

TT 402 A
REMOTE CONTROL
OPERATING MANUAL

NEUTRIK AG

CONTENTS

1	General.....	3
1.1	What level of software to write?	3
1.2	IEEE-488 or RS-232?	3
1.3	Origins of IEEE-488	4
1.4	Neutrik's approach to remote control	5
1.5	Setting the IEEE-488 address	6
1.5.1	IEEE-488 String Termination	7
1.6	Setting the RS-232 Baud Rate	8
1.6.1	RS-232 String Termination - Output	8
1.6.2	RS-232 String Termination - Input	8
1.6.3	RS-232 Cable Connections	8
1.6.4	RS-232 Handshaking	9
1.7	Sending strings to the TT402A/Oscillator System	9
1.8	Receiving strings from the TT402A/Oscillator System	10
1.9	Number to String Conversion	11
1.10	Sample Programme - Number to String Conversion and Frequency Sweep	11
2	Command syntax for the TT402A/Oscillator System.....	16
2.1	Abbreviations	16
2.2	Optional Words	16
2.3	Combining Commands	16
3	TT402A Measurement Section Commands.....	18
3.1	Function Commands (Standard)	18
3.2	Function Commands (Optional)	19
3.3	Input selection commands	20
3.4	Circuit impedance commands	20
3.5	Input termination commands	20
3.6	Rectifier selection commands	21
3.7	Scale selection commands	21
3.8	Filter selection commands	22
3.9	Range setting commands	22
3.10	Clock/Calendar setting commands	24
3.11	Reading queries	24
3.12	Out of range	25
3.13	Bus measurements are in absolute level	26
3.14	Measurements commanded must be read	26
3.15	Local/Remote commands	26
3.16	Status queries	29
3.17	Error Display commands and queries	30
3.18	Other commands and queries	31
4	Sending strings to the Oscillator.....	32
4.1	Oscillator level setting commands	32
4.2	Oscillator frequency setting commands	33
4.3	Oscillator output resistance (impedance) commands	33
4.4	Waveform commands	34
4.5	600R termination commands	34
4.6	Mute control commands	35
4.6.1	Mute control commands - TO402A and TO402	35
4.6.2	Mute control commands - TO404A and TO404	35
4.7	Balance/unbalance commands	35
4.8	Float/ground commands	35

4.9	Local/remote control commands	36
4.10	Status queries	36
4.11	Other commands and queries	39
5	CCITT 0.33 Operation.....	40
5.1	General	40
5.2	Operation - Sequence transmission only	40
5.3	Operation - Sequence reception only	40
5.4	Operation - Simultaneous transmission and reception ...	41
5.5	Format of 0.33 Results	41
5.5.1	Sequence 00	42
5.5.2	Sequence 01	43
5.5.3	Sequence 02	44
5.5.4	Sequence 03	45
5.5.5	Sequence 04	46
5.5.6	Sequence 05	47
5.6	Delay before Reply	47
6	IEEE-488-specific commands.....	48
6.1	Serial poll, parallel poll, LLO, RTL	48
6.2	Device Clear (DCL) and Selected Device Clear (SDC)	48
6.3	Interface Clear (IFC)	48
7	Error reporting.....	49
7.1	Error codes in numerical order	49
8	Errors and Omissions in this version.....	52
8.1	General	52
8.2	0.33	52
9	Compatibility Facilities.....	53
10	Considerations when Writing Measurement Programs.....	54
10.1	Accuracy and Precision of Measurements.	54
10.2	Accuracy and Settling Time	55
10.3	Out-of-Range Measurements	55
10.4	String Input from the TT402A	56
10.5	Speed of execution of test programmes	56
APPENDIX A	Programming Example.....	58
APPENDIX B	CCITT 0.33 Demonstration Programme.....	61

REMOTE CONTROL OF THE TT402A AND OSCILLATOR

1 General

This manual applies particularly to version 6.9 of the TT402A software, and versions 2.10 onwards of the oscillator software, regardless of the actual oscillator model.

The programming examples are mostly in BASIC, because we feel that this is the language most likely to be used for writing control programmes for the TT402A and oscillator. The principles should be clear enough that they can be transferred to other languages.

1.1 What level of software to write?

Neutrik has a software package available, called TS02, which enables automatic test programmes to be written in AMSL (Audio Measuring System Language), which is genuinely extremely easy to understand and use, because it is so English-like that the programmes document themselves. If you use TS02 you will not need to read this manual further. Your agent or dealer will be glad to demonstrate the TS02 package to you.

For ultimate flexibility, you can write programmes in any computer language you like, and communicate with the TT402A and its oscillator via the IEEE-488 bus or via RS-232. This manual is for you.

1.2 IEEE-488 or RS-232?

IEEE-488 allows up to 15 devices to be connected to one controller (nowadays, usually a PC). Because each device has a unique address, assigned by a switch on each device, communication can take place without confusion. In practical terms, this means you can have a system including a TT402A, a digital multimeter, a switch box, and perhaps other pieces of equipment, all controlled by the PC - and all from the one interface. The software is also easy, because all commands are channelled through one input routine and one output routine. The disadvantage is that the IEEE-488 controller is not cheap.

RS-232 only allows one pair of devices to be connected together, one of which is the controller. So, if you want to control several pieces of equipment, you must have several RS-232 interfaces in the controller. Your software must also have an input driver and an output driver for each device. But if you have an RS-232 port spare on your controller (not used by the mouse, modem, printer, etc.), and if you only want to control the TT402A, it's cheap. If you don't have an RS-232 port spare, you should be able to buy one quite cheaply - but check how your software will be able to drive it first.

Unfortunately, MS-DOS on a PC only has COM1: and COM2: safely standardised, which severely restricts the use of more than two serial ports.

If you decide to control via IEEE-488, the TT402A needs to be fitted with a TI33 CCITT 0.33 Interface Card, or a TI10 IEEE-488 Interface Card. The TI33 includes both IEEE-488 and RS-232 interfaces. Either may be selected by a switch on the back panel of the TT402A.

If you are using a PC as an IEEE-488 controller, the choice of IEEE-488 controller card for the PC depends on whether you wish to use the TS02 Software Package. If you do, the PC must be fitted with a "PC<>488" card from Capital Equipment Corporation (CEC). TS02 uses the programme in the EPROM provided (by CEC) with this card. If you are sure you don't wish to use TS02, there are other cards available. We would suggest, though, that your criterion for choice should be the ease with which your test programme will communicate with the IEEE-488 card. The type of software interface, and the ease of programming, vary very widely!

If you decide to control via RS-232, the TT402A must be fitted with the TI33 CCITT 0.33 Interface Card. The controller must be fitted with an RS-232 port. Check that there is no conflict between this and other RS-232 peripherals, such as mouse, modem or serial printer.

Candidly: getting IEEE-488 systems going using a PC and the CEC PC<>488 card works with considerably less problems than RS-232. Neutrik's advice, therefore, is to use IEEE-488.

Even though an RS-232 link is not actually a bus, there are many places in this manual which refer to communication "over the bus", etc. This applies equally to IEEE-488 or RS-232.

The commands sent, and strings returned, are identical via IEEE-488 or RS-232, except for the terminator character(s).

1.3 Origins of IEEE-488

IEEE-488 originated with Hewlett-Packard who called it HPIB (Hewlett Packard Interface Bus). As other companies began to use it the name GPIB (General Purpose Interface Bus) was coined. The designation IEEE-488 refers to the standard which followed commercial introduction to help ensure that users and manufacturers could define products and programs that would reliably work together. In Europe the standard became IEC-625, but unfortunately a different connector type and pin-out were defined; although the rest of the standard is virtually identical. Internationally the IEEE-488 standard is the clear winner.

The need for the bus arose from the requirement to build test systems combining several specialized instruments. The operation of these instruments could be made faster and more consistent if they were operated by one central controller. The controller could be programmed to perform a set of tests, analyse the results and communicate with the operator. Indeed in some cases instruments could pass results directly between each other.

The new bus had to meet the following basic requirements:

-1- High speed

-2- Several instruments can be simultaneously connected, others can be added without changing the controller hardware.

-3- Will work over reasonable distances as found in a typical test system.

-4- Reliable.

Existing bus systems as found inside computer systems meet 1, 2 and 4, but would only work over very short distances.

Parallel ports as - for example - the widely used Centronics Parallel Interface used with printers, meet 1, 3 and 4, but only deal with one device at a time and start to be less reliable over distances greater than 1m.

Serial ports, again commonly used to connect printers and terminals to computers meet 3 and 4, but are slow and only handle one device at a time.

The bus defined in IEEE-488 meets all the requirements. At first it was difficult to design an interface, but before long at least TEXAS INSTRUMENTS and MOTOROLA brought out custom designed processors to provide the required functions. These ICs are still expensive, but the logic is easy to design.

Unfortunately the software provided with some controllers and instruments to handle the bus protocols is not always well designed.

1.4 Neutrik's approach to remote control

The majority of functions of the TT402A and oscillator which can be accessed from the front panel are remotely controllable. The exceptions are: mains on/off switch, MONITOR level control, FINE FREQUENCY and FINE LEVEL controls. All measurements which can be made by the TT402A are available over the bus.

Commands are in plain language (English with some French and German) and readings are returned in a form which is readily comprehensible to humans and computers. The result is that programming an automatic test sequence is easy, and the listing of the programme will be virtually self-documenting. All inputs and outputs to the TT402A system are in the form of ASCII strings.

Each time a MEASUREMENT is commanded, a new measurement is performed. This ensures that results are current rather than relating to the recent past.

The analogue measurements are to a resolution of 10 bits (1 part in 1024) of the meter scale range. The true meter ballistics are included rather than being taken from before the meter driver. Therefore the result is exactly what an operator would read at that instant in time from the meter. Results are expressed in

engineering notation complete with units (always fundamental units). Examples are:

```
+4.76 dBu
-75.4 dB
0.775 V
2.45E-3 V (equals 2.45 mV)
0.012 %
1.234E3 Hz (equals 1.234 kHz)
```

Most computer languages will accept ASCII input in this form, which can be converted by a language command into internal number representation. The ASCII form can usually be printed directly to the screen.

For example in Microsoft and many other basics, the following will do the conversion::

```
100 result$ = "-75.4 dB"    result$ would normally be a
                             string variable sent from the
                             TT402A.
```

```
150 dB = VAL(result$)      The variable dB would contain a
                             floating point value equal to
                             -75.4.
```

1.5 Setting the IEEE-488 address

Selection of IEEE-488 is performed by setting the "X" pole (third from the left), of the DIL-switch on the TT402A rear panel, to "1" (up).

Setting the IEEE-488 address is performed by setting the rightmost five poles of the DIL-switch. Addresses 0 to 30 are valid, address 31 is not (this fact is "built in" to the IEEE-488 standard and is therefore common to all instruments). Note that address 0 is commonly used by controllers as their talk and listen address, therefore it is a good idea to avoid setting the instrument to address 0. Only single primary addresses are recognised; the TT402A has no secondary addressing or dual primary addressing capability.

These switches are only read on power-up, DCL, SDC or after the message "RESET", therefore if you wish to change the address, the instrument must be powered up anew or cleared or RESET after the switches have been changed.

Selection of an IEEE address is as follows:

SWITCH POLE	5	4	3	2	1
BINARY VALUE	16	8	4	2	1

Example:

To set bus address 19. Using the BINARY VALUE from the table above, $1+2+16 = 19$. Therefore switches 1, 2 and 5 are up; 3 and 4 are down.

1.5.1 IEEE-488 String Termination

Each string must be terminated to indicate to the TT402A that the transfer is complete. One of the following may be used:

(i) EOI sent true with the last character, which can therefore be any character. This is the normal method of terminating strings used by modern controllers, and is a hardware line in the IEEE bus.

(ii) Carriage Return <CR> (hex 0D) or Line Feed <LF> (hex 0A) or ETX (hex 03). These extra termination characters can be used even if EOI is also active. If your controller sends more than one terminator (e.g. <CR> followed by <LF>), set the TT402A to the last one sent.

EOI is always sent by the TT402A with the last character, regardless of the settings of the terminator switches. If CR, LF or ETX is set, EOI is true with that terminator; otherwise EOI is true with the last data character.

EOI is always recognised by the TT402A as signifying the last character, regardless of the settings of the terminator switches. If CR, LF or ETX is set, you should be sure that that is the last character sent by the controller, because the TT402A would accept EOI or the chosen terminator, whichever comes first, as signifying end of string.

We recommend that EOI be treated as the normal terminator, i.e. if your controller sends it, then set the TT402A to EOI regardless of any other terminators sent. The notable exception is Neutrik's TS02 Software Package, which requires the terminator to be set to LF.

