OPERATING MANUAL

MODEL 855
PROGRAMMABLE CONTROLLER SYSTEM

Newport Corporation
18235 Mt. Baldy Circle
P.O. Box 8020
Fountain Valley, CA 92728-8020

Ver. #8502

TABLE OF CONTENTS

FIG. 1 - MODEL 855 PROGRAMMABLE CONTROLLER SYSTEM

1.0 INTRODUCTION AND DESCRIPTION

About the 855 Programmable Controller System

The Model 855 Programmable Controller is a microprocessor-based system that allows simultaneous direct or programmable control of up to four Newport linear actuators or rotary stages. Its simple but powerful vocabulary of mnemonic commands allows straightforward programming and control via the handheld 855K Keypad/Display or, with its standard RS-232C and IEEE-488 interfaces, a remote computer, terminal or modem.

The 855C is compatible with Newport's **850 Series Linear Actuators** and **Model 496 Power Rotation Stage.** Each of these devices use DC motor drives with integral optical encoders for smooth operation and high resolution. The 855C fully supports their resolution and range.

The 855C has four basic operating modes. All are accessible via the 855K Keypad/Display or the RS-232C or IEEE-488 ports. EXECUTE-mode, which the 855C is in at power-up, allows manual precision control of four actuators. From EXECUTE-mode, you may move back and forth between PROGRAM EDIT-mode, PROGRAM RUN-mode, and JOG-mode. PROGRAM EDIT-mode is for entering, reviewing and editing up to 300 855C instructions into its memory for later, automatic execution in PROGRAM RUN-mode. JOG-mode provides a convenient means for repetitively moving actuators back and forth by a specified increment.

About this Manual

This manual has been carefully prepared to give all the information needed to use the 855 System productively. To get you up to speed quickly, its basic principles of operation are presented in Section 2.5, "First-Time Operation." Reading through this section will give you the background you'll need to use the complete instructions that follow. Each 855C function, mode and command is described in detailed discussions, and summary tables and indexes are provided for quick reference.

(This page intentionally left blank.)

1.1 The 855C Controller

The 855C Programmable Controller is the nucleus of a system that automatically controls up to four Newport precision positioners and stages. Its large, easy-to-learn instruction set and standard RS-232C and IEEE-488 interface ports allow it to work closely with external computers and other data devices. Its programmability provides stand-alone automatic control of actuator motion, yet no knowledge of programming techniques is required. It also supports the optional 855K handheld Keypad/Display for convenient data entry, control and program editing.

Specifications

Actuator Control:	Provides ±15 VDC, 1.5 Amp power and ±10V velocity control signals for up to 4 Newport linear or rotary positioning devices. Receives and decodes dual output, 90° phase, +12.5, -0.5 V encoder pulses and limit signals.
CPU:	8 bit 6809 high-performance microprocessor
RAM:	4 Kbytes
PROM:	24 Kbytes
EPROM:	2 Kbytes
I/O:	RS-232C Serial Port IEEE-488 Parallel Port
Power:	110-220 VAC, 50/60Hz (selectable) 850 mA with all actuators fully loaded
Dimensions:	5.25" high X 17" wide X 12" deep

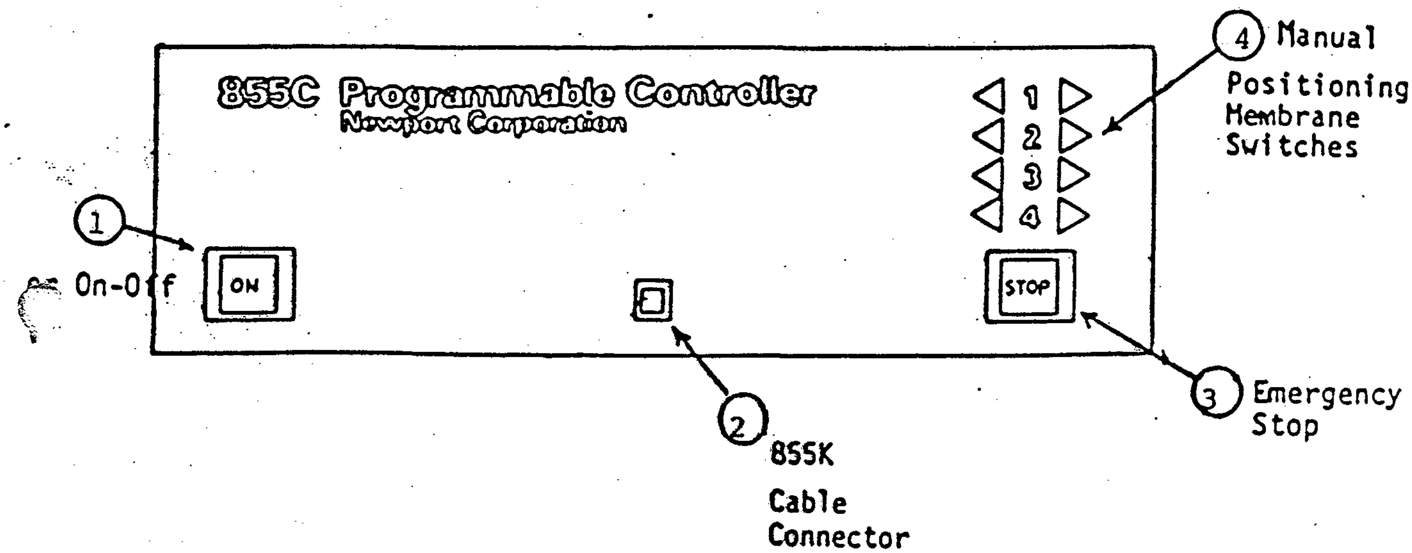


FIG. 2 - 855C FRONT PANEL

1.1.1 Front Panel

The 855C front panel is shown in Fig. 2. The ON button <<1>> applies power. The coiled cable for the 855K Keypad/Display plugs into the front panel connector <<2>>.

The STOP button <<3>> is intended for use as a panic stop control. When depressed, the STOP button:

--halts all actuator motion. Motion will not resume upon release.

--terminates processing of commands and programs. If a program was running, the 855C is placed in EXECUTE mode and "T at xxx" is sent to the active controlling device to indicate program termination at line number xxx.

1.1.2 Manual Slew Operation

The eight arrow-shaped membrane switches <<4>> move the drives under manual control when held down. The left-hand arrow switches retract linear actuators and turn rotational drives counter-clockwise; the right-hand arrow switches extend linear drives and turn rotational drives clockwise.

Manual slew control overrides any command or programmed motion and ignores preset drive parameters such as velocity and soft actuator limits. Manual slew will proceed even if the drive velocity has been set to zero (drive "off") or if the STOP button is depressed. The 855C will, however, keep track of the drives' positions when they are slewed.

When an arrow key is depressed, manual slew motion will commence at a velocity of 0.05 mm/sec (or 0.5^{o}/sec). If the arrow key is held down, the drive will gradually speed up to its maximum velocity of 0.4 mm/sec (4.0^{o}/sec) after about five seconds. The drive will immediately stop when the arrow key is released. (When an arrow key is depressed with the STOP switch engaged, manual slew proceeds at maximum speed, and all drives will immediately be returned to their original positions when the STOP switch is released.)

The manual slew arrow keys can produce a slow jog motion if rapidly depressed and released.

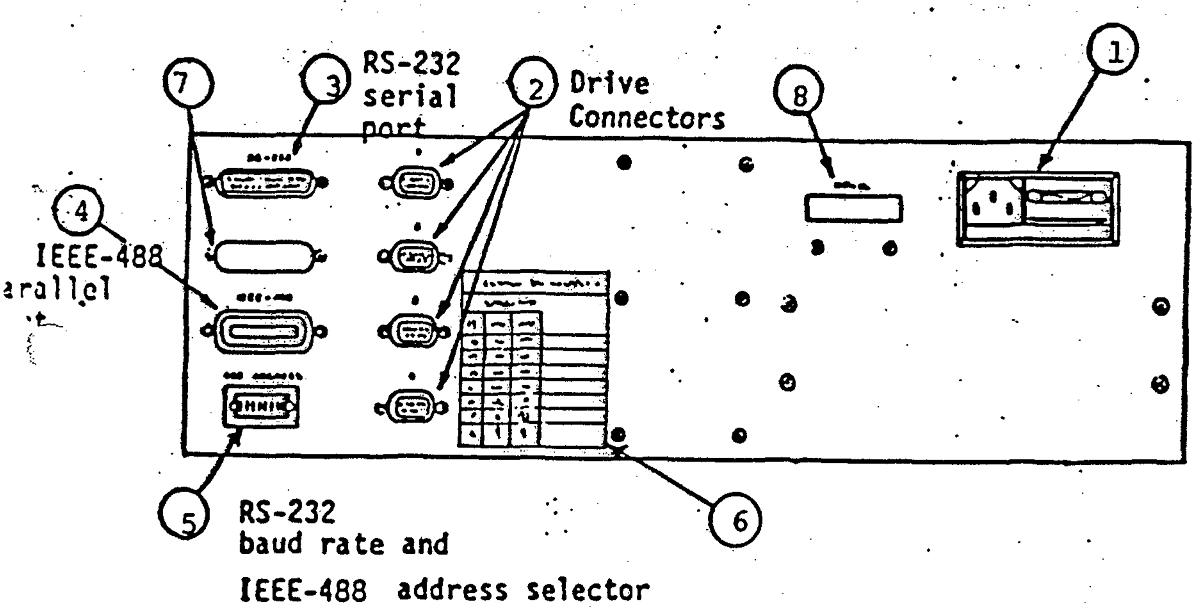


FIG. 3 - 855C REAR PANEL

1.1.3 Rear Panel

The 855C rear panel is shown in Fig. 3. The 855C can be used with 100, 120, 220, or 240 Volt, 50 or 60 Hz service and is supplied ready for operation at 120 volts. The combination line fuse, voltage selector and power connector is located on the upper right corner of the rear panel <<1>>.

Drive Connectors

One to four drives may be connected to the 9-pin miniature D-connectors <<2>>. These connectors mate to Newport's Series 850I-xx Cables (where xx is the cable length). The drives are numbered on the rear panel next to the connectors.

Interface Ports

The 855C's RS-232C port is provided at a female DB-25 connector <<3>> on the rear panel. This port is configured as DTE. See Section 3.4, "RS-232C Serial Port," for detailed discussion of the RS-232C port and protocol.

A standard IEEE-488 connector is shown at <<4>>. This port is discussed in Section 3.5, "IEEE-488 (HPIB) Port."

A DIP switch <<5>>, labelled **RS-232/IEEE-488**, sets the IEEE-488 device address and the RS-232C baud rate. Next to it <<6>> is a reference table showing the switch setting options. Setting this switch is described in Section 3.0.

Changing the Supply Voltage

The voltage selector card is set at the factory for 120 Volt operation. If the 855C is to be used with other line voltages, the selector card position must be changed.

