brain boxes

P.S.I. BOX

PARALLEL, SERIAL AND IEEE INTERFACE

4.2 EDITION August 1995

GUARANTEE.

FULL 36 MONTHS GUARANTEE.

BRAIN BOXES guarantee your PSI BOX for a full 36 months from purchase, parts and labour, provided it has been used in the specified manner. In the unlikely event of failure return your interface to BRAIN BOXES or to your Dealer, with proof of purchase, who will determine whether to repair or replace this product with an equivalent unit.

COPYRIGHT.

PSI BOX is designed, manufactured and supported by

BRAIN BOXES.

Unit 3F, Wavertree Boulevard South,
Wavertree Technology Park,
Liverpool, L7, 9PF,
England.
Telephone: 0151-220 2500.
Fax No: 0151-252 0446

ACKNOWLEDGEMENTS.

BRAIN BOXES is a trademark of BRAIN BOXES.
IBM, COMPAQ and Hewlett Packard are trademarks of the relevant companies. OS/2 and Micro Channel Architecture are trademarks of IBM. Windows is a trademark of Microsoft.

Thank You For Buying The PSI BOX!

We proudly present the PSI BOX.

The PSI BOX is a very flexible 3 way converter. The PSI BOX converts signals between IEEE488, RS232C and Centronics Parallel. The PSI BOX can allow the signals to be passed between any combination of the different ports. The PSI BOX is the one stop solution to your interfacing problems.

The PSI BOX PLUS+ allows any computer with a serial port to become an IEEE bus controller, exercising complete control over a whole bus full of IEEE instruments. The built in industry standard IEEE command language gives the user of any computer, PC, PS/2, micro, mini or mainframe precise control over the huge range of industry standard IEEE 488 devices.
The PSI BOX PLUS+ includes a disk with two sample communications programs for use in Mode 7.

Your interface is designed and manufactured in England, and our policy is one of complete support to our dealers and direct to our users. The PSI BOX is designed ’in house’ and is completely understood by our staff. Its great strength is the support we give it. Our intention is to supply the software and any technical information you may need to allow you to exercise complete control over the PSI BOX allowing you to interface many different types of equipment with each other.

After searching the manuals, do not hesitate to contact us on our HOTLINE number given on page Intro-2, if you need help.

We trust that if you adhere to the following instructions you will enjoy many years of useful service from your interface.

PSI BOX MANUAL

The Layout Of This Manual.

This manual is a complete description of how to install and use the PSI BOX and PSI BOX PLUS+ interfaces.

The PSI BOX is a very flexible 3 way converter. The PSI BOX converts signals between IEEE488, RS232C and Centronics Parallel. The PSI BOX can allow the signals to be passed between any combination of the different ports. The PSI BOX allows data to be transferred between the popular communication standards available today. The PSI BOX is thus the one stop solution to your interfacing problems.

The PSI BOX PLUS+ is an enhanced version of the basic PSI BOX. The PSI BOX PLUS+ includes an extra mode as standard Mode 7. This mode allows any computer with an RS232 port to act as the controller of a whole bus full of IEEE instruments. Mode 7 has a powerful but easy to understand industry standard IEEE control language built in that is designed to make instrument control as flexible and as straight forward as possible.

We have harnessed the power of the RS232 port, the parallel port and the GPIB BUS allowing you to get the most out of your IEEE devices!

The PSI BOX can be customised to suit your individual needs. Whether it means including a new mode to interconnect your peripherals or whether it means adding a specialised port to connect to a non standard device we are happy to quote a cost and delivery designed to suit you!

Chapter 1, Setting Up The PSI BOX, introduces the PSI BOX and explains the function of the various modes and instructs you how to configure the PSI BOX to suit your needs. The use of the Buffer On switch is explained.

Chapter 2, An Introduction To The IEEE-488 Interface Bus Standard, is a general guide to this widely used international standard. It explains how the different IEEE devices always know who is sending information and who is receiving information. It explains what is expected of the controller of the IEEE bus. It explains the general concepts involved in understanding, and using, the IEEE-488 Bus.

Chapter 3, The IEEE Port, gives details of the IEEE connector pin outs, the function of IEEE DIP switches and LEDs.

Chapter 4, The Centronics Parallel Port, gives details of the parallel port connector pin outs, the function of Centronics Output DIP switch and the Centronics LEDs, the parallel port input and output handshake.

Chapter 5, The RS232 Serial Port, gives details of the serial port connector pin outs, the function of RS232 DIP switches and LEDs, how to set the baud rate, parity and RS232 handshake options. Details of how to connect the PSI BOX serial port to both an IBM PC and a modem are included.

Chapter 6, The PSI BOX Modes 1-6 & 8, explains the differences between these operating modes of the PSI BOX. This includes a description of how the data transfer is performed. Typical applications of each mode are given.
Modes 1-6 are typically used to interconnect IEEE printers and plotters to computers with only serial or parallel ports. Or to connect IEEE-488 computers to serial and parallel

devices such as laser printers and modems.

Mode 8 is used at allow an IEEE plotter than uses Hewlett Packard Graphic Language to completely emulate the RS232 equivalent plotter. HPGL plotting commands, HPGL output commands and RS232 flow control being catered for.

Chapter 7, The PSI BOX Mode 7 Overview, contains as a short summary of the Mode 7, details about the Mode 7 power up messages, the Mode 7 default parameters, data pass through mode, the BUS command sequence, sending data to IEEE devices, receiving data from IEEE devices, IEEE control status string, the RESET command and character, the current device, a summary of the BUS command language and finally the Mode 7 memory allocation.

Mode 7 is typically used as a means of controlling a whole bus full of IEEE instruments from the RS232 serial port of any computer.

Chapter 8, The IEEE BUS Command Language Controls and Functions, lists and explains the full range of IEEE controls and functions currently supported by the enhanced PSI BOX PLUS+ Mode 7 IEEE command language. This chapter details the command syntax, parameters and actual IEEE bus transactions that take place. The IEEE functions are a comprehensive set of commands that allow complete control of the IEEE bus, right from the individual line logic levels to performing complete parallel or serial poll routines on multiple devices. Knowing how to access these commands is the key to realising the full benefits of the powerful software that makes the PSI BOX PLUS+ a magnificent asset.

The Cumulative Index covers the complete contents of the manual.

What Machine Do You Need?

The PSI BOX is designed to work with as wide a range of computers IEEE devices and peripheral as possible.

Of course, we have not tried EVERY IEEE device and EVERY computer available but we’ve yet to find one that can not be made to work! Any comers should phone us on our HOTLINE and we will see what we can do!

Outline Contents.

Chapter 1. **Setting Up The PSI BOX.**
Chapter 2. **An Introduction to the IEEE 488 Interface Bus Standard.**
Chapter 3. **The IEEE Port.**
Chapter 4. **The Centronics Parallel Port.**
Chapter 5. **The RS232 Serial Port.**
Chapter 6. **The PSI BOX Modes 1-6 & 8 With Typical Application Examples.**
Chapter 7. **The PSI BOX Mode 7 Overview**
Chapter 8. **The IEEE BUS Command Language Controls and Functions.**

Cumulative Index.

Contents

Chapter 1 Setting Up The PSI BOX.
Introduction. 1
The PSI BOX. 1
The PSI BOX Modes. 3
The Buffer Switch. 4
Setting It All Up. 6
Configuring The PSI BOX. 6
Problems! 8

Chapter 1 List Of Figures.
Figure 1-1. The Buffer Switch. 4
Figure 1-2. Buffer Option. 5
Figure 1-3. The Buffer Allocation. 5
Figure 1-4 The Power LED. 6

Chapter 2 An Introduction to IEEE 488 Bus Standard.
Introduction. 9
The IEEE 488 Standard. 9
The Beginnings Of A New Standard. 9
What Exactly Is IEEE 488? 10
What's In A Name? 10
IEEE 488-1975. 11
The IEEE Signal Lines. 12
The IEEE Signal Lines. 12
The Data Bus Signals. 12
The Handshake Signals. 12
DAV Data Valid. 13
NRFD Not Ready For Data. 13
NDAC Not Data Accepted. 13
The Management Signals. 13
ATN Attention. 14
IFC Interface Clear. 14
REN Remote Enable. 14
SRQ Service Request. 14
EOI End Or Identify. 15
Negative Logic 15

Cabling Lengths and Data Transfer Speeds. 15
Using The IEEE Bus. 16
Types Of IEEE Devices. 16
The Listener. 16
The Talker. 16
The Controller. 17
The Controller In Charge. 17
The System Controller. 17
Controller Summary. 18
Command Mode And Data Mode. 18
Addresses. 19
Talk Only And Listen Only Devices. 20
Serial Poll. 20
Parallel Poll. 21
Device Clear and Device Trigger. 22
Remote - Local Functions. 23
End Of Sequence Termination, 23

Chapter 3 The IEEE Port.

Introduction. 25
The IEEE Leds. 26
The IEEE DIP Switches. 27
Setting The IEEE Address. 28
Setting The Controller State. 28
IEEE Cables. 29

Chapter 3 List Of Figures.

Figure 3-1. The IEEE Leds. 27
Figure 3-2. The IEEE DIP Switches. 27
Figure 3-3. Setting The IEEE Address. 28
Figure 3-4. The Controller State. 29
Figure 3-5. The IEEE Port Pin Outs. 30

Chapter 4 The Centronics Parallel Port.

Introduction. 31
Centronics Leds. 32
Selecting The Direction Of The Centronics Port. 32

The Centronics Output Handshake. 33
The Centronics Input Handshake. 34
Centronics Cables. 35

Chapter 4 List Of Figures.

Figure 4-1. The Centronics Leds. 32
Figure 4-2. Setting Parallel Port Direction. 33
Figure 4-3 The Centronics Output Port Pin Outs. 35
Figure 4-4. The Centronics Input Port Pin Outs. 36
Figure 4-5. Centronics Input From PC Printer Port. 36

Chapter 5 The RS232 Serial Port.

Introduction. 37
RS232 Leds. 38
The RS232 DIP Switches. 39
Setting The Baud Rate. 39
The RS232 Handshake. 39
The Hard Wired Handshake. 40
The XON/ XOFF Handshake. 41
Setting The Parity Options. 43
Setting The Data Word Length. 43
RS232 Port Cables. 44
Connecting To An IBM PC Serial Port. 44
Connecting To A Modem. 45
Loop Back Connector. 46

Chapter 5 List Of Figures.

Figure 5-1. The RS232 Leds. 38
Figure 5-2. The RS232 DIP Switches. 38
Figure 5-3. Baud Rate Settings. 39
Figure 5-4. Selecting The RS232 Handshake. 40
Figure 5-5. Parity Option. 42
Figure 5-6. The Data Word Length. 43
Figure 5-7. The RS232 Port Pin Outs. 45
Figure 5-8. The Loop Back Connector. 46
Figure 5-10. PSI Box To PC Serial Port Cable. 46
Figure 5-9. Connecting To The RS232 Port. 47
Figure 5-11. PSI Box To AT Serial Port Cable. 47

Figure 5-12. IBM AT To PC Serial Port Cable. 48

Chapter 6 Modes 1-6 & 8 With Application Examples.

Introduction. 49
The Mode Switches. 49
Mode 1. 51
Mode 2. 54
The Mode 1+2 IEEE Handshake. 56
The Mode 1+2 Secondary Addresses. 56
Mode 3. 58
Mode 4. 59
Mode 5. 60
Mode 6. 62
Mode 8. 64

Chapter 6 List Of Figures.

Figure 6-1 The Mode DIP Switches Settings. 50
Figure 6-2 HPGL Output Commands. 65
Figure 6-3 HPGL RS232 I/f Flow Control Commands. 66

Chapter 7 The PSI BOX Mode 7 Overview

Introduction. 68
Mode 7 In 15 Minutes!. 68
Mode 7 Summary. 69
Mode 7 Power Up. 70
The Default Mode 7 Configuration. 71
Data Pass Through Mode. 72
The BUS Command Sequence. 73
Sending Data Using The Bus Command. 75
Receiving Data Using The Bus Command. 75
The IEEE Control Status String. 76
The RESET Command and Character. 79
The Current Device. 79
Mode 7 BUS Command Summary. 81
Mode 7 Memory Allocation. 82

Chapter 7 List Of Figures.

Figure 7-1 The Default Mode 7 Configuration.		72
Figure 7-2 The IEEE Control Status String.		78
Figure 7-3 Mode 7 IEEE Command Primitives		81
Figure 7-4 Mode 7 IEEE Line Controls.		81
Figure 7-5 Mode 7 IEEE Command Functions.		82
Chapter 8 The IEEE BUS Command Language		
Introduction.		84
Valid Parameters.		84
ABORT.	Terminate Activity, Reset Devices.	87
ATN.	Assert The Atn Line.	88
CLEAR.	Resets All or Specified Devices.	89
CMD.	Send A Byte With Attention True.	90
CONFIGURE.		
	Specifies Devices Parall Poll Response.	92
DATA.	Send A Byte With Attention False.	94
DCL.	Sends The Device Clear Command.	95
DISABLE.	Prevents Devices Responding Parallel Poll.	96
END=.	Set The End Of Output Method.	97
END=OFF.	Disable EOI During A Bus Write.	99
END=ON.	Set The End Of Output Method.	100
EOI.	Send A Data Byte With Eoi True.	101
EOS=.	Set End Of Sequence Character For Inputs.	102
GET.	Sends Group Execute Trigger.	104
GTL.	Sends The Go To Local Command.	105
IFC.	Pulse The IFC Line.	106
LAG.	Sends The Listen Address.	107
LISTEN.	Sends The Listen Address.	108
LLO.	Sends The Local Lockout Command.	109
LOCAL.	Set Some Or All Devices To Local Control.	110
LOCKOUT.	Sends The Local Lockout Command.	111
MACRO.	Defines Or Lists Macro.	112
MESSAGES=OFF.		
	Disable PSI BOX Messages.	114
MESSAGES=ON.		
	Enable PSI BOX Messages.	115
MLA.	Sends My Listen Address.	116
MLA=.	Sets Or Changes PSI BOX Listen Address.	117

MSA.	Sends My Secondary Address.	118
MSA=.	Set/ Change PSIBOX Secondary Address.	119
MTA.	Sends My Talk Address.	120
MLA=.	Sets Or Changes PSI BOX Listen Address.	121
NO ATN.	Release The Atn Line.	122
NO REN.	Release The Ren Line.	123
NO TO.	Disable The Timeouts.	124
OUTPUT.	Send A String Of Data To A Bus Device.	126
PPC.	Sends Parallel Poll Configure.	127
PPD.	Sends Parallel Poll Disable.	129
PPE.	Sends Parallel Poll Enable Command.	131
PPOLL.	Perform A Parallel Poll.	133
PPU.	Sends Parallel Poll Unconfigure.	135
READ.	Receive One Byte From Addressed Talker.	136
REMOTE.	Put Device Under PSI BOX Control.	137
REN.	Assert The Ren Line.	138
RESET.	Specifies The Power On Reset Character.	139
SEC.	Sends The Secondary Address.	140
SDC.	Sends Selective Device Clear Command.	141
SHOW.	Display & Explain Control Status String.	142
SPD.	Sends Serial Poll Disable Command.	143
SPE.	Sends Serial Poll Enable Command.	144
SPOLL.	Perform A Complete Serial Poll.	146
STATUS.	Send The Status String To RS232 Host.	148
STRING.	Send A String Of Data To A Bus Device.	150
TAD.	Sends The Talk Address.	151
TALK.	Sends The Talk Address.	152
TCT.	Sends The Take Control Command.	153
TIMER.	Wait For X Milli Seconds.	154
TO.	Set The Timeout Period.	155
TRIGGER.	Tell Devices Perform Predefined Action.	157
Unconfigure.	Prevents Devices Responding To PPoll.	158
UNL.	Sends The Unlisten Command.	159
UNT.	Sends The Untalk Command.	160
VERSION.	Send PSI Rom Version To RS232 Host.	161
WAIT.	Wait For X Milli Seconds.	162
WAIT SRQ.	Wait Till Srq Is True.	163
X.	Executes Stored Macro.	164

CHAPTER 1

SETTING UP THE PSI BOX

Introduction.

This chapter introduces the PSI BOX, outlines the function of the various modes and instructs you how to configure the PSI BOX to suit your needs.

Finally, the allocation of the buffer and the function of the Buffer On DIP switch is explained.

The PSI BOX.

The front of the PSI BOX consists of 7 LEDs and two sets of 8 DIP switches. The 16 DIP switches allow the user to select options whilst the LEDs are used to monitor progress of data transfers. The set of DIP switches nearest the LEDs is switch block one. The switches dedicated to the RS232 port options is switch block two.

Mounted on the back of the PSI BOX are a standard IEEE488 plug, a 25 way male RS232 D connector, a 25 way female Centronics Parallel D connector. Each port has two LEDs dedicated to it and a further LED is the mains power indicator. The mains connector is a fused, switched power inlet.

Inside the PSI BOX is an R.F.I. shielded, mains transformer and a printed circuit board containing an enhanced microprocessor, a ram buffer, interface logic and appropriate line drivers for the 3 ports.

The IEEE plug is an industry standard gold plated and EMI shielded connector. The IEEE base address is front panel DIP switchable in the range 0-15. In addition, certain modes use address 15 to select listen only or talk only operation.

The Centronics Parallel port is an IBM PC printer compatible port. The Centronics Parallel port is an IBM PC compatible printer port and may be either an input or an output. The connector is identical to PC parallel port LPT1.

The RS232 port signals are the industry standard +/- 12 Volt level. The serial port is an IBM PC serial compatible port. The connector has the same pin outs as the IBM PC serial port COM1:

The serial port is bi-directional. The RS232 handshake can be either a hardware handshake controlled by the DSR, CTS, DCD and DTR, RTS pins or by the XON/ XOFF software handshake. The baud rate and parity options are selectable from either the front panel or under software control from the IEEE port.

Baud rates of between 75 to 19200 baud are available. Parity may be off or on, even or odd. The may be either 7 or 8 data bits per word.

The internal buffer is 32K ram and may be disabled by front panel DIP switch control. A valuable aide in debugging your communication problems!

The main power inlet is a fused switchable unit.
Fuse rating 3 Amp. The fused connector contains a spare 3 Amp fuse, just in case you might need it. The PSI BOX even comes complete with a fused 13 Amp plug.!

The PSI BOX is a slim, brick shaped and is 10.3 inches long x 2.4 high x 5.5 wide. (262 long x 61 high x 140 wide millimetres.) The PSI BOX weighs 5 lb, that is 2 kg.

The PSI BOX Modes.

The PSI BOX is controlled by its ROM based operating system. It is a stand alone converter unit. It has the ability to work with many different computers and can easily overcome the difficulties in allowing these computers to communicate with each other.

The PSI BOX is flexible enough to allow operation in many different modes, and so solve many different types of problems. These modes are selected by the user when he sets the front panel DIP switches. The PSI BOX may be either an IEEE device or the IEEE controller. Alternatively the PSI BOX can be used to convert between RS232 and Centronics parallel.

The following modes of operation are currently available.

In modes 1 and 2 the PSI BOX is an IEEE 488 device attached to an IEEE controller computer.

<u>MODE 1</u> IEEE 488 to RS232, bi-directional.
IEEE 488 from Centronics Parallel input.

<u>MODE 2</u> IEEE 488 to RS232, bi-directional.
IEEE 488 to Centronics Parallel output.

The IEEE port is not in use in modes 3 and 4.

<u>MODE 3</u> Centronics Parallel input to RS232 output.

<u>MODE 4</u> RS232 input to Centronics Parallel output.

In modes 5, 6 & 8 the PSI BOX is the IEEE 488 controller. Data input is not from an IEEE computer. Only one IEEE device may be accessed and that only as an output device, ie an IEEE printer or plotter.

<u>MODE 5</u> Centronics Parallel input to IEEE 488 output.

<u>MODE 6</u> RS232 input to IEEE 488 output.

<u>MODE 8</u> RS232 input to IEEE 488 output with talk back. Allows HP IEEE plotters to fully emulate Rs232 version.

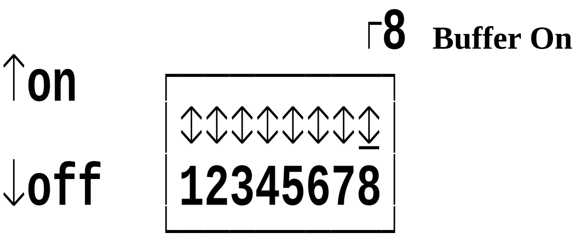
In mode 7, only available on the PSI BOX PLUS+, the PSI BOX is the IEEE 488 controller. The RS232 port of the host computer is used to control the IEEE bus. An enhanced control language is used to address many IEEE devices and to provide complete instrumentation control.

<u>MODE 7</u> RS232 bi-directional to IEEE 488, full control of IEEE bus of up to 14 instruments or devices.

The Buffer Switch.

The 32K ram buffer can be either in operation or bypassed. DIP switch 8 on switch block 2 selects whether the internal ram buffer will be in use.

Figure 1-1. The Buffer Switch.



When the buffer is enabled the ram is divided up between the various ports in use.

In general, the buffer is divided as follows. Half of the available ram is used for the Centronics port and half is used for the RS232 port. The RS232 port partitions its share of the ram buffer evenly between input and output. With the Centronics and RS232 ports both being buffered there is no need for the IEEE port to be buffered.

Due to the many different modes of operation of the PSI BOX the ram buffer partition may be different from that described above. The PSI BOX alters the general case to suit the mode actually in use.

When the buffer is disabled then each of the active ports has a 2 byte buffer. This two byte buffer gives a better flow of data than actually having no buffer at all!.

Figure 1-2. Buffer Option.

DIP 8	BUFFER OPTION CHOSEN.
0=Off	NO BUFFER IN USE.
1=On	BUFFER ENABLED

Figure 1-3. The Buffer Allocation.

Mode	RS232 INPUT	RS232 OUTPUT	CENTRONICS INPUT	CENTRONICS OUTPUT
1	8192	8192	14848	0
2	8192	8192	0	14848
3	0	16128	14848	0
4	16128	0	0	14848
5	0	0	30720	0
6	30720	0	0	0
7	15360	15360	0	0
8	28672	2048	0	0

It is anticipated that enhancements to the basic PSI BOX will be available as time progresses. These enhancements will probably include memory upgrades, improvements to the ROM based software and further input - output options.

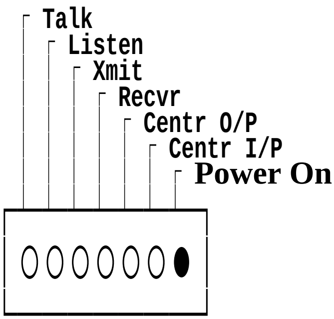
Setting It All Up.

The PSI BOX comes complete with a mains cable and a fused 13 Amp plug.

Connect the mains cable to the inlet on the back of the PSI BOX. Plug the 13 Amp plug end into a suitable mains outlet.

Using the switch on the back of the PSI BOX switch the PSI BOX on. LED 7 on the front of the PSI BOX is the power LED. It lights up RED when the PSI is switched on. If the PSI BOX does not power up check the cable connections and the mains switches. If you still have trouble please contact your dealer.

Figure 1-4 The Power LED.



Once you have got the PSI BOX switched on then you must configure each port ready for use. Consult the sections that follow, for specific details on each port, and the MODE SETTING instructions. The details given should be sufficient to resolve any ambiguities.

Configuring The PSI BOX.

Don't be confused by the flexibility of the PSI BOX, it is really quite easy to make it perform the way you want.

To configure the PSI BOX follow these steps:-

1) Determine whether you are going to use your Centronics Parallel Port as an input or output. If this port is not in use then set it as an output.

2) Set up the RS232 port options, baud rate, parity etc.

3) Determine whether you want the PSI BOX to be an IEEE device or the IEEE controller.
   If it is an IEEE device then it will be controlled by commands from, say, a Commodore PET or Hewlett Packard computer.
   If the PSI BOX is the IEEE controller then you are using it to control an IEEE instrument or other device from the serial or parallel port of some computer.

4) If the PSI is an IEEE device then select the IEEE base address. This gives the address of the RS232 port, the address of the Centronics Parallel port is one more than the base address set by the dip switches.

5) If the IEEE port is acting as the controller to an IEEE instrument then set the mode switches to correctly determine exactly how the IEEE device gets its instructions and which of the RS232 or Centronics port is generating the IEEE data.

6) Connect up all your cables and away you go.

7) If you are having a problem watch the way the LEDs blink this can give you valuable information about where the fault lies.

8) If the RS232 port appears to be at fault, then use a loop back connector on the RS232 port. This will allow you to determine whether the fault is due to the RS232 cross over cable used, incorrect baud rates or wrong mode switch settings.

Problems!

You won’t have any problems! BUT if you do it probably due to the RS232 port cross over cable being wired incorrectly. First make sure that you have read the followed the steps above, next re read the relevant sections of this manual. For further advice, please contact your dealer.

CHAPTER 2
AN INTRODUCTION TO
THE IEEE-488
INTERFACE BUS
STANDARD.

Introduction.

This chapter is divided in to the following main parts.
"The IEEE 488 Standard" takes a look at how the IEEE 488 standard came into being, and gives a simple definition of what the bus actually is.
"The IEEE Signal Lines" is a brief look at the bus signal lines, talks about cable length restrictions and data transfer speed limits.
In "Using The IEEE Bus" the major features of the IEEE 488 system are described.

If you know little about IEEE we suggest that you browse though the first two parts of this chapter and then carefully read the rest.

The IEEE 488 Standard.

The Beginnings Of A New Standard.

In September 1965 the Hewlett Packard Company realised that it needed to standardise the way in which all future H.P. instruments would communicate with each other. The HPIB is defined.

In March 1972, the IEC committee accepts HPIB as a starting point for the draft document of its IEC standard which is published in June 1980 as IEC 625-1.

In April 1975, the Institute of Electrical and Electronics

Engineers published IEEE 488, this is revised in November 1978 and again in 1980.

By 1986 more than 250 manufacturers in more than 14 countries throughout the world are building over 2000 different IEEE 488 based products..... and the PSI BOX lets you control them all from your desk top PC!

What Exactly Is IEEE 488?

Basically IEEE 488 is a way of connecting electronic instruments together and putting them under computer control.

The IEEE 488 standard describes a connecting plug, a set of 16 signal wires and 8 earth wires and sets out the way that each instrument may send and receive data.

By strictly defining these parameters in an unambiguous way, the IEEE 488 standard has become powerful though its ease of use and straight forward implementation.

Any one who has ever attempted to link two RS232C devices together will admire the no soldering, piggy back style, cable connection and the no baud rate, parity free, high speed IEEE 488 data exchange.

The intelligence of the IEEE 488 system is built into the interface and does not lie in the user's ability to decipher how to wire up a cross-over or null modem cable!

