```
42  //  sketch_msp430fr243x_euscia0_uart_01.ino  Modified for Energia HW 20180812
43  //*****************************************************************************
44
45  #include <msp430.h>
46
47  void Init_GPIO();
48
49  int main(void)
50  {
51    WDTCTL = WDTPW | WDTHOLD;                 // Stop watchdog timer
52
53    // Configure GPIO
54    Init_GPIO();
55    PM5CTL0 &= ~LOCKLPM5;                     // Disable the GPIO power-on default high-impedance mode
56                                              // to activate 1previously configured port settings
57
58    __bis_SR_register(SCG0);                  // disable FLL
59    CSCTL3 |= SELREF__REFOCLK;                // Set REFO as FLL reference source
60    CSCTL0 = 0;                               // clear DCO and MOD registers
61    CSCTL1 &= ~(DCORSEL_7);                   // Clear DCO frequency select bits first
62    CSCTL1 |= DCORSEL_3;                      // Set DCO = 8MHz
63    CSCTL2 = FLLD_0 + 243;                    // DCODIV = 8MHz
64    __delay_cycles(3);
65    __bic_SR_register(SCG0);                  // enable FLL
66    while(CSCTL7 & (FLLUNLOCK0 | FLLUNLOCK1)); // Poll until FLL is locked
67    CSCTL4 = SELMS__DCOCLKDIV | SELA__REFOCLK; // set default REFO(~32768Hz) as ACLK source, ACLK = 32768Hz
68                                              // default DCODIV as MCLK and SMCLK source
69
```

```
73      // Configure UART
74      UCA0CTLW0 |= UCSWRST;
75      UCA0CTLW0 |= UCSSEL__SMCLK;
76
77      // Baud Rate calculation
78      // 8000000/(16*9600) = 52.083
79      // Fractional portion = 0.083
80      // User's Guide Table 14-4: UCBRSx = 0x49
81      // UCBRFx = int ( (52.083-52)*16) = 1
82      UCA0BR0 = 52;                             // 8000000/16/9600
83      UCA0BR1 = 0x00;
84      UCA0MCTLW = 0x4900 | UCOS16 | UCBRF_1;
85
86      UCA0CTLW0 &= ~UCSWRST;                    // Initialize eUSCI
87      UCA0IE |= UCRXIE;                         // Enable USCI_A0 RX interrupt
88
89      __bis_SR_register(LPM3_bits|GIE);         // Enter LPM3, interrupts enabled
90      __no_operation();                         // For debugger
91  }
92
```

```
 93  #pragma vector=USCI_A0_VECTOR
 94  __interrupt void USCI_A0_ISR(void)
 95  {
 96    switch(UCA0IV)
 97    {
 98      case USCI_NONE: break;
 99      case USCI_UART_UCRXIFG:
100        while(!(UCA0IFG&UCTXIFG));
101        UCA0TXBUF = UCA0RXBUF;
102        __no_operation();
103        break;
104      case USCI_UART_UCTXIFG: break;
105      case USCI_UART_UCSTTIFG: break;
106      case USCI_UART_UCTXCPTIFG: break;
107      default: break;
108    }
109  }
110
111  void Init_GPIO()
112  {
113      P1DIR = 0xFF; P2DIR = 0xFF; P3DIR = 0xFF;
114      P1REN = 0xFF; P2REN = 0xFF; P3REN = 0xFF;
115      P1OUT = 0x00; P2OUT = 0x00; P3OUT = 0x00;
116  }
```

Screen shot of example program run

```