Disconnect the line cord and slide the clear plastic fuse cover to the left, exposing the fuse and the voltage selector card. Pull the **FUSE/PULL** lever out and to the left to extract the fuse and expose the voltage selector card. This small printed circuit card selects different line voltages when inserted in different ways. Insert it so the desired voltage, printed on the card, is visible. Swing the fuse extractor lever back into place.

Use a 2 Amp fuse for 100 or 120 Volt operation and a 1 Amp fuse for 220 or 240 Volt operation. Install the proper fuse, slide the clear plastic cover back into place, and reconnect the line cord.

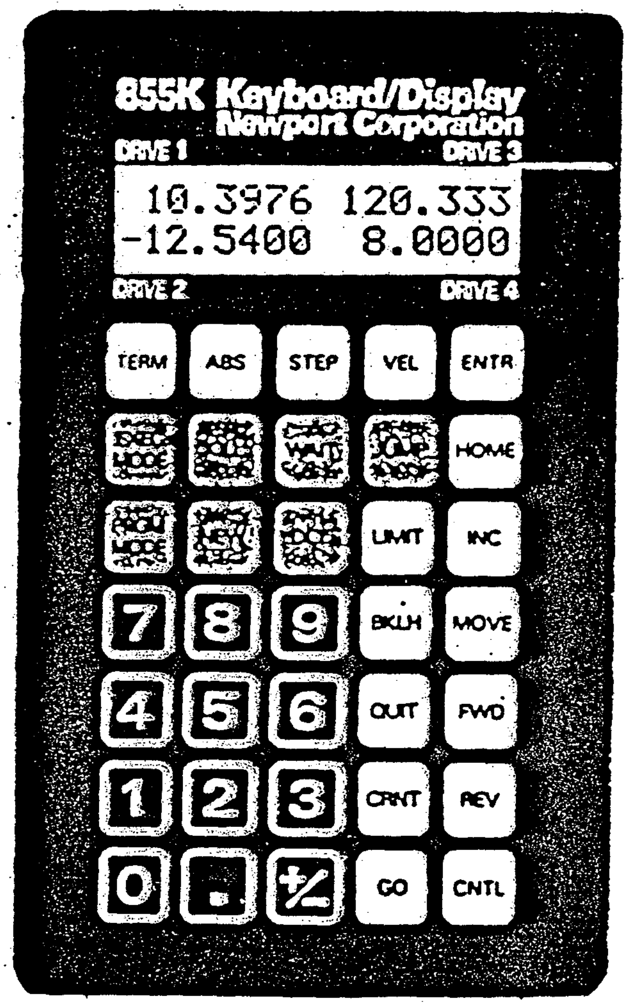


Fig. 4- 855K Keypad/Display

1.2 855K Keypad/Display

The 855K is shown in Fig. 4. It can be used to completely control and program the 855C. It is handy for manual control of actuator position and for reviewing and editing 855C programs-- including programs downloaded from a remote computer. Its backlit alphanumeric liquid-crystal display shows position information for the four drives, programming instructions for the 855C, and messages from the 855C. The 855K's keypad has 35 durable, tactile-response membrane keys for command and data entry. The 855K connects to the 855C via a coiled cable; its jack is on the front panel of the 855C.

Backlight

The backlight for the display is turned on and off by pressing the 855K's CNTL key while holding down the 0 key. (Use the CNTL key with caution. See Section 2.8.4, "PROGRAM RUN-mode," for a discussion of the CNTL key's other functions.)

Assigning Control to the 855K

Upon power-up, control must be assigned to the 855K, the RS-232C port, or the IEEE-488 port. Control assignment and reassignment are discussed in detail in Section 2.2, "Assigning Control to the 855K or External Device." Briefly, to assign control to the 855K Keypad/Display, press QUIT 0 on the 855K immediately after power is applied and the display prompts for control assignment. The system will then be ready to receive control instructions from the 855K.

1.3 System Hardware Description

The 855C is built around an 8-slot STD-Bus chassis. Four empty slots remain in the standard system and may be used for specialized purposes.

The 855C's CPU card is built around a Motorola 6809 microprocessor. This card contains the 855C's memory (system ROM and user memory) and the serial interface for the 855K. The system is supplied with memory sufficient to store 300 855C instructions. The top 200 steps are stored in non-volatile EEPROM that retains its information when the unit is powered-down but which may be overwritten. Several programs may be resident in the 855C at one time.

The general-purpose interface card contains the IEEE-488 and RS-232C interfaces and configuration switches and the eight lines that code the actuators' limits.

The 855C's encoder-decoder board contains the counters that monitor the actuators' direction and position.

A Digital-Analog Converter card provides drive voltage and servo control to the actuators. The voltages can range from -10 to +10 Volts with 12-bit resolution (4096 steps).

Access to the card cage is obtained by removing the front panel of the 855C. Disconnect the AC power cord. Remove the four flathead screws (two on each side) and pull the panel forward. The cards are secured in the cage by a retainer bracket which is fastened to the card cage by two thumbscrews at the top and bottom right-hand side of the cage.

2.0 855 OPERATION

The 855C is controlled by 26 commands sent to it by the controlling device, which may be the 855K Keypad/Display or a remote computer, terminal or modem connected to the RS-232C or IEEE-488 interfaces. The four operating modes of the 855C allow immediate execution of commands (EXECUTE-mode), storage of commands for later review and editing (PROGRAM EDIT-mode), automatic execution of stored commands (PROGRAM RUN-mode), and repetitive back-and-forth actuator motion over a selected increment (JOG-mode).

The 26 commands follow a consistent format: a single-letter mnemonic instruction code, followed (usually) by a single number designating the drive, a second number (with sign and decimal point if needed) giving the command data, and a carriage return, <CR>. On the 855K, the instruction code letter is input by pressing a single button on the keypad, and the <u>ENTR</u> key acts as a carriage return. When the 855C receives an erroneous command, it responds with a question mark (?).

Commands are sent through the RS-232C and IEEE-488 interfaces in the form of standard ASCII strings, as are responses from the 855C. Special considerations for use of these interfaces are given in Sections 3.0 - 3.5. Assigning control to an interface is discussed in Section 2.2, "Assigning Control to the 855K or External Device."

2.1 855C Operating Modes

The 855C has four modes of operation. Each is separate and distinct from the others. The four modes are EXECUTE-mode, JOG-mode, PROGRAM RUN-mode, and PROGRAM EDIT-mode. Fig. 5 shows the interrelationships among the operating modes and the mnemonic commands used to switch between modes.

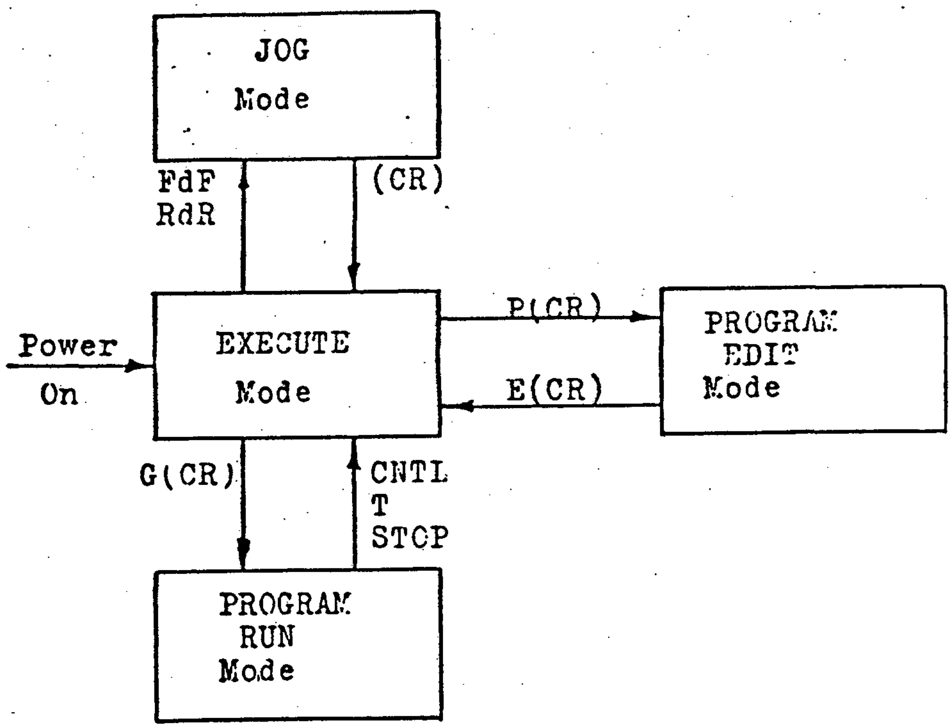


Fig. 5- 855C Operating Modes

EXECUTE-mode

This is the 855C's main operating mode. It is the default mode upon power-up. All the other modes are accessible via EXECUTE-mode.

If the 855K Keypad/Display is connected, the positions of all four actuators are displayed and continuously updated approximately twice per second. This is true even if the RS-232C port or the IEEE-488 port has been assigned control. Position information is not sent to the RS-232C or IEEE-488 ports on a continuous basis; it must be requested by a query command at the time it is desired. (Or, in PROGRAM RUN-mode, a programmed query command will send the information to the active controlling device.)

In EXECUTE-mode, motion commands are executed immediately upon entry. If a motion instruction includes operations for a drive which is currently executing a command, the command will be held until the drive's motion is complete. Then the new command will be executed. During the time the command is held, no new commands will be processed (though they may be queued in the RS-232C and IEEE-488 ports' 1,000-character input buffer). Once entered, commands-in-waiting may be cancelled by the TERMINATE command (which will also bring all drive motion to a halt).

Because of the command-stacking capability of the RS-232C and IEEE-488 ports' input buffer, a sequence of commands can be downloaded for sequential execution by the 855C.

JOG-mode

JOG-mode allows manually-controlled incremental actuator motion at a preset velocity using the FORWARD and REVERSE commands. The jog step size is programmable and must be set to some non-zero value prior to entering JOG-mode.

JOG-mode is activated by entering the FORWARD or REVERSE command (on the 855K, by pressing the FWD or REV key) and the number of the drive to be jogged (1 through 4, or 0 to jog all four drives); then the FORWARD and REVERSE commands may be entered to initiate jog motion of the designated drives. A carriage return <CR> (855K ENTER key) terminates JOG-mode.

PROGRAM EDIT-mode

PROGRAM EDIT-mode is used for entering, editing and reviewing stored command sequences. The 855C can store up to 300 programmable commands in its memory, and several programs can be stored simultaneously for independent execution in PROGRAM RUN-mode.

PROGRAM EDIT-mode, like the 855C's other two special modes, is entered from EXECUTE-mode. It is activated by entering P<CR> from either the RS-232C or IEEE-488 port or by depressing the 855K's **PRGM MODE** and **ENTR** keys.

The 855C's program storage memory is divided into two sections: a volatile memory area (RAM) which stores program step numbers 000-099, and a non-volatile memory area (EEPROM) which retains program steps 100-300 even when the 855C is powered-down. There is no operational difference in using the two types of memory. The RAM area should be used when program changes are frequent or must be loaded or changed quickly. The EEPROM section's permanent (but overwriteable) storage is useful for frequently-needed programs, but writing to this type of memory is a relatively slow process (90 milliseconds per command). EEPROM write-cycle lifetime is also limited to fewer than 10,000 writes.

