MIRACLE SYSTEMS

TRUMP CARD AND TOOLKIT II MANUAL

CONTENTS

Introduction

Thank you for buying the Miracle Systems Disk Interface (TRUMP CARD), the disk interface comes in a variety of forms which are described below.

Versions

TRUMP CARD 768K
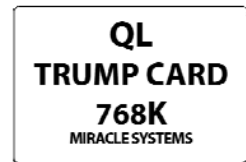
When the TRUMP CARD 768K is plugged into the QL an extra 768K RAM is added to the QL's own memory to give 896K RAM total plus all the extra facilities covered in this manual.

TRUMP CARD 512K
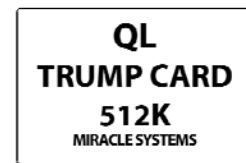
When the TRUMP CARD 512K is plugged into the QL an extra 512K RAM is added to the QL's own memory to give 640K RAM total plus all the extra facilities covered in this manual.

TRUMP CARD 256K / ACE CARD
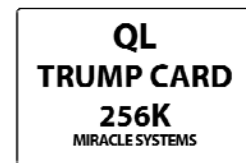
When the TRUMP CARD 256K is plugged into the QL an extra 256K RAM is added to the QL's own memory to give 384K RAM total plus all the extra facilities covered in this manual.

DISK CARD / TRUMP CARD 0K
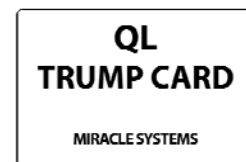
When the DISK CARD is plugged into the QL no memory is added to the QL's own memory. All other features of the TRUMP CARD are included and covered in this manual.

To conserve memory, the extended facilities are not available if there is only 128K of QL RAM. However, an additional command FLP_EXT is included to enable the Printer Buffer, the RAM disk, the Screen Dump and the TK2_EXT command.

TREY CARD

The TREY CARD must be used with the MIRACLE SYSTEMS EXPANDERAM 512K. When the TREY CARD is plugged into the end of the EXPANDERAM 512K an extra 256K RAM is added to the QL's expanded memory to give 896K RAM total. The combination of TREY CARD and EXPANDERAM 512K behave exactly the same as the TRUMP CARD 768K.

Beginners Guide

Always switch off the QL before installing or removing the TRUMP CARD.

To install the TRUMP CARD remove the door on the left-hand-end of the QL and then carefully plug the TRUMP CARD into the exposed Expansion Port.

The extra commands offered by Toolkit II are enabled by entering TK2_EXT after pressing F1 or F2 after switch-on.

A printer buffer for the serial ports is created by entering PRT_USE SER, SER.

To share one printer between 2 or more networked QL's firstly enable Toolkit II by entering TK2_EXT, set up the QL's station numbers by entering NET 1 on 1 QL, NET 2 on the next and so on, enter FSERVE on each QL, then on each remote QL use the device name N1_SER1 assuming the printer is attached to the first QL. For example, after entering FSERVE, put QUILL in MDV1 on QL 2 and enter LRUN MDV1_BOOT. To print, instead of using the default "printer" option, type in _N1_SER1. The document will be printed via SER1 on QL 1.

For disk sharing the QL's must be set up in a similar manner to printer sharing. If the disk is attached to QL 3 then the file "fred" would be loaded into QL 2 by entering on QL 2 LOAD N3_FLP1_FRED.

To run programs unable to run on expanded QL's like Psion Chess enter RES_128 after F1 or F2 after switch on. The QL will now look as though it is unexpanded to the software.

TO USE QUILL FROM DISK INSTEAD OF MICRODRIVE.

If you only have a single disk drive, then use the CONFIG_BAS routine on your QUILL cartridge so that QUILL will expect SYSTEM, DATA and HELP information from MDV1_.

