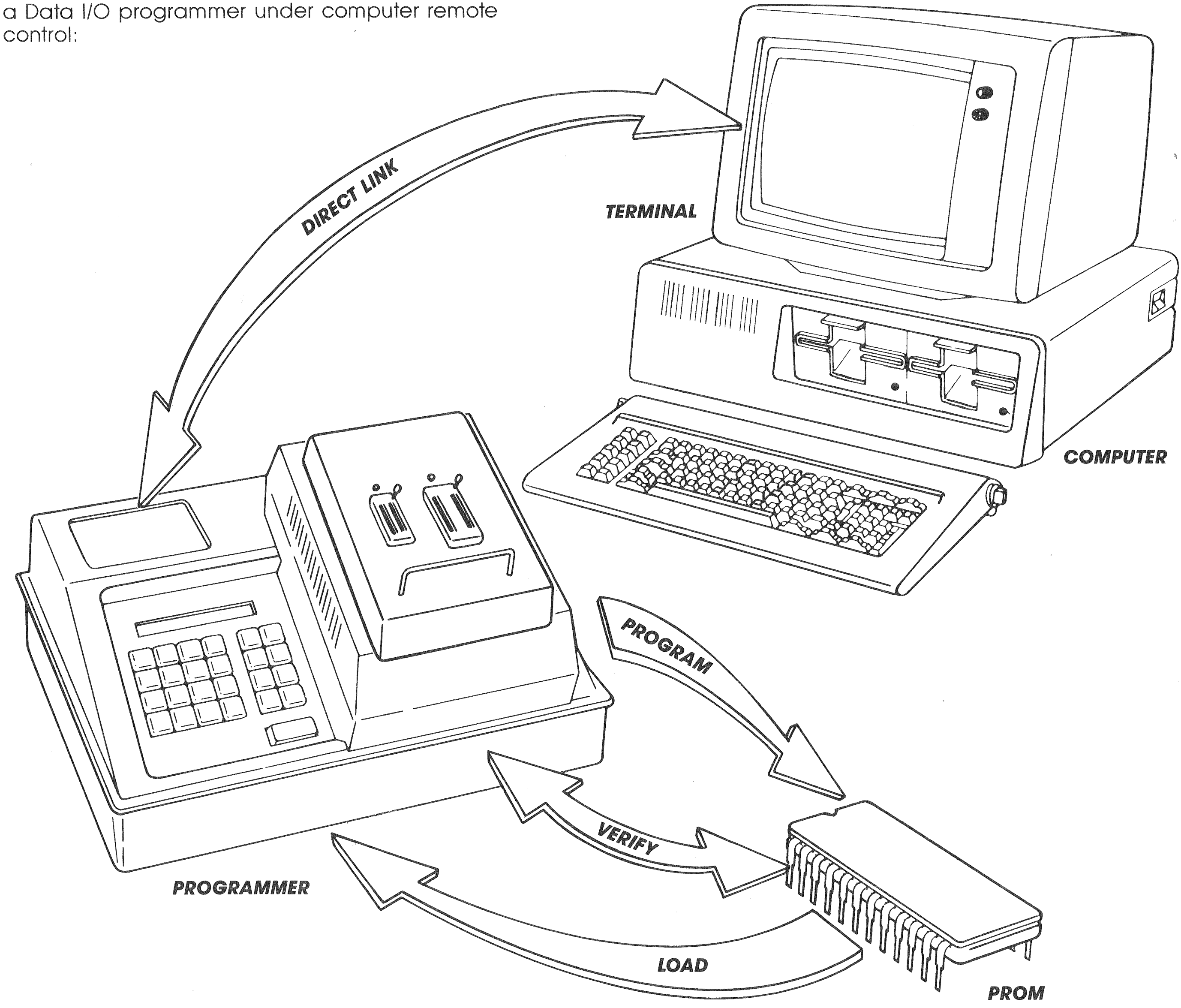
APPLICATION NOTE

USING COMPUTER REMOTE CONTROL ON A DATA I/O PROGRAMMER

The following figure illustrates the basic components of a Data I/O programmer under computer remote control:



INTRODUCTION

Data I/O device programmers can be remotely controlled via the RS232C serial port by sending ASCII characters to the programmer and interpreting the programmer's response. There are three different types of remote control, namely, Computer Remote Control, System Remote Control and Terminal Remote Control. Both System and Terminal Remote Control are intended for operation from a terminal or to accommodate users whose existing operations require either of those modes of operation. The remainder of this application note refers to Computer Remote Control and was written to better explain how to write a Computer Remote Control driver. The driver will generate and send commands to the programmer and react to information returned to it from the programmer. While these commands may be sent by an operator at a terminal, the commands and syntax of Computer Remote Control were designed to be easily incorporated into a computer program.

NOTE: If you see this key symbol, it means to press the noted key on the Data I/O Programmer. X

DATA I/O

TABLE OF CONTENTS

THE COMPUTER REMOTE CONTROL DRIVER

Data I/O has defined a Computer Remote Control standard for its programmers. Each command is assigned an ASCII character. (These ASCII characters are summarized in Table B.) When the programmer is sent one of these characters, followed by a carriage return (CR), it executes the command and sends back a response character followed by a carriage return.

These RESPONSE CHARACTERS are summarized below:

Table A. Response Characters

Character	Name	Description
>	Prompt	Sent on entering remote control, after an ESCAPE or BREAK key has halted a command, or after a command has been successfully executed. The programmer then transmits a carriage return.
F	Fail	Informs the computer that the programmer has failed to execute the last command entered. The programmer then transmits a carriage return.
?	Question	Informs the computer that the programmer does not understand a command or the command was invalid. The programmer then transmits a carriage return.

The first thing a driver should do is verify that the programmer and the computer are properly communicating. The programmer's response will *ALWAYS* contain a response character followed by a carriage return. In addition, the response may contain a line feed (LF) and nulls (ASCII character 00). This is dependent on the null count, which is set using the U command (Set Nulls). If the null count is set to any hexadecimal value from 00 to FE, the programmer's response will contain a line feed followed by that many nulls (0 to 254). The default null count is FF, which results in both line feeds and nulls being suppressed.

> NOTE: The model 100A does not support the U command and will *ALWAYS* terminate its response with a response character followed by a CARRIAGE RETURN and a LINE FEED (no nulls).

EXAMPLE: If we send the programmer ASCII characters "05 U" (zero, five, letter U) followed by a carriage return (where 05 is the hexadecimal null count), the programmer should respond with a response character followed by a CARRIAGE RETURN followed by a LINE FEED followed by five NULLS.

Although any command can be used to verify communication, the U command is suggested because it responds quickly and specifies the exact ASCII characters to send back. The driver could then send any other command to the programmer and expect the response to end with these same ASCII characters. The H command (no operation) is another fast, handy, universal command that could be used to verify communication, including Data I/O's model 100A.

After verifying communication, the driver should allow the selection of available commands, possibly from a menu. The user would make a selection and the ASCII characters corresponding to the selected command would be sent to the programmer (See Table B). The computer would then wait for a response from the programmer. This response could take anywhere from seconds to minutes. For example, the H command (no operation) takes about one second, while the P command (program) can take up to 15 minutes. If you wish to abort a command while it is in operation, just send an ESCAPE (ASCII character 27) to the programmer, which will then send another response character.

The programmer's response will *ALWAYS* contain a response character. (See Table A.) The driver should input and interpret these response characters from the programmer.

Response Characters

If the command succeeded, the response character will be a "greater than" sign (>), which may or may not be preceded by other ASCII characters, depending on the command. These other characters would contain information pertaining to the selected command and should be interpreted by the driver. For example, the G command (Software Configuration) will respond with four other characters before the ">" sign. These characters represent the sum-check of the software in your programmer.

If the command failed, the response character will be the ASCII letter "F". The driver could then use the X command (Error Code Inquiry) to interrogate the programmer as to what the error is. The programmer would then send back a two-digit code (followed by the ">" response character), which represents the last error incurred.

If the programmer did not understand the command, the response character will be an ASCII question mark "?". This occurs when the command is not supported on your programmer, or the wrong ASCII characters were received by the programmer.

After receiving a response from the programmer, the driver should input and interpret the response. Based on this, an appropriate message can be displayed on the screen for the operator (error messages, explanations, etc.).

The driver should then return control to the main menu and start the whole process over again.

