

CALIBRE

I2C for Windows 95 Manual

WINI2C/I95
Issue 2.4
16/07/1999



Welcome to the Calibre I²C for Windows 95 application. This application is designed to allow non-specialist users to run I²C Bus operations via the Calibre I²C adapter (ICA90 or ICA93). Please note that in this manual sometimes refers to the I²C Communications Adapter User Manual. It will prove helpful if you have this to hand.

If you have any queries relating to this or any other I²C product supplied by Calibre please visit our web site www.calibreuk.com.

For technical support please e-mail techsupport@calibreuk.com or send your queries by fax to (44) 1274 730960, for the attention of our I²C Technical Support Department.

COPYRIGHT

This document and the software described within it are copyrighted with all rights reserved. Under copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated, or reduced to electronic medium or machine readable form, in whole or in part, without prior written consent of Calibre UK Ltd ("Calibre"). Failure to comply with this condition may result in prosecution.

This computer program is the property of Calibre. The program is protected under copyright law and is licensed for use only by the user who has purchased the license from Calibre or Calibre's authorised distributor. Sale, lease, hire rental or re-assignment to, or by, a third party without the prior and written permission of Calibre is expressly prohibited.

Calibre does not warrant that this software package will function properly in every hardware/software environment. For example, the software may not work in combination with modified versions of the operating system or with certain network adapter drivers.

Although Calibre has tested the software and reviewed the documentation, CALIBRE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS SOFTWARE OR DOCUMENTATION, THEIR QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. THIS SOFTWARE AND DOCUMENTATION ARE LICENSED 'AS IS', AND YOU, THE LICENSEE, BY MAKING USE THEREOF, ARE ASSUMING THE ENTIRE RISK AS TO THEIR QUALITY AND PERFORMANCE.

IN NO EVENT WILL CALIBRE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE OR DOCUMENTATION, even if advised of the possibility of such damages. In particular, and without prejudice to the generality of the foregoing, Calibre has no liability for any programs or data stored or used with Calibre software, including costs of recovering such programs or data.

Copyright Calibre UK Ltd
(c) 1999 Cornwall House, Cornwall Terrace
 Bradford, BD8 7JS. UK.
 E-mail: sales@calibreuk.com
 Web site www.calibreuk.com
 All World-wide Rights Reserved

Issue 2.4 16/07/1999

All trade marks acknowledged

Calibre operates a policy of continued product improvement, therefore specifications are subject to change without notice as products are updated or revised.
E&OE.



I ² C FOR WINDOWS 95 MANUAL	1
1.1. System requirements	1
1.2. Installation	1
1.3. Application Overview	1
1.4. Configuring I ² C for Windows 95.....	2
1.5. File menu commands	2
1.6. Edit menu commands.....	3
1.7. Function menu commands.....	3
1.8. Progress Window	6
1.9. Parameters - Their meaning.....	6
1.10. Changing Parameters	7
1.11. Trouble Shooting Hardware.....	7
1.12. Trouble Shooting Software	8
1.13. Adding Comments to Data	8
1.14. Changing Files Manually	9
THE MOST COMMONLY ASKED I2C QUESTIONS	10
2.1. General Questions.....	10
2.2. DOS Software Questions	10
2.3. WINDOWS 95 and NT Questions.....	11

I²C FOR WINDOWS 95 MANUAL

1.1. System requirements

1.1.1. The software requires a PC with a 386 or better processor, Windows 95, a VGA monitor and a mouse or other pointer.

1.1.2. You must have installed the appropriate I²C adapter (ICA90 or ICA93) into your PC.

1.2. Installation

1.2.1. Install Procedure

1.2.1.1. Unpack the software in accordance with the instructions provided with the license, then follow the on screen instructions.

1.2.1.2. Assuming an I²C adapter has been registered the Setup program provided by Calibre UK Ltd. installs all the necessary components for this application.

1.2.1.3. If this is the first time an I²C adapter has been used on this PC (running Windows95) then it is necessary to register the device driver to do this follow the registration procedure detailed below.

1.2.2. Registering the Driver

1.2.2.1. From the DOS prompt run the Reg95.exe program when prompted enter the offset of the adapter from 310H for example

Hardware address = 310H enter 0

Hardware address = 312H enter 2

" "

Hardware address = 31EH enter E

Hardware address = 31FH enter F

1.2.2.2. The address offset entered must match the base address link setting of the adapter.

1.2.2.3. When prompted enter 0 for disabled IRQ - this application does not use interrupts.

1.2.2.4. Having registered the driver return to Windows95 and then shut the PC down. Guidance relating to the registry can be found on www.calibreuk.com.

