



OXCF950

Single UART with PCMCIA or Compact Flash interface

EVALUATION BOARD

This document describes the OXCF950 evaluation board, how to set its configuration, install drivers and troubleshoot common problems.

FEATURES

- Single full-duplex asynchronous channel
- 128-byte deep transmitter / receiver FIFO
- Fully software compatible with industry standard 16C550 type UARTs
- Readable FIFO levels
- System clock up to 60MHz
- Flexible clock prescaler from 1 to 31.875
- 9-bit data framing as well as 5,6,7 and 8
- Detection of bad data in the receiver FIFO
- Automated in-band flow control using programmable Xon/Xoff characters
- Transmitter and receiver can be disabled
- Low power CMOS
- 3.3V operation, 5V tolerant I/O
- Extended temperature range -40C to +105C
- Generic embedded driver compatibility mode
- Programmable by external Microwire™ EEPROM (EEPROM programmed via Oxford Semiconductor utilities).
- Extremely low power consumption by use of asynchronous UART core and power down (sleep) modes
- Range of packages -Ultra small 48 pin TQFP package or a 48 ball TFBGA.
- Supports all UART types 450 up to 950 (fully programmable)
- CF+ Compliant (Revision 1.4).
- 16-bit PC Card Compliant (PCMCIA Revision 7.1)
- 8 bit Local Bus interface included for PCMCIA applications
- 2 Multi-purpose I/O pins which can be configured as interrupt inputs
- Software compatible with OXCF950
- 16C950 mode for local bus applications

DESCRIPTION

The OXCF950 rev B is a low cost asynchronous 16-bit PC card (henceforth referred to as PCMCIA) or Compact Flash (henceforth referred to as CF) UART (and Local Bus) device. Local Bus Selection is performed by use of a MODE pin. Note that Local Bus mode uses indirect addressing, which is only supported by PCMCIA.

The 3.3V technology has been specified to operate as low as 2.7 V to allow an in-line regulator to be used for mixed 3V/5V applications. All the inputs and outputs are 5V tolerant with the exception of the clock/crystal oscillator input.

The EEPROM interface allows the programming of the Attribute Memory, UART and Local Configuration Registers during power up or hard/soft reset. This allows different card manufacturers to modify the information contained in the Attribute memory or UART/registers as required, for example PC-Card ID value.

A number of power-down modes are included to keep power consumption to a minimum. Such features include clock division (fully programmable) and sleep modes when a function of the OXCF950 rev B is not being used.

The OXCF950 rev B contains a single-channel ultra-high performance UART offering data rates up to 15Mbps and 128-deep transmitter and receiver FIFOs. Deep FIFOs reduce CPU overhead and allow utilisation of higher data rates.

It is software compatible with the widely used industry-standard 16C550 type devices and compatibles, as well as other OX16C95x family devices.

In addition to increased performance and FIFO size, the OXCF950 rev B also provides enhanced features including improved flow control. Automated software flow control using Xon/Xoff and automated hardware flow control using CTS#/RTS#

and DSR#/DTR# prevent FIFO over-run. Flow control and interrupt thresholds are fully programmable and readable, enabling programmers to fine-tune the performance of their system. FIFO levels are readable to facilitate fast driver applications.

The addition of software reset enables recovery from unforeseen error conditions allowing drivers to restart gracefully. The OXCF950 rev B supports 9-bit data frames used in multi-drop industrial protocols. It also offers multiple ISDN, xDSL.

The OXCF950 rev B also incorporates a bridge to an 8 bit Local Bus in Local Bus Mode. It allows card developers to expand the capabilities of their products by adding

peripherals to this bus. In addition, two user IO pins are included to enhance external device control. These IO pins can also be configured as interrupt inputs. The OXCF950 rev B fixes all known errata of the OXCF950 part. It also adds two new modes of operation. The first mode (pin or EEPROM selectable) allows the device to function in embedded systems where the generic device driver present assumes I/O occupancy of 8 byte addresses, without any additional components. The second mode (pin selectable) allows the device to act as a stand-alone 16C950 type part with the benefits over the 16C950 rev B of 5V tolerance, and packaging options where pin-compatibility with 550 type UARTs is not required.

