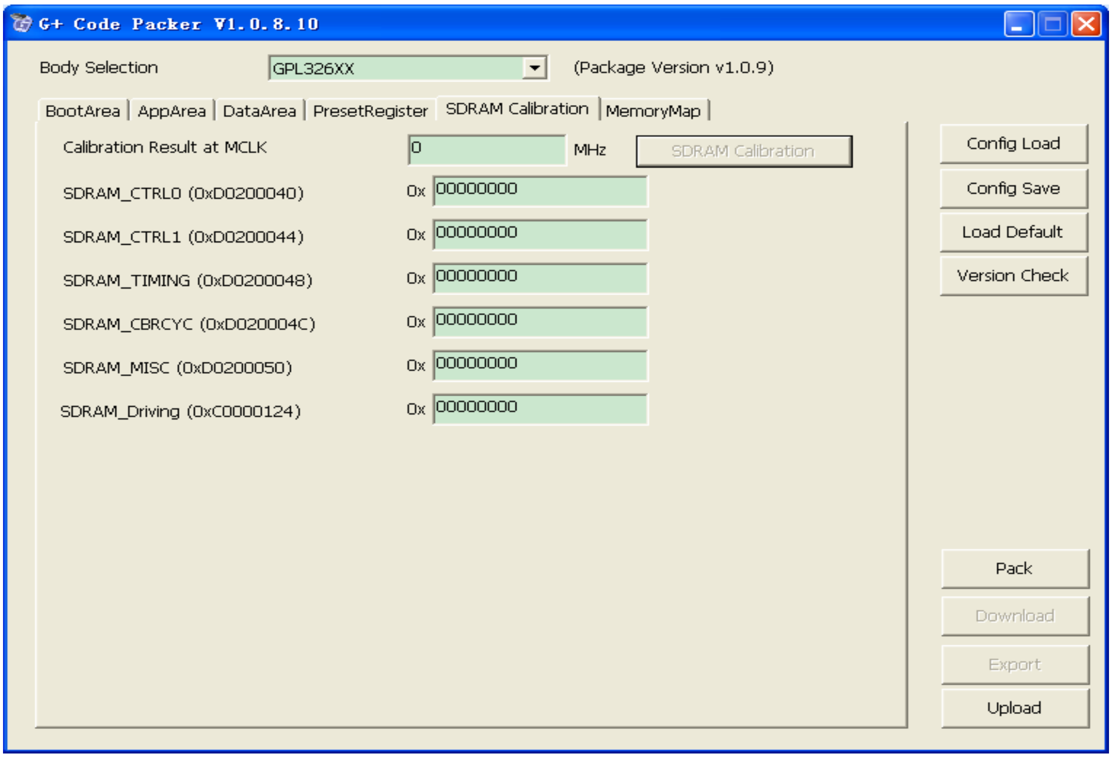
	Address	Value		Address	Value		Address	Value
1	0x 00000000	0x 00000000	2	0x 00000000	0x 00000000	3	0x 00000000	0x 00000000
4	0x 00000000	0x 00000000	5	0x 00000000	0x 00000000	6	0x 00000000	0x 00000000
7	0x 00000000	0x 00000000	8	0x 00000000	0x 00000000	9	0x 00000000	0x 00000000
10	0x 00000000	0x 00000000						

4.5.2 SDRAM Calibration

Step1: Click SDRAM Calibration button

Click SDRAM calibration button to activate SDRAM calibration tool. Connect your development/target board to your PC via USB. Through internal ROM Code boot, let the hardware enters USB Service Mode (IOB3 must Pull High and reset IC). If the hardware is detected by the tool, “Device connected” is shown. If not, “Device Not Found” is shown. All calibration must be performed under “Device Connected” successfully.

SDRAM Calibration Entry for GPL326XX:



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

Calibration Result at MCLK: 0 MHz

SDRAM_CTRL0 (0xD0200040) 0x 00000000

SDRAM_CTRL1 (0xD0200044) 0x 00000000

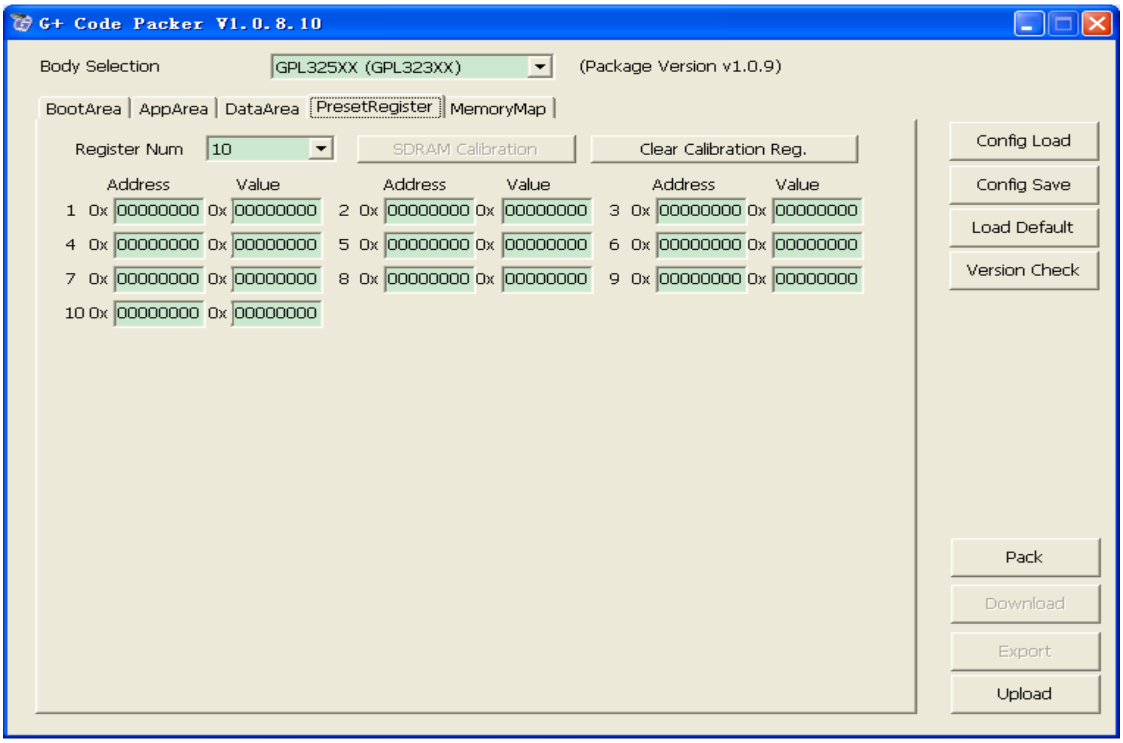
SDRAM_TIMING (0xD0200048) 0x 00000000

SDRAM_CBRCYC (0xD020004C) 0x 00000000

SDRAM_MISC (0xD0200050) 0x 00000000

SDRAM_Driving (0xC0000124) 0x 00000000

SDRAM Calibration Entry for GPL325XX:



G+ Code Packer V1.0.8.10

Body Selection: GPL325XX (GPL323XX) (Package Version v1.0.9)

Register Num: 10

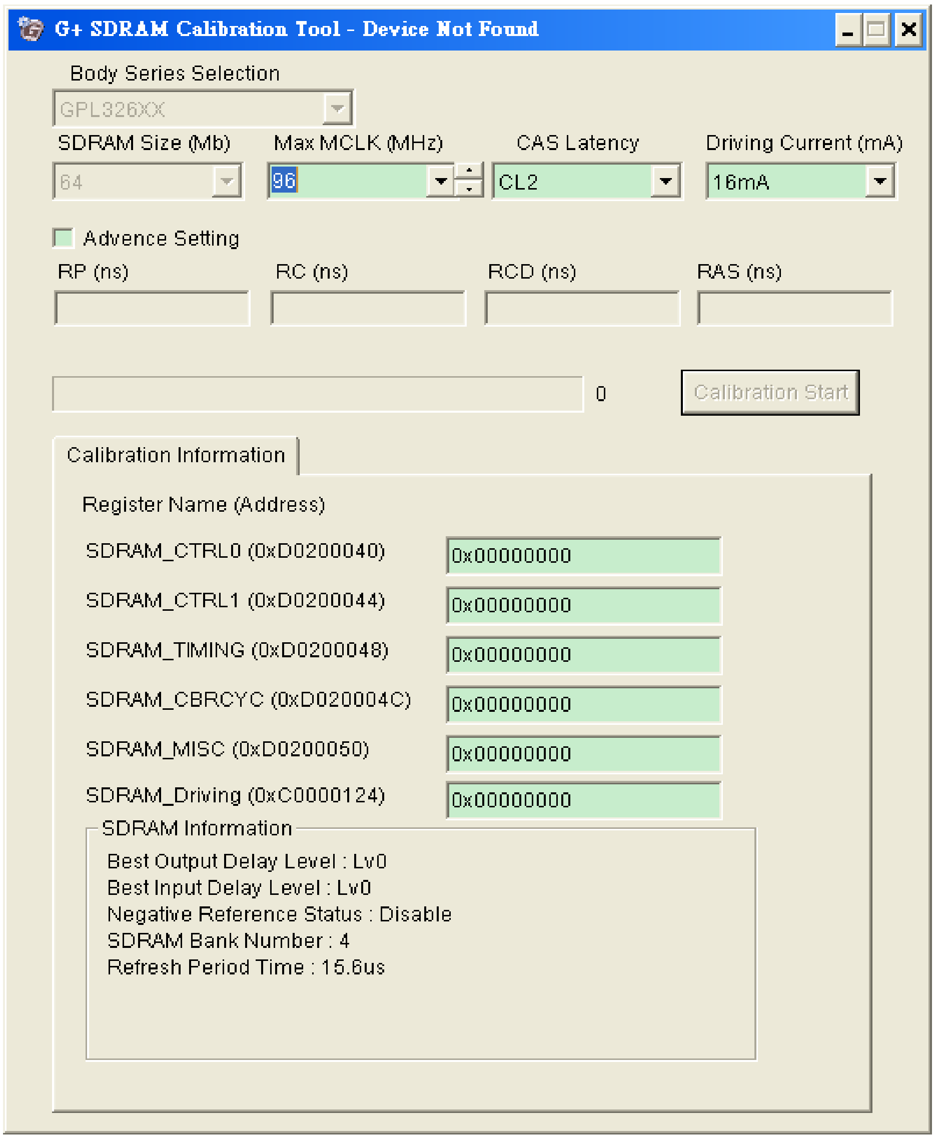
	Address	Value		Address	Value		Address	Value
1	0x 00000000	0x 00000000	2	0x 00000000	0x 00000000	3	0x 00000000	0x 00000000
4	0x 00000000	0x 00000000	5	0x 00000000	0x 00000000	6	0x 00000000	0x 00000000
7	0x 00000000	0x 00000000	8	0x 00000000	0x 00000000	9	0x 00000000	0x 00000000
10	0x 00000000	0x 00000000						

Step 2: Body Selection & SDRAM Size

To run G+ SDRAM calibration individually, please select these two parameters.

Step 3: Entry Calibration Tools (sub tool)

Please select the system operating frequency and SDRAM driving capability. In our experience, EMU board should set higher driving level (16mA or higher). However, in production board, it can be set to 8mA due to the EMI issue (the default is 16mA, suitable for most of development environment). Various driving currents may result in different calibration consequences.



G+ SDRAM Calibration Tool - Device Not Found

Body Series Selection: GPL326XX

SDRAM Size (Mb): 64

Max MCLK (MHz): 96

CAS Latency: CL2

Driving Current (mA): 16mA

□ Advence Setting

RP (ns):

RC (ns):

RCD (ns):

RAS (ns):

0

Calibration Information

Register Name (Address)	
SDRAM_CTRL0 (0xD0200040)	0x00000000
SDRAM_CTRL1 (0xD0200044)	0x00000000
SDRAM_TIMING (0xD0200048)	0x00000000
SDRAM_CBRCYC (0xD020004C)	0x00000000
SDRAM_MISC (0xD0200050)	0x00000000
SDRAM_Driving (0xC0000124)	0x00000000

SDRAM Information

Best Output Delay Level : Lv0
Best Input Delay Level : Lv0
Negative Reference Status : Disable
SDRAM Bank Number : 4
Refresh Period Time : 15.6us

Step4: Select Max. Sys clock, CAS Latency, and Driving Current.

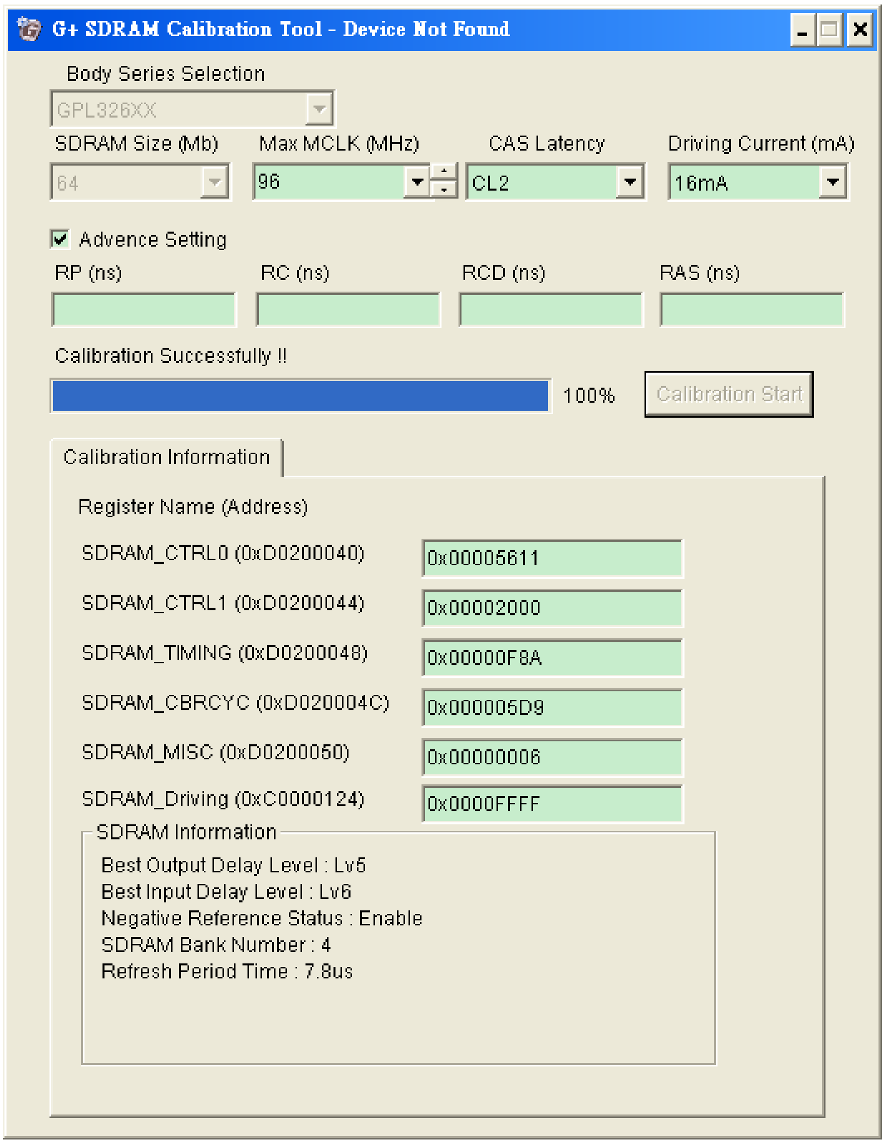
Step5: Advance Setting

Advance user may choose RP/RC/ RCD/ RAS timing (unit in ns) according to the SDRAM specification. Note that floating-point is not supported. If floating point is necessary, please fill in the closest integer instead. If no RP/ RC/ RCD/ RAS timing is defined (Advance setting not selected), tool default parameters will be applied.

Step6: Click Calibration Start

Step7: Calibration Success and a rectified SDRAM parameters is obtained.

Important Note: please REMOVE the GPL32 power (or perform hardware reset) after SDRAM Calibration is completed.



G+ SDRAM Calibration Tool - Device Not Found

Body Series Selection: GPL326XX

SDRAM Size (Mb): 64

Max MCLK (MHz): 96

CAS Latency: CL2

Driving Current (mA): 16mA

☑ Advence Setting

RP (ns)

RC (ns)

RCD (ns)

RAS (ns)

Calibration Successfully !!

100%

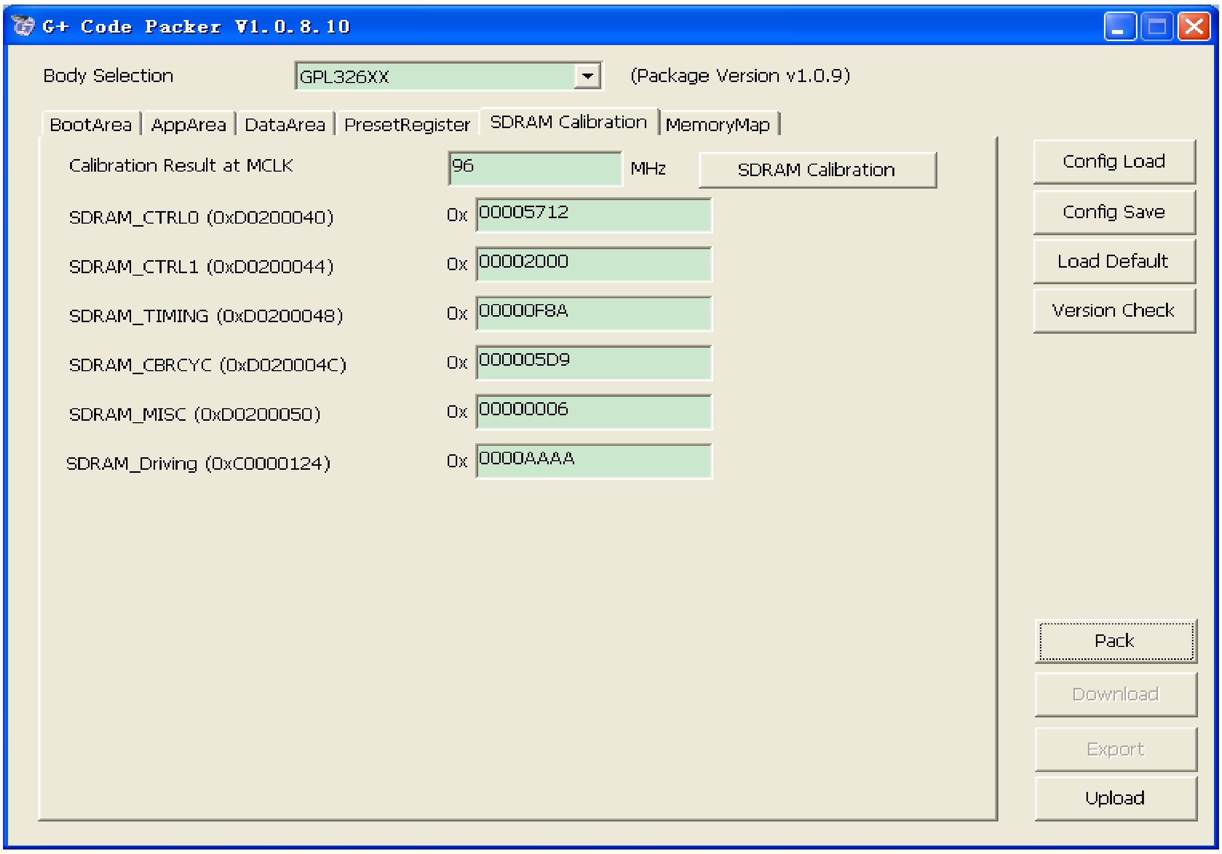
Register Name (Address)	
SDRAM_CTRL0 (0xD0200040)	0x00005611
SDRAM_CTRL1 (0xD0200044)	0x00002000
SDRAM_TIMING (0xD0200048)	0x00000F8A
SDRAM_CBRCYC (0xD020004C)	0x000005D9
SDRAM_MISC (0xD0200050)	0x00000006
SDRAM_Driving (0xC0000124)	0x0000FFFF

SDRAM Information

Best Output Delay Level : Lv5
Best Input Delay Level : Lv6
Negative Reference Status : Enable
SDRAM Bank Number : 4
Refresh Period Time : 7.8us

Step8: Input parameters in G+CodePacker (differs between GPL325XX and GPL326XX)

Turn off SDRAM calibration tool. Tool will automatically bring the SDRAM parameters after calibration into G+ CodePacker's pre-set register. It will store the SDRAM parameters into the first five registers. If parameters are already in the first five registers, the existed parameters will be moved to the location in sequence after the SDRAM parameters. To clear SDRAM parameters, push the clear calibration Reg. button. This button will clear SDRAM parameter. If SDRAM calibration is not performed, this button is void.



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

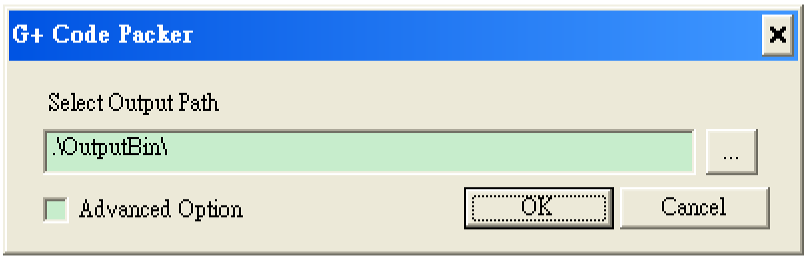
Calibration Result at MCLK: 96 MHz

SDRAM_CTRL0 (0xD0200040)	0x 00005712
SDRAM_CTRL1 (0xD0200044)	0x 00002000
SDRAM_TIMING (0xD0200048)	0x 00000F8A
SDRAM_CBRCYC (0xD020004C)	0x 000005D9
SDRAM_MISC (0xD0200050)	0x 00000006
SDRAM_Driving (0xC0000124)	0x 0000AAAA

4.6 Pack and Download

Step 1: Pack

After all areas (Boot/ App/ Data) are set up completely, click the pack button if necessary. Thus, tool will integrate all resource files to a BIN file according to the settings. It will also store the BIN file under a directory that is designated by user (the default directory is suggested).



G+ Code Packer

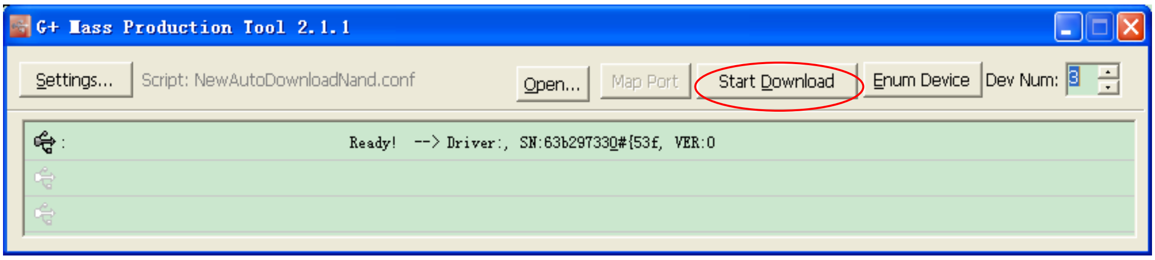
Select Output Path

.\OutputBin\

□ Advanced Option

Step 2: Download

After pack is completed, download button will be triggered. Click the download button, tool will activate USB mass production tool and set up the download parameters. At this moment, simply connect the device with a computer via a USB cable. When USB mass production tool detects the device, push the start download button to enter download procedure.

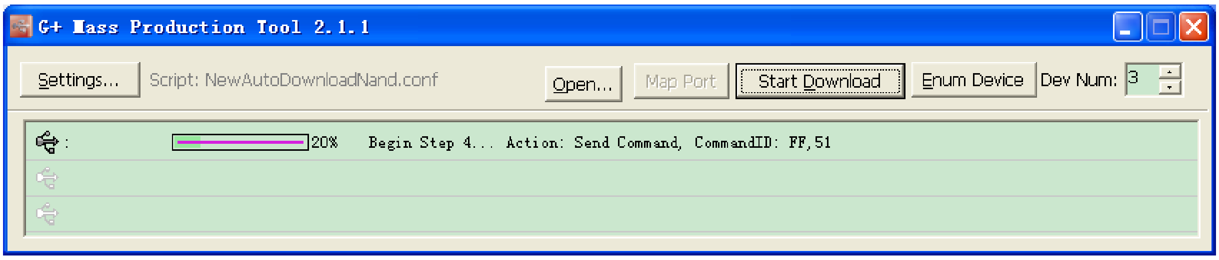


G+ Mass Production Tool 2.1.1

Script: NewAutoDownloadNand.conf

Dev Num: 3

```
Ready!  --> Driver:, SN:63b297330#{53f, VER:0
```



G+ Mass Production Tool 2.1.1

Script: NewAutoDownloadNand.conf

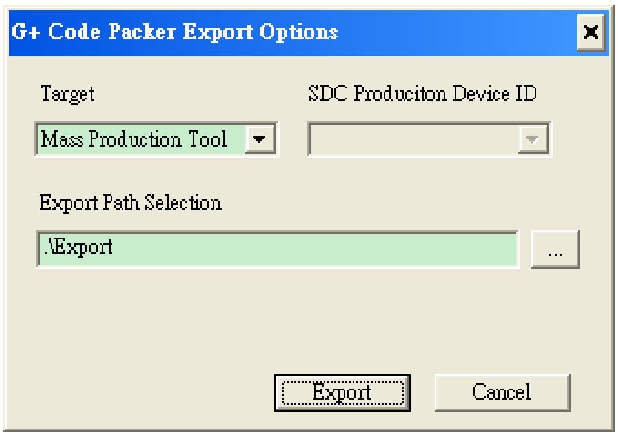
Dev Num: 3

```
20%   Begin Step 4... Action: Send Command, CommandID: FF,51
```

4.7 Export MP Tool to Other Disk (e. g. USB Disk for Factory)

Step 1: Export Window

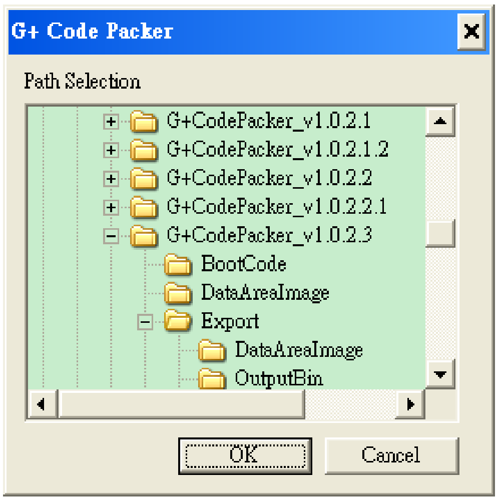
Select Mass Production Tool and set the Export's directory to G+CodePacker\Export or click the Path exchange box to change Export's directory, e.g. U:\.


G+ Code Packer Export Options

Target: Mass Production Tool

SDC Produciton Device ID:

Export Path Selection: .\Export


G+ Code Packer

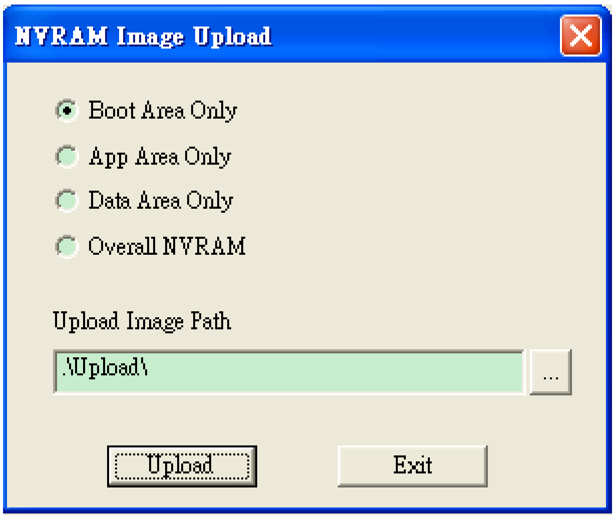
Path Selection

```
G+CodePacker_v1.0.2.1
G+CodePacker_v1.0.2.1.2
G+CodePacker_v1.0.2.2
G+CodePacker_v1.0.2.2.1
G+CodePacker_v1.0.2.3
    BootCode
    DataAreaImage
    Export
        DataAreaImage
        OutputBin
```

Step 2: After files are exported, they are portable and can be transferred easily.

4.8 Upload

G+CodePacker provides user the upload function to read up-to-date NAND Flash Image Data. This function can be used to compare the NAND Flash data with the programmed data, or copying the up-to-date NAND Flash data to another NAND Flash. The following picture is the Upload window, which can be used to read back the data from NAND's specified area (Boot, App or Data area) or entire area. We can store the read data in a specify the destination folder. When clicking "Upload" to read data, we can click the "Exit" to close window.


NVRAM Image Upload

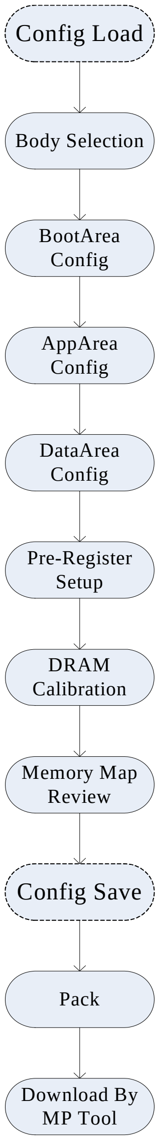
● Boot Area Only
○ App Area Only
○ Data Area Only
○ Overall NVRAM

Upload Image Path: .\Upload\

5 NAND Flash Boot Operating Flow (GPL329XX)

GPL329XX groups the NAND Flash to three areas: Boot Area, App Area, and Data Area. User is able to integrate a programmable bin file via configuring the three areas in G+CodePacker. In addition, the Tool supports Pre-set Register to invoke DRAM Calibration Tool for DRAM parameter test as well as integrating the DRAM Register's parameter into programmed file.

The following sections will introduce Boot Area, App Area, Data Area, Pre-set Register, DRAM Calibration, Pack and Download flow (see the flowchart below), and how to use it.