DATA I/O COMPUTER REMOTE CONTROL STANDARD

Table B summarizes Data I/O's Computer Remote Control standard. The first column shows the ASCII characters that need to be sent to the programmer to initiate the command listed in the second column. Note that some commands have hexadecimal characters preceding them, representing additional information to be passed to the programmer. For example, the] command (Select Extended Function) has a two-digit extended function code preceding it. This extended code is represented by two capital H letters.

Table B. Data I/O Computer Remote Control Standard - Rev. A

ASCII CHAR.	COMMAND MEANING	CORRECT RESPONSE FROM PROGRAMMER	29B	29A	19/17	100A	120A/ 121A	22	60
BREAK	Abort binary transfer	> CR LF	*	*	*	*	*	*	*
ESCAPE	Abort command	> CR LF	*	*	*	*	*	*	*
H ! CR	Binning control	> CR LF	*6			*6			*6
% CR	Handler start	> CR LF (9)	*6			*6			*6
& CR	Enter insert parts mode	NONE					*		
/ CR	# errors /# sockets	HHHH > CR LF					*		
HHHH : CR	Set begin device address	> CR LF	*	*	*		*4	*	
HHHH ; CR	Set block size	> CR LF	*	*	*		*4	*	
HHHH < CR	Set begin RAM address	> CR LF	*	*	*	*	*4	*	
= CR	Disable timeout	> CR LF	*	*	*	*	*	*	
HHHH > CR	Shuffle RAM data	> CR LF	*	*	*			*	
HHHH ? CR	Split RAM data	> CR LF	*	*	*			*	
HHHH @ CR	Select family	> CR LF	*1	*1	*1	*1	*	*	*
HH A CR	Select translation format	> CR LF	*2	*2	*2	*2	*2	*2	
B CR	Blank test	> CR LF	*	*	*	*	*	*	*
C CR	Input compare	> CR LF (7)	*	*	*	*	*	*	
D CR	Select odd parity	> CR LF	*	*	*			*	*
E CR	Select even parity	> CR LF	*	*	*			*	*
F CR	Error status inquiry	HHHHHHHH > CR LF	*	*	*			*	
G CR	Software configuration	HHHH > CR LF (8)	*	*	*	*	*	*	*
H CR	No operation	> CR LF	*	*	*	*	*	*	*
I CR	Input	> CR LF (7)	*	*	*	*	*	*	*
J CR	Set 1 stop bit	> CR LF	*	*	*			*	*
K CR	Set 2 stop bits	> CR LF	*	*	*			*	*
L CR	Load from parts	> CR LF	*	*	*	*	*	*	*
HH M CR	Select record size	> CR LF	*	*	*			*	
N CR	Select no parity	> CR LF	*	*	*			*	*
O CR	Output	...hh...> CR LF (7)	*	*	*	*	*	*	*
P CR	Program	> CR LF	*	*	*	*	*5	*	*
Q CR	Swap nibbles	> CR LF	*	*	*			*	
R CR	Respond device	hHHH/H/H > CR LF (8)	*	*	*	*	*	*	*
S CR	Sumcheck RAM	HHHH > CR LF (8)	*	*	*	*	*5	*	*
T CR	Illegal bit test	> CR LF	*	*	*	*	*5	*	*
HH U CR	Set nulls	> CR LF	*	*	*		*	*	*
V CR	Verify	> CR LF	*	*	*	*	*5	*	*
HHHH W CR	Set address offset	> CR LF	*	*	*	*	*	*	
X CR	Error code inquiry	HH > CR LF (8)	*	*	*	*	*	*	*
Y CR	Parity error	HHHH > CR LF	*	*	*			*	
Z CR	Escape remote control	NONE	*	*	*	*	*	*	*
[CR	Family/pinout inquiry	HHHH > CR LF	*1	*1	*1		*	*	*
\ CR	RAM - RAM block move	> CR LF	*	*	*			*	
HH] CR	Select extended function	...hh...> CR LF	*3	*3	*3	*3	*3	*3	*3
∧CR	Clear all RAM	> CR LF	*	*			*5	*	

NOTES: H = HEXADECIMAL CHARACTER * = COMMAND SUPPORTED h = CHARACTER SOMETIMES PRESENT
CR = CARRIAGE RETURN LF = LINE FEED (present if U command null count is ≠FF)

1) Valid only with paks which utilize this feature.
2) See programmer manual for valid translators available.
3) See pak/programmer manual for available extended functions.
4) Applies to 121A only.
5) May be preceded by two-digit number. (See manual).
6) Only valid with handler connected.
7) Response occurs after data transmission with proper termination.
8) In the model 100A, the given response is preceded by CR LF.
9) Response occurs after start signal from handler.

The third column shows the exact response that the programmer will send back (assuming that the command was successfully executed). Remember that the LINE FEEDS and NULLS are dependent upon the null count, which is set using the U command (Set Nulls). This is discussed in the Computer Remote Control Driver section of this application note.

Notice that some command responses have information preceding the response character (>). This information is represented by capital H and lower case h letters. For example, the O command (Output) will output an entire data file before transmitting the response character.

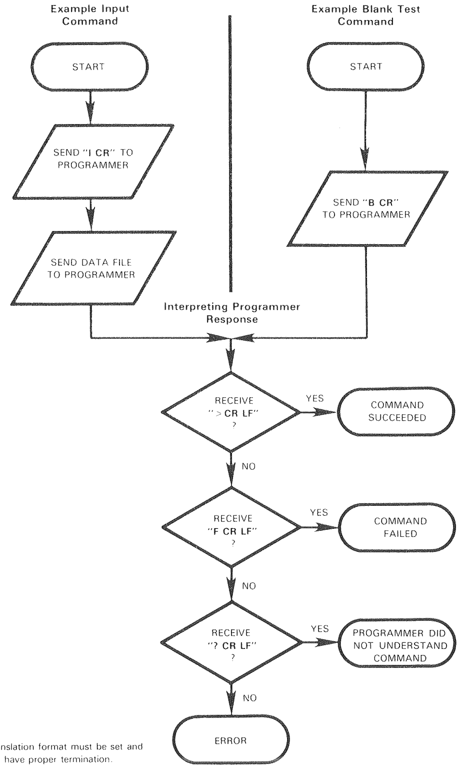
The rest of the figure shows exactly which commands are supported by which Data I/O device programmers. An asterisk means that the command *IS* supported in computer remote control, and a blank means that it is *NOT* supported.

Some important additional notes and comments are located at the bottom of the page.

COMMAND EXECUTION EXAMPLES

The following flowchart shows the chain of events which must happen to implement both the INPUT (I) and the BLANK TEST (B) commands on a Data I/O programmer. (The other commands operate similarly.)

Command Execution Examples



Example Input Command
Example Blank Test Command
START
START
SEND "I CR" TO PROGRAMMER
SEND "B CR" TO PROGRAMMER
SEND DATA FILE TO PROGRAMMER
Interpreting Programmer Response
RECEIVE "> CR LF" ?
YES
COMMAND SUCCEEDED
NO
RECEIVE "F CR LF" ?
YES
COMMAND FAILED
NO
RECEIVE "? CR LF" ?
YES
PROGRAMMER DID NOT UNDERSTAND COMMAND
NO
ERROR

*Proper translation format must be set and data must have proper termination.

NOTES—COMPUTER REMOTE CONTROL

1) Upon powering up, the programming parameters are set to various default values. For instance, most programmers set the translation format to MOS technology format (code 81). Consult your programmer manual for more information on default parameters for your programmer.

2) The programmer sends the ">" response character upon entering remote control.

3) All Data I/O programmers mentioned in this application note support both hardware (RTS,CTS) and software (XON,XOFF) handshaking.

4) The models 60A, 60H, 29B, 29A, and Series 22 all support select code FC from the keyboard
Press: SELECT F C START

One ASCII character will turn on the remote control and another character will turn off remote control. Upon seeing a remote-off character, the programmer will not respond to any other ASCII characters except the defined remote-on ASCII character. Upon receiving the remote-on character the programmer will respond as it normally does in Computer Remote Control.

This select code has many applications. One example would be to connect multiple programmers in parallel in a daisy-chain fashion and define different remote-on codes for each programmer. The remote-off characters could be the same for all programmers. One could "talk" to each programmer individually, depending on the remote-on character that was sent. All other programmers would be in the remote-off state and would be unaffected. This would allow one central system to control multiple Data I/O programmers.

NOTE: Remote-on and remote-off characters should be characters that are not normally used in other programmer communications.