The required terminator is set by the leftmost two poles of the DIL switch. Note, as mentioned in section 1.5 that the switches are only read on power-up, DCL, SDC or "RESET".

Switch settings are printed next to the switch on the interface or rear panel as appropriate. For reference these are as follows:

<u>TERMINATOR</u>	<u>EOI (only)</u>	<u>CR</u>	<u>LF</u>	<u>ETX</u>
SWITCH 7	DOWN	UP	DOWN	UP
SWITCH 8	DOWN	DOWN	UP	UP

1.6 Setting the RS-232 Baud Rate

Selection of RS-232 is performed by setting the "X" pole (third from the left), of the DIL-switch on the TT402A rear panel, to "0" (down).

The baud rate is selected by setting the rightmost four poles of the DIL-switch according to the table below. This table is reproduced on the back panel of most TT402As (TT402s and early TT402As do not have this table).

RS-232 BAUD RATE	SW			
	5	6	7	8
38400	0	0	0	0
19200	0	0	0	1
9600	0	0	1	0
4800	0	0	1	1
2400	0	1	0	0
1200	0	1	0	1
600	0	1	1	0
300	0	1	1	1
150	1	0	0	0
110	1	0	0	1
75	1	0	1	0

Note that the back panel DIL-switches are only read on power-up, after the message "RESET", or after an IEEE-488 DCL or SDC. Therefore, if you wish to change the baud rate, the instrument must be powered up anew or cleared or RESET after the switches have been changed.

1.6.1 RS-232 String Termination - Output

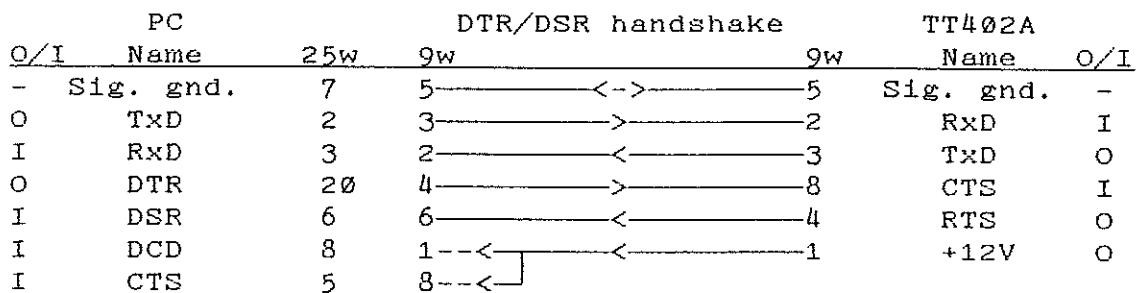
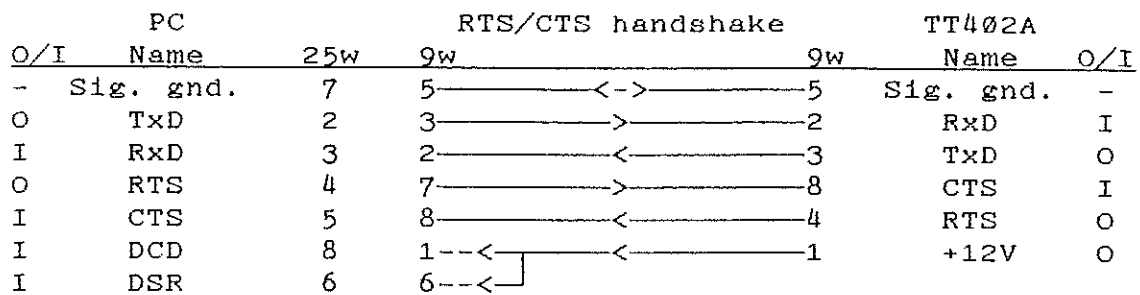
The TT402A sends CR (Carriage Return) followed by LF (Line Feed) when outputting via RS-232. This cannot be changed by switches.

1.6.2 RS-232 String Termination - Input

The TT402A requires LF (Line Feed) as terminator on incoming strings. It doesn't matter if there is a CR (Carriage Return) before the LF. This cannot be changed by switches.

1.6.3 RS-232 Cable Connections

PC-compatible computers are not exactly compatible in the way they work. With RS-232, the necessary connection of handshaking lines is not always predictable. The four lines which are predictable are Signal Ground, TxD, RxD and DCD, which should be connected as below. Some machines require CTS to be permanently true (parallel it with DCD, fed from the TT402A), and handshake via DTR/DSR. Others handshake via CTS/RTS and require DSR to be permanently true (parallel it with DCD, fed from the TT402A). Only experiment will give a reliable answer. See the next section.



1.6.4 RS-232 Handshaking

The TT402A uses hardware handshaking to slow down RS-232 communication, if necessary, in both directions. The data input to the TT402A is on pin 2; the handshake line associated with it (an output) is on pin 4. The data output from the TT402A is on pin 3; the handshake line associated with it (an input) is on pin 8. Positive voltages with respect to signal ground permit transmission; negative voltages stop transmission. If the handshake input to the TT402A is switched negative, the TT402A will not send any characters after the current character (no overrun). It cannot cope with any overrun from the controller.

Note that hardware handshaking to slow down transmission, though very common practice, is not the use of the handshake lines as specified in RS-232; they were intended to be modem control lines. If you experience loss of characters (typically ERROR 1 after the second and subsequent strings sent to the TT402A), we suggest that you investigate the handshaking protocol used in your controller, to see if it can be made to conform with the protocol described in this section.

1.7 Sending strings to the TT402A/Oscillator System

All instructions for the TT402A consist of ASCII strings. Strings are sent from the controller to the TT402A system using commands appropriate to the chosen controller. The required strings are described in sections 3 and 4. The same strings are sent whether IEEE-488 or RS-232 is in use; the only difference is the terminator, which is described in sections 1.5.1 (IEEE-488) and 1.6.2 (RS-232) above. Strings to the oscillator are passed through the TT402A. These are specifically described in section 4.

1.8 Receiving strings from the TT402A/Oscillator System

All strings returned from the TT402A consist of ASCII strings. Strings are input from the TT402A system to the controller using commands appropriate to the chosen controller. The same strings are sent whether IEEE-488 or RS-232 is in use; the only difference is the terminator, which is described in sections 1.5.1 (IEEE-488) and 1.6.1 (RS-232) above. Strings from the oscillator are passed through the TT402A.

The reading is commanded by sending a string to the TT402A which requires a reply to be sent. Reading queries (commands which cause a reply) are detailed in section 3.11.

If remote control is via IEEE-488: The reply string will be terminated with the EOI line true. Depending on the state of the leftmost two poles of DIL-switch mentioned in section 1.5.1 above, a character may be added by the TT402A to the end of the string. This character will be carriage return (hex 0D), line feed (hex 0A) or ETX (hex 03).

If remote control is via RS-232: The reply string will be terminated by carriage return (hex 0D) and line feed (hex 0A).

Once a reading has been commanded, a string will always be sent, even if the measured result is out of range in some way. In that case an out-of-range message will be sent instead of a reading.

There will be a delay between the command and returning the reading. This delay may be from a few tens of milliseconds to 500 ms in present versions of software. Analogue meter readings take a few tens of milliseconds; frequency readings take between 220 ms and 500 ms. When a CCITT sequence is received, the delay will be much longer; see section 5.6 for details.

Later versions of software with other facilities may require longer delays for some of the newer facilities.

It is important to remember that the TT402A cannot accept another string input until any pending or impending string output operation has been completed. Failure to observe this sequence can cause the system to "hang" indefinitely.

When receiving strings via RS-232, or via IEEE-488 with the DIL switches set to <CR>, <LF> or <ETX> as terminator, beware that controllers often include this character as part of the string on input. This can lead to some confusing results. For instance, the obvious way in BASIC of checking for an over range reading would be:

```
IF result$ = "Over range" THEN -
```

If your controller includes the terminator with the string then this test would fail! We suggest you use one of the following:

```
IF INSTR(result$, "Over") <> 0 THEN - - - or
IF LEFT$(result$, 4) = "Over" THEN - - -
```

The ways of getting strings into the controller will depend on your chosen controller hardware and software. If you are programming in BASIC and you need to use an INPUT-like statement, we recommend that you use LINE INPUT (INPUT LINE in some dialects) rather than INPUT. Some INPUT statements will truncate input after some punctuation characters.

1.9 Number to String Conversion

Most computers or controllers store numbers in an internal number format. Thus they must be converted to ASCII strings before being sent to the TT402A. Fortunately, this is a common requirement and most computer languages contain a conversion function to do this job for us.

For example, if you were writing a programme to sweep and measure in 1/2 octave steps, one way to obtain the correct frequencies would be to calculate them.

Steps of 1/2 octave implies a frequency change of $\sqrt{2}$. If the sweep is to start at 20 Hz and go to 20 kHz, then the programme might be as follows if written in Microsoft Basic.

1.10 Sample Programme - Number to String Conversion and Frequency Sweep

Please note that this example does not deal with settling time and its effect on the accuracy of measurements, nor does it deal with the effect of errors or out-of-range measurements on the plotting subroutine.

10 DEF SEG = &HD000	Define address for EPROM routines on CEC PC<>488 card.
20 INIT=0	Address offset within PC's address space of PC<>488 initialisation routine.
30 ADR%=0	Talk/listen bus address of PC<>488 card.
40 LEVEL%=0	Define PC<>488 as controller.
50 CALL INIT(ADR%, LEVEL%)	Perform the initialisation.
60 TRANSMIT%=3	Address offset within PC's address space of PC<>488 transmit routine.

70 RECEIVE%=6	Address offset within PC's address space of PC<>488 transmit routine.
80 TT402A.ADR%=1	Talk/listen bus address of TT402A.
90 MAXLEN=80:'Max length of received string.	Preset string to a length longer than anything which the TT402A sends.
100 GOTO 280	Jump past the subroutines to the main programme.
110 REM *****	
120 REM Transmit TX\$ over bus to address TT402A.ADR%.	
130 OP\$="MTA UNL LISTEN"+STR\$(TT402A.ADR%)+" DATA '"+TX\$+"'" END"	Build up OP\$ (see the PC<>488 Programming and Reference Manual for details).
140 CALL TRANSMIT%(OP\$,STATUS%)	Transmit OP\$ over the IEEE-488 bus.
150 IF STATUS%<>0 THEN PRINT "Error code";STATUS%	Look for errors reported by PC<>488 routines, and display the value if not OK.
160 RETURN	
170 REM *****	
180 REM Receive RX\$ over bus from address TT402A.ADR%.	
190 RX\$=SPACE\$(MAXLEN)	Preset the string length longer than anything the TT402A returns.
200 OP\$="MLA TALK"+STR\$(TT402A.ADR%)	Build up OP\$ (see the PC<>488 Programming and Reference Manual for details). Sets the TT402A to talk.
210 CALL TRANSMIT%(OP\$,STATUS%)	Transmit OP\$ over the IEEE-488 bus.
220 IF STATUS%<>0 THEN PRINT "Turnround error code";STATUS%	Check for errors and print them if not OK.
230 CALL RECEIVE%(RX\$,RXLNG%,STATUS%)	Get RX\$ from IEEE-488 bus.
240 IF STATUS%<> 0 THEN PRINT "Receive error code";STATUS%	Check for errors and print if not OK.

250	RX\$=LEFT\$(RX\$,RXLNG%)	Shorten RX\$ to the length actually sent.
260	RETURN	
270	REM *****	
280	REM Main programme starts here.	
290	TX\$="Function Level; ScaleCommand string to TT402A, dB norm; Osc level +4 dBu"	
300	GOSUB 120	Send command string to TT402A.
310	FRSTART=20	Start frequency 20 Hz.
320	FREND=20000	End frequency 20 kHz.
330	FRNEXT=FRSTART	Initialise "next frequency" to start frequency.
340	FREQ\$=STR\$(FRNEXT)	Take a string representation of the number. The transmit subroutine will only accept string variables.
350	TX\$="Osc freq "+FREQ\$	Build up the command string to the TT402A. The first time, this string will be "Osc freq 20".
360	GOSUB 120	Send TX\$ over the IEEE-488 bus.
370	TX\$="Measure"	
380	GOSUB 120	Then send "Measure" over the IEEE-488 bus to command a measurement.
390	GOSUB 180	Return the string RX\$.
400	PRINT "At";FREQ\$;" Hz the measurement is ";RX\$	Print a result on the screen.
410	X=FRNEXT	Set up the X co-ordinate for a plotting subroutine.
420	Y=VAL(RX\$)	Similarly the Y co-ordinate.
430	GOSUB 1000	Call the plotting subroutine.
440	FRNEXT=FRNEXT*SQR(2)	Multiply the frequency by the square root of 2, to give the next frequency.

```

450 IF FRNEXT<=FREND THEN 340 Check if we've gone over the
                                end frequency. If not, loop
                                back to 340.

460 END                                End of main programme, when
                                frequency has gone past the
                                end frequency.

1000 REM Plotting routine
                                would be here...

1010 RETURN

```

Note that this programme does not deal with settling time considerations (there would normally be a delay between lines 360 and 370), nor with the effects of out-of-range readings. However it should serve to illustrate the requirements for getting communication going with the TT402A over the IEEE-488 bus. It also demonstrates the recommended structure for programmes at a low level, to give general purpose subroutines for transmitting and receiving strings. This programme runs under GWBASIC on a PC with a CEC PC<>488 card.

