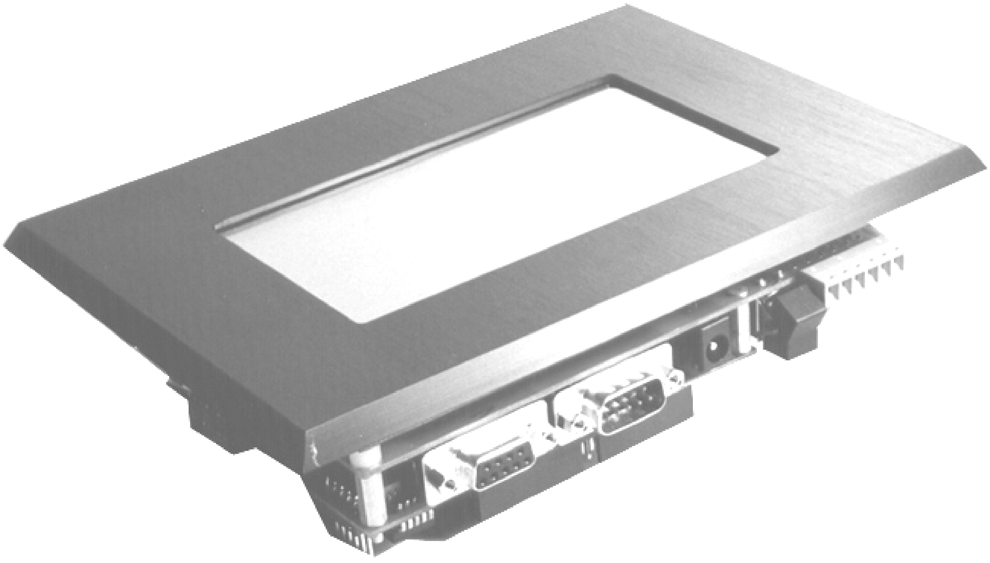
Quick Start Guide to the SmarTouch^{TM} Controller

Version 1.0



(SmarTouch Controller with Optional Bezel)

Table of Contents

Introduction

Congratulations! With your purchase of the SmarTouch Starter Kit, you have at your fingertips powerful software development tools, an I/O rich computer, and a graphical user interface. This gives you a tremendous head start on your instrument development project.

The SmarTouch Starter Kit includes the QED-Flash Board with 128K of battery-backed RAM and 256K of non-volatile flash, 128x240 pixel graphics display with cold cathode fluorescent (CCFL) backlight, 4 row by 5 column transparent touchscreen overlay, plus QED and Motorola manuals, power adapter and serial cable.

The heart of the SmarTouch Controller is the QED-Flash Board, a double sided surface mount controller that integrates many hardware and software functions in a compact package. The board includes digital input/output (I/O), counting and timing signals, high-current driver outputs, keypad and display interfaces, dual serial ports, 8-bit analog to digital (A/D) conversion, 12-bit A/D, and 8 bit digital to analog conversion. Up to 256 kilobytes of memory can be installed on the board. The onboard QED-Forth development firmware includes a high level language, assembler, debugging environment, multitasker, math library, and a comprehensive set of device drivers. These built-in functions make it very easy to get the most out of your board's computational and I/O capabilities.

The following sections guide you through the SmarTouch Controller’s hardware and explain how to establish communications with the SmarTouch Controller, install your compiler (if you are programming in C), and compile and run a program.

Getting to Know Your SmarTouch Hardware

This section describes all of the hardware that is included with the SmarTouch Starter Kit as well as a brief description about each of the important components.

In addition to this document, the following items are included in your SmarTouch Starter Kit:

Quantity	Part #	Description
1	STSK-1	SmarTouch Controller.
1	PCC9-232	9 Pin PC Serial Cable.
1	PS-6V	6 Volt Power Supply.
1	Demo Disk	IBM PC Compatible diskette with sample QED-Forth programs and QEDTerm Program.
1	Doc-1	QED Hardware Manual. The QED-Flash Board
1	MAN-HC11	Motorola 68HC11 Reference. Motorola 68HC11F1 Reference.

The separately purchased Control C compiler package includes the following items:

Getting Started with the QED Board Using the Control C Programming Language.
QED Control C Glossary.
QED Control C compiler diskettes.
Compact WinEdit manual.

If the Control C compiler package was not purchased, the starter kit includes:

Getting Started with the QED Board Using the Forth Programming Language.
QED-Forth Glossary Version 3.1
QED Software Manual

If you are missing any of these items, please contact Mosaic Industries, Inc. at (510) 790-1255.

