www.dediprog.com

1

DediProg SF Software User Manual

V6.1

DediProg Technology Co. LTD 4F., No.7, Ln. 143, Xinming Rd., Neihu Dist., Taipei City 114, Taiwan Email for technical support: support@dediprog.com Email for Sale information: sales@dediprog.com www.DediProg.com

I.	Introduction	4
II.	Software Installation Guide	4
A.	Operating System Requirement	4
B.	USB Installation	4
III.	Dediprog SF Software Engineering GUI	8
A.	Prepare the Environment	
B.	Identify the Target SPI Flash	
C.	Tool Bar Description	9
D.	Edit Window Description	11
E.	Configuration Window Description	12
1	. Batch Operation Option	
2	. Program Configurations	17
3	. Engineering Mode	17
4	. Modify Status Register	
5	. Miscellaneous Settings	
	a) vpp Option b) Vcc Option	
	c) SPI Clock Setting	
	d) Tool Bar ICON Configuration	
F.	Supported Devices, Software Version, Firmware Version	22
TT I	Deding of Software Production CIII	22
1	Sourch and Sologt	<i>43</i>
F	R Batch Config	
Ċ	Single Site programming	
V D	adinrog Windows Command Lina	27
v . D	Jutro duction	····· <i>41</i>
A. D		
B.	How to Start	
C.	Basic Usages	
D.	Basic Switches	
E.	Optional Switches	
F.	Exit Code	34
VI. S	tand Alone Mode (SF300 & SF600)	
A.	SF300 Standalone Programming:	35
1	. Project Preparation	
2	. Stand Alone Mode	
3	. Standalone Programmer Operations	
4 D	. SF300 Hardware Description	
В.	SF600 Standalone Programming:	39
1	. Project preparation	40
2	a) Standalone programming with I CD Keynad	
	b) Standalone programming with LCD Keypad.	
	c) Start Standalone programming.	
3	. SF600 Hardware Description	45
VII.	Specific Functions (SF600)	
• • • • • •	Dual/Ouad IO:	
Ē	B. Isolation Free	47
C	C. Hold Pin Status Setting	
VIII	Revision History	49

Important Notice:

This document is provided as a guide line 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 device connected when using this software can be used together with SF100, SF200, SF300, SF600 and Backup Boot Flash kit. To get more information on our DediProg products and how to use them, please refer to our products specification, presentation and application notes in our web site: www.DediProg.com

II. Software Installation Guide

A. Operating System Requirement

Windows 7 Windows Server® 2008 Windows Vista® Windows XP Support both 32 bit or 64bit OS

B. USB Installation

- 1. Insert the installation CD or download the installation software from **www.dediprog.com**
- 2. Execute SFx.x.x.msi file and click next until the installation is finished.
- 3. After step 2, plug in the USB cable to your computer and Dediprog programmer, A Hardware wizard will show up as follow.

Found New Hardware Wiz	ard
	Welcome to the Found New Hardware Wizard Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission). Read our privacy policy
	Can Windows connect to Windows Update to search for software? Yes, this time only Yes, now and every time I connect a device
	Click Next to continue.
	< <u>B</u> ack <u>N</u> ext > Cancel

www.dediprog.com

Follow the steps showed in the snap shots below to complete the installation.

1. Win XP operation system:

Found New Hardware Wizard	Found New Hardware Wizard
Welcome to the Found New Hardware Wizard	Please choose your search and installation options.
This wizard helps you install software for:	Search for the best driver in these locations.
DediProg SF Programmer driver	Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.
If your hardware came with an installation CD or floppy disk, insert it now.	Search removable media (floppy, CD-ROM) Include this location in the search: C:\Program Files\Dediprog\SF100\USB Driver\Win Browse
What do you want the wizard to do?	Browse For Folder
Install the software automatically (Recommended)	Select the folder that contains drivers for your hardware. Windows does not guarantee that ware.
Click Next to continue.	E C Adobe
	Avira
<back next=""> Cancel</back>	
	E C USB Driver
	4
	WindSB Driver
	To day we will be a first to be a
	OK Cancel
Found New Handmars Ultrand	
Please select the best match for your bardware from the list below	Please wait while the wizard installs the software
	Hardware Installation
- Dadinrog Envilator driver	The software you are installing for this hardware:
	DediProg SF Programmer driver
Description Version Manufacturer Location	t has not passed Windows Logo testing to verify its compatibility
Dedipting USB driver Unknown Dedipting Inc. c:\windows\u00e3	with Windows XP. (Tell me why this testing is important.)
7	Continuing your installation of this software may impair or destabilize the correct operation of your system
4Y	either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.
A This driver is not diaitally signed	9
I ell me why driver signing is important 8	Continue Anyway STOP Installation
Kack Next > Cancel	< Back Next > Cancel

2. Win Vista & 7 Operation Systems:



If the USB driver is installed properly, users can find "Dediprog SF Programmer driver" under device manager when the programmer is plugged into the PC.

🖳 Device Manager	
File Action View Help	
← → 10 4 12 10 2	
🕀 🦇 Disk drives	
庄 🧕 Display adapters	
🗄 🎱 DVD/CD-ROM drives	
🗄 🗃 Floppy disk controllers	
🗄 🔜 Floppy disk drives	
🗄 🗃 IDE ATA/ATAPI controllers	
🗄 🐷 Keyboards	
🗄 🐃 Mice and other pointing devices	
🗄 👰 Monitors	
🗄 🕮 Network adapters	
🗄 🔷 NVIDIA Network Bus Enumerator	
🗄 🎐 Ports (COM & LPT)	
🗄 📾 Processors	
😟 🧐 Sound, video and game controllers	
🗄 🧕 System devices	
E 🖨 Universal Serial Bus controllers	
DediProg SF Programmer driver	
Standard Enhanced PCI to USB Host Controller	
Standard OpenHCD USB Host Controller	
USB Root Hub	
USB Root Hub	
· Ma - 45.55 200.55.5 - 200	

If no, please check "Other devices". After selecting and installing, it will show up correctly under USB controllers.



III. Dediprog SF Software Engineering GUI

Dediprog SF software is used together with SF100, SF200, SF300, SF600 or Backup Boot Flash Kit. The software can be used to program serial flash memory as well as the downloading configuration contents to the reference SPI Flash embedded in SF300 and SF600's Micro SD card for standalone programming purpose. After the software and USB driver are installed, please follow the following steps before running the software.

After the software installation, there will be three software icons on your desktop. Icon "Dediprog Engineering" is for the engineering GUI, Icon "Dediprog Production" is for the production GUI, and Icon "DPCMD" is for the command line interface.

A. Prepare the Environment

- 1. Connect the programmer to the PC through a USB cable.
- 2. For ICP programming, connect the ICP cable to the application (please check the specification in case ISP header pin out are not known).

- For off line and stand alone programming, connect the appropriate socket adaptor to the programmer and insert a serial flash in the socket.

3. Double click on the Dediprog software icon on your desktop.



B. Identify the Target SPI Flash

SPI Flash Detection

Double Click on 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 socket. User does not need to select SPI Flash's location.

Note: If user wants 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.

DediProg Software 6.0.0.8	
a View Help	
💊 🦳 🙆 🥐 🚯 🖌 🛞 💮 🚳	
Detect File Blank Erase Prog Verify Batch Edit Config	
rrenty working on: Application Memory Chip 1 Application Memory Chip 2 Update Stand Alone Project 	
rrently working region: @ Region 1	
D 2011 May 16 00:50:15: Welcome to Deditions 6 0.0.9	
2011-May-16 09:59:15: Start logging	F
2011-May-16 09:59:15: Checking USB connection	Powered by Conterior
2011-May-16 09:59:15: USB OK.	Programmer Info
0 2011-May-16 09:59:15: FPAG version: 45	Type: SF600
2011-May-16 09:59:16: 0.967s elapsed to identify chip.	F/W Version: 6.1.9
2011-May-16 09:59:16: Current Type: Pm25LQ032C	VCC Status: 3.5V / OFF
	VPP/Acc: Not Applicable
	SPI Clock: 12 MHz
	Dual/Quad to: Single 10
Application Memory	Memory Info Pm25LO032C
· • • • •	Manufact PMC.
	Size(KB): 2048
	Manu TD: 0x9d
	JEDEC ID: 0x9d7f46
	File Info
	Name -
	nume.
	Size:
	Size: Checksum(File size) :
	Size: Checksum(File size) : Checksum(Chip size) :
	Size: Checksum(File size) : Checksum(Chip size) :
	Batch Config setting

C. Tool Bar Description

The tool bar provides all SPI Flash operations.

View Help								
• 💮 Detect File	O Blank	() Erase	Prog	V erify	🛞 Batch	edit	(c) Config	

Detect

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

-liters:	Memory List:
Manufacturer	M25P32
<auto detected="" type(s)=""></auto>	
<all></all>	
AMIC	
Atmel	
cFeon/EON	
ESMT	
GigaDevice	
Macropix	
Numonyx	
PMC	
SANYO	
Spansion	
SST	
in the second second	
/	
	OK Cancel
	OK Cancel

www.dediprog.com



File

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

File Path:				Find
Program as —				
Data Format:	Raw Binary	C Intel Hex	C Motorola 519	C ROM

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 be changed until it is re-configured.

Edit

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

Config

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

D. Edit Window Description

SPI Flash content display:

In the edit window, file contents and chip contents can be displayed in the same time so that user can make the comparison. By default the selected file contents are displayed once the user enters into the edit window.

The user can click on "Open" if another file contents are to be shown. The user can click on "Read" in order to read the chip contents are display them on the edit window as well. Checksum of file contents and chip contents are displayed.

