User's Manual

VuePoint™ III Touch Terminal

Manual P/N: 01-016
Revision 1.3
December 2005

GENERAL DIGITAL
The Innovators of Flat Panel Technology

Please take a moment...

...to record the following information in the space provided below. Should the need arise to contact General Digital™ regarding your LCD monitor, this information will help us to better serve you. You will find the model number/serial number label affixed to the rear of your monitor.

Model Number: _____

Serial Number: _____

Date Received: _____

Trademarks

All trademarks in this publication are the property of their respective companies.

Copyright

Disclaimer

General Digital™ Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Appropriate Usage Policy

General Digital™ Corporation components should not be used in medical devices with life support functions, safety equipment (or similar applications where component failure would result in loss of life or physical harm), aerospace equipment, telecommunication equipment (trunk lines) or nuclear power control equipment. Contact a General Digital™ representative before using General Digital™ devices for any applications other than those recommended by General Digital™. General Digital™ assumes no responsibility for any damage resulting from the use of a General Digital™ device which does not comply with the instructions and precautions specified in this manual.

Table of Contents

Illustrations

Listings

Tables

This manual has been designed to introduce you to General Digital's™ VuePoint™ III Touch Input Display Terminal. If you'll be installing or maintaining the VuePoint™ III, this manual is for you.

For first-time VuePoint™ III users in particular, it is important that you familiarize yourself with the material in Section 2, Getting Started, before attempting to unpack and install your terminal. Though it's not difficult to do, you'll save yourself a lot of time and effort if you become familiar with the process first.

How to Use this Manual

This manual is divided into nine major sections plus six appendices, which will provide you with a step-by-step tutorial on the installation, operation and applications of the terminal.

Section 1, *Introduction to the VuePoint™ III*, provides you with an overview of the terminal's characteristics and how they can be applied to meet specific application needs.

Section 2, *Getting Started*, guides you through the unpacking, installation and an initial quick test of the unit.

Section 3, *VuePoint™ III Text Display*, covers the basics of using VuePoint™ III to display alphanumeric text messages. Specific examples, using either another terminal or a host computer to communicate with the VuePoint™ III, are included to guide your learning process.

Section 4, *VuePoint™ III Touch Input*, uses the same approach to show how touch input operates and how you can use it for computer input.

Section 5, *Saving Your Displays*, discusses the concepts behind the VuePoint™ III's *page memory* and how it can save you a lot of time and effort when implementing an application.

Section 6, *Advanced Display Features*, as the name implies, covers some of the more advanced display capabilities of the VuePoint™ III, such as *variable character intensity*, *alternate character sets*, and *character blink*.

Section 7, *Advanced Touch Features*, describes the use of additional touch response modes, as well as *touch sensitivity*.

Section 8, *Advanced Paging Features*, introduces you to some of the more sophisticated–and useful–page memory operation of the VuePoint™ III. *Page copying* and *merging* are among the items covered.

Section 9, *Other VuePoint™ III Commands*, describes the operation of system level commands such as *Bell Control*, *Power Fail Status*, and *Power On Reset*.

Following these major sections are the appendices, which include the *VuePoint™ III Specifications*, an *ASCII Character Code Chart*, the *Factory Settings*, *Additional Resources*, a *Mechanical Drawing*, and a *Command Summary*.

The Equipment You Need

If you're like most people, the "learn by doing" approach used throughout this manual will be the most efficient way for you to learn how to use your VuePoint™. This requires that you have a certain amount of equipment available for trying the various examples discussed. At the very least, you'll need a complete VuePoint™ III terminal!

In addition, you'll also need a device to use to communicate with the VuePoint™ III. For the simpler operations, this can be almost any RS-232C-compatible computer terminal. The more sophisticated operations, however, are best illustrated with a host computer system.

Translating these generalities into part numbers, you will need the following to actually perform the examples in this manual:

A. VuePoint™ III Terminal

At a minimum, you will need a terminal configured as shown below:

- 90-952-001 VuePoint™ III Touch Terminal - Standard Configuration, or
- 90-952-*xxx* VuePoint™ III Touch Terminal - Custom Configuration (where *xxx* designates the custom number that is assigned at order placement)

B. Computer Terminal

If you'll be using a computer terminal as the device to communicate with the VuePoint™ III when trying the examples, most any brand of RS-232C-compatible terminal can be used.

Since one specific terminal had to be chosen to use in developing the examples for this manual, a Digital Equipment Corporation VT-100 was selected.

C. Host Computer

Essentially any computer that includes an RS-232C communications interface and a BASIC interpreter in its software library can be used to run the computer-oriented examples in this manual. Due to the fact that IBM PCs and PC-compatible machines are readily available, that is the computer selected for the examples outlined herein. All you need to run the examples in this manual on an IBM PC is a system with a single 3.5" diskette drive, a serial I/O port, the DOS and BASICA software and the minimum memory required to run those packages.

D. Interconnecting Cables

Since there are so many variations of RS-232 around today, there is no one universal cable that can be used in all cases. However, if you are going to be using the same configuration in this manual, then you will need a *null modem* cable with a 25-pin plug on one end to connect to the VuePoint™ III and the appropriate socket (9 pin or 25 pin) at the other end to connect to your computer terminal or host computer. We'll see just how and why the null modem is used in a later section.

Once you've gathered together the bits and pieces you need, you're all set to go. After covering some introductory material in Section 1, Section 2 will show you how to install the equipment and start using it.

Installation

General Digital™ recommends that you install or leave the VuePoint™ III terminal:

- In a place free of extreme temperatures (i.e., away from radiators, heating vents, direct sunlight).
- In a place free of mechanical vibration and shock.
- In a place far from any equipment that generates a strong magnetic field.
- In a place free of inordinate amounts of dust, dirt or sand (i.e., an open window or outdoor exit). If setting up temporarily in an outdoor environment, be sure to take adequate precautions against airborne dust and dirt.
- On a stable surface where the terminal will be safe from tipping over.

Additional recommendations:

- Ensure that vents are kept free of obstruction and airflow remains unrestricted at all times.
- If the unit is equipped with a grounded AC line input connector, ensure that a reliable grounding path is maintained. Units so equipped are intended to be connected to ground.
- Ensure that the unit is located where the temperature remains within the recommended ambient temperature range listed in the VuePoint™ III Specifications, Appendix A.

Safety Considerations for Rack Mount Units

- Ensure that the unit is located where the temperature remains below the maximum recommended ambient temperature listed in the VuePoint™ III Specifications, Appendix A. Internal temperatures of the rack should be considered for continued safe operation.
- Ensure that power supply vents are kept free of obstruction and airflow remains unrestricted at all times.
- Mechanical loading of rack should be considered so that the rack remains stable and unable to tip over.
- Consideration of the overall loading of the branch circuit should be given before installing any equipment in a rack environment.
- Ensure that a reliable grounding path is maintained in the rack system. This unit is intended to be connected to earth ground.

Characteristics of the LCD Screen

- Keep the LCD from facing direct sunlight for extended periods as this can damage the screen.
- The LCD screen may lose uniformity or otherwise malfunction if pushed, scratched or has a heavy object set upon it.
- If the terminal is used in a cold place, a residual image may appear on the screen. This is not a malfunction. The screen returns to normal as the temperature rises to a normal operating level.
- If a still image or text is displayed for a long time, a residual image may appear for a while. The residual image will eventually disappear.
- The LCD panel becomes warm during operation. This is not a malfunction.

Important Notes on the LCD Screen Characteristics
Please be aware that, although the LCD screen is made with high precision technology, the manufacturer's specifications allow for slight optical anomalies. If your LCD screen exhibits any minor anomalies, it is not an indication of a malfunction. General Digital™ inspects each screen to even more rigid specifications than the manufacturer's and will not ship any product found to be substandard.

Also, optimal performance is achieved after the monitor has reached a stable internal operating temperature. At ambient room temperature (72° F/22° C), this may require as long as 30–60 minutes.

Replacement of the Fluorescent Tube

A specially designed fluorescent tube is installed as the lighting apparatus for the VuePoint™ III monitor. If the screen becomes dark, unstable or does not turn on, replace the fluorescent tube with a new one (General Digital™ Part Number 12-205). Consult General Digital™ when replacing the fluorescent tube.

Maintenance

- Be sure to unplug the power cord from the power outlet before cleaning your terminal.
- Clean the LCD screen with a soft cloth. If you use a glass cleaning liquid, refrain from using any type of cleaner containing an antistatic solution or similar additive as this may scratch the LCD's screen coating.
- Clean the cabinet, panel and controls with a soft cloth lightly moistened with a mild detergent solution. Do not use any type of abrasive pad, scouring powder or solvent, such as alcohol or benzine.
- A scratched LCD surface may result if the screen surface is rubbed, touched or tapped with sharp or abrasive items such as ballpoint pens or screwdrivers.
- Note that material deterioration or LCD screen coating degradation may occur if the terminal is exposed to volatile solvents such as insecticide, or if prolonged contact is maintained with rubber or vinyl materials.

Transportation

- Disconnect all cables from the terminal and grasp the support and base sections of the display stand firmly with both hands when transporting. If you drop the terminal, you may be injured or the terminal may be damaged.
- When you transport this terminal for repair or shipment, use the original carton and packing materials. Failure to do so may void the warranty.

Usage Recommendations

- For each hour of looking at the terminal, let your eyes rest for 5 minutes. This will reduce eye fatigue.

Disposal of the Terminal

The fluorescent tube used in this terminal contains mercury. Disposal of this terminal must be carried out in accordance with the regulations of your local sanitation authority. Do not dispose of this terminal with general household waste.

1.0 Introduction to the VuePoint™ III 13

In this section of the manual, we'll be discussing the general characteristics of the VuePoint™ III and how they can be used to fulfill some specific application requirements.

1.1 The VuePoint™ III Terminal

The VuePoint™ III, shown in Figure 1-1 below, is the third generation in General Digital's™ family of flat panel touch input display terminals. By using a rugged LCD display panel rather than the more traditional–and fragile–cathode ray tube (CRT) technology, the VuePoint™ III gives you a bright 12-line text display in a compact environmentally isolated package.

In addition to its ability to display messages to the operator, the VuePoint™ III also gives you one of the most natural and instinctive human interfaces available today–touch input.

To respond to a message seen on the display, the operator need only do what comes naturally: point to the desired response. VuePoint™ III will sense both the presence and location of the touch and send that information to your computer. Since your program will "know" where the answers are located on the display panel, converting the touch data to an answer couldn't be easier.

Figure 1-1
The VuePoint™ III Terminal

In addition to text display and touch input, features that are standard on every VuePoint™ III, you can increase its capabilities further through the use of several convenient options.

For example, if you have a lot of different standard displays that you'll be using, General Digital™ can program the VuePoint™ III to hold additional screens (or pages, as they're called) of text right in the terminal itself.

Some applications may require a mixture of touch and keyboard input. Again, all you need is a standard PS/2 keyboard and the VuePoint™ III is ready to handle both types of input for you.

For information on available options for the VuePoint™ III, please consult with a Sales Engineer.

1.2 VuePoint™ III Applications

With unique characteristics such as these, there are a host of applications where you'll find the VuePoint™ III to be the ideal terminal for the job.

Perhaps the most common application of the VuePoint™ III is in the industrial area, where it fits perfectly into a wide range of process and equipment control and monitoring applications.

Its compact, rugged package allows the VuePoint™ III to be added to an existing or new instrument panel more easily than most alternative units. That, combined with its large character, bright display, makes it an excellent annunciator device. For an operator who's used to pushing a button to acknowledge a message or an alarm, what's more natural than touch input in response to the message?

The VuePoint™ III is as easy to incorporate into a system as it is to use. With its universal RS-232 interface, it can be used with devices ranging from mainframe computers to programmable controllers, and everything in between.

Going from one environmental extreme to another, you'll also find the VuePoint™ III being used in business office applications. For example, an on-line stock quotation and trading system has been developed around the VuePoint™ terminal. What's more natural than touch input for a broker that has to juggle three telephones, a cup of coffee and a computer terminal to complete a complex deal? And since the coffee cup **will** get knocked over at some point, the VuePoint's™ environmentally sound construction doesn't hurt either!

For exactly these same reasons, you'll also find the VuePoint™ III built into systems ranging from air traffic control stations to point of sale terminals. Anytime you need a man-machine interface, and typing is the **last** thing you want the operator to worry about, the VuePoint™ III is the terminal to use.

While there's no question that the **man** part of the equation will appreciate the VuePoint™ III, so will the **machine**. Since the VuePoint™ III has its own internal microprocessor, the host computer doesn't have to work very hard to communicate with the terminal. Add to that the fact that the VuePoint™ III can hold many additional pages of screen display, and the host's workload is reduced even further. The result is the ability to communicate a maximum amount of information with a minimum amount of effort on the part of the host computer.

Since everyone–man and machine–is more productive when they work smart, rather than just hard, you'll find the VuePoint™ III to be an efficient, cost-effective solution to a wide range of application needs.

2.0 Getting Started 15

2.1 Unpacking and Inspection

All VuePoint™ III terminals are configured at the factory–with any and all options you may have ordered–installed and tested before the unit is shipped. Because of this, you will probably find fewer separate items in the box than you ordered. Extra memory, extra interfaces, panels, etc. will already be in the main chassis. Only external items, such as power supplies and cables, will be found as "loose" parts.

While you're unpacking the unit, look closely for any shipping damage. If any is found, contact the carrier and General Digital™ immediately. And make sure you save the shipping carton, for it may be needed if you submit a claim for damages. Actually, you should save the carton in any case, for if you ever have to ship the unit, the use of the original carton and packing material will ensure that the unit is properly protected.

You should have the following items sitting in front of you:

VuePoint™ III Terminal
AC Line Cord, 6'7"

To these add the needed communications cable described in the Preface and you'll have all the bits and pieces you need to begin connecting things together. Don't forget the *null modem*, which is a short piece of cable that looks similar to the communications cable. This cable is almost always required to connect any two terminals–or a terminal and a PC–together. We'll see exactly how it's used in Section 2.5.

2.2 Initial VuePoint™ III Setup and Test

For the initial setup and testing of your VuePoint™ III, all you need is the terminal itself and the power cable. Ensuring that the terminal is **not** plugged into an AC power outlet, the following steps should be followed to connect everything together:

1) Verify that the On/Off switch on the power supply is **Off**.

2) Plug the power supply's AC power cord into an AC outlet.

3) Turn the On/Off switch to **On**.

If all is well, the screen of the VuePoint™ III will look as shown in Figure 2-1.

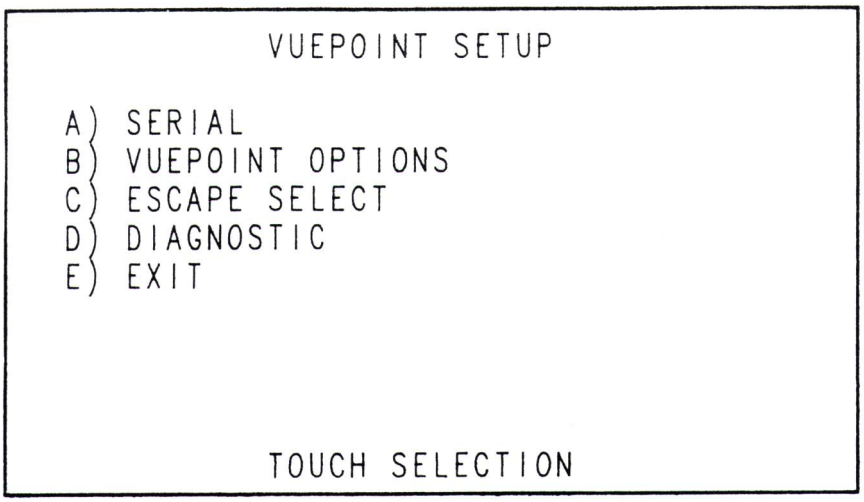


Figure 2-1
Initial Power-Up Display

If the display looks as shown–or close to it, for slightly different messages are displayed for certain of the VuePoint options–you can proceed directly to the initial *Quick Check* test described in Section 2.3.

WARNING!

If you change any parameters in SERIAL, VUEPOINT OPTIONS or ESCAPE SELECT, the settings will not be saved until you touch EXIT on the VUEPOINT SETUP screen (Figure 2-1).

If the FACTORY SETTINGS menu appears instead of the VUEPOINT SETUP, it means the touchscreen must be calibrated. Touch anywhere on the screen and the Calibration screen will appear. Follow the on-screen instructions to calibrate the touchscreen. When finished, touch F) USER OPTIONS to reach the VUEPOINT SETUP screen.

If the display is completely blank, try the following:

1) Turn the power supply **Off** and unplug it from the AC power line.
2) Check the serial number plate on the power supply to verify that you have the correct voltages for the VuePoint™ and AC line voltage you're using. The AC line voltage will be shown as either 120 VAC or 220 VAC, and must agree with what you're using.
3) Remove the fuse and check to see if it has blown. If it has, replace it with **exactly** the same type.
4) Verify that the power cable is properly installed.

Once you've done the above, plug the power supply in and try the unit again. If you still get no response, call the Customer Service Department at General Digital™ for further assistance.

If the terminal looks like it is "alive," (i.e., cursor or screen is illuminated), the most likely cause is that someone got to the unit before you did and changed some of the standard operating parameters. To restore the setup menu, the following procedure is used:

1) Sweep your finger across the top of the screen from right to left, while maintaining steady pressure. This will bring up the FACTORY SETTINGS screen.

2) Touch F) USER OPTIONS.

2.3 VuePoint™ III Quick Test

This section will use some of the built-in diagnostic tests of the VuePoint™ to quickly verify that the major elements of the terminal are working correctly. The procedure is as follows:

1) Assuming the display still looks as shown in Figure 2-1, touch the word DIAGNOSTIC on the display. This activates the built-in test program and generates the display shown in Figure 2-2.

2) Now touch the screen anywhere on the line containing the words STANDARD SET. The standard VuePoint™ character set will be displayed for about 15 seconds, as shown in Figure 2-3, and then the Diagnostic Menu will again be displayed.

3) For the last test, touch the line that contains the ALTERNATE SET choice. The alternate character set will be displayed (more about this later) as shown in Figure 2-4 and then the display will return to the Diagnostic Menu.

In less time than it takes to read about it, you've tested and verified the major internal functions of VuePoint™ III. Touch the word EXIT and you'll be ready to go on with the next section of the manual.

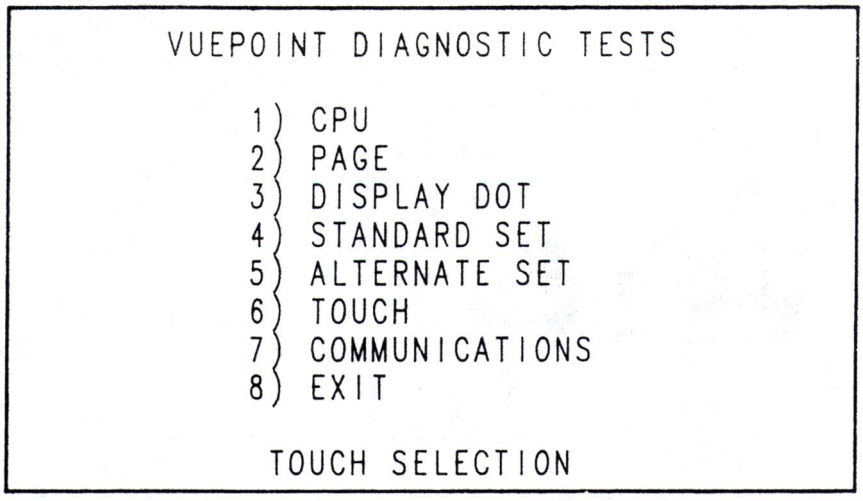


Figure 2-2
Diagnostic Menu Display

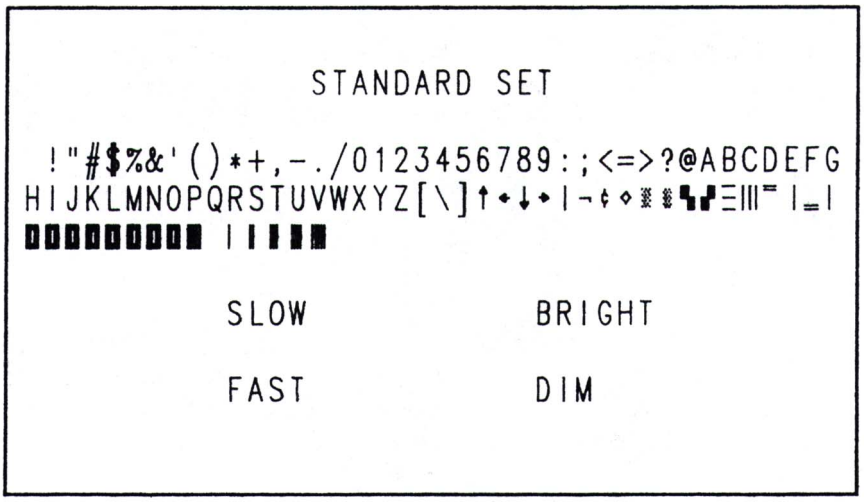


Figure 2-3
Standard Character Set Display

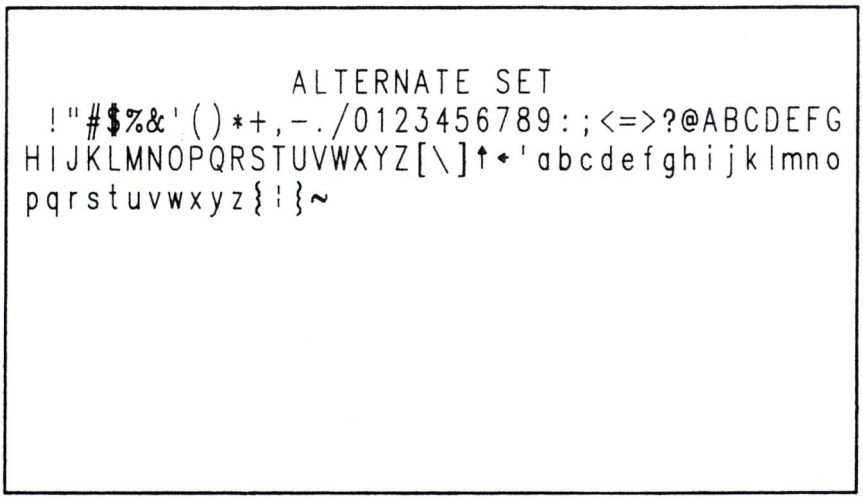


Figure 2-4
Alternate Character Set Display

2.4 Introduction to ASCII Terminals

If you're familiar with how computer terminals operate and communicate with other devices and have successfully connected them together before, you can probably skip this section and go directly to Section 2.5. If not, this section will provide some of the background required to take the installation procedures covered in Section 2.5 out of the realm of "black magic" and into the real world of RS-232 communications.

