PROMAC

Model 2A

Universal MOS Device Programmer

USER'S GUIDE

REV. 11.21

PROSHOW

USER'S GUIDE

Contents

Features	1
Overview	
Operation	3
General Operation	3
Subcommands	5
Typical Displays	7
RAM/ROM Address Control	9 & 10
ROM Type Selection	11
16-Bit Data	13
Serial Port	14
Parallel Port	18
Details	
Operation	19
ROM Type Selection	19
Adding New ROM Types	19
Single Chip Processors	20
Device Commands	20
"NORM" 8-Bit Mode Versus 16-Bit Modes	20
Checksum	20
ROM & RAM Address Control	21
Edit Commands	21
Function Commands	23
16-Bit Data	25
Interface	27
Serial Port	27
Data Formats	27
Parallel Port	29
Remote Control	31
Maintenance	39
Specifications	45
Device Selection Charts	47 – 60
Buffer "Page" Addresses	62
Warranty	Inside Back Cover

Features

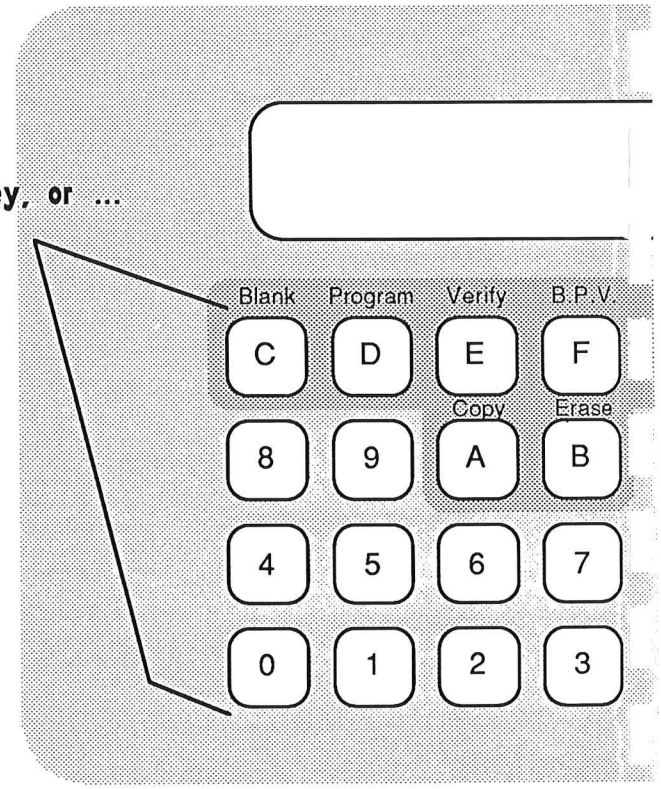
* Menu driven, self prompting operation
* 20 x 2 LCD
* Complete menu of ROMs by manufacturer, device number, and size
* 4MBit, 512KByte buffer
* Programs 1MBit devices now and is ready for future 4MBit devices
* Very fast programming time and general operation (16-bit 68000 CPU and special hardware to reduce device operation times) (programming times: Hitatchi 27256G, 40 sec., Intel Quick Pulse P27256, 17 sec.)
* Parallel port, and serial port with 19.2K baud for fast file transfers
* Extensive editing commands
* Programming verification with Vcc at 4.75V and 5.25V
* Device protection by reverse/incorrect insertion test, "cold" sockets, optional silicon signature check
* Extensive self tests
* Socket adapters available for programming single chip CPUs and EPLD logic devices
* New device algorithms can be added by reading an update ROM placed in the front panel socket
* Full remote control capability
* Battery back-up of settings such as baud rate, ROM type, etc.
* Accepts world wide AC input (85 – 264VAC, 48 – 65Hz)
* Manufacturing checks: optional silicon signature check on every device operated on, optional specific checksum test to be sure correct master data is used.
* Optional silicon signature check to either warn of incorrect device, or automatically adapt to device
* 16-bit data: programs 16-bit ROMs, has several edit and program functions for use with parallel 8-bit ROMs for 16-bit processors, optional socket adapter holds two 8-bit ROMs and operates on them as one 16-bit ROM and displays data as 16-bits, edit commands can arrange data in even wider formats
* Full KEPROM programming and editing capability

OVERVIEW

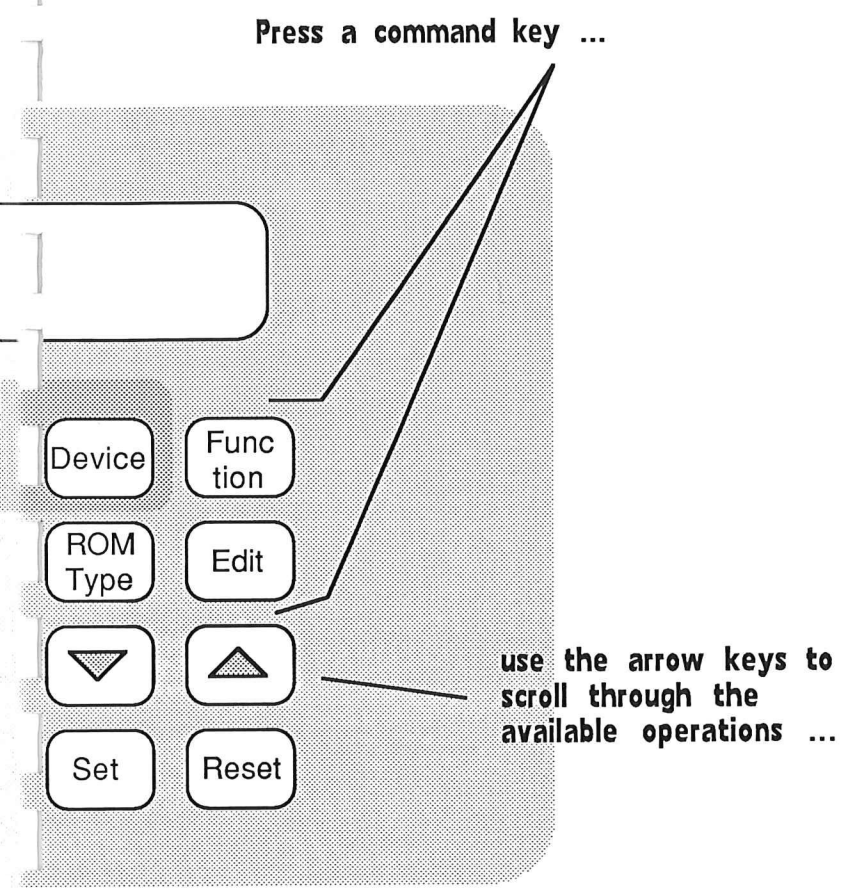
The following few pages will tell you most of what you need to know to operate the P2A in most applications. If you need to know more, continue to the "Details" section.

Scan through this section several times and experiment as you go!

then a Subcommand key, or ...



Operation

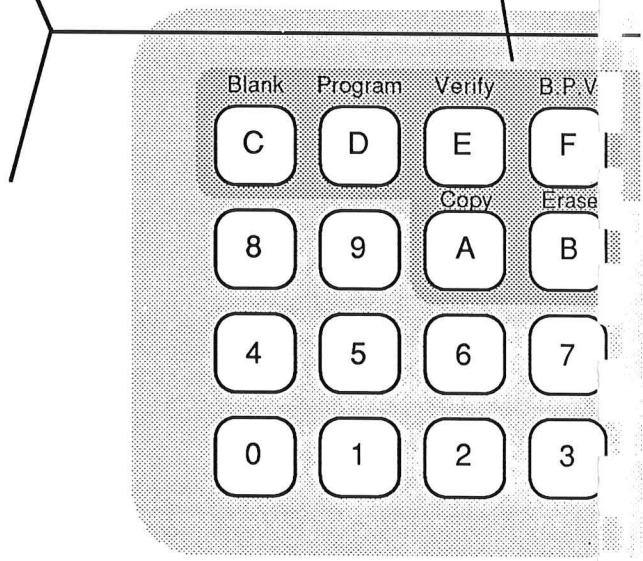

Press a command key ...
Device
Func tion
ROM Type
Edit
use the arrow keys to scroll through the available operations ...
Set
Reset
then follow the prompts on the display.

Function Subcommands

page 23

Buffer Control	**0**	Buffer Page Select
	1	Buffer Start/Stop Address
	3	Buffer Checksum
I/O Control	**4**	BaudRate
	5	Data Format
Serial Port	**6**	Serial Input
	7	Serial Output
	8	Serial Verify
	9	Remote Control
	A	Control Buzzer on/off, etc.
Parallel Port	**C**	Parallel Input
	D	Parallel Output
	E	Parallel Verify

Color highlight denotes Device subcommands.



ROM type selection is covered shortly

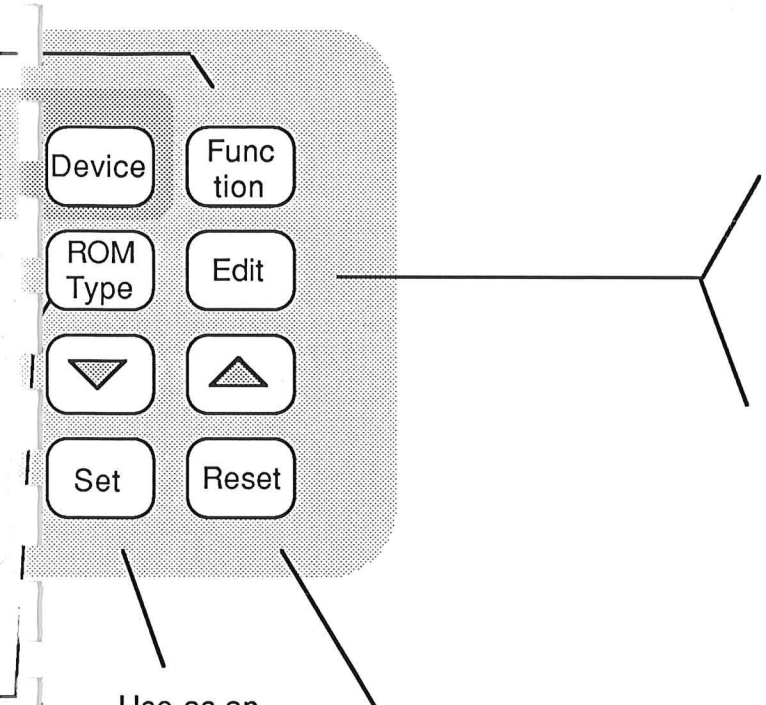
Or just scroll through the options

Press Function, Edit, or ROM Type and scroll.
Arrow keys repeat if held down

Subcommands

Edit Subcommands

To view & alter data, just enter an address.



Use as an Enter key.

