201 EPROM Programmer

Operator's Manual

DATA I/O

Received 7-7-88

201 EPROM PROGRAMMER

User Notes

984-0030-005 OCTOBER 87

Engineering Part No.	Model No.	Version No.	Manual Part No.
990-1913-011	201EP64L	V04	981-0030-001 and up
990-1913-012	201EP64H	V04	981-0030-001 and up

NOTE: The "H" in the above model number refers to the high voltage configuration (180—260V) and "L" refers to the low voltage configuration (90—130V).

Introduction

The following information supplements the 201 EPROM Programmer Operator's Manual so that it reflects the latest equipment update. The configuration numbers of the updated equipment and documentation are provided on the back of the title page of this user notes document. Refer to these configuration numbers when calling a Data I/O Service Center regarding service or updates to your equipment.

Also included with this user notes are manual update instructions, an acceptance test procedure, and the newest device list. Follow the manual update instructions provided in the following subsection to update your operator's manual. The acceptance test procedure is provided for customers whose company policy requires that they perform an acceptance test before accepting the equipment. This is a one-time procedure and may be discarded after the equipment has been accepted. Following the acceptance test is a new device list which shows all of the devices that are programmable with the latest version of 201 firmware installed. The device list may be inserted into your operator's manual behind the "Error Messages/Device List" tab.

Manual Update Instructions

The following items are updates to your operator's manual. Mark the changes on your manual pages, where appropriate, and then place the user notes pages in your operator's manual for future reference.

1. If you have just received your 201 EPROM Programmer from the factory, the voltage selector wheel is not installed in the back of the unit. You must install the voltage selector wheel (contained in a bag included with the 201) in order to operate the unit. Pages 1-2 through 1-3 of the 201 Operator's Manual show the procedure for changing the line voltage selection. Perform this procedure to install the voltage selector wheel for the first time, except ignore step 3. You will also need to remove the warning label covering the ac power connector in order to operate your 201.

2. Check the back of your 201 unit for the part number. If the part number has a 990- prefix, cross out the engineering part number on **page i** of your operator's manual and write in the part number showing on the back of your unit.

3. On **page 1-5** of the operator's manual the line fuse information is described. This information has changed since the printing of the manual. The 201 now comes supplied with two fuses which can be used for voltage ranges 90 through 260. These fuses are 1/4 x 1-1/4 inch fuses and the specifications are shown in the first line of the table below. If you are operating the 201 on a line voltage between 220V and 260V, you can also use a 5 x 20 millimeter fuse (commonly available in Europe) which fits the specifications shown in the second line of the table below.

Line Voltage	Line Fuse Rating Current	Voltage	Type	Data I/O Part Number
90 - 260	1.0A	250V	T (Slow-blow)*	416-3010
220 - 260	0.5A	250V	T (Slow-blow)	—

* Littlefuse type 313, Bussman type MDA

4. The 201's data RAM is now filled with zeroes instead of "FF" characters upon power up. Mark the following changes on your operator's manual pages.

   On page **2-33** of your operator's manual, change the second sentence in the first paragraph on the page to read: "Data following the deleted data byte is shifted down to fill in the deleted address and **00** is filled in the last byte of RAM."

   On page **2-34** the example shows "FF" being filled in address FFFF after the deletion (at the bottom of the right-hand box). Change the "FF" to "**00**."

   On page **2-35** of your operator's manual, change the second sentence in the first paragraph on the page to read: "Data following the deleted block is shifted down to fill in the deleted range and **00** is filled in the vacated RAM addresses at the end of memory."

   On page **3-64** of your operator's manual, change the last sentence in the first paragraph on the page to read: "**00** is filled in the vacated addresses at the end of RAM."

   On page **3-65** of your operator's manual, change the bottom line of the example to show "**00 00 00 00**" instead of "FF FF FF FF" as the last four bytes of data in memory.

5. On **page 4** of the operator's manual, the specifications for the Power Requirements show an operating voltage of "110." Change this operating voltage to **100**.

6. The Terminal Remote Control feature is now an option for the 201. Computer Remote Control is the standard remote control operation language provided with your 201. If you purchased the Terminal Remote Control option with your 201, you received separately an EPROM which must be installed in your 201 in order to use Terminal Remote Control. With the EPROM you also received an installation manual which provides instructions for installing the EPROM. The Terminal Remote Control operation is explained in your operator's manual in the section titled "Terminal Remote Control." When you install the Terminal Remote Control option (EPROM), Computer Remote Control will no longer be available.

   If you did not order the Terminal Remote Control option, which allows you to run the 201 from a terminal connected directly to the RS232 port, but would like to purchase it, contact your local Data I/O Sales Representative. A list of representatives is provided in the back of your 201 EPROM Programmer Operator's Manual.

7. If your version of PROMlink does not support the 201 specifically and you have specified the Model 22 as the programmer type, you may experience some difficulty when programming using the manufacturer and device type selected from the PROMlink menus. If you experience difficulty using the menu selected manufacturer and device type, type in the family/pinout code listed in the 201 Device List instead of selecting the device type from the PROMlink menus.

8. In Computer Remote Control the **Clear RAM** command (explained on page 4-18) now uses the currently selected beginning RAM address and block size to determine the block of data RAM to fill. Data RAM is filled from the beginning RAM address (or zero if no beginning RAM address is selected) to the block size. If no block size is selected, then the 201 uses either the word limit of the selected device or the end of RAM, whichever is smaller, to determine the block size.

9. Do not use the electronic ID feature when using the **Intel EEPROM 2816A**. This device does not support the electronic ID feature and use of it with this device may cause device data to be modified.

10. You can now display the 201's firmware configuration number on the front panel display. You will need to know the firmware configuration number of your 201 when contacting Data I/O service personnel. To display the firmware configuration number on the front panel, perform the following procedure.

Displaying the Firmware Configuration Number

To cause the programmer to display its firmware configuration number on the front panel, hold down the up-scroll and ENTER keys when you press the power switch to the ON position. The programmer displays

```
SELF TESTING
```

with advancing decimal points on the lower portion of the display. When the programmer displays

```
DISPLAY TEST
```

press any front panel key twice. The programmer will then display

```
   FIRMWARE
SUMCHECK = HHHH
```

where "HHHH" is the firmware sumcheck, or configuration number, of the programmer's firmware. To begin normal programmer operation, turn the programmer power off and then on again.

Certificate of RFI/EMI Compliance with VDE 0871 Level B

Data I/O certifies that the 201 EPROM Programmer complies with the Radio Frequency Interference (RFI) requirements of VDE 0871 level B as required in West German postal regulation number vfg 1046/1987, page 1943.

Data I/O further certifies that the German Postal Service (DBP) has been notified of Data I/O's intention to market this equipment in West Germany. Data I/O acknowledges that the DBP reserves the right to retest this equipment to verify compliance with the regulation.

201 EPROM PROGRAMMER

Acceptance Test Procedure

Data I/O Corporation warrants to the original purchaser of the product described by the 201 EPROM Programmer Operator's Manual that the product was fully functional to the extent of its specification at the time of shipment from the factory. Data I/O further certifies that the test equipment used to test the product was calibrated to standards that are traceable to the National Bureau of Standards as appropriate.

This procedure is provided for customers whose company policy requires that an inspection test be performed before the unit may be accepted.

INTRODUCTION

Your 201 EPROM Programmer was tested both electrically and mechanically before it was shipped and was carefully packaged to prevent shipping damage. It should arrive free of any defect, without marks or scratches, and in perfect operating condition. However, carefully inspect the unit for any damage that may have occurred in transit. If you note any damage, file a claim with the carrier and notify your nearest Data I/O Service Center. A list of Service Centers is located at the back of the 201 EPROM Programmer Operator's Manual.

The 201 EPROM Programmer acceptance test consists of four voltage tests of the programmer's socket pins and the successful completion of the self test.

WARNING

The procedures described in this document are for qualified engineering personnel only. Do not attempt to perform these procedures unless you are qualified to do so.

In order to perform the acceptance test procedure, you will need a 3-1/2 digit digital multimeter (DMM). The dcV accuracy of the DMM must be ±(0.3% + 1 digit) or better.

ACCEPTANCE TEST PROCEDURE

The acceptance test procedure consists of powering up the programmer in a special voltage test mode, performing four voltage tests while in the voltage test mode and then allowing the programmer to complete its self test. To perform the acceptance test, complete the following steps in the order presented.

1. Install the voltage selector and check the line fuse as described in the Getting Started section of your operator's manual (see the subsection Verifying/Changing the Operating Voltage).

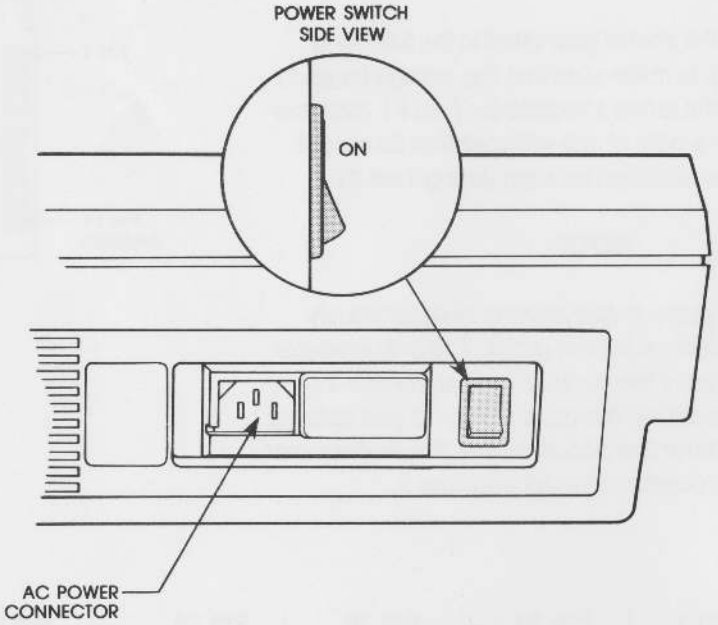
   NOTE

   The voltage selector was not shipped installed in your unit, as described in the operator's manual, but was shipped attached to the 201's chassis in a bag. The voltage selector must be installed (using the instructions in your operator's manual but skipping step 3) before the 201 can be operated.

2. Plug the ac power cord into the rear of the programmer (see illustration) and into a power outlet.

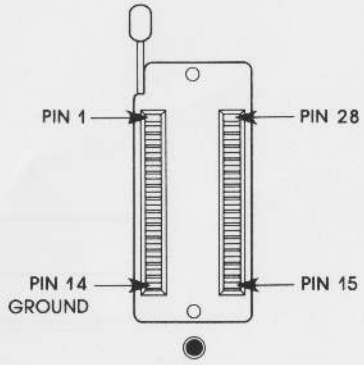
3. While holding down the "A" and "7" keys, press the power switch on the back of the programmer to the ON position (see illustration).

4. After the programmer displays `VOLTAGE TEST #1`, release the "A" and "7" keys. The programmer will continue to display `VOLTAGE TEST #1` and decimal points will advance across the second line of the display. The advancing decimal points indicate that the programmer is operating in the mode shown on the top line of the display.


POWER SWITCH
SIDE VIEW
ON
AC POWER
CONNECTOR

5. Connect the DMM to ground at pin 14 of the socket (see illustration).

6. Probe each of the socket pins listed in the following table and check to make sure that the voltage for each pin falls within the range specified for Test #1 (first row of table). Make a note of any voltages that do not fall within the range specified for a pin during Test #1.



NOTE

Touching the DMM probe to two sockets simultaneously may cause an overcurrent error to occur. The only indicator that an overcurrent error has occurred is that the non-Vol and Voh voltages levels will measure too low. If you suspect that an overcurrent error has occurred, turn the programmer off and repeat this procedure starting with step 3.

	Pin 1	Pin 28	Pin 26	Pin 24	Pin 23	Pin 22
Test #1	21V ±0.5V	5V ±0.2V	Vol*	Vol*	Vol*	Vol*
Test #2	Voh**	Voh**	5V ±0.2V	Voh**	Voh**	21V ±0.5V
Test #3	Vol*	6V ±0.2V	Vol*	Voh**	21V ±0.5V	Voh**
Test #4	Vol*	Voh**	6V ±0.2V	12.5V ±0.5V (ID voltage)	Vol*	10.3V-12.8V (erase voltage)

* Vol is less than 0.8V

** Voh is greater than 2.4V

7. Press the "ENTER" key on the front of the programmer to proceed to Voltage Test #2. The display will read VOLTAGE TEST #2 and the decimal points will continue to advance across the second line of the display.

8. Probe the same socket pins (listed in the previous table) and check to make sure that the voltage of each pin falls within the range specified for Test #2. Make a note of any voltages that do not fall within the specified range.

9. Press the "ENTER" key to proceed to Voltage Test #3. Probe the same pins and check to make sure that the voltage of each pin falls within the range specified for Test #3. Make a note of any voltages that fall outside of the specified range.

10. Press the "ENTER" key to proceed to Voltage Test #4. Probe the socket pins listed in the table and make sure that the voltages fall within the ranges specified for Test #4. Make a note of any voltage that falls outside of the range specified in the table.

11. After completing all of the voltage tests (1 through 4), press the "ENTER" key again. The programmer will complete the self test. Upon completion of the self test the programmer will display

```
 SELF TEST OK
DATA I/O 201 N
```

where "N" is the version number of the 201 software. Then the programmer will display

```
LOAD FROM MASTER
```

Once the self test is complete and the programmer displays `LOAD FROM MASTER`, the acceptance test procedure is completed. Turn the programmer off and then on again before using it to program devices. See the 201 EPROM Programmer Operator's Manual for instructions on powering up the programmer for normal operation.

CAUTION

If you do not turn the programmer off to reset it before programming a part, the programmer will remain in the special self test mode. Attempting a device operation, such as programming, in the special self test mode will be unsuccessful and could potentially damage the device.

If the self test did not complete successfully, or one or more of the pins probed during the four voltage tests did not indicate a voltage within the ranges specified in the table, contact your nearest Data I/O Service Center. A list of Service Centers is located at the back of the 201 EPROM Programmer Operator's Manual.

DEVICE LIST

INTRODUCTION

The following pages list all of the devices that can be programmed by the 201 EPROM Programmer. Also included are the correct family and pinout codes that correspond to each device. The correct device type must be selected in order for the programmer to load, blank check, program, or verify a device safely and correctly. The device type can be entered either by selecting the correct manufacturer and device part number from the 201's menus or by entering the listed family and pinout codes when prompted. The devices are listed alphabetically by manufacturer, and the devices for each manufacturer are listed in numerical order (as the number reads from left to right).

Since the 201's device support capabilities are updated periodically, this Device List has been shipped to you separately. Please insert the Device List into your operator's manual behind the Error Messages/Device List tab. To obtain device support updates for your 201, call your local Data I/O sales representative. When contacting Data I/O personnel about device support updates, refer to the part numbers listed on the back of the title page of this user notes. These numbers identify the version of your 201 system.

CAUTION

Entry of an invalid family/pinout code, i.e., one other than those listed in this list, can cause unpredictable results at the device socket, which may damage a device. A valid family code and a valid pinout code may be combined to produce an invalid (illegal) combination. The correct combination for your device is published in this list. All family and pinout code combinations not contained in this list are considered "illegal." Data I/O assumes no responsibility or liability for results produced by entry of "illegal" family and pinout code combinations.

Key to Headings and Notes:

Device Part Number: The number assigned by the device manufacturer.

Notes: A letter in this column correspondes to a footnote listed below. The footnote contains special information that pertains to the device.

a This device contains an electronic identifier which can be read by the programmer; the electronic I.D. allows the programmer to automatically identify the device so you do not have to select the device type manually. To cause the programmer to read the electronic identifier of a device, enter "FFFF" when prompted for the family/pinout code. If the device does not support the electronic identifier feature, a "No ID Found" error message will appear and you will have to select the device type in the usual manner.

b This device requires the use of socket adapter which is available from the device manufacturer.

c This device does not support the electronic I.D. feature and use of the electronic I.D. feature (family/pinout code FFFF) with this device may cause device data to be modified.

d Do not program this device using family/pinout code FFFF (electronic identifier feature).

e This device appears as C2716 (instead of 27C16) on the programmer's menu.

f Programming yields on Seeq 2817A date code 8442 cannot be assured.

Family Code: A 2-character number that designates the programming algorithm.

Pinout Code: A 2-character number used to differentiate device types based on pin assignment and array size.

Software Version: A number that specifies the earliest version of the 280 Set Programmer software that will program the device to the manufacturer's latest specifications.

Device Part Number	Notes	Family and Pinout Code		Software Version
Advanced Micro Devices (AMD)				
27128	a	AF	51	V01
27128A	a	C1	51	V01
2716		19	23	V01
2716B	a	C2	23	V04
27256	a	C1	32	V01
2732		19	24	V01
2732A		27	24	V01
2732B	a	C2	24	V04
27512	a	DD	A4	V01
2764	a	AF	33	V01
2764A	a	C1	33	V01
27C256	a	C1	32	V04
27C512	d	DD	A4	V04
2817A		BF	A2	V04
Atmel (ATMEL)				
27256	a	93	32	V04
27C128	a	93	51	V04
27C256	a	93	32	V04
27C512	d	4B	A4	V04
27C513	a	5B	5E	V04
27C515	a	5B	CA	V04
27C64	a	93	33	V04
27HC64/L	a	93	33	V04
27HC641/L	a	90	67	V04
Eurotechnique (EUROTECH)				
2716		19	23	V01
2732		19	24	V01
2764		35	33	V01
27C16	e	19	23	V01
27C256	a	93	32	V04
27C64	a	93	33	V04
Exel Microelectronics (EXEL)				
2816A		B7	23	V01
2817A		BF	A2	V04

AM 27C128
(12.5V PGM)

Device Part Number	Notes	Family and Pinout Code		Software Version
Fujitsu Microelectronics (FUJITSU)				
27128	a	45	51	V01
27128A	a	93	51	V04
27256	a	93	32	V01
2732		19	24	V01
2732A		27	24	V01
27512	a	4B	A4	V04
2764		45	33	V01
27C128	a	45	51	V01
27C256	a	45	32	V01
27C256A	a	93	32	V04
27C256H	a	93	32	V04
27C32A		27	24	V01
27C512	a	4B	A4	V04
27C64		45	33	V01
8516		19	23	V01

Device Part Number	Notes	Family and Pinout Code		Software Version
General Instruments (GI)				
27256	a	93	32	V04
27C128		93	51	V04
27C256	d	93	32	V04
27C512	d	4B	A4	V04
27C513	a	5B	5E	V04
27C515	a	5B	CA	V04
27C64		93	33	V04
27HC64	a	93	33	V04
27HC641	a	90	67	V04
5816		37	23	V01

Device Part Number	Notes	Family and Pinout Code		Software Version
Hitachi (HITACHI)				
27128A	a	93	51	V01
27256	a	93	32	V01
27512	a	4B	A4	V01
27C256	a	93	32	V01
27C64		79	33	V01
462532		19	25	V01
462716		19	23	V01
462732		19	24	V01
48016		33	23	V01
4827128		79	51	V01
482732A		27	24	V01
482764		79	33	V01
63701V0	b	93	32	V03
63701X0	b	27	24	V03
63701Y0	b	93	32	V03
63705V0	b	93	32	V03
637A01V0	b	93	32	V03
637A01Y0	b	93	32	V03
637B01V0	b	93	32	V03
637B01Y0	b	93	32	V03

Device Part Number	Notes	Family and Pinout Code		Software Version
Intel (INTEL)				
27128	a	79	51	V01
27128A	a	93	51	V01
2716		19	23	V01
27256	a	93	32	V01
2732		19	24	V01
2732A	a	27	24	V01
27512	a	4B	A4	V01
27513		5B	5E	V01
2764	a	79	33	V01
2764A	a	93	33	V01
27C128		5C	51	V04
27C256	a	93	32	V01
27C513	a	5E	5E	V04
27C64	a	93	33	V01
2816		37	23	V01
2816A	c	37	23	V01
2816B		BF	96	V04
2817A	a	BF	A2	V04
P27128A	a	5C	51	V02
P27256	a	5C	32	V01
P2732A	a	4D	24	V01
P2764	a	79	33	V03
P2764A	a	5C	33	V01

Device Part Number	Notes	Family and Pinout Code		Software Version
Mitsubishi (MITSU.)				
27128		79	51	V01
2716		19	23	V01
27256	a	93	32	V01
2732		19	24	V01
27512	a	4B	A4	V01
2764		79	33	V01
27C128		79	51	V01
27C256	a	93	32	V01
Mostek (MOSTEK)				
2716		19	23	V01
Motorola (MOTOROLA)				
2532		19	25	V01
68764		25	29	V01
68766		25	29	V01

Device Part Number	Notes	Family and Pinout Code		Software Version
National Semiconductor (NATIONAL)				
2532		19	25	V01
2716		19	23	V01
2732		19	24	V01
2758A		19	22	V01
2758B		19	35	V01
27C128	a	5D	51	V04
27C16		19	23	V01
27C16H		BD	23	V01
27C256	a	5D	32	V01
27C32		19	24	V01
27C32H		BD	24	V01
27C512	a	4C	A4	V04
27C64	a	5D	33	V01
27CP128	a	5D	BB	V04
2816		37	23	V01
9817		BF	A2	V04
9817A		BF	A2	V04

881-6444 manual Jim Barnes DATA I-O / 201 Programmer

Device Part Number	Notes	Family and Pinout Code		Software Version
NEC (NEC)				
27128	a	79	51	V01
2716		19	23	V01
27256	a	45	32	V01
2732		19	24	V01
2732A		27	24	V01
2764	a	79	33	V01
27C256	a	45	32	V03
27C64	a	79	33	V04
Oki Semiconductor (OKI)				
2532		19	25	V01
27128		79	51	V01
27128A		93	51	V04
2716		19	23	V01
27256		93	32	V01
2732		19	24	V01
2732A		27	24	V01
27512		5E	A4	V01
2758		19	22	V01
2764		79	33	V01
27C256		93	32	V01
2816A		B7	23	V04

288 SAMSUNG ? EEPROM — KM2816A —

Device Part Number	Notes	Family and Pinout Code		Software Version
Ricoh (RICOH)				
27C64		79	33	V03
5H32		27	24	V01
Rockwell International (ROCKWELL)				
87C32		27	24	V01
87C64		35	33	V01
Seeq (SEEQ)				
27128	a	79	51	V01
2764	a	79	33	V01
27C256	a	93	32	V01
2816A		B7	23	V01
2817A	f	BF	A2	V04
2817AH		BF	A2	V04
5133	a	79	33	V01
5143	a	79	51	V01
5213		37	23	V01
52B13		37	23	V01
5516A		B7	23	V01
5517A		BF	A2	V04

2804 ← PROMLINK Software
2804A ~~B7 82~~ ←
NOT Supported 7/6/88
Use 2804 Selection for 2804A in PROMLINK

Device Part Number	Notes	Family and Pinout Code		Software Version
SGS-ATES (SGS-ATES)				
2532		19	25	V01
27128A	a	93	51	V04
2716		19	23	V01
27256	a	93	32	V04
2732A		27	24	V01
2764		35	33	V01
2764A		93	33	V04
Signetics (SIGNET.)				
27C256F	a	93	32	V04
27C256	a	5C	32	V04
27C64	a	35	33	V01
27C64A	a	5C	33	V04
27C64AF	a	93	33	V04
P27C256	a	5C	32	V04

Device Part Number	Notes	Family and Pinout Code		Software Version
Texas Instruments (TI)				
2508		19	22	V01
2516		BD	23	V01
2532		31	25	V01
2532A		63	25	V04
2564		31	30	V01
25L32		19	25	V01
27128		79	51	V01
27128A	a	93	51	V04
27256	a	93	32	V04
2732		31	24	V01
2732A		27	24	V01
2732A-HS		63	24	V01
27512	a	4B	A4	V04
2764		79	33	V01
27C128	a	93	51	V04
27C256	a	93	32	V04
27C512	a	4B	A4	V04
27C64	a	93	33	V04
27P32A		63	24	V04

Thomson — See Eurotechnique

Device Part Number	Notes	Family and Pinout Code		Software Version
Toshiba (TOSHIBA)				
24128	a	45	51	V04
24128A	a	5C	51	V04
24256	a	45	32	V01
24256A	a	93	32	V04
24512	a	4B	A4	V01
2464	a	45	33	V04
2464A	a	5C	33	V01
27128	a	79	51	V01
27128A	a	93	51	V04
27256	a	45	32	V01
27256A	a	93	32	V04
2732		19	24	V01
2732A		19	24	V01
27512	a	4B	A4	V04
2764	a	79	33	V01
2764A	a	93	33	V04
323		19	23	V01
54256	a	45	32	V01
54256A	a	93	32	V04
54512		4B	A4	V04
57256	a	45	32	V01
57256A	a	93	32	V04

Device Part Number	Notes	Family and Pinout Code		Software Version
VLSI Technology, Inc. (VTI)				
27C256	a	5D	32	V01
27C64	a	5D	33	V01
Xicor (XICOR)				
2804A		B7	82	V01
2816A		B7	23	V01
2864A		C3	98	V01

~~SAMSUNG~~

SAM SUNG
KM2816AP

USE XICOR 2816A for programming

881-2365
Tom Williams @ EAEL/Brown called DATA I/O. 10/21/88

Data I/O has made every attempt to ensure that the information in this document is accurate and complete. However, Data I/O assumes no liability for errors, or for any damages that result from use of this document or the equipment which it accompanies.

