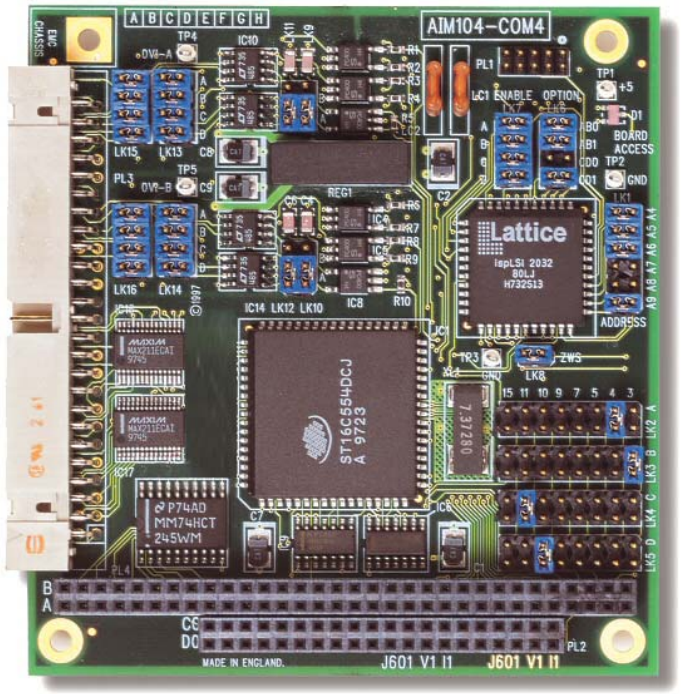
AIM104-COM4

4 Channel RS232/RS422/RS485
PC/104 Board

Technical Manual



EuroTech Ltd

A MEMBER OF THE EUROTECH GROUP

Disclaimer

The information in this manual has been carefully checked and is believed to be accurate. Eurotech Ltd assumes no responsibility for any infringements of patents or other rights of third parties, which may result from its use.

Eurotech Ltd assumes no responsibility for any inaccuracies that may be contained in this document. Eurotech Ltd makes no commitment to update or keep current the information contained in this manual.

Eurotech Ltd reserves the right to make improvements to this document and/or product at any time and without notice.

Warranty

This product is supplied with a 3 year limited warranty. The product warranty covers failure of any Eurotech Ltd manufactured product caused by manufacturing defects. The warranty on all third party manufactured products utilised by Eurotech Ltd is limited to 1 year. Eurotech Ltd will make all reasonable effort to repair the product or replace it with an identical variant. Eurotech Ltd reserves the right to replace the returned product with an alternative variant or an equivalent fit, form and functional product. Delivery charges will apply to all returned products. Please check www.eurotech-ltd.co.uk for information about Product Return Forms.

Trademarks

Windows NT, Windows 2000, and Windows XP are registered trademarks of the Microsoft Corporation.
All other trademarks recognised**.**

Revision History

Manual	*PCB*	*Date*	*Comments*
Issue A	V1.2	February 16th, 1998	First full release of manual.
Issue B	V1.2	July 7th, 1999	[ECO2823]
Issue C	V1.2	September 25th, 2006	Revised format and content.
Issue D	V1.2	December 21st, 2007	Revised format and content.

For contact details, see page 28.

ISO 9001
FM12961

Contents

Introduction 4
Handling your board safely 5
About this manual 6
Conventions 6
What items are provided? 7
Getting started 7
Unpacking and connecting up 7
Description 8
I/O map 9
Interrupts 11
Links 12
Using the differential communications channels – A and B 18
Address and interrupt selection 18
Communications mode 19
Software considerations 22
RS485 flow diagrams 23
Using the RS232 channels – C and D 25
Address and interrupt selection 25
Connector (PL3) pin assignments 26
CAB AIM104-COM4 27
Appendix A – Contacting Eurotech Ltd 28
Appendix B – Example code for DOS compatible platform 29
Appendix C – Notes on using the 16C550 UART 35
Appendix D – Software library 37
Index 39

Introduction

The AIM104-COM4 is an 8-bit or 16-bit PC/104 module providing up to four 16C550 serial asynchronous communications channels: two of optically isolated RS422 or RS485 and two of RS232.

Features

- Four 16C550 compatible UART channels (referred to as channels A, B, C, and D).
- Two opto-isolated RS422/485 ports (channels A and B).
- Two full feature RS232 ports (channels C and D).
- Baud rates up to 115.2Kbaud supported.
- +5V only operation.
- Configurable as standard PC AT COM1-4 or anywhere in I/O address range 000-3FFh.
- Each channel implements link selectable IRQs in the range 3, 4, 5, 7, 9, 10, 11, 15.

Each channel must use a separate IRQ line if interrupt is required. IRQ lines cannot be shared between channels.

- PC/104 16-bit interface. However, it can be used in the 8-bit slot if IRQ10, 11 and 15 are not required.
- Zero wait state bus operation capability.
- Board access LED.
- Isolation:
  - channel A to channel B > 100V DC
  - channels A or B to channels C or D > 100V DC
  - channels A or B to PC/104 ground > 100V DC
  - channels A or B to chassis ground > 100V DC
- Power requirements: 220mA (typical) at +5V DC.
- Temperature range: -20°C to +70°C operating.
  -40°C to +125°C storage.
- MTBF 226,400 hours (using MIL-HDBK-217F generic failure rates at ground benign).

Handling your board safely

Anti-static handling

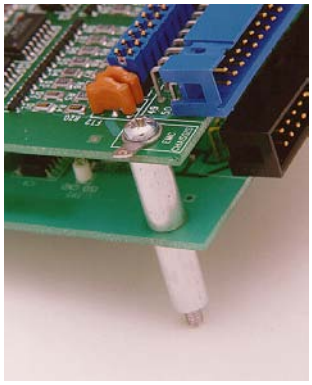
This board contains CMOS devices. These could be damaged in the event of static electricity being discharged through them. Observe anti-static precautions at all times when handling circuit boards. This includes storing boards in appropriate anti-static packaging and wearing a wrist strap when handling them.

Electromagnetic compatibility (EMC)

The AIM104-COM4 is classified as a 'component' with regard to the European Community EMC regulations and it is the user's responsibility to ensure that systems using the board are compliant with the appropriate EMC standards.

The opto-isolation provides a good barrier for noise emissions generated by the high frequency host PC/104 system. The AIM104-COM4 includes additional filter components on-board to minimise the emissions of high frequency noise. Because of this, the earth tab supplied with the module must be connected to the chassis of the system by a good earth wire.

If the electronic system requires input protection against high voltage transients (to meet CE requirements), it is recommended that an external interface board is located at the point where the external wiring enters the electronic system enclosure:



Packaging

Should a board need to be returned to Eurotech Ltd, please ensure that it is adequately packed, preferably in the original packing material.

About this manual

This manual describes the operation and use of the AIM104-COM4 PC/104 module. It is both a reference and user manual and includes information about all aspects of the module.

Conventions

Symbols

The following symbols are used in this guide:

Symbol	Explanation
	Note - information that requires your attention.
	Tip - a handy hint that may provide a useful alternative or save time.
	Caution – proceeding with a course of action may damage your equipment or result in loss of data.
	Link is not fitted.
	Link is fitted.

Terminology

To prevent confusion with the standard PC AT nomenclature of COM1-4, this manual refers to the four channels on the AIM104-COM4 as channels A-D. Where this manual specifies COM1-4, this refers to the standard PC AT communications ports.

What items are provided?

The AIM104-COM4 is supplied with the following items:

- The AIM104-COM4 board.
- A short ribbon cable (CAB AIM104-COM4), which is used to break out from the 50-way, boxed header on the AIM104-COM4 to 4 male 9-way D-sub connectors.
- Mounting kit.