What's In A Name?

There are four international standards governing byte serial bit parallel interface systems for instruments.

IEEE-488 The IEEE-488 standard is also known as the General Purpose Interface Bus, GPIB, the Hewlett Packard Interface Bus, HPIB, IEEE Bus, PLUS Bus and even the ASCII Bus.

ANSI MC1.1 This is identical to the IEEE standard.

<u>IEC 625</u> This European standard is identical to the IEEE standard except for the connector. It uses a 25 pin D type RS232C style connector.

<u>BS6146</u> This British standard is identical to IEC 625

NOTE: Never, ever connect an IEC 625 system to an RS232C system. The =/- 12 Volts of the RS232C system can seriously damage the 5 Volt TTL IEC 625 system!

<u>IEEE 488-1975.</u>

The full name of the IEEE Standard 488-1975 is "A Digital Interface for Programmable Instrumentation". It defines a byte serial bit parallel interface for instrument systems. What does this mean?

By Digital Interface it means that the signal on any of the 16 signal lines is either high or low. The voltage on each line represents the On or Off state. So the data exchanged between the devices connected together is digital.

Bit Parallel means that 8 lines are used to simultaneously transmit a whole byte of data. These 8 lines are called collectively the Data Bus. The data on the bus ranges in value from 0 to 255. Since data is sent 8 bits at a time the IEEE bus is inherently faster than the one bit at time RS232 system.

Byte Serial means that consecutive bytes of data are sent one after another. Another example of bit parallel byte serial data transfer is the Centronics Parallel printer connection found on most PC's.

Programmable Instrumentation means that the devices attached to the IEEE bus are usually capable of being instructed to perform several different tasks in a sequence. Often an IEEE system comprises devices that can measure various parameters and devices that can generate various stumulii. For example a programmable IEEE voltage generator may be instructed to apply several voltages across a resistor, whilst a programmable IEEE Ammeter may be used to measure the current through the resistor once the voltage has stabilised. In this way IEEE instruments can be easily built up to form a simple but effective automatic test system.

So the IEEE Standard 488 specifies a digital interface for programmable instrumentation, that is, it is a way of inter-connecting electronic measuring and control apparatus.

The IEEE Signal Lines.

The IEEE Signal Lines.

The IEEE 488 defines the use of a parallel bus to interconnect up to 15 instruments. Each device on the bus is connected to the 16 signal lines and 8 ground lines on the bus. These 16 signal lines can be considered as three separate groups. Eight of the lines are used to carry data and commands. Three lines control the transfer of the data. Whilst the remaining five lines are for general interface management.
These three groups are the 8 Data Bus signals, the 3 Handshake signals and the 5 Management signals.

The Data Bus Signals.

The 8 data bus signals are called DIO1 to DIO8.
The data bus carries all data, programming instructions and IEEE interface messages. Most instruments communicate using the 7 bit ASCII code, however the information on the data bus may represent ASCII, Binary, Hex, Octal, or whatever, characters.

The Handshake Signals.

The 3 handshake signals are DAV, NRFD and NDAC.
The handshake signals control the actual transfer of each data byte down the data bus. They provide the famous ’3-wire handshake’ that ensures that the data is transmitted correctly and without loss, no matter what the difference in operating speeds

of the participating devices.

DAV Data Valid.

The DAV line indicates that the data on the data lines is good data. The DAV is controlled by the device that is putting the data on the data bus. The DAV line is monitored by all the devices that are accepting data from the data bus. The DAV line is false when the data on the data bus is invalid and is true when the data bus contains valid data.

NRFD Not Ready For Data.

The NRFD line indicates the condition of readiness of devices to accept data. The NRFD line is monitored by the device that puts the data on the data bus. It is driven by all the devices that are accepting data from the data bus.
The NRFD line is false when the all these devices are ready for data and true when one or more of them is not ready for data.

NDAC Not Data Accepted.

The NDAC line indicates acceptance of data by the devices.
The NDAC line is monitored by the device that puts the data on the data bus. It is driven by all the devices that are accepting data from the data bus. When NDAC is true, one or more of the listening devices have not accepted the data.

The Management Signals.

The 5 management signals are ATN, IFC, SRQ REN and EOI. These five signals are used to manage an orderly flow of information across the bus.
The ATN and IFC lines are used by all IEEE bus devices. The remaining three, REN, SRQ and EOI, may or may not be used

by a particular device.

ATN Attention.

All devices on the IEEE bus must monitor the ATN line. The attention line is used by the controller to specify how data on the data bus is to be interpreted and which devices must respond to the data. The ATN line is set true by the controller when it sends interface messages, such as device talk and listen address, secondary addresses, and polling configuration messages. The ATN is set false by the controller so that the active talker can send device dependent messages, such as data and programming information, to the active listeners.

IFC Interface Clear.

The IFC line, interface clear, is used by the controller to place the bus interface in a known inactive state. Only the System Controller can set the IFC line, all other instruments must monitor and respond to IFC. The System Controller may set IFC true at any time.

REN Remote Enable.

The REN line, remote enable, sends the remote message. REN must be true for an device to operate under remote control from a system controller, otherwise the device is programmed locally from its front panel. Only the System Controller can set the REN line, all other instruments must monitor and respond to REN by returning to local control whenever it becomes false. The System Controller may set REN true at any time.

SRQ Service Request.

The SRQ line, service request, is used by a device to indicate the need for attention and to request an interruption of the current sequence of events. The controller can clear the SRQ line only by performing a serial poll of the device requesting service.

EOI End Or Identify.

The EOI line, end or identify, is used by a talker to indicate the end of a data string or, by the controller, in conjunction with ATN, to conduct a Parallel Poll sequence.
When ATN is false, the current talker may send the END message, indicating the end of its data, by setting EOI true at the same time it places the last data byte on the data bus. The controller may send the identify message, IDY, to initiate a Parallel Poll of all devices with the Parallel Poll capability by setting ATN and EOI true.

Negative Logic

One of the things that commonly confuses IEEE bus users is that the IEEE signals are described as using ’negative logic’. This simply means that a signal is true when it is low, at 0.4 Volts or less, and that a signal is false when it is high, at 2.4 Volts or greater. Do not let this worry you, just think of the different signals being either True or False.

Cabling Lengths and Data Transfer Speeds.

You are permitted to connect up to 15 devices together on the IEEE bus, that is one PSI BOX plus 14 other devices. This should be more than adequate for the vast majority of users. When more than 15 devices MUST be placed on one bus, the GPIB can be extended with bus extenders.

The general limit for total cable length on the bus is 2 metres for every IEEE device connected. This is provided the total transmission path length does not exceed 20 metres. There should be not more than 4 metres of cable between any two devices. Within this limit, the IEEE bus is capable of a maximum data transfer rate of 250K bytes per second with open collector drivers, or 500K bytes per second with tri-state drivers.

When shorter distances are involved, 15 metres, with every device turned on, with not more than 1 device per metre of cable and with tri-state drivers the maximum theoretical data rate is 1 Megabyte per second.

Of course, the data is actually transmitted at the speed of the SLOWEST currently addressed listener or talker. The positive 3 wire handshake ensures that data is transmitted correctly and without loss despite the actual speed in use.

Using The IEEE Bus.

Types Of IEEE Devices.

Every device on the IEEE bus must be capable of acting in at least one of following three roles.

1) A device acting as a LISTENER.
2) A device acting as a TALKER.
3) A device acting as a CONTROLLER.

The Listener.

A listener is a device with the capability to receive data from the IEEE bus when addressed. This transmitted data is known as device dependant messages.

A typical listener is a printer. Other listeners include disk drives, programmable oscilloscopes, programmable signal generators, programmable power supplies etc. There can be up to 14 active listeners simultaneously on the IEEE bus.

The Talker.

A talker is a device with the capability to transmit data over the IEEE bus when addressed. This transmitted data is known as device dependant messages.

A typical example of a talker is a disk drive sending data, a digital voltmeter sending data, a programmable oscilloscope sending data, etc. There can be only one active talker on the IEEE bus at any one time. When a device is commanded to talk all other talkers are automatically commanded to untalk.

The Controller.

A controller is a device that can instruct other devices to talk or to listen. In addition, the controller can send interface messages to command specified actions within other devices. A typical example of a controller is your PC with the PSI BOX connected, the PSI BOX can also act as a listener or a talker when necessary.

The Controller In Charge.

If there are several controllers present on the IEEE bus only one of them at a time can be the Controller In Charge. When a controller is not the Controller In Charge it must be in the controller idle state. The Controller In Charge can be in the controller standby state or by asserting ATN true, it is the Active Controller. So the Controller In Charge at will can move between the Active Controller and the controller standby states.

If several controller are connected to one bus then only one of these can be the System Controller.

The System Controller.

Whilst any idle controller can be instructed to become the Active Controller only the System Controller can MAKE ITSELF the Active Controller at any time. Only the System Controller can assert the IFC and REN lines.

The PSI BOX can also act as controller in standby state, as Controller in Charge or as System Controller.

The PSI BOX, in Mode 7, powers up as System

Controller, this suits 99.99% of all users.

Controller Summary.

To sum up the Controller states.

1) A controller commands other devices whether they should talk, listen or perform some other action.
2) To avoid confusion, if several controller are connected to an IEEE system, only one of them at a time is allowed to send commands to the bus. This is the Controller In Charge.
3) Interface messages exist for controllers to pass control between each other, ie make each other the Controller In Charge.
4) Only one controller on an IEEE system can be the System Controller. The System Controller has several unique abilities. The System Controller alone can set the IFC and REN signals true or false. The System Controller can make itself the Controller In Charge, at any time, by sending the IFC signal true for 100 micro seconds.

Command Mode And Data Mode.

Two type of messages may be transmitted on the IEEE bus. These are device dependant messages and interface messages. Device dependant messages are distinguished from interface messages by the state of the ATN line.

The controller asserts ATN true when it wants to send an interface message, the IEEE bus is said to be in COMMAND MODE. When ATN is false the data on the data bus is a device dependant message, the IEEE bus is said to be in DATA MODE.

Only the Active Controller can send interface messages, that is, commands.

Any active talker can send device dependant messages, that is, data.

Device dependant messages are only sent from the active talker to the active listeners, they are ignored by all other devices. A typical example of a device dependant message is the data byte 13, HEX 0dH, when ATN line is false. This would be interpreted by an actively listening IEEE printer as the instruction to perform a carriage return. All the data that is sent between IEEE devices are device dependant messages,

Interface messages are sent from the Controller In Charge to the bus devices. Each device on the bus may accept and process any or all depending on the capability of the device and the type of command sent. The four groups of commands sent by the controller are:-

1) Universal commands accepted by ALL devices.
2) Addressed commands accepted by all devices addressed to listen.
3) Addresses accepted by ALL devices.
4) Secondary addresses or commands accepted by all devices enabled by a primary address or command.

A typical example of an interface message would the data byte 63, with ATN true, this is the interface message UNL, an UNaddress, and it tells all devices to stop listening to any device dependant data transfer.

Addresses.

Every IEEE device has at least one address. Each device has a unique address, often set by DIP switches. The device addresses are used by the active controller in command mode to specify who shall talk and shall listen. Every device must be able to recognise its own address and respond correctly to it. No two IEEE devices, especially talkers, should be set to the same address.

Valid addresses are in the range 0 to 30 inclusive. It is usual that the Talk address of a device is the same as its Listen address.

Talk Only And Listen Only Devices.

The only exception to addressed devices are Talk Only or Listen Only devices. There may be only one Talk Only device on an interface bus. When the controller is in the idle state, ATN low, the Talk Only device will always be ready to send its data. A Listen Only device will receive any data from any talker no matter what other devices have also been commanded to listen.

One of the simplest IEEE systems is with a Talk Only digital meter and a Listen Only printer. The printer will log every reading that the meter sends. This type of system does not require the presence of an IEEE Controller.

In addition to the basic function of instructing devices to talk or listen the IEEE Standard provides interface messages that allow the controller to accomplish the following four functions.

1) Serial Polls.
2) Parallel Polls.
3) Device Clear and Device Trigger.
4) Remote/ Local functions.

Serial Poll.

When a device requires the attention of the controller it asserts SRQ true. The Serial Poll is the reply that the Controller gives to a device asserting SRQ, service request. It is the IEEE device that causes the serial poll to be performed because it asserts SRQ when it needs servicing by the controller.

The SERIAL POLL is used to determine which device has requested service from the controller and why. Serial poll is the only method that can correctly cause the device to return the SRQ line false. In general, you will know which device requested the service and so will know which to poll. However, if it could be one of several devices that is requesting service, you should perform successive serial polls on the different devices until the SRQ line returns to the false state.

The serial poll response of each device is held in bytes 28-30 of the IEEE control status string when that device is addressed as the current device. The serial poll response contains data concerning the service request.

If bit 6 (value 64) of the devices serial poll response is set then that device was requesting service, the other bits in the response byte will contain a code indicating the nature of the service requested.

Conversely if bit 6 (value 64) of the serial poll response is not set then this device did not request service.

Several devices can assert SRQ true at one time so it is good practise to check that SRQ has returned false after the Serial Poll routine to ensure that all devices that have requested service are monitored.

In general, many devices have the ability to assert service request and respond to serial polls.

Parallel Poll.

The Parallel Poll function allows the IEEE controller to periodically check selected bus devices to see if they require some special service.

The Parallel Poll makes good use of the controller’s time since it can quickly check the state of 8 or more devices with one command. Unlike the Serial Poll which is conducted after a device requests service, the Parallel Poll routine is initiated by the controller. It allows a whole group of devices to simultaneously return a status bit on the data lines. The data line, and the sense of each devices response is determined by the controller when it issues a CONFIGURE command or PPC, parallel poll configure. Devices power up in the PPU state and cannot respond to the Parallel Poll until after they have been configured.

The Controller knows which device responds on which data line because it has previously assigned those lines to each device using the CONFIGURE or PPC with PPE commands. Depending on the Parallel Poll Response the controller may take a variety of actions, chosen by the user's program.

During a parallel the configured IEEE devices put their parallel poll status bit on the IEEE data lines. The PSI BOX reads the data line and store the response in bytes 36-38 of the IEEE control string.

These bytes hold a value between 0 and 255. Knowing what parallel poll configure information has sent sent the user can interpret the data accordingly.

In general, few devices have the ability to be configured to respond to a parallel poll.

Device Clear and Device Trigger.

The device clear commands are used to make all or some devices reset themselves to an initial power on state. The actual initial state depends on the particular devices function. Device clearing is very useful when a device hangs or inadvertently is placed in unresponsive mode. It is worth noting that not all IEEE devices are capable of recognising and executing a device clear command, check the devices manual.

The SDC, selective device, clear command can be used to set individual devices to their initial condition whilst the DCL, device clear, command sends all devices to that condition.

Depending on whether any parameters are given the CLEAR command performs an SDC or a DCL command.

Many IEEE devices have the capability to be pre-programmed to perform a given action. When several devices are programmed in this way it may be necessary to have them perform their individual actions simultaneously or in sequence. The TRIGGER command can be used to instruct several devices or an individual device to perform take a reading

Device trigger commands are often used in a test and measurement system to apply a signal to a component and then measure a resulting parameter. Thus the intelligence of IEEE devices can be exploited to a great extent.

Remote - Local Functions.

The remote and local functions are used to place the IEEE device under remote control via the PSI BOX or under local control via its front panel switches.

The Remote command is used to make the controller assert the REN line, remote enable, so allowing the bus devices to respond to the PSI BOX. If the REN line is not asserted the bus device will not respond to the PSI BOX, they will ignore the PSI BOX.

Even when the bus devices are under PSI BOX remote control if you press the devices front panel "LOCAL" button it will return to local control unless it has been put into the LOCAL LOCKOUT state.

Whenever a read or write is performed or whenever any bus command is sent the PSI BOX reasserts the REN line.

The GTL, Go To Local, and Local commands are used to tell all the currently listening devices to respond to their front panel, local, controls rather than to the PSI BOX.

The Local Lockout command is used to tell all the bus devices to ignore their front panel, local, controls and respond only to the IEEE controller. This is necessary when you want the instrument to be totally under the control of the PSI BOX. It prevents accidental pushing of the devices buttons interfering with its operation.

The device will return to local control when it receives the LOCAL or GTL command, or when the IEEE REN line goes false.

End Of Sequence Termination,

How do IEEE devices know that they have received all the data in a string? There are two methods that allow IEEE devices to identify when the talker has sent all its data. These methods allow devices to receive strings of an arbitrary length without first specifying how many bytes are going to be sent. This makes IEEE data exchange extremely flexible.

The first method is using the EOI line. A talker should only set EOI true with the last data byte it intends to send. All listeners will recognise EOI as signalling the end of the data stream, after reading the data byte that has EOI true they will stop listening.

The second method is a predefined End Of Sequence byte. To use this method the talker and all listeners must know what byte ALWAYS marks the end of a string. The talker should only send its EOS byte as the last byte in the data stream. The listeners, on detecting this EOS byte, recognise that no more data is to follow and so stop listening. Usually this End Of Sequence byte will be a carriage return or a line feed character.

Whilst both methods can be used simultaneously, if data is to be exchanged without loss, bus hang ups or time outs both talkers and listeners be correctly configured to detect the end of a data exchange.

The PSI BOX uses the END command to specify how it terminates outputs from the PC to the IEEE bus devices. The EOS command is used to specify how the PSI BOX recognises the end of an input from an IEEE bus device. Note that on inputs from the IEEE bus the PSI BOX always stops reading data when the talking device sends EOI true.

CHAPTER 3
THE IEEE 488 PORT.

Introduction.

This chapter describes the IEEE488 port on the PSI BOX. The IEEE488 port connector pin outs, the function of the DIP switches and LEDs, how to set the controller mode and the device addresses are all explained in detail.

Summary.

The IEEE port provides TTL logic level signals. A standard 24 way Amphenol IEEE 488 connector, with screw locks for secure connections, is used. The IEEE plug is gold plated and EMI shielded.

The IEEE port lines are implemented with the PSI BOX in one of two modes.

1) The PSI BOX may be an IEEE peripheral, here, the ATN, IFC and REN lines are under control from an IEEE computer and the SRQ line is shared by the PSI BOX with all the other IEEE devices present.

2) The PSI BOX may be the IEEE controller, here, the ATN, IFC and REN lines being managed by the PSI BOX, and the SRQ line is shared between any IEEE devices attached to the PSI BOX.

In both modes the DATA 0-7 lines and the DAV, NRFD and NDAC lines are controlled according to the handshake in progress.

The IEEE device address of the PSI BOX is front panel DIP switchable in the range 0-15. A front panel LED flashes whenever the IEEE port sends a byte, ie addressed to TALK. A front panel LED flashes whenever the IEEE port receives a byte, ie addressed to LISTEN.

As an IEEE device, in modes 1 and 2, the PSI BOX responds to two addresses. The first address as set on the front panel and is the IEEE to RS232 converter address. The second address is one higher than this base address and is the IEEE to Centronics Parallel converter address.

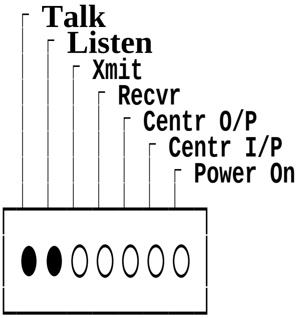
Each port may be addressed using secondary addresses. These secondary addresses allow the IEEE controller to interrogate the PSI BOX to determine the RS232 and Parallel port configuration and indicate whether any data is in the input and output buffers. They also allow the PSI BOX to perform character conversion on the data passing through it.

As an IEEE controller the PSI BOX may accept data from either the RS232 or the Centronics port and send it an IEEE device on the IEEE port. The PSI BOX has a full controller handshake.

The IEEE Leds.

There are two LEDs dedicated to the IEEE port. LED 1 and 2 are used to indicate the flow of data through the IEEE port.

Figure 3-1. The IEEE Leds.

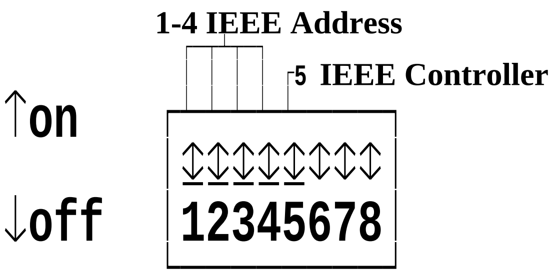


LED 1, TALK, flashes whenever the IEEE port of the PSI BOX sends a byte down the IEEE bus. LED 2, LISTEN, flashes whenever the IEEE port of the PSI BOX accepts a byte from the IEEE bus, the byte is stored in the PSI BOX internal buffer.

The IEEE DIP Switches.

DIP switches 1-5 on switch block 1, in conjunction with the MODE switch 7 & 8, are used to set the all the IEEE options. These options include IEEE address, IEEE controller mode and source of the IEEE data.

Figure 3-2. The IEEE DIP Switches.



The DIP switches are only read when the PSI BOX is powered up. Changing the DIP switch setting whilst the PSI BOX is running will have no effect.

Setting The IEEE Address.

DIP switches 1-4, on switch block 1, are used to set the IEEE address of the PSI BOX. When the PSI BOX is acting as an IEEE device, in modes 1 and 2, the address to which it responds is determined by the front panel DIP switches. To an IEEE controller the PSI BOX appears to be two separate devices, one is an IEEE to RS232 interface, the other an IEEE to Centronics Parallel interface. The data that the IEEE controller sends goes to which ever port is being currently addressed. The RS232 port is as set on the DIP switches, the Centronics port address is one greater than this.

Figure 3-3. Setting The IEEE Address.

DIP 1	DIP 2	DIP 3	DIP 4	ADDRESSES SELECTED.	
0=Off	0	0	0	0 =RS232 PORT	1 =CENTRONICS PORT
1=On	0	0	0	1 =RS232 PORT	2 =CENTRONICS PORT
0	1	0	0	2 =RS232 PORT	3 =CENTRONICS PORT
1	1	0	0	3 =RS232 PORT	4 =CENTRONICS PORT
0	0	1	0	4 =RS232 PORT	5 =CENTRONICS PORT
1	0	1	0	5 =RS232 PORT	6 =CENTRONICS PORT
0	1	1	0	6 =RS232 PORT	7 =CENTRONICS PORT
1	1	1	0	7 =RS232 PORT	8 =CENTRONICS PORT
0	0	0	1	8 =RS232 PORT	9 =CENTRONICS PORT
1	0	0	1	9 =RS232 PORT	10 =CENTRONICS PORT
0	1	0	1	10 =RS232 PORT	11 =CENTRONICS PORT
1	1	0	1	11 =RS232 PORT	12 =CENTRONICS PORT
0	0	1	1	12 =RS232 PORT	13 =CENTRONICS PORT
1	0	1	1	13 =RS232 PORT	14 =CENTRONICS PORT
0	1	1	1	14 =RS232 PORT	15 =CENTRONICS PORT
1	1	1	1	15 =RS232 PORT	16 =CENTRONICS PORT

Setting The Controller State.

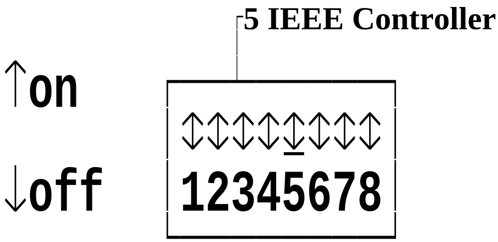
DIP switch 5, on switch block 1, is used to determine whether the PSI BOX is the IEEE controller or not.

If the PSI BOX is the IEEE controller then you are wanting the PSI BOX to convert an RS232 or Centronics signal to one suitable to control some IEEE devices.

If the PSI BOX is not the IEEE controller then you are using an IEEE computer, say a COMMODORE PET, and you are wanting the PSI BOX to convert this signal to communicate with an RS232 and/or a Centronics device

The MODE switches, in conjunction with the IEEE controller switch, determine the exact PSI configuration in use. Please refer to the MODE switch settings for more information.

Figure 3-4. The Controller State.



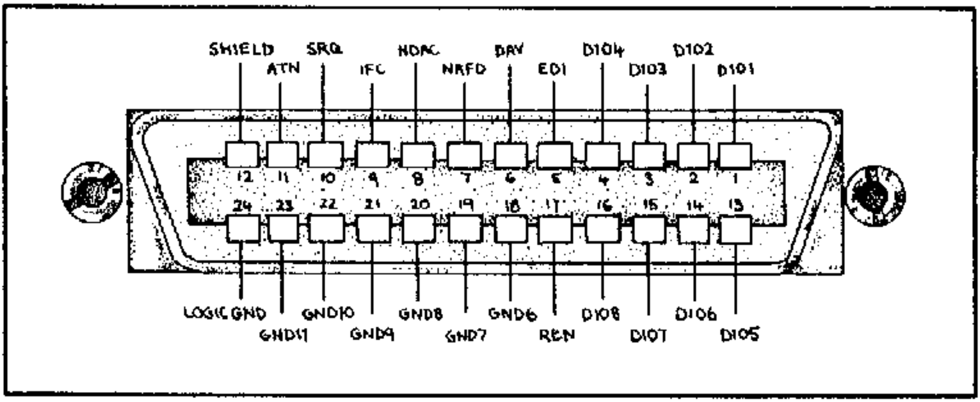
DIP 5	CONTROLLER STATE CHOSEN.
0=Off	PSI BOX IS AN IEEE DEVICE
1=On	PSI BOX IS THE IEEE CONTROLLER.

IEEE Cables.

Details of the IEEE port pin outs are given below. The Connector is a standard IEEE-488 24 pin plug with screw locks. Any standard commercially available IEEE cable will connect to the PSI BOX IEEE port. We always have a good range of high quality R.F.I. shielded cables in stock.

Figure 3-5. The IEEE Port Pin Outs.

PIN	SIGNAL	PIN	SIGNAL
1	DATA 1	13	DATA 5
2	DATA 2	14	DATA 6
3	DATA 3	15	DATA 7
4	DATA 4	16	DATA 8
5	EOI	17	REN
6	DAV	18	GROUND 6
7	NRFD	19	GROUND 7
8	NDAC	20	GROUND 8
9	IFC	21	GROUND 9
10	SRQ	22	GROUND 10
11	ATN	23	GROUND 11
12	SHIELD	24	SHIELD



CHAPTER 4

THE CENTRONICS PARALLEL PORT.

Introduction.

This chapter describes the Centronics parallel port on the PSI BOX. The parallel port connector pin outs, the function of the Centronics Output DIP switch and the Centronics LEDs, the parallel port input and output handshake are all explained in detail.

Summary.

The Centronics Parallel port is an IBM PC printer compatible port, it provides TTL logic level signals. The connector has the same pin out as the IBM PC parallel port LPT1: etc. A 25 way female D connector, with screw locks for secure connections, is used. The port may be either an input or an output. A front panel LED flashes whenever the Centronics port sends a byte. A front panel LED flashes whenever the Centronics port receives a byte.

