PACIFIC SCIENTIFIC

PacCom

Pacific Scientific Motion Control Products

Software ToolKit

Instruction Manual

August 1990

Version 3.0

1 INTRODUCTION

1.1 Description of PacCom

PacCom is a menu driven software package that contains several software utilities to aid in the use of Pacific Scientific Digital Motion Control Products. These utilities are designed to run on an IBM compatible PC and interface to the Pacific Scientific motion control hardware via an RS-232 serial link. The Main Menu summarizes the tools available:

1) Select Hardware
   - The Select Hardware utility is used to set-up PacCom for the hardware being used. All Pacific Scientific positioning hardware both servo and stepper is supported.

2) Terminal Emulator
   - The Terminal Emulator utility allows the PC to be used as a dumb terminal. In this mode, the PC is acting as a terminal and allows the generation and editing of programs directly on the hardware (i.e. on-line).

3) Upload Utility
   - The Upload Utility allows files to be read out of the controller's memory into a file on the PC's floppy or hard disk drive.

4) Download Utility
   - The Download Utility allows files to be taken from the PC's disk drive and transferred into the position controller's memory.

5) Syntax Checker
   - The Syntax Checker allows programs to be checked for errors before transferring them to the controller.

6) Editor
   - The Editor allows the creation and editing of programs. Programs can be created and edited without the controller being connected to the PC (i.e. off-line). The resulting program can be stored on the PC's disk drive for downloading to the controller at a convenient time. Also, programs can be read from the hardware using the Upload Utility and be modified using the Editor.

7) Init Serial Port

- The Init Serial Port utility allows the communication parameters of the serial communications port to be set. These parameters include baud rate, port selection, number of data bits, etc.

PacCom is contained on a single 5-1/4 inch diskette. The disk also contains two binary files, PORT.CFG and HARD.CFG. These are for the configuration of the serial port and hardware selection, when PacCom is brought up. If these do not exist on the disk, they will be created when PacCom is run.

There are also several example programs on the distribution disk. These programs have -05 extensions for PacSci Motion BASIC examples, 524 extensions for 5240 examples, and 544 extensions for 5440 examples.

1.2 Getting Started

PacCom can be executed directly from the PacCom disk supplied or it can be installed on the PC's hard disk. If PacCom is to be executed directly off the disk supplied, it is recommended that a backup copy be made and stored in a safe place.

1.2.1 Using the PacCom Floppy Disk

Follow the steps below to use the PacCom floppy disk:

1. Boot up the PC
2. Set the PC to directory A:
3 Insert the PacCom floppy disk into drive A
4. Type PACCOM **<enter>**

PacCom will be loaded and the Main Menu screen will appear. PacCom is now running and you can select the tool desired.

1.2.2 Installing PacCom on a Hard Drive

The PacCom disk is supplied with an installation program. This program will create a sub-directory \PACCOM on the hard drive and copy the PacCom files from the floppy disk into the sub-directory. The program will also copy a program

PACCOM.BAT onto the root directory of the hard drive so that PacCom can be executed directly from the root directory. The installation program assumes that the hard drive is designated C:.

To install PacCom on the PC's hard drive, follow the steps below:

1. Boot up the PC
2. Insert PACCOM disk in drive A
3. Type A: INSTALL **<Enter>**

PacCom is now installed on the hard disk. To run PacCom get to the root directory, C: , and type PACCOM **< Enter >**.

1.2.3 Serial Port Connections

PacCom can be used to generate programs off-line. However at some point programs have to be downloaded or uploaded between the PC and the motion control hardware. The RS-232 serial link is used to communicate between the PC and the hardware.

Many PCs have 2 serial communication ports, COM1 and COM2. One of these must be wired to the motion control hardware. Refer to the PC's Hardware Reference Manual and the Instruction Manual provided with the motion control hardware for wiring information. Use the Init Serial Port tool on the Main Menu to select the correct COM port and to set the serial link parameters.

2 HOW TO USE THE MAIN MENU AND SUB-MENUS

2.1 Keyboard Commands

The choices from a main menu are highlighted by typing the**<Up >** or **<Down>** arrow keys. Selection of the highlighted item is made by entering the choice with the **<Enter >** key. This selection will bring up a prompt, a list and/or a sub-menu. Choices are made from a sub-menu or list using the **<Up>** key to move up, or the **<Down >** key to move down, then entering a choice with the **<Enter >** key.

