Quickstep™

DSP™ Programming Environment for Personal Computers

USER MANUAL

Control Technology Corporation 25 South Street Hopkinton MA 01748

Copyright Notice

Quickstep™ User Manual

Table of Contents

Reference

Introduction

Quickstep™ is a comprehensive programming environment for the following CTC Automation Controllers:

Model 2200
Model 2400iE
Model 2400iEA
Model 2800iE
Model 2800iEA

Quickstep has been designed to run on standard IBM®-compatible personal computers. It allows you to:

1. Write programs using CTC's DSP™ programming language, with full interactive editing capabilities.

2. Store your programs on disk in the form of ASCII text files.

3. Print out listings of your programs.

4. Compile and debug your programs, then download then into one of the above-mentioned controllers.

5. Monitor the operation of your controller, including the steps being executed, the content of the controller's numeric registers, and the state of the controller's inputs, outputs, flags and analog I/O. Certain of these resources may also be dynamically modified using Quickstep.

Every effort has been made to provide an easy- and fast-to-use programming environment. All operations within Quickstep are performed within a single "screen" — there is no need to change "modes" to perform such tasks as printing a listing of your program.

Within this "modeless" environment, certain operations (for example, entering a Data Table) are performed using pull-down windows and menus, which overlay your normal screen. These windows are activated using the function keys ("F1" through "F10") found on most personal computers.

Prompts are displayed at the bottom of the screen in a non-intrusive manner — for beginners, they can greatly reduce orientation time, while for experienced users they may simply be ignored. A context-sensitive HELP provision displays a full explanation of where you are and what your options are at any time, simply by pressing the "F1" key.

Even during the actual entry of your program, the needs of both beginners and experienced users are respected. Instructions are programmed by entering the first letter (sometimes several letters are required) of the function desired; for example, to enter a "Monitor" instruction, simply press the key "m". Quickstep will automatically fill in the rest of the instruction.

This method of entry "filters" your keystrokes, disallowing syntax errors without requiring a long series of tedious menus.

The above features combine to create an optimum environment for writing DSP programs with a minimum of orientation (Software Engineers are fond of calling such programs "intuitive"). This saves you time, and helps you concentrate on your automation task.

System Requirements & Compatibility

We have attempted to make Quickstep compatible with a wide variety of IBM®-PC compatible computers, many of which exhibit internal differences which may affect program execution. We have tested Quickstep on a number of such systems, but obviously cannot guarantee universal compatibility. Please let us know if you experience difficulties.

The minimum system requirements for running Quickstep are:

IBM®-PC or 100% compatible
DOS version 3.0 or higher
512K memory
Floppy Disk drive
RS-232 Communications Adapter

The preferred development environment is:

IBM-AT® or 100% compatible
DOS version 3.0 or higher
512K memory
Floppy Disk drive
Hard Disk drive
RS-232 Communications Adapter

The latter system will execute Quickstep substantially faster.

You may wish to investigate the purchase of a "lap-top" portable computer. These provide the added convenience of portability; the computer may be easily carried out to where a machine is being developed. Some of these "lap-tops" are battery operated, while others may feature internal hard disk drives and the fast 80286 processor (as used in the IBM-AT®). We have successfully used the Toshiba T3100 (AC-powered, internal hard disk drive, 80286 processor). When evaluating lap-tops, pay careful attention to the screen appearance; some of them have difficult-to-read displays.

Note: as of this printing, we have identified a bug in the version of DOS supplied with certain Zenith lap-top computers. This bug affects the ability of Quickstep to download programs. Although Zenith has committed to resolving this problem, in the meantime we have created a version of Quickstep (identified as V1.0Z) which will work with the existing Zenith version of DOS.

Preparing to Use Quickstep

Quickstep is supplied on an MS-DOS® compatible diskette, which may either be copied to a hard disk drive on your computer, or copied onto another diskette to be used as a "working disk". *Never use your original master diskette as a working disk, or you may accidentally erase your only copy of Quickstep!*

Your master diskette contains a number of files which constitute Quickstep, along with a program called "INSTALL.BAT", which may be used to automatically copy Quickstep onto your hard disk drive. This chapter contains instructions for running INSTALL, followed by a description of each file supplied on the master diskette.

Hard Disk vs. Floppy Diskette Operation

Quickstep may be used on computers whether or not they are equipped with a hard disk drive. Certain operations will be considerable faster using a hard disk drive; these include the initial starting of Quickstep, the saving and reading of files, starting a printed program listing, and using the HELP function. All of these operations require the computer to access its disk, and these accesses take considerably longer with a floppy diskette.

The bulk of the operation of Quickstep, however, is within your computer's memory. Therefore, most operations other than those specifically mentioned above will not be affected by your computer's disk complement. In fact, once Quickstep is initially started on your system, the Quickstep diskette may be removed entirely and replaced with another diskette to be used for data storage. You need then only replace the Quickstep diskette if you press the HELP key (as noted below, the HELP files are stored separately on the Quickstep diskette.

Making a Working Copy of Quickstep

MS-DOS® provides a command for making an exact duplicate copy of a diskette; this is the "DISKCOPY" command. If your computer has two floppy diskette drives, you may copy from one drive to the other; otherwise, DISKCOPY allows you to copy a diskette by switching the original diskette and the copy back and forth within the same drive.

To use the DISKCOPY command with a two-drive system, place a diskette containing the DISKCOPY program ("DISKCOPY.COM") in the drive designated "a:", and a blank diskette in the drive designated "b:". Then, from a DOS "a:" prompt, type the following command:

```
diskcopy a: b:
```

The DISKCOPY program will then start up, prompting you to insert the "source" diskette in drive a:. Remove the diskette containing DISKCOPY and insert the Quickstep master diskette, then press a key on the computer, as instructed, to allow DISKCOPY to continue. DISKCOPY will then proceed to copy all of the files from the Quickstep master diskette to the new diskette.

When copying using a computer with only a single diskette drive, the DISKCOPY command must be entered differently. First, place a diskette containing the DISKCOPY program file in the drive designated "a:", and type the following command in response to the DOS "a:" prompt:

```
diskcopy
```

You will then be prompted to insert the "source" diskette (this is the Quickstep master diskette), and press any key once you have done so. DISKCOPY will read all of the information from the Quickstep master diskette, and then prompt you to insert the "target" diskette. This is the blank disk which you wish to become the working copy of Quickstep. After inserting the blank disk and pressing a key on the computer (as a signal for DISKCOPY to proceed), DISKCOPY will finish the copying operation.

Note that various versions of DOS may have slightly different formats for the DISKCOPY command. If you have any problems, you should refer to the DOS manual which came with your computer.

INSTALLing Quickstep on your hard disk drive

The program "INSTALL.BAT", supplied on your master diskette, contains a series of DOS commands which will automatically copy the Quickstep program files onto your hard disk drive. Upon running INSTALL, the following will happen:

1. A new DOS directory will be created with the name you specify.

2. The Quickstep program files will be copied into the DOS directory created above.

3. A sub-directory called "HELP" will be created below this new directory, and all of the Quickstep help files will be copied into this subdirectory.

4. Additional instructions will then appear on your screen.

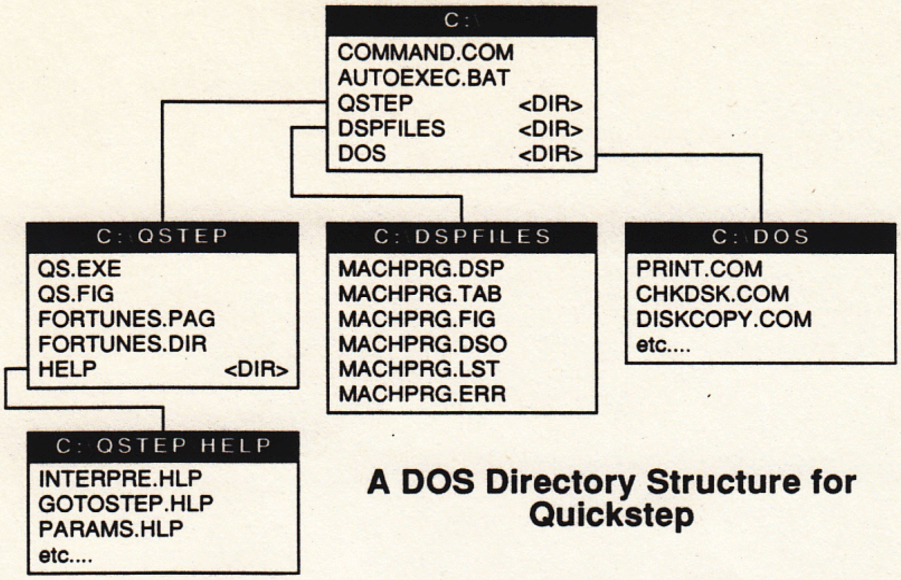
Before running INSTALL, first decide on a name for a Directory in which your Quickstep program files will reside (suggestions: QSTEP, QUICKSTP, QS, or CTC). This directory should not yet exist on your hard disk. In addition, there must be sufficient space on your hard disk for the Quickstep files (approx. 360K).

Then, insert the Quickstep master disk in drive A: of your computer and type the following command:

```
a:install c:\qstep
```

...where "qstep" represents the name chosen for your Quickstep directory. This command triggers the INSTALL program on the diskette, and passes INSTALL the full name (including the DOS path specification) of the Quickstep directory.

You will then see a number of messages appear on the computer screen as the various files are copied.



A DOS Directory Structure for Quickstep

Description of Quickstep Files

In addition to the INSTALL program described above, the following files are supplied on the Quickstep master diskette:

READ.ME
This text file contains information regarding the release of Quickstep contained on the master diskette. It may be read by "print"ing the file to a printer, or by "type"ing the file to the screen.

QS.EXE
This is the main Quickstep program file. It is now started with the command "QS". If not in the currently-active directory, this file must be within the DOS search path, or must be specified with a pathname (e.g.; C:\qstep\qs) to be executed.

QS.FIG
This is the default configuration file for Quickstep, and must be within the directory containing QS.EXE, within a directory called "\LIB", or within the current directory, to start Quickstep.

*.HLP
There are a number of files with the extension ".HLP", which contain the messages displayed at various levels of Quickstep in response to the HELP key (F1). To be functional, these should be contained in a subdirectory named HELP, off of the directory containing QS.EXE. If you do not require HELP, these files may be excluded to conserve disk space.

SAMPLE.DSP, SAMPLE.FIG, SAMPLE.TAB
These files constitute a sample program which may be called up from Quickstep to illustrate available instructions. It is not a practical program.

FORTUNES.PAG, FORTUNES.DIR
These files contain the data displayed in response to the F4 (Philosophy) key, and should reside in the same directory as QS.EXE, within a directory called "\LIB", or within the current directory. Excluding these files will only affect your peace of mind.

Using the DOS "PATH" Command

The "directory" structure allowed by DOS is a convenient way to organize the various programs and files by maintaining separate directories for different types or categories of files. For example, you may wish to keep your automation programs in a directory by themselves, or to establish a directory for each new project in which you're involved.

In such instances, the ability to keep the Quickstep program files in a separate, isolated directory allows you to reduce clutter within the directories with which you work on a day-to-day basis. However, when you start Quickstep (see Chapter 4: "Starting a New Program"), DOS must be able to find these program files in order to execute them.

One way to tell DOS where to find Quickstep is to specify the full DOS "pathname" when starting the program. For example, instead of typing "qs" to start Quickstep, you might type:

```
c:\qstep\qs
```

This command tells DOS to look for the program "qs.exe" on drive "c:" (the hard disk drive), in a directory called "qstep", which is immediately below the top-level directory of drive c:.

