

## Overview

The Serial Flash Utility is a command line program that can be used to read and write data to serial flash memories using the SPI interface, such as are found on the Digilent PmodSF. It is used with the Digilent Adept runtime software and a Digilent Adept compatible programming device supporting the DSPI interface, such as the JTAG-USB Full Speed Module, or the JTAG-USB Programming Cable.

## Usage

In some cases, the SPI flash memory device is built into the Adept compatible product. In this case, no special connections are required. Refer to the documentation for the specific product for information such as the flash memory type, port numbers and other parameters to use.

To use the Serial Flash Utility to interface with a Digilent PmodSF and a Digilent Adept compatible programming cable, such as the JTAG-USB Full Speed Module, it is necessary to provide power to the PmodSF and the programming cable. This can be done using a Digilent PmodREG1 and a Digilent 5V regulated power supply. Attach connector J1 of a PmodREG1 to the programming cable. Then, attach the PmodSF to connector J2 of the PmodREG1. A Digilent 5V power supply or other regulated power supply is then used to power the PmodREG1.

## Syntax

The general command syntax is as follows:

```
sfutil [-<opt1> <val1>] [-<opt1> <val2>] ...
```

The following command line options are supported:

**-d <devname>**

Specify the name of the Digilent Adept compatible programming device. This string is used to open the connection to the programming device. The programming device name is case sensitive. There is no default and so this is a required option. This name can be either the name that is stored in the programming device itself, or a name stored in a device table maintained by the Adept software.

**-m <memtype>**

Specify the type of SPI memory device to be programmed. If this option isn't specified, the program will attempt to read the id code from the device and map it to a known device. The program supports the following values (these are not case sensitive):

M25P80      Micron/ST Microelectronics/Numonyx 8 Mbit part

M25P16      Micron/ST Microelectronics/Numonyx 16 Mbit part

M25P32	Micron/ST Microelectronics/Numonyx 32 Mbit part
M25P64	Micron/ST Microelectronics/Numonyx 64 Mbit part
M25P128	Micron/ST Microelectronics/Numonyx 128 Mbit part
N25Q128	Micron 128 Mbit part
AT45DB08	Atmel 8Mbit part

**-id**

Read and print the manufacturer id code from the device.

**-w <filename>**

Write the contents of the specified file to the memory device. When writing, sector erase commands will be used to erase sectors in the flash memory device that written. This erase can be suppressed by using the **-NE** option.

**-r <filename>**

Read the contents of the memory device and write it to specified file.

**-e**

Erase the range of sectors specified by the **-a** and **-l** parameters. This is done using sector erase commands.

**-E**

Bulk erase the entire flash memory device before programming. This is done by default when writing to the memory device.

**-NE**

Don't erase the device before programming.

**-a <address>**

Specify the device memory address at which to start reading or writing. The default starting address is 0 if this is not specified. The value can be given as either a decimal number or a hexadecimal number (using the 'C language' 0xABCD syntax for specifying a hexadecimal value).

**-l <length>**

Specify the number of bytes to read or write. The value can be given in either decimal or hexadecimal. If this option is not specified when writing the contents of a file to the memory device the entire contents of the file will be programmed. If this option is given but is greater than the length of the file, only the number of bytes specified by the source file will be written. If this option is not specified when reading, the entire contents of memory (starting at the address specified by the **-a** option) will be read from the memory device and written to the file.

**-fb, -fi, -fh**

Specify the file data format: **-fb**, raw binary; **-fi**, Intel hex file format; **-fh** generic hex file format. For the raw binary format, the file data is written to memory unmodified, or the memory data is written to the file unmodified. The generic hex file format assumes each line in the source file contains a series of bytes as two digit hexadecimal numbers separated by spaces.

`-p <port>`

Specify the port number of the SPI port on the device to use. The default is port 0.