DediProg SF Software User Manual

Version 7.8

Table of Content

I. Introduction

This user manual illustrates the usage of DediProg SF Software. The software is able to work with SF100, SF600, SF600*Plus* and SF700 programmers and Backup Boot Flash kit at the same time (SF100/SF600/SF600*Plus* only). However, it is not recommended. 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
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)** ". For other older OS version, please refer to "dp_SF User Manual_6.9"user manual.

III. DediProg SF Software Engineering GUI

DediProg SF software is suited for SF100, SF600, SF600*Plus*, SF700, and Backup Boot Flash Kit. 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 SF600*Plus*/SF700 for standalone programming purpose. After the software and USB driver are installed, please follow the steps below before running the software.

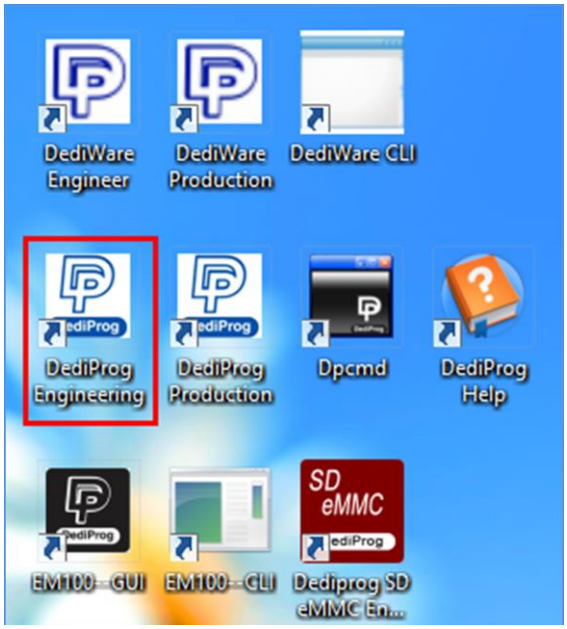
Four software icons will appear on your desktop after installation.
Icon "DediProg Engineering" is the engineering GUI, "DediProg Production" is the production GUI, "DpCmd" is the command line interface and "DediProg Help" is the user manual.

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.



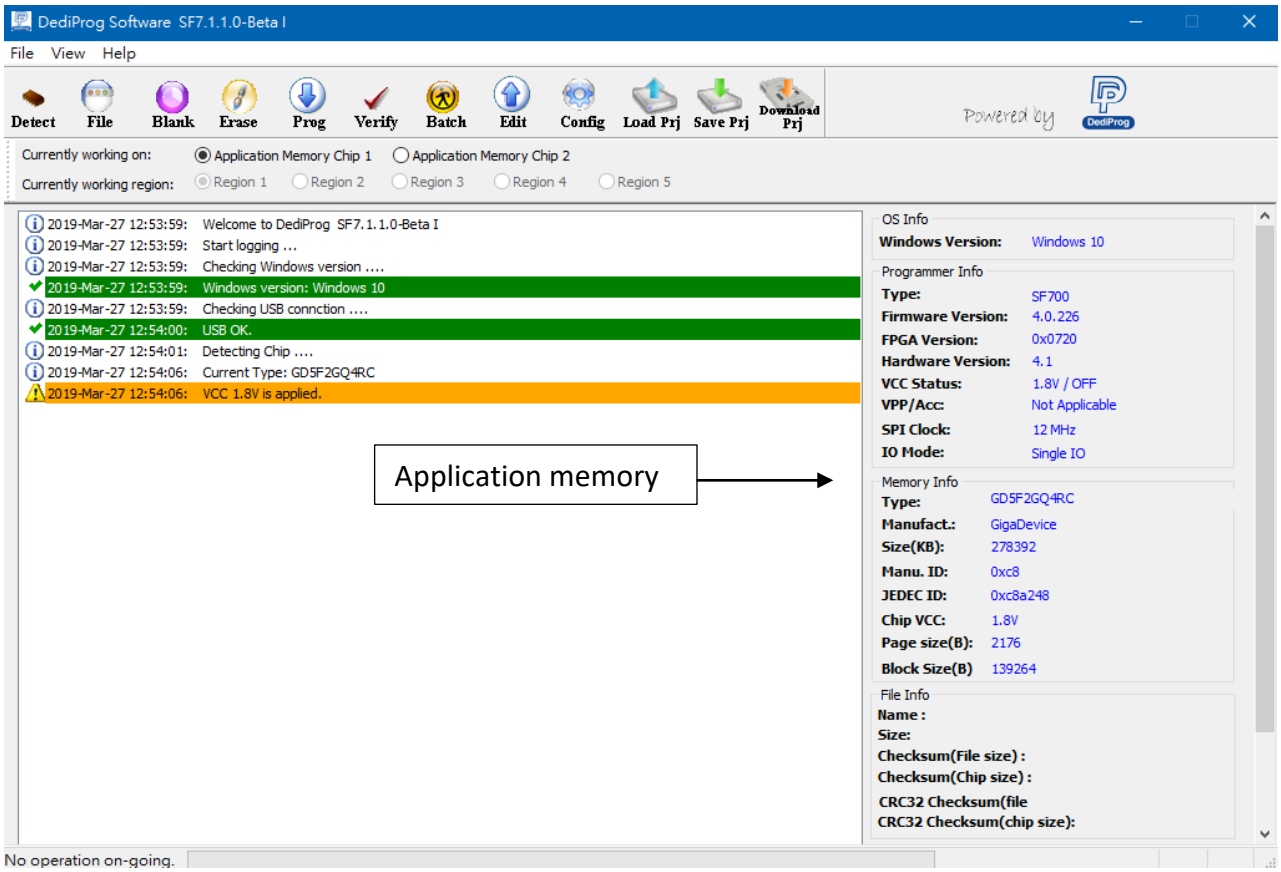
3.2 Identify the Target SPI Flash

SPI 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.

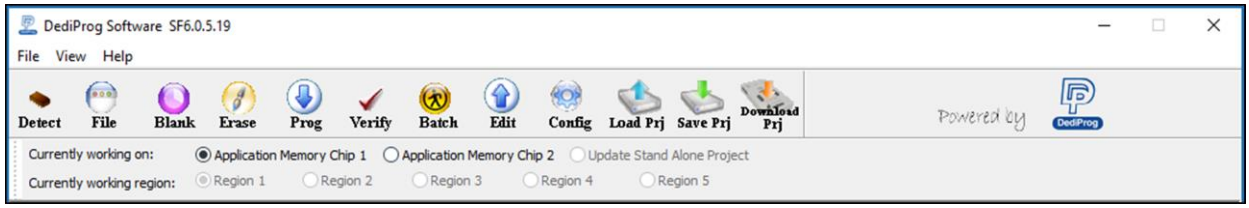
DediProg software will automatically identify the SPI Flash on the application board or the socket. You do not need to select SPI Flash's location.

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



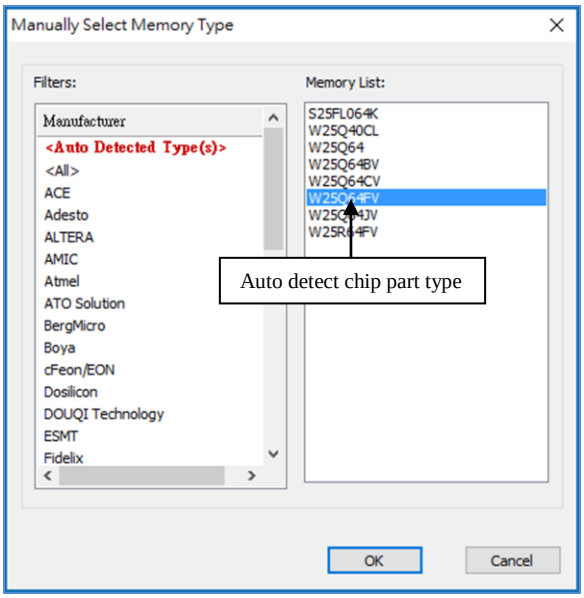
3.3 Tool Bar Description

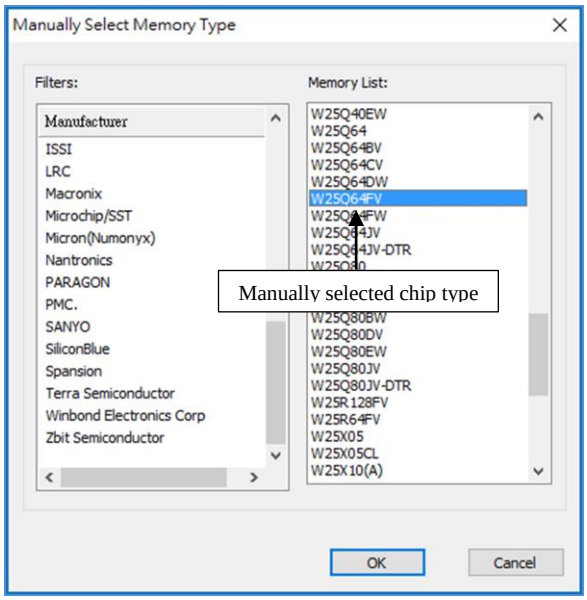
The tool bar provides all SPI Flash operations.



Detect

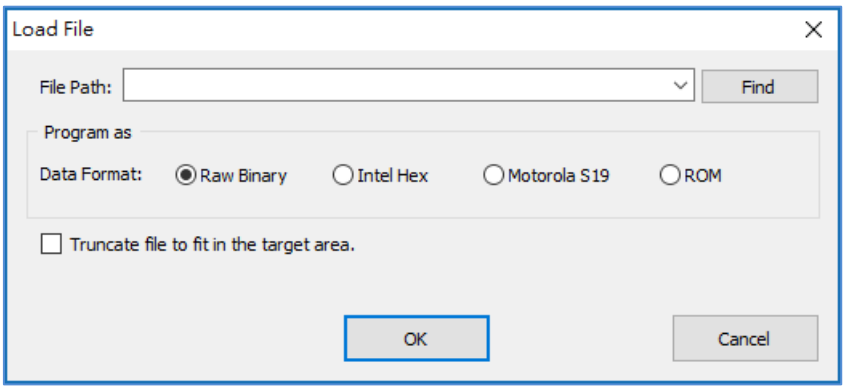
Detect Chip: when a new SPI Flash is placed, click this button to identify it and perform the operations. The auto detected chip types will be displayed on the right side of the screen. In case you would like to manually select a chip type, move the mouse over the chip manufacturer on the left screen,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





File

Select image: load the file you intend to program. The loaded file size cannot be larger the application SPI Flash size.



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

Erase
Erase SPI Flash: Erase the full content in a Serial Flash. After "Erase," the target serial flash shall be blank.

Prog
Program; program the selected image into the Serial Flash.

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

Batch
Batch operation: The programmer will perform a pre-configured set of operations such as (reload file + 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.

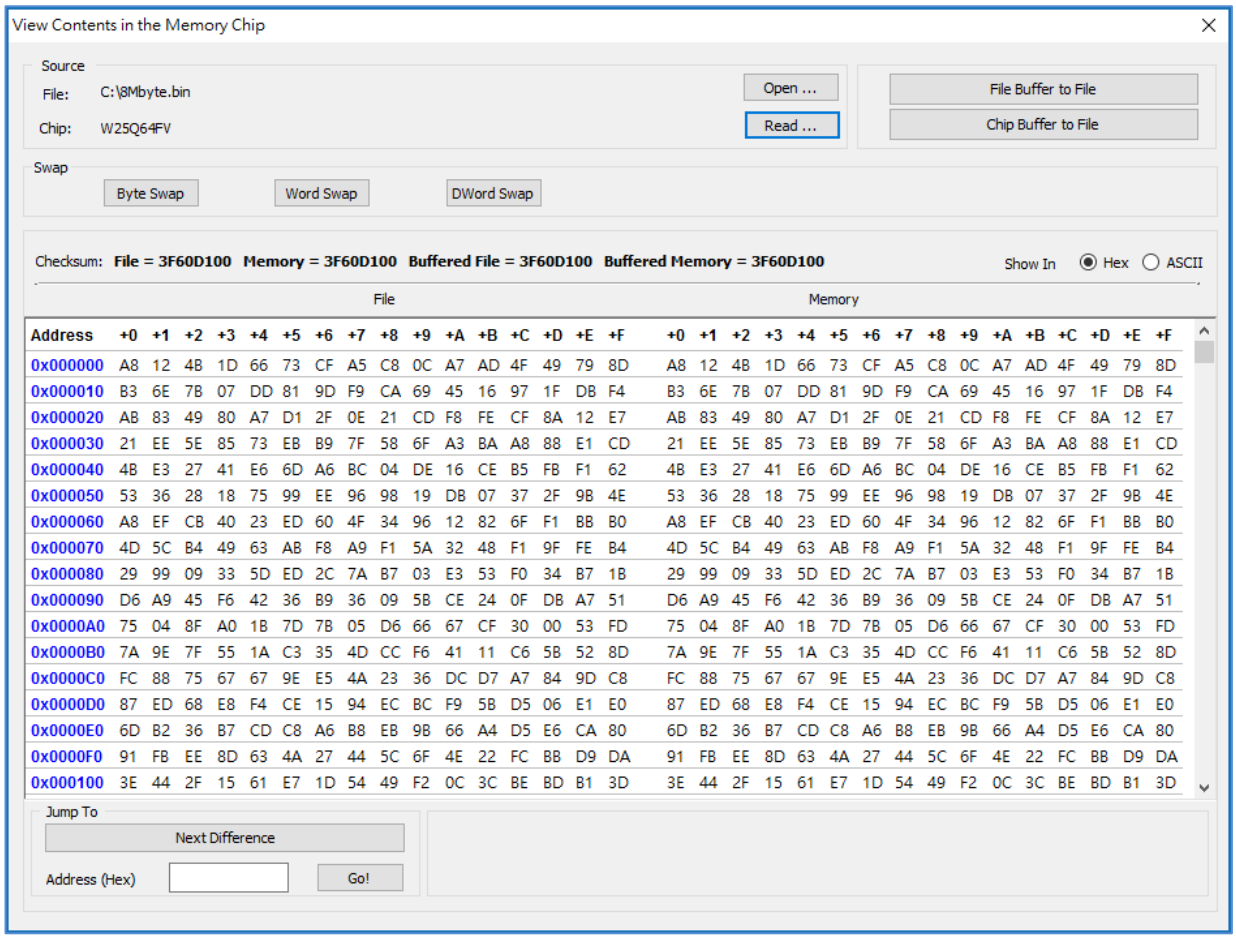
Download Prj
SF600*Plus*/SF700 only, please refer to Chapter 7- **VI. Stand Alone Mode (SF600*Plus/SF700* only).**

