AVR305: Half Duplex Compact Software UART

AIMEL

8-bit **AVR**® Microcontrollers

Application Note

Features

- 32 Words of Code, Only
- Handles Baud Rates of up to 38.4 kbps with a 1 MHz XTAL
- Runs on Any AVR Device
- · Only Two Port Pins Required
- · Does Not Use Any Timer

1 Introduction

In many applications utilizing serial communication, the software handling communication does not have to be performed as a background task. This application note describes how to implement polled software UART capable of handling speeds up to 614,400 bps on an AT90S1200. This implementation is intended for applications where small code is preferred. All bit delays are software delays, so no timer is required. The controller cannot perform any other tasks while receiving or transmitting, but when the operation is complete, the controller is free to continue program execution.

Rev. 0952C-AVR-0905





2 Theory of Operation

In asynchronous serial communication, no clock information is transferred. The data is transferred serially, one bit at a time. In Idle state, the line is held at logical "1". When data is to be transferred, the first bit is a so-called start bit, often given the symbol S. The start bit is a "0", causing a "1" to "0" transition at the line. This can be detected by the receiver, signaling that data is coming. The following bits transferred are the data bits, with the LSB first. Then, one or more stop bits (P) are transferred. The stop bit is a logical "1", putting the line back to idle state before a new start bit followed by a data byte can be transferred. There must be at least one stop bit, so that a "1" to "0" transition can be detected by the receiver, indicating a new start bit. The *frame* shown in Figure 2-1 has got eight data bits and one stop bit. Sometimes, a parity bit is included between the last data bit and the stop bit, and there can be several stop bits.

Figure 2-1. Frame Format



The Receiver must sample the data line in the middle of every bit in order to receive the data properly. The bit length has to be equal for all bits, so that the Receiver knows when to sample the line. The Receiver is synchronized with the Transmitter by the falling edge of the start bit. The Receiver must have an internal timer in order to sample the bits at the right time.

The bit length must be the same for both transmitter and receiver, and some standard speeds are defined in bits per second, *bps*.

3 Implementation

3.1 Bit Length Delays - UART_delay

The delay between the bits are generated by calling a delay subroutine twice (as the subroutine generates a half-bit delay, see receiving data). If very short delays are required (when transmitting and receiving at very high speed), the delay must be implemented inside the *putchar* and *getchar* routines. The required delay length in clock cycles can be calculated using the following equation:

$$c = \frac{f_{CLCL}}{baud\ rate}$$

where c is the bit length in clock cycles and f_{CLCL} is the crystal frequency.

Both *putchar* and *getchar* use nine CPU cycles to send or receive a bit. Hence, a delay of c - 9 cycles must be generated between each bit. The *rcall* and *ret* instructions require a total of seven cycles. If the subroutine is to be called twice to generate the required delay, the delay has to be d cycles:

$$d = \frac{c-9}{2} - 7$$

If the delay is generated as shown below, the total execution time is 3-b cycles plus seven cycles for *rcall* and *ret*.

rcall UART_delay
UART_delay: ldi temp,b
UART_delay1: dec temp
brne UART_delay1
ret

The value *b* is found by the equation:

$$b = \frac{c-9}{2} - 7 = \frac{c-23}{6}$$

The actual delay generated, calling delay twice is

$$d = (3 \times b + 7) \times 2 + 9 = 6 \times b + 23$$

From this, the minimum and maximum delays are $d_{min} = 29$ and $d_{max} = 1,559$ cycles. The c and b values for some bit rates and frequencies are shown in Table 3-1.

Table 3-1. "UART_delay" Subroutine Performance Figures

Parameter	Value	
Code Size	4 words	
Execution Cycles	Min: 7 cycles Max: 772 cycles (including ret)	
Register Usage	Low Registers High registers Global	: None : None : 1

Table 3-2. "UART_delay" Register Usage

Register	Input	Internal	Output
R17	-	"temp" – count variable	-





3.2 Transmitting Data - putchar

The *putchar* subroutine transmits the byte stored in the register *Txbyte*. The data bits are shifted into the Carry bit. The easiest way to generate stop bits is to let the zeros shifted into the transmitted byte be interpreted as ones. If the data byte is inverted before it is shifted, a "0" in Carry must give a "1" on the line, and a "0" in Carry gives a "1" on the line. When 0's are shifted into the data byte, they are handled as 1's. This way, any number of stop bits can be generated by just repeating the transmittal loop. The start bit is generated by setting the carry bit before data are shifted out.

