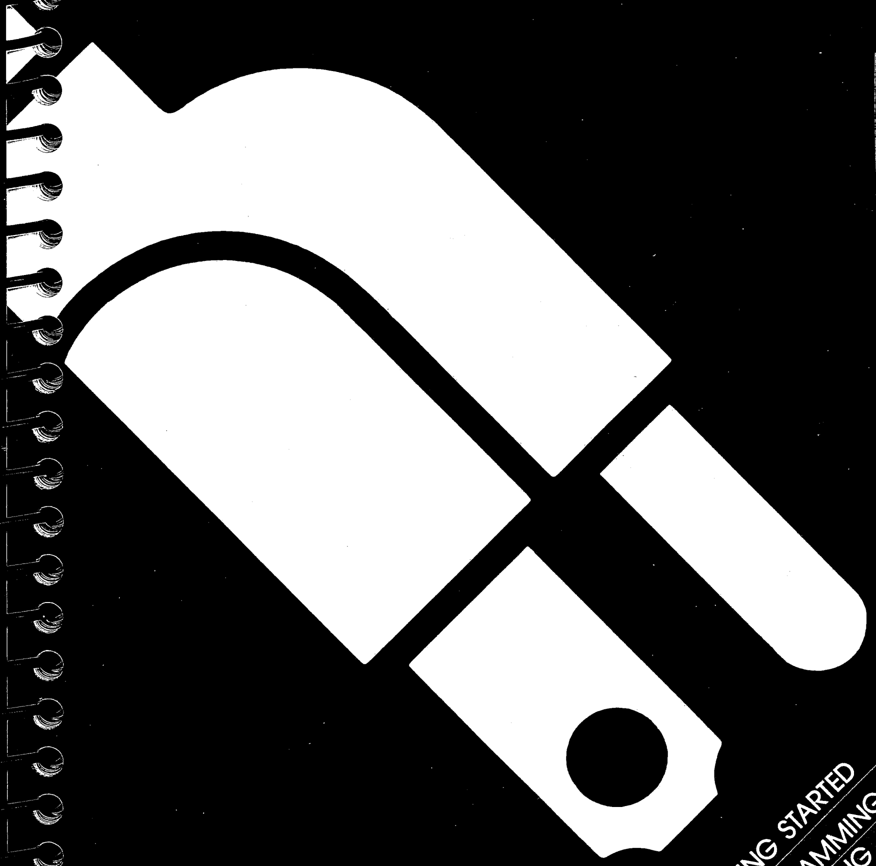
OPERATOR'S GUIDE
21A
EPROM PROGRAMMER

DATA I/O
10525 Willows Rd. N.E., P.O. Box 97046, Redmond, WA 98073-9746

This operator's guide introduces and summarizes the major features and operations of the programmer. This guide supplements the manual shipped with the 21A, which covers in full detail the more complex and less used functions not discussed in this guide.



GETTING STARTED
PROGRAMMING
EDITING
COMMUNICATIONS
INDEX

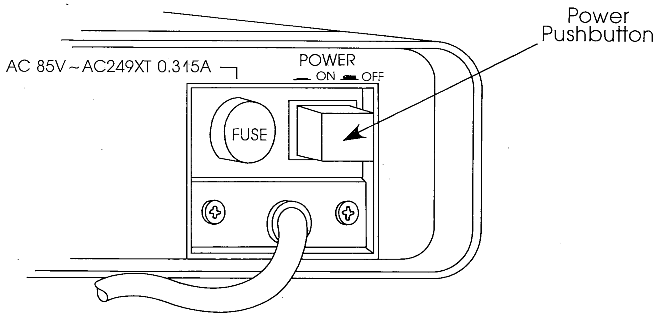
GETTING STARTED

The 21A is a self-contained PROM programmer that features extensive programming, editing, and communications capabilities to program a variety of EPROMs and EEPROMs. The 21A's display, keyboard, device socket, and setup procedures are discussed in this section. Subsequent sections in the operator's guide present programming, editing, and communications operations and procedures.

GETTING STARTED
PROGRAMMING
EDITING
COMMUNICATIONS
INDEX

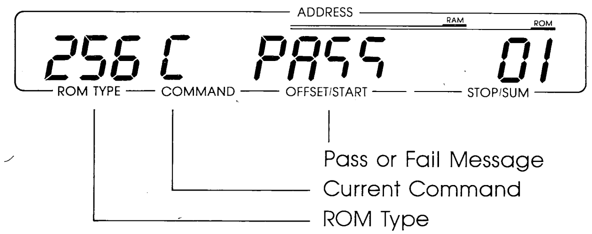
TURNING THE PROGRAMMER ON

1. Make sure that the AC power pushbutton located at the left rear of the programmer is in the *off* position (see illustration).



AC Power Pushbutton, Location and Settings

2. Connect the power cord to a three-prong, grounded AC outlet (85 to 249 volts AC; 48 to 440 Hertz).
3. Depress the AC power pushbutton at the left rear of the programmer. This is the *on* position. The display will light up momentarily and then show some of the default operations settings.



Programmer Display After Turning the Power On

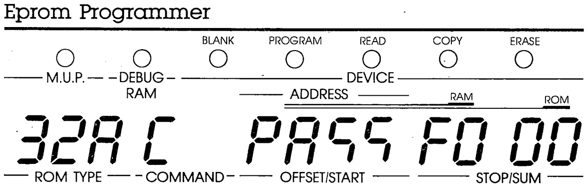
THE DISPLAY

The 21A display is divided into sections that show information applicable to specific operations (see illustration on next page). On the top row of the display are LED indicators with the following functions:

M.U.P. LED: When lit, this indicates that no device is in the device socket or that a device is inserted incorrectly. The lamp is off when a device is correctly inserted in the socket.

DEBUG RAM: This feature is not implemented.

BLANK, PROGRAM, READ, COPY, ERASE: These indicators are lit when the device command with the same name is selected. For some operations, such as B.P.R. and P.R., more than one indicator will be lit.