5) SYSTEM 19 AND SYSTEM 17 ONLY - The input command (I) requires 16 extra characters to be sent after transmitting the end-of-text character. The programmer will not respond until these extra characters are sent. Also, a greater-than-50 ms delay is required between sending the "I" character and actually starting to send data.

NOTE: The "<" command (Set begin RAM address) will clear any previously set block sizes (";" command).

EXTENDED FUNCTIONS AVAILABLE IN COMPUTER REMOTE CONTROL

Extended functions are special commands which are Data I/O programmer/pak specific. Each extended function has a two-digit hexadecimal code associated with it. These functions are accessed or executed (or both) by sending the extended function code followed by a right bracket "]" (ASCII character 93) and a carriage return (CR) to the programmer. As is the case with commands, some extended functions require additional information to be sent to the programmer after the carriage return. Some extended functions respond with information preceding the response character ">". Because extended functions are programmer/pak specific, your manual will have to be consulted for the details of each extended function.

MOSPAK

CE] (CR) - Set reject count to default
CF] (CR) - Set single pulse reject count
EF] (CR) - Display MOSPAK configuration number

UNIPAK 2

BC] (CR) - Disable Electronic Identifier test
BD] (CR) - Enable Electronic Identifier test
C3] (CR) - Access specific Family/Pinout options
CC] (CR) - Examine Family/Pinout code
CD] (CR) - Display device's Electronic Identifier Array
CE] (CR) - Set reject count to default
CF] (CR) - Set single pulse reject count
EF] (CR) - Display UNIPAK 2 configuration number

GANGPAK

CC] (CR) - Examine Family/Pinout code
CD] (CR) - Display device's Electronic Identifier array
E0] (CR) - EEPROM erase routine
E1] (CR) - Select set size (1 - 8)
E2] (CR) - Select word size (multiple of 8)
E3] (CR) - Display checksum of desired device
EF] (CR) - Display GANGPAK configuration number

LOGICPAK

CE] (CR) - Set option attributes
E1] (CR) - Enable terminal mode
E2] (CR) - Receive PALASM source
E3] (CR) - Transmit PALASM source
E4] (CR) - Assemble PALASM source
E5] (CR) - Enter reject count option
E6] (CR) - Enter verify option
E7] (CR) - Enter security fuse option
E8] (CR) - Set number of logic fingerprint cycles
E9] (CR) - Enter starting vector and test sum
EA] (CR) - Display fuse pattern
EB] (CR) - Receive JEDEC data
EC] (CR) - Transmit JEDEC data
ED] (CR) - Display sum-check of fuse data
EE] (CR) - Edit fuse by number
EF] (CR) - Display LOGICPAK configuration number

120A/121A

18] (CR) - Set V_{CC} levels during verify
20] (CR) - Performs EEPROM test
22] (CR) - Convert to 16-bit programmer
23] (CR) - Set number of verify passes
25] (CR) - Select Electronic Identifier options
CC] (CR) - Electronic Identifier Family/Pinout codes
CD] (CR) - Electronic Identifier code inquiry
DF] (CR) - Inquire device status for all 20 sockets

SERIES 22

A4] (CR) - Clear all RAM
B2] (CR) - System configuration number
B7] (CR) - View select functions
B8] (CR) - View family codes
B9] (CR) - Display test
BC] (CR) - Disable Electronic Identifier
BD] (CR) - Enable Electronic Identifier
C1] (CR) - Calibration
C3] (CR) - Programming algorithm option selection
CC] (CR) - Display Family/Pinout code
CD] (CR) - View Electronic Identifier
CE] (CR) - Select normal reject count
CF] (CR) - Set one pulse reject count
F4] (CR) - Set nibble mode
F5] (CR) - Set binary base
F6] (CR) - Set octal base
F7] (CR) - Set hexadecimal base
F8] (CR) - Disable nibble mode
FE] (CR) - Power down save

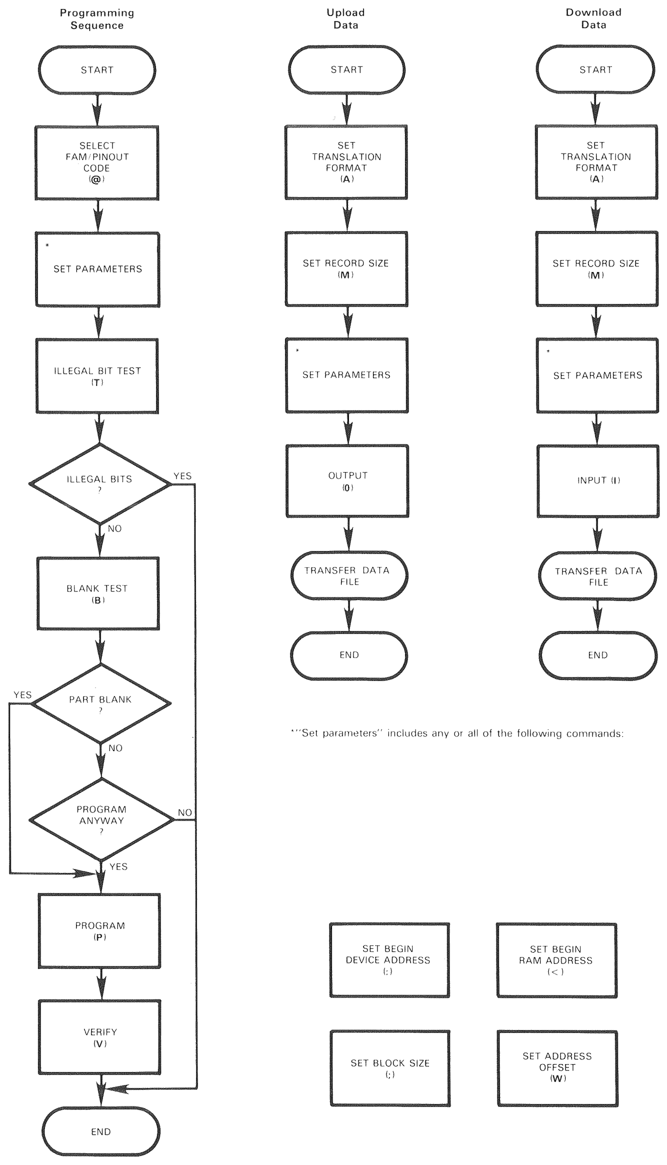
60A AND 60H

A2] (CR) - Fill fusemap
C2] (CR) - Select JEDEC mode
C3] (CR) - Define handler error count
C4] (CR) - Select display format
C5] (CR) - Define handler device counter
C8] (CR) - Enable/disable underblow/overblow output format
CA] (CR) - Select number of test passes
CE] (CR) - List vectors on/off
EA] (CR) - Output fuse map
E5] (CR) - Reject option
E6] (CR) - Testing mode
E7] (CR) - Security fuse enable/disable
E8] (CR) - Define Logic Fingerprint cycles
E9] (CR) - Start/end vector for Logic Fingerprint
EB] (CR) - Input data in JEDEC format
EC] (CR) - Output data in JEDEC format
FE] (CR) - Save parameters in non-volatile RAM

COMMAND SEQUENCES

The following flowcharts show the common sequence of commands used to both program a device and transfer data files.

Common Command Sequences



*"Set parameters" includes any or all of the following commands:

INTERFACE CONNECTIONS TO A DATA I/O PROGRAMMER

An RS232C serial port interface is used to connect Data I/O programmers to other host systems. This requires installing the proper serial interface cabling and setting the correct operational parameters. Data I/O programmers use the following pin numbers and signal descriptions:

PIN #	SIGNAL NAME	DESCRIPTION
1	Ground	Provides a safety ground connection
2	Send Data	Transmits data within RS232C voltage levels
3	Receive Data	Accepts data within RS232C voltage levels
4	Request to Send	This line is normally held high by the programmer. It is dropped to inhibit data transmission from a remote source.
5	Clear to Send	A high level allows the programmer to transfer data. A low level inhibits data transfer. If left unconnected, this line is pulled high internally.
6	Data Set Ready	A high level indicates that the programmer is ready.
7	Signal Ground	This line provides a common signal connection to the RS232C remote source.
8	Carrier Detect	If used, this line is sampled by the programmer and is high when the modem detects a carrier signal. A low inhibits data transfer by the programmer.
20	Data Ready	A high level indicates that the data terminal is ready.

