
SmartFusion2 SoC FPGA In-System Programming Using USB OTG Controller Interface

Demo Guide

Superseded

May 2014

Revision History

Date	Revision	Change
3 May 2014	3.0	Fourth release
16 December 2013	2.0	Third release
27 November 2013	1.0	Second release
09 October 2013	0.0	First release

Confidentiality Status

This is a non-confidential document.

Superseded

Table of Contents

Preface	4
About this document	4
Intended Audience	4
References	4
Microsemi Publications	4
In-System Programming Using USB OTG Controller Interface	5
Introduction	5
Requirements and Details	5
Demo Design	6
Introduction	6
Demo Design Features	8
Demo Design Description	8
Setting Up the Demo Design	11
Alternate Setup	12
Board Setup Snapshot	12
Running the Demo Design	12
Authenticate Operation Mode	17
Verify Operation Mode	19
Program Operation Mode	21
Checking if the Fabric is Programmed Successfully	22
Checking if the eNVM is Programmed Successfully	22
Programming Results	23
Appendix 1: Connecting the SmartFusion2 Device to the Host PC Through the USB to UART (FTDI) Interface	24
Appendix 2: Board Setup when using the USB-RS232 Serial Adapter or RS232 Cable	25
Appendix 3: Board Setup Through the USB to UART (FTDI) Interface using the USB A to Mini - B Cable	26
Appendix 4: Jumper Locations	27
Appendix 5: Error Codes	28
Appendix 6: Generating .spi Programming File using Libero	29
A List of Changes	-31
B Product Support	-32
Customer Service	32
Customer Technical Support Center	32
Technical Support	32
Website	32
Contacting the Customer Technical Support Center	32
Email	32
My Cases	33
Outside the U.S.	33
ITAR Technical Support	33

Preface

About this document

This demo is for SmartFusion[®]2 system-on-chip (SoC) field programmable gate array (FPGA) devices. It provides instructions on how to use the corresponding reference design.

Intended Audience

SmartFusion2 devices are used by:

- FPGA designers
- Embedded designers
- System-level designers

References

Microsemi Publications

- SmartFusion2 Programming User Guide
- SmartFusion2 System Controller User Guide
- SmartFusion2 Microcontroller Subsystem User Guide
- SmartFusion2 SoC FPGA Remapping eNVM, eSRAM, and DDR/SDR SDRAM Memories Application Notes
- Configuring Serial Terminal Emulation Programs Tutorial

See the following web page for a complete and up-to-date listing of SmartFusion2 device documentation:
<http://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2#documents>.

In-System Programming Using USB OTG Controller Interface

Introduction

You can use in-system programming (ISP) to reprogram for design iterations and field upgrades. SmartFusion2 devices support ISP using universal serial bus (USB) on-the-go (OTG) controller interface. This document describes how to program the following using ISP through the USB OTG controller interface:

- Embedded nonvolatile memory (eNVM)
- FPGA fabric
- Both the eNVM and the FPGA fabric

For information on different programming modes supported by SmartFusion2 SoC FPGAs, refer to the [SmartFusion2 Programming User Guide](#). For information on USB OTG controller, refer to the [SmartFusion2 Microcontroller Subsystem User Guide](#).

Requirements and Details

Table 1 • Reference Design Requirements and Details

Reference Design Requirements and Details	Description
Hardware Requirements	
SmartFusion2 Development Kit <ul style="list-style-type: none">• 12 V adapter• FlashPro4 programmer• USB A to Mini-B cable• USB cable with Micro-A end	Rev C or later
USB-RS232 Serial adapter or RS232 cable	-
Host PC or Laptop	Windows 64-bit Operating System
Software Requirements	
Liberio® System-on-Chip (SoC) for viewing the design files <ul style="list-style-type: none">• FlashPro Programming Software v11.3	11.3
Host PC Drivers	USB to UART drivers

Demo Design

Introduction

The demo design files are available for download from the following path in the Microsemi® website:
http://soc.microsemi.com/download/rsc/?f=sf2_isp_using_usb_interface_demo_df

The demo design files include:

- Sample programming files
- Libero SoC project
- STAPL programming file
- readme.txt

Figure 1 shows the top level structure of the design files. For further details, refer to the `readme.txt` file.

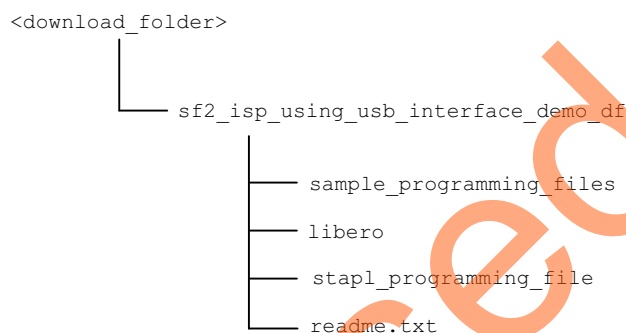


Figure 1 • Demo Design Files Top Level Structure

Figure 2 describes the demo. The SmartFusion2 device application configures the following:

- The MMUART_1 peripheral for serial communication with host PC.
- The USB OTG as mass storage class host, which can read or write files from the mass storage device connected to the SmartFusion2 device through USB cable with Micro A end.

The SmartFusion2 device also initializes the system controller to run the ISP service. The SmartFusion2 device detects the connected USB mass storage device and accesses the programming files. The Cortex-M3 processor reads 512 byte blocks of the programming file data from the USB mass storage device and sends the received blocks of data to the system controller ISP service. The system controller ISP service executes the ISP operation in the requested mode and reports the status to the Cortex-M3

processor. Refer to the "Demo Design Description" section on page 8 for information on the various modes of operation.

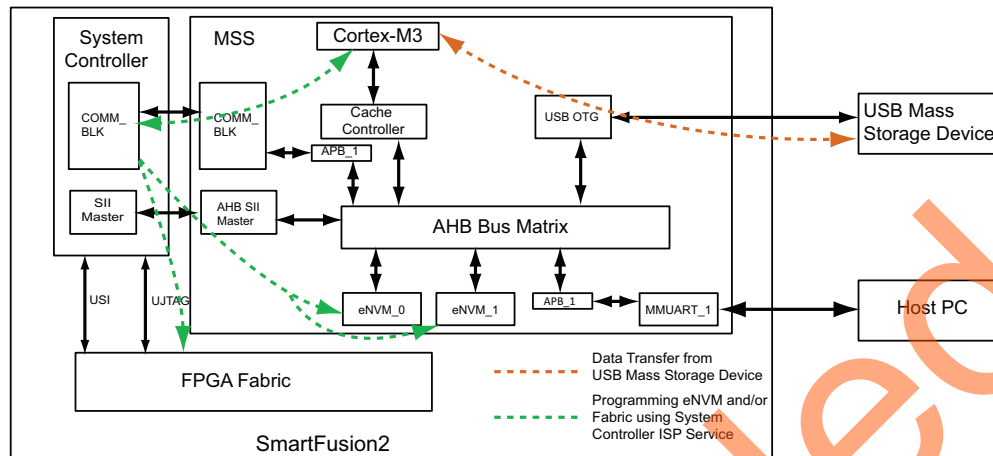


Figure 2 • Top Level Demo Diagram

There are two ways to connect the host PC to the SmartFusion2 device:

- Using the USB-RS232 Serial adapter or the RS232 cable.
 - Refer to the "Running the Demo Design" section on page 12.
- Using the USB to UART (FTDI) interface.
 - Refer to the "Appendix 1: Connecting the SmartFusion2 Device to the Host PC Through the USB to UART (FTDI) Interface" on page 24.

Demo Design Features

This demo design performs three types of programming based on the input provided by the programming file.

- **eNVM programming:** The ISP programming service programs only eNVM. In this case the input programming file has only eNVM content.
- **FPGA fabric programming:** The ISP programming service programs only the FPGA fabric. In this case the input programming file has only the FPGA fabric content.
- **eNVM and FPGA fabric programming:** The ISP programming service programs both the FPGA fabric and eNVM. In this case the input programming file has both the FPGA fabric and eNVM content.

Demo Design Description

The ISP in SmartFusion2 devices is performed by the Cortex-M3 processor and the system controller. The system controller manages the SmartFusion2 device programming and handles the system service requests. The SmartFusion2 device allows the Cortex-M3 processor to directly provide a bitstream to the system controller for programming. The Cortex-M3 processor initializes the system controller and receives the programming bitstream from the USB mass storage device through the USB OTG controller interface. The received bitstream is sent to the system controller to execute the ISP service in one of the following modes of operation:

- **Authenticate:** System controller ISP service validates the integrity of the input data bitstream and reports the status information to the Cortex-M3 processor.
 - For security and reliability reasons, Microsemi recommends that the bitstream is authenticated before the program is executed using Authenticate Operation mode. The SmartFusion2 device application must commit only the bitstream for programming after successful authentication and the integrity of the bitstream is validated.
- **Program:** System controller ISP service programs the following depending on the input data bitstream:
 - eNVM
 - FPGA fabric
 - Both the eNVM and the FPGA fabric
- **Verify:** System controller ISP service verifies the contents of the SmartFusion2 device against the input data bitstream and reports the status information to the Cortex-M3 processor.

The system controller ISP service utilizes the COMM_BLK interface to receive the entire programming data bitstream as a continuous stream of bytes. Refer to the [SmartFusion2 Microcontroller Subsystem User Guide](#) for more information on communication block (COMM_BLK).

The Cortex-M3 processor in the SmartFusion2 device can execute an application image from embedded SRAM (eSRAM), eNVM or DDR/SDR memories. Refer to the [SmartFusion2 SoC FPGA Remapping eNVM, eSRAM, and DDR/SDR SDRAM Memories Application Notes](#) for more information on remapping techniques. In this demo design, the Cortex-M3 processor executes the ISP application image from eSRAM while the eNVM programming taking place, that is during Program operation mode. In order to execute the application image from eSRAM, the Cortex-M3 processor copies the ISP application image (resides in eNVM data client) to the eSRAM and remaps the eSRAM to the Cortex-M3 processor code region. For Verify and Authenticate operation modes, the application image can be executed from either eNVM or eSRAM since the eNVM programming is not taking place.

Programming Files

Sample programming files with the file extension `.spi` are provided to program:

- eNVM
- FPGA Fabric
- Both the eNVM and the FPGA fabric

The folder `<download_folder>\sf2_isp_using_usb_interface_demo_df\sample_programming_files` contains the following sample programming files along with Libero design files.