The Centronics Parallel port is normally used to drive a parallel printer. When used as an input it can receive the output from a computers parallel printer port and so act as the input for a parallel to RS232 or parallel to IEEE interface.

Unlike the two other ports, the Centronics Parallel port is NOT bidirectional. It can act EITHER as an input OR as an output. It can not change over in mid stream from input to output or vice versa.

Centronics Leds.

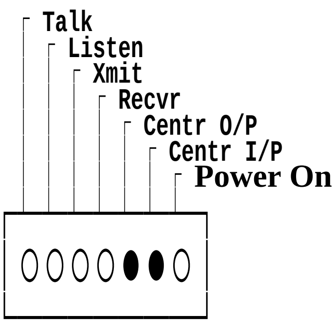
There are two LEDs dedicated to the Centronics Parallel port. LED 5 and 6 are used to indicate the flow of data to the Centronics Parallel port.

LED 5, CENTR O/P, flashes whenever the Centronics Parallel port of the PSI BOX sends a byte, from its buffer, to the parallel bus.

LED 4, CENTR I/P flashes whenever the Centronics Parallel port of the PSI BOX receives a byte from the printer port of a computer, the byte is stored in the PSI BOX internal buffer.

Since the Centronics port is either an input OR an output only one of these LEDs will be in use for the particular PSI BOX configuration chosen.

Figure 4-1. The Centronics Leds.



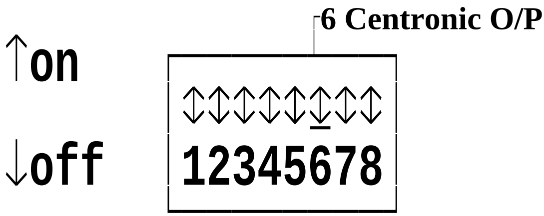
Selecting The Direction Of The Centronics Port.

DIP switch 6 on switch block 1 is used to determine whether the Centronics Parallel port is an input or an output.

When used as an input, it can receive the output from a computers parallel printer port and so act as the input for a parallel to RS232 or parallel to IEEE interface.

When used an an output, the Centronics Parallel port is normally used to drive a parallel printer.

Figure 4-2. Setting Parallel Port Direction.



DIP 6	PORT DIRECTION CHOSEN.
0=Off	Centronics Port INPUT.
1=On	Centronics Port OUTPUT.

The Centronics Output Handshake.

The Centronics port output handshake is as follows.

A) If the Centronics printer is busy, it should raise the BUSY line true. The PSI BOX will not send any data out of the Centronics port if the BUSY INPUT is high.

B) When the BUSY INPUT line to the PSI BOX is low then the PSI BOX will send any data in its buffer to the Centronics port.
The PSI BOX places the first byte on the DATA 0-7 lines. The PSI BOX allows the data lines to stabilise.

C) Then the PSI BOX toggles the STROBE line low, for a minimum of 1 microsecond. The Centr o/p LED flashes.

D) The printer will respond by setting the BUSY line true, high, and will read the data off the data bus.

E) When ready to receive another byte from the PSI BOX the printer sets BUSY low and toggles the ACKNOWLEDGE line low. The PSI BOX will maintain valid data on the data bus for a minimum of 1 microsecond after the STROBE line goes high. Later the printer sets BUSY low and toggles the ACKNOWLEDGE line low, for a minimum of 1 microsecond.

F) The handshake starts again.

The Centronics Input Handshake.

The Centronics port input handshake, which has been improved to cope woth screen dumps via Shift-PrtScr from the latest breed of fast 386 & 486 PCs, is as follows.

A) If the PSI BOX is busy or if its Centronics input buffer is full, it will raise the BUSY line true. Any data that is sent to the PSI BOX while the BUSY line is high is LOST! Even though the Centr i/p LED may flash.

B) When the Centronics device senses that the PSI BOX has set the BUSY line low, it puts data on the data bus.

C) The Centronics device signals the PSI BOX by toggling the STROBE line low, for a minimum of 1 microsecond.

D) The PSI BOX will respond by setting the BUSY line true, high, and will read the data off the data bus.

E) When ready to receive another byte the PSI BOX sets BUSY low and toggles the ACKNOWLEDGE line low, for a minimum of 1 microsecond. If the PSI BOX input buffer becomes full then it will keep BUSY high and will not sent ACKNOWLEDGE until the buffer is emptied.

F) The handshake starts again.

Centronics Cables.

When used as an output the Centronics port can drive a parallel printer. A standard IBM PC parallel printer cable can be used to connect the PSI BOX parallel port to the printer.

When used as an input the PSI BOX parallel port can receive data from the parallel printer port of a computer. To drive the PSI BOX from the LPT1: port of an IBM PC make up a cable as follows. A simple straight through cable with male to male 25 pin D connectors is all that is required. Since in the PSI BOX the PE line, pin12, is pulled low and /ERROR line, pin 15 and the SELECT line, pin 13 are pulled high the PSI BOX emulates a printer port correctly.

Figure 4-3 The Centronics Output Port Pin Outs.

PIN	SIGNAL	DIRECTION
1	STROBE	OUTPUT FROM PSI BOX
2	DATA 0	OUTPUT FROM PSI BOX
3	DATA 1	OUTPUT FROM PSI BOX
4	DATA 2	OUTPUT FROM PSI BOX
5	DATA 3	OUTPUT FROM PSI BOX
6	DATA 4	OUTPUT FROM PSI BOX
7	DATA 5	OUTPUT FROM PSI BOX
8	DATA 6	OUTPUT FROM PSI BOX
9	DATA 7	OUTPUT FROM PSI BOX
10	ACKNOWLEDGE	INPUT TO PSI BOX
11	BUSY	INPUT TO PSI BOX
12-17	NO CONNECTION	
18-25	GROUND	

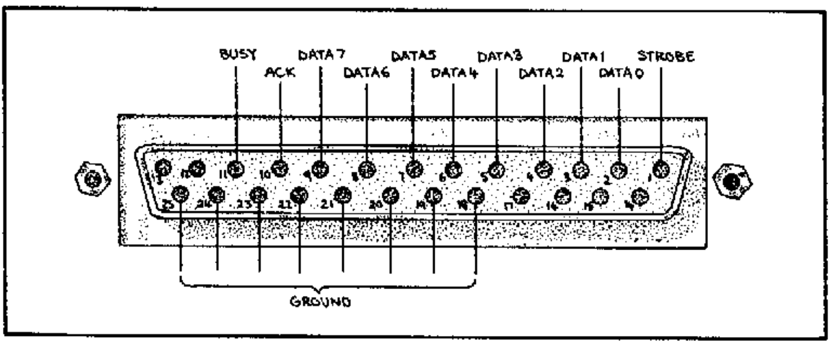
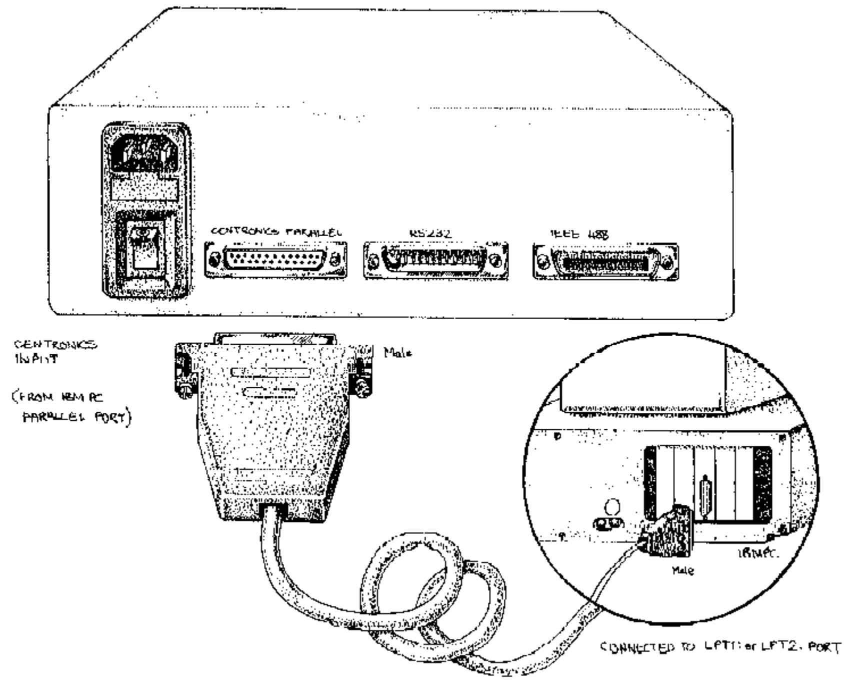


Figure 4-4. The Centronics Input Port Pin Outs.

PIN	SIGNAL	DIRECTION	
1	STROBE	INPUT TO PSI BOX	
2	DATA 0	INPUT TO PSI BOX	
3	DATA 1	INPUT TO PSI BOX	
4	DATA 2	INPUT TO PSI BOX	
5	DATA 3	INPUT TO PSI BOX	
6	DATA 4	INPUT TO PSI BOX	
7	DATA 5	INPUT TO PSI BOX	
8	DATA 6	INPUT TO PSI BOX	
9	DATA 7	INPUT TO PSI BOX	
10	ACKNOWLEDGE	OUTPUT FROM PSI BOX	
11	BUSY	OUTPUT FROM PSI BOX	
12	PE	OUTPUT FROM PSI BOX	PULLED LOW NEW LINE
13	SELECT	OUTPUT FROM PSI BOX	PULLED HIGH NEW LINE
15	/ERROR	OUTPUT FROM PSI BOX	PULLED HIGH NEW LINE
14, 16-17	NO CONNECTION		
18-25	GROUND		

Figure 4-5. Centronics Input From PC Printer Port.



CHAPTER 5

THE RS232 SERIAL PORT.

Introduction.

This chapter describes the RS232 port on the PSI BOX. The RS232 serial port connector pin outs, the function of the DIP switches and LEDs, how to set the baud rate, parity and RS232 handshake options are explained in detail. Details of how to connect the PSI BOX to both an IBM PC and a modem are included.

Summary.

The RS232 port signals are the industry standard +/- 12 Volt level. The serial port is an IBM PC serial compatible port. The connector has the same pin outs as the IBM PC serial port COM1: etc. A 25 way male D connector, with screw locks for secure connections, is used.

The serial port is bi-directional. The baud rate and parity options are selectable from the front panel. Baud rates between 75 to 19200 baud are available. Parity may be off or on, even or odd. The may be either 7 or 8 data bits per word. A front panel LED flashes whenever the RS232 port sends a byte. A front panel LED flashes whenever the RS232 port receives a byte.

The RS232 port lines are implemented with the PSI BOX as the Data Terminal Equipment, DTE. The PSI BOX can be used for asynchronous communications only. It is not for synchronous communications.

RS232 Leds.

There are two LEDs dedicated to the RS232 port. LED 3 and 4 are used to indicate the flow of data to the RS232 port.

LED 3, XMIT, flashes whenever the RS232 port of the PSI BOX sends a byte, from its buffer, to the serial bus.

LED 4, RCVR, flashes whenever the RS232 port of the PSI BOX receives a byte from the serial bus, the byte is stored in the PSI BOX internal buffer.

Figure 5-1. The RS232 Leds.

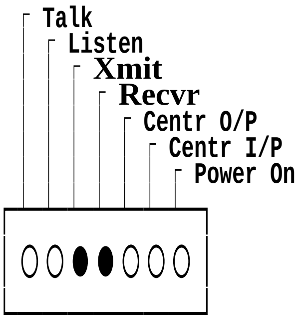
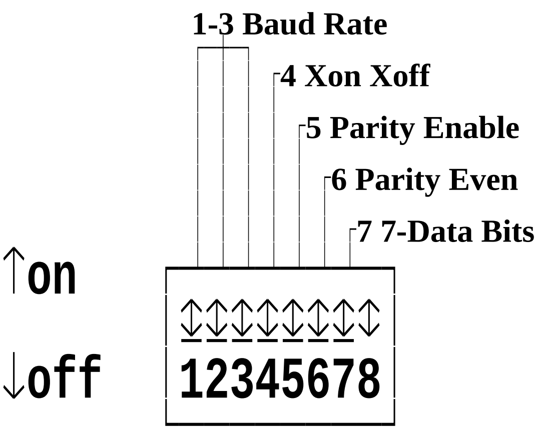


Figure 5-2. The RS232 DIP Switches.



The RS232 DIP Switches.

DIP switches 1-7 on switch block 2 are used to set the all the RS232 options. These options include baud rate, handshake type, parity options and data word length.

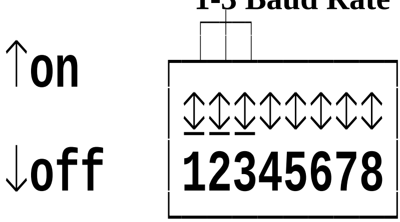
The DIP switches are only read when the PSI BOX is powered up. Changing the DIP switch setting whilst the PSI BOX is running will have no effect.

Setting The Baud Rate.

DIP switches 1-3 on switch block 2 are used to set the RS232 baud rate. The range of baud rates is 75 - 19200 baud.

Figure 5-3. Baud Rate Settings.

1-3 Baud Rate



DIP 1	DIP 2	DIP 3	BAUD RATE SET
0=Off	0	0	75 BAUD
1=On	0	0	150 BAUD
0	1	0	300 BAUD
1	1	0	1200 BAUD
0	0	1	2400 BAUD
1	0	1	4800 BAUD
0	1	1	9600 BAUD
1	1	1	19200 BAUD

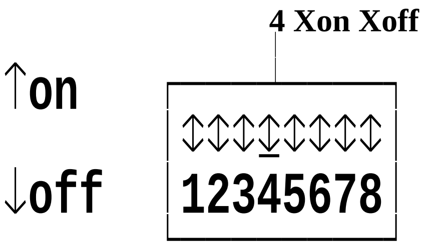
The RS232 Handshake.

DIP switch 4 on switch block 2 is used to select the type of RS232 handshake required.

When the switch is OFF the flow of data is dependent on the state of the DTR, RTS and CTS, DSR and DCD signals, only. This is the hard wired handshake.

When the switch is ON the handshake is conducted in software, the flow of data is controlled by the sending and receiving of the XON and XOFF codes. The handshake lines DTR and RTS are still set true, but the PSI BOX ignores the state of the CTS, DSR and DCD signal inputs and the data flow will be controlled by the XON/ XOFF codes in the data stream. This allows communication without data loss or tears using only 3 wires (TxD, RxD and ground).

Figure 5-4. Selecting The RS232 Handshake.



DIP 4	RS232 HANDSHAKE CHOSEN.
0=Off	DTR, RTS AND CTS, DSR, DCD HARD WIRED HANDSHAKE
1=On	XON/ XOFF SOFTWARE HANDSHAKE.

The Hard Wired Handshake.

The hard wired handshake works in the following manner.

DTR and RTS are outputs from the PSI BOX. CTS, DSR and DCD are inputs to the PSI BOX.

When the PSI BOX sets is powered on it sets RTS out true, indicating that it is present and on line. DTR is also set out true indicating that the PSI BOX is ready to receive data. No data is transmitted from the PSI BOX unless CTS, DSR and DCD are input true.

The PSI BOX sets DTR out low when it is ready to receive data. When the PSI BOX input buffer starts to fill up, less than 128 bytes left (or only 1 if no buffer), it will set DTR out false, this will be detected by the CTS input of the other device. If it continues to receive data it will hold DTR out false, the data received may be lost if there is no room in the PSI BOX buffer. Any data received when DTR or RTS are high high may be LOST, even if the RS232 RCVR LED flashes! When there is more than 128 bytes of room in the PSI BOX buffer, the PSI BOX will set DTR out true. It will remain true provided their is still, at least, 128 bytes of space available in its internal buffer.

The XON/ XOFF Handshake.

The XON/ XOFF handshake works in the following manner. The output lines DTR and RTS are sent out low, and the PSI BOX does not care what state the CTS, DSR and DCD inputs are.

It is recommended that if you are using the XON/ XOFF handshake that the CTS, DSR and DCD inputs to the PSI BOX are forced true.

The flow of data is governed by the sending and receipt of the XON XOFF codes. The XON code is chr$(17), the XOFF code is chr$(19).

When ever the PSI BOX receives XOFF, it will immediately stop sending data until after it has received an XON code.

The PSI BOX sends the XOFF code when it is about to run out of room in which to store incoming data. The minimum amount left before the transmission of the XOFF code depends on whether the 32K buffer is enabled or not.

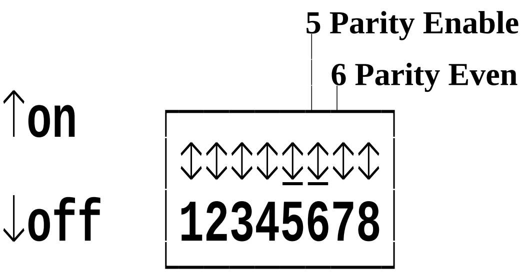
XON XOFF Buffer In Use.

When the PSI BOX input buffer starts to fill up, less than 128 bytes left, it will send an XOFF code to the other device. If it continues to receive data it will send further XOFF codes to the device, the data received may be lost if there is no room in the PSI BOX buffer. When there is more than 128 bytes of room in the PSI BOX buffer, the PSI BOX will transmit the XON code.

XON XOFF Buffer Disabled.

When the PSI BOX is used with the buffer disabled it can only store one byte before it is full. After each RS232 byte is received it will send an XOFF code to the other device. If it continues to receive data it will send further XOFF codes to the device, the data received will be lost since there is no room in the PSI BOX buffer. When there RS232 input buffer is emptied by an IEEE computer reading it the PSI BOX will transmit the XON code, so signalling the RS232 device that it is ready to accept the next byte.

Figure 5-5. Parity Option.



DIP 5	DIP 6	PARITY OPTION CHOSEN.
0=Off	0	NO PARITY
1=On	0	ODD PARITY
0	1	NO PARITY
1	1	EVEN PARITY

Setting The Parity Options.

DIP switches 5 and 6 on switch block 2 are used to set the RS232 parity options. The parity may be enabled (on) or disabled (off), when parity is enabled then the parity bit sent may be either even parity or odd parity.

RS232 Stop Bits.

The number of stop bits sent by the PSI BOX is determined by the parity and data word length chosen.

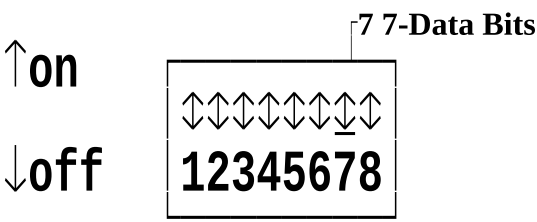
No Parity.
    2 STOP bits are always sent.

Parity Enabled.
    7 data bits gives 2 STOP bits.
    8 data bits gives 1 STOP bit.

Setting The Data Word Length.

DIP switch 7 on switch block 2 is used to set the RS232 data word length. The data word may be either 7 or 8 bits long.

Figure 5-6. The Data Word Length.



DIP 7	DATA WORD LENGTH CHOSEN.
0=Off	8 BITS
1=On	7 BITS

RS232 Port Cables.

The RS232 port is the middle port on the back panel of the PSI BOX. To connect to the PSI BOX RS232 port you will need a cable terminating in a 25 way female D connector. It is a sound practice to use cables with screws fitted that will allow you to fasten the cable securely to the PSI BOX.

In general, you will need to make up a "cross over" cable to correctly interface the PSI BOX to the RS232 port of another computer.

Traditionally, making up the cross over cable has been considered a black art. However, provided you have the pin outs and handshake requirements of both sides of your RS232 connection, the cross over cable becomes a matter of common sense.

The cross over cable is simply to ensure that the right signals going out of one RS232 port go into the appropriate lines of the other RS232 port.

Connecting To An IBM PC Serial Port.

Suppose we want to connect the PSI BOX to the serial port of an IBM PC. See Figures 5-12, 5-13 and 5-14.

1) Connect up the earth lines.
   Lines 1 & 7 of the PSI BOX to lines 1 & 7 of the PC.
   This gives the two devices a common earth level.
2) Connect the Transmit and Receive lines together.
   Line 2, TxD, of the PSI goes to line 3, RxD, of the PC. Line 3, RxD, of the PSI goes to line 2, TxD, of the PC.
   So each receives the data transmitted by the other.
3) Connect up the PSI DTR line, pin 20 to the PC DCD, pin 8 and CTS, pin 5, lines.
   Also, connect up the PC DTR line, pin 20 to the PSI DCD, pin 8 and CTS, pin 5, lines.
   This allows the receiving device to signal when it can no longer accept data. The receiving device sets DTR false

when it it is unable to receive any more data. The sending device reads DTR on its CTS and DCD pins. It should stop sending when CTS goes false.

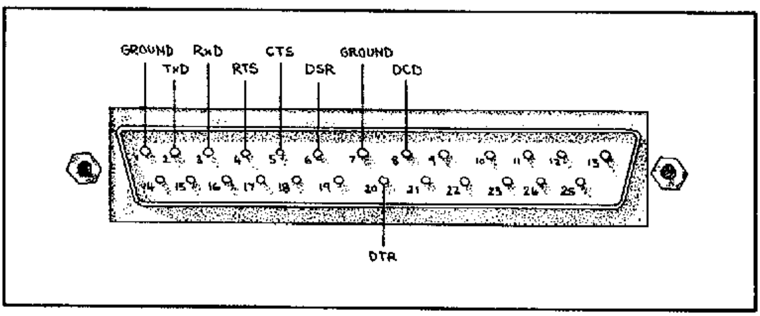
4) Connect the PSI RTS pin 4, to the PC DSR line, pin 6.
Also, connect up the PC RTS line, pin 4, to the PSI DSR line, pin 6.
This RTS line is used to let the other device know that it is ready for data exchange.

Connecting To A Modem.

If you are connecting a MODEM to the PSI BOX then you will not need a cross over cable and a straight through cable connecting pins 1-8 and 20 is all that is required.

Figure 5-7. The RS232 Port Pin Outs.

PIN	SIGNAL	DIRECTION
1	GROUND	N/A
2	TxD TRANSMITTED DATA	OUTPUT FROM PSI BOX
3	RXD RECEIVED DATA	INPUT TO PSI BOX
4	RTS REQUEST TO SEND	OUTPUT FROM PSI BOX
5	CTS CLEAR TO SEND	INPUT TO PSI BOX
6	DSR DATA SET READY	INPUT TO PSI BOX
7	GROUND	N/A
8	DCD DATA CARRIER DETECT	INPUT TO PSI BOX
20	DTR DATA TERMINAL READY	OUTPUT FROM PSI BOX
9-19	NO CONNECTION	
21-25	NO CONNECTION	



Loop Back Connector.

A loop back connector can be used to echo RS232 data transmitted by the PSI BOX back into the RS232 receiver. This is of use in Mode 1 and 2 when data written to the PSI BOX can be read back by the IEEE controller. In this way the function of the serial port can be tested.

A loop back connector is a female 25 pin D connector that has the three groups of signals shorted together as in Figure 5-8.

Figure 5-8. The Loop Back Connector.

PIN	SIGNAL	DIRECTION
2	TxD TRANSMITTED DATA	OUTPUT FROM PSI BOX
3	RXD RECEIVED DATA	INPUT TO PSI BOX
4	RTS REQUEST TO SEND	OUTPUT FROM PSI BOX
5	CTS CLEAR TO SEND	INPUT TO PSI BOX
6	DSR DATA SET READY	INPUT TO PSI BOX
8	DCD DATA CARRIER DETECT	INPUT TO PSI BOX
20	DTR DATA TERMINAL READY	OUTPUT FROM PSI BOX

Figure 5-9. PSI Box To PC Serial Port Cable.

PSI BOX SERIAL PORT Side
USE 25 Pin Female D Connector

IBM PC SERIAL PORT Side.
25 Pin Female D Connector

SIGNAL	PIN	PIN	SIGNAL
GROUND	1	1	GROUND
GROUND	7	7	GROUND
TXD	2	3	RXD
RXD	3	2	TXD
RTS	4	6	DSR
DSR	6	4	RTS
DTR	20	8	DCD
		5	CTS
CTS	5	20	DTR
DCD	8		

Figure 5-10. Connecting To The RS232 Port.

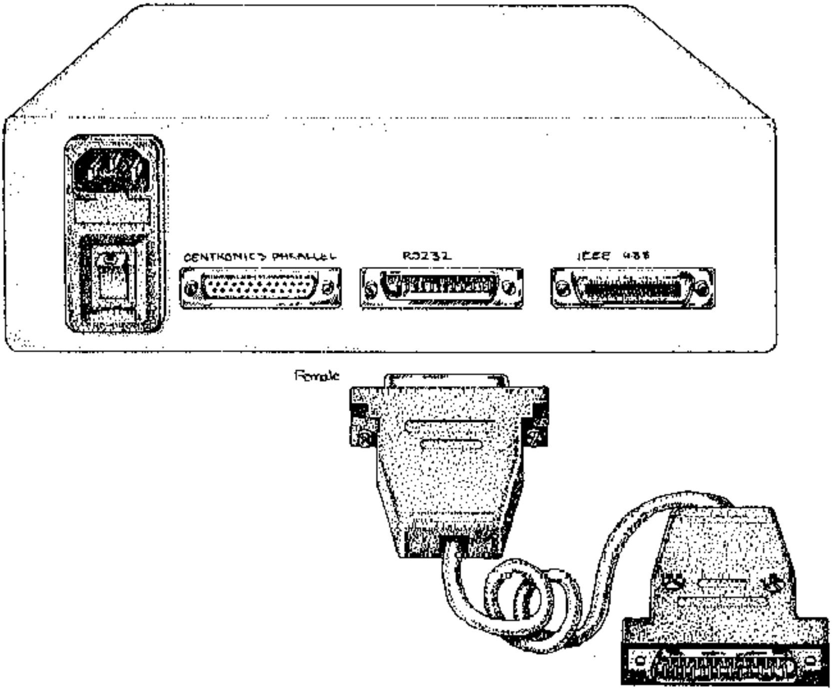


Figure 5-11. PSI Box To AT Serial Port Cable.

