Application Report

SLAA722 – October 2016

How to Use MSP430™ and MSP432™ eUSCI and USCI Modules in RS-485 Networks

MSP Systems

ABSTRACT

This application report is a guide to set up the eUSCI or USCI UART module on Texas Instruments (TI) MSP430™ and MSP432™ microcontrollers in half-duplex RS-485 networks.

The source code for the firmware described in this document can be downloaded from http://www.ti.com/lit/zip/slaa722.

Contents

List of Figures

1 Introduction

RS-485 is a protocol for multidrop communication operation. RS-485 uses differential signaling to achieve superior signal-to-noise characteristics and long range of communication. All RS-485 nodes must share a common ground, so a minimum of three wires are required for half duplex RS-485 communications: a pair of transceive wires and a common ground.

The details of RS-485 communication network are explained in *The RS-485 Design Guide*. Overview for RS-485 Transceivers provides detailed information about TI RS-485 transceiver products. This application report describes how to set up the eUSCI or USCI UART module on the MSP430 microcontrollers in half-duplex RS-485 networks.

2 RS-485 Requirements

In half-duplex RS-485 multidrop applications, a single pair of conductors is used for both transmission and reception between a *master* communicating with multiple *slave* serial devices, or *nodes*. To avoid contention on the RS-485 bus, the application software must assure that only one transmitter is enabled at a time. The master is in charge of designating which receiver is on at any one time. In the simplest scheme, all RS-485 transceivers come up in receive mode when the interface is initialized, and each transceiver node has a unique address known to it and the master. A single master can broadcast

MSP430, MSP432, LaunchPad are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

commands to all of the slaves and can direct commands to an individual slave using its unique address. The master can instruct a single slave to go into transmit mode, and then the master can put itself into receive mode, thereby allowing the master to retrieve data from the slave. After the slave transmission is complete, the slave puts itself back into receive mode so that the master can transmit additional commands. In this manner, data can be exchanged between the master and each slave on the bus.

Figure 1 shows a block diagram for a typical half-duplex RS-485 transceiver.

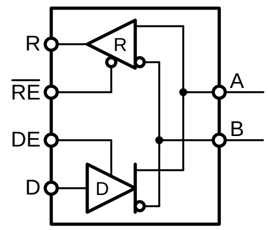

Figure 1. Block Diagram of a Half Duplex RS-485 Transceiver

Terminals A and B are connected to the differential RS-485 bus. Terminal R is connected to the receive pin of the UART module. Terminal D is connected to the transmit pin of the UART module. Terminal /RE is the receiver enable pin. The receiver is enabled when this pin is low. In Figure 1, the receiver is always enabled. Terminal DE is the control signal for enabling RS-485 transmitter. The transmitter is enabled when the DE pin is high.

To ensure proper RS-485 communication, the microcontroller must perform two additional functions in additional to the normal UART operation.

- Dynamically enable and disable the transmitter in the transceiver. The DE signal is asserted (high) only during data transmission. It must be low at all other times.
- A mechanism to detect the error if there is a data collision on the bus.

3 Use eUSCI or USCI for RS-485

On the MSP430 microcontrollers, the additional requirements for RS-485 must be implemented in software, because they are not natively supported by the eUSCI or USCI UART module.

To control to DE pin on the RS-485 transceiver, a GPIO pin can be used. After reset, this GPIO pin must be configured as output and set to low state. Because the GPIO pins default to inputs after reset, an external pulldown resistor is recommended so that the RS-485 transmitter is not falsely turned on in the start-up process. The software must set this pin high before writing data to the eUSCI or USCI transmit data register for transmission. The transmitter can be turned off only after the complete data frame has been shifted out on the bus. In the proposed solution, the receiver in the transceiver is always enabled. Each transmitted data is routed to the receiver of the same UART module. The completion of transmission on the bus can be indicated by the receive interrupt of the last data.

The received data is also used to check if there is a collision on the bus. A collision on the bus could cause two types of error.

- The received data does not match the transmitted data.
- UART errors such as framing error.

