

THE BBC MICROCOMPUTER

IEEE488 INTERFACE USER GUIDE

THE BBC MICROCOMPUTER

IEEE488 INTERFACE USER GUIDE

Designed, developed and manufactured in the United Kingdom by
Acorn Computers Ltd and Intelligent Interfaces Ltd

written by
Andrew G Ray
Intelligent Interfaces Ltd
January 1983

© Copyright Acorn Computers Limited and Intelligent Interfaces Limited 1983

Neither the whole or any part of the information contained in, or the product described in, this manual may be adapted or reproduced in any material form except with the prior written approval of Acorn Computers Limited (Acorn Computers) and Intelligent Interfaces Limited (II).

The product described in this manual and products for use with it, are subject to continuous development and improvement. All information of a technical nature and particulars of the product and its use (including the information and particulars in this manual) are given by Acorn Computers and II in good faith. However, it is acknowledged that there may be errors or omissions in this manual. A list of details of amendments or revisions to this manual can be obtained upon request from Acorn Computers Technical Enquiries. Acorn Computers welcome comments and suggestions relating to the product and this manual.

All correspondence should be addressed to:-

Technical Enquiries
Acorn Computers Ltd
Fulbourn Road
Cherry Hinton
Cambridge CB1 4JN

All maintenance and service on the product must be carried out by Acorn Computers authorised dealers. Acorn Computers and II can accept no liability whatsoever for any loss or damage caused by service or maintenance by unauthorised personnel. This manual is intended only to assist the reader in the use of the product, and therefore Acorn Computers and II shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any error or omission in, this manual, or any incorrect use of the product.

WARNING: THIS APPLIANCE MUST BE EARTHED

IMPORTANT: The wires in the mains lead for this appliance are coloured in accordance with the following code:

GREEN AND YELLOW	EARTH
BLUE	NEUTRAL
BROWN	LIVE

If the socket outlet available is not suitable for the plug supplied, the plug should be cut off and an appropriate plug fitted and wired as described. The plug which is cut off must be disposed of as it would be a potential shock hazard if it were to be plugged in with the cut off end of the mains lead exposed. The moulded plug must be used with the fuse and fuse carrier firmly in place. The fuse carrier is of the same basic colour* as the coloured insert in the base of the plug. Different manufacturers' plugs and fuse carriers are not interchangeable. In the event of loss of the fuse carrier, the moulded plug **MUST NOT** be used. Either replace the moulded plug with another conventional plug wired as described, or obtain a replacement fuse carrier from an authorised BBC Microcomputer/IEEE488 dealer. In the event of the fuse blowing it must be replaced, after clearing any faults, with a 3 Ampere fuse that is ASTA approved to BS1362.

As the colours of the wires may not correspond with the coloured markings identifying the terminals in your plug, proceed as follows:

The wire which is coloured green and yellow must be connected to the terminal in the plug which is marked by the letter E, or by the safety symbol -!- or coloured green, or green and yellow.

The wire which is coloured blue must be connected to the terminal which is marked with the letter N, or coloured black.

The wire which is coloured brown must be connected to the terminal which is marked with the letter L, or coloured red.

* Not necessarily the same shade of that colour.

C O N T E N T S	Pg
INTRODUCTION	1
Connecting the Interface to the Computer	1
The IEEE488 Standard	1
Information Covered and Where to Find It in this User Guide	2
1 GETTING GOING - Simple Data Transfer	3
The IEEE Filing System	3
Data - strings of up to 255 characters	6
Data Output	6
Data Input	7
Checking the State of the Interface	8
An Example of Data Input and Output	8
2 MORE ON DATA TRANSFER	10
Binary Coded Data	10
The TRANSFER Command	11
3 HOW TO FIND OUT WHAT A DEVICE IS DOING	13
Serial Poll	13
Parallel Poll	14
4 OTHER COMMANDS	16
The Timeout Facility	16
Remote Local Operation	
Initialising Devices	17
Triggering Simultaneous Operation of Devices	18
5 PASSING CONTROL	19

6 THE IEEE FILING SYSTEM - A Summary	21
Channels	21
Device Addresses	22
Device Dependent Messages (Data)	22
Glossary of IEEEFS Commands	25
7 USING THE IEEE FILING SYSTEM IN ASSEMBLER	60
8 HARDWARE	64
REFERENCES	
APPENDICES	
1 IEEEFS Commands - Minimum Abbreviations	
2 IEEEFS Errors	
3 Interface Remote Messages	
4 Interface Specification	
5 Fitting the IEEEFS EPROM	
6 Removal of Terminating Resistor Packs	
INDEX	

INTRODUCTION

Before using your IEEE488 interface check that you have received the following items in addition to this User Guide

- **IEEE488 Interface**
- **IEEEFS EPROM**
- **Guarantee Card**
- **Delta Ribbon Cable** (for connecting one IEEE488 device to the computer)

If any item is missing, contact the dealer who supplied the IEEE488 interface quoting the order number given to you when you placed your order.

It is strongly recommended that the IEEEFS EPROM be fitted by a dealer. However, fitting instructions are included in Appendix 5, and information on the use of the terminating resistors in the IEEE488 Interface are contained in Appendix 6.

Connecting the Interface to the Computer

- 1 Turn off the power to the computer and the interface.
- 2 Plug the 34-way IDC socket on the end of the ribbon cable into the IMHZ Bus header on the underside of the computer. Ensure that pin 1 of the socket is aligned with pin 1 of the header. The small indents on the socket should face the printed panel.
- 3 Turn on the power to the interface and computer.

The IEEE488 Standard

The IEEE488 interface is a general purpose system for exchanging digital data bit parallel byte serial form between a number of devices in a local area. The original concept was developed by Hewlett Packard who called it the HP-IB Hewlett Packard Interface Bus. The IEEE (Institute of Electrical and Electronics Engineers) first produced a standard in 1975, IEEE Std 488-1975. This was

revised in 1978 as IEEE Std 488-1978. The interface is identical to that described in the IEC (International Electrotechnical Commission) Publication 625-1, apart from the mechanical connection system. The IEEE standard specifies a 24 pin Microribbon connector whereas the IEC Publication specifies a 25 pin D-type connector as used for the EIA R5232-C data communication interface.

The IEEE499 interface makes use of two types of messages:

- i) interface messages (commands) used to manage the interface
- ii) device dependent messages (data)

A device can have the ability to send data (act as a talker), receive data (act as a listener), or do both (act as a talker-listener), or manage the system (act as a controller).

This User Guide does not attempt to reproduce the IEEE standard or explain every aspect of it. The interested user is referred to Ref 1. However, this User Guide provides all the information required to use the BBC Microcomputer and IEEE488 Interface with any other IEEE488 device.

Information covered and where to find it in this User Guide

Chapter 1 tells you how to connect the BBC Microcomputer's IEEE488 Interface to other IEEE488 devices and communicate with them. Chapter 2 expands on this and covers the exchange of binary coded data. Some IEEE488 devices have the ability to provide information on what state they are in, and ways of obtaining this information are discussed in Chapter 3. Chapter 4 contains details of commands which can be used to change the state of IEEE488 devices. The computer can pass control of the IEEE488 interface to another device and Chapter 5 tells how this is done. Chapter 6 is useful for reference as it provides a glossary of IEEE488 commands in alphabetical order. Information on using the interface in Assembler programs is contained in Chapter 7. Finally, Chapter 8 gives some details of the hardware used in the interface.

Throughout this User Guide examples are given in BASIC. However, the interface can be used with any language running on the computer which permits the use of a filing system. IEEEFS commands are given in full, and in upper case. However, abbreviations and any mix of lower and upper case may be used. See Appendix 1 for minimum abbreviations.

1 GETTING GOING - Simple data transfer

Before connecting devices to the computer turn off the power to both interface and computer.

The interface can be connected to up to 14 other devices using IEEE488 standard cable assemblies. These have a plug and socket at each end, permitting star or linear interconnection of devices. The connectors are provided with two securing screws which allow them to be stacked on the socket of each device. The IEEE socket is normally found on the rear of a device. The IEEE488 standard permits individual cable lengths of up to 4 metres and a total cable length in a system of 2 metres per device or 20 metres whichever is the shorter. Cables can be obtained from the manufacturers of devices.

When all devices have been connected to the interface, turn on the power in the following sequence:

- 1 devices
- 2 interface
- 3 computer

The IEEE Filing System (IEEEFS)

The IEEE488 interface of the BBC Microcomputer appears to the user as an additional filing system. It gives the computer the ability to act as a controller-talker or controller-listener with the ability to pass control and request it back through the use of IEEEFS commands. These commands enable the full use of an IEEE488 system without detailed knowledge of the operation of the interface.