Rarely required
Generally, you can go directly from one operation to another without pressing Reset

Data Change Mark **0**
Keeps track of edited locations

Complement **1**

Insert **2**

Delete **3**

Clear RAM **4**

Store a Block **5**

Copy a Block **6**

Search for a Byte **7**

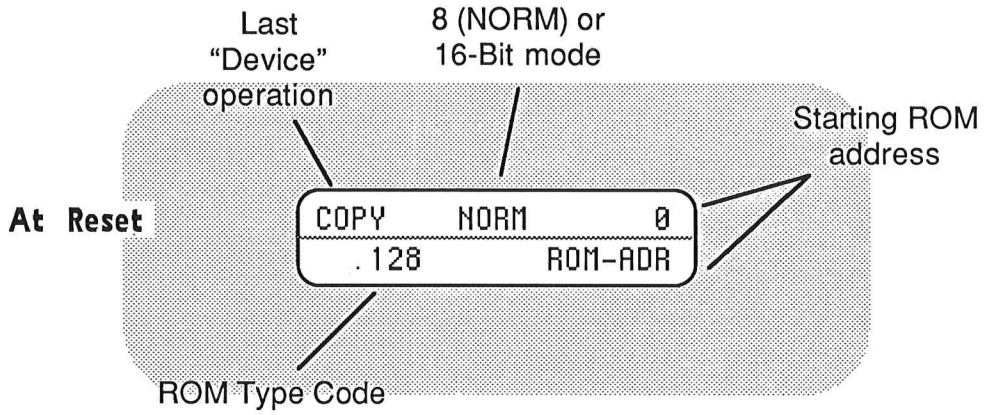
8 -16 Bit Shuffle **8**

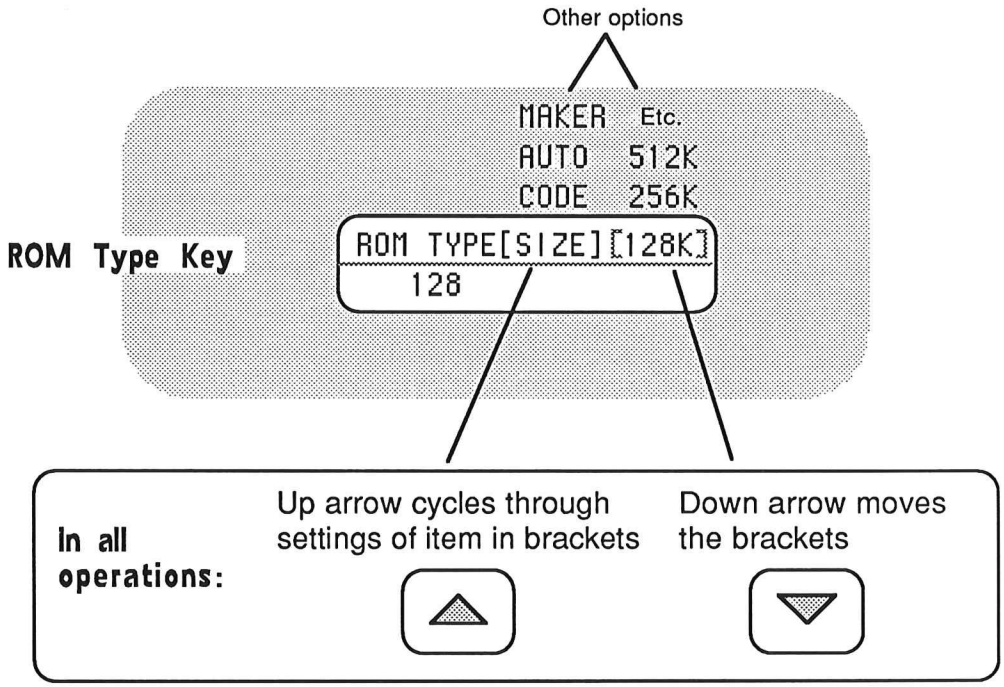
16 - 8 Bit Split **9**

KEPROM **A**
Special KEPROM security codes

Search for a String **B**
1 - 4 bytes long

Typical


Last
"Device"
operation
8 (NORM) or
16-Bit mode
Starting ROM
address
At Reset
COPY NORM 0
.128 ROM-ADR
ROM Type Code

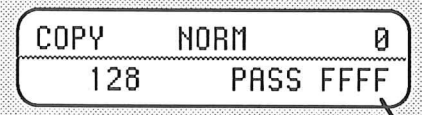

Other options
MAKER Etc.
AUTO 512K
CODE 256K
ROM Type Key
ROM TYPE[SIZE][128K]
128
In all
operations:
Up arrow cycles through
settings of item in brackets
Down arrow moves
the brackets

Displays

Device Key

Buffer Start/Stop address

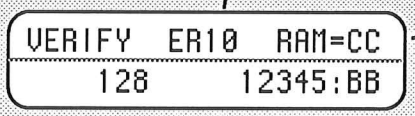
After a Device operation



ROM Checksum

Verify Error

Error Code



RAM Data

ROM Address & Data

Press the up arrow to continue verifying

RAM Address Control

- When a ROM type is selected, the P2A divides the buffer into "pages" the size of the ROM, as shown below

Several of the larger ROM sizes

Addresses for the first four pages

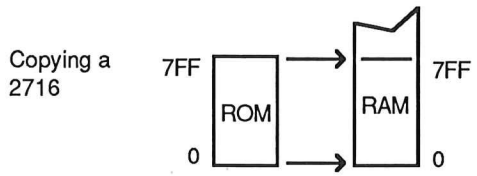
2764 (8K)		27128 (16K)		27256 (32K)		27512 (64K)		271024 (128K)	
Address	Page #	Address	Page #	Address	Page #	Address	Page #	Address	Page #
↑ Continues to end of buffer		↑ Continues to end of buffer		↑ Continues to end of buffer		↑ Continues to end of buffer		End of buffer	
7FFF 6000	3	FFFF C000	3	1FFFF 18000	3	3FFFF 30000	3	7FFFF 60000	3
5FFF 4000	2	BFFF 8000	2	17FFF 10000	2	2FFFF 20000	2	5FFFF 40000	2
3FFF 2000	1	7FFF 4000	1	FFFF 8000	1	1FFFF 10000	1	3FFFF 20000	1
1FFF 0	0	3FFF 0	0	7FFF 0	0	FFFF 0	0	1FFFF 0	0

Abstract from a larger table on page 62

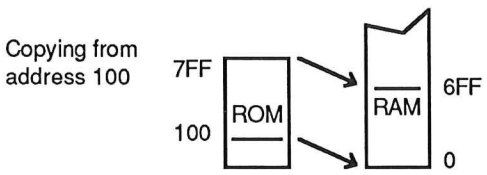
- Most commands assume you want to operate on the address range defined by the current page
- The default page is 0
- A different page can be selected using the Page Set "Function"
- Other start or stop addresses can be selected using the Start/Stop Address "Function"
- These settings are retained until changed, or until a different ROM type is selected
- Most Edit commands will allow entry of a different address range

ROM Address Control

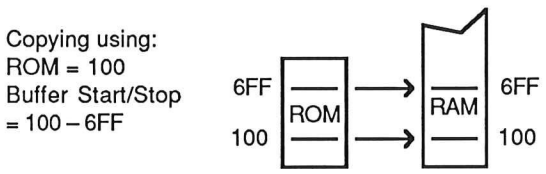
- The default ROM address for all "Device " operations is 0



- A different ROM start address can be specified at the time a Device operation is performed
- This address affects only the ROM, not the buffer

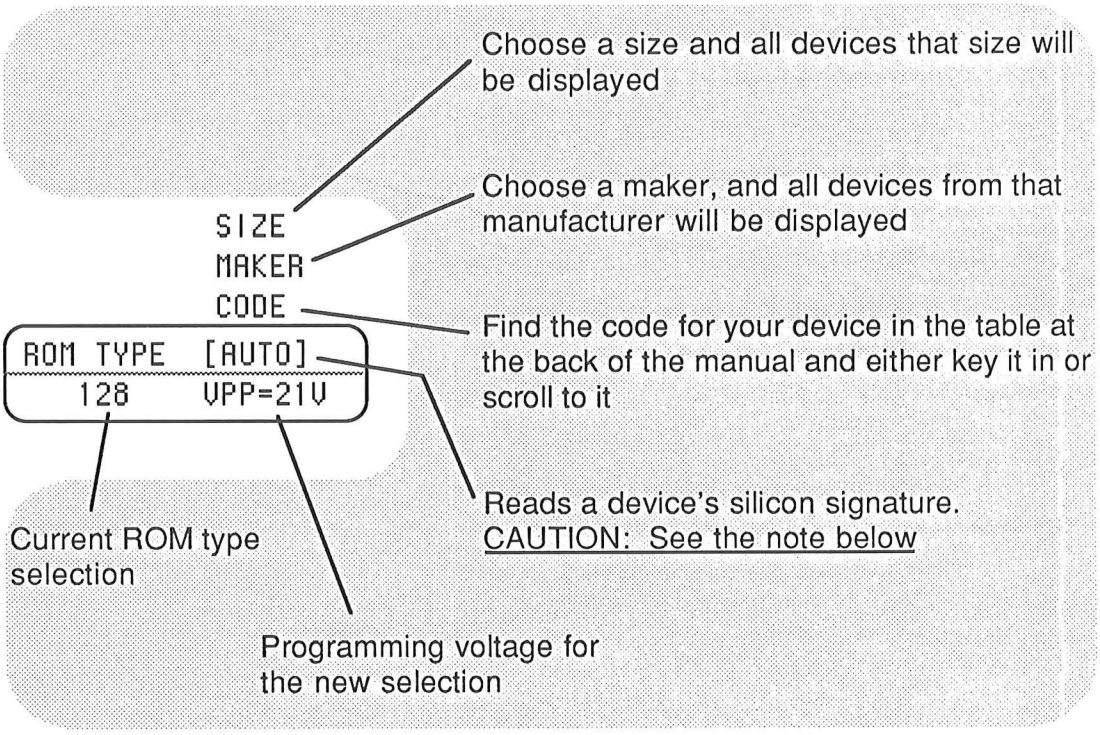


- This setting is retained for future Device operations
- A ROM stop address cannot be specified. To operate on a specific address range, change the buffer start/stop address (using the Start/Stop Function command)



ROM Type Selection

- Press the ROM Type key. Selection can be made by one of four methods



- AUTO: Use this only on devices which have a silicon signature or the device may be damaged. Also, note that once Auto has been used to select the ROM type, all devices placed in the socket will have a silicon signature check performed on them. This test can be turned off by re-selecting the ROM type using one of the other methods

Some Non-Obvious Items

- Block Move Edit command: the data between the first two addresses is moved to the third.

