QUBBESoft P/D

QEP III
Eprom Programmer
For the Sinclair QL

User Manual

(Formerly by Care Electronics)

QEP III is a very flexible EPROM programmer to be used with the Sinclair QL computer. The principal interface between QEP III and the user is a combined menu and command system. QEP III is resident and may be invoked from SuperBASIC at any time with the command:-

```
     EPROM or EPROM size
```

> where 'size' is the working area to be used by QEP III. The size is specified in 1k byte units and the default is 64k bytes.

Throughout QEP III, the ESC key can be used to escape from an action. The menu system allows an item to be selected by using the cursor keys to highlight the required item which may then be selected by pressing the space bar or ENTER. Alternatively, an item may be selected by pressing the key corresponding to the first character of the item. E.g., from the main menu, press 'O' to select the 'Options' sub-menu.

Sometimes it is necessary to specify a value (e.g. start address for 'Edit'). In this case a prompt is written out, followed by a default value. To accept the default value press the space bar or ENTER. To respecify the value, just type a new value, or, if desired, move the cursor and use the normal QL editing keystrokes to change the value.

QEP III will program single supply 5V MOS and CMOS EPROMs from 16k bit (2716/2516) to 512k bit (27512). As well as selecting the type of EPROM, the pinout details may also be changed. This allows PCB layouts to be simplified by connecting data and address lines as convenient, rather than adhering strictly to the manufacturer's own numbering conventions. It is, of course, only possible to redefine data lines as data lines, and address lines as address lines.

The EPROM type and your selected pinout can be saved in a file in the QL's filing system, and re-loaded whenever required.

Verification.

QEP III features under and over voltage verification of EPROM contents. This is an essential feature of any programmer which is required to program EPROMs for reliable service.

When clear, all locations in an EPROM appear as 'i's. A bit is programmed to '0' by injecting enough charge into the floating gate of one of the MOS transistors to switch the transistor. Unfortunately, the threshold voltage at which the switching occurs depends on both the chip temperature and the supply voltaqe.

The effect is that an EPROM which appears to be perfectly correct when cold, may be incorrect when warm or vice versa. Furthermore, an EPROM with correct data at 5V may appear imperfectly erased at a lower supply voltage or imperfectly programmed at a higher voltage.

Most integrated power supply regulators maintain the supply voltage within 5%, so even ignoring temperature variations and power supply noise, verification should be carried out at least 5% from nominal voltage.

QEP III provides verification at 7%, 13%.or 20% from nominal voltage. It is recommended that the 20% variation be used when programming, and the 13% variation be used when testing devices.

Programming Algorithms

QEP III has a choice of algorithms to suit different devices and different requirements. The standard algorithm uses a fixed 50 ms pulse to program each location and is the manufacturer's recommended method for earlier (16k and 32k bit) EPROMs. More recent (64k, 128k and 256k) EPROMs are normally programmed using an interactive algorithm and 1 ms programming pulses. To ensure complete programming, the power supply is held at 6V during the algorithm and each location is overprogrammed. The latest devices have a similar algorithm, but with pulses of down to .1ms and much reduced overproqramming. Manufacturers of EPROMs do not normally recommend verification of EPROMs at low and high supply voltages despite this being normal practice on all good EPROM programmers and also being essential for providing confidence in the reliability of the programmed data in service.

Full, Part, and Overprogramming

QEP III can program all locations in an EPROM, or just a selected range. QEP III does not attempt to program any locations where the data is required to be (hex)FF, although these locations are checked before programming to ensure that they are indeed unset. If a range is specified, then only that part of the EPROM will be checked before programming.

In normal programming, every location is checked to ensure that it is empty before the programming starts. For overprogramming, every location is checked to ensure that there are no programmed bits ('O's) where '1's are required in the final data. When overprogramming, each location is reprogrammed at least once, unless the data is required to be (hex)FF.

Standard Algorithm

In the standard algorithm, each location is checked at low and high supply voltage, each location is programmed (where appropriate) with a 50ms pulse, then each location is verified at low and high supply voltage.

This algorithm corresponds to the normal programming of 2716s (2516 TI), 2532s, 2732s, 2564s and 2764s (TI type only). It should only be used where there is concern that applying a 6V supply voltage during programming could damage the device.

Interactive Algorithm I

QEP III supports two variations of interactive programming which differ according to method used to boost the programmed data. In all other respects they are identical.

