This report centers on a four-month internship I performed in the documentation department of CTC Parker Automation from February to May 1999. My major responsibility was revising previous documentation and writing new material for the company’s computer hardware and software products. My largest assignment was writing the online help for a new industrial computer interface, the Shell.

The first chapter of this report describes CTC Parker Automation and its documentation department. The second chapter highlights my major internship activities. The third chapter discusses my work writing online help for the Shell computer interface. In the final chapter, I analyze my internship challenges and the lessons I learned facing those challenges. I also discuss how I adapted the Problem-Solving Method for Technical and Scientific Communication to my particular situation.
A TECHNICAL WRITING INTERNSHIP
WITH CTC PARKER AUTOMATION

An Internship

Submitted to the
Faculty of Miami University
in partial fulfillment of
the requirements for the degree of
Master of Technical and Scientific Communication
Department of English
by
Todd Allen DeLuca
Miami University
Oxford, Ohio
2001

Advisor ______________________________
Dr. Paul V. Anderson

Reader ______________________________
Dr. Jennie P. Dautermann

Reader ______________________________
Dr. Katharine J. Ronald
# Table of Contents

**Chapter 1: Introduction** .......................................................................................... 1
  - My Background ........................................................ .............................................. 1
  - About CTC Parker Automation ........................................................ ..................... 1

**Chapter 2: Overview of Project Work** .................................................................... 4
  - Orientation and Training ........................................................ ................................. 4
  - Installation Documentation ........................................................ ......................... 5
  - Tutorials ........................................................ ................................ ..................... 7
  - Other Projects ........................................................ ................................ ............. 9

**Chapter 3: Major Internship Project** ................................................................. 10
  - Shell Description ........................................................ ................................ .......11
  - Project Introduction ........................................................ .............................. ..12
  - Information Gathering ........................................................ ............................... 13
  - Preparing the First Draft ........................................................ ............................. 14
  - First Review ........................................................ ................................ .............. 16
  - Second Draft ........................................................ ................................ ............ 17
  - Second Draft Review and Rework ........................................................ ............. 17
  - Third Draft ........................................................ ................................ ................ 19

**Chapter 4: Internship Analysis and Reflection** ............................................... 21
  - Major Challenges ........................................................ ................................ .......21
    - Learning the Tools ........................................................ ................................ .... 21
    - Learning the Conventions ........................................................ ................. 22
    - Gathering Information ........................................................ ....................... 23
    - Getting Feedback ........................................................ ................................... 24
    - Managing Multiple Projects Simultaneousl ............................................. 25
  - Adapting the Problem-Solving Met ........................................................ ....... 26
  - Summary ........................................................ ................................ ................... 28
List of Figures

Figure 3-1: Sample Shell Interface Screen ..................................................... 10
Figure 3-2: Shell Interface Description ........................................................... 11
Figure 3-3: Shell Help Sample Screen ............................................................ 12
Figure 4-1: Problem-Solving Method for Technical and Scientific Communication ........................................................... 26
Figure 4-2: Internship Documentation Steps .................................................... 27

Appendices

Appendix A: P1 PC/104 Cover Installation Document ...................................... 30
Appendix B: Pentium Third Serial Port Installation Document .......................... 37
Appendix C: Excerpt from MachineLogic Quick Start Guide ............................. 52
Appendix D: Excerpt from MachineLogic Tutorial Manual .............................. 73
Appendix E: MachineShop Shell User Interface Specification Document ......... 94
Appendix F: MachineShop Shell Help ............................................................ 120
Appendix G: Sample of Supervisor Markup from Shell Help First Draft .......... 137
Appendix H: Shell Help Description Document .............................................. 149
Appendix I: Shell Help Text ID Document ..................................................... 151
Appendix J: Project Tracker Document .......................................................... 153
Appendix K: Finished Projects Document ....................................................... 155
Chapter 1: Introduction

In this report, I discuss a four-month internship I performed in the Technica Documentation Department of CTC Parker Automation (CTC Parker) from February to May 1999. My work as an intern involved writing documentation for the company’s various computer hardware and software products used in industrial applications, such as in an automated factory.

My Background

The internship position at CTC Parker was a good match with my previous work experience and course of study at Miami University. Prior to my internship at CTC Parker, I was a full-time graduate student in Miami University’s Master’s of Technica and Scientific Communication (MTSC) program, where I held a teaching assistantship and taught ENG 313 (Introduction to Technical Writing). My course of study in the MTSC program was computer focused, with elective classes in the areas of desktop publishing and computer programming. This study was an extension of my work experience before I enrolled in the MTSC program in the fall of 1997; after graduating from Miami University in 1992 with a B.S. in Economics, I had worked approximately four years in the publishing business as a graphic artist, layout designer, pre-press operator, and a computer hardware and software specialist.

About CTC Parker Automation

Computer Technology Corporation (CTC) was founded in 1980 in Cincinnati, Ohio, to serve the factory automation and machine interface markets. In 1998, CTC was purchased by the Parker Hannifin Group of Cleveland, Ohio, which changed the company name to CTC Parker Automation. Parker Hannifin is a global supplier of motion and control components for hundreds of industrial and aerospace markets, with over $4 billion in sales and more than 35,000 employees worldwide. As a division of the Parker Hannifin Group, CTC Parker manufactures and markets industrial machine
control computers and man-to-machine interface (MMI) software to an international market.

CTC Parker’s major hardware product is a line of industrial computer workstations called PowerStations. The PowerStation is a heavy-duty, touchscreen-based computer designed for installation in factories and other industrial facilities. CTC Parker also engineers and produces software products meant to run on the PowerStation computers; these products, designed for computers running DOS or Windows operating systems, include Interact, a modular MMI package, and MachineLogic, a machine control and management package. Besides designing and building hardware and software products, the company also offers training, consulting, and customer support services. Clients such as Abbott Laboratories, AK Steel, Ford Motor Co., and Exxon Chemical Co. use CTC Parker products in a variety of applications.

I performed my internship at the CTC Parker Automation world headquarters in Milford, Ohio, just outside of Cincinnati. The 30,000-square-foot facility is situated in a low-rise suburban office park that houses similar office and light manufacturing and industrial businesses. There are approximately 80 employees in the headquarters’ departments, which include Administration, Sales and Marketing, Manufacturing, Distribution, Engineering (hardware and software), and Technical Documentation. CTC Parker’s Technical Documentation Department is responsible for writing the print and online documentation that supports the software and hardware products that the company manufactures. When I started the internship, there were four other full-time writers in the department, not including my supervisor (who also did some writing).

As a technical documentation intern, my major responsibility was to write assigned documentation for several CTC Parker hardware and software products. David Brower served as my supervisor and manager of the Technical Documentation group. David had been a technical writer for 12 years and had an educational background in electronics training. As department manager, he was responsible for overseeing all documentation projects at CTC Parker, assigning writing projects for the department, building work
schedules, and reporting the status of our group to upper management and the supervisors of other departments.

The next chapter of this report discusses my introduction to the Technical Documentation Department at CTC Parker and the types of projects I worked on during my internship. In the third section, I talk about the primary project I worked on during my internship, the contents of an online help system for a new computer interface (Shell). Finally, in the last chapter I reflect on my internship and lessons I learned from the experience.
Chapter 2: Overview of Project Work

I performed many activities and worked on multiple projects during my internship. For this report, I combine my work into four general categories: orientation and training, installation documentation, tutorials, and other projects.

Orientation and Training

As a new writer at CTC Parker, I needed to become familiar with the company’s policies, products, and documentation tools, as well as with the engineers and writers I would be working with on a daily basis. It was not until I gained experience and became thoroughly familiar with the company’s internal processes and structure that I became an effective and productive writer. Fortunately, the Technical Documentation Department was well regarded in the company and had a good working relationship with the other departments. This relationship was important, especially given the volume of material I would write and the variety of people with whom I would work.

When I first began my internship, I was given a tour of the office headquarters, including the manufacturing facility. I also was introduced to most of the people who worked in the various departments, including the four other writers in the documentation group. The most experienced writer had worked for the company for five years. The least experienced was a person who recently graduated with a technical writing degree but had not held a job in this field previously. One writer was a French national who translated the software manuals to French and coordinated German translation with another company that localized software. I was immediately impressed with the casual and friendly company environment and people; this made it easier for me to adjust and adapt.

Before writing or working on any documentation, I was given time to train myself CTC Parker products and writing tools (software). As part of the training, I was provided with a temporary computer workstation running the Microsoft Windows operating system. Although I had used Windows computers before, at the time I was more familiar
with Apple Macintosh computers. Therefore, I was initially slightly less productive and took more time to accomplish tasks as I learned the new system. The training computer was also loaded with Interact, one of the company’s major software products used to create graphical representations of hardware controls and display them on the company’s touchscreen computers. For example, I could create a button or lever on the screen that simulated the control of a factory machine. To learn the software, I was given a user manual to read and instructed to run the self-paced tutorials that came with the program. Using the tutorials as a guide, I created two sample projects, a car dashboard and an automated deep fryer. My supervisor, David, reviewed my progress on the projects and assigned other tasks when he felt I knew enough of the product to move on.

During the second week of the internship, I was introduced to Adobe FrameMaker, the application used by the department to write all its documentation. As with my training on Interact, I was given a user manual and told to run through examples and self-paced tutorials to learn the software. Initially scheduled for almost a week, training on the application was cut to about three days, which meant that I still had much to learn about how to use the software while working on actual projects.

Although the total time for training originally had been scheduled for more than two weeks, my entire training and orientation was shortened due to pressing deadlines and a backlog of documentation projects. By the end of the second week of my internship, I was already assigned my first projects, which I group into two major categories: installation documents and tutorials. I also worked on other projects (such as product release notes) that do not fit neatly into these specific categories; these projects generally involved the creation of smaller documents and represented a fraction of my internship work.

**Installation Documentation**

One of the major types of writing I produced during my internship involved documenting the installation of computer hardware. Unlike documenting software, which was often incomplete and behind schedule, documenting hardware was more straightforward. Once
the equipment was ready, it became a matter of reviewing the installation with a user or engineer and writing down the steps. In the process, I had to ensure that I followed standard hardware documentation protocols, which included using disclaimer and protection notices for user safety and company liability purposes. I also had to use a procedure-based writing style with an appropriate level of detail, including supporting material such as figures and labels that the user would find helpful.

Appendixes A and B present two examples of documents I worked on that illustrate the basic content and processes for hardware documentation at CTC Parker. The first is an installation document for metal casing covers used to protect expansion cards installed on a specific touchscreen computer, the P1 PC/104 Cover Installation. This document was relatively simple and completely hardware-based; the only initial requirement before I wrote the first draft was a brief introduction and demonstration by one of the Engineering Department supervisors. Since this particular piece of hardware was being supplied from an outside vendor, I also used the included part instructions and specifications for my document. To begin writing the project, I copied a current hardware-based installation document to use as a template for general style and structure and replaced the content. For example, I replaced previous installation steps with the steps I had written during a demonstration provided by an engineer and added illustrations I had scanned from the vendor’s documentation. During a series of three reviews, I incorporated some changes (mostly minor wording and graphics changes) until the document was ready for a quality review. After reviewing the draft with department engineers and self-testing the instructions on a sample system, I submitted the final copy to the Quality Assurance (QA) Department for review and final approval. The QA Department was a critical link in the documentation chain; these employees performed user testing and gave final approval before the document could go to production. Once the document was ready for production, it was handed off for printing and archiving. I also submitted a copy of the final document to my supervisor so he could track the status of the project and make sure that the document fulfilled departmental requirements. With the combined reviews and a delay in receiving the final parts to test, the entire process of completing the P1 PC/104 Cover Installation document took approximately seven weeks.
The second example is a combined hardware and software installation document that explains how to install and configure a CPU (Central Processing Unit) add-on card for the company’s larger touchscreen computers (see Appendix B). For this project, I worked with and interviewed different engineers and performed testing on various computer platforms. As with most projects, I used an earlier document as a template to save time and ensure a higher level of consistency among projects. The overall process of information gathering and draft reviews was very similar to one I used when preparing the cover installation document. I interviewed engineers, examined the hardware parts, and then updated the procedure steps and graphics that were in the previous document. However, because software was included as part of the project, I also went through the process and wrote down the steps required to install the software (which included taking screen captures for the documentation). After I made changes to the drafts that I submitted to the project engineers, the QA Department, and my supervisor, the documents were then ready for production and archiving. This particular project took approximately three weeks to complete.

**Tutorials**

In addition to installation documents, I also worked on tutorial-based material for some of the company’s software products. In particular, I worked on two related assignments for the same product, MachineLogic. The first project was an update to the Quick Star Guide, and the second was an update to a tutorial manual used by instructors for classroom training. As with work on other types of documents for the department, these tutorial-based projects generally involved three or four review and revision cycles until the final document was ready for production.

One of my first internship projects was to update a Quick Start Guide manual for the MachineLogic software product (see Appendix C). Most major software releases use a Quick Start Guide, which introduces customers to the highlights and features of the product and shows them, through examples, how to use the major functions of the product. For this assignment, I updated an earlier version of the document by using the
comments of a draft that had been previously tested by one of the software engineers. After a subject matter expert (SME) review of the document’s most significant changes, I incorporated additional smaller fixes and updates until the document was ready for review by the QA Department and my supervisor, final changes, and printing and archiving. This project was also an early opportunity to become more familiar with FrameMaker, the publishing software that the Technical Documentation used.

I also worked on an update for the MachineLogic Tutorial Manual (see Appendix D). This project was more substantial than the Quick Start Guide I had worked on. First, it was geared for classroom instruction and was being coordinated with the Marketing Department. Second, it had been produced using tools (Microsoft Word) and a document style that were not standards in the Technical Documentation Department. Finally, this document served two distinct audiences, end users with very little product experience and technical instructors with varying degrees of teaching experience. These differences influenced both the process and final delivery of the project, along with a short deadline due to a pending in-house training class that needed the tutorials.

After an initial review of the project, I proposed making high-level editorial changes and converting each chapter into an Adobe Portable Document Format (PDF) document. I thought that this proposal would satisfy the Marketing Department, which wanted minor fixes and smaller file sizes, and also my supervisor, who wanted a substantially improved document that could be distributed and navigated electronically. These changes would also appease each group’s interest in a quick turnaround for the project. After receiving approval for the proposal, I began work on a prototype that automated the conversion of the original document file into a PDF file with electronic links inserted automatically; I felt that this approach achieved my objectives with the least setup work because I was already familiar with the tools and process. After reviewing the prototype with my department supervisor, I proceeded to finish the project in a similar style and submitted it to the Marketing Department.

Within a week of submitting the MachineLogic Tutorial Manual for review, I received a copy of the draft with significant changes, including major revisions to the text and
application screen shots. Initially, I had focused on making consistency changes and fixing errors that I noticed while reviewing the document. Unfortunately, I had been unable to test the tutorials thoroughly or to check the content for accuracy and correctness. When the Marketing Department reviewed the manual in greater detail, they realized that much of the content needed to be updated to the most current version of the software. After reviewing the marked changes and updates with my supervisor and a quality assurance engineer, we decided to prioritize the changes and bring another writer onto the project to finish it as quickly as possible. With new priorities and the help of the other writer, I was able to make the most important changes and finish the project within a week, which included converting the tutorial chapters into individual PDF files and then posting them to the web for internal use.

Other Projects

I also worked on other documentation projects during my internship. Some were minor, unscheduled revisions that took a day or two to complete; others were projects that I started but was unable to finish due to time constraints. In the last weeks of my internship, the projects that I could not finish were handed to other writers in the department. These projects included an Interact Getting Started Guide, a MachineShop Runtime User Guide, and a MachineShop Getting Started Guide.

In the next section of this report, I discuss the largest project that I worked on throughout my internship at CTC Parker: the contents of an online help system for touchscreen computers (Shell).
In the second week of my internship, I was assigned the largest project of my internship: creating user documentation for the Shell. Formally known as the MachineShop Shell, this menu-driven interface and its help system were being developed for inclusion in all future versions of the company’s industrial touchscreen computers (PowerStations). The underlying architecture of the software was designed to run the computer; the user interface that ran on top of the software was designed so that end users could easily configure their computers and access the system help. See the following figure (Figure 3-1) for a basic layout and sample screen of the Shell computer interface.

My initial goal was to write the online help contents for the system. A few weeks later, I was assigned the task of expanding the online help and creating printed versions of the Shell help that would include more detailed information and instructions. However, I did not start the project when it was assigned because the interface software was still being developed by both hardware and software engineers, who were trying to get the system to work as planned. Most of the Shell interface features and functionality were not ye
working, and the computer system that was being used for testing and prototyping was not available for me to review the product. It was almost three weeks until I began any documentation work or research on this project.

**Shell Description**

The new MachineShop Shell was a significant software product and a major component of the company’s industrial touchscreen computers. The original name for the new product had been the PowerStation Shell, but it was changed before public release to match an existing product CTC Parker was already supporting. At the time of my internship, the Shell software was undergoing a complete renovation and upgrade from the previous user interface, making it essentially a new product. The following description (Figure 3-2), taken from the actual help system, describes the software well.

**Shell Interface Description**

The MachineShop Shell is used by runtime systems to receive Interact and MachineLogic files and MachineShop projects. It is a graphical, menu-driven interface that communicates with the PowerStation and its installed components (hardware and software). The Shell displays information about the runtime system and allows you to configure many hardware and software settings.

**Figure 3-2**

A user can navigate through the menu-driven interface by either tapping directly on the touchscreen or by using a combination of keyboard and mouse commands; most users in an industrial environment use the touchscreen. While the user is navigating through the various system menus, he or she can access the online help by tapping a HELP button located at the top left portion of the screen. The monitor then displays a help dialog box that succinctly describes the features and functionality of the buttons for a particular screen. Writing the contents of these help screens and dialog boxes was my main
responsibility for the project. See the following figure (Figure 3-3) for an example help screen.

**Shell Help Sample Screen**

![Shell Help Sample Screen](image)

**Figure 3-3**

**Project Introduction**

When I was first assigned the project, I discussed the basic features and design of the Shell user interface with my department supervisor. We also went over the goals, parameters, and deliverables of the documentation. Since the interface was still in development, my supervisor could not supply any additional information about its actual contents or functionality. However, after some initial research, I was able to find a project specification document that described and outlined the project in detail. After reviewing this information and taking available resources into account, I estimated that I could complete the online help in approximately three weeks.

A specification document is required for all major software products at CTC Parker, including the MachineShop Shell interface. The document is written before any engineers start coding and creating the actual software. It includes detailed information about how the software will look and how it will work from both a technical and end user perspective.
perspective. Basically, it is used as a blueprint for the project engineers and supervisors. Depending on the available project time and deadline schedule, these specification documents vary in both size and quality. Fortunately, the specification document for the Shell interface was fairly complete and contained most of the information I would need to start writing for the project (see Appendix E). Along with initial interviews with project engineers, I used this document extensively to develop my first draft for the project.

**Information Gathering**

To gather information for my first draft, I interviewed the project’s software engineers and used the specification document to outline the general structure and get an idea of the amount of information required to properly document and provide help for the Shell interface. From speaking with the software engineers, I found that the specification document had not been updated in months; it did not reflect the current state of the software and what the final product would look like. A significant portion of the Shell interface described in the specification document had either changed or had not been implemented on the prototype system. This made it difficult to write complete, up-to-date documentation and also restricted my ability to test the documentation before handing it off for initial review.

I also learned that the current prototype was significantly behind schedule and undergoing additional changes. Consequently, I needed to gather new information continuously, even as I was creating my first draft. When I compared the contents of my first draft to the most recent builds, I found interface features that had never been documented and other features that were partially implemented but not working yet. For example, the networking protocol that had been proposed for the system was not working; the engineers were discussing using a completely new configuration. This meant that the help contents I was writing might only be temporary and need to be changed towards the end of the project. These types of changes had a significant impact on both the quality of the documentation content and the information gathering for the different system components.
In order to get the most recent information I needed for the online help, I interviewed project engineers almost daily. Sometimes, while working on the prototype system, I would even accidentally stumble across a new feature that the engineers had neglected to tell me about. This process was inefficient and time-consuming, especially compared to most of the information gathering and draft writing I had done for my previous projects. It also limited the amount and quality of the documentation in my first draft, which I needed to present for review because of the tight deadline. As I would learn during the course of my internship, these kinds of setbacks are common in software development and not atypical scenarios when writing software documentation.

Preparing the First Draft

Over the first three weeks of the project, I devoted a week-and-a-half to writing the first general content, which I felt was rough but enough to discover, after a review, if I was on the right track for style and content. I intended to get the information I had into a structured framework first and then revise the actual contents and make corrections or fill in any information gaps. After I had gathered this initial information, I met with the project engineers to review the actual formatting of the contents for the Shell interface help. This meeting was important because there were a number of limitations and restrictions placed on the contents and design of the Shell interface, including its help.

There were very specific restrictions and styles for the help text that I had to follow so that the information would display properly within the touchscreen windows. For example, the entire contents of the Shell interface (which included help) had to fit within a 320 x 240 pixel window to accommodate the smallest computer screen that the company produced. This meant that the help text could not exceed a certain number of characters per line before wrapping to a second line. Although there was a mechanism for scrolling if the help contents were longer than a single screen page, this practice was discouraged. It was important to limit the amount of unnecessary text so that users could get the most important information as quickly as possible. In addition, the text was also being translated to other languages such as German and French, which typically require
thirty percent more text space than English. Any additional information or lengthier explanations were saved for the written manual that would also accompany the system.