21A Display

The alphanumeric display below the LEDs indicates the currently selected parameters and commands, addresses, hexadecimal data, and results of operations. This information is displayed on different parts of the display as marked. The divisions are described below. Some divisions overlap.

ROM TYPE: Displays a three digit code representing the type of PROM being used. For applicable codes, see "Programming" in this guide.

COMMAND: Displays the command that is currently selected. In general, the command is represented by a letter indicating the type of command, followed by a hexadecimal digit indicating the specific command.

ADDRESS and OFFSET/START: This four character display shows RAM addresses, data entries, *PASS* or *FAIL* according to test results, "*Er*" when operation errors occur, the starting device address, and the address offset depending on the operation being performed.

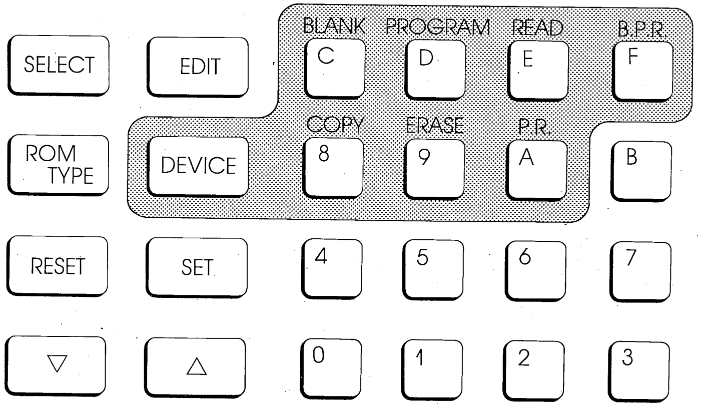
RAM: Displays the contents of a RAM location or an error code when an operation error occurs.

ROM: Displays data loaded from the PROM in the device socket.

STOP/SUM: Displays four-digit checksums and the device "stop" address.

THE KEYPAD

The 21A features a color- and shape-coded keypad with legends as shown in the illustration below. The keypad is logically divided into two sections: command keys, and data keys. Command keys are brown and rectangular and select specific commands or groups of commands to execute. The smaller, square, green, data keys are used to enter addresses, data, and operation codes. The data keys are also used in conjunction with the command keys to select specific commands from within a group.



21A Keypad

The command keys are described on the following page.

SELECT key: Selects miscellaneous operations and functions such as setting communication protocols, data translation formats, and data transfers.

EDIT key: Selects one of seven RAM editing functions. The *EDIT* key is used in conjunction with data keys to select the editing function desired.

ROM TYPE key: Used to enter the PROM code, which defines the type of PROM being used. See "Programming" for PROM codes.

DEVICE key: Selects one of seven device operations. The *DEVICE* key is used in conjunction with data keys to select the device function desired.

RESET key: Terminates operations and resets the programmer after errors. Use this key whenever you need to restart or cancel an operation.

SET key: Confirms selection of an operation, function, or address. The *SET* key also starts operations.

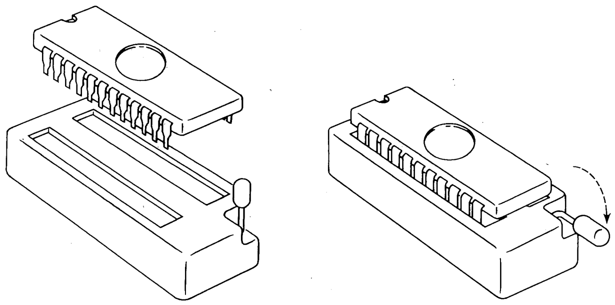
△ key: Used to increment data addresses during RAM editing and display operations. Also confirms entry of an address.

▽ key: Used to decrement data addresses during RAM editing and display operations.

INSERTING A DEVICE IN THE SOCKET

A PROM can be inserted in the device socket as described below. Please refer to the illustration on this page.

1. Lift the lever at the lower right corner of the device socket so that it is vertical.
2. Orient the PROM with the notch toward the back of the programmer and with the pins lined up over the pin holes in the socket. The device pins nearest you should be above the socket pin holes nearest you.
3. Lower the PROM into the socket and press the lever down and toward you so that it is horizontal. If you inserted the PROM correctly, the MUP LED should not be lit.



Inserting a PROM Into the Device Socket

GETTING STARTED
PROGRAMMING
EDITING
COMMUNICATIONS
INDEX

PROGRAMMING

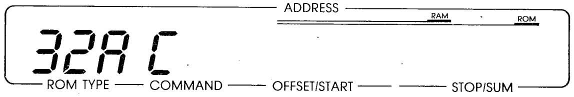
The following sequence of commands loads data from a programmed PROM into programmer RAM, checks a new device for programmed bits, programs the new device with the data contained in RAM, and verifies that the data in the newly programmed PROM matches the RAM data. Individual functions and components of the displays are explained following the command sequence.

The displays shown assume that a 2732A EPROM is being used. Checksums shown are examples only; your checksum will depend on the data you are using. The final state of the display after a key sequence is executed is shown following that sequence. For some operations the programmer display may show intermediate steps. In particular, during all device-related commands except *Program*, four underline characters are shown in the address portion of the display during execution of the command.

1. Insert the programmed PROM in the device socket.

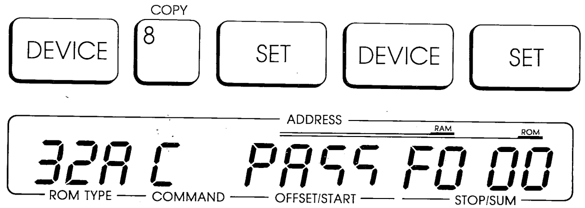
2. Select the correct PROM type (PROM codes are given at the end of this section).