- Search Edit command: the Search-for-a-Byte command contains an optional "Mask" byte which is logically ANDed with the data if only certain bits are to be checked. For normal searches simply ignore it

- Data Change Mark Edit command: buffer locations that have been manually altered are remembered. This function steps through memory and finds them and shows what the data is, and what it was. Any operation which disturbs the buffer (e.g., Complement, Insert, etc.) will cause these locations to be forgotten

- Control Function command: these are: buzzer on/off, reversed-device insertion test on/off, fixed-checksum test on/off (for use in production, to be sure the correct master data is being used), auto silicon signature test on/off (tests the silicon signature of all devices placed in the socket. CAUTION: To avoid damaging devices read the description of the auto test on page xx)

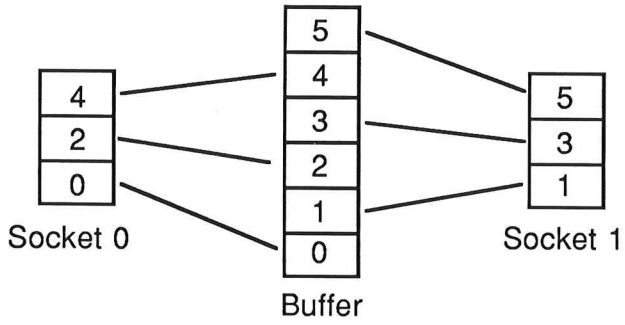
- Verify: the verification performed during Verify, B.P.V., Blank Check, Erase, and Copy is a two pass verification with Vcc at ±5%. The Program command performs a one pass verify. If additional verification is desired, follow the Program command with the Verify command

- Up Arrow: just press the up arrow to enter the data-edit mode just past the last address viewed

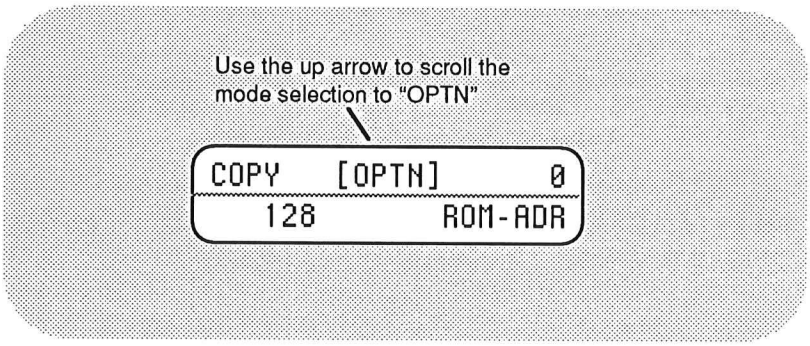
- Buffer Editing: To view/edit data, just enter an address and press Set. The Edit key is not required. To change data, key it in and press Set. Be sure to press Set or the data will not be entered.

16-Bit Data

The P2A has three methods of dealing with 16-bit data. The easiest is to use the optional AF-9778 adapter. It provides two sockets to accept two ROMs acting as one 16–bit ROM. Data from one socket is copied to or from the odd addresses in the buffer, data from the other socket is copied to or from the even addresses in the buffer, as shown



- The "Option" mode must be used. Press the Device key and a subcommand key such as Copy, but before pressing Set use the up arrow to select the Option mode



- The other methods use the Even & Odd Device modes, and the Split & Shuffle Edit commands. See page 24

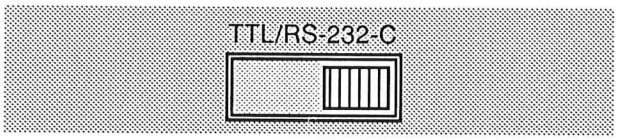
Serial Port

page 27

You will need to know these things about your system

Item	Possibilities	Your System
Baud Rate	110, 300, 600, 1200, 2400, 4800, 9600, 19.2K	
Bits/Character	7 or 8	
Parity Bit	Used or not used. If used, Active (Even or Odd), or Fixed (1 or 0).	
Stop Bits	1 or 2	
DTE (Terminal) or DCE (Modem)	DTE or DCE	
Duplex	Full (echo) or Half (no echo)	
DTR Line	Required?	
RTS Line	Required?	
Other Line	Name	
Control Characters	ACK/NAK XON/XOFF	
Data Format	Intellec Hex, TekHex, etc.	

Be sure the TTL/RS-232-C switch is on RS-232-C for normal operation

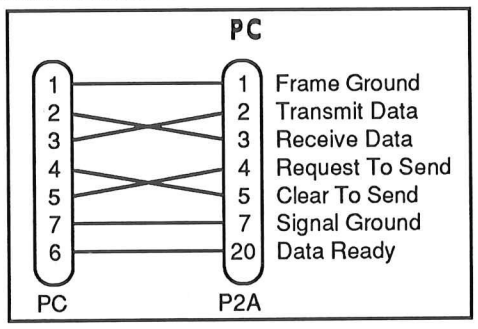


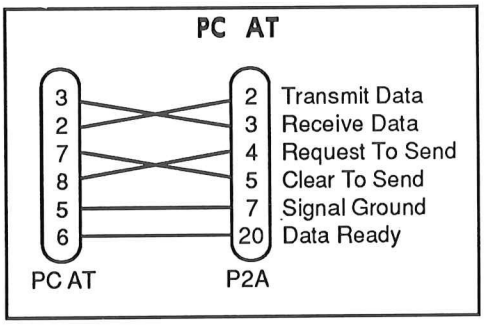
Cabling

In addition to hardware handshaking, XON/XOFF handshaking is also supported. It is selected using the Control "Function" page 5

Signal	Abbre-viation	Pin	Direction	Description
Ground	GND	1		Frame Ground.
Transmit Data	TXD*	2	←	Transmit Data.
Receive Data	RCD*	3	→	Receive Data.
Request To Send	RTS	4	←	Active when the P2A is in the remote or data transfer mode.
Clear To Send	CTS	5	→	Must be held active for the P2A to transmit.
Data Set Ready	DSR	6		Not used.
Signal Ground	SGD	7		Signal Ground.
Data Terminal Ready	DTR	20	←	Held active. 820Ω +12V

NOTES: 1. (*) Indicates low level active.
2. The TXD, RCD, RTS, and CTS lines can all be TTL level or RS-232-C level, depending on the state of the TTL/RS-232-C switch on the back of the unit.



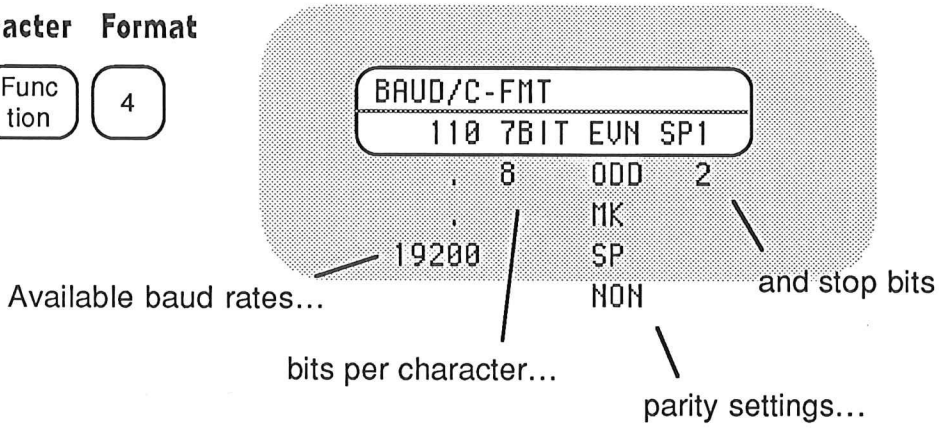


Character & Data Format

Duplex Mode — The P2A operates in the full duplex mode only

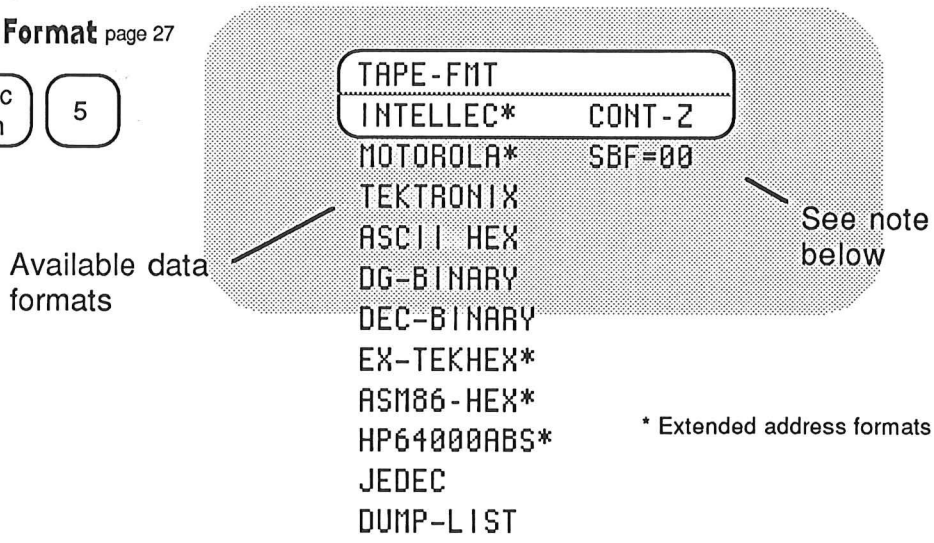
Character Format

Func tion 4



Data Format page 27

Func tion 5

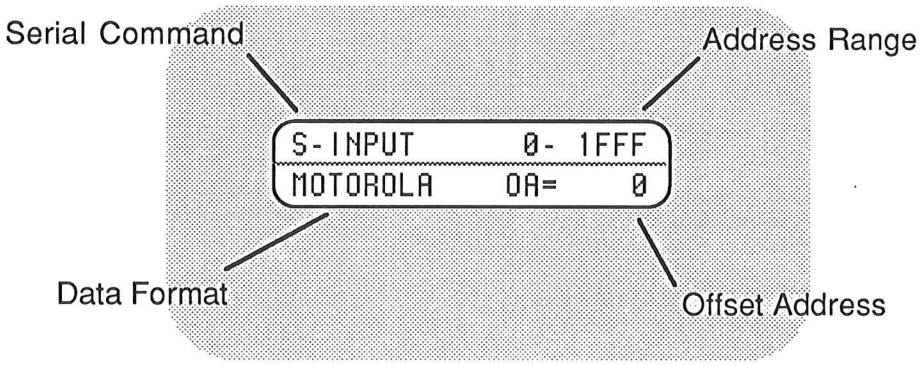


- In the Intellec and ASM86 formats the character sent after an End record is selectable. The options are either Control Z, or a hex character entered by you.

- In the ASCII format a subformat (SBF) must be specified (page 30)

Serial I/O Commands

Function 6 Input 7 Output 8 Verify



Input & Verify

- Data loads to the address specified by the file, not to the "address range"
- The Offset Address is subtracted from the address in the file (e.g., if a data record has an address of 1000 and you want to load it at address 0 in the buffer, use an offset of 1000)
- Maximum offset varies from 4 to 8 digits depending on the capabilities of the format. The P2A's display automatically adapts to this

Output

- The "address range" determines what data is output. The address range can be changed by the Page and Start/Stop Address "Functions" page 5
- The offset is added to the address in the file (the data still comes from the same place, but the address included in the data–record has the offset added to it)
- Maximum offset varies from 4 to 8 digits depending on the capabilities of the format. The P2A's display automatically adapts to this

Special Formats

- The binary formats (DEC and DG Binary) contain no address. Therefore, all transfers happen within the "address range"
- The binary formats are faster (page 28)
- The JEDEC and Dump–List formats are special cases (page 28). JEDEC is for logic devices such as PALs (the P2A accepts fuse data only, not test vector data). Dump–List is an output-only, formatted text-dump, especially useful for viewing data on a screen or print-out

Parallel Port

Page 29

Function C Input D Output E Verify

- Standard Centronics interface and cable
- Easier to establish than the serial port
- Faster than the serial port
- Can be used for data transfer but not for remote control
- Operation is the same as for the serial port
- If connected to a printer, use the Dump–List output which generates a formatted data print-out

DETAILS

OPERATION

For general operation see pages 2 – 13

ROM Type Selection pg. 11

Note: Be sure to use the correct ROM type setting.