26  //*******This is the SOLUTION for default power up clock rate
27  //******************************************************************************
28  #include <msp430.h>
29
30  unsigned char RXData = 0;
31  unsigned char TXData = 1;
32
33  int main(void)
34  {
35      WDTCTL = WDTPW | WDTHOLD;                 // Stop watchdog timer
36
37      PM5CTL0 &= ~LOCKLPM5;                     // Disable the GPIO power-on default high-impedance mode
38                                                // to activate previously configured port settings
39      P1DIR |= BIT0;
40      P1OUT &= ~BIT0;                           // P1.0 out low
41
42      // Configure UART pins
43      P1SEL0 |= BIT4 | BIT5;                    // set 2-UART pin as second function
44
```

```
44
45      // Configure UART for Power UP default clock rate
46      UCA0CTLW0 |= UCSWRST;                       // Put eUSCI in reset
47      UCA0CTLW0 |= UCSSEL__SMCLK;                 // set equal to SMCLK
48
49      // Baud Rate calculation 115200 baud
50       UCA0BR0 = 8;                              // 1000000/115200 = 8.68
51       UCA0MCTLW = 0xD600;                       // 1000000/115200 - INT(1000000/115200)=0.68
52                                                 // UCBRSx value = 0xD6 (See SLAU445G 23.3.10)
53      // 9600 baud
54  //     UCA0BR0 = 104;                             // 1000000/9600 = 104.167  104
55  //     UCA0MCTLW = 0x1100;                        //   0x11 = 0.167
56
57      UCA0CTLW0 &= ~UCSWRST;                      // Initialize eUSCI
58      UCA0IE |= UCRXIE;                           // Enable USCI_A0 RX interrupt
59
60
61          __bis_SR_register(LPM0_bits|GIE);       // Enter LPM0 CPU off, SMCLK running
62
63  }
```

```
64
65  #pragma vector=USCI_A0_VECTOR
66  __interrupt void USCI_A0_ISR(void)
67  {
68      switch(UCA0IV)
69      {
70          case USCI_NONE: break;
71          case USCI_UART_UCRXIFG:
72              while(!(UCTXIFG&UCA0IFG));
73              UCA0TXBUF = UCA0RXBUF;                  // Load data onto buffer
74               break;
75          case USCI_UART_UCTXIFG: break;
76          case USCI_UART_UCSTTIFG: break;
77          case USCI_UART_UCTXCPTIFG: break;
78      }
79  }
```

```
24 //   Ling Zhu    Texas Instruments Inc.   July 2015
25 //   sketch_msp430fr243_uart_03.ino   H Watson Energia Version 20180716
26 //*******This is the SOLUTION for default power up clock rate
27 //******************************************************************************
28 #include <msp430.h>
29
30 unsigned char RXData = 0;
31 unsigned char TXData = 1;
32
33 int main(void)
34 {
35     WDTCTL = WDTPW | WDTHOLD;                 // Stop watchdog timer
36
37     PM5CTL0 &= ~LOCKLPM5;                     // Disable the GPIO power-on default high-impedance mode
38                                               // to activate previously configured port settings