Overview of the SmarTouch Controller Hardware

The brains behind the SmarTouch Controller is the QED-Flash Board. This double sided surface mount board comes with 256K of flash memory and 128K of battery backed RAM. For information about the dip switch settings and the memory map see "Update Notice: The QED-Flash Board (QED-4 Board)".

Figures 1 and 2 show the key components of the SmarTouch Controller and the QED-Flash Board.

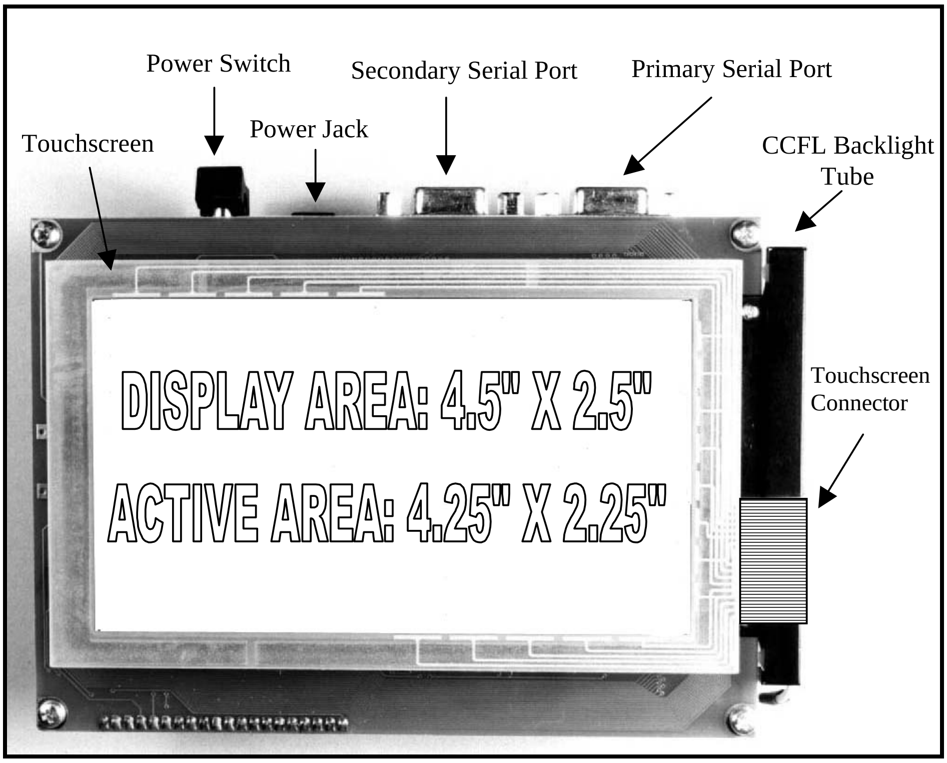


Figure 1: Top view of the SmarTouch Controller

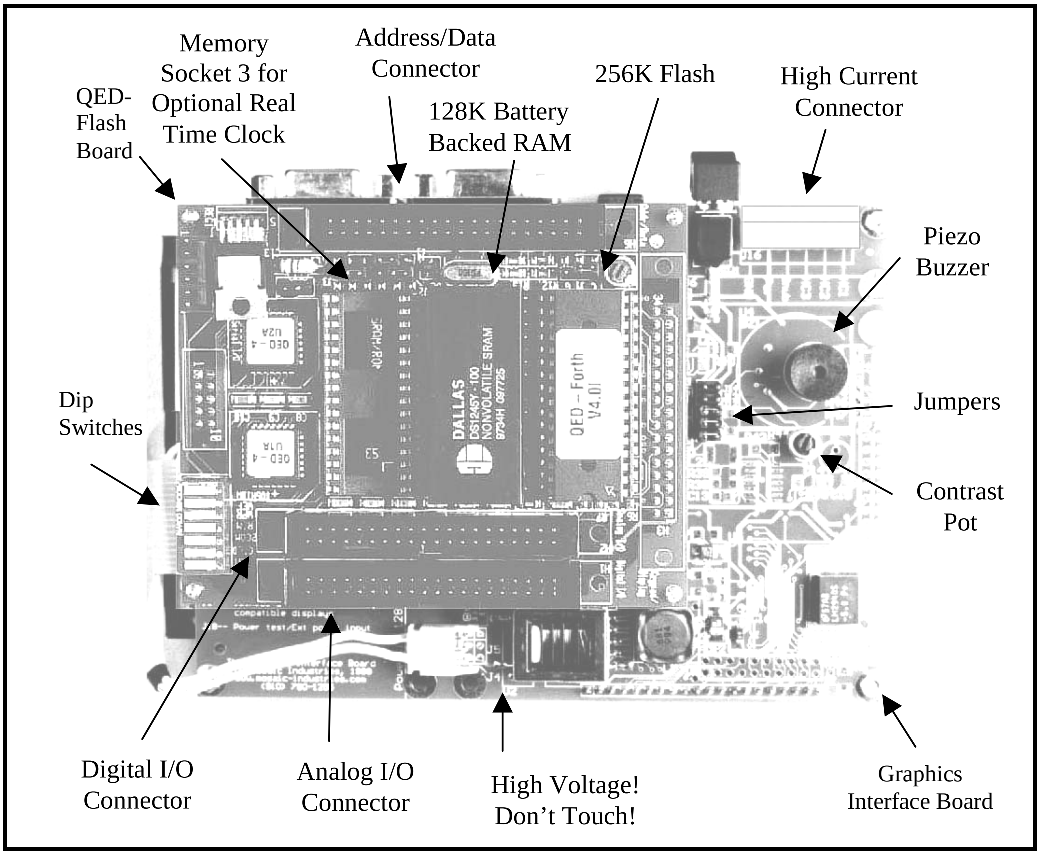


Figure 2: Bottom view of the SmarTouch Controller

The important components of the SmarTouch Controller are the primary and secondary serial ports, power jack, power switch, high current connector, piezoelectric buzzer, contrast pot, and jumpers shown in Figures 1 and 2. All of these components are mounted on the Graphics Interface Board (GIB) which mounts behind the 128x240 pixel graphics display. The primary serial port can be configured for RS-232 communication up to 19,200 baud. Its 9-pin connector is located behind the upper right hand corner of the display as shown in Figure 1. The secondary serial port, which can be used for peripheral devices such as a printer or a barcode reader, has a maximum baud rate of 2,400. Its 9-pin connector is located between the primary serial port and the power jack. To enable the secondary serial port, turn dip switch 5 on the QED-Flash Board ON. In the default configuration, dip switch 5 is OFF. For more information about serial communications see Chapter 11 of the QED Hardware Manual.

The power jack is located between the secondary serial port and the power switch. The SmarTouch Controller can be powered by any power supply that can deliver 6-12VDC @ 500 mA. The power switch, located between the power jack and the high current connector, controls power to the SmarTouch Controller. The switch is in the "off" position when it is depressed away from the GIB.

The piezoelectric buzzer allows audio feedback for software events such as button presses or alarms. As shown in Figure 2, the buzzer is located between the high current connector and the contrast pot.

The contrast potentiometer (pot) adjusts the contrast of the display and is located below the piezoelectric buzzer. The contrast of the display is properly set before each unit is shipped. If further adjustment is required, simply turn the pot with a small screwdriver. Jumpers at J12, located between the QED-Flash Board and the piezoelectric buzzer, enable the onboard contrast pot. If the contrast pot needs to be mounted in a different location (on an external panel, for example), these three jumpers can be removed and an external pot can be wired via a simple cable to the jumper pins. Jumpers J13 and J14, located above the jumpers at J12, are used to override the power switch. If either of the jumpers is installed, the SmarTouch Controller is always on regardless of the switch position.