- `envmonly.spi`: Programs only eNVM. The eNVM client has a simple message display program.
- `fabriconly.spi`: Programs only the FPGA fabric. The FPGA fabric has a light-emitting diode (LED) blinking logic.
- `fabenvm.spi`: Programs both the FPGA fabric and eNVM. The eNVM client has a message display program and the FPGA fabric has an LED blinking logic.
- `isp_demo.spi`: This is the .spi file format version of `isp_demo.stp` file provided in `<download_folder>\sf2_isp_using_usb_interface_demo_dflstapl_programming_file`.

Note: For more information on generating .spi programming files refer to the "Appendix 6: Generating .spi Programming File using Libero" on page 29.

ISP Execution Flow

Figure 3 on page 10 describes the ISP flow. The SmartFusion2 Device application initially configures the USB OTG controller in Host mode and MMUART_1 for serial communication. It also initializes the system controller to start the ISP service in the selected operation mode.

On receiving the ISP operation mode and the programming file index, the application starts reading the input source programming file in a 512 byte blocks. The application stores the received data in a temporary buffer and sends the same data to the ISP service. The application requests the next block of 512 byte data until the entire file gets transferred from the USB mass storage device. The SmartFusion2 Device application is notified with a status code when the ISP service completes the authentication or the verification process. When the operation mode is Program, an internal device reset is generated for the new design to take effect.

Superseded

Figure 3 shows the ISP execution flow.

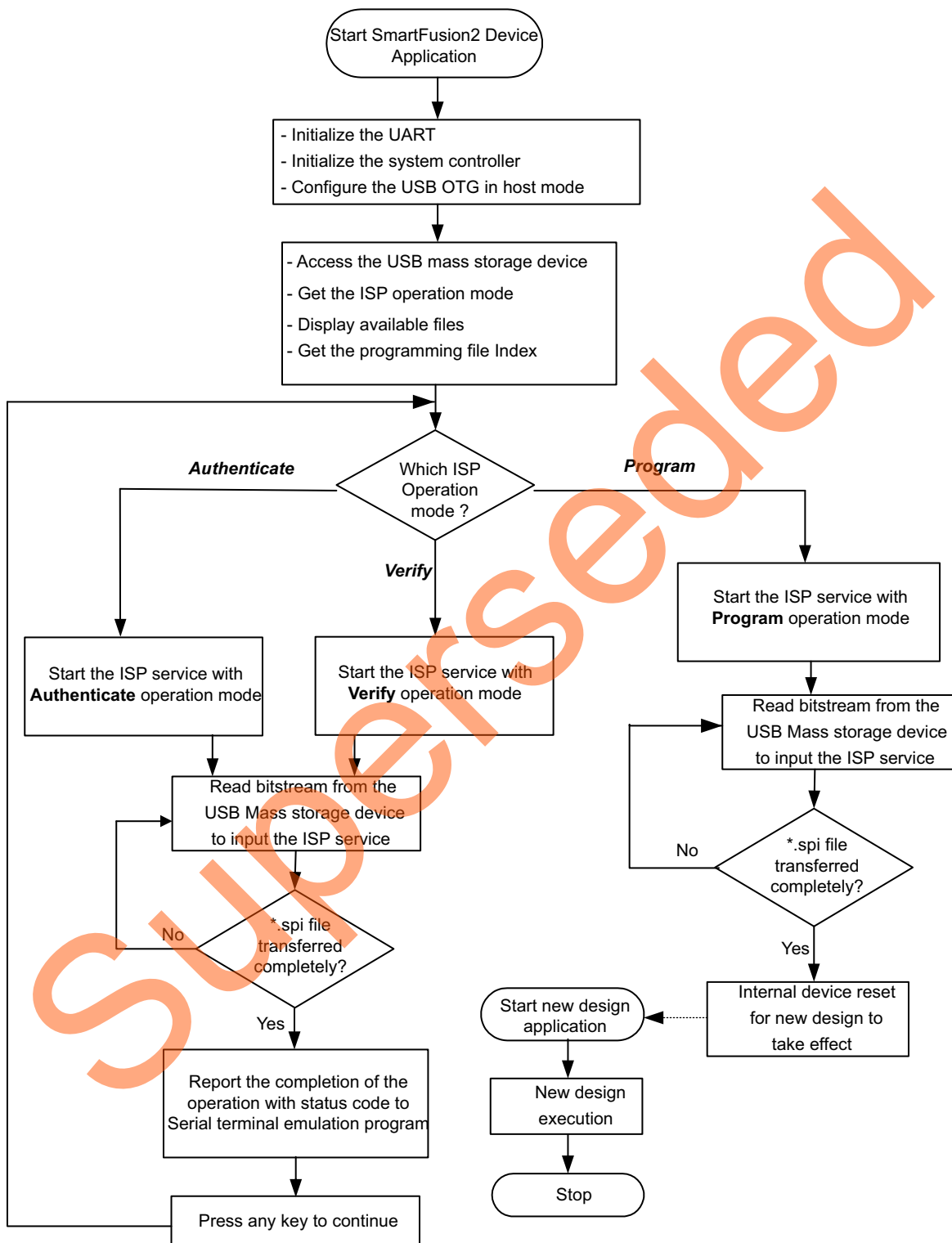


Figure 3 • Execution Flow of ISP Operation Mode

Setting Up the Demo Design

1. Connect the FlashPro4 programmer to the J59 connector of the SmartFusion2 Development Kit board.
2. Connect the host PC to the DB9-RS232 connector provided on the SmartFusion2 Development Kit board using the USB-RS232 serial adapter cable or the RS232 cable.

When using USB-RS232 serial adapter cable, make sure that the USB-RS232 serial adapter drivers are automatically detected. [Figure 4](#) shows an example **Device Manager** window that has the **USB-to-Serial Comm Port** listed under **Ports (Comm & LPT)**. COM port number is required to run the demo design, so make a note of it.

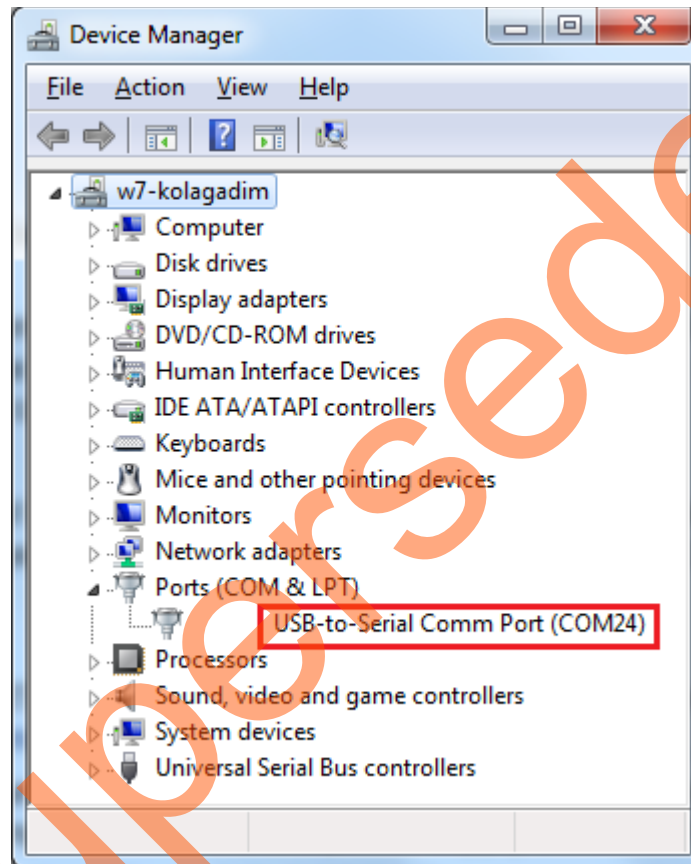


Figure 4 • USB-to-Serial Communication Port

3. Connect the jumpers on the SmartFusion2 Development Kit board, as described in [Table 2](#). For information on jumper locations, refer to the ["Appendix 4: Jumper Locations"](#) on page 27.
 - **Caution:** Before making the jumper connections, switch off the power supply switch, SW7.

Table 2 • SmartFusion2 Development Kit Jumper Settings

Jumper Number	Settings	Notes
J70, J93, J94, J117, J123, J142, J157, J160, J167, J225, J226, J227	1-2 closed	These are the default jumper settings of the Development Kit board. Make sure these jumpers are set accordingly.
J2	1-3 closed	
J23	2-3 closed	
J139	1-2 closed	Jumper to select USB reset.
J163	1-2 closed	Jumper to select USB OTG mode of operation.
J164	1-2 closed	Jumper to provide the VBUS supply to USB when using in Host mode.
J188, 197	1-2 closed	<ul style="list-style-type: none"> • Jumper settings when using MMUART_1. These are not set by default and must be set manually. • Leave these jumpers open if the USB to UART (FTDI) interface is used. Refer to "Appendix 3: Board Setup Through the USB to UART (FTDI) Interface using the USB A to Mini - B Cable".

4. Connect the power supply to the J18 DC jack.

Alternate Setup

This demo can also be run using the USB to UART (FTDI) interface without using the USB-RS232 serial adapter or the RS232 cable. Refer to ["Appendix 1: Connecting the SmartFusion2 Device to the Host PC Through the USB to UART \(FTDI\) Interface"](#) on page 24 for information on how to connect the host PC to the SmartFusion2 Development Kit board for serial communication through the FTDI USB interface using the USB A to Mini - B cable.

Board Setup Snapshot

Snapshots of the SmartFusion2 Development Kit board with all the setup made in both types of connections are given in the following appendices:

- ["Appendix 2: Board Setup when using the USB-RS232 Serial Adapter or RS232 Cable"](#) on page 25
- ["Appendix 3: Board Setup Through the USB to UART \(FTDI\) Interface using the USB A to Mini - B Cable"](#) on page 26

Running the Demo Design

1. Download the demo design from:
http://soc.microsemi.com/download/rsc/?f=sf2_isp_using_usb_interface_demo_df.
2. Switch **ON** the SW7 power supply switch.
3. Start any serial terminal emulation program such as:
 - HyperTerminal
 - PuTTY
 - Tera Term

The configuration for the program is:

- Baud Rate: 57600
- 8 Data bits
- 1 Stop bit
- No Parity
- No Flow Control

For information on configuring the serial terminal emulation programs, refer to the [Configuring Serial Terminal Emulation Programs Tutorial](#).

4. Connect the USB cable Micro A end to the P1 connector of SmartFusion2 Development Kit board and other end to the USB mass storage device.

The USB OTG controller in Host mode is tested to work with mass storage devices listed below:

- Sandisc CruzeerBlade - 16Gb/8Gb/4Gb/1GB
- Kingston DataTraveller - 4Gb/2Gb
- Kingston DT109B - 8GB
- Transcend JetFlash - 4Gb

Make sure to connect preformatted USB Flash drive to the SmartFusion2 device with the sample programming files provided in

<download_folder>\sf2_isp_using_usb_interface_demo_dfsample_programming_files.

