



G+ Code Packer User Guide

V1.0.8 – Oct. 29, 2012

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.

目次

G+ CODE PACKER USER GUIDE	1
1 G+ CODE PACKER 介紹	7
1.1 工具簡介.....	7
1.2 IC BODY支持.....	7
2 安裝說明	8
2.1 安裝與更新工具.....	8
3 G+ CODE PACKER 視窗說明	9
3.1 主畫面說明.....	9
3.2 各區域畫面及說明.....	10
3.2.1 Boot Area.....	10
3.2.2 App Area.....	11
3.2.3 Data Area.....	12
3.2.4 Pre-set Register & SDRAM Calibration.....	13
3.2.5 Memory Map.....	18
3.2.6 Configuration Tool Buttons.....	19
3.2.7 Action Buttons.....	19
3.2.8 Version Check Button.....	20
3.2.9 Check for Update(s).....	21
4 NAND FLASH BOOT操作流程解說(GPL325XX/GPL326XX)	23
4.1 LOAD CONFIG & CONFIG SAVE.....	24
4.2 設置BOOT AREA.....	24
4.3 設置APP AREA.....	26
4.4 設置DATA AREA及FAT IMAGE TOOL之使用方法.....	29
4.4.1 設置Data Area.....	29
4.4.2 FAT Image Tool之使用方法.....	31
4.5 設置PRE-SET REGISTER及SDRAM CALIBRATION之使用方法.....	35
4.5.1 設置Pre-set Register.....	35
4.5.2 SDRAM Calibration的使用.....	35
4.6 PACK與DOWNLOAD.....	40
4.7 EXPORT MP TOOL TO OTHER DISK (EX: USB DISK FOR FACTORY).....	41
4.8 UPLOAD.....	42

5	NAND FLASH BOOT操作流程解說 (GPL329XX)	43
5.1	LOAD CONFIG & CONFIG SAVE	44
5.2	設置BOOT AREA	44
5.3	設置APP AREA	46
5.4	設置DATA AREA及FAT IMAGE TOOL之使用方法	49
5.4.1	設置Data Area	49
5.4.2	FAT Image Tool之使用方法	51
5.5	設置PRE-SET REGISTER及DRAM CALIBRATION之使用方法	55
5.5.1	設置Pre-set Register	55
5.5.2	DRAM Calibration的使用	55
5.6	PACK與DOWNLOAD	58
6	SPI FLASH BOOT操作流程解說	59
6.1	設置BOOT AREA	59
6.2	設置APP AREA	61
6.3	設置DATA AREA	62
7	SDCARD BOOT操作流程解說 (GPL326XX)	64
8	SDCARD BOOT操作流程解說 (GPL329XX)	65
8.1	LOAD CONFIG & CONFIG SAVE	66
8.2	設置BOOT AREA	66
8.3	設置APP AREA	67
8.4	設置DATA AREA及FAT IMAGE TOOL之使用方法	67
8.5	設置PRE-SET REGISTER及DRAM CALIBRATION之使用方法	67
8.6	PACK與DOWNLOAD	67
9	EMMC BOOT操作流程解說 (GPL329XX)	70
9.1	LOAD CONFIG & CONFIG SAVE	71
9.2	設置BOOT AREA	71
9.3	設置APP AREA	72
9.4	設置DATA AREA及FAT IMAGE TOOL之使用方法	72
9.5	設置PRE-SET REGISTER及DRAM CALIBRATION之使用方法	72
9.6	PACK與DOWNLOAD	72
10	SPI NOR FLASH操作流程解說 (GPL327XX/GPDV6XXX)	74
10.1	BOOT AREA配置	74
10.1.1	Boot Area基本配置	74

10.1.2	Boot Area高级设置	75
10.2	APP AREA配置	76
10.3	DATA AREA配置	77
10.4	PRE-REGISTER & DRAM CALIBRATION设置	78
10.5	MEMORY MAP	79
10.6	按钮功能	80
11	SD CARD操作流程解说 (GPL327XX/GPDV6XXX)	81
11.1	BOOT AREA配置	81
11.1.1	Boot Area基本配置	81
11.1.2	Boot Area高级设置	82
11.2	APP AREA配置	83
11.3	DATA AREA配置	84
11.4	PRE-REGISTER & DRAM CALIBRATION设置	86
11.5	MEMORY MAP	87
11.6	按钮功能	87
12	EMMC操作流程解说 (GPL327XX/GPDV6XXX)	89
12.1	BOOT AREA配置	89
12.1.1	Boot Area基本配置	89
12.1.2	Boot Area高级设置	90
12.2	APP AREA配置	91
12.3	DATA AREA配置	92
12.4	PRE-REGISTER & DRAM CALIBRATION设置	94
12.5	MEMORY MAP	95
12.6	按钮功能	95
13	量產程序 (USB MASS TOOL PORTABLE) 說明	97
13.1	USB MASS PRODUCTION TOOL EXPORT	97
13.2	SDC PRODUCTION 生產卡製作方式	98
14	錯誤訊息說明	102

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 G+ Code Packer 介紹

1.1 工具簡介

G+ Code Packer 是由 Generalplus 設計研發的一款資源檔案整合工具，此工具將 Boot Code、User Runtime Code 以及 Resource 等資源檔案整合成生產燒錄的 BIN 檔(ROM Image)。此工具支持的 NVRAM Type 包含了 SPI NOR Flash、Nand Flash 以及 SDCard。

G+ Code Packer 除了主要的資源檔案整合功能外，亦提供了其他功能，列舉如下：

1. 可進行 SDRAM Calibration，並利用 Calibration 結果設置 Pre-set Register。
2. 整合了 FAT Image Tool，可協助用戶製作和產生帶有檔案管理的 Data Image File。
3. 整合了 USB Mass Production Tool，可在完成資源檔案整合步驟後，進行燒錄動作，將產生的 BIN 檔燒錄到指定設備上。
4. 可將產生的 BIN 檔與 USB Mass Production Tool 進行 Export 動作，以利產線的量產動作。

1.2 IC Body 支持

本工具限使用於下列 IC body，請用戶留意可操作之 IC 版本限制。

GPL325XX-003A 版本 IC

GPL326XX 全系列版本 IC

GPL329XX 全系列版本 IC

GPL327XX 全系列版本 IC

GPDV6XXX 全系列版本 IC

2 安裝說明

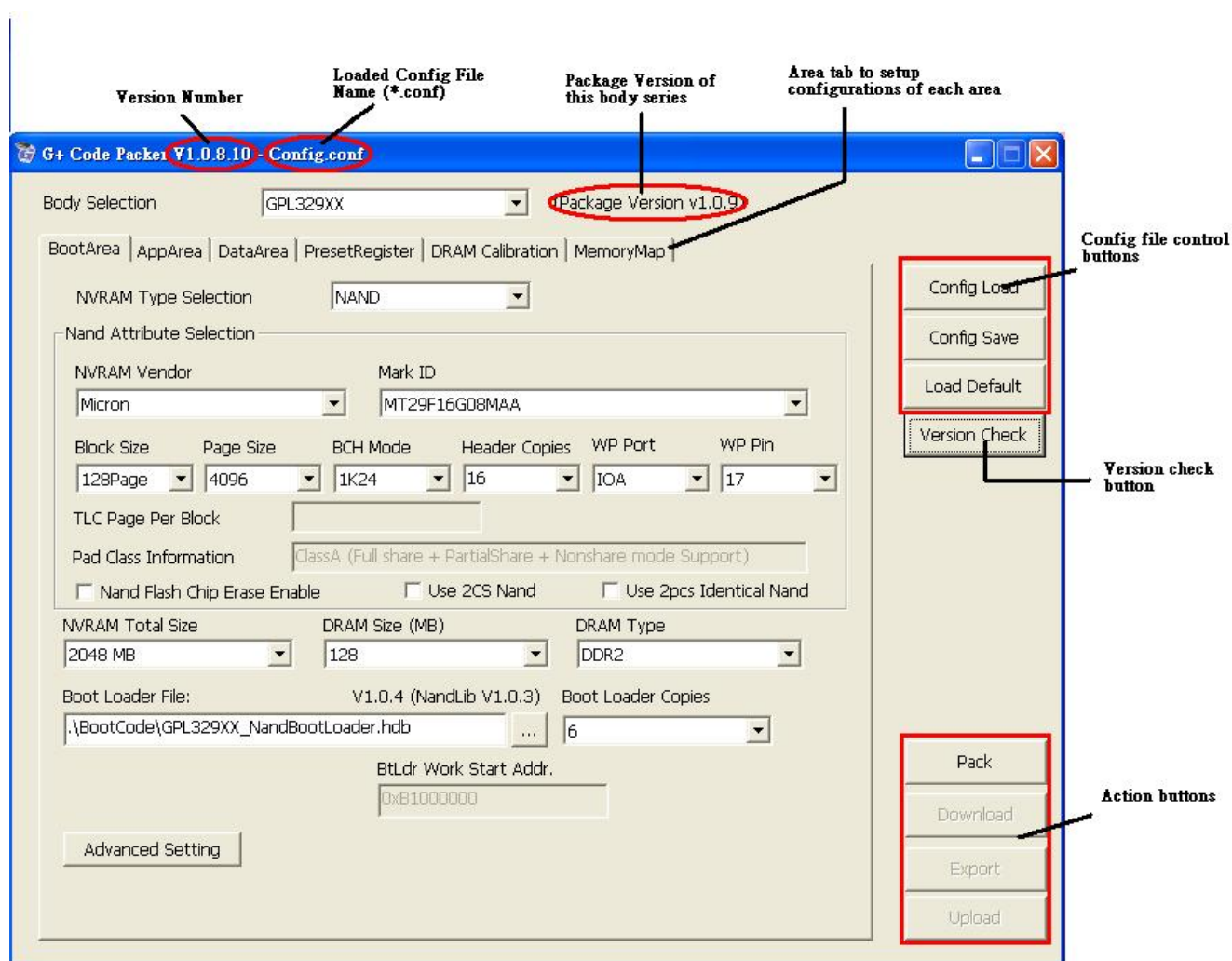
2.1 安裝與更新工具

1. G+ Code Packer 為一套免安裝的工具。安裝使用的方法僅需將取得的壓縮檔解壓縮至任一路徑即可。
2. 若要進行工具的版本升級，也只需將新的壓縮檔解壓覆蓋原本的解壓縮資料夾即可。

3 G+ Code Packer 視窗說明

3.1 主畫面說明

G+ Code Packer 的主要使用者介面與說明如下圖所示：



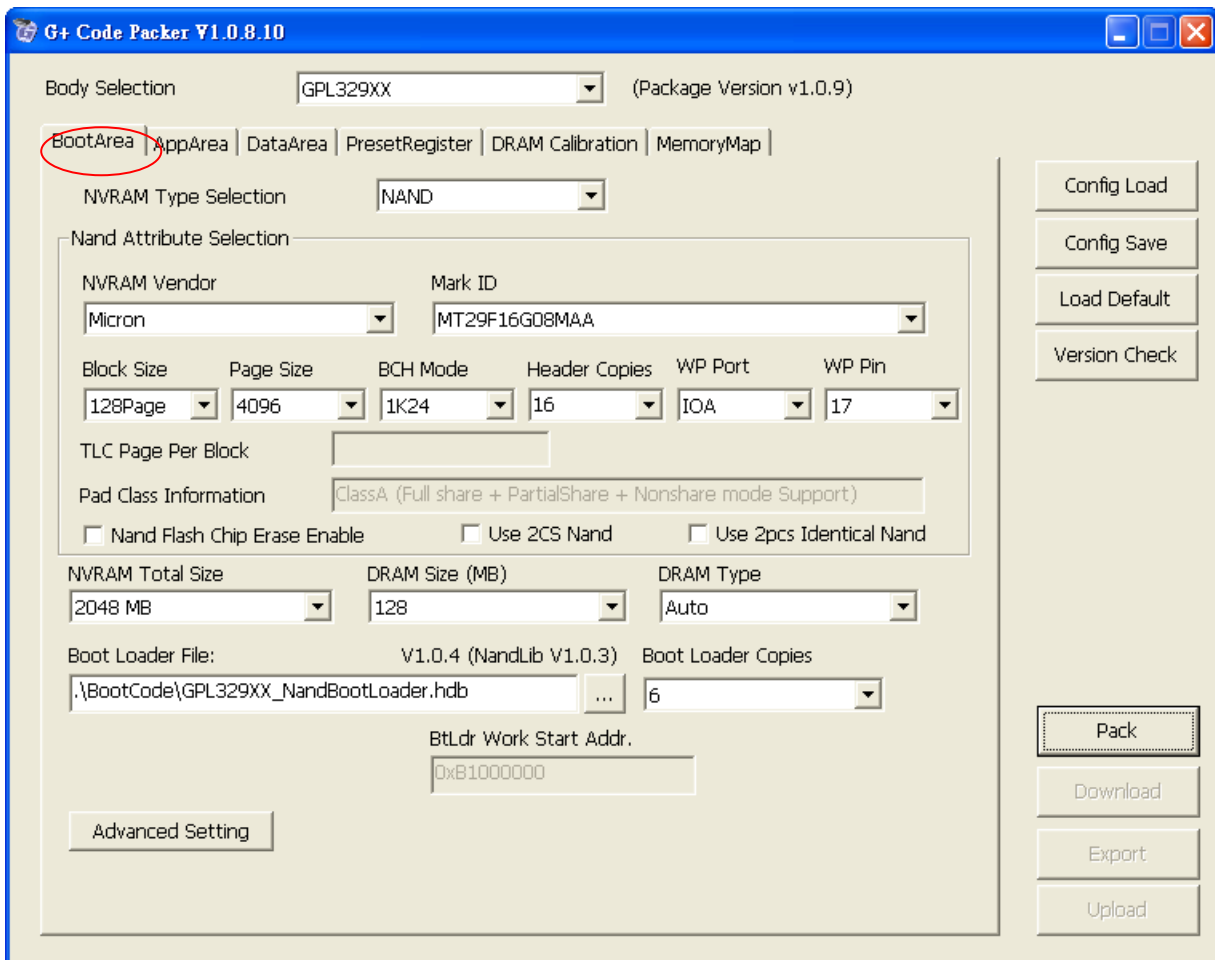
3.2 各區域畫面及說明

3.2.1 Boot Area

Internal ROM Boot (on chip) 後首先會執行到的區域，該區域之執行主檔為 Boot Loader File，為了管理方便，建議引用 Generalplus 所提供之預設 Boot Loader，以達最佳效能，若客戶有客制化需求，亦可改變 Boot Loader File 之路徑。

Boot Loader 除了會對 App Area 做初始化之外，還能夠將客戶提供之 Runtime Code 由 NVRAM App Area 中 Load 到 SDRAM 執行。

Boot Area 的使用者介面如下圖所示：

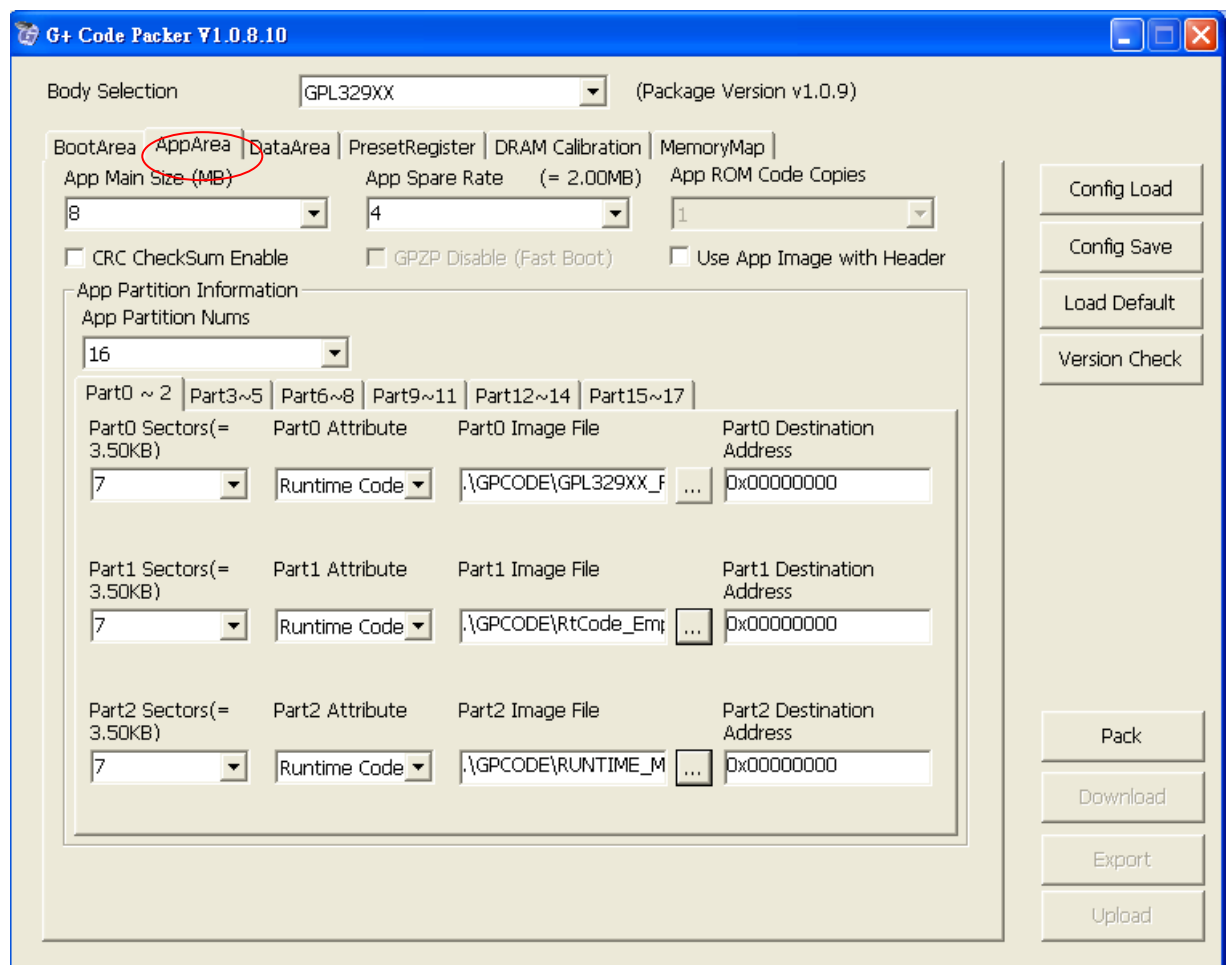


3.2.2 App Area

Generalplus Boot Loader 會將 App Area Initial 起來 (若非使用預載 BootLoader，則 App Area 之服務自動失效)。App Area 設計要點如下：

- App Area 為唯讀區。
- App Area 主要置放客戶之 Runtime Code，所謂 Runtime Code 即為 Runtime Process Image，可能是 DPF 主程式，亦可能為 Game、DV、ELA...等客戶開發出來之可執行主程式。

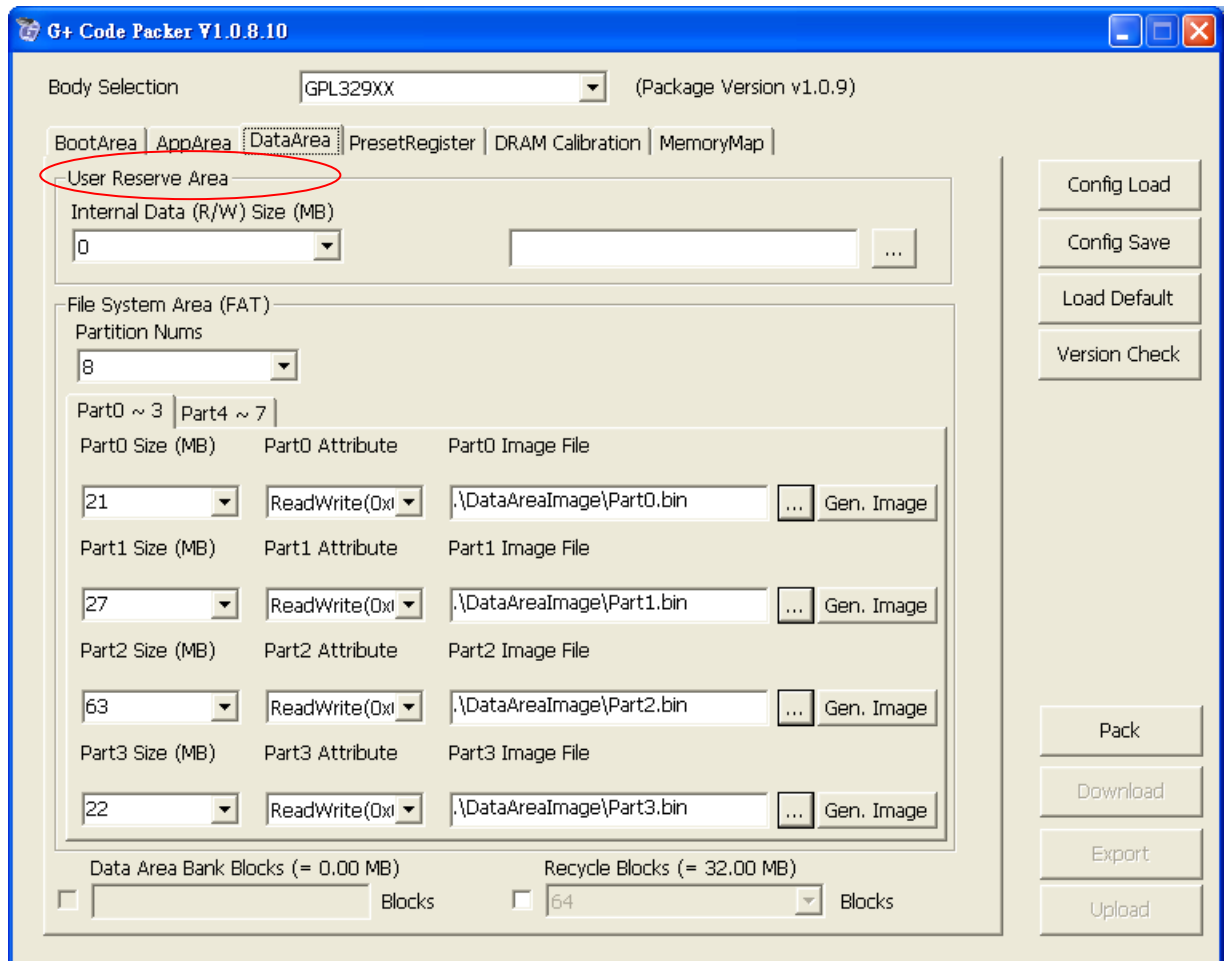
App Area 的使用者介面如下圖所示：



3.2.3 Data Area

為 NVRAM 扣除 Boot Area 與 App Area Size 之區域，此區若使用 General Plus 方案會有 R/W 區 (提供客戶經常性存取之區域，無特定格式) 之設置，與切分 File System Partition 用 (通常為 FAT16/32 或是 SPI FAT12 格式)。

Data Area 的使用者介面如下圖所示：

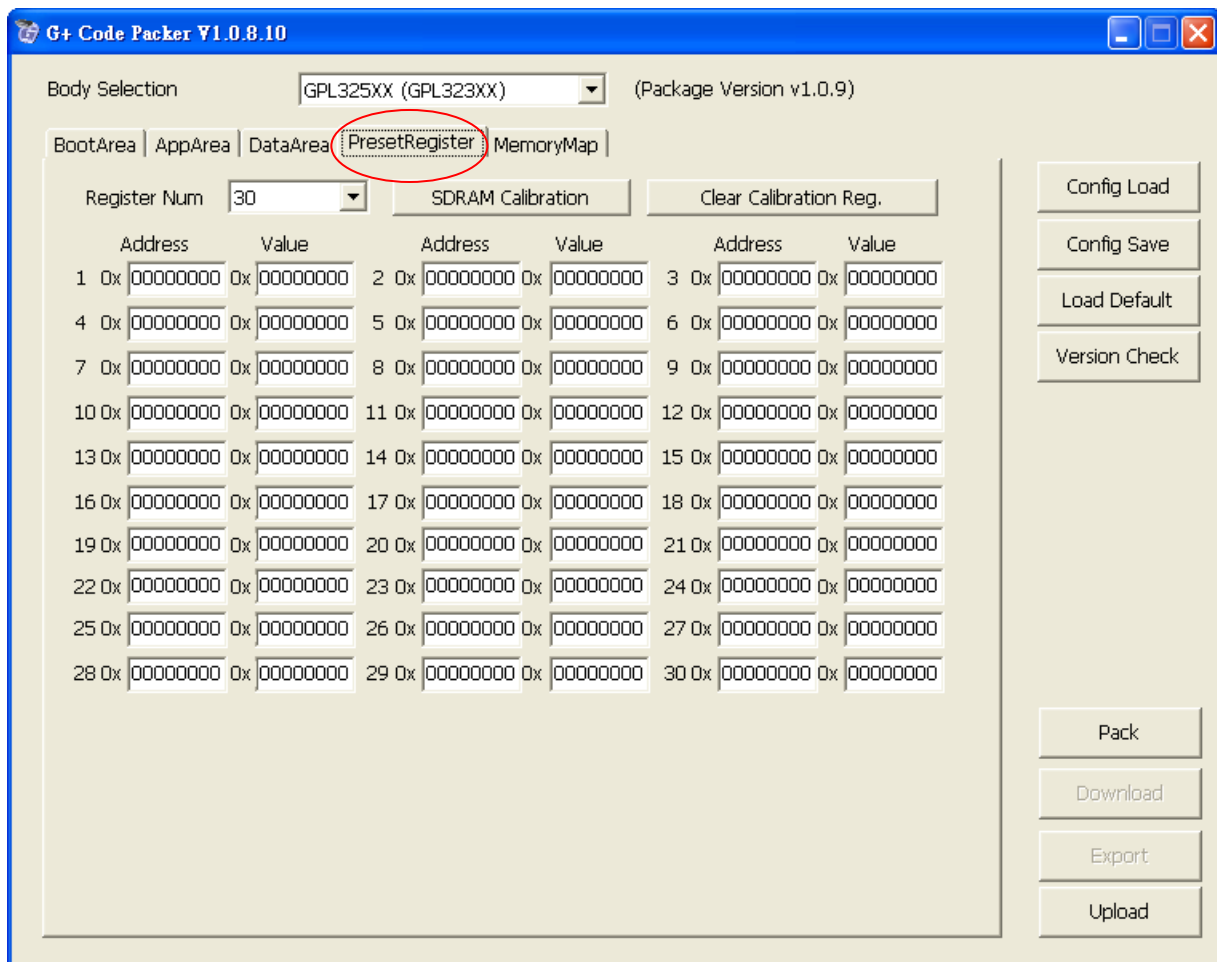


3.2.4 Pre-set Register & SDRAM Calibration

Internal MASK ROM code 跑起來之後可以設置相關 Register。每組參數包括：Register 位址和要寫入該 Register 的值。

Pre-set Register 的使用者介面根據選擇 Body 不同會有不同操作介面，不同 NVRAM type 會有不同可使用 Register 組數，如下圖所示：

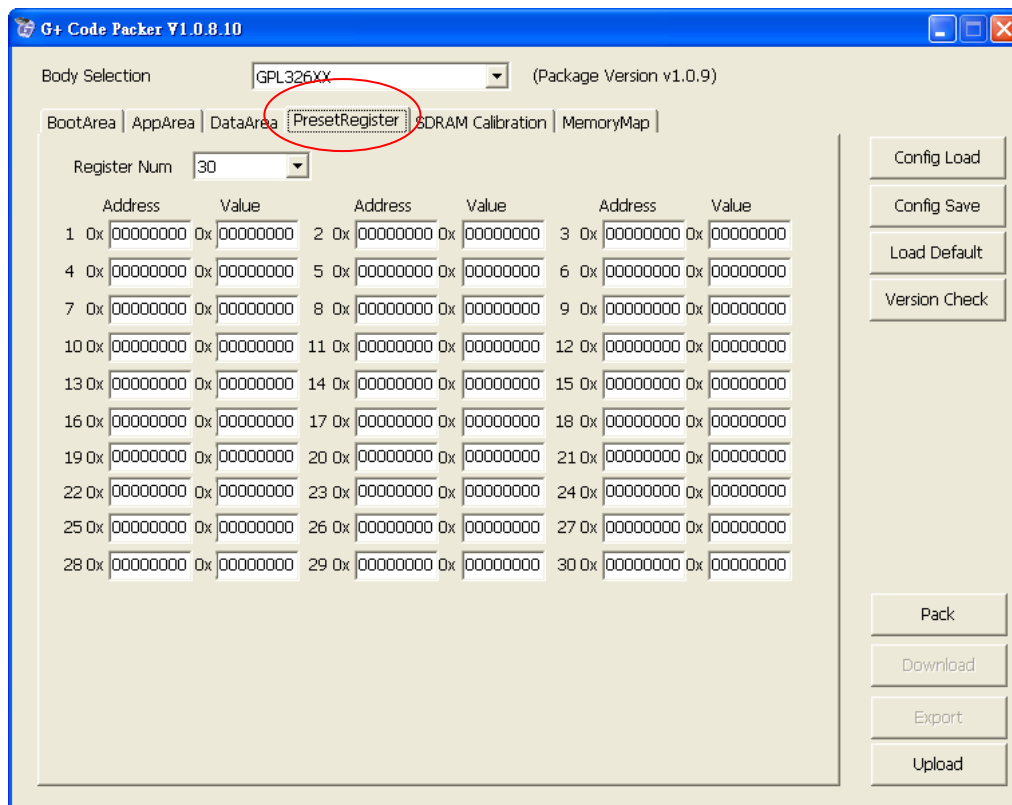
GPL325XX:



The screenshot shows the 'G+ Code Packer V1.0.8.10' window. The 'Body Selection' dropdown is set to 'GPL325XX (GPL323XX)'. The 'PresetRegister' tab is selected and highlighted with a red circle. The 'Register Num' is set to 30. The 'SDRAM Calibration' and 'Clear Calibration Reg.' buttons are visible. The main area displays a table of 30 registers, each with an 'Address' and 'Value' field. The values are all '0x00000000'. On the right side, there are buttons for 'Config Load', 'Config Save', 'Load Default', 'Version Check', 'Pack', 'Download', 'Export', and 'Upload'.

Register Num	Address	Value
1	0x00000000	0x00000000
2	0x00000000	0x00000000
3	0x00000000	0x00000000
4	0x00000000	0x00000000
5	0x00000000	0x00000000
6	0x00000000	0x00000000
7	0x00000000	0x00000000
8	0x00000000	0x00000000
9	0x00000000	0x00000000
10	0x00000000	0x00000000
11	0x00000000	0x00000000
12	0x00000000	0x00000000
13	0x00000000	0x00000000
14	0x00000000	0x00000000
15	0x00000000	0x00000000
16	0x00000000	0x00000000
17	0x00000000	0x00000000
18	0x00000000	0x00000000
19	0x00000000	0x00000000
20	0x00000000	0x00000000
21	0x00000000	0x00000000
22	0x00000000	0x00000000
23	0x00000000	0x00000000
24	0x00000000	0x00000000
25	0x00000000	0x00000000
26	0x00000000	0x00000000
27	0x00000000	0x00000000
28	0x00000000	0x00000000
29	0x00000000	0x00000000
30	0x00000000	0x00000000

GPL326XX：此頁面不包含 SDRAM Calibration 按鈕，此 Body 將 SDRAM 資訊獨立一頁顯示



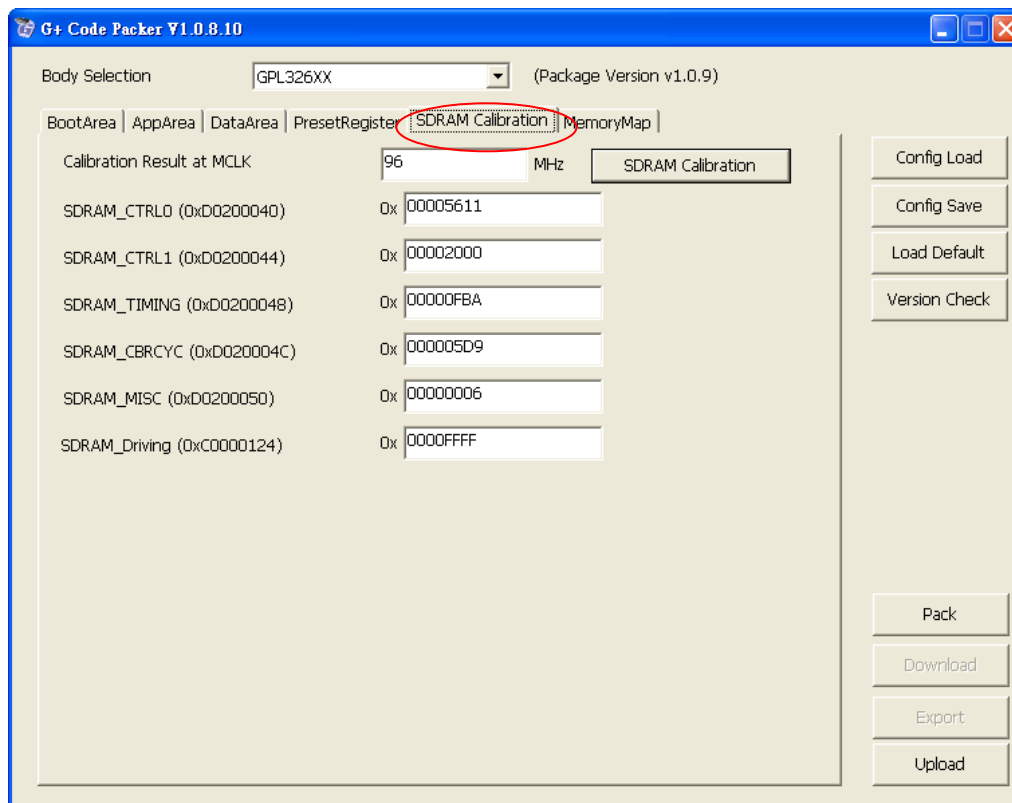
Body Selection: GPL326XX (Package Version v1.0.9)

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

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

Buttons: Config Load, Config Save, Load Default, Version Check, Pack, Download, Export, Upload



Body Selection: GPL326XX (Package Version v1.0.9)

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

Calibration Result at MCLK: 96 MHz

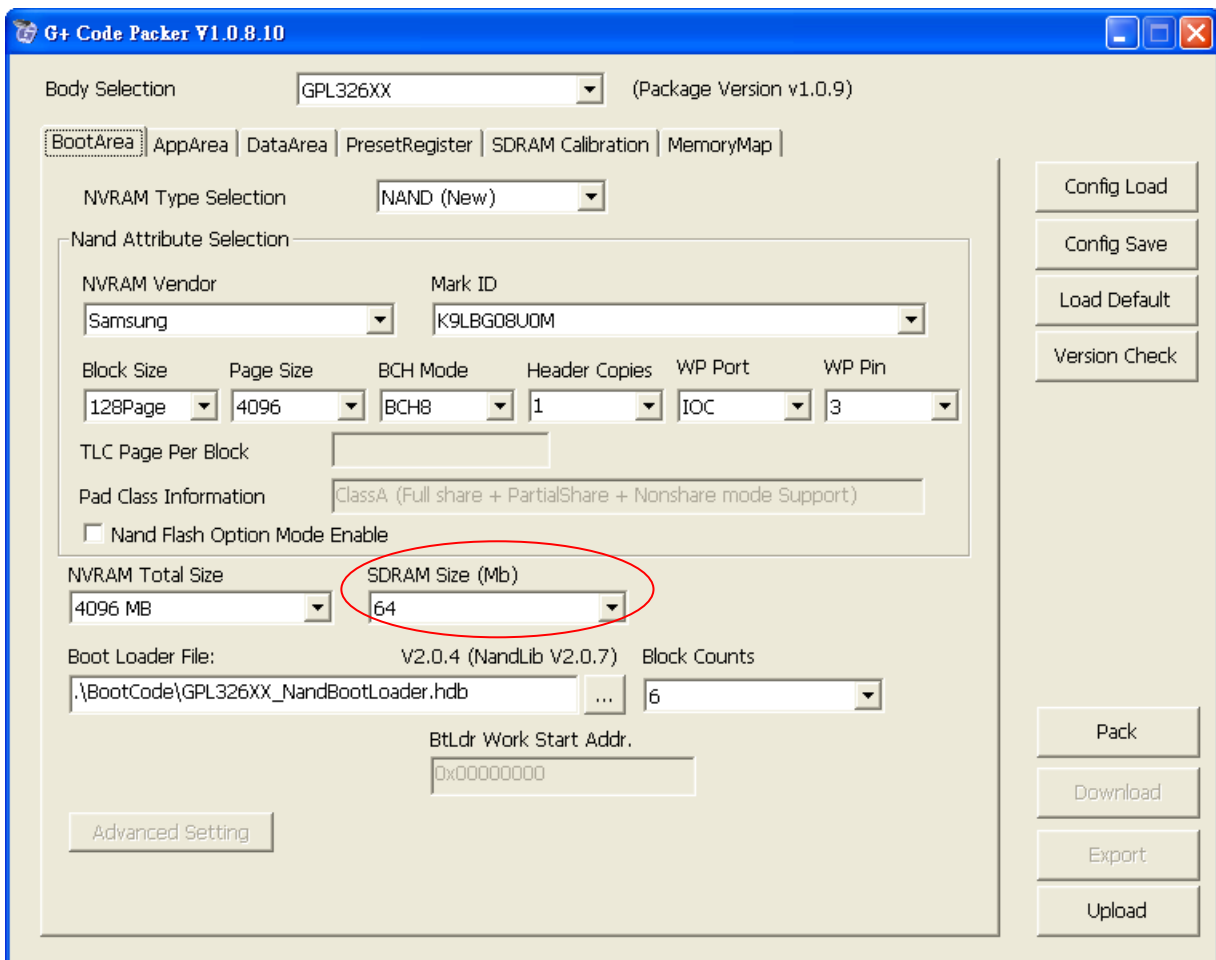
SDRAM Calibration

Parameter	Value
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

Buttons: Config Load, Config Save, Load Default, Version Check, Pack, Download, Export, Upload

為了使 SDRAM 在 App Code 跑起來之前穩定運行，General Plus 使用 SDRAM Calibration 來設置 SDRAM 的參數。點擊 SDRAM Calibration 調用 SDRAM Calibration Tool 對硬體上的 SDRAM 測試，將得到一組 SDRAM 參數，並自動帶入 Pre-set Register 設定中。

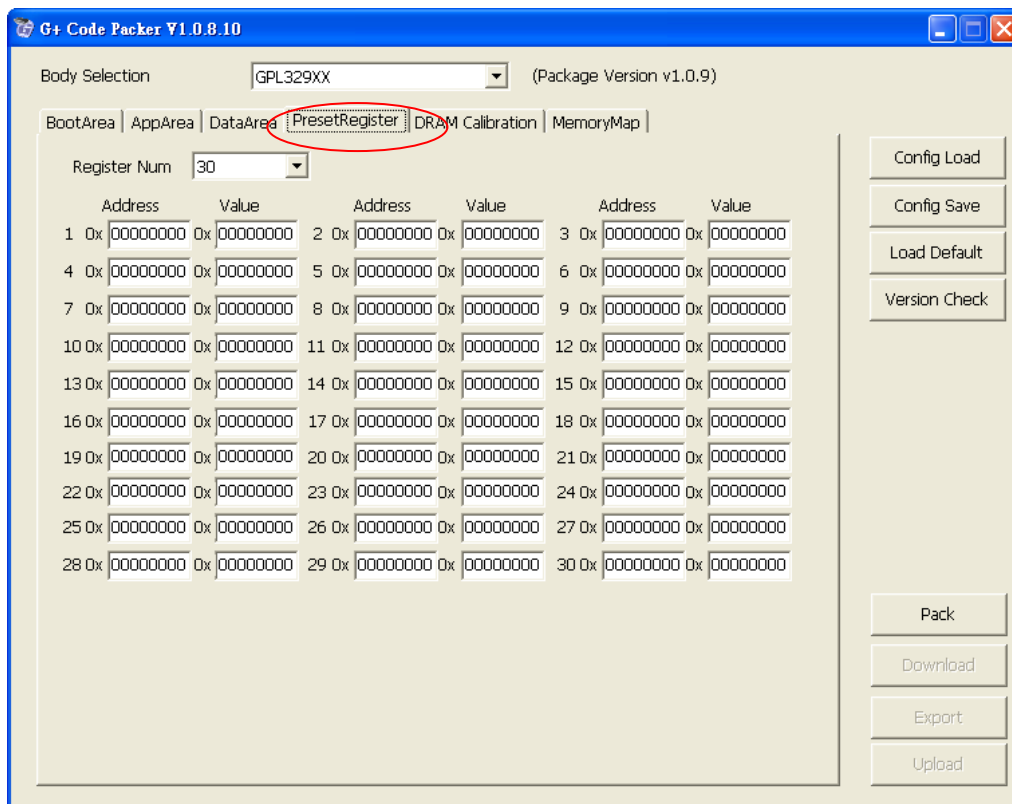
請注意在進行 **SDRAM Calibration** 前務必設定正確的 **SDRAM Size**，否則有可能造成 **Calibration** 結果失敗。



The screenshot shows the 'G+ Code Packer V1.0.8.10' window with the 'SDRAM Calibration' tab selected. The 'Body Selection' is 'GPL326XX' (Package Version v1.0.9). The 'BootArea' tab is active, showing various configuration options. The 'NAND (New)' option is selected for 'NVRAM Type Selection'. Under 'Nand Attribute Selection', 'Samsung' is the 'NVRAM Vendor' and 'K9LBG08U0M' is the 'Mark ID'. 'Block Size' is '128Page', 'Page Size' is '4096', 'BCH Mode' is 'BCH8', 'Header Copies' is '1', 'WP Port' is 'IOC', and 'WP Pin' is '3'. 'TLC Page Per Block' is empty. 'Pad Class Information' is 'ClassA (Full share + PartialShare + Nonshare mode Support)'. The 'Nand Flash Option Mode Enable' checkbox is unchecked. 'NVRAM Total Size' is '4096 MB'. The 'SDRAM Size (Mb)' dropdown is circled in red and set to '64'. 'Boot Loader File' is '.\BootCode\GPL326XX_NandBootLoader.hdb' and 'Block Counts' is '6'. 'BtLdr Work Start Addr.' is '0x00000000'. On the right, there are buttons for 'Config Load', 'Config Save', 'Load Default', 'Version Check', 'Pack', 'Download', 'Export', and 'Upload'. An 'Advanced Setting' button is at the bottom left.