2.4.1 What is ASCII?

The letters ASCII (pronounced ASKEE, and not ASKEE 2) are an acronym for the American Standard Code for Information Interchange. This is a universally recognized method of encoding and decoding the characters on a keyboard (plus some others we'll discuss later) into a series of numbers that can be easily transmitted between–and understood by–two computers, a computer and a terminal, a computer and a printer, etc.

The exact nature of these ASCII codes (which are detailed in Appendix B) is not important at this time. What is important, however, is that you realize that ASCII means the manner in which characters are encoded and decoded, and has nothing to do with how the information is physically transmitted or received.

This means that **if** you can somehow hook two ASCII devices together, they'll be able to communicate with each other because they are "speaking" the same language. But in no way does the term ASCII guarantee that such a connection can be made. For that we have to introduce another bit of jargon: *RS-232C*.

2.4.2 What is RS-232C?

RS-232C is to the interconnection of two devices as ASCII is to the language those devices speak: a standard way of doing things to insure some degree of compatibility.

Technically, RS-232C is the identification number of one of several communications interconnection standards developed and published by the Electronic Industries Association, or EIA as it's more commonly known. It defines a particular type of connector for use on interconnecting cables, what function each of the various signals that are sent through the cable will perform, the various voltages to be used, and so forth. Everything you need to ensure that two devices are physically compatible and can be readily connected together.

RS-232C is an extremely flexible standard designed to cover just about any machine to machine connection you can imagine. And the rules of RS-232C allow a manufacturer to only use as much of that flexibility as he needs to make his particular machine–printer, terminal or whatever–operate. For this reason, as well as a couple of others we'll discuss shortly, you are almost guaranteed that no two RS-232C devices will be able to start communicating with each other by simply connecting them with a piece of cable.

Fortunately, most terminal, printer and microcomputer manufacturers have reached a tacit agreement on just what parts of the RS-232C standard are needed for these devices. However, this still leaves a fair number of variables for us to examine.

2.4.2.1 Baud Rate

Baud Rate is the technical term for speed: how fast the information will be sent and received by the two machines on each end of the cable. This can range from as slow as 10 characters per second (110 Baud) to several thousand, with most terminals operating in the 120 characters per second (cps) to 960 cps range (1200 to 9600 Baud). While in general the faster the better–assuming you don't send the data so fast that the receiver doesn't have time to process it–the real issue is synchronization: if two devices are ever going to successfully carry on a conversation, they must be set to the same rate.

2.4.2.2 Bits per Character

Each ASCII character can be encoded into either 7 or 8 bits (pieces of information) for sending and receiving. While there are reasons for and against both methods, thew issue again is sameness. As long as both devices use and expect the same method of encoding, messages that are sent and received will be understood.

2.4.2.3 Number of Stop Bits

This is really a continuation of the encoding mentioned in the last section. The choices are either 1 or 2, with the actual number selected having no real meaning outside the internal hardware of the two devices that are attempting to talk to each other. As long as the two machines are using the same number of Stop Bits, it makes no difference which is used.

2.4.2.4 Parity

Parity generation and testing is a method of automatic error detection that is often used in ASCII data transmission via RS-232C. It helps verify that a message has been correctly transmitted and received, and is frequently used. Without going into details, there are three types of parity that a device can use: Odd, Even or None. If two devices are to successfully communicate with each other, they must both be set to treat parity in the same manner.

2.4.2.5 Protocol

Did you ever try to take notes while listening to someone who is speaking very quickly, and at some point you say, "Hold it! I need some time to catch up."? Assuming a polite speaker, he or she then paused until you said "OK" and then started up again. That's *protocol*.

As fast as computer equipment might be, it too can get saturated and need some time to catch up. There are two common ways that this sort of protocol can be implemented within the framework of the ASCII code and RS-232 communications. We'll start by looking at how it is done via ASCII codes.

Part of the definition of the ASCII code includes things called *Control Characters*. These are codes that do not appear as separate keys on your keyboard, but instead are used as signaling codes between devices. Among these are the characters XON and XOFF,

which stand for Transmit On and Transmit Off, respectively. These characters are the equivalent of the "OK" and "Hold it!" we used in the analogy on the previous page, and are used in the same manner.

Having spent all that time discussing protocol, the real issue is the same as in all of the other parameters we have been discussing so far–sameness. As long as the two devices trying to talk to each other are either both using or not using XON/XOFF protocol, that's all that really counts.

Note that if you have a choice of using XON/XOFF or not, the deciding factor is the overall system data rate. Any time it is possible for one device to send so much data so quickly that the receiver cannot cope with it all, XON/XOFF protocol is one way to prevent a system overload.

With certain brands of host computers, and when you're using telephone communications links, XON/XOFF is often the only type of protocol you can use. But when you are not using the phone system and the host will allow it, a hardware form of protocol known as handshaking is a better choice.

2.4.2.6 Handshaking

Handshaking is the other common method used to handle the "Hold it!/OK" problem, and is functionally equivalent to the XON/XOFF protocol we just discussed. The difference is that a couple of the wires in the RS-232C cable are used to implement the protocol, rather than the ASCII control characters.

These wires are defined within the RS-232C standard, and have the names Data Terminal Ready (DTR) and Data Set Ready (DSR). They are used as follows:

1) The sender signals via DTR that there is some information ready to be sent.
2) The receiver, when it is ready and able to accept data, signals via DSR that it's okay to do so. This is directly analogous to the receiver sending XON.
3) The sender, upon sensing the DSR "OK," begins transmission.
4) If the receiver ever gets overloaded, it signals that condition using DSR in the same way that XOFF was used. The transmitter then pauses until DSR indicates "OK," at which time it starts sending data again.

The end result is exactly the same as the XON/XOFF protocol, except that discrete wires are used instead of ASCII codes. And just like before, if DTR/DSR handshaking is to be used, both devices must use it if a successful conversation is to take place.

2.4.2.7 DTE vs. DCE

In the previous section a Data Terminal and a Data Set were mentioned. The VuePoint™ III is obviously a Data Terminal, so whatever goes on the other end of the wire must be a Data Set, right?

The answer, unfortunately, is yes and no. Both Data Terminal Equipment (DTE) and Data Communications Equipment (DCE, of which a Data Set is a member) have official RS-232C definitions. And part of that definition says that to carry on a conver-

sation, one end of the wire must have a DTE device attached and the other end a DCE device. That's the "yes" part of the answer.

The "no" part is the fact that almost 100% of the terminal, printer and computer manufacturers ship their equipment set for operation as a DTE device, essentially guaranteeing that they can't communicate with each other as they come out of the box!

Believe it or not, there is a method to what first appears as madness. RS-232C was originally designed for data communications via phone lines. The terminals, which are DTE devices, are connected to telephone adaptors, which are indeed DCE devices. Because this is the way it all started, manufacturers assume that this is the way their equipment will be used and ship it configured as DTE devices.

While this sort of interconnection may have been the norm at one time, it is probably not the method you're going to use. Most likely you're interested in bypassing the phone line and connecting your VuePoint™ III directly to another terminal or computer.

Since this means you'll be trying to connect two DTE devices together, something that by definition won't work, you've obviously got a problem. Fortunately, it's one that has an easy answer. In fact, you've got a choice of two solutions to this dilemma:

1) Convert one of the devices from a DTE device to a DCE device. Most manufacturers include a way of doing this, and though it's not all that hard to do, it does require opening up the unit and changing some controls inside. For that reason, it's not the recommended approach to use at this time.

2) Simulate the existence of the telephone equipment using that short piece of cable called a *null modem* (*modem* is the technical name for the DCE device used to connect to a phone line) that was briefly mentioned in Section 2.1. Since this requires no changes to the internal workings of anything, it's the preferred method for now.

   All this null modem does is switch some signals from one pin to another in the cable such that the DTE in either end thinks it is connected to a DCE device. Everyone's connection requirements are then satisfied, making meaningful communications possible. It's really that simple.

2.4.3 Keyboard Conventions

Regardless of whether you'll be using another terminal or a host computer to communicate with your VuePoint™ III, at some point you'll have to use a keyboard to enter information to be transmitted to the VuePoint™. A few minutes spent on the conventions that we'll be using to do that will make the future sections much easier to understand.

2.4.3.1 Printing ASCII Characters

As you can see in Figure 2-5, most of the keys on a terminal or computer keyboard look just like those on a standard typewriter and are used in the same way. The key with "A" on it means just that, as does the "1" and all the other alphanumeric and punctuation characters.

And just as you would expect, if you hold down the Shift key when you strike a character key, you get the upper case of the key. Thus, "A" gives you an "a" and "Shift/A" gives you an "A", "Shift/1" gives you the "!", etc.

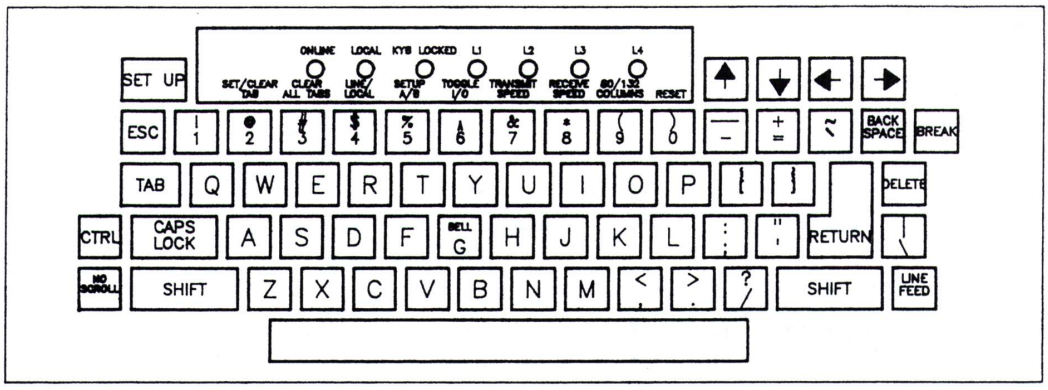


Figure 2-5
Typical ASCII Keyboard

2.4.3.2 Control Characters

A few pages back we covered the protocol characters XON and XOFF, calling them *Control Characters*. They are just two of the 32 possible control characters that are included as part of the definition of the ASCII code.

In general, all of the alphabetic characters (plus a few others, since there are only 26 letters in the alphabet), have a related control function. To generate these control characters, the key marked Ctrl, for Control, is used in a manner similar to the Shift key; that is, you hold it down while striking a character key. For example:

Key	Character
Q	q
Shift/Q	Q
Ctrl/Q	XON/XOFF

Note that Ctrl/Q doesn't cause the character string "X O N" to be generated, but rather the special single character whose name is XON.

Some of these control characters also have keys reserved specifically for them. Carriage Return (CR) and Line Feed (LF) are good examples of two control characters that are used so frequently, the cost of the extra keys is warranted. The keys with the arrows, which are used to move a cursor or marker around on a CRT screen, are also in that category.

2.4.3.3 The ESC Key

The key labeled ESC (for Escape) adds a third level of meaning to the ASCII codes that are generated when a key is struck.

Unlike Ctrl, ESC should not be held down while a second key is struck. Instead, ESC transmits a unique control character that tells the receiving device that the characters which follow it are to be interpreted in some special way. The exact meaning of "special way" is dependent upon the receiving device involved. For VuePoint™ III, it usually means that the ASCII codes which follow the ESC character are to be considered as a command to the device rather than displayable characters.

For example, if you sent the characters "Hello out there" as is, the receiver would normally consider that to be a plain message. On the other hand, if you sent "Hello ESC xxxx out there" the receiving device would realize that the xxxx is not part of the message and interpret it as a command. In other words, ESC tells the receiver to "Escape from the normal mode of operation."

Note: The VuePoint™ allows the ESC character to be changed to any other character the user wishes. This can be done from the VuePoint™ III Setup menu. See Section 2.5.1.2.

2.5 Installation

To put the general theory we've just covered into practice, we'll actually install the VuePoint™ III. Two procedures are given. The first, in Section 2.5.1, assumes you'll be using a Digital Equipment Corporation™ VT-100 family terminal or a similar device for communicating with the VuePoint™ and trying the various examples we'll be covering.

If you'll be using an IBM PC or other computer system instead, use the procedure in Section 2.5.2. Note that if you intend to take this route, you'll need at least a nodding acquaintance with BASIC language programming, the BASICA interpreter and PC DOS. If you don't fit into that category, it's strongly recommended that you not try using the PC approach until you've become more familiar with both the PC and VuePoint™. One unknown at a time is enough in any learning situation!

2.5.1 VuePoint™ III to VT-100

2.5.1.1 Cable Connections

If you are using the cables mentioned in the Preface, all you have to do is install them as outlined below. If you're having a custom cable made or are using another brand, it should be manufactured as shown in Table 2-1 below.

VUEPOINT™ III DB-25P		VT-100 DB-25S
Prot. GND	1<----------->1	Prot. GND
TX Data	2<----------->3	RX Data
RX Data	3<----------->2	TX Data
Data Set Ready	6<----------->20	Data Terminal Ready
Sig. GND	7<----------->7	Sig. GND
Data Terminal Ready	20<----------->6	Data Set Ready

DB-25P: Amphenol 117
DB-25S: Amphenol 117

Table 2-1
VuePoint™ III to VT-100 Cable Specification

1) Locate the male end of the main communication cable (the end where all of the gold pins are visible inside the connector shell) and plug it into the 25-pin connector on the rear panel of the VuePoint™ III.

2) Connect the male end of the short null modem cable to the opposite end of the main cable.

   Note: This step can be bypassed if you're using a custom cable as shown in Table 2-1.

3) Connect the female end of the null modem (or custom cable) to the 25-pin connector on the rear panel of the VT-100.

That's all there is to it; the physical connection is now complete. Next, we'll look at the logical connection between the two devices.

2.5.1.2 VuePoint™ III Setup

Setup is the name commonly given to the process of ensuring logical compatibility between two RS-232C devices. Both the VuePoint™ III and the VT-100 have a Setup Mode of operation to accomplish this. The VuePoint™ III will be set up first, using the following procedure:

1) Turn the VuePoint™ III power supply on, which will give you the Power-up Display shown in Figure 2-1.

2) Touch the line containing the word SERIAL, which will activate the communications setup procedure. The display should look as shown in Figure 2-6. If it is not exactly the same as shown, just touch the words that are different and they will change to the correct settings.

3) Once you've set these parameters to the correct values, touch EXIT to go on to the next screen, which is used to set the Baud Rate (refer to Figure 2-7).

   If the CURRENT VALUE is anything other than 9600 BAUD, touch UP or DOWN to increase or decrease the value. Once 9600 BAUD is displayed, touch EXIT.

4) You should now be at the screen that is used to enable or disable XON/XOFF protocol. If the display doesn't look as shown in Figure 2-8, touch the word ENABLED to change it to DISABLED. Touch EXIT and you're back to the original screen; the setup of VuePoint™ III is complete.

Note that you only have to do this once. Non-volatile memory is used to store these parameters, so you won't have to repeat this procedure unless you are going to install the VuePoint™ with another device and need different parameters.

WARNING!

If you change any parameters in SERIAL, VUEPOINT OPTIONS and ESCAPE SELECT, the settings will <u>not</u> be saved until you touch EXIT on the VUEPOINT SETUP screen (Figure 2-1).

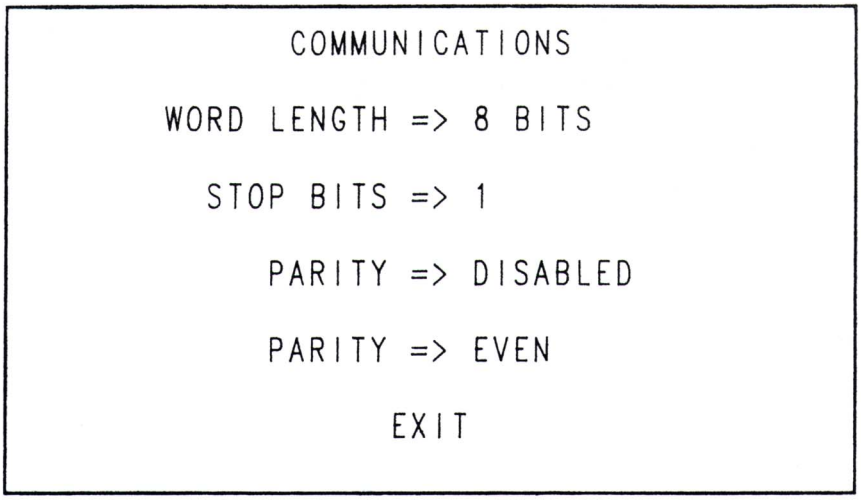


Figure 2-6
Communications Setup Screen

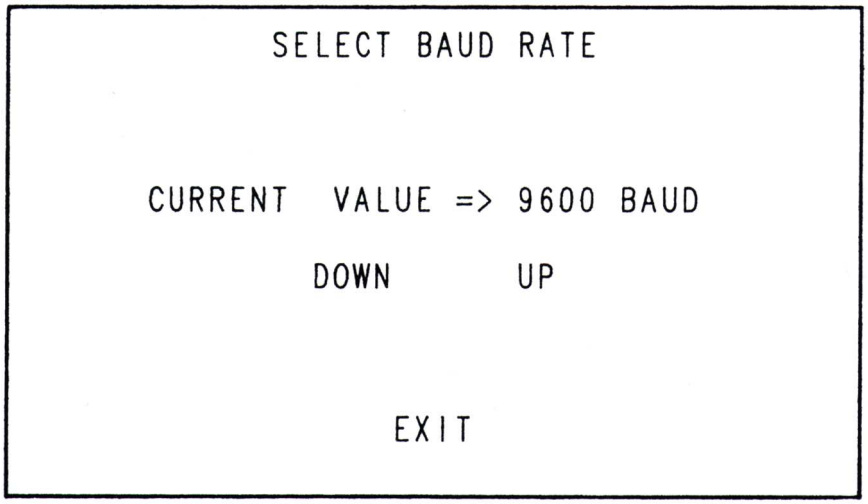


Figure 2-7
Baud Rate Setup Screen

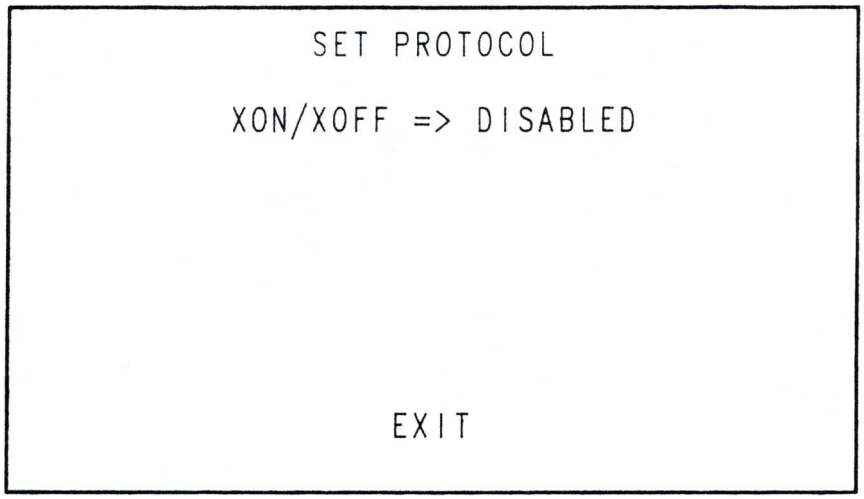


Figure 2-8
Protocol Setup Screen

2.5.1.3 VT-100 Setup

The following procedure is used to ensure that the VT-100 terminal is set to use the same operating parameters as the VuePoint™ III.

1) Turn the VT-100 power on, which will cause the unit to beep and a short dash to be displayed on the CRT.

2) Strike the large key labeled SET-UP in the upper left portion of the keyboard. This will yield the display shown in Figure 2-9.

3) Next strike the 5 key, which you'll notice has the label SETUP A/B printed above it. This moves you to the next part of the setup procedure.

4) If the screen does not look exactly as shown in Figure 2-10, strike the 0 (zero) key, which is used in SET-UP to reset the parameters to their default state. If the VT-100 screen does match the figure, strike SET-UP to exit.

   If you used the 0/RESET key, you'll be back to the original screen at this time. Repeat steps 2 and 3 and again compare your screen to the Figure 2-10 starting point.

   If the VT-100 screen still does not match the figure, refer to the manual that came with your VT-100 to change the needed parameters, striking SET-UP when you are finished to save the values.

At this point, the installation and setup of both the VT-100 and the VuePoint™ III are complete, and the system is ready to use. Continue on to Section 3.0 to begin the actual operating procedures.