When in PROGRAM EDIT-mode, the 855C will send to the controlling device the step number at which the program counter currently points along with the instruction stored there. (When the 855C is new, **TERMINATE** commands fill its program memory. Its volatile RAM--steps 000-099--is cleared and refilled with **TERMINATE** commands at power-up.)

PROGRAM RUN-mode

When the 855C is in PROGRAM RUN-mode, instructions stored in its memory are executed automatically and sequentially.

PROGRAM RUN-mode is entered from EXECUTE-mode by issuing a GO command from the controlling device. Program execution stops and the 855C returns to EXECUTE-mode when a TERMINATE command is encountered, either as a program step or if entered via the controlling device. Program execution will also cease and put the 855C in EXECUTE-mode if the 855K's CNTL key is depressed; this key also transfers control of the 855C to the 855K. Depressing the 855C's front-panel STOP button will also halt a running program and return the 855C to EXECUTE-mode when the button is released.

While executing a program in PROGRAM RUN-mode, the 855C will respond to only three commands: the TERMINATE and CNTL instructions just mentioned, and the ? query, which is discussed in Section 2.8.3 and which is entered via the RS-232C or IEEE-488 ports only. All other inputs will be disregarded and lost.

2.2 Assigning Control to the 855K or External Device

Up to three controlling devices may be connected to the 855C at any time: the 855K Keypad/Display (via its own port on the front panel of the 855C), and remote devices (via the RS-232C and IEEE-488 interfaces on the rear panel). The 855C allows only one active controlling device at a time; the others will be ignored until specifically assigned control.

The QUIT command assigns control of the 855C to a specified port so that subsequent commands from that device will be executed and information can be returned to it. In particular, when the 855C is first powered-up the user must inform it which port will be used for instructions and data. The 855C will poll each of the ports until one assigns control by sending a QUIT command.

Command	855K Key	String Format
QUIT	QUIT	Qp<CR>

The QUIT command assigns control to port number **p**, as designated below.

p = 0	Command assigned to 855K
p = 1	Command assigned to RS-232C
p = 2	Command assigned to IEEE-488

On the 855K, the ENTR key acts as a carriage return. The carriage return is optional when the 855C is polling the ports immediately after a power-up. Subsequent QUIT commands require a carriage return.

The QUIT command may be programmed into the 855C for execution when the program is run. When control is reassigned, the 855C will send "quit to <<new device>>" to the requestor.

Once control has been assigned to a port, only that port can reassign control to another port. For example, if the Q0 command had been entered (assigning control to the 855K), only the 855K could subsequently reassign control. (This would be true even if the Q0 command had been entered via the RS-232C or IEEE-488 ports.)

This means that if no device is connected to a port to which control is assigned, the 855C will not respond to commands from any other port. There is one exception: the CNTL key on the 855K will grab the attention of the 855C at any time--even when it is running a program--and will assign control to the 855K, terminate actuator movement, and put the 855C in EXECUTE-mode. There is no corresponding command string for other controlling devices, and the CNTL command is not programmable.

When control is assigned to either the IEEE-488 or the RS-232C interface port, the 855C's 1,000-character input buffer will prevent data loss if several commands are issued in rapid succession. Commands--including QUIT commands--will be queued for sequential execution at the 855C's pace. Note, however, that control reassignment occurs when a QUIT command is executed, not when it is issued. Control reassignment may be significantly delayed if the 855C is working its way through a block of queued commands.

If the QUIT command is queued in the 855C's input buffer, commands will still be accepted (into the queue) from the device that issued the QUIT command (and will not be accepted from the "new" controlling device) until the QUIT command is actually executed.

It is therefore good practice to ensure that no data is sent from any controlling device after it has reassigned control to another device, and the "new" controlling device should send no commands until the 855C has responded to the QUIT command by sending "quit to <<new device>>" to the requestor.

2.3 Significant Figures and Numeric Field Formatting

Command data for the 855C is interpreted to seven significant figures. The 855C keeps either four or three decimal places depending on whether it has been instructed to expect linear units (the power-up default, in millimeters, with four decimal places) or angular units (in decimal degrees, with three decimal places) for the drive.

Command data may be entered in decimal form (with an explicit decimal point) or in counts of resolution. If a decimal point is not explicitly input, the 855C will intepret the data as counts of resolution (0.0001 mm/count for linear actuators or 0.001^{o}/count for rotary actuators).

Note

Without an explicit decimal point, the numeric input nnnnnnn will be interpreted as counts of resolution, equivalent to either nnn.nnnn mm (linear drive) or $nnnn.nnn^{o}$ (angular drive).

If more than seven digits are entered as data without an explicit decimal point, the 855C will keep the seven most significant digits. (Some commands may not accept strings longer than seven digits unless a decimal point is explicitly input. In these cases, an error message will be returned from the 855C.)

If a decimal point is entered, any digits beyond the three or four places allowed for the drive will be ignored.

Examples:

123, with no explicit decimal point, will be interpreted as counts of resolution. For a rotational drive, this will be equivalent to 0.123^{o}; for a linear actuator, it will be interpreted as 0.0123 mm.

123.45678, when input as data for a rotational drive, will be interpreted as 123.456^{o}. For a linear actuator, it will be interpreted as 123.4567 mm.

2.4 Selecting Linear or Rotary Drives

The 855C assumes it is controlling linear actuators unless told otherwise. For linear actuators, data is displayed and interpreted in millimeters, with four decimal places (resolution to 0.1 micron).

If drive d is a rotary drive, you will want to instruct the 855C to interpret data in decimal degrees with three decimal places (resolution to 0.001^{o}). This is done by pressing . (decimal point), d (the number of the drive--the number of its connector on the 855C back panel), and ENTR on the 855K. The 855C will respond by formatting that drive's position data to three decimal places on the 855K Display.

The same character sequence--with ENTR replaced by a carriage return character-- may be sent to the 855C over the RS-232C or IEEE-488 interfaces.

Recapping:

. d ENTR	Toggles drive d to angular units (3 decimal places);
. d ENTR	...a second time toggles drive d back to linear units.

. d m

AXIS

0 850
1 496
2 495

2.5 First-Time Operation

It's a good idea to read through the entire Operations Manual before using the 855 Controller System. This section is intended as a guide for first-time use for those who prefer the "plug-in-and-go" approach to learning equipment operation. We recommend that you read through the following discussion on 855K Control even if you will be controlling the 855C via one of its interface ports.

855K Control

This section assumes that you will be using the 855K Keypad/Display to control the 855C.

Check the voltage selector card position at the AC line connector for compatibility with your line voltage (See Section 1.1.3 for instructions on changing the line voltage setting). The unit comes configured for 120 Volt operation.

Connect your actuator(s) and the line cord to their receptacles on the rear panel. Connect the 855K's coiled cable to its connector on the 855C's front panel.

Depress the ON switch. The switch will illuminate and the 855C's cooling fan will begin running. Ensure the STOP switch is released (not illuminated).

At this point, any actuator may be operated manually by the membrane arrow switches on the 855C front panel (see Section 1.1.1, "Front Panel").

The message "Assign Control" will be displayed on the 855K. Perform the *following keystroke* sequence: QUIT 0. The message "quit to Handheld-ready for input" will appear, and the 855K is now ready to control the 855C.

If any rotary stage actuators are connected, enter the keystroke sequence: . d ENTR, where d is the drive number of the rotary drive (the number of the rear-panel connector to which the drive is connected). Unless the .d entry is made, the 855C will presume the drive is a linear actuator (see Section 2.4, "Selecting Linear or Rotary Drives").

Enter a velocity parameter for each axis you want to activate. For example, to set the velocity of drive 1 to 0.4 mm/sec, enter the keystroke sequence: VEL 1 .4 ENTR. As each drive is activated (as its velocity is set to a non-zero value), the 855K display will change to show its current position (along with the current position of all the other active drives). Inactive drives will have "OFF" displayed in place of a position on the 855K.

Now the 855C is ready to move the actuators at the velocities you've entered for them. For example, to extend actuator 1 a distance of 1 mm from its present position, enter the keystroke sequence: MOVE 1 1. ENTR. The actuator will move (at the velocity previously set) to a position 1.0 mm from the absolute origin defined when power was applied (or by the COORDINATE command).

All 855C instructions and data are entered in this same manner from the 855K. Even if you'll be controlling the 855C with a remote computer or other data device, the 855K will often come in handy for quick actuator manipulation and program editing.

Serial Interface Control

This section assumes that you'll be using a remote device connected to the RS-232C serial port to control the 855C.

First, read Sections 3.0 - 3.4 thoroughly before attempting to interface to the 855C via the RS-232C port. Successfully interfacing two RS-232C devices requires complete understanding of the operation of both machines.

Check the voltage selector card position at the AC line connector for compatibility with your line voltage (See 1.1.3 for instructions on changing the line voltage setting). The unit comes configured for 120 Volt operation.

Assuming that the mode of interface operation has been chosen and a properly configured cable is available: connect the remote device's cable to the DB25 connector on the 855C's rear panel, and connect your actuators and the AC line cord to their receptacles on the 855C.

Select the appropriate baud rate at the rear panel of the 855C. 600 baud is recommended for initial operation.

Depress the ON switch on the 855C front-panel. It will illuminate, and the cooling fan will run. Ensure that the STOP switch is off (not illuminated).

At this point, any actuator may be operated manually by the membrane arrow switches on the 855C front panel (see Section 1.1.1, "Front Panel").

From the controlling terminal or computer, send a Q1 string, heeding the bit format and other conventions discussed in Sections 3.0 - 3.4.

The 855C will respond with two messages: "quit to RS-232" and "ready for input".

The 855C is now ready for data and instructions. To set the velocity of actuator 1 to 0.4 mm/sec, input the command string: V 1 . 4 <CR>. (Do not imbed spaces or send a linefeed character instead of the carriage return.) Once the velocity is set to a non-zero value, the actuator may be moved. To move actuator 1 exactly 1.0 mm from its present position, enter the command string: M 1 1. <CR>. The 855C will move the actuator to an absolute position of 1.0000 mm (with respect to the absolute origin defined at power-up or by the CURRENT COORDINATE command).

IEEE-488 Interface Control

Read Sections 3.0-3.3 and Section 3.5 thoroughly before attempting to operate the 855C under control of the IEEE-488 port. Successfully interfacing two devices with this interface requires complete understanding of both machines.

Check the voltage selector card position at the AC line connector for compatibility with your line voltage (See Section 1.1.3 for instructions on changing the line voltage setting). The unit comes configured for 120 Volt operation.

Connect your actuators, line cord, and the IEEE-488 standard cable to their rear-panel connectors.

Select the appropriate device address on the 855C's rear-panel DIP switches.

Depress the ON switch. The switch will illuminate and the cooling fan will start. Ensure that the STOP switch is off (not illuminated).