Typing the <Esc> key will abort the current operation, and return to the previous menu, or exit the program. The <Esc> key is also used to exit from the Editor. To exit from a Terminal Emulator session, type <Ctrl><E>.

The keyboard command definitions are displayed in appropriate places on the screen to assist the user.

2.2 Main Menu Tools

2.2.1 Select Hardware

Upon entering this utility, a controller list menu is presented. When exiting this utility, the controller hardware/software configuration is saved on disk, at the current directory, in a file named 'HARD.CFG'.

2.2.2 Terminal Emulator

The PC can be used as a dumb terminal to communicate with the selected controller. The PC's serial port is initialized when PacCom is started. The parameters can be changed with the Init Serial Port utility.

To exit the Terminal Emulator at any time, type <Ctrl><E>, which returns to the Main Menu.

2.2.3 Upload Utility

After selecting this utility, the user is prompted for what filename to name the uploaded file. By default, the uploaded file will be created in the current directory. If the user selects a filename that already exists, then it will be overwritten, thus it is good practice to use a different extension or filename for the uploaded file. In particular, the filename.ext of the source file that was downloaded should not have the same filename.ext as the uploaded file, or the source file contents will be replaced with the uploaded file. (Note that when a BASIC program is uploaded, the comments in the REM lines are not there.) When choosing extensions for filenames, the use of ".LST" should be avoided. The ".LST" extension is used by the PacCom program for the listing files created by the Syntax Checker.

After a file has been successfully uploaded, the message "Upload completed" is displayed. If the **<Path>**\filename.ext is unacceptable, then the message "Couldn't open file" is displayed.

When doing an upload from a 5200 series controller, if the uploaded file is longer than what can fit on one screen of the Upload Utility, the user will be prompted for the next address from which to continue the upload. Thus, the user needs to know the next address before starting the upload. To find out the next address to continue an upload from, the contents of the controller memory can be listed with the Terminal Emulator utility. The proper address can be determined from the listing.

When uploading a file containing a program written in PacSci Indexer Language (5200 Series), labels will be generated for any branch to address command. Note that for a program that has been downloaded, the symbolic labels and references are stripped out and replaced with numeric addresses. At uploading, these have been replaced with numeric sequential labels. It is good practice to use a different extension or filename for uploaded files. Label generation for PacSci Indexer Language is covered in Section 3.5, "Upload Utility".

2.2.4 Download Utility

After selecting this utility, the user is prompted for a **<Path>**\filename.ext which can be typed in or selected from the current directory list. If the download is successful, the message "Download completed" is displayed.

2.2.5 Syntax Checker

The Syntax Checker can only be used on the programs for certain controllers; SC150 and SC450 Series which use PacSci Motion Basic, the 5200 Series which use PacSci BASIC Indexer Language, and 5500/5600 series controllers which use PacSci Motion BASIC. By using the Syntax Checker, programs can be checked for syntax validity, prior to downloading or running.

Syntax Checker creates a listing file with the extension ".LST" given to the file that was sent to Syntax Checker or sent to the Download Utility (which automatically calls the Syntax Checker). If the syntax check is successful, the message "No syntax error(s)

detected" is displayed and the listing file with the message "NO ERRORS FOUND" is sent to the display. If errors are found, then the message "Syntax error(s) detected" is displayed, and the listing file, containing the program code and syntax error messages, is displayed. The Editor can be used to review the syntax errors listed in the .LST file. The syntax checking for PacSci Indexer Language and PacSci Motion BASIC is discussed in Sections 3 and 4.

2.2.6 Editor

After selecting this utility, the user is prompted for a<**Path**>\filename.ext which can be typed in or selected from the current directory's file listing. If the filename does not exist, the user is prompted with "Create?" to which the response is Y (yes) or N (no). If answering "no", the system responds with the message "can't load file" (it doesn't exist), after which, a <**Enter**> returns to the Main Menu level. If answering "yes" to the prompt, the user is put into the Editor.

When exiting the Editor (typing <**Esc**>), the user is prompted with the question, "Save file (y/n) ?". Answering n <**Enter** > , returns to the Main Menu, without saving the file, and any edits are lost. Answering y <**Enter**> will bring up a prompt for the filename. The current file being edited can then be saved under the existing filename by typing <**Enter**> , or, the name can be changed by typing over the current filename, followed with a <**Enter**> .