Data I/O reserves the right to make changes to this document without notice at any time.

Received 7-7-88
SN: BB82439
H/W: E1
F/W: 4.0 CRC

ORDERING INFORMATION

When ordering this manual, use Part Number 981-0030-001.
Applies to 201 Engineering Part Number ~~950-0130-001~~ and up.

990-1913-011

Safety Summary

General safety information for operating personnel is contained in this summary. In addition, specific WARNINGS and CAUTIONS appear throughout this manual where they apply and are not included in this summary.

Definitions

WARNING statements identify conditions or practices that could result in personal injury or loss of life.

CAUTION statements identify conditions or practices that could result in damage to equipment or other property.

Symbols

⚠ This symbol appears on the equipment and it indicates that the user should consult the manual for further detail.

V ∿ This symbol stands for Vac. For example, 120V ∿ = 120 Vac.

Power Source

Check the voltage selector indicator (located inside the rear panel) to verify that the product is configured for the appropriate line voltage.

Grounding the Product

The product is grounded through the grounding conductor of the power cord. To avoid electric shock, plug the power cord into a properly wired and grounded receptacle only. Grounding this equipment is essential for its safe operation.

Power Cord

Use only the power cord specified for your equipment.

Fuse Replacement

For continued protection against the possibility of fire, replace the fuse only with a fuse of the specified voltage, current and type ratings.

Servicing

To reduce the risk of electric shock, do not perform any servicing other than that described in this manual.

Table of Contents

Table of Contents

TERMINAL REMOTE CONTROL (continued)

Table of Contents

Table of Contents

Introduction

The 201 EPROM Programmer is a completely self-contained EPROM programmer that allows you to program both 24- and 28-pin NMOS and CMOS EPROMs and EEPROMs without using special socket adapters or personality modules. You can operate the 201 EPROM Programmer either "locally," using the front panel keys and 32-character display, or "remotely," using a terminal or computer and the RS232 serial I/O port. The 201 can program sets of 8-bit-wide, 16-bit-wide, and 32-bit-wide devices and also includes an electronic identifier mode which reads the electronic identifier of the installed device and automatically selects the correct programming algorithm for the device.

The 201 offers several data translation formats (such as those supplied by Intel, Motorola, and Tektronix) which enable you to transfer files to and accept formatted files from software development systems.

This manual contains the instructions necessary for operating the 201 EPROM Programmer both locally (from the front panel) and remotely. Instructions for setting up the programmer for either remote or local operation are provided in the Getting Started section and operating instructions for each of the modes are provided in separate, tabbed sections.

Manual Contents

A brief outline of the contents of this manual is provided below:

- ☐ INTRODUCTION — This section provides a general description of the programmer, its specifications, warranty and service information and ordering information.
- ☐ GETTING STARTED — This section provides instructions on how to set up the 201 EPROM Programmer for operation and a sample programming session to give you a general idea of how to operate the programmer.
- ☐ FRONT PANEL OPERATION — This section provides detailed instructions on how to program devices using the front panel keys. How to download and upload data in conjunction with the front panel keys is also explained in this section, as well as how to set the communications protocol for the 201.
- ☐ TERMINAL REMOTE CONTROL — This section provides instructions on how to operate the 201 EPROM Programmer from a remote terminal connected to the programmer through the serial interface (RS232) port.
- ☐ COMPUTER REMOTE CONTROL — This section provides a description of the Computer Remote Control command language. This command language can be used to write a software driver that allows you to operate the programmer using a host computer.
- ☐ ERROR MESSAGES/DEVICE LIST — This section provides a list of the error messages displayed on the programmer. The meanings of the error messages are described as well as corrective action that should be taken. The device list provided to you should be placed in this section also.
- ☐ INDEX — This is an alphabetical guide to all major topics covered in this manual.

Specifications

Specifications for the 201 EPROM Programmer are listed below and on the following page.

Functional Specifications

Functional specifications for the 201 are as follows:

- ☐ Data RAM: 64K bytes
- ☐ Translation Formats: Intel Intellec 8/MDS
  Intel MCS-86 Hexadecimal Object
  Motorola Exorciser
  Motorola Exormax
  Tektronix Hexadecimal
- ☐ Keyboard: 16-key hexadecimal, 3-key functional
- ☐ Display: 16 x 2 character alphanumeric display
- ☐ Device Sockets: One 28-pin
- ☐ Input/Output: One serial RS232 compatible
- ☐ Remote Control: Computer Remote Control (CRC)
  Terminal Remote Control (TRC)

Power Requirements

Power requirements for the 201 are as follows:

- ☐ Operating Voltages: ~~110~~ 100, 120, 220, or 240 V, ±10%
- ☐ Frequency Range: 48-63Hz, single-phase
- ☐ Power Consumption: 85VA

Physical and Environmental

Physical and environmental requirements for the 201 are as follows:

- ☐ Dimensions: 32.5 cm x 11.0 cm x 23.7 cm (12.8" x 4.3" x 9.3")
- ☐ Weight: 4.0 kg (8.8 lbs)
- ☐ Operating Temperature Range: 5 to 45° C (41 to 113° F)
- ☐ Storage Temperature Range: -40 to 70° C (-40 to 158° F)
- ☐ Humidity: Up to 90%, noncondensing
- ☐ Operational Altitude: To 10,000 feet

Warranty and Service

Data I/O equipment is warranted against defects in materials and workmanship. The warranty period of one year, unless specified otherwise, begins when you receive the equipment. Refer to the warranty card inside the back cover of this manual for information on the length and conditions of the warranty. For warranty service, contact your nearest Data I/O Service Center.

Data I/O maintains Service Centers throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. This includes not only repairs, but also calibration of all Data I/O products. A list of all Data I/O Service Centers is located at the back of this manual following the index.

Ordering

Orders for Data I/O products must contain the following information:

- ☐ Description of the equipment
- ☐ Quantity of each item ordered
- ☐ Shipping and billing address of firm, including ZIP code
- ☐ Name of person ordering equipment
- ☐ Purchase order number
- ☐ Desired method of shipment

1. Getting Started

Introduction

This section explains how to verify that the operating voltage and line fuse are correct, how to change the operating voltage and line fuse, and how to connect power and power up the unit. Also given in this section is a brief description of the 201 menus and a sample EPROM programming session.

Connecting Power

Before connecting power to the programmer, perform the following checks:

- ☐ Make sure the operating voltage is properly selected on the back panel of the unit.
- ☐ Make sure the correct line fuse is installed.
- ☐ Make sure the unit is properly grounded.

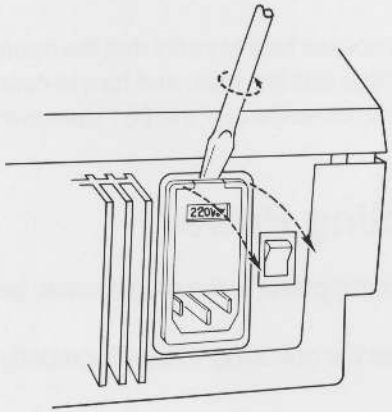
Verifying/Changing the Operating Voltage

The factory has selected the proper operating voltage according to your specification. The current voltage setting is visible through a window in the door that covers the voltage selector wheel, located on the back panel (see illustration). The voltage appearing in the window should be the same as the line voltage on which the programmer will operate. If the voltage that appears in the window is incorrect, change the operating voltage according to the following procedure.

CAUTION

This instrument may be damaged if operated with the wrong operating voltage.

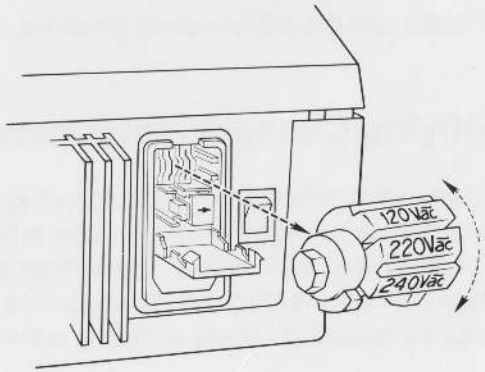
1. Disconnect the power cord if plugged into the unit.
2. Gently pry open the door that covers the voltage wheel selector with a flat-blade screwdriver.



3. Pull the voltage wheel selector out of its slot.

CAUTION

The voltage wheel selector may be damaged if the operating voltage is changed while the wheel is still inserted in its slot.



4. Rotate the voltage selector so that the correct operating voltage is facing out from the back panel of the programmer.

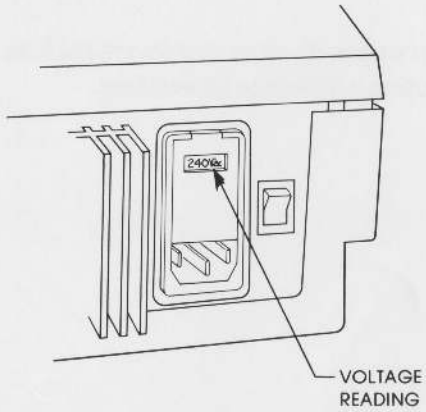
5. Insert the voltage selector into its slot.

NOTE

If you wish to access the line fuse at this time, proceed to step 2 of the next procedure.

6. Snap the door closed.

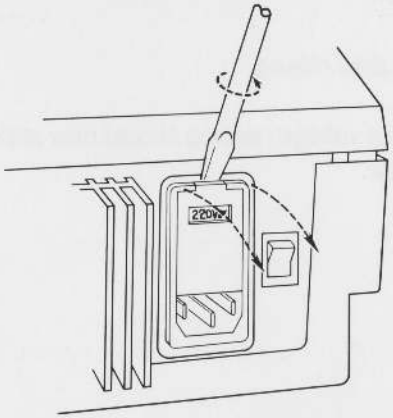
7. The correct voltage reading should now appear in the window.



Verifying/Replacing the Line Fuse

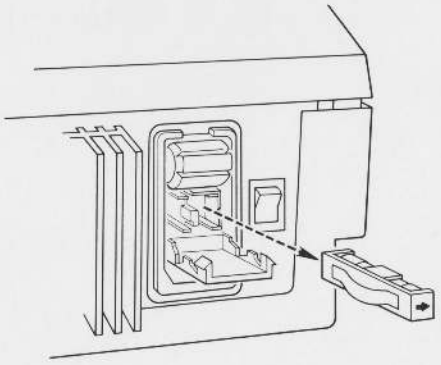
The line fuse is located behind the same door that covers the voltage wheel selector. Perform the following procedure to verify that the line fuse is correct and intact. In the event that the fuse is blown, replace it with one of the correct size.

1. Gently pry open the door that covers the fuse holder using a flat-blade screwdriver.



2. Pull the fuse holder out of its slot.

3. Verify that the fuse is the correct type and value per the following table; if necessary, install the correct fuse into the fuse holder.



Line Voltage	Line Fuse Rating			Data I/O Part Number
	Current	Voltage	Type*	
100	1.0A	250V	T (Slow-blow)	416-3010
120	1.0A	250V	T (Slow-blow)	416-3010
220	0.5A	250V	T (Slow-blow)	416-3048
240	0.5A	250V	T (Slow-blow)	416-3048

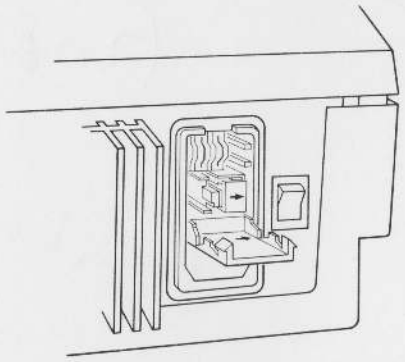
* Littlefuse type 313, Bussman type MDA

CAUTION

For continued protection against the possibility of fire, replace only with a fuse of specified voltage, current, and type ratings.

4. Insert the fuse holder into its slot so that the arrow on the fuse holder points in the same direction as the arrows on the door.

5. Snap the door closed.



PromLink 2.6

Grounding the Programmer

The 201 is shipped with a three-wire power cable. This cable connects the chassis of the programmer to earth ground when connected to a properly grounded three-wire ac receptacle.

WARNING

Continuity of the grounding circuit is vital for the safe operation of the unit. Never operate this equipment with the grounding conductor disconnected.

(HELP)
DATA I/O Customer Resources
(206) 881-6444

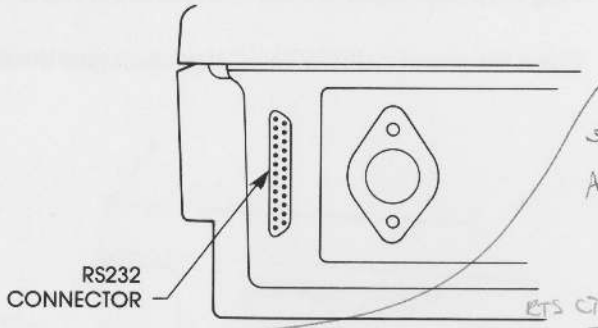
PromLink/2-10
Micky Lynch
1(800)247-5700

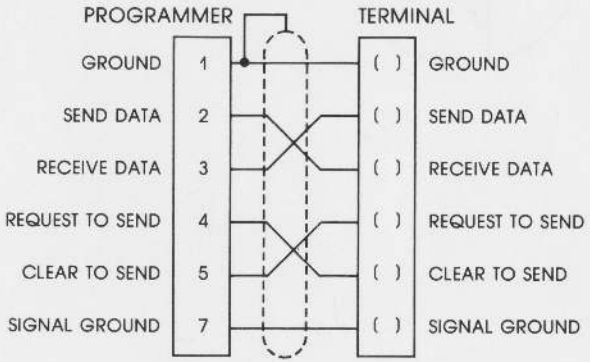
RS232 Port Cable Connections

The 201 is equipped with one serial RS232 compatible I/O port which is located on the back panel of the programmer (see illustration). The RS232 port can be linked to a terminal, computer or other development system in either a handshake or non-handshake mode. The cable connections required to the RS232 connector for each mode are shown in the following illustration.

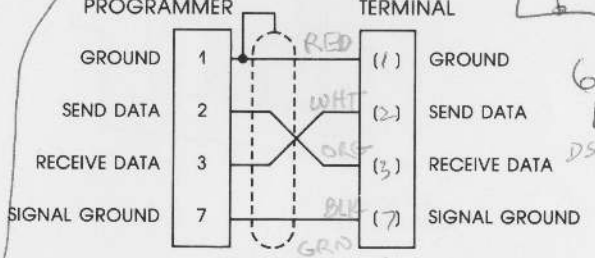
NOTE

For continued compliance with class B limits of VDE 0871, use a properly shielded interconnect cable and peripheral equipment complying with VDE 0871 class B.





Half/Full Duplex, With Handshake



Half/Full Duplex, Without Handshake

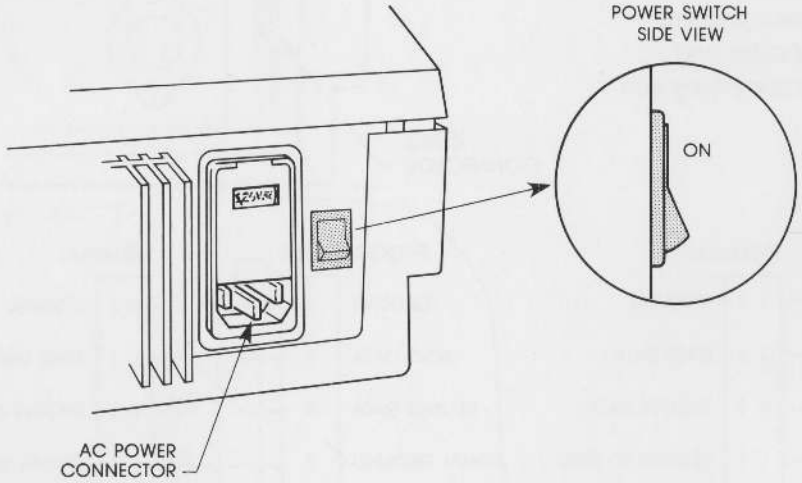
NOTE

All signals are named with respect to the originating unit.

Powering Up the Programmer

To power up the programmer, proceed as follows:

1. Make sure the socket is empty.
2. Plug the power cable into the back panel connector and an ac power receptacle.
3. Press the power switch to the ON position (see the illustration).



When the programmer is turned on, it automatically performs a self test that verifies correct operation of the unit. During execution of the self test, the front panel display shows

```
SELF TESTING
```

with advancing decimal points on the lower portion of the display. Upon completion of the self test, the display shows

```
 SELF TEST OK
DATA I/O 201 N
```

where "N" is the version number of the programmer's software.

The programmer then displays the first main menu item.

```
LOAD FROM MASTER
```

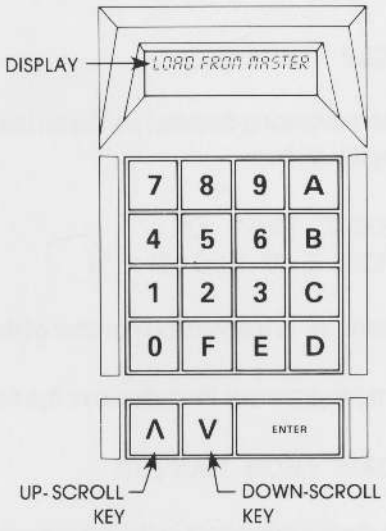
If a device is present in the programmer's socket when the power is switched on, the display reads

```
DEVICE IN SOCKET
 REMOVE DEVICE
```

To correct this condition, remove the device and press ENTER. The self test will then proceed.

The Main Menu Functions

The programmer can perform six basic functions, each of which is presented in the main menu. To display each of these functions, scroll through the main menu by means of the scroll keys shown in the illustration. (Scroll forward with the right-hand key and backward with the left-hand key.) The six main menu functions are:



LOAD FROM MASTER — load from a master device to internal memory.

PROGRAM — program a device from the contents of internal memory.

VERIFY — verify the contents of a device against the contents of internal memory.

BLANK CHECK — perform a blank check on a device.

RS232 — perform RS232 port operations, such as change the port settings, download a file, upload a file, or transfer control of the programmer to a remote computer or terminal.

EDIT — edit the contents of the programmer's data RAM.

Each of the six functions is fully described in the Front Panel Operation section; however, to familiarize you with the basic programming operation, a sample programming session is described in the following pages.

Sample Programming Session

The following steps describe how to program a blank device from a master device. (A master device is a part that has been previously programmed and is used as a "master" to program blank parts.) To perform this programming session, you will need a master device and a blank device. The master device used in the following procedure is an Intel 2716, but could be any device shown on the Device List.

In the following procedure, the blank device is assumed to be of the same type as the master device, although it is possible for the master device to differ from the blank device. For more details on device programming, refer to the Front Panel Operation section of this manual.

1. Make sure the socket of the programmer is empty.

2. Turn on the power switch and wait until the self test is complete and the display reads

```
LOAD FROM MASTER
```

3. Press ENTER to select the LOAD FROM MASTER operation. The display reads

```
LOAD FROM MASTER
 F/P CODE 79/33
```

4. Press the down-scroll key and the display reads

```
LOAD FROM MASTER
     INTEL
```

If the master device you are using in this sample programming session is not an Intel part, press the down-scroll key repeatedly until the display shows the name of the appropriate manufacturer.

NOTE

If you scroll past the desired selection, use the up-scroll key to move backwards through the menu.

5. When the correct manufacturer is displayed, press ENTER to select that manufacturer. The display reads

```
LOAD FROM MASTER
  INTEL  2764
```

NOTE

If you are using a device of a manufacturer other than Intel, the name of that manufacturer and a device number appear on the display.

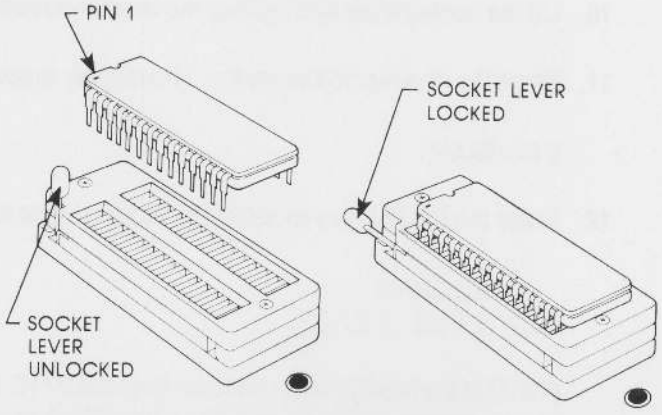
6. Press the up-scroll or down-scroll key repeatedly until the display shows the correct part number of your master device. If you are using an Intel 2716, the display should read

```
LOAD FROM MASTER
  INTEL  2716
```

7. Press ENTER to select the device type. The display reads

```
INSERT  DEVICE
  INTEL  2716
```

8. Insert the master device in the socket by lifting the lever and placing the device in the socket so that the bottom pins of the part are at the bottom of the socket and pin 1 is toward the top of the socket. Push the lever down to lock the device in place (see the illustration).