At this point, any actuator may be operated manually by the membrane arrow switches on the 855C front panel (see Section 1.1.1, "Front Panel").

Minding the addressing and talking/listening conventions discussed in Sections 3.0 - 3.3 and 3.5, send a Q2 command string to the 855C from the host computer/bus controller. The 855C will respond with two message strings: "quit to HPIB" and "ready for input".

The 855C is now under IEEE-488 control and is ready for commands and data. To set the velocity of actuator 1 to 0.4 mm/sec and then move it 1.0 mm, enter the command strings: V 1 .4 <CR> and then M 1 1. <CR> (do not imbed spaces). This will activate drive 1, moving it to an absolute position of 1.0 mm (with respect to the origin defined at power-up or by the CURRENT COORDINATE command). Enter C1<CR>. The 855C will respond with the current position of actuator 1, which will be "1.0000".

2.6 855C Instructions and Command Set

The 855C "wakes up" in EXECUTE-mode when first powered-up and control is assigned. When the 855C is in EXECUTE-mode, instructions are put into action immediately after input. This allows real-time control of actuator position and motion and parameters such as actuator velocity and backlash compensation. Some commands control the 855C itself, and are used to take it into and out of PROGRAM EDIT-mode, JOG-mode, PROGRAM RUN-mode, and so on. All EXECUTE-mode operations fully support the drives' resolution and range capabilities. Most 855C commands may be stored as 855C program steps for later execution.

When the 855C is awaiting a command in EXECUTE mode, the 855K Keypad/Display will show the positions of all four actuators. (If fewer than four actuators are connected, the position of the "missing" actuators will be displayed as zero or "OFF".) The position display will be updated approximately twice per second. The position display will be suppressed during data or command entry on its keyboard.

2.6.1 Data Entry Instructions

Of the 855C's commands, "data entry instructions" are those which set the actuator parameters. The commands are VELOCITY, CURRENT COORDINATE, ABSOLUTE DESTINATON, STEP, LIMIT, BACKLASH, and FORWARD and REVERSE jog step size.

The 855C treats linear and angular drives differently; it keeps seven significant figures and makes certain assumptions if no decimal point is explicitly input. See Section 2.3, "Significant Figures and Numeric Field Formatting," for discussion of the 855C's numeric formatting requirements or if rotational drives are used.

Command	855K Key	String Format
VELOCITY	VEL	Vd.nnnn<CR> or Vdn.nnn<CR>

Sets the linear or angular velocity of drive d to 0.nnnn mm/sec or n.nnn degrees/sec. Velocity is always positive and any leading minus sign entered is ignored. The velocities of all drives are set to zero at power-up; the 855K Display will report that a drive with its velocity set to zero is "off". The maximum speed for **850 Series Actuators** is 0.4 mm/sec. The maximum speed for **Model 496 Rotation Stages** is 4.0 degrees/sec. Entering 0 for the drive designator will set the velocities of all four drives simultaneously. For best controller performance, the velocity of axes with no drive connected should be set to zero.

Command	855K Key	String Format
STEP	STEP	Sdnn.nnnn<CR> or Sdnnn.nnn<CR>

Sets the increment step size of drive d to $\pm$nn.nnnn millimeters or $\pm$nnn.nnn degrees, for use with subsequent INCREMENT commands. The step sizes for all drives are set to zero at power-up and may be set anywhere in the range of ± 99.999 mm (linear drives) or $\pm 999.999^{\circ}$ (rotational drives). Entering 0 for the drive designator sets the step size of all four drives.

Command	855K Key	String Format
CURRENT COORDINATE	CRNT	Cdnn.nnnn<CR> or Cdnnn.nnn<CR>

Assigns the value $\pm$nn.nnnn millimeters or $\pm$nnn.nnn degrees to the current position of drive d. This also establishes the "absolute" origin for the drive, used for subsequent positioning with the MOVE and HOME commands. The coordinates of all drives are set to zero at power-up. The range of coordinates allowed is ± 99.9999 mm or $\pm 999.999^{\circ}$. If the coordinate assignment would place the drive outside the soft limits defined by the LIMIT command, the assignment is not made and an error message is returned to the controlling device. As with other commands, entering 0 for the drive designator will assign the specified value to the current positions of all four drives.

Command	855K Key	String Format
ABSOLUTE DESTINATION	ABS	Adnn.nnnn<CR> or Adnnn.nnn<CR>

Sets the absolute destination of drive <u>d</u> to nn.nnnn mm or nnn.nnn° for subsequent <u>MOVE</u> commands. The absolute destination is with respect to the absolute orgin defined by the <u>CURRENT COORDINATE</u> command. The power-up default absolute destination for all drives is their origin. The absolute destination may vary between ±99.9999 mm or ±999.999°. Entering <u>0</u> for the drive designator will assign the absolute destination of all drives simultaneously.

Command	855K Key	String Format
LIMIT	LIMIT	Ldrnn.nnnn<CR> or Ldrnnn.nnn<CR>

Sets software upper (r=1) or lower (r=2) limits of drive <u>d</u> to ±nnn.nnnn absolute millimeters or ±nnnn.nnn absolute degrees. Absolute measurements are with respect to the origin established by the <u>CURRENT COORDINATE</u> command. The power-up default limits are ±99.9999 mm or ±999.999 degrees; these are also the maximum limits allowed. If the limit assignment would place the current position of the drive outside the limit, the limit assignment is not made and an error message is returned to the controlling device. Entering <u>0</u> for the drive will set the soft limit for all four drives.

Command	855K Key	String Format
FORWARD	FWD	Fdnn.nnnn<CR> or Fdnn.nnn<CR>
REVERSE	REV	Rdnn.nnnn<CR> or Rdnn.nnn<CR>

These commands are used to preset the jog step size of drive <u>d</u> prior to entering JOG-mode. The jog step sizes may be anywhere between ±99.9999 mm (linear drives) or ±999.999° (rotational drives). Entering <u>0</u> for the drive designator will simultaneously set the jog step size for all four actuators. The jog step size for all drives is initially zero at power-up; the jog step size for the drive <u>d</u> must be set to a non-zero value before JOG-mode may be entered. See the discussion of JOG-mode in Section 2.7.

2.6.2 Backlash Compensation

Mechanical assemblies such as stages and actuators almost always exhibit a small amount of backlash or mechanical play. This, if not compensated, can cause position hysteresis or lag when drive direction is reversed.

However, backlash errors do not occur if the drives consistently move to their destination in the same direction. The 855C always moves its drives to their final positions by approaching in the positive direction (extension or clockwise). Movements in the negative direction (retraction or counter-clockwise) automatically overshoot the target by the amount preset by the 855C's BACKLASH parameter, then approach the target position in the positive direction. This takes up the mechanical slack in the apparatus.

Command	855K Key	String Format
BACKLASH	BKLH	Bd.Onnn<CR> or Bd.nnn<CR>

The 855C's BACKLASH command sets the backlash compensation of drive d to 0.0nnn millimeters (linear actuator) or 0.nnn decimal degrees (rotational drive), up to a maximum of 0.0255 mm or 0.255 decimal degrees. Entering 0 for the drive designator sets the backlash compensation of all four drives.

The power-up default backlash compensation values are 0.0128 mm or 0.128 decimal degrees. These exceed the values normally required to compensate for backlash in typical drives and stages. Each drive's backlash compensation should be set to at least the amount of total system play which may be encountered or which can be determined experimentally. To minimize the temporary position overshoot, the recommended minimum backlash compensation value is given in each Newport drive's data sheet.

2.6.3 Queries

Data-entry commands can also be used to request the status of drives and their current settings and parameters. To do this, the instruction is input with the necessary drive designators but without data.

Query Command	855K Key	String Format
VELOCITY	VEL	Vd<CR>

The VELOCITY command, with only a drive number and no data, will return the current velocity of drive d. If V0 is input, the current velocities of all four drives will be returned to the controlling device.

STEP	STEP	Sd<CR>

The STEP command, entered with a drive designator but no data, will return the current step size of the drive. Entering S0 will return the current step sizes of all four drives.

CURRENT COORDINATE	CRNT	Cd<CR>

Entered with a drive designator but without data, the CURRENT COORDINATE command will return the instantaneous position of drive d. If C0 is input, the instantaneous position of all four drives will be returned.

ABSOLUTE DESTINATION	ABS	Ld<CR>

Entered with only a drive designator, the ABSOLUTE DESTINATION query will return the present absolute destination setting for drive d. Entering A0 will return the absolute destination settings for all four drives.

Query Command	855K Key	String Format
LIMIT	LIMIT	Ldr<CR>

The LIMIT command, with a drive number and limit but no data, will return the current absolute soft limit of drive d. If r=1, the upper soft limit will be returned; if r=2, the lower soft limit will be returned. Entering L0r will return the upper or lower soft limits of all four drives.

FORWARD FWD Fd<CR>

REVERSE REV Rd<CR>

Entered with only a drive designator but with a carriage return, the FWD and REV commands return the jog-mode step size setting of drive d. If d = 0, the current jog step size settings of all four drives will be returned. See Section 2.7 for a discussion of JOG-mode.

BACKLASH BKLH Bd<CR>

The BACKLASH command, with no following data, will return the current backlash compensation setting for drive d. If 0 is entered as the drive designator, the current backlash compensation setting for all four drives will be returned to the controlling device. See Section 2.6.2, "Backlash Compensation."

.C .CRNT .C<CR>

.C will return the status of the PROGRAM RUN instruction step pointer. It is programmable. The response to the .C command is "inst # xxx", where xxx is the current value of the instruction step pointer (this is the instruction step that will be executed next if GO<CR> is entered). On the 855K, .C is entered by pressing . and then the CRNT key.

2.6.4 Motion Instructions

These instructions initiate movement of one drive or all drives simultaneously. They require previous entry of drive operation parameters such as velocity, step size, etc. Other commands--including other motion instructions--may be entered while the drives are moving, though commands involving a moving drive will be executed only after it has stopped.

Command	855K Key	String Format
HOME	HOME	Hd<CR>

Causes drive d to move to the origin previously established by the CURRENT COORDINATE command. Motion will proceed at the speed previously set for that drive by the VELOCITY command. Entering HO will bring all drives to their origins.

INCREMENT	INC	Id<CR>

Causes drive d to step a distance previously set for that drive by the STEP instruction. Motion proceeds at the velocity preset for that drive by the VELOCITY command. Entering IO will simultaneously increment all drives.

MOVE	MOVE	Md<CR>

Moves drive d to the absolute destination previously set for the drive with the ABSOLUTE instruction. Motion proceeds at the speed preset by the VELOCITY command. Note: the absolute destination is with respect to the origin established by the CURRENT COORDINATE command. Entering MO will move all drives to their respective absolute destinations.

MOVE can also be combined with a destination parameter in the format Mdnn.nnnn or Mdnnn.nnn. This will cause drive d to move at its preset velocity to position ±nn.nnnn. It is equivalent to the sequence of Adnn.nnnn followed by Md, and the absolute destination parameter will be reset accordingly. (See the discussion of the ABSOLUTE parameter.)