Table 3-3. "putchar" Subroutine Performance Figures

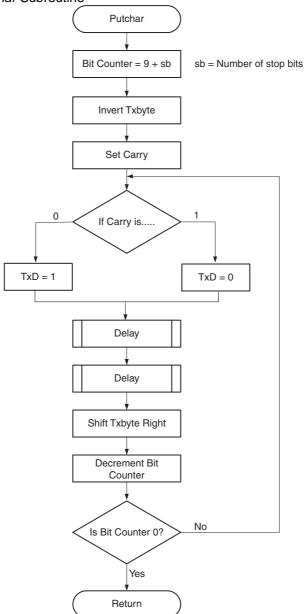
Parameter	Value	
Code Size	14 words	
Execution Cycles	Depends on bit rate	
Register Usage	Low Registers High registers Global	: None : None : 2

Table 3-4. "putchar" Register Usage

Register	Input	Internal	Output
R16	-	"bitcnt" – counts the number of bits transferred	1
R18	"Txbyte – the byte to send	-	-

The algorithm for transmitting data is shown in Figure 3-1

Figure 3-1. putchar Subroutine





3.3 Receiving Data - getchar

First, the routine waits for a logical "0" (not a transition). When the start bit is detected, a 1.5 bit delay is generated. This is performed by calling the delay subroutine three times. Sampling then starts at 1-bit intervals. The Carry is set or cleared according to the logic value at the RxD pin. If less than eight data bits are received, the Carry is shifted into the receive byte. If not, the routine returns with the received byte in *Rxbyte*.

The routine waits for one bit length after the last data bit and terminates in the middle of the stop bit. This is done to prevent detection of a false start bit if the routine is called again immediately after a complete reception.

Table 3-5. "getchar" Subroutine Performance Figures

Parameter	Value	
Code Size	14 words	
Execution Cycles	Waits until byte received	
Register Usage	Low Registers High registers Global	: None : None : 2

Table 3-6. "getchar" Register Usage

Register	Input	Internal	Output
R16	-	"bitcnt" – counts the number of bits received	-
R18	-	-	"Rxbyte" – the received byte

The algorithm for receiving data is shown in Figure 3-2.

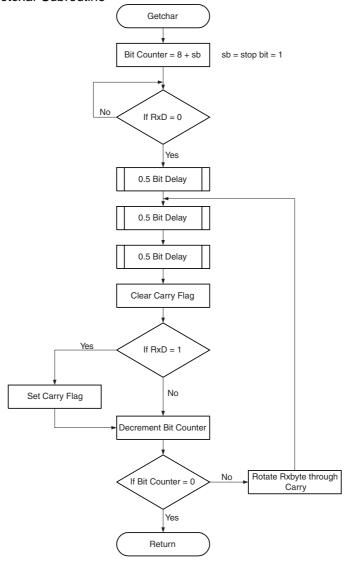


Figure 3-2. getchar Subroutine

4 Example Program

The example program receives a character with *getchar* and echoes it back with *putchar*.

Table 4-1. Overall Performance Figures

Parameter	Value	
Code Size	32 words - UART routines onl 40 words - Complete applicati	•
Register Usage	Low Registers High registers Global	: None : 4 : None
Interrupt Usage	None	
Peripheral Usage	Port D pin 0 and 1 (any two pins can be used)	





Table 4-2. Baud Rate Table

1	M	H
	IVI	П

Cycles BaudRate b-value Error % required 0.6 2400 417 66 4800 208 31 0.3 9600 104 14 2.7 14400 69 8 2.2 19200 52 5 1.8 28800 35 2 0.8 57600 115200

1.8432 MH

1.8432 MHZ			
BaudRate	Cycles required	b-value	Error %
2400	768	124	0.1
4800	384	60	0.3
9600	192	28	0.5
14400	128	18	2.3
19200	96	12	1.0
28800	64	7	1.6
57600	32	2	⁽¹⁾ 9.4
115200	-	-	-

