# Model 273A
# Potentiostat/Galvanostat

## Command Set Handbook

221267C / 0103
Printed in USA

## Advanced Measurement Technology, Inc.

a/k/a Princeton Applied Research, a subsidiary of AMETEK®, Inc.

# WARRANTY

Princeton Applied Research* warrants each instrument of its own manufacture to be free of defects in material and workmanship. Obligations under this Warranty shall be limited to replacing, repairing or giving credit for the purchase price, at our option, of any instrument returned, shipment prepaid, to our Service Department for that purpose within ONE year of delivery to the original purchaser, provided prior authorization for such return has been given by an authorized representative of Princeton Applied Research.

This Warranty shall not apply to any instrument, which our inspection shall disclose to our satisfaction, to have become defective or unworkable due to abuse, mishandling, misuse, accident, alteration, negligence, improper installation, or other causes beyond our control. This Warranty shall not apply to any instrument or component not manufactured by Princeton Applied Research. When products manufactured by others are included in Princeton Applied Research equipment, the original manufacturer's warranty is extended to Princeton Applied Research customers.

Princeton Applied Research reserves the right to make changes in design at any time without incurring any obligation to install same on units previously purchased.

THERE ARE NO WARRANTIES THAT EXTEND BEYOND THE DESCRIPTION ON THE FACE HEREOF. THIS WARRANTY IS IN LIEU OF, AND EXCLUDES ANY AND ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESSED, IMPLIED OR STATUTORY, INCLUDING MERCHANTABILITY AND FITNESS, AS WELL AS ANY AND ALL OTHER OBLIGATIONS OR LIABILITIES OF PRINCETON APPLIED RESEARCH, INCLUDING, BUT NOT LIMITED TO, SPECIAL OR CONSEQUENTIAL DAMAGES. NO PERSON, FIRM OR CORPORATION IS AUTHORIZED TO ASSUME FOR PRINCETON APPLIED RESEARCH ANY ADDITIONAL OBLIGATION OR LIABILITY NOT EXPRESSLY PROVIDED FOR HEREIN EXCEPT IN WRITING DULY EXECUTED BY AN OFFICER OF PRINCETON APPLIED RESEARCH.

### SHOULD YOUR EQUIPMENT REQUIRE SERVICE

A. Contact the Customer Service Department (865-482-4411) or your local representative to discuss the problem. In many cases it will be possible to expedite servicing by localizing the problem.

B. If it is necessary to send any equipment back for service, we need the following information.

1. Model number and serial number.
2. Your name (instrument user).
3. Your address.
4. Address to which the instrument should be returned.
5. Your telephone number and extension.
6. Symptoms (in detail, including control settings).
7. Your purchase order number for repair charges (does not apply to repairs in warranty).
8. Shipping instructions (if you wish to authorize shipment by any method other than normal surface transportation).

C. U.S. CUSTOMERS — Ship the equipment being returned to:

Advanced Measurement Technology, Inc.
801 S. Illinois Avenue
Oak Ridge, TN 37831
ATTN: Customer Service

PHONE: 865-482-4411
FAX: 865-483-2133

D. CUSTOMERS OUTSIDE OF U.S.A. — To avoid delay in customs clearance of equipment being returned, please contact the factory or the nearest factory distributor for complete shipping information.

---

# TABLE OF CONTENTS

# Safety Instructions and Symbols

This manual contains up to three levels of safety instructions that must be observed in order to avoid personal injury and/or damage to equipment or other property. These are:

**DANGER** Indicates a hazard that could result in death or serious bodily harm if the safety instruction is not observed.

**WARNING** Indicates a hazard that could result in bodily harm if the safety instruction is not observed.

**CAUTION** Indicates a hazard that could result in property damage if the safety instruction is not observed.

Please read all safety instructions carefully and make sure you understand them fully before attempting to use this product.

# Cleaning Instructions

| | |
|---|---|
| **WARNING** | Using this instrument in a manner not specified by the manufacturer may impair the protection provided by the instrument. |

To clean the instrument exterior:

- Unplug the instrument from all voltage sources.
- Remove loose dust on the outside of the instrument with a lint-free cloth.
- Remove remaining dirt with a lint-free cloth dampened in a general-purpose detergent and water solution. Do not use abrasive cleaners.

| | |
|---|---|
| **CAUTION** | To prevent moisture inside of the instrument during external cleaning, use only enough liquid to dampen the cloth or applicator. |

- Allow the instrument to dry before reconnecting the power cord.

# 1. INTRODUCTION

## 1.1. Controlling the Model 273A

The Model 273A Potentiostat/Galvanostat can be operated either directly from its front panel or remotely from a personal computer or workstation. Instructions for operating the unit as a stand-alone instrument controlled from the front panel are given in the separately bound *Model 273A Instruction Manual*. Details of its physical and electrical characteristics are also described in the *Instruction Manual*. This *Command Handbook* explains how to operate the Model 273A remotely from the computer of your choice via either the RS-232C or GPIB (IEEE-488) interface port.

By interfacing the Model 273A to an external computer, complete remote control of the instrument is readily accomplished using the Electrochemical Command Set, a group of over 100 mnemonic software statements specifically developed for electrochemical measurements. These commands place unprecedented flexibility in the hands of the electrochemist. They provide:

- Access to all front-panel functions.
- Control of all timing functions.
- Application of pulse and staircase waveforms
- Automatic acquisition of data with or without current auto-ranging.
- Data averaging in real-time.
- Internal storage and arithmetic data manipulation.

## 1.2. About this Manual

This *Command Handbook* is organized into two chapters and seven appendices. Chapter 1 explains how the Model 273A can be controlled remotely from your computer via the RS-232C and GPIB data links, and discusses the different kinds of commands that can be used. Chapter 2 describes each command in detail.

The quickest way to find information about any command, however, is by first referring to the condensed command descriptions in either the Topical Command Index or in the Alphabetical Command Index, both of which follow the Appendices at the end of the manual. Then, if you need more detailed information, go to the complete discussion of that command in Chapter 2. The summary of each command in the Command Indices ends in a reference to the page in Chapter 2 that gives more complete information about the command.

The Topical Command Index and the Alphabetical Command Index provide two different ways to find commands. In the Topical Index, the commands are functionally grouped as in Chapter 2. They are alphabetically arranged in the Alphabetical Command Index.

Detailed explanations of GPIB and RS-232C communications, such as pinouts, rear-panel switch settings, communications protocols, and some useful communications routines, are also located in the appendices.

Appendix A discusses interfacing and communications protocols for both RS-232C and GPIB (IEEE-488) data links. It describes the nature and format of the signals passing between the Model 273A and the computer controlling it.

Appendix B contains a detailed discussion of the GPIB bus standard and its use on the Model 273A. It describes each signal and explains the GPIB switch assembly settings. It also includes a section on troubleshooting GPIB communications problems. A simple static interface program that enables the Model 273A to be controlled from a host computer via the GPIB port is listed in Appendix C.

The RS-232C computer interface is described in detail in Appendix D. This appendix explains the differences between the GPIB and RS-232C interfaces, describes the RS-232C signals, and shows how to set the switches in the RS-232C switch assembly. It also includes a section on troubleshooting RS-232C communications problems.

Appendix E comprises an application note on waveform programming, one of the most useful functions of the Model 273A but perhaps the most complex to understand. A list of the ASCII character codes is provided in Appendix F.

## 1.3. Status Indication

The status of remote-controlled operations is shown by the INTERFACE group of five indicator lamps on the lower left side of the front panel on the Model 273A. Also part of the INTERFACE group is the LOCAL pushbutton, which can be used to switch to front-panel control from remote control via the GPIB interface. Section 5.2E of the *Model 273A Instruction Manual* describes the functions of the INTERFACE indicators and pushbutton.

## 1.4. Polarity Convention

The Model 273A follows the American polarity convention and the Model 273A display indications are consistent with that convention. Positive current is cathodic, that is, a current is defined as positive if reduction is taking place. Negative current is anodic, that is, a current is defined as negative if oxidation is taking place. If the working electrode is driven positive with respect to the equilibrium potential, the resulting current is anodic. If the electrode is driven negative with respect to the equilibrium potential, the resulting current is cathodic. In complex electrochemical systems, there may be more than one equilibrium system. Where this is the case, either polarity with respect to the equilibrium potential could give rise to anodic or cathodic current, according to the system's characteristics.

## 1.5. Command Categories

There are three classes of commands used in controlling the Model 273A with an external computer:

1. *Communications commands*: Communications commands are used to pass information between the host computer and the Model 273A. For example, the IEEE-488 (GPIB) port requires the use of a set of standard communication messages to accomplish certain control actions. Appendices B and D provide GPIB and RS-232C communications information in detail.

2. *Programming commands*: Programming commands are the language commands used to instruct the computer to perform the desired action. The GPIB Static Interface Routines given in Appendix C are provided as useful examples.

3. *Model 273A commands*: These commands are sent to the Model 273A from the host computer. When the Model 273A receives one of these commands, it performs the function defined for it. The Model 273A recognizes more than one hundred different commands, allowing highly sophisticated experiments to be programmed. All are discussed in Chapter 2.

The *Instructional Command Set* listed in Table 1-1 is a subset of the Model 273A commands. These commands permit simple experiments to be readily conducted. They are particularly useful for gaining an understanding of how the Model 273A works, and for getting some operating experience. Their utility in complex measurements is limited. Do not attempt to use these commands solely from the information in Table 1-1. Instead, look up the full description of a command to ascertain how it works prior to using it. Some of these commands interfere with one another, or with other commands in the general command set. These interactions must be understood and avoided for successful measurements.

There are five kinds of Model 273A commands, as follows.

**ACTION:** A command that directly causes an event to happen. An example is DCL, which restores the default parameter values.

| COMMAND | FUNCTION |
|---|---|
| SETE | Set Appl. Potential (CONTROL E MODE) |
| SETI | Set Appl. Current (CONTROL I MODE) |
| READE | Read Cell Potential |
| READI | Read Cell Current |
| I/E | I/E Converter Range |
| CELL | Cell Switch ON/OFF |
| FLT | Filter IN/OUT |
| OVER | Read Overload Status |
| MODE | Sets Operating Mode |
| ID | Read Identification Number |
| VER | Read Firmware Version Number |
| ERR | Read Error Status |

**Table 1-1. Instructional Command Set**

**READ:** A command that causes one or more values to be read and reported to the host computer. An example is VER n, which requests the Model 273A to report its software version to the host computer.

**SET:** A command that sets a parameter value. An example is VERTEX n1 n2, which sets a vertex location and the modulation level at that vertex.

**SET/READ:** A command that, if sent with an operand, sets a parameter value, and if sent without an operand, requests the Model 273A to report the parameter value in effect to the host computer. An example is FLT n, which sets or reads the filter status.

**CONTROL:** A command used to set up loops and user-defined operations. An example is

DO 10;READE;LOOP

which executes READE ten times.

**ACTION READ:** A command that initiates some action and then takes a reading. An example is AS n, which initiates a single auto-range followed by reporting the resulting range value to the host computer.

## 1.6. Notation Conventions

The following conventions are used in the command descriptions:

- "n" is used to denote a numeric parameter.
- "n1...nx" is used denote multiple parameters (x is an integer larger than 1).
- Default parameter values are enclosed by brackets, [ ].

# 2. MODEL 273A COMMANDS

## 2.1. Organization of this Chapter

The quickest way to find information about a command is by first referring to the condensed command descriptions in either the Topical or Alphabetical Command Indices that precede the main index at the end of the book. Then, if you need more detailed information, go to the complete discussion of that command in this chapter. The summary of each command in the Command Indices includes a reference to the page in this chapter that gives more complete information about the command.

Commands in this chapter are organized in the following groups. In many cases, more detailed discussions of the functions of each group of commands are presented at the beginning of each section describing a group.

*Setting commands* (Section 2.2) set or report the setting of parameters used internally by the instrument, such as current range.

*Memory Partitioning commands* (Section 2.3) are used to optimally partition available memory to fit the number and length of the curves. This section also includes *Curve Designation commands* that designate which curves are to be used.

*Bias commands* (Section 2.4) set the level of the fixed bias added to the potential or current applied to the cell.

*Modulation commands* (Section 2.5) are used to modulate the potential applied to the cell during each sweep. For example, ramp modulation can be used so that a different potential is calculated and applied for each data point. Other waveforms can be applied; a detailed discussion is presented in Section 2.5.

*Data Acquisition Control commands* (Section 2.6) are used to initialize and control curve acquisition.

*Sampling and Sweep Control commands* (Section 2.7) are used to control how data is collected. They also control the number of sweeps in an acquisition and the kind of sweep averaging, if any, to be used.

*Uncompensated Resistance commands* (Section 2.8) set or control the measurement of the uncompensated resistance.

*Current Integration commands* (Section 2.9) are used to control the IR integrator of the Model 273A.

*Model 273A/92 Electrochemical Impedance Interface Option commands* (Section 2.10) are additional commands available only with the Model 273A/92 *Electrochemical Impedance Interface* Option. This option is required for use with the AC Impedance software.

*Data Acquisition Monitoring commands* (Section 2.11) are used to report the status of a curve in progress or the current value of a sampled parameter.

*Curve Processing and Curve Data Transfer commands* (Section 2.12) are used to transfer curve data between the Model 273A and the host computer, and for copying data from one curve to another. They also perform various kinds of processing on stored curves.

*Communications Control and Status commands* (Section 2.13). The Communications Control commands specify the delimiter to be used between two numbers in a transmission, and explain how to cause a service request.

Status commands request the Model 273A to report errors, overload, and whether operations have been completed.

*Miscellaneous Commands* (Section 2.14) include:

*Cyclic Voltammetry commands* are used to program the Model 273A for cyclic staircase voltammetry.

*System Identification commands* cause the Model 273A to report its firmware version, model number, and whether an option board is installed.

*Control Structure commands* are used to set up multiple commands, set up USR functions, and interrupt command execution.

*Panel/Display commands* have the same effect as adjusting front-panel controls when a host computer is not connected.

*Auxiliary Interface commands* (Section 2.15) control signals placed on the rear-panel Auxiliary Interface connector, including those that control the Model 303A. Others control the pen relay contacts.

## 2.2. Setting Commands

**I/E n [-3]:** A SET/READ command, I/E (CURRENT TO VOLTAGE CONVERTER) sets the full-scale current range. If "n" is omitted, I/E requests that the Model 273A report the selected current range. The codes are:

| n | CURRENT RANGE |
|---|---|
| 0 | 1 A |
| -1 | 100 mA |
| -2 | 10 mA |
| -3 | 1 mA |
| -4 | 100 µA |
| -5 | 10 µA |
| -6 | 1 µA |
| -7 | 100 nA |

**Note:** A cell current equal to the selected range will give 1 V at the output of the I/E converter and at the I MONITOR and OUTPUT connectors.

**AS n:** An ACTION READ command, AS (AUTO-SENSITIVITY) causes the Model 273A to seek that current range which will result in an output between 15% and 190% of full scale, and to respond with the number that indicates that range. This is done immediately upon receipt of the command. The codes signifying the current range are the same as those listed above for the I/E command. If the signal is too large to achieve this objective, the Sensitivity will be set to the 1 A range, and "n" will be 1000. If the signal is too small to achieve this objective, the 100 nA range is selected and, again, "n" is 1000. "1000" will also be returned and no auto-range occur if:

1. A Curve Acquisition is in progress.

2. The Model 273A is in the GALVANOSTAT mode. THE AUTOSENSITIVITY FUNCTION CAN BE ACTIVATED IN THE POTENTIOSTAT MODE ONLY.

3. Only the "E" signal is being sampled.

**Example:** The Model 273A is set to the 100 A range with a current of 5 µA when the AS command is applied. On receipt of this command, the Model 273A selects the 10 µA range, on which the 5 µA current represents 50%. In addition, the number "-5" is sent to the computer to indicate that the 10 µA range has been selected.

**Note:** Full scale at the output of the analog-to-digital converter is 2000. In terms of the converter output range, the Model 273A seeks to set the converter to a range that will give an output between 150 and 1900 counts.

**AR n [6]:** A SET/READ command, AR (AUTO-RANGE) turns the auto-ranging function ON or OFF for any of three different parameters, CURRENT, POTENTIAL, or AUX A/D INPUT. If "n" is omitted, the command requests the Model 273A to report its autoranging status. AR differs from AS in that, whereas AS causes a single current auto-range to take place, AR establishes a continuously active mode. Auto-ranging is functional in the STANDBY, SCAN, and CURVE ACQUISITION modes. The codes are:

| n | I AUTO-RNGE | AUTO-RNG | AUX AUTO-RNG |
|---|---|---|---|
| 0 | OFF | OFF | OFF |
| 1 | ON | OFF | OFF |
| 2 | OFF | ON | OFF |
| 3 | ON | ON | OFF |
| 4 | OFF | OFF | ON |
| 5 | ON | OFF | ON |
| 6 | OFF | ON | ON |
| 7 | ON | ON | ON |

The default value is 6 (I OFF, E ON, AUX ON).

Current Auto-ranging causes the Model 273A to automatically seek that current range which puts the output between 15% and 190% of full scale. It does not affect the IGAIN, however. This process occurs after each point is acquired, with the constraint that the converter can only change one range at a time. Also, auto-ranging will yield meaningless results if the time base is less than 1000 µs, or if "I" is not being sampled. Data can also be lost if the data amplitude is changing faster than the auto-ranging function can track. No current auto-ranging will take place if the Model 273A is in the GALVANOSTAT mode.

Current auto-ranging can only be performed in potentiostat operation. Also, current auto-ranging should not be enabled when doing sweep averaging (SAM command operand 1 or 2). If it is, meaningless data will result.

When auto-ranging (only) the current-reading data is packed so that the range setting for any acquired point is stored with the value stored at that point. The word width is 16 bits (D0 through D15; D0 is the LSB and D15 is the MSB). The data value is stored as a 12 bit two's complement number in bits D0 through D11, with D11 acting as the sign bit. The code signifying the current range (codes are the same as listed for the I/E command) is stored in bits D12, D13, D14, and D15. D15 is the sign bit for the I/E Range. See the description of the DC command for information on unpacking these types of data.

E Auto-ranging causes the Model 273A to adjust the gain of the voltage sensed by the Reference Electrode as required to maintain the highest possible resolution (consistent with limit constraints) in reported readings of that potential. The gain is controlled by automatically setting the gain of an amplifier at X1 or X5. If $|E| < 1800$ mV, the gain is set at X5 giving a resolution of 1 mV. If $|E| > 1900$ mV, the gain is set at X1, giving 5 mV resolution. Note, however, that the number of counts stored will be the same in both cases. See the EGAIN command discussion for additional information.

The EGAIN command also affects the E auto-range function. If EGAIN is set to 1 or 5, E auto-ranging behaves as described in the preceding paragraph. If, however, EGAIN is set to 10 or 50, an additional factor of ten gain increase occurs, and the auto-ranging decision points become 180 mV and 190 mV, giving resolutions of 100 µV (gain of X1 and X10 in effect) and 500 µV (gain of X5 and X10 in effect) respectively. See the discussion of EGAIN command for description of relationship between reported "counts" and potential.

AUX auto-ranging causes the Model 273A to adjust the gain of the voltage applied to the rear-panel AUX A/D INPUT connector as required to maintain the highest possible resolution (consistent with limit constraints) in reported readings of that potential. The gain, decision points, and resolution are the same as for E auto-ranging, previously discussed, with the exception that there is no provision for an extra X10 gain.

**Note:** The use of user-generated auto-range routines is not recommended. Complex problems stemming from the internal signal-processing architecture of the Model 273A makes the successful function of user routines in high-noise situations problematical.

**AL n [-6]:** A Set/Read command, AL (AUTO-LIMIT) sets the most sensitive current range to which the Model 273A can automatically range in response to the AS or AR 1 command. If "n" is omitted, the AL command requests that the Model 273A report the limit sensitivity setting. The codes are:

| n | CURRENT RANGE |
|---|---|
| 0 | 1 A |
| -1 | 100 mA |
| -2 | 10 mA |
| -3 | 1 mA |
| -4 | 100 µA |
| -5 | 10 µA |
| -6 | 1 µA |
| -7 | 100 nA |

**Example:** If the computer sends the command AL -5, the highest sensitivity to which the Model 273A can auto range in response to an AS or AR 1 command will be 10 µA.

**EGAIN n [1]:** A SET/READ command, EGAIN (POTENTIAL GAIN) sets the potential measurement gain ahead of the Analog-to-Digital Converter to either X1, X5, X10, or X50. The gain determines the value sent to the host computer as the potential reading. If the gain is X1 or X5, the number of counts (number read) is exactly the potential in mV. If the gain is X10 or X50, the number of counts is exactly the potential in tenths of mV. An example follows.

| E | EGAIN | READING |
|---|---|---|
| 100 mV | X1 | 100 |
| 100 mV | X5 | 100 |
| 100 mV | X10 | 1000 |
| 100 mV | X50 | 1000 |

The reading is always "true." However, with a gain of X1 or X5, the units are mV. With a gain of X10 or X50, the units are tenths of mV. The host computer must expect and accommodate the change.

Note that the AR command, described earlier, also controls EGAIN. As explained in the description of the AR command, AR can switch the potential gain between X1 and X5 as required to achieve the best resolution in reported potential readings. However, AR does not affect the additional X10 gain multiplier. If the X10 multiplier was previously activated by the commands EGAIN 10 or EGAIN 50, AR will switch the net gain between X10 and X50. If the X10 multiplier is not active (EGAIN 1 or EGAIN 5), AR will switch the net gain between X1 and X5. The user might note that the READE command also affects EGAIN in that it automatically deactivates the additional X10 gain multiplier whenever it is applied.

For highest accuracy, the gain should be as high as possible consistent with keeping the amplified voltage below 10. If the gain is too high, the range of the internal circuitry will be exceeded. If the gain is too low, resolution will be lost. A listing of recommended gain versus monitored potential follows.

| \|E\| | RECOMMENDED EGAIN |
|---|---|
| <200 mV | X50 |
| between 200 mV and 1 V | X10 |
| between 1 V and 2 V | X5 |
| >2 V | X1 |

The codes for "n" are:

| n | EXPAND STATUS |
|---|---|
| 1 | X1 |
| 5 | X5 |
| 10 | X10 |
| 50 | X50 |

**Note:** The READE command automatically sets EGAIN to X1 or X5, whichever gives the most accurate READE measurement. If the X10 additional gain multiplier is active when READE is applied, READE immediately deactivates it. Thus, an operator-set value of EGAIN X10 or EGAIN X50 can be lost. Recall that the reported value is direct reading in mV when the net EGAIN is X1 or X5.

**ESUP n [0]:** A SET/READ command, ESUP (POTENTIAL SUPPRESS) allows the E potential signal (only; not AUX or ΔE) to be suppressed ahead of the gain stages that precede the Analog to Digital Converter. If applied without an operand, ESUP requests the Model 273A to report the suppression value. Suppression is in units of 2 mV/count with a range of ±5000 for a full-scale range of ±10 V. Positive values are added to the cell signal.

Note that ESUP worked differently on the older Model 273/97 Option board. On the /97 Option, suppression was still ±10 V but n could range from ±10000, so that each count corresponded to 1 mV. A major difference was the sign of the applied signal. Positive values were subtracted from, not added to, the cell signal.

By allowing the static potential value to be suppressed, ESUP allows small fluctuations with respect to the static value to be readily examined. Note that ESUP is effective whether the unit is running a curve, in Standby, or doing a scan.

n range: 5000...-5000

**Example:** An experiment yields a steady signal of 9 V. ESUP -4500 will suppress this signal. (On the older Model 273/97 Option, ESUP +9000 would have been required.) With this level of suppression applied, EGAIN can then be set to X10 or X50 for increased sensitivity in fluctuation measurements. For example, if the EGAIN operand is 10 or 50, the E readings of the fluctuations will read true in tenths of mV.

Do not confuse this command with EOUTSUP. Whereas EOUTSUP is a /92 Option command that affects only the analog output at the EOUT BNC connector, ESUP does not affect the analog output at EOUT, but rather the digital output from the A/D converter.

**IGAIN n [1]:** A SET/READ command, IGAIN (CURRENT GAIN) sets the current measurement gain ahead of the Analog-to-Digital Converter and thus the number of counts (current reading) provided for any given cell current. If "n" is omitted, the IGAIN command requests that the Model 273A report the IGAIN setting. Gains of X1, X5, X10, and X50 are available. A full-scale current

(IGAIN set to 1) gives a reading of 1000 counts. Readings as high as 2000 counts can be accommodated. The following example shows how the current reading in counts varies with IGAIN for a given current level.

| I | IGAIN | READING (COUNTS) |
|---|---|---|
| 4% of f.s. | 50 | 2000 |
| 4% of f.s. | 10 | 400 |
| 4% of f.s. | 5 | 200 |
| 4% of f.s. | 1 | 40 |

For best accuracy, the IGAIN should be as high as possible consistent with avoiding current overload. The recommended IGAIN as a function of the current level is as follows.

| I | RECOMMENDED IGAIN |
|---|---|
| <4% of f.s. | X50 |
| Between 4% and 20% of f.s. | X10 |
| Between 20% and 40% of f.s. | X5 |
| >40% of f.s. | X1 |

The codes are:

| n | EXPAND STATUS |
|---|---|
| 1 | X1 |
| 5 | X5 |
| 10 | X10 |
| 50 | X50 |

**Note:** The READI command automatically sets IGAIN to X1 or X5, whichever gives the most accurate READI measurement. If the X10 additional gain multiplier is active when READI is applied, READI immediately deactivates it. Thus, an operator-set value of IGAIN X10 or IGAIN X50 can be lost. Recall that the reported value is direct reading in mV when the net IGAIN is X1 or X5.

**ISUP n [0]:** A SET/READ command, ISUP (CURRENT SUPPRESS) allows the I/E current signal to be suppressed ahead of the gain stages that precede the Analog to Digital Converter. If applied without an operand, ISUP requests the Model 273A to report the suppression value. Maximum suppression is two times the present current range. By allowing the static potential value to be suppressed, ISUP allows small fluctuations with respect to the static value to be readily examined.

Since n ranges from ±8000, and full scale is two times the current range, one count corresponds to 2/8000 or $0.25 \times 10^{-3}$ times the current range. For example, on the 1 mA current range 1 count corresponds to 0.25 µA of current.

Note that ISUP is effective whether the unit is running a curve, scanning, or in Standby. ISUP cannot be used when auto-ranging.

n range: 8000...-8000

**Example:** An experiment yields a steady signal of 90% of the selected current range. ISUP -3600 will suppress this signal, giving a net input to the A/D input of "0." With this level of suppression applied, IGAIN can then be adjusted to obtain the desired level of fluctuation sensitivity. For example, if the IGAIN operand is 50, a current of 2 mA on the 100 mA scale will yield 8000 counts.

Note that the sign convention for this command has changed from that used with the older Model 273/97 Option board. Values are added to the signal, whereas with the older /97 Option they were subtracted from the signal.

Do not confuse this command with IOUTSUP. Whereas IOUTSUP is a /92 Option command that affects only the analog output at the IOUT BNC connector, ISUP does not affect the analog output at IOUT, but rather the digital output from the A/D converter.

**SUPDAC n [0]:** A Set/Read command that directly controls the value in the /96 suppression DAC. Allowed inputs are -8190 to +8190. This command will be executed in all timebases. Any subsequent SUPDAC, ISUP, or ESUP will overwrite the DAC value.

**AUXGAIN n [1]:** A SET/READ command, AUXGAIN (AUXILIARY GAIN) sets the gain of the rear-panel Auxiliary A/D Input. If "n" is omitted, the command requests that the Model 273A send the AUXGAIN operand. The codes are:

| "n" | GAIN |
|---|---|
| 1 | X1 |
| 5 | X5 |

Auxiliary input potential readings are always in units of mV and "read true" for both gains. For example, an input potential of 2 V will result in a reading of 2000 with a gain setting of either X1 or X5. However, for best accuracy, a gain of X5 should be used whenever possible. Since the product of the gain times the input voltage cannot exceed 10 V, this means using a gain of X5 if the auxiliary potential is less than 2 V, and a gain of X1 if the auxiliary potential is between 2 V and 10 V.

**MODE n [2]:** A SET/READ command, if MODE is applied with an operand, it sets the Model 273A operating mode. If applied without an operand, it causes the mode to be reported.
The codes are:

| n | MODE |
|---|---|
| 0 | MEASURE ONLY |
| 1 | GALVANOSTAT |
| 2 | POTENTIOSTAT |

**FLT n [0]:** A Set/Read command. To use it as a Set command, enter an operand value to select the desired combination of filters. To use it as a Read command, omit the operand value. The command then requests that the potentiostat report its filter status.

To calculate an operand value, select the desired filters from the table that follows and add the decimal values for the selected filters together. Setting a bit to 1 (that is, including its decimal weight in the operand value) turns the associated filter ON. Setting a bit to 0 (not including its decimal weight) turns the associated filter OFF.

The command returns a decimal value representing the current selected filters as listed in the following table. The default is 0 (all filters OFF).

| FILTER | HEX BIT | DECIMAL WEIGHT |
|---|---|---|
| I signal 5.3 Hz low-pass filter | 0 | 1 |
| Reserved | 1 | 2 |
| Reserved | 2 | 4 |
| I signal 590 Hz low-pass filter[1] | 3 | 8 |
| E signal 5.3 Hz low-pass filter | 4 | 16 |
| E signal 590 Hz low-pass filter[2] | 5 | 32 |

**Notes:**

1. You must set both bit 0 and bit 3 for the I signal 590 Hz low-pass filter to be active.

2. You must set both bit 4 and bit 5 for the E signal 590 Hz low-pass filter to be active.

**Example:** To select the I signal and E signal 5.3 Hz low-pass filter, you must set bits 0 and 4 (see table). Adding the associated decimal weights (1 and 16) gives a total operand value of 17. Enter FLT 17 as the command.

**BW n [0]:** A SET/READ command, BW (BANDWIDTH) sets the bandwidth/stability parameter. If "n" is omitted, it is a request for the Model 273A to report this information. The codes are:

| **n** | **MODE** |
|---|---|
| 0 | HIGH STABILITY |
| 1 | HIGH SPEED |

**CELL n [0]:** A SET/READ command, CELL controls the internal cell relay and active FET switch. If "n" is omitted, the command is a request that the Model 273A report the relay/switch status. The front-panel CELL ENABLE switch MUST be in the ON position for the counter-electrode path to the cell to be completed. This is always true and applies to both front-panel and computer-controlled operation. The codes are:

| **n** | **STATUS** |
|---|---|
| 0 | OFF |
| 1 | ON |

**Note:** Also see description of DUMMY Command, which causes the 273A to report the status of the Electrometer Cell/Dummy switch, and the description of the CS command, which causes the 273A to report the status of the CELL ENABLE switch.

**EXT n [0]:** A SET/READ command, EXT determines whether the front-panel EXT INPUT is ON or OFF. If "n" is omitted, it is a request for the Model 273A to report the External Input status to the host computer. The codes are:

| **n2** | **STATUS** |
|---|---|
| 0 | OFF |
| 1 | ON |

**DCL:** An ACTION command, DCL (DEVICE CLEAR) restores the default parameter values with two exceptions: the MSK and DD operands do not change.

**CAL:** An ACTION command, CAL (CALIBRATE) is used to calibrate the Model 273A. CAL has the same effect as keying in FUNCTION 21 from the front panel of the Model 273A. Before using CAL, allow the instrument to warm up for at least ½ hour.

## 2.3. Memory Partitioning and Curve Designation Commands

The Model 273A has memory enough to store 6144 data values in as many as six curves (Curve 0 through Curve 5). The maximum number of curves depends on their length, up to the maximum of six curves. If a curve is short (1024 or fewer points), the 6144 point memory is large enough to allow the maximum of six curves. At the other extreme, if a curve is longer than 3072 points, the 6144 point memory will only be large enough for one curve. Unused memory capacity is simply unavailable. For example, a 5000 point curve would leave 1144 points unused and unavailable. For purposes of memory partitioning, the curve length is defined as LP +1, where LP is the operand of the LP (LAST POINT) command. For example, if the last point as designated by the LP command is 1024, the curve length would be considered to be 1025 points. As indicated in the table that follows, there would only be room for three curves of this length in memory. If the curve length were reduced by a single point, the number of curves that could be stored would increase from three to six, the maximum..

Curves are identified by numbers, and each curve always begins at a specific point in memory. The numbers range from 0 to 5, that is, six numbers to designate as many as the maximum of six curves. The point at which each curve begins is:

| | |
|---|---|
| CURVE 0 | 0 |
| CURVE 1 | 1024 |
| CURVE 2 | 2048 |
| CURVE 3 | 3072 |
| CURVE 4 | 4096 |
| CURVE 5 | 5120 |

In other words, the starting location of each curve in memory is always the same. (Knowing these locations is critical when doing data dumps from the Model 273A to the host computer.) For a given curve length, as determined by the LP command, only specific curves will be available, as indicated in the list that follows. For example, if the curve length were such as to allow six curves, all six curves, 0 through 5, would be available. On the other hand, if the curve length were such as to allow three curves, the available curves would be 0, 2, 4, and the starting location of each would be as listed above. Curves 1, 3, and 5 would not be available and could not designated.

| LENGTH | CURVES AVAIL. | NUMBER(S) |
|---|---|---|
| 3073 to 6144 | 1 | 0 |
| 2049 to 3072 | 2 | 0 & 3 |
| 1025 to 2048 | 3 | 0, 2, & 4 |
| 1 to 1024 | 6 | 0, 1, 2, 3, 4, & 5 |

Each point is a 16 bit signed integer (-32767 to +32767). By making the capacity of the memory locations (nominally 32000) much larger than the full-scale output of the analog-to-digital converter (nominally 2000), room is provided for sweep averaging. In the linear averaging mode, sixteen sweeps containing full-scale values could be averaged without overflowing the memory. In the exponential averaging mode, a very large number of sweeps could be averaged since the value at convergence won't exceed the value at the output of the analog-to-digital converter for that point.

Functionally, there are five kinds of curves. They are:

1. SOURCE CURVE: Data stored in the Source Curve defines a modulation waveform. If the MM 2 command has been executed, the modulation waveform stored in the SOURCE CURVE will be applied during subsequent curve acquisitions.

2. DESTINATION CURVE: This is the curve into which data will be placed during curve acquisitions. I, E, AUX, and E (the current-interrupt compensation potential) could all be stored in successive curves as determined by the SIE command. The first curve is the Destination Curve. The other or others, according to the data designations made with the SIE command, are NEXT CURVES.

3. NEXT CURVE: This is the curve into which E, AUX, or E data is placed during curve acquisitions. There can be zero, one, two, or five Next Curves according to the SIE selection. NEXT CURVES, if present, always immediately follow the DESTINATION CURVE.

4. ALTERNATE CURVE: During a run comprising many sweeps, it is possible to store the measurement results obtained on some number of sweeps in one curve, and the measurement results obtained on the remainder of the sweeps in another curve. The first set of results is stored in the DESTINATION CURVE. The curve in which the results for the remainder are stored is the ALTERNATE CURVE.

5. PROCESSING CURVE: A curve on which post-acquisition processing (see descriptions of Curve Processing Commands, ADD, SUB, EX, MIN, IMIN, MAX, IMAX, INT, IINT, and ILOG) is performed is the PROCESSING CURVE. A given curve can be a PROCESSING CURVE as well as one of the other previously described kinds of curves. For example, you could designate Curve 1 as both the DESTINATION CURVE and the PROCESSING CURVE.

Four commands are provided that allow the operator to designate curve functions. In using these commands, it is essential that the constraints on the number of allowable curves, and on the available curve-designation numbers, be observed, as discussed in the first few paragraphs of this section. A discussion of the four commands follows.

**DCV n[0]:** A SET/READ command, DCV designates the DESTINATION CURVE. If n = -1, no data will be stored in any curve. This allows data from preliminary sweeps to be discarded, useful when the ACV command is used. If n is omitted, this command requests that the Model 273A report the Destination Curve number.

n range: -1...5

DCV also designates the NEXT CURVE (or curves), which will be the next one(s) available.

**Note:** If the user misprograms the Model 273A with the DCV command so as to cause data to be written past the end of the memory, data for that curve will not be stored at all.

For example, if the user sets DCV to "0", SCV to "1" and LP to 1023, there is no problem, since Curve 0 can hold up to 1024 points. If LP is set to 1024 or higher, data would be written onto the source curve. To prevent this, no data would be stored for the curve at all.

**ACV n1 n2 [0 0]:** A SET/READ command, ACV (ALTERNATE CURVE) designates the Alternate Curve, and the number of the sweep with which data storage transfer to the Alternate Curve is to begin. If n1 and n2 are omitted, it is a request for the 273A to report this information.

The ranges are:

n1 range: -1...5
n2 range: 1...65535

An n1 value of -1 means that no curve is designated as the Alternate Curve.

**Example:** The command ACV 1 5 would designate CURVE 1 the ALTERNATE CURVE. Beginning with sweep 5, acquired data will be stored in this curve. The sweep number counts up from one to the number specified by the SWP operand. For example, if ten sweeps were specified, the first would be number 1, the second number 2, etc. For prior sweeps, the DESTINATION CURVE would be located as specified by the DCV command.

If n2 is a "0", the DESTINATION CURVE will not be relocated. Also, assuming a valid ACV command is applied with the resulting relocation of the DESTINATION CURVE at the proper time, when the next NC (NEW CURVE) command is applied, the original location set by DCV will be restored.

**Note:** If the user misprograms the Model 273A with the ACV command so as to cause data to be written past the end of the memory, no data for that curve will be stored.

For example, if the user sets ACV to "0", SCV to "1" and LP to 1023, there is no problem, since Curve 0 can hold up to 1024 points. If LP is set to 1024 or higher, data would be written onto the source curve. To prevent this, no data would be stored for the curve at all.

**SCV n [3]:** A SET/READ command, SCV (SOURCE CURVE) designates the Source curve. If "n" is omitted, the command is a request for the Model 273A to transmit this information.

n range: 0...5

A curve may be designated as both the SOURCE CURVE and DESTINATION CURVE. This would be done, for example, if it were necessary to acquire a 6144 point curve using the arbitrary waveform modulation mode. The SOURCE CURVE data would be replaced by the acquired data as each point is taken. Therefore, only one sweep can be done. Since the NC command clears the DESTINATION CURVE buffer, the SOURCE CURVE should be loaded into the Model 273A after NC but before TC.

An alternative method is to load the Source Curve into the potentiostat and then use the RC command instead of the NC command.

**PCV n [0]:** A SET/READ command, PCV (PROCESSING CURVE) designates the Processing Curve. If "n" is omitted, the command is a request for the Model 273A to send this information.

n range: 0...5

**Example:** The command sequence:

PCV 0;ADD 100;PCV 1;ADD 200

would cause 100 to be added to all points in CURVE 0 and 200 to be added to all points in CURVE 1.

## 2.4. BIAS Commands

The potential (current) applied to the cell is the sum of several sources. When controlling the 273A from the front panel, the potential/current set with the front-panel keys, together with any potential applied to the EXT. INPUT connector, determines the cell potential (current). When controlling from an external computer, the front-panel keys are no longer active, and the cell potential (current) is set by the modulation and bias commands, together with any potential applied to the EXT. INPUT connector (see discussion of EXT command).

As previously explained, the MODULATION DAC allows a cell excitation that changes over the course of a curve acquisition to be applied. The function of the BIAS DAC is similar except that its output is fixed at the programmed level. A detailed discussion of the BIAS, SETE, and SETI command follows.

**BIAS n [0]:** A SET/READ command, BIAS sets the bias level to be applied as soon as the cell is on. If n is omitted, it is a request for the Model 273A to report the bias setting. Full scale is ±8000. Note that the polarity of the applied bias depends on the Model 273A Operating Mode. In Galvanostatic operation, BIAS 8000 causes a bias current of -2 * f.s. In Potentiostatic operation, BIAS 8000 causes a bias potential of +8 V.

n range: -8000...8000

The Model 273A follows the American polarity convention and the Model 273A display indications are consistent with that convention. Positive current is cathodic, that is, a current is defined as positive if reduction is taking place. Negative current is anodic, that is, a current is defined as negative if oxidation is taking place. If the working electrode is driven positive with respect to the equilibrium potential, the resulting current is anodic. If the electrode is driven negative with respect to the equilibrium potential, the resulting current is cathodic. In complex electrochemical systems, there may be more than one equilibrium system. Where this is the case, either polarity with respect to the equilibrium potential could give rise to anodic or cathodic current, according to the system's characteristics.

**SETE n [0]:** A SET/READ command, SETE (SET POTENTIAL) sets the number of millivolts to be applied in the POTENTIOSTAT mode (only). If SETE is applied when the Model 273A is in the GALVANOSTAT mode, a command error (ERROR 11) will be generated. If "n" is omitted, it is a request for the Model 273A to report the SETE operand.

n range: -8000...8000

**Note:** Because the output of the Modulation DAC is set to zero by this command, it interferes with the action of the MOD command.

**SETI n1 n2 [0,0]:** A SET/READ command, SETI (SET CURRENT) command sets the cell current in the GALVANOSTAT mode (only). If SETI is applied when the Model 273A is in the POTENTIOSTAT mode, a command error (ERROR 11) will be generated. If n1 and n2 are omitted, it is a request for the Model 273A to report the SETI operands. The current is n1 * $10^{n2}$ amperes. The ranges are:

n1 range: -2000...2000
n2 range: -3 to -10

The actual current range as a function of n2 is as follows.

| n2 | RANGE |
|---|---|
| -3 | 1 A |
| -4 | 100 mA |
| -5 | 10 mA |
| -6 | 1 mA |
| -7 | 100 µA |
| -8 | 10 µA |
| -9 | 1 µA |

These settings conflict with those used in the I/E command, but allow easy reading of the current level.

