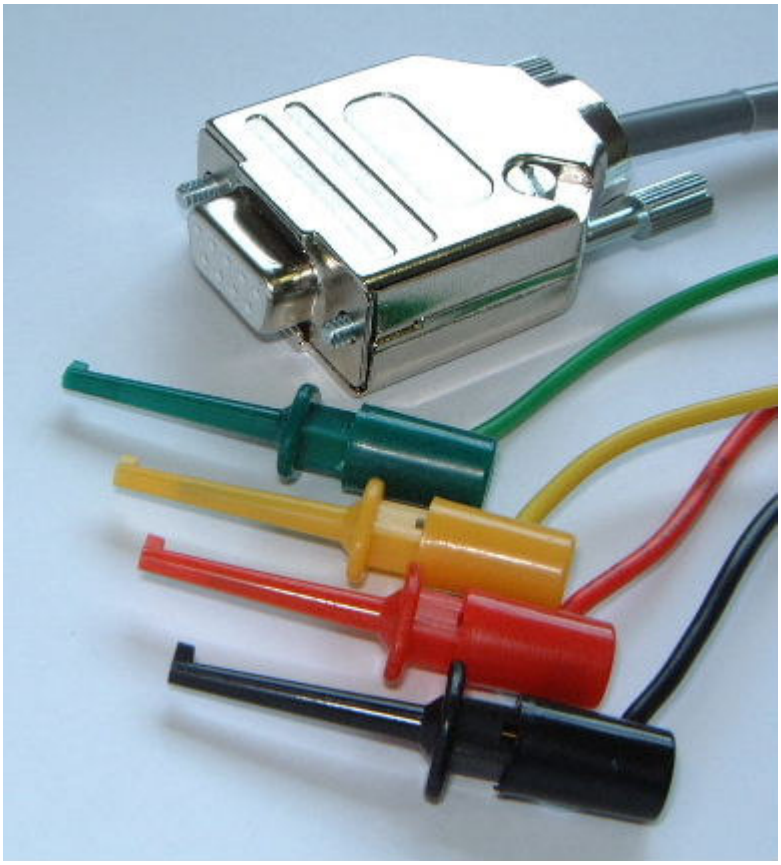
AVIT Research
www.avitresearch.co.uk

I²C Bus Monitor
and
I²C Bus Tool

USER MANUAL



Document Revision 0.1
Date 24 August 2004

AVIT Research Limited • Compass House • Vision Park • Cambridge • CB4 9AD
Tel: +44 (0)1223 257758 • info@avitresearch.co.uk

1. Overview

- Monitors bus traffic on I^2C or SMBus systems at up to 400kHz.
- Connects to standard PC serial port or USB to serial adapter.
- No power supply required.
- Compatible with 5 volt and 3.3 volt bus systems.
- Compact - all circuitry is integrated within the D-Type connector.
- Can be used with standard terminal programs supplied with Windows or Linux.
- Supplied with Windows Software Developer Kit that provides I^2C data to VB, VBA, C or Java applications through an OCX.

Additional features on I^2C Bus Tool:

- Filters messages according to read or write.
- Sends and Receives messages as an I^2C Master device.
- Sends and Receives messages as an I^2C Slave device.
- Capable of transmitting on a Multi-Master bus.
- Trigger connection provides an output to an oscilloscope or can be set to an input for timing events.
- Shows acknowledgements (ACKs) and negative acknowledgements (NACKs).
- Allows VB, VBA, C or Java applications to transmit on the I^2C bus through an OCX.
- Allows user applications to emulate I^2C peripherals using the OCX.

2. This Manual

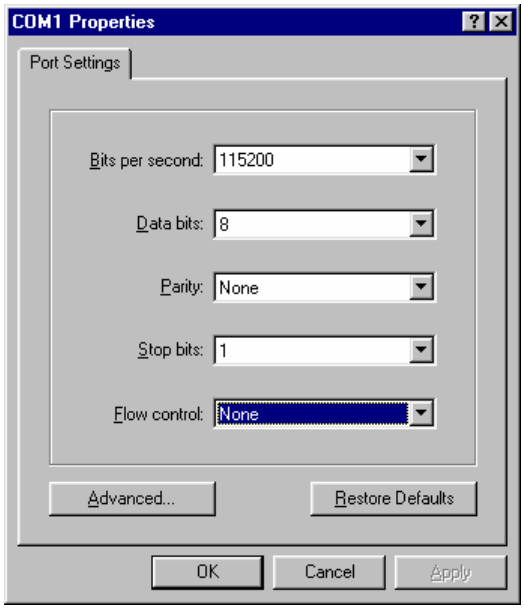
This User manual covers both the I^2C Bus Monitor and the I^2C Bus Tool. The I^2C Bus Tool has all of the features of the I^2C Bus Monitor with additional functionality described in section 5.