The SmarTouch Controller has four MOSFET high current drivers named HC0-HC3. The high current screw-terminal connector shown in Figure 2 provides a convenient way to connect to the high current lines (HC1 – HC3) and their onboard snubber diodes, V+ raw power, and ground of the SmarTouch Controller. The connector is located in the upper right hand corner of the GIB next to the power switch as shown in Figure 2. The legend on the GIB identifies the signals on the 6-pin screw terminal connector. High current driver HC0 is not brought out to the connector because it controls the onboard piezoelectric buzzer. Thus, the onboard snubber diodes cannot be connected to a voltage lower than 5 volts or the piezoelectric buzzer will be continually on. High current driver HC1 controls the CCFL backlight. For more information on how to use the high current drivers or how to disconnect the backlight from HC1, see Appendix A.

We have included a working software demo in each SmarTouch Controller. It automatically starts when power is first applied and the power switch is turned on. The demo displays a row of buttons across the top of the display. Pressing on a button brings up a graphical control screen. This is a good way to verify that all of the hardware is working properly and also demonstrates some basic capabilities of the menu control software.

Establishing Communications with the SmarTouch Controller

Now that you know where your serial and power connectors are, follow these simple steps to set up your system:

1. Examine the settings of the 7 onboard switches**.** Dip switches 1, 5, 6, and 7 should be turned OFF by depressing the side of the switch towards the edge of the board**.** Dip switches 2, 3, and 4 should be turned ON. The function of each dip switch is described in “Update Notice: The QED-Flash Board (QED-4)”**.**

2. Connect the female end of the 9-Pin serial communications cable to your computer terminal's serial communications port (typically "COM2") and the male end to the primary serial port on the SmarTouch Controller (see Figure 1)**.**

3. Power up your PC computer or terminal**.**

4. Install the QEDTerm Terminal Program. If you are using an IBM compatible PC that runs Windows, we recommend that you use the "QEDTerm" program that comes on the QED Demo and Driver disk or downloaded from our website: http://www.mosaic-industries.com.