**Example:** The command SETI 1000 -6 will set the current to 1000 * $10^{-6}$ A, that is, to 1 mA.

**Note 1:** Although a given current can be programmed with different combinations of n1 and n2, resolution and precision considerations will generally make it desirable that n2 *be as small as possible*. This is because n2 sets the current range and n1 sets the fraction of that range to be programmed. The error on any range is a percentage of the range (plus other components). If a current is set by making n2 large and n1 small, the error, a fixed percentage of the n2 range, could well be an appreciable portion of desired setting. For example, both SETI 1 -3 and SETI 1000 -6 will set the current to 1 mA. If the command is SETI 1 -3 the setting is .001 of the 1 A range, that is, 1 mA. However, the error is a percentage of the 1 A, and will be large relative to the 1 mA being set. If the command is SETI 1000 -6 the setting is one times the 1 mA range, and the error, a percentage of the 1 mA range, will be as small as possible.

**Note 2:** Because this command sets the output of the Modulation DAC to zero, it interferes with the action of the MOD command.

## 2.5. Modulation Commands

The Model 273A is capable of applying a modulation program during each sweep such that a different potential is applied to the cell for each point. In many applications, the applied modulation will be a ramp that progresses as programmed. However, there is provision for specifying and applying any arbitrary waveform, if desired. The modulation applied depends on the operating mode. Although full-scale modulation is always ±8000 counts, the available potential or current modulation range depends on the MR command and the actual modulation is easily expressed in units of potential or current (see descriptions of individual modulation commands).

In CONTROL E operation, the full-scale count range of ±8000 corresponds to potentials of ±20 mV, ±200 mV, or ±2 V, as established by the MR (MOD RANGE) command operand. In the GALVANOSTAT mode, +8000 counts is anodic full scale and -8000 counts is cathodic full scale. The available ranges as determined by the MR command are ±200%, ±20%, and ±2% of full scale.

There are several constraints that must be observed in defining a ramp program. If any are violated, a parameter error is generated and the erroneous ramp-program step is ignored. The constraints are:

1. For the INITIAL command, n1 must be the point number of the first point of the curve and n2 must be a valid modulation DAC value, that is, between -8000 and 8000.

2. For the VERTEX command, n1 must be greater than n1 of the previous step and n2 must be a valid modulation DAC value. Also, n1 of the last VERTEX command should equal the memory address of the last point of the curve. Otherwise the ramp program is not specified for the entire curve.

Consider the following example:

INITIAL 0 -8000;VERTEX 999 8000

which generates a ramp from negative full scale to positive full scale. This default ramp program specifies that the modulation change by 16000 counts over 1000 steps with a data point taken at each step. The first data value is stored in point 0 and the last data value is stored in point 999. Therefore, the modulation will advance 16 counts per step.

In some cases, the step size will not be constant. For example, the ramp program:

INITIAL 0 0;VERTEX 999 1001

requires that the modulation change by 1001 counts over 1000 steps. To accomplish this, 999 steps would each be one count, but one step would be two counts.

A discussion of the individual modulation commands follows.

**MR n [2]:** A SET/READ command, MR (MODULATION RANGE) sets the modulation potential or current range. If "n" is omitted, MR is a request for the Model 273A to report the range in effect. Modulation potential ranges of 20 mV, 200 mV, or 2 V can be selected. The corresponding current ranges are 2%, 20%, and 200% of the full-scale current range in effect.

Full-scale modulation is 8000 counts. In potentiostatic operation, this yields a resolution of 2.5µV per count on the 20 mV Modulation range, 25 µV per count on the 200 mV range, and 250 µV per count on the 2 V range. In galvanostatic operation, the corresponding figures, per count, are .025%, .0025%, and .00025% of the full-scale current range. The codes are:

| n | CONTROL E MOD. RANGE | CONTROL I MOD. RANGE |
|---|---|---|
| 0 | 20 mV | 2% of f.s. |
| 1 | 200 mV | 20% of f.s. |
| 2 | 2 V | 200% of f.s. |

**Example:** The command MR 1 will set a modulation range of 200 mV (CONTROL E) or 20% of full scale (CONTROL I).

**MM n [0]:** A SET/READ command, MM (MODULATION MODE) sets the Model 273A Modulation Mode. When "n" is omitted, MM requests that the Model 273A report the Modulation Mode. The codes are:

| n | MODULATION MODE |
|---|---|
| 0 | NO MODULATION |
| 1 | RAMP PROGRAM MOD. ON |
| 2 | ARBITRARY WAVEFORM MODULATION |

In the RAMP PROGRAM mode, the modulation is a ramp specified by the INITIAL n1 n2 and VERTEX n1 n2 commands. As the experiment progresses, this ramp develops point by point and is applied to the cell. If desired, the ASM command can be used to load the ramp program into the SOURCE CURVE for convenient transfer to the host computer.

In the ARBITRARY WAVEFORM mode, the modulation is a waveform specified point-by-point by the computer and stored in the Model 273A as the "SOURCE CURVE" by the LC (LOAD CURVE) or BL (BINARY LOAD) commands.

**INITIAL n1 n2 [0 -8000]:** A SET command, INITIAL (INITIAL MODULATION & POINT) tells the Model 273A the point number where the first data value will be stored, and the modulation value at that point. It is essential that the point specified by "n1" be the same as that specified by the Curve Acquisition FP command. Otherwise a parameter error (ERROR 3) will be generated. Note that the INITIAL command must be the first Ramp Program command executed because it erases any Ramp Program in the Model 273A Memory. The ranges are:

n1:  0...6143
n2: -8000...8000

As explained in the discussion of the MR command, the actual modulation corresponding to the value of "n2" depends on the MR command.

The actual curve modulation/acquisition process begins when the TC (TAKE CURVE) command is applied.

**Example:** Suppose one wished the modulation ramp to begin at point 0 with a value of 50 mV, where the MR 1 command had set the full-scale modulation level to 200 mV. The appropriate command would be INITIAL 0,2000. "0" specifies point 0. "2000" specifies 50 mV. (50 mV/200 mV f.s. modulation output * 8000 counts f.s. for the DAC = 2000 counts)

**VERTEX n1 n2 [999 8000]:** A SET command, VERTEX (VERTEX MODULATION & POINT) specifies the end point of the ramp and the modulation level at that point. At least one VERTEX command is required if an INITIAL command has been applied. However, more than one Vertex command up to a maximum of 50 can be specified, allowing very complex modulation programs to be generated. The program begins at the point specified by the INITIAL command, advances to the point specified by the first VERTEX command, and from there advances to the point specified by the second VERTEX command. From there the ramp modulation sequence advances in similar fashion until the entire program has been executed.

The actual ramp begins when the TC (TAKE CURVE) command is applied.

"n1", the point operand, must be larger than the FP (FIRST POINT) operand but less than or equal to the LP (LAST POINT) operand. Otherwise, a parameter error (ERROR 3) will be generated. The allowable range is 1 to 6143.

"n1" of the last VERTEX command of a ramp program should equal the last point. Otherwise, the ramp values for all points would NOT be fully specified and results for unspecified points would be unpredictable.

As explained in the discussion of the MR command, the actual modulation corresponding to the value of "n2" depends on the MR command.

**Example:** A modulation waveform could be specified by the following program:

INITIAL 0 0;VERTEX 400 4000;VERTEX 600 4000

Assuming that MM 1 had been sent to select the ramp program mode, during the subsequent curve acquisition, the modulation would begin with the acquisition of point 0 and the initial modulation value would be 0 counts. The ramp would proceed in steps to point 400, where it would have a value of 50% of full scale (50% of 8000 is 4000). It would remain at this level to the end of the program (point 600).

**Note 1:** INITIAL n1 n2 must be sent before any VERTEX commands. Sending another INITIAL command reinitializes the function, erasing any previous vertices.

**Note 2:** The power-up default values for these commands are
INITIAL 0 -8000;VERTEX 999 8000, which generates a 1000-point ramp from minus full scale to plus full scale.

The step size will depend on the specified number of points and on the specified modulation levels. For example, in the default case there are 16,000 counts and 1,000 points. The step size in counts per point will be 16. If the user specifies values which do not divide evenly, the Model 273A will vary the step size up and down as the scan progresses to maintain as close to a linear ramp as possible.

**PROG n1 n2 n3 n4....nx:** A READ command, PROG (PROGRAM) is a request for the Model 273A to send the ramp program (INITIAL operands and all VERTEX operands) to the host computer.

n1 range: 0...6143
n2 range: -8000...8000
n3 range: 1...6143
n4 range: -8000...8000

**Example:** Assuming the default ramp program values were in place, the PROG command would result in the following numbers being sent to the host computer.

0 -8000
999 8000

n1, the INITIAL Point, is 0, and n2, the INITIAL Modulation, is -8000. n3, the VERTEX Point, is 999, and n4, the VERTEX Modulation, is 8000.

**ASM:** An ACTION command, ASM (ASSEMBLE) causes the RAMP program to be assembled into a sequence of values that are stored in the SOURCE CURVE. Once assembled and stored, the data in the SOURCE CURVE can be transferred to the host computer (for graphing) using the DC or BINARY DUMP commands. Alternatively, it may be used to generate a ramp in the arbitrary waveform mode.

**MOD n [0]:** A SET/READ command, MOD (MODULATION) allows the user to immediately program a constant modulation output level. The MOD command is used to set the MOD DAC output when the MM operand is 0. If n is omitted, the Model 273A is requested to report the MOD setting. Setting the modulation via this command will result in a conflict if the MM operand is 1 or 2 and data acquisition is underway.

n range: -8000...8000

**INTRP n [1]:** A SET/READ command, INTRP (INTERPOLATION) sets the interpolation function status. If n is omitted, the Model 273A is requested to report the ON/OFF status of this function. The codes are:

| **n** | **STATUS** |
|---|---|
| 0 | OFF |
| 1 | ON |

Interpolation applies to both the Modulation 1 (RAMP PROGRAM MODE) and Modulation 2 (ARBITRARY WAVEFORM) modes.

In understanding the interpolation function, it is important to distinguish between a point and a sample. A sample is taken every time the potential or current is measured and an analog-to-digital conversion done on the measured value. Values placed in memory are points. A point could be a single sample value, or the average of many sample values (see discussion of PAM and S/P commands).

The INTRP command determine how often the Modulation DAC, and hence the modulation level, is updated. If the Interpolation Function is OFF, the MOD DAC output is updated once per point. If the Interpolation Function is ON, the MOD DAC output is updated once per sample. If the number of samples per point is greater than one, this will result in a smoother modulation waveform. In other words, the INTRP command allows the modulation to advance in smaller steps in situations where there are many samples per point.

**Example:** Assume the "Arbitrary Waveform" Modulation Mode (MM 2), the "Store Average of N Samples" Point Averaging Mode (PAM 2), and ten samples per point (S/P 10). Further assume that Point 0 of the Source Curve has a value of 0 and that Point 1 of the Source Curve has a value of 20. If the Interpolation Function were OFF, the MOD DAC output would remain at 0 throughout the ten samples of Point 0, and would jump to 20 after the last sample of Point 0 to prepare for Point 1. If the Interpolation Function were ON, the MOD DAC output would be interpolated between 0 and 20. Its output would start at 0. After Sample 1, which is the first sample taken for the point (up-counting is used), it would be set to 2 in preparation for Sample 2. After Sample 2, it would be set to 4 in preparation for Sample 3. This process would continue through Sample Number 10, at which point the MOD DAC output would be set to 20 in preparation for Sample Number 1 of Point Number 1.

## 2.6. Data Acquisition Control Commands

**FP n [0]:** A SET/READ command, FP (FIRST POINT) sets the location of the start of the stored curve. If "n" is omitted, the location is reported. FP defines a point number in the DCV, not direct memory addresses. If DCV is 0, then FP 0 means to start storing data in point 0 of Curve 0. If DCV is 2, FP 0 means to start storing data in point 0 of Curve 2 (really point 2048 in the memory). For maximum data space, FP is usually set to 0. The same relative addressing applies to LP. The Curve Length, with respect to the maximum number of data values, is simply (LP - FP) +1. However, with respect to memory partitioning considerations, it is LP + 1. Values which would overwrite the end of a curve will result in no data being stored.

"n" range: 0...6143

**Note:** If modulation is being applied using the ramp-program commands, the point designated by the INITIAL n1 operand must be the same as that designated by the FP operand (FP predominates).

**LP n [999]:** A SET/READ command, LP (LAST POINT) sets the last data storage location. Together with FP, LP sets the curve length, which is simply (LP - FP) + 1. However, with respect to memory partitioning, the curve length is defined simply as LP + 1. If "n" is omitted, the value of LP in effect is reported. Like FP, LP is relative addressed with respect to the designated curve.

n range: 1...6143

"n" must be larger than the FP operand.

**Note:** Data points between the first point and the last point inclusive are referred to as ACTIVE POINTS.

**RC:** An ACTION command, RC (RESET CURVE) initializes the Model 273A for curve acquisition. It works like the New Curve (NC) command below except that it does not clear the Destination Curve, the Next Curve, or the Alternate Curve.

**NC:** An ACTION command, NC (NEW CURVE) causes the Model 273A to be initialized for curve acquisition. NC *must be executed before* TC (TAKE CURVE). Specific actions of the NC command are:

1. Clears (sets to zero) all active points in the Destination Curve and Next Curve. It also clears Alternate Curve if the ACV command is activated.

**Note:** The Next Curve or Curves will be cleared only if data was stored in them. See discussions in Section 2.6, Memory Partitioning and Curve Designation Commands, and the description of the SIE command.

2. Makes the point designated by the FP operand the currently accessed point.

3. If a modulation ramp is to be applied (MM operand = "1" or "2"), NC sets the modulation level to its starting value.

4. Halts curve acquisition if curve acquisition is in progress.

5. Calculates the locations at which the first data values (I, E, AUX, E) are to be stored in the Destination Curve.

6. Calculates the locations at which the first data values (I, E, AUX, E) are to be stored in the Alternate Curve if the ACV command is in effect.

7. Sets the Sweep Number to "1."

8. Calculates the address from which the first modulation value is to be taken if the arbitrary waveform mode is selected.

9. Sets up variables used for interpolating the modulation waveform if INTERPOLATION is enabled.

10. Determines the divisor to be used to average samples as determined by the Point Averaging Mode.

**Once NC is applied, do not apply any command that will change what the data acquisition routines do. Specifically, the following setting commands should not be used after NC is invoked.**

| | | | | |
|---|---|---|---|---|
| TMB | MM | INITIAL | VERTEX | INTRP |
| SCV | SAM | SHF | PAM | S/P |
| SEL | DCV | ACV | SIE | FPLP |

**The behavior of the data acquisition routines will be unpredictable if these commands are used to set acquisition parameters after NC is executed.**

**TC:** An ACTION command, TC (TAKE CURVE) causes the curve acquisition to begin. A data set will be acquired and stored in accordance with the conditions established by the other commands.

**HC:** An ACTION command, HC (HALT CURVE) causes the curve acquisition in process to stop. Curve acquisition can be restarted with a TC command. If this is done, data acquisition will begin with the next active point (all points between FP and LP are active points). If the TC command is preceded by NC, curve acquisition will begin with the point designated by the FP operand.

**WCD:** An ACTION command, WCD (WAIT TILL CURVE DONE), if used in a multiple command, allows the command's execution to be synchronized with curve acquisition.

**Example:** Without WCD, the multiple command:

NC;TC;MIN;MAX

would not work properly because the MIN and MAX operations would be carried out before curve acquisition was completed. However, if WCD were included in the multiple command ahead of MIN and MAX, giving the command:

NC;TC;WCD;MIN;MAX

there would be no problem because WCD would cause execution of the multiple command to pause until curve acquisition was complete.

**WAIT n:** An ACTION command, WAIT causes curve acquisition to halt for "n" time-base intervals (as set by the TMB command). This command is meaningful only if curve acquisition (initiated by the TC command) is in progress. If curve acquisition is not in progress, an AQUISITION ERROR (Error #12) will occur.

n range: 0...65535

**DISCARD n:** An ACTION command, DISCARD causes data acquisition to pause for "n" points. This command is meaningful only if curve acquisition (initiated by the TC command) is in progress. If curve acquisition is not in progress, an AQUISITION ERROR (Error #12) will occur.

n range: 0...65535

## 2.7. Sampling and Sweep Control Commands

### 2.7.1. Sampling Control Commands

In understanding the following discussions, it is important to distinguish between a point and a sample. A sample is taken every time the potential or current is measured and an analog-to-digital conversion is performed on the measured value. Values placed in memory are points. A point could be a single sample value, or the average of many sample values (see discussions of PAM and S/P commands).

In the simplest case of Curve Acquisition, one sample of "E" or "I" is taken at each point. In other words, the number of samples required to fill a curve will equal the number of points in it. However, a wide range of sampling control and averaging is provided for improved flexibility. For example, the programmer may wish to take some number of samples at each point, discard all but the "N"th sample, and store its value. If "N" were "5", 5120 samples would be required to fill a 1024 point curve. Alternatively, the programmer might wish to take "N" samples at each point and to store their average value as the value for that point. Again, if "N" were "5", 5120 samples would be required to fill a 1024 point curve.

**PAM n [0]:** A SET/READ command, PAM (POINT AVERAGING MODE) determines whether the Model 273A will store each sample, select a group of samples, or average the samples taken at each point. If "n" is omitted, the command is a request for the Model 273A to send this information. The codes are:

| **n** | **POINT AVERAGING MODE** |
|---|---|
| 0 | No averaging |
| 1 | Averaging of "n" samples |
| 2 | Store average of selected samples |

Note that averaging will provide an improvement in the signal-to-noise ratio proportional to the square root of the number of samples averaged. If the PAM operand is 1, the number of samples to be averaged is set with the S/P command. If the PAM operand is 2, use SEL n1,n2 to select the samples.

**S/P n [1]:** A SET/READ command, S/P (SAMPLES PER POINT) sets "N", the number of samples to take for each data point. If "n" is omitted, the command is a request for the Model 273A to report this information. If the PAM operand is "0", all but the last sample for a given point will be discarded. If the PAM operand is "1", the sum of "N" samples is divided by

"N". If the PAM operand is "2", a selected range out of the "N" samples, as determined by the SEL command, is averaged for each point. In each case, the result is stored in the DESTINATION CURVE.

n range: 1...32767

**SEL n1 n2 [1 1]:** A SET/READ command, SEL (SELECTED) sets the first and last sample to be selected when the PAM operand is "2". The samples delineated will be averaged and the

average value stored. If "n1" and "n2" are omitted, the values in effect for these operands will be reported. The ranges are:

n1 range: 1...32767
n2 range: 1...32767

"n2" must be larger than, or equal to, n1. For example, if the S/P operand is "10", and the SEL operands are "2" and "7", ten samples would be taken at each point. Of these, samples 2, 3, 4, 5, 6, and 7 would be averaged (added and divided by six) and the average stored as the value for the point. Sample 1 would be discarded, as would be samples 8, 9, and 10.

**SIE n [1]:** A SET/READ command, SIE (SAMPLE I/E) determines whether the sampled parameter will be current (I), the Electrometer Monitor potential (E), the potential applied to the rear-panel AUX A/D INPUT (AUX), the Current Interrupt IR Compensation Potential (ΔE), or any combination of these parameters. If "n" is omitted, this command is a request for the Model 273A to report the SIE setting. The codes are:

| **n** | **I** | **E** | **AUX** | **ΔE** |
|---|---|---|---|---|
| 0 | OFF | OFF | OFF | OFF |
| 1 | ON | OFF | OFF | OFF |
| 2 | OFF | ON | OFF | OFF |
| 3 | ON | ON | OFF | OFF |
| 4 | OFF | OFF | ON | OFF |
| 5 | ON | OFF | ON | OFF |
| 6 | OFF | ON | ON | OFF |
| 7 | ON | ON | ON | OFF |
| 8 | OFF | OFF | OFF | ON |
| 9 | ON | OFF | OFF | ON |
| 10 | OFF | ON | OFF | ON |
| 11 | ON | ON | OFF | ON |
| 12 | OFF | OFF | ON | ON |
| 13 | ON | OFF | ON | ON |
| 14 | OFF | ON | ON | ON |
| 15 | ON | ON | ON | ON |

**TMB n [4000]:** A SET/READ command, TMB (TIMEBASE) sets the interval between samples in MICROSECONDS. When "n" is omitted, the command is a request for the Model 273A to report this information. TMB is a curve acquisition parameter; it doesn't affect the standby or scan timing.

The range is:

n range: 50...50000

Although "n" can be set to values as low as 50, it may not be possible to actually take data at the 50 µs rate. The time required to run the data acquisition routine must also be considered and can vary over a wide range. If the amount of time to execute the data acquisition interrupt routine exceeds the period of the timebase, the microprocessor will be unable to perform data acquisition at the desired rate. Furthermore, the microprocessor will be totally monopolized by the data acquisition routine and therefore will be unable to respond to commands until curve acquisition is completed.

When a curve acquisition is initiated with the sequence "NC;TC", the 273A begins by checking the timebase as established by the last TMB command. Depending on the value in effect, one of three different Data Acquisition Routines will be selected. These three are:

**FRILLS:** TMB operand of 4000 or larger.

**SOME FRILLS:** TMB operand range of 200 to 3999.

**NO FRILLS:** TMB operand in the range of 50 to 199.

Each is explained in more detail in the following paragraphs.

**FRILLS:** When the Frills Routine is in effect, no tradeoffs are made; there is time enough for all functions selected to be supported.

**SOME FRILLS:** When the Some Frills Routine is in effect, the following functions are not supported. If selected, or if the corresponding commands are applied, they simply will not take place.

*Measurement of Charge*
*Updating of the front-panel OUTPUT connector*
*Current Interrupt IR Compensation*
*Line Synchronized Sampling*
*Autoranging of E and AUX*
*Maintenance of the A/D Overload bits (reported by OVER command)*
*ESUP and ISUP functions*

Note that the question of minimum attainable timebase is separate from that of supported functions. In the SOME FRILLS mode, the minimum 200 µs per sample rate is only possible if:

| | |
|---|---|
| I only is being sampled | (SIE 1) |
| Constant potential or current is applied | (MM 0) |
| No point averaging is being done | (PAM 0) |
| No sweep averaging is being done | (SAM 0) |
| No autoranging is being done | (AR 0) |

As additional functions are turned on, the minimum sampling time increases, as follows. Where more than one function is operative, the times must be added.

| | | |
|---|---|---|
| SIE 2 | 10 µs | (Sample E) |
| SIE 3 | 80 µs | (Sample I and E) |
| SIE 41 | 40 µs | (Sample AUX) |
| SIE 7 | 230 µs | (Sample I, E, and AUX) |
| MM 1 | 110 µs | (Ramp Program Modulation) |
| MM 2 | 60 µs | (Arbitrary Waveform) |
| PAM 1 | 60 µs | (Point Averaging) |

| | | |
|---|---|---|
| PAM 2 | 70 µs | (Selected Point Averaging) |
| SAM 1 | 10 µs | (Linear Sweep Averaging) |
| SAM 2 | 30 µs | (Exponential Sweep Averaging) |
| AR 1 | 20 µs | (I Autoranging) |

The 120 µs penalty for I Autoranging assumes that no autorange is needed. An additional 100µs is required if an autorange occurs.

Should you select a timebase that is too fast for the selected functions, the 273A will go as fast as it can. The Curve Acquisition Routine will use all of the available CPU time, and the 273A will not respond to any commands until the curve is done.

**NO FRILLS:** If faster sampling is required, you can use the No Frills data acquisition routine so long as the conditions given below are met. However, this routine allows very little flexibility. The sampled signal will be I, E, or AUX, as determined by the most recent SIE command. SIE 1, SIE 3, and SIE 7 will all cause I to be sampled. SIE 2 and SIE 6 will cause E to be sampled. SIE 4 will cause AUX to be sampled. Also, the following functions will be in effect regardless of the user's settings.

| | |
|---|---|
| Arbitrary Waveform selected | MM 0 or MM 2 |
| No Point Averaging | PAM 0 |
| Linear Sweep Averaging | SAM 0 or SAM 1 |
| One Sample per Point | S/P 1 |
| No Autoranging | AR 0 |

The Frills and Some Frills routines compensate for the factor of 5 gain as determined by EGAIN and AUXGAIN. The No Frills routine does not make this compensation, so E and AUX are stored in the curve buffer as straight A/D readings with 2000 being full scale.

**LS n [0]:** A SET/READ command, LS (LINE SYNC) determines whether data acquisition will be synchronized with the power-line frequency. If LS is applied without "n", it is a request for the Model 273A to report this information. The codes are:

| **n** | **STATUS** |
|---|---|
| 0 | OFF |
| 1 | ON |

The sample is taken when the next power-frequency zero crossing after the interval set by the TMB command occurs.

### 2.7.2. Sweep Control Commands

It is possible to make multiple sweeps, using sample selection or averaging, and to average the data resulting from the sweeps, either linearly or exponentially, to obtain the values stored in the curve. Descriptions of the sampling and sweep averaging commands follow.

**SWPS n [1]:** A SET/READ command, SWPS (SWEEPS) sets the number of sweeps in an acquisition. If "n" is omitted, the command is a request for the Model 273A to report the SWPS setting.

n range: 1...65535

**SAM n [0]:** A SET/READ command, SAM (SWEEP AVERAGING MODE) determines whether sweeps will be averaged, and, if averaged, whether the averaging will be linear or exponential. If "n" is omitted, the command is a request for the Model 273A to provide this information.

In linear averaging, the results of each sweep are added point by point, with equal weighting applied to the data from all sweeps. It is essential that the number of sweeps and the magnitude of the readings in each be such as not to exceed the memory capacity (-32768 to +32767 at each point). Linear averaging yields an improvement in signal- to-noise ratio proportional to the square root of the number of sweeps averaged. If data values approach full scale, a maximum of 16 sweeps can be averaged.

In exponential averaging, the data from the current sweep is added to the data from the previous sweeps in such a way as to effect exponential averaging, that is, the value stored at each point asymptotically approaches a final value. The average is weighted such that the most recent sweep has the greatest influence. The Signal-to-Noise Improvement Ratio, at convergence, is a function of N (N is the Shift Number as defined in the following discussion of the SHF command). After $2^n$ sweeps (n is the SHF operand), the data reaches 63% of its final value. After six times $2^n$ sweeps, the data reaches its final value. Additional sweeps provide no increase in accumulated signal amplitude and no further improvement in the signal-to-noise ratio. Fewer than $2^n$ sweeps results in a smaller signal-to-noise ratio than indicated in the discussion of the SHF command, and in a lower amplitude accumulated signal level. In each case the averaging is done as the data is taken. The codes are:

| **n** | **MODE** |
|---|---|
| 0 | No Sweep Averaging |
| 1 | Linear Sweep Averaging |
| 2 | Exponential Sweep Averaging |

**Note:** Sweep averaging, either linear or exponential, should not be used with auto ranging (AR command) enabled. If it is, meaningless data will result.

**SHF n [1]:** A SET/READ command, SHF (SHIFTS) sets the number of times the data is to be shifted before being added to the old data when doing Exponential Averaging. If "n" is omitted, the command is a request for the Model 273A to report the SHF setting. "n" can take values from 1 to 15. The averaged function reaches 63% of its final value in $2^n$ sweeps. The function converges in 6 x $2^n$ sweeps. The signal-to-noise improvement ratio, at convergence, improves 6 dB for each 2n, as indicated in the following table.

| **n** | **SWEEPS TO REACH 63%** | **SWEEPS TO CONVERGE** | **SNIR** | **SNI (dB)** |
|---|---|---|---|---|
| 2 | 4 | 24 | 2 | 6 dB |
| 4 | 16 | 96 | 4 | 12 dB |
| 6 | 64 | 384 | 8 | 18 dB |
| 8 | 256 | 1536 | 16 | 24 dB |
| 10 | 1024 | 6144 | 32 | 30 dB |
| 12 | 4096 | 24576 | 64 | 36 dB |
| 14 | 16384 | 98304 | 128 | 42 dB |

where: SNIR is the Signal-to-Noise Improvement Ratio and SNI is the Signal-to-Noise Improvement in dB.

**Note:** n = 0 is not allowed. However, n can take odd values from 1 to 15.

In a typical application, it may prove impractical to use large values of n because of the long experiment times required.

**DT n [0]:** A SET/READ command, DT (DEAD TIME) sets the time between sweeps (dead time) IN MILLISECONDS. If "n" is omitted, the command is a request for the Model 273A to report the DT setting. "n" can take values from 0 to 65535, with a resolution of 10 ms. For example:

| DT OPERAND | RESULTING DEAD TIME |
|---|---|
| 0 | 0 |
| 1 to 19 | 10 ms |
| 20 to 29 | 20 ms |
| 30 to 39 | 30 ms |

## 2.8. Uncompensated Resistance Commands

**IRMODE n [0]:** A Set/Read command, IRMODE (IR COMPENSATION MODE) sets the IR Compensation mode. If "n" is omitted, the command requests the Model 273A to report this information. The codes are:

| n | COMPENSATION MODE |
|---|---|
| 0 | NONE |
| 1 | POS. FEEDBACK. Activates Positive Feedback IR Compensation and sets the IR compensation resistance to the value specified by the SETIR command. See discussion of SETIR command before applying IRMODE 1. **Note:** Positive feedback is not available in the galvanostatic mode. |
| 2 | CURRENT INTERRUPT. Activates periodic Current Interrupt IR Compensation. If single interrupts via DORUPT command are wanted, use IRMODE 4). |
| 3 | Periodic CURRENT INTERRUPTS performed, but E not corrected. Required to store the error values, if desired, as specified with the SIE command. |
| 4 | Prepare for Current Interrupt but don't do it (necessary preliminary to using DORUPT command). |

**SETIR n1 n2 [0 0]:** A SET/READ command, SETIR specifies the value of $R_u$ in ohms, necessary for doing Positive Feedback IR Compensation. If the operand is omitted, the command requests that the Model 273A report this information. The operand ranges are:

n1 range: 0 to 2000
n2 range: -3 to 12

The two together define the Uncompensated Resistance. For example, a mantissa of 436 and an exponent of -1 could be used to specify a resistance of 43.6Ω.

If the specified value is too low, $R_U$ will not be adequately compensated. If it is too high, the system will be unstable and may even oscillate.

Resistance values specified have a resolution limit and a range limit determined by the size of the current measuring resistor associated with a given current range. The changing resolution can affect the value applied if the current range changes. The resolution limits and the maximum compensation level as a function of current range are given below.

| CURRENT RANGE | MAXIMUM COMPENSATION | RESOLUTION LIMIT |
|---|---|---|
| 1 A | 2 Ω | 1 mΩ |
| 100 mA | 20 Ω | 10 mΩ |
| 10 mA | 200 Ω | 100 mΩ |
| 1 mA | 2 kΩ | 1 Ω |
| 100 µA | 20 kΩ | 10 Ω |
| 10 µA | 200 kΩ | 100 Ω |
| 1 µA | 2 MΩ | 1 kΩ |
| 100 nA | 20 MΩ | 10 kΩ |

Consider how the resolution limit affects the accuracy of the programmed value. For example, assume a resistance of 1.234 kΩ is specified on the 100 µA range. On that range the resistance resolution is 10Ω, giving an actual programmed resistance of 1230Ω. In other words, the programmed resistance differs from that specified by 4Ω.

Let us continue with this example to see how error due to the resolution limit can occur when the current range changes. If, during the experiment, the current range changes to 1 mA, where the resistance resolution is 1Ω, the actual programmed resistance will change to 1234Ω. The improved resolution allows the actual programmed value to be identically that originally specified. Although this change could be a problem in some situations, it will usually be relatively minor.

The real problem occurs when shifting to a more sensitive range. For instance, if, in the example, the current range were to shift to 10 µA, where the resolution limit is 100Ω, the actual programmed resistance will become 1200 ohms because the value can only be represented to the nearest 100Ω. In other words, the error will now total 34Ω. Similarly, a further shift to 1 µA would give a resistance of 1000Ω (234Ω error) and a shift to 100 nA would give a resistance of 0Ω (1234Ω error).

**Note:** Should one later shift to less sensitive ranges, the error will be successively reduced with each current range step.

Clearly, operators need to be mindful of the resolution limit for the "setup" current range and should specify a resistance appropriate to that limit. Also, if more sensitive ranges are used during the experiment, users will have to be mindful of the impact the changing resolution will have on the programmed resistance and of the possible consequences of large resistance errors.

If SETIR is applied without an operand, the previously specified value will be reported. The COMP command can be used to read the actual compensation resistance, which, as discussed in the previous paragraphs, can differ from that specified.

In some situations, the value of uncompensated resistance is not known in advance, but must be determined experimentally. A suitable procedure is provided in Subsection 5.2K, IR COMPENSATION, in the Model 273A Instruction Manual.

**COMP n1 n2:** A READ ONLY command, COMP causes the Model 273A to report the actual resistance compensated as opposed to the entered uncompensated resistance (see discussion of SETIR command). The resistance is reported in ohms. "n1" is the mantissa and "n2" the exponent. Their ranges are:

n1 range: 0...2047
n2 range: -3...12

**Note:** COMP command applies to Positive Feedback IR Compensation only.

**Example:** A response of 400 0 indicates that the actual resistance compensated is 400Ω.

**IRPC n [100]:** A SET/READ command, IRPC (IR PERCENT) establishes the percentage correction to be applied. If n is omitted, it is a request for the Model 273A to report the set percentage factor. This command applies to current interrupt operation only.

n range: 0 to 200

**Example:** To set a correction factor of 100%, apply:

IRPC 100

**IRX n1 n2 n3:** A SET/READ command, IRX (IR EXTRAPOLATION) allows the user to specify the times that will be used to extrapolate back to the interrupt potential in determining the Current Interrupt (only) IR Compensation correction factor. The extrapolation times can be set independently for each current range. The command takes three operands. n1 selects the current range [0 = 1 A, -7 = 100 nA], while n2 and n3 select the points in time used for the extrapolation. n2 and n3 can range from 2 µs to 1997 µs, with n3 representing time elapsed since n2; n2 + n3 ≤ 1999. To request present settings, the command must include n1. A command error will be generated if IRX is sent by itself. The default values for n2 and n3 are 10 µs on the 1 A and 100 mA current ranges and 75 µs on all other current ranges. The codes are:

| OPERAND | RANGE |
|---|---|
| n1 | 0 to -7 |
| n2 | 2 to 1997 |
| n3 | 2 to 1997 |

**Note:** n2 + n3 ≤ 1999

**Examples:**

1. To request the extrapolation times for the 100 µA range, apply the command:

   IRX -4

2. To set the extrapolation times to 60 µs and 100 µs on the 1 mA range, apply the command:

   IRX -3 60 40

**DORUPT n:** An ACTION READ command, DORUPT (DO CURRENT INTERRUPT) initiates an immediate current-interrupt cycle. The resulting reading of the compensation potential, in mV, is reported immediately.

n range: -10000...10000

**Note:** IRMODE 4 must be invoked prior to doing a DORUPT (or series of DORUPT's).

**IRUPT n [250]:** A SET/READ command, IRUPT (INTERMITTENT CURRENT INTERRUPT) establishes periodic Current Interrupt IR Compensation cycling. IRUPT sets the time between current interrupts in number of points based on the present TMB setting. The default corresponds to a 1 s interval. If "n" is omitted, it is a request for the Model 273A to report "n." Current-interrupt cycling begins when IRMODE 2 is applied.

n range: 1 to 32767.

**Example:** If IRUPT 10 were then applied (followed by IRMODE 2), an interrupt would be performed after every ten points. If TMB is 4000, this would be every 40,000 µs or 40 ms.

**RUERR n:** A READ ONLY command, RUERR (UNCOMPENSATED RESISTANCE ERROR) causes the compensation potential value, in mV, obtained from a current interrupt to be reported. It would normally be used when IRUPT cycling is in effect (DORUPT causes the compensation potential, in mV, for the initiated interrupt to be reported immediately.)

n range: -10000...10000

**Note:** This command only applies to Current Interrupt IR Compensation.

## 2.9. Current Integration Commands

**INTEG n:** A SET or READ command, INTEG controls the /96 IR Integrator. The setting codes are:

| n | MEANING |
|---|---|
| 0 | Reset Integrator |
| 1 | Start Integrator |
| 2 | Hold Integrator |

Note that the TC command starts data collection but does not enable the integrator. In order for useful data to be acquired, an INTEG 1 command should precede the TC command.

**ITC n:** A SET or READ command that controls the time constant for the /96 IR Integrator. The time constant combines with the gain factors and the current range to determine the full-scale range of the integrator. The table below relates *n* to RC values involved, the resulting time constants (RC and effective), and the full-scale charge, assuming GIGAIN = 1 and a current range of 1 mA.

| | | TIME | CONSTANTS | FULL-SCALE |
|---|---|---|---|---|
| **n** | **R X C** | **RC** | **EFFECTIVE*** | **CHARGE**** |
| -1 | 10 MΩ x 20 nF | 200 ms | 40 ms | 400 µC |
| -2 | 1 MΩ x 20 nF | 20 ms | 4 ms | 40 µC |
| -3 | 100 kΩ x 20 nF | 2 ms | 400 µs | 4 µC |
| -4 | 10 kΩ x 20 nF | 200 µs | 40 µs | 400 nC |

**Notes:**
* Assumes GIGAIN = 1. Otherwise, Effective Time Constant = RC/(5 x GIGAIN).
**Assumes current range is 1 mA (I/E = -3).

Note that the effective time constants are five times smaller than the RC time constants due to gain factors of 2.5 and 2 in the integrator circuitry. These gain factors are always present and cannot be switched off. The GIGAIN setting will also reduce the effective time constant and full-scale charge value. For example, for ITC = -1 and GIGAIN = 10, the effective time constant is 4 ms and 40 µC on the 1 mA current range.

The effective time constant determines how long a full-scale current must persist to produce a full-scale charge value. A full-scale current that lasts 10 effective time constants will generate a full-scale charge at the A/D Converter. That is, for ITC = -1, GIGAIN = 1 and I/E = -3, a 1 mA current that lasts 400 ms (= 400 µC) will produce a full-scale A/D count (2000 counts). The Model 273A just stores the A/D counts which can be converted to charge using the following general equation:

Q = (A/D Counts/Full-Scale A/D Counts) X Charge Full Scale / GIGAIN

Charge Full Scale = (10 X RC Time Constant X Current Range)/5

The factor of 10 is due to 10 time constants of full-scale current being required to generate a full-scale charge. The factor of 5 is due to the fixed gain factor of 5 in the Integrator circuitry discussed earlier. Combining:

Q = [(A/D Counts/2000) X (10/5) X RC X Current Range / GIGAIN

Substituting 2 X $10^{ITC}$ for RC and $10^{I/E}$ for the current range, and simplifying, yields

Q = [(A/D Counts/500) X $10^{ITC}$ X $10^{I/E}$] / GIGAIN

**GIGAIN n:** A SET or READ command that sets or reads the /96 IR Integrator gain.

n range: 1...500

Note that increasing GIGAIN decreases the full-scale charge and the effective time constant for the experiment. See the ITC command description for more information.

## 2.10. Model 273A/92 Electrochemical Impedance Interface Option Commands

The Model 273A firmware can process a number of new commands that facilitate communication between the potentiostat and other devices using the Model 273A/92 Electrochemical Impedance Interface option. These commands are either wholly new or are modifications of existing commands in the Model 273A Electrochemical Command Set. The commands are described below.

**OSCIN n [0]:** This is a Set/Read command. When the operand is 1, the oscillator signal will modulate the signal sent by the Model 273A to the cell. If the operand is 0, no modulation will occur.

**OSCGAIN n [0]:** This Set/Read command determines the full-scale value controlled by the OSC command. OSCGAIN 2 sets full scale to 2 times the external input voltage. OSCGAIN 1 sets full scale to 0.2 times input voltage, and OSCGAIN 0 sets full scale to 0.02 times input voltage. For a 5 V rms input, OSCGAIN 2 sets the maximum OSC level (= OSC 4000) to 10 V rms. OSCGAIN 1 corresponds to a 1 V maximum, and OSCGAIN 0 corresponds to a 0.1 V maximum.

**OSC n [800]:** This Set/Read command controls the degree of attenuation of an ac input signal. Full-scale attenuation depends on the OSCGAIN setting. If the OSCGAIN setting is 2 and a 5 V signal is input, OSC full scale is 10 V, so 1 count = 10 V/4000 or 2.5 mV. For the same 5 V input, OSCGAIN 1 produces 0.25 mV per count and OSCGAIN 2 yields 0.025 mV per count.

**OSCDC n [0]:** This is a SET/READ command that toggles the attenuator input between ac coupling and dc coupling. DC coupling permits a dc bias to be applied to the attenuator input. AC coupling blocks any dc bias. OSCDC 0 sets the attenuator input for ac coupling. OSCDC 1 sets it for dc coupling.

**EOUTDC n [0]:** This is a SET/READ command that determines whether the rear-panel AC E OUTPUT connector outputs the full ac + dc voltage signal or only the ac component of it without the dc component. EOUTDC 0 sets the AC E OUTPUT port for ac coupling, which passes the ac signal but blocks the dc signal. EOUTDC 1 sets the AC E OUTPUT port for dc coupling, which passes the full ac + dc signal. Note that EOUTSUP functions independently of EOUTDC and can apply a dc bias (offset) whether or not the dc cell signal is output.

**IOUTDC n [0]:** This is a SET/READ command that determines whether the rear-panel AC I OUTPUT connector outputs the full ac + dc current signal or only the ac component of it without the dc component. IOUTDC 0 sets the AC I OUTPUT port for ac coupling, which passes ac current but blocks dc current. IOUTDC 1 sets the AC I OUTPUT port for dc coupling, which passes the full ac + dc current signal from the cell. Note that IOUTSUP functions independently of IOUTDC and can apply a dc bias (offset) whether or not the dc cell signal is output.

