**Appendix C – Code for Part 1**

// Lab 3 Part 1.c

// Chris Pybus and Jeff Pistacchio

//

//-------------------------------------------------------------------------------------------

// Includes

//-------------------------------------------------------------------------------------------

#include <c8051f120.h> // SFR declarations.

#include <stdio.h> // Necessary for printf.

#include "putget.h" // Necessary for printf

//-------------------------------------------------------------------------------------------

// Function PROTOTYPES

//-------------------------------------------------------------------------------------------

void main**(**void**);**

void TIMER\_Init**(**void**);**

void PORT\_Init**(**void**);**

void SYSCLK\_Init**(**void**);**

void UART\_Init**(**void**);**

void getChar**(**void**);**

//-------------------------------------------------------------------------------------------

// MAIN Routine

//-------------------------------------------------------------------------------------------

void main **(**void**)**

**{**

WDTCN **=** 0xDE**;** // Disable watchdog timer.

WDTCN **=** 0xAD**;**

SFRPAGE **=** CONFIG\_PAGE**;**

SYSCLK\_Init**();** // Initialize the oscillator.

TIMER\_Init**();** // Initialize timers

UART\_Init**();** // Initialize UARTs.

PORT\_Init**();** // Configure the Crossbar and GPIO.

SFRPAGE **=** UART0\_PAGE**;**

printf**(**"\033[2J"**);** //Erase screen and move cursor to the home position.

printf**(**"This is UART0.\n\r"**);**

SFRPAGE **=** UART1\_PAGE**;**

printf**(**"\033[2J"**);** //Erase screen and move cursor to the home position.

printf**(**"This is UART1.\n\r"**);**

**while(**1**)**

**{**

getChar**();**

**}**

**}**

void getChar**(**void**)**

**{**

char c**;**

**while(**1**)**

**{**

SFRPAGE **=** UART0\_PAGE**;**

**if(**RI0 **==** 1**)**

**{**

RI0 **=** 0**;**

c **=** SBUF1**;**

**break;**

**}**

SFRPAGE **=** UART1\_PAGE**;**

**if(**RI1 **==** 1**)**

**{**

RI1 **=** 0**;**

c **=** SBUF1**;**

**break;**

**}**

**}**

SFRPAGE **=** UART1\_PAGE**;**

printf**(**"%c\n\r"**,** c**);**

SFRPAGE **=** UART0\_PAGE**;**

printf**(**"%c\n\r"**,** c**);**

**}**

//-------------------------------------------------------------------------------------------

// Timer\_Init

//-------------------------------------------------------------------------------------------

void TIMER\_Init**(**void**)**

**{**

//Timer1, used for UART1, baud 115200

SFRPAGE **=** TIMER01\_PAGE**;**

TCON **=** 0x40**;**

TMOD **=** 0x20**;**

CKCON **=** 0x10**;**

TH1 **=** 0xA0**;**

//Timer2, used for UART0, baud 115200

SFRPAGE **=** TMR2\_PAGE**;**

TMR2CN **=** 0x04**;**

TMR2CF **=** 0x08**;**

RCAP2L **=** 0xF4**;**

RCAP2H **=** 0xFF**;**

**}**

//-------------------------------------------------------------------------------------------

// PORT\_Init

//-------------------------------------------------------------------------------------------

void PORT\_Init**()**

**{**

SFRPAGE **=** CONFIG\_PAGE**;**

XBR0 **=** 0x04**;** //eanble UART0

XBR1 **=** 0x00**;**

XBR2 **=** 0x44**;** //enable UART1

P0MDOUT **=** 0x05**;**

P0 **=** **~**0x05**;**

**}**

//-------------------------------------------------------------------------------------------

// SYSCLK\_Init

//-------------------------------------------------------------------------------------------

void SYSCLK\_Init**(**void**)**

**{**

int i **=** 0**;**

SFRPAGE **=** CONFIG\_PAGE**;**

OSCXCN **=** 0x67**;**

**for** **(**i **=** 0**;** i **<** 3000**;** i**++);** // Wait 1ms for initialization

**while** **((**OSCXCN **&** 0x80**)** **==** 0**);**

CLKSEL **=** 0x01**;**

OSCICN **&=** **~**0x80**;**

**}**

//-------------------------------------------------------------------------------------------

// UART0\_Init

//-------------------------------------------------------------------------------------------

void UART\_Init**()**

**{**

//UART0

SFRPAGE **=** UART0\_PAGE**;**

SCON0 **=** 0x50**;**

SSTA0 **=** 0x05**;**

TI0 **=** 1**;**

//UART1

SFRPAGE **=** UART1\_PAGE**;**

SCON1 **=** 0x30**;**

TI1 **=** 1**;**

**}**

**Appendix D – Code for Part 2**

// Lab 3 Part 2.c

// Chris Pybus and Jeff Pistacchio

//

//-------------------------------------------------------------------------------------------

// Includes

//-------------------------------------------------------------------------------------------

#include <c8051f120.h> // SFR declarations.