The same programme converted to RS-232 operation is as follows. This also runs under GWBASIC on a PC, using COM1: at 9600 baud. For a different RS-232 port or baud rate, change line 50. Note that GWBASIC only supports COM1: and COM2:.

```

50 OPEN "COM1:9600,N,8,1" AS #1
100 GOTO 280
110 REM *****
120 REM Transmit TX$ via RS-232 to TT402A.
140 PRINT #1, TX$+CHR$(10);
160 RETURN
170 REM *****
180 REM Receive RX$ via RS-232 from TT402A.
230 INPUT #1, RX$
260 RETURN
270 REM *****
280 REM Main programme starts here.
290 TX$="Function Level; Scale dB norm; Osc level +4 dBu"
300 GOSUB 120
310 FRSTART=20
320 FREND=20000
330 FRNEXT=FRSTART
340 FREQ$=STR$(FRNEXT)
350 TX$="Osc freq "+FREQ$
360 GOSUB 120
370 TX$="Measure"
380 GOSUB 120
390 GOSUB 180
400 PRINT "At";FREQ$;" Hz the measurement is ";RX$
410 X=FRNEXT
420 Y=VAL(RX$)
430 GOSUB 1000
440 FRNEXT=FRNEXT*SQR(2)
450 IF FRNEXT<=FREND THEN 340
460 END

```

```
1000 REM Plotting routine would be here...  
1010 RETURN
```

2 Command syntax for the TT402A/Oscillator System

2.1 Abbreviations

To give most flexibility, many words which the TT402A can understand may be abbreviated. Some may be omitted altogether. Words which may be abbreviated are shown with square brackets around the optional section. For example, LEV[EL] means that LEVEL, LEVE or LEV are equally valid. Abbreviations must be progressive from the right, therefore in the above example, LEVL is not valid. In practice, it is unlikely that anything other than the complete word or the shortest abbreviation will be used.

2.2 Optional Words

Words which may be omitted altogether are shown enclosed in parentheses. For example, SCA[LE] (DB) NORM[AL] include the following possibilities:

SCALE DB NORMAL
SCALE NORMAL
SCA DB NORM
SCA NORM

Certain words can be either abbreviated or omitted. These are shown with both square brackets and parentheses, for example (FUNC[TION]) LEV[EL] includes the following possibilities:

FUNCTION LEVEL
FUNC LEV
LEVEL
LEV

In general, commands include the legends printed on the front panel near the appropriate controls. This, plus the use of plain language, gives you the best chance of being able to write a test programme without needing to consult these instructions.

Commands can be entered in either UPPER or lower case characters, or any combination of the two.

2.3 Combining Commands

Commands to the TT402A Measuring System and oscillator can be put together into sets of up to 150 characters. Commands are combined by inserting a semicolon (;) between each command. Commands may be sent to the test set in any order, but will be executed left to right. Remember that, just as in manual operation, the test set will recall the last used filters, rectifier and scale when any function is selected. For example:

SCALE DB NORMAL; FUNCTION LEVEL; OSC LEVEL +4DBU

This entire string of commands can be sent together by the controller. But beware; in the above example, the dB normal scale will be set into the function the TT402A was in before the above set of commands. Then the level function will be entered,

and will recall the scale last used in the level function (which might not be dB normal!).

3 TT402A Measurement Section Commands

For oscillator commands see section 4.

3.1 Function Commands (Standard)

All function commands can be preceded by the optional keyword FUNCTION, which can in turn be abbreviated as far as FUNC. So:

(FUNC[TION]) followed by:

LEV[EL]	select the LEVEL function
NIV[EAU]	
PEG[EL]	
READ OSC[ILLATOR]	select the READ OSC function
ROSC	
THD[+NOISE]	select the THD + N function and
KLIRR[FAKTOR]	SINE on the oscillator
XTALK	select the XTALK function and
X-TALK	SINE on the oscillator
CROSS-TALK	
DIAPH[ONIE]	
CROSST[ALK]	
LEV[EL] DIFF[ERENCE]	select the PHASE and LEVEL
NIV[EAU] DIFF[ERENCE]	DIFFERENCE function and SINE on
PEG[EL] DIFF[ERENZ]	the oscillator
IMD (SMPTE)	select IMD function and test
(DIN)	tones on oscillator. If no
(CCIF)	standard is specified following
	IMD, the standard set on the DIL
	switches on the IMD card will be
	used.
NOISE (CCIR) 468[-3]	(WEIGHTED)
468[-2]	(WTD)
ARM	(UNWEIGHTED)
	(UNWTD)

The order of the arguments for weighting and standards definition can be exchanged, so e.g. NOISE WEIGHTED 468-3 works just as well as NOISE 468-3 WEIGHTED. The weighting argument is optional; if not supplied the instrument will recall the last used weighting setting, which defaults (on power-up, DCL, SDC or "RESET") to WEIGHTED for both 468 and ARM functions. One of the arguments 468[-3], 468[-2] and ARM must be supplied, or error 2 will result; and may optionally be preceded by CCIR.

Note that the CCIR 468 ARM function may be disabled in the EPROM, if so ordered. If it is commanded while disabled, ERROR 25 will result.

3.2 Function Commands (Optional)

These functions are only enabled if the TT402A contains the appropriate optional hardware.

W[OW] AND F[LUTTER]	(W[EIGHTED])	(IEC)	(3K15[HZ])
&	(WT[D])	(DIN)	(3150[HZ])
W&F	(UNW[EIGHTED])	(CCIR)	(3.15[KHZ])
DR[IFT]	(UNWT[D])	(BS)	(3K[HZ])
PLEUR[AGE]		(NAB)	(3K00[HZ])
			(3000[HZ])

As in the NOISE functions, the order of the arguments for centre frequency, weighting and standards definition is optional, and any (or all in this case) may be omitted. If the weighting argument is omitted, the instrument will recall the last used setting, which is initialised (on power-up, DCL, SDC or "RESET") to WEIGHTED. If the arguments for standards definition and/or centre frequency are omitted, the instrument will use the respective DIL-switch settings from the card.

RUM[BLE]	(W[EIGHTED])	(IEC)
	(WT[D])	(BS)
	(UNW[EIGHTED])	
	(UNWT[D])	

As before, the weighting and standards arguments are optional and may be in either order. If the weighting argument is absent, the previously used weighting characteristic is recalled. If the standards argument is absent, the DIL-switch setting from the card is used.

FIM

FIM takes no arguments. This rarely used measurement is not normally available from the front panel *, and is indicated by the LED immediately to the right of the RUMBLE LED. After it has been commanded once from the bus, it will be available manually until the next power-up, DCL, SDC or "RESET".

* It can be made available from the front panel by a simple change to the EPROMs. Contact your agent for details.

On entering a previously used function, the previously selected filters, rectifier and scale will automatically be selected, just as in manual operation.

K2	Set to the appropriate K _n
K3	function, and set the oscillator
K4	to sine wave.
K5	
PHASE	Set to the phase (only)
	function with analogue centre-
	zero display.

Examples:

```
FUNCTION THD
DIAPHONIE
KLIRR
READ OSC
WOW AND FLUTTER IEC WEIGHTED 3.15KHZ
W&F DIN           Sets to previous weighting
                   characteristics, and centre frequency
                   from card.
W&F CCIR 3KHZ     Sets to previous weighting
                   characteristic.
W&F NAB WTD       Sets to centre frequency from card.
W&F               Sets to previous weighting
                   characteristic; centre frequency and
                   standard from card.
FUNCTION RUMBLE BS WEIGHTED
RUMBLE WEIGHTED IEC
RUM               Sets to the previous weighting
                   characteristic and the standard set on
                   the card.
```

3.3 Input selection commands

```
INP[UT]          A           select input A
                  B           select input B

INP[UT]           (no argument) changes to the other input
```

None of these commands can be accepted while the instrument is in the READ OSC or LEVEL DIFFERENCE functions. This is consistent with manual operation of the instrument. If this operation is attempted ERROR 4 will be reported.

3.4 Circuit impedance commands

CIRC[UIT]	IMP[EDANCE]	75[R]	The values may be
CCT	RES[ISTANCE]	140[R]	different from
		600[R]	the standard
			values shown.

Examples:

```
CCT RES 75
CIRCUIT IMPEDANCE 600R
```

3.5 Input termination commands

```
TERM[INATED]     select terminations on both inputs
THRO[UGH]         deselect terminations on both inputs
THRU
```

3.6 Rectifier selection commands

RECT[IFIER]	RMS	select RMS rectifier
	AVG AVER[AGE]	select AVG rectifier
	PPM	select PPM rectifier

Example:

```
RECT RMS
RECTIFIER PPM
```

Just as in manual operation, it is not possible to select a rectifier which conflicts with a pre-determined specification. Hence ERROR 4 will be reported if an attempt is made to select a rectifier whilst the function selected is NOISE, LEVEL DIFFERENCE, IMD, WOW AND FLUTTER, RUMBLE, FIM, K_{Σ} , K_{Σ} , K_{+} , K_{Σ} or PHASE.

3.7 Scale selection commands

SCA[LE] (DB) NORM[AL]	select DB NORM scale
EXP[ANDED]	select DB EXPANDED scale
V[OLTS] MILLI[VOLTS] MV MICRO[VOLTS] UV	select VOLTS scale
PER CENT (DIST[ORTION]) PERC[ENT] (DIST[ORTION]) % (DIST[ORTION])	select PER CENT scale

VOLTS etc are only allowed in LEVEL, XTALK, READ OSC and NOISE functions.

PER CENT and % etc, are only allowed in THD, IMD, WOW AND FLUTTER, FIM and K_n functions.

DB EXPANDED is only allowed in LEVEL and READ OSC functions.

DB NORMAL is NOT allowed in WOW AND FLUTTER and FIM functions.

These are consistent with manual operation. ERROR 4 will be indicated if scales are selected which are inappropriate to the selected function.

3.8 Filter selection commands

FILT[ERS]	HIGHP[ASS] 400HZ HP	select 400 Hz high-pass filter
	BP BANDP[ASS] 22HZ-22KHZ	select 22-22k bandpass filter
	LP LOWP[ASS] 30KHZ	select 30 KHz low-pass filter
	SP[ARE] A OPT[ION] B	select OPTION A or OPTION B filter, as appropriate. SP[ARE] and OPT[ION] are synonyms.
	OFF OUT CLEAR AUS	deselect all filters

Filter combinations which are permissible in manual operation can be selected by separating names with comma (,), plus (+), or slash (/).

Examples are:

```
FILTER LOWPASS
FILTER BP, 400HZ
FILTERS 22HZ-22KHZ + OPTION A
FILT OPT A/OPT B
```

Selected filters will always replace any previously selected. Note that this is NOT consistent with manual operation. They will also be stored and recalled next time the particular function is entered.

3.9 Range setting commands

Ranges cannot be set over the bus until the instrument has been sent a LOCAL LOCKOUT (LLO) or REMOTE RANGE command to disable the front panel range switches. Any RANGE command received, if they have not been disabled, will cause ERROR 22.

Ranges can be set in any units, regardless of the SCALE selection. E.g. the 1 volt range could be selected, then readings obtained in dBu.

Alternatively, the TT402A can be set to auto-range by sending the following command:

```
RANG[E] AUTO
```

Manual control of the TT402A's ranges can be restored by the command:

```
RANG[E] MAN[UAL]
```

or, providing the instrument is in the LOCAL LOCKOUT or REMOTE RANGES status, by setting the unit to any legitimate range.

For ease of manual operation, the TT402A has 3 large controls. One affects the DISTORTION range, the other two set the range for all LEVEL related measurements.

