APPLICATION NOTE

Atmel AT03335: Manchester Transceiver Using the USART and XCL Modules on XMEGA E

Atmel AVR XMEGA

Features

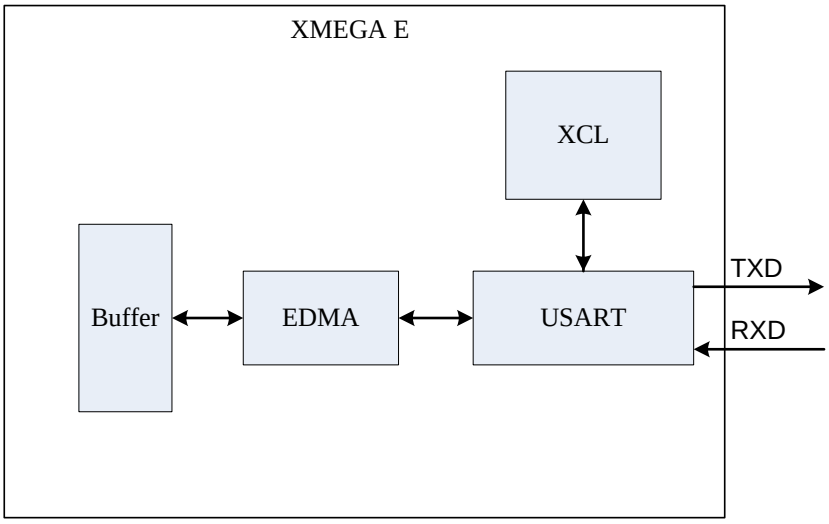
- Manchester Encoding and decoding
- Demonstrate advantage of combining USART and XCL (XMEGA® Custom Logic)
- EDMA(Enhanced Direct Memory Access) used for data transmission

Introduction

Manchester coding is a coding technique widely used in telecommunication (e.g. DALI, RFID, Near Field Communication and IrDA). The encoding of bits results in at least one transition for each bit and the encoded signal will then be a self-clocking signal which means that the clock signal can be recovered from the data stream.

This application note describes one approach to set up the Atmel® AVR® XMEGA E as Figure 1 to do Manchester decoding and encoding with USART and XCL hardware modules.

Figure 1. Connection Diagram of Transceiver



Another method of Manchester decoding and encoding is using Timer and GPIO modules. Timer affords the time base of Manchester clock and GPIO pins are used to sense or generate Manchester code. As comparison, XMEGA E takes advantage of hardware resource USART, XCL and EDMA which reduce remarkably the load of firmware.

Table of Contents

1. Related Items

The following list contains links to the most relevant documents:

- Atmel AVR XMEGA E Manual
  ATxmega E devices used in this solution.
- XMEGA E Using the XCL Module
  XCL is a new module on the XMEGA E devices.

2. Manchester code

Manchester coding was first developed and published by G.E. Thomas in 1943. This was the first convention of Manchester coding and is known as the G.E. Thomas convention. This was followed by a convention used in low speed Ethernet standards and is known as the IEEE 802.3 convention

2.1 G. E. Thomas convention

The G. E. Thomas convention of Manchester encoding states that a bit value of “1” is a transition from “1” to “0” and a bit value of “0” is a transition from “0” to “1”.