Getting started

Unpacking and connecting up

To begin using the AIM104-COM4 board, follow these steps:

1. Power down your computer.
2. Install the board in a spare PCI slot.
3. Power up your computer and install the appropriate driver available from our website www.arcom.com/support/downloads.

Description

The AIM104-COM4 uses a quad 16C550 UART to provide the four standard PC AT communications channels, which are supported by a wide range of third party software and standard operating systems.

Two channels on the AIM104-COM4, channels A and B, are independently electrically isolated from the PC/104 bus, and can be configured for either RS422-type communications or RS485-type communications. The other two channels, C and D, are configured as RS232, supporting all the signals found on a PC AT 9-pin RS232 port.

Using a flexible addressing system, the four channels can be mapped into the standard PC AT communications channel addresses for COM1-4. Alternatively, pairs of channels can be mapped into any I/O location from 000-3FFh. See I/O map, page 9, for more details of address mapping.

Each channel has its own interrupt line that can be linked to the PC/104 interrupts IRQ3-5, 7, 9-11, and 15.

The communications channels cannot share IRQ lines. Each channel must use a separate IRQ if interrupt is required.

The AIM104-COM4 has a 16-bit PC/104 connector but it only utilises interrupt lines IRQ10, 11, and 15 on the J2 connector. This means that the board can also be used in an 8-bit PC/104 slot, providing that these interrupt lines are not required.

If the PC/104 CPU board supports it, a further link on the AIM104-COM4 allows zero wait state operation of the PC/104 bus.

I/O map

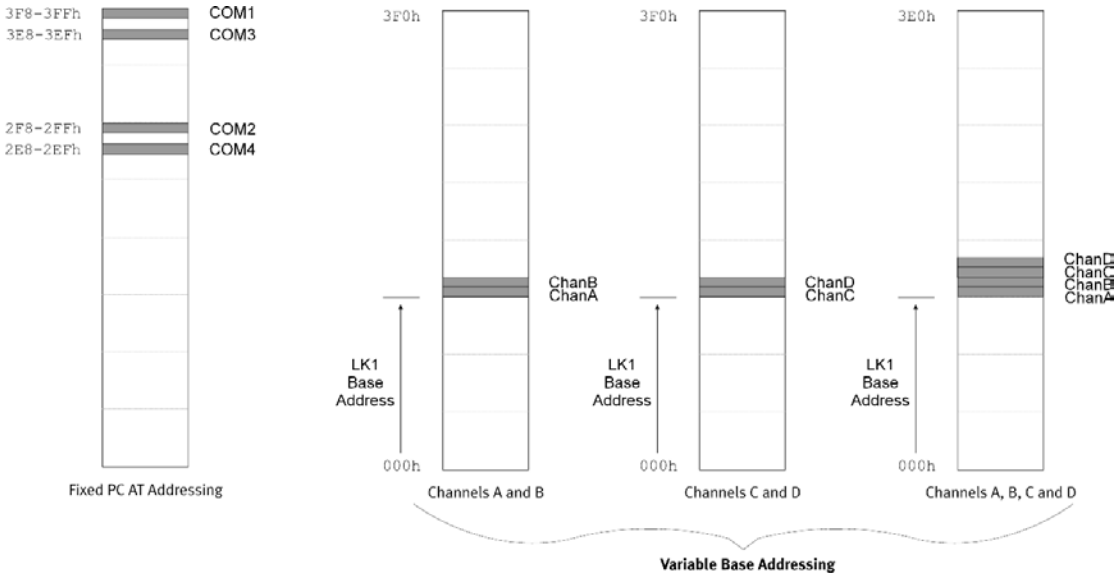
The AIM104-COM4 has a flexible addressing system, which means that the channels A, B, C, and D can be addressed either:

- At the standard PC locations for COM1, COM2, COM3, and COM4.

-or-

- Using the user base address link, LK1.

This is shown in the following diagram:



Each channel occupies 8 bytes of I/O space, with the following registers defined:

General register set (DLAB [ICR:7] clear)

Register offset	Mnemonic	Read register	Write register
0	RHR/THR	Receive holding register	Transmit holding register
1	IER	Interrupt enable register	Interrupt enable register
2	ISR/FCR	Interrupt status register	FIFO control register
3	LCR	Line control register	Line control register
4	MCR	Modem control register	Modem control register
5	LSR	Line status register	Line status register
6	MSR	Modem status register	Modem status register
7	SCR	Scratchpad register	Scratchpad register

Baud rate register (DLAB [ICR:7] set)

Register offset	Mnemonic	Read register	Write register
0	DLL	LSB of divisor latch	LSB of divisor latch
1	DLM	MSB of divisor latch	MSB of divisor latch

If you are planning to write your own low level software for the AIM104-COM4, please refer to the datasheet for the ST16C554 device, supplied in Adobe Acrobat format (PDF) on the software libraries disk. You can also use the example source code supplied on the utilities disk as a reference design.

Interrupts

Four links, LK2 to LK5, define the interrupt mapping for each of the channels A to D respectively. Each channel can be connected to line IRQ3, 4, 5, 7, 9, 10, 11, or 15.

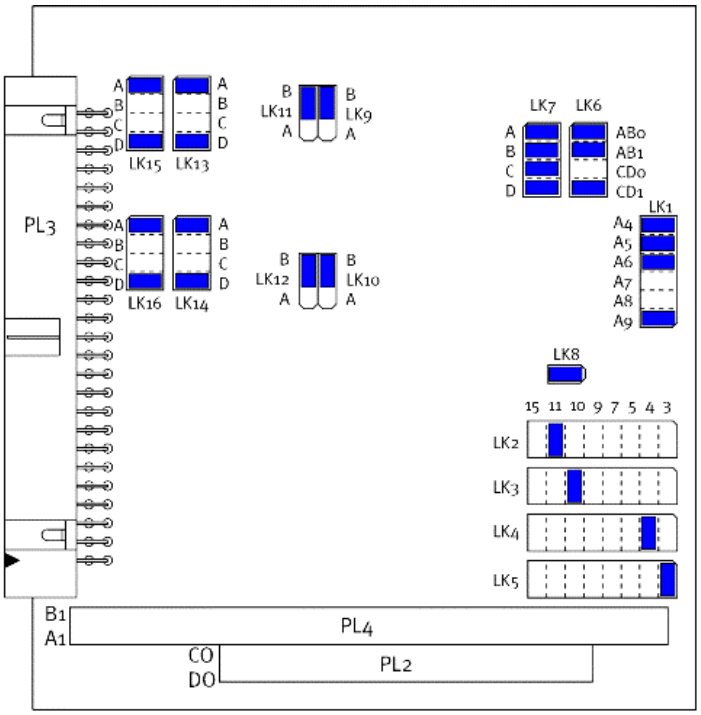
IRQ10, 11, and 15 are only available when the AIM104-COM4 is used in a 16-bit PC/104 stack.

IRQ lines cannot be shared between the channels. Each channel that uses an interrupt must use a unique IRQ line.

Links

The following diagram illustrate the link positions:

Default link position diagram



Default shipment configuration

The default shipment configuration is shown in the table below:

Channel	PC/AT COM	I/O address	IRQ number	Comms type	Terminator
A	n/a	180h	11	RS485	Pulled inactive
B	n/a	188h	10	RS485	Pulled inactive
C	COM3	3E8h	4	RS232	n/a
D	COM4	2E8h	3	RS232	n/a

LK1 – Address link

LK1 is the address decode link. Use this link to set the user base address you require.

If two channels are configured to use the user address then this link defines the base address of a 16 byte access window in the I/O space, and can be set on any 16 byte boundary from 000h to 3F0h.

However, if four channels are configured to use the user address, link A4 is ignored. LK1 defines a 32 byte window on any 32 byte boundary from 000h to 3E0h. The default LK1 setting is a base address of 180h.