	Source																			-			_										
Chip: Pm23LQ032C Read Read Chip Buffer to File Swap Byte Swap Word Swap DWord Swap DWord Swap State Sta	File: (C:\Use	rs\De	an\D	eskto	p∖ran	dom\	8R.bi	n											8	Ope	n			-			File I	Buffe	to Fi	ile		
Swap Byte Swap Word Swap DWord Swap DWord Swap Buffered File Stress Stres Stress	thio: F	Pm25L	0032	с																ſ	Rea	d			T			Chip	Buffe	r to F	ile		
Swap Byte Swap Word Swap DWord Swap Checksum: File = 07F67D7A Memory = 17E67D7A Buffered File = 07F67D7A Buffered Memory = 17E67D7A Show In	eren e			<u>.</u>																								- 54		_			
Byte Swap Word Swap Word Swap Checksum: File 07F67D7A Memory = 17E67D7A Buffered File = 07F67D7A Buffered File = 07F67D7A Buffered File = 07F67D7A Buffered File = 07F67D7A Buffered File 0 1 42 43 44 5 6 7 7 8 9 4A 4B 4C 7D 7E	wap	_	24112-01	_								_			i																		
Checksum: File 07F67D7A Memory = 17E67D7A Buffered File = 07F67D7A Buffered Memory = 17E67D7A Buffered Memory = 17E67D7A Buffered Memory = 17E67D7A Memory = 17E67D7A <t< th=""><th></th><th>Byte</th><th>e Swa</th><th>P</th><th></th><th></th><th>Wo</th><th>rd Sw</th><th>ap</th><th></th><th></th><th>DW</th><th>lord S</th><th>Swap</th><th>ļ</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></t<>		Byte	e Swa	P			Wo	rd Sw	ap			DW	lord S	Swap	ļ																		
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +1 +2 +3 +4 +5 +6 7 *8 +9 +A +B +C +D +1 +2 *3 *4 *5 67 74 8 +6 +7 +8 +9 +A +B +C +D +1 *2 *3 *4 *5 67 74 8 +9 *A +B +C +D +1 *2 *3 *4 *5 *6 7 *8 *9 *A *8 *6 *7 *8 *9 *A *8 *6 *6 *6 *6 *6 *6 *6 *6 *6 *6 *6 *6 *6 *6 *6	hecksum:	File	= 07	F670	7A	Men	nory	= 17	7E67	D7A	Buff	fered	File	= 07	F67	D7A	Buffe	red Me	emoi	γ=	17E6	7D7/	A					Sł	now II	n (<u>о</u> н	ex () AS
Address +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D 0x000000 15 15 D D 1 95 D 2 B A 10 70 26 13 8 E 5 D E 13 8 10 70 26 13 8 P P P P P P P P P P P P P P P P P P P P P									ĝ	File												Me	emory	i.				193		117 DE		0.540.5	
0x00000 61 E4 F2 28 CA 7A 67 64 2E 80 BC 4F 06 4C D4 0E 61 E4 F2 28 CA 7A 67 64 2E 80 BC 4F 06 4C 0x00000 15 8F F1 AF 55 DD 1 95 06 28 7E 6C 22 49 F5 8F 1A 50 28 46 10 70 26 13 84 E0 53 CO DC 80 16 14 92 87 44 98 20 57 43 61 A4 13 EF 11 12 12 18 A4 16 D D 7A 46 FA 28 74 97 27 12 18 44 16 D D D D D D D </th <th>Idress</th> <th>+0</th> <th>+1</th> <th>+2</th> <th>+3</th> <th>+4</th> <th>+5</th> <th>+6</th> <th>+7</th> <th>+8</th> <th>+9</th> <th>+A</th> <th>+B</th> <th>+C</th> <th>+D</th> <th>+E</th> <th>+F</th> <th>+0</th> <th>+1</th> <th>+2</th> <th>+3</th> <th>+4</th> <th>+5</th> <th>+6</th> <th>+7</th> <th>+8</th> <th>+9</th> <th>+A</th> <th>+B</th> <th>+C</th> <th>+D</th> <th>+E</th> <th>+F</th>	Idress	+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
DX00000 F F I AF 55 D E 1 F 5 F I AF 55 D E 1 F S F I AF 55 D E 1 AF 55 D E 1 AF 55 D E 1 A 5 D E 1 A 5 D C D C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C <thc< th=""> C <thc< th=""></thc<></thc<>	000000	61	E4	F2	28	CA	7A	67	64	2E	80	BC	4F	06	4C	D4	OE	61	E4	F2	28	CA	7A	67	64	2E	80	BC	4F	06	4C	D4	OE
DX00000 18 BE 38 D2 B8 46 10 70 26 13 84 E0 53 C0 D 80 15 B 46 10 70 26 13 84 E0 53 C0 D 80 15 D 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16	000010	F5	8F	F1	AF	55	DD	E1	95	D6	2B	DE	88	7E	6C	22	49	F5	8F	F1	AF	55	DD	E1	95	D6	2B	DE	88	7E	6C	22	49
Date Date Vision	000020	18	DE	38	D2	B 8	46	10	70	26	13	84	EO	53	CO	DC	80	18	DE	38	D2	B8	46	10	70	26	13	84	EO	53	CO	DC	80
Day 00000 28 74 9F 2F FF 12 18 A4 D6 D6 DA ND A 46 FA 28 74 9F 2F FF 12 18 A4 D6 D6 DA ND A 46 FA 28 74 9F 2F FF 12 18 A4 D6 D6 DA ND A A6 FA 28 74 9F 2F FF 12 18 A4 D6 D6 DA ND ND <th< td=""><td>000030</td><td>92</td><td>B7</td><td>4A</td><td>09</td><td>82</td><td>0D</td><td>57</td><td>43</td><td>61</td><td>A4</td><td>13</td><td>EF</td><td>B1</td><td>FD</td><td>OF</td><td>14</td><td>92</td><td>B7</td><td>4A</td><td>09</td><td>82</td><td>0D</td><td>57</td><td>43</td><td>61</td><td>A4</td><td>13</td><td>EF</td><td>B1</td><td>FD</td><td>OF</td><td>14</td></th<>	000030	92	B7	4A	09	82	0D	57	43	61	A4	13	EF	B1	FD	OF	14	92	B7	4A	09	82	0D	57	43	61	A4	13	EF	B1	FD	OF	14
Day 000000 ED 90 6F 90 70 70 70 70 70 70	000040	2B	74	9F	2F	FF	12	18	A4	D6	D6	DA	8D	DA	7A	46	FA	2B	74	9F	2F	FF	12	18	A4	D6	D6	DA	8D	DA	7A	46	FA
0x000000 98 C9 8C 81 19 F2 8A 33 73 D8 18 CD 15 8A E8 F8 98 C9 8C 81 19 F2 8A 33 73 D8 18 CD 15 8A E8 F8 98 C9 8C 81 19 F2 8A 33 73 D8 18 CD 15 8A E8 F8 98 C9 8C 81 19 F2 8A 33 73 D8 18 CD 15 8A E8 F8 20 C8 81 64 63 D C1 F8 A4 C1 C5 C1 C1 C1 </td <td>000050</td> <td>ED</td> <td>90</td> <td>6F</td> <td>D9</td> <td>C4</td> <td>DC</td> <td>BC</td> <td>92</td> <td>DD</td> <td>B4</td> <td>05</td> <td>38</td> <td>C5</td> <td>09</td> <td>15</td> <td>D0</td> <td>ED</td> <td>90</td> <td>6F</td> <td>D9</td> <td>C4</td> <td>DC</td> <td>BC</td> <td>92</td> <td>DD</td> <td>B4</td> <td>05</td> <td>38</td> <td>C5</td> <td>09</td> <td>15</td> <td>D0</td>	000050	ED	90	6F	D9	C4	DC	BC	92	DD	B4	05	38	C5	09	15	D0	ED	90	6F	D9	C4	DC	BC	92	DD	B4	05	38	C5	09	15	D0
0x000000 20 B3 3 F A4 F 62 36 0 7 E 0 87 20 26 20 B3 3 F A4 F 62 36 0 7 E 0 87 20 20 E2 20 B3 3 F A4 F 62 36 0 7 E 0 87 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7 38 46 33 20 8 83 46 83 64 30 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 <th2< td=""><td>000060</td><td>98</td><td>C9</td><td>8C</td><td>81</td><td>19</td><td>F2</td><td>8A</td><td>33</td><td>73</td><td>D8</td><td>18</td><td>CD</td><td>15</td><td>8A</td><td>EB</td><td>FB</td><td>98</td><td>C9</td><td>8C</td><td>81</td><td>19</td><td>F2</td><td>8A</td><td>33</td><td>73</td><td>D8</td><td>18</td><td>CD</td><td>15</td><td>8A</td><td>EB</td><td>FB</td></th2<>	000060	98	C9	8C	81	19	F2	8A	33	73	D8	18	CD	15	8A	EB	FB	98	C9	8C	81	19	F2	8A	33	73	D8	18	CD	15	8A	EB	FB
0x000000 A3 C1 24 15 F0 7F 3B 46 B3 A3 2D 8A B3 A3 C1 24 15 F0 7F 3B 46 B3 A3 2D 8A B3 A3 C1 24 15 F0 7F 3B 46 B3 A3 A3 C1 24 15 F0 7F 3B 46 B3 A3 2D 8A B3 A4 A3 C1 24 15 F0 7F 3B 46 B3 A3 A3 C1 24 15 F0 7F 3B 46 B3 A3 C1 24 15 F0 7F 3B 46 B3 A3 A3 C1 24 15 F0 7F 3B 46 B3 A3	000070	2D	B3	3F	8A	4F	62	36	20	7E	D7	EC	90	87	2D	2C	E2	2D	Β3	3F	8A	4F	62	36	20	7E	D7	EC	90	87	2D	2C	E2
0x000000 C4 C1 0F A7 E0 D3 F6 SF 22 F7 65 41 37 A9 92 6D C4 C1 0F A7 E0 D3 F6 SF 22 F7 65 41 37 A9 92 6D C4 C1 0F A7 E0 D3 F6 SF 22 F7 65 41 37 A9 0x000000 25 CD F4 A9 55 44 80 B8 4D A6 B8 F7 25 CD F4 A9 55 44 80 B8 4D A6 B8 F7 25 CD F4 A9 55 44 80 B8 4D A6 B8 F7 25 CD F4 A9 55 44 80 B8 4D A6 B8 F7 25 CD F4 A9 55 44 80 B8 A0 A6 B8 A1 F0 F3 F3 </td <td>000080</td> <td>A3</td> <td>C1</td> <td>24</td> <td>15</td> <td>FO</td> <td>7F</td> <td>3B</td> <td>46</td> <td>B3</td> <td>A3</td> <td>2D</td> <td>8A</td> <td>B3</td> <td>6A</td> <td>86</td> <td>3D</td> <td>A3</td> <td>C1</td> <td>24</td> <td>15</td> <td>FO</td> <td>7F</td> <td>3B</td> <td>46</td> <td>B3</td> <td>A3</td> <td>2D</td> <td>8A</td> <td>B3</td> <td>6A</td> <td>86</td> <td>3D</td>	000080	A3	C1	24	15	FO	7F	3B	46	B3	A3	2D	8A	B3	6A	86	3D	A3	C1	24	15	FO	7F	3B	46	B3	A3	2D	8A	B3	6A	86	3D
0x000000 25 CD F4 A9 55 44 8A F9 64 4A 80 B8 4D A6 B8 F7 25 CD F4 A9 55 44 8A F9 64 A8 80 B8 4D A6 B8 F7 25 CD F4 A9 55 44 8A F9 64 A8 88 4D A6 0x00000B0 4C AF CF B7 BA 68 80 14 6C B2 DB D1 B9 84 50 0A 4C AF CF B7 BA 68 80 14 6C B2 DB D1 B9 84 50 0A 4C AF CF B7 BA 68 80 14 6C B2 D8 D1 B1 F1 P1 P2 C E D3 D4 B1 EA F1 F2 C1 D3 D4 B1 EA C E2 B1 <td>000090</td> <td>C4</td> <td>C1</td> <td>OF</td> <td>A7</td> <td>EO</td> <td>D3</td> <td>F6</td> <td>5F</td> <td>22</td> <td>E7</td> <td>65</td> <td>41</td> <td>37</td> <td>A9</td> <td>92</td> <td>6D</td> <td>C4</td> <td>C1</td> <td>OF</td> <td>A7</td> <td>EO</td> <td>D3</td> <td>F6</td> <td>5F</td> <td>22</td> <td>E7</td> <td>65</td> <td>41</td> <td>37</td> <td>A9</td> <td>92</td> <td>6D</td>	000090	C4	C1	OF	A7	EO	D3	F6	5F	22	E7	65	41	37	A9	92	6D	C4	C1	OF	A7	EO	D3	F6	5F	22	E7	65	41	37	A9	92	6D
0x000000 4C AF CF B7 BA 6B 80 14 6C B2 DB D1 B9 84 50 0A 4C AF CF B7 BA 6B 80 14 6C B2 DB D1 B9 84 50 0A 4C AF CF B7 BA 6B 80 14 6C B2 DB D1 B9 84 50 0A 4C AF CF B7 BA 6B 80 14 6C B2 DB D1 B9 84 50 0A 4C AF CF B7 BA 6B 80 14 6C B2 DB D1 B9 84 50 0A 4C AF CF B7 BA 6B 80 14 6C B2 DB D1 B1 F1 F2 21 0 12 17 EC B9 76 76 S2 S2 C F2 D1 B1 A1 F2	(0000A0	25	CD	F4	A9	55	44	8A	F9	64	4A	80	B8	4D	A6	88	F7	25	CD	F4	A9	55	44	8A	F9	64	4A	80	B 8	4D	A6	B8	F7
0x00000C0 EC B9 97 C5 75 28 30 57 00 75 21 90 12 17 EC B9 97 C5 75 28 30 57 00 75 21 90 12 17 EC B9 97 C5 75 28 30 57 00 75 21 90 0x000000 58 14 F 19 92 5C EC D A 11 C C2 BF 89 58 14 2F 19 92 5C ED D A 1F CE C2 BF 89 58 14 2F 19 92 5C ED D A 1F CE CE D A CE D A T CE CE D A CE D A CE D A CE D A	(0000B0	4C	AF	CF	B7	BA	6B	80	14	6C	B2	DB	D1	B9	84	50	0A	4C	AF	CF	B7	BA	6B	80	14	6C	B2	DB	D1	B9	84	50	0A
0x000000 58 14 2F 19 92 5C EE D3 D4 81 EA 1F CE C2 BF 89 58 14 2F 19 92 5C EE D3 D4 81 EA 1F CE C2 BF 89 58 14 2F 19 92 5C EE D3 D4 81 EA 1F CE C2 0x000000 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 82 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 82 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 4D 79 C5 E5 8F BD 37 7C C4 AC S2 4D 8E 26 03 EC </td <td>(0000C0</td> <td>EC</td> <td>B9</td> <td>97</td> <td>C5</td> <td>75</td> <td>28</td> <td>30</td> <td>57</td> <td>00</td> <td>75</td> <td>D7</td> <td>F5</td> <td>21</td> <td>90</td> <td>12</td> <td>17</td> <td>EC</td> <td>B9</td> <td>97</td> <td>C5</td> <td>75</td> <td>28</td> <td>30</td> <td>57</td> <td>00</td> <td>75</td> <td>D7</td> <td>F5</td> <td>21</td> <td>90</td> <td>12</td> <td>17</td>	(0000C0	EC	B9	97	C5	75	28	30	57	00	75	D7	F5	21	90	12	17	EC	B9	97	C5	75	28	30	57	00	75	D7	F5	21	90	12	17
0x00000E0 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 82 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 82 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 82 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 82 08 7C 5D 18 4F E3 FB 5C 47 74 3D 2A C5 E8 18 20 70 25 E5 8F B0 37 7C C4 AC 58 40 81 41 26 03 EC D9 27 D9 8F	(0000D0	58	14	2F	19	92	5C	EE	D3	D4	81	EA	1F	CE	C2	BF	89	58	14	2F	19	92	5C	EE	D3	D4	81	EA	1F	CE	C2	BF	89
0x0000F0 4D 79 C5 E5 8F BD 37 7C C4 AC C5 82 4D 8E 10 B3 4D 79 C5 E5 8F BD 37 7C C4 AC C5 82 4D 8E 0x000100 26 03 EC D9 2C 67 9D EF CE 49 E9 23 D9 8F 48 4E 26 03 EC D9 2C 67 9D EF CE 49 E9 23 D9 8F JumpTo	(0000E0	08	7C	5D	18	4F	E3	FB	5C	47	74	3D	2A	C5	E8	18	82	08	7C	5D	18	4F	E3	FB	5C	47	74	3D	2A	C5	E8	18	82
0x000100 26 03 EC D9 2C 67 9D EF CE 49 E9 23 D9 8F 48 4E 26 03 EC D9 2C 67 9D EF CE 49 E9 23 D9 8F JumpTo	0000F0	4D	79	C5	E5	8F	BD	37	7C	C4	AC	C5	82	4D	8E	10	B3	4D	79	C5	E5	8F	BD	37	7C	C4	AC	C5	82	4D	8E	10	B3
Jump To	000100	26	03	EC	D9	2C	67	9D	EF	CE	49	E9	23	D9	8F	48	4E	26	03	EC	D9	2C	67	9D	EF	CE	49	E9	23	D9	8F	48	4E
	Jump To																																
Next Difference			ġ	Next	Diffe	rence																											

