

Modbus Slave Stack for the Atmel Family of SAM3 Microcontrollers (Free Modbus Stack from Embedded Solutions)

1. Scope

This application note provides directions and instructions to application engineers adding support for Modbus™ Slave Stack into an industrial application using one of Atmel's ARM® Cortex-M3® based SAM3 devices. The application note does not describe the Modbus protocol. Even though the Atmel SAM3S device and the SAM3S-EK board are used as references in this application note, the Modbus Slave Stack can be easily ported to other devices in Atmel's SAM3 family of Flash MCUs.

2. Associated Documentation and Software

Before going further into this document, refer to the latest documentation for the corresponding SAM3 device on the Atmel web site.

http://www.atmel.com/products/at91/default.asp?family_id=605

3. Keywords

Atmel, ARM Cortex-M3, Modbus, fieldbus, Modicon, Remote Terminal Unit (RTU), Slave Stack, Electronic Industries Alliance (EIA), Industrial Embedded Solutions, FreeModbus.

4. Prerequisites and Requirements

Readers must have knowledge of the IAR Embedded Workbench®, SAM3S Software examples, IAR Embedded Workbench® 5.5 and SAM3S Softpack 2.1, plus patches installed.



**AT91SAM
ARM-based
Flash MCU**

Application Note



5. Overview

Modbus is a well established, robust and powerful communication protocol widely used in the industrial manufacturing market. The protocol is based on master-slave communication. Physical medium can be over an asynchronous RS232 link for single master-slave communication. Network topology is also supported (single master <--> multiple slaves) with RS485 physical medium. Both RS232 and RS485 are based on the serial communication link (EIA Standard).

Modbus was introduced in 1979 by Modicon (now owned by Schneider Electric). Modbus is an openly published and royalty-free communication standard.

For further details about Modbus, see [Section 10.1 “References” on page 10](#).

6. Implementation

The Modbus Slave Stack implemented on the SAM3S-EK provides support for the Modbus ASCII/RTU mode over the on-board RS232 serial link or RS485 link.

The stack is a free Slave Stack implementation, FreeModbus [2], provided by Embedded Solutions [3].

Embedded Solutions designs, develops and produces innovative electronic products for the industrial market and provides commercial, fully-featured, Master and Slave Stack.

FreeModbus stack provides an implementation of the Modbus Application Protocol v1.1a and supports RTU/ASCII transmission modes as defined in the Modbus over serial line specification 1.0.

The following Modbus functions are currently supported:

- Read Input Register (0x04)
- Read Holding Registers (0x03)
- Write Single Register (0x06)
- Write Multiple Registers (0x10)
- Read/Write Multiple Registers (0x17)
- Read Coils (0x01)
- Write Single Coil (0x05)
- Write Multiple Coils (0x0F)
- Read Discrete Inputs (0x02)
- Report Slave ID (0x11)

FreeModbus also supports Modbus TCP as defined in Modbus Messaging on TCP/IP Implementation Guide v1.0a.

All source code for the port on the SAM3S-EK board is available on www.freemodbus.org. A standalone demo and a FreeRTOS based demo are available. User guide and example are also available and well documented.