SUMMARY OF DIFFERENCES BETWEEN OXCF950 REV A AND OXCF950 REV B

The OXCF950 rev B has been designed to be as compatible as possible with the rev A part. A designer considering switching to the new device should be aware of the following: –

- Power supply. The new part has a supply voltage range from 2.7V to 3.6V. The original part had a supply voltage range from 3.15V to 5.25V. A new design requiring operation at both 3.3V and 5V will require an in-line low voltage-drop regulator.
- Pin compatible (other than supply voltage constraints).
- Software compatible – all the modes of the original CF950 are available in the rev B device.
- All I/Os are now 5V tolerant, with the exception of the Crystal oscillator input.
- The crystal oscillator circuit is only suitable for frequencies up to 20 MHz. For operation above this frequency, an external clock source is required.
- A new 'generic' mode has been added allowing generic drivers to access the UART without needing external circuitry to adjust the address ranges. This mode (8 byte I/O space) can be selected either by pin control or through the EEPROM. Access to configuration registers is still possible via a software paging technique.
- A stand-alone 16C950 type mode has been added (pin selectable) to allow the device to be used where pin-compatibility with 16C550 type devices is not required.
- All known errata in the original OXCF950 have been fixed.
- The device is now characterised for an extended operating temperature range of –40C to +105C.
- The CF950 rev B is additionally available in a 48 TFBGA package.
- The input clock frequency may be restricted by the CF950 rev B environment. Where an external EEPROM is being used, the maximum clock frequency of the external EEPROM multiplied by 64 will give an input clock frequency limit. For an external EEPROM powered off 5V with a maximum frequency of 1MHz, this sets no limits. An EEPROM powered off 2.7V may typically have a maximum frequency of 250 kHz, which would restrict the CF950 rev B input frequency to 16 MHz.

Software can identify the difference between an OXCF950 and an OXCF950 rev B by reading the UART REV

For the original OXCF950 this reads 0x06, for the OXCF950 rev B this reads 0x08.

TABLE OF CONTENTS

FEATURES	2
DESCRIPTION	2
SUMMARY OF DIFFERENCES BETWEEN OXCF950 REV A AND OXCF950 REV B	3
TABLE OF CONTENTS	4
1 EVALUATION BOARD DESCRIPTION	5
2 BOARD LAYOUT	6
3 DRIVER INSTALLATIONS	7
4 TROUBLESHOOTING COMMON PROBLEMS	8
5 REFERENCE DRIVERS	10
6 NOTES	11
APPENDIX A: OXPROM - EEPROM PROGRAMMING UTILITY	12
APPENDIX B: APPLICATION NOTES AVAILABLE	15
APPENDIX C: RS232 CONNECTOR PIN ASSIGNMENT	16
CONTACT INFORMATION	21
DISCLAIMER	21

1 EVALUATION BOARD DESCRIPTION

This document SHOULD be read before attempting to use the evaluation board in order to prevent simple setup errors occurring. The evaluation board for OXCF950 users provides an environment in which the various modes and features of the OXCF950 device can be demonstrated. A reference schematic for the OXCF950 evaluation board is available on request from Oxford Semiconductor (cf950b_bga_13.pdf OR B005H03 rev1.2). The evaluation board provides a simple means of realising the following products:-

- 1-port serial (1xRS232)

The specific features available include:-

- Configuration into any of the modes available

The drivers that have been provided for this board have been written for Windows 9*, Windows 2000, Windows NT and Windows XP. There are drivers available for Linux but these HAVE NOT been written by Oxford Semiconductor and we can therefore no be held responsible nor do we provide updates for these.

2 BOARD LAYOUT

The layout of the evaluation board is shown in figure 1.