3.4 Edit Window Description

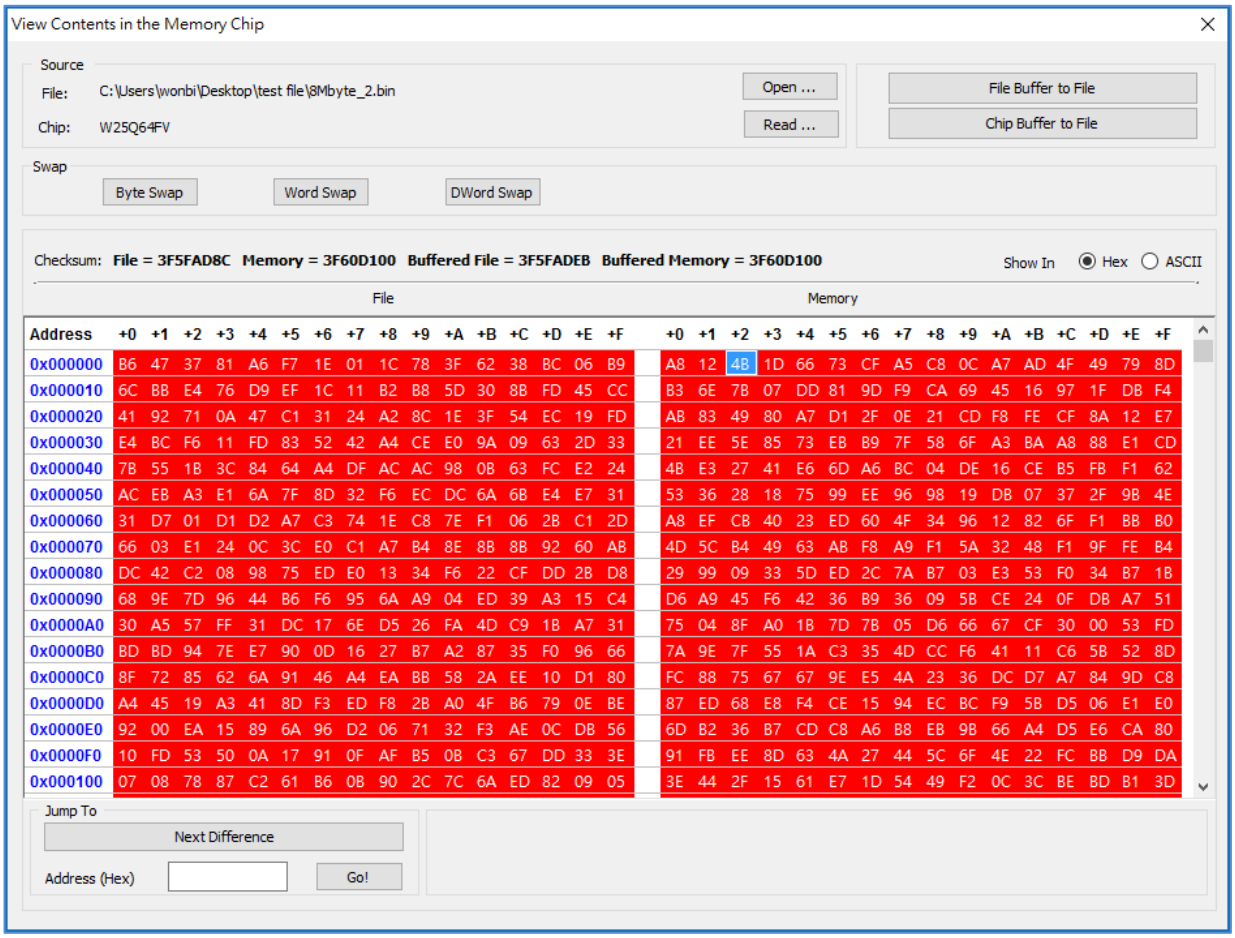
SPI Flash content display:

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, click “Read” to read and display the whole chip memory contents on the edit window. Checksum of the file contents and the chip contents will be displayed.



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.



Chip buffer to file

This will save the chip contents into a binary file; you can set up the file name and the location.

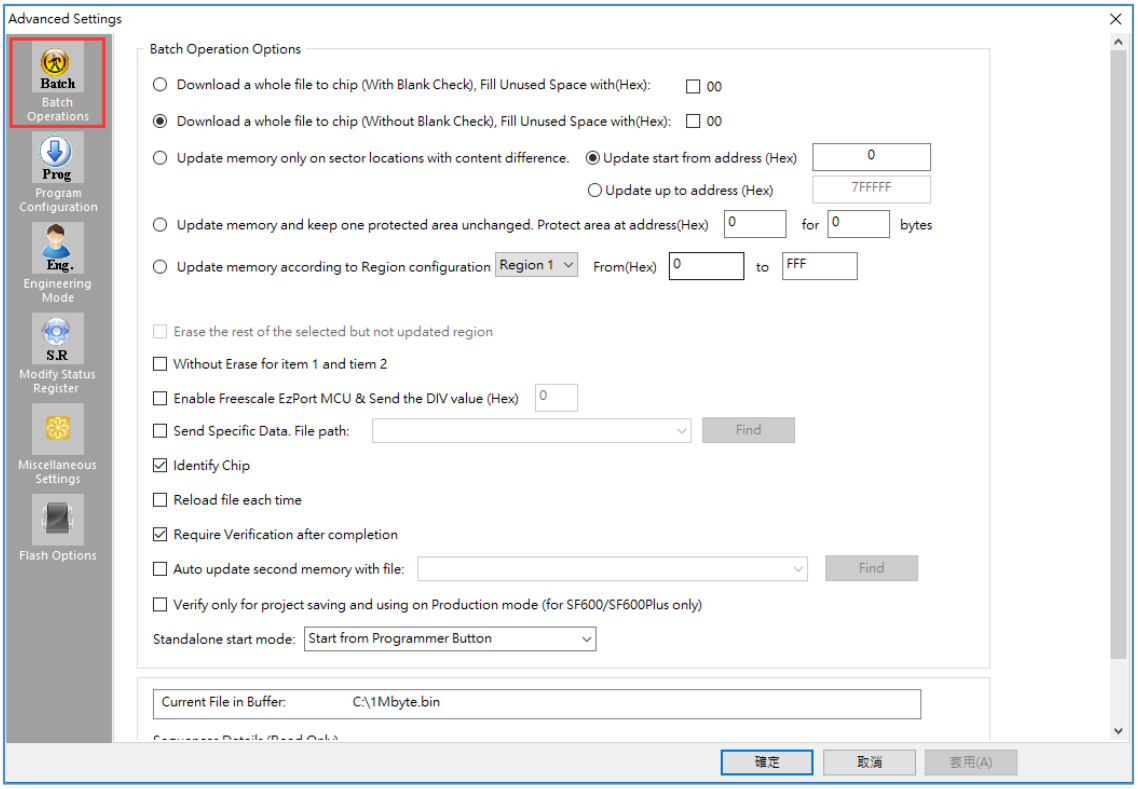
File buffer to file

File buffer can be modified in real time. This will save the file buffer contents into a binary file as well.

3.5 Configuration Window Description

This feature allows users to configure advanced settings.

3.5.1 Batch Operation Option



A. Download a whole file to chip (With Blank Check)

Click **Batch** button on the tool bar, the following operation will be automatically executed:

1) Read the memory content

2) Blank check (Check if the chip is erased. If it is blank, then it will jump to the programming step).

3) Erase the entire memory if it is not blank

4) Program the entire memory with the file

5) Verify if the memory content is identical with the programmed file.

B. Download a whole file to chip (Without Blank Check)

Clicks the **Batch** button on the tool bar, the following operation will be automatically executed:

1) Erase the entire memory

2) Program the entire memory with the file

3) Verify if the memory content is identical with the programmed file.

C. Update memory only on sector locations with content difference

You can select the sector locations of file to program.

- **Update start from address (Hex):**

  Program the entire file starting from the address that you enter.

- **Update up to address (Hex):**

  Program the entire file and ends at the address that you enter. The default ending address will be automatically calculated by the software according to memory's size.

Click the Batch button on the tool bar, the following operations will be automatically executed:

1) Read the memory content

2) Compare the memory content from the given address with the file at the 64KB sector base

3) Erase only the 64KB sectors with the differences

4) Program only the erased sectors with the file data of the corresponding address

5) Verify the data on the updated 64KB sectors

Smart Update can be used in the following cases:

- A small file can be programmed or updated at a given address without changing the rest of the memory (local update).
- A file with only a minor change compare to the memory content can be quickly updated. The sectors without difference are kept unchanged.

Remark:
The file data is identical with the target memory. Therefore, you will need to load the entire file, even if only programming a sector of it.

D. Update memory and keep one protected area unchanged

Click the Batch button on the tool bar, the following operations will be automatically executed:

1) Read the memory content from the given address of the given length

2) Insert the read memory contents into the file buffer

3) Erase the entire chip

4) Program the entire chip with the updated file in step 2

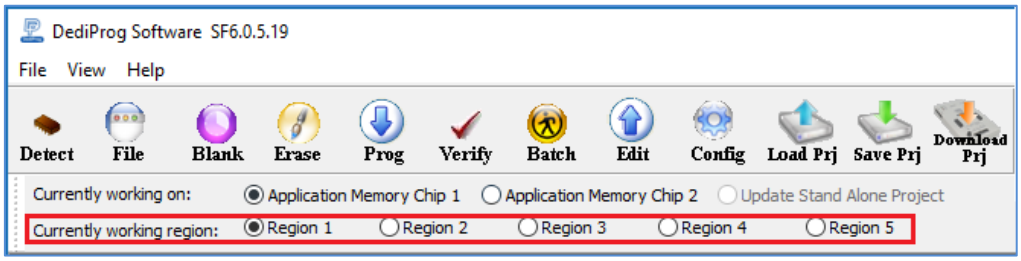
5) Verify the programmed data

E. Update memory according to Region configuration

When you only want to update some part of the data in SPI Flash, you can use this function to update the data in the assigned region. This function saves time when debugging.

1) Assign the Region and set start & end address of the Region.

2) Select working region



F. Erase the rest of the selected region but not updated space

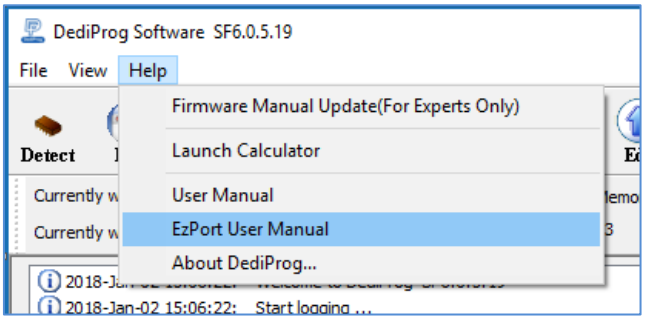
The software will update the selected region, and the rest of the selected region that are not updated will be erased.

G. Without Erase for item 1 and item 2

Remove erase operation from item 1 and item 2.

H. Enable Freescale EzPort MCU & Send the DIV value (Hex)

If the box is checked, the programmer will automatically enable EzPort. Details please see « Help →EzPort User Manual»



I. Send Specific Data

The software will load and send the engineering SPI sequence defined and saved in the “Engineering Mode” Configuration window. This option allows you to create your own SPI instruction.

J. Identify Chip

The software will identify the chip before operation starts.

K. Reload file each time

The software will load the same file from the source destination each time before the batch operations (refresh). This option is helpful when the other software updates the file in parallel (like compiler).

L. Require Verification after completion

The software will verify the contents between the source file and the programmed Serial Flash contents after the batch operations.

M. Auto update second memory with file

The software will auto update the second chip memory after chip 1 has been updated.

N. Verify only for project saving and using on Production mode

The Batch function does not support verify only feature on engineering mode. This feature is for project saving and allows verify only on Production mode and standalone mode.

Methods Comparison:

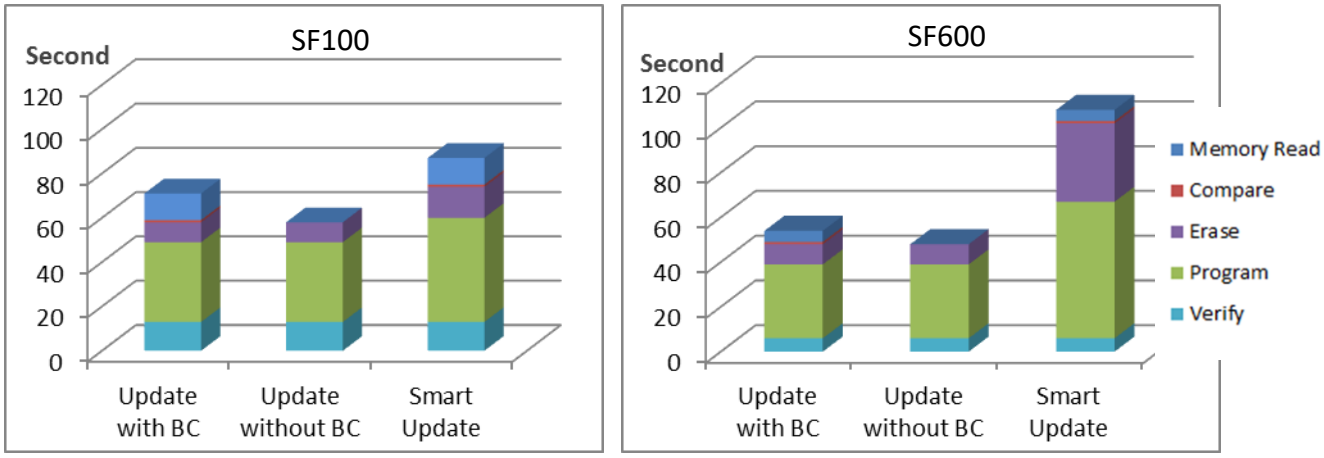
Case 1:

64Mb Serial flash update with 64Mb file that are totally different. Memory has previously been programmed and needs to be erased totally.

Function	Update with BC		Update without BC		Smart Update	
Model name	SF100	SF600	SF100	SF600	SF100	SF600
Memory Read	12	5	x	x	12	5
Compare	1	1	x	x	1	1
Erase	9	9	9	9	14	35
Program	36	33	36	33	47	61
Verify	13	6	13	6	13	6
TOTAL	**71**	**54**	**58**	**48**	**87**	**108**

Time unit: second

Comparison Chart



Conclusion:

If the memory needs to be completely erased for a file update, the "Update without Blank Check" is the optimum choice.