5. Launch the FlashPro software.
6. Click **New Project**.
7. In the **New Project** window, type the project name.

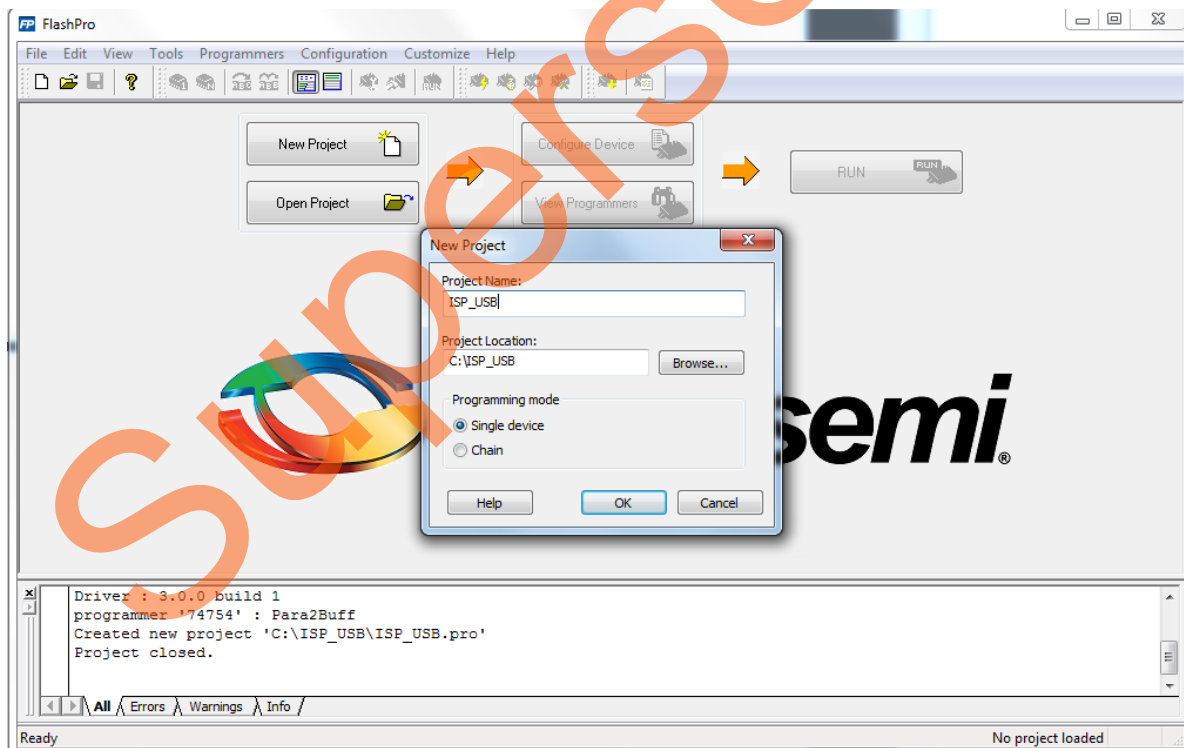


Figure 5 • FlashPro New Project

8. Click **Browse** and navigate to the location where you want to save the project.
9. Select **Single device** as the **Programming mode**.
10. Click **OK** to save the project.

11. Click **Configure Device** on the FlashPro GUI.
12. Click **Browse** and navigate to the location where the `isp_demo.stp` file is located and select the file. The default location is:
<download_folder>\sf2_isp_using_usb_interface_demo_df\stapl_programming_file. The required programming file is selected and is ready to be programmed in the device.

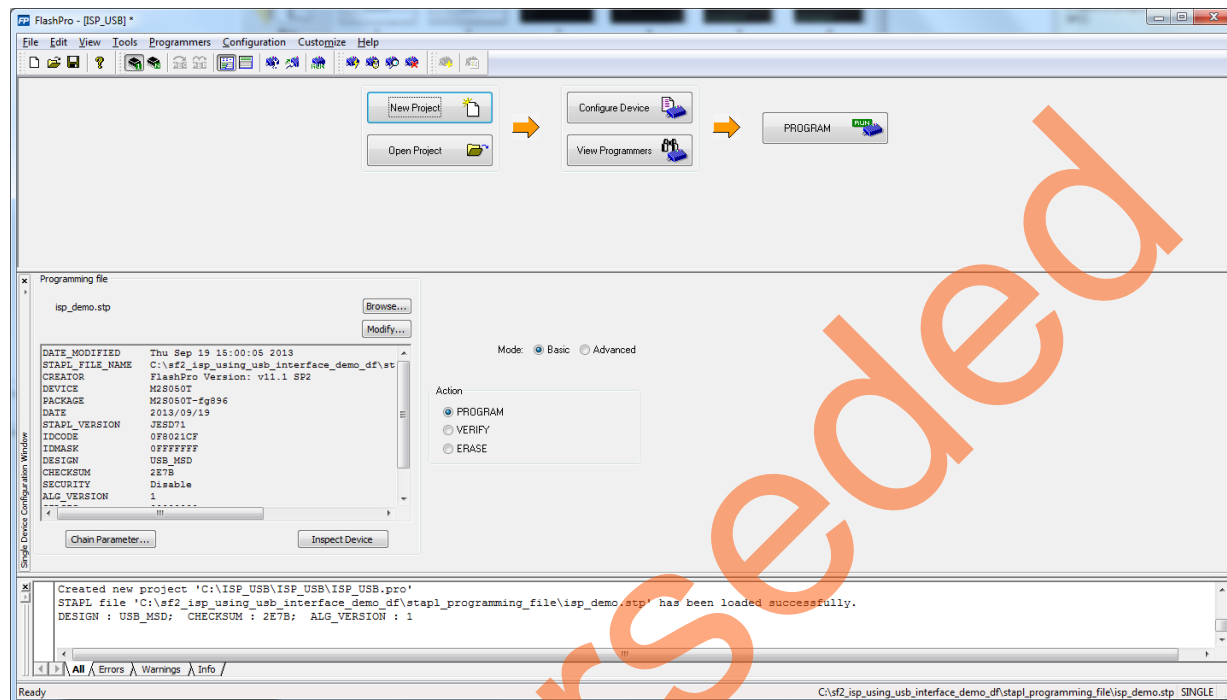


Figure 6 • FlashPro Project Configured

13. Click **PROGRAM** to start programming the device. Wait until you get a message indicating that the program passed. ISP requires the SmartFusion2 device to be preprogrammed with the application code to activate the ISP service. So, the SmartFusion2 device is preprogrammed with the `isp_demo.stp` using FlashPro software.

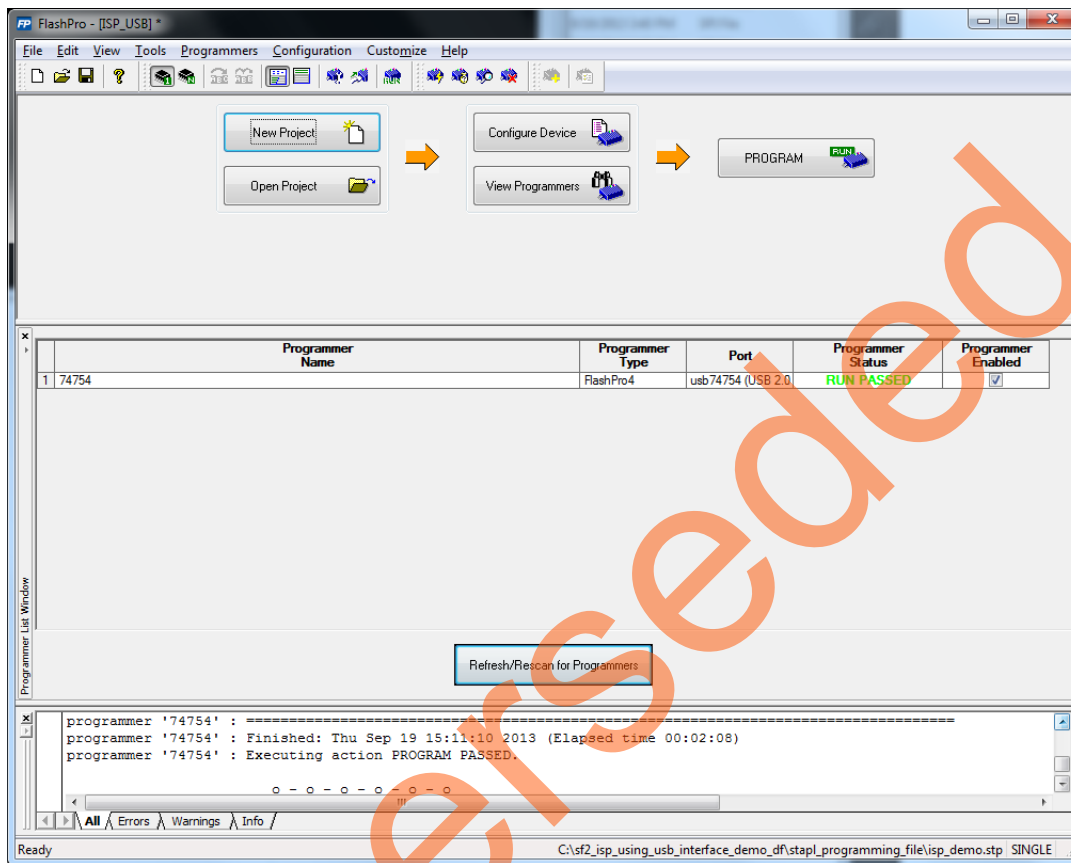
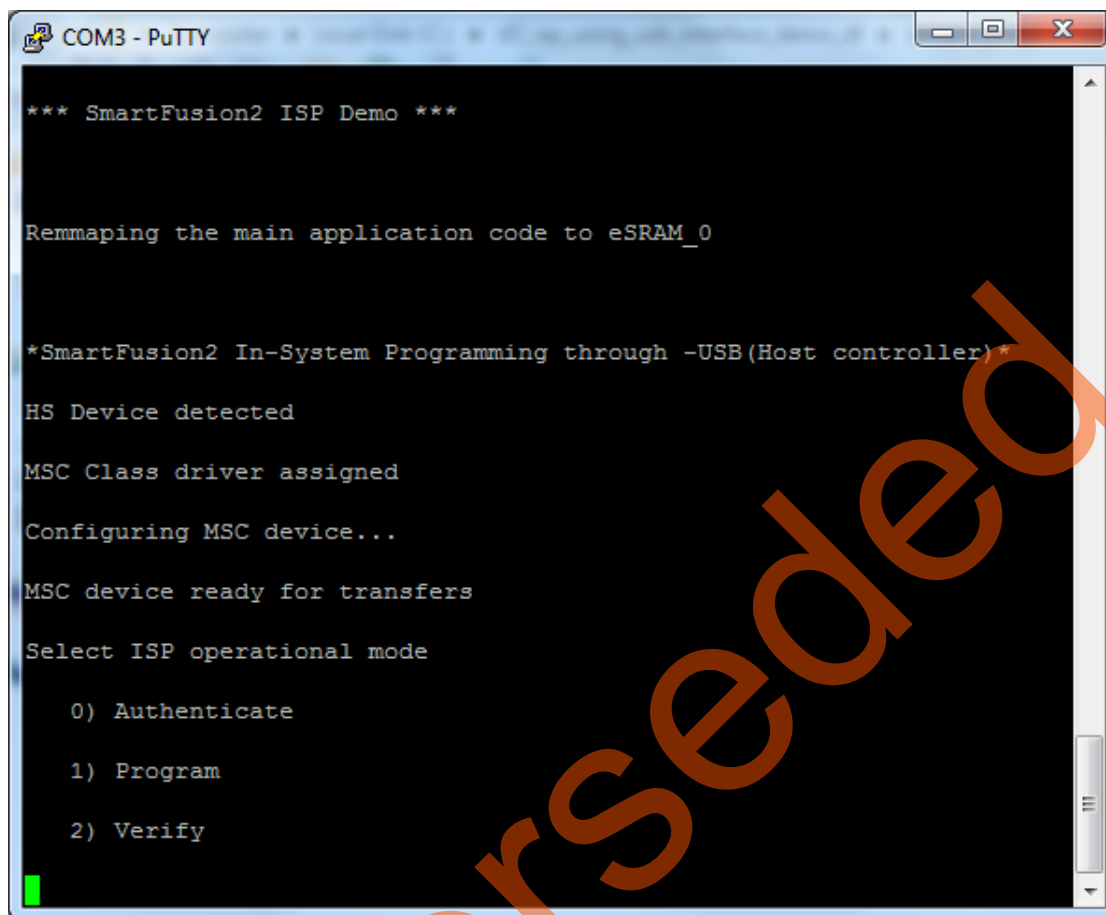