2 MHz

Cycles required	b-value	Error %
833	135	0.0
417	66	0.6
208	31	0.3
139	19	1.4
104	14	2.7
69	8	2.2
35	2	0.8
-	-	-
	833 417 208 139 104	required b-value 833 135 417 66 208 31 139 19 104 14 69 8

2.4576 MHz

BaudRate	Cycles required	b-value	Error %
2400	1024	167	0.1
4800	512	82	0.6
9600	256	39	0.4
14400	171	25	1.4
19200	128	18	2.3
28800	85	10	2.7
57600	43	3	⁽¹⁾ 3.9
115200	-	-	-

3.276 MHz

3.276 WITZ			
BaudRate	Cycles required	b-value	Error %
2400	1365	224	0.1
4800	683	110	0.0
9600	341	53	0.1
14400	228	34	0.2
19200	171	25	1.4
28800	114	15	0.7
57600	57	6	⁽¹⁾ 3.7
115200	28	1	2.0

3.6864 Mhz

BaudRate	Cycles required	b-value	Error %
2400	1536	252	0.1
4800	768	124	0.1
9600	384	60	0.3
14400	256	39	0.4
19200	192	28	0.5
28800	128	18	2.3
57600	64	7	1.6
115200	32	2	⁽¹⁾ 9.4

4 MHz

4 WIT12			
BaudRate	Cycles required	b-value	Error %
2400	1667	⁽²⁾ 274	0.0
4800	833	135	0.0
9600	417	66	0.6
14400	278	42	1.0
19200	208	31	0.3
28800	139	19	1.4
57600	69	8	2.2
115200	35	2	0.8

4.608 MHz

4.006 WITZ			
BaudRate	Cycles required	b-value	Error %
2400	1920	⁽²⁾ 316	0.1
4800	960	156	0.1
9600	480	76	0.2
14400	320	50	0.9
19200	240	36	0.4
28800	160	23	0.6
57600	80	10	⁽¹⁾ 3.8
115200	40	3	2.5

7.3728 MHz

7.3728 MHZ				
BaudRate	Cycles required	b-value	Error %	
2400	3072	⁽¹⁾ 508	0.0	
4800	1536	252	0.1	
9600	768	124	0.1	
14400	512	82	0.6	
19200	384	60	0.3	
28800	256	39	0.4	
57600	128	18	2.3	
115200	64	7	1.6	

8 MHz

BaudRate	Cycles required	b-value	Error %
2400	3333	⁽²⁾ 552	0.0
4800	1667	(2)274	0.0
9600	833	135	0.0
14400	556	89	0.3
19200	417	66	0.6
28800	278	42	1.0
57600	139	19	1.4
115200	69	8	2.2

9.216 MHz

BaudRate	Cycles required	b-value	Error %
2400	3840	⁽²⁾ 636	0.0
4800	1920	⁽²⁾ 316	0.1
9600	960	156	0.1
14400	640	103	0.2
19200	480	76	0.2
28800	320	50	0.9
57600	160	23	0.6
115200	80	10	⁽¹⁾ 3.8

11.059 MHz

11.059 MHz			
BaudRate	Cycles required	b-value	Error %
2400	4608	⁽²⁾ 764	0.0
4800	2304	⁽²⁾ 380	0.0
9600	1152	188	0.1
14400	768	124	0.1
19200	576	92	0.2
28800	384	60	0.3
57600	192	28	0.5
115200	96	12	1.0

14.746 MHz

BaudRate	Cycles required	b-value	Error %
2400	6144	⁽²⁾ 1020	0.0
4800	3072	⁽²⁾ 508	0.0
9600	1536	⁽²⁾ 252	0.1
14400	1024	167	0.1
19200	768	124	0.1
28800	512	82	0.6
57600	256	39	0.4
115200	128	18	2.3

16 MHz

10 1411 12			
BaudRate	Cycles required	b-value	Error %
2400	6667	⁽²⁾ 1107	0.0
4800	3333	⁽²⁾ 552	0.0
9600	1667	⁽²⁾ 274	0.0
14400	1111	181	0.2
19200	833	135	0.0
28800	556	89	0.3
57600	278	42	1.0
115200	139	19	1.4

Notes:

- 1. Baud rates over 3 % are likely to cause communication errors
- The provided delay routine will not give a valid result for b-values larger than 255 when using 8-bits registers.



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

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. 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 the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.