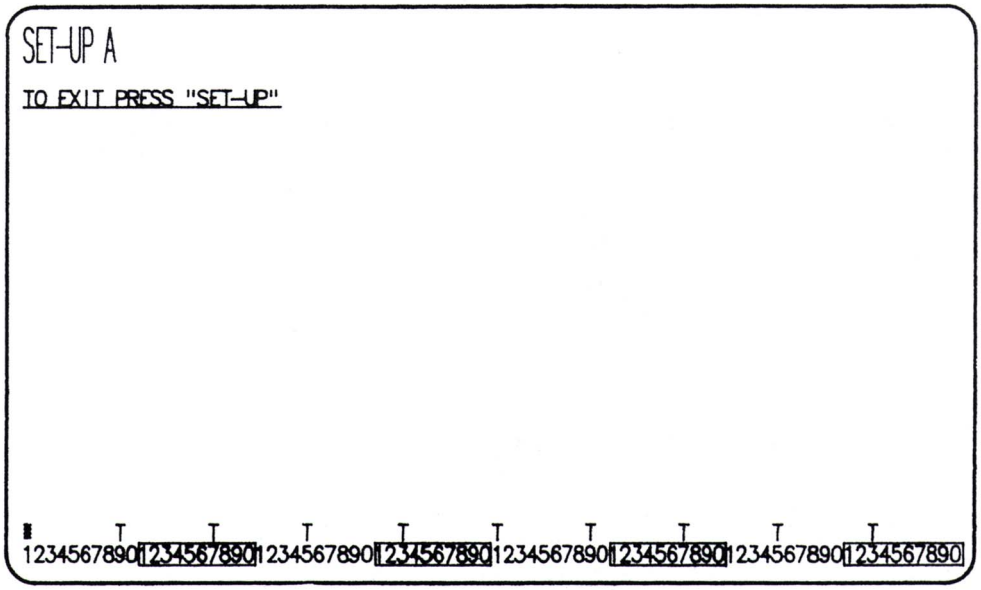


Figure 2-9
VT-100 SET-UP A Screen

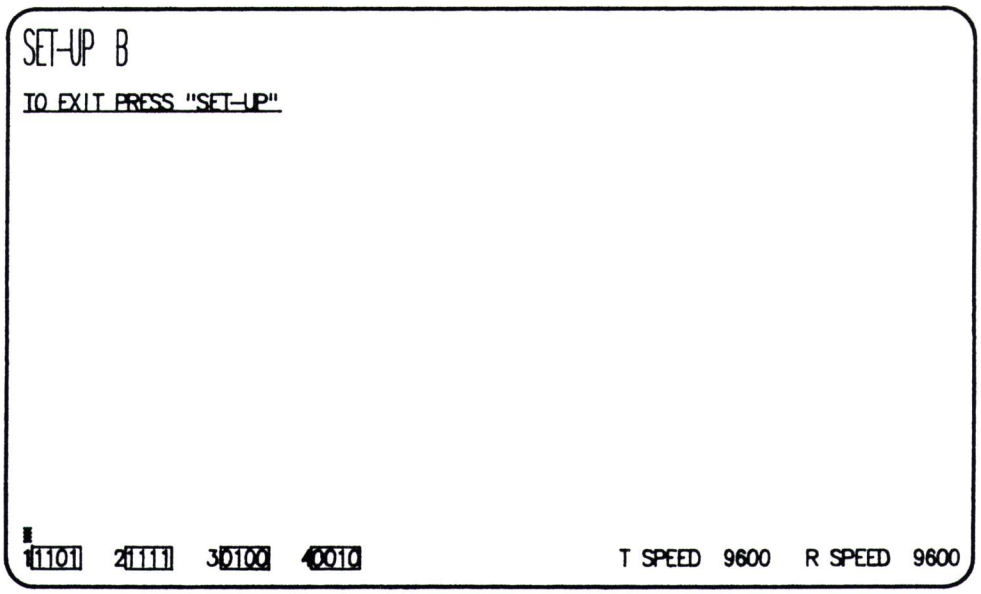


Figure 2-10
VT-100 SET-UP B Screen

2.5.2 VuePoint™ III to an IBM PC

2.5.2.1 Cable Connections

If you are using the cables mentioned in the Preface, all you have to do is install them as described in Section 2.5.1.1, substituting the IBM PC Serial Port for the VT-100. If you're having a custom cable made or are using another brand, it should be manufactured as shown in Table 2-2 below.

2.5.2.2 VuePoint™ III Setup

Since we'll be using the exact same operating parameters as before, perform the steps outlined in Section 2.5.1.2.

VUEPOINT™ III DB-25P		IBM PC DB-25S
Prot. GND	1<----------->1	Prot. GND
TX Data	2<----------->3	RX Data
RX Data	3<----------->2	TX Data
Data Set Ready	6<----------->20	Data Terminal Ready
Sig. GND	7<----------->7	Sig. GND
Data Terminal Ready	20<----------->6	Data Set Ready
	----------->4	Request to Send
	----------->5	Clear to Send

DB-25P: Amphenol 117

DB-25S: Amphenol 117

Table 2-2
VuePoint™ III to IBM PC Cable Specification

2.5.2.3 OPENing the BASIC I/O Channel

In BASICA on the IBM PC, OPENing the I/o Channel to be used with the VuePoint™ is directly analogous to the setup procedure used with the VuePoint™ and the VT-100. The statement to use is:

xxxx OPEN "com1:9600,n,8,,cs,ds,cd" AS #1

where xxxx is the line number of the statement if it is to be part of a program rather than immediately executed. Once this statement is executed, the PC's RS-232C port will be set to the same parameters as the VuePoint™ III.

2.5.2.4 Executing the Examples from BASICA

Once the channel is OPENed, a PRINT statement directed to channel #1 will send text to the VuePoint™, as follows:

xxxx PRINT #1, “PUT THE TEXT BETWEEN QUOTES”

In the next section, which covers the basics of displaying text with VuePoint™, the VT-100 will be used for the examples. To duplicate the results using the PC system, any reference to “type the characters .....” should be treated as “send via a PRINT #1 statement the characters .....”.

If you've come here directly from the previous section, you'll have a blank screen on your VT-100 and the Power-up Screen on the VuePoint™ III. Touch EXIT, which will give you a blank VuePoint™ screen, and you're ready to go.

3.1 The VuePoint's™ Screen Layout

The VuePoint's™ screen provides a display area of 12 rows of text with 40 characters per row, giving a total of 480 characters of display space per screen. Each such screen full of characters (even if they are blanks, they are still characters!) is called a page of text.

The rows in a screen (or page) are numbered from 0 to 11 (top to bottom), and the character positions in a row from 0 to 39 (left to right). Thus, the character in the upper left corner of the screen is in position 0,0 (row 0, column 0), the character in the lower right is in position 11,39, and so forth. This can be seen more clearly in Figure 3-1 below.

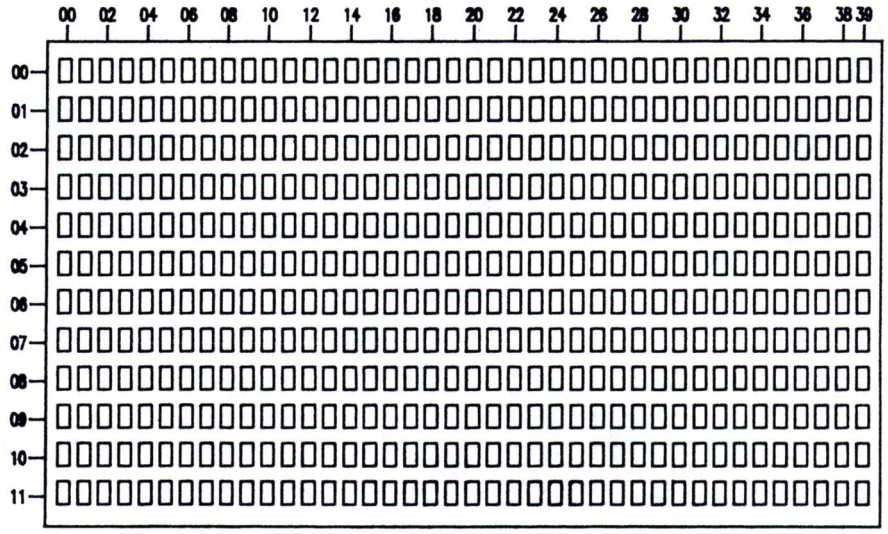


Figure 3-1
VuePoint™ III Screen Layout

3.2 Basic Text Display

Arrange your VuePoint™ III and VT-100 such that the VT-100 keyboard is sitting directly in front of the VuePoint™, allowing you to type on the keyboard while viewing the VuePoint™ screen. If that sounds like we're ignoring the VT-100 CRT, we are. The VT-100 is simply going to be used as a "communications controller," and not a terminal in the traditional sense.

Next, ensure that the CAPS LOCK key to the left of the ASDF.... row of the keyboard is on. In this case, "on" means resting on a level lower than that of the other keys.

This is done to cause the upper case ASCII code for all of the alphabetic characters to

be transmitted as the keys are struck, for upper case is VuePoint's™ standard way of displaying the letters A–Z.

Now type your name remembering not to use the shift key for the first letter. It should appear in the upper left corner of the VuePoint™ screen, with the first character in location 0,0 (see Figure 3-2 for an example).

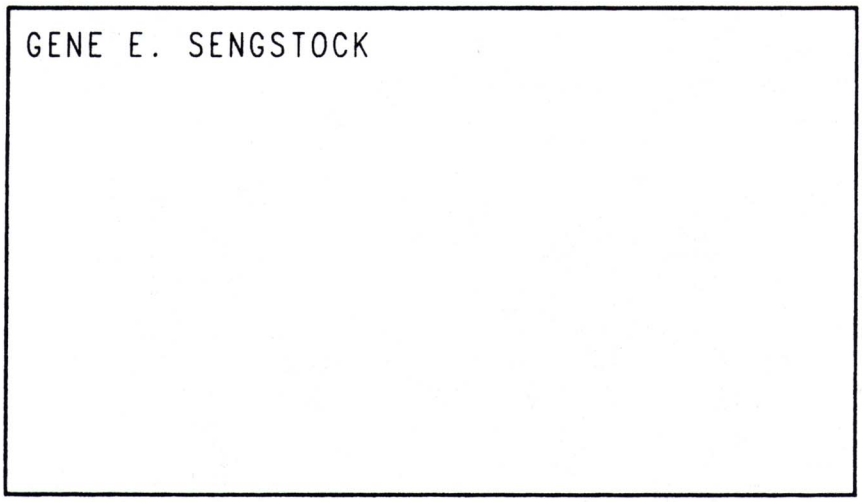


Figure 3-2
Initial Text Display

Now strike the key labeled RETURN, and type your name again. As you would expect, your name is displayed again. But note the double line spacing; for some reason a line was skipped on the screen. This can be resolved by using an additional section of the VuePoint's™ setup menus.

Turn the VuePoint™ off and then back on again to get the Power-on display on the screen, then touch the line containing VUEPOINT OPTIONS. The screen should look as shown in Figure 3-3.

The first line indicates the CURSOR status. The cursor is the VuePoint's™ *current character position* marker–a special symbol to let you know just where on the screen the next character that is received will be displayed–if it has been disabled we cannot see it. Touch the word DISABLED next to CURSOR to change it to ENABLED.

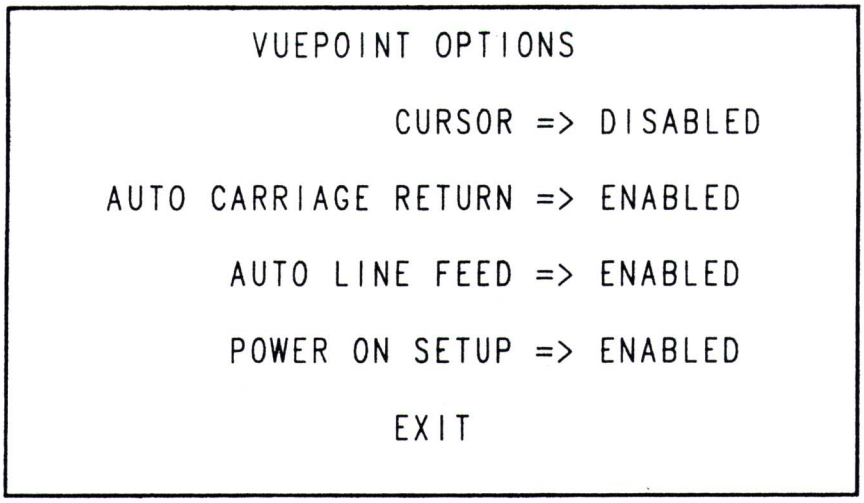


Figure 3-3
VuePoint™ Options Menu

Next, note the line that states AUTO LINE FEED is ENABLED. This means that every time the VT-100 (or whatever host device you're using) sends the RETURN character–which moves the current character position to the left edge of a row–the VuePoint™ will automatically move you down on row, too.

However, the VT-100 also does us this same favor; that is, every time it transmits a RETURN code it automatically sends the code for advancing to the next line. With both terminals automatically advancing you to the next line, the result is double spacing.

To eliminate this, touch the word ENABLED next to AUTO LINE FEED, causing it to change to DISABLED. Now we will get normal single spacing and your VuePoint™ screen should now look like the one in Figure 3-4.

Ignoring the other two parameters on the screen (we'll cover them later), touch EXIT to return to the Power-on setup menu. Touch EXIT again and you'll be back to a blank screen and ready to display more text.

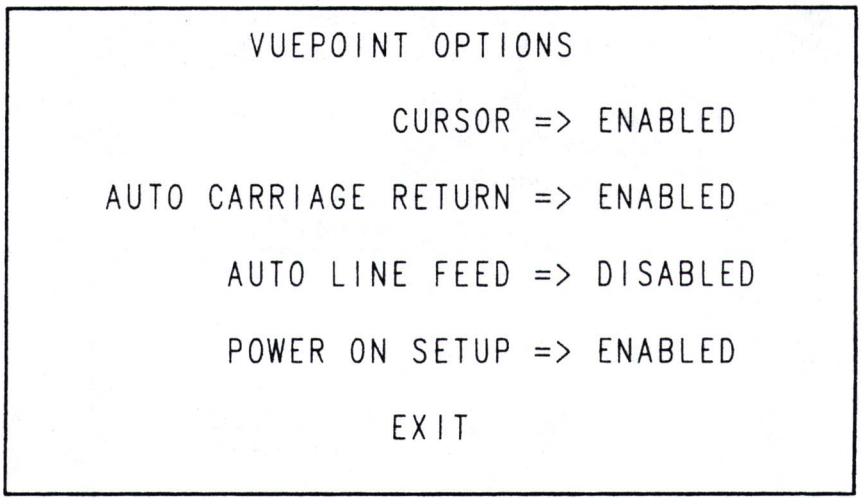


Figure 3-4
Modified VuePoint™ Options

But the screen isn't totally blank, is it? Now there's a blinking rectangle in the upper left corner–that's the cursor–to let you know that's the current character location on the screen.

To see how it's used, type your name just as before. The cursor, you'll notice, always lets you know where the next character will go on the screen. And if you strike the RETURN key, the cursor will show you that only one line advance took place; double spacing is no longer in effect.

Go ahead and try typing some additional characters. Use an assortment of letters, numbers and symbols just to get a better feel for how they look on the screen. Easy, isn't it? And that's all there is to it. Whatever you type (or send over from a host computer) will be displayed, with the cursor always letting you know exactly where the next character will appear.

3.3 Erasing the Screen

After a few minutes of typing, you'll quickly discover that the screen fills up and you need a way to erase it. While you could do this by typing spaces into every display position, that's hardly an efficient means of accomplishing the task. Instead, we'll use our first VuePoint™ *Escape Sequence* to erase the entire screen at once.

Recalling our discussion in Section 2.4.3.3 on the operation of ESC (as the Escape Key is known), VuePoint™ III considers all messages that start with the ESC code to be commands. Thus, what we'll actually be sending to the VuePoint™ is a command to erase the screen.

Noting the current position of the cursor on the screen, strike the key labeled ESC (don't hold it down) followed by the E key and the 0 (zero) key. While no message was displayed, the screen is now blank; it was completely erased.

The cursor, you'll notice, is still displayed in its original position. Assuming you wish to start your new page of text in the upper left corner, the next step is to learn how to move the cursor about on the screen.

3.4 Positioning Text on the Screen

By erasing the screen, we introduced the use of ESC commands to control the VuePoint™. Positioning text, which really means moving the cursor (current character position) about the screen, will introduce the use of *Control Characters*.

Unlike ESC, which actually generates a separate special ASCII code, CTRL (for Control) is used to modify the code that is generated by another key on the keyboard. (See Section 2.4.3.2 for additional information on this.) It is used by holding down the key marked CTRL while another key is struck, just like you would use the SHIFT key.

Table 3-1, below, lists some of the VuePoint™ III CTRL commands that are used to move the cursor about on the screen.

MOTION	KEYSTROKE
Cursor Down	CTRL/J
Cursor Up	CTRL/K
Cursor Left	CTRL/H
Cursor Right	CTRL/L

Table 3-1
Cursor CTRL Codes

Each moves the cursor one character position in the direction specified. Go ahead and try them, remembering that CTRL/J means "Hold down CTRL while typing a J."

3.5 Homing the Cursor

In addition to these single character cursor movement CTRL codes, there is also an ESC command that you'll probably want to use at the beginning of each new screen writing sequence: ESC H. This command moves the cursor to the *Home* position, which is the upper left corner of the screen (row 0, column 0).

At this point, we've covered the fundamentals of displaying text with VuePoint™ III, including the ability to position the cursor–and therefore any text you happen to type in at the cursor position–anywhere on the screen. While this only scratches the surface of VuePoint's™ display capabilities, it does allow you to begin creating useful displays right away. Spend some time becoming familiar with the material in this section before going on to section 4, where we'll cover the basics of adding touch input to your screen displays.

4.0 VuePoint™ III Touch Input 36

This section will cover the basics of using the touch input facility of VuePoint™ III. While the VT-100 will work just fine as the host for the initial examples, a computer is needed as the host to see just how the touch input can best be applied.

4.1 How VuePoint™ III Touch Input Works

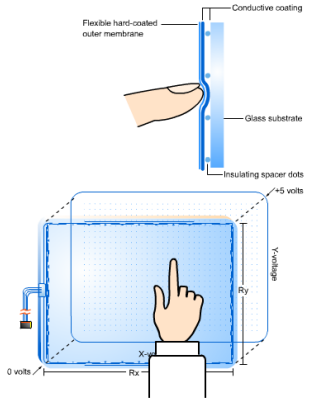


Figure 4-1
The Touchscreen Membrane

4.1.1 The Parts of a Touchscreen

The five-wire resistive touchscreen uses a glass panel with a uniform resistive coating. A thick polyester coversheet is tightly suspended over the top of a glass substrate, separated by small, transparent insulating dots. The coversheet has a hard, durable coating on the outer side and a conductive coating on the inner side.

4.1.2 What Happens During a Touch

When the screen is touched, it pushes the conductive coating on the coversheet against the coating on the glass, making electrical contact. The voltages produced are the analog representation of the position touched.

4.1.3 How the Touchscreen Controller Interprets Screen Measurement

When the controller is waiting for a touch, the resistive layer of the touchscreen is biased at +5V through four drive lines, and the coversheet is grounded through a high resistance. When the touchscreen is not being touched, the voltage on the coversheet is zero. The voltage level of the coversheet is continuously converted by the analog-to-digital converter (ADC) and monitored by the microprocessor on the controller.

When the touchscreen is touched, the microprocessor detects the rise in the coversheet voltage and begins converting the coordinates as follows:

The microprocessor places the X drive voltage on the touchscreen by applying +5V to pins H and X and grounding pins Y and L. An analog voltage proportional to the

X (horizontal) position of the touch appears on the coversheet at pin S of the touchscreen connector. This voltage is digitized by the ADC and subjected to an averaging algorithm, then stored for transmission to the host.

Next, the microprocessor places the Y drive voltage on the touchscreen by applying +5V to pins H and Y and grounding pin X and L. An analog voltage proportional to the Y (vertical position of the touch) now appears on the coversheet at pin S of the touchscreen connector. This signal is converted and processed as described above for the X position

4.1.4 Why the Averaging Algorithm is Important

The averaging algorithm reduces noise resulting from contact bounce during the making and breaking contact with the touchscreen. Successive X and Y samples are tested to determine that their values differ by no more than a certain range. If one or more samples fall outside this range, the samples are discarded and the process is restarted. This is continued until several successive X samples (then Y samples) fall within the range. The average of these values is used as the X and Y coordinates respectively.

Once independent X and Y samples are obtained, coordinate pairs are sampled to eliminate the effects of noise. If a sample does not fall within an internal range, all X and Y coordinates are discarded and the independent X and Y sequence is restarted. Once acceptable coordinates have been obtained, an average coordinate is determined and communicated to the host processor.

4.1.5 Video Alignment

The X and Y values are similar to Cartesian coordinates, with X increasing from left to right and Y increasing from bottom to top. These absolute coordinates are arbitrary and unscaled, and will vary slightly from touchscreen to touchscreen. The touchscreen controller can be calibrated for video alignment. This aligns the touchscreen coordinate system with the display image, reorients each axis, and scales the coordinates before they are transmitted to the host computer.

4.1.6 X- and Y-Axis Measurements Originate from the Glass

The five-wire technology utilizes the bottom glass substrate for both X- and Y-axis measurements. The flexible coversheet acts only as a voltage-measuring probe. This means that the touchscreen will continue working properly even with nonuniformity in the coversheet's conductive coating. The result is an accurate, durable, and reliable touchscreen that offers drift free operation.

4.2 A Technical Comparison with Other Technologies

4.2.1 Why Resistive Touchscreens?

Resistive touchscreens are used in more applications than any other touch technology–for example, in hand-held computers, PDAs, industrial equipment, point of sale equipment, medical equipment, office automation equipment and consumer electronics.

Resistive touchscreens allow all kinds of touch input devices to activate the screen, including fingers, fingernails, styluses, gloved hands, and credit cards–all the while maintaining superb tactile feel. Also, both the touchscreen and its electronics are simple to integrate into imbedded systems, thereby providing the most cost-competitive touchscreen solution.