G+ Code Packer Basic Operation Flow

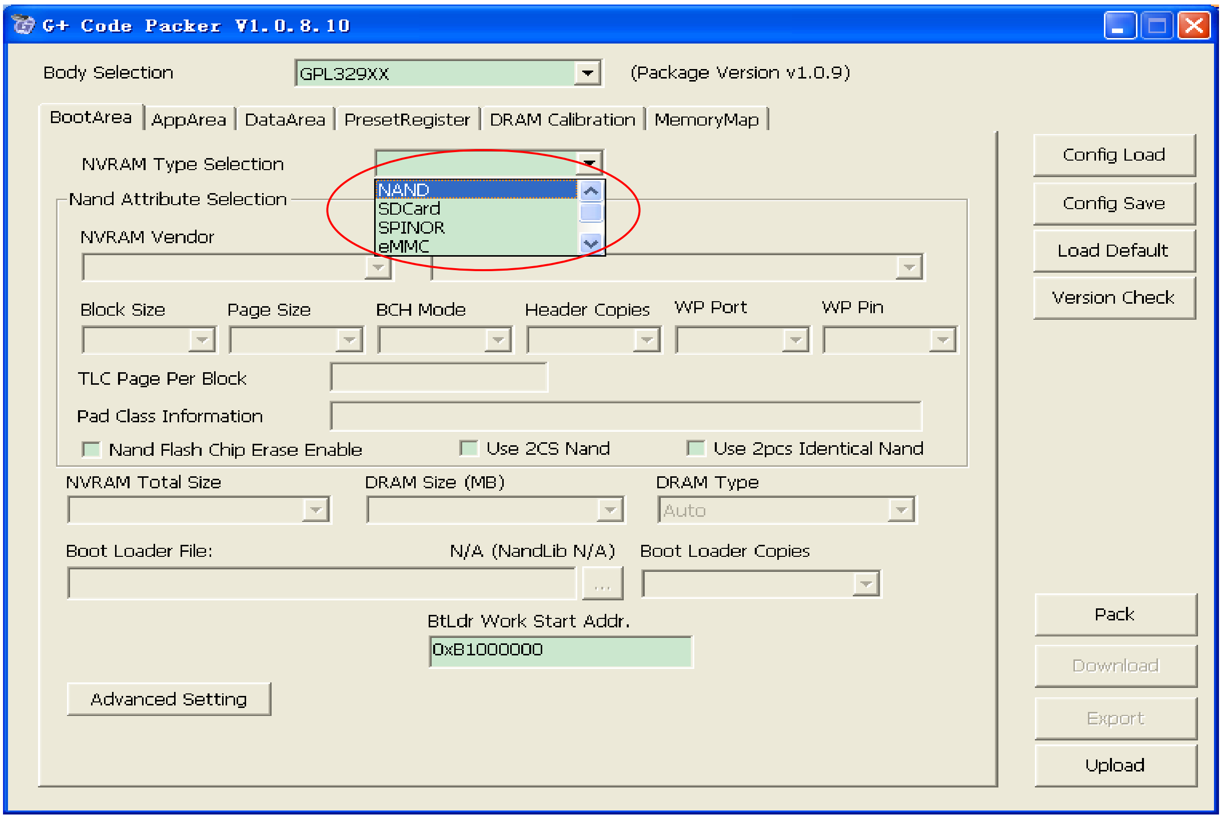
5.1 Load Config & Config Save

After run Config Save, we can re-load the previously selected parameters. We recommend store all resource files under the root path of G+CodePacker; thus, it improves the G+ CodePack compatibility and easily to be transferred to co-workers by Config Save function.

5.2 Boot Area Setup

Step1: NVRAM Type Selection

Select NVRAM (Non-Volatile RAM) type. To support SPINOR, NAND or SD Card, please select NAND.



G+ Code Packer V1.0.8.10

Body Selection: GPL329XX (Package Version v1.0.9)

BtLdr Work Start Addr.: 0xB1000000

DRAM Type: Auto

Step2: NAND Attribute Selection

In Step1, if SPINOR is selected, ignore Step2. If NAND is selected in Step1, fill in NAND Attribute in Step2., including NAND Vendor ID(manufacture ID), NAND Mark ID(product ID), NAND Type, Page Size, Boot Area's BCH mode, number of copy of Boot Area's Header.

Most of NAND Flash on market can be found its corresponding configuration options by vendor and Mark ID. If not found in the list, users need to fill out specification such as NAND Type, page size on their own according to the type of NAND Flash used. (Please select manual mode from NVRAM vendor).

In addition to Manual Mode, through the selection from NAND Mark ID, G+CodePacker will list all valid NVRAM Total Size which is not allowed to be changed. Both SDRAM Size and DRAM Type must be filled correctly by user. It is an essential parameter that will be the major factor to determine the capability of system and be the significant parameter for DRAM Calibration.

In addition, CodePacker is now supporting MLC large page NAND. If manual mode is able to be programming (burning), the auto mode can be programming too; this mode is much more suitable for manufacture's mass production.

Step3：Boot Header Copies

After filling in Mark ID, DRAM Size and DRAM Type, the number of boot header copy can be change if necessary. For non-reused NAND flash, we recommend keeping the manufacture's default settings.

Step4：Write Protect Pin Assignment (depend on HW design)

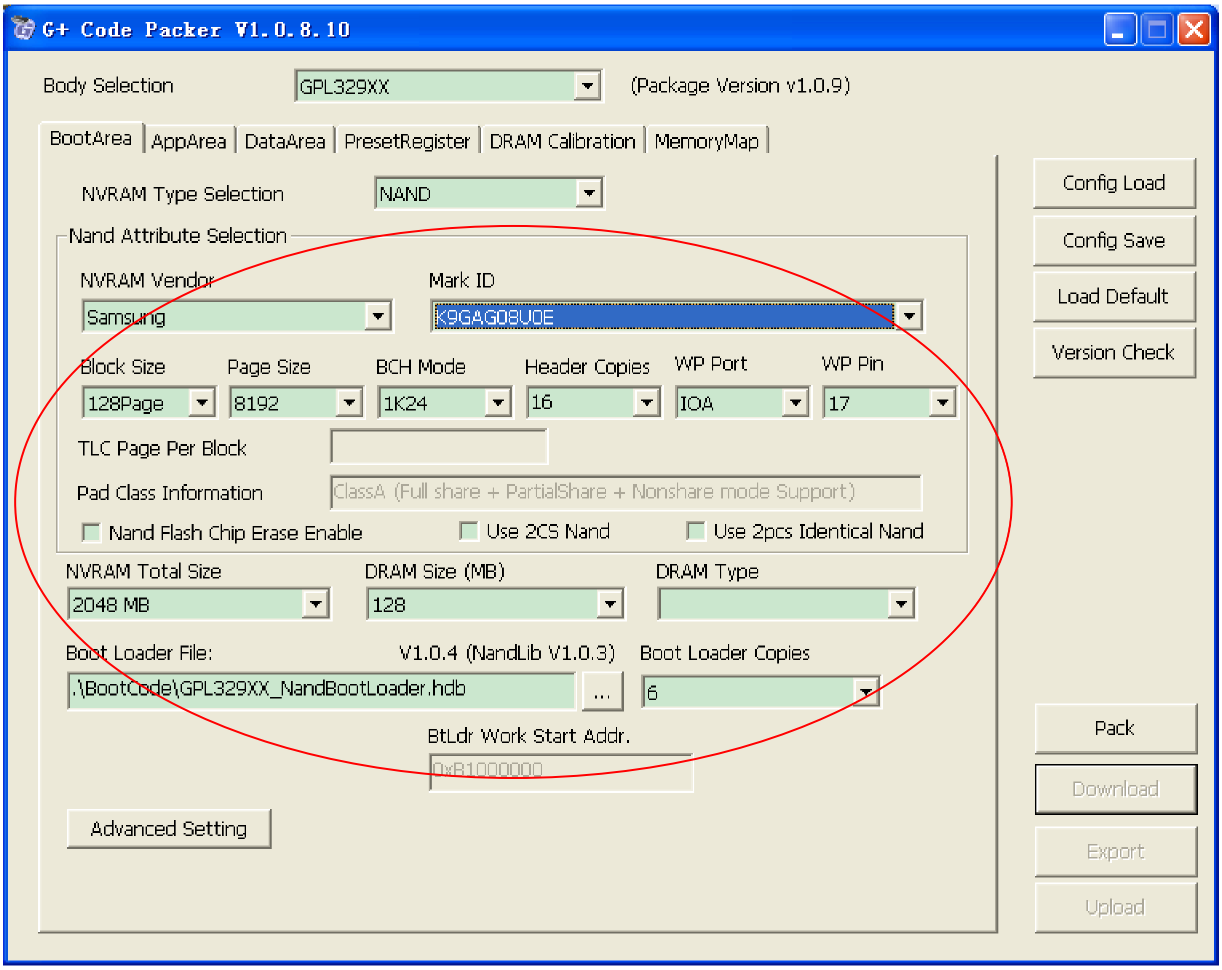
Write Protect Pin is an essential parameter. In most of development boards, the NAND Writer Protect Pin is pulled high. Thus, it can be set as NULL. And set the rest if necessary. Set the rest of settings as necessary. We recommend IOA17 as the default setting (also the default setting in G+CodePacker).

Step5: Boot Loader File Choice

For Boot Loader File, we recommend use the manufacture's default, but it can also be provided by user if necessary. The Loader file is the first program run by GPL329XX ROM Code. The program will be loaded into Generalplus management area, including Runtime Code or Read Only Resource for APP clock initialization as well as the FAT area initialization. This program is important to Generalplus standard platform and the built-in Generalplus Standard NAND Driver Library must be totally consistent with the NAND Library of Runtime code to assure the correctness and stability of NAND.

Step 6: Boot Loader Copies

Boot Loader Copies is the number of copies for Boot Loader File. In order to generate a perfect Boot Loader, we recommend make a number of copies for backup. Add or delete it if necessary (BootLoader is protected by BCH)



Body Selection: GPL329XX (Package Version v1.0.9)

NVRAM Type Selection: NAND

Nand Attribute Selection

NVRAM Vendor: Samsung

Mark ID: K9GAG08U0E

Block Size: 128Page

Page Size: 8192

BCH Mode: 1K24

Header Copies: 16

WP Port: IOA

WP Pin: 17

TLC Page Per Block:

Pad Class Information: ClassA (Full share + PartialShare + Nonshare mode Support)

☐ Nand Flash Chip Erase Enable ☐ Use 2CS Nand ☐ Use 2pcs Identical Nand

NVRAM Total Size: 2048 MB

DRAM Size (MB): 128

DRAM Type:

Boot Loader File: V1.0.4 (NandLib V1.0.3)

.\BootCode\GPL329XX_NandBootLoader.hdb

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0xB1000000

5.3 App Area Setup

Step1：App Area Setup

When NVRAM Type is NAND, App parameters must be set. If NVRAM Type is SPINOR, ignore Step1. App parameters include: App Man Size, App Spare Rate, App ROM Code Copies, and CRC Check Sum Enable.

App Man Size:

Planning App Area: note that App Main Size must be greater than the sum of Run Time ROM Code size and Resource File Size.

App Spare Rate:

App Area spare rate. To avoid the bad App Area which may cause data lose, a spare rate is designed. The App Spare Size is the ratio of App Man Size and App Spare rate, see the following formula:

$$AppSpareSize = \frac{AppMainSize}{AppSpareRate}$$, (unit: NAND Blocks)

Example:

Block Size = 64, Page Size=2048, App Main Size = 16MB (=128 blocks), App Spare Rate = 5. Thus, App Spare Size = 128 blocks / 5 = 25.6 ≒ 26 blocks = 3.25MB

The limitation of App Spare Size(unit: per block) are as follows:

Max. = Page Size / 4 and Min.= 4.

Thus, the legal values must satisfy two conditions:

4 ≦ App Spare Size ≦ Page Size / 4, unit in block.

App Spare Size ≦ App Main Size

Same example as above, to satisfy these two conditions, unit in block,

4 ≦ App Spare Size ≦ 2048 / 4

App Spare Size ≦ 128

Thus, App Spare Size ranges:

4(=0.5MB) ≦ App Spare Size ≦ 128(=16MB), meaning 1 ≦ App Spare Rate ≦ 32

The valid App Spare Rate is 1 ~ 32, corresponding App Spare Size is 4 ~ 128 blocks (=0.5 ~ 16MB).

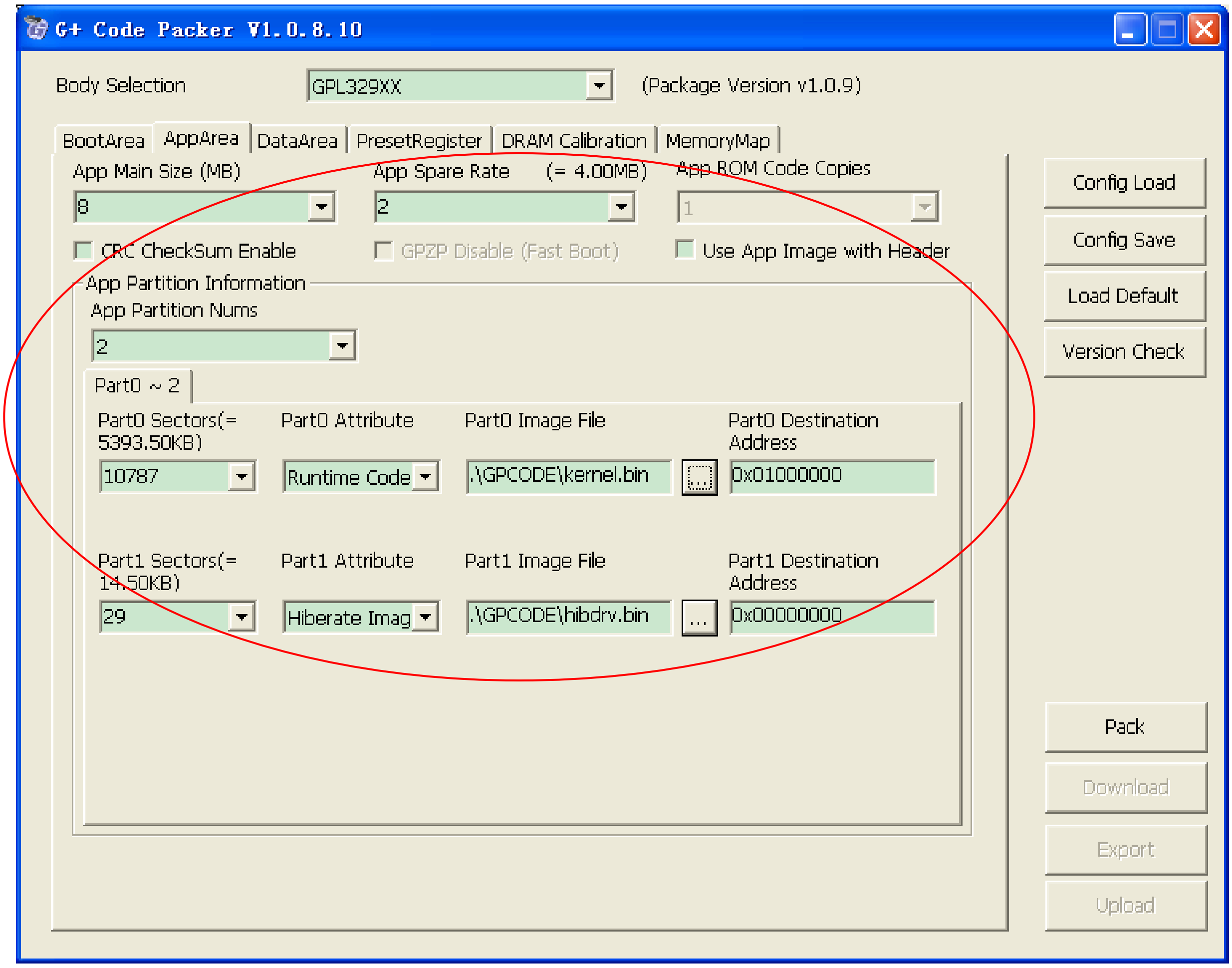
CRC Checksum Enable:

When selecting CRC Checksum Enable, CodePacker will calculate entire App Area's CRC checksum and record it in App Header for further App Area data verification use.

Step2: Setting App Partition

App Area supports up to 16 partitions. Each partition can be either Runtime code or Resource part. Run Time ROM Code or Resource Image File can be loaded and the tool will auto-compute each partition size based on the code file or image file loaded. Note that App Main Size must be greater than or equal to the sum of partition size; otherwise, pack will invalid. Boot Loader will copy the Run Time ROM Code onto DRAM. The copy size of Boot Loader is determined by the actual size of Run Time ROM Code.

Note: Run Time Rom Code Size must not exceed the actual size of DRAM; otherwise, program may execute incorrectly.



G+ Code Packer V1.0.8.10

Body Selection: GPL329XX (Package Version v1.0.9)

App Main Size (MB): 8
App Spare Rate: 2 (= 4.00MB)
App ROM Code Copies: 1

□ CRC CheckSum Enable
□ GPZP Disable (Fast Boot)
□ Use App Image with Header

App Partition Information
App Partition Nums: 2

Part0 ~ 2

Part0 Sectors(= 5393.50KB): 10787
Part0 Attribute: Runtime Code
Part0 Image File: .\GPCODE\kernel.bin
Part0 Destination Address: 0x01000000

Part1 Sectors(= 14.50KB): 29
Part1 Attribute: Hiberate Imag
Part1 Image File: .\GPCODE\hibdrv.bin
Part1 Destination Address: 0x00000000

As the diagram above, it page is mainly for App Partition setup, including number of sub-area and the corresponding parameters for each sub-area.

The parameter setup for App sub-area message are as follows:

- App Partition Num: This item represents the number of current App sub-areas that user define, ranging from 1~16, meaning max sub-area is 16 areas.

- Partition N Size: N ranges 0~15, representing the capacity of current sub-area, unit in Sector. Generally, when we specify the sub-area’s imaging file, tool will automatically fill in the corresponding file size for us. Under one circumstance, user does not specify a file and it becomes AUTO mode. If user intend to assign the sub-area’s size, simply input the value into field.
- Partition N Attribute: Similarly, N ranges 0~15, representing the type of current imaging file. In present, Tool offers 7 types of imaging file: Runtime Code, Resource Part, FastBoot Bin, Quick Image, Hibrate Image, Image Flag, and Customize BtLdr. It is important, especially when First bootloader is moving the code from App area.
- Partition N Image file: This parameter allows user to select the sub-area’s imaging file on the disk.
- Partition N Destination Address: This parameter is to specify where the current sub-area’s file will be stored in DRAM.

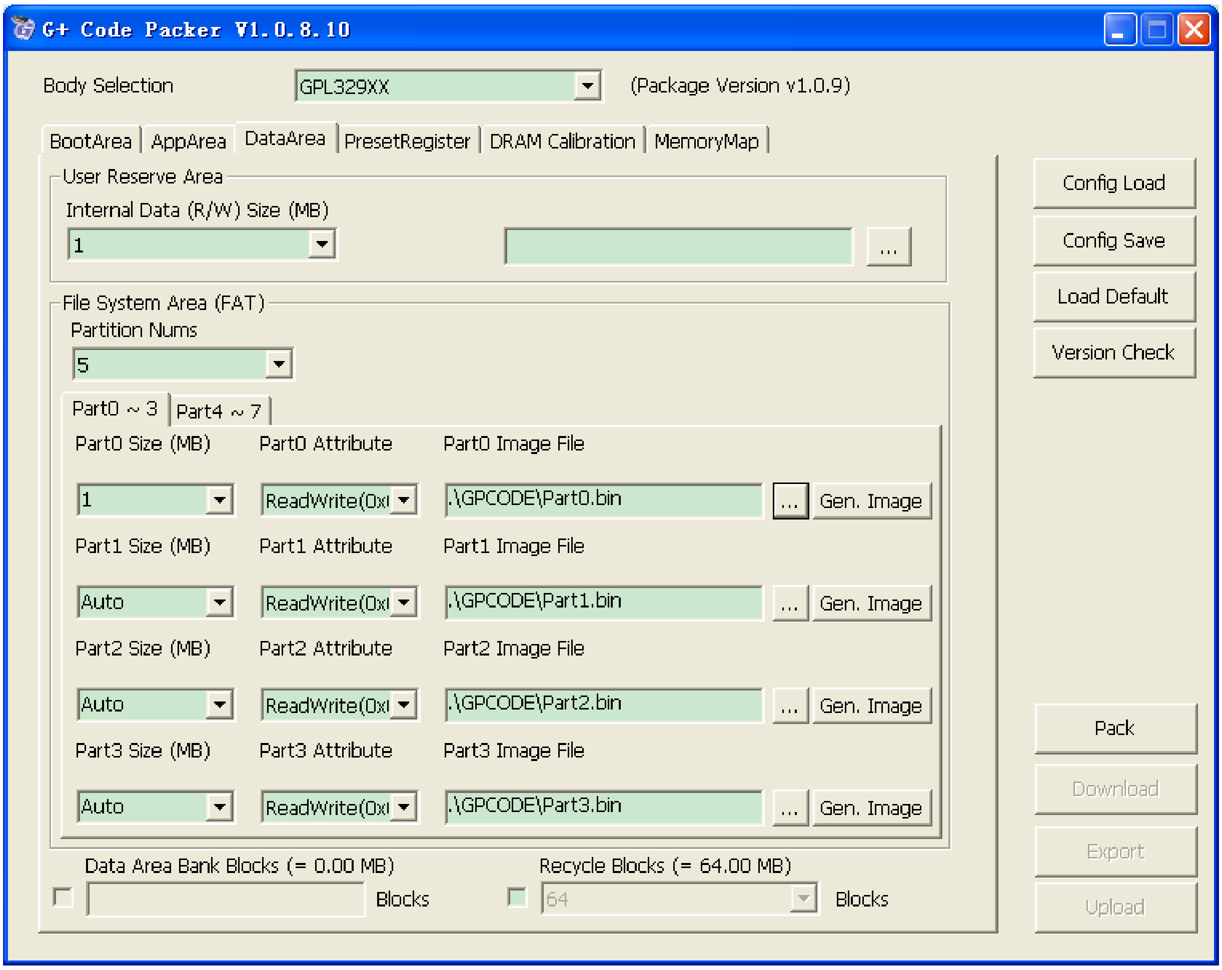
Based on this method, configure and set up all the sub-areas until all areas’ messages are configured completely.

5.4 Set up Data Area and How to Use FAT Image Tool

5.5 Data Area Setup

Step1: Set up User Reservation Area Size and Image file:

User Reserve Area stores App ROM code R/W data. The unit of User Reserve Area Size is per MBytes. R/W area can be pre-loaded data (please provide pre-load file) which can be read/written via the API for Generalplus.



Body Selection: GPL329XX (Package Version v1.0.9)

User Reserve Area
Internal Data (R/W) Size (MB): 1

File System Area (FAT)
Partition Nums: 5

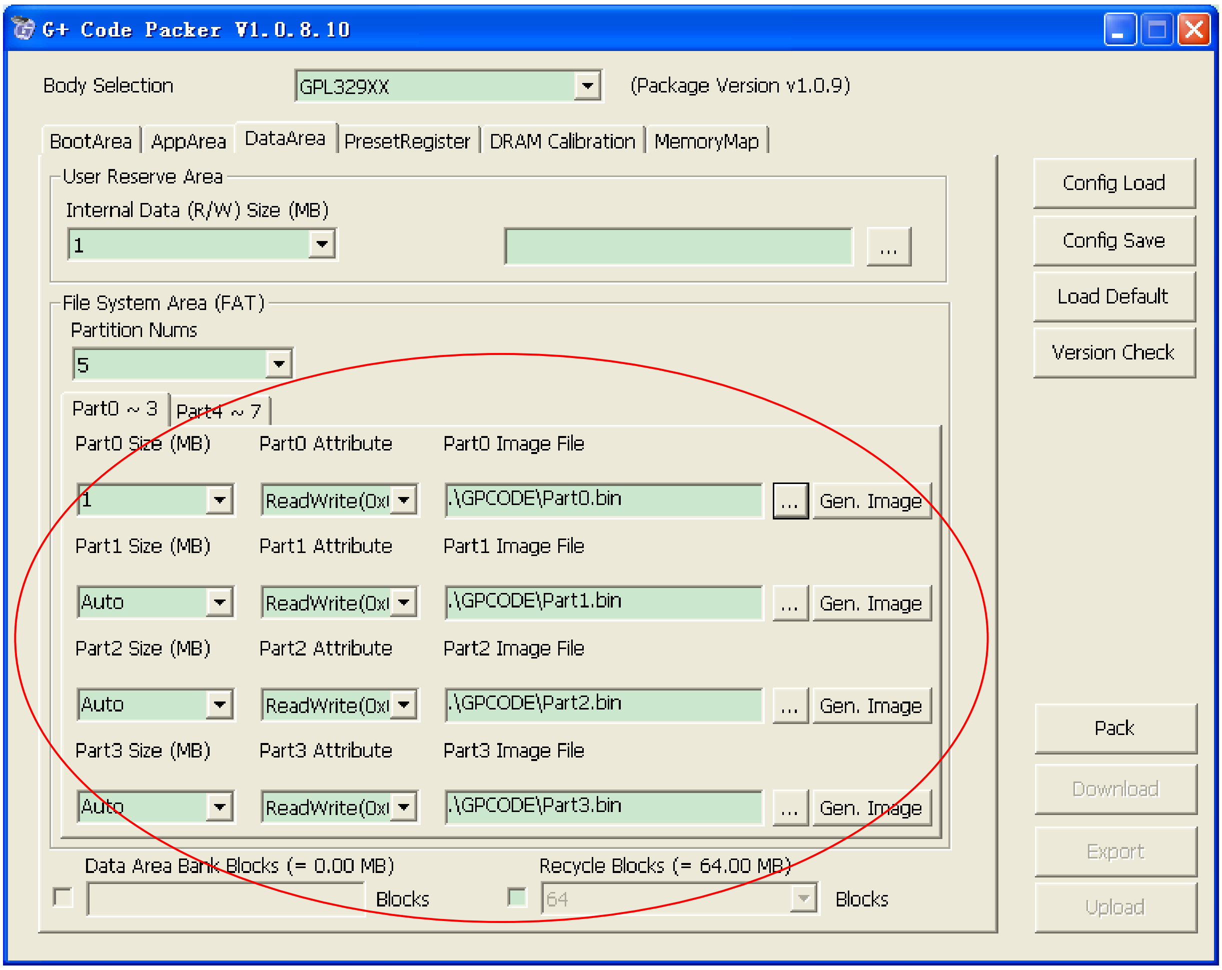
Size (MB)	Attribute	Image File
Part0 Size (MB): 1	Part0 Attribute: ReadWrite(0x	Part0 Image File: .\GPCODE\Part0.bin
Part1 Size (MB): Auto	Part1 Attribute: ReadWrite(0x	Part1 Image File: .\GPCODE\Part1.bin
Part2 Size (MB): Auto	Part2 Attribute: ReadWrite(0x	Part2 Image File: .\GPCODE\Part2.bin
Part3 Size (MB): Auto	Part3 Attribute: ReadWrite(0x	Part3 Image File: .\GPCODE\Part3.bin

☐ Data Area Bank Blocks (= 0.00 MB) Blocks

☐ Recycle Blocks (= 64.00 MB) 64 Blocks

Step2: File System Area Setup

File System Area (partition) is able to configure the size and corresponding image file for each area. All SPINOR, SDCard and NAND Flash are able to support up to 8 file system areas which can be used based on user's needs. The size of each area can be defined, unit in Mbytes. If Auto is selected, tool itself will determine the area size based on the image file chosen. Attribute has three options: Read Only, Write Only and Read/Write.



Body Selection: GPL329XX (Package Version v1.0.9)

User Reserve Area

Internal Data (R/W) Size (MB): 1

File System Area (FAT)

Partition Nums: 5

Part0 ~ 3 | Part4 ~ 7

Part0 Size (MB)	Part0 Attribute	Part0 Image File
1	ReadWrite(0x	.\GPCODE\Part0.bin
Part1 Size (MB)	Part1 Attribute	Part1 Image File
Auto	ReadWrite(0x	.\GPCODE\Part1.bin
Part2 Size (MB)	Part2 Attribute	Part2 Image File
Auto	ReadWrite(0x	.\GPCODE\Part2.bin
Part3 Size (MB)	Part3 Attribute	Part3 Image File
Auto	ReadWrite(0x	.\GPCODE\Part3.bin

☐ Data Area Bank Blocks (= 0.00 MB) Blocks

☐ Recycle Blocks (= 64.00 MB) 64 Blocks

Step3：Advance setup for Recycle Blocks (Option Choice)

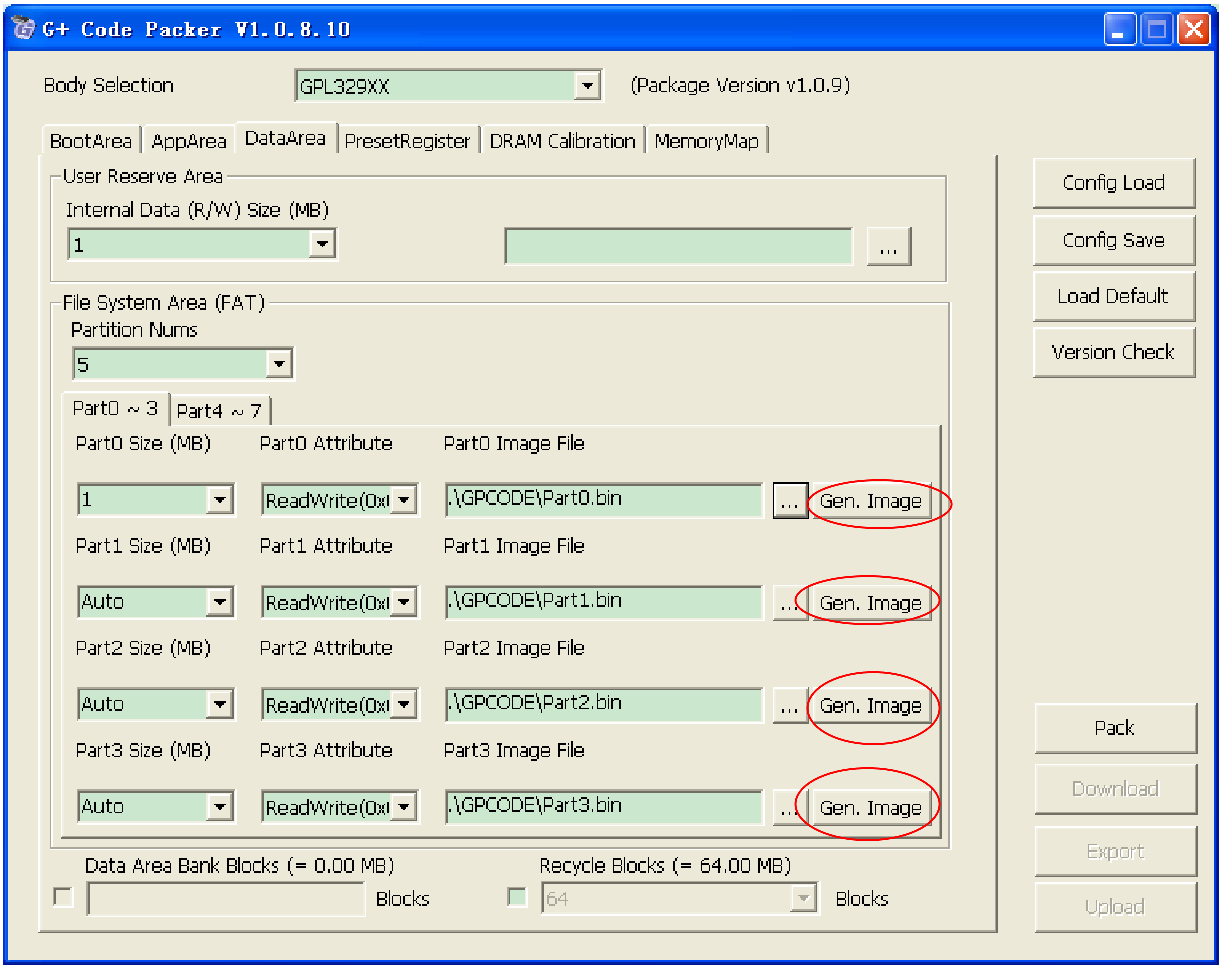
We highly recommend user use the manufacture's default setting. This is the advanced setup for Data Area manage exchange Block number **for each Bank**. The Bank size is 512 blocks, in default. In present, it is not allowed to be changed.

5.5.1 FAT Image Tool

File System Area, in addition to select the produced image file, also integrates the tools to produce image file. The flow is as follows:

Step1：Run FAT Image Tool

The image, generated by FAT Image Tool, will return the filename and partition size back to the tool. Thus, we should first select the Partition Number and click the corresponding "Gen. Image" button. Run FAT Image Tool and make the Image.