ROM TYPE   prom-code   SET



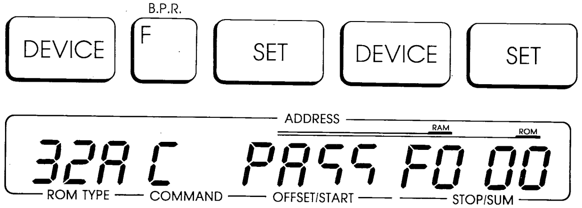
PROGRAMMING
EDITING
COMMUNICATIONS
INDEX

3. Copy the program from the device into programmer RAM:

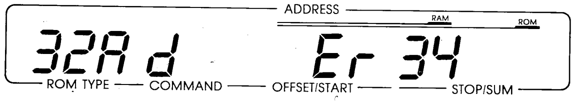


4. Remove the PROM from the device socket and insert a new, blank device into the socket.

5. Initiate the check for programmed bits, the programming of the device, and the check of programmed PROM data versus RAM data:



If any errors were encountered during the B.P.R. (Blank, Program, Read) function, operation is terminated at the point of the error, an error code is displayed, and the LED corresponding to the function being performed is lit. A list of error codes is given preceding the index in this guide. The following display shows a sample error code:

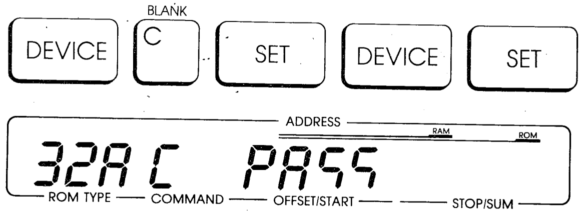


The sequence of operations shown above is general and applies to many programmer applications. The sequence uses a subset of the programming commands available. All programming commands, including *BLANK*, *ERASE*, *PROGRAM*, and *READ*, are described in detail below. The commands are presented in alphabetical order.

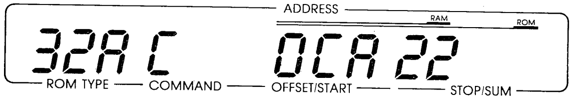
BLANK CHECK

A blank check is a check that determines if a device contains any programmed bits. The blank check passes if no bits are programmed (i.e. all locations contain the data FF hex). The blank check fails if any bit in the device has been programmed. In general, you should perform a blank check on any device that is to be programmed. Such preliminary checks verify that a device does not already contain programmed bits that could alter the operation of the PROM.

The key sequence used to initiate the program command and the display resulting from a successful blank check are:



The word *PASS* in the address section of the display indicates that the device is blank. If the device is not blank, the non-blank address and its contents are displayed:

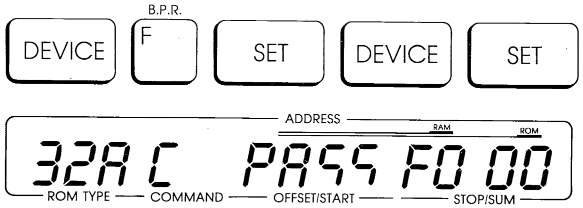


You can continue the blank check after a non-blank location has been found by pressing the △ key.

The blank check is combined with the program command in the *B.P.R* command described in this section.

B.P.R., BLANK PROGRAM READ

The *B.P.R* command is a combination of the blank check, program, and read commands, all of which can be executed independently. The key sequence used to initiate the *B.P.R.* command and the resulting display are:

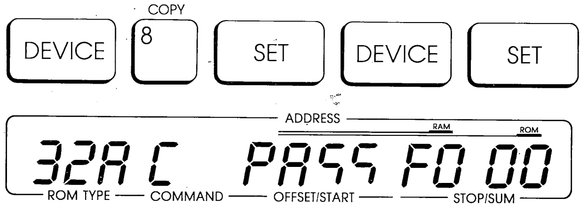


The display shows a checksum and *PASS* or *FAIL*, indicating successful or unsuccessful completion of the command.

COPY DEVICE DATA TO RAM

The copy operation copies data from a PROM in the device socket to programmer RAM. The device is checked for defects before the copy is performed and an error code is displayed if the device is defective.

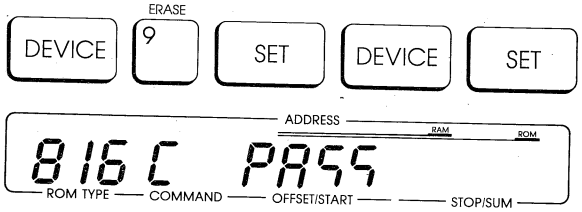
The key sequence used to initiate the copy and the resulting display are:



The display shows a checksum and *PASS* or *FAIL*, indicating successful or unsuccessful completion of the copy.

ERASE AND CHECK

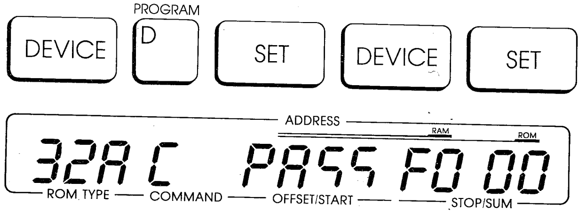
The erase command erases all data in an electrically erasable PROM (EEPROM) and then confirms the erasure by executing a blank check. The key sequence used to initiate the erase command and the resulting display (for a successful erasure) are:



If the erasure was not complete, the display indicates the non-blank address and contents as with the blank check.

PROGRAM

The program command programs the PROM in the device socket with the data contained in programmer RAM. The key sequence used to initiate the program command and the resulting display are:

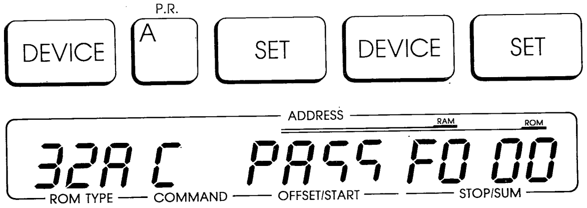


The display shows a checksum and *PASS* or *FAIL*, indicating successful or unsuccessful completion of the command.