4.2.2 Why Five-Wire Resistive?

Thanks to its inherently stable design, five-wire resistive is the most accurate and most reliable touchscreen technology. Operationally tested to over 35 million touches in one location with a stylus similar to a finger, it also has the longest life. There's no need for recalibration when environmental conditions change, and five-wire's resistance to abuse makes them ideal for heavy-usage environments.

The charts below compare five-wire resistive to two other touchscreen technologies in use today: four-wire resistive and eight-wire resistive.

4.2.3 Reliability

FIVE-WIRE TECHNOLOGY	FOUR-WIRE TECHNOLOGY AND EIGHT-WIRE TECHNOLOGY
Utilizes the bottom substrate for both X- and Y-axis measurements. The flexible coversheet acts only as a voltage-measuring probe. This means the touchscreen continues working properly even with nonuniformity in the coversheet's conductive coating. The result is an accurate, durable and reliable touchscreen that offers drift-free operation.	Must use two layers to create X- and Y-axis measurements. For the Y axis, the flexible top coversheet acts as a uniform voltage gradient, while the bottom substrate acts as the voltage probe. The constant flexing that occurs on the outer coversheet will change its electrical characteristics (resistance) with use, degrading the linearity and accuracy of this axis.

Table 4-1
Touchscreen Reliability Comparison

4.2.4 Durability

FIVE-WIRE TECHNOLOGY	FOUR-WIRE TECHNOLOGY AND EIGHT-WIRE TECHNOLOGY
Tested to over 35 million finger touches with no performance degradation.	1 million-touch life max.

Table 4-2
Touchscreen Durability Comparison

4.2.5 Design Flexibility

FIVE-WIRE TECHNOLOGY	FOUR-WIRE TECHNOLOGY AND EIGHT-WIRE TECHNOLOGY
Advanced design allows flat and spherical designs.	Not available in spherical designs.

Table 4-3
Touchscreen Design Flexibility Comparison

4.2.6 Construction

FIVE-WIRE TECHNOLOGY	FOUR-WIRE TECHNOLOGY AND EIGHT-WIRE TECHNOLOGY
Simple construction using glass substrate with a polyester coversheet suspended over the top of the glass. This is also known as plastic-on-glass construction, offering the lowest number of layers and best optical characteristics.	Typically have multiple layers of polyester, known as plastic-on-plastic construction. This assembly is laminated with adhesives onto a glass or plastic backer for support. The additional layers lead to degradation of optical clarity.

Table 4-4
Touchscreen Construction Comparison

4.3 Basic Touch Input

The first thing we have to do is get both the VuePoint™ and the VT-100 to known states with clear screens. For the time being, the easiest way to do that is to turn both units off and then on again. In a later section, we'll see there are also commands we can use for that purpose.

4.3.1 Enabling Touch Input

Since computers, the usual VuePoint™ host, normally don't want to be bothered by input from a peripheral device unless they have requested it, VuePoint™ will not acknowledge or send touch information until it has been told by the host that it's OK to do so. This is done by sending a single control character command to VuePoint™, much like we just did to move the cursor about the screen.

Send a CTRL/Q at this time by holding down the CTRL key while striking the Q. Note that nothing will be seen on either the VuePoint™ or the VT-100 displays except the cursors.

Now touch the lower left corner of the VuePoint™ screen. You'll hear a beep to let you know that a touch was sensed., and the screen of the VT-100 will look as shown in Figure 4-2. Note that if you touch the screen again, no beep will be heard. CTRL/Q allows only one touch to be sensed.

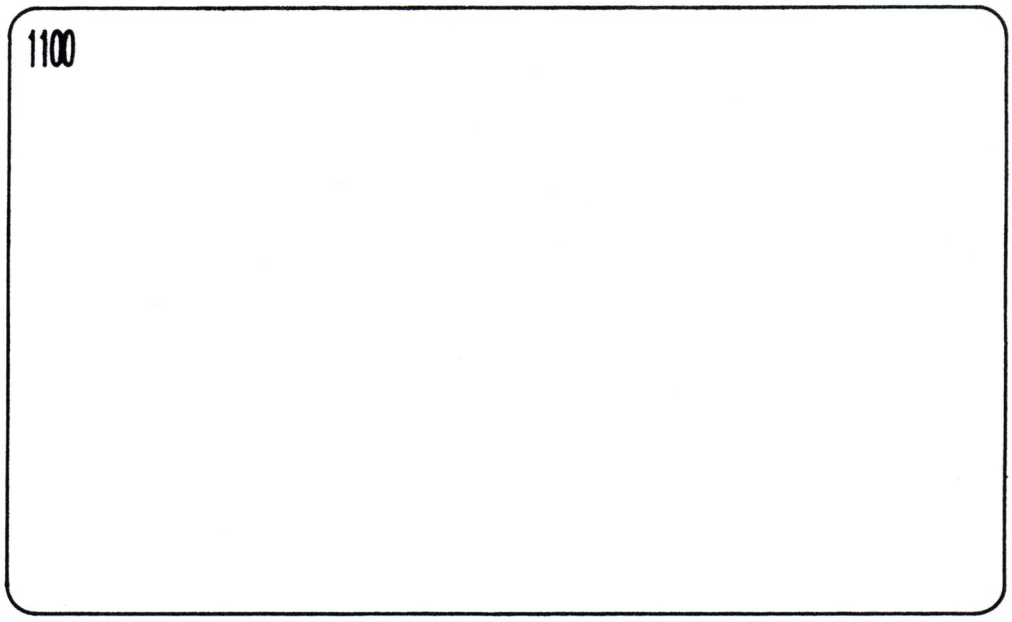


Figure 4-2
VT-100 Touch Input Display

At this point, your VT-100 should have the digits 1100 displayed in the upper left corner of the CRT, with the blinking cursor positioned under the first one. If that's not what you have, check the following:

1) If you VT-100 screen is still blank, the jumper shown in the cable drawing in Figure 2-7 is missing from your cable. You'll have to have it wired in if you're going to continue using the VT-100 for the touch input examples.

2) If some other 4 digit value is shown, you didn't really touch the lower left corner of the screen. The digits will indicate the position you actually touched.

4.3.2 Interpreting the Input

Assuming you've now got the 1100 on the VT-100 CRT, what does it mean? Recalling our earlier discussion on how the touchscreen operates, the 11 means that the touch was sensed in row 11 and the 00 indicates character position 00.

To see how this changes as you touch different areas of the screen, type CTRL/Q again to enable the next touch input and touch the screen in a different location. Each time you do this, a new row and column coordinate location will be seen on the VT-100.

Though it happens too fast for you to see it change, what is actually taking place on the VT-100 screen is this, but all on the same line.

1_
11_
110_
1100_
1100

The underscore represents the VT-100 cursor, which operates much like the one in VuePoint™ III. As each digit of the 1100 was received, the cursor moved one location ahead of it, indicating where the next digit would be displayed. But, in the end, the cursor ended up under the first digit. Why?

The answer is that VuePoint™ always follows the row and column digits with a RETURN (CR, for Carriage Return) code to tell the host that the message has ended. This moves the VT-100 cursor to the left edge of the screen just as it does in VuePoint™. It stays on the same row, however, because no LINE FEED (LF) code was sent to move the cursor to the next line.

4.4 Creating and Using Touch Buttons

This format of touch information, where the VuePoint™ always sends two row digits, two column digits and a CR is known as row/column reporting. Since it is totally general purpose in nature, it can used with any screen display for handling touch input. However, since it requires the host computer program to translate row and column information into an answer, it can be a little complicated to use.

To simplify touch input for its most common application–simulating hardwired push buttons–VuePoint™ provides a second mode of reporting called *Screen Echo Response*. This mode of operation, combined with some special display characters that are built into VuePoint™, makes push-button simulation extremely straightforward to implement.

4.4.1 Enabling Screen Echo Response

The first step in setting up a touch button display is to enable the screen echo form of reporting. This is done by another of the ESC sequence commands we discussed earlier. To see how it operates, perform the following steps from a clear screen:

1) Type the following on the VT-100 keyboard, starting with the A in the upper left corner of the display and including the spaces between the letters: A B C D E F G.

2) Enable screen echo by typing the ESC key followed by an R and then a 2. This will keep the VuePoint™ in screen echo mode until we change it back to row/column reporting.

3) Send the CTRL/Q touch enable.

4) Touch the A on the VuePoint™ screen.

On the VT-100 screen you'll now see the letter A, for VuePoint™ now returns the character that was touched (echoes the screen) instead of reporting the row and column number. Try a few more CTRL/Q and touch sequences with the other letters displayed

on the VuePoint™, and you'll see that the screen contents are always echoed. Note also that the resulting position of the VT-100 cursor again shows us that each letter is always followed with a CR end of message character just as before.

If you haven't tried it yet, do a CTRL/Q and then touch an empty part of the screen. You'll hear a beep, but the VT-100 screen won't change. While your first impression is that the VuePoint™ sent nothing in response to the touch, the beep tells you otherwise. An ASCII character was sent–the DEL character, which has a code value of 127–but the VT-100 cannot display it, so nothing appears to have happened.

At this point, one of the reasons for needing a computer host for using touch input has become apparent, for with the computer we can handle these "invisible" characters. For example, the BASIC program in Listing 4-1 duplicates the manual procedure we used above. Try it and you'll see that the DEL character as well as the ABCD, etc. are displayed for you.

Should you wish to return to row/column reporting, the command is ESC R 0, as opposed to the ESC R 2 which was used to enable screen echo.

```
10   REM OPEN the channel as in Section 2.5.2.3
20   REM
30   OPEN "com1:9600,n,8,,cs,ds,cd" AS #1
40   REM
50   REM Define some constants
60   REM
70   ESC = 27                    :REM The ESC Code
80   CTRLQ = 17                  :REM The CTRL/Q Touch Enable Cmd
90   ECHO$ = CHR$(ESC)+"R2"      :REM The Screen Echo Command
100  VPERASE$ = CHR$(ESC)+"E0"   :REM The Screen Erase Command
110  REM
120  REM Set up the Vuepoint
130  REM
140  PRINT #1, VPERASE$
150  PRINT #1, ECHO$
160  PRINT #1, "A B C D E F G"
170  REM
180  REM The Touch Input Loop
190  REM
200  PRINT #1, CHR$(CTRLQ)
210  INPUT #1, ANS$
220  PRINT ANS$,ASC(ANS$)
230  FOR I = 1 to 1000           :REM Wait a couple of seconds
240  NEXT I
250  GOTO 200
```

Listing 4-1
Screen Echo Reporting with BASIC

4.4.2 Defining Touch Buttons

For multiple choice operator dialogue, such as that used when you go through the VuePoint™ initial power-up sequence, the ABCD… type of touch input we used above is fine. But for most applications, a better human interface results when you use pictures of buttons with application-specific labels rather than choices labeled ABC… or 123….

For these situations, VuePoint™ contains some special characters that are ideal for the job. For example, look at the screen photo in Figure 4-4. Though it' a little simplistic, it's pretty easy to tell that we have two buttons labeled ON and OFF. To duplicate the screen:

1) Erase the VuePoint™ via ESC E 0, and home the cursor with ESC H.

2) Type in ON: p followed by a RETURN. Note that you must leave one and only one space between the colon (:) and the p, for the p must end up in an even numbered column if it is to be touch sensitive.

3) Type in OFF: q and a RETURN, with no spaces between the characters.

Note that though you typed in two different lower case characters–p and q–they both were displayed as "buttons" on the VuePoint™. You'll see why when you use CTRL/Q to enable a touch and push one of the buttons.

If you touched the ON button, the VT-100 will now be showing you the p you sent to VuePoint™; touch OFF and you'll see the q that was sent. Even though they look the same on the screen, the VuePoint™ knows the difference between them and sends the correct ASCII character back.

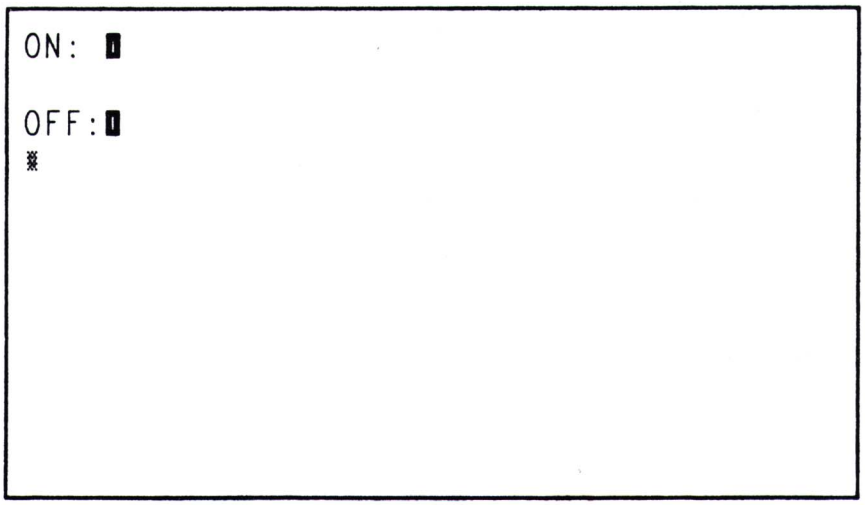


Figure 4-4
A Simple Touch Button Display

Other lowercase characters that work this way, besides p and q, are all eight letters from p through w, inclusive. They can all be used in exactly the same manner. Each displays the same touch button box on the screen, but returns its own unique character when touched. The end result is that the operator gets a friendly push-button interface and the host the ease of multiple choice responses, an ideal situation for both.

4.5 Push-button vs. Keyboard Operation

Up to this point, all of our touch input operations have used what is known as the push-button operating mode. In this mode, the touch logic acts as if the operator were actually holding in a button. The exact operation can be seen in the following sequence:

1) The operator touches the VuePoint™ screen and keeps his finger against the screen for several seconds.
2) While his finger is against the screen, a host computer sends the CTRL/Q touch enable.
3) The host will immediately receive a touch report from the VuePoint™.
4) In a matter of a few microseconds, the computer processes the touch and again sends the CTRL/Q.
5) Since the touch logic is simulating a button that is being held on, the host will immediately receive the same touch again (assuming the operator's finger is still on the screen).

In fact, the host will keep receiving the same touch response every time it enables the touch until the operator removes his finger from the screen. This can be seen easily by running the program in Listing 4-1 with your finger held against the screen for a few tenths of seconds.

While this mode of operation is appropriate for many applications, there will also be times when you will want one and only one response from a touch regardless of how long the operator leaves his finger on the screen. For those situations, keyboard operating mode is used.

When keyboard mode is selected, the touch logic operates much like the keyboard on a typewriter that does not automatically repeat the characters when you hold a key down. That is, a touch is recognized by VuePoint™ only when it occurs after the CTRL/Q is received, and when any touch that may have been present prior to the CTRL/Q is removed and a new touch sensed.

The end result is that the host is guaranteed to receive on and only one response from a touch, and that any "old" touches will be ignored.

To change the VuePoint™ from the default push-button mode to keyboard mode (or back again), another ESC command is used. If ESC T 1 is sent to VuePoint™, all future touches will be handled in keyboard mode; to convert back to push-button mode, the command is ESC T 0. Note that like all other ESC commands, the characters must be sent without the spaces we've shown to make the reading easier.

Compare the operation of the program in Listing 4-2 to the one in Listing 4-1 and you'll see that what sounds complicated when it's described is pretty straightforward once you've seen it operate.

```
10   REM OPEN the channel as in Section 2.5.2.3
20   REM
30   OPEN "com1:9600,n,8,,cs,ds,cd" AS #1
40   REM
50   Define some constants
60   REM
70   ESC = 27                    :REM The ESC Code
80   CTRLQ = 17                  :REM The CTRL/Q Touch Enable Cmd
90   ECHO$ = CHR$(ESC)+"R2"      :REM The Screen Echo Command
100  KEYMODE$ = CHR$(ESC)+"T1"   :REM The Keyboard Mode Command
110  VPERASE$ = CHR$(ESC)+"E0"   :REM The Screen Erase Command
120  REM
130  REM Set up the Vuepoint
140  REM
150  PRINT #1, VPERASE$
160  PRINT #1, ECHO$
170  PRINT #1, KEYMODE$
180  PRINT #1, "A B C D E F G"
190  REM
200  REM The Touch Input Loop
210  REM
220  PRINT #1, CHR$(CTRLQ)
230  INPUT #1, ANS$
240  PRINT ANS$,ASC(ANS$)
250  GOTO 220
```

Listing 4-2
Screen Echo with Keyboard Mode

Looking at Listing 4-2, you'll see the keyboard mode command defined in line 100 and sent to VuePoint™ in line number 170. Notice also that the time delay loop that was included in the first program (Listing 4-1) was removed, for we'll no longer get multiple touch responses from a physical touch.

Though we've only covered the basics of touch input at this point, this section will provide a good base on which to build when we discuss the more advanced VuePoint™ touch capabilities in later sections. Be sure to take the time to try the examples and fully understand the various commands we've used, for they are all applicable to any touch input applications you may want to implement.

5.0 Saving Your Displays 46

Now that we've covered the basics of setting up a VuePoint™ III display and an operator's touch input interface, we'll take a look at some of the features VuePoint™ includes to make it easy for us to save a display and recall it at will.

5.1 VuePoint's™ Page Orientation

Up to this point, we've always considered the VuePoint™ III to have the same basic internal structure as an ordinary display: an RS-232C communications link to a host, some logic to process those communications, and a screen to display the messages. While there's no question you can use VuePoint™ in that way–that's really all we've done so far–you can greatly reduce the time and effort needed to implement and run an application by taking advantage of something know as *Paging*.

5.1.1 What is a Page?

A page, as VuePoint™ knows it, is a block of memory internal to the terminal that is large enough to hold a complete screen full of text, cursor positioning information, touch control information, etc. In short, everything needed to display the text and handle the operator interface. All VuePoints™ contain at least three such blocks (or pages) of memory, with expansion options available for additional memory pages.

5.1.2 The Display Page

As the name suggests, the memory page currently being displayed on the screen is known as the *Display Page*. When power is first applied, the first of the various pages of memory (page number 00) is automatically selected to be the Display Page.

5.1.3 The Working Page

The *Working Page* is the page or block of memory that is assigned to receive all messages, cursor positioning information, touch commands, etc. from the host. It, too, is defined to be page number 00 when power is first applied.

Since both the Working Page (the message receiver) and the Display Page (the message Displayer) are the exact same block of memory, the VuePoint™ acts like a standard terminal: all text, commands, etc. that are sent by the host are immediately seen on the screen. But what would happen if the Display Page and the Working Page were assigned to different pages of memory? In the remainder of this section, we'll look at exactly what happens when that occurs and how we can use it to our benefit.

5.2 Using the Page Memory

As you would expect, if the Display Page and the Working Page are not the same page of memory, information sent to the VuePoint™ by the host will not be seen on the screen as it is received. But the information won't be lost; it will simply be stored in the page memory until you decide to display that particular page.

Before we can demonstrate just how this operates and how you might use it in a particular application, we have to discuss a few more ESC command sequences that VuePoint™ III recognizes.

5.2.1 Selecting a Working Page

To tell the VuePoint™ III which of its various memory pages you wish to use as the current Working Page, the command ESC W pp is used, where pp represents a two digit page number starting with number 00. Thus, the first page is selected with ESC W 00, the second with ESC W 01, and so forth.

5.2.2 Selecting a Display Page

The current Display Page is selected in exactly the same manner using the command ESC D pp. In addition, there is also a separate command to select the next page in sequence to be displayed, regardless of which page is currently on the screen. This is done via ESC N.

If the Display Page and the Working Page happen to be assigned to the same number–"normal" terminal operation–ESC N has another attribute which allows all of the memory pages to act as if they're connected together into one long page; it increments both of the page numbers simultaneously, allowing long text messages to be both displayed and stored with a minimum of effort on the part of the host.

5.2.3 Using the Page Commands

The following procedure illustrates how these three commands operate, and should you a better understanding of the relationships between the Display and Working Pages of the VuePoint's™ memory. Note that the VT-100 host is all that's needed for these basic operations.

1) Go through the power-on procedure for both the VuePoint™ III and the VT-100 to bring them to a known state.

2) Type the message PAGE 0, followed by a RETURN, on the keyboard. Since the Display Page and the Working Page are both assigned to 00 on power-on, the VuePoint™ screen should look as shown in Figure 5-1.

3) Now we'll change the Working Page by sending the command ESC W 01 (remember not to put any spaces between the characters). Note that the screen has not changed, for we're still displaying 00.

4) Next we'll carefully type in the message PAGE 1, followed by a RETURN. Again note that the original screen is still present, for our message went to page 01 and we're displaying page 00.

5) Change the Working Page to the third page via ESC W 02, and send the message PAGE 2, plus the RETURN. Again, there is no change in the VuePoint's™ screen.

6) Now we'll start changing the Display Page so we can see the results of the messages we've been sending, beginning with the command ESC D 01. Send this, and the screen will immediately change to look like Figure 5-2. We can now see the text we sent to page 01 in steps 3 and 4 above.

7) To see the text on the third page, type the command ESC D 02.

8) Lastly, set the Display Page back to 00 by ESC D 00.