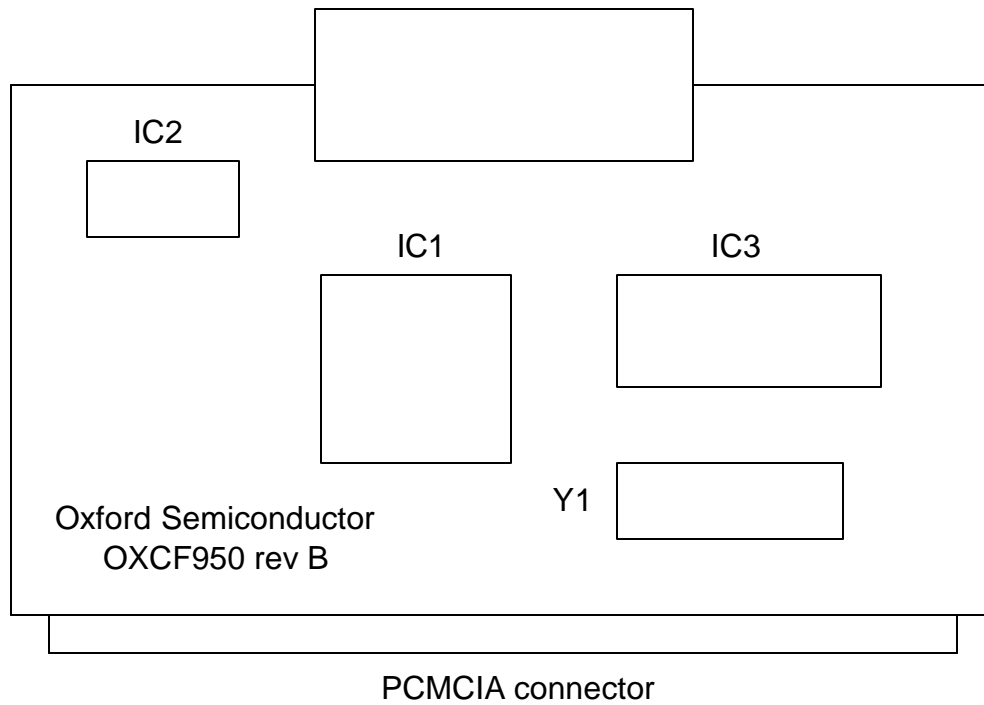


Figure 1: Evaluation board layout

Key

- IC1: OXCF950
- IC2: Serial EEPROM
- IC3: Serial Line driver
- Y1: Crystal Oscillator

3 DRIVER INSTALLATIONS

For detailed instructions on how to install the reference drivers, please refer to the document "Reference Drivers" document. The document 'CustomDrivers.pdf' details how to change the VENDOR ID, DEVICE ID, SUBSYSTEM VENDOR ID and SUBSYSTEM ID fields. In this way, it is possible for the Plug 'n' Play system to identify different boards (even though they use the same chip) and load the correct drivers. If particular problems are being encountered with baud rates and the use of drivers options request Application note 15 'Setting UART baud rates under Windows' from Oxford Semiconductor.

4 TROUBLESHOOTING COMMON PROBLEMS

Problem: The system detects the ports but a terminal application will not recognise them.

Solution: Ensure that the application supports COM ports other than the four legacy addresses (3f8, 2f8, 3e8, 2e8). If not, use an application that supports all COM devices, such as Quarterdeck Procomm Plus®, or Ericom PowerTerm®.

Problem: Some or all of the ports will not communicate with other devices

Solution: Ensure that the baud rates are correct. The OX16C95x family of UARTs has a very flexible baud rate generator, which can accept differing crystal frequencies, prescaler values etc. The serial port configuration utility in the Windows 9x driver can set baud rate multipliers etc. or completely override an application's baud rate setting.

Problem: Doing a loop back test with a terminal program and getting no response.

Solution: Loop back connector has been wired up in correctly.

Problem: The device manager says

"PCMCIA UNKNOWN_MANUFACTURER"

Solution: If there have been some corruptions in the EEPROM for whatever reason the drivers may not be installed or the card be accepted by the OS. In order to get the OS to accept the card the following steps are necessary.

1. Unplug the CF950 card from the system.
2. Locate the EEPROM on the CF950 card unsolder the DATA OUT pin so that this is no longer connected to the CF950. (Alternatively on a B005H03 board remove R7 temporarily on a cf950b_bga_13 board remove R4)
3. Plug the board back in
4. The system will now pick the default settings in the driver and not be overwritten by the contents of the EEPROM. Once the system has recognised the card and installed all the appropriate driver files the board can be unplugged again.
5. Reconnect the DATA OUT pin so that this is connected to the CF950. (As mentioned above replace the resistors)
6. Now using OXPROM erase the contents of the EEPROM
7. The EEPROM can then be reprogrammed using one of the .dat files provided for the CF950.

Problem: The driver will not load

Solution: This can come about due to changes in some of the CIS information in the EEPROM making it different from information found in the OXSER.inf files. For example the line below shows the Manufacturer String "CF CARD" and the Product String "GENERIC" contained in the CISTPL_VERS_1 tuple and found in the OXSER.inf file.

```
%Card0.DeviceDesc% = ComPortCF, PCMCIA\CF_CARD-GENERIC-A569
```


During installation if this information is different then the windows registry will need to be looked at to get the checksum (e.g. A569). A note of this the manufacturer and product string needs to be made. This then needs to be transferred to the OXSER.inf file by editing it. Once this has been done an attempt to load the driver can be made. Once the driver has been loaded OXPROM can be used to reload a working data file. Once this has been done the process needs to be repeated so that the information in the OXSER.inf file is the same as in the .dat file.

Problem: CF950 in PCMCIA mode in 2k

This is an MS QuickFix which enables the CF950 to run in PCMCIA mode (as opposed to CF Mode) in Windows 2k.