7. Installing the FreeModbus Stack Atmel Project

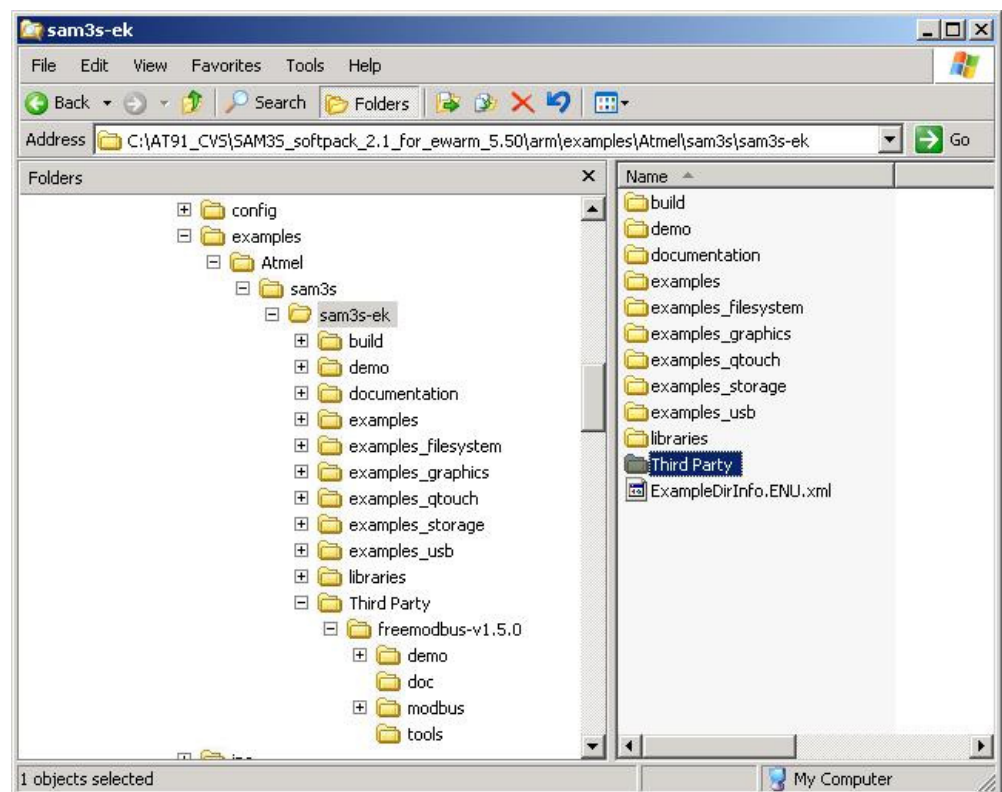
The standalone FreeModbus Slave Stack example is available from the link provided at [2]. It has been slightly modified to add, thanks to the Read Input Register (0x04) function, the reading of:

- the QTouch® Slider
- the SAM3S internal temperature sensor
- the on-board potentiometer.

On-board LEDs can be switched ON and OFF using the Write Input Register (0x03) function.

The modified example is provided in the software package associated with this application note. The project is built-up with IAR Embedded Workbench® for ARM 5.50.5 and SAM3S-EK IAR EWARM Software Package [8].

The original FreeModbus Slave Stack example is available from the link provided at [2]. Once the SAM3S-EK IAR Software Package has been extracted, decompress *atmel_sam3sek_modbus.zip* into the root directory of the software package. It should look like the following folder structure.



8. Running the Application Example

By default, the FreeModbus Slave Stack example uses the RS485 Link of the SAM3S-EK Board.

It requires a wired connection to a host processor and a Modbus master software on the PC side to be useful. Demo versions of Modbus Master Stacks can be found in the links provided at [4], [5], [6] and [7].

The SAM3S-EK evaluation board needs to be modified as described below to enable RS485 data transmission. Place Jumpers JP11, JP10 and JP12 and place a solder drop across R25. The RS485 shares data lines with USART1. It is therefore absolutely necessary to set PA23 to a high level. It is handled by the application.

To run the application with the RS485 link, an RS485 to RS232 adapter is necessary since a personal computer does not feature an RS485 port.

Note: Using the RS485 link on the board is relevant when building multiple slaves (sam3s board) with one master (PC).

For a point-to-point connection, the RS232 standard link can be used. In that case, JP11, 10, 12 and R25 footprint must be open. The USART port used is USART0 on J5 connector and J4 connector for RS485 port.

To run the example in RS232 mode, the portserial.c file must be modified.