To install QEDTerm, simply copy the files QEDTerm.exe, QEDTerm.ini, and QEDTerm.hlp from `A:\qedterm` to your working directory (`C:\work` for example) on your computer.

The icon named "QEDTerm.exe" opens the terminal program and configures it for QED communications. Double clicking on this icon starts the terminal using the configuration file called QEDTerm.ini; this configures communications with the QED Board using COM2 at 9600 baud, no parity, and 1 stop bit**.** Xon/Xoff flow control is enabled, and the file transfer options are set so that the terminal waits for a linefeed (^J) character before sending each line (this is very important)**.**

Configuring QEDTerm is explained in detail in Appendix A of “Getting Started with the QED Board Using the Control C Programming Language” or “Getting Started with the QED Board Using the Forth Programming Language”. Please consult this section if your PC has non-standard interrupt settings or if COM2 has been "tied up" by other hardware such as a fax/modem that uses the serial port.

5. Plug the wall transformer into a 110 VAC outlet**.** Insert the 6 volt DC wall transformer's plug into the power jack on the SmarTouch Controller and turn the power switch ON by depressing the side of the switch towards the edge of the Graphics Interface Board. The demonstration program should appear on the screen. Press one of the buttons at the top of the display to see sample control screens. The demo does not control any digital or analog I/O.

6. Disable the demo software that is installed in the SmarTouch Controller by turning ON dip switch 6, turning ON dip switch 7, turning OFF dip switch 7, and finally turning OFF dip switch 6. Once the demo is disabled, it can not be restarted. This “special cleanup” procedure places the SmarTouch Controller in a pristine state with no application program running. For more information see Chapter 4 in “Getting Started with the QED Board Using the Control C Programming Language” or Chapter 11 in the QED Software Manual.

7. Each time the SmarTouch Controller is turned on or reset, the following message should appear on your QEDTerm screen:

   ```
   QED-Forth V4.xx
   ```

   where "`V4.xx`" represents the current software version number of your QED Board. If no message appears:

   - Verify that power is being properly applied to the QED Board.
   - Verify that the serial cable is properly connected.
   - Follow the detailed terminal configuration directions in Appendix A of “Getting Started with the QED Board Using the Control C Programming Language” or “Getting Started with the QED Board Using the Forth Programming Language”.
   - Call Mosaic Industries if you are still having trouble.

With your SmarTouch Controller powered up and your terminal communicating with it, an "`ok`" message appears on the screen each time you press the carriage return**.** This message indicates that the onboard QED-Forth operating system is ready to receive commands**.** Now you are ready to install your compiler (if you are programming in C) and compile a program. If you are programming in Forth, skip to the section titled “Programming the SmarTouch Controller using QED-Forth”.

Installing Your Control C Compiler

This section describes how to run the installation program that sets up the QED Control C compiler and the WinEdit editor on your IBM-compatible. The next section describes how to start compiling and running programs.

The C Development Software runs on any PC-compatible computer based on a 386, 486, or Pentium processor and running the Microsoft Windows 3.1, Windows 95, Windows 98, or Windows NT operating system. The C compiler is distributed on 3.5" high density diskette media.

The Windows-based installation program is very easy to use. After closing any running applications (word processors, etc.), insert the QED Control C Disk 1 into your floppy disk drive (`A:` or `B:`).

If you have Windows 3.1, choose "`Run`" under the "`File`" menu in either the Windows Program Manager or the Windows File Manager. If you have Windows 95, Windows 98, or Windows NT, click the "`Start`" button and select the "`RUN...`" menu item. Now type the command:

```
A:\INSTALL
```

where `A:` represents the disk drive where you inserted the diskette. Press "`OK`" to start the installation program.

Then follow the instructions that appear on your computer screen to install the compiler and WinEdit software, pressing the "`Next`" button to continue the installation process. The software must be installed onto a hard disk. The installer program will tell you how much disk space is required, and how much space is available on your hard drive. The installation typically requires between 4 and 7 Megabytes, depending on your system configuration.