Solution:

- Install the Q305593 patch (NOTE: Service pack 1 or later must be installed on the system.)
- Install service pack 3 (SP3 which incorporates the above fix)

Problem: The OXCF950 has a 16-byte address range to support the local bus functionality. Legacy drivers expect an 8-byte wide address range, and therefore modifications have to be made to use these legacy drivers.

Observed with Compaq IPAQ and a Cassiopeia there is compatibility issue with the generic WinCE (and other) drivers.

Note: This issue will only arise with OXCF950 rev A chips and not OXCF950 rev B chips.

Solution: Application note 13 available from Oxford Semiconductor details the hardware modifications necessary to use legacy drivers.

Problem: There is a bug in Windows in the power management section of code. There are certain situations where applications try to access the OXCF950 when Windows has powered it down. Windows then powers up the device in the wrong sequence, the device does not respond quickly enough and the O/S assumes that it is a 550 device. All 950 features are then ignored, and thus the baud rate, flow control etc are incorrectly set up. Some applications will crash at this point.

Solution:

Windows 98

Under Windows 98 the following registry key need to be changed globally.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VCOMM
Change EnablePowerManagement DWORD value to 0x00000000:

Windows 2000

Under Windows 2000 the following registry key needs to be changed for each card

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ENUM****\1\Device
Parameters\
Add DWORD 'DisablePowerManagement' with data 1 (0x00000001)

Where ***** = Car d name e.g. CF_CARD-GENERIC-A569

5 REFERENCE DRIVERS

The all the drivers are available from the Oxsemi web site at <http://www.oxsemi.com> under **Product information** [UART / parallel port](#) and [PCI](#). These areas will contain a reference drivers for Windows 9x , Windows 2000, and Windows NT4. It also contains related utilities and documentation.

The contents are listed below:-

\	oxpar.inf - driver .inf file oxpar2.inf - driver .inf file (2) oxpci.inf - driver .inf file oxpci2.inf - driver .inf file (2) oxser.inf - driver .inf file oxmep.sys - PCI support driver oxmf.vxd/sys - bus enumerator oxmfuf.sys - Upper filter driver oxpar.sys - port driver oxpp.vxd - parallel port enumerator oxser.vxd/sys - port driver oxppui.dll - parallel port configuration utility oxserui.dll - driver configuration utility oxui.dll - driver configuration utility
\ oxprom	oxprom.exe - Serial EEPROM programming utility oxeprom.sys - needed by oxprom.exe .dat files - example EEPROM configuration files
\utils	readuart.exe - dump UART registers from within Win9x. poke.exe - memory access utility .vxd files - needed by above utilities
\doc	OXCF950.pdf - this file advancedtopics.pdf - advanced baud rate configuration in Windows 2000 customdrivers.pdf - details on customising drivers referencedrivers.pdf - driver installation and use
\NT4_Parallel	Install_Parallel.exe - install program for Windows NT4 parallel port driver oxpar.inf - installation script oxpar.sys - port driver licence.txt - licence agreement
\NT4_Serial	Install_Serial.exe - install program for Windows NT4 serial port driver oxser.inf - installation script oxser.sys - port driver licence.txt - licence agreement

6 NOTES

This page has been intentionally left blank.

APPENDIX A: OXPROM - EEPROM PROGRAMMING UTILITY

Overview

This program enables card designers to configure Oxford Semiconductor PCI devices by programming the attached EEPROM. It provides a full GUI interface which is used to create a configuration file according to the exact requirements of the vendor. This file can then be used to program the card while it is still in the slot. Then (after a reboot) the device will take on the attributes set in the newly-programmed EEPROM. If an EEPROM has been fitted it WILL not have been programmed with any data and will therefore read back 0xFF when OPROM is used.

Instructions for use

The operation of OXPROM with the OXCF950 differs from the operation with other devices. The following details how to use OXPROM with this device. Select the device being used in your new design, and the mode in which it will be used (dependent on the device used).