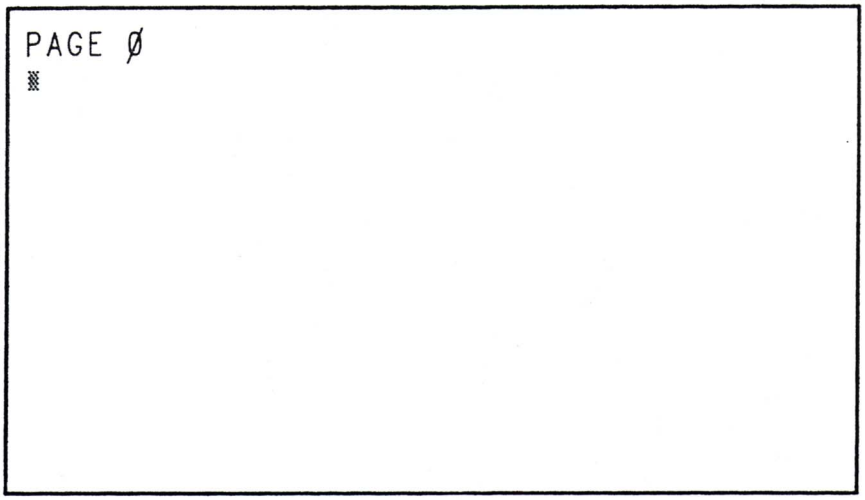


Figure 5-1
Page 00 Display

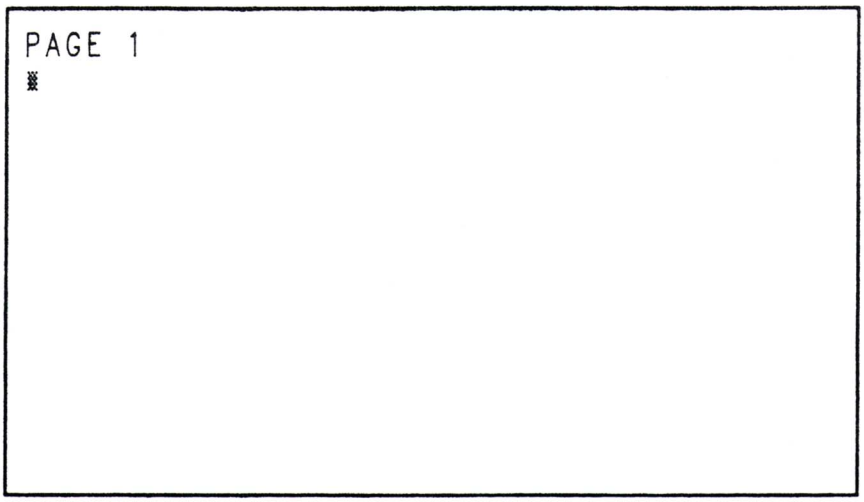


Figure 5-2
Page 01 Display

Our VuePoint™ now has text on three different pages of memory, page 02 is still selected as our Working Page, and we're displaying page 00. Given this scenario, we'll take a look at the operation of ESC N, the *display next page* command.

1) Type the command ESC N, which will move the screen to page number 01. The screen should again look like Figure 5-2.

2) Repeat the ESC N, causing page 02 to be displayed.

3) Since this is also the page that was last selected for the Working Page, any text you type on the keyboard should appear on the screen. Verify this by typing HELLO on the keyboard, which will give you the screen in Figure 5-3.

4) Recalling what we said about ESC N earlier, issuing the command again should simultaneously move both the Display and Working Pages to the next page of memory. Try it. If you have a basic VuePoint™, you'll be back to page 00 (Figure 5-1), for it only has three pages. If not, you'll be on to page 03. In either case, typing in a message will cause it to be displayed immediately, showing you that both the Display and Working Pages moved together.

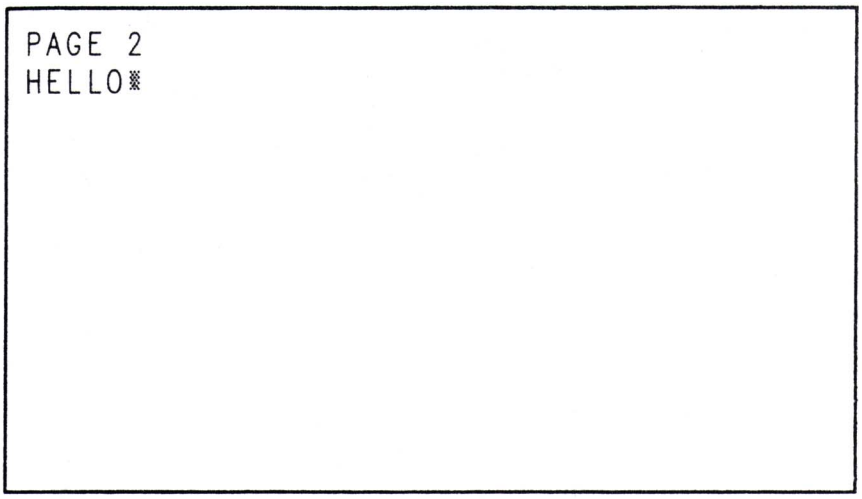


Figure 5-3
Page 02 as the Display and Working Page

5.3 A Paging Application

From the above, the basics of VuePoint™ paging can be seen, and you can probably envision some ways in which it can be used. To better illustrate how an application might be actually implemented, we'll take a look at a simple example written in BASIC. Though only three pages will be used–allowing the program to run with even the smallest VuePoint™–the concepts used are applicable to any size system.

The program will simulate an industrial application where the VuePoint™ III is being used as the operator's control panel. Three different screen displays are needed. The first for the *Master Control*, the second for *Operating Mode* selection when the process is started, and the third for the *Shutdown Mode* to be used when the process is stopped.

Figures 5-4 through 5-6 show the screen layouts we'll be using. Though they're hardly sufficient for a real application, you'll find them adequate for the principle we're demonstrating here.

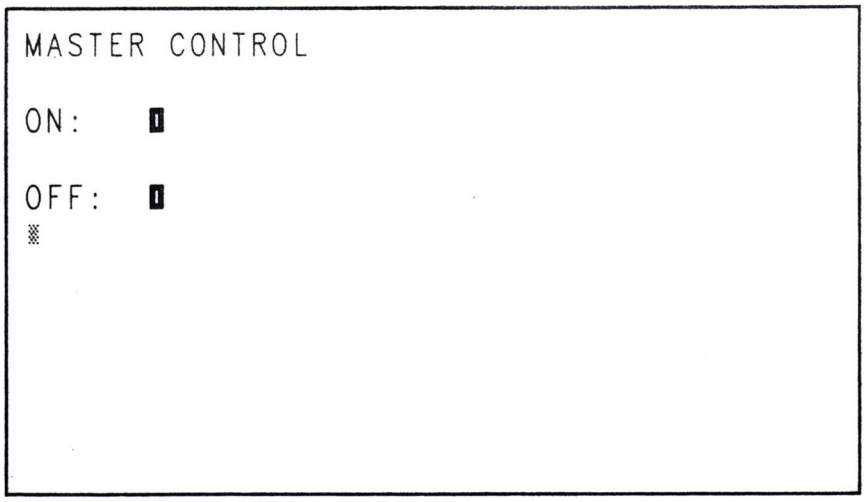


Figure 5-4
The Master Control Panel

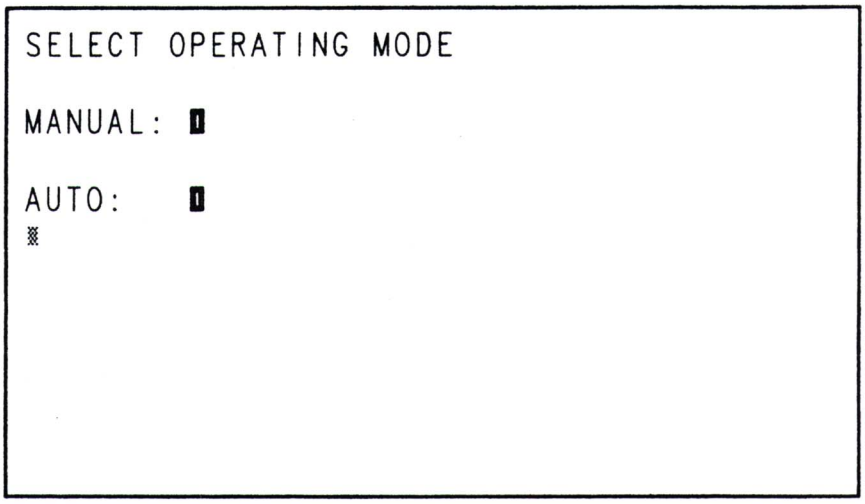


Figure 5-5
The Mode Selection Panel

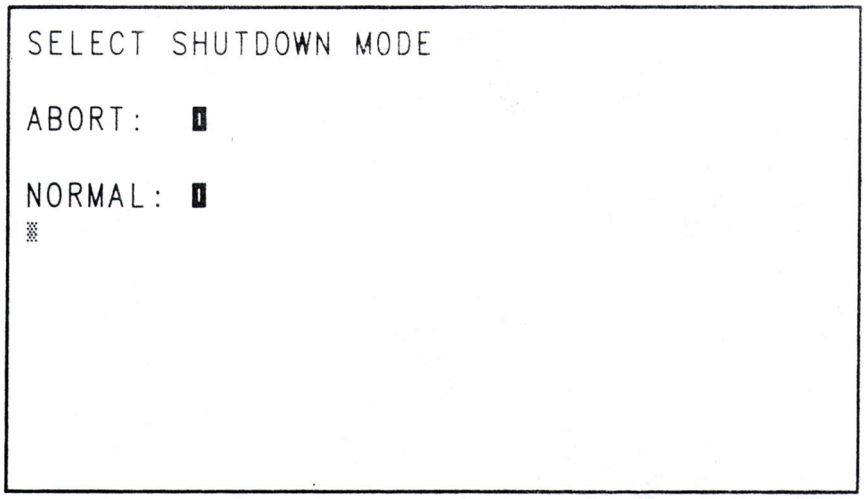


Figure 5-6
The Shutdown Mode Panel

Given the above screen designs, the operation of the program will be as follows:

1) The normal display will be the Master Control panel.

2) If the operator touches the ON button, the program will display the Operating Mode panel to allow him to choose the desired mode.

3) If the OFF button is touched, the program will display the Shutdown Mode panel to allow him to select the manner in which the process is to be terminated.

The program itself begins in Figure 5-7, which shows the initialization section. All we do here is OPEN the I/O channel and define the various constants used in the program.

```
10   REM OPEN the channel as in Section 2.5.2.3
20   REM
30   OPEN "com1:9600,n,8,,cs,ds,cd" AS #1
40   REM
50   REM Define the Control Constants
60   REM
70   ESC = 27                      :REM The ESC Code
80   TOUCH$ = CHR$(17)             :REM Touch Enable
90   DOWN$ = CHR$(10)              :REM Cursor Down
100  ECHO$ = CHR$(ESC)+"R2"        :REM Screen Echo Command
110  KEYMODE$ = CHR$(ESC)+"T1"     :REM Keyboard Mode Command
120  VPERASE$ = CHR$(ESC)+"E0"     :REM Screen Erase Command
130  VPHOME$ = CHR$(ESC)+"H"       :REM Cursor Home Command
140  WORKPG0$ = CHR$(ESC)+"W00"    :REM Select Page 00 for Work
150  WORKPG1$ = CHR$(ESC)+"W01"    :REM Select Page 01 for Work
160  WORKPG2$ = CHR$(ESC)+"W02"    :REM Select Page 02 for Work
170  DISPPG0$ = CHR$(ESC)+"D00"    :REM Select Page 00 for Display
180  DISPPG1$ = CHR$(ESC)+"D01"    :REM Select Page 01 for Display
190  DISPPG2$ = CHR$(ESC)+"D02"    :REM Select Page 02 for Display
```

Figure 5-7
The Initialization Section

The main body of the program begins at line number 200, and is shown in Figure 5-8. The GOSUBs in lines 230 through 250 are used to set up the actual display screens. Page 00 will contain the Master Control screen, Page 01 the Mode Select, and Page 02 the Shutdown Mode. These subroutines are shown in Figure 5-9.

Once the page memories have been loaded, the code starting at line number 290 displays the pages as needed and handles the operator's touch inputs.

The important thing to notice here is how little code and computer time is needed to display and process a display screen once it has been stored in the VuePoint's™ page memory. Only a handful of statements are needed to do the job, since the time consuming data transfers are needed once at the beginning of the program. In comparison, a traditional display would require the data to be retransmitted each time the display is changed, greatly increasing the system overhead. In a simplistic example such as this, that may not make much difference; however, the value quickly becomes apparent in a real application with scores of different screen formats.

```
200  REM
210  REM Set up the Various Pages
220  REM
230  REM GOSUB 700                :REM Send the Page 00 Text
240  REM GOSUB 890                :REM Send the Page 01 Text
250  REM GOSUB 1000               :REM Send the Page 02 Text
260  REM
270  REM The Main Program Loop
280  REM
290  PRINT #1, DISPPG0$           :REM Display Page 00
300  PRINT #1, TOUCH$             :REM Enable a Touch
310  INPUT #1, ANSWER$            :REM Get a Touch
320  IF ANSWER$ = "P: GOTO 360    :REM ON was Touched
330  IF ANSWER$ = "q" GOTO 510    :REM OFF was Touched
340  GOTO 300                     :REM Bad Answer; Try Again
350  REM
360  PRINT #1, DISPPG1$           :REM Now Display Page 01
370  PRINT #1, TOUCH$             :REM And Get a Touch from It
380  INPUT #1, ANSWER$
390  IF ANSWER$ = "r" GOTO 420    :REM MANUAL was Touched
400  IF ANSWER$ = "s" GOTO 460    :REM AUTO was Touched
410  GOTO 370
420  REM
430  REM The Manual Mode Control Code would go here
440  REM
450  GOTO 290
460  REM
470  REM The Automatic Mode Control Code would go here
480  REM
490  GOTO 290
500  REM
510 PRINT #1, DISPPG2$           :REM Display Page 02
520  PRINT #1, TOUCH$             :REM And Get a Touch from It
530  INPUT #1, ANSWER$
540  IF ANSWER$ = "t" GOTO 570    :REM ABORT was touched
550  IF ANSWER$ = "u" GOTO 610    :REM NORMAL was touched
560  GOTO 520
570  REM
580  REM The Abort Shutdown Code goes here
590  REM
600  GOTO 290
610  REM
620  REM The Normal Shutdown Code goes here
630  REM
640  GOTO 290
650  REM
```

Figure 5-8

The Main Body Code

```
660  REM The Subroutines
670  REM
680  REM Page 00 Text
690  REM
700  REM PRINT #1, WORKPG0$       :REM Page 00 for Work
710  GOSUB 780                    :REM Set initial parameters
720  PRINT #1, "MASTER CONTROL"
730  PRINT #1, DOWN$+DOWN$
740  PRINT #1, "ON: p"+DOWN$      :REM ON Button is "p"
750  PRINT #1, DOWN$
760  PRINT #1, "OFF:     q"+DOWN$:REM OFF button is "q"
770  RETURN
780  REM
790  REM Initialize the Current Working Page
800  REM
810  PRINT #1,  VPERASE$          :REM Erase It
820  PRINT #1,  VPHOME$           :REM Cursor Home
830  PRINT #1,  ECHO$             :REM Touch to Screen Echo
840  PRINT #1,  KEYMODE$          :REM Touch to Keyboard Mode
850  RETURN
860  REM
870  REM Page 01 Text
880  REM
890  PRINT #1,  WORKPG1$          :REM Page 01 for Work
900  GOSUB 780                    :REM Set Initial Parameters
910  PRINT #1, "SELECT OPERATING MODE"
920  PRINT #1, DOWN$+DOWN$
930  PRINT #1,  "MANUAL: r"+DOWN$:REM Manual is "r"
940  PRINT #1,  DOWN$
950  PRINT #1,  "AUTO:   s"+DOWN$:REM Automatic is "s"
960  RETURN
970  REM
980  REM Page 02 Text
990  REM
1000 PRINT #1, WORKPG2$           :REM Page 02 for Work
1010 GOSUB 780
1020 PRINT #1,  "SELECT SHUTDOWN MODE"
1030 PRINT #1, DOWN$+DOWN$
1040 PRINT #1,  "ABORT: t"+DOWN$  :REM Abort is "t"
1050 PRINT #1, DOWN$
1060 PRINT #1,  "NORMAL: u"+DOWN$:REM Normal is "u"
1070 RETURN
```

Figure 5-9
The Page Setup Subroutines

Looking at the subroutines in Figure 5-9, note that each starts by selecting the page to be set up by selecting it as the Working Page. This is followed by a call to another routine which erases the page, sets the desired mode, etc.

Once the page has been erased clean, the desired text is then sent. Note that the lower case letters p, q, r, etc. are used for the buttons, giving us the common button picture but unique button codes we discussed in the Touch Input Section. Each of the three pages has been set up in exactly the same manner.

An additional point worth noting is that these routines, as well as the remainder of the program, were coded for ease of understanding and not maximum efficiency. For example, the PRINT statements in lines 810 through 840 could all be combined into one, eliminating the need for a separate routine at all (**`PRINT #1, VPERASE$+VPHOME$+ECHO$+KEYMODE$`**). In a similar manner, most of the PRINT statements in the program could have used concatenation to reduce the number of statements required dramatically. However, it's a lot easier to see just what's being done when it's shown one operation at a time.

Once you've tried the above program and feel comfortable with its operation, you'll have mastered the fundamentals of using all of the major capabilities of VuePoint™: Text Display, Touch Input, and the use of Pages to save your screens. We can now go forward and take a look at the advanced features associated with these areas, seeing how they can be used to enhance the quality of the display, improve the operator interface and simplify the implementation of large applications.

6.0 Advanced Display Features 56

Up to this point, VuePoint™ III has been treated as a basic "upper case alphanumerics plus touchbutton pictures only" display, since that was all that was needed to illustrate the basic operating principles. In this section, we'll take a look at a group of additional features that greatly increase the human factors aspect of data display.

6.1 Character Set Selection

The VuePoint™ III contains two different character sets that may be used when displaying text. The characters which make up these sets may be seen in Figures 2-3 and 2-4, where we briefly looked at them during the initial VuePoint™ installation.

The *Standard Character Set*, which is the one we have been using so far, contains only the upper case of the alphabetic characters; the lower case ASCII codes are used to generate special graphics pictures such as the touchbuttons we discussed in the Touch Input section. To display the lower case alphabetic characters instead, the *Alternate Character Set* must be selected.

Character set selection is performed with the ESC sequence command ESC S C d, where d is a 0 to select the Standard Set and a 1 to select the Alternate Set. To see how it operates, try the following sequence, assuming an erased screen with the cursor at the home position:

1) Select the Standard Set via ESC S C 0, with no spaces separating the characters, and type the following on the VT-100:

```
A B C D E
a b c d e
```

2) Now select the Alternate Set via ESC S C 1 and retype the same two lines. Your screen should look as shown in Figure 6-1. You can see the differences between the two sets in the lower case characters.

Since the VuePoint™ will accept ESC commands at any time, you can intermix the characters from the two sets at will. For example, you might use the Alternate Set to label a series of touchbuttons, allowing upper and lower case labels, and then select the Standard Set to draw the actual touchbutton figures. You'll also notice that the character set selection is a mode change; that is, whatever is selected is used until it is explicitly changed.

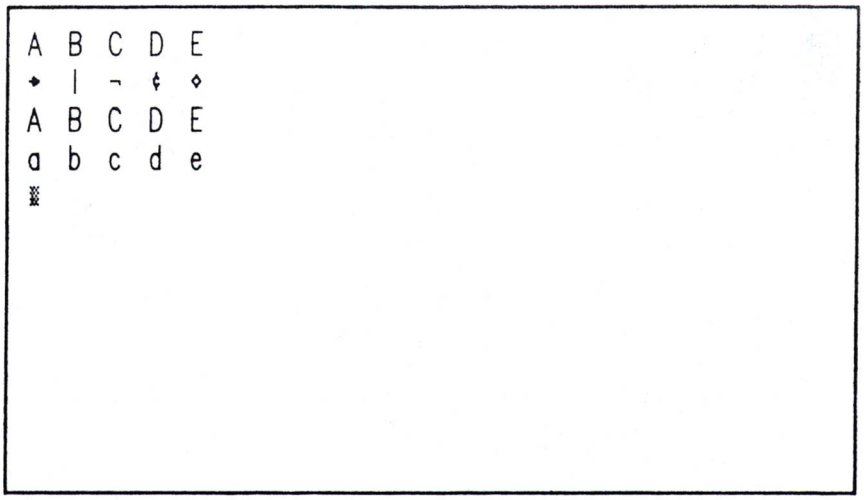


Figure 6-1
Character Set Selection

6.2 Character Blink

All VuePoint™ characters can be displayed as blinking (like the cursor) images in addition to the normal steady display. This capability is most useful when you are trying to get the operator's attention, such as in the case of an alarm message.

Character blinking is controlled by the command ESC S B d, where d can be any of the four values shown in Table 6-1. Note that like character set selection, blink is an attribute that remains in effect until explicitly changed.

D VALUE	OPERATION
0	No Blink (Normal)
1	Blink On, Low Rate
2	Blink On, High Rate
3	Make the Text Invisible

Table 6-1
Blink Attributes

The operation of the first three choices is pretty clear from the description, and if you try intermixing a few blink control commands with text you can see just how they operate. But the fourth–invisible characters–warrants a few extra words, for it's not intuitively obvious why you'd want to display something you can't see!

If you think about the operation of screen echo touch input, however, a very useful application can be seen. Assume you want to display a series of touchbuttons on the screen using the lower case p, q, r, etc. as we did in the Touch Input Section. In addition, since the buttons can be a little small to a large hand in an even bigger glove, you want to provide some margin for touch positioning error without cluttering up

the display. One method might be to use the following for a button, where the underscored characters (all but the middle p) are displayed invisibly:

p_p_p
p_p_p
p_p_p

Now you have a touchbutton that is three times wider and taller than normal, in terms of sensing space, but only visible as the single center button character on the screen.

6.3 Character Intensity

The brightness with which a character is displayed is also variable with VuePoint™ III, allowing you to mix both intensity and blinking to emphasize or de-emphasize particular sections of the screen. Like both character set selection and blink, this too is a mode type command which will stay active until explicitly changed.

The ESC sequence for setting the intensity is ESC S I d, where d is a 0 for normal intensity and a 1 for low intensity. Try intermixing a few ESC S I d commands with text being sent to the VuePoint™ and you'll readily see the difference in brightness between the two intensity levels.

