

PSoC Programmer

By: Alessandro Molini

Associated Project: Yes

Associated Part Family: CY8C26xxx

PSoC Designer Version: 4.2

Associated Application Notes: AN2014, AN2026

Abstract

This Application Note describes a simple programmer for the PSoC™ device.

Introduction

With this Application Note, I want to help the beginner and the professional programmer in the PSoC world with a simple, economic tool for Flash programming. This tool allows one PSoC-based device to serve as a programmer for other PSoC-based devices. The ICE-4000 In-circuit Emulator is still necessary for debugging and programming the first device. However, small-scale, simultaneous production programming, and software development is possible.

The heart of the project is a PSoC CY8C26443. The necessary hardware has been reduced to a minimum. The device sends clock, data in/out and Xres to the programming chip. Programming is established through the PSoC serial COM port connected to a MAX232. The download baud rate is fixed at 19200, N, 8, 1. The hardware is complete with a 5V stabilized power supply. The main peripherals are inside the PSoC.

External Hardware

Figure 5 shows the CY8C26443 external connections. The programmer chip will be clocked at 24 MHz. Precautions should be taken for the rigorous 5V power-supply connections with the programming chips, typically pre-assembled on the production board. As the EXECUTE MODE is planned for a clock at 2 MHz, it is wise to keep the programming wires as short as possible and insert serial resistances as specified in Application Note AN2014 "[Design for In-System Serial Programming \(ISSP\)](#)." Two LEDs, green and red, will signal the programming status as "in progress" or "in error."

Operational Theory

The program flow follows the specifics outlined in Application Note AN2026 "[In-System Serial Programming \(ISSP\) Protocol](#)" and is shown in Figure 1. For a better understanding of the procedures used here, the reader is encouraged to review the material contained in AN2026.

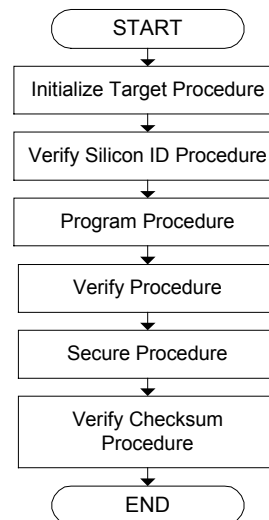


Figure 1. Programming Flow

The programmer uses the reset signal, Xres, to begin the start procedure. Hardware and software changes allow modification of this application with the Power-On-Reset programming (the essential cycle for the smaller devices, CY8C25xxx).

Other essential characteristics are the accuracy and the stability of the frequency in EXECUTE MODE. In all other phases, the frequency and stability for the programming clock is not essential.

Internal Hardware

The PSoC internal hardware is shown in Figure 2. The peripherals are a serial-synchronous Master SPI to communicate commands and data to the programming chip, as well as an asynchronous serial receiver to receive the Intel hexadecimal program from a PC. A timer/counter produce a stable 2 MHz clock signal during the EXECUTE MODE.

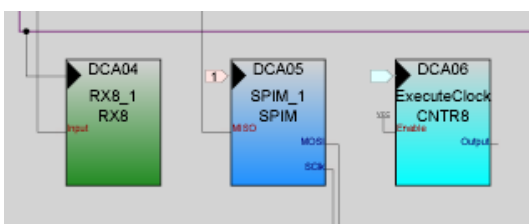


Figure 2. User Module Placement

Software

The software is separated into two sections: the main program written in C and the interrupt routine for asynchronous serial communication written in assembly.

The main C program covers the initialization and manages iterations with the programming chip.

The assembler interrupt section, *rx8_1int.asm*, manages the ASCII record sent from the PC, converts the record to binary, checks the checksum and signals the correct reception to the main C routine.

EXECUTE Procedure

The most critical phase in the system is EXECUTE MODE in which the programmer must send an accurate and stable frequency of at least 2 MHz to the device in order for the chip to be programmed.

To activate the command previously sent by the SPI, the software switches pin P0[4] function from SCLK, connected to the SPI, to CLOCK, connected to the ExecuteClock Counter (cnt8). This is shown in Figure 3. The software then waits for the answer with a low signal from the target chip, which is the signal from the target that the previously sent command has been completed.

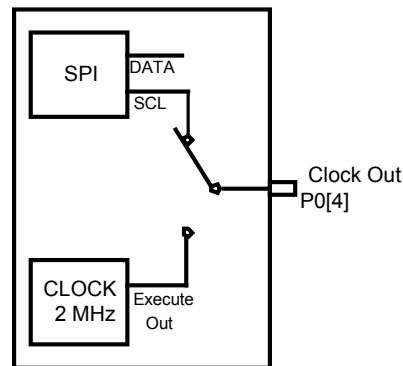


Figure 3. P0[4] Function Switch

Vectors

The start of each procedure is the transmission of vectors represented by specific data, organized in groups of 22 bits. See Figure 4.

```

Initialize 1
110010100000000000000000
000000000000000000000000
000000000000000000000000
100011110110010000000000
110111101110000000000000
110111101100000100000000
1001111100000111010000
1001111100100001011000
110111101010000000000000
1101111010000000011000
100111110110010000000000
110111110010011000000000
110111110100100000000000
110111100000000100000000
110111110000000000000000
110111111100010010000
    
```

Figure 4. Initialize 1 Vectors

The transmission through the SPI, rigorously at 8 bits, is only possible by adding two 0 bits at the beginning of each line. The resultant 24 bits will be transmitted from the SPI in 3 transmissions.