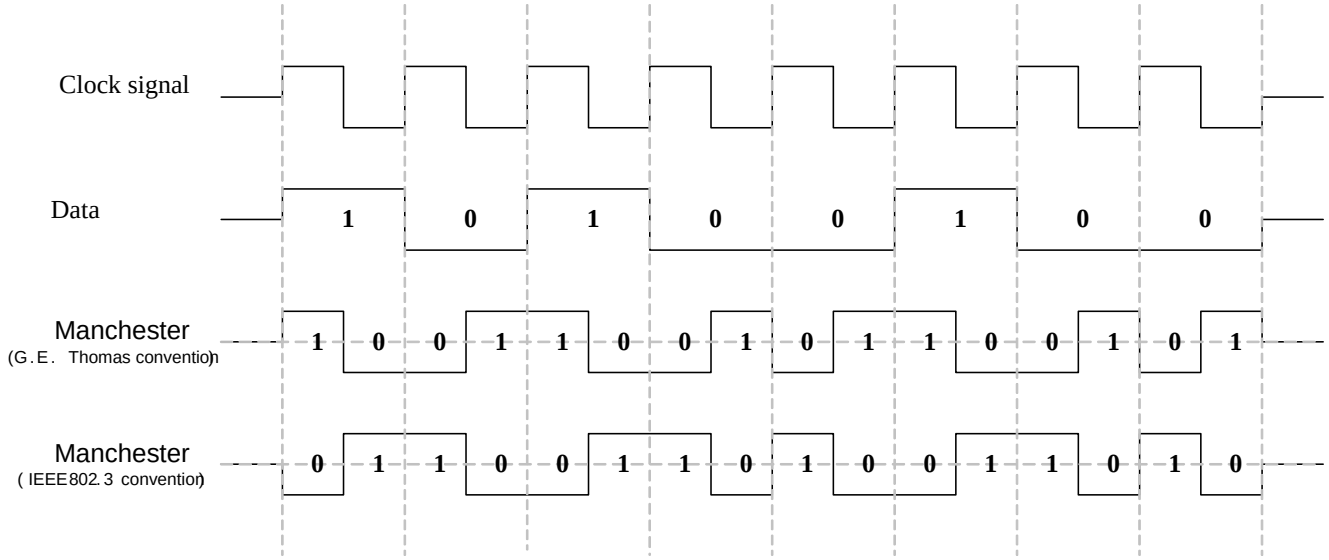
The encoding of the data can be done simply by using XNOR between the data and the clock signal. Decoding of the Manchester code can be done in the same way, by using XNOR between the Manchester data and the clock signal.

2.2 IEEE 802.3 convention

The IEEE 802.3 convention of Manchester encoding states that a bit value of “1” is a transition from “0” to “1” and a bit value of “0” is a transition from “1” to “0”.

The encoding of the data can be done using XOR between the data and the clock signal. The decoding of the data can be done by using XOR between the Manchester code and the clock signal.

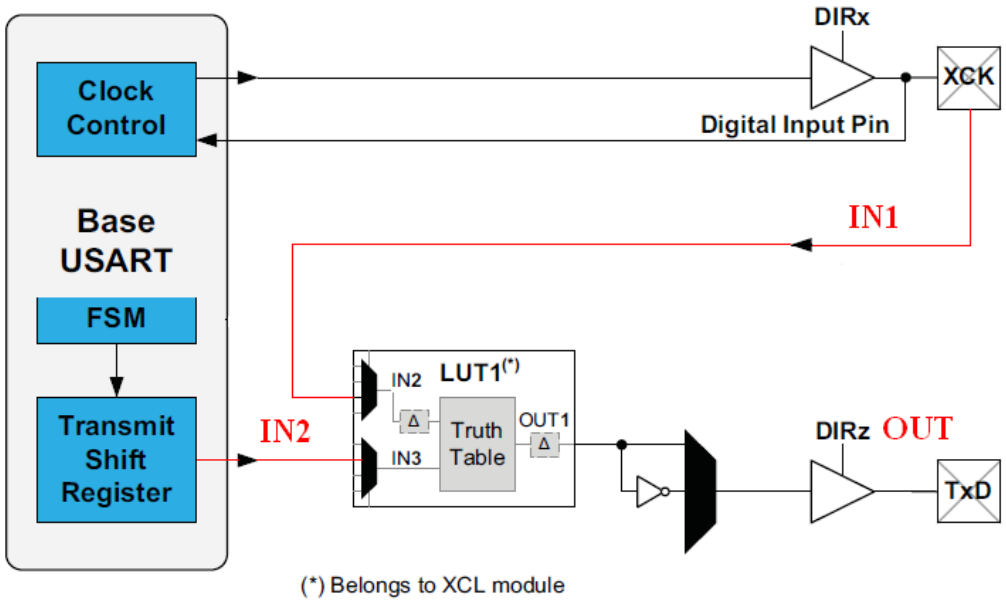
Figure 2-1. Signal Encoding Conventions



3. Manchester encoding

For Manchester encoding, USART needs to operate in synchronous mode. SCK of USART is used as Manchester clock signal. Transmit data from shift register is used as Manchester data. To encode the data, XCL linked with USART is used to execute XNOR logic for G. E. Thomas convention and XOR for IEEE 802.3 convention. The logic output from XCL connects to USART TXD pin.

