SF Software User Manual

For SPI NAND Flash

Please read the instructions carefully before use.

Table of Contents

Important notice:
This document is provided as a guideline and must not be disclosed without consent of DediProg. However, no responsibility is assumed for errors that might appear.

DediProg reserves the right to make any changes to the product and/or the specification at any time without notice. No part of this document may be copied or reproduced in any form or by any means without prior written consent of DediProg.

I. Introduction

This user manual illustrates the usage of DediProg SF Software. The software with SPI NAND flash can work with SF600Plus-G2 and SF700 programmers. Get more information about DediProg products and how to use them.

II.Software Installation Guide

Please refer to the products' specification, presentation, and application notes on our website: www.dediprog.com

2.1 Operating System Requirement

Windows Vista/7/8/8.1/10/11
Windows Server® 2008
Support both 32 bit and 64bit OS

2.2 USB Installation

2.2.1 Insert the installation CD or download the installation software from www.dediprog.com/download
2.2.2 Execute SFx.x.x.x.msi file and follow the setup instructions to finish installation.

The versions after Windows 8 please refer to the "**USB driver Installation Guide (Win 8 / 8.1/10/11)** ".

III. DediProg SF Software Engineering GUI

DediProg SF software with SPI NAND flash is suited for SF600Plus-G2 and SF700. The software can only be used for programming serial flash memory as well as downloading the configuration contents to the reference SPI Flash embedded memory in SF600Plus-G2 and SF700 for standalone programming purpose. After the software and USB driver are installed, please follow the steps below before running the software.

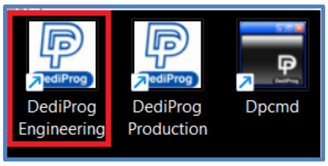
Four software icons will appear on your desktop after installation.
Icon "DediProg Engineering" is the engineering GUI, "DediProg Production" is the production GUI, "Dpcmd" is the command line interface and "DediProg Help" is the user manual.

3.1 Environment Preparation

3.1.1 Connect the programmer to the PC through an USB cable.

- For ICP programming, connect the ICP cable to the application (please check the specification in case ISP header pin out are not known).
- For socket and standalone programming, connect the appropriate socket adaptor to the programmer and insert a serial flash in the socket.

3.1.2 Double click the DediProg software icon on your desktop.



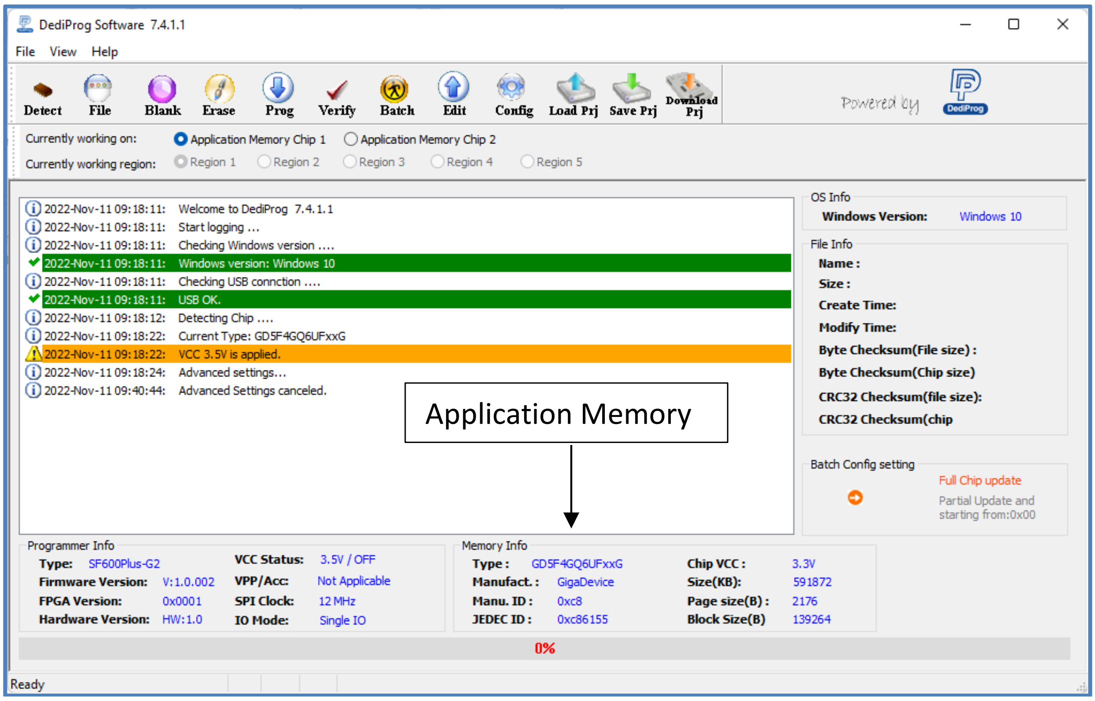
3.2 Identify the Target SPI Flash

SPI Nand Flash Detection

Double Click the DediProg software icon on your PC desktop. The detected Serial Flash information as well as the programmer information will be displayed on the right side of the window.

DediProg software will automatically identify the SPI Nand Flash on the application board or the socket. You do not need to select SPI Nand Flash's location.

※ Note: If you want to work on the second target SPI Nand Flash soldered on the application board, the application board must be designed with proper schematic and the pin outs must match with DediProg ISP pin outs.



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2

Currently working region: ● Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

```
2022-Nov-11 09:18:11:   Welcome to DediProg 7.4.1.1
2022-Nov-11 09:18:11:   Start logging ...
2022-Nov-11 09:18:11:   Checking Windows version ....
2022-Nov-11 09:18:11:   Windows version: Windows 10
2022-Nov-11 09:18:11:   Checking USB connction ....
2022-Nov-11 09:18:11:   USB OK.
2022-Nov-11 09:18:12:   Detecting Chip ....
2022-Nov-11 09:18:22:   Current Type: GD5F4GQ6UFxxG
2022-Nov-11 09:18:22:   VCC 3.5V is applied.
2022-Nov-11 09:18:24:   Advanced settings...
2022-Nov-11 09:40:44:   Advanced Settings canceled.
```

Application Memory

OS Info
Windows Version: Windows 10

File Info
Name :
Size :
Create Time:
Modify Time:
Byte Checksum(File size) :
Byte Checksum(Chip size)
CRC32 Checksum(file size):
CRC32 Checksum(chip

Batch Config setting
Full Chip update
Partial Update and starting from:0x00

Programmer Info
Type: SF600Plus-G2
Firmware Version: V:1.0.002
FPGA Version: 0x0001
Hardware Version: HW:1.0
VCC Status: 3.5V / OFF
VPP/Acc: Not Applicable
SPI Clock: 12 MHz
IO Mode: Single IO

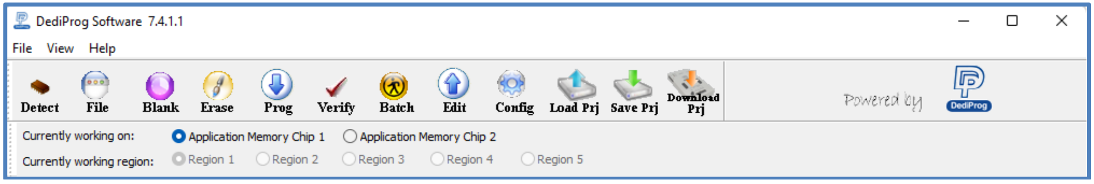
Memory Info
Type : GD5F4GQ6UFxxG
Manufact. : GigaDevice
Manu. ID : 0xc8
JEDEC ID : 0xc86155
Chip VCC : 3.3V
Size(KB): 591872
Page size(B) : 2176
Block Size(B) 139264

0%

Ready

3.3 Tool Bar Description

The tool bar provides all SPI Flash operations.

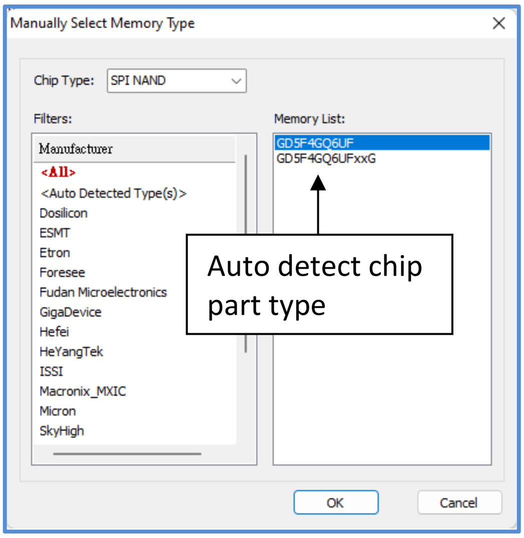


Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2

Currently working region: ○ Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

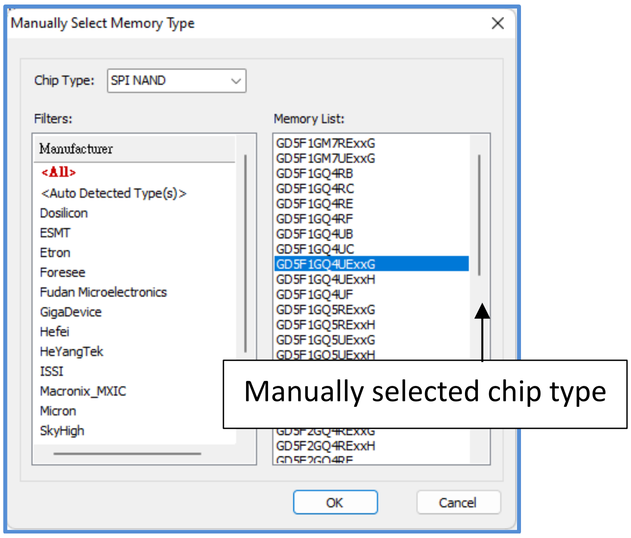
Detect

Detect Chip: when a new SPI NAND Flash is placed, click this button to identify it and perform the operations. The auto detected chip types will be displayed on the right side of the screen. In case you would like to manually select a chip type, move the mouse over the chip manufacturer on the left screen, and then click the chip type on the right screen.



Chip Type: SPI NAND

Auto detect chip part type



Chip Type: SPI NAND

Manually selected chip type

File

There are two ways of loading the programming code: Combine the partition files or set up manually. See "Load File description" for more details.

Blank

Blank check: check if the target serial NAND flash is Blank (All Erased)

Erase

Erase: There are two erase functions, Erase and Force erase. In "Config" setting, user can decide to enable Force erase or disable. Default is disable, and when software close, the erase setting will not be saved. "Erase" function will erase the blocks that are not bad blocks, "Force erase" function will erase the blocks which includes bad blocks, the blocks may be blank, but not necessarily all blocks will be blank. Some chips have factory bad blocks, which cannot be erased even force erase.

Prog

Program: Program the selected image into the Serial NAND Flash.

Verify
Verify the checksum value of the selected image and the programmed Serial NAND Flash content.

Batch
Batch operation: The programmer will perform a pre-configured set of operations such as (erase + program + verify) all together in one click. The configuration can be set by clicking on the "Config" button. The configuration will not change until it is re-configured.
Press start button to allow batch function when running the SF software.

Edit
When click on Edit, the programmer will display the selected file content as default. User can click "read" to read and display the chip contents. See "Edit window description" for more details.

Config
This allows configure advanced settings. See "advanced settings window description" for more details.

Load Prj
Load the existing project to execute the programming operation.

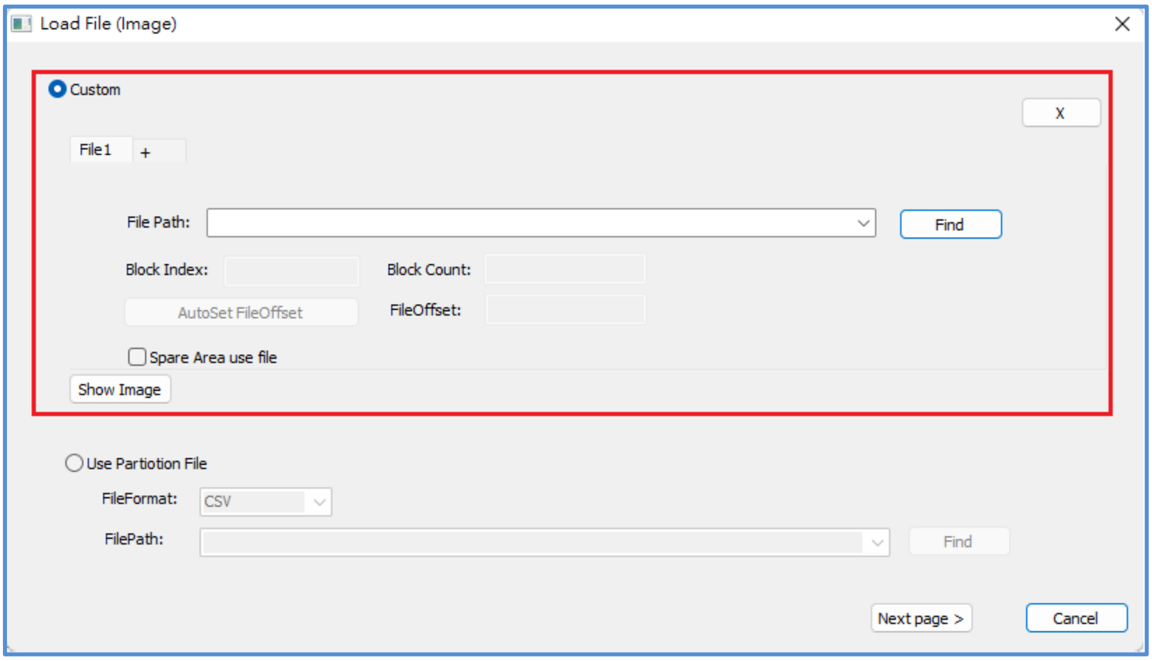
Save Prj
Save all programming settings to a project file for avoid re-setting action.

Download Prj
See Chapter 7- **VI. Stand Alone Mode** for more details.

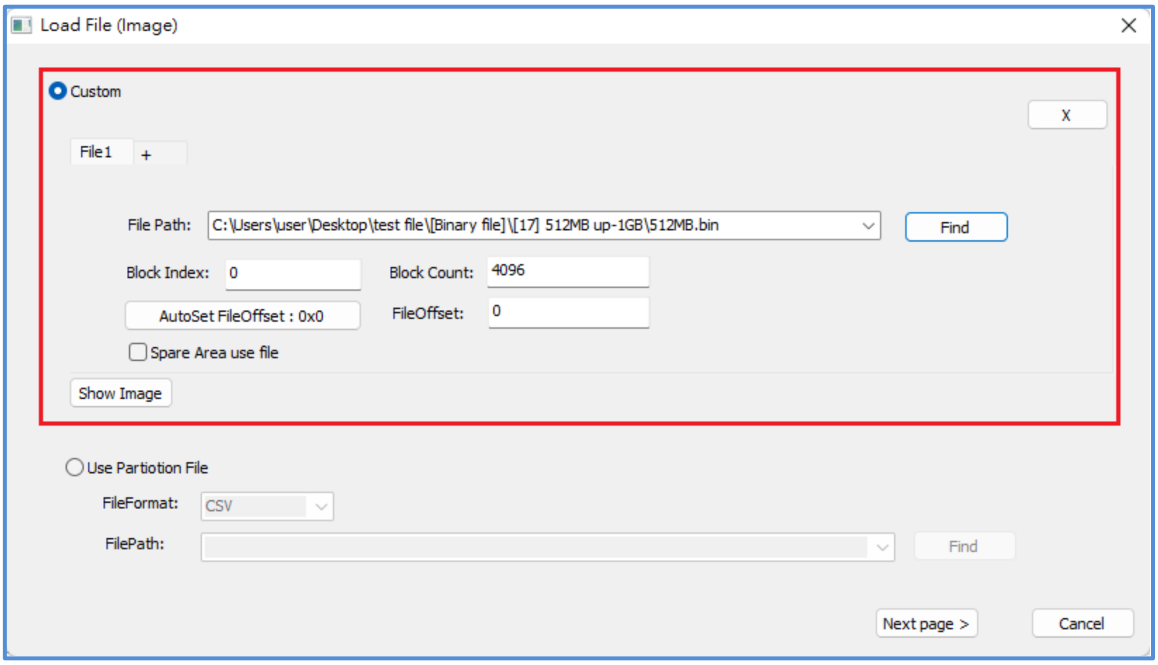
3.4 Load File Description

3.4.1 Custom: Set up Manually

a) Select Custom



b) Function descriptions:



File Path: C:\Users\user\Desktop\test file\[Binary file]\[17] 512MB up-1GB\512MB.bin

Block Index: 0

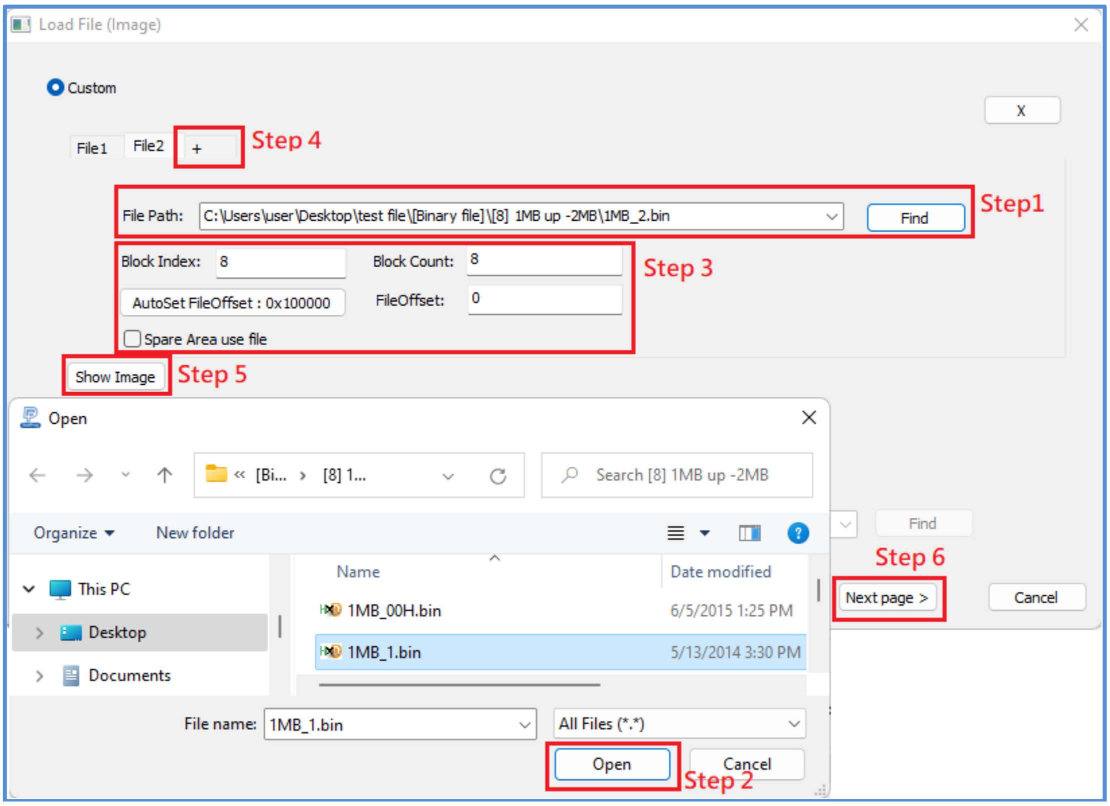
Block Count: 4096

AutoSet FileOffset : 0x0

FileOffset: 0

- **File Path:** The path of the programming file (it can remember the File Path)
- **File Format:** The file format of the image file (Binary Only).
- **File offset:** Assign a file offset address for loading the Buffer.
- **Spare Area use file:** It decides whether the file will include Spare Area or not; check the box if Spare Area is required.
- **Block index:** The beginning of the Block
- **Block count:** The total Block number for programming.
- **AutoSet FileOffset:** When the address of the Image file and the NAND Flash are parallel, you can set up the offset value for Block index. Enter the value, click the AutoSet FileOffset, and then the File offset value will change accordingly.
- **+: Add a programming file.**
- **x: delete the final file**

c) Loading Steps:



Load File (Image)

● Custom

File Path: C:\Users\user\Desktop\test file\[Binary file]\[8] 1MB up -2MB\1MB_2.bin

Block Index: 8 Block Count: 8

AutoSet FileOffset : 0x100000 FileOffset: 0

□ Spare Area use file

Open

« [Bi... › [8] 1...

Search [8] 1MB up -2MB

Name	Date modified
1MB_00H.bin	6/5/2015 1:25 PM
1MB_1.bin	5/13/2014 3:30 PM

File name: 1MB_1.bin All Files (*.*)

Step1: Click the Find button and it will bring up the File Browser.

Step2: Find the programming file that you need, click Open, and then set up the settings.