**EOUTSUP n [0]:** This is a SET/READ command used to change the dc bias at the rear-panel AC E OUTPUT connector. Note that this dc bias is applied regardless of the EOUTDC setting. Also be careful not to confuse EOUTSUP with ESUP. These two commands function independently of each other. EOUTSUP controls the dc bias on the analog output available at the AC E OUTPUT connector, while ESUP affects the dc bias applied to the digital output.

EOUTSUP can supply ±10 V of suppression. The range for the number of counts, *n*, is ±5000, with each count corresponding to 2 mV of suppression.

**IOUTSUP n [0]:** This SET/READ command determines the level of dc suppression applied to the rear-panel AC I OUTPUT connector. Note that this dc bias is applied independently of the IOUTDC setting. Also be careful not to confuse IOUTSUP with ISUP. IOUTSUP affects the *analog* output of the instrument while ISUP affects the *digital* output. See the description of the ISUP command for more information.

IOUTSUP can produce an offset equivalent to ±4 times full-scale current of the selected current range. Since *n*, the number of counts, ranges from ±8000, each count corresponds to $0.5 \times 10^{-3}$ times full-scale current. For example, on the 1 mA current scale, maximum suppression is ±4 mA and 1 count = 0.5 µA. Be aware that changing the current range will change the level of suppression.

**MIE n [1]:** A SET/READ command, MIE (MONITOR I/E) determines whether an ac-coupled E or I signal will be provided at the rear-panel MULTIPLEXED OUTPUT connector. This command and the MULTIPLEXED OUTPUT function are intended for use with Princeton Applied Research ac impedance measurement systems. Multiplexing is accomplished by alternating the operand between "1" and "2" on successive measurements. The codes are:

| **n** | **SIGNAL MONITORING CONNECTORS** |
|---|---|
| 0 | Output current at AC I OUTPUT<br>Output voltage at AC E OUTPUT |
| 1 | Output current at MULTIPLEXED OUTPUT |
| 2 | Output voltage at MULTIPLEXED OUTPUT |

## 2.11. Data Acquisition Monitoring Commands

**A/D n:** A READ command, A/D (ANALOG TO DIGITAL CONVERSION) bypasses the normal timing and control cycling and causes an IMMEDIATE A/D conversion. The resulting reading is transmitted to the host computer. This command is *not* used in normal data acquisition. The range is:

n range: -2000...2000

EGAIN and IGAIN must be taken into account (see EGAIN and IGAIN discussions).

| **PARAMETER** | **CONVERSION FORMULA** |
|---|---|
| I | I = (n/1000 * f.s.)/IGAIN |
| E | E = (n mV) / EGAIN |
| AUX | E = (n mV) / AUXGAIN |

Current readings are in current relative to full scale (Current Range in effect). Potential readings are in mV (gain of 1 or 5) or tenths of mV (gain of 10 or 50).

The parameter selection (I, E, AUX as selected by the SIE command) should be made before applying the A/D command.

**TP n1,n2,n3:** An ACTION READ command, TP (TAKE POINT) causes a single reading to be taken, with the reading value to be reported to the host computer. If TP is executed while a curve is running, it reports the most recent I and E sample values without doing any extra sampling. If TP is executed when a curve is not running, it does a sample and then reports. The point will be taken according to the set SAMPLING CONTROL, MODULATION, and BIAS operands. The Model 273A responds by reporting:

1. n1, the number of the accessed point (0 to 6143).

2. n2, the value of the "I" data (-2048 to +2047). If I is not being sampled, this value will be reported as a "0." The expression relating I to the reported number is:

   I = (#R/1000 * f.s.) / IGAIN

   where f.s. is the selected current range, and assuming IGAIN is 1.

3. n3, the value of the "E" data (-2048 to +2047). If E is not being sampled, this value will be reported as a "0." The expression relating E to the reported number is:

E = (#R mV) / EGAIN

**Example:** In response to a TP command, the Model 273A might respond:

734,789,0250

signifying that the accessed point was 734, that the I data value is 789 (.789 of current range), and that the E data value is 250 (250 mV).

Note that both TP and SP increment PNT automatically, but TP does not store the data collected. For this reason, TP and SP commands should not be mixed in an experimental run.

**SP:** An ACTION command, SP (STORE POINT) functions the same as the TP (TAKE POINT) command except that the resulting data is not reported to the host computer but simply stored in a curve. SP can be used in conjunction with PNT (POINT) to allow custom data acquisition routines to be created, as illustrated in the following discussion of the PNT command.

Note that both TP and SP increment PNT automatically, but TP does not store the data collected. For this reason, TP and SP commands should not be mixed in an experimental run.

**PNT n:** A SET/READ command, PNT (POINT) sets the address of the next point to be processed by the Model 273A. If "n" is omitted, PNT causes the Model 273A to report this information. The point in question is called the current point and determines where the I, E, AUX, or ΔE data will be stored, as well as from what location the Modulation DAC data will be fetched if the arbitrary waveform mode of modulation (MM 2) is selected. Both TP and SP increment the PNT value automatically. SP stores its data at the present PNT. Thus SP may be combined with the PNT command to cause data to be stored in non-contiguous sections of memory. PNT may be used at any time to read the number of the current point but may be used to set the number of the current point only if curve acquisition is not in progress. If curve acquisition is in progress, attempting to set the current point will cause an ACQUISITION ERROR (Error 12) to occur. The range is:

n range: 0...6143

The following example shows how SP and PNT can be combined with some other commands to do custom curve acquisition.

| | |
|---|---|
| NC | (initialize curve acquisition) |
| BIAS 1000 | (set initial desired bias) |
| SP | (take first data value and store it) |
| PNT 10 | (advance to a different point) |
| BIAS 1010 | (set bias desired for second point) |
| SP | (take second data value and store it) |

**M n1,n2,n3,n4,n5,n6:** A READ command, M (MONITOR) causes the Model 273A to report the curve acquisition status. There are six items in the response, as follows.

1. "n", Curve Acquisition in Progress: 1 if YES. 0 if NO.

2. "n2", Number of Current Sweep (1...65535).

3. "n3", Currently Accessed Point (0 to 6193).

4. "n4", Value of the Modulation Output (-8000...8000).

5. "n5", Last "I" Value Acquired. If I is not being sampled, a 0 will be sent. (-2000...2000).

6. "n6", Last "E" Value Acquired. If E is not being sampled, a 0 will be sent. (-10000...10000).

**Example:** In response to an M command, the Model 273A might send the following to the host computer.

1,3,253,2000,1000,0

This response would indicate that a curve acquisition is in progress, that sweep number three is underway, that point 253 is currently being accessed, that the modulation output is 2000 counts (25% of full scale), and that the last "I" value acquired was 1000. The final 0 indicates that "I" data only is being acquired ("E" value reported as 0) or that "E" is being sampled and that its value is "0."

**READE n:** An ACTION READ command, READE (READ POTENTIAL) causes the Model 273A to take ten E samples and report the average of these values.

n range: -10000...10000 (corresponds to -10 V to +10 V)

READE automatically adjusts the X1, X5 aspect (only) of EGAIN as required to get the most accurate reading. Thus, if EGAIN was previously set by the user to a desired value, that setting may be lost. Moreover, if the X10 gain is on (EGAIN 10 or 50), READE always will turn it off, making it necessary for you to reapply EGAIN 10 or 50 afterwards if the X10 or X50 gain level is to be restored. Note that the value reported by READE will "read true", in mV. No conversion from counts to mV is required.

**Note:** If this command is applied when a curve acquisition is in progress, an Acquisition Error (Error #12) is generated and no data are returned.

**READI n1,n2:** An ACTION READ command, READI (READ CURRENT) causes the Model 273A to take ten I samples and report their average to the computer. "n1" is the mantissa and "n2" the exponent. For example, a response of "1000 -6" would indicate a current of 1000 * $10^{-6}$ A, that is, 1 mA.

READI autoranges the I/E converter and automatically adjusts IGAIN to optimize data resolution. As a result, the IGAIN and I/E setting may change as a result of this command. Note that IGAIN settings of X10 and X50 are always reset as they are not used by READI. The READI response always "reads true", that is, it is adjusted for the IGAIN value as appropriate.

**Note 1:** If this command is applied when a curve acquisition is in progress, an Acquisition Error (Error #12) is generated and no data is returned.

**Note 2:** Because READI sets the 1/5 gain of the IGAIN function, after a sequence of SETI, READI commands, IGAIN 1 should be executed. This will re-establish the X1 gain (normal state) and ensures that the I channel will not overload if the user subsequently programs a large current from the front panel.

**READAUX n:** An ACTION command, READAUX (READ AUXILIARY A/D INPUT) causes the Model 273A to take ten samples of the AUX A/D Input potential and report their average to the computer.

n range: -10000...10000 (corresponds to -10 V to +10 V)

READAUX automatically adjusts the X1, X5 aspect of AUXGAIN as required to get the most resolution. Thus, if AUXGAIN was previously set by the user to a desired value, that setting may be lost. Note that the reported value always "reads true" in mV.

**Q n1,n2:** A READ command, Q (CHARGE) requests that the 273A report the total coulombs to the host computer. "n1" is the mantissa and "n2" the exponent, that is:

$Q = n1 \times 10^{n2}$ coulombs
n1 range:  -9999...9999
n2 range:  as required

Note that reported number is valid when a curve is not in progress. The reading will be the same as that obtained using the Output Coulombs function on the front panel.

## 2.12. Processing and Data Transfer

### 2.12.1. Curve Processing Commands

Various kinds of processing can be performed on stored curves, as described in the following discussions of these commands. In cases where the operand curve is not explicitly identified in the command, the PROCESSING CURVE is the operand curve.

Note that the MIN, MAX, and INT commands work on any integer data, including I, E, and AUX. They do *not* work on packed data, that is, data taken with AUTORANGING enabled. The related commands IMIN, IMAX, IINT, and ILOG work on packed current data only, that is, current data taken with AUTORANGING enabled.

**ADD n:** An ACTION command, ADD causes a constant defined by "n" to be added to all points in the Processing Curve. The only constraint on n other than range is that it not be so large as to cause overflow.

n range:  -32767...32767

**SUB n1,n2:** An ACTION command, SUB (SUBTRACT) causes a point-by-point subtraction of one curve from another. "n1" specifies the subtrahend curve. "n2" specifies the minuend curve. The result is stored in the minuend curve.

n1 range:  0...5
n2 range:  0...5

**Example:** The command SUB 2,1 would cause each point in MEMORY 2 to be subtracted from each point in MEMORY 1 and the result stored in MEMORY 1.

**Note:** This command`s effect is not limited to the processing curve. The only other commands not limited to the processing curve are "CLEAR" and "COPY."

**EX n1,n2:** An ACTION command, EX (EXPAND) causes all points in the Processing Curve (designated by the PCV command) to be multiplied by "n1" and divided by "n2." The only constraint on "n1" and "n2", other than range, is that they not cause memory overflow. An additional constraint on "n2" is that it *not* be "0."

n1 range:  -32767...32767
n2 range:  -32767...32767 (but not 0)

**Example:** The command: EX 1,3 would cause all points in the processing curve to be multiplied by one and divided by three, in effect reducing the size of each value in the curve by a factor of three.

Note that this process proceeds via integer division, so values of 6, 7, and 8 result in a value of 2 when using EX 1,3.

**MIN n1,n2:** An ACTION READ command, MIN is a request for the Model 273A to scan the PROCESSING CURVE (as designated by the PCV command), find the point number where the minimum occurs, and report the point number and the value stored there. MIN works on unpacked data only (AUTORANGING not enabled when data was taken), including I, E, and AUX. "n1" is the location and "n2" the value.

n1 range: 0...6145
n2 range: -32767...32767

**Example:** In response to the MIN command, the Model 273A might respond:

782,-1983

indicating that the minimum is located at point 782 and that the value stored in that point is -1983.

**IMIN n1,n2:** An ACTION READ command, IMIN causes the Model 273A to scan the PROCESSING CURVE (must contain I data) as designated by the PCV command, find the point where the minimum occurs, and report the current at that point. Two numbers (n1 and n2), the first a mantissa and the second an exponent, are reported. For example, if the IGAIN was 1 when the data was taken, a response of "1000 -6" would indicate a current of $1000 * 10^{-6}$ A, that is, 1 mA. With higher values of IGAIN, the stored values will read "proportionally higher." The necessary corrections will have to be supplied by the host computer. IMIN only works on packed data (AUTORANGING enabled when data was taken).

n1 range: -2000...2000
n2 range: -3...-10

**MAX n1,n2:** An ACTION READ command, MAX command causes the Model 273A to scan the PROCESSING CURVE (as designated by the PCV command), find the point number where the maximum occurs, and report the point number and the value stored there. MAX only works with unpacked data (AUTORANGING disabled when data was taken), including I, E, and AUX.

n1 range: 0...6143
n2 range: -32767...32767

**Example:** In response to the MAX command, the Model 273A might respond:

491,2036

indicating that the maximum is located at point 491 and that the value stored at that point is 2036.

**IMAX n1,n2:** An ACTION command, IMAX requests the Model 273A to scan the PROCESSING CURVE (I values only) as designated by the PCV command, find the point where the maximum occurs, and report the current at that point. Two numbers (n1 and n2), the first a mantissa and the second an exponent, are reported. For example, a response of "1000 - 5" would indicate a current of $1000 * 10^{-5}$ A, that is, 10 mA. The reported value only "reads true" with IGAIN = 1. With higher IGAIN's, corrections will have to be supplied by the host computer. Note that IMAX only works with packed data, that is, data taken with AUTORANGE enabled.

n1 range: -2000...2000
n2 range: -3...-10

**INT n1,n2:** An ACTION READ command, INT (INTEGRATE) reports the *sum* of all points in the processing curve. Because of integer arithmetic constraints, the number is reported in two parts, "n1" and "n2." The two must be combined by the host computer to determine the actual sum. "n1" is multiplied by 10000 and "n2" added to the product.

The expression is:

$$SUM = n2 + (n1 * 10000)$$

$$Q = (SUM/1000) * (10^{I/E}/IGAIN) * TMB * (10^{-6}\ s/\mu s) * S/P$$

INT only works for unpacked data (data taken with AUTORANGE disabled), including I, E, and AUX.

**IINT n1,n2:** An ACTION READ command, IINT (CURRENT INTEGRATE) reports the *sum* of all points in the processing curve (current only). Two numbers, n1 the mantissa and n2 the exponent, are reported. For example, a response of "10000" would indicate a current of 1000 * $10^0$ A. (current + current = current.) Coulombs can be computed from the formula:

$$Q = 10^{-6} (S/P * TMB/IGAIN) * n1 \times 10^{n2}$$

where:

Q = charge in coulombs
S/P = samples per point as set by S/P command
TMB = timebase in microseconds as set by TMB command
n1 and n2 are the responses to the IINT command

IINT works with packed data only, that is, data taken with AUTORANGE enabled. Do not use IINT with unpacked data.

**ILOG:** An ACTION command, ILOG (CURRENT LOG) requests that the Model 273A compute the log (base 10) of every point in the PROCESSING CURVE (I data values only), replace the PCV with the log data, and report the log data to the computer. Each value is stored as 1000 times the log of the current. For example, if the log of the current is -3.794, the number stored in the memory will be 3794. ILOG only works on packed data, that is, data taken with AUTORANGING enabled.

ILOG does not take into account IGAIN.

## 2.13. Data Transfer Commands

Once curves have been acquired and processing done on them, the user may wish to transmit the resulting data to the host computer. There may also be times when it is desirable to transmit curves from the computer to the Model 273A, such as to place an arbitrary modulation waveform in the SOURCE CURVE. Provision is also made for copying data from one Model 273A curve to another, and for clearing one or all curves, as desired. Both ASCII and binary transfer capabilities are provided, as defined by the following commands.

**CLR:** An ACTION command, CLR (CLEAR) clears (sets to zero) all active points of the PROCESSING CURVE.

**Note:** Points from the first point to the last point inclusive are referred to as active points.

**CLEAR:** An ACTION command, CLEAR (CLEAR ALL) clears all active points of all curves.

**Note:** This command`s effect is not limited to the processing curve. Other commands not limited to the processing curve are "SUB" and "COPY."

**DC n1,n2:** An ACTION command, DC (DUMP CURVE) specifies the points to be transmitted to the host computer on receipt of the subsequent TALK message. "n1" specifies the number of the first point to be dumped (0 to 6143). "n2" specifies the number of points to be dumped (1 to 6144). Data is dumped for the PROCESSING CURVE only. Each value is followed by a carriage return. $\overline{EOI}$ is asserted with the carriage return after the very last character of the very last value only.

**Note:** The limitation of 80 characters in the Model 273A output buffer applies to all commands except DC and BD, which do not have the output buffer constraint.

n1 range: 0...6143
n2 range: 1...6143

**Example:** The command DC 0,6 might elicit the response:

1473 1480 1485 1491 1494 1497

indicating that the value at point 0 is 1473, the value at point 1 is 1480, the value at point 2 is 1485, the value at point 3 is 1491, the value at point 4 is 1494, and the value at point 5 is 1497.

**Notes:**

1. Recall that there can be a maximum of six curves (0 through 5) and that the starting point of each curve is fixed. Consequently, to transfer a specific curve, "n1" must specify the appropriate curve starting point, as follows.

| CURVE # | STARTING POINT NUMBER |
|---|---|
| 0 | 0 |
| 1 | 1024 |
| 2 | 2048 |
| 3 | 3072 |
| 4 | 4096 |
| 5 | 5120 |

2. Current values are acquired by the Model 273A with 12 bits (0.025%) precision. If autoranging is *not* used, the data are represented as a signed integer with each data value occupying two eight-bit bytes in memory. Data values between -2047 and +2048 are possible in this system.

Current-reading data acquired with the autorange function active are packed so that the range setting for any acquired point is stored with the value stored at that point. The word width is 16 bits (D0 through D15; D0 is the LSB and D15 is the MSB.) The data value is stored as a 12 bit number in bits D0 through D11, with D11 acting as the sign bit. The code signifying the current range (codes are the same as listed for the I/E command) is stored in bits D12, D13, D14, and D15. D15 is the sign bit for the I/E Range.

```
0 or 1   X  X  X  X    X  X  X  X  X  X  X  X  X  X  X  X
  Bit   15 14 13 12   11 10  9  8  7  6  5  4  3  2  1  0
        - ----------  ----------------------------------
          I/E Range              A/D Value
```

**Example:** 1110 110101010101 = ED55 hex

Data plotting or interpretation requires that these data be converted to a real number, where the I/E range is the exponent and the A/D value is the current measured at that I/E range. The procedure in Quick BASIC to deconvolute the data is:

```
'C = A/D Counts
'E = Exponent or I/E Range
'P = Packed Integer Value
'N = Real Number Value
E = Int (P/4096)
C = P - (E * 4096)
If C > 2048 then C = C - 4096
N = (C/1000) * 10 ^ E
```

**Example:** P = ED55 hex as above. This corresponds to -4779 as a signed integer (the QB integer format).

E = Int (-4779/4096) = Int (-1.167) = -2 (not -1!)
C = -4779 - (-2) * 4096 = 3415
If C (=3415)>2048 then C = C - 24096 = 3415 - 4096
N = (-683/1000) * $10^{-2}$ = -0.683 * $10^{-2}$

**DP n:** An ACTION command, DP (DUMP POINT) causes a single point in the PROCESSING CURVE to be dumped to the computer. "n" is the point number.

n range: 0...6143

If DP is applied at a time when data are not being actively taken, the value at the specified point is reported immediately. If DP is applied when the curve is running (TAKE CURVE command was applied and the resulting data acquisition is in progress), the response does not occur until the specified point has been taken. The data value is then dumped. This feature is particularly useful in experiments where the data are to be displayed as it is acquired.

**LC n1,n2:** An ACTION command, LC (LOAD CURVE) directs the Model 273A to accept the subsequent data stream from the host computer and to load it into the PROCESSING CURVE at the points specified by "n1" and "n2". "n1" specifies the first storage location (0 to 6143). "n2" specifies the number of points to be loaded (1 to 6144). The data must be transmitted as a string of ASCII decimal numbers separated by any non-numeric character. Exactly the specified number of points must be transmitted.

Because the Model 273A input buffer holds 80 characters, no more than 80 characters can be sent on any one line. More than one line can be used, if necessary, to meet this constraint.

n1 range: 0...6143
n2 range: 1...6143

**Example:** After sending the command LC 0,8, the computer might then send the following data:

1234 2345 3456 5432 6543 0123 9834 9876

These data would be loaded into the PROCESSING CURVE starting at point 0.

If only a few points are to be loaded, the data can be on the same line as the command.

**Example:** LC 0 5 1234 2345 3456 4567 5678 <CR>

**Note:** LC is a legitimate, useful command. However, it cannot be used with the Static Interface Routines provided in Appendix C unless the data string transferred is short enough to fit on a single line (255 characters with most BASIC's). If the data occupy more than one line, there is a problem due to the way the Static Interface Routine and Model 273A interact. On sending the terminator at the end of the first line, the Static Interface Routine checks the status byte to see if the "command done" bit is set. Because the Model 273A is still waiting for additional data, the command-done bit will not be set. As a result, the routine waits indefinitely and the system hangs. To load large blocks of data with the Static Interface Routine (Appendix C), it will be necessary to break them up into small pieces, each less than 255 characters. Alternatively, you can write a transfer routine that does not have this limitation. Another possibility is to modify the Static Interface Routine so that the "Command Done" check is bypassed (see flowcharts in Appendix C). Do not make this modification unless you are experienced enough to determine all of the possible interactions that could result from any software changes performed.

**COPY n1,n2:** An ACTION command, COPY directs the Model 273A to copy the data in one curve into another one. "n1" specifies the curve to copy from (0 to 5) and "n2" specifies the curve to copy to (0 to 5).

n1 range: 0...5
n2 range: 0...5

**Example:** The command COPY 0,1 would cause the data in CURVE 0 to be copied into CURVE 1.

**Note:** This command`s effect is not limited to the processing curve. The only other commands not limited to the processing curve are "SUB" and "CLEAR."

**BD n1,n2:** An ACTION command, BD (BINARY DUMP) is used to make a binary data dump. In the case of the BD command, "n1" specifies the address (0 to 6143) of the first point to be dumped and "n2" the number of points (1 to 6144) to be dumped. The binary dump command can be used where it is necessary to execute dumps at the fastest possible speed. Data are transferred in binary with two bytes per point. The high-order byte of a given point is sent first, followed by the low-order byte.

There is no separator between points. Every byte is a data byte.

Note that binary dumps are more vulnerable to operator error than are decimal dumps. Success is most likely to be achieved if use of this routine is limited to those who are very familiar with computers and with general digital interface operations. The Static Interface Routines provided in Appendix C *cannot* be used to make binary transfers. You must supply your own transfer routine.

In this type of transfer, bytes of data are moved from the Model 273A to the computer without processing of the bytes as they are transferred. As a result, the user must accommodate the following constraints.

1. "n1" must indicate the starting point number for the curve being transferred as follows.

| CURVE # | STARTING POINT NUMBER |
|---|---|
| 0 | 0 |
| 1 | 1024 |
| 2 | 2048 |
| 3 | 3072 |
| 4 | 4096 |
| 5 | 5120 |

2. Exactly the number of points defined by "n2" must be sent. No terminator will follow the last byte of binary data. However, $\overline{EOI}$ will be asserted with the last byte and only with the last byte.

3. Data are transmitted exactly as it is stored in the Model 273A memory. Therefore, if autoranging is used, the user will have to unpack the data in the host computer. See the discussion of the DC command for details of the packing method and a suggested deconvolution routine.

4. There are two ways of aborting a binary transfer. They are to cycle the Model 273A power or to assert $\overline{IFC}$.

**BL n1,n2:** The BL (BINARY LOAD) command allows data to be transferred from the computer to the Model 273A. As with the BD command, "n1" specifies the first memory address (0 to 6143) where data are to be stored and "n2" specifies the number of points (1 to 6144) to be loaded. The same constraints apply to BL as apply to BD.

## 2.14. Communications Control and Status Commands

### 2.14.1. Communications Control Commands

**MSK n [0]:** A SET/READ command, MSK (SRQ MASK) determines which events must take place to cause a service request. If "n" is omitted, MSK is a request for the Model 273A to report the decimal value of the mask byte. The range is 0 to 255. In setting "n", each bit for which a service request is to be generated should be a "1." The other bits should be "0."

A service request will be generated ($\overline{SRQ}$ asserted) when both the Status Byte (see discussion of ST command) and the Mask Byte have a "1" in the same bit. Digitally speaking, the mask byte and the status byte are AND'ed bit by bit. Anytime the result is a "1", $\overline{SRQ}$ is asserted. The codes are the same as for the ST command.

**Example:** If the Mask Byte operand is such as to give a Mask Byte of:

10000001

a service request will be generated anytime bit 0 or bit 7 of the STATUS BYTE is a "1." Service requests will not occur for any of the other bits whatever of the STATUS BYTE status.

**Note:** If the mask is coded to assert $\overline{SRQ}$ on detecting an OVERLOAD YES state, and an overload in fact occurs, $\overline{SRQ}$ will be asserted *and* the Overload Bit (only) in the SRQ MASK byte will be cleared. This ensures that the overload condition will cause *one* service request to be generated, and not a continual stream of service requests. When the overload has been corrected, it will be necessary to restore the SRQ MASK.

**DD n [44]:** A SET command, DD (DEFINE DELIMITER) specifies the ASCII character to be sent by the Model 273A as the delimiter between two numbers. The default or power-up delimiter is the comma. Any other ASCII character can be specified by "n" in the range of 0 to 255. Note that this command governs only the delimiter for transmitted characters. The Model 273A will accept any non-numeric printing character, with the exception of the minus sign, as a delimiter.

**Example:** DD 59 specifies the semicolon (decimal equivalent of ASCII code for a semicolon is 59) as the delimiter to be sent by the Model 273A.

### 2.14.2. Status Commands

**ST n:** The ST (DEVICE STATUS) command is a request for the Model 273A to report its status. The response is an ASCII number that is the sum of the individual bits in the status byte, unlike the serial poll response, which is an eight-bit binary number. The range is 0...255. The weighting function of each bit, with a value of 1 indicating true, is:

| BIT ITEM | BIT LOGIC NO. | DECIMAL STATE | VALUE |
|---|---|---|---|
| Command Done | 0 | 0/1 | 1 |
| Command Erro | 1 | 0/1 | 2 |
| Curve Done | 2 | 0/1 | 4 |
| Unused | 3 | 0 | 0 |
| Overload | 4 | 0/1 | 16 |
| Sweep Done | 5 | 0/1 | 32 |
| Service Requested | 6 | 0/1 | 64 |
| Output Ready | 7 | 0/1 | 128 |

**Example:** Suppose the status were:

| ITEM | TRUE/FALSE | NUMERIC VALUE |
|---|---|---|
| Command Done | FALSE | 0 |
| Command Error | FALSE | 0 |
| Curve Done | TRUE | 4 |
| Overload | FALSE | 0 |
| Sweep Done | TRUE | 32 |
| Service Requested | TRUE | 64 |
| Output Ready | TRUE | 128 |
| | Total | 228 |

In response to the ST command, the Model 273A would return the decimal number 228.

**ERR n:** A READ command, ERR (ERROR STATUS) requests the Model 273A to report its error status. The error code returned reflects the error status at the completion of the previous command, that is, at the end of the command preceding the ERR command. The codes are:

| ERROR CODE | MEANING |
|---|---|
| 0 | No Error |
| 1 | OPTION NOT INSTALLED |
| 2 | INVALID COMMAND SENT. Corrective action is to send the proper command. In GPIB operation, failure to assert $\overline{REN}$ will also generate an ERROR 2. |
| 3 | PARAMETER ERROR. Parameter error checking is performed for all commands. If an invalid parameter is detected, a PARAMETER ERROR is generated and the command is not executed. Correction action is to send the valid parameter. |

**Note:** The Model 273A is designed to accept signed integers from -32768 to 32767 (e.g., ADD n command) or unsigned numbers from 0 to 65535 (e.g., DT n command). The parameter checking function will not work properly if the user attempts to set parameters values outside these bounds.

| ERROR CODE | MEANING |
|---|---|
| 4 | COMMAND OVERRUN. Indicates that a command was sent while the unit was still executing the previous command. The corrective action is to wait until the current operation is complete before sending the command again. In GPIB communications, the serial polling capability should be used to prevent command overruns. In RS-232C communications, a prompt provides this function (see Appendices A through D). |
| 5 | NOTHING TO SAY ERROR. This error occurs if the Model 273A receives a TALK message without having anything to say. Usually it will have nothing to say because no information-request command was sent prior to the TALK message. For example, if the user wanted the Model 273A to report its sensitivity, the I/E command would have to be sent prior to the TALK message. One error that is easily made is that of sending a parameter set command instead of the corresponding information-request command. For example, should the user send the command I/E -3, the Model 273A would range to the corresponding sensitivity, instead of preparing the sensitivity setting information for transmission on receipt of the subsequent TALK message.<br><br>An ERROR 5 will also be generated during a serial poll if the host computer sends the TALK message before sending the SERIAL POLL ENABLE message, or if the host computer does not send the UNTALK message before sending the SERIAL POLL DISABLE MESSAGE. |
| 6 | NUMERIC ERROR. Occurs if numbers that do not follow the required format are applied. |
| 7 | TIME BASE (TMB) TOO SHORT. This error occurs if you ask for a TMB shorter than the total required by the options selected. |
| 8 | Not Used |
| 9 | Not Used |

| | |
|---|---|
| 10 | Not Used |
| 11 | MODE ERROR. Occurs if SETI is applied when the Model 273A is in the CONTROL E mode, or if SETE is applied when the Model 273A is in the CONTROL I mode. |
| 12 | ACQUISITION ERROR. Occurs if READE or READI are applied when a Curve Acquisition is in progress. |

**OVER n1,n2,n3:** A READ command, OVER (OVERLOAD) is a request for the Model 273A to report its overload status. The Model 273A responds with three numbers. "n1" gives the overload status at the time the command is executed. "n2" reports on the status during the interval since the last time an OVER command was applied. If an overload occurred at any time during that interval, it will be reported in the second response number, independent of whether the unit is in overload when the OVER command is applied. "n3" reports the cumulative overload status at the input to the A/D Converter. If IGAIN, EGAIN, or AUXGAIN are X5 or higher (AUXGAIN is limited to X5 maximum), overload can occur at the A/D Converter with no overload elsewhere in the Model 273A. This is true for both manually set and autoranged gain settings. Note that "n3" is only maintained in FRILLS operation. It is not maintained in SOME FRILLS or NO FRILLS operation. For a detailed discussion of the meaning of FRILLS, SOME FRILLS, and NO FRILLS, see the description of the TMB command. The codes for all three responses are:

| **n1,n2,n3** | **MEANING** |
|---|---|
| 0 | NO OVERLOAD |
| 1 | I OVERLOAD |
| 2 | E OVERLOAD |
| 3 | I and E OVERLOAD |
| 4 | AUX OVERLOAD |
| 5 | AUX & I OVERLOAD |
| 6 | AUX & E OVERLOAD |
| 7 | AUX, I, & E OVERLOAD |

Note that "n2 and "n3" are cleared on completing the execution of each "OVER" command. As a result, they reflect the cumulative overload status since the previous time an OVER command was invoked.

The Overload status is available in the Standby, Scan, and Curve Acquisition modes.

**CS n:** A READ command, CS (CELL SWITCH) requests that the Model 273A report the ON/OFF status of the Model 273A front-panel CELL ENABLE pushbutton. The codes are:

| **n** | **STATUS** |
|---|---|
| 0 | OFF |
| 1 | ON |

**DUMMY n:** A READ command, DUMMY requests the Model 273A to report the status of the toggle switch on the front panel of the Electrometer. The codes are:

| **n** | **STATUS** |
|---|---|
| 0 | NOT SET TO DUMMY |
| 1 | SWITCH SET TO DUMMY |

**FF n:** A READ command, FF (Find Frequency) requests that the 273A report the power-line frequency. A response of "0" indicates 60 Hz; "1" indicates 50 Hz.

## 2.15. Miscellaneous Commands

### 2.15.1. Cyclic Voltammetry Commands

The Model 273A can be programmed for cyclic voltammetry via the CV, SS, and MRES commands. In programming this technique, there are five parameters to consider: $E_I$ (initial potential in mV), $E_V$ (vertex potential in mV), $E_F$ (final potential in mV), RATE in mV/s, and resolution in points per volt.

Note that the Model 273A does not do true Cyclic Voltammetry but rather Cyclic Staircase Voltammetry. Measurement results obtained with Cyclic Staircase Voltammetry may differ from those obtained with Linear Cyclic Voltammetry. The better the resolution of the staircase measurement, the better the results will agree with those obtained by Linear Cyclic Voltammetry.

The CV routine proceeds as follows.

1. The parameters $E_I$, $E_V$, $E_F$, and RATE are read from the host computer and checked for validity. If a parameter is out of range, it assumes its limit value.

2. The total number of mV spanned by the waveform defined by $E_I$, $E_V$, and $E_F$ is computed. Both $E_V$ and $E_F$ must be within 2000 mV of $E_I$, allowing total spans as long as 6000 mV to be achieved.

**Example:** Given $E_I$ = 0 V, $E_V$ = +2 V, and $E_F$ = -2 V, the resulting total span would be 6000 mV (2000 mV from $E_I$ to $E_V$ and 4000 mV from $E_V$ to $E_F$).

3. The maximum possible resolution in points/volt is computed. This resolution is limited by two factors.
   a. The Modulation Digital-to-Analog Converter (DAC) is capable of 0.25 mV steps, yielding a resolution of 4000 points per volt.

   b. High scan rates cannot be done with full resolution because the Model 273A can acquire points no faster than approximately 2000 points per second.

4. The maximum possible resolution is computed as 125 to 4000 points per volt.

5. The smaller of the possible maximum resolution and the desired maximum resolution specified by the user is selected as the actual resolution.

6. The first point is set to zero.

7. Based on the resolution and the total mV, the last point is computed.

8. The time base is computed based on the total mV, the RATE, and the number of points in the curve. The time base is selected to be as small a number of microseconds as possible but not less than 500 µs.

9. Based on the rate and the time base, the samples/point is computed. This number is then multiplied by the slow-scan factor set by the SS command to get the actual samples/point. The time for one data point equals the actual samples/point times the time base.

10. The bias DAC is set based on $E_I$.

11. The ramp program is computed based on $E_I$, $E_V$, $E_F$, and total mV. The ramp program consists of only three steps that define the starting point, the first slope, and the second slope. $E_F$ may equal $E_V$, in which case the ramp program will consist of only two steps and one slope. However, $E_V$ may not equal $E_I$.

12. The MM command operand is set to "1" (RAMP PROGRAM Mode).

13. The PAM command operand is set to "1" (Multiple Samples per Reading).

A discussion of the Cyclic Voltammetry commands follows.

**CV n1,n2,n3,n4,n5:** A SET/READ command, CV (CYCLIC VOLTAMMETRY) sets the TMB, FP, LP, PAM, S/P, BIAS, MM, INITIAL and VERTEX commands. When the operands are omitted, the command is a request for the Model 273A to report the value of the parameters corresponding to "n1" through "n4", together with the Actual Resolution (n5).

**Note:** In the following operand discussions, mV and mV/s, it is assumed that the MR operand is 2, that is, that modulation full scale is 2 V.

1. "n1": $E_I$; range is ±8000 mV.

2. "n2": $E_V$; range is ±8000 mV, with the constraint that it can't be more than 2000 mV from "n1". $E_V$ must not equal $E_I$.

3. "n3": $E_F$; range is ±8000 mV, with the constraint that it can't be more than 2000 mV from "n1". $E_F$ may equal $E_V$.

**Note:** Constraints on "n1" (vertex) and "n3" (final potential) allow scans as long as 6000 mV to be obtained. For example, if $E_I$ were 0 V, "n2" could be +2000 and "n3" could be -2000 without violating the constraints, and giving a total scan length of 6000 mV.

4. "n4": RATE (mV/s); range is 1 to 8000 mV/s.

**Note:** At 8000 mV/s, the staircase effect is extreme, and results very different from those obtained with Linear Cyclic Voltammetry could occur.

5. "n5": Actual Resolution (points/volt); range is 125 to 4000.

**Note:** This parameter is not sent from the host computer to the Model 273A. Rather it is computed in the Model 273A and sent by the Model 273A to the host computer in response to the CV command applied without operands.

Note that the CV command will generate a linear ramp instead of a triangular ramp if the VERTEX and FINAL potentials are specified as the same value. However, the INITIAL and VERTEX potentials may *not* be specified as the same value. If they are, a parameter error will result.

**SS n [1]:** A SET/READ command, SS (SLOW SCAN) modifies the set Samples per Reading. As previously explained, the samples/reading is computed based on the rate and time base. This number is then multiplied by the SS operand (value range 1 to 1000) to get the actual samples reading. The SS factor is used only by the CV command. If the SS operand is omitted, the command is a request for the Model 273A to report the operand's value.

n range: 1...1000

**Note:** Since SLOW SCAN is used to slow the curve acquisition down while small resolutions (set by the MRES command) are used to speed the curve up, it makes no sense to set SLOW SCAN greater than "1" if MRES is set to less than 4000. Furthermore, such combinations can lead to arithmetic overflows when the Model 273A computes the various parameters related to the "CV" command. Therefore, if SLOW SCAN is set to a number larger than "1", MRES should be set to "4000."

**MRES n [4000]:** A SET/READ command, MRES (MAXIMUM RESOLUTION) sets the maximum resolution desired by the user in points per volt. If "n" is omitted, the command is a request for the Model 273A to report the operand's value. "n" must be between 125 and 4000. The larger the value of MRES, the larger the number of points on the curve will be.

n range: 125...4000

**Notes:**

1. MRES n must be sent before the CV command to have any effect. Sending MRES n after sending CV n1,n2,n3,n4 will *not* cause recomputation of the waveform.

2. Because SLOW SCAN is used to slow the acquisition and small resolutions (set by the MRES command) are used to speed it up, SLOW SCAN should not be greater than "1" if MRES is set to less than 4000. Improper combinations can cause arithmetic overflows when the CV parameters are computed. If SLOW SCAN is larger than "1", MRES should be set to "4000."

**Example 1:** Sending CV 0 1000 0 1000 will instruct the Model 273A to scan from 0 V to +1 V to 0 V at a rate of 1 V/s (1000 mV/s). In this case, the resolution will be set to 2000, or 0.5 mV/point.

**Example 2:** Sending CV 0 -1000 -1000 1;SS 10 will result in a linear sweep experiment from 0 V to -1 V at a rate of 0.1 mV/s.

### 2.15.2. System ID Commands

**VER n:** A READ command, VER (FIRMWARE VERSION) causes the Model 273A to report its firmware version code.

**ID n [2731]:** A READ command, ID (IDENTIFICATION) requests the Model 273A to report its model number (2731).

**OPTION n1,n2:** A READ command, OPTION reports whether an option is installed. "n1" specifies the option. "n2" is the response.

The query codes available at present are:

| **n1** | **OPTION** |
|---|---|
| 92 | Electrochemical Impedance Interface |
| 93 | Impedance Interface without suppression capabilities. |
| 96 | iR/Integrator |
| 97 | iR Compensation Option |
| 99 | Impedance Multiplexer (lacked variable attenuation levels) |

The /93, /97, and /99 are older option boards that have been replaced by the /92 and /96 option boards. Note that on the 273A the /96 is standard equipment so that Option 96 will return a "1" and Option 99 will return a "0."

The response codes are:

| **n2** | **MEANING** |
|---|---|
| 0 | Option not installed |
| 1 | Option installed |

For example, to find out if Option 92 is installed, the command would be OPTION 92. The response would either be a "0" or a "1", as appropriate.

**Note:** The iR integration function previously available as Option 273/96 for the earlier Model 273 is built in to all Model 273A units. This is why the OPTION command will return a "1" when queried with "OPTION 96."

### 2.15.3. Control Structure Commands

**BEGIN:** A CONTROL command, BEGIN is used to make an endless loop in a multiple command. This capability is useful in program development, where it allows the user to put the Model 273A in an endless loop to facilitate observing its behavior. For example, the command:

BEGIN;BIAS;MOD;AGAIN

would cause the BIAS and MODULATION DAC values to be sent continually from the Model 273A to the host computer. Such a loop must be terminated with "AGAIN."

Once such a loop has been initiated, it can only be terminated by:

1. Asserting $\overline{IFC}$.
2. Cycling the power.
3. Sending a CONTROL C (Restart character).
4. Sending a CONTROL B (Abort character).

**AGAIN:** A CONTROL command, AGAIN is used to mark the end of an endless loop as described above.

**DO n:** A CONTROL command, DO specifies the number of times a loop is to be performed. The end of a loop that is to be executed a specific number of times must be marked by the LOOP command. For example, the multiple command:

DO 1000;BIAS;MOD;LOOP

would cause the Model 273A to report the BIAS and MODULATION DAC values 1000 times. DO/LOOP commands can be terminated in the same way as BEGIN/AGAIN commands.

n range: 1...32767

**LOOP:** A CONTROL command, LOOP is used to mark the end of a loop to be executed a specific number of times.

**Note:** Neither BEGIN/AGAIN nor DO/LOOP commands may be nested. Also, if the command generates any output, that output will be dumped when the words AGAIN or LOOP are interpreted so that the Model 273A output buffer won't overflow. Loop execution will pause each time until the dump is completed.

**USR1:** A CONTROL command, USR1 (USER FUNCTION 1) allows user-defined commands to be created from other commands.

For example, the command string:

USR1 NC;CELL 1;TC;WCD;CELL 0

defines USR1 as the string which follows it. Simply sending USR1 would cause that command to be executed. Note that there must be a space between the name of the user function and the command string that defines it. A user function executes approximately ten times faster than the command string it replaces.

User functions may be part of a multiple command. The command string USR1;NC;TC would execute properly. However, the DC, LC, BD, and BL commands will not function properly in a user function and so should not be used with one.