1. Reset the QL.
2. Press F1 or F2.
3. Type TK2_EXT then press ENTER.
4. Put a blank disk into FLP1.
5. Type FORMAT FLP1_QUILL then press ENTER.
6. When the disk has formatted put your QUILL cartridge into MDV1.
7. Type WCOPY MDV1_ TO FLP1_ then press ENTER.
8. A message saying MDV1_nnn TO FLP1_nnn..Y/N/A/Q? will appear. (Where nnn can be any name eg. BOOT, CLONE, etc.)
9. Press the letter A to copy all the files.
10. When the copying has finished type LOAD FLP1_BOOT then press ENTER.
11. Once the file has loaded type RENUM then press ENTER.
12. Type 90 FLP_USE MDV then press ENTER.

13. Type SAVE FLP1_BOOT then press ENTER.
14. A message saying FLP1_BOOT EXISTS, OK TO OVERWRITE, Y OR N? will appear.
15. Press the letter Y to overwrite the file.
16. When the light on the disk drive goes out, the disk is ready to use.
17. After RESET, once the disk has loaded in, any reference to MDV will access the disk instead of the Microdrive.
18. Use the same procedure to copy your other PSION programs to disk.
19. You can also use the WCOPY command to copy data from the Microdrive cartridge to disk.

DISCLAIMER

In no circumstances will Miracle Systems, Care Electronics, or QJump be liable for any direct, indirect or consequential damage or loss including but not limited to loss of use, stored data, profit or contracts which may arise from any error, defect or failure of the TRUMP CARD software.

TRUMP Card from Miracle Systems

The TRUMP card provides up to 768K of RAM in addition to a floppy disk controller. To match this extremely powerful configuration, the TRUMP card comes with an equally powerful set of system firmware. This includes floppy disk and RAM disk drivers, a dynamic printer buffer, a versatile screen dump utility and the invaluable SuperToolkit II. Just in case you have some software which is written assuming that you have an unexpanded QL, there is also a command to make all our hard work disappear. This command is "RES_128".

The Floppy Disk Driver and RAM disk

The QL computer is delivered with two "mass storage" devices: the Microdrives. These devices have the same function as the floppy disks on more expensive personal computers, being designed for the permanent storage of programs and data. Other devices which behave in the same way as Microdrives (such as floppy or hard disks) may be added to the QL "transparently". This means that QDOS will ensure that a program does not need to "know" where its data is stored. A Microdrive looks, to a program, exactly the same as a floppy disk. This "device independence" is a built-in characteristic of the QDOS operating system.
 The simplest way of using a floppy disk system on the QL is to copy all programs and data to floppy disks, and either add the emulation command "FLP_USE MDV" to all BOOT files, or type this command at the start of a session on the QL. The effect of this command is to make the floppy disks pretend to be rather large and fast Microdrives.

For example, a modified BOOT file for executing the PSION program Quill could look like:

```
 90 FLP_USE mdv:          REMark - emulate microdrives
100 CLOSE #1: CLOSE #2
110 EXEC_W mdv1_quill
....
....
```

On the other hand, it is just as easy to use the floppy disks without changing the name. All the filing system commands described in the "Microdrives" section of the QL Concept Reference Guide will work with floppy disks, provided the filenames start with "FLP" instead of "MDV":

FORMAT flp1_test   formats a new floppy disk in drive 1

DIR flp1_   directory listing of floppy disk 1

SAVE flp1_myprog   save the current SuperBASIC program
                        as "myprog" in floppy disk 1

OPEN_NEW #3,flp2_data	creates and opens a new file "data" in floppy disk 2
COPY mdv1_x TO flp1_x	copies file x from Microdrive 1 to floppy disk 1

The term "RAM disk" is a misnomer. It is used to denote a "virtual" device (one that one only exists in the fertile imagination of the QL) that looks and behaves like a very fast disk device. It is so fast because being virtual, there is virtually nothing to move to get information in and out. It is, in fact, no more than a reserved area of the QL's main memory (its RAM - Random Access Memory). This means, of course, that any space taken by a RAM disk is not available to programs executing in the QL. Furthermore, any data stored in a RAM disk will be lost when the QL is turned off or reset!

