PP28

OPERATING MANUAL

stag PROGRAMMERS

Stag Programmers Ltd
Martinfield
Welwyn Garden City
Herts AL7 1JT
England

1.1 GENERAL DESCRIPTION

The PP28 has been designed for general purpose programming of NMOS EPROMS and EEPROMS. Two ZIF sockets are provided allowing EPROMS with either 24 or 28 pins to be handled. Two slide switches are used to set the PP28 for EPROMS of different types or capacity.

The circutry of the PP28 is protected against common programming hazards such as reversal of the EPROM in the ZIF socket, grounded pins or internal short circuits in the EPROM. All supplies are safely current limited.

The PP28 provides an internal RAM buffer whose capacity is 64K bits (8K x 8 bit words). This uses standard 16K x 1 dynamic RAMS, the remaining capacity being used internally. Data from the RAM is copied into the EPROM when programming. The RAM may be loaded from a Master EPROM, the keyboard or using the serial I/O interface which is supplied as standard.

The display comprises eight 7-segment LEDS and the keyboard has twenty keys. Both display and keyboard serve several purposes.

The main purpose of the display is to show simultaneously address, RAM and EPROM data but it also provides the operator with information such as CHECKSUM and error codes.

The keyboard not only permits entry of hexadecimal data but has 14 dual function keys to give control COMMANDS to the programmer.

The command key marked FN causes the programmer to operate in FUNCTION mode. This enables the keyboard to be used to select and control one of the special functions. Examples of these are the powerful editing functions SPLIT, SHUFFLE, RELOCATE and modification of the UART control parameters.

1.2 MAINS VOLTAGE SELECTION

The PP28 is PRE-WIRED for mains supplies of EITHER 115 VAC ± 10% OR 230 VAC ± 10% and this is clearly marked on the rear panel.

IMPORTANT
Check that you have received a unit suitable for your local supply voltage

Should you wish to modify the PP28 to accept the alternative supply voltage this can be easily accomplished.

Refer to fig 1.2C for details on separation of the two halves of the metal case and to figures 1.2A and 1.2B for details of the transformer connections for each supply voltage.

AC Power may be applied to the PP28 and the display should show the EPROM type corresponding to the settings of the slide switches. The default conditions for the serial I/O interface, see section 2.4.4, are loaded at this time.

It is best to ensure no EPROM is in either ZIF socket whilst AC power is applied or removed.