1.3. Application Overview

1.3.1. The I²C for Windows 95 software package is designed to allow the non-specialist user to run simple I²C-bus operations via an ICA90 or ICA93LV.

1.3.2. The application supports four basic I²C operations

1. Master (the adapter located in the PC) reading from a Slave (located elsewhere)
2. Master (the adapter located in the PC) writing to Slave (located elsewhere)
3. Slave (the adapter located in the PC) being read by a Master (located elsewhere)
4. Slave (the adapter located in the PC) being written to by a Master (located elsewhere)

1.3.3. Variants of these basic operation to suit a particular task, are also available as follows

5. Memory read - the adapter located in the PC is a master and reads the entire contents of the I²C memory.
6. Memory write - the adapter located in the PC is a master and writes data to the complete I²C memory.
7. Teletype - the adapter located in the PC is a master and writes each byte of data to the slave as it is typed in.

- 1.3.4. The Real-Time Bus Monitor cannot be activated from the Windows 95 version, it is recommended that you run the REALTIM1(2) application from DOS as detailed in the adapter manual.
- 1.4. Configuring I²C for Windows 95
 - 1.4.1. To configure the software package and the I²C Card, select *File Menu* and click on *Configure*. The parameters that you can change are as follows.
 - 1.4.1. Base Address
 - 1.4.1.1. This parameter is no longer used by WINI2C, its reference in this manual has been retained for user familiar with the version of WINI2C for Windows 3.x.
 - 1.4.1.2. The base address of the adapter is now determined by the location at which the device driver was registered.
 - 1.4.1.3. You must ensure that the driver is registered at the address as determined by the links on the adapter this is factory set to 310 hex. (See your adapter manual for details on configuring the adapter).
 - 1.4.2. Own Address
 - 1.4.2.1. This is the unique I²C address to which the adapter is to respond to in Slave mode. This forms the upper 7 bits of the 8 bit address, the lowest bit being the Read(1) or Write(0). This means that if the own address is 57 hex, the card will respond to a Write address of AE hex and Read address of AF hex.
 - 1.4.2.2. **Note** An own address of 0 (the 'general call' address) **must not** be used.
 - 1.4.3. Status Wait
 - 1.4.3.1. This is a period of time (in micro seconds) to wait for the required bus status. If this time-out expires the transfer is aborted.
 - 1.4.3.2. The default value (1000) is adequate for most systems and it is unlikely that this value will require altering.
 - 1.4.4. Clock Speed
 - 1.4.4.1. This is the bit rate speed (in Kilo Hertz) for the I²C serial bus when operating as a master. The settings available are 90, 45, 11 and 1.5kHz.
 - 1.4.5. Configuration Controls
 - 1.4.5.1. Clicking the OK button will configure the card with the values specified. Should configuration fail (see Trouble Shooting Hardware).
 - 1.4.5.2. Clicking Cancel will undo all changes made and restore the parameters to those last used.
 - 1.4.5.3. Clicking Default will load up the software default values and configure the adapter. The default values are:-

Own Address	57H
Status Wait	1000
Clock Speed	90
- 1.5. File menu commands
 - 1.5.1. New
 - 1.5.1.1. Selecting New clears the data displayed in the Data Table.
 - 1.5.2. Load
 - 1.5.2.1. This displays the normal Windows File Open dialog box. Select the file you wish to open and select OK. This opens an existing file and loads the information held in that file.