9. Press ENTER to begin the load operation. The display reads

```
LOAD FROM MASTER
. . . . . .
```

(Advancing decimal points on the lower portion of the display indicate that the load operation is taking place.)

When the load operation is complete the display reads

```
LOAD FROM MASTER
 SUMCHECK = HHHH
```

where "HHHH" is the sumcheck of the device (the sum of all the data bytes in the device expressed as a hexadecimal number). Make a note of the sumcheck so that the part programmed later in this session can be verified.

10. Lift the socket lever and replace the master device with a blank device. Push the socket lever down.

11. Press the down-scroll key twice. The display reads

```
PROGRAM
```

12. Press the ENTER key to select the program function. The display reads

```
   PROGRAM
F/P CODE 19/23
```

(19/23 is the family/pinout code for the Intel 2716, and is now the default family/pinout code. If you used a master device other than an Intel 2716, the family/pinout code of the device you selected is now the default.)

13. Press the ENTER key to select the displayed family/pinout code. The display reads

```
INSERT  DEVICE
F/P CODE 19/33
```

14. Press the ENTER key to initiate the programming operation. During the programming operation the display reads

```
    PROGRAM
..............
```

(Advancing decimal points on the lower portion of the display indicate that the program operation is taking place.)

When the program operation is complete the display reads

```
    PROGRAM
SUMCHECK = HHHH
```

where "HHHH" is the sumcheck of the device. This hexadecimal number should match that displayed in step 9.

15. Lift up the socket lever and remove the programmed device.

This completes the sample programming session.

2. Front Panel Operation

Introduction

This section describes how to blank check, program and verify devices using the front panel keys and display of the 201 EPROM Programmer. Also provided are instructions for editing data stored in the programmer's RAM, setting the RS232 communications protocol, and performing download and upload operations using the front panel keys. The information is divided into the following subsections:

- ☐ CHECKING FOR NON-BLANK DEVICES — Describes how to verify that a device is completely blank (contains no data).
- ☐ PROGRAMMING — Describes load and program operations used to program devices.
- ☐ DOWNLOADING DATA — Describes how to perform the transfer of data from a remote data source (computer or development system) to the programmer's memory for subsequent programming into blank devices.
- ☐ VERIFYING PROGRAMMED DEVICES — Describes how to verify the data programmed into a device against the contents of the programmer's memory.
- ☐ SETTING COMMUNICATIONS PROTOCOL — Describes how to set the RS232 serial port communications protocol for the 201.
- ☐ UPLOADING DATA — Describes how to perform the transfer of data from the programmer's memory to a host computer or other development system.
- ☐ EDITING RAM — Describes how to edit the contents of the programmer's data memory, or RAM.

General Operating Notes

The following notes explain features, functions, and displays that are common to nearly all of the front panel operations.

Family/Pinout Codes and Device Part Numbers

The required programming algorithm and the pin assignments of EPROM and EEPROM devices differ greatly. In order for the programmer to program a specific device, the device to be programmed must be selected on the programmer. Device selection may be made by selecting the manufacturer and part number from the 201 menus, or by entering a specific family/pinout code on the front panel keypad. The operating procedures presented in this section allow you to select the part by either method.

The family/pinout code for each programmable part is contained in the Device List accompanying this manual. In the menus and also in the Device List, the parts for each device manufacturer are listed in order of device size, with multiple devices of the same size listed in numerical order.

Action Display

A special action display appears on the programmer during the execution of certain operations, such as self test, load and program. This display consists of decimal points advancing across the lower portion of the front panel display and indicates that the programmer is executing the operation.

Aborting an Operation

Most operations may be aborted by pressing one of the hexadecimal keys. When an operation is halted in this manner, the programmer beeps and reverts to the currently selected mode; e.g., if a hexadecimal key is pressed during a "load from master" operation, the programmer beeps and the display returns to

```
LOAD FROM MASTER
```

If a hexadecimal key is accepted as input, press a scroll key to abort the operation.

Devices with Electronic Identifiers

To benefit from the electronic identifier feature built into many devices, an electronic identifier mode can be selected that causes the programmer to automatically select the correct programming algorithm and pin configuration. To select the electronic identifier mode when blank checking, loading, programming, or verifying a device, enter "FFFF" when prompted for the family and pinout code. When the electronic identifier mode is selected, the programmer determines the programming algorithm and pin configuration from the device installed in the socket. If you then insert a device that requires a different programming algorithm and pin configuration, you must re-select the electronic identifier mode to continue to use this feature. Not all devices have electronic identifiers. Consult the Device List for information on which devices have electronic IDs.

Error Indicators

If the selected device operation is successfully executed for the installed device, a completion message appears such as

```
    PROGRAM
SUMCHECK = 1AE5
```

A pass signal (two beeps) sounds and the lamp below the socket lights green. However, if execution is unsuccessful for the installed device, a failure signal (three beeps) sounds, the socket lamp lights red, and a failure message appears as in the example below. (Refer to the Error Messages section at the back of the manual for a description of the error messages.)

```
    PROGRAM
BAD INSERTION 36
```

DATA I/O - 881-6444

Checking for Non-blank Devices

A blank check is provided that can be used prior to programming operations to insure that no data has been previously written to a device. The blank check verifies that no data exists at any address of the installed device. Perform the blank check as follows:

NOTE

The 201 Displays shown in this manual are examples only. The devices shown in the displays may be different than the devices appearing on your printed Device List or in the menus on your unit since the number of supported devices changes periodically.

Procedure	Example 201 Displays
1. Scroll to the BLANK CHECK function.	BLANK CHECK
2. Press ENTER.	BLANK CHECK F/P CODE 79/33
3. If the correct family/pinout code is displayed, press ENTER and go to step 7. If the correct code is not displayed, key in the 4-digit family/pinout code for the device (e.g., AF33), and skip to step 7	BLANK CHECK F/P CODE AF/33
or	
press a scroll key and then scroll to the device manufacturer (e.g., AMD).	BLANK CHECK AMD

Procedure	Example 201 Displays
4. Press ENTER when the correct manufacturer is displayed.	BLANK CHECK AMD 2716
5. Scroll to the part number of the device (e.g., 2764).	BLANK CHECK AMD 2764
6. Press ENTER to select the part number.	INSERT DEVICE AMD 2764
7. Insert the device into the socket and press ENTER to initiate the check.	BLANK CHECK . . . . .
If the blank check is successful for the installed device, the socket lamp lights green and the display reads	BLANK CHECK OK
If the blank check operation is unsuccessful for the installed device, the socket lamp lights red and the display reads	BLANK CHECK NONBLANK 20
8. Lift the socket lever and remove the device from the socket.	
9. To blank check another device that uses the same family/pinout code (see the Device List for correct family and pinout codes), press ENTER and return to step 7. To exit the Blank Check operation, press a scroll key.	

Programming

The 201 EPROM Programmer can be used to program devices with the data contained in a master device or with data downloaded to the programmer via the serial I/O (RS232) port. The following paragraphs describe the programming operation using a master device. If you choose to download the program data to the programmer's memory instead of loading from a master device, refer also to Downloading Data following this subsection.

Programming a device consists of two basic operations:

- ☐ Loading the data from a master device to the programmer's memory (or downloading the data from a remote source such as a host computer — refer to Downloading Data).
- ☐ Programming a blank device with the data copied to the programmer's memory.

Loading the Data from a Master Device

The first step in programming a device is to load the data from the master device. This operation transfers the program data from the master device to the programmer's memory. When the transfer is complete, the programmer calculates and displays the sumcheck of the data. Use the following procedure to load the data from the master device.

Procedure	Example 201 Displays
1. Scroll to the LOAD FROM MASTER function.	LOAD FROM MASTER
2. Press ENTER.	LOAD FROM MASTER F/P CODE 79/33
3. If the correct family/pinout code is displayed, press ENTER and go to step 7. If the correct code is not displayed, key in the 4-digit family/pinout code for the device (e.g., AF33), and skip to step 7	LOAD FROM MASTER F/P CODE AF/33
or	
press a scroll key and then scroll to the device manufacturer (e.g., AMD).	LOAD FROM MASTER AMD
4. Press ENTER when the correct manufacturer is displayed.	LOAD FROM MASTER AMD 2716

Procedure	Example 201 Displays
5. Scroll to the part number of the device (e.g., 2764).	LOAD FROM MASTER AMD 2764
6. Press ENTER to select the part number.	INSERT DEVICE AMD 2764
7. Insert the master device into the socket and press ENTER to initiate the load operation.	LOAD FROM MASTER . . . . .
If the load operation is successful, the socket lamp lights green. Note the sumcheck displayed on the programmer for later verification.	LOAD FROM MASTER SUMCHECK = BA25
If the load operation is unsuccessful, the socket lamp lights red and the programmer displays an error message. (See the Error Messages section for an explanation of the error message displayed.)	
8. Lift the socket lever and remove the master device from the socket.	

Programming a Device

After the master data has been loaded into programmer RAM from a master device or downloaded to the programmer, use the following procedure to program devices with the data. The program operation copies the data from the programmer's memory to the installed device. When the programming operation is complete, the programmer calculates and displays the sumcheck of the data.

Procedure	Example 201 Displays
1. Scroll to the PROGRAM function.	PROGRAM
2. Press ENTER. If the data to be programmed was just copied from the master device and the master is the same type of device as the device to be programmed, press ENTER and skip to step 7. Otherwise, proceed to select the device type for the part to be programmed.	PROGRAM F/P CODE AF/33
3. Key in the 4-digit family/pinout code for the device (e.g., AF33) and skip to step 7 or press a scroll key and then scroll to the device manufacturer (e.g., AMD).	PROGRAM F/P CODE AF/33 PROGRAM AMD

Procedure	Example 201 Displays
4. Press ENTER when the correct manufacturer is displayed.	PROGRAM AMD 2716
5. Scroll to the part number of the device (e.g., 2764).	PROGRAM AMD 2764
6. Press ENTER to select the part number.	INSERT DEVICE AMD 2764
7. Insert the blank device into the socket and press ENTER to initiate the program operation. If the program operation is successful, the socket lamp lights green. Note that the sumcheck displayed on the programmer matches that displayed at the end of the load operation. If the program operation is unsuccessful, the socket lamp lights red and the programmer displays an error message. (See the Error Messages section for an explanation of the error message displayed.)	PROGRAM . . . . . . PROGRAM SUMCHECK = BA25
8. Lift the socket lever and remove the device from the socket. To program another device of the same type (using the same family and pinout codes) with the same master data, press ENTER and return to step 7. To exit the program operation, press a scroll key.	

Downloading Data

The 201 can receive data transferred from a host computer to its memory over a serial communications link. (Refer to the Getting Started section for information on serial I/O connections.) This feature may be used to download data from a computer, or other software development system, to the programmer in preparation for programming parts. The programmer can accept the data in any of five data translation formats: Intel Intellec 8/MDS, Intel MCS-86 Hexadecimal Object, Motorola Exorciser, Motorola Exormax, and Tektronix Hexadecimal. Descriptions of each of these formats are provided in the Computer Remote Control section of this manual.

When initiating a download operation, the programmer prompts for the entry of three parameters in addition to the data translation format. These parameters are:

- ☐ the beginning address
- ☐ the block size
- ☐ the offset address

Selection of the beginning address allows you to make transfers of data to a specified start address in the programmer's memory. For example, if you specify a beginning address of 1000H (hexadecimal), the data transferred from the external data source will be placed in the programmer's memory starting at location 1000H instead of 0000H.

Selection of the block size allows you to specify how much data is to be transferred to the programmer's memory (starting at the specified beginning address). For example, if you specify a beginning address of 1000H and a block size of 2000H, the subsequent data transfer operation would transfer 8192 (2000H) bytes of data to the programmer's memory, with the first byte written to location 1000H. You enter 0000H to disable the selected block size.

The offset address allows you to select the beginning address, in the host computer memory, of the file to be downloaded. For example, if you specify an offset address of 2000H, then the first byte of data downloaded will be from address 2000H in the computer memory. The default offset address is the first address received from the host. You reset the default offset address by entering all F's.

After you have hooked up the programmer to the host computer and set the proper communications protocol (see Setting Communications Protocol later in this section), perform a download operation as follows:

Procedure	Example 201 Displays
1. Scroll to the RS232 PORT function.	RS232 PORT
2. Press ENTER.	RS232 PORT COMPUTER CONTROL
3. Scroll to the DOWNLOAD function.	RS232 PORT DOWNLOAD
4. Press ENTER.	DOWNLOAD FORMAT INTEL INTELLEC 8
5. Scroll to the desired data translation format (e.g., Motorola Exormax) and press ENTER.	DOWNLOAD EXORMAX BEG RAM ADD=0000

Procedure	Example 201 Displays
6. Press ENTER to select the displayed beginning RAM address (the first programmer memory address that the downloaded data will be transferred to) or key in the hexadecimal beginning address and press ENTER.	DOWNLOAD EXORMAX BLCK SIZE=000000
7. Press ENTER to select the displayed block size, or key in the block size of the data to be downloaded and press ENTER.	DOWNLOAD EXORMAX OFFSET =FFFFFFFF

8. Press ENTER to select the displayed offset address or key in the offset address required to specify the file address in the host computer and press ENTER. (The default offset address, FFFFFFFF, is the first address received by the programmer. The first address received is subtracted from all subsequent addresses.) The terminal will display

   ```
   LOADING PORT
   ```

   You can now initiate the download operation from the host computer.

Verifying Programmed Parts

The verify operation allows you to check programmed devices to make sure that the data in a device matches the data in the programmer's memory. The verify function compares device data with the data read into the programmer's memory during the preceding load or download operation.

The verify operation uses the VccH and VccL and number of passes recommended by the manufacturer of the device being verified. VccH and VccL levels can be selected specifically when using Terminal Remote Control or Computer Remote Control.

Before you can verify the data within a device, you must first make sure that the master data is contained in the programmer's memory. You can load the master data into the programmer's memory by performing either the Load from Master operation or the Download operation described previously in this section.

With the master data contained in the programmer's memory, perform the verify operation as follows:

Procedure	Example 201 Displays
1. Scroll to the VERIFY function.	`VERIFY`
2. Press ENTER. If the data to be verified was just copied from the master device and the master device is the same type of device as the device to be verified, press ENTER and skip to step 7. Otherwise, proceed to select the device type for the part to be verified.	`VERIFY` `F/P CODE AF/33`

Procedure	Example 201 Displays
3. Key in the 4-digit family/pinout code for the device (e.g., AF33) and skip to step 7 or press a scroll key and then scroll to the device manufacturer (e.g., AMD).	VERIFY F/P CODE AF/33 VERIFY AMD
4. Press ENTER when the correct manufacturer is displayed.	VERIFY AMD 2716
5. Scroll to the part number of the device (e.g., 2764).	VERIFY AMD 2764
6. Press ENTER to select the part number.	INSERT DEVICE AMD 2764

Procedure	Example 201 Displays
7. Insert the device to be verified into the socket and press ENTER to initiate the verify operation.	VERIFY . . . . .
If the verify operation is successful, the socket lamp lights green and the sumcheck of the data is displayed.	VERIFY SUMCHECK = BA25
If the verify operation is unsuccessful, an error message is displayed and the socket lamp lights red. (See the Error Messages section for an explanation of the error message number displayed.)	VERIFY VERIFY ERROR NN
8. Lift the socket lever and remove the device from the socket.	
9. To verify another device of the same type (using the same family and pinout codes) against the same master data, press ENTER and return to step 7. To exit the Verify operation, press a scroll key.	

Setting Communications Protocol

The programmer can be made to select the correct communications protocol automatically (see Terminal Remote Control or Computer Remote Control) or you can change the communications protocol from the default values by using the following procedure. The default values are:

baud rate = 9600 parity check = none number of data bits = 8 number of stop bits = 1

Procedure	Example 201 Displays
1. Scroll to the RS232 PORT function.	RS232 PORT
2. Press ENTER.	RS232 PORT COMPUTER CONTROL
3. Scroll to the PORT SETTING function.	RS232 PORT PORT SETTING
4. Press ENTER.	PORT SETTING BAUD RATE = 9600
5. Scroll to the desired data transfer speed and press ENTER.	PORT SETTING PARITY CHK.=NONE

Procedure	Example 201 Displays
6. Scroll to the desired parity and press ENTER.	PORT SETTING DATA BITS = 8
7. Scroll to the desired number of data bits and press ENTER.	PORT SETTING STOP BIT = 1
8. Scroll to the desired number of stop bits and press ENTER.	RS232 PORT

This completes setting of the communications protocol for the serial I/O port.

Uploading Data

The 201 can be used to transfer device data from its memory to a computer or other development system over a serial communications link. (Refer to RS232 Port Cable Connections for information on serial I/O connections.) This feature may be used to upload data from the programmer's memory for the purpose of editing or storing the data and then downloading it back to the programmer's memory. The programmer can transfer the data in any of five data translation formats: Intel Intellec 8/MDS, Intel MCS-86 Hexadecimal Object, Motorola Exorciser, Motorola Exormax, and Tektronix Hexadecimal. Descriptions of each of these formats are provided in the Computer Remote Control section of this manual.

When initiating an upload operation, the programmer prompts for the entry of three parameters in addition to the data translation format. These parameters are:

- ☐ the beginning address
- ☐ the block size
- ☐ the offset address

Selection of the beginning address allows you to make transfers that use only a portion of data contained in the programmer's memory. For example, if you specify a beginning address of 8000H (hexadecimal), the data transfer operation will begin at address 8000H (32,768 decimal).

Selection of the block size allows you to specify how much of the data contained in the programmer's memory is to be transferred, starting at the specified beginning address. For example, if you specify a beginning address of 8000H and a block size of 1000H, the subsequent data transfer operation would transfer only the data contained in the address range 8000H to 8FFFH. If you do not specify a block size, the default block size is the size of the current device.

Selection of the offset address allows you to select the address in the computer memory where the data will be uploaded. For example, if you specify a beginning address of 8000H and an offset address of 2000H, data from programmer address 8000H will be uploaded to host computer address 2000H.

After hooking up the programmer to the host computer and setting the correct communications protocol, perform an upload operation as follows:

Procedure	Example 201 Displays
1. Scroll to the RS232 PORT function.	RS232 PORT
2. Press ENTER.	RS232 PORT COMPUTER CONTROL
3. Scroll to the UPLOAD function.	RS232 PORT UPLOAD
4. Press ENTER.	UPLOAD FORMAT INTEL INTELLEC 8
5. Scroll to the desired data translation format (e.g., Tektronix Hexadecimal) and press ENTER.	UPLOAD TEKTRONIX BEG RAM ADD=0000

Procedure	Example 201 Displays
6. Press ENTER to select the displayed beginning RAM address, or key in the hexadecimal beginning address of the data in the programmer's memory to be uploaded and press ENTER.	UPLOAD TEKTRONIX BLCK SIZE=000000
7. Press ENTER to select the displayed block size, or key in the block size of the data to be uploaded and press ENTER. (Entering a block size of 0 selects the current device size.)	UPLOAD TEKTRONIX OFFSET =FFFFFFFF
8. Press ENTER to select the displayed offset address, or key in the offset address for the data and press ENTER. (The default offset address, FFFFFFFF, causes address 000 to be sent by the programmer.) The terminal will display SENDING PORT	
When the upload operation is complete, the programmer displays the sumcheck of the data that was transferred to the host computer..	SENDING COMPLETE SUMCHECK = HHHH
"HHHH" represents the hexadecimal summation.	

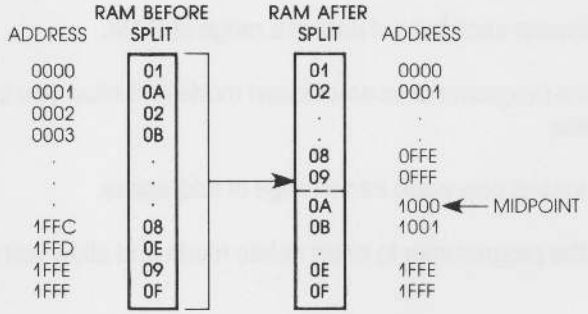
Editing

After the master program data has been loaded into the programmer's RAM, it can be altered, rearranged, added to, or deleted from using the editing functions. These functions are summarized below in the order they occur in the 201 menus and are discussed in more detail in the pages that follow.

- ☐ SPLIT — splits a block of 16-bit-wide data into two adjacent 8-bit-wide blocks that take up the same amount of programmer RAM. Split always starts at address 0.
- ☐ SHUFFLE — combines two blocks of 8-bit-wide data into one block of 16-bit-wide words for subsequent uploading or editing in 16-bit-wide format.
- ☐ COMPLEMENT — inverts each byte of data in a range of RAM.
- ☐ INSERT — causes the programmer to enter insert mode and allow you to insert data bytes into consecutive addresses.
- ☐ BLOCK INSERT — inserts one value into a range of addresses.
- ☐ DELETE — causes the programmer to enter delete mode and allow you to delete data bytes one-by-one.
- ☐ BLOCK DELETE — deletes a block of data.
- ☐ BLOCK FILL — fills a range of addresses with a single value. The data originally contained in the filled addresses is overwritten.
- ☐ DATA SEARCH — searches for up to eight bytes of data in a specified range of RAM.
- ☐ BLOCK MOVE — copies a block of data from one location in RAM to another, overwriting the data originally stored in the addresses copied to (destination addresses).
- ☐ DATA EDIT — enters an editing mode which allows you to change data bytes one-by-one.

Splitting 16-bit-wide Data

Use the Split function to split a block of 16-bit-wide data into two adjacent 8-bit-wide blocks occupying the same block of programmer RAM. The odd-addressed bytes and even-addressed bytes are divided into two blocks around a specified midpoint, with the odd-addressed bytes starting at the midpoint and the even-addressed bytes starting at address 0 (see illustration). This command should be used when the data to be programmed into blank parts is downloaded from the development system in 16-bit-wide format and must be programmed into two 8-bit-wide parts that will be accessed as one 16-bit-wide word. You can reverse the Split function by using the Shuffle function.

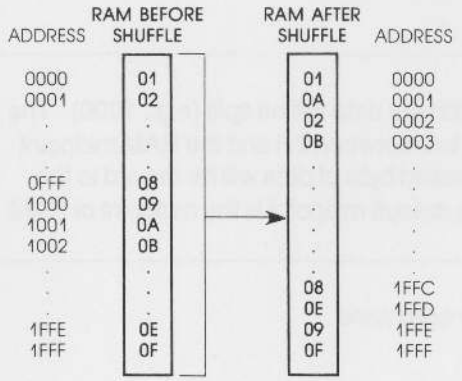


To split a block of data, proceed as follows:

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the SPLIT function (if necessary) and press ENTER.	SPLIT MIDDLE ADR.=8000
4. Key in a midpoint around which the data will be split (e.g., 1000). The midpoint must be a power of two between 2H and the RAM midpoint (8000H). The first odd-addressed byte of data will be moved to the address at the midpoint. The default midpoint is the midpoint of RAM.	SPLIT MIDDLE ADR.=1000
5. Press ENTER to execute the command.	SPLIT . . . . .
After the split has been completed, the display will read	SPLIT COMPLETED

Returning 8-bit-wide Data to 16-bit-wide Format

Use the Shuffle function to converge two adjacent blocks of 8-bit-wide data into one block of 16-bit-wide data occupying the same block of RAM. This operation reverses the split operation, shuffling the two blocks of data together exactly the same way you would shuffle two halves of a deck of cards together (alternating data bytes of one block with data bytes of the other block). The data bytes of the block below a specified midpoint are placed in even-numbered addresses, starting with address 0, and the data bytes of the block at and above the specified midpoint are placed in odd-numbered addresses, starting with address 1. This command should be used to reunite 16-bit-wide data that was loaded into memory from two 8-bit-wide devices for subsequent uploading and editing on a development system or host computer in 16-bit-wide format. (In Terminal Remote Control you can load data in 16-bit-wide format directly.)



Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the SHUFFLE function and press ENTER.	SHUFFLE MIDDLE ADR.=8000
4. Key in the midpoint of the data block to be shuffled (e.g., 1000). (The first byte *after* the midpoint will be the byte placed into address 1.) The midpoint must be a power of two between 2H and the RAM midpoint (8000H). The default midpoint is the midpoint of RAM .	SHUFFLE MIDDLE ADR.=1000
5. Press ENTER to execute the command.	SHUFFLE . . . . .
After the shuffle has been completed, the display will read	SHUFFLE COMPLETED

Complementing Data

Use the Complement function to invert the data contained in a specified range of memory (a 0 will become an F and an F will become a 0). Data complements are based on 8-bit-word size.

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the COMPLEMENT function and press ENTER.	COMPLEMENT START ADR.= 0000
4. Key in the first address of the range of data you want to complement and press ENTER. The default start address is 0.	COMPLEMENT END ADR.= FFFF
5. Key in the last address of the range of data you want to complement and press ENTER. The default end address is the end of RAM (FFFF).	COMPLEMENT . . . . .
When the complement is complete, the display will read	COMPLEMENT COMPLETED

Inserting Single Data Bytes

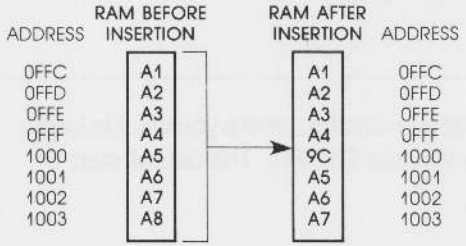
Use the Insert function to insert data bytes one at a time into consecutive addresses. (To insert the same value into a block of memory, use the Block Insert function.) Executing the Insert function causes the programmer to go into an insert mode. You may continue inserting data bytes into consecutive memory addresses until you press a scroll key. Pressing either of the scroll keys will cause the programmer to exit insert mode and return to the main menu. The original data at the address where new data is inserted is pushed up in memory to accommodate the new data. Data at the end of memory is lost.

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the INSERT function and press ENTER.	INSERT ADDRESS = 0000
4. Key in the address of the memory location where you want to begin inserting data (e.g., 1000) and press ENTER. The default start address of insertion is 0.	ADDRESS = 1000 INSERT DATA = __

Procedure	Example 201 Displays
5. Key in the data value that you want to insert at the address displayed on the top line of the display.	ADDRESS = 1000 INSERT DATA = 9C
6. Press ENTER to insert the data into memory. When you press ENTER, the programmer will display the next consecutive address, at which the next keyed in byte of data will be inserted. To continue inserting data into consecutive addresses, return to step 5. To exit insert mode and return to the main menu, press either of the scroll keys.	ADDRESS = 1001 INSERT DATA = __

Example

Insert data 9C at address 1000:



Inserting a Single Value into a Block of RAM

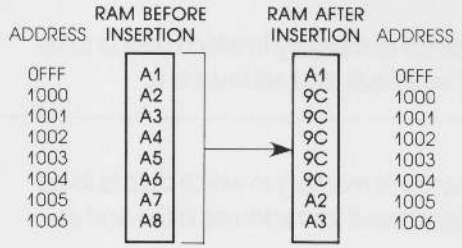
Use the Block Insert function to insert one value into a range of memory addresses. The original data in the range where new values are inserted is pushed up in memory to accommodate the new data. Data at the end of memory is lost. (To overwrite a range of memory addresses with a value, use the Block Fill function.)

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the BLOCK INSERT function and press ENTER.	BLOCK INSERT START ADR.= 0000
4. Key in the first address of the range of memory in which data is to be inserted and press ENTER. The default start address is 0.	BLOCK INSERT END ADR.= FFFF
5. Key in the last address of the range of memory in which data is to be inserted and press ENTER. The default end address is the end of RAM (FFFF).	BLOCK INSERT DATA = __

Procedure	Example 201 Displays
6. Key in a one byte value to be inserted into the specified range of memory.	BLOCK INSERT DATA = 9C
7. Press ENTER to execute the Block Insert function.	BLOCK INSERT . . . . .
When the insert function is complete, the display will read	BLOCK INSERT COMPLETED

Example

Insert data 9C into block from 1000 through 1004:



Deleting Single Data Bytes

Use the Delete function to delete data bytes one at a time. Data following the deleted data byte is shifted down to fill in the deleted address and FF is filled in the last byte of RAM. Executing the Delete function causes the programmer to go into a delete mode. You may continue deleting successive data bytes until you press a scroll key. Pressing a scroll key causes the programmer to exit delete mode and return to the main menu.

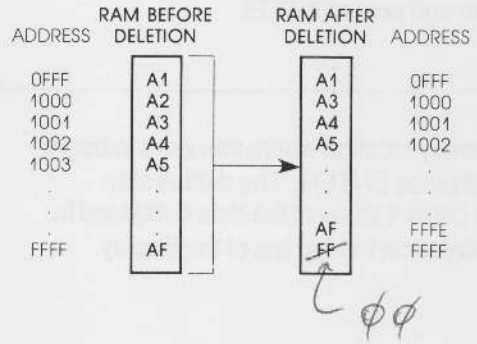
Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the DELETE function and press ENTER.	DELETE ADDRESS = 0000
4. Key in the address of the memory location where you want to begin deleting data (e.g., 1000) and press ENTER. The default start address of deletion is 0. The current value of the data contained in the specified address is displayed on the top line of the display.	DELETE DATA = 93 ADDRESS = 1000

Procedure	Example 201 Displays
5. Press ENTER to delete the current contents of the displayed address. The next data byte is shifted down to the current address displayed. (The same address is displayed but it contains the next data byte.)	DELETE DATA = 94 ADDRESS = 1000

6. To continue deleting successive data bytes, continue repeating step 5. To exit delete mode and return to the main menu, press either of the scroll keys.

Example

Delete data at address 1000:



Deleting a Block of Data

Use the Block Delete function to delete a block of data stored in the programmer's RAM. Data following the deleted block is shifted down to fill in the deleted range and FF is filled in the vacated RAM addresses at the end of memory.

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the BLOCK DELETE function and press ENTER.	BLOCK DELETE START ADR.= 0000
4. Key in the address of the first data byte to be deleted (e.g., 1000) and press ENTER. The default start address is 0.	BLOCK DELETE END ADR.= FFFF
5. Key in the address of the last data byte to be deleted (e.g., 1FFF). The default end address is the end of RAM (FFFF).	BLOCK DELETE END ADR.= 1FFF
6. Press ENTER to execute the Block Delete function.	BLOCK DELETE . . . . .
When the delete function is complete, the display will read	BLOCK DELETE COMPLETED

Filling a Block of Memory

Use the Block Fill function to overwrite the data contained in a range of RAM with a constant value or data byte.

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the BLOCK FILL function and press ENTER.	BLOCK FILL START ADR.= 0000
4. Key in the first address of the range of memory to be overwritten by the new data value (e.g., 1000) and press ENTER. The default start address is 0.	BLOCK FILL END ADR.= FFFF
5. Key in the last address of the range of memory to be overwritten by the new data value (e.g., 1FFF) and press ENTER. The default end address is the end of RAM (FFFF).	BLOCK FILL DATA = __

Procedure	Example 201 Displays
6. Key in a one byte value to be written into the specified range of memory (e.g., 9C).	BLOCK FILL DATA = 9C
7. Press ENTER to execute the Block Fill function.	BLOCK FILL . . . . .
When the Block Fill function is complete, the display will read	BLOCK FILL COMPLETED

Searching for Data

Use the Data Search function to search for up to eight bytes of data in any specified range of memory. You must specify a range of memory to search that is larger than the block of data bytes to be searched for or the data search will not be executed.

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the DATA SEARCH function and press ENTER.	DATA SEARCH START ADR.= 0000
4. Key in the address of the memory location where you want the data search to begin (e.g., 1000) and press ENTER. The default start address is 0.	DATA SEARCH END ADR.= FFFF
5. Key in the address of the memory location where you want the data search to end (e.g., 1FFF) and press ENTER. The default end address is the end of RAM (FFFF).	DATA = __

Procedure	Example 201 Displays
6. Key in the value of a byte to be searched for in the specified range of memory and press ENTER to enter that byte in the search string. (The comma will be provided by the programmer when you press ENTER.)	`DATA = 01,`
7. To enter the next byte of data to be searched for, return to step 6, or press ENTER to execute the Data Search function for the data already entered. You may enter up to eight bytes of data to be searched for. After pressing ENTER to accept the eighth byte, the search will begin immediately and you will not have to press ENTER a second time. A match will occur only if the data is found in the same order as the typed in string of data.	`DATA = 01,02,03,` `04,05,06,07,08`
8. The programmer will search the specified range for the data string entered on the display. If the data is found, the programmer will display the address of the first byte of the string found. You can then press ENTER to continue searching for other occurrences of the data in the specified range, or you can return to the main menu by pressing either scroll key.	`DATA MATCH` `ADDRESS = 10FF`
If the data was not found in the specified range (or the data string was longer than the specified range) the programmer will display a fail message. Press any key to return to the main menu.	`DATA SEARCH` `FAIL`

Copying Data from One RAM Location to Another

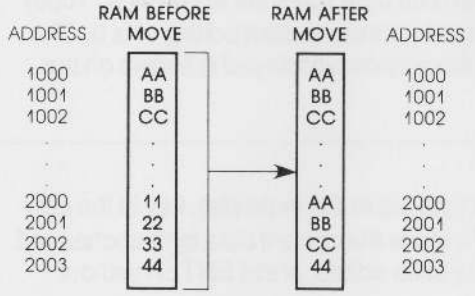
Use the Block Move function to copy data from one RAM location to another. The original data contained in the addresses to which the data was copied is overwritten by the copied data. Also make sure that the destination address of the block to be copied plus the block size of the block to be copied does not exceed programmer RAM (FFFF).

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the BLOCK MOVE function and press ENTER.	BLOCK MOVE START ADR.= 0000
4. Key in the address of the first data byte you want to copy and press ENTER. The default start address is 0.	BLOCK MOVE END ADR.= FFFF
5. Key in the address of the last byte of data you want to copy and press ENTER. (If you want to copy only one byte of data, type in the same value for the end address as you did for the start address.) The default end address is the end of RAM (FFFF).	BLOCK MOVE DESTN ADR.= 0000

Procedure	Example 201 Displays
6. Key in the address to which the first byte of data will be copied (destination address) (e.g., 2000). The rest of the data to be copied will be copied into consecutive addresses following the destination address.	BLOCK MOVE DESTN ADR.= 2000
7. Press ENTER to execute the Block Move function.	BLOCK MOVE . . . . .
When the copy is complete, the display will read	BLOCK MOVE COMPLETED

Example

Copy block 1000 through 1002 to address 2000:



Editing Single Data Bytes

Use the Data Edit function to edit individual bytes of data stored in the programmer's RAM. Executing this command causes the programmer to go into a special edit mode that will allow you to display and edit data bytes one-by-one until you press a scroll key.

Procedure	Example 201 Displays
1. Scroll to the EDIT function.	EDIT
2. Press ENTER.	EDIT SPLIT
3. Scroll to the DATA EDIT function and press ENTER.	DATA EDIT ADDRESS = 0000
4. Key in the address of the first data byte you want to edit (e.g., 1000) and press ENTER. The default address to start editing at is 0. The current value of the data at the address displayed is shown on the second line of the display.	ADDRESS = 1000 DATA 05--> __
5. To change the current data byte (as in the example), key in the new value and press ENTER. To leave the current data byte unchanged and advance to the next byte to be edited, press ENTER without keying in a new value.	ADDRESS = 1000 DATA 05--> 9C

Procedure	Example 201 Displays
6. After pressing ENTER, the next address will be displayed on the top line of the display and the current value of the data at that address will be displayed on the second line of the display. To edit the displayed data byte, repeat step 5. To exit the edit mode and return to the main menu, press either scroll key. If you key in a new value and then press a scroll key before pressing ENTER, the keyed in data will be ignored and the current data byte will remain unchanged.	ADDRESS = 1001 DATA 06--> __

3. Terminal Remote Control

Introduction

This section explains how to operate the 201 EPROM Programmer from a remote terminal. All of the operations that can be performed from the front panel are available in Terminal Remote Control (TRC) mode, as well as set programming commands.

Following this introduction is a summary of all of the commands used to operate the programmer from a remote terminal and some general operating notes follow the command summary. Read the general operating notes before attempting any of the Terminal Remote Control Commands.

The commands for operation from a remote terminal are grouped into the following subsections:

- ☐ ENTERING AND EXITING TERMINAL REMOTE CONTROL MODE — This subsection explains how to transfer control of the programmer to a remote terminal and how to return control of the programmer to the front panel.
- ☐ ON-LINE HELP — This subsection explains how to display the on-line help screen and how to read the help screen symbols.
- ☐ SELECTING A DEVICE TYPE — This subsection explains how to select the device type. Selecting the device type tells the programmer which algorithm and voltages it should use to blank check, load, program, and verify a part. The correct device type must be selected prior to performing programming operations, otherwise damage to the installed device could result.
- ☐ PROGRAMMING DEVICES — This subsection explains how to perform a blank check on devices prior to programming them, load the master data into RAM from previously programmed devices, and program blank devices with the master data.

- ☐ VERIFYING PROGRAMMED DEVICES — This subsection explains how to verify devices after they have been programmed. Devices are automatically verified during the programming operation, but this command allows you to verify the devices at another time.

- ☐ EDITING MEMORY — This subsection explains how to edit the contents of the programmer's data memory, or RAM.

- ☐ SUMCHECKING DATA — This subsection explains how to perform a sumcheck (hexadecimal summation) and a checksum (exclusive-OR summation) on the data stored in the programmer's RAM.

Symbols and Conventions

The following is a list of symbols and conventions used in the Terminal Remote Control section of this document:

<CR>	This symbol represents the carriage return key.
<ESC>	This symbol represents the escape key.
<BS>	This symbol represents the backspace key.
CTRL-S	CTRL- followed by a character means that you should hold down the CONTROL (or CTRL) key and press the character once. For example, to type CTRL-S, you would hold down the CONTROL key and press the S key once.
H	Any value followed by an upper-case H means that the value is in hexadecimal notation.
<lower-case>	Lower-case letters enclosed in angle brackets describe the kind of parameter that you should fill in for the command. Replace the words and the brackets with the appropriate parameter value for the operation you are performing. For example, the Memory Modify command format is M <start address>. You would enter M 6 to modify memory starting at address 6.
[<lower-case>]	Words or characters enclosed in square braces represent optional entries.

Terminal Remote Control Command Summary

Command Format	Description
M	Enter TRC mode
R	Exit TRC mode (Return)
<CR>	Execute command
<ESC>	Abort command
<BS>	Erase previous character
CTRL-S	Suspend display
CTRL-Q	Continue display
?	Display help screen
A?	Select device type
B	Blank check device
L [a1 a2 a3]	Load single master
LS [#]	Load Set of masters
LW [#]	Load Word-wide masters
LL [#]	Load Long-word-wide masters

Command Format	Description
P [a1 a2 a3]	Program single device
PS [#]	Program Set of devices
PW [#]	Program Word-wide set
PL [#]	Program Long-word-wide set
V [H/L] [a1 a2 a3]	Verify single device
VS [H/L] [#]	Verify Set of devices
VW [H/L] [#]	Verify Word-wide set
VL [H/L] [#]	Verify Long-word-wide set
M a1 a2	Display Memory
M a1	Modify Memory
M a1 a2 d1 [d2..d8]	Fill Memory
I a1 d1 [d2..d8]	Insert data
D a1 [a2]	Delete data
T a1 a2 a3	Transfer (copy) data
MS a1 a2 d1 [d2..d8]	Search Memory
CT a1 a2 [a3]	Sumcheck (Total)
CX a1 a2 [a3]	Checksum (EXOR)

List of Symbols

a1 = start address
a2 = end address
a3 = destination address

d1..d8 = data bytes 1 through 8
# = number of devices in set
H/L = Vcc selection (H or L)

General Operating Notes

This subsection contains important information on how to type in the Terminal Remote Control (TRC) commands, as well as information on how to use some special keys to control command execution, what the default parameter values are, and how to read the device error indicators.

Entering Commands

Below are some general rules for entering Terminal Remote Control commands.

- ☐ Each command must be entered at the command prompt (>) and must be followed by a carriage return.
- ☐ All commands must be typed in upper-case letters.
- ☐ In most cases, the first parameter (such as a memory address) that follows the command letter(s) can either be separated from the command letter(s) by a space or it can be typed immediately following the command letter(s) with no space. For example, the Load Set command could be typed LS4 or LS 4 to load a set of four masters. The one exception is the command to verify a single device, which must have a space after the command letter V before the first parameter is specified.
- ☐ Each parameter following the first parameter must be separated from the preceding parameter by a space. For example, to display memory addresses 0 through F, the Display Memory command must be typed M 0 F or M0 F.
- ☐ You do not need to enter leading zeros when entering command parameters, such as memory addresses. For example, 0010 is the same as 10.
- ☐ All command parameters are entered as hexadecimal numbers, unless otherwise specified.

Controlling Command Execution

The following keys are special keys which can be used to control the execution of TRC commands.

BACKSPACE Key (CTRL-H)
While you are entering TRC commands, you can use the backspace key (or the CTRL-H, if your terminal keyboard has no backspace key) to delete the previously entered character. You can use this key repeatedly until you have deleted the entire command line.

ESCAPE Key
You can abort the command currently being executed or the command you are currently typing by pressing the escape key. If you press the escape key while typing in a command (before pressing carriage return), the programmer will ignore the line you were typing, display the command prompt (>) on a new line, and await your input. If you press the escape key while a command is being executed, the programmer will abort the command, display the > prompt on a new line, and await your input.

CTRL-S
Typing a CTRL-S will suspend the displaying of data on the terminal screen. The command that was being executed when the CTRL-S was typed is temporarily halted, but not aborted. To continue the display, type CTRL-Q.

CTRL-Q
Typing a CTRL-Q continues the displaying of data on the terminal screen exactly where it was suspended by the CTRL-S command.

Action Display

While an operation is being executed in TRC mode, advancing dots will appear on the bottom line of the programmer display, indicating that the programmer is performing the operation.

Error Indicators

The lamp (LED) below the socket is used to indicate successful or unsuccessful programming operations. If an operation is completed successfully on a device, a "pass" signal will sound (two beeps) and the LED below the socket will light green. If an operation is not completed successfully on a device, a "fail" signal will sound (three beeps) and the LED below the socket will light red.

A "SYNTAX ERROR" will occur when the format of a command is incorrect or when illegal characters are entered in a command. A caret (^) will appear below the first unrecognized character encountered by the programmer.

Default Parameter Values

Many of the Terminal Remote Control commands can be accompanied by optional parameters, such as starting and ending memory addresses. The default values of these optional parameters when the 201 is powered up are listed below. These default values remain constant except for the default number of devices for a set, word-wide set, or long-word-wide set, which becomes the last value selected after the parameter is assigned a value.

<start address>	0000H
<end address>	1FFFH
<destination address>	0000H
<number of devices>	
for a set	1
for a word-wide set	2
for a long-word-wide set	4

Entering and Exiting Terminal Remote Control Mode

The following is an explanation of how to transfer control of the programmer to the remote terminal (enter TRC mode) and how to return control of the programmer to the front panel (exit TRC mode).

Entering Terminal Remote Control Mode

Procedure	Example 201 Displays
1. Connect the programmer to the terminal as described in the RS232 Port Cable Connections discussion in the Getting Started section of this manual.	
2. Power up the programmer by plugging the power cord into the back of the programmer and into a power outlet and pressing the power switch on the back of the programmer to the ON position.	SELF TESTING . . . . .
3. When the self test is complete and the display reads LOAD FROM MASTER, scroll through the main menu to RS232 PORT and press ENTER.	RS232 PORT COMPUTER CONTROL
4. Scroll through the RS232 PORT menu to TERMINAL CONTROL and press ENTER.	TERMINAL CONTROL

Procedure	Example 201 Displays
5. To establish communications between the programmer and the terminal and automatically select the proper communications protocol, type M<*CR*> on the terminal keyboard. (The "M" will not be displayed on the terminal screen.) You can also set the communications protocol manually, by executing the PORT SETTING function (see the Front Panel Operation section). If the correct port settings are selected when you enter TRC mode, you can press any terminal key to establish communications. When communication is established, a command prompt (>) will appear on the terminal screen. The programmer will display TERMINAL CONTROL until control of the programmer is returned to the front panel.	TERMINAL CONTROL

Exiting Terminal Remote Control Mode

You can exit Terminal Remote Control mode by using either the terminal keys or the programmer keys. To exit TRC using the terminal keys, type

R<*CR*>

(The R command stands for Return to local.) To exit TRC using the programmer keys, press any key.

The programmer will display

```
RS232 PORT
```

You can now continue operating the programmer using the front panel keys.

On-Line Help

A help screen displaying the currently selected device type and a list of all of the TRC commands and their formats can be displayed on the terminal screen by typing

?<CR>

at the command prompt (>). The help screen is shown below:

```
          ***  201 PROGRAMMER COMMAND HELP  ***

SELECT DEVICE TYPE A?                \ SUMCHECK (TOTAL) CT A1 A2[ A3]
BLANK CHECK        B                 \ CHECKSUM (EXOR)  CX A1 A2[ A3]
LOAD SINGLE        L  [A1 A2 A3]     \ MEMORY DISPLAY   M  A1 A2
LOAD SET           LS [ # ]          \ MEMORY MODIFY    M  A1
LOAD WORD          LW [ # ]          \ MEMORY FILL      M  A1 A2 D1[..D8]
LOAD LONG WORD     LL [ # ]          \ INSERT           I  A1 D1[..D8]
PROGRAM SINGLE     P  [A1 A2 A3]     \ DELETE           D  A1[ A2]
PROGRAM SET        PS [ # ]          \ TRANSFER         T  A1 A2 A3
PROGRAM WORD       PW [ # ]          \ MEMORY SEARCH    MS A1 A2 D1[..D8]
PROGRAM LONG WORD  PL [ # ]          \ RETURN TO LOCAL  R
VERIFY SINGLE      V  [H/L][A1 A2 A3]\ HELP             ?
VERIFY SET         VS [H/L][ # ]     \
VERIFY WORD        VW [H/L][ # ]     \
VERIFY LONG WORD   VL [H/L][ # ]     \

DEVICE TYPE: INTEL  2764        F/P: 79/33    MASTER #: 01
```

List of Symbols

A1 = start address
A2 = end address
A3 = destination address

D1..D8 = data bytes 1 through 8
# = number of devices in set
H/L = Vcc selection (H or L)

Selecting a Device Type

Use the Select Device Type command to select the manufacturer and part number of the device you are loading from, programming, blank checking or verifying. After you have selected the device's manufacturer and part number, the 201 EPROM Programmer will automatically supply the proper family and pinout codes for the device. The family and pinout codes tell the programmer which algorithm and voltages to use to program the part. *You must select the correct device type before blank checking a device, loading from a device, programming a device, or verifying a device, otherwise damage to the device could result.* The power-up default device type is Intel 2764.