The installation process takes just a few minutes. You can abort the installation at any time by pressing the "`Cancel`" button. The installer program reminds you to close any running applications before starting, presents the Fabius License Agreement, checks to see if there is a prior version of the Fabius compiler on the hard drive, and then asks for your name, your company name, and your Serial Number. Type in the Serial Number printed on the installation diskette. Then the installer asks you to specify the name of the directory on your hard drive where the QED Control C compiler will be stored on your hard drive. The default directory name is

```
\FABIUS
```

which is the name of the company that created the compiler. We recommend that you accept the default directory as shown. You can specify any directory name that you want, but things will be much simpler for you if you stick with the default name for now. Using the default 'FABIUS' directory name allows you to take advantage of all of the pre-coded batch files that automate compilation without making any changes. To change the default directory name or location, see Chapter 4 in “Getting Started with the QED Board Using the Control C Programming Language”.

During installation, you will be prompted when to insert another specified diskette; press "`OK`" to continue after swapping in the appropriately numbered installation diskette. At the end of the installation process, the program will modify some system files including your `AUTOEXEC.BAT` file so that the new compiler can be easily found by the operating system. It will create some Windows icons that are discussed further below. The last screen asks you to press the "`Finish`" button to complete the installation process; this automatically re-boots your computer to put the new compiler PATH into effect.

The next section guides you through compiling, transferring, and running a simple program.

Programming the SmarTouch Controller Using Control C

In the previous sections we learned about the SmarTouch Controller's hardware, established serial communications, and installed the C compiler and WinEdit development environment on the PC. Now we'll see how easy it is to compile, download and execute a C program. In this section we'll be running a simple application that writes a few lines of text to the display and scans the touchscreen looking for any touches.

Let's Compile a C Program

The source code for the example program is in the `S_TOUCH.C` file. This file is located in the root directory (`A:\`) on the Demo and Driver Disk that is shipped with each SmarTouch Controller. Here is a brief summary of how to compile the program:

1. Copy the source code file, `S_TOUCH.C`, into your working directory on your PC. At a dos prompt type: `copy A:\S_TOUCH.C C:\work\` to copy `S_TOUCH.C` to the `work` directory. Or if you are using Windows Explorer, simply drag the icon named `S_TOUCH.C` on to the `work` directory.
2. Open the WinEdit editor by double-clicking on its icon. If WinEdit is already active, you can also enter it by clicking in its window or by holding down the Alt key and pressing the Tab key until you see the WinEdit announcement.
3. Choose "`Open`" from the WinEdit "`File`" menu, and open the file named `S_TOUCH.C` in the `A:\` directory.
4. Click on the editor's "`MAKE`" icon (the hammer) to compile the source code file and automatically generate a download file named `S_TOUCH.TXT` ready to be sent to the SmarTouch Controller. (For experts - This is equivalent to typing `CC S_TOUCH.C` at the DOS prompt).
5. When the compilation is done, a dialog box says "Compile Complete" and asks you whether you want to "Analyze Results" (that is, step through any error or warning messages, highlighting the corresponding source code line), or "Load Output File" (that is, view the file containing messages generated during compilation), or "Cancel". We know that there are no compilation errors, so it's OK to press "Cancel" now. You can always go back later and analyze the error messages or view the compiler output file if you wish.

Download a C Program to the SmarTouch Controller

Now that the program is compiled, you can download it to the SmarTouch Controller. Transferring the compiled program to the SmarTouch Controller will overwrite the demo program.

1. Enter the "QEDTerm" program by double clicking on the "QEDTerm" icon. Or, if QEDTerm is already active, enter it by clicking in its window or by holding down the Alt key and pressing Tab until you see the QEDTerm announcement.

2. Choose "Send File" from the "File" menu, change the directory to `C:\QED` by clicking on the appropriate directories, and double click on the

   ```
   S_TOUCH.TXT
   ```

   filename to transfer the download file to the SmarTouch Controller.

   You will see a hexadecimal download in Motorola S2-record format scroll across your terminal window. At the end of the download you will see some brief Forth function definitions; these are telling the Forth compiler the names of the individual functions and variables so that we can interactively execute the functions in the `S_TOUCH.C` file. For more information on using the interactive debugger see Chapter 5 in “Getting Started with the QED Board Using the Control C Programming Language”.

Run the Program

To run the program, simply type

```
main
```

from the terminal window. Several things should happen:

1. You should see the message:

   ```
   Hello world!
   ```

   printed in your terminal window.

2. You should see the following message on your display:

   ```
   Welcome!  Touch the
   screen to see how the
   display & touchscreen
   work together
   ```

3. If you press any location on the touchscreen, you should see the message

   ```
   I'd rather be...
   ```

on the first line, followed by a message on the second line that varies depending on the column of the touchscreen that you choose. Try it out. To terminate the test, push the left-most column of the touchscreen; you'll see the message

```
I'd rather be...
Done with this test.
```

Now if you type carriage returns from your terminal, the QED-Forth monitor will respond with the "ok" prompt, meaning that it is ready for the next command.

You can type:

```
main
```

from your terminal any time you like, and the main function as defined in the latest download file will be executed.