The IEEEFS is selected by the command

```
*IEEE
```

To reselect another filing system use the appropriate command

```
*DISK
```

```
*TAPE etc
```

Note When used with BASIC the IEEEFS does not respond to

```
*CAT  
CHAIN  
EOF#  
EXT#  
LOAD  
PTR#  
SAVE
```

Communication between a language and the interface is via two channels
The first is the 'COMMAND' channel used for transmitting IEEEFS commands and receiving information on the state of the interface. The second is the 'DATA' channel used for reading and writing data to other IEEE488 devices.

After the first use of *IEEE the COMMAND and DATA channels must be opened.

Example

```
10 *IEEE  
20 cmd%=OPENIN("COMMAND")  
30 data%=OPENIN("DATA")
```

If required these channels can be closed in the following way

```
1010 CLOSE#data%  
1020 CLOSE#cmd%
```

Each device in an IEEE488 system must have a unique address. Some devices have only one address (a primary address) whilst others have extended addressing (primary and secondary addresses). Primary and secondary addresses may be in the range 0 to 30. Secondary addresses are often used to select different functions within the same device eg an analogue to digital converter with a number of inputs may have the input selected by the secondary address. Most devices are delivered with factory set primary and secondary addresses. These can be found in the manufacturer's manual. If more than one device in a system has the same address, these can be changed by following the manufacturer's instructions. Usually this involves changing switch settings on the rear of the device.

The OPENIN and OPENOUT functions are used to pass the primary and secondary, if specified, addresses of a device to the IEEEFS via a 'device' channel. Up to 14 channels may be open at one time.

Example

A digital voltmeter has a primary address of 4 and no secondary address.

```
100 dvm%=OPENIN("4")
```

A sixteen input analogue to digital converter has a primary address of 5 and a secondary address of 11 indicating the input to be converted.

```
150 adc%=OPENIN("5,11")
```

If no channel is available a zero is returned.

All channels remain open, even if other filing systems are selected, until specifically closed while the IEEEFS is active.

A 'device' channel may be closed using the CLOSE# statement.

Example

```
1110 CLOSE#dvm%
```

The CLOSE#0 statement closes all 'device' channels but not the COMMAND and DATA channels which must be closed individually.

As only 14 'device' channels may be open at one time, it is recommended that channels are closed when no longer required or at the end of a program.

The BBC Microcomputer must also have a unique device address. This is specified using the BBC DEVICE NO command and can be any number in the range 0 to 30 not already assigned to a device in the system.

Example

To specify the primary address of the computer to 7

```
40 PRINT#cmd%,"BBC DEVICE NO",7
```

The CLEAR command is now used to initialise the system.

Example

```
50 PRINT#cmd%,"CLEAR"
```

The REMOTE ENABLE command is used to Prepare devices for remote operation eg under computer control

Example

```
60 PRINT#cmd%,"REMOTE ENABLE"
```

Data strings of up to 255 characters

When strings of ASCII coded data are sent to or received from a device delimiting characters are used. Consult the manufacturer's manual for each device to find out the delimiting characters for that device. The computer defaults to the delimiting character Line Feed (LF, ASCII code 10). If a device requires different delimiting characters these may be specified using the END OF STRING command.

Example

To set delimiting characters to carriage return, line feed

```
80 PRINT#cmd%,"END OF STRING",CHR$(13)+CHR$(10)
```

The computer will then send these characters and expect to receive them after each data string until they are respecified.

Data Output

Channels must be opened to the devices concerned as previously described.

Before data can be sent to the devices, they must be listened, using the LISTEN command.

Example

```
160 voltmeter%=OPENIN("9,3")
180 PRINT#cmd%,"LISTEN",voltmeter%,printer%,"EXECUTE"
```

The EXECUTE command must follow the last device to be listened.

Data can then be sent to the devices through the DATA channel using the PRINT# statement.

Example

```
190 PRINT#data%,"RI"
```

would set the range of the voltmeter and print RI on the IEEE printer. When output has been completed, the devices can be unlistened.

Example

```
340 PRINT#cmd%,"UNLISTEN"
```

Data Input

Channels must be opened to the devices concerned as previously described.

Before a device can send data it must be talked using the TALK command.

Example

```
210 counter%=OPENIN("6")
220 PRINT#cmd%,"TALK",counter%
```

Note If there is already an active talker it will automatically be untalked.

Data is then received from the device via the DATA channel using the INPUT# statement. All active listeners will receive the data sent.

Example

```
230 INPUT#data%,count$
```

When input has been completed the device can be untalked.

Example

```
580 PRINT#cmd%,"UNTALK"
```

Checking the State of the Interface

The IEEEFS maintains a status word. This is available to the user through the STATUS command. The meaning of each bit of the status word is detailed in Chapter 6 under STATUS.

When ASCII coded data is being handled there are three status bits of interest.

- bit 4 if set indicates the input sequence was terminated because over 255 characters were received before the delimiting characters
- bit 6 if set indicates the END message was received from a device with the last character
- bit 7 if set indicates an incomplete source handshake. This is most likely to occur during a TALK or LISTEN sequence if no devices are connected to the interface

Example

```
1030 PRINT#cmd%,"STATUS"  
1040 INPUT#cmd%,state%  
1050 IF (state% AND 64) = 64 THEN PRINT"END received" ELSE PRINT"NO END  
received"
```

An Example of Data Input and Output

It is required to take three readings in the range 0 to 5v from a digital voltmeter with a primary address of 9 and print them on the computer's screen. The voltmeter sends the readings in the format +/- N.NNN CR LF

```
10 *IEEE  
20 cmd%=OPENIN("COMMAND")  
30 data%=OPENIN("DATA")  
40 PRINT#cmd%,"BBC DEVICE NO",0  
50 PRINT#cmd%,"CLEAR"  
60 PRINT#cmd%,"REMOTE ENABLE"  
70 PRINT#cmd%,"END OF STRING",CHR$(13)+CHR$(10):REM sets delimiting characters  
to CR LF  
  
80 voltmeter%=OPENIN("9")  
90 PRINTYkcmd%,"UNLISTEN"
```

```
100 PRINT#cmd%,"LISTEN",voltmeter%,"EXECUTE"  
110 PRINT#data%,"V5"  
120 PRINT#cmd%,"UNLISTEN"  
130 PRINT#cmd%,"TALK",Voltmeter%  
140 FOR I% = 1 TO 3  
150 INPUT#cmd%,voltage$(I%)  
160 PRINT voltage$(I%) :REM display on screen  
170 NEXT I%  
180 PRINT#cmd%,"UNTALK"  
190 CLOSE#Voltmeter%
```

Note In the above example the voltmeter had the same talk and listen addresses. This is usually, but not always, the case.

Detailed information on each command can be found in Chapter 6, and minimum abbreviations are in Appendix 1.

2 MORE ON DATA TRANSFER

Binary Coded Data

Sometimes it is necessary to send and receive more than 255 characters at a time without delimiting characters, so the PRINT# and INPUT# statements cannot be used.

Devices such as spectrum analysers, digital storage oscilloscopes and other microprocessor based devices with their own internal memory send and receive long fixed length sequences of binary coded data. The last byte is normally accompanied by the END end remote message. Data of this type can be sent and received using the WRITE BINARY and READ BINARY commands, together with the BPUT# and BGET# statements.

Example

To send 1024 data bytes held in an array to the memory of a spectrum analyser, primary address 11.

```
870 analyser%=OPENIN("11")
880 PRINT#cmd%,"LISTEN",analyser%,"EXECUTE"
890 PRINT#data%,"WTM",4000,1024" :REM prepare analyser to receive data and store
it in memory starting at location &4000
900 PRINT#cmd%,"WRITE BINARY",1024 :REM send the next 1024 bytes and accompany
the last byte with the END message
910 FOR I% = 1 TO 1024
920 BPUT#data%,wave%(I%) :REM output byte
930 NEXT I%
940 PRINT#cmd%,"UNLISTEN"
950 CLOSE#analyser%
```

Example

To receive 512 data bytes from the memory of a digital storage oscilloscope, primary address 15, and store them in an array.

```
670 scope%=OPENIN("15")
680 PRINT#cmd%,"LISTEN",scope%,"EXECUTE"
```

```

690 PRINT#data%,"CHI" :REM prepare oscilloscope to send data
700 PRINT#cmd%,"UNLISTEN"
710 PRINT#cmd%,"TALK",scope%
720 PRINT#cmd%,"READ BINARY",512 :REM prepare to receive 512 bytes
730 FOR I% = 1 TO 512
740 BGET#data%,trace%(I%) :REM input byte
750 NEXT I%
760 PRINT#cmd%,"UNTALK"
770 CLOSE#scope%

```

The TRANSFER Command

If it is required to exchange ASCII or binary coded data between devices without storing it in the memory of the computer, the TRANSFER command can be used. Two options are available.

a) TRANSFER with a count

Example

To print ten successive readings from a digital voltmeter, primary address 3, on an IEEE488 printer, primary address 9. The voltmeter sends the readings in the format +/- N.NNN CR LF ie 8 bytes per reading giving a count of 80 bytes.

```

540 printer%=OPENIN("9")
550 voltmeter%=OPENIN("3")
560 PRINT#cmd%,"LISTEN",printer%,"EXECUTE"
570 PRINT#cmd%,"TALK",voltmeter%
580 PRINT#cmd%,"TRANSFER",80
590 PRINT#cmd%,"UNTALK"
600 PRINT#cmd%,"UNLISTEN"
610 CLOSE#voltmeter%
620 CLOSE#printer%

```

b) TRANSFER without a count

This can only be used when the END message is sent with the last data byte to be exchanged. It has the advantage of leaving the computer free to continue with other asks.