To set a DISTORTION range, precede RANGE commands by one of the following function commands:

THD, K2, K3, K4, K5, IMD, WOW AND FLUTTER, FIM

To set a LEVEL range, precede RANGE commands by any function command not listed above. However, range commands are not acceptable in the LEVEL DIFFERENCE and PHASE functions. If this is attempted, ERROR 4 will be produced.

RANG[E] <number>	DB
	DBU
	DBM
	DECIB[ELS]
	V[OLTS]
	MILLI[VOLTS]
	MV
	MICRO[VOLTS]
	UV
	PERC[ENT]
	%

Examples:

```
RANGE +20 DB
RANGE 30 UV
RANGE 30 MICRO
RANGE 1000 MV
```

The space between the number and the unit is optional.

Units of VOLTS, MILLIVOLTS, MICROVOLTS, DBU and DBM are only acceptable in LEVEL, XTALK, READ OSC, NOISE and RUMBLE functions regardless of scale.

Units of PERCENT are only acceptable in THD, K₂, K₃, K₄, K₅, IMD, WOW AND FLUTTER and FIM functions.

Units of DECIBELS are acceptable in all functions (except LEVEL DIFFERENCE and PHASE), regardless of the currently selected scale.

The number must be sent in ASCII format. It may be in signed or unsigned, integer, fixed point, or floating point form. Acceptable values are those corresponding to the values of the range switch or switches in use for the particular function. For example in LEVEL function, integers in the range +45 to -95 are

acceptable, though they may be input in fixed- or floating-point notation, when commanding a range in dB.

Examples:

17 are all equivalent and acceptable.
+1.7E+01
17.00000

In LEVEL, X-TALK and READ OSC functions, acceptable values work out to 100, 30, 10, 0.3, 0.1, 0.03, ...30E-6 when taken in combination with the units (V, mV, uV).

In THD+N, K_n and IMD functions and dB scale, acceptable values are from 0 to -80 in steps of 10. In % scales, acceptable values are 100, 30, 10, ... 0.01.

Values which are not acceptable to the instrument will cause ERROR 3.

In WOW and FLUTTER and FIM functions, acceptable values work out to 10, 3, 1, 0.3, 0.1 and must be followed by units PER CENT (or abbreviations or synonyms thereof).

3.10 Clock/Calendar setting commands

TIME hh:mm:ss	Sets the real time clock. hh:mm:ss is in 24-hour format. Leading zeros are not required.
DATE dd/mm/yy	Sets the calendar. dd/mm/yy may include the year between 88 and 99, or between 1988 and 2234. Leading zeros are not required.

The TT402A maintains a clock and calendar. This is not a battery backed clock/calendar as is found in most computers. The system consists of a fixed divider from the main timebase, interrupting the CPU at regular intervals with a precision defined by the timebase specification (± 20 ppm over temperature). Therefore, it needs to be set whenever the TT402A is powered up. Commands in the RESET family do not affect the clock/calendar, however. Leap years are automatically catered for (up to 2100, in theory!). Daylight savings compensation is not included.

3.11 Reading queries

MEAS[URE] SEND	Returns the reading on the analogue meter in units according to the scale currently selected. Acceptable in all functions.
FREQ?	Returns a new frequency reading in units of Hertz. Not acceptable in WOW AND FLUTTER or FIM functions.
PHASE?	Returns a reading of phase in integer units of degrees, e.g. "-9 deg". Only acceptable in LEVEL DIFFERENCE or PHASE functions.

DRIFT?	Returns a drift (centre frequency error) reading. The format is "+n.nn % drift", or "+nn.n % drift", i.e. 3 significant figures or 2 decimal places, whichever is the less accurate. Only acceptable in WOW AND FLUTTER and FIM functions.
TIME?	Returns the time in 24-hour format in the form hh:mm:ss. The time should have been set by means of the TIME command, or else the time returned is simply the elapsed time since power-up. Leading zeros are supplied.
DATE?	If the date has been set since power-up, returns the date in the form dd/mm/yyyy. Leading zeros are supplied. If the date has not been set, returns "Date not defined."

Examples of the strings returned were given in section 1.4.

3.12 Out of range

In response to the MEASURE or SEND queries:

If in the THD+N or IMD function and the level is too low, the string "Level low" will be returned.

If in the THD+N or XTALK function and the frequency is too low, the string "Freq low" will be returned. Similarly if the frequency is too high, the string "Freq high" will be returned.

If the reading is over 102.2% of the linear scale (e.g. over 102.2 mV on the 100 mV range), over approx. +13 dB on the dB NORMAL scale, or over approx +1.78 dB on the dB EXPANDED scale, the string "Over range" will be returned.

If the reading is under -12 dB on the dB NORMAL scale or -1.5 dB on the dB EXPANDED scale, the string "Under range" will be returned.

On volts scales, there is no underrange condition, but accuracy decreases for results below 10% of meter deflection.

If in the PHASE & LEVEL DIFFERENCE function and the reading is too low, the string "Under range" will be returned. Similarly, if the reading is too high, "Over range" will be returned.

If in the PHASE (only) function, the queries MEASURE and SEND are synonymous with the PHASE? query (see below.)

In response to the FREQ? query:

If the alphanumeric display reads "FREQ LOW", the string "LOW" will be sent; if the display reads "FREQ HIGH", the string "HIGH" will be sent.

In response to the PHASE? query:

If the alphanumeric display reads "LEVEL A LO", the string "Level A low" will be sent; if the display reads "LEVEL B LO", the string "Level B low" will be sent; if the display reads "LEVELS LOW", the string "Levels low" will be sent. If the display reads "M.S.R. ERROR", the string "M.S.R. error" will be sent.

In response to the DATE? query, if the calendar has not been set:

The string "Date not defined." will be sent.

We recommend that you read and understand section 1.8 before writing a section of programme which detects these strings!

3.13 Bus measurements are in absolute level

So long as the measurement is within range, level and level-like results from the TT402A will always be absolute. This may not correspond to the meter reading.

EXAMPLE: If the level we are measuring is +4 dBm, and the range is set to +4 dBm, then the front panel meter will read 0 dB (no error with reference to the setting of +4 dBm). However, bus readout will be "+4.0 dBm" when in dB NORM scale, or "+4.00 dBm" when in the dB EXP scale. The reason for this is that the meter was specifically designed for ease of calibrating external equipment to a preset reference, while in systems use this type of indication is in the hands of the programmer and can easily be duplicated in software.

Note that the dB EXP scale is most appropriate where very high precision is required. For most purposes dB NORM will be adequate.

3.14 Measurements commanded must be read

We repeat the warning that any string commanded must be read. No other command can be accepted until this has been completed. Failure to read back a string (or, indeed, trying to read a string when none has been commanded), will result in the system "hanging", or the controller issuing a timeout error.

3.15 Local/Remote commands

LOC[AL] LOCK[OUT]
LLO
REM[OTE]

Inhibit response to all front panel controls in both the TT402A AND the oscillator. The LOCAL LOCKOUT LED on the front panel will be constantly illuminated.

(RET[URN] TO) LOC[AL]
(GO TO) LOC[AL]
RTL
GTL

Restore front panel control to both the TT402A AND oscillator. The LOCAL LOCKOUT LED will be extinguished.

REM[OTE] RA[NGES] Inhibit response to the three (rotary) range switches on the front panel of the TT402A ONLY.

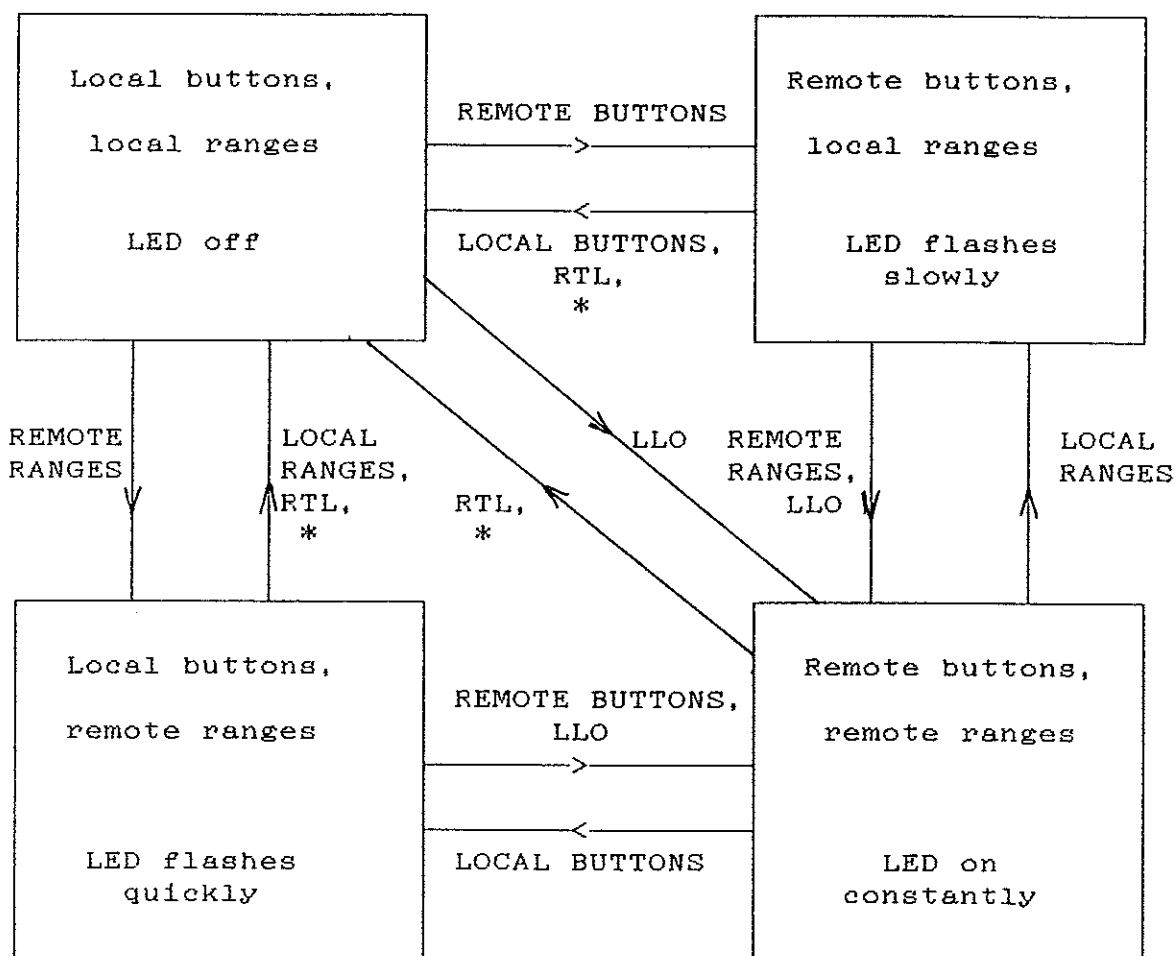
REM[OTE] BU[TTONS] Inhibit response to the pushbuttons on the front panel of the TT402A ONLY. NB. The LOCAL pushbutton remains active.

LOC[AL] RA[NGES] Restore response to the three (rotary) range switches on the front panel of the TT402A ONLY.

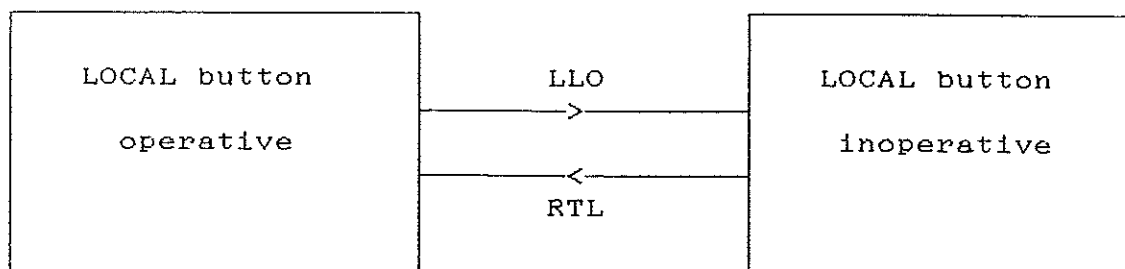
LOC[AL] BU[TTONS] Restore response to the pushbuttons on the front panel of the TT402A ONLY.

Note that local/remote switching in the TT402A is NOT in accordance with IEEE-488 and IEC-625. We feel the arrangement we have chosen is more versatile and useful.

The following diagram shows the four states of the test set, the way they are indicated by the LOCAL LOCKOUT LED, and the transitions between states caused by the various messages. Note (i) that LOCAL LOCKOUT and LLO are synonymous, as are RETURN TO LOCAL, GO TO LOCAL, GTL and RTL; (ii) Only the messages LLO and RTL (and their synonyms) cause the same message to be passed to the oscillator. Therefore, for instance, after power-up, the messages "REMOTE RANGES" and "REMOTE BUTTONS" will cause the test set to be only remotely operable, but will NOT lockout the oscillator. The same effect can be obtained with "LLO; OSC RTL".