You have now compiled, transferred, and executed a program on the SmarTouch Controller! For a more extensive example, see Chapter 4 in “Getting Started with the QED Board Using the Control C Programming Language”.

Programming the SmarTouch Controller Using QED-Forth

In the previous sections we leaned how to install the C compiler and compile, download, and execute a C program. Now we show you how to download and execute a Forth program that writes a few lines of text to the display and scans the touchscreen looking for any touches.

The SmarTouch Controller accepts serial commands from a terminal which is typically a personal computer (PC). QED-Forth’s onboard interpreter, compiler, and assembler compiles your program into executable code in the SmarTouch Controller’s memory, and the code can be immediately executed by simply stating the name of a routine. This encourages a productive iterative programming style.

If you wanted, you could program a SmarTouch Controller application by typing in your application program one line at a time from a terminal. For all but the shortest programs, this would be a tedious process. Fortunately, there is a better way. All you need is a text editor program and a terminal emulation program that runs on your PC. The text editor allows you to create and modify text files and save them on the computer’s disk drive. The terminal program allows you to send and receive characters via the PC’s RS232 serial port to communicate with the SmarTouch Controller. Thus you can rapidly edit your source code into files which can be downloaded to the SmarTouch Controller. Ideally, you should set up your PC so that you can switch between the editor and terminal programs quickly and easily.

A simple program, named “S_TOUCH.4TH”, demonstrates how to use the touchscreen and the display. This program is available on the Demo Disk that is included with each SmarTouch Controller.

Download a Forth Program to the SmarTouch Controller

To download the sample Forth program to the SmarTouch Controller:

1. Enter the "QEDTerm" program by double clicking on the "QEDTerm" icon. Or, if QEDTerm is already active, enter it by clicking in its window or by holding down the Alt key and pressing Tab until you see the QEDTerm announcement.

2. Insert the Demo Disk sent with the SmarTouch Controller and choose "Send File" from the "File" menu. Change the directory to `A:\` by clicking on the appropriate directories, and double click on the

   ```
   S_TOUCH.4TH
   ```

   filename to transfer the file to the SmarTouch Controller.

   You will see the lines of the program scroll across your terminal window. If any errors occur, the terminal program will automatically terminate the transfer process. No errors should occur when transferring this file.

Run the Program

To run the program, simply type

```
hello
```

from the terminal window. Several things should happen:

3. You should see the message:

   ```
   Hello world!
   ```

   printed in your terminal window.

4. You should see the following message on your display:

   ```
   Welcome!  Touch the
   screen to see how the
   display & touchscreen
   work together
   ```

3. If you press any location on the touchscreen, you should see the message

   ```
   I'd rather be...
   ```

on the first line, followed by a message on the second line that varies depending on the column of the touchscreen that you choose. Try it out. To terminate the test, push the left-most column of the touchscreen; you'll see the message

```
I'd rather be...
Done with this test.
```

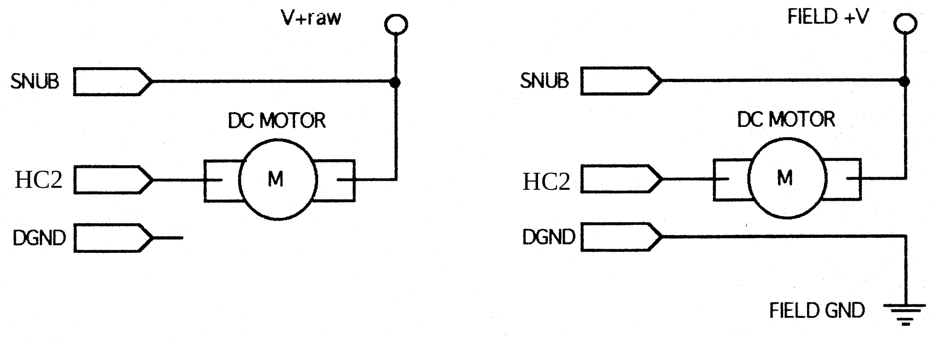
Now if you type carriage returns from your terminal, the QED-Forth monitor will respond with the "ok" prompt, meaning that it is ready for the next command.

You have now compiled, transferred, and executed a program on the SmarTouch Controller! For a more extensive example, see Chapter 2 in “Getting Started with the QED Board Using the Forth Programming Language”.

Appendix A: High Current Drivers

The SmarTouch Controller has four MOSFET high current drivers. Each driver can sink up to 150 mA continuously, or up to 1 Amp on an intermittent basis at voltages as high as 60 V. Onboard snubber diodes suppress inductive kickback, so the drivers can be used to control solenoids, motors and other inductive loads. High current driver HC0 controls the onboard piezoelectric buzzer and high current driver HC1 controls the CCFL backlight. HC1 controls a relay whose contacts control the backlight. The contacts can be shorted by installing a 0 Ohm resister at R29 on the GIB, permanently turning on the backlight. If this is done, HCI can be used for other purposes. Keep in mind that HC1 will still actuate the relay.