MAINS TRANSFORMER CONNECTIONS

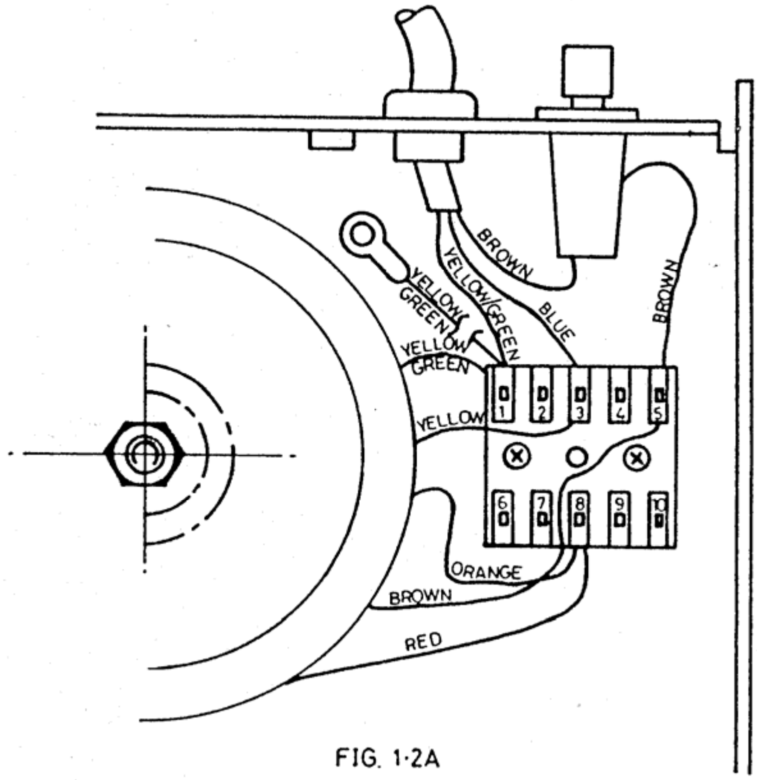


FIG. 1·2A

240V

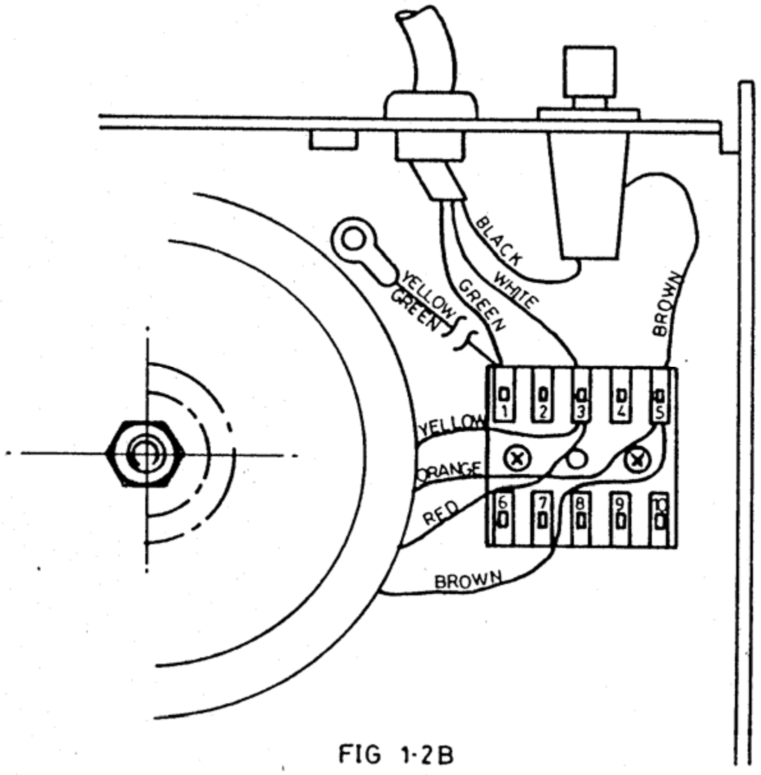
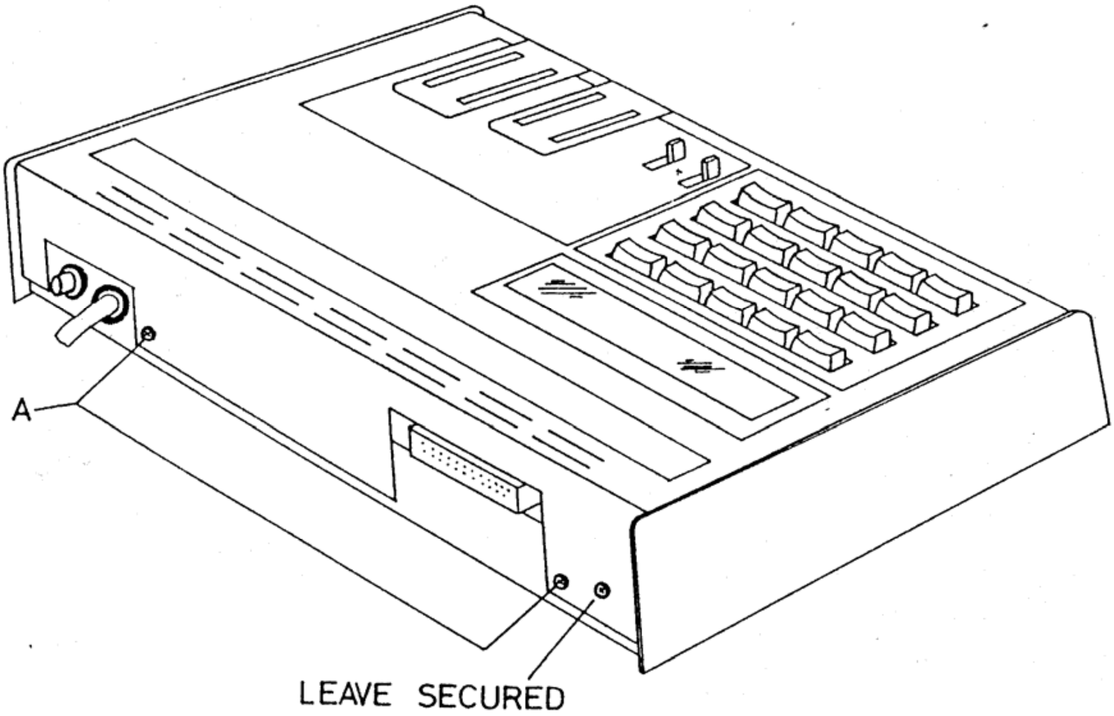
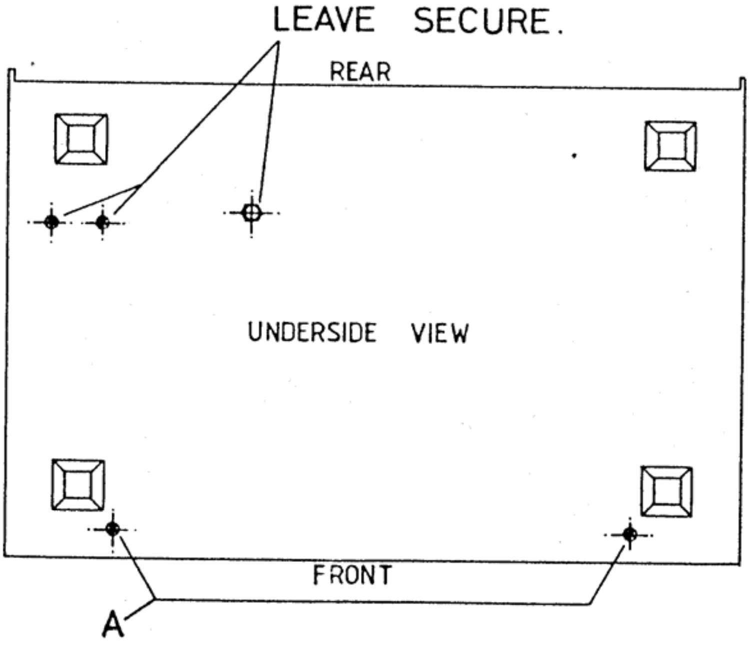


FIG 1·2B

110V



Remove screws 'A' only to open instrument case.



2.1 THE 7-SEGMENT DISPLAY

The display comprises eight 7-segment displays which are arranged in a group of four, the address display and two groups of two, the data display.

Both the address and the data display contain dedicated LEDS. These show the operator whether to enter a RAM address or RAM data when in EDIT mode.

The 7-segment display is not only used for address and data display but also carries numerical information such as checksum values and messages such as "PASS". Further details of the displayed information are given in the sections describing the use of the keyboard COMMANDS and FUNCTIONS.

