lower jtag1 port on experimenter board for 2013

same uart setup

change target board in ccs

#include "msp430x20x3.h"

#define LED\_PIN BIT0

void SPI\_Setup() {

USICTL0 |= USISWRST; // Set UCSWRST -- needed for re-configuration process

USICTL0 |= USIPE5 + USIPE6 + USIPE7 + USIOE; // SCLK-SDO-SDI port enable,MSB first

USICTL1 = USIIE; // USI Counter Interrupt enable

USICTL0 &= ~USISWRST; // \*\*Initialize USCI state machine\*\*

}

void SPI\_Send(unsigned char data) {

USISRL = data; // Load data into the USI shift register

USICNT = 8; // Transmit 8 bits

}

unsigned char SPI\_Receive() {

return USISRL; // Return received data from the USI shift register

}

void LED\_Blink(int count, int delay\_cycles) {

volatile int i;

for (i = 0; i < count; i++) {

P1OUT |= LED\_PIN; // Turn on LED

\_\_delay\_cycles(delay\_cycles);

P1OUT &= ~LED\_PIN; // Turn off LED

\_\_delay\_cycles(delay\_cycles);

}

}

int main(void) {

WDTCTL = WDTPW + WDTHOLD; // Stop WDT

P1DIR |= LED\_PIN; // Set P1.0 as output

SPI\_Setup();

unsigned char rxData = 0;

int multiplier = 4; // Initial multiplier value

int delay\_cycles = 250000; // Initial delay cycles for 1 Hz frequency

while (1) {

if (rxData != 0) {

if (rxData >= 1 && rxData <= 20) {

multiplier = rxData;

delay\_cycles = 250000 / (4 \* multiplier); // Calculate delay cycles for the desired frequency

LED\_Blink(multiplier, delay\_cycles);

} else if (rxData == 0) {

P1OUT |= LED\_PIN; // Turn on LED continuously

} else if (rxData == 255) {

SPI\_Send(multiplier); // Send current multiplier value as response

}

rxData = 0; // Reset received data

}

}

return 0;

}

#pragma vector = USI\_VECTOR

\_\_interrupt void USI\_ISR(void) {

rxData = SPI\_Receive(); // Receive the command

USICNT = 0; // Clear USI counter

\_\_bic\_SR\_register\_on\_exit(LPM0\_bits); // Exit LPM0

}