IKON COMPUTER PRODUCTS

HOBBIT OPERATING SYSTEM
VERSION 2

Phillips mr PCB ---- 3922 536 07665
Mek ------ --- --- 07860

Kiln Lake
Laugharne
Dyfed

Tel: Laugharne 515

The Hobbit is a low priced digital cassette system designed to be connected to a NASCOM PIO port.

The system is fully automatic and uses the ultra reliable Philips digital cassette recorder.

Each side of the cassette can hold 50.5K Bytes, organized in up to 69 files.

Data transfer is at 750 BYTES PER SECOND.

The Hobbit operating system is supplied either in a 2716 or 2 2708 EPROMS normally addressed at D000H (other addresses can be supplied on request).

POWER REQUIREMENTS

+12 Volts @ 120 MA
+5 Volts @ 10 MA

COMPATIBILITY

The Hobbit with version 2 operating system is compatible with all NAS-SYS monitors.

HOBBIT OPERATING SYSTEM VERSION 2

The Hobbit operating system has been enhanced in the following ways.

1. NAS-SYS and Hobbit commands can be freely mixed. All Hobbit commands start with a "£" followed by the command letter.

2. The Hobbit command set is available even when BASIC is running. Just type the command as normal.

3. A new command has been added to speed up multiple file operations.

4. The address of the command table has been moved to reduce conflicts with other programs.

5. Commands which are potentially destructive have been changed to lower case to reduce the risk of accidents.

INSTALLATION

1. Connect the eight wires of the multi-core cable to the "A" port of the NASCOM PIO. If a second drive has been purchased then this should be connected to the "B" port. The connections should be made as follows

BIT	WIRE
0	Purple
1	Blue
2	Green
3	Yellow
4	Orange
5	Red
6	Brown
7	Black

2. Connect the three individual power leads to the NASCOM power supply as follows:

Black	0 Volts
Red	+5 Volts
Blue	+12 Volts

N.B. It is most important that these connections are made correctly.

3. Insert the 2 2708's or the 2716 into the appropriate socket(s) on your NASCOM computer. Select addresses so that the ROMs are at address D000 and D400 unless ROMS at different addresses have been ordered.

4. The Hobbit operating system has been optimized to work in a NASCOM 2 running with a 4MHZ clock with wait states. It is however usually possible to run it without wait states provided the following is borne in mind. If it is intended to create files with the computer running with a clock speed of either 4MHZ with wait states or 2MHZ then the tape must be formated at one of these speeds.

The Hobbit version 2 software is designed to make use of the jump on reset facility of the NASCOM 2. If you do not intend to use this facility then it is desirable to prevent the NASCOM 2 PIO from being reset when the master reset is pressed. This can be achieved by removing IC 8 from its socket and inserting a small wire link between pins 8 and 9 on the socket. Pins 8 and 9 on the IC should be bent such that when the chip is replaced in the socket they are not connected.

