AN909

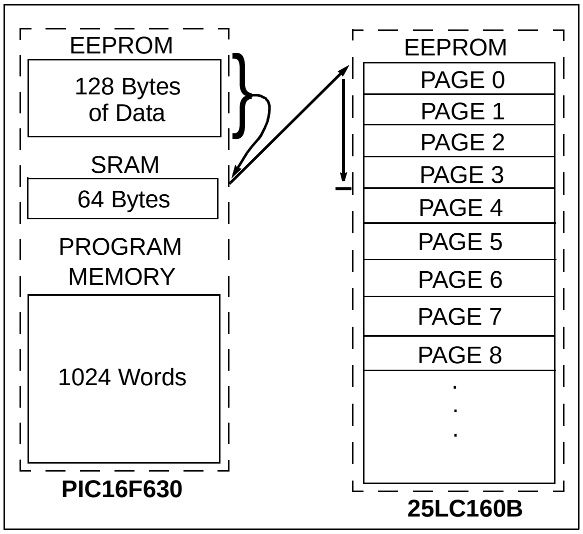
Interfacing SPI™ Serial EEPROMs to PIC16 Devices

Author: *Ken Dietz*
Microchip Technology Inc.

INTRODUCTION

When connecting an SPI™ master device, like a microcontroller, to an SPI slave device such as an EEPROM, understanding the command sequence sent to the memory is vitally important. With the aid of this application note and the associated code examples, Microchip Technology has eased the job of designing SPI Electrically Erasable Programmable Read-Only Memory (EEPROMs) into systems. While this shows one particular implementation, system requirements vary from design to design, which may be different than those shown here. However, this article shows designers how to set up the framework for SPI communications between microcontrollers and SPI EEPROMs. It also gives designers a starting point for complex designs that utilize multiple SPI slaves in electronic systems. Besides these important topics, the most important point that system designers should learn from this application relates to the necessity of reading the STATUS register of SPI EEPROMs prior to sending any command sequences to them.

FIGURE 1: SIMPLIFIED DATA TRANSFER MODEL



FIRMWARE DESCRIPTION

The purpose of the program is to transfer 128 bytes of data from the EEPROM section of the PIC16F630 to 128 sequential locations of the 25LC160B over the SPI bus. Rather than using any hardware modules for communications, this system implements the SPI bus through software so that any I/O port can be used for communicating with EEPROMs. A conceptual model for handling this system is shown in Figure 1. The data from the PIC16F630 internal EEPROM is written to the first four pages of the 25LC160B. Designers can easily change these locations by modifying the definition statements at the beginning of the firmware. Similarly, the page size can also be modified in the definition statements.

As the program advances through its states, a LED array is updated showing the progress of the program. Lengthy delays are added to the beginning of the states for the LED visual indicators, which assists in debugging.