Command Format: A?

Procedure	Example Key Sequence
1. Type in the Select Device Type command letters, A?, and press carriage return.	A?<CR>
2. After the list of device manufacturers is displayed on the screen, type in the number that corresponds to the manufacturer of the device you are blank checking, loading, programming or verifying and press carriage return. If you press carriage return without entering a number, device manufacturer 0 will be selected.	6<CR>
3. After the list of device part numbers is displayed on the screen, type in the number that corresponds to the correct device part number and press carriage return. If you press carriage return without entering a number, part number 0 will be selected. The device type selected will remain the default device type until a new device type is selected.	3<CR>

Example

```
>A?

DEVICE MANUFACTURERS

0.AMD                6.INTEL          12.OKI           18.TEXAS INSTRUMENT
1.EUROTECHNIQUE      7.MITSUBISHI     13.RICOH         19.TOSHIBA
2.EXEL               8.MOSTEK         14.ROCKWELL      20.VTI
3.FUJITSU            9.MOTOROLA       15.SEEQ          21.XICOR
4.GENERAL INSTRU.   10.NATIONAL       16.SGS-ATES
5.HITACHI           11.NEC            17.SIGNETICS

ENTER MANUFACTURER NUMBER AND RETURN 6<CR>

INTEL DEVICES

0. 2716          5. 2732B        10. P2764        15. 27256
1. 2816          6. P2732A       11. P2764A       16. 27C256
2. 2816A         7. 2764         12. 27128        17. 27512
3. 2732          8. 2764A        13. 27128A       18. 27513
4. 2732A         9. 27C64A       14. P27128A

ENTER PART NUMBER AND RETURN 3<CR>

FAMILY AND PINOUT CODE IS 19/24
```

NOTE

The above list is only an example. The number of devices available at the time you purchase your programmer may be different.

Programming Operations

The following commands allow you to perform programming operations from the remote terminal. With these commands you can check to make sure a device is blank, load data into RAM from master devices, and program blank devices with the master data. The commands are presented in the following order:

Command Name	Command Format
Blank Check	B
Load Single	L [<start address> <end address> <destination address>]
Program Single	P [<start address> <end address> <destination address>]
Load Set	LS [<number of devices>]
Program Set	PS [<number of devices>]
Load Word	LW [<number of devices>]
Program Word	PW [<number of devices>]
Load Long Word	LL [<number of devices>]
Program Long Word	PL [<number of devices>]

Checking for Non-blank Devices

The Blank Check command allows you to check a device prior to programming it to make sure that it is blank.

Command Format: B

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Insert the device to be checked in the socket and lock it into place by pushing the socket lever down.	

3. Type in the Blank Check command letter, B, and press carriage return.
 	
If the blank check is successful, the terminal will display
 	
`FUNCTION COMPLETED`
`BLANK CHECK OK`
 	
and the socket lamp will light green.
 	
If the blank check is unsuccessful, the terminal will display an error message below the `FUNCTION COMPLETED` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.
	`B<CR>`

4. Lift the socket lever and remove the device.	

Loading a Single Master

Use the Load Single command to load (or copy) data from a single master device into the programmer's data RAM for subsequent programming of blank devices, uploading or editing. You can load data starting at programmer RAM address 0, or you can specify a different block of programmer RAM to be loaded with data. You can also select the device address that the first byte of data will be loaded from. Loading the master data into RAM is the first step in programming devices. The second step is to copy the data from RAM to a blank device using the Program Single command (which is explained following this subsection).

Command Format: L [<start address> <end address> <destination address>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Insert the master device into the socket and lock it into place by pushing the socket lever down.	
3. Type in the Load Single command letter, L.	L
4. Type in the first address of RAM that you want data written to, or, to load RAM starting with the default device address (0) and the default programmer RAM address (0), go to step 7. (The space after "L" is optional.)	L 0

Procedure	Example Key Sequence
5. Type a space and then type in the last address of RAM that you want data written to. If you entered a start address, you must enter an end address.	`L 0 1FFF`
6. Type a space and then type in the device address that you want the first byte of data to be loaded from.	`L 0 1FFF 0`

7. Press carriage return to execute the command.
 	
When the load operation is complete, the terminal will display
 	
`FUNCTION COMPLETED`
`THE SUMCHECK (TOTAL) IS: HHHH`
 	
where "HHHH" is the sumcheck of the data that was just loaded into programmer memory, and the socket lamp will light green.
	`L 0 1FFF 0<CR>`
	or
	`L<CR>`

8. Lift the socket lever and remove the master device.	

Programming Single Devices

Use the Program Single command to program a device with the master data loaded into the programmer's RAM with a Load Single command or a download operation. You can program the device with programmer RAM starting with the first byte of programmer RAM, or you can specify another range of programmer RAM to be programmed into the device. You can also select the device address that the first data byte will be written to.

Command Format: P [<start address> <end address> <destination address>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device.	
2. Insert the blank device in the socket and lock the device into place by pressing the socket lever down.	
3. Type in the Program Single command letter, P.	P
4. Type in the first address of RAM that you want programmed into each device, or, to program each device starting with the default programmer RAM address (0) and the default device address (0), go to step 7. (The space after "P" is optional.)	P 0

Procedure	Example Key Sequence
5. Type a space and then type in the last address of RAM that you want programmed into each device. If you entered a start address, you must enter an end address.	P 0 1FFF
6. Type a space and then type in the address of the device memory location that you want the first byte of data to be written to (destination address).	P 0 1FFF 0
7. Press carriage return to execute the command. The terminal will display PROGRAMMING If the programming operation is successful, the terminal will display FUNCTION COMPLETED THE SUMCHECK (TOTAL) IS: HHHH where "HHHH" is the hexadecimal sumcheck of the programmed device. The sumcheck should match the sumcheck of the master data loaded into the programmer's RAM.	P 0 1FFF 0<*CR*> or P<*CR*>

Procedure	Example Key Sequence
If the programming operation is unsuccessful, the terminal will display an error message below the `FUNCTION COMPLETED` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	

8. Lift the socket lever and remove the device.

Loading a Set of Masters

Use the Load Set command to load data from a set of master devices into programmer RAM for subsequent programming into a set of blank devices, uploading, or editing. The data of each successive device is loaded into a successive block of programmer RAM. See the table following the procedure steps for the block of RAM assigned to each device loaded. After loading the set of master devices, you can use the Program Set command (described following this subsection) to program the data into a set of blank devices.

Command Format: LS [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Type in the Load Set command letters, LS.	LS
3. Type in the number of devices in the set to be loaded and press carriage return, or, to load the default number of devices, press carriage return without entering a number of devices. (The space after "LS" is optional.) The terminal will display `INSERT DEVICE NO: 01`	LS 4<CR> or LS<CR>

Procedure	Example Key Sequence

4. Insert the correct device of the set into the socket and press carriage return to execute the load command for the device number displayed, or, to load a different device, type in the number of the device you want to load before pressing carriage return. This feature allows you to load data into any specified block of RAM. When the load operation for the installed device is complete, the terminal will display

```
FUNCTION COMPLETED
INSERT DEVICE NO: NN
```

where "NN" is the number of the next device to be loaded. ("NN" is always the number of the device previously loaded plus 1.)

5. Lift the socket lever and remove the device just loaded. To load another device, return to step 4. If all devices have been loaded, note the displayed sumcheck for later verification of sets of devices programmed with this data.

Example

Load a set of four devices, but skip device number 2 and load device number 3 instead.

```
>LS 4<CR>
INSERT DEVICE NO: 01  <CR>
FUNCTION COMPLETED
INSERT DEVICE NO: 02  3<CR>
FUNCTION COMPLETED
INSERT DEVICE NO: 04  <CR>
FUNCTION COMPLETED
THE SUMCHECK (TOTAL) IS: 1E6F
>
```

Programmer RAM blocks assigned to set devices being loaded:

Device Number	Device Size (no. of bits)					
	16K	32K	64K	128K	256K	512K
1	0000 07FF	0000 0FFF	0000 1FFF	0000 3FFF	0000 7FFF	0000 FFFF
2	0800 0FFF	1000 1FFF	2000 3FFF	4000 7FFF	8000 FFFF	
3	1000 17FF	2000 2FFF	4000 5FFF	8000 BFFF		
4	1800 1FFF	3000 3FFF	6000 7FFF	C000 FFFF		
	•	•	•			
	•	•	•			
	•	•	•			
8	3800 3FFF	7000 7FFF	E000 FFFF			
	•	•				
	•	•				
	•	•				
16	7800 7FFF	F000 FFFF				
	•					
	•					
	•					
32	F800 FFFF					

Programming Sets of Devices

Use the Program Set command to program a set of devices with the contents of the programmer's RAM. This command fills each device with successive blocks of programmer RAM, starting with RAM address 0 and device address 0. See the following table to determine which block of RAM will be programmed into which device. You can use this command to produce sets of devices identical to the masters or you can use this command to merge the data into larger devices. To merge the data into larger devices, select the device type of the larger blank devices before executing the Program Set command.

Command Format: PS [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device.	
2. Type in the Program Set command letters, PS.	PS
3. Type in the number of devices to be programmed and press carriage return, or, to program the default number of devices, press carriage return without entering a number of devices. (The space after "PS" is optional.) The terminal will display INSERT DEVICE NO: 01	PS 4<CR> or PS<CR>

Procedure	Example Key Sequence

4. Insert a blank device into the socket and press carriage return to program the device with the block of RAM corresponding to the device number displayed, or, to program the device with a different block of RAM, type in the device number corresponding to the block of RAM that you want to program into the blank device before pressing carriage return. This feature allows you to program a device with any specified block of RAM. (See the table following the example to determine which block of RAM will be programmed into the device.) The terminal will display

   ```
   PROGRAMMING
   ```

   If the program operation is successful for the installed device, the terminal will display

   ```
   FUNCTION COMPLETED
   INSERT DEVICE NO: NN
   ```

   where "NN" is the number of the next device to be programmed. ("NN" is always the number of the device previously programmed plus 1.) If the last device of the set was just programmed the terminal will display

   ```
   THE SUMCHECK (TOTAL) IS: HHHH
   ```

   The sumcheck (HHHH) should match the sumcheck of the set data loaded into the programmer's RAM.

Procedure	Example Key Sequence
If the program operation is unsuccessful, the terminal will display an error message below the FUNCTION COMPLETED message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	

5. Lift the socket lever and remove the programmed device. To program another device, return to step 4. To exit the Program Set operation, press *<ESC>*. (The programmer will continue programming sets until you press *<ESC>*.)

Example

Program a set of two devices.

```
>PS<CR>
INSERT DEVICE NO: 01  <CR>
PROGRAMMING
FUNCTION COMPLETED
INSERT DEVICE NO: 02  <CR>
PROGRAMMING
FUNCTION COMPLETED
THE SUMCHECK (TOTAL) IS: 96D4

INSERT DEVICE NO: 01  <ESC>
>
```

Programmer RAM blocks programmed into set devices:

Device Number	Device Size (no. of bits)					
	16K	32K	64K	128K	256K	512K
1	0000 07FF	0000 0FFF	0000 1FFF	0000 3FFF	0000 7FFF	0000 FFFF
2	0800 0FFF	1000 1FFF	2000 3FFF	4000 7FFF	8000 FFFF	
3	1000 17FF	2000 2FFF	4000 5FFF	8000 BFFF		
4	1800 1FFF	3000 3FFF	6000 7FFF	C000 FFFF		
8	3800 3FFF	7000 7FFF	E000 FFFF			
16	7800 7FFF	F000 FFFF				
32	F800 FFFF					

Loading Word-Wide Masters

Use the Load Word command to load pairs of 8-bit-wide devices in 16-bit-wide word format. This command loads the data from the first device into even-numbered addresses, starting with address 0, and loads the data from the second device into odd-numbered addresses (see illustration). You can load more than one pair of 8-bit-wide devices (with the data for each pair being loaded into the next available RAM block), as long as the total size of all the devices does not exceed 512K bits. See the table following the procedure for the maximum number of pairs that can be loaded with one Load Word command.

Command Format: LW [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Type in the Load Word command letters, LW.	`LW`

3. Type in the number of devices to be loaded (which must be a multiple of 2) and press carriage return, or, to load the default number of devices, press carriage return without entering a number of devices. (The space after "LW" is optional.) The terminal will display
 	
`INSERT DEVICE NO: 01`
	`LW 4<CR>`
	or
	`LW<CR>`

Procedure	Example Key Sequence

4. Insert the correct device of the pair into the socket (the device containing the low-order bytes for the pair should be loaded first and then the device for the same pair containing the high-order bytes should be loaded). Press carriage return to execute the load command for the device number displayed, or, to load a different device, type in the number of the device you want to load before pressing carriage return. This feature allows you to load any device of any pair. When the load operation for the installed device is complete, the terminal will display
 	
`FUNCTION COMPLETED`
`INSERT DEVICE NO: NN`
 	
where "NN" is the number of the next device to be loaded. ("NN" is always the number of the device previously loaded plus 1.)	

5. Lift the socket lever and remove the device just loaded. To load another device, return to step 4. If all devices have been loaded, note the displayed sumcheck for verification of sets of devices programmed with this data.	

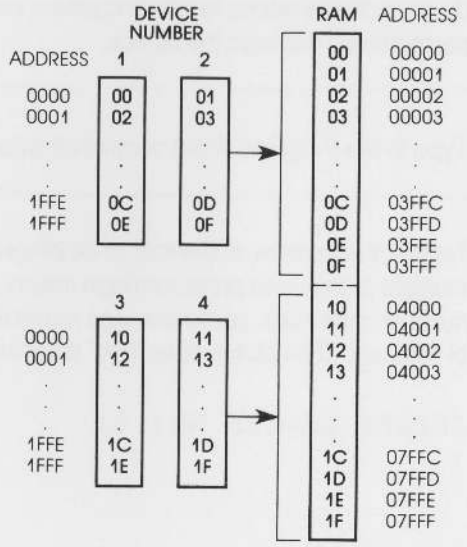
Maximum number of pairs that can be loaded with one Load Word command:

Device Size	Maximum Number of Pairs
16K	16
32K	8
64K	4
128K	2
256K	1

Example

Load two pairs of 64K devices (four devices total).

```
>LW 4<CR>
INSERT DEVICE NO: 01  <CR>
FUNCTION COMPLETED
INSERT DEVICE NO: 02  <CR>
FUNCTION COMPLETED
INSERT DEVICE NO: 03  <CR>
FUNCTION COMPLETED
INSERT DEVICE NO: 04  <CR>
FUNCTION COMPLETED
THE SUMCHECK (TOTAL) IS: E8F4
>
```



Programming Devices Using a Word-Wide Format

Use the Program Word command to program 8-bit-wide devices with data loaded into the programmer's RAM in 16-bit-wide word format. The programmer programs the low bytes (even-addressed bytes) into the first device and the high bytes (odd-addressed bytes) into the second device. You can program as many unique pairs of word-wide devices as were loaded into RAM with the Load Word command (or downloaded) and you can also program as many copies of each pair as you want.

Command Format: PW [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device.	
2. Type in the Program Word command letters, PW.	PW
3. Type in the number of devices to be programmed (which must be a multiple of two) and press carriage return, or, to program the default number of devices, press carriage return without entering a number of devices. (The space after "PW" is optional.) The terminal will display `INSERT DEVICE NO: 01`	PW 4<CR> or PW<CR>

Procedure	Example Key Sequence

4. Insert a blank device into the socket and press carriage return to program the device with the data loaded into RAM that corresponds to the device number displayed, or, to program the device with data loaded into RAM from a different master device, type in the device number corresponding to the data that you want to program into the blank device before pressing carriage return. This feature allows you to program a device with data loaded into RAM from any specified master device. Odd-numbered devices will be programmed with the low-order bytes and even-numbered devices will be programmed with the high-order bytes for the word-wide pair (see the previous illustration). The terminal will display

   ```
   PROGRAMMING
   ```

   If the program operation is successful for the installed device, the terminal will display

   ```
   FUNCTION COMPLETED
   INSERT DEVICE NO: NN
   ```

   where "NN" is the number of the next device to be programmed. ("NN" is always the number of the device previously programmed plus 1.) If the last device of all pairs was just programmed the terminal will display

   ```
   THE SUMCHECK (TOTAL) IS: HHHH
   ```

   The sumcheck (HHHH) should match the sumcheck of all of the word-wide data loaded into the programmer's RAM.

Procedure	Example Key Sequence
If the program operation is unsuccessful, the terminal will display an error message below the FUNCTION COMPLETED message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	

5. Lift the socket lever and remove the programmed device. To program another device, return to step 4. To exit the Program Word operation, press *<ESC>*. The programmer will continue programming pairs of devices until you press *<ESC>*.

Example

Program a pair of word-wide devices.

```
>PW<CR>
INSERT DEVICE NO: 01  <CR>
PROGRAMMING
FUNCTION COMPLETED
INSERT DEVICE NO: 02  <CR>
PROGRAMMING
FUNCTION COMPLETED
THE SUMCHECK (TOTAL) IS: 96D4

INSERT DEVICE NO: 01  <ESC>
>
```

Loading Long-Word-Wide Masters

Use the Load Long Word command to load a set of four 8-bit-wide devices in 32-bit-wide word format. See the illustration following the procedure steps for the order that device data is loaded into RAM. You can load more than one long-word-wide set of 8-bit-wide devices, as long as the total size of all of the devices does not exceed 512K bits. See the table following the procedure for the maximum number of pairs that can be loaded with one Load Long Word command.

Command Format: LL [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected.	
2. Type in the Load Long Word command letters, LL.	LL<*CR*>
3. Type in the number of devices to be loaded (which must be a multiple of 4) and press carriage return, or, to load the default number of devices, press carriage return without entering a number of devices. (The space after "LL" is optional.) The terminal will display `INSERT DEVICE NO: 01`	LL 8<*CR*> or LL<*CR*>

Procedure	Example Key Sequence

4. Insert the correct device into the socket (the devices of each set should be loaded in order, starting with the device containing the lowest-order bytes and ending with the device containing the highest-order bytes). Press carriage return to execute the load command for the device number displayed, or, to load a different device, type in the number of the device you want to load before pressing carriage return. This feature allows you to load any device of any set. When the load operation for the installed device is complete, the terminal will display
 	
`FUNCTION COMPLETED`
`INSERT DEVICE NO: NN`
 	
where "NN" is the number of the next device to be loaded. ("NN" is always the number of the device previously loaded plus 1.)	

5. Lift the socket lever and remove the device just loaded. To load another device, return to step 4. If all devices have been loaded, note the displayed sumcheck for verification of sets of devices programmed with this data.	

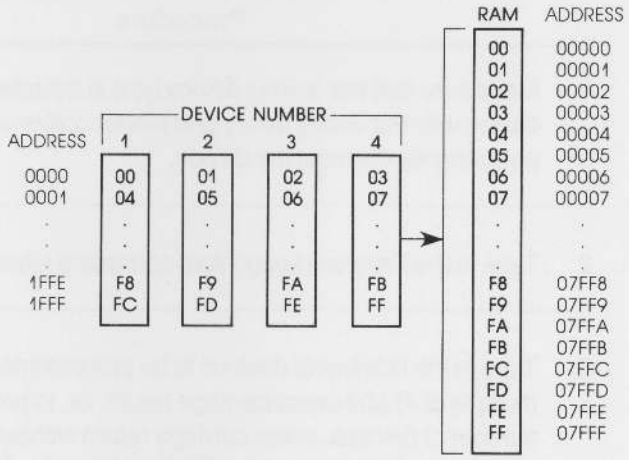
Maximum number of sets that can be loaded with one Load Long Word command:

Device Size	Maximum Number of Sets
16K	8
32K	4
64K	2
128K	1

Example

Load one set of 64K devices (four devices total).

```
>LL 4<CR>
INSERT DEVICE NO: 01  <CR>
FUNCTION COMPLETED
INSERT DEVICE NO: 02  <CR>
FUNCTION COMPLETED
INSERT DEVICE NO: 03  <CR>
FUNCTION COMPLETED
INSERT DEVICE NO: 04  <CR>
FUNCTION COMPLETED
THE SUMCHECK (TOTAL) IS: E8C6
>
```



Programming Devices Using a Long-Word-Wide Format

Use the Program Long Word command to program 8-bit-wide devices with data loaded into the programmer's RAM in 32-bit-wide word format. This command treats four 8-bit-wide devices as a single 32-bit-wide device. The devices of a set are programmed in order; the first device is programmed with the lowest-order bytes of the set and the fourth device of the set is programmed with the highest-order bytes of the set (see the previous illustration). You can program as many unique sets of long-word-wide devices as were loaded into RAM with the Load Long Word command (or a download operation) and you can also program as many copies of each long-word-wide set as you want.

Command Format: PL [<number of devices>]

Procedure	Example Key Sequence
1. Make sure that the correct device type is selected. Programming a device with the wrong family and pinout codes selected could permanently damage the device.	
2. Type in the Program Long Word command letters, PL.	`PL`

3. Type in the number of devices to be programmed (which must be a multiple of 4) and press carriage return, or, to program the default number of devices, press carriage return without entering a number of devices. (The space after "PL" is optional.) The terminal will display
 	
`INSERT DEVICE NO: 01`
	`PL 8<CR>`
	or
	`PL<CR>`

Procedure	Example Key Sequence

4. Insert a blank device into the socket and press carriage return to program the device with the data loaded into RAM that corresponds to the device number displayed, or, to program the device with data loaded into RAM from a different master device, type in the device number corresponding to the data that you want to program into the blank device before pressing carriage return. This feature allows you to program a device with data loaded into RAM from any specified master device. The first device for each set will be programmed with the lowest-order bytes for the set and subsequent devices will be programmed with higher-order bytes, with the fourth device for each set being programmed with the highest order bytes (see the previous illustration). The terminal will display

   ```
   PROGRAMMING
   ```

   If the program operation is successful for the installed device, the terminal will display

   ```
   FUNCTION COMPLETED
   INSERT DEVICE NO: NN
   ```

   where "NN" is the number of the next device to be programmed. ("NN" is always the number of the device previously programmed plus 1.) If the last device of all of the sets was just programmed the terminal will display

   ```
   THE SUMCHECK (TOTAL) IS: HHHH
   ```

Procedure	Example Key Sequence
The sumcheck (HHHH) should match the sumcheck of all of the long-word-wide set data loaded into the programmer's RAM. If the program operation is unsuccessful, the terminal will display an error message below the `FUNCTION COMPLETED` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	
5. Lift the socket lever and remove the programmed device. To program another device, return to step 4. To exit the Program Long Word operation, press *<ESC>*. (The programmer will continue programming sets of devices until you press *<ESC>*.)	