The difference between file contents and chip contents are highlighted with the "Red Fonts". User can click on the "next difference" button to search for the next different content between the chip and the file contents.

w Content	ts in t	he N	lemo	ory C	hip																											L	*
Source File: C	C:\Use	ers\De	an∖p	eskto	p∖ran	dom\	8 <mark>R.b</mark> i	n												Ope	۱						File I	Buffe	r to F	ile]
Chip: P	m25L	Q032	С																	Rea	d	-					Chip	Buffe	r to F	ile			
Swap (Byte	e Swa	p			Wo	rd Sw	/ap			DW	/ord S	Swap]																			
Checksum:	File	= 07	F67[0 7A	Men	ıory	= 17	7E67I	D7A	Buf	fered	l File	= 07	7F67	D19	Buffe	ered Me	emor	γ=	17E6	7 D7 /	•	,				sł	iow Ii	n (ө н	ex () as	
Checksum:	File	= 07	F67L)7A	Men	10ry +5	= 17	7E67I +7	D7A File	Buff	fered	l File	= 07	7F67 +D	D19	Buffe	ered Me	emor +1	γ = +2	17E6 +3	7 D7 / Me	emory	+6	+7	+9	+0	Sł	iow Ii	1 (● H +D	ex () AS	
Checksum:	File +0	= 07 +1	F67[+2	07A +3	Men +4	10ry +5	= 17 +6 67	7E671 +7	D7A File +8	Buff +9	fered +A	File +B	= 07 +C	7F67 +D	019 +E	Buffe +F	ered Me +0	emor +1	γ = +2	17E6 +3	7 D7 / Me +4	emory +5	+6	+7	+8	+9	sh +A	iow Ii +B	1 (+C	● н +D 4C	ex (+E) AS +F	
Checksum: 	File +0 00	= 07 +1 E4 8F	F67 +2 F2 F1	77A +3 28	Men +4 CA	+5 7A	= 17 +6 67 F1	+7 64	D7A File +8 2E D6	Buff +9 80 28	Fered +A BC	+B 4F 88	= 07 +C 06 7F	7F67 +D 4C	D19 +E D4	Buffe +F 0E 49	ered Me +0 61 F5	+1 E4 8F	γ = +2 F2 F1	17E6 +3 28 AF	Me +4 CA	•mor) +5 7A	+6 67	+7 64	+8 2E D6	+9 80 28	sh +A BC DF	10W II +B 4F 88	+C 06 7E	 н +D 4С 6С 	ex (+E D4 22) AS +F 0E 49	
Checksum: 	File +0 00 F5 18	= 07 +1 E4 8F DE	F67 +2 F2 F1 38	+3 28 AF D2	Men +4 CA 55 88	+5 7A DD 46	= 17 +6 67 E1 10	+7 64 95 70	D7A File +8 2E D6 26	Buff +9 80 28 13	Fered +A BC DE 84	+B 4F 88 E0	= 07 +C 06 7E 53	+D 4C 6C C0	b19 +E D4 22 DC	Buffe +F 0E 49 80	ered Me +0 61 F5 18	+1 E4 8F DE	y = +2 F2 F1 38	17E6 +3 28 AF D2	Me +4 CA 55 88	emory +5 7A DD 46	+6 67 E1 10	+7 64 95 70	+8 2E D6 26	+9 80 2B 13	sh +A BC DE 84	+B 4F 88 E0	+C 06 7E 53	 H +D 4C 6C C0 	ex (+E D4 22 DC) AS +F 0E 49 80	

Chip buffer to file

This will save the chip contents into a user named binary file.

