

---

## Software SPI Master

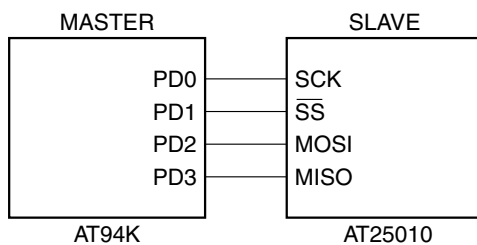
### Features

- Up to 444 Kbytes/Sec Throughput @ 10 MHz
- Directly Supports Large Block Writes
- Easily Expandable for Multiple SPI Slaves
- Operates in SPI Mode 0
- 16-bit Data, Easily Modified to 8-bit Operation
- No Interrupts Required
- Low Register Count; Only 3 Needed (2 if 8-bit Mode)
- Small Code Size; 35 Words, Including Initialization
- Interfaces to Atmel AT25xxx Serial EEPROM

### Introduction

The Synchronous Peripheral Interface (SPI) standard is gaining rapidly in popularity. It allows faster communication than 2-wire serial interface and can be implemented using fewer gates which means lower silicon costs. This application note describes a set of low-level routines for software implementation of the SPI protocol, in Master Mode (all communications originate from the AVR<sup>®</sup> microcontroller embedded in FPSLIC<sup>™</sup> devices). These can be used as the basis for communicating with Atmel's 25XXX family of serial EEPROM memories, as well as a host of other 2-wire serial interface peripherals, such as display drivers. The Assembly code with the SPI Master Routines can be found in the FPSLIC Software section of the Atmel web site (<http://www.atmel.com>), under the **3041.asm** archive.

