MPS

IDE & ANSI Display Lab Exercise

Using the IDE and ANSI Display Commands

Student's name & ID (1): ____

Partner's name & ID (2): ____

Your Section number & TA's name _______________________________________

Notes:

You must work on this assignment with your partner.

Hand in a printer copy of your software listings for the team.
Hand in a neat copy of your circuit schematics for the team.

These will be returned to you so that they may be used for reference.

----------------------------- do not write below this line ----------------------------

	POINTS		TA init.
	(1)	(2)	
Grade for performance verification (30% max.)			
Grade for answers to TA's questions (20% max.)			
Grade for documentation and appearance (50% max.)			
			TOTAL

Grader's signature: ____

Date: ____

Using the IDE and Programming an ANSI Display

GOAL

By doing this lab assignment, you will learn:

1. Exercise the 8051 CPU registers and built-in hardware.
2. To use the VT100 Terminal Interface (with HyperTerminal or ProComm Plus) on the 8051.
3. To program an ANSI terminal display through a C program using the SDCC Compiler.
4. Perform basic I/O with the 8051.

PREPARATION

- Review the 8051 hardware features.
- Review the C language stdio utilities.
- References: C8051F12x-13x.pdf 8051 *C8051F12X Reference Manual, Ch. 1, 11, 18*
  C8051F12x-DK.pdf *C8051F12X Development Kit User's Guide*
  VT100/ANSI ESCAPE SEQUENCES and Hello.c sample program

IDE SETUP

1. Connecting The Board

See the *C8051F12X Development Kit User's Guide* for details on connecting the board and USB Debug Adapter (http://www.ecse.rpi.edu/courses/CStudio/Silabs/C8051F12x-DK.pdf).

2. Using The IDE

Start the SiLabs IDE by clicking on *Silicon Laboratories* on the *Start Menu* and choosing *Silicon Laboratories IDE*. Next, create a new project to which C files can be added. To create a new project, click on the **Project** menu and select **New**. This will create an empty project. Add a new C file to the project by clicking on the **Project** menu and selecting **Add Files to Project**. Give your new file a name, such as the *Hello.c* included in this lab. The file will now be part of the project, and can be compiled. Save the project by clicking on **Project** and selecting **Save Project**. **WARNING**: The project must be saved in a location whose path contains no folders with space characters in its name. It is suggested that you use C:\MPSfiles\filename.wsp.

To compile your project, first verify that all of the code is correct. Click on the **Project** menu and select **Assemble/Compile File**. The window at the bottom of the IDE will alert you of any errors or warnings it finds in your code. If there are errors, correct them first before moving on to the next step. If there are warnings, you may or may not want to correct them, depending on the nature of the warning. Click on the **Project** menu and select **Build/Make Project**. After successfully building your project, you will want to upload it to the development board. Before doing this, you will need to configure the adapter used to download the source code. To do this, click on the **Options** menu and select **Connection Options**. Once the window opens, select the *USB Debug Adapter* and click **OK**. You are now ready to download the code. To do this, click on the **Debug** menu, and select **Connect**. This will connect the IDE to the development board through the USB Debug Adapter. Click on the **Debug** menu again, and select **Download Object File**. Select the file you just compiled, and click **Download**. This will download the code to the development board. Finally, to execute your code on the C8051F120, click on the **Debug** menu, and select **Go**.

ANSI PROGRAMMING TASKS FOR THE 8051

1. Introduction To The User Interface

Input from the terminal keyboard and output to the terminal display can be done using the getchar(a) and putchar(a) functions. Write a simple C program to run on the 8051 that outputs `"The keyboard character is *."` whenever you type a printable character, where * stands for that character. Since you will be waiting for an indefinite number of characters to be typed, use <ESC> (or ^[key combination, where ^ = <Control>) to end the program. Display this information at the top of the screen when the program starts.