In response to these space limitations, I used a short style of writing that eliminated unnecessary words or phrases. I wrote exactly what a system feature or button did, but not how it worked or how to use it. For example, to describe the function of the CALIBRATE TOUCHSCRN menu button, I wrote “Launches the touchscreen utility.” If I were writing for the written manual, I might have said, “Press the CALIBRATE TOUCHSCRN button to launch the touchscreen utility.” In fact, most of my first draft did not contain full sentences. Taking into the account screen limitations and a more technical audience with prior computer experience, the style seemed appropriate.

Not only did the text have to fit within a smaller screen space, but it also had to have special “tags” and “links” to work properly within the Shell interface. The final document also needed to be saved as a pure text file (.txt format) with very limited formatting because the new interface used a text-driven XML (Extensible Markup Language) formatting scheme. Similar to HTML tagging, the Shell interface would read the contents of the text document I created and format it based on the tagging used (see Appendix F). If I wanted to bold a word, such as a user interface term or button, I would need to surround the word with tags that had already been specified to bold the text. For example, to bold the word “Yes,” I would have to type \bYes\b. If any of the styles or tags were incorrect or if the help text file was saved incorrectly, the Shell system would not import the file and the help would not function properly. Therefore, I had to pay extra attention when creating the final document and adding the tag information after I wrote the actual help contents. These requirements did not pose any significant hurdles, besides some extra time, because the system was relatively simple and I was already familiar with writing HTML code for web pages.

I performed a thorough check of the draft’s content and formatting for any technical errors that might cause the contents to display improperly. I also asked an experienced writer in the department to review a portion of the help for writing style and language usage. After the review and document formatting were complete, I submitted the firs
draft to my department supervisor for an initial review, even though it had not been
requested. I wanted to show the progress I was making on the project and to ensure that I
was presenting appropriate writing style and substance. I followed this step with most of
my internship projects, especially the longer or more complicated ones.

First Review

Within a week of submitting the draft of the Shell project to my supervisor, I received a
marked and reviewed copy. Unlike responding with the general comments and
suggestions I was expecting for a first draft, he had either crossed out or rewritten a
significant portion of the text. While I had considered this document a rough draft and
was looking for general feedback, my supervisor was expecting a different style of
writing and a more complete and polished document. His corrections and written
comments on the draft reflected these different expectations (see Appendix G).

My supervisor made three types of markups on the first draft. The first type was a cross-
out where a paragraph or sentence should be deleted. Some of the cross-outs were pieces
of text that were not necessary; others were system functionality that had not yet been
implemented, but I had used placeholder text to mark the spot for future revision. The
second type of correction was rewording, where text would be crossed out and the same
information would be rewritten using different words, a different voice, or another tense.
For example, for a menu button description I had written, “Use this selection to configure
Ethernet port settings on the PowerStation.” His proposed rewording for the sentence
was, “Select this button to enable configuration of the Ethernet port.” Some of these
suggestions were improvements or clarification of the text, and others reflected personal
preferences concerning writing styles. These particular types of changes were difficult to
avoid, especially for a first draft of a new document type that had no style sheet or
parameters. The third type of markup on the draft was a general, non-specific, and simply
written “rewrite.” This was the most difficult type of correction to analyze because there
were no other comments or suggestions about what type of improvement needed to be
made.
Because the draft contained such a significant amount of markup, my supervisor and I scheduled a meeting to discuss the project and review his comments. In addition to going over the corrections he marked on the draft, my supervisor suggested that I continue talking to the project developers to get the extra information I needed to fill some of the gaps in my draft. I also decided that I should patch and rework what I could from the first draft before submitting it to project engineers for review of the second draft.

**Second Draft**

For the second draft, the most important change I made was in writing style. In the first draft, I used a less formal writing style with short sentences and more active, procedural language because of the limited space of the computer interface and the action-oriented usage of the system. On the other hand, my supervisor felt that the project should include more explanation and detail. Unfortunately, we had not talked about this aspect in our initial discussion of the project.

To gather the additional background and explanation I needed, I spoke to both the software engineer and his supervisor. Although I had originally used the project specification document to create my first draft, I discovered after talking to the software engineer that the project was still weeks away from being complete and that many of the components and descriptions used in the specification document had changed or were not being implemented. Needing to move forward with the project, I gathered as much information as I could and examined the most recent Shell prototype. This research helped me to have a second draft ready for my supervisor by the end of the next week.

While working on the second draft, I informed my supervisor of the incomplete development status of the Shell project so he better understood the reason why some of the contents of my documentation were not yet finished.

**Second Draft Review and Rework**

Soon after receiving the initial feedback on my first draft, I made major revisions to the document and submitted the second draft to my supervisor for review. I was confident
that this draft was as complete as possible (given the current pre-production status of the system) and that it addressed most of the questions my supervisor posed on the previous draft. To appease his concerns about the status of the project, he asked the software engineer’s supervisor to also review the document. My supervisor and I hoped that this additional review would satisfy any expectations the software supervisor might have concerning documentation for the project. Even though this project initially had been scheduled for about three weeks of writing, at this point I had been working almost five weeks.

A week after I submitted the second draft, my supervisor returned his comments, which surprised me. He and the software engineer’s supervisor decided that the documentation needed more work and to head in another direction. The revisions for the next draft would need to more closely match the style and contents I had written for the first draft. My supervisor admitted to misunderstanding the scope and status of the project and took responsibility for leading me in the wrong direction on the second draft. Fortunately, I had kept copies of my first draft and early revisions of my work, which made it easier to retrieve the most useful portions of what I had previously completed.

Although there were now two different versions of the same content (represented by each previous draft), my supervisor’s proposed solution was to use the first draft as the basis for the Shell’s online screen help. The content would be reorganized and formatted to fit within the limited screen space of the online system. It would also be tightened using a short sentence style with fewer explanations and less detail. In addition, the second draft of the project would serve as the basis for the printed manual that I had been assigned but had not yet begun to work on. This Shell help manual would include the additional information, definitions, and examples I had implemented in the second draft of the project.

Before I started working on the second round of revisions, I created a description sheet for myself and the project supervisors that outlined the overall goals and content of the Shell help project (see Appendix H). This project description document gave a clear direction for the project (based on information up to that point), reduced the likelihood of
future misunderstandings, and ensured that all those involved were thinking alike. I passed this sheet to the developers and department supervisors for review, and the agreed with the contents and parameters of the document.

**Third Draft**

Two weeks after reviewing the second draft, I began using the project description document as a guide for the second revision of the Shell help document. The process for completing the third draft, including some additional writing and research, took approximately seven weeks. However, most of the work during those weeks was reorganizing and formatting my previous work and filling in missing pieces of information from the software developers; I did not write much new content for the system.

Before completing the third draft and handing it off to my supervisor, I concentrated on organizing the content and structure of the Shell help system. This was important because the online version of the project contained many small pieces of information and very specific formatting that needed additional documentation for future updates, as well as a knowledge transfer to other writers who would work on the project after I had left the company. I also reviewed the latest structure and formatting of the document with the software developers so that all information was up-to-date.

Having this part of the help document correct meant that I could load the document and review its contents on the prototype system to see how well everything worked. Not only did this review act as a quality assurance or testing step, but it also brought me up-to-date with the most current features and functionality of the system. In addition, I created a Text ID spreadsheet that had all of the menu picks and their corresponding Text IDs for reference and double-checking against the tag IDs I had written in my document (see Appendix I). If the IDs in my document did not match the IDs that the prototype system was expecting, the help contents would not work correctly
When I submitted the third draft for review, I was fifteen weeks into my internship, with two weeks remaining to make changes and complete the project as much as possible before leaving CTC Parker. Fortunately, after the third review of the document, my supervisor felt this version of the project was “in good shape”; there were no major changes and only minor corrections to the document. However, due to the delayed deployment of the software and other projects I had been working on while I was writing the Shell online help, the written version of the Shell help was given a lower priority. As a result, the Shell help manual remained in an early draft stage.

During the next two weeks, I added whatever additional information to the Shell online help that I could. Most of the information included late developments in the software and prototype system, which consisted mostly of new menu screens and options that had not been in the specification or current documentation. I also fixed any broken hardware references in the document, updated the project reference sheet, and compiled the help on the most recent prototype system. Before I left the company two weeks later, I thoroughly reviewed the Shell project in its current state and the corresponding files and documentation with my supervisor so he was familiar with my work and could reassign it to another writer.
Chapter 4: Internship Analysis and Reflection

In this section of the report, I reflect on the major challenges of my internship, the things I learned from those challenges, and the ways my internship relates to the problem solving method for technical and scientific communication.

Major Challenges

The major challenges I faced while performing my internship at CTC Parker concerned the following activities: learning the tools, learning the conventions, gathering information, getting feedback, and managing multiple projects simultaneously.

Learning the Tools

Learning the tools and equipment used by the Technical Documentation Department at CTC Parker was an early challenge. Originally scheduled for two weeks, my initial training was designed to help me become familiar with the company’s products and writing tools as quickly as possible. There were no structured training programs or classes for new employees at CTC Parker, so I worked through computer-based tutorials and company training materials on my own. Although this did not give me the contextual or application-specific information I was looking for, it was at least a chance to see the programs and tools I would use before I began writing. Unfortunately, due to a large backlog of projects, my self-taught training program was cut to less than two weeks.

The biggest impact of the reduced training was lower productivity for my early projects; it took me almost twice as long to complete an assignment as an experienced writer who was more familiar with the tools. To overcome this lack of experience, I put in extra time learning the software and tools while writing my first projects during the initial weeks. I ran into obstacles or had specific problems with the company’s software or writing tools, I would approach other writers in the department. More often than not, they were able to point me in the right direction or help solve a specific problem.
I would have preferred more time to prepare before I began writing my first assignments, but I learned that training and preparation are often cut short when there are many projects that need attention. In smaller companies, which generally do not have the staff or resources for lengthy or thorough training, it is more important to get work done than to spend additional time preparing new employees. I also learned that training comes in different packages, whether it is a structured classroom with an instructor or a style sheet or book that necessitates self-instruction.

Learning the Conventions

Another early challenge of my internship was learning the styles and conventions used by members of my department. Before I started my first assignments, I made a conscious effort to make my writing fit as seamlessly as possible into what had been previously written by the other writers. Although a few department standards and procedures had been written down (such as a guide for using the department’s FrameMaker templates), most were not. This made assimilation difficult for me because I was not yet familiar with department writing styles.

Because these materials were not available, I turned to my supervisor and the other department writers for help and guidance. Unfortunately, many times my supervisor was too busy or unable to help if I needed assistance; he was a hands-off administrator who generally was not involved in the daily work of the writers. If I could not get help from my supervisor, I approached the other writers. They were very helpful, providing advice on how to improve my writing and suggesting how I might handle issues or questions I had about fitting into the department or approaching my supervisor. For writing style and word usage, I used the Microsoft Manual of Style for Technical Publications, a popular book for technical writers in the computer software industry. Before approaching the other writers, I often turned to this book, which my supervisor recommended, for specific sentence-level writing assistance.

It takes time to fit into a new company or department; it took me about a month at CTC Parker to learn enough to become comfortable working independently on projects.
However, I still turned to others in the department and external means for assistance. I learned that there are many resources available to help a new writer become an effective member of a team. These resources include asking whether the company or department has a style guide or uses templates to produce documents. When I could not get help from my supervisor or other employees, I would examine existing documents for writing style or turn to external documents or colleagues for support.

Gathering Information

A challenge I faced for almost every writing project was gathering information. As a new writer, I knew very little about the scope of work for my projects when they were first assigned. When my supervisor assigned a project, he gave us a brief introduction and left each writer to decide how to complete the assignment. This meant that I sometimes had to gather additional upfront project information before I could even start content research. For some projects, gathering the necessary information was fairly easy and straightforward: I knew whom I needed to talk to, and they were able to give me what needed to write the assignment. Collecting information for other projects was not so simple: Either the information provided was inaccurate or it was not yet available.

When they were assigned to me, some projects were loosely defined and lacked a clear set of objectives. I turned to subject matter experts or project engineers to get a clearer idea of what I needed to research before I could start writing. I also turned to document examples and other writers to learn about projects I was unfamiliar with. For example, if I was writing a product release document, I would locate and read a prior product release document and use that as a guide for basic content and writing style. After becoming familiar with the material, I would then try to interview the writer who had worked on the document and ask for advice or suggestions. If that writer was not available, I would talk to another writer in the department, an experienced product developer, or my supervisor. I used this approach for many of my writing assignments.

Through the process of gathering information, I learned each project is different and comes with a distinct set of variables when it is assigned. Some of my projects were more
clearly defined or further along than others, and they did not need as much information gathering or upfront research. Even though I was eventually able to get the information I needed for all of my assignments, it often took a lot of patience, persistence, and research before I could start writing.

Getting Feedback

A major challenge for my early assignments was getting thorough and helpful feedback. Feedback was important so that I could gauge my performance on assignments and find out how I was doing overall during my internship. My supervisor provided feedback and judged my work. Generally speaking, however, my supervisor was not interested in assignment reviews until the project was almost finished and the documentation was ready for final production. As a result, the feedback I did receive was usually towards the end of an assignment, after most of the writing for the project had been completed. For some assignments, this level of feedback was enough to finish, but for more complicated or lengthier projects, it made completing the assignment to the expectations of my supervisor more difficult. It was a frustrating process when I had to rewrite a draft or rework a project because the feedback I had received was nonspecific, such as with the Shell Help project. When my supervisor was not available for feedback or if I felt I needed someone else’s opinion on my writing, I relied on the other department writers for a quick opinion or suggestion. They were usually very supportive and helped me on many of my projects.

The QA Department was another important source of feedback. Although writers were not encouraged to give documents to the QA Department until they were ready for final review, I felt it was sometimes necessary to get early feedback. This “unsanctioned” source of feedback was especially important if the project I was documenting was very complex and I was not comfortable with the subject material. QA comments and suggestions were very valuable for these types of projects.

Based on my CTC Parker experience, I learned that getting the feedback and suggestions I needed to produce quality documents could be very difficult. When I did not get
acceptable feedback, I did the best I could and hoped the results were satisfactory. Fortunately, my supervisor was usually pleased with the quality of my writing, and I was able to complete most of my assignments at a high quality level without too many drafts or reviews.

Managing Multiple Projects Simultaneously

A challenge that grew during my internship was how to effectively manage an increasing number of projects simultaneously. Keeping track of projects without falling behind or becoming overwhelmed was a constant process for me, especially when unexpected events would occur. For instance, the resignation of two writers in the department early in my internship quickly shortened turnaround times and reduced the number of resources available to review my projects. This reduction affected the quality of revisions and the amount of material I was able to incorporate into documents before the new deadlines.

To address these challenges, I needed to effectively manage and organize my activities. It was important to establish priorities and work on the most pressing documents first and then try to chip away at my other assignments during a slower period. By staying focused, I was able to avoid the negative impacts of unexpected circumstances and tight deadlines.

To help me manage these various project and deadline challenges, I created a Project Tracker spreadsheet, which identified all my writing projects and highlighted the major actions and events for each, including any important issues or constraints (see Appendix J). I updated this tracking sheet weekly and gave a copy to my supervisor so he knew the current status of all my projects and what I had been working on at the time. The document was a very valuable tool and was well received by my department supervisor because it made better able to track the progress of my work. After I completed a project, I copied the tracking sheet information to a Finished Projects document for archiving and reference (see Appendix K).
It was important to stay organized and establish priorities when I was working with multiple assignments and tight deadlines. I learned that having a written resource to refer to and plan my work around can be very helpful. I used the Project Tracker and Finished Projects sheets I created in combination with a weekly activities journal to stay on top of the many projects I was juggling. These tools were very useful documents, and I referenced them extensively while writing this report.

**Adapting the Problem-Solving Method**

Upon reflection of the process I followed to write my internship projects, I realize that the method I used to produce documentation at CTC Parker was different than the Problem-Solving Method for Technical and Scientific Communication we were taught in the MTSC program (see Figure 4-1). In fact, the method I was using wasn’t really a method at all because a method implies a set of steps or procedures that are followed on a routine basis. My experiences and activities were not routine. They differed from project to project. A major reason for this variance was that there were no clearly established guidelines or steps used to write documentation at the company; each writer was assigned a project and then left to his or her own devices on how to complete the assignment. In the process of learning the tools and systems and completing assignments, I used my own customized methodology to accommodate the challenges and constraints I dealt with on my projects.

**Problem-Solving Method for Technical and Scientific Communication**

<table>
<thead>
<tr>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Define the Problem</td>
</tr>
<tr>
<td>II. Design the Solution</td>
</tr>
<tr>
<td>III. Test the Solution</td>
</tr>
<tr>
<td>IV. Implement the Solution</td>
</tr>
<tr>
<td>V. Evaluate the Solution</td>
</tr>
</tbody>
</table>

Figure 4-1
Each project I worked on had its own set of goals, deliverables, and challenges. Since I had not worked on any CTC Parker documentation projects before joining the company, I first figured out what was needed to accomplish the goals of each assignment. For example, if I was assigned a product release notes document, I would first mark the project in my tracking sheet and then try to find previous release notes to get a feel for the general content and scope of work. For my first projects, I did not follow a defined or clear process; I gathered information, wrote a complete draft, and then tried to get the document finished as quickly as possible to meet the deadlines. Once I had experienced a full document cycle, I then started to follow a loose pattern of documentation steps that I was comfortable with, which I’ve outlined in the following list (see Figure 4-2).

Internship Documentation Steps

1. Receive assignment from department supervisor.
2. Gather information from project participants.
3. Create draft of document using department templates and styles.
4. Submit draft of document for review by project experts.
5. Incorporate draft suggestions and changes to create revised document.
6. Submit revised draft to project experts, QA Department, or supervisor for review and final approval.
7. Submit final document (with incorporated changes) for publication, production, and archiving.

Figure 4-2

While many of these steps are covered in the Problem-Solving Method, there are some distinct differences. The biggest difference is that my “method” does not include one of the major activities of the Problem-Solving Method, Evaluating the Solution (Activity V). Once I turned in and delivered an assignment, it never came back for evaluation or analysis unless an error was found that was important enough to correct. Although it would have been insightful to revisit a document after users had a chance to use it, there simply was not enough time or resources to make changes to documentation until the next major product revision. Another difference between the methods is that my version of Testing the Solution (Activity III) was different than the one outlined in the Problem-Solving Method. Instead of designing any testing or review procedures, I would simpl
hand over a document draft to the most appropriate reviewer at the time, which was a
developer, QA engineer, or my supervisor. Again, time and resource constraints were the
main reasons why testing procedures were not considered. Another important aspect of
Activity III in the Problem-Solving Method that I did not perform was having a draft
reviewed by a sample audience. In fact, none of the documents I worked on during my
internship was actually tested or reviewed by CTC Parker customers before being sen
ted out for final production.

Another major difference between the activities of my internship and the Problem-
Solving Method was that I often went back and forth between steps, especially during a
cycle of numerous document drafts, as with the Shell project. I would even skip
important steps within my own abbreviated “method.” For example, if I was working on
a late document that needed to get into production as soon as I finished it, I sometimes
turned in the document with no more than a “quick glance” internal review. When the
deadline was tight and I had to turn in a document that was not yet ready, I relied on
instincts and used my best judgment to choose the action that I felt would yield the best
results. Other times I would “wing it” and just do whatever was necessary to finish the
most important task at hand. Having to decide which steps to follow and which ones to
skip, especially on an important assignment, was stressful. Fortunately, more often than
not, the results were good enough to fulfill the requirements of the assignment and gained
approval.

**Summary**

At CTC Parker, I had the opportunity to work on a variety of computer-related products,
learn popular writing tools, and apply much of what I had learned in the MTSC program.
There were many challenges, but I managed to work through the obstacles, found
assistance when available, and applied what I had learned to reach suitable solutions. By
breaking larger projects into smaller tasks, keeping organized, and not becoming
distracted by circumstances beyond my control, I was able to adjust and become a much-
improved and effective technical writer. Looking back, I can also see how the disjointed
and abbreviated documentation process that I followed relates to and diverges from the Problem-Solving Method. Finally, and perhaps the most gratifyingly, these internship lessons and experiences have proven very valuable in the subsequent technical writing positions that I have held since I left CTC Parker in June 1999.
Appendix A

P1 PC/104 Cover Installation Document
P1 PC/104 Cover Installation

Introduction

The PC/104 backshell expansion covers are designed to protect P1 PowerStation PC/104 expansion cards (8-bit, +5V, +12V only) and fit securely on the back of the P1 unit. These covers are strong, vented, plastic and metal enclosures with detachable slots on the side panels.

There are three available P1 PowerStation PC/104 expansion kits. Each kit includes an expansion cover with removable side panels, securing screws, all necessary mounting hardware, and a grounding wrist strap. This document provides instructions for installing the single, double, and triple PC/104 expansion covers to the back of the P1 PowerStation.

