

QASAR

DISKETTE    SYSTEM

MAINTENANCE

- (1) QFC-2 Floppy disc controller alignment
- (2) YD-174 Disc drive set-up and alignment
- (3) Disc system fault diagnosis

FAIRLIGHT    INSTRUMENTS

September 1981





DISC SYSTEM SET-UP AND DIAGNOSIS  
 =====

- (1) QFC-2 Floppy disc controller alignment
- (2) YD-174 Disc drive set-up and alignment
- (3) Disc system diagnosis

1) QFC-2 Alignment

Adjustment of two separate pulse lengths is required in the data separator section of the floppy disc controller. The test signals are available at points TP1 and TP2 and adjusted by 10-turn pots VR1 and VR2 respectively, on the QFC-2 board.

With a CRO set to .5uS/div and positive triggered, both signals should be high for 2.7uS when no data is being read from the disc (TP1 gets reset by data pulses).

2) YD-174 Disc Drive Pre-alignment Set-up

- (1) Check that 115V motor is fitted
- (2) Check that 50Hz pulley is fitted
- (3) Link 'Y' on YE Data p.c.b.
- (4) Link 'C' on " " " .
- (5) Move link to DS2 on p.c.b.
- (6) Remove link block package and open circuit links 'X' and 'Z' (break legs off link block package)
- (7) Reinstall link block

### Disc Drive Alignment

New disc drives require the radial alignment to be checked.

- (1) Place the drive on its side, with the main drive motor towards the bottom.
- (2) Connect a CRO as follows to the block of test pins marked "TP" near the centre of the YE Data p.c.b.

Pin	CRO
A	Channel A
B	Channel B
3	Ext trig

- (3) Set the CRO as follows:
  - Inputs on AC
  - trig on external negative
  - time base to 20 mS/div
  - add channels A and B
  - invert one channel
  - vertical sensitivity to 100mV/div
- (4) Load a disc containing the test program DSKTST and run it by typing DSKTST<CR>.
- (5) Type RA,D,S<CR> where D = drive number (0 or 1)  
S = side number (0 or 1)
- (6) Insert Alignment Disc and hit a key. This steps the head to track 38.
- (7) A "double eye" pattern should appear on the CRO. The amplitudes of the two lobes must be within 70% of each other.
- (8) If side 0 is acceptable, repeat test for side 1
- (9) If either side requires adjustment, loosen the two Phillips head screws which clamp the head carriage assembly to the steel stepper motor belt. The screws are accessed through two holes in the side of the drive chassis.
- (10) Gently tap the carriage assembly or move the belt by hand until the lobes are within 70% amplitude with the screws retightened (tightening the screws tends to change the lobe pattern), for both sides of the disc.
- (11) Hit ESC to terminate the radial alignment test.

### Track Zero Sensor Test

After the head is radially aligned, the track 00 sensor should be checked.

- (1) Still running DSKTST, type  
T0,D where D = drive (0 or 1).
- (2) Insert a scratch disc and hit any key.
- (3) T0 causes the head to oscillate between track 00 and track 01. Monitor the sensor signal at pin B12 of the J2 connector block on the drive p.c.b. It should oscillate with movement of the head.
- (4) Terminate test by hitting ESC



3) Disc System Diagnosis

The general procedure to follow in disc system fault tracing is:

- (1) Use the system test program CHECK to determine if the fault is in the drive itself (or the diskette) or the disc controller/DMA data transfer system.
- (2) If the disc drive is faulty, use DSKTST to further analyse the fault.

Test Program CHECK

Allows checking of

- Cyclic Redundancy Check (CRC) errors
- Data transfer between memory and disc
- RAM bit corruption errors

Command Syntax

CHECK <UNIT>, <HEXNUM>; <OPTIONS>

<UNIT> ::= <COLON> <NUMBER>

<HEX NUMBER> ::= <HEX DIGIT> | <HEX DIGIT>

<HEX DIGIT> ::= <NUMBER> | A | B | C | D | E | F

<NUMBER> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0

(1) Disc Integrity Check

Options: none required

This is the default check routine. Entire disc in specified drive is read to check for CRC errors.