3. Getting Started (Windows)

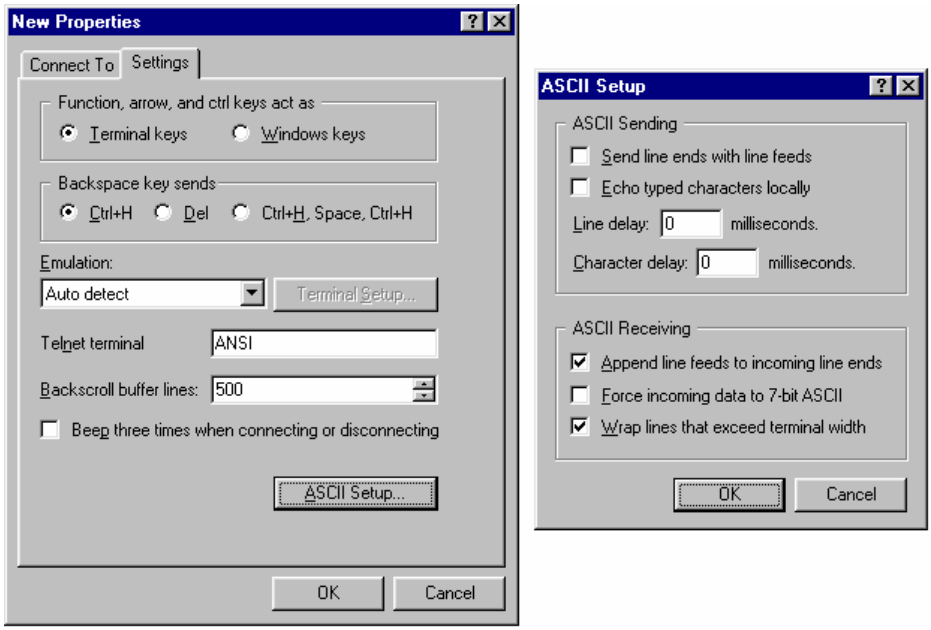
On a Windows system, run the HyperTerminal application. This can usually be found from the start menu under Accessories\Communications\HyperTerminal. This will bring up the contents of the HyperTerminal directory from which the application can be run. It is also available for download from www.hilgraeve.com/htpe/download.html .
Start the HyperTerminal program from the icon:

Hypertrm.exe

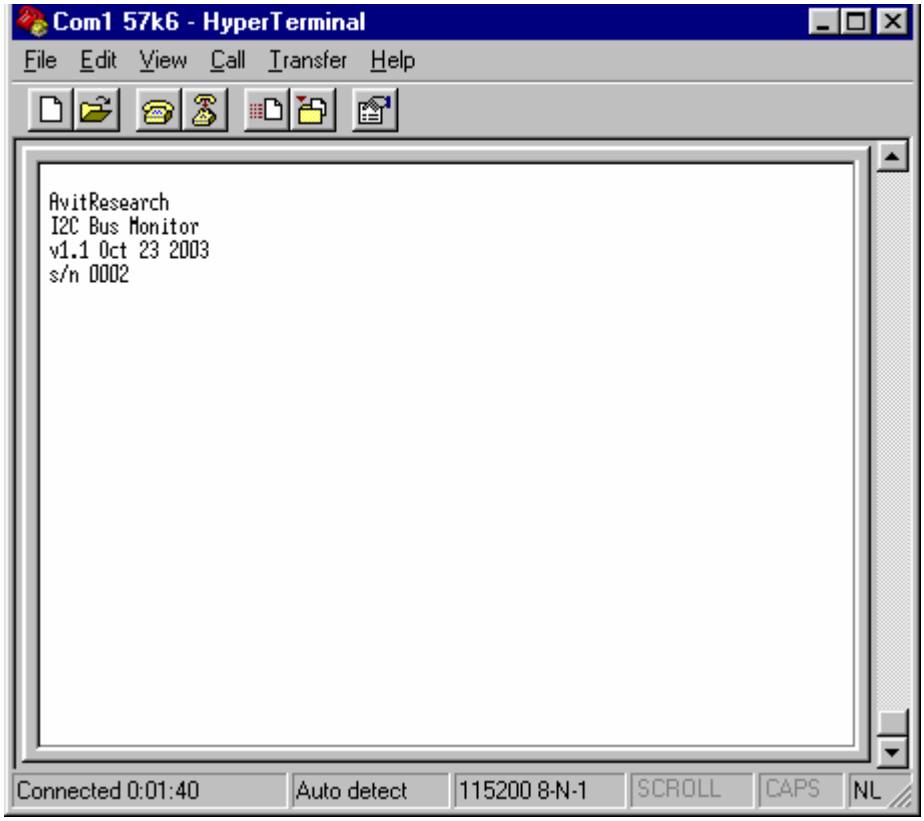
Enter a name for the connection and select an icon. In the following dialog select the Com port that you wish to connect the Bus Tool to. In the next window, select the speed as 115200 bits per second (baud), 8 data bits. No parity, 1 stop bit and no flow control as shown.