Figure 3-1. Manchester encoding structure



3.1 How to configure USART

1. Set control register C (CTRLC)

7	6	5	4	3	2	1	0	
CMODE[1:0]		PMODE[1:0]		SBMODE	CHSIZE[2:0]			
CMODE[1:0]		–	–	–	UDORD	UCPHA	–	(1)

Note: 1. Master SPI mode

Bit 7:6 – CMODE[1:0]: Communication Mode

Select synchronous USART or master SPI communication mode provided with clock output. Which one to be selected depends on the data frame structure. For example, DALI data frame contains one start bit and two stop bits, so synchronous USART mode suits for it.

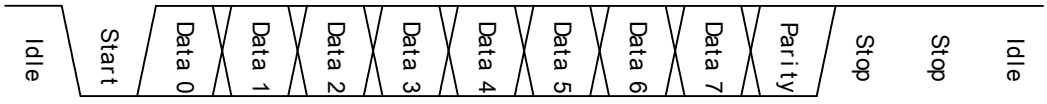
Bit 5:4 – PMODE[1:0]: Parity Mode

Bit 3 – SBMODE: Stop Bit Mode

Bit 2:0 – CHSIZE[2:0]: Character Size

For synchronous USART mode, then select parity mode, stop bit mode and character size. Note that the lsb of the frame data word is transmitted first. But in DALI frame, data msb is first so the bits should be reordered before transmission.

Figure 3-2. USART typical data transmission



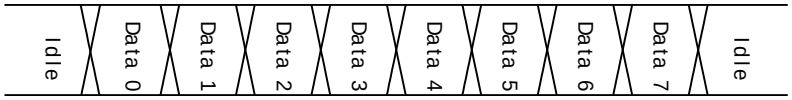
Bit 2 – UDORD: Data Order

Bit 1 – UCPHA: Clock Phase

For master SPI mode, select data order and clock phase. Data order sets the frame format. When written to one, the data word lsb is transmitted first. When written to zero, the msb of data word is transmitted first. In variable data length mode controlled by PEC (peripheral counter) from XCL, this bit

must be set to one. The clock phase determines whether data are setup on the leading (first) edge or the trailing (last) edge of XCK. The communication data line always keeps high in idle state.

Figure 3-3. SPI typical data transmission



2. Set control register D (CTRLD)

7	6	5	4	3	2	1	0
–	–	DECTYPE[1:0]		LUTACT[1:0]		PECACT[1:0]	

Bit 5:4 – DECTYPE[1:0]: Decoding and encoding type

Encoding type can configure LUT (lookup table units) OUT applies during data field only or during start and data field, or Inverted LUT OUT applies during start field while LUT OUT during data field. The last type is selected when Manchester code start bit is 1 such as DALI.

Bit 3:2 – LUTACT[1:0]: LUT Action

Encoding on transmitter engine should be enabled.

Bit 1:0 – PECACT[1:0]: Peripheral Counter Action

Transmitter data length should be controlled by PEC1.

3. Set baud rate control register (BAUDCTRLA BAUDCTRLB)

7	6	5	4	3	2	1	0
BSCALE[3:0]				BSEL[11:8]			
7	6	5	4	3	2	1	0
BSEL[7:0]							

Bit 7:4 – BSCALE[3:0]: Baud Rate Scale Factor

When calculated BSEL is larger than 0xFFF, Baud Rate generator should be prescaled by 2^{BSCALE}.

Bit 3:0 – BSEL[11:8]: Baud Rate Bits

Bit 7:0 – BSEL[7:0]: Baud Rate Bits

This 12-bit value contains USART baud rate setting. For equation refer to the datasheet.

4. Set control register B (CTRLB)

7	6	5	4	3	2	1	0
ONEWIRE	SFDEN	–	RXEN	TXEN	CLK2X	MPCM	TXB8

Bit 7 – ONEWIRE: One-Wire Configuration Enabled

Bit 6 – SFDEN: Start Frame Detection Enable

Bit 4 – RXEN: Receiver Enable

Bit 3 – TXEN: Transmitter Enable

Setting to 1 enables the USART Transmitter.