File buffer to file

File buffer can be modified in real time. This button will save the file buffer contents into a user named binary file.

E. Configuration Window Description

This feature allows users to configure advanced settings

1. Batch Operation Option

	- Batch Operation Options					
Batch	🔘 Download a whole file	to chip (With Blank Check), Fill Unused Sj	pace with(Hex): 🛛 🕅	00		
Batch	Oownload a whole file	to chip (Without Blank Check), Fill Unused	d Space with(Hex): 📃	00		
	O Update memory only o	n sector locations with content difference.	Opdate start from	address (He	x)	0
U	- 10- 100x25v21040000000000000000		🔘 Update up to add:	ress (Hex)		1FFFFF
Prog Program	🔘 Update memory and ke	ep one protected area unchanged. Protect a	rea at address(Hex)	0	for	0 bytes
nfiguration	Update memory accord	ing to Region configuration Region 1	➡ From(Hex) 0	[to	FFF
Eng.	🔲 Enable Freescale EzPort	MCU & Send the DIV value (Hex)				
Mode	🔲 Send Specific Data. File	path:	*	Find	1	
402	📝 Identify Chip		- 10 (b)			
SR	🔲 Reload file each time					
dify Status Register	📃 Require Verification aft	er completion				
	Current File in Buffer:	No file in buffer currently				
	Sequences Details (Read Onl;	<i>y</i>)				
Settings	Steps	Actions				
	1	Identify before operation starts.				
	23	Erase Chip Program Chip				

a. Update a Whole file with Blank check

When the user clicks on Batch button, the following operations will be automatically executed:

- 1) Read the memory content
- 2) Blank check (check if Chip is erased)
- 3) Erase the whole memory if not blank
- 4) Program the whole memory with the file

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

b. Update a Whole file without Blank check

When the user clicks on Batch button, the following operations will be automatically executed:

- 1) Erase the whole memory
- 2) Program the whole memory with the file
- 3) Verify if the memory content is identical with the programmed file.

c. Update memory only on sector locations with contents difference or Smart update

User can select the sector locations to have the file programmed.

- <u>Update start from address (Hex)</u>: To program a whole file starting from address 0 of a chip.

- <u>Update up to address (Hex)</u>: To program a whole file, ending at the last address of a chip. The default ending address will automatically calculated by the software according to memory size.

When the user clicks on Batch button, 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 some 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 any change on the rest of the memory (local update).
- A file with only minor change compare to the memory content can be quickly updated. The sectors without difference are kept unchanged.

Remark: the file data which are identical with the target memory but with an address shift (after compilation) will be interpreted as different and will not benefit of the Smart update advantages.

d. **Update memory and keep one protected area unchanged** When the user clicks on Batch button, the following operations will be

automatically executed:

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

2) Insert the read memory contents into the file buffer

3) Erase the whole chip

- 4) Program the whole chip with the updated file in step 2
- 5) Verify the programmed data

e. Update memory according to Region configuration

Sometimes user only wants to update some part of the data in SPI Flash. User can use this function to update the data in the assigned region. This function saves time when debugging.

A. Assign the Region and set start & end address of the Region.

۲	Update memory according to Region configuration	Region D 👻	From(Hex)	0 to	FFP
---	-------------------------------------------------	------------	-----------	------	-----

B. Select working region

📱 DediProg	g Softwar	e 6.0.0.8							
File View	Help								
Detect	(index) File	O Blank	() Erase	(J) Prog	V erify	🛞 Batch	edit	Config	
Currently wo	rking on: rking regio	App n: Rep	plication Me	mory Chip 1	2 (C)	cation Memo Region 3	ory Chip 2 Rej	Update Stand Alone Project gion 4	

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

If the box is checked, the programmer will automatically enable EzPort. Details please see the « EzPort Support » PDF file on CD-ROM or browse "Other Documents" on http://www.dediprog.com/framework.php?UID=154

g. Send Specific Data

If the box is checked, the software will load and send the engineering SPI sequence defined and saved in the "Engineering Mode" Configuration window. This option allows user to create his/her own SPI instruction.

h. Identify Chip

If the box is checked, the software will identify before operation starts.

i. Reload file each time

If the box is checked, the software will load the same file from the source destination each time before the batch operations (refresh). This option is helpful when another software update the file in parallel (like compiler).

j. Require Verification after completion

If this box is checked, the software will verify the contents between the source file and the programmed Serial Flash contents after the batch operations.

Methods Comparison:

Case 1: 64Mb Serial flash update with 64Mb file totally different. Memory has been previously programmed and need to be totally erased. Chip Erase: 50sec Sector Erase (64KB): 0.8sec Chip: 128 sectors

	Update with BC	Update without BC	Smart Update
Memory Read	32 sec	none	32 sec
Compare	1 sec	none	1 sec
Erase	50 sec	50 sec	100 sec
Program	64 sec	64 sec	64 sec
Verify	32 sec	32 sec	32 sec
TOTAL	179 seconds	146 seconds	229 seconds

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: 20%**

Case 2: 64Mb Serial flash programming with a 64Mb file. Memory has never been programmed (from supplier).

	Update with BC	Update without BC	Smart Update
Memory Read	32 sec	none	32 sec
Compare	1 sec	none	1 sec
Erase	0 sec	50 sec	0 sec
Program	64 sec	64 sec	64 sec
Verify	32 sec	32 sec	32 sec
TOTAL	129 seconds	146 seconds	129 seconds

www.dediprog.com

Comparison Chart



Conclusion: If the memory is blank (from supplier), the "Update with Blank Check" or "Smart update" is the optimum choice. **Time Saving: 12%**

Case 3: 64Mb Serial flash update with a 64Mb file with only data differences on 2 sectors or a small file of 2 sectors size only at a specified address. Sector Erase: 0.8 sec Sector Programming: 0.5 sec Sector Verify: 0.25 sec

	Update with BC	Update without BC	Smart Update
Memory Read	32 sec	none	32 sec
Compare	1 sec	none	1 sec
Erase	50 sec	50 sec	1.6 sec
Program	64 sec	64 sec	1 sec
Verify	32 sec	32 sec	0.5 sec
TOTAL	179 seconds	146 seconds	36 seconds

Comparison Chart



Conclusion: If the difference between the memory content and file are small or if the file to be programmed is small, the "Smart update" is the optimum choice. **Time Saving: 75%**

2. Program Configurations

- 1. Program a whole file starting from address 0 of a chip
- 2. Program from specific address of a chip: To program a whole file starting from address 0 of a chip.
- 3. Program up to specific address of a chip: To program a whole file, ending at the last address of a chip. The default ending address will automatically calculated by the software according to memory size.

If the file is smaller than the target Serial Flash, user can define how to fill the rest of the SPI Flash. By default FFh or 00h if selected in the interface.

Program a whole file starting f	rom address 0 of a chip
Fill Unused Space with(Hex):	00
Program from specific address	of a chip
Starting Address(Hex):	0X0000000
Program up to specific address	of a chip
End Address(Hex):	0X1FFFFF

3. Engineering Mode

This function allows users to define their own SPI command and send it directly to the target SPI flash. This option is powerful as all the non standard SPI commands can be generated even if not supported by our programmer.

Users can define the data bytes to be sent from the programmer to the SPI Flash and the number of bytes to be returned. Users can also define if the status register WIP bit has to be polled to check if the SPI Flash is busy or ready.

Users can save the stream data for future use by click on the "Save" button. **For example:**

User wants to 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:

DediProg T	echnology Co. Ltd	www.dediprog.com	
Advanced Setting	<u>js</u>		×
Batch Operations	To Memory Send byte stream: 06	(example: 03 ff 00 12), bytes. second after sending the instruction.	
Prog Program Configuration	Send Sav From Memory 02 Oseconds elapsed,	e	

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

Advanced Setting	5	×
Batch Batch Operations	To Memory Send byte stream: Ø2 00 00 00 01 02 03 Image: Need to return Image: Monitor SR for	(example: 03 ff 00 12), bytes. second after sending the instruction.
Prog Program Configuration Eng. Engineering	Send Save From Memory 00 1.014seconds elapsed.	

Third: The programmer need 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):

Advanced Setting	s		×
Batch Operations	To Memory Send byte stream: 03 00 00 0 ✓ Need to return ✓ Monitor SR for	0 (example: 03 ff 00 12), 8 bytes. second after sending the instruction.	
Prog Program Configuration Eng. Engineering Mode	Send Sav From Memory 00 Oseconds elapsed. 01 02 03 ff ff ff ff	e	

The return bytes from the SPI Flash are displayed in the "from SPI Flash" window: "01 02 03 FF FF FF FF FF".

The engineering mode can be used to send any instruction to the SPI Flash.

4. Modify Status Register

This function allows users to modify or read the status register value of the target serial flash.

