

Programming an ANSI Display

Student's name & ID: _____

Partner's name(s) & ID(s): _____

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 -----

Grade for performance verification (50% max.)

Grade for answers to TA's questions (20% max.)

Grade for documentation and appearance (30% max.)

POINTS	TA init.

Grader's signature: _____

Date: _____

Programming an ANSI Display

GOAL

By doing this lab assignment, you will learn:

1. To use the VT100 Terminal Interface (with HyperTerminal or ProComm Plus) on the M68HC12.
2. To program an ANSI terminal display through a C program using the Introl C cross-compiler.

PREPARATION

- Review the C language stdio utilities.
- Read the Introl M6812 C Cross-Compiler handout.
- References: VT100/ANSI ESCAPE SEQUENCES

PROGRAMMING TASKS FOR THE MC6812

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 MC6812 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) to terminate 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 things. These codes are called escape sequences because the first character is the `<ESC>` character, or `$1B` in ASCII. The table on the next page 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 12. 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 output "The keyboard character is 'not printable'."

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.

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