詳細的 Pre-set Register & SDRAM Calibration 使用方法請見第四章。

GPL329XX/GPL327XX/GPDV6XXX：這些 body 將 Pre-Register 以及 DRAM Calibration 頁面分離



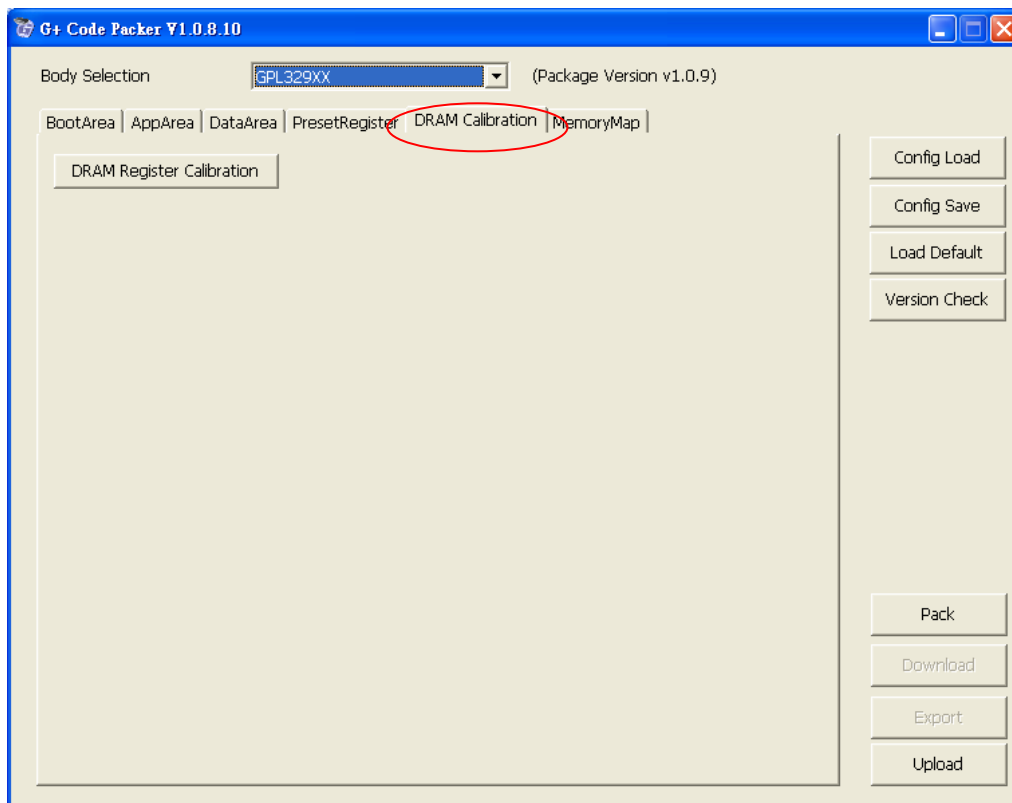
Body Selection: GPL329XX (Package Version v1.0.9)

BootArea | AppArea | DataArea | **PresetRegister** | DRAM Calibration | MemoryMap

Register Num: 30

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

Config Load
Config Save
Load Default
Version Check
Pack
Download
Export
Upload



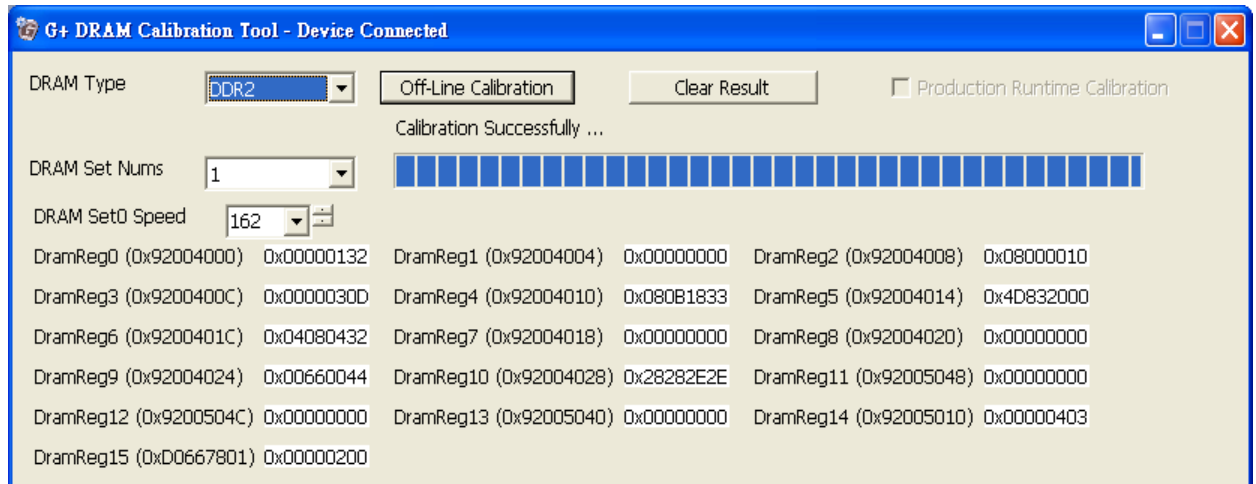
Body Selection: GPL329XX (Package Version v1.0.9)

BootArea | AppArea | DataArea | PresetRegister | **DRAM Calibration** | MemoryMap

DRAM Register Calibration

Config Load
Config Save
Load Default
Version Check
Pack
Download
Export
Upload

按下 DRAM Register Calibration 按鈕，會叫出 DRAM Calibration Tool 進行 Calibration 動作。

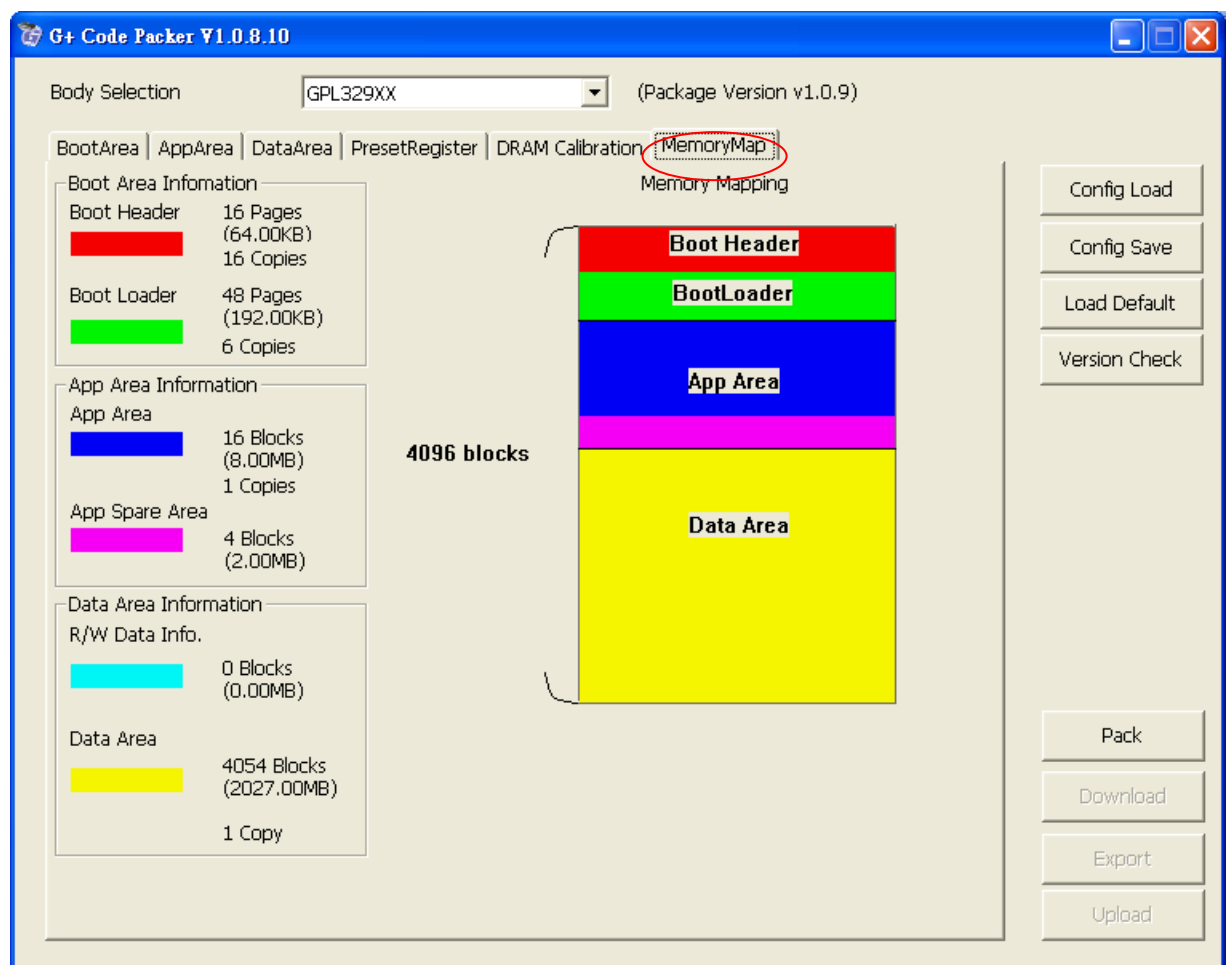


詳細的 Pre-set Register & DRAM Calibration 使用方法請見第五章。

3.2.5 Memory Map

將客戶對 NVRAM 之規劃以圖示的方式告知 NVRAM 之規劃方式，日後生產中 NVRAM 之架構，則以此圖為基準，由圖可清楚了解各區所占大小與 NVRAM 之使用概況，我們會依照各區原則做大型區域劃分。

Memory Map 的使用者介面與解說如下所示：



- Boot Area Information：以 Block 或 Page 為單位，會換算為 MB 或 KB 讓使用者了解實際大小，Block 之單位 Size 會依 NVRAM 的種類或型號而有所不同。
- App Area Information：以 Block 為單位，可闡述客戶之 Read Only 區大小，此區不應太過緊密，以利日後 code update 之便利性。
- Data Area Information：分為 R/W Data Info. (User Reserve Area) 與 Data Area (File System Area Size)。

3.2.6 Configuration Tool Buttons

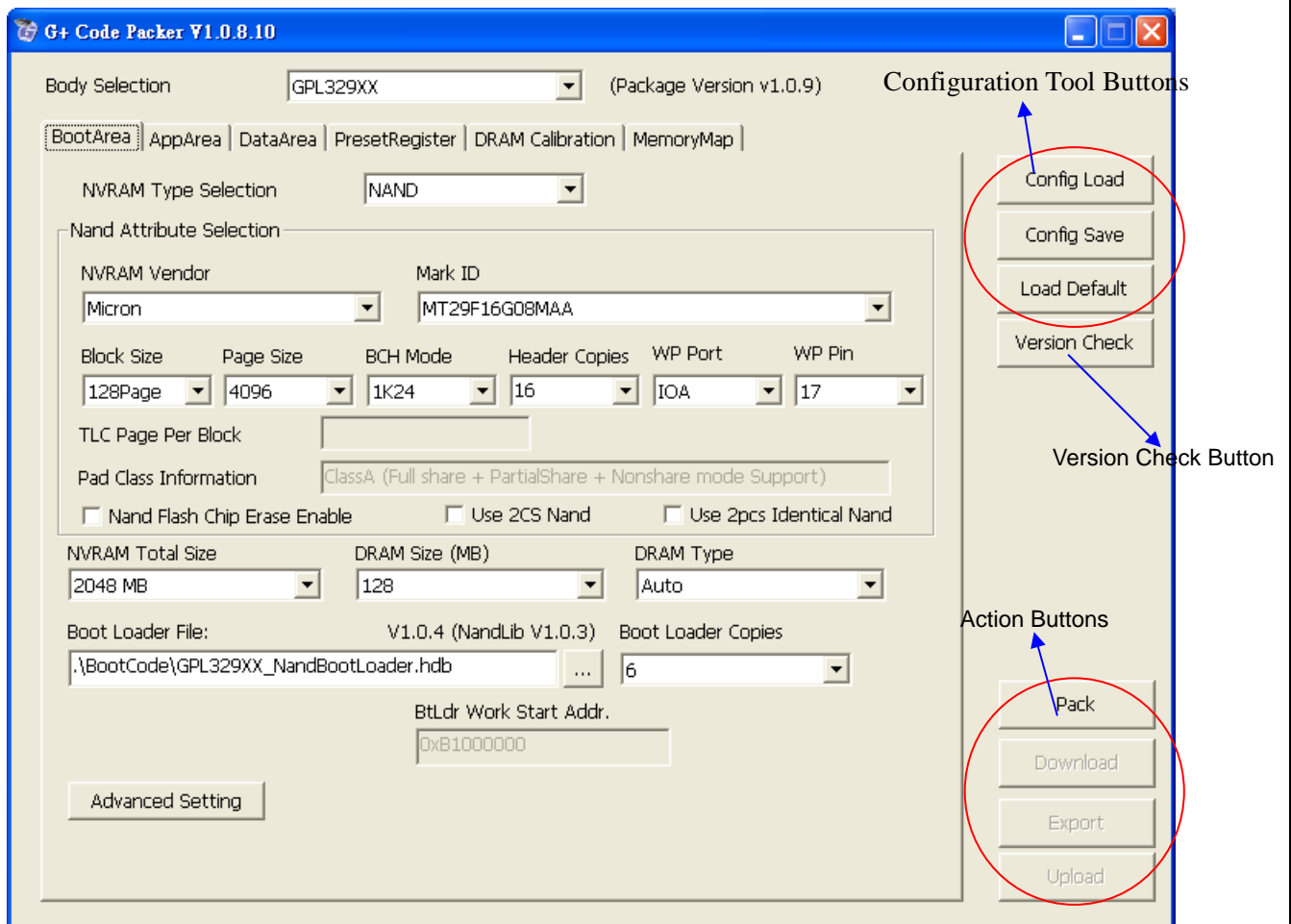
- a. **Config Save**：使用者可將設定好之設置值全數存取為指定之 **Config file**，請以此做為儲存。
- b. **Config Load**：使用者可將先前儲存過之 **Config setting** 透過 **Config Load** 重新叫回。
- c. **Load Default**：會 **Load** 回系統預設值。

3.2.7 Action Buttons

- a. **Pack**：可包裝出目前設置之 **ROM Image**。
- b. **Download**：呼叫 **G+CodePacker** 內建之 **USM Mass Tool**，直接透過 **USB** 下載包裝完成之 **code** 至 **NVRAM**
- c. **Export**：將 **Download** 所需的檔案 **Export** 到指定資料夾，供量產使用。
- d. **Upload**：讀回 **Nand Flash Image Data** 到指定資料夾。

3.2.8 Version Check Button

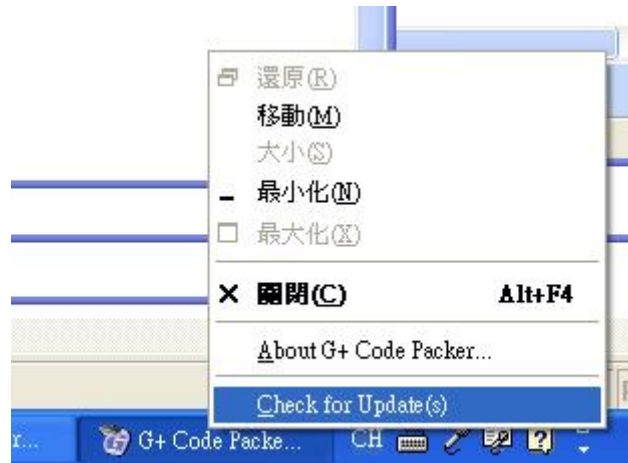
此按鈕提供用戶查詢本工具所內建的所有預設 Binary File (BootLoader, UsbBin...) 的版本以及所使用 Nand Library 版本資訊，**GeneralPlus** 強烈建議用戶的主工程所使用的 **Nand Library** 版本須和本工具所使用 **Nand Library** 版本最好能保持一致，以避免不可預期問題之發生，當然用戶對此工具有版本上之任何疑問或問題，可以連絡 Generalplus FAE 人員協助處理。



3.2.9 Check for Update(s)

在工具列上的 G+CodePacker 圖示上點擊右鍵，即可看到此功能之按鈕選項。此功能用於檢查是否存在更新版本的工具可供使用，並允許擁有 A-Key 的用戶直接下載新版工具使用。此功能的基本操作說明如下所示：

1. 在工具列上 G+CodePacker 圖示點右鍵，即可看到 Check for Update(s)選項。



2. 點擊此按鈕執行 Check for Update(s)動作。

若尚未安裝 SW Tool Online Updater 工具，會彈出以下訊息。



此時請到訊息中提到的目錄下執行 G+ Online Update Bootstrapper.exe，並按照指示進行安裝動作。



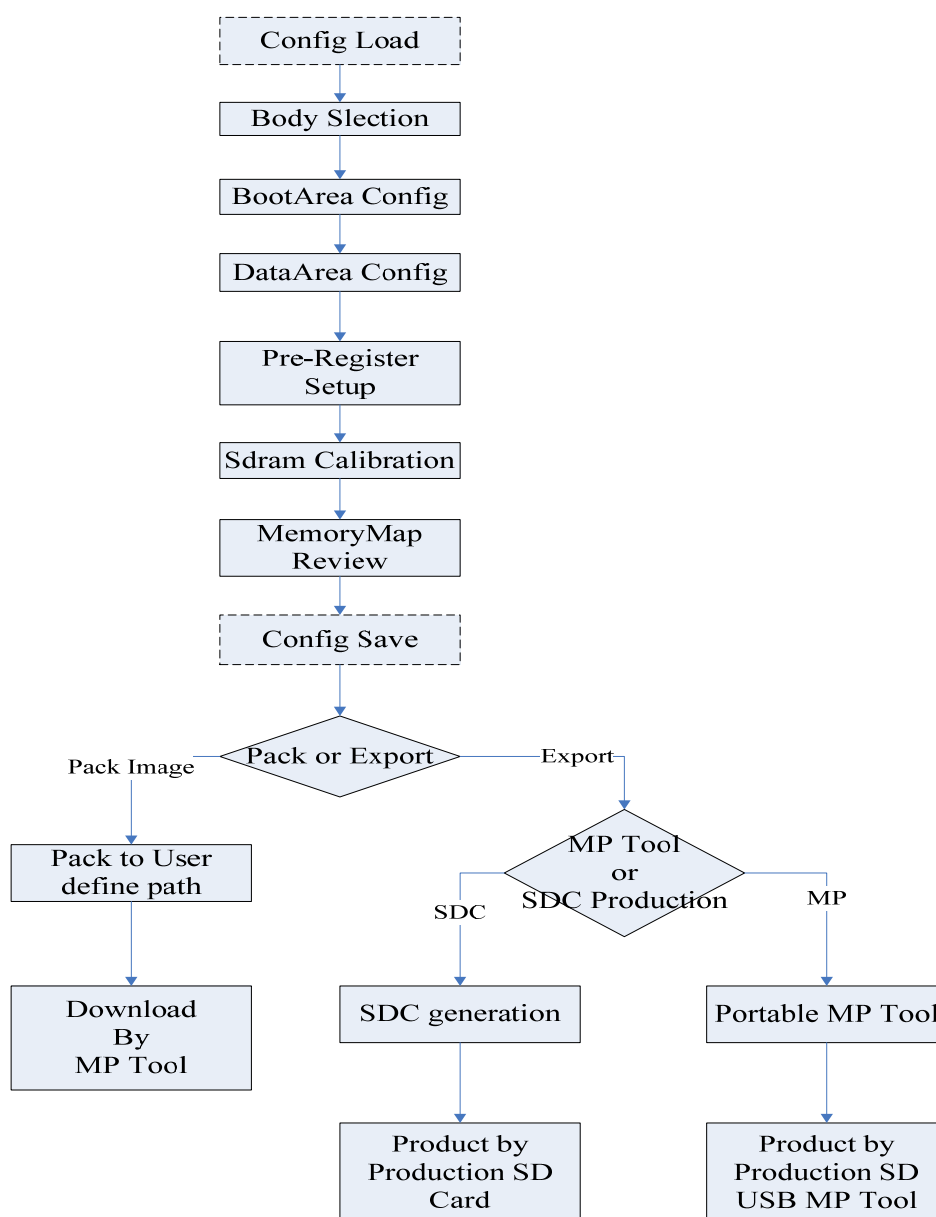
2.2 若已安裝 SW Tool Online Updater 工具，則會彈出此工具介面。可由工具介面上得知目前的工具版本以及最新版本等資訊，也可以設置自動檢查工具版本的週期。當 Tool 版本不是最新的時候，也允許擁有 A-Key 的用戶透過 Get Installation Package 按鈕，下載最新版本的工具來使用。



4 Nand Flash Boot操作流程解說(GPL325XX/GPL326XX)

GPL325XX/GPL326XX 將 Nand Flash 分為 3 個操作區域：Boot Area、App Area 以及 Data Area。使用者可透過 G+ Code Packer 分別設置這三個區域的資源檔來進行整合出燒錄 BIN 檔(ROM Image)。另外，Tool 支援 Pre-set Register 可以調用 SDRAM Calibration Tool 測試 SDRAM 參數，並把 SDRAM Register 參數整合到燒錄文件中。

接下來各個子章節將依序介紹 Boot Area、App Area、Data Area、Pre-set Register 設置、SDRAM Calibration、Pack 以及 Download 的操作流程(如下圖)及使用方式。



G+ Code Packer Basic Operation Flow

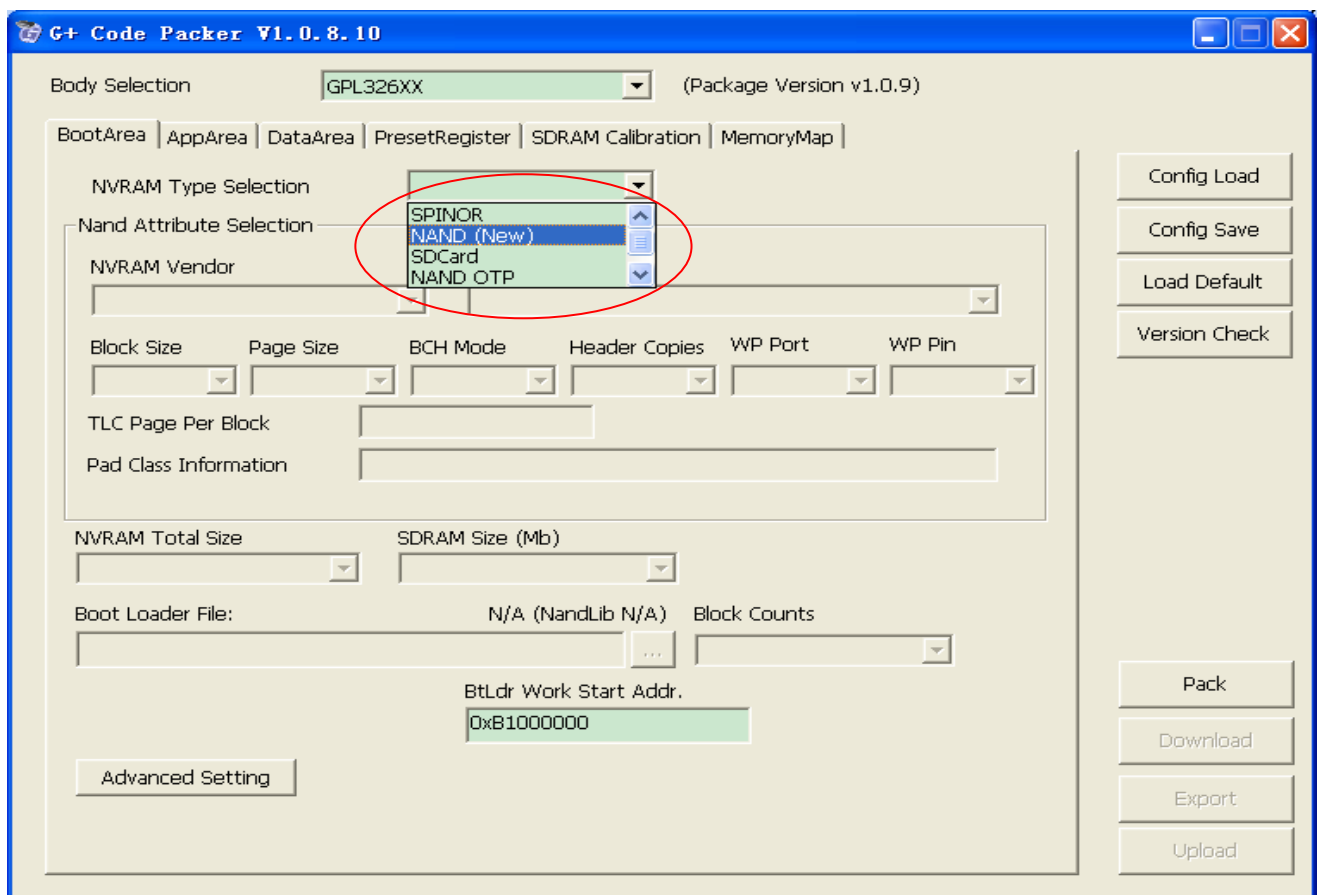
4.1 Load Config & Config Save

若先前已有做過 Config Save, 可透過 Config Load 方式將上次選擇之參數重新引入, Generalplus 強力建議所有之 Resource File, 全數放在 G+CodePacker 為 Root path 之路徑下 (可自行開 folder), 如此可對整體 G+CodePacker 之可攜性有顯著提升, 並可將使用者經驗透過 Config Save, 輕易轉交合作夥伴。

4.2 設置Boot Area

Step1 : NVRAM Type Selection

選擇 NVRAM(Non-Volatile RAM) 種類, 支持 SPINOR, NAND 或 SD Card (only for GPL326XX)。



Step2 : Nand Attribute Selection

若 Step1 選擇的是 SPINOR, 則 Step2 可以跳過。若 Step1 選擇的是 Nand, 則 Step2 就需要填寫 Nand Attribute。包括: Nand Vendor ID (廠商 ID)、Nand Mark ID (產品

ID)、Nand Type、Page Size、Boot Area 的 BCH mode、Boot Area 的 BCH Table 備份個數。

市面上 Nand Flash 大部分可以在 Vendor、Mark ID 查找到對應設置選項。如果用戶使用的 Nand 不包括在列表中，用戶就需要根據 Nand Flash 的 SPEC 填寫 Nand Type、Page Size 等設置(請於 NVRAM vendor 選擇 Manual Mode)。

除了 Manual Mode 外，透過 Nand Mark ID 之選擇，G+CodePacker 會貼心為您列出正確之 NVRAM Total Size，此時該欄位不能竄改，而 SDRAM Size 仍請使用者確實填寫，此乃重要參數，會影響系統之啟動能力與否，日後亦會以此參數做為 SDRAM Calibration 之重要參考依據。

Step3 : Boot Header BackUp Nand Block Numbers

填完 Mark ID 與 SDRAM Size 後，若有需要改寫 Header 之備份塊數，則可視情況增刪，若為優質之 Nand (非 re-use 之 Nand flash)，建議保持原廠預設值即可。

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

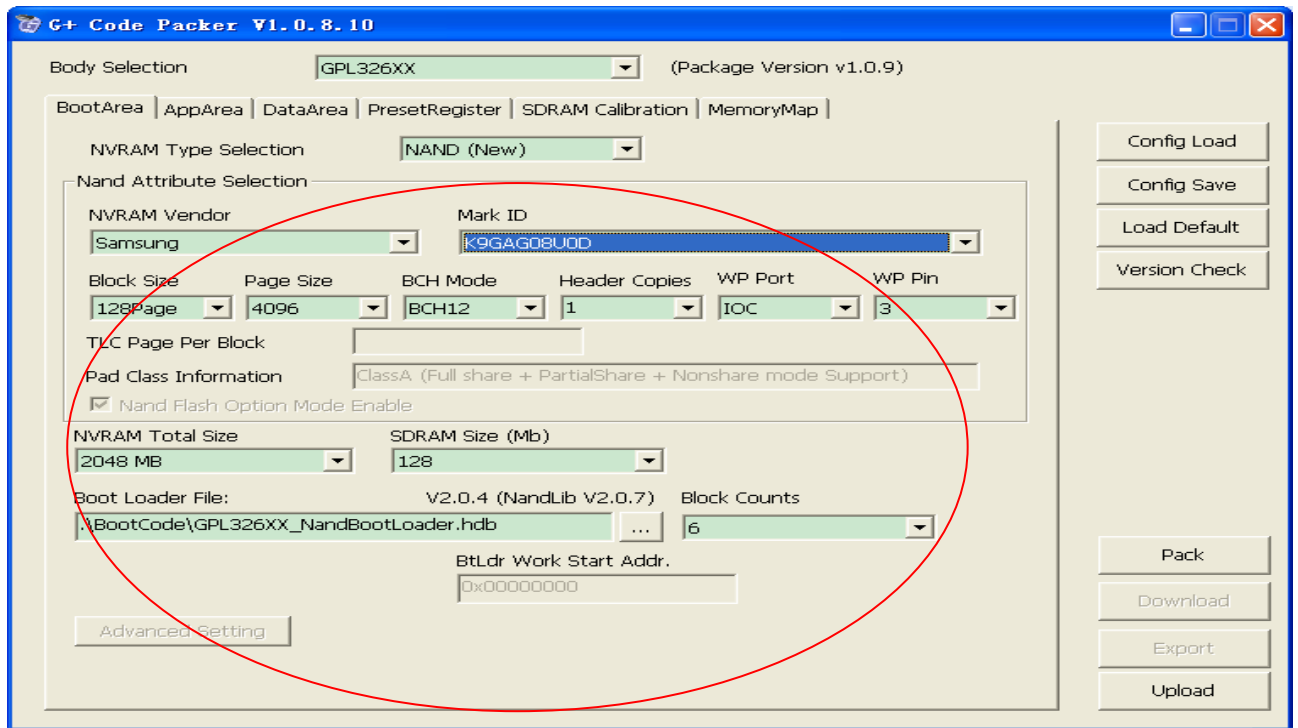
Write Protect Pin 亦為重要參數，在大多數開發板中，會將 Nand Write Protect pin 強制 pull high，此時該參數可設為 NULL，其餘請依線路設計來做設置，IOC3 為 Generalplus 原廠之公版預設值，亦為 G+CodePacker 提供的預設值，請務必理解與確實設置。

Step5: Boot Loader File Choice

Boot Loader File 強力建議採用原廠預設，但特殊需求之客戶亦可自行提供 Boot Loader，該 Loader file 為 GPL32XXX ROM Code 所會帶起之第一套程式，該程式將會帶入 Generalplus 管理區，包含放置客戶之 Runtime Code 或是 Read Only Resource 之 APP 區塊之初始化，還有 FAT 區之初始化皆在此完成，該程序對於使用 Generalplus 標準平台之使用者特別重要，且所內置之 Generalplus Standard Nand Driver Library 亦需與 Runtime code 所採用之 Nand Library 完全一致，始能保證 Nand 使用之正確性與穩定度。

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

Boot Loader Block Count 為 Boot Loader 之備份塊數，以確保 Boot Loader 能完美帶出，因此 Generalplus 建議多做幾份備份塊，Generalplus 之備份塊單位為 Nand Block Size，因此若 Block Size 為 512KB，假定 NandBootLoader 為 35KB，那麼一份備份塊可放置 $512 / 35 = 14$ (bin counts/Block)。請視需求增刪 (BootLoader 已受 BCH 保護)



4.3 設置App Area

Step1：App Area 參數設定

當 NVRAM Type 為 Nand 時，App 參數才需要設置。如果 NVRAM Type 為 SPINOR，則 Step1 可以跳過。App 參數包括：App Man Size、App Spare Rate、App ROM Code Copies 以及 CRC Check Sum。

App Man Size：

App Area 的規劃使用空間。注意：App Main Size 必須大於 Run Time ROM Code Size 和 Resource File Size 的總和。

App Spare Rate:

App Area 交換塊的比率。為了防止 App Area 出現壞塊造成資料丟失，Tool 設計了一定比率的交換塊。App Spare Size 為 App Man Size 與 App Spare Size 之換算結果

$$AppSpareSize = \frac{AppMainSize}{AppSpareRate}, \text{ (unit: Nand Blocks) }。$$

例如：

Block Size = 64, Page Size=2048, App Main Size = 16MB (=128 blocks), App Spare Rate = 5, 那麼, App Spare Size = $128 \text{ blocks} / 5 = 25.6 \approx 26 \text{ blocks} = 3.25\text{MB}$

App Spare Size(以 block 為單位)最大值和最小值限制如下:

最大值 = Page Size / 4, 最小值一律限制為 4.

因此, 合法值必須滿足下列兩條件:

1. $4 \leq \text{App Spare Size} \leq \text{Page Size} / 4$, 單位為 block.
2. $\text{App Spare Size} \leq \text{App Main Size}$

以上述相同例子而言, 要同時滿足這兩條件, 以 block 為單位,

$4 \leq \text{App Spare Size} \leq 2048 / 4$

$\text{App Spare Size} \leq 128$

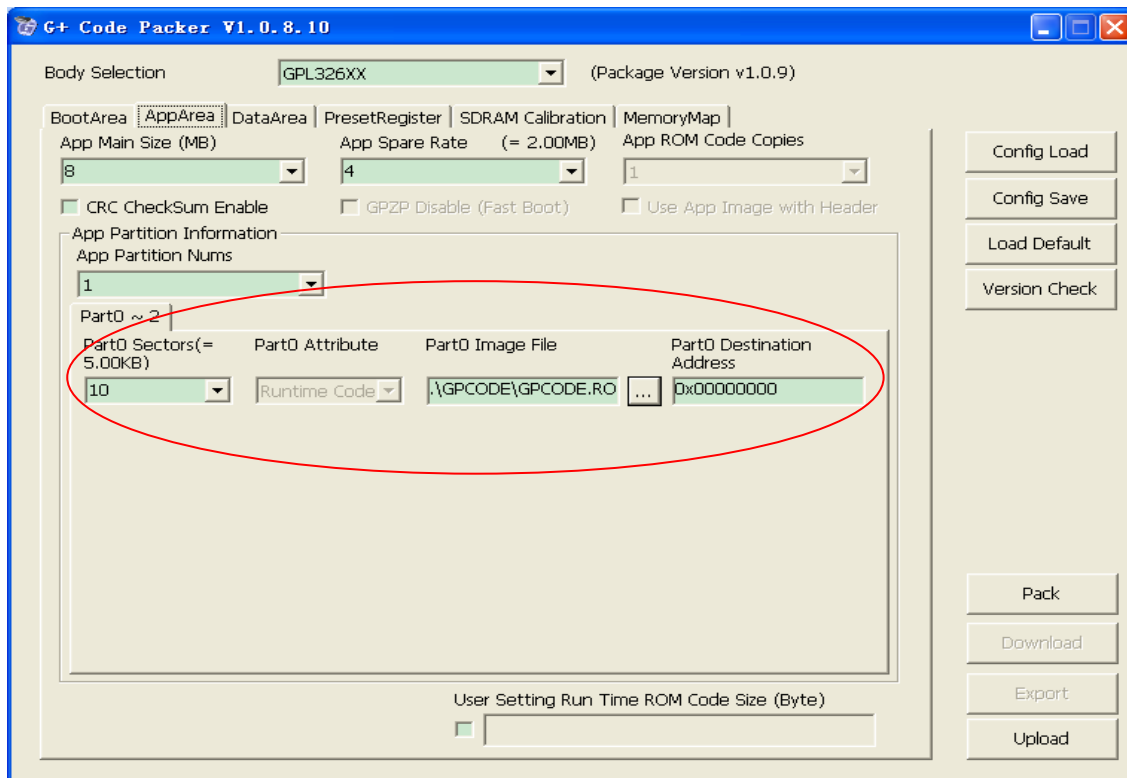
因此 App Spare Size 合法範圍為

$4(=0.5\text{MB}) \leq \text{App Spare Size} \leq 128(=16\text{MB})$ 意味著 $1 \leq \text{App Spare Rate} \leq 32$

所以合法可設定的 App Spare Rate 為 1 ~ 32, 對應的 App Spare Size 為 4 ~ 128 blocks (=0.5 ~ 16MB).