G+ Code Packer V1.0.8.10

Body Selection: GPL329XX (Package Version v1.0.9)

User Reserve Area

Internal Data (R/W) Size (MB): 1

File System Area (FAT)

Partition Nums: 5

Part0 Size (MB): 1 — Part0 Attribute: ReadWrite(0x — Part0 Image File: .\GPCODE\Part0.bin

Part1 Size (MB): Auto — Part1 Attribute: ReadWrite(0x — Part1 Image File: .\GPCODE\Part1.bin

Part2 Size (MB): Auto — Part2 Attribute: ReadWrite(0x — Part2 Image File: .\GPCODE\Part2.bin

Part3 Size (MB): Auto — Part3 Attribute: ReadWrite(0x — Part3 Image File: .\GPCODE\Part3.bin

□ Data Area Bank Blocks (= 0.00 MB) Blocks

□ Recycle Blocks (= 64.00 MB) 64 Blocks

In addition, during setup, a recycle block should also be configured. It is mainly to indicate the programming procedure where if a bad clock is found during programming, it will replace it with ones reserved here. The default value is 64, meaning it reserves 64 blocks for bad blocks if found. User is able to modify the number of recycle block in which the options are 32, 64 and 128. In general, if not ticked off, a default value (64) is assigned.

Step2：Set up FAT Image Tool and make Image file

After run FAT Image Too, the following parameters must be set: FAT Disk Size, Cluster Size, FAT Image Source Folder, and FAT Image Output Path.

1. FAT Disk Size

Set up the generated image's Partition Size. Value given here must be greater than or equal to the selected Source Folder's size, otherwise, the process of generating image will fail.

2. Cluster Size

Set up Cluster Size. We recommend using the tool's default setting (Auto). Incorrect setting may result the image generating failure or the generated file cannot be recognized.

3. FAT Image Source Folder

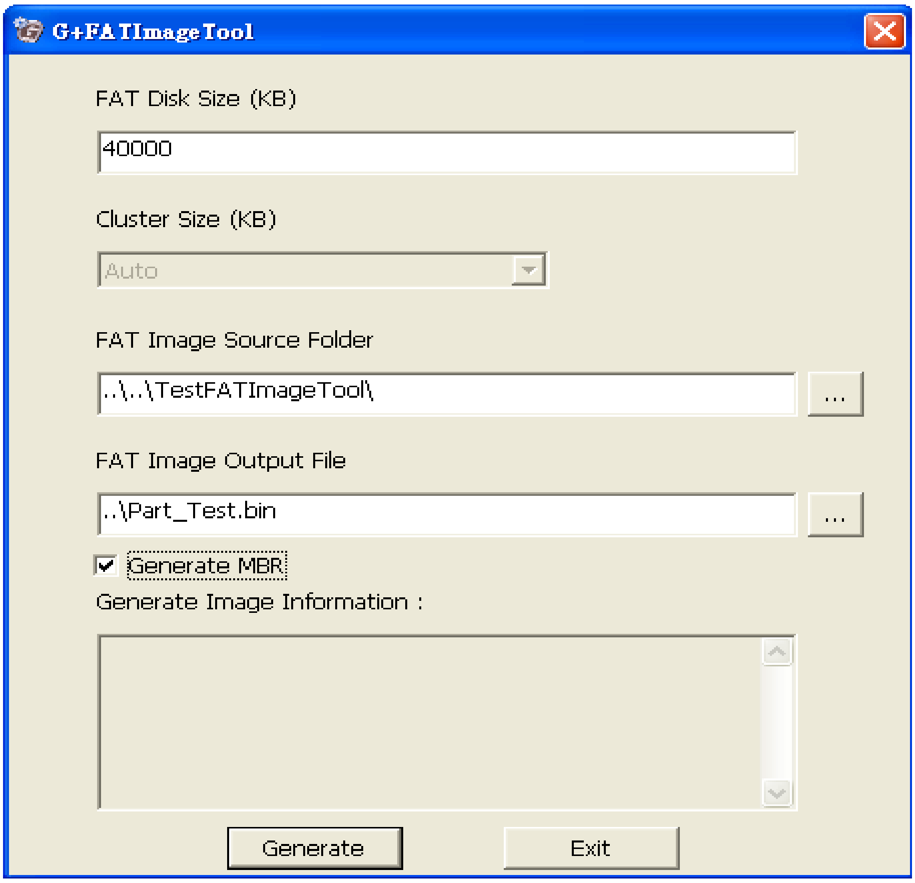
Specify the source directory for the generated image file. Tool will make the image file from the given directories, including all files under all directories, directory structures and sub-directories.

4. FAT Image Output Path

Set up the path for image file.

7. Generate MBR

Specify if the image file requires MBR information. If yes, check off the “MBR”.



FAT Disk Size (KB): 40000

Cluster Size (KB): Auto

FAT Image Source Folder: ..\..\TestFATImageTool\

FAT Image Output File: ..\Part_Test.bin

☑ Generate MBR

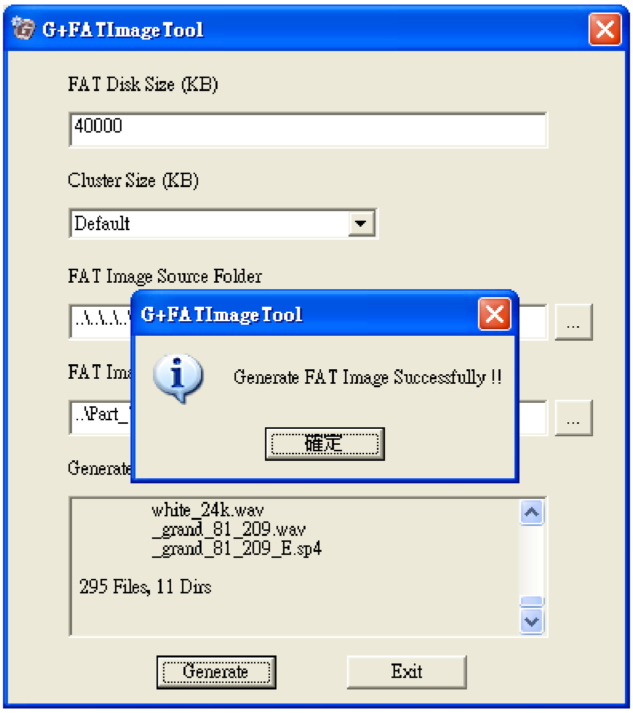
Generate Image Information :

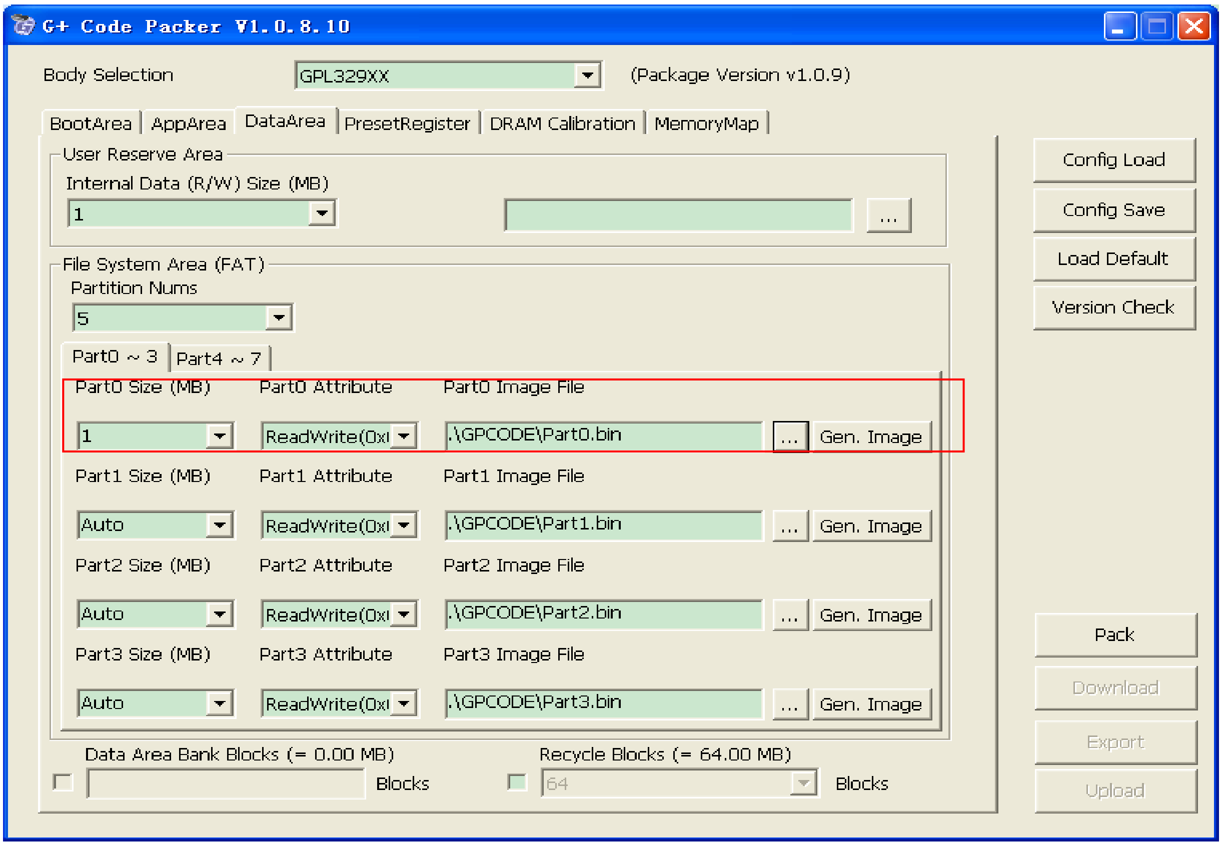
Click “Generate” button after completing configuration and starts to make image file. During image production, we can view the progress, to which folder or file through "Generate Image Information" window.

Note: It is not allowed having Simplified Chinese filename or folder name under the Traditional Chinese Operating System and vice versa; otherwise the image file production will be failed.

Step3: Bring information into Code Packer automatically

After image file is completed, the "Generate FAT Image Successfully" window pops out and click OK to close FAT Image Tool. Relevant information will be auto-loaded into Code Packer and ends the image production.





G+ Code Packer V1.0.8.10

Body Selection: GPL329XX (Package Version v1.0.9)

User Reserve Area

Internal Data (R/W) Size (MB): 1

File System Area (FAT)

Partition Nums: 5

Part0 ~ 3 | Part4 ~ 7

Part0 Size (MB): 1 | Part0 Attribute: ReadWrite(0x| | Part0 Image File: .\GPCODE\Part0.bin | ... | Gen. Image

Part1 Size (MB): Auto | Part1 Attribute: ReadWrite(0x| | Part1 Image File: .\GPCODE\Part1.bin | ... | Gen. Image

Part2 Size (MB): Auto | Part2 Attribute: ReadWrite(0x| | Part2 Image File: .\GPCODE\Part2.bin | ... | Gen. Image

Part3 Size (MB): Auto | Part3 Attribute: ReadWrite(0x| | Part3 Image File: .\GPCODE\Part3.bin | ... | Gen. Image

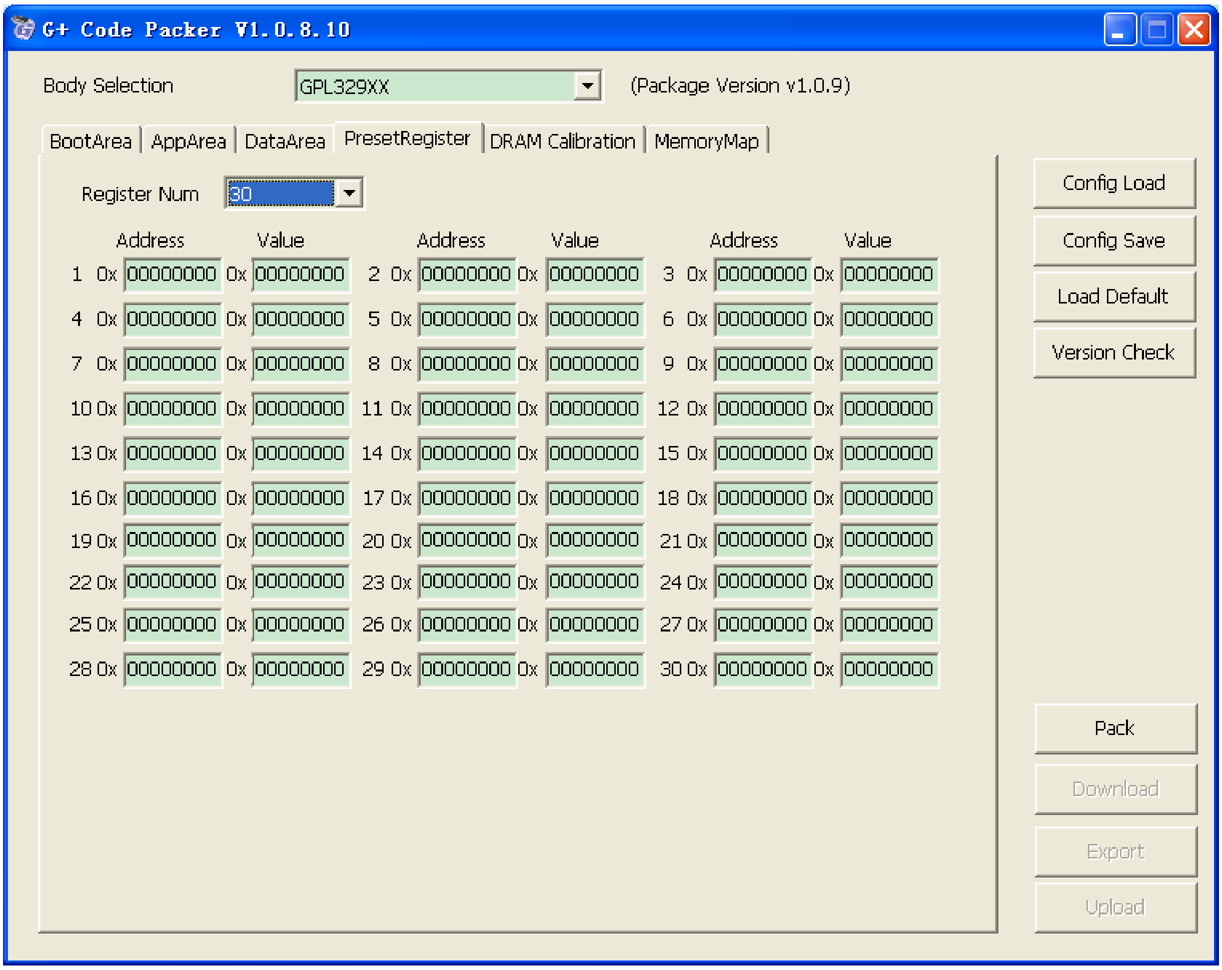
☐ Data Area Bank Blocks (= 0.00 MB) Blocks

☐ Recycle Blocks (= 64.00 MB) 64 Blocks

5.6 How to use Pre-set Register and DRAM Calibration

5.7 Set up Pre-set Register

User should set up the register according to application’s requirement. SPI Flash only supports 10 Registers settings. SDCard supports 23 Registers settings. NAND Flash supports 30 Register settings. Each setting involve Register Address and Register Value.



Body Selection GPL329XX (Package Version v1.0.9)

Register Num 30

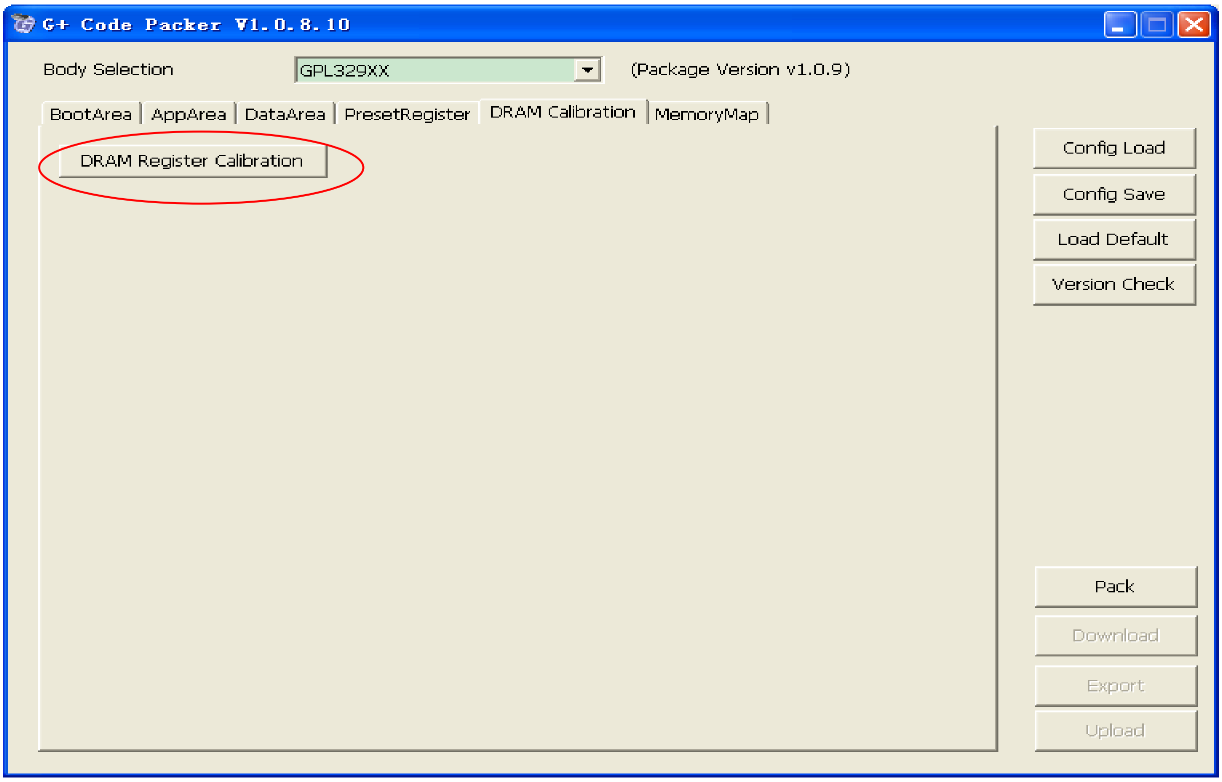
	Address	Value		Address	Value		Address	Value
1	0x 00000000	0x 00000000	2	0x 00000000	0x 00000000	3	0x 00000000	0x 00000000
4	0x 00000000	0x 00000000	5	0x 00000000	0x 00000000	6	0x 00000000	0x 00000000
7	0x 00000000	0x 00000000	8	0x 00000000	0x 00000000	9	0x 00000000	0x 00000000
10	0x 00000000	0x 00000000	11	0x 00000000	0x 00000000	12	0x 00000000	0x 00000000
13	0x 00000000	0x 00000000	14	0x 00000000	0x 00000000	15	0x 00000000	0x 00000000
16	0x 00000000	0x 00000000	17	0x 00000000	0x 00000000	18	0x 00000000	0x 00000000
19	0x 00000000	0x 00000000	20	0x 00000000	0x 00000000	21	0x 00000000	0x 00000000
22	0x 00000000	0x 00000000	23	0x 00000000	0x 00000000	24	0x 00000000	0x 00000000
25	0x 00000000	0x 00000000	26	0x 00000000	0x 00000000	27	0x 00000000	0x 00000000
28	0x 00000000	0x 00000000	29	0x 00000000	0x 00000000	30	0x 00000000	0x 00000000

5.7.1 DRAM Calibration

5.7.2 Step1：Click DRAM Register Calibration button

Click DRAM Register Calibration button to awake DRAM Calibration Tool. Hardware access mask Rom Code to connect with computer. If the tool detects the hardware, it will display ”Device Connected”, otherwise, it shows “Device Not Found”. All Calibration must be proceeded under ”Device Connected” status.

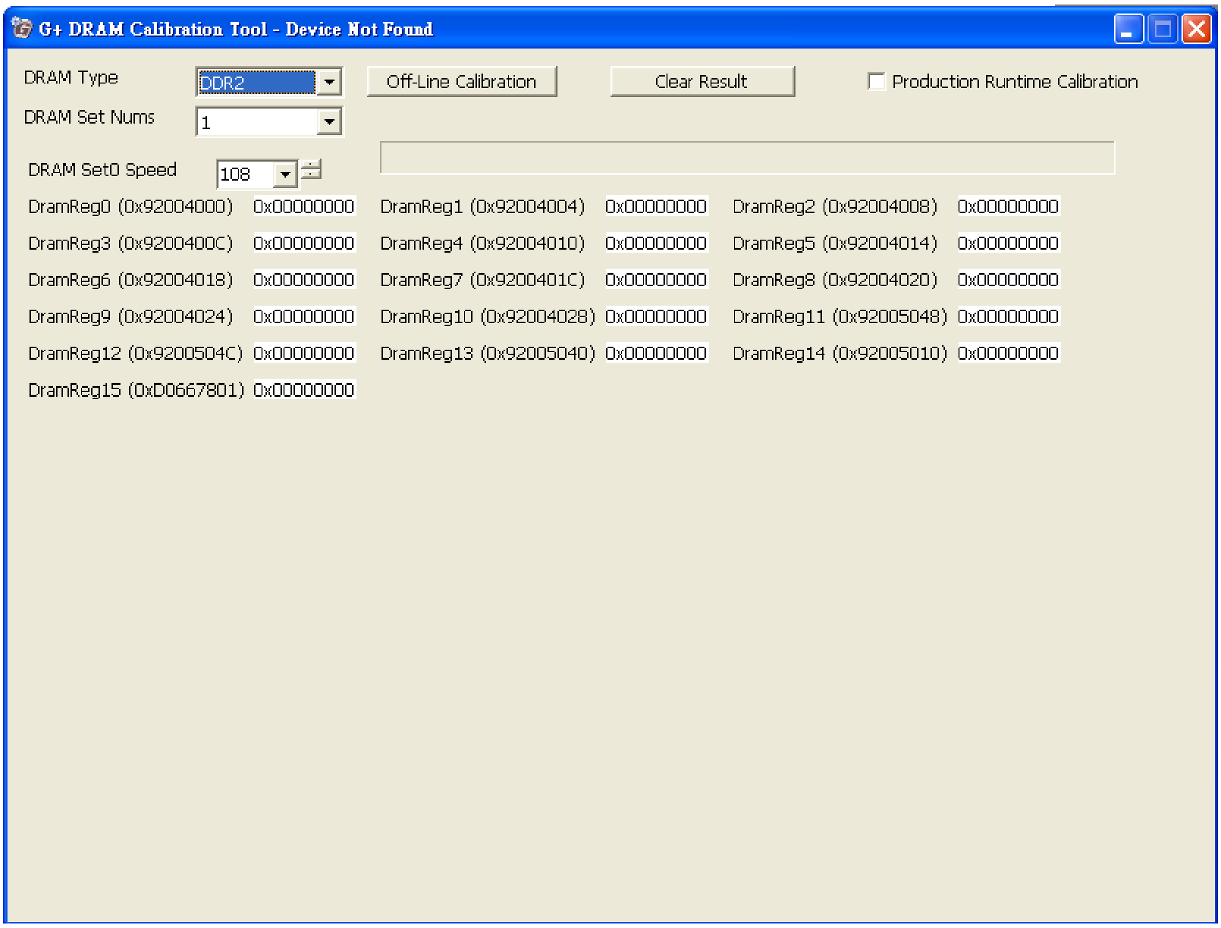
Enter USB service mode via Internal ROM Code Boot.



G+ Code Packer V1.0.8.10
Body Selection
GPL329XX
(Package Version v1.0.9)
BootArea
AppArea
DataArea
PresetRegister
DRAM Calibration
MemoryMap
DRAM Register Calibration
Config Load
Config Save
Load Default
Version Check
Pack
Download
Export
Upload

Step 2. Calibration Tool (sub tool)

Select DRAM Type and set up the DRAM Set Nums and DRAM speed.



G+ DRAM Calibration Tool - Device Not Found

DRAM Type: DDR2

DRAM Set Nums: 1

DRAM Set0 Speed: 108

□ Production Runtime Calibration

DramReg0 (0x92004000)	0x00000000	DramReg1 (0x92004004)	0x00000000	DramReg2 (0x92004008)	0x00000000
DramReg3 (0x9200400C)	0x00000000	DramReg4 (0x92004010)	0x00000000	DramReg5 (0x92004014)	0x00000000
DramReg6 (0x92004018)	0x00000000	DramReg7 (0x9200401C)	0x00000000	DramReg8 (0x92004020)	0x00000000
DramReg9 (0x92004024)	0x00000000	DramReg10 (0x92004028)	0x00000000	DramReg11 (0x92005048)	0x00000000
DramReg12 (0x9200504C)	0x00000000	DramReg13 (0x92005040)	0x00000000	DramReg14 (0x92005010)	0x00000000
DramReg15 (0xD0667801)	0x00000000				

Step3: click Off-Line Calibration.

Step4: Calibration Success and generating a calibrated DRAM

Note: To complete DRAM calibration, please remove (unplug) the entire power from GPL329XX(or perform Hardware reset).

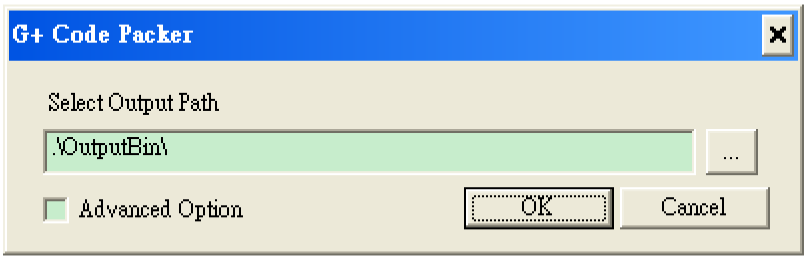
Step5: Load DRAM parameter into Pre-Register

Close DRAM Calibration Tool, it will load the obtained parameters into Pre-set Register in G+ Code Packet. This will load the DRAM parameters in the first 10 registers. If the first 10 registers have already had other settings, these settings will be moved to the registers after the DRAM parameters. Use the Clear Result button to clear DRAM parameter and restore the original settings. If DRAM Calibration is not performed, pressing this button will have no effect.

5.8 Pack and Download

Step 1：Pack

After all areas are set up completely, click the Pack button for file integration. Tool will integrate all file files into a BIN and store it under the directory user choose (default path recommended)

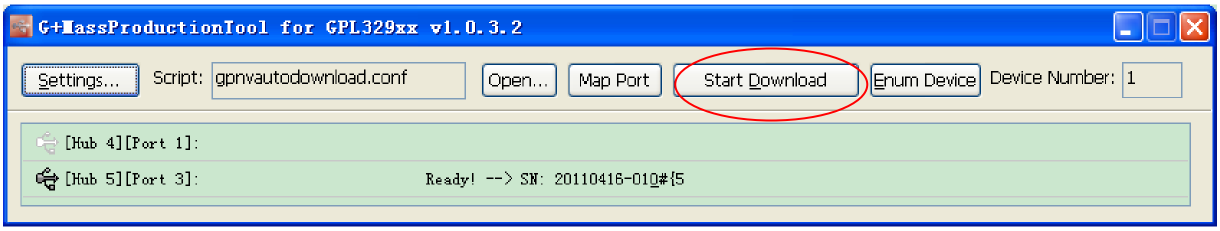


Select Output Path: .\OutputBin\

□ Advanced Option

Step 2：Download

After pack is completed, Download button is triggered. Click Download button to awake USB Mass Production Tool and set up Download parameters. After connecting device with computer via USB, USB Mass Production Tool will detect the device and press the Start Download button to enter Download procedure.

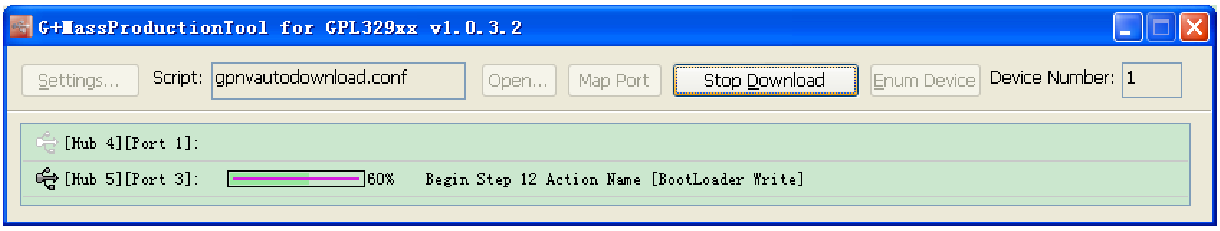


Script: gpnvautodownload.conf

Device Number: 1

[Hub 4][Port 1]:

[Hub 5][Port 3]: Ready! --> SN: 20110416-010#{5



Script: gpnvautodownload.conf

Device Number: 1

[Hub 4][Port 1]:

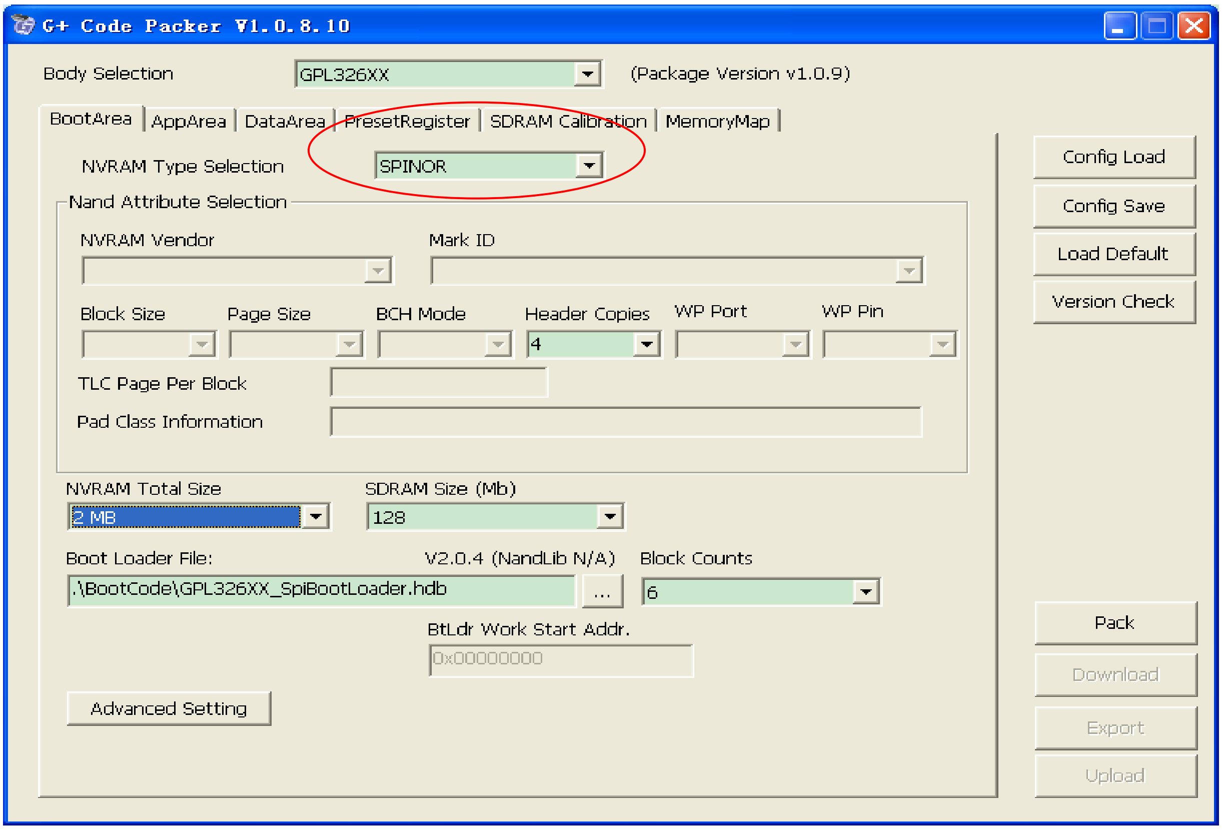
[Hub 5][Port 3]: 60% Begin Step 12 Action Name [BootLoader Write]

6 SPI Flash Boot Operation

G+ Code Packer divides the SPI Flash into three operation areas: Boot Area, App Area, and Data Area. The resource files of these three areas can be integrated and moreover, tool also supports the pre-set register and use SDRAM Calibration Tool for SDRAM parameter test as well as programming the parameters into files. The following sections will introduce the Boot Area, App Area, Data Area, Pre-set Register setup, SDRAM Calibration, Pack and download flows.