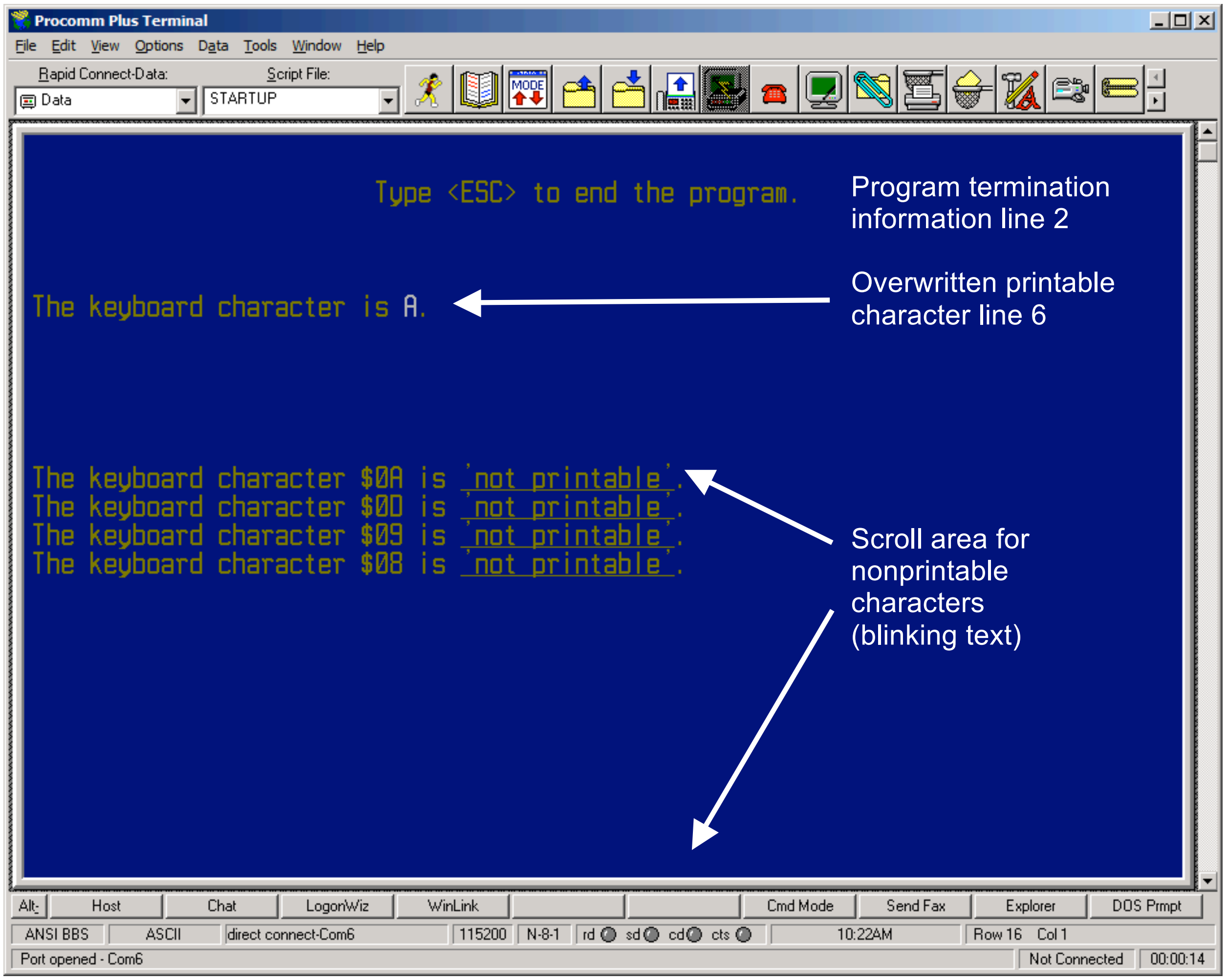
2. VT100/ANSI Terminal Control Escape Sequences

ProComm Plus and HyperTerminal use VT100/ANSI terminal emulation by default. By sending special codes to the terminal, it is possible to clear the screen, position the cursor, set terminal colors, and many other operations. These codes are called escape sequences because the first character is the <ESC> character, or $1B in ASCII (033 octal). A table is included at the end of this lab that contains some useful escape codes.

Modify the C program of Part 1 to display yellow characters on a blue background. Center the program termination information on line 2. Display the keyboard response text on line 6. Change the color of the keyed in character to white (leaving the rest of the characters in yellow). Now for nonprinting characters, have the program blink the output `"The keyboard character $XX is 'not printable'."` and beep, where XX are the hexadecimal digits of the nonprintable character (include the underline on the terminal). This line should appear at the line in the center of the screen (e.g. line 12 if 24 lines are being displayed in the terminal window). Also, although the top message may overwrite the previous message each time a new key is hit, the 'not printable' message should be written on the line beneath the previous line. When it writes on the last line of the screen, the terminal should scroll just the lines on the lower half of the screen. See the figure below of a terminal screen shot. Note that scrolling only occurs when you output a '\n' while on the bottom line of the display. Moving the cursor to the bottom line and writing will only overwrite any text already there. There are escape sequences that simplify these operations, which you are expected to use as much as possible. Unfortunately, the PCs and PC cards in the Sun workstations do not have external speakers so headphones are needed to hear the beep and ProComm must be used instead of HyperTerminal. The version of HyperTerminal used on the PCs does not seem to support the beep sound (^G = Ctrl-G or ASCII 'BEL').

NOTE: If the terminal doesn't respond properly to escape sequences, it may no longer be in an ANSI compatible mode. Make sure the leftmost bottom parameter button in the HyperTerminal window is set to either VT100 or ANSI BBS and likewise for ProComm Plus.

Good programmer's tip: Design the program top-down. Then write the routines bottom-up. Write them one at a time and thoroughly test each one before integrating them. This way you will have isolated any errors to the routine that you are currently writing. Good programmers follow this method.



```
                         Type <ESC> to end the program.



The keyboard character is A.





The keyboard character $0A is 'not printable'.
The keyboard character $0D is 'not printable'.
The keyboard character $09 is 'not printable'.
The keyboard character $08 is 'not printable'.
```

Program termination information line 2

Overwritten printable character line 6

Scroll area for nonprintable characters (blinking text)

3. Port Input/Output

Configure the 8051 to properly execute the following tasks:

- Set Port 2 to output port bits (use a voltmeter or a TA provided 6-LED module with series current limiting resistors to verify values). You will first need to set the corresponding bits in the port's data direction register (P2MDOUT) to a 1 for output and use the latch register (P2) to write the bits for output. See the C8051F120 manual, pp. 250-251, for details.