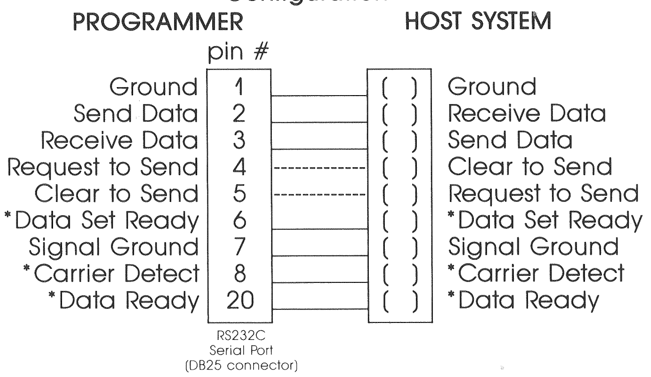
NOTE: Refer to the programmer manual for more detailed signal descriptions and pin locations.

Interface Cabling

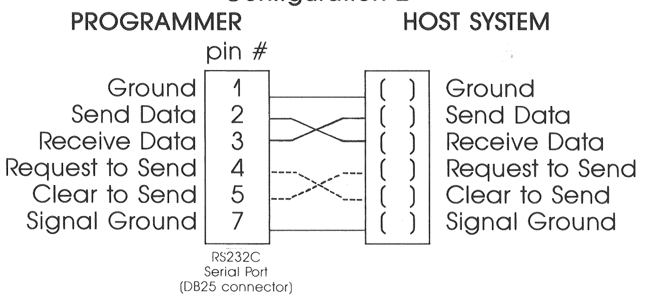
Two typical examples of interface cabling are shown below. A modem typically uses configuration 1 while a terminal typically uses configuration 2. Refer to the operation manuals for more detailed signal descriptions and pin locations.

Figure 1. RS232C Serial Interface Configurations

Configuration 1


PROGRAMMER
HOST SYSTEM
pin #
Ground 1 ( ) Ground
Send Data 2 ( ) Receive Data
Receive Data 3 ( ) Send Data
Request to Send 4 ( ) Clear to Send
Clear to Send 5 ( ) Request to Send
*Data Set Ready 6 ( ) *Data Set Ready
Signal Ground 7 ( ) Signal Ground
*Carrier Detect 8 ( ) *Carrier Detect
*Data Ready 20 ( ) *Data Ready
RS232C Serial Port (DB25 connector)

Configuration 2


PROGRAMMER
HOST SYSTEM
pin #
Ground 1 ( ) Ground
Send Data 2 ( ) Send Data
Receive Data 3 ( ) Receive Data
Request to Send 4 ( ) Request to Send
Clear to Send 5 ( ) Clear to Send
Signal Ground 7 ( ) Signal Ground
RS232C Serial Port (DB25 connector)

*These connections are sometimes required when interfacing to a modem.

NOTES:
1) All undesignated pins remain open.
2) For applications that require hardware handshaking, the programmer's Request To Send (pin 4) and Clear To Send (pin 5) lines should be utilized. (This is shown with dotted lines above.) If hardware handshaking is not required, the programmer's Clear To Send line (pin 5) is pulled up internally.

Setting Serial Port Parameters

Before Data I/O programmers can communicate with other systems via the serial port, three parameters must be set; PARITY, STOP BITS, and BAUD RATE. *THESE PARAMETERS MUST BE THE SAME FOR BOTH SYSTEMS.*

PARITY - Set to one of the following: "odd", "even" or "none."

STOP BITS - Set to either 1 or 2.

BAUD RATE - Set to one of the following: 110, 150, 300, 600, 1200, 2400, 4800, or 9600. Some programmers have additional settings.

NOTE: It may be easier to configure the Data I/O programmer to your particular host system.

For Data I/O models 121A, 120A, 100A, 29B, 29A and System 19—Baud Rate, Parity and Stop Bits are set with switches internal to the programmer. See Figure 2 for the switch locations and settings available for your programmer.

NOTE: For the Data I/O model 100A, the data lock and port I/O switches must be set to the "off" position.

For Data I/O models 60A, 60H and Series 22—Baud Rate, Parity and Stop Bits are set using select codes from the front keyboard.

PARAMETER	SELECT CODE
PARITY	DB
STOP BITS	DC
BAUD RATE	DA

For Data I/O models 60A and 60H the key sequence is:

SELECT *CODE* START

(use SCROLL to pick setting)

START

For the Series 22 the key sequence is:

SELECT *CODE*

(Use SELECT to pick setting)

START

SETTING UP AND RUNNING YOUR DATA I/O DRIVER

The remote mode is initiated by depressing the following keys on your Data I/O programmer:

For Data I/O models 29B and 29A press:

SELECT F 1 START START

For Data I/O models 60A, 60H, System 19 and Series 22 press:

SELECT F 1 START

For Data I/O models 100A,120A and 121A press:

SELECT 0 9 START

For Data I/O System 17 press:

MODE SELECT

until the remote control LED lights then hold down the

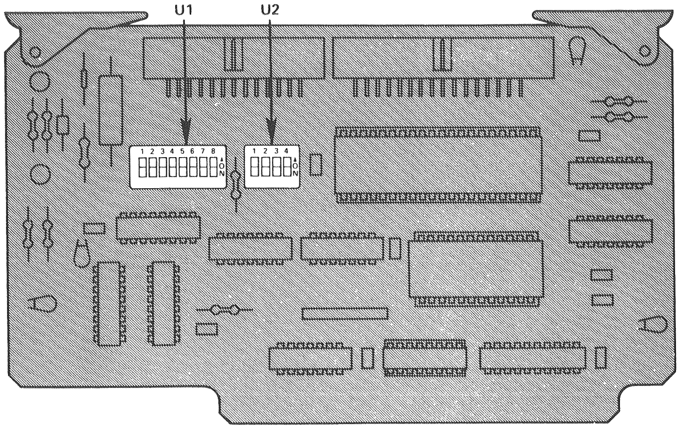
MODE SELECT key and press START

The serial ports should be connected with a cable as described under INTERFACE CONNECTIONS TO A DATA I/O PROGRAMMER. To verify that the programmer and the computer are properly communicating, you could use a short subroutine similiar to the VERIFYING PROGRAMMER CONNECTIONS SUBROUTINE. This is documented in the EXAMPLE COMPUTER REMOTE CONTROL DRIVER section of this application note. This subroutine first sets the null count to 00 using the U command (Set Nulls). It then uses the H command (no operation) and makes sure that the programmer responds with a "greater than" sign (>) followed by a CARRIAGE RETURN and a LINE FEED. Be sure to set the proper baud rate, parity, and stop bits for each system before verifying communication.

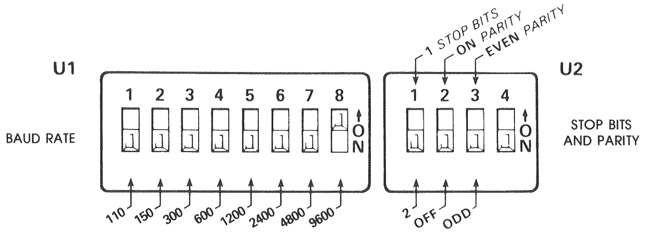
NOTE: For a better idea of exactly how Computer Remote Control works, you can connect a terminal to the programmer. This is described in the INTEFACE CONNECTIONS TO A DATA I/O PROGRAMMER section of this application note. You could then enter remote control (as described above), send ASCII characters to the programmer from the terminal, and watch what characters the programmer responds with. (See Table B for a summary of the various commands and their associated ASCII characters.) Models 29B, 29A, 121A, 120A, 60A, 60H and Series 22 all display action symbols in the remote mode. When the programmer is executing a command, the action symbol will indicate operation. (The action symbol looks similiar to a clock with the hands turning clockwise when a command is executing.)

