

# Industry-standard 2-wire Protocol “Bit-banged” C Routines for the AVR<sup>®</sup> Microcontroller/ISP Code for the AT17CXXX FPGA Configuration Memories

## Features

- C Routines for Serial Interface
- Example Circuit for AVR Programming FPGA Configuration Memories
- No Interrupts
- User Programmable Speed
- Supports Entire AVR<sup>®</sup> Family
- Supports AT17CXXX and AT24CXXX Families of EEPROM Devices

## Introduction

This application note describes how to In-System Program (ISP) an Atmel FPGA Configuration memory (Configurator) using an Atmel AVR<sup>®</sup> Microcontroller and how to “bit-bang” the industry-standard 2-wire protocol that is needed to program the Configurator using port pins on an AT90S8515 AVR Microcontroller. The AT17C series of Configurators, ranges in density from 64K bits to 2M bits.

Users should be familiar with the AT90S8515 and AT17 Series datasheets and the application note titled, “Programming Specification for Atmel’s Configuration Memory EEPROMS AT17C65/128/256/512/010/020”. This application note is written specifically for the 1M bit device, but can be easily modified for other AT17 series family members. C routines to read and write data are included. The code can easily be recompiled for all AVR controllers with SRAM.

## Theory of Operation

The industry-standard 2-wire protocol bus is a two-wire synchronous serial

interface consisting of one data (SDA) and one clock (SCL) line. The industry-standard 2-wire protocol bus is a multi-master bus where one or more devices, capable of taking control of the bus, can be connected. When there is only one master connected to the bus, the resulting code is much simpler because handling of bus contentions and inter master access (a master accessing another master) is not necessary. Only master devices can drive both the SCL and SDA lines while a slave device is only allowed to issue data on the SDA line.

## Software Description

The generic industry-standard 2-wire protocol routines listed below were compiled without optimization using IAR’s C Compiler Version 1.30A (<http://www.iar.se>). These routines implement a single master, industry-standard 2-wire protocol implementation and are available for download from <http://www.atmel.com/atmel/products/prod185.htm>. The AT90S8515 used to perform the master function is clocked by an external 7.3728 MHz crystal. The routine Bit Delay is executed in 15 clock cycles or a 2.03  $\mu$ s period and provides the quarter period bit timing necessary to meet the 3.3V timing specifications found in the above referenced Configurator programming application note.

This code uses PORTB of the AT90S8515. On power-up PORTB is initialized to all inputs with the internal pull-ups turned off, the external pull-ups pull the SDA and SCL lines high and the



## C Routines for the AVR<sup>®</sup> Microcontroller/ AT17CXXX ISP Code

## Application Note

Rev. 1298A–04/20/99



PORTB output latch bits SCL and SDA are initialized to zero. Routine WriteSDA and WriteSCL toggle their respective data direction register bit depending on the value of parameter "state". When state is a "1" the port pin is configured as input (external pull-ups pull high). When state is a "0" the port pin is configured as an output and the latch drives the pin low. Table 1 lists the generic industry-standard 2-wire protocol routines and the amount of code space consumed by each. WriteSDA and WriteSCL are very simple routines that could be incorporated into their respective calling routines to further reduce code size.

Table 1 lists the number of clock cycles consumed while implementing the function. Compiler options were set to default, i.e. no AVR specific optimizations.

**Table 1.** Size and Execution Time for Industry-standard 2-wire Protocol Routines

Routine	Clocks	Bytes
SendStartBit	138	24
SendByte	1050 - 1054	74
SendStopBit	110	18
BitDelay	15	18
SetSCLHigh	33	40
WriteSCL	12 - 13	12
WriteSDA	12 - 13	12
GetByte	1089 - 1090	68

## General Calling Sequence for the Industry-standard 2-wire Protocol Routines

### Write

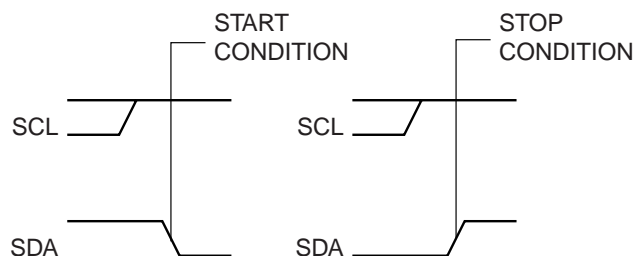
```
SendStartBit() // start
SendByte(byte,msbfirst) // send address MSB first
SendByte(byte,lsbfirst) // send data byte to that
                        address LSB first
SendStop() // stop
```