(2) Read Data D.M.A. Verify

Option: V

Reads entire disc in specified drive twice, into separate blocks of memory and verifies data against itself.

(3) Write Data D.M.A. Verify

Options: W, D (May be used together)

The W option creates a file, writes distinctive data to each sector of the file and reads each sector of the file back, twice, into different areas of memory for verification. All unfree disc space will be allocated to the file.

The D option is a destructive (to the disc contents) test which writes a unique "ADD -29" pattern to each sector in an interleaved fashion, reads it back, and verifies the data.

Interleaving of blocks ensures track boundaries are continually being crossed. A delay can be introduced using the "T" option (see below) to isolate head-load timing problems.

#### (4) Other Options

Option	Use with	
R	W	use random number pattern instead of "29" pattern
P=XX	W	use pattern XX where XX = 'hex number' write the pattern to disk, read back and verify
E=XX	all	print error if total recoverable disc errors exceed XX where XX = 'hex number'. D Default value is 0.
T=XX	all	delay XX*10 ms. after a read/write where XX = 'hex number'
C	all	test continuously alternating between 'add-29' and a random number pattern
L	all	all error messages printed on printer

#### (5) Error messages

##### (a) Disc Read/Write Errors

These are of the form

```
**PROM I/O ERROR -- STATUS = 'status byte' AT h DRIVE i - PSN j
```

where h is not significant

i = drive number

j = physical sector number at which the error occurred

and the status byte can be interpreted as follows:

- 31 data C.R.C. error
- 32 disk is write protected
- 33 disk is not ready for some reason
- 34 deleted data address mark read
- 35 abnormal command termination

36 invalid sector address  
37 seek error (track not found)  
38 data mark read error  
39 address mark read error

(b) Verify Errors

When a verify error is encountered the offending disc sector is re-read into the QDOS sector buffer and matched against system RAM to determine where the error came from. The program then reports the corresponding address in RAM, the data expected, the erroneous data, the physical sector number of the disc where the error occurred, and the byte offset within the sector.

(6) Termination

Test is terminated by -

3SC key (sets system error status word)

More than 20 errors logged

User supplied iteration counter expired (default 1)

System error status word will be set if any error condition has been reported.



## Test Program DSKTST

-----

DSKTST comprises five main test routines and a number of utility commands. The main routines are as follows :-

#1 Write/read test	(destructive)
#2 Read C.R.C. test	(non-destructive)
#3 Worst case seek test	(non-destructive)
#4 Worst case data pattern R/W	(destructive)
#5 Sector/drive uniqueness	(destructive)

Tests can be run separately or in destructive/non-destruct groups by typing as follows -

```
DN,(0 or 1 or B) [,X]<CR> (Do all non-destruct tests)
DD,(0 or 1 or B) [,X]<CR> (Do all destructive tests)
ST#(up to 10 test no's separated by (-)),(0 or 1 or B)[,X]<CR>
```

The extended test option X accumulates error counts over a number of passes.

ESC key will abort test in progress

If stop on error option is selected (in answer to a prompt) the user may choose -

```
C continue
L loop
R reset stop on error
```

if an error stop occurs.

### Error Reporting

Error printouts take the following form :-

```
<drive no.> <error type> <track no> /<physical sector no> <*>
```

Presence of '\*' indicates a "hard" disc error

e.g. 1 E3 1F /0325 \*

```
means :-      drive no 1
              error type 3 (e3)
              track no 1f
              p.s.n 0325
              error was not recoverable on retry (*)
```

If after three retries the error still persists, it will be logged as a hard error (indicated by \*).