The program command is combined with a read check in the *P.R* command, and with the blank check and read check in the *B.P.R* command, both which are described in this section.

P.R., PROGRAM AND READ

The *P.R* command is a combination of the program and read commands, both of which can be executed independently. The key sequence used to initiate the *P.R* command and the resulting display are:

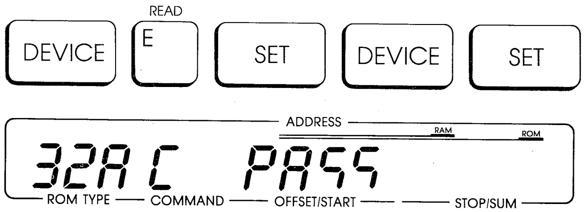


The display shows a checksum and *PASS* or *FAIL*, indicating successful or unsuccessful completion of the command.

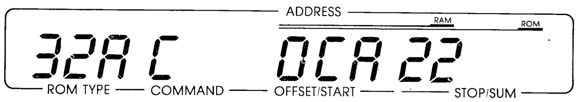
READ CHECK

A read check compares data programmed in a PROM with RAM data. The check passes if the PROM data matches the RAM data, and fails otherwise. Read checks are generally performed after programming a device to ensure correct programming of the part.

The key sequence used to initiate the program command and the display resulting from a successful read check are:



The word *PASS* in the address section of the display indicates that PROM data matches RAM data. If the data do not match, the address and contents of the first RAM location that differs from device contents is displayed:



You can continue the read check after a non-blank location has been found by pressing the △ key.

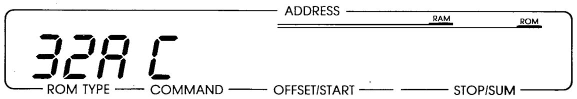
The read check is combined with the program command in both the *P.R* and *B.P.R* commands described in this section.

SET ROM TYPE

The *ROM Type* command is used to set the current ROM type. When the 21A is turned on the ROM type is 256. Executing device commands with an incorrect ROM type will generally cause errors.

The key sequence used to set the ROM type and the display after the operation are:

[ROM TYPE] prom-code [SET]



where *prom-code* is a two to three hexadecimal digit code indicating the PROM to be used. See the PROM code table. The display indicates the newly selected PROM code.

PROM Codes

EPROM Part #	ROM Type
AMD	
2716	16
2732	32
2732A	32A
2764	63
2764A	64A
27128	128
27128A	28A
27256	256
ELECTRONIC ARRAYS	
2716	16
EUROTECHNIQUE	
ET2716	16
ETC2716	16
ET2732	32
ET2764	63
FUJITSU	
2716	16
2732	32
2732A	32A
27C32A	32A
2764	F64
27C64	F64
27128	F28
27C128	F28
27256	F56
27C256	F56
HITACHI	
48016	E16
462716	16
482716	16
462532	532
482532	532
27C32	32
27C32A	32A
462732	32
482732	32
482732A	32A
27C64	63
482764	63
27C128	128
4827128	128
27C256	256
4827256	256
INTEL	
2815	815
2816	816
2716	16
2732	32
2732A	32A
2764	63
2764A	64A
27128	128
27128A	28A
27256	256
MITSUBISHI	
2716	16
2732	32
2764	63
27128	128
2732A	32A
27256	256
MOTOROLA	
MCM2716	16
MCM2532	632
MCM68764	664
MCM68766	664
NATIONAL	
2716	16
27C16	C16
27C16H	C16
2532	532
25C32	532
2732	32
27C32	32
27C32H	32
2764	63
2764H	63
NEC	
2716	16
2732	32
2732A	32A
2764	63
27C64D	63
27128	128
27256	256
OKI	
2716	16
2532	532
2732	32
2732A	32A
2764	63
27128	128
27256	256
MOSTEK	
2716	16
RICOH	
RD5H32	32A
RD687C64	D64
27C64	F64
ROCKWELL	
R87C32	32A
R87C64	63
SEEQ	
5133	63
5133H	63
5143	128
5143H	128
27C256	256
SGS-ATES	
2716	16
2732	32
2732A	32A
2764	63
2532	532
27256	256
SIGNETICS	
27C64	63
SYNERTEK	
2716	16
TEXAS INSTRUMENTS	
2716	16
2516	16
2532	532
2732	32
2732A	32A
2764	63
2564	564
27128	128
25L32	532
TOSHIBA	
323	16
2732	32
2732A	32A
2732D	32
2764	63
27128	128
27256	256
57256	F56
XICOR	
2816A	816

EDITING

The 21A features simple byte-by-byte editing of RAM locations and seven more sophisticated RAM data editing functions. Byte editing is discussed first, followed by a description of the seven editing functions.

BYTE EDITING

Byte editing commands must be started from a clear display, with nothing shown in the address, RAM, and ROM areas of the display. To clear a non-blank display, press the reset command key.

Display the contents of a location in RAM by entering the address and pressing the set key:

address SET

Use the △ and ▽ keys to move forward or backward from a displayed location and display others.

To change the contents of a location in RAM, enter the address to be edited, followed by the new data to be contained at that location as shown:

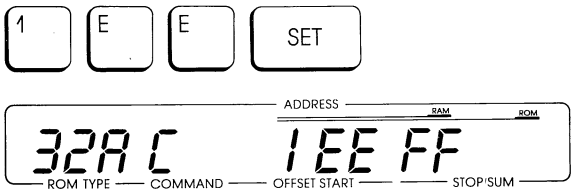
address SET data SET

where *address* is a one- to four-bit address, and *data* is the new data to be put in that location.