* = pushing the LOCAL button, subject to its operative status, as outlined below:



If the command "REMOTE RANGES; REMOTE BUTTONS" is used to lockout the front panel, manual control can be regained by pressing the LOCAL button. This is not the case if the command "LLO" is sent. However, if the command "OSC LLO" is used to lockout the oscillator, the LOCAL button can be used to regain control of the oscillator (providing the button is operative).

3.16 Status queries

STAT[US]? Returns the complete status in the form of multiple strings. The first string is a number equal to the number of status lines which will follow. For example:

```
9
Function Level
Input A
Through
Circuit impedance 600 R
Filters Off
Rectifier RMS
Scale dB normal
Range +0 dBm
Return to Local
```

or:

```
4
Function Level difference
Terminated
Circuit impedance 75 R
Local Lockout
```

In each case, the strings returned are themselves legitimate commands to send back to the test set. In this way, the complete status of the instrument can be saved on a convenient storage medium, eg. floppy disc and later used, with the minimum of processing, to restore the unit to the same settings.

The simplest way to read the complete status into the controller is as follows:

```
1000 TX$="Status?"
1010 GOSUB 100:REM Transmit TX$ to the TT402A.
1020 GOSUB 200:REM Receive RX$ from the TT402A.
1030 REM This will contain the number of lines to follow.
1040 NLines%=VAL(RX$):REM Convert number to a numeric variable.
1050 FOR N%=1 TO NLines%
1060 GOSUB 200:REM Get a line of status.
1070 STATUS$(N%)=RX$
1080 NEXT N%
```

Individual queries can be sent to the test set and the single reply string is again itself a legitimate command. The full set of queries is listed below:

```
FUNC[TION]?
INP[UT]?
TERM[INATED]?
THRU?
THRO[UGH]?
```

CIRC[UIT] IMP[EDANCE]? NB CIRC[UIT] and CCT are
 CCT RES[ISTANCE]? synonyms; IMP[EDANCE] and
 RES[ISTANCE] are synonyms; so
 CCT IMP is also valid, for
 example.

FILT[ERS]?

RECT[IFIER]?

SCA[LE]?

RANG[E]?

REM[OTE]?

LOC[AL]?

3.17 Error Display commands and queries

ERR[OR]	DISP[LAY]	ON OFF	Turns error display on left hand display on/off.
ERR[OR]	RES[ET]		Resets the last error code to 00.
ERR[OR]?			Returns the number of the last error, or "00" if there has not been an error or "ERR[OR] RES[ET]" has been received.
ERR[OR]	DISP[LAY]? DISP[LAY] ON? DISP[LAY] OFF?		Returns the current status of the error display, i.e. "On" or "Off".

Normally, if the TT402A receives a command which it cannot understand or perform, a message "ERROR n" flashes in the left hand display for five seconds, and the TT402A is unable to receive another command or query for that time. See chapter 7 for a full description of the error codes. This display can be switched off or on by remote control. If off, there is also no delay after an error before another command or query can be received. The error display is set to "On" as a result of power-on, RESET, DCL or SDC.

The code of the last error can also be queried by remote control, regardless of the state of the front panel display. The reply is a simple integer equal to the code of the last error, e.g. "01" or "25". If no error has been received, the response is "00". The error may be queried as many times as required. It will be

reset to "00" as a result of an "ERR[OR] RES[ET]" command, power-on, RESET, DCL or SDC.

3.18 Other commands and queries

RESET	Performs a reset of the TT402A Measurement Section only. The oscillator will not be reset.
VER[SION]?	Returns a string containing the software version number, e.g. "Ver 6.9".
MODEL?	Returns a string containing the model designation of the measurement system. For example: TT402A TT402

4 Sending strings to the Oscillator

This is accomplished by sending a string to the TT402A which begins with the keyword "OSCILLATOR". The keyword can be abbreviated as far as "OSC".

Examples are:

```
OSCILLATOR LEVEL 0 dB
OSC MUTE A
```

Please note that OSC is a keyword for the TT402A, while the following word LEVEL is a keyword for the oscillator. In understanding the error messages this distinction is important.

These strings are sent just like any other string to the TT402A, which then passes it on to the oscillator. The oscillator does not have an address of its own.

Sending the string "OSCILLATOR" to the TT402A with no argument following will produce error 2 (missing argument).

Most commands apply equally to oscillators TO402A, TO402, TO404A and TO404. Differences are noted in the appropriate sections.

4.1 Oscillator level setting commands

OSC[ILLATOR]	LEV[EL] <number>	(DB)	Set the
	NIV[EAU]	(DBU)	oscillator
	PEGEL	(DBM)	level in dB
			relative to
			.775 V RMS, or
			1 mW in the
			expected
			termination.
		V	Set the
		MV	oscillator
		UV	level in volts,
			millivolts or
			microvolts RMS.

Note that the unit is optional, and defaults to units of dB.

Examples:

```
OSC LEVEL 0DB
OSC LEV 1V
OSC LEVEL +8 (NB This sets to +8 ref. 0 dB)
```

The resolution of level setting is 0,1 dB, which makes it possible to set levels to $\pm 0,05$ dB. This makes lack of remote control of the fine level control irrelevant in most cases.

Note that the level setting is correct when either the "Fine Level" control is in its centre position (line at the top), or when the oscillator is in the Local Lockout state ("Fine Level" control disabled). Setting a level with the oscillator in the

Return to Local state permits manual adjustment of up to $\pm 0,1$ dB by means of the "Fine Level" control.

4.2 Oscillator frequency setting commands

OSC[ILLATOR]	FRE[QUENCY]	<number>	(HZ)	Set the
	FREQUENCY		KHZ	oscillator
	FREQUENZ			frequency.

Note that the unit is optional, and defaults to units of Hz.

Examples:

```
OSC FREQ 1KHZ
OSC FRE 1000
```

The resolution of frequency setting is a little more difficult to explain. It is 1 part in 8192 of the maximum frequency of the particular frequency range. The ranges are:

```
10 Hz to 133.12 Hz
133.12 Hz to 1827.84 Hz
1827.84 Hz to 24007.68 Hz
24007.68 Hz to 300 kHz
```

The resolution is worst at the bottom of each frequency range, where each step corresponds to about 1 in 600. Therefore the setting should still be accurate to about ± 1 in 1200, or better than $\pm 0,1\%$, in the worst case. This resolution makes the "Fine Frequency" control irrelevant in most cases.

Note that the frequency settings are correct when either the "Fine Frequency" control is in its centre position (line at the top), or when the oscillator is in the Local Lockout state ("Fine Frequency" control disabled). Setting a frequency with the oscillator in the Return to Local state permits manual adjustment of up to $\pm 1\%$ by means of the "Fine Frequency" control.

4.3 Oscillator output resistance (impedance) commands

OSC[ILLATOR]	OUT[PUT]	RES[ISTANCE]	40[R]
		IMP[EDANCE]	200[R]
	ROUT		600[R]
	ZOUT		
	AUSGANGSWID[ERSTAND]		

Note that the actual values may be different from those shown above.

Examples:

```
OSC ROUT 40
OSCILLATOR OUTPUT IMPEDANCE 600R
```

4.4 Waveform commands

OSC[ILLATOR] SIN[E]

SQU[ARE]

IMD SMPTE Sets the signal to 60 Hz +7 kHz
@ 4:1

DIN Sets the signal to 250 Hz +8 kHz
@ 4:1

CCIF Sets the signal to 14 kHz +
15 kHz @ 1:1

DIM-100 Sets the signal to the composite
3k18 Hz + 15 kHz required by
Otala's method for DIM-100 kHz.

DIM-30 Same for DIM-30 kHz.

W&F	DIN	Select the
W[OW] AND F[LUTTER]	IEC	crystal-
&	CCIR	controlled
	3150[HZ]	3k15 Hz signal.
	3000[HZ]	Select the
	3K[HZ]	crystal-
		controlled 3k0 Hz
		signal.

Note that the commands LEVEL DIFFERENCE, THD+N, K_n, and XTALK (and NOISE and RUMBLE, if the EPROM is specially configured) automatically select SINE, so it is not necessary to command the oscillator separately. Similarly IMD (with any argument as required) selects the appropriate composite signal, and W&F selects the appropriate crystal-controlled tone.

4.5 600R termination commands

These commands apply to the TO402A and TO402 only.

OSC[ILLATOR] TERM[INATIONS]	ON	Applies
TERMS		termination to
		both outputs.
	OFF	Removes
		termination from
		both outputs.

4.6 Mute control commands

4.6.1 Mute control commands - TO402A and TO402

OSC[ILLATOR] MUT[E]	A	Mutes the
	B	specified TO402A
		output.
UNM[UTE]	A	Unmutes the
	B	specified TO402A
		output.

4.6.2 Mute control commands - TO404A and TO404

OSC[ILLATOR] MUT[E]	Mutes the
	outputs.
UNM[UTE]	Unmutes the
	outputs.

4.7 Balance/unbalance commands

These apply to the TO404A and TO404 only.

OSC[ILLATOR] BAL[ANCED]	Sets the output to be balanced. In the TO404, this will be centre-grounded. In the TO404A, grounding is separately controlled.
UNBAL[ANCED]	Sets the output to be unbalanced. In the TO404, this will be floating. In the TO404A, grounding is separately controlled.

4.8 Float/ground commands

These apply to the TO404A only.

OSC[ILLATOR] FLOAT[ING]	Sets the output
FLT	to be floating.
GROUND[ED]	Sets the centre
GND	point (balanced)
UNFLOAT	or return
EARTH[ED]	(unbalanced) of the output to be grounded.

4.9 Local/remote control commands

OSC[ILLATOR] LOC[AL] LOCK[OUT] LLO	Inhibits front panel control.
GO TO LOC[AL] RET[URN] TO LOC[AL] GTL RTL	Restore front panel control.

4.10 Status queries

OSC[ILLATOR] STAT[US]?

Returns the complete status in the form of multiple strings. The first string is a number equal to the number of status lines which will follow. For example, from the TO402A:

8
Sine
Frequency 1000.00 Hz
Level +0.0 dBm
Output resistance 600 R
Unmute A
Mute B
Terminations Off
Return to Local

or:

7
Wow and flutter 3150 Hz
Level +10.0 dBm
Output resistance 75 R
Mute A
Unmute B
Terminations On
Local Lockout

For the TO404A:

8
Sine
Frequency 1000.00 Hz
Level +0.0 dBm
Output resistance 600 R
Unmute
Balanced
Floating
Return to Local

or:

```
7
Wow and flutter 3150 Hz
Level +10.0 dBm
Output resistance 0 R
Mute
Unbalanced
Grounded
Local Lockout
```

For the TO404:

```
7
Sine
Frequency 1000.00 Hz
Level +0.0 dBm
Output resistance 600 R
Unmute
Balanced
Return to Local
```

or:

```
6
Wow and flutter 3150 Hz
Level +10.0 dBm
Output resistance 0 R
Mute
Unbalanced
Local Lockout
```

In each case, the strings returned are themselves legitimate commands to send back to the oscillator. In this way, the complete status of the instrument can be saved on a convenient storage medium, eg. floppy disc, and later used, with the minimum of processing, to restore the unit to the same settings.

The status of the TO402A, TO404A and TO404 oscillators are not identical, but reflect the facilities offered by each.

The simplest way to read the complete oscillator status into the controller is as follows:

```
1000 TX$="Osc status?"
1010 GOSUB 100:REM Transmit TX$ to the TT402A.
1020 GOSUB 200:REM Receive RX$ from the TT402A.
1030 REM This will contain the number of lines to follow.
1040 NLines%=VAL(RX$):REM Convert number to a numeric variable.
1050 FOR N%=1 TO NLines%
1060 GOSUB 200:REM Get a line of status.
1070 OSCSTATUS$(N%)=RX$
1080 NEXT N%
```

Individual queries can be sent to the oscillator and the single reply string is again itself a legitimate command. The reply is identical to the particular string from the complete status shown above. The full set of queries is listed below:

```

OSC[ILLATOR] WAVE[FORM]?
              SIN[E]?
              SQU[ARE]?
              W&F?
              W[OW] AND F[LUTTER]?
                  &
              IMD?

              FRE[QUENCY]?
              FREQUENCE?
              FREQUENZ?
              LEV[EL]?
              NIV[EAU]?
              PEGEL?

              OUT[PUT] RES[ISTANCE]?
                  IMP[EDANCE]?
              ROUT?
              ZOUT?
              AUSGANGSWID[ERSTAND]?

              MUT[E] A?                TO402A and TO402 only
              UNM[UTE] A?
              MUT[E] B?
              UNM[UTE] B?

              MUT[E]?                  TO404A and TO404 only
              UNMUT[E]?

              TERM[INATIONS] (ON)?      TO402A and TO402 only
                              (OFF)?
              TERMS                  (ON)?
                              (OFF)?

              FLOAT[ING]?              TO404A only
              FLT?
              GROUND[ED]?
              GND?
              UNFLOAT?
              EARTH[ED]?

              LOC[AL] (LOCK[OUT])?
              LLO?
              REM[OTE]?
              (GO TO) LOC[AL]?
              (RET[URN] TO) LOC[AL]?
              GTL?
              RTL?

