EE 215 Microprocessors LAB #8

Student name：Li Xianzhe
Student ID number：2022214880

Display

Has shown the professor the performance of my board.

Through program control, one microcontroller sends data (1, 2, 3, 4) through TXD after serial port initialization, and the other MCU receives data through RXD after serial port initialization, and makes different LED flashes according to the different data received. The flashing of the LEDs corresponds to the verification of whether the data sent and received is correct.

Code

```
// Lab 8
// UART mode
// TX: 1,2,3,4 are transmitted when S1, S2, S3, and S4 are
pressed,respectively
// RX: if 1 is received, turn LED 1 on for 0.2 second
// if 2 is received, turn LED 2 on for 0.2 second
// if 3 is received, turn LED 3 on for 0.2 second
// if 4 is received, turn LED 4 on for 0.2 second
#include <msp430.h>
unsigned char recv_data;
int main(void) {
 WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
 P8DIR |= BIT1; //LED1
 P8OUT &= ~BIT1;
 P3DIR |= BIT7; //LED2
 P3OUT &= ~BIT7;
 P7DIR |= BIT4; //LED3
 P7OUT &=~ BIT4;
 P6DIR |= BIT3; //LED4
 P6OUT &=~ BIT3;
 //******** PxDIR=0, PxREN=1, PxOUT=1, input with pull up
resister***********
 P1REN |=BIT2+BIT3; // P1.2 and P1.3 (S1 and S2)Pull up/Pull down
resister enabled
 P1OUT |=BIT2+BIT3; // P1.2 and P1.3 (S1 and S2) Pullup esister
enabled
 //********** Interrupt
set*********************************************
 P1IE |=BIT2+BIT3; // P1.2 and P1.3 (S1 and S2) interrupt enabled
```

```
 P1IES|=BIT2+BIT3; // P1IFG flag is set with a high-to-lowtransition
 P1IFG = 0; // clear interrupt flag
 //******** PxDIR=0, PxREN=1, PxOUT=1, input with pull up
resister***********
 P2REN|=BIT3+BIT6; // P2.6 (S4) Pull up/Pull down resister enabled
 P2OUT|=BIT3+BIT6; // P2.6 (S4) Pull up resister enable
 //********** Interrupt
set*********************************************
 P2IE|=BIT3+BIT6; // SWITCH 4, Port 2.6 interrupt enabled 外部按键输
入的中断使能
 P2IES|=BIT3+BIT6; // P1IFG flag is set with a high-to-lowtransition
下降沿触发
 P2IFG = 0; // clear interrupt flag
 P4SEL |= BIT5+BIT4; // P4.4,5使用外设功能 = UCA1 TXD/RXD
 UCA1CTL1 |= UCSWRST; // 复位USCI
 UCA1CTL1 |= UCSSEL_2; // 设置SMCLK时钟，用于发生特定波特率
 UCA1BR0 = 104; // 设置波特率， 1MHz 波特率=9600
 UCA1BR1 = 0;
 UCA1MCTL |= UCBRS_1 + UCBRF_0;
 UCA1CTL0 |= UCMSB + UCSPB; //data format
 UCA1CTL1 &= ~UCSWRST; // 结束复位
 UCA1IE |= UCRXIE; // 使能UCA1接受中断
 _EINT(); //开启中断
 while(1)
 {
 if(recv_data==1)
 {unsigned int j;
 P8OUT |= BIT1; //LED1
 for(j=0;j<30000;j++){};
 P8OUT &= ~BIT1;
 recv_data=0;
 }
 else if (recv_data==2)
 {unsigned int j;
 P3OUT |= BIT7; //LED2
 for(j=0;j<30000;j++){};
 P3OUT &= ~BIT7;
 recv_data=0;
 }
 else if (recv_data==3)
 {unsigned int j;
 P7OUT |= BIT4; //LED3
for(j=0;j<30000;j++){};
P7OUT &= ~BIT4;
```

```
recv_data=0;
 }
 else if (recv_data==4)
 {unsigned int j;
 P6OUT |= BIT3; //LED4
for(j=0;j<30000;j++){};
P6OUT &= ~BIT3;
recv_data=0;
 }
 else
 {
 }
 }
}
#pragma vector=USCI_A1_VECTOR
__interrupt void USCI_A1_ISR(void)
{
switch(__even_in_range(UCA1IV,4))
{
case 0:break;
case 2:
{
if(UCRXIFG)
{
 recv_data=UCA1RXBUF;
 }
 }
 break;
case 4:
 break;
default:break;
}
}
#pragma vector = PORT1_VECTOR
__interrupt void PORT1_ISR(void)
{
 switch(P1IV)
 {
 case 6: //S1
 {
 unsigned int j;
 unsigned char PushKey;
 PushKey=P1IFG&~P1IN;
 for(j=0;j<2;j++){}; // debounce
```

```
 if((PushKey&~P1IN)==PushKey) //If S4 is still pressed
 {
 UCA1TXBUF=1;
 }
 P2IFG &= ~BIT2;
 }
 break;
 case 8: //S2
 {
 unsigned int j;
 unsigned char PushKey;
 PushKey=P1IFG&~P1IN;
 for(j=0;j<500;j++){}; // debounce(消抖)
 if((PushKey&~P1IN)==PushKey) //If S4 is still
pressed
 {
 UCA1TXBUF=2;
 }
 P2IFG &= ~BIT3;
 }
 break;
 }
}
#pragma vector = PORT2_VECTOR
__interrupt void PORT2_ISR(void)
{switch(P2IV)
{
case 8: //S3
 {
 unsigned int j;
 unsigned char PushKey;
 PushKey=P2IFG&~P2IN;
 for(j=0;j<500;j++){};
 if((PushKey&~P2IN)==PushKey)
 {
 UCA1TXBUF=3;
 }
 P2IFG &= ~BIT3;
 }
 break;
case 14: //S4
{
 unsigned int j;
 unsigned char PushKey;
```

```
 PushKey=P2IFG&~P2IN;
 for(j=0;j<500;j++){};
 if((PushKey&~P2IN)==PushKey)
 {
 UCA1TXBUF=4;
 }
 P2IFG &= ~BIT6;
 }
 break;
}
}
```