RAM disks in the QL may be of any size, subject to there being enough memory. The normal usage of a RAM disk would be to copy all working files from Microdrive (or floppy disk) into a RAM disk; rename the RAM device to be MDV (to pretend that the data is really on the Microdrives); execute the programs (e.g. Quill, Archive etc.); and, at the end of the session, rename the RAM device to be RAM before copying the data files back to Microdrive.

On the other hand, it is just as easy to use a RAM disk without changing the name. All the filing system commands described in the "Microdrives" section of the QL Concept Reference Guide will work with RAM disks, provided the filenames start with "RAM" instead of "MDV".

FORMAT ram2_200	creates a new RAM disk 2, see below
DIR ram1_	directory listing of RAM disk 1
SAVE ram1_myprog	save the current SuperBASIC program as "myprog" in RAM disk 1
OPEN_NEW #3,ram2_data	creates and opens a new file "data" in RAM disk 2
COPY mdv1_x TO ram1_x	copies file x from MDV1 to RAM Disk 1

RAM Disk Creation

A dynamic RAM Disk is created just by accessing it with any normal operation (e.g. DIR). This type of RAM Disk takes memory as required, and releases any memory as files are deleted or truncated.

A fixed RAM disk is created by formatting it: the size, in sectors, is given in place of the usual medium name. This pre-allocates all the space that will be available in the RAM disk.

    FORMAT ram2_80

removes the old RAM disk number 2, and sets up a new RAM disk of 80 sectors. A RAM disk may be removed by giving either a null name or zero sectors

    FORMAT ram1_   or   FORMAT ram1_0

The RAM disk number should be between 1 and eight, inclusive, while the number of sectors (512 bytes) is only limited by the memory available. A RAM disk can be formatted from the FILES menu of QRAM.

Heap Fragmentation

The primary storage mechanism in the QL for permanent or semi- permanent memory allocations is a "heap". Allocating space in a heap, and then re-allocating this space as a different size, inevitably causes holes to be left within the heap. This reduces the amount of memory available to either SuperBASIC or executable programs.

This RAM disk driver has precautions to reduce the possibility of heap fragmentation, but it is preferable to consider any fixed RAM disk to be a permanent feature until the QL is reset.

Using a fixed RAM Disk not only reduces the danger of heap fragmentation, but also provides higher access speeds during file creation. Since it always occupies the maximum space you ever wish to use, it is much less flexible.

Microdrive Imaging

Microdrive imaging is a very fast method of loading files from a Microdrive cartridge. To produce a Microdrive image, a RAM Disk is formatted with the name of the Microdrive required:

FORMAT ram1_mdv2   loads an image of mdv2 into RAM Disk 1

The RAM Disk can even load a Microdrive with a damaged directory. It cannot, however, load a Microdrive with a damaged map.

The RAM Disk will try up to 3 times to read a faulty sector. If it fails, the number of good sectors returned from the format will be fewer than the total number. Any file with bad sectors will be marked with an "*" in the RAM Disk directory.

Floppy and RAM Disk Compatibility

The QJUMP Floppy Disk and RAM Disk drivers not only provide all the built-in Microdrive filing system operations, but include the extended filing system operations provided in the Sinclair QL Toolkit and QJUMP Super Toolkit II for Microdrives. This allows all the SuperBASIC extensions provided in the Toolkits (e.g. FOP_OVER, RENAME etc.) to be used with the floppy disks

OPEN OVERWRITE   Trap #2, D0=1, D3=3

> This variant of the OPEN call opens a file for   write/read whether it exists or not. The file   is truncated to zero length before use.

RENAME   Trap #3, D0=4A, A1 points to new name

> This call renames a file. The name should include the drive name (e.g. FLP1_NEW_NAME).

TRUNCATE   Trap #3, D0=4B

> This call truncates a file to the current byte position.

In addition the FS.FLUSH call for a file, not only flushes all the file buffers, but, unlike the Microdrive driver, updates the map and the directory. This means that a new file can be created, and if it is flushed, then in the event of the QL being turned off or reset before the file is closed, then all of the file (up to the point where it was last flushed), is readable. In effect a FLUSH call is just the same as a CLOSE call, except that the file remains open and the file pointer remains unchanged.