Figure 7 • FlashPro Program Passed

- LEDs 5 to 8 blinking in the board indicates that the SmartFusion2 device fabric is preprogrammed successfully.

On programming the SmartFusion2 device successfully using FlashPro, the serial terminal emulation program shows the initialization messages and ISP operation modes as shown in [Figure 8 on page 16](#).



```
*** SmartFusion2 ISP Demo ***

Remapping the main application code to eSRAM_0

*SmartFusion2 In-System Programming through -USB(Host controller)*

HS Device detected
MSC Class driver assigned
Configuring MSC device...
MSC device ready for transfers
Select ISP operational mode
    0) Authenticate
    1) Program
    2) Verify
```

Figure 8 • ISP Operation Modes Selection

14. On selecting the operation mode, the files in the USB storage device are displayed as shown in Figure 9.

Note: Maximum of 10 files can be shown from the USB storage device.

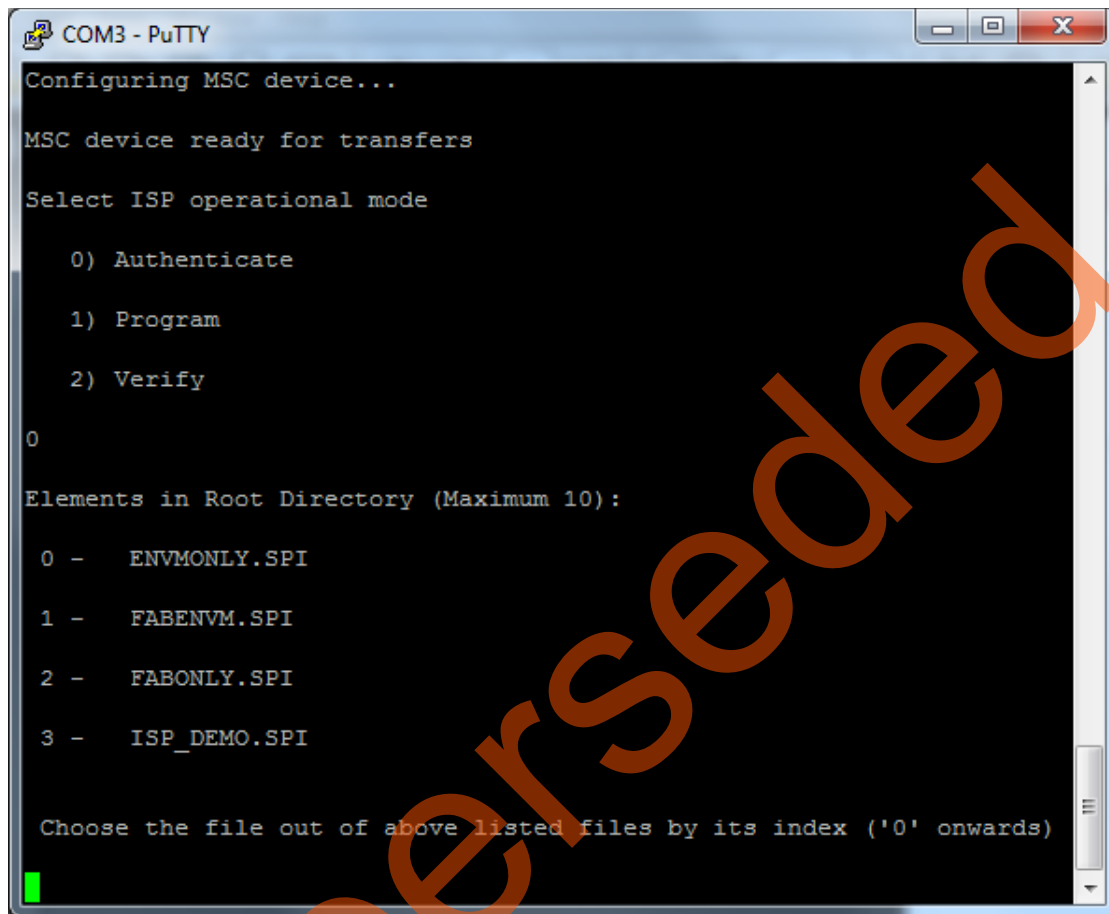


Figure 9 • Available Files in USB Mass Storage Device

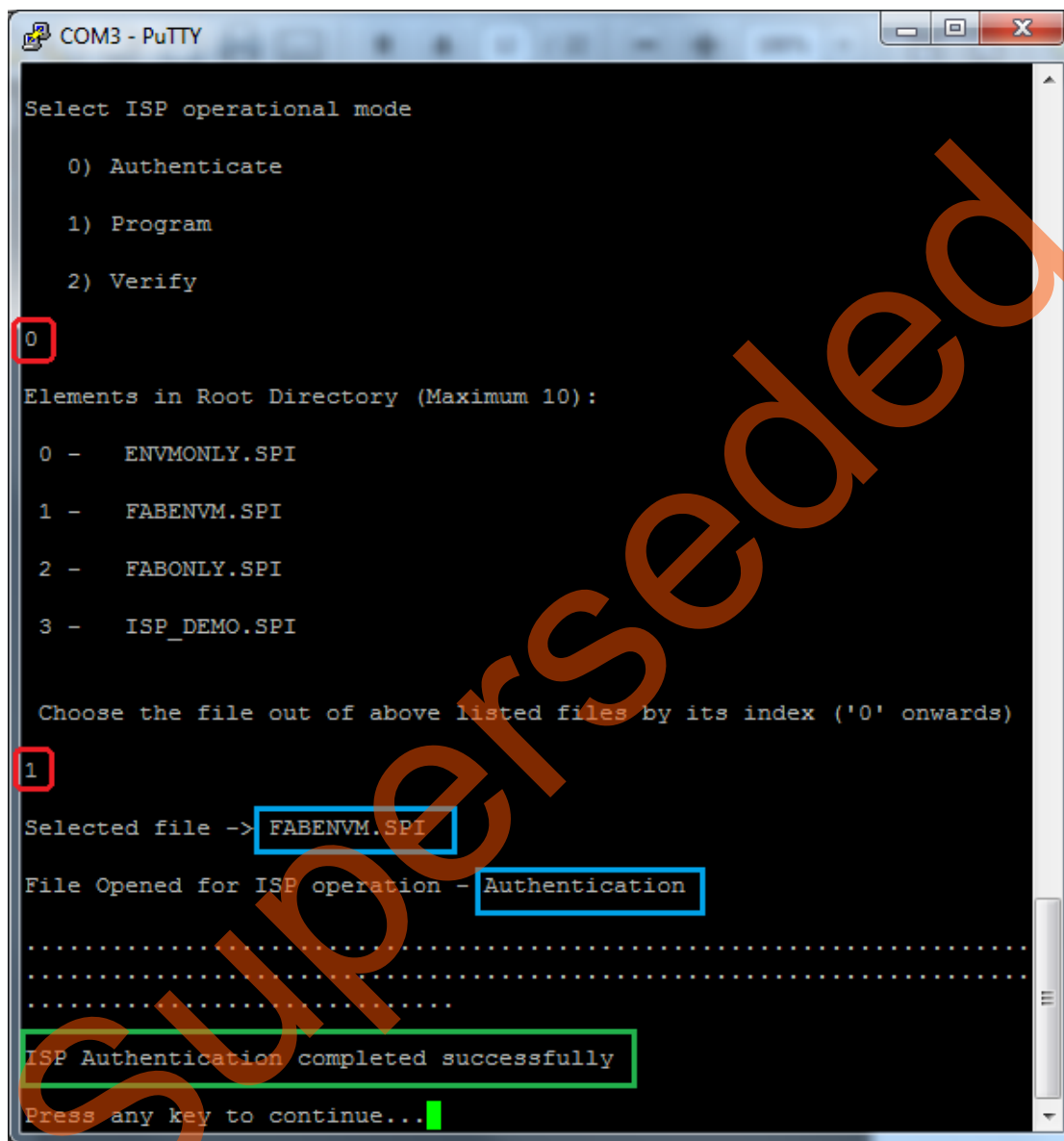
15. Select the programming file from the listed files by its index to perform the selected ISP operation mode.

Authenticate Operation Mode

To authenticate the data from `fabenvm.spi`, enter:

1. **0** to select **Authenticate** operation mode under **Select ISP operation mode**.
2. The corresponding index number to select `fabenvm.spi` programming file.

