GENERAL DESCRIPTION OF FUNCTION

The RS-232-C communications option for the Inficon IC 6000 thin film deposition controller is a hardware/software addition to the normal capabilities of the instrument and allows complete external control of the instrument. Any and all functional capabilities normally available with the standard IC 6000 being operated from the front panel may be invoked by external control through this option. When the external control is that of a controller or computer, the combination of IC 6000 and controller makes for a very powerful and flexible system. If the external control is not a 'smart' device, powerful features of remote control, remote program loading, and data logging are still available.

In order to best use the features of this option, it is necessary to understand certain concepts and terms. The IC 6000 instrument contains a microprocessor which, under program (software) control, performs all of the tasks necessary to make the instrument behave as it does. Under normal circumstances, when the communications option (hereafter called RS-232) is not installed, all primary communication between user and machine are through the front panel/display screen facilities. The front panel keyboard and the display screen CRT (cathode ray tube, or TV) allow the operator to examine data and parameter entry values and to control the behavior of the machine. These abilities are controlled by the software program in the instrument and the features and formats are fixed. Because of this, all interactions with the instrument are very rigid and structured. In order to allow a more flexible and custom approach, the RS-232 interface was developed.

When the RS-232 option is installed, the instrument is then capable of receiving and transmitting data and commands. All exchanges are in the form of MESSAGES. Each message consists of a string or sequence of alphanumeric (alphabetical and numerical and certain special symbols) characters transmitted one after the other either to or from the RS-232 interface. The particular grammar or rules of these messages are termed the syntax or format and are rigidly defined. In order for the instrument to precisely understand these messages, and for the operator or controller attached to the interface to generate and understand those sent by the instrument, a proper definition of each message must be defined. This is very straight forward and uncomplicated. The IC 6000 RS-232 interface was designed to be both fast and concise as well as mnemonic and simple. It is through these messages, then, that an interchange of data and commands may take place.

INSTALLATION INSTRUCTIONS

The RS-232 option for the IC 6000 deposition controller consists of additional software and hardware components to be added to a standard IC 6000 instrument. The option kit contains one (1) RS-232 Interface Module (IPN 205-460), two (2) EPROMs (IPNs 801-027 and 801-028), one (1) RS-232 self-test connector, one (1) mating connector kit, and a RS-232 manual. IC 6000 instruments having serial numbers of 19-xxxx series or higher may have the option field installed at any time.

NOTE: IC 6000 instruments which precede serial number series 19-xxxx are required to be returned to the factory for modification prior to installation of the RS-232 option.

Hardware Considerations

There are three essential hardware elements to the installation of the RS-232 option. Two of these, the RS-232 Interface Module and the EPROMs, are included with the option kit. The third element is the Video Board Module already present inside the instrument (see Fig. 8-1). Instruments capable of accepting the option have a Video Board Module containing two sockets to accept the EPROMs. If your instrument does not have such a board, it is of an early series and is not capable of having the RS-232 option field installed (see NOTE above).

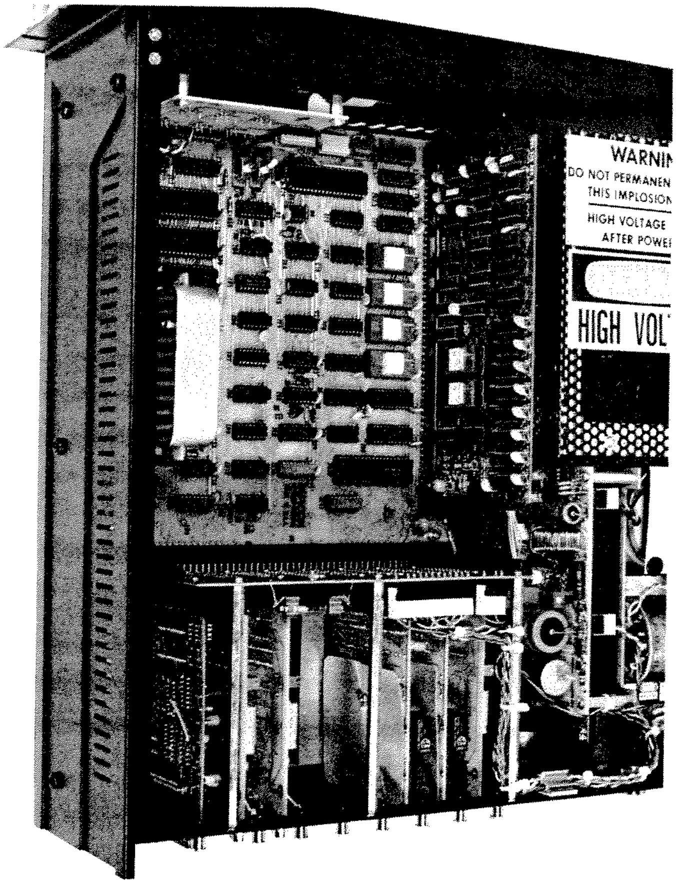


Fig. 8-1 IC 6000 Interior View

Installation of EPROM Memories

In order to install the EPROM Memories containing the option software into the IC 6000 instrument, the top access cover must be removed. This may be done when the instrument is originally configured, since the cover must be removed to gain access to the toggle switch that enables I/O programming.

CAUTION: Whenever inserting or removing components of the IC 6000 be sure that the instrument is in a power off condition. It is preferable that the detachable power cord be removed to insure against unintentional application of power. Observe the proper handling procedures for static sensitive parts and store in a safe location.

After removal of the top access cover, locate the Video Board Module (see Fig. 8-2) and remove it from its card guide/connector assembly. Holding the module so the component side is facing you and with the gold plated fingers along the bottom edge of the board facing down, locate the two (2) 24 pin sockets. If these sockets are not empty, they contain software in EPROMs for a previously installed option. Be certain that the new software being installed supercedes the existing software and that no loss of capability or operation will occur when the existing EPROMs are removed. If present, carefully remove the EPROMs from their sockets using even pressure so as not to angle the device and bend any pins.

The left-most socket is designated Z-21, and the right-most socket is designated Z-22. Also notice the chamfer or diagonal bevel present at the upper left-hand corner of each socket. This is used to indicate the orientation of the EPROM memory integrated circuit (IC) when inserted into the socket. Each of the two EPROMs also has a marking in its upper left-hand corner to indicate pin 1 of the IC. When viewing the EPROM memory with pin 1 in an upper left-hand corner position, the label covering the IC window should be lettered right side up. Insert the EPROM labeled as 801-027 into the socket designated as Z-21, applying even pressure and making sure that all pins enter the socket holes. If any resistance is encountered while trying to seat the IC into the socket, carefully remove the IC and observe the pins for signs of bending or misalignment. Following this procedure, insert the EPROM labeled 801-028 into the socket designated Z-22. When this is done, the Video Board Module may be reinserted into its card guide/connector assembly. This concludes the internal phase of the RS-232 installation. Following installation of the option, diagnostic tests will determine whether the ICs have been properly installed or need to be examined and reinstalled.

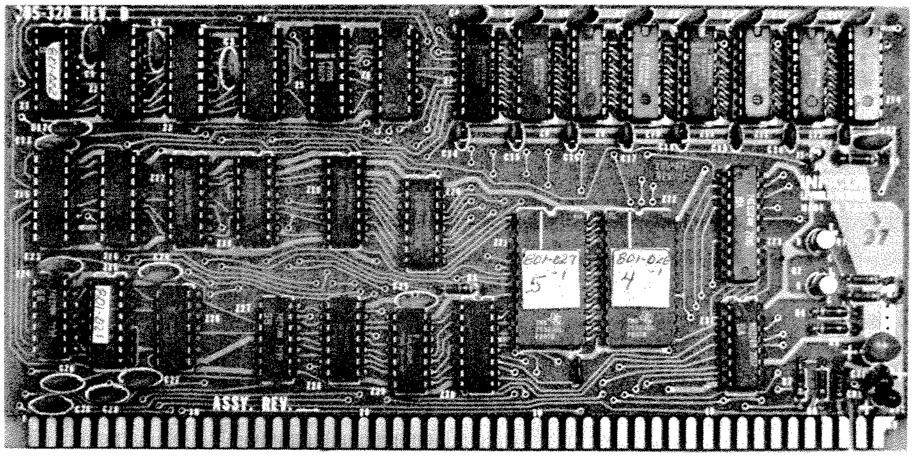


Fig. 8-2 Video Board Component View

Setup and Installation of RS-232 Interface Module

The RS-232 Interface Module is the circuit card containing the active circuitry and interface connector which links a RS-232 device to the main chassis of the IC 6000. Before installing the module the proper user baud rate should be selected. The choice of operating baud rate is normally predetermined either by the peripheral device being connected to the instrument or by the capacity of the communications channel linking the two devices. Greater baud rates will allow for faster transfer of data and more rapid control of the instrument. It is essential that the baud rate of the Interface Module and that of the peripheral device be identical. Most problems encountered during initial setup can be attributed to a failure to meet this requirement.

Selecting User Baud Rate

The user baud rate of the IC 6000 RS-232 option can be selected as one of eight standard speeds. They range from the slowest (150 baud) in steps that double in speed, up to the fastest (19200 baud). Slower speeds are normally compatible with hardcopy terminals and printers. Faster speeds are generally used for direct computer connection. The nonstandard baud rate and data format of teletype operation is not supported by this option, and teletype machines will be unable to interface to this option. To select the desired operating baud rate, an eight element subminiature slide switch has been provided. This is located on the interface module, near the 25 pin connector (see Fig. 8-3). The activation of one and only one switch selects each baud rate. All of the other switches should be off. The state of a given switch is determined by the location of the tip protruding from each switch element. When positioned closest to the RS-232 interface connector, a switch is off. All but one of the switches should be in this state. That switch which is aligned with the number for the desired baud rate is set to the active state. In this manner the photo indicates that the RS-232 module has been set for 300 baud. Once the baud rate has been selected, no further adjustments or settings of the module are needed.

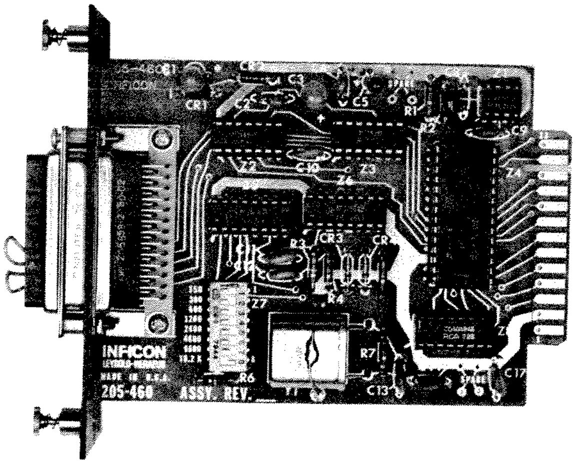


Fig. 8-3 RS-232 Interface Module

Installation of the RS-232 Interface Module

After proper setup of the user baud rate, the circuit module is installed in the rear of the IC 6000 chassis (see Fig. 8-4). A specific location reserved for computer interfaces (labeled IEEE) accepts the interface card. Instruments without the option will have a blank panel plate installed over this slot which is at the left-most edge of the rear panel when viewed from the rear. Remove the blank plate if present to expose the card guide and edge connector to accept the option module. Insert the module gold fingers first so that the notched keyway between fingers 11 and 12 is near the top and will align with a corresponding key in the mating edge connector socket. The labeling on the panel plate under the 25 pin interface connector should be correct side up and readable. Components located on the card should be on the right-hand side of the board, and should clear the left edge of any source/ sensor module installed in the adjacent slot. Slide the card inward along the guide groove of the bottom card guide while keeping the circuit card oriented parallel with respect to the plane of the side of the instrument. If additional access is needed to assure proper alignment of the circuit card, the top access cover may be removed or unfastened. After the circuit module has been properly installed into the mating edge connector, secure the module by tightening the threaded fasteners attached to the panel plate into the instrument's rear chassis rack.

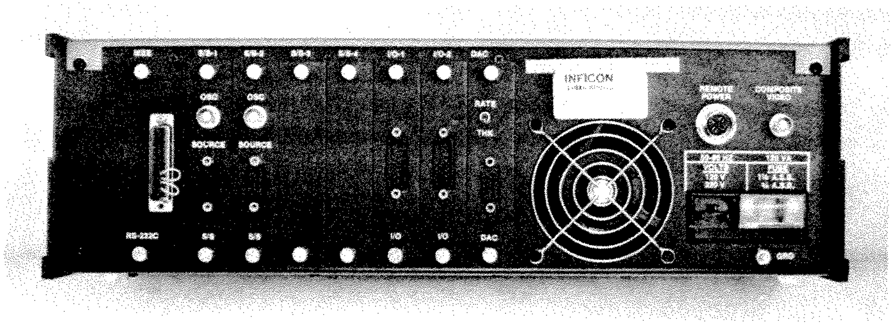


Fig. 8-4 IC 6000 Rear Panel

Preliminary Checkout

Following the insertion of the EPROM memories and the INTERFACE module according to the instructions given above, a self test should be performed to determine that the integrity of the IC 6000 instrument as a whole has not been adversely affected by the installation procedure. A necessary accessory used to perform sections of this checkout is the self-test or loop back connector received with the option kit. This connector is a mating 25 pin male connector which has had pins and jumper wires installed so as to loop back connector pin 2 to pin 3 and pin 4 back to pin 5 (see Fig. 8-5). This connector provides the necessary signals to simulate a peripheral device being attached to the interface, as well as allowing for an "echo" check of transmitted data. The use of this connector during self-test operation assures that the circuitry responsible for transmitting, receiving, and synchronizing of data signals is operating. This is not, however, a complete check of interface operating functions. Complete test of all operating functions requires an operational terminal device or computer attached to the interface. The self test mode of operation is performed only once each time the instrument is powered up. Exact operating behavior under the various instrument conditions is described in the next section.