Step2 : Setting App Partition Information

Boot Loader 執行起來之後會把使用者選擇的 Run Time ROM Code 複製到 SDRAM 執行。如果 Setting Run Time Rom Code Size 沒有勾選，則 Boot Loader 複製的大小將由 tool 根據 Run Time ROM Code 的實際大小計算得到；如果勾選，則複製大小將等於填寫的數字 (以 Byte 為單位)。



如上图，此页面主要是对 App Partition 进行配置，一是设定分区个数；二是为每个分区配置相应的参数。

App 分区信息参数配置如下：

- **App Partion Num:** 此选项表明用户当前设定的 App 区的分区数。目前设定范围 1~2，共 2 组值，也就是用户最多可配置 2 个分区。
- **Partion N Size:** N 的范围从 0~1，它是表明当前这个分区的容量大小，以 Sector 为单位，通常情况下，当我们选定分区镜像档案后，Tool 会自动帮我们将相应的档案大小填入此栏位。有一种情况是，用户可以不选取档案，此时，该栏位会被置为 auto,如果用户想指定此分区大小，可以直接输入该值。
- **Partion N Attribute:** 同理，N 的范围是从 0~1,它表明为当前镜像文件是什么类型的文件，目前 Tool 提供 Runtime Code, Resource Part 共 2 种类型的 App 镜像文件，且是 disable 状态，表明用户暂时不能更改。第一个分区摆放 Runtime Code；第二个分区摆放 Runtime code 所需要用到的 Resource 档案。当 First bootloader 在搬 App 区的 code 时，此参数就显得尤为重要。
- **Partion N Image file:** 此参数是让用户来选取磁盘上的分区镜像文件。

- **Partion N Destination Address:** 此参数是指定当前分区文件,将来会被摆放在 SDRam 哪个地址来跑。

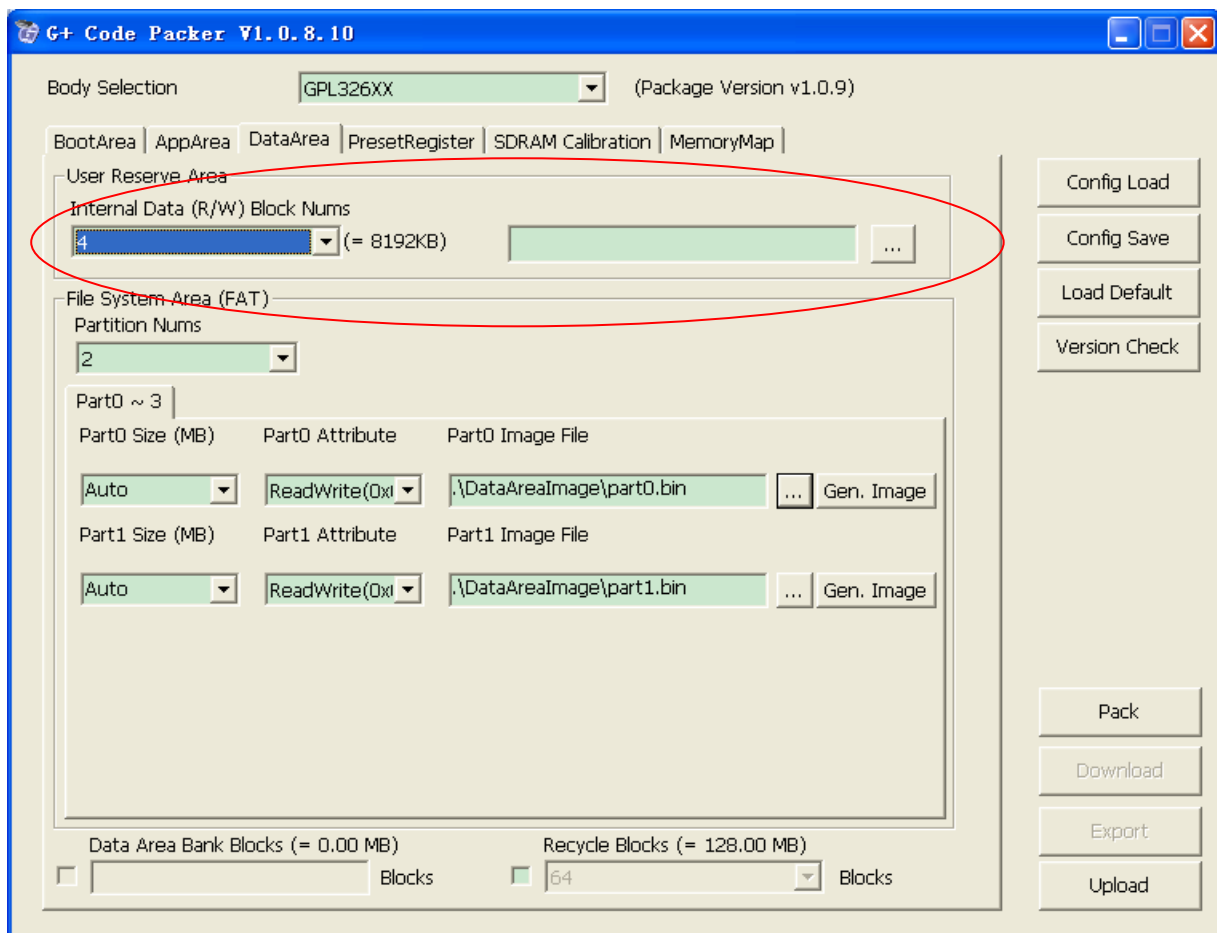
依此方法,对指定数目的分区进行配置,直到所有分区信息都配置完成。

4.4 設置Data Area及FAT Image Tool之使用方法

4.4.1 設置Data Area

Step1：設置 User Reserve Area Size 和 Image 檔案：

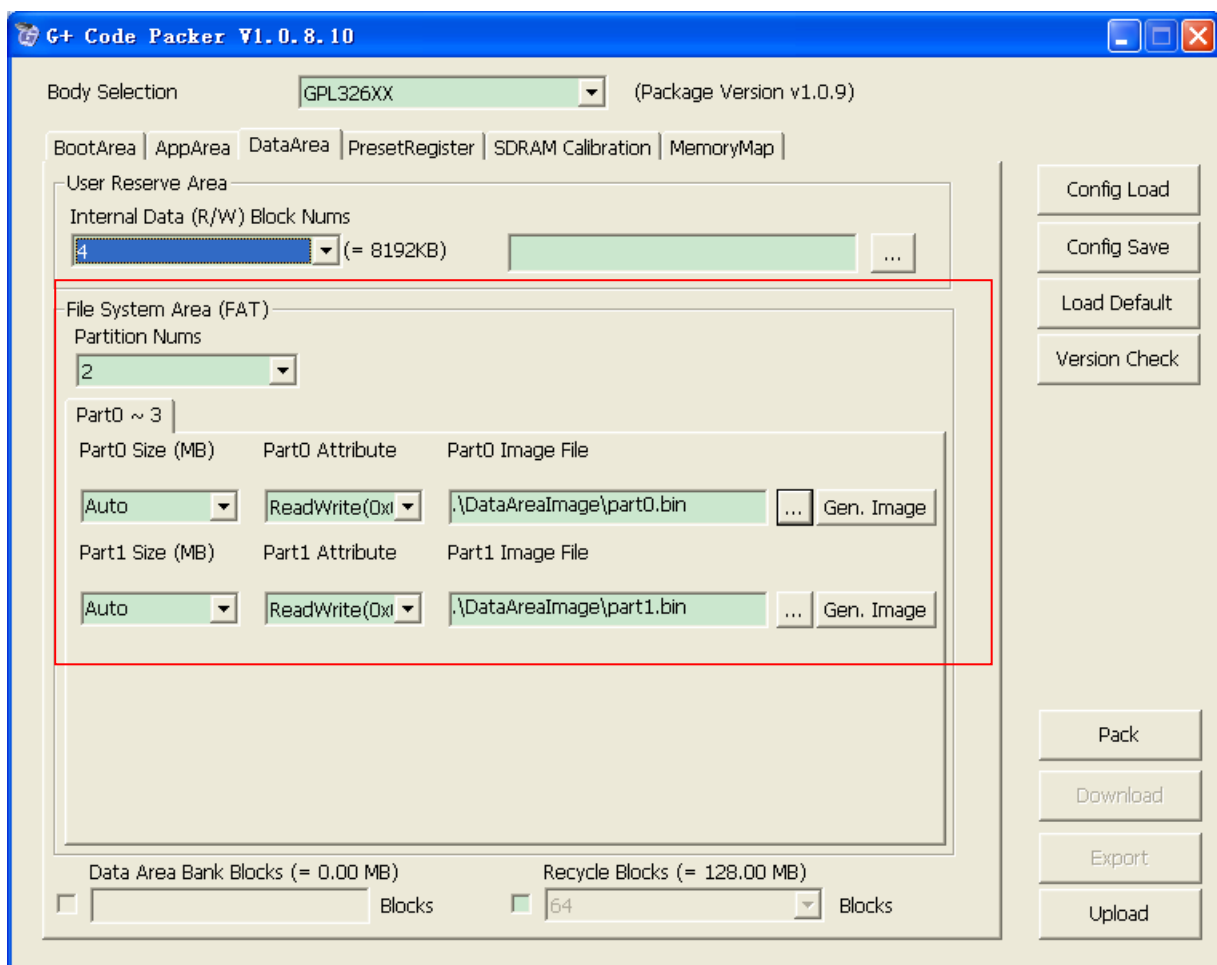
User Reserve Area 用於存放 App ROM code R/W 資料。User Reserve Area Size 的選擇以 Block 為單位,選擇完畢會在選項後方顯示換算出的 Reserve Area Size 以 KBytes 表示,方便使用者了解實際規劃的大小。R/W 區可預載內容(請自行提供 pre-load image),該內容可透過 Generalplus 提供之 APP Area 相關 API 作讀取動作。



Step2 : File System Area 設置

File System Area 分區，設置每個區域的 size 及對應 Image 檔案。SPINOR 只支持一組的 File System 分區，而 Nand Flash 最多支持二組的 File System 分區，使用者可依據應用需求自行規劃。

各區域的 size 可自行輸入規劃的大小，以 MBytes 為單位。若是選擇 Auto，則由 tool 依據選擇的 Image 檔案大小決定此區域的大小。Attribute 部分則有 Read Only、Write Only 以及 Read/Write 三種可供選擇。



Step3 : Advance Recycle Blocks (Option Choice) 設置

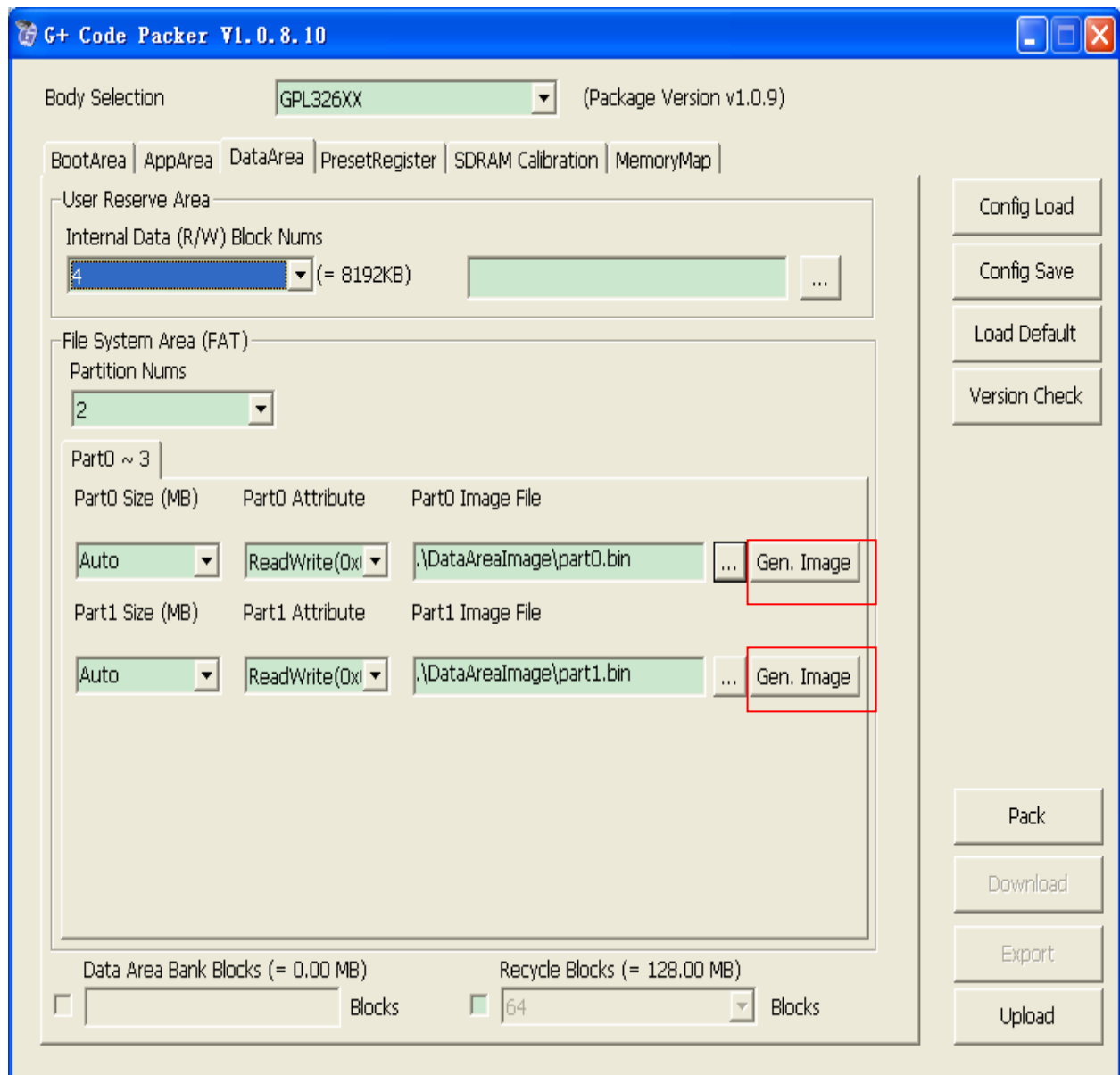
Generalplus 強烈建議採用原廠預設值，此為 Data Area manage exchange Block number for each Bank 之進階設置。而 Bank Size 預設為 512 Block，目前不開放設置。

4.4.2 FAT Image Tool之使用方法

File System Area 除了可以選擇已製作好的 Image 檔案，也整合了製作 Image 檔案的工具以方便用戶使用。操作流程如下所示：

Step1：執行 FAT Image Tool

由 FAT Image Tool 所製作出的 Image 會自動將檔名及 Partition Size 資訊帶回工具中。因此先選擇需要的 Partition Number，再點擊對應 Partition 後的”Gen. Image”按鈕，執行 FAT Image Tool，進行 Image 之製作。



Step2：設置 FAT Image Tool 相關選項並製作 Image 檔案

FAT Image Tool 執行後，需設置 FAT Disk Size、Cluster Size、FAT Image Source Folder 以及 FAT Image Output Path 參數。各參數的意義及設置注意事項如下：

1. FAT Disk Size：

設置產生的 Image 檔案之 Partition Size。這裡設置的值必需大於等於所選擇 Source Folder 的 size，否則在產生 Image 的過程會失敗。

2. Cluster Size：

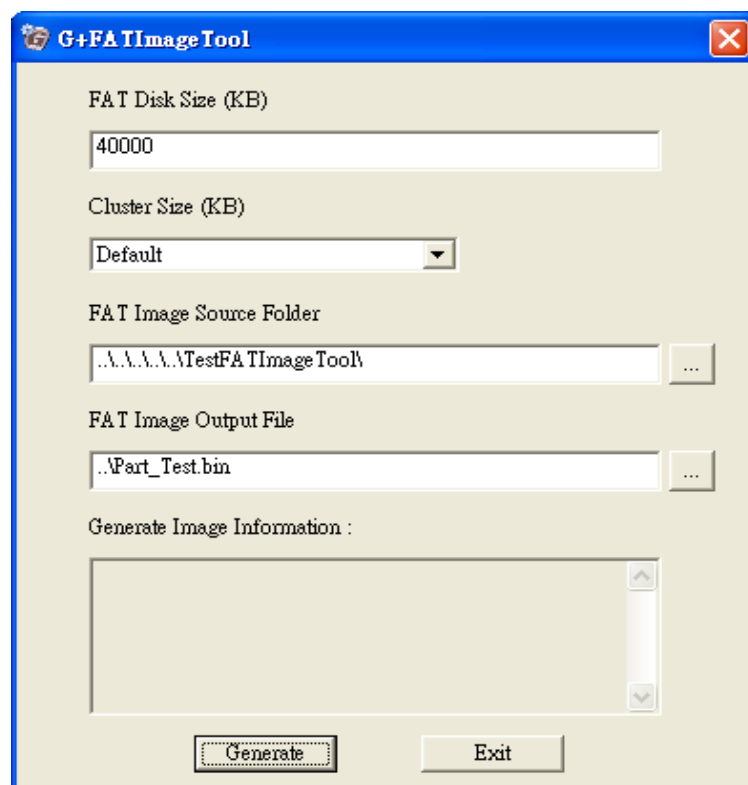
設置 Cluster Size。此設置建議使用預設值，若此設置錯誤，可能會造成 Image 生成失敗或是產生出來的 Image 檔案無法正確被辨識。

3. FAT Image Source Folder：

設置用來產生 Image 檔案的來源目錄。工具會將設置的目錄製作成 Image 檔案，包含所有目錄下的檔案、目錄結構以及子目錄下的檔案。

4. FAT Image Output Path：

設置產生的 Image 檔案之路徑與檔名。

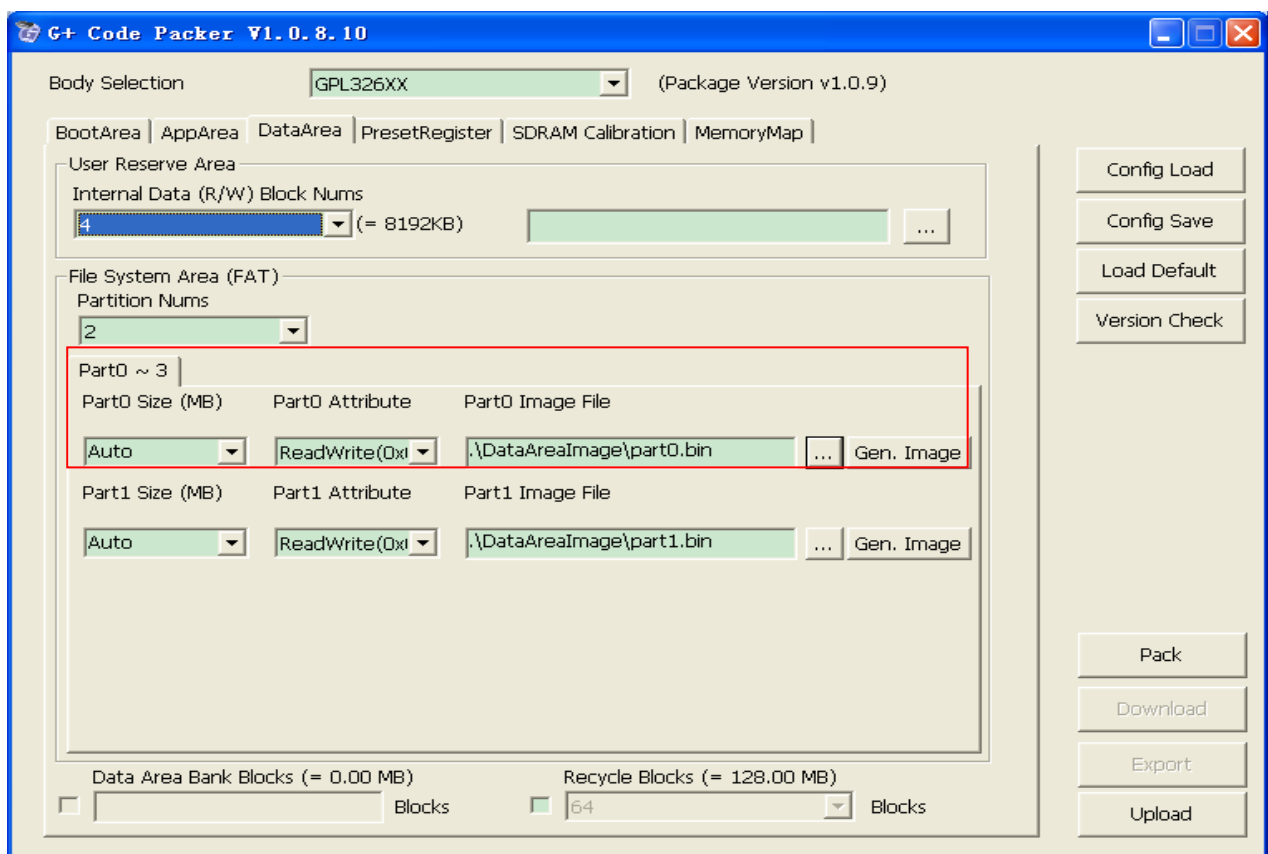
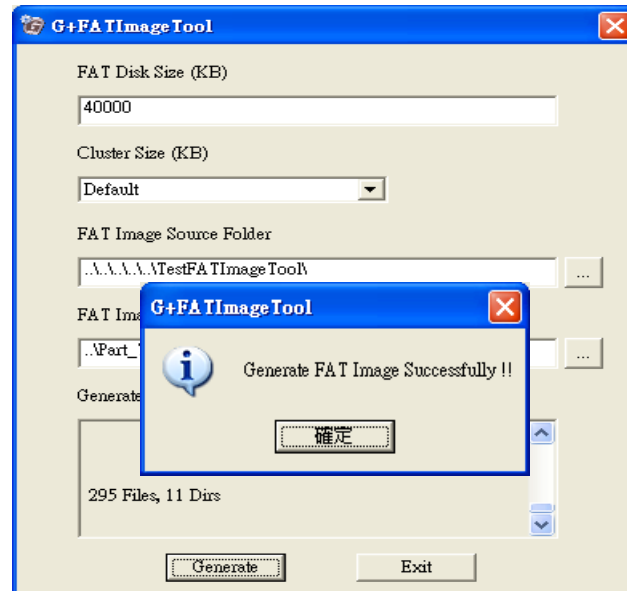


設置完成後點擊“Generate”按鈕，開始進行 Image 檔案之製作。製作過程中可由“Generate Image Information”視窗看到目前的製作進度，製作到哪個目錄及哪個檔案。

Note: 繁體中文作業系統下的來源檔目錄不能有簡體檔名或簡體子目錄名存在，而簡體中文作業系統下的來源檔目錄下不能有繁體檔名或繁體子目錄存在，否則 image file 製作會失敗。

Step3：使資訊自動帶入 Code Packer 中

Image 檔案製作完成，會彈出”Generate FAT Image Successfully”視窗，點擊確定後即可關閉 FAT Image Tool，資訊會自動帶入 Code Packer 中，即完成製作 Image 動作。



Note :

FAT Image Tool 針對用戶設置 **Disk size** 小於 **17MB** 的狀況做了一些特殊處理，說明如下：

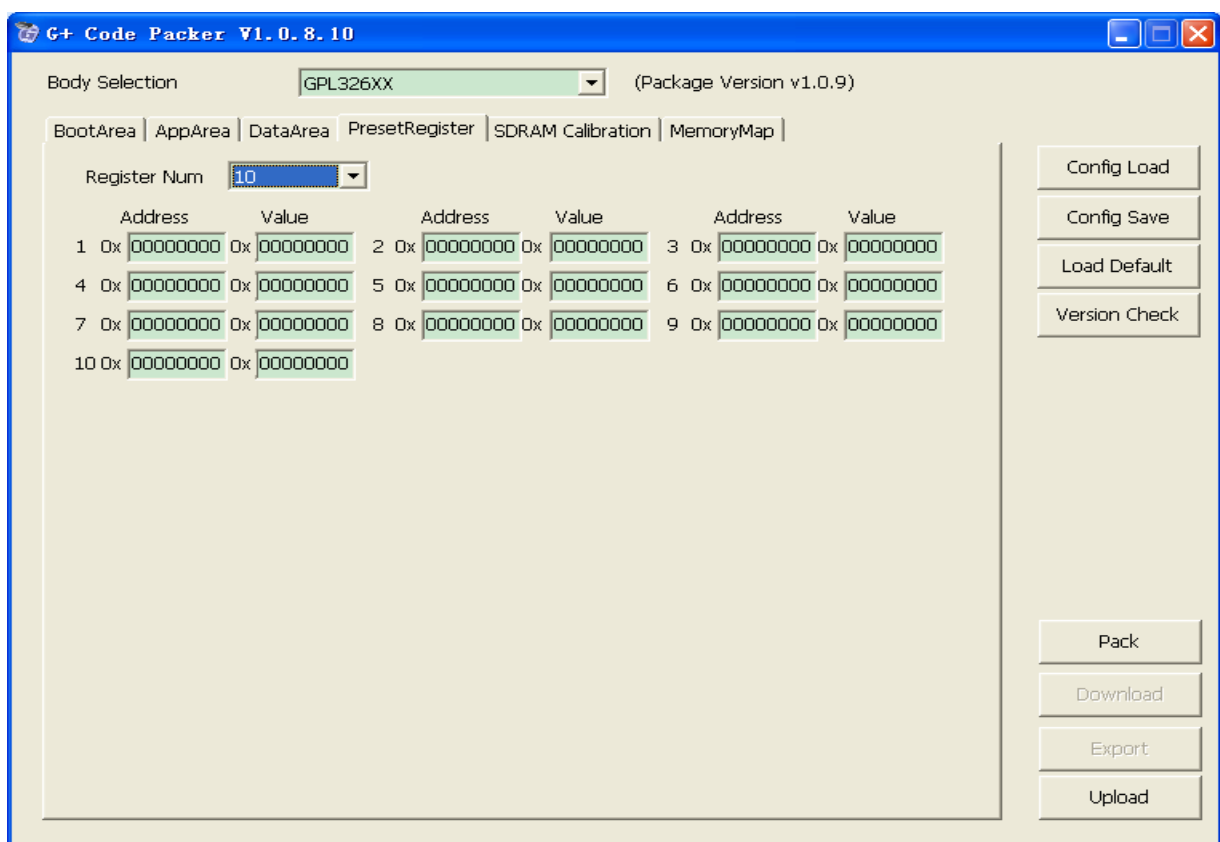
1. 若用戶填入 **Disk size** 小於 **17MB**，**Tool** 將會將磁盤虛擬成 **17MB**，並格式化成 **FAT16**，但實際可用空間仍為用戶實際填入的 **Disk size** 大小。
2. 若用戶填入 **Disk size** 小於 **17MB**，選擇 **MBR** 打包，**MBR** 總空間也將是虛擬後的大小。

此修正將會造成任何 **Body** 只要產生的 **Image** 是小於 **17MB**，當 **PACK** 後再重新開啟一定都會顯示 **17MB**。且載入小於 **17MB** 由 **FAT Image Tool** 產生的 **Image** 也會顯示為 **17MB**。

4.5 設置Pre-set Register及SDRAM Calibration之使用方法

4.5.1 設置Pre-set Register

使用者可依據本身需求設置相關的 Register 值。SDCard 和 SPI Flash 只支持 10 組 Register 設定，而 Nand Flash 支持 30 組的 Register 設定。每一組設定皆需設置 Register Address 與 Register Value。



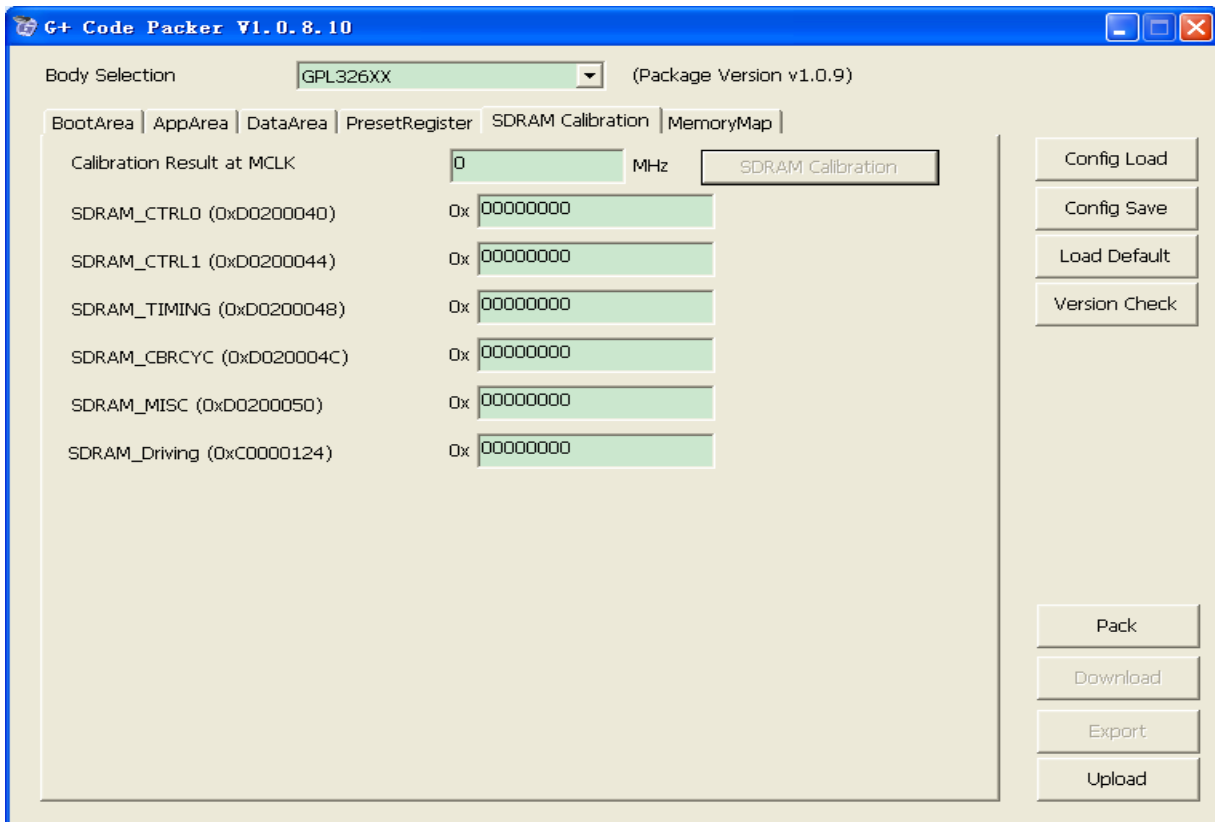
4.5.2 SDRAM Calibration的使用

Step1 : Click SDRAM Calibration button

點擊 SDRAM Calibration 按鈕，喚起 SDRAM Calibration Tool。硬體進入 Mask Rom Code 連接電腦。Tool 若識別到硬件，則會在標題列顯示 Device Connected，否則顯示 Device Not Found。所有 Calibration 操作必須在 Device Connected 狀態下進行。

請由 Internal ROM Code Boot 進入 USB Service Mode (IOB3 Pull High and reset IC, then boot from Rom Code)

SDRAM Calibration Entry for GPL326XX:



Body Selection: **GPL326XX** (Package Version v1.0.9)

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

Calibration Result at MCLK: **0** MHz **SDRAM Calibration**

SDRAM_CTRL0 (0xD0200040): 0x **00000000**

SDRAM_CTRL1 (0xD0200044): 0x **00000000**

SDRAM_TIMING (0xD0200048): 0x **00000000**

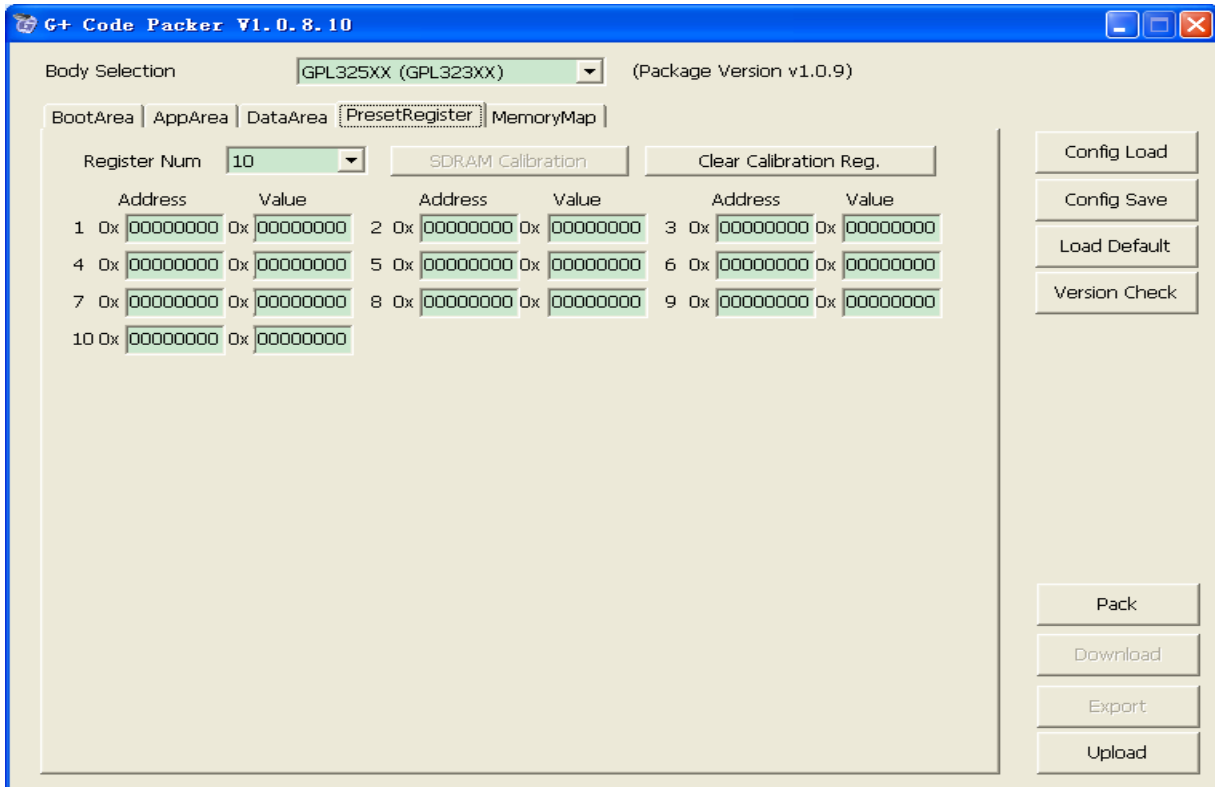
SDRAM_CBRCYC (0xD020004C): 0x **00000000**

SDRAM_MISC (0xD0200050): 0x **00000000**

SDRAM_Driving (0xC0000124): 0x **00000000**

Buttons: Config Load, Config Save, Load Default, Version Check, Pack, Download, Export, Upload

SDRAM Calibration Entry for GPL325XX:



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

BootArea | AppArea | DataArea | **PresetRegister** | MemoryMap

Register Num: **10** **SDRAM Calibration** **Clear Calibration Reg.**

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				

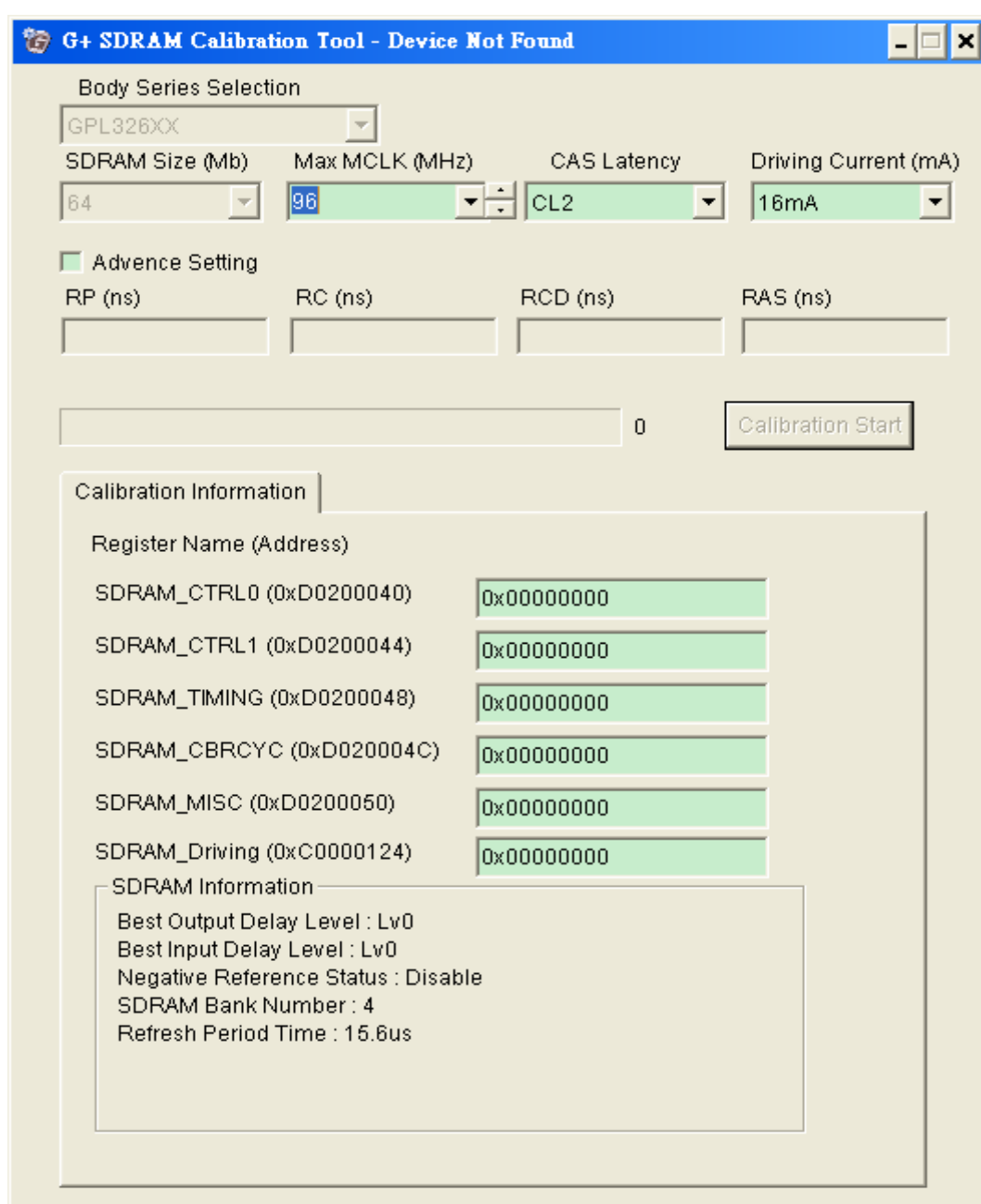
Buttons: Config Load, Config Save, Load Default, Version Check, Pack, Download, Export, Upload

Step 2. Body Selection & SDRAM Size

若由 G+ Sdram calibration 獨立執行請確實選擇此兩項參數。

Step 3. Entry Calibration Tools (sub tool)

請選定最高執行速度 與 Sdram driving 能力, 經驗上來看, EMU Board 設最高, 但量產板考量 EMI issue 可設低於 8mA (預設為 16mA 可適用大部份開發環境)。不同之 Driving current 可能產出不一樣之 Calibration 結果。



Register Name (Address)	Value
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

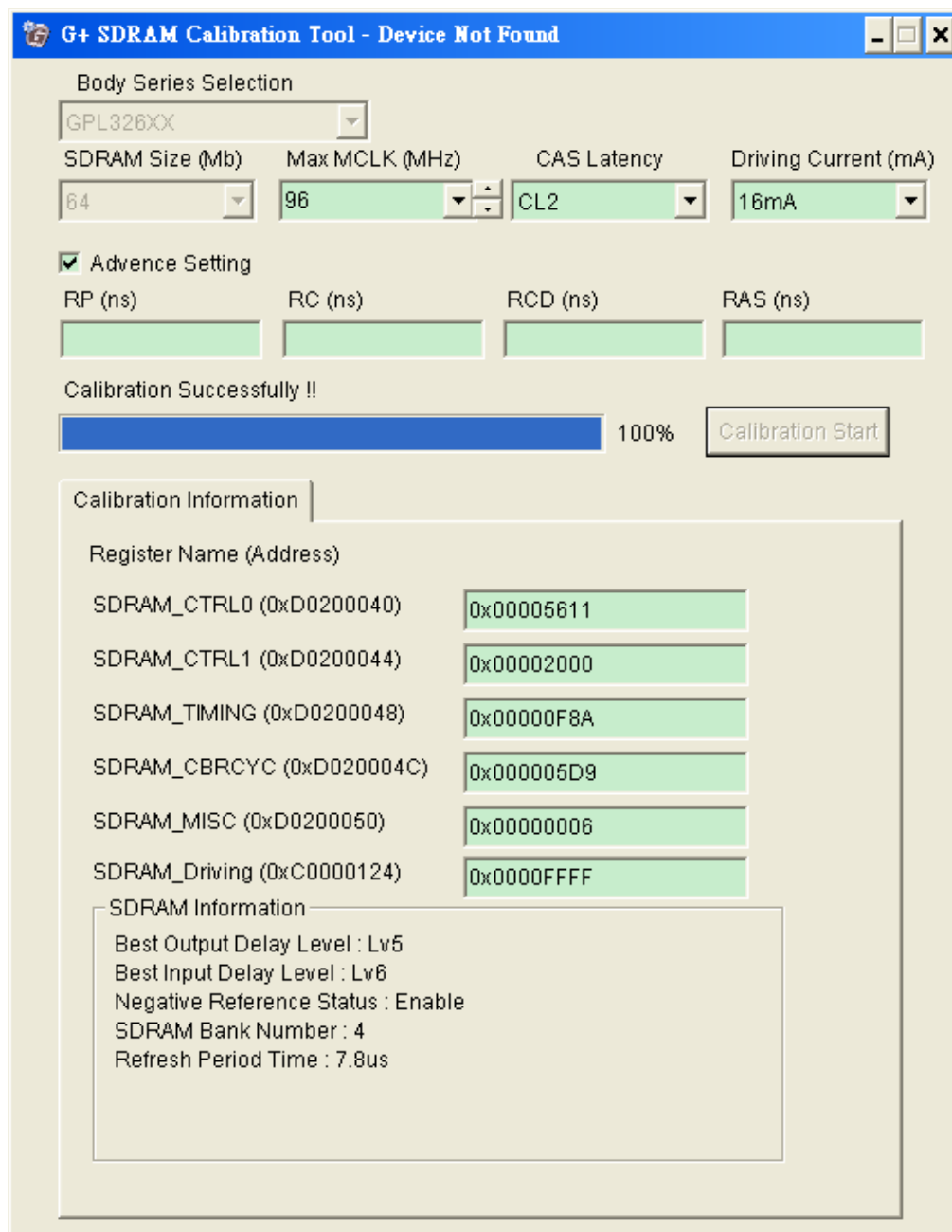
Step4：選擇 **Max Sys clock**、**CAS Latency**、**Driving Current**。

Step5：進階使用者可依據所用 **SDRAM spec** 規範指定 **RP/RC/RCD/RAS** 等 timing 值(單位為 **ns**)，請注意填入值不支持浮點數，若 **spec** 有浮點者，請無條件進位為整數。若用戶不需指定 **RP/RC/RCD/RAS** 等 timing 參數 (**Advance Setting** 不勾選)，這些參數將使用 **tool** 預設值。

Step6：點擊 **Calibration Start**。

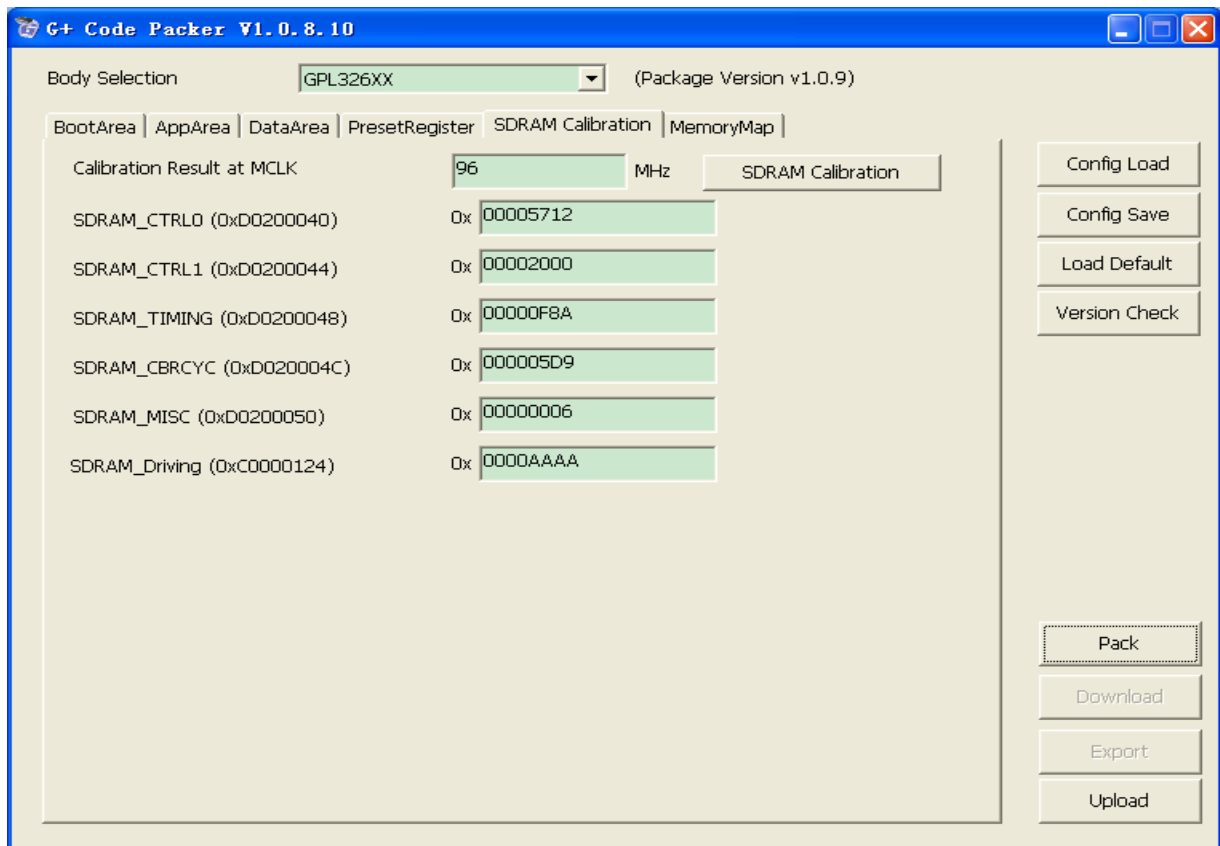
Step7：**Calibration Success** 並會得到一組校正過的 **SDRAM** 參數。

註(重要)：完成 **SDRAM Calibration** 動作請拔除 **GPL32** 整體電源 (or 做 **Hardware reset**)。



Step6：參數帶入 G+CodePacker 中 (GPL325XX/GPL326XX 帶入方式有差異)

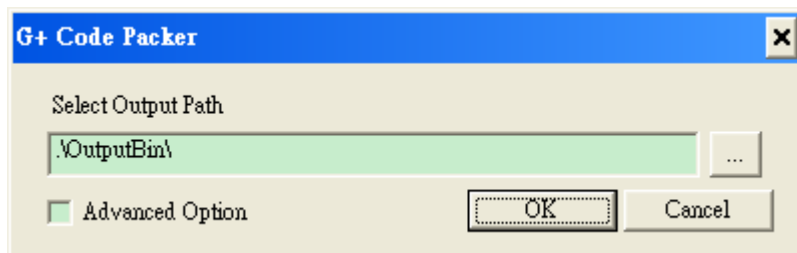
關掉 SDRAM Calibration Tool。Tool 會自動將取得之參數自動帶入 G+ Code Packer 之 Pre-set Register 中。此自動帶入的動作，會固定將 SDRAM 參數放在前五組 Register 設定中，若前五組 Register 設定已存在，則原先設置將會遞移至 SDRAM 參數後依序排列。若想清除 SDRAM 參數，回復原先設置，可使用 Clear Calibration Reg. 按鈕。此按鈕會清除 SDRAM 參數，若未進行 SDRAM Calibration，則此按鈕將沒有作用。