Time Saving:

SF100 saves 33%; SF600 saves 55%

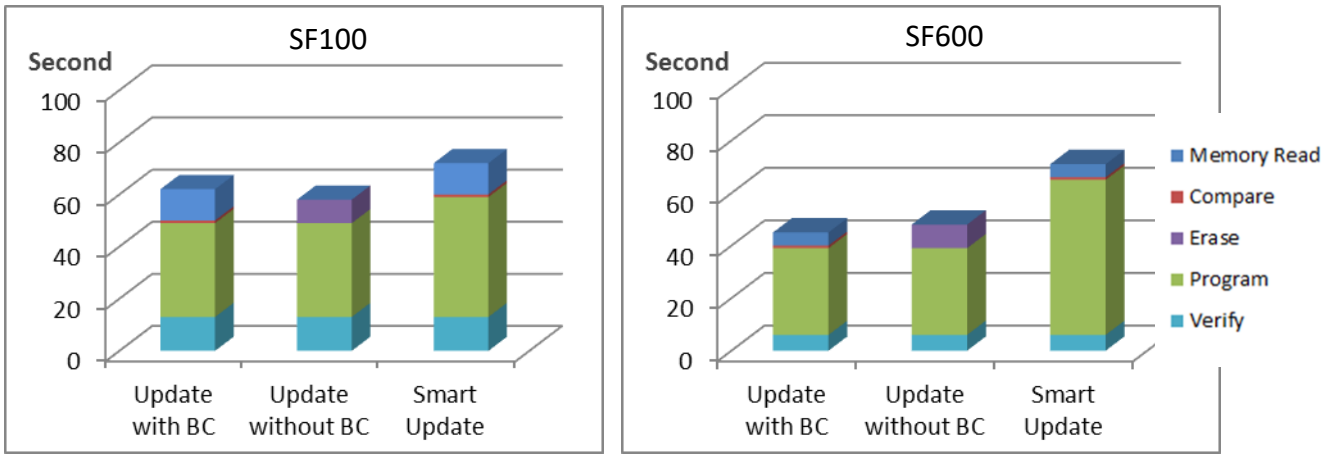
Case 2:

64Mb Serial flash programming with a 64Mb file. Memory has never been programmed (from supplier).

Function	Update with BC		Update without BC		Smart Update	
Model name	SF100	SF600	SF100	SF600	SF100	SF600
Memory Read	12	5	x	x	12	5
Compare	1	1	x	x	1	1
Erase	0	0	9	9	0	0
Program	36	33	36	33	46	59
Verify	13	6	13	6	13	6
TOTAL	**62**	**45**	**58**	**48**	**72**	**71**

Time unit: second

Comparison Chart



Conclusion:

If the memory is blank (from supplier), the “Update with Blank Check” or “Smart update” is the optimum choice.

Time Saving:

SF100 saves 19%; SF600 saves 37%

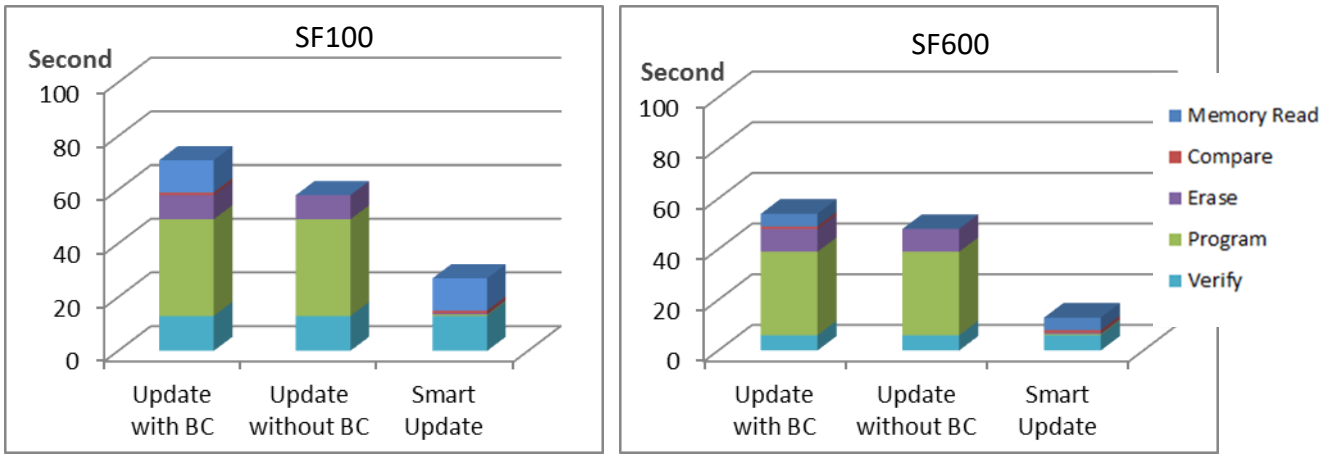
Case 3:

64Mb Serial flash update with a 64Mb file with only data differences on one block or a small file of one block size only at a specified address.

Function	Update with BC		Update without BC		Smart Update	
Model name	SF100	SF600	SF100	SF600	SF100	SF600
Memory Read	12	5	x	x	12	5
Compare	1	1	x	x	1	1
Erase	9	9	9	9	0.5	0.5
Program	36	33	36	33	0.5	0.5
Verify	13	6	13	6	13	6
TOTAL	**71**	**54**	**58**	**48**	**27**	**13**

Time unit: second

Comparison Chart



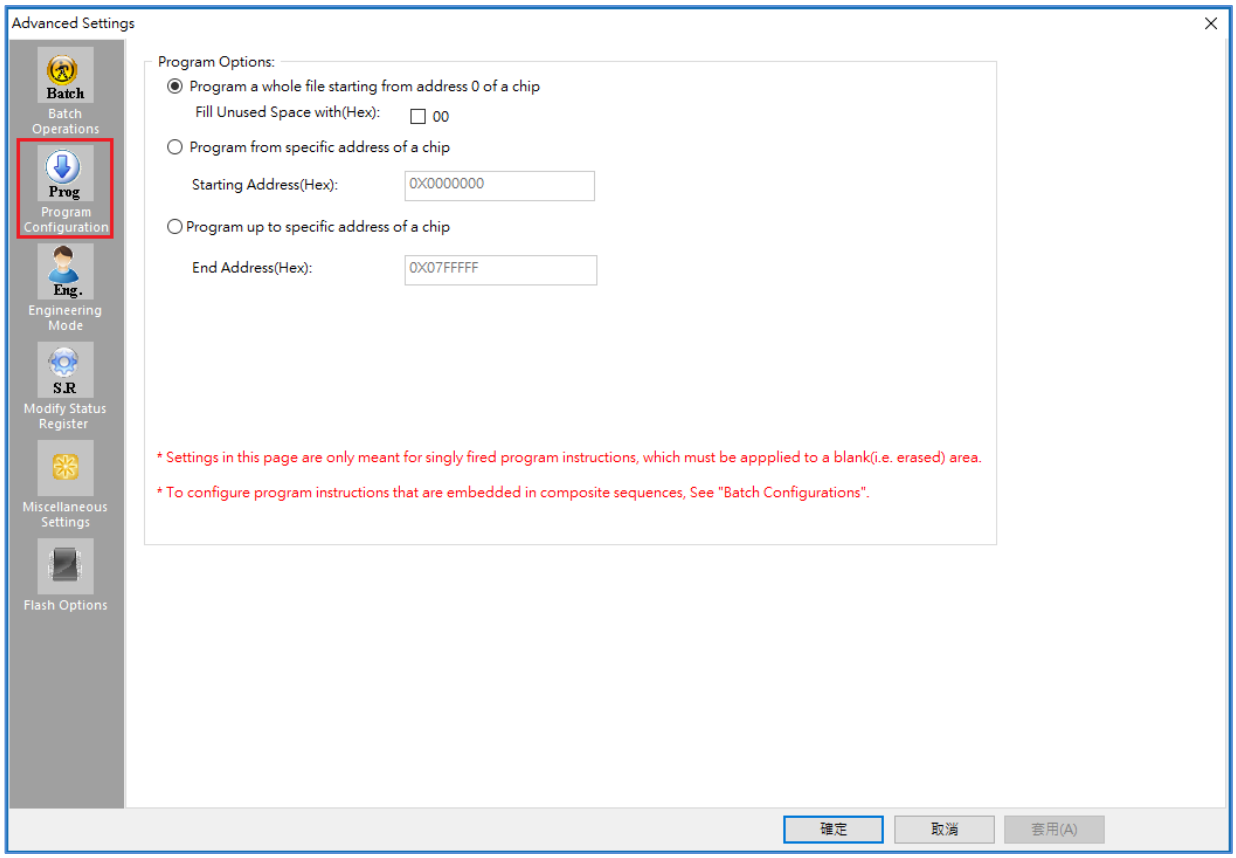
Conclusion:

If the difference between the memory content and the file are small or if the file that needs to be programmed is small, the “Smart update” is the optimum choice.

Time Saving:

SF100 saves 62%; SF600 saves 76%

3.5.2 Program Configurations



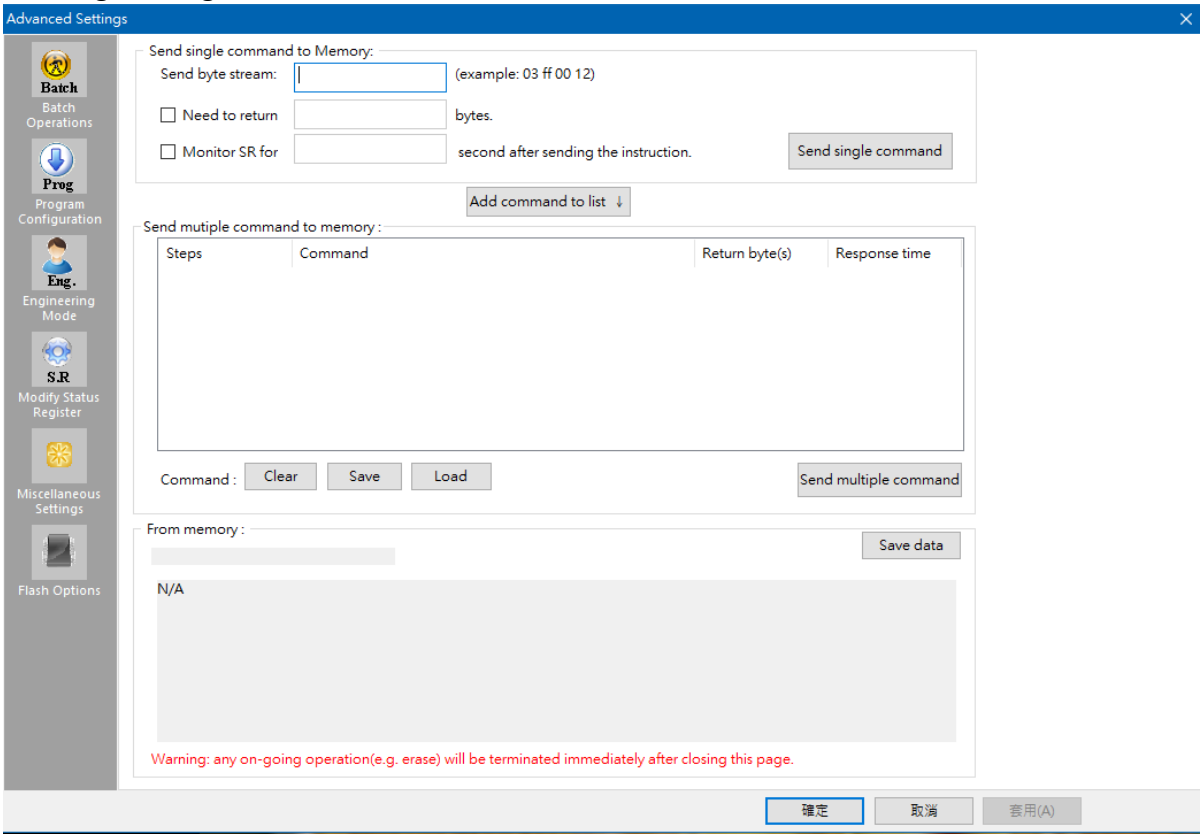
A. Program a whole file starting from address 0 of a chip

B. Program from specific address of a chip: To program the entire file starting from the address that you enter.

C. Program up to specific address of the chip: To program the entire file, ending at the last address of the chip. The default ending address will automatically be calculated by the software according to memory size.

If the file is smaller than the target Serial Flash, you can define how to fill the rest of the SPI Flash. By default FFh or 00h by selecting the box.

3.5.3 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.

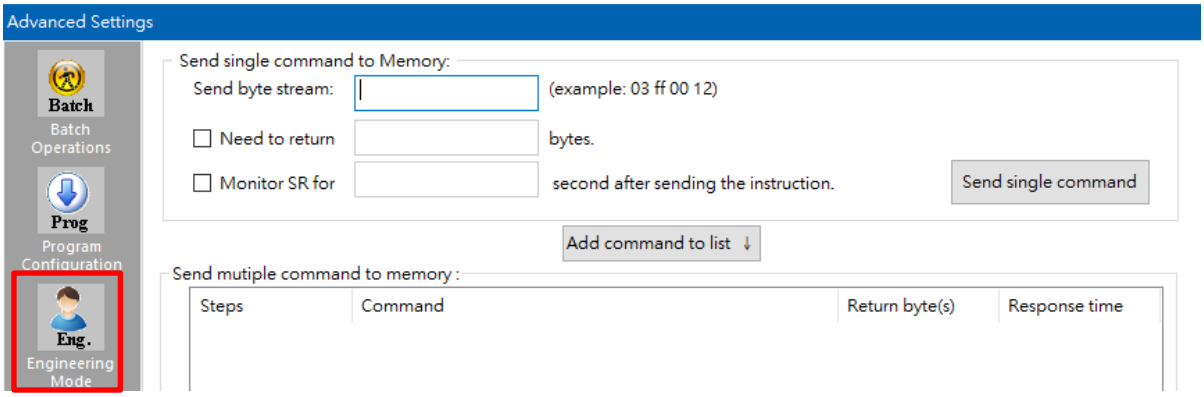
The engineering mode can be used for sending instruction to the SPI Flash.