Example

To send a wave form stored in the memory of a fast data capture unit, primary address 5, to the memory of a spectrum analyser, primary address 8, for analysis.

```
460 analyser%=GPENIN("8")
470 captureunit%=OPENIN("5")
480 PRINT#cmd%,"LISTEN",captureunit%,"EXECUTE"
490 PRINT#data%,"OUT",3000.1024" :REM prepare capture unit to output data
500 PRINT#cmd%,"UNLISTEN"
510 PRINT#cmd%,"LISTEN",analyser%,"EXECUTE"
520 PRINT#data%,"WTM",2000,1024" :REM prepare analyser in input data
530 PRINT#cmd%,"TALK",captureunit%
540 PRINT#cmd%,"TRANSFER",0 :REM transfer with no count

REM computer continuing with other tasks

910 REPEAT :REM read the status word until the END message is received
920 PRINT#cmd%,"STATUS" :REM causing the transfer in progress flag to be reset
930 INPUT#cmd%,state%
940 UNTIL (state% AND 8) = 8
```

3 HOW TO FIND OUT WHAT A DEVICE IS DOING

Serial Poll

If a device requires action from the computer, for example in order to rectify some error condition, it requests this by sending the SRQ (service request) message. When this happens bit 5 of the status word is set.

The SERIAL POLL command is used to determine which device has requested service. Each device with a serial poll capability must be polled. Consult the manufacturer's manual for each device to find out how many bytes it returns. Most devices return a single byte. Bit 6 of the first byte returned conveys the RQS (request service) message. If set, this indicates that the device was requesting service. The remaining bits of the first byte and ensuing bytes, if any, indicate the reason that service was requested. The meaning of this information can be found in the manufacturer's manual, and the appropriate action should be taken.

Example

```
270 PRINT#cmd%,"STATUS" 280 INPUT#cmd%,state%
290 IF (state% AND 32) = 32 THEN GOTO 5000

5000 counter%=OPENIN("`6'")
5010 signalgenerator%=OPENIN("`13'")
5020 PRINT#cmd%,"SERIAL POLL",counter%,1 :REM 1 byte expected
5030 INPUT#cmd%,counterstatus$
5040 IF ASC(counterstatus$) AND 64) = 64 THEN PROCservicecounter
5050 PRINT#cmd%."SERIAL POLL",signalgenerator%,1
5060 INPUT#cmd%,signalgeneratorstatus$
5070 IF (ASC(signalgeneratorstatus$)AND64)= 64 THEN PROCservicesignalgenerator
```

Because each device must be polled separately, a serial poll is time consuming.

Parallel Poll

This is a rapid way of finding out whether one or more devices require action by the computer. However the computer must be committed to carrying out a parallel poll periodically.

The PARALLEL POLL REQUEST command returns a single parallel poll response byte, each bit of which is assigned to a device or devices and, if set, indicates that the device requires action by the computer.

Devices with a parallel poll capability are assigned a bit in the parallel poll response byte using the PARALLEL POLL ENABLE command. This command uses an integer, the least significant 3 bits of which specify the associated bit of the parallel poll response byte as follows

b2	b1	b0	parallel poll response byte
0	0	0	0 0 0 0 0 0 0 1
0	0	1	0 0 0 0 0 0 1 0
0	1	0	0 0 0 0 0 1 0 0
0	1	1	0 0 0 0 1 0 0 0
1	0	0	0 0 0 1 0 0 0 0
1	0	1	0 0 1 0 0 0 0 0
1	1	0	0 1 0 0 0 0 0 0
1	1	1	1 0 0 0 0 0 0 0

Bit 3 of the integer specifies the 'sense' of the response. Refer to the manufacturer's manual to decide on its setting.

During a parallel poll request each enabled device compares its individual status with the value of the sense bit it received when it was enabled. If the comparison succeeds, the bit assigned to that device is set in the parallel poll response byte.

If a device is no longer required to take part in a parallel poll, the PARALLEL POLL DISABLE command is used. The PARALLEL POLL UNCONFIGURE command renders all devices unable to take part in a parallel poll.

Example

```
640 siggen%=OPENIN("5")
650 counter%=OPENIN("6")
660 analyser%=OPENIN("IO")
670 PRINT#cmd%,"PARALLEL POLL UNCONFIGURE" :REM to ensure no other devices take
part in parallel poll
680 PRINT#cmd%,"PARALLEL POLL ENABLE",siggen%,8 :REM sense bit = 1 ppr bit 0
690 PRINT#cmd%,"PARALLEL POLL ENABLE",counter%,9 :REM sense bit = 1 ppr bit 1
700 PRINT#cmd%,"PARALLEL POLL ENABLE",analyser%,2 :REM sense bit = 0 ppr bit 2
710 PRINT#cmd%,"PARALLEL POLL REQUEST"
720 INPUT#cmd%,responsebyte%
730 IF (responsebyte% AND 1) = 1 THEN PROCservicesiggen
740 IF (responsebyte% AND 2) = 2 THEN PROCservicecounter
750 IF (responsebyte% AND 4) = 4 THEN PROCserviceanalyser

1020 PRINT#cmd%,"PARALLEL POLL DISABLE",analyser%,"EXECUTE" :REM analyser no
longer required to take part in parallel poll
```

4 OTHER COMMANDS

The Timeout Facility

The TIMEOUT ON command can be used to enable a 2.5 sec timeout. This is useful in preventing a program from 'hanging up' when a device fails to respond. Bit 0 in the status word is set if a timeout occurs during output and bit 1 is set if a timeout occurs during input. If timeouts are no longer required the TIMEOUT OFF command is used.

Timeouts are not part of the IEEE488 standard and so the IEEEFS defaults to TIMEOUT OFF.

Example

```
500 PRINT#cmd%,"TIMEOUT ON"  
510 adc%=OPENIN("17,3")  
520 PRINT#cmd%,"TALK",adc%  
530 INPUT#cmd%,reading$  
540 PRINT#cmd%,"STATUS"  
550 INPUT#cmd%,state%  
560 IF(state% AND 2 = 2 THEN PROctimeoutoninput  
  
730 PRINT#cmd%,"TIMEOUT OFF"
```

Remote Local Operation

When a device is listened after a REMOTE ENABLE command control of the device passes from its front panel to the computer. The LOCAL LOCKOUT command prevents devices with this capability from inadvertently being returned to front panel control. If it is wished to return control to the front panel the GO TO LOCAL command is used. However, when the device is next listened, the front panel switches will again be locked out. To disable the locking out of the switches, the REMOTE DISABLE command should be used. This also has the effect of returning all devices to local control.

Example

```
110 PRINT#cmd%,"REMOTE ENABLE"  
120 voltmeter%=OPENIN("21")  
130 PRINT#cmd%,"LOCAL LOCKOUT"  
140 PRINT#cmd%,"LISTEN",voltmeter%,"EXECUTE" :REM voltmeter under control with  
front panel switches ineffective  
150 PRINT#data%,"R2" :REM set up voltmeter  
160 PRINT#cmd%,"UNLISTEN"  
170 PRINT#cmd%,"TALK",voltmeter%  
180 INPUT#data%,reading$  
190 PRINT#cmd%,"UNTALK"  
200 PRINT#cmd%,"GO TO LOCAL",voltmeter%,"EXECUTE" :REM voltmeter under front  
panel control  
  
460 PRINT#cmd%,"REMOTE DISABLE" :REM returns all devices to local control and  
disables front panel lockout  
470 CLOSE#voltmeter%
```

Initialising Devices

A device may have been set up incorrectly or an error may have occurred. Some devices have the capability of being returned to a known condition. The **SELECTED DEVICE CLEAR** command may be used to return individual devices to a known state or the **DEVICE CLEAR** command to return all devices with the capability to a known state.

Example

```
410 voltmeter%=OPENIN("6")  
  
530 PRINT#cmd%,"SELECTED DEVICE CLEAR",voltmeter%,"EXECUTE" :REM return  
voltmeter to known condition
```

```
790 PRINT#cmd%,"DEVICE CLEAR" :REM returns all devices with the capability to a  
known condition
```

Triggering Simultaneous Operation of Devices

Sometimes, for example in test systems, it may be required to start a process and commence measurements at the same time. This can be achieved using the TRIGGER command which sends the GET (group execute trigger) message.