The instructions used are:

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

8	Current Value(Hex):	New Value(Hex):
Batch Batch Jerations	9C	FF
3	Read Again	Write to Flash

5. Miscellaneous Settings

Vpp Option:	
Batch Apply Vpp for program and erase when the memory supports it.	
Batch Operations © 3.5V © 2.5V Prov	
Program Configuration SPI Clock Setting Select Clock: 12 MHz	
Eng. Engineering Mode S.R Modify Status Register Werify: I mable Bank Verify Button Bath: I mable Status Verify: I mable Status Verify: I mable Bath Button Bath: I mable Verify Button Bath: I mable Status	
Miscellaneous Settings O Lual/Quad IO Option: Always Single IO Enable Dual IO when available Enable Quad IO when available	
Isolation Free Option:	

a) Vpp

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

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

b) Vcc

SF series programmers support 3.5V, 2.5V, and 1.8V Vcc. Default of 3.5V Vcc is applied after the software installation. User is able to change the Vcc configuration here and the Vcc setting will be changed and saved until next modification.



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 in the supplier specification for a maximum capacitance usually of 30pf or 15pF max. The application is therefore designed to not 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, user cannot expect to program the on board SPI flash according to the maximum frequency of the datasheet as 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

Users can hide some tool bar icons if they unselect the icon items in the "tool bar icon configuration setting". For example, if the engineer only wants the operators to use batch icon, he/she can leaves only batch icon selected and save the setting. The operators will only see the batch icon on the tool bar.

	Vpp Option:
Batch	☐ Apply Vpp for program and erase when the memory supports it
	Vcc Option:
Pmg	⊙ 3.5∀
Program	○ 2.5♥
onfiguration	O 1.8V
Eng	SPI Clock Setting
incering Mode	Select Clock:
0	12 MHz
S.R biy Status	Toolber Icon Configuration:
	Blank: 🔲 Enable Blank Button
SIZ	Erase: 🔲 Enable Erase Button
	Prog: Enable Prog Button
cellaneous Settings	Verify: Enable Verify Button
	Batch: F Enable Batch Button

e View Help	55,610
Detect File	Batch Edit Config
urrently working on:	Application Memory Chip 1 O Application Memory Chip 2 O Update Stand Alone Project
(1) 2009-Oct-07 16:54:5	i1: Welcome to DediProg 5.3.0.10
2009-Oct-07 16:54:5	 Start logging Checking USB connection
✓ 2009-Oct-07 16:54:5	1: USB OK.
(1) 2009-Oct-07 16:54:5	i1: 0.516s elapsed to identify chip.
A 2000 Oct 07 16-54-5	1: Warning: device unrecognizable or not found.

www.dediprog.com

F. Supported Devices, Software Version, *Firmware Version*

User can check the Serial flash support list in our web site. This support list is valid for the latest software and firmware so user will have to check the current software and firmware version he is using and update it if necessary.

Detect File Blank Erase Prog Verify Batch Edit Config	
rently working on: Papel Region 1 Region 2 Region 3 Region 4 Region 5 2011-May 16 10:40:54: Welcome to DedIProg 6.0.0.8 2011-May 16 10:40:54: Welcome to DedIProg 6.0.0.8 2011-May 16 10:40:54: Checking USB connection 2011-May 16 10:40:54: Checking USB connection 2011-May 16 10:40:54: CC3.3v is applied. 2011-May 16 10:40:55: 0.983s elapsed to identify chip. 2011-May 16 10:40:56: Current Type: Pm25LQ032C Firmware Version Firmware Version	Programmer Info Type: SF600 F/W Version: 6.1.9 VCC Status: 3.5V / OFF VPP/Acc: Not Applicable SPI Clock: 12 MHz Dual/Quad IO: Single IO Memory Info Type: Programmer SLQ032C Memory Info
Click Help/About Dediprog to check the supported devices of software	Manufact: 1164 Size(KB): 2048 Manu. ID: 0x9d JEDEC ID: 0x9d7F46 File Info Name : Size: Checksum(File size) : Checksum(Chip size) : Batch Config setting Full Chip update Partial Lindsta and

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 users to plug in and operate multiple SF100/SF200/SF300/SF600 in the same time.

The new software will remove the old USB driver when it detects such driver during the 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.

In order to run more than one SF programmer in 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, USB plug in of all the intended programmers is required prior to opening the software. It is not recommended to add (plug in) or reduce (unplug) any number of programmers when the software is already opened.

The production software does not provide auto chip detect feature and therefore "programmer search" and "chip select" are required prior to any other operations.

The production GUI manual will only illustrate the items not covered by the engineering GUI. Therefore function explanations such as Program, Erase, Blank check, etc will not be repeated here.



A. Search and Select

When click on "search", the software will show programmer type. The default of programmer type is SF100. Please select the programmer you are using and click Rescan.

Search Programmer:

The found programmers will be listed along with site number. The site number is given by the Window OS randomly and therefore users can use the "blink" and "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. Users can use this

www.dediprog.com

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.

Site	Programmer	UID	
Site #1	SF600	DP000000	Blink
Site #2	SF600	DP000000	Blink
			Blink

Select Memory Type:

The production software does not provide auto chip detect feature users will need to select the target memory manually.

ilters:	Memory List:			
Manufacturer AMIC Atmel CFon/EON ESMT Freescale GigaDevice Intel Macronix Numonyx PMC. SANYO SiliconBule Spansion SST	IB00338 1003337 25,F020A 25,F020A 25,F040A 25,F040B 25,F04B			
	ОК	Cancel		Manually selected chip type
	OK	Cancel	L	

After the search step and the to-be-programmed chip is selected, the main GUI will have updated information on the Programmer SITE Status bar, the status window and the log window.

liProg Technology Co. Ltd		W	ww.dedipro	g.com	
DediProg Software 6.0.0.7					
View Help					
Search Select File Blank Erase Prog	Verify Batch Conf	ig Load Prj Save Pr	ġ		
Site #1 Site #2 Site #3	Site #4	Site #5	Site #6	Site #7	Site #8
		P			
Type: SF600 Type: SF600 Type: F/W Ver: 6.1.7 F/W Ver: 6.1.7 F/W Ver:	Type: F/W Ver:	Type: F/W Ver:	Type: F/W Ver:	Type: F/W Ver:	Type: F/W Ver:
Status Window	Statistics		Log Window		
Site Command Status	Success : 0		05/10/11 16:53:15	: Start logging	101 102 00 F-
Site #1 Select Chip OK	Failure: 0	Reset	(i) 05/10/11 16:53:15: Checking USB connection		tion
Site #2 Select Chip OK	lotal: 0		(1) 05/10/11 16:53:20	: Site 1 - SF600	
	Count Down : Disabled Remains : 0 of 0		(1) 05/10/11 16:53:20: Site 2 - SF600 (1) 05/10/11 16:53:25: Site 1 - Type Pm25LQ032C is applied (1) 05/10/11 16:53:25: Site 2 - Type Pm25LQ032C is applied		
	Reload	Count Down	(i) 05/10/11 16:53:30 • 05/10/11 16:53:30	: Loading C: \Users \Dear : C: \Users \Dean \Deskto	n \Desktop \random \8R)p \random \8R.bin Loac
rnject Info			(1) 05/10/11 16:53:30 (1) 05/10/11 16:53:30	: Operation completed. : 0.046 seconds elanser	t.
Memory Info File Info	Batch Config se	tting	S		
Type: Pm25LQ032C Name: 8R.bin					
Manufact.: PMC. Size 0x100000 Charlesum: 0x75747					
Size(KB): 2048 Circcisum: 0x/16/07	3				
JEDEC ID: 0x9d7f46					

B. Batch Config

By clicking on the "config" icon, users can access to configure the batch setting. Users may click on the option "Send Specific Data" for sending the stream data before reading/writing the device. This customized SPI sequence can be created in the "engineering interface".

Users may click on the "Add" or "Remove" for Batch Operation Options directly.

tch.	Send Specific Data. File path:	* Find
ations	Batch Operation Options:	Operation Options Selected :
	Identify Chip (Recommanded) Blank Check Erase Whole Chip Program Chip Checksum Verify	Add >> < Remove

C. Single Site programming

By click on the right mouse button after pointing to a specific Programmer Site number, users will have the access of programming options to the pointed programmer site.



V. Dediprog Windows Command Line

A. Introduction

The window command line has been designed to control our programmer from another 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 following folders: Windows XP:

C:\Documents and Settings\User\Application Data\DediProg\SF100

Windows Vista and Windows 7: 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.

Below are the error messages in the log.txt file.FAIL Identify FailFAIL Blank FailFAIL Erase FailFAIL Program FailFAIL Read FailFAIL Send Specific data Fail

FAIL Verify Fail

FAIL Unknow

To get more information about these methods please contact DediProg.