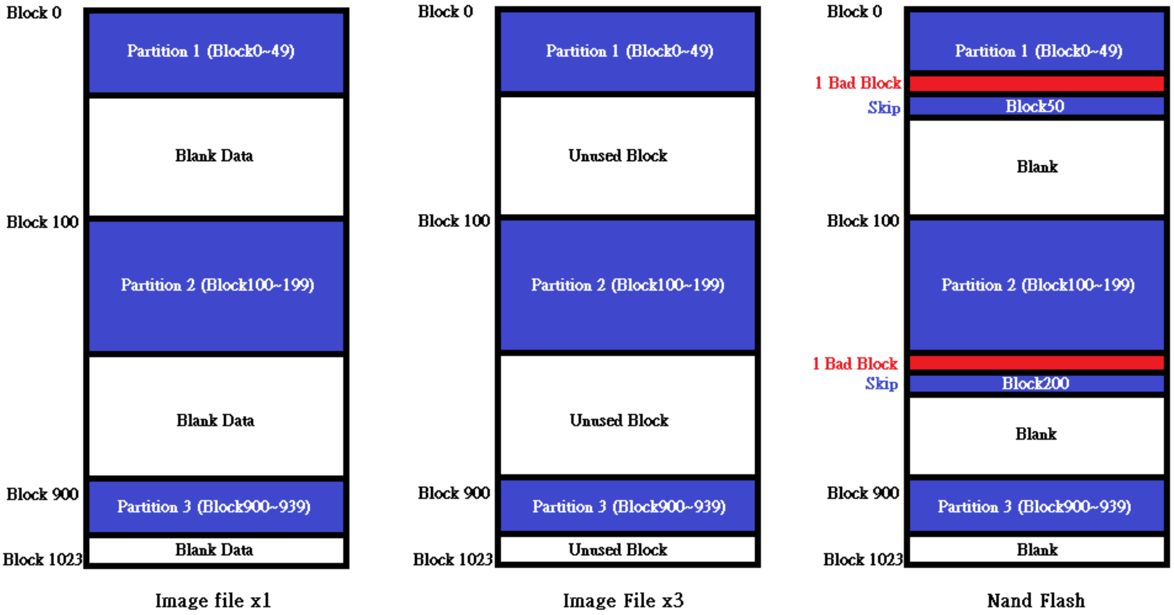
Step3: Confirm the values and the settings.

Step4: Click + to add a file. Please repeat step 1 to 4 for multiple files.

Step5: The file information will show in the window.

Step6: Finish setting and click next.

When you assign BlockIndex and need the File offset to move with it, there is a function called AutoSet FileOffset in DediProg for combining multiple Partition Images into one Image File, so you will only need to load one file Image and set up the BlockIndex separately (Shown as the below image).

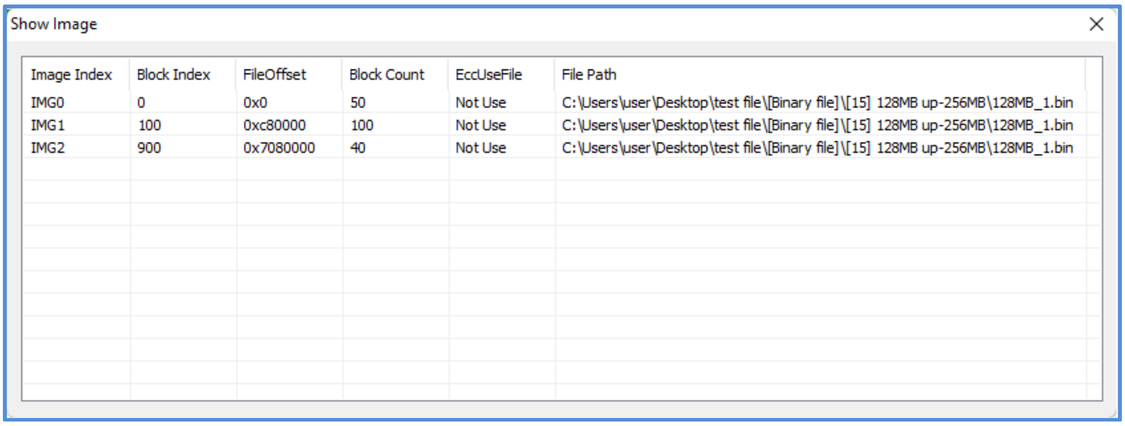


The Image File on the left (Image File x1) is a combination of three Partitions; the software will set up the block division when the file is loaded, which will load the same Image File and able to set up three different settings.

For Example:

Set up Partition 1, BlockIndex=0, BlockCount=50, FileOffset=0.
Set up Partition 2, BlockIndex=100, BlockCount=100, FileOffset=0xCE4000.
Set up Partition 3, BlockIndex=900, BlockCount=40, FileOffset=0x7404000.

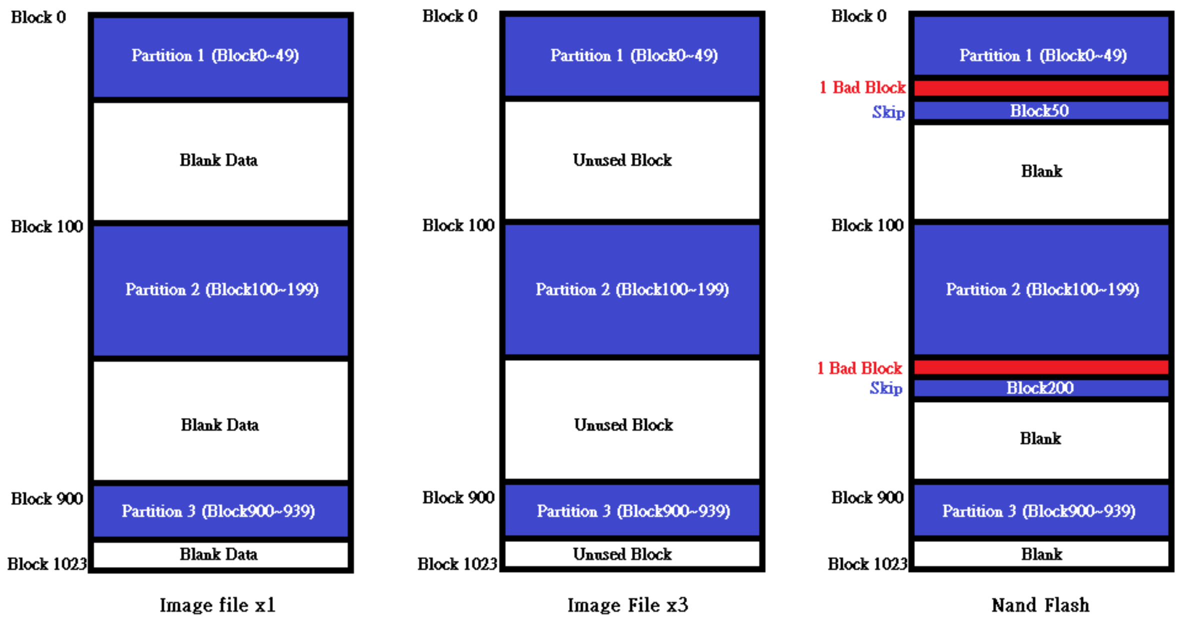
FileOffset will calculate the corresponding address (Byte wide) automatically while setting up the BlockIndex. As for Partition2, when BlockIndex is 100 while 1 Block of the IC = 64pages and 1Page =2112Bytes, then 1 Block is 64*2112=135168Bytes (0x21000). For Block 100, the starting address will be 0x21000 * 0x64=0xCE4000; for Partition 3, it will be 0x21000 * 0x384=0x7404000 (Shown as the below image).



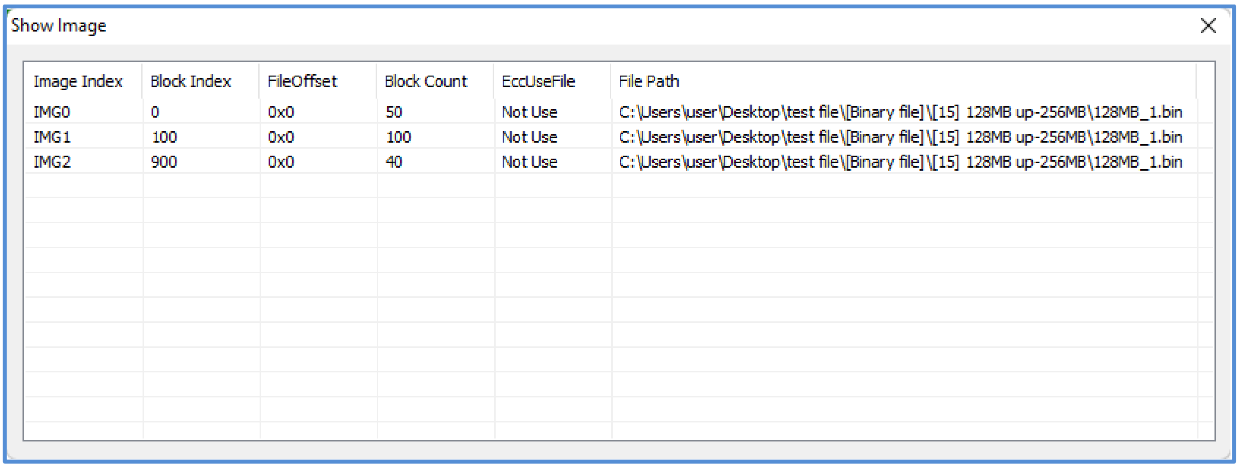
Show Image

Image Index	Block Index	FileOffset	Block Count	EccUseFile	File Path
IMG0	0	0x0	50	Not Use	C:\Users\user\Desktop\test file\[Binary file]\[15] 128MB up-256MB\128MB_1.bin
IMG1	100	0xc80000	100	Not Use	C:\Users\user\Desktop\test file\[Binary file]\[15] 128MB up-256MB\128MB_1.bin
IMG2	900	0x7080000	40	Not Use	C:\Users\user\Desktop\test file\[Binary file]\[15] 128MB up-256MB\128MB_1.bin

The middle image (Image Filex3) shows Partition 1~3 will be divided into three images while loading, but remember to set the FileOffset to zero for BlockIndex in order to avoid data offset.



Set up Partition 1, BlockIndex=0, FileOffset=0, BlockCount will be 50 automatically.
Set up Partition 2, BlockIndex=100, FileOffset=0, BlockCount will be 100 automatically.
Set up Partition 3, BlockIndex=900, FileOffset=0, BlockCount will be 40 automatically.

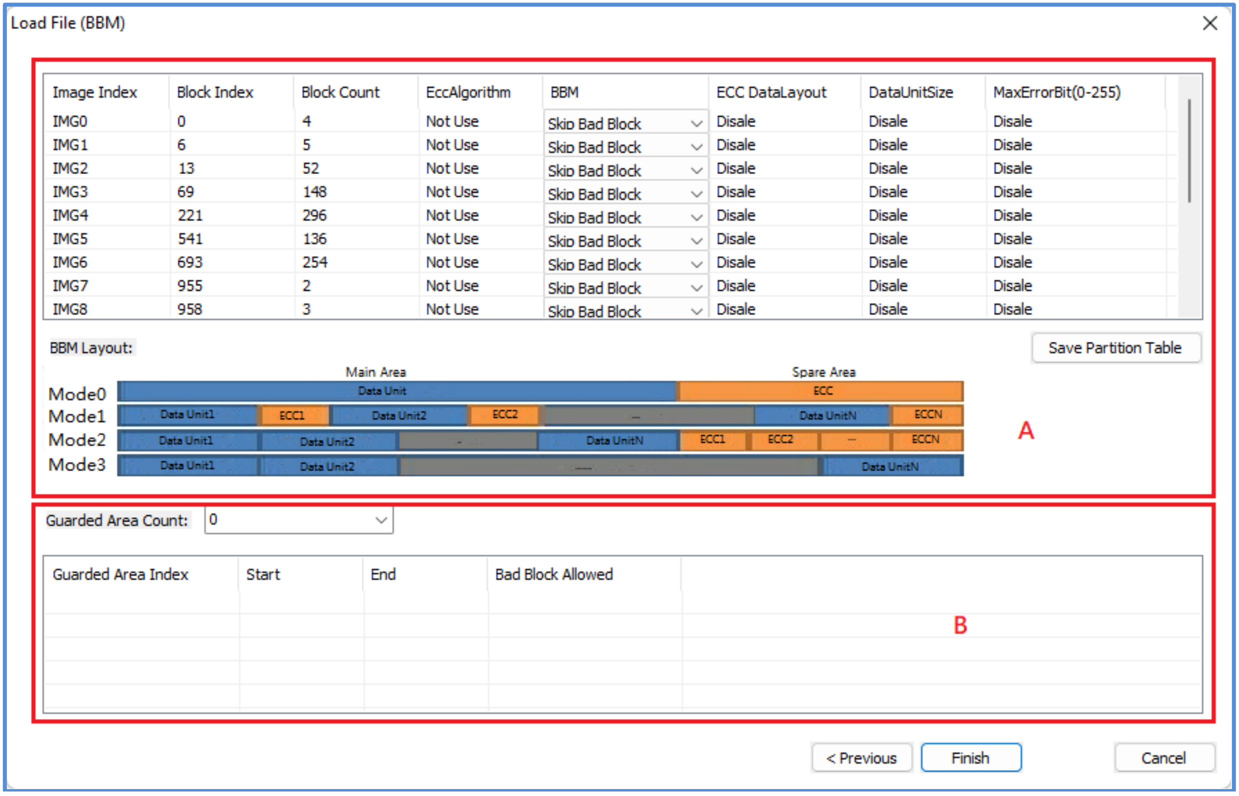


Show Image

Image Index	Block Index	FileOffset	Block Count	EccUseFile	File Path
IMG0	0	0x0	50	Not Use	C:\Users\user\Desktop\test file\[Binary file]\[15] 128MB up-256MB\128MB_1.bin
IMG1	100	0x0	100	Not Use	C:\Users\user\Desktop\test file\[Binary file]\[15] 128MB up-256MB\128MB_1.bin
IMG2	900	0x0	40	Not Use	C:\Users\user\Desktop\test file\[Binary file]\[15] 128MB up-256MB\128MB_1.bin

The image on the right is the listed order after the image has been programmed.
If Partition 1 and Partition 2 have detected bad block separately, then this will be the list order after the Skip process.

Step7: Open BBM Setting



Load File (BBM)

Image Index	Block Index	Block Count	EccAlgorithm	BBM	ECC DataLayout	DataUnitSize	MaxErrorBit(0-255)
IMG0	0	4	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG1	6	5	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG2	13	52	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG3	69	148	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG4	221	296	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG5	541	136	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG6	693	254	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG7	955	2	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG8	958	3	Not Use	Skip Bad Block	Disale	Disale	Disale

BBM Layout:

Guarded Area Count: 0

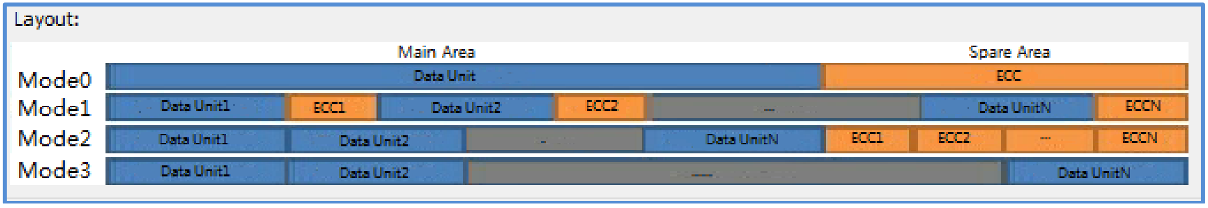
Guarded Area Index	Start	End	Bad Block Allowed

A. **BBM Configuration:**

Set up the BBM according to the actual amounts of the loaded file. As the above image, there are three image files need EccAlgorithm, BBM, EccDataLayout, DataUnitSize, and MaxErrorBit.

- **EccAlgorithm:**
  - If "SpareArea UseFile" is checked on the "Load File-IMAGE" page, it will display:
    - Use File: indicating that the Image File has data containing SpareArea.
  - If “SpareArea UseFile” is not checked on the “Load File-IMAGE” page, it will display:
    - NotUse: Indicates that the Image File does not contain SpareArea data, and the ECC algorithm is not used.
    - BCH8 MDM: This ECC algorithm is only provided to Flex customers. Due to the confidentiality agreement, we cannot provide the detailed file of this algorithm.
- **BBM:** A choice of bad block management. Currently only supports Hard Copy, Skip Bad Block and No management. If you need other kinds of BBM, please feel free to contact us.

- **EccDataLayout:** This is the range definition for Bit Error analyzing. In the below graph, it provides four modes of data layouts.



For example: A page is 2048 + 64.
Mode 0: 2048 + 64
Mode 1: 512 + 16 + 512 + 16 + 512 + 16 + 512 + 16
Mode 2: 512 + 512 + 512 + 512 + 16 + 16 + 16 + 16
Mode 3: 2112

※ **Note: Verify would fail, if Bit Error occurs in the ECC section of Mode0~2 (The orange section).**

- **DataUnitSize:** Set up Data unit size according to different EccDataLayout modes. In this diagram, Mode0 will be 2048Bytes a unit.

- **MaxErrorBit:** Correspond to the volume of the DataUnit, the maximum acceptable Bit Error amounts of each DataUnit. In this diagram, 1 bit is acceptable for 2048Byte. Furthermore, this function cannot apply to verification if the internal ECC is on.

  EccDataLayout, DataUnitSize, and MaxErrorBit are for verification, which will analyze the data accuracy and filter Bit Error to ensure the IC will work normally on board. For example, when ECC needs 528/2bit for CPU, then set EccDataLayout as Mode 3, DataUnitSize as 528Bytes, and MaxErrorBit as 2bit. If you want to filter the NAND Flash more strictly, set the MaxErrorBit to 1 bit. The default setting will depend on the Datasheet definition.

※ **Note: If the ECC enable is selected in configuration setting, then IC will turn on the Internal ECC, which should not have Bit Error. Therefore, MaxErrorBit will be invalid.**

- **Save Partition Table**

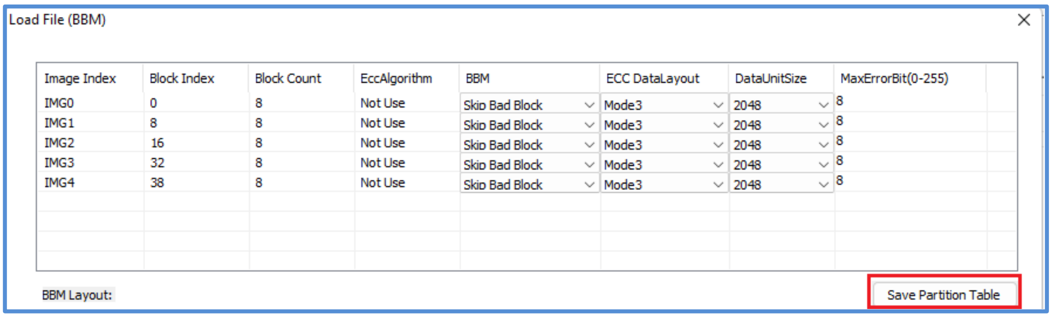
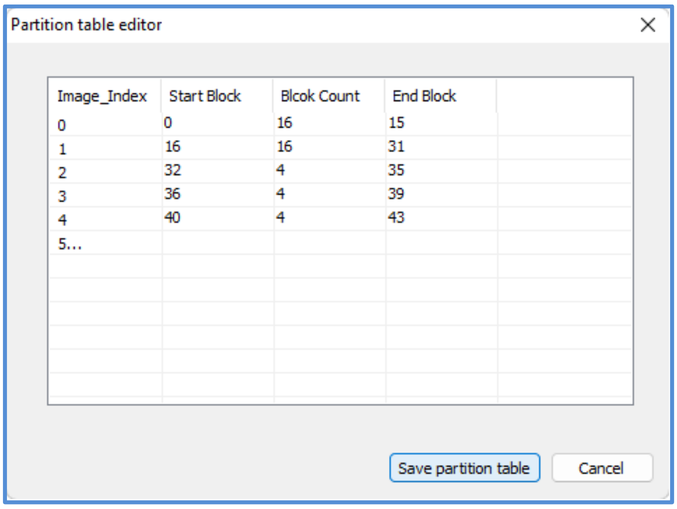
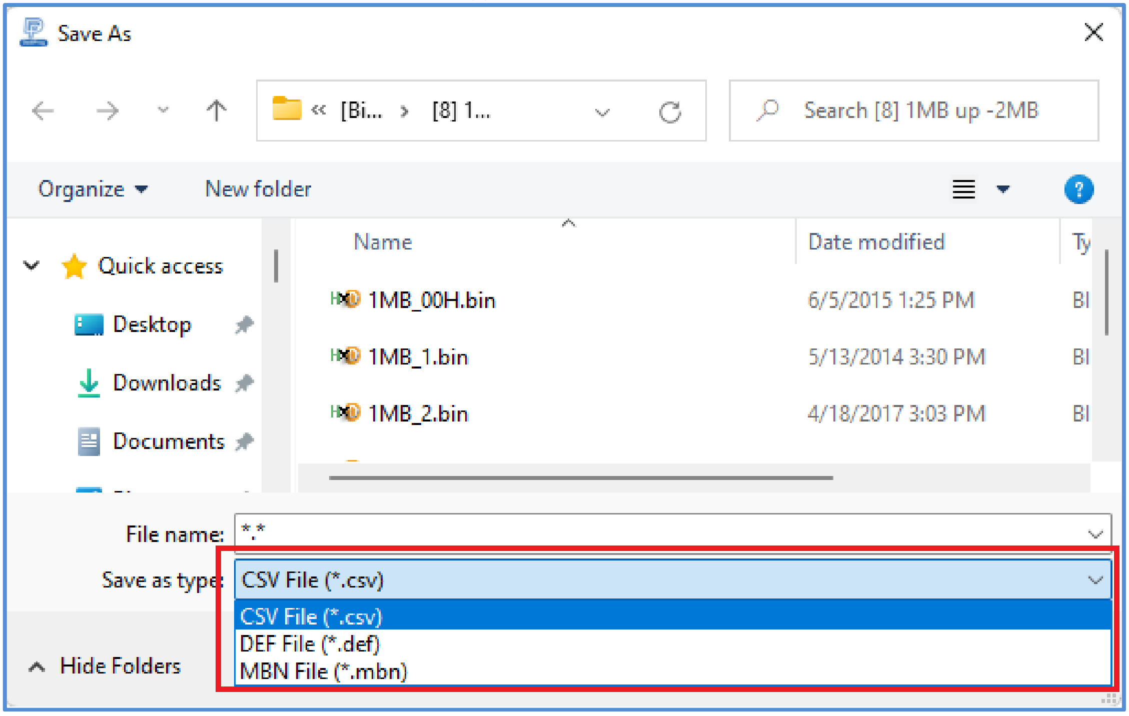


Image Index	Block Index	Block Count	EccAlgorithm	BBM	ECC DataLayout	DataUnitSize	MaxErrorBit(0-255)
IMG0	0	8	Not Use	Skip Bad Block	Mode3	2048	8
IMG1	8	8	Not Use	Skip Bad Block	Mode3	2048	8
IMG2	16	8	Not Use	Skip Bad Block	Mode3	2048	8
IMG3	32	8	Not Use	Skip Bad Block	Mode3	2048	8
IMG4	38	8	Not Use	Skip Bad Block	Mode3	2048	8


Partition table editor

Image_Index	Start Block	Blcok Count	End Block
0	0	16	15
1	16	16	31
2	32	4	35
3	36	4	39
4	40	4	43
5...			