There is, however, a shortcut within DOS for allowing it to find frequently-used program files without specifying the full pathname. This is the DOS "PATH" command. The PATH command allows you to specify one or more directories in which DOS will automatically look for a program to be executed, if that program is not found in the current directory. For example, the command:

```
path c:\;c:\qstep
```

...tells DOS, upon attempting to execute a program file, to first look in the current directory (DOS will always do this anyway), then look for the program file in the top-level directory ("c:\"), then look for the program in a directory off of the top-level directory called "qstep" ("c:\qstep"). Note that a semicolon is used to separate the directory paths to be searched.

This means that, if your Quickstep program files are in the directory c:\qstep, you need merely type "qs" from any directory, and DOS will find and execute Quickstep. The logical way to use this capability is to first change directories to the directory containing your project files, then type "qs" to start Quickstep. Then, unless you specify otherwise, any time you save or read a file, that file access will be to your project directory.

A Word About "AUTOEXEC.BAT"

DOS also allows you to create a special-purpose file called "AUTOEXEC.BAT", containing one or more commands which DOS will automatically execute whenever power is first applied to your computer (or when you "reboot"). If you insert the PATH command mentioned above in a file called AUTOEXEC.BAT, this path will be set each time you start your computer. Otherwise, you would have to retype the PATH command each time.

For further information about the PATH command, the AUTOEXEC.BAT file, the DOS directory system, and other related features of DOS, refer to the DOS manual that came with your computer, or to one of the many reference books on MS-DOS® available from any bookstore with a computer section.

For first-time users of computers, a little reading will result in much more powerful use of your PC, and a more efficient use of your time.

Starting a New Program

This chapter assumes you have already installed Quickstep on the hard disk drive of your computer, or are prepared to run Quickstep from a duplicate diskette, as discussed in the previous chapter. Now, we will go through the necessary steps to start Quickstep and begin writing a DSP program.

Starting Quickstep

The command to start Quickstep is "qs". Note that, as mentioned in chapter 3, DOS must be able to find the Quickstep program files when you have entered this command. This means that these files must either be in the current directory, or the search path for DOS must include the directory in which they reside.

The main Quickstep screen will then appear on your monitor, along with inset windows containing the serial number of your copy, and a "sign-on" message. To make these two windows disappear, simply press the space bar (any key will do) on the computer's keyboard. You are now ready to begin using Quickstep.

A Guided Tour of the Quickstep Screen

All Quickstep operations take place within a single "screen", shown in the associated illustration. The main "working area" of this screen is used to build the steps of a DSP program, one step at a time. As you program a DSP step, the full step appears in this area. If the step will not fit within the screen, you will be able to scroll the screen up and down to view the remainder of the step.

The bottom line of the screen is reserved for prompts; watch this area for instructions as to which keys to press next, etc.

The Function Keys & Pull-Down Windows

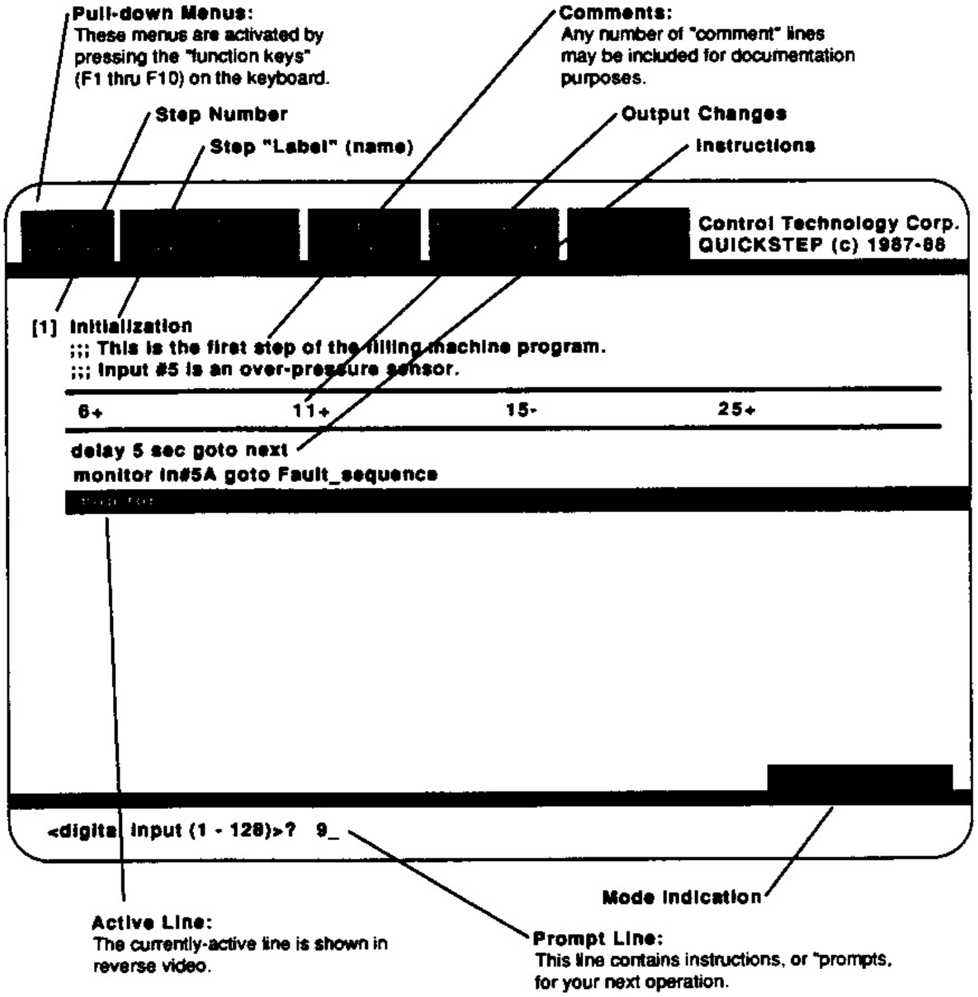
A series of pull-down windows are accessible via the function keys ("F1" through "F10") present on the computer keyboard. At the top of the Quickstep screen are labels indicating the function performed by each of these keys.

Not all of the function keys trigger pull-down windows; for example, the F10 key ("STEP#") allows you to quickly jump to any new step desired. Similarly, the F9 key ("SEARCH") does not trigger a window, but rather directly initiates the search function.

The operations performed by these function keys are described briefly below. For a detailed description, refer to the Reference Section's "Guide to Quickstep Function Keys".

F1: The HELP Function – This key may be used at any time during the use of Quickstep to obtain further information about what to do next. Upon pressing the F1 key, a window will appear within the screen containing text which should prove helpful.

The Quickstep™ Screen



```
                                                  Control Technology Corp.
                                                  QUICKSTEP (c) 1987-88

[1] Initialization
    ;;; This is the first step of the filling machine program.
    ;;; Input #5 is an over-pressure sensor.

    6+                   11+                   15-                  25+

    delay 5 sec goto next
    monitor in#5A goto Fault_sequence

<digital input (1 - 128)>? 9_
```

Pull-down Menus:
These menus are activated by pressing the "function keys" (F1 thru F10) on the keyboard.

Step Number

Step "Label" (name)

Comments:
Any number of "comment" lines may be included for documentation purposes.

Output Changes

Instructions

Active Line:
The currently-active line is shown in reverse video.

Mode Indication

Prompt Line:
This line contains instructions, or "prompts, for your next operation.

F2: The FILE Menu – The function key F2 triggers a pull-down menu, allowing you to:

"New" – Start a new program.
"Save" – Save your program to disk.
"Read" – Read an existing program from disk.
"List" – Lists your files, and allows you to select one to read.
"Print" – Prints a listing of your program.
"Compile" – Prepares a program for downloading.
"Download" – Sends a compiled program to a CTC controller.
"Zuspend" – Suspends Quickstep and starts a new copy of DOS.
"Quit" – Leaves Quickstep and return to DOS.

F3: The PARAMETERS Table – When starting a new program, you must first tell Quickstep what controller you intend to use, and set certain parameters relating to the use of dedicated inputs, I/O counts, etc. The "PARAMETERS" table is where this information is stored.

F4: The PHILOSOPHY Key – Provides instantaneous access to the wisdom of the ages, all at the mere press of a key.

F5: The DEBUG Key – After compiling a program, allows you to quickly locate certain common types of errors.

F6: RSVRD – Not currently active, this key is reserved for use in future versions of Quickstep.

F7: The MONITOR Menu – When connected to a CTC controller, this key allows you to select among a variety of monitoring, diagnostic and override functions. These functions allow real-time interaction with the controller, while your program executes.

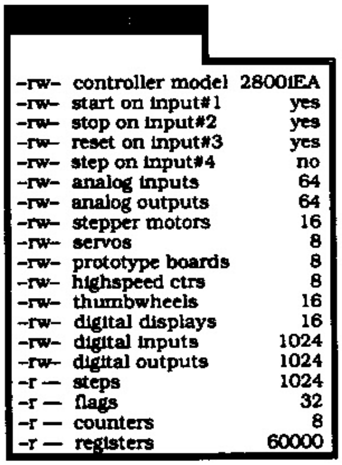
F8: The DATA TABLE – This key opens a large, spread-sheet-like window, in which you may enter and edit a data table. This data table may then be used as a numeric resource for your program.

F9: The SEARCH Key – This key allows you to search through your program for any text (for example, to locate all references to a given step name).

F10: The STEP# Key – This key causes Quickstep to instantly move to any given step number of your program, displaying that step within the screen.

Setting the Controller PARAMETERS

Before beginning a new program, you must set the appropriate parameters for the controller which you intend to use. This is accomplished by pressing the F3 key to pull down the "PARAMETERS" table.



```
-rw-  controller model  2800IEA
-rw-  start on input#1      yes
-rw-  stop on input#2       yes
-rw-  reset on input#3      yes
-rw-  step on input#4        no
-rw-  analog inputs          64
-rw-  analog outputs         64
-rw-  stepper motors         16
-rw-  servos                  8
-rw-  prototype boards        8
-rw-  highspeed ctrs          8
-rw-  thumbwheels            16
-rw-  digital displays       16
-rw-  digital inputs       1024
-rw-  digital outputs      1024
-r —  steps                1024
-r —  flags                  32
-r —  counters                8
-r —  registers           60000
```

The Parameter Table

As shown in the illustration, the Parameters table lets you enter the model number of the controller being used, along with other information to configure your system. At the left side of each line of the Parameters table, the "permission" information for that parameter is indicated, with "-r-" indicating a parameter which may only be read, and "-rw-" indicating a parameter which may be either read or changed ("written") by you.

To change a parameter, you must first select the appropriate parameter by using the up-arrow and down-arrow keys to move the video highlight to the appropriate line. In the instance of the model number and the

Summary of Operations to Start a New Program

1. Start Quickstep by typing "qs" from a DOS prompt.

2. Press any key to eliminate the copyright notice.

3. Press the "F3" key to access the "Parameters" table, and enter the appropriate controller model number, along with other configuration information.

4. Press the "F3" key (or "ESC") to remove the Parameters table from the screen.

5. Press "Enter" to begin entering the first step óf your program. Follow the prompts at the bottom of the screen to enter a step number, step label, comments, output changes for the step, and instructions within the step.

For further information, refer to the detailed information in this manual, and to the "Guide to Direct Sequential Programming".

numeric parameters (analog inputs, analog outputs, etc.), you may then delete the old entry by using the "delete" key, re-entering the desired value in its place. The "yes/no" parameters may be changed by typing "y" or "n", or by using the edit function (Alt-E) to toggle the selection.

The model number may be any valid CTC Automation Controller currently supported by Quickstep. These include:

2200
2400iE
2400iEA
2800iE
2800iEA

Any selection other than the above will result in an error message, and the model number will be defaulted to "2800iE".

