

```

#include "msp430g2231.h"
#define TXD BIT1 // TXD on P1.1
// Conditions for 9600/4=2400 Baud SW UART, SMCLK = 1MHz
#define Bitime_5 20 // ~ 0.5 bit length + small adjustment
#define Bitime 52
unsigned char BitCnt; //Bit Counter
unsigned int TXByte; //Data to send
void ConfigureTimerUart(void);
void Transmit(void);
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop WDT
    BCSCTL1 = CALBC1_1MHZ; // Set clock range
    DCOCTL = CALDCO_1MHZ;
    P1SEL |= TXD; // Select I/O pins for UART
    P1DIR |= TXD;
    __enable_interrupt();
    while(1)
    {
        ConfigureTimerUart();
        TXByte = 0x4B; //K
        Transmit();
    }
}

void ConfigureTimerUart(void)
{
    CCTLO = OUT; // TXD Idle as Mark
    TACTL = TASSEL_2 + MC_2 + ID_3; // SMCLK/8, continuous mode
}

// Function Transmits Character from TXByte
void Transmit()
{
    BitCnt = 0xA; // Load Bit counter, 8data + ST/SP
    while (CCR0 != TAR) // Prevent async capture
        CCR0 = TAR; // Current state of TA counter
    CCR0 += Bitime; // Some time till first bit
    TXByte |= 0x100; // Add mark stop bit to TXByte
    TXByte = TXByte << 1; // Add space start bit
    CCTLO = CCIS0 + OUTMOD0 + CCIE; // TXD = mark = idle
    while ( CCTLO & CCIE ); // Wait for TX completion
}

// Timer A0 interrupt service routine
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A (void)
{
    CCR0 += Bitime; // Add Offset to CCR0
    if (CCTLO & CCIS0) // TX on CCI0B?
    {
        if ( BitCnt == 0)
            CCTLO &= ~ CCIE; // All bits TXed, disable interrupt
        else
        {
            CCTLO |= OUTMOD2; // TX Space
            if (TXByte & 0x01)
                CCTLO &= ~ OUTMOD2; // TX Mark
            TXByte = TXByte >> 1;
            BitCnt --;
        }
    }
}

```