- Set Port 1 to input logic levels applied to port pins (use 0 V (ground) for a logic low and +5 V through a 1 k resistor for a logic high). Again, you will first need to set the registers (P1MDIN & P1MDOUT on pp. 249-250) bits for input and read from the latch register (P1 on p. 249).

- Continuously read in a value on a bit on input Port 1 and output the value to the corresponding bit on Port 2, which has been connected to an LED. After wiring the LEDs to the correct output pins and applying a voltage to the input pins, the minimum steps necessary to read in a voltage on pin P1.0 (Port 1) are:

  1. Enable the crossbar (XBR0, XBR1 & XBR2: C8051F120 manual, pp. 245-247).

  2. Configure P1.n (n = 7 – 0) to open drain; P2.n (n = 7 – 0) to push-pull.

3. Read the 8-bit numerical value from P1 into a local variable.

4. Write the value to P2.

   If a potentiometer (use the pot modules assigned to lab groups by the TA) whose end terminals are between ground and +5 V with the wiper connected to the A/D input is used to give various voltage readings, you should be able to confirm that voltages near 0 V yields a numerical value of 0, and +5 V yields values yields 1. Explain why this makes sense. Determine if the input uses a Schmitt trigger (hysteresis) on voltage levels.

NOTE: Make sure you provide a common ground connection between the +5V of your external TTL circuit and the +3.3V of the C8051F120 board. Also be sure the +5V supply is NOT connected to the +3.3V supply on the microcontroller. It isn't critical, but probably a little safer, if the potentiometers on the A/D Converter Input Module are connected to +3.3V and ground from the C8051F120 board rather than +5V.

VT100/ANSI ESCAPE SEQUENCES

Name	Escape Code	Hexadecimal	Description
Reset Device	<ESC>c	$1B $63	Resets all terminal settings to default.
Enable Line Wrap	<ESC>[7h	$1B $5B $37 $68	Enables wrapping text to the next line if text is longer than the display area.
Disable Line Wrap	<ESC>[7l	$1B $5B $37 $6C	Disables wrapping text; text will be clipped if longer than display area.
Cursor Home	<ESC>[H	$1B $5B $48	Moves the cursor to the home position (upper left hand corner).
Cursor Position	<ESC>[{ROW};{COL}H	$1B $5B ${ROW} $3B ${COL} $48	Sets the position of the cursor at ({ROW}, {COL}).
Cursor Up	<ESC>[{NUM}A	$1B $5B ${NUM} $41	Moves the cursor up {NUM} rows; {NUM} defaults to 1 if omitted.
Cursor Down	<ESC>[{NUM}B	$1B $5B ${NUM} $42	Moves the cursor down {NUM} rows; {NUM} defaults to 1 if omitted.
Cursor Left	<ESC>[{NUM}D	$1B $5B ${NUM} $44	Moves the cursor left {NUM} columns; {NUM} defaults to 1 if omitted.
Cursor Right	<ESC>[{NUM}C	$1B $5B ${NUM} $43	Moves the cursor right {NUM} columns; {NUM} defaults to 1 if omitted.
Save Cursor	<ESC>[s	$1B $5B $73	Saves the current cursor position.
Restore Cursor	<ESC>[u	$1B $5B $75	Restores the previously stored cursor position.
Erase End of Line	<ESC>[K	$1B $5B $4B	Erases from the current cursor position to the end of the current row.
Erase Start of Line	<ESC>[1K	$1B $5B $31 $4B	Erases from the start of the current row to the current cursor position.
Erase Line	<ESC>[2K	$1B $5B $32 $4B	Erases the entire current row.
Erase Down	<ESC>[J	$1B $5B $4A	Erases from the current row down to the bottom of the screen.
Erase Up	<ESC>[1J	$1B $5B $31 $4A	Erases from the current row to the top of the screen.
Erase Screen	<ESC>[2J	$1B $5B $32 $4A	Erases the entire screen and moves the cursor to the home position.
Scroll All	<ESC>[r	$1B $5B $72	Enables scrolling for the entire display.
Scroll Section	<ESC>[{SRT};{END}r	$1B $5B ${SRT} $3B ${END} $72	Enables scrolling only for rows {SRT} to {END}.
Scroll Down	<ESC>D	$1B $44	Scrolls the display down one line.
Scroll Up	<ESC>M	$1B $4D	Scrolls the display up one line.
Attribute Mode set	<ESC>[{ATR1};...;{ATRn}m	$1B $5B ${ATR1} $3B ... $3B ${ATRn} $6D	Sets multiple display attribute settings; any number can be set. ATRn may be any of the following values:
Standard Values for Attribute Mode Set	0 Reset Attributes 1 Bright 2 Dim 4 Underscore 5 Blink 7 Reverse 8 Hidden	**Foreground Colors** 30 Black 31 Red 32 Green 33 Yellow 34 Blue 35 Magenta 36 Cyan 37 White	**Background Colors** 40 Black 41 Red 42 Green 43 Yellow 44 Blue 45 Magenta 46 Cyan 47 White

Do not include the ‘{‘ or ‘}’ characters in the print statement.
Ex.) <ESC>[{ROW 10};{COL 20}H would be “\033[10;20H”.

Table 11.3. Special Function Registers

SFRs are listed in alphabetical order. All undefined SFR locations are reserved.