For more details, see the section I/O map, page 9.

LK2 to LK5 - Interrupt links

LK2-LK5 are the interrupt select links for channels A to D respectively. The IRQ lines available are IRQ3, 4, 5, 7, 9, 10, 11, and 15.

It is only possible to use IRQ lines 10, 11, and 15 when the AIM104-COM4 is inserted in a 16-bit PC/104 stack. The default links are: LK2 - position 11; LK3 - position 10; LK4 - position 4; LK5 - position 3.

For more details, see the section Interrupts, page 11.

LK6 - Option link

The option link, LK6, defines the addressing mode for channels A and B, and channels C and D, as follows:

CD1	CD0	AB1	AB0
X	X	Omit	Omit
X	X	Omit	Fit
X	X	Fit	Omit
X	X	Fit (default)	Fit (default)
Omit	Omit	X	X
Omit	Fit	X	X
Fit (default)	Omit (default)	X	X
Fit	Fit	X	X

Channel address mapping

Channel A	Channel B	Channel C	Channel D
COM1	COM2	X	X
COM2	COM3	X	X
COM3	COM4	X	X
Variable base address		X	X
X	X	COM1	COM2
X	X	COM2	COM3
X	X	COM3	COM4
X	X	Variable base address	

In addition, you can use LK7 to prevent any channel from being addressed by removing link A, B, C, or D, as appropriate.

See the section I/O map, page 9, for more information.

LK7 - Channel enable link

Use LK7 to enable each of the channels. You should fit a link to enable the addressing of a channel, or omit it to disable. The default setting is all channels enabled.

LK8 - Zero wait state link

Fit LK8 to the AIM104-COM4 to assert the PC/104 ZWS signal. On CPU boards that support this signal, this results in faster bus accesses to the board. The default is the link fitted.

Some CPU boards cannot support this function.

LK9 and LK11 - Channel A mode links

Details of these links are shown in the table below:

LK9	LK11	Mode	Comment
A	A	RS422 point-point	Transmit on A-TXB, receive on A-RX
A	B	RS422 multi-drop	Transmit on A-TXB, receive on A-RX
B (default)	B (default)	RS485	Transmit and receive on A-TXB

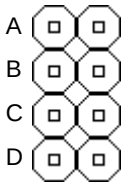
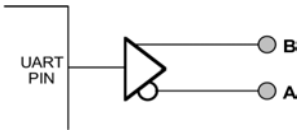
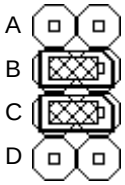
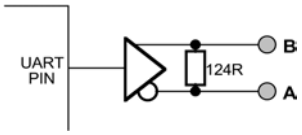
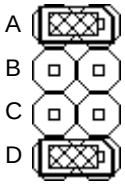
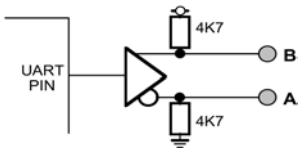
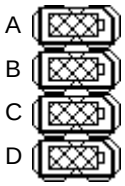
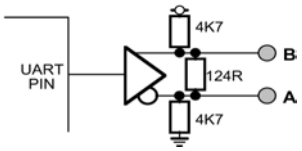
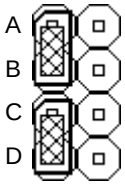
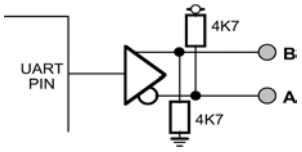
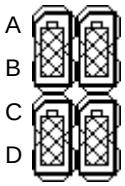
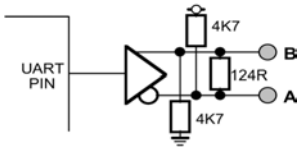
LK10 and LK12 - Channel B mode links

Details of these links are shown in the table below:

LK10	LK12	Mode	Comment
A	A	RS422 point-point	Transmit on B-TXB, receive on B-RX
A	B	RS422 multi-drop	Transmit on B-TXB, receive on B-RX
B (default)	B (default)	RS485	Transmit and receive on B-TXB

LK13 to LK15 – Differential pair termination links

Each differential pair on channels A and B can be terminated in six ways. These are illustrated in the following table:

Link configuration	Circuit configuration	Description
		**Unterminated** Line driver connects directly to the differential pair.
		**Terminated** Differential pair is terminated with 124R resistance between the signal lines.
		**Pulled inactive (default)** Differential pair is pulled apart to mimic no character being transmitted from the UART.
		**Pulled inactive and terminated** Differential pair is pulled apart to mimic a UART sending a break condition.
		**Pulled active** Differential pair is pulled apart to mimic a UART sending a break condition.
		**Pulled active and terminated** Differential pair is terminated with 124R and is pulled apart to mimic a break condition.

Optical isolation is not shown on the above diagrams to aid clarity.

The following table shows the relationship between the links and the differential pairs:

Link	Channel	RS422 pair	RS485 pair	Signal names
LK13	A	Receive	n/a	A-RXA, A-RXB
LK14	B	Receive	n/a	B-RXA, B-RXB
LK15	A	Transmit	Bi-directional	A-TXBA, A-TXBB
LK16	B	Transmit	Bi-directional	B-TXBA, B-TXBB

Using the differential communications channels – A and B

You must configure channels A and B on the AIM104-COM4 appropriately before you use them.

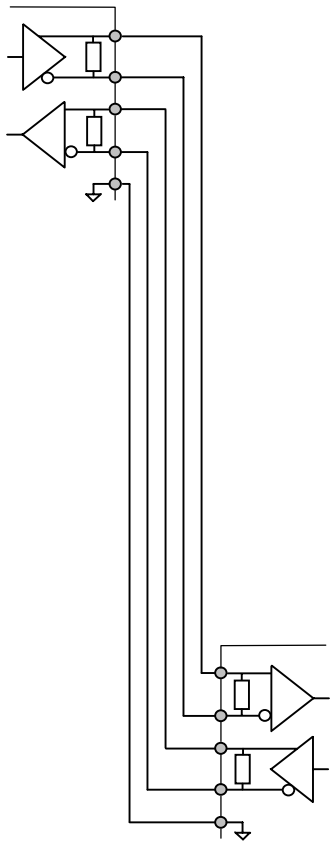
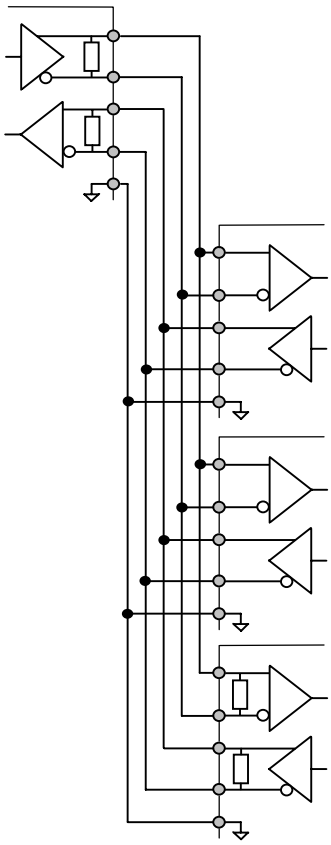
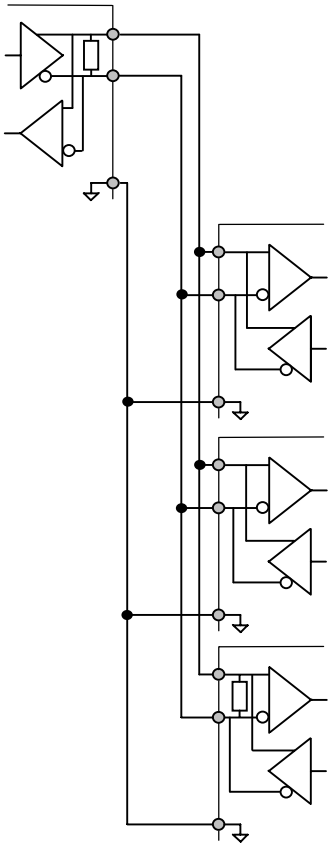
Address and interrupt selection

First, you should configure the I/O address of channels A and B, together with the interrupt line for each channel. Refer to the documentation on your CPU card to establish spare I/O and interrupt channels that you can use, then see the following sections of this guide to configure links LK1-LK7:

- I/O map, page 9.
- Interrupts, page 11.
- Links, page 12.