Figure 2. Setting Parameters

MODELS 120A AND 121A
KEYBOARD DISPLAY & SERIAL I/O CARD
701-1829-XXX

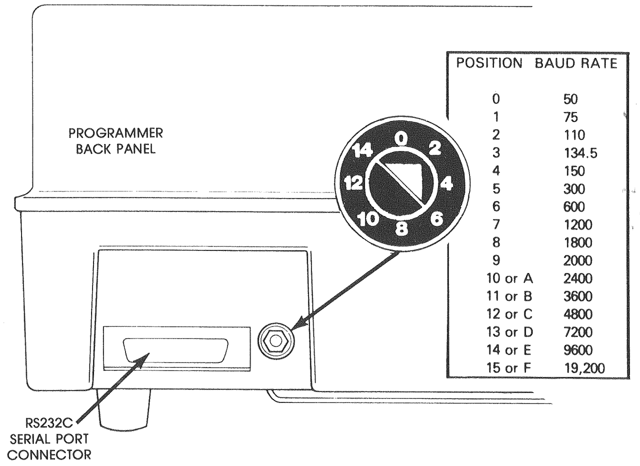


PARITY, STOP BITS, BAUD RATE SETTINGS

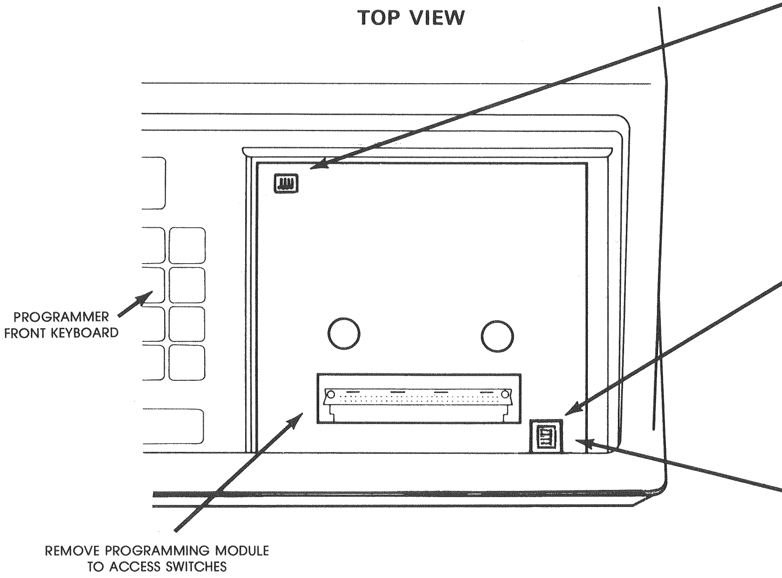


NOTE: Models 120A and 121A require removal of top cover.

100A, 29B, 29A AND SYSTEM 19
BAUD RATE SETTINGS



POSITION	BAUD RATE
0	50
1	75
2	110
3	134.5
4	150
5	300
6	600
7	1200
8	1800
9	2000
10 or A	2400
11 or B	3600
12 or C	4800
13 or D	7200
14 or E	9600
15 or F	19,200


TOP VIEW
PROGRAMMER FRONT KEYBOARD
REMOVE PROGRAMMING MODULE TO ACCESS SWITCHES

SYSTEM 19
PARITY/STOP BIT SETTINGS

OFF	ON
Odd Parity	Even Parity
Parity Off	Parity On
2 Stop Bits	1 Stop Bit
Spare	Spare

STATUS SWITCH

100A PARITY/STOP BIT SETTINGS
(SET DATA LOCK & PORT I/O TO OFF)

OFF	ON
Data Lock Off	Data Lock On
No Parity	Even Parity
2 Stop Bits	1 Stop Bit
Port I/O	DCU I/O

STATUS SWITCH

29B, 29A
PARITY/STOP BIT SETTINGS

OFF	ON
Odd Parity	Even Parity
No Parity	Parity
Two Stop Bits	One Stop Bit
Spare	Spare

STATUS SWITCH

EXAMPLE COMPUTER REMOTE CONTROL DRIVER

The following pages contain an example driver written for a Data I/O programmer. This driver could be copied directly or could be easily expanded to meet your particular needs.

The first section is called VERIFY PROGRAMMER CONNECTIONS. It verifies that the programmer and the computer are communicating. If so, the program continues to the MAIN MENU section. This portion of the program displays a menu with all of the available commands on it. It waits for the user to make a selection and then jumps to the appropriate INDIVIDUAL COMMAND subroutine. Each INDIVIDUAL COMMAND subroutine then sets a string variable called COMMAND to the ASCII characters corresponding to that command.

The OUTPUT COMMAND subroutine sends the characters contained in the string COMMAND to the programmer. The INPUT CHARACTER subroutine waits for a programmer response, inputs that response and returns to the OUTPUT COMMAND subroutine. The OUTPUT COMMAND subroutine will then interpret that response and print out an explanatory message based on whether the command succeeded or failed. The whole process then starts over again.

There are two other subroutines worth mentioning, the TRANSFER PROGRAMMER RAM TO DATA FILE subroutine and the TRANSFER DATA FILE TO PROGRAMMER RAM subroutine. These are used for uploading and downloading data files.

The following mnemonics are used:
CR = Carriage Return - ASCII char. 13 (∧M)
LF = Line Feed - ASCII char. 10 (∧J)
XON = ASCII char. 17 (∧Q)
XOFF = ASCII char. 19 (∧S)
ESC = ASCII char. 27 (∧[ )

The following global variables are used:

COMMAND = A string variable which gets set in the INDIVIDUAL COMMAND subroutines to the ASCII characters to be output to the programmer for that particular command sequence.

RESPONSE = A string variable containing the ASCII characters received from the programmer.

CHARACTER = A string variable containing the last character received from the programmer.

STATUS = A string variable which gets set to either "succeeded," "failed," or "question," depending on the programmers response.
(The programmer will respond with either >, F, or ?)

The example driver is divided into nine sections:

1) VERIFYING PROGRAMMER CONNECTIONS PURPOSE: To verify that the computer and the programmer are talking to each other properly. The U command (set nulls) and the H command (no command) are used to do this.

GLOBAL VARIABLES:	RESPONSE
LOCAL VARIABLES:	None
CALLED FROM:	None
CALLS:	None

2) MAIN MENU PURPOSE: To display a menu containing the commands available, then to input the user's selection and go to the proper INDIVIDUAL COMMAND subroutine.

GLOBAL VARIABLES:	None
LOCAL VARIABLES:	SELECTION = The selected command
CALLED FROM:	None
CALLS:	"Individual Command" subroutines "Data Transfer" subroutines "Programming Part" subroutine

3) INDIVIDUAL COMMAND SUBROUTINES PURPOSE: To set a string variable called COMMAND to the ASCII characters corresponding to that command and then go to the OUTPUT COMMAND subroutine. Optionally, after returning from the OUTPUT COMMAND subroutine, an explanatory message can be displayed, depending on the string variable STATUS.

GLOBAL VARIABLES:	COMMAND; STATUS
LOCAL VARIABLES:	None
CALLED FROM:	Main menu "Programming" subroutine
CALLS:	"Output Command" subroutine

4) TRANSFER PROGRAMMER RAM TO DATA FILE SUBROUTINE PURPOSE: To transfer data files from the programmer's RAM to the computer.

GLOBAL VARIABLES:	None
LOCAL VARIABLES:	FILENAME = A string variable which contains the name of the data file to transfer to
CALLED FROM:	Main menu
CALLS:	None

5) TRANSFER DATA FILE TO PROGRAMMER RAM SUBROUTINE PURPOSE: To transfer data files from the computer to the programmer's RAM.

GLOBAL VARIABLES:	COMMAND
LOCAL VARIABLES:	FILENAME = A string variable which contains the name of the data file to transfer from FILELINE = A string variable containing one line of FILENAME
CALLED FROM:	Main menu
CALLS:	"Output Command" subroutine (for reading response)

6) PROGRAMMING PART SUBROUTINE PURPOSE: To illustrate how INDIVIDUAL COMMAND subroutines can be strung together to form larger subroutines. This particular subroutine combines the ILLEGAL BIT CHECK, BLANK CHECK, PROGRAM and VERIFY subroutines into one subroutine.

GLOBAL VARIABLES:	STATUS
LOCAL VARIABLES:	None
CALLED FROM:	Main menu
CALLS:	The following INDIVIDUAL COMMAND subroutines: ILLEGAL BIT CHECK subroutine BLANK CHECK subroutine PROGRAM subroutine VERIFY subroutine

7) OUTPUT COMMAND SUBROUTINE PURPOSE: To output the ASCII characters contained in the string COMMAND and to call the INPUT CHARACTER subroutine, which inputs the programmer response. The OUTPUT COMMAND subroutine then interprets the response and prints out a response message.