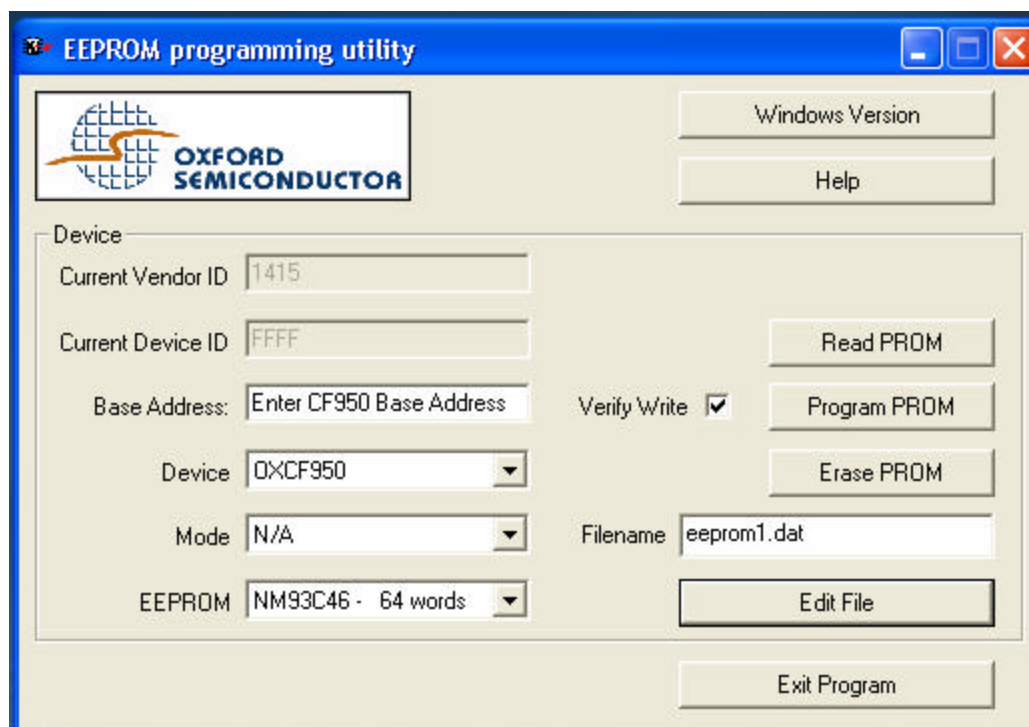


Figure 2

NOTE: OXPROM does not yet support editing CF950 .dat files

When the OXCF950 is selected certain options are removed, and there is a box to enter the CF950 base address, as shown in Figure 2.

The base address can be found in the device manager, as shown in figures 3 and 4.

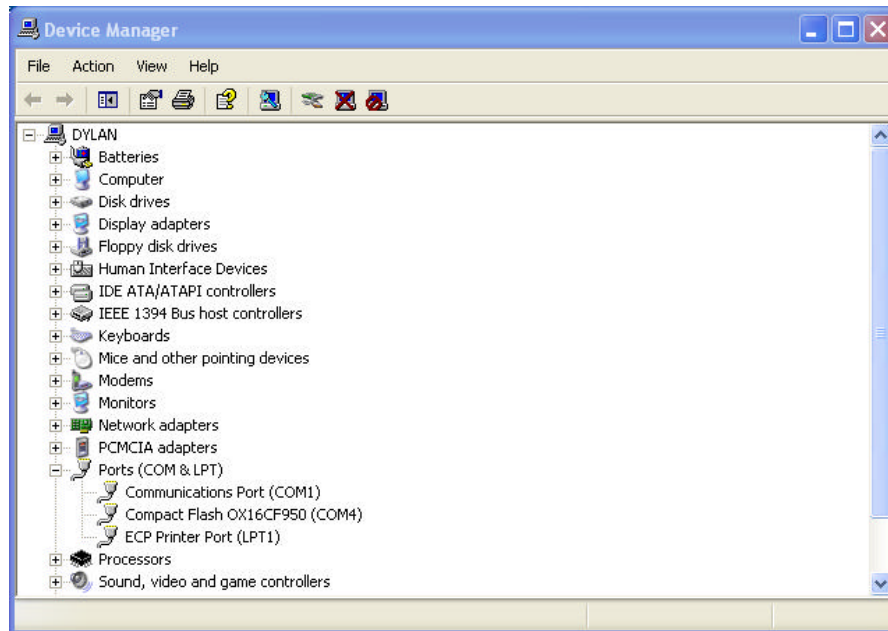


Figure 3 – Device Manager

Right click on 'Compact Flash OX16CF950 (COM4)' The COM port allocated may not be COM4. Select 'Properties'.