Communications mode

The AIM104-COM4 supports both RS422 and RS485 type differential communications modes. The RS422 mode of operation can function as either point to point or multi-drop. The differences between each of the configurations are illustrated below:

RS422 Point-point	RS422 Multi-drop	RS485
		
Number of wires 5	**Number of wires** 5	**Number of wires** 3
Transmitters enabled always	**Transmitters enabled** active RTS	**Transmitters enabled** active RTS
Receivers enabled always	**Receivers enabled** always	**Receivers enabled** inactive RTS
Duplex mode full	**Duplex mode** full	**Duplex mode** half
LK9 (LK10) position A	**LK9 (LK10)** position A	**LK9 (LK10)** position B
LK11 (LK12) position A	**LK11 (LK12)** position B	**LK11 (LK12)** position B

LK9 and LK11 are the mode links for channel A; LK10 and LK12 (shown in brackets in the above table) are for channel B.

Selecting the appropriate communication mode

If you are designing your own systems implementation from scratch, you can select the most appropriate communications type to act as standard for your system. The types available are explained below:

- RS422 point-point is a good way of connecting two pieces of equipment together because the differential communications system provides good noise immunity and the full duplex gives high speed.

- RS422 multi-drop is suitable for systems where one piece of equipment is controlling a number of other pieces of equipment in a master/slave type relationship. The full duplex nature of the system enables high speed communication.

The software must ensure that no two transmitters are driving the bus at once. The line drivers have protection circuitry in them to prevent damage occurring, but communications data will be lost.

- RS485 is also a multi-drop configuration. It is ideal for peer-peer networking or for systems where the cabling cost is large (so there are advantages to be gained with a three wire system).

As for RS422, the software must ensure that only one transmitter is driving the communications pair at any one time.

Configure links LK9-LK12 for the differential communications mode you require, for both channels A and B.

Line terminations

You should terminate differential communications pairs in order to prevent reflections from either end of the communications cables. Terminate each pair at each end of the wire run using a resistance that matches the impedance of the cable.

The AIM104-COM4 can use links LK13-LK15 to terminate each of the differential pairs on channels A and B. Two types of termination network can be configured: line matching terminators and pull-apart resistors. These are described in the following sections.

Line matching terminators

124 Ohm resistors can be linked across each communications pair (see Links, page 12). This is a fair match with twisted pair wiring. You should only configure these on boards at either end of the wiring run.

Pull-apart resistors

The AIM104-COM4 can also pull each line of a communications pair to either +5V or 0V through a 4K7 resistor. This facility is primarily intended for use in RS422 multi-drop or RS485 systems, to ensure that the communications pair floats to a known state when no transmitters are active.

Normally, a communications pair is pulled apart in the inactive state. This mimics the condition where no character is being transmitted from the UART. It is, however, possible for the AIM104-COM4 to pull the communications pair in either direction.

Take care when using the pull-apart resistors in systems with multiple pieces of equipment on the same pair, as each fitted pull-apart network will load the line driver devices. Reduce the number of fitted resistor networks to prevent this type of problem.

If there are no pull-apart resistor networks fitted anywhere on a multi-drop communications pair, the software may need to wait in excess of one character transmission time after asserting the RTS signal before loading the first character into the UART transmit buffer. This gives the receiving equipment time to clear any error character caused by a rogue start bit when the line driver was enabled at the transmitter.

Some UARTS (including those on the AIM104-COM4) have circuitry that prevents false start bit triggering, so this scenario may not occur with certain equipment.

Software considerations

No special consideration within the software is required for RS422 point-point, although, of course, hardware handshaking is not possible. If flow control is required, you must implement it with software such as the XON/XOFF protocol.

With the RS422 multi-drop and RS485 configurations, careful control of the RTS signal from the UART that is used to enable the transmitter on the differential pair is required.

RTS should be asserted prior to the beginning of the first start bit of the first character transmitted from the UART, and should be disabled after the last stop bit of the last character has actually been transmitted from the UART.

The opto-isolation on the AIM104-COM4 has a fixed delay of approximately 2µs between RTS becoming active on the UART and the line driver switching on. An RC network attached to the CTS pin of the UART has been configured to mimic this time delay. Therefore, after asserting RTS, the software can poll CTS becoming active before loading the first character into the UART transmit buffer.

After a character has been transmitted on the UART, it is possible to monitor the TEMT (Transmitter Empty) bit of the Line Status register. It is only valid to negate the RTS signal after this has become asserted.

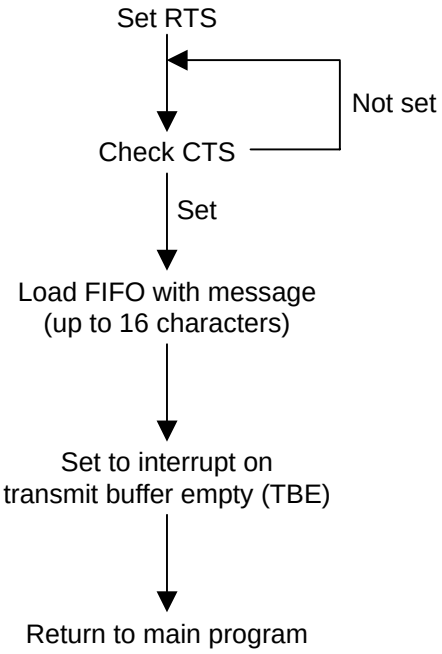
Reception of a Transmit Buffer Empty (TBE) interrupt is not a valid condition for negating RTS. The output shift register of the UART is still active until the TEMT bit is set.

If running at 9600 baud, the time between the TBE interrupt and the TEMT bit becoming set is approximately 1ms. Even at 115.2Kbaud, the time is around 90 µs. Therefore, when running interrupt driven RS485 (or RS422 multi-drop) communications, it is suggested that timer driven interrupts are used to avoid unacceptable delays within your ISR whilst it polls TEMT. You should set the timer interrupt to interrupt after a period of 10 x the transmit bit period (if 8,n,1 comms), initialised on receipt of the TBE interrupt.

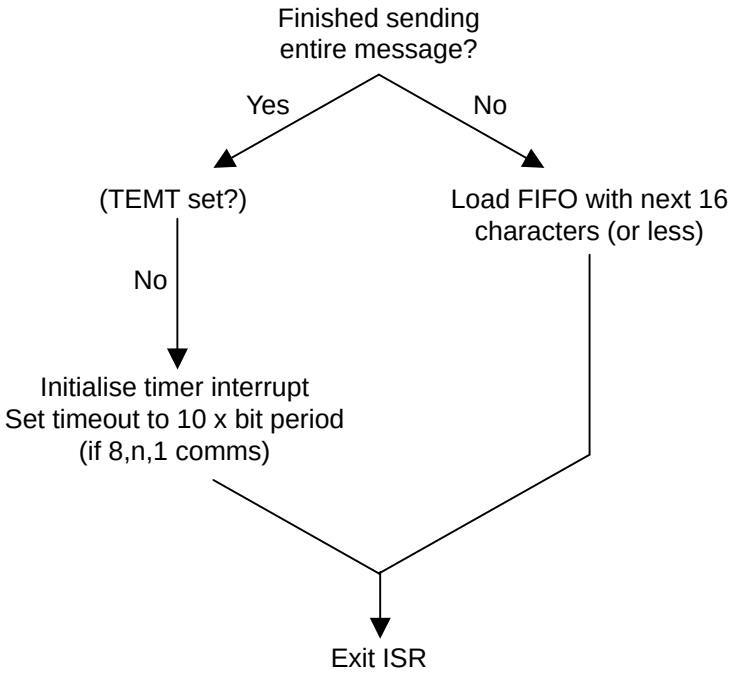
RS485 flow diagrams

This section contains flow diagrams for RS485.