GLOBAL VARIABLES:	COMMAND; STATUS; CHARACTER; RESPONSE
LOCAL VARIABLES:	ERROR# = A string variable containing the number of the last error
CALLED FROM:	"Individual command" subroutines "Transfer data file to programmer RAM" subroutine
CALLS:	"Input character" subroutine

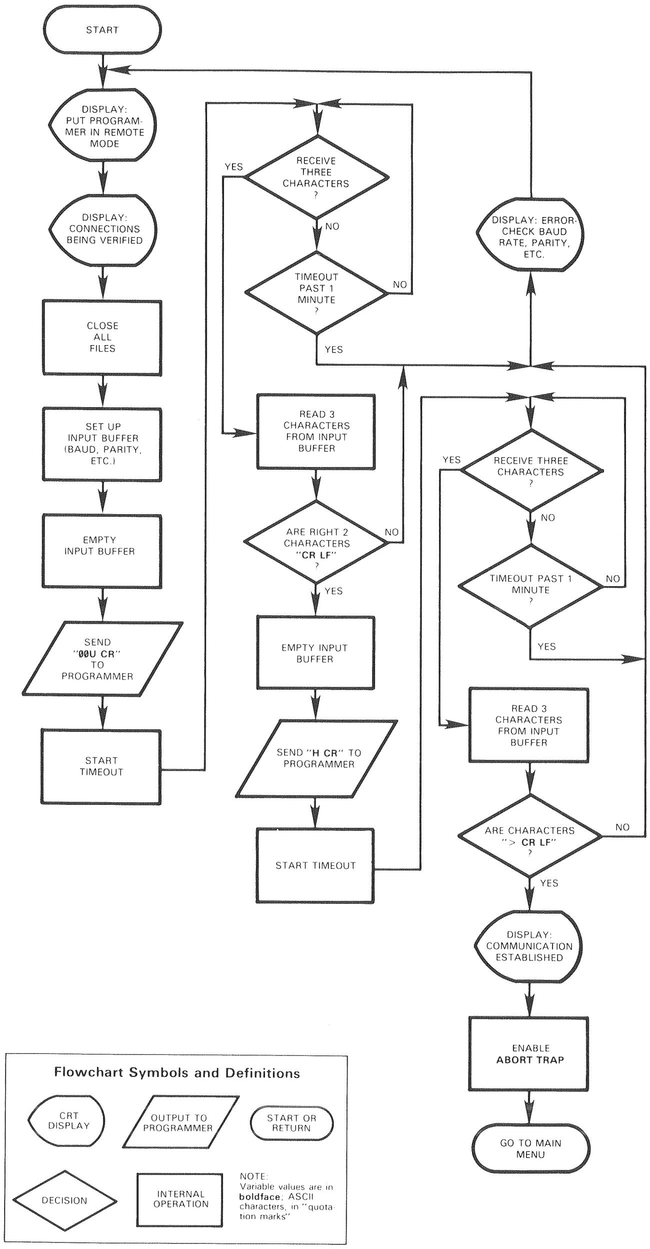
8) INPUT CHARACTER SUBROUTINE PURPOSE: To wait for a response from the programmer and then input one character into the string CHARACTER. The string RESPONSE keeps track of the programmer's total response.

GLOBAL VARIABLES:	CHARACTER; RESPONSE
LOCAL VARIABLES:	None
CALLED FROM:	"Output command" subroutine
CALLS:	None

9) ABORT TRAP PURPOSE: To abort the current command and return to the main menu.

GLOBAL VARIABLES:	None
LOCAL VARIABLES:	None
CALLED FROM:	Abort trap key
CALLS:	None (returns to main menu)

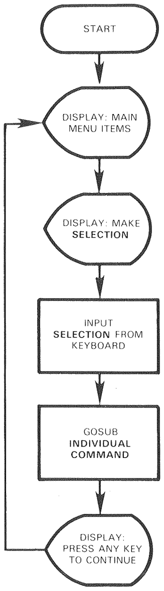
Verify Proper Connections


START
DISPLAY: PUT PROGRAMMER IN REMOTE MODE
DISPLAY: CONNECTIONS BEING VERIFIED
CLOSE ALL FILES
SET UP INPUT BUFFER (BAUD, PARITY, ETC.)
EMPTY INPUT BUFFER
SEND "00U CR" TO PROGRAMMER
START TIMEOUT
RECEIVE THREE CHARACTERS ?
YES
NO
TIMEOUT PAST 1 MINUTE ?
NO
YES
READ 3 CHARACTERS FROM INPUT BUFFER
ARE RIGHT 2 CHARACTERS "CR LF" ?
NO
YES
EMPTY INPUT BUFFER
SEND "H CR" TO PROGRAMMER
START TIMEOUT
DISPLAY: ERROR-CHECK BAUD RATE, PARITY, ETC.
RECEIVE THREE CHARACTERS ?
YES
NO
TIMEOUT PAST 1 MINUTE ?
NO
YES
READ 3 CHARACTERS FROM INPUT BUFFER
ARE CHARACTERS "> CR LF" ?
NO
YES
DISPLAY: COMMUNICATION ESTABLISHED
ENABLE ABORT TRAP
GO TO MAIN MENU
CRT DISPLAY
OUTPUT TO PROGRAMMER
START OR RETURN
DECISION
INTERNAL OPERATION

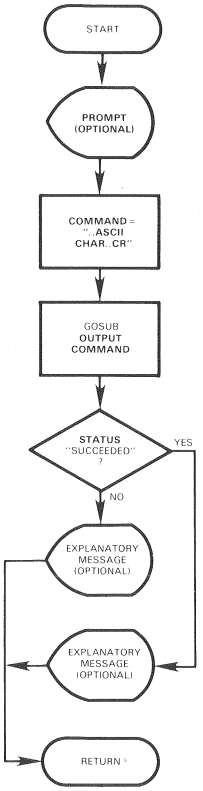
Flowchart Symbols and Definitions

NOTE: Variable values are in **boldface**; ASCII characters, in "quotation marks"

Main Menu

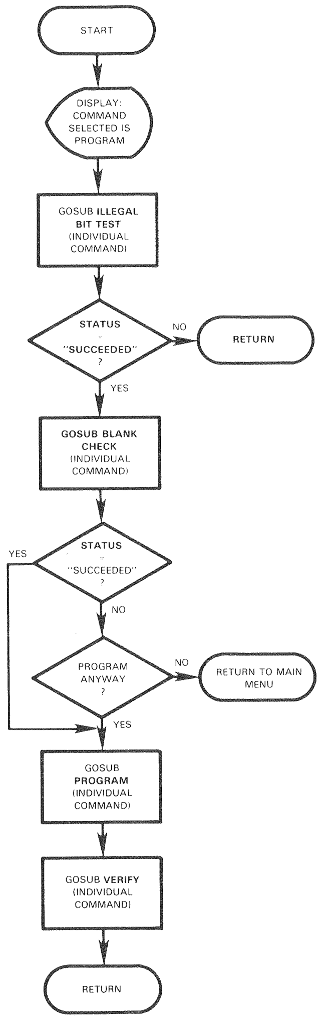


Individual Command Subroutines (General Form)

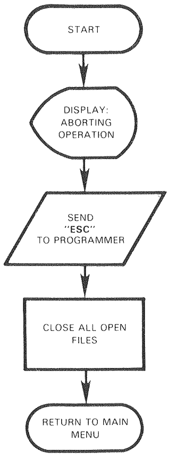

START
PROMPT (OPTIONAL)
COMMAND = "..ASCII CHAR..CR"
GOSUB OUTPUT COMMAND
STATUS "SUCCEEDED" ?
YES
NO
EXPLANATORY MESSAGE (OPTIONAL)
EXPLANATORY MESSAGE (OPTIONAL)
RETURN

NOTE: If you have further questions about Computer Remote Control or questions in general about any Data I/O product, contact your nearest Data I/O support engineer. Our offices are listed on the back of this publication.

