AVR303: SPI-UART Gateway

Features

- **Communicate with a SPI slave device using a RS232 terminal**
- **SPI Modes 0–3, bit order and slave select (SS) is supported**
- **SPI clock frequency 28kHz to 1.8MHz when using a 3.6864MHz crystal**
- **Single byte transfer or multi-byte sequences**
- **Auto toggle of SS pin for accurate data transfer simulation**

8-bit AVR Microcontrollers

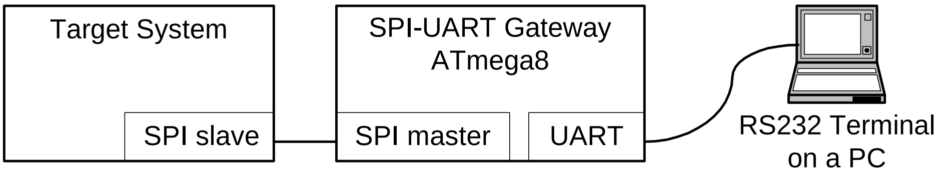
Application Note

1 Introduction

The Serial Peripheral Interface (SPI) is an excellent method of providing inter-processor communications for multi-processor embedded systems. However, concurrent development of the master and slave applications can result in increased test and debugging time, due to both ends of the communication link being unproven software. The SPI-UART Gateway application allows the developer to test and debug an SPI slave application isolated from the master, using manually controlled communications via a suitable RS232 terminal; for example a serial port terminal emulator running on a PC.

The SPI-UART Gateway is designed to run in master mode on an ATmega8(L) device, and allows all of the configurable parameters of the hardware SPI module to be modified. This means that communication should be possible with any SPI slave device, including other AVR microcontrollers, any other SPI-capable micros from Atmel or 3rd party manufacturers, and any dedicated SPI peripheral devices.

Figure 1-1. SPI-UART Gateway.



Rev. 2557A-AVR-03/05

2 Software Description

Full source code is provided for build under IAR C compiler 2.28a or later. Users of other compiler packages must translate the code to their respective format, although the SPI-UART Gateway application is stand-alone and should require no modification. A hex file of the compiled software is also provided.

The software has been constructed in three distinct sections, and is split into three C source files. Individual functions will not be described in detail here, but the comments in the source code should give a good overview of what the functions are doing.

2.1 MAIN.C

This file contains the user interface handler, and demonstrates many useful techniques for building a UART based user interface system. In particular, the storage and handling of string literals should be noted, as these are stored in flash and form a significant percentage of the total flash space used. Other features that may be useful as code templates for other applications include ASCII to binary conversions, basic menu systems, and the checksum-verified EEPROM storage of configuration data.

2.2 SPI.C

This short file is based on the example code provided in the Atmel data sheets for the ATmega8 and other AVRs that utilize the hardware SPI module. The two resulting functions handle the initialization and transmission of data through the SPI interface, with the initialization function modified to support set up of the Slave Select pin.

2.3 USART2.C

This file contains an implementation of the interrupt driven USART ring-buffer code described in application note AVR306. This version is virtually identical to the original AVR306 version, the only differences being the separation of header information into a separate file (USART2.h) and changing of the frame format from 2 stop bits to 1 stop bit. A receive buffer overflow error handler has not been implemented, as it is not required.

3 Hardware Implementation

The SPI-UART Gateway application is perfectly suited for running on an STK500 AVR development board. An ATmega8 or ATmega8L should be fitted in the SCKT3200A2 (green) socket and programmed with the application hex file using the normal programming procedure. All fuses should be in their default state, except for clock selection, which should be “Ext. Clock; Start-up time: 6 CK + 64mS”. Ensure the STK500 oscillator is set to “3.69MHz” and that the board voltage is set to a level compatible with the target slave device. Note that the target cannot be another AVR device on the same STK500, a second development board or custom circuit will be required.

Link PD0 to RXD and PD1 to TXD on the “RS232 Spare” header, the link to the RS232 terminal is then made using the STK500 “RS232 Spare” 9-way D socket. Serial communication is 19200 Baud, using 8 bit data, no parity and one stop bit.

Connection to the target slave device is via the port pins as shown in Table 1-1.

Table 1-1. Connections to target slave device.