Select OK then select the menu item File\Properties and select the settings tab as indicated. Select the ASCII Setup button and switch on the option to append line feeds to incoming line ends.



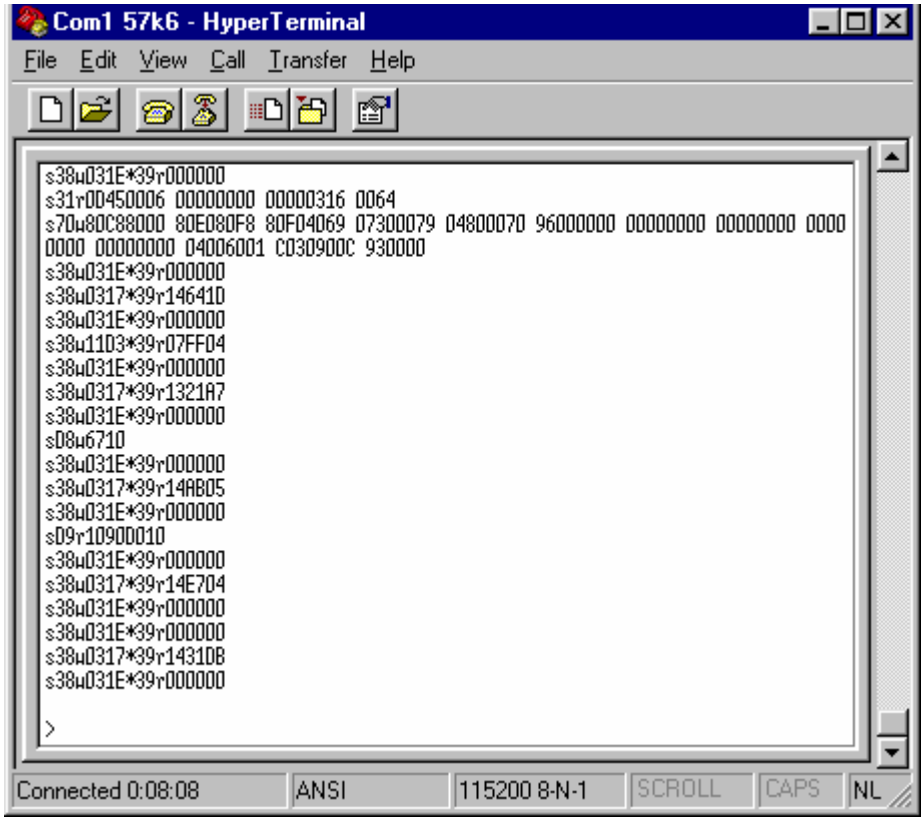
The connection settings can now be saved from the item in the File menu. Plug the I^2C Bus Tool into the com port and the text should appear as shown below.



The probes can now be connected to the target system. The black probe should be connected to ground, the white or yellow to the I^2C data and the blue or green to the I^2C clock.

Important: The ground wire should always be connected first and removed last. This is to avoid excessive voltages on the data and clock pins that could cause permanent damage to the I^2C device.

With the cables connected (and with the target system powered) the messages should be seen scrolling on the screen. Press the space bar to halt the messages and return to set them going again.



Each line shows a separate message (terminated by a Stop condition). The ‘s’ character indicates a start condition. The next hexadecimal number is the address and the character after that will be either ‘w’ for a write operation or ‘r’ for a read. The data section of the message then follows in groups of four bytes – all shown in Hex. A new line is started when a stop condition is seen on the bus (or optionally by a repeated start – see section below). The ‘*’ character in place of an ‘s’ indicates a repeated start condition where there was no stop between one message and the next. While the messages are scrolling, the enter key can be used to insert a blank line. This can be useful for marking positions within a sequence of messages.

HyperTerminal will allow you to scroll back through the messages received but the size of this is limited. To record messages, HyperTerminal can be configured to capture all the text to a file. This can be found under the Transfer/Capture Text… menu item. It will prompt for the filename and once the Start button is pressed, all text appearing on the screen will be written to the file. To view the file the capture must first be stopped which can be achieved through the menu Transfer/Capture Text ►/Stop. It can then be viewed using a text editor such as Notepad or WordPad.