While using the Editor, many operations such as justify, cut, paste, etc. are available. These operations are assigned to various keys:

ESC	Exit the Editor.
INSERT	Toggles insert and typeover mode, (default: typeover mode).
F7	Attach file to end of current file (default: off).
F8	Toggles whether hard and soft carriage returns are symbolically displayed in the edit window, (default: not displayed)
F9	Toggles editor display from 128 ASCII characters to 256 IBM characters, (default: ASCII characters).
F10	Reformats a paragraph.
UP ARROW	Moves cursor up a line.

DOWN ARROW	Moves cursor down a line.
LEFT ARROW	Moves cursor left a space.
RIGHT ARROW	Moves cursor right a space.
CTRL-A	Move cursor one word to the left.
CTRL-F	Move cursor one word to the right.
CTRL-Y	Delete line cursor is on.
CTRL-Z	Scroll up a line.2.
CTRL-W	Scroll down a line.
HOME	Move cursor to beginning of line.
END	Move cursor to the end of line.
PGUP	Previous Page.
PGDN	Move to top of screen.
CTRL-END	Move to bottom of screen.
CTRL-PGUP	Move to beginning of file.
CTRL-PGDN	Move to end of file.
TAB	Tab.
ALT-M	Begin marking a block of text, use cursor arrows to mark rest of block.
ALT-C	Copy marked block to scrap.
ALT-X	Cut marked block to scrap.
ALT-P	Paste scrap at cursor position.
ALT-S	Search for pattern (target); starts search, cancels search.

2.2.7 Init Serial Port

After selecting this utility, the serial port parameters can be initialized or changed. If the parameters do not need to be changed, then type **<Esc>**. Typing **<Esc>** at any point in the Init Serial Port menu will exit without saving any changes to the parameters.

The parameters are selected by using the up or down arrow keys. As each parameter is highlighted, it's sub-menu displays the list of each parameter's values. The choices can be selected with the **<Up>** and **<Down>** keys. A parameter and it's value are entered with a **<Enter>**. To exit the Init Serial Port menu after making changes to any parameters, type **<Enter>** after the last parameter in the menu.

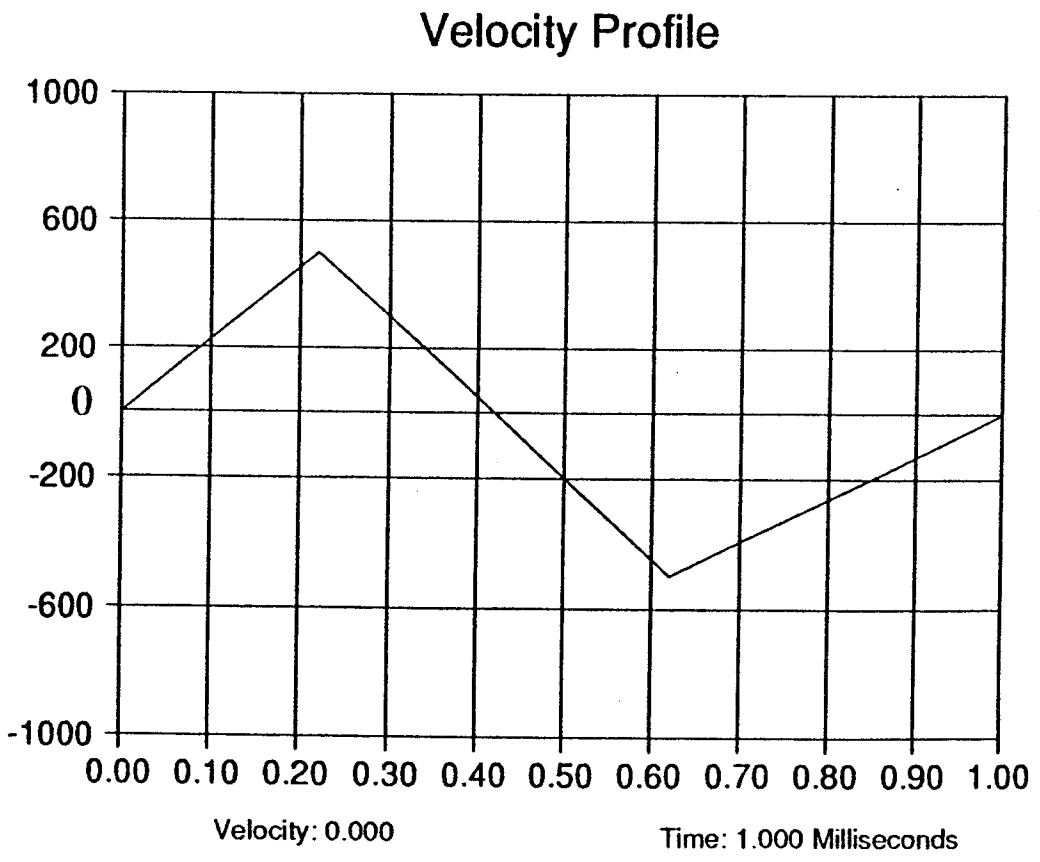
Any changes made to the parameters are saved to a disk file in the current directory after exiting the Init Serial Port menu, and thus are used as the default settings, upon the next invocation of PacCom. These default settings are stored in a binary data file named 'PORT.CFG'.