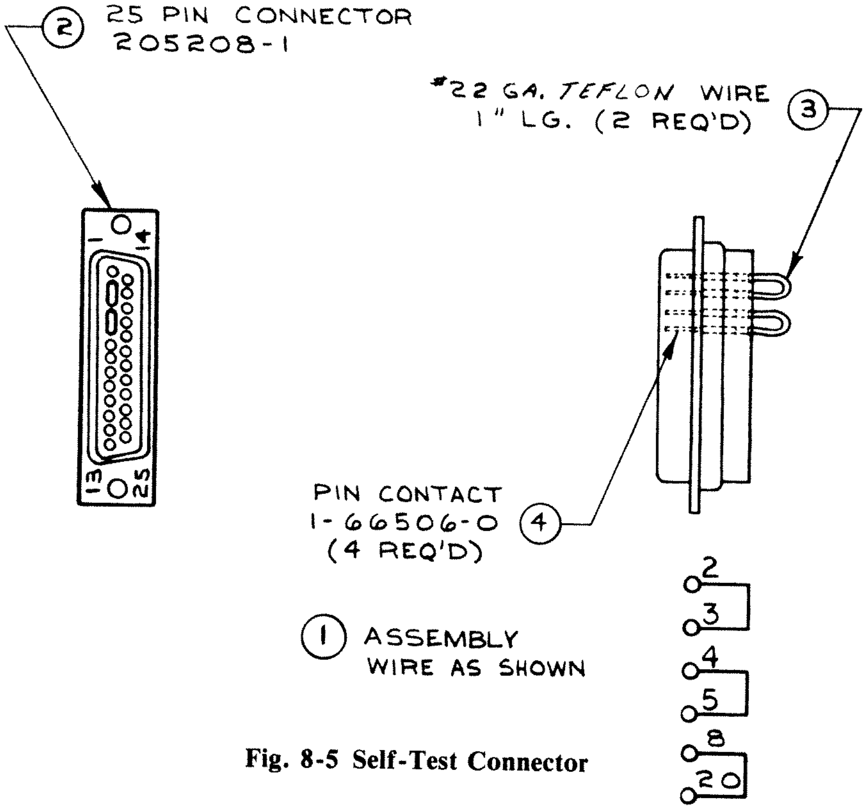


Fig. 8-5 Self-Test Connector

INITIAL OPERATIONS OF RS-232 INTERFACE MODULE

When the IC 6000 is turned on a confidence check is performed, as explained in the main section of the manual. Failure of elements of this check result in the display of certain 'bar' patterns on the screen. This feature is extended to the RS-232 option. If the option software is installed, it will be tested for validity, in much the same way as the main software is. If the software is invalid, a bar pattern (see Fig. 8-6) will be displayed. If the test of the option ROMs is successful then a test for compatibility between the RS-232 option software and the main software will be made. Non-compatibility will result in a failure to 'link' the option into the system, and IC 6000 will continue operating as normally. If compatible, the presence of the interface option module is checked. If absent, no 'link up' will occur.

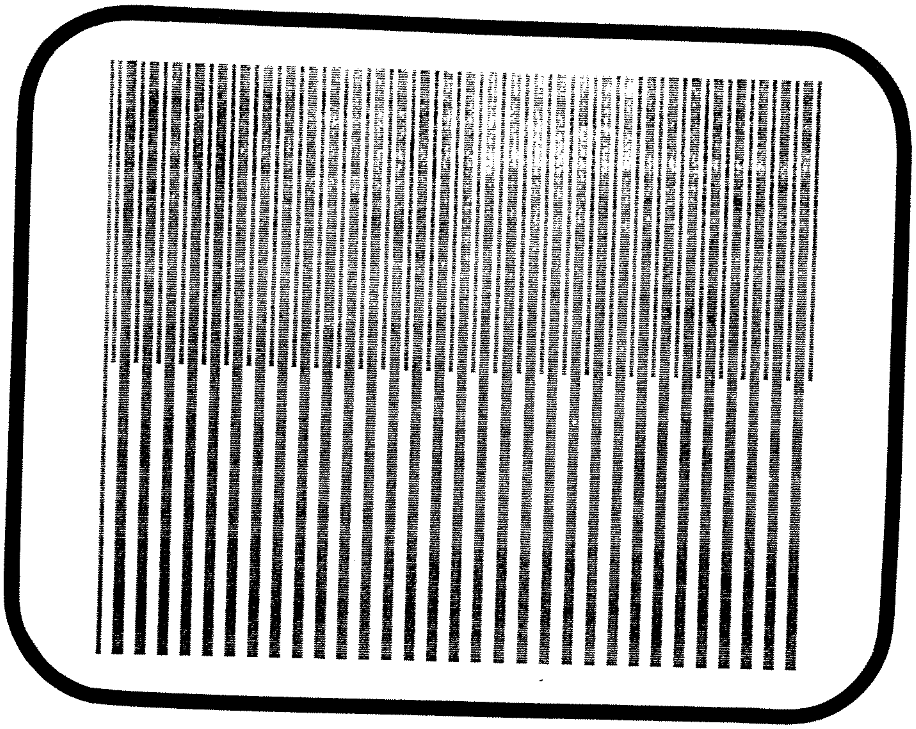
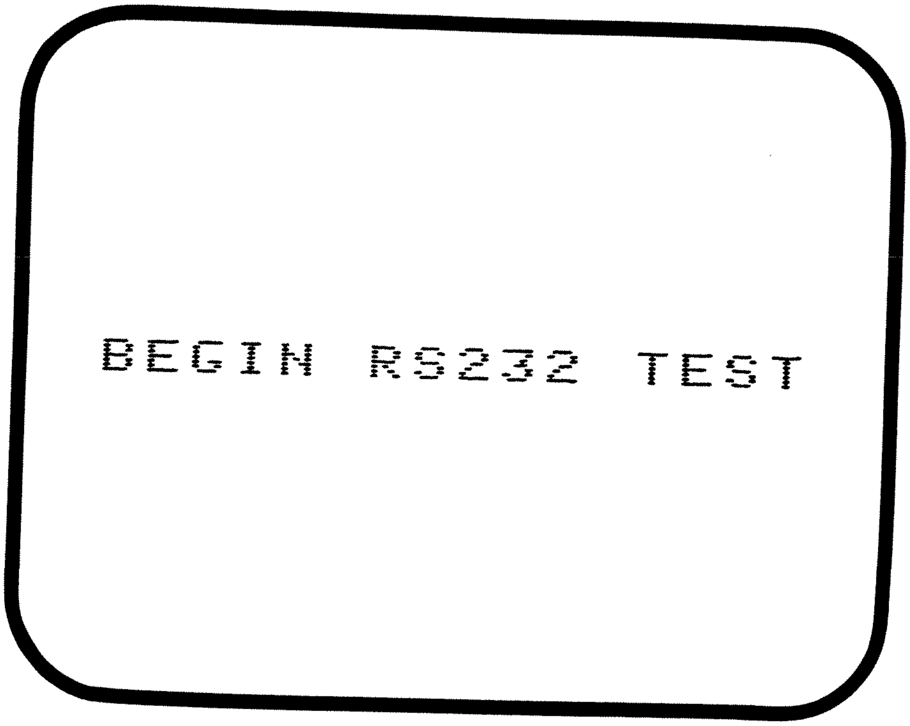


Fig. 8-6 RS-232 ROM Self-Test Failure

Under normal (initial) RS-232 interface operating conditions, the instrument transmits a sign-on messsage to indicate that the interface option hardware and software is installed and functional. At the start of this event the video display of the IC 6000 is cleared and a **BEGIN RS232 TEST** legend (see Fig. 8-7) is displayed. If the instrument successfully completes the RS-232

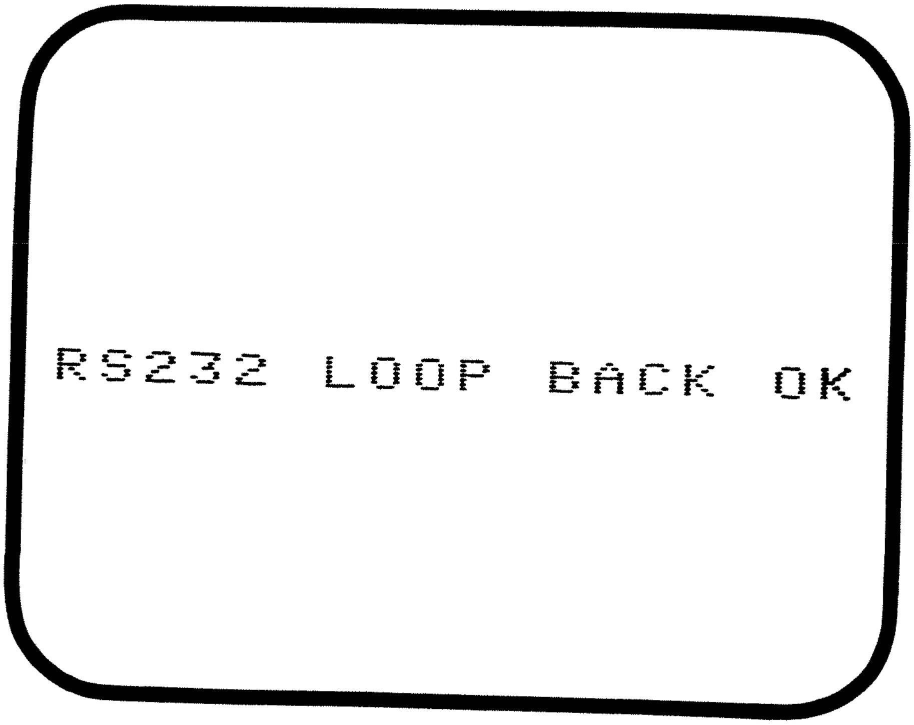


BEGIN RS232 TEST

Fig. 8-7 RS-232 Test Message - Start

check and a fast baud rate is selected, this message may be on the screen for only a few tenths of a second. If the IC 6000 is unable to successfully transmit its sign on message the instrument will remain in this 'hung up' condition and will not continue with its operation, rendering the IC 6000 inoperative. The indication of this problem is the continuous display of the **BEGIN RS232 TEST** legend for longer than ten (10) seconds. If this condition is encountered the IC 6000 may be put into service either by locating and correcting the cause of the interface failure, or by removing the RS-232 interface option module from the rear of the instrument (remember to ALWAYS disconnect power to the instrument when inserting or removing circuit components and/or modules). The most likely cause of this type of hang up condition is missing or improper connections to the interface connector or a failure of the peripheral device to control the handshake lines of the RS-232 interface properly.

During this initial transmission, the instrument monitors or looks for the same data it's sending to be looped back to its input. This is a form of self-test and allows for verification of the majority of the RS-232 interface hardware. If there is a logical continuity between data output and input then a **RS232 LOOP BACK OK** message (see Fig. 8-8) will be displayed on the IC 6000 CRT display screen. If no loop back takes place, normal instrument operation continues without a message being displayed. The final portion of the interface sign on message is a special message called a prompt. This message indicates to the peripheral or user that the instrument has completed any tasks and is waiting for user input. Any time the last message sent by the IC 6000 is a prompt message, the user is informed that the IC 6000 is waiting for further instructions. The form of the prompt message in the IC 6000 RS-232 interface is **>OK↑M↑J** where the printable characters sent are the **>OK.** Following this message the interface remains quiet until commanded by the peripheral or until the automatic data log mode is triggered. At this point all exchanges over the interface are initiated by messages sent from the peripheral.



RS232 LOOP BACK OK

Fig. 8-8 RS-232 Test Message - Complete

Default Conditions

When the IC 6000 with an RS-232 interface option is initially powered up, it defaults to a "stand alone" mode. This "stand alone" mode consists of the following states of configurable modes.

OPTION	DEFAULT STATE	MEANING
Echo Mode	EML condition	Echo Mode Long response format
Parity Mode	ODD and insensitive	Characters sent by the IC 6000 will have odd parity. Received characters will not be checked for parity.
Options Selected	OPT 6	RUN FINISH Group 1
	OPT 7	RUN FINISH Group 2
	OPT 8	RUN FINISH Layer Time
	OPT 9	RUN FINISH Thickness
	OPT 12	STOP ALERT
	OPT 13	ABORT ALERT
	OPT 14	END ALERT
	OPT 15	MAX POWER ALERT
	OPT 16	TIME POWER ALERT
	OPT 17	CRYSTAL FAILURE ALERT

BASIC SERIAL COMMUNICATIONS CONCEPTS

The IC 6000 instrument communicates logically with the operator through messages. These message strings are communicated physically through electrical connections. The description of this method is according to a standard convention for interconnecting computer-like data devices, and is called EIA RS-232-C. This standard defines the number and types of electrical signals and their voltages and meanings, as well as the particular mechanical (plug-socket) relationship for these signals. A technical description of these definitions is shown in Fig. 8-9. One physical parameter related to this standard but not strictly defined is a non-mechanical, non-electrical variable. This variable is time. The signals sent by the RS-232 interface are sent according to a strict time format, and each signal element occurs for a fixed and definite period of time. Since a given character requires a fixed quantity of signal elements (as defined by the standard), the repetition rate or 'speed' of signalling varies according to the duration of each signal element. Each signal element represents one binary digit (BIT) of 1 or 0. The number of bits per second that can be sent is a measure of the signalling rate and is called the BAUD rate. If the BAUD rate for a given interface is 300 baud, then 300 signalling elements per second may be sent over the interface. With a standard RS-232 format this works out to 30 characters per second. Several standard speeds or BAUD rates are defined and are available for use by the IC 6000 RS-232 interface. The one which is compatible with the communicator or controller attached to the interface is selected. Speeds ranging from 150 baud to 19200 baud are available to accomodate a wide range of peripheral types. Generally, selecting the greatest speed available with the peripheral will result in the greatest utility.

IC6000 is DTE

RS-232-C INTERFACE CONNECTOR.

SIGNAL	PIN #	DIRECTION	
TRANSMIT DATA	2	FROM IC6ØØØ	
RECEIVED DATA	3	TO IC6ØØØ	
REQUEST TO SEND	4	FROM IC6ØØØ	HIGH WHEN ABLE TO RECEIVE.
CLEAR TO SEND	5	TO IC6ØØØ	HIGH WHEN IC6ØØØ ENABLED TO TRANSMIT.
DATA SET READY	6	TO IC6ØØØ	NOT USED.
COMMON GROUND	7	COMMON	
TX, RX BAUD CLOCK	13	FROM IC6ØØØ	BAUD RATE X16 CLOCK SIGNAL.
DATA TERM. READY	2Ø	FROM IC6ØØØ	HIGH LEVEL WHEN IC6ØØØ ON.

