# UART-to-SPI Bridge Using Low-Memory MSP430™ MCUs

## Introduction

The universal asynchronous receiver transmitter (UART) interface and the serial peripheral interface (SPI) both enable serial communication between the MSP430™ microcontroller (MCU) and another device, such as a personal computer (PC) or another MCU or processor. UART is asynchronous, and SPI is synchronous, so many devices may only be able to communicate by one or the other. Some designs require communication between devices with these different serial protocols. This can be done using a bridge to convert packets from one protocol to the other.

The MSP430FR2000 MCU can be used as a low-cost UART-to-SPI bridge (configured as a SPI master) by using its enhanced Universal Serial Communication Interface (eUSCI) SPI module and its Timer module. To get started, download project files and a code example demonstrating this functionality.

## Implementation

Figure 1 shows the block diagram for the UART-to-SPI bridge. The MSP-TS430PW20 target development board was used for connecting the peripherals to the MSP430FR2000 MCU. Ensure that jumpers JP14 and JP15 are populated (leave JP13 unpopulated), jumper J16 is set to UART, and jumpers JP11, JP17, and JP18 are all removed. These jumper settings allow the backchannel UART interface on the MSP-FET programmer and debugger to simulate the UART device that will be communicating with the bridge. Using jumper wires, connect J4.14 (P2.0) to JP11.3, and connect J4.13 (P2.1) to JP11.4. To connect to a SPI device, connect the SPI device's clock pin to J4.17 (P1.5), the MOSI pin to J4.16 (P1.6), the MOSI pin to J4.15 (P1.7), and GND to J2.2. A simple SPI slave project was implemented on an MSP430FR2311 LaunchPad™ development kit to demonstrate the functionality of the UART-to-SPI bridge. The UART-to-SPI bridge functions as a SPI master in 3-wire mode. Low polarity is used, and the phase is 0 (TX data is shifted out on the rising clock edge).

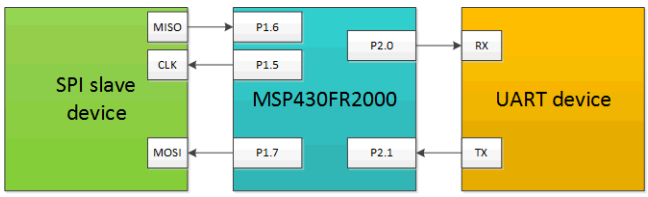


**Figure 1. UART-to-SPI Bridge Block Diagram**

Using a PC, open a new serial connection with a terminal program, and connect to the back-channel UART interface on the MSP-FET by selecting the COM port called *MSP Application UART1*. In the terminal window, change the baud rate to 9600 bps. To demonstrate the functionality of the UART-to-SPI bridge, enter a single byte into the terminal window and send it (see documentation on the terminal program you are using for specific instructions). It will be sent to the SPI slave device, and whatever value was in the TX buffer of the SPI slave device will be displayed in the serial terminal.

Communication must be initiated by the UART device because the UART-to-SPI bridge is configured as a SPI master. This could be changed by configuring the bridge as a SPI slave and connecting it to a SPI master, which could then initiate the communication.

For the firmware implementation, the main code initializes the digitally controlled oscillator (DCO), the hardware and software UART pins, the eUSCI SPI module, and the Timer module. Then, the central processing unit (CPU) goes to sleep by entering low-power mode 0 (LPM0). When the MCU receives UART or SPI interrupts, the CPU wakes up, enters active mode, and transmits the received data as quickly as possible before going back to sleep.

Figure 2 shows the flowchart for the SPI and UART code.

When a SPI packet is received on the MOSI pin (P1.6), the UCRXIFG interrupt flag is set, and the data bits are read from the SPI RX buffer. Next, the Timer module is started and delayed repeatedly to send the start bit, then each of the eight data bits, and then the stop bit over the software UART transmit pin (P2.0).

The software UART receive pin (P2.1) is initially configured as a general purpose input-output (GPIO) that provides an interrupt on the falling edge of an input signal.

When a UART packet is received by P2.1, the falling edge of the start bit triggers this interrupt flag. Next, the Timer module is started and delayed repeatedly to read each of the eight data bits (see Figure 3). The data bits are placed in the SPI TX buffer and sent over the SPI MOSI pin (P1.7).