Choose a file type to save


Save As

Name	Date modified	Ty
1MB_00H.bin	6/5/2015 1:25 PM	BI
1MB_1.bin	5/13/2014 3:30 PM	BI
1MB_2.bin	4/18/2017 3:03 PM	BI

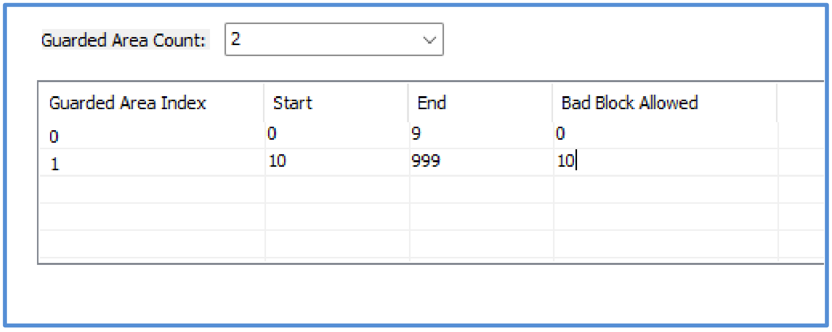
File name: *.*

Save as type: CSV File (*.csv)

- CSV File (*.csv)
- DEF File (*.def)
- MBN File (*.mbn)

B. **Guarded Area Count**:

This function can set up a range for block and detect the maximum acceptable value for bad blocks. In this chart, there two sets of ranges; the first condition does not allow any bad block from Block0 to Block9 while the second condition only allows ten bad blocks from Block 10 to Block 999. Then, during the process of creating BBT, the software will determine accordingly. If the above conditions are true, then the IC will be listed as NG parts.



Guarded Area Count: 2

Guarded Area Index	Start	End	Bad Block Allowed
0	0	9	0
1	10	999	10

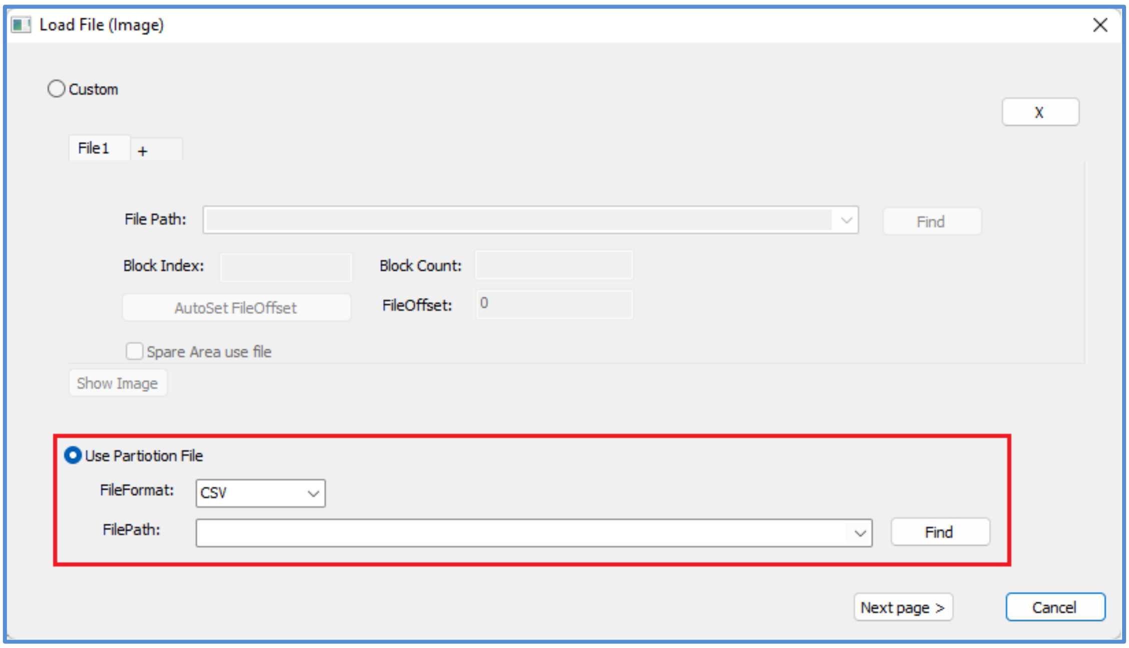
※ **Note: Since NAND Flash programming requires configuration of BBM and ECC, so if you cannot find any suitable BBM and ECC, please contact DediProg for further evaluation.**

3.4.2 Use Partition File:

If your Partition file is in CSV (Comma Separated Values Format), DEF (Group Define File Format), or MBN (Qualcomm Multiply Partition Format), then you can select this option for loading the Image File **and Partition File.**

For example, load a Partition table that is in MBN format (Please go to chapter VI for detailed definition).

Step 1. Load Partition table file



Load File (Image)

○ Custom

File1 +

File Path:

Block Index:

Block Count:

FileOffset: 0

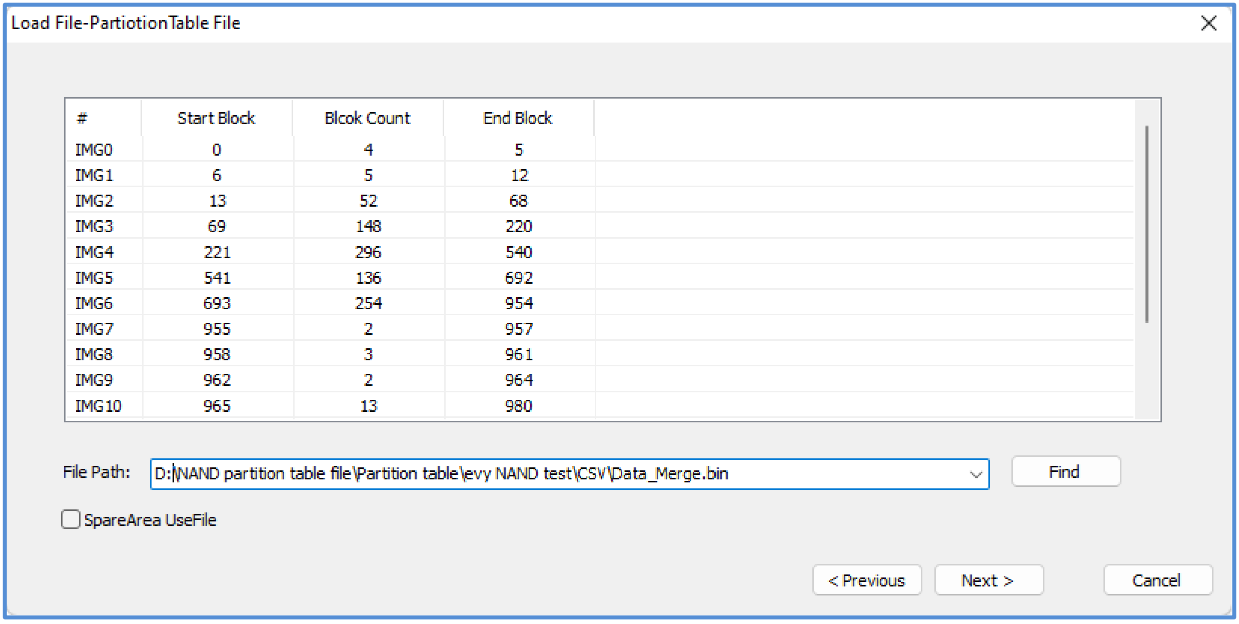
● Use Partiotion File

FileFormat: CSV

FilePath:

Step 2. After the file is loaded, it will show the definition of each Partition. For FilePath, choose the image file that you are going to program; as for File Checksum, choose a method for Checksum calculation.

SpareArea UseFile: It decides whether the file will include SpareArea or not, check the box if SpareArea is required.



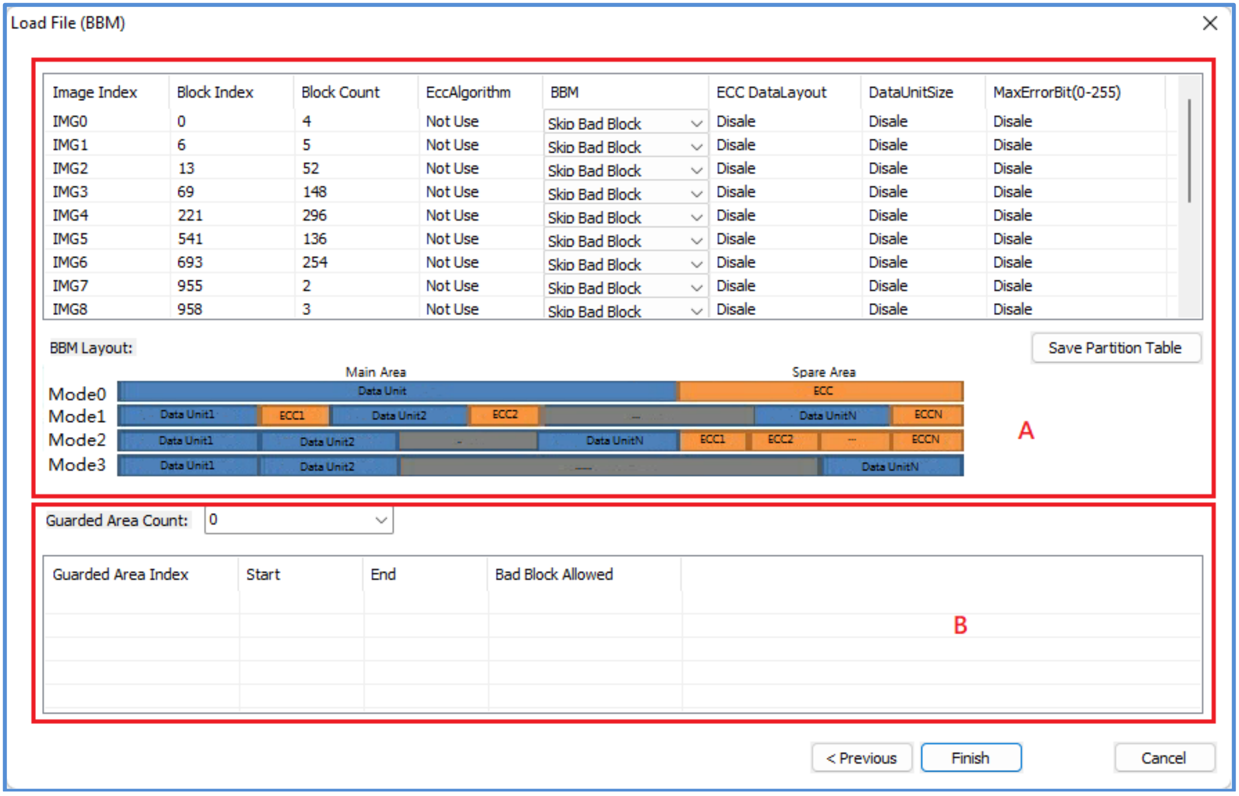
Load File-PartiotionTable File

#	Start Block	Blcok Count	End Block
IMG0	0	4	5
IMG1	6	5	12
IMG2	13	52	68
IMG3	69	148	220
IMG4	221	296	540
IMG5	541	136	692
IMG6	693	254	954
IMG7	955	2	957
IMG8	958	3	961
IMG9	962	2	964
IMG10	965	13	980

File Path: D:\NAND partition table file\Partition table\evy NAND test\CSV\Data_Merge.bin

☐ SpareArea UseFile

Step 3. Set up BBM and Guarded Area Conditions



Load File (BBM)

Image Index	Block Index	Block Count	EccAlgorithm	BBM	ECC DataLayout	DataUnitSize	MaxErrorBit(0-255)
IMG0	0	4	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG1	6	5	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG2	13	52	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG3	69	148	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG4	221	296	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG5	541	136	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG6	693	254	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG7	955	2	Not Use	Skip Bad Block	Disale	Disale	Disale
IMG8	958	3	Not Use	Skip Bad Block	Disale	Disale	Disale

BBM Layout:

Guarded Area Count: 0

Guarded Area Index	Start	End	Bad Block Allowed

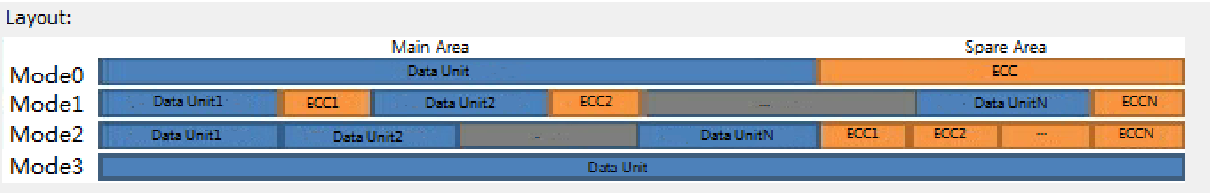
A. **BBM Configuration:**

Set up BBM according to the actual amounts of the loaded file. As the above image, there are multiple images files need EccAlgorithm, BBM, EccDataLayout, DataUnitSize, and MaxErrorBit.

- **EccAlgorithm:**
  - If "SpareArea UseFile" is checked on the "Load File-IMAGE" page, it will display:
    - Use File: indicating that the Image File has data containing SpareArea.
  - If “SpareArea UseFile” is not checked on the “Load File-IMAGE” page, it will display:
    - NotUse: Indicates that the Image File does not contain SpareArea data, and the ECC algorithm is not used.
    - BCH8 MDM: This ECC algorithm is only provided to Flex customers. Due to the confidentiality agreement, we cannot provide the detailed file of this algorithm.

- **BBM:** A choice of bad block management. Currently only supports Hard Copy, Skip Bad Block and No management. If you need other kinds of BBM, please feel free to contact us.

- **EccDataLayout:** This is the range definition for Bit Error analyzing. In the below graph, it provides four modes of data layouts.



For example: A page is 2048 + 64.
Mode 0: 2048 + 64
Mode 1: 512 + 16 + 512 + 16 + 512 + 16 + 512 + 16
Mode 2: 512 + 512 + 512 + 512 + 16 + 16 + 16 + 16
Mode 3: 2112

※ **Note: Verify would fail, if Bit Error occurs in the ECC section of Mode0~2 (The orange sections).**

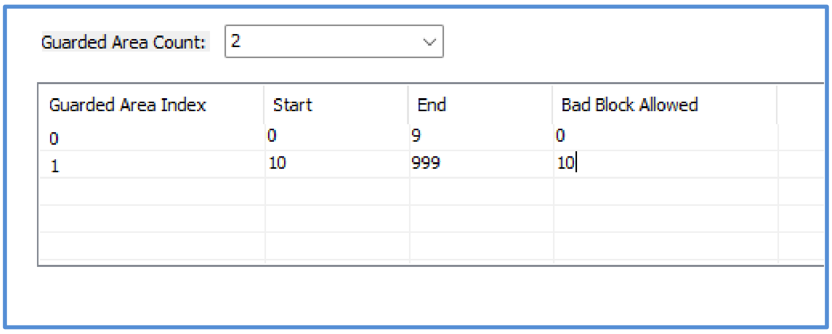
- **DataUnitSize:** Set up Data unit size according to different EccDataLayout modes. In this diagram, Mode0 will be 2048Bytes a unit.
- **MaxErrorBit:** Correspond to the volume of the DataUnit, the maximum acceptable Bit Error amounts of each DataUnit. In this diagram, 1 bit is acceptable for 2048Byte.

EccDataLayout, DataUnitSize, and MaxErrorBit are for verification, which will analyze the data accuracy and filter Bit Error to ensure the IC will work normally on board. For example, when ECC needs 528/2bit for CPU, then set EccDataLayout as Mode 3, DataUnitSize as 528Bytes, and MaxErrorBit as 2bit. If you want to filter the NAND Flash more strictly, set the MaxErrorBit to 1 bit. The default setting will depend on the Datasheet definition.

※ Note: If the ECC enable is selected in configuration setting, then IC will turn on the Internal ECC, which should not have Bit Error. Therefore, MaxErrorBit will be invalid.

B. Guarded Area Count:

This function can set up a range for block and detect the maximum acceptable value for bad blocks. In this chart, there two sets of ranges; the first condition does not allow any bad block from Block0 to Block9 while the second condition only allows ten bad blocks from Block 10 to Block 999. Then, during the process of creating BBT, the software will determine accordingly. If the above conditions are true, then the IC will be listed as NG parts.



Guarded Area Count: 2

Guarded Area Index	Start	End	Bad Block Allowed
0	0	9	0
1	10	999	10

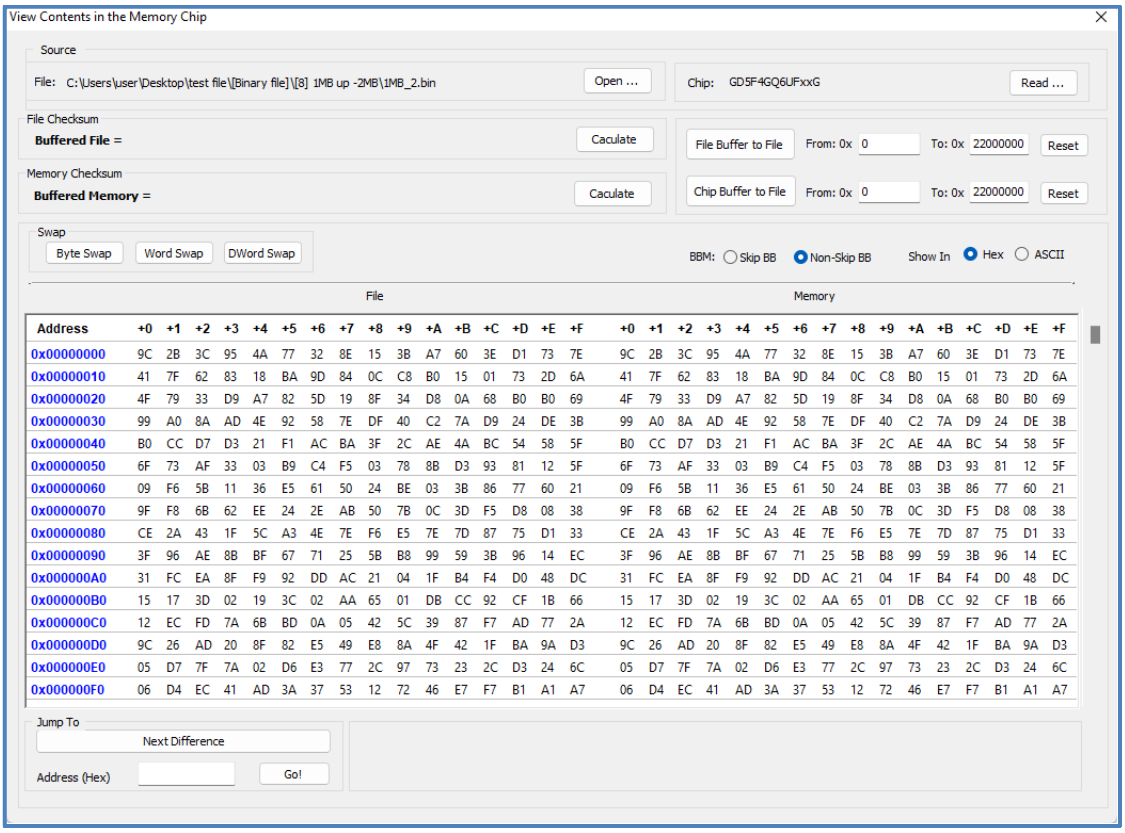
3.5 Edit Window Description

SPI Nand Flash content display:

In the edit window, file contents and chip contents can be displayed at the same time for comparison. By default, the selected file contents will be displayed as soon as you enter the edit window.

