

EFM[®]32

... the world's most energy friendly microcontrollers

USART

AN0008 - Application Note

0 1 2 3 4

Introduction

This application note describes how to use the EFM32G USART both in synchronous (SPI) and asynchronous (UART) mode.

This application note includes:

- This PDF document
- Source files (zip)
 - Example C-code
 - Multiple IDE projects

1 USART

1.1 Introduction

The EFM32G Universal Synchronous Asynchronous serial Receiver and Transmitter (USART) is a very flexible serial communication module. It operates in either synchronous or asynchronous mode.

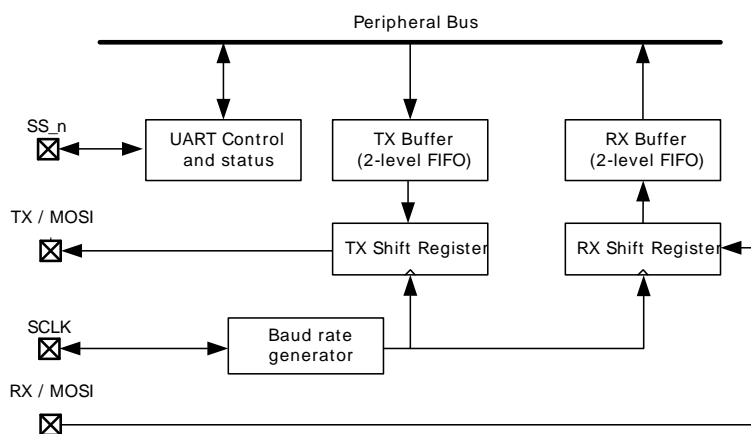
In synchronous mode, a separate clock signal is transmitted with the data. This clock signal is generated by the bus master, and both the master and slave sample and transmit data according to this clock. Both master and slave modes are supported by the USART. The synchronous communication mode is compatible with the Serial Peripheral Interface Bus (SPI) standard.

In asynchronous mode, no separate clock signal is transmitted with the data on the bus. The USART receiver thus has to determine when to sample the data on the bus. To make this possible, additional synchronization bits are added to the data when operating in asynchronous mode, resulting in a slight overhead.

1.2 Operation Overview

An overview of the USART is shown in Figure 1.1 (p. 2).

Figure 1.1. USART Overview



Transmission is enabled by writing to the TXEN bit in the USARTn_CMD register. Any data written to the TX buffer will be transmitted. The USART performs this by first moving the data to the shift register, from which the data is shifted to the TX pin. When data has been moved from the TX buffer, the TXBL bit in the USARTn_STATUS register is set, indicating that a new transmit byte may be written. When all available transmit data (both in the shift register and in the TX buffer) have been transmitted, the TXC bit in the USARTn_STATUS register is set. Both the TXC and TXBL are also available as interrupt flags.

When the RXEN bit in the USARTn_CMD register has been written, data received on the RX pin will be accepted. When new data is available from in the RX buffer, the RXDATAV bit in the USARTn_STATUS register is set. This bit will be cleared when the RX buffer is read. The RXDATAV is also available as an interrupt.

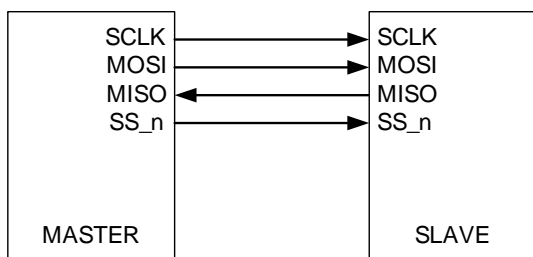
For a detailed description of the USART please refer to the EFM32G Reference Manual.

2 Synchronous (SPI) Mode

2.1 Theory

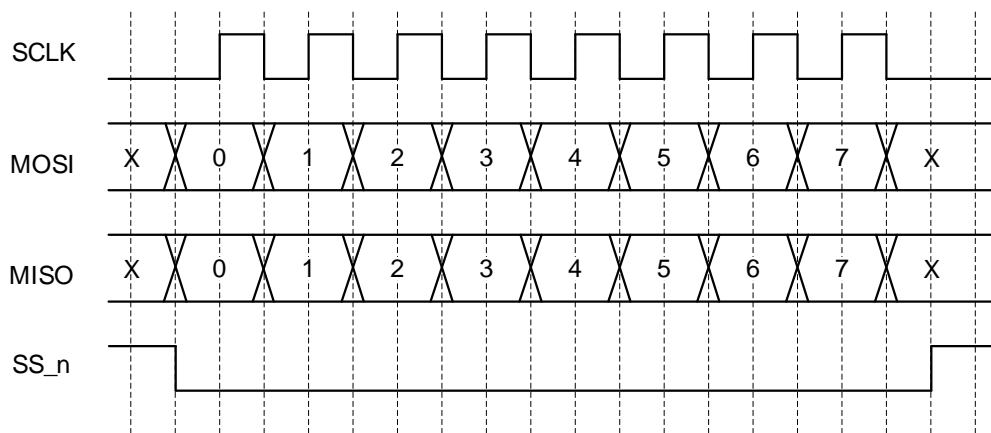
The synchronous communication is normally set up using 4 lines; clock (SCLK), 2 data lines and slave select (SS_n, also known as chip select, CS), see Figure 2.1 (p. 3). The data lines consist of MOSI (Master Out Slave In) and MISO (Master In Slave Out). These lines are driven by the master and the slave, respectively.