Original code lines:

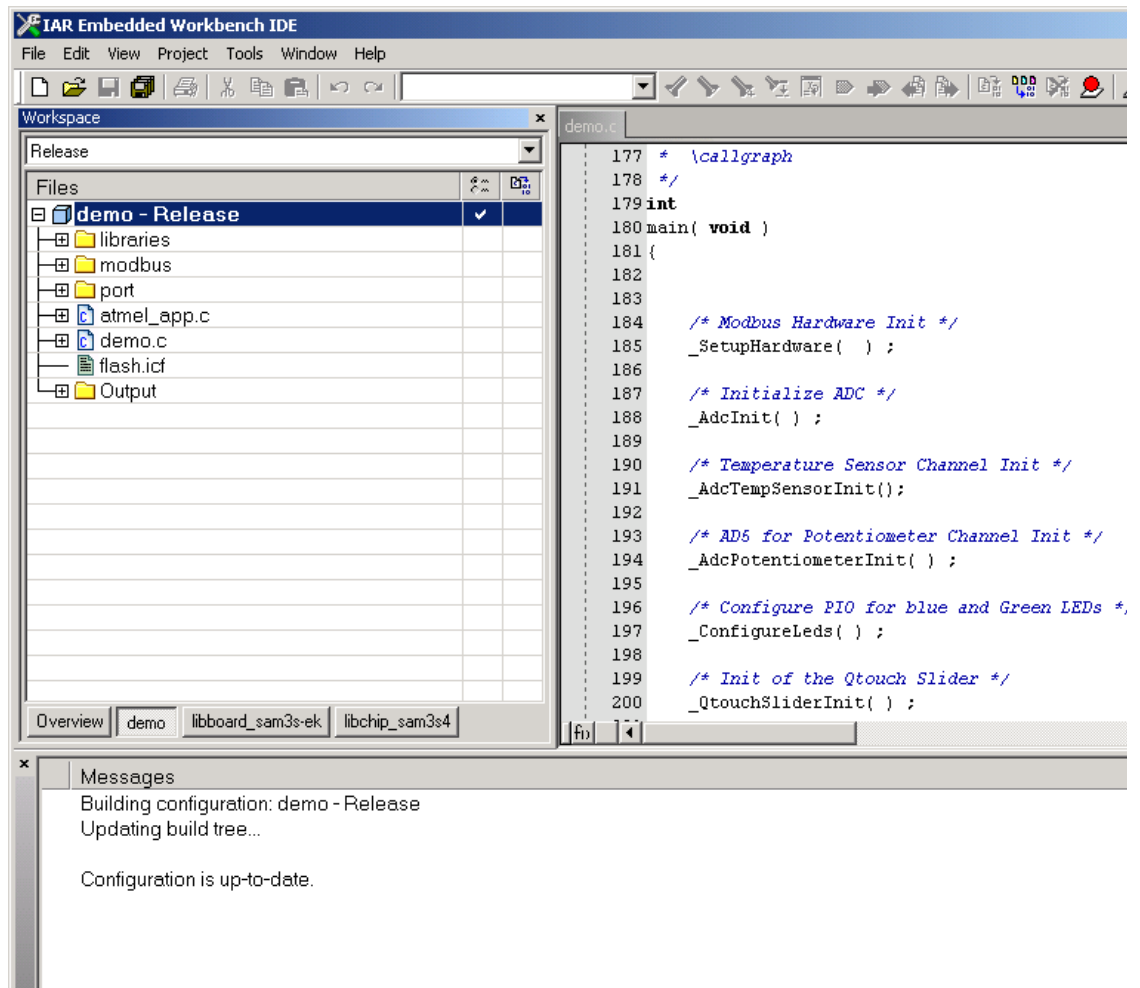
```
/* Choose RS232 (one slave only) or RS485 (one or several slaves) */
/* For RS485, see readme.txt file for board settings*/
#define RS485_ENABLED          ( 1 )
#define RS232_ENABLED          ( 0 )
```