Example

```
660 signalgenerator%=OPENIN("IO")  
670 recorder%=OPENIN("9")  
  
930 PRINT#cmd%,"TRIGGER",signalgenerator%,recorder%,"EXECUTE" :REM start  
measurement process
```

5 PASSING CONTROL

The system controller is the device which can send the IFC (interface clear) and REN (remote enable) messages, and only one is allowed in a system. The IEEEFS and the interface hardware is supplied configured so that the computer acts as system controller.

The controller in charge is the device which manages the system. This is not necessarily the system controller.

After the CLEAR command has been executed the computer is controller in charge. The TAKE CONTROL command can be used to pass management of the system to another device.

Example

To pass control to another computer with an IEEE488 interface, primary address 13

```
340 secondcontroller%=OPENIN("13")
350 PRINT#cmd%,.'TAKE CONTROL",secondcontroller%
```

After control has been passed to another device there are three ways in which the computer can regain management of the system.

- i) When the current controller in charge addresses the computer and sends the TCT (take control) message.
- ii) The REQUEST CONTROL command can be used to indicate to the current controller in charge that the BBC Microcomputer wishes to regain control of the system. It does this by sending the SRQ (service request) message and when polled by the current controller in charge during a serial poll sends the RQS message indicating it was requesting service together with a STB (status byte) of zero showing it wishes to regain control. There is no obligation on the current controller in charge to pass back control to the BBC Microcomputer. However, if it does, this occurs as described in the previous paragraph. To find out when the computer has regained control, bit 2 of the status word should be checked.

Example

```
870 PR'INT#Cmd%, "REQUEST CONTROL"  
880 REPEAT  
  
890 PRINT#cmd%, "STATUS"  
900 INPUT#cmd%, state%  
  
910 UNTIL (state% AND 4) = 4
```

- iii) through the use of the CLEAR command the BBC Microcomputer sends the IFC (interface clear) message. This instantly halts any current activity and returns the system to a known state. Data being transferred may be lost during this process. The computer is then controller in charge.

Example

```
100 PRINT#cmd%, "CLEAR"
```

6 THE IEEE FILING SYSTEM - A Summary

The IEEE488 Interface appears to the Operating System of the BBC Microcomputer as an additional filing system. It is selected by the command

```
*IEEE
```

The IEEEFS and the interface hardware is supplied so that the computer is system controller. The computer can act as a controller-talker controller- listener with the ability to pass control to another device and request it back.

Channels

Communication between a language and the IEEEFS is via two special channels. The first is a 'COMMAND' channel used for transmitting commands and receiving information on the state of the interface. The second is the 'DATA' channel used for reading and writing data.

Example

```
200 cmd%=OPENIN("COMMAND")
210 data%=OPENIN("DATA")
```

The IEEEFS maintains a status word, accessible to the user, which enables recovery from bus communication errors. When a command is received from the language via the COMMAND channel the IEEEFS interprets it and checks the status word. If it is satisfactory then the appropriate action is taken and, finally, the status word is updated.

The BBC DEVICE NO and CLEAR command should always follow the opening of the COMMAND and DATA channels.

Device Addresses

The OPENIN or OPENOUT functions are used to pass the primary and secondary, if specified, addresses of a device to the IEEEFs via a 'device' channel. Addresses must be in the range 0-30. Note that the primary address of the BBC Microcomputer is specified by the BBC DEVICE NO command.

Example

A voltmeter has a primary address of 4 and a secondary address of 2.

```
100 voltmeter%=OPENIN("4,2")
```

An analyser has a primary address of 11.

```
100 analyser%=OPENIN("11")
```

A 'device' channel may be closed using the CLOSE# statement.

Example

```
100 CLOSE#voltmeter%
```

```
100 CLOSE#0      closes all 'device' channels but not the 'COMMAND' and  
                  'DATA' channels which must be individually closed
```

Device Dependent Remote Messages (Data)

Strings – Input

Strings of up to 255 characters can be input by the IEEEFs and passed to the language via the 'DATA' channel. The INPUT# statement is used. The IEEEFs expects to receive the end of string characters after each string input from device. However, if the END end remote message is received before the end of string characters this terminates the input sequence.

Remote Messages

ATN

true		computer addresses itself as a listener
false	DAB 1	
false	DAB 2	
	etc	
false	DAB n	
false	EOS 1	end of string
false	EOS 2	and END characters
true		computer unaddresses itself as a listener

Example

```
100 PRINT#cmd%,"END OF STRING",CHR$(13)+CHR$(10)
110 PRINT#cmd%,"TALK",analyser%
120 INPUT#data%,response$
```

Strings – Output

Strings of up to 255 characters passed to the IEEFES via the 'DATA' channel can be output. The PRINT# statement is used. The IEEFES sends the UNT untalk remote message before the string and the end of string characters after each string it outputs to a device. It sends the END end remote message with the last character.

Remote messages

ATN

true	UNT	computer addresses itself as a talker
false	DAB 1	
false	DAB 2	
	etc	
false	DAB n	
false	EOS 1	end of string
false	EOS 2	and END characters
true		computer unaddresses itself as a talker

Example

```
100 PRINT#cmd%,"END OF STRING",CHR$(13)+CHR$(10)
110 PRINT#cmd%,"LISTEN",analyser%,"EXECUTE"
120 PRINT#data%,"AV4 RE"
```

Binary Coded Data

If this is to be handled, the READ BINARY and WRITE BINARY commands are used. Single data bytes are passed to and from the language via the 'DATA' channel. The BPUT# and BGET# statements are used. For further details see the READ BINARY and WRITE BINARY commands.

IEEEFS Commands

The following pages describe the IEEE commands under a number of headings.

Purpose

A plain English description of what the command does.

Example

An example to illustrate the use of the command.

Notes

Lists restraints regarding type and range of variables.

IEEE Status Required before Execution

Lists the status flags checked by the IEEEFS before carrying out a command.

IEEE Status after Execution

Lists the status flags changed by the IEEEFS after successfully carrying out a command.

Remote messages

Shows the logical state of the ATN remote message and the other remote messages sent by the IEEEFS.

Description

A brief description of the command and its effect.

Associated Commands

Similar commands and those which are concerned with the IEEE488 function.

BBC DEVICE NO

Purpose

To specify the primary address of the computer.

Example

```
100 *IEEE
110 cmd%=OPENIN("COMMAND")
120 data%=OPENIN("DATA")
    etc
200 PRINT#cmd%,"BBC DEVICE NO",6
```

Notes

Integer constants and variables in the range 0 - 30 may be used for the primary address.

IEEE Status Required before Execution

No requirement.

IEEE Status after Execution

The BBC DEVICE NO command executed flag is set.

Remote Messages

None

Description

The computer stores the primary address which is set by the CLEAR command. It requires a primary address so that, after control has been passed to another device using the TAKE CONTROL command, it may be addressed as a talker and control passed back to it. This normally follows a REQUEST CONTROL command.

Associated Commands

CLEAR

CLEAR

Purpose

To clear the interface.

Example

```
100 PRINT#cmd%,"BBC DEVICE NO",address%  
110 PRINT#cmd%,"CLEAR"
```

IEEE Status Required before Execution

The BBC DEVICE NO command executed flag must be set.

IEEE Status after Execution

The

- timeout on output flag is reset
- timeout on input flag is reset
- computer is controller in charge flag is set
- TRANSFER in progress flag is reset
- string of more than 255 characters received flag is reset
- SRQ service request remote message received flag is reset
- END end remote message received flag is reset
- incomplete source handshake flag is reset
- binary coded data being handled flag is reset
- a device has been TALKed flag is reset
- a device has been LISTENed flag is reset
- CLEAR command executed flag is set
- input error flag is reset
- output error flag is reset
- conflict for control of interface flag is reset
- computer in process of becoming controller in charge flag is reset
- computer in process of acknowledging an unidentified command from another device flag is reset
- computer in process of entering the controller active state following a TRANSFER command flag is reset
- REN remote enable remote message being sent flag is reset non fatal error flag is reset

Remote Messages

IFC

IFC false

Description

The primary address of the computer, as specified by the BBC DEVICE NO command, is set before sending the IFC interface clear remote message. This places the interface in a known quiescent state. The computer exits the command as controller in charge.

Associated Commands

BBC DEVICE NO

DEVICE CLEAR

Purpose

To clear all devices

Example

```
100 PRINT#cmd%,"DEVICE CLEAR"
```

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

No change

Remote Messages

ATN

true DCL

Description

After this command has been executed all devices with a device clear capability are initialised.

Associated Commands

SELECTED DEVICE CLEAR

END OF STRING

Purpose

To specify the characters delimiting device dependent messages.