6.4 Cursor Positioning

In the paragraphs above, we looked at several character attributes, as things like *blink* are called, that make it possible for us to design much more attractive and useful display screens. Now we'll take a look at some additional cursor control commands that make it easier to send those screen designs to the terminal.

6.4.1 Cursor On/Off Control

In Section 3.0, the VuePoint™ Power-on Menu was used to enable the display of the cursor on the screen. There is also an ESC sequence command that can be used for that purpose, allowing you to change the cursor state from an application program.

The general form of the command is ESC K d, where d is 0 to turn the cursor display off and 1 to turn it on. Note that since the command will be received by the current Working Page in VuePoint™, you won't see the effect of the command until you actually display that page.

6.4.2 Position Cursor

Up to this point, we've considered the cursor to be something that could only be positioned in relative terms. Though we could use ESC H to home it, all other movement was via CTRL codes (see Table 3-1) which moved it relative to its current position. While this will indeed allow you to send a complex screen design to VuePoint™, the job can be simplified greatly by using absolute positioning to place the cursor–and therefore the text which follows it–in exactly the row and column you want.

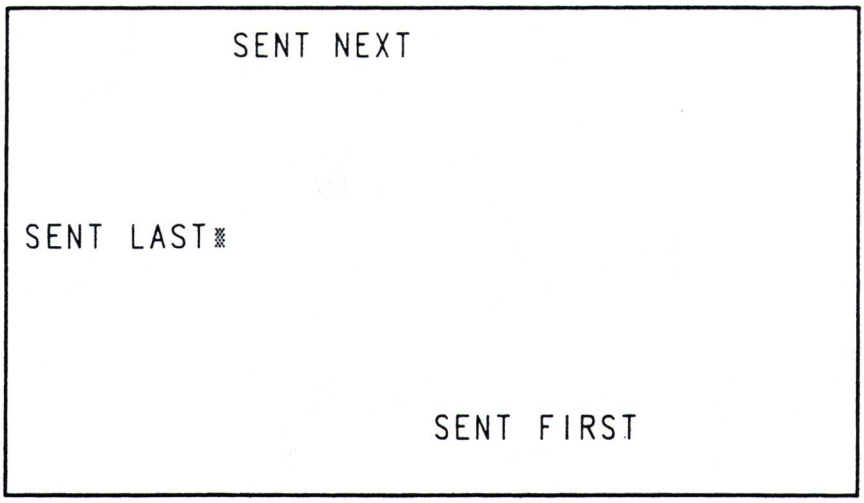


Figure 6-2
Cursor Positioning

The general format of the command is ESC P rr cc, where rr and cc are the desired row and column numbers. Note that you must **always** use two digits for each (1 must be 01, etc.), and that the rows and columns are numbered starting at 00.

Figure 6-2 shows how this can be used. ESC P 10 20 was sent prior to SENT FIRST, ESC P 00 10 prior to SENT NEXT, and ESC P 05 00 before the text SENT LAST. This is clearly a lot less work than using a series of CTRL codes to move the cursor around until you've got it in the desired location.

6.4.3 Cursor Advance

The VuePoint™ III offers four different methods of handling the cursor when the end of a row on the screen is reached. The desired choice is set via ESC A d, where d can be one of the four values shown in Table 6-2.

Note that there is no "right" method to select, for it depends upon the application and the message format transmitted by the host device. In general, though, a d value of 1 or 3 will probably be the one you'll use.

D VALUE	OPERATION
0	No Auto RETURN or LINE FEED. Upon reaching the end of a row, the cursor stays in the last column of that row. If additional characters are sent they are stored one top of the other in that one position.
1	Auto RETURN, but no Auto LINE FEED. The cursor returns to the first column of the same row, storing any new text that is sent on top of the text that is currently there.
2	No Auto RETURN, but Auto LINE FEED. The cursor moves to the last column of the next row.
3	Auto RETURN and Auto LINE FEED. The cursor moves to the first column of the next row.

Table 6-2
Cursor Advance Commands

6.4.4 Screen Scrolling

Related to how the cursor advances from one row to the next is how the cursor behaves when it reaches the end of the page.

Normally, the VuePoint™ operates in what is call *Block Mode*. Each page of text is considered to be a solid block of memory, and when the cursor gets to the end it starts over again at the beginning.

When you're building up pages of text to be displayed at some future time, this is probably the best mode to use. But when you're displaying text in real time–the Working Page and the Display Page are the same–*Scrolling Mode* is usually the better choice.

There is no visible difference between these two modes until the cursor reaches the very last column of the last row (rr,cc of 11,39). If you are operating in Block Mode at that time, the cursor will return to 00,00 and any text that follows will be written over the current contents of the page.

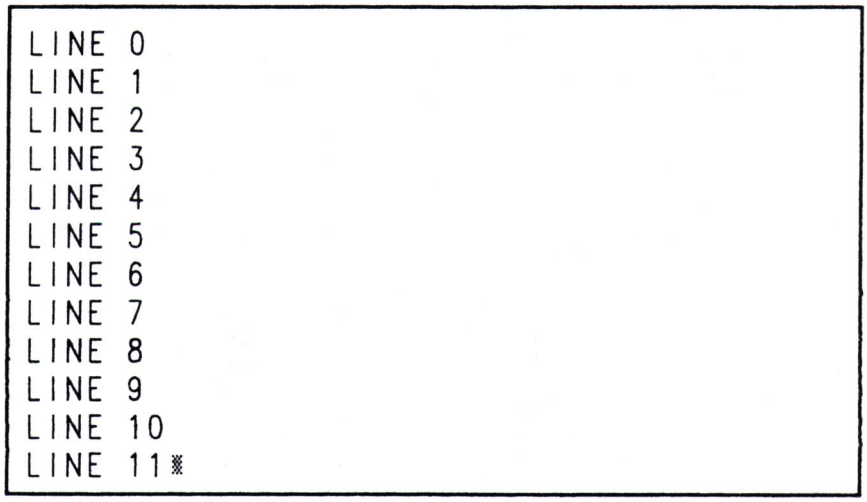


Figure 6-3
Screen Before Scrolling

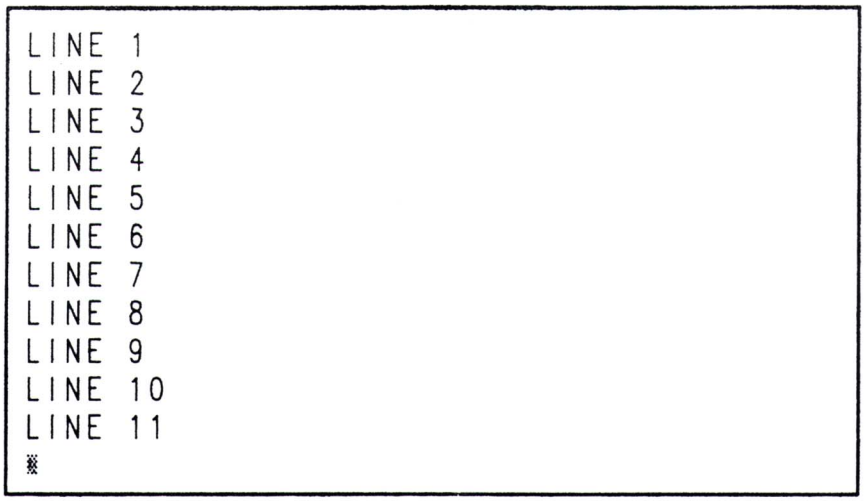


Figure 6-4
Screen After Scrolling

Scroll Mode, however, scrolls the text (and all the associated attributes of that text) upward one row, leaving the cursor at the 11,00 position and the last row of the screen blank. Note that though the contents of the first row are lost in the process, it does insure that each new row of text the operator sees is written on a clean row of the VuePoint™. Figures 6-3 and 6-4 show how this looks on the screen.

The command ESC B d is used to select Block vs. Scroll Mode operation, with d a 0 for Block and a 1 for Scroll.

6.4.5 Setting and Clearing Tab Stops

The general manner in which tabs are set and cleared is to first position the cursor to an appropriate rr,cc location and then issue the command ESC Y d, where the value of d selects the exact operation to be performed. Table 6-3 shows the various values for d and how they are used.

Note that setting and using tabs is a two step operation. First you have to move the cursor about and set the tabs. Then you must send the text (with embedded TAB - CTRL/I - characters) to take advantage of them. While this may sound a little cumbersome right now, it can be very useful when used in combination with some of the advanced paging functions we'll cover in Section 8.0.

D VALUE	OPERATION
0	Clear the Tab Stop where the cursor is positioned.
1	Set a Tab Stop where the cursor is positioned.
2	In each row, clear the Tab Stop in the column in which the cursor is currently positioned.
3	In each row, set a Tab Stop in the column in which the cursor is currently positioned.
4	Clear all Tab Stops on the Page.

Table 6-3
Tab Control Commands

6.4.6 Right to Left Entry

For most applications, normal left to right data entry and display will be all you'll ever see. But if the VuePoint™ III is being used in an application that is accounting oriented, the ability to enter numbers in a right to left direction–like many adding machines and calculators–can be beneficial.

To enter data in this manner, the first step is to position the cursor to the rightmost column of the display field that is to be filled from right to left. Next you issue the command ESC L ww, where ww defines the number of character positions (including the cursor position) in the field. Note that like all the other "ww" commands we've used, if you reach the edge of the row first–in this case the left edge–that, rather than the ww count, terminates the field.

Once you've performed these steps, the text that is sent by the host will be stored into the field from the right rather than the left. Note that since you have defined a specific field width, if you send more characters than there are positions in the field, only the last ww number of characters will be stored; the earlier characters will be shifted in from the right edge of the field, out from the left edge, and lost.

Since this too is a mode, Right to Left Entry will stay enabled until it is specifically disabled. This is done by sending the same command with ww equal to 00.

6.5 Modifying Character Attributes

By combining the cursor positioning commands we just covered with the character attribute material discussed in the first few sections of this chapter, we can take a look at how to modify the attributes of text characters that have already been stored in the VuePoint's™ memory.

6.5.1 Modify Blink

The ESC S B d command, you'll recall, was used to set the blink attribute of text characters as they were initially sent to the VuePoint™, with the value of d determining the type of blinking that was to be used. To change the blink attribute of text that has already been stored, we again use that same d value. But in addition, since we are changing something that already exists, we have tell the VuePoint™ just exactly which text characters we want to change.

The first step in the procedure is to move the cursor using the ESC P rr cc command to position the cursor to the row and column location of the first character whose blink attribute is to be modified. Once you've done that, you then use the ESC M B d ww command to actually modify the attribute.

d is the attribute value just as before, and is defined in Table 6-1. To this is added the argument ww, which tells VuePoint™ exactly how many characters–beginning with the character at the current cursor position–are to have their blink attribute altered. Once this command is received, the VuePoint™ will start changing the blink attribute until the ww count is satisfied *or* the end of the current row is reached, whichever occurs first.

6.5.2 Modify Intensity

The ESC M I d ww sequence is used to alter the brightness attribute of a series of characters in a similar manner. If d is 0, the normal intensity will be used; if it is a 1, low intensity. ww is again the number of characters to be modified, starting at the current cursor position. Note that the end of the row will also terminate this operation if it is reached before the ww count is satisfied.

6.6 Text Protection

Operationally, any text that is protected is guarded against erasure (ESC E 0 command) and overwriting, a concept that is pretty easy to understand. It's application, however, is a little less obvious unless you are working in an environment where you have many different pages of text to display. In that case, you may want to protect a page title, fixed touchbutton area, or the like from overwriting or erasure as other text is displayed and erased.

6.6.1 Set Protection

Setting protection On and Off operates like blink and intensity; an ESC command turns the attribute on, and all text following it is protected until it is turned off.

To see how it works, try the following sequence:

1) Erase the screen and home the cursor via ESC E 0 and ESC H.
2) Type in NOT PROTECTED followed by a RETURN.
3) Issue the command ESC S P 1, which turns protection on.
4) TYPE PROTECTED, then ESC S P 0 to turn protection off.
5) Lastly, type a RETURN followed by NOT PROTECTED again.

The screen on your VuePoint™ will now look as shown in Figure 6-5. Try erasing the display with ESC E 0, and what happens? The PROTECTED text is still there, as you can see in Figure 6-6. Move the cursor to the start of that text and try typing on top of it and see what happens. That message is there to stay! The only way to erase it is to turn off the VuePoint™ power or modify the protection of those characters.

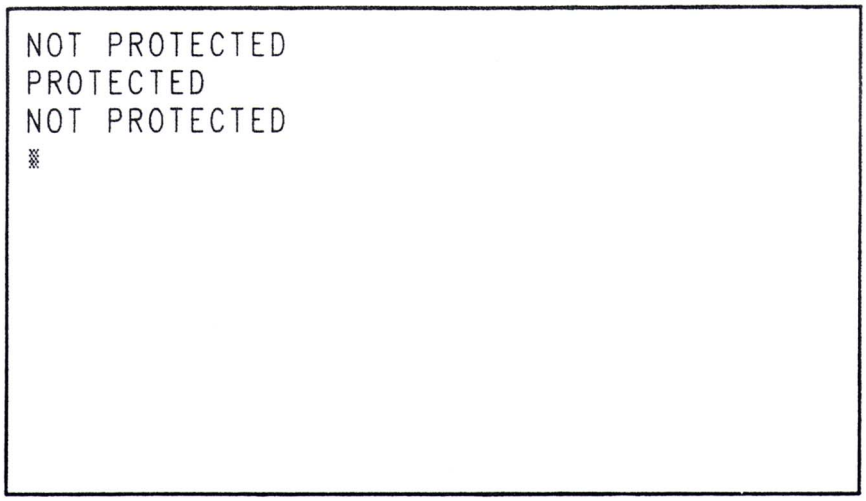


Figure 6-5
Protected Text Before Erasure

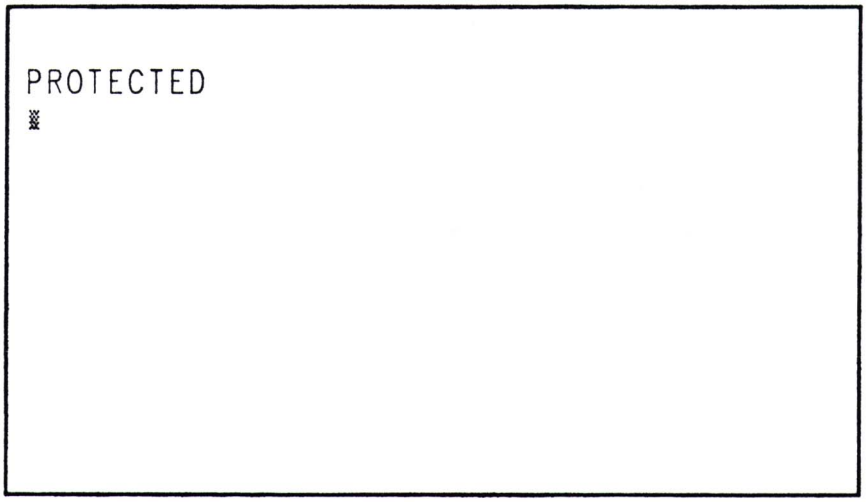


Figure 6-6
Protected Text After Erasure

6.6.2 Modify Protection

To modify the protection attribute of a text area, the same basic procedure that was used for blink and intensity is used.

First move the cursor to the beginning of the area to be modified using ESC P rr cc. Then the command ESC M P d ww is used to change the protection. d can be 0 or 1, for off or on, and ww is a character count, just as before. Note that the "end of row" rule applies here, also.

6.7 Reset

The last of the primarily display oriented commands is the one that is used to bring the VuePoint™ back to a known state without cycling the power off and then on again. With the exception of clearing the memory, that is what the *Reset* command does, but only for the current Working Page.

To reset a page, you send the sequence ESC X d, where d determines the type of reset to be performed. The values for d and their operation can be seen in Table 6-4.

D VALUE	OPERATION	
0	The following attributes for the current Working Page are set as shown:	
	Data Entry Mode:	Normal Left to Right
	Cursor Control:	Auto RETURN and Auto LINE FEED
	Cursor Position:	Home
	Touch Response:	Full Screen, Row/Column Reporting
	Scrolling:	Off (Block Mode On)
	Cursor Display:	Off (Not Displayed)
	Touch Mode:	Push Button
1	The default conditions for all the character attributes are invoked, resulting in the following:	
	Intensity:	Full
	Blink:	None, with all characters visible
	Character Set:	Standard
	Protection:	None
	Touch Sensitive:	No

Table 6-4
Reset Operation

All of the VuePoint™ III features that are display oriented have been discussed at this time. Though some of them are difficult to relate to out of context, if you continue on, we'll cover the advanced touch and paging features and show how these various characteristics are interrelated.

7.0 Advanced Touch Features 66

In this section, we'll look at three VuePoint™ III commands designed to improve and simplify the use of touch input. We'll then apply those commands, plus some of the display features covered in the last section, to build a BASICA program that duplicates a simple numeric keypad on the VuePoint™ screen.

7.1 Selective Touch Sensitivity

Selective touch sensitivity is the ability to make only specified areas of the VuePoint's™ screen touch sensitive, something that both simplifies the programming of the host computer and makes the system easier to operate.

Assuming that only the "legal" button areas of the screen are defined to be touch sensitive, the host program is simplified since it no longer has to accept and process invalid or meaningless touches. In a similar manner, the operator can also immediately tell if he or she has touched a legitimate button, for the VuePoint™ will only "beep" when a sensitized area is touched. A touch at any other point will simply be ignored.

7.1.1 Set Touch Sensitivity

Internal to the VuePoint™, touch sensitivity is a character attribute just like blink and intensity, though since it's not a visible attribute, it wasn't covered at the same time as they were. Since it is an attribute, the process to selectively sensitize or desensitize an area will look familiar:

1) Move the cursor to the desired rr,cc for the start of the text you wish to send.
2) Issue the command ESC S S d, where d is a 0 for insensitive and a 1 for touch sensitive.
3) Send the text. As each character is stored, the touch attribute will be set as specified.

Note that this is a mode type operation, as were all of the other attribute setting commands; the mode selected by d will remain active until you explicitly change it.

Since the default setting for this attribute is Off, the normal sequence of events is to turn it on via ESC S S 1, send a touchbutton graphic character, and then turn it off with ESC S S 0.

7.1.2 Modify Touch Sensitivity

As you would expect, since touch sensitivity is really a character attribute, it has a modify command just like all the others. And it operates in the very same way.

First, the cursor is positioned to the start of the text to be changed. Next, the command ESC M S d ww is issued. d is the same parameter that was used for setting the attribute, and ww is the familiar character count. Don't forget that end of row will also terminate the operation if it is encountered before the count is satisfied.

7.2 Touch Response Modes

Once the desired areas of a page have been sensitized, the next step is to set the VuePoint™ to the corresponding touch mode.

If you recall, we looked at a portion of this commanding Section 4.4 during the discussion on row/column vs. screen echo touch operation. In addition, the touch mode command can take advantage of the touch sensitivity attribute.

The general form of the command is ESC R d, where d is a parameter that determines exactly what the command will do. The allowable values for d and their operation can be seen in Table 7-1 below.

Note that only d values of 1 and 3 utilize the touch sensitivity attribute; if 0 or 2 is used, the attribute is ignored and the entire screen is made sensitive.

D VALUE	OPERATION	TOUCH MODE
0	Full Screen Response	Row/Column Reporting
1	Sensitized Characters Only	Row/Column Reporting
2	Full Screen Response	Screen Echo Reporting
3	Sensitized Characters Only	Screen Echo Reporting

Table 7-1
Response Mode Parameters

7.3 Using Selective Touch

Since selective touch makes the VuePoint™ both easier to program and easier to use, most your programs will use that mode of input. To see how it's done, an example program is shown in Listings 7-1 and 7-2.

Looking at Listing 7-1, you'll see that almost half of it–lines 10 through 200–are used to set up the constants used in the remaining lines. In lines 240 and 250, you'll find the VuePoint™ initialization statements. By using string concatenation, seven different commands are sent with only two PRINT statements. The end result is that Page 00 has been selected for both Display and Work, the screen is erased, the cursor homed, touch set to keyboard operation, selective sensitivity, screen echo input, and right to left entry disabled.

Next, a picture of a numeric keypad is drawn, with the touch attribute set on for only those characters. Note that the characters have been placed in even column numbers, for only those positions have touch sensitivity enabled.

The operator is then prompted for an input in line 380, with blink being used to get his attention and let him know it's time to enter a value. (Figure 7-1 shows how the VuePoint™ screen will look at this point.) Line 390 then sets up a right to left data entry field for use in displaying the data the operator will enter. Note the use of the semicolon at the end of the statement. This prevents BASICA from sending a RETURN after the text, insuring that the cursor remains where we put it.

In Listing 7-2, you can see how touch input is enabled and the input processed. If a C is received, it is treated as a CLEAR key and the display field set to all blanks in line 490.

An E (for EXIT) displays a DONE message at line 510 and terminates the program. Any other value, which must be one of the digits, is sent to the display for the operator to see.