Error types are as follows (per QDOS ROM codes) :-

```
E1 data CRC error
E2 disk is write protected
E3 disk is not ready for some reason
E4 deleted data address mark read
E5 abnormal command termination
E6 invalid sector address
E7 seek error (track not found)
E8 data mark read error
E9 address mark read error
```

Additional error types are :-

E@ data read back is not the same as data written

Additional error types from the drive uniqueness test are :-

EA body of data buffer is not zero after test data  
EB unique data for this drive/sector is incorrect.

### Plotted Error Graphs

Errors may be summarised by use of the 'PG' command  
This summary plots the track no as the vertical ordinate  
and the number of errors as the horizontal ordinate.

A horizontal line may contain up to 11 error types (codes)  
each character represents (n\*horizontal scale) errors

The error graph is divided into two blocks. The left hand  
block relates to drive 0 errors, the right hand block to  
drive 1.

The graph is printed starting at the first track with errors  
logged and finishes with the last track with errors logged.

To stop the display rolling off the screen, control W can be  
used to stop printing. Subsequent carriage returns will  
print a little at a time, an escape will terminate the 'PG',  
and any other character will resume continuous printing.

In the case of double sided systems, each disc 'cylinder'  
is considered two tracks, so even track numbers correspond  
to side 0 of the disc and odd track numbers correspond to  
side 1.

## Utility Commands

Commands for utility programmes are as follows

HD,d,hhhh Head load timing test on drive d at speed hhhh (100 m.s = D8F0)

IX,d Index sensor alignment test on drive d.  
t1=tk 1. t2=tk 76

AT,d,s Read data amplitude test on drive d.  
s is optional side select (0 or 1).  
t1=tk 0. t2=tk 76

RA,d,s Radial alignment test on drive d.  
s is optional side select (0 or 1).  
t1=0-38. t2=77-38. t3=39-38. t4=37-38.

AZ,d,s Head azimuth test on drive d.  
s is optional side select.  
t1=0-76. t2=75-76

T0,d Track zero sensor alignment test on drive d.  
t1=1-2 lp. t2=0-1 lp. t3=0-2 lp

SK,d,s Head skew test on drive d.  
s is optional side select (0 or 1).  
t1=1-76 lp

RS,d,hhhh Read sector hhhh from drive d to buffer

WS,d,hhhh Write buffer to sector hhhh on drive d

DB Display buffer in hex and ascii

FB,hhhh Fill buffer with repeating pattern hhhh

The running test may be aborted by escape key

The next test of the sequence is entered by depressing space key  
Tests followed by letters "lp" move head between tracks shown.

The drive manufacturer's manual should be consulted for information on the tests which should be applied

Some tests require the appropriate alignment diskette and ask that it be inserted

Other tests require a scratch diskette and ask that it be inserted.

Typing OS<CR> will return the user to the operating system (reboot).



## CHAPTER 6

### QDOS BASIC

#### 6.1 INTRODUCTION

This chapter describes the operation of the BASIC Interpreter with the QDOS Flexible Disk Operating System.

#### 6.2 SYSTEM REQUIREMENTS

Use of the BASIC Interpreter with QDOS requires the following minimum system:

- . BASIC or QBAS Interpreter diskette
- . QASAR Microcomputer
- . 16K bytes of memory
- . TTY or RS-232C compatible terminal
- . QDOS Flexible-disk operating system.

#### 6.3 LOADING THE BASIC INTERPRETER

With the above minimum system up and running and the BASIC containing diskette installed, the BASIC Interpreter may be invoked by use of the BASIC or QBAS command described below. Procedures for system start up, should they be needed, are provided in Chapter 1 of the QDOS Disk Operating System User's Guide.

#### 6.4 COMMAND DESCRIPTION

Statement format:

= BASIC <DELIM><IFILE>[,<OFILE>]

or:

QBAS<DELIM><IFFILE>[,<OFILE>]

Where:

<DELIM> is a command line delimiter, <IFILE> is the file specification of the Basic source programme to be loaded, and <OFILE> is the file specification of the Basic source programme to be saved. Both file specifications are of the form:

<FILENAME>[.<SUFFIX>][:<LOG.DRV>].