39     P1DIR |= BIT0;
40     P1OUT &= ~BIT0;                           // P1.0 out low
41
42     // Configure UART pins
43     P1SEL0 |= BIT4 | BIT5;                    // set 2-UART pin as second function
44
45     // Configure UART for Power UP default clock rate
46     UCA0CTLW0 |= UCSWRST;                     // Put eUSCI in reset
```

Done uploading.

```
          Finished: 55%
          Setting PC to entry point.: 55%
info: MSP430:  Flash/FRAM usage is 294 bytes. RAM usage is 0 bytes.
Running...
Success
```


/dev/ttyACM1
The quick brown fox jumps over the lazy dog
Send
Now is the time for all good men
to come to the aid of their country
Autoscroll
Both NL & CR
115200 baud

```
 *   sketch_BackUART2433.ino  20180814  H. Watson  copyright
 */

#include <msp430.h>
#include <stdio.h>  // use for sprintf(), no putchar() required;

void UARTSendArray( char TxArray[]);  // our simple printf()
unsigned char data;  // received char data

char MyString [25];  // work buffer for sprintf()

int main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT

// clock system setup
     __bis_SR_register(SCG0);                          // disable FLL
    CSCTL3 |= SELREF__REFOCLK;                         // Set REFOCLK as FLL reference source
    CSCTL0 = 0;                                        // clear DCO and MOD registers
    CSCTL1 &= ~(DCORSEL_7);                            // Clear DCO frequency select bits first
    CSCTL1 |= DCORSEL_3;                               // Set DCOCLK = 8MHz
    CSCTL2 = FLLD_1 + 121;                             // FLLD = 1, DCODIV = 4MHz
    __delay_cycles(3);
    __bic_SR_register(SCG0);                           // enable FLL
    while(CSCTL7 & (FLLUNLOCK0 | FLLUNLOCK1));         // Poll until FLL is locked
    CSCTL4 = SELMS__DCOCLKDIV | SELA__XT1CLK;          // set ACLK = XT1 = 32768Hz, DCOCLK as MCLK and SMCLK source
    CSCTL5 |= DIVM1;                                  // SMCLK = MCLK = DCODIV/2 = 1MHz, by default

     PM5CTL0 &= ~LOCKLPM5;            // Disable the GPIO power-on default high-impedance mode
                                      // to activate 1previously configured port settings
```

```
48
49      P1DIR |= BIT0 | BIT1 ; // RED = 0, GREEN = 1
50      P1OUT &= ~(BIT0 | BIT1); // Clear P1.0
51
52
53      /* Configure hardware UART */
54    // Configure UART pins
55     P1SEL0 |= BIT4 | BIT5;    // set 2-UART pin as second function  P1.4 - TX  P1.5 RX
56
57    // Configure UART
58      UCA0CTLW0 |= UCSWRST;
59      UCA0CTLW0 |= UCSSEL__SMCLK;
60
61      UCA0BR0 = 107;       // 1MHz SMCLK/9600 BAUD
62  //    UCA0BR1 = 0x00;
63      UCA0MCTLW = 0x1100; // | UCOS16 | UCBRF_1;
64      UCA0CTLW0 &= ~UCSWRST;     UCA0IE |= UCRXIE;                          // Enable USCI_A0 RX interrupt
65
66   // wait for input
67        UARTSendArray("Waiting on input: \n");  // Polled ouptut, so can send w/o GIE