4.6 Pack與Download

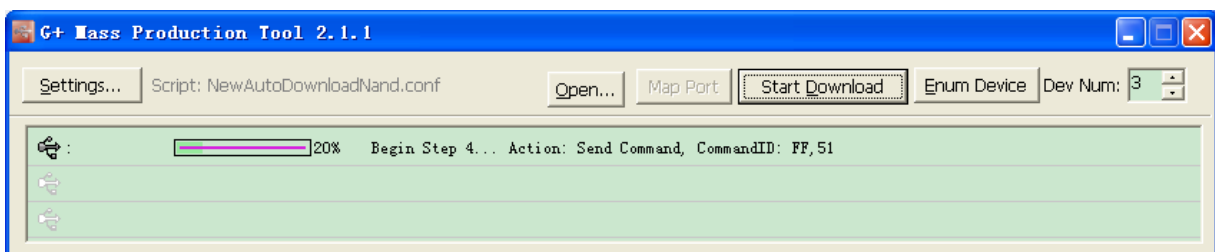
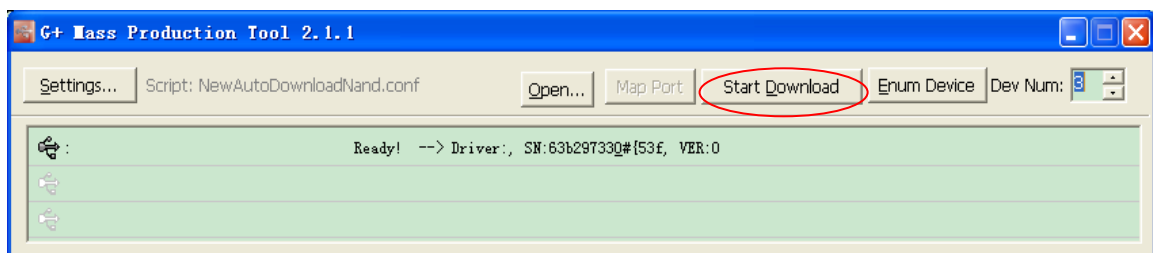
Step 1 : Pack

設置完各區域後，可以照需求點擊 **Pack** 按鈕進行整合動作。此時 tool 會依據設置將所有檔案整合至最終的 BIN 檔當中，並將 BIN 檔放置於使用者選擇的目錄下(建議使用預設路徑)。



Step 2 : Download

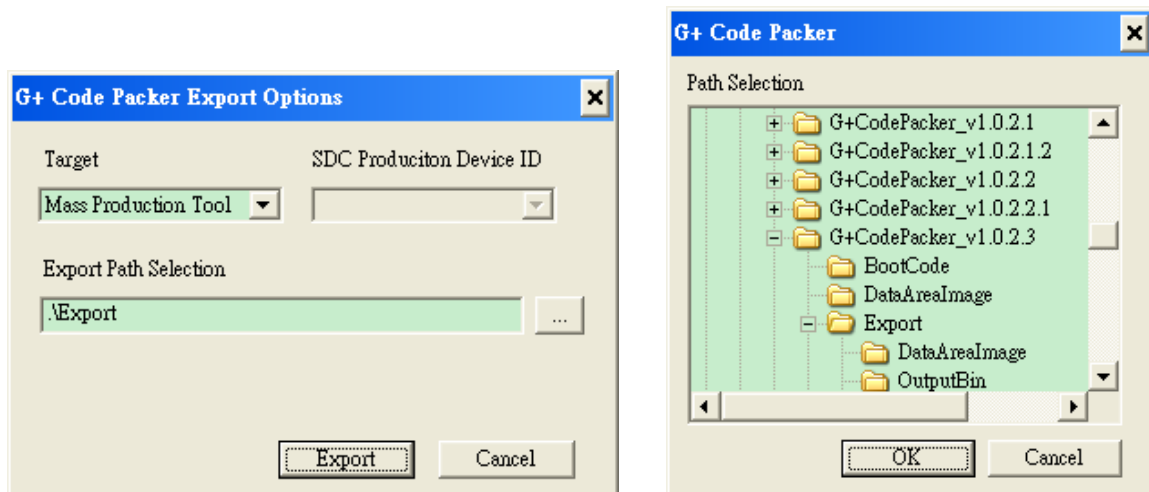
Pack 動作完成後，Download 按鈕會被激發。點擊 Download 按鈕，tool 會喚起 USB Mass Production Tool，並設置 Download 參數。使用者只需將裝置以 USB 連接電腦，當 USB Mass Production Tool 辨識到裝置後，按下 **Start Download** 按鈕，即可進入 Download 程序。



4.7 Export MP Tool to Other Disk (ex: USB Disk for Factory)

Step 1: Export Window

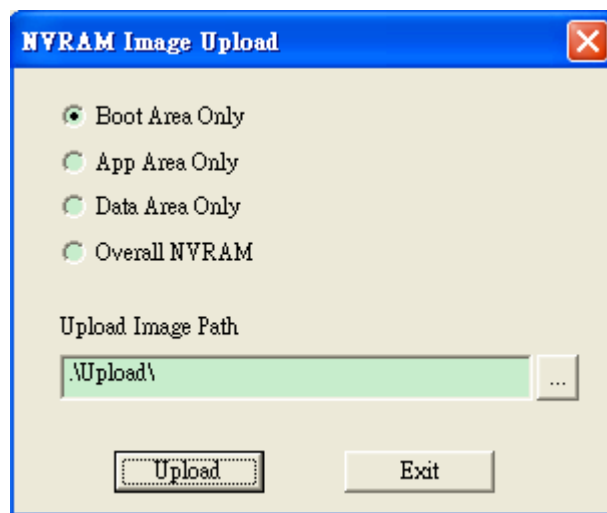
選擇 Mass Production Tool, 預設 Export 路徑為 G+CodePacker 根目錄下之 Export folder, 或可點選 Path exchange box 改變 Export 路徑 (如 U 盤)。



Step 2: Export 之資料夾已做好 **Portable** 工程, 可將整體資料夾做可攜性移動配置。

4.8 Upload

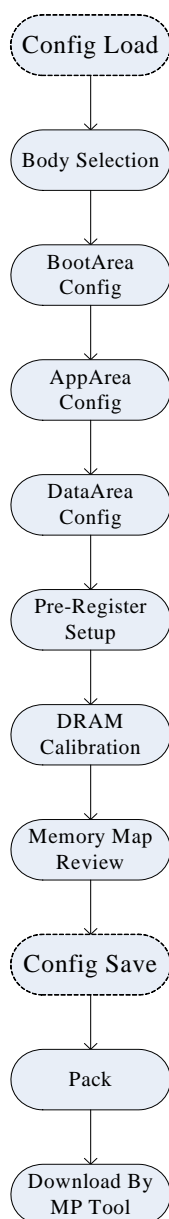
G+CodePacker 另有提供 upload 功能給使用者讀回目前 Nand Flash Image Data, 如需比對 Nand Flash 的實際資料和當時燒錄的資料有無差異, 或是目前 Nand Flash 資料欲複製到另一顆 Nand Flash, 可使用此功能. 下面視窗為 Upload 功能畫面, 可供設定讀回 Nand 某一區域(Boot or App or Data Area)或全部區域資料, 指定讀回資料放置的資料夾, 當點選"Upload"進行讀回動作, 點選"Exit"關閉該視窗。



5 Nand Flash Boot操作流程解說(GPL329XX)

GPL329XX 將 Nand Flash 分為 3 個操作區域：Boot Area、App Area 以及 Data Area。使用者可透過 G+CodePacker 分別設置這三個區域的資源檔來進行整合出燒錄 BIN 檔(ROM Image)。另外，Tool 支援 Pre-set Register 可以調用 DRAM Calibration Tool 測試 DRAM 參數，並把 DRAM Register 參數整合到燒錄文件中。

接下來各個子章節將依序介紹 Boot Area、App Area、Data Area、Pre-set Register 設置、DRAM Calibration、Pack 以及 Download 的操作流程(如下圖)及使用方式。



G+ Code Packer Basic Operation Flow

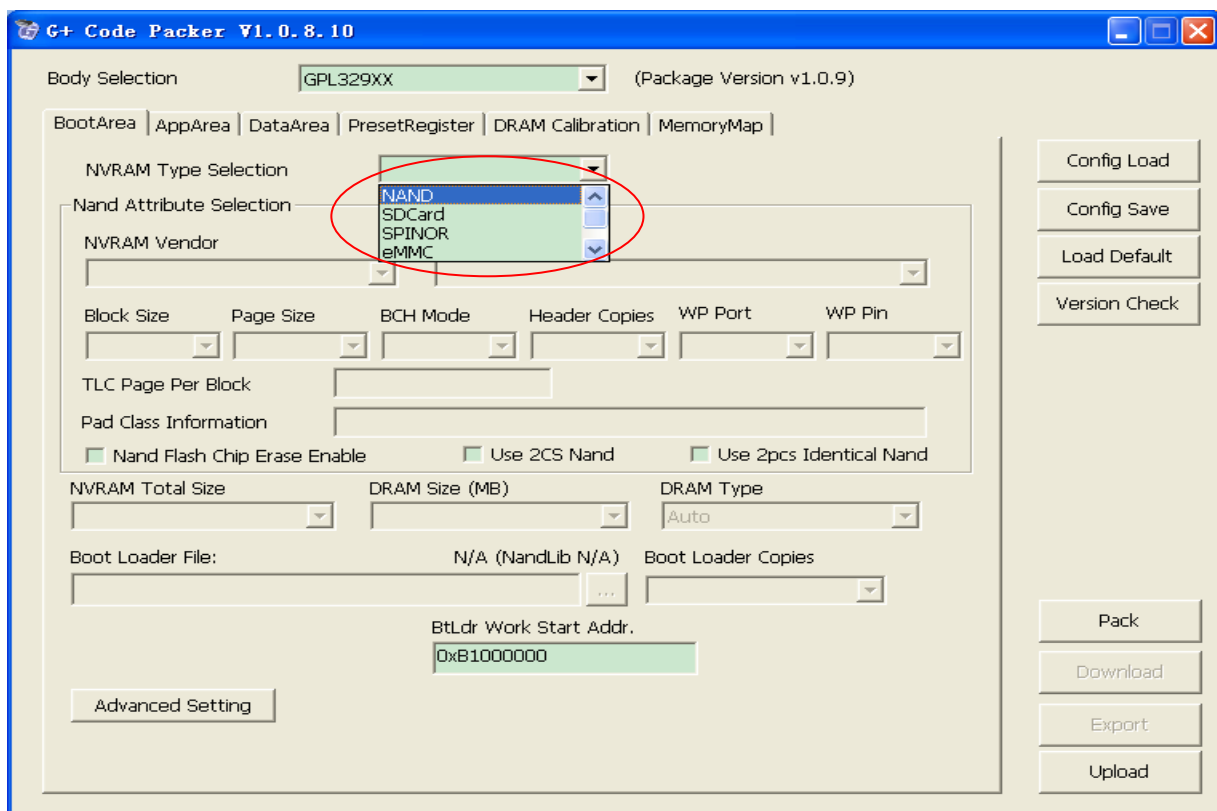
5.1 Load Config & Config Save

若先前已有做過 Config Save, 可透過 Config Load 方式將上次選擇之參數重新引入, Generalplus 強力建議所有之 Resource File, 全數放在 G+CodePacker 為 Root path 之路徑下 (可自行開 folder), 如此可對整體 G+CodePacker 之可攜性有顯著提升, 並可將使用者經驗透過 Config Save, 輕易轉交合作夥伴。

5.2 設置Boot Area

Step1 : NVRAM Type Selection

選擇 NVRAM(Non-Volatile RAM) 種類, 共支持 SPINOR, NAND 或 SD Card, 請選擇 NAND.



Step2 : Nand Attribute Selection

若 Step1 選擇的是 SPINOR, 則 Step2 可以跳過。若 Step1 選擇的是 NAND, 則 Step2 就需要填寫 Nand Attribute。包括: Nand Vendor ID (廠商 ID)、Nand Mark ID (產品 ID)、Nand Type、Page Size、Boot Area 的 BCH mode、Boot Area 的 Header 備份個數。

市面上 Nand Flash 大部分可以在 Vendor , Mark ID 查找到對應設置選項。如果用戶使用的 Nand 不包括在列表中,用戶就需要根據 Nand Flash 的 SPEC 填寫 Nand Type、Page Size 等設置(請於 NVRAM vendor 選擇 Manual Mode)。

除了 Manual Mode 外,透過 Nand Mark ID 之選擇, G+CodePacker 會貼心為您列出正確之 NVRAM Total Size, 此時該欄位不能竄改,而 SDRAM Size 和 DRAM Type 仍請使用者確實填寫,此乃重要參數,會影響系統之啟動能力與否,日後亦會以此參數做為 DRAM Calibration 之重要參考依據。

另外,目前 CodePacker 中所支持的 MLC 大 Page 的 Nand, 若其 Manual Mode 可以燒錄,則其 Auto Mode 也可以燒錄,且此種 Mode 更適合廠家量產。

Step3 : Boot Header Copies

填完 Mark ID 與 DRAM Size 和 DRAM Type 後,若有需要改寫 Header 之備份塊數,則可視情況增刪,若為優質之 Nand (非 re-use 之 Nand flash),建議保持原廠預設值即可。

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

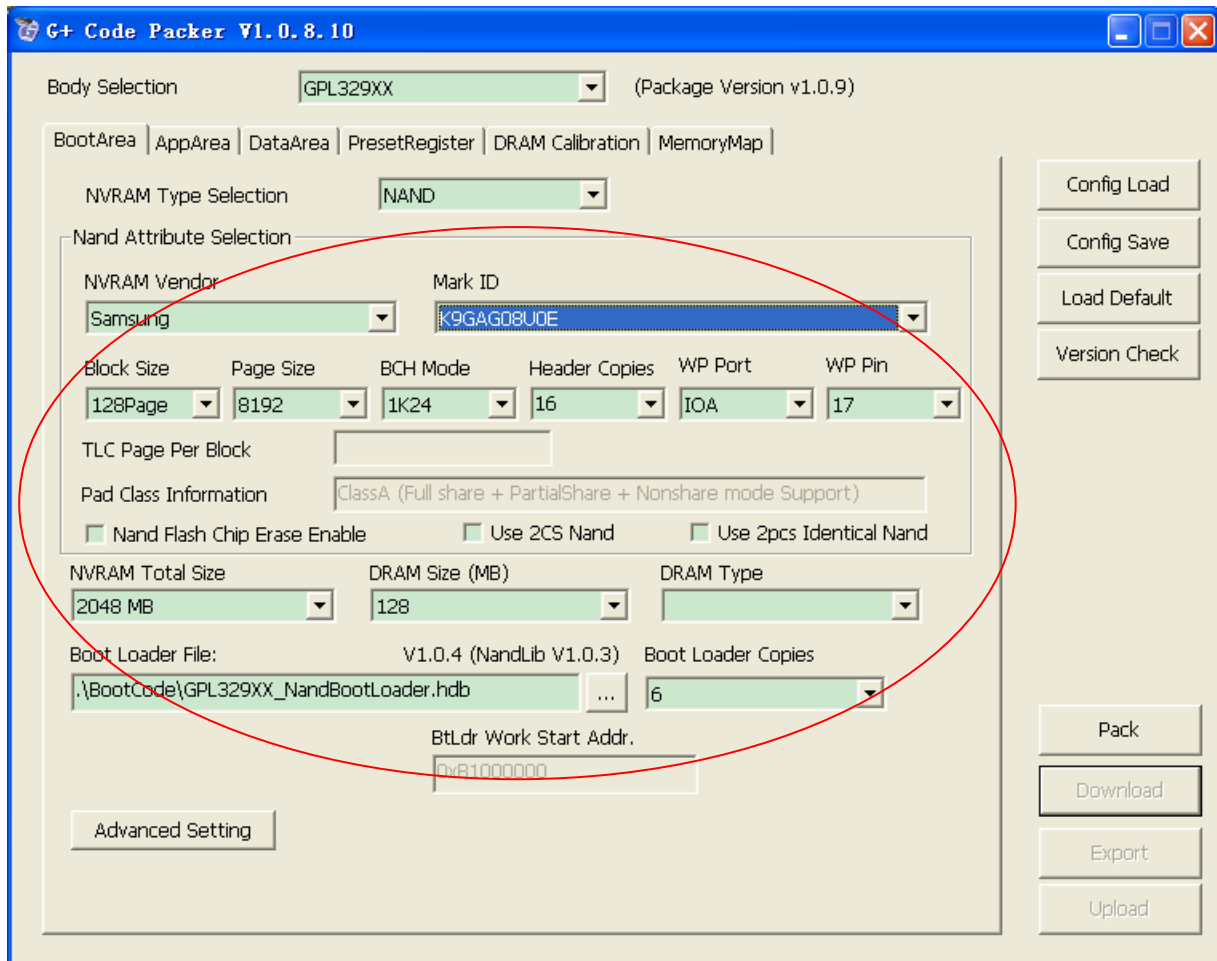
Write Protect Pin 亦為重要參數,在大多數開發板中,會將 Nand Write Protect pin 強制 pull high,此時該參數可設為 NULL,其餘請依線路設計來做設置,IOA17 為 Generalplus 原廠建議預設值,亦為 G+CodePacker 提供的預設值,請務必理解與確實設置。

Step5: Boot Loader File Choice

Boot Loader File 強力建議採用原廠預設,但特殊需求之客戶亦可自行提供 Boot Loader,該 Loader file 為 GPL329XX ROM Code 所會帶起之第一套程式,該程式將會帶入 Generalplus 管理區,包含放置客戶之 Runtime Code 或是 Read Only Resource 之 APP 區塊之初始化,還有 FAT 區之初始化皆在此完成,該程序對於使用 Generalplus 標準平台之使用者特別重要,且所內置之 Generalplus Standard Nand Driver Library 亦需與 Runtime code 所採用之 Nand Library 完全一致,始能保證 Nand 使用之正確性與穩定度。

Step 6: Boot Loader Copies

Boot Loader Copies 為 Boot Loader File 之備份數,以確保 Boot Loader 能完美帶出,因此 Generalplus 建議多做幾份備份。請視需求增刪 (BootLoader 已受 BCH 保護)



5.3 設置App Area

Step1：App Area 參數設定

當 NVRAM Type 為 Nand 時，App 參數才需要設置。如果 NVRAM Type 為 SPINOR，則 Step1 可以跳過。App 參數包括：App Man Size、App Spare Rate、App ROM Code Copies 以及 CRC Check Sum Enable。

App Man Size：

App Area 的規劃使用空間。注意：App Main Size 必須大於 Run Time ROM Code Size 和 Resource File Size 的總和。

App Spare Rate:

App Area 交換塊的比率。為了防止 App Area 出現壞塊造成資料丟失，Tool 設計了一定比率的交換塊。App Spare Size 為 App Man Size 與 App Spare Size 之換算結

果

$$AppSpareSize = \frac{AppMainSize}{AppSpareRate}, \text{ (unit: Nand Blocks) }。$$

例如:

Block Size = 64, Page Size=2048, App Main Size = 16MB (=128 blocks), App Spare Rate = 5, 那麼, App Spare Size = 128 blocks / 5 = 25.6 \div 26 blocks = 3.25MB

App Spare Size(以 block 為單位)最大值和最小值限制如下:

最大值 = Page Size / 4, 最小值一律限制為 4.

因此, 合法值必須滿足下列兩條件:

3. $4 \leq App\ Spare\ Size \leq Page\ Size / 4$, 單位為 block.
4. $App\ Spare\ Size \leq App\ Main\ Size$

以上述相同例子而言, 要同時滿足這兩條件, 以 block 為單位,

$$4 \leq App\ Spare\ Size \leq 2048 / 4$$

$$App\ Spare\ Size \leq 128$$

因此 App Spare Size 合法範圍為

$$4(=0.5MB) \leq App\ Spare\ Size \leq 128(=16MB) \text{ 意味著 } 1 \leq App\ Spare\ Rate \leq 32$$

所以合法可設定的 App Spare Rate 為 1 ~ 32, 對應的 App Spare Size 為 4 ~ 128 blocks (=0.5 ~ 16MB).

CRC Checksum Enable:

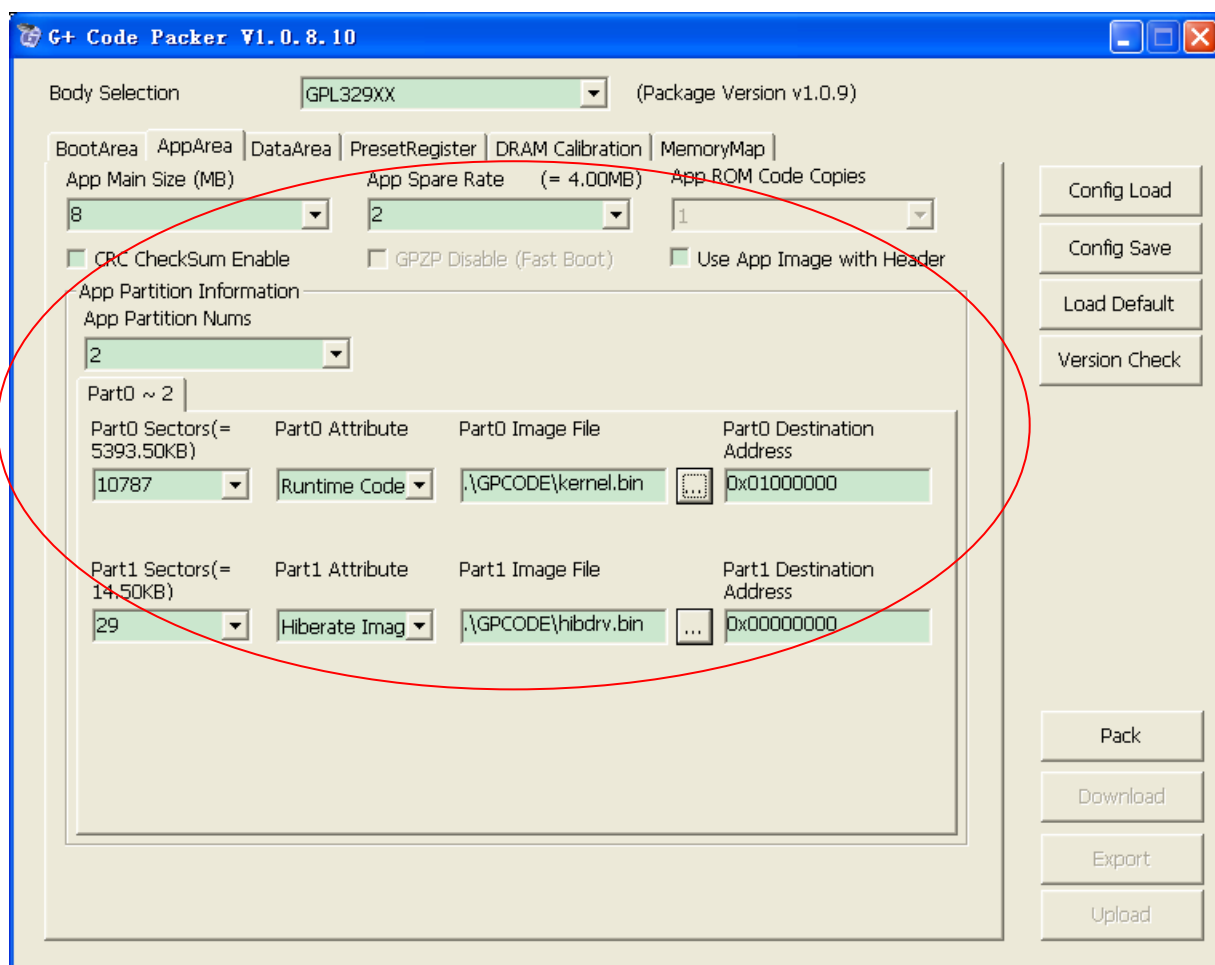
勾選 CRC Checksum Enable 時, CodePacker 將計算整個 App Area 的 CRC checksum 值紀錄在 App Header 裡, 可供 App Area 資料正確性校驗使用。

Step2 : Setting App Partition

App Area 可支持最多到 16 個 partition,每個 partition 可依用戶需求設定為 Runtime Code 或 Resource Part, 並由用戶自行載入 Run Time ROM Code 或 Resource Image File, tool 將根據所載入 code file 或 image file 自動計算每個 partition 的 size, 注意 App Main Size 需設定大於或等於所有 partition size 之總和, 否則 pack 時會出錯。

Boot Loader 執行起來之後會把使用者選擇的 Run Time ROM Code 複製到 DRAM 執行。Boot Loader 複製的大小將由 tool 根據 Run Time ROM Code 的實際大小計算得到。

注意：Run Time Rom Code Size 不得超過 DRAM 實際所能使用空間，否則程式可能會發生執行錯誤。



如上图，此页面主要是对 App Partition 进行配置，一是设定分区个数；二是为每个分区配置相应的参数。

App 分区信息参数配置如下：

- **App Partion Num:** 此选项表明用户当前设定的 App 区的分区数。目前设定范围 1~16，共 16 组值，也就是用户最多可配置 16 个分区。
- **Partion N Size:** N 的范围从 0~15，它是表明当前这个分区的容量大小，以 Sector 为单位，通常情况下，当我们选定分区镜像档案后，Tool 会自动帮我们将相应的档案大小填入此栏位。有一种情况是，用户可以不选取档案，此时，该栏位会被置为 auto，如果用户想指定此分区大小，可以直接输入该值。
- **Partion N Attribute:** 同理，N 的范围是从 0~15，它表明为当前镜像文件是什么类型的文件，目前 Tool 提供 Runtime Code, Resource Part, FastBoot Bin, Quick Image, Hibrate Image, Image Flag, Customize BtLdr 共 7 种类型的 App 镜像文件。当 First bootloader 在搬 App 区的 code 时，此参数就显得尤为重要。
- **Partion N Image file:** 此参数是让用户来选取磁盘上的分区镜像文件。
- **Partion N Destination Address:** 此参数是指定当前分区文件，将来会被摆放在 DRAM 哪个地址来跑。

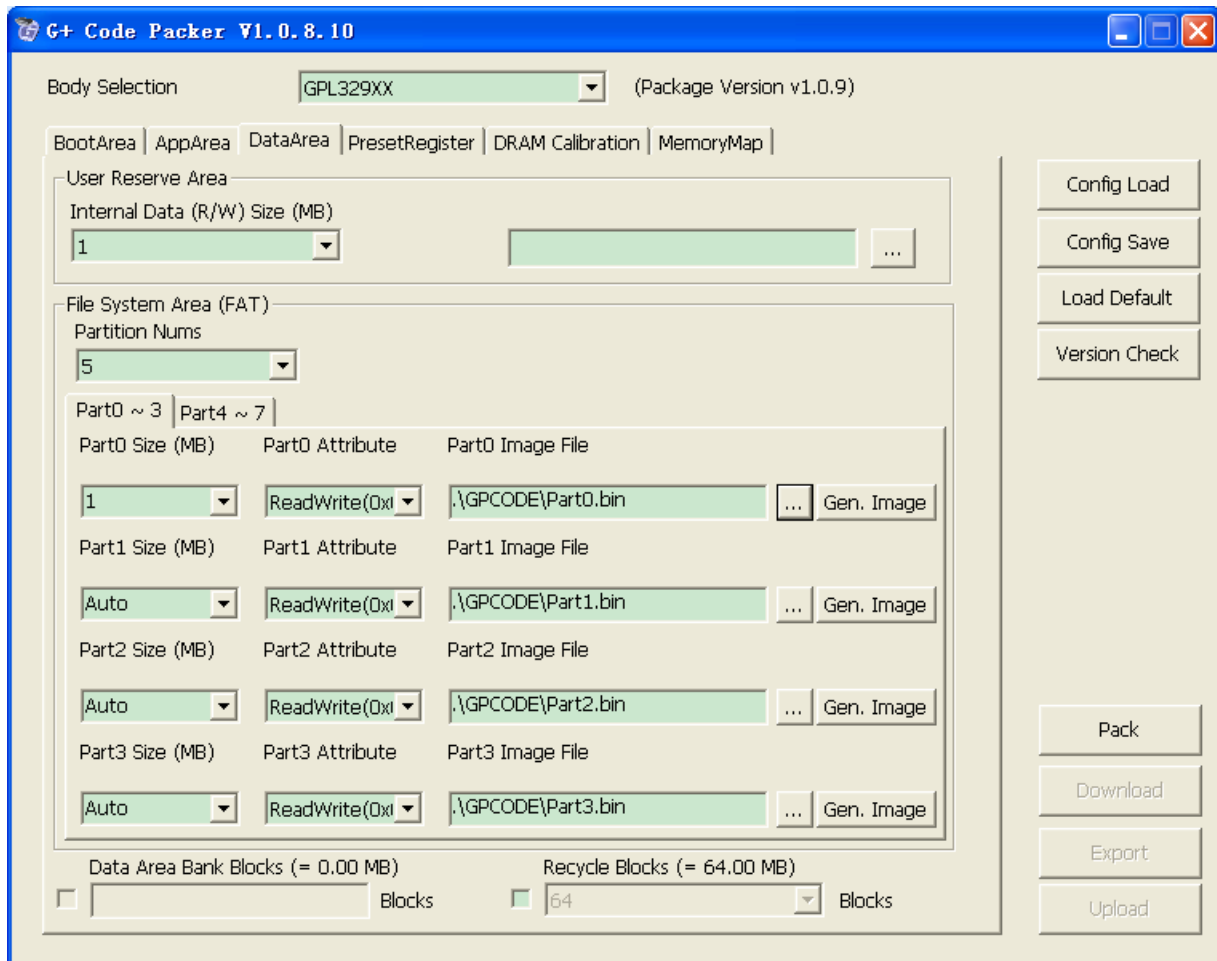
依此方法，对指定数目的分区进行配置，直到所有分区信息都配置完成。

5.4 設置Data Area及FAT Image Tool之使用方法

5.4.1 設置Data Area

Step1：設置 User Reserve Area Size 和 Image 檔案：

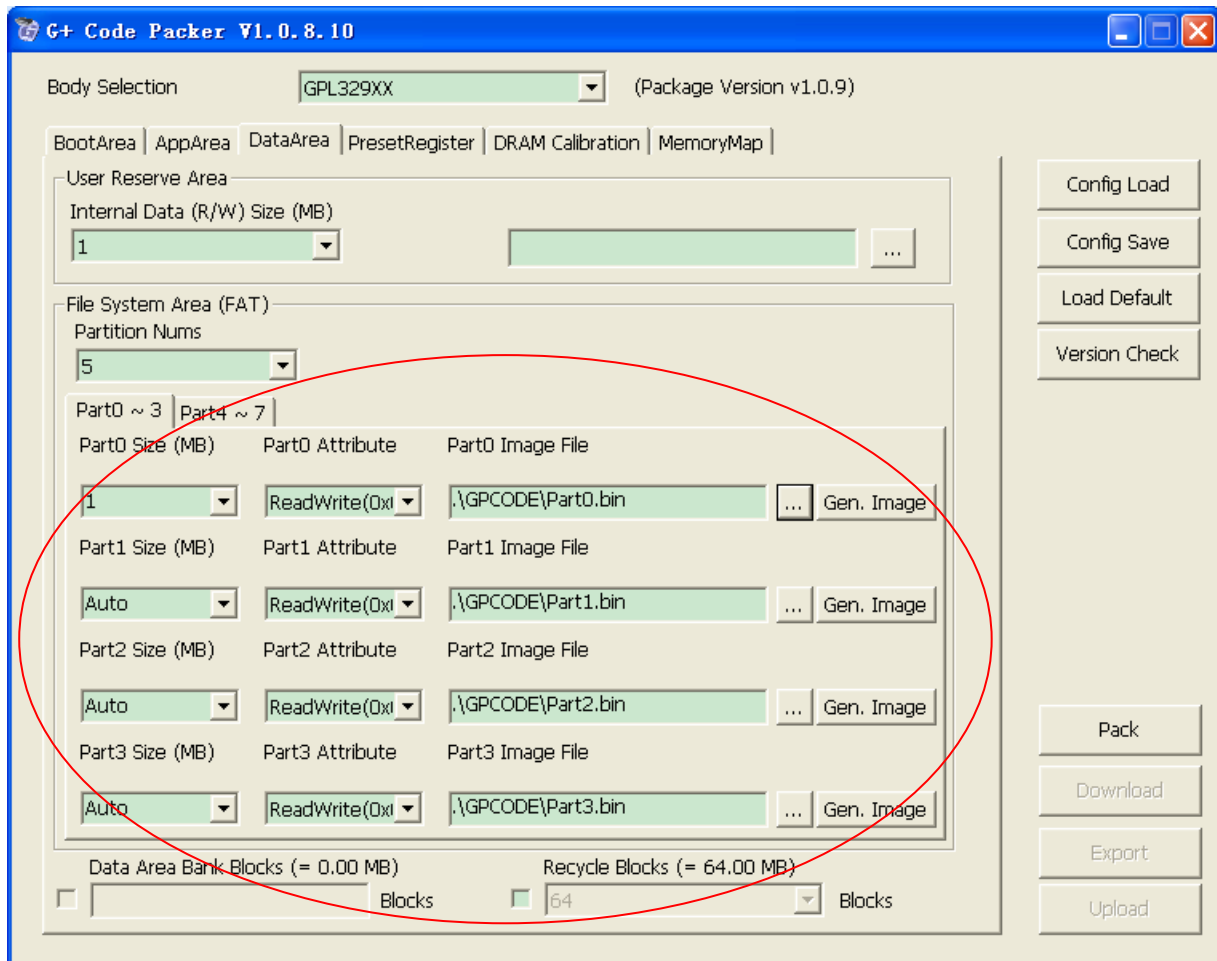
User Reserve Area 用於存放 App ROM code R/W 資料。User Reserve Area Size 的選擇以 MBytes 為單位。R/W 區可預載內容(請自行提供 pre-load file)，該內容可透過 Generalplus 提供之 APP Area 相關 API 作讀取動作。



Step2 : File System Area 設置

File System Area 分區(partition)可設置每個分區的 size 及對應 Image 檔案。無論 SPINOR, SDCard, 或 Nand Flash 最多支持八組的 File System 分區，使用者可依據應用需求自行規劃。

各區域的 size 可自行輸入規劃的大小，以 MBytes 為單位。若是選擇 Auto，則由 tool 依據選擇的 Image 檔案大小決定此區域的大小。Attribute 部分則有 Read Only、Write Only 以及 Read/Write 三種可供選擇。



Step3 : Advance Recycle Blocks (Option Choice) 設置

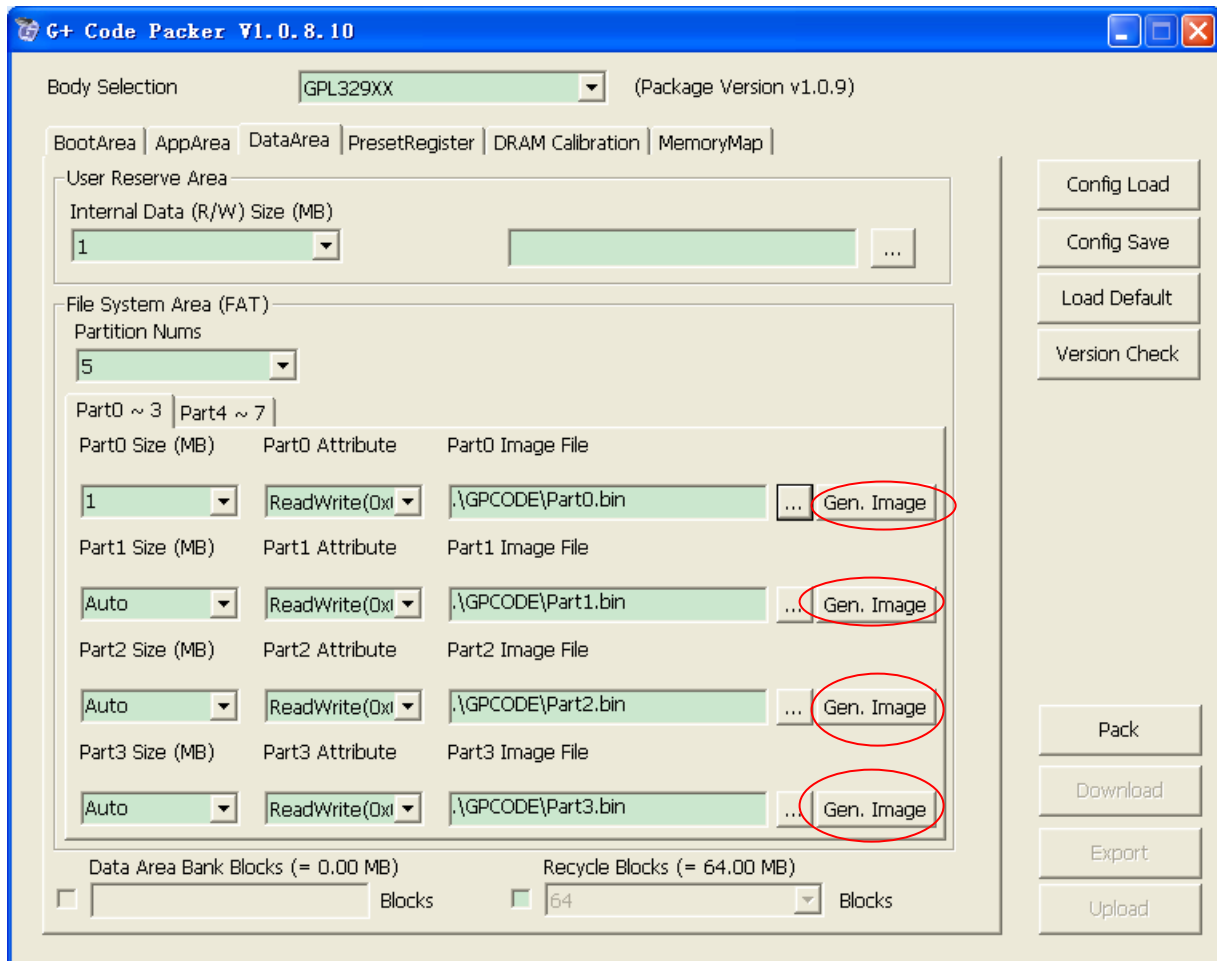
Generalplus 強烈建議採用原廠預設值，此為 Data Area manage exchange Block number for each Bank 之進階設置。而 Bank Size 預設為 512 Block，目前不開放設置。

5.4.2 FAT Image Tool之使用方法

File System Area 除了可以選擇已製作好的 Image 檔案，也整合了製作 Image 檔案的工具以方便用戶使用。操作流程如下所示：

Step1 : 執行 FAT Image Tool

由 FAT Image Tool 所製作出的 Image 會自動將檔名及 Partition Size 資訊帶回工具中。因此先選擇需要的 Partition Number，再點擊對應 Partition 後的“Gen. Image”按鈕，執行 FAT Image Tool，進行 Image 之製作。



另外，在数据区的配置过程中，有个 **Recyle Block** 的可选项进行设定。此项主要是用来告诉烧录程序，在烧写数据区时，如果发生坏块时，可以用此项预留的 **block** 来进行替换，默认值是 **64**，即为数据区准备 **64** 个好块，用于替换在数据烧写时遇到的坏块。用户可以勾选此选项进行 **Recyle block** 数目的修改，目前提供 **32/64/128** 三组值供用户挑选。通常情况下，不需要勾选，以默认值为参考即可。

Step2：設置 FAT Image Tool 相關選項並製作 Image 檔案

FAT Image Tool 執行後，需設置 **FAT Disk Size**、**Cluster Size**、**FAT Image Source Folder** 以及 **FAT Image Output Path** 參數。各參數的意義及設置注意事項如下：

1. FAT Disk Size：

設置產生的 **Image** 檔案之 **Partition Size**。這裡設置的值必需大於等於所選擇 **Source Folder** 的 **size**，否則在產生 **Image** 的過程會失敗。

2. Cluster Size：

設置 **Cluster Size**。此設置建議使用 **Tool** 預設值(**Auto**)，若此設置錯誤，可能會造成 **Image** 生成失敗或是產生出來的 **Image** 檔案無法正確被辨識。

3. FAT Image Source Folder：

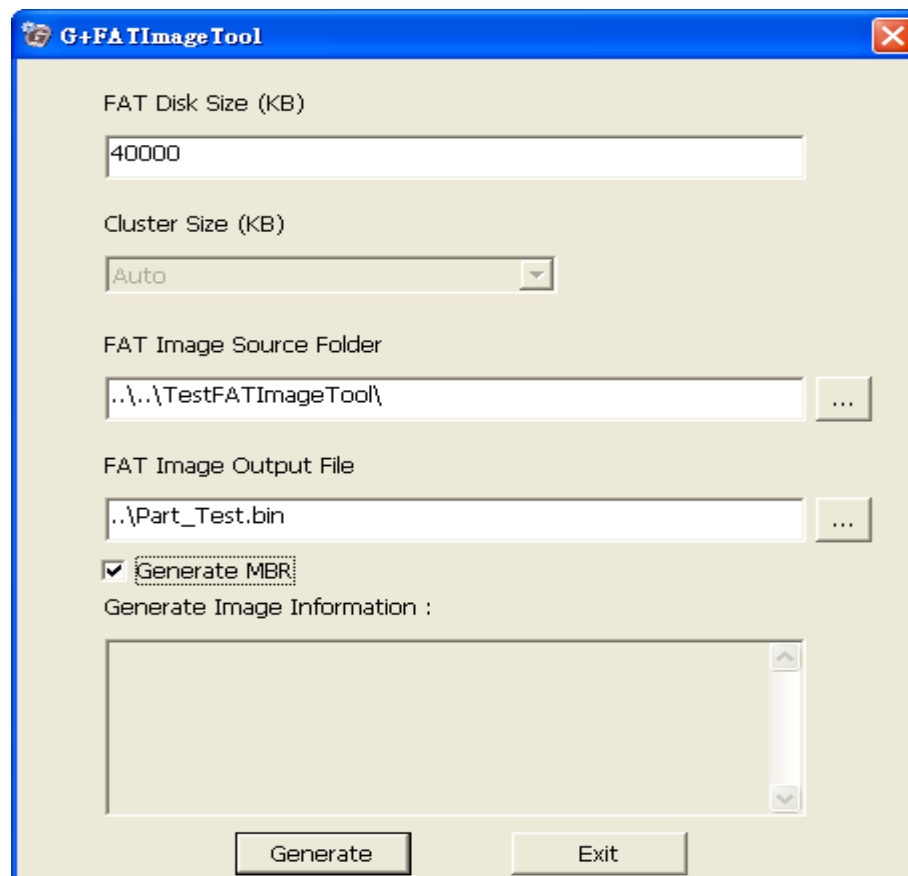
設置用來產生 Image 檔案的來源目錄。工具會將設置的目錄製作成 Image 檔案，包含所有目錄下的檔案、目錄結構以及子目錄下的檔案。