Use of the wrong setting to program a part may cause the part to be inadequately programmed or even damaged. Don't make assumptions; for instance a 2764A should not be programmed using the 2764 setting; a Fujitsu 2764 should not be programmed using the Intel 2764 settitng. Either find the correct setting from the table at the back of the manual, or use the Size or Maker menus to find the exact part you are using, or use the silicon signature as described below.

Auto Selection

Selection of the ROM type by "AUTO" reads the devices silicon signature. Silicon signature checking is also implemented in two other ways, as follows.

CAUTION: If an attempt is made to read the silicon signature of a device which does not have a silicon signature, the device may be damaged. Read the following to avoid damaging devices.

Once Auto has been used to select the ROM type, *all* devices placed in the socket will have a silicon signature check performed on them (at the start of each Device operation). This can *only* be turned off by re-selecting the ROM type using one of the other methods (as described on page 11. Since the correct ROM type code is now displayed, simply re-select the ROM type using that code.) Turning off the Auto selection under the Control "Function" does *not* affect this. (This Auto Control selection is a separate feature which also affects silicon signature, and of which you should be aware in order to avoid damaging parts. It is covered on page 23.)

Adding New ROM Types

New devices are constantly being added to the list of ROM types the P2A can program. If a device you wish to program is not shown on the most recent ROM list you have, first determine if the device's programming algorithm is the equivalent of one already on the list (check the device's data sheet or check with your vendor). If not, see if your Promac distributor has a more recent list (the revision number on the list should match the revision number displayed by the P2A at power-on, though there still may be a more recent list). The new list may show a setting for the device, or there may be an update to the P2A that includes the device.

The P2A has a valuable feature in its ability to easily have new devices added to its memory. Update ROMs can be acquired from your Promac distributor as they become available. The update ROM is installed in the P2A's front panel socket and read into internal EEPROM memory. The "Option" Function is used for this purpose. The new devices will show-up in the P2A's device listing in its display.

Single Chip Processors

There are basically two kinds of 40-pin devices, the megabit EPROMs (271024 and up), and the single-chip processors. The megabit EPROMs can be programmed directly from the P2A's 40-pin socket. The single-chip processors require special socket adapters. The processors and their corresponding adapters are shown on the ROM list at the back of the manual. The adapters are available from your Promac distributor.

Device Commands

Device commands are commands involving the ROM. They are:

Blank — Checks that a device is all FFs

Copy — Loads data into the buffer and verifies it

Program — Programs the ROM

Verify — Verifies the ROM to the buffer

B.P.V. — Performs a Blank–Program–Verify sequence

Erase — Erases EEPROMs and blank-checks them. Some EEPROMs can, in effect, have new data written over the old. The Erase command does not apply to these parts.

Except during Program, verification is performed at both Vcc +5% and Vcc -5%. During Program a single verification is performed at whatever level Vcc is required to be during programming. This is for speed in programming. If additional verification is required, use the Verify command.

ROM & RAM Address Control pgs. 9 – 10

"NORM" 8-Bit Mode Versus 16-Bit Modes pgs. 13 & 24

The display of NORM, EVEN, ODD, & OPTN during Device commands relates to the use of 16-bit data. For normal 8-bit operation use the NORM mode.

Checksum pg. 8

The checksum displayed at the end of each Device operation is of the ROM, or of that portion of it operated on. A checksum of the RAM can be displayed using the Checksum Function (pgs. 5 & 23). Also note the Fixed Checksum feature under the Control Function which ensures the use of the correct master data (pg. 24).

ROM & RAM Address Control pgs. 9 & 10

ROM and RAM addresses are controlled separately. RAM addresses affect both Device commands (determining what section of the buffer will interact with the ROM), and Edit commands (determining what area of the buffer will be Complemented, Searched, etc). ROM address control consists of specifying an alternative ROM start address at the time a Device command is performed. A ROM stop address cannot be specified; rather, the buffer stop address can be used to limit an operation.

RAM Addressing pg. 9

When a ROM type is selected, the P2A automatically breaks the buffer up into "pages" the size of the device type. The buffer page is set to zero and the buffer start and stop addresses are set to equal the begining and ending of that page. Thereafter, most operations involving the buffer occur within this address range; the exceptions are certain Edit commands which ask for an address range, and the RAM Clear command which affects the entire buffer.

The RAM start and stop address can be changed in one of three ways. Using the Page Set Function a different page can be selected. This is convenient for operations such as Programming a large block of data into several ROMs. (e.g., To program 4K of data into two 2K ROMs, program the first ROM using the default page 0, then select page 1 and program the second ROM.)

To select odd start or stop addresses not even with the boundries of the ROM, use the Start–Stop Address Function. Selections made using the Page Set or Start–Stop Address Functions are remembered until changed, or until a different ROM type is selected. The third way to alter the buffer start and stop addresses is to change them when performing an Edit command. As mentioned, some commands request an address range; others offer the current start–stop addresses, but will allow a different address range to be specified. These options are indicated in the display. Note that addresses entered in this fashion are not remembered.

ROM Addressing pg. 10

Operations involving the ROM assume the operation should be on the entire ROM. A different start address can be specified when a Device command is entered. This setting is then retained for future Device operations. Note that this address affects *only* the ROM, not the buffer (e.g., if a start address of 100 is entered for a Copy command, the data will be read starting at address 100 in the ROM but will still load to address 0 in the buffer (or whatever the current buffer start address is)). To Copy from address 100 in the ROM to address 100 in the buffer, use the Start–Stop Address Function to change the buffer start address to 100, and specify ROM address 100 when the Copy command is entered.

A ROM stop address cannot be specified. Device operations continue until either the end of the ROM is reached, or the buffer stop address is reached. Therefore, to Copy ROM addresses 0 – 100 to the buffer, change the buffer stop address to 100.

Edit Commands pg. 6

Data Edit/Review - To simply view and alter data does not require use of the Edit key. Rather, simply key in an address and press Set. That address, the address preceeding it, and three following it will be displayed. Use the arrow keys to move through memory. To

change data, key it in and press Set. *Be sure to press Set*, not one of the arrow keys, or the data will not be entered. The new data is shown in the display. Also note that the P2A remembers where you last viewed data. To return to that location just press one of the arrow keys.

Data Change Mark - 0 - Remembers up to 256 of the last locations manually altered. This command steps through memory and shows what the data is and what it was. This command operates on the entire buffer. Note that any command which disturbs the buffer will cause these locations to be forgotten (e.g. loading a ROM into the buffer, clearing the buffer, complementing the buffer, serial input, etc.).

Complement - 1 - Performs a 1's complement of a section of memory.

Insert - 2 - Opens a "hole" in memory, either one byte or a block, and fills it with a data byte specified by the user. Data from the insertion point *to the end of memory* is shifted upward. The data that was at the end of memory is lost.

Delete - 3 - A byte or a block of data is deleted and all memory from the deletion point *to the end of memory* is shifted downward to cover the deleted area. The "hole" that is opened up at the end of memory is filled with FF.

Ram Clear - 4 - Fills the *entire* buffer with FF.

Block Store - 5 - Fills the specified address range with the specified data byte.

Block Move - 6 - Copies the data specified by the first two addresses to the location specified by the third address. The original is retained as well. The original and the duplicate must not overlap.

Search - 7 - Searches the specified address range for matches with the specified data byte ("Ref–Data"). The Search command steps through memory showing all the matches. An optional mask byte ("Msk–Data") can be specified. For normal searches no mask byte should be specified. The mask byte can be used if only certain bits are to be checked. For instance if only the least-signifigant-bit is to be searched, the mask would be 01. Note that the unused bits of the reference data byte should be 0s.

Shuffle -8 & Split - 9 - These are for use with 16-bit data and are specified on page 25.

Special Mode (27916) - A - The 27916 KEPROM has several locations, related to its data protection capabilities, whose data must be specified separately from normal programming data. This command gives access to these bytes. Address 401 is displayed first. The arrow keys can be used to step through the rest of the security-related addresses. To change the data in a location, enter the new data and press Set. Be sure to press Set, not the arrow key, or the data will not be entered.

Note that this data is not the same as data in the regular data buffer and does not affect the regular data buffer. When a 27916 is programmed, the data specified by this Edit function will be used to program the security addresses.

String Search - B - Searches the specified address range for matches to a string of data from 1 to 4 bytes long. The command steps through memory and shows all matches.

Function Commands pg. 5

Page Set - 0 - Changes the buffer page. See "RAM Addressing" page 9.

Start-Stop Address - 1 - Changes the buffer start and stop addresses. See "RAM Addressing" page 9.

Option - 2 - Adds new programming algorithms. See "Adding New Devices" page 19.

Checksum (RAM) - 3 - Gives a checksum of the specified address range of RAM, whereas the checksum shown after Device commands is of the ROM. The checksum is four digits, and is the sum of the data. There is also a fixed checksum feature which is part of the Control Function described later in this section.

Baud/Character Fmt. - 4 & Data Fmt. - 5 - These two commands are described in the "Interface" section on page 16.

Serial Input - 6 & Output - 7 & Verify - 8 - See page 17.

Remote Control - 9 - See page 31.

Control - A - This is a miscellaneous command. It provides control of:

Buzzer - On or off.

Insertion Check - A test is constantly being performed on the socket to see if a device has been inserted reversed or incorrectly. This test is performed without damaging the part. Occasionally a device type will trigger this error even when inserted correctly. The test can be turned off using this command.

Auto Check - The Auto check reads the silicon signature of the device in the socket every time the socket lever is locked down.

CAUTION: If an attempt is made to read the silicon signature of a device which does not have silicon signature, the device may be damaged.

There are two versions of this check. We will call them "Control Auto" and "ROM Type Auto". ROM Type Auto is enabled by selecting the ROM type using the "Auto" method (pg. 11). The difference between the two is that the Control Auto check is performed when the device is inserted in the socket and the P2A changes its ROM type setting to match. With ROM Type Auto, the check is performed at the start of each Device operation, and if the P2A finds the device's signature does not match the

current ROM type, an error is generated. Control Auto is a convenience feature. ROM Type Auto is a safety feature. Note that for the Control Auto test to be enabled, the Insertion Check must also be enabled. The P2A checks that the device is installed correctly before reading its signature in order to avoid damaging it.

XON/XOFF - Enables or disables use of XON/XOFF control character handshaking in serial port operation.

Sum - This feature ensures that the correct data is being used for programming, etc. Enter a checksum and enable the feature's operation using this command. Thereafter, whenever a Device command is executed, if the ROM checksum displayed at the end of the operation does not match the number entered here, an error is indicated. This avoids problems such as accidentally altering the data in the buffer and then programming devices with the erroneous data.

DC Test -B - Described in the "Maintenance" section, page 39.

Parallel Input - C & Output - D & Verify - E - Described in the "Interface" section, page 18.

Self Test -F - Described in the "Maintenance" section, page 39.