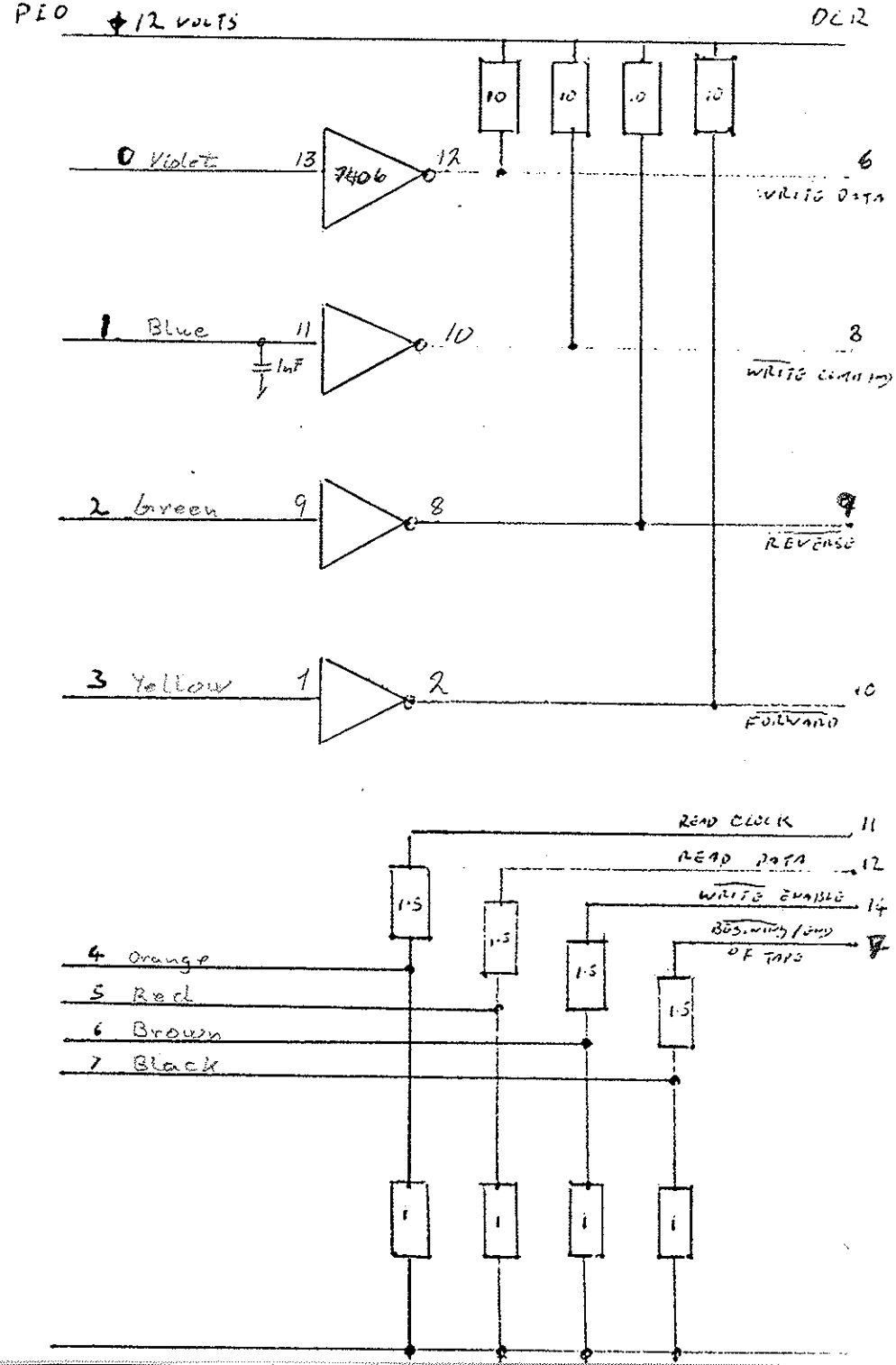
USING THE HOBBIT

When the computer is switched on the motor in the cassette unit will be running. For this reason you should not have a cassette in the drive until the Hobbit monitor has been started.

STARTING THE HOBBIT MONITOR

If your computer is running at 4MHZ then type ED000 or use the jump on reset facility of the NASCOM 2 to automatically start the Hobbit monitor.

For those working at 2MHZ type ED00 1 once the Hobbit monitor has been initialized the cassette motor will stop and the cassette may now be inserted (open end first).



MEMORY

When the Hobbit monitor is started it automatically sizes the computer memory and uses the top 1325 (decimal) 52DH Bytes of memory. (Two drive configurations require 2570 (decimal) AOAH.)

It is important that useres make sure that they do not over write this area.

The address of the command table is put into OC1EH (ARG10).

THE COMMANDS

All the NAS-SYS commands work as normal, the only exceptions are the N and U command. The N command will switch the Hobbit command set off the the U command will switch it on again.

Hobbit commands consist of a square bracket [followed by a command letter.

File names: The W, R, L, D and C commands require file names to be supplied. File names are entered after the command has been executed. The computer will display the message "NAME". File names consist of between 1 and 6 characters terminated by a new line.

[f
FORMAT:

New cassettes require certain timing information pre-recorded to enable the Hobbit file handling programs to operate. This command is used to do this. The cassette will first rewind then run to the far end, rewind again and then go forward to the centre of the cassette and stop. This process will take about 4 minutes. Remember to format both sides of the cassette.

[M
MOUNT: This command is used to transfer the cassette index, which is in the middle of the cassette, into the computer memory. Whenever you change a cassette this must be the first command executed, otherwise the index in the memory will not correspond to the index on the cassette. The only exception to this rule is when a FORMAT command has been executed, when the MOUNT command is automatically evoked.