- 1.5.2.2. The parameters displayed in the transfer parameter frame are those relevant to the function selected (see Function Menu Commands) the data contained in the file is displayed in the data table.
- 1.5.2.3. The file name is displayed alongside the function selected at the top of the form.
- 1.5.2.4. When a file is loaded the adapter is re-configured using the parameters contained in the file.
- 1.5.2.5. **Note** The Status Wait is not changed when a file is loaded. This allows the data files to be shared between several I²C for Windows applications running with different hardware configurations.
- 1.5.3. Save
 - 1.5.3.1. Use Save to store the parameters to the file whose name is displayed alongside the function selected at the top of the form. The parameters are saved as text files which can be loaded up into any text editor, where they can be modified. The files follow a simple format see Changing Files Manually for details.
 - 1.5.3.2. All parameters, except the Status Wait are saved, in the named file.
 - 1.5.3.3. To save the data into a different file use SaveAs.
 - 1.5.3.4. If a file name has not been allocated then Save becomes SaveAs and requests a file name using the standard Windows File SaveAs Dialog box.
- 1.5.4. SaveAs
 - 1.5.4.1. Use SaveAs to save the parameters displayed in the parameter frame and the data when either a file name has not been entered (displayed alongside the function selected at the top of the form) or when you wish to save to a different file name.
 - 1.5.4.2. All parameters, except the Status Wait are saved, in the named file.
- 1.5.5. Print
 - 1.5.5.1. Use print to obtain a paper copy of the data file via the Windows default printer.
- 1.5.6. Configure
 - 1.5.6.1. Lets you specify set-up parameters for the adapter and the software package. See Configuring I²C for Windows for details of the set-up parameters.
- 1.5.7. Exit
 - 1.5.7.1. Exits the application.
- 1.6. Edit menu commands
 - 1.6.1. Clear Table
 - 1.6.1.1. The data displayed in the data table is cleared to 00.
 - 1.6.1.2. To edit the data displayed in the data table click on the data you wish to alter and then enter the new data into the dialog box shown.
 - 1.6.2. Set Table
 - 1.6.2.1. The data displayed in the data table is set to FF hex.
 - 1.6.2.2. To edit the data displayed in the data table click on the data you wish to alter and then enter the new data into the dialog box shown.
- 1.7. Function menu commands
 - 1.7.1. Master Read
 - 1.7.1.1. Sets up the software to run as a MASTER (the adapter located in the PC) reading from a SLAVE. The parameters required for this function are Slave Address, Offset Address, and the Number of Bytes to be read.

- 1.7.1.2. If the Slave does not require an offset as part of the protocol (See the component data sheet for details) click on the check box alongside the Offset Address, this will ensure that an offset is not transmitted. To re-enable the offset click on the check box.
- 1.7.1.3. If the offset is specified the following protocol is used for the transfer:-
 - Slave Write Address
 - Offset Address
 - Slave Read Address
 - Read Data
 - Stop
- 1.7.1.4. If the offset is not specified the following protocol is used for the transfer:-
 - Slave Read Address
 - Read Data
 - Stop
- 1.7.1.5. Select the protocol appropriate to the slave access you require.
- 1.7.1.6. To start the transfer click on Read.
- 1.7.2. Master Write
- 1.7.2.1. Sets up the software to run as a MASTER (the adapter located in the PC) writing to a SLAVE. The parameters required for this function are Slave Address, Offset Address, Block Size, and the Number of Bytes to be written.
- 1.7.2.2. If the Slave does not require an offset as part of the protocol (See the component data sheet for details) click on the check box alongside the Offset Address, this will ensure that an offset is not transmitted. To re-enable the offset click on the check box.
- 1.7.2.3. If the offset is specified the following protocol is used for the transfer:-
 - Slave Write Address
 - Offset Address
 - Write Data
 - Stop
- 1.7.2.4. If the offset is not specified the following protocol is used for the transfer:-
 - Slave Write Address
 - Write Data
 - Stop
- 1.7.2.5. Select the protocol appropriate to the slave access you require.
- 1.7.2.6. To start the transfer click on Write.
- 1.7.3. Memory Read
- 1.7.3.1. Sets up the software as a MASTER (the adapter located in the PC) reading from an memory. The Parameters required are Slave Address and the Size of the memory.
- 1.7.3.2. This function read the entire memory, if the memory consists of more than one page (e.g. 24C04 EEPROM) then, when the first page has been read, the software automatically increments the slave address given and continues reading the data from the second and subsequent pages.
- 1.7.3.3. To start the transfer click on Read.
- 1.7.4. Memory Write
- 1.7.4.1. Sets up the software as a MASTER (the adapter located in the PC) writing to an memory. The Parameters required are Slave Address, Block Size and the size of the memory.
- 1.7.4.2. This function writes the given data to the entire memory, if the memory consists of more than one page (e.g. 24C04 EEPROM) then, when the first page has been written, the software automatically increments the slave address given and continues writing the data to the second and subsequent pages.