Click “Open” to show another file contents if needed. Also, the memory contents will auto- read and display the memory contents with one block size on the edit window. Click “Calculate” will get Checksum of the file and memory contents.

Click “Chip buffer to file” will save the chip contents which can set the range into a binary file; you can set up the file name and the location. Click “File buffer to file” will save the file buffer contents which can set the range into a binary file as well. In additional, File buffer can be modified in real time.



View Contents in the Memory Chip

Source

File: C:\Users\user\Desktop\test file\[Binary file]\[8] 1MB up -2MB\1MB_2.bin

Chip: GD5F4GQ6UFxxG

File Checksum

Buffered File =

From: 0x 0 To: 0x 22000000

Memory Checksum

Buffered Memory =

From: 0x 0 To: 0x 22000000

BBM: ○ Skip BB ● Non-Skip BB

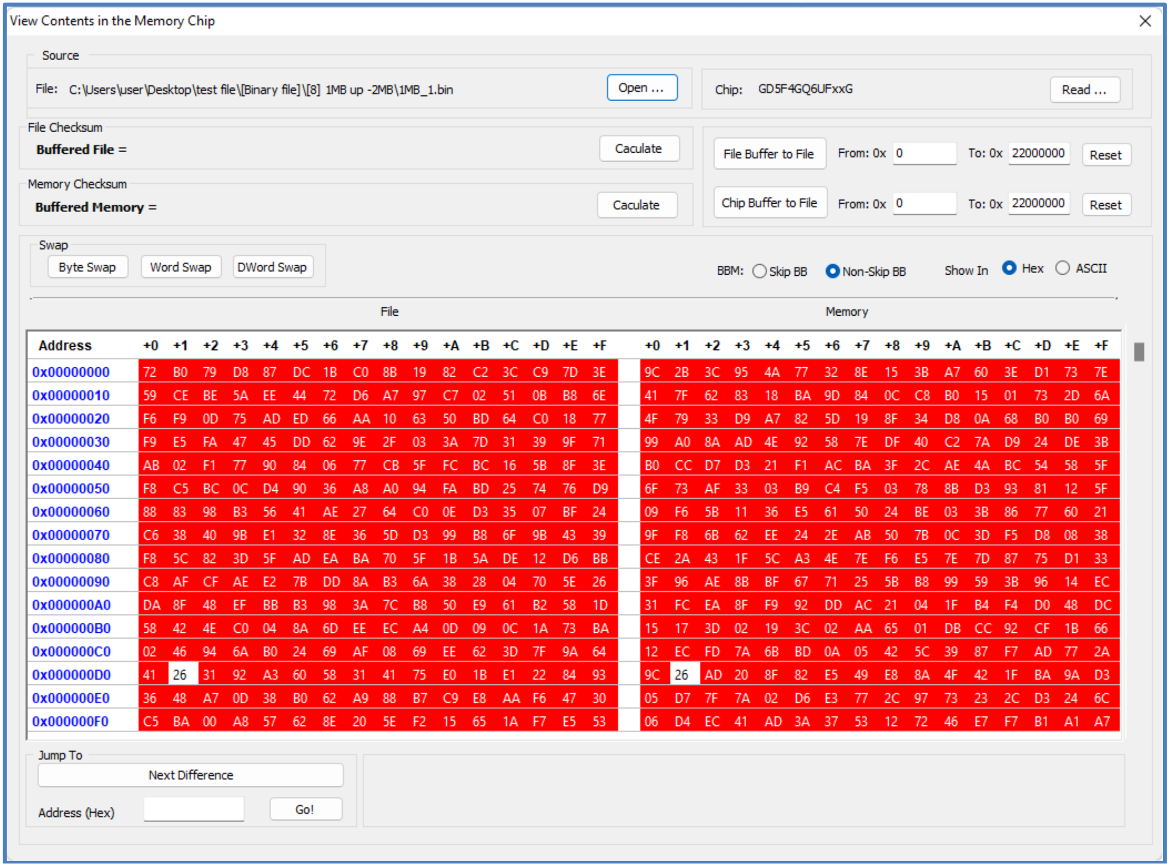
Show In ● Hex ○ ASCII

	File																Memory															
Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x00000000	9C	2B	3C	95	4A	77	32	8E	15	3B	A7	60	3E	D1	73	7E	9C	2B	3C	95	4A	77	32	8E	15	3B	A7	60	3E	D1	73	7E
0x00000010	41	7F	62	83	18	BA	9D	84	0C	C8	B0	15	01	73	2D	6A	41	7F	62	83	18	BA	9D	84	0C	C8	B0	15	01	73	2D	6A
0x00000020	4F	79	33	D9	A7	82	5D	19	8F	34	D8	0A	68	B0	B0	69	4F	79	33	D9	A7	82	5D	19	8F	34	D8	0A	68	B0	B0	69
0x00000030	99	A0	8A	AD	4E	92	58	7E	DF	40	C2	7A	D9	24	DE	3B	99	A0	8A	AD	4E	92	58	7E	DF	40	C2	7A	D9	24	DE	3B
0x00000040	B0	CC	D7	D3	21	F1	AC	BA	3F	2C	AE	4A	BC	54	58	5F	B0	CC	D7	D3	21	F1	AC	BA	3F	2C	AE	4A	BC	54	58	5F
0x00000050	6F	73	AF	33	03	B9	C4	F5	03	78	8B	D3	93	81	12	5F	6F	73	AF	33	03	B9	C4	F5	03	78	8B	D3	93	81	12	5F
0x00000060	09	F6	5B	11	36	E5	61	50	24	BE	03	3B	86	77	60	21	09	F6	5B	11	36	E5	61	50	24	BE	03	3B	86	77	60	21
0x00000070	9F	F8	6B	62	EE	24	2E	AB	50	7B	0C	3D	F5	D8	08	38	9F	F8	6B	62	EE	24	2E	AB	50	7B	0C	3D	F5	D8	08	38
0x00000080	CE	2A	43	1F	5C	A3	4E	7E	F6	E5	7E	7D	87	75	D1	33	CE	2A	43	1F	5C	A3	4E	7E	F6	E5	7E	7D	87	75	D1	33
0x00000090	3F	96	AE	8B	BF	67	71	25	5B	B8	99	59	3B	96	14	EC	3F	96	AE	8B	BF	67	71	25	5B	B8	99	59	3B	96	14	EC
0x000000A0	31	FC	EA	8F	F9	92	DD	AC	21	04	1F	B4	F4	D0	48	DC	31	FC	EA	8F	F9	92	DD	AC	21	04	1F	B4	F4	D0	48	DC
0x000000B0	15	17	3D	02	19	3C	02	AA	65	01	DB	CC	92	CF	1B	66	15	17	3D	02	19	3C	02	AA	65	01	DB	CC	92	CF	1B	66
0x000000C0	12	EC	FD	7A	6B	BD	0A	05	42	5C	39	87	F7	AD	77	2A	12	EC	FD	7A	6B	BD	0A	05	42	5C	39	87	F7	AD	77	2A
0x000000D0	9C	26	AD	20	8F	82	E5	49	E8	8A	4F	42	1F	BA	9A	D3	9C	26	AD	20	8F	82	E5	49	E8	8A	4F	42	1F	BA	9A	D3
0x000000E0	05	D7	7F	7A	02	D6	E3	77	2C	97	73	23	2C	D3	24	6C	05	D7	7F	7A	02	D6	E3	77	2C	97	73	23	2C	D3	24	6C
0x000000F0	06	D4	EC	41	AD	3A	37	53	12	72	46	E7	F7	B1	A1	A7	06	D4	EC	41	AD	3A	37	53	12	72	46	E7	F7	B1	A1	A7

Jump To

Address (Hex)

If the file contents and chip contents are different, then those will be highlighted with the "Red Fonts". Click "next difference" button will go to the next different content or fill the address in Address (Hex), and then click "Go" to go to the assigned address.



View Contents in the Memory Chip

Source

File: C:\Users\user\Desktop\test file\[Binary file]\[8] 1MB up -2MB\1MB_1.bin

Chip: GD5F4GQ6UFxxG

File Checksum

Buffered File =

From: 0x 0 To: 0x 22000000

Memory Checksum

Buffered Memory =

From: 0x 0 To: 0x 22000000

BBM: ○ Skip BB ● Non-Skip BB

Show In ● Hex ○ ASCII

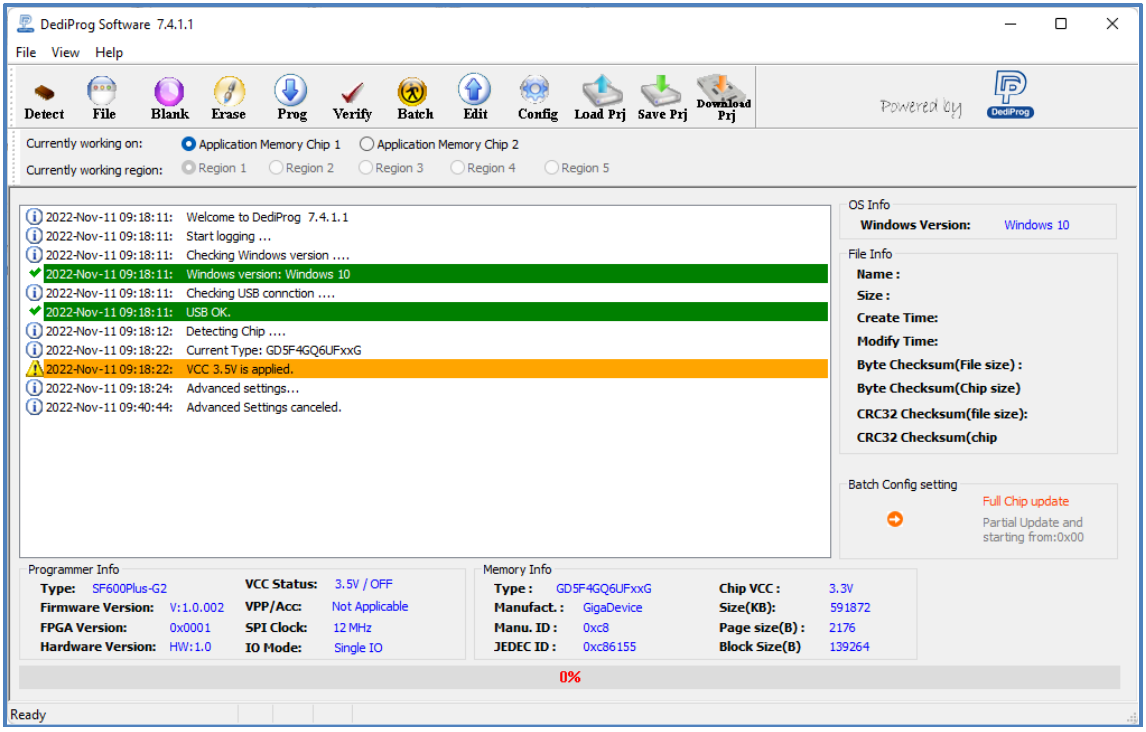
	File																Memory															
Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x00000000	72	B0	79	D8	87	DC	1B	C0	8B	19	82	C2	3C	C9	7D	3E	9C	2B	3C	95	4A	77	32	8E	15	3B	A7	60	3E	D1	73	7E
0x00000010	59	CE	BE	5A	EE	44	72	D6	A7	97	C7	02	51	0B	B8	6E	41	7F	62	83	18	BA	9D	84	0C	C8	B0	15	01	73	2D	6A
0x00000020	F6	F9	0D	75	AD	ED	66	AA	10	63	50	BD	64	C0	18	77	4F	79	33	D9	A7	82	5D	19	8F	34	D8	0A	68	B0	B0	69
0x00000030	F9	E5	FA	47	45	DD	62	9E	2F	03	3A	7D	31	39	9F	71	99	A0	8A	AD	4E	92	58	7E	DF	40	C2	7A	D9	24	DE	3B
0x00000040	AB	02	F1	77	90	84	06	77	CB	5F	FC	BC	16	5B	8F	3E	B0	CC	D7	D3	21	F1	AC	BA	3F	2C	AE	4A	BC	54	58	5F
0x00000050	F8	C5	BC	0C	D4	90	36	A8	A0	94	FA	BD	25	74	76	D9	6F	73	AF	33	03	B9	C4	F5	03	78	8B	D3	93	81	12	5F
0x00000060	88	83	98	B3	56	41	AE	27	64	C0	0E	D3	35	07	BF	24	09	F6	5B	11	36	E5	61	50	24	BE	03	3B	86	77	60	21
0x00000070	C6	38	40	9B	E1	32	8E	36	5D	D3	99	B8	6F	9B	43	39	9F	F8	6B	62	EE	24	2E	AB	50	7B	0C	3D	F5	D8	08	38
0x00000080	F8	5C	82	3D	5F	AD	EA	BA	70	5F	1B	5A	DE	12	D6	BB	CE	2A	43	1F	5C	A3	4E	7E	F6	E5	7E	7D	87	75	D1	33
0x00000090	C8	AF	CF	AE	E2	7B	DD	8A	B3	6A	38	28	04	70	5E	26	3F	96	AE	8B	BF	67	71	25	5B	B8	99	59	3B	96	14	EC
0x000000A0	DA	8F	48	EF	BB	B3	98	3A	7C	B8	50	E9	61	B2	58	1D	31	FC	EA	8F	F9	92	DD	AC	21	04	1F	B4	F4	D0	48	DC
0x000000B0	58	42	4E	C0	04	8A	6D	EE	EC	A4	0D	09	0C	1A	73	BA	15	17	3D	02	19	3C	02	AA	65	01	DB	CC	92	CF	1B	66
0x000000C0	02	46	94	6A	B0	24	69	AF	08	69	EE	62	3D	7F	9A	64	12	EC	FD	7A	6B	BD	0A	05	42	5C	39	87	F7	AD	77	2A
0x000000D0	41	26	31	92	A3	60	58	31	41	75	E0	1B	E1	22	84	93	9C	26	AD	20	8F	82	E5	49	E8	8A	4F	42	1F	BA	9A	D3
0x000000E0	36	48	A7	0D	38	B0	62	A9	88	B7	C9	E8	AA	F6	47	30	05	D7	7F	7A	02	D6	E3	77	2C	97	73	23	2C	D3	24	6C
0x000000F0	C5	BA	00	A8	57	62	8E	20	5E	F2	15	65	1A	F7	E5	53	06	D4	EC	41	AD	3A	37	53	12	72	46	E7	F7	B1	A1	A7

Jump To

Address (Hex)

3.6 Configuration Window Description

This feature allows users to configure advanced settings.



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2

Currently working region: ○ Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

```
2022-Nov-11 09:18:11:  Welcome to DediProg 7.4.1.1
2022-Nov-11 09:18:11:  Start logging ...
2022-Nov-11 09:18:11:  Checking Windows version ....
2022-Nov-11 09:18:11:  Windows version: Windows 10
2022-Nov-11 09:18:11:  Checking USB connction ....
2022-Nov-11 09:18:11:  USB OK.
2022-Nov-11 09:18:12:  Detecting Chip ....
2022-Nov-11 09:18:22:  Current Type: GD5F4GQ6UFxxG
2022-Nov-11 09:18:22:  VCC 3.5V is applied.
2022-Nov-11 09:18:24:  Advanced settings...
2022-Nov-11 09:40:44:  Advanced Settings canceled.
```

OS Info
Windows Version: Windows 10

File Info
Name :
Size :
Create Time:
Modify Time:
Byte Checksum(File size) :
Byte Checksum(Chip size)
CRC32 Checksum(file size):
CRC32 Checksum(chip

Batch Config setting
Full Chip update
Partial Update and starting from:0x00

Programmer Info
Type: SF600Plus-G2
Firmware Version: V:1.0.002
FPGA Version: 0x0001
Hardware Version: HW:1.0
VCC Status: 3.5V / OFF
VPP/Acc: Not Applicable
SPI Clock: 12 MHz
IO Mode: Single IO

Memory Info
Type : GD5F4GQ6UFxxG
Manufact. : GigaDevice
Manu. ID : 0xc8
JEDEC ID : 0xc86155
Chip VCC : 3.3V
Size(KB): 591872
Page size(B) : 2176
Block Size(B) 139264

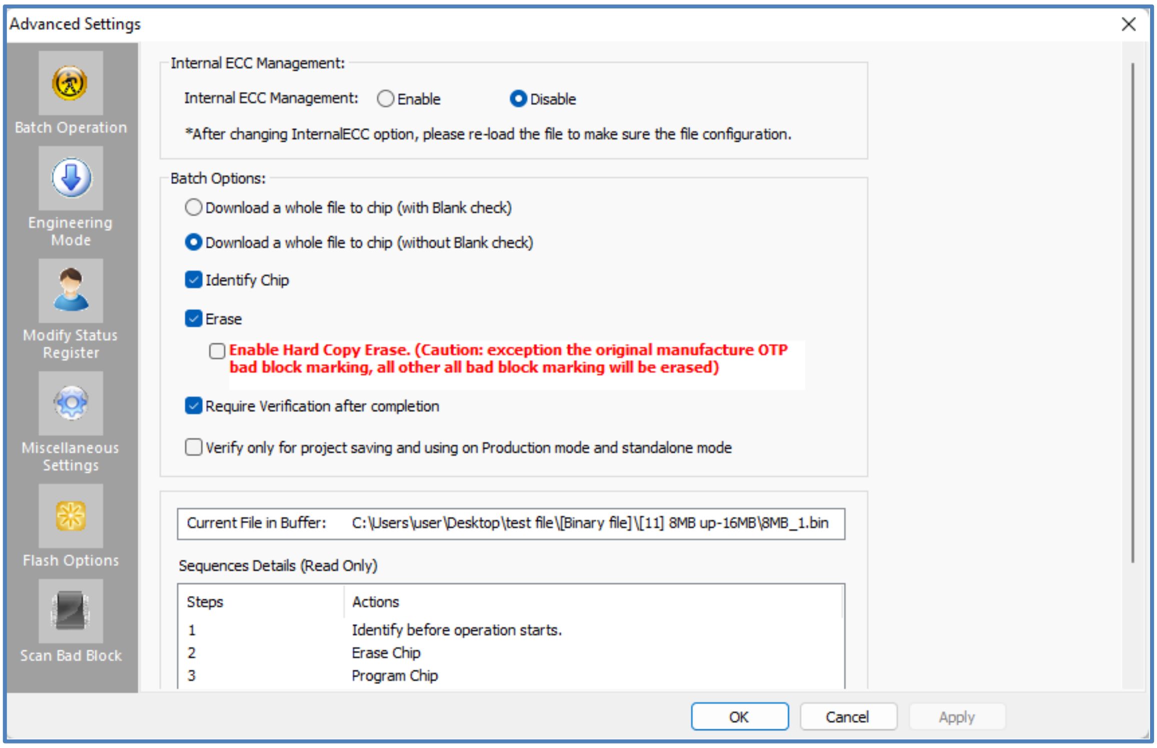
0%

Ready

3.6.1 Batch Operation Option

Config:

All the settings for SPI NAND are in the Config setting. Below section will describe more about each function.



Internal ECC Management:

Internal ECC Management: ○ Enable ● Disable

*After changing InternalECC option, please re-load the file to make sure the file configuration.

Batch Options:

○ Download a whole file to chip (with Blank check)

● Download a whole file to chip (without Blank check)

☑ Identify Chip

☑ Erase

☐ Enable Hard Copy Erase. (Caution: exception the original manufacture OTP bad block marking, all other all bad block marking will be erased)

☑ Require Verification after completion

☐ Verify only for project saving and using on Production mode and standalone mode

Current File in Buffer: C:\Users\user\Desktop\test file\[Binary file]\[11] 8MB up-16MB\8MB_1.bin

Sequences Details (Read Only)

Steps	Actions
1	Identify before operation starts.
2	Erase Chip
3	Program Chip

Bad Block Management:
It is the action that it will take if bad block appears during IC programming. The default setting is Skip, otherwise, choose Hard copy.

Internal ECC management:
The Internal ECC of SPI NAND can be turn on or off. When Internal ECC is needed during application, then you can choose Enable; on the other hand, if the file has its own ECC Code, then choose Disable.

Batch Options:
This is the working procedure for the programmer when Auto batch.
Download a whole file to chip (with Blank check): During Auto Batch, after execute the Erase action, it needs to execute the blank check action.

Download a whole file to chip (without Blank check): During Auto Batch, after execute the Erase action, it **does not** need to execute the blank check action.