The election to use the controller's dedicated inputs (start, stop, reset and step) may be made by indicating a "yes" next to the parameters "start on input#1", etc. Any of these first four inputs which are not dedicated via these entries may then be used as general-purpose inputs for limit switches, etc.

The remaining choices in the Parameters table indicate limits which will be placed on you while writing your program within Quickstep. For example, if you indicate that there will be 16 Analog Inputs in your control system, you will only be allowed a choice of "1" through "16" when programming an instruction for an analog input. These limits may, of course, be changed at any time simply by pulling down the Parameters table again.

Note that the limits for quantity of steps, flags, counters and registers are fixed, as a function of the controller model which you are programming. The register limit of 60000 indicates not the true number of registers within a controller, but rather an allowable range of register references which include a number of existing special-purpose registers. Refer to your controller's documentation for more specific information about register capacity.

When you have finished setting up the Parameters table to reflect the controller you will be using, press the "F3" key again (or ESC) to make the table disappear.

Entering the First Step

Having set the parameters for Quickstep, you may now begin the entry of the first step of the program. This is accomplished by pressing "Enter", which will make the "skeleton" of a new step appear in the central working area of the screen.

When entering a new step, Quickstep will prompt you, in order, for the following elements of the step:

1. A Step Number
2. A Step Label
3. Comments for the step
4. Output changes for the step
5. Instructions (i.e.; time delays, input monitoring, etc.)

After entering these elements, Quickstep enters an interactive editing mode, where you may scroll through the step, making any desired changes, additions or deletions. It is generally advisable to fully enter a step before going back to make changes, as this will make the entry process somewhat smoother.

The Step Number

The Step Number of the current step is shown within square brackets - like this "[Ø]" - in the upper left corner of the screen's working area. When programming the first step of your program, this Step Number will initially be shown as "Ø" (there is no "Step zero", this is simply a place holder).

This step number will be highlighted in reverse video, and you will then be prompted for a Step Number with the following message on the prompt line at the bottom of the screen:

"Step Number (1-256):"

Note: If the controller which you specified in the Parameters table has a higher step capacity (e.g.; the models 2400iEA/2800iEA), the prompt will reflect the allowable range ("Step Number (1-1024):").

Typically, you will want to program Step #1 first; this may be done by typing the number "1", and then pressing the "Enter" key. (In this particular case, pressing "Enter" alone will automatically enter "1".) At this point, the Step Number at the top of the screen will change to "[1]".

and the video highlight will automatically move to the Step Label.

The Step Label

The "Step Label" is a name which you may give to the step; this is simply another way in which you may refer to the step. For example, if you have given Step #50 the name "Parts_feed", because this is the function performed by the step, you may jump to this step from another part of the program with either of the following two instructions:

"goto [50]"
"goto Parts_feed"

Note that the second instruction carries more information, and results in a clearer program. For this reason, *always try to use labels when jumping between steps.*

One additional reason for using labels is to maintain the relationship between one step and the step to which it jumps, when additional steps are inserted between the two. As described in the section on program editing, new steps may be inserted into an existing program, resulting in the potential renumbering of some subsequent steps. Labels, however, will not be changed by this process.

Labels may also be used when starting multi-tasking using the "do" instruction. By naming the first step of each task with a name relating to the function performed by the whole task, your program becomes very "English-like", for example:

"do (Clamping_sequence, Pick_and_place, Fault_monitor) goto next"

Having previously entered a Step Number, as described above, the temporary label "<empty step>" will be highlighted and you will be prompted for a Step Label with the following message at the bottom of the screen:

"New label for this step?"

You may enter a label up to 80 characters long (we recommend keeping labels to 20 characters or less to conform to future CTC software introductions), and the label may contain any of the following characters:

A. Upper-case and lower-case alphabetic characters
(note: upper- and lower-case letters are treated as equivalent)

B. Numbers Ø through 9

C. The characters:
. (period)
/ (forward slash)
\ (back-slash)
: (colon)
_ (underline)

D. Spaces
(note: spaces will automatically be converted to underlines by Quickstep to make the label more visually cohesive)

Some examples of allowable labels are:

"Initialize:_phase_1"
"Pressure_too_high"
"Stamping_sequence"
"Wait_for_fault_clear"
"EXTEND_RAM"
"adopt_position_3"

After typing in the desired label, press the "Enter" key. If you do not wish to enter a Step Label, simply press "Enter" without typing a label. This will cause the temporary label "<empty step>" to disappear.

In either case, the video highlight will move to a newly-opened "Comment" line.

Entering Comments

Comments are lines of information which you may insert into your program for documentation purposes. *They have no impact whatsoever on the execution of your program by the controller and are, in fact, never downloaded to the controller.* Thus, they do not take up any of the controller's memory. They will, however, be printed out when you print a listing of your program.

The sole purpose of comments is to allow you to explain the functioning of your program, or the machine it controls. When writing programs, especially programs of any complexity, *we highly recommend the extensive use of comments* to explain:

- ⇨ The purpose of a step.
- ⇨ The usage of numeric registers within the step.
- ⇨ The input and output connections of the controller.
- ⇨ An overview of the sequence of events.
- ⇨ Diagnostic information for the step.

Comments may be many lines long. For example, it is not unusual to enter a multi-page comment in the first step of a program to document the "overview" of the program and the machine being controlled.

Having entered a Step Label, as described above, a Comment line will automatically open below the label, indicated by three semicolons followed by a flashing prompt. You may now type in any information, using any printable characters found on your keyboard. When you approach the end of a line, pressing the "Enter" key will end the current line and open a new Comment line.

This process will continue until you press "Enter" in response to a blank Comment line. This will close out the blank line (it will disappear) and make Quickstep proceed to the entry of Output information for the step.

Note that you may create blank comment lines (to delineate paragraphs, for example) by typing the space bar before pressing "Enter". Quickstep will interpret the resultant space as a valid comment entry and thus will not remove the line as being blank.

Output Changes

Below the "Comment" section of the Quickstep screen are two horizontal lines, between which are shown the output changes programmed for the step. It is possible to program instructions for up to 128 outputs to turn on or off in any combination, or to program a universal instruction to turn all outputs off.

Having finished entering Comments for the step, Quickstep will now highlight the output section, brightening the horizontal lines and highlighting the default output instruction:

"<NO CHANGE IN DIGITAL OUTPUTS>"

A prompt will also appear at the bottom of the screen:

"Output number (Ø for all off):"

You may now enter the first output which you wish to change state within this step. The order in which outputs are entered does not matter; Quickstep will automatically sort the outputs by number, and the controller will execute the instructions roughly simultaneously, regardless of the order of entry.

When entering the desired output, type in only the output number, followed by the "Enter" key. For example, to change output #15, type the characters "1", "5" and "Enter". Quickstep will respond with the prompt:

"Turn output #15 on<+> or off<->?"

The "+" and "-" keys are then used to program the output to turn on or off, respectively. Notice the use of angular brackets ("<" and ">"); these are frequently used within Quickstep to indicate the proper key to press to make a given selection.

As soon as you type either "+" or "-", Quickstep will show the output change between the two horizontal lines, and a prompt will appear requesting another output number to be changed. You may now type in another output number followed, once again, by a choice of on or off.

This sequence will continue until you simply press the "Enter" key in response to a prompt for another output number. At that point, Quickstep will move on to the programming of other instructions.

To program the special–case instruction to turn all outputs off, type "Ø", followed by the "Enter" key, when asked for an output number. The following text will then appear in the output section of the screen:

"<TURN ALL DIGITAL OUTPUTS OFF>"

Programming the Instructions

After entering the desired output information for a step, a blank line will be highlighted just below the output section of the screen, and the following prompt will appear at the bottom of the screen:

"Instruction?"

At this point, you may enter a programmable instruction (e.g.; delay, monitor, turn motor, etc.). This is accomplished by first pressing the key corresponding to the first letter of the desired instruction. For example, to enter a "Monitor" instruction, simply press the "M" key; Quickstep will automatically fill in the rest of the word "monitor".

In instances where more than one instruction begins with the same letter (for example; Cancel, Clear and Count), pressing the first letter ("C") will result in the following prompt:

"c<a>ncel, c<l>ear, c<o>unt?"

This prompt indicates the second letter of the instruction must now be pressed in order to fully qualify your choice. Quickstep will frequently use angular brackets - like this: <a> - to indicate the key to press for a given selection.

Important note:
The HELP key (F1) may be of great assistance in selecting an instruction. Pressing F1 in response to the "Instruction?" prompt will display a HELP window showing all of the possible instructions which may be entered, with the letters shown in CAPITALS to indicate the keys which must be pressed to select each instruction. You may directly press these keys from the HELP screen, without first returning to the main screen; your chosen instruction will be entered.

Once you have entered the basic type of instruction to be programmed, Quickstep will prompt you with a series of questions to "build" the rest of the instruction. For example, a "delay" instruction must contain a time *duration* to be executed, as well as a step destination to jump to after the delay; a "monitor input" instruction must contain the number of the input(s) to be monitored, etc.

When programming a delay instruction, for instance, the following prompt will appear at the bottom of the screen:

"delay t<w>heel, an<i>nput, <t>able, <c>ounter, <r>egister, <integer>?"

This indicates that the duration of the delay may be derived from any of the following controller resources:

twheel: a 4-digit thumbwheel array connected to the controller.
aninput: the value of an analog signal connected to an analog input.
table: a Data Table stored along with the program.
counter: the current value of one of the controller's internal counters.
register: the current value in one of the controller's numeric registers.
integer: a fixed-duration delay, expressed as an integer value in hours, minutes, seconds and hundredths of a second.

As with other choices, the letter in brackets (<>) must be typed to select a given choice. In the instance of <integer> entry, simply type the desired number.

You will then notice a further series of prompts to finish building the instruction. As you answer the questions within the prompts, you will see the instruction being built on the currently-active line of the working section of the screen. Note that pressing the "Escape" key at any time during the entry of an instruction will cause the termination of that instruction.

Programming Multiple Instructions

Each time you finish entering an instruction, a new line will open on the screen, allowing you to enter another instruction. In this manner, you may program any number of instructions within the step (up to the capacity of the step). If you should reach the bottom of the screen, the entire working area of the screen will scroll up, allowing you to continue programming.

When you are finished programming the instructions for a given step, press the "Enter" key in response to the prompt "Instruction?". This will close out the line and put Quickstep into an interactive editing mode, allowing you to scroll freely throughout the step, making any changes desired.

Using an Existing Program

You may bring into Quickstep a program which you had previously created, either for further entry/editing or simply to download it into a controller. This may be accomplished either by using the "F2: FILE" menu from within Quickstep, or by starting Quickstep originally with a command which includes your program's filename.

Reading a Program into Quickstep

There are two methods for reading an existing program file from within Quickstep. The most convenient method is to use the "List" option on the "F2: FILE" menu, as described below.

The "List" option allows you to view a listing of all files with the filename extension ".DSP" (i.e.; all DSP-language program files) within any specified directory on disk. You may then easily select any of those files to be read into Quickstep.

To select the List option, first pull down the FILE menu by pressing the F2 function key. Then, if the "List" option is not already highlighted, either press the "L" key or use the up-arrow/down-arrow keys to select "List". Upon pressing the "Enter" key, Quickstep will display a prompt at the bottom of the screen, as follows:

"List directory (C:\DSPFILES):"

Here, Quickstep is asking for the name of a directory, including its full DOS path, to be displayed on the screen. Quickstep uses, as its best guess, the name of directory into which you were logged when you originally started Quickstep. If this is, in fact, the directory containing the program you wish to read, then simply press the "Enter" key. Otherwise, type in the desired directory, along with its DOS path (refer to your DOS manual for details), followed by the "Enter" key.

Quickstep will now create a window on the screen, displaying a directory of all files it found within the specified directory having the filename extension ".DSP". The first file in the list will be highlighted. If this is not the file you wish to read into Quickstep, use the up-arrow/down-arrow keys to select the desired file. When the appropriate filename is highlighted, press the "Enter" key; Quickstep will then automatically read that file, allowing you to begin working with it.