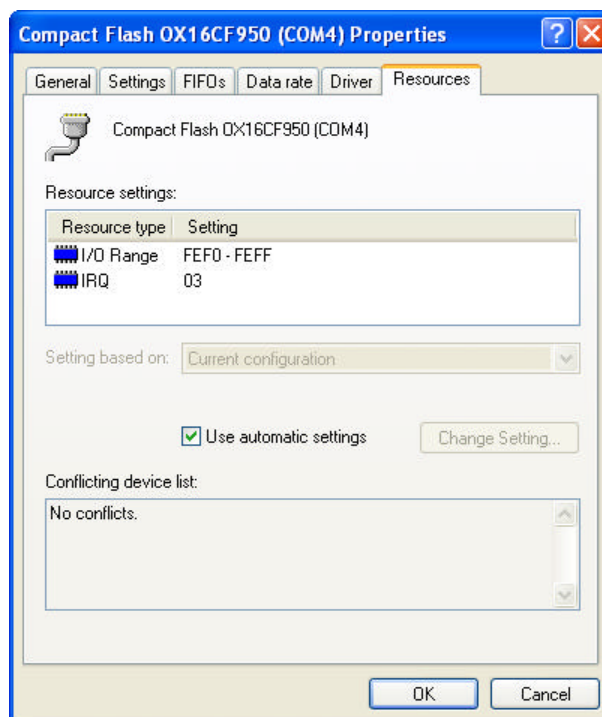


Figure 4 – Device Properties

The first address has to be entered into the base address box, as shown in figure 5.

**Figure 5 – Entering the Base Address**

Check that the EEPROM selected matches the EEPROM on the board. Then program the EEPROM using a pre-prepared CIS file, which needs to reside in the OXPROM directory, as detailed in the data sheet for the CF950 which can be obtained from Oxford Semiconductor.

Once your file is ready, select the type of EEPROM that is being used, and click "Program PROM".

Note: if the device has already been rebooted with a custom vendorID / deviceID, it will be necessary to enter the new settings in the appropriate boxes; otherwise the application will not be able to find the device.

After a short delay, a message box should appear to inform you that the chip has been successfully programmed. If this does not occur (or it reports an error), check the hardware/EEPROM/connections etc. as appropriate and try again

It is possible to read back the contents of a programmed EEPROM. Just click "Read PROM", and then click "Go". The contents will be displayed on screen

APPENDIX B: APPLICATION NOTES AVAILABLE

These are available from Oxford Semiconductor on request.

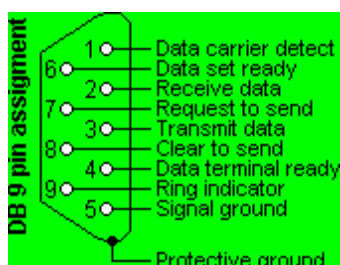
App note number	Title	Product
AN2	Using the OX16C950B in an ISA environment	OX16C950B
AN3	Using the OX16C95x in isochronous clock mode	OX16C950B
AN3	Using the OX16C95x in isochronous clock mode	OX16C954
AN4	Interfaces and line drivers for standard serial and parallel applications	OX16C950B
AN4	Interfaces and line drivers for standard serial and parallel applications	OX16C954
AN4	Interfaces and line drivers for standard serial and parallel applications	OX16PCI954
AN4	Interfaces and line drivers for standard serial and parallel applications	OX16PCI952
AN4	Interfaces and line drivers for standard serial and parallel applications	OX16CB950
AN4	Interfaces and line drivers for standard serial and parallel applications	OX12PCI840
AN4	Interfaces and line drivers for standard serial and parallel applications	OX16CF950
AN5	Software examples for the OX16C95x family of products	OX16C950B
AN5	Software examples for the OX16C95x family of products	OX16C954
AN5	Software examples for the OX16C95x family of products	OX16PCI954
AN7	Common problems with the OX16PCI954 and NT4	OX16PCI954
AN8	Using the OX16PCI954 in a low-cost four-port serial card	OX16PCI954
AN9	Using the OX16PCI954 in a combo four-port serial, one-port parallel card	OX16PCI954
AN 10	8-port serial card design	OX16PCI954
AN 10	8-port serial card design	OX16C954
AN 11	12,16,20-port card design	OX16PCI954
AN 11	12,16,20-port card design	OX16C954
AN12	Common Connectivity Examples for the OX16PCI954	OX16PCI954
AN13	Using the OXCF950 with legacy drivers	OXCF950
AN14	Using the OXCF950 under Windows	OXCF950
AN15	Setting UART baud rates under Windows	OX16PCI954
AN15	Setting UART baud rates under Windows	OX16PCI952
AN15	Setting UART baud rates under Windows	OX16CB950
AN15	Setting UART baud rates under Windows	OXCF950
AN16	Using OXPROM with the OXCF950	OXCF950
AN17	How to write a CIS for the OXCF950	OXCF950
AN18	Physical details of PC and CF cards	OXCF950