6.1 Boot Area setup

Step1: NVRAM Type Selection -- SPINOR


G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

NVRAM Type Selection: SPINOR

Header Copies: 4

NVRAM Total Size: 2 MB

SDRAM Size (Mb): 128

Boot Loader File: .\BootCode\GPL326XX_SpiBootLoader.hdb

V2.0.4 (NandLib N/A)

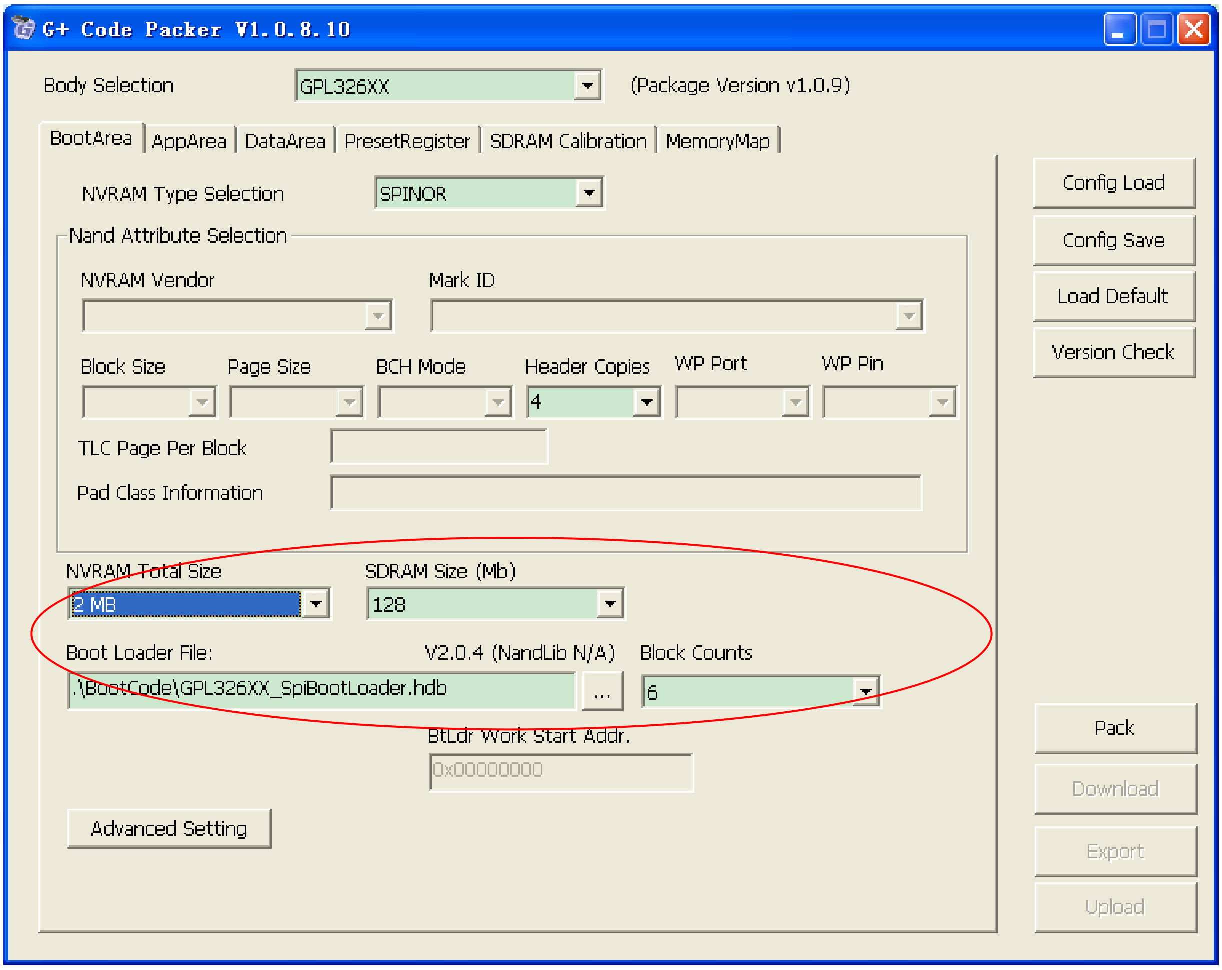
Block Counts: 6

BtLdr Work Start Addr.: 0x00000000

Step2: NVRAM Size, SDRAM Size, and Boot Loader File

User must exercise cares in the unit of NVRAM Size and SDRAM Size. The flash size is per Mbytes and the engineering SDRAM size is per Mbit.

SPI Flash and NAND Flash will use different Boot Loader Files. If NVRAM Type is the SPI Flash, it is not necessary to define the number of block (Block counts) to be copied for Boot Loader. In general, tool will input the default Boot Loader File according to the selected NVRAM Type. We suggest using the default Boot Loader File from the tool.



Body Selection: GPL326XX (Package Version v1.0.9)

NVRAM Type Selection: SPINOR

NVRAM Vendor:

Mark ID:

Block Size:

Page Size:

BCH Mode:

Header Copies: 4

WP Port:

WP Pin:

TLC Page Per Block:

Pad Class Information:

NVRAM Total Size: 2 MB

SDRAM Size (Mb): 128

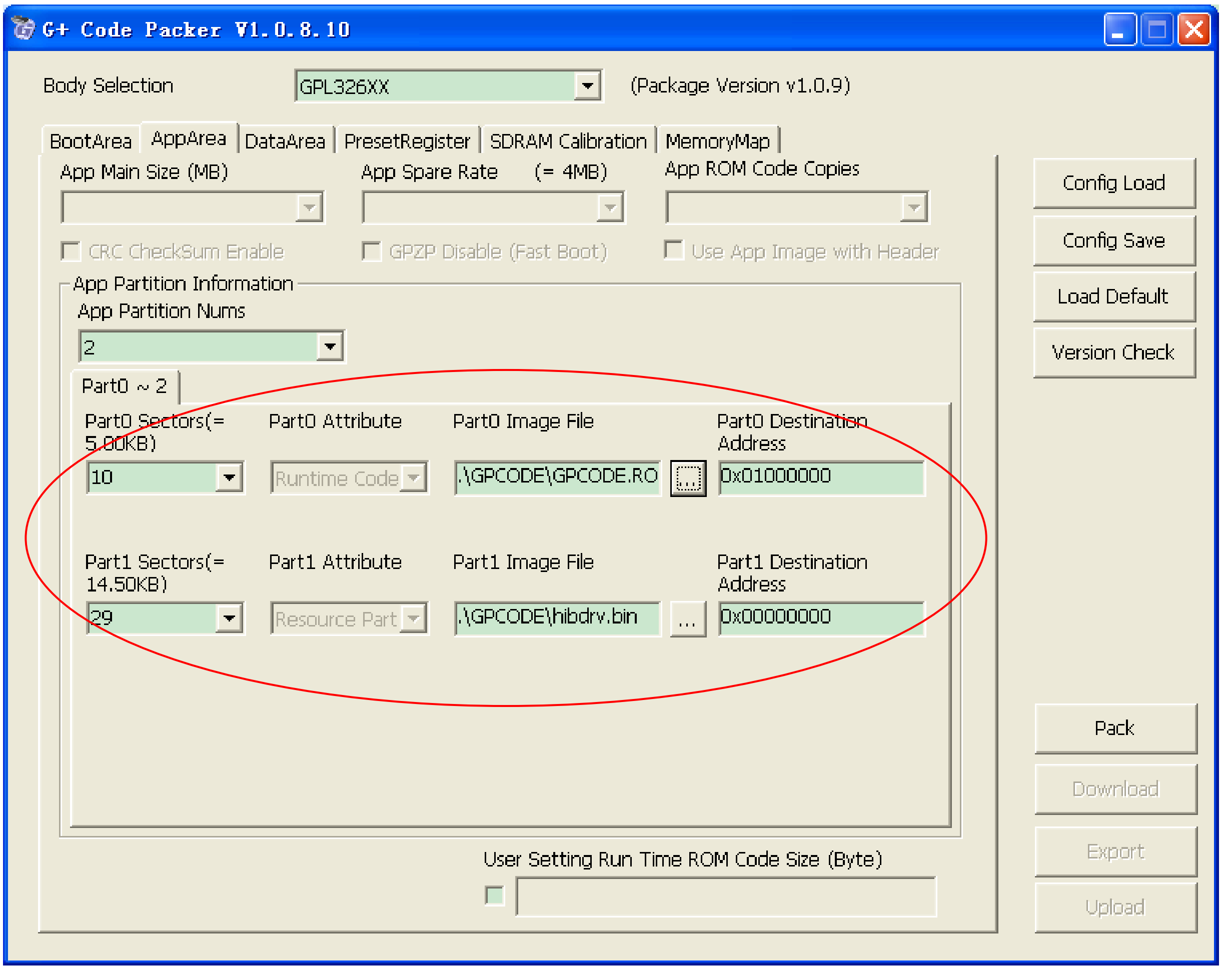
Boot Loader File: .\BootCode\GPL326XX_SpiBootLoader.hdb

V2.0.4 (NandLib N/A)

Block Counts: 6

BtLdr Work Start Addr.: 0x00000000

6.2 App Area Setup



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

App Main Size (MB)
App Spare Rate (= 4MB)
App ROM Code Copies
□ CRC CheckSum Enable
□ GPZP Disable (Fast Boot)
□ Use App Image with Header

App Partition Information
App Partition Nums: 2

Part0 ~ 2

Part0 Sectors(= 5.00KB)	Part0 Attribute	Part0 Image File	Part0 Destination Address
10	Runtime Code	.\GPCODE\GPCODE.RO	0x01000000

Part1 Sectors(= 14.50KB)	Part1 Attribute	Part1 Image File	Part1 Destination Address
29	Resource Part	.\GPCODE\hibdrv.bin	0x00000000

User Setting Run Time ROM Code Size (Byte)
□

As the diagram above, this window is mainly for App Partition setup, including number of sub-area and the corresponding parameters for each sub-area.

The parameter setup for App sub-area message are as follows:

- App Partition Num: This item represents the number of current App sub-areas that user defines, ranging from 1~2, meaning max sub-area is 2 areas.
- Partition N Size: N ranges 0~1, representing the capacity of current sub-area, unit in Sector. Generally, when we specify the sub-area's imaging file, tool will automatically fill in the corresponding file size for us. Under one circumstance, user does not specify a file and it becomes AUTO mode. If user intend to assign the sub-area's size, simply input the value into field.
- Partition N Attribute: Similarly, N ranges 0~1, representing the type of current imaging file. In present, Tool offers two types of imaging file: Runtime Code, Resource Part. It is important, especially when First bootloader is moving the code from App area.

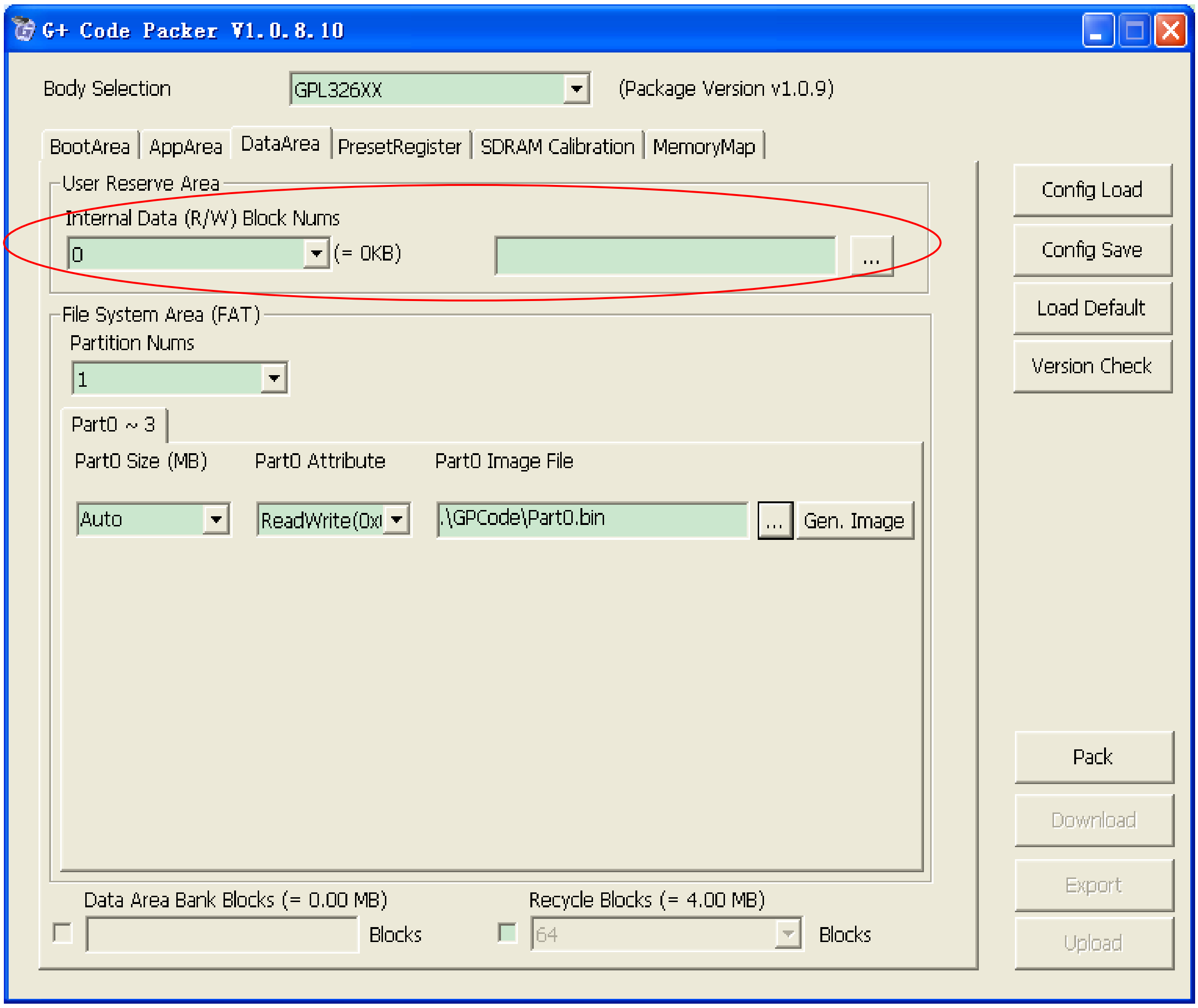
- ➢ Partition N Image file: This parameter allows user to select the sub-area's imaging file on the disk.
- ➢ Partition N Destination Address: This parameter is to specify where the current sub-area's file will be stored in SDRAM.

Based on this method, configure and set up all the sub-areas until all areas' messages are configured completely.

6.3 Data Area setup

Step1: User Reserve Area Size and Image setup

User Reservation Area stores the App ROM code R/W data. The unit of User Reservation Area Size is per Block. After choice is made, the reserve area size will be shown after the choice made (unit: Kbytes).



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

User Reserve Area

Internal Data (R/W) Block Nums: 0 (= 0KB)

File System Area (FAT)

Partition Nums: 1

Part0 ~ 3

Part0 Size (MB): Auto

Part0 Attribute: ReadWrite(0x

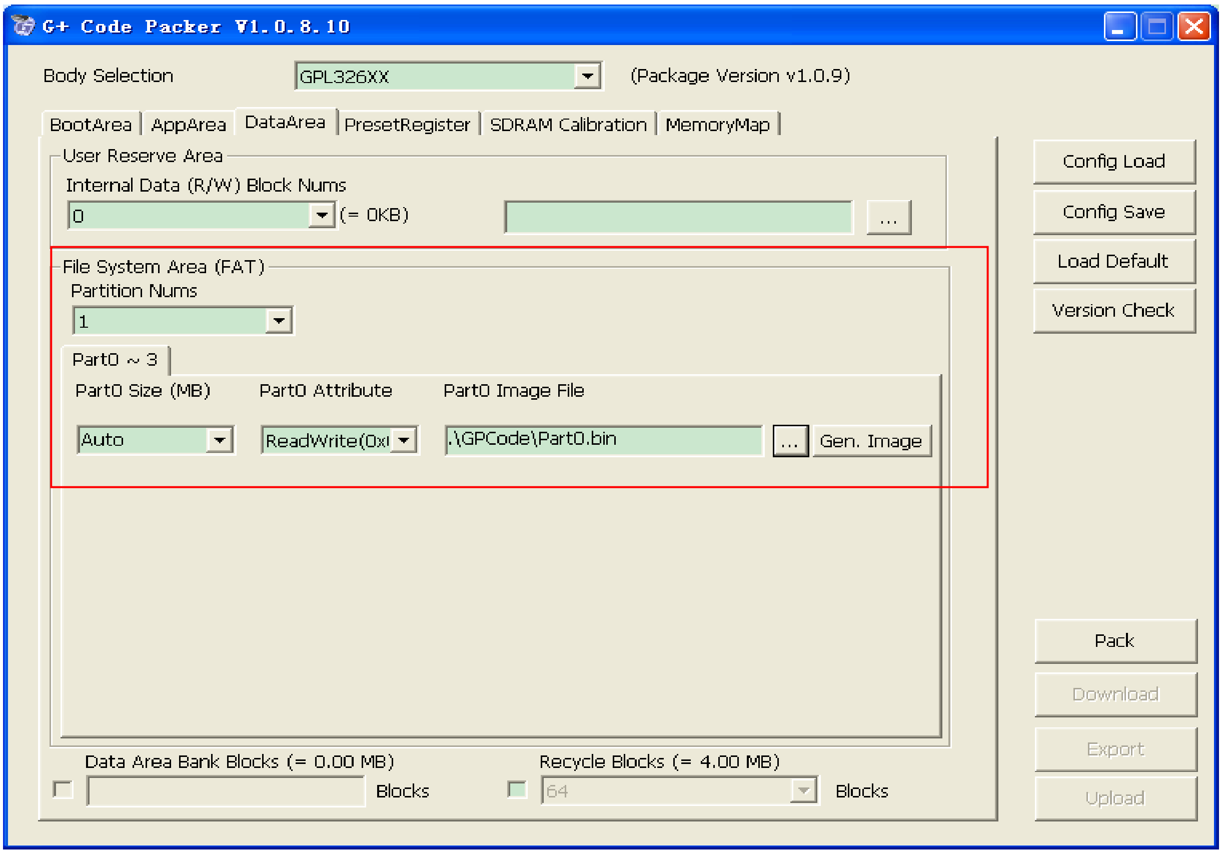
Part0 Image File: .\GPCode\Part0.bin

□ Data Area Bank Blocks (= 0.00 MB) Blocks

□ Recycle Blocks (= 4.00 MB) 64 Blocks

Step2: File System Area setup

File System Area is to define each area's sizes and corresponding image files. SPI supports one set of File System (FAT12 format). Each area's size can be defined based on needs, unit per Mbytes. If Auto is selected, tool will determine the area's size based on the image size. Attribute includes three options: Read Only、Write Only, and Read/Write.



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

User Reserve Area
Internal Data (R/W) Block Nums: 0 (= 0KB)

File System Area (FAT)
Partition Nums: 1

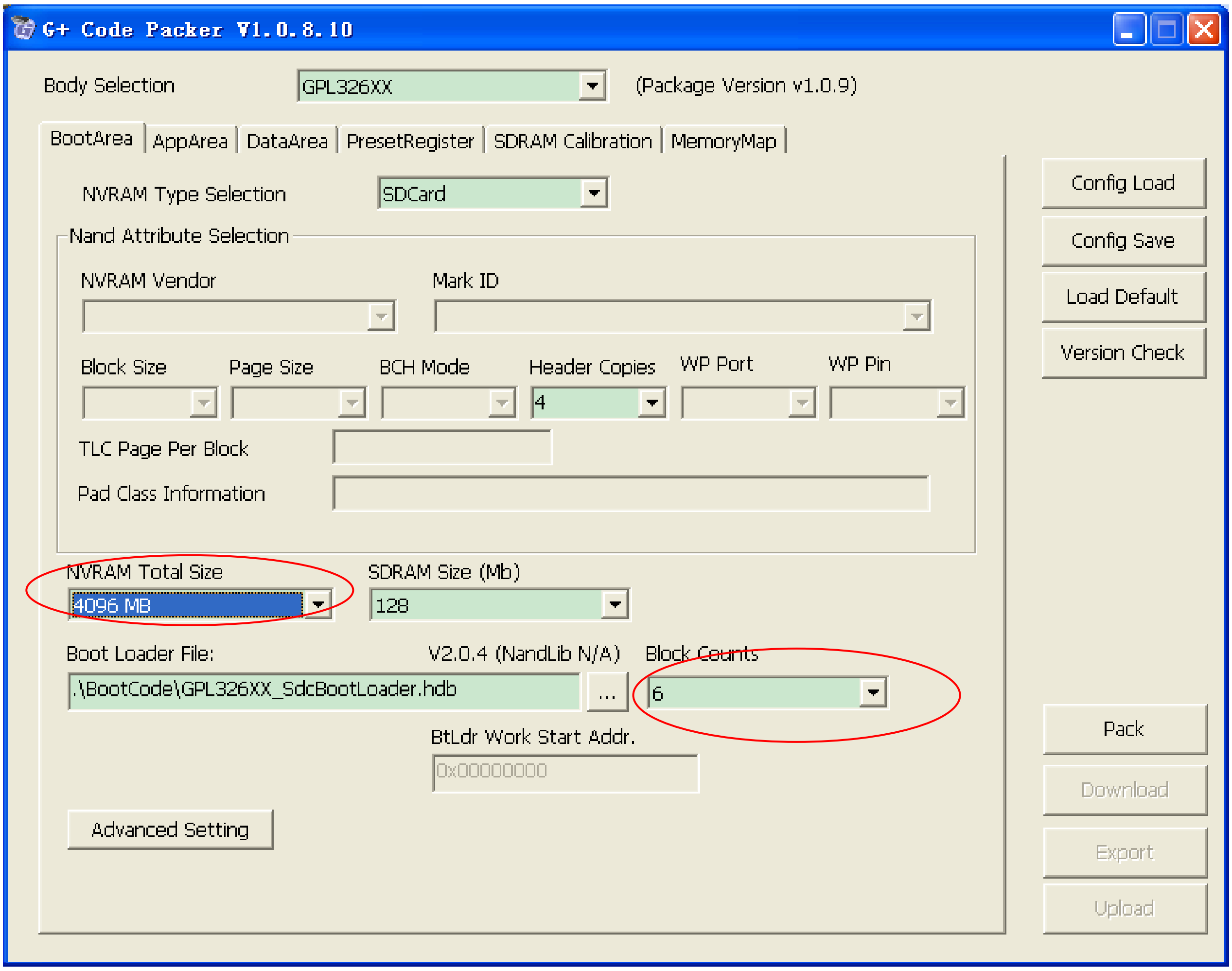
Part0 ~ 3
Part0 Size (MB): Auto
Part0 Attribute: ReadWrite(0x(
Part0 Image File: .\GPCode\Part0.bin

Data Area Bank Blocks (= 0.00 MB) □ Blocks
Recycle Blocks (= 4.00 MB) □ 64 Blocks

7 SDCard Boot Operation

G+ Code Packer divides the SDCard into three operation areas: Boot Area, App Area, and Data Area. The resource files of these three areas can be integrated together. SDC Boot is a development mode. For usage, it is the same as SPI NOR Flash (please refer to Chapter 6 for more information).

If SDC Boot compatibility issue rises, we suggest entering the Advance Mode (check the Advance Setting) to set the SDC Clock division that will lower the SDCard operating speed.



G+ Code Packer V1.0.8.10

Body Selection: GPL326XX (Package Version v1.0.9)

NVRAM Type Selection: SDCard

Header Copies: 4

NVRAM Total Size: 4096 MB

SDRAM Size (Mb): 128

Boot Loader File: .\BootCode\GPL326XX_SdcBootLoader.hdb

V2.0.4 (NandLib N/A)

Block Counts: 6

BtLdr Work Start Addr.: 0x00000000

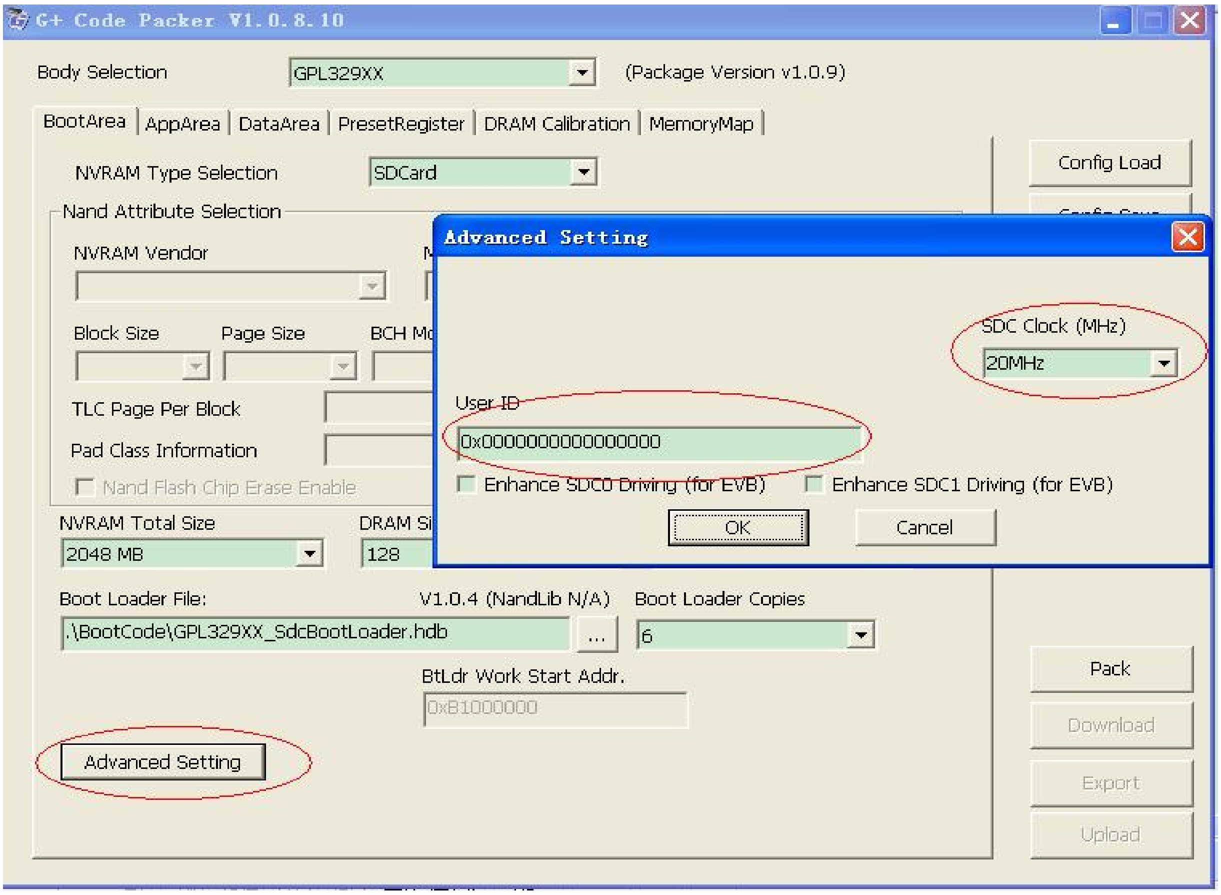
8 SDCard Boot Flow (GPL329XX)

GPL329XX will divide the SDCard into three areas: Boot Area, App Area, and Data Area. The resources in these three areas can be integrated via G+CodePacker.

If SDC Boot compatibility issue is encountered, we recommend enable Advance Mode (check off Advance Setting) and lower the SDC Clock for SDCard. If the Target Board (e.g. Generalplus EVB) has heavy loading, check SDC driving to enhance the loading capability.

Note:

3. Before using SDCard, we highly recommend user back up all sources files and data onto a backup disk.
4. After using SDCard, if the memory size becomes less, we can format it using the format tool, e.g. freeware, SD Formatter.. Please visit the following site: http://www.sdcard.org/consumers/formatter_3.



G+ Code Packer V1.0.8.10

Body Selection: GPL329XX (Package Version v1.0.9)

NVRAM Type Selection: SDCard

NVRAM Total Size: 2048 MB

DRAM Si: 128

Boot Loader File: .\BootCode\GPL329XX_SdcBootLoader.hdb

V1.0.4 (NandLib N/A)

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0xB1000000

Advanced Setting

SDC Clock (MHz): 20MHz

User ID: 0x0000000000000000

□ Enhance SDC0 Driving (for EVB)

□ Enhance SDC1 Driving (for EVB)

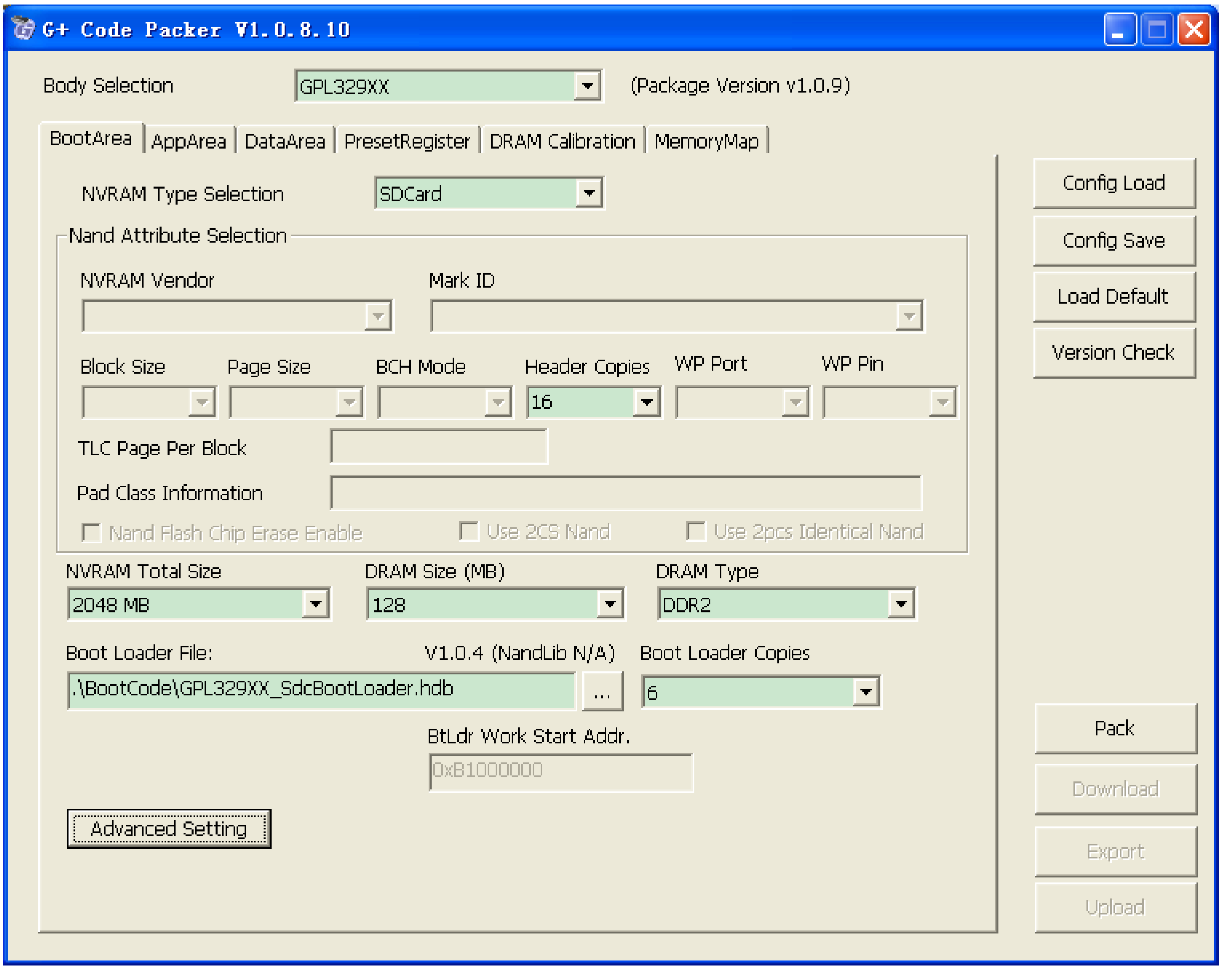
The following sections describe the setups for Boot Area, App Area, and Data Area

8.1 Load Config & Config Save

If we've done the Config Save, we are allowed to re-load the previous parameters via Config Load. We highly recommend storing all resource files under the G+CodePacker's root path. Thus, it increases the portability to easily transport files to associate partners via Config Save.