Table1. Transmitted Bytes for Figure 4, Line 1.

Byte 1	Byte 2	Byte 3
00110010	10000000	00000000

Control Software

Operation does not require any particular download procedures. Any serial communication program can handle the download procedures without problems. The required baud rate should be at 19200, N, 8, 1. The start condition is realized through the dispatch of any ASCII character through the serial port. After this, a delay of about 50 mS is needed for the programmer to initialize the target. Then the Hex file can be transmitted.

The programming result is displayed by the LEDs. When programming is successful, the green LED lights. If programming is not successful, the red LED lights.

A small Visual Basic application developed to operate the programmer has been attached. The software does not have any controls. When the Program button is clicked, a File Open dialog box appears. Select the .hex file to be programmed and click OK. The chip will be programmed and a message box will appear saying, "The Device has been Programmed." This message will appear even if the programming failed. This is because there is no feedback to the PC from the programmer about the result of the operation.

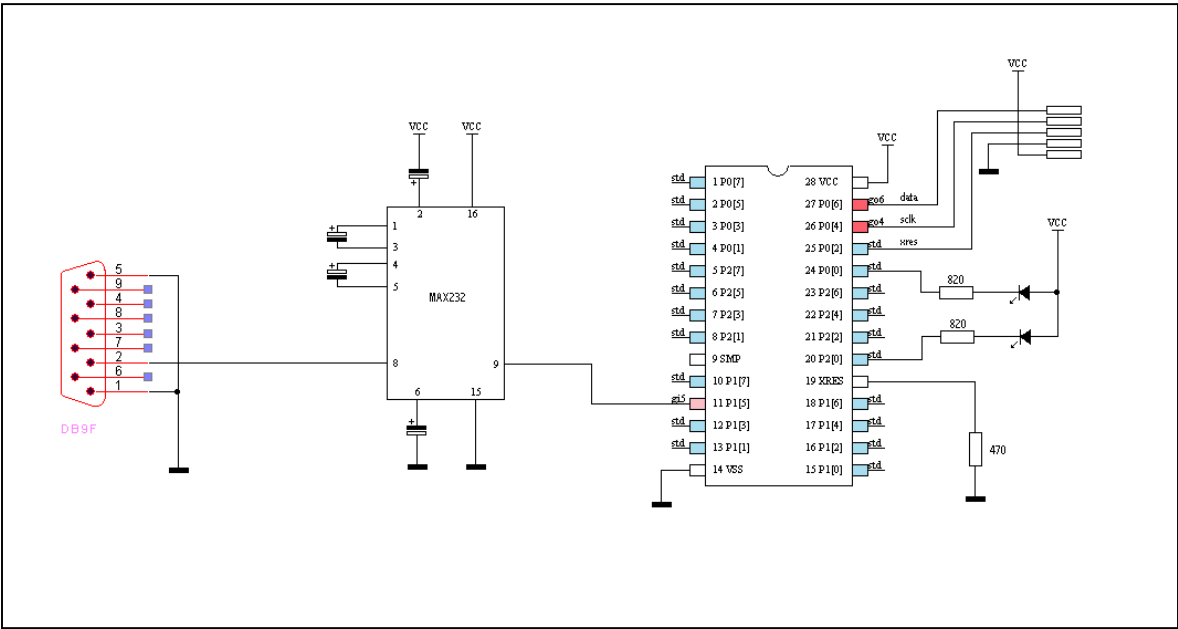


Figure 5. External Connections

About the Author

Name: Alessandro Molini

Background: HW-SW industrial automation designs, motion control, PLC, and motor control.

Contact: alex.mol@tiscali.it

Cypress MicroSystems, Inc.
2700 162nd Street SW, Building D
Lynnwood, WA 98037
Phone: 800.669.0557
Fax: 425.787.4641

<http://www.cypress.com/> / <http://www.cypress.com/support/mysupport.cfm>

Copyright © 2003-2004 Cypress MicroSystems, Inc. All rights reserved.
PSoC™, Programmable System-on-Chip™, and PSoC Designer™ are trademarks of Cypress MicroSystems, Inc.
All other trademarks or registered trademarks referenced herein are the property of their respective owners. The information contained herein is subject to change without notice. Made in the U.S.A.