If the desired file is, for any reason, not displayed, you may terminate the operation by pressing the ESCape key.

The second method of reading a program file from within Quickstep is to use the "Read" option on the "F2: FILE" menu. When you select this option, Quickstep will prompt you for a filename at the bottom of the screen. If your file is within the directory into which DOS was logged when you first started Quickstep, you may simply enter the name of the desired file, with or without the filename extension ".DSP". Otherwise, you must specify a full DOS path-and-filename specification to allow Quickstep to find your file.

Starting Quickstep with a Program File

From a DOS prompt, it is possible to start Quickstep with a command including the name of a program which you desire to edit. Quickstep will then "wake up" with your program already loaded.

To accomplish this, instead of starting Quickstep with the command "qs", use the following command format:

"qs yourprog"

...where "yourprog" is the name of the program you wish to load. A space should be used between the command "qs" and the name of your program file. Note that you need not use the extension ".DSP" in specifying your file.

This program file must be within the currently-logged directory, or you must specify a full pathname for DOS to find the file. In addition, of course, Quickstep must be within the DOS search path to be started successfully.

Editing a Step

Quickstep has been designed to allow rapid editing of all of the components of your program. After the initial entry of a step, a video highlight is used to indicate the element of the step which may be edited; this highlight may be moved by using the cursor-control keys (the up-arrow and down-arrow keys). Within the "output" section of the Quickstep screen, the left-arrow and right-arrow keys may be used to select a specific "output change" instruction for editing.

Once you have selected the part of the step you wish to edit, you then have the option of inserting new information (comment lines, output changes or instructions), deleting existing information or changing existing information. The following paragraphs provide the details for accomplishing these tasks.

The Editing Keys, A General Description

The keys used for editing within the various sections of the Quickstep screen are generally the same. The "Insert" and "Delete" keys are used to insert or delete additional elements (comment lines, output changes, instructions) within the step. The Quickstep edit function (the Alt-E key combination) is used to make a change to the currently-highlighed element on the screen.

One word of caution, though; when the step label is highlighed, it is possible to delete the entire step by pressing the "Delete" key. Quickstep will, however, ask you to confirm your intention before deleting the step.

Changing the Step Label

The step label may be changed by placing the highlight over the existing label and pressing the Alt-E key combination. You may now type in a new label, ignoring the remnants of the old label on the screen. When you have finished typing in the new label, press "Enter" to store the label.

A similar process may be used if you wish to add a label to a step which previously had no label at all. First, use the cursor-control keys to position the highlight in the label area of the screen (this will be possible even though no label had been previously entered). Then press Alt-E. You will now be able to enter a label for the step.

Do not press the "Delete" or "Insert" keys to enter or edit a label; these keys are used to delete or insert entire steps when the highlight is on the step label!

Editing Output Changes

There will always be some information within the "output" section of the Quickstep screen (this is the area delineated by two horizontal lines). There may be a message indicating that no output changes have been programmed for the step: "<NO CHANGE IN DIGITAL OUTPUTS>". If output information has been programmed, this may consist either of specific output changes (for example, "12+", "15-", etc.), or a global instruction reading "<TURN ALL DIGITAL OUTPUTS OFF>".

If you wish to program an output change in a step which currently reads "<NO CHANGE IN DIGITAL OUTPUTS>", or if you wish to add more output changes to a step where some are already programmed, first position the highlight within the output section of the screen using the cursor-control keys (up-arrow or down-arrow). If some output changes are already programmed within the step, it is not important on which specific output change the highlight is positioned.

Then, press the "Insert" key. A prompt will appear at the bottom of the screen asking you to enter an output number to be programmed for a change. Upon entering the number, and pressing the "Enter" key, you will be asked to program an "on" or "off" state for that output using the "+" or "-" keys, just as when you originally programmed output changes in the initial entry of the step. Quickstep will then re-sort the output changes by output number and display them in order in the output section of the screen.

To insert additional changes, just repeat the process by pressing the "Insert" key again.

To delete output changes (including the global instruction "<TURN ALL DIGITAL OUTPUTS OFF>"), position the highlight over the output change to be deleted. It may be necessary to use the left-arrow and right-arrow keys to accomplish this, as the output changes are shown four to a line. Then, just press the "Delete" key.

Within the output section of the Quickstep screen, the Alt-E ("edit") function performs a very specific task; pressing Alt-E when a given output change is highlighted will change the state of that output. For example, an instruction to turn "on" output #15 ("15+"), will be changed to an instruction to turn that same output "off" ("15-") upon pressing Alt-E.

Editing the Instructions

When editing a step, it is possible to insert or delete programmed instructions (for example, "delay", "monitor", "turn motor" instructions, etc.), or to replace an existing instruction with an instruction of a different type altogether.

To insert an additional instruction, first use the up-arrow or down-arrow key to move the highlight to the instruction *following* where you wish the new instruction to appear. Then press the "Insert" key; a new line will open, moving all subsequent instructions down one line, and you will then be prompted to enter a new instruction.

If you wish to add a new instruction at the end of the step, simply press the down-arrow key to move the highlight *past* the last programmed instruction; this will cause a new line to automatically open without having to press the "Insert" key. Once you have finished entering a new instruction, Quickstep will continue prompting you for additional instructions to add to the end of the step until you press the "Enter" key in response to a newly-opened line.

To delete an existing instruction from a step, position the highlight over the instruction you wish to delete and press the "Delete" key. The instruction will be removed and the remaining subsequent instructions in the step (if any) will be moved upward to fill in the gap.

The replacement of an existing instruction is accomplished using the Quickstep edit function (Alt-E). Position the highlight over the instruction you wish to change, and then press the Alt-E key combination. You will be prompted to enter a new instruction, which will take the place of the old instruction as you enter it. Pressing the "Escape" key midway through this process will cause Quickstep to revert to the previous, unchanged instruction.

Entering a Data Table

One of the most useful resources within DSP-based controllers is the Data Table. The Data Table is simply a two-dimensional array of numbers, stored along with your program, which may be accessed by your program as a source of numeric data.

The uses of this array include:

- The storage of coordinates for "X-Y" tables, robots, and other multi-position devices.

- The storage of parameters for flexible machines capable of making multiple types of products.

- The storage of ASCII messages to be transmitted via the controller's RS-232 port to an external message display, printer, etc.

Further information about the use of Data Tables in DSP programs may be found in the "Guide to Direct Sequential Programming". Additional information about the storage and RS-232 transmittal of ASCII messages may be found in the application note "ASCII Message Transmitting".

Setting up a Data Table

Quickstep allows you to establish, enter and edit a Data Table via a pull-down window. The resultant spreadsheet-like screen allows rapid entry and review of the data within the Table. When developing a DSP program, the Data Table is established by pressing the function key "F8". Assuming a Data Table had not previously been set up within the current program, a prompt will appear at the bottom of the screen:

"No data table currently defined; create one (yes)?"

By pressing the "Enter" key, the default response of "yes" will be entered. Quickstep will then ask you to enter the desired number of rows for the Data Table, followed by the number of columns. Note that for current CTC controllers, a maximum of 32 columns may be specified.

The allowable number of rows is determined by the total Data Table capacity of the controller. This capacity is currently 1724 elements (numbers), with the exception of the model 2800iEA ("Expanded Architecture"), which may store up to 5300 elements. Thus, in the former case, a three-column Data Table may be up to 574 rows long.

IMPORTANT: Because the Data Table intrudes into the program memory space within the controller, it is disadvantageous to program a larger Data Table than that which is required by an application.

Entering the Data Table

After you have configured the size of your Data Table, you will see the "spreadsheet" window open in the center of the screen. Column designations ("001", "002", etc.) will appear at the top of each column, and row designations ("0001", "0002", "0003", etc.) will appear down the left side of the screen.

All elements of the Data Table will initially be set to a value of 0. Note that one row (initially row #0001) will be highlighted with a reverse-video bar, as will one column (initially column #001). You may enter a new value only where these two bars intersect. This is accomplished simply by typing in a number, followed by the "Enter" key.

An existing number may be changed by positioning the highlights to select the appropriate point in the Data Table, and then typing in the new value over the old value.

The cursor control keys (up-arrow, down-arrow, right-arrow and left-arrow) may be used to manipulate the two highlights, allowing you to select any location in the Data Table for entry or editing. Using these keys to force one of the highlights beyond the limit of the Data Table will cause the highlight to "wrap-around" to the opposite side of the Table. The "Home" and "End" keys will position the highlights at the first-row/first-column and last-row/last-column positions, respectively.

Entering ASCII Messages

Quickstep also provides a convenient means of entering ASCII messages into the Data Table, to make use of the message-transmitting capabilities of the models 2200/2400iE/2800iE/2800iEA. These messages must be stored in the form of ASCII-encoded characters, with each character occupying one element of the Data Table. Typically, one message occupies one row of the Data Table, although it is possible to concatenate (string together) message information stored in multiple rows.

To enter a message, position the highlights on the first column of the desired row, and simply type in the message using the alphabetic keys of the computer's keyboard. For each key which is pressed, Quickstep will place the ASCII numeric equivalent in the current location of the Data Table, and will then automatically move to the next column of the current row.

Reading the ASCII Messages

The ASCII character equivalents of the numbers in the currently-highlighted row appear on the bottom line of the Quickstep screen (labelled "ASCII:"), allowing you to read the message which you just entered. All such messages may later be reviewed by scrolling from row to row, while watching the bottom line of the screen.

The data for the current row of the Data Table are represented on this bottom line as follows:

1. All printing ASCII characters are shown literally, except the codes for numeric characters, which are shown in single quotes ('1', '2', '3', etc.).

2. ASCII control codes (values 0 through 31) are shown as their control-key equivalents, using the caret symbol (^), as follows: 1 is represented

Saving, Compiling and Downloading a Program

While you are writing a DSP program using Quickstep, it is extremely important that you periodically save your program to disk. While you are working with Quickstep, your program is maintained in the computer's RAM memory. If anything should happen to the computer while you are in Quickstep (including a power outage!), any part of your program which had been entered since the last time you saved the program to disk would be lost completely. By saving your program frequently (perhaps after entering each new step...) you minimize the risk of such a loss.

This chapter describes the process of saving your program, as well as the steps required to transfer your finished program into the controller for execution.

Saving your Program

At any point in the development of your program (except in the midst of entering an instruction) you may save the program to any of the computer's disk drives. Because we highly recommend that you do this frequently, it has been made an easily-accomplished task from within Quickstep.

To save your program, press the "F2" function key (the "FILE" menu). Typically, if you have made any changes to your program since the last save, Quickstep will automatically highlight the "Save" option on the FILE menu. If another option happens to be highlighted, the Save option may be selected either by pressing the "S" key, or by using the up-arrow/down-arrow keys. When the Save option has been highlighted, press the "Enter" key.

A prompt will now appear at the bottom of the screen. If you had not yet saved your program to disk, the prompt will read:

"Save filename (no default):"

Quickstep is asking for a name for your program. Enter a name, perhaps relating to the machine which you are automating, up to eight characters long. Quickstep will automatically add the filename extension ".DSP" to the name you enter.

Once you have initially entered a name (or if you had loaded in a previously-written program), Quickstep will keep your original name associated with the program. From that point on, each time you execute a "Save", Quickstep will prompt with:

"Save filename (yourprog.dsp):"

...where "yourprog" is the name you originally assigned. If you wish to maintain this same name for your program, simply press the "Enter" key, and your program will be saved again under that name. This means that, typically, to save your program at any time during program entry, all you need do is press the "F2" key, followed by pressing the "Enter" key twice.