8.2 Set up Boot Area

Select SD Card and assign NVRAM Total Size (i.e. SDCard Size), BootHeader Copies, BootLoader Copies, DRAM Size, DRAM Type, Boot Loader File, and App Area Size.



G+ Code Packer V1.0.8.10

Body Selection: GPL329XX (Package Version v1.0.9)

NVRAM Type Selection: SDCard

Nand Attribute Selection

NVRAM Vendor:

Mark ID:

Block Size:

Page Size:

BCH Mode:

Header Copies: 16

WP Port:

WP Pin:

TLC Page Per Block:

Pad Class Information:

□ Nand Flash Chip Erase Enable □ Use 2CS Nand □ Use 2pcs Identical Nand

NVRAM Total Size: 2048 MB

DRAM Size (MB): 128

DRAM Type: DDR2

Boot Loader File: V1.0.4 (NandLib N/A)

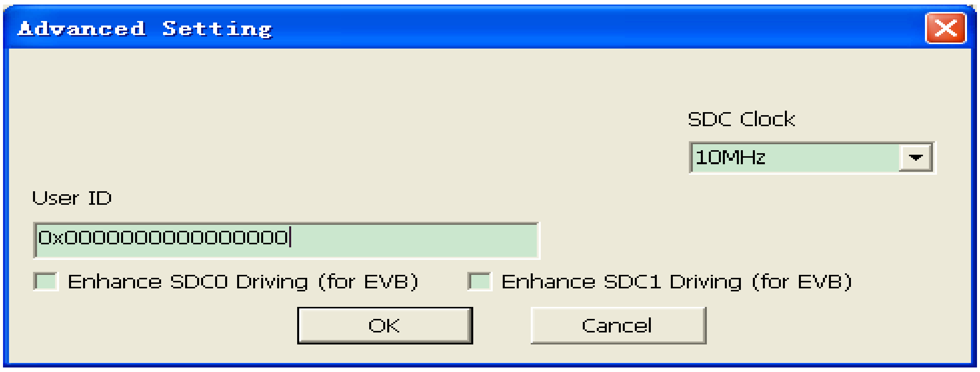
.\BootCode\GPL329XX_SdcBootLoader.hdb

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0xB1000000

Advance setup:

In SD card setup, an advance setting is provided:



Advanced Setting

SDC Clock: 10MHz

User ID: 0x0000000000000000

□ Enhance SDC0 Driving (for EVB) □ Enhance SDC1 Driving (for EVB)

SDC Clock: This setting is to define the access speed from IC to SD card. 10MHz is the default. Options include 400KHz, 5MHz, 10MHz, 20MHz, and 40MHz.

User ID: Defined by user, e.g. devoted for device manufacture's lot number, date, version, etc for future upgrade or maintenance.

Other two are mainly the enhancement of driving capability while SDC0/1 is programming on the EVB.

8.3 Set up App Area

Setup is the same as NAND Flash. Please refer to Chapter 5.3 for more details.

8.4 Set up Data Area and use FAT Image Tool

Setup is the same as NAND Flash. Please refer to Chapter 5.4 for more details.

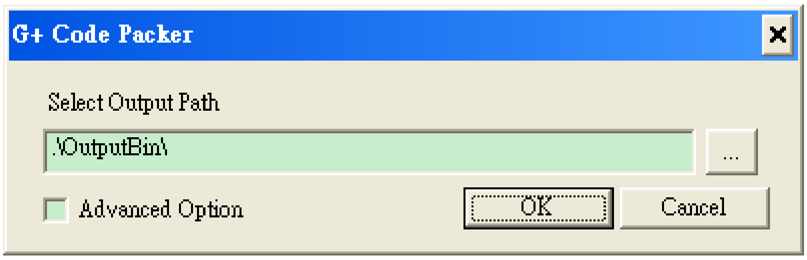
8.5 Set up Pre-set Register and use DRAM Calibration

Setup is the same as NAND Flash. Please refer to Chapter 5.5 for more details.

8.6 Pack and Download

Step 1：Pack

After all areas (Boot/ App/ Data) are set up completely, click the pack button if necessary. Thus, tool will integrate all resource files to a BIN file according to the settings. It will also store the BIN file under a directory that is designated by user (the default directory is suggested).



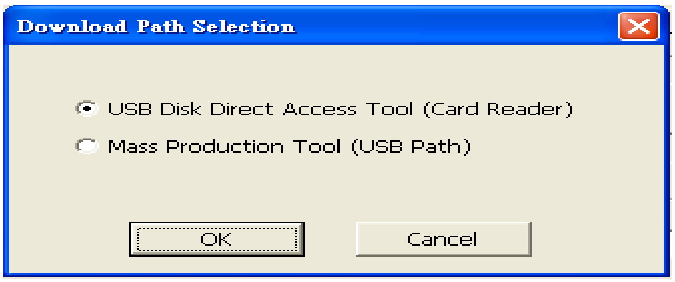
G+ Code Packer

Select Output Path: .\OutputBin\

□ Advanced Option

Step 2：Download

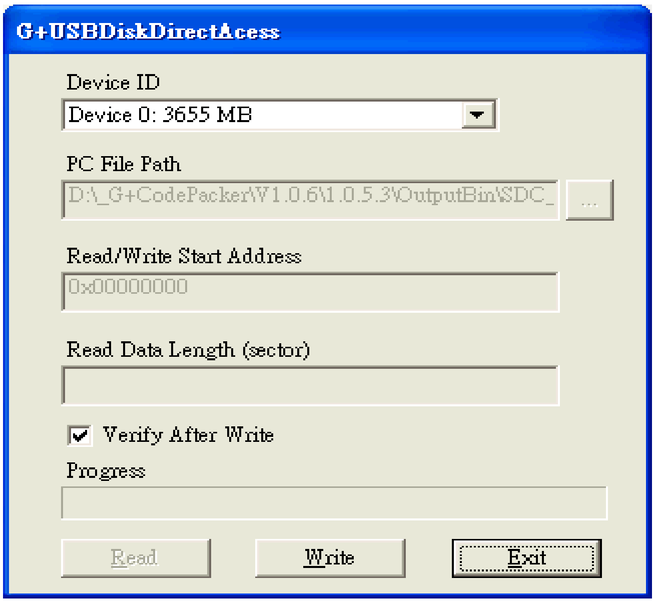
After pack is completed, Download button is triggered. Click Download button to select reader (reader must connect with a PC) or MP tool for programming SD Card. The reader download will evoke USB diskDirectAccess tool and set up all parameters before pressing Write key to write into SDCard. Mount the SDCard onto a target board to proceed SD boot. MP Tool download will evoke G+MassProductionTool and download the SD boot image onto the SD Card on the target board.



Download Path Selection

● USB Disk Direct Access Tool (Card Reader)

○ Mass Production Tool (USB Path)



G+USBDiskDirectAcess

Device ID: Device 0: 3655 MB

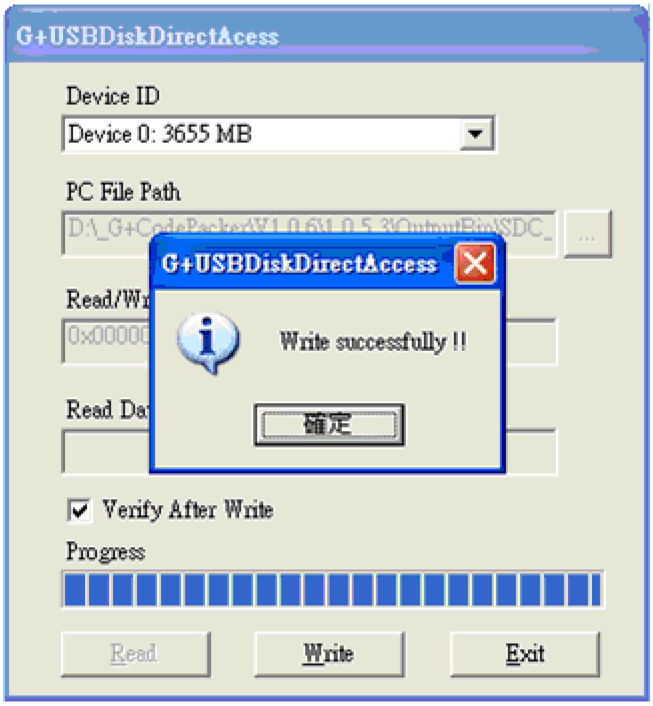
PC File Path: D:_G+CodePacker\V1.0.6\1.0.5.3\OutputBin\SDC_

Read/Write Start Address: 0x00000000

Read Data Length (sector):

☑ Verify After Write

Progress



G+USBDiskDirectAcess

Device ID: Device 0: 3655 MB

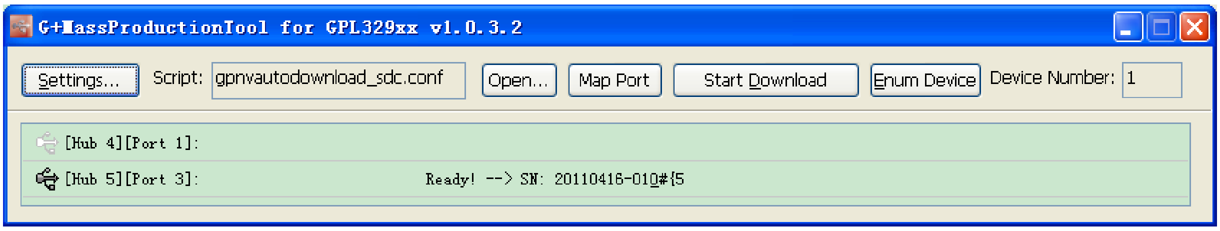
PC File Path: D:_G+CodePackerV1.0.6\1.0.5.3\OutputBin\SDC_

Read/Wr: 0x00000

G+USBDiskDirectAccess

Write successfully !!

☑ Verify After Write



G+MassProductionTool for GPL329xx v1.0.3.2

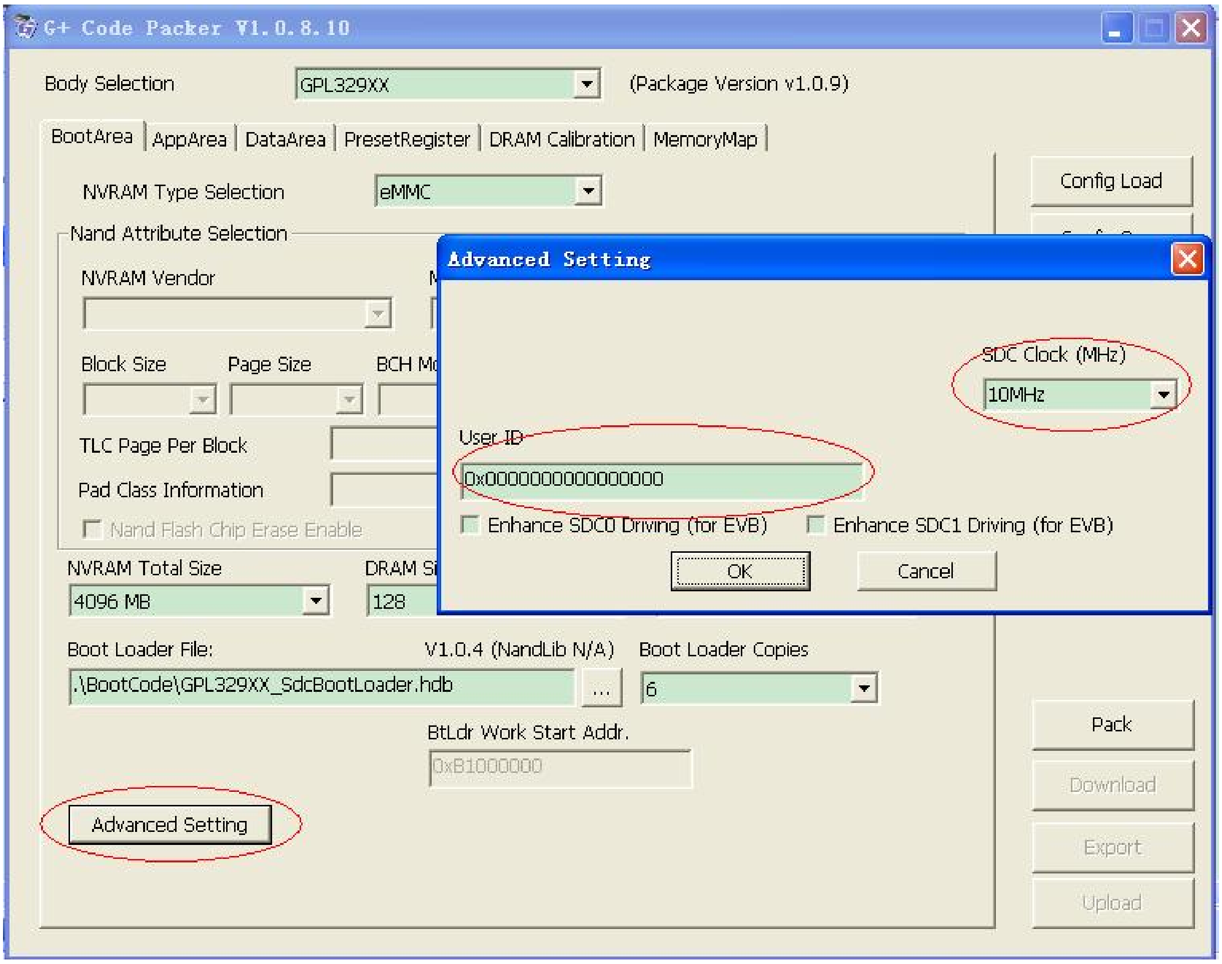
Script: gpnvautodownload_sdc.conf

Device Number: 1

```
[Hub 4][Port 1]:
[Hub 5][Port 3]:                    Ready! --> SN: 20110416-010#{5
```

9 eMMC Boot Flow (GPL329XX)

GPL329XX will divide eMMC into three operating areas: Boot Area, App Area, and Data Area. GPL329XX will divide the SDCard into three areas: Boot Area, App Area, and Data Area. The resources in these three areas can be integrated via G+CodePacker. If SDC Boot compatibility issue is encountered, we recommend enable Advance Mode (check off Advance Setting) and lower the SDC Clock for eMMC. If the Target Board (e.g. Generalplus EVB) has heavy loading, check SDC driving to enhance the loading capability.



Body Selection: GPL329XX (Package Version v1.0.9)

NVRAM Type Selection: eMMC

NVRAM Total Size: 4096 MB

DRAM Si: 128

Boot Loader File: .\BootCode\GPL329XX_SdcBootLoader.hdb V1.0.4 (NandLib N/A)

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0xB1000000

Advanced Setting

SDC Clock (MHz): 10MHz

User ID: 0x0000000000000000

□ Enhance SDC0 Driving (for EVB) □ Enhance SDC1 Driving (for EVB)

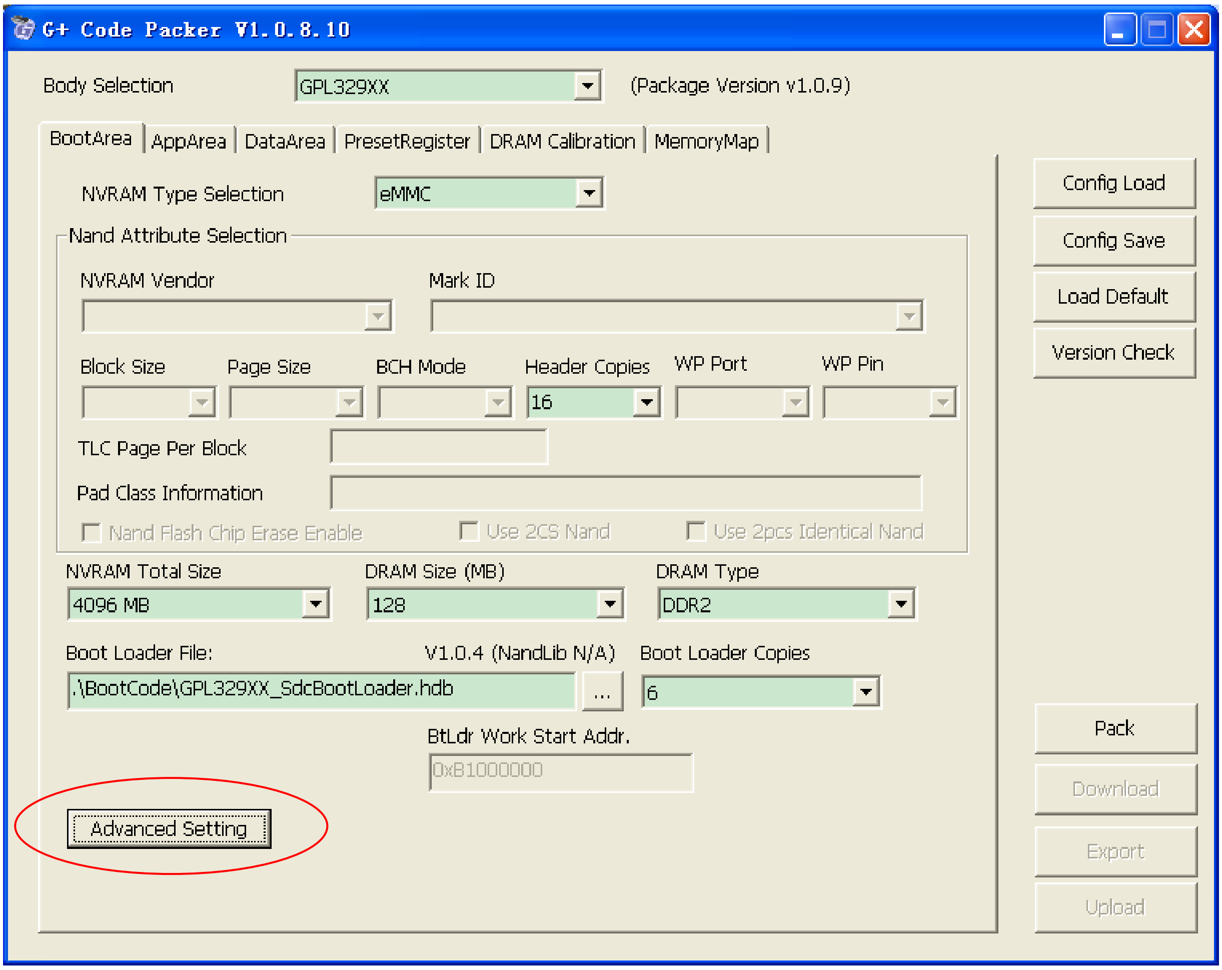
The following sections describe setups for Boot Area, App Area, and Data Area:

9.1 Load Config & Config Save

If we've done the Config Save, we are allowed to re-load the previous parameters via Config Load. We highly recommend storing all resource files under the G+CodePacker's root path. Thus, it increases the portability to easily transport files to associate partners via Config Save.

9.2 Set up Boot Area

Select eMMC and assign NVRAM Total Size (i.e. eMMC Size), DRAM Size, DRAM Type, Boot Loader File, and App Area Size.



G+ Code Packer V1.0.8.10

Body Selection: GPL329XX (Package Version v1.0.9)

NVRAM Type Selection: eMMC

Nand Attribute Selection

NVRAM Vendor:

Mark ID:

Block Size:

Page Size:

BCH Mode:

Header Copies: 16

WP Port:

WP Pin:

TLC Page Per Block:

Pad Class Information:

□ Nand Flash Chip Erase Enable

□ Use 2CS Nand

□ Use 2pcs Identical Nand

NVRAM Total Size: 4096 MB

DRAM Size (MB): 128

DRAM Type: DDR2

Boot Loader File: .\BootCode\GPL329XX_SdcBootLoader.hdb

V1.0.4 (NandLib N/A)

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0xB1000000

9.3 Set up App Area

Same as NAND Flash, refer to Section 5.3.

9.4 Set up Data Area and use FAT Image Tool

Same as NAND Flash, refer to Section 5.4.

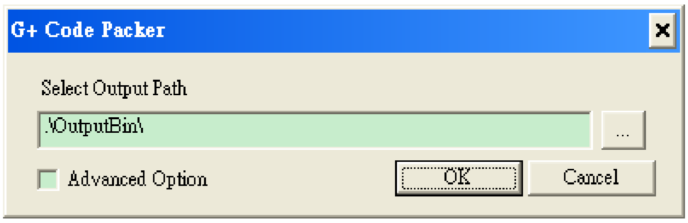
9.5 Set up Pre-set Register and use DRAM Calibration

Same as NAND Flash, refer to Section 5.5

9.6 Pack and Download

Step 1：Pack

After all areas (Boot/ App/ Data) are set up completely, click the pack button if necessary. Thus, tool will integrate all resource files to a BIN file according to the settings. It will also store the BIN file under a directory that is designated by user (the default directory is suggested).

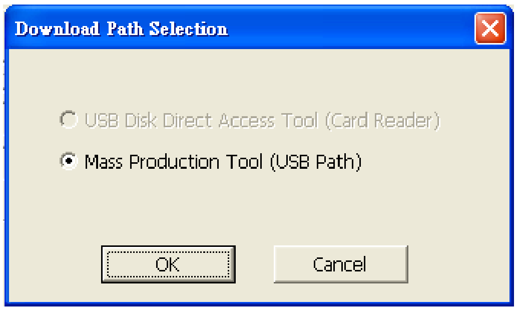

G+ Code Packer

Select Output Path

.\OutputBin\

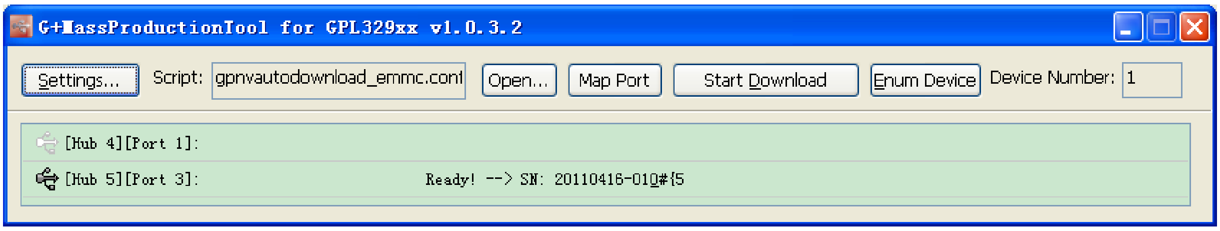
Step 2：Download

After pack is completed, Download button is triggered. Click Download button to select MP tool download which will evoke G+MassProductionTool and download the eMMC boot image onto the eMMC on the target board.


Download Path Selection

○ USB Disk Direct Access Tool (Card Reader)

● Mass Production Tool (USB Path)


G+MassProductionTool for GPL329xx v1.0.3.2

Script: gpnvautodownload_emmc.conf

Device Number: 1

```
[Hub 4][Port 1]:
[Hub 5][Port 3]:                    Ready! --> SN: 20110416-010#{5
```

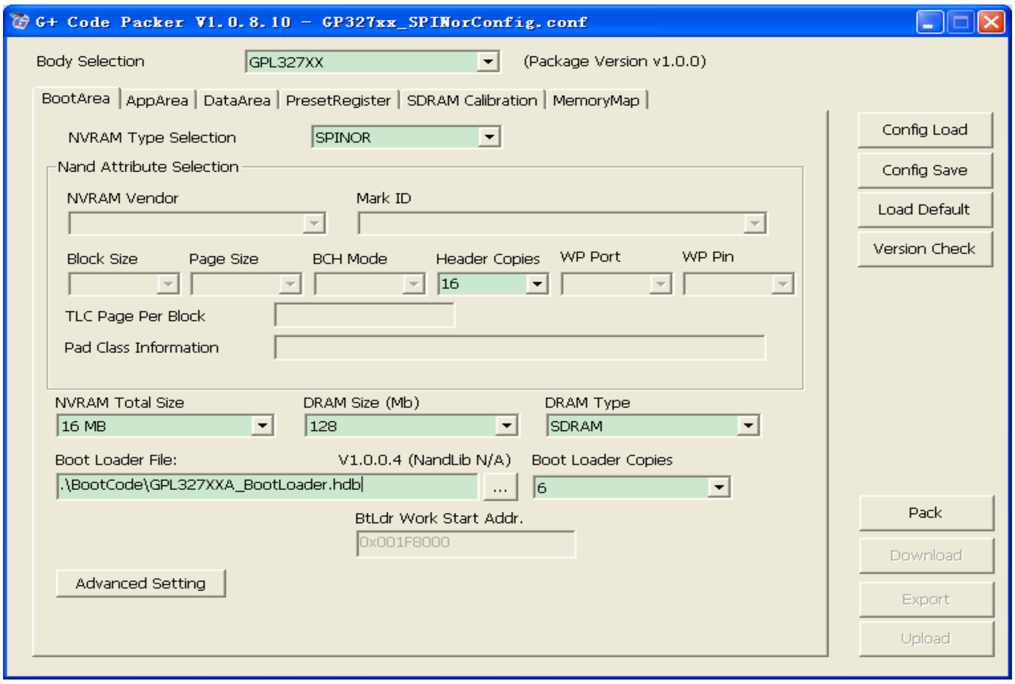
10 SPI NOR Flash (GPL327xx/GPDV6xxx)

Open Code Packer tool and select GPL327xx/GPDV6xxx from body selection pull down menu. The body setup is composed of four parts, Boot Area, App Area, Data Area, and Pre-Register setting.

10.1 Boot Area

10.1.1 Basic Setup for Boot Area

We divide SPI NOR Flash into three areas: Boot Area, App Area, and Data Area. The boot area is to store Boot Header and First Boot Loader in the way of backup plan designed in this area. Boot Header's relevant parameters are assigned in the following window:



G+ Code Packer V1.0.8.10 – GP327xx_SPINorConfig.conf

Body Selection: GPL327XX (Package Version v1.0.0)

NVRAM Type Selection: SPINOR

Header Copies: 16

NVRAM Total Size: 16 MB

DRAM Size (Mb): 128

DRAM Type: SDRAM

Boot Loader File: .\BootCode\GPL327XXA_BootLoader.hdb

V1.0.0.4 (NandLib N/A)

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0x001F8000

From the window above, there are 7 options we need to set up:

- NVRAM Type Selection:

  In present, the pull-down menu has three NVRAMs to select: SPI NOR, SD Card, and eMMC. Here, we select SPINOR for SPI NOR Flash.

- Header Copies:

  To prevent Boot Area being damaged by outer interference, the tool will make backup copy for Boot Header and Boot Loader. In general, the boot header's size packed by Codepacker is about one sector of SPI Nor Flash, i.e. 512Byte; therefore, the number of copy user assign is the number of sector of SPI NOR Flash.

- NVRAM Total Size:

In this field, it indicates the present SPI NOR memory size, unit in byte. CodePacker currently supports 512KB, 1MB, 2MB, 4MB, 8MB, 16MB, 32MB. User is allowed to select it via pull-down menu.

- DRAM Size:

  In this field, it indicates the DRAM memory size corresponding to GPL327XX/GPDV6xxx, unit in Mbit. The tool currently supports 16Mb, 64Mb, 128Mb, 256Mb, 512Mb, 1024Mb, 2048Mb, 4096Mb (8Mb-1MB). User is allowed to select it via pull-down menu.

- DRAM Type:

  This field specifies the RAM type for calibration and the register for programming ISP bin code.

  Boot Loader Files:

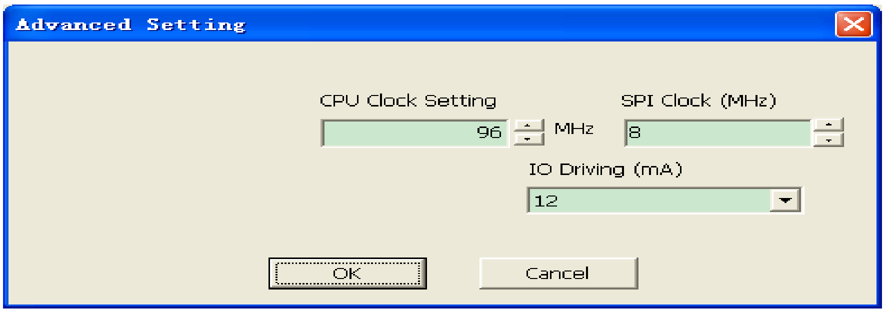
  This field is to select First BootLoader file. In general, which First Bootloader binary file to use is determined by the IC and NV RAM Type selected. This field should be maintain by Generalplus engineer and we recommend user leave this field in its default setting.

- Boot Loader Copies:

  Similar to the Boot Header Copies, this is to assign the number of backup copy. This filed is to assign the number of First Boot loader's copy. Rom code will restore it based on theses copies given to increase the yield rate of SPI NOR Flash.

10.1.2 Advance setup for Boot Area

In addition to the basic setup in Boot Area, an advance setup is also provided. In general,, these advance settings are default value and user can ignore them. SPI Nor Flash advance setup is as follows:



CPU Clock Setting: 96 MHz

SPI Clock (MHz): 8

IO Driving (mA): 12

There are three options in the advance settings: CPU clock setting, SPI Clock an I/O Driving.

- CPU Clock Setting:

CPU clock setting ranges from 4~118MHZ, default in 96MHz. User is allowed to fill in a value to change the clock speed.