Figure 3 shows how to connect a DC load (a DC motor is shown) to the MOSFETs.



Connecting a load using onboard power.

Connecting a load using external power.

Figure 3: How to Connect a DC Load to the MOSFETs

The routines SET.HIGH.CURRENT and CLEAR.HIGH.CURRENT (SetHighCurrent() and ClearHighCurrent() in C) are used to control high-current drivers. SET.HIGH.CURRENT (SetHighCurrent() in C) accepts a 4-bit mask and turns on the drivers that correspond to "1"s in the mask. CLEAR.HIGH.CURRENT (ClearHighCurrent() in C) accepts a 4-bit mask and turns off the drivers that correspond to "1"s in the mask. Note that "turning on" a driver allows it to sink current, thereby pulling the voltage at the output pin towards ground. "Turning off" a driver prevents it from sinking current. For example, to turn on all four high current drivers, execute

```
SetHighCurrent( 0x0F )                                 (C)
HEX   0F  SET.HIGH.CURRENT                             (Forth)
```

Then, if you execute

```
ClearHighCurrent( 0x01 )                               (C)
01  CLEAR.HIGH.CURRENT                                 (Forth)
```

you will turn off driver number 0 while leaving the state of the other three drivers unchanged.

Appendix B: Guide to the QED Manuals

The following guide will help you to select the correct manual for the information that you need.

Getting Started with the QED Board Using the Control C Programming Language
This manual combines an introduction to the QED Board with a detailed software manual for Control C programmers. It provides an overview of the QED Board, Product Design Kit (PDK), and the Industrial Control System (ICS) hardware, includes information about installing the C compiler, programming the QED Board in C, using the interactive debugger, introduction to the keypad and display, introduction to the serial communication ports, using the analog converters, review of arrays and EEPROM, digital I/O, high current drivers, timekeeping and interrupts, a turnkeyed application program example, and QEDTerm configurations.

Getting Started with the QED Board Using the Forth Programming Language
This is a brief introduction to the QED Board for Forth programmers. It includes information about QED, PDK, and ICS hardware, programming with QED-Forth, introduction to the serial ports, using the analog to digital converter, using the digital to analog converter, and QEDTerm configurations.

QED Hardware Manual
This book provides comprehensive documentation on all of the hardware aspects of the QED Board including the general purpose digital I/O, interrupts, programmable timer and pulse accumulator, serial peripheral interface (SPI), A/D and D/A converters, EEPROM, external interrupts, resets, operating modes, COP, memory mapped I/O, keypad and LCD display, serial communications, battery-backed real-time clock, powering the QED Board, a trouble shooting guide, QED connector pinouts, QED Board schematics, 68HC11F1 Hardware Registers, specifications of the QED Board and Product Design Kit, and data sheets for key onboard components.

QED Software Manual
This book provides comprehensive documentation on all of the software aspects of the QED Board in Forth including program development techniques, floating point mathematics, the heap memory manager, array, matrices, fast vector operations, solving simultaneous equations, curve fitting, fast Fourier transforms, structures, designing re-entrant code, multitasking, interrupts, register initializations, the LCD display, the keypad, the assembler, mass memory interface, a tour

of the users area, advanced topics, a turnkeyed application program example, the QED Board memory map, error handling, and the header format for dictionary names.