If, however, you wish to create a new file under a different name, leaving your previous program on the disk intact, then type in a new filename before pressing "Enter". In this manner, you may easily create duplicate files, or create several slightly different versions of your program, without having to re-enter the entire program.

The Result of Saving your Program

When you save your program to disk, Quickstep will create the following files on disk (the filename "yourprog" indicates the name which you have given your program):

"YOURPROG.DSP" - This is an ASCII text file containing the body of your program. The text in this file will be virtually identical to the appearance of the steps you have written as they appear within Quickstep. Note that it is possible to use a word processing program to write and/or modify programs which will later be compiled by Quickstep (see "Other Ways to Use Quickstep").

"YOURPROG.FIG" - This file contains the "configuration" of your program, specifically the information contained within the PARAMETERS menu of Quickstep. This is saved along with your program to avoid having to re-enter the information each time you load your program into Quickstep.

"YOURPROG.TAB" - If you have created a Data Table as part of your program, Quickstep will store the contents of that Data Table in a file with the filename extension ".TAB". This, too, is an ASCII text file, and may be created with a word processor if desired.

"YOURPROG.#SP" - When you save your program for the second, or subsequent, time, Quickstep will save the previous version of your program in a backup file with the filename extension ".#SP".

"YOURPROG.#AB" - As with the main body of your program, Quickstep will create a backup file for your Data Table, with the filename extension ".#AB", during the second, or subsequent, saves to disk.

Note that it is possible to recover disk space by deleting all backup files with a single command:

"del *.#*"

If you elect to do this, do it very carefully. A good practice is to keep a separate, archival backup disk, and to update that disk prior to the deletion of the Quickstep backup files. Some people keep multiple backup disks, which they update on a rotating basis to guard against the possibility of corrupted data on a disk being propagated throughout all outstanding backup copies.

Compiling Your Program

When you have completed your program, Quickstep requires one additional step before loading your program into a controller for execution. This step is called "compiling", and it converts the text file which is created by Quickstep into the codes necessary for the controller to interpret and execute your program.

During the compiling process, Quickstep will also check your program for certain structural errors, and will warn you about such errors. A "DEBUG" function is accessible using the "F5" function key which, after compiling your program, will provide you with very specific error messages to facilitate the repair of the problems found by the compilation. This process is described further below.

To compile a program file, press the "F2" function key to open up the FILE menu. If you have not made any changes to your program since the last "Save", the "Compile" option will be highlighted at this time. (If the Save option is highlighted, save your program prior to continuing with the compilation process.) Press the "Enter" key; Quickstep will now begin the compilation of the file, displaying a message on the prompt line at the bottom of the screen which will indicate the status of the compilation. Typically, this process will not take longer than a few seconds.

The final message will indicate the filename compiled, along with the number of steps it contained, the percentage of the controller's memory which will be used by it, and the number of errors found, if any.

During compilation, Quickstep will create the following new file(s):

"YOURPROG.DSO" - This file, with the extension ".DSO", contains the binary information which will ultimately be loaded into a controller. It is not human-readable.

"YOURPROG.ERR" - If any errors were found during compilation, a file with the extension ".ERR" is created containing information to help Quickstep find the errors when you subsequently use the DEBUG function. This file is not human-readable.

"YOURPROG.LST" - If errors were found during compilation, an additional file with the extension ".LST" is also created. This file contains explicit error messages and locations in the form of an ASCII text file, which may be printed out or listed on the computer's screen as you would any ASCII file. Note, however, that you will typically use Quickstep's built-in DEBUG provision, instead.

The "DEBUG" Function

If, during compilation, Quickstep identified errors within your program, you may use the built-in DEBUG function to quickly locate these errors. To use this function, you must have the program you wish to debug loaded into Quickstep (as you normally would during compilation). After compiling your program, press the "F5" function key to trigger the DEBUG function.

Quickstep will then automatically change to display the step containing the first error which was located during compilation. The specific instruction containing the error will be highlighted, and an explicit error message will appear at the bottom of the screen. You may either make a note of the problem, or fix the error on the spot, prior to continuing with the DEBUG process.

You may then continue to press the F5 key to locate any subsequent errors until Quickstep informs you that no additional errors were found.

Once you have repaired the errors, note that it will now be necessary to compile your program once again prior to downloading it into a controller.

Downloading Your Program

Once you have successfully compiled your program, it is now possible to download it into a controller. To accomplish this, you must first connect your computer's RS-232 port to the controller's RS-232 port. This should be done using the adapter and cable provided with Quickstep for this purpose.

Two adapters are provided with Quickstep; one is for use with 9-pin "D" connectors, which are frequently being used in newer computer designs for RS-232 ports, the other is for use with the traditional 25-pin "D" connector. Plug the appropriate adapter into your computer, then connect the "modular" (telephone-type) cable provided with Quickstep between this adapter and the controller's RS-232 port.

Power must be applied to the controller for successful downloading!

It is now possible to download your program. Pull down the FILE menu by pressing the F2 function key. If you have just completed a successful Compile operation, the Download option will automatically be highlighted (otherwise, ask yourself if you shouldn't really do another Save/ Compile operation prior to Downloading).

With the Download option highlighted, press the "Enter" key. Quickstep will now display a prompt at the bottom of the screen:

"Download filename (yourprog.dso):"

...where "yourprog" is the name of the program file currently within Quickstep. If this is the program you wish to download, simply press the "Enter" key. Quickstep will then display a status message as the program is being downloaded; if transmission errors are detected during this process, an error message will be displayed indicating the nature of the problem.

After a successful Download, a message will appear indicating the number of steps downloaded, etc. This is an indication that your program was transferred successfully.

Problems during Download

The single most trouble-prone area when working with non-standard (i.e.; non-IBM®, or "clone") computers is in the access to RS-232 communications ports. If you have difficultly establishing communications between the computer and the controller, please check the following:

1. Port Designation
The RS-232 port to which you are connected on the computer must be internally designated as "COM1" to allow Quickstep to find it.

2. Port Wiring
There are standards for the wiring of connectors used for RS-232 ports. Insure that the connector on the computer you are using is wired correctly. Refer to the wiring charts shown in your computer manual and in the Installation Guide for your controller for verification.

3. "DCE" Configuration
RS-232 ports are wired in one of two configurations: "DCE" or "DTE". The only real difference between these two wiring schemes is that the receive and transmit lines (along with certain control/status lines) are reversed from one another. The purpose here is to allow a "DCE" device to be directly connected to a "DTE" device using a cable which simply connects pin #1 to pin #1, pin #2 to pin #2, etc.

Some communications boards which are available to plug into computers may be configured via jumpers to be either a DCE or DTE device. These must be set to the DCE position for proper operation with the adapters supplied with Quickstep.

If, for any reason, your computer provides a hard-wired DTE configuration, an inexpensive device known as a "null-modem" may be used to reconfigure the signals. The null-modem is simply a pair of connectors interwired so as to switch the receive/transmit lines, as well as the pairs of control signals.

4. Computer Compatibility
It is possible that your computer is inherently incompatible. We have already determined that certain lap-top computers contain firmware bugs which don't allow Quickstep to communicate properly (see page 7). If you suspect a problem here, please contact CTC for further information.

Using Quickstep Diagnostics

One of the most important requirements in the development of a new, highly-sophisticated automated machine is the availability of strong diagnostic and troubleshooting capabilities within the design tools being used. In the creation of Quickstep, we have attempted to provide you with a comprehensive set of such tools, to decrease the time required to bring up a new machine.

This section describes the diagnostic functions available within Quickstep, and provides some suggestions as to how they may be effectively applied.

The "MONITOR" Menu

Within Quickstep, the F7 ("MONITOR") function key triggers a pull-down menu which provides a number of options for monitoring the operation of an attached controller. You select one of these options either by pressing the first letter of the option's name (i.e.; press "R" for "Registers") or by using the up-arrow/down-arrow keys to highlight the appropriate choice; pressing "Enter" will then execute that selection. The computer will then begin continuous polling of the controller, and Quickstep will create a window within the screen which will display the desired information.

In summary, the choices offered by the MONITOR menu are as follows:

Program Status - It is often desirable to discover exactly which step(s) the controller is executing at any given time. The "Program Status" selection allows you to view a diagram illustrating the "hierarchy" of all steps currently being executed by the controller. In addition, the controller status (i.e.; "running" or "stopped") is shown and, if the controller has sensed one of 25 different types of software faults, it will provide an explicit error message and highlight the step containing the problem.

Configuration - This choice allows you to view the actual I/O configuration of the controller (quantity of I/O, motor axes, etc.).

Registers - An extremely useful selection, this choice allows you to view any of the numeric registers within the controller (including the special purpose registers), *and* to force new values into any of these registers. In addition to its use for monitoring purposes, this provision is often used to "fine tune" various parameters on a new machine while it operates.

Flags - All 32 Flags within the controller may be monitored and, if desired, toggled between their "set" and "clear" states.

Digital Inputs - The status of the inputs within the controller may be observed. This selection is useful for troubleshooting limit switch wiring problems.

Digital Outputs - The status of the controller's outputs may be observed using this selection, and any of the outputs may be forced on or off.

Servos - A servo diagnostic window provides full status information for all servos within a controller, including the servo's actual position, the instantaneous servo error, and the status of the servo's control inputs. This provision is very useful for tuning in a new servo system.

Analog Inputs - Groups of analog inputs within the controller may be monitored in real time.

Analog Outputs - The current values stored to the analog outputs within the controller may be viewed, and new values may be forced (thus changing the analog output voltage).

How Monitoring Works

When one of the above selections is made from the F7 (MONITOR) menu, the computer begins transmitting a continuing series of RS-232 commands to the controller, requesting the appropriate data. This data is then displayed within a window on the screen of the computer. The controller may be either stopped or running its program; this data transfer will not interfere with the normal operation of the controller (note, however, that in critical applications there may be a perceptible decrease in the controller's execution speed).

The controller must, of course, be turned on and properly connected to the computer's RS-232 port with the same adapter and cable used for programming the controller. If, for any reason, Quickstep is unable to communicate properly with the controller, an error message will be displayed at the bottom of the screen.

When using one of the above menu choices to force a new value in the controller (for example, changing a register value, forcing an output on or off, etc.), Quickstep will send a single command to make the desired change, and then resume monitoring. Subsequent commands within the DSP program being executed by the controller will still be able to alter the resource being monitored, perhaps even changing it back to its original status.

"Program Status" Monitoring

In debugging a newly-written program, there are often instances where the program does not seem to follow the intended path. In such instances, it can be extremely helpful to determine exactly which steps are currently being executed by the controller so the program may be corrected. Program Status monitoring provides a means of determining this quickly.