- SPI Clock:

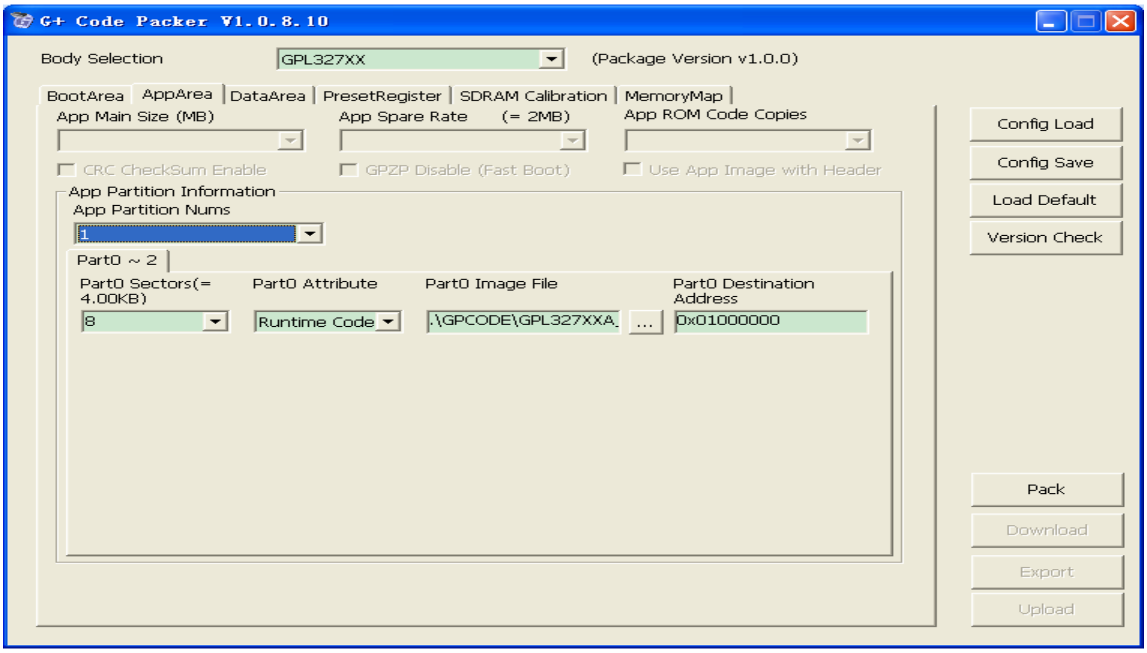
This option is to adjust SPI NOR flash clock, ranging from 0~24MHz, default in 8MHz. It changes the CPU access speed to Flash memory.

- I/O Driving:

This filed is to adjust the driving current on I/O pin; options include 4, 8, 12, 16. When the wiring between EVB's I/O and IC (or SPI NOR Flash) is too short or too long, this option can be used to magnify the IO's output signal or enhance the EMI immune capability. After all options are set completely, click "OK" to enable the modification or click "Cancel" to cancel the new settings.

10.2 Set up App Area

Click App Area tag and UI is as follows:



Body Selection: GPL327XX (Package Version v1.0.0)

App Partition Nums: 1

Part0 Sectors(= 4.00KB): 8

Part0 Attribute: Runtime Code

Part0 Image File: .\GPCODE\GPL327XXA_

Part0 Destination Address: 0x01000000

As the diagram above, it page is mainly for App Partition setup, including number of sub-area and the corresponding parameters for each sub-area.

The parameter setup for App sub-area message are as follows:

- App Partition Num: This item represents the number of current App sub-areas that user define, ranging from 1~16, meaning max sub-area is 16 areas.
- Partition N Size: N ranges 0~15, representing the capacity of current sub-area, unit in Sector. Generally, when we specify the sub-area's imaging file, tool will automatically fill in the corresponding file size for us. Under one circumstance, user does not specify a file and it becomes AUTO mode. If user intend to assign the sub-area's size, simply input the value into field.
- Partition N Attribute: Similarly, N ranges 0~15, representing the type of current imaging file. In

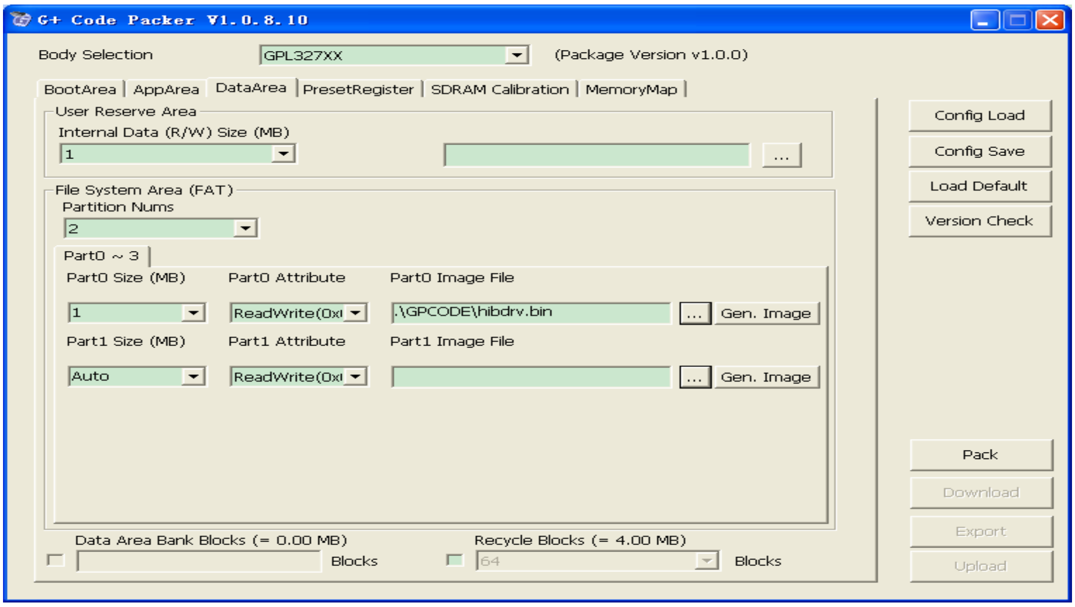
present, Tool offers 7 types of imaging file: Runtime Code, Resourse Part, FastBoot Bin, Quick Image, Hibrate Image, Image Flag, Customize BtLdr. It is important, especially when First bootloader is moving the code from App area.

- Partition N Image file: This parameter allows user to select the sub-area's imaging file on the disk.
- Partition N Destination Address: This parameter is to specify where the current sub-area's file will be stored in SDRAM.

Based on this method, configure and set up all the sub-areas until all areas' messages are configured completely.

10.3 Set up Data Area

After the message in App's sub-area is set up completely, switch it to the Data Area Window, see the following:


G+ Code Packer V1.0.8.10

Body Selection: GPL327XX (Package Version v1.0.0)

User Reserve Area

Internal Data (R/W) Size (MB): 1

File System Area (FAT)

Partition Nums: 2

Part0 ~ 3

Part0 Size (MB): 1
Part0 Attribute: ReadWrite(0x(
Part0 Image File: .\GPCODE\hibdrv.bin

Part1 Size (MB): Auto
Part1 Attribute: ReadWrite(0x(
Part1 Image File:

□ Data Area Bank Blocks (= 0.00 MB) Blocks

□ Recycle Blocks (= 4.00 MB) 64 Blocks

The Data area is composed of user reservation area and file system area. In present, the Recycle block is not supported.

- User reservation area:

This area is to reserve a storage area from data area for user. User is allow to access this area through the read sector/write sector from SPI Flash Driver

- File system area:

This area is similar to the App sub-area. First, we define the expected logic sub-area and then set up each message for every logic sub-area. We now introduce the following options for logic sub-area:

Partition Nums: This filed define the number of logic sub-area, ranging from 0 ~8, total of 8 sub-areas. "0" means no sub-area, no any message for sub-area.

PartN Size: N ranges 1~8, representing the capacity of current sub-area. Generally, when user does not specify a file, it becomes AUTO mode. If user intend to assign the sub-area's size, simply input the value into field.

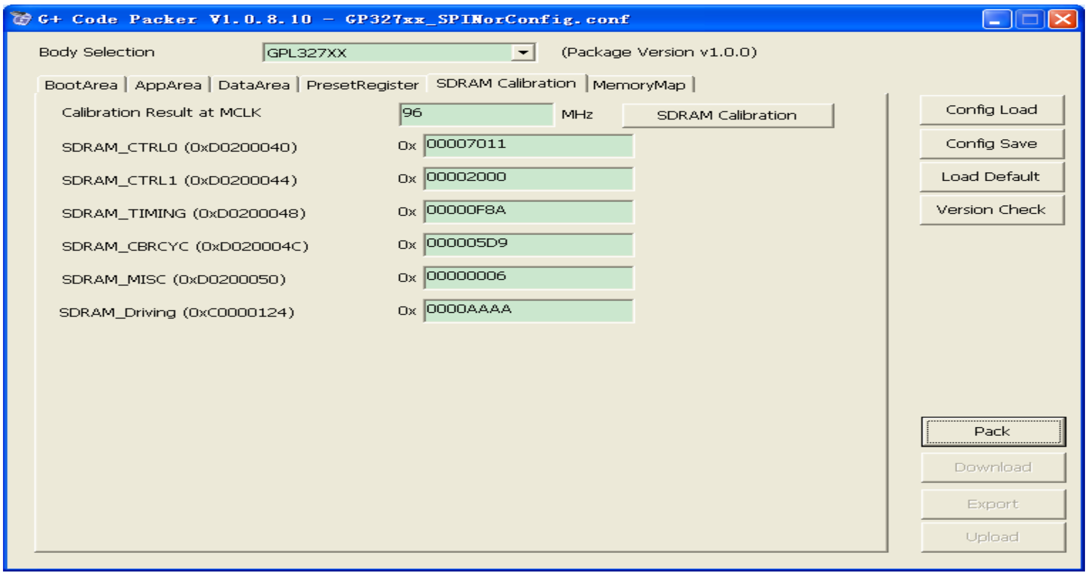
PartN Attribute: N ranges 1~8, representing the type of current imaging file. It is read-only, write-only and read-write options.

PartN Image File: N ranges from 1 ~ 8. This option is to specify the current sub-area's imaging file. There are two ways to specify this filed, clicking "…" button to open file dialog box and select file from local. The other, clicking "Gen.image" button to generate file via G+FatImage Tool. For more information about G+FatImage Tool, please refer to its User's Guide.

In general, although user's reservation area and data sub-area are all data area, their purposes are different. The user's reservation area is to operate this area using the read/write function from SPI Nor Flash driving procedure. The data sub-area, however, is to operate various logic sub-area via file system

10.4 Set up Pre-Register & Dram Calibration

In general, we first run the DRAM calibration to GP327xx/GPDV6xxx and then switch back to Pre-Register to check the corresponding register's content. Before entering the code programming process, make sure DRAM calibration has been completed. Switch to Dram Calibration tag, DRAM Calibration's UI is as follows:



Body Selection: GPL327XX (Package Version v1.0.0)

Calibration Result at MCLK: 96 MHz

SDRAM_CTRL0 (0xD0200040) 0x 00007011

SDRAM_CTRL1 (0xD0200044) 0x 00002000

SDRAM_TIMING (0xD0200048) 0x 00000F8A

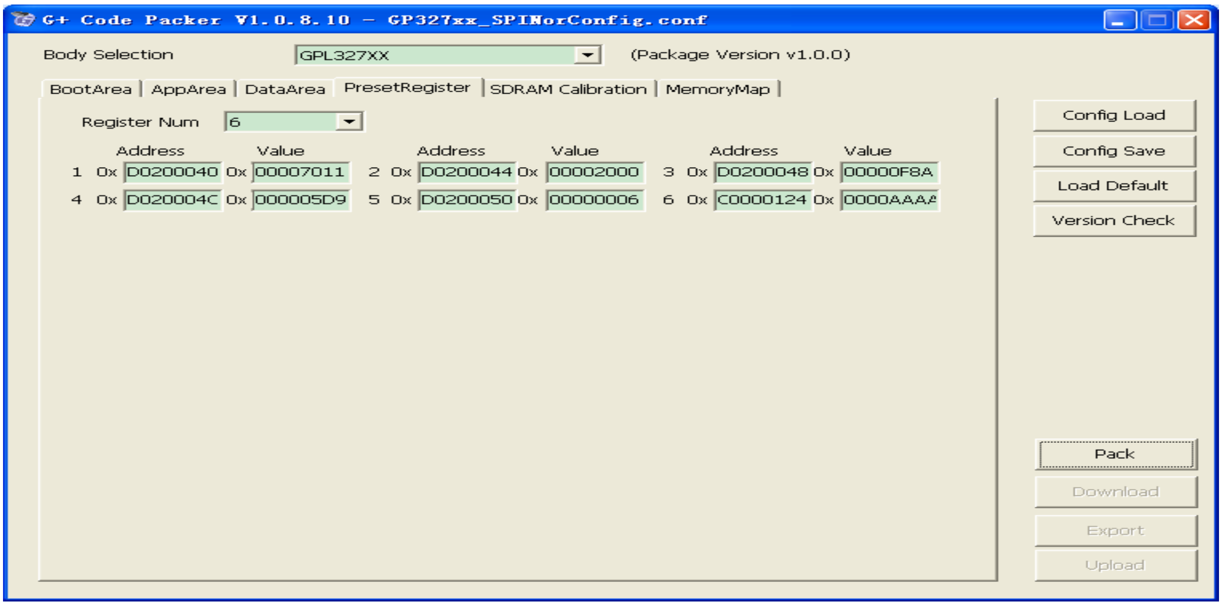
SDRAM_CBRCYC (0xD020004C) 0x 000005D9

SDRAM_MISC (0xD0200050) 0x 00000006

SDRAM_Driving (0xC0000124) 0x 0000AAAA

Click "SDRAM calibration button" to activate G+ SDRAM calibration tool. Calibrate SDRAM register via this tool. For more information, please refer to G+ SDRAM Calibration Tool User's Guide. After

SDRAM calibration is completed, switch it to the Pre-register tag and we will see the values in calibration tool has been updated in the Pre-register window. It looks like:



G+ Code Packer V1.0.8.10 – GP327xx_SPINorConfig.conf

Body Selection: GPL327XX (Package Version v1.0.0)

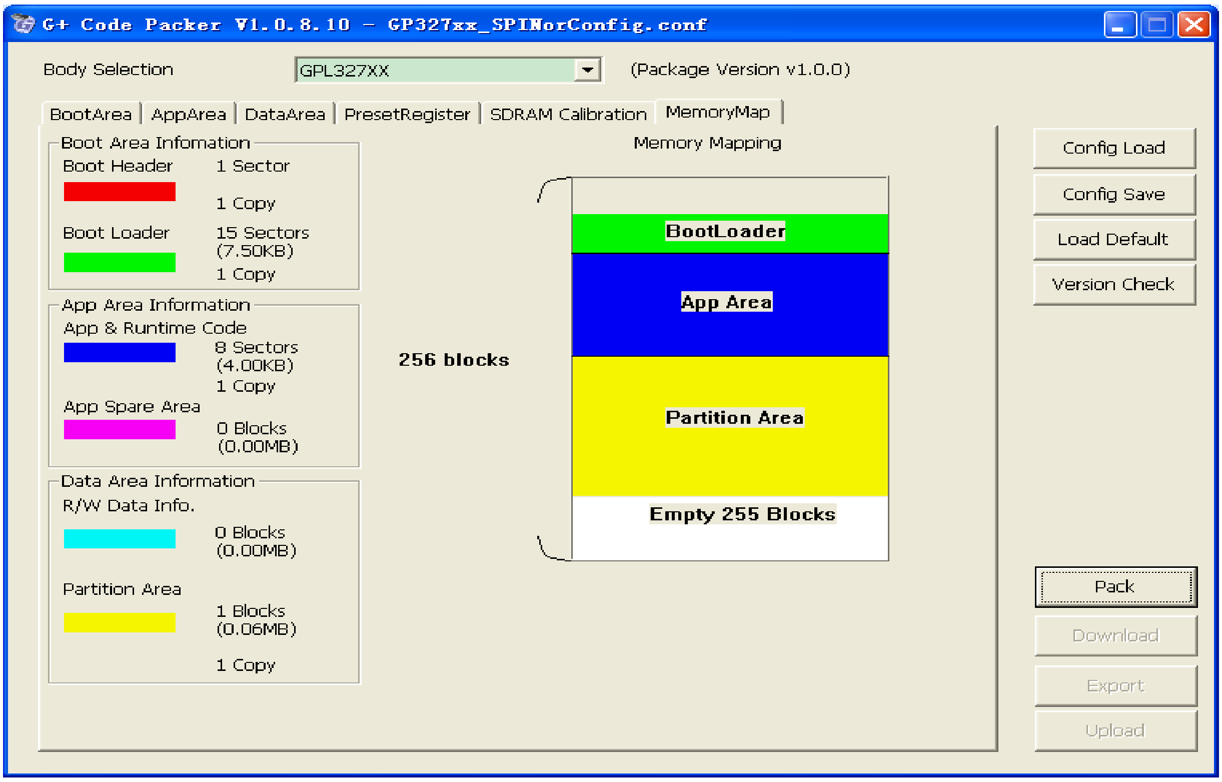
Register Num: 6

	Address	Value		Address	Value		Address	Value
1	0x D0200040	0x 00007011	2	0x D0200044	0x 00002000	3	0x D0200048	0x 00000F8A
4	0x D020004C	0x 000005D9	5	0x D0200050	0x 00000006	6	0x C0000124	0x 0000AAAA

From the window above, it is composed of two parts: number of registers and register's address and the content in address. For some reasons, DRAM calibration may not be effective as we expect in some ways or some configuration adjustments for booting system (e.g. clock or IO port status). In such cases, we are allowed to modify the settings in this window manually.

10.5 Memory Map

After Boot, App, and data zones are configured, we can switch to this page and view the scope of SPI NOR Flash configuration; see the following figure.



Body Selection GPL327XX (Package Version v1.0.0)

Boot Area Infomation
Boot Header 🟥 1 Sector 1 Copy
Boot Loader 🟩 15 Sectors (7.50KB) 1 Copy

App Area Information
App & Runtime Code 🟦 8 Sectors (4.00KB) 1 Copy
App Spare Area 🟪 0 Blocks (0.00MB)

Data Area Information
R/W Data Info. 0 Blocks (0.00MB)
Partition Area 🟨 1 Blocks (0.06MB) 1 Copy

Note that the figures in this window may not be accurate. We will fix it in the later version. This will not affect the actual layout in the physical register.

10.6 The Button Function

After all settings are completed, click “Config Save” to save the settings. We can re-load the previous settings by using “Config Load”. Note that the new version of Code Packer Tool should not be used to load the old settings saved by older version of tool because various versions of tools may result in compatibility issue.

Load Default: This button allows user to clear all present settings to the tool’s default settings without closing Code Packer.

Version Check: this button checks the version of hdb file.

Pack: This button packs certain messages to a file for programming based on the rule given.

Download：Call G+ Mass Production Tool to run programming task.

Export: This button outputs the packed file and program onto a disk. Simply open G+ Mass Production tool to program files. It is especially for mass production purpose.

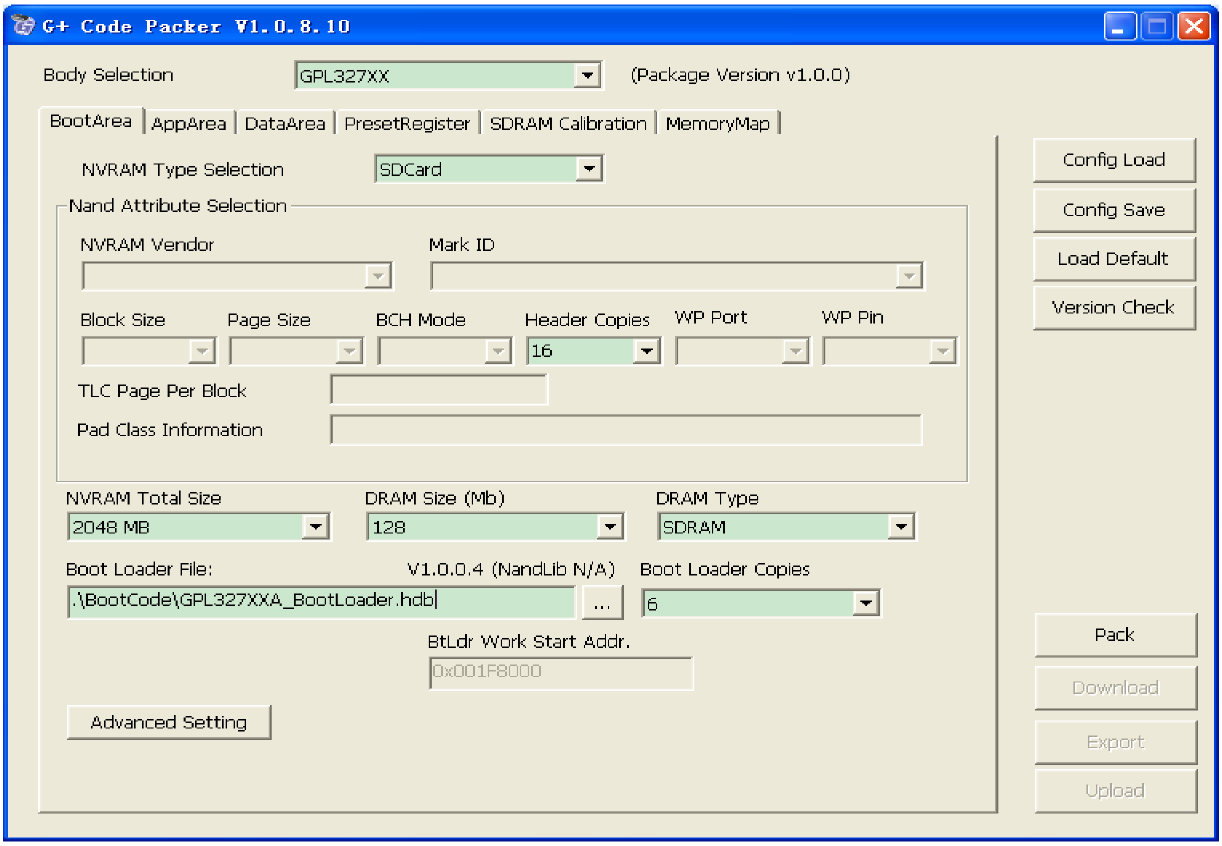
11 SD Card Operation (GPL327xx/GPDV6xxx)

Open Code Packer tool and select GPL327xx/GPDV6xxx from body selection pull down menu. The body setup is composed of four parts, Boot Area, App Area, Data Area, and Pre-Register setting.

11.1 Set up Boot Area

11.1.1 Basic setup for Boot Area

We divide SD Card into three areas: Boot Area, App Area, and Data Area. The boot area is to store Boot Header and First Boot Loader in the way of backup plan designed in this area. Boot Header's relevant parameters are assigned in the following window:



G+ Code Packer V1.0.8.10

Body Selection: GPL327XX (Package Version v1.0.0)

NVRAM Type Selection: SDCard

Nand Attribute Selection

NVRAM Vendor:

Mark ID:

Block Size:

Page Size:

BCH Mode:

Header Copies: 16

WP Port:

WP Pin:

TLC Page Per Block:

Pad Class Information:

NVRAM Total Size: 2048 MB

DRAM Size (Mb): 128

DRAM Type: SDRAM

Boot Loader File: .\BootCode\GPL327XXA_BootLoader.hdb

V1.0.0.4 (NandLib N/A)

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0x001F8000

Here we have seven items to set up:

- NVRAM Type Selection:

  Three NVRAM options are available: SPI Nor, SD Card, and eMMC. Here, we select SD Card.

- Header Copies:

  To prevent Boot Area being damaged by outer interference, the tool will make backup copy for Boot Header and Boot Loader. In general, the boot header's size packed by Codepacker is about one sector of SPI Nor Flash, i.e. 512Byte; therefore, the number of copy user assign is the number of sector of SD Card.

- NVRAM Total Size:

  In this field, it indicates the current SD Card memory size, unit in Mbyte. CodePacker currently supports 1024MB, 2048MB, 4096MB, 8192MB, 16384MB, 32768MB. User is allowed to select it via pull-down menu.

- DRAM Size:

  In this field, it indicates the DRAM memory size corresponding to GPL327xx/GPDV6xxx, unit in Mbit. The tool currently supports 16Mb, 64Mb, 128Mb, 256Mb, 512Mb, 1024Mb, 2048Mb, 4096Mb (8Mb-1MB). User is allowed to select it via pull-down menu.

- DRAM Type:
- This field specifies the RAM type for calibration and the register for programming ISP bin code.

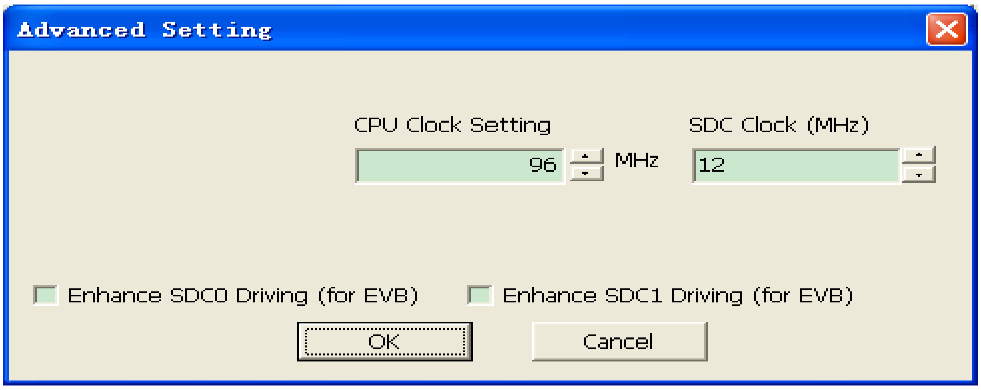
  Boot Loader Files:
- Boot Loader Files:

  This field is to select First BootLoader file. In general, which First Bootloader binary file to use is determined by the IC and NV RAM Type selected. This field should be maintain by Generalplus engineer and we recommend user leave this field in its default setting.
- Boot Loader Copies:

  Similar to the Boot Header Copies, this is to assign the number of backup copy. This filed is to assign the number of First Boot loader's copy. Rom code will restore it based on theses copies given to increase the yield rate of SD Card.

11.1.2 Advance setup for Boot Area

In addition to the basic setup in Boot Area, an advance setup is also provided. In general, these advance settings are default value and user can ignore them. SD card advance setup is as follows:



Advanced Setting

CPU Clock Setting: 96 MHz

SDC Clock (MHz): 12

□ Enhance SDC0 Driving (for EVB)

□ Enhance SDC1 Driving (for EVB)

There are three options in the advance settings: CPU clock setting, SDC Clock, and Enhance SDC0/1 Driving(for EVB).

- CPU Clock Setting

CPU clock setting ranges from 4~118MHZ, default in 108MHz. User is allowed to fill in a value to change the clock speed.

- SDC Clock

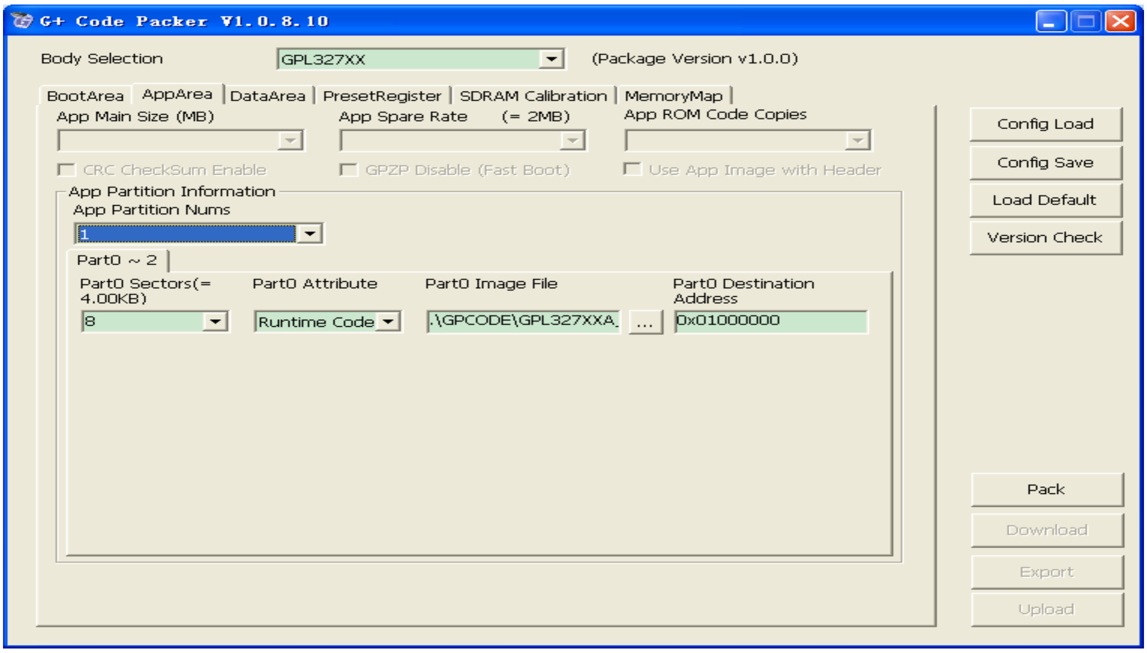
This option is to adjust SPI NOR flash clock, ranging from 0~50MHz, default in 12MHz. It changes the CPU access speed to SD card.

- Enhance SDC0/1 Driving(for EVB)

This filed is to enhance the SDC0/1 driving current. When the wiring (route) between EVB's I/O and IC (or SDC0/1) is too short or too long, this option can be used to magnify theSD Card I/O output signal or enhance the EMI immune capability. After all options are set completely, click “OK” to enable the modification or click “Cancel” to cancel the new settings.

11.2 Set up App Area

Click App Area Tag and the window is as follows:



Body Selection: GPL327XX (Package Version v1.0.0)

App Partition Nums: 1

Part0 Sectors(= 4.00KB): 8

Part0 Attribute: Runtime Code

Part0 Image File: .\GPCODE\GPL327XXA_

Part0 Destination Address: 0x01000000

As the diagram above, it page is mainly for App Partition setup, including number of sub-area and the corresponding parameters for each sub-area.

The parameter setup for App sub-area message are as follows:

- App Partition Num: This item represents the number of current App sub-areas that user define, ranging from 1~16, meaning max sub-area is 16 areas.
- Partition N Size: N ranges 0~15, representing the capacity of current sub-area, unit in Sector.