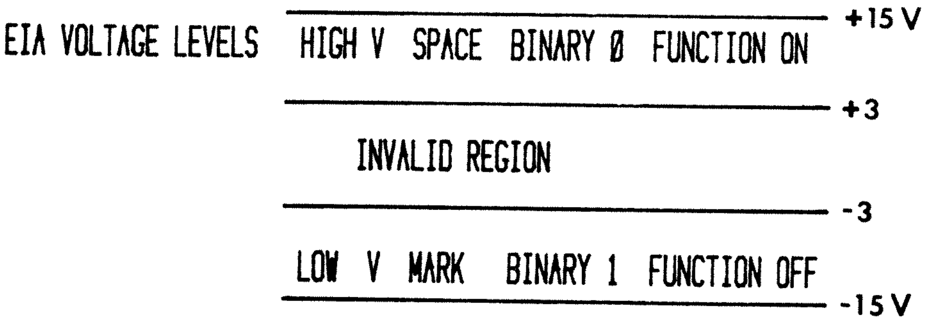

EIA VOLTAGE LEVELS
HIGH V SPACE BINARY Ø FUNCTION ON
+15 V
+3
INVALID REGION
-3
LOW V MARK BINARY 1 FUNCTION OFF
-15 V

Fig. 8-9 RS-232 Signal Definition

DATA REPRESENTATION CONVENTIONS

All messages communicated by the RS-232 interface are constructed of sequences or strings of characters. The coding of each character is according to the ASCII code for alphanumeric data. In addition to the standard upper and lower case alphabetical, numerical and punctuation characters, certain non-printing control codes are defined. To symbolize these codes, the up arrow or '^' symbol preceding an upper case letter will be used. The RUB-OUT or DELete code is a special nonprinting control code which is not represented in this manner, and is symbolized by RUB. This symbol has a special editing application, and indicates a desire to remove or forget the previous character transmitted. In this way the character sequence "w" "x" "y" "RUB" "z" is identical to the sequence "w" "x" "z." The IC 6000 interprets n (9) special control codes to have unique meanings or uses as defined herein:

Character	Symbol	ASCII	Code	Direction and Meaning
Control C	^C	03H	ETX	To the IC 6000. Stop or abort present command line execution.
Control G	^G	07H	BEL	From the IC 6000. A special 'framing' character sent be fore and after a real time ORIGINAL message. For more detailed explanation see Original Message Formats and Timing.
Control J	^J	0AH	LF	From the IC 6000. A printed control character sent following a CR or M character to cause the peripheral device to advance a line.
Control M	^M	0DH	CR	To and/or from the IC 6000. A control character used to signify the end of a COMMAND message sent to the IC 6000 (terminator) or, when sent to the peripheral, used to denote end of a line.
Control Q	^Q	11H	DC1	To the IC 6000. A control character causing the IC 6000 to resume transmission of data to the peripheral. For more detailed explanation see Handshaking.
Control R	^R	12H	DC2	To the IC 6000. A control character causing the IC 6000 to review or repeat the previous characters sent as part of a COMMAND message back to the peripheral. This allows an operator or computer to determine what the effective COMMAND line will be interpreted as. For more detailed explanation see Editing.
Control S	^S	13H	DC3	To the IC 6000. A control character causing the IC 6000 to suspend transmission of data to the peripheral. Useful to prevent data from being lost. For more detailed explanation see Handshaking.

Control X	^X	18H	CAN	To the IC 6000. A control character causing the IC 6000 to ignore or erase any previous CONTROL message characters sent by the peripheral. It effectively cancels the characters sent to the IC 6000 since the last termination (CR) character. For more detailed explanation see Editing.
RUBOUT	^RUB	7FH	Rubout	To the IC 6000. A control character causing the previous character sent to the IC 6000 to be ignored or forgotten. For more detailed explanation see Editing.

HARDWARE INTERFACING FUNDAMENTALS

The RS-232 Interface Option for the IC 6000 instrument consists of both software and hardware definitions and designations. Those elements of the interface pertaining to hardware are defined so as to insure compatibility between the IC 6000 instrument and the user's peripheral device. The communications standard defined by EIA RS-232 defines the mechanical elements of the interface (such as electrical connector types and pin designations), the electrical elements of the interface (such as voltage levels, current drive capability, bandwidth, etc.), and the logical elements of the data and handshake signals. Adherence to the definitions of the hardware interface as defined in this manual will insure the proper and orderly transfer of data between the IC 6000 and the user. The definition of the hardware conventions is seperate and distinct from the definition of the contents of the messages and their meanings. The hardware interface is said to be transparent with regard to the content of the messages.

The communications interface as defined by RS-232 is designed as an information interconnect between two (2) devices. The RS-232 interface is not therefore a bus, since a bus implies the multiple connection of many devices in parallel. The RS-232 interface can only be used to interconnect the IC 6000 instrument to another single peripheral device. This device may be a simpler terminal or printer, or it may be a computer mainframe. The IC 6000 instrument may appear to be connected to many peripherals through the operation of the connected computer, but the interface cable from the IC 6000 physically connects to only one peripheral. When defined, the RS-232 standard was intended as an interface between a Data Terminal Equipment (DTE) device such as a teleprinter, banking transaction terminal, etc. and a Data Communications Equipment (DCE) device such as a modem or similar signalling equipment for transmission of data over a telephone or similar carrier.

Each type of device (DTE or DCE) has particular needs for support of its operation, as well as particular inter-relationships between itself and the companion device attached. The RS-232 standard defines the existence, use and meaning for the signals that interconnect a DTE device to a DCE device. The direction of 'travel' (to or from a DTE or DCE device) as well as the use and meaning of any given signal is defined so that complementary signals are interconnected between a DTE and a DCE device with a straight forward one-to-one cable arrangement. In this way the male 25 pin cable of a DTE device will plug directly into the female 25 pin socket of a DCE device. This standard definition of a hardware interface allows for compatible inter-connection between equipment from many different vendors.

Certain variations on this definition of the standard, however, tend to cause confusion and improper interconnection of DTE and DCE devices. Because of individual packaging requirements, some DTE devices are not equipped with a male 25 pin plug attached to a cable, instead allowing the user to chose a cable length to suit the application. These DTE devices use a female chassis socket as a termination for the interface cable. Although the sex of the connector is female, it is important to remember that the device's function (as a DTE) determines the direction and use of the signals present on the connector. Because of this 'non-standard' mixing of device functions and connector sex, the user must be aware of the device function and connector arrangement of the two devices to be interconnected with a RS-232 interface. With regard to this distinction, the IC 6000 RS-232 interface is defined as a DTE or Data Terminal Equipment device.

Since the IC 6000 RS-232 interface defines the instrument as a DTE device, it is possible to directly connect the instrument through a simple one-to-one cable only to a suitable DCE device connector. Most computers and terminals and printers are also DTE devices, and the interconnection of the IC 6000 to one of these requires that each be connected to suitable DCE devices (such as modems) and that these communications devices be properly interconnected. In a direct wire or direct connect application, a special adapter cable to take the place of the two modems must be used. This cable is often referred to as a dummy modem or null modem as it serves to translate the interconnection of the interface signals between two DTE devices without actually performing a data modulation/demodulation function. More detailed definitions of particular interface signals can be found in the EIA RS-232C standards publication. With the exception of the use of the Request to Send signal which will be described in the section on Handshaking to follow, the use of the interface lines by the IC 6000 instrument conform to the standard.

Communications between the instrument and the peripheral device is in a bit serial-character serial asynchronous ASCII representation. This means that the content of the message is sent a single bit element at a time until these elements compose a character. Messages consisting of multiple characters are sent an element at a time in sequence to compose a string of multiple characters. Asynchronous operation in this sense means that the sending and receiving elements of the interface are not linked or synchronized to each other with respect to the timing of the individual characters. Each ASCII coded character is sent as a unit or frame. The IC 6000 RS-232 interface uses an asynchronous format of one (1) start bit, seven data bits sent least significant bit first, followed by one (1) parity bit of either odd or even sense (software selectable), and one (1) stop bit. This framing format is a common standard and should be compatible with the majority of computer and data processing communications devices in use. This format is not, however, compatible with the TELETYPE communications format. Since each character frame contains ten (10) signal elements, the transmission speed in terms of characters per second is one-tenth that of the baud rate (which is signal element rate). A baud rate of 300 corresponds to a character rate of 30 c.p.s. Serial data is transmitted in each direction (to and from a device) over a serial channel. One independent channel is required for each direction. In its simplest form, these two channels take the form of a pair of conductors (wires) and a common signal ground between the two devices (the IC 6000 instrument and the peripheral).

Although some communications systems allow for different sending and receiving rates for a given device, the hardware configuration of the IC 6000 RS-232 interface fixes the transmit and receive baud rates at the same speed. In addition to the two serial data channels required for carrying the ASCII data to and from a device, there are several additional lines or signals defined for the RS-232 interface standard which are used for signalling certain interface conditions between the two interconnected devices. These signals were defined for initialization, maintainance, synchronization and termination of the communication channel between the DTE device and its companion DCE device, as well as the link between interconnected DCE devices through a carrier (such as telephony). The IC 6000 instrument uses certain of these to perform "handshaking" between itself and the interconnected device. The exact use and meaning for these signals will be defined in the following section on Handshaking.

Handshaking and Timing

Since the transmission of data using the IC 6000 RS-232 interface is asynchronous, neither device has any means of anticipating or knowing the status of the other without some additional communications between devices. The primary problem that arises in such a scheme relates to the accepting and processing of the transmitted data by the devices. At high baud rates a new character can be transmitted at a rate of two per millisecond. This would require that the IC 6000 be capable of responding to each new character in at least that short an interval. Because of the normal operations being performed by the instrument at all times, this is not possible. It is therefore necessary to adopt a means of acknowledging between devices when each one is ready to receive more data, as well as a means for "holding off" the other device to prevent it from transmitting. This is referred to as handshaking, and is an important element of a protocol.

Because of the variety of modes and operating environments for which this option was designed, several mutual and inter-reacting mechanisms exist for the acknowledgement and hold off data transmissions. Some of these are performed through the transmission of additional control data along the serial channels, and can be used in a three wire or modem type of link. These types of handshake arrangements are referred to as "soft" handshakes since they require no hardware complications above those needed to transmit the data. These are specified more fully in a discussion of the TERMINAL protocol mode. Those handshaking arrangements which do require additional hardware connections and considerations will now be defined.

The IC 6000 instrument has one primary hardware control over its transmitting of serial data. This signal line is referred to as CLEAR TO SEND (CTS) and is an input signal to a DTE device as defined by the standard. The IC 6000 will be unable to transmit serial data unless this input signal is valid or active (a EIA positive voltage). Whenever this signal goes to an inactive or false state (EIA negative voltage) the instrument will continue to transmit any character presently being transferred, and then will cease to transmit any more data until the line again becomes active. Because of the direct hardwired function of this signal, it is absolutely essential that this signal be properly terminated or controlled. Failure to properly interface this signal is the most common cause of interface malfunction.

In order for the IC 6000 instrument to control a similar CTS signal of the connected peripheral device (to allow for hold-off of it) a signal from the IC 6000 is used to indicate whether the instrument is able to accept data or not. The interface signal used to perform this operation is the REQUEST TO SEND (RTS) signal. The use of this signal by the IC 6000 does NOT comply with the definition of this signal by the RS-232 standard. The meaning of this signal as it is used by the IC 6000 instrument is to signal the attached peripheral that the IC 6000 instrument is ready to accept new data and that the instrument requesting the peripheral to send data if any. This is the direct opposite of standard meaning used to indicate that the DTE sending the RTS signal is requesting that it be allowed to send. Observance of this deviation is very critical.

PROTOCOLS

Before further defining the exact structure of the messages, there are certain conventions and operating behaviors which are independent of the content of the messages sent. Like the hardware considerations, these conventions are for the purpose of insuring orderly and meaningful transfer of signals between the two elements (the IC 6000 and the peripheral) of the interface. These conventions are referred to as the protocol of the RS-232 interface. Various protocols already exist in the communications domain, and the ones used by the IC 6000 RS-232 interface are similar to many of these.

Communication protocols are defined to allow for the synchronizing of signals sent and received over an interface. The IC 6000 supports two separate protocols, each designed with a particular need or operating environment in mind. The differences between each are to accomodate various hardware and software needs of the peripheral.

One protocol is basically a straight forward full duplex protocol. It was designed primarily as a protocol to support connection to a simple terminal device (teleprinter or CRT terminal) to be operated by a human operator. When used in this mode, certain support features assist the operator with normal situations that can be encountered. Human operation requires indication to the user of what characters were already sent, as well as the ability to recall or 'backspace' over errant characters sent. In this way, simple errors can be corrected during transmission before they are interpreted by the IC 6000, sparing the user from retyping the entire command line again free of error. Should an error be encountered, an error message will indicate to the user where the problem was encountered.

Terminal Protocol Mode

In this protocol, characters sent by the peripheral are received by the IC 6000 and echoed back to the peripheral. This is both to synchronize and to indicate to the peripheral that the character sent was received, as well as to allow the peripheral to perform error checking. If the peripheral sends the message "ABCDEF^M," the IC 6000 RS-232 interface will echo back each character as it is received, so that the message "ABCDEF^M^J" will be received at the peripheral as a result of its sending the earlier message. This message did not originate in the IC 6000 and contains no more information than the original message sent by the peripheral. The signals present on the interface, represented on a time line showing relative elapsed time, would look like the following:

Peripheral	A		B		C		D		E		F		^ M		
IC 6000		A		B		C		D		E		F		^ M	^ J

From this depiction it is evident that the carriage return code is not echoed as sent, but is modified or interpreted. The manner by and purpose for which this interpretation is done is based on the assumed operating environment and needs of the devices on the interface. Since this interface was designed to allow for the connection of simple, 'dumb,' devices the interpreting of the carriage return to a carriage return/line feed combination is to insure proper operation with such devices. The line feed would be superfluous to a computer device, which would have been capable of interpreting the echoed carriage return code alone. The act of displaying a line of text on a 'dumb' terminal requires two separate control actions, the carriage return (CR) and line feed (LF) codes are used for this purpose. Certain other characters have special purposes and do not echo back as received, and will be discussed separately.

Under normal operating conditions, the peripheral transmits a string of characters which make up a message one character at a time and then waits to receive each echoed character before transmitting the next. This insures that the IC 6000 has processed the character and is ready to receive the next one. If this requirement is ignored and a second character is sent before the IC 6000 has processed the first, an overrun error occurs. When this happens the IC 6000 echoes neither the first character which was overrun and lost, nor the last character sent, but echoes the question mark '?' character. This indicates to the peripheral to send again the character which follows the last correctly received and echoed character. A time line representation of this exchange would look like the following:

Peripheral	A		B		C D		C		D		E F G		E		F		G	
IC 6000		A		B		?		C		D		?		E		F		G

By this example, it can be seen that the peripheral sent the characters "A" and "B" and they were properly received and echoed by the IC 6000. The peripheral then sent the characters "C" and "D" without waiting for the acknowledgement echo from the IC 6000. This resulted in an overrun error indicated by the "?" being sent by the IC 6000. This tells the peripheral that neither the "C" or "D" was properly received, so it must send them again. The peripheral then transmits the "E" "F" "G" string without waiting for an acknowledgement echo. This results in another "?" being sent by the IC 6000 to indicate the overrun error. This time it is necessary for the peripheral to go back 3 characters, until just after the last correctly received and echoed character, and repeat them. To avoid going thorough this awkward procedure, it is recommended that the peripheral comply with the necessary "wait for acknowledgement echo" protocol.