PSI BOX SERIAL PORT Side
USE 25 Pin Female D Connector

IBM PC SERIAL PORT Side.
9 Pin Female D Connector

SIGNAL	PIN	PIN	SIGNAL
GROUND	1	5	GROUND
GROUND	7		
TXD	2	2	RXD
RXD	3	3	TXD
RTS	4	6	DSR
DSR	6	7	RTS
DTR	20	1	DCD
		8	CTS
CTS	5	4	DTR
DCD	8		

Figure 5-12. IBM AT To PC Serial Port Cable.

```
 IBM AT SERIAL PORT Side                  IBM PC SERIAL PORT Side.
USE 9 Pin Female D Connector              25 Pin Male D Connector

SIGNAL   PIN                                    PIN   SIGNAL
GROUND 5 ─────────────────────────────────┬──── 1     GROUND
                                          └──── 7     GROUND

DCD   1 ─────────────────────────────────────── 8     DCD
RXD   2 ─────────────────────────────────────── 3     RXD
TXD   3 ─────────────────────────────────────── 2     TXD
DTR   4 ─────────────────────────────────────── 20    DTR
DSR   6 ─────────────────────────────────────── 6     DSR
RTS   7 ─────────────────────────────────────── 4     RTS
CTS   8 ─────────────────────────────────────── 5     CTS

RI   9 ──────────────────────────────────────── 22    Ring Indicator.
```

This cable makes the IBM AT 9 pin serial port look like the IBM PC 25 pin serial port. It is NOT a cross over cable! To make this work with the PSI Box you will also require the cable in Figure 5-12.

The Cable in Figure 5-13 will do the job by itself or else use BOTH cable Figure 5-12 with cable Figure 5-14

CHAPTER 6

THE PSI BOX
MODE 1-6 & 8

Introduction.

This chapter explains, in detail, the 8 PSI BOX operating modes, the DIP switch settings used to select each mode, the exact nature of the data transfer and applications of each mode.

The Mode Switches.

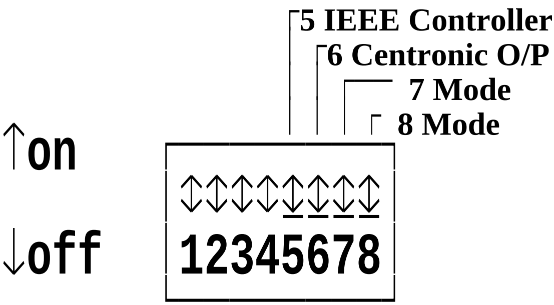
The mode switches determine the type of data conversion the PSI BOX is to perform and specify which port is supplying data and which is accepting data. DIPs 5, 6, 7 & 8 on switch block 1 are the mode control switches.

The IEEE Controller switch, DIP 5, and the Centronics O/P switch, DIP 6, are mode switches which have a dedicated role. DIP switches 7 & 8 on switch block 1 have no dedicated function but are used in conjunction with the DIPs 5 & 6 to define the mode of operation of the PSI BOX.

With the exception of the Pet to Ascii character conversion, the DIP switches are only read when the PSI BOX is powered up. Changing the DIP switch setting whilst the PSI BOX is running will have no effect. If you wish to change the mode of operation of the PSI BOX you must power it off, reset the switches and power it on again.

In Mode 1 and 2 Commodore Pet to Ascii conversion is selected by the state of Mode DIP 8 at the time the data is received. In this way the Pet to Ascii conversion can be changed on the fly, without the necessity of powering the PSI BOX off and on.

Figure 6-1 The Mode DIP Switches Settings.



MODE	IEEE CONT DIP 5	CENT O/P DIP 6	MODE DIP 7	MODE DIP 8	FUNCTIONS SELECTED
1	0	0=Off	0	0	IEEE TO RS232 BI-DIRECTIONAL, IEEE TO CENTRONICS INPUT.
1P	0	0	0	1	MODE 1 With Pet To PC Ascii Conversion.
2	0	1=On	0	0	IEEE TO RS232 BI-DIRECTIONAL, IEEE TO CENTRONICS OUTPUT.
2P	0	1	0	1	MODE 2 With Pet To PC Ascii Conversion.
3	1	0	0	0	CENTRONICS INPUT TO RS232 OUTPUT.
4	1	1	0	0	RS232 INPUT TO CENTRONICS OUTPUT.
5	1	0	1	0	CENTRONICS INPUT TO IEEE OUTPUT.
6	1	1	0	1	RS232 INPUT TO IEEE OUTPUT.
7	1	1	1	1	RS232 TO IEEE BI-DIRECTIONAL. ADVANCED CONTROL LANGUAGE ALLOWS FULL USE OF INSTRUMENTATION. PSI BOX PLUS+ ONLY
8	1	0	0	1	Hewlett Packard Plotter Mode. RS232 INPUT TO IEEE OUTPUT. IEEE TALK BACK TO RS232 OUTPUT

Invalid DIP combinations, or a standard PSI BOX configured for Mode 7, which is only found in the PSI BOX PLUS+, cause the PSI BOX to flash all its LEDs in sequence to indicate an error.

Mode 1.

Function: **An IEEE to RS232 bi-directional and an IEEE from Centronics input interface.**
Also Listen Only IEEE to RS232 interface.
Switchable CBM Pet to PC Ascii conversion.

Use: **PSI makes RS232 and parallel equipment function as devices attached to IEEE controller.**

Example: **HP or CBM PET IEEE computer to RS232 laser printer or modem.**
IBM PC parallel port data to IEEE computer.

Summary.

The PSI BOX acts as an IEEE bus device taking its instructions from an IEEE controller such as a Hewlett-Packard 9825 or Commodore Pet computer, an RS232 device is connected to the PSI RS232 port or the output from the parallel printer port of a PC is connected to the parallel input port of the PSI BOX. The PSI BOX acts like two separate IEEE devices, one device is an IEEE to RS232 interface the other device is an IEEE from parallel interface.

Listen Only Operation.

The PSI BOX can also be set to act as a Listen Only IEEE to RS232 output interface. To select the Listen Only configuration set the Mode switches to Mode 1 and set the IEEE address dip switches all upward, ie address 15. Instead of the PSI BOX becoming IEEE device 15 it will be in Listen Only mode. Here the a controller instructs one device to talk and none, one or more devices to listen, the PSI BOX will listen in to all the data transmitted on the bus irrespective of the IEEE device that is addressed to listen, and send it to the RS232 port.

In addition, the PSI BOX will happily listen to a Talk Only device, in a system without a controller, that does not provide any addressing information.

Addressed Operation.

The address of the RS232 input output port is as set on the IEEE address DIP switches. The Centronics port address is one greater than the RS232 address.

Data written to the RS232 address will be buffered and sent to the RS232 when the RS232 handshake permits. If the RS232 output buffer is full then the IEEE handshake will freeze, with NRFD and NDAC held low, waiting for the RS232 device to become ready to receive the contents of the PSI BOX buffer. This ensures that no data is lost!

When data is requested from the RS232 port the PSI BOX sends any data that is in its RS232 input buffer. The last byte from the buffer is sent with EOI true. This informs the controller that the buffer is now empty.

If a byte is requested from the PSI BOX and there is nothing in its buffer to send then a ’talk with nothing to say’ message is sent, this is the 00 byte with EOI true. This method of flagging an empty buffer prevents a hang-up in the handshake.

In a similar manner, when data is requested from the Centronics input port the PSI BOX sends any data from the buffer. When the PSI BOX sends the last byte in the buffer it sets EOI true, indicating to the controller that there is no more to get. If data is requested when the buffer is empty then a talk with nothing to say message is generated.

Any data SENT to the PSI BOX for output via Centronics port is lost since in Mode 1 the Centronics port is input only.

In Mode 1 half the PSI BOX ram acts as the Centronics input buffer, the other half is evenly split between the RS232 input and output buffers.

Pet To PC Ascii Conversion.

From version 4.3 onwards of the PSI BOX, Commodore Pet to PC Ascii conversion will be performed depending on the state of mode DIP switch 8. When the switch is UP, ie ON, all out going data to the RS232 port will be converted from the Pet character to standard PC Ascii. All input from the RS232 and Centronics ports will be converted from PC Ascii to Petscii. The output conversion is generally used to connect standard RS232 Ascii printers to CBM Pet or C64/128 computers.

Unlike all other PSI BOX modes, in Mode 1 the state of DIP switch 8 IS monitored whilst the PSI BOX is in operation. This means that the conversion can be switched in and out at will whilst the PSI BOX is on. Data already in the buffer will not be changed only the new incoming data will be converted according to the setting of DIP 8.

Mode 2.

Function: **An IEEE to RS232 bi-directional and an IEEE to Centronics output interface.**
Switchable CBM Pet to Ascii conversion.
Use: **PSI makes RS232 and parallel equipment function as devices attached to IEEE controller.**
Example: **HP or CBM PET IEEE computer to RS232 laser printer or modem.**
HP or CBM PET IEEE computer to parallel plotter or laser printer.

Summary.

The PSI BOX acts as an IEEE bus device taking its instructions from an IEEE controller such as a Hewlett-Packard 9825 or Commodore Pet computer, an RS232 device is connected to the PSI RS232 port or a parallel printer or plotter is connected to the parallel output port of the PSI BOX. The PSI BOX acts like two separate IEEE devices, one device is an IEEE to RS232 interface the other device is an IEEE to parallel interface.

Listen Only Operation.

The PSI BOX can also be set to act as a Listen Only IEEE to Centronics output interface. To select the Listen Only configuration set the Mode switches to Mode 2 and set the IEEE address dip switches all upward ie address 15. Instead of the PSI BOX becoming IEEE device 15 it will be in Listen Only mode. Here the a controller instructs one device to talk and none, one or more devices to listen, the PSI BOX will listen in to all the data transmitted on the bus irrespective of the IEEE device that is addressed to listen, and send it to the Centronics port.

In addition, the PSI BOX will happily listen to a Talk Only device, in a system without a controller, that does not provide any addressing information.

Addressed Operation.

The address of the RS232 input output port is as set on the IEEE address DIP switches. The Centronics port address is one greater than the RS232 address.

The description of the RS232 port is identical to Mode 1.

Any data sent to the Centronics port is buffered in the PSI BOX. The PSI BOX sends this buffered data when the Centronics handshake permits. If the Centronics output buffer is full then the IEEE handshake will freeze, with NRFD and NDAC held low, waiting for the Centronics device to become ready to receive the contents of the PSI BOX buffer. This ensures that no data is lost!

Attempts to data read from the PSI BOX Centronics port causes the ’talk with nothing to say’ message to be sent, this is the 00 byte with EOI true, since in Mode 2 the Centronics port is OUTPUT only. This prevents a hang-up in the handshake.

In Mode 2 half the PSI BOX ram acts as the Centronics output buffer, the other half is evenly split between the RS232 input and output buffers.

Pet To PC Ascii Conversion.

Commodore Pet to PC Ascii conversion will be performed depending on the state of mode DIP switch 8. When the switch is UP, ie ON, all out going data to the RS232 port and the Centronics port will be converted from the Pet character to standard PC Ascii. All input via the RS232 port will be converted from PC Ascii to Petscii. The output conversion is generally used to connect standard Centronics or RS232 Ascii printers to CBM Pet or C64/128 computers.

Unlike all other PSI BOX modes, in Mode 2 the state of DIP switch 8 IS monitored whilst the PSI BOX is in operation. This means that the conversion can be switched in and out at will whilst the PSI BOX is on. Data already in the buffer will not be changed only the new incoming data will be converted according to the setting of DIP 8.

The Mode 1+2 IEEE Handshake.

The controlling computer needs to address the PSI BOX as a listener or a talker, at the end of a data transfer sequence the PSI BOX should be un-addressed by the unlisten or the untalk command. When the PSI BOX is a listener, receiving data, it recognises that EOI is the end of stream terminator. When addressed to talk the PSI BOX uses EOI to flag the end of its data stream to the controller.

In modes 1 and 2 the Data transfer to the IEEE port from the controlling computer is:

PSI BOX listen address, from IEEE controller computer.
Data bytes.. from IEEE controller computer.
Last data byte + EOI from IEEE controller computer.
Unlisten, from IEEE controller computer.

In modes 1 and 2 the Data from the IEEE port to the controlling computer is:

PSI BOX talk address, from IEEE controller computer.
Data bytes.. from PSI.
Last data byte + EOI from PSI box.
Untalk, from IEEE controller computer.

The Mode 1+2 Secondary Addresses.

When working in Mode 1 and Mode 2 the IEEE port recognises several secondary addresses.

The IEEE device number, the IEEE primary address, of the PSI BOX is as set on the front panel DIP switches. Secondary addresses are specified in the OPEN statement that is sent from the CBM PET computer or HP computer. By requesting or sending data using these special secondary addresses the PSI Box can communicate its status and other information to the IEEE controlling computer. The secondary address in use are as follows:-

<u>S.A.</u>	<u>Talk Function. PSI Commanded To Talk By Controller.</u>
0-12	No special function. PSI BOX sends its buffered data.
13	PSI BOX sends one byte indicating buffer contents status. 00 means no data in PSI BOX buffer. 01 means some data in PSI BOX buffer.
14	PSI Box sends the state of the RS232 port pins. The PSI Box returns one byte which holds the state of the handshake pins of the RS232 port. A 0, zero bit means that the line is held true, +12 Volts. A 1, one bit means that the line is held false, -12 Volts. DTR is bit 0 decimal value 1 an output from PSI Box. RTS is bit 1 decimal value 2 an output from PSI Box. DSR is bit 2 decimal value 4 an input to PSI Box. CTS is bit 3 decimal value 8 an input to PSI Box. DCD is bit 4 decimal value 16 an input to PSI Box.
15	PSI Box sends an identification message, memory size and software revision date eg P.S.I. BOX V4.3 02/05/90 32K.

<u>S.A.</u>	<u>Listen Function. PSI Commanded To Listen By Controller.</u>
0-12 +15	No special function. Data is buffered for output to port.
13	PSI BOX flushes the RS232 or Centronics buffer. PSI BOX accepts one byte of data from the controller. Bit 0 decimal value 1 set means flush output buffer. Bit 1 decimal value 2 set means flush input buffer.
14	RS232 only. The PSI BOX accepts one byte which specifies the level of the RS232 port output pins RTS and DTR. A 0, zero bit causes the line to be held true, +12 Volts. A 1, one bit causes the line to be held false, -12 Volts. DTR is bit 0, decimal value 1, an output from PSI BOX. RTS is bit 1, decimal value 2, an output from PSI BOX. **The state of the Input cannot be set but only read.**

Mode 3.

Function: **Centronics input to RS232 output interface.**

Use: **PSI connects RS232 printers and plotters to the parallel printer port of any computer.**

Example: **HP laser printer to IBM PC parallel port.**

Summary.

The PSI BOX acts as a one way data converter reading information from a computer’s printer port port and passing it straight through to an RS232 serial printer or plotter. This is a popular way of connecting the HP Laser Jet printer, via a buffer, to the parallel port of a PC.

Data is received according to the Centronics input handshake and output according to the RS232 output handshake. In this mode no other sort of data transmission is possible. The IEEE port is disabled and does not respond to any commands or data.

If the Centronics input buffer of the PSI BOX becomes full then it will keep the BUSY line high until the RS232 device accepts data from the PSI BOX.

In Mode 3 the whole of the PSI BOX ram acts as the Centronics input buffer.

Mode 4.

Function: **RS232 input to Centronics output interface.**

Use: **PSI connects parallel printers to the RS232 port of a computer.**

Example: **Parallel plotter or laser printer running from COM1 port of a PC.**
Enables parallel printer to be 50 feet or more remote from the host computer.

Summary.

The PSI BOX acts as a one way data converter reading RS232 information from a computer's serial port and passing it straight through to a parallel printer or plotter. This is often used as a method of placing parallel printers etc in a remote location with the data being sent down a twisted pair RS232 line at high speed.

Data is received according to the RS232 input handshake and output according to the Centronic output handshake. In this mode no other sort of data transmission is possible. The IEEE port is disabled and does not respond to any commands or data.

If the RS232 input buffer of the PSI BOX becomes full then it will signal unable to receive more RS232 data until the Centronics device accepts data from the PSI BOX.

In Mode 4 the whole of the PSI BOX ram acts as the RS232 input buffer.

Mode 5.

Function: **A Centronics input to IEEE output interface.**

Use: **PSI connects an IEEE output device to the parallel printer port of a computer.**

Example: **HP 7475 IEEE plotter from printer port of a PC running LOTUS 123, AutoCAD etc.**
Commodore IEEE printer from PC printer port.

Summary.

The PSI BOX acts as a one way data converter reading information from a computer's printer port and passing it straight through to an IEEE bus output device.

Data is received according to the Centronics input handshake and output according to the IEEE output handshake. In this mode no other sort of data transmission is possible.

Talk Only Operation.

The PSI BOX can also be set to act as a Talk Only IEEE from Centronics parallel input interface. To select the Listen Only configuration set the Mode switches to Mode 5 and set the IEEE address dip switches all upward ie address 15. Instead of the PSI BOX becoming IEEE device 15 it will be in Talk Only mode. Here the a controller instructs one or more devices to listen, the PSI BOX will take data from its Centronics parallel input buffer and send it to all the listeners on the bus. The controller should not instruct any device other than the PSI BOX to Talk.

In addition, the PSI BOX will happily Talk to a Listen Only device, in a system without a controller, that does not provide any addressing information.

Addressed Operation.

The PSI BOX acts as the IEEE controller and is the source of all IEEE commands. It controls the flow of data on the IEEE bus by addressing and un-addressing devices. Only one IEEE device can be accessed when in this mode.

The address of the IEEE device that the PSI BOX accesses is as set on the IEEE address DIP switches. The PSI BOX sends a LISTEN command to the IEEE bus before each data byte is sent.

If the Centronics input buffer of the PSI BOX becomes full then it will hold the BUSY line high until the IEEE device accepts data from the PSI BOX.

In Mode 5 the whole of the PSI BOX ram acts as the Centronics input buffer.

Mode 6.

Function: **An RS232 input to IEEE output interface.**

Use: **PSI connects an IEEE output device to the RS232 port of a computer.**

Example: **HP IEEE Thinkjet printer from RS232 port of a PC running LOTUS 123, AutoCAD etc.**
Commodore IEEE printer from PC serial port.

Summary.

The PSI BOX acts as a one way data converter reading RS232 information from a computer's serial port and passing it straight through to an IEEE bus output device.

Data is received according to the RS232 input handshake and output according to the IEEE output handshake. In this mode no other sort of data transmission is possible.

Use Mode 8 for IEEE plotter control from a computer's RS232 serial port.

Talk Only Operation.

The PSI BOX can also be set to act as a Talk Only IEEE from RS232 input interface. To select the Listen Only configuration set the Mode switches to Mode 6 and set the IEEE address dip switches all upward ie address 15. Instead of the PSI BOX becoming IEEE device 15 it will be in Talk Only mode. Here the a controller instructs one or more devices to listen, the PSI BOX will take data from its RS232 input buffer and send it to all the listeners on the bus. The controller should not instruct any device other than the PSI BOX to Talk.

In addition, the PSI BOX will happily Talk to a Listen Only device, in a system without a controller, that does not provide any addressing information.

Addressed Operation.

The PSI BOX acts as the IEEE controller and is the source of all IEEE commands. It controls the flow of data on the IEEE bus by addressing and un-addressing devices. Only one IEEE device can be accessed when in this mode.

The address of the IEEE device that the PSI BOX accesses is as set on the IEEE address DIP switches. The PSI BOX sends a LISTEN command to the IEEE bus before each data byte is sent.

If the RS232 input buffer of the PSI BOX becomes full then it will signal unable to receive more RS232 data until the IEEE device accepts data from the PSI BOX.

In Mode 6 the whole of the PSI BOX ram acts as the RS232 input buffer.

Mode 8.

Function: **An RS232 input to IEEE output interface with IEEE talk back and RS232 flow control.**

Use: **Driving a IEEE plotter from the serial RS232 port of a computer so that it completely simulates an RS232 plotter.**

Example: **HP 7475 IEEE plotter from RS232 port of a PC running LOTUS 123, AutoCAD etc.**

Summary.

Many plotters understand Hewlett Packard Graphic Language, HPGL, amongst these are HP's RS232 7475 and IEEE 7475 plotter. There are three essential types of operation that RS232 HPGL plotters perform.

1) Execution of HPGL plotting commands.
2) Reply to HPGL OUTPUT commands.
3) Response to RS232 interface Flow Control commands.

1) Execution Of HPGL Plotting Commands.

Most HPGL commands are concerned with positioning and selecting pens, plotting shapes and specifying line thickness, pen speeds etc. These HPGL commands are routed from the host computer directly to the plotter for execution. The PSI BOX buffer, in conjunction with any buffer in the plotter, are used to accept as large an amount of data from the computer as quickly as possible thus freeing up the computer for other processing.

2) Reply To HPGL Output Commands.

The HPGL specification allows plotters can be asked to output to the computer various information. This includes the pens current coordinates, size of the drawing window, error states, plotter identification information etc. Several of the output commands are listed below.

Figure 6-2 HPGL Output Commands.

```
OA    Output Actual.
OC    Output Coordinates.
OD    Output Digitised Point.
OE    Output Error.
OI    Output Identity.
OP    Output P1 And P2.
OS    Output Status.
OW    Output Window.
```

The talk back facility of Mode 8 allows the plotter to receive all these output commands and reply to them. When the PSI BOX RS232 input buffer is empty, the PSI BOX polls the plotter for a response to the computers request after every three second period of inactivity on the RS232 bus. This mechanism allows any data waiting in the plotter buffer to be correctly received by the computer within a short time of it becoming available.

3) HPGL RS232 Interface Flow Control Commands.

Whilst the plotting and output commands that the two plotters understand are identical the RS232 plotter also responds to several FLOW CONTROL commands that are absent from IEEE plotters. These Flow Control commands regulate the transfer of information along the RS232 interface between the computer and the plotter. Thus the computer can clear the plotters buffer, request the plotter reply with the size of the plotters, return an error status and set the output mode of data between the plotter and the computer. IEEE plotters do not require these functions but software packages driving RS232 plotters may expect the plotter to behave as if it does have them. Mode 8 of the PSI BOX allows any IEEE plotter to completely emulate the Flow Control of an RS232 plotter.

The Flow Control commands are three byte commands that all start with the ESCAPE character, decimal 27 followed by the decimal point character. The third character identifies the specific flow control function. Depending on the command, various parameters may be specified and a response from the PSI BOX may be expected.

Figure 6-3 HPGL RS232 I/f Flow Control Commands.

```
ESC.( Turn plotter On.
ESC.Y Turn plotter On.
ESC.) Turn plotter Off.
ESC.Z Turn plotter Off.
ESC.@ Set monitor mode.
ESC.B Output Available Buffer Size.
ESC.E Output Error Status Code.
ESC.H Handshake Mode 1.
ESC.I Handshake Mode 2.
ESC.J Abort Flow Control Command.
ESC.K Abort Plot, Clear Buffer.
ESC.L Output Maximum Buffer Size.
ESC.M Specify Output Mode.
ESC.N Specify Output Mode.
ESC.O Output Buffer Status Code.
ESC.R Initialise Handshake Parameters.
```

Addressed Operation.

Since Mode 8 requires the plotter to both listen to HPGL plotting commands and reply to HPGL output command the plotter must be set for addressed operation. It must not be set for Listen only use.

The PSI BOX acts as the IEEE controller and is the source of all IEEE commands. It controls the flow of data on the IEEE bus by addressing the plotter to talk or listen. Only one IEEE device, the plotter, can be accessed when in this mode.

The address of the plotter that the PSI BOX accesses is as set on the IEEE address DIP switches. The PSI BOX sends a LISTEN command to the plotter before each HPGL command is sent, when the trailing CR character is detected the PSI BOX waits ready to address the plotter again.

If no RS232 bus activity has occurred for 3 seconds, and the PSI BOX RS232 input buffer is empty then the PSI BOX polls the plotter for any response it may have. The plotter is addressed to TALK any response is sent back to the host computer via the RS232 port. At the end of the plotters output the plotter is commanded to UNTALK. At every 3 second interval the plotter will be repolled for further responses.

If the PSI BOX receives a Flow Control command then it immediately responds to it in the appropriate way.

If the RS232 input buffer of the PSI BOX becomes full then it will signal that it is unable to receive more RS232 data until the IEEE device accepts data from the PSI BOX.

In Mode 8 the 28,672 bytes of the PSI BOX ram acts as the RS232 input buffer with 2048 bytes for the RS232 output buffer.

CHAPTER 7
AN OVERVIEW OF THE
PSI BOX
MODE 7

Introduction.

This chapter is an overview of the operation of the PSI BOX in mode 7, precise details about the syntax and parameters of the IEEE commands language are to be found in the next chapter.

As well as a short summary of the Mode 7 this chapter contains details about the Mode 7 power up messages, the Mode 7 default parameters, data pass through mode, the BUS command sequence, sending data to IEEE devices, receiving data from IEEE devices, IEEE control status string, the RESET command and character, the current device, a summary of the BUS command language and finally the Mode 7 memory allocation.

Mode 7 In 15 Minutes!.

A good way of using the PSI BOX in Mode 7 is interactively from the keyboard of your PC via a terminal emulation program or a modem program such as one of the many share ware programs like PC TALK or PROCOM. In this way the user can get a good feel of the power and ease of use of the Mode 7 command language in a very short time.

To get up and running within the shortest possible time we have included two terminal emulation programs with the PSI BOX PLUS+. These are written in PC BASIC and come compiled ready to run and also as source code for you to modify as you wish. Both programs have been complied using Microsoft's excellent Quick Basic V4.5 compiler, the source code is also compatible with other compiler such as Borland's Turbo Basic. The two programs are called COMM and PSIBOX.

COMM.EXE

The COMM.BAS file is the source code for the compiled COMM.EXE program. The COMM program is a very simple, bare bones program, with just the minimum needed code to talk to the PSI BOX. It opens COM1 to talk to the PSIBOX at 1200 baud, 7 data bits, Even Parity and 1 stop bit. Any key pressed on the PC, is echoed on the screen and is then sent to the PSIBOX. Every character received from the PSI BOX is printed on the PC screen. To change the communications parameters it is necessary to edit the source code and either recompile or run the program using the BASIC interpreter.

PSIBOX.EXE

The PSIBOX.BAS file is the source code for the compiled PSIBOX.EXE program. The PSIBOX program is more sophisticated and has menu driven serial port parameter selection, predefined Function Keys to access PSI features etc etc. We encourage you to use the programs and you are free to modify and incorporate the code into your own programs provided they are used to drive our products.

Mode 7 Summary.

Mode 7 is an RS232 to IEEE bi-directional interface. Data is sent and received using the RS232 and IEEE handshakes. In this mode no other sort of data transmission is possible, the parallel port can not be used as either an input nor as an output. Typically mode 7 is used to control several IEEE devices or instruments from the RS232 port of a computer. eg Commodore Amiga IEEE controller, BBC Master IEEE controller, Acorn Archimedes IEEE controller, Atari ST IEEE controller or IBM

PS/2 IEEE controller.

The PSI BOX acts as the IEEE controller and is the source of all IEEE commands. It controls the flow of data on the IEEE bus by addressing and un addressing devices to talk or listen. Up to 14 IEEE devices may be accessed when in this mode.