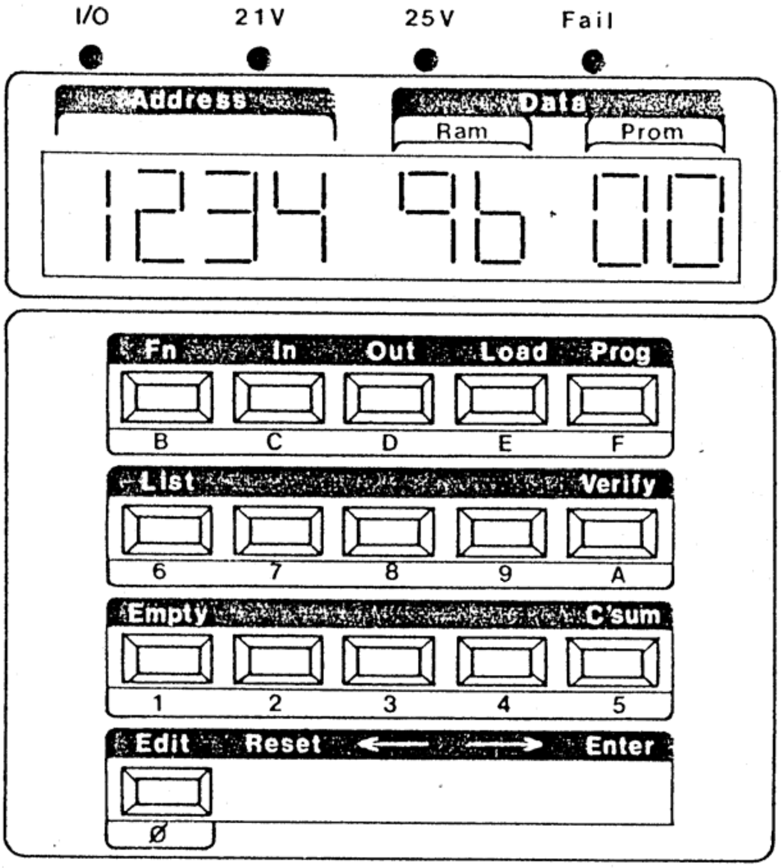
A diagram showing the layout of the display and keyboard is given in fig 2.1.

2.2 DISCRETE LED DISPLAYS

Above the main 7-segment display there are four discrete LEDS. Two of these remind the operator to check that the programming voltage (21V or 25V) is appropriate for the device which it is intended to program.

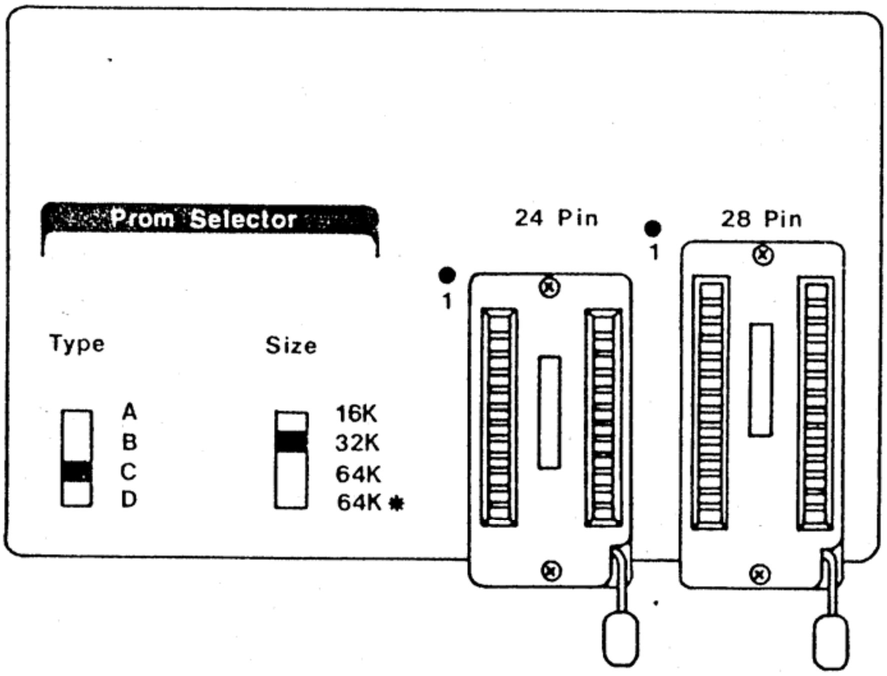
The I/O LED is only illuminated when successful I/O transfers are taking place. It therefore reveals incorrect interface inter-connections or failure of the interface equipment. This is especially useful when making a interface connection for the first time.

Two LEDS are also provided adjacent to Pin 1 of each ZIF socket, see fig 2.2. These show which ZIF socket is active and the correct orientation of the EPROM in the socket.



I/O
21V
25V
Fail
Address
Data
Ram
Prom
1234 96 00
Fn
In
Out
Load
Prog
B
C
D
E
F
List
Verify
6
7
8
9
A
Empty
C'sum
1
2
3
4
5
Edit
Reset
Enter
Ø

FIG 2.1



Prom Selector
Type
A
B
C
D
Size
16K
32K
64K
64K ✱
24 Pin
28 Pin
1
1

FIG. 2.2

2.3 EPROM SELECTION

The PP28 must be set up to correspond to the particular kind of EPROM to be programmed. This is accomplished using the two slide switches. The slide switches are shown in fig 2.2 and the table below gives the exact positions required.

	16K	32K	64K	64K*
A	2716/2516/HN48016 2508/2758	2532	2564	
B		2732	2764	68764/6
C		2732A	2816	MK2764
D	Available for expansion			

Note that the position of the right-hand switch usually corresponds to the EPROM size. The position marked 64K is used for EPROMS which are programmed by the application of VPP (21V or 25V) to pin 1 of the 28 pin socket and 64K* for those which use other pins.

In response to the RESET key the PP28 shows a number which describes the EPROM type which has been selected. EPROM types and the number displayed by the PP28 are as follows:

	EPROM description	VPP	Display
2716/2516	single voltage 16K	25V	2716
	(Intel, Texas or other equivalent see *NOTE below)		
TMS 2532	(Texas)	25V	2532
2732 NMOS	(Intel or equivalent)	25V	2732
2732A HMOS	(Intel)	21V	732A
68764/6	(Motorola)	25V	8764
TMS 2564	(Texas)	25V	2564
2764 HMOS	(Intel)	21V	2764
MK2764	(Mostek Bytewyde)	25V	0764
2816	(Intel EEPROM)	21V	2816
	(see **NOTE below)		

*NOTE

It may be necessary to suppress the in-program verify for Motorola 2716/2532, see sections 2.4.3 and 2.5.3.

The PP28 can also program two kinds of 8K EPROMS, the Intel 2758 and the Texas 2508. In both cases use the same switch setting as for a 2716. Function FB may be used to restrict programming to an upper address limit of 03FF.

The Hitachi EPROM HN48106 should be treated as an ordinary 2716 for program and read operations. In addition use of the function FE will permit electrical erasure of this part.