**Figure 1.** Connection between the Microcontroller and the Slave Device

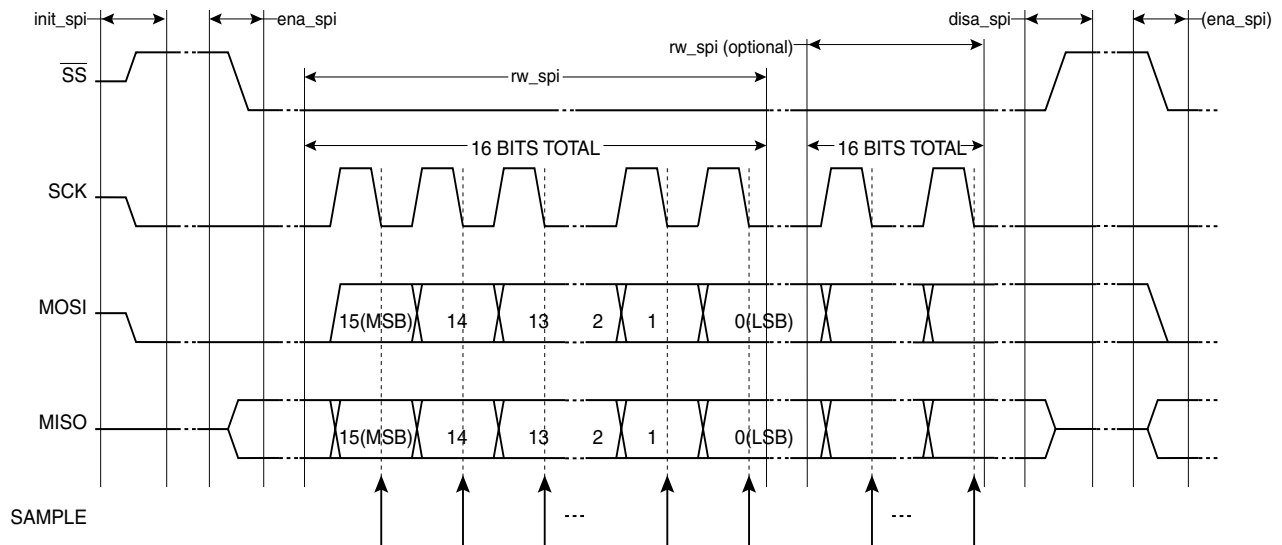


**Programmable  
SLI  
AT94K  
AT94S**

### Application Note



**Figure 2. SPI Transmission Timing**



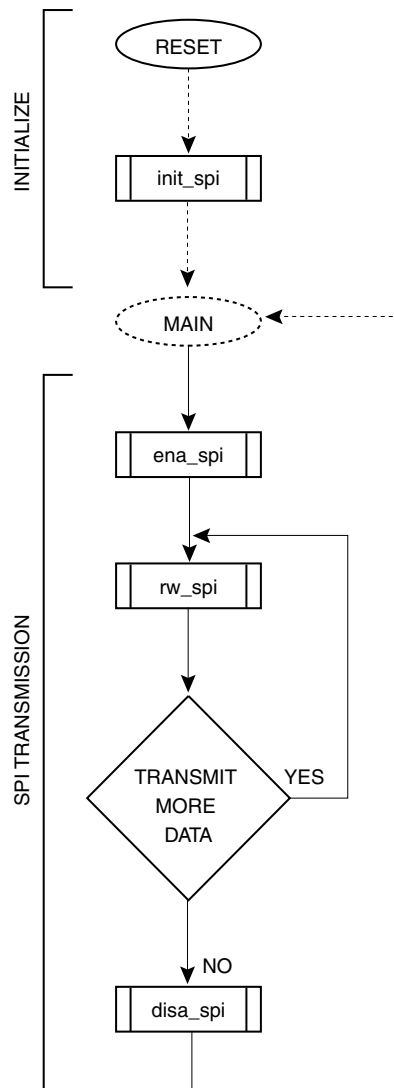
## Theory of Operation

SPI Mode 0 implies the following conditions:

1. The SCK signal is Low when idle,
2. MOSI data (from the microcontroller) must be stable at some time (setup) before the rising edge of the clock,
3. MISO data (from the slave) is valid at some time ( $t_{VALID}$ ) after the falling edge of the clock,
4. Data is always sent MSB first.

These routines have been written to meet those standards, with special attention being paid to the boundary conditions, such as  $t_{VALID}$  and  $t_{SETUP}$ .

**Figure 3. SPI Master Flowchart**



## Subroutine Description

### init\_spi:

This routine initializes the SPI port lines. The macros will need to be modified if PORTD is not used; otherwise, simply changing the Port Definition .EQU's for the SCK, MOSI, MISO and NSS (not-Slave-Select) is adequate. This routine has no entry requirements and returns nothing.

### ena\_spi:

This routine makes sure that SCK is low before setting  $\overline{SS}$  to the active state. This routine has no entry requirements and returns nothing.

### disa\_spi:

This routine brings the  $\overline{SS}$  signal high (inactive). It should be called when a transmission sequence is complete, to prevent spurious clocking of the SPI slaves. It also has no entry requirements and returns nothing.

### rw\_spi:

This routine sends/receives either an 8-bit or 16-bit data word, depending on whether the user has modified the source code. It leaves the SCK signal Low upon exit, and does not modify the  $\overline{SS}$  signal; therefore, many back-to-back writes are possible by simply calling this routine repeatedly. Entry requirements are simply that the spi\_lo (and spi\_hi, if used in 16-bit mode) are initialized with the data to be sent, prior to calling this routine. Upon exit, the same register(s) contain the data received from the SPI slave.

## Conversion to 8-bit Operation

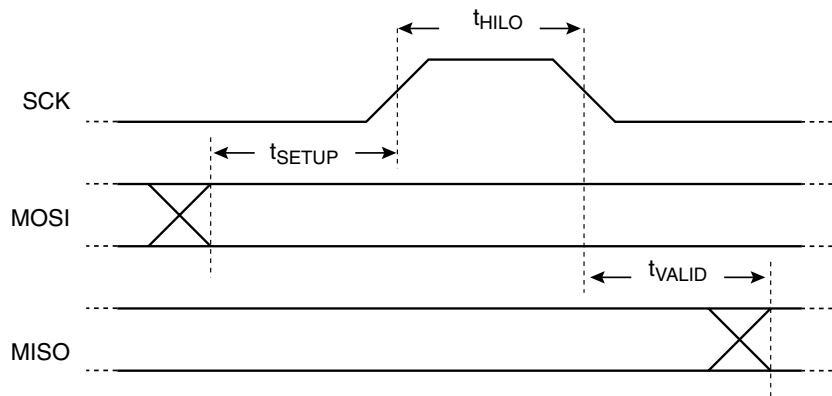
There are only two lines of code that require modification in order for the application note to work with 8-bit data words. Both of the changes are in the rw\_spi routine. First, the bit counter initialization value must be changed from 16 to 8; secondly, the line “rol spi\_hi” must be commented out. No other changes are necessary, except that the spi\_hi register is no longer needed and is never used.

## Macros

Macros are used to make the code more readable. It is assumed that Port B will be used for the SPI interface; if this is not the case, then the macros themselves must be modified to reflect the correct port. Pins for the 4 SPI signals are listed under “PORT DEFINITIONS” as .EQU’s and are easily modified.