#include <stdio.h> // Necessary for printf.

#include "putget.h" // Necessary for printf

//-------------------------------------------------------------------------------------------

// Function PROTOTYPES

//-------------------------------------------------------------------------------------------

void main**(**void**);**

void TIMER\_Init**(**void**);**

void PORT\_Init**(**void**);**

void SYSCLK\_Init**(**void**);**

void UART\_Init**(**void**);**

void getChar**(**void**);**

void UART0int **(**void**)** \_\_interrupt 4**;**

void UART1int **(**void**)** \_\_interrupt 20**;**

void sendToUART0**(**char c**);**

void sendToUART1**(**char c**);**

char sendtouart0 **=** 0**;**

char sendtouart1 **=** 0**;**

char uart1char **=** 0**;**

char uart0char **=** 0**;**

char sendingtouart1 **=** 0**;**

//-------------------------------------------------------------------------------------------

// MAIN Routine

//-------------------------------------------------------------------------------------------

void main **(**void**)**

**{**

WDTCN **=** 0xDE**;** // Disable watchdog timer.

WDTCN **=** 0xAD**;**

SFRPAGE **=** CONFIG\_PAGE**;**

SYSCLK\_Init**();** // Initialize the oscillator.

TIMER\_Init**();** // Initialize timers

UART\_Init**();** // Initialize UARTs.

PORT\_Init**();** // Configure the Crossbar and GPIO.

SFRPAGE **=** UART0\_PAGE**;**

printf**(**"\033[2J"**);** //Erase screen and move cursor to the home position.

printf**(**"This is UART0.\n\r"**);**

SFRPAGE **=** UART1\_PAGE**;**

printf**(**"\033[2J"**);** //Erase screen and move cursor to the home position.

printf**(**"This is UART1.\n\r"**);**

IE **=** 0x90**;** //enable all interrupts, and uart0 interrupt

EIE2 **=** 0x40**;** //enable uart1 interrupts

**while(**1**)**

**{**

**if(**sendtouart0 **==** 1 **||** sendtouart1 **==** 1**)**

**{**

**if(**sendtouart0 **==** 1**)**

**{**

sendToUART0**(**uart0char**);**

sendtouart0 **=** 0**;**

uart0char **=** 0**;**

**}**

**else** **if(**sendtouart1 **==** 1**)**

**{**

sendToUART1**(**uart1char**);**

sendtouart1 **=** 0**;**

uart1char **=** 0**;**

**}**

**}**

**}**

**}**

//-------------------------------------------------------------------------------------------

// Interrupts

//-------------------------------------------------------------------------------------------

void UART0int **(**void**)** \_\_interrupt 04

**{**

char c**;**

IE**=**0x00**;** //Turning off interrupt to prevent recursion

SFRPAGE **=** UART0\_PAGE**;**

**if(**RI0**==**1**)**

**{**

c**=**SBUF0**;** //read in character from buffer, page 0

RI0 **=** 0**;** //set flag back to low

sendtouart1 **=** 1**;**

uart1char **=** c**;**

sendingtouart1**=**1**;**

**}**

SBUF0**=**c**;** //send character back to console, page 0

**while(**TI0**==**0**);** //wait until TI0 is high because flag goes high when transmit is complete

TI0**=**0**;** //set flag back to 0

IE**=**0x90**;** //turn flag back on

**return;**

**}**

void UART1int **(**void**)** \_\_interrupt 20

**{**

char c**;**

IE**=**0x00**;** //Turning off interrupt to prevent recursion

SFRPAGE **=** UART1\_PAGE**;**

**if(**sendingtouart1**==**1**)** //if interrupt is coming from UART0, simply print to screen (which will transmit to other UART1)