The RS-485 transmitter must be turned off as soon as the error is detected. The software can then respond to the error according the requirements of the application. In this approach, the error is detected after the full byte is transmitted. The RS-485 protocol does not define how the error should be detected and handled. The error handling is defined in the higher-level protocols. For example, the Modbus protocol requires parity checking per character and CRC checking per message. The above approach meets the Modbus requirement.

Figure 2 shows the process of transmitting a RS-485 message. For the efficient use of microcontroller resources, interrupts are used. The reception of a RS-485 message can be managed the same as the reception of a UART message.

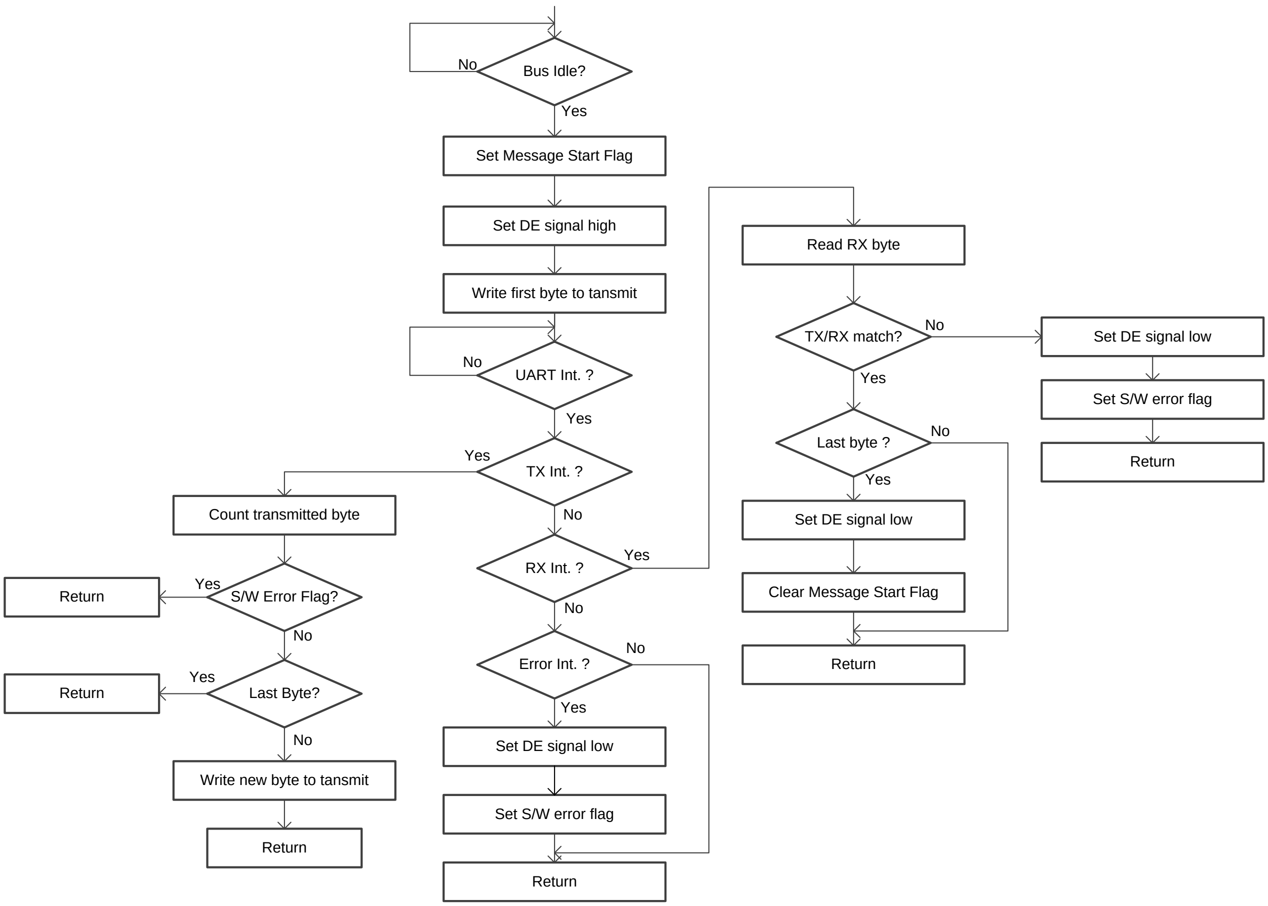

Figure 2. Process of Sending a RS-485 Message

For the USCI and eUSCI modules, the UCIDLE bit in the UCAxSTAT register can be used to check the bus idle condition. When the idle-line multiprocessor mode is selected, the UART state machine checks for an idle line after receiving a character. The UCIDLE flag is set after 10 ones are received.

Two software flags are used in Figure 2 as indicators of RS-485 message transmission status. The Message Start flag indicates that a RS-485 message is in the middle of transmission. The S/W Error flag notifies the system about the error during RS-485 transmission.

4 Application Examples