Figure 2.1. Typical SPI Setup



A typical synchronous transaction is shown in Figure 2.2 (p. 3). The master initiates the transfer by asserting the slave select line. Then the clock is driven by the master, and both master and slave shift data onto their data line while sampling the other (e.g. The slave drives the MISO line while listening to the MOSI line). This enables data transmission in both directions simultaneously. When the master has received/sent the desired data, it terminates the transaction by deasserting the SS_n line.

Figure 2.2. Typical SPI Transaction



The polarity of the signals as well as the sampling instant may be changed. Please see the EFM32G Reference Manual for further details.

2.2 SPI Example

The attached SPI example illustrates the use of the USART in synchronous mode. USART1 is configured as slave, whereas USART2 is master. The following transactions take place:

- Data transmission from master to slave.
- Data transmission from slave to master.
- Data transmission from master to slave and from slave to master simultaneously.

In order to make the attached SPI example work as intended, the IO lines of USART1 and 2 must be connected as specified in Table 2.1 (p. 4). If the EFM32G Development Kit (DVK) is used, these

connections are easy to implement on the Prototyping Board. USART1 uses IO location 1 for its pins, whereas USART2 uses IO location 0.

Table 2.1. SPI Connection Table

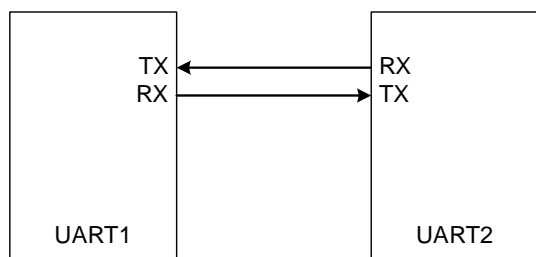
| Function | USART1 #1 | Protoboard Pin | USART2 #0 | Protoboard Pin |
|----------|-----------|----------------|-----------|----------------|
| MOSI | PD0 | P5.3 | PC2 | P4.5 |
| MISO | PD1 | P5.4 | PC3 | P4.6 |
| SCLK | PD2 | P5.5 | PC4 | P4.7 |
| SS_n | PD3 | P5.6 | PC5 | P4.8 |

3 Asynchronous (UART) Mode

3.1 Theory

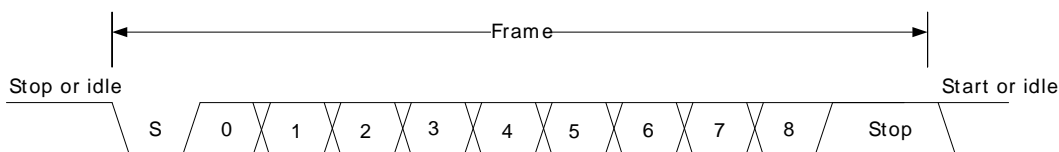
The asynchronous communication is normally set up using 2 data lines, one for transmission (TX) and one for reception (RX). The TX line of the first device is connected to the RX line of the second devices and vice versa, as seen in Figure 3.1 (p. 5). Please note that there is no master and slave in terms of setup. However, a higher level protocol may implement such a relationship.

Figure 3.1. Typical UART Setup



The asynchronous operation differs from synchronous operation in that both units provide their own timing, i.e. there is no master clock determining sampling instants. Therefore, the transmission speed (baud rate) must be agreed upon prior to the transaction taking place. This can be done by convention or by detection. A typical asynchronous transfer is shown in Figure 3.2 (p. 5) .

Figure 3.2. Typical UART Transaction



A typical transfer is initiated by a start bit followed by 8 data bits and one or two stop bits to terminate the frame. Transmission of each frame is initiated at senders will. The number of data and stop bits may vary, and it is also possible to insert a parity bit for light error detection. Please see the EFM32G Reference Manual for further details on configuration possibilities.