Note You can install up to three PC/104 expansion cards on the P1 PowerStation, provided that the cards do not exceed the total available power to the expansion cards (+5V @ 0.75A, +12V @ 0.1A). Note that the P1 will only support 8-bit PC/104 expansion cards.

Electrostatic Discharge Precautions

Modern integrated electronic devices, especially CPUs and memory chips, are extremely sensitive to electrostatic discharges (ESD) and fields. Before you install a PC/104 expansion cover, be sure to follow these precautions to protect you and the PowerStation from harm resulting from ESD.

1. To avoid electric shock, always disconnect the power from the PowerStation before you remove the backshell cover or separate the backshell from the front panel. Do not touch any components of the CPU card or other cards while the PowerStation is on.

2. Disconnect power before making any hardware configuration changes. The sudden rush of power as you connect a jumper or install a card may damage sensitive electronic components.

3. Only handle internal components in an ESD safe location using appropriate grounding methods.
   - Always ground yourself to remove any static charge before you touch the CPU card.
Be particularly careful not to touch the chip connectors.

Keep the adapter in its anti-static packaging when it is not installed in the PowerStation, and place it on a static dissipative mat when you are working on it.

Wear a grounding wrist strap for continuous protection.

Installing the PC/104 Cover for One Card

Follow these steps to install the expansion cover for a single PC/104 card:

1. Turn off the PowerStation.
2. Disconnect the power source from the PowerStation.

   **Important** The PC/104 expansion cover kit includes an ESD wrist strap. To prevent static discharge, you must wear this wrist strap, or an equivalent one, connected to a known earth ground.

3. Remove the two fastening screws that secure the original metal PC/104 protective cover.
4. Install the new silver metallic cover included with the expansion cover and fasten with two metric 3x5 (M3x5) screws.

This will expose the PC/104 connector and the two pre-installed standoffs on the metallic cover.
5. Install the two short standoff screws included with the PC/104 cover kit into the two empty screw holes located at the lower two corners of the backshell of the P1 unit.

   Refer to the preceding diagram for standoff locations.

   Note: Be careful not to overtighten the standoffs.

6. Install the PC/104 card securely onto the PC/104 connector.

   The PC/104 card should rest evenly on the four standoffs.

7. Snap off the detachable plastic slots of the PC/104 side panels, as needed, for any PC/104 card connector(s) to show through the side panel(s).

8. Place the new PC/104 expansion cover on the backshell and align over the PC/104 card. The cover will align with the standoffs in only one orientation for securing to the backshell; the solid side of the cover should face the top of the unit.

9. Secure the PC/104 cover to the backshell using the four longer, slotted screws provided with the PC/104 expansion cover kit.

10. Perform any cable connections or further setup required for the PC/104 card.

The P1 PowerStation with the expansion cover is now ready to be mounted back in its enclosure.

   Note: Access to the Compact Flash (CF) card on the PowerStation is restricted once the PC/104 expansion cover is installed on the P1 unit. If you wish to remove or replace the CF card on the PowerStation, the PC/104 cover may need to be removed.

---

Installing the PC/104 Cover for Two Cards

Installation of a PC/104 Expansion Cover for two cards is similar to the installation of the cover for one card, except for a few additional steps.

Follow these steps to install the expansion cover for two cards:

1. Follow Steps 1 through 6 for installing a PC/104 expansion cover for one card. See the preceding set of instructions.

2. Fasten four of the longer standoff screws to the four corner points of the first PC/104 card.
3. Install the second PC/104 card securely onto the first PC/104 card connector.

The second PC/104 card should rest evenly on the longer set of standoffs.

**Note** If you are not installing a MachineLogic Control Adapter as the top card, be sure that no metallic parts of the second PC/104 card are touching the touch relay of the Control Adapter (metallic box on bottom of the card). Otherwise, a short may occur and damage the MachineLogic Control Adapter.

If you are installing a PC/104 MachineLogic Control Adapter with a UCS adapter, it must be installed last (top card) in order for the card to fit properly within the expansion cover.

4. Fasten a second set of the longer standoff screws onto the second PC/104 card.

5. Remove the detachable metal side panels of the PC/104 cover, as needed, for the connectors of the PC/104 cards to show through the side panel.

6. Place the new PC/104 expansion cover on the backshell and align over the PC/104 card. Make sure the non-detachable side of the cover is facing the top of the unit.

7. Secure the PC/104 cover to the backshell using four metric 3x5 (M3x5) screws provided with the PC/104 expansion cover kit.
8. Perform any cable connections or further setup required for the PC/104 card.

The P1 PowerStation with the expansion cover is now ready to be mounted back in its enclosure.

**Note**  Access to the Compact Flash (CF) card on the PowerStation is restricted once the PC/104 expansion cover is installed on the P1 unit. If you wish to remove or replace the CF card on the PowerStation, the PC/104 cover may need to be removed.

---

**Installing the PC/104 Cover for Three Cards**

Installation of a PC/104 Expansion Cover for three cards is similar to the installation of the cover for two cards, except for a few additional steps.

Follow these steps to install the expansion cover for three cards:

1. Follow Steps 1 through 6 for installing a PC/104 expansion cover for one card. See the first set of instructions at the beginning of this document.

2. Follow Steps 2 through 4 for installing a PC/104 expansion cover for two cards. See the preceding set of instructions.

   Refer to the preceding two card installation diagram for a visual representation of the following installation steps.

3. Install the third PC/104 card securely onto the second PC/104 card connector.

   The third PC/104 card should rest evenly on the longer set of standoffs.

   **Note**  If you are not installing a MachineLogic Control Adapter as the top card, be sure that no metallic parts of the second or third PC/104 card are touching the touch relay of the Control Adapter (metallic box on bottom of the card). Otherwise, a short may occur and damage the MachineLogic Control Adapter.

   If you are installing a PC/104 MachineLogic Control Adapter with a UCS adapter, it must be installed last (top card) in order for the card to fit properly within the expansion cover.

4. Fasten a third set of the longer standoff screws onto the third PC/104 card.
P1 PC/104 Cover Installation

5. Remove the detachable metal side panels of the PC/104 cover, as needed, for the connectors of the PC/104 cards to show through the side panel.

6. Place the new PC/104 expansion cover on the backshell and align over the PC/104 card. Make sure the non-detachable side of the cover is facing the top of the unit.

7. Secure the PC/104 cover to the backshell using four metric 3x5 (M3x5) screws provided with the PC/104 expansion cover kit.

8. Perform any cable connections or further setup required for the PC/104 card.

The P1 PowerStation with the expansion cover is now ready to be mounted back in its enclosure.

Note Access to the Compact Flash (CF) card on the PowerStation is restricted once the PC/104 expansion cover is installed on the P1 unit. If you wish to remove or replace the CF card on the PowerStation, the PC/104 cover may need to be removed.
Appendix B

Pentium Third Serial Port Installation Document
Introduction

These instructions explain how to add a third serial port to P3, P4, P5, or P7 PowerStations with a Pentium CPU card.

Warning You must observe normal precautions against static discharge to prevent damage to the PowerStation.

Electrostatic Discharge Precautions

Modern integrated electronic devices, especially CPUs and memory chips, are extremely sensitive to electrostatic discharges (ESD) and fields. Before you disassemble the PowerStation, be sure to follow these simple precautions to protect you and the PowerStation from harm resulting from ESD.

1. To avoid electric shock, always disconnect the power from the PowerStation before you remove the backshell cover or separate the backshell from the front panel. Do not touch any components of the CPU card or other cards while the PowerStation is on.

2. Disconnect power before making any hardware configuration changes. The sudden rush of power as you connect a jumper or install a card may damage sensitive electronic components.

3. Only handle internal components in an ESD-safe location using appropriate grounding methods.
   - Always ground yourself to remove any static charge before you touch the CPU card.
   - Be particularly careful not to touch the chip connectors.
   - Keep any card in its anti-static packaging when it is not installed in the PowerStation and place it on a static dissipative mat when you are working on it.
   - Wear a grounding wrist strap for continuous protection.
Before You Start

Be sure you have the following tools:

- Keyboard (If you have a P3 or P4 PowerStation, you also need a Y cable)
- Flat-head and Phillips screwdrivers
- Needle nose pliers
- Serial port card

Reconfigure the Touchscreen

The following steps describe how to reconfigure the touchscreen to use COM4 instead of COM3 for MS-DOS and Windows.

Configuring the Touchscreen Drivers in MS-DOS

Follow the directions below to configure the touchscreen drivers:

1. Enter MS-DOS mode, and change to the C:\TOUCH\DOX directory.
2. Start the touchscreen calibration program by typing touchcfg. The following touchscreen driver configuration screen will appear.
3. Press I for Interface. The following Interface Configuration Options screen will appear.

```
Interrupt: IRQ11
Address: 3E8h
Connection: UART
calibration data: Controller
Set COM1
Set COM2
Set COM3
Set COM4
Reset: off
Passed:
OK:
```

Enter your selection (I,A,C,R,1,2,3,4,B,D,O): 

4. Press I to change the Interrupt.
   A list of available interrupts appears.
5. Select IRQ9 (2) from the displayed list.
6. Press A to change the Address.
   A list of available addresses appears.
7. Select 2E8h from the displayed list.
8. Press C to change the Connection.
   **Important**  The Connection must be set to UART.
9. Press O to set your touchscreen driver.
10. Press Escape or X to exit.
    **Important**  You must reboot your system for the changes to take effect.

### Configuring the Touchscreen Drivers in Windows

If you are using Microsoft Windows 95 or Windows NT, you can configure the touchscreen drivers using a utility provided by the manufacturer of the touchscreen.

Follow the directions below to configure the touchscreen drivers:
1. Select the Start button, and select Programs.
2. Select the Touch Screen Utilities folder from the Programs menu.
3. Select the Configuration icon from the Touch Screen Utilities folder. The Configuration Utilities dialog box should appear as shown below.

4. Select the Interface button from the Configuration Utilities dialog box. The Interface Configuration Options dialog box should appear as shown below.

5. The touchscreen is set to COM3, IRQ11, 3E8h by default. Click on the drop-down arrows to change to these settings:
   - COM Port: Custom
   - Interrupt: IRQ9
   - Address: 2E8h

   **Note** As you change the COM Port, the other settings will change automatically.

6. Select OK when finished.
Configure the System Support Card

Open the PowerStation

The following steps describe how to open the PowerStation.

1. Turn off the PowerStation unit.
2. Disconnect the unit from its power source.
3. Remove the PowerStation from its enclosure (if installed).
   Consult the PowerStation Installation Instructions if you need help with this removal.
4. Remove the back cover from the unit by removing the screws.
5. Remove the Compact Flash (CF) card.

**Important** Always make sure power to the PowerStation is turned OFF when removing the CF card.

Disconnect the Cables

Use the following diagram to find the cables and other components mentioned in the following steps:
1. Press out the locking ears to disconnect both video cables. The video cable connects at the top left corner of the card. The wide ribbon cable connects very close to the other video cable.

2. Lift the locking ears to disconnect the serial cable from the top center of the System Support card.

3. Disconnect the backlight cable (P6) by depressing the latch on the top and sliding the connector out.

4. Disconnect the touchscreen cable (P5) by depressing the latch on the top and sliding the connector out.

Remove the System Support Card

Use the following diagram to find the cables and other components mentioned in the following steps:

1. Remove the tang screw from the System Support card tang.

2. Carefully pull out the System Support card, and lay it to the side.  

   **Important** Be careful to observe ESD guidelines when handling the System Support card.
Configure the System Support Card

Use the following diagrams to help you physically move the touchscreen from COM3 to COM4. Follow these steps to locate and change the jumpers.


2. Move the TOUCH jumper at IRQ11 to IRQ9 as shown below.
3. Change jumper JP8 according to the RS setting you need.

4. Check that jumper JP9-A is OFF, and jumper PJ9-B is ON, as shown.

This sets the System Support card to COM2.
5. Change jumper JP11-B from OFF to ON.

This sets the touchscreen to COM4.

Configure the Serial Port Card

Change the RS Settings (if required)

The following steps describe how to change the RS communication settings. Perform this step only if you need to change the interface to RS232 or RS422 communication. Use the following diagram to locate and set the jumpers.

1. Lay the serial port card on an anti-static mat.
2. Set JP1A to the correct RS setting, as shown below.

**RS232 Configuration**

![RS232 Configuration Diagram]

**RS422 Configuration**

![RS422 Configuration Diagram]
RS485 Configuration

Set the Port Address

Use the following diagram to set the Port 1 address for the serial port card:
Reinstall the System Support Card

The following steps describe how to reinstall the System Support card. Use the following diagram to locate the correct cables.

1. Slide the System Support card back into its slot.
2. Reconnect the touchscreen (P5) and backlight (P6) cables to the System Support card.
3. Reconnect the video cable and wide ribbon cable to the System Support card.
4. Attach the card with the tang screw.

Install the Serial Port Card

The following steps describe how to install the serial port card.

1. Slide the card into the appropriate ISA slot.
2. Insert screw into tang hole and tighten.
3. Insert the CF card into the PowerStation’s flash card slot.
   **Important** Always make sure power to the PowerStation is turned OFF when inserting the CF card.
4. Replace the back cover.
5. Reconnect the power and boot up the machine.

Configure the New Serial Port in Windows

The following steps describe how to configure the new serial port in Windows.

1. Run Windows 95.
2. Click the **Start** button, and select **Settings/Control Panel**.
3. Double-click the **Add New Hardware** icon.
4. In the first page of the Add New Hardware dialog box, click **Next**.
5. Click the **No** radio button; then, click **Next**.
   
   You do not want Windows to search for your new hardware.
6. Scroll down the list of hardware types to select “Ports (COM & LPT);” then, click **Next**.
7. Highlight “Communications Port;” then, click **Next**.
8. Click **Next**; then, click **Finish** to complete the installation.
9. Return to the Control Panel, and double-click the **System Icon**.
10. Select the Device Manager tab of the System Properties property sheet.

11. Double-click **Ports**.

12. Highlight the last port (this will be the one you just installed - COM3), and click **Properties**.

13. Select the Resources tab of the System Properties property sheet.

14. If necessary, correct the address and interrupt information by highlighting the incorrect data and clicking “Change Setting.”

   **Note** The Windows Device Manager may show the touchscreen IRQ/Address incorrectly. The system will function as long as you have you set the touchcfg.exe settings correctly.

15. Reboot the system to activate your changes.

You have successfully installed and configured the third serial port on your PowerStation. You can now use this port to communicate with external devices.
Appendix C

Excerpt from MachineLogic Quick Start Guide
Getting Started

Welcome to MachineLogic. MachineLogic is a complete programming environment designed to create programs for performing operations that are typically performed by PLCs. MachineLogic is based on the IEC 1131-3 specification in order to provide complete compatibility with other software and hardware that is based on this standard.

Chapter Contents

<table>
<thead>
<tr>
<th>Chapter Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using This Manual</td>
<td>2</td>
</tr>
<tr>
<td>Included Documentation</td>
<td>3</td>
</tr>
<tr>
<td>Related Documentation</td>
<td>4</td>
</tr>
<tr>
<td>Documentation Standards</td>
<td>5</td>
</tr>
<tr>
<td>Customer Support Services</td>
<td>6</td>
</tr>
<tr>
<td>Product Technical Support</td>
<td>6</td>
</tr>
<tr>
<td>Chapter Summary</td>
<td>6</td>
</tr>
</tbody>
</table>
Using This Manual

This manual is designed to provide detailed information about MachineLogic. You will learn how to install and configure MachineLogic. You will also learn how to create and download MachineLogic projects. You should read and understand the concepts presented in this manual prior to operating MachineLogic. This manual is divided into the following chapters:

Chapter 1 - Getting Started: Presents an overview of MachineLogic and the documentation that is provided with MachineLogic. This chapter also provides an overview of the support services provided by CTC.

Chapter 2 - Installing MachineLogic: Provides detailed instructions for installing and configuring MachineLogic. Describes each of the components that need to be installed and configured to develop and run MachineLogic projects.

Chapter 3 - Using the Default Template: Provides general instruction on using the MachineLogic Default Template to create, download, and run MachineLogic projects. Also describes how to back up and upload projects.

Appendix A - MachineLogic Extended Flags: Provides a list of each of the available extended flags that may be specified from MachineLogic Development. Includes the address, data type, name, and a description for each flag.

Appendix B - MachineLogic System Flags: Provides a list of each of the available system flags that may be specified from MachineLogic Development. Includes the address, data type, name, and a description for each flag.

Appendix C - Default Template Variables: Provides a list of each of the variables and data types that have been pre-defined in the MachineLogic Default Template. Includes the name, data type, start address, end address, and the number of available variables.

Appendix D - Autoexec.bat and Config.sys Files: Provides the text of the AUTOEXEC.BAT and CONFIG.SYS files that are required for MachineLogic Runtime systems.
Included Documentation

The MachineLogic User Manual includes all of the MachineLogic documentation you need to install, configure, and run the Beta release of MachineLogic. In addition to the MachineLogic User Manual, MachineLogic includes extensive online help that is available throughout the product. Please review the following list to verify that you have received all necessary documentation:

License Agreement - This section of the MachineLogic User Manual explains the terms of the MachineLogic License Agreement and serves as a certificate of authenticity. Make sure you read and understand the information in this agreement before you install and operate MachineLogic.

Customer Support Information - This section of the MachineLogic User Manual describes how to contact CTC with your comments, questions, and concerns regarding the MachineLogic product.

MachineLogic Quick Start Guide - This section of the MachineLogic User Manual is the manual you are reading. Refer to the Using This Manual section on page 2 for a description of this manual.


MachineLogic Tutorial - This section of the MachineLogic User Manual provides step-by-step instructions for using the MachineLogic Tutorial Template to create, download, and run a MachineLogic project.

Online Help Files - MachineLogic includes extensive online help that is accessible from various points throughout MachineLogic Development. The online help contains detailed information on MachineLogic and describes how to perform specific operations in MachineLogic Development and the MachineLogic driver for Interact. To access the online help available for a specific part of the program, press the F1 key on your keyboard.
In addition to the documentation that is included in the MachineLogic User Manual, you should become familiar with the following documentation:

**I/O Interface Card Documentation** - This documentation is supplied by the manufacturer of your I/O interface card. Please read this documentation in its entirety prior to downloading or running a MachineLogic project.

**IPM User Guide** - This document describes how to install and use the Interact Project Manager to download Interact projects from a PC to a PowerStation.

**PowerStation User Guide** - This guide explains how to configure, install and operate the PowerStation. This manual also contains information about the PowerStation Shell Utility. Please familiarize yourself with this manual prior to operating the unit.

**PowerStation CPU Manual** - This document is the manual provided by the manufacturer of the CPU board found in your PowerStation. Please familiarize yourself with this manual prior to operating the unit.

**Interact Getting Started Guide** - The Interact Getting Started Guide is designed to help you get up and running with Interact. This manual takes you through the installation of Interact, describes Interact concepts, and includes example applications detailing the features of most Interact modules.

**Interact Tutorial** - The Interact Tutorial is an online help file that is designed to be run on the Interact development system. This help file includes step-by-step instructions for creating example applications using the Panel Toolkit Module (PTM), the Graphics Monitoring Module (GMM), and the Alarm Management Module (AMM), which will help you become familiar with Interact.

**Interact Online Help Files** - Each Interact module and device driver includes several online help files that are accessible from various points in the Interact software. These files contain detailed information on Interact and describe how to perform specific operations in Interact. To access the online help available for a specific part of the program, press the F1 key on your keyboard.
Windows Documentation - This manual assumes you have a thorough knowledge of the Windows operating system when working with MachineLogic. If this is not the case, then refer to the documentation shipped with your operating system for details on Windows.

Documentation Standards

As you read this manual, you will notice that the following documentation standards have been followed.

1. **Important** terms are shown in **bold**.

2. Text to be entered from the keyboard is shown in **Courier font**.

3. Buttons, menu titles, and keyboard keys are shown in **Initial Caps**.

4. Indented paragraphs denote one of the following:
   - **Note** - Describes alternative approaches or issues you should be aware of while using a particular function.
   - **Important** - Contains information that needs particular attention while reading the text. Follow this information to save development time and/or minimize problems.
   - **Warning** - Contains information on safety issues. Follow this information to prevent equipment damage or personal injury.

ISO Symbols

This symbol is the International Standards Organization (ISO) symbol for Caution (ISO 3864 No. B.3.1). This symbol denotes information that could affect operation of the PowerStation if not properly followed.

This symbol is the ISO symbol for Caution - risk of electrical shock (ISO 3864 No B.3.6). This symbol denotes information that could cause personal injury from electrical shock or damage to equipment if not properly followed.
Customer Support Services

CTC welcomes your thoughts and suggestions on our products and services. You can contact CTC by telephone, email, or fax. You can also visit CTC on the World Wide Web to learn the latest about CTC hardware, software, and customer support services.

- Telephone: 513-831-2340
- Fax: 513-831-5042
- E-mail: Miami@ctcusa.com
- World Wide Web: http://www.ctcusa.com

Product Technical Support

The MachineLogic Development Group welcomes any question that might arise as you develop or run your applications. We offer complimentary support for any customer, whether you are an end user, original equipment manufacturer (OEM), system integrator, or distributor.