### Read

```
SendStartBit() // start
SendByte(byte,msbfirst) // send address MSB first
byte = GetByte(lastbyte) // read byte from that
                        address ... last byte
                        = 1 for the last byte
                        in a serial stream
SendStop() // stop
```

The routines SendStartBit, SendByte, and GetByte all leave the SCL signal low on exit, allowing the next routine to write data to SDA. (The industry-standard 2-wire protocol allows changes on SDA only when SCL is low; otherwise the industry-standard 2-wire protocol device will interpret a start or stop condition). SendByte returns a flag indicating a successful write to the industry-standard 2-wire protocol slave, 0 x 01 signals that the slave did not acknowledge the transfer and that something is wrong on the industry-standard 2-wire protocol bus or with the slave. WritePage and ProgramResetPolarity use this flag for data polling. Figure 1 shows industry-standard 2-wire protocol start and stop bit conditions.

**Figure 1.** Industry-standard 2-wire Protocol Start and Stop Bits



The AT17C010 device is programmed/verified on a 128-byte page boundary. During normal FPGA configuration operations, the read of the device starts at address 0 and continues until the FPGA has completed reading its configuration. The routines WritePage and ReadPage write and read 128-byte pages from the configuration memory and use the generic industry-standard 2-wire protocol routines to perform this function. WritePage and ReadPage are both called with the page address to write/read to and a pointer to a 128-byte page buffer. At the end of a page write the data polling methodology is used to determine the end of the internal page programming cycle. ProgramResetPolarity and VerifyResetPolarity write and read data from memory locations 0 x 20000 – 0 x 20003 in effect setting and verifying the RESET/ $\overline{OE}$  polarity.

**Table 2.** Code Size and Execution Time for Page Read/Write

Routines	Cycles	Bytes
WritePage	287872	152
ReadPage	145901	122
ProgramResetPolarity	156104	102
VerifyResetPolarity	157269	90

PORTB on the AT90S8515 is used to communicate with the FPGA Configuration Memory. Bit assignments are as follows:

```
// PB0 = SDA
// PB1 = SCL
// PB2 = SER_EN - used to put AT17C010 in serial
programming mode
```

The routine Init.c initializes the AT90S8515 peripherals. Routines Timer0.c and Timer1.c are general-purpose time out routines. Main is used to call WritePage, ReadPage, ProgramResetPolarity, and VerifyResetPolarity and serves to illustrate proper calling conventions for those routines.

```
Init.c
Main.c
Timer0.c
Timer1.c
```

## Modifications and Optimizations

### Impact on Changing Crystal Frequency

If the user decides to change oscillator frequencies then the following routines would have to be modified:

```
BitDelay
ProgramResetPolarity
SetSCLHigh
WritePage
```

BitDelay uses NOP's to effect a quarter bus period delay. Add or subtract NOP's to increase or decrease the delay. In the routines ProgramResetPolarity and WritePage timer 1 is used to time-out after 20 milliseconds, the programming operation should have completed by then. Timer 0 is used in SetSCLHigh to time out after 35 microseconds. If the SCL line is not high by then, then something is wrong on the bus.

### Impact on Changing PAGE\_SIZE

PAGE\_SIZE is defined in at17c.h and initially set to 128 to support the AT17C512/010/020 devices. PAGE\_SIZE can be changed to support other configuration memories or other generic industry-standard 2-wire protocol devices. Routines effected are:

```
Main
ReadPage
WritePage
```



## **Atmel Headquarters**

### *Corporate Headquarters*

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

### *Europe*

Atmel U.K., Ltd.  
Coliseum Business Centre  
Riverside Way  
Camberley, Surrey GU15 3YL  
England  
TEL (44) 1276-686-677  
FAX (44) 1276-686-697

### *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**

### *Atmel Colorado Springs*

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

### *Atmel Rousset*

Zone Industrielle  
13106 Rousset Cedex  
France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

---

### *Fax-on-Demand*

North America:

1-(800) 292-8635

International:

1-(408) 441-0732

### *e-mail*

literature@atmel.com

### *Web Site*

<http://www.atmel.com>

### *BBS*

1-(408) 436-4309

## **© Atmel Corporation 1999.**

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.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

1298A-04/20/99/xM