On selecting the programming file, the application starts reading the programming file from USB mass storage device to execute the ISP operation mode. On completion of the ISP authentication, the serial terminal emulation program displays an operation success message. Figure 10 shows the operation success message.



```
COM3 - PuTTY

Select ISP operational mode

  0) Authenticate
  1) Program
  2) Verify

0

Elements in Root Directory (Maximum 10):

  0 - ENVMONLY.SPI
  1 - FABENV.M.SPI
  2 - FABONLY.SPI
  3 - ISP_DEMO.SPI

Choose the file out of above listed files by its index ('0' onwards)

1

Selected file -> FABENV.M.SPI
File Opened for ISP operation - Authentication
.....
ISP Authentication completed successfully
Press any key to continue...
```

Figure 10 • ISP Authentication Results

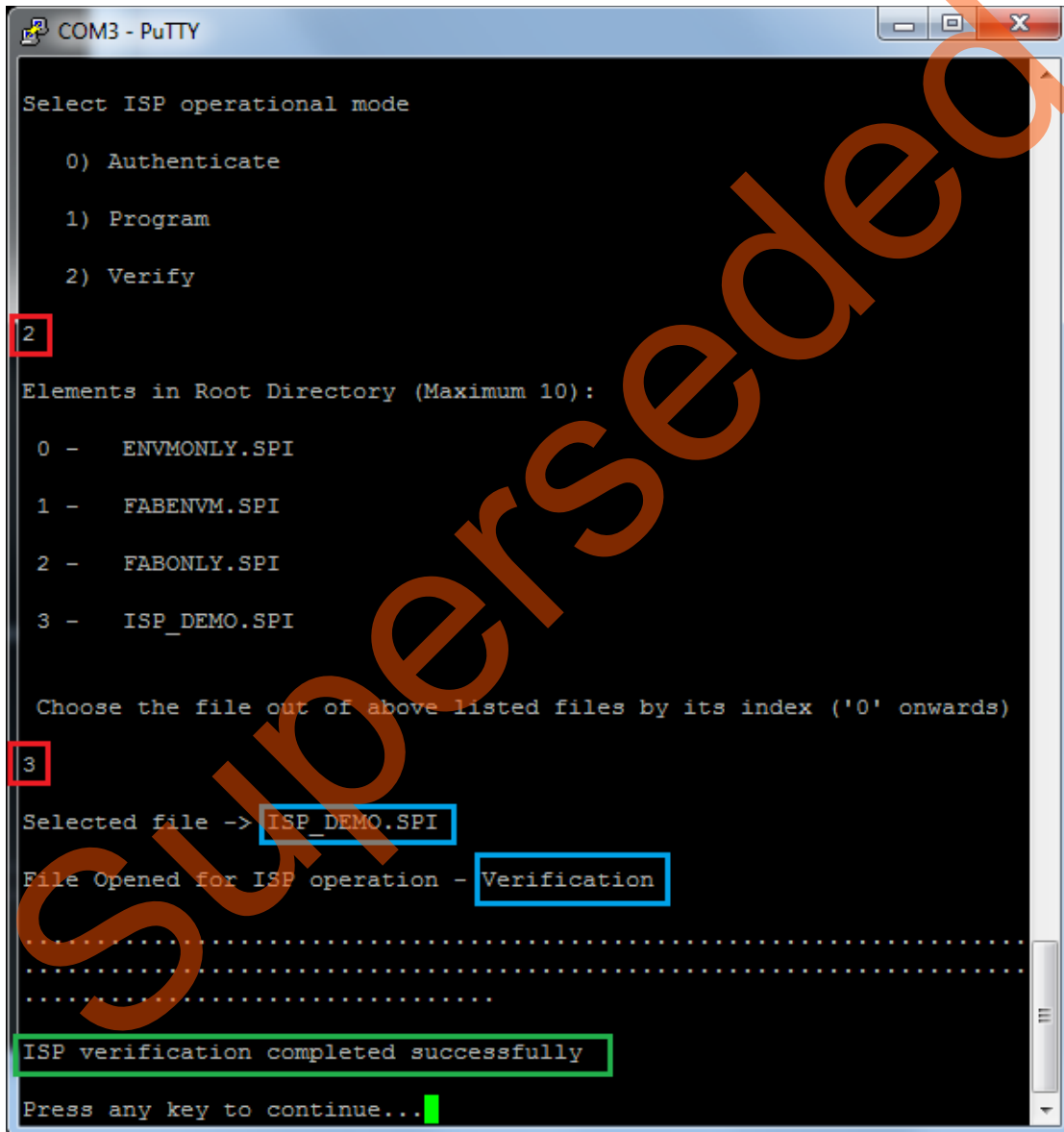
3. Enter any value to try other ISP operation modes. If the USB storage device files are not displayed on serial terminal emulation program, press **SW9** to reset the board.

Verify Operation Mode

To verify the device FPGA fabric and eNVM contents, enter:

1. **2** to select **Verify** operation mode under **Select ISP operation mode**.
2. The corresponding index number to select `isp_demo spi` programming file.

On selecting the programming file, the application starts reading the programming file from USB mass storage device to execute the ISP operation mode. On completion of the ISP verification, the serial terminal emulation program displays an operation success message. Figure 10 shows the operation success message.



```

COM3 - PuTTY

Select ISP operational mode

  0) Authenticate
  1) Program
  2) Verify
2
Elements in Root Directory (Maximum 10):

  0 - ENVMONLY.SPI
  1 - FABENVM.SPI
  2 - FABONLY.SPI
  3 - ISP_DEMO.SPI

Choose the file out of above listed files by its index ('0' onwards)
3
Selected file -> ISP_DEMO.SPI
File Opened for ISP operation - Verification
.....
ISP verification completed successfully
Press any key to continue...
  
```

Figure 11 • ISP Verification Results

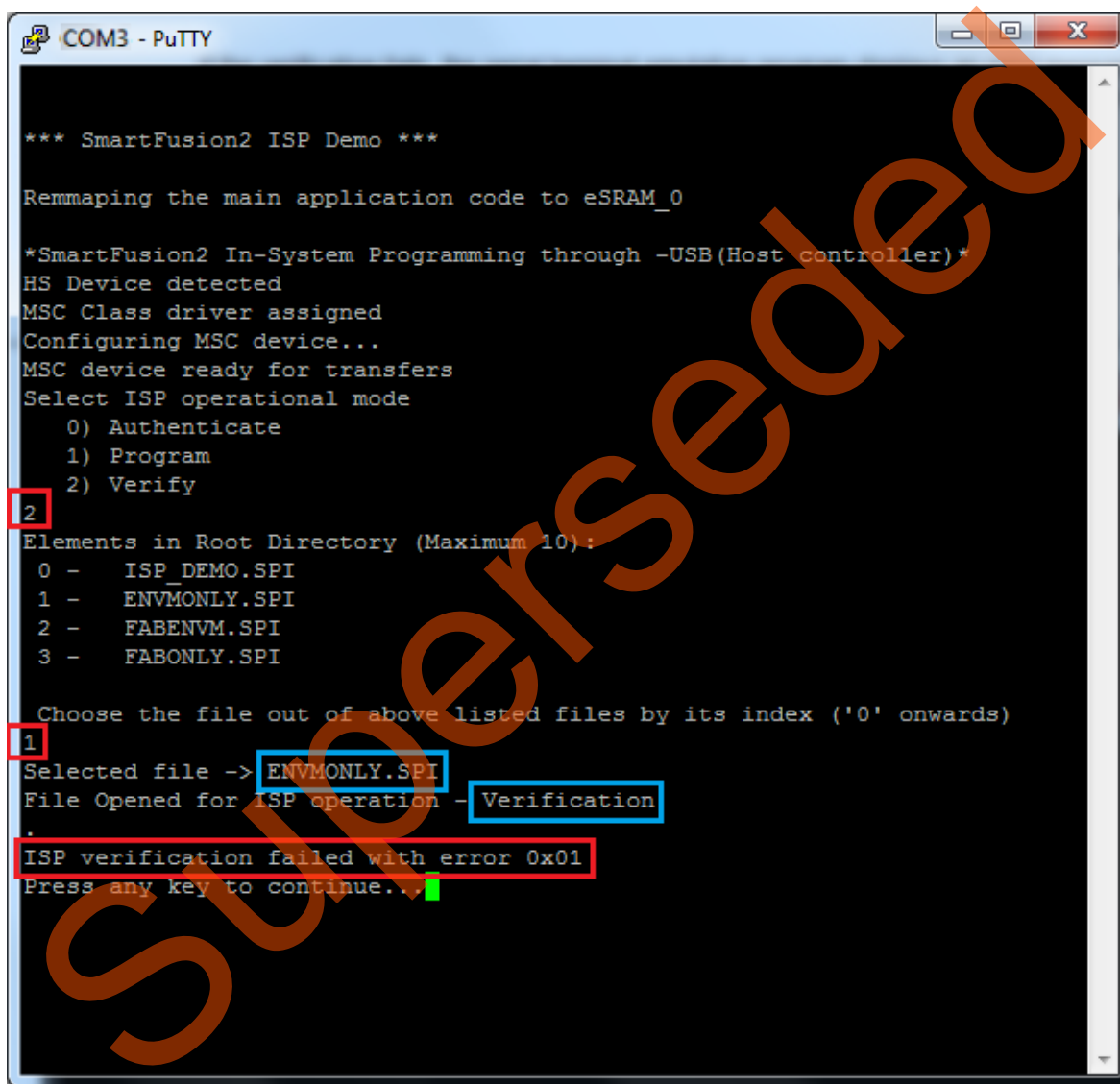
The verification operation demonstrated is for the `isp_demo.stp` file that is already running in the SmartFusion2 device. If any other `.spi` file is verified while the `isp_demo.stp` file is still running, that verification operation fails.

If the verification fails, the serial terminal emulation program displays an error message with an error code. Figure 12 shows an example error message. For more information on error codes, refer to the "Appendix 5: Error Codes" on page 28.

The programming files are at:

<download_folder>\sf2_isp_using_usb_interface_demo_df\sample_programming_files.

All of them do not pass the verification. Only the `isp_demo.spi` file passes the verification operation as it matches with the SmartFusion2 device contents (`isp_demo.stp`). The other programming files fail verification.



```
*** SmartFusion2 ISP Demo ***

Remmapping the main application code to eSRAM_0

*SmartFusion2 In-System Programming through -USB(Host controller)*
HS Device detected
MSC Class driver assigned
Configuring MSC device...
MSC device ready for transfers
Select ISP operational mode
  0) Authenticate
  1) Program
  2) Verify
2
Elements in Root Directory (Maximum 10):
0 -  ISP_DEMO.SPI
1 -  ENVMONLY.SPI
2 -  FABENVM.SPI
3 -  FABONLY.SPI

Choose the file out of above listed files by its index ('0' onwards)
1
Selected file -> ENVMONLY.SPI
File Opened for ISP operation - Verification
ISP verification failed with error 0x01
Press any key to continue...
```

Figure 12 • ISP Verification Failure Error Message

3. Enter any value to try other ISP operation modes. If the USB storage device files are not displayed on the serial terminal emulation program, press **SW9** to reset the board.