Example

Program one set of long-word-wide devices.

```
>PL<CR>
INSERT DEVICE NO: 01  <CR>
PROGRAMMING
FUNCTION COMPLETED
INSERT DEVICE NO: 02  <CR>
PROGRAMMING
FUNCTION COMPLETED
INSERT DEVICE NO: 03  <CR>
PROGRAMMING
FUNCTION COMPLETED
INSERT DEVICE NO: 04  <CR>
PROGRAMMING
FUNCTION COMPLETED
THE SUMCHECK (TOTAL) IS: C6D8

INSERT DEVICE NO: 01  <ESC>
>
```

Verifying Programmed Devices

Use the following commands to verify device data against data loaded into the programmer's RAM. Each device is automatically verified after it is programmed; however, if you wish to verify devices at another time, you can use these commands. Before you use any of the following verify commands, you must load the programmer's RAM with the data which the devices are to be verified against using either a load or a download command. The verify commands are presented in the following order:

Command Name	Command Format
Verify Single	V [<Vcc Selection>] [<start address> <end address> <destination address>]
Verify Set	VS [<Vcc Selection>] [<number of devices>]
Verify Word	VW [<Vcc Selection>] [<number of devices>]
Verify Long Word	VL [<Vcc Selection>] [<number of devices>]

Verifying Single Devices

The Verify Single command allows you to verify a single device against the master data loaded into the programmer's memory. You can verify a device beginning with master data at programmer RAM address 0, or you can select a different block of RAM to verify the device against. You can also select the address of the device memory location where the verify operation will begin.

Command Format: V [<Vcc Selection>] [<start address> <end address> <destination address>]

Procedure	Example Key Sequence
1. Load the master data into the programmer's RAM using either a Load Single operation or a download operation.	
2. Make sure that the correct device type is selected.	
3. Insert the device to be verified into the socket.	
4. Type in the Verify Single command letter, V.	V

Procedure	Example Key Sequence
5. Type a space and then type the letter (L or H) that corresponds to the proper Vcc high and low settings shown in the following table (in this case, the space following the command letter V is required), or, to verify the installed device with the voltage settings recommended by the manufacturer, do not type anything for this parameter.	V L
6. Type in the first address of RAM that you want the device to be verified against, or, to verify the entire device data, go to step 9.	V L 0
7. Type a space and then type in the last address of RAM that you want the device to be verified against. If you entered a start address, you must enter an end address.	
8. Type a space and then type in the first address of device memory that you want to be verified against the selected block of RAM.	V L 0 1FFF 0

Selection	High Voltage	Low Voltage
L	5.2 V	4.8 V
H	5.5 V	4.5 V

Procedure	Example Key Sequence

9. Press carriage return to execute the verify command.
 	
If the operation is successful, the terminal will display
 	
`FUNCTION COMPLETED`
`THE SUMCHECK (TOTAL) IS: HHHH`
 	
where "HHHH" is the hexadecimal summation of the data contained in the verified device.
 	
If the operation is unsuccessful, the terminal will display an error message below the `FUNCTION COMPLETED` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.
	`V L 0 1FFF 0<CR>`
	or
	`V L<CR>`

10. Remove the verified device. To verify another device of the same type against the same master data, return to step 3.	

Verifying Sets of Devices

The Verify Set command allows you to verify sets of devices (each containing different data) against the master set data stored in the programmer's RAM.

Command Format: VS [<Vcc Selection>] [<number of devices>]

Procedure	Example Key Sequence
1. Load the master set data into the programmer's RAM using either a Load Set command or a download operation.	
2. Make sure that the correct device type is selected.	
3. Type in the Verify Set command letters, VS.	VS
4. Type in the letter (L or H) that corresponds to the proper Vcc high and low settings shown in the following table, or, to verify the installed device with the voltage settings recommended by the manufacturer, do not type anything for this parameter. (The space after "VS" is optional.)	VS L

Selection	High Voltage	Low Voltage
L	5.2 V	4.8 V
H	5.5 V	4.5 V

Procedure	Example Key Sequence

5. Type a space, type in the number of devices to be verified and press carriage return, or, to verify the default number of devices, press carriage return without entering a number of devices. The terminal will display
 	
`INSERT DEVICE NO: 01`
	`VS L 4<CR>`
	or
	`VS L<CR>`

6. Insert the correct device of the set into the socket (see the table in Programming Sets of Devices for information on which RAM address corresponds to which device) and press carriage return to execute the command for the device number displayed, or, to verify a different device, type in the number of the device before pressing carriage return. This feature allows you to verify a partial set.
 	
If the operation is successful, the terminal will display
 	
`FUNCTION COMPLETED`
`INSERT DEVICE NO: NN`
 	
where "NN" is the number of the next device to be verified. ("NN" is always the number of the device previously verified plus 1.)	

Procedure	Example Key Sequence
If the last device of the set was just verified, the terminal will display `THE SUMCHECK (TOTAL) IS: HHHH` where "HHHH" is the hexadecimal summation of the data contained in the set that was just verified. If the operation is unsuccessful, the terminal will display an error message below the `FUNCTION COMPLETED` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	

7. Remove the verified device. To verify another device, return to step 6. To exit the Verify operation, press *<ESC>*. The programmer will continue to verify device sets until you press *<ESC>*.

Verifying Word-Wide Devices

Use the Verify Word command to verify devices against data that is loaded into RAM in 16-bit-wide word format. This command compares two 8-bit-wide devices to 16-bit-wide data stored in the programmer's RAM. The data in odd-numbered devices is compared to low-order bytes (even-addressed bytes) stored in RAM, and the data in even-numbered devices is compared to high-order bytes (odd-addressed bytes) stored in RAM. More than one pair of devices can be compared, provided the master data for all pairs to be verified is loaded into the programmer's RAM. Also, multiple copies of each pair can be verified with one command.

Command Format: VW [<Vcc Selection>] [<number of devices>]

Procedure	Example Key Sequence
1. Load the master set data into the programmer's RAM using either a Load Word command or a download operation.	
2. Make sure that the correct device type is selected.	
3. Type in the Verify Word command letters, VW.	VW

Procedure	Example Key Sequence
4. Type in the letter (L or H) that corresponds to the proper Vcc high and low settings shown in the following table, or, to verify the installed device with the voltage settings recommended by the manufacturer, do not type anything for this parameter. (The space after "VW" is optional.)	`VW L`

5. Type a space, type in the number of devices to be verified (which must be a multiple of 2) and press carriage return, or, to verify the default number of devices, press carriage return without entering a number of devices. The terminal will display
`INSERT DEVICE NO: 01`
	`VW L 4<CR>`
	or
	`VW L<CR>`

6. Insert the correct device of the set into the socket; odd-numbered devices will be verified against low-order bytes for the pair and even-numbered devices will be verified against high-order bytes for the pair. Press carriage return to execute the command for the device number displayed, or, to verify a different device, type in the number of the device before pressing carriage return. This feature allows you to verify any device of any pair against the loaded data.	

Selection	High Voltage	Low Voltage
L	5.2 V	4.8 V
H	5.5 V	4.5 V

Procedure	Example Key Sequence

If the operation is successful, the terminal will display
 	
`FUNCTION COMPLETED`
`INSERT DEVICE NO: NN`
 	
where "NN" is the number of the next device to be verified. ("NN" is always the number of the device previously verified plus 1.)
 	
If the last device of all of the pairs was just verified, the terminal will display
 	
`THE SUMCHECK (TOTAL) IS: HHHH`
 	
where "HHHH" is the hexadecimal summation of all of the data contained in all of the pairs.
 	
If the operation is unsuccessful, the terminal will display an error message below the `FUNCTION COMPLETED` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	

7. Remove the verified device. To verify another device, return to step 6. To exit the Verify operation, press *<ESC>*. The programmer will continue to verify devices until you press *<ESC>*.	

Verifying Long-Word-Wide Devices

Use the Verify Long Word command to verify device data against data that is loaded into RAM in 32-bit-wide word format. This command compares four 8-bit-wide devices to 32-bit-wide data in the programmer's RAM. (See Loading Long Word-Wide Masters for information on which device number corresponds to which block of RAM data.) More than one unique set of devices can be verified with one Verify Long Word command, provided that the data for all devices is loaded into the programmer's RAM, and multiple copies of each set can be verified with one Verify Long Word command.

Command Format: VL [<Vcc Selection>] [<number of devices>]

Procedure	Example Key Sequence
1. Load the master data into the programmer's RAM using either a Load Long Word command or a download operation.	
2. Make sure that the correct device type is selected.	
3. Type in the Verify Long Word command letters, VL.	VL

Procedure	Example Key Sequence
4. Type in the letter (L or H) that corresponds to the proper Vcc high and low settings shown in the following table, or, to verify the installed device with the voltage settings recommended by the manufacturer, do not type anything for this parameter. (The space after "VL" is optional.)	VL L
5. Type a space, type in the number of devices to be verified (which must be a multiple of 4) and press carriage return, or, to verify the default number of devices, press carriage return without entering a number of devices. The terminal will display INSERT DEVICE NO: 01	VL L 8<CR> or VL L<CR>
6. Insert the correct device of the set into the socket; the devices will be verified in the same order that they were programmed in originally. Press carriage return to execute the command for the device number displayed, or, to verify a different device, type in the number of the device before pressing carriage return. This feature allows you to verify a partial set.	

Selection	High Voltage	Low Voltage
L	5.2 V	4.8 V
H	5.5 V	4.5 V

Procedure	Example Key Sequence

If the operation is successful, the terminal will display
 	
`FUNCTION COMPLETED`
`INSERT DEVICE NO: NN`
 	
where "NN" is the number of the next device to be verified. ("NN" is always the number of the device previously verified plus 1.)
 	
If the last device of all of the sets was just verified, the terminal will display
 	
`THE SUMCHECK (TOTAL) IS: HHHH`
 	
where "HHHH" is the hexadecimal summation of all of the data contained in the sets just verified.
 	
If the operation is unsuccessful, the terminal will display an error message below the `FUNCTION COMPLETED` message and the socket lamp will light red. See the Error Messages section for an explanation of the error message.	

7. Remove the verified device. To verify another device, return to step 6. To exit the Verify operation, press *<ESC>*. The programmer will continue to verify device sets until you press *<ESC>*.

Editing Memory

The following commands allow you to edit data that has been loaded into the 201 EPROM Programmer's RAM. With these commands, you can display the data, edit individual addresses, fill a segment of memory with data, insert new data, delete data, copy data, and search memory for up to 8 bytes of data. The commands are presented in the following order:

Command Name	Command Format
Memory Display	M <start address> <end address>
Memory Modify	M <start address>
Memory Fill	M <start address> <end address> <data1> [<data2...data8>]
Insert	I <start address> <data1> [<data2...data8>]
Delete	D <start address> [<end address>]
Transfer (Copy)	T <start address> <end address> <destination address>
Memory Search	MS <start address> <end address> <data1> [<data2...data8>]

You may also want to use the Sumcheck (Total) command to calculate a sumcheck of the data stored in the programmer's memory after you have finished editing and are ready to program devices. The Sumcheck (Total) command is explained following this subsection.

Displaying Memory

Use the Memory Display command to display the contents of RAM on the terminal screen. The RAM addresses are arranged on the screen in a table format, with each row containing 16 bytes of consecutive data (see example). To determine the address of a byte of RAM, add the 5-character RAM address shown at the very left of the line containing the data to the 2-character address shown at the top of the column containing the data. The ASCII-CODE column on the far right contains the ASCII equivalent of the hexadecimal data on each line.

Command Format: M <start address> <end address>

Procedure	Example Key Sequence
1. Type the Memory Display command letter, M.	M
2. Type in the address of the first byte of RAM that you want to display on the screen.	M 0
3. Type a space, then the address of the last byte of RAM that you want to display on the screen followed by a carriage return.	M 0 1F<*CR*>
4. Type CTRL-S to halt the display and CTRL-Q to resume the display.	

Example

Display the contents of RAM addresses 0 through 1F:

```
>M 0 1F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 32 1A 5B 47 43 52 03 4B CB 20 54 33 4A 11 20 52  2.[GCR.K. T3J. R
00010 52 20 11 4A 33 54 20 CB 4B 03 AA 52 43 47 4B 1A  R.J3T .K..RCGK..
```

NOTE

To determine the memory address of a data byte, add the column heading of the data byte column to the row heading (5-character hexadecimal number at the far left of each line).

A period under the ASCII-CODE heading represents hexadecimal data without a printable ASCII equivalent.

Modifying Single Memory Locations

Use the Memory Modify command to display and edit the contents of RAM locations one-by-one. The Memory Modify command causes the programmer to enter an editing mode which allows you to display the contents of RAM addresses one-by-one and type in new data for the address displayed. To exit Memory Modify mode and return to the command prompt (>), press *<ESC>*.

Command Format: M <start address>*<CR>*
<new data>*<CR>*
.
.
.
<ESC>

Procedure	Example Key Sequence
1. Type the Memory Modify command letter, M.	M
2. Type in the address of the first memory location you want to modify and press carriage return.	M 6*<CR>*
3. After the RAM address and its contents have been displayed, type in the new data and press carriage return to accept it. The new data will appear to the right of the old data. If you do not want to modify the address displayed, but want to display the next RAM address for modification, press carriage return without entering new data.	E5*<CR>* or *<CR>*

Procedure	Example Key Sequence
4. To modify the next memory location, return to step 3.	
5. To exit Memory Modify mode and return to the command prompt (>), press <*ESC*>. If you press <*ESC*> after typing in new data but before pressing carriage return to accept the new data, the new data for the last address displayed will be ignored.	

Example

Enter Memory Modify mode starting at address 6 and type in new data for addresses 6, 7, and 8:

```
>M 6<CR>
00006  F1   E5<CR>
00007  FF   21<CR>
00008  00   21<CR>
00009  43   <ESC>
```

Filling a Segment of Memory

Use the Memory Fill command to fill (replace) a segment of RAM with up to eight bytes of new data. The original data in these RAM locations are overwritten with the new data.

Command Format: M <start address> <end address> <data1> [<data2>...<data8>]

Procedure	Example Key Sequence
1. Type the Memory Fill command letter, M.	M
2. Type in the address of the first byte of RAM you want to fill with new data.	M 20
3. Type a space and then type in the address of the last byte of RAM you want to fill with new data.	M 20 2F
4. Type a space and then type in the data to be input into the specified memory range followed by a carriage return. Separate each hexadecimal byte with a space. You can insert up to eight hexadecimal bytes (or eight ASCII characters) with one Memory Fill command. If you input more data than can fit in the range specified, the excess data is ignored.	M 20 2F 11 22<*CR*>

Example

Display memory contents before fill:

```
>M 20 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00020 FF FF FF FF EE EE EE EE DD DD DD DD CC CC CC CC  ................
00030 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88  ................
```

Fill memory:

```
>M 20 2F 11 22<CR>
```

Display memory contents after fill:

```
>M 20 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00020 11 22 11 22 11 22 11 22 11 22 11 22 11 22 11 22  "."."."."."."."."
00030 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88  ................
```

Inserting New Data

Use the Insert command to insert new data into RAM. The original data is not overwritten by this command, as it is with the Memory Modify and the Memory Fill commands, but is shifted up to accommodate the new data. Data at the end of memory is lost.

Command Format: I <start address> <data1> [<data2>...<data8>]

Procedure	Example Key Sequence
1. Type the Insert command letter, I.	I
2. Type in the starting address where you want the new data to be inserted.	I 5
3. Type a space and then type in the new data followed by a carriage return. Separate each hexadecimal byte with a space. You can insert up to eight hexadecimal bytes (or eight ASCII characters) with one Insert command.	I 5 11 11 11 11<*CR*>

Example

Display memory contents before insertion of new data:

```
>M 0 1F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 FF FF FF FF EE EE EE EE DD DD DD DD CC CC CC CC  ................
00010 BB BB BB BB AA AA AA AA 99 99 99 99 88 88 88 88  ................
```

Insert new data:

```
>I 5 11 11 11 11<CR>
```

Display memory contents after insertion of new data:

```
>M 0 1F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 FF FF FF FF EE 11 11 11 11 EE EE EE DD DD DD DD  ................
00010 CC CC CC CC BB BB BB BB AA AA AA AA 99 99 99 99  ................
```

Deleting Data

Use the Delete command to delete data bytes from memory. Data following the deleted addresses are shifted down to fill in the deleted addresses. FF is filled in the vacated addresses at the end of RAM.

Command Format: D <start address> [<end address>]

Procedure	Example Key Sequence
1. Type the Delete command letter, D.	D
2. Type in the address where you want to begin deleting data.	D 4
3. To delete a single byte of data, press carriage return without entering an end address. To delete a range of data, type a space, then the address of the last memory byte that you want to delete followed by a carriage return.	D 4<CR> or D 4 7<CR>

Example

Display memory contents before deletion:

```
>M 0 1F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 66 66 66 66 11 11 11 11 77 77 77 77 77 77 77 77  FFFF....WWWWWWWW
00010 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88  ................
```

Delete data:

```
>D 4 7<CR>
```

Display memory contents after deletion:

```
>M 0 1F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 66 66 66 66 77 77 77 77 77 77 77 77 88 88 88 88  FFFFWWWWWWWW....
00010 88 88 88 88 88 88 88 88 88 88 88 88 FF FF FF FF  ................
```

Transferring (Copying) Memory

Use the Transfer command to copy a block of RAM data from one RAM location to another. This command copies the block specified by the start address and end address to the destination address. The original data starting at the destination address is overwritten by the copied block.

Command Format: T <start address> <end address> <destination address>

Procedure	Example Key Sequence
1. Type the Transfer command letter, T.	T
2. Type in the address of the first byte of RAM to be copied.	T 0
3. Type a space and then type in the address of the last byte of RAM to be copied.	T 0 F
4. Type a space and then type in the address that you want the first byte of the specified data range to be copied to, or the destination address, and press carriage return. The data will be copied into consecutive RAM addresses, starting with the destination address.	T 0 F 25<*CR*>

Example

Display memory contents before transfer:

```
>M 0 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00000 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  ................
00010 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  WWWWWWWWWWWWWWWW
00020 66 66 66 66 66 44 44 44 44 44 44 44 44 44 44 44  FFFFFDDDDDDDDDDD
00030 44 44 44 44 44 11 11 11 11 11 11 11 11 11 11 11  DDDDD...........
```

Transfer data:

```
>T 0 F 25<CR>
```

Display memory contents after transfer:

```
M 0 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    ASCII-CODE
00000 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  ................
00010 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77  WWWWWWWWWWWWWWWW
00020 66 66 66 66 66 99 99 99 99 99 99 99 99 99 99 99  FFFFF...........
00030 99 99 99 99 99 11 11 11 11 11 11 11 11 11 11 11  ................
```

Searching Memory

Use the Memory Search command to find a block of data up to eight hexadecimal bytes long.

Command Format: MS <start address> <end address> <data1> [<data2>...<data8>]

Procedure	Example Key Sequence
1. Type the Memory Search command letters, MS.	MS
2. Type in the address of the memory location where you want the search to begin.	MS 0
3. Type a space and then type in the memory address where you want the search to end.	MS 0 3F
4. Type a space and then type in the data that you want to search for and press carriage return. Separate each hexadecimal byte with a space. You can type in up to eight hexadecimal bytes (eight ASCII characters).	MS 0 3F 12 34<*CR*>

Example

Search memory block 0 through 3F for the hexadecimal data 12 34:

```
>MS 0 3F 12 34<CR>
MEMORY MATCH AT ADDRESS 000038
```

Display memory block searched:

```
>M 0 3F<CR>
      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F     ASCII-CODE
00000 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77   WWWWWWWWWWWWWWWW
00010 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77   WWWWWWWWWWWWWWWW
00020 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77   WWWWWWWWWWWWWWWW
00030 77 77 77 77 77 77 77 77 12 34 77 77 77 77 77 77   WWWWWWWW.4WWWWWW
```

Sumchecking Data

Use the following commands to cause the programmer to sumcheck (total) or checksum (EXOR) the data stored in the programmer's RAM. These commands are useful for obtaining a new sumcheck of any range of RAM, or verifying that the data stored in RAM is the same as data programmed into a device or data downloaded through the serial port.

Performing a Sumcheck (Total)

Use the Sumcheck (Total) command to calculate a hexadecimal summation of data stored in the programmer's RAM. This sumcheck can be used to verify that data was downloaded to the programmer correctly, to sumcheck only a portion of memory, or to produce a current sumcheck after you have edited memory. A sumcheck is automatically performed after most programming operations, so you should not need to use this command when programming or verifying devices from a master device.

Command Format: CT <start address> <end address> [<destination address>]

Procedure	Example Key Sequence
1. Type in the Sumcheck (Total) command letters, CT.	`CT`
2. Type in the address of the first byte of RAM to be checked by the sumcheck operation.	`CT 0`
3. Type a space and then type in the address of the last byte of RAM to be checked.	`CT 0 3FFF`

Procedure	Example Key Sequence
4. To perform the sumcheck without storing the sumcheck value in RAM, press carriage return, or, to store the sumcheck value in RAM, type a space and then type in the RAM address that you want the sumcheck value to be stored in and press carriage return. If you store the sumcheck value in RAM, it will overwrite any data previously stored in the RAM locations to which the sumcheck is written. Make sure that there is no data that you want to retain located in the addresses that the sumcheck will be written to. The sumcheck (total) is calculated and then displayed on the screen. The sumcheck is also stored in memory if you specified a destination address.	CT 0 3FFF<*CR*> or CT 0 3FFF 4000<*CR*>

Example

Calculate the sumcheck total of memory addresses 0 through 3FFF and store the sumcheck value in address 4000:

```
>CT 0 3FFF 4000<CR>
THE SUMCHECK (TOTAL) IS:  0201
```

NOTE

The two-byte sumcheck is stored in memory addresses 4000 and 4001, with the low-byte (01 in above example) stored in 4000 and the high-byte (02 in above example) stored in 4001.