16-Bit Data pg. 13

While the P2A's buffer is 8-bits wide, it can deal with 16-bit and even wider data formats. Generally this is done by treating every two bytes in the buffer as a word. There are four methods of using 16-bit data. The easiest, as described below, is to use the optional AF–9778 socket adapter. It provides two sockets, one interacting with the even addresses in the buffer, the other interacting with the odd addresses. It is a convenient way of dealing with two 8-bit ROMs being used as one 16-bit ROM, and is recommended for frequent work with 16-bit data. The adapter is used with the Option ("OPTN") Device mode.

The second method is similar, but without the adapter. It involves using the Even and Odd Device modes. The Even mode causing the ROM to interact only with the even addresses in the buffer, and the Odd mode doing the complement. This performs the same function as the adapter, only operating on one ROM at a time.

The third method uses the Shuffle and Split Edit commands. These convert 8- and 16-bit data formats as described below. These commands can also be used to deal with wider data formats.

The fourth method is to use true 16-bit ROMs, as some of the newer 1Mbit ROMs are, being arranged as 512K x 16 rather than 1024K x 8. The P2A can accommodate these.

Note that the P2A's data display is convenient for viewing 16- or 32-bit data because it displays several bytes side by side.

Finally, a reminder not to overlook the obvious: Once you've made a set of master ROMs, it's no longer necessary to deal with them as 16-bit data. Just duplicate them as you would any other ROM.

16-Bit Adapter pg. 13

Place the AF–9778 adapter in the P2A's socket and select the Option mode of Device operation (selected by scrolling from "Norm" at the start of a Device operation). Socket 0 on the adapter interacts with the even addresses in the buffer, socket 1 with the odd addresses. While the buffer stop address will reflect the ROM size as always, since two ROMs are being used twice as much of the buffer is used. The P2A automatically adapts to this and doubles the address at which it stops. This only affects Device operations, not Edit or other operations. Also note that in this mode the P2A cannot accept an alternate ROM start address, it must be 0. Otherwise, operation is normal.

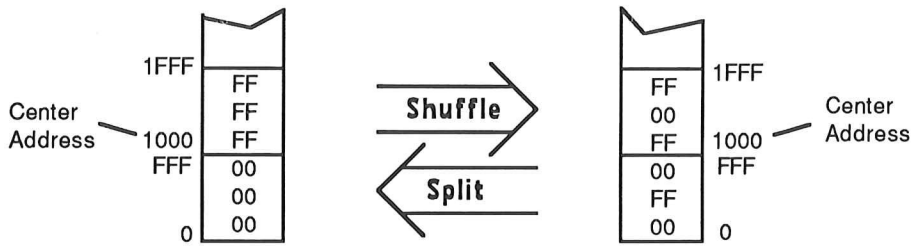
The devices are operated on one at a time. So during Verification for instance, the P2A first steps through all the errors in the Socket 0/Even Addresses combination, and then steps through the Socket 1/Odd Addresses. The address shown in the display is the ROM address, as always.

Even/Odd Mode

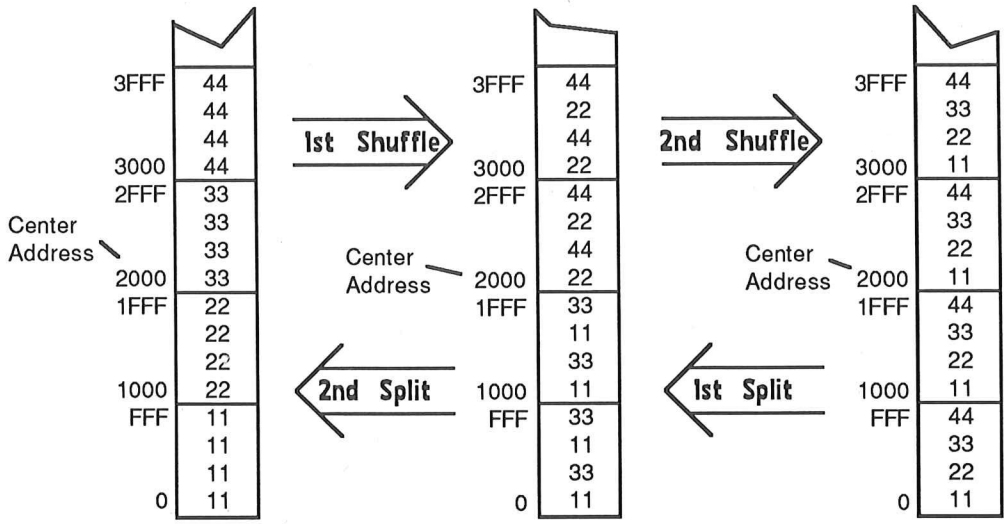
Even and Odd modes (selected by scrolling from "Norm" at the start of a Device operation) operate exactly the same as the 16-bit adapter mode except on one ROM at a time. The only other difference is that a ROM start address can be specified. Note that if the buffer start address is an even address and you're using the Odd mode, or vice versa, the operation simply starts at the next higher buffer address.

Shuffle/Split pg. 5

The Shuffle and Split Edit commands convert data between the 8- and 16-bit modes as shown in the figure. Shuffle will take the data from two 8-bit ROMs that have been loaded into the buffer and interleave them so they can be read or output as 16-bit data. Split will take 16-bit data and separate it so it can be programmed into two ROMs. Using this method, the Normal Device mode is used, and all operations are as normal. Both commands request a starting address and a center address ("CA").

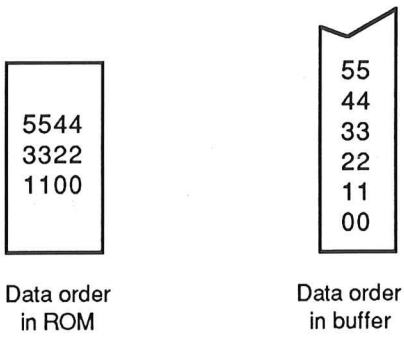


These same commands can also be used to deal with wider data formats. Simply repeat the procedure, as shown.



16-Bit ROMs

A number of the new 1Mbit ROMs are arranged as 512K x 16-bits. The P2A can program these in its 40-pin socket, and needs no adapter. The Normal Device mode is used and all operations are normal. Note that the low order-byte comes first in the buffer, as shown below. If there is an error during verification, the data is displayed as four digits (16-bits) and in the order shown below (e.g., in the example below, the first word would appear as 1100).



INTERFACE

For general interface information see pages 14 – 18

Serial Port pg. 14

TTL/RS-232-C pg. 14

This switch on the back of the unit causes the following serial port lines to operate on either the TTL level (0 – 5V) or the RS-232-C level (-12 – +12V): Transmit Data, Receive Data, Request To Send, and Clear To Send. Note that the Data Terminal Ready line is pulled up to 12V through a resistor and is not affected by this switch.

PC Cabling pg. 15

The PC and PC/AT cable diagrams shown on page 15 show the CTS, and DSR inputs on the PC as being connected. Whether these lines actually need to be connected, or whether the Carrier Detect line needs to be connected as well, depends on your interface software.

Parity pg. 16

The Mark and Space parity settings are available only when 7 data bits are used. The parity setting None, is available only when 8 data bits are used.

Address Offset pg. 17

Most of the data formats the P2A uses contain an address in the file with the data. Normally, on input the P2A loads the data into the buffer at the address specified in the file. An address offset can be used to cause the data to load to a lower address. This serves two purposes. First, data can be directed to the begining of the buffer. Second, if the address in the file is above the end of the buffer, the data *must* be re-directed to a lower address. The offset is subtracted from the address in the file (e.g., a file with an address of 1000 loaded using an offset of 1000 will load starting at 0).

On output, the address the P2A sends with the file is normally the address the data came from in the buffer. An offset can be used to send a different address with the file. The data still comes from the same place, it is the address sent with the file that is changed. The offset is added to the address the data comes from.

Different data formats can handle a different number of address digits. These are spelled out in the following section. On input, the P2A can accept as large an address as the format can provide, but if the address runs above 7FFFF (the end of the buffer) an address offset must be used. On output, the P2A can output as large an address as the format allows. Normally the output address is limited to 7FFFF, but with an address offset added it may run to more digits. The P2A automatically limits the number of offset digits that can be keyed-in to the appropriate number for the current format.

Data Formats pg. 16

Most of the data formats are manufacturers' formats, e.g. Intel Hex, Tektronix Hex, etc. For most applications, simply select the format that matches your equipment. The ASCII Hex format is the P2A's own format, and has many selectable characteristics. The Dump–List format is simply a text-dump. The binary formats are roughly 3 times faster than the other formats, the DG

Binary format being the simplest (a header of 8 binary 1s, followed by the data in binary). (As a reference point, a file for a 27512, in Intelec format, takes about three and a quarter minutes at 9600 baud.)

In all appropriate formats, the format output by the P2A automatically adapts to extended addresses. For instance the Intellec format only uses extended address records when needed, and the Motorola format uses S1 records for four-digit addresses, S2 records for six-digit addresses, and S3 records for eight-digit addresses.

Also note that on output the P2A precedes all transmissions with three CR/LFs followed by 100 nulls, and ends all transmissions with 100 nulls followed by a CR/LF.

Intellec - Five address digits.

Motorola - Eight address digits.

Tektronix - Four address digits. This means that this format cannot be used for addresses above FFFF. For higher addresses use Ex Tekhex.

Ascii - Four address digits; use only for addresses up to FFFF. This format can be somewhat customized. The subformat ("SBF") character requested when this format is selected, determines many of its characteristics. Refer to the following figure.

DG Binary - No address field, address offset does not apply. Be sure to have the serial port set for 8 data bits. Since this format has no end mark, the P2A has no way of knowing when a transmission to it is completed. Therefore you must either press the Reset key, or set the buffer stop address so that the incoming data reaches that address.

DEC Binary - See DG Binary

Ex TekHex - Eight address digits.

ASM86 Hex - Five address digits. Any ASM86 record type can be received. For output the P2A uses record type 81 for data records, and record type 85 for extended address records.

HP64000ABS - Eight address digits. This format is approximately twice as fast as most of the other formats. The data bus width and data word width are both eight. Be sure to have the serial port set for 8 data bits. Since this format has no end mark, the P2A has no way of knowing when a transmission to it is completed. Therefore, you must either press the Reset key, or set the buffer stop address so that the incoming data reaches that address.

JEDEC - No address; address offset does not apply. This is the standard format for files for logic devices such as PALs. The P2A accepts only fuse data, not test vector data.

Dump-List - Address offset does not apply. This is an output-only, formatted text-dump. If data input is selected the P2A changes its format to Intellec.

Parallel Port pg. 18

The parallel port can be faster and easier. Easier because there are fewer variables involved in initially getting the port working. The Centronics standard is pretty well established, and there aren't as many potential incompatibilities, such as the baud rate setting, which handshake lines to use, etc. The P2A's parallel port can transfer data about 10 times faster than the serial port running at 9600 baud, though actual speed depends on the system its communicating with. (As a point of reference, a file for a 27512, in Intelec format, could be transfered in as little as 16 seconds.) The disadvantages are that it cannot be used for remote control, and that, though the P2A's parallel port is bidirectional, the parallel port on many systems is for output only.