Register	Address	SFR Page	Description	Page No.
ACC	0xE0	All Pages	Accumulator	page 153
ADC0CF	0xBC	0	ADC0 Configuration	page 62[1], page 80[2]
ADC0CN	0xE8	0	ADC0 Control	page 63[1], page 81[2]
ADC0GTH	0xC5	0	ADC0 Greater-Than High Byte	page 66[1], page 84[2]
ADC0GTL	0xC4	0	ADC0 Greater-Than Low Byte	page 66[1], page 84[2]
ADC0H	0xBF	0	ADC0 Data Word High Byte	page 64[1], page 82[2]
ADC0L	0xBE	0	ADC0 Data Word Low Byte	page 64[1], page 82[2]
ADC0LTH	0xC7	0	ADC0 Less-Than High Byte	page 67[1], page 85[2]
ADC0LTL	0xC6	0	ADC0 Less-Than Low Byte	page 67[1], page 85[2]
ADC2	0xBE	2	ADC2 Data Word	page 99[3]
ADC2CF	0xBC	2	ADC2 Configuration	page 97[3]
ADC2CN	0xE8	2	ADC2 Control	page 98[3]
ADC2GT	0xC4	2	ADC2 Greater-Than	page 102[3]
ADC2LT	0xC6	2	ADC2 Less-Than	page 102[3]
AMX0CF	0xBA	0	ADC0 Multiplexer Configuration	page 60[1], page 78[2]
AMX0SL	0xBB	0	ADC0 Multiplexer Channel Select	page 61[1], page 79[2]
AMX2CF	0xBA	2	ADC2 Multiplexer Configuration	page 95[3]
AMX2SL	0xBB	2	ADC2 Multiplexer Channel Select	page 96[3]
B	0xF0	All Pages	B Register	page 153
CCH0CN	0xA1	F	Cache Control	page 215
CCH0LC	0xA3	F	Cache Lock	page 216
CCH0MA	0x9A	F	Cache Miss Accumulator	page 217
CCH0TN	0xA2	F	Cache Tuning	page 216
CKCON	0x8E	0	Clock Control	page 315
CLKSEL	0x97	F	System Clock Select	page 188
CPT0CN	0x88	1	Comparator 0 Control	page 123
CPT0MD	0x89	1	Comparator 0 Configuration	page 123
CPT1CN	0x88	2	Comparator 1 Control	page 124
CPT1MD	0x89	2	Comparator 1 Configuration	page 125
DAC0CN	0xD4	0	DAC0 Control	page 108[3]
DAC0H	0xD3	0	DAC0 High Byte	page 107[3]
DAC0L	0xD2	0	DAC0 Low Byte	page 107[3]
DAC1CN	0xD4	1	DAC1 Control	page 110[3]
DAC1H	0xD3	1	DAC1 High Byte	page 109[3]
DAC1L	0xD2	1	DAC1 Low Byte	page 109[3]
DPH	0x83	All Pages	Data Pointer High Byte	page 151
DPL	0x82	All Pages	Data Pointer Low Byte	page 151
EIE1	0xE6	All Pages	Extended Interrupt Enable 1	page 159
EIE2	0xE7	All Pages	Extended Interrupt Enable 2	page 160
EIP1	0xF6	All Pages	Extended Interrupt Priority 1	page 161

	EIP2	0xF7	All Pages	Extended Interrupt Priority 2	page 162
	EMI0CF	0xA3	0	EMIF Configuration	page 221
	EMI0CN	0xA2	0	EMIF Control	page 220
	EMI0TC	0xA1	0	EMIF Timing Control	page 226
	FLACL	0xB7	F	Flash Access Limit	page 206
	FLSCL	0xB7	0	Flash Scale	page 208
	FLSTAT	0x88	F	Flash Status	page 217
	IE	0xA8	All Pages	Interrupt Enable	page 157
	IP	0xB8	All Pages	Interrupt Priority	page 158
	MAC0ACC0	0x93	3	MAC0 Accumulator Byte 0 (LSB)	page 174[4]
	MAC0ACC1	0x94	3	MAC0 Accumulator Byte 1	page 173[4]
	MAC0ACC2	0x95	3	MAC0 Accumulator Byte 2	page 173[4]
	MAC0ACC3	0x96	3	MAC0 Accumulator Byte 3 (MSB)	page 173[4]
	MAC0AH	0xC2	3	MAC0 A Register High Byte	page 171[4]
	MAC0AL	0xC1	3	MAC0 A Register Low Byte	page 172[4]
	MAC0BH	0x92	3	MAC0 B Register High Byte	page 172[4]
	MAC0BL	0x91	3	MAC0 B Register Low Byte	page 172[4]
	MAC0CF	0xC3	3	MAC0 Configuration	page 170[4]
	MAC0OVR	0x97	3	MAC0 Accumulator Overflow	page 174[4]
	MAC0RNDH	0xCF	3	MAC0 Rounding Register High Byte	page 174[4]
	MAC0RNDL	0xCE	3	MAC0 Rounding Register Low Byte	page 175[4]
	MAC0STA	0xC0	3	MAC0 Status Register	page 171[4]
	OSCICL	0x8B	F	Internal Oscillator Calibration	page 186
	OSCICN	0x8A	F	Internal Oscillator Control	page 186
	OSCXCN	0x8C	F	External Oscillator Control	page 189
	P0	0x80	All Pages	Port 0 Latch	page 248
	P0MDOUT	0xA4	F	Port 0 Output Mode Configuration	page 248
⇒	P1	0x90	All Pages	Port 1 Latch	page 249
⇒	P1MDIN	0xAD	F	Port 1 Input Mode	page 249
⇒	P1MDOUT	0xA5	F	Port 1 Output Mode Configuration	page 250
⇒	P2	0xA0	All Pages	Port 2 Latch	page 250
⇒	P2MDOUT	0xA6	F	Port 2 Output Mode Configuration	page 251
	P3	0xB0	All Pages	Port 3 Latch	page 251
	P3MDOUT	0xA7	F	Port 3 Output Mode Configuration	page 252
	P4	0xC8	F	Port 4 Latch	page 254
	P4MDOUT	0x9C	F	Port 4 Output Mode Configuration	page 254
	P5	0xD8	F	Port 5 Latch	page 255
	P5MDOUT	0x9D	F	Port 5 Output Mode Configuration	page 255
	P6	0xE8	F	Port 6 Latch	page 256
	P6MDOUT	0x9E	F	Port 6 Output Mode Configuration	page 256
	P7	0xF8	F	Port 7 Latch	page 257
	P7MDOUT	0x9F	F	Port 7 Output Mode Configuration	page 257
	PCA0CN	0xD8	0	PCA Control	page 335
	PCA0CPH0	0xFC	0	PCA Module 0 Capture/Compare High	page 339
	PCA0CPH1	0xFE	0	PCA Module 1 Capture/Compare High	page 339

