SF100/SF600/SF600Plus For Linux System

Version 1.0

©DediProg Technology Co., Ltd 2015 All right reserved.

Table of contents：

I. General Description.....2

II. Root Privileges 2

III. USB Installation.....2

IV. Introduction.....2

4.1 Basic Usages 2

4.2 Basic Switches 3

4.3 Optional Switches that add fine-tune ability to Basic Switches: 4

4.4 Miscellaneous options: 5

V. Revision History.....6

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. General Description

DediProg have developed command line interface (DpCmd) especially for Linux operating system, which provide various command for the user's programming needs. This allows the user to have more flexibility in developing own software to call "DpCmd" thus add convenient IC programming.

II. Root Privileges

Root rights is required before each execution of "DpCmd", so DediProg has provided a script (setup_udev.sh) which the user needs to execute as Root before executing the "DpCmd" for the first time, then "DpCmd" can be executed directly.

III. USB Installation

Using Linux command to Installed USB driver.

Ubuntu: sudo apt-get install libusb-dev Fedora/Centos: sudo yum install libusb-devel.

IV. Introduction

4.1 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

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 –uf:\file.bin

4.2 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
-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 file data from address 0
-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 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. - needs to work with a file issue raw serial flash instructions. - use spaces(" ") to delimit bytes. - instructions must be enclosed in double quotation marks("") Example: dpcmd --raw-instruction "03 FF 00 12"
--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

4.3 Optional Switches that add fine-tune ability to Basic Switches:

-a [--addr] arg	hexadecimal starting address hexadecimal (e.g. 0x1000), - works with --prog/read/sum/auto 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 whole file if omitted
-v [--verify]	verify checksum file and chip - works with --prog/auto only
-x [--fill] arg (=FF)	fill spare space with an hex value(e.g.FF), - works with --prog, --auto only
--type arg	Specify a type to override auto detection - use --list arguement to look up supported type.
--lock-start arg	hexadecimal starting address(e.g. 0x1000), - works with --prog/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 with --auto only.
--blink arg	- 1: Blink the programmer connected to USB1 3 times.
--list-device-id arg	- 1: Prompt the device ID of programmer connected to USB1.

4.4 Miscellaneous options:

Note: The programming operation always uses the default value for command. If users want to use other setting, must add the wanted option to every command.

-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 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 /tmp/log.txt Note: If user didn't use this command, the operation result will be recorded in "log.txt"
-i [--silent]	supress 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
--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)

V. Revision History

Date	Version	Change
07/14/2015	1.0	Initial release

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

- **Shanghai Office** TEL: 86-21-5160-0157 FAX: 86-21-6126-3530
  Room 503, Block E, No.1618, Yishan Road, Shanghai, China

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

Sales Support：**sales@dediprog.com**

www.DediProg.com

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

All rights reserved
Printed in Taiwan.