



CY3207ISSP

ISSP™ User Guide

Revision C

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intl): 408.943.2600
<http://www.cypress.com>

Copyrights

Copyright © 2002 - 2005 Cypress Semiconductor Corporation. All rights reserved.

Programmable System-on-Chip, PSoC, PSoC Designer, and PSoC Express are trademarks of Cypress Semiconductor Corporation (Cypress), along with Cypress® and Cypress Semiconductor™. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

The information in this document is subject to change without notice and should not be construed as a commitment by Cypress. While reasonable precautions have been taken, Cypress assumes no responsibility for any errors that may appear in this document. No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Cypress. Made in the U.S.A.

Disclaimer

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Contents



1. Introduction	5
1.1 Functionality.....	5
1.2 Kit Contents	5
1.3 ISSP Hardware	6
2. Software Installation	9
2.1 CD Installation	9
2.2 Download Installation.....	9
2.3 Main Screen Operation.....	10
3. Using the ISSP	11
3.1 Socket Programming	11
3.2 In-System Programming	11
3.2.1 5.2.1 ISSP Cable.....	11
3.2.2 5.2.2 Methods	12
3.2.2.1 In-System – Vdd Stable On	12
3.2.2.2 In-System – Vdd Rising Edge.....	12
3.2.3 5.2.3 Access.....	13
3.3 DOS Command Line Controls	13
3.4 Tester I/O Connector	14
3.5 Tester I/O Signaling	15
3.6 Test Counters	15

1. Introduction



The In-System Serial Programming (ISSP) programmer programs PSoC ICs with HEX files created with Cypress' PSoC™ Designer software.

1.1 Functionality

The programmer programs a PSoC chip mounted on your PCB, one at a time. It connects to your PCB with a 5-wire cable and to your PC with a USB cable.

Programming operation can be automated by incorporating the programmer into a PC-based test system. The tester software communicates with the programmer-control software through a command line interface. The ISSP programmer can also be operated manually using the supplied Windows® GUI software.

The ISSP functions are as follows:

- **Program** – Programs a HEX file into a PSoC device.
- **Verify** – Verifies that the data in a PSoC device's Flash memory matches the data in the selected HEX file.
- **Read** – Displays the data from a PSoC device's Flash memory.
- **Flexible Connections to Target PSoC Device** – Devices can be inserted directly into ISSP, connected via a test fixture or programmed in-system on a product circuit board.

For more information on In-System Serial Programming, refer to Application Note AN2014 "Design for In-System Serial Programming (ISSP)" and AN2026 "In-System Serial Programming (ISSP) Protocol." Both are available at <http://www.cypress.com/>.

1.2 Kit Contents

- The ISSP Programmer
- ISSP Software CD-ROM
- 6-Foot USB Cable
- 2-Foot Programming Cable
- 9-Volt Power Supply

1.3 ISSP Hardware



Figure 1-1. Top View of the ISSP

Down the center of the programmer is a graphic showing the orientation of the various DIP packages; pin 1 down and the package justified to the top of the socket.

The ZIF socket accepts the DIP packages directly and surface mount packages via adapter boards. Following is a description of the ISSP buttons and LEDs:

- Along the right side of the programmer, the POWER LED indicates a valid power source is connected.
- The USB ACTIVE LED lights when the USB connection is made.
- The STATUS LED is lit after the initialization sequence completes.
- The PROGRAM button starts a programming sequence.
- The PASS and FAIL LEDs report the results of programming and verification sequences.



Figure 1-2. Left Side View of the ISSP

- 9-Volt Power Connection
- USB Connection



Figure 1-3. Right Side View of the ISSP

- RJ-45 Target Connector
- 10-Pin Tester I/O Connection

2. Software Installation



The software can be installed from the CD-ROM included in the box or from the Cypress Semiconductor web site at www.cypress.com.

2.1 CD Installation

If the ISSP software has previously been installed, you must run the install program twice. The first time uninstalls the previous version. The second run installs the new version. To install the ISSP software, perform the following procedure.