Command	855K Key	String Format

TERMINATE TERM T (no <CR> required)

This immediately stops all actuator motion when entered in EXECUTE-mode, and control is returned to the currently active port. If a program is running, the program step currently being executed will be completed, the program will halt, and the string "T at xxx" will be returned to the controlling device to indicate program termination at line number xxx. Program execution can be resumed by issuing the GO command.

TERMINATE, when entered as a program statement, serves to halt program execution or mark a program's end; see Sections 2.8.2 and 2.8.3.

2.6.5 Echo Suppression Control

Command	855K Key	String Format
!	(NONE)	!<CR> (RS-232C/IEEE-488 only)

Often only query responses and error messages are of interest to the user. The ! command suppresses echo responses to input commands sent via the RS-232C or IEEE-488 interface ports. The ! command is not available for the 855K Keypad/Display. It is not programmable. Repeated ! inputs will toggle the echo response state. (The RS-232C port continues to oerate at full duplex, allowing simultaneous, bi-directional data transfer, when echo response is inhibited.)

2.7 JOG-Mode Instructions

In JOG-mode, the 855C can repetitiously jog any actuator--or all of them--for an individually preset distance and speed. The jog step size and velocity must be set to non-zero values for the drive prior to entering JOG-mode.

JOG-mode is not programmable. The FWD and REV commands have completely different functions when the 855C is in PROGRAM EDIT-mode (Section 2.8.2).

Command	855K Key	String Format
FORWARD	FWD	Fdjjj.jjjj<CR> or Fdjjjj.jjj<CR>,
REVERSE	REV	Rdjjj.jjjj<CR> or Rdjjjj.jjj<CR>,

These commands set up the 855C to jog drive d forward or backward by the jog step of jjj.jjjj millimeters or jjjj.jjj degrees. The jog step must be set to a non-zero value before JOG-mode may be initiated. Both commands have the same effect when setting the jog step size parameter; it doesn't matter which is used to set the parameter. Jog motion will occur at the speed set previously for each actuator by the VELOCITY command.

After the jog step size for drive d has been set, JOG-mode may be initiated by entering the command FWDd or REVd without a carriage return (either command may be used to initiate JOG-mode). Subsequent FWD and REV commands (without a drive designator or carriage return) will jog drive d back and forth by its preset jog step size and at its preset drive velocity.

A carriage return (855K ENTR key) terminates JOG-mode. The jog size for drive d, once set, remains in memory until changed or the 855C is powered-down. Once the jog size has been set, the JOG-mode for drive d may be reinitiated by the FWDd or REVd command, followed by the string of FWD and REV commands.

Note that if a negative jog step is entered, JOG-mode's FWD and REV commands switch meanings.

All actuators may simultaneously be jogged (with their individually preset velocities and jog steps) by the initiating JOG-mode with the **FWDO** or **REVO** command, followed by the string of **FWD** and **REV** commands.

2.8 Programming the 855C

The 855C is in EXECUTE-mode after power-up. Most 855C commands may be stored as progam steps in the 855C's internal memory. To do this, the 855C must first be placed in PROGRAM EDIT-mode. To execute the program, the 855C is returned to EXECUTE-mode and then placed in PROGRAM RUN-mode.

The 855C's standard hardware configuration has enough memory to store 300 program steps. The program steps from 000 to 099 are stored in volatile RAM, which is cleared and filled with TERMINATE commands when the 855C is powered-up. Program steps from 100 to 300 are stored in EEPROM and remain until overwritten; there is no difference in procedure for entering or executing programs in the different sections of memory. Several programs, separated by TERMINATE statements, may be resident in memory simultaneously.

2.8.1 Entering PROGRAM EDIT-Mode

Command	855K Key	String Format
PROGRAM MODE	PRGM	P<CR>

This command places the 855C in PROGRAM EDIT-mode. Each command string with data, entered in this mode, forms one program step in the 855C's internal memory. The program may then be run when the 855C is returned to EXECUTE-mode.

When the 855C is in PROGRAM EDIT-mode, it sends the contents of the current program step (the one to which the "program pointer" points) to the controlling device. The program step's command mnemonic and data will be preceded by its line number. Entering a new command will overwrite the current program step and move the program pointer to the next program step. Use the NEW command to insert new commands between existing program steps.

Command	855K Key	String Format
EXECUTE MODE	EXEC	E<CR>

This returns the 855C to EXECUTE-mode for program execution or real-time processing of commands.

2.8.2 PROGRAM EDIT-Mode

In PROGRAM EDIT-mode, commands may be entered, reviewed and changed for later automatic execution. Several commands are available for easy PROGRAM EDIT-mode review and editing of 855C programs. None are themselves programmable.

Command	855K Key	String Format
NEW	**NEW**	N<CR>

Inserts a new program step following the current program step and copies the current program step into it. The current line may then be overwritten by entering a new command. This, effectively, inserts the new command. All following program steps--up to the first two consecutive TERMINATE statements--are moved up a line. (Separating programs by blocks of **TERMINATE** commands ensures that they may be edited independently.) When program steps are inserted using the NEW, be sure to update any **JUMP** statements' destinations that might have been shifted. They are not automatically updated.

KILL	KILL	Knnn-rrr<CR>

Deletes program steps nnn through rrr inclusive. If no line numbers are entered, the **KILL** command deletes the current program step. All subsequent program steps--up to the first two consecutive **TERMINATE** statements--are then moved downwards, replacing the deleted program steps. (Separating programs by blocks of **TERMINATE** commands ensures that they may be edited independently.) When deleting program steps with the **KILL** command, be sure to update any **JUMP** statements' destinations that might have been shifted. They are not automatically updated.

GO	**GO**	**Gnnn<CR>**

In PROGRAM EDIT-mode, sets the program pointer to program step nnn and returns the contents of the program step to the controlling device.

Command	855K Key	String Format
FORWARD	FWD	F (no <CR> needed)
REVERSE	REV	R (no <CR> needed)

These commands, in PROGRAM EDIT-mode, move the program pointer one program step forward or backward, returning the contents of the newly current program step to the controlling device.

.P	.PRGM	.Pd<CR>

This command may be used in PROGRAM EDIT-mode or EXECUTE-mode to "teach" the 855C a sequence of motions.

After moving an actuator to a desired position by manual or JOG-mode control, enter .Pd<CR> via the controlling device. The 855C will then store in its memory (starting at the current program pointer position):

1) The drive's position as the parameter for an ABSOLUTE DESTINATION command for drive d, and

2) A MOVE command for that drive.

When these commands are executed as programmed commands, drive d will be brought to the same position it was at when the .Pd command was issued (provided the absolute origin had not been changed).

Entering 0 for the drive designator will store the current positions of all four active (non-zero velocity) actuators, followed by a MO command. In EXECUTE-mode, the current program pointer position may be queried by the .C command.

2.8.3 Special Program Statements

The 855C has several powerful statements which are useful for structuring programs.

Command	855K Key	String Format
DELAY	DLY	Dsssssss<CR>

Pauses execution of program statements for (sssssss/128) seconds.

WAIT	WAIT	Wd<CR>

Waits for drive d to reach its destination before executing the next program step.

JUMP	JUMP	Jiii-ttt<CR>

As a programmed command, causes program execution to jump to program step iii. If program step iii precedes the current program step, execution will loop ttt times; if ttt = 0, the program will loop indefinitely. If program step iii lies after the current program step, JUMP acts as an unconditional jump, and no intervening program steps will be executed. Remember when editing that the destination line number, iii, is not updated automatically when the KILL and NEW commands are used.

TERMINATE	TERM	T (no <CR> needed)

Suspends program execution, leaving the 855C in EXECUTE-mode. The string "T at nnn" (where nnn is the number of the program step containing the TERMINATE command) will be returned to the controlling device.

TERMINATE commands may be placed anywhere in a program. After termination, entering a GO command (with no step number) in EXECUTE-mode will resume execution at the line immediately following the TERMINATE command.

The 855C's volatile RAM (program steps 000-099) is cleared and filled with TERMINATE commands at power-up; any program subsequently loaded into this portion of memory will automatically have a TERMINATE command immediately following it to properly mark the end of the program. Be sure to write a TERMINATE command at the end of any program that extends into the non-volatile EEPROM memory (program steps 100-300), or subsequent program steps from partially-overwritten programs may be executed.

When program steps are inserted or deleted (with the NEW and KILL commands, respectively), the 855C will move the following block of program instructions to make room for the edit. The block is delimited by strings of two or more TERMINATE statements. It is therefore good practice to separate programs with blocks of several TERMINATE commands. Then only the desired program will be affected by edits.

2.8.4 PROGRAM RUN-mode

Program execution commences when the 855C is placed in PROGRAM RUN-mode (from EXECUTE-mode). The following instructions control the execution of a program that has been loaded into the 855C's memory.

Command	855K Key	String Format
GO	GO	Gnnn<CR>

From EXECUTE-mode, GO initiates PROGRAM RUN-mode, and program execution commences starting at program step number nnn. If nnn is omitted, program execution commences at the current program step.

For example, the GOO command begins executing program steps at the first line in memory; execution will continue until a TERMINATE command is encountered. A subsequent GO command (without a line number) would resume execution at the program step immediately following the TERMINATE command.

TERMINATE	TERM	T (no <CR> needed)

In PROGRAM RUN-mode, stops all actuator motion and terminates execution of any 855C program. May be entered via the currently active port. The 855C is placed in EXECUTE-mode, and control is returned to the currently active port.

CONTROL	CNTL	CNTL key on 855K only

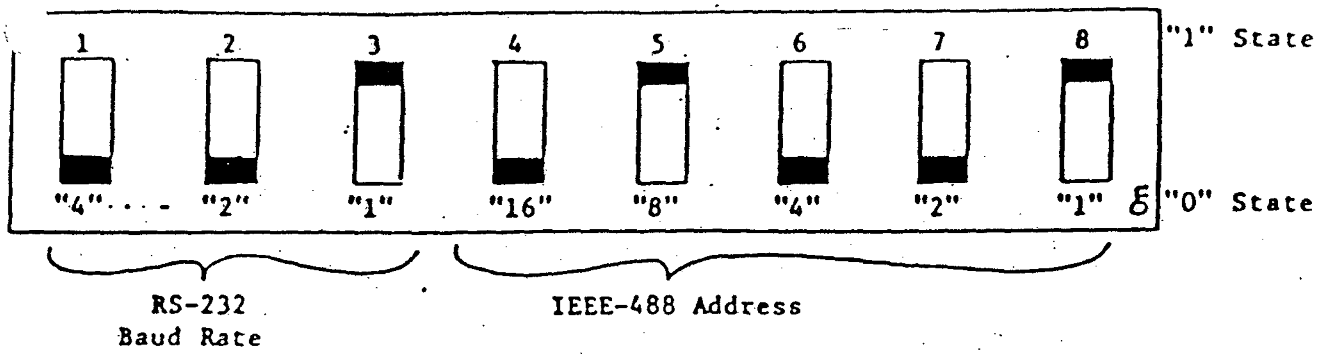
The 855K's CNTL key terminates all actuator motion, halts program execution, and assigns control of the 855C to the 855K. It may be entered by the 855K even if another device has been assigned control. There is no corresponding command string for the other communications ports. The CNTL key may also be used in EXECUTE-mode to terminate all actuator movement.