**RISE TIME CONTROL MODULE (2816)

The Intel EPROM 2816 requires special rise time control of the programming voltage. This is obtained by insertion of the RISE-TIME CONTROL MODULE into the 28 pin ZIF socket. The 2816 itself is placed in the 24 pin ZIF socket. As a result of this the 2816 requires that the size selector switch is placed in the 64K position. It is not necessary to perform as erase operation before programming the Intel 2816 as this is included in the programming algorithm. To perform a "chip erase" set the RAM data for every address to FF using the "fill command" and program the part in the normal way.

2.4 USE OF THE KEYBOARD

The PP28 keyboard is multi-purpose and is used to enter HEXADECIMAL data, programmer control COMMANDS and to select and manipulate the special FUNCTIONS.

The control COMMANDS fall into four groups:

1. loading and editing the contents of the RAM
2. programming and checking the EPROM
3. control of the I/O interface
4. entry to FUNCTION mode.

The following paragraphs describe the action AFTER the appropriate command key has been depressed.

2.4.1 RAM BUFFER COMMANDS

Load
copies the data from an EPROM in the ZIF socket into the internal RAM and shows the resultant checksum of the RAM data in the address display on completion.

List
enables simultaneous display of RAM address, RAM and EPROM data.

The LED above the address display will be illuminated. The keyboard can now be used to enter a hexadecimal value into the address field. The data display will show a comparison of the RAM and EPROM data for the displayed address.

If an error in address entry is made it is only necessary to continue entering on the hexadecimal keys until the display contains the required address.

The lowest usable RAM address is 0000 and the highest is 1FFF. If an attempt is made to enter a RAM address outside this range it will be automatically masked to yield a legal address.

The keys marked ← and → may be used to INCREMENT or DECREMENT the address. Note that these keys have repeat action so that holding down the key will cause continuous address modification.

A new address may be entered at any time using the hexadecimal keys. The RAM data is protected against accidental alteration. The RESET key should be used to leave list mode and return to command mode.

Edit
provides a similar display to list mode BUT alteration of the data in the RAM by use of the keyboard is allowed.

The address is entered initially in the same way as with list mode and the ENTER key is then used to change from entry of address to entry of data values. The LED indication moves across to the data display and the keyboard can now be used to alter the RAM data.

The ← and → keys still operate on the address permitting sequential data entry. Audible warning is given when address or data is modified. To reach a non-sequential address the reset key must be used followed by a new edit sequence.

C'sum
causes calculation and display of the checksum of the RAM data. As an alternative a cyclic redundancy check CRC, can be obtained using function F3 as described in section 2.5.

2.4.2 EPROM COMMANDS

Empty
checks that the EPROM is unprogrammed.

An erased EPROM will cause the message "PASS" in the address display whilst a programmed part will fail and the LED will be illuminated. There will also be an audible warning.

Verify
compares the data in the EPROM with that in the RAM.

The "PASS" message is used to indicate that both sources contain identical data. Any difference will cause a display of the address containing the error and a comparison of the RAM and EPROM data.

The ENTER key can now be used to obtain display of any remaining data errors. If ALL ERRORS have been DISPLAYED the "PASS" message will appear.

Prog
starts the complete programming sequence. This contains an illegal-bit test, the actual programming and a verify operation after programming.

The illegal-bit test does not require that the EPROM is unprogrammed but checks that NO BIT which must be left in the unprogrammed state is already programmed. This permits some modifications to already programmed parts.

If the bit test is successful the program sequence follows automatically and if not then the address which contains the illegal bit together with a comparison of RAM and EPROM data is displayed.

After all addresses have been programmed the VERIFY function is automatically used and its operation is as described above.

Since the program sequence only requires one key-stroke a number of EPROMS may be programmed with minimum effort.

2.4.3 NOTES ON THE PP28 PROGRAMMING ALGORITHMS

Normally programming is sequential from address 0000 and an address count is displayed. At each address the EPROM is compared with the RAM before programming and the pulse omitted if the address contains correct data. After the pulse has been applied the EPROM is again compared with RAM. If this test fails, programming is normally aborted. These measures greatly reduce wasted time if the EPROM is already programmed or faulty.

Some EPROMS have been found to require programming of the whole device before it is possible to read any address and so it is possible to inhibit the comparison of RAM and EPROM after each programming pulse using function FD, see section 2.5.

Some types of EPROM, notably MC68764/6 require a block programming algorithm. The PP28 provides this and displays a FLASHING BLOCK COUNT rather than an address count. After three blocks the EPROM is verified and if this is successful an additional three blocks are given to complete programming. If the device fails to verify, programming will continue in groups of three blocks up to a maximum of 24 blocks.

2.4.4 I/O INTERFACE COMMANDS

The I/O interface is controlled partly by the two keys marked IN and OUT and partly by the use of FUNCTIONS which are only used to modify the default conditions of the UART.

The following default conditions are set up when AC power is applied.

data format	INTELLEC
Baud rate	9600
data word length	8 bits (bit 8 = 0)
number of stop bits	2
parity	not generated, not checked

Details of the use of the functions to modify the above are given in section 2.5.2.

The IN key causes the programmer to accept a serial input from an external source. The source will be controlled by the control lines if these have been connected.

The OUT key causes the programmer to transmit a serial data stream. It may be inhibited by the control lines if necessary.

In both the above cases the default conditions will apply unless the operator has used one of the functions to alter them.

The MDS formats such as INTELLEC contain address information in addition to the data bytes. The addresses are those which will apply to the EPROM when finally in circuit and may be outside the range of the RAM in the PP28.
The PP28 will expect the user to enter an OFFSET ADDRESS terminated with the ENTER key before performing the serial transfer.

On INPUT the OFFSET is SUBTRACTED FROM THE ADDRESSES IN THE SERIAL DATA STREAM AND IF THE RESULTING ADDRESS LIES BETWEEN 0000 and 1FFF the data will be loaded at that address.
On OUTPUT the OFFSET is ADDED to the RAM address.