4. FAT Image Output Path :

設置產生的 Image 檔案之路徑與檔名。

5. Generate MBR :

設置產生的 Image 檔案是否要帶 MBR 資訊。需 MBR 請勾選。

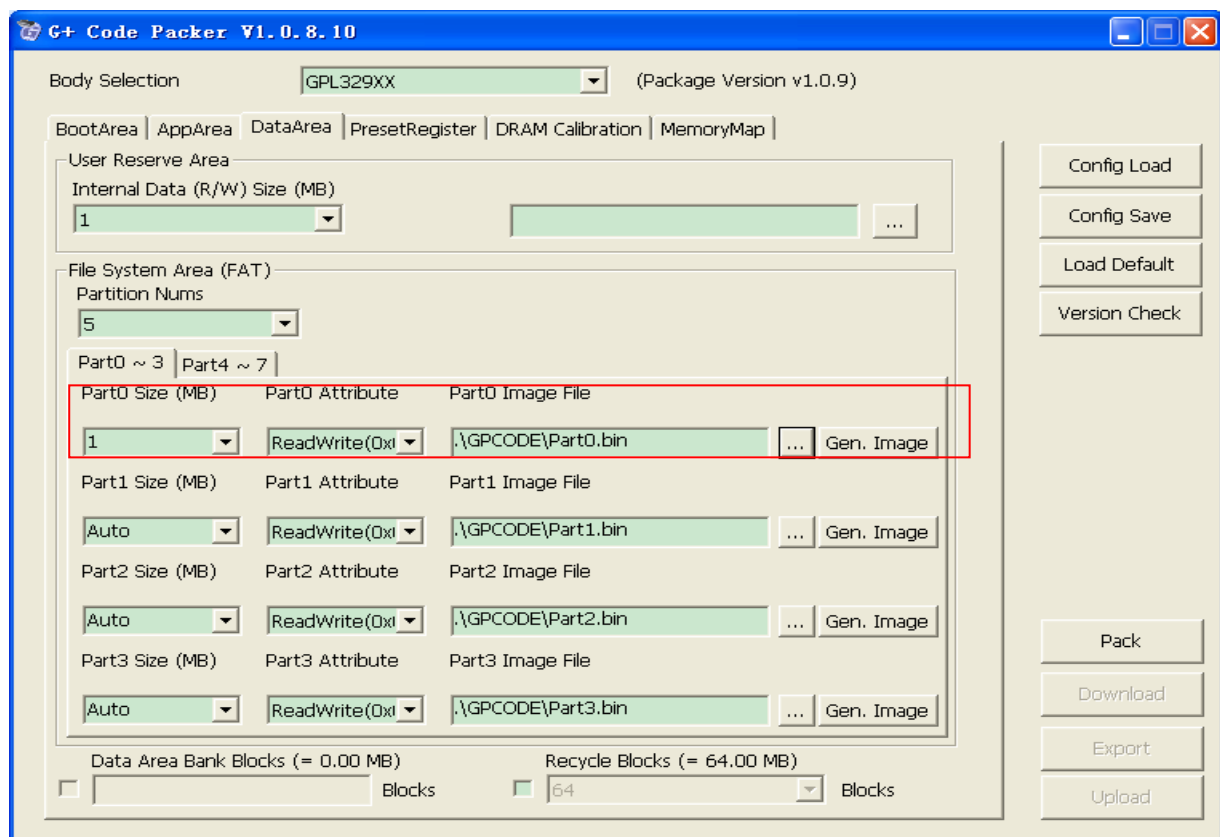


設置完成後點擊"Generate"按鈕，開始進行 Image 檔案之製作。製作過程中可由 "Generate Image Information"視窗看到目前的製作進度，製作到哪個目錄及哪個檔案。

Note: 繁體中文作業系統下的來源檔目錄不能有簡體檔名或簡體子目錄名存在，而簡體中文作業系統下的來源檔目錄下不能有繁體檔名或繁體子目錄存在，否則 **image file** 製作會失敗。

Step3：使資訊自動帶入 Code Packer 中

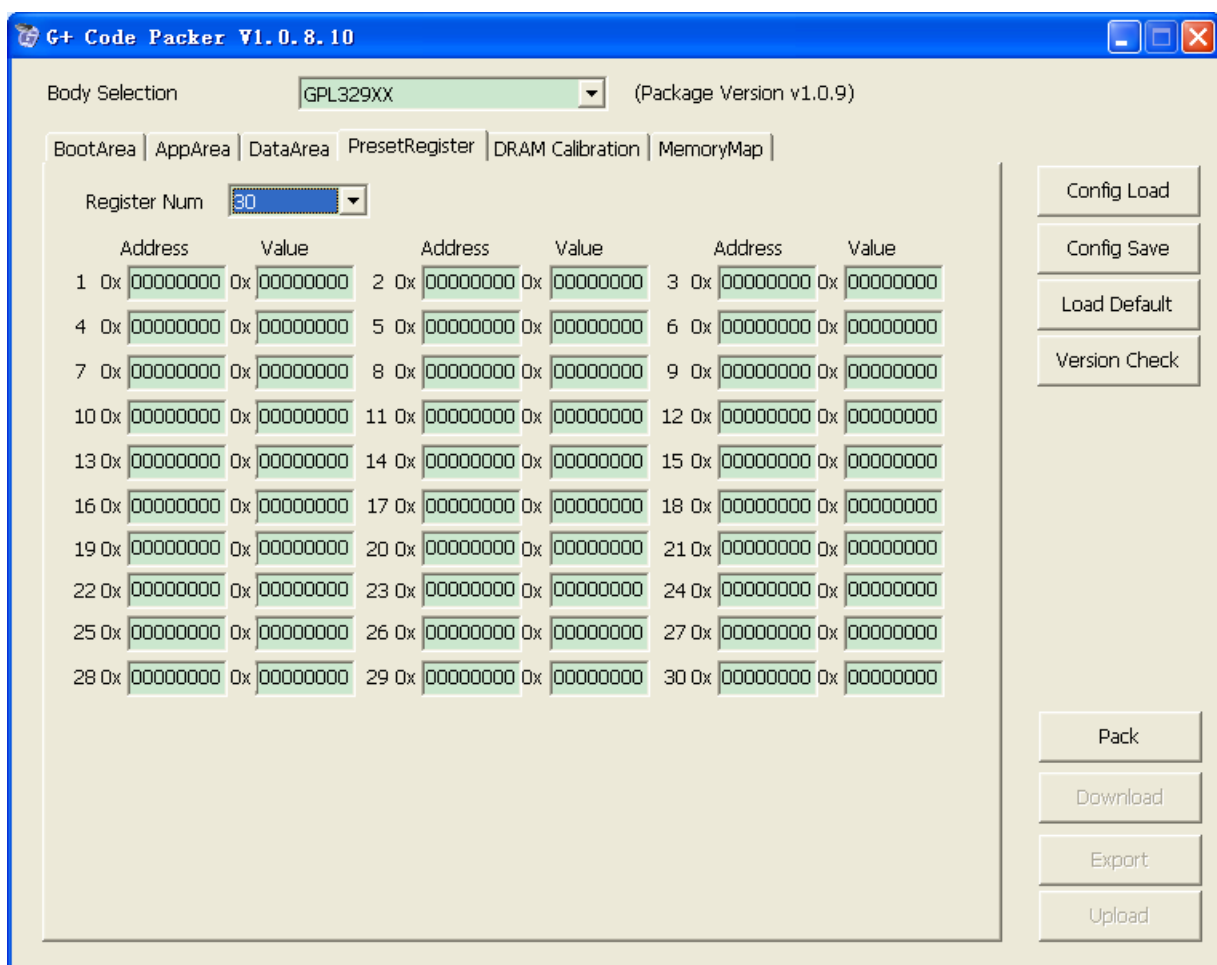
Image 檔案製作完成，會彈出"Generate FAT Image Successfully!!"視窗，點擊確定後即可關閉 FAT Image Tool，資訊會自動帶入 Code Packer 中，即完成製作 Image 動作。



5.5 設置Pre-set Register及DRAM Calibration之使用方法

5.5.1 設置Pre-set Register

使用者可依據本身需求設置相關的 Register 值。SPI Flash 只支持 10 組 Register 設定，SDCard 支持 23 組 Register 設定，而 Nand Flash 支持 30 組的 Register 設定。每一組設定皆需設置 Register Address 與 Register Value。

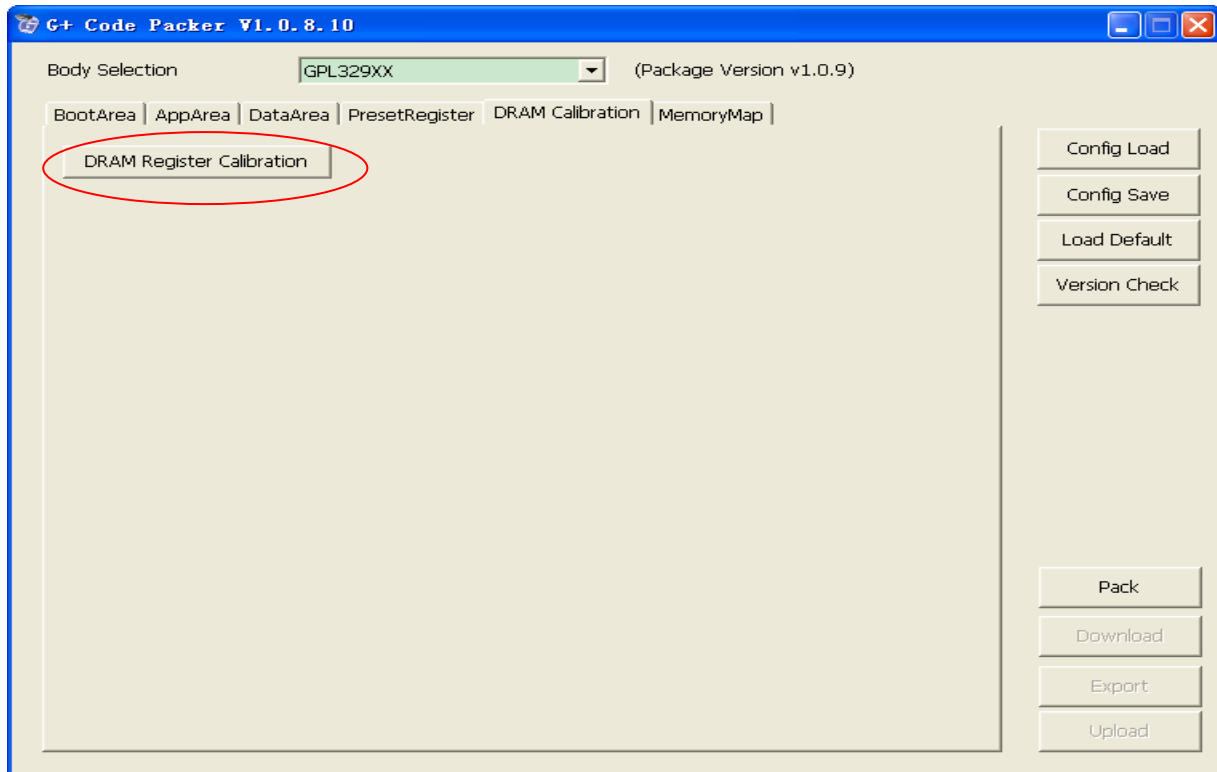


5.5.2 DRAM Calibration的使用

Step1 : Click DRAM Register Calibration button

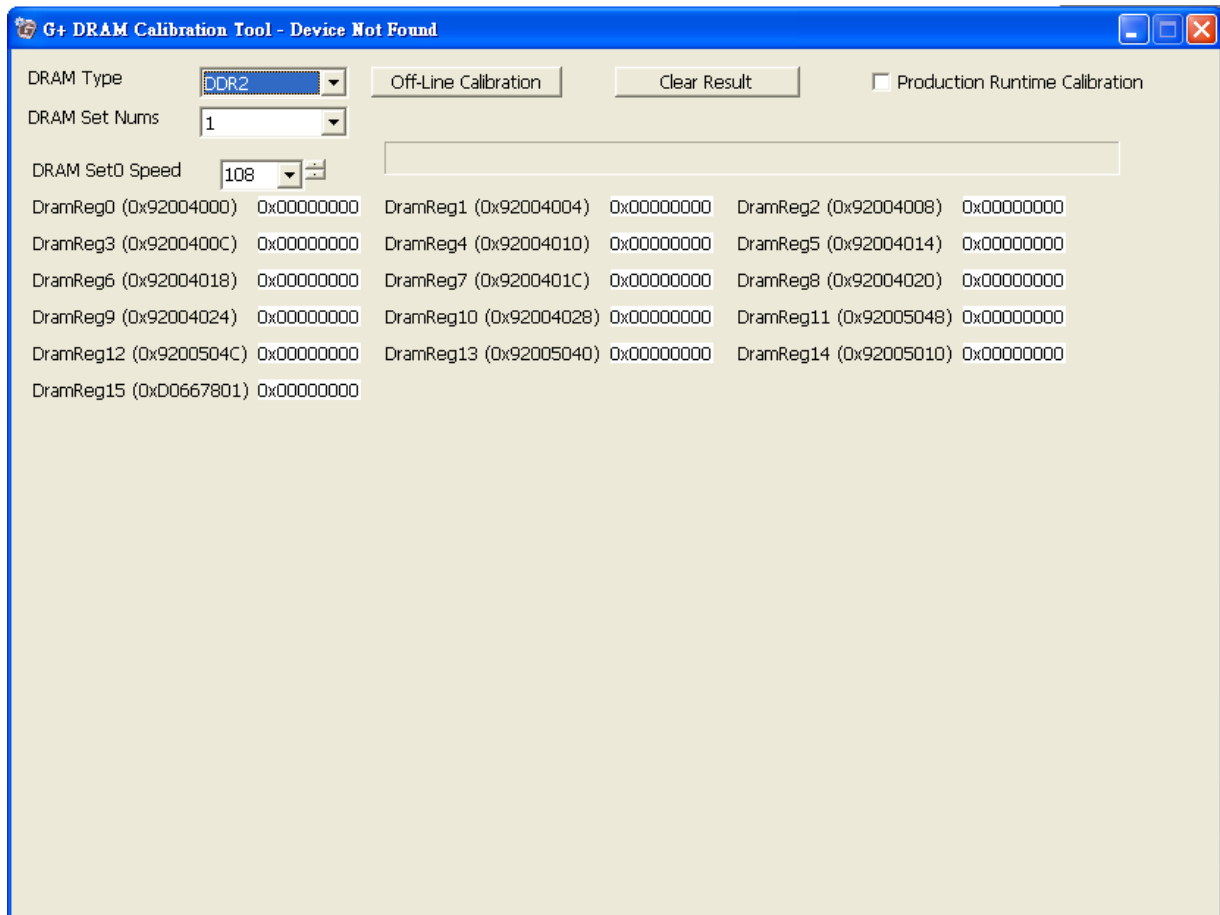
點擊 DRAM Register Calibration 按鈕，喚起 DRAM Calibration Tool。硬體進入 Mask Rom Code 連接電腦。Tool 若識別到硬件，則會在標題列顯示 Device Connected，否則顯示 Device Not Found。所有 Calibration 操作必須在 Device Connected 狀態下進行。

請由 Internal ROM Code Boot 進入 USB Service Mode.



Step 2. Entry Calibration Tools (sub tool)

請選定 DRAM Type, 設置 DRAM Set Nums 和 DRAM speed。



Step3 : 點擊 Off-Line Calibration。

Step4 : Calibration Success 並會得到一組校正過的 DRAM 參數。

註(重要) : 完成 DRAM Calibration 動作請拔除 GPL329XX 整體電源 (or 做 Hardware reset)。

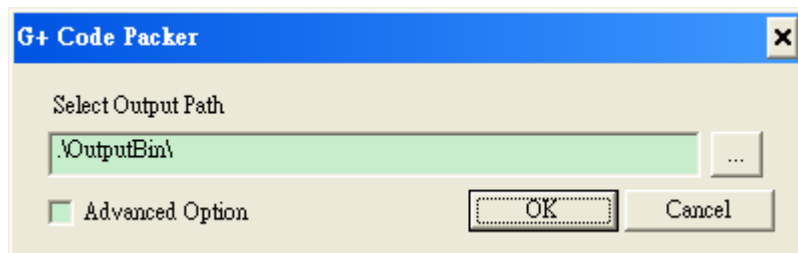
Step5 : DRAM 參數帶入 Pre-Register 中

關掉 DRAM Calibration Tool，Tool 會自動將取得之參數自動帶入 G+ Code Packer 之 Pre-set Register 中。此自動帶入的動作，會固定將 DRAM 參數放在前 10 組 Register 設定中，若前 10 組 Register 設定已存在，則原先設置將會遞移至 DRAM 參數後依序排列。若想清除 DRAM 參數，回復原先設置，可使用 Clear Result 按鈕。此按鈕會清除 DRAM 參數，若未進行 DRAM Calibration，則此按鈕將沒有作用。

5.6 Pack與Download

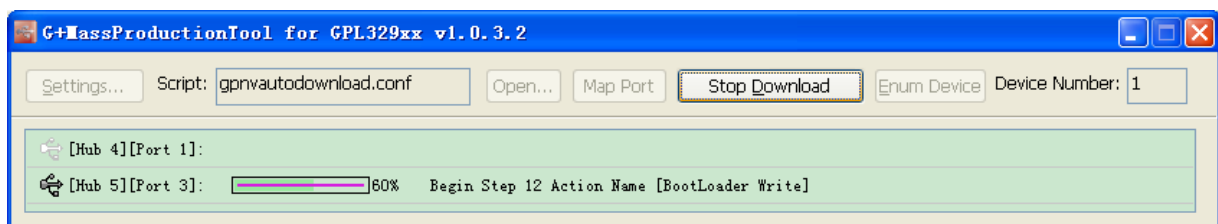
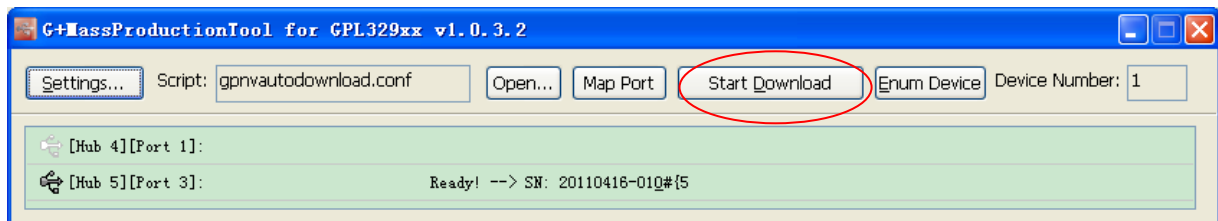
Step 1 : Pack

設置完各區域後，可以照需求點擊 **Pack** 按鈕進行整合動作。此時 tool 會依據設置將所有檔案整合至最終的 BIN 檔當中，並將 BIN 檔放置於使用者選擇的目錄下(建議使用預設路徑)。



Step 2 : Download

Pack 動作完成後，Download 按鍵會被激發。點擊 Download 按鍵，tool 會喚起 USB Mass Production Tool，並設置 Download 參數。使用者只需將裝置以 USB 連接電腦，當 USB Mass Production Tool 辨識到裝置後，按下 **Start Download** 按鍵，即可進入 Download 程序。



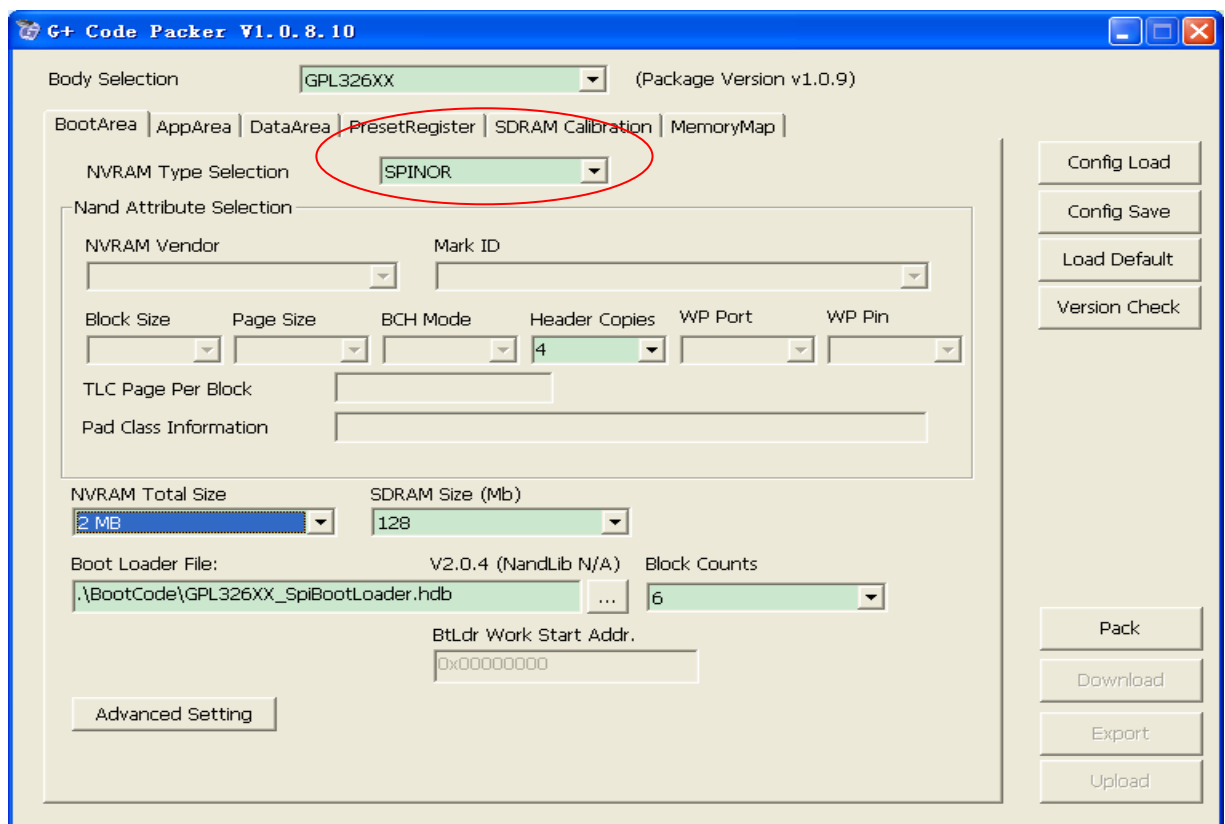
6 SPI Flash Boot操作流程解說

G+ Code Packer 將 SPI Flash 分為 3 個操作區域：Boot Area、App Area 以及 Data Area。使用者可分別設置這三個區域的資源檔來進行整合。另外，Tool 支援 Pre-set Register 可以調用 SDRAM Calibration Tool 測試 SDRAM 參數，並把 SDRAM Register 參數整合到燒錄文件中。

接下來各個子章節將依序介紹 Boot Area、App Area、Data Area、Pre-set Register 設置、SDRAM Calibration、Pack 以及 Download 的操作流程及使用方式。

6.1 設置Boot Area

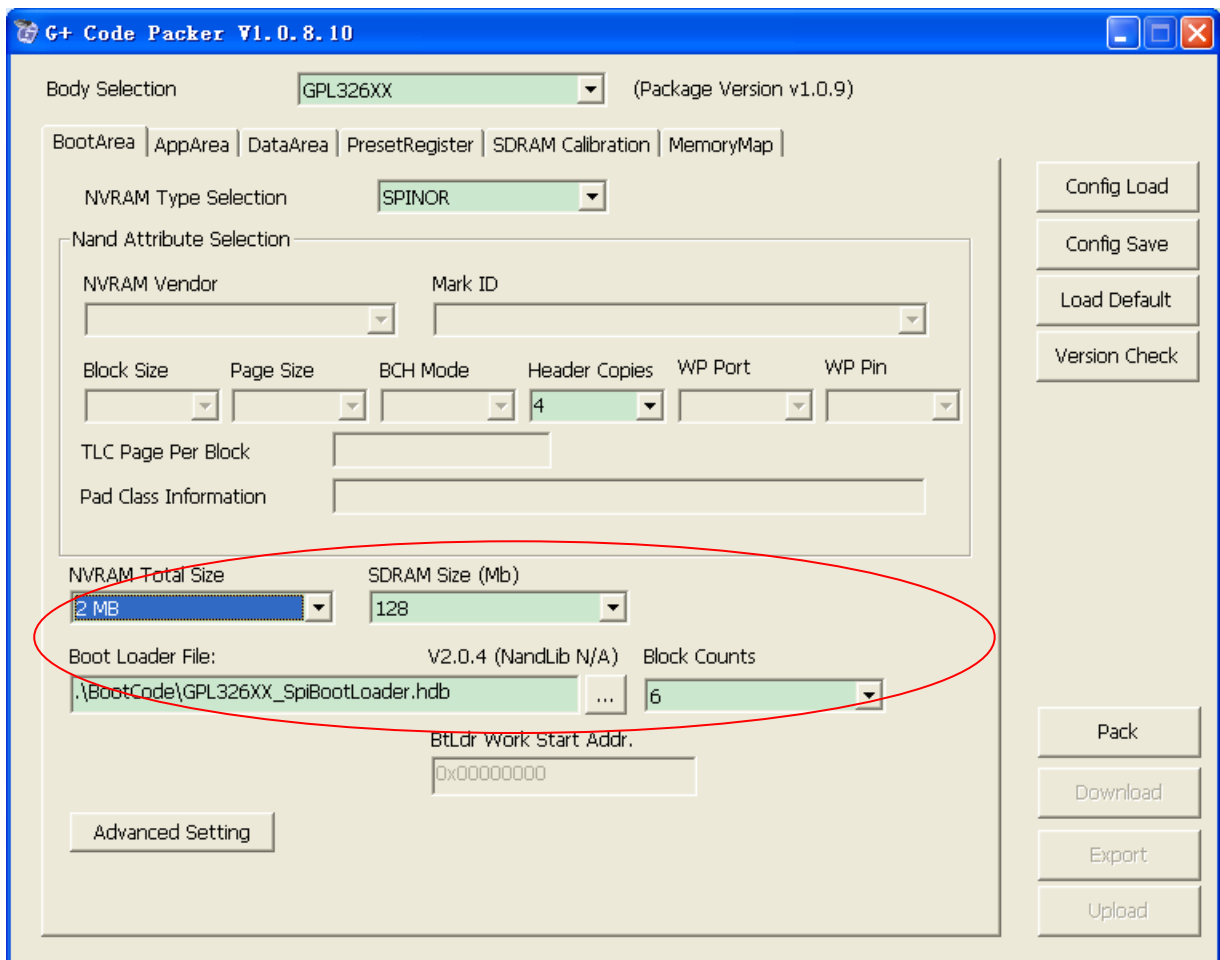
Step1：NVRAM Type Selection -- SPINOR。



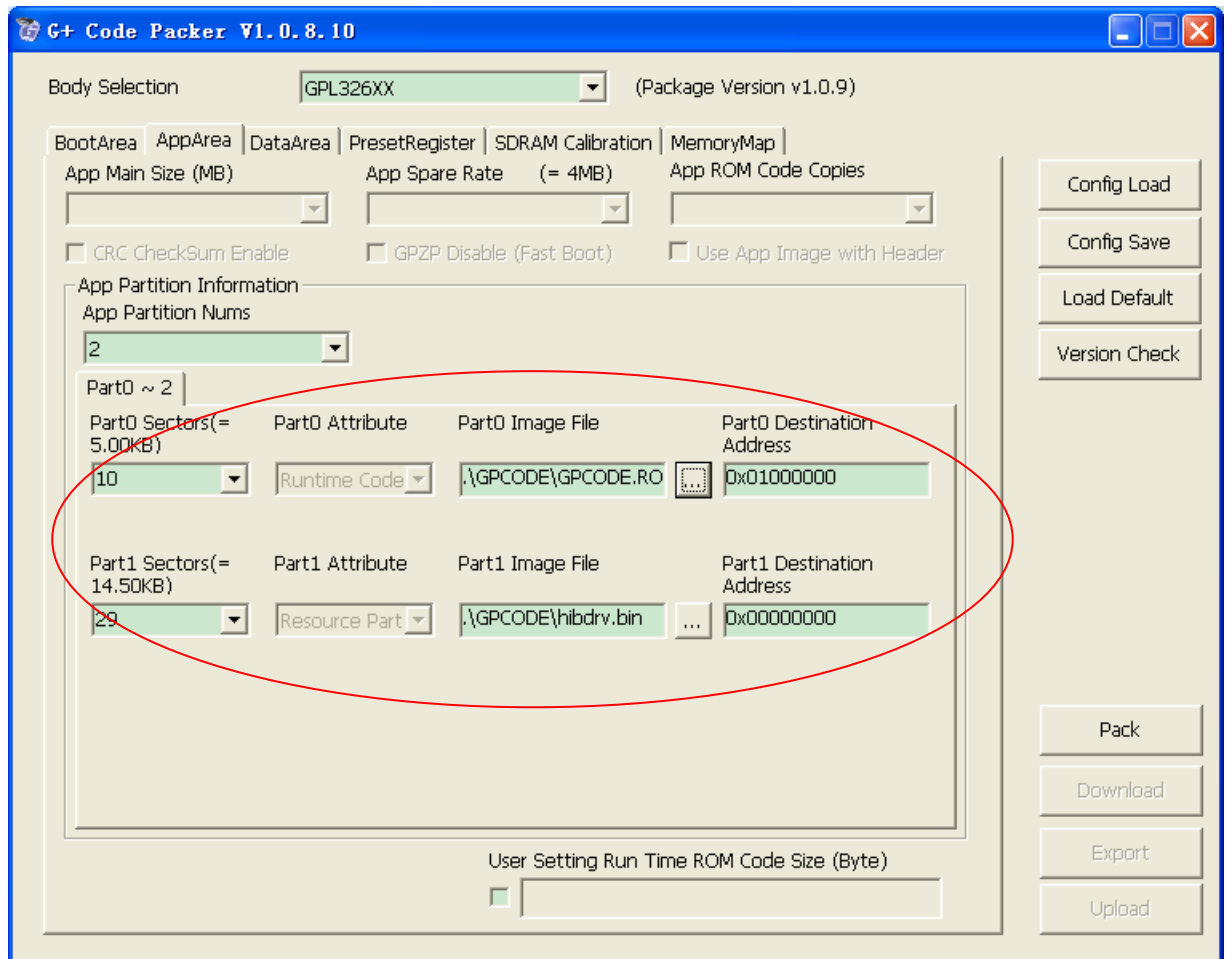
Step2 : NVRAM Size、SDRAM Size 以及 Boot Loader File 的選擇

NVRAM Size 及 SDRAM Size 的選擇需要注意標示的單位。選擇 Flash 的 size 單位是 MBytes，而選擇工程使用的 SDRAM Size 單位是：Mbit。

SPI Flash 和 Nand Flash 將使用不同的 Boot Loader File，如果 NVRAM Type 為 SPI Flash，則不需要設置 Boot Loader 要備份多少個 Block (Block Counts)。原則上，Tool 會依據選擇的 NVRAM Type 帶入預設的 Boot Loader File，Generalplus 建議使用 tool 內建之 Boot Loader File。



6.2 設置App Area



如上图，此页面主要是对 **App Partition** 进行配置，一是设定分区个数；二是为每个分区配置相应的参数。

App 分区信息参数配置如下：

- **App Partion Num:** 此选项表明用户当前设定的 **App** 区的分区数。目前设定范围 1~2，共 2 组值，也就是用户最多可配置 2 个分区。
- **Partion N Size:** N 的范围从 0~1，它是表明当前这个分区的容量大小，以 **Sector** 为单位，通常情况下，当我们选定分区镜像档案后，**Tool** 会自动帮我们将相应的档案大小填入此栏位。有一种情况是，用户可以不选取档案，此时，该栏位会被置为 **auto**，如果用户想指定此分区大小，可以直接输入该值。
- **Partion N Attribute:** 同理，N 的范围是从 0~1，它表明为当前镜像文件是什么类型的文件，目前 **Tool** 提供 **Runtime Code**, **Resource Part** 共 2 种类型

的 App 境像文件。当 First bootloader 在搬 App 区的 code 时，此参数就显得尤为重要。

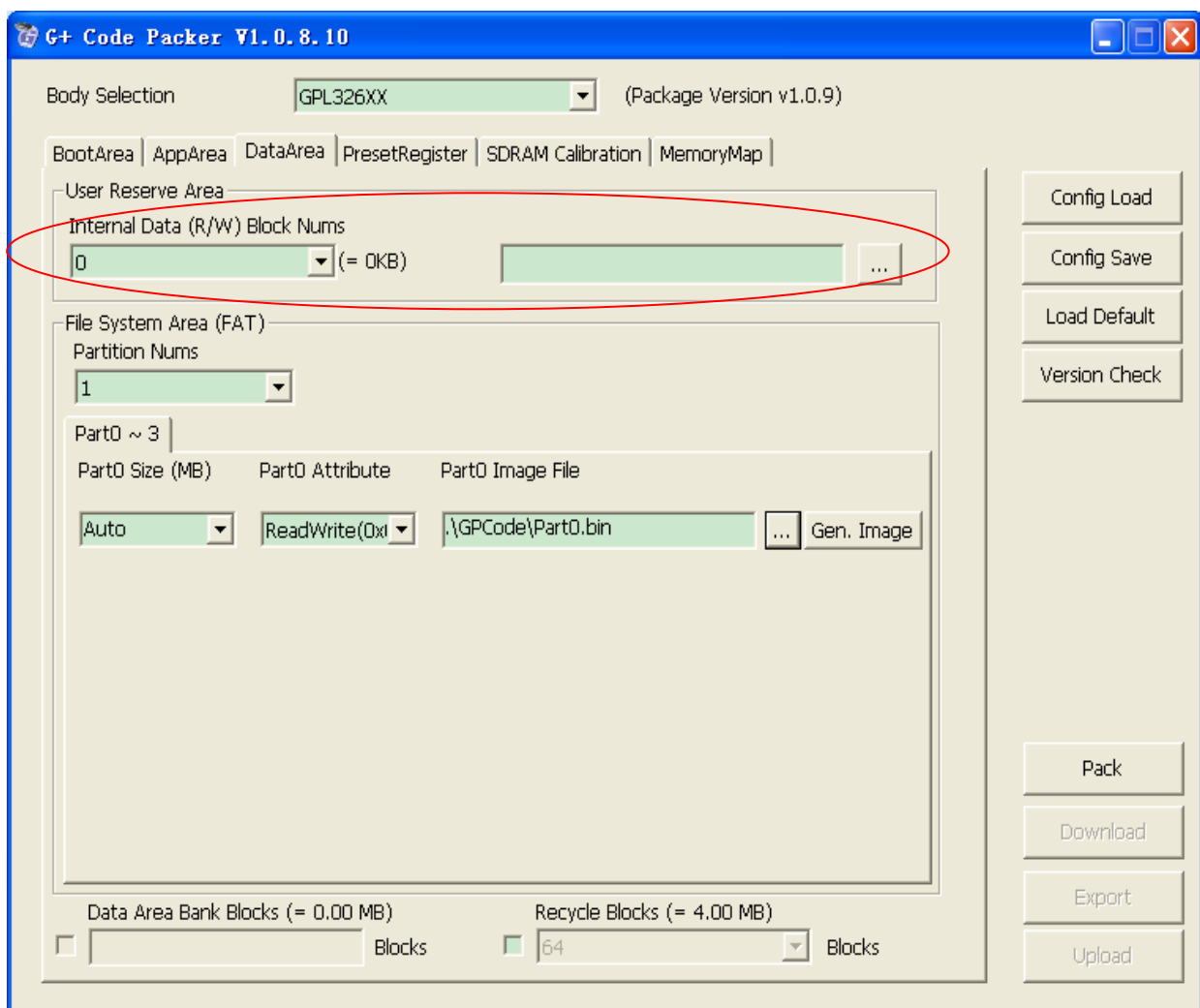
- Partion N Image file: 此参数是让用户来选取磁盘上的分区境像文件。
- Partion N Destination Address: 此参数是指定当前分区文件，将来会被摆放在 SDRam 哪个地址来跑。

依此方法，对指定数目的分区进行配置，直到所有分区信息都配置完成。

6.3 設置Data Area

Step1：設置 User Reserve Area Size 和 Image 檔案：

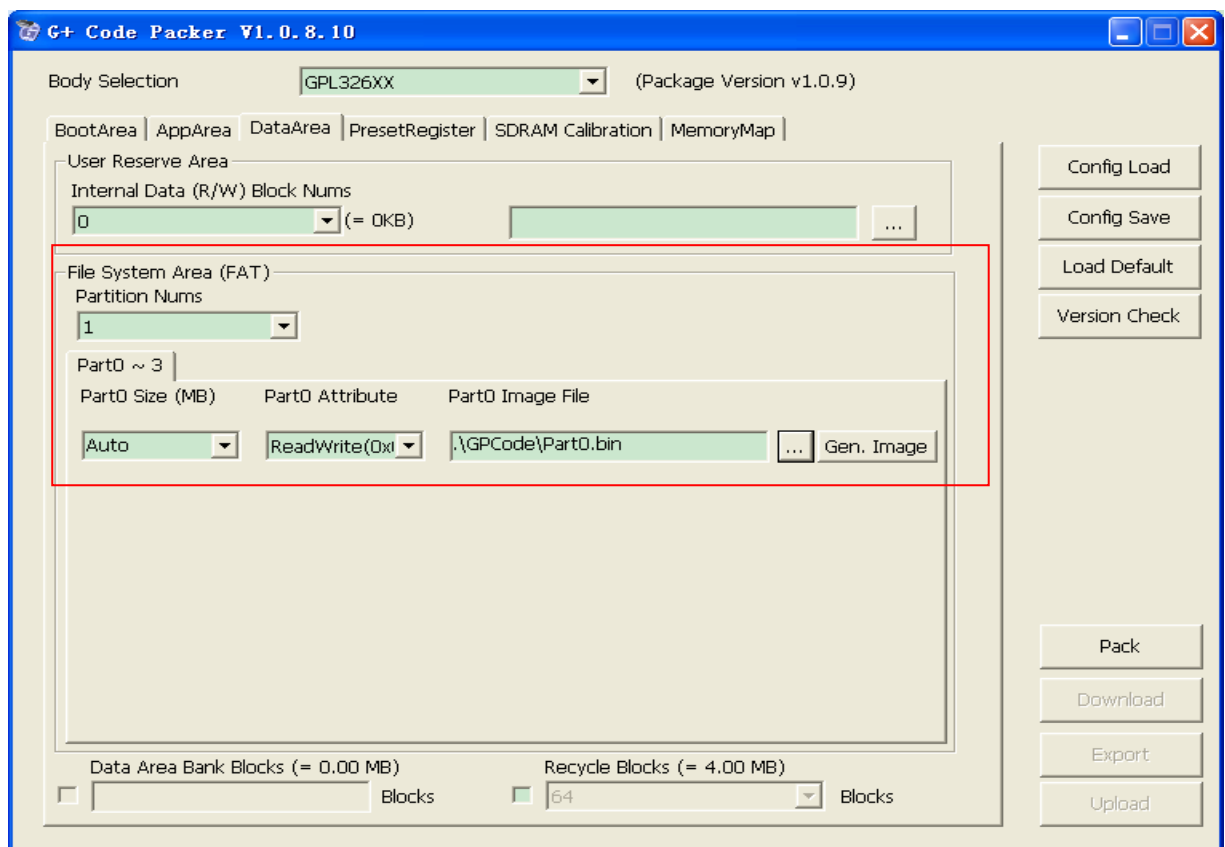
User Reserve Area 用於存放 App ROM code R/W 資料。User Reserve Area Size 的選擇以 Block 為單位，選擇完畢會在選項後方顯示換算出的 Reserve Area Size 以 KBytes 表示，方便使用者了解實際規劃的大小。



Step2 : File System Area 設置

File System Area 分區，設置每個區域的 size 及對應 Image 檔案。SPI 支持一組的 File System 分區 (FAT12 format)，使用者可依據需求規劃。

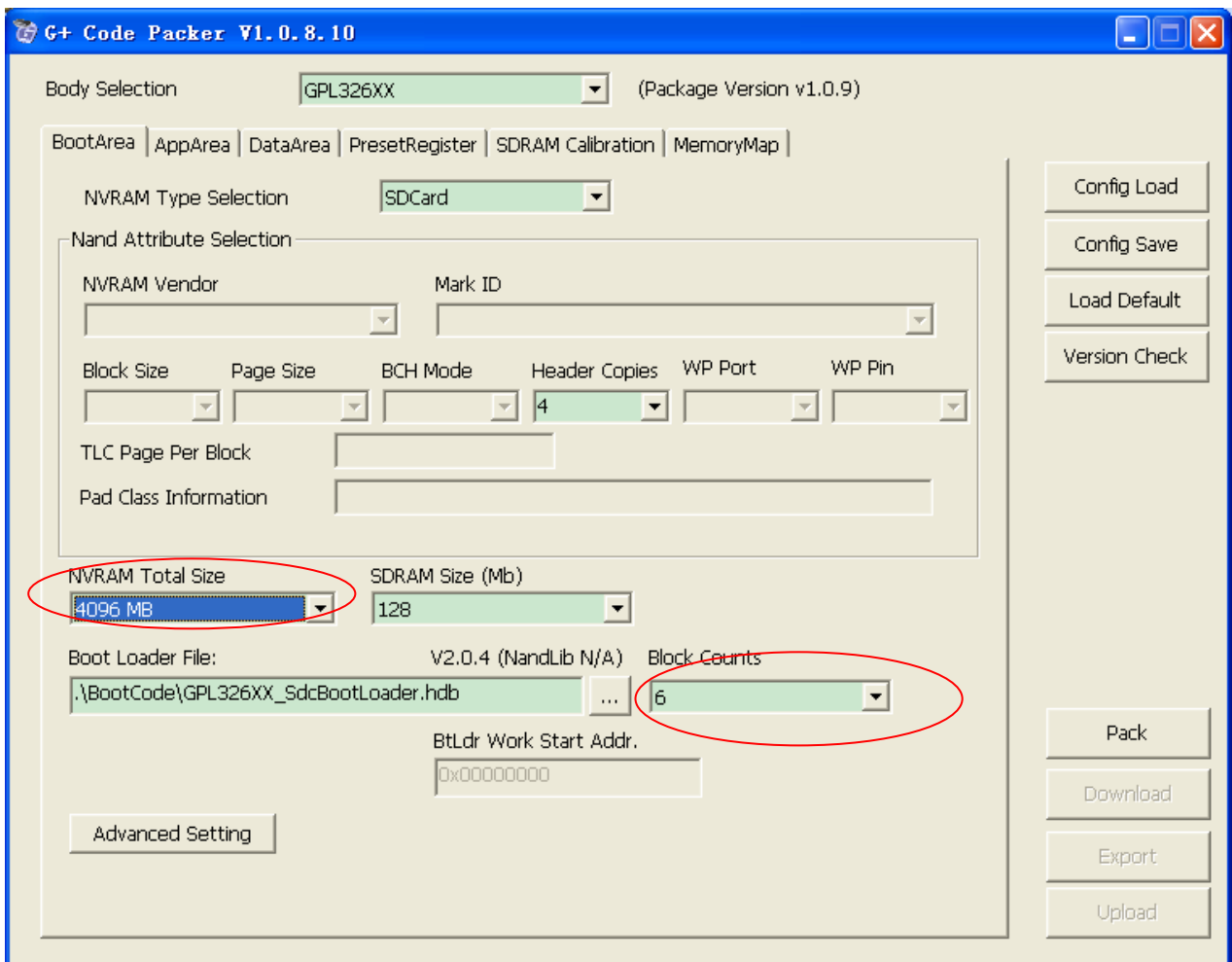
各區域的 size 可自行輸入規劃的大小，以 MBytes 為單位。若是選擇 Auto，則由 tool 依據選擇的 Image 檔案大小決定此區域的大小。Attribute 部分則有 Read Only、Write Only 以及 Read/Write 三種可供選擇。