Auto-boot

If there is a disk in drive 1 when the QL is turned on (this may be risky with some makes of floppy disk drive, particularly those with permanently loaded heads) or reset (this should be safe with all drives), then the QL will boot from the disk in drive 1, otherwise the QL will boot from Microdrive 1 as usual. There is no direct control over the disk drive motor as the motor is turned off by the hardware in the interface after 10 disk rotations. To stop the motor, insert a disk into drive 1.

When a "directory device", such as a floppy disk, is accessed for the first time, QDOS will allocate a block of memory for the device. In the case of a floppy disk, the Sinclair standard format requires a block of memory about 1.6 kilobytes long. This is rather larger than the Microdrive block which is only about 0.6 kilobytes long. The auto-boot procedure used ensures that if there is no disk in drive 1 when the QL is reset, then the 1.6 kilobyte block for disk drive 1 will not be allocated. Programs that are too large to execute when floppy disks are being used, should still execute from microdrives.

Microdrive Emulation

The standard drivers also include the SuperBASIC procedures FLP_USE and RAM_USE to change the name of the floppy disk and RAM disk drivers.

   FLP_USE mdv   or   FLP_USE 'mdv'

will reset the name of the floppy disk driver to "mdv", so that all subsequent open calls for Microdrives will use the floppy disks instead. While

   RAM_USE mdv   or   RAM_USE 'mdv'

will do the same for the RAM disks.

For example

    FLP_USE mdv
    ....
    ....
    OPEN #3,mdv1_myfile

will actually open the file "myfile" on floppy disk 1, rather than trying to open a file on Microdrive 1.

Any three letters may be used as a new device name, in particular
   FLP_USE flp   and   RAM_USE ram

will reset the drivers to their normal state.

Floppy Disk Options

There are three parameters of the floppy disk system which are available as user options.
   The security level is selectable to allow a user to choose   higher speed of access at the cost of reduced immunity to   erroneous disk swapping. There are three levels of security,   the lowest level still being at least as secure as common   disk based operating systems (e.g. MSDOS and CPM).

   A user may specify the time taken for the disk drive motor   to get the disk speed to within the specification.

   A user may specify the number of tracks to be formatted on   a disk.

The parameters are specified by three commands:

   FLP_SEC   security level

   FLP_START   start up time (in 50ths of a second)

   FLP_TRACK   number of tracks

Security

The Microdrive filing system is unusual in that, although the data is stored in "sectors" in just the same way as on a floppy disk, each sector holds information which identifies the cartridge. When a cartridge is changed the filing system will recognise the change the next time any access is made to Microdrive. Standard floppy disk formats do not allow this type of security, so the format used for QL floppy disks includes identifying information in Track 0 Sector 1 of the disk. Clearly if this were checked every time any access were made to the disk, then the floppy disk system would be very slow indeed. Security, in the context of this user option, is the extent to which the floppy disk system may be abused by changing disks, while they are in use, without destroying data stored on the disks.

There are four operations which affect the security: the first is the operation to check if the disk has been changed, the second is the operation to flush the slave blocks, the third is the operation to update the map and the fourth is the operation to update the directory.

In these definitions, the term "the drive has stopped" is usually taken to mean that the motors have stopped and no drive select light is visible.

Security Level 0

The disk is only checked when a file is opened and the drive has stopped since the last time it was checked and there are no files already open on the drive. The map is only updated after a file is closed (or flushed) when half a second has elapsed without any other disk operation.

At this lowest level of security, confusion or loss of data can be expected if a disk is changed while there are still files open or the motor is running.

Security Level 1

The disk is checked when a file is opened, or data or the map is to be written, and the drive has stopped since the last time it was checked. The map is only updated after a file is closed (or flushed) when half a second has elapsed since the previous disk operation.

At this level of security, disks should only be changed while the motor is stopped (all select lights off). If a disk is changed while there are files open, then read operations will be confused but any write operations will be aborted. This should maintain the integrity of the data on the disk.