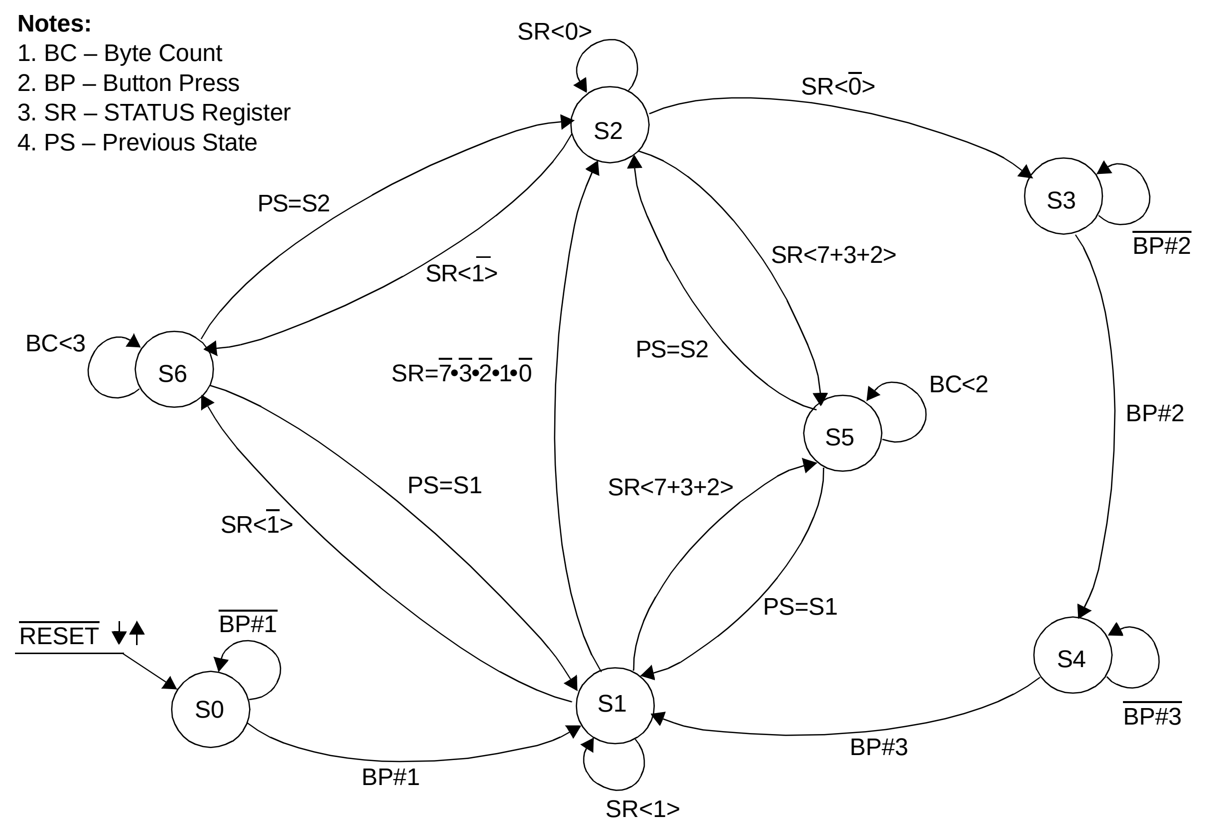
After loading the 25LC160B with 128 bytes of information, the PIC16F630 reads and verifies the contents of the 25LC160B with the values written to its own data EEPROM in the form of a checksum. The program then returns to the first state to repeat the communication sequence again.

The timing for the SPI bus is set up in such a way that communications between the microcontroller and the EEPROM depends on the 4 MHz internal oscillator of the PIC16F630. This system performs correctly between 4 MHz and 10 MHz, inclusive with external oscillators.

> **Note:** If faster data rates are needed or if a faster crystal is used for the system, then the firmware timing may need to be modified to include time delays to accommodate the timing specifications in the 25LC160B data sheet.

The firmware is designed to operate with two microcontrollers, so conditional assembly is implemented to show a simplified migration path between two mid-range PICmicro® microcontrollers. In this case, the secondary processor is the PIC16F876, which is programmed using the MPLAB® ICD 2 and a modified PICDEM™ 2 Plus Demonstration Board.

FIGURE 2: FINITE STATE MACHINE FOR SPI COMMUNICATIONS



STATES

Note:	Any references to the STATUS register pertain to the EEPROM rather than the microcontroller.

When communicating with SPI EEPROMs, it is imperative to read the STATUS register to understand the state of the memory. The STATUS register contains bits that provide information related to enabling and disabling write functions, block protect features, and a write in progress bit. The write-protect enable bit and the block protect bits are nonvolatile bits that can be written or read. Two bits are volatile read-only bits; one shows the status of the write enable latch and the other indicates whether or not a write cycle is in progress. Status bits 4, 5 and 6 are not implemented on the 25LC160B, but are always read as zeros.

The Finite State Machine (FSM) shown in Figure 2 takes into account only one EEPROM for storing information. The model can be further extended to take advantage of placing multiple slaves on the same SPI bus. The primary differences between complex models with multiple slaves and the one shown in Figure 2 is the capability of the master to assert multiple chip select outputs and the ability to discern which slave should be communicating at any given time.

A discussion for getting started with state diagrams and microcontrollers is included with the *PICkit™ 1 User's Guide* (DS40051). The processor controls I/O functions during states rather than during transitions, which indicates that this FSM is a Moore Model.

S0: Initialization

After powering up the microcontroller or after resetting the demo board, the program enters state S0. The code contained in S0 initializes the microcontroller and holds the processor in a polling loop until the user presses the push-button switch on the demo board. After the button switch is pressed and released, the program advances to the next state, S1. Pressing the button is represented in Figure 1 with the state variable "BP".