[Wxxxx YYYY ZZZZ
WRITE: The write command writes a file onto the cassette. Data from XXXX up to but not including YYYY is written. If it is a program then ZZZZ is the address at which the program is to start.

[R
READ: This command reads a file created by the WRITE command. Data is read back into the memory locations from which it was written. Program control is then passed to the address specified in the third parameter of the write command.

[Lxxxx
LOAD: This command loads a file created by the WRITE command into memory. The parameters supplied in the WRITE command are ignored. Instead the data is loaded into the specified memory locations.

[D
DELETE: This command is used to delete a file and make the blocks occupied by it available for new files.

[k
KILL: This is very similar to the DELETE command except that it deletes all the files on that side of the cassette. It is much quicker than deleting each file individually.

[C
CHANGE: This command is used to change a file name. After the command has been entered the message "NAME" will be displayed. The name of the file to be changed should be entered. After the name has been entered the message will be displayed again. This time enter the new name of the file.

[N
NAMES: The names of the files will be displayed on the screen in groups of five. Press new line for the next group. At the end of the list the message FREE NN will be displayed where NN is the number of free blocks available for new files. Each block can hold 747 Bytes of user data. A blank cassette has 69 free blocks. The order in which the file names are displayed does not necessarily relate to the order in which they were created.

[E
END: This command rewinds the cassette ready for removal. It is advisable to do this so as to safeguard the cassette from the effects of dust.

[Zxxxx YYYY
ZEAP: This command may be used to create files from areas of memory where the first word in the file is the length of the file. ZEAP files are of this type. For example, to save a file made by tape ZEAP which creates its file from 2000H and has a warm start address of 1003H type Z2000 1003.

[Bxxxx YYYY
BASIC: This command is similar to the Z command except the first word in memory is the address of the end of the file plus one. BASIC and NASPEN files are of this type. For example to save BASIC program created by ROM BASIC type B10D6 FFFD and to save a NASPEN file type B101A B806.

[SX
SELECT: If you have a two Hobbit cassette system attached to your computer then use the command to select the appropriate unit.

[T
TRANSFER: Transfer a file from the selected unit to the other one. After the command has been entered the message "NAME" will be displayed. The name of the file to be transferred should be entered. When the name has been entered the message will again be displayed. This time enter the name to be assigned to the file on the other drive.

[ix
INHIBIT: When a file is created or deleted the index on block 35 of the tape is updated. If a number of files are to be created or deleted in a batch it may be desirable to prevent the index on the tape being updated until all the files have been dealt with. To prevent the system automatically updating the index on the tape after every file operation type
[i1
Before the cassette is removed from the Hobbit it will be necessary to instruct the operating system to copy the index in memory on to the cassette. This is achieved by typing
[i0
This command also enables automatic cassette index updating.

This command should be used with caution. The i0 command will copy the index in memory onto the cassette even if the computer has just been switched on and the index is full of rubbish!!

ERRORS

If the Hobbit operating system detects an error the message "ERROR N" will be displayed, where N is the error code.

A FILE EXISTS
An attempt has been made to create a file using a name already assigned to another file.

B BAD FILE STRUCTURE:
This error is only likely to be produced in files created by assembly language programs. Each block of a file contains a header consisting of the number of the next block in the file and the index entry number. This number is the position of the file name in the index. When a block is read from the cassette this number is compared with the original I.E.N. If they are different then clearly the file has not been created properly and the error message is produced.

C HARD READ ERROR:
If the Hobbit misreads a block on the cassette then it will automatically reread it up to 15 times. In the unlikely event that it still cannot read the block correctly then this error message is produced.

D DEVICE FULL UP:
A cassette has become full during the creation of a file. The file is automatically closed.

E NO SUCH FILE:
An attempt has been made to read or delete a non-existent file.

G CASSETTE WRITE PROTECTED:
An attempt has been made to write to a cassette which has been write protected (by removing the write plug from the cassette). Note: If a cassette is write protected the delete and format commands will not work correctly.