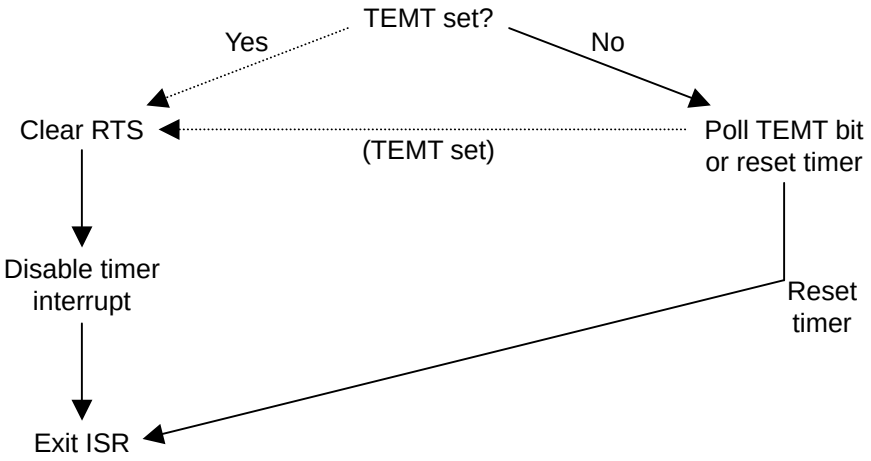
Transmit message



Transmit buffer empty (TBE) interrupt service routine (ISR)



Timer interrupt (ISR)



Using the RS232 channels – C and D

The RS232 channels on the AIM104-COM4 are of a standard implementation providing for all the standard signals available on a nine-way RS232 interface.

Address and interrupt selection

First you must configure the I/O address of channels C and D, together with the interrupt line for each channel. Refer to the documentation for your CPU card to establish spare I/O and interrupt channels that you can use, then see the following sections of this guide to configure links LK1-LK7:

- I/O map, page 9.
- Interrupts, page 11.
- Links, page 12.

Connector (PL3) pin assignments

The following table shows pin assignments for the connector PL3:

Channel	Channel	Pin	Pin	Pin
	Data carrier detect (D-DCD)	**1**	**2**	(D-DSR) Data set ready
	Received data (D-RXD)	**3**	**4**	(D-RTS) Request to send
D	Transmitted data (D-TXD)	**5**	**6**	(D-CTS) Clear to send
	Data terminal ready (D-DTR)	**7**	**8**	(D-RI) Ring indicator
	RS232 signal ground (GND)	**9**	**10**	(CHASSIS) Screen or shield
	Data carrier detect (C-DCD)	**11**	**12**	(C-DSR) Data set ready
	Received data (C-RXD)	**13**	**14**	(C-RTS) Request to send
C	Transmitted data (C-TXD)	**15**	**16**	(C-CTS) Clear to send
	Data terminal ready (C-DTR)	**17**	**18**	(C-RI) Ring indicator
	RS232 signal ground (GND)	**19**	**20**	(CHASSIS) Screen or shield
	RS232 signal ground	**21**	**22**	No connect
	No connect	**23**	**24**	No connect
	No connect	**25**	**26**	No connect
	No connect	**27**	**28**	No connect
	RS232 referenced +5V	**29**	**30**	RS232 referenced +5V
	Isolated channel B +5V	**31**	**32**	No connect
	Isolated channel B ground (GND)	**33**	**34**	(GND) Isolated channel B ground
B	RS422 Tx / RS485 line B (B-TXBB)	**35**	**36**	(B-TXBA) RS422 Tx / RS485 line A
	RS422 Rx line B (B-RXB)	**37**	**38**	(B-RXA) RS422 Rx line A
	Isolated chan B res gnd (RGND)	**39**	**40**	(CHASSIS) Screen or shield
	Isolated channel A +5V	**41**	**42**	No connect
	Isolated channel A ground (GND)	**43**	**44**	(GND) Isolated channel A ground
A	RS422 Tx / RS485 line A (A-TXBB)	**45**	**46**	(A-TXBA) RS422 Tx / RS485 line A
	RS422 Rx line A (A-RXB)	**47**	**48**	(A-RXA) RS422 Rx line A
	Isolated chan A res gnd (RGND)	**49**	**50**	(CHASSIS) Screen or shield

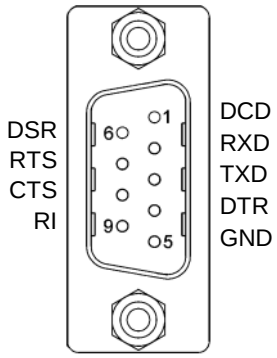
CAB AIM104-COM4

PL3 is wired so that it can be easily split into 9 way male D-sub connectors with standard pinouts for communications channels. Supplied with the AIM104-COM4 is an example cable that you can use for initial product development and as a template for making further cables. The CAB AIM104-COM4 is wired as follows:

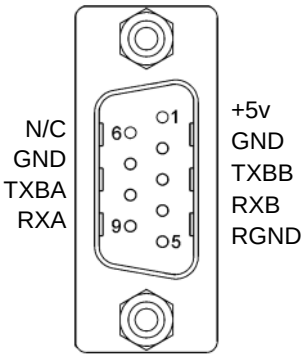
50 way IDC ribbon header	Male 9 way IDC D-sub	Channel	Pinout
Pins 1-9	Pins 1-9	D	PC/AT RS232 port
Pins 11-19	Pins 1-9	C	PC/AT RS232 port
Pins 31-39	Pins 1-9	B	Eurotech Ltd SERT-485 pinout
Pins 41-49	Pins 1-9	A	Eurotech Ltd SERT-485 pinout

The pinouts are illustrated in the diagrams below:

PC/AT RS232 port pinout



Eurotech Ltd SERT-485 pinout



Appendix A – Contacting Eurotech Ltd

Eurotech Ltd sales

Eurotech Ltd's sales team is always available to assist you in choosing the board that best meets your requirements.

Eurotech Ltd
3 Clifton Court
Cambridge
CB1 7BN
UK

Tel: +44 (0)1223 403410
Fax: +44 (0)1223 410457
Email: sales@eurotech-ltd.co.uk

Comprehensive information about our products is also available at our web site: www.eurotech-ltd.co.uk.

While Eurotech Ltd's sales team can assist you in making your decision, the final choice of boards or systems is solely and wholly the responsibility of the buyer. Eurotech Ltd's entire liability in respect of the boards or systems is as set out in Eurotech Ltd's standard terms and conditions of sale. If you intend to write your own low level software, you can start with the source code on the disk supplied. This is example code only to illustrate use on Eurotech Ltd's products. It has not been commercially tested. No warranty is made in respect of this code and Eurotech Ltd shall incur no liability whatsoever or howsoever arising from any use made of the code.

Eurotech Ltd technical support

Eurotech Ltd has a team of dedicated technical support engineers available to provide a quick response to your technical queries.