Window DOS command

Basic Usages:	
Dpcmd –uxxx	
Dpcmd /uxxx	
Dpcmdauto=xxx	
(space is not needed between	the switches and parameters. E.g. dpcmd –ubio.bin \rangle
Basic Switches(switches in t)	his group are mutual exclusive):
-? [he lp]	show this help message
list	nwint sunnowted chin list
-d [detect]	detect chin
$-h \left[h \right]_{ank}$	
-e [erase] -r [read] arg	read chip contents and save to a bin/hex/s19
	- upp CIDOUT for the corports
-p tprog 1 arg	program chip without erase
-u Lauto J 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 Ø
-z [batch] arg	automatically run the following sequence:
	 check if the chip is blank or not;
	- erase the entire chip(if not blank);
	- program a whole file starting from address Ø
-s [sum]	display chip content checksum
-f [fsum] arg	display the file checksum
U U	- needs to work with a file
waw-instruction awa	issue waw sewial flash instwuctions
Taw Instruction arg	- use sparses("") to delimit butes
	- instanctions must be enclosed in double
	QUULALIUN MARKSV 7
	Example:
	dpcmdraw-instruction "03 FF 00 12"
raw-require-return arg (=	=0) decimal bytes of result to return in decimal after issuing raw instructions.
	- used along withraw-instruction only.
	Example:
	dpcmdraw-instruction "03 FF 00 12"raw-req
	uire-return 1
Optional Switches that add fi	ine-tune ability to Basic Switches:
-a [addr] arg he	<pre>kadecimal starting address hexadecimal(e.g.</pre>
0×1	1000),
- ,	works withprog/read/sum/auto only
- 6	lefaults to 0. if omitted.
-1 [length] arg hes	adecimal length to read/program in butes.
	works withmrog/read/sum/auto onlu
	lefaults to whole file if omitted
-u [uerifu]	ifu checksum file and chip
	iny enconsum file and enip
	l anone once with an her welve a PP
$- \frac{1}{1}$ arg (- FF) $F1$	is spare space with an nex value(e.g.FF),
- (works withprog,auto only
type arg Spe	city a type to override auto detection
	ise

www.dediprog.com

IUCK-SUARU ARY	hexadecimal starting address(e.g. 0x1000),
	- works withprog/read/sum/auto only
	- defaults to 0, if omitted.
lock-length arg	hexadecimal length of area that will be kept
	- used along withauto only.
blink arg	
	- Ø : Blink green LED 3 times from USB1 to USBn
	(Default)
	note: the sequence is assigned by US during USB
	- 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 "demice" is not used, the command will
	be executed on all connected programmer.
fix-device arg	Fix programmer serial number with programmer
	sequence.
	 instructions must be enclosed in double quotation
	Example:
	dpcmdfix-device "1 DP000001"
list-device-id arg	
	- Ø : 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. - nº Prompt the deujce ID of programmer connected to
	USBn.
iscellaneous options:	
iscellaneous options: -t [timeout] arg (= -r [target] avg (=1	300) Timeout value in seconds
liscellaneous options: -t [timeout] arg (≕ -g [target] arg (=1	300) Timeout value in seconds) Target Options Available values:
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1	300) Timeout value in seconds) Target Options Available values: 1, Chip 1(Default)
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1	300) Timeout value in seconds > Target Options Available values: 1. Chip 1(Default) 2. Chip 2
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1	300) Timeout value in seconds > Target Options Available values: Chip 1(Default) Chip 2 Socket
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 ucc arg (=0)	300) Timeout value in seconds > Target Options Available values: Chip 1(Default) Chip 2 Socket reference card specify vcc
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 ucc arg (=0)	<pre>300) Timeout value in seconds) Target Options Available values: 1, Chip 1(Default) 2, Chip 2 3, Socket 0, reference card specify vcc 0, 3.5U(Default)</pre>
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0)	<pre>300) Timeout value in seconds > Target Options Available values: 1, Chip 1(Default) 2, Chip 2 3, Socket 0, reference card specify vcc 0, 3.5U(Default) 1, 2.5U</pre>
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0)	300) Timeout value in seconds > Target Options Available values: Chip 1(Default) Chip 2 Socket reference card reference card specify vcc 3.50(Default) 2.50 1.80
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 ucc arg (=0)	<pre>300) Timeout value in seconds > Target Options Available values: 1, Chip 1(Default) 2, Chip 2 3, Socket 0, reference card specify vcc 0, 3.5U(Default) 1, 2.5U 2, 1.8U 1800 ~ 3800, 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only)</pre>
liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0)	<pre>300) Timeout value in seconds > Target Options Available values: 1, Chip 1(Default) 2, Chip 2 3, Socket 0, reference card specify vcc 0, 3.5U(Default) 1, 2.5U 2, 1.8U 1800 ~ 3800, 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase.</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 ucc arg (=0)</pre>	<pre>300) Timeout value in seconds > Target Options Available values: 1. Chip 1(Default) 2. Chip 2 3. Socket 0. reference card specify vcc 0. 3.5U(Default) 1. 2.5U 2. 1.8U 1800 ~ 3800. 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply vpp when the memory chip supports it</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 ucc arg (=0) upp log</pre>	<pre>300) Timeout value in seconds) Target Options Available values: 1. Chip 1(Default) 2. Chip 2 3. Socket 0. reference card specify vcc 0. 3.5U(Default) 1. 2.5U 2. 1.8U 1800 ~ 3800. 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply vpp when the memory chip supports it write operation result into file "%appdata%\dedip mon SF100>log tyt"</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0) vcc arg (=0) log -i [silent]</pre>	<pre>300) Timeout value in seconds) Target Options Available values: 1. Chip 1(Default) 2. Chip 2 3. Socket 0. reference card specify vcc 0. 3.5U(Default) 1. 2.5U 2. 1.8U 1800 ~ 3800. 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply vpp when the memory chip supports it write operation result into file "%appdata%\dedip rog\SF100\log.txt" suppress the display of real-time timer counting</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0) vcc arg (=0) log -i [silent]</pre>	<pre>300) Timeout value in seconds > Target Options Available values: 1, Chip 1(Default) 2, Chip 2 3. Socket 0, reference card specify vcc 0, 3.5U(Default) 1, 2.5U 2, 1.8U 1800 ~ 3800, 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply vpp when the memory chip supports it write operation result into file "%appdata% dedip rog%SF100\log.txt" suppress the display of real-time timer counting - used when integrating with 3rd-party tools(e.g.</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0) vpp log -i [silent]</pre>	<pre>300) Timeout value in seconds > Target Options Available values: 1, Chip 1(Default) 2, Chip 2 3, Socket 0, reference card specify vcc 0, 3.5U(Default) 1, 2.5U 2, 1.8U 1800 ~ 3800, 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply vpp when the memory chip supports it write operation result into file "%appdata% dedip rog%SF100\log.txt" suppress the display of real-time timer counting - used when integrating with 3rd-party tools(e.g. IDE)</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0) vcc arg (=0) vcc arg (=0) vcc arg (=0) vcc arg (=0) vcc arg (=2)</pre>	<pre>300) Timeout value in seconds > Target Options Available values: 1, Chip 1(Default) 2, Chip 2 3, Socket 0, reference card specify vcc 0, 3.5U(Default) 1, 2.5U 2, 1.8U 1800 ~ 3800, 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply upp when the memory chip supports it write operation result into file "%appdata%\dedip rog\SF100\log.txt" suppress the display of real-time timer counting - used when integrating with 3rd-party tools(e.g. IDE) specify SPI clock: 2 12 MHz(Default)</pre>
<pre>liscellaneous options: -t [timeout] arg (=: -g [target] arg (=1 ucc arg (=0) ucc arg (=0) ucc arg (=0) ucc arg (=0) </pre>	<pre>300) Timeout value in seconds > Target Options Available values: 1. Chip 1(Default) 2. Chip 2 3. Socket 0. reference card specify vcc 0. 3.50(Default) 1. 2.50 2. 1.80 1800 ~ 3800. 1.8 ~ 3.80 (minimum step 100mU) (For SF600 only) - work withprog anderase. apply upp when the memory chip supports it write operation result into file "%appdata%\dedip rog\SF100\log.txt" suppress the display of real-time timer counting - used when integrating with 3rd-party tools(e.g. IDE> specify SPI clock: 2. 12 MHz(Default) 0. 24 MHz</pre>
<pre>liscellaneous options: -t [timeout] arg (=: -g [target] arg (=1 ucc arg (=0) ucc arg (=0) upp log -i [silent] spi-clk arg (=2)</pre>	<pre>300) Timeout value in seconds > Target Options Available values: 1. Chip 1(Default) 2. Chip 2 3. Socket 0. reference card specify vcc 0. 3.50(Default) 1. 2.50 2. 1.80 1800 ~ 3800. 1.8 ~ 3.80 (minimum step 100mU) (For SF660 only) - work withprog anderase. apply vpp when the memory chip supports it write operation result into file "%appdata%\dedip rog\SF100\log.txt" suppress the display of real-time timer counting - used when integrating with 3rd-party tools(e.g. IDE) specify SPI clock: 2. 12 MHz(Default) 0. 24 MHz 1.8 MHz</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0) vcc arg (=0) vcc</pre>	<pre>300) Timeout value in seconds) Target Options Available values: 1. Chip 1(Default) 2. Chip 2 3. Socket 0. reference card specify vcc 0. 3.5U(Default) 1. 2.5U 2. 1.8U 1800 ~ 3800, 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply vpp when the memory chip supports it write operation result into file "%appdata%\dedip rog\SF100\log.txt" suppress the display of real-time timer counting - used when integrating with 3rd-party tools(e.g. IDE) specify SPI clock: 2, 12 MHz(Default) 0, 24 MHz 1, 8 MHz 3, 3 MHz 3, 3 MHz</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0) vcc arg (=0) log -i [silent] spi-clk arg (=2)</pre>	<pre>300) Timeout value in seconds) Target Options Available values: 1. Chip 1(Default) 2. Chip 2 3. Socket 0. reference card specify vcc 0. 3.5V(Default) 1. 2.5U 2. 1.8U 1800 ~ 3800, 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply vpp when the memory chip supports it write operation result into file "%appdata%\dedip rog\SF100\log.txt" suppress the display of real-time timer counting - used when integrating with 3rd-party tools(e.g. IDE) specify SPI clock: 2, 12 MHz(Default) 0, 24 MHz 1, 8 MHz 3, 3 MHz 4, 2.18 MHz 5, 1.5 MHz</pre>
<pre>liscellaneous options: -t [timeout] arg (= -g [target] arg (=1 vcc arg (=0) vcc arg (=0) i [silent] spi-clk arg (=2)</pre>	<pre>300) Timeout value in seconds) Target Options Available values: 1. Chip 1(Default) 2. Chip 2 3. Socket 0. reference card specify vcc 0. 3.5U(Default) 1. 2.5U 2. 1.8U 1800 ~ 3800, 1.8 ~ 3.8U (minimum step 100mU) (For SF600 only) - work withprog anderase. apply vpp when the memory chip supports it write operation result into file "%appdata% dedip rog%SF100\log.txt" suppress the display of real-time timer counting - used when integrating with 3rd-party tools(e.g. IDE) specify SPI clock: 2, 12 MHz(Default) 0, 24 MHz 1, 8 MHz 3, 3 MHz 4, 2.18 MHz 5, 1.5 MHz 6, 750 KHz</pre>