A sophisticated command language is used to control the IEEE bus via the PSI BOX. It is similar in concept to the HAYES modem control language, and the commands and syntax are based on those used in the our Professional 488 Device Driver for our range IBM PC and PS/2 compatible IEEE interface. The character sequence that invokes the IEEE command language is <CR> BUS
Characters sent to the PSI BOX that do not start with the <CR> BUS sequence are treated as data to be sent direct to the current IEEE device, the data is thus passed straight through from the RS232 port to the IEEE bus.

Mode 7 Power Up.

The basis of the Mode 7 IEEE handler is an intelligent command handler. This is a piece of software resident on the PSI BOX rom. It is called into operation automatically whenever the PSI BOX is powered up or reset when DIP switches are set to Mode 7 as shown in the PSI BOX manual page 7 Figure 4.

Once the PSI BOX has been powered on in Mode 7 you can access the IEEE bus in two ways. By default data sent to the PSI BOX from the RS232 computer is passed though to the current IEEE device. Alternatively the data sent to the PSI BOX is interpreted as a Mode 7 BUS command.

When the PSI BOX is switched on in MODE 7 it sets up and maintains a list of parameters, the IEEE control status string. These parameters describe the current IEEE address, secondary address, state of ATN, SRQ and REN lines, input and output byte counts, SERIAL POLL and PARALLEL POLL status bytes, EOS and END bytes etc.

The IEEE control status string is very important, its format is identical to the PC IEEE control status string. Changing the parameters in the IEEE status string controls the operation of the PSI BOX, the source and destination of IEEE data exchanges and many other vital criteria.

The precise format of the data in the IEEE control status string is given on the PSI BOX Mode 7 Quick Reference Card and is given later on in this chapter.

On power up in mode 7 the PSI BOX sends a string of data to the RS232 computer. This message consists of the current PSI BOX rom version and the default IEEE control status string.

```
PSI BOX Version 4.0f 16/03/90
STATUS 04,  ,0,1,1,0,00,00000,00000,000,000,000,0,0,1 etc
```

The ’my listen’ and the ’my talk’ address of the PSI BOX are by default zero.

The address of the IEEE current device that the PSI BOX is to access is that set on the IEEE address DIP switches, and is the first field in the status string. The current device can be changed at any time from the RS232 computer using a BUS command. The default secondary address is NONE, this is stored as the second field in the status string.

The Default Mode 7 Configuration.

The default PSI BOX Mode 7 configuration as it powers up without any changes being made via the BUS command. Is given below.

Figure 7-1 The Default Mode 7 Configuration.

```
DEFAULT IEEE DEVICE NUMBER          =AS SET BY DIP SWITCHES
DEFAULT IEEE SECONDARY ADDRESS      =NONE
TIME OUT VALUE                      =10 (1 second)
PSI BOX MY LISTEN ADDRESS           =00
PSI BOX MY SECONDARY ADDRESS        =NONE
PSI BOX MY TALK ADDRESS             =00
END OF INPUT TERMINATION (EOS=)     =EOI ASSERTED BY DEVICE
END OF OUTPUT METHOD (END=13)       =EOI SENT WITH <CR>
RS232 PORT ECHO STATE               =OFF. ECHO=OFF
EMERGENCY RESET CHARACTER           =@. DECIMAL 64
INTERFACE CLEAR LINE IFC            =PULSE TRUE>100 micro sec
REMOTE ENABLE LINE REN              =ASSERTED TRUE
ATTENTION LINE ATN                  =ASSERTED TRUE C-A-C
```

Data Pass Through Mode.

By default any data, that does not start with the BUS command sequence, is treated as data to passed straight though the PSI BOX to the current IEEE device.

At the start of the data stream the PSI BOX sends the current device LISTEN address. Then the current secondary address, if any. Then the data received from the RS232 host computer is sent byte by byte. The EOI will be set true concurrently with a data byte and will be followed by the UNL command according to the End Of Sequence method specified by the EOS= command.

Any data that starts with a <Carriage Return> or <Line Feed> character will be buffered until it has been checked for the BUS command sequence. Thus the data sequence:-
<CR> BUFFET
will have the three bytes <CR> BU stored by the PSI BOX until the F character arrives, as this point the PSI BOX determines that <CR> BUS has not been received from the host RS232 computer. Therefore the PSI BOX sends <CR>BUF to the current device on the IEEE bus and the later characters FET are sent as soon as they have been received.

For unambiguous data output to the IEEE bus devices use the OUTPUT command.

The BUS Command Sequence.

All data output from the controlling computer's RS232 port is inspected for the following sequence of bytes:-

<Carriage Return> BUS

Here B is the Ascii character decimal 66.
U is the Ascii character decimal 85.
S is the Ascii character decimal 83.
<Carriage Return> is Ascii character decimal 13.
One or more Carriage Returns and Line Feeds, Ascii character 10, may be inserted before the BUS characters.

After receiving the <CR> BUS sequence the data following up until a trailing <CR> with optional <LF> is treated as an Ascii string containing IEEE bus keywords and numeric parameters. These keywords are interpreted as commands to perform Serial or Parallel Polls, send any IEEE bus commands or specify the timeout period.

The command interpreter checks the syntax of your IEEE string and responds with 'All done OK' or 'Error in command' as appropriate.

If an error has been detected then all the commands in the received string prior to the error will have been executed. The exact place the error occurred is given by the field in the IEEE control status string positioned at bytes 44-46. The value in this field is the position of the byte in the previous BUS command that the PSI BOX did not recognise, the PSI BOX starts counting from the B in BUS as position 1.

In addition to the PSI BOX responding with the message 'Error In Command' there is a syntax error flag is the byte at position 42, it is 1 when there was a syntax error in the previous string and 0 when no syntax error was detected. This syntax error flag is of most used for programmed applications whilst the message ’Error In Command’ is particularly effective during interactive use.

If any new line of information starts with the characters BUS, in upper case, then that line is treated as a PSI BOX command. The PSI BOX processes the command and takes any necessary action.

If any line of information does not start with BUS then the PSI BOX treats it as data to be output to the current IEEE device.

A new line is said to start after a CR or CR+LF character sequence.

Every BUS command is terminated by a carriage return. All the parameters in an BUS command are numerals 0-9 or are any mix of upper or lower case letters, a-z and A-Z, with the +"= characters. Parameters are separated by spaces or commas. Extra spaces can be used to separate parameters to make the commands more readable. These extra spaces are ignored by the PSI BOX command parser.

The BUS command can contain one or more valid commands in any sequence desired provided that not more than 128 characters are sent at any one time. Remember a <CR> must be used to terminate the string.

BUS SPOLL 6 Listen 5 DATA 65,76,80 EOI 13 UNL <CR>

About 60 IEEE commands are available in MODE 7. These are listed below. The PSI BOX Mode 7 Quick reference Card lists the commands and gives the exact BUS command language syntax. A later chapter in this manual gives command syntax as well as the IEEE bus activity and data returns that results from each command.

Sending Data Using The Bus Command.

Data can be sent to the IEEE bus devices in several ways. As mentioned previously data sent in pass through to the current device. Additionally data can be sent to the IEEE bus devices using the BUS command.

The data may be sent in one of three ways:- as specified by decimal numbers after a DATA or EOI command, as literal data within delimiters after a STRING command, as literal data within delimiters after an OUTPUT command.

BUS LISTEN 5 DATA 72,69,76,76,79 UNL <CR>

BUS LISTEN 5 STRING "HELLO" UNL <CR>

BUS OUTPUT "HELLO" <CR>

Whilst these methods the main differences lie in the fact that the addressing of the devices has to be explicitly given when using the DATA and STRING commands but that the OUTPUT command addresses the current device by default.

Receiving Data Using The Bus Command.

Data can be requested from any IEEE bus device using either the READ or ENTER command.

BUS TALK 5 READ UNT <CR>

Like the DATA command the READ commands requires the user to give explicit addresses and unaddresses when receiving data. The READ command reads one data byte from the addressed talker, the byte is stored in the IEEE control status string, position 66-68.

BUS ENTER <CR>
BUS ENTER 200 <CR>

The ENTER command is much more flexible, a complete bus handshake sequence is performed with the current device being addressed to talk. The input sequence is terminated by the receipt of the number of bytes specified or by EOS received. The maximum number of bytes allowed to be requested using the ENTER command is 65535 bytes.

The IEEE Control Status String.

One of the great advantages of the IEEE 488 bus it that up to 16 devices may be simultaneously present on it, without confusion as to which one we want to use. Each device has a unique IEEE address, or device number. Each stream of data that is transferred on the IEEE bus is preceded by addressing bytes that inform the bus which device is to transmit data and which devices are to receive the data. But how do we tell the PSI BOX which device we want next.

The PSI BOX stores a table of data that it uses to inform itself of which devices to access and in what manner it should control the bus. This table of data is called the IEEE control status string. The IEEE control status string also holds information on parallel and serial poll responses, time out data, byte counts of data transferred during the previous i/o and many other interesting parameters. The first field of the IEEE control status string holds the current IEEE device number, the second field is that device’s secondary address.

The PSI BOX status string can be accessed in two formats.
The STATUS command causes the PSI BOX to send the string as a series of ascii numbers separated by comma’s, it is about 128 bytes long.
The SHOW command sends the status string back with explanations. It has each of the fields labeled with text to help explain its function. The format is ideal for display on an 80 character screen, about 1500 characters are sent. The SHOW command is intended for interactive debugging by a person, the STATUS command returns a string that is easier for a program to handle. The large size of the SHOW response, especially

down a low speed RS232 line makes of best use whilst programs are being debugged.

BUS SHOW <Carriage Return>

The first few lines of the SHOW response looks like:-

```
Current Device=  04 Curr Sec Add=NONE Service ReQ =  0 ATtentioN  =  0
Remote ENable =   0 Bus Status  =  00 Input Count =000 Output Count=00
Serial Poll   = 000 Init SP Data= 000 P Poll Data =000 ON SRQ Done = 0
Syntax Error  =   0 Error Pos   = 000 ECHO Flag =    0
My Listen Add =  00 My Secdy Add=NONE My Talk Add = 00 Debug     = 000
Time Out Value=  10 Read Data   = 000 RESET Byte = 064 Address   = 000
EOS Byte      = 013 EOS Bit Mask=   8 END Byte  =  013 END Bit Mask= 8
```

The current PSI BOX status string will be sent from the PSI BOX to the RS232 host computer whenever the PSI BOX receives the STATUS command.

BUS STATUS <Carriage Return>

Alternatively, for increased ease of use, a BUS command without any parameters or keywords will be treated as a STATUS command. Any spaces after BUS but before the <CR> are ignored.

BUS   <Carriage Return>

A request for the STATUS causes the PSI BOX to return the IEEE control status string to the RS232 host computer. This is a string of Ascii characters preceded by the word STATUS and terminated by a CR character, decimal 13. The string is made up of a series of numbers separated by commas and looks like:-

```
STATUS 04,  ,0,1,1,0,00,00000,00000,000,000,000,0,0,1 etc
```

Currently the IEEE control status string is approximately 90 bytes long, as new features are added to the PSI BOX software the length of the control status string increases. It is advised that 256 bytes of space are set aside for the IEEE control status string in your program to allow complete compatibility with future version of the Mode 7 software. The format of the data returned is given in below.

Figure 7-2 The IEEE Control Status String.

BYTE	RANGE FIELD	DESCRIPTION.
1-2	00-30	Current device IEEE device number
4-5	00-30	Current device IEEE secondary address. SPACE SPACE means no sa.
7	0 or 1	SRQ input line. 1 means SRQ is asserted true by a device so requesting service.
9	0 or 1	ATN output line. 1 means that PSI BOX is the active controller asserting attention.
11	0 or 1	REN output line. 1 means that the PSI BOX is asserting REN true. Devices under IEEE control.
13-14	00-15	IEEE bus status field. 4 bits each representing a particular state. Bit0 = time out on o/p, bit1 = time out on i/p, bit2 = EOI/EOS on i/p, bit3 = no device present.
16-2	000000-65535	The number of bytes read from bus on last input.
22-26	00000-65535	The number of bytes sent to the bus on last output.
28-30	000-255	The SERIAL POLL data for the current device, current device address as in bytes 1-2.
32-34	000-255	The SERIAL POLL data for the first device addressed in the previous SPOLL command.
36-38	000-255	The PARALLEL POLL data from the previous parallel poll command.
40	0 or 1	ON SRQ flag. 1 means an ON SRQ has been automatically performed since IEEE data was last read.
42	0 or 1	IEEE control data syntax error. 1 means the previous BUS string written to the PSI BOX contained an error.
44-46	000-255	Syntax error byte. This is the position of the first byte in the previous BUS command that the PSI BOX did not recognise. Starts counting from the B in BUS as 1.
48	0-9	ECHO flag. 0 is the default ECHO=OFF. 1 is ECHO=ON.
50-51	00-30	MLA. The MY LISTEN ADDRESS of the PSI.
53-54	00-30	MSA. The MY SECONDARY ADDRESS of the PSI. SPACE SPACE means no MSA.
56-57	00-30	MTA. The MY TALK ADDRESS of the PSI.
59-61	000-255	Debug field .. reserved.
63-64	00-15	The current time out value. 00= wait for ever till device is ready.

66-68	000-255	READ command returned data field.
70-72	000-255	RESET byte, The PSI BOX does Power On Reset when it receives this byte.
74-76	000-255	Reserved.
78-80	000-255	EOS byte. PSI BOX recognises that this is the last byte in a string sent by a talker. 000 when unused.
82	7 or 8	Number of bits to match in EOS byte. 7 or 8 bits.
84-86	000-255	END byte. PSI BOX only sends EOI true when this byte is sent as last in a stream. When EOS width = 7 or 8
88	0,7 or 8	Number of bits to match in END byte. 7 or 8 bit bytes. If =0 then never send EOI true when PSI BOX sending data!
89		Comma
90		Carriage Return, End marker.
91-255		Reserved for future use. Make sure you allow for these extra bytes.

The RESET Command and Character.

Of particular interest is the RESET command and character. Whenever the PSI BOX receives the reset character from the RS232 host computer it performs a power on reset. Everything is returned to the default state and an IEEE bus ABORT command is performed. An ABORT command pulses the IFC line for at least 100 micro seconds, then asserts the REN and ATN lines true.

This reset function is to get you out of trouble if the IEEE bus devices hang when , say, timeouts are disabled and a device crashes. The default reset character is @ the Ascii character decimal 64.

The RESET command can be used to specify which Ascii character is to be used instead of the default character @, so allowing for flexible data formats.

The Current Device.

A concept that effects many bus operations is the current device. The current device is extremely important since all data sent via the PSI BOX in data pass through and all data received from bus via the ENTER command is to and from the current IEEE device. In addition, many of the IEEE controls and functions access the current device when no device number is explicitly specified.

The PSI BOX front panel DIP switches allows you to specify the INITIAL value of the IEEE current device number, the default secondary address is none.

During operation, the current device, can be set in two ways. It may be set at any time when an Ascii number is the first one or two parameters after a BUS command.

To set device 5 no secondary address as the current device send the following sequence:-
<CR> BUS 5 <Carriage Return>

To set device 6 with a secondary address of 2 as the current device send the following sequence:-
<CR> BUS 6, 2 <Carriage Return>

The first two fields of IEEE control status string will be updated to reflect the information send.

Secondly, PSI BOX allows the current device number to be changed using the TALK, TAD, LAG, LISTEN and other addressed IEEE functions. With this method the current device is automatically set to the last device address specified.

<CR> BUS LISTEN 2 <Carriage Return>
Changes the current device to 2

<CR> BUS LISTEN 2+3 <Carriage Return>
Changes the current device to device 2 secondary address 3.

<CR> BUS TALK 2 LISTEN 3,4
Changes the current device to device 4.

<CR> REMOTE 2,3,4 <Carriage Return>
Changes the current device to device 4.

<CR> BUS SPOLL 3,4,5 <Carriage Return>
Changes the current device to device 4.

The golden rule is that the last device specified is now the current device.

Mode 7 BUS Command Summary.

Figure 7-3 Mode 7 IEEE Command Primitives

DCL	Device Clear.	GET	Group Execute Trigger
GTL	Go To Local.	LAG	Listen Address.
LLO	Local Lockout.	MLA	My Listen Address.
MTA	My Talk Address.	PPC	Parallel Poll Configure.
PPD	Parallel Poll Disable.	PPE	Parallel Poll Enable.
PPU	Parall Poll Unconfigure	SDC	Select Device Clear.
SPE	Serial Poll Enable.	SPD	Serial Poll Disable.
TAD	Talk Address.	TCT	Take Control.
UNL	Unlisten.	UNT	Untalk.
CMD	Send byte ATN true.	DATA	Send byte ATN false.
EOI	Send DATA with EOI true	LISTEN	Listen Address.
SEC	Secondary Address.	TALK	Talk Address.

Figure 7-4 Mode 7 IEEE Line Controls.

ATN	Asserts ATN when not sending data.
IFC	Pulse the IFC line true for 100 micro-sec
NO ATN	Leaves ATN false when not sending.
NO REN	Leaves REN false.
REN	Asserts REN line continuously true.

Figure 7-5 Mode 7 IEEE Command Functions.

```
ABORT       Resets the bus ,pulse IFC, assert REN, assert ATN.
CLEAR       Resets ALL devices. Performs DCL, device clear.
CLEAR 6,7   Resets devices 6 and 7. Performs SDC.
CONFIGURE   Configures device’s response to PPOLL.
DISABLE     Prevents some or all devices responding to PPOLL.
ECHO=       Instructs PSI BOX to echo RS232 data back to host.
END=        Sets the End of Sequence method for OUTPUTs from PSI.
ENTER       Ask for data from IEEE device.
EOS=        Specifies EOS string for IEEE data INPUTs to PSI BOX.
LOCAL       Enable front panel control. REN and ATN sent false.
LOCKOUT     Disable front panel controls. Performs LLO command.
MACRO       Defines or lists a user’s own macro command.
MESSAGES=   Enable or disable PSI BOX to host error messages.
NO TO       Disables time out.
OUTPUT      Sends data to current bus device.
PPOLL       Performs a complete Parallel Poll
READ        Reads one byte of data from addressed talker.
REMOTE      Asserts REN line continuously true, sets ATN false.
SHOW        Display & Explain Control Status String.
SPOLL       Serial Polls the current device.
SPOLL 6,7   Serial Polls devices 6 and 7.
STATUS      Sends the current PSI BOX parameter table to RS232.
TIMER       Wait for a given number of milli seconds
TO          Sets an i/o timeout value.
TRIGGER     Performs Group Execute Trigger.
UNCONFIGURE Prevents devices responding to Parallel Poll.
VERSION     Returns the ROM version number and date.
WAIT        Wait for a given number of milli seconds
WAIT SRQ    Waits till SRQ is asserted by device.
X           Execute stored macro a specified number of times.
@           Emergency RESET.
RESET       Specify the RESET character.
```

Mode 7 Memory Allocation.

The built in 32K buffer of the PSI BOX can be bypassed by setting the BUFFER ON front panel DIP switch low when the PSI BOX is powered on or reset.

Regardless of whether the buffer is enabled the PSI BOX allocates approximately 256 bytes of ram to hold the Mode 7 IEEE control status parameter table. In addition, another 256 bytes are always set aside for the RS232 input buffer, this allows the PSI BOX to a compile a complete BUS command in its memory, as it waits for the trailing <CR> before the command is interpreted.

When the buffer is enabled then one half of the PSI BOX ram, 16K, acts as the RS232 input buffer, the other half as the RS232 output buffer.

If the RS232 input buffer of the PSI BOX becomes full then it will be unable to receive more RS232 data until the IEEE device accepts data from the PSI BOX.

If the RS232 output buffer of the PSI BOX becomes full then the PSI BOX will be unable to receive any more IEEE data until the RS232 device accepts data from the PSI BOX.

The transfer of data between the PSI BOX and the computer's RS232 port is governed by either the XON/XOFF handshake or by the DTR/DSR etc lines, as selected by the user via the XON XOFF front panel DIP switch.

CHAPTER 8
THE IEEE BUS COMMAND LANGUAGE CONTROLS & FUNCTIONS

Introduction.

This chapter describes the PSI BOX Mode 7 command language. First there is a discusion of the types of parameter that the various commands require. Then there follows complete alphabetical list of all the Mode 7 commands giving a precise description of the command syntax, function and usage.

Valid Parameters.

With the Mode 7 language covering such a wide range of functions a variety of parameters is required for the different IEEE command keywords. Some IEEE commands require no parameters, some commands require exactly one parameter, others may optionally have none, one, two or more parameters.

The separator between commands and successive parameters is either a space or a comma. The parameters fall into the following types, note the address list <add list> is of particular interest.

<none> No parameters are required. Some commands never have any parameters such as ABORT. Others such as CLEAR or SPOLL perform differently with and

without parameters.
ABORT Terminate bus activity.
CLEAR Reset all devices.
SPOLL Serially Poll current device.

<add> A valid primary IEEE address. Thus it is in the range 00 - 30 and 128-158. Often devices only require a primary address.
CLEAR 2 Reset IEEE device 2.

<sa> A valid secondary address. Thus it is in the range 00-30 and 128-158.
SEC 4 Send secondary address 4 to listener.

<add+sa> A valid primary address with a valid secondary address.
CLEAR 2+4 Reset IEEE device 2 with sa of 4.

<add list> A list of none, one or more primary addresses with optional secondary addresses. Here are several examples of valid address lists. The action that is taken, eg Serial Poll, is performed on the devices in the order that they are listed.
CLEAR
CLEAR 2
CLEAR 2+4,6,3
SPOLL 2,4,1,0,7,8
LISTEN 4,3+2,9
Please note that whenever an address list is used, then <none>, <add>, <add+sa> are all valid parameters. The address list is a powerful and handy feature!

<bits> A single digit, either 7 or 8, that defines the number of significant bits to match in the EOS and END commands.
EOS=13,7 Input stops on Carriage Return received, 7 bit compare. ie 13 or 141 valid match.

EOS=13,8 Input stops on Carriage Return received, 8 bit compare. ie 13 is only valid match.

<byte> A decimal number in the range 0-255. Used mainly in a data or a command byte.

DATA 13 Carriage return char sent to IEEE bus.
EOI 10 Line feed with EOI true sent to bus.
CMD 63 The UNLISTEN command sent to bus.

<bytes> One or more decimal numbers in the range 0-255.

DATA 79,75,13 OK Carriage return sent to bus.
CMD 63,33,4 The UNLISTEN, LISTEN 1 and SDC commands are sent to bus.

<number> Decimal number in the range 00000-65535. Used to specify how many bytes of data to be read using the ENTER command.

<sppp> Parallel Poll data. The range is 00-15 and is used to specify how and on which data line the device responds to the Parallel Poll command, POLL.
CONFIGURE 4+7 Tell device 4 to use data line 8 to reply to POLL's. 7=sppp here.

<string> A series of ascii characters, used inside quotation marks or other delimiters, to specify the literal data bytes to send. Valid delimiter is any non space character.
OUTPUT !HELLO! here the string HELLO is sent to the current device.

<tt> Specify timeout period, range is 0-15. 0 means never timeout, 1 is 1 millisec, 15 is 50 sec wait

TO 7 Timeout of 1 second.

ABORT. Terminate Activity, Reset Devices.

Purpose. The PSI BOX pulses the IFC, interface clear line, asserts the REN, remote enable line, and asserts the ATN, attention line, true. This causes all IEEE bus activity to stop immediately, all devices come under REMOTE, ie PSI BOX, control. All devices are untalked and unlistened. The PSI BOX becomes the ACTIVE controller.

The ABORT command is usually issued when you power your PSI BOX up or, at a subsequent time, when you want to initialise the IEEE bus devices to their power on state.

This command will prevent any data transfer between two devices continuing. All bus devices return to their device dependent power on state and become ready to receive address or other IEEE commands from the PSI BOX. Byte 9 in the IEEE control status string contains 1 when the ATN line is asserted true, controller active and 0 when the ATN line is false, PSI BOX is not the active controller. Byte 11 in the IEEE control status string contains 1 when the REN line is asserted true, devices under PSI BOX control.

At the end of each subsequent data transfer sequence ATN is reasserted.

Format. ABORT

Params. None.

Returns. IEEE control status string byte 9 = 1
IEEE control status string byte 11 = 1

Example.

```
BUS ABORT              :REM SEND COMMAND
BUS STATUS             :REM READ ATN
ATN at position 9      :REM ATN = 1
REN at position 11     :REM REN = 1
```

Bus Activity.

pulse IFC for approx 100 micro seconds.
assert REN.
assert ATN.

ATN. Assert The Atn Line.

Purpose. To cause the PSI BOX to assert the ATN, attention, line when it is not transferring data. The PSI BOX becomes the ACTIVE controller.

This command will prevent any data transfer between two devices continuing. All bus devices will become ready to receive address or other IEEE commands from the PSI BOX. Byte 9 in the IEEE control status string contains 1 when the ATN line is asserted true, controller active and 0 when the ATN line is false, controller active.

At the end of each data transfer sequence ATN is reasserted.

Format. ATN

Params. None.

Returns. IEEE control status string byte 9 = 1
If the PSI BOX was already the active controller then this command has no effect.

Example.

```
BUS STATUS               :REM READ STRING
ATN at position 9        :REM ATN = 0
BUS ATN                  :REM SEND COMMAND
BUS STATUS               :REM READ ATN
ATN at position 9        :REM ATN = 1
```

Bus Activity.
assert ATN

CLEAR. Resets All or Specified Devices.

Purpose. To send all or selected devices to their pre determined power on state.

The CLEAR command is used to send bus devices to a device dependent state. If no device addresses are given then the PSI BOX uses the universal command DCL to send all bus devices to their power on state. If one or more device addresses are specified then those devices are commanded to Listen and then told to perform a selective device clear.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command DCL, or a series of listen addresses followed by the SDC command, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The CLEAR command is a very flexible way to reset devices to a known state after they have become confused by an error or after they have been set to an unknown state by a previous user.

Format. CLEAR

Params. None, one or several.

Returns. None.

Example.

```
BUS CLEAR         Resets ALL devices
BUS CLEAR 2       Resets device 2
BUS CLEAR 2,3,4   Resets devices 2 3 and 4.
```

Bus Activity.

assert ATN if not already so
DCL

or: assert ATN if not already so
MTA of PSI BOX
UNL all stop listening
LAG 2 first addresses specified
....
LAG 4 last address specified
SDC addressed device to perform a clear

CMD. Send A Byte With Attention True.

Purpose. To send a command byte to the IEEE bus.

CMD is available for the user to send any bytes he wishes to the IEEE bus, with the ATN line asserted. This means that the byte is interpreted by the devices as a command rather than as a data byte. This command is the complement to DATA which sends a byte with ATN false.