- 1.7.4.3. To start the transfer click on Write.
- 1.7.5. Slave Sender
 - 1.7.5.1. Sets up the software as SLAVE (the adapter located in the PC) device being read by a MASTER (located elsewhere on the I²C bus). The Parameters required are Own Address and the Buffer Size.
 - 1.7.5.2. The Buffer Size simply defines the size of the data table into which data can be entered.
 - 1.7.5.3. If the master reads more data than the slave has been initialised for then the Slave Sender will continue to respond to the masters request for data by sending 00.
 - 1.7.5.4. The number of bytes transferred in the Progress Window is the total number of requests for data made by the master.
 - 1.7.5.5. The Slave Sender can detect and automatically respond to two protocols:-
 - 1.7.5.6. Protocol 1 (as seen from the Master)
 - Slave Write Address
 - Offset Address
 - Slave Read Address
 - Read Data
 - Stop
 - 1.7.5.7. This allows the master to access data from anywhere within the data table and as such the Slave Sender can be used to emulate an I²C memory.
 - 1.7.5.8. **Note** If the master sets the offset to zero then this protocol transfers the same data a protocol 2.
 - 1.7.5.9. Protocol 2 (as seen from the Master)
 - Slave Read Address
 - Read Data
 - Stop
 - 1.7.5.10. This allows the master to read data from the **start** of the data table and as such the Slave Sender can be used to emulate I²C devices other than memories.
 - 1.7.5.11. To start this function click on Send.
 - 1.7.6. Slave Receiver
 - 1.7.6.1. Sets up the software as SLAVE (the adapter located in the PC) device being written to by a MASTER (located elsewhere on the I²C bus). The Parameters required are Own Address and the Buffer Size.
 - 1.7.6.2. The Buffer Size simply defines the size of the data table into which data can be entered.
 - 1.7.6.3. If the master writes more data than the slave has been initialised for then the Slave Receiver will continue to respond to the masters data but the data will be discarded.
 - 1.7.6.4. The number of bytes transferred in the Progress Window is the total number of data transfers made by the master.
 - 1.7.6.5. The user can select between two protocols:-
 - 1.7.6.6. Protocol 1 (as seen from the Master)
 - Slave Write Address
 - Offset Address
 - Write Data
 - Stop
 - 1.7.6.7. This allows the master to write data to anywhere within the data table and as such the Slave Receiver can be used to emulate an I²C memory.

- 1.7.6.8. Protocol 2 (as seen from the Master)
 - Slave Write Address
 - Write Data
 - Stop
- 1.7.6.9. The data received using this protocol will always be placed at the start of the data table. This allows the Slave Receiver to emulate other I²C devices for example the TDA8444 Octuple 6-bit DAC where the first byte is an instruction.
- 1.7.6.10. **Note** The Slave Receiver simply receives the data it does not emulate what the I²C device would do with the data.
- 1.7.6.11. To start this function click on Receive.
- 1.7.7. Teletype
 - 1.7.7.1. Sets up the software as a MASTER (the adapter located in the PC) writing single bytes to a SLAVE. Each time the SEND button is clicked data will be written to the SLAVE, this will continue until the STOP button is clicked at which point the transfer between the MASTER and SLAVE will be terminated and the bus released. The parameters required for this function are Slave Address and Offset Address.
 - 1.7.7.2. If the offset is specified the following protocol is used for the transfer:-
 - Slave Write Address
 - Offset Address
 - Write Data
 - Stop
 - 1.7.7.3. If the offset is not specified the following protocol is used for the transfer:-
 - Slave Write Address
 - Write Data
 - Stop
 - 1.7.7.4. Select the protocol appropriate to the slave access you require.
 - 1.7.7.5. To start this function click on Start, this sends the Slave address and, if specified, the Offset Address.
- 1.8. Progress Window
 - 1.8.1. The progress window is displayed whenever an I²C transfer is executed, except Teletype and Monitor functions.
 - 1.8.2. The darker text indicates the part of the I²C transfer cycle currently being performed.
 - 1.8.3. The Operation column provides a simple description of the part of the I²C transfer cycle currently being performed, including the state of the acknowledge for the transfer.
 - 1.8.4. The Data column details the I²C address transmitted / received (NOTE slave functions also display the Offset address received).
 - 1.8.5. The I²C Status column details the status of the bus at the various parts of the I²C transfer cycle (see the adapter manual for status bit definitions).
 - 1.8.6. To stop a transfer before completion click ABORT to close this window once the transfer is complete click OK.
- 1.9. Parameters - Their meaning
 - 1.9.1. Own Address
 - 1.9.1.1. This is only available for Slave Send and Receive functions, as it is not relevant to any other function. The Own Address is a 7 bit address that this application will respond to when it addressed as a SLAVE by a MASTER else where on the bus. This **MUST** be a unique address on the bus.