**Note:** Four User Functions can be defined. The first, as explained above, is USR1. The other three are USR2, USR3, and USR4. The user functions may not be nested. Note that the DCL command will erase all programmed USR functions.

**P n:** An ACTION command, P (PAUSE) causes command interpretation to pause for approximately "n" seconds. "n" can take values from 0 to 65535. For example, the sequence:

BIAS 1000;P 3;BIAS 0

would cause the Bias DAC to be set to 1000 for three seconds, and then reset to zero.

### 2.15.4. Panel/Display Commands

**LREF n [0]:** A SET/READ command, BASE sets the log reference current range. The specified range is that which yields 0 V out in the LOG mode (as selected by the OUT command) at the front-panel OUTPUT connector. The codes are:

| **n** | **CURRENT RANGE** |
|---|---|
| 0 | 1 A |
| -1 | 100 mA |
| -2 | 10 mA |
| -3 | 1 mA |
| -4 | 100 µA |
| -5 | 10 µA |
| -6 | 1 µA |
| -7 | 100 nA |

**KEY n:** An ACTION command, KEY has the same effect as pressing a front-panel key. Each key is designated by a different number. Using that number as the operand duplicates the front-panel key action. The numeric designation for each key follows.

| SWITCH NAME | n | SWITCH NAME | n |
|---|---|---|---|
| E/I 1 | 1 | HIGH STABILITY | 42 |
| SCAN 2 | 2 | POTENTIOSTAT | 43 |
| 1 YES | 3 | GALVANOSTAT | 44 |
| 4 | 4 | MEASURE ONLY | 45 |
| 7 | 5 | EXT ON | 46 |
| . | 6 | HOLD/CONTINUE | 47 |
| LOCAL | 7 | FILTER | 49 |
| DELAY 1 | 9 | CURRENT INTERRUPT | 50 |
| E/I 3 | 10 | | |
| 2 NO | 11 | | |
| 5 | 12 | | |
| 8 | 13 | | |
| 0 | 14 | | |
| E/I APPLIED | 15 | | |
| SCAN 1 | 17 | | |
| EXP | 18 | | |
| 3 | 19 | | |
| 6 | 20 | | |

| SWITCH NAME | n | SWITCH NAME | n |
|---|---|---|---|
| 9 | 21 | POSITIVE FEEDBACK | 51 |
| +/- | 22 | SET IR | 52 |
| CONTINUOUS | 23 | CURRENT RANGE UP ARROWS | 53 |
| E/I 2 | 25 | CURRENT RANGE DOWN ARROWS | 54 |
| FUNCTION | 26 | START | 55 |
| SCAN SETUP UP ARROW | 27 | RESET INTEGRAL | 57 |
| STEP | 28 | SET I OFFSET | 58 |
| SEC | 29 | AUTO | 59 |
| CELL OFF | 30 | LOG | 60 |
| STOP | 31 | | |
| DELAY 2 | 33 | | |
| OVERRIDE | 34 | | |
| SCAN SETUP DOWN ARROW | 35 | | |
| PASS | 36 | | |
| mV/SEC | 37 | | |
| ENTER | 38 | | |
| ADVANCE | 39 | | |
| CELL ON | 41 | | |

**OUT n [2]:** A SET/READ command, OUT sets the Output DAC mode. When applied without an operand, this command requests the Model 273A to read the mode and report it to the computer. The codes are:

| n | MODE |
|---|---|
| 0 | NONE |
| 1 | LOG I |
| 2 | LINEAR I |
| 3 | COULOMBS |
| 4 | VALUE IN SETOUT |

**SETOUT n [0]:** A SET/READ command. SETOUT 1000 applies a 1 V signal at the OUTPUT BNC connector if OUT 4 is set. The range is +2047...-2047.

**TYPE:** The TYPE command is an Action command that allows you to print a customized message on the bottom row of the LCD display. For example, the command line

TYPE my experiment is running"

would cause the text "my experiment is running" to appear on the bottom row of the LCD display. There must be a space between the command and the string and you must include the closing quote. The string cannot contain an ASCII character lower than 32 (decimal). Note the lack of a leading quote and the presence of a trailing quote mark.

## 2.16. Interface Commands

### 2.16.1. General

**TRIG n:** An ACTION command, TRIG (TRIGGER) is used to place a pulse on the Trigger Output line (pin 4) of the rear-panel AUXILIARY INTERFACE connector. The Trig Out base line is either a logic 0 or a logic 1 (default is logic 0), according to the operand of the last applied TRIG command. TRIG 1 establishes a logic 0 base line. Once a logic 0 base line is established, subsequent TRIG 1's will cause a 10 to 20 ms wide logic 1 pulse to be generated. Similarly, once a logic 1 base line is established, TRIG 0's will cause a 10 to 20 ms wide logic 0 pulse to be generated.

**WFT n:** An ACTION command, WFT (WAIT FOR TRIGGER) allows an operation in progress to be halted. If WFT 0 is applied, operation will resume when a logic 0 is applied to the EXT TRIG line (pin 2) of the AUXILIARY INTERFACE connector. If WFT 1 is applied, operation resumes when a logic 1 is applied to the EXT TRIG line. The control level applied to EXT TRIG line must remain at the required state for at least 10 ms.

The WFT command can be used to synchronize command execution with an external event if the external event can generate the pulse required at the EXT TRIG line of the AUXILIARY INTERFACE connector. For example, suppose data acquisition isn't to begin until a specific external event occurs. One might apply the sequence:

NC;WFT 1;TC

NC prepares the 273A to take the data. WFT 1 prevents data acquisition from beginning when the following TC command is applied. When the external event occurs, a TTL logic 1 is applied to the EXT TRIG line. At that point, the previously applied TC command becomes effective and data acquisition begins.

**Note:** The level applied to the EXT TRIG line is *not* continuously sensed. Rather the line is polled periodically. The interval between polls depends on how "busy" the Model 273A is at the time, and can vary over a wide range. In other words, there can be considerable jitter in the EXT TRIG sense timing.

**PEN n [0]:** A SET/READ command, PEN causes the pen relay contacts to be closed or opened. If n is omitted, it is a request for the Model 273A to report the status of the pen-relay contacts. The pen relay also closes when [START] or [CONTINUE] are pressed on the front panel. Pressing [HOLD] or [STOP] makes the contacts open. The codes are:

| n | STATUS |
|---|---|
| 0 | OPEN |
| 1 | CLOSED |

**BIT n [0]:** A READ command, BIT causes the Model 273A to report the status of the BIT 0 IN line (pin 5) of the rear-panel AUXILIARY INTERFACE connector. The operand "n" is always 0 (there is only one input bit). Take care not to confuse this command with the BIT n1,n2 command, which is the next command discussed.

The RESPONSE codes are:

| RESPONSE | STATUS |
|---|---|
| 0 | TTL logic 0 |
| 1 | TTL logic 1 |

**BIT n1,n2:** This command sets the level on the BIT 0 OUT line (pin 7) of the rear-panel AUXILIARY INTERFACE connector. "n1" is always 0 (there is only one output bit). "n2" is either 0 or 1. **Note:** This command will not control the BIT 0 OUT line if Current Interrupt IR Compensation is active. (See description of AUXILIARY INTERFACE connector in Appendix A of the Model 273A Instruction Manual for additional information.) The "n2" codes are:

| n2 | BIT 0 OUT |
|---|---|
| 0 | 0 |
| 1 | 1 |

### 2.16.2. MODEL 303A INTERFACE COMMANDS

Use of the Model 303A Static Mercury Drop Electrode with the Model 273A requires use of the Model 407A Interface. This accessory, designed to allow the Model 273A to be used with the Model 303A Static Mercury Drop Electrode, requires two cables to make the necessary connections. One cable connects to the Model 273A AUXILIARY INTERFACE connector and the other to the Model 303A INPUT connector. The Model 407A requires +24 V at 1 A. A suitable power supply is included with it.

**DISP:** An ACTION command, DISP (DISPENSE) initiates a Model 303A DISPENSE/DISLODGE operation by providing a pulse (50 µs, logic 0) at pin 3 of the rear-panel AUXILIARY connector.
**PURGE n [0]:** A SET/READ command, PURGE controls the Model 303A's purge function via pin 8 of the rear-panel AUXILIARY connector. It allows a Model 303A purge to be turned on or off (operand applied) or the purge status to be reported (command applied without operand). The codes are:

| n | PURGE STATUS |
|---|---|
| 0 (applies TTL high) | OFF |
| 1 (applies TTL low) | ON |

**STIR n [0]:** A SET/READ command, STIR allows the Model 273A to control a Model 305 Stirrer via the Model 303A. If applied with an operand, the Stirrer is turned ON or OFF. If applied without an operand, the Model 273A reads the status of the Stirrer Control message being generated by the Model 303A. The control signal is provided on the STIR line (pin 9) of the rear-panel AUXILIARY connector. The codes are:

| **STIRRER** | **n** |
|---|---|
| 0 (applies TTL low) | OFF |
| 1 (applies TTL high) | ON |

# APPENDIX A. COMMUNICATIONS PROTOCOL AND INTERFACING

## A.1. Introduction

This appendix discusses the 273A communications protocol together with considerations germane to establishing successful communications. Unless otherwise indicated, this information applies equally to both GPIB and RS-232C communications.

## A.2. Communications Protocol

**Introduction**
A transmission to the Model 273A consists of a command followed by a space, a number or numbers if necessary, and a terminator. If a command is followed by one or more parameters, it is taken to be a request to set the associated function according to the parameters. If the command is NOT followed by one or more parameters, it is interpreted as a request to read the associated setting or status.

Each command consists of one or more upper-case letters. The letters used to delineate the command suggest the associated function. Where necessary, the letters are followed by one or more decimal numbers to complete the command. In the case of ON/OFF functions, "1" is used to denote ON and "0" to denote OFF. In the case of commands used to choose between "E" (potential) and "I" (current), "1" is used to denote "I" and "2" to denote "E."

If a command has parameters associated with it, a space MUST separate the command from the parameters. Parameters of a command will be integer numbers separated by a delimiter. With the exception of ".", "e", "+", "-", and the numerals 0 to 9, the Model 273A will recognize any non-numeric printing character as a delimiter. The Model 273A sends a comma as the delimiter unless the DD command is used to specify some other character. Note that the DD command has no effect on the characters recognized as a delimiter by the 273A. As a final note on delimiters, bear in mind that many BASIC language implementations will not accept the comma as a delimiter in Input Statements.

Numbers are transmitted as ASCII decimal with the most significant digit transmitted first. No more than five leading blanks are permitted before a command or number sent to the 273A.

**Terminator**
The terminator that follows each command message can be either a CARRIAGE RETURN or a CARRIAGE RETURN followed by a LINE FEED, as selected by Switch 8 of the rear-panel IEEE-488 Switch Assembly. In communications with a computer, the terminator must be set according to the requirements of the host computer. If the computer is expecting a line feed and <CR> has been selected, the system will "hang" as the computer waits forever for the missing line feed. If the computer expects <CR> and <CRLF> is selected, the computer will respond to the carriage return in the <CRLF> sequence all right, but then will make the line feed the first character of the next command, probably with unexpected and undesired results.

**Compound Commands**
Several commands can be sent on one line by separating them with a semicolon. For example, the sequence:

AL -5 <CR>
AR 1 <CR>
AS <CR>

could be transmitted as a single compound command with the sequence:

AL -5;AR 1;AS <CR>

However, if an error is detected at any point in the line, the rest of the command will be ignored. The size of the 273A's input buffer is 80 characters. Commands longer than 80 characters will be truncated at 80 characters. Similarly, the size of the 273A's output buffer is also 80 characters. Any compound command that generates more than 80 characters of output will cause data to be lost. NOTE: The 80 output character limit does not apply to Curve Dump commands.

**Control Characters**
Table A-1 lists the control characters to which the 273A responds and describes the instrument's response. All other control characters are ignored. A control character such as CONTROL R and CONTROL B must be sent by the host computer as a single byte without any terminator. The EOI line on the GPIB may be in either state when a control character is sent.

**Communications Control**
Commands cannot be transmitted to the 273A while it is busy executing a previous command. Should a command be sent when the 273A is busy, it will be ignored and a Command Overrun error will be generated. In GPIB communications, the Serial Poll function can be used to maintain orderly bus control. The Output Ready and Command Complete bits of the Serial Poll Status byte are provided for this purpose. In RS-232C communications, the 273A sends a prompt to the host computer at powerup and after executing each command. This prompt can be used by the host computer to prevent command overruns. Use of the Serial Poll Status byte to control GPIB communications is discussed in Appendix B. Use of the RS-232C prompt is discussed in Appendix D.

**Table A-1. Control Characters Recognized By Model 273A**

| ASCII CODE | NAME | ACTION |
|---|---|---|
| 13 | <CR> | Starts interpretation of the command if <CR> has been selected as the terminator. |
| 10 | <LF> | Starts interpretation of the command if <CRLF> has been selected as the terminator. If <CRLF> has not been selected, the Model 273A reconfigures itself to accept <CRLF> and interpretation of the command begins anyway. |
| 3 | CTRL C | RESTART: Action is the same as for power-up restart. |
| 2 | CTRL B | ABORT: Terminates the execution of a multiple command on completion of the individual command currently being carried out. Also terminates any command in its output phase. |
| 18 | CTRL R | REPEAT: Causes execution of the previous command to be repeated. |
| 19 | CTRL S | X-OFF: Prevents instrument from transmitting via the RS-232C port (only). The host computer must be ready to accept a couple of characters after it transmits X-OFF. Does not prevent characters from being echoed if the ECHO mode is ON. Transmission resumes when X-ON (CONTROL Q) is applied. |
| 17 | CTRL Q | X-ON: Once RS-232C transmission has been halted by X-OFF, transmission can be resumed by applying CONTROL Q (X-ON). |

**Model 273A Front-Panel Interface Indicators**
There are five LED Interface indicators and a switch on the front panel of the Model 273A. The indicators give the interface status. The switch facilitates transfer from REMOTE to LOCAL (front-panel) operation. A more detailed discussion of each follows.

1. TALK: This indicator lights when the 273A has output ready to be sent to the host computer. It goes out when the 273A has finished sending this output. The TALK indicator does NOT signal that the host computer has transmitted a TALK message. Rather it indicates that the Model 273A has more output to dump before the command is completed.

2. LISTEN: This indicator lights when the 273A senses the first character of a command. It remains lighted until the terminator is sensed. This indication does not mean that the host computer has transmitted the LISTEN message. Rather it means that the 273A is expecting more input before the current command is fully defined.

3. SRQ: This indicator lights when an event occurs that causes the 273A to assert $\overline{SRQ}$ (make a service request). It remains lighted until the controller (host computer) has completed a serial poll of the 273A.

4. REMOTE: This indicator lights when REMOTE operation has been established ($\overline{REN}$ asserted and LISTEN address applied). It remains lighted until LOCAL (front panel) control is restored. LOCAL can be restored by cycling the power, deasserting $\overline{REN}$, pressing the LOCAL pushbutton (providing the LLO message has not been applied), or applying the GTL message.

5. ERROR: This indicator lights to indicate that the most recently executed command contained an error. The nature of the error can be evaluated with the ERR command response (see Chapter 2).

6. LOCAL: This pushbutton transfers the 273A from REMOTE to LOCAL control if the LLO (LOCAL LOCKOUT) message has not been applied. It is effective even if $\overline{REN}$ is asserted. However, if $\overline{REN}$ continues to be asserted, the 273A will return to REMOTE control if the unit's LISTEN address is applied.

**Note:** LOCAL versus REMOTE is a GPIB consideration only. The RS-232C port is always active.

## A.3. Interfacing to a Host Computer

**Introduction**

Successful interfacing between the 273A and an external computer requires detailed knowledge of a number of areas, including:

1. The response of the instrument to incoming commands.

2. The protocol required by the instrument to enable an orderly flow of commands and responses.

3. The protocol required by the host computer.

4. The capability and requirements of the language in which the host computer is programmed.

Additionally, the user must be aware of interactions between these various factors. In the event that the system does not function as expected, there is ample room for uncertainty as to the cause of the malfunction. By taking a methodical approach, the user can eliminate much of this uncertainty.

A suggested approach to establishing successful communications follows.

1A. GPIB INTERFACE:
   a. Assert the REN line on the GPIB and send the 273A LISTEN address. The front-panel REMOTE indicator should light. If it doesn't, either the address is wrong or there is a problem with the 273A, computer, or interconnecting cable.

b. Execute a GPIB serial poll of the 273A from the host computer and see if the expected response is sent. A serial poll is the simplest possible data transfer and does not involve the 273A's microprocessor. If the serial poll status byte can be read successfully, it indicates that the 273A's GPIB address is configured properly and that the 273A is able to transmit and receive bytes.

1B. RS-232C INTERFACE: Determine whether the computer receives the RS-232 prompt (asterisk) when the 273A is first turned on. Assuming the baud rate, number of stop bits, and parity-bit configuration of the 273A and the host computer are compatible, the 273A will transmit a prompt when it is ready to receive a command. The prompt is transmitted after power-up, and after completion of any command. The prompt is provided to facilitate orderly data and command flow when controlling the system via an RS-232C interface. The prompt is normally an asterisk. If the command generates an error condition, the prompt is a question mark.

2. Send the instrument a simple command that requires a number response. The ID command, for example, should cause the 273A to respond with "2731." This test demonstrates whether bi-directional data transfer is possible.

   If this step is successful, it confirms that the terminator character has been properly specified. If it is not successful, the cause may be that the instrument is configured for a <CR> terminator and the host computer expects a <CRLF>. This is a common problem when interfacing to Hewlett-Packard equipment, which generally expects a <CRLF> terminator.

3. At this point it is often useful to know whether the host computer can receive a response consisting of more than one number. The CV command, for example, should cause five numbers to be returned.

   If the interface passes the test in (2) but fails the test in (3), it is possible that the host computer will not accept the comma as a delimiter between numbers (some BASIC's will not accept comma delimiter on input statements). This delimiter can be changed to an acceptable character, such as a space, with the DD command.

4. The host computer must pace the flow of commands so that the instrument completes the previous command before it receives a new one. For this reason, a single subroutine is recommended for sending all commands to the instrument. A string containing the commands can be passed to this subroutine. In the same vein, it is also recommended that a single subroutine be used to receive responses from the instrument. The Simple Static Interface and Device Driver Routines discussed in Appendix C can frequently be used to good advantage in establishing communications. Implementations of these routines for several common computers are provided.

   It is essential that the host computer only send a command to the Model 273A when the 273A has finished processing the previous command. In GPIB communications this is done by checking the COMMAND DONE bit of the 273A's Serial Poll byte as described in Subsection B.4 (Appendix B). In RS-232C communications, it is accomplished by always waiting for the prompt (asterisk) that the 273A transmits when it is ready for a command.

   **Note:** If the 273A detects an error condition, the prompt is a question mark. See Subsection D.2D (Appendix D).

5. A useful exercise at this point is to write a simple program that displays and updates the status of the 273A. Once this is accomplished, the resulting program structure can be used for reliable sequencing and monitoring of the 273A.

6. A final recommendation is that the controlling program be as modular and structured as possible to permit quick and logical tracing if a problem occurs.

The important point of this hierarchical process of bringing the 273A up on a host computer is that, if a problem occurs at any step, the user can go back to the previous step and figure out what caused the new problem to occur.

# APPENDIX B. GPIB INTERFACE

## B.1. Introduction

The IEEE 488-1978 Instrument Bus Standard defines a bit-parallel, byte-serial bus structure designed to allow communications between intelligent instruments. Using this standard, many instruments may be interconnected and remotely controlled or programmed. Data can be taken from, sent to, or transferred between instruments via one connector or port. The standard defines all voltage and current levels, pinouts, connector specifications, timing, and handshake requirements. As a result, it should be possible to take two or more devices equipped with a GPIB port, remove them from their shipping cartons, connect them to the bus, and expect that they will be able to communicate on the bus. However, to operate the 273A from a remote computer, it is necessary to know and use both standard GPIB and device-dependent commands as required to accomplish the intended measurement. This appendix contains detailed information about the GPIB as implemented in the Model 273A, together with a more detailed general description of the GPIB (Subsection B.7).

## B.2. GPIB Switch Assembly

**Introduction**

The GPIB Parameter Switch Assembly, located on the rear panel, contains eight miniature rocker switches, as indicated in Figure A-1. When the right end of one of these switches is pressed, the parameter value symbolized to the right is selected. Depressing the left end selects the parameter value or status symbolized to the left. With respect to logic level selections, depressing the left end of a switch selects a logic "0". Depressing the right end of switch selects a logic "1." The microprocessor checks the position of these switches at powerup only. Consequently, if the setting of any switch is changed after the unit is turned on, it will be necessary to cycle the power for a setting change to become effective. The normal position of these switches for use with Princeton Applied Research software is with switches 2, 3, and 4 ON, and with 1, 5, 6, and 7 OFF. See your software manual for its individual requirements.

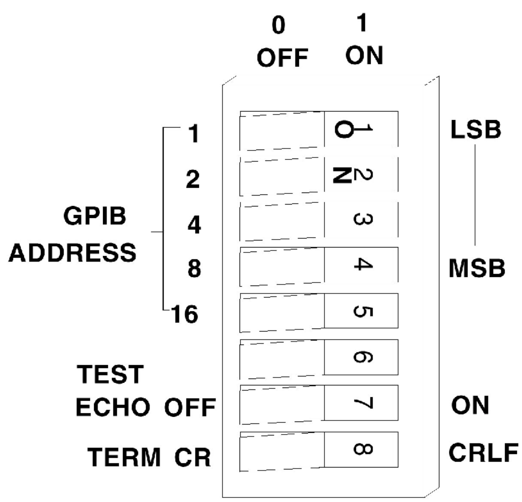


**Fig. B-1. GPIB Switch Assembly**

A detailed discussion of the GPIB switches follows.

**Switches 1 Through 5 (Addressing)**
When the controller communicates with a device on the bus, it begins by placing the address of that device on the bus with $\overline{ATN}$ asserted. Naturally, each device must "know" its own TALK and LISTEN address. In the case of the Model 273A, these addresses are set by switches 1 though 5 of the GPIB switch assembly.

**Table B-1. Address Switch Coding. ON selects logic "1"; OFF selects logic "0."**

| SWITCH BIT SEQUENCE | DECIMAL EQUIV | HEX LISTEN ADDRESS | HEX TALK ADDRESS |
|---|---|---|---|
| 00000 | 0 | 20 | 40 |
| 00001 | 1 | 21 | 41 |
| 00010 | 2 | 22 | 42 |
| 00011 | 3 | 23 | 43 |
| 00100 | 4 | 24 | 44 |
| 00101 | 5 | 25 | 45 |
| 00110 | 6 | 26 | 46 |
| 00111 | 7 | 27 | 47 |
| 01000 | 8 | 28 | 48 |
| 01001 | 9 | 29 | 49 |
| 01010 | 10 | 2A | 4A |
| 01011 | 11 | 2B | 4B |
| 01100 | 12 | 2C | 4C |
| 01101 | 13 | 2D | 4D |
| 01110 | 14 | 2E | 4E |
| 01111 | 15 | 2F | 4F |
| 10000 | 16 | 30 | 50 |
| 10001 | 17 | 31 | 51 |
| 10010 | 18 | 32 | 52 |
| 10011 | 19 | 33 | 53 |
| 10100 | 20 | 34 | 54 |
| 10101 | 21 | 35 | 55 |
| 10110 | 22 | 36 | 56 |
| 10111 | 23 | 37 | 57 |
| 11000 | 24 | 38 | 58 |
| 11001 | 25 | 39 | 59 |
| 11010 | 26 | 3A | 5A |
| 11011 | 27 | 3B | 5B |
| 11100 | 28 | 3C | 5C |
| 11101 | 29 | 3D | 5D |
| 11110 | 30 | 3E | 5E |

Each address is seven bits long. A listen address begins with "01", and a talk address with "10", with the controller supplying the "01" or "10" as required. Thus, even though we speak of a device's GPIB address, the five address switches at any time define not one address, but two. It is fairly common practice to refer to the decimal equivalent of the binary switch sequence set by the switches as the device's GPIB address. To determine the actual hex LISTEN and TALK addresses, simply precede the binary sequence by "01" or "10" as appropriate, and convert the resulting sequence to hex. An alternative technique for determining the hex LISTEN address is to begin with the decimal equivalent of the five-bit sequence, add 32, and convert to hex. Similarly, the hex TALK address can be determined by adding 64 to the decimal address and then converting to hex. Table B-1 shows the available addresses. Note that Switch 1 sets the least-significant bit and Switch 5 the most-significant bit.

**Switch 6**
This switch is reserved for future expansion.

**Switch 7 (Test Echo)**
A special GPIB TEST ECHO mode is established when Switch 7 is set to ON. When the Test Echo function is ON, every character transmitted or received via the GPIB port will be echoed to the RS-232C Interface Connector. This feature is particularly useful when developing programs. If a "dumb" CRT terminal is connected to the RS-232C Interface, the programmer will see all communications on the CRT. Such a terminal will allow direct interaction, thereby facilitating the programmer's understanding of the commands and responses. In addition, the programmer will be able to monitor the program data and intervene if necessary. The RS-232C baud rate should be high to prevent slowing down the GPIB. During normal operation, Switch 7 should be set to OFF.

**Switch 8 (Terminator)**
Switch 8 selects the terminator, that is, the character that marks the end of each transmission from the Model 273A to the host computer or from the host computer to the Model 273A. When the switch is OFF, the terminator is a CARRIAGE RETURN. When the switch is ON, it is a CARRIAGE RETURN followed by a LINE FEED. The terminator must be selected according to the requirements of the host computer. Failure to satisfy this requirement is one of the most common causes of problems in establishing communications via the GPIB. Note that, independent of Switch 8 considerations, the Model 273A accepts $\overline{EOI}$ asserted as a terminator, and will itself assert $\overline{EOI}$ with the last character in each message. The terminator requirement for some popular computers follows.

| COMPUTER | SELECT AS TERMINATOR |
|---|---|
| APPLE | <CR> |
| HP85 | <CRLF> |
| HP9825 | <CRLF> |
| HP9826 | <CRLF> |
| HP9816 | <CRLF> |
| HP9836 | <CRLF> |
| HP9845 | <CRLF> |
| TEKTRONIX 4051 | <CR> |
| FLUKE 1722A | <CRLF> |

## B.3. General Interface Management Lines

There are five general interface lines. Their function is defined by the standard and briefly discussed in Subsection B.7. Some additional comment is appropriate for them as they apply to the Model 273A.

First, the instrument recognizes $\overline{ATN}$ and $\overline{IFC}$ as defined in the standard.

$\overline{REN}$ (Remote Enable) transfers the Model 273A from local (front-panel controlled) to remote (GPIB or RS-232C port) operation. The Model 273A powers up in the local mode. When $\overline{REN}$ is asserted and the proper LISTEN address applied, it transfers to the remote mode, in which the front-panel controls are locked out. Return to local can be accomplished either by deasserting $\overline{REN}$, by applying the multi-line GTL (GO TO LOCAL) message, or by pressing the front-panel LOCAL pushbutton. Note, however, that if the LLO (LOCAL LOCKOUT) multi-line message is applied, transfer to local via the LOCAL pushbutton will be inhibited.

A command sent via the GPIB bus when the Model 273A is in the local mode (under front-panel control) will be ignored and an ERROR 2 (invalid command) generated. The Model 273A accepts $\overline{EOI}$ as a terminator for a multi-byte message and asserts $\overline{EOI}$ with the last byte of each response. The instrument asserts $\overline{SRQ}$ when the conditions established by the MSK (MASK) command are satisfied.

## B.4. Remote Messages

IEEE-488 defines a large number of multiline messages the controller can place on the bus when $\overline{ATN}$ is asserted. Of these, the Model 273A recognizes TALK, UNTALK, LISTEN, UNLISTEN, GTL (GO TO LOCAL), LLO (LOCAL LOCK OUT), DCL (DEVICE CLEAR), SPE (SERIAL POLL ENABLE), and SPD (SERIAL POLL DISABLE). Each is discussed in the following paragraphs.

1. TALK, UNTALK, LISTEN, and UNLISTEN: The TALK and LISTEN messages direct the Model 273A to assume the talker and listener states respectively. Similarly the UNTALK and UNLISTEN commands direct the Model 273A to assume the talker-idle and listener-idle states respectively.

2. GTL (GO TO LOCAL) returns the instrument to local control. Other ways of restoring local (front- panel) control are to cycle the power, deassert $\overline{REN}$, or to assert $\overline{IFC}$.

If GTL is used to return to local control, it is only necessary to place the Model 273A's listen address on the bus to re-establish control via the bus.

3. LLO (LOCAL LOCK OUT) inhibits returning the Model 273A to the local mode with the front-panel LOCAL button.

4. DEVICE CLEAR: The DEVICE CLEAR (DCL) message restores the default values of all Model 273A parameters.

5. SERIAL POLL ENABLE and SERIAL POLL DISABLE: These messages control the serial polling process. The SERIAL POLL ENABLE message tells the Model 273A to prepare to place its status byte (eight-bit binary) on the bus. This actually occurs when the talk address is applied. The SERIAL POLL DISABLE message directs the 273A to take its Serial Poll Status byte off of the bus. In making a serial poll, the correct sequence is:
   a. The host computer sends the SERIAL POLL ENABLE message ($\overline{ATN}$ asserted).
   b. The host computer sends the TALK message ($\overline{ATN}$ asserted).
   c. The host computer deasserts $\overline{ATN}$.
   d. The 273A sends the Serial Poll Status byte to the host computer.
   e. After reading the Serial Poll Status byte, the host computer reasserts $\overline{ATN}$.
   f. The host computer sends the UNTALK message ($\overline{ATN}$ asserted).
   g. The host computer sends the SERIAL POLL DISABLE message ($\overline{ATN}$ asserted).

When the host computer reads the Serial Poll Status byte, it must transmit the UNTALK and SERIAL POLL DISABLE messages and go on to its next operation. Table B-2 indicates the meaning of each bit of the serial-poll status byte.

By monitoring the COMMAND DONE and OUTPUT READY bits of the Serial Poll Response byte, the computer can obtain the information it needs to pace the flow of commands so as to proceed as rapidly as possible without interfering with the Model 273A; the polling response is handled entirely by the Model 273A's GPIB Interface chip, and does not involve the microprocessor. When the 273A has finished carrying out the previous command, Bit 0 will be set. Whenever it has output of any kind ready to send to the computer, Bit 7 will be set. By polling frequently, the computer will know as soon as possible that there is data available for processing, and/or that the Model 273A is free to accept another command.

The question of when to perform the poll is important. If the computer is dedicated solely to the 273A, simply have the computer continually poll the 273A whenever it isn't busy performing another function. The approach is incorporated into the Static Interface Routines provided in Appendix C.

**Table B-2. Serial Poll Status Byte Bit Weighting**

| BIT | MEANING | COMMENT |
|---|---|---|
| BIT 0 (LSB) | COMMAND DONE | Bit 0 is cleared when Model 273A is busy executing a command. It is set when execution of previous command is completed and the Model 273A is ready for a new command. |
| BIT 1 | COMMAND ERROR | Bit 1 is cleared if previous command did not contain an error. It is set if the previous command did contain an error. |
| BIT 2 | CURVE DONE | Bit 2 is cleared if a data acquisition is in progress. It is set if data acquisition is complete. |
| BIT 3 | NOT USED | |
| BIT 4 | OVERLOAD | Bit 4 is cleared if no OVERLOAD is detected. It is set if a Poten-tiostat overload condition is detected. |
| BIT 5 | SWEEP DONE | Bit 5 is cleared if a data acquisition sweep is in progress. It is set if the previous sweep is complete and next one hasn't started yet. |
| BIT 6 | SRQ | Bit 6 is cleared if service is not being requested. It is set if service is being requested. |
| BIT 7 | OUTPUT READY | Bit 7 is cleared if the Model 273A has no output to dump to the host computer. It is set if the Model 273A does have output to dump to the computer. |

Two approaches can be taken in a multi-tasking system. The first is to have the computer poll the 273A from time to time as part of its basic operating routine. The second is to have the 273A make a service request (assert $\overline{SRQ}$) to generate an interrupt that causes the computer to do a serial poll. The MSK (MASK) command provides the means for determining which events must occur to cause a service request.

The MASK byte and SERIAL POLL STATUS byte are AND'ed in the 273A. When any corresponding bits of both the MASK and STATUS bytes are a logic 1, $\overline{SRQ}$ is asserted. This is the mechanism by which the 273A makes a service request. The computer's response should be to do a Serial Poll, that is, to read the Serial Poll Status byte of every device on the bus. In so doing, the computer will determine which instrument requires service and take the appropriate action. The serial poll clears $\overline{SRQ}$.

**Note:** Understand that it is not necessary that the 273A make a service request (assert $\overline{SRQ}$) for the computer to do a serial poll. The computer can do a serial poll at any time. The service request is simply one means of initiating a serial poll, one best suited to use in complex systems where the computer is controlling many devices.

## B.5. Troubleshooting GPID Communications Problems

1. Check the address as set via the GPIB Switch Assembly.

2. Check the Controller's Timeout function. Many controllers (early PET's, HP, etc.) will only wait a short time for an instrument to respond on the bus. However, some 273A commands could take some time to execute.

3. Check that each command executes fully before sending another one. Some commands take time to execute and must not be interrupted while in execution. Characters received while another command is executing will cause a Error Condition (Error 4; Command Overrun). As discussed in B.4, the OUTPUT READY and COMMAND COMPLETE bits of the Serial Poll Status byte provide a convenient means for maintaining orderly, efficient communications between the 273A and the host computer.

4. Check the output format of the host computer. Sometimes extra characters may be added to the command being sent. For instance, the default output format for sending a single string from the HP 85 pads the string with blanks until it is twenty characters long. The 273A can normally ignore extra blanks, but the HP puts out too many. The solution here is to use the HP 85's PRINT USING "K" statement, which causes the HP 85 to print only the command.

5. Check the terminator options on both the controller and the 273A. The terminator selected at the 273A must be that of the host computer.

## B.6. GPIB (IEEE 488, 1978), An Overview

**Introduction**
It is not necessary to understand the material in the following paragraphs to use the GPIB interface. This information is provided for background only. You may find it helpful in the sense that any background information gives insights useful in problem solving.

**Description**
The IEEE 488-1978 Instrument Bus Standard defines a bit-parallel, byte-serial bus structure designed to allow communications between intelligent instruments. Using this standard, many instruments may be interconnected and remotely controlled or programmed. Data can be taken from, sent to, or transferred between instruments via one connector or port. The standard defines all voltage and current levels, pinouts, connector specifications, timing, and handshake requirements. As a result, it should be possible to take two or more devices equipped with a GPIB port, remove them from their shipping cartons, connect them to the bus, and expect that they will be able to communicate on the bus. However, the standard cannot guarantee that they will necessarily understand one another.

**Operating States**
With respect to a device (Model 273A) operating on the bus, there are only three operating states, CONTROLLER, TALKER, and LISTENER. The controller (computer) coordinates communications on the bus by commanding the other devices connected to it. Talkers can put information on the bus. Listeners accept messages that have been placed on the bus. Clearly, any given device may be able to support more than one of these states. The controller generally both talks and listens. Individual instruments or peripherals connected to the bus may have one or either capability. For example, the Model 273A can act as both a talker and listener, as commanded by the controller. Systems could exist where a voltmeter connected to the bus would function as a talker only. A printer on the same bus, by way of contrast, might function as a listener only, providing a permanent record of the voltage readings when commanded to do so by the controller. Although a system could contain more than one controller, only one can be active at a time. Similarly, only one talker at a time can be active. Since listening is a passive activity, more than one device is allowed to listen. Devices are assigned addresses for identification. The controller activates each device at the proper time by placing its address on the bus. Once the instrument has been "alerted" in this manner, the appropriate command can be transmitted to it. There is provision for commands that will be recognized by all devices, or by only those designated via the addressing technique.

**Number and Kinds of Lines**
The bus operates with sixteen signal lines. Of these, eight are data lines. The data lines are bi-directional and carry both data and commands. Of the remaining eight lines, three are designated the byte transfer control group. Their function is to implement the handshake required to transmit a data byte or command. Last is the general interface management group of five lines.

These are for single-line commands and status messages. They function independent of the handshake requirement. TTL levels with negative true logic are used throughout.

**Table B-3. GPIB Pinout**

| PIN # | FUNCTION |
|---|---|
| 1 | $\overline{DIO1}$ |
| 2 | $\overline{DIO2}$ |
| 3 | $\overline{DIO3}$ |
| 4 | $\overline{DIO4}$ |
| 5 | $\overline{EOI}$ |
| 6 | $\overline{DAV}$ |
| 7 | $\overline{NRFD}$ |
| 8 | $\overline{NDAC}$ |
| 9 | $\overline{IFC}$ |
| 10 | $\overline{SRQ}$ |
| 11 | $\overline{ATN}$ |
| 12 | SHIELD |
| 13 | $\overline{DIO5}$ |
| 14 | $\overline{DIO6}$ |
| 15 | $\overline{DIO7}$ |
| 16 | $\overline{DIO8}$ |
| 17 | $\overline{REN}$ |
| 18 | GND 6 |
| 19 | GND 7 |
| 20 | GND 8 |
| 21 | GND 9 |
| 22 | GND 10 |
| 23 | GND 11 |
| 24 | LOGIC GROUND |

**The Handshake**

As mentioned in the preceding paragraph, the handshake is implemented on the three byte-transfer control lines, designated $\overline{DAV}$ (data valid), ($\overline{NRFD}$ not ready for data), and $\overline{NDAC}$ (no data accepted). The following sequence occurs each time a data word or command is transferred. Figure B-2 illustrates a data transfer.

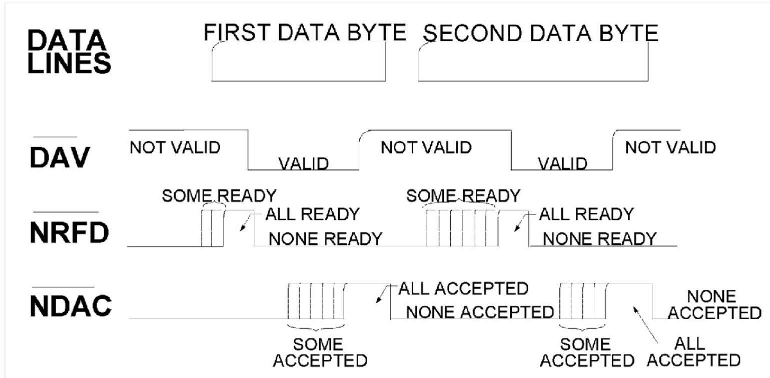


**Fig. B-2. GPIB Data Transfer**

1. Quiescently, the talker may or may not have meaningful data on the bus. As far as the upcoming transfer is concerned, it doesn't matter.

2. The talker continuously monitors the $\overline{NRFD}$ line. This line is controlled by all active listeners in such a way that it can change state (low to high) only if every active listener is ready to accept data. As long as any active listener is not ready to accept data, $\overline{NRFD}$ will be low. Once all active listeners are ready, $\overline{NRFD}$ goes high, initiating the transfer.

3. On sensing that $\overline{NRFD}$ has gone high, the talker is free to put data on the bus and pull $\overline{DAV}$ down. If the data is already on the bus (it may have been there for some time), $\overline{DAV}$ goes low immediately. If there is no data on the bus, the byte to be transmitted is placed on the bus first. Then, at least 2 s later, $\overline{DAV}$ is pulled down, signaling the active listeners that they can read the data byte.

4. On sensing that $\overline{DAV}$ is low, the active listeners know that there is a valid data word on the bus and that they are to read it. In so doing, they become "busy" and $\overline{NRFD}$ returns to the low state, where it remains until every active listener is ready to receive data again.

5. The active listeners also control the $\overline{NDAC}$ line. It is held low during the quiescent state and while the data is being accepted. The active talker monitors the $\overline{NDAC}$ line. Only when every listener has accepted the data byte does this line go high, notifying the talker that the data byte has been transmitted.

6. On sensing the positive transition of $\overline{NDAC}$, the talker releases $\overline{DAV}$, allowing it to return to the high state. The listeners sense this and respond by again pulling down $\overline{NDAC}$, thereby restoring the quiescent conditions in preparation for the next data transfer.

This sequence repeats with the transmission of every data or command byte.

**General Interface Management Group**

1. $\overline{ATN}$ (Attention): This is a critically important line. When asserted (low), it causes the active talker to relinquish control of the $\overline{DAV}$ line. The controller takes the place of the active talker, and both talkers and listeners alike accept the specific control information transmitted when $\overline{ATN}$ is asserted.

Some of the codes transmitted when $\overline{ATN}$ is asserted will have different but equally valid meanings if transmitted when $\overline{ATN}$ isn't asserted. The commands that can be transmitted when $\overline{ATN}$ is asserted, Remote Message commands, are defined by the standard. $\overline{ATN}$ is also asserted when transmitting an address. A device becomes an active talker when its talk address is placed on the bus. It becomes an active listener when its listen address is placed on the bus. Commands asserted when $\overline{ATN}$ is not asserted are device dependent.

A given device connected to the bus needn't understand all of the standard commands. Individual device manufacturers can select the commands their device will recognize. Should the others appear on the bus, they will simply be ignored.

2. $\overline{IFC}$ (Interface Clear): This line is asserted (pulled low) to override all bus activity and return the bus to a known "clear" state. Ordinarily, it is not used, but is reserved for system initialization or for situations where something has gone wrong. Any data on the bus may be lost when $\overline{IFC}$ is asserted.

3. $\overline{REN}$ (Remote Enable): This line is asserted to enable transfer of devices on the bus from local to remote control. The device does not actually go from the local control to remote control until it is addressed to listen while $\overline{REN}$ is asserted.