Security Level 2

The disk is checked whenever a file is opened or whenever the map or data is to be read from or written to the disk and the drive has stopped since the last time the disk was checked. The map and directory are updated and the buffers are flushed immediately after a file is closed, or after an FS.FLUSH call.

This is the default security level and data should be quite secure unless a disk is changed while the motors are running.

Security System Errors

There are two error messages which may be written to the screen by the floppy disk filing system. These are in the form of the disk name followed by the message itself. The first message indicates that an attempt to read or write a sector on the disk has failed:

   disk name   read/write failed

The second message indicates that a disk has been changed while it is still in use:

   disk name   files still open

If the floppy disk system attempts to write to a disk which has been changed, then you may get both messages indicating that the attempt to write the data has been aborted, and that files were still open when the disk was changed.

Start Up Time

The floppy disk system will always try to read data from a disk as soon as it can. However, to preserve the data integrity of the disk, write operations are held up until the disk has been "run up" for long enough for the speed to be stable. As a default this is set to .6 second which is more than enough for most modern drives. The start_up_time parameter is in 20 millisecond units, so the default value is 30. A value of 13 (260 milliseconds) is adequate for the most recent direct drive 3.5 inch drives, while some older drives may require a value of about 60 (1.2 seconds).

Number Of Tracks

The QL format for disks allows the number of tracks on a disk to be read from the disk itself. However, the number of tracks must be determined when a disk is to be formatted. Normally the disk system will do this itself by checking if there are at least 55 tracks on a disk. If there are, then there are assumed to be 80 tracks, otherwise it is assumed that there are 40 tracks. This internal check may be overwritten, allowing 37 track and 75 track drives to be formatted as well as saving possible wear or damage to a 40 track drive when seeking track 55 (somewhere in the middle of the jacket).

Direct Sector Read/Write

The software includes provision for reading sectors of a disk using direct addressing. To do this a special file is opened on the disk. The name is:

FLPn_*Dsd   where s is the sector length   0=128 bytes
                                         1=256 bytes
                                         2=512 bytes
                                         3=1024 bytes

        and d is the density   D=double (MFM)

When opening a disk for direct sector read/write from SuperBASIC, the name should be enclosed in quotes (or apostrophes).

    OPEN #3,'flp1_*d2d'

When this file is open, no other file may be open on the drive. The only IO calls supported for this type of file are IO.FSTRG, IO.SSTRG, IO.POSAB and IO.POSRE, to read or write complete sectors or to set the position. The parameter (D1) to the POSRE call is ignored, but the current position is returned. Reading or writing a sector does not change the file position.

The position is a composite of the required sector, side and track:

    sector number + side * 256 + track * 65536

To ensure compatibility with string I/O the length specified in the SSTRG and FSTRG calls may be one of three values:

       sector length the complete sector is read or written

    2   returns the sector length (IO.FSTRG)
                  ignored (IO.SSTRG)

    2 + sector length   returns the sector length followed by the
                  sector (IO.FSTRG)
                  skips the first two bytes, and writes the
                  rest to the sector (IO.SSTRG)

This variety enables sectors to be read and written in SuperBASIC using the normal string I/O in the Super Toolkit II, as well as by assembler programs. For example, sector 1 of side 1 on track 2 may be read into the string A$ using the following command:

    GET #n\1+256+2*65536, a$

When using the direct sector read/write calls for a 40 track disk in an 80 track drive, the track number should be doubled. Seek errors will not be detected. If a read/write error is returned from a direct sector read/write call, then it will be safest to make another call to read from

track zero. Calls to read from or write to track zero will cause a "restore" rather than a seek, and will thus reset the drive to a known state.