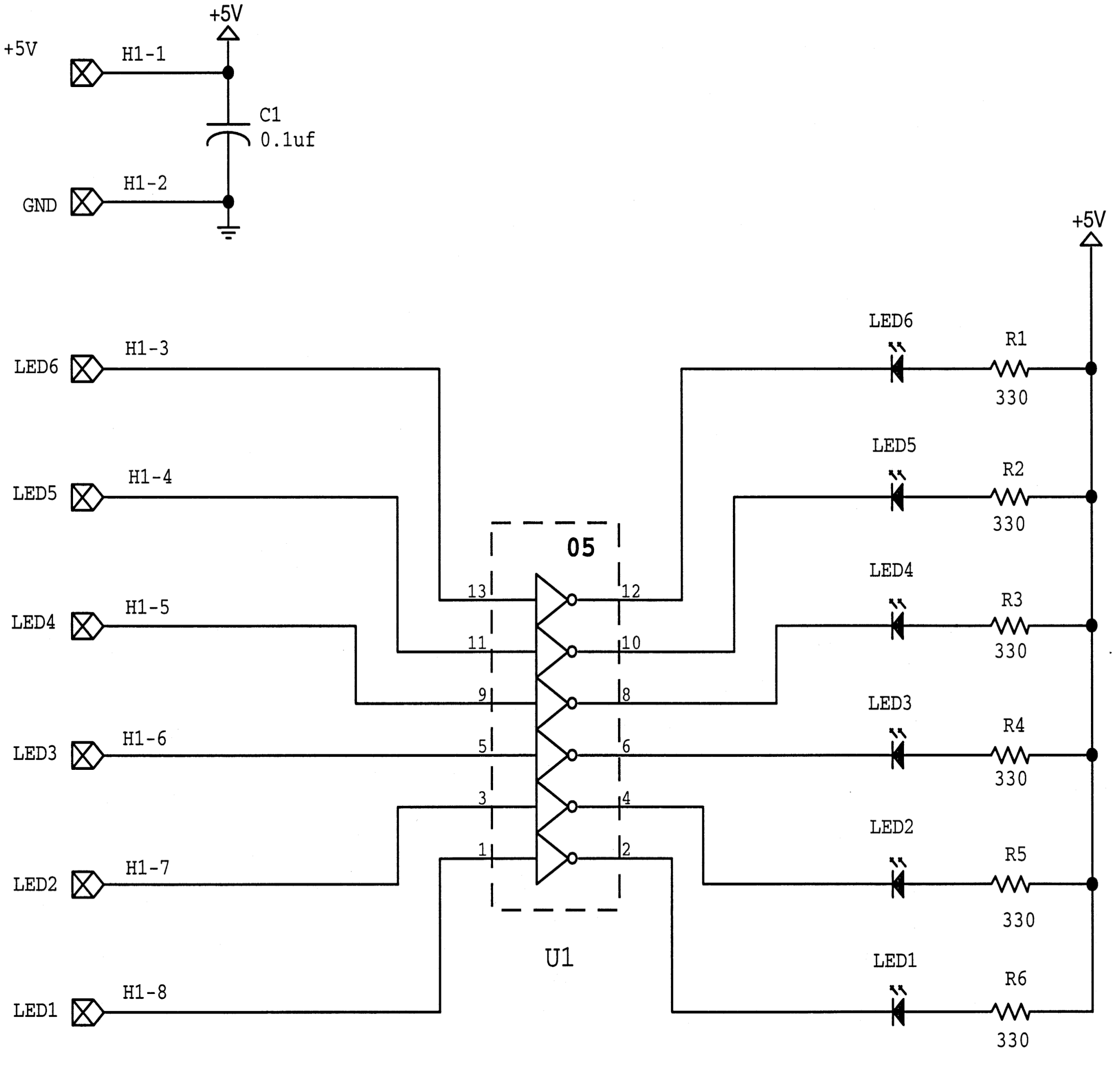
PCA0CPH2	0xEA	0	PCA Module 2 Capture/Compare High	page 339
PCA0CPH3	0xEC	0	PCA Module 3 Capture/Compare High	page 339
PCA0CPH4	0xEE	0	PCA Module 4 Capture/Compare High	page 339
PCA0CPH5	0xE2	0	PCA Module 5 Capture/Compare High	page 339
PCA0CPL0	0xFB	0	PCA Module 0 Capture/Compare Low	page 338
PCA0CPL1	0xFD	0	PCA Module 1 Capture/Compare Low	page 338
PCA0CPL2	0xE9	0	PCA Module 2 Capture/Compare Low	page 338
PCA0CPL3	0xEB	0	PCA Module 3 Capture/Compare Low	page 338
PCA0CPL4	0xED	0	PCA Module 4 Capture/Compare Low	page 338
PCA0CPL5	0xE1	0	PCA Module 5 Capture/Compare Low	page 338
PCA0CPM0	0xDA	0	PCA Module 0 Mode	page 337
PCA0CPM1	0xDB	0	PCA Module 1 Mode	page 337
PCA0CPM2	0xDC	0	PCA Module 2 Mode	page 337
PCA0CPM3	0xDD	0	PCA Module 3 Mode	page 337
PCA0CPM4	0xDE	0	PCA Module 4 Mode	page 337
PCA0CPM5	0xDF	0	PCA Module 5 Mode	page 337
PCA0H	0xFA	0	PCA Counter High Byte	page 338
PCA0L	0xF9	0	PCA Counter Low Byte	page 338
PCA0MD	0xD9	0	PCA Mode	page 336
PCON	0x87	All Pages	Power Control	page 164
PLL0CN	0x89	F	PLL Control	page 193
PLL0DIV	0x8D	F	PLL Divider	page 194
PLL0FLT	0x8F	F	PLL Filter	page 195
PLL0MUL	0x8E	F	PLL Multiplier	page 194
PSBANK	0xB1	All Pages	Flash Bank Select	page 134
PSCTL	0x8F	0	Flash Write/Erase Control	page 209
PSW	0xD0	All Pages	Program Status Word	page 152
RCAP2H	0xCB	0	Timer/Counter 2 Capture/Reload High	page 323
RCAP2L	0xCA	0	Timer/Counter 2 Capture/Reload Low	page 323
RCAP3H	0xCB	1	Timer 3 Capture/Reload High Byte	page 323
RCAP3L	0xCA	1	Timer 3 Capture/Reload Low Byte	page 323
RCAP4H	0xCB	2	Timer/Counter 4 Capture/Reload High	page 323
RCAP4L	0xCA	2	Timer/Counter 4 Capture/Reload Low	page 323
REF0CN	0xD1	0	Voltage Reference Control	page 114[5], page 116[6], page 117[7]
RSTSRC	0xEF	0	Reset Source	page 182
SADDR0	0xA9	0	UART 0 Slave Address	page 298
SADEN0	0xB9	0	UART 0 Slave Address Mask	page 298
SBUF0	0x99	0	UART 0 Data Buffer	page 298
SBUF1	0x99	1	UART 1 Data Buffer	page 305
SCON0	0x98	0	UART 0 Control	page 296
SCON1	0x98	1	UART 1 Control	page 304
SFRLAST	0x86	All Pages	SFR Stack Last Page	page 143
SFRNEXT	0x85	All Pages	SFR Stack Next Page	page 143
SFRPAGE	0x84	All Pages	SFR Page Select	page 142