4. $\overline{EOI}$ (End or Identify): This line has two functions. First, it may be asserted by the talker to designate the last byte of a multi-byte message. This is referred to as the END message. Second, it can be used as part of the parallel polling process described under the following discussion of the $\overline{SRQ}$ line. **Note:** The 273A does NOT support parallel polling.

5. $\overline{SRQ}$ (Service Request): All devices on the bus share this line. Any device on the bus can assert the line, indicating to the controller that some device requires attention. For example, a voltmeter might assert $\overline{SRQ}$ after it has taken a reading to tell the controller it's ready to place the reading on the bus for transfer and further processing. All service requests look alike. The controller must use either serial polling or parallel polling to identify the devices requiring service.

In serial polling, the SERIAL POLL ENABLE message is transmitted by the controller to put all devices into the serial-poll mode. Then, every device on the bus is successively queried as to its status by the controller. In each case, the device answers with an eight-bit status byte that carries the necessary information. The controller then transmits a SERIAL POLL DISABLE message to return the device to the data mode. Assuming the devices on the bus support serial polling, by the time the controller has queried each device, it will "know" which one required the service, and will act accordingly.

In parallel polling, the controller asserts $\overline{ATN}$ and $\overline{EOI}$ simultaneously, thereby requesting a parallel poll. As many as eight devices can respond simultaneously, each on a different previously assigned data line. The data line corresponding to the device requiring service will be low, allowing it to be recognized by the controller. THE 273A DOES NOT SUPPORT PARALLEL POLLING.

**Addressing**

Addressing is an essential GPIB concept. Each device on the bus is assigned a listen address and a talk address (assuming both are relevant to the device in question). These addresses are set at the device. When the controller wishes to communicate with a specific device on the bus, it places the listen address of the device on the bus. Only the device having the corresponding address will respond to the subsequent message. Similarly, if the controller wants a device to talk, it sends the talk address of the device in question. The addressed device will transmit only until a different talker is designated.

### Summary

The preceding discussion of how the GPIB functions is by no means complete. A great deal of additional detail has to be considered by device designers. The principal responsibility of the user is to provide the controller program that coordinates all the bus activity necessary to accomplish the task at hand. For more detailed information concerning IEEE 488, 1978, the reader is advised to purchase a copy of the standard from:

IEEE
345 East 47th St.
New York, New York 10010

# APPENDIX C. GPIB INTERFACE COMMUNICATION PROGRAMS

## C.1. Simple Static Interface Routine

This program provides the software necessary to control the Model 273A from a host computer via the GPIB port. The routine follows the program flow illustrated in Figure C-1. It has been coded for several different computers. The code listings are provided in Section C.4. The utility of the routine is that it facilitates controlling the 273A from the host computer's keyboard. Every command and function can be checked and studied in a controlled, methodical manner. Actual measurements would most probably be entirely controlled by appropriate complex programs running on the host computer and not by typing individual commands at the computer's keyboard using the simple static interface routine. Nevertheless, the static routine can be used to achieve system familiarity, and variations of the routine can be readily incorporated into complex control programs with good effect.

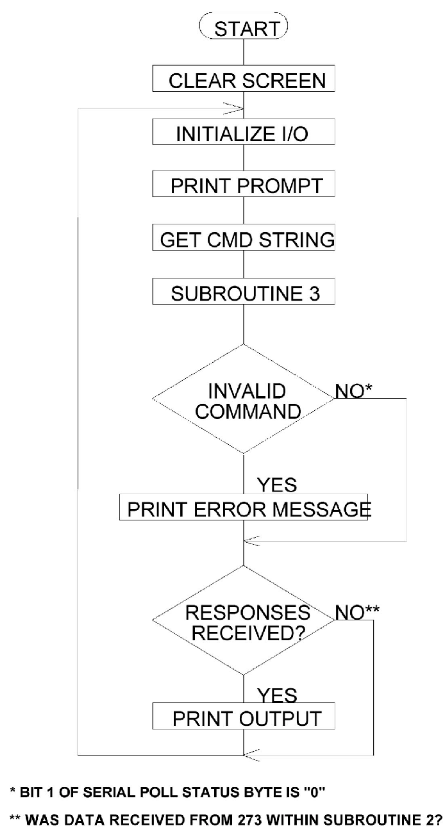


**Fig. C-1. Simple Static Interface Routine**

Note that the routine illustrated in Figure C-1 reads the Serial Poll Status byte to maintain orderly data/command flow. As indicated, Bit 0 tells the computer whether execution of the previously transmitted command was complete, Bit 1 tells the computer whether the command is valid, and Bit 7 tells the computer whether the 273A has output ready to send. As shown in the figure, the routine initially clears the video terminal of the host computer, asks the operator for the instrument's

GPIB address, and then issues the prompt ENTER COMMAND:. The program then sends the command string entered from the host computer's keyboard to the instrument. If the instrument generates an output as a result of this string, the host computer's video terminal will display the output and then the prompt, ENTER COMMAND. If there is an invalid command in the command string, the routine issues the error message COMMAND ERROR before the prompt.

Since the Simple Static Interface Routine doesn't involve complex interface and control software, programmers can use it to become familiar with the essential features of 273A software control. These include sending commands, receiving responses, and checking the instrument's SERIAL POLL status register.

## C.2. The Device Driver Subroutines

Programmers are advised to use these subroutines (Figure C-2) to handle basic input and output. With these subroutines, I/O is accomplished by putting a command string into a string variable and calling the appropriate subroutine. Subroutine 1 sends the command string to the instrument. Subroutine 2 reads the output from the instrument and places the output in the host computer's input array. Subroutine 3 is a concatenation of subroutines 1 and 2. Note that Subroutine 3 is incorporated into the Static Interface Routine (Figure C-1). In these subroutines, responses from the 273A are read on a number-by-number basis, using the OUTPUT READY bit to determine when all response have been read. To take output from the 273A on a byte-by-byte basis, use the END message ($\overline{EOI}$ asserted with the last byte of data) to determine when all responses have been read, instead of the OUTPUT READY bit.

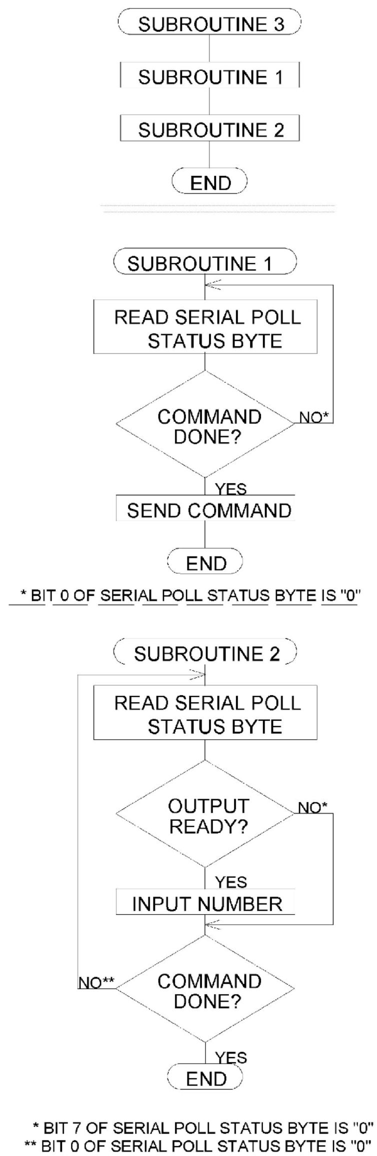


**Fig. C-2. Device Driver Subroutines Flowchart**

**Note:** If binary transfers are to be used, a special binary input routine will have to be written.

## C.3. User's Routines

The factory's engineers seek additional implementations of the Simple Static Interface Routine for other computers. Anyone willing to share an implementation is encouraged to get in touch with the factory.

## C.4. Code Listings

**IBM XT-PC**

```
1      CLEAR   ,59000!        ' National Instruments        GPIB-PC Rev. C.0 (PC2)
2      IBINIT1 = 59000!       ' BASICA Declaration File
3      IBINIT2 = IBINIT1 + 3   ' Lines 1-6
4      BLOAD "bib.m",IBINIT1
5      CALL IBINIT1 (IBFIND, IBTRG, IBCLR, IBPCT, IBSIC, IBLOC, IBPPC, IBBNA, IBONL,
       IBRSC, IBSRE, IBRSV, IBPAD, IBSAD, IBIST, IBDMA, IBEOS, IBTMO, IBEOT, IBRDF,
       IBWRTF)
6      CALL IBINIT2 (IBGTS, IBCAC, IBWAIT, IBPOKE, IBWRT, IBWRTA, IBCMD, IBCMDA, IBRD,
       IBRDA, IBSTOP, IBRPP, IBRSP, IBDIAG, IBXTRC, IBRDI, IBWRTI, IBRDIA,  IBRTIA,
       IBSTA%, IBERR%, IBCNT%)
10     REM
20     REM    IBM PC-XT Simple Static Interface Routine
30     REM
40     REM    Princeton Applied Research Corporation
50     REM    Copyright 4/25/85
60     REM
70     REM    Lines 1-6 are from the National Instruments GPIB-PC Rev. C.0
80     REM    DECL.BAS    Program
90     REM
100    REM    Configuration of National Instruments GPIB0 and DEV1
110    REM
115    REM    The GPIB Routine IBSTART must be run to install
116    REM    the GPIB drivers into the boot disk
117    REM    IBCONF must then be run to set the following settings
118    REM
120    REM    Board: GPIB0
130    REM
140    REM    Primary GPIB address..........00H
150    REM    Secondary GPIB address........NONE
160    REM    Timeout Setting...............T10s
170    REM    EOS byte......................00H
180    REM    Terminate Read on EOS.........no
190    REM    Set EOI with EOS on Write.....no
200    REM    Type of Compare on EOS........7-bit
210    REM    Set EOI w/last byte of Write..yes
220    REM
230    REM    GPIB-PC Model.................PC2
240    REM    Board is System Controller....yes
250    REM    Local Lockout an all Devices..yes
260    REM    Disable Auto Serial Polling...no
270    REM    High-Speed Timing.............no
280    REM    Interrupt Jumper Setting.......7
290    REM    Base I/O Address..............2B8H
300    REM    DMA..........................1
310    REM
320    REM
330    REM     Device: DEV1
340    REM
350    REM     Primary GPIB Address..........0E hex (14 decimal) Set This Address to the   Address
              of the Instrument Being Used
351    REM
360    REM    Secondary GPIB Address........NONE
370    REM    Timeout Setting...............T10s
380    REM    EOS byte......................0DH
```

```
390   REM     Terminate Read on EOS.........yes
400   REM     Set EOI with EOS on Write.....yes
410   REM     Type of Compare on EOS........7-bit
420   REM     Set EOI w/last byte of Write..yes
430   REM
440   REM
1000  REM
1010  REM    Initialization Section
1020  REM
1030  DIM A% (6144)
1040  CLS
1050  BDNAME$ = "DEV1"
1060  CALL IBFIND (BDNAME$, DEV1%) 'Unit descriptor returned
1070  IF DEV1% < 0 THEN GOTO 2210 'Check for error
1080  BDNAME$ = "GPIB0"
1090  CALL IBFIND (BDNAME$, GPIB0%) 'Unit descriptor returned
1100  IF GPIB0% < 0 THEN GOTO 2210 'Check for error
1110  CALL IBSIC (GPIB0%) 'Send IFC
1120  IF IBSTA% < 0 THEN GOTO 2410
1130  V%=1:CALL IBDMA(GPIB0%, V%)
1140  C$ = "DD 13" 'Define CR as Delimiter  Instrument dependent <---
1150  GOSUB 1400
1200  REM
1210  REM   Main Loop
1220  REM
1230  PRINT
1240  PRINT "ENTER COMMAND: ";
1250  INPUT C$
1260  GOSUB 1800 'Send Command and get Responses If Any
1270  IF (SPR%\2 MOD 2) = 1 THEN PRINT "COMMAND ERROR": GOTO 1300
1280  IF I%=0 THEN 1300 'Are there any Responses from the Device?
1290  GOSUB 2600 'If Response Then Display Response Strings
1300  GOTO 1200
1400  REM
1410  REM   Output Command
1420  REM
1430  GOSUB 2000 'Do serial poll
1440  IF (SPR% MOD 2) = 0 THEN 1430 'Wait for Previous Command Done
1450  CALL IBWRT (DEV1%, C$):REM Send the command
1460  IF IBSTA% < 0 THEN GOTO 2410
1470  RETURN
1600  REM
1610  REM   Get Responses
1620  REM
1630  I% = 0
1640  GOSUB 2000 'Check Output Ready and Cmd Done Bits
1650  IF ((SPR%\128) MOD 2) = 0 THEN 1710
1660  I%=I%+1
1670  Z$ = SPACE$(30)
1680  CALL IBRD (DEV1%, Z$)
1690  A%(I%) = VAL(Z$)
1700  IF IBSTA% < 0 THEN GOTO 2410 'IEEE error message
1710  IF (SPR% MOD 2) = 0 THEN 1640 'more data ???
1720  RETURN
1800  REM
1810  REM   Device Driver
1820  REM
1830  GOSUB 1400
```

```
1840  GOSUB 1600
1850  RETURN
2000  REM
2010  REM   Do a Serial Poll
2020  REM
2030  CALL IBRSP (DEV1%,SPR%)
2040  RETURN
2200  REM
2210  REM   IBFIND Error Routine
2220  REM
2230  PRINT "CALL TO IBFIND FAILED"
2240  PRINT "PLEASE CONSULT DRIVER SOFTWARE"
2250  PRINT "CONFIGURATION PROCEDURES"
2260  END
2400  REM
2410  REM   GPIB Error Checking Routine
2420  REM
2430  PRINT "A GPIB ERROR HAS OCCURRED"
2440  PRINT "ERROR CODE: "; IBERR%
2450  END
2600  REM
2610  REM   Print Response Strings on Console
2620  REM
2630  FOR N%=1 TO I%
2640  PRINT A%(N%);" ";
2650  NEXT N%
2660  RETURN
```

**APPLE II**

**Note:** APPLE GPIB card must be installed to use this routine.

```
10    REM APPLE II SIMPLE STATIC INTERFACE
20    D$=CHR$(4)
30    Z$=CHR$(26)
40    DIM A(100): REM ARRAY OF RESPONSES FROM DEVICE
100   HOME
110   PRINT "DEVICE MUST BE SET FOR"
120   PRINT "CARRIAGE RETURN TERMINATOR"
130   INPUT "ENTER DEVICE GPIB ADDRESS:  "; A9
140   INPUT "ENTER SLOT NUMBER OF APPLE GPIB CARD:  "; SL
150   PRINT
160   LA$ = "WT" + CHR$(32 + A9) + Z$ : REM GPIB LISTEN COMMAND
170   TA$ = "RD" + CHR$(64 + A9) + Z$ : REM GPIB TALK COMMAND
180   SP$ = "SP" + CHR$(64 + A9) + Z$ : REM GPIB SERIAL POLL COMMAND
190   C$="DD 13" : REM DEFINE CR AS DELIMITER
195   PRINT D$;"PR#";SL : PRINT D$;"IN#";SL : PRINT "RA" : GOSUB 1000 : PRINT D$;"PR#0" :
      PRINT D$;"IN#0"
200   REM MAIN LOOP
205   N=0
210   PRINT: INPUT "ENTER COMMAND: ";C$
220   PRINT D$; "PR#";SL : PRINT D$;"IN#";SL
230   GOSUB 1200 : REM SEND COMMAND AND GET RESPONSES IF ANY
240   PRINT D$; "PR#0" : PRINT D$; "IN#0"
250   S9=INT(S9/2) : REM CHECK FOR ERROR
260   IF S9 < > 2 * INT (S9/2) THEN PRINT "COMMAND ERROR"
300   REM PRINT RESPONSES
310   IF N = 0 THEN 350
```

```
320   FOR I = 1 TO N
330   PRINT "RESPONSE#";I;" = ";A(I)
340   NEXT I
350   GOTO 200
1000  REM OUTPUT COMMAND
1010  GOSUB 1400
1020  IF S9 = 2 * INT(S9/2) THEN 1010 : REM WAIT FOR PREVIOUS CMD      DONE
1030  PRINT LA$;C$ : REM SEND CMD
1040  RETURN
1100  REM GET RESPONSES
1110  GOSUB 1400
1120  IF S9 < 128 THEN 1140
1130  N=N+1 : PRINT TA$; : INPUT A(N)
1140  IF S9 = 2 * INT(S9/2) THEN 1110
1150  RETURN
1200  REM DEVICE CLEAR
1210  GOSUB 1000
1220  GOSUB 1100
1230  RETURN
1400  REM DO SERIAL POLL
1410  PRINT SP$;
1420  INPUT S$
1430  PRINT "UT" : REM UNTALK DEVICE
1440  REM CONVERT HEX SERIAL POLL STATUS TO A NUMBER
1450  S1 = ASC(RIGHT$(S$,1)) - 48 : IF S1>9 THEN S1 = S1-7
1460  S2 = ASC(LEFT$(S$,1)) - 48 : IF S2>9 THEN S2 = S2-7
1470  S9 = S1 + (16 * S2)
1480  RETURN
```

**HEWLETT PACKARD 85**

```
10      ! HP85 SIMPLE STATIC INTERFACE
20      RESET 7
30      SET TIMEOUT 7;5000
40      DIM A(100)
100     CLEAR
110     DISP "ENTER DEVICE GPIB ADDRESS: ";
115     INPUT A9
120     A9=A9+700
130     DISP "DEVICE MUST BE SET FOR"
140     DISP "LINEFEED TERMINATOR"
150     DISP ""
160     C$="DD 10" @ GOSUB 1000 ! LINEFEED IS DELIMITER
200     ! MAIN LOOP
210     DISP "ENTER COMMAND: ";
220     INPUT C$
230     GOSUB 1200 ! SEND COMMAND AND GET RESPONSES IF ANY
240     IF BIT (S9,1)=1 THEN DISP "COMMAND ERROR"
250     IF N=0 THEN 290
260     FOR I=1 TO N
270     DISP "RESPONSE#";I;" = ";A(I)
280     NEXT I
290     GOTO 200
1000    ! OUTPUT COMMAND
1010    GOSUB 1400 ! DO SERIAL POLL
1020    IF BIT (S9,0)=0 THEN 1010 ! WAIT FOR PREVIOUS CMD DONE
1030    OUTPUT A9 USING "K"; C$
1040    RETURN
```

```
1100    ! GET RESPONSES
1110    N=0
1120    GOSUB 1400 ! CHECK CMD DONE AND OUTPUT READY BITS
1130    IF BIT (S9,7)=0 THEN 1150
1140    N=N+1 @ ENTER A9;A(N)
1150    IF BIT (S9,0)=0 THEN 1120
1160    RETURN
1200    ! DEVICE DRIVER
1210    GOSUB 1000
1220    GOSUB 1100
1230    RETURN
1400    ! DO A SERIAL POLL
1410    S9=SPOLL (A9)
1420    RETURN
```

**HEWLETT PACKARD 9825**

```
0:      "HP9825 SIMPLE STATIC INTERFACE":
1:      "":
2:      "INITIALIZE I/O":
3:      fxd 0
4:      ent "Enter device GPIB address: ",A
5:      "HPIB ADDRESS ADDED TO DEVICE ADDRESS": 700 + A -> A
6:      dim B[500],C$[20],D$[30]
7:      1 -> N
8:      beep; dsp "Set PARC device to CRLF terminator"
9:      wait 2000; beep
10:     "main loop": ent "Enter command: ",C$
11:     gsb "Device driver"
12:     if bit(1,S)=1;beep;dsp "COMMAND ERROR";beep;wait 2000
13:     "COMMAND: " & C$ -> D$;prt D$
14:     if I=0;gto 20
15:     for K=1 to I
16:     "rsp#" & str(K) & ": " & str(B[K]) -> D$
17:     prt D$
18:     next K
19:     spc
20:     gto "main loop"
21:     " ":
22:     "SUBROUTINE 3":
23:     "Device driver":
24:     gsb "Output command"
25:     gsb "Read responses"
26:     ret
27:     " ":
28:     "SUBROUTINE 1":
29:     "Output command":
30:     gsb "Serial poll"
31:     if bit (0,S)=0; jmp -1
32:     wrt A,C$
33:     ret
34:     " ":
35:     "SUBROUTINE 2":
36:     "Read responses":
37:     0 -> I
38:     gsb "Serial Poll"
39:     if bit (7,S)=0; gto 42
40:     I+1 -> I
```

```
41:     red A,B[I]
42:     if bit (0,S)=0; gto 38
43:     ret
44:     "Serial poll":
45:     cmd 7,CHAR (24);rdb(A) -> S
46:     cmd 7, char(95)&char(25)
47:     ret
```

**HEWLETT PACKARD 9826, 9816, & 9836**

```
10      ! HP9826, 9816, OR 9836 SIMPLE STATIC INTERFACE
11      PRINT CHR$(12)
20      DIM A(4096)
100     INPUT "ENTER DEVICE GPIB ADDRESS: ",A9
110     A9=A9+700
120     CLEAR A9
130     PRINT CHR$(12)
140     PRINT "DEVICE MUST BE SET FOR CR/LF TERMINATOR."
150     PRINT " "
160     C$="DD 10"
170     GOSUB 1000
200     ! MAIN LOOP
210     LINPUT "ENTER COMMAND: ",C$
211     PRINT "ENTER COMMAND: ";C$
220     GOSUB 1200 ! SEND COMMAND AND GET RESPONSES IF ANY
230     IF BIT (S9,1)=0 THEN 250
240     PRINT "COMMAND ERROR";CHR$(7)
250     ! PRINT RESPONSES
260     IF N=0 THEN 291
270     FOR I=1 to N
280     PRINT "RESPONSE #";I;" = ";A(I)
290     NEXT I
291     PRINT CHR$(10)
300     GOTO 200
1000    ! OUTPUT COMMAND
1010    GOSUB 1400 ! DO SERIAL POLL
1020    IF BIT (S9,0)=0 THEN 1010 ! WAIT FOR PREVIOUS CMD DONE
1030    OUTPUT A9;C$
1040    RETURN
1100    ! GET RESPONSES
1101    N=0
1120    GOSUB 1400 ! CHECK CMD DONE AND OUTPUT READY BITS
1130    IF BIT (S9,7)=0 THEN 1160
1140    N=N+1
1150    ENTER A9;A(N)
1160    IF BIT (S9,0)=0 THEN 1120
1170    RETURN
1200    ! DEVICE DRIVER
1210    GOSUB 1000
1220    GOSUB 1100
1230    RETURN
1400    ! DO A SERIAL POLL (A9)
1410    S9=SPOLL(A9)
1420    RETURN
1430    END
```

**HEWLETT PACKARD 9845**

```
10      ! HP 9845 SIMPLE STATIC INTERFACE
20      ABORTIO 7 ! INTERFACE CLEAR
30      DIM A(100)
100     PRINT PAGE
110     DISP "ENTER DEVICE GPIB ADDRESS: ";
120     INPUT A9
130     A9=A9+700 ! GPIB INTERFACE IS INTERFACE 7
140     DISP "DEVICE MUST BE SET FOR LINEFEED TERMINATOR"
150     C$="DD 10" ! FOR 5205/6 ONLY, LINE 150 C$="C 10"
160     GOSUB 1000 ! LINEFEED IS DELIMITER
200     ! MAIN LOOP
210     PRINT "ENTER COMMAND: ";
220     INPUT C$
230     GOSUB 1200 ! SEND COMMAND AND GET RESPONSES IF ANY
240     IF ((S9 DIV 2) MOD 2) THEN PRINT "COMMAND ERROR"
250     IF N=0 THEN 300
260     FOR I=1 TO N
270     PRINT "RESPONSE#";I;" = ";A(I)
280     NEXT I
290     PRINT
300     GOTO 200
1000    ! OUTPUT COMMAND
1010    GOSUB 1400 ! DO SERIAL POLL
1020    IF (S9 MOD 2) = 0 THEN 1010 ! WAIT FOR PREVIOUS CMD DONE
1030    OUTPUT A9 USING "K"; C$
1040    RETURN
1100    REM GET RESPONSES
1110    N=0
1120    GOSUB 1400 ! CHECK CMD DONE AND OUTPUT READY BITS
1130    IF ((S9 DIV 128) MOD 2) = 0 THEN 1150
1140    N=N+1
1145    ENTER A9;A(N)
1150    IF (S9 MOD 2) = 0 THEN 1120
1160    RETURN
1200    REM DEVICE DRIVER
1210    GOSUB 1000
1220    GOSUB 1100
1230    RETURN
1400    ! DO A SERIAL POLL
1410    SENDBUS 7;95,24,64+A9 ! UNTALK,SPE,TALK
1420    S9=READBIN(A9)
1430    SENDBUS 7;95,25 ! UNTALK,SPD
1440    RETURN
```

**TEKTRONIX 4051**

```
10      REM TEKTRONIX SIMPLE STATIC INTERFACE
20      DIM A(100)
30      REM Maximum number of responses expected is 100
100     PAGE
110     PRINT "Enter device GPIB address: "
120     INPUT A9
130     REM Define carriage return as delimiter
140     C$="DD 13"
150     GOSUB 1000
200     REM Main loop
210     PRINT "Enter command: ";
```

```
220     INPUT C$
230     GOSUB 1200
240     REM Check for error
250     S9 = INT(S9/2)
260     IF S9 = 2 * INT(S9/2) THEN 300
270     PRINT "COMMAND ERROR"
300     REM Print responses if any
310     IF N=0 THEN 350
320     FOR I=1 TO N
330     PRINT "Response#";I;" = ";A(I)
340     NEXT I
350     GOTO 200
1000    REM Output command
1010    REM Wait for previous cmd to finish
1020    GOSUB 1400
1030    IF S9 = 2 * INT (S9/2) THEN 1020
1040    PRINT @A9,32: C$
1045    REM Use of 32 suppresses secondary address
1050    RETURN
1100    REM Get responses
1110    N=0
1120    GOSUB 1400
1130    IF S9 < 128 THEN 1160
1140    N=N+1
1150    INPUT @A9,32:A(N)
1160    IF S9 = 2 * INT(S9/2) THEN 1120
1170    RETURN
1200    REM Device driver
1210    GOSUB 1000
1220    GOSUB 1100
1230    RETURN
1400    REM Do a serial poll
1410    WBYTE @95,24:
1420    WBYTE @A9+64:
1430    RBYTE S9
1440    WBYTE @95,25:
1450    RETURN
```

**FLUKE 1722A CONTROLLER**

```
10      ! FLUKE 1722A SIMPLE STATIC INTERFACE
15      DIM A (100)           ! Room for 100 ASCII responses
20      ES$=CHR$(27%)+"["     ! Escape display sequence
25      CH$=ES$+"2J"+ES$+";H"  ! Clear screen and home cursor
30      DS$=ES$+"1p"          ! Double size graphics
35      NS$=ES$+"p"           ! Normal size graphics
40      TIMEOUT 2000          ! Device must react in 2 seconds
45      K%=KEY                ! Discard old touch
50      PRINT DS$;CH$;CPOS(2,0);
55      PRINT "TOUCH THE PORT # THE PARC INSTRUMENT"
60      PRINT "IS CONNECTED TO: GP0 OR GP1"
65      K%=KEY \ IF K% = 0 THEN 65
70      IF K% = 15% THEN PT% = 0 \ GOTO 90
75      IF K% = 18% THEN PT% = 1 \ GOTO 90
80      PRINT K%              ! Missed the box
85      WAIT 1000 \ GOTO 45
90      INIT PORT PT% \ WAIT 1000
100     PRINT CH$; "TOUCH THE DEVICE GPIB ADDRESS: "
```

```
105    PRINT CPOS (2,6);" 1  2  3  4  5  6  7  8  9 10"
110    PRINT CPOS (3,6);"11 12 13 14 15 16 17 18 19 20"
115    PRINT CPOS (4,6);"21 22 23 24 25 26 27 28 29 30"
120    K%=KEY \ IF K% = 0 THEN 120
125    PRINT "YOU SELECTED ";K%
130    PRINT "PRESS Y TO GO ON; ELSEWHERE TO RE-ENTER"
135    K1%=KEY \ IF K1% = 0 THEN 135
140    IF K1% < > 41% THEN 100
145    AD% = K% + PT% * 100
155    PRINT DS$;CPOS(2,7);"INSTRUMENT MUST BE SET FOR"
160    PRINT CPOS(3,7);"    LINE FEED TERMINATOR"; CHR$(7)
165    C$ = "DD 10"          ! Line feed is separator of multi-responses
170    GOSUB 1000 \ WAIT 4000 \ PRINT NS$
200    ! MAIN LOOP
210    PRINT "ENTER INSTRUMENT COMMAND: ";
220    INPUT C$
230    GOSUB 1200            ! Send command and get any responses
240    IF (SP% AND 2%) THEN PRINT "COMMAND ERROR" + CHR$(7)
250    IF IX%  = 0% THEN 290  ! No responses
260    FOR I% = 1% TO IX%
270    PRINT "RESPONSE#";I%;" = ";A(I%)
280    NEXT I%
290    GOTO 200
1000   ! OUTPUT COMMAND SUBROUTINE
1010   GOSUB 1400            ! Check if instrument is busy
1020   IF (SP% AND 1%) = 0% THEN 1010  ! wait till free
1030   PRINT @AD%, C$        ! Transmit command
1040   RETURN
1100   ! RESPONSE INPUT SUBROUTINE - NOT FOR CURVE TRANSFER
1105   IX%=0%
1110   GOSUB 1400            ! Check instrument status
1120   IF (SP% AND 128%) = 0% THEN 1150
1130   IX% = IX% + 1%
1140   INPUT @AD%, A(IX%)     ! Read responses
1150   IF (SP% AND 1%) = 0% THEN 1110  !If no more output then command done
1160   RETURN
1200   ! Instrument driver to send command AND get responses
1201   ! To just send commands use the subroutine at 1000
1210   GOSUB 1000
1220   GOSUB 1100
1230   RETURN
1400   ! DO A SERIAL POLL
1410   SP% = SPL(AD%)
1420   RETURN
```

# APPENDIX D. RS-232C INTERFACE

## D.1. Introduction

RS-232C is an industry standard serial data communications interface. It is widely used for communications between digital devices such as computers, terminals, printers, and telephone links. RS-232C is capable of bidirectional data transfer between digital devices.

Both devices must be equipped with an RS-232C interface for data transfer. Virtually every modern computer has RS-232C capability as standard or optional equipment. RS-232C interfacing specifies data transfer as a stream of serial characters. Usually ASCII (American Standard Code for Information Interchange) coding, which defines each character as a specific set of seven bits, is used. For example, the character sequence "-0.200V" is seven characters long. The RS-232C circuit translates each of the seven characters into the proper bit sequence and transmits each character sequentially to the receiving device.

The RS-232C Standard specifies many parameters, such as voltage (logic) levels and transfer rates, for data transfer. However, some parameters have optional values that are user selectable. The user must be aware of these parameters, since all parameters must be the same on both devices for proper communication.

The user-selectable RS-232C parameters are baud rate, word length, parity, and stop-bit configuration. These four parameters must be selected to be exactly the same on both devices. Some of the parameters may be fixed by one of the devices, in which case the other device must be configured to match the fixed parameters.

If the devices are transferring data back and forth, some method of determining the readiness of the receiving device is needed, especially for rapid data transfer. "Handshaking" is a term used to describe this function. As explained in Subsection D.4, a more detailed general discussion of RS-232C, the RS-232C handshake system was designed for a specific limited application, and is not well suited to general interface use. As a result, RS-232C communications are particularly prone to handshake problems.

## D.2. RS-232C As Implemented in the 273a

### D.2.1. Introduction

A key difference between GPIB and RS-232C communications as implemented in the 273A has to do with the concept of LOCAL versus REMOTE, that is, control via a front panel or by remote messages applied to an interface port, neither of which can be simultaneously active. LOCAL versus REMOTE operation is a GPIB concept only and does not apply to the RS-232C port. The RS-232C port is always active. In principle, if GPIB REN is asserted, one could use both the GPIB and RS-232C ports simultaneously (not recommended). There is no logic in the Model 273A to arbitrate between RS-232C and GPIB communications.

Before the Serial Interface can be used, several factors have to be considered. Included are the details of the pinout, construction of the interconnecting cable, the handshake, and several user-controlled serial interface parameters, all discussed in the following paragraphs.

### D.2.2. RS-232C Switch Assembly

This rear-panel switch assembly contains eight miniature rocker switches as indicated in Figure D-1. When the right end of one of these switches is depressed, the function or value symbolized to the right of the switch is selected. Depressing the left end of one of these switches selects the function or value symbolized to the left of the switch. These switches control the user-determined

RS-232C parameters. The Baud rate, number of data bits, parity, and number of stop bits must be the same for both the 273A and the device with which it is communicating via the RS-232C bus. Each of the switch-selectable functions is discussed in the following paragraphs.

**Switch 1 (Stop Bits):** Either one or two stop bits can be selected. Two stop bits gives more reliable communications at high baud rates.

**Switch 2 (Echo):** Gives the user the choice of echo or no-echo operation. If echo is selected, each character received via the RS-232C port will be echoed back to the character source. The RS-232C echo is normally used only when the 273A is connected to a CRT terminal. **Note:** Do not confuse this function with the GPIB TEST ECHO.

**Switch 3 (Parity):** The user has the choice of PARITY ON or PARITY OFF. If ON, there is the further choice of ODD or EVEN parity. Parity maintenance involves keeping the total number of "1's" in the data (START and STOP bits don't count) either even or odd, as desired. If odd parity is selected, the parity bit will be "1" when that is necessary to have the total number of data bits, including the parity bit, odd. For example, if the data word in question is 0010010, the parity bit would be "1" so that there would be three "1's" in all. If the number of "1's" in the word is already odd, such as in the word 0010110, the parity bit would be a "0." Even parity is similar except that the parity bit is made "1" or "0" as necessary to keep the total number of "1's", including the parity bit, even. The utility of parity maintenance is that it provides a direct way of detecting garbled data if the computer is programmed to only accept words having the selected parity. If a word having incorrect parity is read, it is a message to the computer that data loss is occurring. Although the 273A generates the parity bit, if selected, it does not itself detect the incoming parity information.

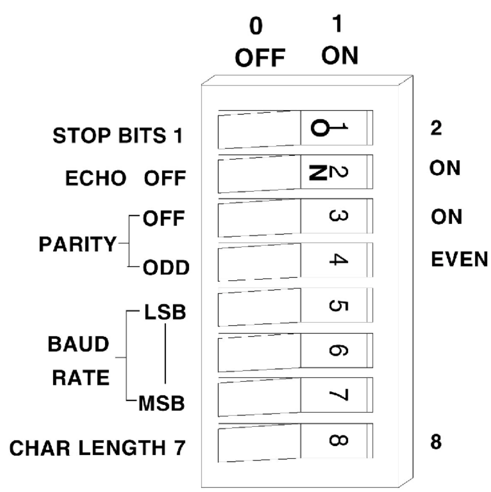


**Fig. D-1. RS-232C Switch Assembly**

**Switches 5, 6 & 7 (Baud Rate):** These two switches allow baud rates from 110 to 19.2 k baud to be selected. The codes are:

| BAUD RATE | SW7 | SW6 | SW5 |
|---|---|---|---|
| 110 | 0 | 0 | 0 |
| 300 | 0 | 0 | 1 |
| 600 | 0 | 1 | 0 |
| 1.2 k | 0 | 1 | 1 |
| 2.4 k | 1 | 0 | 0 |
| 4.8 k | 1 | 0 | 1 |
| 9.6 k | 1 | 1 | 0 |
| 19.2 k | 1 | 1 | 1 |

**Note:** At baud rates of 9.6 k and 19.2 k, the recommended parameters are a character length of eight bits, no parity, and two stop bits.

**Switch 8 (Character Length):** A character length of either seven bits or eight bits can be selected. Most communications are in ASCII, a seven-bit code. However, eight-bit coding, convenient for byte-oriented I/O, is provided. If "8" is selected, the 273A will always send a "0" as the eighth (MSB) bit when transmitting ASCII codes.

### D.2.3. Word Construction

Between transmissions, the transmit data line rests at the low voltage level. Each transmission begins with a START bit, the leading edge of which is marked by the transmit line going to the positive voltage level. The START bit is then followed by the seven data bits that define the character being transmitted. Bit 8, if transmitted, is a logic 0 for ASCII characters.

Negative true logic is used for data. A logic 1 is represented by -12 V. A logic 0 is represented by +12 V. (In no case apply voltages higher than 12 V to the RS-232C port.)

The ASCII Character code is used and the most significant bit is sent first. If the user has elected to include a parity bit, it directly follows the bits that define the character. The ASCII codes are listed in Appendix F.

The parity bit is followed by either one or two STOP bits, as selected. The STOP bits are at the negative voltage level, returning the line to the quiescent state. If the user has selected "no parity", the parity bit time slot will be deleted, and the time slot immediately following the last data bit will be allocated to a STOP bit.

### D.2.4. Prompts and Command Overruns

In RS-232C communications, command overruns (sending a command while the 273A is still busy executing the previous command) are prevented by using the prompt function provided. Assuming all RS-232C parameters have been set properly, the host computer will receive a prompt character from the 273A when it is ready for a new command. If the previously executed command generated no errors in the 273A (such as Invalid Command), the prompt character will be an asterisk (*). If the previously executed command did generate an error in the 273A, the prompt is a question mark (?). The prompt is transmitted after power up and after completion of any command. By waiting for the prompt character, Command Overrun errors will be avoided.

### D.2.5. Terminator

When two RS-232C compatible devices are communicating, their protocol requirements may well be such as to require no special terminator considerations. More frequently, however, one device or the other will have specific terminator requirements that will have to be observed for proper operation.

If the 273A is communicating with an external computer via the RS-232C port, the choice of terminator is critical. The terminator is the character that marks the end of each transmission from the Model 273A to the external computer or from the external computer to the Model 273A. The terminator must be selected according to the requirements of the host computer. Failure to satisfy this requirement is one of the most common causes of problems in establishing communications with an external computer via the RS-232C port. Note that the terminator selection is done with Switch 8 of the IEEE-488 switch assembly, also located on the rear panel. This switch gives the choice of CARRIAGE RETURN or CARRIAGE RETURN followed by a LINE FEED.

### D.2.6. Cable and Pinout

No serial interface cable is supplied with the 273A. Suitable cables and connectors are readily available from commercial sources. A standard null-modem cable will work with most micro-computers whose RS-232C port is configured as Data Terminal Equipment (DTE). The RS-232C port pinout is provided in Appendix A of the Model 273A Instruction Manual.

### D.2.7. RS-232C Summary

The preceding paragraphs contain all the information a user needs to know about the Serial Interface per se. To actually use the interface to control the 273A, the user will additionally have to understand and implement the appropriate user commands applied according to the requirements of the 273A Interface Communications Protocol. The protocol is discussed in Appendix A. The device-dependent commands recognized by the 273A are discussed in Chapter 5 of the Model 273A Instruction Manual.

### D.2.8. Troubleshooting RS-232C Communications Problems

1. Check the wiring of the RS-232C connector.

   The Transmit Data line of one device must be connected to the Receive Data line of the other.

   The Clear to Send Input line of each device must be connected to the other's Request to Send or to a voltage in the range of +5 V to +12 V. The Model 273A's Clear to Send is internally pulled up to +12 V and so may be left unconnected.

2. Check that the following settings agree on both the 273A and host computer.

   BAUD rate
   Number of character bits in the serial data word
   Logic state of the eighth character bit, if selected
   Parity settings
   Number of stop bits

3. Check that the host and 273A agree on the terminator character, either <CR> or <CRLF>. The terminator set at the 273A should be that of the host computer.

4. Avoid command overruns; make the host wait for the RS-232C prompt character before sending a new command.

5. Check that there are no timing problems. Two different timing problems are commonly encountered when RS-232C is used to interface a host computer to a device. Both are associated with responses from the device to the host.

   First, when the host sends a command to the device, the host must be ready to accept the response from the device before the device sends the first character of the response. There are three methods for solving this problem. The host's program can be written to have a fast turnaround from the portion that sends the command to the portion that receives the response. This is not always possible with a slow, high-level language. It may work at low baud rates, but it is not a guaranteed solution. Alternatively, the host can use its Request to Send line to inhibit the device's Clear to Send line until the host is ready to receive the device's output. Finally, the host can be programmed to accept the device's output stream under interrupt control. This is a reliable method but it requires intimate knowledge of the host computer's hardware and assembly language.

   Second, once the device's output stream starts, the host must be able to accept characters as fast as the device can send them. This can be assured by selecting a very low baud rate. To allow data transfer at high baud rates, it may be necessary to code the host's input routine in assembly language. Whether in high-level or assembly language, the host's program should accept the entire output stream of the device in a tight loop.

   It is advisable to start out at a very low baud rate such as 300 baud. Once the host's program is seen to be working, the baud rate can be raised as high as the host's software will allow.

The data stream can be cut off by making the Clear-to-Send line negative or by applying X-OFF (CONTROL S). If this is done, the Model 273A will finish transmitting the character being processed at that instant, and possibly one more besides. Data transmission can be resumed by setting Clear to Send positive or by sending X-ON. As long as the computer's software anticipates that one or two more characters will be forthcoming, this is a perfectly legitimate means of controlling the data flow with a Model 273A. However, since this is definitely NOT in accordance with the RS-232C Standard, users should not infer that RS-232C communications with other devices can be controlled in this way.

6. In the case of intractable problems, it might prove helpful to monitor the controller's output line with an oscilloscope. One effective approach to doing this is to program the controller to transmit the letter "U" (alternate 1's and 0's) in a repeating loop. If the scope is adjusted to trigger about ground on a positive transition and the X-axis display is calibrated so that the time per division equals one bit period, the serial data can be easily examined and decoded. Any RS-232C parameter discrepancies can then be determined and corrected. The output of the 273A can be similarly checked by turning the RS-232C ECHO on and observing the 273A echo the "U" from the host computer.

   A logic bus analyzer, such as those manufactured by Hewlett Packard, Tektronix, or Interface Technology, may also prove useful.