In interactive algorithm I, each location is checked at low and high supply voltage. The supply voltage is raised to 6V and then, at each location, 1ms programming pulses are applied until the contents verify correctly and a single programming pulse is applied to boost the data. This boost programming pulse is four times as long as the total of programming pulses already applied to that location. No more than 15 programming pulses are applied to each location, so that the maximum length of the boost pulse is 60ms. When all locations have been programmed, the contents are verified at low and high supply voltage.

This algorithm is specified for most 2764s, for 27128s and for some 27256s.

(Manufacturers specifications are liable to change.)

Interactive Algorithm II

In interactive algorithm II, each location is checked at low and high supply voltage. The supply voltage is raised to 6V and then, at each location, 1ms programming pulses are applied until the contents verify correctly. When all locations have been programmed, the supply voltage is dropped to 5V and a single 2ms boost programming pulse is applied to each location. The contents are then verified at low and high supply voltage.

This algorithm is specified for some 27256s and for 27512s.

Accelerated Algorithm

The accelerated algorithm of QEP III is designed to take advantage of its programmability to provide the fastest possible programming consistent with reliability at least as good as the manufacturer's recommended algorithms. The method is similar to the interactive algorithm II, but there is no boost programming. Instead, the contents are verified at low and high voltage, and if any locations fail verification, they are overprogrammed. The programming pulse width is reduced to .4ms.

Menus

QEP III is fully menu driven, the following items may be selected from the main menu.

Options: For selecting device type, programming range, programming algorithm etc.

Fill: Fills the memory with zero bytes or $FF bytes.

Load: Loads data into the memory, or loads an options file.

Save: Saves data from the memory, or creates an options file.

Edit: Starts the editor to examine or change the memory contents.

Read: Reads the contents of an EPROM into the memory.

Check: Checks that an EPROM is blank, or that it may be over-programmed.

Program: Programs an EPROM using the options as set up.

Verify: Verifies the contents of an EPROM against the memory contents.

QUIT: Leaves QEP III, the memory contents will be lost.

The 'Options' sub-menu is used to set up all the many options which QEP III provides. Once set up, the options may be 'Saved' for future use.

Device: Selects a device type, this also presets the pinout etc.

Jumble: Jumbles the data on address lines.

Variation of Vcc: Sets the variation of the EPROM supply voltage to be 0%,7%,13% or 20%.

Programming Vpp: Sets the programming voltage to be 12.7, 21 or 25 volts.

Algorithm: Selects the programming algorithm.

Over Programming: Turns the over-programming flag on or off. If over-programming is on, then QEP III will reprogram a non-blank EPROM if this is possible

Set of EPROMS: Defines the size of a set of EPROMs to be programmed. E.g. if you wish to use the EPROMs on a 32 bit data bus, then you will need 4 EPROMs per set: QEP III will program one out of each group of four bytes in the memory into each EPROM in the set.

Range in EPROM: Sets the range of addresses which will be read, checked programmed or verified. This allows EPROMs to be part programmed.

EPROM base address: Sets the base address of the EPROM and adjusts the range appropriately. QEP III does not assume that an EPROM is resident at address zero.

Memory bas address: Sets the base address of the memory.

Beeping: Turns the BEEP (on completion of a command) on or off.

Test: Provides test voltages for setting up the internal power supplies.

Appendix 1 MOS EPROM Pinouts

Pin

Pin	No	2716	2732	2532	2764	2564	27128	27256	27512
		(2516)							
(28)	(24)								
	1				Vpp	Vpp	Vpp	Vpp	A15
	2				A12	OE	A12	A12	A12
3	1	A7	A7	A7	A7	A7	A7	A7	A7
4	2	A6	A6	A6	A6	A6	A6	A6	A6
5	3	A5	A5	A5	A5	A5	A5	A5	A5
6	4	A4	A4	A4	A4	A4	A4	A4	A4
7	5	A3	A3	A3	A3	A3	A3	A3	A3
8	6	A2	A2	A2	A2	A2	A2	A2	A2
9	7	A1	A1	A1	A1	A1	A1	A1	A1
10	8	A0	A0	A0	A0	A0	A0	A0	A0
11	9	D0	D0	D0	D0	D0	D0	D0	D0
12	10	D1	D1	D1	D1	D1	D1	D1	D1
13	11	D2	D2	D2	D2	D2	D2	D2	D2
14	12	gnd	gnd	gnd	gnd	gnd	gnd	gnd	gnd
15	13	D3	D3	D3	D3	D3	D3	D3	D3
16	14	D4	D4	D4	D4	D4	D4	D4	D4
17	15	D5	D5	D5	D5	D5	D5	D5	D5
18	16	D6	D6	D6	D6	D6	D6	D6	D6
19	17	D7	D7	D7	D7	D7	D7	D7	D7
20	18	CE	CE	A11	CE	A11	CE	CE	CE
21	19	A10	A10	A10	A10	A10	A10	A10	A10
22	20	OE	OE	CE	OE	CE	OE	OE	OE
23	21	Vpp	A11	Vpp	A11	A12	A11	A11	A11
24	22	A9	A9	A9	A9	A9	A9	A9	A9
25	23	A8	A8	A8	A8	A8	A8	A8	A8
26	24	Vcc	Vcc	Vcc	NC	Vcc	A13	A13	A13
27					PGM	OE	PGM	A14	A14
28					Vcc	Vcc	Vcc	Vcc	Vcc