Note that a serial OUTPUT will always start from address 0000 in the RAM and the last address will be determined by the EPROM size. Function FB may be used to set an upper limit smaller than the EPROM size.

2.5 FUNCTIONS

The special FUNCTIONS are made available by the key marked FN which takes the PP28 from the normal command mode and puts it into the function mode. The functions divide into three groups:

1. manipulation of data in the RAM
2. control of the I/O interface parameters
3. miscellaneous test and control operations.

Depression of the FN key will cause the PP28 to display two dashes in the data display and the user should enter a legal two digit function number. Any key error may be corrected by re-entering the number and the ENTER is finally used to start the selected function.

Some functions require more than the function number to define the action to be taken. This information, called the FUNCTION ARGUMENTS, is entered from the keyboard and each argument is terminated by the ENTER key.

The following paragraphs list FUNCTION NUMBERS, FUNCTION ARGUMENTS and the OPERATION performed by the function.

2.5.1 RAM and MISCELLANEOUS FUNCTIONS (F)

This group of functions is mainly concerned with loading or modifying the data in the RAM.
There are also some miscellaneous functions such as CRC check.
For all this group the left-hand digit of the function number is F.

Function Number	Operation performed	Arguments
F0	fill RAM addresses 0000 up to 1FFF with data 00	none
F2	fill RAM with arbitrary variable from lower to upper address limit	lower limit upper limit data
F3	calculate and display CRC in address display	none
F7	copy data between lower limit 1. and upper limit to another area of RAM defined by lower limit 2.	lower limit 1 upper limit lower limit 2
F8	split packed 16 bit data for programming two 8 bit Eproms	none
F9	shuffle data from two 8 bit blocks into one packed 16 bit block	none
FA	display the software issue number of the PP28 internal EPROM	none
FB	set a reduced address range which must be less than the EPROM size	new range
FC	complement RAM data in addresses 0000 up to 1FFF	none
FD	suppress in-program verification: (a) use for Motorola 2716/2532 (b) effect cannot be cancelled except by power-off	
FF	fill RAM addresses 0000 up to 1FFF with data FF	none

2.5.2 INTERFACE FUNCTIONS (E)

This group of functions are all used to modify the default conditions of the UART. The appropriate function must be performed BEFORE starting the I/O transfer with either the IN or the OUT key. The UART parameters which result from the use of these functions are not affected by the RESET key and will remain until altered or until power off.

The function numbers and the corresponding parameter are given below. For convenience the default conditions are also given.

Function	Parameter	Default value
E0	data format	INTELLEC
E1	Baud rate	9600
E2	data word length	8 bits
E3	number of stop bits	2
E4	parity	not generated or checked

The following tables give the value of the argument needed for each possible condition of the UART.

DATA FORMAT SELECTION

E0	Data Format	Argument Value
	Intellec	00
	Hex-Ascii	01
	Exorcisor	02
	Tek — Hex	03
	PPX	04

BAUD RATE SELECTION

E1	Baud Rate	Argument Value
	9600	01
	7200	02
	4800	03
	3600	04
	2400	05
	1800	06
	1200	07
	600	08
	300	09
	150	0A
	135	0B
	110	0C
	75	0D

DATA WORD LENGTH SELECTION

E2	Data Word Length	Argument Value
	8 bits	00
	7 bits	01
	6 bits	02
	5 bits	03

NUMBER OF STOP BITS

E3	Number of Stop Bits	Argument Value
	1	01
	2	02
	NOTES (a) 1 stop bit if word length is 8 bits and parity (b) 1.5 stop bit if word length is 5 bits and no parity	

PARITY SELECTION

E4	Parity	Argument Value
	Parity disabled: No parity bit generated No parity bit received	00
	ODD parity: Receiver and transmitter	01
	EVEN parity: Receiver and transmitter	03
	Mark parity bit transmitted parity check disabled	05
	Space parity bit transmitted parity check disabled	07

2.6 ERROR CODES

If the PP28 detects an incorrect operation by the user, a failure of the EPROM or programming circuits it will advise the user by lighting the "FAIL" LED and by showing an error code number in the PROM data display.

The following table gives a list of code numbers and describes the error condition.

Error	Fault
00	transmission error or NON ascii—hex data byte received in serial input
01	time out error during serial input i.e. character not received within 300 msec
02	checksum error during serial input
03	invalid EPROM selector switch setting
04	invalid address range
05	reversed or faulty EPROM in ZIF socket
06	Split or Shuffle attempted with a 64k switch setting

2.7 RS 232C PIN CONNECTIONS

The following table gives a description of the signals and connections on the 25 way D-type connector on the rear panel.

Pin	Function
1	Ground . . . connected to pin 7
2	data OUTPUT from the PP28 using RS232C voltage levels
3	data INPUT to the PP28 using RS232C voltage levels
4	RTS: request to send. This line is normally held high but is dropped by the PP28 to inhibit data transmission to the PP28
5	CTS: clear to send. This line must be high to enable transmission by the PP28. An internal resistor does this in the absence of an external connection
6	DSR: data set ready. Joined by an internal link to pin 20
7	Ground . . . connected to pin 1
8	DCD: data carrier detect. This line is held high by an internal resistor
20	DTR: data ready. Joined by an internal link to pin 6

A USERS GUIDE TO THE PP28 SERIAL I/O INTERFACE

STAG

CONTENTS

1.1 I/O INTERFACE COMMANDS

The I/O interface is controlled partly by the two keys marked IN and OUT and partly by the use of FUNCTIONS which are only used to modify the default conditions of the universal asyncronous receiver/tramsmitter (UART)

The following default conditions are set up when AC power is applied.

data format	INTELLEC
Baud rate	9600
data word length	8 bits (bit 8 = 0)
number of stop bits	2
Parity	not generated, not checked

Details of the use of the FUNCTIONS to modify the above are given in the operating manual.

The IN key causes the programmer to accept a serial input from an external source. The source will be controlled by the control lines if these have been connected.

The OUT key causes the programmer to transmit a serial data stream. It may be inhibited by the control lines if necessary.

In both the above cases the default conditions will apply unless the operator has used one of the functions to alter them.