Tel: +44 (0)1223 412428
Fax: +44 (0)1223 410457
Email: support@eurotech-ltd.co.uk

Eurotech Group

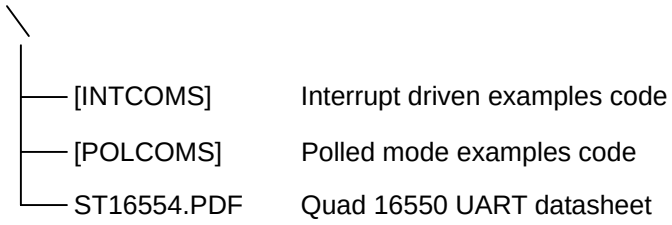
Eurotech Ltd is a subsidiary of Eurotech Group. For further details see www.eurotech.com

Appendix B – Example code for DOS compatible platform

A utilities disk is supplied with the AIM104-COM4 containing C source examples. These illustrate the setup of the COM4 UARTs for both interrupt driven and polled mode Rs232 serial communications.

Disk contents

The utilities disk contains the following directory structure:



INT_COMS.C

This Borland C source file illustrates the setup of an 8250 compatible UART for serial communications over an RS232 link (i.e. 8250, 16450, 16550).

Standard I/O and IRQ locations for COM1 and COM2 are used by default, but these can be changed easily in the source code (ref. ComPort and ComPortIrq variables).

The UART is set up with interrupt driven (buffered) Rx and direct (unbuffered) Tx.

The C source file will compile under Borland C++ 3.1 or Borland C++ 4.52. The source should be compiled as C rather than C++.

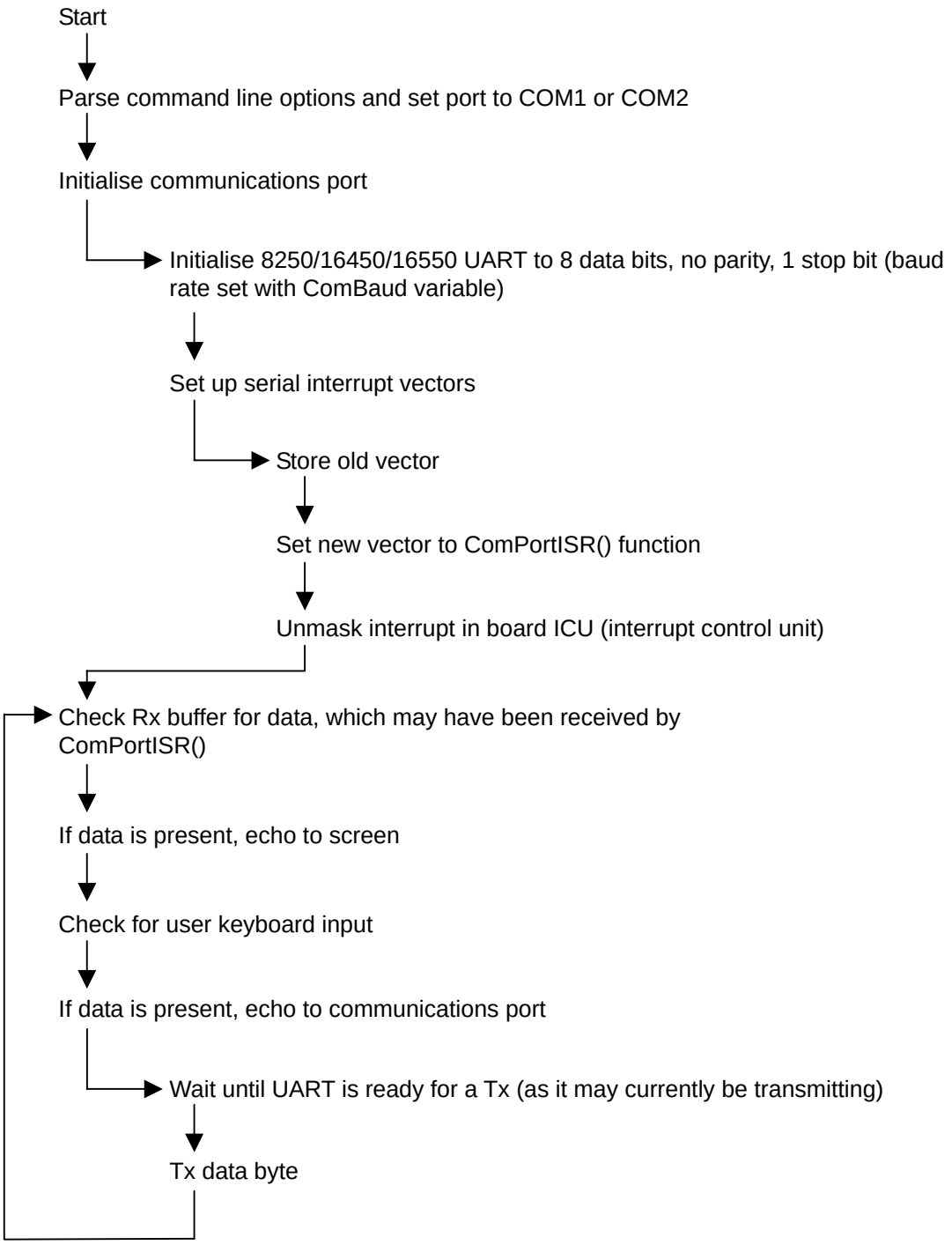
To recompile use:

```
BCC INT_COMS.C
```

To step through with Turbo Debugger use:

```
BCC -v INT_COMS.C
TD INT_COMS
```

Flow diagram



POL_COMS.C

This Borland C source file illustrates the setup of an 8250 compatible UART for polled mode communications over an RS232 link (i.e. 8250, 16450, 16550).

Standard COM1, COM2 I/O locations are used by default, but these can be changed easily in the source code (ref. ComPort variable).

The UART is set up for polled Rx and Tx. This is useful when using a system with a limited number of free hardware interrupts (IRQs).

To ensure that RxD data is received without dropping characters, the DOS time of day interrupt (INT08, Timer0) is reconfigured to chain an additional interrupt service routine. This function is called approximately every 5.5ms, which is sufficient for communications at up to 19200, 8, n, 1.

To support a faster baud rate, the timer should be set up to call the ISR more frequently, with the rate given by

```
Timer Rate = [ (SDC Packet Size) * Rx FIFO Depth ] / [ Baud Rate ]
```

where

`Timer Rate`	is the frequency at which the timer should generate interrupts calling the handler.
`SDC Packet Size`	is the total number of bits per data packet, e.g. for 8, n, 1 this implies: 1 start bit + 8 data units + 1 stop bit = 10
`Rx FIFO Depth`	is the level of FIFO buffering supported by the UART. The 16550 UART supports a 16 byte Rx FIFO.
`Baud Rate`	is the fastest baud rate to be supported.

Example 1

For 19200, 8, n, 1, 16 byte FIFO communications, the timer rate = 10*16/19200
= 8.3ms

Therefore the standard Timer0 divisor (DIVISOR) of 10 is sufficient for this baud rate.

Example 2

For 115200, 8, n, 1, 16 byte FIFO communications, the timer rate = 10*16/115200
= 1.38ms

Therefore the standard Timer0 divisor (DIVISOR) needs to be increased to 40+ to support communications at this rate, and the ISR will generate a high CPU load.

Example 3

For 19200, 8, n, 1, no FIFO communications, the timer rate = 10/19200
= 0.5ms

You can see that polled implementation on a UART without FIFO (e.g. 8250 or 16450) is not practical as the ISR would need to be called every 500µs.

DOS timer tick accuracy

The DOS timer tick is chained from the polled mode ISR in this implementation. This means that, depending on the divisor value used (DIVISOR), there is a degree of inaccuracy in the DOS tick whilst operating polled communications.

The DOS timer tick is reset from the real time clock (RTC) on a power cycle or, alternatively, the application may reset the DOS tick from the RTC directly.

You need to be particularly aware of this if a timing critical application is implementing polled communications.