Since in the previous protocol mode the only exchange of signals between the-IC 6000 and the peripheral took place on the two serial data lines which can be transmitted with a minimum of wires or with a modem, this mode is very suitable for operating where a minimum of signals or wires is available or desirable. The repeating of data sent to the IC 6000 back to the peripheral, necessary for proper operation of a simple displaying terminal, allows for complete error detection. If while in this mode a computer were serving as the peripheral device, it could compare the received (echoed) character against the one sent and verify that it was sent and received properly. This type of protocol has its drawbacks however. If the peripheral device is a computer or similar automatic machine, it can be capable of generating characters at a rate faster than the IC 6000 can process them. Although it is unlikely a human operator can type characters fast enough to overload the IC 6000, a computer device could. As mentioned earlier, this would cause an overrun error resulting in the echoing of a '?' by the IC 6000.

This mode, when used by a high speed device, therefore requires a constant checking for the echo of each character as it is sent. This is both a software overhead as well as a waste of communications bandwidth. Additionally, some computer's communications software does not allow for the receiving of separate characters one at a time, and this technique would not be suitable. Also, under certain circumstances, the IC 6000 may produce data faster than the peripheral is willing to accept it. When using the 'TERMIMAL' mode of protocol, a special character code is sent to tell the IC 6000 to temporarily stop transmission of data. When the peripheral is again willing to receive data, another control character is sent to the IC 6000, and it will resume transmission of data. This is referred to as the XON/XOFF (device on, device off) protocol. In this manner a complete two way handshake is accomodated with only a wire or modem link.

Computer Protocol Mode

In situations where connection to a high speed peripheral device such as a computer is desired, a second protocol is defined that handles the problems of synchronization and data transfer differently. In this mode, the automatic repeating or echoing of characters sent to the IC 6000 is suppressed. Since a computer is sending the command messages to the instrument, it already knows what is being sent and doesn't need the echoed data as a reminder, as the human operator does. This relieves the peripheral computer from needing to deal with any messages coming from the IC 6000 other than complete messages containing data requested or real time data. Since the echoed data is lacking for determining whether the IC 6000 is ready to receive more data or if a transmission error has occurred, these conditions are detected separately.

In order to use the IC 6000 in the COMPUTER mode, additional hardware 'handshake' signals need to be controlled. Although these lines are always operative, they may not be needed in the TERMINAL mode if not desired, allowing a simple 3 wire (or modem) circuit. In the COMPUTER mode, these lines are the only method of preventing a data overrun condition as described earlier. To prevent the IC 6000 from sending data too fast for the peripheral, a second hardwired 'handshake' line can be used to hold off the IC 6000, much the same way as the XON/XOFF characters do. Using the COMPUTER mode does not disengage the XON/XOFF control over the IC 6000 nor does it cancel the question mark (?) response to overruns, should they occur (which is unlikely if the hardwired handshake is working properly). While operating in the COMPUTER mode the only method usable for transmission error detection is parity checking. This method is available when operating in the TERMINAL mode, and may be selected and used in place of or in addition to echo comparison error checking Parity checking is done by adding an additional bit, called a parity bit, to each character transmitted. The state of the bit (1 or 0) is determined based on the content of the character being transmitted and the rule or mode of paritesting being used. Unless configured to EVEN parity mode by the proper command, the IC 6000 transmits all data in the ODD parity mode of testing. In this mode the number or count of bits that are at a one (1) state in each character transmitted is made to be odd when sent by setting the state of the parity bit accordingly. When the data is received the bits at a one (1) state are counted, and if not odd in number, an error has occurred. The same method is used for the EVEN mode, and either one is equivalent in error detection capability. Two modes are supported to allow for compatibility with the peripheral device. In this manner, the peripheral can be sure that the information received from the IC 6000 is valid, and this capability is useful both TERMINAL and COMPUTER modes of protocol. In addition, when in COMPUTER mode, in the only way for the peripheral to be sure that the IC 6000 is properly receiving data is to enable parity error sensitivity in the IC 6000. Since the IC 6000 doesn't echo the commands sent it for error comparison by the peripheral, the peripheral must rely on the error detection ability of the IC 6000.

To use this mode, it is necessary to enable parity checking in the IC 6000 and to insure that the data being sent by the peripheral conforms to the parity convention established. This means setting the ODD/EVEN mode to match between the IC 6000 and the peripheral as well as activating parity sensitivity (normally insensitive on power up) in the instrument. Under this mode of operation, the IC 6000 will accept messages from the peripheral without echo but will transmit a dollar sign '$' character to the peripheral in the event of receipt of a character with invalid parity. The peripheral device can then elect to retransmit that character, or cancel the entire line and repeat it in its entirety.

REMOTE PROGRAMMING AND CONTROL

The IC 6000 is normally programmed through the integral front panel keyboard and display screen. Programming is accomplished by entering or storing numeric representations of the desired information or data in the appropriate locations on the display screen. This programming method is referred to as a menu or directed form of access. The user is given a list of element choices and through the use of a cursor function selects the item of interest. In this way the available choices of parameter types and their functional grouping can be easily represented. This is a very human oriented method well suited for an interactive, user prompting form of programming. One drawback to this method is the need for presenting a large quantity of information (many parameter lines) to the user any time even a single entry or change is to be made, in order to allow the user to select the item of interest. This also impedes the random accessability of the programmable parameters. The user must step through undesired elements to reach the desired one. Since the exchange of this menu and selection information must take place over a communication channel of limited capacity (in terms of speed) when performed by the RS-232 interface, a more direct access to the parameters to be programmed was developed. Access to any parameter is through a unique identification operation. Since sequential access to many parameters of a related group may still be desired, the ability to step forward to subsequent parameters without needing to uniquely identify them is also provided

In a manner similar to that used when programming the IC 6000 from the front panel, the parameters are grouped into functional areas of FILM, EXECUTIVE, and I/O (input/output) parameters. In addition to these parameters normally associated with the operation of the IC 6000, other parameters were defined for use with the RS-232 interface. Information normally presented to the operator via the screen display (rates, powers, active sources, layer timer, etc.) and certain operator alterable conditions (TEST mode, source powers, manual process film) have been given a parameter designation to allow the user to access the data associated with these variables. Some of these parameters are for readout or display only, and are referred to as 'STATUS' variables. Others, such as the manual process film number, are entry or programmable only, and are 'DATA' variables. TEST mode is both readable (as a status variable) and programmable (as a data variable). Parameters which are accessed by a numerical identifier are both readable and programmable.

In order to blend the random access nature of unique parameter identification with the 'handy' sequential access nature of the cursor choice method of parameter selection, the RS-232 interface uses an index or pointer which imitates the function of th screen cursor. This imaginary cursor is a software function and does not physically move nor is its indexing communicated to the user in the same way that the visual cursor is. This index is referred to as the CURRENT VARIABLE INDEX (C.V.I.) and it is used to select or point to the data or status variable of interest. Unlike the visual cursor displayed on the screen, whose position is altered sequentially forward and backward, the C.V.I. can be preset to point into variable storage at any point. The area or parameter selected by the index is derived from several components of the C.V.I. The primary component of the index is the mode. In the NUMERIC identifier mode of operation, the C.V.I. points to those parameters that have a numeric identifier. These are the usual parameters associated with the FILM, EXECUTIVE, and I/O menu pages of the IC 6000. These are further divided between FILM and NON-FILM parameters. The FILM parameters are uniquely identified by both their FILM ID and PARAMETER ID. The FILM ID is identical to the film number of the display index (1 - 6) for that film, and may be selected independently of the PARAMETER ID, upon which it has no affect. The PARAMETER ID is the same number as shown on a menu page on the screen for that film parameter, and may be of a value 1 - 37. By combining FILM and PARAMETER identifiers, any individual film parameter can be selected. For NON-FILM (executive and I/O) numeric identifier parameters, the PARAMETER IDs 38 - 60 are used. These numbers do not require a FILM ID as they uniquely select the parameter. The PARAMETER has no affect on the value of the FILM ID. previously set by the user. By individually setting the FILM and/or PARAMETER ID indexes, any numerically identified parameter may be selected. The act of setting either the FILM or PARAMETER ID, or both, sets the C.V.I. to the numerical identification mode, and the variable so selected is specified by the combination of the two ID types. For PARAMETER ID numbers between 36 and 60, the FILM ID is not used or altered. PARAMETER ID number and corresponding parameter name for ID numbers 1 - 60 are listed in Table 8-1.

When the sequential access form of operation is used to point to successive variables, the C.V.I. points to the next numerically ordered variable, with certain limitations. If the present C.V.I. selects a FILM type of numerically identified variable, it will index forward to the next parameter number within that film until pointing to the last parameter (ID 37 plot dwell). If this is not the last film (film number 6), it will then index forward to the first parameter of the next film number. When indexing forward past the last parameter of the last film, the C.V.I. will then point to the first parameter of the first film (film number 1). In this way sequential access through film parameters remains within the numeric range from 1 - 37 inclusive, and indexing to other films does not require unique film identification. This allows the complete programming of all six films automatically by computer or similar device without excessively repetitive identification operations.

When sequentially accessing non-film (executive or I/O) type variables, the C.V.I. points to the next variable (as determined by its ID number) until it points to the last non-film variable (ID number 60, INPUT 8). The C.V.I. will index no higher at this point, and any further incremental indexing commands will have no affect.

The other mode of designating the unique variable location desired is the SYMBOLIC or parameter name mode. This mode is reserved for those newly defined parameters or variables unique to the RS-232 interface. When a symbolic identifier is used, the C.V.I. points to the chosen variable area and cancels the meaning of the present value or state of the FILM and PARAMETER components of the C.V.I. designated previously. Incremental indexing used while in the symbolic mode of access will point to successive variable areas in order of listing of their corresponding symbolic identifiers in Table 8-1. This process will continue until the C.V.I points to the last symbolically referenced variable (crystal life status, symbolic ID XLif), at which time further incrementing commands will cease to affect the C.V.I. pointer, which remains selecting the crystal life status variable for inspection.

Whereas all numerically identified parameters are both status and data types and and may be inspected and modified, certain symbolically identified variables are classed as status variables and any attempt to perform a modification to these by use of the 'equals' (=) command will result in a command error. Also, certain symbolic identifiers reference variables associated with optional hardware elements (source powers to source sensor modules, relays and inputs of Input/Output modules) and explicitly or sequentially identifying these variables may result in a configuration error if the required modules are absent. Awareness of this condition when performing sequential access to symbolically identified variables is essential to error free operation.

MESSAGE GENERATION, EDITING, AND VERIFICATION

As mentioned previously, the RS-232 interface was designed to work in a number of application environments. When used as a computer interface, messages sent to the instrument will probably be constructed by a software program operating in the user's computer. Once this program is properly designed and operating, the messages being sent to the instrument will be correct and any user prompting or correcting will be performed by the program operating in the user's computer. In this mode the editing and verification operations incorporated into the RS-232 interface will probably not be needed. However, when the user is first familiarizing himself with the operation of the interface, a more interactive or conversational mode is more useful. This mode is referred to as the terminal mode and has features designed to support interactive use of the interface. It is in this mode that the message generation, editing, and verification functions are most useful.

Generation

When generating a command message to the instrument, a series or string of letters or characters are sent one after the other. It is the combinations of these character groups or words that defines the operation to be performed by the command. A command message is defined as a string of characters between one and eighty characters in length which is terminated by a carriage return (hereafter referred to as CR) character. It is not until the receipt of this final termination character that the transmitted command is interpreted and the desired operations are performed. Prior to the receipt of this "go ahead" character, a number of editing and verification operations may take place. The uses for these techniques become apparent during interactive operation of the instrument. In normal interactive or "terminal" use, a command message is generated or built up one character at a time. Each time a character other than a CR or other control character is sent to the instrument, the character is stored in proper sequence in the IC 6000 memory. If the interface is in the TERMINAL mode of operation each character will be repeated or echoed back to the peripheral device as both a form of acknowledgement and for prompting the user. In this way the display medium (either paper hard copy or screen) will reflect to the user the present sequence of characters received by the instrument. It is this sequence of characters which will then be interpreted and acted upon following the receipt of the CR terminator. Any errors in this command message during interpretation may result in either incorrect or inadvertent response by the instrument. If an error occurs, in some cases the entire command message may be ignored and require its transmission in its entirety. In order to avoid these consequences, several features have been provided for verification and modification (editing) of the command message characters following their having been received by the instrument. It is important to remember that at any time prior to receiving a CR terminator, any characters received by the instrument are only being held in a memory buffer, and do not have any influence upon the operation of the instrument.

Editing

Following receipt of one or more characters but preceding receipt of a CR character, the present contents of the command message is stored in a command buffer. Should an incorrect character be sent to the instrument (and subsequently echoed if in the TERMINAL mode) it can be cancelled and corrected by retransmitting the proper character in its place. This operation is similar to the backspace operation of a typewriter, and has the effect of moving backwards one character at a time through the buffer memory from the present end or last character sent. This operation is performed through the use of a control code or character commonly referred to as RUBOUT or DELETE. Each time a RUBOUT code is sent to the instrument the last character left in the buffer is deleted or forgotten and the new end of the message is the spot where the preceding non-deleted character is stored. In this way individual character errors may be cancelled and re-sent. If the errant character was not detected until after having been followed by one or more subsequent characters, it will be necessary to RUBOUT backwards over these characters until the erroneous character or characters have been canceled. To assist the user in determining how many characters have been RUBBED out, the use of the RUBOUT key causes the character which was deleted to be repeated or echoed to the user's output device. This method is used in order to be compatible with hard copy peripherals as well as CRT type terminals, since hardcopy terminals have no means to delete or erase previously printed characters. Use of the RUBOUT with an empty buffer results in a "#" being echoed. The deleted characters come back at the user in the reverse order sent in, resembling a mirror image. This effect will continue from the start of a series of RUBOUTs until this string is broken by sending the correction characters. It is apparent that determining the buffer contents, and hence what the instrument will interpret and act upon, is difficult when viewing the data presented to the user on the output display. It only gets worse following a succession of

RUBOUT/ correction operations. Should the errant character be located early (deep) in the command message, it might be preferable to abort or cancel the entire command and resend it from the beginning rather than individually cancelling and retyping most of the message again. This total deletion form of editing is accomplished by another control character. This character is the ASCII CTRL-X character. When this code is received by the instrument, the entire contents of the command buffer is cleared and a '#' character is echoed to indicate this.

Verification

From the preceding example of editing, it can be seen that the information displayed on the output device following several edit/retype operations is difficult to interpret. A special command is used to determine what the actual contents of the command buffer is so that correctness of the command message can be verified before the CR terminator is sent. This command is invoked through use of the CTRL-R ASCII control code. When this code is received by the instrument the actual contents of the command buffer is echoed or reviewed to the user's output device. In this way the effect of any preceding editing functions can be observed. An example of what would appear on the output device during a typical edit and verification operation is shown below.