S1: FSM Main

In S1, the STATUS register is read and tested for certain conditions. All of the outbound transitions for leaving S1 depend on the value read from the STATUS register of the 25LC160B. Moreover, the code in S1 is set up according to priorities for handling the conditions read from the STATUS register. Here are the priorities in order of importance:

- Read STATUS register
- If STATUS register = 0x02, advance to S2
- If write enable latch bit is low, go to S6
- If block protect bits or write-protect enable bit are high, go to S5
- If write in progress bit is high, go to start of S1

The value of 0x02 is represented in the state diagram by ANDing bits 7, 3, 2, 1 and 0 of the STATUS register. A bar over any of the bits in Figure 1 indicates that the bit is active low. If the bar is not shown, the bit is active high. Again, bits 4, 5 and 6 are always read as zero.

In this application, 0x02 is the expected value for the STATUS register. But in other systems, the block protect bits may be logically high, so values such as 0x0A, 0x06 or 0x0E are possible.

S2: Write EEPROM, Initiate Write Cycle

Within this state, the microcontroller loads 128 bytes from the data EEPROM of the microcontroller into the buffer of the 25LC160B one page at a time. A page for this EEPROM is 32 bytes in length. After loading one full page, a write cycle is initiated. The program then loops around until all four pages of data are transferred to the 25LC160B.

The data transfer checksum is simultaneously stored in RAM during this state. The checksum is a simple checksum that is the result of adding all the bytes transferred across the SPI bus. Since there are 128 bytes that can have a maximum value of 0xFF, the largest checksum possible is 0x7F80, which is 0x80 multiplied by 0xFF. Because of this, the checksum is saved to two consecutive RAM locations, CheckSumH and CheckSumL. When bytes are added to CheckSumL, the Carry bit in the microcontroller STATUS register is tested to determine whether or not the result is big enough to increment CheckSumH. When the Carry bit is high, CheckSumH should be incremented, and if the bit is low, then CheckSumH is not affected. After all the pages of data are written to the EEPROM by the PIC16F630, the program advances to S3.

S3: EEPROM Written, Get User Input

In this state, the microcontroller completed its programming sequence on the 25LC160B with 128 bytes of data and is now waiting for user input. After the user presses and releases the push-button switch, the program advances to S4.

S4: Verifying EEPROM

In this state, the microcontroller utilized the unlimited sequential byte read feature of the 25LC160B. All 128 bytes are read and added together to create a new checksum. This new checksum is subtracted from the previous value obtained during S2. If the result is zero, the program advances back to S1, and if it is not zero, the program enters an infinite loop and lights up all the LEDs on the PICkit™ 1, which indicates an error occurred.

S5: Send Write STATUS register

S5 is accessed whenever any nonvolatile bits in the STATUS register are high. After directing the microcontroller to write 0x02 to the STATUS register, the program advances back to the state from where it entered S5. Since the program enters S5 from different locations (S1 and S2), the previous state must be stored in RAM in order to transition back to the state where the program entered S5.

S6: Send Write Enable

Another focal issue to note when working with SPI EEPROMs concerns their condition on power-up or after a write cycle takes place. If the Write Enable latch is disabled, the slave disregards any write commands from the master until the slave receives a Write Enable command. As such, S6 is required to bring the EEPROM into a mode where it can be written. Similar to S5, after the sequence of commands is completed for S6, the program jumps back to the state where it originally accessed S6. So, the previous state must be stored in RAM not only to exit S5, but also S6.