   One final note. If you find it necessary to change from the default settings of the computer to make it work with the 273A, be sure to bypass the computer's I/O re-initialization routines. They may restore the default values and so cause messages to be garbled.

## D.3. HISTORY AND PROBLEMS OF RS-232C

The remainder of this appendix consists of general discussion of the history and problems of RS-232C. An understanding of this material is in no way essential to using the RS-232C Interface. It is simply provided as a source of additional background for those who may be interested.

Serial data transmission is not new. Beginning with Morse Code and its use in telegraphy, the technology of serial data transmission has improved steadily to where today there is widespread reliable electronic data communications via serial coding. With the development of computers, the efficacy of serial data transmission in communications between humans and computers was quickly recognized. Serial data communications in computer applications were widely adopted. Initially, teletypewriters were almost exclusively used as combination terminal/display devices. With time, terminals and display devices advanced far beyond the teletypewriters used initially, but the serial I/O data handling characteristics of teletypewriters was maintained.

These communications reached a critical point when the distances involved became long. Clearly it was impractical to string wires between widely separated devices if it could be avoided. The ready solution was to make use of the existing telephone line system.

Unfortunately, the telephone system was not designed with serial data transmission in mind, but rather voice communications with telephones serving as the I/O devices. The situation was additionally complicated by the inclusion in the telephone system of non-wire "connections" such as microwave data links. The probable chaos of allowing direct connections between the various serial I/O devices and telephone lines was not acceptable, and some sort of standard had to be established. RS-232C is that standard. It defines signal and mechanical requirements for communications between a DTE (Data Terminal Equipment), typically a computer or a video terminal, and a DCE (Data Communications Equipment), a modem.

This standard has served well and is the basis for almost all long-distance data transmission. However, its very success has led to its adoption in applications for which it was not intended. Today there are countless communications links that use the RS-232C standard, even though the standard is not intended to apply to them. Computers, printers, terminals, measuring instruments, etc. are frequently provided with "RS-232C compatible" serial data links. However, since many of these devices are neither terminals nor modems, they cannot, by definition, comply strictly with RS-232C. Although users may expect that two pieces of equipment having RS-232C ports should successfully communicate as soon as they are connected, it frequently happens that they do not.

There are several possible reasons why successful communications may be thwarted. The first stems from the direction of the data flow as a function of the assigned interconnecting wire. The standard specifies that data transmitted from a terminal to a modem be routed via pin 2 of the terminal connector. Data received from the modem must enter the terminal via pin 3 of its interface connector. Where the device in question is neither a terminal nor a modem, the manufacturer has to arbitrarily select which pin is used to output data and which to receive data. As a result, it sometimes happens that users have to cross-wire from pin 2 to pin 3 for two pieces of "RS-232C Standard" equipment to communicate.

Handshake compatibility is another problem, sometimes a serious one. RS-232C has limited handshake requirements. The standard is concerned solely with terminal-modem communications, and the Request-to-Send and Clear-to-Send lines (pins 4 and 5 respectively) fill that requirement. They do not necessarily satisfy the handshake requirements for other equipment. According to the standard, the DTE is supposed to assert Request-to-Send when it has data to send. Having asserted Request-to-Send, it is then supposed to wait for Clear-to-Send to be asserted by the DCE. Thus far no problem.

Once data transfer begins, however, the DCE is not allowed to drop Clear-to-Send until the DTE drops Request-to-Send. Clearly there is a potential for the DCE being asked to take a longer drink than it can handle. If the Clear-to-Send line is arbitrarily used as a spigot to turn the data flow on and off (definitely NOT according to the standard), there is the question of determining just when it should switch. If it changes state when its buffer is full, it might stop in the middle of a character and garble the data. If it waits, the buffer will overflow and data will be lost. The problem is simply that RS-232C was not designed for the job it is often asked to do, even though it can generally be made to work.

**Note:** As stated in D.3, RS-232C data flow from the 273A can be cut off by bringing Clear-to-Send to its negative state, or by applying X-OFF (CONTROL S). The 273A will finish sending the character being processed, and possibly one more as well. If the host computer's software is prepared to handle these one or two additional characters, this is a perfectly legitimate way of controlling the data flow. Bear in mind, however, that controlling in this manner is not in accordance with the standard, and may not work for other RS-232C compatible devices.

Another problem has its origin in the RS-232C voltage levels. Because TTL is so ubiquitous in modern electronic designs, it is all too easy to assume that TTL devices and levels apply everywhere. However, RS-232C was around long before TTL, and its voltage levels differ from those of TTL. Negative true logic is used for data communications; positive true logic is used for commands. The levels are +3 V to +25 V for the positive logic level, and -3 V to -25 V for the negative logic level. The region between ±3 V is undefined. As a result, designers cannot use TTL devices in implementing RS-232C communications.

**Note:** The actual levels used in the Model 273A are ±12 V. Do not apply voltages higher than ±12 V to the Model 273A's RS-232C port.

Users also must take care to ascertain that both pieces of equipment operate at the same data-transmission rate (baud rate), agree on parity, have the same number of stop bits, and use the same character code (usually ASCII). A mismatch in any one of these parameters will cause data to be garbled or lost.

Between data transmissions, the data lines idle at the negative logic level. As shown in Figure D-2, each transmission begins with a start bit, the leading edge of which is defined by the data line going positive (logic 0). The start bit lasts for one bit time (the duration of each bit is set by the baud rate), and acts as a signal to the receiving device to begin sampling the level on the line at the center of each bit time interval in the sequence. Clearly, both devices must agree as to the number of bits to follow if they are to communicate successfully.

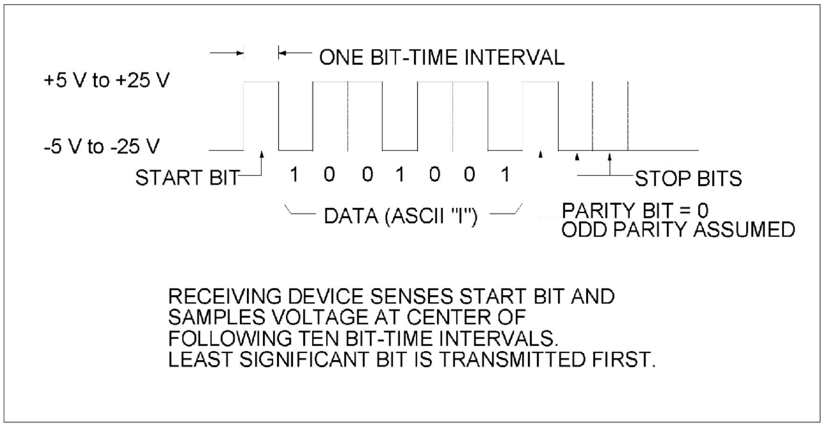


**Fig. D-2. Typical RS-232C Data Word**

The start bit is followed by the data bits, seven for ASCII, that define the character transmitted. Next is the parity bit. RS-232C allows even parity, odd parity, or no parity (no parity-bit time interval included in the sequence). However, both instruments have to be set the same with respect to the parity bit. The parity bit is then followed by either one or two stop bits. The line comes back to the negative logic level for the duration of the stop bits and continues idling at the negative level until another start bit is transmitted to mark the beginning of the subsequent communication.

Most RS-232C compatible devices have provision for user selection and control of the parity bit, the number of stop bits, and the baud rate. Most use the ASCII code. Sometimes eight bits are used instead of seven, convenient for byte-oriented I/O. Successful communications can usually be established. It's just not so straightforward as expected.

Serial interfacing is popular and reliable. However, when two pieces of equipment are to communicate serially, the communications link must be established with care and attention to detail, even though both devices have serial ports implemented "in accordance with the RS-232C Standard." Users who wish to have more detailed information about the RS-232C standard can purchase a copy of the standard from:

Electronic Industries Association
Engineering Department
2001 I Street, N.W.
Washington, DC

# APPENDIX E. WAVEFORM PROGRAMMING

## E.1. Introduction

The Model 273A Potentiostat/Galvanostat is a powerful and versatile instrument for computer-assisted electrochemistry. Its internal command set allows a wide variety of electrochemical experiments. Two 14 bit Digital-to-Analog Converters (DAC's) provide flexible and precise control of the controlled potential or current for electrochemical techniques. A 12 bit Analog-to-Digital Converter (ADC) allows rapid and convenient data acquisition.

The Model 273A's functions can be divided into the following categories:

Waveform Generation
Data Acquisition
Data Manipulation
Data Transfer

Of these functions, waveform generation is the most complex to understand, due to the great variety of techniques that can be employed to achieve the experimental goal. Some of the capabilities of data acquisition, data manipulation, and data transfer will be discussed, since all of these functions can be used in waveform generation. However, this appendix will deal primarily with waveform programming techniques.

## E.2. Model 273A Memory Usage

The Model 273A has memory enough to store 6144 data values in as many as six curves (Curve 0 through Curve 5). The maximum number of curves depends on their length, up to the maximum of six curves. If a curve is short (1024 or fewer points), the 6144-point memory is large enough to allow the maximum of six curves. At the other extreme, if a curve is longer than 3072 points, the 6144-point memory will only be large enough for one curve. Unused memory capacity is simply unavailable. For example, a 5000-point curve would leave 1144 points unused and unavailable.

For purposes of memory partitioning, the curve length is defined as LP +1, where LP is the operand of the LP (LAST POINT) command. For example, if the last point as designated by the LP command is 1024, the curve length would be considered to be 1025 points. As indicated in the table that follows, there would only be room for three curves of this length in memory. If the curve length were reduced by a single point, the number of curves that could be stored would increase from three to six, the maximum.

Curves are identified by numbers, and each curve always begins at a specific point in memory. The numbers range from 0 to 5, that is, six numbers to designate as many as the maximum of six curves. The point at which each curve begins is given in Figure E-1.

In other words, the starting location of each curve in memory is always the same. (Knowing these locations is critical when doing data dumps from the Model 273A to the host computer.) For a given curve length, as determined by the LP command, only specific curves will be available, as indicated in the list that follows. For example, if the curve length were such as to allow six curves, all six curves, 0 through 5, would be available. On the other hand, if the curve length were such as to allow three curves, the available curves would be 0, 2, 4, and the starting location of each would be as listed in Figure E-1. Curves 1, 3, and 5 would not be available and could not be designated.

| | |
|---|---|
| CURVE 0 | 0 |
| CURVE 1 | 1024 |
| CURVE 2 | 2048 |
| CURVE 3 | 3072 |
| CURVE 4 | 4096 |
| CURVE 5 | 5120 |

**Fig. E-1. Starting Memory Address For Each Curve**

| CURVE LENGTH | # AVAIL. | NUMBER(S) |
|---|---|---|
| 3073 to 6144 | 1 | 0 |
| 2049 to 3072 | 2 | 0 & 3 |
| 1025 to 2048 | 3 | 0, 2, & 4 |
| 1 to 1024 | 6 | 0, 1, 2, 3, 4, & 5 |

Each point is a 16 bit signed integer (-32767 to +32767). By making the capacity of the memory locations (nominally 32000) much larger than the full scale output of the analog-to-digital converter (nominally 2000), room is provided for sweep averaging. In the linear averaging mode, sixteen sweeps containing full-scale values could be averaged without overflowing the memory. In the exponential averaging mode, a very large number of sweeps could be averaged since the value at convergence won't exceed the value at the output of the analog-to-digital converter for that point.

Functionally, there are five kinds of curves. They are:

1. SOURCE CURVE: Data stored in the Source Curve defines a modulation waveform. If the MM 2 command has been executed, the modulation waveform stored in the SOURCE CURVE will be applied during subsequent curve acquisitions.

2. DESTINATION CURVE: This is the curve into which data will be placed during curve acquisitions. I, E, AUX, and ΔE (the current-interrupt compensation potential) could all be stored in successive curves as determined by the SIE command. The first curve is the Destination Curve. The other or others, according to the data designations made with the SIE command, are NEXT CURVES.

3. NEXT CURVE: This is the curve into which E, AUX, or E data is placed during curve acquisitions. There can be zero, one, two, or five Next Curves according to the SIE selection. NEXT CURVES, if present, always immediately follow the DESTINATION CURVE.

4. ALTERNATE CURVE: During a run comprising many sweeps, it is possible to store the measurement results obtained on some number of sweeps in one curve, and the measurement results obtained on the remainder of the sweeps in another curve. The first set of results is stored in the DESTINATION CURVE. The curve in which the results for the remainder are stored is the ALTERNATE CURVE.

   The Alternate Curve capability facilitates collecting two data sets in one experiment. For example, in cyclic voltammetry, the two sweeps of most interest are the first sweep and the Nth (or steady state sweep). The Model 273A can be used to collect data for the first sweep in the Destination Curve, then collect the steady state data in the Alternate Curve.

5. PROCESSING CURVE: A curve on which postacquisition processing (see descriptions of Processing Commands, ADD, SUB, EX, MIN, IMIN, MAX, IMAX, INT, IINT, and ILOG) is performed is the PROCESSING CURVE. A given curve can be a PROCESSING CURVE as well as one of the other previously described kinds of curves. For example, you could designate Curve 1 as both the DESTINATION CURVE and as the PROCESSING CURVE.

As discussed in Chapter 2, commands are provided that allow the operator to designate curve functions. In using these commands, it is essential that the constraints on the number of allowable curves, and on the available curve-designation numbers be observed.

Using these curve names, the Model 273A memory can be used with great efficiency to perform a wide variety of experiments. Since data acquisition is the desired end of producing a waveform with this system, you will note the commands FP (First Point) and LP (Last Point) used with every experiment in this appendix. FP and LP instruct the Model 273A on the number of data points to collect. FP and LP define point numbers in the Destination Curve memory, not direct memory addresses. If DCV is 0, then FP 0 means to start storing data in point 0 of Curve 0. If DCV is 2, FP 0 means to start storing data in point 0 of Curve 2, which is really point 2048 in the Model 273A memory. FP is usually set to 0. The same relative ad-dressing applies to LP. For all examples given here, DCV will always be 0 for consistency.

## E.3. DAC Descriptions

One DAC is called the Bias DAC. The Bias DAC has a range of ±8000 mV in controlled potential mode or two times the full-scale setting of the I/E converter in controlled current mode.

The second DAC is the Modulation DAC. This DAC can be used to apply waveforms of virtually any shape to the cell. The Modulation DAC can be used to apply complex waveforms far beyond the capabilities of an analog waveform programmer. Since the Modulation DAC is the heart of waveform generation using the Model 273A Potentiostat/Galvanostat, it will be discussed in detail.

The Modulation DAC, unlike the Bias DAC, has three full scale ranges selected by the MR command (Modulation Range). Each range can cover 8000 counts, which means that the resolution (or precision) of the DAC can be varied according to the needs of the experiment. The ranges and resolution for each scale are shown below:

| CMD | FS (mV) | CNTL E CNTS/mV | RESOLUTION (mV) |
|---|---|---|---|
| MR 0 | ±20 | 400 | 0.0025 |
| MR 1 | ±200 | 40 | 0.025 |
| MR 2 | ±2000 | 4 | 0.25 |

The available resolution of the Modulation DAC is dependent on the full scale setting for the DAC. In all cases, however, the percent precision for any range is 0.012% of full scale.

In the Control I mode, the Modulation DAC has the same range as the Bias DAC, 2 times full scale of the I/E converter. The range and resolution as a function of the MR command are as shown below.

| CMD | RANGE (% F.S.) | RESOLUTION (% f.s.) |
|---|---|---|
| MR 0 | 2 % f.s. | 0.00025 |
| MR 1 | 20 % f.s. | 0.0025 |
| MR 2 | 200 % f.s. | 0.025 |

The Modulation DAC can be programmed by the CV command, the Ramp Program Commands (INITIAL and VERTEX), or point-by-point in a mode called the Arbitrary Waveform Mode. The method selected depends on the complexity of the desired waveform and/or on the degree of control over the experiment desired by the chemist. The CV command is straightforward as well as limited in applications and so will not be covered here.

## E.4. Modulation Modes

There are three modulation modes available in the Model 273A. The desired mode is selected by the Modulation Mode Command, MM. The types of modulation are defined as:

| | |
|---|---|
| MM 0 | No Modulation |
| MM 1 | Ramp Program Modulation |
| MM 2 | Arbitrary Waveform Modulation |

The Modulation Mode determines the source of modulation information for the Model 273A. Modulation Mode 0 has no modulation at all; the Modulation DAC is not updated during the experiment (it is not automatically zeroed in this mode, simply not changed. The command MOD 0 can be used to zero the Modulation DAC). Modulation Mode 1 uses the Ramp Program commands to generate the Modulation DAC values during the experiment. Modulation Mode 2 requires that the user specify the modulation waveform point-by-point and store the waveform in the Model 273A memory before the experiment begins.

## E.5. Experiment Timing

The primary topic of this application note is to discuss methods of generating waveforms with the Model 273A. However, there is some overlap between the data-acquisition commands and the waveform-programming commands. This overlap deals with the timing of the experiment.

Experiment timing can be referred to as "scan rate" or "step time" when dealing with waveform generation or "data acquisition rate" when discussing data acquisition. It's obvious that the two functions must be related. Otherwise, synchronization between the modulation waveform and the data would be difficult to determine.

The commands to control this timing are TMB (Timebase) and S/P (Samples per Point). In most cases, the time to acquire one data point or to update the Modulation DAC according to the programmed waveform is equal to the product of the two parameters:

Rate = TMB (microseconds) * S/P

## E.6. Waveform Constraints

The design of the Model 273A requires that the Modulation DAC be updated at least once for each data point collected and stored by the instrument. This is an important consideration in designing waveforms on the Model 273A, since it is not possible to program in varying "dead times" between data points. The time between data points is the timebase multiplied by the samples/point.

If the timebase is less than 200 µs, the Model 273A uses a special high speed data-acquisition mode, in which most of the instrument's special functions are bypassed. The only waveform mode available is the Arbitrary Waveform Mode. Samples/point (S/P), point averaging (PAM), overload checking (OVER), and experiment monitoring functions are ignored.

In the Arbitrary Waveform Mode the waveform is usually (but not always) sent point-by-point from the host computer to the Model 273A, then stored in the Source Curve memory (defined by the SCV command). Point- by-point definition provides maximum flexibility in waveform shape. In this mode, the Model 273A can acquire data as rapidly as 100 µs/point (10000 points/second). The penalty is memory; the Source Curve does occupy a portion of the data memory. If the waveform is to only be applied once, the data can overlay the waveform, allowing use of the full 6144 point memory (one parameter stored; see E.2, MODEL 273A MEMORY USAGE).

**Note:** This overlay technique cannot be used if doing Sweep Averaging (SAM 1 or 2). SCV and DCV must be separate curves, limiting the maximum curve length to 3072 points. Also, note that linear sweep averaging is automatically selected if the timebase (TMB operand) is less than 200 µs/point.

In the Ramp Program Mode the waveform is calculated "on the fly" by the Model 273A during the experiment. The waveform does not occupy any of the 6144 data point memory of the instrument during the experiment. The penalty for this feature is time.

An important command in the Model 273A is INTRP (Interpolate Modulation). This command can have two values, "1" for on and "0" for off. When on, the INTRP command instructs the Model 273A to update the Modulation DAC once per sample, rather than once per point. If samples/point (S/P) is greater than one, the Model 273A will automatically calculate the smallest step size that it can apply to the cell while maintaining the experiment timing. This feature aids in approximating an analog waveform by applying small steps. There will be occasions where this function is not desired. For example, if a discreet step is required, such as in square-wave voltammetry, the INTRP function must be turned off.

## E.7. No Modulation Mode - "MM 0"

Some experiments require no modulation. Chronoamperometry is often performed by applying a constant potential to the working electrode, then monitoring current as a function of time. Chronopotentiometry involves the application of a constant current, monitoring the potential as a function of time. A third type of experiment requiring no modulation is one in which the open-circuit potential of the cell is monitored as a function of time.

In these experiments, Modulation Mode 0 is used. Any applied potential or current is set using one of the following commands:

| | |
|---|---|
| BIAS | Set Bias DAC (Control I or E Mode) |
| SETE | Set Cell Potential (Control E Mode) |
| SETI | Set Cell Current (Control I Mode) |

## E.8. Ramp Program Mode - "MM 1"

The Ramp Program Mode is selected by the command MM 1 (Modulation Mode 1). The Model 273A is instructed to generate a waveform defined by the vertices of the waveform. Up to 50 vertices can be defined, allowing a great degree of flexibility. This flexibility is the result of using a digital waveform programmer.

The use of the Ramp Program Mode requires that the vertices of the waveform be defined in terms of data-point numbers and Modulation DAC values. DAC values are used in place of real units, such as mV, since the modulation may be either current or potential, thus providing the greatest flexibility. The INITIAL command is always used to define the first vertex. The INITIAL command instructs the Model 273A that a new waveform program is starting, plus it defines the first data point to be used during the acquisition. Subsequent vertices are defined by a VERTEX command. Each vertex of the waveform must be defined in the same sequence as their appearance in the experiment.

For example, suppose we wished to perform a linear sweep voltammetry experiment starting at 0 V and ending at +1.000 V (all potentials will be versus the reference electrode). Since the potential range is 1000 mV, we must use the ±2000 mV range (MR 2) of the Modulation DAC. There are two vertices in the waveform, one being the Initial Potential and the other the Final Potential. Since the potentials are 0 and 1000 mV, the Modulation DAC values at the vertices are 0 and 4000, respectively (4 counts/mV * 1000 mV).

The next parameter to decide is the resolution of the data points. If we wish to collect one data point/mV, we would collect 1000 data points. Thus the commands to define our waveform are:

FP 0 (First Data Point Number)
LP 999 (Last Data Point Number)
INITIAL 0 0 (start at point zero, MOD DAC 0 counts)
VERTEX 999 4000 (ramp to point 999, MOD DAC 4000 counts)

Note that we have only defined the shape of the waveform, not the scan rate. If we wished to perform this experiment at a scan rate of 100 mV/second, the following commands must be added:

TMB 10000 (Timebase 10000 µs)
S/P 1 (Collect 1 Sample/Point)

Note that any combination of TMB and S/P equal to 10000 would result in the same scan rate. Figure E-2 depicts the ramp waveform in this example, indicating the commands to define each vertex. If we change the data density to collect one data point for each 2 mV of the scan, the second command would become VERTEX 499 4000 and we would change the Last Point command to LP 499. To maintain the same scan rate, we would have to slow the timebase by a factor of two, to 20000. Otherwise, collecting half the data points would require only half the time, doubling the scan rate.

| | |
|---|---|
| FP 0 | (First Data Point Number) |
| LP 999 | (Last Data Point Number) |
| INITIAL 0 0 | (start at point zero, MOD DAC 0 counts) |
| VERTEX 999 4000 | (ramp to point 999, MOD DAC 4000 counts) |
| TMB 10000 | (Timebase 10000 µs) |
| S/P 1 | (Collect 1 Sample/Point) |

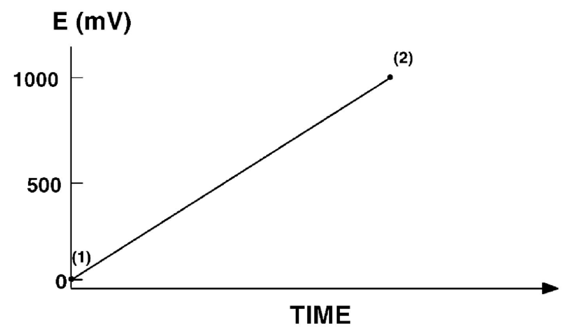


**Fig. E-2. Ramp Waveform For Linear Sweep Experiment**

Consider another experiment, one in which we wish to apply the waveform shown in Figure E-3. The Initial

Potential is -0.2 V, ramping to +1.2 V, then ramping to -0.6 V, and finally ramping to 0.0 V. There are four vertices in this waveform as shown in Figure E-3. We wish to scan at 20 mV/second. In addition, we wish to maximize data resolution, i.e., we want to collect as many data points as possible. The absolute maximum is 6144 in the Model 273A memory.

Since the greatest potential range is 1800 mV (-0.6 to +1.2 V), we must use the 2000 mV range on the Modulation DAC (MR 2). If we number our vertices one through four, the Modulation DAC will have the following values at each vertex:

| VERTEX | POTENTIAL | MODULATION DAC |
|---|---|---|
| 1 | -200 mV | -800 |
| 2 | +1200 mV | +4800 |
| 3 | -600 mV | -2400 |
| 4 | 0 mV | 0 |

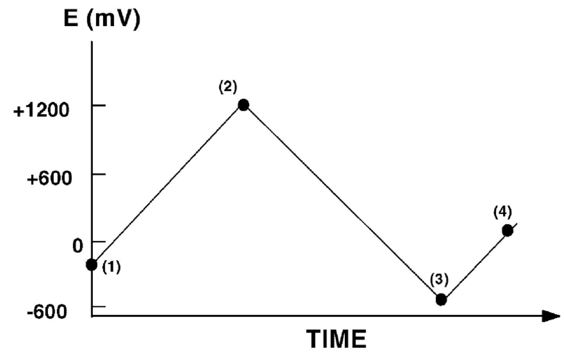


**Figure E-3. Waveform For Ramp Program Experiment**

The next step is to calculate the resolution we can achieve for the experiment. This is done by adding up the total mV scanned by each ramp of the waveform:

| | FROM | TO | RANGE |
|---|---|---|---|
| | -200 mV | +1200 mV | 1400 mV |
| plus | +1200 mV | -600 mV | 1800 mV |
| plus | -600 mV | 0 mV | 600 mV |
| TOTAL RANGE | | | 3800 mV |

The best resolution we can achieve is 1 mV/point, for a total of 3800 data points. Since 10 mV/second is equal to 1 mV every 100 ms, the TMB * S/P must be equal to 100000. The maximum value for TMB is 50000, so one acceptable pair of values would be TMB 50000 and S/P 2. The complete series of commands to define the waveform are:

```
FP 0
LP 3799
TMB 50000
S/P 2
INITIAL 0 -800
VERTEX 1399 4800
VERTEX 3199 -2400
VERTEX 3799  0
```

Though the INITIAL and VERTEX commands are called Ramp Program Mode Commands, they can be used to define the waveform of a step pulse experiment. For example, the step pulse experiment shown in Figure E-4 can easily be performed using the Model 273A in the Ramp Program Mode. The Initial Potential is -0.2 V, which is then stepped to -0.5 V, and then to -1.0 V.

As shown in Figure E-4, there are six vertices in this experiment. The time at each potential can be varied simply by varying the number of data points collected at each potential. There should always be only one point separating the vertices of the potential step, i.e., vertex pairs 2 and 3, 4 and 5. In our example, we will spend an equal period of time, collecting 1000 data points, at each potential, collecting one data point every 500 µs.  The command set to perform this experiment is:

FP 0
LP 2999
TMB 500
S/P 1
INITIAL 0 -800
VERTEX 999 -800
VERTEX 1000 -2000
VERTEX 1999 -2000
VERTEX 2000 -4000
VERTEX 2999 -4000

Note that since we can define up to 50 vertices and that each potential step requires two, we can define a pulse experiment with up to 24 potential steps (don't forget one vertex at the beginning and one at the end).

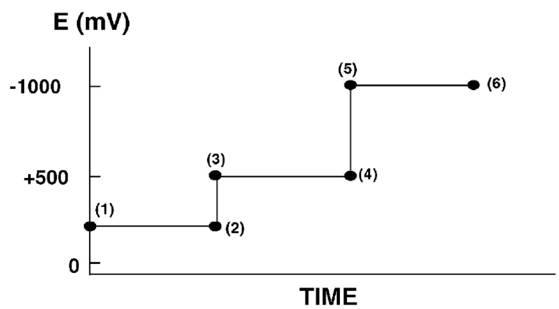


**Fig. E-4. Step Pulse Experiment With Ramp Program Mode**

Staircase ramps and pulses can be combined in any combination desired, provided that the maximum of 50 vertices is not exceeded.

Though the greatest range of the Modulation DAC is ±2000 mV, we can perform a linear sweep or cyclic experiment over a 4 volt range by using the Bias DAC to offset the Initial Potential. For example, if we wish to perform a cyclic voltammogram from 0 V to +4.0 V then back to 0 V, we could use the command set below:

FP 0
LP 3999
BIAS 2000
INITIAL 0 -8000
VERTEX 1999 8000
VERTEX 3999 -8000

Note that the values of the Bias DAC and the Modulation DAC are additive. At the Initial Potential we are applying +2000 mV from the Bias DAC and -2000 mV from the Modulation DAC for a net applied potential of zero.

## E.9. Arbitrary Waveform Mode - "MM 2"

The Arbitrary Waveform mode is selected by the command MM 2 (Modulation Mode 2). This mode gains it name from the fact that the user defines the modulation waveform point-by-point in the host computer, then sends the waveform to the Model 273A, which will then apply the waveform to the cell. Since the waveform is defined point-by-point, it may have any shape desired as long as the DAC values defined by the waveform are valid, i.e., ± 8000 DAC counts. Please note that there are often several ways to define a particular waveform, some using the Ramp commands, some using the Load Curve command, and some using both. Complex waveforms can often be divided into components for easier programming.

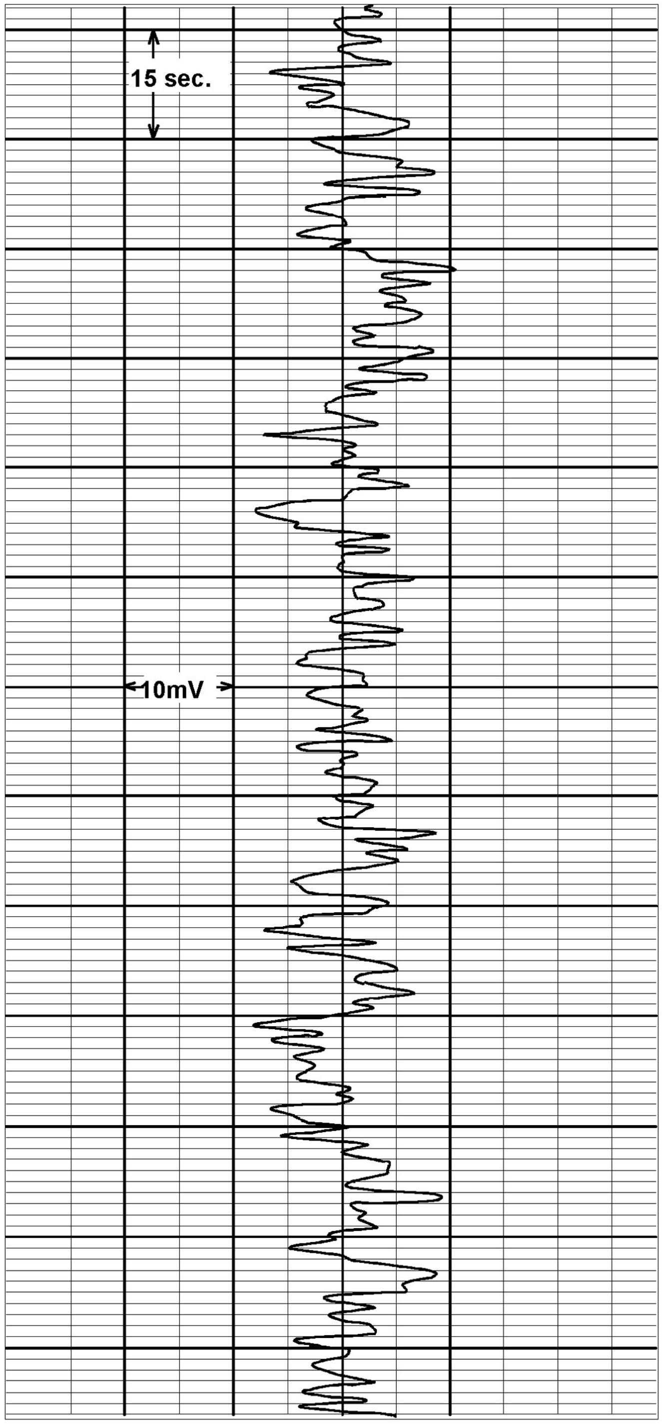


**Fig. E-5. An FFT Waveform for a Base Frequency of 50mHz and Maximum Amplitude of 10mV**

The Arbitrary Waveform mode must be used in all cases where the waveform is too complex to define using Ramp commands or in cases where the Ramp functions slow the experiment execution speed to unacceptable rates.

A major use of the Arbitrary Waveform mode is for experiments which require a data acquisition rate faster than that possible with the Ramp Modulation mode. (See the description of the TMB command in Chapter 2 for a discussion of the various factors that influence the maximum sampling speed.) For experiments where the waveform can be defined by the Ramp commands but require the Arbitrary mode for speed, a command called ASM is available. ASM instructs the Model 273A to assemble the waveform defined by the Ramp commands point-by-point in the Source Curve (defined by the command SCV) memory. ASM allows the facility of Ramp commands to be used in the Arbitrary waveform mode.

Many electrochemical experiments require complex waveforms that cannot be defined using the Ramp Program Commands in spite of their flexibility. For example, in Princeton Applied Research's Model 378/388 AC Impedance System, ac frequencies from 0.0001 Hz to 10 Hz are generated using an FFT method in which 20 frequencies are combined into a single, complex waveform by the computer (Figure E-5), then applied to the cell by the Model 273A. The Arbitrary Waveform Mode must be used to apply this waveform. For additional information, request Application Note AC-1, Basics of AC Impedance Measurements.

Any waveform that can be produced using the Ramp Program commands can also be generated using the Arbitrary Waveform Mode. One of our examples for Ramp Program Mode was a step-pulse experiment. Let's start with a simple step-pulse experiment.

In this experiment, we wish to start at -0.1 V, then step to -0.9 V. The purpose of the experiment is to record the transient current in a short period of time after the application of the pulse. The Arbitrary Waveform Mode allows us to sample the current as rapidly as 50 µs per point to collect data during the transient period. In our example, we will collect a total of 2000 data points, with 5 points at -0.1 V and the balance at -0.9 V.

We can use the fact that the Model 273A has two DAC's to perform this experiment simply. The command set for this experiment is:

| | |
|---|---|
| FP 0 | First Point is 0 |
| LP 1999 | Last Point is 1999 |
| MM 2 | Arbitrary Waveform Mode |
| MR 2 | Modulation Range ±2000 mV |
| TMB 100 | 100 µs/point |
| BIAS -900 | Set Bias DAC to -900 mV |
| DCV 0 | Destination Curve 0 |
| SCV 2 | Source Curve 2 |
| PCV 2 | Processing Curve 2 |
| CLR | Clear Points in PCV |
| LC 0 5 3200 3200 3200 3200 3200 | |
| NC | New Curve |
| CELL 1 | Cell On |
| TC | Take Curve |

This experiment shows a technique for simplifying waveform programming in the Arbitrary Waveform Mode. Instead of defining the Modulation DAC value at each point for 2000 points, we used the CLR command to set each point in the PCV (identically the SCV) to zero. We then defined the first five points of the SCV to be +800 mV (3200 DAC counts) and set the Bias DAC to -900 mV. Since the two DAC's are additive, the applied potential for the first five points will be -900 + 800 = -100 mV and the applied potential for the last 1995 points will be -900 + 0 = -900 mV.

## E.10. Square-Wave Voltammetry Using the Arbitrary Waveform Mode

Square-wave voltammetry is an example of an experiment that requires a waveform not definable by the Ramp commands. This waveform is the sum of a staircase ramp and a symmetrical square wave, as shown in Figure E-6. For more information on the theory of square-wave voltammetry, request Application Note S-7, Basics of Square-Wave Voltammetry.

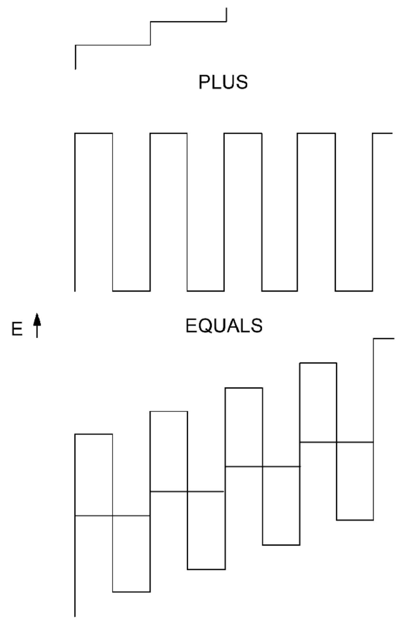


**Fig. E-6. Waveform Used For Square-Wave Volammetry.**

The scan rate for square-wave voltammetry is defined as the frequency (units inverse seconds) multiplied by the potential step in mV. Two data points are required per square-wave cycle, one during the forward pulse and one during the reverse pulse. Since the time for a single data point is equal to the Timebase multiplied by the Samples per Point (TMB * S/P), the frequency for square-wave voltammetry can be related to the TMB and S/P by the equation:

$$\text{Frequency (Hz)} = 1{,}000{,}000/(\text{TMB} * \text{S/P} * 2).$$

If S/P is 1, the TMB can be used to define frequencies between 5000 Hz at TMB = 100 (the theoretical maximum for the Model 273A, though in reality it is considerably lower) and 10 Hz at TMB = 50000. Lower frequencies can be achieved by increasing S/P.

There are at least two methods that can be used to define a square-wave voltammetry waveform. In the first method, the host computer defines the entire waveform point-by-point, then uses the LC (Load Curve) command to transfer the waveform to the Model 273A. Let's set up a square-wave voltammetry experiment using this technique.

1. DEFINE EXPERIMENT: We wish to perform square-wave voltammetry on a solution containing lead and cadmium. The Initial Potential will be -0.2 V and the Final Potential will be -0.8 V. We will collect a data point every 2 mV, i.e., $E_{STEP}$ = 2 mV. Since a square-wave voltammogram requires two measurements per step, one on the forward pulse and one on the reverse pulse, we must use 1 mV/point. For our example, the pulse height, $E_{SW}$ will be 10 mV, corresponding to a 20 mV peak-to-peak square wave.

2. CALCULATE WAVEFORM: We will use the Bias DAC to set the Initial Potential with the BIAS -200 command, so the first point of our waveform will be 0. The second point will be the sum of our potential step of -2 mV (-8 Modulation DAC counts) and the forward pulse, -10 mV (-40

Modulation DAC counts). The third point will have the same potential step, but with the reverse pulse added (+18 mV or +40 Modulation DAC counts). At the fourth point, we will apply a staircase step with the forward pulse added, and so on. Figure E-7 shows a BASIC program to calculate the square wave waveform.

3. SET UP MODEL 273A FOR EXPERIMENT:

COMMANDS TO SEND

```
MM 0
DCV 0
SCV 1
PCV 1
FP 0
LP 600
BIAS -200
LC 0 601
```

```
10 ! Square Wave Calculation Routine
20 ! Written for HP85 Computer
30 ! **Entry Conditions:
40 !      I = Initial Mod DAC Value (0)
50 !      S = Potential Step in DAC counts
(8)
60 !      P = Pulse Height in DAC counts
(40)
70 !      N = Number of Points (600)
80 !
90 DIM A(N)
100 A(0) = I + P
110 A(1) = I - P
120 A(2) = I + S + P
130 FOR X = 3 TO N
140 A(X) = A(X-2) + S
150 NEXT X
```

**Fig. E-7. Calculate Square Wave in Computer.**

4. START THE EXPERIMENT: The experiment is now set up and ready to run. The commands to start the experiment are:

```
CELL 1
NC
TC
```

A complete program to set up, execute, and plot the data from square-wave voltammetry experiments is provided at the end of this appendix. The program was written for an HP85 computer in BASIC, but may be translated to another computer by the user, if desired.

The second method is an excellent example of a useful programming technique for the Model 273A. In this technique, the Ramp commands are used to produce one part of the waveform and the host computer constructs the rest. Referring to Figure E-7, we see that a square-wave voltammetry waveform is the sum of a square wave and a staircase. The Model 273A can be used to create a staircase to which we simply add a square wave.

We will use the same cadmium and lead solution from the first square-wave voltammetry example. The Initial Potential will be - 0.2 V and the Final Potential will be -0.8 V. We will collect a data point every 2 mV, i.e., $E_{STEP}$ = 2 mV. Since a square-wave voltammogram requires two measurements per step, one on the forward pulse and one on the reverse pulse, we must use 1 mV/point (we really want 2 mV steps every two points, but we can't program it directly; we have to use trickery). In addition, we need one point at the initial potential, which we obtain with an extra VERTEX command. The staircase ramp will thus be defined as:

| | | |
|---|---|---|
| INITIAL | 0 | -800 |
| VERTEX | 1 | -800 |
| VERTEX | 600 | -3200 |

If we define the Source Curve memory as 2 with the SCV 2 command, we can instruct the Model 273A to assemble the waveform defined by the Ramp commands into the Source Curve by sending the command ASM.

Next, we program the host computer to generate a square wave. If we want $E_{SW}$ to be 10 mV (which corresponds to a 20 mV pk-pk square wave) the computer would fill an array with alternating -40's and +44's, totaling 601 elements, with the first element equal to 0 to coincide with our first point in the staircase.

0 -40 44 -40 44 -40 44 -40 44...

This may seem strange, since square-wave voltammetry usually is performed with the forward pulse in the same direction as the staircase step, plus the fact that one step is -10 mV while the other is +11 mV.

The apparent polarity reversal of the square wave can be explained by the commands available in the Model 273A. You may have guessed by now that we will add two waveforms in the Model 273A to form our final square wave signal. The Model 273A does not have a command to add two curves, but does have a command to subtract one curve from another. This command will be used. By starting with reversed polarity on the square wave generated by the computer and then subtracting the square wave curve from the staircase curve, we will obtain the correct polarity in the final waveform.