Program Operation Mode

To program the FPGA fabric and the eNVM of the SmartFusion2 device using the `fabenvm.spi` file, enter:

1. **1** to select **Program** operation mode under **Select ISP operation mode**.
2. The corresponding index number to select `fabenvm.spi` programming file.

On selecting the programming file, the application starts reading the programming file from the USB mass storage device to execute the ISP operation mode. The application checks the data integrity of the selected programming file prior to perform the ISP program operation. Once the programming operation is completed, an internal reset is generated for the new design to take effect.

Figure 13 shows selection of program operation mode for the `fabenvm.spi` programming file.

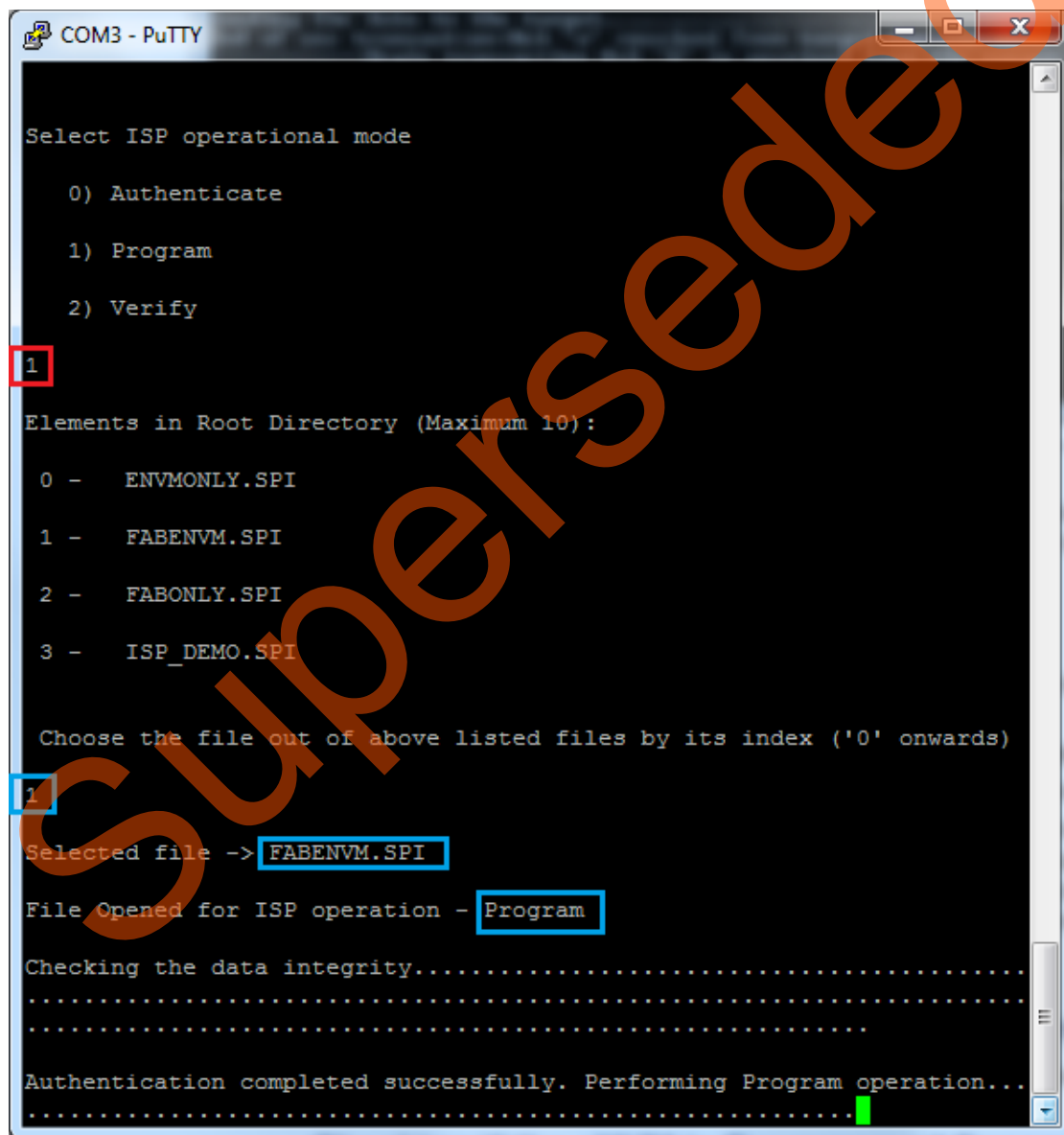


Figure 13 • ISP Program Operation Mode

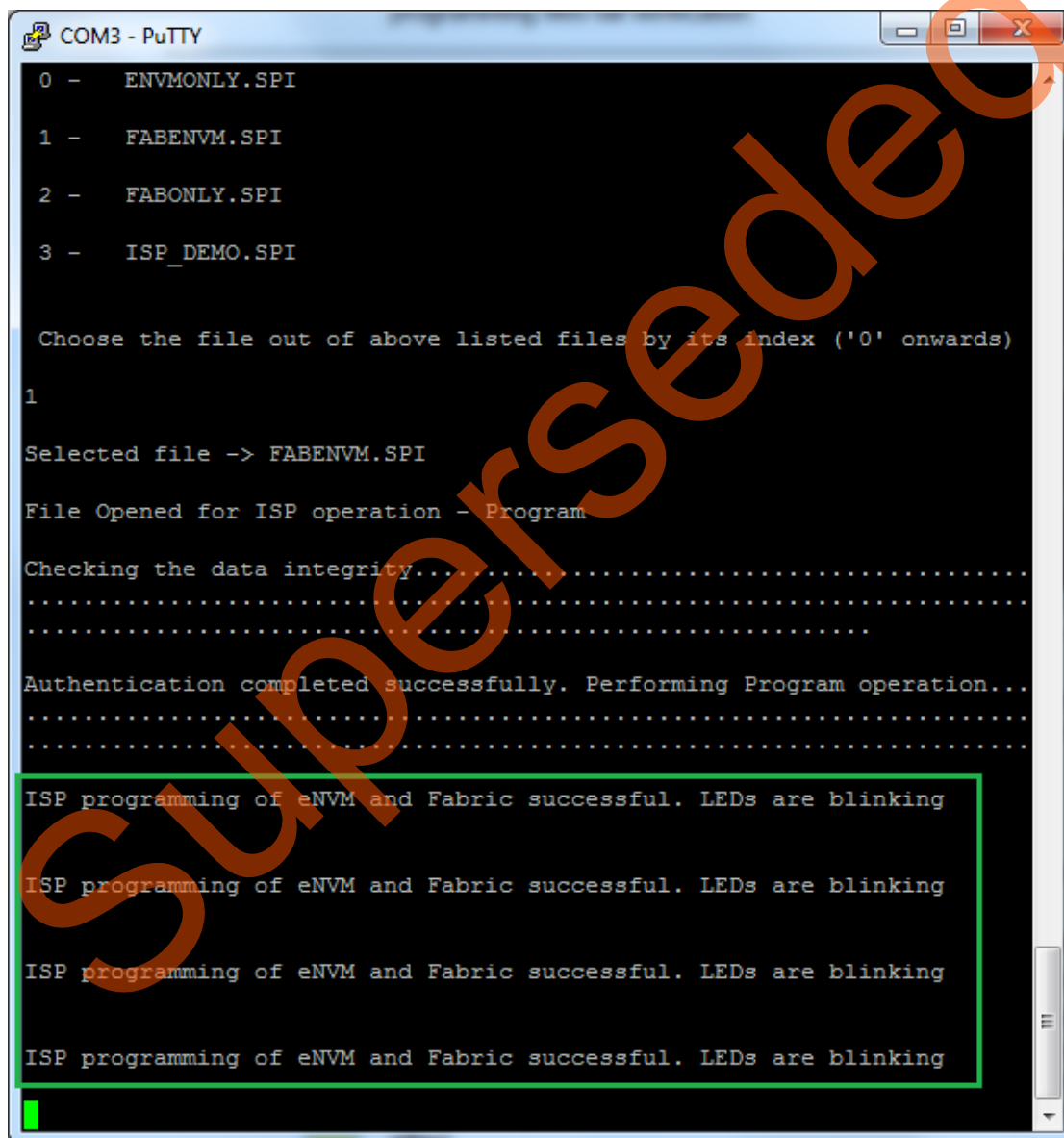
Press **SW9** to reset the SmartFusion2 Development Kit or Power Cycle the SmartFusion2 Development Kit.

Checking if the Fabric is Programmed Successfully

LEDs 1 to 4 blinking in the board indicates that the fabric is programmed successfully.

Checking if the eNVM is Programmed Successfully

The serial terminal emulation program displays the success message as shown in Figure 14 if the eNVM is programmed successfully.



```
COM3 - PuTTY
0 - ENVMONLY.SPI
1 - FABENVM.SPI
2 - FABONLY.SPI
3 - ISP_DEMO.SPI

Choose the file out of above listed files by its index ('0' onwards)
1
Selected file -> FABENVM.SPI
File Opened for ISP operation - Program
Checking the data integrity.....
.....
Authentication completed successfully. Performing Program operation...
.....
ISP programming of eNVM and Fabric successful. LEDs are blinking
ISP programming of eNVM and Fabric successful. LEDs are blinking
ISP programming of eNVM and Fabric successful. LEDs are blinking
ISP programming of eNVM and Fabric successful. LEDs are blinking
```

Figure 14 • ISP Program Results

Programming Results

The result shown in [Figure 14 on page 22](#) is for the `fabenvm spi` file. [Table 3](#) shows the possible results for ISP Program operation mode for sample programming files provided in folder `<download_folder>\sf2_isp_using_usb_interface_demo_df\sample_programming_files`. Not all `.spi` files listed in the table are demonstrated.

Table 3 • ISP Programming Results

*.spi Programming File Name	eNVM Programming Result	FPGA Fabric Programming Result
<code>envmonly spi</code>	The serial terminal emulation program shows successful eNVM program message	NA
<code>fabonly spi</code>	NA	SmartFusion2 LEDs 1 to 4 blinks
<code>fabenvm spi</code>	The serial terminal emulation program shows successful eNVM program message	SmartFusion2 LEDs 1 to 4 blinks

After successful ISP Program operation, the SmartFusion2 Development Kit board must be reprogrammed with the original `isp_demo.stp` file to try the ISP operation modes again.