3.5.3.1 Send single command to Memory

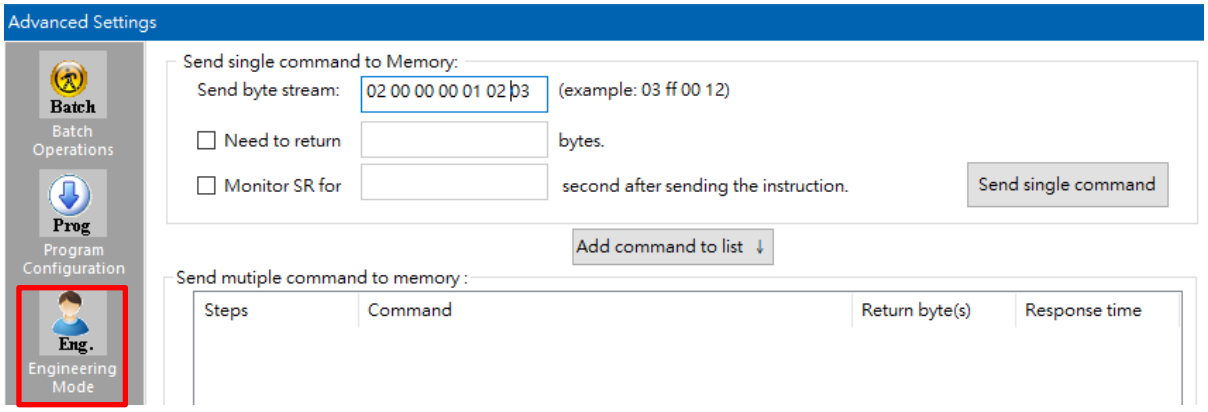
You can define the data bytes to be sent from the programmer to the SPI Flash and the number of bytes to be returned. Also define if the status register WIP bit has to be polled to check if the SPI Flash is busy or ready. Send single command by clicking “Send single command” button.

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

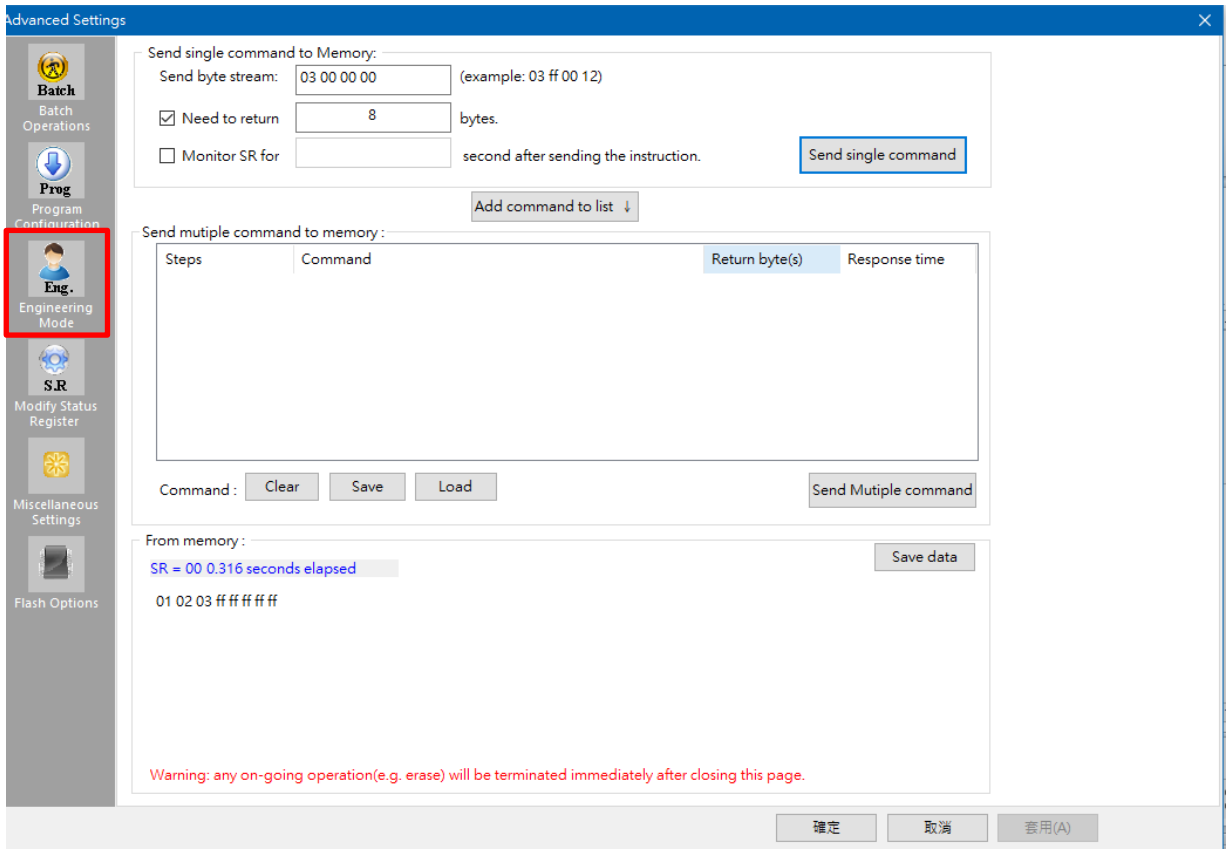
First: programmer needs to set the WEL bit by sending the WREN (06h) command to the SPI Flash as described below:



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:



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):

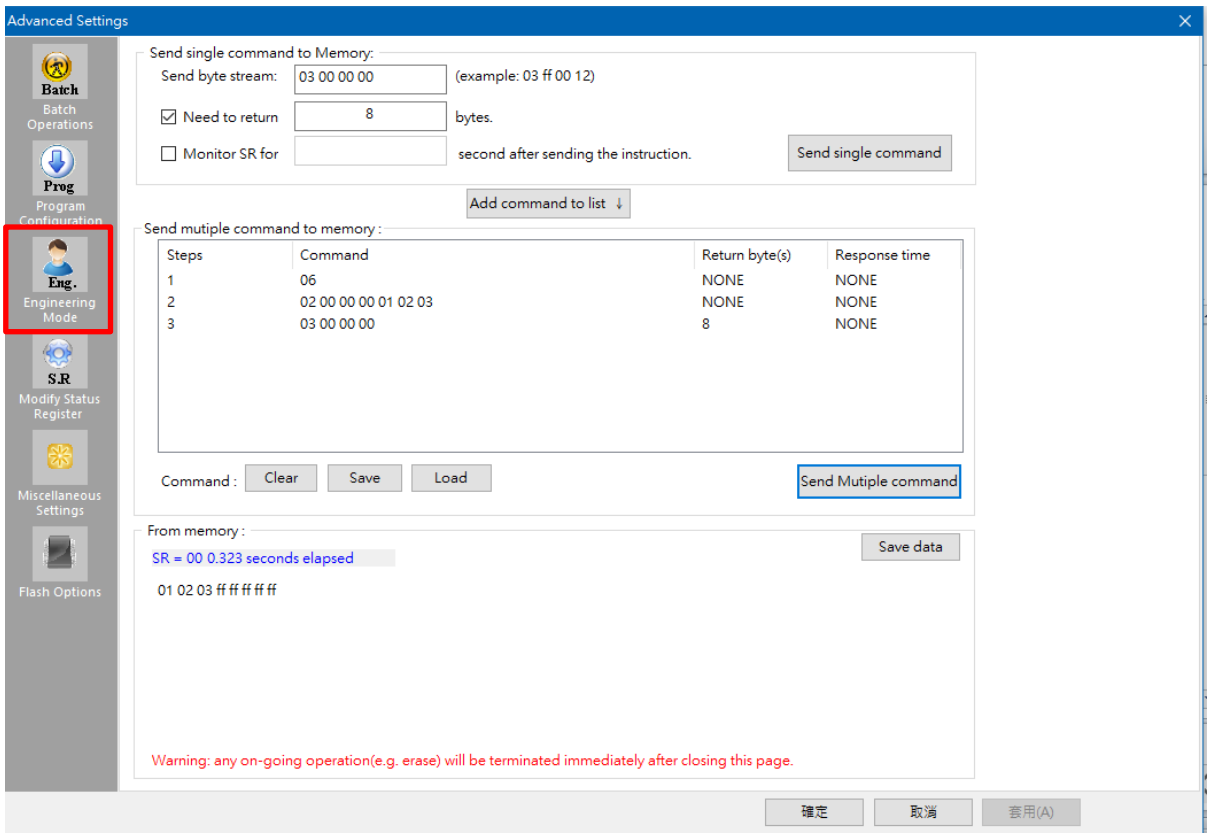


The return bytes from the SPI Flash are displayed in the “from memory” window.

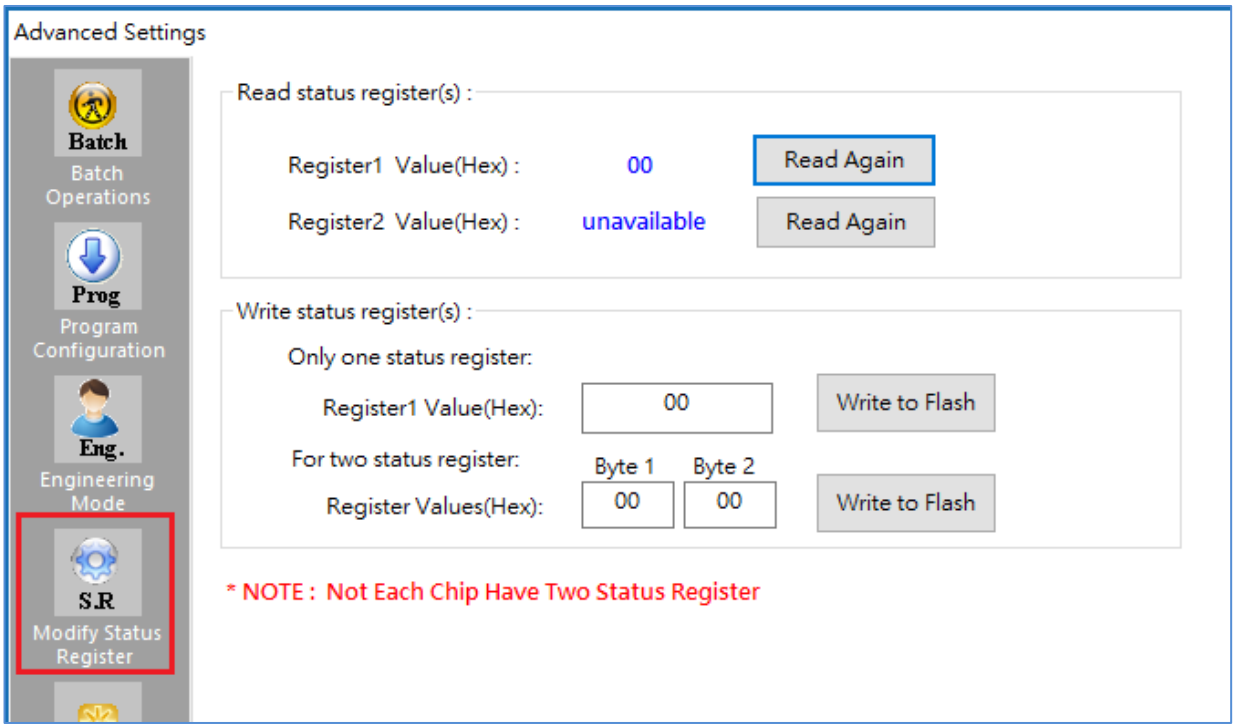
3.5.3.2 Send multiple commands

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.



3.5.4 Modify Status Register



This function allows you to modify or read the status register(s) value of the target serial flash.

Please note each chip has their own command to write status registers.

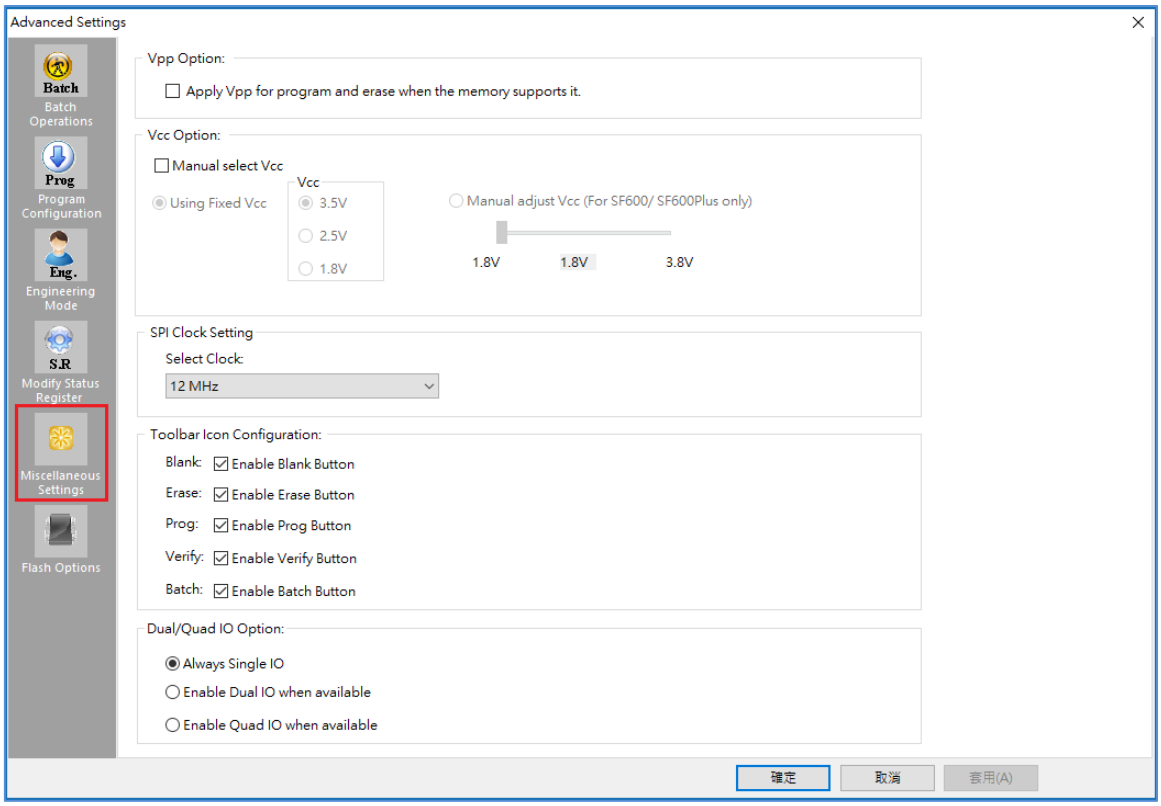
For the chip that only has one status register:

- For write: “06h” to set the Write Enable; “01h” and user data to write the status register
- For Read: “05h” to read the status register

For the chip that has two status registers:

- Please refer to the device specification for parameter setting.

3.5.5 Miscellaneous Settings



A. Vpp Option

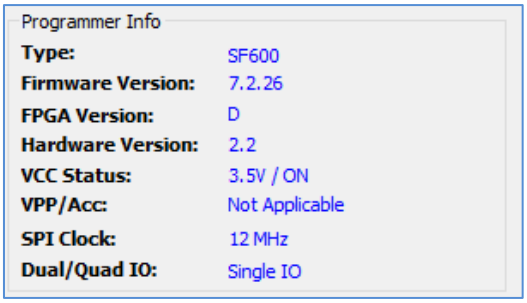
This setting enables the Vpp option so the High voltage is applied on the SPI Flash Wp pin to reduce the programming and the erasing time.