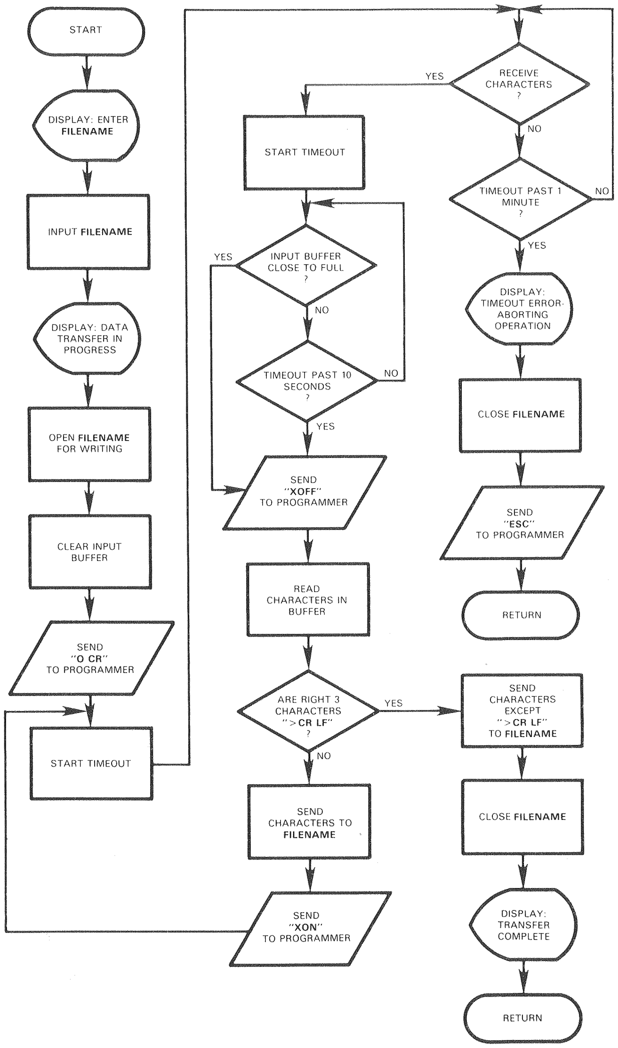
"Programming Part" Subroutine


START
DISPLAY: COMMAND SELECTED IS PROGRAM
GOSUB ILLEGAL BIT TEST (INDIVIDUAL COMMAND)
STATUS "SUCCEEDED" ?
NO
RETURN
YES
GOSUB BLANK CHECK (INDIVIDUAL COMMAND)
STATUS "SUCCEEDED" ?
YES
NO
PROGRAM ANYWAY ?
NO
RETURN TO MAIN MENU
YES
GOSUB PROGRAM (INDIVIDUAL COMMAND)
GOSUB VERIFY (INDIVIDUAL COMMAND)
RETURN

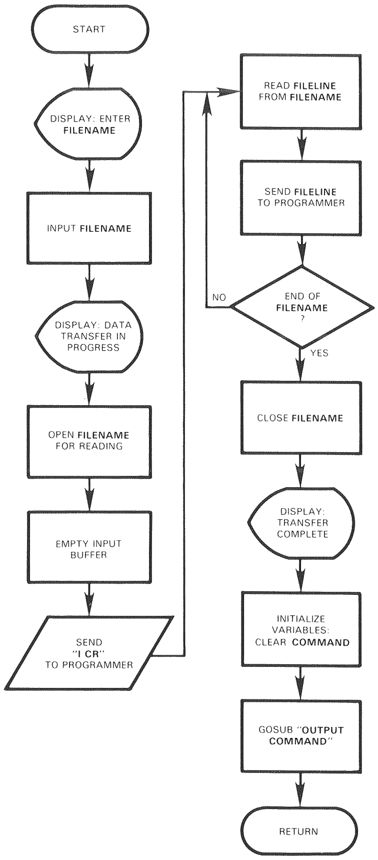
Abort Trap



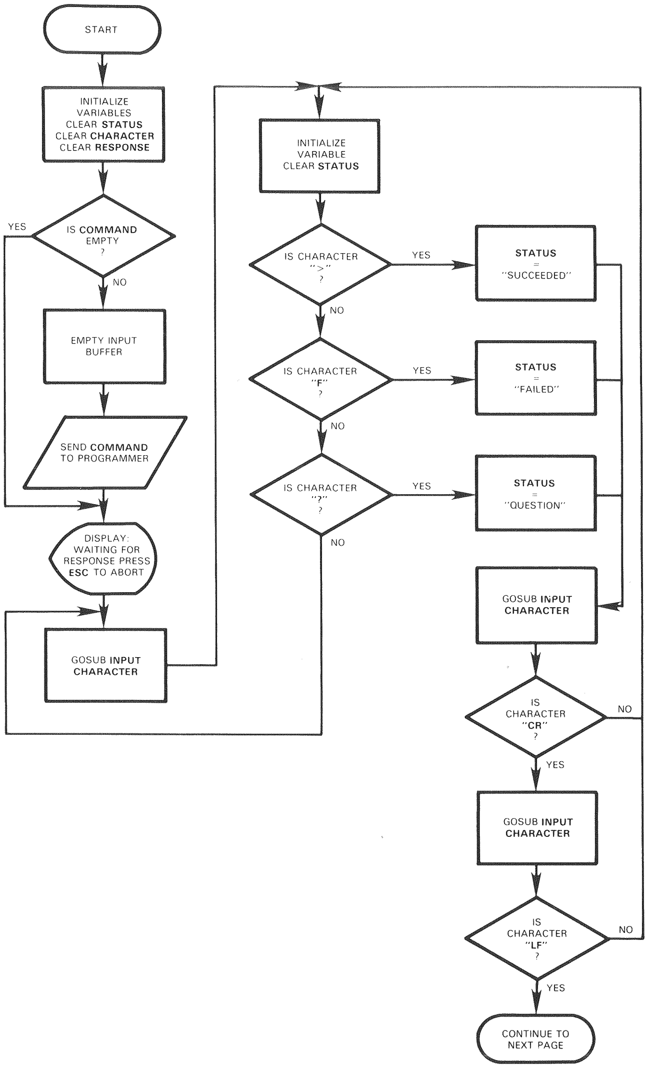
Transfer Programmer RAM to Data File Subroutine


START
DISPLAY: ENTER FILENAME
INPUT FILENAME
DISPLAY: DATA TRANSFER IN PROGRESS
OPEN FILENAME FOR WRITING
CLEAR INPUT BUFFER
SEND "O CR" TO PROGRAMMER
START TIMEOUT
RECEIVE CHARACTERS ?
YES
NO
START TIMEOUT
TIMEOUT PAST 1 MINUTE ?
NO
YES
INPUT BUFFER CLOSE TO FULL ?
YES
NO
TIMEOUT PAST 10 SECONDS ?
NO
YES
SEND "XOFF" TO PROGRAMMER
READ CHARACTERS IN BUFFER
ARE RIGHT 3 CHARACTERS ">CR LF" ?
YES
NO
SEND CHARACTERS TO FILENAME
SEND "XON" TO PROGRAMMER
SEND CHARACTERS EXCEPT ">CR LF" TO FILENAME
CLOSE FILENAME
DISPLAY: TRANSFER COMPLETE
RETURN
DISPLAY: TIMEOUT ERROR-ABORTING OPERATION
CLOSE FILENAME
SEND "ESC" TO PROGRAMMER
RETURN

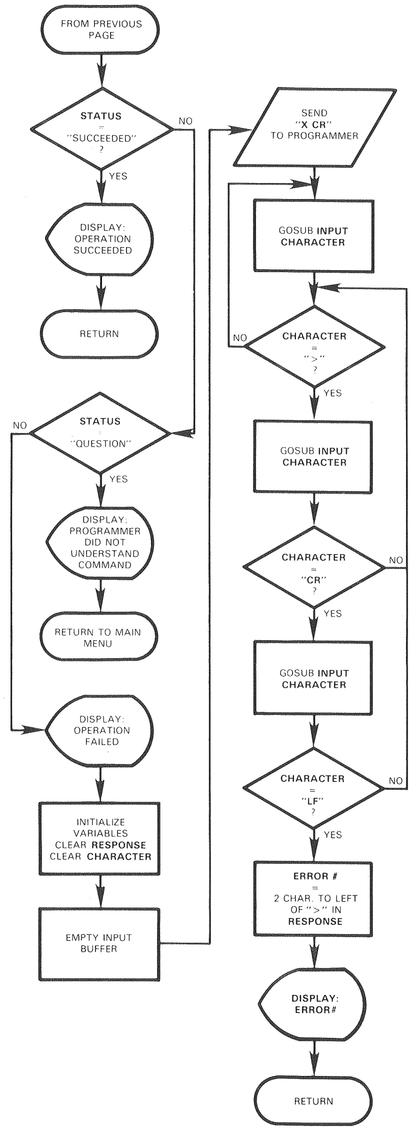
Transfer Data File to Programmer RAM Subroutine