APPENDIX C: RS232 CONNECTOR PIN ASSIGNMENT

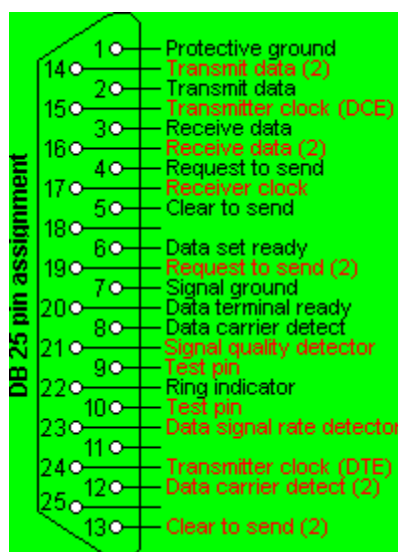
The diagrams below show the signals common to both connector types in black. The signals only present on the larger connector are shown in red. Note, that the protective ground is assigned to a pin at the large connector where the connector outside is used for that purpose with the DB9 version.

Although this interface is differential (the receive and transmit have their own floating ground level) it is possible to connect RS232 compatible devices with this interface.

RS232 DB 9 pin assignment



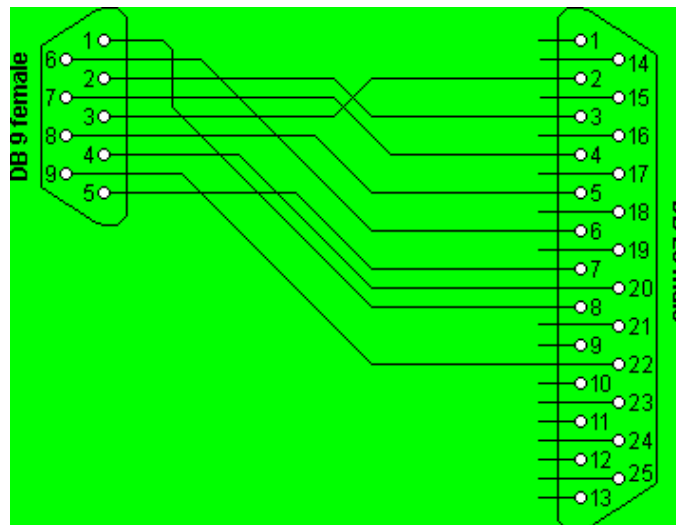
RS232 DB 25 pin assignment



DB9 to DB25 converter

The original pin layout for RS232 was developed for a 25 pins sub D connector. Since the IBM-AT, 9 pins connectors are commonly used. In mixed applications, a 9 to 25 pins converter can be used to connect connectors of different sizes.

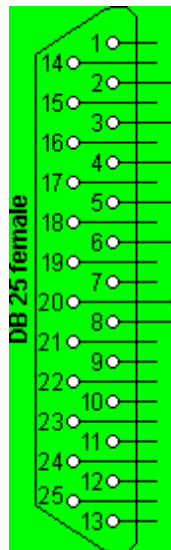
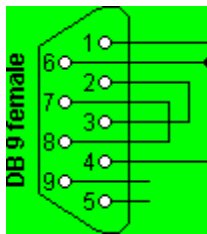
RS232 DB9 to DB25 converter



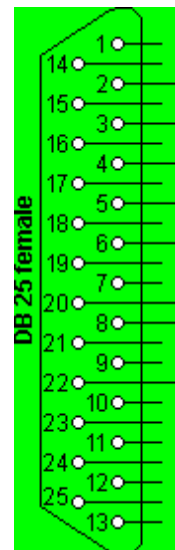
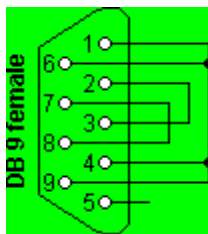
RS232 loop back test plug

The following connectors can be used to test a serial port on your computer. The data and handshake lines have been linked. In this way all data will be sent back immediately. The PC controls its own handshaking. The first one can be used to check the function of the serial port with standard terminal software. The second version can be used to test the full functionality of the serial port with Norton Diagnostics or CheckIt.