2.2.8 Velocity Profile Generator

Introduction Use the PacCom Velocity Profile generator to "draw" your motion profile. Drawing a profile is more convenient than programming the equivalent multiple sets of variables and motion functions for the profile in Motion BASIC.

To draw the profile, you specify the boundary parameters, then use the arrow and other keys to define the line segments that comprise the profile. The program will not allow you to specify unrealistic motion, such as an immediate acceleration or deceleration faster than specified.

The completed profile is named, saved, and downloaded to the servocontroller. The Motion BASIC program calls the profile to run using the GO.PROF (Go Profile) command.



Accessing the Velocity Program

1. Use arrows to move to "Velocity Profile", then press **<enter>**
2. Press **<y>** to display the Velocity Profile Parameters screen. (After the program is created, you may press **<n>** here to download your program for editing).

Establishing the grid

Enter the following parameters to establish the boundary variables for the grid. Press **<enter>** after each selection.

- Total time (seconds)
- Minimum Velocity (RPM) -- may be positive (cw) or negative (ccw)
- Maximum Velocity (RPM) -- may be positive (cw) or negative (ccw)
- Maximum Acceleration (RPM/sec)
- Maximum Deceleration (RPM/sec)

Press **<enter>** when complete. The grid is displayed with an arrow at the zero position.

Creating the profile

1. Use the arrow keys to move the arrow to the endpoint for the first motion line segment. Refer to the location readouts at the bottom of the grid. To move the key faster, press **<shift>** and the arrow key.

   Note: *You may only move to a point that defines an allowable motion.*

2. When at the appropriate position, press **<spacebar>**. The program draws a line from the previous endpoint to this line segment endpoint position.

   Note: *When crossing the zero velocity line, define an endpoint at zero if the absolute value of the preceding acceleration or deceleration is less than the absolute value of the desired new value. When the zero line is crossed, deceleration changes to acceleration and, if no endpoint is specified, the new rate is limited by the opposite-type rate.*

3. Continue specifying line segments as desired. The last line segment must end at zero, or a final segment will be drawn and forced to zero.

4. Press **<end>** to complete the profile. Once **<end>** is pressed, you may edit the profile.

Editing the profile

1. Edit the profile, creating new endpoints or deleting an endpoint, as follows:

   a. Press **<spacebar>** to enter the edit mode.

   Creating new line segments:

   Add new line segments by moving the arrow to define the endpoint. The line segments on either side of the arrow "stretch" to conform to the arrow position.

   Deleting line segments:

   Delete line segments by moving the arrow to the endpoint between the line segments on the profile and pressing **<delete>** to eliminate that endpoint. The line segment reforms between the two endpoints on either side of the eliminated endpoint.

   Note: *The arrow must be moved to within 5 screen units of an endpoint to delete that endpoint.*

   b. Press <enter> to stop editing and allow moving the arrow on the profile.

Exiting and saving the program

1. Press **<esc>** to exit the Velocity Profile screen.
2. Press **<y>** to save the profile, then type in up to eight characers for the profile name. The program is saved in two files, with extensions to the profile name of filename.DLD and filename.BIN
3. Download the profile to the drive by pressing **<y>**. Any profile currently on the controller is overwritten,

Running the profile

Perform the motion profile in your Motion BASIC program using the GO.PROF command. When this command is encountered, the program accesses andd runs the defined profile.

3 PACSCI INDEXER LANGUAGE FOR THE 5200 SERIES

3.1 Introduction

PacSci Indexer Language is a modified form of the program language of the PacSci Model 5220/5240 Stepping Motor Indexer/Driver. See the 5220/5240 Instruction Manual concerning programming of the controller.