If you try the program, you'll agree that selective touch is the best way to handle most touch input, and is particularly valuable in keyboard entry emulation such as this.

```
10   REM OPEN the channel as in Section 2.5.2.3
20   REM
30   OPEN "com1:9600,n,8,,cs,ds,cd" AS #1
40   REM
50   REM Define the Control Constants
60   REM
70   ESC = 27                    :REM The ESC Code
80   TOUCH$ = CHR$(17)           :REM Touch Enable
90   ECHO$ = CHR$(ESC)+"R3"      :REM Screen Echo+Selective Cmnd
100  KEYMODE$ = CHR$(ESC)+"T1"   :REM Keyboard Mode Command
110  VPERASE$ = CHR$(ESC)+"E0"   :REM Screen Erase Command
120  VPHOME$ = CHR$(ESC)+"H"     :REM Cursor Home Command
130  CURSOR$ = CHR$(ESC)+"P"     :REM Cursor Position Prefix
140  TCHON$ = CHR$(ESC)+"SS1"    :REM Touch Attribute On Command
150  TCHOFF$ = CHR$(ESC)+"SS0"   :REM Touch Attribute Off Command
160  BLKON$ = CHR$(ESC)+"SB1"    :REM Blink On Command
170  BLKOFF$ = CHR$(ESC)+"SB0"   :REM Blink Off Command
180  RTOL$ = CHR$(ESC)+"L"       :REM Right to Left Command
190  WORKPG0$ = CHR$(ESC)+"W00"  :REM Select Page 00 for Wrkg Pg
200  DISPPG0$ = CHR$(ESC)+"D00"  :REM Select Page 00 for Dsply Pg
210  REM
220  REM Initialize the VuePoint III
230  REM
240  PRINT #1,WORKPG0$+DISPPG0$+VPERASE$
250  Print #1, VPHOME$+ECHO$+KEYMODE$+RTOL$+"00"
260  REM
270  REM Draw the Keypad
280  REM
290  PRINT #1, TCHON$            :REM Turn Touch Attribute On
300  PRINT #1,CURSOR$+"0534"+"7 8 9"
310  PRINT #1,CURSOR$+"0734"+"4 5 6"
320  PRINT #1,CURSOR$+"0934"+"1 2 3"
330  PRINT #1,CURSOR$+"1134"+"C 0 E"
340  PRINT #1,TCHOFF$            :REM Turn Touch Attribute Off
350  REM
360  REM Set Up Right to Left Input Area
370  REM
380  PRINT #1,CURSOR$+"0134"+BLKON$+"VALUE?"+BLKOFF$
390  PRINT #1,CURSOR$+"0339"+RTOL$+"06";
```

Listing 7-1
Selective Touch Setup

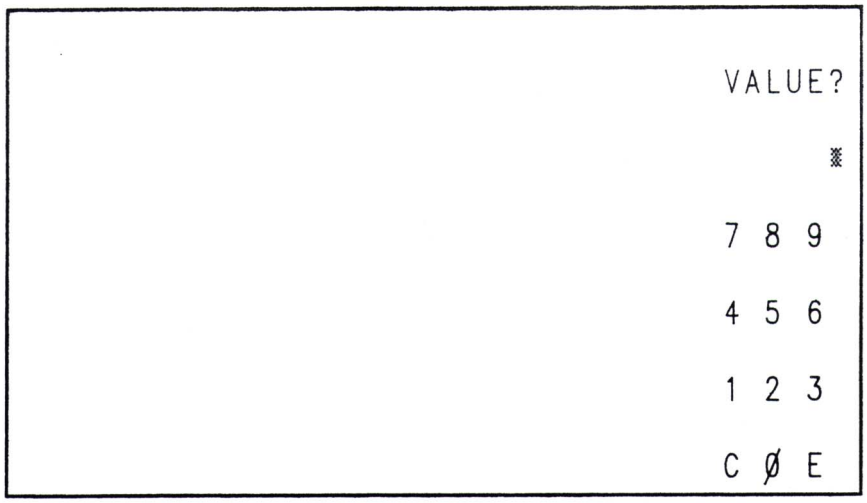


Figure 7-1
Numeric Keypad Display

```
400  REM
410  REM Enable and Accept a Touch Input
420  REM
430  PRINT #1,TOUCH$;
440  INPUT #1,TCHIN$
450  IF TCHIN$ = "C" GOTO 490     :REM C means Clear
460  IF TCHIN$ = "E" GOTO 510     :REM E means Exit
470  PRINT #1,TCHIN$;             :REM Else Display the Digit
480  GOTO 430
490  PRINT #1,"      ";           :REM Clear the Display Area
500  GOTO 430
510  PRINT #1," DONE ";
520  END
```

Listing 7-2
Numeric Keypad Input Processing

8.0 Advanced Paging Features 70

In the previous sections, we considered the VuePoint's™ page memory simply as a convenient means of storing displays until they were needed. Now we'll take a look at a series of commands that allow us to manipulate those stored pages in ways that greatly increase the versatility of the page memory facility.

8.1 Copy Page

The command ESC C pp is used to copy the contents of a page number pp into the currently selected Working Page. All text and text attributes are copied just as if they had been transmitted by the host processor, replacing completely the current contents of the Working Page. For example, if the Working Page is set to 00 by ESC W 00 and the command ESC C 01 issued, the contents of page 01 will be duplicated into page 00, making both pages identical.

From an application point of view, *Copy Page* is used primarily to generate "fill in the blank" screens on the VuePoint™. Saving frequently used report titles and column headings on individual pages, for example, allows the host to save a lot of data transmission time by using Copy Page to move the title, etc. to the current Display Page and then transmitting only the data values needed for the report.

8.2 Merge Page

As the name implies, the *Merge Page* command allow you to combine the contents of two pages into one. This is performed by the command ESC m pp, where the lower case m requests the merge and pp specifies the page from which the new data will come.

When the command is executed, the VuePoint™ treats each character in the following manner:

1) The character from page pp is examined to see if it is a displayable character (not a blank). If it is not displayable, it is bypassed and the next character in sequence examined.

2) Each time a displayable character is found in the sending page, the corresponding character position in the current Working Page is examined. If a blank is found, the character from the sending page–with its attribute–is stored in place of the blank.

3) If the character in the Working Page position was not a blank, it is left alone and the logic goes about examining the next character in sequence in the sending page.

This process is performed until every character in the sending page has been examined. The end result is that the current Working Page will contain a combination of the two pages, with the characters and attributes from the Working Page taking precedence whenever there is a conflict.

Though it may sound a little complicated, in operation it's very straightforward. For example, assume that page 00 contains the text shown in Figure 8-1, page 01 the text in Figure 8-2, and the following commands issued:

1) ESC W 01 to set the Working Page to 01.

2) ESC D 01 to set the Display Page to 01.

3) ESC m 00, to merge page 00 into page 01 (the current Working Page).

At the completion of step 3, page 01 will look as shown in Figure 8-3; the original text remains, but the spaces between the characters were filled with the text from page 00.

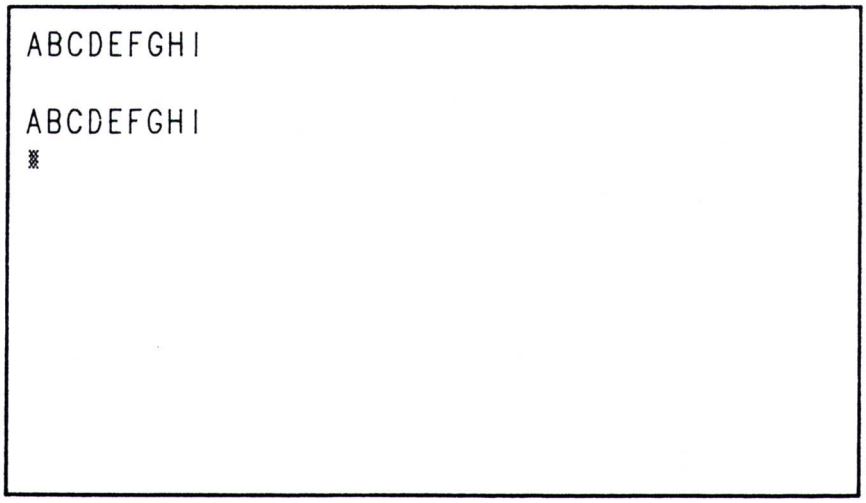


Figure 8-1
Page 00 Text Contents

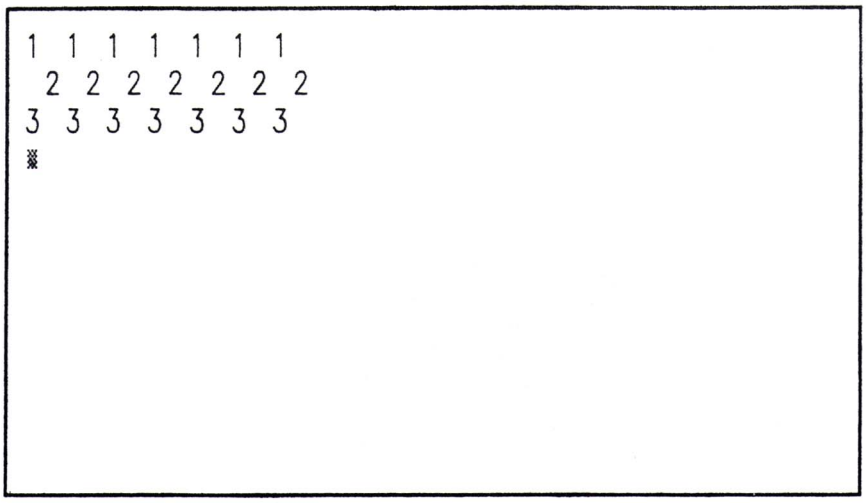


Figure 8-2
Page 01 Text Contents

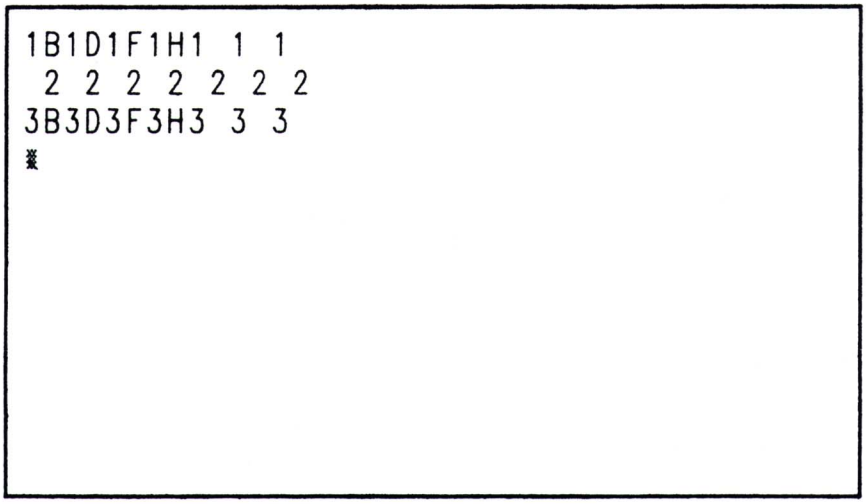


Figure 8-3
Page 01 After the Merge

To see how this might be applied, consider the situation where you have a dozen or so different displays, all of which may at some time need a numeric keypad for operator input. While you could include the keypad in all of the displays and only enable it when input is valid, this could be confusing to the operator, for sometimes the keypad would work and sometimes it wouldn't.

A better approach would be to store all of the displays, each on a separate page, without the keypad text. However, each display would leave the keypad area of the screen blank. In addition, one page would contain a keypad layout designed to fit into the blank area you left on the other pages, and another would be left blank and used as the Display Page.

To display a page without the keypad, the program would use Copy Page to move it to the page assigned for Display Page use. The operator would see the text less the keypad, getting the proper message and avoiding any confusion.

If at some later time that same message display required operator input, simply do the following:

1) Copy the message display to the Display Page.

2) Merge the numeric keypad display into the Display Page.

By simply issuing two escape sequence command, you've tailored the display to the situation, minimizing both operator confusion and the host workload.

8.3 Screen Control

Since a real application may require that several pages be copied and merged before the desired Display Page is finally generated, the various screen changes that will take place while this is going on may prove confusing to the operator. To avoid that situation, two new commands will be introduced.

8.3.1 Disable Screen

The command CTRL/R tells VuePoint™ to stop the transfer of information from the Display Page in memory to the physical display screen. Once the *Disable Screen* command is issued, the data in the Display Page can be freely modified with all of the updates remaining invisible to the operator.

To cancel this condition and re-enable the screen, issue either an ESC D pp, where pp would normally be the current Display Page number, or ESC N to display the next page in sequence.

8.3.2 Clear Screen

Disable screen, discussed in the previous section, stops the screen update but retains the current contents. While this may be appropriate, there will also be times when you want the screen to be blank while the updating is going on. In those cases, the CTRL/T *Clear Screen* command is used in conjunction with CTRL/R.

First, CTRL/R is used to prevent further updates. This is followed with CTRL/T to clear the text from the screen. When the updating is finally finished, either ESC D pp or ESC N can be used just as before to re-enable the screen.

8.4 Select User Page Bank

All of our paging examples to date have used just the three standard memory pages, but your standard configuration VuePoint™ III can be set up with as many as 128 pages. See Appendix C for instructions on increasing your page count. Additionally, you can order an increased memory capacity for your VuePoint™. Please speak with a Sales Engineer for more information.

Although the VuePoint™ III does not use page banks, the page bank commands are supported.

Note, however, that the number 128 has three digits and all of the page selection commands are limited to only two digits. Memory Bank Selection is the way we get around this apparent dilemma.

The following is excerpted from the VuePoint™ II User's Manual:

> *The VuePoint's™ available "Type B" page memory space is divided into multiple groups (or banks) of 64 pages each. That means that if you have more than 64 pages, you must first use Page Bank Selection to select the bank of interest. Then the two digit pp parameter in the various page commands selects the page number within the selected bank. It is important to note that the "Type B" pages are appended to the existing "Type A" pages.*
>
> *The command to select a memory bank is ESC Q d, where d is a 0 for the first bank of 64 pages and a 1 for the second. Upon power-on, bank 0 is always assumed, since most VuePoints™ have less than 64 pages. In addition, whenever a bank selection command is given, VuePoint™ automatically sets both the Working and Display Pages to page 00 of the selected bank.*

An important point to remember if you have a multiple bank system is that pages of data may not be copied or merged from one bank to another; each memory bank is a separate entity. While this may cause some confusion if you don't remember it, it's unlikely it will hinder you in implementing even the most complex of applications. It is possible, however, to copy or merge pages from a "Type B" memory page of one bank to a "Type A" page (i.e., 00, 01, …) and then copy to a "Type B" page of a different bank.

The VuePoint™ page memory is divided into two types. "Type A" memory consists of the 3 standard memory pages included with VuePoint™ and is expandable to 7 or 15 pages. These pages are always accessible. The VuePoint's™ available "Type B" memory space is divided into multiple groups (or banks) of 64 pages each. That means that if you have more than 64 pages of "Type B" memory, you must first use Page Bank Selection to select the bank of interest. Then the two digit pp parameter in the various page commands can select the page number within the selected bank. It is important to note that the first "Type B" page comes after the last "Type A" page. For example, if the VuePoint™ is ordered with 143 pages of memory, then in any bank, page 00 to 14 is "Type A" memory and page 15 to 78 is a bank of 64 "Type B" memory pages. Bank 0 contains the first 64 pages of "Type B" memory and Bank 1 contains the next 64 pages of "Type B" memory.

The command to select a memory bank is ESC Q d, where d is a 0 for the first bank of 64 "Type B" pages and 1 for the second. Upon power-on, bank 0 is always assumed. In addition, whenever a bank selection command is given, VuePoint™ automatically sets both the Working and Display Pages to 00.

It is possible, however, to copy or merge pages from a "Type B" memory page of one bank to a "Type A" page (i.e., 00, 01, …) and then copy a "Type B" page of a different bank.

9.0 Other VuePoint™ III Commands 75

This section describes other VuePoint™-oriented commands that may be useful to the system programmer.

9.1 Bell Control

The internal bell (really a beeper) that is sounded to acknowledge a touch input can also be controlled from the host. ESC G d is the sequence used to control the bell logic, where d is a 0 to disable the bell and a 1 to enable it. Note that if the bell is disabled, it cannot be sounded from the host and will not beep in response to a touch.

Assuming an enabled bell, the host can sound it at any time by sending the ASCII bell code (CTRL/G) to VuePoint™.

9.2 Power Fail Status/Reset

Included as a standard feature of VuePoint™, but used most frequently in conjunction with battery backed RAM, is the ability to test for a power failure and the attendant power-on initialization.

The command ESC F 1 is used to make the test. If the power-on initialization has occurred, VuePoint™ will respond with an ASCII NAK (Negative Acknowledgement; 15 Hex) character followed by a RETURN. Otherwise, the response will be an ASCII ACK (Positive Acknowledgement; 06 Hex) and a RETURN.

Note that these responses will only be sent if no other VuePoint™ responses (touch or keyboard) are in the process of being transmitted to the host. If one of these is in process, the ESC F 1 command will be ignored.

If an ACK is received, no further effort is required; you've verified that all is well and can continue on with your application.

A NAK response, however, must be acknowledged with the command ESC F 2. This command resets the power fail status indicator to the ACK state, where it will remain until the next power failure and power-on initialization occurs. If you don't acknowledge the NAK with ESC F 2, there will be no way to sense and test for the next power-on initialization.

The command ESC F 3 can alternatively be used to make the test. If the power-on initialization has occurred, VuePoint™ will respond with an ASCII 0 (Negative Acknowledgement; 30 Hex) character followed by a RETURN. Otherwise, the response will be an ASCII 1 (Positive Acknowledgement; 31 Hex) and a RETURN.

9.3 Power-on Reset

This command provides a full power-on firmware reset. When ESC Z is issued, the VuePoint™will begin its initialization procedure which includes clearing of all RAM pages (except for battery backed pages), initialization of all page and VuePoint™ specific modes, as well as installed hardware.

If the power-on setup menu is enabled, it will appear on the screen and will be waiting for a touch input.

GENERAL DIGITAL™ CORPORATION

(hereinafter Seller)

STANDARD TERMS AND CONDITIONS OF SALES

1. Acknowledgment and acceptance of any order by seller is expressly limited to and made conditional upon the terms and conditions contained herein. Any of the buyer's terms and conditions which are in addition to or different from those contained herein and which are not separately agreed to in writing (except additional provisions specifying quality, character of the products ordered, and shipping instructions) are hereby objected to and shall be of no effect. The buyer shall be deemed to have assented to all terms and conditions contained herein if any part of the products described herein is accepted.

2. ENTIRE AGREEMENT. This instrument contains the entire and only agreement between the parties hereto relating to the subject matter hereof, and any representation, affirmation of fact, and course of prior dealings, promise or condition in connection, therewith, or usage of the trade not incorporated herein, shall not be binding on either party. The rights and obligations contained herein shall inure to and be bind upon the parties, their legal representatives, successors, and permitted assigns.

3. WARRANTY. Seller warrants that upon delivery to the Buyer, the product sold thereunder shall be free of defects in materials and workmanship and that such product shall substantially conform to the specifications provided by Seller. This Warranty commences on the date of delivery to the Buyer unless another time for commencement has been negotiated between Seller and Buyer prior to such delivery. Should the product be delivered through a third party (UPS, USPS, Federal Express, etc.) the warranty period shall commence on the date that such third party receives the product unless another time for commencement has been negotiated between Seller and Buyer prior to such delivery. The duration of the Warranty shall be one (1) year from the date of delivery as set forth above. This Warranty shall be effective only if Seller receives notice of such defects in materials and workmanship during the period of the Warranty stated above. After Seller receives such notice, Buyer shall obtain a Return Material Authorization (hereafter RMA) number. To obtain an RMA number, Buyer shall contact Seller's Customer Service Department, with the model number(s), serial number(s), and original purchase order number(s) of the product or products. Buyer shall include a brief description of the problem or failure mode experienced. Buyer shall mark the outside of the package containing the product or products, conspicuously, with the letters RMA and the number assigned by Seller. Any packaged returned without RMA and the RMA number clearly visible will be rejected at Seller's receiving dock. After Buyer has provided notice and obtained the appropriate RMA number, Buyer shall return the product, upon request, to General Digital™ Corporation, 8 Nutmeg Road South, South Windsor, CT 06074, not later than thirty (30) days after Seller's receipt of notice of the alleged problem or failure mode. Buyer shall prepay all transportation charges for return of the product(s) to Seller. Seller shall pay for the cost of sending the product(s) back to Buyer. Transportation charges to Buyer shall be deemed Best Way and any other transportation charges shall be at the Buyer's cost. All costs of customs clearance, shipping, and any other related charges for transportation outside of the contiguous 48 United States shall be at the Buyer's expense. After Buyer's compliance, Seller shall, at Seller's option, repair or replace the product(s). Excluded from this Warranty are all problems or failures resulting from:

- Improper or inadequate maintenance of the product by the Buyer.
- Unauthorized modification of the product by any means.
- Operation of the product outside its environmental specifications.
- Neglect, misuse, or abuse of the product.
- Modification or integration with other components not covered by a Seller warranty when such integration increases the likelihood of problems, failures, or damage.

Further, this warranty is between seller and buyer only, and does not extend to buyer's customers or users of buyer's products. In addition, and in the event that buyer integrates seller's product with components manufactured or supplied by any other company, this warranty shall only relate to defects in materials and workmanship of seller's product and not to problems with or failures of seller's product caused by other components integrated with it. Seller's rendering of technical advise and/or assistance or Seller's performance of service work in connection with Buyer's order of the product(s) shall not change or enhance this warranty in any way. Buyer's reliance on Seller's skill or judgment to select or furnish suitable goods shall be at Buyer's risk. Seller shall not be liable for any direct, indirect, special incidental, or consequential damages including, but not limited to, loss of profits and/or destruction of other property, caused by any application of the product(s) and/or its integration with other components. Seller's liability shall be limited to the amount paid by the Buyer for the product(s). The aforementioned remedies are in lieu of and exclude all others. Buyer assumes all other risks and liabilities for loss, damages or injury, whether or not caused by Seller's negligence, to persons or property and arising out of, connected with, or resulting from, the use of Seller's products or services either alone or in combination with other products or services. Seller specifically excludes any and all other warranties, written or oral, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose.

4. SPECIAL PRODUCTS. The following provisions are to be considered a part of all Special Product Acknowledgments and Orders. Special Product orders are those calling for products not contained in Seller's current catalog and price list, or those requiring modifications to catalog products or those requiring sample, environmental, mechanical or life testing, 100% reliability screening, quality conformance qualifications, or any combination thereof. These provisions supersede any part of the other clauses relating to the same project.

A. Delivery dates are best estimates only and are subject to (1) Receipt by Seller of order and negotiated specifications containing, where applicable, all quoted waivers and/or exceptions, (2) successful, first time passage of products submitted to electrical performance test, to environmental and/or life test processing required by applicable specifications.