7 SDCard Boot操作流程解說 (GPL326XX)

GPL326XX 將 SDCard 區分為 3 個操作區域：Boot Area、App Area 以及 Data Area。使用者透過 G+CodePacker 可分別設置這三個區域的資源檔來進行整合。SDC Boot 為開發者模式，使用方式與 SPI Nor Flash 完全相同（請參考第六章使用說明）。

若遇到 SDC Boot 相容性問題，Generalplus 建議打開 Advance Mode (勾選 Advance Setting) 設置 SDC Clock 除頻將 SDCard 降速操作。



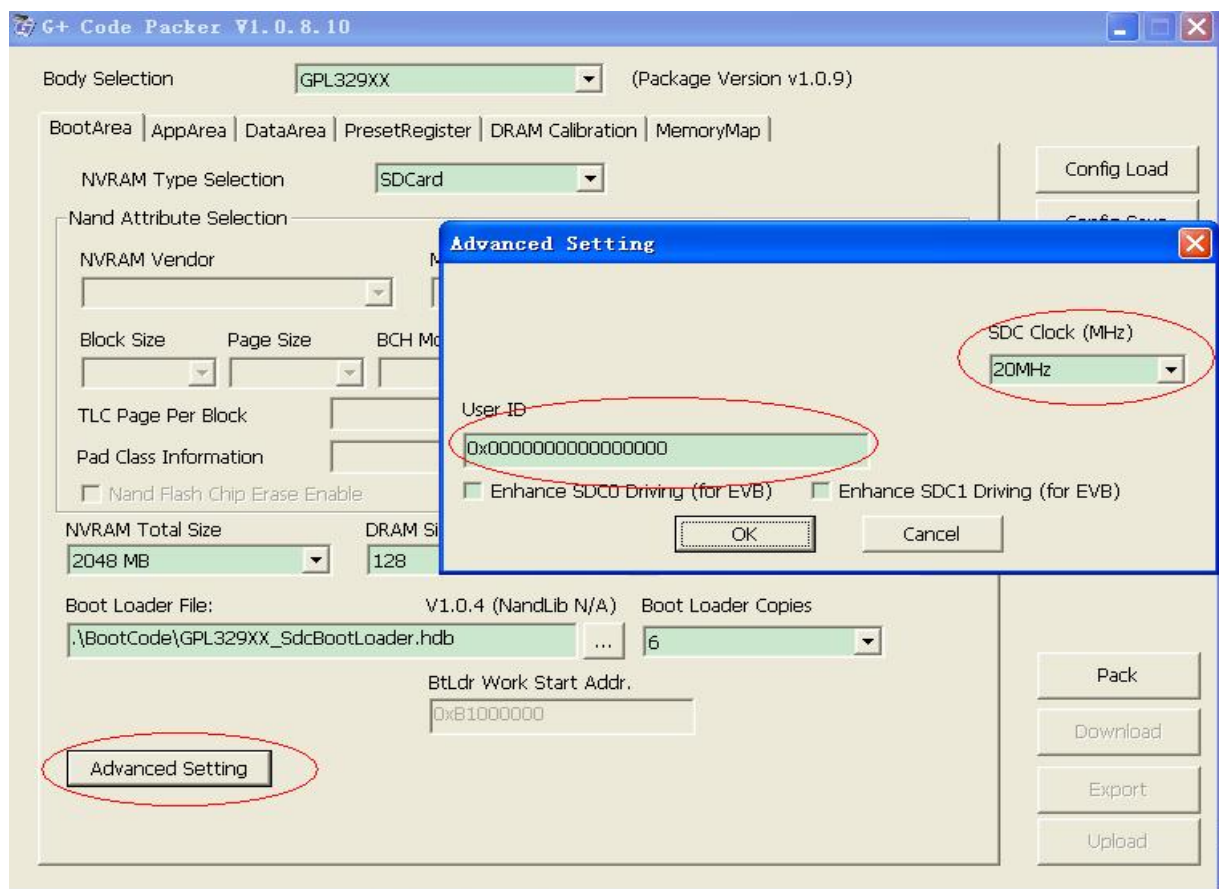
8 SDCard Boot操作流程解說 (GPL329XX)

GPL329XX 將 SDCard 區分為 3 個操作區域: Boot Area、App Area 以及 Data Area。使用者透過 G+CodePacker 可分別設置這三個區域的資源檔來進行整合。

若遇到 SDC Boot 相容性問題, Generalplus 建議打開 Advance Mode (勾選 Advance Setting) 設置較低 SDC Clock 將 SDCard 降速操作, 如果 Target Board (例如: Generalplus EVB) 的負載較重可以勾選 SDC driving 增強。

注意:

3. SDCard 使用前, Generalplus 建議應先備份好原始資料檔到其它 disk.
4. SDCard 使用後, 在 PC 上若發現卡容量變小, 可使用任何格式化工具復原, 例如免費軟體 SD Formatter. 參考網站為 <http://www.sdcard.org/consumers/formatter> 3.



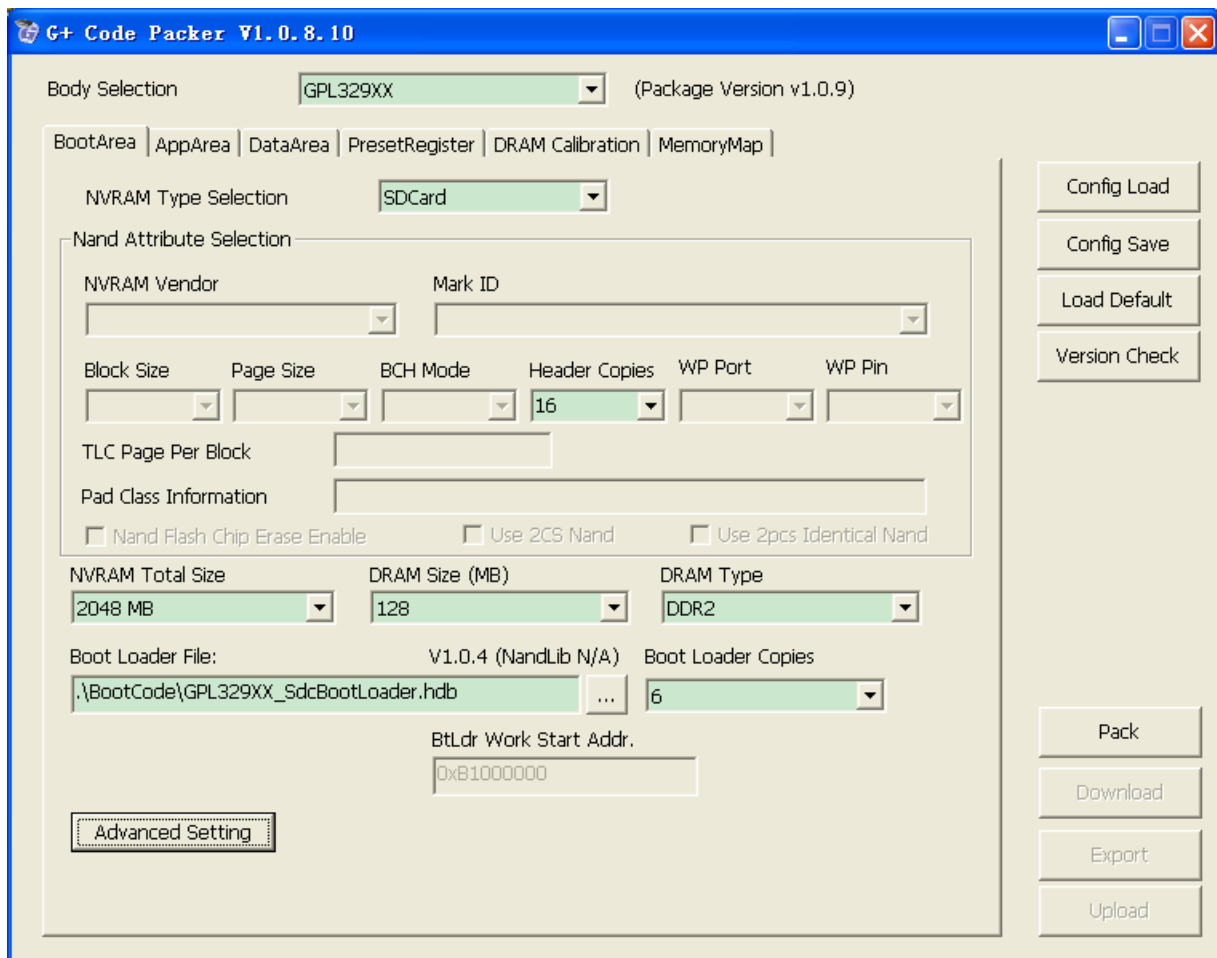
下面為 Boot Area、App Area 以及 Data Area 設置方式說明：

8.1 Load Config & Config Save

若先前已有做過 Config Save，可透過 Config Load 方式將上次選擇之參數重新引入，Generalplus 強力建議所有之 Resource File，全數放在 G+CodePacker 為 Root path 之路徑下（可自行開 folder），如此可對整體 G+CodePacker 之可攜性有顯著提升，並可將使用者經驗透過 Config Save，輕易轉交合作夥伴。

8.2 設置Boot Area

選擇 SD Card，設置 NVRAM Total Size（也就是 SDCard Size），BootHeader Copies，BootLoader Copies，DRAM Size，DRAM Type，Boot Loader File 和 App Area Size。

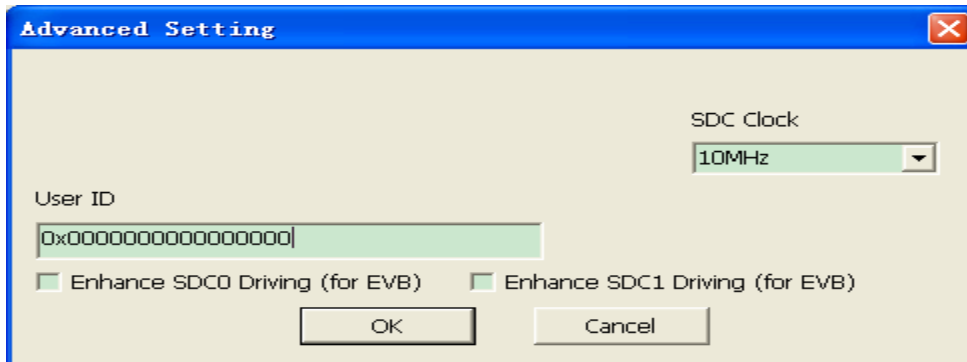


The screenshot shows the 'G+ Code Packer V1.0.8.10' window with the 'BootArea' tab selected. The interface includes the following fields and controls:

- Body Selection:** A dropdown menu set to 'GPL329XX' with '(Package Version v1.0.9)' next to it.
- BootArea | AppArea | DataArea | PresetRegister | DRAM Calibration | MemoryMap:** A series of tabs, with 'BootArea' currently active.
- NVRAM Type Selection:** A dropdown menu set to 'SDCard'.
- Nand Attribute Selection:** A group box containing:
 - NVRAM Vendor:** A dropdown menu.
 - Mark ID:** A dropdown menu.
 - Block Size:** A dropdown menu.
 - Page Size:** A dropdown menu.
 - BCH Mode:** A dropdown menu.
 - Header Copies:** A dropdown menu set to '16'.
 - WP Port:** A dropdown menu.
 - WP Pin:** A dropdown menu.
 - TLC Page Per Block:** A text input field.
 - Pad Class Information:** A text input field.
 - Options:** Three checkboxes: 'Nand Flash Chip Erase Enable' (unchecked), 'Use 2CS Nand' (unchecked), and 'Use 2pcs Identical Nand' (unchecked).
- NVRAM Total Size:** A dropdown menu set to '2048 MB'.
- DRAM Size (MB):** A dropdown menu set to '128'.
- DRAM Type:** A dropdown menu set to 'DDR2'.
- Boot Loader File:** A text input field containing '..\BootCode\GPL329XX_SdcBootLoader.hdb' and a browse button (...).
- V1.0.4 (NandLib N/A):** A text label.
- Boot Loader Copies:** A dropdown menu set to '6'.
- BtLdr Work Start Addr.:** A text input field containing '0x81000000'.
- Buttons:** A vertical stack of buttons on the right: 'Config Load', 'Config Save', 'Load Default', 'Version Check', 'Pack', 'Download', 'Export', and 'Upload'. There is also an 'Advanced Setting' button at the bottom left.

高级设置:

对 SD 卡的设置还提供了高级设置项, 其用户界面如下:



SDC Clock: 此项是设定 IC 访问 SD 存储器的速度, 目前默认值是 10MHz,同时提供 400KHz, 5MHz, 10MHz, 20MHz, 40MHz 等 5 组值。

User ID:此项的含义可由用户自己指定, 比如可以用作给 Device 指定出厂批次, 日期以及版号等信息, 方便以后升级或维修。

其它两项, 主要是针对 Evb 上 SDC0/1 在烧写时, 有可能因布线或其他原因造成驱动电流不足, 可以勾选此选项来增强以弥补。

8.3 設置App Area

設置方式與 Nand Flash 完全相同, 請參考第 5.3 章節使用說明。

8.4 設置Data Area及FAT Image Tool之使用方法

設置方式與 Nand Flash 完全相同, 請參考第 5.4 章節使用說明。

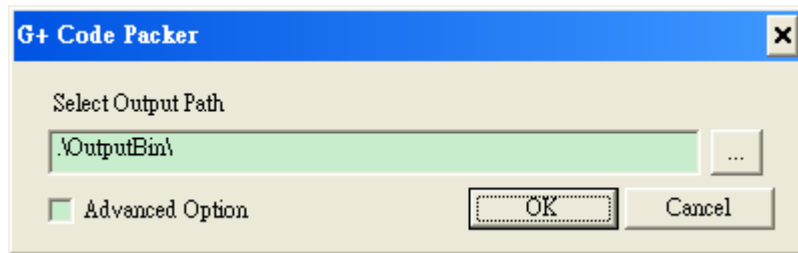
8.5 設置Pre-set Register及DRAM Calibration之使用方法

設置方式與 Nand Flash 完全相同, 請參考第 5.5 章節使用說明。

8.6 Pack與Download

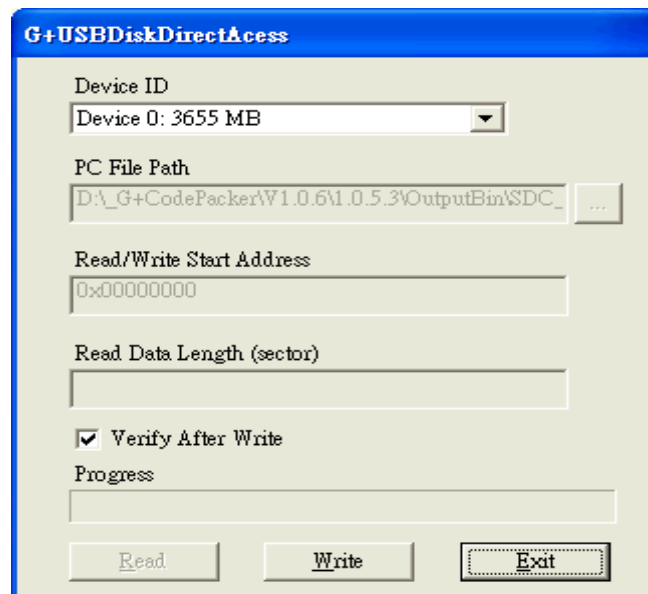
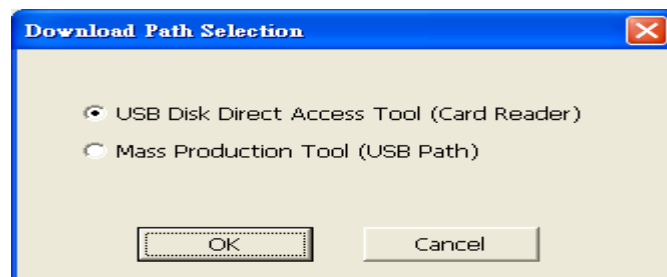
Step 1 : Pack

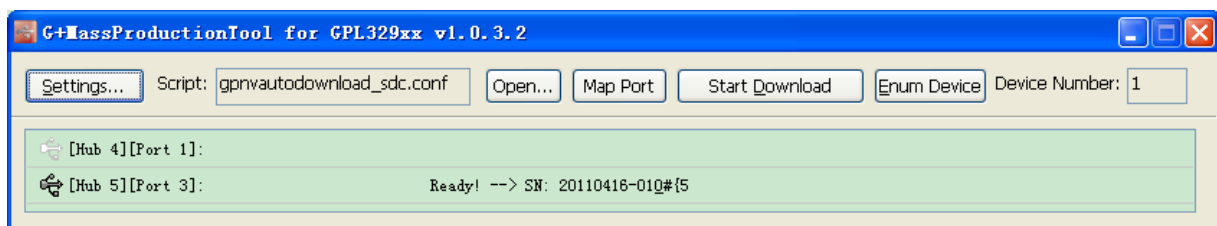
設置完各區域後, 可以照需求點擊 Pack 按鈕進行整合動作。此時 tool 會依據設置將所有檔案整合至最終的 BIN 檔當中, 並將 BIN 檔放置於使用者選擇的目錄下(建議使用預設路徑)。



Step 2 : Download

Pack 動作完成後，Download 按鈕會被激發。點擊 Download 按鈕，可以選擇讀卡機(讀卡機需連接 PC)或 MP Tool 等兩種 download 模式燒錄 SDCard，讀卡機 download 會喚起 USB DiskDirectAccess 子工具，設置好參數按 Write 鍵將 SD boot image 存入 SDCard。將此 SDCard 放到 target board 即可進行 SD boot。MP Tool download 會喚起 G+MassProductionTool 將 SD boot image download 到 target board 上的 SDCard。

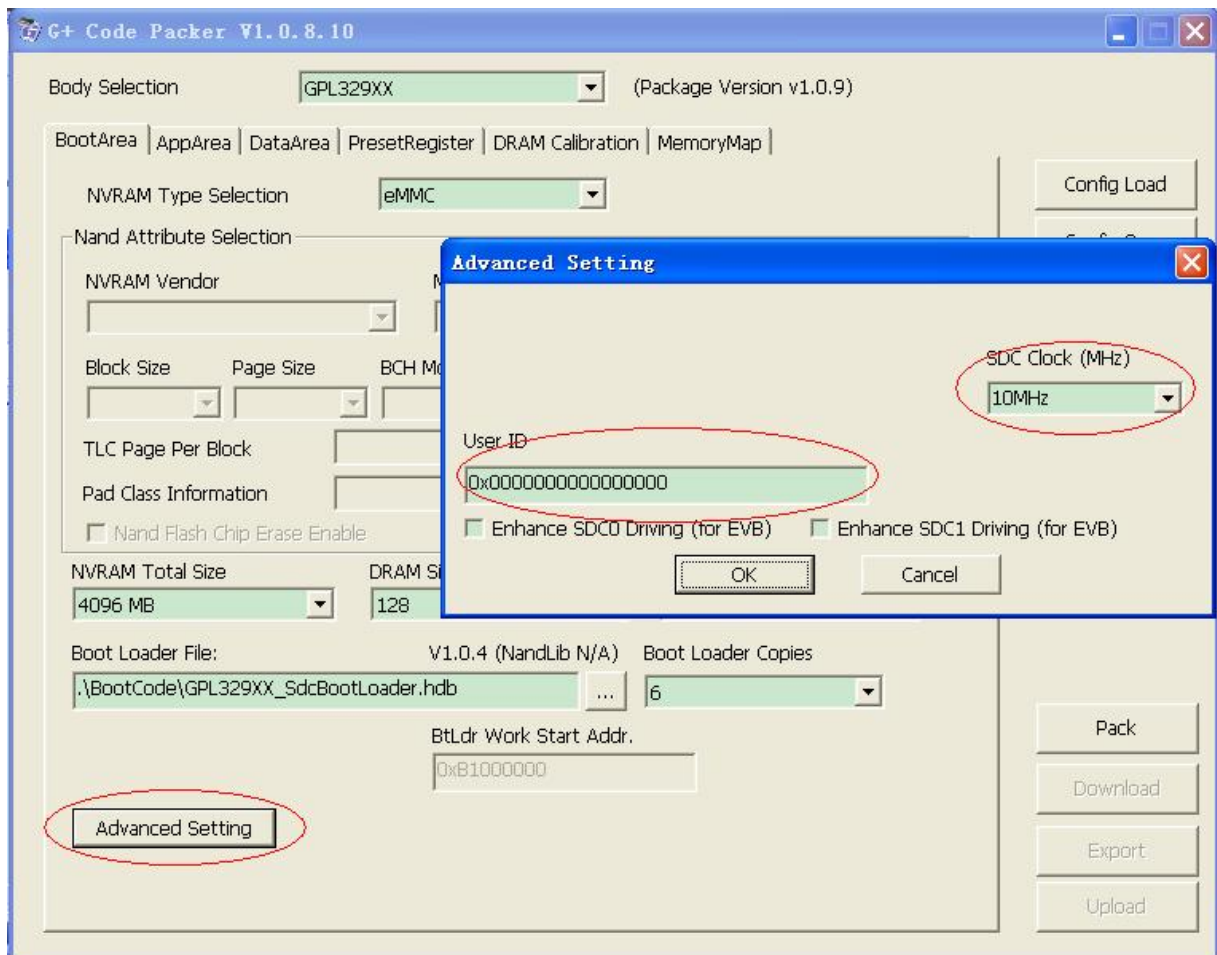




9 eMMC Boot操作流程解說 (GPL329XX)

GPL329XX 將 eMMC 區分為 3 個操作區域：Boot Area、App Area 以及 Data Area。使用者透過 G+CodePacker 可分別設置這三個區域的資源檔來進行整合。

若遇到 eMMC Boot 相容性問題，Generalplus 建議打開 Advance Mode (勾選 Advance Setting) 設置較低 SDC Clock 將 eMMC 降速操作，如果 Target Board (例如：Generalplus EVB) 的負載較重可以勾選 SDC driving 增強。



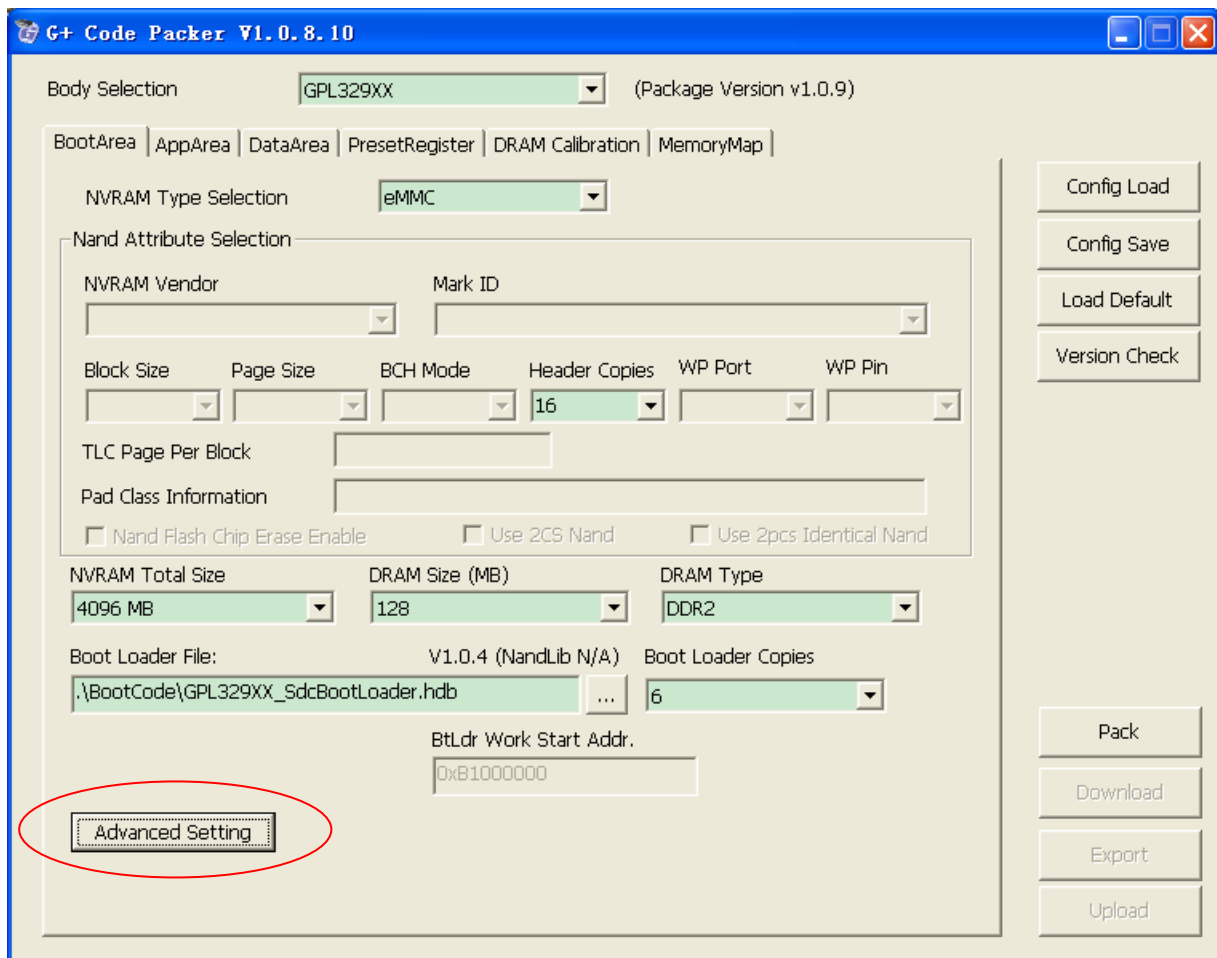
下面為 Boot Area、App Area 以及 Data Area 設置方式說明：

9.1 Load Config & Config Save

若先前已有做過 Config Save，可透過 Config Load 方式將上次選擇之參數重新引入，Generalplus 強力建議所有之 Resource File，全數放在 G+CodePacker 為 Root path 之路徑下（可自行開 folder），如此可對整體 G+CodePacker 之可攜性有顯著提升，並可將使用者經驗透過 Config Save，輕易轉交合作夥伴。

9.2 設置 Boot Area

選擇 eMMC，設置 NVRAM Total Size（也就是 eMMC Size），DRAM Size，DRAM Type，Boot Loader File 和 App Area Size。



9.3 設置App Area

設置方式與 Nand Flash 完全相同，請參考第 5.3 章節使用說明。

9.4 設置Data Area及FAT Image Tool之使用方法

設置方式與 Nand Flash 完全相同，請參考第 5.4 章節使用說明。

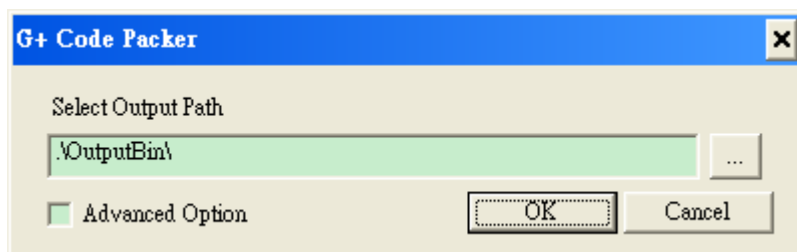
9.5 設置Pre-set Register及DRAM Calibration之使用方法

設置方式與 Nand Flash 完全相同，請參考第 5.5 章節使用說明。

9.6 Pack與Download

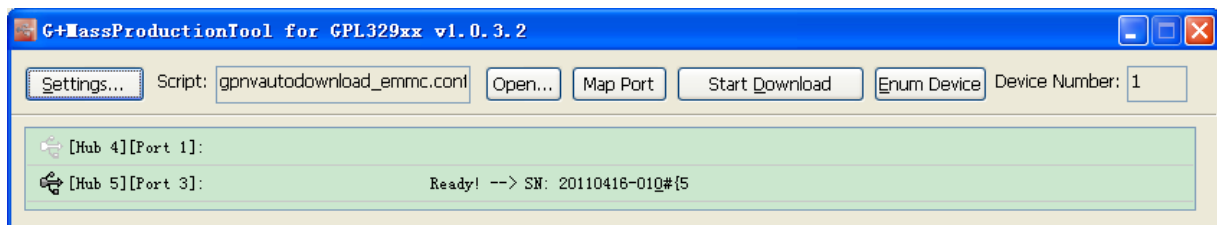
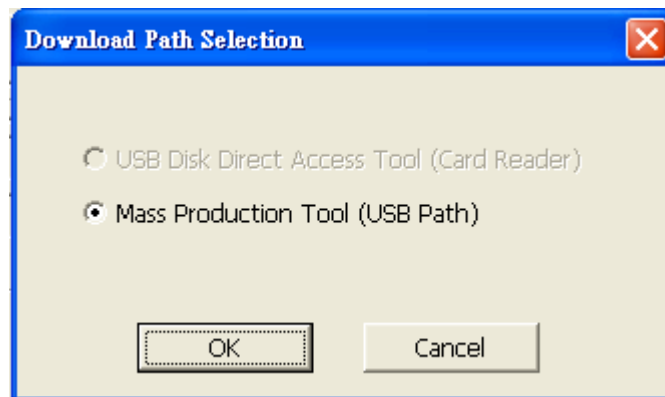
Step 1 : Pack

設置完各區域後，可以照需求點擊 **Pack** 按鈕進行整合動作。此時 tool 會依據設置將所有檔案整合至最終的 BIN 檔當中，並將 BIN 檔放置於使用者選擇的目錄下(建議使用預設路徑)。



Step 2 : Download

Pack 動作完成後，Download 按鍵會被激發。點擊 Download 按鍵，選擇 MP Tool download 後喚起 G+MassProductionTool 將 eMMC boot image download 到 target board 上的 eMMC。



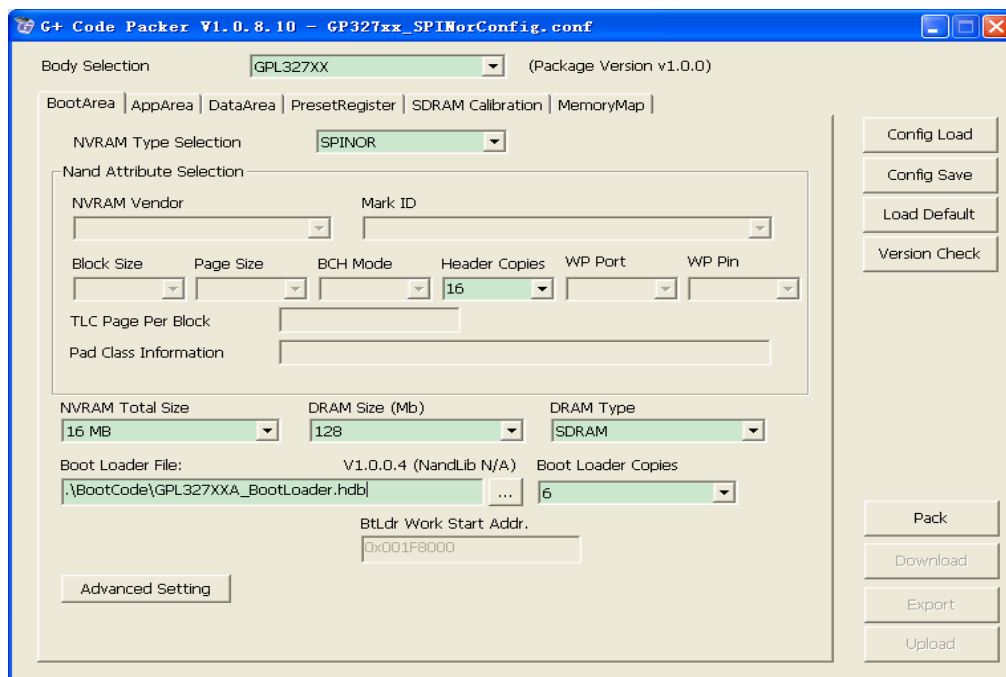
10 SPI Nor Flash操作流程解说 (GPL327xx/GPDV6xxx)

打开 Code Packer tool, 在 Body Selection 的下拉菜单中选择 GPL327xx/GPDV6xxx, 表明本次进行的所有配置都是建立在 GPL327xx/GPDV6xxx 这个 Body 系列的。此 Body 的配置主要由 Boot Area, App Area, Data Area 及 Pre-Register setting 四部分来组成。下面分别对这几种配置加以描述。

10.1 Boot Area配置

10.1.1 Boot Area 基本配置

前面有提到, 我们有将 SPI Nor Flash 划分成三个区, 即 Boot Area, App Area, Data Area. 那这里的 Boot Area 的配置则主要是将 Boot Header 及 First Boot Loader 按照目前此区规划的备份机制来摆放。Boot Header 的相关参数则是透过如下画面来进行配置的:



从上面的图形, 不难看出, 有如下七项需要我们来设置:

➤ **NVRAM Type Selection:**

目前此下拉菜单中, 有三组 NVRam 可供选择: SPI Nor, SD Card, eMMC, 在这里, 因为是对 SPI Nor Flash 进行操作, 所以此 option 应选择 SPINOR.

➤ **Header Copies:**

为了防止 Boot Area 受外界干扰而被破坏, 造成 Boot 失败. 新版 Tool 有对 Boot Header

及 Boot Loader 进行备份，一般，Codepacker 包出来的 Boot Header 大小为 SPI Nor Flash 的一个 Sector，即 512Byte,所以用户所填的份数，即代表占用 SPI Nor Flash 的 Sector 数。

➤ **NVRAM Total Size:**

这个栏位，是用来告诉 FW Code 当前 SPI Nor 的总容量大小,以 Byte 为单位。

CodePacker 目前支持 512KB, 1MB, 2MB, 4MB, 8MB, 16MB, 32MB 共 7 组值. 用户可以透过从下拉菜单中选取。

➤ **DRAM Size:**

此栏位是指 GPL327xx/GPDV6xxx 小机上的对应的 Dram Type 的容量，以 Mbit 为单位，Tool 目前支持 16Mb,

64Mb,128Mb, 256Mb, 512Mb, 1024Mb, 2048Mb, 4096Mb 共 8 组值, 8Mb = 1MB。同样，用户只需要点选下拉菜单就可以找到相应的值。

➤ **DRAM Type:**

此栏位主要是指指定后续做 Calibration 的 Ram 的类型，以及烧录时 ISP Bin code 指定的存储器。

➤ **Boot Loader Files:**

此栏位是选取 First BootLoader 文件，通常是由 Tool 根据 IC 系列及 NV Ram Type 来决定使用哪个

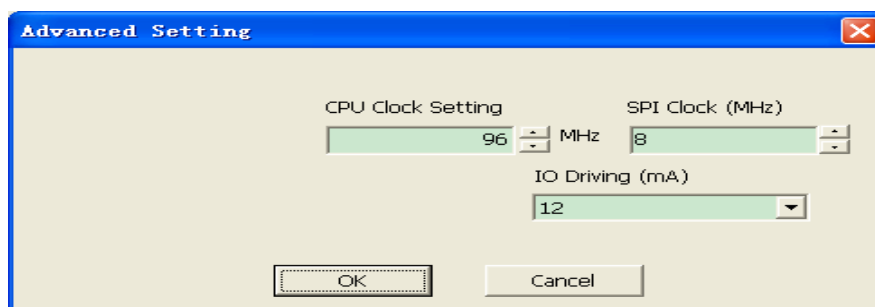
First Bootloader 二制档，此档由 Generalplus 相关人员开发维护，建议用户使用此默认档。

➤ **Boot Loader Copies:**

同前面的 Boot Header Copies 相似，都是用来指定备份数的，只不过，此栏位是用来表明 First Boot loader 的备份数的。Rom code 会根据这些 copies 来进行恢复，从而有效降低 SPI Nor Flash 的不良率。

10.1.2 Boot Area 高级设置

Boot Area 除了基本的设置外，还增加了一组高级设置。通常情况下，这些高级设置选项，都会产生一组默认值，用户也不需要理会它。SPI Nor Flash 的高级设置画面如下：



里面有三个选项可供用户设置，分别为 CPU clock setting, SPI Clock, I/O Driving。

➤ CPU Clock Setting:

CPU clock setting 的设置范围为 4~118MHZ, 默认值为 96MHz, 用户可以自己填入值..对它的设置直接影响 CPU 跑 code 的速度。

➤ SPI Clock:

此选项是用来调整 SPI Nor flash 的时钟, 其设置范围为 0~24MHz, 默认值为 8MHz, 对它的调整会影响 CPU 对此 Flash 存储器的存取速度。

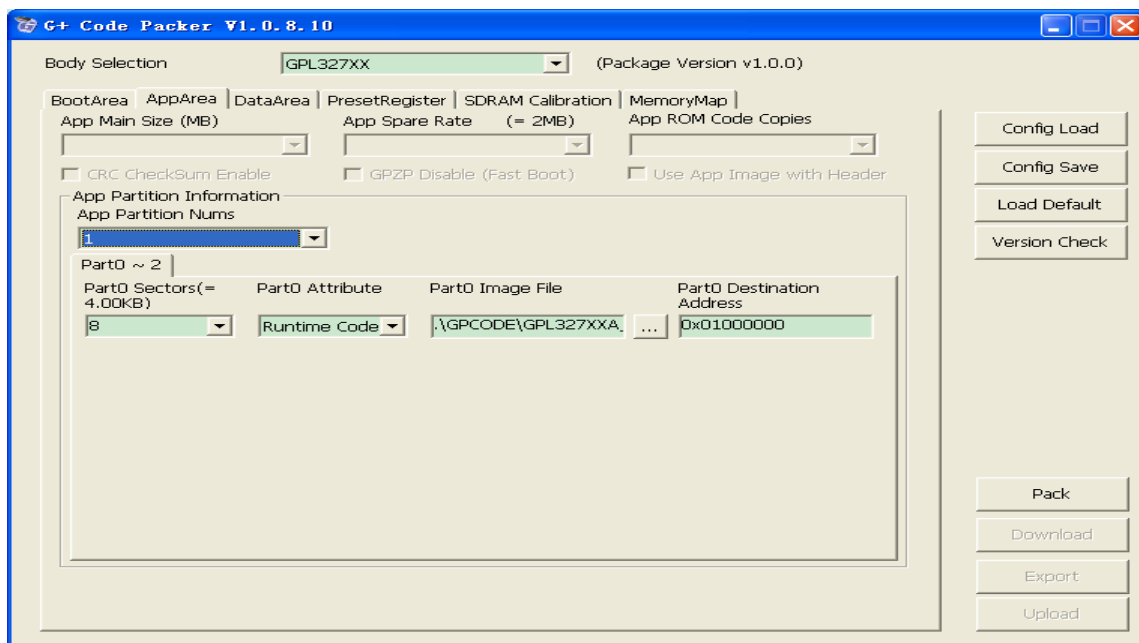
➤ I/O Driving:

此栏位是用来调整 I/O Pin 脚的驱动电流, 目前有四组值, 分别为 4, 8, 12, 16, 当用户 Evb 上 I/O 口与 IC 之间或 SPI Nor Flash 之间的连线太长或太短, 就可以适当调整此 Option, 来增大 I/O 口的输送信号或降低线路间 EMI 干扰的能力。

对所有的选项设定完成后, 点"OK"按钮, 使当前修改生效; 点"Cancel"按钮, 取消当前设置。

10.2 App Area 配置

点选 App Area 标签, 切换到 App Area, UI 长得样子如下:



如上图, 此页面主要是对 App Partition 进行配置, 一是设定分区个数; 二是为每个分区配置相应的参数。

App 分区信息参数配置如下:

- App Partion Num: 此选项表明用户当前设定的 App 区的分区数。目前设定范围 1~16, 共 16 组值, 也就是用户最多可配置 16 个分区。
- Partion N Size: N 的范围从 0~15, 它是表明当前这个分区的容量大小, 以 Sector 为单位, 通常情况下, 当我们选定分区镜像档案后, Tool 会自动帮我们将相应的档案大小填入此栏位。有一种情况是, 用户可以不选取档案, 此时, 该栏位会被置为 auto, 如果