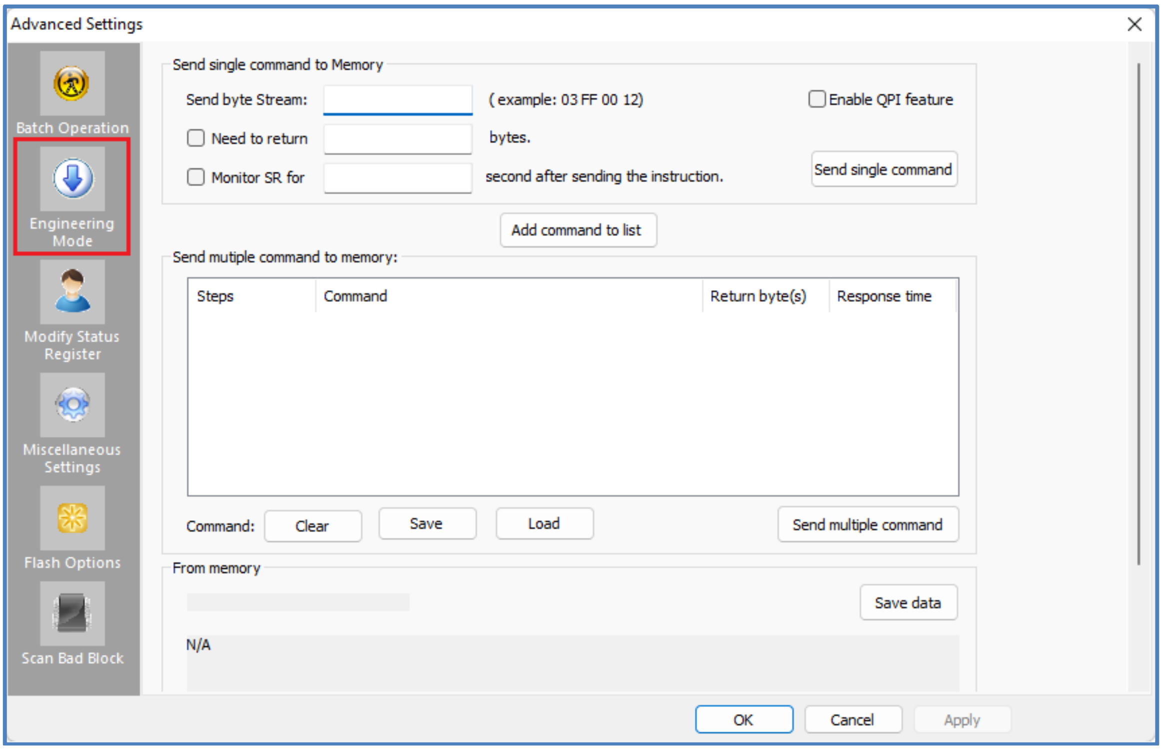
Identify Chip:
You can decide whether you want to do ID check or not.

Erase:
This is to set up whether you want to erase the SPI NAND during Auto Batch. Default will only erase the Good blocks. If you want to erase forcibly (Force Erase), you can select the “Enable Hard Copy Erase” However, please think about it before you enable the hard copy erase.

Require Verification after completion:
This is for verification after programming.

Verify only for project saving and using on Production mode and standalone mode:
If you only want to verify the Flash without any programming action, then you can select this option. This option is only for the standalone or the production mode.

3.6.2 Engineering Mode



This function allows you to define your own SPI command and send it directly to the target SPI flash. This option allows you to add other SPI commands even if it was not originally added on the programmer.

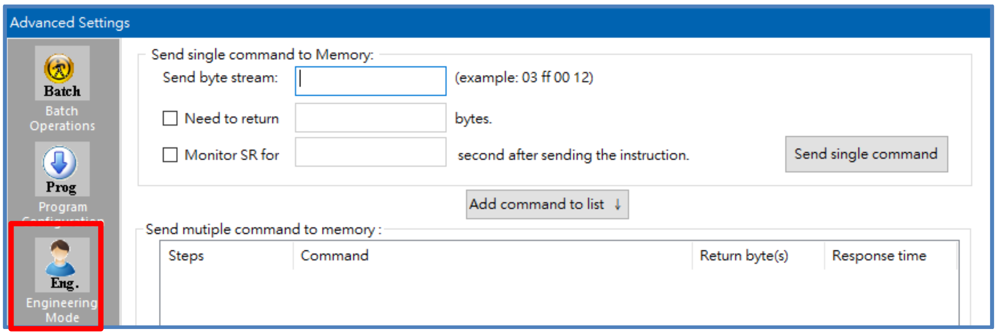
The engineering mode can be used for sending instruction to the SPI Nand Flash.

3.6.2.1 Send single command to Memory

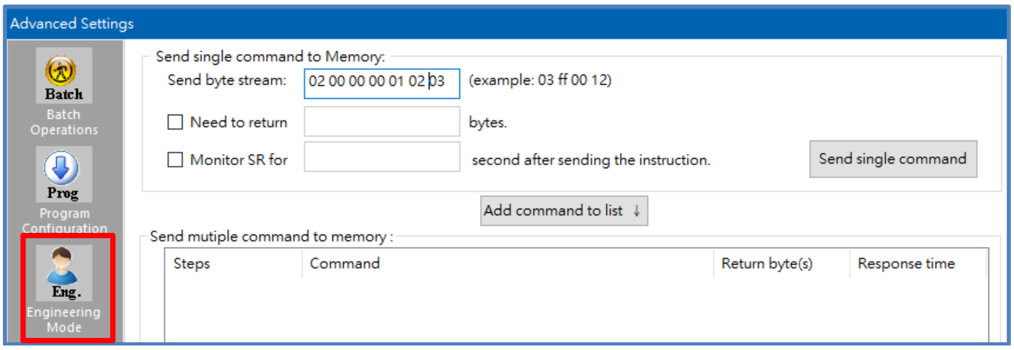
You can define the data bytes to be sent from the programmer to the SPI Flash and the number of bytes to be returned. Also define if the status register WIP bit must be polled to check if the SPI Flash is busy or ready. Send single command by clicking “Send single command” button.

For example:
Write “01 02 03” data bytes at the address “00 00 00” and verify.

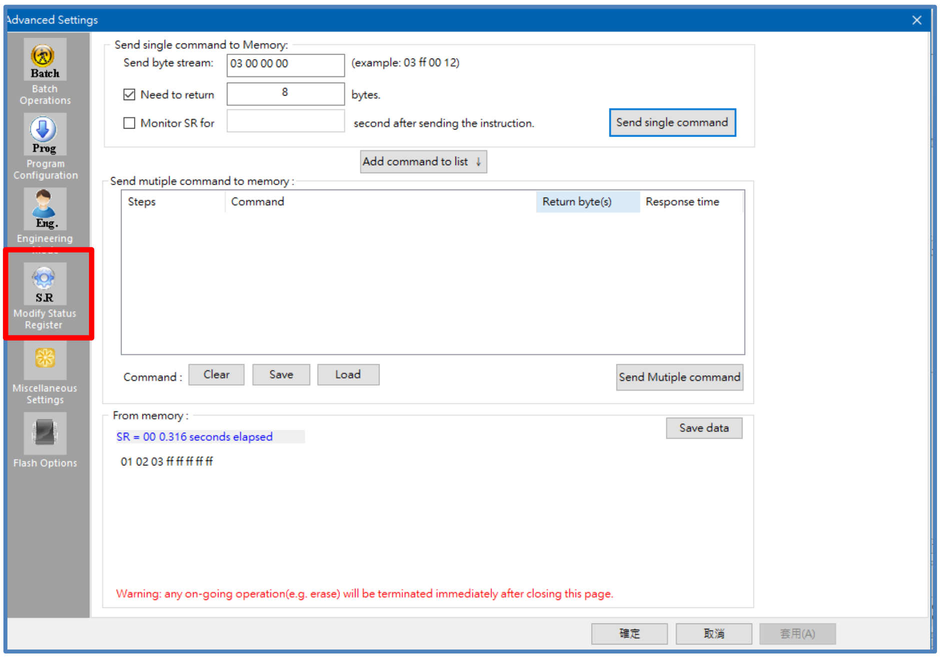
First: Programmer needs to set the WEL bit by sending the WREN (06h) command to the SPI Flash as described below:



Second: Programmer needs to send the programming instruction “02h” followed by the address “00 00 00” and the data “01 02 03” while monitor the Status register WIP bit as described below:


Send byte stream: 02 00 00 00 01 02 03

Third: The programmer needs to verify the SPI Flash content by sending the Read instruction “03h” and the address “00 00 00”, then read the return bytes from the SPI Flash (we read 8 bytes in the following example):


Send byte stream: 03 00 00 00

☑ Need to return 8 bytes.

From memory :

SR = 00 0.316 seconds elapsed

01 02 03 ff ff ff ff ff

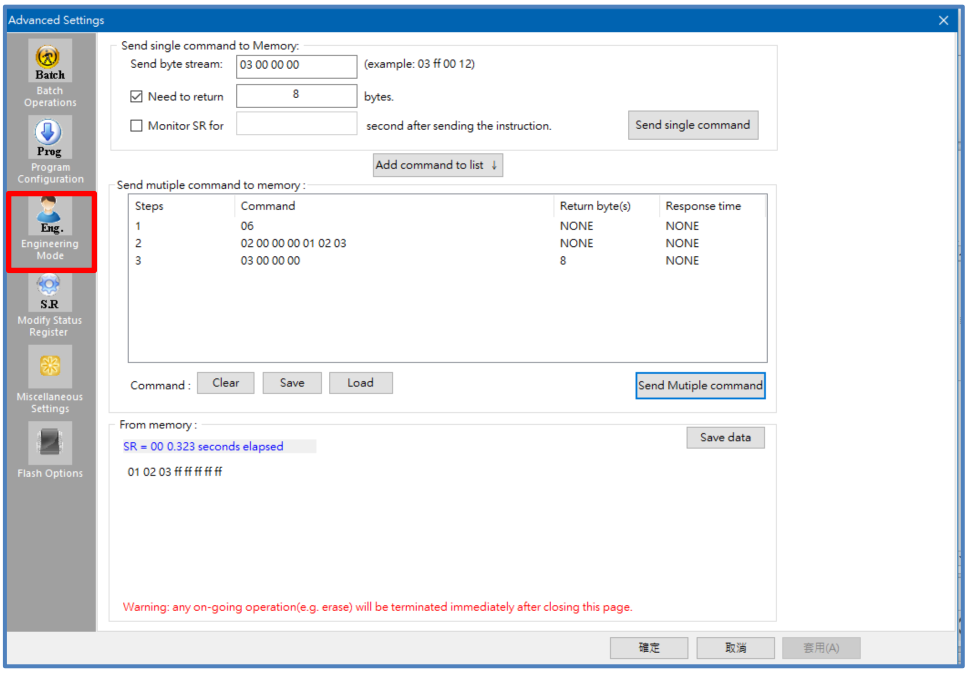
Warning: any on-going operation(e.g. erase) will be terminated immediately after closing this page.

The return bytes from the SPI Flash are displayed in the “from memory” window.

3.6.2.2 Send multiple commands

In order to save time from doing repetitive commands, DediProg provides multiple command sending function, so you can save or load command to.ini file. In order to add command to the command list, click “Add command to list” button and click "Send Multiple command" to send command by priority.

※ **NOTE: Delete the command by double clicks the number of the steps item.**


Advanced Settings

Send single command to Memory:
Send byte stream: 03 00 00 00 (example: 03 ff 00 12)
☑ Need to return 8 bytes.
☐ Monitor SR for second after sending the instruction.

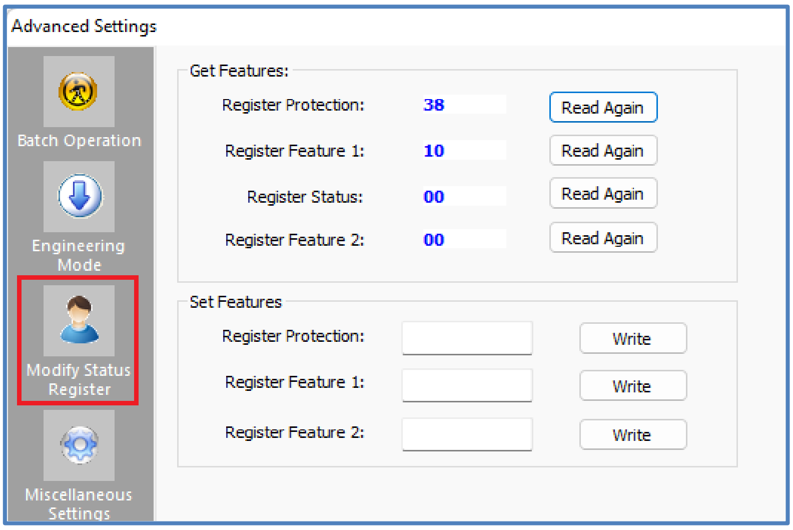
Send mutiple command to memory :

Steps	Command	Return byte(s)	Response time
1	06	NONE	NONE
2	02 00 00 00 01 02 03	NONE	NONE
3	03 00 00 00	8	NONE

From memory :
SR = 00 0.323 seconds elapsed
01 02 03 ff ff ff ff ff

Warning: any on-going operation(e.g. erase) will be terminated immediately after closing this page.

3.6.3 Modify Status register

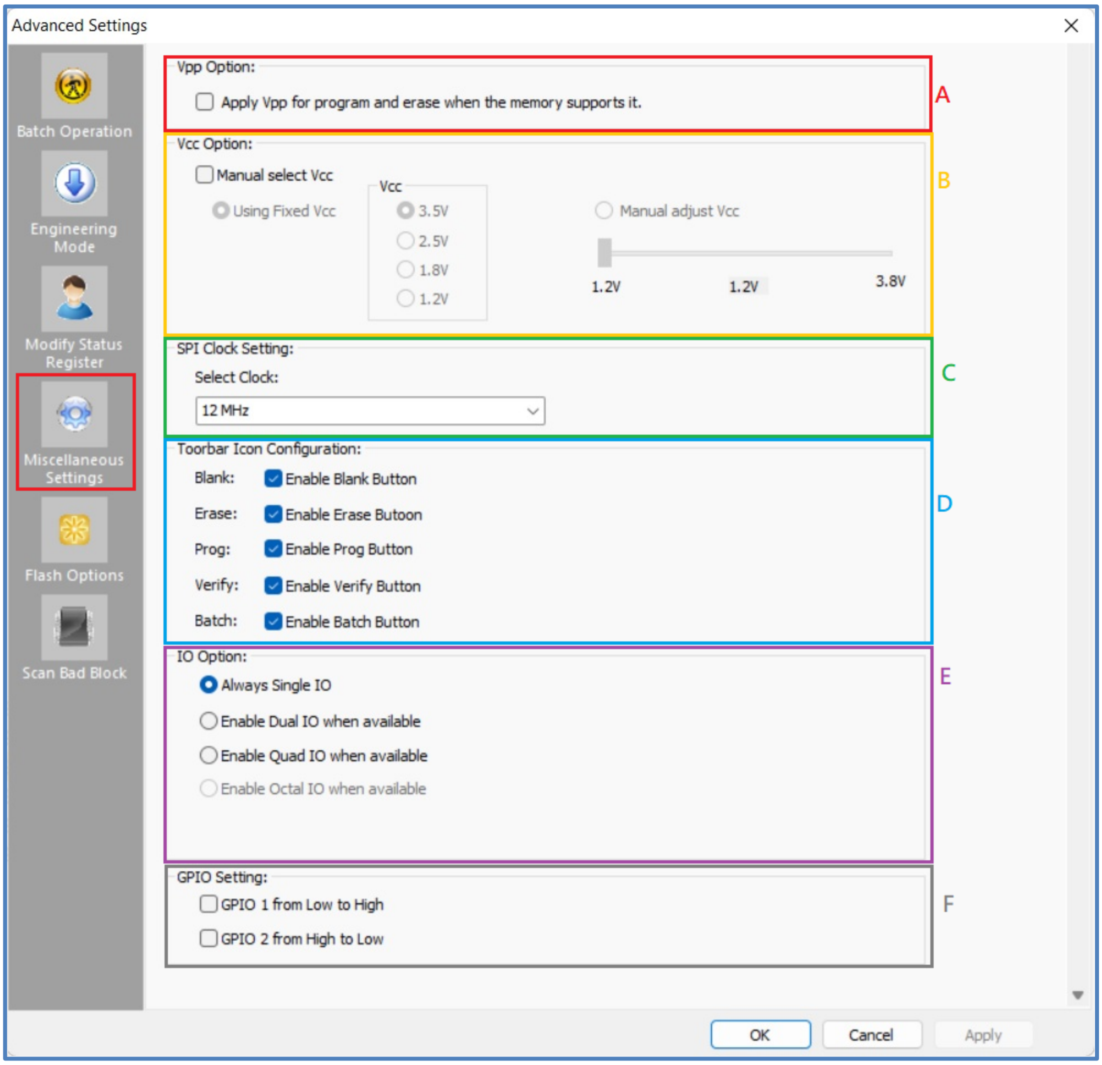

Advanced Settings

Get Features:
Register Protection: 38
Register Feature 1: 10
Register Status: 00
Register Feature 2: 00

Set Features
Register Protection:
Register Feature 1:
Register Feature 2:

This function allows you to modify or read the Features value of the target serial NAND flash. When writing data to the serial NAND flash, the data will read immediately.

3.6.4 Miscellaneous Settings



Vpp Option:
☐ Apply Vpp for program and erase when the memory supports it.

Vcc Option:
☐ Manual select Vcc
○ Using Fixed Vcc
Vcc
● 3.5V
○ 2.5V
○ 1.8V
○ 1.2V
○ Manual adjust Vcc
1.2V 1.2V 3.8V

SPI Clock Setting:
Select Clock:
12 MHz

Toorbar Icon Configuration:
Blank: ☑ Enable Blank Button
Erase: ☑ Enable Erase Butoon
Prog: ☑ Enable Prog Button
Verify: ☑ Enable Verify Button
Batch: ☑ Enable Batch Button

IO Option:
● Always Single IO
○ Enable Dual IO when available
○ Enable Quad IO when available
○ Enable Octal IO when available

GPIO Setting:
☐ GPIO 1 from Low to High
☐ GPIO 2 from High to Low

A. VPP Option

This setting enables the VPP option so the High voltage is applied on the SPI Flash Wp pin to reduce the programming and the erasing time.

This option can only be enabled on Serial Flash supporting the VPP feature.

B. VCC Option

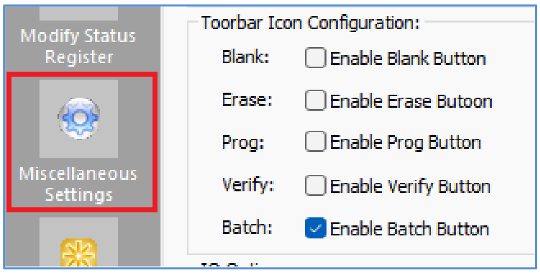
SF series programmers support 3.5V, 2.5V, 1.8V, and 1.2 VCC. The default VCC status will be 3.5V when plug in the programmer without IC on it. You will be able to modify the VCC configuration, and then the VCC setting will be changed and saved.

C. SPI Clock Setting

The SPI clock frequency can be adjusted by user to fit the application requirements or SPI Flash performance. Notice that the SPI Flash frequency is defined from the supplier specification for a maximum capacitance usually is 30pf or 15pF. The application is therefore designed not to exceed this maximum capacitance. In-circuit programming does not fulfill anymore this original design as additional capacitance will be added according to the cable length and programmer. Therefore, you cannot expect to program on board SPI flash to the maximum frequency of the datasheet since the SPI flash will not be able to drive such capacitance at such high frequency. In order to comply with the different capacitance and SPI flash driving capability, DediProg provides frequency adjustment of the programmer. Frequency needs to be reduced if the data timings do not comply with the specification.

D. Tool Bar ICON Configuration

You can hide the tool bar icons by uncheck the icon items in the “Toolbar Icon configuration setting”. For example, if you only want the batch icon, you can leave only batch button selected and save the setting, then only the batch icon will appear on the tool bar.



Toorbar Icon Configuration:

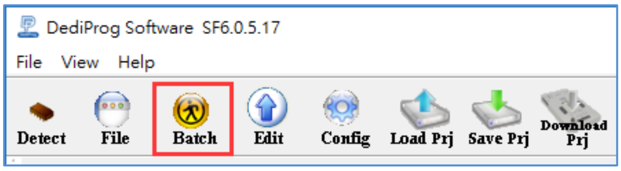
Blank: □ Enable Blank Button

Erase: □ Enable Erase Butoon

Prog: □ Enable Prog Button

Verify: □ Enable Verify Button

Batch: ☑ Enable Batch Button



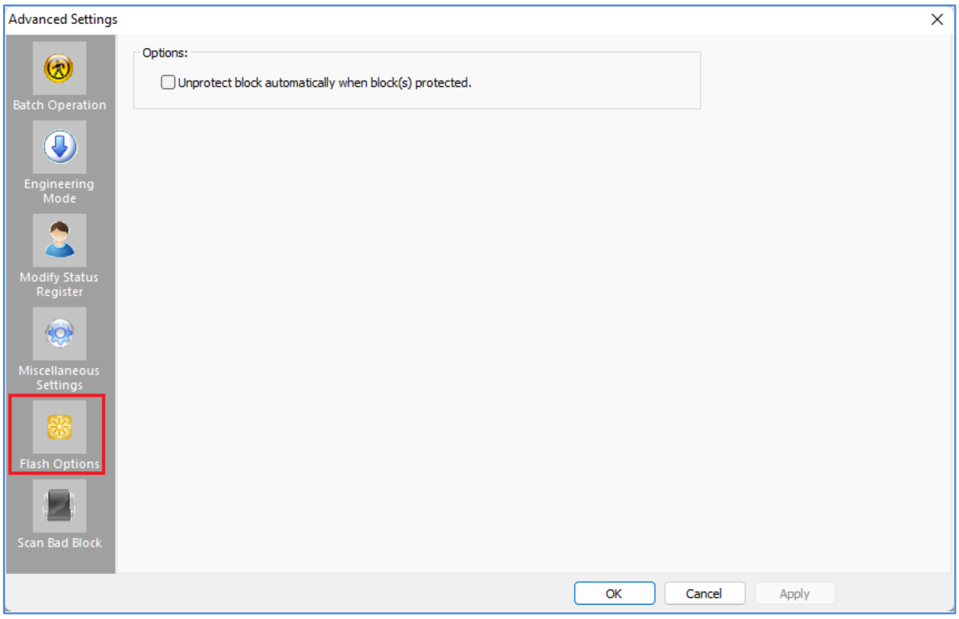
E. IO Option:

When the selected chip and the software support multiple IO command, it can change the IO Mode when programming. However, the options will be disabled and displayed in grayscale if the software does not support multi-IO programming for the selected chip.