Appendix 1: Connecting the SmartFusion2 Device to the Host PC Through the USB to UART (FTDI) Interface

The following procedure describes how to connect the SmartFusion2 device to the host PC through the USB to UART (FTDI) interface using a USB A to Mini-B Cable for serial communication:

1. Connect the host PC to the J24 connector using the USB A to Mini-B cable.
2. Make sure that the *USB to UART bridge drivers* are automatically detected. Of the four COM ports, select the one with **Location** as *on USB Serial Converter D*. [Figure 15](#) shows an example **Device Manager** window that has the **USB Serial Port** and its properties showing the port number and location. COM port number is required to run the demo design, so make a note of it.

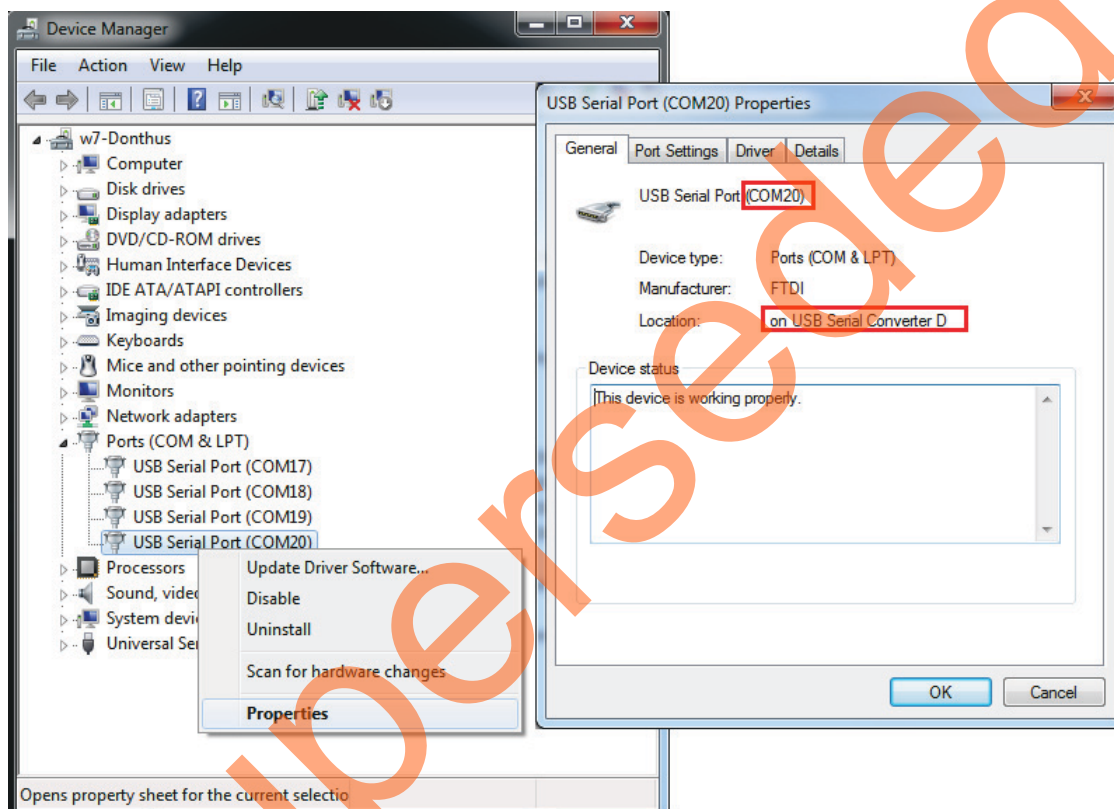


Figure 15 • Device Manager window Showing the USB Serial Port

3. Connect the jumpers as follows:
 - Pin 2 of J197 to pin 3 of J129
 - Pin 2 of J188 to pin 3 of J133
 - [Figure 17 on page 26](#) shows these pin connections.
 - Refer to [Figure 18 on page 27](#) for the location of the jumpers. These connections are required for connecting the MMUART_1 TXD and RXD signals to the FTDI USB to UART bridge available in the SmartFusion2 Development Kit board.

Appendix 2: Board Setup when using the USB-RS232 Serial Adapter or RS232 Cable

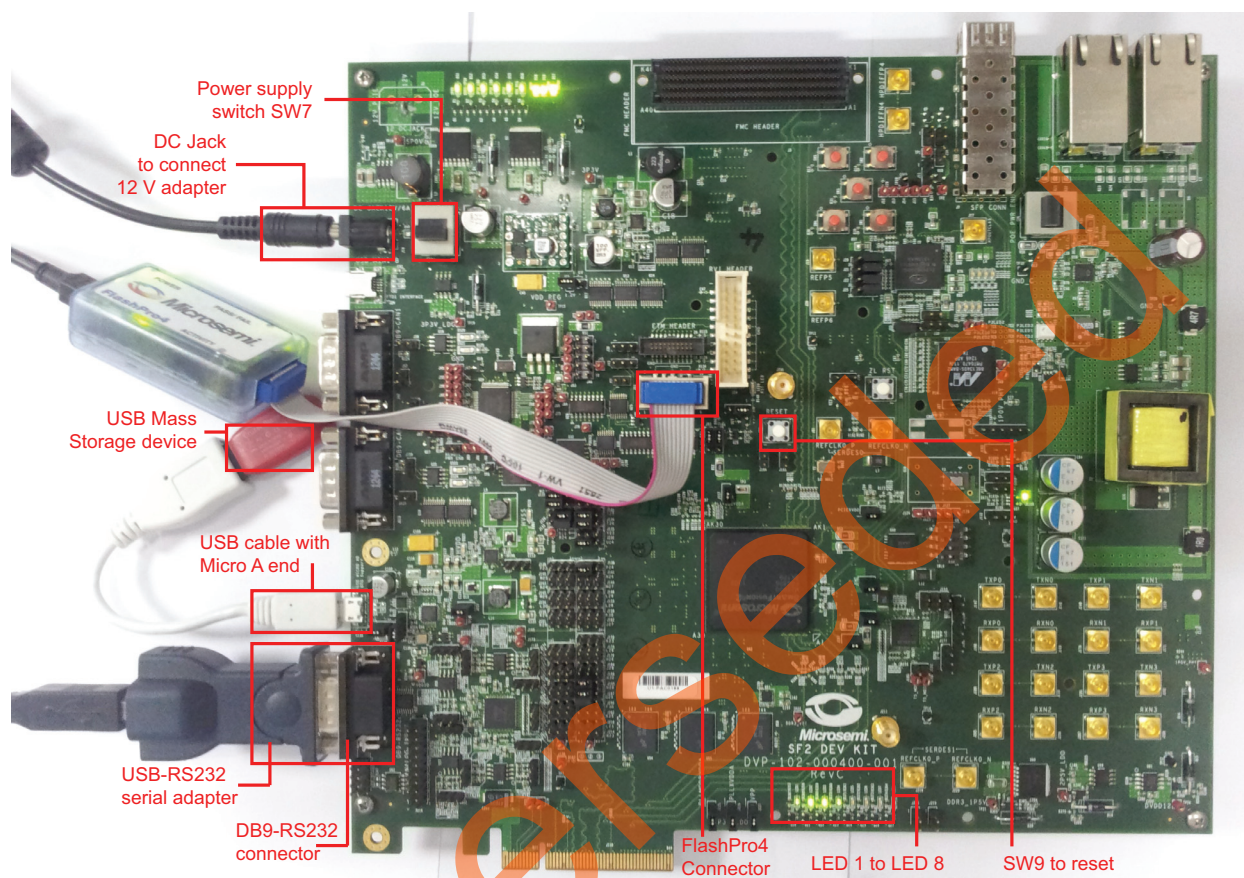


Figure 16 • Board Setup when using the USB-RS232 Serial Adapter or RS232 Cable

Appendix 3: Board Setup Through the USB to UART (FTDI) Interface using the USB A to Mini - B Cable

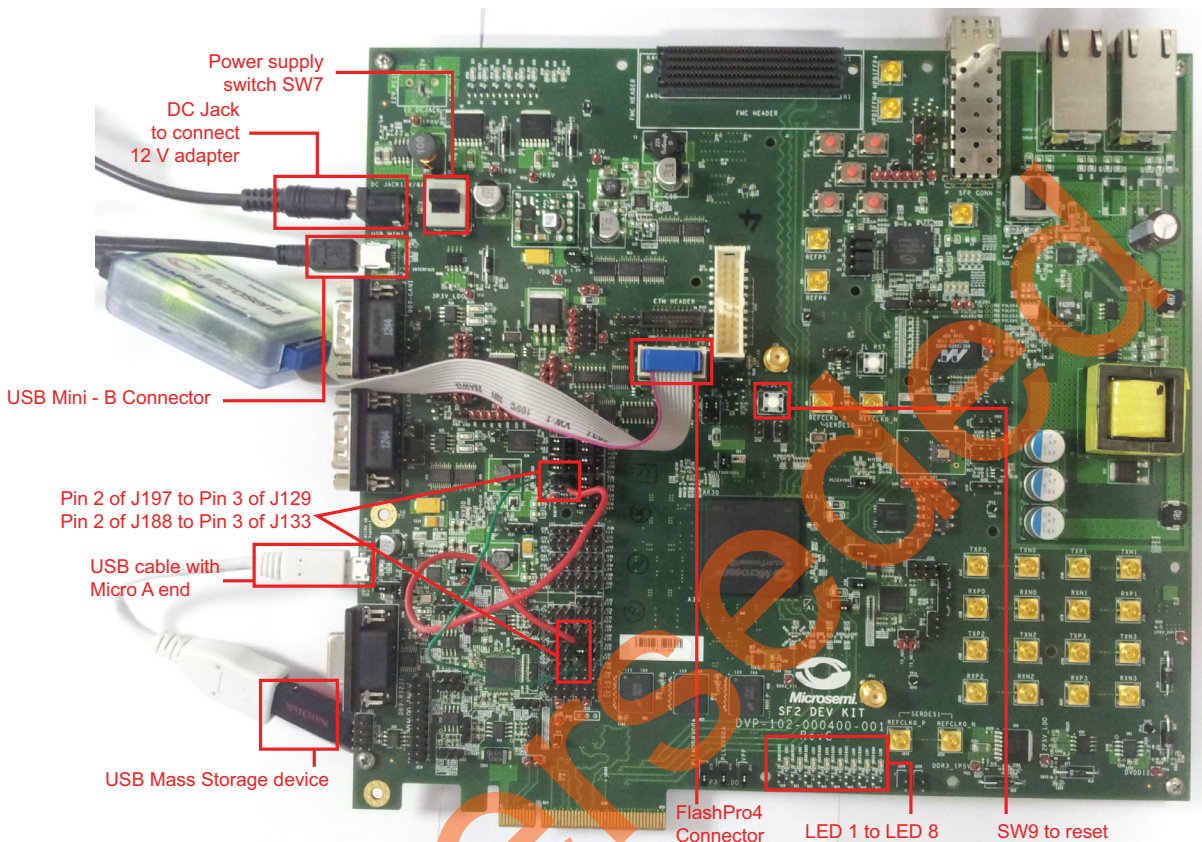


Figure 17 • Board Setup when Through the USB to UART (FTDI) Interface using the USB A to Mini - B Cable



- Jumpers highlighted in red are set by default.
- Jumpers highlighted in green must be set manually.
- The location of the jumpers in [Figure 18 on page 27](#) are searchable.