4. Monitor Commands

4.1. Command mode

The display normally shows the I^2C messages scrolling up the screen as they arrive. To change settings the device must be put into the command mode. This is done by pressing the space bar (or virtually any other key). This will cause the messages to halt and display the command prompt ‘>’. Whilst the prompt is shown any I^2C messages received will be discarded. From here settings can be viewed and changed. Entering a ‘?’ and pressing enter will show the available commands.

```
>?
Commands
c -Colour On/Off
cc -Clear Colours
d -TimeDots On/Off
t -Timestamps On/Off
f{+|-}XX[,YY] -Filter
f0 -Filter Off
r -Newline for repeated start
n -CR+LF On/Off
s -Show settings
S -Save settings
bX -Set to X kbaud (eg b115 = 115200 baud)
v -Version Info

>
```

These commands are described in the sections below.
To return to the message display, press the enter key.

4.2. Colour

Entering the character ‘c’ then pressing enter will toggle the colour setting between on and off. With the colour on messages will be coloured according to their address.

```
s38w031E*39r000000
sD8w6710
sD9r1090D010
s38w031E*39r000000
s38w0317*39r137489
s30w81
s38w031E*39r000000
s31r05170041 52AA
s38w031E*39r000000
s38w0317*39r14DB7F
s38w031E*39r000000
s38w031E*39r000000
s38w0317*39r13CD47
s38w031E*39r000000
s31r0D450006 00000000 00000315 0063
s70w80C88000 80E080F8 80F04069 05300079 04800070 96000000 00000000
00000000 00000000 00000000 04006001 C030900C 930000
s38w031E*39r000000
s38w0317*39r123DF7
```

The colours are chosen in order of arrival of messages on returning to the message display. The first address will be assigned red, the second blue, the third yellow (appearing on HyperTerminal as a fawn colour) and all following ones cyan. The assigned colours are saved with the other settings (see section below) so that the appearance can be consistent across sessions. The assigned colours can be cleared by entering the command ‘cc’ at the prompt. When returning to the message display the colours will be re-assigned in order of messages received.
When filters are used to show only messages from a certain addresses (see section below) the colours are allocated differently. In this case the first message in the filter list is assigned blue for write messages and red for read. The second is assigned cyan for write and yellow for read.

4.3. Time Dots

The time dots can be toggled on and off with the command ‘d’. With time dots enabled, periods of inactivity of the I^2C bus are indicated with a ‘.’ character for every elapsed second without a message being received.

4.4. Timestamps

The timestamps can be toggled on and off with the command ‘t’. When timestamps are on a time is shown at the start of each line. The display shows the seconds and milliseconds of the time of arrival of the message. The seconds will count up to 99 and wrap round to zero.

```
00.097 s30w81
00.100 s31r05170030 5097
00.146 sD8w6710
00.147 sD9r1090D010
00.203 s30w81
00.206 s31r05170030 5097
00.307 s30w81
00.349 s31r05170030 5097
00.390 s31r0D450003 00000000 00000510 005D
00.417 sD8w6710
00.418 sD9r1090D010
00.493 s30w81
00.497 s31r05170030 5097
00.598 s30w81
00.606 s31r05170030 5097
00.658 sD8w6710
00.659 sD9r1090D010
```

4.5. Filters

The messages can be filtered to remove one or two addresses from the display or to show only messages from one or two addresses. The filter addresses apply to both read and write transfers i.e. if 0x30 is entered it will apply to both 0x30 and 0x31. If odd addresses are entered they will be converted to the corresponding even address.
If you wish to see only messages for address 0x30 (and 0x31) the command ‘f+30’ should be used. If you wished to see messages for addresses 0x30 and 0x70 (and 0x31 and 0x71) you should use the command ‘f+30,70’. The order is only significant if colours are used (see section above).
If you wish to exclude messages for address 0x38 (and 0x39) the command ‘f-38’ should be used. If you wish to exclude messages from addresses 0x38 and 0xD8 (and 0x39 and 0xD9) the command ‘f-38,D8’ should be used.
The filters can only be applied on up to two addresses at a time. The I^2C Tool provides read/write filtering options as described in section 5.1 below. The windows application supplied with the I^2C Bus Monitor can provide more filtering options.

4.6. Repeated Start New Line

Some protocols that use the I^2C bus or the SMBus require that stop conditions are not used between messages. This results in the message display showing all messages running together without new lines in between. It means that timestamps cannot show the time of each of the messages. For this situation the command ‘r’ should be used to switch on the new line for repeated starts. This will cause each message to be shown on a new line whether or not the previous one was terminated by stop condition.

```
94.138 s38w031E
94.138 *39r03A362
94.149 s38w031E
94.149 *39r039CF7
94.151 s38w0317
94.152 *39r16B158
94.154 s38w031E
94.155 *39r0343D9
94.162 s30w81
94.165 s31r05170002 3049
94.167 s38w031E
94.167 *39r039058
94.169 s38w0317
94.170 *39r154D47
```

4.7. Show Settings

Entering the command 's' (lower case) causes the current settings to be shown.

```
>s

Baud=115200
Colour On
Filter = -38
TimeStamp On
TimeDots On
RepeatedStartNewline On
CR+LF Off

>
```

4.8. Save Settings

The command 'S' (upper case) causes the current settings (as shown in the above section) to be saved into non-volatile memory so that they will be used the next time the I^2C Bus Tool is plugged in.