"SA" and "Ø" will be the default suffix and logical drive number if they are not explicitly given on the command line.

Action:

If <IFILE> already exists, it will be the file used for input (to be loaded). If <IFILE> does not exist, it will be the file name of any Basic source programme to be created and saved.

The following programme is an example of a BASIC disk/I/O operation in which data is read from an existing disk file for use by BASIC, both old and newly generated data placed in a new file, and both files closed, as required, for subsequent operation with QDOS.

```
0010 OPEN #3, IFILE, I
0020 OPEN #4, OFILE, O
0030 FOR I=1 TO 10
0040 INPUT #3,A,B,C,D$
0050 X=(A+B+C)
0060 PRINT X,D$
0070 PRINT #4,A,B,C,X,D$
0080 NEXT I
0090 CLOSE #3
0100 CLOSE #4
0110 END
```

PAGE 001 IFILE XSA:0

```
153 285 678 JONES
842 372 377 SMITH
330 130 472 WILLIAMS
220 553 664 JOHNSON
133 448 663 BATES
229 225 773 DOOLEY
294 613 773 MACK
394 553 112 JAMES
852 227 163 CARLSON
223 665 140 PETERSON
```

PAGE 001 OFILE .SA:0

```
153 285 678 1116 JONES
842 372 377 1591 SMITH
330 130 472 932 WILLIAMS
220 553 664 1437 JOHNSON
133 448 663 1244 BATES
229 225 773 1227 DOOLEY
294 613 773 1680 MACK
394 553 112 1059 JAMES
852 227 163 1242 CARLSON
223 665 140 1028 PETERSON
```

Note that due to the variable length of data records when using A.S.C.I.I. format files, it is not possible to update a particular space compressed record by opening the file in UPDATE mode. An example of a valid use of UPDATE mode is shown below.

```
READY
#LIST
0005 OPEN #5,FILE2,U
0010 REM PRINT A STRING ENCLOSED IN QUOTES
0020 Q$=CHR$(34)
0030 S$="MOTOROLA,INC"
0040 PRINT #5,Q$;S$;Q$
0060 RESTORE #5
0070 INPUT #5,S$
0080 PRINT S$
0090 CLOSE 5
0110 END
```

FILENAME .SUFFIX:LUN may be replaced by a string variable.

e.g.

```
10 N$ = "TEST"
20 OPEN #3, N$,I
```

will cause the file TEST to be opened as an Input File, the default suffix ".SA" and L.U.N. ":0" being used.

M is one of the following data transfer modes:

I is open input only, file name must exist

O is open output only, file name must not exist

U is open update, allows input or output, creates the file if non-existent.

When a file is opened the record pointer is positioned at the beginning of the file. Each read or write moves the pointer one record (one line of ASCII text, terminated by CR) and reads in as many variables as specified in the INPUT statement, the rest of the line being discarded.

#### 6.6.2 Command description. (CLOSE)

Statement Format: L CLOSE #n

Where:

L is the source line number

CLOSE is the statement or command identifier

n is the I/O stream number (an integer from 3 through 9) assigned to the QDOS file name to be closed.

Action:

The CLOSE command/statement releases the I/O stream number associated with the QDOS file name when the file was opened, and insures that QDOS properly closes the file.

If a file is opened in UPDATE mode and then closed with the file position pointer still at the beginning of the file, the file will be deleted.

When the execution of a programme is abnormally terminated by an error or the BREAK key, I/O stream numbers are not released nor are any open files closed. Prior to running the programme again, any open files must be closed. Any files that were opened for output, and written to, will be entered in the directory and must be deleted before re-execution. This may be accomplished in the following manner without returning to QDOS.

```
#CLOSE #n (closes file n)
#OPEN #n, File name, U }
#CLOSE #n                } (deletes file n)
```

NOTE: Use of either of these statements without a line number causes immediate execution of that statement. The initial # is the BASIC prompt character.