B. 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 direct the dpcmd command to the "dpcmd.exe" location.

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

C. Basic Usages

- 1. dpcmd -r"f:\file.bin",
- reads the chip and save it into a file "file.bin"
- 2. dpcmd -rSTDOUT -a0x100 -l0x23, reads 0x23 bytes starting from 0x100 and display it on the screen
- 3. dpcmd -ufile.bin, erases and then program file.bin into the serial flash
- 4. dpcmd -pfile.bin -a0x100, writes file.bin into the serial flash starting from address 0x100
- 5. dpcmd -pfile.bin -x0xaa, programs file.bin into the serial flash and fill the rest area with 0xaa

Remarks: -a, -l only works with -p, -r, -s

Remarks: -x only works with -p

Remarks: space is not needed between the switches parameters. E.g. dpcmd -ubio.bin

D. Basic Switches

-? [help]	show this help message
list	print supported chip list
-d [detect]	detect chip
-b [blank]	blank check
-e [erase]	erase entire chip
-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

31

-z [batch] arg	 Erase onl Program data from automatica check if t erase the program 	y the sectors with some differences only the erased sectors with the file address 0 ally run the following sequence: he chip is blank or not; entire chip(if not blank); a whole file starting from address 0
-s [sum]	display cl	nip content checksum
-f [fsum] arg	display the	e file checksum
_	- needs to	work with a file
raw-instruction arg		issue raw serial flash instructions.
raw-require-return	arg (=0)	 use spaces(" ") to delimit bytes. instructions must be enclosed in double quotation marks("") Example: dpcmdraw-instruction "03 FF 00 12" decimal bytes of result to return in decimal after issuing raw instructions. used along withraw-instruction only. Example: dpcmdraw-instruction "03 FF 00 12" raw-require-return 1

E. Optional Switches

(specify the following switches to change default values):

-a [addr] arg	hexadecimal starting address hexadecimal
	(e.g. 0x1000),
	- works withprog/read/sum/auto only
	- defaults to 0, if omitted.
-l [length] arg	hexadecimal length to read/program in bytes,
	- works withprog/read/sum/auto only
	- defaults to whole file if omitted
-v [verify]	verify checksum file and chip
	- works withprog/auto only
-x [fill] arg (=FF)	fill spare space with an hex value(e.g. FF),
	- works withprog,auto only
type arg	Specify a type to override auto detection
	- uselist arguement to look up supported type.
lock-start arg	hexadecimal starting address(e.g. 0x1000),
	- works withprog/read/sum/auto only
	- defaults to 0, if omitted.
lock-length arg	hexadecimal length of area that will be kept

	unchanged while updating
	- used along withauto only.
blink arg	
	- 0 : Blink green LED 3 times from USB1 to USBn (Default)
	note: the sequence is assigned by OS during
	- 1 : Blink the programmer connected to USB1 3
	- n : Blink the programmer connected to USBn 3 times.
device arg	(work with all Basic Switches)
C	- 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 on all connected programmer
fix-device arg	Fix programmer serial number with programmer
	- instructions must be enclosed in double quotation marks("")
	Example:
	dpcmdfix-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
	- 1 : Prompt the device ID of programmer connected
	- n : Prompt the device ID of programmer connected to USBn.
Miscellaneous options:	

iscellaneous options:

-t [timeout] arg (=300)	Timeout value in seconds
-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(Default)
	1, 2.5V
	2, 1.8V

33

	1800 ~ 3800, 1.8 ~ 3.8V (minimum step
	100mV) (For SF600 only)
	- work withprog anderase.
vpp	apply vpp when the memory chip supports it
log	write operation result into 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:
	2, 12 MHz(Default)
	0, 24 MHz
	1, 8 MHz
	3, 3 MHz
	4, 2.18 MHz
	5, 1.5 MHz
	6, 750 KHz
	7, 375 KHz

F. Exit Code

```
enum ErrorCode
{
EXCODE_PASS,
EXCODE_FAIL_ERASE,
EXCODE_FAIL_PROG,
EXCODE_FAIL_VERIFY,
EXCODE_FAIL_READ,
EXCODE_FAIL_BLANK, // 5
EXCODE_FAIL_BATCH,
EXCODE_FAIL_CHKSUM,
EXCODE_FAIL_IDENTIFY,
EXCODE_FAIL_OTHERS=99,
};
```

VI. Stand Alone Mode (SF300 & SF600)

In addition to the functions provided by SF100 and SF200, SF300 & 600 further allow users to program serial flash memories in the standalone mode.

A. SF300 Standalone Programming:

SF300 embeds a 128Mb memory which allows users to pre-download the reference contents and configurations before performing the programming in standalone mode.

Standalone mode procedures:

- 1. Project preparation
- 2. Mode switch
- 3. Standalone programming

1. Project Preparation

In order to perform standalone programming, the contents and the programming operation procedures have to be pre-downloaded to the SF300 through the USB with the software provided by Dediprog. SF300 has embedded a 128Mb SPI Flash and therefore it is capable of supporting standalone programming for all serial flash equal and under 128Mb.

Prepare a standalone programming project

- a. Connect the SF300 to a PC with SF software installed and make sure the programmer is switched to USB mode.
- b. Open the SF software and click on the "update standalone project" tab.

SF300 software interface

Delipmg Software 4. 1. 70. 3838				
Eile View Help				
Detect File Config Update				
Currently working on: O Application Memory Chip 1 O Application Memory Chip 2 O Update Stand Alone Project				
(1) 07/29/08 16:57:05: Start logging (1) 07/29/08 16:57:05: Checking US8 connection ♥ 07/29/08 16:57:05: US8 OK.	Powered by			
	Programmer Info Type: SF200 F/W Version: 30.1 VCC Status: OFF VPP/Acc: Apply If SA Board Info Type: Manufact.: Max			
	- S/A Config Info Name : Size: Checksum : Vpp/Acc: SA Operations: Target Chip Size:			

- c. Load the file
- d. Click on "Config" to select the target chip type and the standalone operation procedures
 - Erase + Program + Verify
 - Blank Check (Erase if no Blank) + Program + Verify
 - Program + Verify
- e. Click on "Update" to download the project information to the SF300

2. Stand Alone Mode

SF300 provides 2 operation modes through a button switch.

- USB mode: to control the programming via the computer tool
- Stand Alone mode: to work independently from computer

3. Standalone Programmer Operations

When the project is prepared in the SF300 internal Serial Flash and the mode is switched to SA mode, the user will see pass counter, fail counter and the target chip type on the LCD screen. The user can start the standalone programming by the following steps.

- a. insert the target chip into the socket(or insert the ISP cable header to the target application header for standalone ISP)
- b. press the start button
- c. wait for the "PASS" LED ON and remove the chip from the socket

The LCD screen will display information on the programming on going: like pass/fail counter and error handling.

When programming error occurs, the Red LED will be on. The user can exit the error status by pressing the start button once and SF300 will increment the fail counter.

Reset Counter

By pressing the reset button, SF300 will reset both the pass counter and fail counter to zero.



4. SF300 Hardware Description

a. LCD Display

LCD displays the standalone project related information such as the file checksum, chip type, pass counter, and fail counter.

b. Start Button

By pressing the start button, the SF300 starts to execute the operation procedures defined in the project pre-downloaded to the SF300.

c. Reset Counter Button

The reset button is used to reset the counter information.

d. USB Connector

USB connector is used to communicate with the SF software during the USB mode or to provide the power during the standalone mode.

e. **DIP Socket**

DIP socket is used to connect to different socket adaptors provided by Dediprog in order to support all serial flash packages.

f. LED Display

Red Led: error Orange Led: operation on going Green Led: pass

B. SF600 Standalone Programming:

To work in Stand Alone mode, SF600 needs to be connected to DediProg LCD Keypad and MicoSD card. SF600 will also use the Micro SD card to run the project.

Important Notice:

Every MicroSD card from Dediprog is encrypted by Dediprog before shipment. SF600 only works with the MicroSD card encrypted by Dediprog.

SF600 offers 2 kinds of standalone programming mode:

- 1. Standalone programming with LCD Keypad.
- 2. Standalone programming without LCD Keypad.
- A. Project definition from the Dediprog Production Software.
- B. Save Project to the Micro SD card.
- C. Press start button to run project in standalone mode.

1. Project preparation

In order to perform stand alone programming, the contents and the programming operation procedures have to be pre-downloaded to the SF600 MicroSD card through the Dediprog Production Software provided by Dediprog.

