**MSP430 UART**

The development on the MSP430 UART was done with Code Composer Studio version 5 and OLIMEX MSP430F169 prototype board. Appendix A contains the code for a single UART communication. The OLIMEX prototype board has only one available UART and to test UART communication a string sent to the MSP430 would be echoed back to the user. Communication tests used an XBEE connect to the MSP430 prototype board and another XBEE on a XBEE explorer from sparkfun.

Double UART MSP430 code was created. However, testing of this code could not be achieved. The code would enable UART0 and UART1 of the MSP430. If one UART received information then it would be sent to the other UART and vice versa. The idea was to connect one UART to an XBEE and another to USB. If data is received from the XBEE, it would be transmitted through the USB to the host computer. If data was sent through the USB, it would be transmitted through the XBEE to the Beagle Bone.

**Appendix A: Single UART code**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Khashi Xiong

// eyeCU

// MSP430 UART

// Setup the UART for the MSP430 and allow for data to be sent

// and received thru the UART ports.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#include** "msp430x16x.h"

**#include** <intrinsics.h>

**void** **uartconfig**(**void**);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Configure UART on MSP430

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **uartconfig**(**void**)

{

// UART Configuration

P3SEL |= 0xFF; // P3.6,7 = USART1 TXD/RXD

ME2 |= UTXE1 + URXE1; // Enable USART1 TXD/RXD

UCTL1 |= CHAR; // 8-bit character

UTCTL1 |= SSEL0; // UCLK = ACLK = 32.768kHz

UBR01 = 0x03; // 32.768kHz/9600 - 3.41

UBR11 = 0x00;

UMCTL1 = 0x4a; // Modulation

UCTL1 &= ~SWRST; // Initialize USART state machine

IE2 |= URXIE1; // Enable USART1 RX/TX interrupt

}

**void** **main**(**void**)

{

WDTCTL = WDTPW + WDTHOLD; // Stop Watchdog Timer

uartconfig(); // Configure UART Ports

\_BIS\_SR(GIE); // Check Interrupt if Interrupt occurs

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Interrupt to receive bits and echo back bits received

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#pragma** vector=USART1RX\_VECTOR

**\_\_interrupt** **void** **usart1\_rx** (**void**)

{

**while** (!(IFG2 & UTXIFG1)); // Check if UART Buffer is ready

TXBUF1 = RXBUF1; // TX buffer equals RX buffer

}

**Appendix B: Dual UART code**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Khashi Xiong

// eyeCU

// MSP430 UART

// Setup the UART for the MSP430 and allow for data to be sent

// and received thru the UART ports.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#include** "msp430x16x.h"

**#include** <intrinsics.h>

**void** **uartconfig**(**void**);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Configure UART on MSP430

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **uartconfig**(**void**)

{

// UART Configuration

P3SEL |= 0xF0; // P3.4,5,6,7 = USART0 and USART1 TXD/RXD

ME1 |= UTXE0 + URXE0; // Enable USART0 TXD/RXD

ME2 |= UTXE1 + URXE1; // Enable USART1 TXD/RXD

UCTL0 |= CHAR; // 8-bit character

UCTL1 |= CHAR; // 8-bit character

UTCTL0 |= SSEL0; // UCLK = ACLK = 32.768kHz

UTCTL1 |= SSEL0; // UCLK = ACLK = 32.768kHz

UBR00 = 0x03; // 32.768kHz/9600 - 3.41

UBR01 = 0x03; // 32.768kHz/9600 - 3.41

UBR10 = 0x00;

UBR11 = 0x00;

UMCTL0 = 0x4a; // Modulation

UMCTL1 = 0x4a; // Modulation

UCTL0 &= ~SWRST; // Initialize USART state machine

UCTL1 &= ~SWRST; // Initialize USART state machine

IE1 |= URXIE0 + UTXIE0; // Enable USART0 RX/TX interrupt

IE2 |= URXIE1 + UTXIE1; // Enable USART1 RX/TX interrupt

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Main function

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**void** **main**(**void**)

{

WDTCTL = WDTPW + WDTHOLD; // Stop Watchdog Timer

uartconfig(); // Configure Ports

\_BIS\_SR(GIE); // Check for interrupt

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Interrupt to receive bits and transmit to UART 1 RX Buffer

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#pragma** vector=USART0RX\_VECTOR

**\_\_interrupt** **void** **usart0\_rx** (**void**)

{

**while** (!(IFG1 & UTXIFG0)); // Check if buffer is ready

TXBUF0 = RXBUF1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Send UART 1 RX buffer to UART 1 TX buffer

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#pragma** vector=USART1RX\_VECTOR

**\_\_interrupt** **void** **usart1\_rx** (**void**)

{

**while** (!(IFG2 & UTXIFG1)); // Check if buffer is ready

TXBUF1 = RXBUF0;

}