The following listings are examples of the use of the CLOSE command:



If <OFILE> is not specified on the command line, and the programme in memory is to be saved, <IFILE> will be updated to reflect any changes made to the programme.

Basic Source Programmes are ASCII files; thus, they can be created or modified by the QDOS Editor, and generally treated as any other QDOS ASCII files.

## 6.5 SPECIAL CONTROL CHARACTERS

### A. BREAK Key

Depressing the BREAK key on the terminal keyboard will cause the BASIC programme to halt the current operation and to respond with the word READY. The line number of the statement being executed when the BREAK key was depressed will be typed out.

### B. PRINT PAUSE (CONTROL W)

Depressing and holding down the CTRL key on the terminal keyboard and then depressing the W key allows the operator to temporarily suspend any BASIC operation. The operator may allow the BASIC programme to resume operation by depressing any key on the keyboard.

This operation may be useful when listing a programme to a CRT.

## 6.6 DISK FILE I/O

The QDOS BASIC has been enhanced with disk file I/O. Three new commands/statements have been added to open, close, and restore disk data files. The enhanced BASIC interpreter QBAS provides a further two disk oriented commands which allow programme files to be loaded from disk using a direct command or appended using a programme statement.

### 6.6.1. Command Description (OPEN)

The OPEN statement assigns an I/O stream number to a QDOS file name, and opens the file for data transfers. This stream number is then used with the PRINT and INPUT statements to write and read data records to the disk file. A maximum of three data files may be open at one time. Each is opened as follows:

Statement Format: L OPEN #n, FILENAME[.SUFFIX:LUN],M

Where:

L is the source line number

OPEN is command or statement identifier

n is the I/O stream number (an integer from 3 through 9) associated with the QDOS file name. This I/O stream number is subsequently used to refer to the file for subsequent INPUT, PRINT, RESTORE or CLOSE statements.

FILENAME[.SUFFIX:LUN] is the QDOS file name specification in the standard QDOS format. If .SUFFIX and :LUN are not specified .SA will be the default suffix and drive zero will be the default LUN. When using the QBAS interpreter,

```
#  
READY  
#LIST  
0010 REM CLOSE OPEN FILES AFTER AN ERROR  
0020 OPEN #3,DATA,I  
0030 INPUT #3,A,B,C,D$  
0040 K=SQR(A)  
0050 PRONT X  
0060 END
```

```
READY  
#RUN
```

```
ERROR# 03 IN LINE 0050
```

```
READY  
#50 PRINT X  
#RUN
```

```
ERROR #28 IN LINE 20
```

```
READY  
#CLOSE #3
```

```
READY
```

```
#RUN  
3.162278
```

```
READY  
#LIST  
0010 REM DELETE A QDOS FILE FROM BASIC  
0020 OPEN #4,DATAF,O  
0030 A=10  
0040 B=20  
0050 PRINT #4,A,B  
0060 ERROR  
0070 CLOSE #4  
0080 STOP
```

```
READY  
#RUN
```

```
ERROR# 03 IN LINE 0060
```

```
READY  
#60  
#CLOSE #4
```

```
READY  
#OPEN #4,DATAF,U
```

```
READY  
#CLOSE #4
```

```
READY  
#RUN  
STOP 0080  
READY
```

6.6.3. Command Description (RESTORE)

Statement Format: L RESTORE #n

Where:

L is a source line number

RESTORE is the command or statement

n is the I/O stream number (an integer from 3 through 9) assigned to the QDOS file that is to be rewound or restored.

Action:

The RESTORE command/statement resets file pointers so that subsequent reads or writes access the beginning of the disk file.