4.9. Baud Rate

The baud rate which the I^2C Bus Monitor uses to communicate with the PC is set to 115200 as default. It can be changed to any standard rate from 57600 to 921600 to allow it to be used with slower systems or for a greater message throughput.
The baud rate is set with the command 'b' followed by a number in kbaud. e.g. for a baud rate of 460800 the command would be 'b460' and the display would appear as:

```
>b460

Baud=460800
```

A prompt is not shown after the confirmation because the I^2C Tool is waiting for communication at the new baud rate. At this point the HyperTerminal should be disconnected (menu Call/Disconnect), then the baud rate changed in the properties window (menu File/Properties then the Configure button). In this example the Bits per second should be changed to 460800 then the OK button pressed on both windows. HyperTerminal should then be reconnected (menu Call/Call) and the

enter key pressed. This will set the I²C Tool into the message display mode at the new baud rate and if it is connected to an active I²C bus the messages should be seen.
At this point the baud rate is only set temporarily so that if it is physically disconnected then plugged back in it will revert to the previous baud rate. To keep the new baud rate the command prompt should be called up with the space bar and the command 'S' used to save the settings.
If you discover that HyperTerminal will not connect at the new baud rate, it can be set back to the original rate and the I²C Tool removed from the connector and plugged back in (this may need to be done twice before it will communicate at the original baud rate).
More discussion of baud rates and bus speeds are in section 8 below.

4.10. Version Info

The 'v' command displays the version information that is seen at start-up.

4.11. CR/LF

Some terminal emulators do not provide the facility to add a line feed (LF) character to the carriage return (CR) characters they receive. The I²C Tool normally only transmits the CR character at the end of each line to reduce the amount of data that needs to be sent over the serial link. Therefore if the terminal program does not add the LF character itself all the text can appear on a single line.
To get over this problem the I²C Tool can be set to send both a CR and LF character at the end of each line. This is switched on and off with the 'n' command. The setting will be saved by the "Save Settings" command described in section above.

5. I²C Bus Tool Commands

5.1. Additional Filtering

In addition to the filtering described in section 4.5 above the I²C Bus Tool can filter out messages depending on whether they are read or write using a 'r' or 'w' suffix to the address. Example commands are:

f+30w	Only display writes to address 0x30.
f+30w,38w	Only display writes to addresses 0x30 and 0x38.
f-30r,38	Display everything except reads from address 0x30 (shown as 0x31) and any accesses to address 0x38.
f-30r,38r	Display everything except reads from address 0x30 (shown as 0x31) and 0x38 (shown as 0x39).

5.2. Acknowledge Bit Detection

The I²C Bus Tool can detect and display the condition of the Acknowledge bit that occurs at the end of each byte on the bus. There are two levels of display. The "aa" command causes the Tool to show the condition for each byte as either acknowledged (Ack – shown as "a") or not-acknowledged (NAck – shown as "n"). The stop condition at the end of each message is shown with a "p" character. Note that all read messages (odd addresses) should finish with a NAck.

```
26.881 s38wa03a17a*39ra3Fa55a04np
26.884 s38wa03a1Ea*39ra0Da79aB8np
26.886 sD8wa67a10ap
26.887 sD9ra10a90aD0a10np
26.908 s38wa03a1Ea*39ra0EaD7a3Dnp
26.910 s38wa03a17a*39ra41aD6a56np
26.913 s38wa03a1Ea*39ra0DaA0a1Enp
26.929 s30wa81ap
26.931 s31ra05a17a00a02a30a49np
26.939 s38wa03a1Ea*39ra0BaACa3Bnp
26.942 s38wa03a17a*39ra3FaC1a65np
```

If the command “an” is used the Tool will just display the NAck conditions with the “n” character.

```
51.651 s38w0317*39r3EFD4An
51.653 s38w031E*39r0C7927n
51.665 s30w81
51.668 s31r05170002 3049n
51.680 s38w11D3*39r000004n
51.682 s38w031E*39r0A3D18n
51.684 s38w0317*39r3DBD49n
51.686 s38w031E*39r0A935Dn
51.689 sD8w6710
51.690 sD9r1090D010n
```

The Ack/NAck display can be turned off with the command “a0”.

