

USART Module in UART Mode with the EasyWeb2 Hardware Platform

by Alex Milenkovich, milenkovic@computer.org

Objective: This tutorial discusses the use of MSP430's USART Device.

Note: Required are all previous tutorials.

1. Echo a Character

Let us consider a C application shown in Figure 1. A system with 32KHz crystal on ACLK is connected to a personal computer running HyperTerminal application using a serial RS232 protocol. To realize this communication we use the MSP430's USART1 device in UART mode.

First, the USART1 is configured in UART mode with the following parameters. The communication speed is 2400 bits/sec (one bit period is 416 us). The USART clock UCLK is connected to ACLK (32 KHz). To achieve the baud rate of the 2400, the internal divider registers are initialized to UBR01=0x0D, and UBR11=0x00. Why? The modulator register UMCTL1=0x6B. See the reference manual for the details how the value in UMCTL1 is determined. You may also use the following utility to determine the values in the USART registers for a given configuration of serial communication (see <http://msp gcc.sourceforge.net/baudrate.html>). The format of a character is 8-bit ASCII.

The main program is an infinite loop. The MSP430 is in the LPM3 low-power mode. What clock signals are down in this mode? When a new character is received the USART1Rx ISR is executed. In the ISR the MSP430 status register on the stack is modified to enable that the processor is active after executing the ISR. In the main program the received character is sent back to the HyperTerminal (echo), and the processor goes back in the LPM3.

```

//*****
// MSP-FET430P140 Demo - USART1 UART 2400 Ultra-low Power Echo ISR, 32kHz ACLK
//
// Description; Echo a received character, RX ISR used. In the Mainloop UART1
// is made ready to receive one character with interrupt active. The Mainloop
// waits in LPM3. The UART1 ISR forces the Mainloop to exit LPM3 after
// receiving one character which echo's back the received character.
// ACLK = UCLK1 = LFXT1 = 32768, MCLK = SMCLK = DCO~ 800k
// Baud rate divider with 32768hz XTAL @2400 = 32768Hz/2400 = 13.65 (000Dh)
// /*An external watch crystal is required on XIN XOUT for ACLK*/
//
//
//           MSP430F149
//           -----
//           ||\|                 XIN|-
//           |  |                 | 32kHz
//           --RST                 XOUT|-
//           |                     |
//           |                     P3.6|----->
//           |                     | 2400 - 8N1
//           |                     P3.7|<-----
//
//
// M. Buccini
// Texas Instruments, Inc
// October 2003
// Built with IAR Embedded Workbench Version: 1.26B
// January 2004
// Updated for IAR Embedded Workbench Version: 2.21B
//*****
#include <msp430x14x.h>

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;           // Stop WDT
    P3SEL |= 0xC0; // P3.6,7 = USART1 option select
    ME2 |= UTXE1 + URXE1; // Enable USART1 TXD/RXD
    UCTL1 |= CHAR; // 8-bit character
    UTCTL1 |= SSEL0; // UCLK = ACLK
    UBR01 = 0x0D; // 32k/2400 - 13.65
    UBR11 = 0x00;
    UMCTL1 = 0x6B; // Modulation
    UCTL1 &= ~SWRST; // Initialize USART state machine
    IE2 |= URXIE1; // Enable USART1 RX interrupt

// Mainloop
for (;;)
{
    _BIS_SR(LPM3_bits + GIE); // Enter LPM3 w/interrupt
while (!(IFG2 & UTXIFG1)); // USART1 TX buffer ready?
TXBUF1 = RXBUF1; // RXBUF1 to TXBUF1
}
}

// USART1 RX ISR will for exit from LPM3 in Mainloop
interrupt[USART1RX_VECTOR] void usart1_rx (void)
{
    _BIC_SR_IRQ(LPM3_bits); // Clear LPM3 bits from 0(SR)
}

```

Figure 1. C program for echoing a character sent from HyperTerminal.

2. Assignments

Assignment #1: Serial communication (PC to EasyWeb2 and EasyWeb2 to PC)

Write a C application that accepts a character from the HyperTerminal and then echo it twice to the HyperTerminal. The main program is an infinite loop waiting for a new character (received through an interrupt service routine); after a character is received it sends its value back to the HyperTerminal twice, followed by a *new line* and a *carriage return* characters.

Note UART mode: Baud rate is 38400 bps, 8-bit characters

To set up the HyperTerminal do the following:

- Connecting using: COM1
- Bit per second: 38400
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: use Xon/Xoff

Assignment #2: Serial communication (EasyWeb2 to PC)

Write a C application that will maintain a real-time clock in the following format:

<<hh:mm:ss>>. The current value of the time is sent serially to the PC via asynchronous serial link. Utilize low-power modes of the MSP430 to conserve energy.