1. Place the CD-ROM in the CD drive. The software installer should start automatically.
If the CD-ROM does not run or to run it a second time, browse to My Computer and double click the CD-ROM icon labeled ISSP 1.80.
2. Follow the on-screen instructions to complete the installation.

2.2 Download Installation

If the ISSP software has previously been installed you must run the install program twice. The first time uninstalls the previous version. The second run installs the new version. To install the ISSP software perform the following procedure.

1. Browse to www.cypress.com.
2. Click on the Software and Drivers entry under the Design Resources heading.
3. Select the PSoC Mixed-Signal Controllers in the Product Group drop down box and click Apply Filters.
4. Scroll down to In-System Serial Programming (ISSP) CY3207ISSP software and select it.
5. Click on the download icon, and unzip and run the install package.

2.3 Main Screen Operation

Once you have installed the ISSP software, the Main ISSP window is available for viewing.

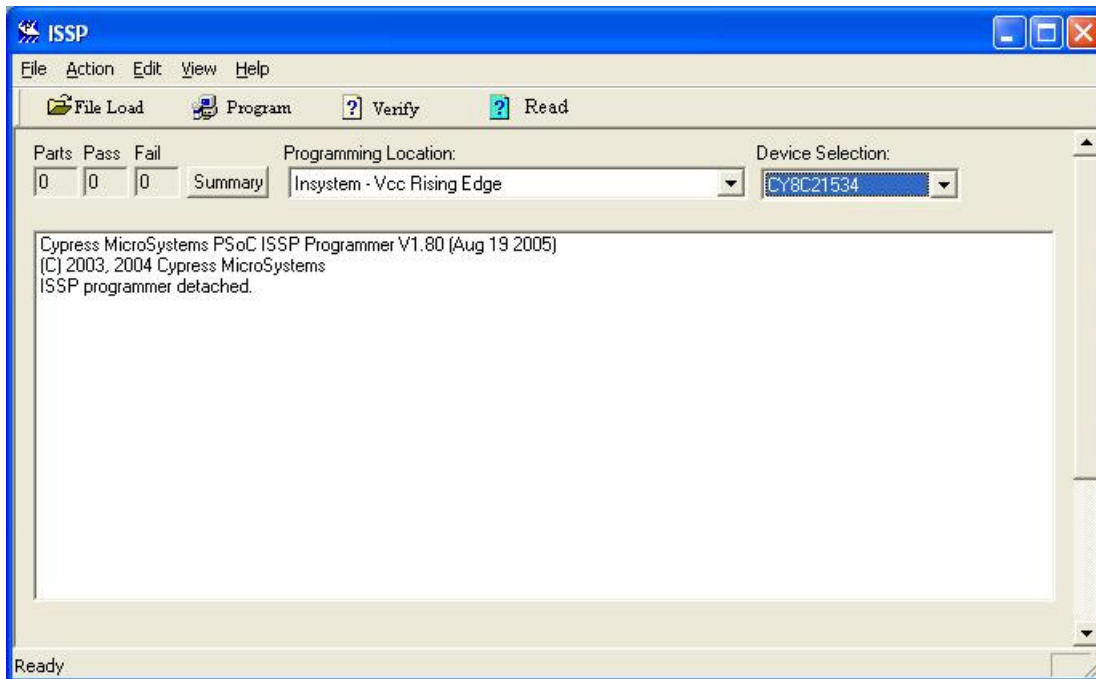


Figure 2-1. Main Window After Startup and Connection of ISSP

The system programs the part and reports the result to the appropriate pass or fail LED. The following is a typical usage sequence.

1. Place a part, CY8C26443-24PI, in the ZIF socket.
2. Select File Load button, browse to the desired HEX file and load it.
3. Select Socket in the Programming Location drop-down box.
4. Select a CY8C26443-24PI in the Device Selection drop-down box.
5. Select the Program button.

3. Using the ISSP



Programming PSoC ICs with ISSP is versatile using the available options.

3.1 Socket Programming

All packages are supported by the socket programming method. Surface-mount parts require an adapter. The part to be programmed is inserted into the Socket on the ISSP programmer. Power is supplied by the ISSP. Execute the following:

1. Select Socket in the Programming Location drop-down box.
2. Select the appropriate part number in the Device selection drop-down box.
3. Load a hex file.
4. Place a part in the socket and click on one of the function buttons; Program, Verify or Read.