Data formats and address offset apply to the parallel port just as they do the serial port.

ASCII Hex Subformat Byte

0	1	2	3	4	5	6	7	Data Bit
Stop Mark	End Mark	Data Mark		Comment Mark	Address Mark	Start Mark		Characteristic

Stop Mark	0	None
	1	) output) or % input see note 1
End Mark	0	None
	1	ETX
Data Mark	00	, output , or . input
	01	,
	10	, output , or % input
	11	, output , or ' input
Comment Mark	0	None
	1	$
Address Mark	0	#
	1	$A
Start Mark	00	None
	01	STX
	10	[see note 1
	11	None

Sample File

STX # 0000 , 01 , 23 , 45 , etc. , $ Comment LF ETX %

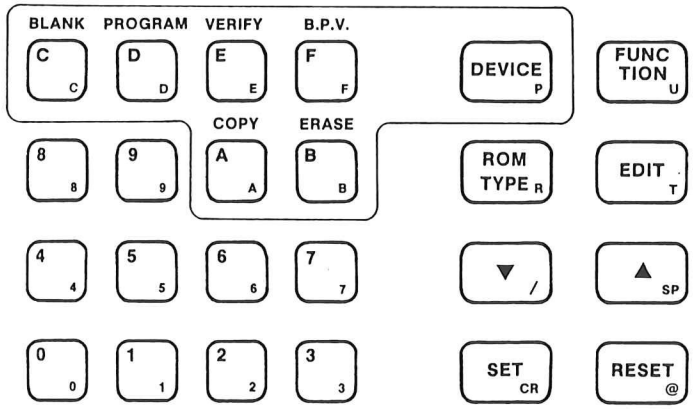
1. When Start Mark [is used, the Stop Mark automatically becomes] regardless of the Stop Mark setting. The Stop Mark] is for input only, it is not output.
2. A Data Mark must precede the first data byte and follow the last.
3. Comments must be ended with a LF. If more data follows, a new Address Mark and address must be sent.
4. Send the Stop Mark last. It causes the P2A to leave the data input mode.

REMOTE CONTROL

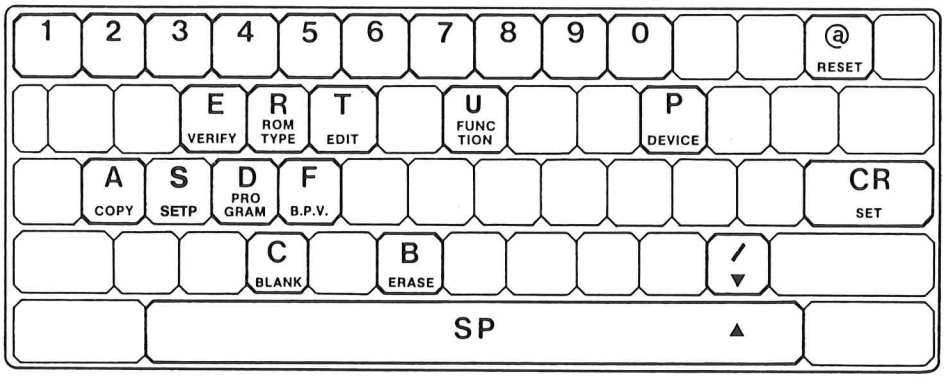
(Function 9)

Remote control is for use in production environments where the programming process is to be automated, or if the user wishes to write a program to drive the P2A. Otherwise operation from the P2A's own keypad is probably easier, with its labeled keys and its display prompts. In the future a menu driven control program will be made available for the P2A.

A key on the keyboard corresponds to each of the P2A's keys, as shown. All operations of the P2A can be performed through remote control, and operation is the same as in the stand-alone mode. If power is turned off while in the remote mode, at power-on the P2A will return to that mode. The "at" key (@) causes the P2A to return to the idle state, but stays in the remote mode. Remote control can be ended either by pressing Reset on the P2A, or by giving the remote control command remotely (U 9 Return).



0-F on the P2A correspond to 0-F on the keyboard.



Remote Control Examples

The following is a sample session in remote control. Commands entered from the keyboard are shown in bold type. Operation in remote mode is generally just as it is in the stand-alone mode. For instance to program a device press P (= Device), D (= Program), and Enter (= Set). The P2A ends all its responses with a carriage return/line feed.

One thing that is different in remote mode is that you cannot scroll through the available commands or options. Rather they are specified by number. Commands are specified by the same number as they normally are, as in the example below where the Edit command Block Store is selected using T (= Edit), and 5, which is the same number used to select Block Store in the stand-alone mode. Certain items which can be selected only by scrolling, such as the Even/Odd/Norm programming mode, are also given number designations in the remote mode. These are shown in the following tables. The one item that can be scrolled through is the list of ROM type codes, as shown in the example below.

Command/Response	Description
[PASS]	On entering remote control.
R	Pressing R and Enter selects the ROM type by silicon signature, regardless of the ROM selection mode indicated in display.
[PASS]	Silicon signature read sucsessfully.
R64A	Entering R and a ROM type code selects the ROM type by code, regardless of the ROM selection mode indicated in display.
R [064A]	R and space causes the current ROM type code to be displayed.
[0264]	Space steps to the next type code.
/[064A]	Slash steps to the previous type code.
@[PASS]	Abort and return to idle.
PA	P for Device, A for Copy.
[FFFF]	Checksum; Copy was sucsessful.
0	Enter an address to view/edit memory.
[0000 FF]	Address and data displayed.
[0001 FF]	Press space to increment data display.
@[PASS]	Exit view/edit mode.
[0002 FF]	Press Space and the data view/edit mode is entered just after the last location viewed.

00	Enter new data.
[0003 FF]	Next location is shown.
/[0002 00]	Slash to decrement data display.
@[PASS]	Exit view/edit mode.
PE	Verify device.
[0002 00,FF] [ERROR 11]	Error indicated by, address, buffer data, ROM data, and an error code.
[FFFF]	Space to continue verifying. Verification complete; ROM checksum shown.
T50 FF AA	T for Edit, 5 for Block Store, 0 is the starting address, FF the ending address, AA the data.
[PASS]	Operation complete.
U6	Prepares the P2A to receive data. Send the data.
[PASS]	P2A received the data.
U9	Exit remote mode.
[PASS]	Sign-off.

Device Commands

Device	Command	Options		Set
Device = P	Copy = A Erase = B Blank = C Program = D Verify = E B.P.V. = F	Norm = 0 Odd = 3 Even = 4 Optn = 5	ROM Address	Enter

Examples

P A Enter	Copy using current ROM address and current mode selection.
P A 3 Enter	Copy, with mode change to Odd.
P A 0 Space address Enter	Copy using a new ROM start address. Mode must be entered first even if staying in the same mode.

Edit Commands

Edit	Command	Options	Set
Edit = T	Data Change Mark = 0 Complement = 1 Insert = 2 Delete = 3 RAM Clear = 4 Block Store = 5 Block Move = 6 Search = 7 Shuffle = 8 Split = 9 KEPROM = A String Search = B	Address, data, etc.	Enter

Example

T 5 0 space 7FF space FF Enter — Block Store 0 to 7FF with FF

Edit Command Formats

Command	Keys	Additional Information
Data Change Mark	T 0	(no additional information)
Complement	T 1	(no additional information) 1st address space 2nd address
Insert	T 2	address space data 1st address space 2nd address space data
Delete	T 3	address 1st address space 2nd address
RAM Clear	T 4	(no additional information)
Block Store	T 5	data 1st address space 2nd address space data
Block Move	T 6	1st address space 2nd address space 3rd address
Search	T 7	mask byte search byte 1st address space 2nd address space mask byte search byte
Shuffle	T 8	center address
Split	T 9	center address
KEPROM	T A	(no additional information)
String Search	T B	data

Comments

Displays new data followed by old. Use space bar to step through memory
Current buffer address range
Specified address range
Insert one byte
Insert a block
Delete one byte
Delete a block
Fill entire buffer with FF
Current buffer address range
Specified address range
1st through 2nd is moved to 3rd
Current buffer address range. Use Mask byte FF for normal searches
Specified address range. Use Mask byte FF for normal searches
Uses current buffer start address. Specify Center address
Uses current buffer start address. Specify Center address.
Operates same as Data View/Edit. See example earlier
1 to 4 bytes with no spaces inbetween

Function Commands

Function	Command	Options	Set
Function = A	Page Set = 0 Start Address = 1 Stop Address = 2 RAM Checksum = 3 Data Format = 5 Serial Input = 6 Serial Output = 7 Serial Verify = 8 Remote Control = 9 Control = A Parallel Input = H Parallel Output = I Parallel Verify = J	New settings, etc.	Enter

Example

U 0 1 Enter Changes the current buffer page to 1.

Function Command Formats

Command	Keys	Additional Information
Page Set	U 0	page number
Start Address	U 1	address
Stop Address	U 2	address
Checksum	U 3	(no additinal information)
Data Format	U 5	format number space subformat number
Serial Input	U 6	offset address
Serial Output	U 7	offset address
Serial Verify	U 8	offset address
Remote Control	U 9	(no additional information)
Control	U A	0
		1
		2
		3
		4 checksum
Parallel Input	U H	offset address
Parallel Output	U I	offset address
Parallel Verify	U J	offset address

Data Formats

0	Intellec	Subformat end character required (pages x and x)
1	Motorola	
2	Tektronix	
5	ASCII Hex	Subformat specification required (pages x and x)
6	DG Binary	
7	DEC Binary	
8	Ex TekHex	
9	ASM86-Hex	Subformat end character required (pages x and x)
A	HP64000ABS	
B	JEDEC	
C	Dump-List	

Comments

To check current page number, do not enter a new page number
To check current address, do not enter a new address
To check current address, do not enter a new address
Checksum of current page of RAM
To check current format, do not enter a new format. Subformat where applicable
Offset is optional
Offset is optional
Offset is optional
Ends remote control
Buzzer on/off
Reverse insertion test on/off
Automatic adaption to ROM type by silicon signature on/off
XON/XOFF software handshaking on/off
Fixed checksum on/off. To turn it off, do not specify a checksum
Offset is optional
Offset is optional
Offset is optional

MAINTENANCE

Introduction

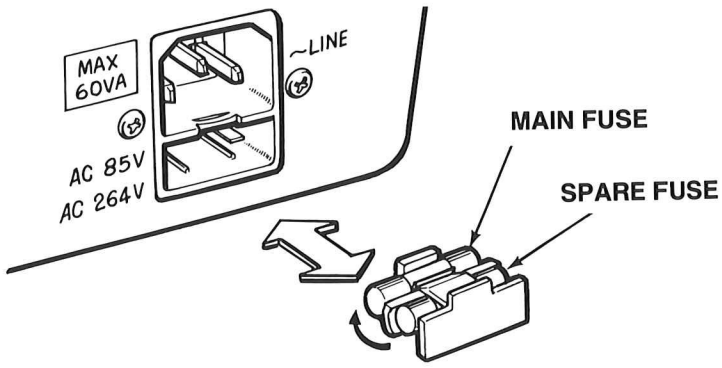
The P2A includes numerous built-in tests. At power-on both the buffer RAM and working RAM are tested. Also the battery for the battery-backed data is tested. An error is indicated if either fail.