If you have a question about MachineLogic, be sure to complete the following steps before contacting the MachineLogic Development Group:

1. Check the Readme file installed with the MachineLogic software. This file provides general information about the release.

2. Consult the MachineLogic documentation and other printed materials included with MachineLogic.

3. Check the online help. MachineLogic has an extensive online help facility that covers all aspects of the product.

If you cannot find a solution using one of the above sources, refer to the Customer Support Info. section of the MachineLogic User Manual.

Chapter Summary

This chapter has provided a general overview of the documentation that is provided with MachineLogic as well as the supporting documentation that you may find useful as you develop MachineLogic projects. This chapter has also provided an overview of CTC’s support products and services.
Chapter 2

Installing MachineLogic

This chapter provides detailed instructions for installing and configuring MachineLogic software and hardware on a development system and a runtime system. The chapter is divided into two main sections with tasks grouped by system type.

<table>
<thead>
<tr>
<th>Chapter Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development System</td>
<td>10</td>
</tr>
<tr>
<td>Runtime System</td>
<td>18</td>
</tr>
<tr>
<td>Chapter Summary</td>
<td>31</td>
</tr>
</tbody>
</table>
The development system is the computer you plan to use to create and manage your MachineLogic projects. This computer will not be used to run the projects that you create. Instead, you will use this computer to download your projects to a PowerStation that will be used to run your projects. The PowerStation is referred to as the runtime system throughout this manual.

Your development system must meet or exceed the following system requirements in order for MachineLogic to perform as intended:

- Processor: Pentium 133MHz
- Memory: 16MB RAM (Windows 95 systems)
- Memory: 32MB RAM (Windows NT systems)
- Fixed Storage: 50MB of free hard disk space
- Removable Storage: 3.5” floppy and CD-ROM drive
- Input Devices: keyboard and mouse

Your MachineLogic development PC requires the following software in order to develop MachineLogic projects:

- Operating System: Windows 95/NT

  Note: DCOM 95 must be installed if using Windows 95. A self-extracting executable install is provided on the MachineLogic install disk in the DCOM 95 directory.

- Interact: Version 5.2 or later
- MachineLogic Development: Version 2.0 Build 457 or later

If you need to install any or all of the software listed above, you can follow the steps provided in this section to do so. The MachineLogic Beta2 CD includes all of the components that are necessary to install and configure MachineLogic and Interact.
Chapter 2: Installing MachineLogic Development System

Installing Interact

The procedure for installing Interact is described in detail in the Interact Getting Started Guide. If you are using the MachineLogic Beta2 CD to install Interact, substitute references to a:\setup with d:\Interact\Disk 1\Setup.exe. You may also need to substitute the drive letter of your CD-ROM drive for d:

Installing MachineLogic Development

You must install Interact prior to installing MachineLogic Development. See Installing Interact above for more information.

The procedure for installing MachineLogic Development is described in detail below. Refer to this section whenever you need to install or reinstall MachineLogic Development.

1. Insert the MachineLogic Beta2 CD into your CD-ROM drive.
2. Click on Start, and choose Run.
   The Run dialog box appears.
3. Type d:\MachineLogic\Disk 1\Setup.exe next to the word “Open” in the Run dialog box (If necessary, substitute your CD-ROM drive letter for d:).
4. Press Enter.
   The InstallShield Wizard displays for several seconds as the installation program prepares to prompt you through the install process. Next, the Welcome screen appears.
5. Click Next to leave the Welcome screen.
The Software License Agreement screen appears.

6. Review the License Agreement.
7. Click Yes if you agree to the terms of the License Agreement, or click No if you do not agree to the terms.

If you click Yes, the Information screen appears. If you Click No, the installation program will stop immediately.

8. Review the Information screen.
9. Click **Next** to leave the Information screen.
   The User Information screen appears.
10. Enter your name and company name.
11. Click **Next** to leave the User Information screen.
   The Choose Destination Location screen appears as shown below:

   ![Choose Destination Location Screen]

   By default, MachineLogic Development will be installed to the path `C:\MachineLogic`. If you wish to install MachineLogic in a different directory, click **Browse** to select the desired folder.

12. Click **Next** to leave the Choose Destination Location screen.
The Select Program Folder screen appears.

![Select Program Folder Screen](image)

13. Enter a name for the program folder. MachineLogic is the default program folder name.
14. Click **Next** to continue.

The Start Copying Files screen appears as shown below:

![Start Copying Files Screen](image)
15. Review the Start Copying Files screen.

**Important** The MachineLogic installation program detects if Interact and Interact Project Manager (IPM) are installed on your development system. If Interact is installed, the installation program will install an updated version of Interact and the Interact MachineLogic driver which allows Interact to run with MachineLogic.

In addition, the MachineLogic installation will automatically install Interact Project Manager (IPM) if it was not detected by the installation program.

16. Click **Next** to begin copying files onto your system and to the Start Copying Files screen.

Once the installation is complete, the Setup Complete screen appears.

17. Click **Finish** to finalize the installation.

**Getting a MachineLogic Enable Code**
MachineLogic requires an Interact security key and enable code for Beta2 development. Before running MachineLogic, make sure that the MachineLogic development system has the Interact software installed and the security key attached to the parallel port.

Contact CTC Technical Support to receive a MachineLogic enable code for the Interact security key.

**Enabling the Security Key**
Once you have received a MachineLogic enable code, you will need to program the Security Key to enable MachineLogic using the Interact Key Utility. For information on how to use the Interact Key Utility, see the following instructions.
Using the Interact Key Utility

The Interact Key Utility allows you to program your Security Key. You can enable additional software options or transfer enabled software options between keys.

Selecting Interact Key Setup from the Interact folder or from the Interact Program Group displays the Interact Key Utility screen.

This screen displays a check mark next to the options that are enabled within your key. A single check mark in the Runtime column means that option is enabled for runtime operation only. Check marks in the Configuration and Runtime columns indicate that the option is enabled for application development and runtime.

**Note** Make sure to contact CTC Technical Support to receive a MachineLogic enable code.

For the MachineLogic enable code, select the Enable Option command from the menu, and enter the code as prompted. After the code has been validated by the Key Setup Utility, it will enable the option within the key and display check marks in the appropriate columns. See *Enabling Software Options* below for step-by-step procedures.

**Enabling Software Options**

When you call CTC Technical Support for your MachineLogic enable code, you will receive a Module Enable Code Certificate listing the enable code for MachineLogic. This enable code is used to activate MachineLogic on the Interact security key. The code is generated for the key number and serial number printed on your key and are valid ONLY for that key.
To enable software options, follow these steps:

1. You must first install Interact on your development system.

2. Make sure your key is attached to the parallel port of your computer.

3. Go to the CTC Interact folder or Interact Program Group.

4. Click the Interact Key Setup icon.
   The Key Setup Main Menu is displayed.

5. Select the Enable Option command.

6. Enter the enable code to activate MachineLogic.
   As the enable code is entered, you will see a check mark appear on the display to the right of the software option that is enabled. If the enable code was for activating development, two check marks will appear, one in the Configuration column and one in the Runtime column.

7. Return to the Key Setup Main Menu.

8. Exit the Interact Key Setup by selecting the Exit Menu option.
   You are returned to the Windows desktop.

For more information about security keys, see Working with Interact Security Keys in the Interact Getting Started manual.
Runtime System

The MachineLogic runtime system is the PowerStation you plan to use to run your MachineLogic projects. This computer will only be used to run the projects you create. You will develop your projects on a development system and download them to this computer.

Hardware Requirements

Your PowerStation must meet or exceed the following MachineLogic system requirements in order for MachineLogic Runtime to perform as intended:

- Processor: 80386 40MHz or higher
- Memory: 4MB
- Storage: 4MB
- I/O Card: S-S Technologies (SST) 5136-PFB/5136-PFB-104 Profibus card
- MachineLogic Control Adapter

Important The MachineLogic Control Adapter must be installed before you can use MachineLogic Runtime.

Installing the Hardware

In order to run MachineLogic projects on your runtime system, you must install the MachineLogic Control Adapter and a Profibus I/O card. The MachineLogic Control Adapter is currently available in PC/104 format and soon to be available in ISA format.

Electrostatic Discharge Precautions

Modern integrated electronic devices, especially CPUs and memory chips, are extremely sensitive to electrostatic discharges (ESD) and fields. Before you install the MachineLogic Control Adapter, be sure to follow these simple precautions to protect you and the PowerStation from harm resulting from ESD.

1. To avoid electric shock, always disconnect the power from the PowerStation before you remove the back cover. Do not touch any internal components while the PowerStation is on.
2. Disconnect power before making any hardware configuration changes. The sudden rush of power as you connect a jumper or install a card may damage sensitive electronic components.
3. Only handle internal components in an ESD-safe location using appropriate grounding methods.
   - Be particularly careful not to touch the chip connectors.
   - Keep the PC/104 module in its anti-static packaging when it is not installed in the PowerStation, and place it on a static-dissipative mat when you are working on it.
   - Wear a grounding wrist strap for continuous protection.

If you have read and understand the ESD safety precautions, you are ready to disassemble the PowerStation.

Disassembling the PowerStation

If you have read and understand the ESD safety precautions, you are ready to disassemble the PowerStation. Follow these steps to disassemble the unit:

1. Turn off power to the unit.
2. Disconnect the unit from the power source.
3. Place the unit on a static-dissipative mat in a location free from dirt and moisture and protected against static discharge. You must also wear an ESD wrist strap connected to a good known earth ground.
4. Remove the screws securing the back cover.
5. Remove the back cover from the unit.

Installing the MachineLogic Control Adapter

When you finish disassembling the PowerStation, you are ready to install the MachineLogic Control Adapter. The MachineLogic Control Adapter is currently available in PC/104 format and soon to be available in ISA format.
To install the MachineLogic Control Adapter into a PC/104 slot, complete the following steps:

1. Carefully align the PC/104 module connector over the PC/104 connector.

2. Press down firmly on the PC/104 module until the module is completely seated in the connector.

   **Important** Be careful not to bend any of the pins on the PC/104 module as you press the module into place.
3. Secure the PC/104 module using the hardware that was supplied with the module. Use screws if you are only installing one module; use stand-offs if you are planning to install additional modules.

![install screws or stand-offs here](image)

**Note** The PowerStation CPU card requires metric screws and standoffs; however, the Profibus I/O card comes with SAE screws and standoffs. For your convenience, metric screws and standoffs have been provided with the MachineLogic Beta2 package.

4. If you are installing a second PC/104 module, such as the SST Profibus card, connect the second module to the first one by pressing the second modules’ pins into the connector of the first module.

**Note** You can install multiple PC/104 modules, one on top of the other, provided that the modules do not exceed the electrical requirements of the PC.

5. Remove one or more expansion slot covers to create an opening for routing cables connected to the card(s).

6. Connect any necessary cabling to the PC/104 module(s), and route it through the slot opening you created in step 5.
Installing the I/O Card

The I/O card is the hardware device that allows MachineLogic to send and receive data from your equipment over an I/O bus. The I/O card is available from a third-party manufacturer. Refer to the manufacturer’s documentation for information on installing and configuring the I/O card.

Note: If you are installing a PC/104 I/O card, CTC recommends that you install the I/O card on top of the MachineLogic Control Adapter in order to make the switches on the I/O card more accessible.

Re-assembling the PC

Once you have finished installing the MachineLogic Control Adapter, you can re-assemble the PC. Follow these steps to re-assemble the unit:

1. Install the back cover.
2. Install the screws that secure the back cover.
3. Re-connect the unit to the power source.
4. Turn on the unit.

Configuring the MachineLogic Control Adapter

Once you have installed the MachineLogic Control Adapter, you are ready to configure and/or change the MLCA.INI file. The MLCA.INI file stores configuration settings used by MLCA.EXE. The text of MLCA.INI appears below:

MLCA.INI

[ControlAdapter]
BaseSeg=\x0000
BaseSize=\x0000
IODBase=\x0710
IRQ=\x05

[UCS]
UCSBase=\x710
UCS1irq=\x0
Appendix D

Excerpt from MachineLogic Tutorial Manual
A Division of Parker Hannifin

MachineLogic Tutorial

50 West TechneCenter Drive
Milford, Ohio 45150
Phone 513.831.2340 • Fax 513.831.5042
# Table of Contents

## Introduction
- What is MachineLogic Development? .......................................................... 1
- Programming Languages ............................................................................. 2
- Program Organization Units (POUs) .............................................................. 4
- Variables and Data Types ............................................................................. 7
- Configuration Elements ............................................................................... 9

## Project 1
- Building Your First Project
  - Start MachineLogic/Setup a Project .......................................................... 12
  - Navigate the Main Menu Bar .................................................................... 14
  - Edit and Add Variables ............................................................................ 19
  - Edit POU Descriptions ........................................................................... 22
  - Create a Ladder Logic Program ................................................................. 23
  - Save and Compile the Project .................................................................. 34
  - Download and Run the Project ................................................................. 35
  - Test the Project ....................................................................................... 38
  - Stop the Project ...................................................................................... 40

## Project 2
- Building a Structured Text Project
  - Insert New Variables ............................................................................... 43
  - Insert a New POU .................................................................................... 45
  - Create a Structured Text Project ............................................................... 47
  - Save the Project ...................................................................................... 50
  - Insert a New Task ................................................................................... 51
  - Assign a POU to a Task ......................................................................... 52
  - Compile the Project ................................................................................ 52
  - Download and Run the Project ............................................................... 53
  - Test the Project ...................................................................................... 55

## Project 3
- Building a Function Block Diagram Project
  - Insert a New POU ................................................................................... 61
  - Create a Function Block Program ............................................................. 63
  - Save the Project ...................................................................................... 70
  - Assign a POU to a Task ......................................................................... 71
  - Compile the Project ................................................................................ 71
  - Download and Run the Project ............................................................... 72
  - Test the Project ...................................................................................... 75

## Project 4
- Building an Instruction List Project
  - Insert a New POU ................................................................................... 83
  - Create an Instruction List Program .......................................................... 84
  - Assign a POU to a Task ......................................................................... 92
  - Compile the Project ................................................................................ 93
  - Download and Run the Project ............................................................... 93
  - Test the Project ...................................................................................... 96

## Project 5
- Building a Sequential Function Chart Project
  - Insert New Variables ............................................................................... 103
  - Insert a New POU ................................................................................... 105
  - Create a SFC Program .......................................................................... 107
  - Save the Project ...................................................................................... 120
  - Insert a New Task ................................................................................... 121
  - Assign a POU to a Task ......................................................................... 122
  - Compile and Download the Project .......................................................... 123
  - Test the Project ...................................................................................... 125

## Project 6
- Combining Interact with MachineLogic
  - Edit Existing Variables .......................................................................... 133
  - Open the I/O Configuration .................................................................... 135
  - Set the I/O Configuration ....................................................................... 135
  - Set the Resource .................................................................................... 137
  - Compile the Project ................................................................................ 137
  - Download and Run the Project ............................................................... 138
  - Test the Project ...................................................................................... 140
Introduction

This Introduction provides you with a basic overview of the software. If you wish to bypass the Introduction, you can proceed to page 12 and immediately begin the first project using the Introduction as a reference section. However, if you do have the time, it will benefit you greatly to read through the Introduction section before proceeding.

This tutorial teaches you the basics of MachineLogic Development. You will learn how to develop a project, how to insert Program Organizational Units (POUs), and how to create programs using the programming languages that execute the project.

We have gone to great lengths in this tutorial to provide you with the most vital information that will get your system up and running as soon as possible. As you start with the step by step procedures, you will quickly find that there is not a lot of unnecessary reading. Every piece of information on these pages is critical to your success.

Before beginning this tutorial, you should review the Getting Started Manual. This will guide you through the setups needed to install MachineLogic Development properly.

IMPORTANT: Light Bulb Icon

Whenever you see this light bulb, please take the time to read the important note next to it. The light bulb icon is used throughout this manual to signify very important information that you do not want to miss because it could impact the rest of the steps in the project. Although all the information in this tutorial is vital to your success in learning MachineLogic, it is especially important that you read these special notes.

What is MachineLogic Development?

MachineLogic Development is a standard programming system for IEC designed PLCs and traditional control systems. It is based on the standard IEC 1131-3 foundation, and includes the full range of IEC features. MachineLogic Development offers powerful features for the different developing phases of a PLC application, which include:

* Editing
* Compiling
Project 1: Building Your First Project

Objectives
You will have mastered the objectives of this project when you can:

- Start MachineLogic
- Setup a Tutorial Template
- Navigate the Main Menu Bar
- Navigate the Project Tree
- Edit Global_Bool_Variables
- Edit POU Descriptions
- Create a Ladder Logic Program
- Save the Project
- Compile the Project
- Download the Project
- Run the Project
- Test the Project
- Stop the Project
To Start MachineLogic and Setup a Project

1. Start MachineLogic by double-clicking the MachineLogic shortcut icon on the desktop.

If you do not have a shortcut defined, click the Start button, and then choose MachineLogic from the Programs folder.

2. Close all active projects. On the File menu, click Close Project.

If you have been working on a project and exited MachineLogic, the last project you were working on will be the first project displayed when you restart MachineLogic.
3

On the File menu, click New Project.

The Project Template Window appears.

Figure 3

4

Select the Template for Tutorial.

You must select a template. The Tutorial Template has pre-configured the data types that you will be using for your project.

Figure 4

A template is a set of configurations for a specific project. MachineLogic has a number of pre-configured templates. One template, the Tutorial template, is setup to allow the user to develop from a pre-configured list of variables. Another template, the RTXDO16 template, is a fully configured MachineLogic template. This template has all of the necessary variables pre-defined for the user.
Select OK.

Figure 5 shows how the first project will look when it is displayed.

To Navigate through the Main Menu Bar.

Drag the cursor across the menu bar.
Click File.

Figure 7 shows you a table of each File command and its description.

<table>
<thead>
<tr>
<th>SUB-TOPIC</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Project</td>
<td>Opens a new template and project</td>
</tr>
<tr>
<td>Open Project/Unzip Project</td>
<td>Open or Unzip a project</td>
</tr>
<tr>
<td>Save Project/Zip Project As</td>
<td>Save project with different filename</td>
</tr>
<tr>
<td>Close Project</td>
<td>Close the project</td>
</tr>
<tr>
<td>Delete Project</td>
<td>Deletes project from hard drive</td>
</tr>
<tr>
<td>Save</td>
<td>Save the displayed project</td>
</tr>
<tr>
<td>Close</td>
<td>Close the displayed project</td>
</tr>
<tr>
<td>Print</td>
<td>Prints the active project</td>
</tr>
<tr>
<td>Print Preview</td>
<td>Displays full pages</td>
</tr>
<tr>
<td>Printer Setup</td>
<td>Change the printer and printer options</td>
</tr>
<tr>
<td>Exit</td>
<td>Exits entire project and closes MachineLogic</td>
</tr>
</tbody>
</table>

Figure 7

As you drag across the menu bar, you will pull down the following information. This information resides only on the opening menus. As you progress through the project, the menu options will change.

Click Edit.

Figure 8 shows a table of each Edit command and its description.

<table>
<thead>
<tr>
<th>SUB-TOPIC</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undo</td>
<td>Undo last action</td>
</tr>
<tr>
<td>Redo</td>
<td>Removes previous undo</td>
</tr>
<tr>
<td>Cut</td>
<td>Removes selected area to clipboard</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies highlighted area</td>
</tr>
<tr>
<td>Paste</td>
<td>Place contents of clipboard at cursor location</td>
</tr>
<tr>
<td>Delete</td>
<td>Removes highlighted area</td>
</tr>
<tr>
<td>Insert</td>
<td>Allows user to add or insert defined functions</td>
</tr>
<tr>
<td>Select All</td>
<td>Select entire document</td>
</tr>
<tr>
<td>Find</td>
<td>Searches page for keyed sequence</td>
</tr>
<tr>
<td>Find Next</td>
<td>Finds next occurrence</td>
</tr>
<tr>
<td>Replace</td>
<td>Replace specific text with different text</td>
</tr>
</tbody>
</table>

Figure 8
Click View.

Figure 9 shows a table of each View command and its description.

<table>
<thead>
<tr>
<th>SUB-TOPIC</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Tree Window</td>
<td>Directory structure of project</td>
</tr>
<tr>
<td>Message Window</td>
<td>Compile window</td>
</tr>
<tr>
<td>Cross Reference Window</td>
<td>Displays compiled cross-references</td>
</tr>
<tr>
<td>Watch Window</td>
<td>Used with Watch List</td>
</tr>
<tr>
<td>Edit Wizard</td>
<td>Language Edit feature</td>
</tr>
<tr>
<td>Status Bar</td>
<td>Text help on bottom of screen</td>
</tr>
</tbody>
</table>

Figure 9

Click Project.

Figure 10 shows a table of each Project command and its description.

<table>
<thead>
<tr>
<th>SUB-TOPIC</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Library</td>
<td>Append new library to a project</td>
</tr>
<tr>
<td>Add Datatype</td>
<td>Append a new Data Type Worksheet to a project</td>
</tr>
<tr>
<td>Add POU</td>
<td>Function Block, Program, Function POU</td>
</tr>
</tbody>
</table>

Figure 10

Click Build.

Figure 11 shows a table of each Build command and its description.