3.2 In-System Programming

The device to be programmed is installed on a circuit board or held in an external fixture. Vdd power is supplied by the circuit board or external fixture.

3.2.1 5.2.1 ISSP Cable

Connections to target systems are as follows:

- Vdd (Vdd sense line)
- GND (Ground)
- XRES (External reset)
- SCLK (Serial clock)
- SDAT (Serial data)

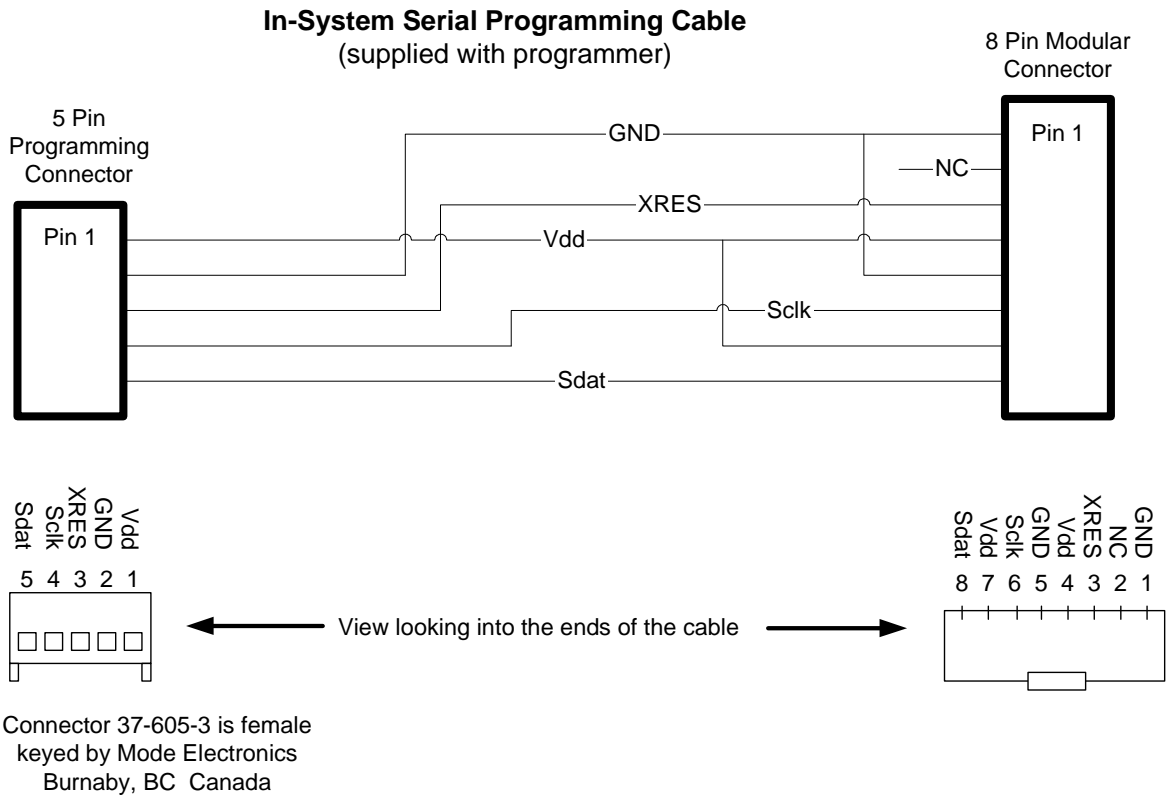


Figure 3-1. Cable Schematic

The Connector part shown here is 37-605-3 female keyed by Mode Electronics Ltd. Burnaby, BC, Canada. <http://www.mode-elec.com/>.

3.2.2 5.2.2 Methods

There are two ISSP methods. One uses the XRES signal to control programming and the other uses the Vdd power-on ramp to control programming.

3.2.2.1 In-System – Vdd Stable On

This is the recommended method of controlling the target part. The programmer cycles XRES high and begins programming. This method does not work on 8-pin parts because they lack the XRES pin.