Bit 2 – CLK2X: Double Transmission Speed

Bit 1 – MPCM: Multi-processor Communication Mode

Bit 0 – TXB8: Transmit Bit

3.2 How to configure XCL

1. Set control register A (CTRLA)

7	6	5	4	3	2	1	0
LUT0OUTEN[1:0]		PORTSEL[1:0]		–	LUTCONF[2:0]		

Bit 7:6 – LUT0OUTEN[1:0]: LUT0 Output Enable

Bit 5:4 – PORTSEL[1:0]: Port Selection

Select the corresponding USART used with PEC.

Bit 2:0 – LUTCONF[2:0]: LUT Configuration

Select the two independent 2-input LUT configuration.

2. Set control register B (CTRLB)

7	6	5	4	3	2	1	0
IN3SEL[1:0]		IN2SEL [1:0]		IN1SEL [1:0]		IN0SEL [1:0]	

Bit 7:0 – INxSEL[1:0]: Input Selection

Select USART TXD as IN3, XCK pin as IN2.

3. Set control register D (CTRLD)

7	6	5	4	3	2	1	0
TRUTH1[3:0]				TRUTH0[3:0]			

Bit 7:0 – TRUTHx[3:0]: LUT Truth Table

Select LUT1 truth table as XOR.

4. Set control register E (CTRLE)

7	6	5	4	3	2	1	0
CMDSEL	TCSEL[2:0]			CLKSEL[3:0]			

Bit 7 – CMDSEL: Command Selection

Bit 6:4 – TCSEL[2:0]: Timer/Counter Selection

PEC1 should be selected for USART transmitter.

5. Set peripheral length control register (PLC)

7	6	5	4	3	2	1	0
PLC[7:0]							

Bit 7:0 – PLC[7:0]: Peripheral Length Control Bits

Set the length as the USART Transmitter data length subtracts 1.

3.3 Other configuration

1. Enable the peripherals clock

   The bits in power reduction register should be clear to ensure peripherals enabled.
2. Set USART pins

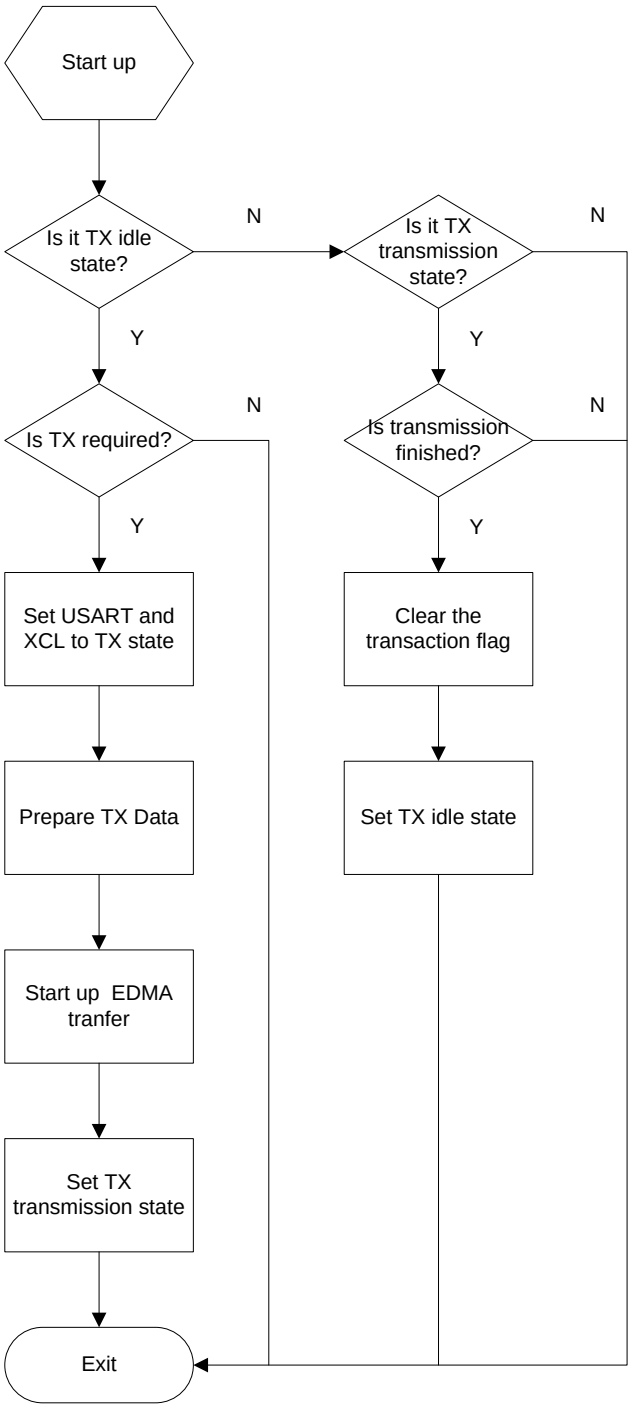
   Set TXD and XCK pin as output in I/O port register.
3. Set EDMA