In addition, there are times when the controller will detect a fault within your program which could not be detected during its compilation within Quickstep. Perhaps you referenced an input number which does not exist within your controller, or perhaps a cable is improperly connected to a thumbwheel switch; problems of this nature may be

"Program Status" Monitoring Screen

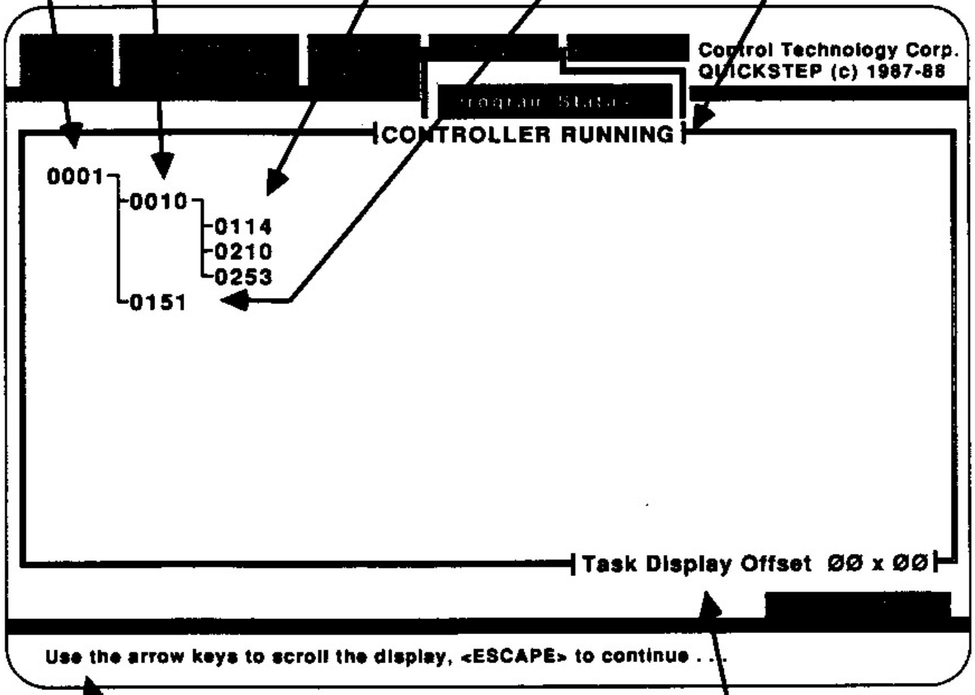
The Primary Task
This task is suspended, waiting for two subsidiary tasks to complete.

Subsidiary Task
This task is suspended at step #10, where a "DO" instruction has called up three additional tasks.

Three Active Tasks
These tasks are executing at steps #114, #210 and #253, having been called up from step #10

Active Task
This task, called up from step #1 along with the task now suspended at step #10, is currently executing step #151.

Controller Status
Indicates whether controller is running or stopped.



```
                                                   Control Technology Corp.
                                                   QUICKSTEP (c) 1987-88
                                  Program Status
 ----------------------------------|CONTROLLER RUNNING|----------------------
   0001┐
       ├0010┐
       │    ├0114
       │    ├0210
       │    └0253
       └0151




 ----------------------------------------|Task Display Offset ØØ x ØØ|-

 Use the arrow keys to scroll the display, <ESCAPE> to continue . . .
```

If many tasks are running, and the status display runs off the edge of the screen, the arrow keys (up, down, left, right, home and end) may be used to scroll the display. To cease monitoring, you may press either the ESCape key, or press the F7 function key again.

"Task Display Offset" indicates the current position of the viewing screen relative to the display of task data. This is always "00 x 00" unless you use the arrow keys to scroll the display.

detected by the controller, resulting in a "software fault" and a subsequent automatic shutdown. In this case, Program Status monitoring will provide you with a specific error message, while highlighting the step number in which the fault was first detected.

Using Program Status Monitoring

When you select "Program Status" from the "F7 MONITOR" menu, Quickstep will create a large window occupying nearly the entire screen, as shown in the illustration. In the upper border of this window will be shown the current status of the controller (i.e.; either "RUNNING" or "STOPPED").

Within the window, in the upper left corner, will be shown the step number being executed by the primary task within the controller. If you are not using multi-tasking within your program, this will be the only number shown in the window. If you wish, you may then return to viewing your program within Quickstep, to determine what instructions are being executed at that step number. This may be accomplished by pressing the ESCape key twice (once to make the window disappear, and once to make the MONITOR menu disappear).

If, however, multi-tasking is in progress, the window will show not only the primary task, but also any tasks being executed subsidiary to the primary task. This is shown in the illustration; a line is drawn from the primary task to any tasks running subsidiary to it, and the step numbers being executed by those tasks are shown. If any of these subsidiary tasks have, in turn, called up additional tasks, these are shown in a similar fashion.

Note that, when a task has other tasks running subsidiary to it, that task is suspended until the subsidiary tasks are "done". This is in accordance with the way multi-tasking works, as described in the "Guide to Direct Sequential Programming".

There may be instances where the levels of tasks being executed will no longer fit within the window, either in width or depth. If this happens, you may view the additional tasks by scrolling with the arrow keys (the "Home" and "End" keys also work). The "Task Display Offset" shown in the lower right border of the window will indicate that the screen has been scrolled.

Program Error Indication

As mentioned above, there are a number of errors which might exist within your program which cannot be detected until actual execution of the program within the controller. When the controller detects such an error, it will immediately halt execution of your program, although communications will still be allowed.

If you are monitoring the controller's status when such an error occurs (or begin monitoring it subsequently), the computer will begin chirping (yes, chirping) and a specific error message will appear at the bottom of the screen. In addition, if the error relates to a specific step of your program, that step number will be highlighted in the task display within the monitoring window. This will assist you in quickly finding the source of the problem and correcting it.

"Configuration" Monitoring

This selection from the "F7 MONITOR" menu allows you to view the quantities of various resources (e.g.; inputs, outputs, etc.) contained within your controller. Upon making this selection, a window will open within the Quickstep screen showing a list of resources, similar to the list within the PARAMETERS menu. To the right of each resource name will be shown the quantity of that particular resource within the controller. Note that some earlier versions of certain controller models will not be able to report all of these parameters; in such instances, an asterisk (*) will appear in place of the quantity.

This configuration window, unlike the other MONITOR provisions, will not continuously update. There is no need for such an update as the quantity of I/O should never change while the controller is still energized (as explained in the Installation Guides for the various I/O boards, power should *always* be removed from the controller prior to removing or inserting I/O boards).

"Register" Monitoring and Modification

One of the more frequently-used choices on the MONITOR menu is the "Registers" selection. This choice allows you to monitor any group of 16 numeric registers within the controller, viewing them dynamically as they change under program control. You may also force changes to any of these registers simply by typing in a new value.

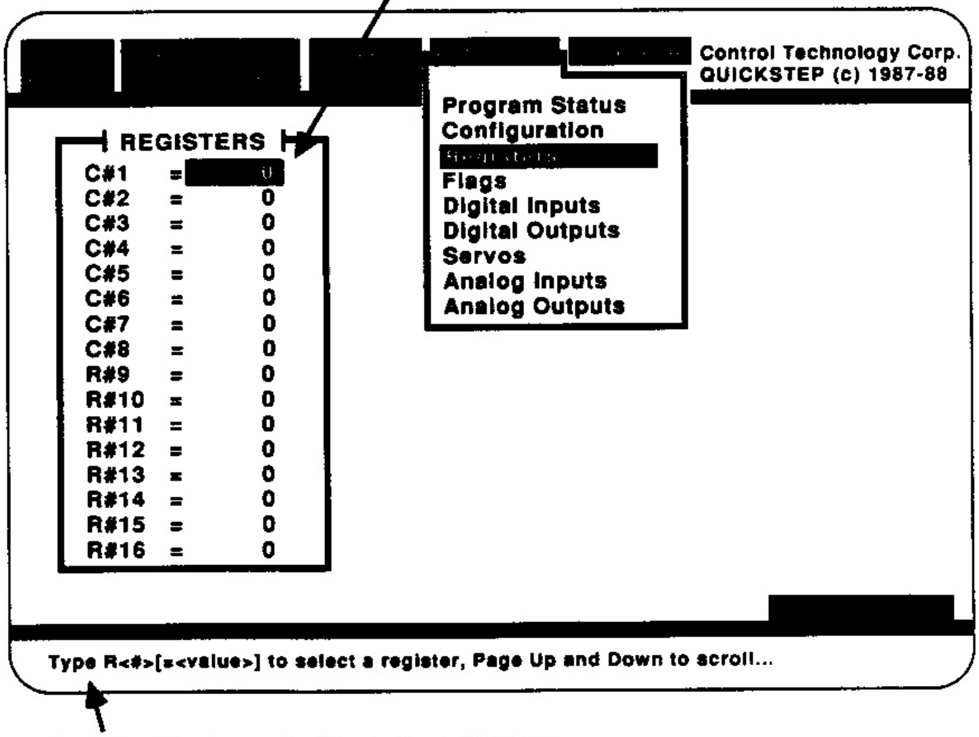
When you select "Registers" from the MONITOR menu, Quickstep will create a window showing the current contents of the first 16 registers in the controller (note that the first 8 registers are designated as counters - "C#1" through "C#8", reflecting their dual role within the controller). These values will be continuously updated via constant communication with the controller. The value within the first register (C#1) will be highlighted on the screen, indicating that you may change the value within this register; this is accomplished by simply typing in a new value, followed by the "Enter" key.

To select a different register for modification, you may use the up-arrow/down-arrow keys to move the highlight among the sixteen registers within the window. To view a different group of registers, type the letter "R", followed by the number of the first of sixteen sequential registers you wish to view, then press the "Enter" key. For example, if you wish to view reg#501 through reg#516, type "R501", then press "Enter".

When you modify a register value using this provision, the change will not take effect until you have pressed the "Enter" key following the entry of the new value. If, prior to pressing the "Enter" key, you decide to terminate the change, you may do so by pressing the ESCape key. The value shown on the screen will revert to its original value, and *no*

"Registers" Monitoring Screen

The current value of all 16 registers are continuously updated during monitoring. The highlighted value may be changed simply by typing in a new value, followed by the "Enter" key.



```
REGISTERS
C#1  =    0
C#2  =    0
C#3  =    0
C#4  =    0
C#5  =    0
C#6  =    0
C#7  =    0
C#8  =    0
R#9  =    0
R#10 =    0
R#11 =    0
R#12 =    0
R#13 =    0
R#14 =    0
R#15 =    0
R#16 =    0
```

```
Type R<#>[=<value>] to select a register, Page Up and Down to scroll...
```

To select a different range of registers to view, type the letter R, followed by the number of the first register in the desired range. For example, to view reg#501 through reg#516, type "R501", followed by the "Enter" key.

change will have been made to the controller's registers.

Using Register Monitoring

Access to the controller's registers while its program is operating is a powerful tool, offering a potentially-valuable window into the operation of your machine. Although this provision has a variety of applications, one of the primary uses of the Quickstep register monitoring provision is to allow a new machine to be "tuned-in" quickly and easily.

Often, when working with a new machine design, there may be uncertainty as to the proper values for various time delays, motor motion parameters, analog setpoints, and other critical parameters. If, when writing your original program for a new machine, you program these parameters to be derived from various numeric registers (instead of programming fixed values), you may later use Quickstep to modify these register values until your machine is operating properly.

As an example, stepping motors have historically been considered time-consuming to tune initially. A number of parameters, including base speed, maximum speed, and accel/decel rates, must be adjusted for optimum operation, and safety margins must be determined to avoid "over-driving" the motor. Without the proper tools, the repetitive "trial-and-error" adjustments required to determine the appropriate values would require a great deal of time.

If, however, you write your program to derive these parameters from the controller's numeric registers, you may then use Quickstep's register monitoring provision to modify the parameters *while the machine is running*. To do this, your program should contain an instruction such as:

```
        profile motor#1  (half)  basespeed=reg#501
   maxspeed=reg#502  accel=reg#503  decel=reg#504
```

This instruction should be placed so that it will be executed each time a new motor motion is commenced. In doing so, you will insure that the controller will update the motor parameters, based on the then-current values in registers #501 through #504, for each new motion.

After downloading the program, you may then use Quickstep to monitor registers #501 through #504, entering new values for these registers and watching the operation of the stepping motor. Through the entry of a series of such values, you may quickly determine both the limits of the motors capabilities, and the optimum operating values for your application. If you wish, you may then edit your program to contain these values on a permanent basis.