SFRPGCN	0x96	F	SFR Page Control	page 142
SMB0ADR	0xC3	0	SMBus Slave Address	page 269
SMB0CN	0xC0	0	SMBus Control	page 266
SMB0CR	0xCF	0	SMBus Clock Rate	page 267
SMB0DAT	0xC2	0	SMBus Data	page 268
SMB0STA	0xC1	0	SMBus Status	page 269
SP	0x81	All Pages	Stack Pointer	page 151
SPI0CFG	0x9A	0	SPI Configuration	page 280
SPI0CKR	0x9D	0	SPI Clock Rate Control	page 282
SPI0CN	0xF8	0	SPI Control	page 281
SPI0DAT	0x9B	0	SPI Data	page 282
SSTA0	0x91	0	UART 0 Status	page 297
TCON	0x88	0	Timer/Counter Control	page 313
TH0	0x8C	0	Timer/Counter 0 High Byte	page 316
TH1	0x8D	0	Timer/Counter 1 High Byte	page 316
TL0	0x8A	0	Timer/Counter 0 Low Byte	page 315
TL1	0x8B	0	Timer/Counter 1 Low Byte	page 316
TMOD	0x89	0	Timer/Counter Mode	page 314
TMR2CF	0xC9	0	Timer/Counter 2 Configuration	page 324
TMR2CN	0xC8	0	Timer/Counter 2 Control	page 324
TMR2H	0xCD	0	Timer/Counter 2 High Byte	page 324
TMR2L	0xCC	0	Timer/Counter 2 Low Byte	page 323
TMR3CF	0xC9	1	Timer 3 Configuration	page 324
TMR3CN	0xC8	1	Timer 3 Control	page 324
TMR3H	0xCD	1	Timer 3 High Byte	page 324
TMR3L	0xCC	1	Timer 3 Low Byte	page 323
TMR4CF	0xC9	2	Timer/Counter 4 Configuration	page 324
TMR4CN	0xC8	2	Timer/Counter 4 Control	page 324
TMR4H	0xCD	2	Timer/Counter 4 High Byte	page 324
TMR4L	0xCC	2	Timer/Counter 4 Low Byte	page 323
WDTCN	0xFF	All Pages	Watchdog Timer Control	page 181
⇒ XBR0	0xE1	F	Port I/O Crossbar Control 0	page 245
⇒ XBR1	0xE2	F	Port I/O Crossbar Control 1	page 246
⇒ XBR2	0xE3	F	Port I/O Crossbar Control 2	page 247