Code lines modified:

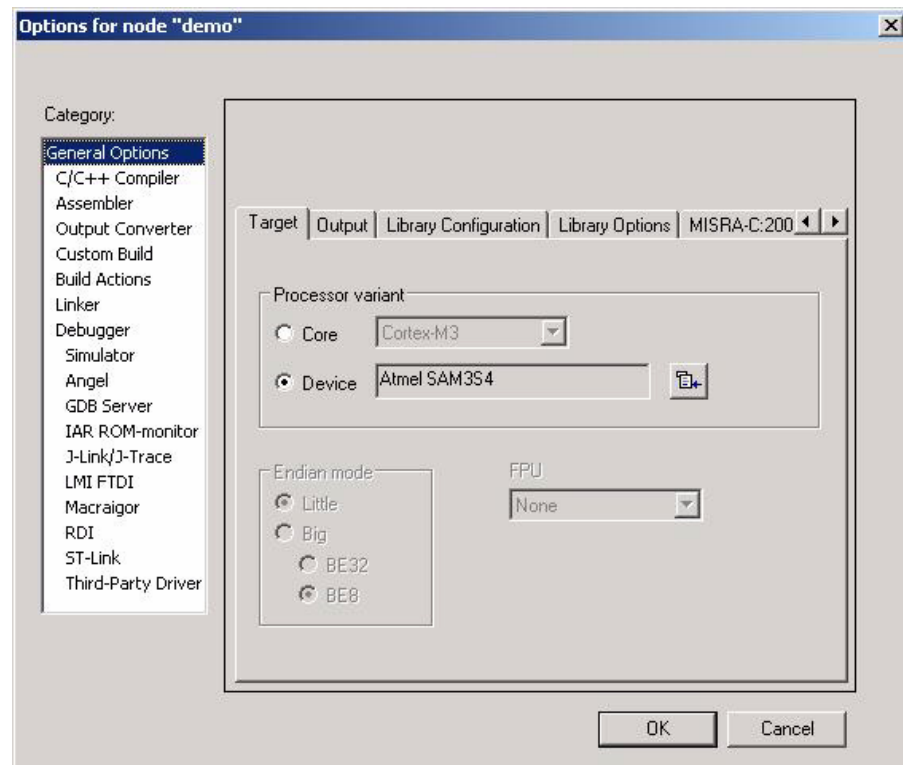
```
#define RS485_ENABLED          ( 0 )
#define RS232_ENABLED          ( 1 )
```

Open the IAR demo.eww project. It may happen that, when opening the project, the *Processor variant* setting is not set correctly.

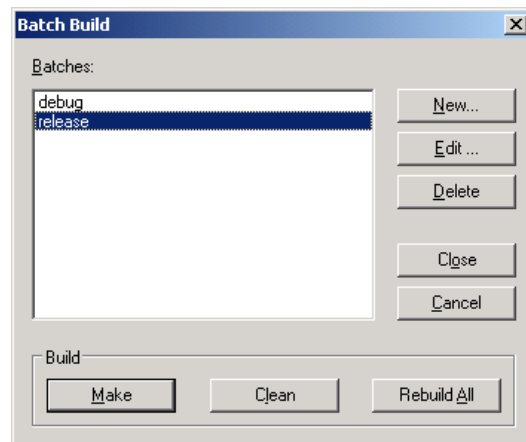
Right-click on *demo - Release* as shown below.



Verify that the *Device* is set as *Atmel SAM3S4*, as in the screen below.