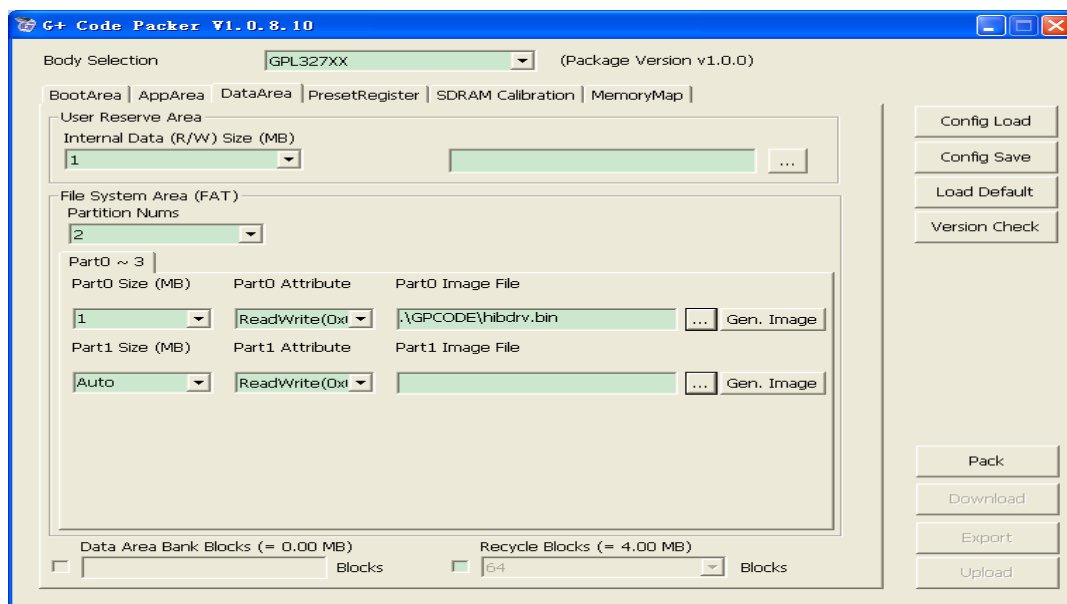
用户想指定此分区大小，可以直接输入该值。

- **Partion N Attribute:** 同理，N 的范围是从 0~15,它表明为当前镜像文件是什么类型的文件，目前 Tool 提供 Runtime Code, Resource Part, FastBoot Bin, Quick Image, Hibrate Image, Image Flag, Customize BtLdr 共 7 种类型的 App 镜像文件。当 First bootloader 在搬 App 区的 code 时，此参数就显得尤为重要。
- **Partion N Image file:** 此参数是让用户来选取磁盘上的分区镜像文件。
- **Partion N Destination Address:** 此参数是指定当前分区文件，将来会被摆放在 SDRam 哪个地址来跑。

依此方法，对指定数目的分区进行配置，直到所有分区信息都配置完成。

10.3 Data Area 配置

在 App 区的分区信息设定好后，可以根据需要切换到 Data Area。其画面如下：



数据区的配置主要有由用户预留区和文件系统区两部分组成，暂不支持 Recycle Block 功能。

- **用户预留区：**

此区域主要是在数据区域划分一段存储空间，给用户自行使用的。用户可以预先配置一个文件，也可以不为此存储空间指定文件。不论怎样，用户都可以透过 SPI Flash Driver 中诸如 Read Sector / Write Sector 等接口函数来操作此片区域。

- **文件系统区：**

此区域的配置与 App 分区信息的配置比较类似，首先去选定当前用户所期望配置的逻辑分区数，再去对每个具体的逻辑区进行信息的设定。下面简要介绍逻辑区的设定项：

Partition Nums:此项主要是指定逻辑分区数目，共 8 个分区，范围从 1~8，0 表示不分区，也即不设定数据区的分区信息。

PartN Size: N 的范围从 1 到 8，此项主要是指定当前分区的存储容量，在不选中镜像文件

情况下，默认为 Auto，用户也可以自己手动填写此值。

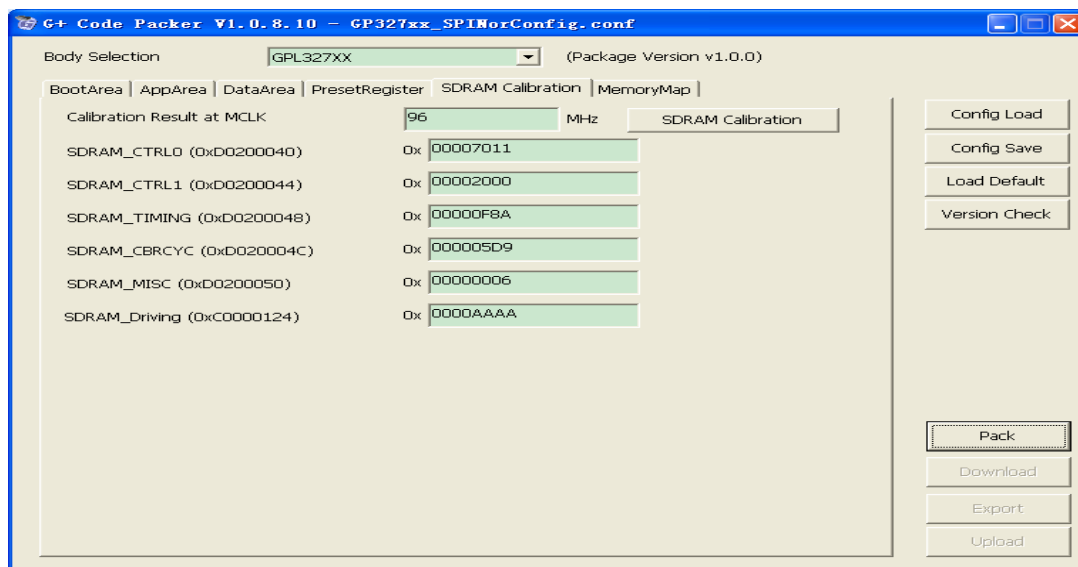
PartN Attribute: N 的范围从 1 到 8，此项主要是指定当前分区的属性，目前有只读，只写，读写三种选项。

PartN Image File:N 的范围从 1 到 8，此项主要是指定当前分区的镜像文件，有两种方式来填充此项，一种是点选“...”按钮，打开文件对话框，在本地机上选二进制文件；另一种方法，则是点选“Gen.Image”按钮来透过 G+FatImageTool 来生成此档案，至于 G+FatImage Tool 的使用请参看其 User Guide。

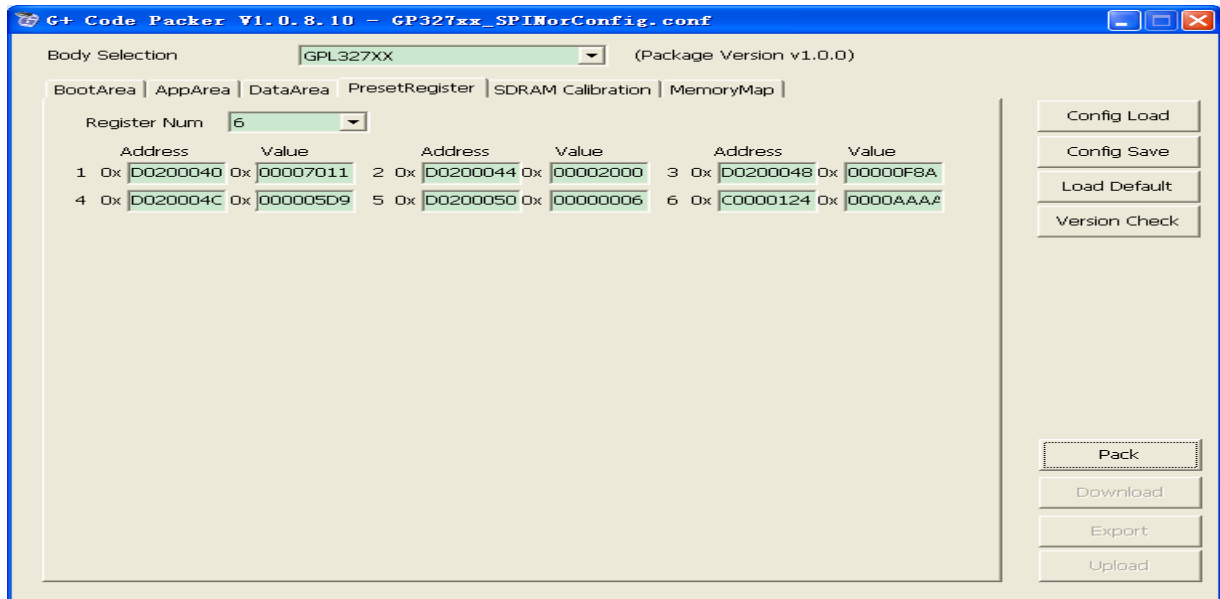
一般来讲，用户保留区与数据分区虽同属数据区，但提供给用户的用途不同，前者主要是透过 SPI Nor Flash 的驱动程序中读写接口函数来直接操作此区域；后者则是透过文件系统来操作各个不同的逻辑分区。

10.4 Pre-Register & Dram Calibration设置

通常情况下，我们都是先对 GP327xx/GPDV6xxx 的 IC 做 Dram Calibration，然后，再切回到 Pre-Register 页面，来查看相应的寄存器的值。在进入到烧录代码前，用户得确保有做 Dram Calibration 的动作。切换到 Dram Calibration 的标签，可看到 Dram Calibration 的 UI 如下：



用户切换此页面后，点选“SDRam Calibration”按钮，进入到 G+ SDRAM Calibration Tool，透过此 Tool 对 SDRam 的 Register 进行 Calibration。至于此 Tool 的使用，用户就要参考下 G+ SDRAM Calibration Tool 的用户指南。SDRam Calibration 成功后，切换到 Pre-Register 标签项，可以看到 Calibration 后的值被正确反映到 Pre-Register 页面中，其内容长得如下样子：

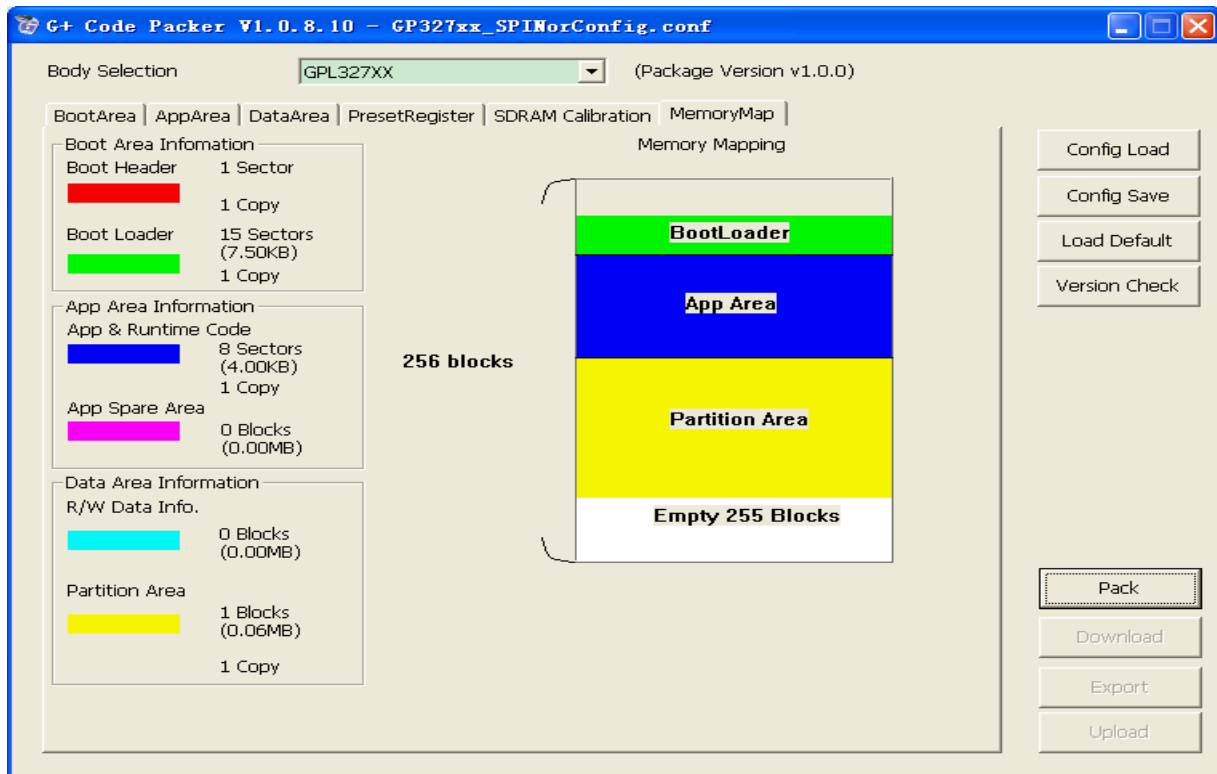


从上图可以看出，它由两部分组成，一部分是寄存器数目；另一部分是第组寄存器的地址及地址单元里的值。

当然，用户如果因某些原因，发现 **Dram calibration** 没有达到预期效果，或者为调整开机阶段性的设定(ex. Clock 或 IO 口的状态)等情况，这时就可以直接在此页面，透过手动修改的方式来增加相应寄存器的设定。

10.5 Memory Map

前面对 **Boot** 区，**App** 区，**Data** 区进行配置后，切换到此页面，即可看到根据前面配置所映射到 **SPI Nor Flash** 存储器上的布局情况。如下图：



目前看来，此页面里的数据反映的有些问题，这在后续版本中会进行修正，且并不影响数据在实际存储器的分布情况。

10.6 按钮功能

所有的配置都完成后，用户可以点选“Config Save”按钮，将当前的配置保存起来，这样，下次就可以透过“Config Load”按钮来加载之前保存好的配置文件，这里要注意的是，新版本的Code Packer Tool中尽量不要加载旧版本Tool保存下来的配置文件，因为很有可能二者并不兼容，特别是新旧版本号隔得太远，兼容性就更欠佳。

Load Default:此按钮主要是用户在不需关闭Code packer的情况下，清除用户当前配置，恢复到Tool初始状态。

Version Check:此按钮主要是用来查看各个IC系列所发布的HDB档案的版控信息。

Pack:此按钮主要是将配置好的一些信息，按照一定的规则，包出烧录程式所需要的文件。

Download: 此按钮的作用是将烧录工具G+ Mass Production Tool呼叫起来，准备执行程序的烧写。

Export:此按钮则是在将包出来的文件及烧录程式一同导出到磁盘某个地方，这样，只要打开G+ Mass Production 量产工具，就要可以进行档案的烧录，这个功能主要是方便工厂进行量产之用。

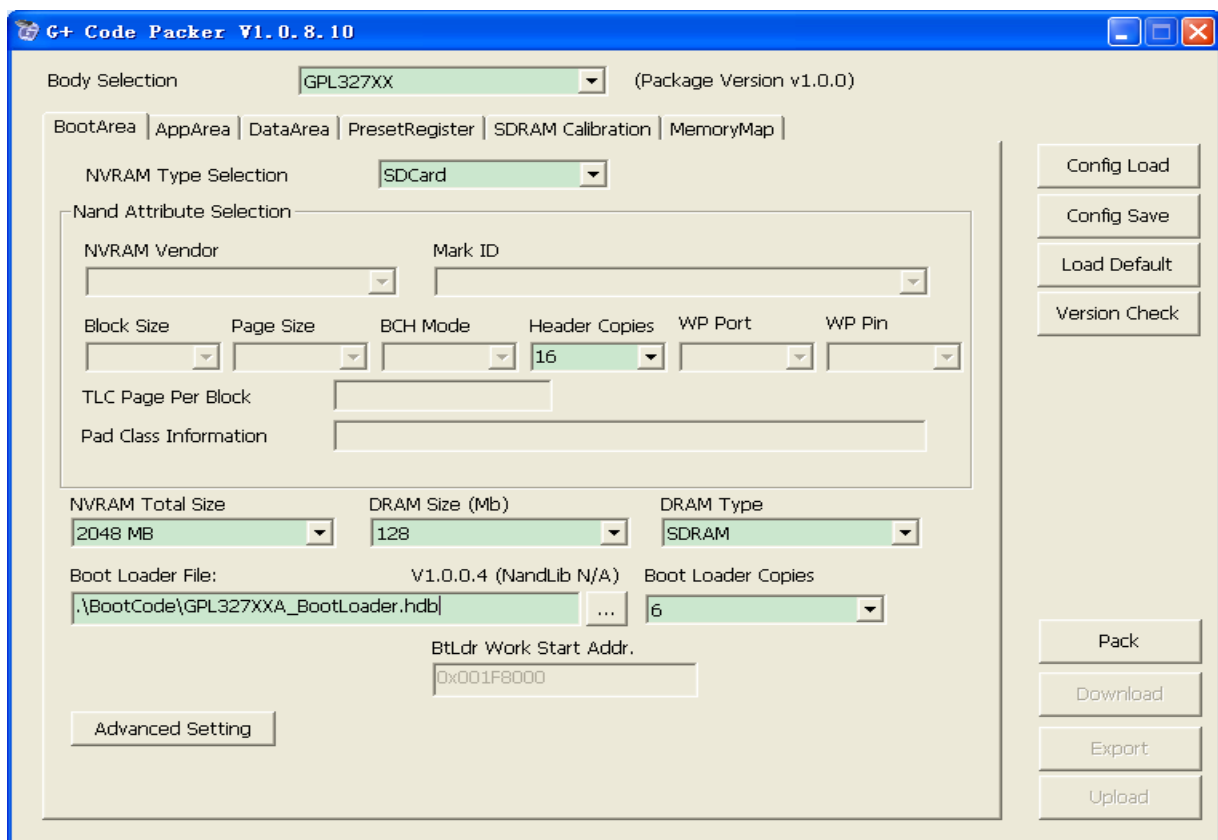
11 SD Card操作流程解说 (GPL327xx/GPDV6xxx)

打开 Code Packer tool, 在 Body Selection 的下拉菜单中选择 GPL327xx/GPDV6xxx, 表明本次进行的所有配置都是建立在 GPL327xx/GPDV6xxx 这个 Body 系列的。此 Body 的配置主要由 Boot Area, App Area, Data Area 及 Pre-Register setting 四部分来组成。下面分别对这几种配置加以描述。

11.1 Boot Area配置

11.1.1 Boot Area 基本配置

前面有提到, 我们有将 SD Card 存储器划分成三个区, 即 Boot Area, App Area, Data Area. 那这里的 Boot Area 的配置则主要是将 Boot Header 及 First Boot Loader 按照目前此区规划的备份机制来摆放。Boot Header 的相关参数则是透过如下画面来进行配置的:



从上面的图形, 不难看出, 有如下七项需要我们来设置:

- NVRAM Type Selection:

目前此下拉菜单中，有三组 NVRam 可供选择:SPI Nor, SD Card, eMMC, 在这里，因为是对 SD Card 进行操作，所以此 option 应选择 SDCard.

➤ **Header Copies:**

为了防止 Boot Area 受外界干扰而被破坏,造成 Boot 失败。新版 Tool 有对 Boot Header 及 Boot Loader 进行备份，一般，Codepacker 包出来的 Boot Header 大小为 SD Card 的一个 Sector，即 512Byte,所以用户所填的份数，即代表占用 SD Card 的 Sector 数。

➤ **NVRAM Total Size:**

这个栏位，是用来告诉 FW Code 当前 SD Card 的总容量大小,以 MByte 为单位。

CodePacker 目前支持 1024MB, 2048MB, 4096MB, 8192MB, 16384MB, 32768MB 共 6 组值. 用户可以透过从下栏菜单中选取。

➤ **DRAM Size:**

此栏位是指 GPL327xx/GPDV6xxx 小机上的对应的 Dram Type 的容量，以 Mbit 为单位，Tool 目前支持 16Mb,

64Mb,128Mb, 256Mb, 512Mb, 1024Mb, 2048Mb, 4096Mb 共 8 组值, 8Mb = 1MB。同样，用户只需要点选下栏菜单就可以找到相应的值。

➤ **DRAM Type:**

此栏位主要是指指定后续做 Calibration 的 Ram 的类型，以及烧录时 ISP Bin code 指定的存储器。

➤ **Boot Loader Files:**

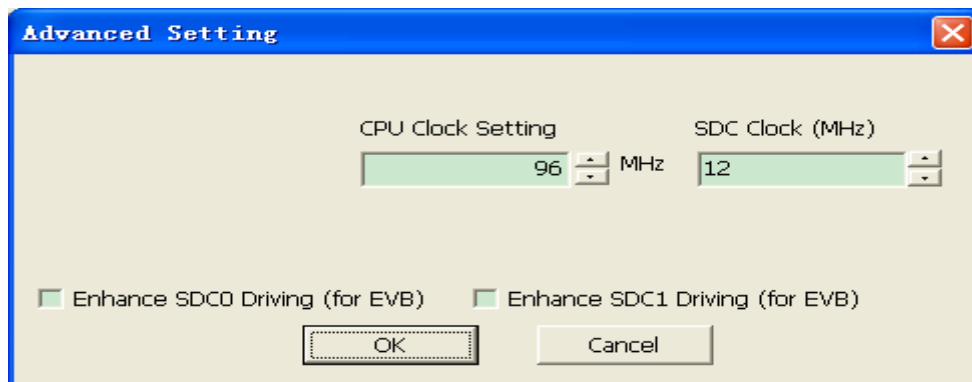
此栏位是选取 First BootLoader 文件，通常是由 Tool 根据 IC 系列及 NV Ram Type 来决定使用哪个 First Bootloader 二制档，此档由 Generalplus 相关人员开发维护，建议用户使用此默认档。

➤ **Boot Loader Copies:**

同前面的 Boot Header Copies 相似，都是用来指定备份数的，只不过，此栏位是用来表明 First Boot loader 的备份数的。Rom code 会根据这些 copies 来进行恢复，从而有效降低 SD Card 的不良率。

11.1.2 Boot Area 高级设置

Boot Area 除了基本的设置外，还增加了一组高级设置。通常情况下，这些高级设置选项，都会产生一组默认值，用户也不需要理会它。SD Card 的高级设置画面如下：



里面有三个选项可供用户设置，分别为 CPU clock setting, SDC Clock, Enhance SDC0/1 Driving(for EVB)。

➤ CPU Clock Setting:

CPU clock setting 的设置范围为 4~118MHZ，默认值为 108MHz，用户可以自己填入值.. 对它的设置直接影响 CPU 跑 code 的速度。

➤ SDC Clock:

此选项是用来调整 SPI Nor flash 的时钟，其设置范围为 0~50MHz，默认值为 12MHz，对它的调整会影响 CPU 对此 SD Card 存储器的存取速度。

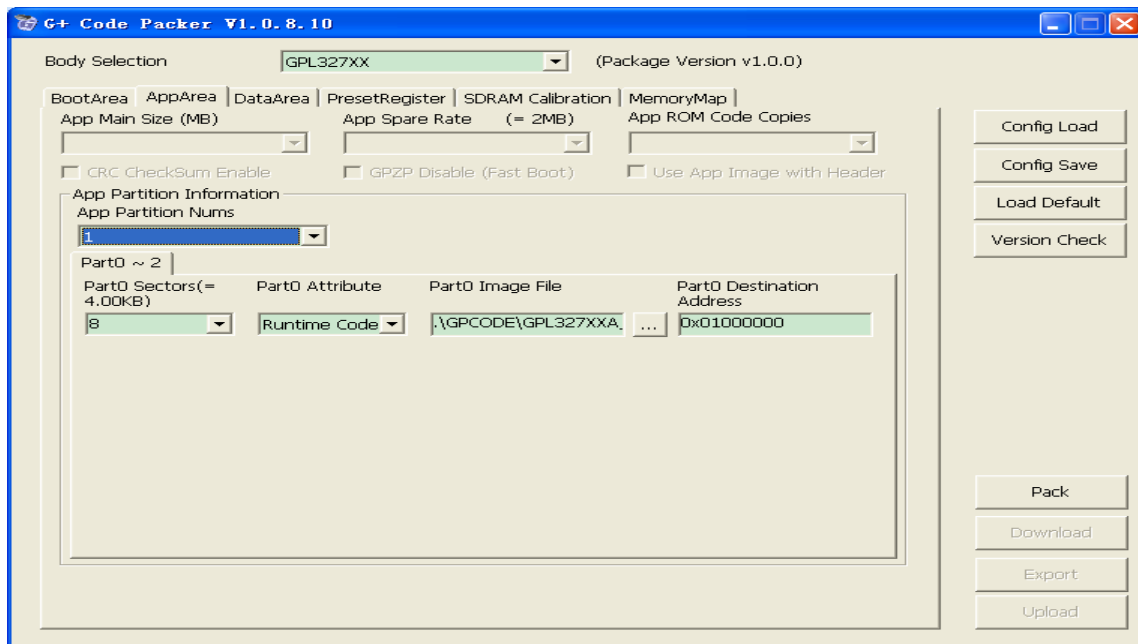
➤ Enhance SDC0/1 Driving(for EVB):

此栏位若勾选则表明增强 SDC0/1 的驱动电流，当用户 Evb 上 I/O 口与 IC 之间或 SDC0/1 之间的连线太长或太短，就可以适当调整此 Option，来增强 SD Card I/O 口的输送信号或降低线路间 EMI 干扰的能力。

对所有的选项设定完成后，点"OK"按钮，使当前修改生效；点"Cancel"按钮，取消当前设置。

11.2 App Area 配置

点选 App Area 标签，切换到 App Area，UI 长得样子如下：



如上图，此页面主要是对 App Partition 进行配置，一是设定分区个数；二是为每个分区配置相应的参数。

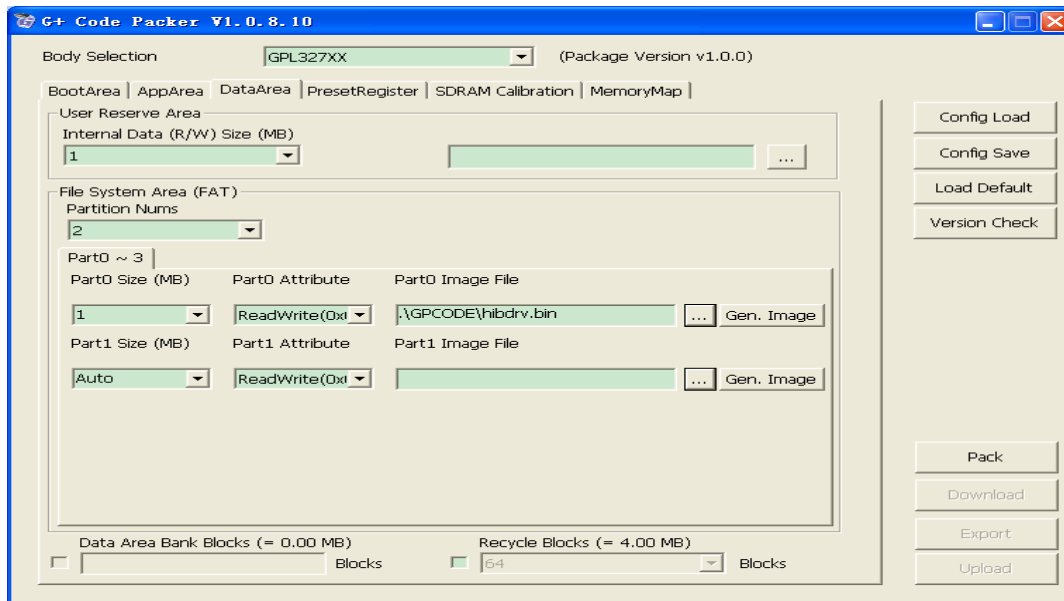
App 分区信息参数配置如下：

- **App Partion Num:** 此选项表明用户当前设定的 App 区的分区数。目前设定范围 1~16，共 16 组值，也就是用户最多可配置 16 个分区。
- **Partion N Size:** N 的范围从 0~15，它是表明当前这个分区的容量大小，以 Sector 为单位，通常情况下，当我们选定分区镜像档案后，Tool 会自动帮我们将相应的档案大小填入此栏位。有一种情况是，用户可以不选取档案，此时，该栏位会被置为 auto,如果用户想指定此分区大小，可以直接输入该值。
- **Partion N Attribute:** 同理，N 的范围是从 0~15,它表明为当前镜像文件是什么类型的文件，目前 Tool 提供 Runtime Code, Resourse Part, FastBoot Bin, Quick Image, Hibrate Image, Image Flag, Customize BtLdr 共 7 种类型的 App 镜像文件。当 First bootloader 在搬 App 区的 code 时，此参数就显得尤为重要。
- **Partion N Image file:** 此参数是让用户来选取磁盘上的分区镜像文件。
- **Partion N Destination Address:** 此参数是指定当前分区文件，将来会被摆放在 SDRam 哪个地址来跑。

依此方法，对指定数目的分区进行配置，直到所有分区信息都配置完成。

11.3 Data Area 配置

在 App 区的分区信息设定好后，可以根据需要切换到 Data Area。其画面如下：



数据区的配置主要有由用户预留区和文件系统区两部分组成,暂不支持 **Recycle Blocks** 的设置。

➤ 用户预留区:

此区域主要是在数据区域划分一段存储空间,给用户自行使用的。用户可以预先配置一个文件,也可以不为此存储空间指定文件。不论怎样,用户都可以透过 **SD Card Driver** 中诸如 **Read Sector / Write Sector** 等接口函数来操作此片区域。

➤ 文件系统区:

此区域的配置与 **App** 分区信息的配置比较类似,首先去选定当前用户所期望配置的逻辑分区数,再去对每个具体的逻辑区进行信息的设定。下面简要介绍逻辑区的设定项:

Partition Nums:此项主要是指定逻辑分区数目,共 8 个分区,范围从 0~8, 0 表示不分区,也即不设定数据区的分区信息。

PartN Size: N 的范围从 1 到 8,此项主要是指定当前分区的存储容量,在不选中镜像文件情况下,默认为 **Auto**,用户也可以自己手动填写此值。

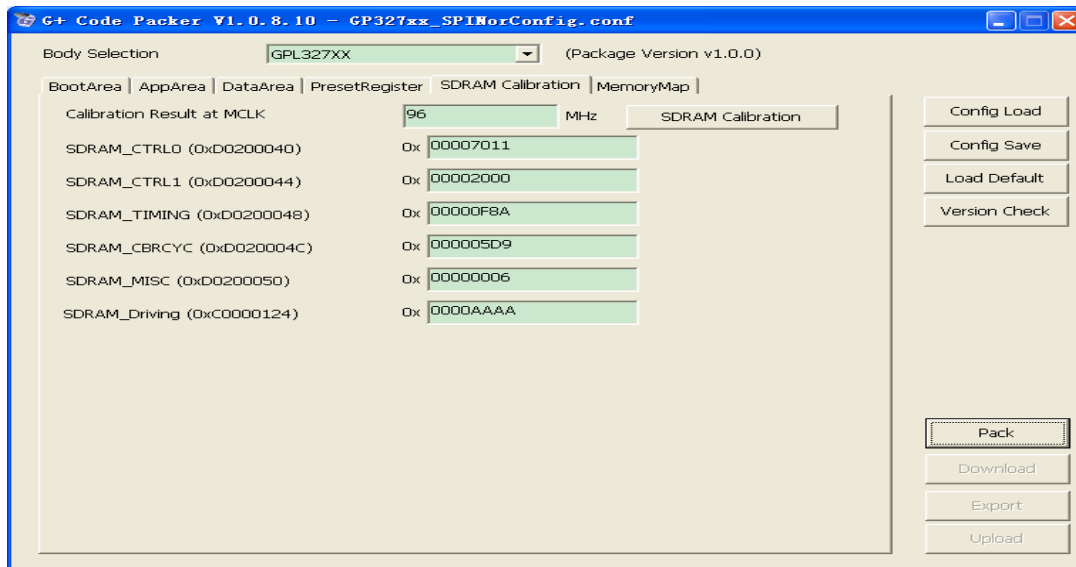
PartN Attribute: N 的范围从 1 到 8,此项主要是指定当前分区的属性,目前有只读,只写,读写三种选项。

PartN Image File:N 的范围从 1 到 8,此项主要是指定当前分区的镜像文件,有两种方式来填充此项,一种是点选"... "按钮,打开文件对话框,在本地机上选二进制文件;另一种方法,则是点选"**Gen.Image**"按钮来透过 **G+FatImageTool** 来生成此档案,至于 **G+FatImage Tool** 的使用请参看其 **User Guide**。

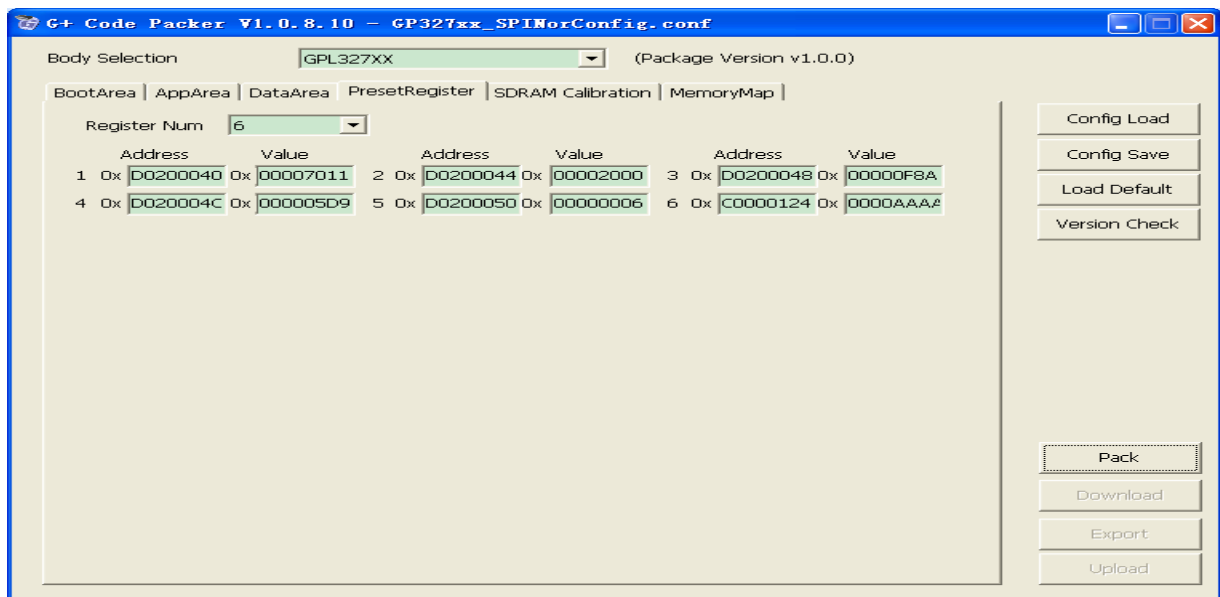
一般来讲,用户保留区与数据分区虽同属数据区,但提供给用户的用途不同,前者主要是透过 **SD Card** 的驱动程序中读写接口函数来直接操作此区域;后者则是透过文件系统来操作各个不同的逻辑分区。

11.4 Pre-Register & Dram Calibration设置

通常情况下，我们都是先对 GP327xx 的 IC 做 Dram Calibration，然后，再切回到 Pre-Register 页面，来查看相应的寄存器的值。在进入到烧录代码前，用户得确保有做 Dram Calibration 的动作。切换到 Dram Calibration 的标签，可看到 Dram Calibration 的 UI 如下：



用户切换此页面后，点选”SDRam Calibration”按钮，进入到 G+ SDRAM Calibration Tool，透过此 Tool 对 SDRam 的 Register 进行 Calibration。至于此 Tool 的使用，用户就要参考下 G+ SDRAM Calibration Tool 的用户指南。SDRam Calibration 成功后，切换到 Pre-Register 标签项，可以看到 Calibration 后的值被正确反映到 Pre-Register 页面中，其内容长得如下样子：

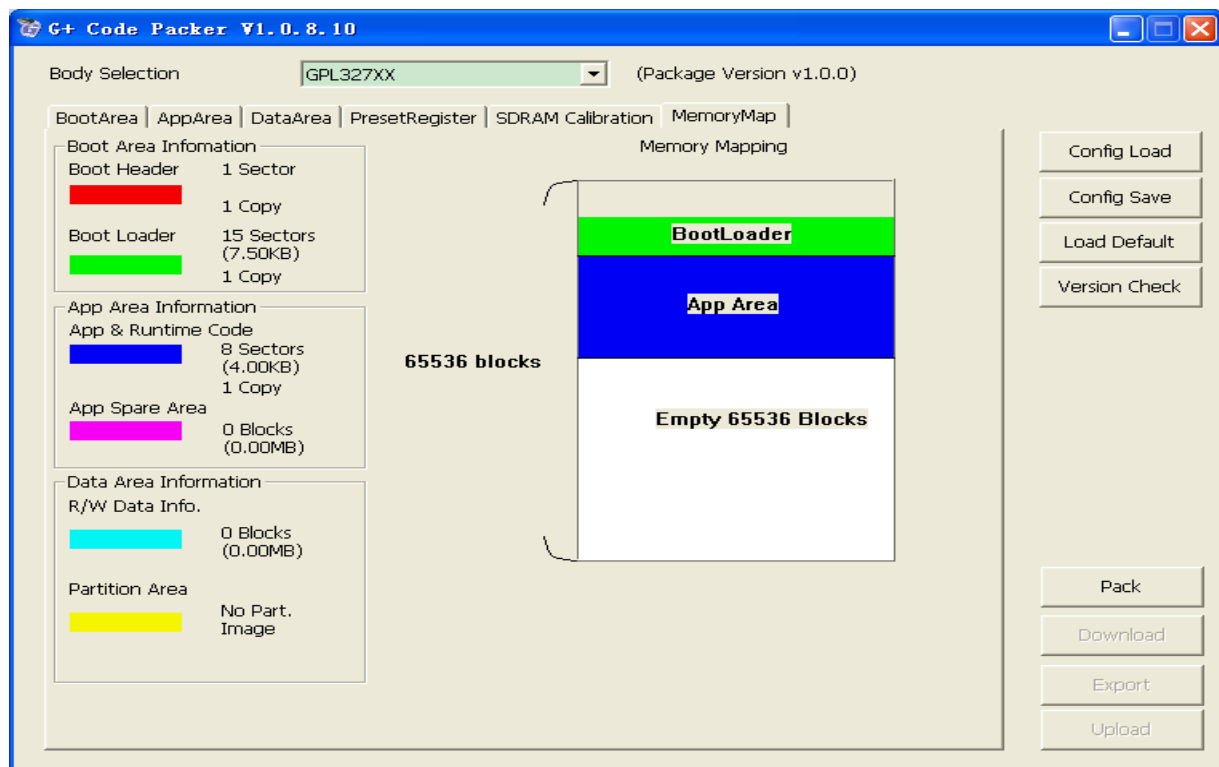


从上图可以看出，它由两部分组成，一部分是寄存器数目；另一部分是第组寄存器的地址及地址单元里的值。当然，用户如果因某些原因，发现 Dram calibration 没有达到预期效果，或者为调整

开机阶段的设定(ex. Clock 或 IO 状态)等情况，这时就可以直接在此页面，透过手动修改的方式来增加相应寄存器的设定。

11.5 Memory Map

前面对 Boot 区，App 区，Data 区进行配置后，切换到此页面，即可看到根据前面配置所映射到 SD Card 存储器上的布局情况。如下图：



目前看来，此页面里的数据反映的有些问题，这在后续版本中会进行修正，且并不影响数据在实际存储器的分布情况。

11.6 按钮功能

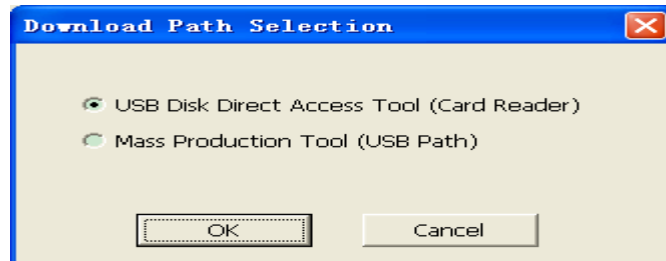
所有的配置都完成后，用户可以点选“Config Save”按钮，将当前的配置保存起来，这样，下次就可以透过“Config Load”按钮来加载之前保存好的配置文件，这里要注意的是，新版本的 Code Packer Tool 中尽量不要加载旧版本 Tool 保存下来的配置文件，因为很有可能二者并不兼容，特别是新旧版本号隔得太远，兼容性就更欠佳。

Load Default:此按钮主要是用户在不需关闭 Code packer 的情况下，清除用户当前配置，恢复到 Tool 初始状态。

Version Check:此按钮主要是用来查看各个 IC 系列所发布的 HDB 档案的版控信息。

Pack:此按钮主要是将配置好的一些信息，按照一定的规则，包出烧录程式所需要的文件。

Download: 此按钮的作用是将烧录工具 G+ Mass Production Tool 或者 G+USB Disk Direct Access Tool 呼叫起来，其画面如下：



两种方式中，前者直接透过 SD 读卡器来对 SD 卡进行数据的存取；后者则是将 SD 卡插在 EVB 上的 SD 卡 slot 上，由透过 GP327xx/GPDV6xx 等 IC 来对它进行读写。(可参考 section 8.6)

Export: 此按钮则是在将包出来的文件及烧录程式一同导出到磁盘某个地方，这样，只要打开 G+ Mass Production 量产工具，就要可以进行档案的烧录，这个功能主要是方便工厂进行量产之用。

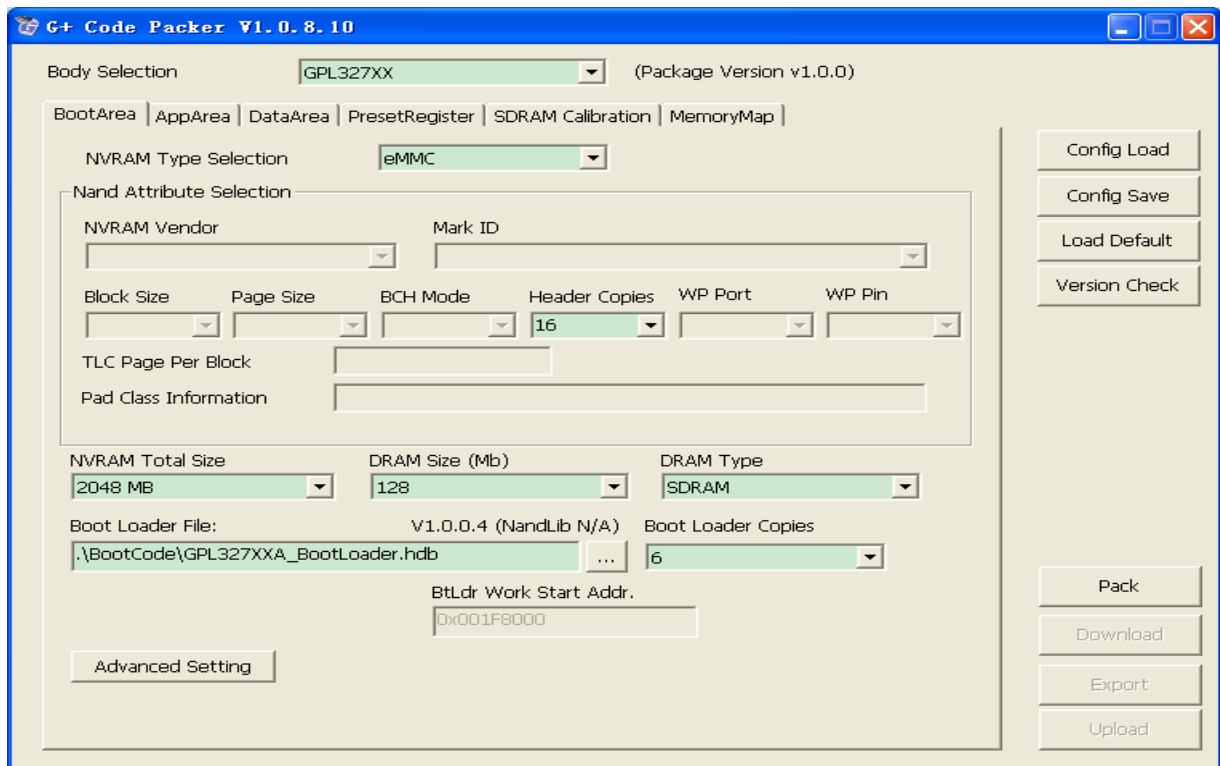
12 eMMC操作流程解说 (GPL327xx/GPDV6xxx)