Key:	**Vcc**	power supply (5V)
	gnd	reference (0V)
	Vpp	programming voltage (5V read, 12-25V program)
	CE	chip enable (power down)
	OE	output enable (usually faster than CE)
	PGM	program enable

Appendix II Protocols

All protocols can be aborted by pressing the ESC key.

Standard Checksum

QEP III uses the same simple checksum calculation as most other EPROM programmers. The checksum is calculated as the sum of all the byte values, modulo 65536. This method unfortunately gives different results when checksumming partially programmed EPROMs and checksumming the data due to the unprogrammed locations of the EPROM being (hex)FF rather than zero.

BINARY CS Compatible with GP1000

This format sends data as 8 bit bytes.

This format is based on 8080 data conventions where the least significant byte of a multi byte value is transmitted first. Up to 64k bytes may be sent (OK for up to 1 27512, 2 27256 etc.).

The data is preceded by 4 bytes: a 2 byte data length, then a 2 byte standard checksum.

BINARY QEP

This format sends data as 8 bit bytes.

This format is based on 68000 data conventions where the most significant byte of a multi byte value is transmitted first.

The data is sent in any number of blocks. Each block is preceded by 10 bytes: the length of the data block itself (long word), the start address of the data block (long word) and the standard checksum of the data block. The last block is followed by a long word with a negative value (e.g. -1 (hex)FFFFFFFF).

HEX Compatible with Stag PP28

This format sends data as 7 bit ASCII characters.

This format does not allow the length of the data to be specified. When reading, all characters are ignored until the first hex digit is encountered. When writing, the first character sent is a wakeup CTRL A (value 1). The data follows as a stream of hexadecimal coded bytes (two hex digits) separated by spaces or <CR> or <LF>. <CR> and <LF> are used together to end a line when writing, excess separators are ignored when reading. The maximum line length is limited to 80 characters when writing. There is no limit to the line length on reading. To terminate a read action press the ESC key.

HEX QEP

This format sends data as 7 bit ASCII characters.

The data is sent in any number of blocks. Each block is preceded by 3 values. The length of the data block itself, the start address of the data block, and the standard checksum of the data block. The last block is followed by a negative long word value (e.g. FFFFFFFF).

The headers and the data are sent as a stream of hexadecimal coded values separated by spaces or <CR> or <LF>. The data is sent as bytes. <CR> and <LF> are used together to end a line when writing, excess separators are ignored when reading. The maximum line length is limited to 80 characters when writing. There is no limit to the line length on reading.

HEX Motorola S Record

This format sends data as records of 7 bit ASCII characters. There are no separators within a record.

Each record starts with the letter 'S'. The record 'S9' is used to terminate the data. The only other record type processed by QEP III is the 'S1' record.

This has the format 'S1' followed by two hex digits giving the number of bytes to follow in the record (half the number of characters), followed by four hex digits specifying the start address for the data in the record (high byte / low byte). The data follows and is followed by a checksum byte. This is the ones complement of the sum of all the bytes (not the characters!) from after 'S1' up to the checksum itself. On input a record may be up to 253 bytes long, and any characters before 'S' and after the end of record are ignored. On output there is a maximum of 16 bytes in a record, and each record is followed by <CR><LF>.

Appendix III - Programming method

```
set read state with Vcc high
check contents clear or reprogrammable
set read state with Vcc low
check contents clear or reprogrammable

for up to maximum program attempts
  set state 1 (prepare for programming)
  set Vcc to Vcc+
  set state 2 (programming idle)

  for each location
    for maximum program pulses
      set address at state 3 (preprogram) put data on data lines
      set state 4 (program)
      wait for program pulse width
      remove data from data linesPVER
      set state 5 (post program)
      set state 6 (programming idle)
      set state 7 (program verify)
      verify data
      set state 8 (programming idle)
      if excess '1's: next program pulse
      if excess 'O's: device failed
    end program pulses

    boost=number of program pulses * boost ratio
    if boost
      do PVER with pulse width=boost
      if not verified: device failed
    end boost
  end of locations

  set Vcc to 5V
  if fixed boost
    for all locations
      do PVER with pulse width
      if not verified: device failed
    end of locations
  end fixed boost

  set state 9 (end of programming)
  set read state with Vcc low
  verify all locations
  if excess 'O's: device failed
  set read state with Vcc high
  verify all locations
  if excess '1's
    double width and halve max pulses (alternate attempts)
    next program attempt (only locations where verify failed)
  end of attempts
  if not all verified: programming failed
```