3.2.2.2 In-System – Vdd Rising Edge

This method works for all packages. The ISSP takes control of the target part when the Vdd is turned on. Each function sequence requires a rising edge on the Vdd. The rising edge triggers the start of the function. When the ISSP is commanded to perform a function it waits until it senses a rising edge on Vdd.

The sequence and delay between command and power application are important. The graphical interface will prompt the user to apply power when needed and to turn it off when done.

3.2.3 5.2.3 Access

The software .exe file runs as a GUI if launched from Windows. If the software is started from a command-line window or directly from another program, it operates in command-line mode.

3.3 DOS Command Line Controls

A DOS batch file called `issp.bat` included on the CD-ROM is a multi-functional file that can be used to view the return code from an ISSP call (among other things). Commands can also be typed directly on the command line.

An ISSP command, followed by a description of each field, is presented below.

```
issp [cy8cxxxxx] [example.hex] [-p|-v] [-s|-i|-n]
```

- ISSP is the call to the PSoC programmer executable. Precede it with a file path if `issp.exe` is not in the current directory.
- `cy8cxxxxx` is the device part number you wish to program (upper or lower case)
- `example.hex` is the HEX file to program into the part or to verify the programmed part against

Function options:

- `-p` invokes the Program function (one function equals Erase, Program, Verify and Protect)
- `-v` invokes the Verify function

Location options:

- `-s` invokes Socket Programming
- `-i` invokes In-System Vdd Rising Edge Programming
- `-n` invokes In-System Vdd Stable On Programming

Specific command examples:

```
issp cy8c26443 example.hex -p -s
```

Programs *example.hex* into the device in the socket.

```
issp cy8c26443 example.hex -v -s
```

Verifies the device in the socket against *example.hex*.

```
issp cy8c26443 example.hex -p -i
```

Programs *example.hex* into the device in the system connected by the programming cable. The Vdd must rise after 4 seconds, but no later than 20 seconds after the command is executed.

```
issp cy8c26443 example.hex -p -n
```

Programs *example.hex* into the device in the system connected by the programming cable. The Vdd is stable during the whole programming sequence.

These DOS commands return '0' if the function is successful and a negative number if the function fails.

3.4 Tester I/O Connector

This connector is used to control the ISSP with logic level signals. The HEX file and programming settings are loaded while the USB is connected to the PC. After the USB is disconnected, the ISSP can be powered by the 9 volt supply and controlled through the Tester I/O connector. See [Figure 3-2](#) for a description of the pins in the Tester I/O connector.

