

## AVR320: Software SPI Master

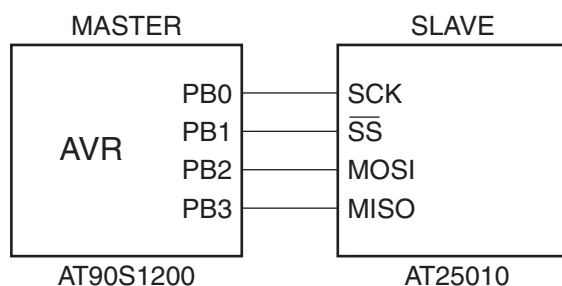
### Features

- Up to 444Kb/S 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 Three Needed (Two 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 TWI and can be implemented using fewer gates which means lower silicon costs. Although the larger AVR family members include a fully functional SPI interface, the smaller members do not. 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). 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 peripheral IC's such as display drivers.

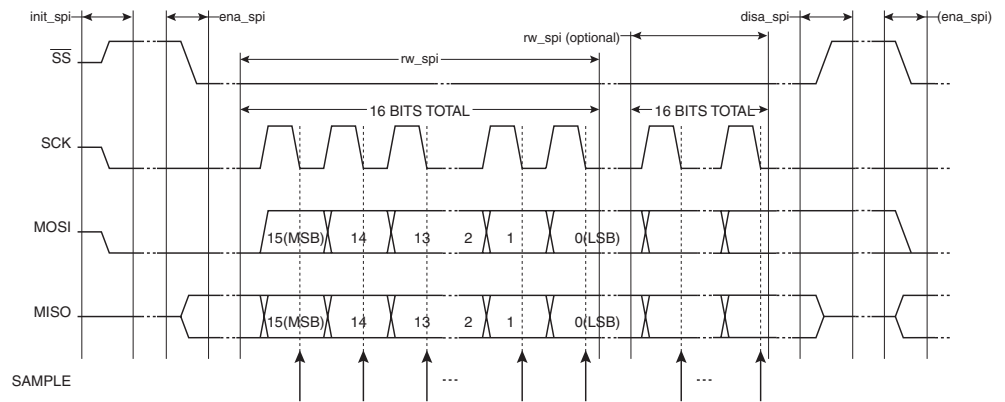
**Figure 1.** Connection Between the AVR MCU and the Slave Device Shown as a Serial EEPROM



8-bit **AVR**<sup>®</sup>  
Microcontroller

### Application Note

**Figure 2. SPI Transmission Timing**



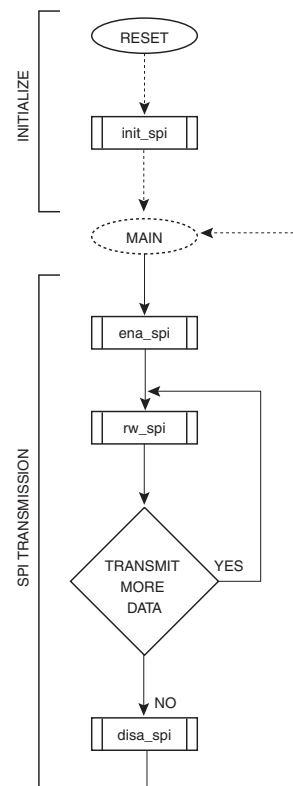
## Theory of Operation

SPI Mode 0 implies the following conditions:

1. The SCK signal is low when Idle.
2. MOSI data (from the AVR) 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 ( $t_{\text{VALID}}$ ) 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  $t_{\text{VALID}}$  and  $t_{\text{SETUP}}$ . These timings are detailed later in this application note.

**Figure 3. SPI Master Flowchart**



## Subroutine Description

`init_spi:`

This routine initializes the SPI port lines. The macros will need to be modified if Port B 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- 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

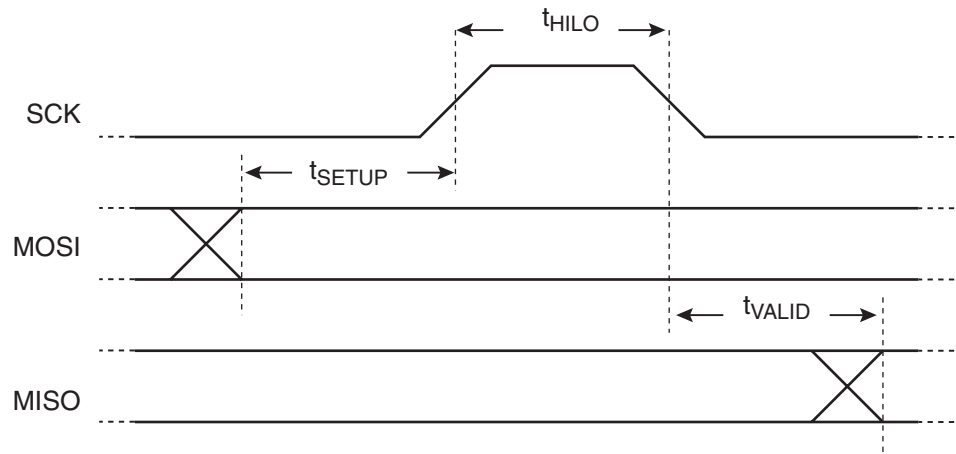
There are only two lines of code that require modification in order for the application note to work with 8-bit data words. Both of the changes are in the `rw_spi` routine. First, the bit counter initialization value must be changed from 16 to 8; Secondly, the line "`rol spi_hi`" must be commented out. 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 four 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 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 three bits and watching for a rollover, we can track time without affecting the lowest five 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 three 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 two clock cycles if MOSI is changing to low, three 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 ( $t_{\text{HILO}}$ ) can be reduced to match the minimum SCK-high specification.

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

The low-time delay must be set to meet the “SCK fall to MISO (slave data out) Valid” period ( $t_{\text{VALID}}$ ), which is determined by the peripheral. With a delay value of one (1), this routine will have three 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 Corporation

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 487-2600

## Regional Headquarters

### Europe

Atmel Sarl  
Route des Arsenalux 41  
Case Postale 80  
CH-1705 Fribourg  
Switzerland  
Tel: (41) 26-426-5555  
Fax: (41) 26-426-5500

### Asia

Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimshatsui  
East Kowloon  
Hong Kong  
Tel: (852) 2721-9778  
Fax: (852) 2722-1369

### Japan

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

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 436-4314

### Microcontrollers

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 436-4314

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

Zone Industrielle  
13106 Rousset Cedex, France  
Tel: (33) 4-42-53-60-00  
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906, USA  
Tel: 1(719) 576-3300  
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park  
Maxwell Building  
East Kilbride G75 0QR, Scotland  
Tel: (44) 1355-803-000  
Fax: (44) 1355-242-743

### RF/Automotive

Theresienstrasse 2  
Postfach 3535  
74025 Heilbronn, Germany  
Tel: (49) 71-31-67-0  
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906, USA  
Tel: 1(719) 576-3300  
Fax: 1(719) 540-1759

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

Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
Tel: (33) 4-76-58-30-00  
Fax: (33) 4-76-58-34-80

---

## Literature Requests

[www.atmel.com/literature](http://www.atmel.com/literature)

**Disclaimer:** The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© Atmel Corporation 2005. All rights reserved. Atmel®, logo and combinations thereof, Everywhere You Are®, AVR®, AVR Studio®, and others, are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.



Printed on recycled paper.