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

B. Vcc Option

SF series programmers support 3.5V, 2.5V, and 1.8V 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.

Note: Firmware version 4.x.x and early version of SF100 not support 1.8V.



C. SPI Clock Setting

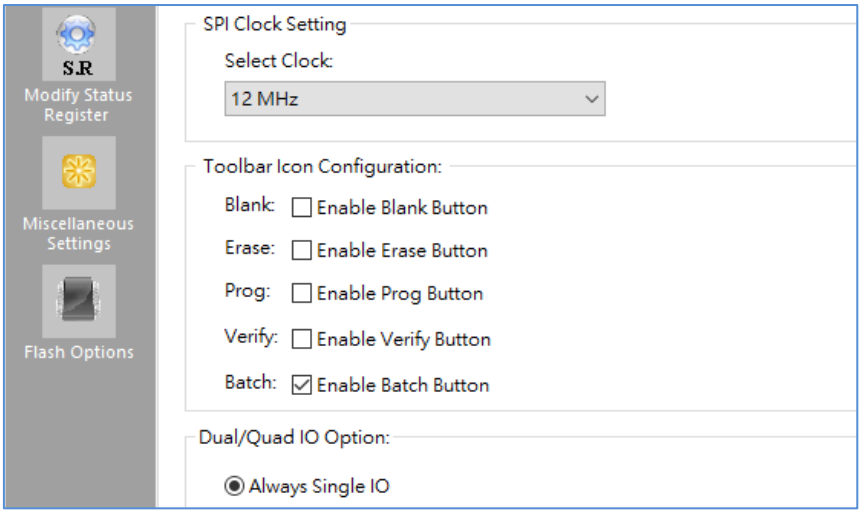
The SPI clock frequency can be adjusted by user to fit the application requirements or SPI Flash performance. Notice that the SPI Flash frequency is defined from the supplier specification for a maximum capacitance usually is 30pf or 15pF. The application is therefore designed not to exceed this maximum capacitance.

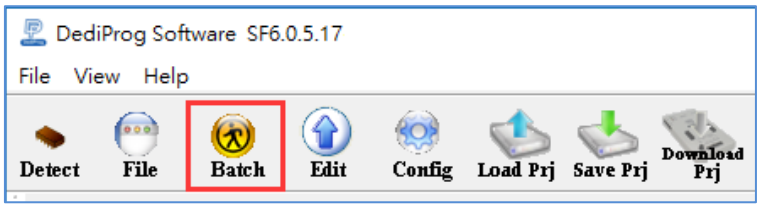
In-circuit programming does not fulfill anymore this original design as additional capacitance will be added according to the cable length and programmer. Therefore, you cannot expect to program on board SPI flash to the maximum frequency of the datasheet since the SPI flash will not be able to drive 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.

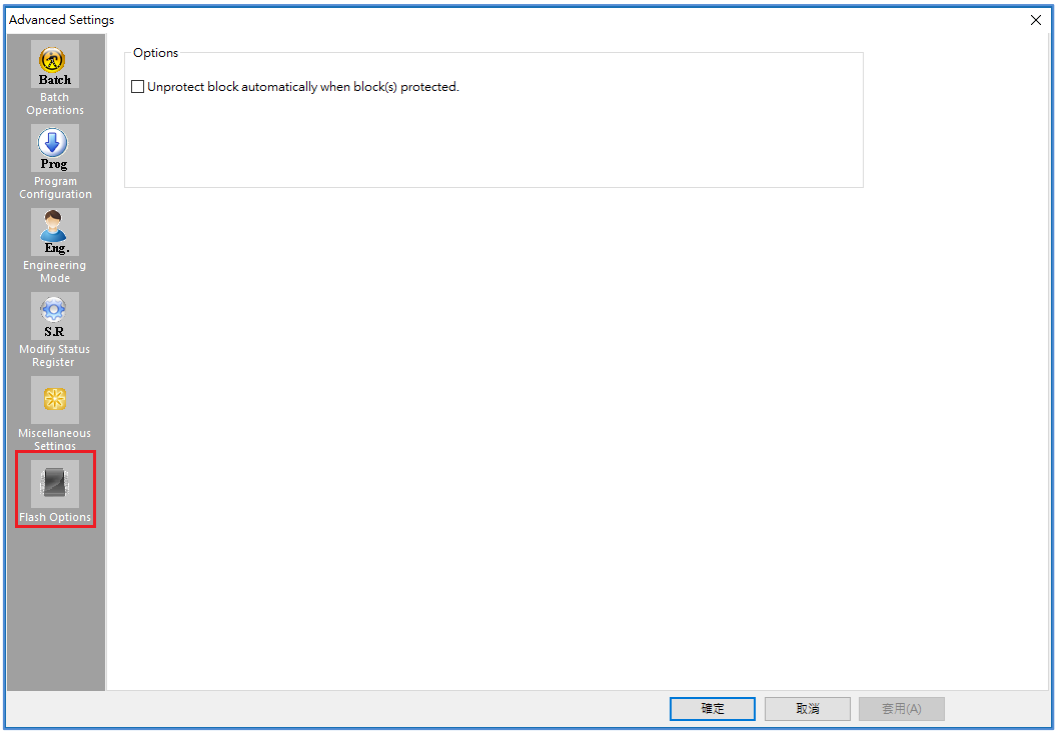
D. Tool Bar ICON Configuration

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.

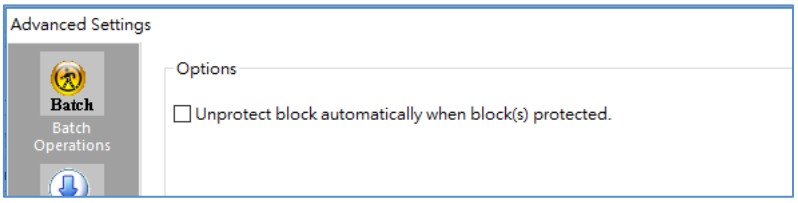




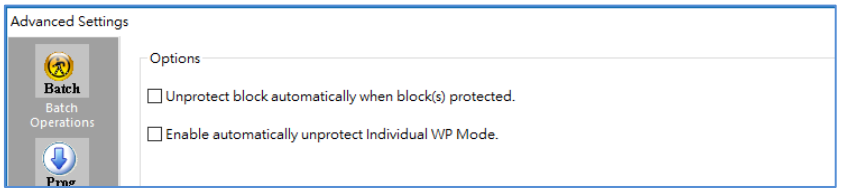
3.5.6 Flash Option



A. Unprotect block automatically when block(s) protected.

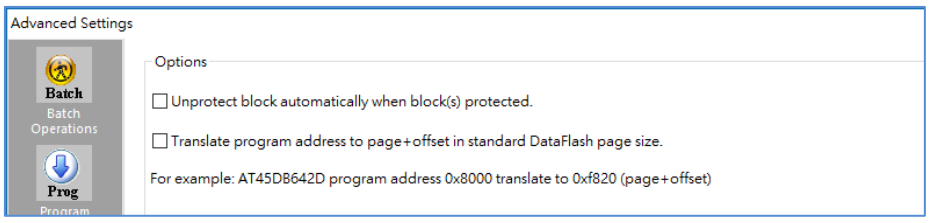


B. Enable automatically unprotect Individual WP mode



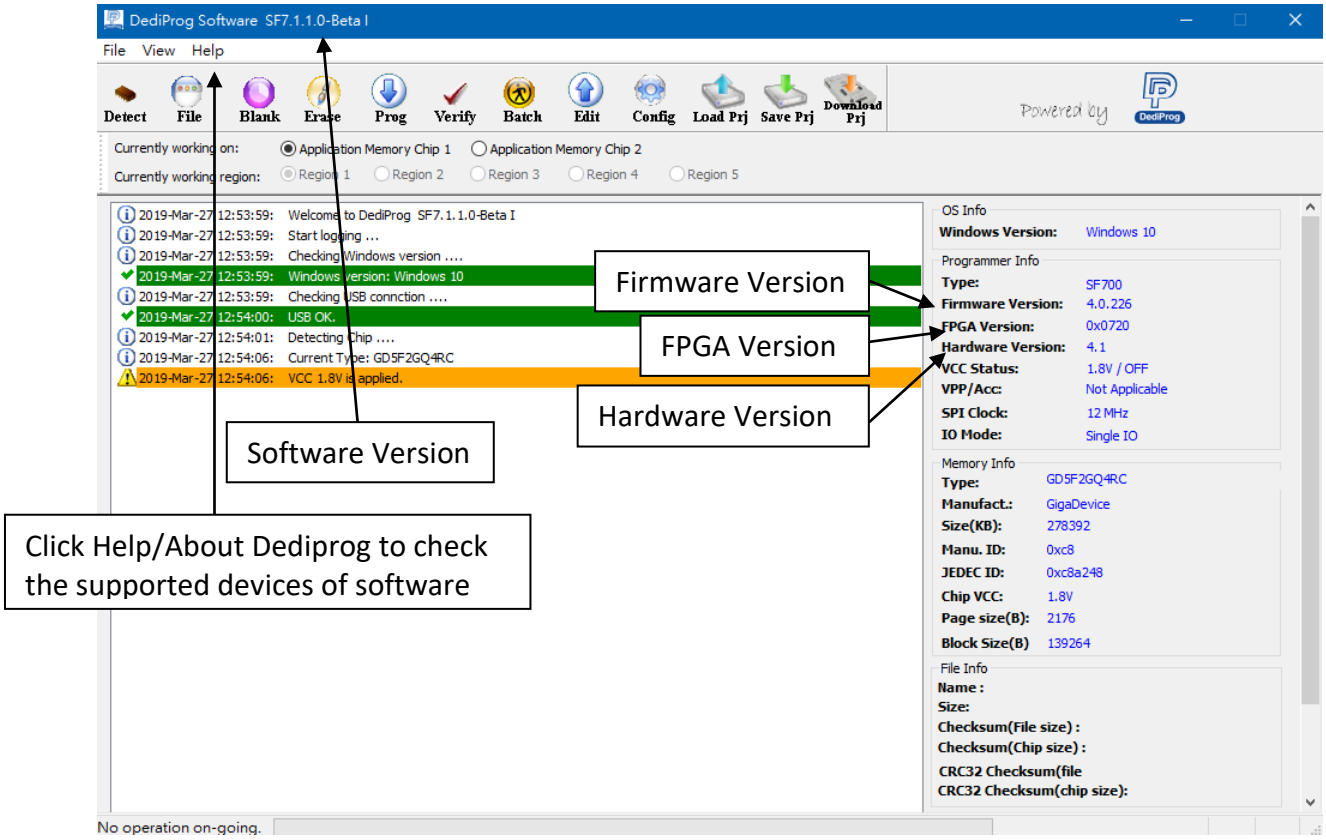
C. Translate program address to page+offset in standard DataFlash page size.

For example: AT45DB642D program address 0x8000 translate to 0xF820 (page+offset)



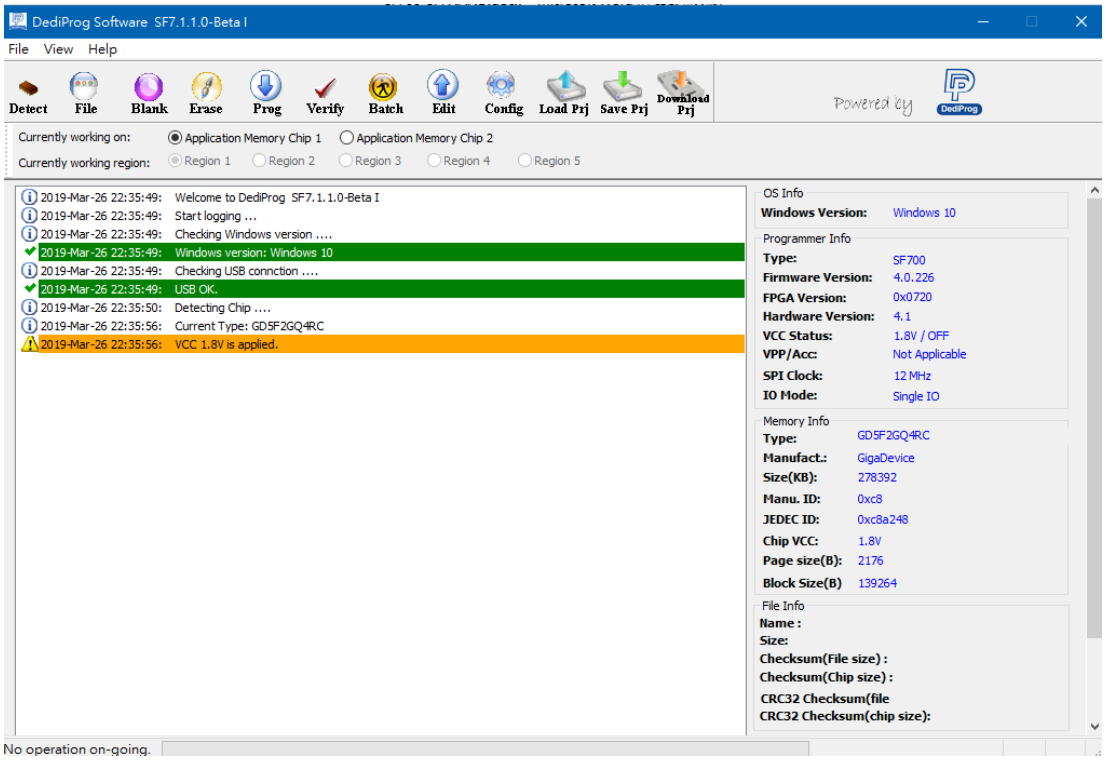
3.6 Supported Devices, Software Version, Firmware Version

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.

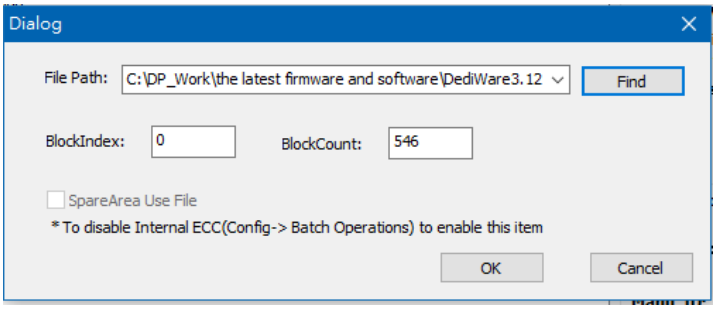


3.7 SPI NAND Programming Interface

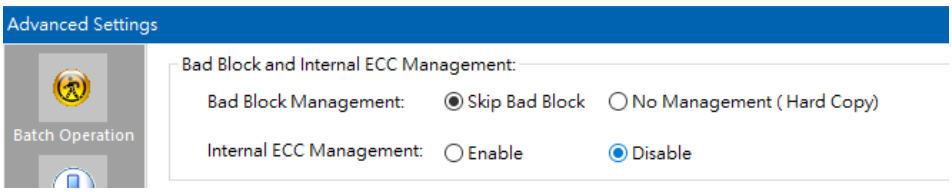
SF700 supports SPI NAND programming. The below image is the software interface, and the operating method is similar to the method for SPI NOR Flash. This section will describe the software functions for SPI NAND.