Figure 3 shows a test setup for verifying basic RS-485 operations. There are two RS-485 nodes constructed by an MSP430FR5994 LaunchPad™ development kit and an SN65HVD96 half-duplex RS-485 transceiver EVM. Two pieces of test firmware are available: RS-485 transmitter and RS-485 receiver. The RS-485 transmitter firmware transmits a 10-byte message repeatedly. There is a 10-ms time gap between messages. The data that is transmitted is routed back to the receiver to check if there is any corruption on the bus. The RS-485 receiver firmware reads the data from the RS-485 bus and saves them to a buffer in RAM. The default buffer size is 100 bytes. At the end of the test, the user can check the data in the receive buffer. The complete Code Composer Studio IDE projects for the firmware can be downloaded from http://www.ti.com/lit/zip/slaa722.

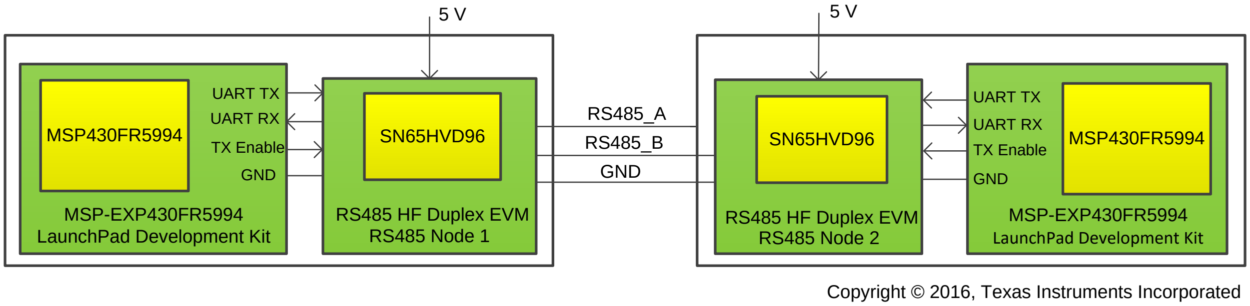

Figure 3. RS-485 Test Setup

Two use scenarios are tested. In the first scenario, Node 1 is programmed with the RS-485 transmitter firmware and Node 2 is programmed with the RS-485 receiver software. In the test, Node 2 should receive 100 data bytes from Node 1. Node 1 should not indicate any error.

In the second scenario, both nodes are programmed with the RS-485 transmitter firmware. Errors are expected on both nodes, because of the transmission data collision on the bus.

5 References

1. *RS-485 Design Guide*
2. *MSP430FR58xx, MSP430FR59xx, MSP430FR68xx, and MSP430FR69xx Family User's Guide*
3. *MSP430x5xx and MSP430x6xx Family User's Guide*
4. *SN65HVD96 RS-485 Transceiver*
5. *MSP430FR5994 LaunchPad™ Development Kit (MSP-EXP430FR5994) User's Guide*
6. *RS-485 Half-Duplex Evaluation Module User's Guide*
7. *RS-485 Reference Guide*

IMPORTANT NOTICE

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265