Prepare a standalone programming project



40

2. Select IC brand and part number.

Site #1 Site #2 Site #3 Site #4 Site #5 Site #6 Vipe: Type: Type: Filers: Menually Select Memory Type Visite #3 Site #6 Filers: Menually Select Memory Type Visite #3 Site #6 Visite #6 Filers: Menually Select Memory Type Visite #7 Visite #7 Visite #7 Status Window Visite #7 Visite #7 Visite #7 Visite #7 Status Window Nantronics Nantronics Visite #7 Visite #7 Visite Command Status Status Window Visite #7 Visite #7 Project Info Nantronics Visite #7 Visite #7 Visite #7 Project Info Nantronics Visite #7 Visite #7 Visite #7 Project Info Nantronics Corp Visite #3 Visite #7 Visite #7 Project Info Nantronics Corp Visite #7 Visite #7 Visite #7 Manufact: Size (K8): Nantronics Corp Visite #4 Visite #7 Visite #7 Visite #10 Nantronics Visite #7 Visite #7 Visite #7	le View Help Search Select File Bla) (9) (8) (8) (8) (9) (9) (9) (9) (9) (9) (9) (9) (9) (9	Config Load Prj Save Prj	1	
Project Info Memory Info Type: Name : Name : Size Size Size(KB): Checksum(Chip size) : JEDEC ID: SA Operations: Checksum(Chip size) : Checksum(Chip	Site #1 Site #2 Site #	site #3 Site #4 Manually Select Memory Type Filters: Manufactuar Fudam Microlectronics GigaDevice Intel Macronix Nantronics Nantronix Status Status 2 Sarto SkortiBule Spansion 2 SST Terra Sension Justre	4 Site #5 Sith Memory List:	site #7 Site #7 Type: F/W Ver: D04:30: Start logging D04:31: Entror Hardware not D04:31: Hint: Drogrammer m E	Site #8
	Project Info Memory Info Type: Manufact.: Size(KB): Manu. ID: JEDEC ID:	File Info Name : Size Checksum(Chip size) : SA Operations:	₩25x80(Å) ₩25x806 ₩25x806 Cancel		

3. Load the file. - 🗆 X 2 ediProg Software 6.0.3.3 1. () Prog (100) 🔞 🧐 🔥 🦶 Batch Config Load Prj Save Prj Search Select Blank () Erase **√** Verify File Site #1 Site #2 Site #3 Site #4 Site #5 Site #6 Site #7 Site #8 P P P P P P P X Type: F/W Ver: Type: F/W Ver: Load File Type: F/W Ver: Type: F/W Ver: File Path: D:\random\2R.bin Find Status Window Program as Site Command Start logging ... Data Format:
 Raw Binary
 Intel Hex
 Motorola S19
 ROM Checking USB connection Error: Hardware initialization failed. Hint: USB connection might not be working. Hint: Programmer might not be present. Truncate file to fit in the target area. 3. Erase buffer before loading. Erase with ОК Cancel Project Info File Info Memory Info Batch Config setting Type: W25X16(A) Name : Manufact.: Winbond Size Size(KB): 2048 Checksum(File size) : Manu. ID: 0xef Checksum(Chip size): JEDEC ID: 0xef3015 SA Operations: 111

4. Click "Config" Icon to set programming flow. **Important Notice:**

"Identify Chip" is necessary for SF600 standalone programming. Be sure to include "Identify Chip" in programming flow.



5. Save dprj file to MicroSD card.

Site #1	Site #2	Sit	te #3	Site #4		Site #5	Site #6		Site #7	Site #8
P	P			電腦 > 抽取式	磁碟 (H:)			- 4+	授業 相政式選擇(H)	-×
o	0			新増資料夾	* *		*		10	- 0
ype: /W Ver:	Type: F/W Ver:	F		* 名種	[*		修改日期	類型	7
tatus Window	2	1	■ 電腦		BBB.dprj			2012/01/03	下午 DPRJ 檔案	
				2.	CCC.dprj			2012/01/03	下午 DPRJ 檔案	
Site Commany	1	status	抽取式磁码	≆ (H:) =	DDD.dprj			2012/01/03	下午 DPRJ 福案	
			Volume :		EEE.dprj			2012/01/03	下午 DPRJ 個英 下午 DPDJ 増廃	
			ftp.dedip	roa.c	rrr.aprj GGG davi			2012/01/03	下十 DPRO 個與 下午 DDPI 標準	
			My Web	Sites in	HHH dori			2012/01/03	下午 DPRI檔案	
			0.0	n n	∏ dori			2012/01/03	下午 DPRI 檔案	
			🕨 網路		JJJ.dori			2012/01/03	下午 DPRJ 檔案	
			📕 123-PC	+ 6		3.	-10			
roject Info			****	D. Indian						
demory Info	Få	e Info	個杂合傳							
Type: W25X:	.6(A) Na	ime :	存個類型	(T): Dedi Projec	t Files (*.dprj)		4,	1	
Size(KB): 2048		ecksun						- Г		
Manu ID: 0xef	d	ecksun	隱藏資料夾						OK (Cancel
Turius AD1		censul							100 C	

2. Standalone programming

Important Notice:

Be sure to turn SF600 off before insert MicroSD card to SF600.

a) Standalone programming with LCD Keypad

When the project is available in MicroSD card and the LCD Keypad is connected. User can refer to the user manual of LCD Keypad to select project name and start standalone programming.

b) Standalone programming without LCD Keypad.

SF600 also can do standalone programming without LCD Keypad. User has to generate a txt file named **Autorun** and key the name of project user want s to program into the Autorun.txt file. Then save the Autorun.txt file to MicroSD card. SF600 will automatically detect the project keyed in Autorun.txt file and execute the project.

Important Notice:

Be sure to include the subname (.dprj) when saving the project name in Autorun.txt file.



www.dediprog.com

rganize 🔻 🦷 Share v	vith 🔻 Burn New folder			
Favorites	Name	Date modified	Туре	Size
🐞 Downloads	鷆 project	2012/01/04 下午 0	File folder	
📃 Recent Places	📋 Autorun	2012/01/04 下午 0	Text Document	1 KB
	W25X16A.dprj	2012/01/03下午0	DPRJ File	597 KB
🗃 Libraries				
Documents				
J Music				
E Pictures				
Videos				
👌 Homegroup				
Computer				
🏭 OS (C:)				
👝 DATA (D:)				
📕 DVD RW Drive (E:)				
SD/MMC (F:)				
🍌 project				
C volume 1 0\1021				

When pressing start bottom of SF600, SF600 will automatically execute the project "W25X16A.dprj".

c) Start Standalone programming.

Press **Start** button for 2 seconds to run the project stored in MicroSD card in standalone mode.

3. SF600 Hardware Description



a. Micro SD Card Slot

Insert the Micro SD card with programming project when executing stand alone programming.

X SF600 is only compatible with the Micro SD card supplied from DediProg.

b. LCD Keypad Connector

Connect LCD Module to SF600 when executing stand alone programming. LCD displays the standalone project related information such as the file checksum, chip type, pass counter, and fail counter.

c. Com Port

Com Port is for the application of integrating SF600 to customers' own system.

Pin Out	
1	2
GND	NC
3	4
5V	START
5	6
FAIL	PASS
7	8
BUSY	NC

d. Start Button

By pressing the start button, the SF600 starts to execute the operation procedures defined in the project pre-downloaded to the Micro SD card.

e. ICP Connector

ICP connector is used to connect ICP cable when executing ICP programming.

f. Socket Adaptor Headers

Socket adaptor headers are used to connect to different socket adaptors provided by Dediprog in order to support all serial flash packages.

g. Power LED

Power LED shines when SF600 is powered by USB or power adaptor.

h. Power Connector

Connect power adaptor to SF600 when executing stand alone programming. USB can also be used as power source during standalone programming.

i. USB Connector

USB connector is used to communicate with the SF software during the USB mode or to provide the power during the standalone mode.

j. Operation LED

Red Led: error **Orange Led:** operation on going **Green Led:** pass

VII. Specific Functions (SF600)

A. Dual/Quad IO:

User can find Dual/Quad IO option in **Config/Miscellaneous Settings**. The default of Dual/Quad IO option is "Always Single IO". SF600 supports Dual and Quad IO. When using a SPI Flash with Dual/Quad IO function, user can select Dual or Quad IO mode.

500	Batch: 🔽 Enable Batch Button
	Dual/Quad IO Option:
Miscellaneous Settings	Always Single IO
	🔵 Enable Dual IO when available
	🔘 Enable Quad IO when available
	Isolation Free Option:
	Apply Isolation free when using ICP mode

B. Isolation Free

SF600 is the only programmer in the market able to support the isolation free method. Isolation free method is the ideal solution if the application controller does not release the SPI bus in high impedance during reset and if the isolation circuit has not been implemented due to cost reason or design history.

User can find this function in **Config/Miscellaneous Settings** and enable this function to avoid signal conflict when the isolation circuit has not been implemented on the application board.



47

C. Hold Pin Status Setting

SF600 programmer is available to set HOLD pin status through software. Please go through Config and change it under engineering mode. Click on HOLD button to change the status of HIGH or LOW. This function is available at SF600 only.

2 DediProg Software 6.0.2.11 File View Help	
• (iii) Detect File Blank Frase Pr	rog Verify Batch Edit Config
Currently working on: Application Memory (Chip 1 O Application Memory Chip 2 O Update Stand Alone Project
Currently working region: Advanced Settings	
0 2011-Dec-09 11:25: 1 2011-Dec-09 11:25: 2 2011-Dec-09 11:25:	Batch Operation Options Download a whole file to chip (With Elank Check), Fill Unused Space with(Hex): O Download a whole file to chip (Without Elank Check), Fill Unused Space with(Hex): O Update memory only on sector locations with content difference. Update start from address (Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. Protect area at address(Hex) Update memory and keep one protected area unchanged. From Hex O From Hex O
Miscellaneous	Current File in Buffer: No file in buffer currently
Settings	Sequences Details (Read Only)
	Steps Actions
< No operation on-going	1 Identify before operation starts. 2 Erase Chip 3 Frogram Chip

Setect File B	ank Erase	Prog Verify Batch Edit Config Load Prj Save Prj	
rently working on:	Application Men Region 1 Advanced Settin	ory Chip 1 O Application Memory Chip 2 Update Stand Alone Project Region 2 Region 3 Region 4 Region 5 gs	X
2011-Dec-09 11:25: 2011-Dec-09 11:25: 2011-Dec-09 11:25: 2011-Dec-09 11:25: 2011-Dec-09 11:25: 2011-Dec-09 11:25: 2011-Dec-09 11:25:	Image: Construction of the second	To Memory Send byte stream: (example: 0.3 ff 00 12), Need to return. bytes. Monitor SR for Serve HOLD pin control HOLD Hold pin High From Memory N/A	
		Warning: any on-going operation(e.g. ense) will be terminated immediately after closing this page	

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 stand alone programming.
2012/12/20	6.1	Revise the CLI detail and add exit codes.

For more information please contact us or your motherboard suppliers.

We also recommend motherboard makers to enter in contact with our technical team to create a dedicated document that will take into consideration all your motherboard updating constraints and references. This documentation will then be very helpful to simplify the Bios update and avoid any mistake in the field.

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.

All rights reserved Printed in Taiwan.