Command	855K Key	String Format
?	(NONE)	? (No <CR> needed)

? is used then the 855C is under control from the RS-232C or IEEE-488 ports to query the current position of all actuators and the program status. A ? cannot be entered via the 855K.

The 855C will respond to a ? with a string consisting of the four actuator positions and the program step pointer value at the time the ? was received. This command is not programmable.

Note that TERMINATE, CONTROL, and ? are the only commands which may be issued during a running program. None requires a carriage return. Any other commands issued during program execution are ignored and discarded.



Note: Some units may have "ON" (Ø) at the top. It will be clearly marked on the switch.

Switch Number 1	2	3	Setting	Baud Rate
		ON	0	110
		ON	1	300
		ON	2	600
		ON	3	1200
		ON	4	2400
		ON	5	4800
		ON	6	9600
		ON	7	19,200

Indicates pushed in

For the switch positions shown above, the IEEE-488 address is 9.

FIG.6- Switch Settings

3.0 INTERFACING

The 855C's RS-232C and IEEE-488 ports can be connected to a wide variety of computers, terminals, modems and other data communications and control devices. This allows complete, two-way control and data exchange between the 855C and a remote device.

These communications options offer several additional capabilities. Experimental results and procedures can be easily recorded by the remote or host computer for future reference, 855C programs can be archived, edited and downloaded by the controlling device, and the 855C's operating status and error messages may be monitored at all times and in all modes.

As discussed in Section 2.2, "Assigning Control to the 855K or External Device," only one controlling port can be "active" at a time, though all the ports may have devices connected to them at once. For example, if the RS-232C port has been assigned control, commands from other devices will be ignored, and only the RS-232C port will receive data and messages from the 855C. Control may be reassigned to another port at any time, but again only the active controlling device may do this. (The sole exception is the CNTL key on the 855K keypad, which is discussed in Section 2.8.4.)

The RS-232/IEEE-488 configuration DIP switches, on the back panel of the 855C, set the operational parameters (RS-232C baud rate and IEEE-488 device address) of the two interfaces. These switches must be set to match the 855C interface with that of the remote device. The switch settings are read only when power is first applied; any changes in the switch settings should only be made with the power off. The switch setting options are illustrated in a decal next to them on the rear panel and in Fig. 6.

3.1 Input Data Format

All input data must be entered in standard ASCII format. All alphabetical characters must be in upper case. The general command format will be a string of one or two alpha characters followed by numeric data and terminated by a carriage return, <CR>. The 855C will not accept space characters (ASCII Decimal 32, Hex 20). Inputting a space character will cause an error message to be returned to the controlling device; the command string will not be executed. Line feeds (ASCII decimal 10, Hex 0A) are ignored.

Examples:

A1-6.5<CR>	Sets Absolute Destination of drive 1 to -6.5 mm.
.2<CR>	Toggles angular/linear mode for drive 2.
V1<CR>	Queries velocity parameter for drive 1.
I0<CR>	Initiates incremental motion command for all active drives.
A1 -6.5<CR>	Invalid; no spaces are allowed.
H0<CR><LF>	Invalid; no linefeed characters are allowed.

3.2 Output Data Format

All input commands are echoed to the 855C's output buffer unless the Echo Suppression command is in effect (Section 2.6.5). The output format is the same as the input format (and so will have a carriage return at the end) but will also have a line feed character (ASCII Decimal 10, Hex OA) appended at the end, even if none was sent. In general, all output strings--including error messages and parameter data--will be terminated by a <CR><LF> sequence.

In the case of a query command input, the reponse data will be contained in a separate information block, also terminated by a <CR><LF> (carriage return-line feed) sequence. For example, if the current coordinate of drive 1 is 1.2345 mm:

Input Data	Output Data
C1<CR>	C1<CR><LF><SP><SP>1.2345<CR><LF> where <SP> is a space character (ASCII Decimal 32, Hex 20)

Other examples:

I1<CR>	I1<CR><LF>
M12.<CR>	M12.<CR><LF>

3.3 Buffers

All input and output data passing through the RS-232C and IEEE-488 ports is transferred through 1,000-character buffers. These buffers are used for short-term storage of data being transferred through the interface. This minimizes delays in operation of both the 855C and the host computer during other processing, while preventing loss of data.

The I/O buffers provide several capabilities:

1) Upload/Download of program listings

2) Stacked command operation in the EXECUTE-mode

3) Monitoring error messages

4) Queried and programmed status information

5) On-line program editing

The I/O buffers are shared between the RS-232C and IEEE-488 ports. The input and output buffers are continuously monitored and any overflow condition is indicated when it occurs.

Note that if a lengthy series of commands is sent quickly into the 855C's input buffer, it may take some time for the last commands to be executed. In particular, if the QUIT command is queued, control reassignment will be delayed until its turn comes. This is of no consequence if the guidelines discussed in Section 2.2, "Assigning Control to the 855K or External Device," are observed.

Buffer Control

Data will be lost if the input or output buffers overflow. Hardware and software schemes exist to prevent buffer overflow and data loss and are discussed in the RS-232C and IEEE-488 interface sections.

If an input buffer overflow occurs, an asterisk ("*" [ASCII Decimal 42, Hex 2A]) will be placed in the output buffer. The 855C will continue to execute the oldest commands, and new commands will be lost as long as the overflow condition exists. You may wish to instruct the host computer to suspend operations with a TERMINATE command if an asterisk is received in the 855C's output.

If an output buffer overflow occurs, the oldest data is overwritten with new data, and a "#" character (ASCII Decimal 35, Hex 23) is placed at the point in the buffer at which the overflow occured. Data transfer through the port will continue, and overwriting will stop when the overflow condition ceases to exist.

3.4 RS-232C Serial Interface Port

Baud Rate Selection and Transmission Speed

The RS-232C port provides the 855C with a flexible, standardized interface to a wide variety of controlling computers and communications devices. The baud rate may be selected by the leftmost three back-panel RS-232/IEEE-488 DIP switches for 110, 300, 600, 1200, 2400, 4800, 9600 or 19200 baud. The switch settings for the various baud rates are shown in Fig. 6 and on the back-panel decal on the 855C. The baud rate switch settings are read only upon power-up; after any change in switch settings, the 855C must be cycled on and off to implement the change.

The actual character transmission rate is limited by the time the 855C's microprocessor is available to service the communications port. Character transmission rates of up to 60 per second may be achieved under any conditions for baud rates of 600 and up. However, higher baud rates may not always achieve higher actual character transmission rates in some processor-intensive applications. Any baud rate setting may be used at any time, but the real character transfer rate must be throttled-down under these conditions.

The 855C's microprocessor is freed of motion-control tasks when all four axes' velocities are set to zero. This will maximize character transfer through its ports. This is handy for rapidly transferring programs to and from the controlling device. In this case, the 855C can transfer 120 characters per second at 1200 baud and up to 240 cps at higher baud rates.

The RS-232C port will echo all characters received (with the exception of XON and XOFF characters), unless echoes are inhibited by the ! command (Section 2.6.5). To ensure proper transfer of data at baud rates higher than 600, we recommend that commands be sent character-by-character and that the echoed character be received by the controlling device before the next character is sent to the 855C.

Bit Data Format

The 855C's RS-232C port requires characters to be formatted with:

1 Start Bit
8 Data Bits
1 Stop Bit
No Parity Bit.

Preventing Buffer Overflow: The XON/XOFF Protocol

The XON/XOFF protocol is a software convention frequently implemented in RS-232C interfacing to prevent buffer overflow and consequent data loss. The 855C requires this protocol unless special hardware handshaking provisions are devised or other provisions are made to ensure that its buffers do not fill up.

When 900 characters are stored in the 855C's input buffer, the 855C will send an XOFF character (the Control-S or DC3 character, ASCII Decimal 19, or Hex 13) through the RS-232C port (bypassing the output buffer). The controlling computer or device is expected to recognize this character and stop sending characters. As the 855C continues processing, the number of characters queued in its input buffer will drop. When there are 600 characters left in the input buffer, the 855C will send an XON character (the Control-Q or DC1 character, ASCII Decimal 17, Hex 11). This is to instruct the host computer to resume sending characters.

When the 855C output buffer is filling particularly quickly, as might happen if a series of ? commands are received (each requiring a response string of 40 characters), it may send an XOFF character to ensure that the buffer does not overflow later on. When the output buffer has worked back down to 600 characters, the 855C will send an XON character to request that input be resumed.

This protocol works the other way, too. When the 855C receives an XOFF character, it will stop sending from its output buffer and will resume transmission only when an XON character is received. These control characters are given immediate attention by the 855C within one interrupt cycle (1/128 second). They are not queued in the input buffer, nor are they echoed.

Hardware Handshaking

RS-232C signal conventions defined under the RS-232C standard are depicted in Fig. 7. A logic 1 (MARK) is indicated when the voltage at the interface (which has a total input resistance between 3 and 7 kOhms and a capacitance less than 2,500 picofarads) is more negative than -3 Volts; a logic 0 (SPACE) is indicated when the voltage is more positive than +3 V. A ± 2 V noise margin is defined to lie above each threshold. Thus, to indicate a logic 1 per the RS-232C standard, the driver must assert at least -5 V (but no more than -15 V) across the interface. To indicate a logic 0, the driver must assert at least +5 V (but no more than +15 V) across the interface. Other RS-232C conventions require that logic state voltages be maintained above the threshold until the next logic transition, that transitions always be unidirectional and that they not exceed the lesser of 1 millisecond or 4 percent of the nominal signal period. In the 855C, the signal noise margin (discriminator level) is actually set at about 1 V to facilitate simple TTL or open collector drivers at moderate baud rates.

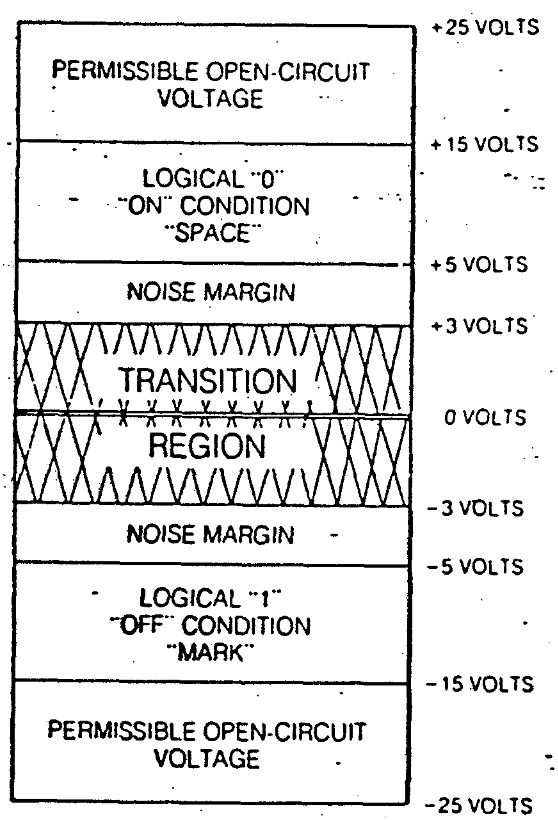


Fig. 7-
RS-232C Signal Levels

The RS-232C Connector

Cabling conventions have not been officially established for the RS-232C interface, but the 855C connector should work with most laboratory computers and terminals without recabling. Note that this port requires a CTS input handshake line and is configured as a DTE device.