Algorithm constants

	Standard	**Int I**	**Int II**	**Accel**
Maximum attempts	1	1	1	8
Maximum pulses	1	15	25	64
Pulse length (x0.1ms)	500	10	10	4
Boost ratio (x0.1)	0	40	0	0
Fixed boost (x0.1ms)	0	0	20	0

Read conditions

Pin numbers are referred to 28 pin device.

	2716	**2732**	**2532**	**2764**	**2564**	**27128**	**27256**	**27512**
Vcc pin	26	26	24	28	28	28	28	28
Vpp pin (=Vcc)	21		21	1	1	1	1	
Chip Enables	20	20		20	2	20	20	20
(low)	22	22	22	22	22	22	22	22

Programming conditions

2716	**23**	**20**	**22**
State 1-9	Vcc	low	high
State 2-6-8	Vpp	low	high
State 3-5	Vpp	low	high
State 4	Vpp	high	high
State 3-5	Vpp	low	high
State 2-6-8	Vpp	low	high
State 7	Vpp	low	low
State 2-6-8	Vpp	low	high
State 1-9	Vcc	low	high

2732	**20**	**22**
State 1-9	high	high
State 2-6-8	high	high
State 3-5	high	Vpp
State 4	low	Vpp
State 3-5	high	Vpp
State 2-6-8	high	high
State 7	low	low
State 2-6-8	high	high
State 1-9		

2532	**23**	**22**
State 1-9	Vcc	high
State 2-6-8	Vcc	high
State 3-5	Vpp	high
State 4	Vpp	low
State 3-5	vpp	high
State 2-6-8	Vcc	high
State 7	Vcc	low
State 2-6-8	Vcc	high
State 1-9	Vcc	high

2764	**1**	**20**	**22**	**27**
State 1-9	Vcc	high	high	high
State 2-6-8	Vpp	low	high	high
State 3-5	Vpp	low	high	high
State 4	Vpp	low	high	low
State 3-5	Vpp	low	high	high
State 2-6-8	Vpp	low	high	high
State 7	Vpp	low	low	high
State 2-6-8	Vpp	low	high	high
State 1-9	Vcc	high	high	high

2564	**1**	**22**	**2**	**27**
State 1-9	Vcc	high	low	low
State 2-6-8	Vcc	high	low	low
State 3-5	Vpp	high	low	low
State 4	Vpp	low	low	low
State 3-5	Vpp	high	low	low
State 2-6-8	Vcc	high	low	low
State 7	Vcc	low	low	low
State 2-6-8	Vcc	high	low	low
State 1-9	Vcc	high	low	low

27128	**1**	**20**	**22**	**27**
State 1-9	Vcc	high	high	high
State 2-6-8	Vpp	low	high	high
State 3-5	Vpp	low	high	high
State 4	Vpp	low	high	low
State 3-5	Vpp	low	high	high
State 2-6-8	Vpp	low	high	high
State 7	Vpp	low	low	high
State 2-6-8	Vpp	low	high	high
State 1-9	Vcc	high	high	high

27256 NEC	**1**	**20**	**22**
State 1-9	Vcc	high	high
State 2-6-8	Vpp	high	low
State 3-5	Vpp	high	high
State 4	Vpp	low	high
State 3-5	Vpp	high	high
State 2-6-8	Vpp	high	low
State 7	Vpp	low	low
State 2-6-8	Vpp	high	low
State 1-9	Vcc	high	high

27256	**1**	**20**	**22**
State 1-9	Vcc	high	high
State 2-6-8	Vpp	high	high
State 3-5	Vpp	high	high
State 4	Vpp	low	high
State 3-5	Vpp	high	high
State 2-6-8	Vpp	high	high
State 7	Vpp	high	low
State 2-6-8	Vpp	high	high
State 1-9	Vcc	high	high

27512	**20**	**22**
State 1-9	high	high
State 2-6-8	high	high
State 3-5	high	Vpp
State 4	low	Vpp
State 3-5	high	Vpp
State 2-6-8	high	high
State 7	low	low
State 2-6-8	high	high
State 1-9	high	high