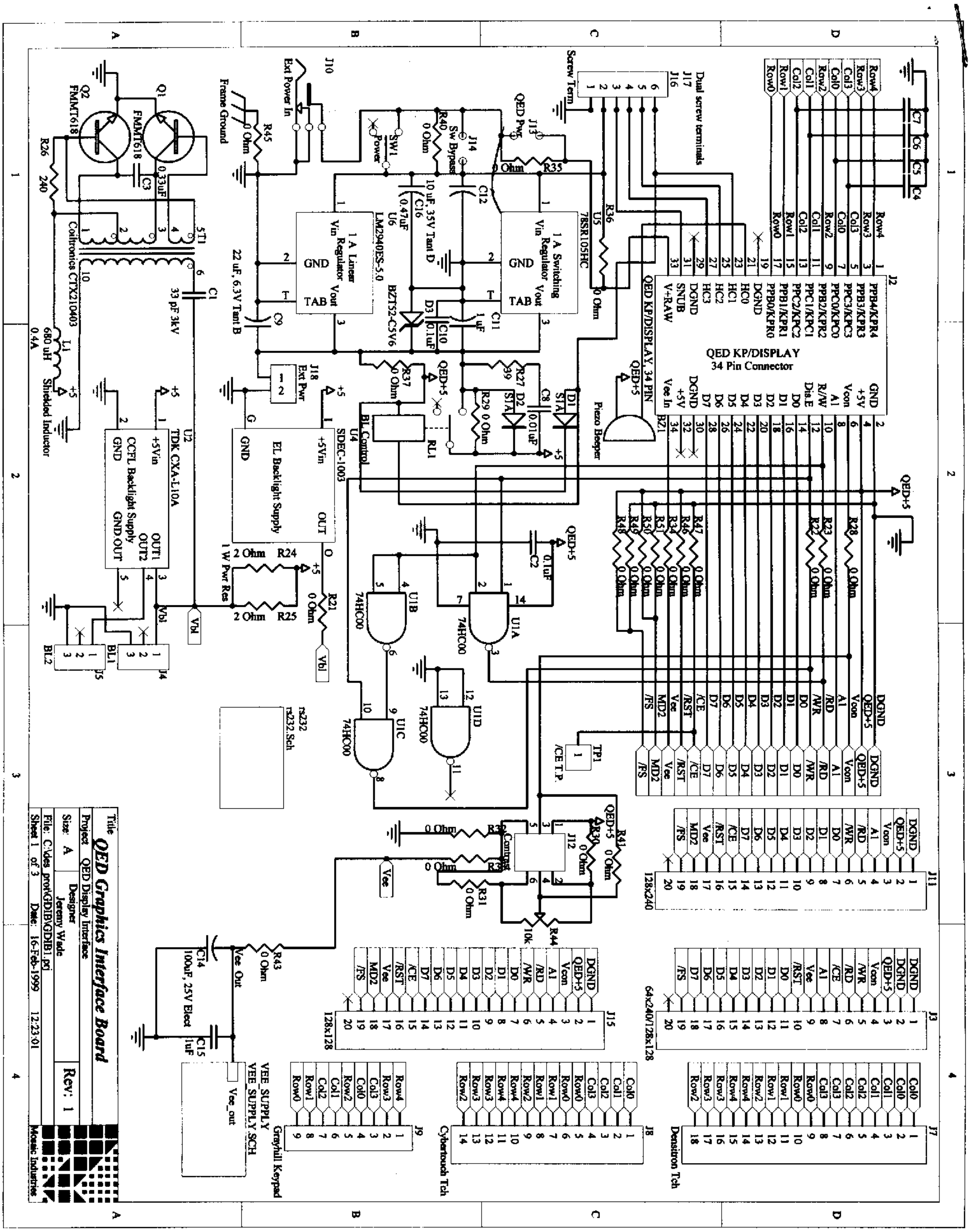
QED Control C Glossary
This comprehensive glossary of all Control C library functions includes: analog I/O functions, array functions, communications functions, heap functions, interrupt functions, multitasker functions, and extended memory functions for QED-2 and QED-3 Boards.

QED-Forth Glossary
This comprehensive glossary of all QED-Forth functions includes: array routines, control structures, device drivers, floating point routines, interrupt routines, math routines, matrix math routines, memory routines, multitasking and time-keeping routines, numeric conversion routines, stack operations, structures, and the assembly glossary for QED-2 and QED-3 Boards.

The QED-3 Board Hardware and Software Update Documentation
Documents the hardware and software differences between the QED-2 Board and the QED-3 Board including information about 16MHz operation, the new display interface and support for graphics displays, the keypad and touchscreen interface, high current drivers, A/D conversion, and multidrop RS-232.

Update Notice: The QED-Flash Board (QED-4 Board)
Documents the hardware and software differences between QED-3 Board and the QED-Flash Board including information about flash memory, dip switches, and memory map changes.

Appendix C: Schematic of the Graphics Interface Board


QED Graphics Interface Board
Project QED Display Interface
Designer Jeremy Wade
Size A
Rev: 1
Sheet 1 of 3
Date: 16-Feb-1999 12:23:01
Mosaic Industries
QED KP/DISPLAY 34 Pin Connector
QED KP/DISPLAY 34 PIN
1 A Switching Regulator
78SR105HC
1 A Linear Regulator
LM2940ES-5.0
EL Backlight Supply
SDEC-1003
CCFL Backlight Supply
TDK CXA-L10A
Coiltronics CTX210403
FMMT618
74HC00
Piezo Beeper
Dual screw terminals
Ext Power In
Frame Ground
rs232
rs232.Sch
VEE_SUPPLY
VEE_SUPPLY.SCH
128x240
64x240/128x128
128x128
Grayhill Keypad
Cybertouch Tch
Densitron Tch