Signal	Port Pin	Direction
SCK	PB5	Output
MISO	PB4	Input
MOSI	PB3	Output
SS	PB2	Output
GND	GND	na

4 How to use the SPI-UART Gateway

The RS232 terminal or terminal emulator should be configured as an ANSI type with no local echo of characters typed, and with no appending of line feeds to incoming line ends. Following connection to the terminal, reset of the SPI-UART Gateway processor will result in a welcome message being displayed on the terminal. If an SPI configuration had previously been stored in EEPROM, then this will be loaded and the status of the major parameters will be displayed, otherwise a default configuration will be used. Finally, a command prompt will be shown:

4.1 Command: [H]-help >

Typing ‘H’ will display a help screen showing all the available commands (which are not case sensitive) with descriptions for alternative configurations where necessary. Commands available are as follows:

4.2 Command: 0-9,A-F

Entering a valid hex character prepares the gateway to expect a second hex character, the combination of which will form a single byte for transmission through the SPI port. If the second character typed is not a valid hex character, the sequence will be aborted. A valid hex sequence will result in the byte being immediately transmitted through the SPI port. The hex value transmitted will be shown on the terminal, together with a hex representation of the byte received by the SPI module as a result of the transmission.

4.3 Command: ?

This will display the current status of the SPI module configuration on the terminal. There will be no activity on the SPI port.

4.4 Command: X

This indicates that a number of hex encoded bytes are to be entered and transmitted sequentially through SPI with minimal delay between bytes. Up to sixteen bytes can be entered, and transmission will begin when either sixteen byte values have been entered or the ‘Enter’ key has been pressed. During entry, any character typed that is not a valid hex value will cause the entire sequence to be aborted. If only one hex character has been entered when the ‘Enter’ key is pressed, that character will be ignored and any previous whole byte values will be transmitted. The hex bytes transmitted will be shown on the terminal, together with a hex representation of the byte received by the SPI module as a result of each individual transmission. The Slave Select pin can also be configured to ‘auto toggle’ at the beginning and end of transmission of X-command sequences. See the ‘P’ command for details.

4.5 Command: Q

This is used to set the SCK frequency, with eight possible values being selected by the second character typed (0-7). For a mega8 running from a 3.6864MHz crystal, the SCK frequencies available are displayed on the help screen, and range from 28.8kHz to 1.8432MHz.

4.6 Command: S

This gives manual control over the level of the Slave Select pin. Typing ‘S0’ will make the pin go low, typing ‘S1’ will make the pin go high.

4.7 Command: P

This is used to set the ‘Slave Select pin auto toggle’ mode, which is only used in conjunction with the ‘X’ command. Auto toggle will more accurately simulate a master to slave transmission sequence by first ensuring the SS pin is in the idle (inactive) state, then toggling SS to active before transmission of the first byte. The SS pin remains active until the last byte of the sequence has been transmitted, after which it will return to the idle state. Typing ‘P0’ turns on auto toggle with an active low SS pin, typing ‘P1’ turns on auto toggle with an active high SS pin. Typing ‘P2’ turns off auto toggle mode.

4.8 Command: T

This sets the bit order for SPI transmission. Typing ‘T0’ will mean the Most Significant Bit of the transmission and reception bytes is transferred on the first SCK cycle. Typing ‘T1’ will mean the Least Significant Bit is transferred first. Changing the bit order will not result in any immediate activity on the SPI port.

4.9 Command: M

This sets the SPI Mode, which affects the relationship between clock edges and data-transfer activity. The help screen gives a textual description of these relationships, however a graphical representation is provided in the mega8 data sheet, which should be referred to if clarification is required. All four possible SPI modes can be configured. Changing the SPI mode will not result in any immediate activity on the SPI port, and following a mode change the idle state of the SCK pin may be incorrect for that mode until a transmission has taken place.

4.10 Command: W

This will save the current configuration by writing it to the mega8 EEPROM. Following power-up or reset, the saved values will be automatically loaded.

5 About the author

This software and applications note have been written by Andy Gayne, an Atmel certified Field Applications Engineer for GD Technik Ltd. UK, part of the ATeG group of companies.

Email: avr@gayne.co.uk or andy.gayne@gd-technik.com

Web Site: http://www.gd-technik.com

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 Arsenaux 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

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. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.