68      __bis_SR_register(LPM0_bits + GIE); // Enter LPM0, interrupts enabled, wait for command
69  }
```

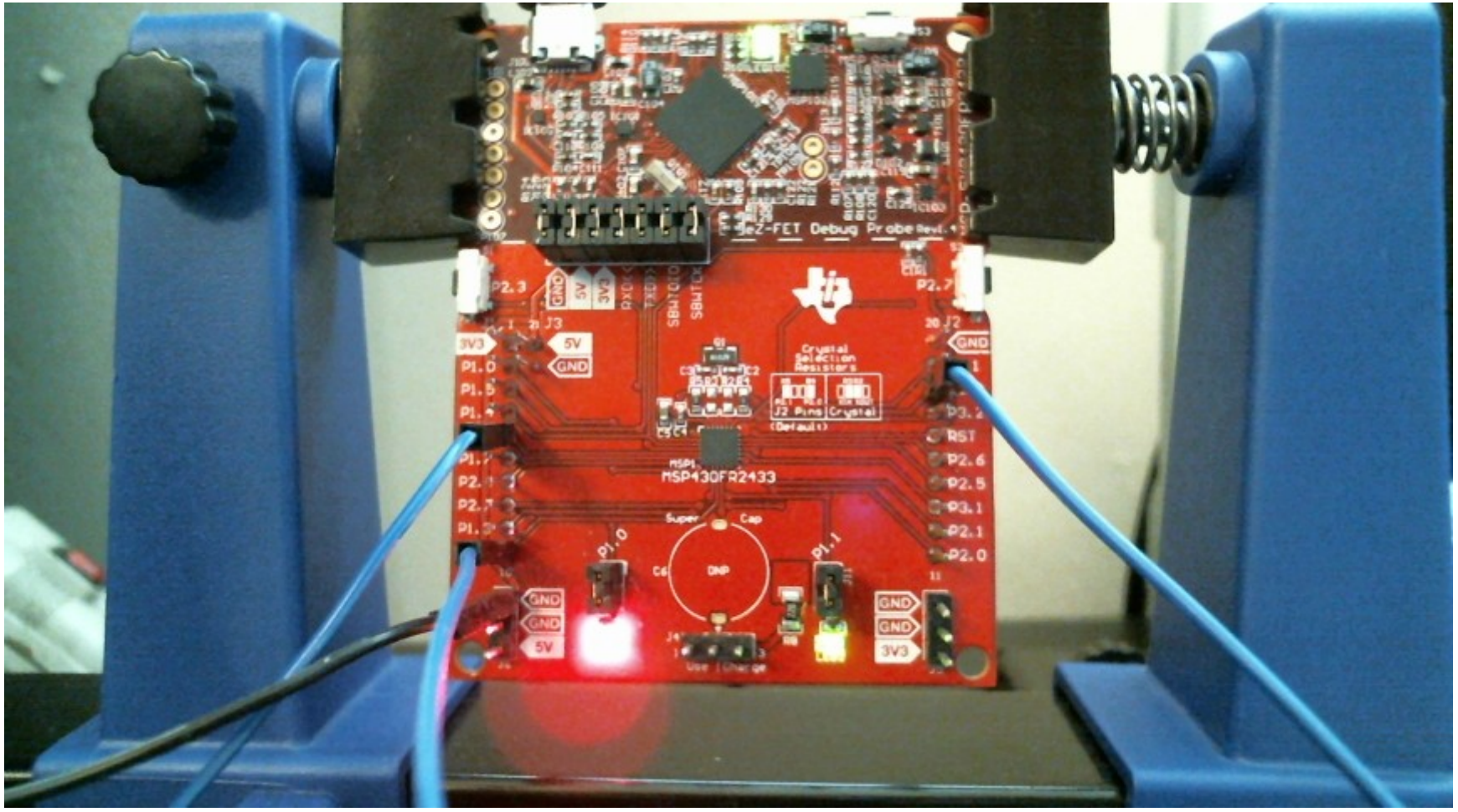
```
71 // RXed character ISR - others unused
72 #pragma vector=USCI_A0_VECTOR
73 __interrupt void USCI_A0_ISR(void)
74 {
75   // decode the UART Interrupt Vector see 21.4.12 UCAxIV Register
76     switch(UCA0IV)
77     {
78         case USCI_NONE: break;
79         case USCI_UART_UCRXIFG:
80               data = UCA0RXBUF;           // read the received char - also clears Interrupt
81 //              // Echo received char
82 //              while(!(UCA0IFG & UCTXIFG)); // Wait for TX buffer to be ready for new data
83 //                UCA0TXBUF = data; //Echo Write the character at the location specified py the pointer
84               RxInput();   // process the received char
85             break;
86         case USCI_UART_UCTXIFG:  // Tx Interrupt - unused
87             break;
88         case USCI_UART_UCSTTIFG: break; // unused
89         case USCI_UART_UCTXCPTIFG: break; // unused
90   }
91 }
92
```

```
void RxInput()
{
    // use sprintf to prepare string for output
      sprintf(MyString, "Received command: %c\n", data);
      UARTSendArray(MyString);
    switch(data)
    {
    case 'R':
    {
        P1OUT |= BIT0;
    }
    break;
    case 'r':
    {
        P1OUT &= ~BIT0;
    }
    break;
    case 'G':
    {
        P1OUT |= BIT1;
    }
    break;
    case 'g':
    {
        P1OUT &= ~BIT1;
    }
    break;
```

```
    break;
    case 'L':
    {
      int i;
      for (i=0; i<10; i++)
      {
          sprintf(MyString, "MyString Value: %d\n",i);
          UARTSendArray(MyString);
      }
      break;
    }
    default:
    {
        UARTSendArray("Unknown Command: \n");
    }
    break;
    }
}

void UARTSendArray(char *TxArray)
{
  // send a 'C' language string to the transmit port
  //  string has to be terminated with a binary '0' (according to C conventions)
  //  blocking is implicit in this function (will wait to finish string before returning)
  //  Impossible to run Tx Interrupt without circular queue - Then use putchar() w/ queue
    while(*TxArray)  // loop until binary zero (EOS)
    {
       while(!(UCA0IFG & UCTXIFG)); // Wait for TX buffer to be ready for new data
        UCA0TXBUF = *TxArray++; //Write the character at the location specified py the pointer
    }
}
```


/dev/ttyACM1
Send
Received command: r
Received command: R
Received command: G
Received command: g
Received command: L
MyString Value: 0
MyString Value: 1
MyString Value: 2
MyString Value: 3
MyString Value: 4
MyString Value: 5
MyString Value: 6
MyString Value: 7
MyString Value: 8
MyString Value: 9
Autoscroll
No line ending
9600 baud


eZ-FET Debug Probe
MSP430FR2433
Super Cap
DNP
Crystal Selection Resistors
J2 Pins
Crystal
(Default)
P1.0
P1.1
GND
5V
3V3
RST