F. GPIO Setting

There two GPIOs, GPIO1 and GPIO2, for using. You can select the behavior you need. When the option is selected, the IO will keep the status until un-plug the programmer. For example, when GPIO1 is selected, the GPIO1 will keep high until un-plug the programmer.

3.6.5 Flash Option

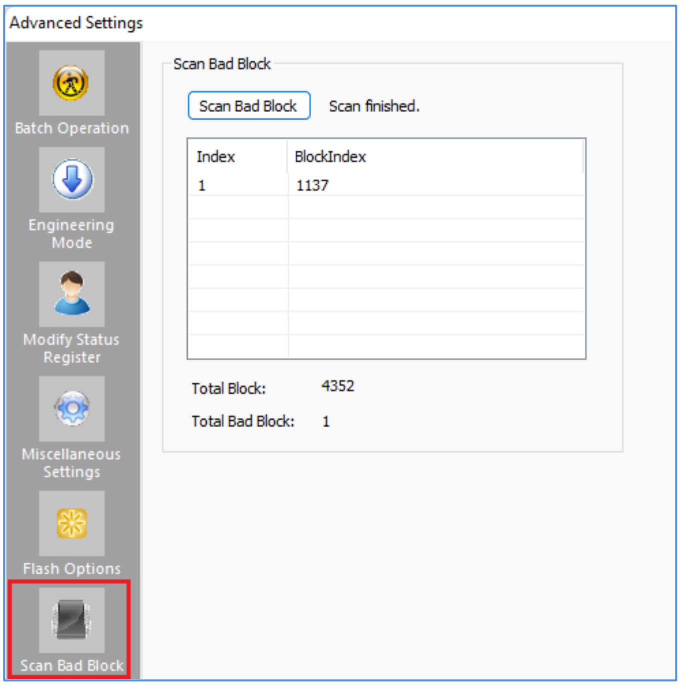


Options:

□ Unprotect block automatically when block(s) protected.

Unprotect block automatically when block(s) protected.

3.6.6 Scan Bad Block



Scan Bad Block

Scan finished.

Index	BlockIndex
1	1137

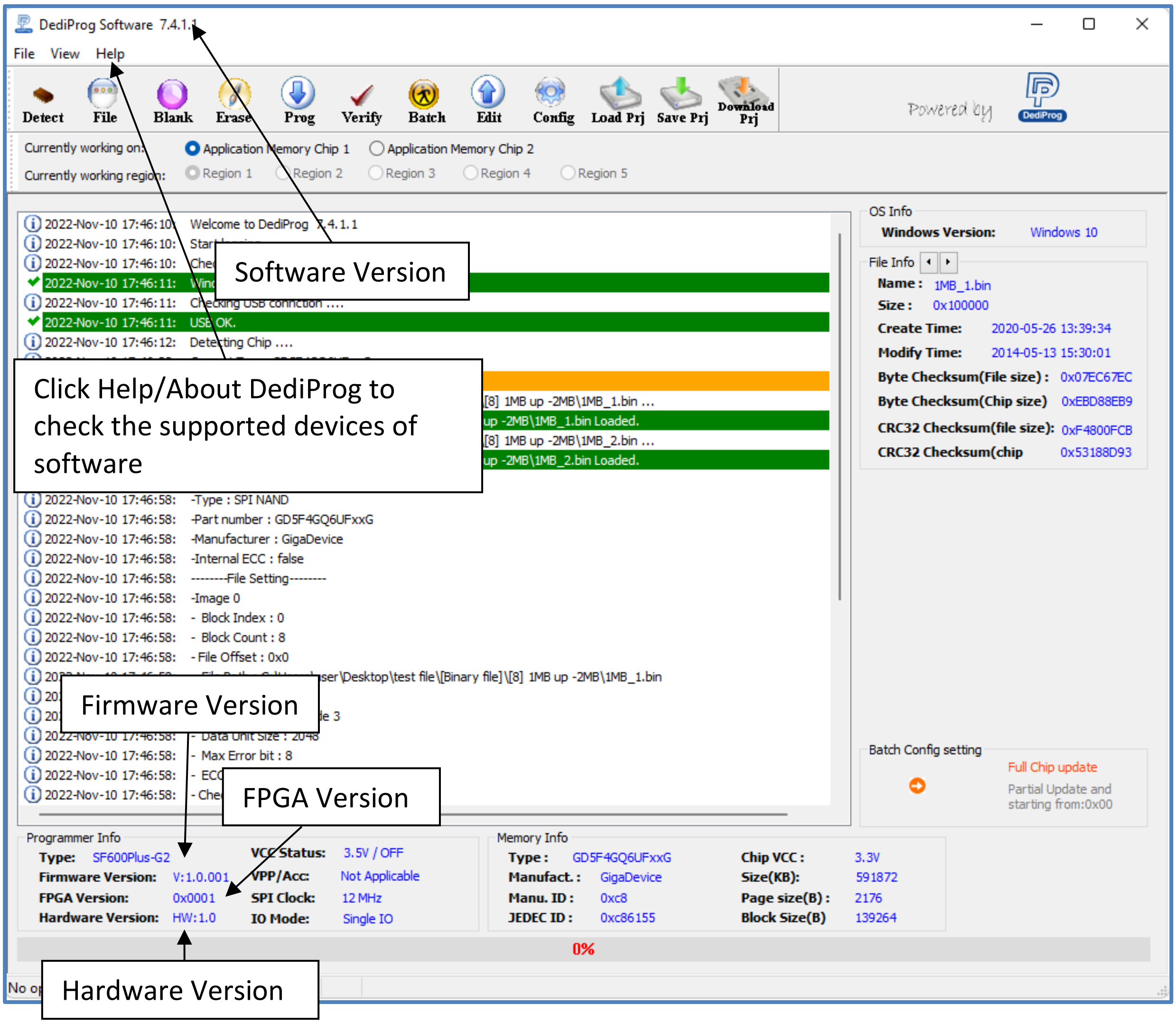
Total Block: 4352

Total Bad Block: 1

Click Scan Bad Block button, software will scan the SPI NAND flash bad block(s), which will be very beneficial to NAND analysis.

3.7 Supported Devices, Software Version, Firmware Version

You can check the Serial flash support list on our website. The list is valid for the latest software and firmware, so check the current version that you are using and update it if necessary.



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2

Currently working region: ○ Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

```
2022-Nov-10 17:46:10:  Welcome to DediProg 7.4.1.1
2022-Nov-10 17:46:10:  Star
2022-Nov-10 17:46:10:  Che
2022-Nov-10 17:46:11:  Winc
2022-Nov-10 17:46:11:  Checking USB connection ....
2022-Nov-10 17:46:11:  USB OK.
2022-Nov-10 17:46:12:  Detecting Chip ....
                       [8] 1MB up -2MB\1MB_1.bin ...
                       up -2MB\1MB_1.bin Loaded.
                       [8] 1MB up -2MB\1MB_2.bin ...
                       up -2MB\1MB_2.bin Loaded.
2022-Nov-10 17:46:58:  -Type : SPI NAND
2022-Nov-10 17:46:58:  -Part number : GD5F4GQ6UFxxG
2022-Nov-10 17:46:58:  -Manufacturer : GigaDevice
2022-Nov-10 17:46:58:  -Internal ECC : false
2022-Nov-10 17:46:58:  --------File Setting--------
2022-Nov-10 17:46:58:  -Image 0
2022-Nov-10 17:46:58:  - Block Index : 0
2022-Nov-10 17:46:58:  - Block Count : 8
2022-Nov-10 17:46:58:  - File Offset : 0x0
20                     ser\Desktop\test file\[Binary file]\[8] 1MB up -2MB\1MB_1.bin
20
20                     de 3
2022-Nov-10 17:46:58:  - Data Unit Size : 2048
2022-Nov-10 17:46:58:  - Max Error bit : 8
2022-Nov-10 17:46:58:  - ECC
2022-Nov-10 17:46:58:  - Che
```

OS Info
Windows Version: Windows 10

File Info
Name : 1MB_1.bin
Size : 0x100000
Create Time: 2020-05-26 13:39:34
Modify Time: 2014-05-13 15:30:01
Byte Checksum(File size) : 0x07EC67EC
Byte Checksum(Chip size) 0xEBD88EB9
CRC32 Checksum(file size): 0xF4800FCB
CRC32 Checksum(chip 0x53188D93

Batch Config setting
Full Chip update
Partial Update and starting from:0x00

Programmer Info
Type: SF600Plus-G2
Firmware Version: V:1.0.001
FPGA Version: 0x0001
Hardware Version: HW:1.0
VCC Status: 3.5V / OFF
VPP/Acc: Not Applicable
SPI Clock: 12 MHz
IO Mode: Single IO

Memory Info
Type : GD5F4GQ6UFxxG
Manufact. : GigaDevice
Manu. ID : 0xc8
JEDEC ID : 0xc86155
Chip VCC : 3.3V
Size(KB): 591872
Page size(B) : 2176
Block Size(B) 139264

0%

No op

Software Version

Click Help/About DediProg to check the supported devices of software

Firmware Version

FPGA Version

Hardware Version

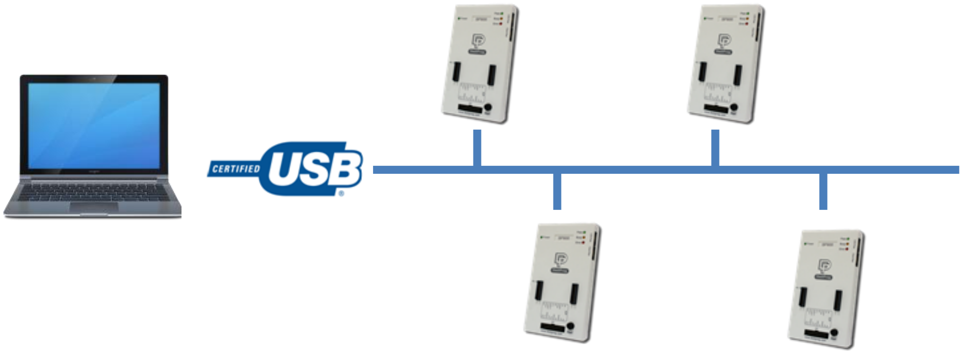
IV. DediProg SF Software Production GUI

DediProg SF software production GUI is only available after the software version 5.x.x. The production GUI allows you to plug in and operate multiple SF100/SF600/SF600Plus/SF700 at the same time.

The new software will remove the old USB driver when it detects such driver during installation. New USB driver is required in order to run the software and the driver will come together with the software CD ROM or it can be downloaded from DediProg website. www.dediprog.com/download

In order to run more than one SF programmer at the same time reliably, USB hub with individual power supply is highly recommended.

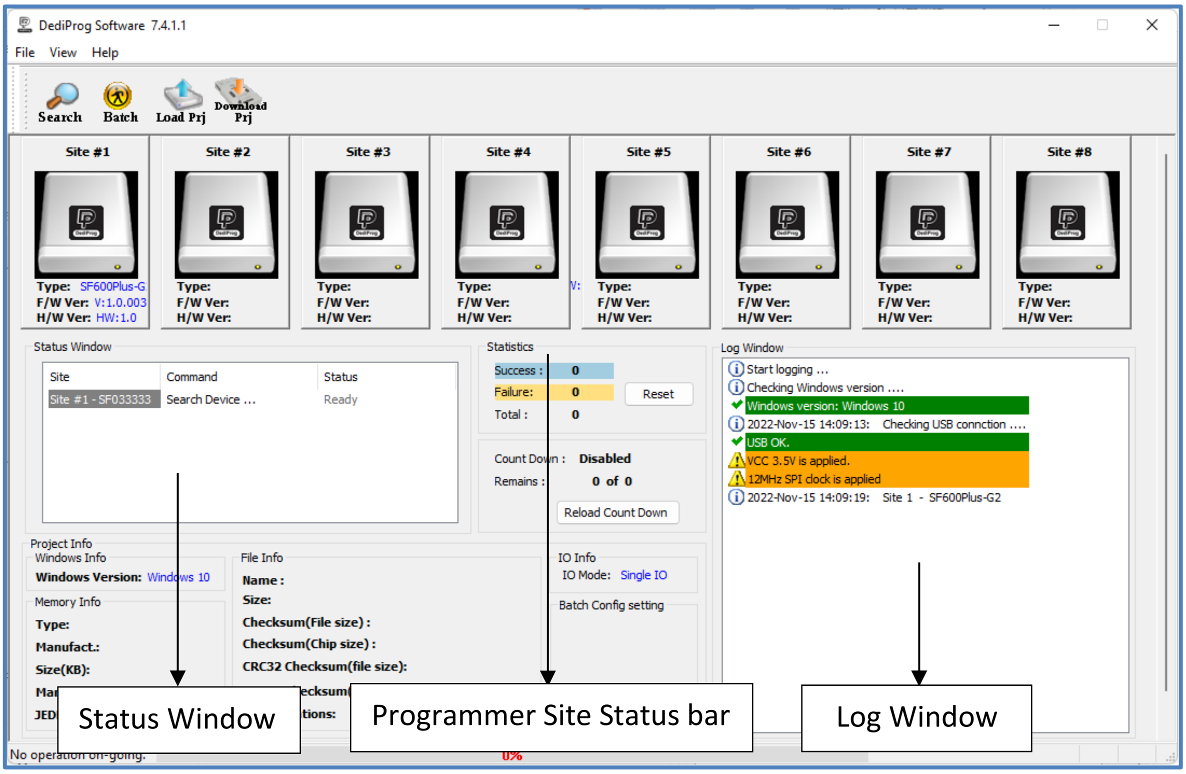
Multi-Programmers Capability for SF series programmers



In order to run production GUI, please plug in all USB of the intended programmers prior opening the software. It is not recommended to add (plug in) or remove (unplug) the programmers when the software is running.

The production software does not provide auto chip detect feature, therefore use “programmer search” and “load project” prior the operations.

The production GUI manual will only illustrate the items that not covered in the engineering GUI. Therefore, function descriptions such as Program, Erase, and Blank check will not be repeated here.



Site #1 – Type: SF600Plus-G
F/W Ver: V:1.0.003
H/W Ver: HW:1.0

Sites #2 to #8 – Type:, F/W Ver:, H/W Ver:

Status Window

Site	Command	Status
Site #1 - SF033333	Search Device ...	Ready

Statistics

Success : 0
Failure: 0
Total : 0

Count Down : Disabled
Remains : 0 of 0

Log Window

```
Start logging ...
Checking Windows version ....
Windows version: Windows 10
2022-Nov-15 14:09:13:   Checking USB connction ....
USB OK.
VCC 3.5V is applied.
12MHz SPI clock is applied
2022-Nov-15 14:09:19:   Site 1  -  SF600Plus-G2
```

Project Info

Windows Info
Windows Version: Windows 10

Memory Info
Type:
Manufact.:
Size(KB):

File Info
Name :
Size:
Checksum(File size) :
Checksum(Chip size) :
CRC32 Checksum(file size):

IO Info
IO Mode: Single IO

Batch Config setting

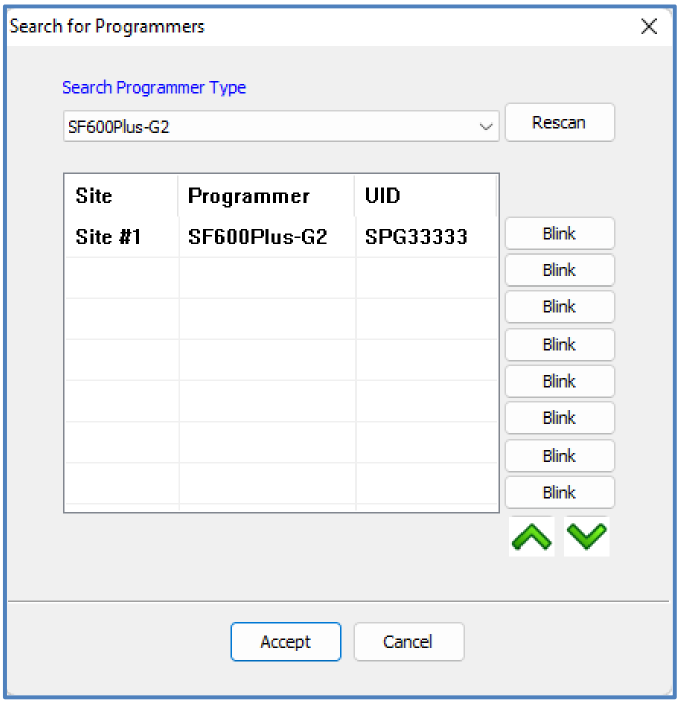
No operation on-going.
0%

4.1 Search

Click “search”, the software will show programmer type. The default programmer type is SF100. Please select the programmer you are using and click Rescan.

Search Programmer:

The detected programmers will be listed along with the site number. The site number is given by the Window OS randomly; you can use the “blink”, “up” and “down” button to adjust the real sequence of the connected programmer. When click on “blink”, the connected programmer will blink on its green LED once. You can use this feature to locate the programmer associated with its site number. For programmers with firmware version after 5.x.x, DediProg will write a serial number in the hardware before shipping out and the serial number will be displayed in the following screen snapshot.



Search Programmer Type: SF600Plus-G2

Site	Programmer	UID
Site #1	SF600Plus-G2	SPG33333

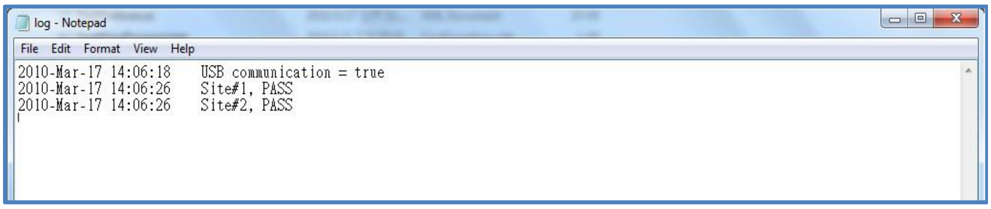
※ Note: SF software doesn’t support different programmer at the same time, and only supports same programmer on the production mode.

V. DediProg Windows Command Line

5.1 Introduction

The window command line has been designed to control DediProg programmer from the other software. This feature will be convenient to synchronize the two software in development (For example: program the memory automatically after the code has been compiled) or in production (for example: Program automatically the Serial Flash via the ICT tester after the hardware has been checked).

Command result “log.txt” file will be automatically saved under the following folders:
C:\Users\user\AppData\Roaming\DediProg\SF Programmer



```
2010-Mar-17 14:06:18    USB communication = true
2010-Mar-17 14:06:26    Site#1, PASS
2010-Mar-17 14:06:26    Site#2, PASS
```

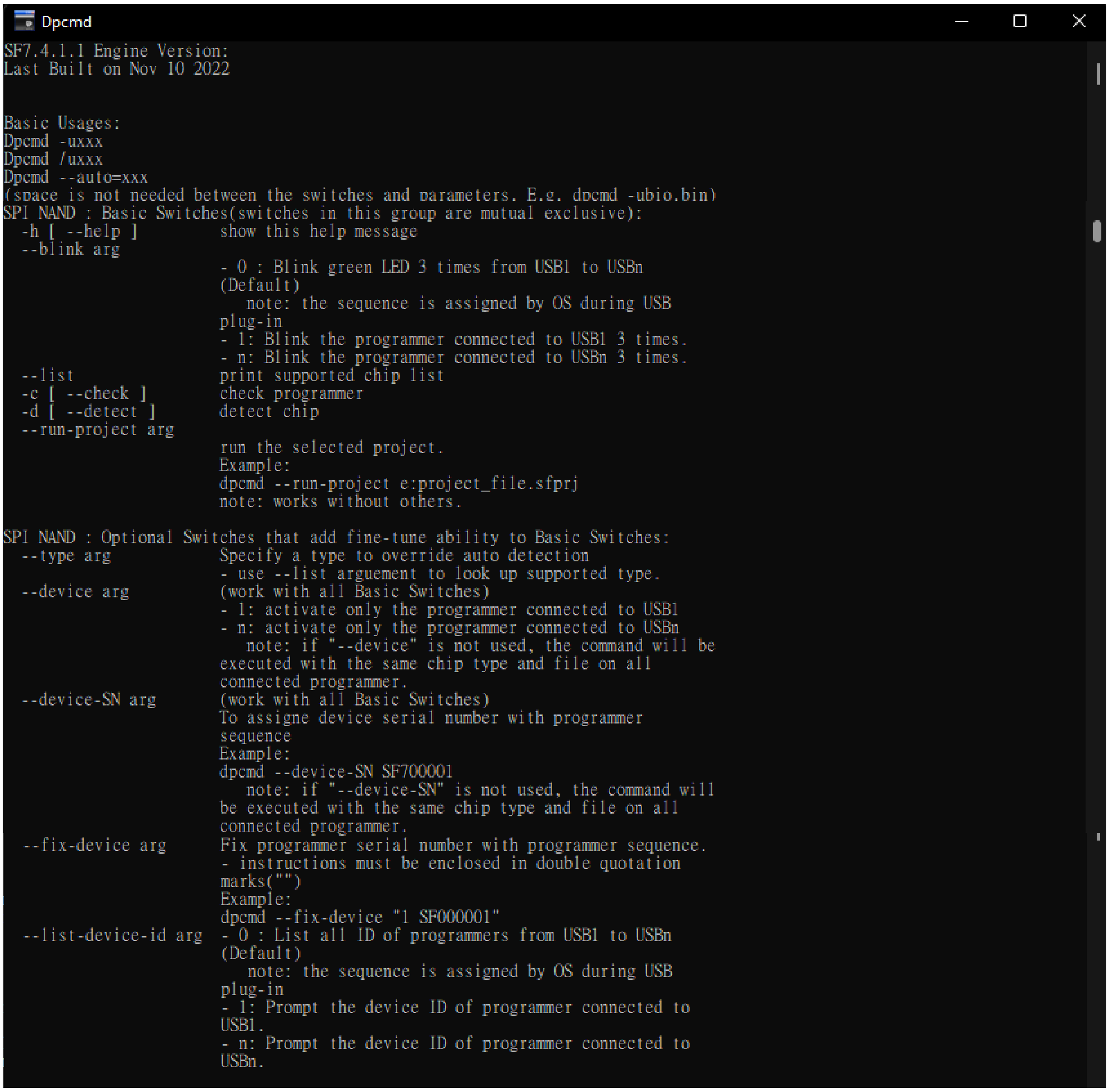
This .txt file must be checked to make sure that the operation has been successful. Time stamp can also be checked to be sure that the result has been updated with a new value.