Disk Drive Specifications

It is a requirement that disk drives used with this version of the disk driver should be set to have the motor on when provided with a "motor on" signal and there is a disk in the drive. Drives which turn the motor off when the drive is not selected will not give reliable service. The disk driver will automatically adjust itself to use any mixture of disk drives, 40 or 80 track, single or double sided. In addition it will adjust itself to use slow step rate drives. Disks need not have been formatted and written on the same specification drive as a drive being used to read them.

```
Disk format ->  40T SS  40T DS  80T SS  80T DS
  Drive
  40T SS        OK    ?     X     X
  40T DS        OK    OK    X     X
  80T SS        RO    ?     OK    ?
  80T DS        RO    RO    OK    OK
```

```
OK = compatible   RO = read only
? = incompatible but may not be detected on some drives
```

The format procedure automatically checks the drive specification and will format the drive in an appropriate manner. Note that 40 track drives which do not have an end stop, or which would suffer damage when stepped beyond the 40th track (to track 55) should not be formatted unless the number of tracks has been specified in an FLP_TRACK command. It is possible to force the disk driver to format a disk as single sided on a double sided drive by making the 11th character (it is invisible) of the medium name an asterisk: e.g.

   FORMAT 'FLP1_DISK_NAME *'

Dynamic Printer Buffer

The Printer Buffer has two names: the "usage" and the "device". The default usage name is PRT and the default device is SER. The printer buffer works by intercepting any OPEN call to a device whose name starts with its usage name. It substitutes the device name for the usage, and tries to open the device. If it succeeds, then all the output is buffered within the QLs main memory. If the device is in use, then the output is also buffered until the device is available. There is no limit to the number of buffered output files open at one time. If an error occurs during output, the buffer contents are thrown away. The current printed output may also be thrown away by the command

   PRT_ABT

This will ABORT the file with the message "****** ABORTED ******"

Using the default usage and device all references to a device called PRT will use the serial driver SER. Any parameters appended to the PRT name will by transferred to the SER name:

    OPEN #3, prt   will open SER with a buffer

    COPY flp1_fred, prt2c   will copy to SER2C with a buffer

The usage and device can be changed by using the command

    PRT_USE   usage, device

Two cases are particularly useful. In the first, the usage and device names are the same. This has the effect of introducing a buffer transparently into a device. In the second, the device name is of zero length. This means that the usage name may be followed by any device name.

    PRT_USE ser,ser   buffer all output to SER1 and SER2

    PRT_USE b_,''   b_ser1 is buffered SER1, b_par is
                           buffered PAR etc.

Screen Dumps

The screen dump facilities are available in three ways. Screen dumps may be invoked at any time with a user definable hotkey, screen dumps may be invoked using a SuperBASIC command or screen dumps may be made through the IO system from programs written in any language.

SuperBASIC Commands

There are four SuperBASIC commands for screen dumps. The principal command is SDUMP. This has three formats, which are self explanatory:

```
SDUMP                       dump whole screen
SDUMP #channel              dump SuperBASIC window
SDUMP width, height, x origin, y origin
                                    dump specified pixel area
SDUMP address of flag       dump save area
SDUMP address, width, height, x origin, y origin
                                    dump window within save area
```

The last two forms are to dump all or part of a window save area as saved by the QJUMP pointer interface command PSAVE.

The other three commands control the dump facilities. SDP_KEY sets the single key dump facility. When activated, pressing ALT and the specified key will invoke the "hotkey" dump routines.

```
SDP_KEY p                   select key "p" for hotkey dump

SDP_KEY 'p'                 ... the same

SDP_KEY                     inhibit hotkey dump
```

SDP_DEV sets the screen dump device. The default is "SER", this may be changed to any QDOS device or file name.

```
SDP_DEV 'ser2'              dump to SER2

SDP_DEV n2_par              dump to PAR on network node 2

SDP_DEV flp2_dump1          dump to DUMP1 on FLP2
```

SDP_SET sets the printer type, scaling, and print method. You may give between one and four parameters for this command.

> SDP_SET   printer number, scale, inverse, random

The printer numbers and scales are given in the following table. The "inverse" and "random" parameters are either true or false (non-zero or zero). An "inverse" dump prints black for white and vice versa. A random dump provides some impression of grey scales even at one dot per pixel. If a parameter is not given, the setting remains unchanged.

> SDP_SET 2,3,0   select Epson FX80, scale 3, not inverse
> (random unchanged)
>
> SDP_SET 1,1,1,0   select Epson MX80,scale 1,inverse,not random
> (this is the default)

IO Device SDUMP

The SuperBASIC commands all access the screen dump software through the IO device SDUMP.