THE HOBBIT AND MICROSOFT BASIC

All the commands will work from "within BASIC". Thus it is not necessary to leave basic to save your program. It must be remembered that when you cold start BASIC sufficient room must be left at the top of the memory for the Hobbit workspace. For those who wish to read and write Hobbit files from basic programs there is available a microsoft basic upgrade kit. This allows files to be created using print statements and read using input statements in much the same way as information is printed to the screen and input from the keyboard.

If you already have a basic upgrade kit for the version 1 operating system it will be necessary to amend the addresses of SELECT, COMMAND, GETSEL, OPENW, ERROR, PUT, CLOSE, GET and OPENR. The new addresses are given in this manual. See paragraph 5 of the microsoft basic upgrade kit instructions.

THE HOBBIT AND ASSEMBLY LANGUAGE

General Information

The IY register is used to point to a table of information about the system. This table is called the device table. Each Hobbit drive has its own device table. The values in brackets after the name are the offset from the bottom of the table followed by the length in Bytes. E.g., the value at IY+[illegible] and IY+[illegible] is the address of the INDEX.

AMNOW(0,1) This contains the current position of the tape.

PORT(1,1) This is the port address (data) for this drive.

SPEED(2,1) 0 for 4MHZ clock 1 for 2MHZ clock.

ERRORS(3,1) Read errors are counted here and an error "C" is declared if it reaches 16.

INH(4,1) Inhibit flag set to 1 if index updating is inhibited and 0 if updating is automatic.

COMS(5,2) Used for communication between some of the timing routines.

SEL(7,1) Only defined in the device table associated with unit 0. If this is a 1 then unit 0 is selected, if it is 0 then unit 1 has been selected.

BITMAP(8,2) This is the address of the bitmap which is a 9 Byte area of memory which has a bit set for each block used by a file and a bit reset for each unassigned block. Thus by counting the number of zero bits in the bitmap the number of free blocks can be ascertained.

INDEX(10,2) This is the address of the file name index. The index is made up of 69 slots. Each slot is 7 Bytes long. The first 6 Bytes contain the file name and the 7th Byte contains the block number of the start of the file. A slot is considered empty when the first 6 Bytes are zero. When a file is created the system searches the index for a vacant slot and inserts the file name and the first block number.

DATA BUFFERS

The IX register is used to point at a data buffer which is used to hold the data read from or to be written to a block on the tape. This buffer has a header which contains the following

INDX(-4,1) This is the index entry number obtained when the file was opened.

POINT(-3,2) This points to the next data Byte in the buffer. It is used by the GET and PUT subroutines.

NSEC(0,2) This is the address of the next block in the file. If bit 15 of this is set then the block just read is the last block in the file and bits 0-14 constitute a pointer to the last data Byte in the file.

IEN(2,1) This is a copy of (IX+IENM). It is written on to the tape when a block is recorded.

DESCRIPTION OF PRINCIPAL SUBROUTINES

Error handling

If a subroutine detects an error then the error code (ASCII) is put in the A register and the C flag is set.

GETSEL: This subroutine must be called before any other Hobbit subroutine can be used. It returns with IY pointing to the correct device table and IX pointing to a cassette data buffer. If two or more files are being accessed simultaneously on the same drive then additional data buffers will be required. Data buffers are set up as follows

BUFFER DEFS 755 Make 755 (decimal) Byte buffer

LD IX,BUFFER+4 Load address+4 into IX

OPENW: Opens a file for writing. HL must point to a 6 Byte file name.

PUT Puts the Byte of data contained in the A register into the file.

CLOSE: Closes a file that has been opened for writing. It is important that this subroutine is only called to close a file opened with the OPENW subroutine. There is no close required for files opened for read.

OPENR: Opens a file for reading. The file name is pointed to by the HL register as in the OPENW subroutine.

GET: Gets a Byte of data from a file and returns it in the A register. If the end of the file is detected then the C flag will be set and the A register will contain FFH.

DELETE: Deletes a file. The HL register must point at the file name.

ERROR: Displays the message "ERROR A" on the screen. The error code (ASCII) should be put in the A register.

FIND: Finds an entry in the index. HL points to the file name. On return DE points to the entry if found and the C register contains the I.E.N. The HL register is preserved. This subroutine is also used to find an empty slot by setting HL to point at a six Byte string of zeros. It can also be used to check whether a file name is already in use. (cy=0)

GETSEC: Gets a block from the bitmap and returns the block number in DE. The bit is set in the bitmap.

