**Appendix C – Code for Lab 5**

// Lab 5

// Chris Pybus and Jeff Pistacchio

//

// This program reads and writes characters to external memory

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

// Includes

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

#include <c8051f120.h>

#include <stdio.h>

#include "putget.h"

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

// Global Constants

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

#define EXTCLK 22118400 // External oscillator frequency in Hz

#define SYSCLK 22118400 // Output of crystal oscillator

#define BAUDRATE 28800 // UART baud rate in bps

unsigned char \_sdcc\_external\_startup**(**void**)**

**{**

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

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

**return** 0**;** // init everything else normally

**}**

void SYSCLK\_INIT**(**void**)**

**{**

int i**;**

char SFRPAGE\_SAVE**;**

SFRPAGE\_SAVE **=** SFRPAGE**;** // Save Current SFR page SFRPAGE = CONFIG\_PAGE;

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

OSCXCN **=** 0x67**;** // Start ext osc with 22.1184MHz crystal

**for(**i**=**0**;** i **<** 3000**;** i**++);** // Wait for the oscillator to start up

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

CLKSEL **=** 0x01**;** // Switch to the external crystal oscillator

OSCICN **=** 0x00 **;** // Disable the internal oscillator

SFRPAGE **=** SFRPAGE\_SAVE**;** // Restore SFR page

**}**

void PORT\_INIT**(**void**)**

**{**

char SFRPAGE\_SAVE **=** SFRPAGE**;** // Save Current SFR page

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

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

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

XBR2 **=** 0x40**;** // Enable Crossbar and weak pull-up

P0MDOUT **|=** 0x01**;** // Set TX0 pin to push-pull

P4MDOUT **=** 0xFF**;** // Output configuration for P4 all pushpull

P5MDOUT **=** 0xFF**;** // Output configuration for P5 pushpull EM addr

P6MDOUT **=** 0xFF**;** // Output configuration for P6 pushpull EM addr

P7MDOUT **=** 0xFF**;** // Output configuration for P7 pushpull EM data

P5 **=** 0xFF**;**

P6 **=** 0xFF**;**

P7 **=** 0xFF**;**

// EMI\_Init, split mode with no banking

SFRPAGE **=** EMI0\_PAGE**;**

EMI0CF **=** 0x3b**;** //34

EMI0TC **=** 0xFF**;**

SFRPAGE **=** SFRPAGE\_SAVE**;** // Restore SFR page

**}**

void UART0\_INIT**(**void**)**

**{**

char SFRPAGE\_SAVE**;**

SFRPAGE\_SAVE **=** SFRPAGE**;** // Save Current SFR page

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

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

TMOD **&=** 0x0F**;**

TMOD **|=** 0x20**;** // Timer1, Mode 2, 8-bit reload

CKCON **|=** 0x10**;** // Timer1 uses SYSCLK as time base

TH1 **=** 0xE8**;** // 0xE8 = 232

TR1 **=** 1**;** // Start Timer1

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

SCON0 **=** 0x50**;** // Mode 1, 8-bit UART, enable RX

SSTA0 **=** 0x00**;** // SMOD0 = 0, in this mode

// TH1 = 256 - SYSCLK/(baud rate \* 32)

TI0 **=** 1**;** // Indicate TX0 ready

SFRPAGE **=** SFRPAGE\_SAVE**;** // Restore SFR page

**}**

void writeMem**(**unsigned short int address**,** unsigned char var**)**

**{**

\_\_xdata unsigned char **\***ext\_ram**;**

ext\_ram **=** **(**\_\_xdata unsigned char **\*)(**address**);**

**\***ext\_ram**=**var**;**

**}**

unsigned char readMem**(**unsigned short int address**)**

**{**

\_\_xdata unsigned char **\***ext\_ram**;**

ext\_ram **=** **(**\_\_xdata unsigned char **\*)(**address**);**

**return** **\***ext\_ram**;**

**}**

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

**{**

char str**[]** **=** "This class suck "**;**

unsigned short int address **=** 0x1FF0**;**

//unsigned char a;

SYSCLK\_INIT**();** // Initialize the oscillator

PORT\_INIT**();** // Initialize the Crossbar and GPIO

UART0\_INIT**();** // Initialize UART0

SFRPAGE **=** UART0\_PAGE**;** // Direct output to UART0

printf**(**"\033[2J"**);** // Erase ANSI terminal & move cursor to home position

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

**{**

**{**

writeMem**(**address**,** str**[**address**%**16**]);**

printf**(**"%X: %c \r\n"**,** address**,** readMem**(**address**));**

**}**

**}**

**}**