OPERATION PERFORMED	OUTPUT DISPLAY	BUFFER CONTENTS
COMMAND TYPED	FILM 1 PARAM 2=1..23	FILM 1 PARAM 2=1..23
AFTER 1 RUBOUT	FILM 1 PARAM 2=1..233	FILM 1 PARAM 2=1..2
AFTER 2 RUBOUTS	FILM 1 PARAM 2=1..2332	FILM 1 PARAM 2=1..
AFTER 3 RUBOUTS	FILM 1 PARAM 2=1..2332.	FILM 1 PARAM 2=1.
AFTER CTRL-R	FILM 1 PARAM 2=1.	FILM 1 PARAM 2=1.
AFTER CORRECTIONS	FILM 1 PARAM 2=1.23	FILM 1 PARAM 2=1.23

Use of the verification command should be limited primarily to the TERMINAL mode of operation and should be used whenever clarification of the present contents of the command buffer is needed. Unnecessary use of the command will result in a lower interface throughput since all characters contained in the command buffer will be repeated to the user every time the command is issued. Use of this command in a computer interface environment should not be needed since the user's software operating in the host computer can review and verify the data to the operator without the need to interact with the instrument.

INTERFACE MESSAGE TYPES

One subset of all interface messages is referred to as the COMMAND subset. It is through the use of these commands that the user obtains control over the IC 6000 instrument. Command types are divided between CONTROL and EQUIVALENCE commands. The IC 6000 performs the desired action as requested (if possible) and does not generate any interface messages of its own other than the prompt message. In the DISPLAY type group of commands, a message from the the IC 6000 containing the data desired to be displayed is sent to the peripheral. Combinations of these commands may be sent to generate the desired response from the IC 6000.

In addition to the COMMAND subset of interface messages, there are messages sent by the IC 6000 to the peripheral. These message types are divided into RESPONSE, ERROR, and ORIGINAL categories. The ERROR message type is sent to the peripheral in response to a COMMAND message and is used to indicate to the peripheral that some problem was encountered while the IC 6000 was attempting to perform the tasks requested by the COMMAND message. It has a unique characteristic to identify it as being different than the normal RESPONSE type of message. The RESPONSE type of message is the normal type sent by the IC 6000 to the peripheral in answer to a DISPLAY type of COMMAND message. The format of the message depends on the particular information requested as well as the mode of response operation selected. The two modes of response are the LONG and SHORT modes. The short mode is most likely to be used in a system configuration where the peripheral device is a computer or other 'smart' device which requires only the data from the IC 6000. In this short mode, the requested data is sent to the peripheral without any additional identifiers or labels or titles. It is the responsibility for the peripheral to keep track of which field being received is for what data variable.

The long mode of response, on the other hand, adds additional identification fields to the data in the response message. This mode can also be used with a computer, but it is less likely since the additional overhead of the title information reduces the throughput of the interface and complicates the software in the peripheral, which now must be able to accept and interpret the meaning of these additional fields. The long mode is therefore most likely to be used with a simple 'dumb' terminal or printer tied into the IC 6000 interface, or during initial system design and debugging, when the additional identification fields are useful for clarifying what messages were sent.

The last type of message sent by the IC 6000 is the ORIGINAL message. As the name implies, the contents of such a message are original in the IC 6000 and are not in response to any COMMAND message sent from the peripheral. Such messages may be generated by the IC 6000 at certain times, and these occurrences do not coincide with any interface timing, condition or previous message sent. Because of this, it is referred to as an asynchronous message. The conditions which cause such messages to be sent will be explained later, but one cause already mentioned is the AUTO DATA LOG mode. In this mode, certain selected information is generated and sent by the IC 6000 to the peripheral on the occurrence of a certain TRIGGER condition. When this occurs, the RS-232 interface starts sending an ORIGINAL type message. Like the ERROR message type, it possesses a unique format characteristic which 'flags' its occurrence. In this way the peripheral device knows that the characters it is receiving are neither the echo of COMMAND characters previously sent nor an ERROR type of message, but rather specific data that the system has been configured to generate. The peripheral devices software must be designed to accomodate the occurrence of this asynchronous message type at any time, if such options are desired.

MESSAGE FORMATS

Command Words

Every message has a rule or syntax for its construction. The COMMAND messages sent from the peripheral to the IC 6000 have a syntax which allows for flexible and free styled structure. Since the IC 6000 has the ability to recognize and interpret a given command in many different formats, the format chosen by the system designer can be selected to best suit the needs of the system. The elements of flexibility were built in to allow trade offs between human user suitability and computer peripheral suitability. Whereas the computer peripheral can be programmed to use any character sequence to represent a given message (such as a number code), a human user would like to see English-like command words with names similar to their meanings. This concept is called mnemonics, or aiding to the memory.

The set of all COMMAND message types is a list of character sequences. Each command word consists of one or more characters. Certain commands also require numeric information fields to be complete. The list of COMMANDS is an alphabetical sequence. By starting with the first alphabetical (A-Z) character and through but not including the first following non-alphabetical character of a message string, a COMMAND label is recognized. This label is used to identify the COMMAND. A series of commands is operated on only when followed by the terminator (CR) code.

In addition to these alphabetically named COMMANDS there are several commands those label is a non-alphabetical string. These COMMANDS are, in order of their list ranking:

SINGLE QUOTE
'

Indicates start or end of a literal comment string. Any text within a COMMAND message that is enclosed by a pair of single quotes is not interpreted for content but is considered a comment field or sentence. In this way explanations of the meaning of commands can be included along with the COMMAND message. When the COMMAND message containing the comment field is terminated (by a CR), it is interpreted and executed. When the section of the COMMAND message containing the comment field is encountered, this comment is echoed or repeated to the peripheral device, as a comment of function and an indication of progress. In this way a log or hardcopy printout of interface exchanges will be properly documented.

COMMA
,

A DISPLAY type command that causes the display of current data, then increments the data index.

SEMI-COLON
;

A multifunction command that, depending on previous state of displayed or equivalenced data variable, will either display current variable, increment data index and display new current variable, or increment data index only.

EQUAL SIGN
=

An EQUIVALENCE type command that causes the current data variable to take on the new value indicated by the numeric data to follow. Valid values for this numeric argument depend on the type and status of the current data variable.

Those commands whose label is an alphabetical string can be recognized in several forms. Upper case characters are used to identify which characters are necessary to uniquely select the desired command. Any characters following in lower case may also be used for clarity but are not necessary.

The list of alphabetically labeled COMMAND names is as follows:

Ab A CONTROL type command that aborts the IC 6000.

ABR A CONTROL type command that performs an abort-reset.

AF A CONTROL type command that sets the current data index to the active film status variable.

AP A CONTROL type command that sets the current data index to the active process status variable.

AS A CONTROL type command that sets the current data index to the active source status variable.

AVP A CONTROL type command that sets the current data index to the average deposition power status variable.

AVR A CONTROL type command that sets the current data index to the average rate (over 1 second) status variable.

Clk An EQUIVALENCE type command that sets the time interval of the internal real time clock. Values 1-100 are valid to set the interval from .1 to 10 seconds in .1 second increments. A value of 0 deactivates the timing.

COmp A CONTROL type command that sets the RS-232 interface protocol into the computer (non-echo) mode.

CONt A CONTROL type command that requests a CONTINUE operation.

Eml A CONTROL type command that sets the display mode to ECHO MODE LONG (default mode).

EMS A CONTROL type command that sets the display mode to ECHO MODE SHORT.

EVen A CONTROL type command that sets the mode of the RS-232 interface to an even parity state.

Film An EQUIVALENCE type command that sets the program parameter film index to the numerical value that follows, and sets the current data index to the program parameter mode. Values from 1 to 6 are valid.

FP A CONTROL type command that sets the IC 6000 program security mode to locked and prevents unlocking from the IC 6000 front panel.

FR A CONTROL type command that requests a film-reset operation.

Gr A DISPLAY type command that requests the return of the rate deviation graph data as a response.

Icnd A CONTROL type command that sets the current data index to the input conditions status variable.

IN An EQUIVALENCE type command that sets the current data index to input condition status variable specified by the numerical value following. Valid values are 1-8 providing the corresponding I/O option module is installed.

La A CONTROL type command that sets the current data index to the front panel abort lamp status variable.

LCnd A CONTROL type command that sets the current data index to the front panel lamp conditions status variable.

LM A CONTROL type command which sets the current data index to the front panel manual lamp status variable.

LR A CONTROL type command that sets the current data index to the front panel ready lamp status variable.

LX A CONTROL type command that sets the current data index to the front panel crystal switch lamp status variable.

LYr A CONTROL type command that sets the current data index to the active layer status variable.

LYRT A CONTROL type of command that sets the current data index to the layer timer status variable.

Man An EQUIVALENCE type command that sets the IC 6000 into or out of manual mode (if in non-stopped operation) according to the numeric truth value following (1 means in manual, 0 means out of manual).

MF An EQUIVALENCE type command that, when operating in manual film select (process 4) and in a ready condition, sets the active film data variable to the numeric value following. Valid values are 1-6.

MP A CONTROL type command that sets the current data index to the 'at maximum power level' status variable.

Nf A CONTROL type command that sets the current data index to the pre-soak film number status variable.

NS A CONTROL type command that sets the current data index to the pre-soak source number status variable.

Odd A CONTROL type command that sets the RS-232 interface to operate in the odd parity state.

OPt An EQUIVALENCE type command that activates or deactivates the chosen interface option according to the numeric value that follows. For further description of the valid values and their meaning see the Special Operations and Optional Modes.

Param An EQUIVALENCE command, similar to FILM, that sets the program parameter index to the numeric value that follows and sets the current data index to the program parameter mode. For further description of valid values and their meaning see Remote Control and Programming Via Option.

PARIty An EQUIVALENCE type command that sets the RS-232 interface into either a parity error sensitive or insensitive state, depending on the truth value of the variable following. A value of 1 sets the parity sensitive mode, a value of 0 sets the parity insensitive mode.

PH A CONTROL type command that sets the current data index to the process phase status variable.

PHT A CONTORL type command that sets the current data index to the phase timer status variable.

POw An EQUIVALENCE type command that sets the current data index to the source power status or data variable corresponding to the numeric value following (1-4).

PRe An EQUIVALENCE type command that activates or deactivates the pre-soak enable by RS-232 remote according to the truth value of the argument following. A 1 value activates the pre-soak enable. A 0 value deactivates it.

PRS A CONTROL type command that requests a pre-soak start.

Q A CONTROL type command that sets the current data index to the quality error accumulator status variable.

Rate A CONTROL type command that sets the current data index to the instantaneous rate status variable.

RCnd A CONTROL type command that sets the current data index to the relay conditions status variable.

RD A CONTROL type command that sets the current data index to the rate deviation status variable.

RY An EQUIVALENCE type command that sets the current data index to the relay status or data variable corresponding to the numeric value following (1-8).

Sa A CONTROL type command that sets the current data index to the stability error accumulator status variable.

SPro An EQUIVALENCE type command that sets the requested new active process parameter variable to the numeric value that follows (1-4).

ST A CONTROL type command that requests a start operation.

STAt A CONTROL type command that sets the current data index to the 'machine annunciator status' status variable.

STOp A CONTROL type command that requests a stop operation.

Thick A CONTROL type command that sets the current data index to the measured thickness status variable.

TRg A CONTROL type command that requests a final thickness trigger operation.

TRM A CONTROL type command that sets the RS-232 interface protocol into the terminal or echo mode.

TSt A CONTROL type command that sets the current data index to the test mode data or status variable.

Xfl A CONTROL type comand that sets the current data index to the crystal failure status variable.

XInh An EQUIVALENCE type command that sets the IC 6000 either into or out of the peripheral crystal failure inhibit mode depending on the truth value of the numeric value that follows (1 = INHIBIT mode, 0 = NORMAL mode).

XLif A CONTROL type command that sets the current data index to the crystal life status variable.

XNum A CONTROL type command that sets the current data index to the measurement crystal sensor number status variable.

XSw A CONTROL type command that requests a crystal switch operation.

Zero A CONTROL type command that requests a zero thickness operation.

Command Formats

Messages sent to the IC 6000 instrument are in the form of commands. A command message consists of one or more command word combinations properly delimited from each other, ended with the CR terminator. Each command word combination consists of a string or sequence of the characters needed to select the desired command, as well as any numeric arguments that may be required. With the exception of the non-alphabetical commands (single quote, comma, semi-colon, and equals sign) a command word is defined as a word composed of a contiguous string of alphabetical characters in the command message. The occurrence of the 'space' character delimiter or of any numeric character or non-alphabetical commands delimits the sequence of characters which makes up the command word.

The command message is parsed or examined from left to right (from earliest received character towards the terminator) with the first character of the message assumed to be a command word character. This starts the parsing task, and the detection of further command words is based on the previously encountered command words. If the first character or group of characters of the message is not a valid command, then this will result in a command error. Following successful interpretation of a command word, subsequent message contents are examined for command validity. This process continues until the entire message has been examined. Any time the assumed condition of a command word being present is not verified, an error results. In this way multiple commands can be combined in a single command message line. After successful parsing of a command word combination occurring in a multiple command message, the instrument executes or performs the desired action requested by the command sent. After execution of the command, the instrument then continues to parse the command message for additional commands. Because of this piece-meal behavior, commands in a message can be successfully parsed and executed even though an error is detected in the message later on. All commands occurring prior to the erroneous command will be properly executed. In order to determine to what extent the parsing and execution procedure has proceeded, the error message gives an indication of where in the command message the error was encountered. This assists the user in determining the cause for the error. Only after the entire message has been parsed and executed and the CR terminator encountered by the command parser will the user receive a '>OK' prompt message indicating that the user can send an additional command message. The prompt message will follow any data response messages sent to the user as a result of command execution requesting the data.

When combining command words and numerical arguments into command groups, the occurrence of the numerical elements serve as command word delimiters and therefore may take the place of 'space' character delimiters if desired. Additional 'space' character delimiters may be added for clarity or appearance if desired. Multiple spaces between command word elements are not significant and are ignored. Embedded spaces within command word or numeric argument fields will cause erroneous parsing of these fields, as they will appear as delimiters. The following are typical examples of command messages and their meanings.

`FILM 1 PARAM 23 = 3:54`

Sets C.V.I. to Film 1 Parameter 23 (Ramp Time 4) and then sets that parameter to the value of 3 minutes and fifty-four seconds.

`F1P23=3:54`

Performs same function except unnecessary spaces to be used as delimiters have been removed. The occurence of the '1' delimits the 'F' from the 'P' and the '23' delimits the 'P' from the '=' as well as serving to provide the necessary numerical argument. The 'F' and 'P' are the minimum words to select the commands.

`STOP CONT ZERO ST`

Causes instrument to stop, continue from remote stop, zero accumulated thickness, and re-start by remote start. Spaces between command words are necessary as no numeric arguments are needed for these commands which could be used as delimiters.

`STO CON Z ST`

Minimum wording to perform same as above.

`AF;AP;AS;`

Causes instrument to return response messages containing the data requested. In this case three response messages will result to return the data requested in the order requested.

`AF;;;`

Has same affect as previous command, but uses the incremental form of access for variables which are ordered one after the other. Refer to Table 8-1 and the section on Remote Programming for a more detailed description of the use of this feature.