Example

```
100 endchr$=CHR$(13)+CHR$(10)
110 PRINT#cmd%,"END OF STRING",endchr$
```

Notes

String constants and variables of 1 or 2 characters may be used to specify the end of string characters. The default character is LF (line feed).

IEEE Status Required before Execution

No requirement

IEEE Status after Execution

No change

Remote Messages

None

Description

The computer stores the End of String characters and sends them after each string of data bytes it outputs and expects to receive them after each string of data bytes it inputs.

GO TO LOCAL

Purpose

To return the referenced devices to local control.

Example

```
100 voltmeter%=OPENIN("9")
110 plotter%=OPENIN("5")
120 PRINT#cmd%,"GO TO LOCAL",voltmeter%,plotter%,"EXECUTE"
```

Notes

The last channel must be followed by the 'EXECUTE' command.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The device or devices have been LISTENed flag is reset.

Remote Messages

ATN

true UNIL
true MLA 1
true MSA 1 if specified
true MLA 2
true MSA 2 if specified
etc
true MLA n
true MSA n if specified
true GTL
true UNL

Description

After this command has been executed, devices with a remote capability referenced by the previously OPENINed channels are returned to local control.

Associated Commands

REMOTE ENABLE
REMOTE DISABLE
LISTEN
LOCAL LOCKOUT

LISTEN

Purpose

To address the referenced devices as listeners.

Example

```
100 voltmeter%=OPENIN("2,1")
110 printer%=OPENIN("5")
120 PRINT#cmd%,"LISTEN",voltmeter%,printer%,"EXECUTE"
```

Notes

Up to 14 devices may be LISTENed at any one time. The last channel must be followed by the 'EXECUTE' command.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
controller in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status After Execution

The device or devices have been LISTENed flag is set.

Remote Messages

ATN

```
true  MLA 1
true  MSA 1      if specified
true  MLA 2
true  MSA 2      if specified
      etc
true  MLA n
true  MSA n      f specified
```

Description

The computer sends the MLA, my listen address, and MSA, my secondary address if specified, remote messages of the devices referenced by the previously OPENINed channels.

Associated Commands

UNLISTEN

LOCAL LOCKOUT

Purpose

To prevent conflict between the remote and local operation of all devices.

Example

```
100 PRINT#cmd%,"LOCAL LOCKOUT"
```

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

No change

Remote Messages

ATN

True LL0

Description

After this command has been executed, devices with a local lockout capability may only be returned to local (front panel) from remote (interface) control by using the GO TO LOCAL command.

Associated Commands

REMOTE ENABLE

REMOTE DISABLE LISTEN

GO TO LOCAL

PARALLEL POLL DISABLE

Purpose

To disable the referenced devices so that they are unable to take part in a parallel poll

Example

```
100 analyser%=OPENIN("11")
110 voltmeter%=OPENIN("6")
120 PRINT#cmd%,"PARALLEL POLL DISABLE",analyser%,voltmeter%,"EXECUTE"
```

Notes

The last channel must be followed by the 'EXECUTE' command.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The device or devices have been LISTENed flag is reset.

Remote Messages

ATN

true UNL

true MLA 1

true MSA 1 if specified

true MLA 2

true MSA 2 if specified

etc

true MLA n

true MSA n if specified

true PPC

true PPD

true UNL

Associated Commands

PARALLEL POLL ENABLE

PARALLEL POLL UNCONFIGURE

PARALLEL POLL REQUEST

PARALLEL POLL ENABLE

Purpose

To enable the referenced device to take part in a parallel poll.

Example

```
100 spectrumanalyser%=OPENIN("9")
110 PRINTVpcmd%,"PARALLEL POLL ENABLE",spectrumanalyser%,10
```

Notes

The parallel poll response remote message is specified by the least significant 4 bits of an integer constant or variable.

Bit 3 specifies the sense of the response and Bits 0-2 specify the response.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The device or devices have been LISTENed flag is reset

Remote Messages

ATN

true UNL

true MLA

true MSA

true PPC

true PPE

true UNL

Associated Commands

PARALLEL POLL DISABLE

PARALLEL POLL UNCONFIGURE

PARALLEL POLL REQUEST

PARALLEL POLL REQUEST

Purpose

To conduct a parallel poll.

Example

```
100 PRINT#cmd%,"PARALLEL POLL REQUEST"  
110 INPUT#cmd%,pollresponse%
```

Notes

The response may only be returned to the least significant byte of an integer variable via the command channel.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
CLEAR command executed flag must be set

IEEE Status after Execution

No change

Remote Messages

ATN IDY
true true PPR

Description

If a TRANSFER is in progress then the computer waits until the current data byte has been transferred before conducting the PARALLEL POLL; it then restarts the transfer. In this way no data is lost.

If no TRANSFER is in progress the PARALLEL POLL is conducted immediately.

Associated Commands

PARALLEL POLL ENABLE
PARALLEL POLL DISABLE
PARALLEL POLL UNCONFIGURE

PARALLEL POLL UNCONFIGURE

Purpose

To render all devices unable to take part in a parallel poll.

Example

```
100 PRINT#cmd%,"PARALLEL POLL UNCONFIGURE"
```

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

No change

Remote Messages

ATN

true PPU

Associated Commands

PARALLEL POLL ENABLE

PARALLEL POLL DISABLE

PARALLEL POLL REQUEST

READ BINARY

Purpose

To enable the input of a specified number of data bytes.

Example

```
90 DIM reading%(5)
100 recarder%=OPENIN("5")
110 PRINT#cmd%,"TALK",recarder%
120 PRINT#cmd%,"READ BINARY",512
130 FORI%=0TO511
140 reading%(I%)=BGET#data%
150 NEXT
160 CLOSE#recarder%
```

Notes

Integer constants and variables in the range 1 to &7FFFFFFF may be used for the number of data bytes.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
a device has been TALKed flag must be set
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The binary coded data being handled flag is set.

Remote Messages

ATN

false DAB 1 data bytes received

false DAB 2

etc

false DAB n and END the END remote message is not required

true

Description

This command enables the input of long, in excess of 255 data bytes, binary coded device dependent remote messages. The computer flags binary coded data being handled in the IEEE status word and addresses itself as a listener. A single data byte is input in response to each BGET#. When the specified number of data bytes have been input, the computer resets the flag in the IEEE status word and unaddresses itself as a listener.

If the END end remote message is received during a READ BINARY command, this is flagged in the IEEE status word. However, it does not terminate the input.

Associated Commands

TALK

UNTALK

REMOTE DISABLE

Purpose

To return all devices to local control.

Example

```
100 PRINT#cmd%,"REMOTE DISABLE"
```

IEEE Status Required before Execution

The CLEAR command executed flag must be set.

IEEE Status after Execution

The REN remote enable remote message being sent flag is reset.

Remote Messages

REN false

Description

After this command has been executed all devices with a remote local capability are returned to local control, including those put in the local lockout state by a previous LOCAL LOCKOUT command.

Associated Commands

REMOTE ENABLE

LISTEN

LOCAL LOCKOUT

GO TO LOCAL

REMOTE ENABLE

Purpose

To enable the remote operation of devices.

Example

```
100 PRINT#cmd%,"REMOTE ENABLE"
```

IEEE Status Required before Execution

The CLEAR command executed flag must be set.

IEEE Status after Execution

The REN remote enable message being sent flag is set.

Remote Messages

REN

Description

After this command has been executed, the computer can change devices from local (front panel) control to remote (Interface) control using the LISTEN command.

Associated Commands

REMOTE DISABLE LISTEN

LOCAL LOCKOUT

GO TO LOCAL

REQUEST CONTROL

Purpose

To request that control be transferred back to the computer.

Example

```
100 PRINT#cmd%,"REQUEST CONTROL"
```

IEEE Status Required before Execution

The computer is controller in charge flag must be set
CLEAR command executed flag must be set
computer in process of becoming controller in charge flag must be unset

IEEE Status after Execution

No change

Remote Messages

ATN	SRQ	sent by current controller in charge
	STB, RQS and SRQ false	when polled by controller in charge
ATN		sent by computer
true		after TCT received from current controller in charge

Description

The computer sends the SRQ service request remote message and then exits the command. When the controller in charge conducts a serial poll the STB status byte and RQS request service remote messages are sent. The seven bits of the STB remote message are all 0. If the controller in charge sends the TCT, take control, remote message the computer becomes controller in charge and sets the appropriate flag in the IEEE status word.

Associated Commands

TAKE CONTROL
CLEAR

SELECTED DEVICE CLEAR

Purpose

To clear the referenced devices.

Example

```
100 plotter%=OPENIN("7")
110 counter%=OPENIN("3")
120 PRINT#cmd%,"SELECTED DEVICE CLEAR",plotter%,counter%,"EXECUTE"
```

Notes

The last channel must be followed by the 'EXECUTE' command.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The device or devices have been LISTENed flag is reset.