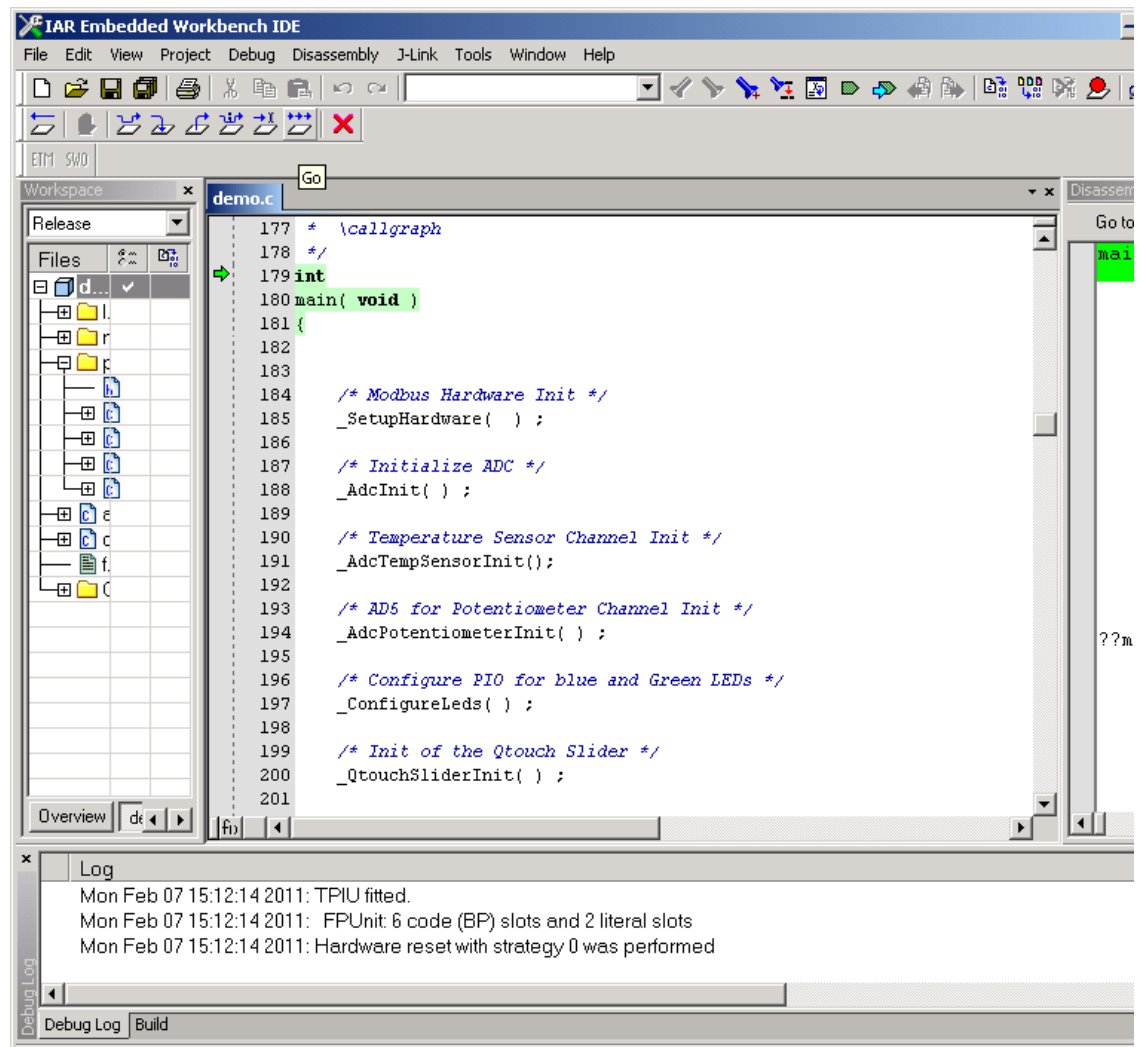


Click *OK* and rebuild the project by pressing *F8* key and choose either *debug* to run from SRAM, or *release* to run from FLASH.



If there are no errors, press *CTRL + D*. The application is ready to run in FLASH. Press *F5* key. If the *debug* variant must be used to run the application from SRAM, all the settings above must be checked too.