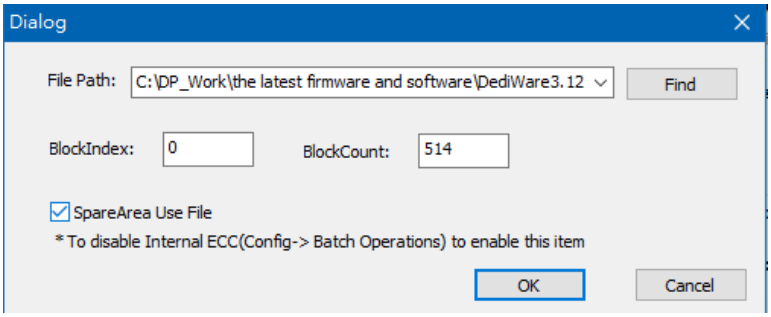
: Load the programming file. Attention: If there is ECC code in the programming project file, which means you need to set up the Spare area, please disable the Internal ECC in the Config setting.



If the image file includes ECC code or has used the Spare area, please follow the below image and select Disable.

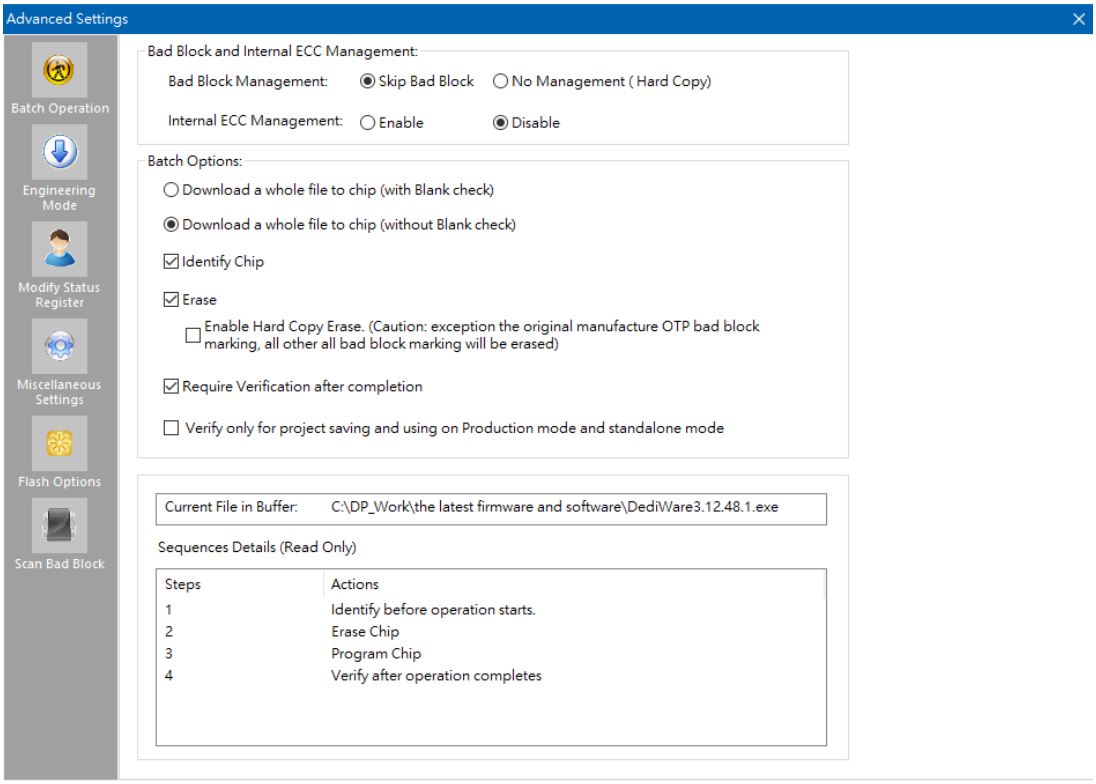


After disable the Internal ECC, click File, and then there will be a SpareArea Use File option to select.



Programming Basic Function: Please refer to Chapter 3.3.

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



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.

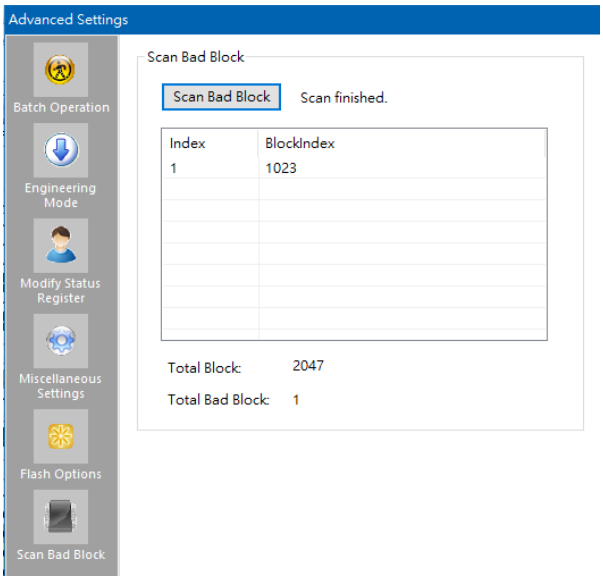
Identify Chip: You can decide whether you want to do ID check or not.

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.

There is a Scan bad block function in the Config setting that can read the Bad block distribution in NAND, which is very beneficial to NAND analysis.



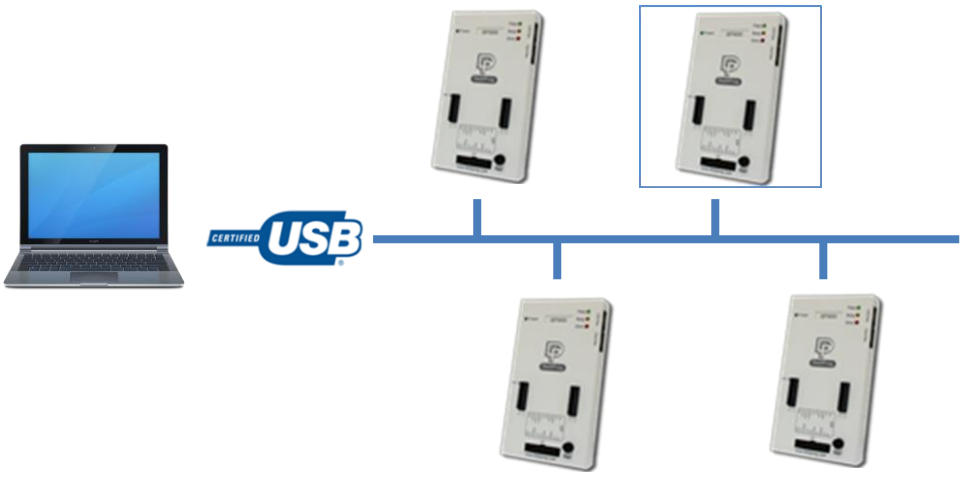
IV. DediProg SF Software Production GUI

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/SF600*Plus/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.

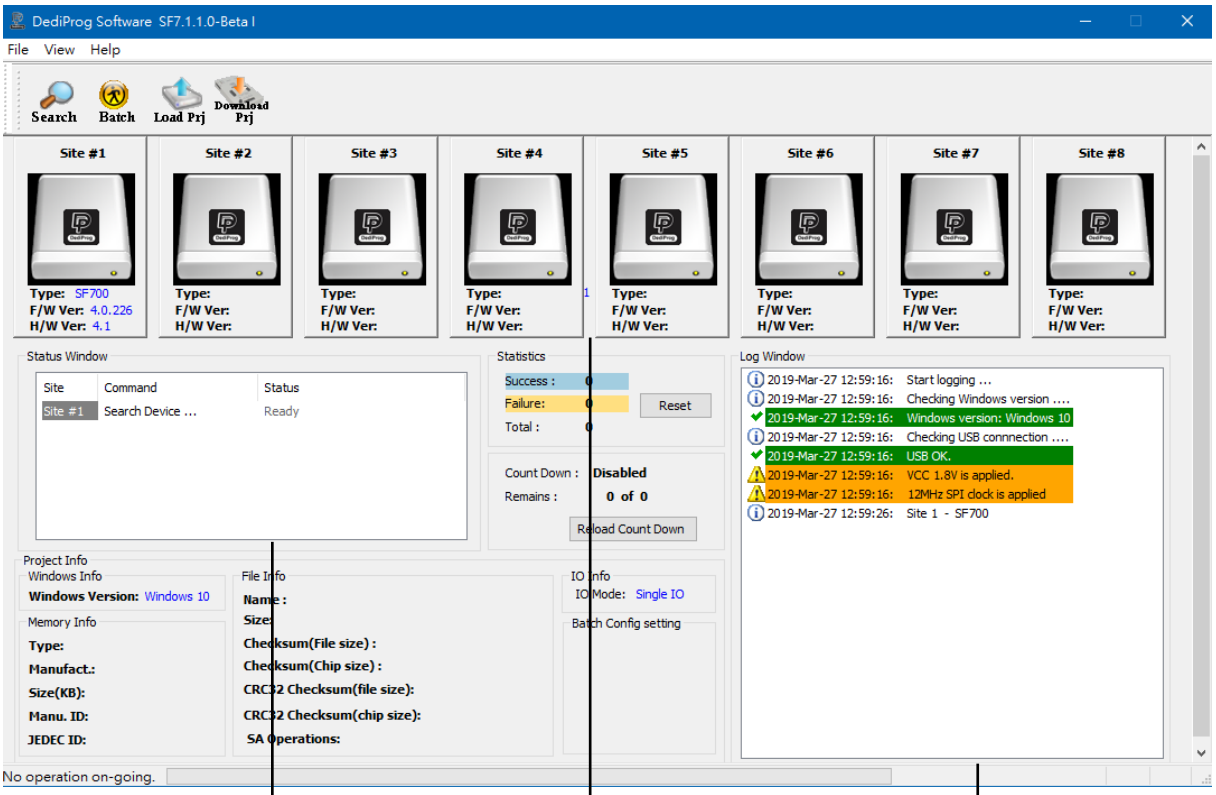
Multi-Programmers Capability for SF series programmers



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.



Status window

Programmer Site Status bar

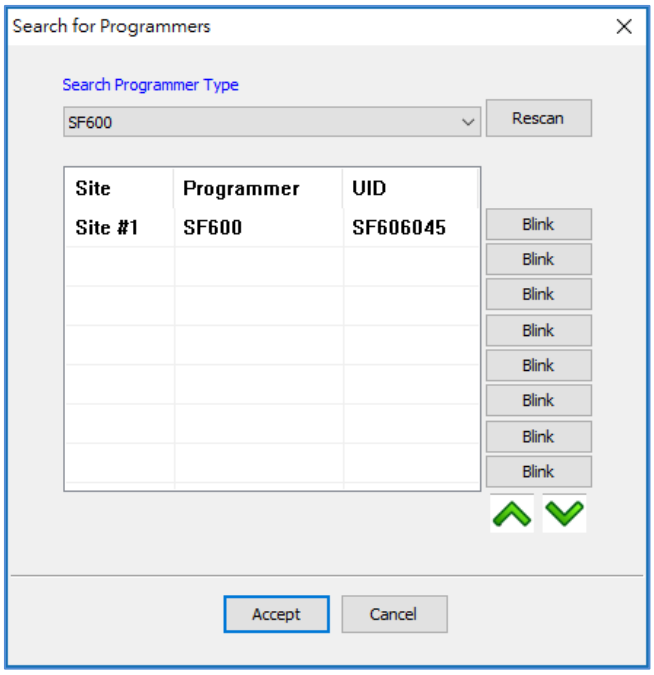
Log Window

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.



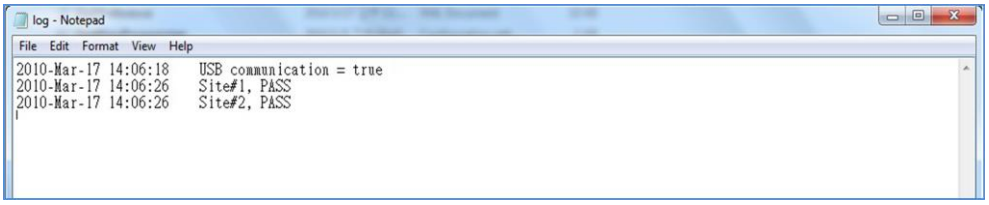
Note: SF software doesn’t support different programmer at the same time, and only supports same programmer on the production mode.