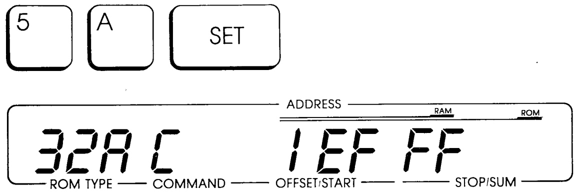
EDITING
COMMUNICATIONS
INDEX

When the *SET* command key is pressed after the address is entered, the current contents of that location are displayed. You can then enter the new contents for that location or use the △ or ▽ keys to move to the next or previous memory location without changing the contents of the current one. For example, the following sequence of commands displays and changes the contents of several RAM locations:

1. Display the contents of location 1EE hex:

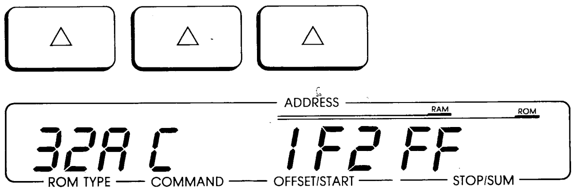


2. Change the contents to 5A hex:

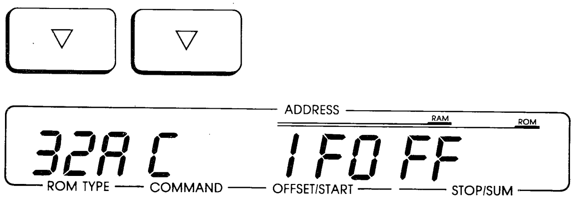


The display now shows the contents of the next RAM location.

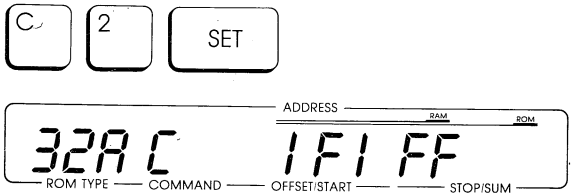
3. Move forward three addresses to 1F2:



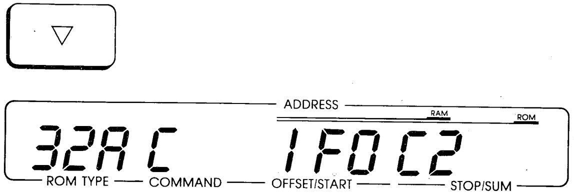
4. Move backward two addresses to 1F0:



5. Change the contents of 1F0 to C2:



6. Look at location 1F0 to confirm that the data was changed to C2:



EDITING FUNCTIONS

All seven editing functions are started by pressing the following keys:

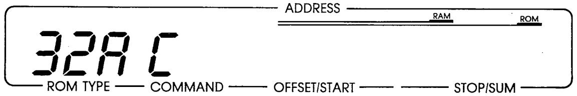
EDIT edit-function

where *edit-function* is the number of the particular function to be executed. Edit functions are explained below:

COMPLEMENT

All or part of RAM can be complemented with the complement operation. To complement all RAM data, press:

EDIT 0 SET



To complement a section of RAM data, press:

EDIT 0 fa △ la SET

where *fa* and *la* are the first and last addresses, respectively, to be complemented. Everything between and including *fa* and *la* is complemented.

For example:

EDIT 0 1 5 △ A
A SET

complements the data in RAM locations 15 hex through AA hex.

INSERT

Data can be inserted into one or more locations with the insert function. To insert data in one RAM location, press:

EDIT 1 fa △ data SET

where *fa* is the address at which the data is inserted. To insert data in more than one location, press:

EDIT 1 fa △ la △
data SET

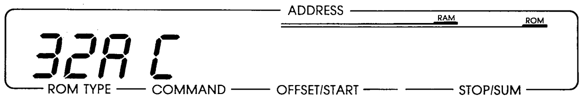
where *fa* and *la* are the first and last locations, respectively, to receive the specified data. In all insert operations, data residing initially in the locations to which data is to be inserted are moved to the next highest address following the last address to receive data.

EDITING
COMMUNICATIONS
INDEX

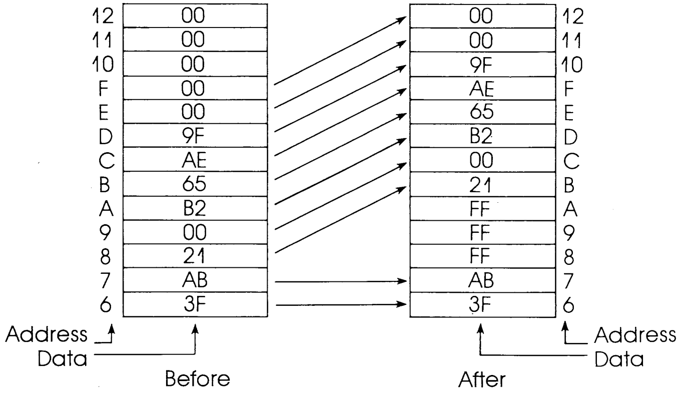
For example, pressing the following keys:

EDIT 1 8 △ A

△ F F SET



causes this insertion:



Address	Data (Before)	Data (After)	Address
12	00	00	12
11	00	00	11
10	00	9F	10
F	00	AE	F
E	00	65	E
D	9F	B2	D
C	AE	00	C
B	65	21	B
A	B2	FF	A
9	00	FF	9
8	21	FF	8
7	AB	AB	7
6	3F	3F	6

Inserting Data Into RAM

DELETE

Data can be deleted from one or more locations with the delete function. To delete data from one RAM location, press:

EDIT 2 fa SET

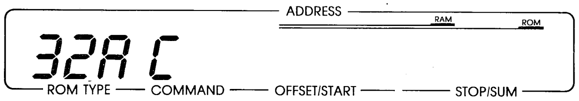
where *fa* is the address from which the data is deleted. To delete data from more than one location, press:

EDIT 2 fa △ la SET

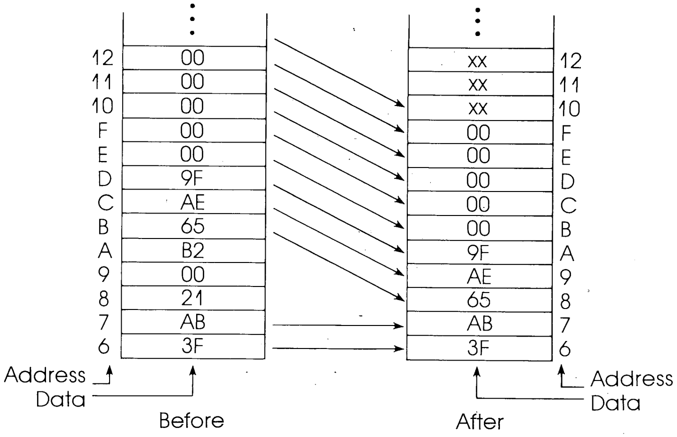
where *fa* and *la* are the first and last locations, respectively, from which data is deleted. In all delete operations, data is moved down from addresses above the deleted data to take the deleted data's place.

For example, pressing the following keys:

EDIT 2 8 △ A SET



causes this deletion:



Address	Data (Before)	Data (After)	Address
	⋮	⋮	
12	00	xx	12
11	00	xx	11
10	00	xx	10
F	00	00	F
E	00	00	E
D	9F	00	D
C	AE	00	C
B	65	00	B
A	B2	9F	A
9	00	AE	9
8	21	65	8
7	AB	AB	7
6	3F	3F	6

Deleting Data From RAM

BLOCK STORE

Data can be stored in a range of RAM locations or in every location with the block store function. To store data in every location of the programmer RAM, press:

EDIT 3 data SET

where *data* is the data to be stored in RAM. To store data in a range of RAM locations, press:

EDIT 3 fa △ la △

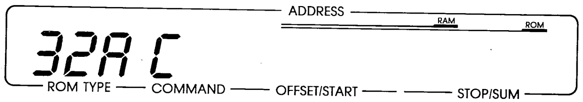
data SET

where *fa* and *la* are the first and last locations, respectively, in which data is stored. Data stored in a RAM location replaces the data that previously was contained in that location.

For example, pressing the following keys:

EDIT 3 8 △ A

△ F F SET



replaces the data in locations 8 hex through A hex of RAM with the data FF hex.

BLOCK MOVE

A block or group of RAM data can be moved to another part of RAM with the block move editing command. To perform a block move, press:

EDIT 4 fa △ la △

n SET

where *fa* designates the first address whose contents are to be moved, *n* indicates the number of consecutive addresses whose data are to be moved, and *la* indicates the address to which the data is moved.

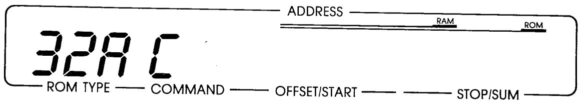
Data moved from one location to another replaces the data at the new location. The data being moved is not deleted from its original location—a copy of that data is written to the new location. The values *fa*, *la*, and *n* are subject to the following rules:

1. fa + n < la
2. fa + n < 8000 hex

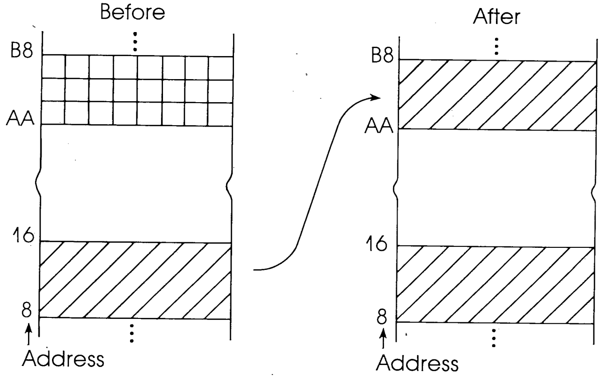
These rules prevent moving data on top of itself, or beyond the end of programmer RAM. As an example of the block move command, pressing:

EDIT 4 8 △ A A

△ F SET



makes a copy of the F hex (15 decimal) bytes of RAM data from locations 8 hex through 16 hex in locations AA hex through B8 hex, replacing the data in AA through B8, as shown in the following figure:



Moving a Block of RAM Data

SEARCH

RAM data can be searched for particular values with the search function. To search all locations in programmer RAM for specific data, press:

EDIT 5 data SET

where *data* is the data you are looking for in RAM. To search a range of locations in programmer RAM for specific data, press:

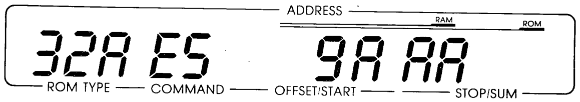
EDIT 5 fa △ la △

data SET

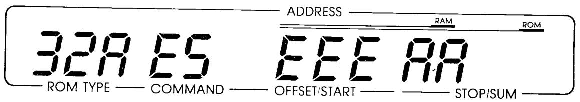
where *fa* and *la* are the first and last locations, respectively, to be searched for *data*.