Remote Messages

```
ATN
true UNL
true MLA 1
true MSA 1      if specified
true MLA 2
true MSA 2      if specified
etc
true MLA n
true MSA n      if specified
true SDC
true UNL
```

Description

After this command has been executed the devices with the device clear capability referenced by the previously OPENINed channels are initialised.

Associated Commands

DEVICE CLEAR

SERIAL POLL

Purpose

To conduct a serial poll of the referenced device.

Example

```
100 logicanalyser%=OPENIN("7")
110 PRINT#cmd%,"SERIAL POLL",Iogicanalyser%,1
120 INPUT#cmd%,statusmessage$
```

Notes

Integer constants and variables in the range 1 to 255 may be used for the number of status bytes sent by the device to be polled. The status bytes may only be returned to a string variable via the command channel.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
controller in process of acknowledging an unidentified command from another device flag must be unset

IEEE Status after Execution

If the RQS, request service, remote message is received, the SRQ, service request, remote message received flag is reset
The device has been TALKed flag is reset
The device or devices have been LISTENed flag is reset

Remote Messages

ATN
true UNL
true SPE
true MTA
true MSA if specified
false STB 1 and RQS?
false STB 2
etc
false STB n
true SPD
true UNT

Description

Following the receipt of an SRQ, service request, remote message the SERIAL Poll command should be used to determine which device requires service.

Associated Commands

STATUS

STATUS

Purpose

To return information on the current state of the interface.

Example

```
100 PRINT#cmd%,"STATUS"  
120 INPUT#cmd%,state%
```

Notes

The status may only be returned to an integer variable via the command channel.

IEEE Status Required before Execution

No requirement

IEEE Status after Execution

No change

Remote Messages

None

Description

The individual bits of the status word indicate the following

Bit

0	timeout on output
1	timeout on input
2	computer is controller in charge
3	TRANSFER in progress
4	string of more than 255 characters received
5	SRQ service request remote message received
6	END end remote message received
7	Incomplete source handshake
8	binary coded data being handled
9	TIMEOUT ON
10	"COMMAND" file open
11	"DATA" file open
12	a device has been TALKed
13	a device or devices have been LISTENed
14	BBC DEVICE NO executed
15	CLEAR executed
16	input error
17	output error
18	conflict for control of interface
19	computer in process of becoming controller in charge
20	computer in process of acknowledging an unidentified command from another device
21	computer in process of entering the controller active state following a TRANSFER command
22	unused
23	REN remote enable remote message being sent
24	unused
25	unused
26	unused
27	unused
28	unused
29	unused
30	unused
31	non fatal error

TAKE CONTROL

Purpose

To pass control to another device.

Example

```
100 secondcontroller%=OPENIN("2'  
110 PRINT#cmd%,"TAKE CONTROL",secondcontroller%
```

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The computer is controller in charge flag is reset
device or devices have been LISTENed flag is reset
device has been TALKed flag is reset

Remote Messages

ATN
true UNL
true MTA
true MSA if specified
true TCT
false

Description

After this command has been executed the computer may regain control of the interface through REQUEST CONTROL or CLEAR commands.

Associated Commands

REQUEST CONTROL
CLEAR

TALK

Purpose

To address the referenced device as a talker.

Example

```
100 digitalvoltmeter%=OPENIN("2,1")
110 PRINT#cmd%,"TALK",digitalvoltmeter%
```

Notes

Only one device may be TALKed at any one time.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
controller in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The device has been TALKed flag is set.

Remote Messages

ATN

true MTA

true MSA if specified

Description

The computer sends the MTA, my talk address, and, if specified, the MSA, my secondary address, remote messages of the device referenced by the previously OPENINed channel.

Associated Commands

UNTALK

TIMEOUT OFF

Purpose

To disable timeouts.

Example

```
100 PRIN#cmd%,"TIMEOUT OFF"
```

Notes

Timeouts are not a part of the IEEE488 standard. TIMEOUT OFF is the default IEEE status.

IEEE Status Required before Execution

No requirement.

IEEE Status after Execution

The TIMEOUT ON flag is reset.

Remote Messages

None

Description

With TIMEOUT OFF, if a device fails to respond during a source or acceptor handshake, then the ESC key must be pressed to exit from a command.

Associated Commands

TIMEOUT ON

TIMEOUT ON

Purpose

To enable timeouts.

Example

```
100 PRINT#cmd%,"TIMEOUT ON"
```

Notes

Timeouts are not a part of the IEEE488 standard. TIMEOUT OFF is the default IEEE status.

IEEE Status Required before Execution

No requirement.

IEEE Status after Execution

The TIMEOUT ON flag is set.

Remote Messages

None

Description

After this command has been executed and a device fails to respond in 2.55 seconds during a source or acceptor handshake then the appropriate timeout error is flagged in the IEEE status word.

Associated Commands

TIMEOUT OFF

TRANSFER

Purpose

To initiate the transfer of data bytes between an active talker and all active listeners.

Example (a)

```
100 printer%=OPENIN("`4,2")
110 counter%=OPENIN("`9")
120 PRINT#cmd%,"LISTEN",printer%,"EXECUTE"
130 PRINT#cmd%,"TALK",counter%
140 PRINT#cmd%,"TRANSFER",1024
150 PRINT#cmd%,"UNTALK"
160 PRINT#cmd%,"UNLISTEN"
```

Example (b)

```
100 printer%=OPENIN("`4,2")
110 counter%=OPENIN("`9")
120 PRINT#cmd%,"LISTEN",printer%,"EXECUTE"
130 PRINT#cmd%,"TALK",counter%
140 PRINT#cmd%,"TRANSFER",0
150

190 REPEAT
200 PRINT#cmd%,"STATUS"
210 INPUT#cmd%,status488%
220 UNTIL (status488% AND 8) = 8
```

Notes

Integer constants and variables in the range to &7FFFFFFF may be used for the number of data bytes to be transferred.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
a device has been TALKed flag must be set
a device or devices have been LISTENed flag must be set

TRIGGER

Purpose

To initiate simultaneous operation of the referenced devices.

Example

```
100 generator%=OPENIN("9")
110 voltmeter%=OPENIN("14")
120 PRINT#cmd%,"TRIGGER",generator%,voltmeter%,"EXECUTE"
```

Notes

The last channel must be followed by the 'EXECUTE' command.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The device or devices have been LISTENed flag is reset.

Remote Messages

ATN

```
true  UNL
true  MLA 1
true  MSA 1      if specified
true  MLA 2
true  MSA 2      If specified
      etc
true  MLA n
true  MSA n      if specified
true  GET
true  UNL
```

Description

This command causes all devices with the device trigger capability referenced by the previously OPENINed channels to start some basic operation.

Associated Commands

None

UNLISTEN

Purpose

To unaddress all active listeners.

Example

```
100 PRINT#cmd%,"UNLISTEN"
```

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
controller in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The device or devices have been LISTENed flag is reset.

Remote Messages

ATN
true UNL

Description

The computer sends the UNL unlisten remote message to all active listeners.

Associated Commands

LISTEN

UNTALK

Purpose

To unaddress an active talker.

Example

```
100 PRINT#cmd%"UNTALK"
```

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
controller in process of acknowledging an unidentified command from another
device flag must be unset

IEEE Status after Execution

The device has been TALKed flag is reset.

Remote Messages

ATN

true UNT

Description

The computer sends the UNT untalk remote message to the active talker.

Associated Commands

TALK

WRITE BINARY

Purpose

To enable the output of a specified number of data bytes.

Example

```
100 analyser%=OPENIN("11")
110 PRINT#cmd%="LISTEN",analyser%,"EXECUTE"
120 PRINT#cmd%,"WRITE BINARY",1024
130 FOR 1% = 1 TO 1024
140 BPUT#data%,waveform%(1%)
150 NEXT
160 CLOSE#analyser%
```

Notes

Integer constants and variables in the range 1 to &7FFFFFFF may be used for the number of data bytes.

IEEE Status Required before Execution

The computer is controller in charge flag must be set
TRANSFER in progress flag must be unset
a device or devices have been LISTENed flag must be set
CLEAR command executed flag must be set
conflict for control of interface flag must be unset
computer in process of acknowledging unidentified command from another device
flag must be unset

IEEE Status after Execution

The binary coded data being handled flag is set

Remote Messages

```
ATN
true  UNT
false DAB 1          data bytes sent
false DAB 2
      etc
false DAB n          and END
true
```

Description

This command enables the output of long, over 255 data bytes, binary coded device dependent messages. The computer flags binary coded data is being handled in the IEEE status word and after sending the UNT untalk remote message addresses itself as a talker. A single data byte is output in response to each BPUT#. The END end remote message is sent with the last data byte. When the specified number of data bytes have been output the computer resets the flag in the IEEE status word and unaddresses itself as a talker.