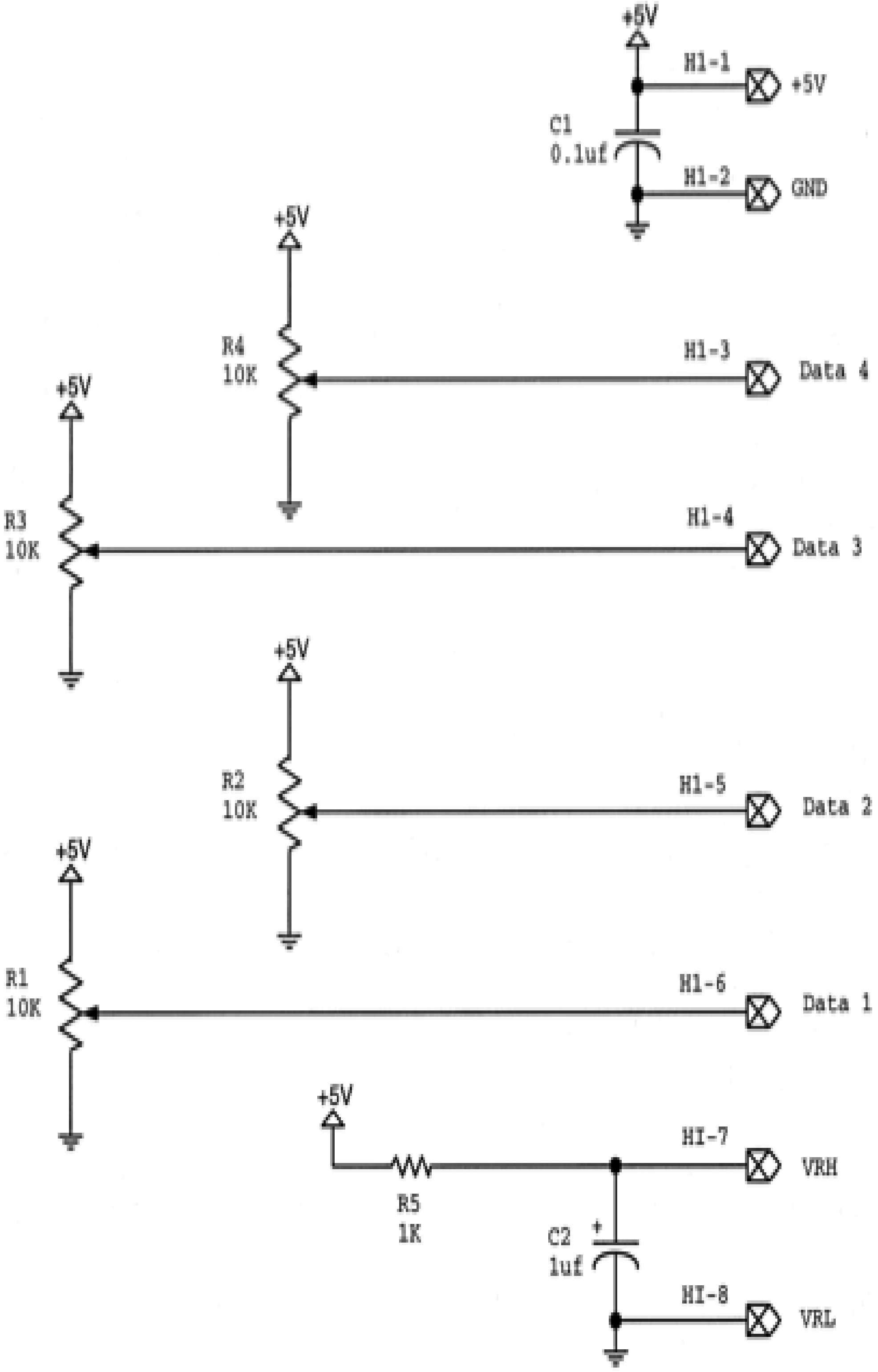
Notes:

