Hardware UART on MSP430

Tutorial

This document contains a tutorial for using the hardware UART to communicate with the MSP430 via USB.

Reference is: Hardware UART MSP430

An example showing how to use the hardware UART on the MSP430G2553 to transmit and receive characters between a terminal on the computer and the MSP430 launchpad over the USB connection. In this example sending an ‘R’ or ‘G’ will turn the red and green LEDs on and sending an ‘r’ or ‘g’ will turn them off, respectively.

The source code follows:

```c
/*
 * Example code demonstrating the use of the hardware UART on the
 * MSP430G2553 to receive and transmit data back to a host computer over the USB
 * connection on the MSP430
 * */

/*
 * Note: After programming it is necessary to stop debugging and reset the uC before
 * connecting the terminal program to transmit and receive characters.
 * This demo will turn on the Red LED if an R is sent and turn it off if a r is sent.
 * Similarly G and g will turn on and off the green LED
 * It also transmits the received character back to the terminal.
 */

#include "msp430g2553.h"
void UARTSendArray(unsigned char *TxArray, unsigned char ArrayLength);

static volatile char data;

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    P1DIR |= BIT0 + BIT6; // Set the LEDs on P1.0, P1.6 as outputs
    P1OUT = BIT0; // Set P1.0

    BCSCTL1 = CALBC1_1MHZ; // Set DCO to 1MHz
    DCOCTL = CALDCO_1MHZ; // Set DCO to 1MHz

    /* Configure hardware UART */
    P1SEL = BIT1 + BIT2 ; // P1.1 = RXD, P1.2=TXD
    P1SEL2 = BIT1 + BIT2 ; // P1.1 = RXD, P1.2=TXD
    UCA0CTL1 |= UCSSEL_2; // Use SMCLK
    UCA0BR0 = 104; // Set baud rate to 9600 with 1MHz clock (Data Sheet 15.3.13)
    UCA0BR1 = 0; // Set baud rate to 9600 with 1MHz clock
    UCA0MCTL = UCBRS0; // Modulation UCBRSx = 1
    UCA0CTL1 &= ~UCSWRST; // Initialize USCI state machine
    IE2 |= UCA0RXIE; // Enable USCI_A0 RX interrupt

    __bis_SR_register(LPM0_bits + GIE); // Enter LPM0, interrupts enabled
}


// Echo back RXed character, confirm TX buffer is ready first
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void)
{
    data = UCA0RXBUF;
    UARTSendArray("Received command: ", 18);
    UARTSendArray(&data, 1);
    UARTSendArray("\n\r", 2);

    switch(data)
    {
        case 'R': P1OUT |= BIT0 ; break;
        case 'r': P1OUT &= ~BIT0; break;
        case 'G': P1OUT |= BIT6 ; break;
        case 'g': P1OUT &= ~BIT6; break;
        default : UARTSendArray("Unknown Command: ", 17);
                  UARTSendArray(&data, 1);
                  UARTSendArray("\n\r", 2);
                  break;
    }
}

void UARTSendArray(unsigned char *TxArray, unsigned char ArrayLength)
{
```

```
67        /*
68         * Send number of bytes Specified in ArrayLength in the array at
69         * using the hardware UART 0
70         *
71         * Example usage: UARTSendArray("Hello", 5);
72         * int data[2]={1023, 235};
73         *
74         * UARTSendArray(data, 4); // Note because the UART transmits bytes
75         * it is necessary to send two bytes for each integer hence the data
76         * length is twice the array length
77         * */
78
79        while(ArrayLength--); // Loop until StringLength == 0 and post decrement
80        while(!(IFG2 & UCA0TXIFG)); // Wait for TX buffer to be ready for new data
81        UCA0TXBUF = *TxArray; //Write the character at the location specified py the pointer
82        TxArray++; //Increment the TxString pointer to point to the next character
83     }
84    }
85
```