<table>
<thead>
<tr>
<th>SUB-TOPIC</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make</td>
<td>Used to compile the current worksheet (F9)</td>
</tr>
<tr>
<td>Patch POU</td>
<td>Used to compile changes to the existing worksheet</td>
</tr>
<tr>
<td>Compile Worksheet</td>
<td>Compiles changes while debugging (See Note)</td>
</tr>
<tr>
<td>Rebuild Project</td>
<td>Rebuilds entire project (Ctrl+F9)</td>
</tr>
<tr>
<td>Stop Compile</td>
<td>Stops compile in mid process</td>
</tr>
<tr>
<td>Go to Next Error</td>
<td>Toggles next display warnings</td>
</tr>
<tr>
<td>Go to Previous Error</td>
<td>Command to display previous errors and warnings</td>
</tr>
<tr>
<td>Build Cross-references</td>
<td>Builds a cross reference of tags</td>
</tr>
</tbody>
</table>

Figure 11

Note: A Patch POU can be used to compile the changes you have done while debugging. If you have detected a programming error using the online mode and have switched to the off-line mode to remove the programming error, you can use Patch Worksheet to compile these changes.
Click Online.

Figure 12 shows a table of each Online command and its description.

<table>
<thead>
<tr>
<th>SUB-TOPIC</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug</td>
<td>Switch to Debug Mode</td>
</tr>
<tr>
<td>Resource Control</td>
<td>Show control dialog</td>
</tr>
<tr>
<td>Powerflow</td>
<td>Switch powerflow on/off</td>
</tr>
<tr>
<td>Online Layout</td>
<td>Change Online Layout</td>
</tr>
</tbody>
</table>

Figure 12

Click Options.

Figure 13 shows a table of each Options command and its description.

<table>
<thead>
<tr>
<th>SUB-TOPIC</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pagelayout Editor</td>
<td>Opens the pagelayout editor</td>
</tr>
<tr>
<td>Shortcuts</td>
<td>Opens the Shortcut Keys dialog</td>
</tr>
<tr>
<td>Preferences</td>
<td>Opens the Options dialog to change preferences</td>
</tr>
</tbody>
</table>

Figure 13

A brief explanation of the Options dialog appears below:

### DIALOG FIELD | MEANING
---|--------------------------------------------------
Toolbars | hide, show or create toolbars.
Commands | displays short description for each icon, assign icons to the toolbars and customize the menu bar.
General | specifies general settings for the user interface.
Build | specifies settings for compiling and saving the worksheets.
Directories | defines the default paths for the project directory, library directory and page layout directory.
Graphical Editor | displays the default size of the graphical worksheets and the default settings
Default Page layouts | displays the different page layouts
Text Editor | define the tab size, the font type and size, the column width in online worksheets and graphical worksheets
Text Colors | defines the text color in which the specific information appears in the text editor
Debug | show or hide the specific warning message when debugging worksheets.

Figure 14
Click Help.
Figure 15 shows a table of each Help command and its description.

Double-click the Logical POU's icon.
This causes the Project Tree to open (Figure 16).

Note: The Project Tree is discussed in detail in the Introduction section of this Tutorial.
To Edit and Add Variables

1. Double-click the Ladder branch of the Project Tree.

As the Ladder branch opens you see the options for POU Description, Local Variables, and the Ladder Program (Figure 17).

Figure 17

Variables are used instead of direct addressing of memory regions in IEC 1131-3 programming systems. Variables are names or place holders which are defined by the user to represent the corresponding data. In the following steps, we will configure 8 Direct_Global_Bool_Variables, and we will modify their names and AT addresses.
Double-click the Physical Hardware section of the Project Tree.

Notice that this is where the global variables reside for the entire project (Figure 18).

You will notice that there are 10 variables previously declared in this Tutorial Template. We will be adding new names and adding new memory addresses.
Double-click the Global_Bool_Variables icon.

The Declaration of variables or FB instances dialog appears.

Notice the list of Direct_Global_Bool variables that are listed (Figure 19).

![Figure 19](image)

We will change the above variables to look like the ones that are listed below.

<table>
<thead>
<tr>
<th>DIRECT_GLOBAL_BOOL_0</th>
<th>changed to START_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>%MX0.0 to %IX0.0</td>
<td></td>
</tr>
<tr>
<td>DIRECT_GLOBAL_BOOL_1</td>
<td>changed to START_2</td>
</tr>
<tr>
<td>%MX0.1 to %IX0.1</td>
<td></td>
</tr>
<tr>
<td>DIRECT_GLOBAL_BOOL_2</td>
<td>changed to STOP_1</td>
</tr>
<tr>
<td>%MX0.2 to %IX0.2</td>
<td></td>
</tr>
<tr>
<td>DIRECT_GLOBAL_BOOL_3</td>
<td>changed to STOP_2</td>
</tr>
<tr>
<td>%MX0.3 to %IX0.3</td>
<td></td>
</tr>
<tr>
<td>DIRECT_GLOBAL_BOOL_4</td>
<td>changed to RUNNING_1</td>
</tr>
<tr>
<td>%MX0.4 to %QX0.4</td>
<td></td>
</tr>
<tr>
<td>DIRECT_GLOBAL_BOOL_5</td>
<td>changed to RUNNING_2</td>
</tr>
<tr>
<td>%MX0.5 to %QX0.5</td>
<td></td>
</tr>
<tr>
<td>DIRECT_GLOBAL_BOOL_6</td>
<td>changed to RESET</td>
</tr>
<tr>
<td>%MX0.6 to %IX0.6</td>
<td></td>
</tr>
<tr>
<td>DIRECT_GLOBAL_BOOL_7</td>
<td>changed to IN_RANGE_2</td>
</tr>
<tr>
<td>%MX0.7 to %QX0.7</td>
<td></td>
</tr>
</tbody>
</table>

Notice we are not only changing the name of the variable, we are also changing its memory address. The steps that follow will lead you through the editing process.
4. Change the name of each variable and edit its address.

5. Close and save the variable sheet.

---

To Edit POU Descriptions

1. Double-click the POU Description.

   Enter the text.

   Close the dialog and select Yes to save the string.

   First, delete any current text, then type the description that you see in Figure 21.
To Create a Ladder Logic Program

1. Double-click the Ladder Program icon.

Delete any text that you find in this dialog window (Figure 22). You may not find any text on your worksheet.

Figure 22

For ease of configuration you may wish to maximize your windows in such a way as to display both the Project Tree and the Ladder programming window as shown below in Figure 23.

Figure 23
2
Click the Ladder or Graphic Worksheet.
This will display the ladder “tools” across the top portion of your screen (Figure 24).

Figure 24

💡 **Note**: You will notice a black plus icon on your ladder window.

3
Click the Contact Network button to insert a contact network.

We are developing a ladder program for our first project. This ladder logic will demonstrate the two ways in which you can configure a latched to unlatched circuit. Although this is a simple example, it will show you how to configure a ladder using this package.

Figure 25

Notice that when you select the Contact Network icon, the contact and coil appear in the ladder window.
These are the contact/coil setup options that will need to be configured in Figure 26.

**Contact /Coil Setup**

- Contact/Coil set to **Contact** and Type set to **Normally Open Contact**.
- Scope set to **Global**.
- Local Variables Worksheets: **global_variables**.
- Global Variables Worksheets: **global_bool_variables**.
- Select the proper variable name by clicking the Variable list and selecting **Start_1** (see Figure 27).
- After you have entered in the proper data, click **OK**.

4

Place your cursor on the first normally open contact labeled C000.

Double-click the highlighted and replace C000 with Start_1 (see Figure 27).

You must highlight the contact. This will call up the Contact/Coil dialog window as shown in Figure 26.

![Figure 26](image-url)
5

Set the following parameters.

Spend some time here to make sure you understand how these options are set for each ladder instruction.

Click OK.

---

6

Double-click the coil C001.

Assign Running_1 to the coil.

Click OK.
After you have assigned the Running_1 variable name to the coil and have selected OK, you will notice that the rung displays the variable name (Figure 29).

Click the Start_1 contact.
Click the Add Contact/Coil Below button.
This will be C0002.

This option adds the branched contact and coil. Pay close attention to how the branch is structured (Figure 30).
Appendix E

MachineShop Shell User Interface
Specification Document
MACHINE SHOP / POWERSTATION SHELL
USER INTERFACE SPECIFICATION

REV: 0.1
DATE: January 15, 1999
BY: Kurt Schulz

This document contains proprietary information of CTC-Parker Automation and is tendered subject to the conditions that the information:
   a) be retained in confidence,
   b) not be reproduced or copied in whole or part, and
   c) not be used or incorporated in any product except under an expressed written agreement

Approvals

<table>
<thead>
<tr>
<th>CTC-Parker: Product Manager</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTC-Parker: Software Development Manager</td>
<td>Date</td>
</tr>
<tr>
<td>CTC-Parker: Miami Team - Technical Lead</td>
<td>Date</td>
</tr>
<tr>
<td>CTC-Parker: Miami Team - Marketing Lead</td>
<td>Date</td>
</tr>
<tr>
<td>Revision</td>
<td>Date</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>0.1</td>
<td>January 15, 1999</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. **SCOPE**

1.1. The scope of this document is to describe the user interface design for the CTC-Parker Runtime Shell, hereafter referred to as the "Shell".

1.2. The details of the basic major functionality and communications protocol of MachineShop and Runtime Shell are contained in separate documents and will not be described here except as required for clarity. Please refer to the *MachineShop Specification*, *MachineShop/PowerStation Shell Communication Protocol Specification* and the *Miami/Powerstation Shell Marketing Objectives* documents for this information.
2. GENERAL

2.1. As stated in the Marketing Objectives document, the Shell serves two basic functions: communications with MachineShop and hardware support functions for configuration and information. This document focuses on the user interface and support functions.

2.2. The Shell must run under DOS 6.0 and higher, including DOS 7.x used by Windows 95/98 Dos Mode. The Shell will use Tenberry's DOS 16/M extender to run in extended memory in order not to be confined by the limitations of conventional memory. It will also run in graphics mode using the MetaWindows library from MetaGraphics Corp. for display generation. For maintainability, whenever possible the Shell should use the same versions of these packages as used by the most current release of Interact runtime.

2.3. The screens of the Shell must be designed to run on a variety of hardware platforms supporting various display resolutions and dimensions. At the low end, a resolution of 320x240 must be supported for the P1 PowerStation 6" display. The screen designs must also be practical for use on larger displays which can range up to 14". To accomplish this, the Shell screens will be designed with a maximum size of 320x240 pixels running in a 640x480 resolution mode.

2.4. For use on the P1, the screens must be rendered in the upper left corner, which is the visible area on a 1/4 VGA display. For all other platforms, the screens must be rendered in the center of the display area. On larger displays designed for use at higher resolutions, the Shell will still run in a 640x480 resolution mode -- the screens will just appear larger as they are expanded by the video system to fill the display area.

2.5. The Shell must detect whether or not the video system can support VESA mode 101h. If so, it will use that mode and render the screens in 256 colors. If not, it must use standard VGA mode 10h and render the screens in 16 colors (or 16 shades if on a monochrome system).
3. SCREEN AND MENU DESIGNS

3.1. The Shell screens and menus are designed differently than the existing Shell in order to make it easier to provide one Shell for use on a variety of display systems and to produce a more aesthetically pleasing appearance. The primary user input device will be touch screen, and so large graphical buttons and touch areas will be implemented.

3.2. Basic Screen Layout

3.2.1. The basic layout of the screens consists of a "title area" across the top of the display, 6 "buttons" down the left side for selecting menu options, and a "display area" covering the middle and right side for showing information. All screens have a selection for calling up on-line help text. An example is shown below.

3.2.2. A horizontal "sub-menu" with three or more buttons typically appears when the user presses one of the six buttons. The user may make a selection in the sub-menu, or may press outside the sub-menu to have it disappear without making a selection.

3.2.3. It is proposed that a 320x240 "background graphic" exists as a separate file which is loaded and displayed by the Shell as it initializes. The graphic can be used to easily customize the display with titles, logos, colors, or any other unique features as required. Monochrome, 16 color and 256 color versions of the graphic can be provided for use on the various video systems. The Shell can load the appropriate background graphic during initialization after determining the video display type. All text and menu buttons are then drawn on the background during runtime as needed. The default background graphic for use on PowerStations is shown below. The
3.2.4. **Mouse Usage**

3.2.4.1. The Shell will support the use of a standard Microsoft mouse pointing device, allowing it to also work with existing touch screen drivers emulating a mouse. The left mouse button will be used to initiate all actions – the right button will not be used. Actions will take place upon button “release”.

3.2.4.2. The cursor will be an arrow pointing upward and to the left, similar to that used in the Interact product.

3.2.5. **Keyboard Usage**

3.2.5.1. The arrow keys on a keyboard can be used to move the screen cursor to the active touch areas on the display. The cursor (an arrow pointing upward and left) will automatically jump to an *area inside the lower right corner* of the nearest touch area in the direction of the arrow key pressed.

3.2.5.2. The ENTER key may be used to make the selection, emulating pressing the left mouse button or touching the screen. In addition to the arrow keys, it is proposed that the TAB key be supported for cycling through the active touch areas. Pressing TAB will cycle forward and pressing SHIFT-TAB will cycle backwards.
3.3. The Screens and Menus

3.3.1. The display area of the Main Screen shows information about the current project loaded on the runtime system. This includes the Project Name, the Interact Modules, Drivers, and Applications, and the MachineLogic I/O subsystem. If no Project exists, then "None" is displayed after Project Name and no further information is displayed. If Interact or MachineLogic are not a part of the Project, then their respective information is not displayed.

3.3.2. The RUN menu option is used for selecting a program to launch from the Shell. At this time, Interact is the only option needed. Upon pressing the RUN button, the menu shown below is displayed. After pressing the INTERACT button, Interact is launched by the Shell, just as it does now in the current shell.
3.3.3. The SETTINGS menu button is used to access options for making configuration settings. Two buttons may appear upon selecting this menu item: GENERAL settings and PWRSTA settings. The PWRSTA button appears only if the Shell detects that it is running on a PowerStation.

3.3.4. Upon pressing the GENERAL button, the SETTINGS menu disappears and the "General Settings" screen is displayed.

3.3.5. The display area of the General Settings screen shows the current configuration settings, including Power On Operation, whether or not to run MachineLogic at startup, the Download Port, and the IRQ settings for the serial ports.
3.3.6. There are two options for Power On Operation which are accessible by pressing the POWER ON OPERATION button as shown in the diagram below. Upon making a selection, the POWER ON OPERATION menu disappears and the selection is reflected in the display area.

3.3.7. The user may configure whether or not MachineLogic runtime is started at the time the system boots. If YES is chosen, then the Shell makes the appropriate modifications to the system startup files (i.e. autoexec.bat) to have MachineLogic runtime loaded and executed at boot time. If NO is chosen, the startup files are modified to ensure that MachineLogic is not run at boot time.

3.3.8. The user may also configure the primary Download Port. Upon pressing this button a menu appears for selecting COM1, COM2, or COM3. This selection informs the Shell which port to use for receiving Projects over the serial interface. It does not specify the port for MachineLogic development to connect to for online debug and communication. That is a setting that must be configured on the development machine, either by running a utility to reconfigure RTXDOS for the desired serial port, or by copying the appropriate version of RTXDOS to a runtime directory prior to downloading the Project.
3.3.9. The last setting on this screen allows the user to specify the IRQs assigned to each of the COM ports.

![Image of COM ports and Serial Port IRQs]

3.3.10. After making a COM port selection, a Numeric Keypad appears for the user to enter the proper IRQ number. Valid selections are the same as those provided in the existing Shell which are 2 through 15.

![Image of Numeric Keypad and PowerStation]

3.3.11. Numeric Keypad Operation

3.3.11.1. The number keys are used to enter numeric data for the selected option. The BACK button is used as a backspace to edit entries as they're being made. Once a numeric value is entered, the user can press the ENTER button to accept the value at which time the Numeric Keypad disappears. The ESC button can be pressed to abort the Numeric Keypad operation, which also causes the Numeric Keypad to disappear, however the original data value remains unchanged. *(Should the popup Numeric Keypad allow hex characters to be entered also? Some hardware configuration settings will require this...)*

3.3.11.2. Upon invoking the Numeric Keypad, the calling "control" should pass a list of the acceptable values, or a range. The user should not
be allowed to leave the Numeric Keypad with an invalid data value unless the ESC button is pressed.

3.3.12. Pressing the BACK button returns the user to the Main Screen.

3.3.13. The "PowerStation Settings" screen is displayed upon pressing the PWRSTA settings button.

3.3.14. The display area of the PowerStation Settings screen shows the current configuration settings, including Brightness level, Contrast level, and the Ethernet port address, IRQ assignment, IP address, and Host Name.

3.3.15. A menu appears as shown below if the user presses the CALIBRATE TOUCHSCRN button. This feature works the same as in the existing shell, where if the user presses YES, then the touch screen calibration utility is launched and run from DOS. If the user presses NO, then the menu merely goes away.
3.3.16. A menu appears as shown below when the user presses the KEYBOARD TEST button. This feature works the same as in the existing shell, where the user presses keys on the keyboard or front panel and they are shown on the display after the KEY PRESS: test. The user must press the F1 key twice to exit the keyboard test, upon which the menu disappears.

3.3.17. To adjust the display brightness and contrast, the user presses either the ADJUST BRIGHT or ADJUST CONTRAST buttons, and a menu appears as shown below for increasing or decreasing the setting, which can range from 1 to 10. As the value is adjusted, it is updated in the display area to reflect the new setting. The user can touch anywhere off the INCREASE / DECREASE menu area to cause it to go away.
3.3.18. The user can configure the ethernet port settings by pressing the ETHERNET PORT button. The desired option (PORT, IRQ, IP ADDRESS, or HOST NAME) can be selected from the menu for editing the setting.

3.3.19. The popup Numeric Keypad is used for entering the IRQ and IP ADDRESS. Note that a keyboard will be required to edit either the PORT setting (which is entered in hex -- or should the popup keypad support entering hex characters?) and the HOST NAME setting.

3.3.20. Pressing the BACK button returns the user to the Main Screen.

3.3.21. The INFO button presents a menu for viewing information about the system and software versions.
3.3.22. The SYSTEM button is used for accessing the System Information screen as shown below.

3.3.23. The System Information displays the Processor type, the amount of available ("free") conventional and extended memory, the logical drive assignments, and the port address assignments for all printer, serial, and ethernet ports in the runtime system.

3.3.24. The black opposing arrows in the upper right corner are used to indicate touch areas for scrolling the display area. Pressing the area on and above the up arrow causes the display to scroll upwards, and pressing the area on and below the down arrow causes the display to scroll downwards. Pressing and holding these areas should result in a continuous (smooth) scroll. The line connecting the arrows represents "no man's land" which is not an active touch area.
3.3.25. The VERSIONS button is used for accessing version information for the various software on the runtime system.

3.3.26. The Version Information screen displays the version numbers for the Interact Application Manager, Modules and Drivers. It is not necessary to display the entire EX-XXXXX-XXX number as done in the Interact development system. Only the version number part of the EX number need be displayed.

3.3.27. The Version Information screen also displays the version numbers for the MachineLogic software, including the runtime kernel, RTXDOS, and any other support software such as the Control Card TSR.

3.3.28. The Version Information screen also displays the version numbers for the system software such as DOS, BIOS, touchscreen driver, and the Shell itself.

3.3.29. The scroll arrows work as described in the System Information sections.
3.3.30. The UTILITIES button presents a menu for creating and restoring backups, accessing the security key, and running the Control Panel. The CONTROL PANEL button appears only if MachineLogic is included in the Project.

3.3.31. Selecting BACKUP PROJECT allows the user to go the Backup Project screen.

3.3.32. The user can select one of three pre-configured paths as the target location for performing the backup. This makes it easy to select a location for the backup without requiring the use of a keyboard to specify the path. A keyboard is needed only when configuring each of the paths in advance using the CONFIGURE PATHS button.
3.3.33. After pressing a USE PATH # button, the user is asked to verify the selection:

3.3.34. If NO is pressed, the menu is removed and the user may make another selection. If YES is pressed, then the backup begins with progress being displayed as shown below. A similar display is used for informing the user of errors during the backup, such as no disk present, write errors, etc.

3.3.35. The user may abort the backup operation at any time by pressing the CANCEL button.
3.3.36. After pressing the CONFIGURE PATHS button, a menu is provided for selecting the path number to configure.

3.3.37. After choosing a PATH # button, the user can enter a new path specification using the display shown below. The current path setting is displayed with a cursor at the end ready for editing.

3.3.38. The user must have a keyboard attached in order to edit a path setting. The INS, DEL, HOME, END and BS keys should be supported for editing the selection.

3.3.39. The path entry should be checked by the Shell to allow only characters that are valid for DOS path settings. The new path setting cannot be verified, however, until the user selects it to perform the actual backup since the device or logical location may not exist at the time the path is configured.
3.3.40. Pressing the RESTORE PROJECT button from the Utilities menu displays the Restore Project screen.

3.3.41. This screen works exactly the same as the Backup Project screen and uses the same path settings, except that a project is restored from (not backed up to) the specified path location.