```

4.11 Other commands and queries

OSC[ILLATOR] RESET	Performs a reset of the TO402A oscillator only.
VER[SION]?	Returns a string containing the software version number, e.g. "Version 2.20 of 26 July 1990"
SER[IAL]?	Returns a string containing the serial number of the oscillator, e.g. "Serial number 30"
CAL[IBRATED]?	Returns a string containing the date of the last factory calibration in the format dd-mm-yyyy, e.g. "04-06-1990".
MODEL?	Returns a string containing the model designation of the oscillator, e.g.: TO402A TO404A TO402 TO404

5 CCITT 0.33 Operation

5.1 General

The TI33 CCITT 0.33 Interface and the accompanying software in EPROM are designed to permit the Neutrik TT402A to make measurements to the CCITT 0.33 recommendation as of September 1987. Sequences 00 to 05 are supported. The sequences can be transmitted only, received only, or simultaneously transmitted and received. In the latter case, the transmitted and received sequences may be different (i.e. control of the receiver and transmitter is entirely independent). Interfaces to the ANSI/IEEE-488 and RS-232 standards are also fitted. Either may be used to give commands and to return measurements.

The TI10 IEEE-488 Interface does not contain the circuitry necessary for sending and receiving 0.33 sequences.

The TO404A and TO404 oscillators cannot correctly generate the 0.33 sequences.

5.2 Operation - Sequence transmission only

Send one of the following messages from the remote controller:

SEQT0	Initiate the transmission of sequences
SEQT1	00 to 05 respectively.
SEQT2	
SEQT3	
SEQT4	
SEQT5	

There is a period of approx. 1 second of silence before the header is transmitted. This helps the sequence receiver not to be confused by signals in the 1600 Hz to 1900 Hz range, which could be taken as a carrier.

There is at present nothing to stop remote commands being sent to the oscillator while the sequence is being transmitted. However this is liable to upset internal operation of the instrument and so should NOT be done.

5.3 Operation - Sequence reception only

Send the following message from the remote controller:

SEQRX

This initiates a search for the 0.33 header. Once received, the header contains information as to which sequence is being transmitted. The TT402A decodes this and interprets the sequence accordingly. It does not need to know in advance which sequence is to be transmitted.

After the sequence has finished, the results are output via the remote control interface. The number of lines varies from sequence to sequence. See the later section "Format of Results" for details.

Only three events can restore normal operation after the "SEQRX" command has been issued. One is the reception of an ETX character (the last character in the 0.33 sequence header). The second is reception of a DCL (Device Clear) or SDC (Selected Device Clear) via the IEEE-488 bus. This is, of course, not possible via RS-232. The third is power-down of the instrument.

5.4 Operation - Simultaneous transmission and reception

The same commands are used as above, but the command to transmit MUST appear before the command to receive. This is because the receive command blocks reception of any subsequent messages, and interpretation of any subsequent commands on the same command line, until the results have been transmitted of the sequence it is expecting to receive.

The commands may be on the same line or on two separate lines.

Examples:

SEQT0;SEQRX

or

SEQT1
SEQRX

There is no relation between the sequence being transmitted and the sequence being received. Therefore it is perfectly possible to transmit one sequence to one place, and to simultaneously receive a (possibly different) sequence from another place.

As remarked above, only reception of an ETX character, DCL or SDC, or power down, will restore normal operation of the TT402A.

5.5 Format of 0.33 Results

These are discussed under the heading of the individual sequence number. Reference should be made to a copy of CCITT Recommendation 0.33, available from the CCITT in Geneva or your national Standards Institution or equivalent.

5.5.1 Sequence 00

This is a mono sequence. Results will be taken on the input which was selected at the moment the receive command was accepted.

Source Identification: T402	}	Header
Special signalling: 0		
Program Number: 00		
-0.0 dBu		Received Level
-12.0 dBu		Frequency Response Reference
+0.0 dB	}	Frequency Response
-0.0 dB		
+0.0 dB		
+0.0 dB		
+0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
<-42.0 dB		Total Harmonic Distortion
<-42.0 dB		
+5.9 dBu	}	Compandor Test
-6.0 dBu		
+5.9 dBu		
<-52.0 dBq		Signal-to-Noise Ratio

Received Level is an absolute measurement at 1020 Hz.

Frequency Response Reference is an absolute measurement.

Frequency Response consists of 13 frequencies, the results of which are referred to the Frequency Response Reference result.

Total Harmonic Distortion is measured at 1020 Hz and 60 Hz on the -30 dB range. Results off the scale are reported as "<-42 dB" or ">-18 dB".

Compandor Test consists of three absolute level measurements at 820 Hz.

Signal-to-Noise Ratio is an absolute measurement of noise according to CCIR 468-3, on the -40 dB range. The highest reading occurring in the allowed 8 second period is reported. Readings off the scale are reported as "<-52 dB" or ">-28 dB".

5.5.2 Sequence 01

This is a stereo sequence.

Source Identification: T402			Header
Special signalling: 0			
Program Number: 01			
-0.1 dBu	+0.0 dB	+0 deg	Received Level
-12.1 dBu	+0.0 dB	+0 deg	Frequency Response Ref.
+0.1 dB	+0.0 dB	+0 deg	Frequency Response
0.0 dB	+0.0 dB	+0 deg	
+0.0 dB	+0.0 dB	+0 deg	
+0.0 dB	+0.0 dB	+0 deg	
+0.0 dB	+0.0 dB	+0 deg	
0.0 dB	+0.0 dB	+0 deg	
0.0 dB	+0.0 dB	+0 deg	
0.0 dB	+0.0 dB	+0 deg	
0.0 dB	+0.0 dB	+0 deg	
0.0 dB	+0.0 dB	+0 deg	
0.0 dB	+0.0 dB	+0 deg	
0.0 dB	0.0 dB	+0 deg	
			Total Harmonic Dist'n
			Crosstalk and Circuit Transposition
			Compandor Test
			Signal-To-Noise Ratio

Received Level and Frequency Response Reference are measured at 1020 Hz. The first column is the absolute level of channel A, the second is the level difference (B relative to A), and the third is the phase difference in degrees.

Frequency Response is similar, except that the first column is not absolute, but referred to the Frequency Response Reference result.

Total Harmonic Distortion is similar to sequence 00 above. However, since the TT402A only has one complete channel, it cannot measure THD in two channels simultaneously. It measures the input which was selected at the moment the receive command was accepted.

Crosstalk and Circuit Transposition is measured at 2040 Hz, successively on inputs A and B. The dead channel is measured on the -30 dB range, and referred to the Received Level result. Results off the scale are handled as before. If the level of the "dead" input is high and the "live" input is low, then the message "The circuits are transposed" is reported.

Compandor Test and Signal-to-Noise Ratio are handled like sequence 00, on the input which was selected at the moment the receive command was accepted.

5.5.3 Sequence 02

Sequence 02 is just like sequence 00; the frequencies chosen for the frequency response section are suitable for medium-bandwidth channels.

Source Identification: T402	}	Header
Special signalling: 0		
Program Number: 02		
-0.0 dBu		Received Level
-12.0 dBu		Frequency Response Reference
+0.0 dB	}	Frequency Response
-0.0 dB		
+0.0 dB		
-0.0 dB		
+0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
<-42.0 dB		Total Harmonic Distortion
<-42.0 dB		
+5.9 dBu	}	Compandor Test
-6.0 dBu		
+5.9 dBu		
<-52.0 dBq		Signal-to-Noise Ratio

Received Level is an absolute measurement at 1020 Hz.

Frequency Response Reference is an absolute measurement.

Frequency Response consists of 13 frequencies, the results of which are referred to the Frequency Response Reference result.

Total Harmonic Distortion is measured at 1020 Hz and 60 Hz on the -30 dB range. Results off the scale are reported as "<-42 dB" or ">-18 dB".

Compandor Test consists of three absolute level measurements at 820 Hz.

Signal-to-Noise Ratio is an absolute measurement of noise according to CCIR 468-3, on the -40 dB range. The highest reading occurring in the allowed 8 second period is reported. Readings off the scale are reported as "<-52 dB" or ">-28 dB".

5.5.4 Sequence 03

Sequence 03 is similar to sequence 00; the frequencies chosen for the frequency response section are suitable for narrow-bandwidth (telephone) channels, and the frequency response is performed at -10 dB rather than -12 dB. THD measurement is performed at 1020 Hz only.

Source Identification: T402	}	Header
Special signalling: 0		
Program Number: 03		
-0.0 dBu		Received Level
-10.0 dBu		Frequency Response Reference
-0.0 dB	}	Frequency Response
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.1 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.1 dB		
<-42.0 dB		Total Harmonic Distortion
<-52.0 dBq		Signal-to-Noise Ratio

Received Level is an absolute measurement at 1020 Hz.

Frequency Response Reference is an absolute measurement.

Frequency Response consists of 13 frequencies, the results of which are referred to the Frequency Response Reference result.

Total Harmonic Distortion is measured at 1020 Hz on the -30 dB range. Results off the scale are reported as "<-42 dB" or ">-18 dB".

Signal-to-Noise Ratio is an absolute measurement of noise according to CCIR 468-3, on the -40 dB range. The highest reading occurring in the allowed 8 second period is reported. Readings off the scale are reported as "<-52 dB" or ">-28 dB".

5.5.5 Sequence 04

Sequence 04 simply adds a compandor test to sequence 03.

Source Identification: T402	}	Header
Special signalling: 0		
Program Number: 04		
-0.0 dBu		Received Level
-10.0 dBu		Frequency Response Reference
-0.0 dB	}	Frequency Response
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.1 dB		
-0.0 dB		
-0.0 dB		
-0.0 dB		
-0.1 dB		
<-42.0 dB		Total Harmonic Distortion
+5.9 dBu	}	Compandor Test
-6.0 dBu		
+5.9 dBu		
<-52.0 dBq		Signal-to-Noise Ratio

Received Level is an absolute measurement at 1020 Hz.

Frequency Response Reference is an absolute measurement.

Frequency Response consists of 13 frequencies, the results of which are referred to the Frequency Response Reference result.

Total Harmonic Distortion is measured at 1020 Hz on the -30 dB range. Results off the scale are reported as "<-42 dB" or ">-18 dB".

Compandor Test consists of three absolute level measurements at 820 Hz.

Signal-to-Noise Ratio is an absolute measurement of noise according to CCIR 468-3, on the -40 dB range. The highest reading occurring in the allowed 8 second period is reported. Readings off the scale are reported as "<-52 dB" or ">-28 dB".

5.5.6 Sequence 05

Sequence 05 is a level alignment sequence. However, at the time of writing, it is not clear how the results should be presented.

Source Identification: T402			Header
Special signalling: 0			
Program Number: 05			
-12.0 dBu	0.0 dB	+0 deg	Measurement Level
-12.0 dBu	0.0 dB	+0 deg	
-0.1 dBu	-0.0 dB	+0 deg	Alignment Level
-0.1 dBu	-0.0 dB	+0 deg	
-0.1 dBu	-0.0 dB	+0 deg	
-0.1 dBu	-0.0 dB	+0 deg	
-0.1 dBu	-0.0 dB	+0 deg	
-0.1 dBu	-0.0 dB	+0 deg	
-0.1 dBu	-0.0 dB	+0 deg	
-0.1 dBu	-0.0 dB	+0 deg	
-0.0 dB			PML (Left channel)
-0.0 dB			PML (Right channel)
-0.0 dB			
-0.0 dB			

Measurement Level is -12 dB in both channels. The first column is channel A; the second column is level difference with A as reference; the third column is phase difference in degrees.

Alignment Level is 0 dB in both channels; the three columns are the same as for Measurement Level.

Permitted Maximum Level (PML) is in dB relative to 0 dBm or 0 dBu.