5.3. Master Transmit

To transmit a message as an I^2C Master the data to be sent must be set up in a buffer. The I^2C Tool has 8 buffers of 32 bytes each. Therefore the maximum size of a transmit message is 32 bytes plus the address. The buffers are numbered 1 to 8 and the command format is:

tx*(1..8)=AA DD DD DD DD*

Where *AA* is the address in hex and *DD* are the data bytes in hex (each hex number must be entered as two digits e.g. ‘07’, not ‘7’). The address and data bytes can be separated by spaces. The command must be entered in command mode at the '>' prompt. Example commands are:

tx1=A0 00 12	Sets buffer 1 to address A0 with data bytes 00 and 12
tx8= 3001020304050607080 90a0b0c0d0e0F	Sets buffer 8 for address 30 with data bytes from 01 to 0F
tx1	Displays the contents of tx buffer 1
tx	Displays the contents of all tx buffers

Once the buffers have been set up the messages are sent from the message display mode. Pressing one of the number keys from 1 to 8 will cause that messaeg buffer to be transmitted at the next opportunity. The message will appear in the normal scrolling display beginning with a capital 'S' to indicate that it originated from the I^2C Tool.

5.4. Master Transmit then Receive

Many I^2C devices require a message to be sent to them followed immediately by a repeated start condition and then an I^2C read from the same device. This can be achieved by appending the transmit buffer command with 'r*LL*' where *LL* is the length of the data to be received from the slave

device in hex (must be entered as two digits). For example, to read 4 bytes of data from an EEPROM device that takes a single byte address, the transmit buffer would be set up with the command "tx1=A0 00 r04". When this buffer is transmitted with the '1' key in the message display mode it will appear as "SA0w00*A1rFFFFFFFF" (assuming the device is blank). Example commands are:

tx1=a040r0A	Read 10 bytes of data from EEPROM address 0x40 using transmit buffer 1.
tx2=3801ECr03	Send bytes 01 and EC to device at address 0x38 then send a repeated start and read 3 bytes from address 0x39.

5.5. Master Receive/Request

To perform a master receive from an I^2C device the receive parameters must be set up. This is done with the command "rx*AA=LL"*. Where *AA* is the address to read from (must be odd) and *LL* is the number of bytes to read. Both are entered as two digit hex values. The command "rxA1=04" will set up the receive operation to read 4 bytes of data from the slave device at address 0xA1. The receive is performed from the message display mode by pressing the 'r' key. This will produce a display such as: "SA1rFFFFFF".

For some slave devices the length of data to be read is not known by the master. In these cases it is common for the slave device to indicate the length of data by the value of one of the first bytes it sends. The I^2C Tool can retrieve the length from the message by the command "rx*AA>PP*{+|-}*LL*" where AA is the address to read from (must be odd), PP is the position within the received message to extract the length from and LL is an offset to that byte so that it gives the total number of bytes to receive. All these are entered as two digit hex values. Examples are:

rx31>00+00	Receive from address 0x31 a message whose length is given by the first byte received. e.g. S31r04010203 where 04 is the total number of bytes in the message.
rx31>01+02	Receive from address 0x31 a message whose length is given by the value of the second byte plus 2. e,g, S31r0A051122 334455 where the number of bytes received is given by the byte value 05 plus 2 to give a total of 7 bytes.
rx91>03-F0	The length is given by the 4th byte received minus 0xF0. e.g. S91r0B2400F8 00000000 where the length is given by the byte 0xF8 minus 0xF0 to give a total of 8 bytes.

5.6. Slave Transmit

The I^2C Tool can emulate slave devices such as DACs by configuring buffers with data to be sent when it is requested by an I^2C Master device. This is done in the same way as the master transmit buffers are configured but with an odd address.

tx1=A1 0001020304	Configure buffer 1 to send data 00 01 02 03 04 when requested by a master.
tx2= 310000	Configure buffer 2 to send data 00 00 when requested by a master.

The buffers are searched in reverse order so if two buffers have the same slave address, the higher numbered buffer will be used. If the master device requests more data than is held in the buffer the I^2C Tool will send zero bytes until the master terminates the transfer.

5.7. Slave Receive

The I²C Tool automatically acknowledges a write operation to an address for which a slave buffer has been configured. For example if buffer1 is set for address 0xA1 the I²C Tool will acknowledge address 0xA0 and any following data bytes.