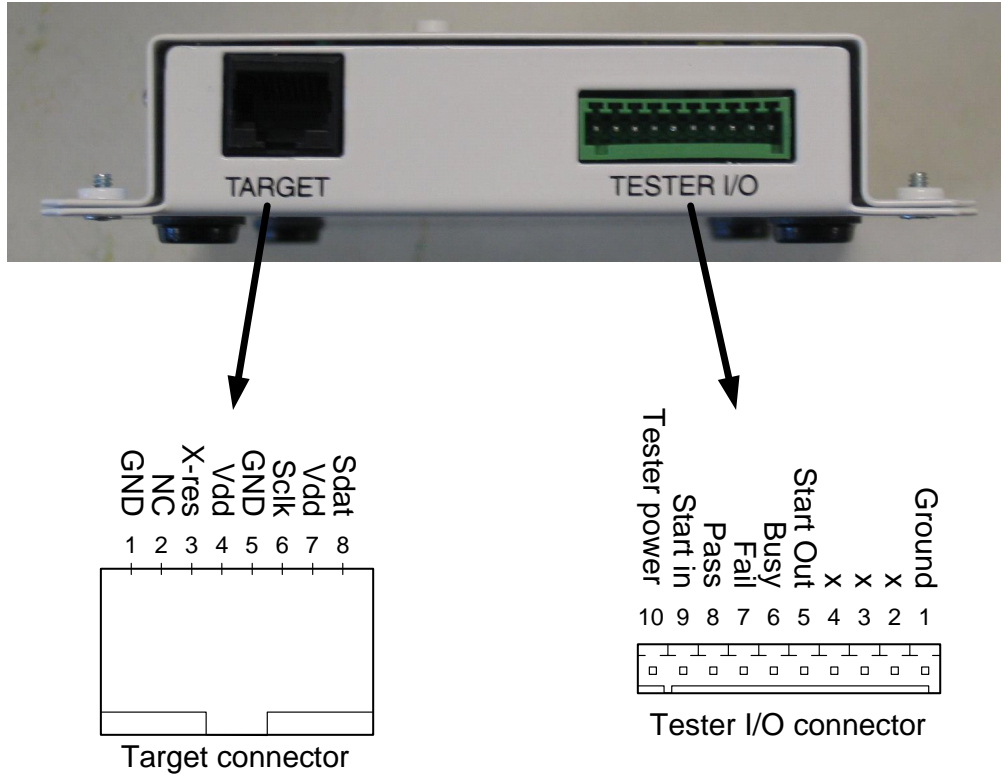


Figure 3-2. Tester IO Connector

3.5 Tester I/O Signaling

Pin 1 must be connected to the test systems ground. Pin 10 must be connected to a DC level equal to the testers logic high level output. Pin 10 controls a level shifter between the ISSP and the tester. [Figure 3-3](#) shows the Tester I/O signals during a programming sequence. The tester drives the “Start in” signal. The ISSP drives the Pass, Fail and Busy lines.

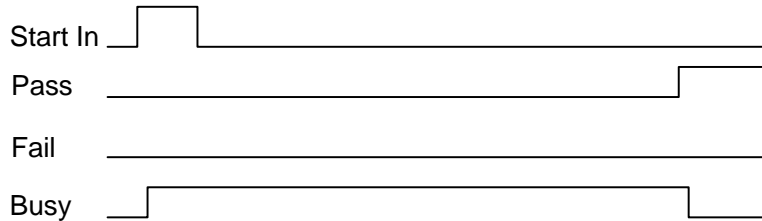


Figure 3-3. Tester I/O Signals

3.6 Test Counters

A new feature in ISSP version 1.80 and later is the ability to automatically keep track of the parts that are programmed. This feature is only available while the ISSP is connected to the PC via the USB interface. The total number of parts programmed, as well as the number of parts that passed or failed is displayed near the top of the window.