1.9.1.2. An own address of 0 ("general call address") **MUST NOT** be used as the adapter will go into monitor mode.

1.9.2. Slave Address

1.9.2.1. This is only available for Master functions, The Slave Address is a 7 bit address uniquely identifying a device (the slave) that you will be reading or writing to. The application converts this into an eight bit address by adding the read / write bit as required.

1.9.3. Offset Address

1.9.3.1. This is only available for Master Read, Write and Teletype functions. The offset is a 8 bit address specifying where in the device you want to read from or write to.

1.9.3.2. For Memory functions this is always set to zero enabling the entire memory to be written to or read from.

1.9.3.3. If the Slave does not require an offset as part of the protocol (See the component data sheet for details) click on the check box alongside the Offset Address, this will ensure that an offset is not transmitted. To re-enable the offset click on the check box.

1.9.3.4. If the offset is specified following protocol is used for the transfer:-

Master Write \ Teletype		Master Read
Slave Write Address		Slave Write Address
Offset Address		Offset Address
Write Data		Slave Read Address
Stop		Read Data
		Stop

1.9.3.5. If the offset is not specified the following protocol is used for the transfer:-

Master Write \ Teletype		Master Read
Slave Write Address		Slave Read Address
Write Data		Read Data
Stop		Stop

1.9.3.6. Select the protocol appropriate to the slave access you require.

1.9.4. Block Size

1.9.4.1. This is only available for Master and Memory Write functions. The Block Size determines how many bytes are written to a device, before waiting for the device to be ready to accept the next block of bytes. This is useful for some devices e.g. EEPROMS, that have a maximum number of bytes which can be written before they can accept more data.

1.9.4.2. The application divides the data in the table into the correct block size and ensures that all data is sent.

1.9.5. Number of Bytes

1.9.5.1. This is the number of bytes to be read or written, for the Memory Read \ Write functions the entire memory is read, hence this parameter is specified via the Memory Size list.

1.9.6. Buffer Size

1.9.6.1. This parameter is used by the Slave Sender and Slave Receiver to determine how big the data table should be and consequently the maximum number of bytes the Master should read / write.

1.10. Changing Parameters

1.10.1. To change a parameter, simply click on it. You will then be prompted to enter the new value or edit the previous one.

1.11. Trouble Shooting Hardware

1.11.1. Should the configuration fail a warning message will be displayed. The probable cause is either the adapter is not located at the base address specified.

1. Check the link configuration of the adapter and also the addresses of other cards within your system then re-start the application and re-configure the adapter.
 2. See Configuring I²C for Windows for details of the set-up parameters.
- 1.11.2. Should a master function report that the slave is not responding the most probable causes are:-
1. The slave is not located at the slave address specified - check the address of the slave.
 2. The slave is incorrectly connected or not powered up - check external connections.
 3. The bus is not correctly terminated or there are no pull ups on the bus - check the termination and pull up in accordance with the adapter manual.
- 1.11.3. Should a slave function not receive data from a master the most probable causes are:-
1. The master is not addressing the slave - check master operations, check that the Slave Address is that which the master is attempting to access.
 2. The master and slave are not connected - check connections.
 3. The bus is not correctly terminated or there are no pull ups on the bus - check the termination and pull up in accordance with the adapter manual.
- 1.12. Trouble Shooting Software
- 1.12.1. Should an error be made during a user operation you will be warned and invited to enter a valid parameter. The invalid parameter will not be used.
- 1.12.2. If the software runs but no I²C transfers are possible check that the I²C driver has been registered. If an I²C adapter has not been registered use the regnt.exe on the source disc to register the driver. If the address at which the adapter has been register does not match base address selected in WINI²C adjust the base address accordingly.
- 1.12.3. There are no errata sheets appertaining to this issue of WINI²C. Should an untrapped error occur please provide our I²C Technical Support Department (see page 2 for contact details) with the following information:
- | | | |
|----|--|---|
| 1. | The model and serial number of you I ² C Adapter | |
| 2. | The type and speed of your PCs processor | e.g. 486DX2 66MHz |
| 3. | The operating system | e.g. Windows 95 Rev x |
| 4. | The issue of WINI ² C | e.g. Version 1.0 |
| 5. | The issue of the DLL | e.g. Version 2.0 |
| 6. | The configuration including all addresses | e.g. Base addr 310H
Own addr 57H
etc. |
| 7. | The Bus Status | e.g. Bus Free |
| 8. | A brief description of the problem and if possible a file containing the data. | |
- 1.12.4. Every effort will be made to provide answers to problems relating to this product.
- 1.13. Adding Comments to Data
- 1.13.1. The data displayed in the data table and hence stored in a file can have comments added.
- 1.13.2. To add a comment:-
1. Move the horizontal scroll bar to the right so that the comments column can be seen.
 2. Click on the row containing the data you wish to comment.
 3. Enter the comment, **NOTE** the application will automatically add the comment symbol ([) if you omit it.
- 1.13.3. If you wish to delete a comment:-
1. Move the horizontal scroll bar to the right so that the comments column can be seen.
 2. Click on the row containing the comment you wish to remove
 3. Select CANCEL at the Comment Editor.