Associated Commands

LISTEN
UNLISTEN

7 USING THE IEEE FILING SYSTEM IN ASSEMBLER

Read Chapter 43 of the BBC Microcomputer User Guide before writing assembler programs that use the IEEE488 interface. General information on using a filing system in assembler is presented there. This chapter gives details of the relevant operating system calls when using the IEEEFS.

OSFIND	&FFCE	Open or close a channel
OSGBP	&FFD1	Not supported by IEEEFS
OSBPUT	&FFD4	Send a single byte to channel from A
OSBGET	&FFD7	Receive a single byte from channel in A
OSARGS	&FFDA	Not fully supported by IEEEFS
OSFILE	&FFDD	Not supported by IEEEFS
OSWORD	&FFF1	Perform IEEE commands using parameter block to pass parameters
OSCLI	&FFF7	Interpret IEEEFS command line

Throughout this section the term file is used to maintain compatibility with Chapter 43 of the BBC Microcomputer User Guide. However, it should be noted that IEEEFS 'files' are not true files in that they have no storage capability and serve only to pass information to the IEEEFS.

OSFIND

Opens a file for reading and writing. The routine is entered at &FFCE which indirects through &21C. The value in A determines the type of operation.

A=&00 causes a file or files to be closed
A<>&00 causes a file to be opened

IF A<>0 then Y (high byte) and X (low byte) must contain the address of a location in memory which contains the file name terminated with CR (MD).

On exit A will contain the channel number allocated to the file for all future operations. If A = 0 then the IEEEFS was unable to open the file.

If A=0 then a file, or all files will be closed depending on the value of Y. Y=0 will close all files, otherwise the file whose channel number is in Y will be closed.

On exit C, N, V and Z are undefined and D=O. The interrupt state is preserved.

OSBPUT

This call is used to send data. It write (puts) a byte in A to the file previously opened using OSFIND. The routine is entered at &FFD4 which indirects through &218. On entry Y contains the channel number allocated by OSFIND.

On exit A, X and Y are preserved, N, V and Z are undefined and 0=0. The interrupt state is preserved.

OSBGET

This call is used to receive data. It gets (reads) a byte from a file into A. The file must have been previously opened using OSFIND and the channel number allocated must be in V. The routine is entered at &FF07 which indirects through &216.

On exit C=0 indicates a valid character in A. X and Y are preserved, N V and Z are undefined and 0=0. The interrupt state is preserved.

OSARGS

This routine is entered at &FFDA and indirects through &214. The IEEEFS only supports a call with A=&(0 and Y=&(0; this returns with A=U7 indicating that the IEEEFS is in use.

On exit X and Y are preserved, C, N, V and Z are undefined and D=0. The interrupt state is preserved.

OSWORD

This routine is entered at &FFF1 and indirects through &20C. IEEE commands are sent and information on the state of the interface received through the use of a single OSWORD call with A=&80. This fulfils the same function as the COMMAND channel. Y (high byte) and X (low byte) point to a parameter block. The structure of this is

first byte	no of bytes in argument
second byte	no of bytes in result
third byte	command code
subsequent bytes	as defined by the first three bytes

Details of each command are given in the table.

OSCLI

This call is used to select the IEEEFS by placing the address of the command line IEEE in V (high byte) and X (low byte) and calling &FFF7 which indirects through &2M. The command line should not start with an asterisk and must end with &OD. Any leading spaces will be stripped.

DATA

When data is sent using the OSBPUT and OSBGET calls, strings are transferred as type (&00) length and characters (in reverse order). When binary coded data is being handled it is sent and received a byte at a time.

Example

Typical sequence of operating system calls.

Select IEEEFS by calling OSCII

Open COMMAND and DATA channels by using OSFIND

Execute IEEE commands eg CLEAR, REMOTE ENABLE etc by calling OSWORD

Open 'device' channels by calling OSFIND

Listen or talk devices by calling OSWORD

Send and receive data by calling OSBPUT and OSBGET Close channels by calling OSFIND

Structure of OSWORD Block

Command	arg0	arg1	command	b0	b1	b2		
BBC DEVICE NO	1	0	&40	dev no				
CLEAR	0	0	&50					
DEVICE CLEAR	0	0	&62					
END OF STRING	1 or 2	0	&81	eos2	eos1			
GO TO LOCAL	n	0	&22	chn	"	ch2	ch1	
LISTEN	n	0	&20	chn	"	ch2	ch1	
LOCAL LOCKOUT	0	0	&61					
PARALLEL POLL DISABLE	n	0	&21	chn	"	ch2	ch1	
PARALLEL POLL ENABLE	2	0	&48	ch1	ppr			
PARALLEL POLL REQUEST	0	1	&02	ppr				
PARALLEL POLL UNCONFIGFURE	0	0	&60					
READ BINARY	4	0	&14	cnt3	cnt2	cnt1	cnt0	
REMOTE DISABLE	0	0	&71					
REMOTE ENABLE	0	0	&70					
REQUEST CONTROL	0	0	&64					
SELECTED DEVICE CLEAR	n	0	&24	chn	"	ch2	ch1	
SERIAL Poll	2	n	&44	ch1	pllcnt	bn	b1	
STATUS	0	4	&01	b3	b2	bi	bo	
TAKE CONTROL	1	0	&42	ch1				
TALK	1	0	&41	ch1				
TIMEOUT OFF	0	0	&52					
TIMEOUT ON	0	0	&51					
TRANSFER	4	0	&10	cnt3	cnt2	cnt1	cnt0	
TRIGGER	n	0	&28	chn	"	ch2	ch1	
UNLISTEN	0	0	&54					
UNTALK	0	0	&58					
WRITE BINARY	4	0	&18	cnt3	cnt2	cnt1	cnt0	

8 HARDWARE

The interface uses the Texas Instruments TMS 9914A GPIB Adapter, together with the SN75160A and SN75162A octal GPIB transceivers. The TMS 9914A interfaces the computer via the IMHZ bus to the IEEE488 bus. It relieves the computer's processor of the task of handling IEEE488 protocol when commands and data are sent and received. The TMS 9914A generates interrupts and so the processor does not have to spend time polling it.

As supplied, the hardware is configured as system controller, Molex link SI on interface PCB in position A, and with the IEEE488 data bus to be driven by open collector outputs, Molex link 52 on interface PCB in position C.

A 5 MHz crystal provides a clock for timing.

The TMS 9914A is connected to the IMHZ bus in accordance with the guidelines given in Ref 2. The interface is supplied with four terminating resistor packs. They should be removed if the interface is not to be the last device on the IMHZ bus.

The read and write registers of the interface are located at addresses &FC20 to &FC27.

The user who wishes to program the TMS 9914A directly in order to configure the computer as a simple talker-listener should obtain Ref 3, and change the Molex link SI on the interface PCB from position A to position B.

REFERENCES

- Ref 1 IEEE Std 488-1978
IEEE488 Standard Digital Interface for Programmable Instrumentation
The Institute of Electrical and Electronic Engineers Inc 1978
- Ref 2 Application Note No 1
BBC Microcomputer IMHZ Bus
Acorn Computers ltd 1982
- Ref 3 TMS 9914 GPIB Adapter Data Manual
Texas Instruments

APPENDIX 1

IEEE Commands

	Minimum Abbreviation
BBC DEVICE NO	B.
CLEAR	C.
DEVICE CLEAR	D.
END OF STRING	E.
EXECUTE	E.
GO TO LOCAL	G.
LISTEN	L.
LOCAL LOCKOUT	L0.
PARALLEL POLL DISABLE	PARALLEL POLL D.
PARALLEL POLL ENABLE	PARALLEL POLL E.
PARALLEL POLL REQUEST	P.
PARALLEL POLL UNCONFIGURE	PARALLEL POLL U.
READ BINARY	R.
REMOTE DISABLE	REMOTE D.
REMOTE ENABLE	REM.
REQUEST CONTROL	REQ.
SELECTED DEVICE CLEAR	SEL.
SERIAL POLL	SE.
STATUS	S.
TAKE CONTROL	TAK.
TALK	T.
TIMEOUT OFF	TIMEOUT OFF
TIMEOUT ON	TI.
TRANSFER	TR.
TRIGGER	TRI.
UNLISTEN	UNI.
UNTALK	U.
WRITE BINARY	W.

Note - Standard IEEE488 mnemonics can be used as string variable names.