Other tests include, check and adjustment of Vcc, Vpp, and the P2A's internal comparator voltages, and checks of the address and data buses to the ROM socket. These tests should be part of any calibration or trouble-shooting procedure. The P2A also has the ability to be put in a programming loop, continually performing the programming algorithm for the current ROM type. There are also tests of the serial and parallel ports, and of the display.

Fuse Replacement

The fuse is 1AG, 250V, 630mA. The fuse and a spare are located directly under the power cord connector.

CAUTION: In order to avoid electrical shock, be sure the unit is turned off and the power cord disconnected before replacing the fuse.



Regular Maintenance

The only regular maintenance needed is to periodically replace the ROM socket and the internal battery. The ROM socket should be replaced after about 5000 insertions, and the P2A has an internal counter to keep track of this. Refer to the "Socket Replacement" heading. It may also help to occasionally clean the contacts of the socket with alcohol or some suitable cleaner to ensure good contact with the ROM.

The internal battery should last several years. When it fails, an error inication will be given at power-on. The battery can be replaced as described below.

Battery Replacement

The internal battery is for retaining settings such as ROM type, baud rate, etc. When it fails, an error indication is given at power-on. To replace the battery you will need to open the P2A, remove the main PC board, de-solder the old battery, and install a new one. The battery is a lithium type, Toshiba CR–2032 or equivalent.

1. Turn power off and disconect the power cord.

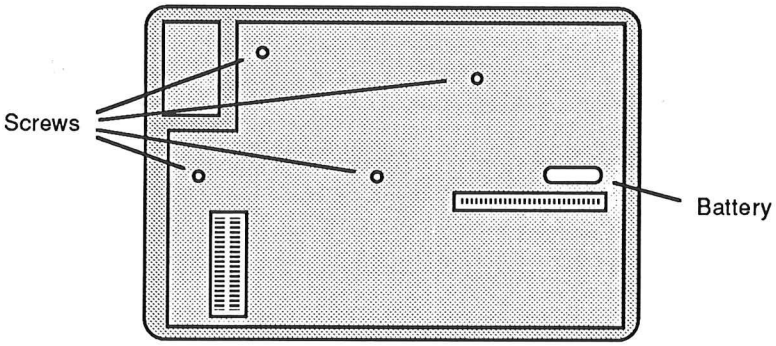
CAUTION: AC wiring is exposed once the P2A is opened. Disconnect the power cord before opening the P2A in order to avoid electrical shock.

2. Remove the four screws in the corners of the bottom of the unit.

3. Open the P2A and disconnect the ribbon cable connecting the two halves.

4. Disconnect the power cable connector.

5. Remove the four screws securing the lower PC board.



6. De-solder and replace the battery. Be sure the polarity matches that indicated on the PC board.

7. Re-assemble the P2A. Be sure to reconnect both the power cable and the ribbon cable connecting the two halves of the unit.

8. Test the unit. The first time it is turned on it will give a battery failure error. Turn it off and back on, and the error should be gone.

Socket Replacement

The ROM socket should be replaced after about 5000 insertions. It may also need to be cleaned from time to time with alcohol or some suitable cleaner in order to assure good contact with the ROM.

The P2A has an internal counter to keep track of the number of times the socket has been used. To check this counter, press Function F (Self-Test) and use the up arrow to scroll to "INS. COUNT". To leave the count as is, press Set or Reset. To clear the counter when the socket is replaced, use the down arrow to move the brackets to the number. Press the up arrow and the number will be replaced by the word "CLEAR". Press Set. The counter is now reset. (If the counter is ever cleared unintentionally, it can be restored by moving the brackets to the word "CLEAR" and pressing the up arrow. If this is done immediately after the counter was cleared, the previous count will be restored.)

To replace the socket, remove the two screws on its top and pull straight up. Replace it with a Textool 240-1280-19-0602J or equivalent.

Voltage Tests and Adjustments

The voltages to be checked and adjusted include Vcc, Vpp, and the internal comparator voltages. A test is also included in this section to check the P2A's ability to deliver these voltages to the appropriate pins. There are separate tests for the address and data buses to the socket. These are covered in the "Address and Data Bus Tests" section.

These tests will require the following items.

Potentiometer: 0 – 1.25KΩ, 2W
or
Potentiometer: 0 – 300Ω, 2W &
Resistor: 1.25KΩ, 0.5W
Also
VOM: up to 125mA, up to 50V, down to 10mV

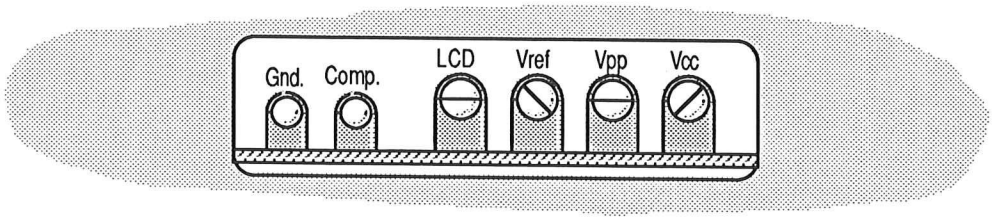
The voltages can be checked without the resistors, but not under load. The removeable door on the side of the unit gives access to the adjustment potentiometers.

Select the test by pressing Function B (DC Test) and using the up arrow to step through the tests. Use the table shown opposite to go through the tests.

Address and Data Bus Tests

The address and data bus tests require connecting numerous pins of the ROM socket together. The P2A then outputs signals on some of the pins and reads them back on others. The results are shown in the display and any errors indicated. The easiest way to make the socket connections is to use a socket adapter as a test jig and solder the wires to it. Then just place it in the P2A's socket as needed. Pin connection lists for the two test jigs are given below.

CAUTION: Do not make test connections until the desired test number is displayed, and remove test connections before stepping to the next test.



Use Function, B, and the Up Arrow to step through the tests. Pin 20 is ground

Test #	Voltage	Tolerance	Load	Pin	Comment
1	25.00	±1.00	1.25KΩ 20mA .5W	1	
2	21.00	±0.50	280Ω 75mA 2W	5	Adjust 25V,
3	12.75	±0.25	212Ω 60mA 1W	7	Vpp then check
4	12.50	±0.30	208Ω 60mA .75W	28	the others
5	5.00	±0.25	125Ω 40mA .25W	29	
1	6.25	±0.25	50Ω 125mA 1W	40	
2	6.00	±0.25	48Ω 125mA .75W	36	Adjust 6.25V,
3	5.25	±0.25	42Ω 125mA .75W	34	Vcc then check the
4	5.00	±0.25	40Ω 125mA .75W	32	others
5	4.75	±0.25	38Ω 125mA .75W	40	
6	2.35	±0.24		Use	Use comparator test
7	1.50	±0.15		comparator	Com- point & comparator
8	0.50	±0.05	None	test	parator ground. Only -1.0V
9	-3.70	±0.37		points	can be adjusted.
10	-1.00	±0.10			Adjust with Vref pot.
11	-5.00	±0.50		4	
12	-5.00	±0.50		23	Delivery to
13	5.00	±0.25	None	1, 11, & 30	pins test. No
14	5.00	±0.25		1	adjustment
15	12.00	±0.50		39	
16	≤0.40	—		39	

Address Bus Test Connections

From	To
1	8
2	7
4	30
5	20
6	11
9	39
10	38
12	37
13	36
14	35
15	34
16	33
17	32
18	31
19	29

Data Bus Test Connections

Connect all togeather

1, 3, 5, 7, 9, 12, 14, 16, 18, 22, 24, 26, 28

Connect all togeather

4, 6, 8, 10, 13, 15, 17, 19, 21, 23, 25, 27, 40

Address Bus Test

Select the test by pressing Function B (DC Test) and stepping to test 17, then install the address bus test jig.

CAUTION: Do not install the test jig until you have stepped to test 17. Remove it before going on to the next test.

Each time the up arrow is pressed, the P2A increments the data which is being output. The data is incremented in the manner (00, 11, 22, 44, 88, FF). This process is repeated five times to test all the inputs and outputs. The display indicates, which pass of the test it is on, the data output, and the data read back. If there is an error the P2A will display an error code and the pin affected.

No harm will be done if you press the up arrow one too many times and get to the first indication of test 18, but do not go beyond this point without removing the test jig!

Data Bus Test

Press Function B (DC Test) and step to test 18. Place the data bus test jig in the socket.

CAUTION: Do not install the test jig until you have stepped to test 18. Remove it before performing any other operation.

Each time the up arrow is pressed, the P2A alternately outputs data AA and 55. The data is then read back. This process is repeated three times. The display will indicate the pass the P2A is on, and the data output and read back. If there is an error, the P2A will display an error code and the pin affected. At the end of test 18, the Self Test is completed and the P2A returns

to the idle mode. Be sure to remove the test jig before entering any further commands on the P2A!

Programming Algorithm Test

The P2A can be put into a continuous loop, performing the programming algirithm for the currently selected device type.

CAUTION: Be sure to remove any device from the ROM socket before executing this test.

The test is initiated by pressing Function F (Self Test) and scrolling to the "PULSE-TEST". The P2A will run through a series of addresses and data as it performs the test and these will be indicated in the display. That the algorithm is being performed correctly can be verified by using an oscilloscope and referring to a copy of the device's data sheet and programming specifications.

Display Test

The display test is initiated by pressing Function F (Self Test) and cycling to the "DISPLAY" test. Under the display test are two options, "ALL" or "DATA". The "ALL" test cycles all pixels of the LCD off and on several times. The "DATA" test sequences through all alpha-numeric characters in all display locations.

Serial Port Test

The serial port is tested by connecting the port so that the data is fed back on itself. In the serial port test the P2A displays the data output (which is entered from the P2A's keypad) and the data input, which should match.

The port should have its data lines (pins 2 & 3) and its handshake lines (pins 4 & 5) connected to each other. If the handshake input is not high, the buzzer will sound to indicate an error.

In order for the input and output data to match, the port should be configured for 8 data bits. The test is initiated by pressing Function F (Self Test) and cycling to the "SERIAL" test. The display will indicate the data input and output in both Hex and ASCII.

Parallel Port Test

The parallel port must be connected to an external device to be tested. Connect it to any device with a Centronic's compatible input, such as a printer, to test the P2A's output. Connect it to an output device such as a terminal with a parallel port to test the P2A's input. The test is initiated by pressing Function F (Self Test) and cycling to the "PARALLEL" test. Under the parallel test are options for either an input or an output test. The output test sends a series of

lines of data consisting of the alpha-numeric character set. The input test shows in the display the data received through the port.