Appendix 5: Error Codes

Table 4 • Error Codes

Error Code	#define	Description
0x01(1)	#define MSS_SYS_CHAINING_MISMATCH	Device contents mismatch
0x02(2)	#define MSS_SYS_UNEXPECTED_DATA_RECEIVED	Data is not supported
0x03(3)	#define MSS_SYS_INVALID_ENCRYPTION_KEY	Invalid encryption key
0x0B(11)	#define MSS_SYS_UNSUPPORTED_BITSTREAM_PROT_VER	Bitstream is not supported
0x0C(12)	#define MSS_SYS_VERIFY_NOT_PERMITTED_ON_BITSTR	Verification is not allowed for input bitstream
0x7F(127)	#define MSS_SYS_ABORT	Operation aborted
0x81(129)	#define MSS_SYS_NVM_VERIFY_FAILED	eNVM verification failed
0x82(130)	#define MSS_SYS_DEVICE_SECURITY_PROTECTED	Device is secured
0x83(131)	#define MSS_SYS_PROGRAMMING_MODE_NOT_ENABLED	Programming mode is not enabled

Appendix 6: Generating .spi Programming File using Libero

1. Launch the Libero SoC software to open a Libero project for `fabenvm.spi` programming file. The Libero design file is provided in
`<download_folder>\sf2_isp_using_usb_interface_demo_df\sample_programming_file\fabric_and_envm`.
2. Right-click **Export Programming File** under **Handoff Design for Production** in the **Design Flow** tab, and click **Configure Options** from the context menu.

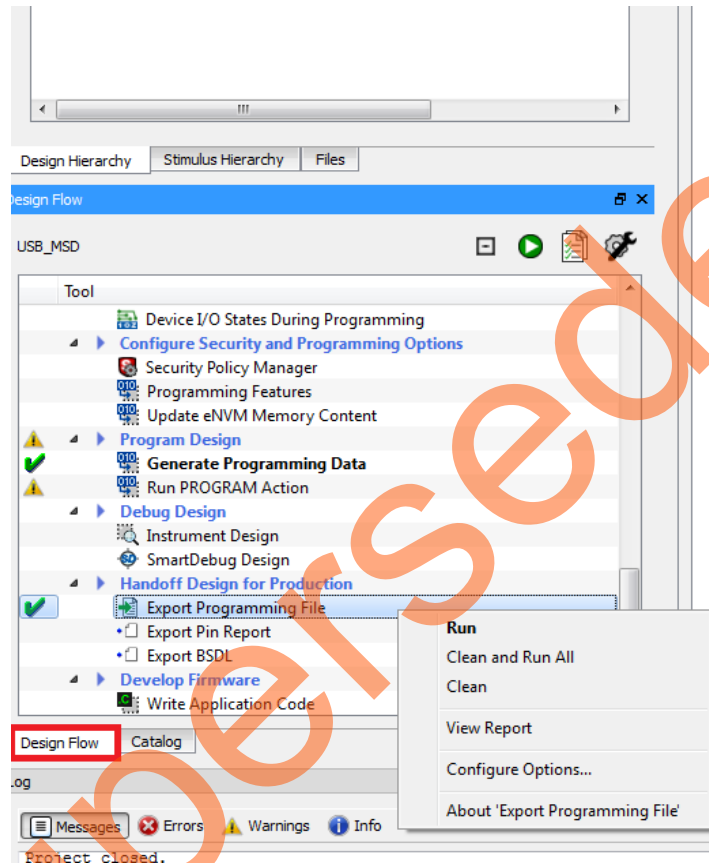


Figure 19 • Configuring Export Programming File

- On the **Export Programming File Options** window, select the **SPI file** check box.

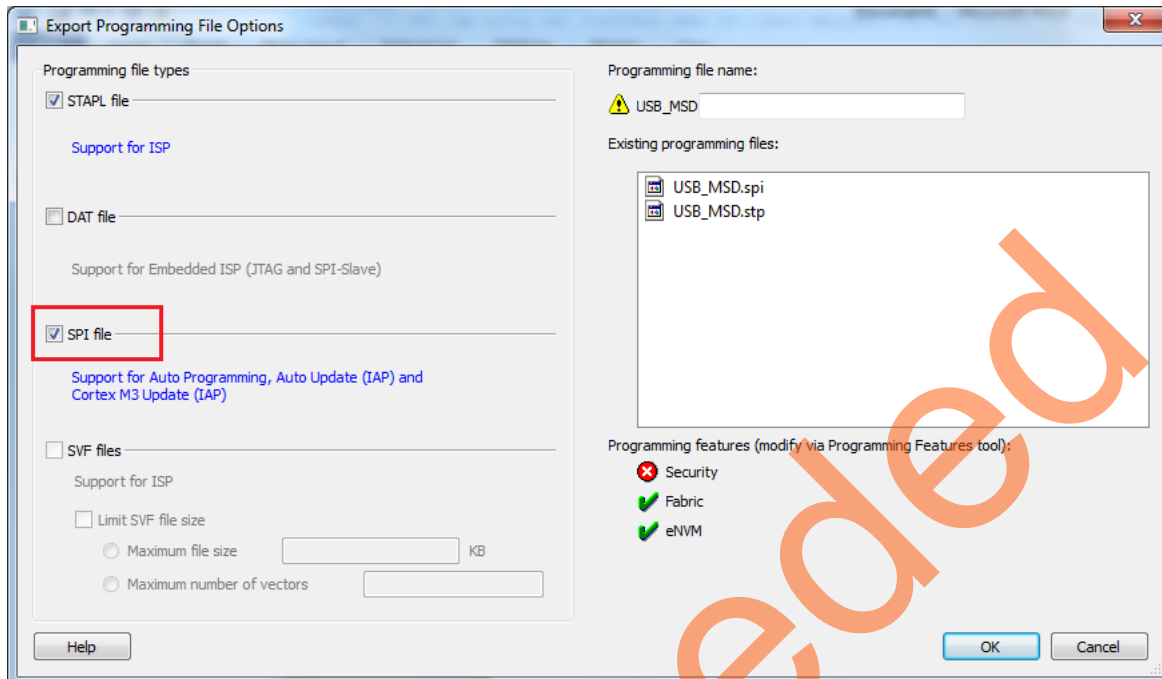


Figure 20 • Export Programming File Options window

- Click **OK**.
- Double-click **Export Programming File** under **Handoff Design for Production** in the **Design Flow** tab to generate the .spi file (Figure 19 on page 29). Figure 21 shows the .spi file location in **Reports** tab.

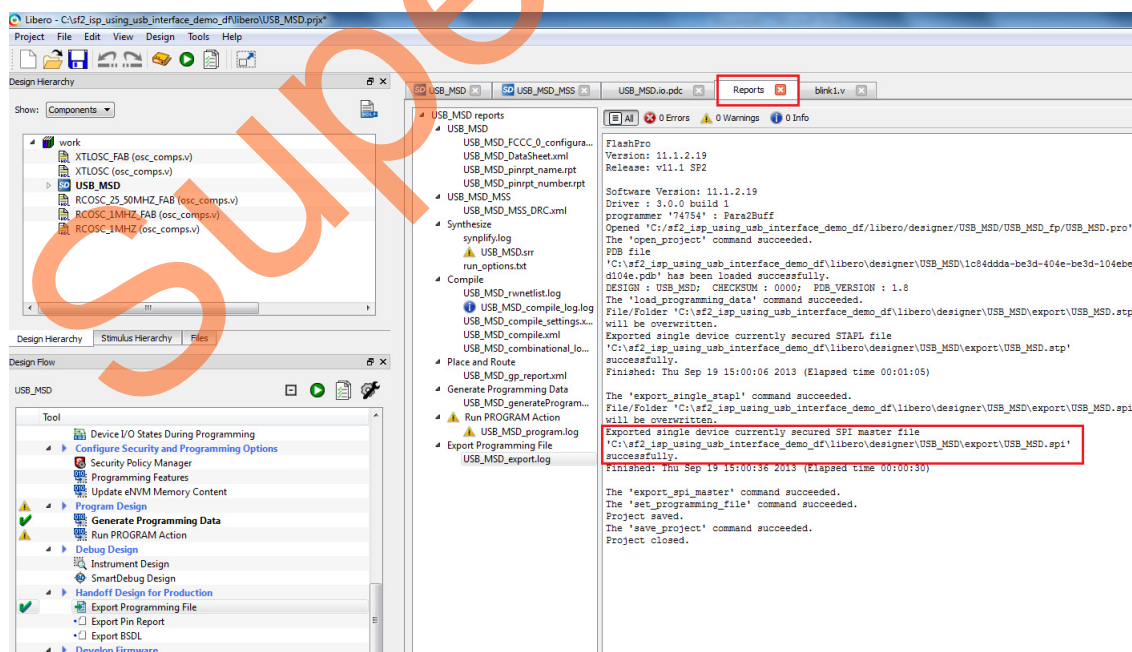


Figure 21 • .SPI File Location

A – List of Changes

The following table lists critical changes that were made in each revision of the chapter in the demo guide.

Date	Changes	Page
Revision 3 (May 2014)	Updated the document for Libero v11.3 software release (SAR 56662).	NA
Revision 2 (December 2013)	Updated " Demo Design Description " section (SAR 53451).	8
Revision 1 (November 2013)	Updated the document for Libero v11.2 software release (SAR 52963).	NA
Revision 0 (October 2013)	Initial release.	NA

Superseded

B – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060

From the rest of the world, call 650.318.4460

Fax, from anywhere in the world, 408.643.6913

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Customer Support website (www.microsemi.com/soc/support/search/default.aspx) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

Website

You can browse a variety of technical and non-technical information on the SoC home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to [My Cases](#).

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. [Sales office listings](#) can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within [My Cases](#), select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the [ITAR](#) web page.

Superseded

Superseded



Microsemi

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and security, aerospace, and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs, and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at www.microsemi.com.

© 2014 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.