If serial data is being either transmitted or received by the PP28 the I/O LED will be on. If serial transfer either fails to start or is suspended the I/O LED will time-out to indicate this.

The PP28 can accept a continous data stream at 9600 baud and store it in RAM without needing to stop the sending equipment.

1.2 DESCRIPTION OF I/O FORMATS

HEX-ASCII format

This format which is not in common use by micro-development systems has been included for the following reasons:

a. it has a simple structure which makes it useful to a user who is writing software to generate serial output, rather than using an existing software package.

b. it provides a more readable form than the MDS formats when printing EPROM data.

description

1. Start character — a control-A (ASCII code 01) or control-B (ASCII code 02)

2. Data Bytes — Data are sent as hex bytes separated by a space or any other non-hex character.

   Carriage-return is a legal byte terminator.

   The hex data bytes are loaded into into sequential RAM addresses starting at address 0000.

3. Stop character — control-C (ASCII code 03)

example

```
^A
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E
11,22,33,44,55,66,77,88,
99 AA BCBB
^C
```

This example shows:

a. loading 23 data bytes, starting at address 0000 and ending at address 0016

b. the start and stop characters may be separated from the data by carriage-return/line feed if desired.

c. the last data byte on the first line is terminated by the carriage return, but on the second line a comma is optionally used.

d. in the third line the first data byte (BC) is over-written by the BB because the first byte has not been terminated.

MICROPROCESOR DEVELOPMENT SYSTEM (MDS) FORMATS

The MDS formats such as INTELLEC contain address information in addition to the data bytes. The addresses are those which will apply to the EPROM when finally in circuit and may be outside the range of the RAM in the PP28.
The PP28 will expect the user to enter an OFFSET ADDRESS terminated with the ENTER key before performing the serial transfer.

On INPUT the OFFSET is SUBTRACTED from the addresses in the serial data stream and if the resulting address lies between 0000 and 1FFF the data will be loaded at that address.
On OUTPUT the OFFSET is ADDED to the RAM address. This makes output exactly the reverse of input.

Note that a serial OUTPUT will always start from address 0000 in the RAM and the last address will be determined by the EPROM size. Function FB may be used to set an upper limit smaller than the EPROM size.

Intel-Hex (INTELLEC) format

1. Start character — always a colon (:)

2. Byte count — 2 hex digits representing the number of data bytes. Record length of 00 indicates the last record of the file

3. Loading address — 4 hex digits representing the memory address of the first data byte.

4. Record type
00 means a normal data record
01 means the last record of the file

5. Data bytes
the number of hex data bytes is defined by the record length

6. Checksum
2 digit hex number representing the 2's complement (modulo 256) of the preceeding hex bytes

example

```
:02319400923176
:00310001CE
```

This example shows:

a. loading two data bytes at addresses 3194 and 3195

b. the data bytes are 92 and 31

c. 02+31+94+00+92+31=8A
and the 2's complement of 8A is 76

d. an end-of-file record containing the address 3100.

It is usual for an MDS to send the start address of the program code in the end-of-file record.
The PP28 sends the OFFSET address on serial output since the true start address is unknown to it.

EXORCISOR format

1. Start character
always an S

2. Byte count
2 hex digits representing the number of bytes in the record including data, start address and checksum.

3. Loading address
4 hex digits representing the memory address of the first data byte.

4. Record type
0 means an optional header record
1 means a data record
9 means and end-of-file record

5. Data Bytes
the number of hex data bytes is defined by the record length

6. Checksum
2 digit hex number representing the 1's complement (modulo 256) of the preceeding hex bytes

example

```
S0
S10F0040090001000500000E00000F1A6A
S9030000F0
```

This example shows:

a. an optional header record

b. loading 12 data bytes at addresses 0040 to 004B

c. the data bytes are 09,00,01,00,05,00,00,0E,00,00,0F,1A

d. 0F+00+40+09+00+01+00+05+00+00+0E+00+00+0F+1A=95
and the 1's complement of 95 is 6A

e. an end-of-file record containing the address 0000

TEKHEX format

1. Start character	always a /
2. Loading address	4 hex digits representing the memory address of the first data byte.
3. Byte count	2 hex digits representing the number of data bytes in the record
4. First checksum	2 hex digits representing the sum of the 6 digits that make up the loading address and byte count
5. Data Bytes	the number of hex data bytes is defined by the record length (max 30 bytes)
6. Second checksum	2 digit hex number representing the sum (modulo 256) of the digits that make up the data bytes
7. CR	must have carriage return at end of the record

example

```
/0040100500550020202020204D373830304D454967
/00130004
```

This example shows:

a. loading 16 data bytes at addresses 0040 to 004F

b. the data bytes are:
00,55,00,20,20,20,20,20,4D,37,38,30,30,4D,45,49

c. first checksum:

0+0+4+0+1+0=05

second checksum:

0+0+5+5+0+0+2+0+2+0+2+0+2+0+2+0+
4+D+3+7+3+8+3+0+3+0+4+D+4+5+4+9=67

d. a termination record containing the address 0013

1.3 ERROR HANDLING WHEN LOADING PP28 WITH SERIAL DATA

With a typical PP28/MDS RS232C data link spurious data transmission errors are found to be very infrequent.

A more common situation is that the user is aware that his data contains an error but that he still wishes to try to load it into the PP28.

Two examples of ways in which this might occur are, an attempt to "repair" a broken paper tape or an MDS which is generating data which departs slightly from the strict syntax of the I/O format.

In some cases the PP28 will be able to load the data bytes correctly even if it detects an I/O error. For this reason it notes the occurence of the error and continues to try to load.

The user will be warned by an error message when the serial data stream is terminated.

As a example of a non-fatal I/O error consider INTELLEC format which has a 2's complement checksum at the end of each record. If the user accidently sends a 1's complement checksum or merely replaces the checksum with 00, the data bytes will still be loaded correctly. The message "Err 02" will appear when the end record is seen.

The presence of a non-hex character in the data will not abort the load. It will be replaced with 00 and the error message "Err 01" will be given.