5.6 Delay before Reply

Most or all replies from the TT402A and oscillator are returned within 500 ms of the commands or queries which requested them. The exceptions are the results from 0.33 sequences. The command must be sent before the header is received, and can in fact be sent hours or even days before; there is no timeout. The results are not sent until after the end of the sequence. Therefore there must be a delay of at least the duration of the sequence, which in most cases is in the region of 30 s. This delay should be borne in mind when writing the overall control programme. IEEE-488 controllers, in particular, usually have timeouts built in which are often below this value. Once the first string is ready, the subsequent strings follow with extremely short delays between them.

Appendix B shows a simple programme written to demonstrate the 0.33 sequences. This programme was implemented in GWBASIC using the CEC PC<>488 card, which does have a timeout.

6 IEEE-488-specific commands

6.1 Serial poll, parallel poll, LLO, RTL

These are not implemented at this time.

Local lockout and return to local are handled as strings, as shown above.

6.2 Device Clear (DCL) and Selected Device Clear (SDC)

Either of these will cause both the TT402A and oscillator to return to the state which they enter on power-on.

6.3 Interface Clear (IFC)

Asserting the IFC line will cause the TT402A to await a new string input. Any output strings not yet sent will be lost.

7 Error reporting

If a string is received which is in some way unacceptable to the TT402A it will normally indicate an error by flashing an error message on the alphanumeric display for five seconds. This error message consists of the word ERROR followed by a number showing the type of error, in order to help you to find and correct it. See also section 3.17.

7.1 Error codes in numerical order

- 1- Keyword Not Understood. The keyword is the first word in a string. For example, the command FUCTION THD would produce this error since FUCTION is a misspelt version of the acceptable keyword FUNCTION. Failure to precede oscillator instructions by OSC[ILLATOR] may also result in this error.
- 2- Argument Missing. The argument is the word, or one of the words, after a keyword. For instance, in the phrase FUNCTION LEVEL, the word LEVEL is the argument. Sending the string "FUNCTION" by itself would cause ERROR 2.
- 3- Argument Not Understood. For instance, the phrase FUNCTION RMS would cause ERROR 3, since RMS is not an acceptable argument following the keyword FUNCTION. ERROR 3 would also occur if a number cannot be understood following the keyword RANGE, or is in a range beyond the instrument's capability.
- 4- Illegal Command While In This Function. Examples are: attempting to change input while in the READ OSC function or specifying volts scale while in THD + N function.
- 5- Excess Arguments. This would be produced if there are too many words in the string, e.g. "INPUT A ON" or "FUNCTION THD LEVEL".
- 6- Illegal Rectifier With This Filter. It is not permissible to specify other than QUASI-PEAK with the CCIR 468-2 filter, or AVERAGE with the CCIR ARM filter.
- 7- Punctuation Not Understood. Words are normally separated by the space or horizontal tab (hex 20 or hex 09) characters. Filters are compounded using "+", ",", or "/". Some other characters can cause ERROR 7.
- 8- Oscillator Keyword Not Understood. In ERROR 3 above it is pointed out that an Oscillator instruction would cause an error if the string OSC[ILLATOR] was omitted. This string causes the TT402A to send the remaining characters in that instruction to the oscillator. The following portion of the command therefore becomes the Keyword for the oscillator. The string "OSC LE +20 DB" would cause this error since "LE" is not a proper abbreviation of the keyword LEVEL.
- 9- Oscillator Argument Missing. The string "OSC LEVEL" would cause this error since no level is provided.

- 10- Oscillator Argument Not Understood. The string "OSC W&F 2000" would cause this error since 2000 Hz is not an acceptable wow and flutter frequency.
- 11- Oscillator Argument Out Of Range. For example, the string OSC FREQ 400000 Hz would cause error 11 since 400 kHz is out of the oscillator's frequency range.
- 12- Oscillator Input Buffer Overflow. The oscillator can only accept command strings up to 31 characters long. This is more than enough for any ordinary command. If more than 31 characters follow the OSC keyword before the end of string or multiple statement delimiter character, error 12 will be caused.
- 13- Oscillator Not Installed. Commands having the key word "OSC[ILLATOR]" will cause this error if an oscillator is not fitted to the TT402A.
- 14- Unrecognised Oscillator Error. This would probably only occur if there is an error in communications between the TT402A and the oscillator.
- 15- IMD Command but IMD Card Not Installed. Self explanatory.
- 16- IMD Command but IMD Card Turned Off. The IMD measurement sub-system has switches selecting the option available from the front panel. If the card is set to disabled condition, this error will be caused.
- 17- Illegal IMD Command for this Type of Card. The IMD card provides IMD measurements according to the SMPTE, DIN or CCIF methods. This error is reserved for future versions of software, in case other methods should be implemented.
- 18- Internal Error. This should never occur. If it does, please report to your agent or to us directly, the circumstances under which the error occurred.
- 19- Illegal Filters in this Function. For example in "READ OSC" function, no filters are permitted in order to prevent accidental mistakes due to checking oscillator settings with filters.
- 20- Illegal Combination of Filters. Filters which apply to specific measurement standards may not have other filters in use at the same time.
- 21- Input Buffer Overflow. The TT402A can accept command strings up to 150 characters long. This is more than enough for any reasonable application, but if any more are attempted this error will result.
- 22- Local Ranges Set. An attempt was made to set a range whilst the TT402A was either in LOCAL status or LOCAL RANGES status (see section 3.9).

- 23- Oscillator Timeout. This should never occur. If it does, please report to your agent or to us directly, the circumstances under which the error occurred.
- 24- W&F Command But W&F Card Not Installed. Self explanatory, but also applies to RUMBLE and FIM commands.
- 25- Disbarred Function. An attempt was made to select a function which is either not implemented or the user has chosen to be disabled, eg. NOISE CCIR ARM, K_n.

8 Errors and Omissions in this version

8.1 General

- 1) Serial poll and parallel poll (IEEE-488 only) are not implemented.
- 2) Error 17 is not implemented. If, for example, the command FUNCTION IMD DIM-30 is sent, ERROR 3 will instead be produced.

8.2 0.33

The present software version does not contain all the facilities envisaged to be incorporated later.

9 Compatibility Facilities

The following commands have been included to facilitate compatibility with users who have written programmes for the TP401/MJS401D.

LOA[D]	A	ON	select the load on the specified input
	B	EIN	
		OFF	deselect the load on the specified
		AUS	input
LOA[D]	ON		select the load on the currently
	EIN		selected input
	OFF		deselect the load on the currently
	AUS		selected input
FILT[ERS]	CCIR 468[-2]		select NOISE CCIR 468-3 function
			and QUASI-PEAK rectifier.
	468[-3]		
	ARM		select NOISE CCIR ARM function
			and AVERAGE rectifier.

These filter commands are only valid in the LEVEL or NOISE functions and will retain the same SCALE as in these functions. If an attempt is made to set these filters in any other function, ERROR 19 will be produced.

When in either of the NOISE functions, subsequent FILTER OFF or other FILTER commands will return the instrument to the LEVEL function (rather than produce ERROR 19). The rectifier will return to that set when the LEVEL function was last used but the scale will remain as it was in the NOISE function.

RECT[IFIER]	QUASI-PEAK	select QUASI-PEAK rectifier
	QPK	

Just as in manual operation, it is not possible to select a rectifier which conflicts with a pre-determined specification. If this is attempted ERROR 4, and not ERROR 6, will be reported. ERROR 6 is included for compatibility and should not in normal circumstances be obtained.

The following command has been included in the TO402A and TO402 oscillators to facilitate compatibility with TO404A and TO404.

OSC[ILLATOR]	BAL[ANCED]	The outputs of the TO402A and
		TO402 are always balanced and
		floating.

10 Considerations when Writing Measurement Programs

Many measurement programs are very simple. In Neutrik production automated functional test and computer assisted calibration of all products is now standard practice. The programs for these measurement systems were written by test engineers and technicians having little prior experience of computer programming. Certainly, the time spent on programming results in reduced measurement time, and increased consistency. There are however some new factors to be considered which may not apply during manual measurements.

10.1 Accuracy and Precision of Measurements.

The inherent accuracy of the TT402A/oscillator measurement system is not a function of bus control. However, this accuracy and precision is reflected in the way measured results are returned from the instrument.

The TT402A is designed to give highly accurate measurements of level and to allow for easy calibration of the equipment under test. For this reason the dB EXP scale is provided, to allow setting up of a reference 0 which is equal to the required measurement, and making the errors from this level obvious. The dB EXP scale is only valid for LEVEL and READ OSC functions.

For all other functions uncertainties in the equipment under test, noise and inherent measurement errors conspire to reduce the accuracy of the result. Fortunately, for these practical reasons readings of X-TALK, THD, IMD, etc. are rarely required to an accuracy of better than 0.5 dB. The TT402A is capable of measurements significantly better than this, but meaningless precision is not returned for these types of measurement.

The TT402A therefore outputs results differently according to the function and scale as set out below:

FUNCTION	SCALE			
	dB NORM	dB EXP	VOLTS	%
LEVEL	3 digits	4 digits	3½ digits	N/A
READ OSC	1 decimal	2 decimals		
THD+N	3 digits	N/A	N/A	2 digits
IMD	1 decimal			
FREQ?	The lower of 5 significant digits or 0,001 Hz			

The instrument will round its internal result to match the required precision.

Example: For an input of exactly +4.01 dBm at 1.3 kHz with no measurement error:

"LEVEL; SCALE DB NORM; MEASURE" will output "+4.0 dBu"

"LEVEL; SCALE DB EXP; MEASURE" will output "4.01 dBu"

"FREQ?" will output "1.3000E3 Hz"

The accuracy of these measurements is as stated in the TT402A Operating Manual.

10.2 Accuracy and Settling Time

Beware

As in all A.C. measurements, a settling time is necessary between changing a signal and taking a reading. Both the UNIT UNDER TEST (UUT) and the TT402A need to settle to a final "steady state" value before the result can be considered trustworthy. When manual measurements are made, this is automatically done by the user who notices that the meter or digital readout is changing and waits for it. Unfortunately computers will only do this if asked.

The importance of settling, and the amount of time spent, is directly related to the required accuracy of the measurement.

In the case of the TT402A, settling time depends upon the rectifier in use, the meter scale chosen, and the frequency selected. It is also dependent on the amount of change in frequency or level that is required. A change of any of scale, frequency, level, range or function will require some settling time.

The frequency counter will occasionally count one cycle too many and give a wrong reading. A human operator will naturally tend to ignore such readings and take the "majority vote" reading. Sometimes you will be unlucky when working over the bus, and get an erroneous reading. The same problem can also cause the THD measurement system to momentarily tune to the wrong frequency and give a distortion reading which is not settled, and therefore much too high.

The UUT will also take time to settle due to the constants within it. For example a high Q filter can take several seconds to reach its settled output. Additionally movement of controls on the UUT can take time to be effected. In processor controlled equipment there is a finite time, however small, between an action and a result.

The effect or delay caused by these factors can be made worse by choosing the dB EXP scale when accuracy to 2 decimal places of dBs is not really required.

The best approach to settling a result is to take a number of measurements and compare them. When several results are within a defined error from each other, the result can be considered to have settled. A sample programme written in Microsoft Basic is given in Appendix A to accomplish this.

10.3 Out-of-Range Measurements

Most controllers, when called upon to take the numeric value of the string "Over range", will not report an error, but simply return a value of zero. Routines or subroutines to return

numeric values from the TT402A over the bus should input the results as strings, check for out-of-range messages and flag them, and only take the numeric value of the string when no out-of-range message is detected. See section 1.8. An example of out-of-range handling is shown in line 540 of the sample programme in Appendix A.

10.4 String Input from the TT402A

Some controllers need to know in advance the maximum length of string to expect from a device. The TT402A sends back variable length strings. The longest string at the present time will be 55 characters long, including the end-of-string character(s). Many controllers will include the end-of-string character(s) as part of the string returned to the control programme, hence for example, "Under range" may be taken as 12 characters rather than the 11 visible, if IEEE-488 is in use and a string termination character has been set on the DIL switches; or 13 characters, if RS-232 is in use.

Do not attempt to read a string from the TT402A if you have not commanded it to send one. Do not fail to read a string if you have commanded it to send one. Both circumstances will cause the controller to either hang up or time out.

10.5 Speed of execution of test programmes

Speed of execution is limited by several factors:

- 1) Settling time necessary for correct readings.
- 2) Speed of execution of the test programme itself, usually a high level language.
- 3) Speed of response of the TT402A to commands, limited by its internal programme.
- 4) Gate time for frequency measurements, which is an integral number of cycles lasting at least 200 ms. At low frequencies there may be a significant delay between the command and the first transition of the measured signal which opens the gate.
- 5) Speed of communication via IEEE-488 or RS-232.