The language for the 5220/5240 has been modified so that symbolic labels can be used as the operand for branch to address commands, for example "J", the JUMP command. Syntax checking is done on the Indexer Language, and labels are translated to their target address by the Syntax Checker before a program is downloaded. When a program is uploaded, labels are generated for the operands of branch to address commands.

3.2 Use of Symbolic Labels in Programming

A line of program code has this format:

<LABEL> COMMAND OPERAND1 OPERAND2

where LABEL is optional, and OPERAND2 exists for some commands. One or more blanks must be used between a label and a command, or between operands, but blanks are not necessary between the command and OPERAND1. Tabs may be used in place of blanks. The branch to address commands, "G" (GOTO), "J" (JUMP) and "U" (LOOP), may have a label or an address for OPERAND1. The label reference in OPERAND1 must have a matching label somewhere in the program, or a syntax error is generated. Duplicate label names are not allowed and will cause a syntax error to be generated.

The format for a label is:

: < STRING >

where there are no blanks between the colon (:) and the string. The string is limited to eight characters composed of any sequence of

digits and letters (including the underscore). Any variation from this format will generate a syntax error. The maximum number of labels allowed for any program is 200.

Here is an example program that uses labels:

```
: begin         k
                j : begin 1
: loop          +1000
                u : loop 5
                g : end
: end           s
```

The labels in this program are "begin", "loop", and "end".

3.3 Syntax Checker

The Syntax Checker checks for valid syntax for each line of code. If no syntax errors are found, the Syntax Checker takes out all symbolic labels and label references, and inserts the target address for each label reference, (a label reference is a label used as the operand for a branch to an address command).

If an error is found, the program and error messages are put into a listing file which is displayed. The possible error messages are listed below:

1. "invalid input; label or program command expected"
2. "invalid label; expecting alphanumeric after delimiter"
3. "invalid label or, expecting valid command after label"
4. "invalid operand; digit '1' expected"
5. "invalid operand; expecting digit"
6. "EOL encountered; expecting operand"
7. "invalid input;no operand required"
8. "invalid input; expecting EOL character"
9. "invalid input; expecting valid number or label"
10. "illegal for this command to have a label reference"
11. "invalid operand; +,- character not allowed after +,- command"
12. "invalid number; input number is out of range"
13. "invalid operand; reference to nonexistent label"

14. "invalid label; duplicate label name exists"

Note: *The end of line character (EOL), which is defined as the linefeed character, is expected at the end of each line of code.*

3.4 Download Utility - Address Translation of Labels

Before a 5220/5240 program is downloaded, it is automatically syntax checked. If there are no syntax errors, the Syntax Checker takes out all labels from the program and translates the label references to address. The program thus contains only commands and numeric operands, and is downloaded to the controller.

3.5 Upload Utility - Label Generation for Addresses

When a 5220/5240 program is uploaded, labels are generated for the operands of branch to address commands. The labels are then inserted in the appropriate places in the program. The labels generated have the form :Lxxx,where the labels are sequentially generated starting with :L000.

Note that if a 5220/5240 program is written off-line on the PC with labels and then downloaded, when the program is uploaded the labels will take the form described above. The labels used in the program prior to downloading will exist only in the original file on the PC.

To avoid confusion when developing 5220/5240 programs, it is recommended that the labeling scheme used by the Upload Utility be followed. That is, when writing a 5220/5240 program on the PC, use labels of the form: Lxxx starting with :L000 and progressing sequentially through the program. By following this practice, a program will not change labels when downloaded and then uploaded.

4 PACSCI MOTION BASIC FOR THE SC150 AND SC450 SERIES

4.1 Introduction

PacSci Motion BASIC is available on certain controllers. This option is only available for the SC150 and SC450 Series servo controllers and on 5500/5600 series stepper controllers.

4.2 Syntax Checker and Download Utility

The Syntax Checker checks for valid syntax for each line of code. If no syntax or system errors are found, then the program is downloaded. If an error is found, the program and error messages are put into a listing file which is displayed. This .LST file can be examined with the Editor.

The type of syntax errors detected are identical to the type of syntax errors detected by the PacSci Moton BASIC Interpreter syntax checker. Consult the SC150 or SC450 PacSci Motion BASIC Instruction manual provided with your position controller for syntax error interpretation.