This routine can be used to send any of the standard IEEE commands. The user must ensure that he has got the sequence of commands and secondary commands correct.

The PSI BOX asserts ATN, if not already true, and becomes the active controller

Use NO ATN to drop ATN false again when you have finished sending your stream of commands.

The bit 7 of the byte values sent is a don’t care bit, but are sent as entered.

Format. CMD byte

Params. One or some.

Returns. None, ATN is left true.

Example.

```
BUS CMD 32 33 63        :REM ISSUE COMMAND
```

Bus Activity.

```
assert ATN if not already so
LISTEN 0 (=command 32)
LISTEN 1 (=command 33)
UNLISTEN (=command 63)
```

CMD Equivalents Of IEEE Bus Commands.

1 or	129	GTL	GO TO LOCAL.
4 or	132	SDC	SELECTIVE DEVICE CLEAR.
5 or	133	PPC	PARALLEL POLL CLEAR.
8 or	136	GET	GROUP EXECUTE CLEAR.
9 or	137	TCT	TAKE CONTROL.

17or	145	LLO	LOCAL LOCKOUT.
20or	148	DCL	DEVICE CLEAR.
21or	149	PPU	PARALLEL POLL UNCONFIG
24or	152	SPE	SERIAL POLL ENABLE.
25or	153	SPD	SERIAL POLL DISABLE.
32or	160	LAG 0	LISTEN ADDRESS 0
..	..	..	...
62or	190	LAG 30	LISTEN ADDRESS 30
63or	191	UNL	UNLISTEN
64or	192	TAD 0	TALK ADDRESS 0
..	..	..	...
94or	222	TAD 30	TALK ADDRESS 30
95or	223	UNT	UNTALK
96or	224	SCG 0	SECONDARY COMMAND 0
..	..	..	...
127or	254	SCG 30	SECONDARY COMMAND 30

CONFIGURE. Specifies Devices Parall Poll Response.

Purpose. This command is used to instruct one or more devices as to how they should respond to a parallel poll.

Devices respond to a parallel poll command by asserting a particular data line high or low, this informs the PSI BOX whether the device requires the services of the controller. A device cannot respond to a parallel poll command unless it has been configured by either the CONFIGURE command or by the PPC PPE sequence. This CONFIGURE command is a straight forward means of assigning a data line, for response to the parallel poll command, to the specified device.

The PSI BOX addresses the specified device performs the PPC PPE sequence as explained in the PPE command explanation.

As well as the device address the CONFIGURE command requires a number in the range 0-15. This parameter has the bit value %0000sppp. Here s is the sense of response bit and ppp is the data line assigned to the device.

The sense information is in bit 3 (decimal 8).
If bit 3=0 then assert line when the device’s individual status bit, IST, is zero.
If bit 3=1 then assert line when the device’s IST is one.

The data line is assigned using bits 0-2.
If bits= 000, decimal 0, then data line 0
If bits= 001, decimal 1, then data line 1
If bits= 010, decimal 2, then data line 2
If bits= 011, decimal 3, then data line 3
If bits= 100, decimal 4, then data line 4
If bits= 101, decimal 5, then data line 5
If bits= 110, decimal 6, then data line 6
If bits= 111, decimal 7, then data line 7

Format. CONFIGURE ad+sppp

Params. Two or more

Returns. None.

Example.

```
BUS CONFIGURE 2+5, 3+6
device 2 is asserts data line 5 true during PPOLL
device 3 is asserts data line 6 true during PPOLL
if they require service.
```

Bus Activity.

assert ATN if not already so
UNT
UNL
LAG of specified device
PPC
PPE includes line assignment and sense bit data
UNL
repeats with LAG of next device as necessary

DATA. Send A Byte With Attention False.

Purpose. To send a data byte to the IEEE bus.

DATA is available for the user to send any bytes he wishes to the IEEE bus, with the ATN line false. This means that the byte is interpreted by the devices as a data byte rather than as an IEEE command. This command is the complement to CMD which sends a byte with ATN true.
The EOI line is false whilst the data byte is being sent.

The PSI BOX drops ATN false, if not already so, at the end of the data sequence ATN is left false until further instructions makes the PSI BOX become the active controller.

Use ATN to set ATN true again when you have finished sending your stream of commands.
The all 8 data bits are sent as entered.

Format. DATA byte

Params. One or some.

Returns. None, ATN is left false.

Example.

```
BUS LISTEN 4 DATA 68 65 72 72 79 13 UNL
This will cause an IEEE printer device 4 to print the text
"HELLO".
```

Bus Activity.

assert ATN if not already so
LISTEN 4
DAB 68 (H)
DAB 65 (E)
DAB 72 (L)
DAB 72 (L)
DAB 79 (0)
DAB 13 (Carriage Return)
UNLISTEN

DCL. Sends The Device Clear Command.

Purpose. The universal command Device Clear is sent to the IEEE bus.

The DCL command is used to send all bus devices to a device dependent state.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command DCL, hex 14h decimal 20, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The DCL command is used automatically in the CLEAR routine but it can be called separately to perform your own device clear routine. The DCL command can also be sent by using CMD 20 or CMD 148

Format. DCL

Params. None.

Returns. None.

Example.

```
BUS DCL
```

Bus Activity.

assert ATN if not already so
DCL

DISABLE. Prevents Devices Responding Parallel Poll.

Purpose. To prevent all or selected devices responding to the parallel poll command.

The DISABLE command is a very flexible way to prevent one or more devices, which has been previously CONFIGUREd, from responding to parallel polls.

If no device address is specified then the PSI BOX uses the universal PPU command to prevent all bus devices responding to the parallel poll. All the devices will need to be reCONFIGUREd if they are to take part in future parallel polls. If one or more device addresses are given then those devices are commanded to Listen and then told to via the PPD command not to respond to the parallel poll.

Format. DISABLE

Params. None, one or more.

Returns. None.

Example.

```
BUS DISABLE             all devices now unable to respond
BUS DISABLE 2,3,4       device 2,3 & 4 unable to respond
```

Bus Activity.

assert ATN if not already so
PPU all devices cannot respond
or assert ATN if not already so
UNT
UNL
LAG listen 1st device in sequence.
PPC
PPD
UNL repeat sequence with next device if any

END=. Set The End Of Output Method.

Purpose. Sets or changes the End Of Sequence terminator for OUTPUTS from the PSI BOX.

This command is used to inform the PSI BOX of when to set the EOI line true when writing data to the IEEE device.

The PSI BOX can be instructed never to set EOI true, or EOI can be set true when a particular character, in the range 00-255, is the last byte to be sent from the PSI BOX. The PSI BOX can be instructed to compare all 8 bits of the character written or just the 7 least significant bits.

Alternatively, using EOS=ON the EOI signal can be set when the last byte in the current string is output, this is the default condition.

Format.

a) END= ;EOI never sent true during an IEEE bus write. Equivalent to END=OFF.

b) END= character

;EOI sent true only when the PSI BOX sends the character specified as the last byte in the output sequence.
An 8 bit comparison is made.

c) END= character,8

;EOI sent true only when the PSI BOX sends the character specified as the last byte in the output sequence.
An 8 bit comparison is made.

d) END= character,7

;EOI sent true only when the PSI BOX sends the character specified as the last byte in the output sequence.
Only the bottom 7 bits of the data byte output by the PSI BOX is compared to the END character.

Params. None, One or Two.

Returns. None.

Example.

BUS END=

When no parameter is given or when **END=OFF** the **PSI BOX** never sets the **EOI** line true when writing data to the bus. This is of use when sending a large data stream to the bus in several chunks, so preventing the **IEEE** device from thinking that the output has finished early.

BUS END=13,8

The **EOI** line is set true only when the **PSI BOX** sends the Carriage Return character, 13, as the last byte in the current output sequence.

BUS END=10,7

The **EOI** line is set true only when the **PSI BOX** sends the Line Feed character 10, or character 138 since a 7 bit comparison is performed, as the last byte in the current output sequence.

END=OFF. Disable EOI During A Bus Write.

Purpose. Sets or changes the End Of Sequence terminator so that EOI is never sent true during the bus write routine.

This command is used to inform the PSI BOX of when to set the EOI line true when writing data to the IEEE device.

This command prevents the EOI signal being set true when the last byte in the current string is output.

The PSI BOX is instructed never to set EOI true.

This command is used mainly when a large stream of data is being sent to the IEEE bus in several smaller chunks. it prevents EOI being set early and so prevents the listener terminating the bus handshake early.

Alternatively, using EOS=ON and EOS= the EOI signal can be set when the last byte in the current string is output, or EOI can be set true when a particular character, in the range 00-255, is the last byte to be sent from the PSI BOX. The PSI BOX can be instructed to compare all 8 bits of the character written or just the 7 least significant bits.

Format.
a) END=OFF
b) END= ;EOI never sent true during an IEEE bus write.

Params. None.

Returns. None.

Example.
```
BUS END=
BUS END=OFF
```
When no parameter is given or when END=OFF the PSI BOX NEVER sets the EOI line true when writing data to the bus. This is of use when sending a large data stream to the bus in several chunks, so preventing the IEEE device from thinking that the output has finished early.

END=ON. Set The End Of Output Method.

Purpose. Sets or changes the End Of Sequence terminator so that the EOI line is asserted true with the last byte in the current string.

This command is used to inform the PSI BOX of when to set the EOI line true when writing data to the IEEE device.

The EOI signal is set true when the last byte in the current string is output.

Alternatively, the PSI BOX can be instructed never to set EOI true, or EOI can be set true when a particular character, in the range 00-255, is the last byte to be sent from the PSI BOX. The PSI BOX can be instructed to compare all 8 bits of the character written or just the 7 least significant bits.

Format.
a) END=ON
;EOI sent true with the last byte of each IEEE bus write.

Params. None.

Returns. None.

Example.

```
BUS END=ON
When no parameter is given the PSI BOX returns to its default
configuration. The PSI BOX sets the EOI true when sending the
last data byte in the current sequence.
```

EOI. Send A Data Byte With Eoi True.

Purpose. To send a data byte to the IEEE bus with the EOI line true.

The EOI instruction is used to terminate a DATA sequence by sending the data byte to the listeners with EOI true. The listeners sending EOI true receive the END message and know that the byte specified is the last in the current sequence that the PSI BOX is sending.

The PSI BOX drops ATN false, if not already so, at the end of the EOI with data sequence ATN is left false until further instructions makes the PSI BOX become the active controller.

Use ATN to set ATN true again when you have finished sending your stream of commands.
The all 8 data bits are sent as entered.

Format. EOI byte

Params. One.

Returns. None, ATN is left false.

Example.

```
BUS LAG 4 DATA 68 65 72 72 79 EOI 13 UNL
An IEEE printer device 4 will print the text "HELLO".
```

Bus Activity.

assert ATN if not already so
LISTEN 4
DAB 68 (H)
DAB 65 (E)
DAB 72 (L)
DAB 72 (L)
DAB 79 (0)
EOI asserted TRUE
DAB 13 (Carriage Return) drop EOI false.
UNLISTEN

EOS=. Set End Of Sequence Character For Inputs.

Purpose. Sets or changes the End Of Sequence terminator for INPUTS from the IEEE bus.

This command is used to inform the PSI BOX of how the IEEE device signals that it has sent all its data to the PSI BOX.

The EOS terminator can be just the EOI signal set true by the IEEE device.

Or it can be any character in the range 00-255. The EOS byte can be set to compare all 8 bits of the character read or just the 7 least significant bits.

Of course, regardless of the value of the EOS terminator byte if a talker sets EOI true with a byte sent to the PSI BOX then the PSI BOX knows that the sequence has ended.

Format.

a) EOS=

;Bus inputs terminate only when EOI is set true by the IEEE device.
This is the default setting.

b) EOS= character

;Bus inputs terminate only when the IEEE device sends the character specified.
An 8 bit comparison is made.

c) EOS= character,8

;Bus inputs terminate only when the IEEE device sends the character specified.
An 8 bit comparison is made.

d) EOS= character,7

;Bus inputs terminate only when the IEEE device sends the character specified.
Only the bottom 7 bits of the data read from the IEEE bus is compared to the EOS character.

Params. None, One or Two.

Returns. None.

Example.

BUS EOS=

When no parameter is given the PSI BOX returns to its default configuration. Inputs from the IEEE bus device only terminate when the device sets the EOI true when sending its last data byte.

BUS EOS=13,8

All inputs from the IEEE bus device terminate when the device sends the Carriage Return character, 13.

BUS EOS=10,7

All inputs from the IEEE bus device terminate when the device sends the Line Feed character 10, or character 138 since a 7 bit comparison is performed.

GET. Sends Group Execute Trigger.

Purpose. The addressed command Group Execute Trigger is sent to the IEEE bus.

The GET command is used to initiate a pre-programmed action in the responding devices.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command GET, hex 08h decimal 8, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The GET command is used automatically in the TRIGGER routine but it can be called separately to initiate your own device trigger. The GET command can also be sent by using CMD 8 or CMD 136

Format. GET

Params. None.

Returns. None.

Example.

```
BUS GET
```

Bus Activity.

assert ATN if not already so
GET

GTL. Sends The Go To Local Command.

Purpose. The addressed command Go To Local is sent to the IEEE bus.

The GTL command is used to tell all the currently listening devices to respond to their front panel, local, controls rather than to the PSI BOX.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command GTL, hex 01h decimal 1, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The GTL command is used automatically in the LOCAL routine but it can be called separately to initiate your own return to local routines. The GTL command can also be sent by using CMD 1 or CMD 129

Format. GTL

Params. None.

Returns. None.

Example.

```
BUS GTL
```

Bus Activity.

assert ATN if not already so
GTL

IFC. Pulse The IFC Line.

Purpose. To cause the PSI BOX to momentarily assert the IFC, InterFace Clear, line.

This command will prevent any data transfer between two device continuing. All devices are untalked and unlistened so causing all IEEE bus activity to stop immediately.

Good to use IFC or ABORT shortly after power on as the first IEEE bus command so that the IEEE bus devices are put into a known state.

The IFC command is usually issued when you power your PSI BOX up or, at a subsequent time, when you want to initialise the IEEE bus devices to their power on state.

Format. IFC

Params. None.

Returns. None.

Example.

```
BUS IFC          :REM RESET BUS
```

Bus Activity.

pulse IFC true for approx 100 micro seconds.

LAG. Sends The Listen Address.

Purpose. The listen address is sent to the IEEE bus.

The LAG command is used to send a listen address in the range 0 to 30 to the bus devices.

The PSI BOX asserts ATN and becomes the active controller, then the number entered by the user is ORed with hex 020h, 32 decimal, and is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The LAG is used to instruct one or more bus devices that they should listen to the addressed commands or data that follows the listen command. The LAG command can also be sent by using CMD 32-62 or CMD 160-190 or by using LISTEN 0-30 or 128-158

Format.

LAG 1	Tell device 1 to listen.
LAG 1,3,4	Devices 1, 3 and 4 to listen.
LAG 1+2,3,4	Device 1 Secondary address 2, and devices 3 and 4 told to listen.

Params. One or more parameters each in the range 0-30 or 128 to 158.

An optional secondary address can be specified for each primary address by using +sa after the primary address. The sa is also in the range 0-30.

Returns. None.

Example.

```
BUS LAG 1,3,4
Tell devices 1, 3 and 4 all to listen.
```

Bus Activity.

assert ATN if not already so
LAG 1
LAG 3
LAG 4

LISTEN. Sends The Listen Address.

Purpose. The listen address is sent to the IEEE bus.

The LISTEN command is used to send a listen address in the range 0 to 30 to the bus devices.

The PSI BOX asserts ATN and becomes the active controller, then the number entered by the user is ORed with hex 020h, 32 decimal, and is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The LISTEN is used to instruct one or more bus devices that they should listen to the addressed commands or data that follows the listen command. The LISTEN command can also be sent by using CMD 32-62 or CMD 160-190 or by using LAG 0-30 or 128-158

Format.

LISTEN 1	Tell device 1 to listen.
LISTEN 1,3,4	Devices 1, 3 and 4 to listen.
LISTEN 1+2,3,4	Device 1 Secondary address 2, and devices 3 and 4 told to listen.

Params. One or more parameters each in the range 0-30 or 128 to 158.
An optional secondary address can be specified for each primary address by using +sa after the primary address. The sa is also in the range 0-30.

Returns. None.

Example.

```
BUS LISTEN 1,3,4
Tell devices 1, 3 and 4 all to listen.
```

Bus Activity.

assert ATN if not already so
LAG 1
LAG 3
LAG 4

LLO. Sends The Local Lockout Command.

Purpose. The universal command Local Lockout is sent to the IEEE bus.

The LLO command is used to tell all the bus devices to ignore their front panel, local, controls and respond only to the PSI BOX.

The LLO command is used to disable a device from responding to its own front panel controls. This is necessary when you want the instrument to be totally under the control of the PSI BOX. It prevents accidental pushing of the devices buttons interfering with its operation.

The device will return to local control when it receives the GTL command, or when the IEEE REN line goes false.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command LLO, hex 11h decimal 17, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The LLO command is used automatically in the LOCAL LOCKOUT routine but it can be called separately to initiate your own return to local routines. The LLO command can also be sent by using CMD 17 or CMD 145

Format. LLO

Params. None.

Returns. None.

Example.

```
BUS LLO
```

Bus Activity.

assert ATN if not already so
LLO

LOCAL. Set Some Or All Devices To Local Control.

Purpose. Commands some or all devices to return to front panel control.

The LOCAL command is a very flexible way to command one or more devices to respond to their front panel, local, controls rather than to the PSI BOX.

If no device address is specified then the PSI BOX drops both REN and ATN lines false, ALL IEEE bus devices are now under front panel control. If one or more devices addresses are given then those devices are commanded to Listen and then told via the GTL command to respond to their front panel controls rather than to the PSI BOX controller.

Format. LOCAL

Params. None, one or many.

Returns. None.

Example.

```
BUS LOCAL   all devices control by front panel
BUS LOCAL 2,3,4   devices 2,3 & 4 to front panel control
```

Bus Activity.

drop REN
drop ATN
or assert ATN if not already so
MTA of PSI BOX
UNL all stop listening
LAG 2 first addresses specified
....
LAG 4 last address specified
GTL addressed device to go to local control

LOCKOUT. Sends The Local Lockout Command.
or LOCAL LOCKOUT.

Purpose. To disable the front panel controls on all IEEE bus devices.

The LOCKOUT command is used to tell all the bus devices to ignore their front panel, local, controls and respond only to the PSI BOX.

The LOCKOUT command or it alternative form LOCAL LOCKOUT are used to prevent a device from responding to its own front panel controls. This is necessary when you want the instrument to be totally under the control of the PSI BOX. It prevents accidental pushing of the devices buttons interfering with its operation.

The device will return to local control when it receives the GTL command via a LOCAL command, or when the IEEE REN line goes false.

The PSI BOX asserts ATN and becomes the active controller and sends , then the IEEE command LLO, hex 11h decimal 17. After the command has been sent the ATN remains true, PSI BOX is the active controller.

Format. LOCKOUT or LOCAL LOCKOUT

Params. None.

Returns. None.

Example.

```
BUS LOCKOUT
```

Bus Activity.

assert ATN if not already so
LLO

MACRO. Defines Or Lists Macro.

Purpose. Allows the user to define a new macro or list the current macro.

A MACRO is a series of PSI BOX commands grouped together, that are executed by issuing the X keyword. Whenever the X command is found the whole series of commands defined in the macro is executed. Thus the macro facility allows the user to define his own pseudo commands, that can be simply executed.

This is a powerful feature, especially when used with the repeating form of X and when a WAIT command has been given in the macro. This allows multiple, timed, bus sequences to be performed by simply issuing one character. The overhead on the RS232 line is cut dramatically and the PSI BOX can be used in an automated fashion whilst the controlling terminal is busy performing other jobs.

The MACRO command followed by only spaces and a carriage return will cause the PSI BOX to send the current macro definition back to the user. IE it lists the macro.

The MACRO command followed by any parameters and data will cause that sequence to be stored as the new current macro overwriting and discarding any previous macro.

ON power up the macro is set to perform a BUS STATUS command.

ON RESET, using the PSI BOX reset character any stored macro is kept intact. Ensure that it has not been corrupted by listing it.

If the X command is placed within a macro, it and its parameter, are skipped over when the macro executes.

If the MACRO command contains the MACRO keyword and a new macro definition then an ERROR is flagged when the macro is executed. A MACRO command may end with the MACRO keyword. This causes the MACRO to list itself whenever it executes.

Format. MACRO
MACRO <ANY PSI BOX COMMAND >

Params. ANY PSI BOX COMMANDS IN ANY ORDER

Returns. DEPENDS ON USERS COMMAND SPECIFIED IN MACRO.

Example.

```
BUS MACRO               :REM lists the current macro
BUS MACRO OUTPUT "SET?" ENTER
      :REM current macro is now the sequence
                                   OUTPUT "SET?" ENTER

BUS MACRO ENTER 1024 WAIT 5000
      :REM macro will enter 1K data every 5 sec from
                                               device.
```

Bus Activity.
any

MESSAGES=OFF. Disable PSI BOX Messages.

Purpose. Prevents the PSI BOX responding with the ALL DONE OK or ERROR IN COMMAND message upon completion of a BUS command.

The default is that messages are ON. Messages are usually switched off when a fully debugged program is used to acquire large amounts of data via the PSI BOX, without having to process the extra data associated with the PSI BOX messages. Note: NO embedded spaces between S in messages, the = sign and the O in off.
This is the compliment of the MESSAGES=ON command.

Format. MESSAGES=OFF

Params. None.

Returns. None.

Example.

```
BUS MESSAGES=OFF
```

Bus Activity.
None.

MESSAGES=ON. Enable PSI BOX Messages.

Purpose. Allows the PSI BOX to respond with the ALL DONE OK or ERROR IN COMMAND message upon completion of a BUS command.

The default is that messages are ON. Messages are usually switched off when a fully debugged program is used to acquire large amounts of data via the PSI BOX without having to process the extra data associated with the PSI BOX messages. Note: NO embedded spaces between S in messages, the = sign and the O in on.
This is the compliment of the MESSAGES=OFF command.

Format. MESSAGES=ON

Params. None.

Returns. None.

Example.

```
BUS MESSAGES=ON
```

Bus Activity.

None.

MLA. Sends My Listen Address.

Purpose. The Listen Address of the PSI BOX device is sent to the bus.

The MLA command is used to tell all the bus devices that the PSI BOX controller is a listener in the next data transfer sequence.

Any data sent by the talker will be received by the PSI BOX. Any number of listeners can be listening simultaneously to one talker.

The PSI BOX asserts ATN and becomes the active controller, then the listen address of the PSI BOX is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The MLA command is usually used when to initiate your own data exchange routines.

Format. MLA

Params. None.

Returns. None.

Example.

```
BUS MLA
```

Bus Activity.

assert ATN if not already so
LAG 0 the default listen address of the PSI BOX

MLA=. Sets Or Changes PSI BOX Listen Address.

Purpose. To set or change the Listen Address of the PSI BOX device.

The MLA= command is used to change the Listen address of the PSI BOX from the default value of 0 to an other address more convenient to the user.

Bytes 50-51 in the IEEE control status string contains the current MLA of the PSI BOX.

Format. MLA=

Params. One address in the range 00-30

Returns. The My Listen Address of the PSI BOX is stored in control status at bytes 50-51.

Example.

```
BUS MLA= 2  Sets the listen address of PSI BOX to 2
```

Bus Activity.

None.

MSA. Sends My Secondary Address.

Purpose. The Secondary Address of the PSI BOX device is sent to the bus.

The MSA command is used after a MLA or a MTA to tell all the bus devices that the PSI BOX controller is listening or talking on the bus using a secondary address. Secondary addresses are used to tell the device that a special feature is being accessed by the user. You will have to supply your own special features for the PSI BOX secondary addresses.

The MSA should only be sent immediately after a MTA or MLA command. PSI BOX asserts ATN and becomes the active controller, then the secondary address of the PSI BOX is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The MSA command is usually as part of your own data exchange routines.

Format. MSA

Params. None.

Returns. None.

Example.

```
BUS MSA
```

Bus Activity.

assert ATN if not already so
SEC 0 if the PSI BOX sec add has been set to 0
Default is none.

MSA=. Sets Or Changes PSIBOX Secondary Address.

Purpose. To set or change the Secondary Address of the PSI BOX device.

The MSA= command is used to change the secondary address of the PSI BOX from the default value of none to an other address more convenient to the user.

Bytes 53-54 in the IEEE control status string contains the current MSA of the PSI BOX.

Format. MSA=

Params. One address in the range 00-30 or none.

Returns. The My Sec Address of the PSI BOX is stored in control status at bytes 53-54.

Example.

```
BUS MSA= 2  Sets the Sec address of PSI BOX to 2
BUS MSA=    Resets the Sec add of PSI BOX to None.
```

Bus Activity.

None.

MTA. Sends My Talk Address.

Purpose. The Talk Address of the PSI BOX device is sent to the bus.

The MTA command is used to tell all the bus devices that the PSI BOX controller is the talker, the source of the data, in the next data transfer sequence.

The listener on the bus will accept data from the PSI BOX. Since only one talker can be active on the bus at a time this command automatically UNTalks any other talkers.

The PSI BOX asserts ATN and becomes the active controller, then the Talk address of the PSI BOX is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.
The MTA command is usually used when to initiate your own data exchange routines.

Format. MTA

Params. None.

Returns. None.

Example.

```
BUS MTA
```

Bus Activity.

assert ATN if not already so
TAD 0 the default talk address of the PSI BOX

MLA=. Sets Or Changes PSI BOX Listen Address.

Purpose. To set or change the Talk Address of the PSI BOX device.

The MTA= command is used to change the Talk address of the PSI BOX from the default value of 0 to an other address more convenient to the user.

Bytes 56-57 in the IEEE control status string contains the current MTA of the PSI BOX.

Format. MTA=

Params. One address in the range 00-30

Returns. The My Talk Address of the PSI BOX is stored in control status at bytes 56-57.

Example.

```
BUS MLA= 2  Sets the Talk address of PSI BOX to 2
```

Bus Activity.

None.

NO ATN. Release The Atn Line.

Purpose. To cause the PSI BOX to release the ATN, attention, line when it is not transferring data. The PSI BOX stops being the ACTIVE controller and returns to the controller idle state.
Byte 9 in the IEEE control status string contains 1 when the ATN line is asserted true, controller active and 0 when the ATN line is false, controller idle.

The PSI BOX will return to the idle state in between each data transfer sequence.

Format. NO ATN

Params. None.

Returns. IEEE control status string byte 9 = 0
If the PSI BOX was already in the controller idle state then this command has no effect.

Example.

```
BUS NO ATN         :REM ISSUE COMMAND
BUS STATUS         :REM READ STRING
ATN at position 9 :REM ATN = 0
```

Bus Activity.
release ATN.