A few moments' consideration should reveal that it is difficult to predict how long a programme will take to run. However, (3) is rarely more than 60 ms per reading, and (5) is even shorter. IEEE-488 is normally quicker than RS-232 - but RS-232 at 9600 baud is theoretically capable of transmitting a byte approximately every 1 ms. So even a 20 character string should only take about 22 ms including terminators. In practice the controller is often incapable of keeping up with the serial chip.

High level languages, despite the claims made by the writers of the interpreters and compilers, are normally slow to run. Most high level languages have much the same tasks to do, and so they run at similar speeds. Don't be fooled by names such as "Turbo

Pascal" or "Lightning BASIC", which imply dazzling speeds. If speed is of the essence to you, do some simple tests to check performance and see where the time is spent. Most languages have some sort of timer function. Above all, minimise calculation of floating point numbers, especially transcendental functions (e.g. dB calculations!), unless you have an arithmetic coprocessor or a fast central processor.

You will never get over the fundamental physical requirement to average an A.C. signal over some period of time.

APPENDIX A Programming Example

The following programme sends a sine wave from the oscillator at 0 dBu and 1 kHz. The measurement result from the equipment under test is auto-ranged and settled to the required tolerance and the result is printed to the screen. The main programme is very short. All of the lines from 240 onwards are subroutines which would be reused by an extended main programme.

A detailed explanation of each line is given at the end of the programme.

```
10 DEF SEG = &HD000
20 INIT%=0
30 ADR%=0
40 LEVEL%=0
50 CALL INIT%(ADR%,LEVEL%)
60 TRANSMIT%=3
70 RECEIVE%=6
80 TT402A.ADR%=1
90 MAXLEN=80:'Max length of received string.
100 NSAMPLES%=3:REM No. of samples which must be within
    tolerance.
110 TIMEOUT% = 10:REM Number of times samples are checked before
    timeout.
120 SETTLE.INTERVAL=.2:REM Settling interval, seconds.
130 TOL=.5
140 DIM TEMP(NSAMPLES%): REM a variable array to hold
    intermediate results
150 POINTER%=1:REM Initialise pointer to TEMP() array to 1st
    element.
160 RESULT$="":ERFLAG%=1
170 TX$="LEVEL; RANGE AUTO; SCALE NORM; OSC LEVEL 0; OSC FREQ
    1KHZ"
180 GOSUB 240
190 GOSUB 1000
200 PRINT "Measured level is ";RESULT;" dBu"
210 GOTO 190
220 END
230 REM *****
240 REM Transmit TX$ over bus to address TT402A.ADR%.
250 OP$="MTA UNL LISTEN"+STR$(TT402A.ADR%)+ " DATA '"+TX$+" ' END"
260 CALL TRANSMIT%(OP$,STATUS%)
270 IF STATUS%<>0 THEN PRINT "Error code";STATUS%
280 RETURN
290 REM *****
300 REM Receive RX$ over bus from address TT402A.ADR%.
310 RX$=SPACE$(MAXLEN)
320 OP$="MLA TALK"+STR$(TT402A.ADR%)
330 CALL TRANSMIT%(OP$,STATUS%)
340 IF STATUS%<>0 THEN PRINT "Turnround error code";STATUS%
350 CALL RECEIVE%(RX$,RXLNG%,STATUS%)
360 IF STATUS%<>0 THEN PRINT "Receive error code";STATUS%
370 RX$=LEFT$(RX$,RXLNG%)
380 RETURN
390 REM *****
500 REM - Subroutine to Get a Measurement
510 TX$="MEASURE"
```

```

520 GOSUB 240:REM Send TX$ to TT402A.
530 GOSUB 300:REM Get RX$ from TT402A.
540 IF LEFT$(RX$,4)="Over" THEN ERFLAG%=2 ELSE IF
    LEFT$(RX$,5)="Under" THEN ERFLAG%=3 ELSE ERFLAG%=1
550 RESULT=VAL(RX$):REM Convert result string to a number.
560 RETURN
570 REM *****
1000 REM - Subroutine to "Settle" Measurements
1010 COUNT%=0
1020 GOSUB 500
1030 TEMP(POINTER%)=RESULT
1040 IF POINTER% >= NSAMPLES% THEN 1080
1050 POINTER%=POINTER%+1
1060 GOSUB 1500:REM Wait.
1070 GOTO 1020
1080 POINTER%=1
1090 IF ABS(TEMP(NSAMPLES%)-TEMP(POINTER%)) > TOL THEN 1120
1100 POINTER%=POINTER%+1
1110 IF POINTER% >= NSAMPLES% THEN RETURN ELSE 1090
1120 POINTER%=2:REM Begin to move all samples back one place.
1130 TEMP(POINTER%-1)=TEMP(POINTER%):POINTER%=POINTER%+1:IF
    POINTER% <= NSAMPLES% THEN 1130
1140 POINTER% = NSAMPLES%
1150 GOSUB 1200:REM Wait.
1160 COUNT%=COUNT%+1
1170 IF COUNT% <= TIMEOUT% THEN 1020
1180 PRINT "Settling not achieved"
1190 RETURN
1200 REM *****
1500 REM Delay by SETTLE.INTERVAL seconds.
1510 ENTER=TIMER:REM TIMER is seconds since midnight, reset every
    midnight.
1520 TIME.NOW=TIMER
1530 IF TIME.NOW<ENTER THEN 1550
1540 IF TIME.NOW >= ENTER+SETTLE.INTERVAL THEN RETURN ELSE 1520
1550 ENTER=ENTER-(24*60*60):REM Correction if we happen to run
    over midnight.
1560 GOTO 1540

```

LINE NO. EXPLANATION

```

10..90    Initialise all the constants required by the CEC PC<>488
         card and its subroutines, and the TT402A's IEEE-488 address.

100..160    Initialise settling constants.  Create an array to hold
         enough samples for the settling algorithm.

170..180    Initialise the TT402A to Level function, dB scale, and
         auto-ranging.

190..210    Main loop.  Get a settled result, print it, and loop
         round again for ever.  This programme can only be ended by
         pushing the CONTROL and BREAK keys on the computer.

220    END of Main Programme.  This statement is not actually
         necessary, but is as a reminder to the programmer.

```

240..280 Transmit TX\$ over the IEEE-488 bus to the TT402A. See the CEC PC<>488 manual for further explanation.

300..380 Receive RX\$ over the IEEE-488 bus from the TT402A. See the CEC PC<>488 manual for further explanation.

500..560 The subroutine to make a measurement. Send the command "MEASURE" to the TT402A, then read a reply string from it. Set ERFLAG% to 1 if the reading is within range, 2 if over range, or 3 if under range.

1000..1190 This subroutine makes NSAMPLES% measurements, puts them in a temporary store and compares them. If all are within the required tolerance of the last one then we have a result. If not, the samples are moved back one place, another measurement is made and compared with the previous and so on. It is a good idea to give up after a while or a certain number of tries, if not the programme will get trapped here forever. This could happen if a very noisy equipment was being tested. Lines 1160 and 1170 do this by allowing a maximum of TIMEOUT% measurements.

1500..1560 Wait SETTLE.INTERVAL seconds. The only complication is that the timer function resets to zero every midnight; if this happens, line 1530 detects it, and line 1550 subtracts 24 hours (in seconds). This section of programme is virtually independent of CPU speed, and should be used in preference to "FOR N=1 TO 10000:NEXT N" type of delay loops.

APPENDIX B CCITT Q.33 Demonstration Programme

An explanation of some important points follows the programme.

```
10 DEF SEG = &HD000
20 INIT=0
30 ADR%=0
40 LEVEL%=0
50 CALL INIT(ADR%,LEVEL%)
60 TRANSMIT%=3
70 RECEIVE%=6
80 MAXLEN=80:'Max length of received string.
90 GOTO 250:REM Skip round the low-level subroutines.
100 REM*****Low level subroutines start here*****
110 OP$="MTA UNL LISTEN 1 DATA '"+TX$+"'" END"
120 CALL TRANSMIT%(OP$,STATUS%)
130 IF STATUS%<>0 THEN PRINT "Error code";STATUS%
140 RETURN
150 REM*****
160 OP$="MLA TALK 1"
170 CALL TRANSMIT%(OP$,STATUS%)
180 RETURN
190 REM*****
200 RX$=SPACE$(MAXLEN)
210 CALL RECEIVE%(RX$,RXLNG%,STATUS%)
220 RX$=LEFT$(RX$,RXLNG%)
230 RETURN
240 REM*****
250 REM Main programme starts here.
260 CLS
270 PRINT "Which action do you require:"
280 PRINT "(T) Transmit a sequence
290 PRINT "(R) Receive a sequence
300 PRINT "(B) Transmit and receive a sequence simultaneously"
310 PRINT "(S) Stop and return to BASIC command level"
320 PRINT "(Q) Quit to DOS"
330 PRINT
340 INPUT A$
350 PRINT
360 IF LEN(A$)<>1 THEN 260
370 IF ASC(A$)>=ASC("a") AND ASC(A$)<=ASC("z") THEN
A$=CHR$(ASC(A$)-32)
380 IF A$="T" THEN OP%=0:GOTO 440
390 IF A$="R" THEN OP%=1:GOTO 520
400 IF A$="B" THEN OP%=2:GOTO 440
410 IF A$="S" THEN END
420 IF A$="Q" THEN CLS:LOCATE 1,1:SYSTEM
430 GOTO 260
440 PRINT "Which sequence do you wish to transmit (00..05)"
450 PRINT
460 INPUT SEQ%
470 IF SEQ%<0 OR SEQ%>5 THEN 440
480 SEQNO$=MID$(STR$(SEQ%),2):REM Remove the leading blank.
490 TX$="seqt"+SEQNO$
500 IF OP%=2 THEN TX$=TX$+";segrx"
510 GOTO 530
520 TX$="segrx"
530 GOSUB 110:REM Send the command string, TX$, to the TT402.
```

```

540 PRINT "Command sent"
550 PRINT
560 IF OP%=0 THEN 820
570 GOSUB 160:REM Turn the IEEE-488 bus round to receive.
580 GOSUB 200:REM Attempt to receive a string.
590 REM Since the sequences take 10 to 30 seconds to run, the
600 REM first attempt will certainly time out & so will have to
610 REM be repeated. Once the first string has been read, the
620 REM others will follow almost immediately.
630 IF STATUS%=8 THEN 580:REM Status%=8 indicates a timeout.
640 N%=1:REM N% counts the number of incoming strings.
650 PRINT RX$
660 IF N% MOD 20 = 0 THEN 700
670 GOSUB 200:REM Read another string in from the TT402.
680 IF STATUS%=0 THEN N%=N%+1: GOTO 650
690 IF STATUS%=8 THEN 820 ELSE 800
700 Y%=CSRLIN
710 PRINT "Press <return> to continue: ";
720 A$=INPUT$(1)
730 IF A$<>CHR$(13) THEN 720
740 LOCATE Y%,1
750 PRINT "
760 LOCATE Y%,1
770 N%=N%+1: GOTO 670
780 IF STATUS%=8 THEN 820
790 PRINT "Receive error code";STATUS%
800 PRINT "Receive status";STATUS%
810 GOTO 820
820 LOCATE 24,1
830 PRINT "Press <return> to return to menu: ";
840 A$=INPUT$(1)
850 IF A$=CHR$(13) THEN 260 ELSE 840

```

The programme is written for GWBASIC on a PC, communicating via a CEC PC<>488 card and the IEEE-488 bus. The PC<>488 address in the PC's memory map is set at &HD000. The TT402A's IEEE-488 address is set at 1 within lines 110 and 160.

The subroutine at line 110 transmits TX\$ to the TT402A. The subroutine at line 160 turns the IEEE-488 bus round to prepare to receive from the TT402A. The subroutine at line 200 receives RX\$ from the IEEE-488 bus, i.e. from the TT402A.

Lines 260..430 put up a menu on the screen and select the action in response to input from the operator. Lines 440 onwards are used if a sequence is to be sent only, or sent and received. Operation begins at line 520 if receive only is selected.

Line 570 turns the bus round ready for reception from the TT402A. Line 580 attempts to get a string back. The first attempt will almost certainly fail due to a timeout by the PC<>488 card, which would return error STATUS% 8. If so, the programme loops back repeatedly until success. From then on, strings are received and printed to the screen in batches of 20, so as not to scroll off the screen before being read by the operator.

After the last string has been received, the PC<>488 will time out again; this time, however, the screen is held until the

operator presses the <return> key, when control returns to the top of the loop at line 260.