SCRAP: Clears an entry in the bitmap. DE contains the block number.

PREADX: Reads a block from the tape. DE contains the block number. No I.E.N. check is performed.

PWRITX: Writes a block on to the tape. DE contains the block number. The I.E.N. is not copies from (IX+IENM) to (IX+IEN).

PWRITE AND PREAD are as the above but include I.E.N. handling.

CALLING THE COMMANDS FROM ASSEMBLY LANGUAGE

Commands may be called provided that GETSEL is called first. Any arguments should be put in ARG1, ARG2 and ARG3 as required.

THE COMMAND TABLE

The address of the command table is contained in OC1EH. The command table is set up as shown

```
TABLE DEFB "W
      DEFW WRITE
      DEFB "L
      DEFW LOAD
        :
        :
      NOP  End of table
```

ADDRESSES

The addresses of the various routines in the Hobbit are given by the address in the table added to the address of the start of the monitor. Thus if your monitor starts at D000H then the address of PUT would be 6FFH+D000H=D6FFH.

NAME	ADDRESS
GETSEL	11F = -24289
OPENW	661
PUT	6FF
CLOSE	3EE
OPENR	643
GET	6B2
DELETE	749
ERROR	193
FIND	61A
GETSEC	5F2
SCRAP	76C
PREADX	463

NAME	ADDRESS
PWRITX	434
PWRITE	42E
PREAD	456

MONITOR COMMAND ADDRESSES

COMMAND	ADDRESS
W	1B0 = -24464
L	1ED = -24083
R	1FC = -24068
S	22D = -23955
f	490 = -23408
E	500 = -23296
M	797 = -22601
k	23E = -24002
N	245 = -23995
C	2AC = -23896
D	2CD = -23859
T	2D7 = -23849
Z	173 = -24205
B	189 = -24183
i	7ED = -22547

BASIC WRITE

DONE 4100, [illegible] : H8=USR(H0)

DEMONSTRATION SUBROUTINE

This subroutine copies the files "ALPHA" and "BETA" into the file "DELTA". The file "ALPHA" is then deleted. The system buffer has been used for the output file and a second buffer called INBUF has been used for the input file. The system buffer has also been used to delete the file "ALPHA".

```
ROOT    EQU DOCCH     Start of Hobbit monitor
GETSEL  EQU ROOT+11FH
OPENW   EQU ROOT+661H
PUT     EQU ROOT+6FFH
CLOSE   EQU ROOT+3EEH
```

```
OPENR   EQU ROOT+643H
GET     EQU ROOT+682H
DELETE  EQU ROOT+749H
ERROR   EQU ROOT+193H
        CALL GETSEL     Get device table and system buffer
        LD HL,DELTA     Point to output file name
        CALL OPENW      Open for writing
        JP C ERROR      If error display message and return
        LD (OUTSAV),IX  Save buffer
        LD IX,INBUF+4   Input buffer address
        LD HL,ALPHA     Point to first file name
        CALL OPENR      Open for read
        JP C ERROR      Abandon if error
LOOP1   CALL GET        Read a Byte
        JR NC ABC       Continue if no carry
        CP FFH          Check for end of file
        JR Z NEXT       Go and do the next file
        JP   ERROR      Anything else is an error
ABC     LD IX,(OUTSAV)  Recover O/P buffer
        CALL PUT        Put data into file
        JP C ERROR
        LD IX,INBUF+4   Get input buffer
        JR LOOP1
NEXT    LD HL,BETA      Now read file "BETA"
        CALL OPENR
        JP C ERROR
LOOP2   CALL GET        Get Byte from file
        JR NC DEF
        CP FFH          Check for end of file
        JR Z FINISH
        JP ERROR        Otherwise error
DEF     LD IX,(OUTSAV)  Output buffer
        CALL PUT
        JP C ERROR
        LD IX,INBUF+4
        JR LOOP2
FINISH  LD IX,(OUTSAV)  Get O/P Buf
        CALL CLOSE      Close the file
        LD HL,ALPHA     Delete file "ALPHA"
        CALL DELETE
        RET             Return to calling program
```



```
ALPHA   DEFM /ALPHA /
BETA    DEFM /BETA  /
DELTA   DEFM /DELTA /
INBUF   DEFS 755
OUTSAV  DEFW 0          Save O/P buf address
```