Normally the PP28 displays the RAM checksum at the end of an error free data load.

1.4 I/O INTERFACE CONNECTIONS

It is possible to send or receive data with the PP28 serial I/O interface using only a 3-wire link to the 25-way connector. This is sometimes called a "data lines only" interface.

Alternatively the data and control lines on the PP28 are set up in such a way that, in many cases, a cable in which each of the 25 pins on the PP28 is joined directly to the corresponding pin on the serial port of the users equipment (UE) may be used.

Such a connection, referred to as a one-to-one cable, may be made conveniently using insulation displacement connectors/cable (IDC)

Circuit details and pin connections for the I/O interface are given in fig ????

Note that the data are sent from the PP28 on pin 2 and received by the PP28 on pin 3. Hence the PP28 may be regarded as similar to a VDU, printer or paper tape reader/punch for interface purposes.

Most micro-development systems provide at least one port capable of interfacing to a PP28.
This is often designated for a printer or reader/punch.
The one-to-one cable referred to above will normally be suitable in such a case.

The one-to-one cable will be unsuitable for connection of the PP28 to directly to a device such as a VDU or printer as pins 2 and 3 need to be cross-connected.
In this case a 3-wire "data lines only" connection as shown in fig 1.4 below should be used.

```
        PP28                                                 VDU


Data from PP28      2 >----------)))))-----------> 3  Data   to   VDU

Data   to   PP28      3 <----------(((((-----------< 2  Data from VDU

                                      Ground
                       7 ------------------------------ 7
```

1.5 USE OF THE RS232C CONTROL LINES

In order to send or receive serial data the PP28 must have the correct voltage levels on the RS232C control lines. Normally the PP28 generates the correct voltage levels itself and this permits a "data lines only" connection.
The user may use the control lines if desired and this section explains, for each line, how this may be done.

The term "user equipment" (UE) is used to mean any RS232C device to which the PP28 is to be connected.

A voltage between +1.5 and +12V is needed to define a "high" level on the control lines and a voltage between +0.75 and -12V for a "low".

```
*********************************
*                               *
*   DTR                         *
*   ---                         *
*                               *
*   Data Terminal Ready         *
*   Output signal from the PP28 *
*   Pin 20                      *
*                               *
*********************************
```

The PP28 drives this pin high at all times to indicate to the users equipment (UE) that the PP28 transmitter and receiver are both enabled.

Any other pin on the UE which needs to be held high may be connected to pin 20 on the PP28.

```
*********************************
*                               *
*   DSR                         *
*   ---                         *
*                               *
*   Data Set Ready              *
*   Input signal to the PP238   *
*   Pin 6                       *
*                               *
*********************************
```

The state of this pin does not affect either the transmitter or receiver of the PP28. However it should not be left un-connected.

For reasons of compatibility with other PROM programmers

DSR is internally connected to DTR (Pin 20).

If the UE provides an output which drives DSR then either the internal link should be removed or pin 6 on the PP28 be left un-connected.

This is most important if the UE drives pin 6 low as two outputs in opposite states should not be connected.

The best position in which to remove the internal link is shown in fig ???? . The user may find it easier to omit the DSR connection in the cable.

```
*********************************
*                               *
*   RTS                         *
*   ---                         *
*                               *
*   Request To Send             *
*   Output signal from the PP28 *
*   Pin 4                       *
*                               *
*********************************
```

The PP28 drives this output high to indicate to the UE that it wishes to transmit serial data to the UE.

Some MDS monitor their RTS (Pin 4) input and when a high is seen generate a "handshake" output on CTS (Pin 5).

```
*********************************
*                               *
*   CTS                         *
*   ---                         *
*                               *
*   Clear To Send               *
*   Input signal to the PP28    *
*   Pin 5                       *
*                               *
*********************************
```

The CTS input is used to control the PP28 transmitter. The normal state of CTS is high ,which enables the transmitter.

In the absence of an external connection resistor R44 generates a high on this pin.

The user may drive CTS low to hold off the start of data transmission by the PP28. Transmission will start when CTS is driven high.

CTS will be obtained as a "handshake" response to RTS with some MDS, see above.

The UE may drive CTS high permanently if desired.

```
*********************************
*                               *
*    DCD                        *
*    ---                        *
*                               *
*    Data Carrier Detect        *
*    Input signal to the PP28   *
*    Pin 8                      *
*                               *
*********************************
```

The state of DCD does not affect the transmitter of the PP28 but a low on DCD will disable the PP28 receiver.

Normally resitor R45 provides a high on DCD.

2.1 TEKTRONIX 8550 MDS

Interface Connection

The recommended method is to connect the PP28 to the J103 port which is intended for use by a paper tape reader/punch using a cable whose connections are as follows:

```
       PP28                                                TEKTRONIX
                                                             8550
                                                           J103 Port

 Data from PP28      2 >---------->>>>>-----------> 2   Data  to  8550

 Data  to  PP28      3 <----------<<<<<-----------< 3   Data from 8550

      RTS            4 >---------->>>>>-----------> 4       RTS

      CTS            5 <      no connection


      DTR       +-----20 >---------->>>>>-----------> 20      DTR
                I
   internal     I
     link       I
```

```
             I
DSR          +----- 6 <        no connection

DCD                 8 <        no connection

                                    Ground
              7 ------------------------------ 7
```

Note that both ends of the cable will need 25-way connectors with male pins and a female shell.

Although it will introduce unneccesary interconnections it has been found in practice that the PP28 will interface correctly to a TEKTRONIX 8550 if a 25-way one-to-one cable using IDC is used.

TEKTRONIX 8550 Keyboard commands

The general method used to transfer data between the TEKTRONIX 8550 and the PP28 requires the use of the TEKRONIX RHEX and WHEX commands.

Since the PP28 is treated by the TEKTRONIX 8550 as a tape reader/punch the data transfers operate on the PROGRAM MEMORY in the 8550 rather than directly with FILES.