   One EDMA channel can be enabled to transfer the data to the USART. This can be used to reduce the CPU load when larger data block to be transmitted. The EDMA channel is configured to have one byte burst length to transfer. The USART DRE (Data Register Empty) is used as transfer trigger source. For more information about EDMA, please refer to the datasheet.

3.4 Encoding flowchart

When Manchester encoding is required, USART and XCL should be configured to encoding state. Then prepare Manchester data and start up the EDMA transfer. After transmission, reset TX to idle for next encoding. Refer to Figure 3-4 flowchart.

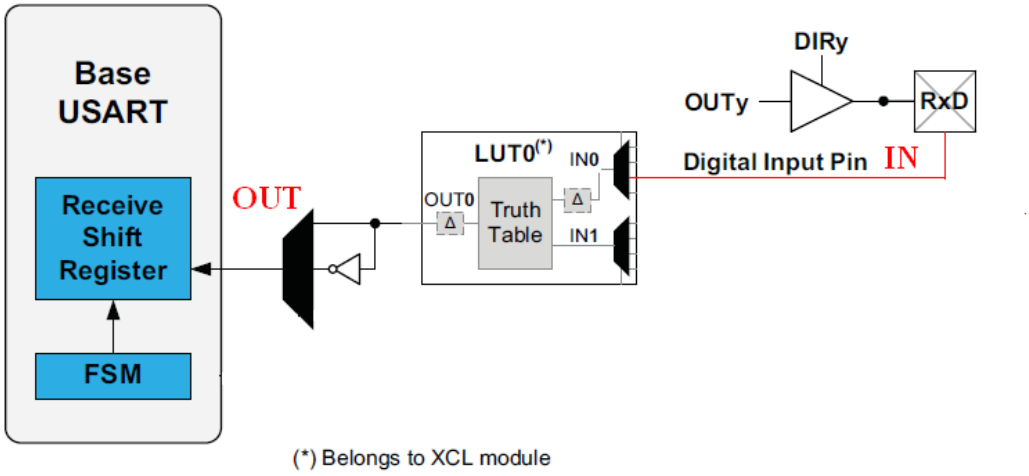
Figure 3-4. Manchester encoding flowchart