In either case, if the data is found, the first address at which it is found and the data at that address are displayed. You can find the next location containing the same data by pressing the △ key. When no more matches with the specified data are found, *PASS* is displayed. For example, the following command sequence searches for and finds data at two locations in RAM:

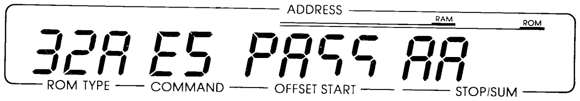
EDIT 5 A A SET



△



△



BLOCK SEARCH

The block search function is similar to the search function, except that you can search for a sequence of up to four byte values. Block searching can be performed only on the entire RAM area. Start the block search by pressing:

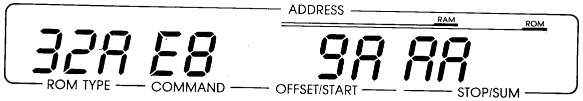
EDIT 8 byte1 byte2 byte3 byte4

SET

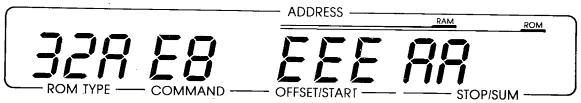
where *byte1* through *byte4* are the consecutive data bytes that are searched for in RAM. If the sequence of data bytes is found in RAM, the location at which that sequence begins is displayed. You can find the next location containing the same data by pressing the △ key. When no more matches with the specified data are found, *PASS* is displayed. For example, the following command sequence searches for and finds a sequence of data bytes at two locations in RAM:

EDIT 8 A A B B

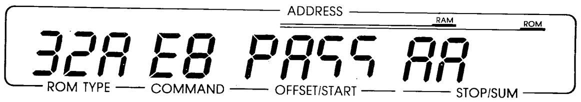
C C D D SET



△



△



Location 9B hex contains BB hex, 9C contains CC, and 9D contains DD. Location EEF contains BB hex, EF0 CC, and EF1 contains DD. The sequence AABBCCDD is not found anywhere else in RAM.

COMMUNICATIONS

The 21A can communicate with a variety of external devices. Communication protocols and data translation formats can be set from the keyboard. This section explains the setting of the protocols and translation formats, procedures for transferring files to and from the 21A, and remote control of the programmer.

The basic procedure for setting up the 21A for communication is:

1. Set the communication protocols.
2. Connect the communications cable.
3. If needed, select a data translation format.
4. Select the desired communication mode.

Each step is described below.

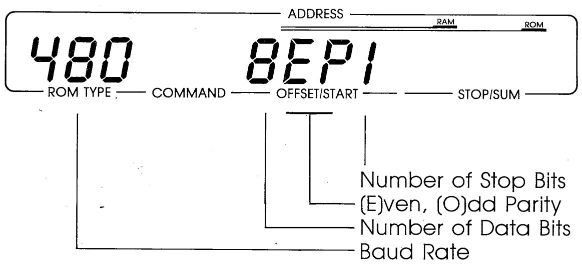
PROTOCOLS

The correct baud rate, number of data bits and stop bits, and parity must be set before communicating with an external device. Perform the following steps:

1. Display the current protocols:

   SELECT A SET

The following is a sample display of the protocols:



2. Set the baud rate, parity, and number of data and stop bits. Press:

SELECT A baud △ config SET

where *baud* is the baud rate code and *config* is the word configuration code according to the following two tables:

Baud Rate Codes

Code	Baud Rate
11	110
30	300
60	600
120	1200
240	2400
480	4800
960	9600

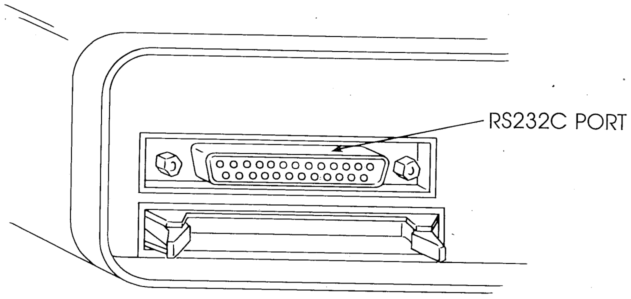
Word Configuration Codes

Code	Description
20	7 bits, even parity, 2 stop bits
24	7 bits, odd parity, 2 stop bits
28	7 bits, even parity, 1 stop bit
2C	7 bits, odd parity, 1 stop bit
30	8 bits, 2 stop bits
34	8 bits, 1 stop bit
38	8 bits, even parity, 1 stop bit
3C	8 bits, odd parity, 1 stop bit

The baud rate can be set without changing the word configuration:

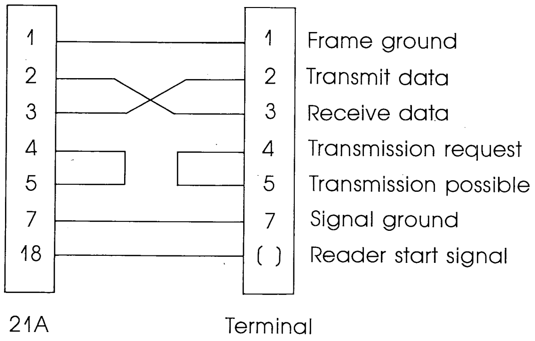
SELECT A baud SET

3. Connect a cable to the RS232C port at the right rear of the programmer:



Location of RS232C Port

The cable should be wired as follows:



Cable Configuration

4. Make sure that the external device connected to the 21A is set to the same communication protocols as the 21A.

5. Set the desired translation format, if necessary. Press:

[SELECT] [9] offset [△] tf [SET]

where *offset* is an offset address, and *tf* is the desired translation format. In general the offset address can be set to zero, but in some cases a non-zero offset may be needed. Consult the programmer manual for more information about the offset. The translation format code *tf* should be entered according to the following table:

Translation Format Codes

Code	Format
01	DG binary
02	DEC binary
28	Subformat specification (see manual)
29	ASCII Hex: TR-Hex without stop mark
2A	ASCII Hex: TR-Hex with stop mark
30	Intellec Hexadecimal
40	Motorola Exormacs
50	Tektronix Hexadecimal
60	Extended Tektronix Hexadecimal
70	ASM-86 Hexadecimal

You are now ready to perform any of the communications operations discussed in the remainder of this section.

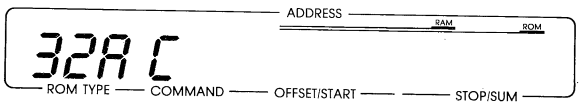
TRANSFERRING DATA TO THE 21A

Data can be transferred to the 21A through the serial RS232C port at the right rear of the programmer.

1. Make sure that the correct translation format has been selected and that the external device is ready to transmit data.