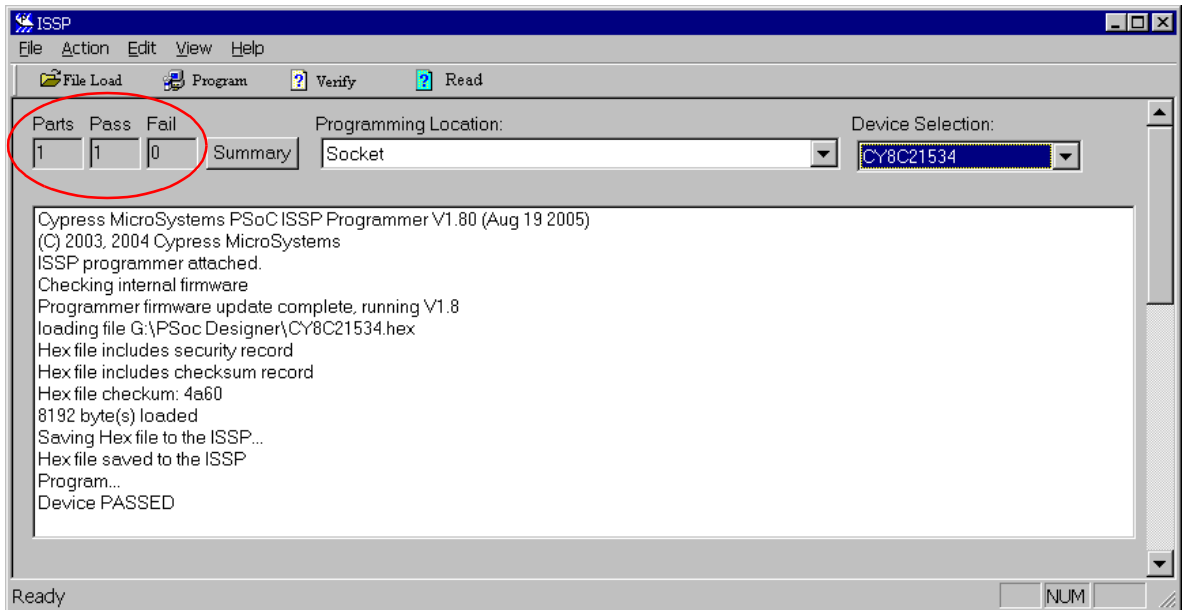
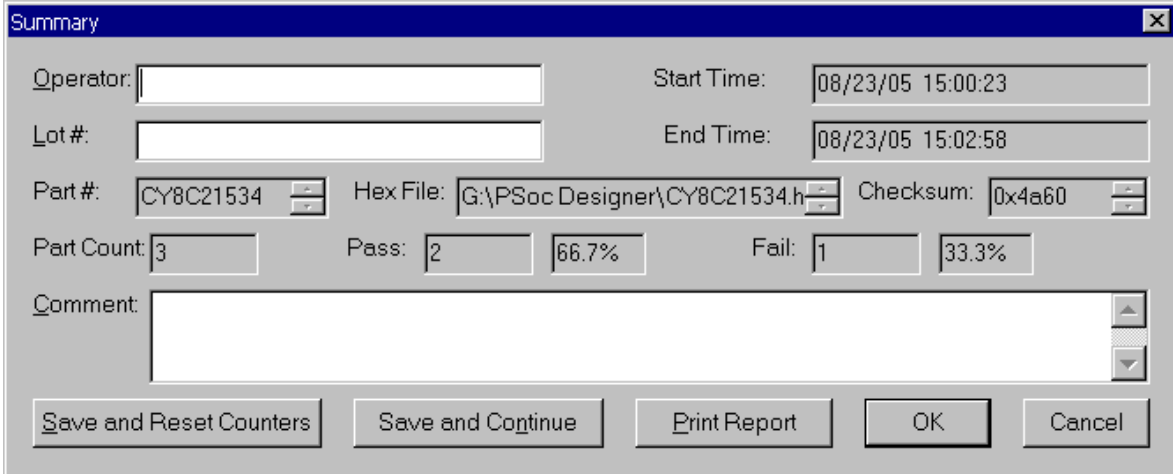


Figure 3-4. Text Counters Window

To obtain additional information, click the Summary button to bring up the following dialog:



The Summary dialog box contains the following fields and controls:

- Operator:** Text input field.
- Lot #:** Text input field.
- Part #:** Text input field with value "CY8C21534".
- Hex File:** Text input field with value "G:\PSoc Designer\CY8C21534.h".
- Checksum:** Text input field with value "0x4a60".
- Start Time:** Text input field with value "08/23/05 15:00:23".
- End Time:** Text input field with value "08/23/05 15:02:58".
- Part Count:** Text input field with value "3".
- Pass:** Text input field with value "2".
- Fail:** Text input field with value "1".
- Percentages:** Two text input fields showing "66.7%" and "33.3%".
- Comment:** Large text area with scrollbars.
- Buttons:** "Save and Reset Counters", "Save and Continue", "Print Report", "OK", and "Cancel".

Figure 3-5. Summary Dialog Box

This displays the time that the first and last parts were programmed, the part number, hex file, checksum, and pass/fail counts with percentages.

Note that if more than one part number was programmed, or if more than one hex file was used, the number of passed and failed parts for each configuration is recorded, but not displayed in this dialog box. To obtain this detailed data, print the report or save it to a log file. Any information entered in the Operator, Lot#, or Comment field will be included in the printout and the log file.

To reset the counters, click the Save and Reset Counters button from the Summary dialog. The counters will not be reset until the data has been saved successfully. To save an intermediate copy of the data without resetting the counters, select Save and Continue. Clicking OK will save what has been entered in the Operator, Lot# and Comment fields so that they will be filled in the next time the Summary dialog is opened, but the data will not be written to disk or printed. Cancel exits the dialog and discards any changes made to the Operator, Lot# or Comment fields.