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\SF100



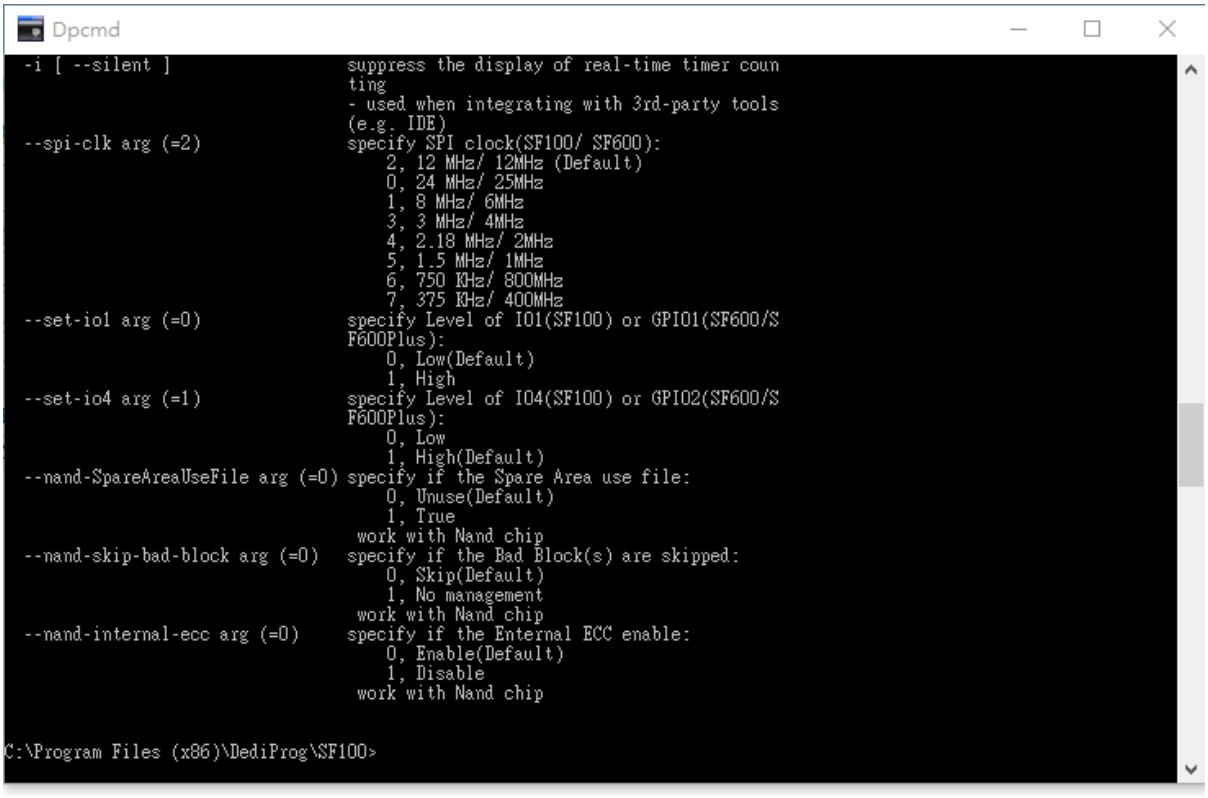
This .txt file has to be checked to make sure that the operation has been successful. Time stamp can also be checked to be sure that the result has been updated with a new value.

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 Unknown

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

Window DOS command


Dpcmd
-v [--verify] verify checksum file and chip
- works with --prog/auto/batch/load-file/addr only
-x [--fill] arg (=FF) fill spare space with an hex value(e.g.FF),
- works with --prog/batch only
--type arg Specify a type to override auto detection
- use --list arguement to look up supported type.
--lock-length arg hexadecimal length of area that will be kept unchanged while updating
- used along with --auto/lock-start only.
Example:
dpcmd -u file.bin --lock-start 0x1000 --lock-length 0x100 -v
--lock-start arg hexadecimal starting address(e.g. 0x1000),
- must work with --lock-length
- defaults to 0, if omitted.
--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.
--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.
--load-file arg Load a bin/hex/s19 file and compare with memory content
- work with --verify only
Example:
dpcmd --verify --load-file d:\xxx.bin
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
--vcc arg specify vcc
0, 3.5V
1, 2.5V
2, 1.8V
1800 ~ 3800, 1.8 ~ 3.8V (minimum step 100mV) (For SF600/ SF600Plus only)
--vpp apply vpp when the memory chip supports it
- work with --prog and --erase.
--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%\dediprog\SF100\log.txt"



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\SF100
3. Type in the following command to auto directs the dpcmd command to the “dpcmd.exe” location.

Set path=%path%;”c:\program files\dediprog\SF100

5.3 Basic Usages

1. dpcmd –r "f:\file.bin",
   reads the chip and save it into a file "file.bin" in Partition f
2. dpcmd –r STDOUT –a 0x100 -l 0x23,
   reads 0x23 bytes starting from 0x100 and display it on the screen
3. dpcmd –u f:\file.bin,
   erases and then program file.bin in Partition f into the serial flash
4. dpcmd –p f:\file.bin –a 0x100,
   writes file.bin in Partition f into the serial flash starting from address 0x100
5. dpcmd –p f:\file.bin –x 0xaa,

programs file.bin in Partition f into the serial flash and fill the rest area with 0xaa

6. Able to open multiple DpCmd windows to control different programmers.

Remarks: -a only works with -p, -r, -s, -v, -u, -z
Remarks: -a with -l only works with -p, -r, -s, -v, -u,
Remarks: -x only works with -p, -z
Remarks: --load-file only works with –v
Remarks: --lock-start must work with –lock-length each other
Remarks: space is not needed between the switches parameters. E.g. dpcmd -u f:\file.bin
Remarks: default target is chip 1. Please changing the target if need.
Remarks: adding –type will decrease the command execution time.
Remarks: Only “batch” command support EzPort programming.
Remarks: if “–vcc” not be used, detected voltage will be used when operation. It’s possible to use lower voltage to work to cause operation fail. So recommending use “–type” to get work voltage from chip data base.

5.4 Basic Switches

-? [--help]	Show the help message
--list	Print supported chip list
-d [--detect]	detect chip
-b [--blank]	blank check
-e [--erase]	erase entire chip
--force-erase	erase entire chip work with Nand chip only
-r [--read] arg	read chip contents and save to a bin/hex/s19 file -use STDOUT for the console.
-p [--prog] arg	program chip without erase
-u [--auto] arg	automatically run the following sequence: - Read the memory content - Compare the memory content - Erase only the sectors with some differences - Program only the erased sectors with the file data from address 0
-z [--batch] arg	work with SPI NOR and SPI NAND SPI NOR automatically run the following sequence: - check if the chip is blank or not - erase the entire chip(if not blank) - program the entire file starting from address 0 SPI NAND automatically run the following sequence: - check if the chip is blank or not; - erase the chip memory which skip bad block(if not blank); - program a whole file starting from address 0
--nand-batch-forceerase arg	automatically run the following sequence: - check if the chip is blank or not; - force erase the entire chip(if not blank); - program a whole file starting from address 0
-s [--sum]	display chip content checksum
-f [--fsum] arg	display the file checksum - needs to work with a file

--raw-instruction arg	Issue raw serial flash instructions. - use spaces(" ") to delimit bytes. - instructions must be enclosed in double quotation marks("") - use "|" to send continuous command Example: dpcmd --raw-instruction 06 dpcmd --raw-instruction "06|02 00 00 00 11 22 33”
--raw-require-return arg (=0)	decimal bytes of result to return in decimal after issuing raw instructions. - Used along with --raw-instruction only. Example: dpcmd --raw-instruction "03 FF 00 12" --raw-require-return 1 dpcmd --raw-instruction "06|05" --raw-require-return “0|2”

5.5 Optional Switches

(Specify the following switches to change default values):

-a [--addr] arg	hexadecimal starting address hexadecimal (e.g. 0x1000), - works with --prog/read/sum/auto/batch only - defaults to 0, if omitted.
-l [--length] arg	hexadecimal length to read/program in bytes, - works with --prog/read/sum/auto only - defaults to the entire file if omitted
-v [--verify]	verify checksum file and chip - works with --prog/auto/load-file only
-x [--fill] arg (=FF)	fill spare space with an hex value(e.g. FF), - works with --prog/batch only
--type arg	Specify a type to override auto detection - Use –list argument to look up supported type.
--lock-start arg	hexadecimal starting address(e.g. 0x1000), - must work with --lock-length - defaults to 0, if omitted.
--lock-length arg	hexadecimal length of area will kept unchanged while updating - Used along with --auto/lock-start only. Example: dpcmd -u file.bin --lock-start 0x1000 --lock-length 0x100 -v
--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.

--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.
--load-file arg	Load a bin/hex/s19 file and compare with memory content - work with --verify only Example: dpcmd --verify --load-file d:\xxx.bin

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
--vcc arg (=0)	specify vcc 0, 3.5V 1, 2.5V 2, 1.8V 1800 ~ 3800, 1.8 ~ 3.8V (minimum step 100mV) (For SF600/SF600*Plus* only)
--vpp	apply vpp when the memory chip supports it - work with --prog and --erase.
--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\SF100\log.txt"

-i [--silent]	suppress the display of real-time timer counting - used when integrating with 3rd-party tools (e.g. IDE)
--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
--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)
--nand-SpareAreaUseFile arg (=0)	specify if the Spare Area use file: 0, Unuse(Default) 1, True work with Nand chip
--nand-skip-bad-block arg (=0)	specify if the Bad Block(s) are skipped: 0, Skip(Default) 1, No management work with Nand chip
--nand-internal-ecc arg (=0)	specify if the internal ECC enable: 0, Enable(Default) 1, Disable work with Nand chip