Specifications

* AC: 85 – 264VAC, 48 – 65Hz, 15VA Max. during programming
* Environment: 0 – 40°C operating, -10 – 60°C storage, 0 – 80% RH
* Physical: ≈ 52mm H x 290mm W x 210mm D, ≈1.5Kg

P2A Device Chart

AMD	Atmel	Exel	Fujitsu	GI	Hitatchi	Intel	
2716			8516 2716		2716	2716 2816A	
2716B 2817A		46C16 2817A 2816A				2817A	

NEC	Oki	Ricoh	Seeq	Sharp	Seiko	TI	
2716	2716		52B13			2516	
			2817A 5517A 2816A 5516A				

Code	Lattice	Motorola	Matsushita	Mitsubishi	NS	NCR
16		2716	2716	2716	2716 27C16	
16A						
10000						
15000						
1D100						
38100						

Code	Thomson	Toshiba	VTI	WSI	Xicor	
16	2716 C2716					
16A						
10000						
15000						
1D100						
38100					2816A 2816B 2816H	

P2A Device Chart

AMD	Atmel	Exel	Fujitsu	GI	Hitatchi	Intel	
2732 2732A			8532 2732 2732A 27C32A		2732 2732A 2532	2732 2732A	
2732B		46C32					

NEC	Oki	Ricoh	Seeq	Sharp	Seiko	TI	
2732 2732A	2732	5H32 27C32				2732A 2532	
			52B23				

Code	Lattice	Motorola	Matsushita	Mitsubishi	NS	NCR
32 32A			2732	2732	2732 27C32	
232 532		2532			2532	2832
B23 10001 15001 25100		2833				

Code	Thomson	Toshiba	VTI	WSI	Xicor	
32 32A	2732	2732				
232 532 B23						
10001 15001 25100						

P2A Device Chart

AMD	Atmel	Exel	Fujitsu	GI	Hitatchi	Intel	
2764					27C64 2764	2764	
	27HC64			27C64		2764A	
				27HC64		27C64 87C64	
2864A	28C64	2864A	28C64	28C64		2864A	
			2764 27C64		58064		
2764A	27HC641						
		48C64					
9864 2864B		2865A	28C65		58C65	P2764A 2864B	

NEC	Oki	Ricoh	Seeq	Sharp	Seiko	TI	
2764 27C64	2764 27C64	27C64	2764		27C64	2764	
				5762 5763			
28C64	2864A		2864			2564	
			2864H 52B33				
				5764			
		687C64					
				5749			

Code	Lattice	Motorola	Matsushita	Mitsubishi	NS	NCR
64			2764	M5L2764 M5M2764		
64A						
264						52864
664		68764 68766				
864					28C64A	
E64 F64						
10002 12000 15100						
1D000 1D101	64K8					
26000					27C64	

Code	Thomson	Toshiba	VTI	WSI	Xicor	
64	2764	2464 2764				
64A			27C64			
564 864					2864A	
B33					2864B 2864H	
1D000		2764A 2464A				
1D101			28H64			
26000 2B000 2F000						
37000 37001				57C49 57C64		

P2A Device Chart

AMD	Atmel	Exel	Fujitsu	GI	Hitatchi	Intel	
27128					27128	27128	
			27128A		27128A	27128A	
			27128			27128B	
			27C128				
27128A							

NEC	Oki	Ricoh	Seeq	Sharp	Seiko	TI	
27128	27128		27128			27128	
	27C128			57126			
				57127			
				57128			

Code	Lattice	Motorola	Matsushita	Mitsubishi	NS	NCR
128			27128	27128 27C128		
12A						
F12						
916						
10003						
1D001						
37002						

Code	Thomson	Toshiba	VTI	WSI	Xicor	
128	27128	24128 27128				
12A			27C128			
F12						
916						
10003						
1D001		24128A 27128A				
37002				57C128		

P2A Device Chart

AMD	Atmel	Exel	Fujitsu	GI	Hitatchi	Intel	
	27256 27C256		27256 27C256A 27C256H	27256 27C256	27256 27C256	27256 27C256 87C256	
27256 28C256			27C256				
						27256	

NEC	Oki	Ricoh	Seeq	Sharp	Seiko	TI	
27256A 27C256A 27256		27C256	27256 27C256			27256	
27C256							
				57256			

Code	Lattice	Motorola	Matsushita	Mitsubishi	NS	NCR
256				27256 27C256		
F25 10004 10100						
1D002 26001 37003					27C256	
38101						

Code	Thomson	Toshiba	VTI	WSI	Xicor	
256			27C256			
F25		24256				
		27256 24256A 27256A				
10004 10100 1D002		54256				
		57256 54256A 57256A				
26001 37003 38101				57C256	28256 28C256	

P2A Device Chart

AMD	Atmel	Exel	Fujitsu	GI	Hitatchi	Intel	
27512	27C512		27512 27C512	27C5120	27512	27512	
	27C513 27C515			27C5123 27C5121		27513 P27512 P27513	

NEC	Oki	Ricoh	Seeq	Sharp	Seiko	TI	
27C512							

Code	Lattice	Motorola	Matsushita	Mitsubishi	NS	NCR
512				27512		
A51						
513						
12001						
1D003						
1D004						

Code	Thomson	Toshiba	VTI	WSI	Xicor	
512			27C512			
A51						
513						
12001						
1D003						
1D004						

P2A Device Chart

AMD	Atmel	Exel	Fujitsu	GI	Hitatchi	Intel	
27C1024			27C1000 27C1001 27C1024 27C1028		27C101 27C301 27C1024	27010 27011 27210	

NEC	Oki	Ricoh	Seeq	Sharp	Seiko	TI	
27C1000 27C1001 27C1024							

Code	Lattice	Motorola	Matsushita	Mitsubishi	NS	NCR
10005						
17000						
17001						
17002						
17003						
1A000						
1A001						
1A002						
1D005						
1D006						
1D007						
22000				27C100		
22001				27C101		
22002				27C102		
28000						
28001						
28002						
35000						

Code	Thomson	Toshiba	VTI	WSI	Xicor	
10005						
17000						
17001						
17002						
17003						
1A000						
1A001						
1A002						
1D005		571000				
1D006						
1D007		571024				
22000						
22001						
22002						
28000						
28001						
28002						
35000		571001				

Error Codes

#	Display	Description	#	Display	Description
1	**Err 01** **Device Miss Insert**	Device inserted incorrectly	26	**Err 26** **Command Error**	Undefined subcommand. Also, ROM add. not 0 in OPTN mode
2	**Err 02** **Device None**	No device in the socket to operate on	27	**ROM Type Err 27** **Silicon Signature**	No silicon signature
3	**Err 03** **Buffer RAM Error**	Buffer memory error	28	**Err 28** **Silicon Signature**	Wrong silicon signature
4	**Err 04** **Working RAM Error**	P2A's scratch pad memory error	29	**Special Mode Err 29** **KEPROM Set Error**	Invalid KEPROM data
5	**Err 05** **Battery Backup Error**	Battery failure	30	**Err 30** **Address Over 7FFFF**	Address exceeds 7FFFF
10	**Verify Er 10 RAM:xx** **xxxxx:xx**	Error at 5V	31	**Err 31** **STA Over SPA**	Start add. > Stop add. Also, file transfer add. over 7FFFF
11	**Verify Er 11 RAM:xx** **xxxxx:xx**	Error at 4.75V, low Vcc test	32	**Err 32** **1stA Over 2ndA**	1st add. is greater than 2nd add.
12	**Verify Er 12 RAM:xx** **xxxxx:xx**	Error at 5.25V, high Vcc test	33	**Block Move Err 33** **3rdA Input Error**	3rd add. must be beyond 2nd add.
13	**Verify Er 13 RAM:xx** **xxxxx:xx**	Error at 6V during programming. Also KEPROM program error	34	**Err 34** **Center Address Error**	Center add. too high
20	**Err 20** **Subcommand Error**	No subcommand entered	35	**Page Err 35** **Page Over**	Page number too high
21	**Err 21** **Data Non-Input**	Data or address not entered	36	**Err 36** **ROM Size Over**	ROM start add. exceeds ROM size
22	**ROM Type Err 22** **ROM Type Error**	Undefined ROM type	40	**Err 40** **Parity Error**	Interface parity error
23	**Baud/C-Fmt Err 23** **Baud Rate Undefined**	Undefined baud rate	41	**Err 41** **Checksum Error**	Interface checksum error or fixed device checksum error
24	**Tape Fmt Err 24** **Tape Fmt Undefined**	Undefined data format	42	**Err 42** **Verify Error**	Interface verify error
25	**Erase Err 25** **ROM Type Error**	ROM type not erasable or can be programmed without erasing	43	**Err 43** **Tape Format Error**	Data format error

Buffer Page Addresses

2716 (2K)		2732 (4K)		2764 (8K)		27128 (16K)		27256 (32K)		27512 (64K)		271024 (128K)	
Address	Page #	Address	Page #	Address	Page #	Address	Page #	Address	Page #	Address	Page #	Address	Page #
≠ Continues to end of buffer		≠ Continues to end of buffer		≠ Continues to end of buffer		≠ Continues to end of buffer		End of buffer					
7FFF	F	FFFF	F	1FFFF	F	3FFFF	F	7FFFF	F				
7800		F000		1E000		3C000		78000					
77FF	E	EFFF	E	1DFFF	E	3BFFF	E	77FFF	E				
7000		E000		1C000		38000		70000					
6FFF	D	DFFF	D	1BFFF	D	37FFF	D	6FFFF	D				
6800		D000		1A000		34000		68000					
67FF	C	CFFF	C	19FFF	C	33FFF	C	67FFF	C				
6000		C000		18000		30000		60000					
5FFF	B	BFFF	B	17FFF	B	2FFFF	B	5FFFF	B				
5800		B000		16000		2C000		58000					
57FF	A	AFFF	A	15FFF	A	2BFFF	A	57FFF	A				
5000		A000		14000		28000		50000					
4FFF	9	9FFF	9	13FFF	9	27FFF	9	4FFFF	9				
4800		9000		12000		24000		48000					
47FF	8	8FFF	8	11FFF	8	23FFF	8	47FFF	8	End of buffer			
4000		8000		10000		20000		40000					
3FFF	7	7FFF	7	FFFF	7	1FFFF	7	3FFFF	7	7FFFF	7		
3800		7000		E000		1C000		38000		70000			
37FF	6	6FFF	6	DFFF	6	1BFFF	6	37FFF	6	6FFFF	6		
3000		6000		C000		18000		30000		60000			
2FFF	5	5FFF	5	BFFF	5	17FFF	5	2FFFF	5	5FFFF	5		
2800		5000		A000		14000		28000		50000			
27FF	4	4FFF	4	9FFF	4	13FFF	4	27FFF	4	4FFFF	4	End of buffer	
2000		4000		8000		10000		20000		40000			
1FFF	3	3FFF	3	7FFF	3	FFFF	3	1FFFF	3	3FFFF	3	7FFFF	3
1800		3000		6000		C000		18000		30000		60000	
17FF	2	2FFF	2	5FFF	2	BFFF	2	17FFF	2	2FFFF	2	5FFFF	2
1000		2000		4000		8000		10000		20000		40000	
FFF	1	1FFF	1	3FFF	1	7FFF	1	FFFF	1	1FFFF	1	3FFFF	1
800		1000		2000		4000		8000		10000		20000	
7FF	0	FFF	0	1FFF	0	3FFF	0	7FFF	0	FFFF	0	1FFFF	0
0		0		0		0		0		0		0	

JMC

Japan Macnics Corp.

516 Imaiminami-cho, Nakahara-ku,
Kawasaki-City, Kanagawa-ken 211, Japan
TEL: (044) 711-0330 FAX: (044) 733-7150
TLX: JAPMAC J28988