2. Press:

[SELECT] [6] [SET]



The character, C, appears in the command field when the transfer is complete. Address offsets can be applied to data transfers. See your programmer manual for more information.

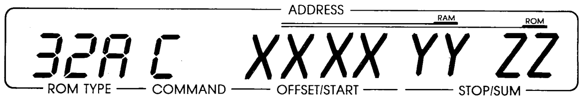
COMPARING INCOMING DATA TO RAM

Data being transferred to the 21A through the RS232C port can be compared to the contents of the programmer RAM. Follow these steps:

1. Make sure that the correct translation format has been selected and that the external device is ready to transmit data.
2. Press:

SELECT 7 SET

If the incoming data matches that already in RAM, the word *PASS* is shown in the address section of the display. If the incoming data does not match RAM data, the first mismatch is displayed as follows:



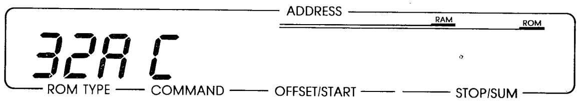
where *xxxx* is the RAM address containing data that does not match the incoming data, *yy* is the RAM data at that location, and *zz* is the incoming data that mismatched.

TRANSFERRING DATA FROM THE 21A

Data can be transferred from the 21A to an external device through the serial RS232C port at the right rear of the programmer.

1. Make sure that the correct translation format has been selected and that the external device is ready to receive data.
2. Press:

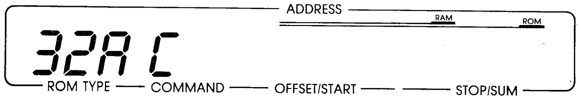
SELECT 8 SET



The character, C, appears in the command field when the transfer is complete.

The procedure presented above transfers all RAM data to the external device. Instead, you can transfer just a section of RAM data by specifying a starting and ending RAM address. Press:

SELECT 8 *fa* △ *la* SET



where *fa* and *la* are the first and last addresses, respectively, to be transferred.

For example, the command sequence:

SELECT 8 0 △ 1 0 0 SET

transfers bytes 0 through 100 hex to an external device via the RS232C port. Address offsets can be applied to data transfers. See your programmer manual for more information.

REMOTE CONTROL OF THE 21A

The 21A has two remote control modes, terminal mode and CPU mode. The desired mode is selected as shown in the following table:

Key Sequence	Mode Selected
SELECT C 0 SET	Terminal mode
SELECT C 1 SET	CPU

Both modes provide normal programmer operation from a remote terminal or computer. Terminal mode and CPU mode differ in the following ways:

Terminal Mode: Echoes commands and resulting displays to the remote terminal. A null code of 100 characters is sent before and after any translation format output.

CPU Mode: Does not echo commands to the remote unit. Does not send a null code before or after translation format output. Displays an asterisk to indicate command and function completion.

Terminal mode is the usual mode used to provide simple operation of the programmer from a remote terminal. CPU mode is provided so that a computer can drive the programmer; this will, in general, be performed by custom software.

In both terminal and CPU modes, one-character codes are used to enter commands and data from the remote unit. The table on the following page shows the proper codes.

To terminate remote control mode, press the reset key. If you select remote control mode from a remote unit, the 21A returns to normal, front-panel operation.

Correspondence Between Keys and Commands: Remote Control

21A Key	Remote Key	21A Key	Remote Key	21A Key	Remote Key
SELECT	S	0	0	8	8
EDIT	O	1	1	9	9
ROM TYPE	R	2	2	A	A
DEVICE	P	3	3	B	B
RESET	ESC	4	4	C	C
SET	CR	5	5	D	D
△	SP	6	6	E	E
▽	/	7	7	F	F

NOTE:
ESC is an escape character ($1B_{hex}$)
CR is a carriage return ($0D_{hex}$)
SP is a space character (20_{hex})

Error Messages

The following table lists the most common error messages you will encounter while using the 21A. If you receive an error message that is not listed here, refer to the programmer manual.

Error Number	Error Description
06	No device is inserted or the device is inserted incorrectly. Insert a device or try turning it around in the socket.
10-17	Verify errors. See manual.
34	The PROM code entered is invalid.
35	The PROM is not electrically erasable. The PROM code must be E16, 815, or 816.
40	You have not specified a subcommand number after pressing either the *EDIT* or *SELECT* command key. These commands require selection of a subcommand.
42	The first address specified must be less than the last address specified for block operations. This is true for both RAM and device addresses.
43	For a block operation, you did not specify the first and last addresses, or the first address, last address, and data were specified in the wrong order.
83	The translation format code entered is invalid.
86	A translation format error has occurred. Retransmit data.
8A	A checksum error has occurred. Retransmit data.
8C	A parity error has occurred. Retransmit data.

CORRECTION: REMOTE OPERATION

Step 2 of the procedure for setting the baud rate and word configuration on page 34 of the 21A Operator's Guide will not work in all cases. To ensure correct remote operation, the following steps should be taken.

1. Turn the programmer off by depressing the power pushbutton at the rear of the programmer.
2. Locate the switches inside the door on the front of the programmer.
3. Set the switches to the appropriate setting according to the following tables:

BAUD Switch Settings for Correct Baud Rate

Switch Position	Baud Rate
0	110
1	300
2	600
3	1200
4	2400
5	4800
6	9600

WORD Switch Settings for Correct Word Configuration

Switch Position	Decription
0	7 bits, even parity, 2 stop bits
1	7 bits, odd parity, 2 stop bits
2	7 bits, even parity, 1 stop bit
3	7 bits, odd parity, 1 stop bit
4	8 bits, 2 stop bits
5	8 bits, 1 stop bit
6	8 bits, even parity, 1 stop bit
7	8 bits, odd parity, 1 stop bit

4. Turn the programmer on.

Note: The translation format codes can be entered from the keyboard as described in the Operator's Guide.