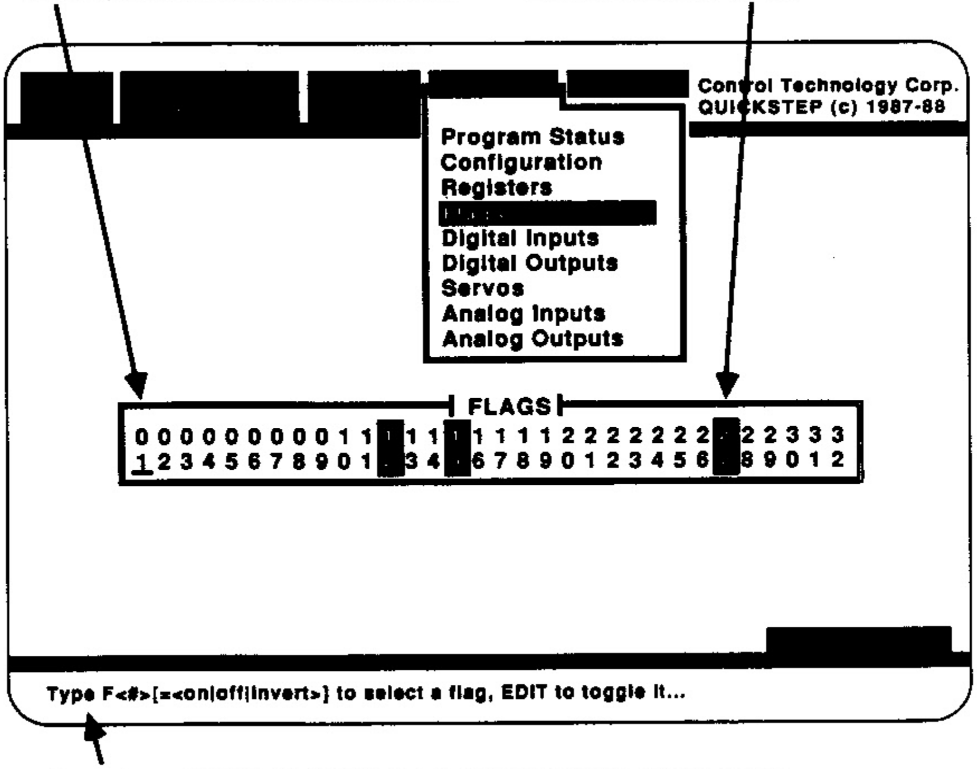
"Flag" Monitoring and Alteration

Each controller contains 32 "Flags", which are simply memory elements useful for storing binary (yes/no) information. These flags are always in one of two states, called SET and CLEAR. The flags are sometimes used within a program to trigger certain sequences, to store

"Flags" Monitoring Screen

All 32 flags are shown simultaneously. A blinking cursor may be moved to select a flag for modification.

Flags which are currently SET are shown as highlighted; all others are CLEAR.



```
                                        Control Technology Corp.
                                        QUICKSTEP (c) 1987-88

                          Program Status
                          Configuration
                          Registers
                          Flags
                          Digital Inputs
                          Digital Outputs
                          Servos
                          Analog Inputs
                          Analog Outputs

              ---------------------| FLAGS |---------------------
               0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
               1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2

Type F<#>[=<on|off|invert>] to select a flag, EDIT to toggle it...
```

To select a specific flag for modification, you may either use the right-arrow/left-arrow keys to scroll to that flag, or type the letter F, followed by the flag number. Flags may be modified either via a typed command, or by using the EDIT (Alt-E) function.

information (i.e.; "good-part/bad-part") for future use, or for communicating between tasks.

The "Flags" choice on the MONITOR menu allows you to view the state of all 32 flags, and also allows you to "toggle" the state of any flag between SET and CLEAR. As shown in the illustration, the 32 flags are shown from left to right across the screen, each flag being designated by a two digit number. This number is shown with the tens digit on top of the ones digit (for example, flag #27 is shown with the number "2" over the number "7").

If a flag happens to be SET, it is highlighted (reverse video) on the monitoring screen; if CLEAR, it is shown normally. When you first select the "Flags" choice from the MONITOR menu, the first flag (#01) will be blinking. This flag is now selected for possible alteration. To select a different flag, use the right-arrow/left-arrow keys to move the blinking cursor to a different position.

If you wish to change the state of a flag, first select it with the blinking cursor, then press the Alt-E (edit) key combination. Quickstep will then transmit a command to the controller to change the state of the selected flag. This new state will then be shown within the monitoring screen.

It is also possible to alter a flag using a typed command. For example, to set flag #15 you may, while viewing the flags using the MONITOR function, type the command "F15=on", followed by the "Enter" key. The blinking cursor does not have to be positioned over flag #15 to accomplish this. Similarly, you may use the command "F15=off" to clear the flag, and the command "F15=invert" will toggle it.

Simply typing "F15", followed by the "Enter" key, will instantly move the cursor to the position of flag #15.

Using Flag Monitoring

In troubleshooting a new machine, it is often useful to be able to monitor the flags, especially if they are being used to determine the course of your program. When flags are used to store information, to be used in a later part of the program, the status of the flags may be critical to a full understanding of the operation of the program.

Another use of the flag monitoring provision is to provide you with a temporary means of altering the course of your program manually. For example, you may be running a test program on a new machine, into which you have programmed "monitor flag" instructions which allow you to alternatively skip around or execute certain portions of the program depending on the state of certain flags. Then, by using flag monitoring to manually alter the relevant flags, you may quickly select the desired operations to be executed within a given test.

Many other such creative uses of the flags are possible, and the MONITOR menu provides the access necessary to make full use of them.

"Digital Inputs" and "Digital Outputs" Monitoring

These selections from the MONITOR menu allow you to view, respectively, any of the Digital (e.g.; limit switch, etc.) Inputs or Digital (e.g.; solenoid valve, etc.) Outputs within the controller. When you select one of these choices from the MONITOR menu, a window will appear within the screen showing up to 32 inputs or outputs. These are shown in a format similar to the "Flags" monitoring screen, except the inputs and outputs are designated with 4 digit numbers to allow for the large I/O quantities possible with expanded versions of larger CTC controllers.

When viewing Digital Inputs, each input which is sensing a contact closure to ground will be shown as highlighted on the screen. You will therefore be able to watch the inputs change state as your machine operates. If your controller contains more than 32 input points, you may scroll left and right to view the additional inputs by using the page-up and page-down keys, respectively.

Similarly, when monitoring Digital Outputs, each output which is currently "on" will be shown as highlighted on the screen. Again, the page-up/page-down keys may be used to scroll the display left and right if your controller contains more than 32 outputs.

It is also possible to "force" outputs on and off. To do so, you must first move the flashing cursor to the output which you wish to change. This is accomplished through the use of the left-arrow/right-arrow keys. To change (toggle) the current state of the selected output, press the Alt-E key combination. Quickstep will then send a single command to the controller to change that output. Note, however, that the output will only remain in its new state until the program running within the controller instructs it to change once again.

Using Digital I/O Monitoring

One of the primary uses of Digital I/O monitoring is to check the wiring and functionality of the input and output devices connected to the controller. For example, with a newly-wired machine, it is a good practice to insure that the various sensors and limit switches on the machine are connected properly to the intended inputs of the controller. This may be accomplished by using Quickstep's input monitoring provision, and having another person manually (if possible) actuate each of the sensors individually. You should be able to see the appropriate input designators highlighted on the computer screen as each sensor is actuated.

Similarly, the outputs may be tested by forcing the outputs on, one at a time, using the Quickstep output monitoring provision. In each instance, the appropriate actuator on the machine should react.

The Digital I/O monitoring may also be used as a troubleshooting aid to determine the cause of a machine malfunction. This is particularly valuable if you (and the computer) are physically remote from the machine being controlled. If you have wired an RS-232 link from the machine to your computer, Quickstep allows you to view the status of the machine without having to walk to the machine's control cabinet. (Note that the individual I/O boards within the controller will have LED indicators for each I/O point. These may also be used for troubleshooting.)

"Servos" Monitoring

This selection from the MONITOR menu allows you to view a window containing status information for each of up to 8 servo control boards within your controller. Highly useful for both tuning and troubleshooting, this window provides the instantaneous position of each servo, its instantaneous error, and the status of each of the auxiliary inputs to the servo board.

As with all of the programmable instructions for servos, the position and error information is shown in units of "steps", where each step corresponds to one "edge" from the encoder providing the servo feedback. The auxiliary inputs are shown as highlighted if a contact closure is present on that input, otherwise the text designating each input is shown normally.

This servo information is continually updated via communication with the controller, allowing you to observe the operation of the servo system while a motion progresses.

Using Servo Monitoring

One of the primary purposes of servo monitoring is to greatly ease the task of tuning a servo system to insure optimum operation. Servo systems often suffer from poor performance due to overshoot and instability; sometimes the detection of such inadequacies is difficult without the proper troubleshooting tools. Quickstep's servo monitoring provision allows you to "watch" the internal performance of the servo loop while the servo is operating, providing you with the information you need to detect and correct problems quickly.

The chief cause of poor performance in servo systems is attempting to drive the motor (the term "motor" is used generically for any actuator within the servo loop) beyond its capabilities. This results in "following error", an error which gradually increases throughout the servo motion. This following error can ultimately cause an overshoot at the end of the motion, as the servo system attempts to compensate for the accumulated error by applying excessive drive to the motor.

By observing the servo error throughout a motion, you will instantly be able to determine if error is building throughout the acceleration period, or throughout the entire motion. This will indicate the need to either reduce the programmed accel rate or the programmed maximum speed to a figure which is within the motor's capabilities (or, alternatively, get a bigger motor). If you have programmed the "profile servo" instruction to derive these parameters from numeric registers within the controller, you may even use Quickstep's register monitoring to change these values without having to reprogram.

"Analog Inputs" and "Analog Outputs" Monitoring

The analog I/O selections from the Quickstep MONITOR menu provide a way to observe the readings of analog sensors connected to the controller's analog inputs, as well as the values being commanded on the controller's analog outputs. Upon selecting either "Analog Inputs" or "Analog Outputs" from the MONITOR menu, a window will appear on the screen containing the status of up to 16 analog inputs (or outputs). This status display will be continually updated with new information from the attached controller.

If your controller contains more than 16 analog I/O points, you may view additional points by using the Page-up/Page-down keys. When monitoring analog outputs, new values may be forced into the outputs by scrolling the highlight (cursor) to the value you wish to change. Typing in a new value, followed by the "Enter" key, will result in Quickstep sending a single command to the controller to send that value to the specified analog output. (Note that the effect of this override will only last until the controller's program stores a different value to this analog output.)

If you wish to abort the entry of a new value for an analog output, prior to pressing the "Enter" key, simply press the ESCape key; your entry will disappear and the value on the screen will revert to the scanned reading from the controller. No command will have been sent to the controller.