3.3.42. Projects are backed up using a compression format yet to be determined, which will be the same as used by MachineShop. Investigations are currently underway to evaluate suitable compression methods. Once this has been decided, the Shell will incorporate the necessary compression and decompression routines. A backup generated by the Shell should be identical to a backup generated by MachineShop, and will be able to be restored by either one.

3.3.43. Pressing the SECURITY KEY button from the Utilities menu displays the Security Key screen.
3.3.44. The display area shows the serial and key number, the Interact modules and features that are enabled in the “Runtime Key”, which is always the first key found by the Shell or, in the case of P1, the modules that are enabled internally by default. The display area lists the module’s acronym followed it’s status which can be RUNTIME, CONFIG, DEVELOPMENT, or (TRANSFERRED).

3.3.45. If there are more modules than can be displayed on the screen at one time, then the scroll arrows and touch areas are displayed and active in the upper right corner.

3.3.46. The user can press the ENABLE MODULE button to enter the enable code for a module using the popup keypad.

3.3.47. After entering the 8 digit enable code, the Shell will verify it’s authenticity and enable the module in the Runtime Key, which will then be shown in the display area. If found to be invalid, an error message is displayed to inform the user.

3.3.48. The user can also attach a second key to the system and choose to transfer module enable information from or to the Runtime Key. After pressing the TRANSFER MODULE button, a menu is displayed for choosing which direction to transfer.
3.3.49. After choosing the transfer direction, a menu is displayed listing the modules on the key to be transferred from. The user selects the module and the transfer takes place, assuming that it is a valid selection. A user may not transfer a module to another key that already has that module enabled. The rules of transferring module information are specified in the existing key utility. An error message is displayed for the user if there are any problems with the selections or the transfer. Otherwise, the display area is updated to indicate the results of the transfer.

3.3.50. If there are more modules than can fit on one menu, it proposed that the last button be labeled NEXT PAGE which can be used to display the remaining selections.

3.3.51. Pressing the CONTROL PANEL button from the Utilities menu displays the MachineLogic Control Panel screen.
3.3.52. The display area shows the current mode of the MachineLogic runtime, which can be STOP, RUN, or HALT, and it shows the current Error # and message. The message is displayed below the ERROR #: line if an error condition exists. Also displayed is the Task Information (contents yet to be determined).

3.3.53. The user can choose to stop the control logic from executing by pressing the STOP button. A menu appears for confirmation, and the display area is updated to reflect the selection.

3.3.54. The user can also choose to start the control logic by pressing the START button. A menu appears with options for selecting the start mode: COLD, WARM, or HOT. After making a selection, the control logic is started and the display area is updated to show that the logic is running.
3.3.55. If the Hardware Watchdog trips, the RESET WATCHDOG button alternates (flashes) colors from default to red at a half second rate, and an error message is displayed. The user may reset the watchdog by pressing the RESET WATCHDOG button.

![MachineLogic Control Panel](image1)

3.3.56. The HARDWARE SETUP button is used to access the MachineLogic Hardware Setup screen.

![MachineLogic Hardware Setup](image2)

3.3.57. The display area lists the hardware configuration options which include the Retentive Memory Size, Memory Base Address, Port Base Address, Interrupt Request Number, Power Fail, and Watchdog settings.
3.3.58. Pressing the RETENTIVE MEMORY button presents a menu for choosing to configure either the size or the base address.

3.3.59. Choosing the MEMORY SIZE button causes the keypad to popup for entering a new value which can be 0, 4, 8, 16 or 32 (KB). Editing the MEMORY ADDRESS will require a keyboard (unless the keypad supports entry of hex characters).

3.3.60. Editing the PORT ADDRESS will also require a keyboard (unless the keypad supports entry of hex characters). The INTERRUPT REQUEST number may be entered using the popup keypad. Valid settings are 3, 4, 5, 7, 10, 11, 12, or 15 (as specified in the .ini file).

3.3.61. Pressing either the POWER FAIL or WATCHDOG buttons will display a menu to ENABLE or DISABLE the feature.

3.3.62. The changes made to the Hardware Settings are written to the .ini file for the hardware card. The changes do not actually take effect, however, until the next time the runtime system boots and the support TSR is loaded.
3.3.63. The user may choose to exit the Shell by pressing the EXIT button on the Main Menu. A menu is displayed for confirmation.

3.3.64. Help may be accessed from any screen by pressing the HELP button. A help page is displayed containing information for the screen currently being viewed. The scroll arrows and touch areas become active if the help text is more than what can be displayed on a single page.
Appendix F

MachineShop Shell Help
(presented in original text format)
The PowerStation Shell is a graphical, menu-driven interface that communicates with the PowerStation and its installed components (hardware and software). It displays detailed information about the system and allows you to configure many system hardware and software settings.

**POWERSTATION SHELL MAIN MENU**

- **RUN** Determine the applications that can be executed directly from the Shell.
- **SETTINGS** View and edit various configurable Shell and PowerStation options.
- **INFO** Access information about PowerStation hardware and Interact/MachineLogic software installed on the unit.
- **UTILITIES** Manage projects and configure various hardware and software components, including the Interact Security Key, and MachineLogic Control Adapter.
- **LANGUAGE** Select the language used by the Shell to display screen text and online files.
- **EXIT** Leave the Shell to work in another application or environment.

**POWERSTATION SHELL NAVIGATION**

Navigate within the PowerStation Shell by using the touchscreen and pressing on a button or selection area to activate the selection. If a keyboard is attached, you can press the highlighted letter of the selection on the keyboard. For example, press **Help** to activate Help.

Screens that are longer than one page will display a set of small up and down arrows in the top right section of the Shell. These arrows access additional screens and scroll to information that cannot fit inside the standard display area. To use the scrolling arrows, select the area on the touchscreen that surrounds the arrow.

To return to a previous menu or screen after making a selection, press the **BACK** button that displays below the Help button or press **BACK** on a keyboard.
To exit or cancel selection, press the touchscreen on an area outside of the displayed menu, click the right mouse button, or press the [ESC] key on the keyboard.

Menu options that require the input of numerical or hex values will display an onscreen pop-up numeric keypad. To use the keypad, enter values by selecting the appropriate number on the touchscreen. Press the Enter key to finish the selection.

\bPOWERSTATION SHELL HELP\b

To obtain Help for a displayed menu or submenu, select the Help button located above the main menu or press \b\uH\u\b on a keyboard. This will display a Help screen in the display area with additional information. Underlined characters in Help screens represent keyboard shortcuts for menu and submenu selections, but do not act as actual links in the Help screens.

To exit a Help screen, press the touchscreen on an area outside of the help screen, click the right mouse button, or press the [ESC] key on the keyboard.

Refer to the PowerStation user manual for more specific and complete help information about Shell components.


</TEXT>

<TEXT ID="1010" TITLE="SHELL RUN">
Executes the Interact application from the Shell. It provides the proper environment for Interact to create and use the applications within the currently selected project.

When the Interact editing/runtime session has completed, you will be returned to the Shell main menu.
</TEXT>

<TEXT ID="1020" TITLE="SHELL SETTINGS">
View and edit various Shell and PowerStation configuration options. Displays a list of current settings in the display area and a menu of configurable options, including General and PowerStation.

\b\uG\uGENERAL\b Displays current configuration settings and loads menu of configurable options including: Power On Operation (applications that launch on start up), Machine Logic (enables the control adapter), Download Port (select the COM port for downloading), and Serial IRQ (specify the IRQs assigned to the COM ports).

\b\uP\uPOWERSTATION\b Displays current PowerStation settings and loads menu of configurable functions including: touchscreen calibration, keyboard test, screen brightness and contrast, and ethernet port configuration. These PowerStation settings will be visible only if the Shell is currently running on a PowerStation.
</TEXT>

<TEXT ID="1100" TITLE="GENERAL SETTINGS">
Displays the current General Settings in the display area and a main menu to modify general PowerStation settings and configurations.
\b\uP\uOWER ON OPERATION\b Determines applications that launch on start up.

\b\uM\uACHINE LOGIC\b Enables the MachineLogic Control Adapter on start up.

\b\uD\uOWNLOAD PORT\b Selects the COM port used for downloading projects.

\b\uS\uSERIAL IRQ\b Specifies the IRQ assigned to the PowerStation COM ports.

\b\uP\uOWER ON OPERATION\b

Specifies which application will run after the PowerStation is turned on. You can choose Interact or the PowerStation Shell.

\b\uR\uUN \uI\uNTERACT\b Loads Interact when the unit is turned on. When you exit Interact, the Shell is displayed. Useful if a power cycle should occur because the unit would automatically run Interact when the power is restored.

\b\uR\uUN \uS\uHELL\b loads the Shell and displays the Shell main menu.

Power On initialization applications can be changed during any Shell session, but changes will not take effect until the unit reboots.

\b\uU\uY\uES\b Directs the Shell to modify the system startup files (i.e. autoexec.bat) to load and execute MachineLogic when the PowerStation is turned on.

\b\uU\uN\uO\b Directs the Shell to modify the system startup files (i.e. autoexec.bat) to NOT load and execute MachineLogic when the PowerStation is turned on.

\b\uP\uOWER ON OPERATION\b

Enables MachineLogic when the PowerStation is turned on.

Choose which serial port (\bCOM \u1\b, \bCOM \u2\b, \bCOM \u3\b) is used for downloading projects to the PowerStation.

When a serial port is enabled for download, the Shell takes control of the port to monitor the start of a project download; this takeover may interfere with TSRs or network drivers that use the serial port. To allow other software drivers to use a selected serial port, that port should be disabled for download.

\g7NOTE: This setting does not specify the COM port for MachineLogic Development to connect to for online debugging or communication. The Download Port setting must be configured on the development machine, either by running a utility to reconfigure RTX DOS for the desired serial port, or by copying the appropriate version of RTX DOS to a runtime directory prior to downloading a project.
\g7NOTE: If the Shell detects a serial port in use by a touchscreen or external mouse, that port will be automatically disabled for downloading and cannot be enabled until the port is no longer in use.

Used to specify and match the IRQ numbers assigned to each of the serial ports (COM1, COM2, COM3) on the PowerStation with the hardware IRQ jumper numbers on the serial board. The IRQ numbers in the Shell must match the hardware jumper settings for proper operation of the serial ports.

The COM1 serial port usually uses IRQ4 and COM2 uses IRQ3. There are no standard IRQ assignments for COM3 and COM4, these settings must be determined by examining the system hardware configuration.

For instructions on how to use the pop-up keypad, press the Help button.

Displays current PowerStation Settings in the display area and configurable functions in the main menu.

\b\uC\uALIBRATE TOUCHSCRN\b Launches the touchscreen utility.

\b\uK\uEYBOARD TEST\b Displays keyboard test screen to test keyboard.

\bADJUST B\uR\uIGHT\b Manually adjust touchscreen brightness.

\bADJUST CO\uN\uTRAST\b Manually adjust touchscreen contrast.

\g7NOTE: A keyboard is required to calibrate the touchscreen or perform a keyboard test.

Launches the touchscreen calibration utility. You will be asked to confirm your selection before the calibration utility is actually launched.

\g7NOTE: A keyboard is required to configure the touchscreen.

Refer to the PowerStation User Guide for more information about calibrating the touchscreen.

Displays the Keyboard Test screen to test a keyboard connected to the PowerStation. You will be asked to confirm your selection before the keyboard can be tested.

\g7NOTE: A keyboard is required to perform this test.

To use the KEYBOARD TEST, press a key on the keyboard and check the test screen to confirm that the display results match the pressed key.
Pressing the touchscreen will display the "SELECT" textstring. The CTRL, ALT, and SHIFT keys will not display on the test screen when pressed on the keyboard.

To exit the KEYBOARD TEST, press the [F1] function key twice.
</TEXT>

+P5 only

<TXT ID="XXXX" TITLE="KEYBOARD TEST"> Displays Keyboard Test options for the P5 PowerStation. Also allows you to perform tests on the use and the configuration of the SHIFT key.

\g7NOTE: A keyboard is required to perform this test. </TXT>

+P5 only

<TXT ID="XXXX" TITLE="P5 KEYBOARD TEST"> Displays the Keyboard Test screen to test a keyboard connected to the PowerStation. You will be asked to confirm your selection before the keyboard can be tested.

\uShift Key xx\u Displays the Keyboard Test Shift xx screen to test the SHIFT key on the keyboard. The PowerStation physically supports twenty function keys on the front of the unit labeled F1-F20. The SHIFT key provides access to a second set of twenty keys (F21-F40).

\uShift Key Config\u Displays the Keyboard Test Shift Key Config screen. Use Shift Key Config to select the mode for the Shift key that appears on the faceplate of P5 PowerStations. The three modes of operation for the SHIFT key are Momentary, Lock, and Lock/Release modes. An indicator on the Shift key is lit whenever the Shift key is activated or in the locked mode.

\g7NOTE: Operating a Message Input tool or a pop-up numeric keypad tool when the faceplate Shift mode is on, will input values of 2, 4, 6, and 8 for the Down, Left, Up, and Right arrow keys, respectively. To avoid entering these values when you want to use the arrow keys to control the cursor, make sure the Shift mode is off.

To exit the Keyboard Test, press the [F1] function key twice.
</TXT>

+not on P1

<TXT ID="XXXX" TITLE="ADJUST BRIGHTNESS"> Use to manually adjust the brightness of the PowerStation touchscreen display. You can either Decrease or Increase the screen brightness.
</TXT>

+not on P1

<TXT ID="XXXX" TITLE="ADJUST CONTRAST"> Use to manually adjust the contrast of the PowerStation touchscreen display. You can either Decrease or Increase the screen contrast.
</TXT>

<TXT ID="1030" TITLE="SHELL INFO"> Access information about PowerStation hardware and Interact/MachineLogic software installed on the unit. Displays a
submenu with the choice of viewing PowerStation System information or installed software information.

\b\SYSTEM\b Displays information about the Processor type, the amount of available conventional and extended memory, the logical drive assignment, and the port address assignments for all printer, serial, and ethernet ports in the runtime system.

\b\VERSIONS\b Displays version numbers for the Interact Application Manager, Modules, and Drivers. Also displays the version numbers for MachineLogic, support software, and system software (including DOS, BIOS, touchscreen driver, and the Shell).

Information displayed from this menu is for viewing purposes only and cannot be edited from this menu.

\g7NOTE: Interact version numbers will not display the entire EX-XXXXX-XXX number as done in the Interact development system. Only the version number part of the EX number will display.

Information displayed from this menu is for viewing purposes only and cannot be edited from this menu.

\b\BACKUP PROJECT\b Archives and backs up the loaded project files using a selected path (Path 1, 2, or 3).

\b\RESTORE PROJECT\b Decompresses and loads projects files to the PowerStation from using a selected path (Path 1, 2, or 3).

\b\SECURITY KEY\b Accesses utilities used to enable or transfer Interact modules on the Security Key.

\b\CONTROL PANEL\b Displays MachineLogic settings and allows configuration of the MachineLogic Control Adapter.
\g7NOTE: There is only one set of paths (Path 1, 2, and 3) used to backup and restore PowerStation projects. The Backup Paths used to backup project files are the same paths used to restore projects, but the paths may be configured from either utility.

\g7NOTE: The Control Panel utility will not be available on the submenu if MachineLogic is not included in the loaded Project.
</TEXT>

<TEXT ID="1500" TITLE="BACKUP PROJECT UTILITY">
Displays the current project backup paths in the display area and a main menu of Backup Project selection buttons (\bPATH \u1\u\b, \bPATH \u2\u\b, \bPATH \u3\u\b). You can configure or edit project backup paths from the main menu.

Project file backups archive the entire project, not just an individual file. Backups are automatically compressed using the zip compression method. When a project file is restored, the compressed project is automatically decompressed.

A verification submenu will display to confirm the selection before the backup occurs.

\g7NOTE: Before a backup can occur, the target source must be connected to the PowerStation or a transfer error will occur and you will need to connect the source or perform another backup.
</TEXT>

<TEXT ID="XXXX" TITLE="BACKUP PROJECT">
Displays the current project name and backup path in the display area. Initiates a project backup using the displayed backup path. A verification submenu will display to confirm the selection before the backup occurs.

\uYES\u Begins backup of the displayed project to the target source. During backup, a status screen will display the project files being backed up and the percentage of the backup completed.

\uCANCEL\u Aborts the backup operation.

\uNO\u Clears the path submenu display.

\g7NOTE: The target source must be connected to the PowerStation before performing a backup.

If an error occurs during backup...
</TEXT>

<TEXT ID="XXXX" TITLE="BACKUP PROJECT">
Displays the current project name and backup path in the display area. Initiates a project backup using the displayed backup path. A verification submenu will display to confirm the selection before the backup occurs.

\uYES\u Begins backup of the displayed project to the target source. During backup, a status screen will display the project files being backed up and the percentage of the backup completed.
\uCANCEL\u Aborts the backup operation.
\uNO\u Clears the path submenu display.

\g7NOTE: The target source must be connected to the PowerStation before performing a backup.

If an error occurs during backup..</TEXT>

<TEXT ID="XXXX" TITLE="BACKUP PROJECT">
Displays the current project name and backup path in the display area. Initiates a project backup using the displayed backup path. A verification submenu will display to confirm the selection before the backup occurs.

\uYES\u Begins backup of the displayed project to the target source. During backup, a status screen will display the project files being backed up and the percentage of the backup completed.

\uCANCEL\u Aborts the backup operation.
\uNO\u Clears the path submenu display.

\g7NOTE: The target source must be connected to the PowerStation before performing a backup.

If an error occurs during backup..</TEXT>

<TEXT ID="1560" TITLE="BACKUP PROJECT - CONFIGURE PATHS">
Displays a submenu of backup path selection buttons (\bPATH \u1\u\b, \bPATH \u2\u\b, \bPATH \u3\u\b). Selecting a path displays an entry screen in the display area that allows you to enter the backup path for the selected button.

\g7NOTE: A keyboard is used to input or edit a path setting. Only valid DOS characters can be used for a path. [INS], [DEL], [HOME], [END], and [BACKSPACE] keys are supported.

\g7NOTE: A backup path cannot be verified until an actual backup is performed. If the destination source is invalid or not available for backup, an error message will display.

</TEXT>

<TEXT ID="XXXX" TITLE="BACKUP PROJECT - CONFIGURE PATH">
Displays an entry screen in the display area which allows you to input or edit the backup path for PowerStation project files.

\uOK\u Confirms the backup path configuration you enter.

\uCANCEL\u Invalidates the selection and leaves the current backup path untouched.
A keyboard is used to input or edit a path setting. Only valid DOS characters can be used for a path. [INS], [DEL], [HOME], [END], and [BACKSPACE] keys are supported.

A backup path cannot be verified until an actual backup is performed. If the destination source is invalid or not available for backup, an error message will display.

Displays an entry screen in the display area which allows you to input or edit the backup path for PowerStation project files.

Confirms the backup path configuration you enter.

Invalidates the selection and leaves the current backup path untouched.

A backup path cannot be verified until an actual backup is performed. If the destination source is invalid or not available for backup, an error message will display.

Displays an entry screen in the display area which allows you to input or edit the backup path for PowerStation project files.

Confirms the backup path configuration you enter.

Invalidates the selection and leaves the current backup path untouched.

A keyboard is used to input or edit a path setting. Only valid DOS characters can be used for a path. [INS], [DEL], [HOME], [END], and [BACKSPACE] keys are supported.

A backup path cannot be verified until an actual backup is performed. If the destination source is invalid or not available for backup, an error message will display.

Displays the current project restoration paths in the display area and a main menu of Restore Project selection buttons (PATH 1, PATH 2, PATH 3). You can configure or edit project restoration paths from the main menu.

The paths used to restore projects from this main menu are the same as those used to backup projects. When a project file is restored, the archived backup project is automatically decompressed.

A verification submenu will display to confirm the selection before the restoration occurs.
NOTE: Before a restoration can occur, the target source must be connected to the PowerStation or a transfer error will occur and you will need to connect the source or perform another backup.

<TEXT ID="XXXX" TITLE="RESTORE PROJECT">
Displays the current Project name and restore target source in the display area. Initiates a project restoration using the restore path, but a verification submenu will display before the restoration occurs.

YES Begins restoration of the project using the displayed path. During restoration, a status screen will display the project files being loaded and the percentage of the restoration completed.

CANCEL Immediately aborts the project restoration.

NO Clears the path submenu display.

NOTE: The target source must be connected to the PowerStation before restoring a project.

If an error occurs during backup..
</TEXT>

<TEXT ID="XXXX" TITLE="RESTORE PROJECT">
Displays the current Project name and restore target source in the display area. Initiates a project restoration using the restore path, but a verification submenu will display before the restoration occurs.

YES Begins restoration of the project using the displayed path. During restoration, a status screen will display the project files being loaded and the percentage of the restoration completed.

CANCEL Immediately aborts the project restoration.

NO Clears the path submenu display.

NOTE: The target source must be connected to the PowerStation before restoring a project.

If an error occurs during backup..
</TEXT>

<TEXT ID="XXXX" TITLE="RESTORE PROJECT">
Displays the current Project name and restore target source in the display area. Initiates a project restoration using the restore path, but a verification submenu will display before the restoration occurs.

YES Begins restoration of the project using the displayed path. During restoration, a status screen will display the project files being loaded and the percentage of the restoration completed.

