

```
#include <msp430.h>
```

```
volatile int connected = 0;
```

```
volatile int led_pwm = 0;
```

```
void timerred(int);
```

```
void timerblue(int);
```

```
void timergreen(int);
```

```
char rxData[8];
```

```
int main(void)
```

```
{  
    WDTCTL = WDTPW + WDTHOLD;           // Stop watchdog timer  
    if (CALBC1_1MHZ == 0xFF)           // If calibration constant erased  
    {  
        while (1)  
            ;                          // do not load, trap CPU!!  
    }  
    DCOCTL = 0;                         // Select lowest DCOx and MODx settings  
    BCSCCTL1 = CALBC1_1MHZ;             // Set DCO  
    DCOCTL = CALDCO_1MHZ;  
    UCA0CTL1 |= UCSSEL_2;  
    P1SEL |= BIT1 + BIT2;               // P1.1 = RXD, P1.2=TXD  
    P1SEL2 |= BIT1;  
    UCA0BR0 = 104;                       // 8MHz 9600  
    UCA0BR1 = 0;  
  
    UCA0MCTL = UCBRS1 + UCBRS0;         // Modulation UCBRSx = 3  
    UCA0CTL1 &= ~UCSWRST;               // **Initialize USCI state machine**  
    IE2 |= UCA0RXIE;                   // Enable USCI_A0 TX/RX interrupt  
    __bis_SR_register(GIE);            // Enter LPM3 w/ interrupts enabled  
  
    while (1)  
        ;  
}  
void timerred(int value)                //Function for pwm of RED Led at pin 2.2  
{  
    P2DIR |= BIT2;  
    P2SEL |= BIT2;  
    P2SEL2 &= ~BIT2;  
    TA1CCR0 = 255;  
    TA1CCTL2 = OUTMOD_7;  
    TA1CCR2 = value;  
    TA1CTL = TASSEL_2 + MC_1;  
}  
  
void timergreen(int value)              //Function for pwm of Green Led at pin 2.4  
{  
    P2DIR |= BIT4;  
    P2SEL |= BIT4;  
    P2SEL2 &= ~BIT4;  
    TA1CCR0 = 255;  
    TA1CCTL1 = OUTMOD_7;
```

```

    TA1CCR1 = value;
    TA1CTL = TASSEL_2 + MC_1;
}

void timerblue(int value)      //Function for pwm of BLUE Led at pin 1.6
{
    P1DIR |= BIT6;
    P1SEL |= BIT6;
    TA0CCR0 = 255;
    TA0CCTL1 = OUTMOD_7;
    TA0CCR1 = value;
    TA0CTL = TASSEL_2 + MC_1;
}
// USCI A0/B0 Receive ISR
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void)
{
    static unsigned int count = 0;

    rxData[count++] = UCA0RXBUF;          // Display RX'ed charater
    if (count == 8)
    {
        if (rxData[7] == '1')
        {
            if (rxData[6] == '1')
            {
                led_pwm = (100) * (rxData[1] - 48) + (10) * (rxData[2] - 48)
                    + (rxData[3] - 48);
                if (rxData[5] == '1' && rxData[4] == '0')    //BLUE BAR
                {
                    //timerred(led_pwm);
                    timerblue(led_pwm);
                }
                else if (rxData[5] == '0' && rxData[4] == '1')    //RED BAR
                {
                    timerred(led_pwm);
                }
                else if (rxData[5] == '1' && rxData[4] == '1')    //GREEN BAR
                {
                    timergreen(led_pwm);
                }
            }
        }
        else
        {
            count = 0;
        }
    }
}

```