NO REN. Release The Ren Line.

Purpose. To cause the PSI BOX to release the REN, remote enable, line. The PSI BOX is returning the IEEE bus devices to local control and they cannot return to PSI BOX control until the PSI BOX reasserts REN again.
Byte 11 in the IEEE control status string contains 1 when the REN line is asserted true, PSI BOX remotely controlling bus devices, and 0 when the REN line is false, IEEE devices under local control.

Whenever a read or write is performed or whenever any bus command is sent the PSI BOX reasserts the REN line.

Format. NO REN

Params. None.

Returns. IEEE control status string byte 11 = 0
If the REN line was already false then this command has no effect.

Example.

```
BUS NO REN               :REM ISSUE COMMAND
BUS STATUS               :REM READ STRING
REN at position 11       :REM REN = 0
```

Bus Activity.

release REN.

NO TO. Disable The Timeouts.

Purpose. To make the PSI BOX wait indefinitely in a bus handshake routine till the responding device is ready for the data transfer.

When the PSI BOX is exchanging data with a bus device it may need to wait a long time for the data to become present, or for the device to make data available. The NO TO command can be used to make the PSI BOX wait indefinitely for the device to be ready to handshake data. By contrast the TO command can be used to set the maximum time that the PSI BOX waits for the device to become ready.

Whilst most IEEE devices are fast enough to quickly respond to your PSI BOX, IEEE printers and most especially IEEE plotters are quite slow devices. When the printer or plotter buffer fills up the PSI BOX must wait for it to print the current buffer contents and so make room for more input. During this wait time the PSI BOX will timeout the IEEE printer or plotter. To prevent this timeout occuring and so causing the program to think that the printer has gone dead, you should specify the NO TO, no timeout, option.
Now the PSI BOX will always wait until the plotter or printer can cope with the data it receives.

Two Notes Of Caution!.
a) If the printer or plotter hangs up due to some reason or other the PSI BOX will not timeout but will continue to wait.

b) Be sure that you have properly set the EOS byte for terminating inputs. If the PSI BOX does not know how to recognise the last byte in the input sequence it may go on to ask for more and so wait indefinitely if the Talker has no more to send. See the EOS command.

Bytes 57 and 58 in the IEEE control status string contains the current timeout value. A value of 0 indicates that the timeouts have been disabled.

The timeouts step up in 1,2,5,10 increments from 1

millisecond to 50 seconds. Therefore each timeout increment represents an increase in PSI BOX wait time of approximately a factor of 2.
The timeouts can also be disabled using NOTO or by TO 0.

Format. NO TO

Params. None.

Returns. The IEEE control string holds the current timeout value in position 57-58.
NO TO gives a TO value of 0.

Example.

```
BUS NO TO              :REM ISSUE COMMAND
BUS STATUS             :REM READ STRING
TMO at position 57     :REM TMO = 0.
```

OUTPUT. Send A String Of Data To A Bus Device.

Purpose. To send a string of data to the current bus device using a complete handshake sequence.

The current IEEE device is addressed to listen and then then the data enclosed with the delimiters is sent to the bus. Finally the UNL command is sent to complete the sequence. The EOI line is set true concurrent with the last data byte according to how the END parameter is set. The default is sent only with <CR> character.

The OUTPUT command is a simple method of sending text to the current device.

Format. OUTPUT "HELLO"

Params. Delimiter string delimiter.

Returns. None, ATN is left false.

Example.

```
BUS OUTPUT !HELLO!
This will send the text HELLO to the current bus device.
```

Bus Activity.

assert ATN if not already so
LISTEN current device address
DAB 68 (H)
DAB 65 (E)
DAB 72 (L)
DAB 72 (L)
DAB 79 (0)
UNLISTEN

PPC. Sends Parallel Poll Configure.

Purpose. The addressed command Parallel Poll Configure is sent to the IEEE bus.

This command is used as part of a sequence to allow the addressed listener to be assigned a data line for response to the parallel poll command.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command PPC, hex 05h decimal 5, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The PPC command must be followed by a PPE command or a PPD command. The PPE command is used to tell the addressed listener which particular data line it is to use when responding to a parallel poll, the PPE command also tells the addressed device whether to assert the line when it requires service or to assert the line when it does not require service.

The PPD command is used to prevent the addressed device from responding to parallel polls. The previously assigned data lines is remembered.

The PPC command has been included for completeness only, it is advised that the CONFIGURE or DISABLE commands be used to configure a device for Parallel Poll since both these commands automatically send the PPC command.
The PPC command can also be sent by using CMD 5 or CMD 133

Format. PPC

Params. None.

Returns. None.

Example.

```
BUS PPC
recommended sequence (if not using CONFIGURE)
BUS UNT UNL LISTEN 1 PPC PPE 5 UNL
```

Bus Activity.

assert ATN if not already so
PPC

recommended sequence (if not using CONFIGURE)
UNT
UNL
LAG
PPC
PPE includes line assignment and sense bit data
UNL

PPD. Sends Parallel Poll Disable.

Purpose. The secondary command Parallel Poll Disable is sent to the IEEE bus.

The PPD command is used as part of a sequence to prevent the particular addressed device from responding to parallel polls. The previously assigned data lines is remembered.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command PPD, hex 070h decimal 112, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The PPC command must precede the PPD command for a correct bus sequence to be performed. The PPD command is used to tell the addressed listener that it must no longer respond to the controller's parallel poll command. Though unable to respond the device remembers its parallel poll line assignment made by the previously issued PPC and PPE commands, probably using the CONFIGURE instruction.

The PPD command has been included for completeness only, it is advised that the DISABLE command be used to disable the response of a device to Parallel Poll.

The PPE command can also be sent by using CMD 112 or CMD 240 or SEC 16

Format. PPD

Params. None.

Returns. None.

Example.

```
BUS PPD
recommended sequence (if not using DISABLE )
BUS UNT UNL LISTEN 1 PPC PPD UNL
```

Bus Activity.

assert ATN if not already so
PPC
recommended sequence (if not using DISABLE)

UNT
UNL
LAG
PPC
PPD
UNL

PPE. Sends Parallel Poll Enable Command.

Purpose. The secondary command Parallel Poll Enable is sent to the IEEE bus.

This command is used as part of a sequence to allow the addressed listener to be assigned a data line for response to the parallel poll command.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command PPC, hex 060h decimal 96, ORed with byte the user typed and is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The PPC command must precede the PPE command for a correct bus sequence to be performed. The PPE command is used to tell the addressed listener which particular data line it is to use when responding to a parallel poll, the PPE command also tells the addressed device whether to assert the line when it requires service or to assert the line when it does not require service.

The PPE command has the bit value %0110sppp. Here s is the sense of response bit and ppp is the data line assigned to the device.

The sense information is in bit 3 (decimal 8).
If bit 3=0 then assert line when the device's individual status bit, IST, is zero.
If bit 3=1 then assert line when the device's IST is one.

The data line is assigned using bits 0-2.
If bits= 000, decimal 0, then data line 0
If bits= 001, decimal 1, then data line 1
If bits= 010, decimal 2, then data line 2
If bits= 011, decimal 3, then data line 3
If bits= 100, decimal 4, then data line 4
If bits= 101, decimal 5, then data line 5
If bits= 110, decimal 6, then data line 6
If bits= 111, decimal 7, then data line 7

The sense value and the data line value are logically OR'd together with the PPE command ($60) and sent to the bus after PPC with ATN true.

The PPE command has been included for completeness only, it is advised that the CONFIGURE or DISABLE commands be used to configure a device for Parallel Poll since both these commands automatically send the PPE command.
The PPE command can also be sent by using CMD 96-111 or CMD 224-239 or SEC 0-15

Format. PPE

Params. One in the range 0-15

Returns. None.

Example.

```
BUS PPE 1
recommended sequence (if not using CONFIGURE)
BUS UNT UNL LISTEN 1 PPC PPE sppp UNL
```

Bus Activity.

assert ATN if not already so
PPC

recommended sequence (if not using CONFIGURE)
UNT
UNL
LAG
PPC
PPE includes line assignment and sense bit data
UNL

PPOLL. Perform A Parallel Poll.

Purpose. To perform the Parallel Poll function, this allows the controller to simultaneously determine the status of several devices.

Unlike the Serial Poll which is conducted after a DEVICE Requests Service, the Parallel Poll routine is initiated by the controller. It allows a whole group of devices to simultaneously return a status bit on the data lines. The data line, and the sense of each devices response is determined by the controller when it send CONFIGURE or PPC, parallel poll configure. Devices power up in the PPU state and cannot respond to the Parallel Poll until after they have been configured.

The Parallel Poll makes good use of the controller's time since it can quickly check the state of 8 or more devices with one command.

The PSI BOX knows which device is responsible for which data line because it has previously assigned those lines to each device using the PPC with PPE commands. Depending on the Parallel Poll Response the controller may take a variety of actions, chosen by the user's program.

The PSI BOX asserts EOI and ATN, so becoming the active controller, the PSI BOX then waits minimum of 25 micro seconds for the configured IEEE devices to put their parallel poll status bit on the IEEE data lines. The PSI BOX reads the data line and store the response in bytes 36-38 of the IEEE control string. The PSI BOX then releases the EOI line so ending the parallel poll. The ATN remains true, PSI BOX is the active controller.

The 3 bytes 36-38 in the IEEE control status string contains the parallel poll response read during the last PPOLL . These bytes hold a value between 0 and 255. Knowing what parallel poll configure information he has sent the user can interpret the data accordingly.

Format. PPOLL

Params. None.

Returns. IEEE control status string byte 36-38 hold the parallel poll response.

Example.

```
BUS POLL                :REM ISSUE COMMAND
BUS STATUS              :REM READ STRING
PP at position 36 :REM PP=000 to 255
```

Bus Activity.

assert ATN and EOI
after 25 micro sec read data lines into PSI BOX
drop EOI

PPU. Sends Parallel Poll Unconfigure.

Purpose. The universal command Parallel Poll Unconfigure is sent to the IEEE bus. Resets ALL devices with the Parallel Poll ability to the idle state.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command PPU, hex 15h decimal 21, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The PPU command is used to make all devices that have the Parallel poll ability unable to respond to a parallel poll command.

For a device to respond to the Parallel Poll it must receive a PPC and PPE command sequence or be programmed using the CONFIGURE command. All devices power up in the PPU state. i.e they cannot respond to Parallel Poll.

The PPC command can also be sent by using CMD 21 or CMD 149.

Format. PPC

Params. None.

Returns. None.

Example.

```
BUS PPU
```

Bus Activity.

assert ATN if not already so
PPU

READ. Receive One Byte From Addressed Talker.

Purpose. To receive one byte of data from the currently addressed talker.

The READ command is used to receive data from the currently addressed talker one byte at a time. The data byte is returned in the IEEE control status string at bytes 66-68. The READ command must be used as one command in a series that explicitly addressed devices to talk and listen.

The READ command can be used time and again to retrieve a whole string of data however this is a tedious method, a much easier solution is the ENTER command.

The PSI BOX drops ATN false, if not already so, at the start of the data input sequence ATN is left false until further instructions makes the PSI BOX become the active controller.

Format. READ

Params. None.

Returns. Byte 66-68 in control status string contains the decimal value of the byte read.

Example.

```
BUS MLA TALK 4 READ UNT
Bytes 66-68 in status string hold the data.
```

Bus Activity.

assert ATN if not already so
LISTEN address of the PSI BOX
TAD 4 command this device to send data
DAB XXX the byte of data
UNT tell device to stop talking

REMOTE. Put Device Under PSI BOX Control.

Purpose. The REMOTE command is used to put one or more devices under PSI BOX control and make them ready to receive IEEE commands.

If no device address is specified then the PSI BOX assert the REN, remote enable, line. The PSI BOX is putting the IEEE bus devices under its control via the bus. If you press the devices front panel "LOCAL" button it will return to local control unless it has been put into the LOCAL LOCKOUT state. If one or more devices addresses are given then those devices are commanded to Listen and become ready to receive instruction from the PSI BOX controller.

Byte 11 in the IEEE control status string contains 1 when the REN line is asserted true, PSI BOX remotely controlling bus devices, and 0 when the REN line is false, IEEE devices under local control.

Whenever a read or write is performed or whenever any bus command is sent the PSI BOX re asserts the REN line.

Format. REMOTE

Params. None, one or several.

Returns. IEEE control status string byte 11 = 1
If the REN line was already true then this command has no effect.

Example.

```
BUS REMOTE                  :REM ISSUE COMMAND
BUS REMOTE 2,3,4            :device 2,3,4 wait for data
BUS STATUS                  :REM READ STRING
REN at position 11          :REM REN = 1
```

Bus Activity.

assert REN & assert ATN
or assert ATN if not already so
LAG 2 first addresses specified
....LAG 4 last address specified

REN. Assert The Ren Line.

Purpose. To cause the PSI BOX to assert the REN, remote enable, line. The PSI BOX is putting the IEEE bus devices under its control via the bus. If you press the devices front panel "LOCAL" button it will return to local control unless it has been put into the LOCAL LOCKOUT state.
Byte 11 in the IEEE control status string contains 1 when the REN line is asserted true, PSI BOX remotely controlling bus devices, and 0 when the REN line is false, IEEE devices under local control.

Whenever a read or write is performed or whenever any bus command is sent the PSI BOX re asserts the REN line.

Format. REN

Params. None.

Returns. IEEE control status string byte 11 = 1
If the REN line was already true then this command has no effect.

Example.

```
BUS REN                    :REM ISSUE COMMAND
BUS STATUS                 :REM READ STRING
REN at position 11         :REM REN = 1
```

Bus Activity.

assert REN.

RESET. Specifies The Power On Reset Character.

Purpose. To change the character that the PSI BOX recognises as a command to perform an immediate power on reset.

The @ character, decimal 64, is the default reset character. Whenever the PSI BOX in mode 7 receives the reset character from the Host RS232 computer it performs a complete power on reset. Everything is returned to the default state and an IEEE bus ABORT command is performed. An ABORT command pulses the IFC line for approximately 100 micro seconds, then asserts the REN and ATN line true.

This reset function is to get you out of trouble if the IEEE bus devices hang up when, say, timeouts are disabled and a device crashes.

Bytes 70-72 in the IEEE control status string contains the current RESET value, default is 64, the @ character.

Format. RESET 33

Params. One in the range 000 to 255.

Returns. The IEEE string holds the current RESET value in position 70-72.

Example.

```
BUS RESET 33
The emergency reset byte is now !
```

SEC. Sends The Secondary Address.

Purpose. The secondary address is sent to the IEEE bus.

The SEC command is used to send a secondary address in the range 0 to 30 to the bus devices.

The PSI BOX asserts ATN and becomes the active controller, then the number entered by the user is ORed with hex 060h, 96 decimal, and is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The SEC should only be used after a listen address or a talk address or the PPC command. The SEC command can also be sent by using CMD 96-20 or CMD 148

Format. SEC 1

Params. One in the range 0-30 or 128 to 158

Returns. None.

Example.

```
BUS LISTEN 5 SEC 1
Tell device 5 secondary address 1 to listen.
```

Bus Activity.

assert ATN if not already so
LAG 5
SCG 1

SDC. Sends Selective Device Clear Command.

Purpose. The addressed command Selective Device Clear is sent to the IEEE bus.

The SDC command is used to send the currently listening devices to a device dependent state.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command SDC, hex 04h decimal 4, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The SDC command is used automatically in the CLEAR routine but it can be called separately to perform your own device clear routine. The SDC command can also be sent by using CMD 4 or CMD 132

Format. SDC

Params. None.

Returns. None.

Example.

```
BUS SDC
```

Bus Activity.

assert ATN if not already so
SDC

SHOW. Display & Explain Control Status String.

Purpose. The PSI BOX sends the control status string back to the host computer as a series of labelled fields.

The control status string is an ASCII string of 0's, 1's and other bytes separated by commas. It is a table of information about the IEEE bus, the PSI BOX and the current device.

Now that the status string is over 100 bytes long, the SHOW command has been added to enable the user to have a complete and quick explanation of the PSI BOX status state from any application.

Format. SHOW

Params. None.

Returns. Prints on screen explanation of the PSI BOX status string.

Basic Example.

```
BUS SHOW                                    :REM" Explain status
```

Bus Activity.

None.

SPD. Sends Serial Poll Disable Command.

Purpose. Disables serial poll mode.

The PSI BOX asserts ATN, if not already true, and becomes the active controller, then the IEEE command SPD, hex 019h decimal 25, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The SPD command is used automatically in the SPOLL routine but it can be called separately to finish your own serial poll routine. The SPD command can also be sent by using CMD 25 or CMD 153

Format. SPD

Params. None.

Returns. Only if used as part of a full serial poll sequence would a response from the IEEE bus be forthcoming.

Example.
```
BUS SPD                  :REM ISSUE COMMAND
```

Bus Activity.

assert ATN if not already so
SPD

Example.
```
A COMPLETE SERIAL POLL ROUTINE.
BUS SPE TALK 4 READ SPD UNT
BUS STATUS               :REM READ STRING
SP at position 11        :REM HOLDS REPLY.
```

Bus Activity.

assert ATN if not already so
SPE
TAD 4
drop ATN and read serial poll status byte.
SPD
UNT

SPE. Sends Serial Poll Enable Command.

Purpose. Initiates a serial poll.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command SPE, hex 018h decimal 24, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The SPE command is used automatically in the SPOLL routine but it can be called separately to initiate your own serial poll routine. If you have, say, two devices that can request service, instead of serial polling them sequentially using SPOLL, you may wish to poll them together. You would then use SPE as part of your own routine. The SPE command can also be sent by using CMD 24 or CMD 152

Format. SPE

Params. None.

Returns. Only if used as part of a full serial poll sequence would a response from the IEEE bus be forthcoming.

Example.

```
BUS SPE           :REM ISSUE COMMAND
```

Bus Activity.

assert ATN if not already so
SPE

Example.

```
A COMPLETE SERIAL POLL ROUTINE.
BUS SPE TALK 4 READ SPD UNT
BUS STATUS              :REM READ STRING
SP at position 11 :REM HOLDS REPLY.
```

Bus Activity.

assert ATN if not already so
SPE
TAD 4

drop ATN and read serial poll status byte.
SPD
UNT

SPOLL. Perform A Complete Serial Poll.

Purpose. To determine which device has requested service from the controller and why.

The SERIAL POLL is usually called after a device asserts SERVICE REQUEST by setting SRQ low. Serial poll is the only method that can correctly cause the device to return the SRQ line false. In general, you will know which device requested the service and so will know which to poll. However, if it could be one of several devices that is requesting service, you should perform successive serial polls on the different devices until the SRQ line returns to the false state.

NOTE: It is the IEEE device that causes the serial poll to be performed because it asserts SRQ when it needs servicing by the controller.

Format. SPOLL serially polls the currently addressed device. The serial poll response is stored in bytes 28-30, the current device serial poll data, and in bytes 32-34, the first addressed device serial poll data, in the IEEE control status string.

6,SPOLL makes device 6 the current device, then serially polls the currently addressed device.

SPOLL 6,7,8 serially polls the three devices 6,7 and 8 one after another. The serial poll response of each device is held in bytes 28-30 of the IEEE control status string when that device is addressed as the current device. The serial poll response for device 6 is placed in bytes 32-34, the serial poll data of the first addressed device.

Params. None, One or Some.

Returns. The serial poll response contains data concerning the service request.

it 6 (value 64) set if it was the device requesting service, if so then the other bits will contain a code indicating the nature of the service requested.

Bit 6 (value 64) not set then this device did not request service.

Example.

```
BUS STATUS              :REM PUT THE WHOLE STRING HERE
SRQ at position 7       :REM SRQ I/P =1 DEVICE NEEDS SERVICE
IF SRQ = 0 THEN ...     :REM WAIT FOR SRQ....
      :REM GOT SRQ
BUS SPOLL               :REM POLL CURRENT DEVICE
BUS STATUS              :REM READ THE REPLY
SP at position 28       :REM POLL DATA
IF SP AND 64 = 64 THEN ...    :REM DID WE TRY THE RIGHT DEVICE
            :REM MUST BE ANOTHER DEVICE REQUESTING SERVICE
                              :REM GOT RIGHT DEVICE
IF SP =.....                  :REM SP HOLDS REASON FOR SRQ
```

Bus Activity.(SPOLL) or (SPOLL n)

UNL
MLA
SPE
TAD (drop ATN)
ACPTR (receive byte from device)
assert ATN
SPD
UNT (leave ATN asserted)

Bus Activity.(SPOLL n,n,n)

UNL
MLA
SPE
TAD first address (drop ATN)
ACPTR (receive byte from 1st device)
assert ATN
TAD second address (drop ATN)
ACPTR (receive byte from 2nd device)
assert ATN
TAD third address (drop ATN)
ACPTR (receive byte from 3rd device)
assert ATN
SPD
UNT (leave ATN asserted)

STATUS. Send The Status String To RS232 Host.

Purpose. To cause the PSI BOX to send the IEEE control string back to the host RS232 computer.

The PSI BOX stores a table of data that it uses to inform itself of which devices to access and in what manner it should control the bus. This table of data is called the IEEE control status string. The IEEE control status string also holds information on parallel and serial poll responses, time out data, byte counts of data transferred during the previous i/o and many other interesting parameters. The first field of the IEEE control status string holds the current IEEE device number, the second field is that device's secondary address.

The current PSI BOX status string will be sent from the PSI BOX to the RS232 host computer whenever the PSI BOX receives the STATUS command.

Alternatively, for increased ease of use, a BUS command without any parameters or keywords will be treated as a STATUS command. Any spaces after BUS but before the <CR> are ignored.

The IEEE control string can be obtained in a more human readable form by the SHOW command.

A request for the STATUS causes the PSI BOX to return the IEEE control status string to the RS232 host computer. This is a string of Ascii characters preceded by the word STATUS and terminated by a CR character, decimal 13. The string is made up of a series of numbers separated by commas and looks like:-

```
STATUS 04,  ,0,1,1,0,00,00000,00000,000,000,000,0,0,1 etc
```

Currently the IEEE control status string is approximately 90 bytes long, as new features are added to the PSI BOX software the length of the control status string increases. It is advised that 256 bytes of space are set aside for the IEEE control status string in your program to allow complete compatibility with future version of the Mode 7 software. The format of the data returned is given in chapter 7.

Format. STATUS

Params. None.

Returns. Up to 255 bytes of data starting with
STATUS 04, ,00 etc.

Example.

```
            BUS STATUS <Carriage Return>
or          BUS  <Carriage Return>
```

Also see the SHOW command.

STRING. Send A String Of Data To A Bus Device.

Purpose. To send a string of data to the IEEE bus but without any addressing information.

The STRING command does not provide any addressing information about who should receive the data, that must be specified before the STRING command. The STRING command simply sends the text enclosed in the delimiters to the bus.

The STRING command is a simple method of adding text in the middle of a bus data exchange.

Format. OUTPUT "HELLO"

Params. Delimiter string delimiter.

Returns. None.

Example.

```
BUS MTA LISTEN 4 STRING !HELLO! EOI 13 UNL
This will send the text HELLO to the current bus device.
```

Bus Activity.

assert ATN if not already so
TAD PSI BOX is the talker default address is 0
LISTEN 4 command device 4 to listen
DAB 68 (H)
DAB 65 (E)
DAB 72 (L)
DAB 72 (L)
DAB 79 (0)
DAB 13 (0) with EOI true
UNLISTEN

TAD. Sends The Talk Address.

Purpose. The talk address is sent to the IEEE bus.

The TAD command is used to send one talk address in the range 0 to 30 to the bus devices.

The PSI BOX asserts ATN and becomes the active controller, then the number entered by the user is ORed with hex 040h, 64 decimal, and is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The TAD is used to instruct one bus device that it is the active talker following in the Serial Poll routine or in data transfers. Only one talker can be active, addressed at one time else confusion would reign! The TAD command is treated as an UNTALK command by all the other talkers. When the current talker recognises that another talker has been addressed to talk then it is automatically put in the untalk state. The TAD command can also be sent by using CMD 64-94 or CMD 192-222 or by using TALK 0-30 or 128-158

Format.

TAD 1	Tell device 1 to talk.
TAD 1+2	Device 1 with Secondary address 2, is the new talker.

Params. One parameter only in the range 0-30 or 128 - 158.
An optional secondary address can be specified for the primary address by using +sa after the primary address. The sa is also in the range 0-30.

Returns. None.

Example.

```
BUS TAD 1+2 ;tell device 1 with second address 2 to talk.
```

Bus Activity.

assert ATN if not already so
TAD 1
SCG 2

TALK. Sends The Talk Address.

Purpose. The talk address is sent to the IEEE bus.

The TALK command is used to send one talk address in the range 0 to 30 to the bus devices.

The PSI BOX asserts ATN and becomes the active controller, then the number entered by the user is ORed with hex 040h, 64 decimal, and is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The TALK is used to instruct one bus device that it is the active talker following in the Serial Poll routine or in data transfers. Only one talker can be active, addressed at one time else confusion would reign! The TALK command is treated as an UNTALK command by all the other talkers. When the current talker recognises that another talker has been addressed to talk then it is automatically put in the untalk state. The TALK command can also be sent by using CMD 64-94 or CMD 192-222 or by using TAD 0-30 or 128-158

Format.

TALK 1	Tell device 1 to talk.
TALK 1+2	Device 1 with Secondary address 2, is the new talker.

Params. One parameter only in the range 0-30 or 128 - 158.
An optional secondary address can be specified for the primary address by using +sa after the primary address. The sa is also in the range 0-30.

Returns. None.

Example.

```
BUS TALK 1+2 ;tell device 1 with second address 2 to talk.
```

Bus Activity.

assert ATN if not already so
TAD 1
SCG 2

TCT. Sends The Take Control Command.

Purpose. The addressed command Take Control is sent to the IEEE bus.

The TCT command is used after a talk address to tell the device to take control of the IEEE bus.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command TCT, hex 04h decimal 4, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller. The NO ATN command should be used to drop the ATN line false so allowing the addressed controller to assert ATN and become the active controller.

The TCT command is used automatically in the PASS routine but it can be called separately to perform your own transfer control routine. The TCT command can also be sent by using CMD 9 or CMD 137

Format. TCT

Params. None.

Returns. None.

Example.

```
BUS TCT
recommended sequence
BUS UNL TALK 3 TCT NO ATN
```

Bus Activity.

assert ATN if not already so
TCT

recommended sequence.
UNL
TAD 3
TCT
(PSI BOX drops ATN new controller asserts ATN)

TIMER. Wait For X Milli Seconds.

Purpose. The PSI BOX waits for the given number of milli seconds to pass.

The TIMER command is used to allow the PSI BOX to wait a fixed interval between bus accesses. Typically it is used in a repeating MACRO so that a series of readings are taken once every given interval.