The following are the error messages in the log.txt file.

FAIL Identify Fail
FAIL Blank Fail
FAIL Erase Fail
FAIL Program Fail
FAIL Read Fail
FAIL Send Specific data Fail
FAIL Verify Fail
FAIL Load Project Fail
FAIL Save Fail
FAIL Unknown

To get more information about these methods, please contact with DediProg.

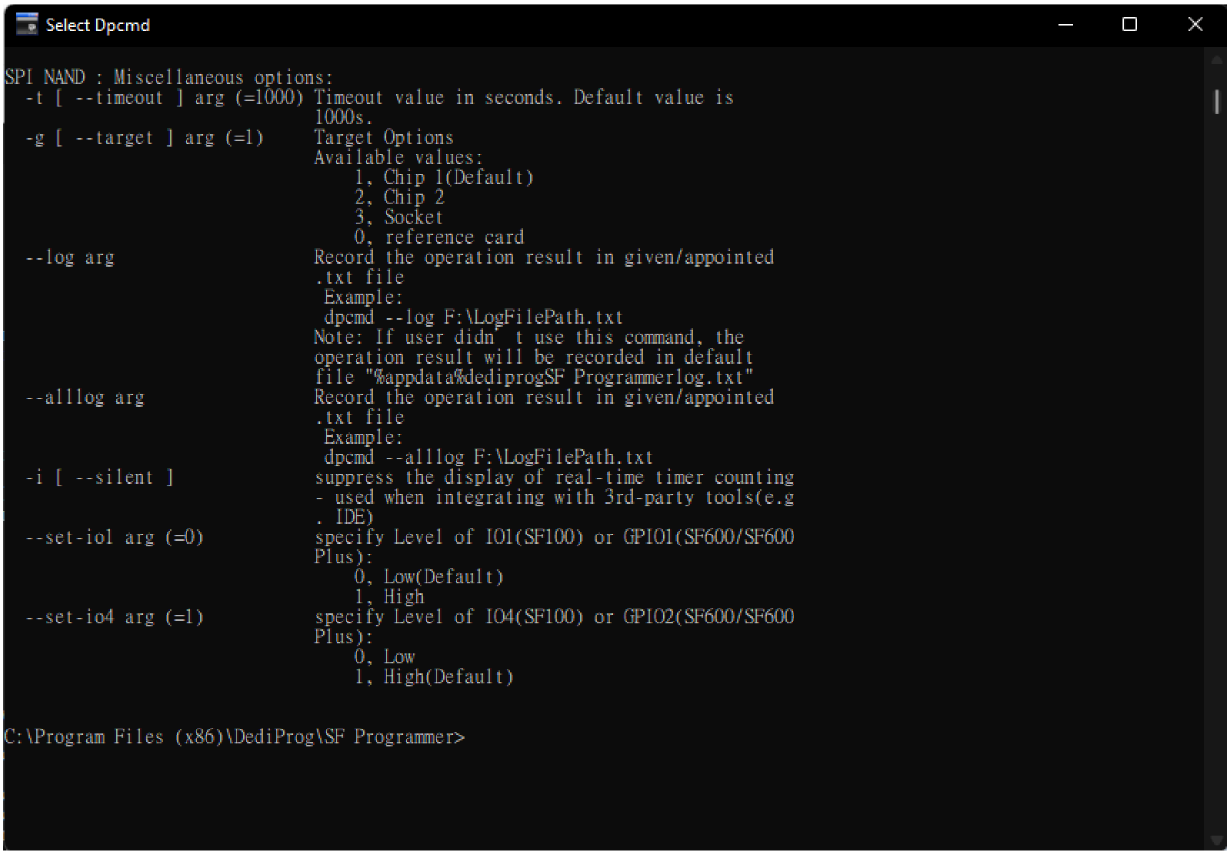
Window DOS Command



```
SF7.4.1.1 Engine Version:
Last Built on Nov 10 2022


Basic Usages:
Dpcmd -uxxx
Dpcmd /uxxx
Dpcmd --auto=xxx
(space is not needed between the switches and parameters. E.g. dpcmd -ubio.bin)
SPI NAND : Basic Switches(switches in this group are mutual exclusive):
  -h [ --help ]         show this help message
  --blink arg
                        - 0 : Blink green LED 3 times from USB1 to USBn
                        (Default)
                           note: the sequence is assigned by OS during USB
                        plug-in
                        - 1: Blink the programmer connected to USB1 3 times.
                        - n: Blink the programmer connected to USBn 3 times.
  --list                print supported chip list
  -c [ --check ]        check programmer
  -d [ --detect ]       detect chip
  --run-project arg
                        run the selected project.
                        Example:
                        dpcmd --run-project e:project_file.sfprj
                        note: works without others.

SPI NAND : Optional Switches that add fine-tune ability to Basic Switches:
  --type arg            Specify a type to override auto detection
                        - use --list arguement to look up supported type.
  --device arg          (work with all Basic Switches)
                        - 1: activate only the programmer connected to USB1
                        - n: activate only the programmer connected to USBn
                           note: if "--device" is not used, the command will be
                        executed with the same chip type and file on all
                        connected programmer.
  --device-SN arg       (work with all Basic Switches)
                        To assigne device serial number with programmer
                        sequence
                        Example:
                        dpcmd --device-SN SF700001
                           note: if "--device-SN" is not used, the command will
                        be executed with the same chip type and file on all
                        connected programmer.
  --fix-device arg      Fix programmer serial number with programmer sequence.
                        - instructions must be enclosed in double quotation
                        marks("")
                        Example:
                        dpcmd --fix-device "1 SF000001"
  --list-device-id arg  - 0 : List all ID of programmers from USB1 to USBn
                        (Default)
                           note: the sequence is assigned by OS during USB
                        plug-in
                        - 1: Prompt the device ID of programmer connected to
                        USB1.
                        - n: Prompt the device ID of programmer connected to
                        USBn.
```



```
SPI NAND : Miscellaneous options:
  -t [ --timeout ] arg (=1000) Timeout value in seconds. Default value is
                               1000s.
  -g [ --target ] arg (=1)     Target Options
                               Available values:
                                   1, Chip 1(Default)
                                   2, Chip 2
                                   3, Socket
                                   0, reference card
  --log arg                    Record the operation result in given/appointed
                               .txt file
                                Example:
                                dpcmd --log F:\LogFilePath.txt
                               Note: If user didn' t use this command, the
                               operation result will be recorded in default
                               file "%appdata%dediprogSF Programmerlog.txt"
  --alllog arg                 Record the operation result in given/appointed
                               .txt file
                                Example:
                                dpcmd --alllog F:\LogFilePath.txt
  -i [ --silent ]              suppress the display of real-time timer counting
                               - used when integrating with 3rd-party tools(e.g
                               . IDE)
  --set-io1 arg (=0)           specify Level of IO1(SF100) or GPIO1(SF600/SF600
                               Plus):
                                   0, Low(Default)
                                   1, High
  --set-io4 arg (=1)           specify Level of IO4(SF100) or GPIO2(SF600/SF600
                               Plus):
                                   0, Low
                                   1, High(Default)

C:\Program Files (x86)\DediProg\SF Programmer>
```

5.2 How to Start

DediProg window dos command line software is executed by the file “dpcmd.exe.” There are three different ways to run the dos command line.

1. Double click on the “dpcmd” icon on your desktop and type in dpcmd and enter.

2. Change your dos directory to the same location where “dpcmd.exe” is located. C:\program files\DediProg\SF Programmer

3. Type in the following command to auto directs the dpcmd command to the “dpcmd.exe” location.

Set path=%path%;”c:\program files\dediprog\SF Programmer

5.3 Basic Usages

1. dpcmd --run-project D:\SPINAND_Project.sfprj
   Run the selected project

5.4 Basic Switches

-? [--help]	Show the help message
--blink arg	-0 : Blink green LED 3 times from USB 1 to USB n (Default) Note: the sequence is assigned by OS during USB plug-in -1 : Blink the programmer connected to USB 1 for 3 times -n : Blink the programmer connected to USB n for 3 times
--list	Print supported chip list
-c [--check]	Check programmer information
-d [--detect]	Detect chip
--run-project arg	Run the selected project. Example: Dpcmd --type GD5F4GM5UExxG --run-project D:\nand_prj.sfprj

5.5 Optional Switches

(Specify the following switches to change default values):

--type arg	Specify a type to override auto detection - Use –list argument to look up supported type.
--device arg	(Work with all Basic Switches) - 1 : activate only the programmer connected to USB1 - n : activate only the programmer connected to USBn Note: if "–device" is not used, the command will be executed with the same chip type and file on all connected programmer.
--fix-device arg	Fix programmer serial number with programmer sequence. - instructions must be enclosed in double quotation marks ("") Example: dpcmd --fix-device "1 DP000001"
--list-device-id arg	- 0 : List all ID of programmers from USB1 to USBn (Default) note: the sequence is assigned by OS during USB plug-in - 1 : Prompt the device ID of programmer connected to USB1. - n : Prompt the device ID of programmer connected to USBn.

Miscellaneous options:

※ **Note: The programming operation always uses the default value for command. For other settings, must add the wanted option to every command.**

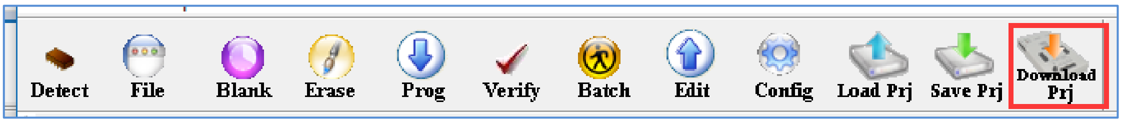
-t [--timeout] arg (=1000)	Timeout value in seconds. Default value is 1000s.
-g [--target] arg (=1)	Target Options Available values: 1, Chip 1(Default) 2, Chip 2 3, Socket 0, reference card
-i [--silent]	suppress the display of real-time timer counting - used when integrating with 3rd-party tools (e.g. IDE)
--vcc	Specify vcc 0, 3.5V 1, 2.5V 2, 1.8V 3, 1.2V 1200 ~ 3800, 1.2V ~ 3.8V (minimum step 100mV)
--spi-clk arg (=2)	specify SPI clock (SF100/ SF600): 2, 12 MHz/ 12MHz (Default) 0, 24 MHz/ 25MHz 1, 8 MHz/ 6MHz 3, 3 MHz/ 4MHz 4, 2.18 MHz/ 2MHz 5, 1.5 MHz/ 1MHz 6, 750 KHz/ 800MHz 7, 375 KHz/ 400MHz
--log arg	Record the operation result in given/appointed .txt file Example: dpcmd –log F:\LogFilePath.txt Note: If you didn’t use this command, the operation result will be recorded in the default file “%appdata%\dediprog\SF Programmer\log.txt”
--alllog arg	Record the operation result in given/appointed .txt file Example: dpcmd --alllog D:\LogFilePath.txt
--set-io1 arg (=0)	specify Level of IO1(SF100) or GPIO1(SF600/SF600Plus): 0, Low (Default) 1, High
--set-io4 arg (=1)	specify Level of IO4(SF100) or GPIO2(SF600/SF600Plus): 0, Low 1, High (Default)

5.6 Exit Code

```
enum ErrorCode
{
  EXCODE_PASS,
  EXCODE_FAIL_ERASE,
  EXCODE_FAIL_PROG,
  EXCODE_FAIL_VERIFY,
  EXCODE_FAIL_READ,
  EXCODE_FAIL_BLANK,
  EXCODE_FAIL_BATCH,
  EXCODE_FAIL_CHKSUM,
  EXCODE_FAIL_IDENTIFY,
  EXCODE_FAIL_FIRMWARE,
  EXCODE_FAIL_SAVELOG,
  EXCODE_FAIL_FIXDEVICE,
  EXCODE_FAIL_SAMEID,
  EXCODE_FAIL_RUNPROJECT,
  EXCODE_FAIL_SERIALSN,
  EXCODE_FAIL_LISTDEVICE,
  EXCODE_FAIL_BLINK,
  EXCODE_FAIL_DEVICE,
  EXCODE_FAIL_SWINIT,
  EXCODE_FAIL_PROGCONNECT,
EXCODE_FAIL_LOADFILEWITHVERIFY,
  EXCODE_FAIL_SAVEMEM,
  EXCODE_FAIL_OTHERS=99,
};
```

VI. Standalone Mode

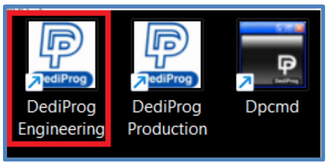
In addition to the functions provided by SF700 further allow you to download project to SF700 directly and program serial flash memories in standalone mode.



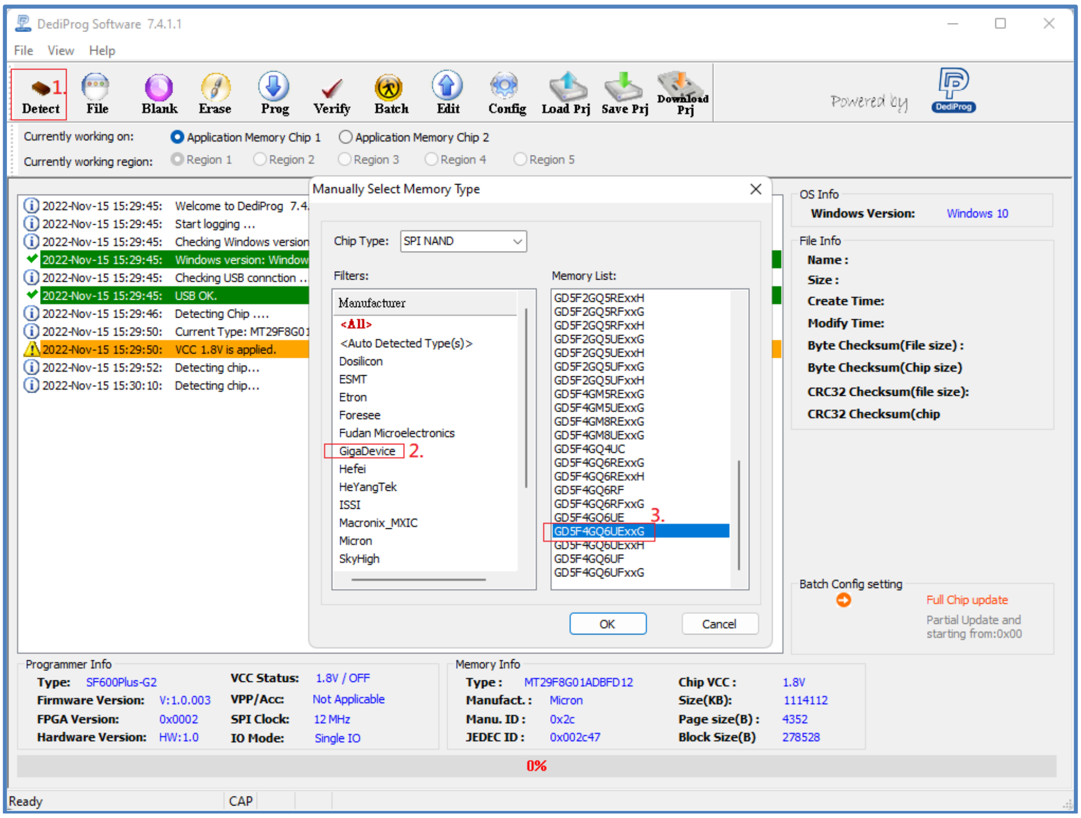
6.1 Project Preparation

Prepare a standalone programming project.

6.1.1 Open DediProg Engineer software



6.1.2 Select IC brand and part number



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2

Currently working region: ● Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

```
2022-Nov-15 15:29:45:   Welcome to DediProg  7.4.
2022-Nov-15 15:29:45:   Start logging ...
2022-Nov-15 15:29:45:   Checking Windows version
2022-Nov-15 15:29:45:   Windows version: Window
2022-Nov-15 15:29:45:   Checking USB connction ..
2022-Nov-15 15:29:45:   USB OK.
2022-Nov-15 15:29:46:   Detecting Chip ....
2022-Nov-15 15:29:50:   Current Type: MT29F8G01
2022-Nov-15 15:29:50:   VCC 1.8V is applied.
2022-Nov-15 15:29:52:   Detecting chip...
2022-Nov-15 15:30:10:   Detecting chip...
```

Chip Type: SPI NAND

Filters:

Manufacturer
- <All>
- <Auto Detected Type(s)>
- Dosilicon
- ESMT
- Etron
- Foresee
- Fudan Microelectronics
- GigaDevice
- Hefei
- HeYangTek
- ISSI
- Macronix_MXIC
- Micron
- SkyHigh

Memory List:
- GD5F2GQ5RExxH
- GD5F2GQ5RFxxG
- GD5F2GQ5RFxxH
- GD5F2GQ5UExxG
- GD5F2GQ5UExxH
- GD5F2GQ5UFxxG
- GD5F2GQ5UFxxH
- GD5F4GM5RExxG
- GD5F4GM5UExxG
- GD5F4GM8RExxG
- GD5F4GM8UExxG
- GD5F4GQ4UC
- GD5F4GQ6RExxG
- GD5F4GQ6RExxH
- GD5F4GQ6RF
- GD5F4GQ6RFxxG
- GD5F4GQ6UE
- GD5F4GQ6UExxG
- GD5F4GQ6UExxH
- GD5F4GQ6UF
- GD5F4GQ6UFxxG

OS Info
Windows Version: Windows 10

File Info
Name :
Size :
Create Time:
Modify Time:
Byte Checksum(File size) :
Byte Checksum(Chip size)
CRC32 Checksum(file size):
CRC32 Checksum(chip

Batch Config setting
Full Chip update
Partial Update and starting from:0x00

Programmer Info

Type:	SF600Plus-G2	**VCC Status:**	1.8V / OFF
Firmware Version:	V:1.0.003	**VPP/Acc:**	Not Applicable
FPGA Version:	0x0002	**SPI Clock:**	12 MHz
Hardware Version:	HW:1.0	**IO Mode:**	Single IO

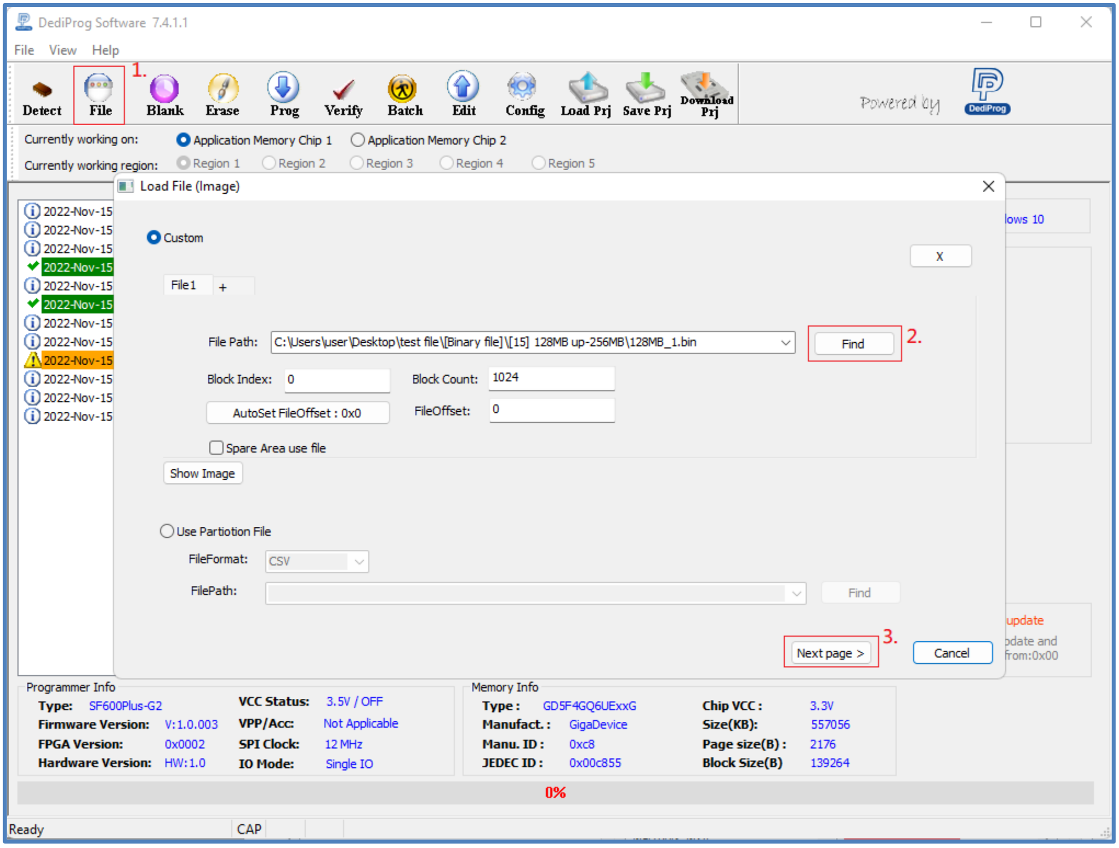
Memory Info

Type :	MT29F8G01ADBFD12	**Chip VCC :**	1.8V
Manufact. :	Micron	**Size(KB):**	1114112
Manu. ID :	0x2c	**Page size(B) :**	4352
JEDEC ID :	0x002c47	**Block Size(B)**	278528

0%

6.1.3 Load the programing file

See “Load File description” for more details.