The screen dump parameters may be set by sending bytes to the device SDUMP using the IO.SBYTE or IO.SSTRG IO calls. A dump is invoked by defining a window using the SD.WDEF call. In the following examples, SuperBASIC is used for simplicity. Similar calls may be made from any language.

To set the various parameters of the screen dump routines, a code byte is sent to the device followed by the parameter.

Code	Parameter	
0	character	sets the screen dump hotkey
0	0	suppresses the screen dump hotkey
1	byte	sets printer number
2	byte	sets scale
3	byte	sets inverse flag
4	byte	sets random flag

8	standard string	sets device name
9	byte length string	sets device name

Note that the streams of bytes 8,0,3,65,66,67 and 9,3,65,66,67 both set the dump device name to "ABC". The bytes may be sent one at a time (IO.SBYTE) or many at a time, using the send multiple bytes (IO.SSTRG) operation.

To dump an area of the screen, a call is made to set the "window" area using SD.WDEF. To dump from the screen, the border width (D2) should be specified as zero.

```
OPEN #4,'sdump'            open dump device
BPUT #4,1,2,2,3            set printer 2, scale 3
BPUT #4,8                  set dump device ...
PUT #4,'n2_ser'            ... to "N2_SER"
WINDOW #4,256,202,256,0    dump window (mode 4 SuperBASIC #1)
CLOSE #4                   done
```

The parameters which have been set remain set until reset by further calls. Closing the SDUMP channel has no effect other than to keep the QL tidy.

The parameter setting calls will always complete immediately, the window definition call will complete when the dump is finished. If zero timeout is specified, then the dump will continue after the window definition call has returned not complete.

From assembly language, D2 is a key defining the area of memory to be dumped. If D2 is 0, the dump will be from the screen memory. If D2 is set to 1, then the partial save area pointed to by A2 will be dumped.

If D2 is 2, then a window (defined by (A1)) within the partial save area will be dumped. This means that part (or all) of the screen can be saved before dumping. If D2 is non-zero, A2 must be set.

Screen Dump Formats

Printer	scale	dots/in	lines/in	dots 512	max width	ratio	dots 256	max width	ratio
1 Epson MX80	1	120	72	1x1	512	1.23			
or similar	1	60	72				1x1	256	1.23
	2	60	72	1x2	480	1.23	2x2	240	1.23
	3	120	72	2x2	480	1.23	4x2	240	1.23
2 Epson FX80	1	90	72	1x1	512	0.92			
additional	1	60	72				1x1	256	1.23
formats	2	90	72	1x1	512	0.92	2x1	256	0.92
	3	90	72	2x2	360	0.92	4x2	180	0.92
3 Epson FX100	1	90	72	1x1	512	0.92			
wide	1	60	72				1x1	256	1.23
carriage	2	90	72	1x1	512		2x1	256	0.92
	3	90	72	2x2	512		4x2	256	0.92
4 Epson JX80	1	90	72	1x1	512	0.92			
	1	60	72				1x1	256	1.23
	2	90	72	1x1	512	0.92	2x1	256	0.92
	3	90	72	2x2	360	0.92	4x2	180	0.92
5 Epson LQ2500	1	80	60	1x1	512	0.99			
8 pin	1	60	60				1x1	256	1.48
	2	120	60	2x1	512	0.74			
	2	80	60				2x1	256	0.99
	3	80	60	2x2	512	0.99	4x2	256	0.99
6 Epson LQ2500	1	120	180	1x2	512	0.99	1x1	256	0.99
24 pin	2	180	180	2x3	512	1.11	3x2	256	0.99
	3	180	180	3x4	512	0.99	6x4	256	0.99
7 Epson LQ2500	1	80	60	1x1	512	0.99			
8 pin	1	60	60				1x1	256	1.48
colour	2	120	60	2x1	512	0.74			
	2	80	60				2x1	256	0.99
	3	80	60	2x2	512	0.99	4x2	256	0.99