Compilation

The C source file compiles under Borland C++ 3.1 or Borland C++ 4.52. The source should be compiled as C rather than C++.

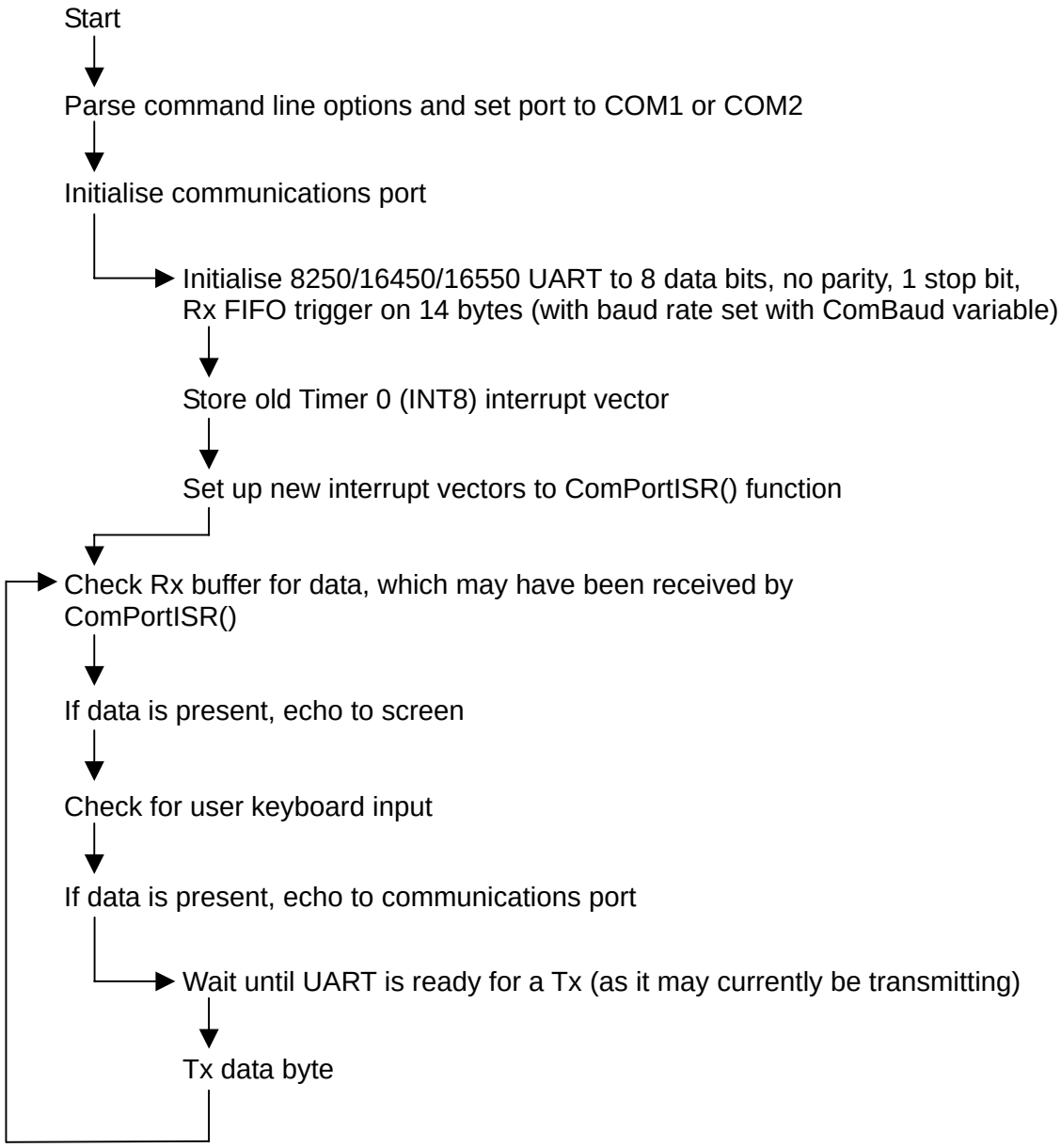
To recompile use:

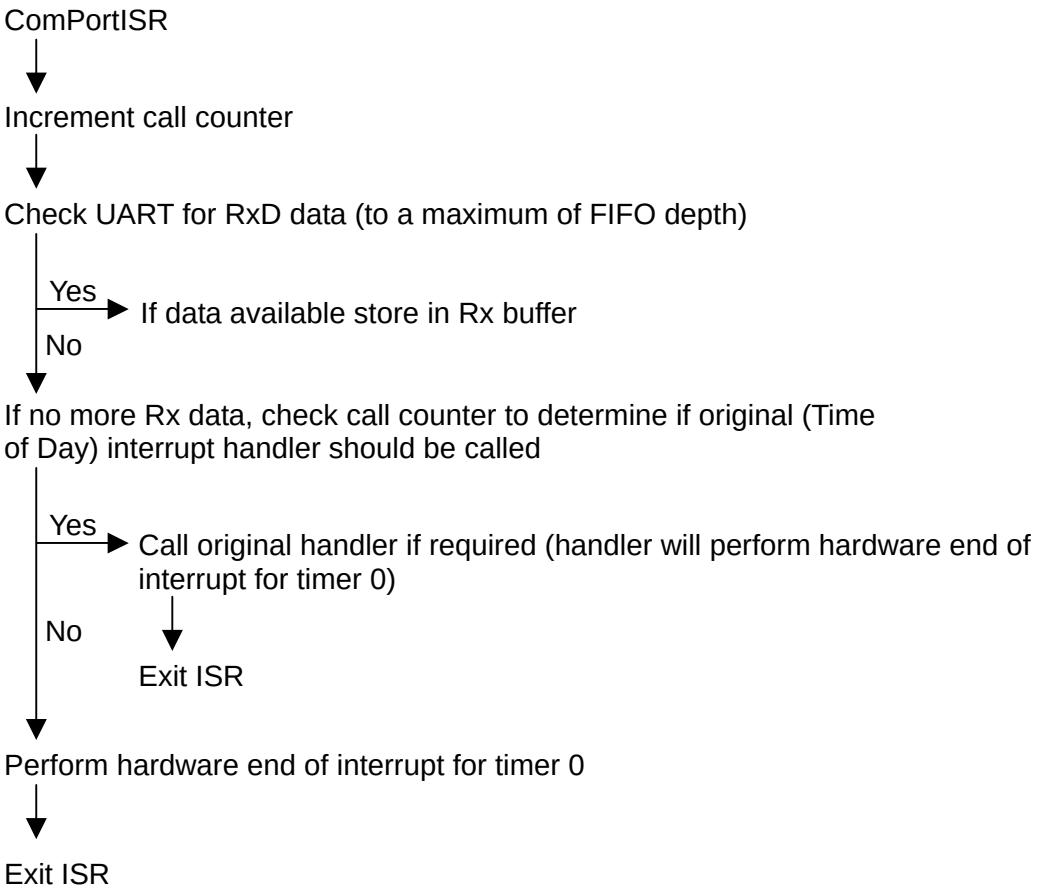
```
BCC POL_COMS.C
```

To step through with Turbo Debugger use:

```
BCC -v POL_COMS.C
TD POL_COMS
```

Flow diagrams




ComPortISR
Increment call counter
Check UART for RxD data (to a maximum of FIFO depth)
Yes
If data available store in Rx buffer
No
If no more Rx data, check call counter to determine if original (Time of Day) interrupt handler should be called
Yes
Call original handler if required (handler will perform hardware end of interrupt for timer 0)
No
Exit ISR
Perform hardware end of interrupt for timer 0
Exit ISR

Appendix C – Notes on using the 16C550 UART

This appendix contains information about using the 16C550 UART.