The reason for using different amplitudes for the forward and reverse pulses is to compensate for the form of the staircase generated by the Ramp commands. As shown in Figure E-8, the staircase generated is not exactly the one required. In staircase voltammetry, the time for each potential step is equal to one square-wave cycle. The staircase obtained with the Ramp commands causes a step of 1 mV halfway through the square-wave cycle. By making the original square wave reverse pulse 1 mV larger in amplitude, the resultant waveform is corrected.

The Model 273A commands needed to load the square wave into memory 0 are:

```
FP 0
LP 600
PCV 0
LC 0 601
```

The LC, or Load Curve, command always loads into the Processing Curve, specified by the PCV command. More information on using the Load Curve command is provided further on.

Let's summarize the steps in generating the square wave voltammetry waveform for our example:

1. CREATE STAIRCASE
   FP 0
   LP 600
   INITIAL 0 -800
   VERTEX 1 -800

```
VERTEX 600 -3200
SCV 2
ASM
```

The staircase is now in memory 2 of the Model 273A.

2. CREATE SQUARE WAVE IN COMPUTER

```
10 DIM A%(600)
20 A%(0) = 0
30 FOR X = 1 TO 600
40 IF X = INT(X/2) * 2 THEN A%(X) = 44
50 IF X < > INT(X/2) * 2 THEN A%(X) = -40
60 NEXT X
```

3. SEND SQUARE WAVE TO THE MODEL 273A MEMORY 0

```
PCV 0
FP 0
LP 600
LC 0 601 ...
```

4. PRODUCE COMPOSITE WAVEFORM

```
SUB 0 2
```

## E.11. Asymmetrical Waveforms

Asymmetric waveforms all require use of the Arbitrary Waveform mode in the Model 273A. The Model 273A is not as useful for experiments involving asymmetric waveforms, such as normal pulse or differential pulse voltammetry, as it is for symmetric waveforms. This is due to the timing of the experiment. The timebase of the experiment cannot be smaller than the time of the shortest event of the waveform. In the case of differential pulse voltammetry, the shortest timed event is the duration of the pulse. For such an experiment, if the pulse duration is one tenth the step time for the staircase, ten points must be defined for every step. As shown in Figure E-9, this results in the Model 273A collecting eight data points that are not required for the experiment, i.e., eighty percent of the Model 273A's memory is necessarily wasted to maintain the timing. Because of the asymmetry and the method used by the Model 273A to update the Modulation DAC, waveforms of this type must be generated point-by-point in the host computer, then transferred to the Model 273A for execution.

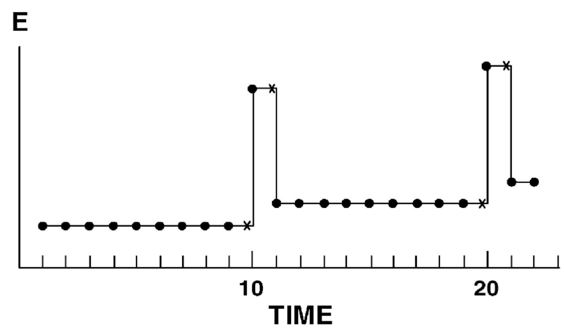


**Fig. E-9. Waveform Timing for Differential Pulse Polarography**

Let's set up a differential-pulse voltammetry experiment. We have a solution containing cadmium and lead again. We wish to perform a differential pulse voltammetry experiment from -0.2 V to -0.8 V with a step time of 0.5 seconds, a step potential of 2 mV, a pulse time of 0.05 seconds, and a pulse height of 50 mV.

The shortest event in this experiment is the pulse duration, which is 50 msec. Since the pulse duration is one tenth of the step time, we must collect 10 points per step, nine at the step potential and one during the pulse. There are 300 steps in the experiment, so we have to collect a total of 3000 data points. A computer routine to calculate the waveform for this experiment is shown in Figure E-10.

```
10 ! Routine to Calculate Waveform for
20 ! Differential Pulse Voltammetry
Experiment.
30 !   Entry Conditions:
40 !        I = Initial Mod DAC counts
50 !        S = Step Size in DAC counts
60 !        P = Pulse Height in DAC counts
70 !        N = Total Points
80 !
90 DIM C(N)
100 FOR X = 0 TO N STEP 10
110 FOR Y = 1 TO 9
120 C(X+Y) = I
130 NEXT Y
140 C(X+10) = I + P
150 I = I + S
```

**Fig. E-10. Routine to Calculate Differential Pulse Waveform.**

The following commands can be used to set up the experiment:

```
FP 0
LP 2999
PCV 0
DCV 0
SCV 0
BIAS -200
MM 2
MR 2
INTRP 0
TMB 10000
S/P 10
NC
LC 0 3000 ...
CELL 1
TC
```

Note that the New Curve command (NC) is executed before the Source Curve is loaded with the waveform. This is important because one function of the NC command is to clear all points in DCV. Since DCV and SCV are the same in this experiment, the NC command would erase the waveform if sent after the execution of the LC command.

## E.12. Load Curve Command

The Load Curve command (LC) is used to load data from the computer into the Model 273A memory. The data will be loaded into the curve memory defined by the PCV (Processing Curve) command, not the SCV (Source Curve). Normally, the PCV is set to be equal to the SCV before the LC command is executed to ensure that the data is loaded into the correct area of memory.

The Load Curve (LC) command requires special handling in a program. It cannot be used to transfer a large number of data points from the computer to the Model 273A using the GPIB routines found in the Simple Static Interface Routines. GPIB protocol causes each number sent on the bus to be translated into a string of ASCII characters. For example, the number 1234 would be translated into a string of five characters, the first character for the sign followed by the ASCII representation of 1, 2, 3, and 4. This string would then be followed by one more character, called a delimiter, to separate it from the next number in a series, or a terminator if it is the last number sent. This translation is performed automatically by the GPIB driver of the computer. The Model 273A input buffer is limited to a maximum of 80 ASCII characters. It's obvious that a few numbers can occupy the full input buffer. If the input buffer is overfilled, an error results.

The following subroutine was written for an HP85 microcomputer to illustrate the method of sending a Source Curve from the computer to the Model 273A. This routine can be adapted to any computer by changing the GPIB command lines.

```
10    ! Routine to Send Source Curve from the
20    ! Computer to the Model 273A.
30    !   Entry Conditions:
40    !       N = Number of Points
50    !       A9= Model 273A Address + 700
60    !
70    !   IEEE Routines are assumed to be:
80    !      Send and Receive  - Line 5000
90    !      Send Only         - Line 5100
100   !
110   C$ = "LC 0 "&VAL$(N)  ! LC Command
120   GOSUB 5100 ! Send Only
130   FOR X = 1 TO N
140   OUPUT A9 USING "K";A(X) ! Send a
      number
150   NEXT X
```

**Fig. E-11. Load Curve Routine**

## E.13. Conclusion

The Model 273A has very flexible waveform programming capabilities. As the waveform becomes more complex, the need for thorough understanding on the part of the programmer becomes more and more important. There may be several ways to achieve a selected goal. Practicing various waveform programming techniques can result in the ability to quickly and easily program the Model 273A for new experiments.

**SQUARE WAVE VOLTAMMETRY PROGRAM FOR HP85**

```
10 ! Square Wave Voltammetry
20 !
30 ! HP85 Version 1.0
40 ! 28 August 1983
50 ! L. E. Fosdick
60 ! PARC
70 !
80 !      File: SWV
90 !
100 ! ** Initialize
110 CLEAR
120 DIM C$[80] ! Command Line
130 SHORT A(2048),A1(1024)
140 DISP "Square Wave Voltammetry"
150 DISP ""
160 DISP "NOTE: To simplify program, the only error check is # points"
```

```
170 DISP ""
180 DISP "Use LF Terminator"
190 DISP ""
200 DISP "Model 273A Address";
210 INPUT A9
220 A9=A9+700
230 C$="DD 10" ! LF Delimiter
240 GOSUB 5000
290 !
300 !
1000 ! ** Experiment Setup
1010 CLEAR
1020 DISP "Enter Parameters"
1030 DISP ""
1040 DISP "Initial E (+/-1900 mV)";@ INPUT I
1050 DISP "Final E (+/-1900 mV)";@ INPUT F
1060 DISP "Scan Increment (1 - 10 mV)";@ INPUT S
1070 DISP "Pulse Height (1 - 100 mV)";@ INPUT P
1080 DISP "Frequency (1 - 1000 Hz)";@ INPUT T
1090 DISP "Current Range Exponent";@ INPUT R
1100 DISP "Cycles";@ INPUT C
1110 DISP "Filter ON (Y/N)";@ INPUT Q$
1120 IF Q$="Y" THEN F9=1 ELSE F9=0
1130 IF Q$< >"N" THEN 1110
1140 DISP "Run Experiment (Y/N)";@ INPUT Q$
1150 IF Q$="Y" THEN 2000
1160 IF Q$< >"N" THEN 1140
1170 DISP "End Program" @ BEEP
1180 END
1190 !
2000 !
2010 ! ** Run Experiment
2020 CLEAR
2030 DISP "Computing Parameters"
2040 DISP ""
2050 ! Number of Points
2060 N=ABS(F-I)*2 DIV S+2
2070 IF N>2047 THEN BEEP @ DISP "ERROR: >2048 Points" @ GOTO 1170
2100 ! Pulse Height DAC Counts
2110 P=ABS(P)
2120 IF F<I THEN P=-P
2130 P1=P*4
2140 ! Scan Inc DAC Counts
2150 S=ABS(S)
2160 IF F<I THEN S=-S
2170 S1=S*4
2180 ! Timebase (TMB) and
2190 ! Samples/Point (S/P)
2200 ! Freq = 1/(TMB * S/P * 2)
2210 ! High Frequency Case
2220 IF T>500 THEN T2=1 @ T1=10^6 DIV (2*T) @ GOTO 2300
2230 T2=1
2240 T1=10^6/2/T/T2 @ IF T1>10000 THEN T2=T2+1 @ GOTO 2240
2300 ! Calculate Waveform Pts
2310 A(0)=I*4 ! Initial E
2320 A(1)=A(0)+P1
2330 A(2)=A(0)-P1
2340 FOR X=3 TO N
```

```
2350 A(X)=A(X-2)+S1
2360 NEXT X
2370 !
3000 !
3010 ! Run Exp & Get Data
3020 DISP "Sending Commands to M273A"
3030 DISP ""
3040 ! Configure MOD DAC
3050 C$="DCL;MM 2; MR 2;INTRP 0; MOD "&VAL$(A(0))
3060 GOSUB 5000 ! Send Command
3070 ! Configure Memory Usage
3080 C$="DCV 0;SCV 2;PCV 2;FP 0;LP "&VAL$(N)
3090 GOSUB 5000
3100 ! Send Waveform
3110 GOSUB 5500
3120 ! Configure Acquisition
3130 C$="PAM 0;SAM 1;DT 2000;SWPS
        "&VAL$(C)&";TMB "&VAL$(T1)
3140 GOSUB 5000
3150 C$="S/P "&VAL$(T2)&";I/E "&VAL$(R)
3160 GOSUB 5000
3200 !
3210 ! Start Experiment
3220 DISP "Starting Experiment"
3230 DISP ""
3240 C$="NC;CELL 1"
3250 GOSUB 5000
3260 WAIT 2000 ! Dead Time
3270 C$="TC;WCD;CELL 0"
3280 GOSUB 5000
3290 DISP "Experiment Complete"
3300 DISP "Getting Data"
3310 DISP ""
3320 C$="PCV 0;DC 1 "&VAL$(N)
3330 GOSUB 5000
3340 DISP "Converting Data"
3350 X1=0
3360 V1=-10^9
3370 V2=-V1
3380 FOR X=1 TO N STEP 2
3390 X1=X1+1
3400 A1(X1)=A(X)-A(X+1)
3410 V1=MAX(V1,MAX(A(X),MAX(A(X+1),A1(X1))))
3420 V2=MIN(V2,MIN(A(X),MIN(A(X+1),A1(X1))))
3430 NEXT X
4000 !
4010 !  ** Plot Data
4020 !
4030 ! Round X-axis limits
4040 IF I<0 THEN I1=(I-99) DIV 100*100 ELSE I1=(I+99) DIV 100*100
4050 IF F<0 THEN F1=(F-99) DIV 100*100 ELSE F1=(F+99) DIV 100*100
4060 ! Tick Mark Spacing
4070 E1=245 DIV (ABS(F1-T1) DIV 100)
4080 ! E Plot Increment
4090 E2=245/ABS((F1-I1)/S)
4100 ! I Plot Increment
4110 Y2=180/(V1-V2)
4120 ! I Zero Offset
```

```
4130 IF V2<0 THEN Y3=V2*Y2 ELSE Y3=0
4200 ! Forward & Reverse
4210 GOSUB 4800 ! Plot Axes
4220 X=10 ! X Offset
4230 FOR X9=1 TO N STEP 2
4240 X=X+E2 @ Y=A(X9)
4250 GOSUB 4900 ! Plot Point
4260 Y=A(X9+1)
4270 GOSUB 4900
4280 NEXT X9
4290 COPY ! Hardcopy
4300 ! Difference Plot
4310 GOSUB 4800 ! Plot Axes
4320 X=10
4330 N1=N/2 ! # Diff Points
4340 FOR X9=1 TO N1
4350 X=X+E2 @ Y=A1(X9)
4360 GOSUB 4900 ! Plot Point
4370 NEXT X9
4380 PRINT "" @ PRINT ""
4390 COPY ! Hardcopy
4400 PRINT "" @ PRINT 22
4410 PRINT "I max = ";V1+"x10^";R-3;"   A"
4420 PRINT "I min = ";V2;"x10^";R-3;"   A"
4430 PRINT "Frequency = ";T;" Hz"
4440 PRINT "Scan Increment = ";S;" mV"
4450 PRINT "Pulse Height = ";P;" mV"
4460 PRINT "Cycles = ";C
4470 IF F9=1 THEN PRINT "Filter ON" ELSE PRINT "Filter OFF"
4480 FOR X9=1 TO 5 @ PRINT "" @ NEXT X9
4490 BEEP ! End of Experiment
4500 GOTO 1000 ! Next Experiment
4510 !
4790 ! Plot Axes
4800 GCLEAR
4810 SCALE 0,255,0,192
4810 XAXIS 10,E1,10,255
4830 MOVE 3,0 @ LABEL VAL$(I1)
4840 Q-253-LEN(VAL$(F1))*7
4850 MOVE Q,0 @ LABEL VAL&(F1)
4860 MOVE 125, 0 @ LABEL "E (mV)"
4870 YAXIS 10,18,10,192
4880 MOVE 2,105 @ LABEL "I"
4890 RETURN
4900 ! Plot a Point
4910 Y=10+Y*Y2-Y3
4920 MOVE X,Y
4930 PLOT X,Y
4940 RETURN
5000 !
5010 ! ** IEEE Routines
5020 !
5030 !  Device Driver
5040 GOSUB 5100 ! Send
5050 GOSUB 5200 ! Receive
5060 RETURN
5070 !
5100 ! Send Commands
```

```
5110 GOSUB 5300 ! Serial Poll
5120 IF BIT(S9,0)=0 THEN 5110
5130 OUTPUT A9 USING "K" ; C$
5140 RETURN
5150 !
5200 ! Receive Responses
5210 I9=0 ! Counter
5220 GOSUB 5300 ! Serial Poll
5230 IF BIT(S9,7)=0 THEN 5250
5240 I9=I9+1 @ ENTER A9 ; A(I9)
5250 IF BIT(S9,0)=0 THEN 5220
5260 RETURN
5270 !
5300 ! Serial Poll
5310 S9=SPOLL(A9)
5320 RETURN
5330 !
5500 ! Load Curve
5510 !    N =# Points - 1 (LP)
5520 C$="LC 0 "&VAL$(N+1)
5530 GSUB 5100 ! Send
5540 FOR X=0 TO N
5550 OUTPUT A9 USING "K" ; A(X)
5560 NEXT X
5570 RETURN
```

# APPENDIX F. ASCII CHARACTER CODES

| BINARY | DECIMAL | HEX | CHARACTER | WHAT TO TYPE |
|---|---|---|---|---|
| 0000000 | 0 | 00 | NULL | CONTROL @ |
| 0000001 | 1 | 01 | SOH | CONTROL A |
| 0000010 | 2 | 02 | STX | CONTROL B |
| 0000011 | 3 | 03 | ETX | CONTROL C |
| 0000100 | 4 | 04 | ET | CONTROL D |
| 0000101 | 5 | 05 | ENQ | CONTROL E |
| 0000110 | 6 | 06 | ACK | CONTROL F |
| 0000111 | 7 | 07 | BEL | CONTROL G |
| 0001000 | 8 | 08 | BS | CONTROL H |
| 0001001 | 9 | 09 | HT | CONTROL I |
| 0001010 | 10 | 0A | LF | CONTROL J |
| 0001011 | 11 | 0B | VT | CONTROL K |
| 0001100 | 12 | 0C | FF | CONTROL L |
| 0001101 | 13 | 0D | CR | CONTROL M or RETURN |
| 0001110 | 14 | 0E | SO | CONTROL N |
| 0001111 | 15 | 0F | SI | CONTROL O |
| 0010000 | 16 | 10 | DLE | CONTROL P |
| 0010001 | 17 | 11 | DC1 | CONTROL Q |
| 0010010 | 18 | 12 | DC2 | CONTROL R |
| 0010011 | 19 | 13 | DC3 | CONTROL S |
| 0010100 | 20 | 14 | DC4 | CONTROL T |
| 0010101 | 21 | 15 | NAk | CONTROL U |
| 0010110 | 22 | 16 | SYN | CONTROL V |
| 0010111 | 23 | 17 | ETB | CONTROL W |
| 0011000 | 24 | 18 | CAN | CONTROL X |
| 0011001 | 25 | 19 | EM | CONTROL Y |
| 0011010 | 26 | 1A | SUB | CONTROL Z |
| 0011011 | 27 | 1B | ESCAPE | CONTROL [ or ESC |
| 0011100 | 28 | 1C | FS | CONTROL \ |
| 0011101 | 29 | 1D | GS | CONTROL ] |
| 0011110 | 30 | 1E | RS | CONTROL ^ |
| 0011111 | 31 | 1F | US | CONTROL _ |
| 0100000 | 32 | 20 | SPACE | SPACE |
| 0100001 | 33 | 21 | ! | ! |
| 0100010 | 34 | 22 | " | " |
| 0100011 | 35 | 23 | # | # |
| 0100100 | 36 | 24 | $ | $ |
| 0100101 | 37 | 25 | % | % |
| 0100110 | 38 | 26 | & | & |
| 0100111 | 39 | 27 | ' | ' |
| 0101000 | 40 | 28 | ( | ( |
| 0101001 | 41 | 29 | ) | ) |
| 0101010 | 42 | 2A | * | * |
| 0101011 | 43 | 2B | + | + |
| 0101100 | 44 | 2C | , | , |
| 0101101 | 45 | 2D | - | - |
| 0101110 | 46 | 2E | . | . |
| 0101111 | 47 | 2F | / | / |
| 0110000 | 48 | 30 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 0110001 | 49 | 31 | 1 | 1 |
| 0110010 | 50 | 32 | 2 | 2 |
| 0110011 | 51 | 33 | 3 | 3 |

| BINARY | DECIMAL | HEX | CHARACTER | WHAT TO TYPE |
|---|---|---|---|---|
| 0110100 | 52 | 34 | 4 | 4 |
| 0110101 | 53 | 35 | 5 | 5 |
| 0110110 | 54 | 36 | 6 | 6 |
| 0110111 | 55 | 37 | 7 | 7 |
| 0111000 | 56 | 38 | 8 | 8 |
| 0111001 | 57 | 39 | 9 | 9 |
| 0111010 | 58 | 3A | : | : |
| 0111011 | 59 | 3B | ; | ; |
| 0111100 | 60 | 3C | < | < |
| 0111101 | 61 | 3D | = | = |
| 0111110 | 62 | 3E | > | > |
| 0111111 | 63 | 3F | ? | ? |
| 1000000 | 64 | 40 | @ | @ |
| 1000001 | 65 | 41 | A | A |
| 1000010 | 66 | 42 | B | B |
| 1000011 | 67 | 43 | C | C |
| 1000100 | 68 | 44 | D | D |
| 1000101 | 69 | 45 | E | E |
| 1000110 | 70 | 46 | F | F |
| 1000111 | 71 | 47 | G | G |
| 1001000 | 72 | 48 | H | H |
| 1001001 | 73 | 49 | I | I |
| 1001010 | 74 | 4A | J | J |
| 1001011 | 75 | 4B | K | K |
| 1001100 | 76 | 4C | L | L |
| 1001101 | 77 | 4D | M | M |
| 1001110 | 78 | 4E | N | N |
| 1001111 | 79 | 4F | O | O |
| 1010000 | 80 | 50 | P | P |
| 1010001 | 81 | 51 | Q | Q |
| 1010010 | 82 | 52 | R | R |
| 1010011 | 83 | 53 | S | S |
| 1010100 | 84 | 54 | T | T |
| 1010101 | 85 | 55 | U | U |
| 1010110 | 86 | 56 | V | V |
| 1010111 | 87 | 57 | W | W |
| 1011000 | 88 | 58 | X | X |
| 1011001 | 89 | 59 | Y | Y |
| 1011010 | 90 | 5A | Z | Z |
| 1011011 | 91 | 5B | [ | [ |
| 1011100 | 92 | 5C | \ | \ |
| 1011101 | 93 | 5D | ] | ] |
| 1011110 | 94 | 5E | ^ | ^ |
| 1011111 | 95 | 5F | _ | _ |
| 1100000 | 96 | 60 | ` | ` |
| 1100001 | 97 | 61 | a | a |
| 1100010 | 98 | 62 | b | b |
| 1100011 | 99 | 63 | c | c |
| 1100100 | 100 | 64 | d | d |
| 1100101 | 101 | 65 | e | e |
| 1100110 | 102 | 66 | f | f |
| 1100111 | 103 | 67 | g | g |
| 1101000 | 104 | 68 | h | h |

| | | | | |
|---|---|---|---|---|
| 1101001 | 105 | 69 | i | i |
| 1101010 | 106 6A | | j | j |
| 1101011 | 107 6B | | k | k |

| BINARY | DECIMAL | HEX | CHARACTER | WHAT TO TYPE |
|---|---|---|---|---|
| 1101100 | 108 | 6C | l | l |
| 101101 | 109 | 6D | m | m |
| 1101110 | 110 | 6E | n | n |
| 1101111 | 111 | 6F | o | o |
| 1110000 | 112 | 70 | p | p |
| 1110001 | 113 | 71 | q | q |
| 1110010 | 114 | 72 | r | r |
| 1110011 | 115 | 73 | s | s |
| 1110100 | 116 | 74 | t | t |
| 1110101 | 117 | 75 | u | u |
| 1110110 | 118 | 76 | v | v |
| 1110111 | 119 | 77 | w | w |
| 1111000 | 120 | 78 | x | x |
| 1111001 | 121 | 79 | y | y |
| 1111010 | 122 | 7A | z | z |
| 1111011 | 123 | 7B | { | { |
| 1111100 | 124 | 7C | \| | \| |
| 1111101 | 125 | 7D | } | } |
| 1111110 | 126 | 7E | ~ | ~ |
| 1111111 | 127 | 7F | RUBOUT | |

## CONTROL CHARACTER ALTERNATE CODE NAMES

| CHARACTER | MEANING |
|---|---|
| NUL | NULL, CTRL SHIFT P, TAPE LEADER |
| SOH | START OF HEADER, SOM |
| STX | START OF TEXT, EOA |
| ETX | END OF TEXT, EOM |
| EOT | END OF TRANSMISSION, END |
| ENQ | ENQUIRY, WRU, WHO ARE YOU |
| ACK | ACKNOWLEDGE, RU, ARE YOU |
| BEL | BELL |
| BS | BACKSPACE, FE0 |
| HT | HORIZONTAL TAB, TAB |
| LF | LINE FEED, NEW LINE, NL |
| VT | VERTICAL TAB, VTAB |
| FF | FORM FEED, FORM, PAGE |
| CR | CARRIAGE RETURN, EOL |
| SO | SHIFT OUT, RED SHIFT |
| SI | SHIFT IN, BLACK SHIFT |
| DLE | DATA LINK ESCAPE, DC0 |
| DC1 | XON, READER ON |
| DC2 | TAPE, PUNCH ON |
| DC3 | XOFF, READER OFF |
| DC4 | TAPE, PUNCH OFF |
| NAK | NEGATIVE ACKNOWLEDGE, ERR |
| SYN | SYNCHRONOUS IDLE, SYNC |
| ETB | END OF TEXT BUFFER, LEM |
| CAN | CANCEL, CANCL |
| EM | END OF MEDIUM |
| SUB | SUBSTITUTE |
| ESC | ESCAPE, PREFIX |

| | |
|---|---|
| FS | FILE SEPARATOR |
| GS | GROUP SEPARATOR |
| RS | RECORD SEPARATOR |
| US | UNIT SEPARATOR |

# APPENDIX G. ABBREVIATED COMMAND DESCRIPTIONS: TOPICAL INDEX

## G.1. Introduction

The table beginning on page 114 uses the following definitions:

**COMMAND TYPE CODES**

A = ACTION command
R = READ only command
S = SET only command
S/R = SET or READ (omit parameters to read)
AR = ACTION-READ command
C = CONTROL command

**PARAMETER CODES**

n = numeric parameter
n1...nx = multiple-parameters (x is integer greater than 1)
1...x = integer numbers from 1 to x inclusive (x is integer)
na = not applicable

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| **SETTING COMMANDS** | | | | | | |
| I/E n | | S/R | Set/Read Current Range. | -3 | 0...-7 | 6 |
| AS n | | A/R | One Auto Range with Response. | na | na | 6 |
| AR n | | S/R | Set/Read Auto-Range Status. | 6 | 0...7 | 7 |
| AL n | | S/R | Set/Read Max. Auto-Range Sens. | -6 | 0...-7 | 8 |
| EGAIN n | | S/R | Set/Read pre. A/D Potential Gain. | na | 1,5,10,50 | 8 |
| ESUP n | | S/R | Set/Read potential suppression. | 0 | 5000...-5000 | 9 |
| IGAIN n | | S/R | Set/Read pre. A/D current Gain. | 1 | 1,5,10,50 | 9 |
| ISUP n | | S/R | Set/Read current suppression. | 0 | 8000...-8000 | 10 |
| SUPDAC n | | S/R | Set/Read value in /96 suppression DAC. | 0 | -8190...+8190 | 11 |
| AUXGAIN n | | S/R | Set/Read pre. A/D Aux. Gain. | na | 1,5 | 11 |
| MODE n | | S/R | Set/Read operating mode. | 2 | 0...2 | 11 |
| FLT n | | S/R | Set/Read Filter Status. | 0 | 0...57 | 11 |
| BW n | | S/R | Set/Read Bandwidth/Stability. | 0 | 0...1 | 12 |
| CELL n | | S/R | Set/Read Cell Relay Status. | 0 | 0...1 | 12 |
| EXT n | | S/R | Set/Read External Input Status. | 0 | 0...1 | 12 |
| DCL | | A | Restores Default Parameters. | na | na | 12 |
| CAL | | A | Calibrate Model 273A | na | na | 12 |
| **UNCOMPENSATED RESISTANCE COMMANDS** | | | | | | |
| IRMODE | n | S/R | Set/Read IR Compensation mode. | 0 | 0...4 | 27 |
| SETIR | n1<br>n2 | S/R | Set/Read uncomp. resis. mantissa.<br>Set/Read uncomp. resis. exponent. | 0 | 0...2000<br>-3...12 | 27<br>27 |
| COMP | n1<br>n2 | R | Read actual compen. resis. mant.<br>Read actual compen. resis. exp. | na<br>na | 0...2047<br>-3...12 | 28<br>28 |
| IRPC | n | S/R | Set/Read % correction (Irupt) | 100 | 0...200 | 28 |
| IRX | n1<br>n2<br>n3 | S/R | Sets I range for extrap. points<br>Set/Read 1st Irupt extrap. point.<br>Set/Read 2nd Irupt extrap. point. | na<br>10 or 75<br>10 or 75 | 0...-7<br>2...1997<br>2...1997 | 28<br>28 |
| DORUPT | n | AR | Initiates single cur. interrupt. | na | -10000...10000 | 29 |
| IRUPT | n | S/R | Sets # points per cur. interrupt. | 5 | 1...32767 | 29 |
| RUERR | n | R | Reads compensation potential. | na | -10000...10000 | 29 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| **MODULATION COMMANDS** | | | | | | |
| MR n | | S/R | Set/Read full-scale mod. poten. | 2 | 0...2 | 17 |
| MM n | | S/R | Set/Read Modulation Mode | 0 | 0...2 | 17 |
| INITIAL | n1<br>n2 | S | Set mod. first point location.<br>Set first point mod. level. | 0<br>-8000 | 0...6143<br>-8000...8000 | 18 |
| VERTEX | n1<br>n2 | S | Sets vertex point location.<br>Sets modulation at vertex (as many as fifty vertices may be specified). | 9998000 | 1...6143<br>-8000...8000 | 18<br>18 |
| PROG | n1<br>n2<br>n3<br>n4 | R | Reads mod. first point location.<br>Reads first point mod. level.<br>Reads vertex point location.<br>Reads modulation at vertex (additional vertices, if in program, will be similarly reported). | na<br>na<br>na<br>na | 0...6143<br>-8000...8000<br>1...6143<br>-8000...8000 | 19<br>19 |
| ASM | | A | Stores Ramp Proram in SCV. | na | na | 19 |
| MOD n | | S/R | Set/Read constant mod. level. | 0 | -8000...8000 | 19 |
| INTRP n | | S/R | Set/Read Interpolation function status. | 1 | 0...1 | 19 |
| **BIAS COMMANDS** | | | | | | |
| BIAS n | | S/R | Set/Read bias level. | 0 | -8000...8000 | 15 |
| SETE n | | S/R | Set/Read App. Potential. | 0 | -8000...8000 | 15 |
| SETI | n1<br>n2 | S/R | Set/Read App. Current mantissa.<br>Set/Read App. Current exponent. | 0<br>-6 | -2000...2000<br>-3...-10 | 16<br>16 |
| **MEMORY PARTITIONING COMMANDS** | | | | | | |
| DCV n | | S/R | Set/Read Destination Curve. | 0 | 0...5 | 14 |
| ACV | n1<br>n2 | | Set/Read Alternate Curve.<br>Set/Read number of sweep with which ACV storage begins. | 0 | 1...5<br>1...65535 | 14<br>14 |
| SCV n | | S/R | Set/Read Source Curve. | 3 | 0...5 | 14 |
| PCV n | | S/R | Set/Read Processing Curve. | 0 | 0...5 | 15 |
| **CURVE PROCESSING COMMANDS** | | | | | | |
| ADD n | | A | Add "n" to every PC point. | na | -32767...32767 | 35 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| SUB | n1<br>n2 | A | "n1" Specifies subtrahend curve.<br>"n2" Specifies minuend curve.<br>**Note:** Command causes point-by-point subtraction of subtrahend curve from minuend curve with result stored in minuend curve. | na<br>na | 0...5<br>0...5 | 35<br>35 |
| EX | n1<br>n2 | A | PCV points multiplied by n1.<br>PCV points divided by n2. | na<br>na | -32767...32767<br>-32767...32767 (but not 0) | 35<br>35 |
| MIN | n1<br>n2 | AR | Point number of minimum.<br>Data value at minimum. | na<br>na | 0...6143<br>-32767...32767 | 35 |
| IMIN | n1<br>n2 | AR | Report current minimum mantissa.<br>Report current minimum exponent. | na<br>na | -2000...2000<br>-3...-10 | 36<br>36 |
| MAX | n1<br>n2 | AR | Point number of maximum.<br>Data value at maximum. | na<br>na | 0...6143<br>-32767...32768 | 36<br>36 |
| IMAX | n1<br>n2 | AR | Report current maximum mantissa.<br>Report current maximum exponent. | na<br>na | -2000...2000<br>-3...-10 | 36<br>36 |
| INT | n1<br>n2 | AR | Sum all data in PCV. SUM = n2 + (10000 * n1). | na<br>na | na<br>na | 36<br>36 |
| IINT | n1<br>n2 | AR | Mantissa of sum of all I data.<br>Exponent of sum of all I data. | na<br>na | na<br>na | 37<br>37 |
| ILOG | | A | Replace PCV with log of PCV. | na | na | 37 |
| **SAMPLING CONTROL COMMANDS** | | | | | | |
| PAM n | | S/R | Set/Read Point Averaging Mode. | 0 | 0...2 | 22 |
| S/P n | | S/R | Set/Read Samples Per Point. | 1 | 1...32767 | 23 |
| SEL | n1<br>n2 | S/R | Set/Read first samp. to be aver.<br>Set/read last samp. to be aver. | 11 | 1...32767<br>1...32767 | 23<br>23 |
| SIE n | | S/R | Set/Read sampled parameter.<br>**Note:** SIE 16 sets the Sampled Parameter to Q. No other parameter can be collected while Q is being measured. | 1 | 0...16 | 23<br>23 |
| TMB n | | S/R | Set/Read Timebase. | 4000 | 50...50000 | 23 |
| LS n | | S/R | Set/Read Line Sync Status | 0 | 0...1 | 25 |
| **SWEEP CONTROL COMMANDS** | | | | | | |
| SWPS n | | S/R | Set/Read Number of Sweeps. | 1 | 1...65535 | 25 |
| SAM n | | S/R | Set/Read Sweep Aver. Mode. | 0 | 0...2 | 25 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| SHF n | | S/R | Set/Read number of Exp. Avg. shifts. | 1 | 1...15 | 26 |
| DT n | | S/R | Set/Read time between sweeps. | 0 | 0...65535 | 26 |
| **DATA ACQUISITION CONTROL COMMANDS** | | | | | | |
| FP n | | S/R | Set/Read first point location. | 0 | 0...6143 | 20 |
| LP n | | S/R | Set/Read last point location. | 999 | 1...6143 | 20 |
| RC | | A | Init. 273A for Curve Acquisition (unlike NC, RC doesn't clear the Destination Curve, Next Curve, or Alternate Curve). | na | na | 21 |
| NC | | A | Init. 273A for Curve Acquisition. | na | na | 21 |
| TC | | A | Starts Curve Acquisition. | na | na | 21 |
| HC | | A | Halts Curve Acquisition until TC is applied. | na | na | 21 |
| WCD | | A | Postpones next cmd. in sequence until curve acquisition is complete. | na | na | 22 |
| WAIT n | | A | Halts Curve Acq. for "n" tmb. | na | 0...65535 | 22 |
| DISCARD n A | | | Pauses data acq. for "n" points. | na | 0...65535 | 22 |
| **DATA ACQUISITION MONITORING COMMANDS** | | | | | | |
| A/D n | | R | Reads and reports A/D output. | na | -2000...2000 | 32 |
| TP | n1<br>n2<br>n3 | AR | Point # read and reported.<br>Read and report I data value.<br>Read and report E data value. | na<br>na<br>na | 0...6143<br>-2000...2000<br>-10000...10000 | 32<br>32 |
| SP | | A | Store Point. | na | na | 33 |
| PNT n | | S/R | Set/Read next point to be read. | na | 0...6143 | 33 |
| M | n1<br>n2<br>n3<br>n4<br>n5<br>n6 | R | Report Curve Acquisition Status.<br>Report Number of Current Sweep<br>Report Number of Curr. Acc. Pt.<br>Report Modulation Level.<br>Report last I value.<br>Report last E value. | na<br>na<br>na<br>na<br>na<br>na | 0...1<br>1...65535<br>0...6143<br>-8000...8000<br>0,-2000...2000<br>0,-2000...2000 | 33<br>33 |
| READE | n | AR | Take 10 E samp. Report average. | na | -10000...10000 | 34 |
| READI | n1<br>n2 | AR | Mantissa of aver. of 10 I samp.<br>Exponent of aver. of 10 I samp. | na<br>na | -2000...2000<br>-3...-10 | 34 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| READAUX n | | AR | Take 10 AUX samp. Report aver. | na | -10000...10000 | 34 |
| Q | n1<br>n2 | R | Reports coulombs (mantissa).<br>Coulombs exponent. | na<br>na | -9999...9999<br>na | 34<br>34 |
| **CURVE DATA TRANSFER COMMANDS** | | | | | | |
| CLR | | A | Sets all PCV points to "0". | na | na | 37 |
| CLEAR | | A | Sets all points to "0". | na | na | 37 |
| DC | n1<br>n2 | A | Sets # of 1st point to dump.<br>Sets number of points to dump.<br>This command is used to dump a data set. | na<br>na | 0...6143<br>1...6143 | 37<br>37 |
| DP n | | A | Sets # of point to dump. This command is used dump a point. | na | 0...6143 | 39<br>39 |
| LC | n1<br>n2 | A | Sets first storage location.<br>Sets number of points to store.<br>This command is used to store data from host computer in 273A Processing Curve. | na<br>na | 0...6143<br>1...6143 | 39<br>39 |
| COPY | n1<br>n2 | A | Curve to copy from.<br>Curve to copy to.<br>This command is used to copy data from one curve to another. | na | 0...5<br>0...5 | 39<br>39 |
| BD | n1<br>n2 | A | Sets first binary dump point.<br>Sets # of points to binary dump. | na<br>na | 0...6143<br>1...6143 | 40<br>40 |
| BL | n1<br>n2 | A | Sets first binary load location.<br>Sets # of points to binary dump. | na<br>na | 0...6143<br>0...6143 | 40<br>40 |
| **CYCLIC VOLTAMMETRY COMMANDS** | | | | | | |
| CV | n1<br>n2<br>n3<br>n4<br>n5 | S/R | Set/Read CV Start Potential.<br>Set/Read CV Vertex Potential.<br>Set/Read CV Final Potential.<br>Set/Read CV Scan Rate.<br>Read actual resolution (response CV command sent without operands) | 0 | -8000...8000<br>-8000...8000<br>-8000...8000<br>1...8000<br>125...4000 | 45<br>45 |
| SS n | | S/R | Set/Read Samples/Reading Mult. | 1 | 1...1000 | 45 |
| MRES n | | S/R | Set/Read Points/Volt Resolution. | 4000 | 125...4000 | 46 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| **SYSTEM ID COMMANDS** | | | | | | |
| VER n | | R | Read software version. | | | 46 |
| ID n | | R | Read model number. | 2731 | none | 46 |
| OPTION nn | | R | Read option status. Returns 0 if entered option number not installed; returns 1 if entered option number installed. | none | 92, 96 | 46 |
| **COMMUNICATIONS CONTROL COMMANDS** | | | | | | |
| MSK n | | S/R | Set/Read the mask byte. | 0 | 0...255 | 40 |
| DD n | | S | Set Delimiter. | 44 | 0...255 | 41 |
| **STATUS COMMANDS** | | | | | | |
| ST n | | R | Status Byte (same as Ser. Poll).<br>Bit / Decimal = Meaning<br>0 / 1 = Command Done<br>1 / 2 = Command Error<br>2 / 4 = Curve Done<br>3 / 8 = Unused<br>4 / 16 = Overload<br>5 / 32 = Sweep Done<br>6 / 64 = Service Requested<br>7 / 128 = Output Ready<br>For each bit, 0 = NO, 1 = YES | | 0...255 | 41<br>41 |
| ERR n | | R | Read Error Status<br>0 = No Error<br>1 = Option Not Installed<br>2 = Invalid Command Sent<br>3 = Parameter Error<br>4 = Command Overrun<br>5 = Nothing to Say Error<br>6 = Numeric Error<br>7 = Time Base (TMB) too Short<br>8 = Not Used<br>9 = Not Used<br>10 = Not Used<br>11 = Mode Error<br>12 = Acquisition Error | na | 0...12 | 42<br>42 |
| OVER | n1<br>n2<br>n3 | R | Read Ovl. Status at Exec. Time.<br>Read Ovl. Stat. since last OVER.<br>Read cumulative Ovl. Status. | na<br>na<br>na | 0...3<br>0...3<br>0...7 | 43<br>43 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| CS n | | R | Read Cell Enable key status. | na | 0...1 | 43 |
| DUMMY n | | R | Read Elec. Mode switch status. | na | 0...1 | 43 |
| FF n | | R | Read power frequency.<br>0 = 60 Hz<br>1 = 50 Hz | na | 0...1 | 43<br>43 |
| **CURRENT INTEGRATION COMMANDS** | | | | | | |
| INTEG | | S/R | Controls /96 iR Integrator | 0 | 0 = Reset Integ.<br>1 = Start Integ.<br>2 = Hold Integ. | 29<br>29 |
| ITC | | S/R | /96 iR Integrator Time Constant<br>10 MΩ x 20 nF<br>1 MΩ x 20 nF<br>100 kΩ x 20 nF<br>10 kΩ x 20 nF<br>**Note:** The Effective Integrator Time Constant = Integrator Time Constant/(GIGAIN * 2.5 * 2) | -1 | Effective Time Constants:<br>-1 = 40 ms<br>-2 = 4 ms<br>-3 = 400 µs<br>-4 = 40 µs | 30<br>30 |
| GIGAIN | | S/R | /96 iR Integrator Gain<br>**Note**: The counts stored in the 273A memory and the charge that flowed through the cell are related as follows:<br>$Q = 1/500 * (\text{counts} * 10^{ITC} * 10^{I/E})/\text{GIGAIN}$ | 1 | 1...500 | 30 |
| **CONTROL STRUCTURE COMMANDS** | | | | | | |
| BEGIN | | C | Put 273A in endless loop. | na | na | 47 |
| AGAIN | | C | Mark end of endless loop. | na | na | 47 |
| DO n | | C | Sets # times to repeat loop. | na | 1...32767 | 47 |
| LOOP | | C | Mark end of loop to be repeated specific number of times. | na | na | 47 |
| USR1, 2,3,4 | | C | Allows user-defined commands. | na | na | 47 |
| P n | | A | Causes pause in cmd. exec. | na | 0...65535 | 48 |
| **PANEL/DISPLAY COMMANDS** | | | | | | |
| LREF n | | S/R | Sets log reference range. | 0 | 0...-7 | 48 |
| KEY n | | A | Duplicate f.p. key press. | na | 1...60 | 48 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| OUT n | | S/R | Sets Output Mode.<br>0 = none<br>1 = LOG I<br>2 = LINEAR I<br>3 = COULOMBS<br>4 = SETOUT Controlled | 2 | 0...3 | 49<br>49 |
| SETOUT n | | S/R | mV Resolution. Control of OUT BNC. | 0 | 2047...-2047 | 50 |
| TYPE nnnn...n" | | A | Prints msg. on LCD display. | na | ASCII | 50 |
| **GENERAL INTERFACE COMMANDS** | | | | | | |
| TRIG n | | A | Puts pulse on TRIG OUT line. | na | 0...1 | 50 |
| WFT n | | A | Wait For Trigger. | na | 0...1 | 50 |
| PEN n | | S/R | Set/Read Pen Relay status. | 0 | 0...1 | 51 |
| BIT n | | R | Report Bit 0 In Line Status. | 1 | 0...1 | 51 |
| BIT | n1<br>n2 | S | Must Always be "0".<br>Sets level on Bit 0 Out Line. | na<br>na | 0<br>0...1 | 51<br>51 |
| **MODEL 303A INTERFACE COMMANDS** | | | | | | |
| DISP | | A | Dispense/Dislodge cmd. to 303A. | na | na | 51 |
| PURGE n | | S/R | Purge to 303A. | 0 | 0...1 | 51 |
| STIR n | | S/R | Stir to 305 via 303A. | 0 | 0...1 | 52 |
| **MODEL 273A/92 ELECTROCHEMICAL IMPEDANCE INTERFACE OPTION COMMANDS** | | | | | | |
| OSCIN n | | S/R | Attenuator enable | 0 | 0, 1 | 31 |
| OSCGAIN n | | S/R | Full-scale (FS) range select<br>0 = 0.02 x input voltage<br>1 = 0.2 x input voltage<br>2 = 2 x input voltage | 0 | 0, 1, 2 | 31<br>31 |
| OSC n | | S/R | Attenuation level | 800 | 0...4000 | 31 |
| OSCDC n | | S/R | DC couple attenuator input<br>0 = AC coupled<br>1 = DC coupled | 0 | 0, 1 | 31<br>31 |
| EOUTDC n | | S/R | DC couple E output<br>0 = AC coupled<br>1 = DC coupled | 0 | 0, 1 | 31<br>31 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| IOUTDC n | | S/R | DC couple I output<br>0 = AC coupled<br>1 = DC coupled | 0 | 0, 1 | 31 |
| EOUTSUP n | | S/R | E DC suppression. 1 bit = 2mV suppression. | 0 | -5000...+5000 | 31 |
| IOUTSUP n | | S/R | I DC suppression. 1 bit = 0.5 x $10^{-3}$ x current range (e.g., 1 bit = 0.5 µA on the 1 mA current range). | 0 | -8000...+8000 | 31 |
| MIE n | | S/R | Measure I/E<br>0 = output current at AC I OUTPUT, and voltage at AC E OUTPUT<br>1 = output current at MULTIPLEXED OUTPUT<br>2 = output voltage at MULTIPLEXED OUTPUT | 1 | 0, 1, 2 | 32 |