CANCEL Immediately aborts the project restoration.

NO Clears the path submenu display.
NOTE: The target source must be connected to the PowerStation before restoring a project.

If an error occurs during backup.
</TEXT>

<TEXT ID="1660" TITLE="RESTORE PROJECT - CONFIGURE PATHS">
Displays a submenu of restore path selection buttons (\bPATH \u1\u\b, \bPATH \u2\u\b, \bPATH \u3\u\b). Selecting a path displays an entry screen in the display area that allows you to enter the restore path for the selected button. The paths used to restore projects from this main menu are the same as those that are configured using the Backup Project utility.

NOTE: A keyboard is used to input or edit a path setting. Only valid DOS characters can be used for a path. [INS], [DEL], [HOME], [END], and [BACKSPACE] keys are supported.

NOTE: A restore path cannot be verified until an actual project is restored. If the project source is invalid or not available, an error message will display.
</TEXT>

<TEXT ID="XXXX" TITLE="RESTORE PROJECT - CONFIGURE PATH">
Displays an entry screen in the display area, which allows you to input or edit the restore path for PowerStation project files. The paths used to restore projects from this main menu are the same as those that are configured using the Backup Project utility.

OK\u Confirms the backup path configuration you enter.

CANCEL\u Invalidates the selection and leaves the current restore Path untouched.

NOTE: A keyboard is used to input or edit a path setting. Only valid DOS characters can be used for a path. [INS], [DEL], [HOME], [END], and [BACKSPACE] keys are supported.

NOTE: A restore path cannot be verified until a project is restored. If the project source is invalid or not available for backup, an error message will display.
</TEXT>

<TEXT ID="XXXX" TITLE="RESTORE PROJECT - CONFIGURE PATH">
Displays an entry screen in the display area, which allows you to input or edit the restore path for PowerStation project files. The paths used to restore projects from this main menu are the same as those that are configured using the Backup Project utility.

OK\u Confirms the backup path configuration you enter.

CANCEL\u Invalidates the selection and leaves the current restore Path untouched.

NOTE: A keyboard is used to input or edit a path setting. Only valid DOS characters can be used for a path. [INS], [DEL], [HOME], [END], and [BACKSPACE] keys are supported.
\g7NOTE: A restore path cannot be verified until a project is restored. If the project source is invalid or not available for backup, an error message will display.

</TEXT>

<TEXT ID="XXXX" TITLE="RESTORE PROJECT - CONFIGURE PATH">
Displays an entry screen in the display area, which allows you to input or edit the restore path for PowerStation project files. The paths used to restore projects from this main menu are the same as those that are configured using the Backup Project utility.

\uOK\u Confirms the backup path configuration you enter.

\uCANCEL\u Invalidates the selection and leaves the current restore Path untouched.

\g7NOTE: A keyboard is used to input or edit a path setting. Only valid DOS characters can be used for a path. [INS], [DEL], [HOME], [END], and [BACKSPACE] keys are supported.

\g7NOTE: A restore path cannot be verified until a project is restored. If the project source is invalid or not available for backup, an error message will display.

</TEXT>

+not on P1

<TEXT ID="1700" TITLE="SECURITY KEY UTILITY">
Displays Interact security key information (Serial Number and Key Number) and installed module information (a listing of enabled modules) in the display area, including the status of enabled modules (Runtime, Config, Development, or Transferred).

\b\uE\uNABLE MODULE\b Used to enter an enable code for an Interact module. Displays a numeric pop-up menu to enter the eight-digit enable code. If the code is valid, the module will be enabled and the status will be updated in the display area. If the enable code is invalid, an error message will display.

\b\uT\uRANSFER MODULE\b Used to transfer module enable information from one security to another. Displays a Transfer submenu to choose which direction to transfer the chosen module. After the direction button is selected, a menu will display the available modules to transfer. Selecting a listed module to begin the transfer. You may not transfer a module to another key that has the same module enabled. If there is a problem with the module transfer, an error message will display.

</TEXT>

<TEXT ID="1720" TITLE="SECURITY KEY - ENABLE MODULE">
Displays a numeric pop-up menu to enter the eight-digit enable code for an Interact module. If the code is valid, the module will be enabled and the status will be updated in the display area. If the enable code is invalid, an error message will display.

</TEXT>

<TEXT ID="1730" TITLE="SECURITY KEY - TRANSFER MODULE">

</TEXT>
Displays a Transfer submenu to choose which direction to transfer the chosen Interact module. After the direction button is selected, a menu will display the available modules to transfer. Selecting a listed module to begin the transfer. You may not transfer a module to another key that has the same module enabled. If there is a problem with the module transfer, an error message will display.

`b\uF\uROM RUNTIME\b` Transfers an enabled module from the PowerStation key to a second attached security key.

`b\uT\uO RUNTIME\b` Transfers an enabled module from the second key to the first security key attached to the PowerStation.

After selecting the direction button, a submenu screen will list the available modules to transfer. Select a listed module to begin the transfer between keys.

You may not transfer a module to another key that has the same module enabled. If there are any problems with the module selections or transfer, an error message will display.

</TEXT>

<TEXT ID="1800" TITLE="CONTROL PANEL UTILITY">
Displays current MachineLogic settings including the current mode, any error information, and task information. Also loads MachineLogic main menu with configurable options for the MachineLogic Control Adapter.

`b\uS\uTOP\b` Stops the Control Adapter from executing. A verification submenu appears to confirm the selection and the control logic mode will be updated in the display area to reflect the new status.

`b\S\uT\uART\b` Displays system status options for starting the Control Adapter. Status options include Cold, Warm, and Hot.

`b\uR\uESET WATCHDOG\b` Resets the MachineLogic Hardware Watchdog alarm if tripped. The button flashes in red and an error message is displayed in the display area if the alarm is tripped.

`b\uH\uARWARE SETUP\b` Accesses features of the Control Adapter to configure. Displays the current MachineLogic Control Adapter settings and a main menu that allows you to configure or edit certain settings. Menu options include Retentive Memory, Port Base Address, Interrupt Request Number, Power Fail, and Watchdog.

</TEXT>

<TEXT ID="1820" TITLE="CONTROL PANEL - STOP">
Changes the Control Panel mode and stops the MachineLogic Control Adapter from executing. Displays a verification submenu to confirm the selection.

`b\uY\uES\b` Directs the Shell to stop executing the MachineLogic Control Adapter.

`b\uN\uO\b` Directs the Shell to continue executing the Control Adapter and to return to the MachineLogic Main menu.
</TEXT>
Select this main menu button to choose the Control Panel start mode for the MachineLogic Control Adapter. Selecting this button will display a submenu with Control Adapter start mode options (Cold, Warm, and Hot). Differences between the start modes are in how the Memory Area and Retentive Memory Areas of the Control Adapter are handled.

\g7NOTE: Only the Cold start mode option is available all the time, the Warm and Hot options are available under certain conditions. Pressing Stop during the control process will make all three options available.

\b\uC\uOLD\b Resets ALL MachineLogic memory.
\b\uW\uARM\b Resets the Memory Area (NOT the Retentive Memory Area) of MachineLogic.
\b\uH\uOT\b Keeps all of the memory settings of MachineLogic. Not available if you download a new control application or have just started the control system.

Flashes in red if the MachineLogic Control Adapter Watchdog timer is tripped and displays an error number with a corresponding message in the display area. Press button if flashing to reset the Watchdog timer.

The Watchdog timer on the Control Adapter must be enabled for this feature to work. Watchdog timer is configured by selecting the Watchdog button on the Hardware Settings main menu.

Displays the current Control Adapter settings in the display area and a main menu with configuration buttons. Control Adapter hardware options include the Retentive Memory (Address and Size), the Memory Base Address, the Port Base Address, the hardware interrupt request, the Watchdog and Power Fail status, and the UCS communications card settings (Base Address and IRQ).

\g7NOTE: Changes made to Hardware Settings are written to the MLCA.INI file for the MachineLogic Control Adapter. These changes do not take effect until the next runtime system boot and loading of support TSR. A confirmation message will display after any changes asking if you wish to reboot the system.

Configures the Retentive Memory Size and the Memory Address for the MachineLogic Control Adapter.

\bMEMORY \uS\uIZE\b Defines the NVSRAM in kilobytes from 4k to 32k, incrementing in values 4k, 8k, 16k, and 32k. A value of 0 will disable the NVSRAM. The default value of the Retentive Memory Size in the MLCA.INI file is: MemSize=4.

\bMEMORY \uA\uADDRESS\b Defines the base memory segment for the non-volatile shared memory (NVSRAM). The base memory segment must be
defined at an address that is evenly divisible by the memory size and allows the entire memory window to fit within certain ranges based on the model of PowerStation. The default value of the Retentive Memory Address in the MLCA.INI file is: MemBaseSeg=0xD800.

Configures the I/O Port Address for the MachineLogic Control Adapter. The Port Address defines the I/O port address parameter, ranging from 200 to FFF8 in multiples of eight.

The default value of the Port Address in the MLCA.INI file is: IOBase=0x710.

Configures the hardware interrupt for the MachineLogic Control Adapter. The Interrupt Request defines the assigned IRQ on the Control Adapter. Valid IRQ numbers are 3, 4, 5, 7, 10, 11, 12, or 15 (as long as the number does not conflict with any other PC IRQ assignments). Set the IRQ to zero (0) if Powerfail and Watchdog functions are disabled on the Adapter Card.

The default value of the Interrupt Request in the MLCA.INI file is: IRQ=5.

Configures the I/O port address for the UCS interface card, if installed on the MachineLogic Control Adapter. The UCS Base Address defines the I/O port address parameter, ranging from 200 to FFF8 in multiples of eight.

The default value of the UCS Base Address in the MLCA.INI file is: UCSBase=0x718.

Configures the hardware interrupt for the UCS interface card, if installed on the MachineLogic Control Adapter. The UCS IRQ defines the assigned IRQ on the UCS card. Valid IRQ numbers are 3, 4, 5, 7, 10, 11, 12, or 15 (as long as the number does not conflict with any other PC IRQ assignments). Set the IRQ to zero (0) if no interrupt is used.

The default value of the UCS Base Address in the MLCA.INI file is: UCSIrq=0.

Configures the Power Fail feature of the MachineLogic Control Adapter. \b\uE\uDISABLE\b Changes the hardware setting and activates Power Fail. \b\uD\uDISABLE\b Changes the hardware setting and deactivates Power Fail.
Configures activation of the Watchdog timer feature of the MachineLogic Control Adapter.

\b\uE\uN\uABLE\b Changes the hardware setting and activates the Watchdog timer.

\b\uD\uIS\uABLE\b Changes the hardware setting and deactivates the Watchdog timer.

Displays a submenu of available languages used by the Shell to display screen text and online files. Language submenu options depend on the installed language files detected by the Shell.

\g7\NOTE: Only languages with language files detected by the Shell will display in the submenu.

Selecting a submenu button directs the Shell to load the corresponding language file. After loading the file, all screen text and online files will display in the selected language.

Displays verification submenu confirming the selection to exit the Shell.

To cancel the EXIT selection, press \b\uN\uO\b or select an area outside of the submenu.

To return to the Shell after exiting, type \b\ushell\b\u at the DOS prompt.
Appendix G

Sample of Supervisor Markup from
Shell Help First Draft
POWERSTATION SHELL

Copyright (c) 1999
CTC Parker Automation
All Rights Reserved

The PowerStation Shell is organized into five major components.

The first component is the Shell Title Area, an open line across the top of the screen that can display a project name or company logo. The second Shell component is the HELP display that is located on the left side of the screen along the separator bar. The third component is the Main Menu, a set of six vertical-aligned buttons located along the left side of the screen below the HELP display. The fourth component is the Submenu, a group of horizontal buttons associated with a vertical button. The final component of the PowerStation Shell is the Display Area on the bottom right section of the screen. The Display Area displays various information pertaining to the Menu and Submenu options.

To change the Title Area graphic, XXXX XXXX XXXXXXXX

To place or change a background image that surrounds the Shell (P2 and higher), XXXX XXXXXXXXXX XXXXX

To change the color of the Shell separator bar, XXXXXXXXX XXXXXX XXXXX

To change the color of the Shell main menu buttons, XXXXXXXXX XXXXXXXXXX XXXXXX

<POWERSTATION SHELL NAVIGATION

You can navigate the PowerStation Shell by using a mouse, keyboard, or by tapping on the touchscreen.

Using a mouse, move the cursor on the screen to a selection area or button, and click the mouse button to activate the selection.

Using a keyboard, press the arrow keys or [TAB] key to move the cursor to a selection area, and press [ENTER] to activate the selection. Pressing the [TAB] key cycles the cursor forward thru the screen and [TAB]+[SHIFT] cycles the cursor backward.

Using the touchscreen, tap the touchscreen on a selection to select and activate the selection.

Display area: Screens that are longer than one page will activate and display a set of up and down arrows in the top right section of the Shell on the separator bar. These arrows access screens that display additional information that cannot fit inside the standard display area. To activate these arrows, move the cursor over an arrow and click the mouse button. Press the [PAGE UP] or [PAGE DOWN] buttons on the keyboard, or tap the arrow section on the touchscreen.

To return to a previous menu or submenu, tap the BACK button that displays on the top of the main menu.

?? Keyboard or mouse options.
To exit a menu, click the right mouse button, press the [ESC] key on the keyboard, or tap the touchscreen on an area outside of the displayed menu.

Menu options that require user input of numerical values will display an embedded pop-up numeric keypad. To use this keypad, move the cursor over a number or function and press the mouse button, press the corresponding key on the keyboard, or tap the number or function on the touchscreen.

<POWERSTATION SHELL HELP>

To obtain context-sensitive help on a selection or display screen, move the cursor over the desired item and press the [F1] function key, move the cursor over the HELP menu option and click the mouse button, or tap the HELP section of the Shell menu.

To obtain help on a screen or menu that is displayed on the touchscreen, press the [F1] function key or tap the HELP section of the Shell menu.

Help information that is longer than one page will display a set of up and down arrows in the top right portion of the screen on the separator bar. To activate the arrows, move the cursor over an arrow and click the mouse button, press the PAGE UP or PAGE DOWN buttons on the keyboard, or tap the arrow region on the touchscreen.

To exit a help screen, click the right mouse button, press the [ESC] key on the keyboard, or tap the touchscreen on an area outside of the help menu or screen.

%RUN
< SHELL RUN

Use this selection to choose an application to launch directly from the PowerStation Shell. Display a submenu of available applications to launch. Up to xx applications can be launched from the Shell. Applications include: INTERACT, MS-DOS, xx

To launch an application, move the cursor over the desired button and activate the selection by clicking the mouse button, pressing [ENTER], or tapping the button on the touchscreen.

To add an application to the Shell RUN menu, xxxxxxxxxx xxxxxx xxxxxxxxxx

To delete an application from the Shell RUN menu, xxxxxxx xxxxxxxxxx xxxxxx xxxxx

%RUN-INTERACT
< SHELL RUN INTERACT

Launches the Interact application. To launch Interact, move the cursor over the button and activate the selection by clicking the mouse button, pressing [ENTER], or tapping the button on the touchscreen.

To change the application, xxxxxxx xxxxxxxxxx xxxxxx xxxxx
To delete the application, xxxxxxx x x x x x x x x x x

%SETTINGS
<SHLL SETTINGS

Use this selection to view or edit various Shell and PowerStation options.
Displays a submenu of configuration options for the CPU and PowerStation (if the
PowerStation is running the Shell).

Options include: GENERAL, PWRSRTA, xx

NOTE: PWRSRTA settings will not display if the Shell is not running on a
PowerStation unit.

%SETTINGS-GENERAL
<GENERAL SETTINGS

Use this selection to manually adjust general PowerStation settings and
configurations. Displays current GENERAL SETTINGs in the display area and load
the GENERAL SETTINGs main menu. Additional selectitin that can not be modified
may or may not default.

Options include: POWER ON OPERATION, MACHINE LOGIC, DOWNLOAD PORT, and
SERIAL IRQ.

%SETTINGS-GENERAL-POWER ON OPERATION
<GENERAL SETTINGS - POWER ON OPERATION

Use this selection to choose the applications that launch when the
PowerStation is turned on. Displays submenu with application options.

Options include: RUN INTERACT, RUN SHELL, xx (You can select with the
arrow keys and the select key).

To add an application to the POWER ON OPERATION submenu, xxx x x x x x x x x

To delete an application from the POWER ON OPERATION submenu, xxx x x x x x x x x

%SETTINGS-GENERAL-POWER ON OPERATION-RUN INTERACT
<POWER ON OPERATION - RUN INTERACT

Launches the application when the PowerStation is turned on.

To edit RUN INTERACT, xxx x x x x x x x x

To delete RUN INTERACT from the POWER ON OPERATION submenu, xxx x x x x x x x x

%SETTINGS-GENERAL-POWER ON OPERATION-RUN SHELL
<POWER ON OPERATION - RUN SHELL

Launches the PowerStation Shell when the PowerStation is turned on.

see above
To edit RUN SHELL, xxxx xxxx xxxx xxxx

To delete RUN SHELL from the POWER ON OPERATION submenu, xxxx xxxx xxxx xxxx

%SETTINGS-GENERAL-MACHINE LOGIC
<GENERAL SETTINGS - MACHINE LOGIC

Use this selection to launch MachineLogic Runtime when the PowerStation is turned on. Displays option submenu asking if you wish to enable this feature.

Options include: YES or NO

YES directs the Shell to modify the system startup files (i.e. autoexec.bat) to load and execute MachineLogic when the PowerStation is turned on.

NO directs the Shell to modify the system startup files (i.e. autoexec.bat) to NOT load and execute MachineLogic when the PowerStation is turned on.

%SETTINGS-GENERAL-DOWNLOAD PORT
<GENERAL SETTINGS - DOWNLOAD PORT

Use this selection to choose which COM port to use for receiving projects over the serial interface. Displays submenu of DOWNLOAD PORT options.

Options include: COM1, COM2, COM3

NOTE: This setting does not specify the COM port for MachineLogic Development to connect to for online debugging or communication. The DOWNLOAD PORT setting must be configured on the Development machine, either by running a utility to reconfigure RXTXDOS for the desired serial port, or by copying the appropriate version of RXTXDOS to a runtime directory prior to downloading a PROJECT.

%SETTINGS-GENERAL-DOWNLOAD PORT-COM1
<DOWNLOAD PORT - COM1

Directs the Shell to use the COM1 port for receiving projects over the serial interface.

To change the COM port for downloading, select the DOWNLOAD PORT menu option and choose a different COM port.

To exit the DOWNLOAD PORT submenu, click the right mouse button, press [ESC], or 

%SETTINGS-GENERAL-DOWNLOAD PORT-COM2
<DOWNLOAD PORT - COM2

Directs the Shell to use the COM2 port for receiving projects over the serial interface.

To change the COM port for downloading, select the DOWNLOAD PORT menu option and choose a different COM port.
To exit the DOWNLOAD PORT submenu, click the right mouse button, press [ESC], or tap outside the submenu.

%SETTINGS-GENERAL-DOWNLOAD PORT-COM3
<DOWNLOAD PORT - COM3

Directs the Shell to use the COM3 port for receiving projects over the serial interface.

To change the COM port for downloading, select the DOWNLOAD PORT menu option and choose a different COM port.

To exit the DOWNLOAD PORT submenu, click the right mouse button, press [ESC], or tap outside the submenu.

%SETTINGS-GENERAL-SERIAL IRQ
<GENERAL SETTINGS - SERIAL IRQ

Use this selection to specify the IRQs assigned to each of the COM ports on the PowerStation. Displays submenu of available COM ports.

Options include: COM1, COM2, COM3

A pop-up numeric keypad is used to set the IRQ settings for the COM ports. Press HELP when the keypad is displayed for information on how to use the pop-up numeric keypad.

To change the IRQs assigned to the selected port:
1. Enter a valid IRQ for the selected port on the keypad
2. Press ENTER to accept the IRQ

%SETTINGS-GENERAL-SERIAL IRQ-COM1
<SERIAL IRQ - COM1

Use this selection to specify the IRQ assigned to the COM1 port on the PowerStation. Displays pop-up numeric keypad to enter values.

To change the IRQs assigned to COM1:
1. Enter a valid IRQ for the selected port on the keypad
2. Press ENTER to accept the IRQ

Press HELP when the keypad is displayed for information on how to use the pop-up numeric keypad.

%SETTINGS-GENERAL-SERIAL IRQ-COM2
<SERIAL IRQ - COM2

Use this selection to specify the IRQ assigned to the COM2 port on the PowerStation. Displays pop-up numeric keypad to enter values.

To change the IRQs assigned to COM2:
1. Enter a valid IRQ for the selected port on the keypad
2. Press ENTER to accept the IRQ
Press HELP when the keypad is displayed for information on how to use the pop-up numeric keypad.

@SESSION-GENERAL-serial_IRQ-COM3
<serial_IRQ - COM3

Use this selection to specify the IRQ assigned to the COM3 port on the PowerStation. Displays pop-up numeric keypad to enter values.

To change the IRQs assigned to COM3:
1. Enter a valid IRQ for the selected port on the keypad
2. Press ENTER to accept the IRQ

Press HELP when the keypad is displayed for information on how to use the pop-up numeric keypad.