Example

```
100 PPU$="PARALLEL POLL UNCONFIGURE"  
110 PPE$="PARALLEL POLL ENABLE"  
120 PPR$="PARALLEL POLL REQUEST"  
130 PPD$="PARALLEL POLL DISABLE"  
  
650 spectrumanalyser%=OPENIN("9")  
660 PRINT#cmd%,PPU$  
670 PRINT#cmd%,PPE$,spectrumanalyser%,10  
  
940 PRINT#cmd%,PPR$  
950 INPUT#cmd%,response%  
  
990 PRINT#cmd%,PPD$,spectrumanaiyser%,"EXECUTE"
```

APPENDIX 2

IEEEFS ERRORS

Error number

&DC	syntax
&88	out of range
&DE	channel
&8A	no device listened type mismatch
&8B	no BBC DEVICE NO
&8D	string too long
&89	no device talked
&8c	CLEAR not executed
&FE	Bad command

APPENDIX 3

INTERFACE REMOTE MESSAGES

List of Messages Referenced in this Guide

Mnemonic

ATN	attention
DAB	data byte
DCL	device clear
END	end
EOS	end of string
GET	group execute trigger
GTL	go to local
IDY	identify
IFC	interface clear
LLO	local lockout
MIA	my listen address
MTA	my talk address
MSA	my secondary address
PPC	parallel poll configure
PPE	parallel poll enable
PPD	parallel poll disable
PPR	parallel poll response
PPU	parallel poll unconfigure
REN	remote enable
RQS	request service
SDC	selected device clear
SPD	serial poll disable
SPE	serial poll enable
SRQ	service request
STB	status byte
TCT	take control
UNL	unlisten
UNT	untalk

APPNDIX 4

INTERFACE SPECIFICATION - note 1

SH	source handshake function
SH1	complete capability
AH	acceptor handshake function
AH1	complete capability
T	talker function - note 2
T5	complete capability
L	listener function - note 2
L3	complete capability
SR	service request
SR1	complete capability
RL	remote local function
RL0	no capability
PP	parallel poll function
PP0	no capability
DC	device clear function
DC0	no capability
DT	device trigger function
DT0	no capability
C	control function
C1, C2, C3	complete capability
C4 and C5	

Note 1 - This specification applies to the IEEEFS and interface hardware configured as system controller.

Note 2 - The interface only sends and receives data when the computer addresses itself to talk or listen, not when addressed by other devices.

Power supply 220-240VAC 50-60Hz 3 watt
Manufactured in accordance with BS415/79 Class 1

Designed and manufactured to comply with Acorn Computers Ltd 1MHZ Bus Peripheral Specification.

APPENDIX 5

FITTING THE IEEEFS EPROM

CAUTION - STATIC SENSITIVE DEVICES
DO NOT HANDLE UNTIL YOU HAVE READ THESE INSTRUCTIONS

Please read these instructions carefully before attempting this conversion. If you are not completely confident of your ability to carry out this conversion yourself it will be in your own interests to take this unit to an authorised BBC Microcomputer dealer for him to carry out this upgrade.

- 1 Determine which operating system is fitted in your computer by typing *FX 0 and noting the version number displayed
- 2 Switch off and disconnect the computer from the mains supply

WARNING

DO NOT REMOVE THE LID OF THE COMPUTER WHILST IT IS ONNNECTED TO THE MAINS

- 3 Unscrew the two large-headed screws on the rear panel of the computer and the two large-headed screws on the underside of the computer (near the front). Carefully remove the lid.
- 4 Unscrew the two (or three on some computers) bolts securing the keyboard assembly to the lower case and CAREFULLY unplug the 17-way ribbon cable connecting the keyboard assembly to the main Printed Circuit Board (PCB) by pulling the socket away from PLI3 on the main PCB.
- 5 Move the keyboard away from the main PCB until the loudspeaker plug is accessible. Unplug the loudspeaker by CAREFULLY removing the socket from PLI5. Place the keyboard assembly to one side on a flat surface.
- 6 If your computer contains an operating system of version 1.00 or later proceed to instruction 9
- 7 The current Machine Operating System (MOS) is situated in the bottom right-hand corner of the main PCB in one of two formats as follows:
 - a) One integrated circuit (IC) in socket 1C51 identified by the legend "Bo2.1 printed on top. If you have this format carefully remove the MOS IC from 1C51 using an IC extractor (a small screwdriver used with extreme care will do).

- b) Four ICs in sockets IC52, IC88, IC100 and IC101 (these sockets are all next to each other despite the numbering) identified with the legends "C", "D", "E" and "F" respectively printed on the top. If this is the case the BASIC language IC (identifiable by the legend "BO1" printed on the top) and all four MOS ICs should be removed as described in 'a' above. Replace the BASIC IC in socket IC100 as described in '8'.
- 8 The new MOS is supplied in a single IC identified by the legend "B04" printed on the top. Insert the ROM carefully into the socket IC51 ensuring that all the 'legs' locate properly into the holes and are not 'bent under' upon insertion. The IC should be orientated so that the semi-circular cutout in the top of the IC is at the top, i.e. away from the keyboard. Avoid touching the IC legs with your fingers wherever possible.

Now set the following links:

518 North
519 East
S20 North
521 Two off East/West
S22 North
S32 West
S33 West

N.B. North is the edge of the PCB nearest to the back of the case and South is the edge nearest to the keyboard. East and West follow logically from these.

- 9 Insert the IEEEFs EPROM IC into any free socket (IC88, IC100 or IC101) as described in 8. If you have a disc filing system (DFS) ensure that the IEEEFs EPROM is inserted to the right of the DFS IC.
- 10 Reassemble the BBC Microcomputer by reversing instructions 2 to 5.
- 11 To check that the IEEEFs EPROM has been fitted correctly type

***HELP IEEE**

The computer should display

```
IEEE Filing System <version number>  
O.S. <version number>
```

If it does not disconnect the computer from the mains supply and ensure that all the instructions have been followed correctly. Reconnect to the supply and try again. If you still have problems contact an authorised dealer.

APPENDIX 6

REMOVAL OF TERMINATING RESISTOR PACKS

The interface is supplied with four terminating resistor packs. They should be removed following the instructions below if the interface is not to be the last device on the IMHZ bus.

- 1 Switch off the interface and disconnect it from the mains supply
WARNING - DO NOT REMOVE THE LID OF THE INTERFACE WHILST IT IS CONNECTED TO THE MAINS
- 2 Disconnect the IMHZ Bus connecting cable from the computer or other 1MHz bus peripheral.
- 3 The two halves of the case may be separated by undoing three fixing screws two on the rear panel and one on the underside in the centre at the front. When reassembling push the two halves together whilst tightening the two rear fixing screws. Take care not to lose the two captive nuts for the rear fixing screws which are pushed onto the upper moulding.
- 4 The four resistor packs, RPI, RP2, RP3 and RP4, are located on the PCB towards the front of the case. Note the orientation of the resistor packs for future reference. Carefully remove them using an IC extractor (a small screwdriver used with extreme care will do) and store them safely so that they can be refitted in the future should the interface become the last peripheral on the IMHZ bus.
- 5 Reassemble by following steps 2 and 3 in the reverse order.

INDEX

Abbreviations Appendix 1
Assembler 60

BBC DEVICE NO 5, 21, 26
BGET# 10, 11
BPUT# 10

CLEAR 6, 19, 20, 21, 27
CLOSE 5
COMMAND channel 4, 21
Controller in charge 19

DATA channel 4, 21
Data input 7, 10, 22, 62
Data output 6, 10, 23, 62
Delimiters 6
DEVICE CLEAR 17, 29

END message 8
END OF STRING 6, 30
EOI 8
EPROM - fitting Appendix 5
EXECUTE 7

GO TO LOCAL 16, 17, 31
Group execute trigger 18, 55

Hardware 64

IEEE488 - standard 1
- *IEEE 3, 21
- IEEEFS 3, 60

INPUT# 7

LISTEN 6, 33
LOCAL LOCKOUT 16, 17, 34

OPENIN 5
OPENOUT 5

OSARGS 61
OSBGET 61
OSBPUT 61
OSCII 62
OSFIND 60
OSWORD 61, 63

PARALLEL POLL 14
-DISABLE 14, 15, 35
-ENABLE 14, 15, 36
-REQUEST 14, 15, 37
-UNCONFIGURE 14, 15, 38

Primary address 4, 22
PRINT# 7

READ BINARY 10, 11, 39
REMOTE DISABLE 16, 17, 41
REMOTE ENABLE 6, 16, 42
REQUEST CONTROL 19, 43

Secondary address 4, 22
SELECTED DEVICE CLEAR 17, 44
SERIAL POLL 13, 45
Source handshake 8
STATUS 8, 47, 48
System controller 19

TAKE CONTROL 19, 45
TALK 7, 50
Terminating resistors Appendix 6
TIMEOUT OFF 16, 51
TIMEOUT ON 16, 52
TRANSFER 11, 12, 53, 54
TRIGGER 18, 55

UNLISTEN 7, 56
UNTALK 7, 57

WRITE BINARY 10, 58, 59

ACORN COMPUTERS LTD
Fulbourn Road
Cherry Hinton
Cambridge
CB 1 4JN