5.8. Baud Rate

All transmit and receive operations done as a master cause the I²C Tool to generate the I²C clock. The default for this is 50kHz. It can be set to other values with the commands:

Command	Speed
i400	400kHz
i200	200kHz
i100	100kHz
i50	50kHz

5.9. Trigger Commands

The trigger connection (Red wire) can be set as either an input or output. When set as an input it will show rising or falling edges (or both) in the message display. The commands for this are:

ti+	Set Trigger as an input to capture rising edges
ti-	Set Trigger as an input to capture falling edges
ti+-	Set Trigger as an input to capture rising and falling edges
ti0	Switch off the trigger input
ti	Show Trigger input settings

The message display shows a character "T" for rising edges and "t" for falling edges. If these occur during an I²C message the characters will be seen within that message. If they occur outside a message they will be shown on a separate line with a timestamp (if timestamps are enabled).

```
21.848 s30w81
21.849 t
21.849 s31rT05170002 3049
21.920 sD8w6710
21.921 sD9r1090D010
21.950 s30w81
21.951 t
21.953 s31rT05170002 3049
```

The trigger can be set to output a pulse when a sequence of characters is seen on the bus. This uses transmit buffer 8 to store the match sequence so any transmit messages stored in buffer 8 will be lost when a match sequence is set. Examples of the command are:

tm	Show trigger match settings
tm=s300102	Output a trigger pulse when the sequence s30w0102 is seen.
tm=s31FFFECC	Output a trigger pulse when the sequence s31rFFFECC is seen.
tm=300102	Output a trigger pulse when the sequence s30w0102 or 300102 is seen.
tm0	Switch trigger match off.

Whilst in the message window the trigger pin can be toggled between high and low by sending Ctrl-T characters. If trigger input is on then this will be ignored. If Trigger on Match is on then it will be switched off. Each Ctrl-T will toggle the trigger pin between high and low states and show a trigger character of "T" or "t" in the message window (as happens with trigger input).

Using the supplied OCX, windows applications can set the trigger to output pulses on other conditions. It can be set to output whenever a message is transmitted by the tool or whenever a slave message is received or transmitted by the tool. It also allows the polarity to be changed.

6. I²C Bus Speed

The tool is designed to work at bus speeds up to 400kHz. However, to achieve these speeds it is necessary for the tool to hold the bus for short periods of time to prevent it losing data. Whilst monitoring the bus at 400kHz the tool will not slow the bus at all unless the Ack/Nack detection is on (see section 5.2 above). In this case the bus will be held on each byte just before the acknowledge bit for a maximum time of 2µs.
When slave addresses have been set the tool will hold the bus for a longer time at the acknowledge bit for start addresses in order to determine whether or not it is being addressed. The maximum time it will be held is 5µs.
This has the requirement that, for the tool to correctly monitor bus traffic, all I^2C Master devices must support clock stretching (as required by the I^2C specification) when operating over 400kHz or over 100kHz when the tool is emulating slave devices.

7. Windows Application Capabilities

Certain restrictions to the operation of the tool exist when using the terminal interface. The transmit messages are limited to 32 bytes and it is not possible to set certain modes for the trigger connection. Windows applications using the supplied OCX can transmit messages of up to 64k bytes in length and can operate as a slave device where the data requested by the master is calculated just prior to being sent (e.g. emulating a memory device where the address to be read is sent just before the read).

To achieve these features the OCX exchanges data with the I^2C Tool during message transmission. The I^2C Tool holds the bus at appropriate times for the data to be transferred between the PC and the tool for it then to be sent on the I^2C bus. Due to the reaction time of the PC this can result in the bus being held for periods of around 4ms. This is well inside the SMBus limit of 35ms and so communications involving these types of message are possible on SMBus systems.

The OCX allows the trigger output to be activated whenever the I^2C Tool transmits as an I^2C Master or whenever it participates in bus transfers as an I^2C Slave device. The polarity of the trigger connection can be changed between active high and active low. The trigger pin can be set permanently high or low. It is possible to configure the trigger as a slave poll request line such that when the application has data to transfer to the master the trigger line is activated and when the master reads the data from the tool it is de-activated. Similarly when acting as the I^2C Master device the tool can respond to the trigger line (as an input) by reading from a slave device.
The methods and properties for implenting these features of the tool are described in the help file for the OCX which is installed as part of the Software Developer's Kit. Examples of using them from Visual Basic and Visual C++ are also incuded.

8. Communication Speeds

The baud rate of the PC serial port needs to be high enough to ensure that the received message data can be transferred quickly enough to the PC without the buffer in the I^2C Tool filling up. If the buffer does get full, data will be lost and a '#' character will be sent. Therefore whenever this character is seen on the display, data around it may not be correct. If this happens frequently a number of steps can be taken to improve the throughput.

- Messages from certain addresses can be filtered out. Any message that is filtered out of the display will not consume bandwidth and will not be stored in the I^2C Tool buffer therefore increasing the bandwidth for the other messages.
- Colour can be switched off. For each colour change in HyperTerminal five characters have to be stored in the I^2C Tool and sent through the serial port. When timestamps are on they are displayed in black so each message requires two colour changes (10 characters). Switching colour off can therefore increase the message throughput significantly.
- Timestamps could be switched off if not required. These take 7 characters each so switching them off can improve message throughput.
- Increase Serial Port speed. This is the most convenient solution but the serial port drivers supplied with Windows usually support 115200 as the highest speed. See section 9 below for details of how to increase this limit.