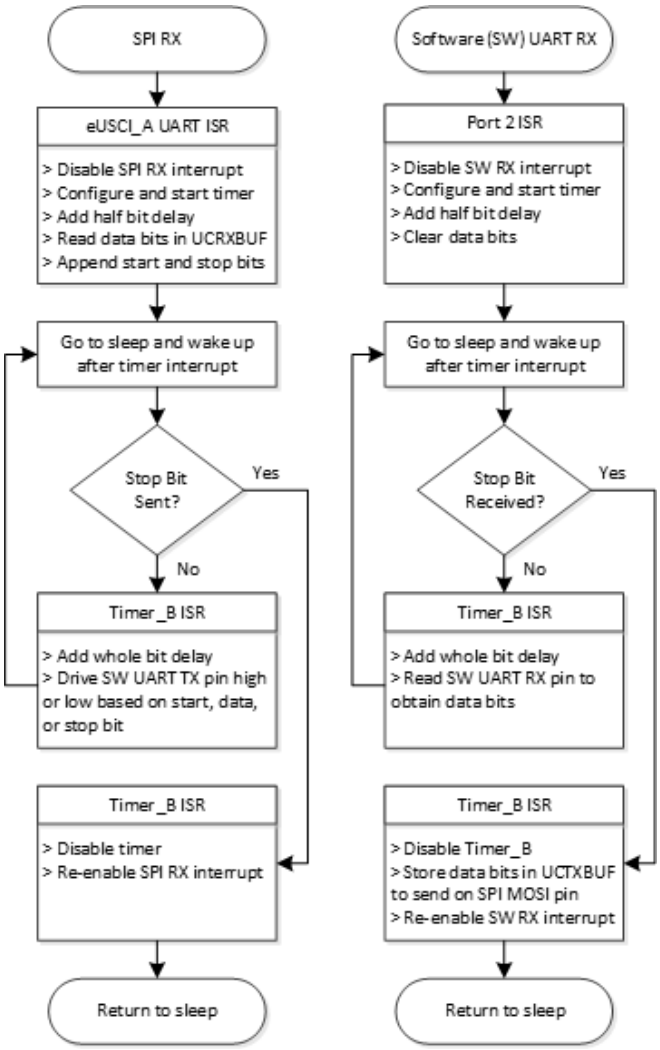


**Figure 2. UART and SPI RX and TX Code Flow**

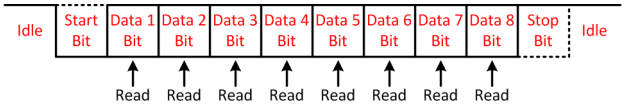


**Figure 3. UART Packet and Read Timings for Software UART RX**

### Performance

The firmware supports half-duplex UART communication only, which means one direction at a time. It also supports UART packets with eight data bits, least significant bit (LSB) first, no parity bit, and one stop bit. Because two serial interfaces are required, and the MSP430FR2000 MCU has one eUSCI module, the UART interface is implemented in software using the Timer module.

When two consecutive SPI packets are received by the SPI interface, the first packet is processed while the next packet is stored in UCRXBUF until it can be processed. If more than two consecutive packets are received before the software UART interface has finished sending the first packet, only the last packet received will be transmitted. This is not a concern with the current implementation as a second SPI packet will not be received by the bridge until it has received another UART packet (because it is a SPI master and determines when the SPI slave should send the next SPI packet), which will not happen because it is still sending out the previous UART packet. This limitation does not affect the software UART interface when consecutive UART packets are received.

To change the baud rate of the software UART interface, change the *WHOLE_BIT* definition in the code, which equals the subsystem master clock (SMCLK) divided by the baud rate. The *HALF_BIT* definition is just half this value. Note that SMCLK operates at 16 MHz. The SPI interface is operating with a bit clock equal to half of SMCLK. To change this, see the *MSP430FR4xx and MSP430FR2xx Family User's Guide* for the proper configuration and the *MSP430FR21xx, MSP430FR2000 Mixed-Signal Microcontrollers* data sheet for the maximum SPI clock value. After making these changes, close the serial terminal, rebuild the code, reprogram and reset the MCU, and then reopen the terminal with the new baud rate. Table 1 lists the maximum rates supported by both interfaces. Note that this is the maximum rate for the SPI master on the MSP430FR2000 MCU, and the maximum rate for the SPI device needs to be taken into account when setting the bit clock rate on the UART-to-SPI bridge.

**Table 1. Maximum Rates**

| Interface | Maximum Rate |
|---|---|
| SPI | 8-MHz bit clock |
| Software UART | 38400-bps baud rate |

To reduce power consumption while the MCU is not receiving or transmitting data, LPM0 is used. Other low-power modes may achieve lower power consumption, but they may require an external crystal oscillator and may limit the maximum baud rate due to increased wake-up times.

Due to limited code space on the MSP430FR2000 MCU, to implement more advanced features such as enabling the SPI interface to receive more than two consecutive SPI packets and adding UART support for odd or even parity bits, multiple stop bits, and CRC error detection, it may be necessary to upgrade to the 1KB MSP430FR2100 MCU.

Code space could be reduced by using hardware SPI as well as hardware UART, though, depending on the application and implementation, this could also increase the delay between packets being sent and received. This is due to the fact that the eUSCI module would have to be switched between being configure for SPI and being configured for UART.

### Device Recommendations

The device used in this example is part of the MSP430 Value Line Sensing portfolio of low-cost MCUs, designed for sensing and measurement applications. This example can be used with the devices shown in Table 2 with minimal code changes. For more information on the entire Value Line Sensing MCU portfolio, visit www.ti.com/MSP430ValueLine.

**Table 2. Device Recommendations**

| Part Number | Key Features |
|---|---|
| MSP430FR2000 | 0.5KB FRAM, 0.5KB RAM, eComp |
| MSP430FR2100 | 1KB FRAM, 0.5KB RAM, 10-bit ADC, eComp |
| MSP430FR2110 | 2KB FRAM, 1KB RAM, 10-bit ADC, eComp |
| MSP430FR2111 | 3.75KB FRAM, 1KB RAM, 10-bit ADC, eComp |

## IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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