A Guide to Instruction Names

cancel (cancel all other tasks)

clear (clear flag #XX)

count (up / down, ctr #X)

delay (time delay)

disable (ctr #X)

do<ENTER>

done

enable (ctr #X)

goto (formerly called "branch")

if ("if reg-10 >= twhl-5 goto 150")

monitor (**b**oolean)

monitor **fl**ag (set / clear)

monitor (**i**nputs)

monitor **s**ervo (stopped / running)

monitor **m**otor (stopped / running)

profile **m**otor

profile **s**ervo

reset (ctr #X)

rotate (flags)

set (set flag #XX)

search (and zero motor/servo #X)

shift (flags)

start (ctr #X)

stop

stop **m**otor

stop **s**ervo

store ("store ctr-1 * 50 to dis-1")

test and set (flag #XX)

turn **m**otor

turn **s**ervo

zero **m**otor

zero **s**ervo

Quickstep Control and Command Keys

cursor control: up-arrow, down-arrow, home, end
viewing long instructions: right-arrow, left-arrow
editing: Ins, Del, Alt-E (="edit"), Shift-Ins (= "append")
next step: PgDn
previous step: PgUp
next step with programmed info: Ctrl-PgDn
previous step with programmed info: Ctrl-PgUp
first programmed step: Ctrl-Home
last programmed step: Ctrl-End

Optional Command Keys

Quickstep supports a number of alternative key sequences for the control of the cursor position and for editing. These may be more convenient for people who are proficient typists, or who are used to control key sequences used in other software packages. The optional key sequences are shown below:

Key Command	Equivalent	Meaning
Ctrl-A	Ctrl-LeftArrow	Move to beginning of line
Ctrl-B	LeftArrow	Move left (backward) one character (or output)
Ctrl-C	Ctrl-Break	Quit Quickstep
Ctrl-D	Del	Delete item under cursor
Ctrl-E	Ctrl-RightArrow	Move to end of line
Ctrl-F	RightArrow	Move right one character (or output)
Ctrl-G	End	Move to bottom of current step
Ctrl-H	Backspace	
Ctrl-I	Tab	
Ctrl-J		New line.
Ctrl-K		Delete text to end of line.
Ctrl-L		Redraw screen.
Ctrl-M	Enter	
Ctrl-N	DwnArrow	Move to next line
Ctrl-O	Ins	Insert a line, step, or output
Ctrl-P	UpArrow	Move to previous line
Ctrl-R	\	Search backwards (Reverse)
Ctrl-S	/	Search forwards
Ctrl-T	Home	Move to top of current step
Ctrl-U	PgUp	Go to previous step
Ctrl-V	PgDn	Go to next step
Ctrl-Z		Zuspend (suspend Quickstep, start new DOS)
1	Ctrl-Home	Go to first programmed step
Ø	Shift-Ins	Append
+	Enter	Start new line, start new step
-	UpArrow	Move to previous line
Esc	Home	Move to top of page
Alt-E		Edit item under cursor
Alt-S		Enter "Step mode"
Alt-O		Enter "Output mode"
Alt-C		Enter "Comment mode"
Alt-I		Enter "Instruction mode"
Ctrl-PgUp		Go to next programmed step
Ctrl-PgDn		Go to previous programmed step

Guide to Quickstep Function Keys

This appendix outlines the functions performed by the Function Keys ("F1" through "F10"), as used within Quickstep. It is intended both as a summary guide to their functions, and a reference to locate additional information.

F1: The HELP Function

This key may be used at any time during the use of Quickstep to obtain further information about what to do next. The HELP key is almost always active (except in the midst of entering a numeric parameter, or during the execution of certain other functions). Upon pressing the F1 key, a window will appear within the screen containing text which should prove helpful.

To make the HELP screen disappear, either press the F1 key again, or press the ESC key (Note: when using HELP to select a DSP instruction, you may simply press the first letter of the selected instruction name; the window will disappear and the letter will be registered appropriately).

F2: The FILE Menu

The function key F2 triggers a pull-down menu, with a number of choices allowing you to perform certain file-related tasks (i.e.; saving or reading a DSP program, compiling your program, etc.). These choices are outlined below.

When you press the F2 key, Quickstep will attempt to guess which of the functions below you wish to perform (there is, after all, a logical order to them: List, Save, Compile, Download). The menu will appear with this function highlighted with a reverse-video cursor. To select that choice, simply press the Enter key.

Otherwise, you may move the cursor to the appropriate choice using either of two methods; the up-arrow and down-arrow keys allow you to move the cursor up and down within the menu, or, you may press the first letter of the choice you desire (for example, the letter "s" for Save). Note that each of the choices on the menu starts with a unique letter. When the cursor is located on the command you wish to execute, press the Enter key.

"New" - This menu choice starts a new program. If there is an existing program within Quickstep which you have modified since the last time it was saved to disk, you will be asked if Quickstep should save the program prior to "dumping" it to start a new program.

"Save" - *Use this selection a lot!* This will save your program to disk, first asking for a name under which to save it. If you had previously saved the program under a given name, that name will be presented to you as a default. You may save under that name simply by pressing the "Enter" (or "carriage return") key, or you may elect to save under a new name by typing the new name, optionally including a DOS pathname (e.g.; "a:\myfiles\machprg").

When saving a program, Quickstep creates a file with the extension "*.DSP" (for example, "machprg.dsp") containing the step information you have programmed. This is stored in the form of an ASCII text file. If you have programmed a data table, it is stored separately in a file with the extension "*.TAB" (for example, "machprg.tab"). The configuration for your program (more about this later...) is stored in a file with the extension "*.FIG" (e.g.; "machprg.fig").

If a copy of your program had already existed within the same directory when you perform the "Save", Quickstep will preserve the old copy as a backup, changing its filename extension to "*.#SP" for the "*.DSP" file and "*.#AB" for the "*.TAB" file.

By saving your program frequently, you protect yourself against the possibility of power failures, system crashes and accidents which may otherwise result in lost data.

"Read" – This command tells Quickstep to read a specified DSP program file from disk, bringing it up on the screen for editing, etc. You will be asked for a file name, and you may enter the name, along with any necessary DOS pathname (e.g.; "a:\myfiles\machprg"). Note, however, that a far easier method of reading in a file is to use the "List" command described below.

"List" – The List command performs two functions. First, it will display a listing of all DSP program files within a given directory (you are able to specify the DOS directory to be used). Then, you may move a cursor to any of the files shown and, by pressing the Enter key, that file will be read into Quickstep.

This is a fast, convenient way to read in a pre-existing file from disk.

"Print" – This command takes the program file with which you are working and places it in the queue for the DOS "PRINT" command. This PRINT command then works in the background to print a listing of your program, while you continue editing with Quickstep. Note that, in order for this selection to work properly, the DOS utility "PRINT.COM" must be within the DOS search path (see chapter 3 for a discussion of the search path).

"Compile" – The form in which your program is saved to disk is an ASCII text file, and this file must be converted into encoded commands prior to downloading it to a CTC controller. The Compile command performs this conversion, creating a file with the extension "*.DSO" which is then stored to disk. This command also checks your program for certain types of errors and, if any exist, creates two additional files with the extensions "*.ERR" and "*.LST".

The file with the "*.ERR" extension is not human-readable, but rather allows Quickstep's "DEBUG" function to find and report the errors from within Quickstep. The file with the "*.LST" extension is a text file which you may read by "type"ing or "print"ing it (via DOS), which contains a detailed description of the location and types of errors present within the program.

"Download" – Once your program has been successfully compiled, it may be downloaded to a CTC controller. Specific instructions for accomplishing this are to be found in chapter 6.

"Zuspend" – Occasionally, you may find it convenient to be able to put on "hold" everything you are doing within Quickstep, go back to DOS to look at files, print out an unrelated file, copy something to a floppy diskette, etc., and then return to Quickstep exactly where you left off...

The Zuspend command allows you to do exactly that. This command maintains Quickstep, along with your program, in the memory of the computer, and starts up a "new" copy of DOS. You are presented with a DOS prompt, at which you may perform virtually any operation allowed by DOS (be aware, however, that the computer's memory is now somewhat limited by the still-active Quickstep).

To return to Quickstep, type the command "exit" at any DOS prompt. *Do not "restart" Quickstep by typing the command "qs"; this will begin a new copy of Quickstep instead of returning you to the suspended copy! In addition, such an attempt may fail due to lack of sufficient memory.*

"Quit" – This selection allows you to leave Quickstep and return to DOS. If you have made any changes to a DSP program since the last time you saved the program to disk, Quickstep will first ask if you wish to save the program.

F3: The PARAMETERS Table

When starting a new program, you must first tell Quickstep what controller you intend to use, and set certain parameters relating to the use of dedicated inputs, I/O counts, etc. The "PARAMETERS" table is where this information is stored.

The topmost choice on the PARAMETERS menu, "Node number", allows you to use Quickstep with a ring network of controllers (refer to your controller's Installation Guide for more information about configuring a ring network). Normally, this parameter should be set to 0 (zero). If, however, you set the Node number to a value other than zero, Quickstep will format all of its external communications (i.e.; downloading and controller monitoring) according to CTC's ring network protocol, and these communications will reach the appropriate controller in the ring network.

The next choice on the PARAMETERS menu allows you to specify the controller model number you will be programming. This choice must be made properly to insure that your program will be compiled and downloaded correctly, and that the Quickstep monitoring function will work as intended. Various models of CTC controllers use different formats for expressing their programs internally, and these must be accomodated by the compiler.

More information regarding the controller models supported by Quickstep, as well as the setting of other parameters within the PARAMETERS menu, may be found in chapter 4; "Starting a New Program".

F4: The PHILOSOPHY Key

This key performs a function which some may view as frivolous, and which management will certainly view as dangerous (people have been known to spend hours staring into a computer screen, compulsively pressing the F4 key without regard to time or circumstances). The purpose of the PHILOSOPHY key is quite straightforward; it will provide, within a window inset into the computer screen, a quotation carefully selected (actually randomly selected) for appropriateness by Quickstep.

Pressing either the F4 key again, or the ESCape key, will make this window disappear.

F5: The DEBUG Key

This key allows you to quickly locate certain common structural errors within your program. It may only be used after your program has been compiled. During compilation, Quickstep will create a file containing the location and type of any errors which it located; this file is used by the DEBUG key to locate the errors.

Upon pressing the F5 key, Quickstep will move to the step number in which the first error in the error file is located. The instruction containing the error will be highlighted, and a specific error message will be shown at the bottom of the screen. You may then edit the program to resolve the error.

The F5 key may then be pressed again to locate the second error, and etc. When the last error has been identified by the DEBUG function, pressing the F5 key will result in a message at the bottom of the screen indicating that no further errors are present.

For more information about editing programs, refer to chapter 6; "Editing".

F7: The MONITOR Menu

This key provides a menu of choices for monitoring the operation of a controller attached to your computer via an RS-232 link. A broad array of diagnostic functions are available, including Program Status monitoring, Register and Flag monitoring and modification, and I/O monitoring and modification.

Chapter 9, entitled "Using Quickstep Diagnostics", provides full information about the use of these provisions.

F8: The D-TABLE Key

The F8 key allows you to establish, program and edit a Data Table, to be downloaded along with your program into the controller. The Data Table is a two-dimensional array of numbers, which may be used for a variety of purposes.

The programming and editing of Data Table information is described in great detail in Chapter 7; "Entering a Data Table".

F9: The SEARCH Function

The SEACH key allows you to search through the body of your program for any text string (for example, a reference to a particular register, or a "cancel other tasks" instruction, etc.). Upon pressing the F9 key, you will first be prompted to enter a direction specifier; Quickstep will search either forward or backward from the current location of the cursor (video highlight). The direction to be searched is specified by entering a forward-slash for forward searches, or a back-slash for backward searches.

You then enter the text string for which you wish to search (for example, reg#10), followed by the "Enter" key, and Quickstep will move to the first occurance of that text string in the direction you specified. You may then review and edit the instruction which was located, if desired. Pressing the F9 key again will display your previous search string as a default; to search for the next occurance of that string, simply press the "Enter" key.

F10: The STEP# Key

This key provides a convenient means of moving from one step to another within Quickstep. If, for example, you wish to move from your present location viewing step #1, to step #53, press the F10 function key. Quickstep will prompt you for a step number, at which point you may enter "53", followed by the "Enter" key. Quickstep will then proceed immediately to step #53 and display that step on the screen.

The F10 key may also be used to search for a step bearing a specific label. For example, if you wish to move to a step which had previously been labelled "Fault_monitoring", you may press the F10 key, and enter "Fault monitoring" as the step label to be located, followed by the "Enter" key. Quickstep will search for a step with this label and, if found, will display that step on the screen.

Note that there are a number of other keys which may be used to move from step to step; these include the Page-up/Page-down keys (to move one step at a time), the Ctrl-Page-up/Ctrl-Page-down key combinations (to skip over empty steps), and the Home and End keys (to move to the first and last programmed steps, respectively). You may decide which is the best method to move to the desired step in any given circumstance.

Index