START
DISPLAY: ENTER FILENAME
INPUT FILENAME
DISPLAY: DATA TRANSFER IN PROGRESS
OPEN FILENAME FOR READING
EMPTY INPUT BUFFER
SEND "I CR" TO PROGRAMMER
READ FILELINE FROM FILENAME
SEND FILELINE TO PROGRAMMER
END OF FILENAME ?
NO
YES
CLOSE FILENAME
DISPLAY: TRANSFER COMPLETE
INITIALIZE VARIABLES: CLEAR COMMAND
GOSUB "OUTPUT COMMAND"
RETURN

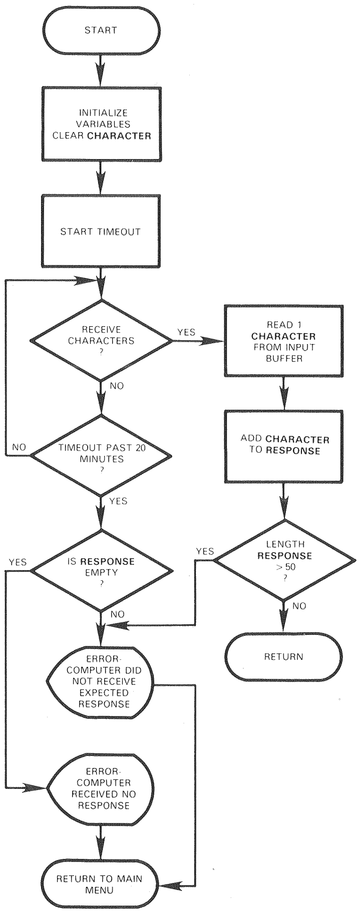
"Output Command" Subroutine



"Output Command" Subroutine (Continued)



Input Character Subroutine


START
INITIALIZE VARIABLES CLEAR CHARACTER
START TIMEOUT
RECEIVE CHARACTERS ?
YES
NO
READ 1 CHARACTER FROM INPUT BUFFER
ADD CHARACTER TO RESPONSE
TIMEOUT PAST 20 MINUTES ?
NO
YES
IS RESPONSE EMPTY ?
YES
NO
LENGTH RESPONSE > 50 ?
YES
NO
RETURN
ERROR: COMPUTER DID NOT RECEIVE EXPECTED RESPONSE
ERROR: COMPUTER RECEIVED NO RESPONSE
RETURN TO MAIN MENU

U.S. SALES OFFICES

Data I/O	(206) 881-6444
Western Region	(714) 662-1182 (408) 727-0641
Eastern Region	(603) 889-8511
Central Region	(214) 235-0044
Alabama	(205) 881-9298
Alaska	(206) 881-8857
Arizona	(602) 263-6022
Arkansas	(214) 644-5010 (713) 956-0837 (512) 327-7033 (918) 665-7788
California, Northern	(408) 727-0641
California, Southern	(714) 662-1182
Colorado	(303) 321-4246
Connecticut	(603) 889-8511
Delaware	(301) 321-1411
Florida	(305) 421-4989 (305) 678-6809
Florida, Northern	(205) 881-9298
Georgia	(404) 424-1931
Hawaii	(206) 881-8857
Idaho	(801) 534-0500
Illinois, Northern	(312) 945-8700
Illinois, Southern	(314) 839-0800
Indiana	(317) 244-7867
Iowa	(319) 373-0200
Kansas	(314) 839-0800 (816) 229-4007
Kentucky, Western	(317) 244-7867
Kentucky, Eastern	(412) 487-3801
Louisiana	(214) 644-5010 (713) 956-0837 (512) 327-7033 (918) 665-7788
Maine	(603) 889-8511
Maryland	(301) 321-1411
Massachusetts	(603) 889-8511
Michigan	(313) 474-7320 (616) 323-2416
Minnesota	(612) 835-2414
Mississippi	(205) 881-9298
Missouri	(314) 839-0800 (816) 229-4007
Montana	(801) 534-0500
Nebraska	(314) 839-0800 (816) 229-4007
Nevada	(602) 263-6022
New Hampshire	(603) 889-8511
New Jersey, North	1-800-858-5803
New Jersey, South	(609) 662-5222
New Mexico	(505) 842-6633
New York, Metro	1-800-858-5803
New York, Upstate	(315) 446-0220
North Carolina	(919) 852-6000
North Dakota	(612) 835-2414
Ohio	(513) 426-5551 (216) 729-0190
Oklahoma	(214) 644-5010 (713) 956-0837 (512) 327-7033 (918) 665-7788
Oregon	(503) 646-9966
Pennsylvania, Western	(412) 487-3801
Pennsylvania, Eastern	(609) 662-5222
Puerto Rico	(603) 889-8511
Rhode Island	(603) 889-8511
South Carolina	(919) 852-6000
South Dakota	(612) 835-2414
Tennessee	(404) 424-1931 (205) 881-9298
Texas	(214) 644-5010 (713) 956-0837 (512) 327-7033 (918) 665-7788
Texas, El Paso Co.	(505) 842-6633
Utah	(801) 534-0500
Vermont	(603) 889-8511
Virginia	(703) 385-0600 (609) 662-5222
Washington, Western & Spokane	(206) 881-8857
Washington, Tri Cities & Vancouver	(503) 646-9966
Washington D.C.	(703) 385-0600 (609) 662-5222
West Virginia	(412) 487-3801
Wisconsin	(414) 784-7736
Wyoming	(303) 321-4246

INTERNATIONAL SALES OFFICES AND REPRESENTATIVES

THE NETHERLANDS	(20) 622866
GERMANY	(6182) 3088/89
JAPAN	(03) 574-0211
AUSTRALIA (Warburton Franki)	
Adelaide	3567333
Brisbane	(07) 277-0222
Melbourne	(03) 795 901199
Newcastle	049 528722
Sydney	(02) 648-1711
Perth	(09) 277-7000
Lidcombe	(02) 647-2266
AUSTRIA	(222) 827474
BELGIUM	
Simac Electronics	(2) 2192451
BRAZIL (Systronics)	
Sao Paulo	(11) 247-5588
Rio de Janeiro	(21) 283-2036
CANADA (Allan Crawford Associates)	
Mississauga, Ontario	(416) 890-2010
Ottawa, Ontario	(613) 722-7682
St. Laurent (Montreal)	(514) 731-8564
Calgary, Alberta	(403) 230-1341
Burnaby, B.C.	(604) 294-1326
CHINA (Dorado Company)	
U.S.A.	(206) 583-0000
Hong Kong	0-4986401/2
Beijing	89-0621, ext. 1324888
Shanghai	219103 or 210464
DENMARK	
ITT Multikomponent A/S	(2) 451822
FINLAND	
Instrumentarium Elektroniikka	(0) 5284312
FRANCE	
M.B. Electronique	(3) 9568131
GERMANY, FEDERAL REPUBLIC OF	
Instrumatic Electronic GmbH	(89) 85802-0
GREECE	
Eltronics Ltd.	(1) 7249511-15 or 7210669
HONG KONG	
Eurotherm (Far East) Ltd.	5-546391
INDIA (Transmarketing Private Ltd.)	
Bombay	022 4921874, 4920320 or 4926044
Bangalore	560-046
ISRAEL	
Telsys Ltd.	(3) 494891-5
ITALY (Sistrel SpA)	
Rome	(6) 5915551
Milan	(2) 6120129 or 6181893
KOREA	
Elcom Systems Inc.	(2) 555-5222 or 557-3836
MALAYSIA	
GEA Technology PTE, Ltd.	65 2729412
MEXICO	
Christensen, S.A.	546-25-95 or 546-29-55
NETHERLANDS	
Simac Electronics	(40) 533725
NEW ZEALAND (Anitech)	
Auckland	(9) 504-458
NORWAY	
Teleinstrument A/S	(2) 789460
PORTUGAL	
Decada Espectral	(1) 2103420
SINGAPORE	
GEA Technology PTE, Ltd.	(65) 2729412
SOUTH AFRICA	
Electronic Building Elements	(12) 46-9221/7
SPAIN (Unitronics)	
Madrid	242 52 04
SWEDEN	
Macrotek AB	(8) 7330220
SWITZERLAND	
Instrumatic AG	(1) 7231410
Instrumatic SA	(22) 360830
TAIWAN	
Sertek International, Inc.	(2) 713-5435
THAILAND	
Dynamic Supply Engineering	(2) 3928532 or 3925313
UNITED KINGDOM	
Microsystem Services	(494) 41661

Addresses on this list subject to change without notice.

DATA I/O