Then, in the debugger window, click on *Go*.



8.1 Modbus Master Software

Any Modbus Master software can be used. The one used in this example is Modbus Doctor.

Start the Modbus Doctor tool and use the following general configuration.

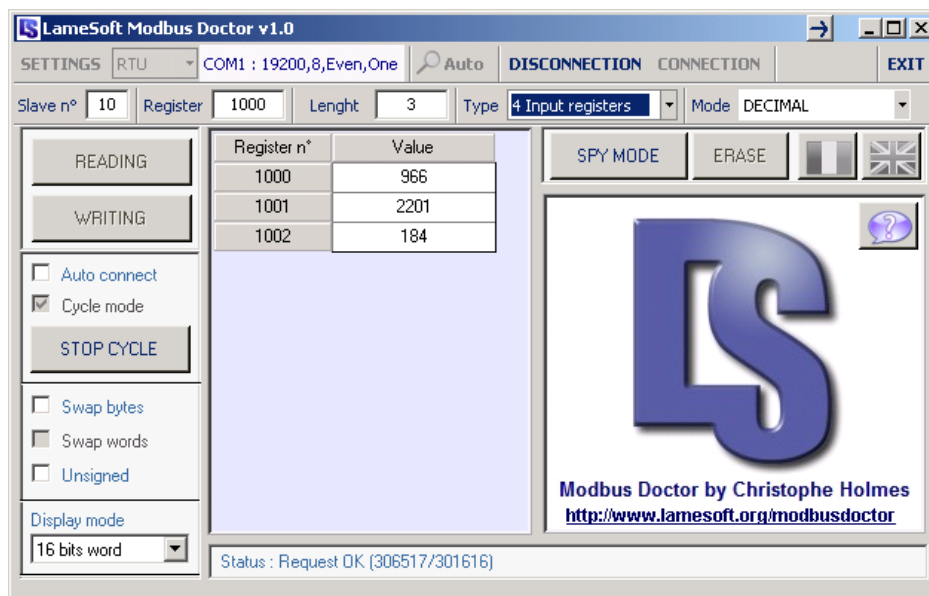
- *SETTINGS = RTU*

Then click *CONNECTION*.

8.1.1 Reading Input Registers

Specific settings must be as follows.

- *Slave n°* = 10
- *Register* = 1000
- *Length* = 3
- *Type* = 4 Input registers
- *Mode* = DECIMAL (if HEXADECIMAL is used, *Slave n°* and *Register* values must be converted into hexadecimal values)



The mapping of the Input registers in the board is:

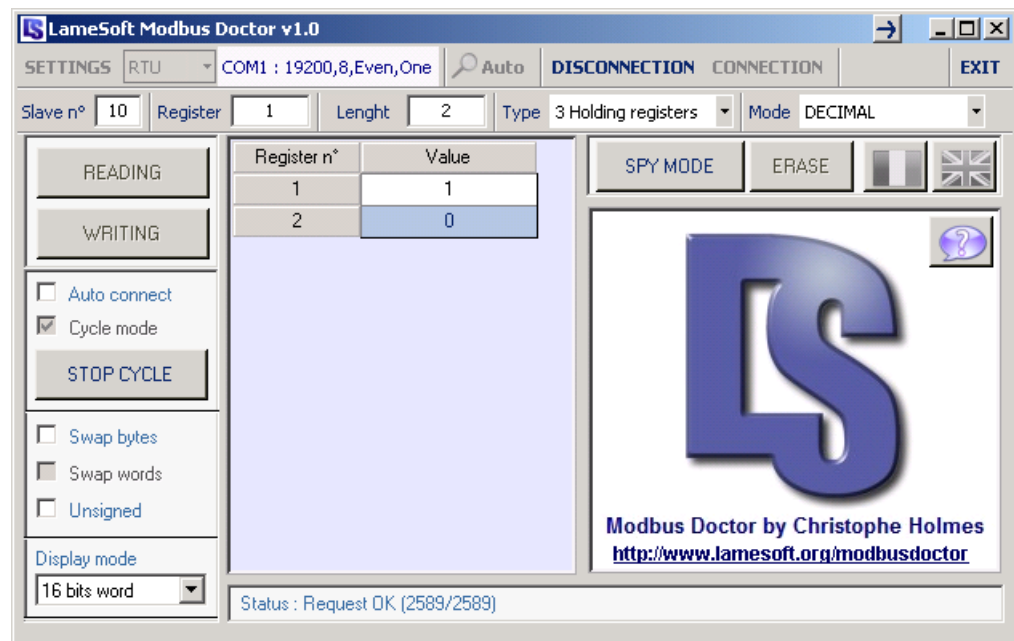
- 1000 = SAM3S temperature sensor (ADC channel number 15)
- 1001 = On-board potentiometer (ADC channel number 5)
- 1002 = QTouch slider position