Printer	scale	dots/in	lines/in	dots 512	max width	ratio	dots 256	max width	ratio
8 Epson	1	120	60	1x2	512	0.99	1x1	256	0.99
LQ2500	2	180	60	2x3	512	1.11	3x2	256	0.99
colour	3	180	60	3x4	512	0.99	6x4	256	0.99
9 Brother	1	60	72	1x1	512	1.23			
HR4	1	60	72				1x1	256	1.23
	2	60	72	1x2	480	1.23	2x2	240	1.23
	3	120	72	2x2	480	1.23	4x2	240	1.23
10 Olivetti	1	110	72	1x1	512	1.13			
JP101	1	110	108				1x1	256	0.75
	2	110	108	1x1	512	0.75	3x2	256	1.00
	3	110	72	2x2	440	1.13	4x2	220	1.13
11 Seikosha	1	60	63	1x1	480	0.70	1x1	256	1.41
GP-100A	2	60	63	2x2	480	1.41	2x2	240	1.41
	3	60	63	2x2	480	1.41	2x2	240	1.41
12 Seikosha	1	60	72	1x1	480	0.61	1x1	256	1.23
GP-250X	2	60	72	1x2	480	1.23	2x2	240	1.23
	3	60	72	1x2	480	1.23	2x2	240	1.23
13 Seikosha	1	80	80	1x1	512	0.74	1x1	256	1.48
GP-700A	2	80	80	1x2	512	1.48	2x2	256	1.48
	3	80	80	1x2	512	1.48	3x2	212	0.99
14 Canon	1	80	80	1x1	512	0.74	1x1	256	1.48
PJ1080A	2	80	80	1x2	512	1.48	2x2	256	1.48
	3	80	80	1x2	512	1.48	3x2	212	0.99
15 Centronics	1	75	72	1x1	512	0.77	1x1	256	1.42
739	2	75	72	1x1	512	0.77	2x1	256	0.77
	3	75	72	2x2	300	0.77	3x2	200	1.03
16 C.Itoh	1	120	72	1x1	512	1.23			
7500	1	60	72				1x1	256	1.23
	2	160	72	2x1	512	0.82			
	2	120	72				2x1	256	1.23
	3	120	72	2x2	480	1.23	4x2	240	1.23

Printer	scale	dots/in	lines/in	dots 512	max width	ratio	dots 256	max width	ratio
17 Toshiba	1	180	180	1x2	512	1.48	2x2	256	1.48
TH 2100H	2	180	180	2x3	512	1.11	3x2	256	0.99
24 pin	3	180	180	3x4	512	0.99	6x4	256	0.99
18 Brother	1	70	72	1x1	512	0.72	1x1	256	1.44
8056	2	70	72	1x1	512	0.72	2x1	256	0.72
	3	70	72	2x2	280	0.72	3x2	186	0.96
19 Epson MX100	1	120	72	1x1	512	1.23			
or similar	1	60	72				1x1	256	1.23
	2	60	72	1x2	512	1.23	2x2	256	1.23
	3	120	72	2x2	512	1.23	4x2	256	1.23
20 Tandy	1	100	72	1x1	512	1.03			
DMP 105	1	60	72				1x1	256	1.23
	2	60	72	1x2	512				
	2	100	72				2x1	256	1.03
	3	100	72	2x2	400	1.03	4x2	200	1.03
21 OKI Microline	1	100	66	1x1	512	1.12			
82 / 84 OK Writer	1	60	66				1x1	256	1.35
	2	100	66	1x1	512	1.12	2x1	256	1.12
	3	100	66	2x2	400	1.12	4x2	200	1.12
22 Fastext 80	1	72	72	1x1	512	0.74			
	1	60	72				1x1	256	1.23
	2	60	72	1x2	480	1.23	2x2	240	1.23
	3	72	72	2x3	288	1.11	3x2	192	0.99
23 MT-80	1	85	82	1x1	512	0.77	1x1	256	1.53
	2	170	82	2x1	512	0.77	3x1	256	1.02
	3	170	82	3x2	425	1.02	6x2	212	1.02