Pin 1 is chassis ground.

Pin 7 is signal ground.

Pin 2 (output) is the "Transmitted Data" (TxD) line.

Pin 3 (input) is the "Received Data" (RxD) line.

Pin 4 (output) is the "Ready to Send" (RTS) line. This is set high at 855C power-up and remains high at all times. It merely indicates that the 855C is ready for operation and is not implemented for handshaking. It may be connected to the controlling device's DSR and DSD inputs if that device requires those lines to be high to initiate data transfer.

Pin 5 (input) is the "Clear to Send" (CTS) line. It must be high for the 855C to transmit. If CTS drops low, the 855C will immediately cease transmission. This line is normally connected to the controlling device's RTS line, and may be exploited in hardware handshaking procedures.

Pin 20 (output) is the "Data Terminal Ready" (DTR) line. This line reflects the 855C's input buffer status. It is high when the buffer has room for input and goes low when the buffer is full. It should be attached to the CTS line of the host computer to make the host stop sending characters. (This hookup is similar to that used by some printers.)

Pin 6 (output) is defined in the RS-232C standard as the "Data Set Ready" line. It is not implemented on the 855C and has no effect on operation.

Pin 8 (input) is defined as the "Data Carrier Detect" line. It is not implemented on the 855C and has no effect.

Sample Cable Configurations

Here are some examples of the many ways the RS-232C interface may be configured. Where two pins are to be connected together, they are shown with a "+".

1) Three-wire hookup

This configuration relies on XON-XOFF for handshake. Only three signal wires (plus ground) are required.

855C Pins	Host Device (DTE)	Comments
1	1	Chassis GND
7	7	Signal GND
2 (TxD)	3 (RxD)	Data from 855C
3 (RxD)	2 (TxD)	Data to 855C
Connect at each device (but do not cable between devices):		
4+5 (RTS+CTS)	4+5 (RTS+CTS)	Enable transmitter

2) Full Hardware Handshake: DTE to DTE

855C Pins	Host Device (DTE)	Comments
1	1	Chassis GND
7	7	Signal GND
2 (TxD)	3 (RxD)	Data from 855C
3 (RxD)	2 (TxD)	Data to 855C
4 (RTS)	6+8 (DSR+DCD)	Shows 855C is on
5 (CTS)	4 (RTS)	Stops 855C output
20 (DTR)	5 (CTS)	Stops host output

3) Full Hardware Handshake: DTE to DCE

855C Pins	Host Device (DCE)	Comments
1	1	Chassis GND
7	7	Signal GND
2 (TxD)	2 (TxD)	Data from 855C
3 (RxD)	3 (RxD)	Data to 855C
4 (RTS)	20 (DTR)	Shows 855C is on
5 (CTS)	5 (CTS)	Stops 855C output
20 (DTR)	4 (RTS)	Stops host output

3.5 IEEE-488 Port

The parallel communications port on the 855C is compatible with computers and other devices conforming to the IEEE-488-1978 interface standard. This interface is also known as GPIB (General Purpose Interface Bus) and HPIB (Hewlett-Packard Interface Bus). Interfacing to an IEEE-488 bus is very easy since the standard establishes connector pinouts, signal levels, and so on. The only hardware set-up procedure required is setting the device address, which in the 855C is determined by the RS-232/IEEE-488 DIP switches on its back panel.

Successful interfacing with any bus demands a thorough understanding of the bus structure as well as its handshaking protocol, interrupt criteria, data format requirements, transfer commands, etcetera-- for both devices being interfaced. We strongly recommend a review and understanding of the IEEE-488-1978 specifications and operation of the 855C and remote controlling device.

IEEE-488 Bus Operation

The 855C is always set up as a talker-listener; it is incapable of acting as the IEEE-488 bus controller device. It is assumed that a reasonably intelligent bus controller will be used to communicate with it.

Data Transfer- Input

To send commands to the 855C, it is only necessary to address it as a listener and enter commands just as they would be entered by hand with the 855K Keypad/Display. The data can be entered character-by-character or in the form of a data string containing the complete instruction.

The command string format must conform to the rules discussed in Section 3.1. Specifically:

--Upper case alphabetical characters are required, and

--The string must be terminated by a carriage return <CR> (ASCII Decimal 13, Hex OD).

Data Transfer- Output

To receive data from the 855C's IEEE-488 port, it should be addressed as a talker with the host computer (bus controller) set as listener. Data is normally sent from the 855C in the form of a string of characters.

An important consideration in controlling data transfer through the IEEE-488 bus is the transfer terminate condition. The bus controller must be told when the talker has no more information to send so it can end the transfer operation and go about its business. The 855C does this in two ways: by sending a <CR><LF> character string and by asserting the IEEE-488 EOI line.

<CR><LF> Transfer Termination

Messages, query responses and command echoes are output from the 855C IEEE-488 port in string form, terminated by a carriage return/line feed character sequence. Most computers and bus controllers are set up to terminate input transfers on receipt of the <CR><LF> sequence.

EOI Transfer Termination

Some bus controllers have the capability of terminating input data transfer when the End-Or-Identify (EOI) hardware line is asserted. The 855C will assert EOI when the last character is sent, which will always be a line feed.

Buffer Control

All data passing into and out of the 855C's IEEE-488 port is transferred through 1,000-character input and output buffers which serve as storage until the data can be processed by the 855C or bus controller. If the buffers are allowed to overflow, data will be lost. The indications and consequences of buffer overflow are discussed in Section 3.3. Buffer overflow can be avoided by careful use of the IEEE-488 bus' information flow features.

Timeouts, Interrupts, and Polling

To efficiently control the flow of information on the IEEE-488 bus, the bus controller must somehow know when and which downline devices have information to send it. The IEEE-bus has three features which are useful for controlling information flow.

Timeouts

A brute-force way to find out if the 855C has information ready to send is to simply attempt an input transfer to the bus controller. If there is data in the 855C output buffer, it will be transferred out. However, if no data is in the buffer, the bus controller will hang unless precautions are taken.

To avoid this, most bus controllers have a TIMEOUT feature that aborts the transfer operation if it is not terminated within a specified amount of time. This allows the bus controller to resume other duties rather than wait to receive data that doesn't exist. We recommend use of the TIMEOUT feature even when more sophisticated schemes are implemented to terminate transfer of data to the 855C.

Interrupts

The IEEE-488 bus has a dedicated Service ReQuest (SRQ) line which may be used by any device on the bus to signal to the bus controller that it needs service. The 855C uses the SRQ line to signal that information is ready in its output buffer. The bus controller can then respond and transfer the data out. The 855C's SRQ will remain active until reset by a Serial Poll performed on the 855C by the bus controller.

SRQ is asserted whevever one of the following conditions occur:

--Arrival of the tenth character in the output buffer,

--Arrival of a linefeed character <LF> in the output buffer, or

--Any character is in the output buffer and the 855C is not set as a talker.

A suggested control sequence for using the SRQ line to transfer data from the 855C is:

1) Have the bus controller monitor the SRQ line by looping or by a preconfigured or programmed interrupt.

2) When SRQ is asserted, enter the data from the 855C to the bus controller using its particular data transfer procedure.

3) Serial Poll the the 855C to reset its SRQ line.

4) Continue monitoring the SRQ line, watching for the next SRQ assertion.

Polling

A third feature of the IEEE-488 interface is the ability of the bus controller to poll a specific device on the bus and determine its status. This is useful to determine which device has data ready to send when more than one 855C is used on the IEEE-488 bus. When a Serial Poll is issued to the 855C, a status byte is returned to the bus controller. Two bits in the byte represent the SRQ line and output buffer status:

--Bit 6 is 1 when the SRQ line is asserted, 0 when the SRQ line is not active.

--Bit 7 is 1 when the 855C's output buffer is empty, 0 when the output buffer contains characters.

All other bits in the byte may be ignored. Note that the Serial Poll will reset the 855C's SRQ line if the conditions for SRQ assertion no longer exist.

4.0 SAMPLE PROGRAMS

Programming the 855C is simple: if you wish, you can simply key in commands just as you would in EXECUTE-mode and have them execute automatically. Programs can do a lot more than just parrot keystrokes, though. The 855C's looping and other program structure capabilities, its simultaneous control of up to four actuators, plus its ability to work closely with sophisticated computers and other data devices, all mean it can perform very complicated precision movements.

Here we present a few sample programs to illustrate the 855C's command mnemonics and typical interfacing with popular computers.

4.1 855C Sample Program: Raster Scan

This program uses two linear actuators to perform a simple raster scan. Drive 1 is the X-axis actuator, and drive 2 drives the Y-axis. The scan is 1.0 mm square and is composed of 50 X-axis sweeps incremented by 0.01 mm. The listing is shown in the approximate format that the 855C would use in listing it to a controlling device. If you key it into the 855C, be sure to remove any imbedded spaces.

Step/Instruction/Data				Comment
0	C	0	.0000	Set coordinates of all drives to 0
1	V	1	.0500	Set drive 1 velocity to 0.05 mm/second
2	V	2	.4000	Set drive 2 velocity to 0.4 mm/second
3	A	1	1.000	Set absolute destination of 1 to 1 mm
4	S	2	.0100	Set drive 2 step size to 0.01 mm
5	M	1		Move 1 to its absolute destination
6	W	1		Wait for 1 to finish moving
7	I	2		Increment 2 by its step size
8	W	2		Wait for 2 to finish moving
9	H	1		Send 1 home (to its absolute origin)
10	W	1		Wait for 1 to finish moving
11	I	2		Increment 2 by its step size
12	W	2		Wait for 2 to finish moving
13	J	5	49	Jump to step 5, repeat 49 times
14	T			Terminate program

4.2 HP-85 Sample Programs

Following are two simple programs for the Hewlett-Packard HP-85 microcomputer. Both programs are written in the HP-85's built-in ROM HP-BASIC. One program uses the HP-85 RS-232C serial port (and the supporting BASIC routines built into the HP-85's Serial Port Adaptor ROM); the other uses the computer's IEEE-488 interface (and supporting BASIC routines). (The supporting BASIC routines for each port come with the HP-85's port interfaces. They are not valid BASIC statements for HP-85 computers that aren't equipped with the interfaces.) Both programs input commands to the 855C and read its responses.