To start reading Input registers, check *Cycle mode* and click *READING*. Move your finger on the slider or turn the potentiometer to see data reading.

8.1.2 Writing Holding Registers

Specific settings must be as follows.

- *Slave n° = 10*
- *Register = 1*
- *Length = 2*
- *Type = 3 Holding registers*
- *Mode = DECIMAL* (if *HEXADECIMAL* is used, *Slave n°* and *REGISTER* values must be converted into hexadecimal values)
- Check *Cycle mode*



The mapping of the Holding registers is:

- 1 = Blue LED (D2)
- 2 = Green LED (D3)

To turn ON the LEDs, write 1 in the corresponding register(s) and click on *WRITING*.

To turn OFF the LEDs, write 0 in the corresponding register(s) and click on *WRITING*.

9. Conclusion

By means of the FreeModbus Slave Stack, a rapid Modbus protocol implementation can be added to an embedded system for the industrial market. A fully-featured Slave and Master Commercial Stack is available from Embedded Solutions [3].

10. Appendix

10.1 References

1. Modbus:
 - Modbus Organization, <http://www.modbus.org/>
 - Modicon Modbus Protocol Reference Guide, PI-MBUS-300, Rev. J, June 1996, MODICON, Inc., USA, <http://www.modicon.com>
2. FreeModbus - A Modbus ASCII/RTU and TCP implementation.
<http://www.freemodbus.org/>
 - SAM3S-EK port: <http://www.freemodbus.org/index.php?idx=111>
 - Slave Stack download page: <http://www.freemodbus.org/index.php?idx=5>
3. Embedded Solutions, DI Christian Walter. <http://www.embedded-solutions.at/>
4. WinTech ModScan32: <http://www.win-tech.com/html/modscan32.htm>
5. Modbus Poll: http://www.modbustools.com/modbus_poll.asp
6. FieldTalk Modpoll: <http://www.focus-sw.com/fieldtalk/modpoll.html>
7. Modbus Doctor, Christophe HOLMES, <http://kikos31.developpez.com/modbusdoctor/>
<ftp://ftp-developpez.com/kikos31/modbusdoctor/fichiers/ModbusDoctor.zip>
8. SAM3S-EK IAR EWARM 5.4 Software Package (14 MB, revision 2.1, updated 9/10), http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4705

Revision History

Doc. Rev	Comments	Change Request Ref.
11098A	First issue	



Headquarters

Atmel Corporation

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

International

Atmel Asia Limited

Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH

Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel 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

Product Contact

Web Site

www.atmel.com
www.atmel.com/AT91SAM

Technical Support

AT91SAM Support
Atmel technical support

Sales Contacts

www.atmel.com/contacts/

Literature Requests

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.



© 2011 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, QTouch® and others, are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. ARM®, ARMPowered® logo, Cortex™-M3 are registered trademarks or trademarks of ARM Ltd. Other terms and product names may be trademarks of others.