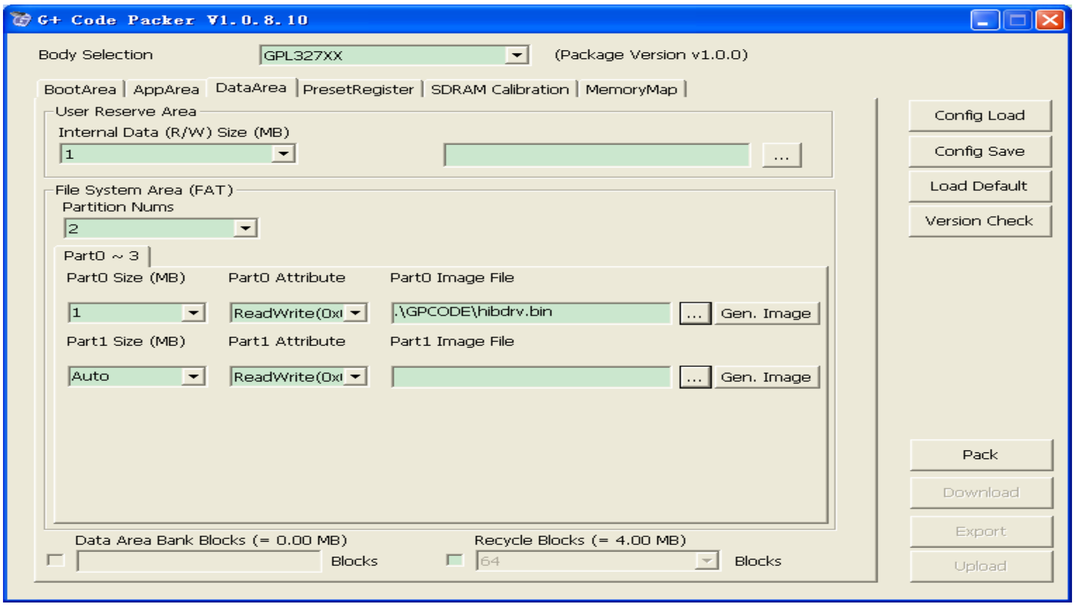
Generally, when we specify the sub-area’s imaging file, tool will automatically fill in the corresponding file size for us. Under one circumstance, user does not specify a file and it becomes AUTO mode. If user intend to assign the sub-area’s size, simply input the value into field.

- Partition N Attribute: Similarly, N ranges 0~15, representing the type of current imaging file. In present, Tool offers 7 types of imaging file: Runtime Code, Resourse Part, FastBoot Bin, Quick Image, Hibrate Image, Image Flag, Customize BtLdr. It is important, especially when First bootloader is moving the code from App area.
- Partition N Image file: This parameter allows user to select the sub-area’s imaging file on the disk.
- Partition N Destination Address: This parameter is to specify where the current sub-area’s file will be stored in SDRAM.

Based on this method, configure and set up all the sub-areas until all areas’ messages are configured completely.

11.3 Set up Data Area

After the App area’s sub-area message is set completely, switch the window to Data Area; it looks like the following:



G+ Code Packer V1.0.8.10

Body Selection: GPL327XX (Package Version v1.0.0)

User Reserve Area

Internal Data (R/W) Size (MB): 1

File System Area (FAT)

Partition Nums: 2

Part0 Size (MB): 1

Part0 Attribute: ReadWrite(0x

Part0 Image File: .\GPCODE\hibdrv.bin

Part1 Size (MB): Auto

Part1 Attribute: ReadWrite(0x

Part1 Image File:

□ Data Area Bank Blocks (= 0.00 MB) Blocks

□ Recycle Blocks (= 4.00 MB) 64 Blocks

The data area composed of user reservation area and file system area. In present, recycle block is not supported.

- User reservation area:

This area is to reserve a storage area from data area for user. User is allow to access this area through the read sector/write sector from SD Card Driver.

- File system area:

This area is similar to the App sub-area. First we define the expected logic sub-area and then set up each message for every logic sub-area. We now introduce the following options for logic sub-area:

Partition Nums: This filed define the number of logic sub-area, ranging from 0 ~8, total of 8 sub-areas. "0" means no sub-area, no any message for sub-area.

PartN Size: N ranges from1 ~8, representing the capacity of current sub-area. When imaging file is not selected, default in Auto or user is allowed to specify the value.

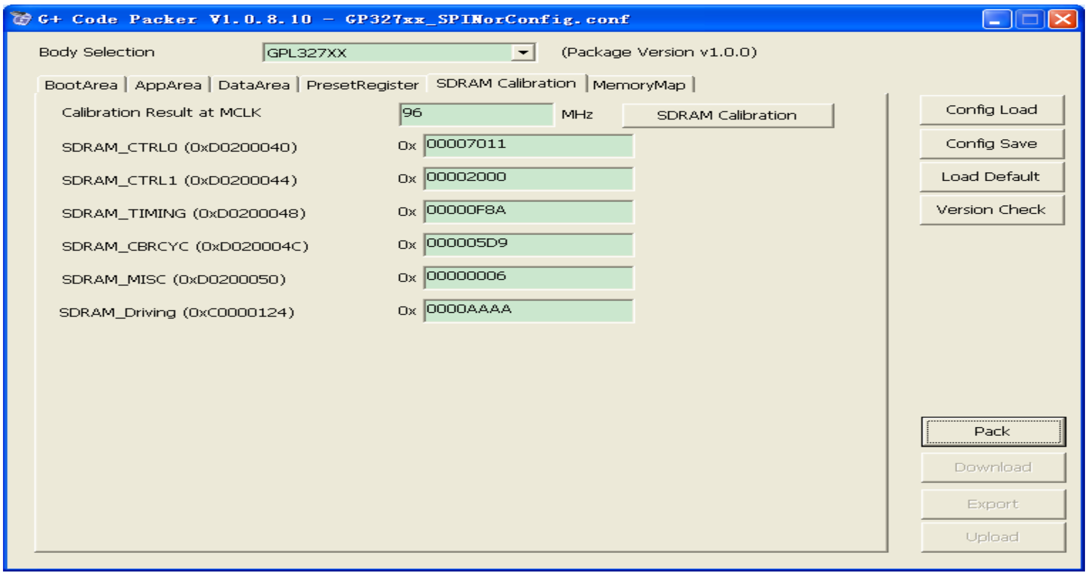
PartN Attribute: N ranges 1 ~8, representing the attribute of current sub-area. There are three options: read-only, write-only and read-write.

PartN Image File:N ranges from 1 ~ 8. This option is to specify the current sub-area's imaging file. There are two ways to specify this filed, clicking "…" button to open file dialog box and select file from local. The other, clicking "Gen.image" button to generate file via G+FatImage Tool. For more information about G+FatImage Tool, please refer to its User's Guide.

In general, although user's reservation area and data sub-area are all data area, their purposes are different. The user's reservation area is to operate this area using the read/write function from SD card driving procedure. The data sub-area, however, is to operate various logic sub-area via file system

11.4 Pre-Register & DRAM Calibration

In general, we first run the DRAM calibration to GP327xx and then switch back to Pre-Register to check the corresponding register's content. Before entering the code programming process, make sure DRAM calibration has been completed. Switch to Dram Calibration tag, DRAM Calibration's UI is as follows:



G+ Code Packer V1.0.8.10 – GP327xx_SPINorConfig.conf

Body Selection: GPL327XX (Package Version v1.0.0)

Calibration Result at MCLK: 96 MHz

SDRAM_CTRL0 (0xD0200040) 0x 00007011

SDRAM_CTRL1 (0xD0200044) 0x 00002000

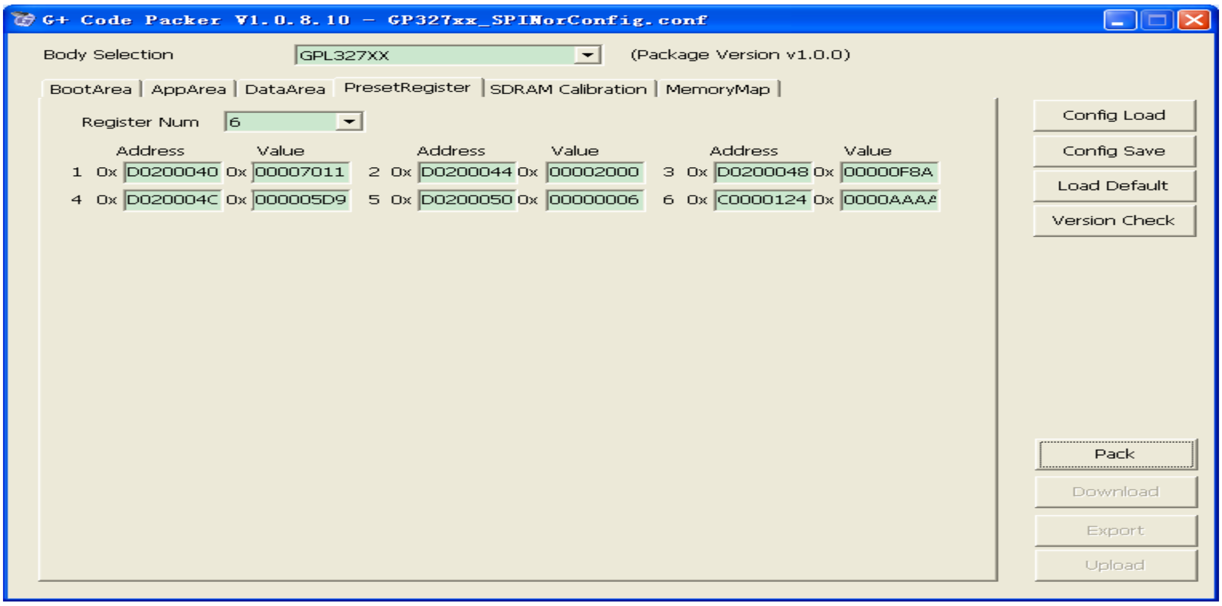
SDRAM_TIMING (0xD0200048) 0x 00000F8A

SDRAM_CBRCYC (0xD020004C) 0x 000005D9

SDRAM_MISC (0xD0200050) 0x 00000006

SDRAM_Driving (0xC0000124) 0x 0000AAAA

Click "SDRAM calibration button" to activate G+ SDRAM calibration tool. Calibrate SDRAM register via this tool. For more information, please refer to G+ SDRAM Calibration Tool User’s Guide. After SDRAM calibration is completed, switch it to the Pre-register tag and we will see the values in calibration tool has been updated in the Pre-register window. It looks like:



G+ Code Packer V1.0.8.10 – GP327xx_SPINorConfig.conf

Body Selection: GPL327XX (Package Version v1.0.0)

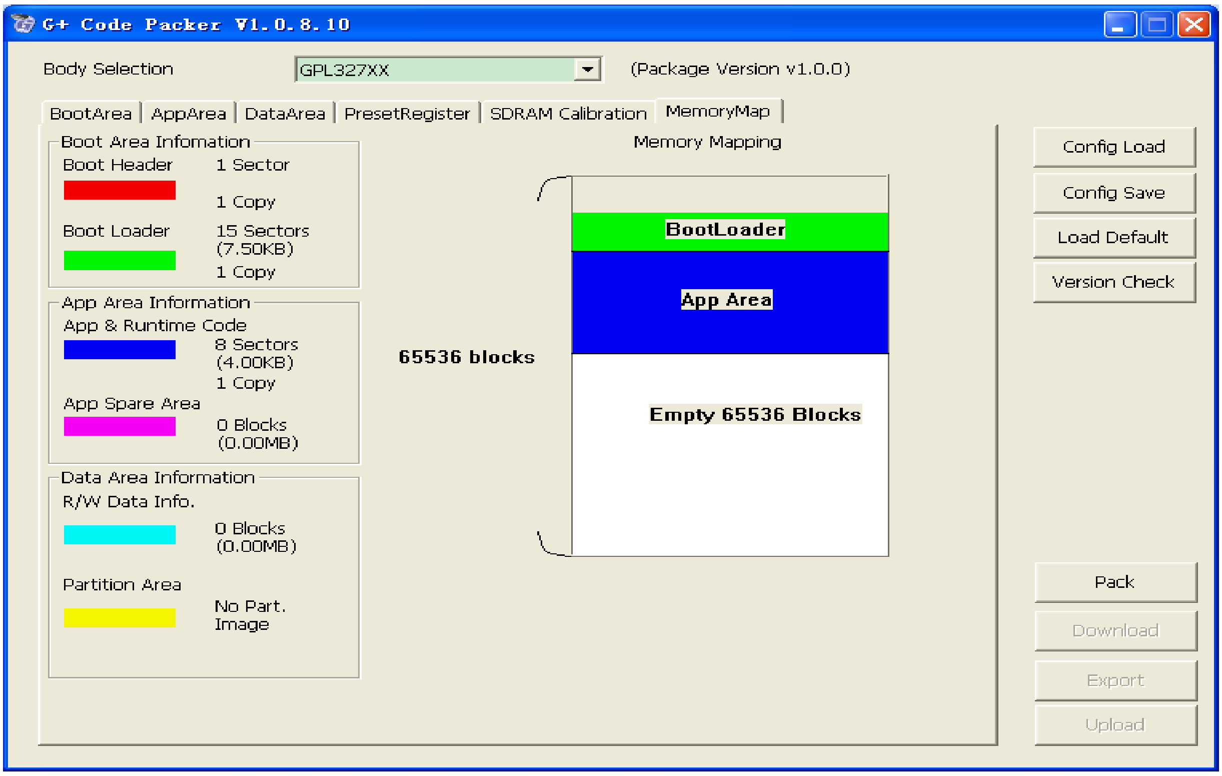
Register Num: 6

	Address	Value		Address	Value		Address	Value
1	0x D0200040	0x 00007011	2	0x D0200044	0x 00002000	3	0x D0200048	0x 00000F8A
4	0x D020004C	0x 000005D9	5	0x D0200050	0x 00000006	6	0x C0000124	0x 0000AAAA

From the window above, it is composed of two parts: number of registers and register’s address and the content in address. For some reasons, DRAM calibration may not be effective as we expect in some ways or some configuration adjustments for booting system (e.g. clock or IO port status). In such cases, we are allowed to modify the settings in this window manually.

11.5 Memory Map

After configuring Boot Area, App Area and Data Area, switch to this page and we will see the SD Card layout as follows:



Body Selection: GPL327XX (Package Version v1.0.0)

Boot Area Infomation
- Boot Header 🟥 1 Sector, 1 Copy
- Boot Loader 🟩 15 Sectors (7.50KB), 1 Copy

App Area Information
- App & Runtime Code 🟦 8 Sectors (4.00KB), 1 Copy
- App Spare Area 🟪 0 Blocks (0.00MB)

Data Area Information
- R/W Data Info. 🟦 0 Blocks (0.00MB)
- Partition Area 🟨 No Part. Image

Note that the figures in this window may not be accurate. We will fix it in the later version. This will not affect the actual layout in the physical register.

11.6 The Button Function

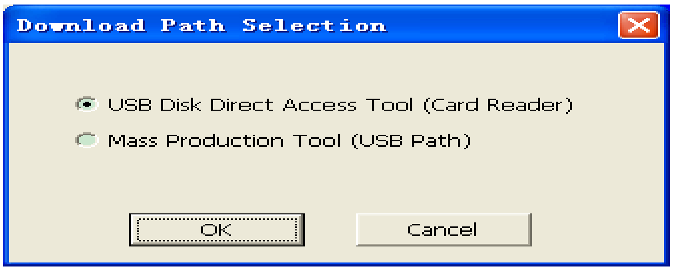
After all settings are completed, click “Config Save” to save the settings. We can re-load the previous settings by using “Config Load”. Note that the new version of Code Packer Tool should not be used to load the old settings saved by older version of tool because various versions of tools may result in compatibility issue.

Load Default: This button allows user to clear all present settings to the tool’s default settings without closing Code Packer.

Version Check: this button checks the version of hdb file.

Pack: This button packs certain messages to a file for programming based on the rule given.

Download：Call G+ Mass Production Tool or G+ USB Disk Direct Access Tool, see the following window:



● USB Disk Direct Access Tool (Card Reader)

○ Mass Production Tool (USB Path)

In these two methods, the first method is to access the SD card via a SD card reader. The other method is to insert SD card onto the EVB’s SD socket which is allowed to be written and read via GP327xx/GPDV6xx. (please refer to section 8.6 for more details)

Export: This button outputs the packed file and program onto a disk. Simply openG+ Mass Production tool to program files. It is especially for mass production purpose.

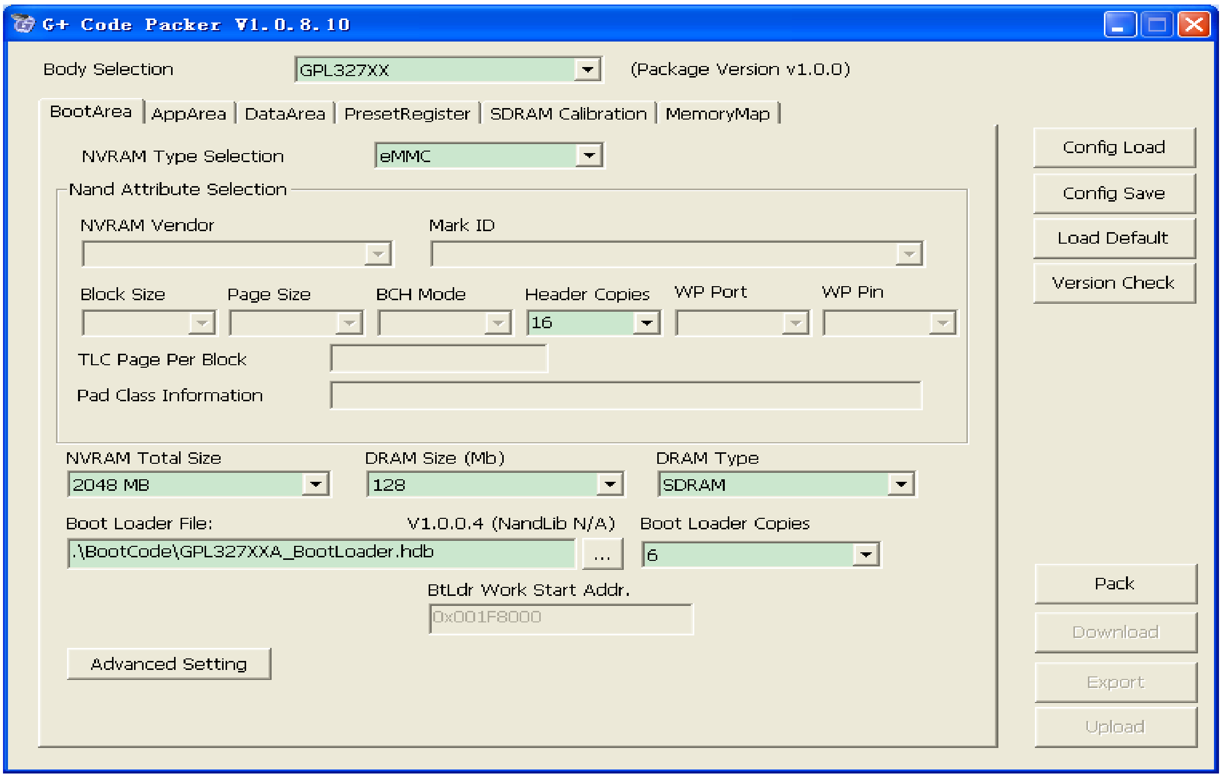
12 eMMC (GPL327xx/GPDV6xxx)

Open Code Packer tool and select GPL327xx/GPDV6xxx from body selection pull down menu. The body setup is composed of four parts, Boot Area, App Area, Data Area, and Pre-Register setting.

12.1 Set up Boot Area

12.1.1 Basic setup for Boot Area

We divide SD Card into three areas: Boot Area, App Area, and Data Area. The boot area is to store Boot Header and First Boot Loader in the way of backup plan designed in this area. Boot Header's relevant parameters are assigned in the following window:



G+ Code Packer V1.0.8.10

Body Selection: GPL327XX (Package Version v1.0.0)

NVRAM Type Selection: eMMC

Nand Attribute Selection

NVRAM Vendor:

Mark ID:

Block Size:

Page Size:

BCH Mode:

Header Copies: 16

WP Port:

WP Pin:

TLC Page Per Block:

Pad Class Information:

NVRAM Total Size: 2048 MB

DRAM Size (Mb): 128

DRAM Type: SDRAM

Boot Loader File: .\BootCode\GPL327XXA_BootLoader.hdb

V1.0.0.4 (NandLib N/A)

Boot Loader Copies: 6

BtLdr Work Start Addr.: 0x001F8000

We have seven items to set up:

- ➢ NVRAM Type Selection:

  Three NVRAM options are available: SPI Nor, SD Card, and eMMC. Here, we select eMMC.

- ➢ Header Copies:

  To prevent Boot Area being damaged by outer interference, the tool will make backup copy for Boot Header and Boot Loader. In general, the boot header's size packed by Codepacker is about one sector of SPI Nor Flash, i.e. 512Byte; therefore, the number of copy user assign is the number of sector of eMMC.

- NVRAM Total Size:

  In this field, it indicates the current eMMC memory size, unit in Mbyte. CodePacker currently supports 1024MB, 2048MB, 4096MB, 8192MB, 16384MB, 32768MB. User is allowed to select it via pull-down menu.

- DRAM Size:

  In this field, it indicates the DRAM memory memory size corresponding to GPL327XX, unit in Mbit. The tool currently supports 16Mb, 64Mb, 128Mb, 256Mb, 512Mb, 1024Mb, 2048Mb, 4096Mb (8Mb-1MB). User is allowed to select it via pull-down menu.

- DRAM Type:

  This field specifies the RAM type for calibration and the register for programming ISP bin code.

  Boot Loader Files:

- Boot Loader Files:

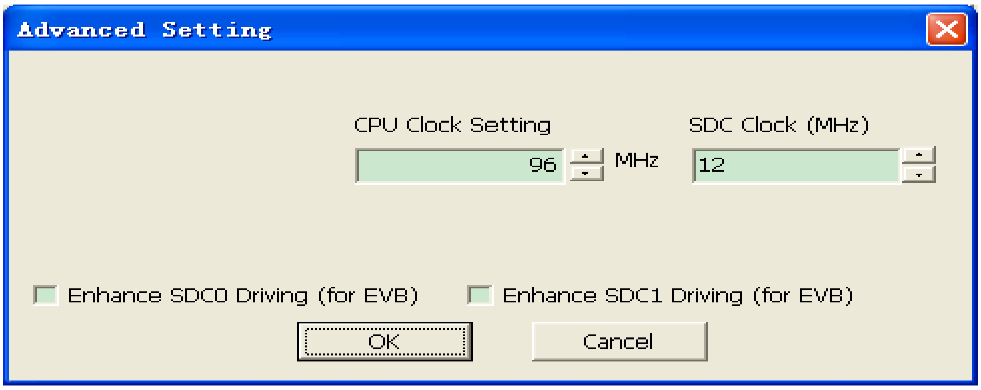
  This field is to select First BootLoader file. In general, which First Bootloader binary file to use is determined by the IC and NV RAM Type selected. This field should be maintain by Generalplus engineer and we recommend user leave this field in its default setting.

- Boot Loader Copies:

  Similar to the Boot Header Copies, this is to assign the number of backup copy. This filed is to assign the number of First Boot loader's copy. Rom code will restore it based on theses copies given to increase the yield rate of eMMC.

12.1.2 Advance setup for Boot Area

In addition to the basic setup in Boot Area, an advance setup is also provided. In general, these advance settings are default value and user can ignore them. eMMC advance setup is as follows:



Advanced Setting

CPU Clock Setting: 96 MHz

SDC Clock (MHz): 12

□ Enhance SDC0 Driving (for EVB)

□ Enhance SDC1 Driving (for EVB)

There are three options in the advance settings: CPU clock setting, SDC Clock, and Enhance SDC0/1 Driving(for EVB).

- CPU Clock Setting

CPU clock setting ranges from 4~118MHZ, default in 108MHz. User is allowed to fill in a value to change the clock speed.

- SDC Clock

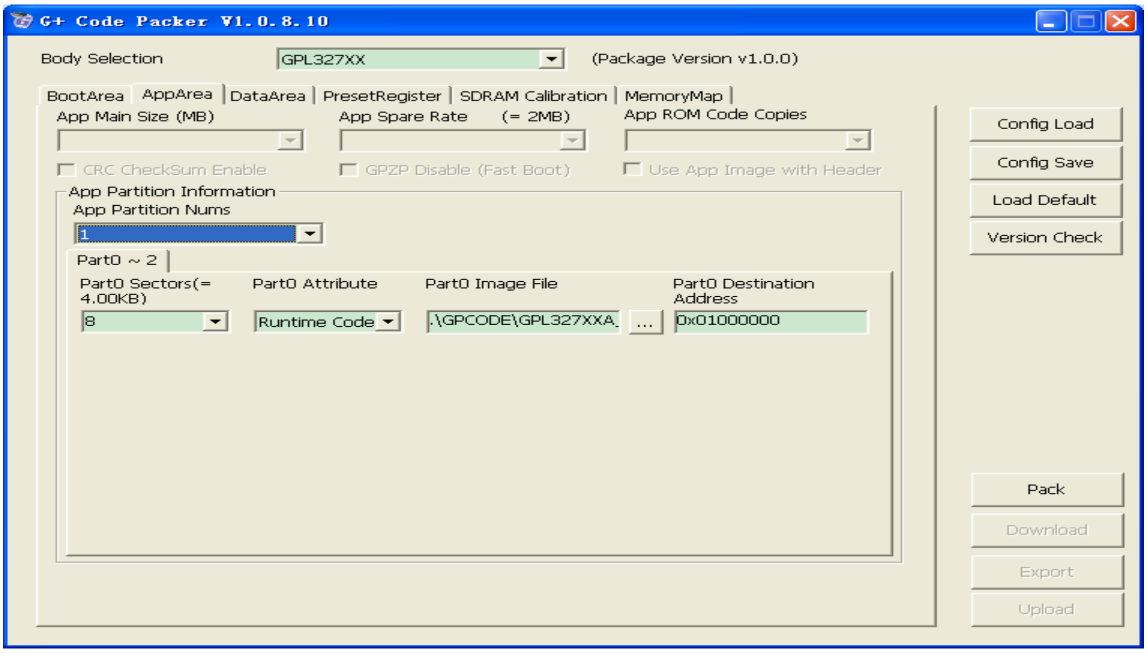
This option is to adjust eMMC clock, ranging from 0~50MHz, default in 12MHz. It changes the CPU access speed to Flash memory.

- Enhance SDC0/1 Driving(for EVB)

This filed is to enhance the eMMC0/1 driving current. When the wiring (route) between EVB's I/O and IC (or eMMC0/1) is too short or too long, this option can be used to magnify the eMMC I/O output signal or enhance the EMI immune capability. After all options are set completely, click "OK" to enable the modification or click "Cancel" to cancel the new settings.

12.2 Set up App Area

Click App Area tag and switch it to App Area, UI is as follows:



Body Selection: GPL327XX (Package Version v1.0.0)

App Main Size (MB)

App Spare Rate (= 2MB)

App ROM Code Copies

□ CRC CheckSum Enable

□ GPZP Disable (Fast Boot)

□ Use App Image with Header

App Partition Information

App Partition Nums: 1

Part0 ~ 2

Part0 Sectors(= 4.00KB)	Part0 Attribute	Part0 Image File	Part0 Destination Address
8	Runtime Code	.\GPCODE\GPL327XXA.	0x01000000

As the diagram above, it page is mainly for App Partition setup, including number of sub-area and the corresponding parameters for each sub-area.

The parameter setup for App sub-area message are as follows:

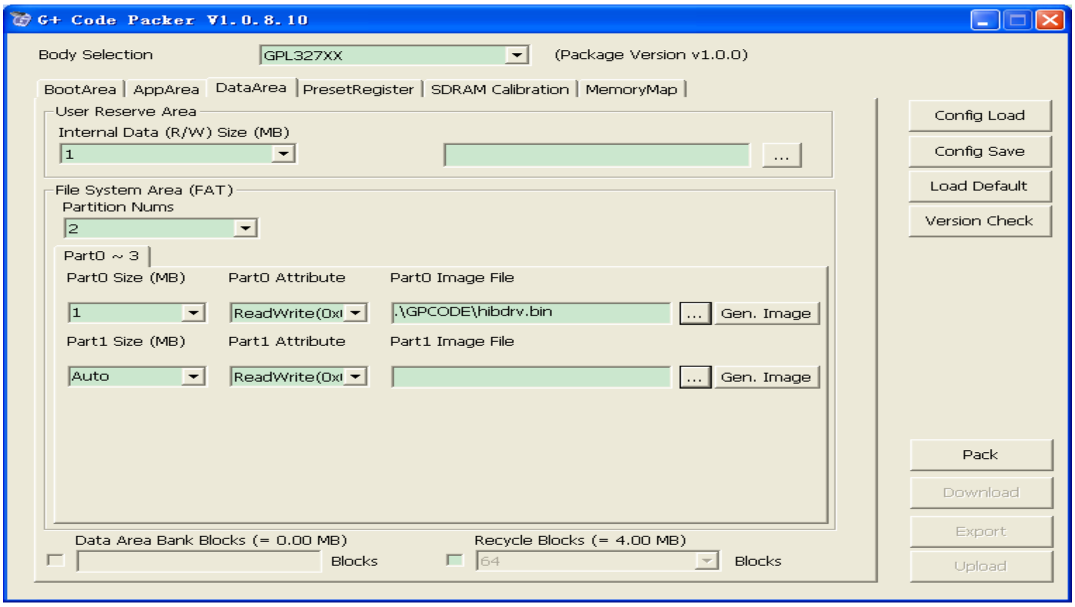
- App Partition Num: This item represents the number of current App sub-areas that user define, ranging from 1~16, meaning max sub-area is 16 areas.

- Partition N Size: N ranges 0~15, representing the capacity of current sub-area, unit in Sector. Generally, when we specify the sub-area’s imaging file, tool will automatically fill in the corresponding file size for us. Under one circumstance, user does not specify a file and it becomes AUTO mode. If user intend to assign the sub-area’s size, simply input the value into field.
- Partition N Attribute: Similarly, N ranges 0~15, representing the type of current imaging file. In present, Tool offers 7 types of imaging file: Runtime Code, Resourse Part, FastBoot Bin, Quick Image, Hibrate Image, Image Flag, Customize BtLdr. It is important, especially when First bootloader is moving the code from App area.
- Partition N Image file: This parameter allows user to select the sub-area’s imaging file on the disk.
- Partition N Destination Address: This parameter is to specify where the current sub-area’s file will be stored in SDRAM.

Based on this method, configure and set up all the sub-areas until all areas’ messages are configured completely.

12.3 Set up Data Area

After the message in App’s sub-area is set up completely, switch to the Data Area, see the following:



G+ Code Packer V1.0.8.10

Body Selection: GPL327XX (Package Version v1.0.0)

User Reserve Area

Internal Data (R/W) Size (MB): 1

File System Area (FAT)

Partition Nums: 2

Part0 ~ 3

Part0 Size (MB): 1

Part0 Attribute: ReadWrite(0x

Part0 Image File: .\GPCODE\hibdrv.bin

Part1 Size (MB): Auto

Part1 Attribute: ReadWrite(0x

Part1 Image File:

Data Area Bank Blocks (= 0.00 MB) Blocks

Recycle Blocks (= 4.00 MB) 64 Blocks

The data area is composed of user reservation area and file system area. In present, recycle block is not supported.

- User reservation area

This area is to reserve a storage area from data area for user. User is allow to access this area through the read sector/write sector from eMMC Driver.