Performing an Exclusive-OR Checksum

Use the Checksum (EXOR) command to calculate an exclusive-OR checksum of RAM. You may wish to use this command instead of the sumcheck (total) command if data being downloaded to the programmer includes an exclusive-OR checksum, or if you prefer to work with an exclusive-OR summation. The exclusive-OR checksum is a 1-byte hexadecimal value representing the result of exclusive-ORing each successive byte of the data range specified. For example,

```
  0000 0001
+ 1111 1111
  1111 1110
```

Command Format: CX <start address> <end address> [<destination address>]

Procedure	Example Key Sequence
1. Type in the Checksum (EXOR) command letters, CX.	CX
2. Type in the address of the first byte of RAM to be checked by the checksum operation.	CX 0
3. Type a space and then type in the address of the last byte of RAM to be checked by the checksum operation.	CX 0 3FFF

Procedure	Example Key Sequence
4. To perform the checksum operation without storing the checksum value in RAM, press carriage return. To store the checksum in RAM, type a space, then the RAM address that you want the checksum to be stored in followed by a carriage return. If you store the checksum value in RAM, it will overwrite any data previously stored in that RAM location. The checksum (EXOR) value is calculated and then displayed on the screen. The checksum is also stored in RAM if you specified a destination address.	CX 0 3FFF<*CR*> or CX 0 3FFF 4000<*CR*>

Example

Calculate the checksum of memory addresses 0 through 3FFF and store the checksum value in address 4000:

```
>CX 0 3FFF 4000<CR>
THE CHECKSUM (EXOR) IS:  FE
```

4. Computer Remote Control

Introduction

The 201 EPROM Programmer can be controlled via a remote host computer using the Computer Remote Control (CRC) commands described in this section. The Computer Remote Control commands were designed to be incorporated into a software program, or driver, which would allow an operator to control the 201 EPROM Programmer using the software program. The driver generates and sends commands to the programmer which executes the commands and returns a response character, and in some cases also data, which the driver then reacts to and uses to generate messages and prompts for the operator.

The CRC, or driver, commands are ASCII characters which are summarized in the command summary table following this introduction and described in more detail in the pages that follow. The commands are grouped into the following subsections:

- ☐ ENTERING AND EXITING COMPUTER REMOTE CONTROL MODE — This subsection explains how to transfer control of the programmer to a remote host computer and how to return control of the programmer to the front panel.
- ☐ VERIFYING PROPER COMMUNICATION — This subsection describes the commands used to verify that proper communication has been established between the computer and the programmer.
- ☐ PROGRAMMING OPERATIONS — This subsection describes the commands used to set parameters prior to performing programming operations, load data into RAM, program blank devices, test devices and verify that devices were programmed properly.
- ☐ TRANSFERRING DATA — This subsection describes the commands used to upload or download data to or from the remote host computer.

- ☐ INQUIRING ABOUT OPERATING AND ERROR STATUS — This subsection describes the commands used to inquire about parameters and options selected and error status.
- ☐ EDITING RAM DATA — This subsection describes the commands used to edit data stored in the programmer's RAM.
- ☐ DATA TRANSLATION FORMATS — This subsection describes the five data translation formats available for the 201 EPROM Programmer.

Symbols and Conventions

The following is a list of symbols and conventions used in this section of the manual.

A	Capital letters in a command must be sent to the programmer through the serial port to execute the command.
h	A lower-case h represents a hexadecimal digit.
n	A lower-case n represents a decimal digit.
ff	Two lower-case f's represent the family code of the device.
pp	Two lower-case p's represent the pinout code of the device.
<CR>	This symbol represents a carriage return, which must follow each command entry.
<ESC>	This symbol represents the escape key.

Computer Remote Control Command Summary

Command Format	Description	Command Format	Description
M	Enter CRC mode (automatic setup of communications protocol)	=	Disable timeout
		hhhhhhhhW	Set I/O address offset
Z	Exit CRC mode	hhM	Set I/O record size
<CR>	Execute command	cnnA	Select control code and data translation format
<ESC>	Abort operation		
		I	Input data from host computer
hhU	Set nulls	O	Output data to host computer
H	No operation	C	Compare data
ffpp@	Select family and pinout codes	B8]	View all family and pinout codes
hhhh:	Set beginning device address	[	View current family and pinout codes
hhhh<	Set beginning RAM address		
hhhh;	Set block size	CC]	View current electronic ID family and pinout codes
L	Load master device	CD]	View current electronic ID code
P	Program device	R	Respond with device parameters
		X	View error codes
T	Test device for illegal bits	G	View software configuration
B	Blank check device		
V	Verify device	hhhh?	Split RAM data
S	Perform sumcheck	hhhh>	Shuffle RAM data
		\	RAM to RAM block move
		^	Clear all RAM to zero

Response Characters

The response characters, summarized in the table below, are characters that the programmer sends to the host computer after attempting to execute a command. The programmer's response to a command will always contain a response character followed by a carriage return. In addition, the response may contain data and a line feed and nulls (ASCII character 00). Whether or not the response contains a line feed or nulls is dependent upon the null count setting. How to set the null count is described in the Verifying Proper Communication subsection.

Programmer Response Characters

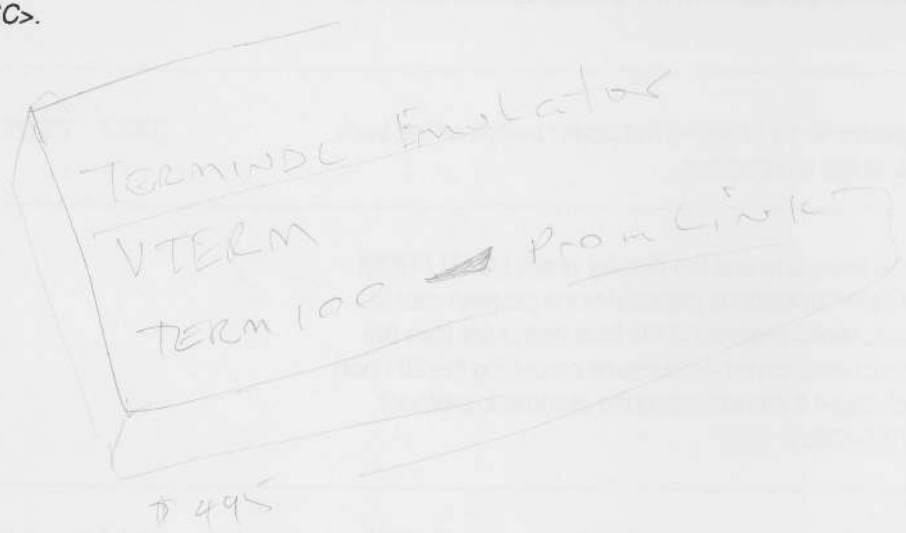
Character	Name	Description
>	Prompt	Sent upon entering Computer Remote Control mode, when <*ESC*> halts a command, or when the programmer successfully executes a command. The programmer then transmits a carriage return.
F	Fail	Sent when the programmer fails to execute a command. The programmer then transmits a carriage return. The cause of the failure can be determined by displaying the error code with the "X" command.
?	Question	Sent when the programmer does not understand a command or the command was invalid. The programmer then transmits a carriage return.

Using PROMlink™ to Operate the 201

If you are using PROMlink™ as the driver program to operate the 201, you must set the programmer's serial port (I/O) settings to match PROMlink's™ I/O settings before entering Computer Remote Control mode. Refer to the Front Panel Operation section for instructions on how to set the 201's serial port (I/O) settings. If your version of PROMlink™ does not support the 201 EPROM Programmer specifically, you can still use PROMlink™ to operate the 201 by selecting the Series 22 as the programmer type.

Aborting an Operation

Any of the Computer Remote Control commands except Split and Shuffle can be aborted by pressing *<ESC>*.



Entering and Exiting Computer Remote Control Mode

The following procedures describe how to transfer control of the programmer to a host computer and then how to return control of the programmer to the front panel.

Entering Computer Remote Control Mode

Procedure	201 Displays
1. Connect the programmer to the computer as described in the RS232 Port Cable Connections discussion in the Getting Started section of this manual.	
2. Power up the programmer by pressing the power switch on the back of the programmer to the ON position.	SELF TESTING . . . . .
3. When the self test is complete and the display reads LOAD FROM MASTER, set the communications protocol of the programmer to match the communications protocol of the host computer (see the Front Panel Operation section for instructions on setting the 201 port settings), or skip to step 4 if you are using the automatic protocol setting command (M<*CR*>).	
4. Scroll through the main menu to RS232 PORT and press ENTER.	RS232 PORT COMPUTER CONTROL

Procedure	201 Displays
5. Scroll through the RS232 PORT menu (if necessary) to COMPUTER CONTROL and press ENTER.	COMPUTER CONTROL
6. See the driver instructions for entering the driver program if you are using a previously written driver. If you are writing your own driver, you can establish communication between the programmer and the host computer by either of two methods. You can cause the programmer to set the communications protocol automatically (and thus avoid requiring the operator to set the communications protocol manually) by sending an "M<*CR*>" to the programmer. Or if the communications protocol will always be set correctly prior to this step, you can send any character to the programmer to establish communications. A prompt (>) character will be returned to the host computer if the port settings were set correctly and the programmer and computer are communicating. The programmer will display COMPUTER CONTROL until control of the programmer is returned to the front panel.	COMPUTER CONTROL

Exiting Computer Remote Control Mode

To exit Computer Remote Control (CRC) mode, use the appropriate exit command specified by the CRC driver program or press any 201 front panel key. If you are writing your own driver, the driver's exit command must send a "Z<*CR*>" to the programmer to cause it to exit CRC mode. The user should still be able to exit CRC mode by pressing a programmer key when using any driver.

After exiting CRC mode, the programmer displays

```
RS232 PORT
```

You can now continue to operate the programmer using the front panel keys.

Verifying Proper Communication

The first thing that the driver program should do is verify that the computer and the programmer are communicating properly. To do this, use the following commands.

Command Format	Command Name	Data Output to Host	Description
hhU	Set Nulls	none	Sets the number of nulls (hh) sent to the host computer after the response character and carriage return, and enables line feeds. Values 0 through FE enable line feeds and cause the specified number of nulls (0 through FE) to be sent to the computer. The default value, FF, causes no nulls and no line feeds to be sent.
H	No Operation	none	Causes the programmer to return a prompt (>). The driver program should send this command to the programmer to verify that communications are established.

Programming Operations

Use the following commands to set the family and pinout codes, set programming parameters and options, load device data into memory, program devices, test devices and verify programmed devices.

Command Format	Command Name	Data Output to Host	Description
Setting Programming Parameters and Options			
ffpp@	Select Family and Pinout Codes	none	Selects the family (ff) and pinout (pp) codes for the device to be loaded, programmed, tested or verified. The correct family and pinout codes must be selected prior to programming a device, otherwise damage to the device could result.
hhhh:	Set Beginning Device Address	none	Sets the first device address from which or to which data is to be transferred. The default beginning device address is 0. This command is also used to set the destination address for a RAM to RAM block move.
hhhh<	Set Beginning RAM Address	none	Sets the first RAM address to be used for data transfers. This command is also used to set the beginning RAM address for a block move. Setting the beginning RAM address clears any previously entered block size. The default beginning RAM address is 0.
hhhh;	Set Block Size	none	Sets the block size (number of bytes) to be used in data transfers. The default block size is the device word limit for device related operations. There is no default value for block moves.

Command Format	Command Name	Data Output to Host	Description
Programming Devices			
L	Load Master Device	none	Loads data into RAM from a master device using the defined beginning RAM address, block size and beginning device address.
P	Program Device	none	Programs and verifies a device with the data loaded into RAM using the defined beginning RAM address, block size and beginning device address. Always make sure that the correct family and pinout codes are selected prior to programming a device, or damage to the device could result. The program operation in CRC does not include an illegal bit test or a blank check.
Testing Devices			
T	Test Device for Illegal Bits	none	Tests the device for illegal bits (programmed bits that are not stored in RAM). If an illegal bit is found, the 201 returns an "F" to the computer and stores the appropriate error code in programmer memory. To display the stored error code, use the "X" command.

Command Format	Command Name	Data Output to Host	Description
B	Blank Check Device	none	Checks the installed device for programmed bits. If a programmed bit is found, the 201 returns an "F" to the computer and stores the appropriate error code in programmer memory. To display the stored error code, use the "X" command.
V	Verify Device	none	Compares RAM data with data programmed into a device using the defined beginning RAM address, block size and beginning device address.
S	Sumcheck Data	hhhh	Calculates the 4-digit hexadecimal summation of RAM data from the beginning RAM address (or zero if no beginning RAM address is selected) to the block size. If no block size is selected, then the 201 uses either the word limit of the selected device or the end of RAM, whichever is smaller, to determine the block size.

Transferring Data

Use the following commands to input data to the programmer from the host computer or output data from the programmer to the host computer.

Command Format	Command Name	Data Output to Host	Description
=	Disable Timeout	none	Disables 25 second I/O timeout.
hhhhhhhhW	Set I/O Address Offset	none	Specifies an address to be subtracted from all addresses input to the programmer and added to all addresses output from the programmer. This allows for the adjustment of RAM addresses to the address range of a larger memory. The default address offset is the first address received (input). To reset default offset, enter all F's for the address offset.
hhM	Set I/O Record Size	none	Defines the number of bytes per record in serial port output operations. The default value is 16 bytes per record for translation formats with a variable record size.
cnnA	Select Control Code and Data Translation Format	none	Selects the instrument control code (c) and data translation format (nn) used for input and output of data. See the Data Translation Formats subsection at the end of the CRC section for details on the instrument control code and data translation formats. The default format is Intel Intellec 8/MDS, format 83, and the default control code is 0.

Command Format	Command Name	Data Output to Host	Description
I	Input Data from Host	none	Instructs the programmer to accept formatted data from the host computer.
O	Output Data to Host	hhhhh...	Translates data into the selected format and outputs this data to the computer. The programmer will stop outputting data upon receipt of an X-OFF character, DL3 (CTRL-S), and will resume output of data upon receipt of the X-ON character, DC1 (CTRL-Q).
C	Compare Data	none	Compares data in RAM with the data received through the serial port from the host computer.

Inquiring About Operating and Error Status

Use the following commands to cause the programmer to output all family and pinout codes available, the status of the current device selected, the electronic ID settings, error status, and the programmer's software configuration.

Command Format	Command Name	Data Output to Host	Description
B8]	View all Family and Pinout Codes	hh...	Outputs all family and pinout codes available for programming on the 201.
[	View Current Family and Pinout Codes	ffpp	Outputs the family (ff) and pinout (pp) codes of the device currently selected.
CC]	View Current Electronic ID Family and Pinout Codes	ffpp	Outputs the electronic ID family (ff) and pinout (pp) codes from the most recent, successful, electronic ID operation.
CD]	View Current Electronic ID Code	h1...h16	Outputs the sixteen byte electronic identifier code of the installed device.

Command Format	Command Name	Data Output to Host	Description
R	Respond with Device Parameters	wl/ws/n	Outputs the status of the device selected by the current family and pinout codes. wl = word limit (3 or 4 hex digits) ws = word size (4 or 8 bits) n = VOL or VOH (1=VOL and 0=VOH) A VOL device is one in which the programmed state is low, and a VOH device is one in which the programmed state is high.
X	View Error Codes	h1...h16	Outputs up to 16 error codes stored in memory. This command also clears the error codes from memory. See the Error Messages section for explanations of the meanings of the error codes.
G	View Software Configuration	hhhh	Outputs the configuration number of the 201's software.

Editing RAM

Use the following commands to move a block of RAM, rearrange the data to allow for use of 16-bit-wide data, or clear RAM.

Command Format	Command Name	Data Output to Host	Description
hhhh?	Split RAM Data	none	Splits a block of 16-bit-wide word data into two 8-bit-wide blocks around the given midpoint (hhhh), for programming into 8-bit-wide devices. The split places even-addressed bytes into consecutive addresses starting at address 0, and places odd-addressed bytes into consecutive addresses starting with the specified midpoint, thus splitting the data into two adjacent blocks occupying the same original block of RAM. The specified midpoint must be a power of two between 2H and the RAM midpoint (8000H).
hhhh>	Shuffle RAM Data	none	Reverses the split operation, converging two adjacent blocks of 8-bit-wide data into one block of 16-bit-wide data occupying the same original block of RAM. The data bytes below the specified midpoint (hhhh) are placed in even-numbered addresses, starting with address 0, and data bytes with addresses from the midpoint and higher are placed in odd-numbered addresses, starting with address 1.

Command Format	Command Name	Data Output to Host	Description
\	RAM to RAM Block Move	none	Moves the number of bytes specified by the Set Block Size command from the RAM location specified by the Set Beginning RAM Address command to the RAM location specified by the Set Beginning Device Address command.
^	Clear All RAM	none	Clears (fills) all of the programmer's data RAM with zeroes.

Data Translation Formats

Introduction

This subsection defines the data translation formats available for the 201 EPROM Programmer. The 201 EPROM Programmer is capable of interfacing with most RS232 serial equipment employing a data translation format described in this subsection.

Any of the five available data translation formats can be selected from the front panel for upload and download operations. In Computer Remote Control mode a 2-digit code assigned to each data translation format must be entered into the programmer through the serial port to send or receive data in that format. In addition to the data translation format code, there is a 1-digit instrument control code that can be entered in CRC mode. This instrument control code specifies control characters to be transmitted to, or received from, peripheral instruments.

Available Data Translation Formats

The following is a quick reference list of data translation formats available for the 201 EPROM Programmer and their corresponding codes. The formats are listed in alphabetical order. Each format is described in detail in the pages that follow.

Format	Code
Intel Intellec 8/MDS	83
Intel MCS-86 Hexadecimal Object	88
Motorola Exorciser	82
Motorola Exormax	87
Tektronix Hexadecimal	86

Data Verification

For data verification, the 201 calculates a sumcheck of all data sent to or from the programmer. At the end of a successful input operation, the programmer will display the sumcheck of all data transferred. It will also compare any received sumcheck fields with its own calculation. If the two agree, the programmer will display the sumcheck; a mismatch will produce an error message. Output data is always followed by a sumcheck field which may be printed on disk or tape for use in subsequent input operations.

Description of Format Codes

Each format is assigned a 2-digit data translation format code which you must enter when transferring data in CRC mode to tell the programmer which format to use. In addition to this code, a 1-digit instrument control code may be used to specify control characters for peripheral equipment. The codes must be formatted as follows: xyy, where "x" is the instrument control code and "yy" the data translation format code. If no codes are entered into the programmer, the current default values will be in effect.

The following list shows the instrument control codes, with the corresponding 201 action.

Control Code	Programmer Action
0	Sends data immediately and continuously until acknowledging a "reader off" code. It will then stop sending data until receiving a "reader on" code. If no control code is selected, control code 0 is the default value.
1	Sends "reader on" (ASCII DC1/Hex 11) when ready to receive data, and "reader off" (ASCII DC3/Hex 13) when all data is received. Also sends "punch on" (ASCII DC2/Hex 12) before sending data, and "punch off" (ASCII DC4/Hex 14) after sending data.
2	Sends data after acknowledging a "reader on" (ASCII DC1/Hex 11), and stops sending data after acknowledging a "reader off" (ASCII DC3/Hex 13).

Leader and Null Output

A leader is a string of characters that is attached to the beginning and end of a data file. It is used to separate different files from one another and allows extra room which may be necessary for loading and unloading the data medium to or from equipment. For the 201 EPROM Programmer, the leader is sent at the beginning and end of a data output operation. With one exception, this leader will always be comprised of a carriage return, a line feed, and 50 nulls in succession.

Null count is the number of null characters in the string of characters between each record or line within a file. What actually comprises a data record depends upon the format that is being used. Records and lines can basically be thought of as separations of data within a file.

Null count is a parameter which can be defined by the 201 user (with the "H" command) for use with printers with a slow carriage return response time. The number of nulls can be set to any value from zero to 254 decimal (FE hexadecimal). The string of characters actually sent between each and every record or line of the file includes a carriage return, a line feed, and the number of nulls defined by the null count, with one exception. When the user defines the null count to be equal to the value of "FF" hexadecimal (or 255 decimal), the leader is made up of a solitary carriage return (no line feed and no nulls). Also, the string separating the records of the file is a carriage return (no line feeds and no nulls).

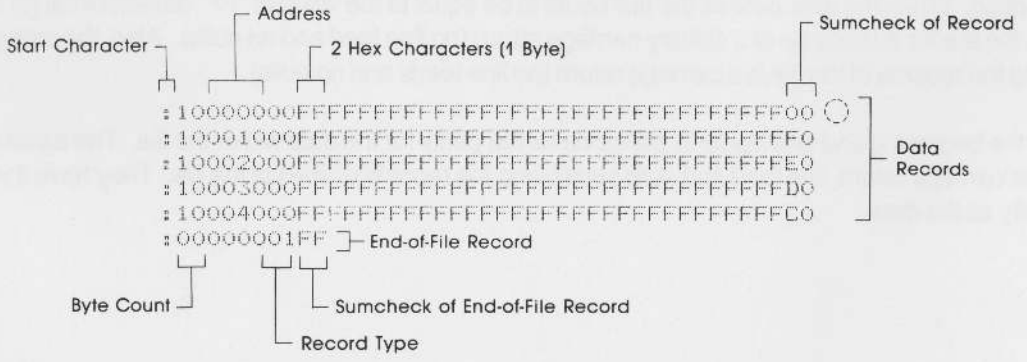
Parity for the beginning and end leader is the same as the parity for the data within the file. The same is true for the carriage return, line feed and nulls separating the records or lines of the file. They have the same parity as the data.

Intel Intellec 8/MDS Format, Code 83

Intel data records begin with a 9-character prefix and end with a 2-character suffix. The illustration represents a series of valid Intel data records.

Each record begins with a colon, which is followed by a 2-character byte count. The byte count must equal the number of data bytes in the record. The four digits following the byte count give the address of the first data byte. Each data byte is represented by two hexadecimal digits; the number of data bytes in each record must equal the byte count. Following the data bytes of each record is the sumcheck (the two's complement, in binary, of the sum of the preceding bytes, including the byte count, address and data bytes) expressed in hexadecimal.

The end-of-file record consists of the "colon" start character, the byte count (equal to "00"), the execution address, the record type (equal to "01") and the sumcheck of the record. The execution address is ignored on input and is always zero during output by Data I/O translator firmware.



LEGEND

◌ Nonprinting Carriage Return, line feed, and nulls

Intel MCS-86 Hexadecimal Object, Code 88

The Intel 16-bit Hexadecimal Object file record format has a 9-character (4-field) prefix that defines the start of record, byte count, load address, and record type and a 2-character sumcheck suffix. The illustration shows some sample records of this format.

The four record types are:

00 = data record
01 = end record (signals end of file)
02 = extended address record (added to the offset to determine the absolute destination address)
03 = start record (ignored during input and not sent during output by Data I/O translator firmware)

Record type 00, a data record, begins with the colon start character. The colon is followed by the byte count (in hexadecimal notation), the address of the first data byte, and the record type (equal to "00"). These parameters are followed by the data bytes, and then the sumcheck.

Record type 01, the end-of-file record, also begins with the colon start character. The colon is followed by the byte count (equal to "00"), the address (equal to "0000"), the record type (equal to "01") and the sumcheck (equal to "FF").

Record type 02, the extended address record, defines bits 4 to 19 of the segment base address. It can appear randomly anywhere within the object file and in any order; i.e., it can be defined such that the data bytes at high addresses are sent before the bytes at lower addresses. Record type 02 begins with the colon start character. The colon is followed by the byte count (equal to "02"); the address, which is ignored (equal to "0000"); the record type (equal to "02"); the 16-bit offset address; and then the sumcheck.

In all record types, the sumcheck is the two's complement (in binary) of the sum of the preceding bytes in the record, including the byte count, address and data bytes.

The following example illustrates how the extended address is used to determine a destination address.