4. Manchester decoding

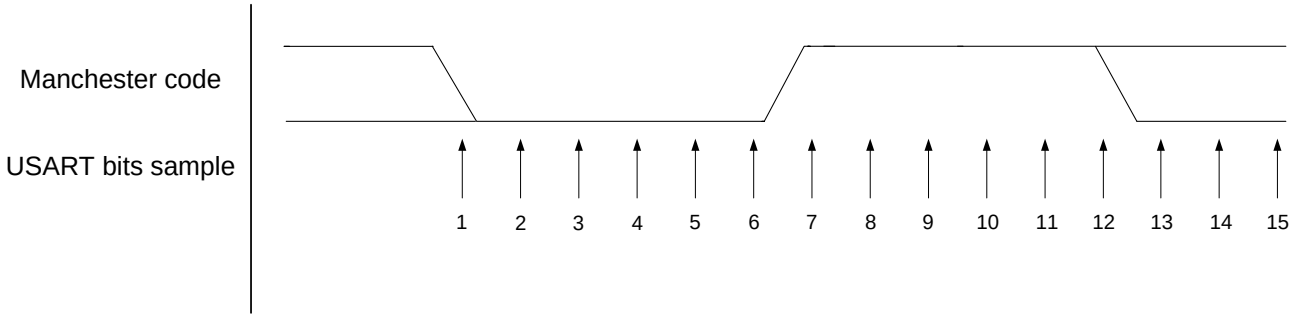
Since there is no synchronous clock signal input, it is impossible to decode with LUT XOR or XNOR logic as Manchester encoding directly. Manchester code can be sampled by UASRT received bits. USART runs in asynchronous RX mode for data reception and PLC of LUT controls the variable length of data bits stream. The maximum length of the stream is limited to 256 by PLC. EDMA can be used to transmit data from USART receiver register.

Figure 4-1. Manchester decoding structure



Manchester code can be over sample with a higher USART baud rate than Manchester clock. Then Manchester code is decoded from USART sampled bits by firmware. This way can set tolerance of Manchester clock rate error flexibly because the error is judged by firmware. Figure 4-2 illustrates a sample example with USART baud rate twelve times of Manchester code. The first six USART bits sample zero for Manchester low level and the next six sample one for Manchester high level. In this way the maximum number Manchester code can be decoded limits to 256/12.

Figure 4-2. USART bits sampling



4.1 How to configure USART

1. Set control register C (CTRLC)

7	6	5	4	3	2	1	0
CMODE[1:0]		PMODE[1:0]		SBMODE	CHSIZE[2:0]		

Bit 7:6 – CMODE[1:0]: Communication Mode

Select asynchronous USART to sample Manchester code.

Bit 5:4 – PMODE[1:0]: Parity Mode

Bit 3 – SBMODE: Stop Bit Mode

Bit 2:0 – CHSIZE[2:0]: Character Size

Set up according to Manchester code frame structure.

2. Set control register D (CTRLD)

7	6	5	4	3	2	1	0
–	–	DECTYPE[1:0]		LUTACT[1:0]		PECACT[1:0]	

Bit 5:4 – DECTYPE[1:0]: Decoding and encoding type

Bit 3:2 – LUTACT[1:0]: LUT Action

Bit 1:0 – PECACT[1:0]: Peripheral Counter Action

Receiver data length should be controlled by PEC0.

3. Set baud rate control register (BAUDCTRLA BAUDCTRLB)

7	6	5	4	3	2	1	0
BSCALE[3:0]				BSEL[11:8]			
7	6	5	4	3	2	1	0
BSEL[7:0]							

Bit 7:4 – BSCALE[3:0]: Baud Rate Scale Factor

Bit 3:0 – BSEL[11:8]: Baud Rate Bits

Bit 7:0 – BSEL[7:0]: Baud Rate Bits

To calculate asynchronous baud rate setting, refer to the equation in datasheet.

4. Set control register B (CTRLB)

7	6	5	4	3	2	1	0
ONEWIRE	SFDEN	–	RXEN	TXEN	CLK2X	MPCM	TXB8

Bit 7 – ONEWIRE: One-Wire Configuration Enabled

Bit 6 – SFDEN: Start Frame Detection Enable

Bit 4 – RXEN: Receiver Enable

Setting to 1 enables the USART Receiver.

Bit 3 – TXEN: Transmitter Enable

Bit 2 – CLK2X: Double Transmission Speed

Bit 1 – MPCM: Multi-processor Communication Mode

Bit 0 – TXB8: Transmit Bit

4.2 How to configure XCL

1. Set control register A (CTRLA)

7	6	5	4	3	2	1	0
LUT0OUTEN[1:0]		PORTSEL[1:0]		–	LUTCONF[2:0]		

Bit 7:6 – LUT0OUTEN[1:0]: LUT0 Output Enable

Bit 5:4 – PORTSEL[1:0]: Port Selection

Select the corresponding USART used with PEC.