打开 Code Packer tool, 在 Body Selection 的下拉菜单中选择 GPL327xx/GPDV6xxx, 表明本次进行的所有配置都是建立在 GPL327xx/GPDV6xxx 这个 Body 系列的。此 Body 的配置主要由 Boot Area, App Area, Data Area 及 Pre-Register setting 四部分来组成。下面分别对这几种配置加以描述。

12.1 Boot Area 配置

12.1.1 Boot Area 基本配置

前面有提到, 我们有将 SD Card 存储器划分成三个区, 即 Boot Area, App Area, Data Area. 那这里的 Boot Area 的配置则主要是将 Boot Header 及 First Boot Loader 按照目前此区规划的备份机制来摆放。Boot Header 的相关参数则是透过如下画面来进行配置的:



从上面的图形, 不难看出, 有如下七项需要我们来设置:

➤ **NVRAM Type Selection:**

目前此下拉菜单中, 有三组 NVRam 可供选择: SPI Nor, SD Card, eMMC, 在这里, 因

为是对 eMMC 进行操作，所以此 option 应选择 eMMC。

➤ **Header Copies:**

为了防止 Boot Area 受外界干扰而被破坏,造成 Boot 失败。新版 Tool 有对 Boot Header 及 Boot Loader 进行备份，一般，Codepacker 包出来的 Boot Header 大小为 eMMC 的一个 Sector，即 512Byte,所以用户所填的份数，即代表占用 eMMC 的 Sector 数。

➤ **NVRAM Total Size:**

这个栏位，是用来告诉 FW Code 当前 eMMC 的总容量大小,以 MByte 为单位。

CodePacker 目前支持 1024MB, 2048MB, 4096MB, 8192MB, 16384MB, 32768MB 共 6 组值. 用户可以透过从下栏菜单中选取。

➤ **DRAM Size:**

此栏位是指 GPL327xx 小机上的对应的 Dram Type 的容量，以 Mbit 为单位，Tool 目前支持 16Mb,

64Mb,128Mb, 256Mb, 512Mb, 1024Mb, 2048Mb, 4096Mb 共 8 组值, 8Mb = 1MB。同样，用户只需要点选下栏菜单就可以找到相应的值。

➤ **DRAM Type:**

此栏位主要是指指定后续做 Calibration 的 Ram 的类型，以及烧录时 ISP Bin code 指定的存储器。

➤ **Boot Loader Files:**

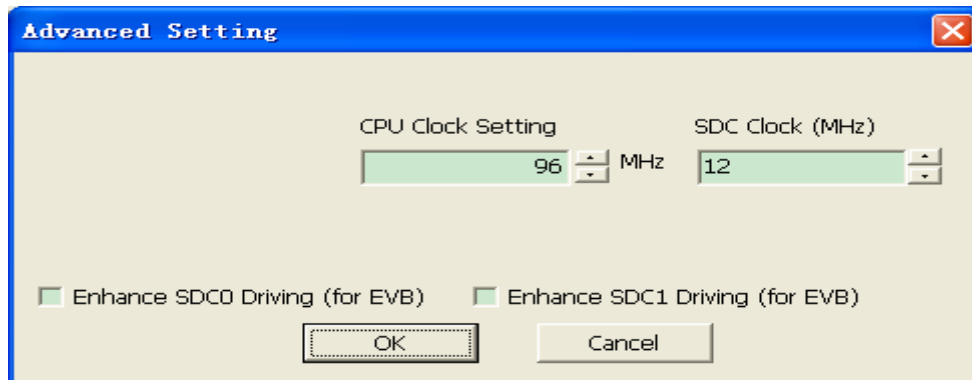
此栏位是选取 First BootLoader 文件，通常是由 Tool 根据 IC 系列及 NV Ram Type 来决定使用哪个 First Bootloader 二制档，此档由 Generalplus 相关人员开发维护，建议用户使用此默认档。

➤ **Boot Loader Copies:**

同前面的 Boot Header Copies 相似，都是用来指定备份数的，只不过，此栏位是用来表明 First Boot loader 的备份数的。Rom code 会根据这些 copies 来进行恢复，从而有效降低 eMMC 的不良率。

12.1.2 Boot Area 高级设置

Boot Area 除了基本的设置外，还增加了一组高级设置。通常情况下，这些高级设置选项，都会产生一组默认值，用户也不需要理会它。eMMC 的高级设置画面如下：



里面有三个选项可供用户设置，分别为 CPU clock setting, SDC Clock, Enhance SDC0/1 Driving(for EVB)

➤ CPU Clock Setting:

CPU clock setting 的设置范围为 4~118MHZ，默认值为 108MHz，用户可以自己填入值.. 对它的设置直接影响 CPU 跑 code 的速度。

➤ SDC Clock:

此选项是用来调整 eMMC 的时钟，其设置范围为 0~50MHz，默认值为 12MHz，对它的调整会影响 CPU 对此 Flash 存储器的存取速度。

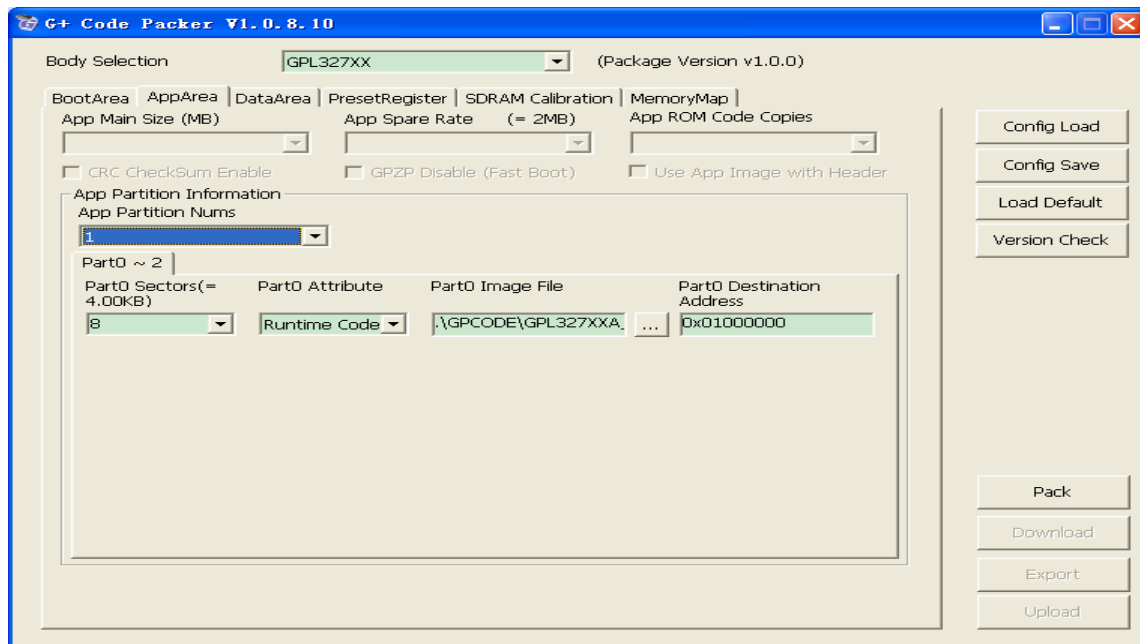
➤ Enhance SDC0/1 Driving(for EVB):

此栏位若勾选则表明增强 eMMC0/1 的驱动电流，当用户 Evb 上 I/O 口与 IC 之间或 eMMC0/1 之间的连线太长或太短，就可以适当调整此 Option，来增强 eMMC I/O 口的输送信号或降低线路间 EMI 干扰的能力。

对所有的选项设定完成后，点"OK"按钮，使当前修改生效；点"Cancel"按钮，取消当前设置。

12.2 App Area 配置

点选 App Area 标签，切换到 App Area，UI 长得样子如下：



如上图，此页面主要是对 App Partition 进行配置，一是设定分区个数；二是为每个分区配置相应的参数。

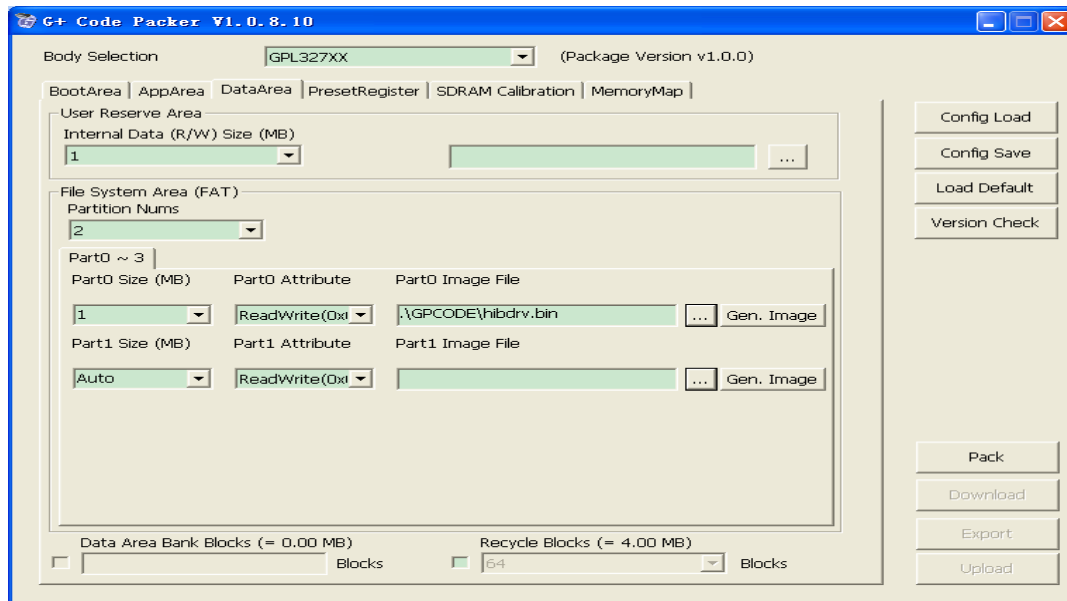
App 分区信息参数配置如下：

- **App Partion Num:** 此选项表明用户当前设定的 App 区的分区数。目前设定范围 1~16，共 16 组值，也就是用户最多可配置 16 个分区。
- **Partion N Size:** N 的范围从 0~15，它是表明当前这个分区的容量大小，以 Sector 为单位，通常情况下，当我们选定分区镜像档案后，Tool 会自动帮我们将相应的档案大小填入此栏位。有一种情况是，用户可以不选取档案，此时，该栏位会被置为 auto，如果用户想指定此分区大小，可以直接输入该值。
- **Partion N Attribute:** 同理，N 的范围是从 0~15，它表明为当前镜像文件是什么类型的文件，目前 Tool 提供 Runtime Code, Resource Part, FastBoot Bin, Quick Image, Hibrate Image, Image Flag, Customize BtLdr 共 7 种类型的 App 镜像文件。当 First bootloader 在搬 App 区的 code 时，此参数就显得尤为重要。
- **Partion N Image file:** 此参数是让用户来选取磁盘上的分区镜像文件。
- **Partion N Destination Address:** 此参数是指定当前分区文件，将来会被摆放在 SDRam 哪个地址来跑。

依此方法，对指定数目的分区进行配置，直到所有分区信息都配置完成。

12.3 Data Area 配置

在 App 区的分区信息设定好后，可以根据需要切换到 Data Area。其画面如下：



数据区的配置主要有由用户预留区和文件系统区两部分组成,暂不支持 **Recycle Blocks** 的设置。

➤ 用户预留区：

此区域主要是在数据区域划分一段存储空间，给用户自行使用的。用户可以预先配置一个文件，也可以不为此存储空间指定文件。不论怎样，用户都可以透过 **eMMC Driver** 中诸如 **Read Sector / Write Sector** 等接口函数来操作此片区域。

➤ 文件系统区：

此区域的配置与 **App** 分区信息的配置比较类似，首先去选定当前用户所期望配置的逻辑分区数，再去对每个具体的逻辑区进行信息的设定。下面简要介绍逻辑区的设定项：

Partition Nums:此项主要是指定逻辑分区数目，共 8 个分区，范围从 0~8，0 表示不分区，也即不设定数据区的分区信息。

PartN Size: N 的范围从 1 到 8，此项主要是指定当前分区的存储容量，在不选中镜像文件情况下，默认为 **Auto**，用户也可以自己手动填写此值。

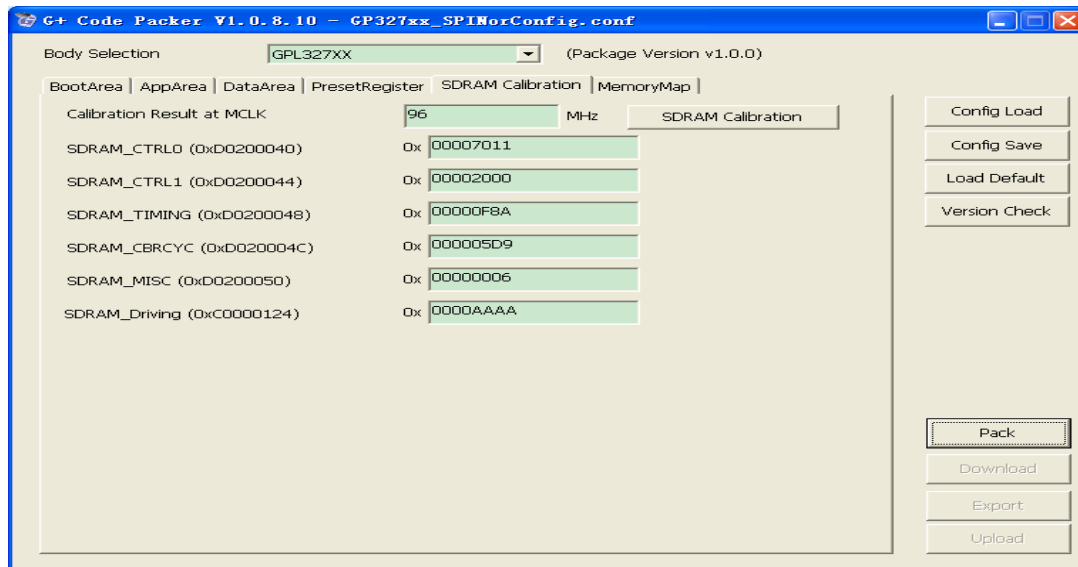
PartN Attribute: N 的范围从 1 到 8，此项主要是指定当前分区的属性，目前有只读，只写，读写三种选项。

PartN Image File:N 的范围从 1 到 8，此项主要是指定当前分区的镜像文件，有两种方式来填充此项，一种是点选“...”按钮，打开文件对话框，在本地机上选二进制文件；另一种方法，则是点选“**Gen.Image**”按钮来透过 **G+FatImageTool** 来生成此档案，至于 **G+FatImage Tool** 的使用请参看其 **User Guide**。

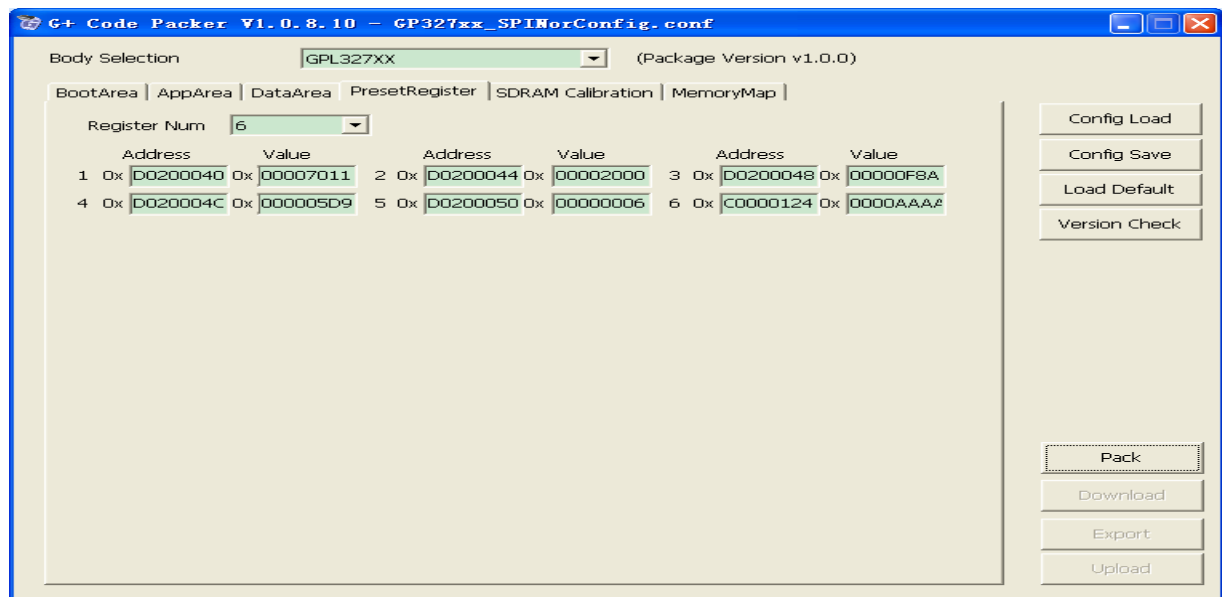
一般来讲，用户保留区与数据分区虽同属数据区，但提供给用户的用途不同，前者主要是透过 **eMMC** 的驱动程序中读写接口函数来直接操作此区域；后者则是透过文件系统来操作各个不同的逻辑分区。

12.4 Pre-Register & Dram Calibration 设置

通常情况下，我们都是先对 GP327xx 的 IC 做 Dram Calibration，然后，再切回到 Pre-Register 页面，来查看相应的寄存器的值。在进入到烧录代码前，用户得确保有做 Dram Calibration 的动作。切换到 Dram Calibration 的标签，可看到 Dram Calibration 的 UI 如下：



用户切换此页面后，点选“SDRam Calibration”按钮，进入到 G+ SDRAM Calibration Tool，透过此 Tool 对 SDRam 的 Register 进行 Calibration。至于此 Tool 的使用，用户就要参考下 G+ SDRAM Calibration Tool 的用户指南。SDRam Calibration 成功后，切换到 Pre-Register 标签项，可以看到 Calibration 后的值被正确反映到 Pre-Register 页面中，其内容长得如下样子：

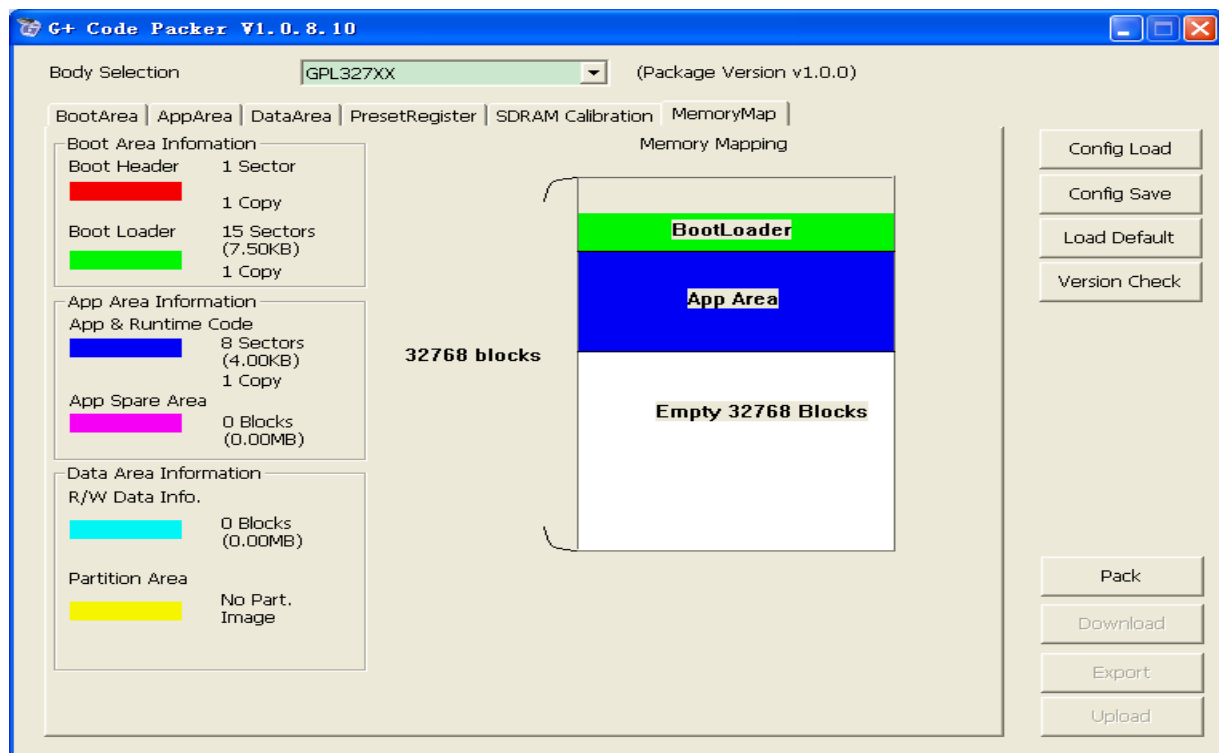


从上图可以看出，它由两部分组成，一部分是寄存器数目；另一部分是第组寄存器的地址及地址单元里的值。当然，用户如果因某些原因，发现 Dram calibration 没有达到预期效果，或者为调整

开机阶段的设定(ex. Clock 或 IO 状态)等情况，这时就可以直接在此页面，透过手动修改的方式来增加相应寄存器的设定。

12.5 Memory Map

前面对 Boot 区，App 区，Data 区进行配置后，切换到此页面，即可看到根据前面配置所映射到 eMMC 存储器上的布局情况。如下图：



目前看来，此页面里的数据反映的有些问题，这在后续版本中会进行修正，且并不影响数据在实际存储器的分布情况。

12.6 按钮功能

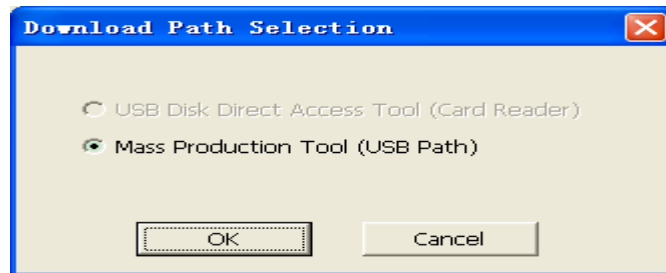
所有的配置都完成后，用户可以点选“Config Save”按钮，将当前的配置保存起来，这样，下次就可以透过“Config Load”按钮来加载之前保存好的配置文件，这里要注意的是，新版本的 Code Packer Tool 中尽量不要加载旧版本 Tool 保存下来的配置文件，因为很有可能二者并不兼容，特别是新旧版本号隔得太远，兼容性就更欠佳。

Load Default:此按钮主要是用户在不需关闭 Code packer 的情况下，清除用户当前配置，恢复到 Tool 初始状态。

Version Check:此按钮主要是用来查看各个 IC 系列所发布的 HDB 档案的版控信息。

Pack:此按钮主要是将配置好的一些信息，按照一定的规则，包出烧录程式所需要的文件。

Download: 此按钮的作用是将烧录工具 G+ Mass Production Tool 呼叫起来，其画面如下：



此种方式则是将 eMMC 卡插在 EVB 上的 eMMC 卡 slot 上，由透过 GP327xx/GPDV6xx 等 IC 来对它进行读写。

Export:此按钮则是在将包出来的文件及烧录程式一同导出到磁盘某个地方，这样，只要打开 G+ Mass Production 量产工具，就要可以进行档案的烧录，这个功能主要是方便工厂进行量产之用。

13 量產程序 (USB Mass Tool Portable) 說明

13.1 USB Mass Production Tool Export

為了能方便將產生的 BIN 檔(ROM Image)順利帶到工廠進行量產程序，G+ Code Packer 提供了 Export 功能來完成這項操作，注意此功能在 GPL326XX 僅支援 Nand Flash 量產燒錄，GPL329XX 可支援 Nand Flash/SDCard/eMMC 量產燒錄。操作步驟如下：

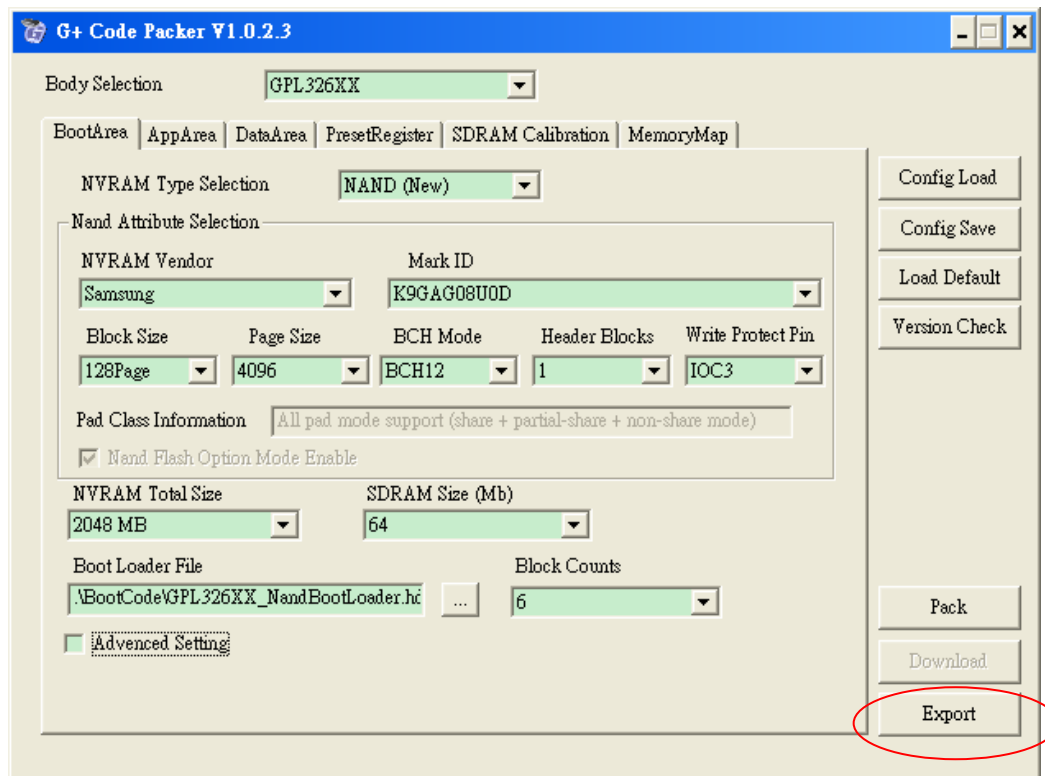
Step 1 : Pack

執行 Pack 動作之後，首先點擊 Download 按鈕，使用 Mass Production Tool 將產生的 BIN 檔燒錄到裝置，測試產生的 BIN 檔是否能正確執行。Download 之後，Export 按鈕將被激發。

Step 2 : Export

點擊 Export 按鈕，選擇要 Export 的路徑後點擊 ok，開始進行 Export 動作。Export 動作會將所有產生的 BIN 檔及其他所需檔案放到所選擇的路徑下，只需要將此目錄資料帶至工廠，即可進行量產動作。

註：需執行 Export 目錄下的 MassProduction.bat 呼叫 Mass Tool 進行量產程序。



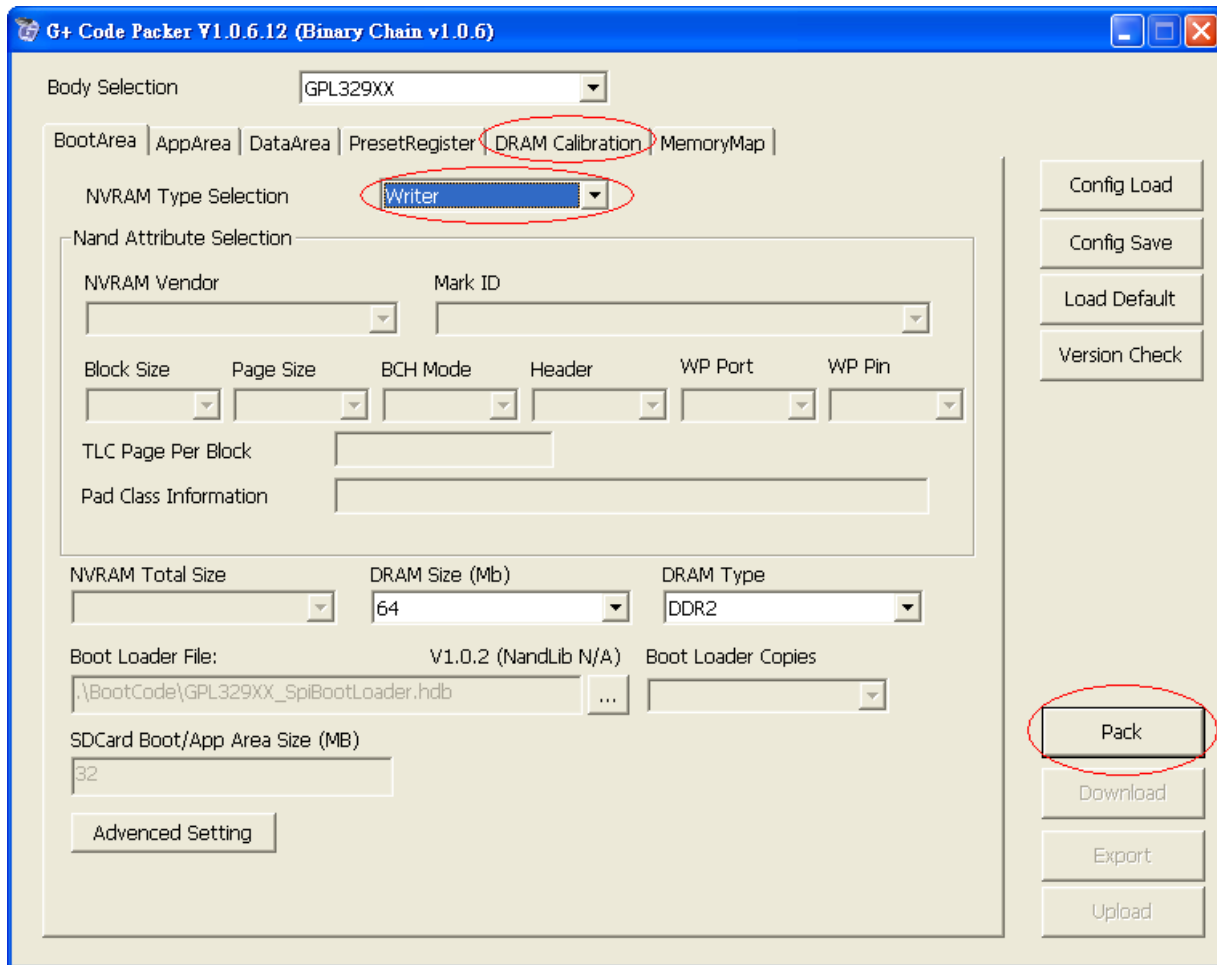
13.2 SDC Production 生產卡製作方式

透過製作 SDC Production 生產卡方式可以生產各式 NVRAM，目前 GPL326XX 僅支持 Nand Flash 生產，GPL329XX 可支持 Nand Flash/SDCard/eMMC 等生產。

Step1. Writer DRAM Calibration:

Writer 本身的生產環境參數(例如 DRAM 參數)應該要先做過校正，確保 Writer 能正確運作並進行 NVRAM 生產，目前只有 GPL329XX 提供此功能，如果要進行 Writer 的 DRAM Calibration，首先在 Tool 上 NVRAM Type 選擇 Writer，接著切至 DRAM Calibration 頁面進行 DRAM 參數校正，DRAM 校正完成後按 Pack，Tool 會儲存 DRAM Calibration 參數，以便自動整合到後續 SDC Production 生產卡製作。

注意：如是 GPL326XX 可跳過此 step.

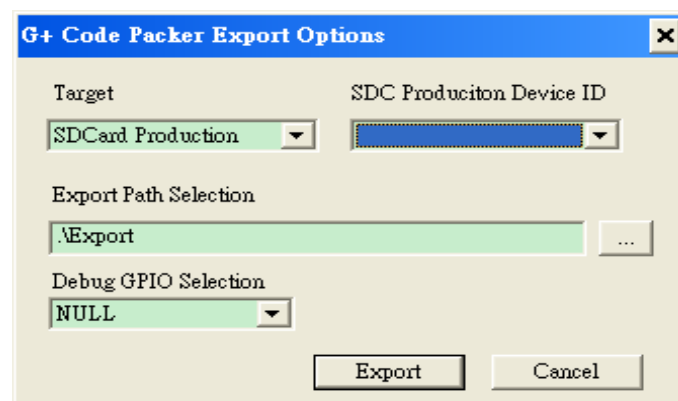


Step2. 進入 Export 模式:

壓下 Export Button , Target 選則 SDC Production

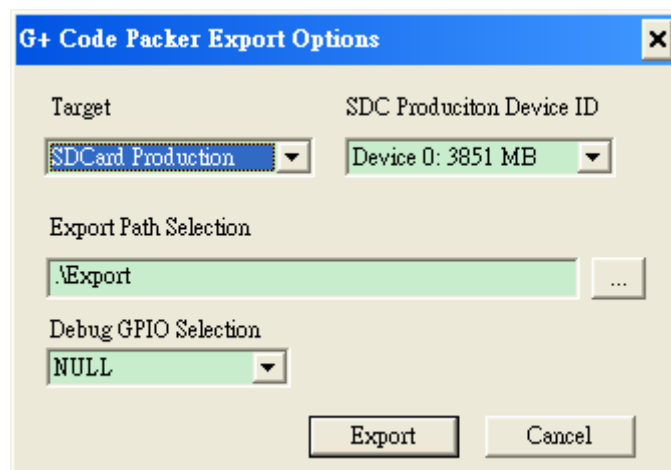
目前僅開放以下兩種情況使用

- 1) GPL326XX Body, NVRAM Type 為 Nand Flash
- 2) GPL329XX Body, NVRAM Type 為 Nand Flash/SDCard/eMMC 始能使用

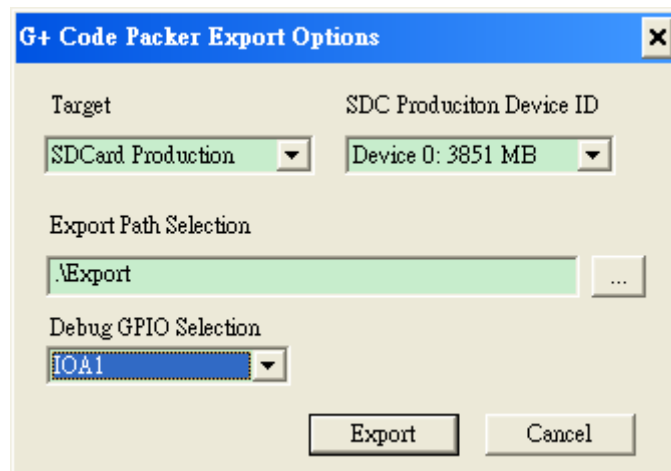


Step3: 使用規則:

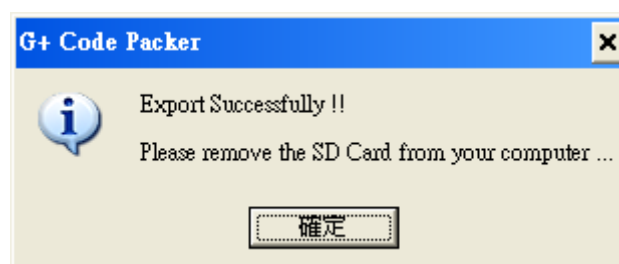
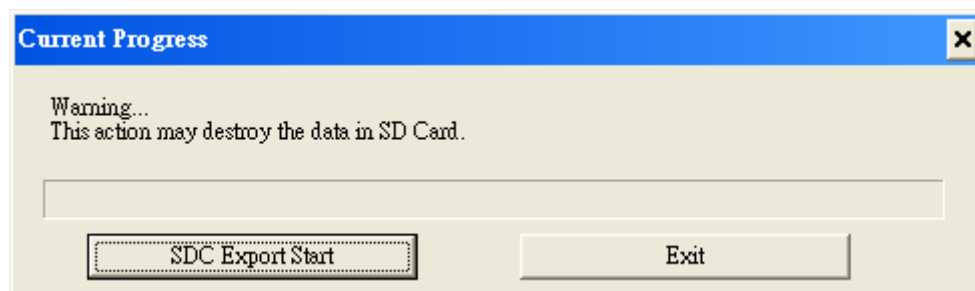
因為若將生產 Binary file 燒到 SDCard 產出 SD production card, 該 SD 卡片原始資料會被破壞, 因此請先備份好原本卡片內容再行生產, 生產卡 generation 工具要求卡本身為生產卡 (已被該 Tool 規劃過), 或是 FAT 卡但於 Root path 需帶有 SDC_Production_Start.bin 之檔案 (無關內容, 只認檔名), 此兩種卡片才能被該 tool 做生產卡產出工具。並強烈建議採用 Single Lung 之單功能讀卡機(只有 SD 卡座)

**Step4: 抓到生產卡或準生產卡:**

Step5: 可選擇採用 **Debug GPIO** 供生產過程閃燈使用，藉以得知生產結果與進程



Step6: 生產卡產出開始 (可連續生產，每張卡符合 **Step2** 之規則)



14 錯誤訊息說明

若使用者對工具的操作流程錯誤，可能造成工具發生不可預期的錯誤。使用者亦可能會忘記設置某些選項，或是疏忽了選項之間彼此存在的關係(ex. 包出來的 **code size** 大於所規劃的 **area size**)。若發生了這些情形，可能造成產生出來的檔案錯誤或是燒錄至裝置上無法正確執行等狀況發生。

為了儘可能的避免這些情形發生，工具做了一些保護的措施。當使用者進行了錯誤的操作或設置時，工具會送出錯誤訊息告知使用者此操作有問題，則使用者可以依據錯誤訊息更正錯誤的地方，以確保整個流程及產生檔案的正確性。

工具送出的錯誤訊息以及其對應的生成原因可參照下表所列：

錯誤訊息	對應描述
Open selected file failed	選擇的 resource file 無法被開啟
Open RunTime ROM Code File Failed	開啟所選擇的 RunTime ROM code 失敗
Please input your Runtime ROM Code Size	當勾選 RunTime ROM Code size 卻未輸入任何 size 資訊
Runtime Code + Resource file size is greater than APP Area Size	使用者所選的 App Area 大小大於所規劃的 App Area Size
The selected "APP Main Size" is too small please modify the setting of "APP Main size"	備份數選擇過大，造成整體 App Area 大小大於規劃的 App Area Size
The selected size is greater than remained.	改變了 App Main Size ，且選擇了過小的 size
The selected "spare rate" is invalid.	選擇的 Spare Rate 錯誤 (可能由超過交換塊的最大值等原因造成)
Config File Not Found	找不到配置檔 (可能由 PackConfig 資料夾不存在等因素造成)
Please select File First	對於 Resource File List 中的檔案進行操作，卻未選擇任何檔案
The element can't be moved down.	Resource File List 中所選擇的檔案不能上移
The element can't be moved up.	Resource File List 中所選擇的檔案不能下移
BootLoader File size is greater than Boot Block Count	所選擇的 BootLoader file 大小大於 Boot Block Count
Open R/W image file failed	開啟所選擇的 R/W image file 失敗
The size of selected R/W image file is greater than R/W area size	所選擇的 R/W image file 大小大於所規劃的 R/W Area size
Can't open the image file	開啟所選擇的 Image file 失敗
Total size is greater than selected NVRAM total size	整體規劃大小(Boot Area + App Area + Data Area)大於選擇的 NVRAM total size
Select NVRAM Total Size First	NVRAM Total Size 尚未被選擇
Create Dialog Failed!	開啟主視窗失敗 (可能由於開啟過多視窗或是記憶

錯誤訊息	對應描述
The Application will be closed	體不足等因素造成)
Download tool is in active	由 G+ Code Packer 呼叫執行的 Mass production tool 仍處於執行狀態
Load Packing DLL Fail	載入 Pack 所需之 DLL 失敗 (找不到 BootAreaPacker.dll 或 AppCodePacker.dll)
Load BootAreaPacker.dll failed	載入 BootAreaPacker.dll 失敗
Load AppCodePacker.dll failed	載入 AppCodePacker.dll 失敗
Input Address is invalid	輸入的位址不存在 (不為合法的 16 進制表示)
The max. number of digits is 8.	最大的輸入位數為 8 位
Open selected path fail! Please select another path	開啟所選擇的路徑失敗 (可能由於目錄不存在或是創建路徑失敗造成)
All options must be selected	有某個或某些必選的選項未被設置
Save User Options Failed	儲存目前的設置失敗(可能 UserOption.ini 不存在)
Write Options to Pack Config Failed	儲存使用者設置至配置檔失敗 (可能由於 PackConfig 目錄下對應檔案不存在)
The bootloader file is not a supported format!	預設的 boot loader 版本或是其餘檔案資訊錯誤
Bank Size + Recycle size should be smaller than page size!	Data Area 中所選擇的 Bank Size 與 Recycle Size 總和必須小於 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".	在執行 SDC Production 時，必須使用合法的 SD Card 裝置。 合法裝置的定義如下： 這張 SD Card 已經是張生產卡 或者 這張 SD Card 中包含了"sdc_production_start.bin"這個檔案
Read version config file failed!	讀取保存 Version 資訊的配置檔 (.\PackIni\DefaultBinary.ini)失敗