Problem: Find the address for the first data byte for the following file.

```
:02 0000 02 1230 BA
:10 0045 00 55AA FF .....BC
```

Solution:

Step 1: Find the record address for the byte. The first data byte is 55. Its record address is 0045 from above.

Step 2: Find the offset address. The offset address is 1230 from above.

Step 3: Shift the offset address one place left, then add it to the record address, like this:

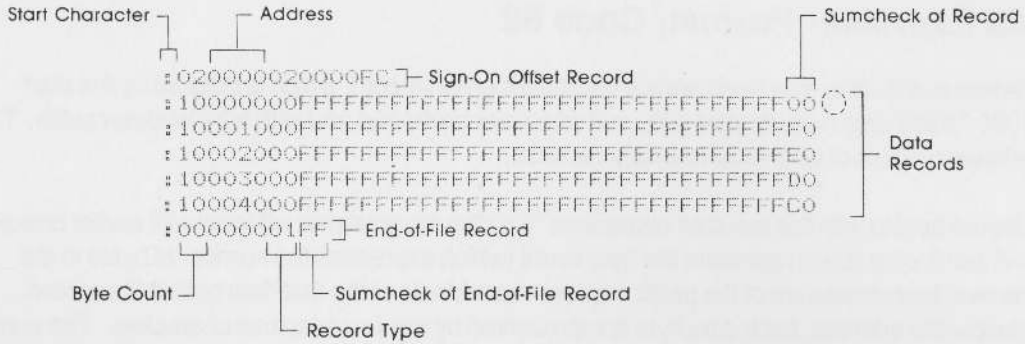
offset address	1230	(upper 16 bits)
+ record address	0045	(lower 16 bits)
	12345	(20-bit address)

The address for the first data byte is therefore 12345.

NOTE

Always specify the address offset when using this format, even when the offset is zero.

During output translation, the firmware will force the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.



LEGEND

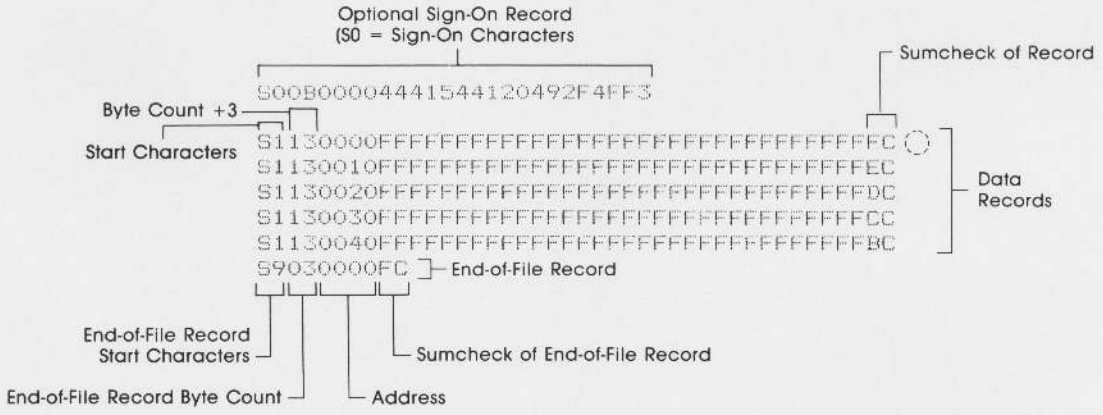
◌ Nonprinting Carriage Return, line feed, and nulls

Motorola Exorciser Format, Code 82

Motorola Exorciser data files may begin with an optional sign-on record, which is initiated by the start characters "S0." Valid data records start with an 8-character prefix and end with a 2-character suffix. The illustration shows a series of valid Motorola data records.

Each data record begins with the two start characters "S1"; the programmer will ignore all earlier characters. The third and fourth characters represent the byte count (which expresses the number of bytes in the record). The next four characters of the prefix express the address of the first data byte in the record. Data bytes follow the address; each data byte is represented by two hexadecimal characters. The number of data bytes occurring must be three less than the byte count. The suffix is a 2-character sumcheck, which equals the one's complement of the binary summation of the byte count, address and data bytes.

The end-of-file record consists of the start characters "S9," the byte count (equal to "03"), the address (in hexadecimal) and a sumcheck.



LEGEND

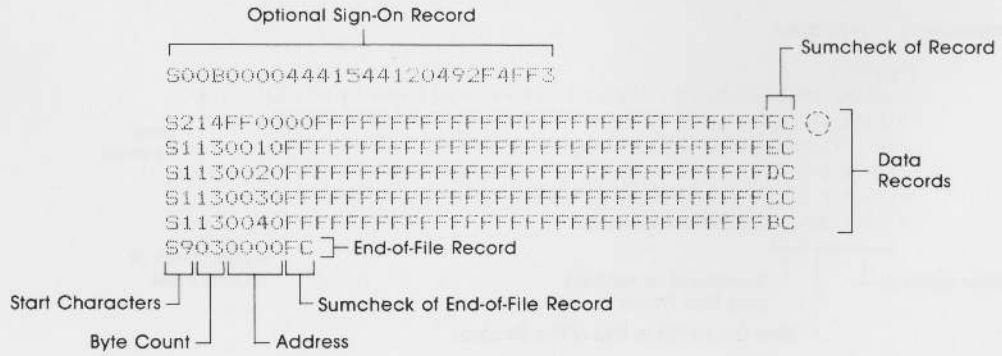
◌ Nonprinting Carriage Return, line feed, and nulls

Motorola Exormax Format, Code 87

Motorola data files may begin with an optional sign-on record, initiated by the start characters "S0." Data records start with an 8- or 10-character prefix and end with a 2-character suffix. The illustration shows a series of Motorola Exormax data records.

Each data record begins with the start characters "S1" or "S2"; a record begins with "S1" if the following address field has four characters, with "S2" if it has six characters. The third and fourth characters represent the byte count (which expresses the number of bytes in the record). The next four characters (six characters for "S2" records) express the address of the first data byte in the record. Data bytes follow the address; each data byte is represented by two hexadecimal characters. The number of data bytes occurring must be three or four less than the byte count, depending on the record type. The suffix is a 2-character sumcheck, the one's complement (in binary) of the preceding bytes in the record, including byte count, address and data bytes.

The end-of-file record begins with the start characters "S8" or "S9." The start characters must be "S9" if the previous data record started with an "S1"; otherwise, either "S8" or "S9" may be used. Following the start characters are the byte count (equal to "03"), the address (equal to "0000") and a sumcheck.



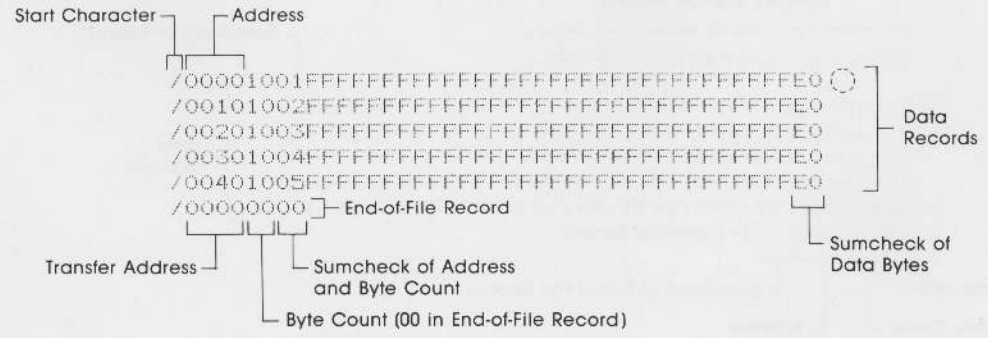
Tektronix Hexadecimal Format, Code 86

The illustration shows a valid Tektronix data file. Each record begins with the start character (a slash). Following the start character, the next four characters of the prefix express the address of the first data byte. The address is followed by a byte count, which represents the number of data bytes in the record, and by a sumcheck of the address and byte count. Data bytes follow, represented by pairs of hexadecimal characters. Succeeding the data bytes is their sumcheck, an 8-bit sum, modulo 256, of the 4-bit hexadecimal values of the digits making up the data bytes. All records are followed by a carriage return.

Data is output from the programmer starting at the first RAM address specified and continuing until the number of bytes in the specified block have been transmitted. The programmer divides output data into records prefaced by a start character and an address field for the first byte in the record.

The end-of-file record consists of a start character (slash), followed by the transfer address, the byte count (equal to "00"), and the sumcheck of the transfer address and byte count.

An optional abort record contains two start characters (slashes), followed by an arbitrary string of ASCII characters.



LEGEND

◌ Nonprinting Carriage Return, line feed, and nulls

5. Error Messages

The following is a list of error codes and their corresponding messages. The circumstances which cause an error message to be displayed are described in the Description column and corrective action to take upon receiving the error message is explained in the right-hand column. The error codes are listed in numerical order.

Code	Name	Description	Corrective Action
17	BANK ERROR	The bank selector circuit is unable to switch banks.	Contact your local Data I/O Service Center.
20	NONBLANK DEVICE	Device failed blank test.	Erase the device.
21	ILLEGAL BIT	Unable to program device due to already programmed bit of incorrect polarity.	Erase the device if possible or discard it.
22	PROGRAM FAIL	The device failed to program properly.	Replace device if faulty.
23	VERIFY ERROR	The programmed PROM data failed to verify against the master data in RAM.	Reprogram the device, if possible, or try another device.
27	BLOCK LIMIT ERROR	PROM or RAM size is insufficient to perform the current operation using the begin RAM address, block size, or set size specified.	Make sure the begin RAM address, block size, and set size are set correctly. You may need to have your RAM serviced.

Error Messages

Code	Name	Description	Corrective Action
31	OVERCURRENT	The device to be programmed is drawing excessive current.	Indicates that the device is faulty. Replace the device.
32	VCC ERROR	When 5 V Vcc is applied to pin 26/28, the voltage drops to an unacceptable level.	Replace device if faulty.
33	VPP ERROR	The programming power supplies cannot be set at the proper levels.	Replace device if faulty.
34	BAD DEVICE CODE	An incorrect family and pinout code was entered.	Consult the Device List for the correct family and pinout codes and enter the correct code.
35	NOT TRISTATE™	Device failed to TRISTATE™ all data pins.	Indicates that the device is faulty. Replace the device.
36	DEVICE NOT ENABLED	All the device's data pins are not driving when the chip is enabled.	Indicates that the device is faulty. Replace the device.
37	BAD INSERTION	The device was inserted in the socket incorrectly.	Align the bottom-most pin of the device with the bottom of the socket. Pin 1 should be at the top of the device.

Code	Name	Description	Corrective Action
41	FRAME ERROR	The serial interface detected a start bit but the stop bit was in the wrong position.	Check the current baud rate and stop bit settings.
42	OVERRUN ERROR	The serial interface received characters when the programmer was unable to accept them.	Check the serial port connections. Make sure handshake lines are properly connected and try again.
46	I/O TIMEOUT	No characters, or only nulls and rubouts, were received upon serial input for 25 seconds after pressing the ENTER key; or, no characters could be transmitted for a period of 25 seconds due to the state of the handshake lines.	Check all connections and attempt transmission again.
48	BUFFER FULL	The serial port input buffer received more characters than the programmer was able to process.	Make sure the handshake lines are hooked up and operative.
52	I/O VERIFY ERROR	The data from the serial port does not match the data in RAM.	Reload data to RAM. If the problem persists, service the programmer or contact your local Data I/O Service Center.

Error Messages

Code	Name	Description	Corrective Action
64	DRAM ERROR	The self-test shows a dynamic RAM read/write verification error.	Contact your local Data I/O Service Center.
67	I/O FORMAT ERROR	Data sent to the programmer are non-hexadecimal characters. Input data is in the incorrect format.	Specify the correct data translation format for the data being transferred, or correct the data in the file.
71	PIN 1 ERROR	The self-test indicates that the socket pin 1 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
72	ADDRESS SHORT	The device contains a short on its address line.	Indicates that the device is faulty. Replace the device.
73	DATA SHORT	The device contains a short on its data line.	Indicates that the device is faulty. Replace the device.
74	DATA BUS FAIL	Indicates that the part data bus may be damaged or the contact to ZIF socket is faulty.	Replace the faulty device or contact your local Data I/O Service Center.
75	PIN 22 ERROR	The self-test indicates that the socket pin 22 driver is at an inaccurate level.	Contact your local Data I/O Service Center.

Code	Name	Description	Corrective Action
76	PIN 23 ERROR	The self-test indicates that the socket pin 23 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
77	PIN 26 ERROR	The self-test indicates that the socket pin 26 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
78	PIN 24 ERROR	The self-test indicates that the socket pin 24 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
79	PIN 28 ERROR	The self-test indicates that the socket pin 28 driver is at an inaccurate level.	Contact your local Data I/O Service Center.
81	PARITY ERROR	The incoming data has incorrect parity.	Check the parity setting and re-attempt transmission.
82	I/O FORMAT ERROR	The sumcheck field received by the programmer does not agree with its own calculated sumcheck.	Reload data to RAM. If the problem persists, service the programmer or contact your local Data I/O Service Center.
90	INVALID FORMAT	Non-existent or unsupported I/O format is selected in Computer Remote Control.	Check the list of supported I/O formats and enter a valid format code.

Error Messages

Code	Name	Description	Corrective Action
91	I/O FORMAT ERROR	The programmer received an invalid address field.	Check all connections, check the data format and data source, and then try again.
92	5.6 V FAIL	The 5.6 V power supply circuit failed.	Contact your local Data I/O Service Center.
93	ID VOLT FAIL	The electronic ID power supply circuit failed.	Contact your local Data I/O Service Center.
94	VPP FAIL	The programming pulse power supply circuit failed.	Contact your local Data I/O Service Center.
95	DAC FAIL	The digital analog converter circuit failed.	Contact your local Data I/O Service Center.
98	VCC FAIL	The Vcc supply circuits are not working properly.	Contact your local Data I/O Service Center.
A1	NO ID FOUND	The installed device has no electronic ID.	Check the device type. Consult the Device List for the correct family and pinout codes.
A2	INVALID ID	The device's electronic ID is incompatible with the family and pinout codes selected.	Consult the Device List for the correct family and pinout codes and make sure the correct codes are entered. Programming with incorrect codes could damage a part.

6. Index

Index

UNITED STATES OFFICES AND REPRESENTATIVES

CORPORATE
Data I/O FutureNet
10525 Willows Road N.E.
Redmond, WA 98073
(206) 881-6444
1-800-426-1045
FAX (206) 882-1043

FutureNet
9310 Topanga Canyon Blvd.
Chatsworth, CA 91311-5728
(818) 700-0691
Telex 910-494-2681
FAX (818) 700-1853

EASTERN REGION
Data I/O FutureNet
Birch Pond Business Cntr
22 Cotton Road
Nashua, NH 03063
(603) 889-8511
FAX (603) 880-0697

NORTHWEST REGION
Data I/O FutureNet
1701 Fox Drive
San Jose, CA 95131
(408) 437-9600
FAX (408) 437-1218

SOUTHWEST REGION
Data I/O FutureNet
3505 Cadillac Avenue
Suite L-1
Costa Mesa, CA 92626
(714) 662-1182
FAX (714) 662-0930

ALABAMA
(205) 881-9298

ALASKA
(503) 645-9000

ARIZONA
(602) 263-6022
1-800-528-4512

ARKANSAS
(214) 235-0044

CALIFORNIA, NORTHERN
Data I/O
(408) 437-9600

CALIFORNIA, SOUTHERN
(714) 662-1182

COLORADO
(303) 477-5161

CONNECTICUT
(603) 889-8511

DELAWARE
(603) 889-8511

FLORIDA
(305) 394-7400
(305) 660-4555
(305) 366-8706

FLORIDA, NORTHERN
(205) 881-9298

GEORGIA
(404) 751-1604

HAWAII
(503) 645-9000

IDAHO
(801) 487-9402

ILLINOIS, NORTHERN
(312) 520-0710

ILLINOIS, SOUTHERN
(314) 839-0800

INDIANA
(317) 244-7867

IOWA
(319) 373-0200

KANSAS
(314) 839-0800

KENTUCKY, EASTERN
(412) 322-0900

KENTUCKY, WESTERN
(317) 244-7867

LOUISIANA
(214) 235-0044

MAINE
(603) 889-8511

MARYLAND
(703) 759-7225

MASSACHUSETTS
(603) 889-8511

MICHIGAN
(313) 421-7500
(616) 323-2416

MINNESOTA
(612) 835-2414

MISSISSIPPI
(205) 881-9298

MISSOURI
(314) 839-0800

U.S. Representatives

MONTANA
(801) 487-9402

NEBRASKA
(314) 839-0800

NEVADA
(602) 263-6022
1-800-528-4512

NEW HAMPSHIRE
(603) 889-8511

NEW JERSEY, NORTH
1-800-858-5803

NEW JERSEY, SOUTH
(703) 759-7225

NEW MEXICO
(505) 842-6633
1-800-528-4512

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) 235-0044

OREGON
(503) 645-9000

PENNSYLVANIA, EASTERN
(703) 759-7225

PENNSYLVANIA, WESTERN
(412) 322-9000

PUERTO RICO
(603) 889-8511

RHODE ISLAND
(603) 889-8511

SOUTH CAROLINA
(919) 852-6000

SOUTH DAKOTA
(612) 835-2414

TENNESSEE
(404) 751-1604
(205) 881-9298

TEXAS
(214) 235-0044
(512) 346-8901

TEXAS, EL PASO CO.
(505) 842-6633
1-800-528-4512

UTAH
(801) 487-9402

VERMONT
(603) 889-8511

VIRGINIA
(703) 771-1444

WASHINGTON, TRI-CITIES & VANCOUVER
(503) 645-9000

WASHINGTON, WESTERN & SPOKANE
(206) 881-8857

WASHINGTON D.C.
(703) 759-7225

WEST VIRGINIA
(412) 322-9000

WISCONSIN
(414) 784-7736

WYOMING
(303) 477-5161

U.S. SERVICE CENTERS

Redmond, WA
(206) 881-6444
1-800-247-5700

San Jose, CA
(408) 437-9609

Costa Mesa, CA
(714) 662-2498

Nashua, NH
(603) 889-8513

Richardson, TX
(214) 235-0044

INTERNATIONAL OFFICES AND REPRESENTATIVES

DATA I/O EUROPE
World Trade Center
Strawinskylaan 633
1077 XX Amsterdam
The Netherlands
31 (0) 20-622866
Fax 020-624427
Telex 16616 DATIO NL

DATA I/O JAPAN COMPANY, LTD.
Sumitomoseimei Higashishinbashi Bldg. 8F
2-1-7, Higashi-Shinbashi Minato-Ku
Tokyo 105
Japan
(03) 432-6991
Fax 0118134326094
Telex 2522685 DATAIO J

DATA I/O CANADA
6725 Airport Road
Suite 302
Mississauga, Ontario L4V 1V2
Canada
(416) 678-0761
Fax (416) 678-7306
Telex 06968133

ARGENTINA
701-4462

AUSTRALIA
Brisbane
07 275-1766

Melbourne
03 795-9011

Sydney
02 648-1711

AUSTRIA
0222 827474

BELGIUM
02 2192451

BRAZIL
Sao Paulo
011 247-5588

Rio de Janeiro
21 284-1248

CHINA
U.S.A.
(206) 583-0000

Beijing
507766

Hong Kong
3 770-2021

DENMARK
02 456645

FINLAND
90 5284312

FRANCE
13 9568131

GERMANY
089 858020

GREECE
01 7249511

HONG KONG
5-546391

INDIA
Bombay
022 4938585

Bangalore
0812 564389

ISRAEL
03 494891

ITALY
Rome
06 5041367

Milan
02 6181893

KOREA
2 555-5222

MALAYSIA
65 2729412

MEXICO
595-7594

NETHERLANDS
040 582911

NEW ZEALAND
09 444-2645

NORWAY
02 789460

PORTUGAL
01 4103420

International Representatives

SINGAPORE
65 2729412

SOUTH AFRICA
12 46-9221

SPAIN
Madrid
01 242-5204

Barcelona
03 239-2655

SWEDEN
08 380370

SWITZERLAND
Thalwil
01 7231410

Geneva
022 360830

TAIWAN
2 5010055

THAILAND
02 3925313

TURKEY
143 62 12-13

UNITED KINGDOM
0494 41661

The information contained in this list is subject to change without notice.

Product Warranty

Data I/O warrants that product(s) sold pursuant to any resultant agreement shall be free from defects in material or workmanship at the time of delivery. Such warranty shall extend for one (1) year for hardware and ninety (90) days for software, unless specified otherwise, beginning from the date of shipment. Buyer must provide notice to Data I/O within this prescribed warranty period of any defect; if the defect is not the result of improper usage, service, maintenance or installation and equipment has not been otherwise damaged or modified after delivery, Data I/O shall either replace or repair the defective part or parts of equipment or replace the equipment or refund the purchase price at Data I/O's option after return of such equipment by buyer to Data I/O. Shipment to Data I/O's facility shall be borne on account of buyer, including all duties, fees, brokerage and special handling charges. Shipment from Data I/O's facility shall be borne by Data I/O.

(a) Consequential Damages — Data I/O shall not be liable for any incidental or consequential damages incurred as a result of any defect in any equipment sold hereunder and Data I/O's liability is specifically limited to its obligation described herein to repair or replace a defective part or parts covered by this warranty.

(b) Exclusive Warranty — The warranty set forth herein is the only warranty, oral or written made by Data I/O and is in lieu of and replaces all other warranties, expressed or implied, including the WARRANTY OF MERCHANTABILITY AND THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 41 REDMOND, WA 98052

POSTAGE WILL BE PAID BY ADDRESSEE

DATA I/O

Warranty/Registration
Technical Support Manager
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA 98073-9987

090-0242-002

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 41 REDMOND, WA 98052

POSTAGE WILL BE PAID BY ADDRESSEE

DATA I/O

Reader Comments
Technical Publications Supervisor
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA 98073-9987

090-0242-002

Warranty Registration / License Acknowledgment

Please sign and return this card to Data I/O within five days

Did the packaging of this equipment exhibit any outward signs of physical damage? ☐ YES ☐ NO

Did this equipment arrive intact, without loose parts or cable damage? ☐ YES ☐ NO

Did the equipment operate on power-up? ☐ YES ☐ NO

Did you attain adequate system performance? ☐ YES ☐ NO

Were any electrical adjustments required? ☐ YES ☐ NO

If you required assistance, was a local Data I/O representative contacted? ☐ YES ☐ NO

Comments: _____

Name _____ Title _____

Department _____ M/S _____

Company _____ Phone (_____) _____

Address _____

City _____ State _____ Zip _____

The undersigned hereby agrees to the terms and conditions of the software license agreement for the product listed below:

Model or Description _____ Serial or Part No. _____

Date Received _____ Signature _____

Reader Comments

The manual's completeness, accuracy, organization, usability, and reliability: _____

_____

Did you find errors in this manual? _____

How can this manual be improved? _____

_____

_____

Additional comments: _____

_____

Name _____ Title _____

Department _____ M/S _____

Company _____

Address _____

City _____ State _____ Zip _____

Phone ( ) _____ Manual or Part No. _____