The delay function macros deserve some additional explanation. To help conserve registers, this routine gets double-duty out of the TEMP register by keeping a 5-bit counter value and the delay counter for measuring the SCK High and Low time. The latter are kept in the uppermost bits. By simply incrementing the upper 3 bits and watching for a rollover, we can track time without affecting the lowest 5 bits. Note that, in reality, we are subtracting values rather than adding; the end result is the same, except that Carry is cleared (rather than set) when the value of the upper 3 bits rolls from 7 to 0.

**Figure 4.** Setup & Hold Timing



## Delay from MOSI Update to SCK Rising Edge

The delay period from updating MOSI to the rising edge of SCK is a critical period, as it is actually the data setup time ( $t_{\text{SETUP}}$ ) for the connected peripheral. In this routine, the amount of setup time is 2 clock cycles if MOSI is changing to Low, 3 if changing to High (this skew allows for different hi/lo drive levels as well). At a 10 MHz clock, this equates to 200 – 300 ns. If more time is required, NOPs should be placed immediately before the sck\_hi statement.

## SCK Duty Cycle

SCK will spend most of its time Low, due to the fact that it is low during the “overhead” portions of the send loop. However, if the connected peripheral can handle a faster rate, the high-time delay ( $t_{HILO}$ ) can be reduced to match the minimum SCK-high specification.

With a delay value of 1 being used, the routine requires approximately 22.5 AVR clocks per data bit, and the SCK-high time is exactly 4 clocks. At 10 MHz, this would correspond with an equivalent SCK-high spec of 400 ns, and an overall throughput of 444 Kbytes/sec.

The low-time delay must be set to meet the “SCK fall to MISO (slave data out) Valid” period ( $t_{VALID}$ ), which is determined by the peripheral. With a delay value of 1, this routine will have 3 AVR clocks delay before latching the MISO signal into the pin’s synchronization register. Again, at 10 MHz, this is a 300 ns period, and can be adjusted upward by increasing the delay value.



## Atmel Headquarters

**Corporate Headquarters**  
2325 Orchard Parkway  
San Jose, CA 95131  
TEL 1(408) 441-0311  
FAX 1(408) 487-2600

### Europe

Atmel Sarl  
Route des Arsenaux 41  
Casa Postale 80  
CH-1705 Fribourg  
Switzerland  
TEL (41) 26-426-5555  
FAX (41) 26-426-5500

### Asia

Atmel Asia, Ltd.  
Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimhatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### Japan

Atmel Japan K.K.  
9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
TEL (81) 3-3523-3551  
FAX (81) 3-3523-7581

## Atmel Operations

### Memory

Atmel Corporate  
2325 Orchard Parkway  
San Jose, CA 95131  
TEL 1(408) 436-4270  
FAX 1(408) 436-4314

### Microcontrollers

Atmel Corporate  
2325 Orchard Parkway  
San Jose, CA 95131  
TEL 1(408) 436-4270  
FAX 1(408) 436-4314

### Atmel Nantes

La Chantrerie  
BP 70602  
44306 Nantes Cedex 3, France  
TEL (33) 2-40-18-18-18  
FAX (33) 2-40-18-19-60

### ASIC/ASSP/Smart Cards

Atmel Rousset  
Zone Industrielle  
13106 Rousset Cedex, France  
TEL (33) 4-42-53-60-00  
FAX (33) 4-42-53-60-01

Atmel Colorado Springs  
1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL 1(719) 576-3300  
FAX 1(719) 540-1759

Atmel Smart Card ICs  
Scottish Enterprise Technology Park  
Maxwell Building  
East Kilbride G75 0QR, Scotland  
TEL (44) 1355-803-000  
FAX (44) 1355-242-743

### RF/Automotive

Atmel Heilbronn  
Theresienstrasse 2  
Postfach 3535  
74025 Heilbronn, Germany  
TEL (49) 71-31-67-0  
FAX (49) 71-31-67-2340

Atmel Colorado Springs  
1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL 1(719) 576-3300  
FAX 1(719) 540-1759

### Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Atmel Grenoble  
Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
TEL (33) 4-76-58-30-00  
FAX (33) 4-76-58-34-80

---

**Atmel Programmable SLI Hotline**  
(408) 436-4119

**Atmel Programmable SLI e-mail**  
fpslic@atmel.com

### FAQ

Available on web site

### e-mail

literature@atmel.com

### Web Site

<http://www.atmel.com>

### © Atmel Corporation 2002.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

ATMEL® and AVR® are the registered trademarks of Atmel. FPSLIC™ is the trademark of Atmel.

Other terms and product names may be the trademarks of others.



Printed on recycled paper.