A typical sequence for loading the PP28 with data from a FILE using TEKHEX format is as follows:

TEKTRONIX	PP28	
	FN E0 ENTER 03	select TEKHEX format
>FILL 0000 0FFF FF >LOAD filename		fill memory with FF and load the file
	IN 0123 ENTER	select serial input on PP28
>WHEX PPTP 0000 0FFF		start serial output on 8550

notes:

a. the command sequence to select TEKHEX format need only be performed once and will stay in operation until the PP28 is switched off.

b. the example assumes that the PP28 is programming a 32K EPROM so that the address range is is 0000 to 0FFF.

c. if the file does not fill the whole of the address range then this should be filled with data FF to reduce the time taken to program the EPROM.

d. the OFFSET address (0123) entered on the PP28 is optional. The offset is subtracted from the load address sent in the serial data stream as explained in section 1.1

A typical sequence for loading the 8550 program memory from the PP28 is as follows:

TEKTRONIX	PP28	
RHEX PPTR 0000		select serial input on 8550
	OUT ENTER	start serial output on PP28

notes:
------

a. the PP28 will send data starting at RAM address 0000 and ending on the upper address limit.
b. an offset may be added to the load address as explained in section 1.1

UART Conditions

The following have been found suitable for data transmission from the 8550 to the PP28:

Baud rate	9600
data word length	7 bits
number of stop bits	1
Parity	not generated, not checked

The following have been found suitable for data transmission from the PP28 to the 8550:

Baud rate	9600
data word length	8 bits (bit 8 = 0)
number of stop bits	2
Parity	not generated, not checked

2.2 FUTUREDATA [GenRad]

Interface Connection

The PP28 can be connected to a FUTUREDATA AMDS using the SERIAL 2 port. A simple "data lines" only connection without any cross-connection will permit transfer of data to or from the FUTUREDATA AMDS.

```
      PP28                                              FUTUREDATA AMDS
                                                           Serial 2
                                                             Port

Data from PP28      2 >---------->>>>>----------> 2   Data  to  AMDS

Data  to  PP28      3 <----------<<<<<----------< 3   Data from AMDS


     RTS            4 >      no connection

     CTS            5 <      no connection


     DTR    +-----20 >       no connection
            I
  internal  I
    link    I
            I
      DSR   +----- 6 <       no connection

      DCD          8 <       no connection

                               Ground
                 7 ---------------------------- 7
```

Note that both ends of the cable will need 25-way connectors with male pins and a female shell.

Although it will introduce unneccesary interconnections it has been found in practice that the PP28 will interface correctly to a FUTUREDATA AMDS if a 25-way one-to-one cable using IDC is used.

FUTUREDATA AMDS Keyboard commands

The general method used to transfer data from the FUTUREDATA AMDS to the PP28 requires the use of the system program BINTEL.

The purpose of this is to convert an existing OBJECT file into a temporary file in INTELLEC format. This file is then automatically transmitted through the SERIAL 2 port.

A typical sequence ,assuming an object file with name OBJ on drive 1 and a temporary file on drive 0 already exist, is as follows:

```
FUTUREDATA             PP28
----------             ----


                       FN          fill PP28 RAM
                       FF          with FF
                       ENTER
                       IN          select serial input
                       0123        on PP28
                       ENTER

JB                                 start BINTEL
LBE                                specify: options
1:OBJ                                       object file
TEMP                                        temporary file
```

notes:

a. if the file does not fill the whole of the address range then the PP28 RAM should be filled with FF to reduce the time taken to program the EPROM.

b. the OFFSET address (0123) entered on the PP28 is optional. The offset is subtracted from the load address sent in the serial data stream as explained in section 1.1

UART Conditions

The following have been found suitable for data transmission from the FUTUREDATA AMDS to the PP28:

Baud rate	9600
data word length	8 bits (bit 8 = 0)
number of stop bits	2
Parity	not generated, not checked

2.3 SUPERBRAIN

Interface Connection

The following describes connection to the MAIN PORT of SUPERBRAIN. This is configured in a very similar way to the PP28 as far as the pin connections and signal directions are concerned.

Hence a number of cross-connections are needed in the cable and a 25-way one-to-one cable using IDC will not be suitable.

Note that SUPERBRAIN needs to have its CTS driven high and DTR on the PP28 is used to do this.

```
     PP28                                                    SUPERBRAIN
                                                             Main Port

Data from PP28      2 >---------->>>>>-----------> 3   Data to SUPERBRAIN

Data  to  PP28      3 <----------<<<<<-----------< 2   Data from SUPERBRAIN

     RTS            4 >       no connection

     CTS            5 <       no connection

     DTR     +-----20 >---------->>>>>-----------> 5        CTS
             I
  internal   I
    link     I
             I
     DSR     +----- 6 <       no connection

     DCD            8 <       no connection

                              Ground
                  7 ---------------------------- 7
```

SUPERBRAIN Keyboard commands

The general method used to transfer data between SUPERBRAIN and the PP28 requires the use of the system utility program PIP.

The devices RDR: and PUN: should be specified in the PIP command for input to and output from SUPERBRAIN.

The transfers take place using INTELLEC format.

UART Conditions

The following have been found suitable for data transmission from SUPERBRAIN to the PP28:

Baud rate	9600	
data word length	8 bits	[bit 8 = 0]
number of stop bits	2	
Parity	not generated, not checked	

The following have been found suitable for data transmission from the PP28 to SUPERBRAIN:

Baud rate	1200	
data word length	8 bits	[bit 8 = 0]
number of stop bits	2	
Parity	not generated, not checked	

AM28 Operating Instructions

The AM28 is for use with the Stag PP28 Eprom Programmer. This unit permits 27128 Eproms to be programmed by the PP28, by switching the most significant address line of the 27128 being programmed. In each case, the device is treated as if it were a 2764.

1. Place the AM28 in the 28 Pin ZIF socket (alinging Pin 1's) of the PP28.

2. Set to program 2764 Eproms.

3. Set the AM28 switch to the "LOWER" position to select the least significant 64K data.

4. Load data either via the RS232 of from the Master Eprom. (edit if required).

5. Program data into the 27128.

6. Switch the AM28 to the "UPPER" position and repeat steps 1 to 5 program the most significant 64K of data.