- File system area:

This area is similar to the App sub-area. First, we define the expected logic sub-area and then set up each message for every logic sub-area. We now introduce the following options for logic sub-area:

PartN Size: N ranges from1 ~8, representing the capacity of current sub-area. When imaging file is not selected, default in Auto or user is allowed to specify the value.PartN

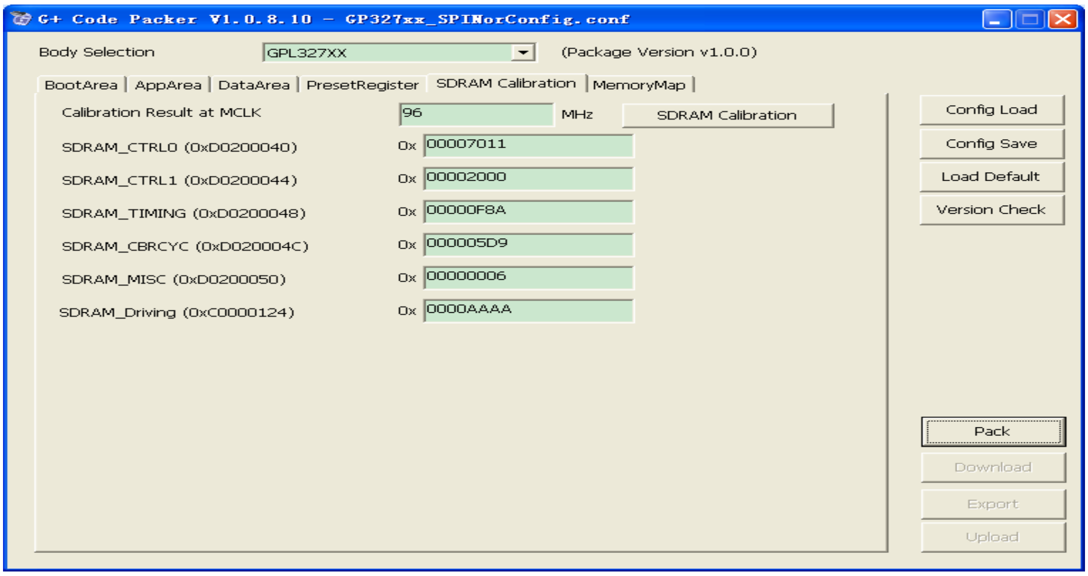
PartN Attribute: N ranges 1 ~8, representing the attribute of current sub-area. There are three options: read-only, write-only and read-write.

PartN Image File: N ranges from 1 ~ 8. This option is to specify the current sub-area's imaging file. There are two ways to specify this filed, clicking "…" button to open file dialog box and select file from local. The other, clicking "Gen.image" button to generate file via G+FatImage Tool. For more information about G+FatImage Tool, please refer to its User's Guide.

In general, although user's reservation area and data sub-area are all data area, their purposes are different. The user's reservation area is to operate this area using the read/write function from SD card driving procedure. The data sub-area, however, is to operate various logic sub-area via file system

12.4 Set up Pre-Register & Dram Calibration

In general, we first run the DRAM calibration to GP327xx and then switch back to Pre-Register to check the corresponding register's content. Before entering the code programming process, make sure DRAM calibration has been completed. Switch to Dram Calibration tag, DRAM Calibration's UI is as follows:



G+ Code Packer V1.0.8.10 – GP327xx_SPINorConfig.conf

Body Selection GPL327XX (Package Version v1.0.0)

Calibration Result at MCLK 96 MHz

SDRAM_CTRL0 (0xD0200040) 0x 00007011

SDRAM_CTRL1 (0xD0200044) 0x 00002000

SDRAM_TIMING (0xD0200048) 0x 00000F8A

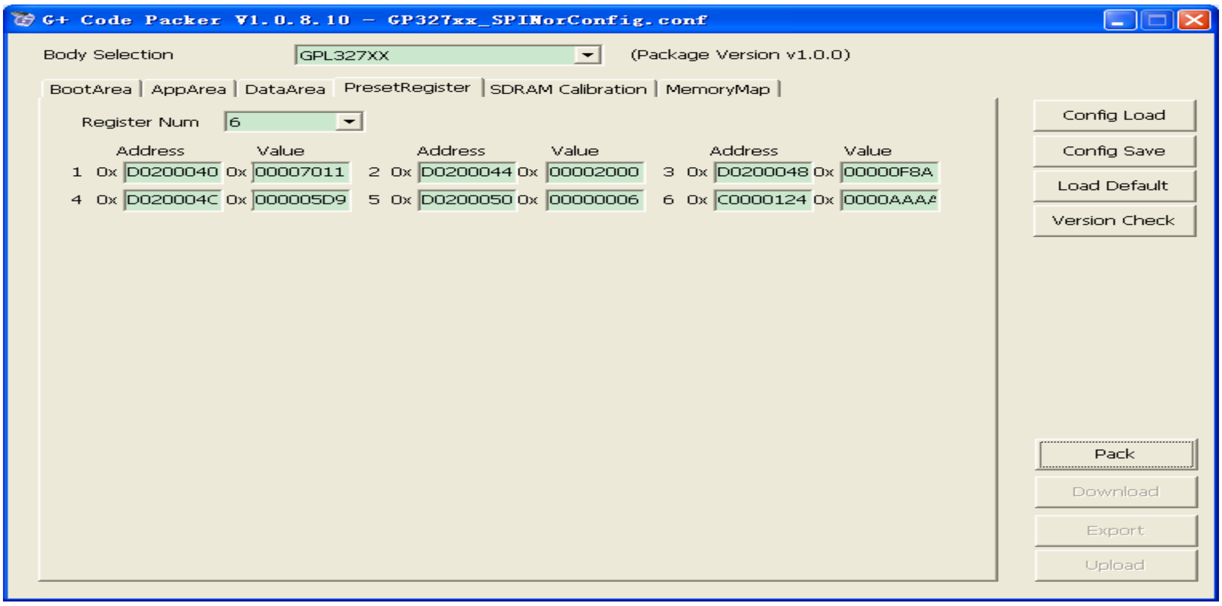
SDRAM_CBRCYC (0xD020004C) 0x 000005D9

SDRAM_MISC (0xD0200050) 0x 00000006

SDRAM_Driving (0xC0000124) 0x 0000AAAA

Click "SDRAM calibration button" to activate G+ SDRAM calibration tool. Calibrate SDRAM register via this tool. For more information, please refer to G+ SDRAM Calibration Tool User's Guide. After

SDRAM calibration is completed, switch it to the Pre-register tag and we will see the values in calibration tool has been updated in the Pre-register window. It looks like:



G+ Code Packer V1.0.8.10 – GP327xx_SPINorConfig.conf

Body Selection: GPL327XX (Package Version v1.0.0)

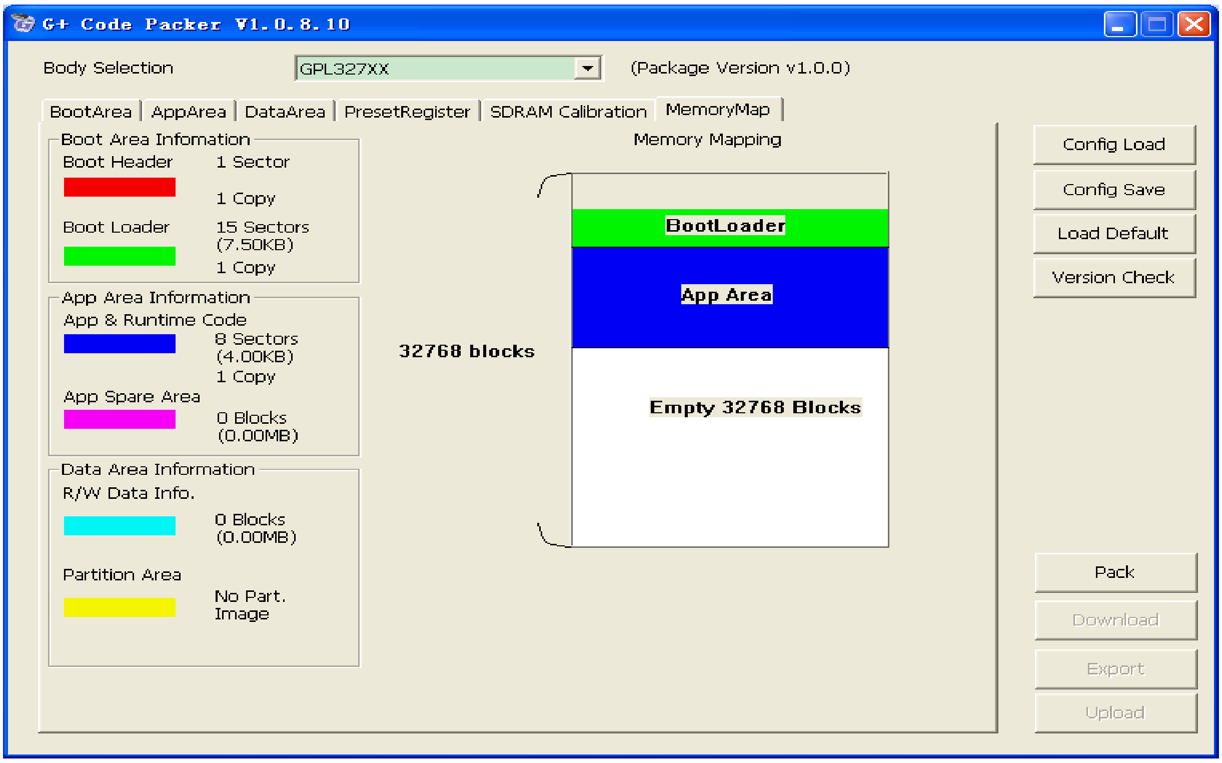
Register Num: 6

	Address	Value		Address	Value		Address	Value
1	0x D0200040	0x 00007011	2	0x D0200044	0x 00002000	3	0x D0200048	0x 00000F8A
4	0x D020004C	0x 000005D9	5	0x D0200050	0x 00000006	6	0x C0000124	0x 0000AAA

From the window above, it is composed of two parts: number of registers and register's address and the content in address. For some reasons, DRAM calibration may not be effective as we expect in some ways or some configuration adjustments for booting system (e.g. clock or IO port status). In such cases, we are allowed to modify the settings in this window manually.

12.5 Memory Map

After configuring Boot Area, App Area and Data Area, switch to this page and we will see the eMMC layout as follows:



G+ Code Packer V1.0.8.10

Body Selection: GPL327XX (Package Version v1.0.0)

BootArea | AppArea | DataArea | PresetRegister | SDRAM Calibration | MemoryMap

Boot Area Infomation
Boot Header 🟥 1 Sector, 1 Copy
Boot Loader 🟩 15 Sectors (7.50KB), 1 Copy

App Area Information
App & Runtime Code 🟦 8 Sectors (4.00KB), 1 Copy
App Spare Area 🟪 0 Blocks (0.00MB)

Data Area Information
R/W Data Info. 0 Blocks (0.00MB)
Partition Area 🟨 No Part. Image

Note that the figures in this window may not be accurate. We will fix it in the later version. This will not affect the actual layout in the physical register.

12.6 The Button Function

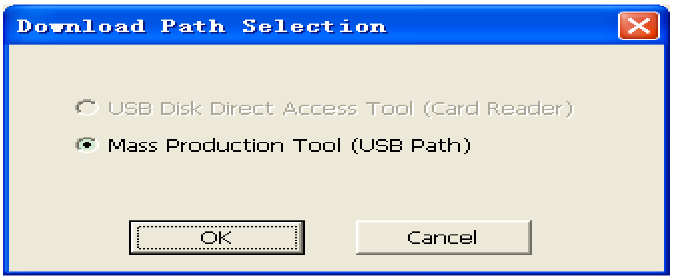
After all settings are completed, click “Config Save” to save the settings. We can re-load the previous settings by using “Config Load”. Note that the new version of Code Packer Tool should not be used to load the old settings saved by older version of tool because various versions of tools may result in compatibility issue.

Load Default: This button allows user to clear all present settings to the tool’s default settings without closing Code Packer.

Version Check: this button checks the version of hdb file.

Pack: This button packs certain messages to a file for programming based on the rule given.

Download： Call G+ Mass Production Tool, see the following window:



Download Path Selection

○ USB Disk Direct Access Tool (Card Reader)
● Mass Production Tool (USB Path)

This is to insert eMMC card onto the EVB's eMMC socket which is allowed to be written and read via GP327xx/GPDV6xx.

Export: This button outputs the packed file and program onto a disk. Simply openG+ Mass Production tool to program files. It is especially for mass production purpose.

13Mass Production Guideline (USB Mass Tool Portable)

13.1 USB Mass Production Tool Export

To transfer the BIN file to manufacture for mass production, G+ CodePacker features the “Export” function to make this easily. Note that GPL326XX only supports NAND Flash programming and GPL329XX supports NAND Flash/SDCard/eMMC programming. Steps are as follows:

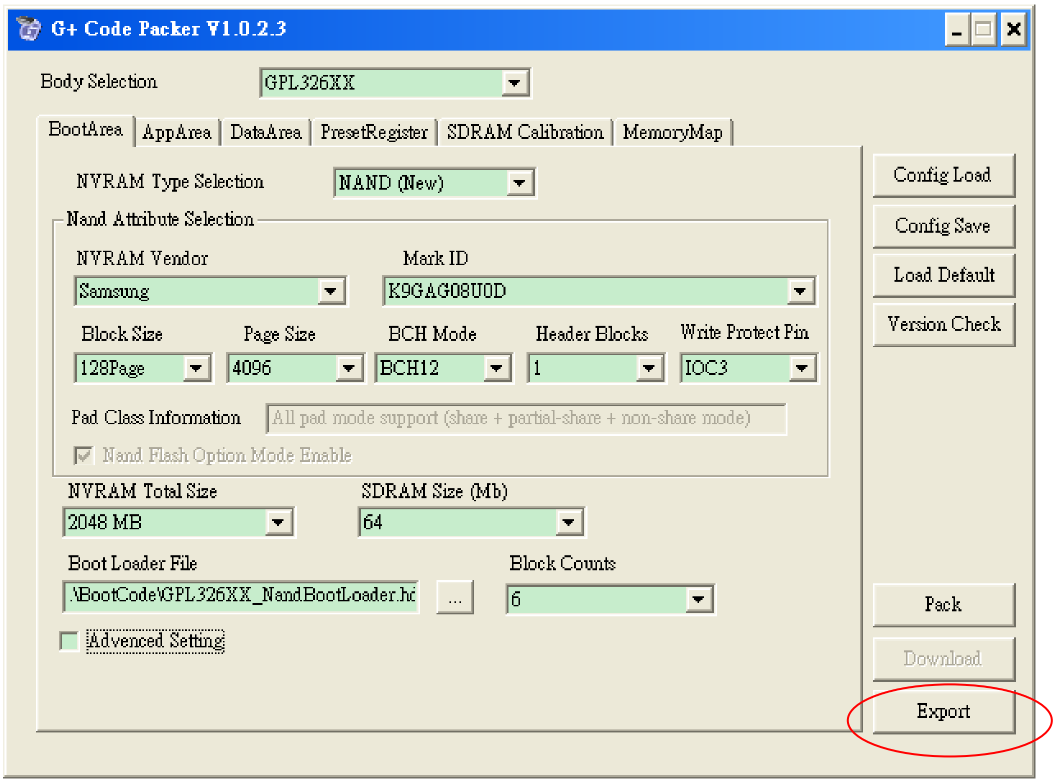
Step 1：Pack

After Pack is completed, click the download button. Program the BIN file onto device via Mass production Tool and test if BIN file is correct and test if the system can work well. After download is executed, Export button is triggered.

Step 2：Export

Click Export button and select the export directory and click OK to start exporting. Export will generate BIN file and all necessary files under the selected directory. It can be easily transferred (portable) to manufacture for mass production.

Note: Run the MassProduction.bat, under the export directory, to call Mass Tool for mass production procedure.



G+ Code Packer V1.0.2.3

Body Selection: GPL326XX

NVRAM Type Selection: NAND (New)

Nand Attribute Selection

NVRAM Vendor: Samsung

Mark ID: K9GAG08U0D

Block Size: 128Page

Page Size: 4096

BCH Mode: BCH12

Header Blocks: 1

Write Protect Pin: IOC3

Pad Class Information: All pad mode support (share + partial-share + non-share mode)

☑ Nand Flash Option Mode Enable

NVRAM Total Size: 2048 MB

SDRAM Size (Mb): 64

Boot Loader File: .\BootCode\GPL326XX_NandBootLoader.hd

Block Counts: 6

☐ Advanced Setting

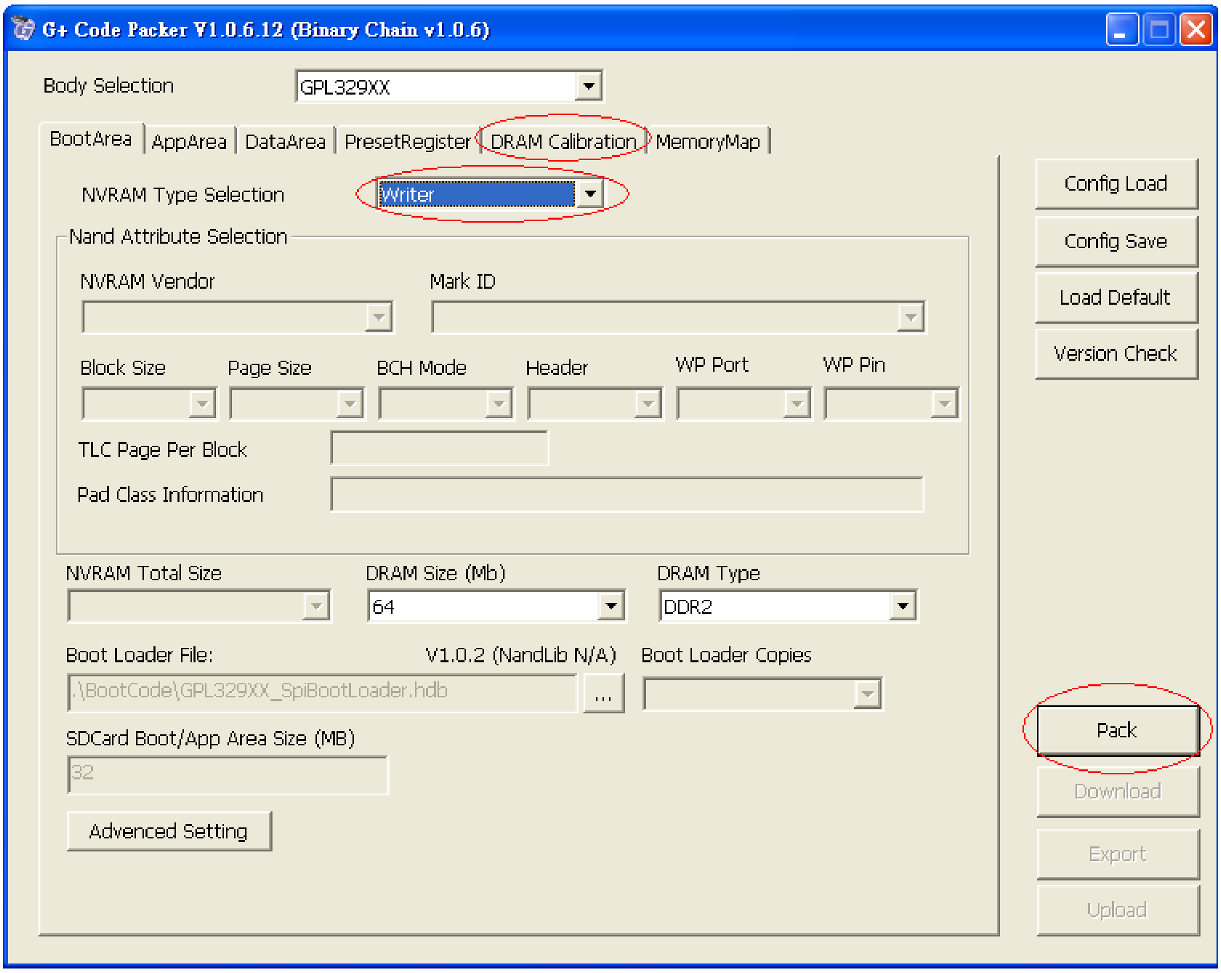
13.2 Make a SDC Production Card

We can make various types of NVRAM through SDC Production. Now, GPL326XX only supports NAND Flash. GPL329XX supports all NAND Flash/ SDCard/ eMMC.

Step1. Writer’s DRAM Calibration

Writer’s parameters for production (e.g. DRAM parameter) should be calibrated to assure proper operation as well as proceed the NVRAM production. Currently, only GPL329XX provides this function. To calibrate writer’s DRAM, first select writer from NVRAM type and switch the window to DRARM Calibration page for DRAM parameter calibration. After it is done, press “Pack” to save the DRAM Calibration parameters which will later be integrated while making SDC production card.

Note: For GPL326XX body, you may ignore this step.



G+ Code Packer V1.0.6.12 (Binary Chain v1.0.6)

Body Selection: GPL329XX

NVRAM Type Selection: Writer

DRAM Size (Mb): 64

DRAM Type: DDR2

Boot Loader File: .\BootCode\GPL329XX_SpiBootLoader.hdb

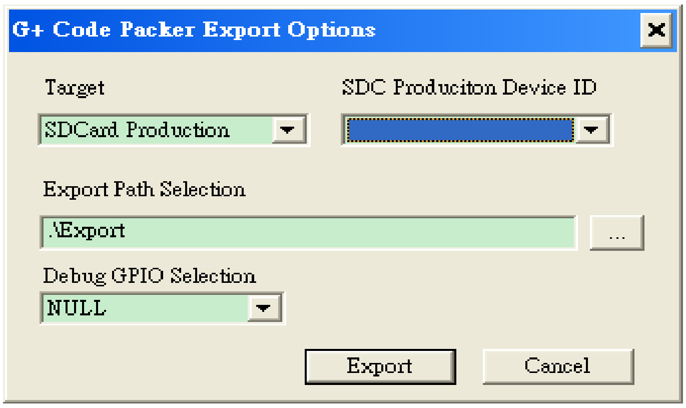
V1.0.2 (NandLib N/A)

SDCard Boot/App Area Size (MB): 32

Step2. Enter Export mode

Push Export Button and select Target to SDCard Production. This function is only available for the following 2 conditions: (only GPL326XX is available and NVRAM Type must be NAND Flash).

1) GPL326XX is available and NVRAM Type must be NAND Flash
2) GPL329XX is available and NVRAM Type must be NAND Flash/SDCard/eMMC.



G+ Code Packer Export Options

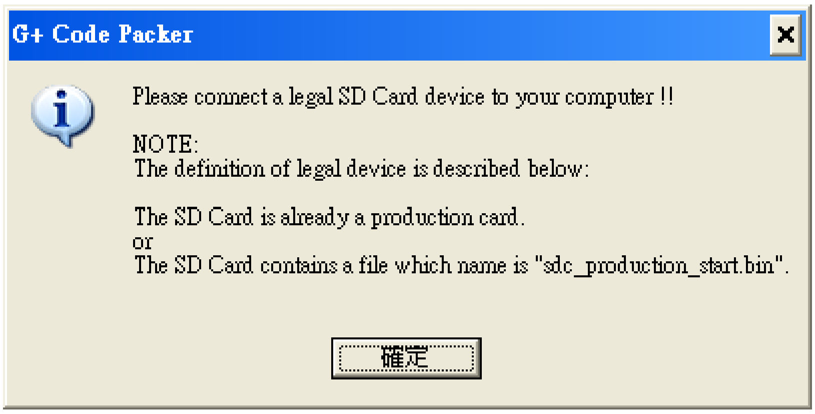
Target: SDCard Production

Export Path Selection: .\Export

Debug GPIO Selection: NULL

Step3: Guideline

When the tool programs the binary file onto SD card to make a SD production card, the original data on the SD card may be destroyed. Therefore, the SD card should be backed up before enter the mass production flow. The generating tool for making production card require card itself must be a production card (pre-defined by the tool). Or it is a FAT card with a SDC_Production_Start.bin (empty file is ok) existing in its root path. Only these two types of cards are supported to make a production card by the tool. We highly recommend using the single function card reader by Single Lung (only SD socket available)



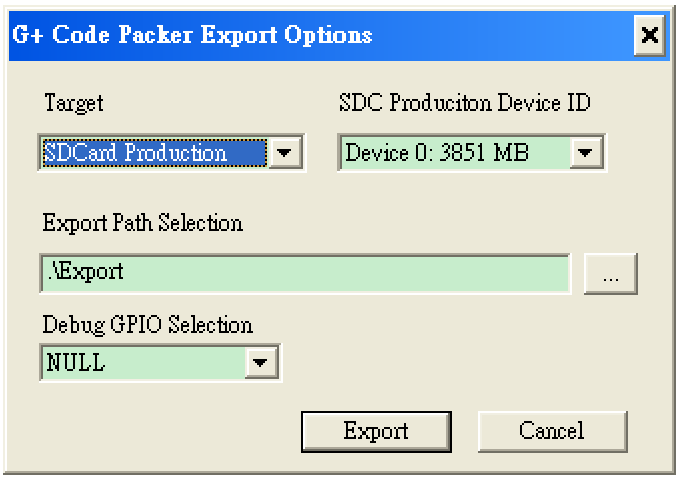
G+ Code Packer

Please connect a legal SD Card device to your computer !!

NOTE:
The definition of legal device is described below:

The SD Card is already a production card.
or
The SD Card contains a file which name is "sdc_production_start.bin".

Step4: Production card or semi-production card is detected



G+ Code Packer Export Options

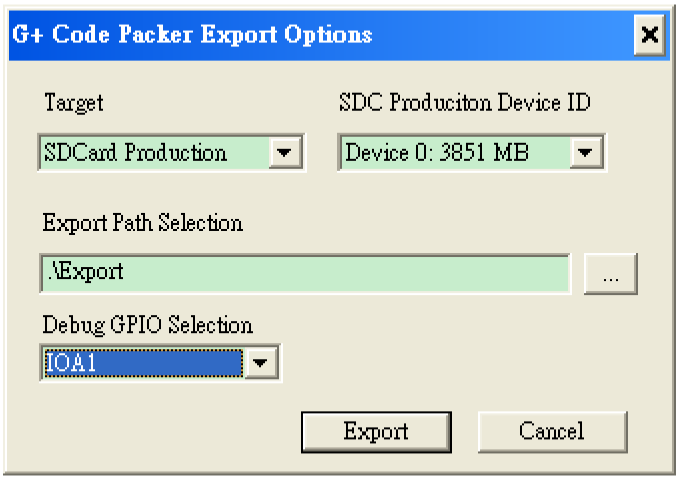
Target: SDCard Production

SDC Produciton Device ID: Device 0: 3851 MB

Export Path Selection: .\Export

Debug GPIO Selection: NULL

Step5: Selecting Debug GPIO enables the flash indicator for production progress and consequence acknowledgement



G+ Code Packer Export Options

Target: SDCard Production

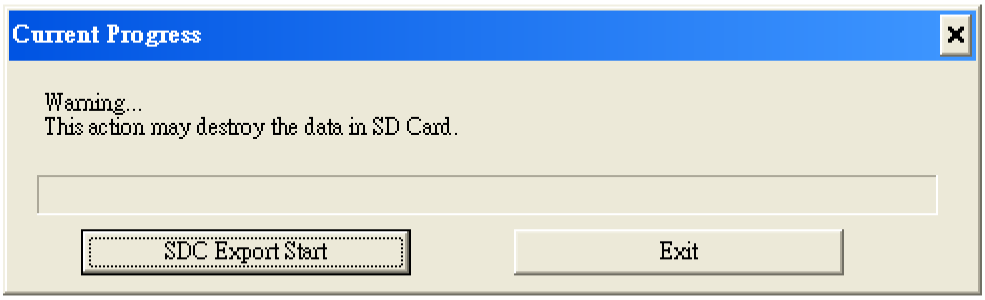
SDC Produciton Device ID: Device 0: 3851 MB

Export Path Selection: .\Export

Debug GPIO Selection: IOA1

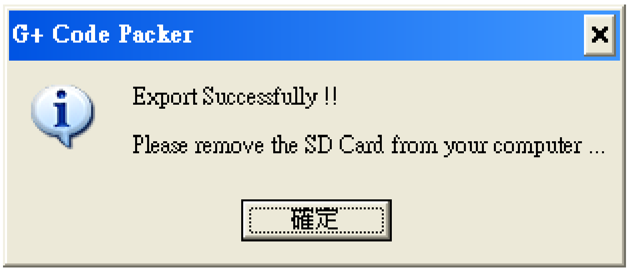
Step6: Start making SDC production card

Users can make a lot of SDC production card at this step. Don't need to repeat step 1 ~ 4. Remove the old card and insert new card, click SDC Export Start again.



Current Progress

Warning...
This action may destroy the data in SD Card.



G+ Code Packer

Export Successfully !!

Please remove the SD Card from your computer ...

14 Error Message

It is possible that users may incorrectly operate the tool, forget to define some settings or ignore the relations between options (e.g. code size is greater than the actual area size). If these occur, target file may be incorrectly generated or programmed. To prevent users away from these mistakes, the tool will show some error messages to remind users and assist users to make a correct adjustment. The following table is a list of error messages that tool may show.

Error message	Description
Open selected file failed	Fail to open resource file
Open RunTime ROM Code File Failed	Fail to open RunTime ROM code
Please input your Runtime ROM Code Size	No size is entered when RunTime ROM Code size is checked off.
Runtime Code + Resource file size is greater than APP Area Size	The size user select for APP Area is greater than the defined App Area Size.
The selected “APP Main Size” is too small please modify the setting of “APP Main size	Number of copy is too large. The entire App area exceeds the defined App Area Size.
The selected size is greater than remained.	Intend to change App Main size and App Main size is selected too small.
The selected “spare rate” is invalid.	Incorrect spare rate (may exceed the max. value)
Config File Not Found	Configuration file not found (PackConfig may not exist)
Please select File First	Intend to operate the resource files, but no file is selected.
The element can’t be moved down.	Intend to move down the last resource file.
The element can’t be moved up.	Intend to move up the first resource file.
BootLoader File size is greater than Boot Block Count	The size of selected BootLoader file is greater than the selected Boot Block Count
Open R/W image file failed	Fail to open R/W image
The size of selected R/W image file is greater than R/W area size	The size of selected R/W image file is greater than R/W Area Size.
Can’t open the image file	Unable to open the partition image file.
Total size is greater than selected NVRAM total size	Total size is greater than the selected NVRAM total size (App area + Boot area + Data area)
Select NVRAM Total Size First	NVRAM Total Size is not selected (unable to calculate)
Create Dialog Failed! The Application will be closed	Fail to create main window (may open too many windows or memory not enough)
Download tool is in active	Mass production tool (called by G+Code packer) still running
Batch Pack only supports NAND Flash	“Batch Pack” only supports NAND Flash
Load Packing DLL Fail	Fail to load DLL (no BootAreaPacker.dll or AppCodePacker.dll found)
Load BootAreaPacker.dll failed	Fail to load BootAreaPacker.dll

Error message	Description
Input Address is invalid	Input address does not exist (illegal hex-decimal symbol)
The max. number of digits is 8.	Max. number of digits is 8.
Open selected path fail! Please select another path	Fail to open the selected directory (directory not exist or create failure)
All options must be selected	One or some of options are not selected (defined)
Save User Options Failed	Fail to save current settings (UserOption.ini may not exist)
Write Options to Pack Config Failed	Fail to write options to pack Configure file (PackConfig may not exist)
The bootloader file is not a supported format!	Version of default boot loader is incorrect or other file information incorrect.
Bank Size + Recycle size should be smaller than page size!	The sum of selected Bank Size and Recycle size must be smaller than the Page size.
Please connect a legal SD Card device to your computer! Note: The definition of legal device is described below: The SD card is already a production card. Or The SD contains a file which name is “sdc_production_start.bin”.	Under running SDC production, a legal SD card device must be used. Legal device means: This SD card is a production card. Or This SD card contains “sdc_production_start.bin”.
Read version config file failed!	Fail to read the configure file (.\PackIni\DefaultBinary.ini)