RS232 loop back test plug for terminal emulation software



RS232 loopback test plug for Norton Diagnostics and CheckIt

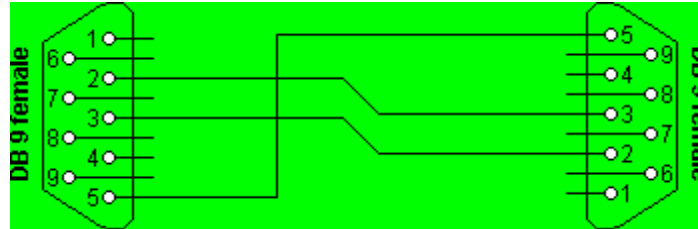


RS232 null modem cables

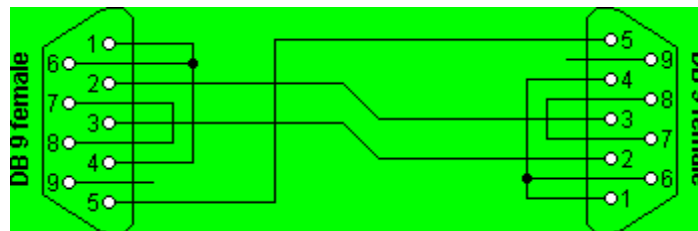
The easiest way to connect two PC's is using a null modem cable. The only problem is the large variety of null-modem cables available. For simple connections, a three line cable connecting the signal ground and receive and transmit lines is sufficient. Depending of the software used, some sort of handshaking may however be necessary. For a Windows 95/98 Direct Cable Connection, the null modem cable with loop back handshaking is a good choice.

Null modem cables with handshaking can be defined in numerous ways, with loop back handshaking to each PC, or complete handshaking between the two systems. The most common cable types are shown here.

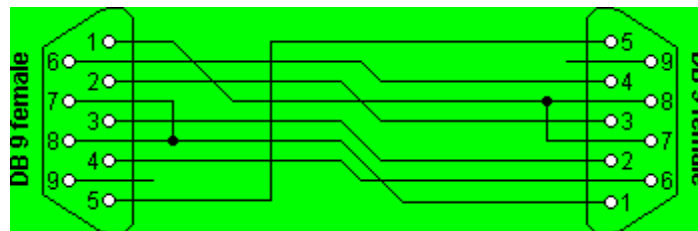
Simple null modem without handshaking



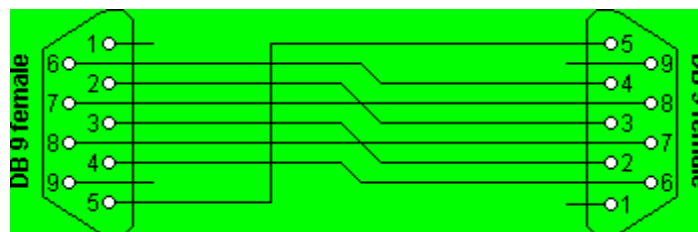
Null modem with loop back handshaking



Null modem with partial handshaking



Null modem with full handshaking

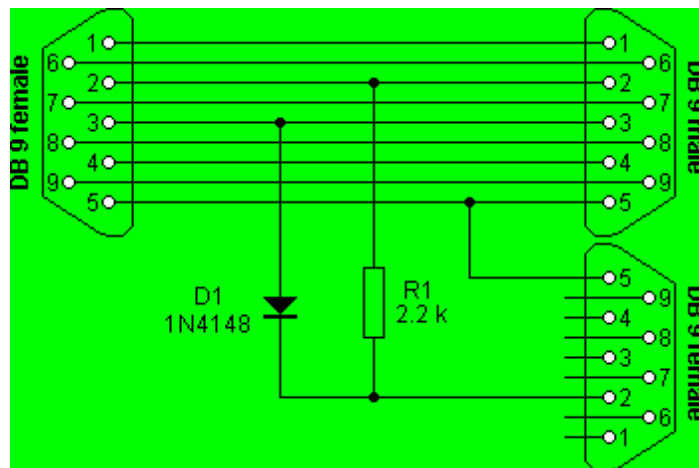


RS232 monitor cable

It is not difficult to monitor the serial communication between two devices with a PC. To do this you need the monitor cable which is displayed in the next picture. Two sockets are connected straight through. The PC is connected to the third one.

This monitor cable taps communication from both sides. This means that if the two devices happen to talk simultaneously, the monitored information will be garbage. In most circumstances communication software works half duplex, in which case this problem does not exist.

RS232 monitor cable



CONTACT INFORMATION

For further information please contact:

Oxford Semiconductor Ltd.

25 Milton Park

Abingdon

Oxfordshire

OX14 4SH

United Kingdom

Telephone: +44 (0)1235 824900

Fax: +44 (0)1235 821141

Sales e-mail: sales@oxsemi.com

Technical support: Support@oxsemi.com

Web site: <http://www.oxsemi.com>

DISCLAIMER

Oxford Semiconductor believes the information contained in this document to be accurate and reliable. However, it is subject to change without notice. No responsibility is assumed by Oxford Semiconductor for its use, nor for infringement of patents or other rights of third parties. No part of this publication may be reproduced, or transmitted in any form or by any means without the prior consent of Oxford Semiconductor Ltd. Oxford Semiconductor's terms and conditions of sale apply at all times.