@SESSION-PWRSTA
<POWERSTATION SETTINGS

Select this button to display current POWERSTATION SETTINGS in the display area and load the POWERRTA SETTINGS main menu.

Main menu options include: CALIBRATE TOUCHSCREEN, KEYBOARD TEST, ADJUST BRIGHTNESS, etc.

Select this button to access the touchscreen calibration and keyboard options.

Required - A keyboard is required to select test POWERSTATION SETTINGS (except ADJUST BRIGHTNESS).

@SESSION-PWRSTA-CALIBRATE TOUCHSCREEN
<POWERSTATION SETTINGS - CALIBRATE TOUCHSCREEN

Select this button to calibrate the touchscreen. Displays confirmation and keyboard reminder before running DOS and launching the touchscreen calibration utility. You must confirm your selection before the utility will launch.

Calibration options include: YES or NO.

Required - A keyboard is required to adjust this PowerStation setting.

To calibrate the touchscreen from DOS (using utility), XXXXXXXX

@SESSION-PWRSTA-KEYBOARD TEST
<POWERSTATION SETTINGS - KEYBOARD TEST

Select this button to test the keyboard connected to the PowerStation. Displays a KEYBOARD TEST screen for confirmation of key presses on the keyboard.

KEYBOARD TEST options include: YES or NO.

Required - A keyboard is required to adjust this PowerStation setting.
To use the KEYBOARD TEST, press a key on the keyboard or front panel and check the test screen to confirm that the display results match the pressed key.

To exit the KEYBOARD TEST, press the [F1] function key twice.

NOTE: CTRL, ALT, and SHIFT keys will not display on the test screen when pressed on the keyboard.

## SETTINGS - POWERSTA-KEYBOARD TEST
<POWERSTATION SETTINGS - KEYBOARD TEST

Use this selection to test the keyboard connected to the PowerStation. Displays keyboard submenus with KEYBOARD options.

Options include: KEYBOARD TEST, SHIFT KEY xx, SHIFT KEY xx

REQUIRED - A keyboard is required to adjust this PowerStation setting.

To use the KEYBOARD TEST, press a key on the keyboard or front panel and check the test screen to confirm the display results match the key pressed.

To exit the KEYBOARD TEST, press the [F1] function key twice.

NOTE: CTRL, ALT, and SHIFT keys will not display on the test screen when pressed on the keyboard.

To use the SHIFT KEY xx, xxxxx xxxxx xxx.

To exit the SHIFT KEY xx, press the [F1] function key twice.

To use the SHIFT KEY xx, xxxxx xxxxx xxx.

To exit the SHIFT KEY xx, press the [F1] function key twice.

NOT ON P1

## SETTINGS - POWERSTA-ADJUST BRIGHT
<POWERSTATION SETTINGS - ADJUST BRIGHT

Use this selection to manually adjust the brightness of the PowerStation touchscreen display. Displays submenu with current brightness setting and ADJUST options.

ADJUST OPTIONS INCLUDE: DECREASE OR INCREASE

To adjust screen brightness, move the cursor over the DECREASE or INCREASE button, press xx or xx on the keyboard, or use the DECREASE or INCREASE buttons on the touchscreen.

To exit the ADJUST submenu, click the right mouse button, press the [ESC] key on the keyboard, or tap the touchscreen on an area outside of the submenu.

NOT ON P1

## SETTINGS - POWERSTA-ADJUST CONTRAST
<POWERSTATION SETTINGS - ADJUST CONTRAST

Use this selection to manually adjust the contrast of the PowerStation touchscreen display. Displays submenu with current contrast setting and ADJUST options.

ADJUST options include: DECREASE or INCREASE

To adjust screen contrast, move the cursor over the DECREASE or INCREASE button, press XX or XX on the keyboard, or tap the DECREASE or INCREASE buttons on the touchscreen.

To exit the ADJUST submenu, click the right mouse button, press the [ESC] key on the keyboard, or tap the touchscreen on an area outside of the submenu.

<SETTINGS-POWERSTA-ETHERNET PORT
<POWERSTATION SETTINGS - ETHERNET PORT

Use this selection to configure Ethernet port settings of the PowerStation. Displays Ethernet submenu with various port options. Each option will display a pop-up numeric keypad for user input to enter port values. See your network administrator for specific values to input for the ETHERNET PORT settings.

Options include: HOST NAME, IP ADDRESS, XXXX, XXXX, XXXX

REQUIRED - A keyboard is required to adjust this PowerStation setting.

To change PowerStation Ethernet port settings:
1. Enter a valid number for the port option on the keypad
2. Press ENTER to accept the user-input number

Press HELP when the keypad is displayed for information on how to use the pop-up numeric keypad.

<SETTINGS-POWERSTA-ETHERNET PORT-GATEWAY ADDRESS - Sample gateway address

Use this selection to specify the gateway address assigned to the PowerStation Ethernet port. Displays pop-up numeric keypad to enter address.

See your network administrator for a specific value to input for the GATEWAY ADDRESS setting.

To change the GATEWAY ADDRESS assigned to the Ethernet port:
1. Enter a valid gateway address for the selected port on the keypad
2. Press ENTER to accept the gateway address

Press HELP when the keypad is displayed for information on how to use the pop-up numeric keypad.

<SETTINGS-POWERSTA-ETHERNET PORT-IP ADDRESS
<ETHERNET PORT - IP ADDRESS
Use this selection to specify the IP address assigned to the PowerStation Ethernet port. Displays pop-up numeric keypad to enter address.

See your network administrator for a specific value to input for the IP ADDRESS setting.

To change the IP ADDRESS assigned to the Ethernet port:
1. Enter a valid IP address for the selected port on the keypad
2. Press ENTER to accept the IP address

Press HELP when the keypad is displayed for information on how to use the pop-up numeric keypad.

%SETTINGS- POWERSTA-ETHERNET PORT-HOST NAME
<ETHERNET PORT - HOST NAME

Use this selection to enter the HOST NAME for the PowerStation Ethernet port.

REQUIRED - A keyboard is required to adjust this PowerStation setting.

See your network administrator for a specific value to input for the HOST NAME setting.

To change the HOST NAME assigned to the Ethernet port:
1. Enter the host name for the selected port on the keyboard
2. Press ENTER to accept the host name

Press xx to exit the HOST NAME menu and return to the ETHERNET PORT menu.

%INFO
<SHITL INFO

Displays information about the PowerStation System and Interact Module Versions installed on the unit in the display area.

INFO submenu options include: SYSTEM, VERSIONS

To edit an INFO submenu button, XXXX XXXXXX XXXXXX
To add an INFO submenu button, XXXX XXXXXX XXXXXX
To delete an INFO submenu button, XXXX XXXXXX XXXXXX

%INFO-SYSTEM
<POWERSTATION SYSTEM INFORMATION

Displays current PowerStation information in the display area.

SYSTEM INFORMATION includes: Processor type, Amount of available memory (conventional and extended), logical drive assignments, and port address assignments (printer, serial, and ethernet).

NOTE: SYSTEM INFORMATION displayed from this menu option is for viewing purposes only and cannot be edited.
If there is more system information than can fit in the standard display area, a set of up and down arrows will appear in the top right portion of the screen on the separator bar. To activate the arrows, move the cursor over an arrow and click the mouse button, press the PAGE UP or PAGE DOWN buttons on the keyboard, or tap the arrow region on the touchscreen.

To return to the main menu, click xx on the mouse, press the xx key on the keyboard, or tap the BACK button below HELP.

INFO-VERSIONS
<POWERSTATION VERSIONS INFORMATION
Displays current Interact Module, MachineLogic, and system software version information in the display area.

Version information includes:
Interact (Application Manager, Modules, and Driver)
MachineLogic (Rt1Prt, FTS1DOS, and support software such as the Control Adapter)
System Software (DOS, BIOS, touchscreen driver, and Shell)

NOTE: SYSTEM INFORMATION displayed from this menu option is for viewing purposes only and cannot be edited.

If there is more system information than can fit in the standard display area, a set of up and down arrows will appear in the top right portion of the screen on the separator bar. To activate the arrows, move the cursor over an arrow and click the mouse button, press the PAGE UP or PAGE DOWN buttons on the keyboard, or tap the arrow region on the touchscreen.

To return to the main menu, click xx on the mouse, press the xx key on the keyboard, or tap the BACK button below HELP.

UTILITIES
<POWERSTATION UTILITIES
Displays a submenu with available Powerstation UTILITIES options.

Options include:
- BACKUP PROJECT
- RESTORE PROJECT
- SECURITY KEY
- CONTROL PANEL

NOTE: The CONTROL PANEL utility will not display on the subnet if MachineLogic is not included in the loaded Project.

To edit a UTILITIES submenu button, xxxxx
To add a UTILITIES submenu button, xxxxx
To delete a UTILITIES submenu button, xxxxx

UTILITIES-BACKUP PROJECT
<BACKUP PROJECT UTILITY
Select this button to configure.

Main menu options include: USE PATH 1, USE PATH 2, USE PATH 3, CONFIGURE PATHS.

To return to the main menu, click XX on the mouse, press the XX key on the keyboard, or tap the BACK button below HELP.

**UTILITIES-Backup Project-Use Path 1**

**Backup Project - Path 1**

Use this selection to choose the user-defined PATH 1 for backing up the Project loaded on the PowerStation. Displays confirmation submenu screen showing Project name and backup destination with backup options.

Options include: YES or NO

Selecting YES begins the backup and displays a status screen that represents the backup completion percentage as the project is backed up. The submenu will also display any errors encountered during backup (such as no disk present, write errors, etc.)

**NOTE:** The destination source must be linked to the PowerStation before backing up.

Selecting CANCEL immediately aborts the backup operation.

Pressing the NO button removes the PATH 1 submenu and returns to the main backup menu.

To exit the PATH 1 submenu, click the right mouse button, press [ESC], or tap outside the submenu.

**UTILITIES-Backup Project-Use Path 2**

**Backup Project - Path 2**

Use this selection to choose the user-defined PATH 2 for backing up the Project loaded on the PowerStation. Displays confirmation submenu screen showing Project name and backup destination with backup options.

Options include: YES or NO

Selecting YES begins the backup and displays a status screen that represents the backup completion percentage as the project is backed up. The submenu will also display any errors encountered during backup (such as no disk present, write errors, etc.)

**NOTE:** The destination source must be linked to the PowerStation before backing up.

Selecting CANCEL immediately aborts the backup operation.

Pressing the NO button removes the PATH 2 submenu and returns to the main backup menu.
Appendix H

Shell Help Description Document
PowerStation Shell Online Help

Description:

The Shell Help is an online reference file for users of the PowerStation. It provides the user with basic descriptive information about the Shell and its components. It is not a comprehensive file with in-depth information, lengthy explanations, or how-to instructions. It contains information about how to use and navigate the Shell and short descriptions of the menus and options available on the system. A more thorough version of the Shell Help, with additional information, will be incorporated into the MachineShop Getting Started Guide.

The Shell Help is a screen-based system accessed by pressing the Help button on the touchscreen (no external peripherals, such as a mouse or keyboard can access Help).

Each Help screen displays information that pertains to the visible menu or submenu. Help information is not available for every button or selection (context-sensitive).

Help contents are displayed using a fixed width font in the main display area window (40 characters per line, 18 lines per screen). Contents that cannot fit within one window screen display a set of small arrows, which are used to scroll to additional screens.

Parameters/Style:

The file used for Shell Help is a text document that is formatted as an Extensible Markup Language (XML) file with minor formatting. A standard tag and formatting system for the help file has been established for incorporation into the PowerStation Shell. Each help item for a screen or submenu is assigned a location code, which is processed by the Shell to display the help information for that particular item.

Help XML File Definition:

```
<Data>
<Application>
<Name>Shell Help</Name>
<Comment>File Description</Comment>
-Version>EX-00000-000</Version>
<Language>English</Language>
</Application>

<Text ID="xxxx" TITLE="Item Title">
Text for help item goes here.
</Text>
</DATA>
```

Help File Format Definitions:

\n Carriage Return
\b Bold
\i Italic
\u Underline
\s Strike-out
\h Halftone
\g# Hanging Indent with space #
\\ Backslash (character)

Samples:
\bword\b (word in bold)
\g4 (4 space hanging indent)
Appendix I

Shell Help Text ID Document
<table>
<thead>
<tr>
<th>Menu/Submenu Label</th>
<th>Text ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu Screen</td>
<td>1000</td>
</tr>
<tr>
<td>Run</td>
<td>1010</td>
</tr>
<tr>
<td>Settings</td>
<td>1020</td>
</tr>
<tr>
<td>Info</td>
<td>1030</td>
</tr>
<tr>
<td>Utilities</td>
<td>1040</td>
</tr>
<tr>
<td>Language</td>
<td>1050</td>
</tr>
<tr>
<td>Exit</td>
<td>1060</td>
</tr>
<tr>
<td>General Settings Main Menu</td>
<td>1100</td>
</tr>
<tr>
<td>Power On Operation</td>
<td>1120</td>
</tr>
<tr>
<td>Run MachineLogic</td>
<td>1130</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>2200</td>
</tr>
<tr>
<td>TCPIP Enable</td>
<td>2220</td>
</tr>
<tr>
<td>Host Name</td>
<td>2230</td>
</tr>
<tr>
<td>API IRQ</td>
<td>2240</td>
</tr>
<tr>
<td>TCP/IP Net Settings</td>
<td>2300</td>
</tr>
<tr>
<td>IP Address</td>
<td>2321</td>
</tr>
<tr>
<td>IP Mask</td>
<td>2322</td>
</tr>
<tr>
<td>Interface Type</td>
<td>2323</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>2326</td>
</tr>
<tr>
<td>Download Only</td>
<td>2327</td>
</tr>
<tr>
<td>PowerStation Settings Main Menu</td>
<td>1200</td>
</tr>
<tr>
<td>Calibrate Touchscreen</td>
<td>1210</td>
</tr>
<tr>
<td>Keyboard Test</td>
<td>1220</td>
</tr>
<tr>
<td>Keyboard Test (PS)</td>
<td>xxxxx</td>
</tr>
<tr>
<td>Adjust Brightness</td>
<td>1230</td>
</tr>
<tr>
<td>Adjust Contrast</td>
<td>1240</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Menu/Submenu Label</th>
<th>Text ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Project Main Menu</td>
<td>1500</td>
</tr>
<tr>
<td>Backup Path X</td>
<td>1502</td>
</tr>
<tr>
<td>Backup Configure Paths</td>
<td>1560</td>
</tr>
<tr>
<td>Backup Configure Path X</td>
<td>1561</td>
</tr>
<tr>
<td>Restore Project Main Menu</td>
<td>1600</td>
</tr>
<tr>
<td>Restore Path X</td>
<td>1620</td>
</tr>
<tr>
<td>Restore Configure Paths</td>
<td>1660</td>
</tr>
<tr>
<td>Restore Configure Path X</td>
<td>1661</td>
</tr>
<tr>
<td>Security Key Main Menu</td>
<td>1700</td>
</tr>
<tr>
<td>Security Enable Module keypad</td>
<td>1720</td>
</tr>
<tr>
<td>Security Transfer</td>
<td>1730</td>
</tr>
<tr>
<td>Security Transfer Type</td>
<td>4030</td>
</tr>
<tr>
<td>Control Panel Main Menu</td>
<td>1800</td>
</tr>
<tr>
<td>Control Panel Stop</td>
<td>1820</td>
</tr>
<tr>
<td>Control Panel Start</td>
<td>1830</td>
</tr>
<tr>
<td>Task Info</td>
<td>2100</td>
</tr>
<tr>
<td>Hardware Setup Main Menu</td>
<td>1900</td>
</tr>
<tr>
<td>Hardware Retentive Memory</td>
<td>1920</td>
</tr>
<tr>
<td>Hardware Power Fail</td>
<td>1950</td>
</tr>
<tr>
<td>Hardware Watchdog</td>
<td>1960</td>
</tr>
</tbody>
</table>
Appendix J

Project Tracker Document
<table>
<thead>
<tr>
<th>Project #</th>
<th>Project Name</th>
<th>Description</th>
<th>Status</th>
<th>Assigned</th>
<th>Completed</th>
<th>Work Done/Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxxxx-100</td>
<td>PowerStation &quot;shell&quot; Help</td>
<td>New text help system (DOS-based) for revamped &quot;shell&quot; interface on PowerStation units. Will create more complete print doc after online system and help are finished.</td>
<td>Active</td>
<td>11-Feb-99</td>
<td></td>
<td>Reviewed User Interface specifications and looked at current help system (text-based). Brief meeting with David B. for basic system guidelines. Began writing and organizing individual screen help copy for document (2/26). Created first draft of help system (3/09). First draft edit and submittal for review to David B. (3/11). Made changes to document style and added additional information to draft 2 (3/19). Talked to Kurt S. and Dan K. - system still in development (3/23). Loaded draft of online help into shell template - worked on minor system fixes with Dan K. &amp; Gary (3/26). Checked with Dan K. for additional work on Shell - will add a few screens for keypad help and error messages (4/09). Made minor fixes/changes and added extra keypad info. (4/15). Will work on changes/fixes/additions as system is developed (once or twice a week). Meeting with Dan K. - made corrections and fixes (5/03). Installed new file on test system for review (5/07). Created text ID reference file and made additional changes (5/10).</td>
</tr>
</tbody>
</table>

Appendix K

Finished Projects Document
<table>
<thead>
<tr>
<th>Project #</th>
<th>Project Name</th>
<th>Description</th>
<th>Status</th>
<th>Assigned</th>
<th>Completed</th>
<th>Work Done/Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>05665-100</td>
<td>5x86 Serial Port Installation</td>
<td>Installation guide for third serial port install on non-pentium PowerStation units.</td>
<td>Finished</td>
<td>15-Feb-99</td>
<td>26-Feb-99</td>
<td>Make changes to documentation based on notes and meeting with Dave K. Check accuracy and consistency, update graphics and tables. Given to Doug C. for edit (2/16), changes noted and reviewed (2/17), 2nd review with Doug C., reviewed by Dave K., fixes made, and given to QC (2/19). Made consistency fixes (2/22). Fixed double-submital mixup with Alex R. - file resubmitted for audit (2/24). PDF and zip files submitted on disk (2/26). Loaded to servers and HTML updated (3/02).</td>
</tr>
<tr>
<td>Document ID</td>
<td>Description</td>
<td>Status</td>
<td>Start Date</td>
<td>Finish Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------</td>
<td>---------</td>
<td>------------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05691-100</td>
<td>P2 Release Notes</td>
<td>Finished</td>
<td>23-Feb-99</td>
<td>08-Mar-99</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>P2 PowerStation Release Notes with changes and/or corrections</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update &quot;Miami&quot; Beta1 Quick Start Guide to updated &quot;MachineLogic&quot; Beta2 QSG.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05661-100</td>
<td>Pentium Serial Port Installation</td>
<td>Active</td>
<td>15-Feb-99</td>
<td>12-Apr-99</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Installation guide for third serial port install on pentium PowerStation units.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05726-100</td>
<td>MachineLogic Tutorial</td>
<td>Finished</td>
<td>21-Mar-99</td>
<td>06-May-99</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edit of former MachineLogic Training Manual and prepare for online distribution (PDF).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Make changes to documentation based on notes and meeting with Dave K. Check accuracy and consistency, update graphics and tables. Given to Doug C. for edit (2/16), changes noted and reviewed (2/17). Made consistency changes and resubmitted draft to Doug C. (3/23). Added additional information and resubmitted draft (3/29). Made changes to draft using other docs and submitted with audit (4/05). Archived and posted files after making font-related fixes and zipping, updated webpage. (4/12).

<table>
<thead>
<tr>
<th>Project Code</th>
<th>Project Title</th>
<th>Description</th>
<th>Status</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>05741-100</td>
<td>P1 PC/104 Cover Installation</td>
<td>Installation instructions for P1 PowerStation PC/104 covers (3 models).</td>
<td>Finished</td>
<td>09-Feb-99</td>
<td>20-May-99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Small changes and updates for consistency with current documentation (including graphics). Worked with Dave K. Working on revision to update to new P1 specs, might need change to kit cover documentation. Clarification from Marianne B. - new doc for covers (3 models) and maintain current PC/104 Adapter doc for older units. Met with Mike B. for basic installation instructions (2/23). Created draft of instructions (2/24). Waiting for parts to come in for pictures and specific procedures. Checked with Mike B. - parts still not in (3/23). Quick review of installation with Mike B. (3/26). Submitted draft to David B. for review (4/05). Made changes based on notes, added graphic - draft given to Mike B. for review (4/12). Incorporated minor changes and submitted to QC with audit form (4/14). Received minor changes from QC - resubmitted with new project # (keeping old doc 04987-100) (5/14). Submitted archive on disk and updated servers (5/20).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Initial meeting with David B. to review project assignment (3/29). Read MachineShop spec and begin search of Shell and IPM references (3/30). Began collecting information and jotting down changes and questions (4/15). Project assigned to Mary Pat, reviewed project highlights (5/06). Additional review with M.P. - check consistency of models and help with updates to Px (week of 5/10). Worked on updates and consistency changes, printed draft to review (5/19). Project transferred to Mary Pat.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>