4. Confirm the comment is to be deleted.

1.13.4. To view the data again move the horizontal scroll bar to the left.

1.13.5. If the data has a comment associated with it and you click the data table to alter the value, the comment will be displayed in the entry dialog box.

1.14. Changing Files Manually

1.14.1. All files saved using the Save command, can be edited manually by loading them up into any text editor. The file can be in hexadecimal, binary or decimal format. The layout of the file is quite simple. First you specify the parameter heading, described in the table below, then on the next line the you specify it's value.

Heading	Description

[Clock...]	Specifies that the clock speed is on the next line
[own]	Specifies that the own address is on the next line
[slave]	Specifies that the slave address is on the next line
[offset]	Specifies that the offset address is on the next line
[no.bytes]	Specifies that the number bytes is on the next line
[data]	Specifies that the data bytes are on the next multiple lines The data bytes must be comma separated. Comment can be inserted into the data area as follows:- ,[type your comment here. Comments should always occur on the end of the line.

1.14.2. This is a typical data file

[Clock Speed 0 = 90kHz 1 = 45KHz 2 = 11kHz 3 = 1.5kHz]

```

0
[own]
57
[slave]
50
[offset]
00
[no.bytes]
40
[data]
20      ,[ comment this data if you wish
00      ,[ note spaces or tabs can be used to
86      ,[separate the comment from the data
00
00      ,[you dont have to comment all lines

```

THE MOST COMMONLY ASKED I2C QUESTIONS

2.1. General Questions

Question **Will my adapter work in a Pentium PC?**

Answer Yes and there is no need to alter LK 4 from the position in which your adapter was shipped.

Question **Will my adapter run I2C clock speeds greater than 90KHz?**

Answer At the moment your adapter is limited by the Bus Controller chip fitted, to a maximum of 90KHz as a master and 100KHz as a slave.

Question **Will my adapter work under Windows NT4* or Windows 95*?**

Answer The adapter will work under both these systems without problems, but the software supplied with the adapter is 16 bit only. Windows NT and Windows 95 DLLs are available please contact our sales team for further information.
The bus monitors are best run from DOS (not Windows DOS Shell) as this minimises the risk of missing part of a transfer.

* All trade marks acknowledged

Question **How can I find out if the adapter is clashing with another card in my PC?**

Answer If your adapter is located at a base address of 310H (as shipped) then run the bus monitor (REALTIM1) application supplied following the instructions in the adapter manual. With no activity on the bus the monitor should display a fix 'A', stop the monitor and edit the file produced if the file states that the bus was free and the status was 81H then your adapter is indeed located at 310H and there are no clashes within the PC.
If your adapter is at 312H then repeat the above using the REALTIM2 bus monitor.

Question **I get corrupted transfers why is this?**

Answer The most likely reason for corrupted transfers is either incorrect bus termination or excessive capacitance - see the manual for details.

Question **Do you have software to talk to my.....?**

Answer Unfortunately there are too many I2C devices for us to be able to offer complete solutions - although we can supply a windows based application called WINI2C which is designed for those just starting I2C or wishing to perform simple I2C tasks, please contact our sales team or look on our web site, www.calibreuk.com for further information.

