ECED 3204 Microprocessor

Assignment #6 Reference Solution

http://www.jasongu.org/3204/assignments.html

Assignment #6 contains the following problems:

E13.3 Write a subroutine and a C function to configure the USART3 of the MEGA640/1280/2560 MCU to operate in asynchronous mode at 38400 baud rate using the data format of one start bit, 8 data bits, and one stop bit assuming that f_{OSC} = 16 MHz.

Solution:

```
void USART3(void)
{
    DDRJ      = 0x06;     // configure RxD3/PJ0, TxD3/PJ1, XCK3/PJ2 for input, output, and output
    UBRR0L    = 25;       // set baud rate to 38400
    UBRR0H    = 0;        //      "
    UCSR0A    = 0;
    UCSR0B    = 0x18;
    UCSR0C    = 0x06;
}
```

E13.7 Write a subroutine (call it **putHex2**) and a C function that can output the value in register r16 as two hexadecimal digits to the USART0 modules.

Solution:
Version for USART0:
Assembly subroutine:

```
; ---------------------------------------------------------------------------------------------------------
; The next subroutine outputs the binary value in r16 as two hex digits from USART0
; ---------------------------------------------------------------------------------------------------------
putHex2:    push    r16                 ; save a copy
            andi    r16,0xF0            ; mask out the lower 4 bits
            swap    r16                 ; swap to lower 4 bit positions
            cpi     r16,0x0A            ; is the number larger than 9?
            brlt    addH30              ; Add 0x30 to convert to ASCII if less than 10
            ldi     r20,0x37            ; Add 0x37 if between 10 and 15
            add     r16,r20             ;     "
            rjmp    putIt               ;     "
addH30:     ldi     r20,0x30
            add     r16,r20
putIt:      call    putchUSART0
            pop     r16
            andi    r16,0x0F            ; extract the low four bits
            cpi     r16,0x0A            ; is it less than 10?
            brlt    adH30
            ldi     r20,0x37            ; lower nibble is between 9 and 16
```

```
                add     r16,r20             ; so add 0x37 to convert to ASCII code
                rjmp    put2
adH30:          ldi     r20,0x30            ; lower nibble is less than 10, so add
                add     r16,r20             ; 0x30 to convert to ASCII code
put2:           call    putchUSART0         ;     “
                ret
```

C function is as follows:

```
void putHex2(unsigned char cx)
{
    unsigned char tmp;
    tmp = cx & 0xF0;
    tmp >>= 4;
    if(tmp < 10)
        putchUSART0(tmp + 0x30);
    else
        putchUSART0(tmp + 0x37);
    tmp = cx & 0x0F;
    if(tmp < 10)
        putchUSART0(tmp + 0x30);
    else
        putchUSART0(tmp + 0x37);
}
```