# APPENDIX H. ABBREVIATED COMMAND DESCRIPTIONS: ALPHABETICAL INDEX

## H.1. Introduction

The table beginning on page 124 uses the following definitions:

**COMMAND TYPE CODES**

| | | |
|---|---|---|
| A | = | ACTION command |
| R | = | READ only command |
| S | = | SET only command |
| S/R | = | SET or READ (omit parameters to read) |
| AR | = | ACTION-READ command |
| C | = | CONTROL command |

**PARAMETER CODES**

| | | |
|---|---|---|
| n | = | numeric parameter |
| n1...nx | = | multiple-parameters (x is integer greater than 1) |
| 1...x | = | integer numbers from 1 to x inclusive (x is integer) |
| na | = | not applicable |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| A/D n | | R | Reads and reports A/D output. | na | -2000...2000 | 32 |
| ACV | n1<br>n2 | | Set/Read Alternate Curve.<br>Set/Read number of sweep with which ACV storage begins. | 0 | 1...5<br>1...65535 | 14<br>14 |
| ADD n | | A | Add "n" to every PC point. | na | -32767...32767 | 35 |
| AGAIN | | C | Mark end of endless loop. | na | na | 47 |
| AL n | | S/R | Set/Read Max. Auto-Range Sens. | -6 | 0...-7 | 8 |
| AR n | | S/R | Set/Read Auto-Range Status. | 6 | 0...7 | 7 |
| AS n | | A/R | One Auto Range with Response. | na | na | 6 |
| ASM | | A | Stores Ramp Proram in SCV. | na | na | 19 |
| AUXGAIN n | | S/R | Set/Read pre. A/D Aux. Gain. | na | 1,5 | 11 |
| BD | n1<br>n2 | A | Sets first binary dump point.<br>Sets # of points to binary dump. | na<br>na | 0...6143<br>1...6143 | 40<br>40 |
| BEGIN | | C | Put 273A in endless loop. | na | na | 47 |
| BIAS n | | S/R | Set/Read bias level. | 0 | -8000...8000 | 15 |
| BIT n | | R | Report Bit 0 In Line Status. | 1 | 0...1 | 51 |
| BIT | n1<br>n2 | S | Must Always be "0".<br>Sets level on Bit 0 Out Line. | na<br>na | 0<br>0...1 | 51<br>51 |
| BL | n1<br>n2 | A | Sets first binary load location.<br>Sets # of points to binary dump. | na<br>na | 0...6143<br>0...6143 | 40<br>40 |
| BW n | | S/R | Set/Read Bandwidth/Stability. | 0 | 0...1 | 12 |
| CAL | | A | Calibrate Model 273A | na | na | 12 |
| CELL n | | S/R | Set/Read Cell Relay Status. | 0 | 0...1 | 12 |
| CLEAR | | A | Sets all points to "0". | na | na | 37<br>37 |
| CLR | | A | Sets all PCV points to "0". | na | na | 37 |
| COMP | n1<br>n2 | R | Read actual compen. resis. mant.<br>Read actual compen. resis. exp. | na<br>na | 0...2047<br>-3...12 | 28<br>28 |
| COPY | n1<br>n2 | A | Curve to copy from.<br>Curve to copy to.<br>This command is used to copy data from one curve to another. | na | 0...5<br>0...5 | 39<br>39 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| CV | n1<br>n2<br>n3<br>n4<br>n5 | S/R | Set/Read CV Start Potential.<br>Set/Read CV Vertex Potential.<br>Set/Read CV Final Potential.<br>Set/Read CV Scan Rate.<br>Read actual resolution (response CV command sent without operands) | 0 | -8000...8000<br>-8000...8000<br>-8000...8000<br>1...8000<br>125...4000 | 45<br>45 |
| DC | n1<br>n2 | A | Sets # of 1st point to dump.<br>Sets number of points to dump.<br>This command is used to dump a data set. | na<br>na | 0...6143<br>1...6143 | 37<br>37 |
| DCL | | A | Restores Default Parameters. | na | na | 12 |
| DCV n | | S/R | Set/Read Destination Curve. | 0 | 0...5 | 14 |
| DD n | | S | Set Delimiter. | 44 | 0...255 | 41 |
| DISCARD n A | | | Pauses data acq. for "n" points. | na | 0...65535 | 22 |
| DISP | | A | Dispense/Dislodge cmd. to 303A. | na | na | 51 |
| DO n | | C | Sets # times to repeat loop. | na | 1...32767 | 47 |
| DORUPT | n | AR | Initiates single cur. interrupt. | na | -10000...10000 | 29 |
| DP n | | A | Sets # of point to dump. This command is used dump a point. | na | 0...6143 | 39<br>39 |
| DT n | | S/R | Set/Read time between sweeps. | 0 | 0...65535 | 26<br>26 |
| DUMMY n | | R | Read Elec. Mode switch status. | na | 0...1 | 43<br>43 |
| EGAIN n | | S/R | Set/Read pre. A/D Potential Gain. | na | 1,5,10,50 | 88 |
| EOUTDC n | | S/R | DC couple E output<br>0 = AC coupled<br>1 = DC coupled | 0 | 0, 1 | 31<br>31 |
| EOUTSUP n | | S/R | E DC suppression. 1 bit = 2mV suppression. | 0 | -5000...+5000 | 31<br>31 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| ERR n | | R | Read Error Status<br>0 = No Error<br>1 = Option Not Installed<br>2 = Invalid Command Sent<br>3 = Parameter Error<br>4 = Command Overrun<br>5 = Nothing to Say Error<br>6 = Numeric Error<br>7 = Time Base (TMB) too Short<br>8 = Not Used<br>9 = Not Used<br>10 = Not Used<br>11 = Mode Error<br>12 = Acquisition Error | na | 0...12 | 42<br>42 |
| ESUP n | | S/R | Set/Read potential suppression. | 0 | 5000...-5000 | 9 |
| EX | n1<br>n2 | A | PCV points multiplied by n1.<br>PCV points divided by n2. | na<br>na | -32767...32767<br>-32767...32767 (but not 0) | 35<br>35 |
| EXT n | | S/R | Set/Read External Input Status. | 0 | 0...1 | 12 |
| FF n | | R | Read power frequency.<br>0 = 60 Hz<br>1 = 50 Hz | na | 0...1 | 43<br>43 |
| FLT n | | S/R | Set/Read Filter Status. | 0 | 0...57 | 11 |
| FP n | | S/R | Set/Read first point location. | 0 | 0...6143 | 20 |
| GIGAIN | | S/R | /96 iR Integrator Gain<br>**Note**: The counts stored in the 273A memory and the charge that flowed through the cell are related as follows:<br>$Q = 1/500 * (counts * 10^{ITC} * 10^{I/E})/GIGAIN$ | 1 | 1...500 | 30 |
| HC | | A | Halts Curve Acquisition until TC is applied. | na | na | 21 |
| I/E n | | S/R | Set/Read Current Range. | -3 | 0...-7 | 66 |
| ID n | | R | Read model number. | 2731 | none | 46 |
| IGAIN n | | S/R | Set/Read pre. A/D current Gain. | 1 | 1,5,10,50 | 9 |
| IINT | n1<br>n2 | AR | Mantissa of sum of all I data.<br>Exponent of sum of all I data. | na<br>na | na<br>na | 37<br>37 |
| ILOG | | A | Replace PCV with log of PCV. | na | na | 37<br>37 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| IMIN | n1<br>n2 | AR | Report current minimum mantissa.<br>Report current minimum exponent. | na<br>na | -2000...2000<br>-3...-10 | 36<br>36 |
| INITIAL | n1<br>n2 | S | Set mod. first point location.<br>Set first point mod. level. | 0<br>-8000 | 0...6143<br>-8000...8000 | 18 |
| INT | n1<br>n2 | AR | Sum all data in PCV. SUM = n2 + (10000 * n1). | na<br>na | na<br>na | 36<br>36 |
| INTEG | | S/R | Controls /96 iR Integrator | 0 | 0 = Reset Integ.<br>1 = Start Integ.<br>2 = Hold Integ. | 29<br>29 |
| INTRP n | | S/R | Set/Read Interpolation function status. | 1 | 0...1 | 19<br>19 |
| IOUTDC n | | S/R | DC couple I output<br>0 = AC coupled<br>1 = DC coupled | 0 | 0, 1 | 31<br>31 |
| IOUTSUP n | | S/R | I DC suppression. 1 bit = 0.5 x $10^{-3}$ x current range (e.g., 1 bit = 0.5 µA on the 1 mA current range). | 0 | -8000...+8000 | 31<br>31 |
| IRMODE | n | S/R | Set/Read IR Compensation mode. | 0 | 0...4 | 27 |
| IRPC | n | S/R | Set/Read % correction (Irupt) | 100 | 0...200 | 28 |
| IRUPT | n | S/R | Sets # points per cur. interrupt. | 5 | 1...32767 | 29 |
| IRX | n1<br>n2<br>n3 | S/R | Sets I range for extrap. points<br>Set/Read 1st Irupt extrap. point.<br>Set/Read 2nd Irupt extrap. point. | na<br>10 or 75<br>10 or 75 | 0...-7<br>2...1997<br>2...1997 | 28<br>28 |
| ISUP n | | S/R | Set/Read current suppression. | 0 | 8000...-8000 | 10 |
| ITC | | S/R | /96 iR Integrator Time Constant<br>10 MΩ x 20 nF<br>1 MΩ x 20 nF<br>100 kΩ x 20 nF<br>10 kΩ x 20 nF<br>**Note:** The Effective Integrator Time Constant = Integrator Time Constant/(GIGAIN * 2.5 * 2) | -1 | Effective Time Constants:<br>-1 = 40 ms<br>-2 = 4 ms<br>-3 = 400 µs<br>-4 = 40 µs | 30<br>30 |
| KEY n | | A | Duplicate f.p. key press. | na | 1...60 | 48<br>48 |
| LC | n1<br>n2 | A | Sets first storage location.<br>Sets number of points to store.<br>This command is used to store data from host computer in 273A Processing Curve. | na<br>na | 0...6143<br>1...6143 | 39<br>39 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| LP n | | S/R | Set/Read last point location. | 999 | 1...6143 | 20 |
| LREF n | | S/R | Sets log reference range. | 0 | 0...-7 | 48<br>48 |
| LS n | | S/R | Set/Read Line Sync Status | 0 | 0...1 | 25 |
| M | n1<br>n2<br>n3<br>n4<br>n5<br>n6 | R | Report Curve Acquisition Status.<br>Report Number of Current Sweep<br>Report Number of Curr. Acc. Pt.<br>Report Modulation Level.<br>Report last I value.<br>Report last E value. | na<br>na<br>na<br>na<br>na<br>na | 0...1<br>1...65535<br>0...6143<br>-8000...8000<br>0,-2000...2000<br>0,-2000...2000 | 33<br>33 |
| MAX | n1<br>n2 | AR | Point number of maximum.<br>Data value at maximum. | na<br>na | 0...6143<br>-32767...32768 | 36<br>36 |
| MIE n | | S/R | Measure I/E<br>0 = output current at AC I OUTPUT, and voltage at AC E OUTPUT<br>1 = output current at MULTIPLEXED OUTPUT<br>2 = output voltage at MULTIPLEXED OUTPUT | 1 | 0, 1, 2 | 32<br>32 |
| MIN | n1<br>n2 | AR | Point number of minimum.<br>Data value at minimum. | na<br>na | 0...6143<br>-32767...32767 | 35 |
| MM n | | S/R | Set/Read Modulation Mode | 0 | 0...2 | 17 |
| MOD n | | S/R | Set/Read constant mod. level. | 0 | -8000...8000 | 19 |
| MODE n | | S/R | Set/Read operating mode. | 2 | 0...2 | 11 |
| MR n | | S/R | Set/Read full-scale mod. poten. | 2 | 0...2 | 17 |
| MRES n | | S/R | Set/Read Points/Volt Resolution. | 4000 | 125...4000 | 46 |
| MSK n | | S/R | Set/Read the mask byte. | 0 | 0...255 | 40 |
| NC | | A | Init. 273A for Curve Acquisition. | na | na | 21 |
| OPTION nn | | R | Read option status. Returns 0 if entered option number not installed; returns 1 if entered option number installed. | none | 92, 96 | 46<br>46 |
| OSC n | | S/R | Attenuation level | 800 | 0...4000 | 31<br>31 |
| OSCDC n | | S/R | DC couple attenuator input<br>0 = AC coupled<br>1 = DC coupled | 0 | 0, 1 | 31<br>31 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| OSCIN n | | S/R | Attenuator enable | 0 | 0, 1 | 31 |
| OUT n | | S/R | Sets Output Mode.<br>0 = none<br>1 = LOG I<br>2 = LINEAR I<br>3 = COULOMBS<br>4 = SETOUT Controlled | 2 | 0...3 | 49<br>49 |
| OVER | n1<br>n2<br>n3 | R | Read Ovl. Status at Exec. Time.<br>Read Ovl. Stat. since last OVER.<br>Read cumulative Ovl. Status. | na<br>na<br>na | 0...3<br>0...3<br>0...7 | 43<br>43 |
| P n | | A | Causes pause in cmd. exec. | na | 0...65535 | 48 |
| PAM n | | S/R | Set/Read Point Averaging Mode. | 0 | 0...2 | 22 |
| PCV n | | S/R | Set/Read Processing Curve. | 0 | 0...5 | 15 |
| PEN n | | S/R | Set/Read Pen Relay status. | 0 | 0...1 | 51 |
| PNT n | | S/R | Set/Read next point to be read. | na | 0...6143 | 33 |
| PROG | n1<br>n2<br>n3<br>n4 | R | Reads mod. first point location.<br>Reads first point mod. level.<br>Reads vertex point location.<br>Reads modulation at vertex (additional vertices, if in program, will be similarly reported). | na<br>na<br>na<br>na | 0...6143<br>-8000...8000<br>1...6143<br>-8000...8000 | 19<br>19 |
| PURGE n | | S/R | Purge to 303A. | 0 | 0...1 | 51<br>51 |
| Q | n1<br>n2 | R | Reports coulombs (mantissa).<br>Coulombs exponent. | na<br>na | -9999...9999<br>na | 34<br>34 |
| RC | | A | Init. 273A for Curve Acquisition (unlike NC, RC doesn't clear the Destination Curve, Next Curve, or Alternate Curve). | na | na | 21 |
| READAUX n | | AR | Take 10 AUX samp. Report aver. | na | -10000...10000 | 34<br>34 |
| READE | n | AR | Take 10 E samp. Report average. | na | -10000...10000 | 34 |
| READI | n1<br>n2 | AR | Mantissa of aver. of 10 I samp.<br>Exponent of aver. of 10 I samp. | na<br>na | -2000...2000<br>-3...-10 | 34 |
| RUERR | n | R | Reads compensation potential. | na | -10000...10000 | 29<br>29 |
| S/P n | | S/R | Set/Read Samples Per Point. | 1 | 1...32767 | 23 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| SEL | n1<br>n2 | S/R | Set/Read first samp. to be aver.<br>Set/read last samp. to be aver. | 11 | 1...32767<br>1...32767 | 23<br>23 |
| SETE n | | S/R | Set/Read App. Potential. | 0 | -8000...8000 | 15 |
| SETI | n1<br>n2 | S/R | Set/Read App. Current mantissa.<br>Set/Read App. Current exponent. | 0<br>-6 | -2000...2000<br>-3...-10 | 16<br>16 |
| SETIR | n1<br>n2 | S/R | Set/Read uncomp. resis. mantissa.<br>Set/Read uncomp. resis. exponent. | 0 | 0...2000<br>-3...12 | 27<br>27 |
| SETOUT n | | S/R | mV Resolution. Control of OUT BNC. | 0 | 2047...-2047 | 50 |
| SHF n | | S/R | Set/Read number of Exp. Avg. shifts. | 1 | 1...15 | 26<br>26 |
| SIE n | | S/R | Set/Read sampled parameter.<br>**Note:** SIE 16 sets the Sampled Parameter to Q. No other parameter can be collected while Q is being measured. | 1 | 0...16 | 23<br>23 |
| SP | | A | Store Point. | na | na | 33 |
| SS n | | S/R | Set/Read Samples/Reading Mult. | 1 | 1...1000 | 45 |
| ST n | | R | Status Byte (same as Ser. Poll).<br>Bit / Decimal = Meaning<br>0 / 1 = Command Done<br>1 / 2 = Command Error<br>2 / 4 = Curve Done<br>3 / 8 = Unused<br>4 / 16 = Overload<br>5 / 32 = Sweep Done<br>6 / 64 = Service Requested<br>7 / 128 = Output Ready<br>For each bit, 0 = NO, 1 = YES | | 0...255 | 41<br>41 |
| STIR n | | S/R | Stir to 305 via 303A. | 0 | 0...1 | 52<br>52 |
| SUB | n1<br>n2 | A | "n1" Specifies subtrahend curve.<br>"n2" Specifies minuend curve.<br>**Note:** Command causes point-by-point subtraction of subtrahend curve from minuend curve with result stored in minuend curve. | na<br>na | 0...5<br>0...5 | 35<br>35 |
| SUPDAC n | | S/R | Set/Read value in /96 suppression DAC. | 0 | -8190...+8190 | 11 |
| SWPS n | | S/R | Set/Read Number of Sweeps. | 1 | 1...65535 | 25 |
| TC | | A | Starts Curve Acquisition. | na | na | 21 |

| | | TYPE | FUNCTION | DEF. VALUE | PARAMETER RANGE | PG |
|---|---|---|---|---|---|---|
| TP | n1<br>n2<br>n3 | AR | Point # read and reported.<br>Read and report I data value.<br>Read and report E data value. | na<br>na<br>na | 0...6143<br>-2000...2000<br>-10000...10000 | 32<br>32 |
| TRIG n | | A | Puts pulse on TRIG OUT line. | na | 0...1 | 50 |
| TYPE nnnn...n" | | A | Prints msg. on LCD display. | na | ASCII | 50<br>50 |
| USR1, 2,3,4 | | C | Allows user-defined commands. | na | na | 47 |
| VER n | | R | Read software version. | | | 46 |
| VERTEX | n1<br>n2 | S | Sets vertex point location.<br>Sets modulation at vertex (as many as fifty vertices may be specified). | 9998000 | 1...6143<br>-8000...8000 | 18<br>18 |
| WAIT n | | A | Halts Curve Acq. for "n" tmb. | na | 0...65535 | 22 |
| WCD | | A | Postpones next cmd. in sequence until curve acquisition is complete. | na | na | 22<br>22 |
| WFT n | | A | Wait For Trigger. | na | 0...1 | 50 |

# INDEX

# COMPRESSED COMMAND INDEX

**COMMAND TYPE CODES**

A = ACTION command
R = READ only command
S = SET only command
S/R = SET or READ (omit parameters to read)
AR = ACTION-READ command
S = SET only command

**PARAMETER CODES**

n = numeric parameter
n1...nx = multiple-parameters (x is integer greater than 1)
1...x = integer numbers from 1 to x inclusive (x is integer)
na = not applicable

| CMD | TYPE | | FUNCTION | DEFAULT | RANGE | PG |
|---|---|---|---|---|---|---|
| A/D | n | R | Reads and reports A/D output. | na | -2000...2000 | 23 |
| ACV | n1 | S/R | Set/Read Alternate Curve. | 0 | 1...5 | 14 |
| | n2 | | Set/Read number of sweep | | | |
| | | | with which ACV storage begins. | 0 | 1...65535 | |
| ADD | n | A | Add "n" to every PC point. | na | -32767...32767 | 35 |
| AGAIN | | C | Mark end of endless loop. | na | na | 47 |
| AL | n | S/R | Set/Read Max. Auto-Range Sens. | -6 | 0...-7 | 8 |
| AR | n | S/R | Set/Read Auto-Range Status. | 6 | 0...7 | 7 |
| AS | n | A/R | One Auto Range with Response. | na | na | 6 |
| ASM | | A | Stores Ramp Proram in SCV. | na | na | 19 |
| AUXGAIN | n | S/R | Set/Read pre. A/D Aux. Gain. | na | 1,5 | 11 |
| BEGIN | C | | Put 273A in endless loop. | na | na | 47 |
| BD | n1 | A | Sets first binary dump point. | na | 0...6143 | 40 |
| | n2 | | Sets # of points to binary dump. | na | 1...6143 | |
| BIAS | n | S/R | Set/Read bias level. | 0 | -8000...8000 | 15 |
| BIT | n | R | Report Bit 0 In Line Status. | 1 | 0...1 | 51 |
| BIT | n1 | S | Must Always be "0". | na | 0 | 51 |
| | n2 | | Sets level on Bit 0 Out Line. | na | 0...1 | |
| BL | n1 | A | Sets first binary load location. | na | 0...6143 | 40 |
| | n2 | | Sets # of points to binary dump. | na | 0...6143 | |
| BW | n | S/R | Set/Read Bandwidth/Stability. | 0 | 0...1 | 12 |
| CAL | | A | Calibrate Model 273A | na | na | 12 |
| CELL | n | S/R | Set/Read Cell Relay Status. | 0 | 0...1 | 12 |
| CLEAR | | A | Sets all points to "0". | na | na | 37 |
| CLR | | A | Sets all PCV points to "0". | na | na | 37 |
| COMP | n1 | R | Read actual compen. resis. mant. | na | 0...2047 | 28 |
| | n2 | | Read actual compen. resis. exp. | na | -3...12 | |
| COPY | n1 | A | Curve to copy from. | na | 0...5 | 39 |
| | n2 | | Curve to copy to. | 0...5 | | |
| | | | This command is used to copy data | | | |
| | | | from one curve to another. | | | |
| CS | n | R | Read Cell Enable key status. | na | 0...1 | 43 |
| CV | n1 | S/R | Set/Read CV Start Potential. | 0 | -8000...8000 | 45 |
| | n2 | | Set/Read CV Vertex Potential. | 0 | -8000...8000 | |
| | n3 | | Set/Read CV Final Potential. | 0 | -8000...8000 | |
| | n4 | | Set/Read CV Scan Rate. | 0 | 1...8000 | |
| | n5 | | Read actual resolution (response | | | |
| | | | CV command sent without operands) | 0 | 125...4000 | |
| DC | n1 | A | Sets # of 1st point to dump. | na | 0...6143 | 37 |
| | n2 | | Sets number of points to dump. | na | 1...6143 | |
| | | | This command is used to dump a data set. | | | |
| DCL | | A | Restores Default Parameters. | na | na | 12 |
| DCV | n | S/R | Set/Read Destination Curve. | 0 | 0...5 | 14 |
| DD | n | S | Set Delimiter. | 44 | 0...255 | 41 |
| DISCARD | n | A | Pauses data acq. for "n" points. | na | 0...65535 | 22 |
| DISP | | A | Dispense/Dislodge cmd. to 303A. | na | na | 51 |
| DO | n | C | Sets # times to repeat loop. | na | 1...32767 | 47 |
| DORUPT | n | AR | Initiates single cur. interrupt. | na | -10000...10000 | 29 |
| DP | n | A | Sets # of point to dump. | na | 0...6143 | 39 |
| | | | This command is used dump a point. | | | |
| DT | n | S/R | Set/Read time between sweeps. | 0 | 0...65535 | 26 |
| DUMMY | n | R | Read Elec. Mode switch status. | na | 0...1 | 43 |
| EGAIN | n | S/R | Set/Read pre. A/D Potential Gain. | na | 1,5,10,50 | 8 |
| EOUTDC | n | S/R | DC couple E output | 0 | 0, 1 | 31 |
| | | | 0 = AC coupled | | | |
| | | | 1 = DC coupled | | | |
| EOUTSUP | n | S/R | E DC suppression | 0 | -5000...+5000 | 31 |
| | | | 1 bit = 2mV suppression | | | |
| ERR | n | R | Read Error Status | na | 0...12 | 42 |
| | | | 0 = No Error | | | |
| | | | 1 = Option Not Installed | | | |
| | | | 2 = Invalid Command Sent | | | |
| | | | 3 = Parameter Error | | | |
| | | | 4 = Command Overrun | | | |
| | | | 5 = Nothing to Say Error | | | |
| | | | 6 = Numeric Error | | | |
| | | | 7 = Time Base (TMB) too Short | | | |
| | | | 8 = Not Used | | | |
| | | | 9 = Not Used | | | |
| | | | 10 = Not Used | | | |
| | | | 11 = Mode Error | | | |
| | | | 12 = Acquisition Error | | | |
| ESUP | n | S/R | Set/Read potential suppression. | 0 | 5000...-5000 | 9 |
| EX | n1 | A | PCV points multiplied by n1. | na | -32767...32767 | 35 |
| | n2 | | PCV points divided by n2. | na | -32767...32767 | |
| | | | | | (but not 0) | |
| EXT | n | S/R | Set/Read External Input Status. | 0 | 0...1 | 12 |
| FF | n | R | Read power frequency. | na | 0...1 | 43 |
| | | | 0 = 60 Hz | | | |
| | | | 1 = 50 Hz | | | |
| FLT | n | S/R | Set/Read Filter Status. | 0 | 0...57 | 11 |
| FP | n | S/R | Set/Read first point location. | 0 | 0...6143 | 20 |
| GIGAIN | | S/R | /96 iR Integrator Gain | 1 | 1...500 | 30 |
| | | | **Note:** The counts stored in the 273A | | | |
| | | | memory and the charge that flowed | | | |
| | | | through the cell are related as follows: | | | |
| | | | Q = 1/500 * (counts * $10^{ITC}$ * $10^{I/E}$)/GIGAIN | | | |
| HC | | A | Halts Curve Acquisition until | | | |
| | | | TC is applied. | na | na | 21 |
| I/E | n | S/R | Set/Read Current Range. | -3 | 0...-7 | 6 |
| ID | n | R | Read model number. | 2731 | none | 46 |
| IGAIN | n | S/R | Set/Read pre. A/D current Gain. | 1 | 1,5,10,50 | 9 |
| IINT | n1 | AR | Mantissa of sum of all I data. | na | na | 37 |
| | n2 | | Exponent of sum of all I data. | na | na | |
| ILOG | | A | Replace PCV with log of PCV. | na | na | 37 |
| IMAX | n1 | AR | Report current maximum mantissa. | na | -2000...2000 | 36 |
| | n2 | | Report current maximum exponent. | na | -3...-10 | |

| CMD | TYPE | | FUNCTION | DEFAULT | RANGE | PG |
|---|---|---|---|---|---|---|
| IMIN | n1 | AR | Report current minimum mantissa. | na | -2000...2000 | 36 |
| | n2 | | Report current minimum exponent. | na | -3...-10 | |
| INITIAL | n1 | S | Set mod. first point location. | 0 | 0...6143 | 18 |
| | n2 | | Set first point mod. level. | -8000 | -8000...8000 | |
| INT | n1 | AR | Sum all data in PCV. SUM = | na | na | 36 |
| | n2 | | n2 + (10000 * n1). | na | na | |
| INTRP | n | S/R | Set/Read Interpolation function status. | 1 | 0...1 | 19 |
| IOUTDC | n | S/R | DC couple I output | 0 | 0, 1 | 31 |
| | | | 0 = AC coupled | | | |
| | | | 1 = DC coupled | | | |
| IOUTSUP | n | S/R | I DC suppression | 0 | -8000...+8000 | 31 |
| | | | 1 bit = 0.5 x $10^{-3}$ x current range | | | |
| | | | (e.g., 1 bit = 0.5 µA on the | | | |
| | | | 1 mA current range) | | | |
| INTEG | | S/R | Controls /96 iR Integrator | 0 | 0 = Reset Integ. | 29 |
| | | | | | 1 = Start Integ. | |
| | | | | | 2 = Hold Integ. | |
| IRMODE | n | S/R | Set/Read IR Compensation mode. | 0 | 0...4 | 27 |
| IRPC | n | S/R | Set/Read % correction (Irupt) | 100 | 0...200 | 28 |
| IRUPT | n | S/R | Sets # points per cur. interrupt. | 5 | 1...32767 | 29 |
| IRX | n1 | S/R | Sets I range for extrap. points | na | 0...-7 | 28 |
| | n2 | | Set/Read 1st Irupt extrap. point. | 10 or 75 | 2...1997 | |
| | n3 | | Set/Read 2nd Irupt extrap. point. | 10 or 75 | 2...1997 | |
| ISUP | n | S/R | Set/Read current suppression. | 0 | 8000...-8000 | 10 |
| ITC | S/R | | /96 iR Integrator Time Constant | -1 | Effective TC | 30 |
| | | | 10 MΩ x 20 nF | | -1 = 40 ms | |
| | | | 1 MΩ x 20 nF | | -2 = 4 ms | |
| | | | 100 kΩ x 20 nF | | -3 = 400 µs | |
| | | | 10 kΩ x 20 nF | | -4 = 40 µs | |
| | | | **Note:** The Effective Integrator Time Constant = | | | |
| | | | Integrator Time Constant/(GIGAIN * 2.5 * 2) | | | |
| KEY | n | A | Duplicate f.p. key press. | na | 1...60 | 48 |
| LC | n1 | A | Sets first storage location. | na | 0...6143 | 39 |
| | n2 | | Sets number of points to store. | na | 1...6143 | |
| | | | This command is used to store data from | | | |
| | | | host computer in 273A Processing Curve. | | | |
| LOOP | | C | Mark end of loop to be repeated | | | |
| | | | specific number of times. | na | na | 47 |
| LP | n | S/R | Set/Read last point location. | 999 | 1...6143 | 20 |
| LREF | n | S/R | Sets log reference range. | 0 | 0...-7 | 48 |
| LS | n | S/R | Set/Read Line Sync Status | 0 | 0...1 | 25 |
| M | n1 | R | Report Curve Acquisition Status. | na | 0...1 | 33 |
| | n2 | | Report Number of Current Sweep. | na | 1...65535 | |
| | n3 | | Report Number of Curr. Acc. Pt. | na | 0...6143 | |
| | n4 | | Report Modulation Level. | na | -8000...8000 | |
| | n5 | | Report last I value. | na | 0,-2000...2000 | |
| | n6 | | Report last E value. | na | 0,-2000...2000 | |
| MAX | n1 | AR | Point number of maximum. | na | 0...6143 | 36 |
| | n2 | | Data value at maximum. | na | -32767...32768 | |
| MIE | n | S/R | Measure I/E | 1 | 0, 1, 2 | 32 |
| | | | 0 = output current at AC I OUTPUT, | | | |
| | | | output voltage at AC E OUTPUT | | | |
| | | | 1 = output current at MULTIPLEXED OUTPUT | | | |
| | | | 2 = output voltage at MULTIPLEXED OUTPUT | | | |
| MIN | n1 | AR | Point number of minimum. | na | 0...6143 | 35 |
| | n2 | | Data value at minimum. | na | -32767...32767 | |
| MM | n | S/R | Set/Read Modulation Mode | 0 | 0...2 | 17 |
| MOD | n | S/R | Set/Read constant mod. level. | 0 | -8000...8000 | 19 |
| MODE | n | S/R | Set/Read operating mode. | 2 | 0...2 | 11 |
| MR | n | S/R | Set/Read full-scale mod. poten. | 2 | 0...2 | 17 |
| MRES | n | S/R | Set/Read Points/Volt Resolution. | 4000 | 125...4000 | 46 |
| MSK | n | S/R | Set/Read the mask byte. | 0 | 0...255 | 40 |
| NC | | A | Init. 273A for Curve Acquisition. | na | na | 21 |
| OPTION | nn | R | Read option status | none | 92, 96 | 46 |
| | | | returns 0 if entered option number not installed | | | |
| | | | returns 1 if entered option number installed | | | |
| OSC | n | S/R | Attenuation level | 800 | 0...4000 | 31 |
| OSCDC | n | S/R | DC couple attenuator input | 0 | 0, 1 | 31 |
| | | | 0 = AC coupled | | | |
| | | | 1 = DC coupled | | | |
| OSCGAIN | n | S/R | Full-scale (FS) range select | 0 | 0, 1, 2 | 31 |
| | | | 0 = 0.02 x input voltage | | | |
| | | | 1 = 0.2 x input voltage | | | |
| | | | 2 = 2 x input voltage | | | |
| OSCIN | n | S/R | Attenuator enable | 0 | 0, 1 | 31 |
| OUT | n | S/R | Sets Output Mode. | 2 | 0...3 | 51 |
| | | | 0 = none | | | |
| | | | 1 = LOG I | | | |
| | | | 2 = LINEAR I | | | |
| | | | 3 = COULOMBS | | | |
| | | | 4 = SETOUT Controlled | | | |
| OVER | n1 | R | Read Ovl. Status at Exec. Time. | na | 0...3 | 43 |
| | n2 | | Read Ovl. Stat. since last OVER. | na | 0...3 | |
| | n3 | | Read cumulative Ovl. Status. | na | 0...7 | |
| P | n | A | Causes pause in cmd. exec. | na | 0...65535 | 48 |
| PAM | n | S/R | Set/Read Point Averaging Mode. | 0 | 0...2 | 22 |
| PCV | n | S/R | Set/Read Processing Curve. | 0 | 0...5 | 15 |
| PEN | n | S/R | Set/Read Pen Relay status. | 0 | 0...1 | 51 |
| PNT | n | S/R | Set/Read next point to be read. | na | 0...6143 | 33 |
| PROG | n1 | R | Reads mod. first point location. | na | 0...6143 | 19 |
| | n2 | | Reads first point mod. level. | na | -8000...8000 | |
| | n3 | | Reads vertex point location. | na | 1...6143 | |
| | n4 | | Reads modulation at vertex. | na | -8000...8000 | |
| | | | (additional vertices, if in program, | | | |
| | | | will be similarly reported) | | | |
| PURGE | n | S/R | Purge to 303A. | 0 | 0...1 | 51 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Q | n1 | R | Reports coulombs (mantissa). | na | -9999...9999 | 34 |
| | n2 | | Coulombs exponent. | na | na | |
| RC | | A | Init. 273A for Curve Acquisition. | na | na | 21 |
| | | | (Unlike NC, RC doesn't clear the | | | |
| | | | Destination Curve, Next Curve, | | | |
| | | | or Alternate Curve) | | | |
| READAUX | n | AR | Take 10 AUX samp. Report aver. | na | -10000...10000 | 34 |
| READE | n | AR | Take 10 E samp. Report average. | na | -10000...10000 | 34 |
| READI | n1 | AR | Mantissa of aver. of 10 I samp. | na | -2000...2000 | 34 |
| | n2 | | Exponent of aver. of 10 I samp. | na | -3...-10 | |
| RUERR | n | R | Reads compensation potential. | na | -10000...10000 | 29 |
| S/P | n | S/R | Set/Read Samples Per Point. | 1 | 1...32767 | 23 |
| SAM | n | S/R | Set/Read Sweep Aver. Mode. | 0 | 0...2 | 25 |
| SCV | n | S/R | Set/Read Source Curve. | 3 | 0...5 | 14 |
| SEL | n1 | S/R | Set/Read first samp. to be aver. | 1 | 1...32767 | 23 |
| | n2 | | Set/read last samp. to be aver. | 1 | 1...32767 | |
| SETE | n | S/R | Set/Read App. Potential. | 0 | -8000...8000 | 15 |
| SETI | n1 | S/R | Set/Read App. Current mantissa. | 0 | -2000...2000 | 16 |
| | n2 | | Set/Read App. Current exponent. | -6 | -3...-10 | |
| SETIR | n1 | S/R | Set/Read uncomp. resis. mantissa. | 0 | 0...2000 | 27 |
| | n2 | | Set/Read uncomp. resis. exponent. | 0 | -3...12 | |
| SETOUT | n | S/R | mV Resolution. Control of OUT BNC. | 0 | 2047...-2047 | 50 |
| SHF | n | S/R | Set/Read number of Exp. Avg.shifts. | 1 | 1...15 | 26 |
| SIE | n | S/R | Set/Read sampled parameter. | 1 | 0...16 | 23 |
| | | | **Note:** SIE 16 sets the Sampled Parameter | | | |
| | | | to Q. No other parameter can be | | | |
| | | | collected while Q is being measured. | | | |
| SP | | A | Store Point. | na | na | 33 |
| SS | n | S/R | Set/Read Samples/Reading Mult. | 1 | 1...1000 | 45 |
| ST | n | R | Status Byte (same as Ser. Poll). | 0...255 | 41 | |
| | | | Bit / Decimal = Meaning | | | |
| | | | 0 / 1 = Command Done | | | |
| | | | 1 / 2 = Command Error | | | |
| | | | 2 / 4 = Curve Done | | | |
| | | | 3 / 8 = Unused | | | |
| | | | 4 / 16 = Overload | | | |
| | | | 5 / 32 = Sweep Done | | | |
| | | | 6 / 64 = Service Requested | | | |
| | | | 7 / 128 = Output Ready | | | |
| | | | For each bit, 0 = NO, 1 = YES | | | |
| STIR | n | S/R | Stir to 305 via 303A. | 0 | 0...1 | 52 |
| SUB | n1 | A | "n1" Specifies subtrahend curve. | na | 0...5 | 35 |
| | n2 | | "n2" Specifies minuend curve. | na | 0...5 | |
| | | | **Note:** Command causes point-by-point | | | |
| | | | subtraction of subtrahend curve from | | | |
| | | | minuend curve with result stored | | | |
| | | | in minued curve. | | | |
| SUPDAC | n | S/R | Set/Read value in /96 suppression DAC. | 0 | -8190...+8190 | 11 |
| SWPS | n | S/R | Set/Read Number of Sweeps. | 1 | 1...65535 | 25 |
| TC | | A | Starts Curve Acquisition. | na | na | 21 |
| TMB | n | S/R | Set/Read Timebase. | 4000 | 50...50000 | 23 |
| TP | n1 | AR | Point # read and reported. | na | 0...6143 | 32 |
| | n2 | | Read and report I data value. | na | -2000...2000 | |
| | n3 | | Read and report E data value. | na | -10000...10000 | |
| TRIG | n | A | Puts pulse on TRIG OUT line. | na | 0...1 | 50 |
| TYPE | nnnn...n" | | A Prints msg. on LCD display. | na | ASCII | 50 |
| USR1, 2,3,4 | | | C Allows user-defined commands. | na | na | 47 |
| VER | n | R | Read software version. | | | 46 |
| VERTEX | n1 | S | Sets vertex point location. | 999 | 1...6143 | 18 |
| | n2 | | Sets modulation at vertex. | 8000 | -8000...8000 | |
| | | | (as many as fifty vertices may be specified) | | | |
| WAIT | n | A | Halts Curve Acq. for "n" tmb. | na | 0...65535 | 22 |
| WCD | | A | Postpones next cmd. in sequence | | | |
| | | | until curve acqui. is complete. | na | na | 22 |
| WFT | n | A | Wait For Trigger. | na | 0...1 | 50 |