Question **I am trying to read from a device, the first time my software works fine but when I try again I can't get anything what's wrong?**

Answer Please check that you are changing the value of Setnack in accordance with the manual, it is likely that you have not made Setnack 1 for the last **AND** last but one bytes being read.

2.2. DOS Software Questions

Question **My I2C adapter locks up with a constant status - why?**

Answer If you are using either the 'C' or Basic library functions supplied with the adapter on a fast PC it is possible that the PC is polling the status register too quickly, the simplest way to prevent this is to add a small delay prior to reading the status in the getstatus routine. Alternatively use the windows DLL supplied as these automatically allow for speed of the PC at run-time.

2.3. WINDOWS 95 and NT Questions

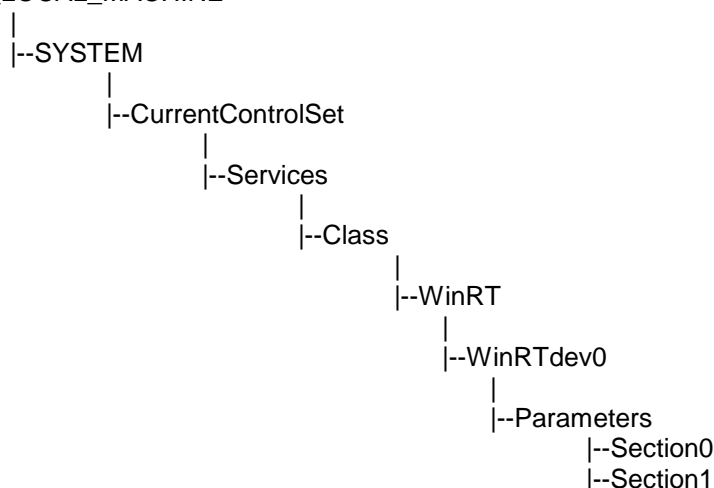
Question **My software cannot find the adapter. Your Windows software reports that it cannot configure the adapter. Why is this?**

Answer Have you registered the device driver as detailed in the software manual? If so check that the address links (see adapter manual for details) are correct for the location at which you registered the driver.

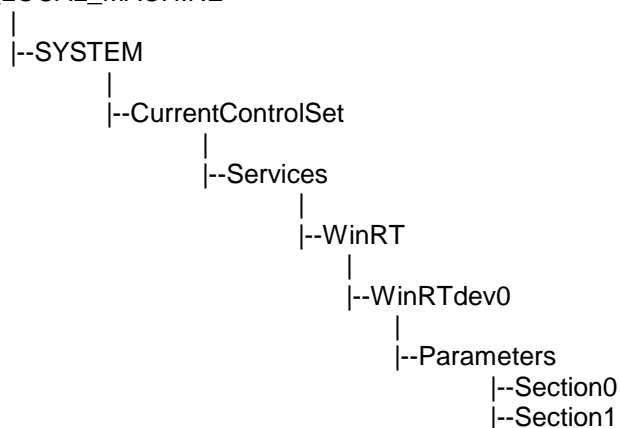
Question **I think I have registered the driver how can I find out if I have?**

Answer You need to inspect the registry as follows

Windows 95 START - Run regedit
HKEY_LOCAL_MACHINE



Windows NT START - Run regedit
HKEY_LOCAL_MACHINE



Question **I am using your Windows 95 / NT DLL and I am always getting a time out error code. Why?**

Answer Check the Syntax of the setup function, this problem is most usually caused by swapping the Status Wait and Sclk parameters.
The correct syntax is i2cstatus = setup (baseaddress, ownaddress, sclk, statuswait)

Question **I have read the manual and still cannot get the communications to run. What do I do next?**

Answer Check that you have fully implemented the protocol between the adapter and the other I2C devices see the device manufacturers data sheet for details.
Check that the software you have written is logically and syntactically correct - this is probably the most common cause of software faults we have to deal with.



Send us the following details:-

- 1)The link settings of the adapter.
- 2)A sketch of the relevant I2C hardware including the location of bus termination.
- 3)The type and speed of processor within your PC and which operating system, you are running.
- 4)Brief software listings, or which Calibre software you are running.
- 5)The serial number of your I2C adapter, or when you purchased it.

PLEASE EMAIL YOUR QUERY TO: techsupport@calibreuk.com

OR FAX YOUR QUERY TO: 44-1274-730960

We will endeavour to help you.