3.2 UART Example

The attached UART example illustrates the use of the USART in asynchronous mode. Both USART1 and USART2 are configured to use a baud rate of 115200, 8 data bits, no parity bit and one stop bit. The High Frequency Crystal Oscillator (HFXO) is used to increase the absolute accuracy of the baudrate. The following transactions are added:

- USART2 transfers data using polling. USART1 receives using interrupt.
- USART2 transfers data using interrupt. USART1 receives using polling.

In order to make the attached UART example work as intended, the IO lines of USART1 and 2 must be connected as specified in Table 3.1 (p. 5). If the EFM32G Development Kit (DVK) is used, these connections are easy to implement on the Prototyping Board.

Table 3.1. UART Connection Table

| Direction | USART1 #1 | Protoboard Pin | USART2 #0 | Protoboard Pin |
|-----------|-----------|----------------|-----------|----------------|
| 1 -> 2 | PD0 | P5.3 | PC3 | P4.6 |
| 2 -> 1 | PD1 | P5.4 | PC2 | P4.5 |

4 Revision History

4.1 Revision 1.06

2012-04-20

Adapted software projects to new peripheral library naming and CMSIS_V3.

Fixed a pin configuration issue with the uart example.

4.2 Revision 1.05

2012-03-14

Removed AF-Pin definitions from code example.

Fixed makefile-error for CodeSourcery projects.

4.3 Revision 1.04

2011-08-19

Corrected GPIO configuration for SPI slave.

4.4 Revision 1.03

2011-03-24

Changed the location description for USART1.

4.5 Revision 1.02

February 1st, 2011.

Changed GPIO configuration to reduce code size.

4.6 Revision 1.01

November 16th, 2010.

Changed example folder structure, removed build and src folders.

Added chip-init function.

Small code changes in spi.c

4.7 Revision 1.00

September 20th, 2010.

Initial revision.

A Disclaimer and Trademarks

A.1 Disclaimer

Energy Micro AS intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Energy Micro products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Energy Micro reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Energy Micro shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Energy Micro. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Energy Micro products are generally not intended for military applications. Energy Micro products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

A.2 Trademark Information

Energy Micro, EFM32, EFR, logo and combinations thereof, and others are the registered trademarks or trademarks of Energy Micro AS. ARM, CORTEX, THUMB are the registered trademarks of ARM Limited. Other terms and product names may be trademarks of others.

B Contact Information

B.1 Energy Micro Corporate Headquarters

| Postal Address | Visitor Address | Technical Support |
|---|---|---|
| Energy Micro AS P.O. Box 4633 Nydalen N-0405 Oslo NORWAY | Energy Micro AS Sandakerveien 118 N-0484 Oslo NORWAY | support.energymicro.com Phone: +47 40 10 03 01 |

www.energymicro.com

Phone: +47 23 00 98 00

Fax: + 47 23 00 98 01

B.2 Global Contacts

Visit **www.energymicro.com** for information on global distributors and representatives or contact **sales@energymicro.com** for additional information.

| Americas | Europe, Middle East and Africa | Asia and Pacific |
|--|--|--|
| www.energymicro.com/americas | www.energymicro.com/emea | www.energymicro.com/asia |

Table of Contents

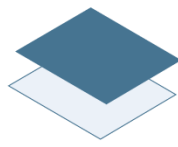
- 1. USART 2
 - 1.1. Introduction 2
 - 1.2. Operation Overview 2
- 2. Synchronous (SPI) Mode 3
 - 2.1. Theory 3
 - 2.2. SPI Example 3
- 3. Asynchronous (UART) Mode 5
 - 3.1. Theory 5
 - 3.2. UART Example 5
- 4. Revision History 6
 - 4.1. Revision 1.06 6
 - 4.2. Revision 1.05 6
 - 4.3. Revision 1.04 6
 - 4.4. Revision 1.03 6
 - 4.5. Revision 1.02 6
 - 4.6. Revision 1.01 6
 - 4.7. Revision 1.00 6
- A. Disclaimer and Trademarks 7
 - A.1. Disclaimer 7
 - A.2. Trademark Information 7
- B. Contact Information 8
 - B.1. Energy Micro Corporate Headquarters 8
 - B.2. Global Contacts 8

List of Figures

| | |
|-------------------------------------|---|
| 1.1. USART Overview | 2 |
| 2.1. Typical SPI Setup | 3 |
| 2.2. Typical SPI Transaction | 3 |
| 3.1. Typical UART Setup | 5 |
| 3.2. Typical UART Transaction | 5 |

List of Tables

| | |
|----------------------------------|---|
| 2.1. SPI Connection Table | 4 |
| 3.1. UART Connection Table | 5 |



ENERGY[®]
micro

*Energy Micro AS
Sandakerveien 118
P.O. Box 4633 Nydalen
N-0405 Oslo
Norway*

www.energymicro.com