**{**

sendingtouart1 **=** 0**;**

**if(**RI1**==**1**)**

**{**

c**=**SBUF1**;** //read in character from buffer, Page 1

RI1 **=** 0**;** //set flag back to low

sendtouart0 **=** 1**;**

uart0char **=** c**;**

**}**

SBUF1**=**c**;** //send character back to console, page 1

**while(**TI1**==**0**);** //wait until TI0 is high because flag goes high when transmit is complete

TI1**=**0**;** //set flag back to 0

**}**

**else** //if transmit is coming from other UART1, simply send along to UART0. Do not print to screen.

**{**

**if(**RI1**==**1**)**

**{**

c**=**SBUF1**;** //read in character from buffer, Page 1

RI1 **=** 0**;** //set flag back to low

sendtouart0 **=** 1**;**

uart0char **=** c**;**

**}**

**}**

IE**=**0x90**;** //turn flag back on

**return;**

**}**

//-------------------------------------------------------------------------------------------

// Other Functions

//-------------------------------------------------------------------------------------------

void sendToUART0**(**char c**)**

**{**

IE**=**0x00**;**

SFRPAGE **=** UART0\_PAGE**;**

SBUF0**=**c**;** //send character back to console

**while(**TI0**==**0**);** //wait until TI0 is high

TI0**=**0**;** //set flag back to 0

IE**=**0x90**;** //turn flag back on

**}**

void sendToUART1**(**char c**)**

**{**

IE**=**0x00**;**

SFRPAGE **=** UART1\_PAGE**;**

SBUF1**=**c**;** //send character back to console

**while(**TI1**==**0**);** //wait until TI0 is high

TI1**=**0**;** //set flag back to 0

IE**=**0x90**;** //turn flag back on

**}**

//-------------------------------------------------------------------------------------------

// Timer\_Init

//-------------------------------------------------------------------------------------------

void TIMER\_Init**(**void**)**

**{**

//Timer1, used for UART1, baud 115200

SFRPAGE **=** TIMER01\_PAGE**;**

TCON **=** 0x40**;**

TMOD **=** 0x20**;**

CKCON **=** 0x10**;**

TH1 **=** 0xA0**;**

//Timer2, used for UART0, baud 115200

SFRPAGE **=** TMR2\_PAGE**;**

TMR2CN **=** 0x04**;**

TMR2CF **=** 0x08**;**

RCAP2L **=** 0xF4**;**

RCAP2H **=** 0xFF**;**

**}**

//-------------------------------------------------------------------------------------------

// PORT\_Init

//-------------------------------------------------------------------------------------------

void PORT\_Init**()**

**{**

SFRPAGE **=** CONFIG\_PAGE**;**

XBR0 **=** 0x04**;** //eanble UART0

XBR1 **=** 0x00**;**

XBR2 **=** 0x44**;** //enable UART1

P0MDOUT **=** 0x05**;**

P0 **=** **~**0x05**;**

**}**

//-------------------------------------------------------------------------------------------

// SYSCLK\_Init

//-------------------------------------------------------------------------------------------

void SYSCLK\_Init**(**void**)**

**{**

int i **=** 0**;**

SFRPAGE **=** CONFIG\_PAGE**;**

OSCXCN **=** 0x67**;**

**for** **(**i **=** 0**;** i **<** 3000**;** i**++);** // Wait 1ms for initialization

**while** **((**OSCXCN **&** 0x80**)** **==** 0**);**

CLKSEL **=** 0x01**;**

OSCICN **&=** **~**0x80**;**

**}**

//-------------------------------------------------------------------------------------------

// UART0\_Init

//-------------------------------------------------------------------------------------------

void UART\_Init**()**

**{**

//EIP2 = 0x40; //set UART1 interrupt priority to high

//UART0

SFRPAGE **=** UART0\_PAGE**;**

SCON0 **=** 0x50**;**

SSTA0 **=** 0x05**;**

TI0 **=** 1**;**

//UART1

SFRPAGE **=** UART1\_PAGE**;**

SCON1 **=** 0x30**;**

TI1 **=** 1**;**

**}**

**Appendix E – Code for Part 3**

// Lab 3 Part 3.c

// Chris Pybus and Jeff Pistacchio

//

//-------------------------------------------------------------------------------------------

// Includes

//-------------------------------------------------------------------------------------------

#include <c8051f120.h> // SFR declarations.

#include <stdio.h> // Necessary for printf.

#include "putget.h" // Necessary for printf

\_\_sbit \_\_at **(**0x82**)** MOSI **;**

\_\_sbit \_\_at **(**0x83**)** MISO **;**

\_\_sbit \_\_at **(**0x84**)** SCK **;**

\_\_sbit \_\_at **(**0x85**)** NSS **;**

//-------------------------------------------------------------------------------------------

// Function PROTOTYPES

//-------------------------------------------------------------------------------------------

void main**(**void**);**

void TIMER\_Init**(**void**);**

void PORT\_Init**(**void**);**

void SYSCLK\_Init**(**void**);**

void UART\_Init**(**void**);**

void SPI\_Init**(**void**);**

char SPI\_Transfer **(**char SPI\_byte**);**

//-------------------------------------------------------------------------------------------

// MAIN Routine

//-------------------------------------------------------------------------------------------

void main **(**void**)**

**{**

char byte**;**

WDTCN **=** 0xDE**;** // Disable watchdog timer.