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2

Currently working region: ○ Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

Load File (Image)

● Custom

File1

File Path: C:\Users\user\Desktop\test file\[Binary file]\[15] 128MB up-256MB\128MB_1.bin

Block Index: 0   Block Count: 1024

AutoSet FileOffset : 0x0   FileOffset: 0

□ Spare Area use file

Show Image

○ Use Partiotion File

FileFormat: CSV

FilePath:

Programmer Info

Type: SF600Plus-G2   VCC Status: 3.5V / OFF

Firmware Version: V:1.0.003   VPP/Acc: Not Applicable

FPGA Version: 0x0002   SPI Clock: 12 MHz

Hardware Version: HW:1.0   IO Mode: Single IO

Memory Info

Type : GD5F4GQ6UExxG   Chip VCC : 3.3V

Manufact. : GigaDevice   Size(KB): 557056

Manu. ID : 0xc8   Page size(B) : 2176

JEDEC ID : 0x00c855   Block Size(B) 139264

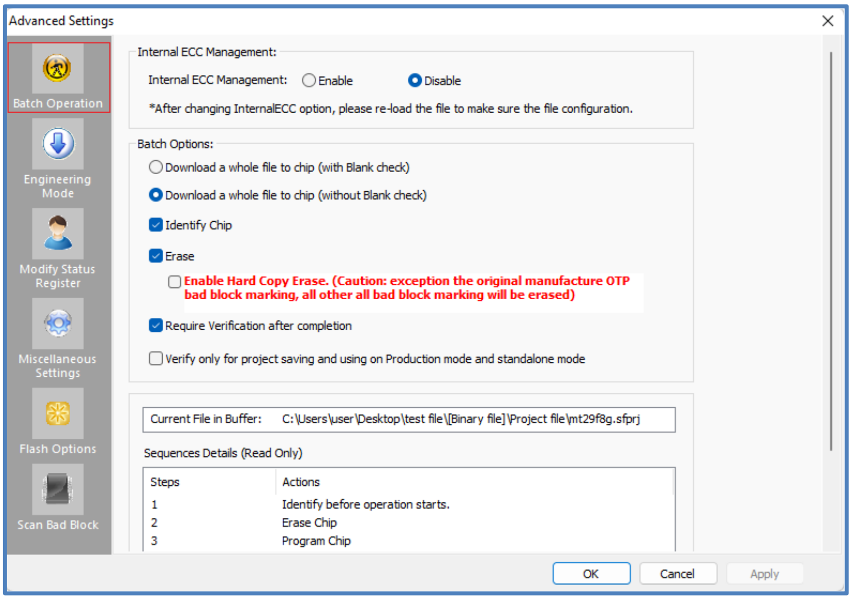
0%

Ready CAP

6.1.4 Click “Config” icon to set programming flow

※ Important Notice:

“Identify Chip” is necessary for Standalone programming. Make sure to select “Identify Chip” in programming flow.



Internal ECC Management:

Internal ECC Management: ○ Enable ● Disable

*After changing InternalECC option, please re-load the file to make sure the file configuration.

Batch Options:

○ Download a whole file to chip (with Blank check)

● Download a whole file to chip (without Blank check)

☑ Identify Chip

☑ Erase

□ Enable Hard Copy Erase. (Caution: exception the original manufacture OTP bad block marking, all other all bad block marking will be erased)

☑ Require Verification after completion

□ Verify only for project saving and using on Production mode and standalone mode

Current File in Buffer: C:\Users\user\Desktop\test file\[Binary file]\Project file\mt29f8g.sfprj

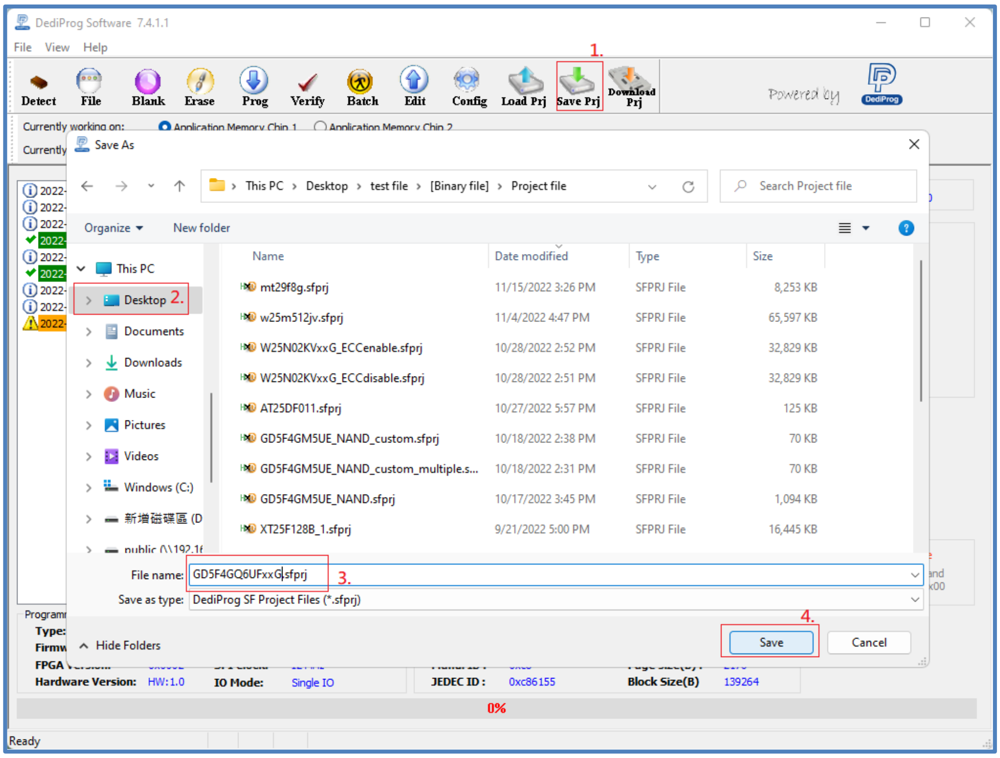
Sequences Details (Read Only)

Steps	Actions
1	Identify before operation starts.
2	Erase Chip
3	Program Chip

6.1.5 Choosing Standalone start mode

※ Note: SF700 only supports Start from Programmer Button.

6.1.6 Save .sfprj file to PC



1. Save Prj

2. Desktop

Name	Date modified	Type	Size
mt29f8g.sfprj	11/15/2022 3:26 PM	SFPRJ File	8,253 KB
w25m512jv.sfprj	11/4/2022 4:47 PM	SFPRJ File	65,597 KB
W25N02KVxxG_ECCenable.sfprj	10/28/2022 2:52 PM	SFPRJ File	32,829 KB
W25N02KVxxG_ECCdisable.sfprj	10/28/2022 2:51 PM	SFPRJ File	32,829 KB
AT25DF011.sfprj	10/27/2022 5:57 PM	SFPRJ File	125 KB
GD5F4GM5UE_NAND_custom.sfprj	10/18/2022 2:38 PM	SFPRJ File	70 KB
GD5F4GM5UE_NAND_custom_multiple.s...	10/18/2022 2:31 PM	SFPRJ File	70 KB
GD5F4GM5UE_NAND.sfprj	10/17/2022 3:45 PM	SFPRJ File	1,094 KB
XT25F128B_1.sfprj	9/21/2022 5:00 PM	SFPRJ File	16,445 KB

3. File name: GD5F4GQ6UFxxG.sfprj

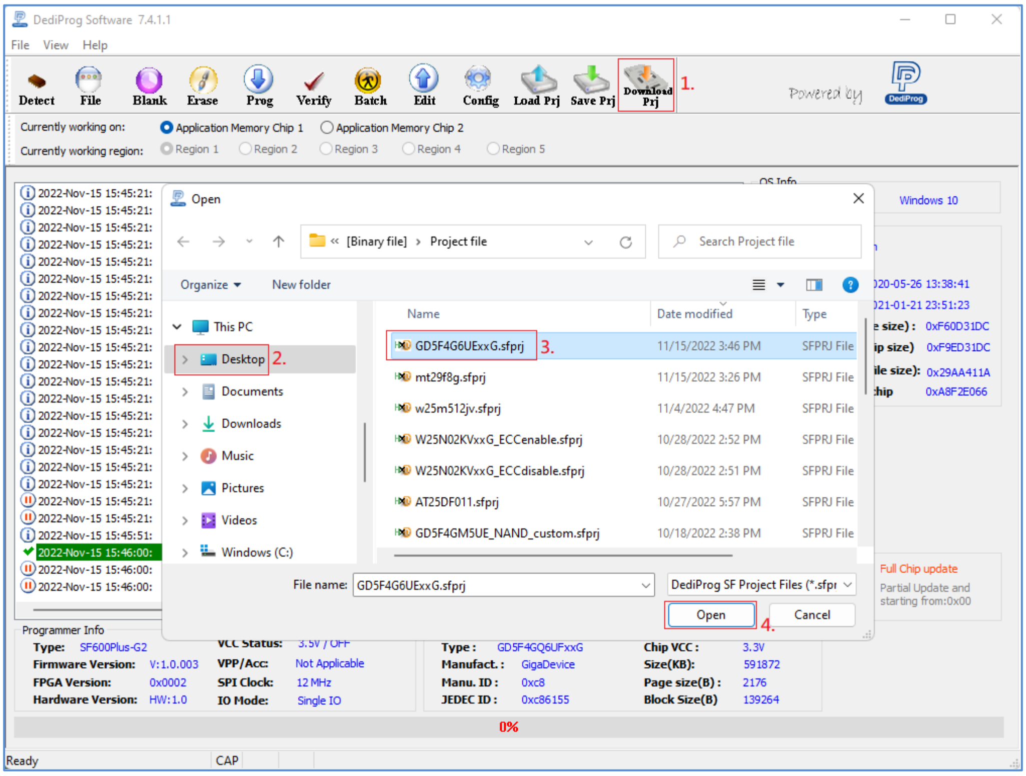
Save as type: DediProg SF Project Files (*.sfprj)

4. Save

Hardware Version: HW:1.0 IO Mode: Single IO JEDEC ID : 0xc86155 Block Size(B) 139264

0%

6.1.7 Press “Download Prj” button to download project to programmer embedded memory



1. Download Prj

Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2

Currently working region: ○ Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

2. Desktop

Name	Date modified	Type
GD5F4G6UExxG.sfprj	11/15/2022 3:46 PM	SFPRJ File
mt29f8g.sfprj	11/15/2022 3:26 PM	SFPRJ File
w25m512jv.sfprj	11/4/2022 4:47 PM	SFPRJ File
W25N02KVxxG_ECCenable.sfprj	10/28/2022 2:52 PM	SFPRJ File
W25N02KVxxG_ECCdisable.sfprj	10/28/2022 2:51 PM	SFPRJ File
AT25DF011.sfprj	10/27/2022 5:57 PM	SFPRJ File
GD5F4GM5UE_NAND_custom.sfprj	10/18/2022 2:38 PM	SFPRJ File

3. GD5F4G6UExxG.sfprj

File name: GD5F4G6UExxG.sfprj DediProg SF Project Files (*.sfpr

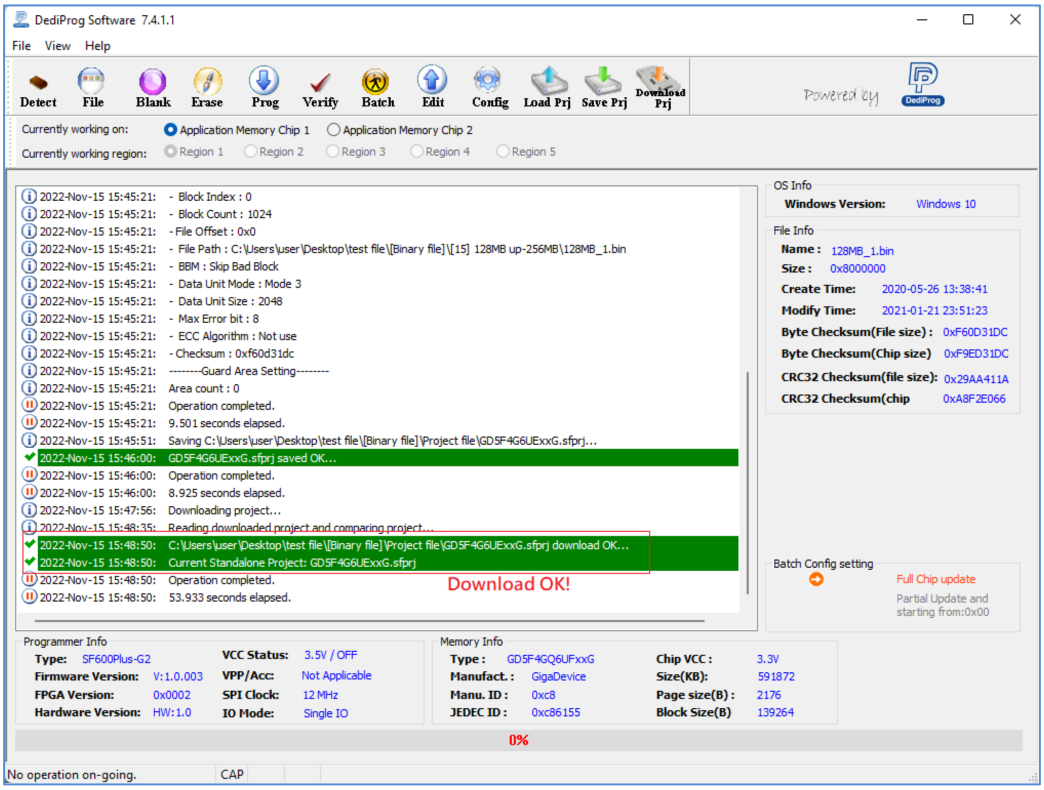
4. Open

Programmer Info

Type:	SF600Plus-G2	VCC Status:	3.5V / OFF	Type :	GD5F4GQ6UFxxG	Chip VCC :	3.3V
Firmware Version:	V:1.0.003	VPP/Acc:	Not Applicable	Manufact. :	GigaDevice	Size(KB):	591872
FPGA Version:	0x0002	SPI Clock:	12 MHz	Manu. ID :	0xc8	Page size(B) :	2176
Hardware Version:	HW:1.0	IO Mode:	Single IO	JEDEC ID :	0xc86155	Block Size(B)	139264

0%

6.1.8 Download project successfully



Currently working on: ● Application Memory Chip 1 ○ Application Memory Chip 2

Currently working region: ○ Region 1 ○ Region 2 ○ Region 3 ○ Region 4 ○ Region 5

```
2022-Nov-15 15:45:21:   - Block Index : 0
2022-Nov-15 15:45:21:   - Block Count : 1024
2022-Nov-15 15:45:21:   - File Offset : 0x0
2022-Nov-15 15:45:21:   - File Path : C:\Users\user\Desktop\test file\[Binary file]\[15] 128MB up-256MB\128MB_1.bin
2022-Nov-15 15:45:21:   - BBM : Skip Bad Block
2022-Nov-15 15:45:21:   - Data Unit Mode : Mode 3
2022-Nov-15 15:45:21:   - Data Unit Size : 2048
2022-Nov-15 15:45:21:   - Max Error bit : 8
2022-Nov-15 15:45:21:   - ECC Algorithm : Not use
2022-Nov-15 15:45:21:   - Checksum : 0xf60d31dc
2022-Nov-15 15:45:21:   --------Guard Area Setting--------
2022-Nov-15 15:45:21:   Area count : 0
2022-Nov-15 15:45:21:   Operation completed.
2022-Nov-15 15:45:21:   9.501 seconds elapsed.
2022-Nov-15 15:45:51:   Saving C:\Users\user\Desktop\test file\[Binary file]\Project file\GD5F4G6UExxG.sfprj...
✓ 2022-Nov-15 15:46:00:   GD5F4G6UExxG.sfprj saved OK...
2022-Nov-15 15:46:00:   Operation completed.
2022-Nov-15 15:46:00:   8.925 seconds elapsed.
2022-Nov-15 15:47:56:   Downloading project...
2022-Nov-15 15:48:35:   Reading downloaded project and comparing project...
✓ 2022-Nov-15 15:48:50:   C:\Users\user\Desktop\test file\[Binary file]\Project file\GD5F4G6UExxG.sfprj download OK...
✓ 2022-Nov-15 15:48:50:   Current Standalone Project: GD5F4G6UExxG.sfprj
2022-Nov-15 15:48:50:   Operation completed.
2022-Nov-15 15:48:50:   53.933 seconds elapsed.
```

Download OK!

OS Info
Windows Version: Windows 10

File Info
Name : 128MB_1.bin
Size : 0x8000000
Create Time: 2020-05-26 13:38:41
Modify Time: 2021-01-21 23:51:23
Byte Checksum(File size) : 0xF60D31DC
Byte Checksum(Chip size) 0xF9ED31DC
CRC32 Checksum(file size): 0x29AA411A
CRC32 Checksum(chip 0xA8F2E066

Batch Config setting
Full Chip update
Partial Update and starting from:0x00

Programmer Info

Type:	SF600Plus-G2	VCC Status:	3.5V / OFF
Firmware Version:	V:1.0.003	VPP/Acc:	Not Applicable
FPGA Version:	0x0002	SPI Clock:	12 MHz
Hardware Version:	HW:1.0	IO Mode:	Single IO

Memory Info

Type :	GD5F4GQ6UFxxG	Chip VCC :	3.3V
Manufact. :	GigaDevice	Size(KB):	591872
Manu. ID :	0xc8	Page size(B) :	2176
JEDEC ID :	0xc86155	Block Size(B)	139264

0%

No operation on-going. CAP

6.2 Standalone Programming

Start Standalone programming.

6.2.1 "Start from Programmer Button" mode

Press "**Start**" button for two seconds to run the project in Standalone mode.

6.2.2 "Start from Ctrl Port" mode

The Com Port design is for integrating SF600Plus-G2 with customer's system. All programmer pin outs (except 5V and NC) are default with Low status. Once customer/system sends a High signal to trigger START which needs hold for one second and make the programmer working (i.e. BUSY becomes High status accordingly), SF600Plus-G2 will also feedback PASS or FAIL result with High signal after programming.

VII. Firmware Support for Microsoft Windows

Check the Windows OS version and refer to the following table before you upgrade to the new firmware and software for SF700 and SF600Plus-G2.
If you are using Windows 8.1/10/11, please make sure the programmer firmware and SF software are the latest version. However, for older Windows OS version, there is no need to upgrade the programmer FW to the latest version.

You can download the latest version on DediProg website.
www.dediprog.com/download

SF700

Windows OS	Current Firmware Version	Upgrade Firmware	Upgrade Software
Win8.1/10/11	**7.x.x**	Latest firmware version (Please contact DediProg sales)	**SF 7.4.**x.x

SF600Plus-G2

Windows OS	Current Firmware Version	Upgrade Firmware	Upgrade Software
Win8.1/10/11	**x.x.x**	Latest firmware version (Please contact DediProg sales)	**7.4.x.x**

※ Please note that support and updates for older hardware versions are no longer available.

VIII. Revision History

Date	Version	Changes
2022/11/15	1.0	1. Separate SPI NAND description from the SPI NOR User manual. 2. Update the description for SPI NAND.

DediProg Technology Co., Ltd. (Headquarter)
No. 142, Ankang Rd., Neihu Dist., Taipei, Taiwan
TEL: 886-2-2790-7932 FAX: 886-2-2790-7916

Technical Support: **support@dediprog.com** Sales Support: **sales@dediprog.com**

Information furnished is believed to be accurate and reliable. However, DediProg assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties which may result from its use. Specifications mentioned in this Publication are subject to change without notice.
This Publication supersedes and replaces all information previously supplied.