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_OTHERS=99,
};
```

VI. Standalone Mode (SF600*Plus*/SF700 only)

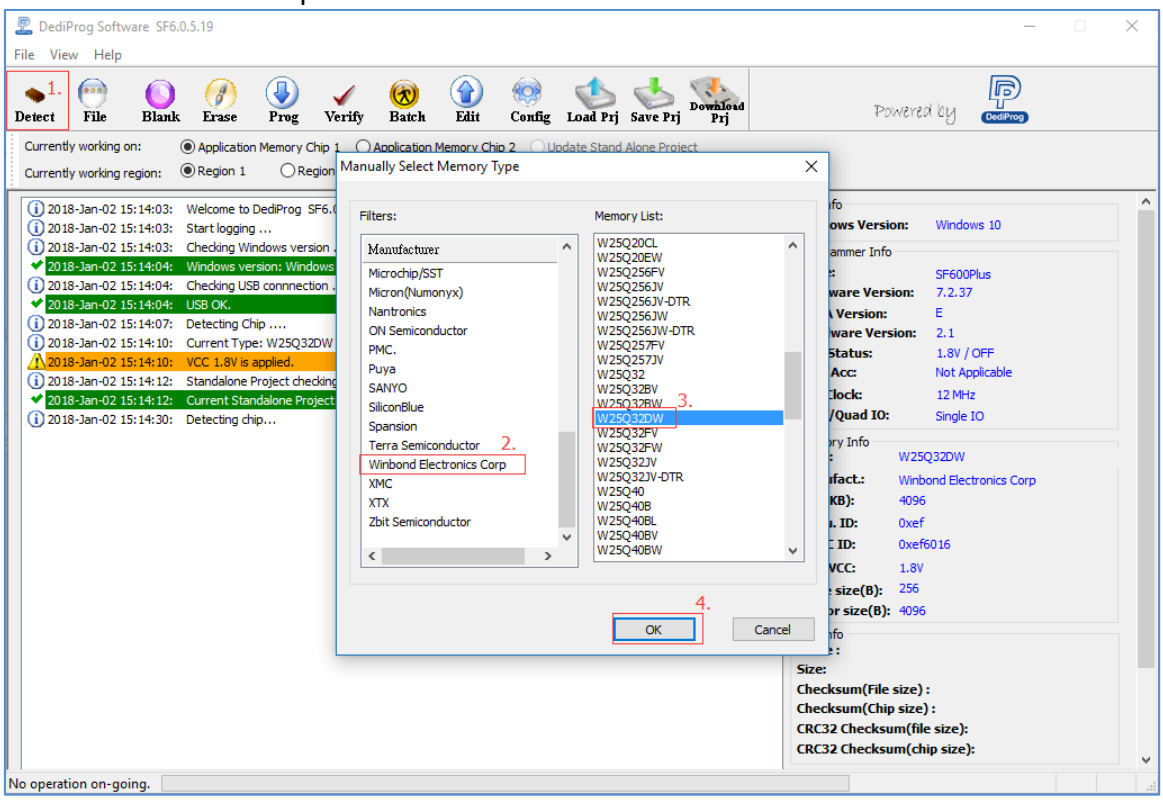
In addition to the functions provided by SF600*Plus*/SF700 further allow you to download project to SF600*Plus*/SF700 directly and program serial flash memories in standalone mode.

6.1 Project Preparation

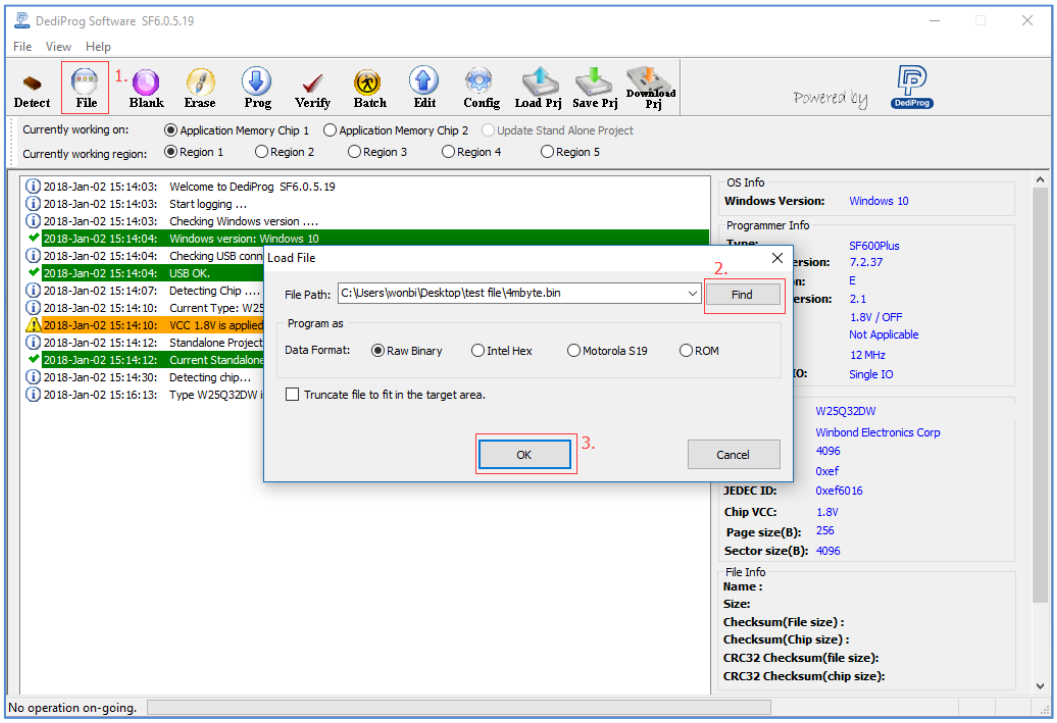
Prepare a standalone programming project.

6.1.1 Open DediProg Engineer software.

6.1.2 Select IC brand and part number.



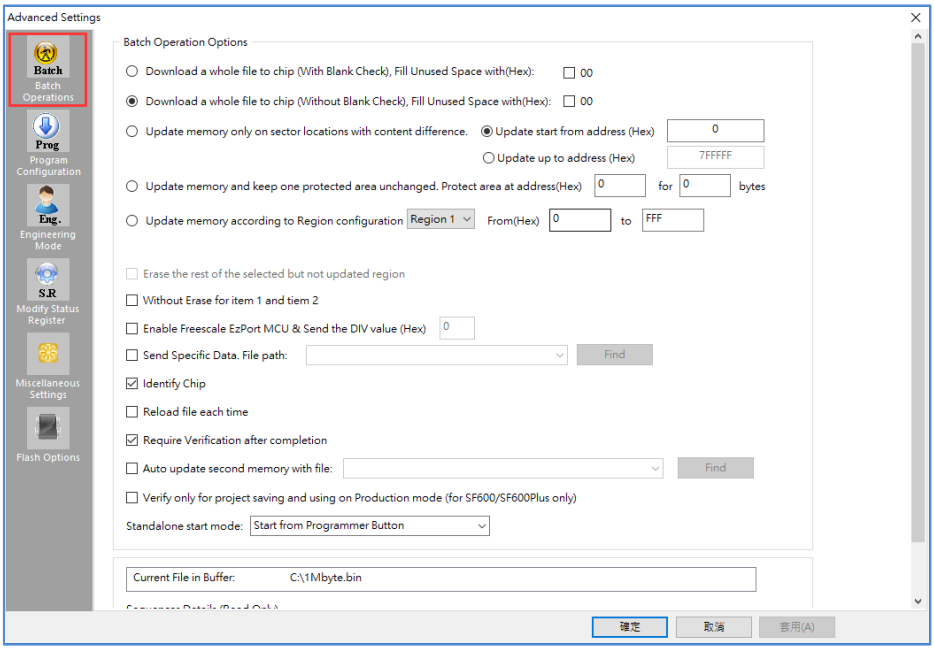
6.1.3 Load the programing file.



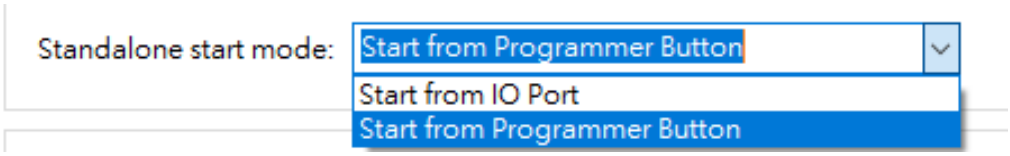
6.1.4 Click “Config” icon to set programming flow.

Important Notice:

“Identify Chip” is necessary for SF600*plus*/SF700 standalone programming. Make sure to select “Identify Chip” in programming flow.

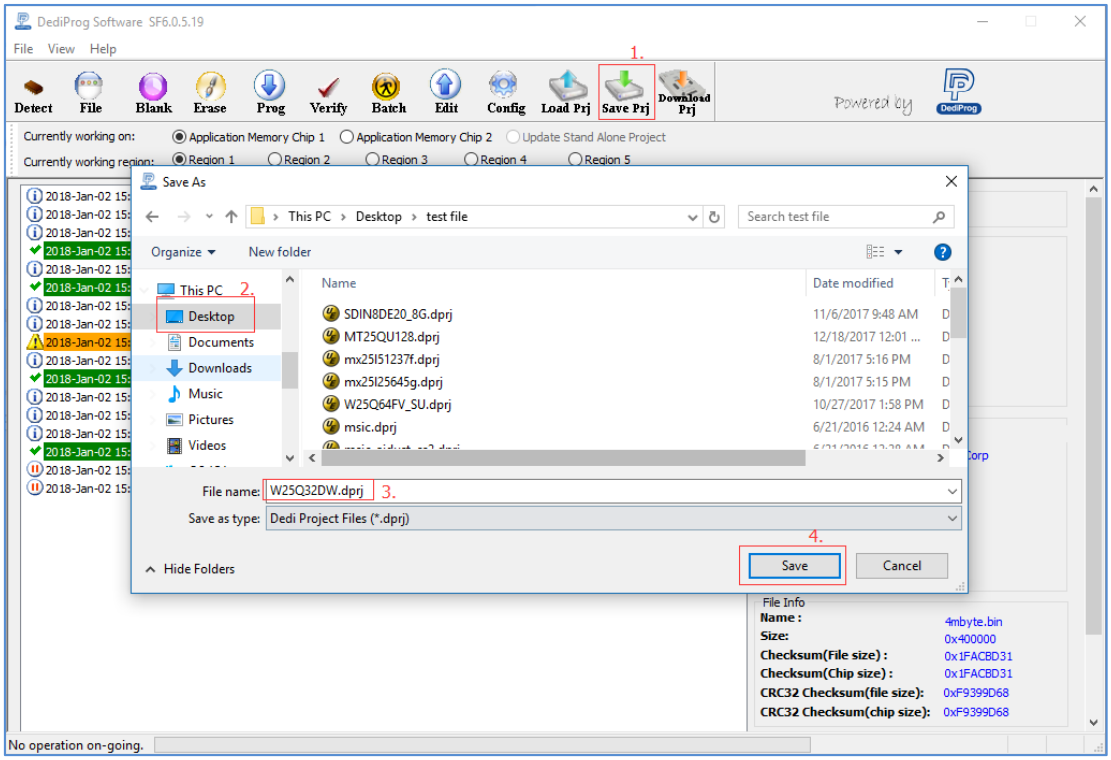


6.1.5 Choosing Standalone start mode

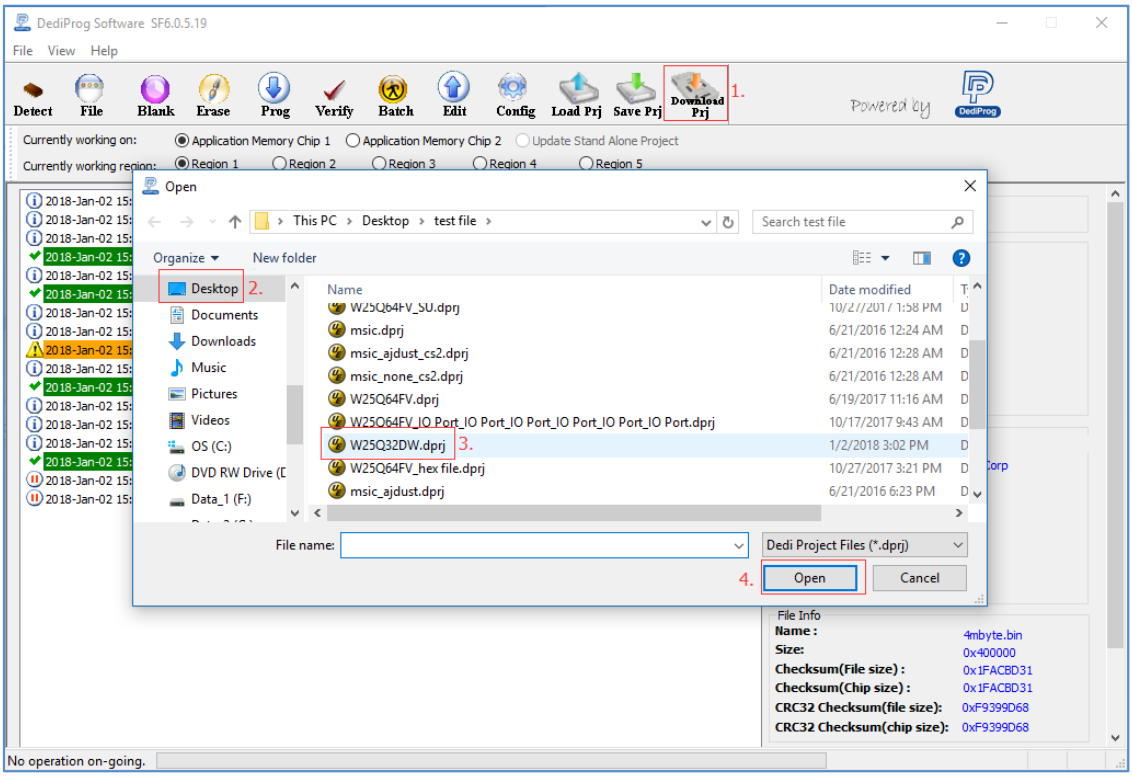


Note : SF700 only supports Start from Programmer Button.

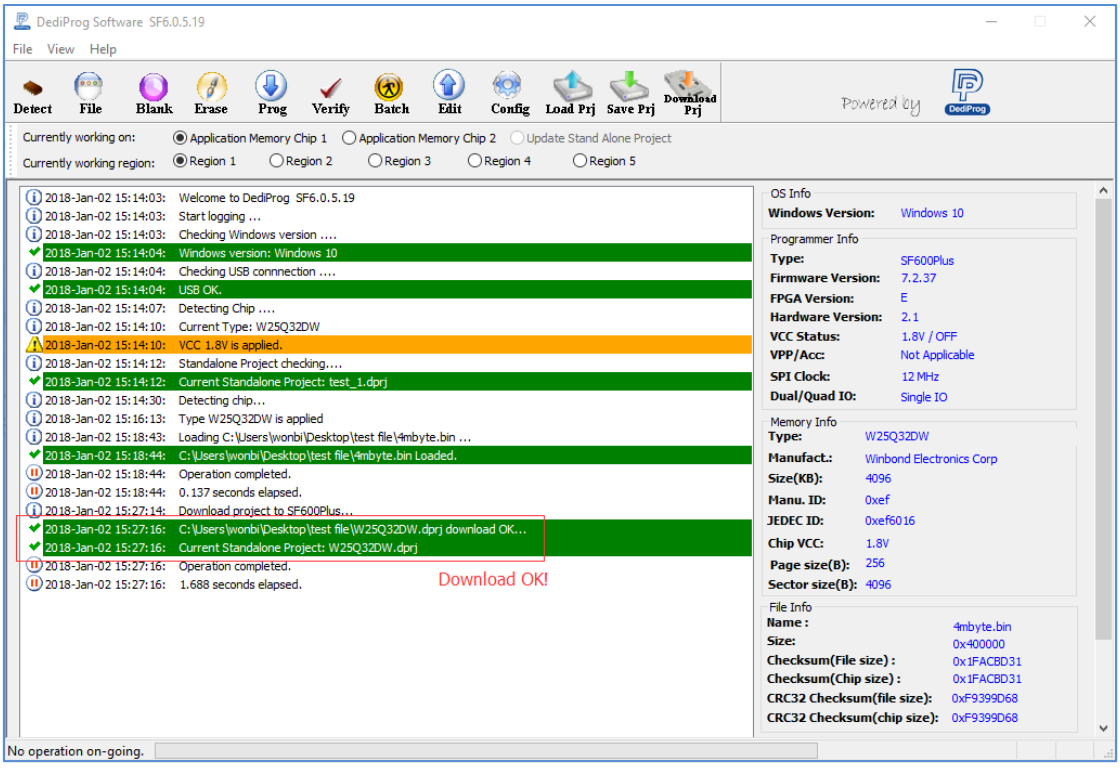
6.1.6 Save .dprj file to PC



6.1.7 Press "**Download Prj**" button to download project to SF600*Plus*/SF700 embedded memory



6.1.8 Download project successfully



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 COM Port"** mode

The Com Port design is for integrating SF600/SF600*Plus* 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), SF600/SF600*Plus* 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 SF100/SF600/SF600*Plus*.

If you are using Windows 8.1/Windows 10, 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

SF100

Windows OS	Current Firmware Version	Upgrade Firmware	Upgrade Software
Win8.1/Win10	6.x.xx	6.5.03	SF 6.0.5.19
	5.x.xx	5.5.03	SF 6.0.5.19
	1.x.x to 4.x.x	Please contact DediProg sales	
Older versions	5.x.xx and later	5.5.xx	SF 6.0.5.19
	1.x.x to 4.x.x	There are no restriction	

SF600 / SF600*Plus*

Windows OS	Current Firmware Version	Upgrade Firmware	Upgrade Software
Win8.1/Win10	6.x.x	6.9.0	SF 6.0.5.19
	7.x.x	Latest firmware version (Please contact DediProg sales)	SF 6.0.5.19
Before Win 8.1	6.x.x	earlier than 6.9.0	There are no restriction

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

VIII. Revision History

Date	Version	Changes
2010/03/19	5.5	Added: Enable EzPort Function on Configuration; log.txt file available on Commend line; Blink/Device/Fix-Device on Dpcmd.
2010/04/14	5.6	Added: Update up to address option on Batch and Program Configuration operation options.
2010/05/10	5.7	Minor improvement
2011/05/18	5.8	1. Added specific function. 2. Added region configuration programming function.
2011/08/26	5.9	Added SF600 Hold pin status setting method.
2012/01/09	6.0	Added SF600 standalone programming.
2012/12/20	6.1	Revise the CLI detail and add exit codes.
2013/08/23	6.2	1. Added status register-2 function 2. Added the multiple-Dpcmd function.
2013/12/18	6.3	1. Remove part of SF200 and SF300 2. Remove “isolation free” from software
2014/02/25	6.4	New feature for SF600*Plus*
2014/04/28	6.5	Replenish COM Port feature of Stand Alone mode
2014/05/20	6.6	Modify log saving command
2014/06/04	6.7	1. Add –load-file command for “verify only” feature 2. Updated case study contents and testing time.
2014/08/01	6.8	Added IO1/IO4(SF100) and GPIO1/GPIO2(SF600/SF600*Plus* setting)
2014/10/28	6.9	Added Chapter VIII. Firmware Support for Microsoft Windows
2015/04/28	7.0	1. Added Chapter 3.5.3.2 Send multiple commands 2. –raw-instruction arg 3. use “|” to send continuous command 4. Modified Chapter 3.5.3 Engineering Mode- send command function on Engineering mode 5. Added scroll bar for the software window 6. Updated Chapter VIII. Firmware version
2016/03/01	7.1	1. Remove Dual and Quad IO function

2016/7/12	7.2	1. Modify the USB driver installation content 2. Modify Edit content about Go button 3. Add multiple DpCmd window function 4. Update firmware version and software version
2016/10/14	7.3	Update the latest software version.
2016/12/22	7.4	EzPort programming support on DpCmd (--batch command).
2017/06/10	7.5	1. Add DpCmd Remark content 2. Update Engineering GUI pictures (Logo changed, Programmer and Memory Info display more information)
2018/01/3	7.6	1. Add DpCmd Remark content. 2. Add “verify only for project saving and using on Production mode and Standalone mode” feature. 3. Modify region programming including feature which rest of selected region is erased. 4. Delete icons on production mode.
2019/01/2	7.7	1. Add SF700 information 2. Modify the picture of engineering mode in Ch.3.5.3. 3. Modify picture of standalone start mode.
2019/3/27	7.8	1. Add SPI NAND information in Ch.3.7. 2. Update Command line information in Ch.5

DediProg Technology Co., Ltd

Taiwan Headquarter TEL: 886-2-2790-7932 FAX: 886-2-2790-7916
4F., No.7, Ln. 143, Xinming Rd., Neihu Dist., Taipei City 114, Taiwan

China Office TEL: 86-21-5160-0157
Room 518, Building 66, Lane1333, Xinlong Road, Vanke Hongqiao CBD.Min Hang District, Shanghai, P.R.C. 201101

U. S. Office TEL: 1-909-274-8860
209 E Baseline RD, Suite E208 #8, Tempe, AZ, 85283, USA

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