WDTCN **=** 0xAD**;**

SFRPAGE **=** CONFIG\_PAGE**;**

SYSCLK\_Init**();** // Initialize the oscillator.

TIMER\_Init**();** // Initialize timers

UART\_Init**();** // Initialize UARTs.

PORT\_Init**();** // Configure the Crossbar and GPIO.

SPI\_Init**();**

**while(**1**)**

**{**

printf**(**"Testing write \n\r"**);**

//get data from keyboard

byte **=** getchar**();**

//send data on spi

SPI\_Transfer**(**byte**,** 0**);**

//recieve data from spi

byte **=** SPI\_Transfer**(**0**,**1**);**

//print on screen

printf**(**"Data recieved: %c\n\r"**,** byte**);**

**}**

**}**

//-------------------------------------------------------------------------------------------

// Timer\_Init

//-------------------------------------------------------------------------------------------

void TIMER\_Init**(**void**)**

**{**

//Timer1, used for UART1, baud 115200

SFRPAGE **=** TIMER01\_PAGE**;**

TCON **=** 0x40**;**

TMOD **=** 0x20**;**

CKCON **=** 0x10**;**

TH1 **=** 0xA0**;**

//Timer2, used for UART0, baud 115200

SFRPAGE **=** TMR2\_PAGE**;**

TMR2CN **=** 0x04**;**

TMR2CF **=** 0x08**;**

RCAP2L **=** 0xF4**;**

RCAP2H **=** 0xFF**;**

**}**

//-------------------------------------------------------------------------------------------

// PORT\_Init

//-------------------------------------------------------------------------------------------

void PORT\_Init**()**

**{**

SFRPAGE **=** CONFIG\_PAGE**;**

P0MDOUT **=** 0x35**;**

XBR0 **=** 0x06**;**

XBR1 **=** 0x00**;**

XBR2 **=** 0x40**;**

**}**

//-------------------------------------------------------------------------------------------

// SYSCLK\_Init

//-------------------------------------------------------------------------------------------

void SYSCLK\_Init**(**void**)**

**{**

int i **=** 0**;**

SFRPAGE **=** CONFIG\_PAGE**;**

OSCXCN **=** 0x67**;**

**for** **(**i **=** 0**;** i **<** 3000**;** i**++);** // Wait 1ms for initialization

**while** **((**OSCXCN **&** 0x80**)** **==** 0**);**

CLKSEL **=** 0x01**;**

OSCICN **&=** **~**0x80**;**

**}**

//-------------------------------------------------------------------------------------------

// UART0\_Init

//-------------------------------------------------------------------------------------------

void UART\_Init**()**

**{**

//IE = 0x90; //enable all interrupts, and uart0 interrupt

//UART0

SFRPAGE **=** UART0\_PAGE**;**

SCON0 **=** 0x50**;**

SSTA0 **=** 0x05**;**

TI0 **=** 1**;**

**}**

//-------------------------------------------------------------------------------------------

// SPI

//-------------------------------------------------------------------------------------------

void SPI\_Init**(**void**)**

**{**

SFRPAGE **=** SPI0\_PAGE**;**

SPI0CFG **=** 0x40**;**

SPI0CN **=** 0x09**;**

SPI0CKR **=** 0x0A**;**

**}**

char SPI\_Transfer **(**char SPI\_byte**,** char rw**)**

**{**

//rw determines read or write. 0 for write,1 for read

**if** rw

**{**

//Read

char c **=** 0**;**

NSSMD0 **=** 0**;** //release slave select

**for(**int i **=** 0**;** i**<**256**;** i**++);** //wait 1-2ms

SPI0DAT **=** 0xFF**;** //write dummy bit to register

**while(!**SPIF**);** //Check if SPI is busy

SPIF **=** 0**;** //clear flag

c **=** SPI0DAT**;** //read data

**return** c

**}**

**else** **if** **!**rw

**{**

//write

NSSMD0 **=** 1**;** //enable slave select

**while(!**SPIF**);** //check if spi is busy

SPI0DAT **=** SPI\_byte**;** //write data to spi

**}**

**}** // END SPI\_Transfer