B. Seller assumes no responsibility for, refund or replacement of products shipped at the Buyer's request prior to successful completion of acceptance and/or qualification test performed by Seller, except those subject only to Buyer's incoming visual inspection.

5. PATENTS. Seller shall defend any suit or proceeding which is based on a claim that any products or part thereof, furnished under this contract constitutes an infringement of any valid United States patent, if notified promptly in writing and given authority, information and assistance for the defense or settlement of same. In case said product, or part thereof, is in such suit held to constitute infringement and the use of said product or part is enjoined, then Seller shall at its own

expense and at its option, either procure for the Buyer the right to continue using said product or part, or replace same with non-infringing comparable products, or modify it so it becomes non-infringing, or refund the purchase price thereof. The forgoing states the entire liability of Seller for patent infringement by said product or part thereof.

6. TERMS OF PAYMENT. Payment terms for sales within the USA are Net Thirty (30) days from date of shipment. International sales require a Letter of Credit on a US Bank prior to shipment from Seller.

7. TITLE AND DELIVERY. All shipments within the 48 contiguous United States by Seller shall be F.O.B. point of shipment. Title to hardware products and risk of loss pass to Buyer upon delivery to carrier at shipping point. Buyer shall pay all transportation and storage charges. Seller will select the carrier if Buyer does not. Claims for shortages must be made with Ten (10) days after receipt of the order.

8. CONTINGENCIES. Seller shall not be responsible for any failure to perform due to unforeseen circumstances or to causes beyond its control. Examples of such causes are acts of God, War, Riot, Embargoes, Acts of Civil or Military Authorities, Fire, Floods, Accidents, Strikes or Shortages of Transportation, Facilities, Fuel, Energy, Labor or Materials. Seller may defer delivery for a period equal to the delay caused by such contingency.

9. CANCELLATION.
A. Default. If Buyer defaults, Seller may decline to make further shipments and/or may terminate Buyer's order without affecting Seller's rights and remedies including, but not limited to, any right to cancellation and bill-back charges. If Seller continues to make shipments after Buyer default, action shall not constitute a waiver nor affect Seller's legal remedies.

B. Cancellation: Standard Products. Buyer is responsible for one hundred percent (100%) payment if any order scheduled for delivery within thirty (30) days is canceled. Upon written notice to Seller, Buyer may cancel any order for standard products scheduled for shipment beyond thirty (30) days after Seller receives Buyer's cancellation notice. Buyer shall be liable for any applicable bill-back charges resulting from such cancellation. Buyer shall also pay, as a rescheduling/ restocking charge, twenty-five percent (25%) of the price specified on order sheet for each canceled product unless Buyer reschedules the order for delivery within one hundred twenty days (120) days.

C. Cancellation: Custom Products. If Buyer cancels orders for products containing d esign, markings, packaging or other specifications unique to Buyer, Buyer shall pay the following cancellation charges in addition to any applicable bill-back charges.

Percentage of Purchase Price to be Paid Upon Cancellation Notice	Number of Days Before Scheduled Delivery that Written Cancellation is Received
100	0-30
75	31-60
50	61-90
20	91-120

10. SEVERABILITY. In case any one or more of the provisions or parts of a provision contained in this agreement shall for any reason be held to be invalid, illegal or unenforceable in any respect, such invalidity, illegality or unenforceability shall not affect ay other provision or part of a provision of this agreement but this agreement shall be construed as if such invalid, illegal or unenforceable provision or part of a provision had never been contained herein.

11. ARBITRATION. Any controversy or claim arising out of or relating to this agreement shall, at the option of either party, be settled by arbitration in accordance with the rules of American Arbitration Association. Provided, however that in the event of a violation of the provisions of paragraph 13, General Digital shall be entitled to injunctive or other relief to enforce its rights under this agreement without prejudice to its rights under this paragraph. Such injunctive or equitable relief shall not preclude the award of monetary damages, if appropriate. The parties acknowledge that, by agreeing to this provision, each waives its right to a jury trial in exchange for a quicker and less expensive method of resolving any dispute concerning this agreement or anything relating thereto.

12. CONTROLLING LAW. This Agreement shall be governed by, subject to, and construed according to the laws of the State of Connecticut. For purposes of applying to Connecticut law, this Agreement shall be deemed to have entered into and wholly performed in Connecticut. Buyer warrants, and it is a condition of this acknowledgment, that Buyer has complied, and will continue to comply with applicable laws, regulations, and order of the United States Government, and of any state or political subdivision thereof, including all statutes, regulations, and directives that are applicable to Buyer's purchase thereunder. Without limiting the foregoing, Buyer further warrants that they have complied with all applicable provisions of the Fair Labor Standards Act of 1938, as amended; and all regulations and orders issued by the US Department of Labor, as well as US export regulations including, but not limited to, the International Trafficking and Arms Regulations (ITAR). Buyer further agrees to defend, indemnify and hold Seller harmless from and against claims, losses, damages, costs and expenses that may be sustained by reason of Buyer's failure to meet any of the above requirements.

13. PROPRIETARY RIGHTS AND CONFIDENTIALITY.
A. Rights in any and all of Seller's information, know-how, trademarks, data, trade secrets, inventions, patents, plans, drawings, specifications, processes, designs, etc., whether or not acquired or reduced to practice during the course or as a result of this order, shall remain the property of Seller and Seller shall not be obligated to provide any of such property to Buyer. Buyer shall obtain no rights in any of such property and the price charged shall include no consideration for such property unless specifically so stated on the face of Seller's Standard Terms and Conditions of Sale. Listing as a separate line item and separately billing for Non-Recurring Engineering (Engineering Services) does not accrue to Buyer rights in any of the aforementioned property of Seller unless specifically stated on the face of Seller's Standard Terms and Conditions of Sale.

B. Buyer acknowledges that propriety property of Seller is a valuable business asset of Seller and that unauthorized use or disclosure of such would result in irreparable damage to Seller. Buyer agrees that it will:

- use proprietary information of Seller only for the purposes intended and necessitated by this order, not otherwise use or disclose any proprietary information of Seller without Seller's express informed written consent;
- take all reasonable precautions to insure protection against use or disclosure;
- and return, at Seller's request, all proprietary property of Seller which is in the possession or control of Buyer.

C. Any data, information, specifications, processes, drawings, designs, plans, know-how, etc. of Buyer supplied to Seller in conjunction with this order must be considered free from proprietary restrictions unless expressly agreed otherwise in writing signed by and authorized by an officer of General Digital™ Corporation.

11.0 Contact Information 78

We want you to be completely satisfied with your product. Please feel welcome to contact us should you require any of our services.

General Digital™ Corporation
8 Nutmeg Road South
South Windsor, CT 06074

Phone: 860.282.2900
Toll-Free: 800.952.2535
Fax: 860.282.2244

Hours: Monday–Friday, 9:00 A.M.–5:00 P.M. EST

Web: http://www.GeneralDigital.com/

Sales

Phone: 860.282.2900, Option 1
Fax: 860.282.2244
E-mail: gd_mn@GeneralDigital.com

Service

Phone: 860.282.2900, Option 4
Fax: 860.282.0198
E-mail: service@GeneralDigital.com

In your correspondence with the Service Department, please provide the following information:

- Company name
- Company address
- Contact name
- Contact phone and fax numbers

You can expect to receive a reply within one business day.

A.1 Display

- High Brightness Liquid Crystal Display (LCD) with 4.6" H by 8.3" W Viewing Area (10.4" Diagonal)
- 480 Characters (12 Rows by 40 Columns)
- 5 x 7 Character Matrix with a Character size of 0.26" H x 0.15" W
- +70°, -40° Horizontal/±70° Vertical Viewing Angle

A.2 Operator Interface

- Resistive Touchscreen with Audible Touch Input Feedback
- Visual Touch Input Feedback Signal
- Optional ASCII Keyboard Input

A.3 Functions

- Power-on Setup for Key Operating Parameters
- Software Character Set Selection
- Scroll or Block Data Formats, with Field Protection
- Software Cursor Control and Enable/Disable
- 128 Display Pages Standard; More Pages Available Optionally
- Independent Page Editing and Display
- Page to Page Copy and Merge
- Software Touch Enable/Disable, for All or Selected Areas
- Programmable Character Intensity and Blink Rate

A.4 Communications

- EIA RS-232C Full Duplex Standard
- Rates from 300 to 19,200 Baud
- Selectable XON/XOFF Handshaking
- Selectable Parity, Number of Data Bits, and Number of Stop Bits

A.5 Physical

- Protective Enclosure with Available Rack Mount Panel or Right Angle Mounting Brackets
- Front Surface Sealed with Non-porous Gasket Adhered Between Front Bezel and Touchscreen to Minimize Liquid and Dust Contamination
- 8.625" H x 12.0" W x 4.0" D
- Operating Weight: 7.4 Pounds

A.6 Electrical

- 85–264 VAC
- 47 Hz–63 Hz
- 2 Amp Slo-Blo Fuse

A.7 Environmental

- Operating Temperature: 0 to 50 Degrees Centigrade
- Storage Temperature: -20 to 70 Degrees Centigrade
- Relative Humidity: 0 to 95 Percent Non-condensing

Standard Character Set

MSD / LSD	0	1	2	3	4	5	6	7
0	–	–	(SP)	0	@	P	↓	▯
1	–	(DC1)	!	1	A	Q	→	▯
2	–	(DC2)	"	2	B	R	|	▯
3	–	(DC3)	#	3	C	S	¬	▯
4	–	(DC4)	$	4	D	T	¢	▯
5	–	(NAK)	%	5	E	U	◇	▯
6	(ACK)	–	&	6	F	V	▒	▯
7	(BEL)	–	'	7	G	W	▒	▯
8	(BS)	–	(	8	H	X	▚	▮
9	(HT)	–	)	9	I	Y	▞	BLANK
A	(LF)	–	*	:	J	Z	☰	|
B	(VT)	(ESC)	+	;	K	[	|||	‖
C	(FF)	–	,	<	L	\	=	|||
D	(CR)	–	–	=	M	]	|	||||
E	–	–	.	>	N	↑	=	|||||
F	–	–	/	0	O	←	|	(DEL)

Alternate Character Set

	6	7
	'	p
	a	q
	b	r
	c	s
	d	t
	e	u
	f	v
	g	w
	h	x
	i	y
	j	z
	k	{
	l	|
	m	}
	n	~
	o	(DEL)

NOTE 1: ▯ IS USED AS A "PUSH BUTTON". EACH ONE APPEARS AS A BOX. BUT IN ECHO MODE RETURNS A DIFFERENT CHARACTER WHEN TOUCHED.

NOTE 2: CODES 78–7E ARE USED FOR CONTINUOUS, FINE RESOLUTION BAR GRAPHS. CONSTRUCT THE BULK OF THE BAR WITH SOLID BLOCKS (78 OR 7E) AND FILL OUT THE LAST POSITION WITH 1–4 LINES (7A–7D).

The Factory Settings should be adjusted by advanced users only. The page can be accessed from any screen by placing a finger in the upper right-hand corner of the screen and, while maintaining steady pressure, sliding it straight across to the upper left-hand corner. Switching to the Factory Menu will not destroy the contents of any VuePoint™ page, but all subsequent serial communications with the unit will be ignored until the menus have been exited. The Factory Settings will not be saved until you touch G) REBOOT or exit the VUEPOINT SETUP, which is reached via F) USER OPTIONS. Cycling the power without performing one of these two actions will result in lost settings.

Figure C-1
Factory Settings Menu

C.1 Palette

The Palette Settings adjust the text color and screen background color.

Touch UP and DOWN for RED, GREEN and BLUE in FOREGROUND to adjust the text color. The default setting is 31/31/31.

Touch UP and DOWN for RED, GREEN and BLUE in BACKGROUND to adjust the screen background color. The default setting is 0/0/0.

When finished, touch EXIT to return to the Factory Settings.

C.2 Configuration

The Configuration Settings adjust the Bell Frequency and Duration, V Touch, and the number of active Pages.

Touch UP and DOWN to adjust the Bell Frequency. The default setting is 90.

Touch V TOUCH to enable or disable the Visual Touch Indicator, which is a small circle in the lower left corner of the screen. The default setting is DISABLED.

Touch UP and DOWN to adjust the maximum number of pages available to users. The maximum number of pages available for a standard VuePoint™ is 128. The default setting is 3 pages.

Touch UP and DOWN to adjust the duration of the bell. The default setting is 3.

When finished, touch EXIT to return to the Factory Settings.

C.3 Factory Defaults

Touch CR ON TOUCH to enable or disable the carriage return that appends the Screen Echo Touch Response. All other touch response modes are unaffected by this setting. The default setting is ENABLED.

Touch KB REDIRECT to enable or disable the keyboard input. When a standard PC keyboard is plugged into the unit, it will normally act like a standard VuePoint™ keyboard, sending its output through the serial port. When KB REDIRECT is active, the keyboard may be used to type on the VuePoint™ screen as if it were the remote computer. The default setting is FALSE.

Touch STANDARD SET to cycle the character set through Normal, Underline and Symbolic. The default setting is SYMBOLIC.

Touch ALTERNATE SET to cycle the character set through Normal, Underline and Symbolic. The default setting is NORMAL.

When finished, touch EXIT to return to the Factory Settings.

C.4 Touch Calibration

This screen will display the current touch calibration settings for informational purposes.

The touch calibration can also be reset. Touch CLEAR SETTINGS to clear the current calibration settings. A subsequent touch anywhere on the screen will bring up the calibration screen. Touch the crosshairs in the upper left quadrant and the lower right quadrant of the screen, then touch GOOD.

Note: If any point other than the word GOOD is touched immediately after touching the two crosshair points, the calibration procedure will automatically restart.

When finished, touch EXIT to return to the Factory Settings.

C.5 Factory Reset

Touch FACTORY RESET to reset all Factory Settings to their original configuration, except for the touch calibration settings. The bell will sound when touched but the screen will do nothing.

C.6 User Options

Touch USER OPTIONS to return to the standard startup screen, VUEPOINT SETUP.

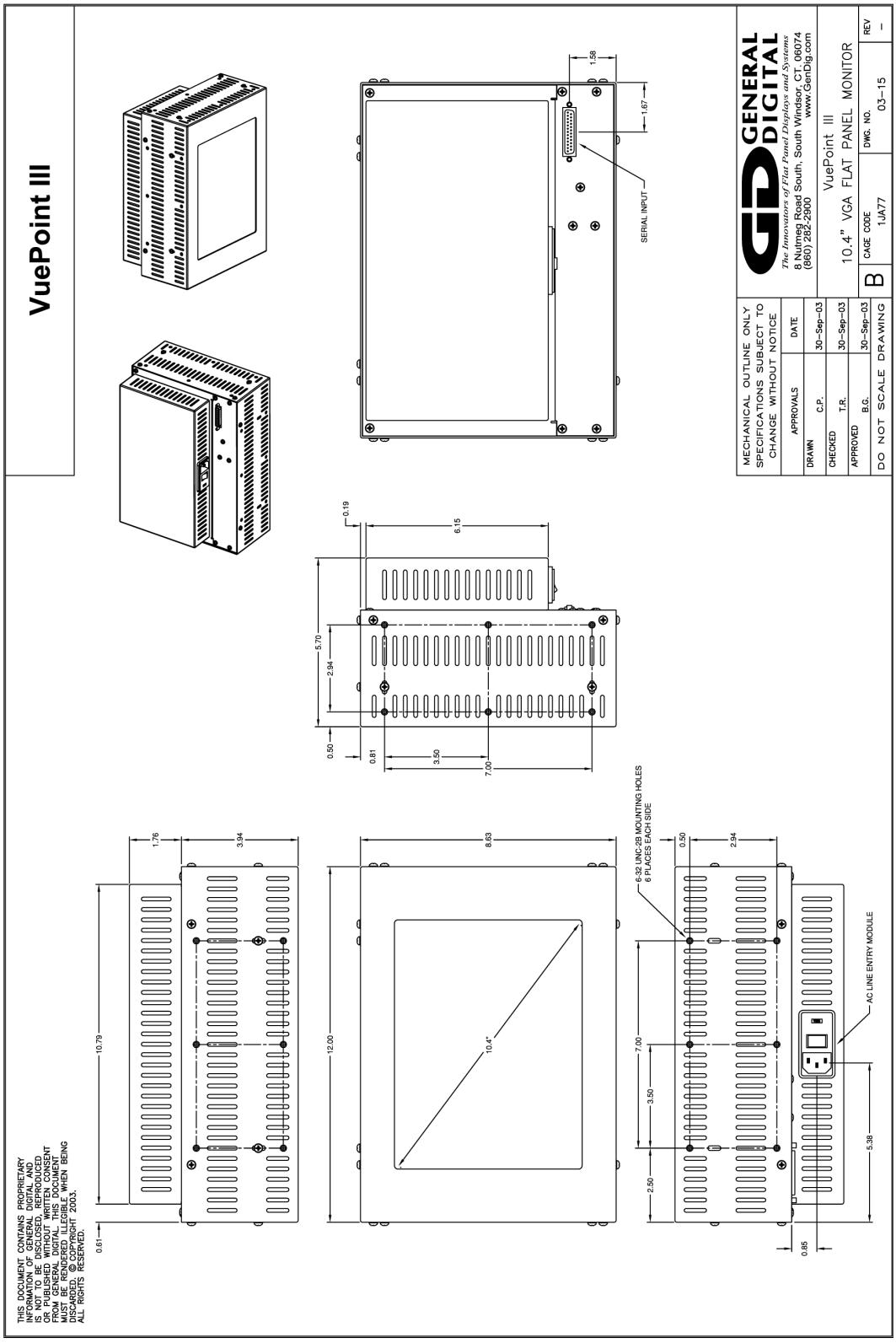
C.7 Reboot

Touch REBOOT to restart the VuePoint™ III. Note that this will save any changes you have made to the FACTORY SETTINGS and VUEPOINT SETUP.

D.1 Documentation

VuePoint™ II Programmer's Manual

Note: Because the VuePoint™ III was programmed to function as a VuePoint™ II, the creation of a VuePoint™ III Programmer's Manual was deemed unnecessary. All of the programming information in the previous generation manual applies to the current incarnation of VuePoint™. Please contact a Sales Engineer if you require a VuePoint™ II Programmer's Manual.


VuePoint III
SERIAL INPUT
AC LINE ENTRY MODULE
6-32 UNC-2B MOUNTING HOLES
6 PLACES EACH SIDE
10.4"
MECHANICAL OUTLINE ONLY
SPECIFICATIONS SUBJECT TO
CHANGE WITHOUT NOTICE
APPROVALS
DATE
DRAWN C.P. 30-Sep-03
CHECKED T.R. 30-Sep-03
APPROVED B.G. 30-Sep-03
DO NOT SCALE DRAWING
GENERAL DIGITAL
The Innovators of Flat Panel Displays and Systems
8 Nutmeg Road South, South Windsor, CT. 06074
(860) 282-2900
www.GenDig.com
VuePoint III
10.4" VGA FLAT PANEL MONITOR
B
CAGE CODE 1JA77
DWG. NO. 03-15
REV -
THIS DOCUMENT CONTAINS PROPRIETARY INFORMATION OF GENERAL DIGITAL AND IS NOT TO BE DISCLOSED, REPRODUCED OR PUBLISHED WITHOUT WRITTEN CONSENT FROM GENERAL DIGITAL. THIS DOCUMENT MUST BE RENDERED ILLEGIBLE WHEN BEING DISCARDED. © COPYRIGHT 2003. ALL RIGHTS RESERVED.

F.1 Page Selection

Select Working Page	ESC W pp
Select Display Page	WSC D pp
Copy Page	ESC C pp
Merge Page	ESC m pp
Display Next Page	ESC N
Select User Bank	ESC Q d

F.2 Set Character Attributes

Blink Rate And Visibility	ESC S B d
Intensity	ESC S I d
Protection	ESC S P d
Sensitivity	ESC S S d
Character Set	ESC S C d

F.3 Modify Character Attributes

Blink And Visibility	ESC M B d ww
Intensity	ESC M I d ww
Protection	ESC M P d ww
Sensitivity	ESC M S d ww

F.4 Page Mode Control

Set Response	ESC R d
Set Touch	ESC T d
Reset	ESC X d
Erase	ESC E d
Set Tabs	ESC Y d
Cursor Advance At End Of Line	ESC A d
Right To Left Data Entry	ESC L ww
Scroll	ESC B d
40 Character Line Entry	ESC J dd bb

F.5 Printer Expansion

Direct Printer Output	ESC u d

F.6 Additional Commands

Input Device Control	ESC I d
Print Page	ESC O ww
PROM Program	ESC p d
Bell Enable/Disable	ESC G d
Save/Erase Battery-Backed Memory	ESC w d
Power Fail Status/Reset	ESC F d
Power-On Reset	ESC Z
Enter Self-Diagnostics Mode	ESC d b
Extended PROM Program	ESC q dd

F.7 Device Control

	Dec Value	Command Sequence	Keyboard Function
Bell	7		Ctrl G
Clear Screen	20		Ctrl T
Disable Screen	18		Ctrl R
Response Enable	17		Ctrl Q
Response Disable	19		Ctrl S

F.8 Cursor Control

	Dec Value	Command Sequence	Keyboard Function
Home		ESC H	
Cursor Display		ESC K d	
Position Cursor		ESC P rr cc	
Carriage return	13		Ctrl M
Down Cursor	10		Ctrl J
Left Cursor	8		Ctrl H
Up Cursor	11		Ctrl K
Right Cursor	12		Ctrl L
Tab	9		Ctrl I
Rubout	127		Del

F.9 Speech Expansion (Requires Speech Option to Function)

Load Speech Buffer	ESC s dddd
Enable Speech Processor	ESC t 0
Speech Status	ESC t 1

F.10 Barcode Expansion (Requires Barcode Option to Function)

Barcode Entry	ESC b d