8.1. Resetting Baud Rate

If the I^2C Monitor is used on a PC with a high baud rate and then moved to another that cannot achieve that baud rate, communications will not initially be possible. To get the tool working at the default baud rate of 115200, disconnect the blue and white probes (yellow and green for the Tool) from the system under test and connect them to each other. Unplug the I^2C Monitor from the PC and plug it back in (with HyperTerminal running at 115200 baud). Initially you will see some corrupt characters as the banner is displayed at the higher baud rate. Then you should see the banner displayed correctly at 115200 baud. This baud rate will only be temporary and if it is to be kept the settings should be saved using the command 'S' at the command prompt.

9. Increasing PC Serial Port Speed

The drivers for the serial ports that are supplied with Windows are generally only capable of speeds up to 115200 baud although the hardware in most PCs is capable of running faster. There are alternative drivers available from the following locations:

For all versions of Windows and Linux: www.devdrv.com/shsmod/index.htm

For Windows 2000 and XP only: www.rippstein.net/HiSerialEN.htm

Alternatively the I^2C Tool can be connected through a USB to serial port adapter. Some of these will support baud rates up to 921600 but you need to check the specifications carefully. We recommend the product code ZP43 from www.maplin.co.uk which will support 460800 baud.

USB serial ports will generally be assigned to the next available COM port. It is possible that this will place it out of range of the ports that can be selected from certain versions of Hyperterminal. To change the allocated port in Windows 98:

Run the Windows Device Manager (Control Panel/System, Device Manager tab). The USB to Serial Port device should be selected on this list and the properties button pressed. On the Resources tab, the "Use automatic settings" box should be cleared and the "Input/Output Range" for the device should be changed to the following settings to assign it to one of the conventional COM ports.

COM Port	Input/Output Range
COM1	03F8-03FF

COM2	02F8-02FF
COM3	03E8-03FF
COM4	02E8-02FF

To change the allocated port in Windows XP/2000:

Display properties of My Computer and select the “Device Manager” button on the Hardware page. Select the device in the list under the “Ports (COM & LPT)” heading. View the properties and under the “Port Settings” tab select the “Advanced” button. The COM port can be selected from this window. The USB to serial convertor may need to be disconnected then reconnected for the new port setting to take effect.

10. Windows Terminal Programs

This manual has referred to the I²C Bus Monitor and I²C Bus Tool being used with the HyperTerminal program that is supplied with Windows operating systems. Unfortunately the version that ships with Windows XP has problems with the display of the messages that have scrolled off the top of the screen. If you use the scroll bar to view previous messages they do not appear correctly. The latest version of HyperTerminal can be downloaded from www.hilgraeve.com/htpe/download.html.

Other terminal emulation software is available.
A trial version of the ZOC terminal emulator from Emtec can be downloaded from www.emtec.com/zoc/index.html.

11.Linux Systems

The I²C Tool can be used on Linux systems using the MiniCom application. This application is shipped with most distributions or can be obtained from http://alioth.debian.org/projects/minicom/. The program should be configured to use one of the serial ports e.g. /dev/ttyS1 or /dev/ttyS2. The speed should be set to 115200 8N1. It is advisable to delete the modem initialisation string to prevent it sending it when connecting. If the colours of the I²C Tool are to be used then MiniCom colours should be set to a white background and black foreground so that the timestamps are visible. When MiniCom is running entering Ctrl-A,Z will bring up the main menu and option A will switch on the “Add Linefeed” option which is required to produce the correct display (unless the I²C Tool option CR+LF is switched on).

12. Palm Pilot

The I²C Tool can also be used with Palm devices connected via a modem cable. A terminal emulator program called ptelnet (formerly PalmTelnet) is available from http://netpage.em.com.br/mmand/ptelnet.htm.
Under the Serial menu set the port to Serial, the Baud to 115200, parity to N, Word to 8 and StopBits to 1. Clear the tick boxes for Xon/Xoff and RTS/CTS.
Under the Terminal menu set the Mode to Serial, Return to CR and make sure the Local echo tick box is cleared.
Connect the I²C Monitor and press the On button at the bottom left of the screen. This should cause the banner to be displayed. If the text appears all on the top line of the display then the option for CR+LF needs to be turned on in the I²C Monitor. To do this enter a space character which should

cause the prompt character ('>') to appear at the left of the screen. Then enter an 'n' character and a newline. This should cause the display to show:

```
>nn 0012 23 2003
CR+LF On

>
```

Then enter an 'S' (must be upper case) followed by newline to save the settings. When connecting the next time the display should be the same as that seen with HyperTerminal.
Version 0.61 of ptelnet has an option to log the data to a MemoPad entry. This operates much like the text capture in HyperTerminal.