Bit 2:0 – LUTCONF[2:0]: LUT Configuration

2. Set control register E (CTRLE)

7	6	5	4	3	2	1	0
CMDSEL	TCSEL[2:0]			CLKSEL[3:0]			

Bit 7 – CMDSEL: Command Selection

Bit 6:4 – TCSEL[2:0]: Timer/Counter Selection

PEC0 should be selected for USART receiver.

3. Set peripheral length control register (PLC)

7	6	5	4	3	2	1	0
PLC[7:0]							

Bit 7:0 – PLC[7:0]: Peripheral Length Control Bits

Set the length as the USART receiver data length subtracts 1.

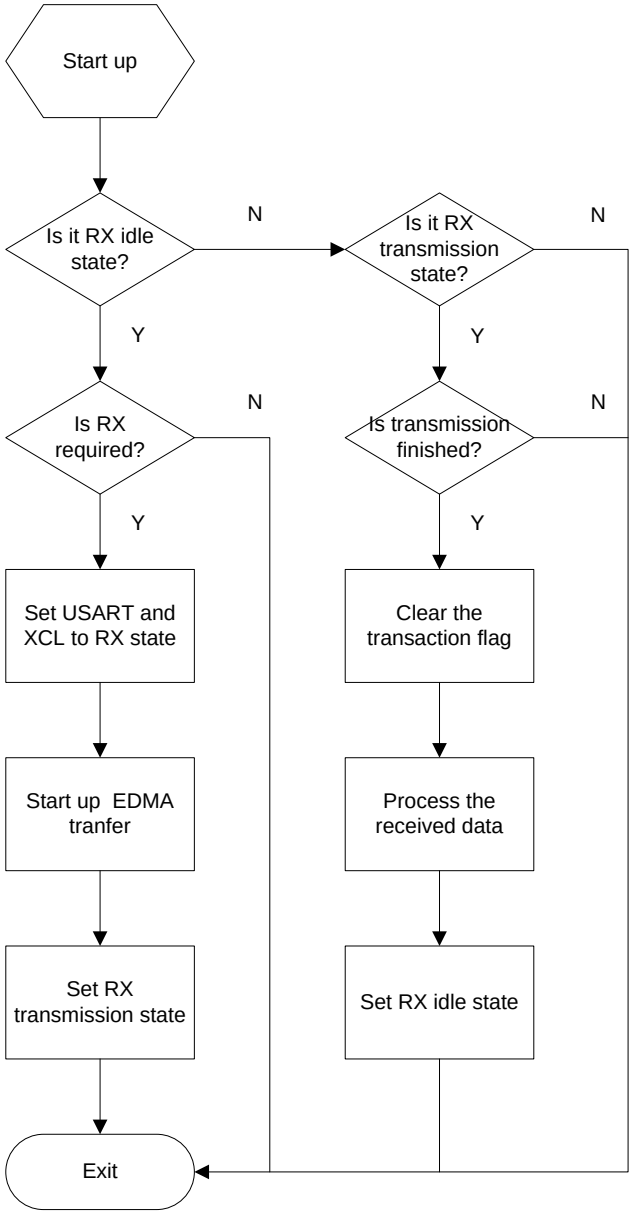
4.3 Other configuration

1. Enable the peripherals clock
   The bits in power reduction register should be clear to ensure peripherals enabled.
2. Set USART pins
   Set RXD pin as input in I/O port register.
3. Set EDMA
   One EDMA channel can be enabled to transfer the data from USART to data memory. The USART RXC(Receive Complete) is used as transfer trigger source.

4.4 Decoding flowchart

When Manchester decoding is required, USART and XCL should be configured to RX state. Then start up the EDMA transfer from USART receiver register to data buffer. After EDMA transmission is finished, the received data should be processed to get the Manchester data. See Figure 4-3 for the whole flowchart.

Figure 4-3. Manchester decoding flowchart



5. Revision History

Doc. Rev.	Date	Comments
42164A	07/2013	Initial revision

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

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

Atmel Japan G.K.
16F Shin-Osaki Kangyo Building
1-6-4 Osaki
Shinagawa-ku, Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370

© 2013 Atmel Corporation. All rights reserved. / Rev.: 42164A–AVR–07/2013

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, AVR®, XMEGA®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

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 THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, 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 AND 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 products 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 products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.