Enabling the interrupt outputs

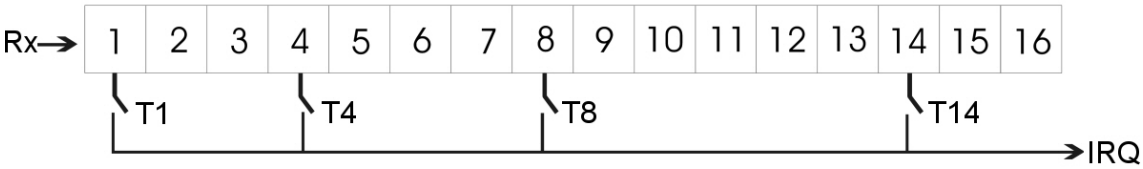
If you intend to write your own low level UART handler, you must enable the IRQ output from the UART by setting bit 3 in the modem control register (MCR), so that the UART channel on the AIM104-COM4 can generate interrupts to the host CPU.

This can be done with standard driver software (including that supplied on the library disk with the AIM104-COM4) as part of the port initialisation. No action is necessary if you are using Eurotech Ltd's software or that of a third party vendor.

The receiver FIFO trigger point and timeout

The transmit buffer and receive buffer on the 16C550 have 16 byte FIFOs. This enhances the performance of the serial ports and reduces the interrupt rate to the main processor. However, you need to set up the receive trigger point optimally for your application.

With the FIFO enabled, any received character is added to the 16 byte receive FIFO. This has four programmable trigger points as shown in the following diagram:



Once the FIFO has received sufficient characters to reach the FIFO trigger point, an interrupt is generated that is serviced in order to remove all the characters from the FIFO.

If the FIFO trigger point is set to a value other than 1, a situation may arise where the input data stream stops adding characters to the FIFO before the trigger point is reached. For example, the trigger may be set at 8 but only 6 characters are received. In this situation the UART generates an IRQ after a specific timeout period from the end of the last character received.

The timeout period, expressed in the number of characters can be calculated using the following equation:

$$T = \frac{(4 \times n_{Char}) + 12}{(n_{Parity} + n_{Char} + n_{Stop} + 1)}$$

Where: n_{Char} is the number of programmed bits per character (5, 6, 7 or 8)

n_{Parity} is the number of programmed parity bits (1 unless no parity)

n_{Stop} is the number of stop bits (1 ,1½ or 2)

The actual time for the timeout is simply defined as:

$$t = \frac{(4 \times n_{Char}) + 12}{F_{Braud}}$$

where: F_{Baud} is the baud rate.

For example, communications at 9600 baud, 8 data bits, no parity, and 1 stop bit have the following timeout period:

$$T = \frac{(4 \times 8) + 12}{0 + 8 + 1 + 1} = \frac{44}{10} = 4.4 \text{ chars}$$

$$t = \frac{(4 \times 8) + 12}{9600} = \frac{44}{9600} = 4.58 \text{ ms}$$

This timeout can significantly impact on the performance of the serial communications within an application. If your serial communications is single character oriented, it is better to set the FIFO trigger point lower and handle more frequent receive interrupts. If your application transfers large blocks of data it is likely to be better to set the FIFO trigger point higher.

Remember that the FIFO trigger point does not define the size of the FIFO; characters are still received after the trigger has been activated until all sixteen positions in the FIFO have been filled.

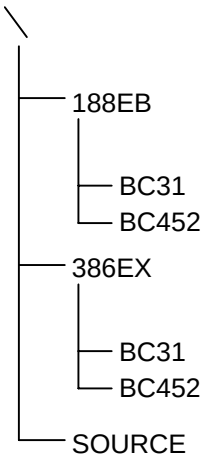
Appendix D – Software library

Supplied with the AIM104-COM4 is a library disk containing extensions to the Eurotech Ltd Target Libraries for the Target188EB and Target386EX CPU boards produced by Eurotech Ltd. If you are using a PC AT type host, a large number of third party vendors produce serial communications libraries for the 16C550 UART device, and standard operating systems such as MS-DOS, ROM-DOS, and Microsoft Windows inherently support this device.

If you intend to write your own low level software, you can start with the source code on the disk, which is supplied as example code only for use on Eurotech Ltd products. No warranty is made on this code.

Installing the software

The library disk contains the following directory structure:



To install the library support for your development platform, copy all the files from the appropriate platform or compiler directory to your development directory.

Using the AIM104-COM4 with the target CPU board

To use the library extensions with your target board you must do the following:

1. Include the file 16C550.H immediately after including the file ArComm.H
2. Ensure that the library file 165500x.LBR is included at link time. x denotes the build model and can be s, m, c, l, or h for the small, medium, compact, large, or huge model respectively.

Once these criteria are met, the ports on the AIM104-COM4 are initialised using the following function:

```
SINT16 wInit16550 (UINT16 uIoAdr, UINT16 uIrq, UINT16 uFIFO)
```

This initialises the 16550 port (returning either SUCCESS or FAILED) and assigns it the next port number. The parameters of this function are:

uIoAdr Port base I/O address.

uIrq IRQ number that the port uses on the CPU.

uFIFO a number that denotes the trigger point on the FIFO. The FIFO is configured according to the table below.

uFIFO	FIFO mode	Receive FIFO trigger point	Transmit FIFO operations
0	Disabled	Not applicable - functions as a 16450 UART	
1-3	Enabled	1 character in receive FIFO	Copy up to 16 characters into FIFO
4-7	Enabled	4 characters in receive FIFO	Copy up to 16 characters into FIFO
8-13	Enabled	8 characters in receive FIFO	Copy up to 16 characters into FIFO
14-...	Enabled	14 characters in receive FIFO	Copy up to 16 characters into FIFO

Once initialised, you can access the UART on the AIM104-COM4 using the target library functions such as ArOpen(), ArPutc(), ArRts(). Full documentation for the use of the target libraries is supplied with the target libraries software disk.

Index

A

address link · 9, 13
address map · 8, 9
anti-static · 5
AT communications channels · 8

B

baud rate · 4
board · 5, 7
bus · 4

C

CAB AIM104-COM4 · 7, 27
cable · 7, 27
channel A mode links · 14
channel B mode links · 15
channel enable link · 14
channels · 4, 8, 11
  A & B · 18
  C & D · 25
code · 29, 37
communication mode · 20
connector pin assignments · 26
contact details · 28
copyright · 2

D

differential communications channels · 18
  line terminations · 20
  modes · 19
differential pair termination links · 16

E

earth wire · 5
electromagnetic compatibility · *See* EMC
EMC · 5
example code · 29
external interface board · 5

G

ground · 4

H

handling · 5
high voltage transients · 5

I

I/O · 4
  map · 8, 9
interrupt links · 11, 13
interrupts · 8
IRQs · 4, 11
isolation · 4

L

LED · 4
line terminations · 20
links · 12
  default configuration · 12
  LK1 · 9, 13
  LK10 & LK12 · 15
  LK13-LK15 · 16
  LK2-LK5 · 11, 13
  LK6 · 13
  LK7 · 14
  LK8 · 14
  LK9 & LK11 · 14

M

MTBF · 4

N

noise emissions · 5

O

operating temperatures · 4
option link · 13
opto-isolation · 4, 5

P

packaging · 5
PC/104 · 4, 8
PL3 · 26, 27
ports · 4
power requirements · 4

R

registers · 9
ribbon cable · 7, 27
RS232 · 4, 25
RS422 · 4, 19
RS485 · 4, 19
  flow diagrams · 23

S

safety · 5
software library · 37
source code · 28, 37
static · 5
storage · 5
support, technical · 28
symbols · 6

T

technical support · 28
temperature range, operating · 4
terminations · 20
terminology · 6
trademarks · 2

U

UART · 4, 8, 35

V

voltage · 4
  high voltage transients · 5

Z

zero wait state · 4, 8
  link · 14