SPI SPECIFIC FUNCTIONS

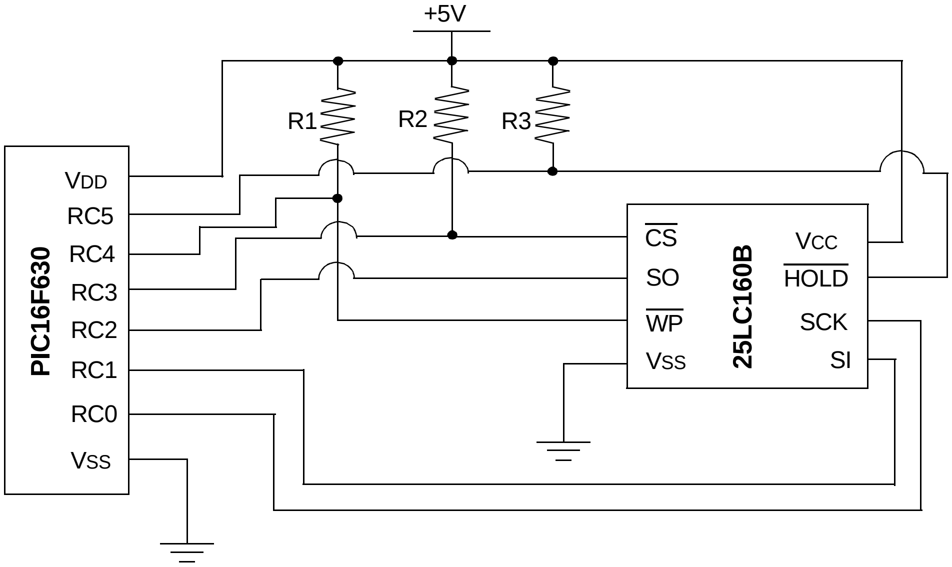
The SPI functions for the firmware are written in such a way that they can be used generically for SPI communications between PICmicro® microcontrollers and SPI EEPROMs. Even though this code is developed for the 25XX160B series, the functions can be used on any of the other SPI EEPROMs that Microchip Technology offers with little or no modification.

Two sets of SPI commands are divided into smaller components, namely, the SPI read and write functions. For each of these commands, there is a call to a header function, which sends the command byte and address word to the slave, followed by a call to TransferSPIData. This latter function, TransferSPIData, is called from within loops to transfer a pre-assigned number of bytes between the master and slave chips.

PICkit™ 1 FLASH STARTER KIT DEVELOPMENT METHODOLOGY

A modified PICkit 1 Flash Starter Kit was used for developing part of this application. The modification includes a ZIF socket attached to the demo board with the necessary connections for SPI EEPROMs to the processor. After using the MPLAB ICD to write and assemble code for the PIC16F630, designers can download their hex files to their target processors for testing using the PICkit 1 Flash Starter Kit. This development methodology continues in an iterative fashion until the program is complete. Figure 3 shows a simplified schematic for this system using the PICkit 1 Flash Starter Kit.

FIGURE 3: SCHEMATIC FOR SPI INTERFACE ON PICkit DEMO BOARD



Notes:
1. Refer to the PICkit demo board schematic for PORTA connections.
2. All resistors are 10 kOhms.

CONCLUSION

Designing communications for the SPI bus requires forethought and planning. With the aid of this application note, designers can begin to build their own SPI libraries for device-level SPI communications whether they are creating simple systems with one master and one slave or complex systems containing a master with multiple slaves.

REFERENCES

Microchip Technology Inc. (2003). *25AA160A/B, 25LC160A/B 16K SPI™ Bus Serial EEPROM* (DS21807). Chandler, Arizona: Microchip Technology Inc., http://www.microchip.com

Microchip Technology Inc. (2003). *PIC16F630/676 Data Sheet* (DS40039). Chandler, Arizona: Microchip Technology Inc., http://www.microchip.com

Microchip Technology Inc. (2003). *PIC16F87X Data Sheet* (DS30292). Chandler, Arizona: Microchip Technology Inc., http://www.microchip.com

Microchip Technology Inc. (2003). *PICkit 1 Flash Starter Kit User's Guide* (DS40051). Chandler, Arizona: Microchip Technology Inc., http://www.microchip.com

Tinder, R. (1991). *Digital Engineering Design: A Modern Approach.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Migratable Memory, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Linear Active Thermistor, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance and WiperLock are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2005, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

♻ Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

San Jose
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Qingdao
Tel: 86-532-502-7355
Fax: 86-532-502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066
China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-2229-0061
Fax: 91-80-2229-0062

India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Penang
Tel: 604-646-8870
Fax: 604-646-5086

Philippines - Manila
Tel: 011-632-634-9065
Fax: 011-632-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsinchu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Weis
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393
Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-352-30-52
Fax: 34-91-352-11-47

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

07/01/05