The interval range is from 1 to 65535 milli seconds, ie from 1/1000th of a second to 65.535 seconds.

It must be noted that due to the overhead of decoding the command etc, timings less than 100 milli seconds may appear inaccurate. Use as long a time as possible for higher percentage accuracy.

The TIMER command is identical to the WAIT command.

Format. TIMER

Params. One. In the range 1-65535 milli seconds.

Returns. None.

Example.

```
BUS TIMER 1000                  :REM Waits for 1 sec
BUS TIMER 60000                 :REM Waits for 1 minute
```

Bus Activity.

None.

TO. Set The Timeout Period.

Purpose. To set the length of time the PSI BOX will wait in a bus handshake routine till the responding device is ready for the data transfer. This prevents the PSI BOX hanging up when the device is not ready.

When the PSI BOX is exchanging data with a bus device it may need to wait a long time for the data to become present, or for the device to make data available. The TO command can be used to set the maximum time that the PSI BOX waits for the device to become ready, thus preventing the PSI BOX waiting for too long when a device is not ready. By contrast, the NO TO command can be used to make the PSI BOX wait indefinitely for the device to be ready to handshake data.

The TO parameter is used to determine the length of wait before the PSI BOX times the device out.

Bytes 57 and 58 in the IEEE control status string contains the current timeout value.

A value of 0 indicates that the timeouts have been disabled.

The timeouts step up in 1,2,5,10 increments from 1 milli second to 50 seconds. Therefore each timeout increment represents an increase in PSI BOX wait time of approximately a factor of 2.

The timeouts can also be disabled using NOTO or by TO 0.

Format. TO

Params. One in the range 00 to 15.

Returns. The IEEE string holds the current timeout value in position 57-58.

NO TO gives a TO value of 0, means wait forever.

TO 0	Means wait forever.!!! careful!!	
TO 1	1	milli second wait till timeout.
TO 2	2	milli second wait till timeout.
TO 3	5	milli second wait till timeout.
TO 4	10	milli second wait till timeout.
TO 5	20	milli second wait till timeout.

TO 6	50	milli second wait till timeout.
TO 7	100	milli second wait till timeout.
TO 8	200	milli second wait till timeout.
TO 9	500	milli second wait till timeout.
TO 10	1	Second = 1000 milli seconds.
TO 11	2	Second wait till timeout.
TO 12	5	Second wait till timeout.
TO 13	10	Second wait till timeout.
TO 14	20	Second wait till timeout.
TO 15	50	Second wait till timeout.

Example.

```
BUS TO 10               :REM ISSUE COMMAND
BUS STATUS              :REM READ TIMEOUT
TMO at position 57      :REM SHOULD BE 10.
                        :IE 1 second wait.
```

TRIGGER. Tell Devices Perform Predefined Action.

Purpose. One or more devices receives the addressed command Group Execute Trigger so causing them to respond by performing a previously defined set of actions.

The TRIGGER command is a very flexible way to command one or more devices to take part in a test and measurement routine. If the devices are say device 2 a voltage supply and device 3 an ammeter. Having previously specified that the voltage source outputs put 5 volts dc across the device under test, a resistor, and the ammeter measures the current in milli Amps, the TRIGGER 2,3 command performs the test on the resistor.

If no device address is specified then the PSI BOX simply sends the GET command, this will active all currently listening devices. If one or more devices addresses are given then those devices are commanded to Listen and then told via the GET command to perform the pre programmed action.

Format. TRIGGER

Params. None, one or more.

Returns. None.

Example.

```
BUS TRIGGER              REM listeners to trigger
BUS TRIGGER 2,3,4        REM devices 2,3 & 4 trigger
```

Bus Activity.

assert ATN
GET prior addressed devices to perform action.

or assert ATN if not already so
MTA of PSI BOX
UNL all stop listening
LAG 2 first address specified
....
LAG 4 last address specified
GET addressed devices simultaneously act.

UNCONFIGURE. Prevents Devices Responding To PPoll.

Purpose. To prevent all devices responding to the parallel poll command.

The UNCONFIGURE command is included for completeness, it performs identically to the PPU command, it prevents all devices, which have been previously CONFIGUREd, from responding to parallel polls.

All the devices will need to be reCONFIGUREd if they are to take part in future parallel polls.

Format. UNCONFIGURE

Params. None.

Returns. None.

Example.

```
BUS UNCONFIGURE   all devices now unable to respond
```

Bus Activity.

assert ATN if not already so
PPU all devices cannot respond

UNL. Sends The Unlisten Command.

Purpose. The UNLISTEN command is sent to the IEEE bus.

The UNL command is to stop the current listeners from listening to the following IEEE commands and data. After an UNLISTEN command devices must be LISTENed before they can receive data or respond to any of the addressed command group.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command UNL, hex 03Fh decimal 63, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The UNL command is used automatically in the write routine when data is sent to the bus using the OUTPUT command. It can be called separately to unlisten all bus devices prior to selectively listening one or more devices. The UNL command can also be sent by using CMD 63 or CMD 191

Format. UNL

Params. None.

Returns. None.

Example.

```
BUS UNL
```

Bus Activity.

assert ATN if not already so
UNL

UNT. Sends The Untalk Command.

Purpose. The UNTALK command is sent to the IEEE bus.

The UNT command is to stop the current talker from sending any more data. After an UNTALK command one device must be TALKed before it can source any data.

The PSI BOX asserts ATN and becomes the active controller, then the IEEE command UNT, hex 05Fh decimal 95, is sent over the IEEE bus. After the command has been sent the ATN remains true, PSI BOX is the active controller.

The UNT command is used automatically in the read routine when data is read to the bus using the ENTER command. It can be called separately to untalk all bus devices prior to terminate a data transfer sequence. A TALK command to a device other than the current talker automatically untalks the active talker. The UNT command can also be sent by using CMD 95 or CMD 223

Format. UNT

Params. None.

Returns. None.

Example.

```
BUS UNT
```

Bus Activity.

assert ATN if not already so
UNT

VERSION. Send PSI Rom Version To RS232 Host.

Purpose. To cause the PSI BOX to send its operating system version number and date back to the host RS232 computer.

The PSI BOX ROM version can be used to identify the VERSION of software and release date and so help to prevent any problems from inconsistencies in the implementation of the PSI BOX firmware.

The current PSI BOX Version will be sent from the PSI BOX to the RS232 host computer whenever the PSI BOX receives the VERSION command.

A request for the VERSION causes the PSI BOX to return a message to the RS232 host computer. This is a string of Ascii characters of the following form :-

```
PSI BOX Version 4.0f 16/06/90
```

The Version string is approximately 30 bytes long and is terminated by a Carriage Return linefeed pair. It is advised that 40 bytes of space are set aside for the Version status string in your program.

Format. VERSION

Params. None.

Returns. Up to 40 bytes of data starting with
PSI BOX Version x.xx xx/xx/xx <CR LF>

Example.

```
BUS VERSION <Carriage Return>
```

WAIT. Wait For X Milli Seconds.

Purpose. The PSI BOX waits for the given number of milli seconds to pass.

The WAIT command is used to allow the PSI BOX to wait a fixed interval between bus accesses. Typically it is used in a repeating MACRO so that a series of readings are taken once every given interval.

The interval range is from 1 to 65535 milli seconds, ie from 1/1000th of a second to 65.535 seconds.

It must be noted that due to the overhead of decoding the command etc, timings less than 100 milli seconds may appear inaccurate. Use as long a time as possible for higher percentage accuracy.

The WAIT command is identical to the TIMER command.

Format. WAIT

Params. One. In the range 1-65535 milli seconds.

Returns. None.

Example.

```
BUS WAIT 1000                    :REM WAIT for 1 sec
BUS WAIT 60000                   :REM WAIT for 1 minute
```

Bus Activity.

None.

WAIT SRQ. Wait Till Srq Is True.

Purpose. The PSI BOX waits till the SRQ line is asserted true by a device.

The WAIT SRQ command is used to allow the PSI BOX to wait till a device requests service. This usually signals that the device has some new data ready or has finished completing some pre programmed job. The PSI BOX will wait for SRQ until timeout or until the SRQ is true. If you want the PSI BOX to wait indefinitely for SRQ then you should disable the timeout using the NO TO command.

Once the SRQ line has been asserted true you should perform a serial poll, using the SPOLL command, to determine which device has requested service and exactly why.

On entry the PSI BOX checks the state of SRQ and if it is already asserted true then it exits the routine immediately. Otherwise it waits till the SRQ changes from the false to the true state. If timeouts are enabled then the PSI BOX will exit the WAIT SRQ routine if SRQ has not been asserted within the timeout period.

Format. WAIT SRQ

Params. None.

Returns. None.

Example.

```
BUS TO 1 WAIT SRQ            :REM WAIT secs for SRQ
BUS NO TO WAIT SRQ           :REM WAIT indefinitely
```

Bus Activity.

None.

X. Executes Stored Macro.

Purpose. Causes the PSI BOX to execute the stored macro command.

The X command can be embedded within any valid PSI BOX command string or be by itself. The X command takes one optional parameter that determines the number of times it is to execute. The X parameter range is 1-65535. If X 0 is specified the PSI BOX will perform the macro indefinitely, SO BEWARE!

The X command cannot call itself. IE The X command cannot be embedded within a macro command string, otherwise an error is flagged when the macro is executed.

Used carefully and intelligently the X and MACRO commands provide a powerful and elegant means of automating the repetitive collection of large amounts of data.

Format. X

Params. One or None.

Returns. Any depending on macro contents.

Example.

```
BUS X       :REM Executes macro once.
BUS MTA LISTEN 4 X UNT UNL    :REM  macro  within  command
string
BUS X,1000  :REM Executes macro 1000 times.
BUS OUTPUT "CURVE?" X 1024 UNL UNT
                                  :REM Executes 1024 times
```

Bus Activity.

any

Index.

7 bits.................................43, 69, 97, 102
8 bits.................................2, 11, 37, 43, 79, 94, 97, 99, 100-102

Abort.................................66, 79, 82, 84, 85, 87, 106, 139
Active controller...............17, 18, 19, 78, 87, 88, 89, 90, 94, 95, 101, 104, 105, 107, 108, 109, 111, 116, 118, 120, 122, 127, 129, 131, 133, 135, 136, 140, 141, 143, 144, 151, 152, 153, 159, 160
Ascii..................................10, 12, 49, 50, 51, 52, 53, 54, 55, 73, 76, 77, 79, 80, 86, 142, 148, 161
Atn....................................13, 14, 15, 17, 18, 19, 20, 25, 30, 70, 72, 78, 79, 81, 82, 87, 88, 89, 90, 93, 94, 95, 96, 101, 104, 105, 107, 108, 109, 110, 111, 116, 118, 120, 122, 126, 127, 128, 129, 131, 132, 133, 134, 135, 136, 137, 139, 140, 141, 143, 144, 145, 147, 150, 151, 152, 153, 157, 158, 159, 160
Attention............................14, 20, 72, 77, 78, 87, 88, 90, 94, 122
AutoCAD..........................60, 62, 64

Basic.................................5, 20, 68, 69, 142
Baud rate...........................2, 7, 10, 37, 38, 39
Buffer................................1, 2, 4, 5, 27, 32, 33, 34, 38, 41, 42, 52, 53, 55, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 82, 83, 124
Buffer switch....................4
Bus command...................23, 68, 70, 71, 72, 73, 74, 75, 77, 78, 80, 81, 82, 106, 114, 115, 123, 137, 138, 148
Bus status..........................77, 78, 87, 88, 112, 122, 123, 125, 134, 137, 138, 143, 144, 147, 149, 156

C.......................................33, 34, 97, 102
Cable length......................9, 15

Cables................................ 7, 29, 35, 44
Carriage return.................. 19, 24, 74, 79, 85, 86, 94, 98, 101, 103, 112, 161
Centronics handshake....... 55
Centronics input................ 34, 36, 50, 51, 52, 58, 60, 61
Centronics leds................. 31, 32
Centronics output............. 31, 33, 35, 50, 54, 55, 59
Centronics port................. 5, 7, 26, 28, 31, 32, 33, 34, 35, 52, 54, 55
Clear................................. 14, 20, 22, 45, 46, 65, 66, 72, 81, 82, 84, 85, 87, 89, 90, 91, 95, 106, 141
Cmd.................................. 81, 86, 90, 94, 95, 104, 105, 107, 108, 109, 127, 129, 132, 135, 140, 141, 143, 144, 151, 152, 153, 159, 160
Com1................................ 2, 37, 59, 69
Commodore...................... 7, 28, 49, 51, 52, 54, 55, 60, 62, 69
Commodore pet................ 7, 28, 49, 51, 52, 54, 55
Compiled.......................... 69
Configure.......................... 1, 6, 21, 22, 81, 82, 86, 92, 93, 127, 128, 129, 132, 133, 135
Connecting........................ 10, 44, 45, 47, 58
Controller.......................... 3, 4, 7, 14-23, 25-29, 46, 49, 50, 51, 52, 54, 56, 57, 60-63, 66, 69, 70, 78, 87-90, 92, 94, 95, 101, 104, 105, 107-111, 116, 118, 120, 122, 127, 129, 131, 133, 135, 136, 137, 140-144, 146, 151, 152, 153, 159, 160
Controller in charge.......... 17, 18, 19
Controller state................. 28, 29
Cr..................................... 66, 74, 77, 148
Cts.................................... 2, 40, 41, 44, 45, 46, 47, 48, 57
Current device.................. 21, 68, 71, 72, 75, 78, 79, 80, 81, 82, 85, 86, 126, 142, 146, 147
Current ieee device........... 70, 72, 74, 76, 79, 126, 148
Current secondary addrss.. 72

Data word......................... 39, 43
Dav.................................. 12, 13, 26, 30

Dcd.................................. 2, 40, 41, 44, 45, 46, 47, 48, 57
Dcl.................................... 22, 81, 82, 89, 91, 95
Default.............................. 68, 70, 71, 72, 75, 78, 79, 80, 97, 100, 102, 103, 114, 115, 116, 117, 118, 119, 120, 121, 126, 139, 150
Device clear...................... 20, 22, 81, 82, 89, 90, 91, 95, 141
Device=............................ 77
Dip switch......................... 1, 2, 4, 27, 28, 31, 32, 39, 43, 49, 52, 53, 55, 82, 83
Disable.............................. 81, 82, 91, 96, 99, 109, 111, 114, 124, 127, 129, 132, 143, 163
Disk.................................. 16, 17
Drop.................................. 90, 101, 110, 134, 143, 145, 147, 153
Dsr.................................... 2, 40, 41, 45, 46, 47, 48, 57
Dtr.................................... 2, 40, 41, 44, 45, 46, 47, 48, 57

End.................................... 6, 15, 23, 24, 56, 67, 71, 72, 77, 79, 82, 85, 87, 88, 94, 97, 99, 100, 101, 102, 112, 126
End marker....................... 79
End=................................. 82, 97, 98, 99
End=off............................. 97, 98, 99
End=on............................. 100
Enter................................. 75, 76, 79, 82, 86, 113, 136, 160
Eoi.................................... 13, 15, 24, 30, 52, 55, 56, 72, 74, 75, 79, 81, 86, 94, 97, 98, 99, 100, 101, 102, 103, 126, 133, 134, 150
Eos.................................... 24, 71, 76, 77, 79, 82, 85, 102, 124
Eos=.................................. 72, 82, 99, 102, 103
Error................................. 35, 50, 64, 65, 66, 73, 74, 77, 78, 82, 89, 112, 114, 115, 164

Forced true........................ 41
Format.............................. 71, 76, 78, 87-164
Function............................ 1, 20, 21, 22, 25, 31, 37, 46, 49, 51, 54, 57, 58, 59, 60, 62, 64, 65, 69, 76, 79, 84, 133, 139

Get.................................... 52, 68, 79, 81, 90, 104, 139, 157
Go to local......................... 23, 81, 90, 105, 110
Gtl.................................... 23, 81, 90, 105, 109, 110, 111

Handle.............................. 76
Handshake......................... 2, 12, 16, 26, 31, 33, 34, 37, 39, 40, 41, 44, 52, 55, 56, 57, 58, 59, 60, 62, 66, 75, 83, 99, 124, 126, 155
Handshakes........................ 69
Hard wired handshake...... 40
Hewlett packard................ 7, 9, 10, 50, 64
Hold.................................. 22, 41, 61, 82, 133, 134, 136
Hp..................................... 4, 51, 54, 56, 58, 60, 62, 64
Hpgl.................................. 64, 65, 66

IBM PC............................. 2, 31, 35, 37, 44, 46, 47, 48, 51, 58, 70
IBM PS/2.......................... 69
Idle.................................... 17, 20, 122, 135
Ieee bus status................... 78
Ieee cables......................... 29
Ieee commands.................. 61, 63, 66, 68, 70, 74, 84, 87, 88, 90, 137, 159
Ieee control data................ 78
Ieee controls...................... 79
Ieee device number........... 56, 72, 76, 78, 148
Ieee functions.................... 80
Ieee handshake.................. 52, 55, 56
Ieee leds............................ 26, 27
Ieee plotter........................ 60, 62, 64, 65
Ieee printer........................ 3, 19, 60, 62, 94, 101, 124
Ifc...................................... 13, 14, 17, 18, 25, 30, 72, 79, 81, 82, 87, 106, 139
Input.................................. 2-5, 7, 24, 26, 31-36, 40, 41, 42, 45, 46, 50, 51, 52, 53, 55, 57-67, 69, 70, 72, 76, 77, 78, 82, 83, 85, 86, 124, 136
Instrument......................... 7, 10, 11, 23, 109, 111
Interface clear................... 14, 72, 87, 106

Lag.................................. 80, 81, 89, 91, 93, 96, 101, 107, 108, 110, 116, 128, 130, 132, 137, 140, 157
Leds................................ 1, 7, 25, 26, 27, 31, 32, 37, 38, 50
Listen.............................. 2, 6, 14, 17, 18, 19, 20, 26, 27, 32, 38, 51, 54, 56, 57, 60, 61, 62, 63, 66, 70, 71, 72, 74, 75, 77, 78, 80, 81, 85, 86, 89, 90, 91, 94, 96, 101, 107, 108, 110, 116, 117, 121, 126, 127, 129, 132, 136, 137, 140, 150, 157, 164
Listen address.................. 14, 19, 56, 72, 78, 81, 91, 107, 108, 116, 117, 121, 136, 140
Listen only....................... 2, 20, 51, 54, 60, 62, 66
Llo.................................. 81, 82, 91, 109, 111
Local............................... 14, 20, 23, 81, 82, 90, 91, 105, 109, 110, 111, 123, 137, 138
Local lockout................... 23, 81, 91, 109, 111, 137, 138
Lockout............................ 23, 81, 82, 91, 109, 111, 137, 138
LOTUS............................ 60, 62, 64
LOTUS 123..................... 60, 62, 64
Lpt1................................ 2, 31, 35

Macro............................... 82, 112, 113, 154, 162, 164
Mains power.................... 1
Memory........................... 5, 57, 68, 82, 83
Mla.................................. 78, 81, 116, 117, 118, 136, 147
Mode dip switches........... 50
Mode switches................. 7, 29, 49, 51, 54, 60, 62
Msa................................. 78, 118, 119
Mta.................................. 78, 81, 89, 110, 118, 120, 121, 150, 157, 164
My listen address............. 72, 78, 81, 116, 117
My secondary address...... 72, 78, 118
My talk address................ 72, 78, 81, 120, 121

Ndac................................ 12, 13, 26, 30, 52, 55
No atn.............................. 81, 90, 122, 153
No ren.............................. 81, 123
No secondary address....... 80

No to.................................82, 124, 125, 155, 163
Nrfd..................................12, 13, 26, 30, 52, 55

Open.................................16, 56
Output...............................2, 3, 4, 5, 7, 26, 31, 32, 33, 35, 36, 41, 45, 46, 50-55, 57, 58, 59, 60, 62, 64, 65, 66, 67, 69, 70, 72, 73, 74, 75, 77, 78, 82, 83, 86, 97, 98, 99, 100, 113, 126, 150, 159, 164
Output status......................65

Parallel poll........................15, 21, 22, 71, 78, 81, 82, 86, 90, 91, 92, 96, 127, 129, 131, 132, 133, 134, 135, 158
Parallel poll configure......21, 22, 81, 127, 133
Parallel poll data...............78, 86
Parallel poll disable..........81, 129
Parallel poll enable...........81, 131
Parallel poll response........22, 133, 134
Parallel poll status.............22, 71, 133
Parallel poll unconfigure...135
Parallel port.......................2, 7, 26, 31, 32, 33, 35, 51, 58, 69
Parallel port direction.......33
Parallel port input.............31
Parallel printer..................11, 31, 33, 35, 51, 54, 58, 59, 60
Parity.................................2, 7, 10, 37, 38, 39, 42, 43, 69
Parity enable......................38, 42
Parity even........................38, 42
Parity option......................42
Pass...................................18, 68, 72, 75, 79, 153, 154, 162
Pin outs.............................2, 25, 29, 30, 31, 35, 36, 37, 44, 45
Plotter...............................3, 50, 54, 58, 59, 60, 62, 64, 65, 66, 67, 124
Poll....................................15, 20, 21, 22, 71, 76, 77, 78, 81, 82, 85, 86, 90, 91, 92, 96, 127, 129, 131, 132, 133, 134, 135, 143, 144, 145, 146, 147, 148, 151, 152, 158, 163
Ppc....................................21, 22, 81, 90, 92, 93, 96, 127, 128,

129, 130, 131, 132, 133, 135, 140
Ppd....81, 96, 127, 129, 130
Ppe....22, 81, 92, 93, 127, 128, 129, 131, 132, 133, 135
Ppoll....82, 93, 133, 158
Ppu....21, 81, 91, 96, 133, 135, 158
Print....94, 101, 124
Printer....2, 3, 11, 16, 19, 20, 31, 32, 33, 34, 35, 36, 51, 54, 58, 59, 60, 62, 94, 101, 124
Printer cable....35
Ps/2....70

Read....8, 9, 23, 27, 33, 34, 39, 46, 49, 55, 57, 75, 77, 78, 79, 82, 86, 87, 88, 102, 122, 123, 125, 133, 134, 136, 137, 138, 143, 144, 145, 147, 156, 160
Read data....77, 134
Reading data....24
Receiving data....56, 68, 75
Remote....14, 23, 59, 72, 77, 80, 82, 87, 123, 137, 138
Ren....13, 14, 17, 18, 23, 25, 30, 70, 72, 78, 79, 81, 82, 87, 109, 110, 111, 123, 137, 138, 139
Reset....22, 49, 68, 70, 72, 77, 79, 82, 85, 87, 89, 106, 112, 139
Reset bus....106
Reset devices....87, 89
Rs232....1, 2, 3, 4, 5, 7, 8, 11, 26, 28, 29, 31, 33, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 50, 51, 52, 53, 54, 55, 57, 58, 59, 62, 63, 64, 65, 66, 67, 69, 70, 71, 72, 73, 77, 79, 82, 83, 112, 139, 148, 161
Rs232 handshake....2, 37, 39, 40, 52
Rs232 leds....38
Rs232 port....1, 2, 4, 5, 7, 8, 28, 37, 38, 44, 45, 47, 51, 52, 53, 54, 55, 57, 59, 62, 64, 67,

69, 70, 72, 73, 83

Sdc................................... 22, 81, 82, 86, 89, 90, 141
Sec................................... 72, 77, 81, 85, 86, 113, 118, 119, 129, 132, 134, 140, 154, 162
Secondary address............ 56, 70, 71, 72, 76, 78, 80, 81, 85, 107, 108, 118, 119, 140, 148, 151, 152
Selective device clear....... 89, 90, 141
Sending data..................... 17, 41, 56, 68, 75, 79, 81
Serial poll.......................... 15, 20, 21, 71, 76, 77, 78, 81, 85, 91, 133, 143, 144, 145, 146, 148, 151, 152, 163
Serial poll data.................. 78, 146
Serial poll status................ 143, 145
Serial port.......................... 2, 37, 44, 46, 47, 48, 59, 62, 69
Service request.................. 14, 20, 21, 146
Software handshake.......... 2, 40
Source code....................... 69
Spd................................... 81, 91, 143, 144, 145, 147
Spe................................... 81, 91, 143, 144, 147
Speed................................. 9, 10, 16, 59, 77
Spoll.................................. 74, 78, 81, 82, 84, 85, 143, 144, 146, 147, 163
Srq.................................... 13, 14, 20, 21, 25, 30, 70, 77, 78, 82, 146, 147, 163
Status................................ 21, 22, 56, 57, 65, 66, 68, 70, 71, 73, 75, 76, 77, 78, 80, 82, 87, 88, 92, 112, 117, 119, 121, 122, 123, 124, 125, 131, 133, 134, 136, 137, 138, 139, 142, 143, 144, 145, 146, 147, 148, 149, 155, 156, 161
Stop bits............................ 43
Syntax error...................... 73, 74, 77, 78

Tad.................................... 80, 81, 91, 120, 136, 143, 144, 147, 150, 151, 152, 153
Take control...................... 81, 90, 153
Talk................................... 2, 4, 6, 14, 17, 18, 19, 20, 26, 27, 32,

38, 50, 51, 52, 54, 55, 56, 57, 60, 62, 64, 65, 66, 67, 68, 69, 70, 71, 72, 75, 76, 77, 78, 80, 81, 91, 120, 121, 136, 140, 143, 144, 151, 152, 153, 160
Talk only............................2, 20, 51, 54, 60, 62
Tct.....................................81, 90, 153
Terminal............................37, 45, 46, 68, 112
Terminal emulation program.68
Termination.......................23, 72
Terminator.........................56, 97, 99, 100, 102
Timeout.............................73, 82, 86, 124, 125, 155, 156, 163
Timer.................................82, 154, 162
Trigger..............................20, 22, 81, 82, 104, 157

Unconfigure.......................81, 82, 135, 158
Unl.....................................19, 72, 74, 75, 81, 89, 91, 93, 94, 96, 101, 110, 126, 127, 128, 129, 130, 132, 147, 150, 153, 157, 159, 164
Unlisten.............................56, 81, 86, 90, 91, 94, 101, 126, 150, 159
Unt.....................................75, 81, 91, 93, 96, 127, 128, 129, 130, 132, 136, 143, 144, 145, 147, 160, 164
Untalk................................17, 56, 67, 81, 91, 151, 152, 160

Wait...................................78, 82, 86, 112, 113, 124, 125, 137, 147, 154, 155, 156, 162, 163
Wait srq.............................82, 163
Write..................................23, 97, 99, 100, 123, 137, 138, 159
Writing data.......................97, 98, 99, 100

Xoff...................................2, 38, 40, 41, 42, 83
Xon....................................38, 40, 41, 42, 83