`F1P1=1.1;=1.7;=123`

This command points the C.V.I. to Film 1 Parameter 1 (Density) and then incrementally programs three variables to 1.1, 1.7, and 123% by using the ';' command. The ';' command causes the C.V.I. to increment to the next variable if it occurs immediately following an '=' command. In this way only the first of a string of parameters need be explicitly selected. Following the last equivalence, the C.V.I. remains pointing to the Tooling variable that was updated to 123%.

`F1P1;=1.1;;=1.7;;=123`

Same programming effect as previous command except that each ';' command which preceeds the '=' commands causes the previous data value for that parameter to be displayed (response message sent). Each ';' command after a '=' command causes the C.V.I. to index to the next variable.

In the previous example, the instrument would have performed the desired command and would have generated responses that contain the desired data. A typical exchange if in the 'ECHO MODE LONG' mode might appear as follows:

the user transmits-
F1P1;=1.1;;=1.7;;=123 CR (carriage return)

for which the instrument responds-
F1P1;=1.1;;=1.7;;=123 if terminal mode is selected (default condition)
and then responds-

```
CR LF
F1bPb1bDENSITYb-----------------b3.65bbG/CC CR LF
F1bPb2bZ-RATIOb----------------b2.164b----b CR LF
F1bPb3bTOOLINGb------------------b100bb%b-b CR LF
>OK CR LF
```

where the b or hyphen indicate a transmitted space or blank.

Numerical Argument Fields

Certain commands (those of the equivalence type) require a numeric argument to follow them in order to properly specify the action to be taken by the command. In some cases this numeric quantity is the value to be stored in a parameter variable, as in the case of the '=' command. Others require numeric arguments as additional specifiers to indicate what or where. Commands which perform a similar function but with different variables or elements are the most common of these. The 'RY' command, for instance, uses numerical argument values between 1 and 8 inclusive to indicate which relay is the target of the desired operation. Still another use of the numerical argument is in the mode of 'TRUTH' operator. In this mode, the numeric value of '1' or '0' is used in a true or false, active or inactive sense. The 'PARITY' command uses this type of numeric specifier. When followed by a '1', parity sensitivity is enabled, and when followed by a '0', no parity detection is performed. For logical numerical arguments, any number not a '1' or '0' is invalid and will generate an error. The value taken for a numeric argument may have other re-

strictions on its range or use (see Error Messages). Also, at times it may appear that the instrument may have taken an invalid number as valid. An example of this behavior would be:
F996 P902 = 999.900001.1

In this case, the apparent values of '996', '902' and '999.900001.1' should be in error when used as arguments for their respective commands. However, numeric entry only uses the maximum needed last digits entered for a command or parameter. In this case, the 'F' command has a legitimate one digit range of from one to six, and only examines the last single digit, which is a six, and is valid. Likewise for the 'P' command, which accepts two-digit arguments within a range of from one to sixty. The last two digits being '02' are therefore within range. This combination then selects the variable of Z-RATIO, which has a legitimate four-digit decimal range of from .1 to 3.999. It therefore will accept the '999.900001.1' as a '001.1' in the same manner as would the front panel entry for that parameter.

Numerical arguments, when required for programming data variables which are normally programmable from the instrument front panel, take the same form and have the same limitations as do their counterparts entered on the instrument front panel. When numerical data is entered at the front panel, it is entered most significant digit first, with decimal points (or colons, using the decimal point key) entered into their proper position. As each digit is entered, the present string shifts left on the screen, until the number of digits being entered equals the maximum number of digits allowed for that value. After this point the earliest entered digits and decimal points are lost at the left and the value for the entered data is represented by the numbers left on the display. In this way only the last required number of digits entered is accepted for entry and storage update. Although the roll-left and fall off of digits can not be seen when using the interface, the same phenomenon takes place. In this way data representations and limitations are standard between the normal front panel use and RS-232 interface use.

From the preceding examples it can be seen that the desired control and data gathering operations to be performed using the RS-232 interface require that the operator compose the correct sequence of commands to send. This task is similar to that of programming a computer, and involves the same degree of attention to detail. The IC 6000 instrument does not make any effort to determine the validity (usefullness or sensibility) of the commands given it. So long as the commands are correct in structure and the numerical arguments are within their allowed ranges, no error messages will be generated. It is the responsibility of the user to determine the correctness of the commands and responses for the given application.

Error Message Formats and Meanings

As mentioned previously, command messages sent to the instrument are examined for the occurrence of certain invalid input conditions, numeric ranges, and machine statuses or conditions. Illegally constructed commands or commands which are inconsistent with the state of the instrument will cause error messages to be generated when they occur. These errors are detected during the parsing of the command message and subsequent command execution, and the error message responses will be transmitted to the peripheral device at a point in time related to the sequence of events that the command message represented. In order for the peripheral device (especially if a computer) to recognize the response message as an error response and not part of any expected data response messages, an error message is flagged or framed with the occurrence of a special character. This character is the '!' exclamation mark character. Errors may be encountered under two types of operation, and each of these is slightly different in the use of the framing characters.

The primary use of the error response messages is in the detection of syntax or environment (status and configuration) errors in a command message. When these are detected, the interface transmits a lead-in '!' character indicating the start of an error response message. This is followed by the '#' character to indicate that an error number identifier will follow. Following the '#' symbol is a two digit error ID (see Table 8-2). This ID is for the benefit of a computer peripheral to determine the type of error without having to scan the entire error response and command message.

Following the numeric ID for the type of error is an error name identifier to assist the human user in identifying the type of error encountered without resorting to looking up a numeric ID in an error table. This name is a mnemonic contraction of a description of the type of error encountered. A list of the numeric and mnemonic IDs and the types and causes of the errors associated with them is given in Table 8-2. After the numeric and name identifiers for the error encountered is transmitted, a new line is started by the CR/LF combination. The next message sent is a repeat of the entire command line as received by the instrument. This message is terminated by a CR/LF sequence. Finally, that portion of the command message up to where the error was encountered by parsing is repeated. This section of the command message is terminated with an additional '!' framing character and a CR/LF sequence. A new '>OK' prompt is then sent to signal readiness for more command messages. In this way it is possible to determine where in the command message the error was encountered as well as how much of the command was successfully parsed and executed. An example of a command error encountered in a command message would appear as follows;

user sends command-

```
COMP EVEN PAROTY 1 CR
```

command to enable computer mode, even parity and parity sensitivity, with 'PARITY' spelled incorrectly.

instrument responds-

```
!#03 CMDERR
COMP EVEN PAROTY 1
COMP EVEN P!
>OK
```

meaning that a command error was encountered when parsing for a command in the field starting with a 'P'. The 'COMP' and 'EVEN' commands were successfully executed, and only the commands following the error were lost.

If a computer was used as a peripheral device, it would recognize the first occurrence of the '!' as an indication of an error, and would examine the following '#xx' combination to determine the type of error. It could then skip ahead to the next occurence of the framing '!' to bypass the material containing information about the cause of the error.

The other type of error message encountered is related to the use of the optional operational features. The Real Time Clock and Run Finish modes of operation produce time-relevant data messages as a result of internal events or conditions occuring. These messages contain a fixed number of character elements depending on the data desired and the type of echo mode (short or long). The rate of triggering for the generation of this data is variable depending on frequency of occurrence of the real time clock or of successive run finishes. These two facts combine to form conditions where data may be generated faster than it can be disposed of (transmitted over the interface). In this case there exists the opportunity for time-relevant data to be lost.

When this occurs because of a throughput limitation, the instrument disables any further requests of the type causing the data overflow (either RTC or Run Finish options) and generates a data loss error message. These differ from the command syntax type of error messages in that they do not have a numeric identifier, and are framed by double '!' characters as lead in and terminating delimiters, and are only one line as they do not result from a command message and do not echo the previous command line. The formats for these two data loss error messages can be found at the end of Table 8-2.

Original Message Formats and Timing

Original messages are those messages transmitted by the IC 6000 instrument to the peripheral that are not a direct result of a command message being received. The first of these messages is the sign on message first transmitted as part of the interface initialization. Other original messages are those sent as a result of selected options being active (either by explicit command or initially by default) and being triggered. When an original message event triggers, the message contents is buffered into an outgoing buffer specifically designed for this purpose. As soon as any pending original messages are deposited in the buffer, the normal interchanges which take place over the interface are suspended and the contents of the original message buffer are transmitted to the peripheral. When this operation commences a lead-in framing character of an ASCII bell or CTRL-G is sent to indicate an asynchronous break in the normal flow of messages being transmitted by the instrument. This alerts the receiving peripheral that the subsequent characters are part of an original message and have no logical connection to those characters previously received. Following the transmission of all original messages pending in the original message buffer, a trailing framing character (another bell) is sent to indicate the end of the suspension of normal interface exchanges and the resumption of the operation of the interface. In this way a computer device monitoring the interface can direct original messages through a different routing when they occur and handle them as needed and still maintain non-interfered interactive operation. Should multiple messages result from a trigger, as in the case of multiple RTC or Run Finish options enabled, one bell will proceed all the pending messages, and one bell will trail them. Rapid re-triggering may cause a continuous stream of original messages to be generated, and this may cause the indefinite hold-off of normal interactive operation.

Those original messages that function in a data-log mode generate response messages identical to a display type message used to obtain the desired data. They differ in that the numerical or status information contained in them is time synchronized to the occurrence of the triggering event (see Special Operations and Optional Modes). These messages will occur within the 'BELL' framing characters, and this clue discriminates RTC and Run Finish display response messages from those that might occur from an interactive command to display the same data. Original messages which do not contain data (the Alert Mode original messages) have no counterparts in normal interactive operation. These messages have a format which includes '*' framing characters. Detailed descriptions of the Original Message formats appears in Table 8-3.

DISPLAY REQUEST OPERATION AND RESPONSE FORMATS

In addition to the ability to control the operation of the instrument by use of the interface programming and controlling instructions, the interface provides access to measurement data for logging and evaluation. This type of operation is referred to as a DISPLAY function because of its similarity to the use of the instruments display for data examination. Since the interface communicates with serial character messages, however, it is necessary to select or 'point' to the specific data desired. For most display operations the C.V.I. pointer is used in a similar manner to its use for programming data variables. Instead of using the '=' command to alter the data value, a display request command (either ';' or ',') is used. The operation of these two display request commands is different and each has a particular use and purpose. The operation of the ',' is the most straight forward, as its affect is always the same without regard for previous commands or operations. When executed, the ',' command performs an 'examine and increment C.V.I.' action which causes a response message containing the data in question of the present C.V.I., and then increments the C.V.I. to the next variable (see section on Remote Programming and Control for a detailed explanation of the C.V.I. operation). This command is most useful for singular or repetitive examination of data or status variables, since it always requests the display of a variable. Another application of this command is following a '=' command. This will cause the display response of the new updated value, and then increment the C.V.I. pointer to the next variable. In this way the multiple programming and verification or logging of new values may be performed by a single explicit selection (using the Film,Param and symbolic modes of selection) followed by repetitive '=XXX ,' command combinations. This is the only method for examining a newly programmed variable without explicitly re-selecting the variable and examining it. The limitation of this command is that it increments the C.V.I. after examining a variable and generating a response message.

If following examination it is desired to change a variable, a command which examines without incrementing the C.V.I. is needed to eliminate the need to re-select the desired variable. The command which performs this operation (among others) is the ';' command. This command performs a general examine and/or pointer increment function which is dependent upon previous operations. The general operation of the command is such that its execution, when applied to a variable (selected by the present C.V.I.) whose value or state has not been determined by the immediate previous C.V.I. related command, will cause the display response message for that variable to be generated so that the value of the variable can be determined. In this way, the ';' command will cause the display of a data variable when following a new C.V.I. reference (by use of a C.V.I. incrementing, Film, Param, or symbolic reference command) to a variable. If a ';' command is applied to a variable whose value has been determined by the immediate previous C.V.I. related command, then the execution of the ';' command has the effect of incrementing the C.V.I. past the variable whose value has been determined, to the next variable. If the ';' command has been applied after an immediately preceding '=' command, this will be the only affect of the command's execution, allowing for C.V.I. incrementing without display when used in a repetitive '=XX ; =YY ;' command combination sequence. This allows for the complete programming of films and other indexed variables incrementally with only a single explicit reference. If the ';' command is not following a '=' command, however, it will cause the display request and response of the newly selected variable as well.

Response Formats

Once a command to produce a display response is executed, the desired data element(s) are examined and a response message formulated. The response message can be generated in two forms, either short or long format. The format produced is controlled by the EML and EMS (echo mode long and short) commands. Initially the interface defaults to a long message format, where all response messages contain identifiers to indicate the data being examined and transmitted by the message. The type of response mode can be changed by issuing the proper command, and the instrument remains in that mode until altered by a new command, or reinitialized to defaults through removal and restoration of line power. When in the short mode, only the numeric data representing the variable's value is contained in the message, and it is the responsibility of the user to maintain the identity of the source of the data.

Film Parameter Response Formats

Film parameter data variables (parameters numbered 1 through 37) all have the same long or short formats. When in the short format, a film parameter response message contains only the numeric field information as would appear on the display of a given film page. This field is five character spaces in width, and numeric quantities including a decimal have that decimal occupying one of the spaces. Numeric quantities are right justified so that leading character positions contain an ASCII space. In this way a list of parameter data is always read in by a computer (or displayed on a terminal) as a fixed number of characters. When a long format is used with these film parameters, additional identifying and descriptive fields precede and follow the numeric information field. The format for the long response message for a film parameter variable is:

```
FXbPYYbLABELb------------------bNMFLDbbUNIT CR LF
```

Where the b or hyphen indicate spaces,
'FX' is the film identifier field, any one of 'F1' through 'F6',
'PYY' is the parameter ID field, where 'YY' can range from 'b1' to '37',
'LABEL' is the parameter name as shown on a film page display,
'NMFLD' is the five character numeric field as in the short mode above,
'UNIT' is a descriptive field as on a page display (ex. 'G/CC','M:Sb', etc.)
'CR LF' is the message termination combination.

The parameter label field's two components (label and trailing spaces) may vary in length from a minimum of four letters followed by twenty-one spaces, to a maximum of eleven characters followed by fourteen spaces. Certain parameter labels contain embedded spaces (such as 'FINAL THK') and these spaces are considered part of the label field character count. The occurence of two consecutive spaces will signal the end of the LABEL field and the next non-blank character will be the first significant number of the numeric field.