RS-232C Serial Port Program

This simple program takes keyboard input and formats it for the 855C with standard BASIC commands. A simple, 3-wire interface configuration is assumed. (The 855C's character echo should be suppressed when in JOG-mode, since no string terminating character is sent after JOG-mode F and R commands, as this will cause timeout errors.)

```
 10 ! 3WIRE SERIAL -JOG
 20 ! 855
 30 CLEAR
 40 RESET 10
 50 CONTROL 10,3 ; 7 ! 600 BAUD
 60 CONTROL 10,4 ; 3 ! PROTOCOL
 70 DIM P$[1000] ! ENTER STRING
 80 DIM A$[50] ! O/P STRING
 90 C1$=CHR$(13)
100 ENABLE INTR 10;16
110 ON INTR 10 GOSUB 340
120 SET TIMEOUT 10;300
130 ON TIMEOUT 10 GOTO 390
140 CLEAR
150 ! KEYBOARD INPUT ROUTINE

160 DISP "ENTER COMMAND"
170 INPUT A$
180 CLEAR
190 ! INPUT COMMAND FILTER
200 IF A$="01" THEN 270
210 IF A$="" THEN A$=CHR$(13)
220 IF A$="F" THEN 270
230 IF A$="R" THEN 270
240 IF LEN(A$)=3 AND A$[1,1]="F"
     THEN 270
250 IF LEN(A$)=3 AND A$[1,1]="R"
     THEN 270
260 A$=A$&C1$
270 OUTPUT 10 USING "#,K" ; A$
280 ! BEGIN IDLE LOOP
290 KEY LABEL
300 ON KEY# 1,"COMMAND" GOTO 140
310 FOR X=1 TO 10
320 NEXT X
330 GOTO 290
340 ! INTR SERVICE ROUTINE
350 P$=""
360 ENTER 10 ; P$
370 DISP P$
380 ENABLE INTR 10;16 @ RETURN
390 ! TIMEOUT SERVICE ROUTINE
400 RESET 10
410 CONTROL 10,3 ; 7
420 CONTROL 10,4 ; 3
430 ENABLE INTR 10;16
440 GOTO 290
450 END
```

Lines 30-130 set up the RS-232C interface, enable the port interrupt, set up the strings, and establish the timeout parameters.

Lines 150-260 take the command string from the computer keyboard and prepare it for transfer to the 855C.

Line 270 outputs the string to the 855C, formatted with the linefeed character suppressed.

Lines 280-330 are an idle loop which may be interrupted for input from the 855C.

Lines 340-380 service the port interrupt to bring 855C information into the computer.

Lines 390-440 service the timeout interrupt if required.

IEEE-488 Port Program

This program uses the IEEE-488 SRQ interrupt to initiate data transfer from the 855C; the response string is displayed on the HP-85's CRT screen. Note that since the IEEE-488 bus allows several devices on-line simultaneously, the 855C must be addressed specifically at its address (in this case, device address 9) on the IEEE-488 port (HP-85 port 7). This is why all ENTERs and OUTPUTs are to <u>709</u>.

```
 10 ! 855 SIMPLE PROGRAM
 20 ! HPIB
 30 DIM A$[15] ! O/P STRING
 40 DIM P$[1000] ! i/P STRING
 50 RESET 7
 60 ENABLE INTR 7;8 ! SRQ
 70 ON INTR 7 GOSUB 290
 80 ! KEYBOARD INPUT ROUTINE
 90 CLEAR
100 A$=""
110 DISP "ENTER COMMAND"
120 INPUT A$
130 CLEAR
140 ! INPUT COMMAND FILTER
150 IF A$="02" THEN 220
160 IF A$="" THEN A$=CHR$(13)
170 IF A$="F" THEN 220
180 IF A$="R" THEN 220
190 IF LEN(A$)=3 AND A$[1,1]="F"
     THEN 220
200 IF LEN(A$)=3 AND H$[1,1]="R"
     THEN 220
210 A$=A$&CHR$(13)
220 OUTPUT 709 USING "#,K" ; A$
230 ! BEGIN IDLE LOOP
240 KEY LABEL
250 ON KEY# 1,"COMMAND" GOTO 80
260 FOR X=1 TO 10
270 NEXT X
280 GOTO 240
290 ! SRQ SERVICE ROUTINE
300 STATUS 7,1 ; A
310 ENTER 709 USING "#%,#%K" ; P
    $
320 ! DISP FILTER ROUTINE
330 E=LEN(P$)
340 FOR Y=1 TO E
350 IF P$[Y,Y]=CHR$(10) THEN DIS
    P @ GOTO 370
360 DISP P$[Y,Y];
370 NEXT Y
380 S=SPOLL(709)
390 ENABLE INTR 7;8 @ RETURN
400 END
```

<u>Lines 30-70</u> set up the string variables and interrupts and initialize the interface.

<u>Lines 80-210</u> take the command string from the computer keyboard and prepare it for output (A$).

<u>Line 220</u> outputs A$ to the 855C.

<u>Lines 230-280</u> are an idle loop which can be interrupted by SRQ.

<u>Lines 290-390</u> are the SRQ service routine, which enters the response from the 855C, removes the linefeed characters, and displays the string (P$) on the CRT screen.

<u>Line 380</u> resets the SRQ line by performing a Serial Poll on the 855C.

5.0 APPENDICES

5.1 Error Messages

The 855C is designed to provide information to the controlling device whenever a command results in an error condition or when an axis exceeds its range of travel. These error messages are are formatted by the 855C in the same manner as query responses and character echoes.

"**axis d** position error"

If the limit is not sensed or if the actuator fails to track the controller position by an error exceeding 20,000 counts, the 855C will report this error message. This condition can arise if the actuator is overloaded. The velocity setting for the actuator will be zeroed to protect the actuator.

"**axis d** fwd hard limit"
"**axis d** rev hard limit"
"**axis d** fwd soft limit"
"**axis d** rev soft limit"

The actuator has moved or tried to move past either its physical range of travel or the absolute limit established by the LIMIT command or default.

"limit not set--bad limit"

An attempt was made to set an invalid soft limit with the LIMIT command. The limit setting was not changed.

"pos. not set--out of limit"

An attempt was made to set the current position of the actuator with the CURRENT COORDINATE command outside the limits established by the LIMIT command or by default. The current coordinate of the actuator was not changed.

"all axes are off"

You pressed **.Pn**, but all axes specified had zero velocities.

5.2 Table of Commands

The following table reviews the 855C's commands, their 855K keypad labels and their ASCII string format. It also indicates whether each command is programmable and in what modes it is valid. Lastly, the reference Sections are given in which each command is discussed.

The key for modes is: E=EXECUTE MODE, J=JOG-mode, R=PROGRAM RUN-mode, P=PROGRAM EDIT-mode. On the 855K Keypad/Display, the ENTR key serves as a carriage return.

Bracketed data (e.g., {nn.nnn}) is optional. For most commands, omission of data acts to query the 855C for the current settings of the parameter. Exceptions are the JOG-mode commands FWD and REV, the EXECUTE-mode command GO, the PROGRAM EDIT-mode command KILL, and the MOVE command. Each of these exceptions are discussed in their referenced Sections. Also refer to Section 2.3 for numeric formatting conventions.

The STATUS QUERY and ECHO SUPPRESSION commands are only available to the RS-232C and IEEE-488 interface ports. There is no command string corresponding to the 855K CNTL command.

Note that the mnemonic letters O, U, W, X, Y and Z are not used for any purpose by the 855C.

Command	Key	String	Prog?	Mode	Ref.
ABSOLUTE DESTINATION	ABS	Ad{±nn.nnn}<CR>	Y	E,P	2.6.1, 2.6.3
BACKLASH	BKLH	Bd{±.nnnn}<CR>	Y	E,P	2.6.2, 2.6.3
CURRENT COORDINATE	CRNT	Cd{±nn.nnnn}<CR>	Y	E,P	2.6.1, 2.6.3
DELAY	DLY	Dssssss<CR>	Y	E,P	2.8.3
EXECUTE MODE	EXEC MODE	E<CR>	N	P	2.8.1
FORWARD	FWD	Fd{±nn.nnnn}<CR>	N	E,P,J	2.7, 2.8.2, 2.6.1
GO	GO	Gnnn<CR>	N	E,P	2.8.2, 2.8.4

Command	Key	String	Prog?	Mode	Ref.
HOME	HOME	Hd<CR>	Y	E,P	2.6.4
INCREMENT	INC	Id<CR>	Y	E,P	2.6.4
JUMP	JUMP	Jiii-ttt	Y	P	2.8.3
KILL	KILL	K{nnn-rrr}	N	P	2.8.2
LIMIT	LIMIT	Ldr{±nn.nnnn}<CR>	Y	E,P	2.6.1, 2.6.3
MOVE	MOVE	Md{±nn.nnnn}<CR>	Y	E,P	2.6.4
NEW	NEW	N<CR>	N	P	2.8.2
PROGRAM MODE	PRGM MODE	P<CR>	N	E	2.8.1
QUIT	QUIT	Qp<CR>	Y	E,P	2.2
REVERSE	REV	Rd{±nn.nnnn}<CR>	N	E,P,J	2.7, 2.8.2, 2.6.1
STEP	STEP	Sd{±nn.nnnn}<CR>	Y	E,P	2.6.1, 2.6.3
TERMINATE	TERM	T (no <CR>)	Y	E,P,R	2.8.2, 2.8.3, 2.6.4
VELOCITY	VEL	Vd{n.nnnn}<CR>	Y	E,P	2.6.1, 2.6.3
STATUS QUERY	(NONE)	? (no <CR>)	N	R	2.8.4
ECHO SUPPRESS	(NONE)	!<CR>	N	E	2.6.5
LINEAR/ ANGULAR	.	.<CR>	Y	E,P	2.4
STEP POINTER	.CRNT	.C<CR>	Y	E,P	2.6.3
TEACH POSITION	.PRGM	.Pd<CR>	N	E,P	2.8.2
KEYPAD CONTROL	CNTL	(855K only)	N	E,R	2.2, 2.8.4

5.3 List of Figures

5.4 System Schematics and Wiring Diagrams

CPU PC BOARD DESCRIPTION

TOP ASSEMBLY-- 855C/K

BLOCK DIAGRAM-- 855C/K

WIRING DIAGRAM-- 855C

ENCLOSURE ASSEMBLY-- 855C

CPU CARD ASSEMBLY

CPU CARD WIRING ASSEMBLY

RSTD-ENCODE 4 PC BOARD ASSEMBLY

RSTD-ENCODE 4 SCHEMATIC

RSTD-SERIAL PC BOARD ASSEMBLY

RSTD SERIAL SCHEMATIC

RSTD-DAC 4 PC BOARD ASSEMBLY

RSTD-DAC 4 SCHEMATIC

ACTUATOR PC BOARD ASSEMBLY

ACTUATOR BOARD SCHEMATIC

POWER UP/DN PC BOARD ASSEMBLY

JOG SWITCH INTERFACE PCB ASSEMBLY

REMOTE TERMINAL CONNECTOR PCB ASSEMBLY

HAND-HELD TERMINAL (855K) PCB ASSEMBLY

HAND-HELD TERMINAL (855K) SCHEMATIC