6.6.4. Input Files

Records from the input data file may be read by the BASIC command/statement INPUT #n,[VARB]<sup>n</sup>.

Where:

n is the I/O stream number (from 3 through 9) assigned to a QDOS file name by the OPEN command, and must be an integer.

[VARB]<sup>n</sup> are variable names, separated by commas, to be assigned the numeric values or strings in the data record.

Each INPUT statement reads in a full line from the disk file, if there are more variable fields in the data record than variables requested in the INPUT statement, the surplus will be ignored.

6.6.5. Data Records

Numbers and strings must be separated by spaces and/or commas. If strings of text are to include commas or spaces, they must be in quotes.

6.6.6. Output Files

Data records may be written to an QDOS file by the BASIC command/statement: PRINT #n,[VARB]<sup>n</sup>.

Where:

n is the I/O stream number (from 3 through 9) assigned to a QDOS file name by the OPEN command, and must be an integer.

[VARB]<sup>n</sup> are variable names and/or strings separated by commas or semicolons.

The records will be formatted as if they were printed on the console or printer.



READY  
#RUN  
MOTOROLA, INC

READY  
#

6.6.7. Command Description APPEND (QBAS only)

Statement Format: L APPEND #n

Where:

L is the source line number

APPEND is the command or statement identifier

n is the I/O stream number (an integer from 3 to 9) associated with the QDOS file which has been opened in INPUT or UPDATE mode.

Action:

The file associated with #n is a programme source file. It will be read into the QBAS execution buffer on a line-number match basis and execution will proceed uninterrupted. This facility allows overlaying source programmes.

e.g.:

```
LIST
0010 INPUT "WHICH PROGRAMME DO YOU WANT TO RUN",N$
0015 IF N$="QUIT" GOTO 120
0020 OPEN #5,N$,I
0030 APPEND #5
0060 PRINT
0100 CLOSE #5
0110 GOTO 10
0120 END
READY
#RUN
WHICH PROGRAMME DO YOU WANT TO RUN ? PROG1
```

THIS IS PROGRAMME 'PROG1' EXECUTING

WHICH PROGRAMME DO YOU WANT TO RUN ? QUIT

```
READY
#LIST
0010 INPUT "WHICH PROGRAMME DO YOU WANT TO RUN",N$
0015 IF N$="QUIT" GOTO 120
0020 OPEN #5,N$,I
0030 APPEND #5
0040 PRINT
0050 PRINT "THIS IS PROGRAMME 'PROG1' EXECUTING"
0060 PRINT
0100 CLOSE #5
0110 GOTO 10
0120 END
READY
#EXIT
SAVE(Y/N)
N
=LIST PROG1
```

```
0040 PRINT
0050 PRINT "THIS IS PROGRAMME 'PROG1' EXECUTING"
0060 PRINT
=
```

6.6.8. Command Description LOAD (QBAS only)

Statement Format:           LOAD #n

Where:

#n is the I/O stream number associated with a QDOS file which has been opened in INPUT or UPDATE mode.

Action:

The input file is a source programme file. It is loaded into the programme execution buffer, the existing programme being deleted. This command should be used as a direct command only not as a programme statement.

e.g.:

```
#LIST
0010 REM THIS IS A BASIC PROGRAMME
0020 END
```

```
READY
#OPEN#4,TESTPGM,I
READY
#LOAD #4
READY
#LIST
0100 REM THIS IS TESTPGM
0200 END
```

READY

6.7 ERROR MESSAGES

Error #28	Invalid or duplicate I/O stream number
Error #29	Maximum of three data files may be opened
Error #30	Invalid file name
Error #31	Invalid data transfer type (not I,O, or U)
Error #32	Data file cannot be opened, closed, or accessed
Error #33	QDOS file not ASCII format
Error #34	Disk data record format error
Error #35	Log function error
Error #36	BASIC programme cannot be altered and continued from a stop.