Executive and I/O Page Parameter Response Formats

Programmable parameters that are normally displayed and programmed on display pages seven and nine by using a movable cursor are accessed by a unique ID number when using the RS-232 interface. These numbers (listed in Table 8-1) are an extension of the series of numbers used for the film page parameters. These parameters are referenced by the C.V.I. command when the 'P' command is used with a numeric argument of between 38 and 60, inclusive (see Remote Control and Programming). When such a reference is followed by a display request command (either ';' or ',') a response message is generated for that variable in either the long or short format. The short format for each parameter consists of only the numerical field for that variable as defined in the long formats for each parameter type below:

Executive Page Parameter Response Formats

Lock Code Parameter:

```
b-bP38bLOCKbCODEb--------------bNNNNN CR LF where-
the b or hyphen are blanks or spaces,-
'P38' is the parameter ID field,-
'LOCKbCODE' is the parameter label,-
'NNNNN' is the lock code numeric field ranging from 'b--b0' to 'b9999' and an
                exceptional case of 'b%342' following execution of an 'FP'
                command. Leading zeros are blanked with spaces.-
'CR LF' is the message line terminator.
```

Requested Active Process Parameter:

```
b-bP39bREQUESTEDbACTIVEbPROCESSbNNNNN CR LF where-
'NNNNN' is the active process numeric field ranging from 'b--bl' to 'b--b4'.
```

Layer to Start Parameter:

```
b-bP40bLAYERbTObSTARTb---------bNNNNN CR LF where-
'NNNNN' is the layer to start numeric field ranging from 'b--bl' to 'b-b32'.
```

Run Number Parameter:

```
b-bP41bRUNbNUMBERb-------------bNNNNN CR LF where-
'NNNNN' is the run number numeric field ranging from 'b--b0' to 'b9999'.
```

Process Sequence 1 through 3:

```
b-bPXXbPROCESSbSEQUENCEbYb-----bS.............................. CR LF where
'PXX' is the parameter ID field ranging from 'P42' through 'P44' for process
                sequences 1 through 3, respectively,-
'Y' is the sequence ID field ranging from 'l' to '3',-
'S.............................' is a variable length field of from 0 to 32
                integers of a value ranging from 1 to 6. This field is used
                to indicate the films contained in the desired sequence.
```

I/O Page Parameter Response Formats

Output Parameter Formats:

```
b-bPXXbOUTPUTbYb---------------bXY.Zb-bFUNCTION LBbQQ CR LF where-
'PXX' is the parameter ID field ranging from 'P45' to 'P52' for outputs one
                through eight, respectively-
'OUTPUTbY' is the parameter label field ranging from 'Y'=1 to 8,-
```

'XY.Z' is the output numeric field containing the value which selects the desired relay function, as explained in the main manual section on I/O programming.-

'FUNCTION LB' is the eleven character field which contains the symbolic name for the relay function selected by the numeric value. These names are the same as those which appear on display page 9 for these programmable functions.-

'QQ' is the contact code label field containing either 'bb', 'NO','NC', 'PO', or 'PC'.

Input Parameter Formats:

b-bPXXbINPUTbYb----------------bXYb---bFUNCTION LB CR LF where-

'PXX' is the parameter ID field ranging from 'P53' to 'P60' for inputs one through eight,respectively,-

'INPUTbY' is the parameter label field ranging from 'Y'=1 to 8,-

'XY' is the input parameter numeric field containing the value which selects the desired input function, as explained in the main manual section on I/O programming.-

'FUNCTION LB' is the eleven character field which contains the symbolic name for the input function seleceted by the numeric value. These names are the same as those which appear on display page 9 for these programmable functions.-

Symbolically Referenced Variables Response Formats

Active Film Number Response Format:

ACTIVEbFILMb-------bN CR LF where-

'ACTIVEbFILM' is the parameter label field,-

'N' is the numeric field ranging from '1' to '6'.

Active Process Number Response Format:

ACTIVEbPROCESSb----bN CR LF where-

'ACTIVEbPROCESS' is the parameter label field,-

'N' is the numeric field ranging from '1' to '4'.

Active Source Number Response Format:

ACTIVEbSOURCEb-----bN CR LF where-

'ACTIVEbSOURCE' is the parameter label field,-

'N' is the numeric field ranging from '1' to '4' if sources present.

Average Power Value Response Format:

AVERAGEbPOWERb-----bNN.N CR LF where-

'AVERAGEbPOWER' is the parameter label field,-

'NN.N' is the numeric field ranging from '00.0' percent power to '99.9' .

Average Rate Value Response Format:

AVERAGEbRATEbA/Sb--bNNN.NN CR LF where-

'AVERAGEbRATEbA/S' is the parameter label field,-

'NNN.NN' is the one second average rate in angstroms/second ranging in value from '000.00' to '999.99'.

Input Conditions Response Format:

INPUTbCONDITIONSb--bIIIIIIII CR LF where-
'INPUTbCONDITIONS' is the parameter label field,-
'IIIIIIII' is the numeric field representing the state of each input. Each 'I' may be either a '1' or a '0' indicating either active or inactive, respectively. The order of listing for the inputs is '87654321' for each respective input terminal.

Specific Input Condition Response Format:

INPUTbNbCONDITIONb-bI CR LF where-
'INPUTbNbCONDITION' is the parameter label field with 'N' ranging from 1 to 8 identifying a particular input terminal.
'I' is the numeric field representing the state of the input as either a '1' (active) or '0' (inactive).

Layer Number Response Format:

ACTIVEbLAYERbNO.b--bNN CR LF where-
'ACTIVEbLAYERbNO.' is the parameter label field,-
'NN' is the numeric field ranging from '01' to '32'.

Lamp Status Conditions Response Format:

LAMPbSTATUSbMXARb--bLLLL CR LF where-
'LAMPbSTATUSbMXAR' is the parameter label field,-
'LLLL' is the numeric field representing the state of each front panel lamp. Each 'L' may be either a '1' or a '0' indicating either illuminated or extinguished, respectively. The order of listing for the lamps is Manual, Xtal, Abort, and Ready, as given by the order of the 'MXAR' letters in the label field.

Manual Lamp Status Response Format:

MANUALbLAMPbSTATb--bL CR LF where-
'MANUALbLAMPbSTAT' is the parameter label field,-
'L' is the numeric field representing the state of the Manual front panel lamp, with a '1' indicating an illuminated state and a '0' an extinguished state.

Crystal (Xtal) Switch Lamp Status Response Format:

XTLbSWbLAMPbSTATb--bL CR LF where-
'XTLbSWbLAMPbSTAT' is the parameter label field,-
'L' is the numeric field representing the state of the Xtal Sw front panel lamp, with a '1' indicating an illuminated state and a '0' an extinguished state.

Abort Lamp Status Response Format:

ABORTbbLAMPbSTATb--bL CR LF where-
'ABORTbbLAMPbSTAT' is the parameter label field,-
'L' is the numeric field representing the state of the Abort front panel lamp with a '1' indicating an illuminated state and a '0' an ex-

tinguished state.

Ready (Start/Cont) Lamp Status Response Format:

READYbbLAMPbSTATb--bL CR LF where-

'READYbbLAMPbSTAT' is the parameter label field,-

'L' is the numeric field representing the state of the Ready front panel lamp with a '1' indicating an illuminated state and a '0' an extinguished state.

Layer Timer Value Response Format:

LAYERbTIMERbbM:Sb--bTT:TT CR LF where-

'LAYERbTIMER' is the parameter label field,-

'TT:TT' is the numeric field ranging from '00:00' to '99:59' mins:secs.

Maximum Power Condition Status Response Format:

MAXbPOWERbSTATUSb--bN CR LF where-

'MAXbPOWERbSTATUS' is the parameter label field,-

'N' is the numeric field indicating the presence ('1') or absence ('0') of the maximum power condition.

Next Film (presoaking film) Number Response Format:

PRESOAKbFILMbNO.b--bN CR LF where-

'PRESOAKbFILMbNO.' is the parameter label field,-

'N' is the numeric field ranging from '1' to '6'.

Next Source (presoaking source) Number Response Format:

PRESOAKbSOURCEbNO.bbN CR LF where-

'PRESOAKbSOURCEbNO.' is the parameter label field,-

'N' is the numeric field ranging from '1' to '4'.

NOTE: When presoak is not operating, Next Film and Source are the same as active film and source numbers.

Process Phase Response Message Format:

PROCESSbPHASEb-----bNNbbPHASE-MESAG CR LF where-

'PROCESSbPHASE'is the parameter label,-

'PHASE-MESAG' is a phase label identical to that displayed on the Data Display Page 8 as the Phase message as listed in the table below. -

'NN' is the numeric field ranging from '00' to '15' and indicating phases according to the following table:

'00'	'b-bREADYb-b'	
'01'	'bbRISEb1b-b'	
'02'	'bbSOAKb1b-b'	
'03'	'bbRISEb2b-b'	
'04'	'bbSOAKb2b-b'	
'05'	'bFEEDbRAMPb'	
'06'	'bFEEDbSOAKb'	
'07'	'bIDLEbRAMPb'	
'08'	'b-bIDLEb--b'	ZERO IDLE POWER PHASE
'09'	'b-bIDLEb--b'	NON ZERO IDLE POWER PHASE
'10'	'SHUTTERbDLY'	
'11'	'bbMANUALb-b'	

'12'	'bbDEPOSITbb'
'13'	'RATEbRAMPbl'
'14'	'RATEbRAMPb2'
'15'	'TIME-POWERb'

Phase Timer Value Response Format:

PHASEbTIMERbbM:Sb--bTT:TT CR LF where-
'PHASEbTIMER' is the parameter label field,-
'TT:TT' is the numeric field ranging from '00:00' to '99:59' mins/seconds.

Source Power Levels Response Format:

SOURCEbXb%bPOWERb--bNN.NNNNNN CR LF where-
'SOURCEbX' is the parameter label field and ranges from 'SOURCEbl' to 'SOURCE b4' providing the source/sensor module is installed.
'NN.NNNNNN' is the numeric field ranging from '00.000000' percent power to '99.000000' for either the active or presoak source (not to exceed the maximum power in their respective film programs) and up to '99.999999' for any sources not presently defined as active or presoak sources, and not subject to maximum power limits programmed in film programs.

Quality Error Accumulator Response Format:

QUALITYbERRORbVALb-bNN CR LF where-
'QUALITYbERRORbVAL' is the parameter label field,-
'NN' is the numeric field ranging from '00' to '50' and representing the same value as displayed on Data Display Page 8 alternately with crystal life when the decimal point front panel key is depressed.

Rate Deviation Response Format:

'RATEbDEVIATIONb---bNN CR LF where-
'RATEbDEVIATION' is the parameter label field,-
'NN' is the numeric field ranging from '00' to '80' and representing bipolar rate deviations from -40 Angstroms/Second to +40 A/S with a deviation of 0 being represented by '40'. This value corresponds to the Y-axis displacement of the real time graph as displayed on Data Display Page 8 of the instrument.

Instantaneous Rate Response Format:

INSTANTbRATEbA/Sb--bNNN.NN CR LF where-
'INSTANTbRATE' is the parameter label field,-
'NNN.NN' is the numeric field representing the instantaneous (single sample) rate in angstroms/second ranging in value from '000.00' to '999.99'.

Relay Contact Status Response Format:

RELAYbXbCONDITIONb-bN CR LF where-
'RELAYbX' is the parameter label field and ranges from 'RELAYbl' to 'RELAYb8' providing the optional I/O module for that relay number is installed.

'N' is the numeric field representing the status of the designated relay contact. A value of '1' indicates a closed relay contact, and a value of '0' indicates an open relay contact.

Relay Conditions (all 8 at once) Status Response Format:

RELAYbCONDITIONSb--bNNNNNNNN CR LF where-
'RELAYbCONDITIONS' is the parameter label field,-
'NNNNNNNN' is the numeric field representing the status of the eight relay contacts. Each 'N' may have a value of '1' or '0', indicating a closed or open condition, respectively. The listing order of the eight relays is '87654321' and non-installed relays show a '0' status.

Stability Error Accumulator Response Format:

STABILITYbACCUMb---bNNNN. CR LF where-
'STABILITYbACCUM' is the parameter label field,-
'NNNN.' is the numeric field ranging from '0000.' to '9999.' and representing the same value as displayed on Data Display Page 8 alternately with crystal life when the decimal point front panel key is depressed.

Annunciator Status Response Format:

ANNUNCIATORbSTATb--bNXXbSTAT-LABEL CR LF where-
'ANNUNCIATORbSTAT' is the parameter label field,
'NXX' is the Status ID numeric field, whose values and corresponding meanings are listed in the table below:,-
'STAT-LABEL' is the Status ID label field comprised of elements from the table below, based on the condition of the status.

NOTE: The unique ID of '0b0' indicates a "NULL" status condition which has no annunciator message appearing on Data Display Page 8 and no Status ID label field present in the response message.

Status ID Number and Label Table:

'N' Value	Associated Label
'0'	'STOPb'
'1'	'ABORTb'
'2'	'ENDb'
'3'	'POWERb'
'4'	'FAULTb'

'XX' Value	Associated Label
'b0'	'b----------b'
'b1'	'bMAXbPOWERbb'
'b2'	'FRONTbPANELb'
'b3'	'bbNObFILMb-b'
'b4'	'bXTALbFAILbb'
'b5'	'b-bMANUALb-b'
'b6'	'bTIME-POWERb'

'b7'	'b--bFILMb--b'
'b8'	'bbEXTERNALbb'
'b9'	'bPOWERbFAILb'
'10'	'bbPROCESSb-b'
'11'	'bIObPROGRAMb'
'12'	'FAILURE-SAFE'
'13'	'FAILURE-LOST'
'14'	'bNObSOURCEbb'
'15'	'bFILMbRESETb'
'16'	'PROC-bRESETb'
'17'	'PROC-bCONTIN'

Test Mode Status Response Format:

This parameter is unique in that the numeric field presented in the response message of a short format type is not contained in the same form in the status field of the long format response message.

Echo mode long format:
TESTbSTS CR LF where-
'TEST' is the parameter label field,-
'STS' is the status field containing either 'ON' or 'OFF' as appropriate.

Echo mode short format:
N CR LF where-
'N' is a numeric field with '0' indicating test off and '1' test on.

Thickness Response Format:

THICKNESSbINbKAb---bNNN.NNN CR LF where-
'THICKNESS' is the parameter label field,-
'NNN.NNN' is the numeric value field ranging from '000.000' to '999.999'.

Crystal Failure Status Response Format:

XTALbFAILUREbSTATb-bN CR LF where-
'XTALbFAILUREbSTAT' is the parameter label field,-
'N' is the numeric field indicating the status of the "crystal failed" condition. A '1' indicates that a crystal has previously failed and that the "XTAL" legend area of the Data Display Page 8 is flashing and that the crystal fail relay is active (if it is programmed). A '0' indicates that a crystal failure condition has been cleared by a front panel or remote crystal switch function or by a new film initialization.

Crystal (sensor) Number Response Format:

XTALbSENSORbNO.b---bN CR LF where-
'XTALbSENSORbNO.' is the parameter label field,-
'N' is the numeric field indicating the sensor in use and ranging from '1' to '4' in value.