5. **Refers to a register in the C8051F120/1/4/5 only.**
6. **Refers to a register in the C8051F122/3/6/7 and C8051F130/1/2/3 only.**
7. **Refers to a register in the C8051F120/1/2/3/4/5/6/7 only.**
8. **Refers to a register in the C8051F120/1/2/3 and C8051F130/1/2/3 only.**
9. **Refers to a register in the C8051F120/2/4/6 only.**
10. **Refers to a register in the C8051F121/3/5/7 only.**
11. **Refers to a register in the C8051F130/1/2/3 only.**

```
//------------------------------------------------------------------------------------
// Hello.c
//------------------------------------------------------------------------------------
//8051 Test program to demonstrate serial port I/O.  This program writes a message on
//the console using the printf() function, and reads characters using the getchar()
//function.  An ANSI escape sequence is used to clear the screen if a '2' is typed.
//A '1' repeats the message and the program responds to other input characters with
//an appropriate message.
//
//Any valid keystroke turns on the green LED on the board; invalid entries turn it off
//------------------------------------------------------------------------------------
// Includes
//------------------------------------------------------------------------------------
#include <c8051f120.h>
#include <stdio.h>
#include "putget.h"

//------------------------------------------------------------------------------------
// Global Constants
//------------------------------------------------------------------------------------
#define EXTCLK      22118400            // External oscillator frequency in Hz
#define SYSCLK      49766400            // Output of PLL derived from (EXTCLK * 9/4)
#define BAUDRATE    115200              // UART baud rate in bps

//------------------------------------------------------------------------------------
// Function Prototypes
//------------------------------------------------------------------------------------
void main(void);
void SYSCLK_INIT(void);
void PORT_INIT(void);
void UART0_INIT(void);

//------------------------------------------------------------------------------------
// MAIN Routine
//------------------------------------------------------------------------------------
void main(void)
{
    char choice;

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

    PORT_INIT();                        // Initialize the Crossbar and GPIO
    SYSCLK_INIT();                      // Initialize the oscillator
    UART0_INIT();                       // Initialize UART0

    SFRPAGE = UART0_PAGE;               // Direct output to UART0

    printf("\033[2J");                  // Erase screen & move cursor to home position
    printf("Test of the printf() function.\n\n");

    while(1)
    {
        printf("Hello World!\n\n\r");
        printf("( greetings from Russell P. Kraft )\n\n\n\r");
        printf("1=repeat, 2=clear, 0=quit.\n\n\r"); // Menu of choices

        choice = getchar();
//      putchar(choice);

        // select which option to run
        P1 |= 0x40;                     // Turn green LED on
        if (choice == '0')
            return;
        else if(choice == '1')
            printf("\n\nHere we go again.\n\n\r");
        else if(choice == '2')          // clear the screen with <ESC>[2J
            printf("\033[2J");
        else
        {
            // inform the user how bright he is
            P1 &= 0xBF;                 // Turn green LED off
            printf("\n\rA \"");
            putchar(choice);
            printf("\" is not a valid choice.\n\n\r");
        }
    }
}

//------------------------------------------------------------------------------------
// SYSCLK_Init
//------------------------------------------------------------------------------------
//
// Initialize the system clock to use a 22.1184MHz crystal as its clock source
//
void SYSCLK_INIT(void)
{
    int i;
    char SFRPAGE_SAVE

    SFRPAGE_SAVE = SFRPAGE;             // Save Current SFR page
    SFRPAGE = CONFIG_PAGE;

    OSCXCN  = 0x67;                     // Start ext osc with 22.1184MHz crystal
    for(i=0; i < 256; i++);             // Wait for the oscillator to start up
    while(!(OSCXCN & 0x80));
    CLKSEL  = 0x01;
    OSCICN  = 0x00;

    SFRPAGE = CONFIG_PAGE;
    PLL0CN  = 0x04;
    SFRPAGE = LEGACY_PAGE;
    FLSCL   = 0x10;
    SFRPAGE = CONFIG_PAGE;
    PLL0CN |= 0x01;
    PLL0DIV = 0x04;
    PLL0FLT = 0x01;
    PLL0MUL = 0x09;
    for(i=0; i < 256; i++);
    PLL0CN |= 0x02;
    while(!(PLL0CN & 0x10));
    CLKSEL  = 0x02;

    SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}

//------------------------------------------------------------------------------------
// PORT_Init
//------------------------------------------------------------------------------------
//
// Configure the Crossbar and GPIO ports
//
void PORT_INIT(void)
{
    char SFRPAGE_SAVE;

    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 on P0.0 pin to push-pull
    P1MDOUT |= 0x40;                    // Set green LED output P1.6 to push-pull

    SFRPAGE  = SFRPAGE_SAVE;            // Restore SFR page
}

//------------------------------------------------------------------------------------
// UART0_Init
//------------------------------------------------------------------------------------
//
// Configure the UART0 using Timer1, for <baudrate> and 8-N-1
//
void UART0_INIT(void)
{
    char SFRPAGE_SAVE;

    SFRPAGE_SAVE = SFRPAGE;             // Save Current SFR page
    SFRPAGE = TIMER01_PAGE;

    TMOD   &= ~0xF0;
    TMOD   |=  0x20;                    // Timer1, Mode 2, 8-bit reload
    TH1      = -(SYSCLK/BAUDRATE/16);     // Set Timer1 reload baudrate value T1 Hi Byte
    CKCON   |= 0x10;                      // Timer1 uses SYSCLK as time base
    TL1      = TH1;
    TR1      = 1;                         // Start Timer1

    SFRPAGE = UART0_PAGE;
    SCON0    = 0x50;                      // Mode 1, 8-bit UART, enable RX
    SSTA0    = 0x10;                      // SMOD0 = 1
    TI0      = 1;                         // Indicate TX0 ready

    SFRPAGE = SFRPAGE_SAVE;               // Restore SFR page
}
```



LED Display Board



A/D Converter Input Voltage Board