Crystal Life Value Response Format:

XTALbLIFEbNUMBERb--bNN CR LF where-
'XTALbLIFEbNUMBER' is the parameter label field,-

'NN' is the numeric value field ranging from '00' to '99'.

SPECIAL OPERATIONS AND OPTIONAL MODES

The IC 6000 RS-232 interface has optional features that allow the IC 6000 to generate original messages under certain conditions of interest. Certain of these options are selected or installed automatically every time the IC 6000 instrument is turned on. These selected options are referred to as the default options and are a subset of all of the options. That is to say that certain options available are not installed automatically and can only be selected by the suitable command. The list of all options available follows, and those which are of default nature are followed by an asterisk, as well as listed separately in the section on Default Conditions.

OPT 0 Special OPTION command that deactivates all previously selected options, whether from a default or requested state.

OPT 1 REAL TIME CLOCK Thickness. When the real time clock interval expires a data log dump of the present thickness is generated.

OPT 2 REAL TIME CLOCK Rate. When the real time clock interval expires a data log dump of the present instantaneous rate is generated.

OPT 3 REAL TIME CLOCK Average Rate. When the real time clock interval expires a data log dump of the present average rate (over previous one second) is generated.

OPT 4 REAL TIME CLOCK Rate Deviation. When the real time clock interval expires a data log dump of the last averaged rate deviation value as defined by the analog plot of the IC 6000 is generated.

OPT 5 REAL TIME CLOCK Active Power. When the real time clock interval expires a data log dump of Active Power is generated.

OPT 6* RUN FINISH Group 1. When the IC 6000 ends a run from stop, abort or final thickness, a data log dump of the RUN NO., FILM NO., LAYER NO., and PROCESS NO. is generated.

OPT 7* RUN FINISH Group 2. As above except the data logged is STATUS, CRYSTAL NO., CRYSTAL LIFE NO., AND PHASE.

OPT 8* RUN FINISH Layer Time. As above except the data logged is contents of layer timer.

OPT 9* RUN FINISH Thickness. As above except the data logged is thickness at the time of run finish.

OPT 10 RUN FINISH Average Power. As above except the data logged is average deposition power during the rate control.

OPT 11 RUN FINISH Graph. As above except the data logged is the values for the average rate deviations as recorded for the analog plot.

OPT 12* STOP ALERT. When the IC 6000 enters a stopped state an alert message is generated.

OPT 13* ABORT ALERT. When the IC 6000 enters an aborted state an alert message is generated.

OPT 14* END ALERT. When the IC 6000 status enters an END state an alert message is generated.

OPT 15* MAX POWER ALERT. When the IC 6000 reaches a maximum power limit condition an alert message is generated.

OPT 16* TIME POWER ALERT. When the IC 6000 enters the time power deposition phase an alert message is generated.

OPT 17* CRYSTAL FAILURE ALERT. When the IC 6000 enters a state where the crystal failure relay is active or display field is flashing an alert message is generated.

Alert Mode Operation

In order to reduce the need for the peripheral device to request status or 'poll' the IC 6000 instrument to determine important operating conditions, an automatic Alert Mode ability was designed into the interface system. Through these optional modes, the instrument can be programmed to generate an original message (one which is not a response to a user command) upon the occurrence of certain events or changes within the IC 6000. This frees the peripheral device or user from the need to routinely poll or query the IC 6000 to determine if one of these major events or changes has occurred. This communication by exception results in greater system throughput. If an Alert Mode option is selected, then an Alert Message will be generated whenever the desired event occurs. One and only one message will be generated for each occurrence of the event, regardless of the event's duration. It is necessary for the condition that causes the event to cease and re-occur for another message to be generated. A typical example of this operation is in the use the STOP ALERT option.

When this option is selected (either on turn on as a default, or by a command) the instrument will generate a message (**STOP ALERT**) whenever the instrument first enters a stopped condition. The message will only be generated once at the beginning of the stopped condition. If the instrument is removed from the stopped condition (through an ABORT RESET or CONTINUE function), the instrument will then be capable of generating a new stop alert message should it re-enter a stopped condition. This type of operation is referred to as edge-triggered. All alert message options are of this type. In addition to being edge-triggered, the alert message options are also non-cumulative. This means that if multiple edges for an event (a stopped-ready-stopped sequence for example) occur in a period of time before the instrument can generate the proper alert as a response for the event, only one message will be generated even though the instrument went in and out of a stopped state more than once. If more than one of the Alert options is active, then the occurrence of one or more events may cause the generation of several alert messages. As soon as any one alert message is needed, that need is latched or flagged on a message queue. As soon as any original messages are needed, the needed message queue will be processed in a fixed order not related to the order in which the triggering events occured. The order of listing as an option (options 1-17) indicates the order of priority from highest to lowest for the original messages used by the IC 6000. In this way, when simultaneous events occur, the sequence of messages generated is fixed. Only when a sufficient time interval for one message to be fully processed elapses between random real time events will each event's message be generated to the peripheral device in the sequence in which the actual events took place.

Run Finish Mode Operation

In addition to the exception reporting of special machine status changes by way of the Alert Message options, certain process variables and numerical values may also be generated as an original message upon the occurrence of an internal machine status or event. This capability is provided through use of the Run Finish Mode options. Most of the available options of this mode are selected automatically as defaults following initial power on. Any or all of the options may be selected through use of the OPT command. When enabled, these options allow for the data logging of the values of particular variables or internal conditions at the instant of trigger. The condition that constitutes the trigger is the finishing of a run. This may occur either by way of a final thickness limit being encountered during a deposition phase, or by a stop, abort, or remote final thickness trigger commands.

Since both the Alert Message and Run Finish modes of operation are enabled automatically as defaults by the IC 6000, it is not necessary to send a command to enable these options. In this way the IC 6000 may be used automatically upon power on when attached to a receive only RS-232 peripheral device. A simple printer device may then be used as a data logging device providing the variety of data supplied through the Alert and Run Finish modes is sufficient.

Real Time Clock Operation

Under normal interface operation, commands sent to the IC 6000 instrument are processed under an as-soon-as-possible basis, so as not to conflict with the primary instrument functions of measurement and control. Because of the randomness of the timing between the ongoing processes within the IC 6000 and the incoming command messages, the amount of delay or time slew between the time a command is issued to request data (such as rate) and the time that command is executed and a response formulated is indeterminate. Because a great deal of the information generated by the IC 6000 is time related, this phenomenom is undesirable when current or synchronized data must be transmitted. To overcome this problem, a special data logging mode has been provided by the RS-232 interface.

This mode is referred to as the Real Time Clock or RTC mode. This mode is not related to absolute time-of-day clock operation, and is not meant to provide chronometric information. Rather, its purpose is to provide an internal, non slewed, repetitive timebase for the sampling of time-critical data. When this option is selected, the IC 6000 will generate original messages containing the desired data values on a rate which is determined by the period setting of the real time clock time base. This time base is programmable (through use of the CLK command) from between 0 and 100 tenth second intervals. The value of 0 deactivates the timing operation and causes the generation of RTC messages to cease. It does not, however deactivate or cancel the Real Time Clock options previously selected, and they will again start generating messages should the CLK command be used to program a non-zero interval. The maximum period of the interval is 100 tenth-seconds, or ten seconds. With this setting, the desired data elements will be sampled every ten seconds. The samplings all occur during the same internal phase of the measurement and represent synchronized or simultaneous data values.

Table 8-1 Parameter Reference List

Parameter Number	Variable Name
1	Density
2	Z-Ratio
3	Tooling
4	Sensor
5	Source
6	Gain
7	Approach
8	Limiter
9	Soak Power 1
10	Ramp Time 1
11	Soak Time 1
12	Soak Power 2
13	Ramp Time 2
14	Soak Time 2
15	Rate
16	Shutter Delay
17	Final Thickness
18	Thickness Limit
19	Feed Power
20	Ramp Time 3
21	Feed Time
22	Idle Power
23	Ramp Time 4
24	Max Power
25	Stop if greater than Max Power
26	Stop or Continue on Time Power if Xtal Fail
27	Q Factor
28	S Factor
29	Time Limit
30	Presoak Enable
31	New Rate of Rate Ramp 1
32	Start Ramp
33	Ramp Time
34	New Rate of Rate Ramp 2
35	Start Ramp
36	Ramp Time
37	Analog Plot Dwell
38	Lock Code
39	Requested Active Process
40	Layer To Start
41	Run Number
42	Process Sequence 1
43	Process Sequence 2
44	Process Sequence 3
45	Output 1
46	Output 2
47	Output 3
48	Output 4
49	Output 5
50	Output 6

Parameter Number	Variable Name
51	Output 7
52	Output 8
53	Input 1
54	Input 2
55	Input 3
56	Input 4
57	Input 5
58	Input 6
59	Input 7
60	Input 8

Symbolically Defined Variables and Parameters

Symbol Name	Parameter Description
AF	Active Film Number
AP	Active Process Number
AS	Active Source Number
AVp	Average Power Value
AVR	Average Rate Value
Icnd	Input Conditions
IN (1-8)	Specific Input Condition 1 through 8.
LYr	Layer Number
LCnd	Lamp Conditions (Man,Xtal,Abort,Ready)
LM	Lamp, Manual Condition
LX	Lamp, Crystal Condition
La	Lamp, Abort Condition
LR	Lamp, Ready Condition
LYRT	Layer Timer Value
MP	At Max Power Flag Status
Nf	Next Film Number
NS	Next Source Number
PH	Phase Name Data
PHT	Phase Timer Value
POw (1-4)	Percent Power to Sources 1 through 4.
Q	Quality Accumulator Value
RD	Rate Deviation Value
Rate	Instantaneous Rate Value
RY (1-8)	State of Relay Contacts 1 through 8
RCnd	State of all Eight Relay Contacts at once
Sa	Stability Accumulator Value
STAt	Annunciator Status Data
TSt	Test Status or data Variable
Thick	Thickness Value
Xfl	Crystal Failure Flag data
XNum	Crystal Number (1-4) value
XLif	Crystal Life Value

Table 8-2 Error Message Formats and Descriptions

Error Message Formats and Descriptions:

ERROR MESSAGE	REASON FOR OCCURRENCE
Buffer Overflow !#01 BUFOVR!	Sent when a command line exceeds eighty characters in length.
Value Error !#02 VALERR ...	Sent when the numeric value of an argument is improper for the command to which it is being applied. Followed by a repeat of the entire command message as well as that segment up to the bad numeric value.
Command Error !#03 CMDERR ...	Sent when a field presumed to be a command word is not recognized as a valid command. Followed by a command message echo indicating the location of the unrecognized field.
Data Missing Error !#04 DATERR ...	Sent when a command requiring a numeric argument is followed by another command and not a numeric value. Followed by a message echo indicating the location where the needed numeric field should be.
Quotation Error !#05 QUTERR ...	Sent when a command contains unmatched single quotes used to demarcate a comment field. Ending quote was missing.
Status Error !#06 STATERR ...	Sent when the status of the instrument does not allow a valid command to act as it should. Examples include entering test mode while not stopped or ready. This is interlocked and results in a status error message being sent. It is followed by a message echo indicating which command was unable to perform.
Programming Conflict Error !#07 PROGERR ...	Sent when a command can't act because the instrument is programmed so as to conflict with the use of the command. This occurs when a command to control a relay's contacts addresses a relay which is programmed to have a dedicated internal use.
Configuration Error !#08 CNFGERR ...	Sent to indicate a command has addressed an optional or removable hardware element (plug in module) which is not installed.

Error Message Formats and Descriptions (cont.)

ERROR MESSAGE

REASON FOR OCCURRENCE

Real Time Clock Data Loss
!! RTC DATA LOSS !!

Sent when real time clock generated data is produced faster than it can be disposed of. Following the generation of this message, all RTC options are disabled. The RTC interval remains set and reselecting any RTC options enables production of RTC triggered data. This error message occurs within the framing bell characters of an original message generated by the RTC options.

Run Finish Data Loss
!! RFN DATA LOSS !!

Sent when a run finishes and produces option selected data for which a previous run finish event has also data-logged, but has not yet disposed of. This results in the loss of the latest requested data and in the deactivation of any further run finish options until re-enabled by an OP command. This error message occurs within the framing bell characters of an original message which was generated by a Run Finish option.

Table 8-3 Original Message Formats

Alert Mode Original Message Formats and Descriptions

ORIGINAL MESSAGE	REASON FOR OCCURRENCE
STOP ALERT	Sent when Option 12 is selected and the unit enters the stopped condition.
ABORT ALERT	Sent when Option 13 is selected and the unit enters the aborted condition.
END ALERT	Sent when Option 14 is selected and the unit enters a state displaying the END message in the annunciator field of display 8.
MAX POWER ALERT	Sent when Option 15 is selected and the unit enters a condition where the maximum power limit warning (a flashing of the source power number in the activity display area) occurs for either the active or presoak source.
** TIME POWER ALERT**	Sent when Option 16 is selected and the unit enters the Time Power phase of deposition.
CRYSTAL FAILURE ALERT	Sent when Option 17 is selected and the unit enters a state where the 'XTAL' display field of display 8 is flashing indicating a previous failure of a crystal sensor.

Real Time Clock Mode Original Message Formats

Thickness Data Log	A display response message with the same format as a 'Thick' display request response will be sent when the RTC interval expires if Option 1 is selected.
Instantaneous Rate Data Log	A display response message with the same format as a 'Rate" display request response will be sent when the RTC interval expires if Option 2 is selected.
Average Rate Data Log	A display response message with the same format as an 'AVR' display request reponse will be sent when the RTC interval expires if Option 3 is selected.

Rate Deviation Data Log

A display response message with the same format as a 'RD' display request response will be sent when the RTC interval expires if Option 4 is selected.

Active Power Data Log

A display response message with the same format as a 'POw' display request response of the active source power will be sent when the RTC interval expires if Option 5 is selected.

Run Finish Mode Original Message Formats

Run Finish Group 1

A list of display response messages having the same formats as display request responses for 'P41' (Run Number), 'AF', 'LYr', and 'AP' commands will be sent when the unit enters a Run Finish state if Option 6 is selected.

Run Finish Group 2

A list of display response messages having the same formats as display request responses for 'STAt','XNum', 'XLif', and 'PH' commands will be sent when the unit enters a Run Finish state if Option 7 is selected.

Run Finish Layer Time

A display response message with the same format as a 'LYRT' display request response will be sent when the unit enters a Run Finish state if Option 8 is selected.

Run Finish Thickness

A display response message with the same format as a 'Thick' display request response will be sent when the unit enters a Run Finish state if Option 9 is selected.

Run Finish Average Power

A display response message with the same format as an 'AVp' display request response will be sent when the unit enters a Run FInish state if Option 10 is selected.

Run Finish Graph

A display response message with the same format as a 'Gr' command generates will be sent when the unit enters a Run FInish state if Option 11 is selected.