



SPECTRUM
SYSTEMENTWICKLUNG MICROELECTRONIC GMBH

PCI.248 / PCI.258
200 (400) MS/s - 250 (500) MS/s
8 bit transient recorder
for PCI bus

Hardware Manual
Driver Manual

© Spectrum Systementwicklung Microelectronic GmbH - 2003
Bültbek 26, 22962 Siek, Germany

SBench is a registered trademark of Spectrum Systementwicklung Microelectronic GmbH.

MS-DOS, Windows and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

LabVIEW is a trademark of National Instruments Corporation.

MATLAB is a registered trademark of The MathWorks Inc.

Agilent VEE is a trademark of Agilent.

FlexPro is a registered trademark of Weisang & Co.

DASYLab is a registered trademark of DATALOG GmbH.

Spectrum reserves the right to make improvements and/or changes to the products and/or programs at any time in order to improve design and to supply the best product possible.

Table of Contents

Preface	5
Versions	5
Product Introduction	6
General Information	6
Software	6
Additional information	6
Order information PCI.248	6
Order information PCI.258	7
Installation	8
System Requirements	8
Hardware Installation	8
Driver Installation	8
DOS	9
Win 98/ME/2000/XP (WDM).....	9
Windows 95 (VXD).....	9
Windows NT (Legacy)	10
Windows NT.....	10
Installation for Linux.....	11
Login	11
Select the right driver	11
Load Driver	11
Major Number	11
Installing the Device	11
End	11
Info	12
Utilities	12
SBench 5.x.....	12
DRVCONFIG.EXE	12
PCITEST.EXE	12
Hardware Description	13
Trigger Information.....	13
Option Multiple Recording	13
Option Gated Sampling.....	13
Block diagram PCI.248 / PCI.258	15
Technical data PCI.248	15
Technical data PCI.258	16
Placement PCI.248 / PCI.258	16
Connectors	16
Software Description	17
General Information	17
Header files on CD	17
DLLTYP.H.....	17
SPECTRUM.H	17
REGS.H	17
ERRORS.H.....	17
Driver functions.....	17
int16 SpclnitPCIBoards (int16* count, int16* PCIVersion).....	18
int16 SpclnitBoard (int16 nr, int16 typ).....	18
int16 SpcSetParam (int16 nr, int32 reg, int32 value)	18
int16 SpcGetParam (int16 nr, int32 reg, int32* value).....	18
int16 SpcSetData (int16 nr, int16 ch, int32 start, int32 len, dataptr data)	18
int16 SpcGetData (int16 nr, int16 ch, int32 start, int32 len, dataptr data)	19
Error Codes.....	19
Valid Board Types	19
Hints for programming the boards.....	19
Software - Register	20
PCI register.....	20
PCI Features register	20
Error registers	21
Status register	21
Command register	21
Memory register	21
Posttrigger register.....	21

Features register	22
Samplerate register.....	22
Triggermode register.....	22
Triggerlevel register	22
Gated Sampling (Option)	23
Double Mem (Option)	24
Multiple Recording (Option)	24
Triggeroffset register (From version 2.3 on).....	24
PulseWidth register (From version 2.3 on)	24
Offset and Gain settings (from hardware version 2.2 on).....	25
Input range register.....	26
Data (Read)	26
Example of Driver use	27
Transfer speed.....	28

Vorwort

Diese Anleitung enthält detaillierte Informationen über die Hardware Möglichkeiten der PCI.248 und der PCI.258 von Spectrum Systementwicklung. Diese Informationen enthalten die technischen Daten, die Spezifikationen, die Beschreibung der Schnittstellen.

Außerdem führt diese Beschreibung durch den Installationsprozess sowohl der Karte als auch der Treiber für das jeweilige Betriebssystem.

Zuletzt enthält dieses Handbuch die komplette Software Beschreibung der Karte und des zugehörigen Treibers. Der Leser wird in die Lage versetzt, diese Karte in einem beliebigen PC System unter einem der unterstützten Betriebssysteme einzusetzen.

Achtung, in diesem Handbuch ist keine Beschreibung der speziellen Treiber für die Produkte von Drittherstellern wie LabVIEW oder MatLab enthalten. Diese Treiber sind nicht im normalen Lieferumfang enthalten.

Neuerungen der Karte, zusätzliche Optionen oder Speicher-ausrüstungen werden auf der Homepage <http://www.spec.de> bekannt gegeben. Hier kann ebenfalls die neueste Treiberversion mit den letzten Fehlerbereinigungen gefunden werden.

Versions

Version	Date	Changes/New options/Enhancements
1.X.		Internal versions.
2.0.		Pilot series: only for selected customers as test products. No longer supported by driver.
2.1.	Until November 2000	First series delivered to customers including all options.
2.2.	Until June 2001	Offset adjusting by user and automatic offset adjusting in the driver minimising offset errors. Enhanced gain amplifier: better gain and bandwidth flatness. Trigger output could be used without 50 Ohm termination. Additional jumper for DC coupling of clock output.
2.3.	Until November 2001	Trigger offset readable Pulsewidth trigger with 7 bit available
2.4.	Current version	Max available memory is doubled to 512 MS. TTL trigger can combine with pulsewidth. Pulsewidth trigger with 9 bit available. For special purpose carry the clock out a fix frequency of max. samplerate / 40.

Preface

This manual provides detailed information on the hardware features of the PCI.248 and PCI.258 from Spectrum Systementwicklung. This information includes specifications, block diagram, connector description.

In addition, this guide takes you through the process of installing your board and also describes the installation of the delivered driver package for each operating system.

Finally this manual provides you with the complete software information of the board and the related driver. The reader of this manual is able to integrate the board in any PC system with one of the supported operating systems.

Please note that in this manual there is no description for specific driver parts like LabVIEW or MatLab software that are not normally enclosed in the hardware.

For any new information on the board as well as new available options or memory upgrades please contact our website <http://www.spec.de>. You will also find the current driver package with the latest bug fixes on our site.

Spectrum reserves the right to make changes at any time in order to improve design and to supply the best product possible.

Product Introduction

Allgemeine Information

Die PCI.248/PCI.258 ist eine schnelle A/D-Wandler Karte für PCI Systeme. Zwei A/D-Wandler ermöglichen es, Signale simultan abzutasten, ohne den Zeitversatz von Multiplexsystemen. Dies ist wichtig, wenn auch die Phase eines Signals Informationen enthält. Verschiedene Betriebsarten, wie Speichersegmentierung, interner/externer Takt und Trigger sowie Pre- und Posttrigger, erlauben eine flexible Anpassung an das Meßproblem.

Der große on-board Speicher ermöglicht die Aufzeichnung extrem breitbandiger Signale. Außerdem ist die Karte Busmasterfähig und in der Lage, die aufgezeichneten Daten automatisch in den CPU-Speicher zu transferieren.

Über die Takt- und Triggerleitungen ist die Karte erweiterbar zu einem vielkanaligen Meßsystem.

Anwendungsbeispiele: Ultraschall, LDA/PDA, Time-of-flight, Radar, Spektroskopie.

Software

Kostenlos mitgeliefert werden Treiber für Linux, DOS und Windows 9x/ME/NT/2000/XP. Für die einfache Programmierung sind Beispiele in C/C++, Delphi und Visual Basic enthalten. Darüber hinaus steht zur komfortablen Steuerung die Signalverarbeitungssoftware SBench 5.2 kostenlos zur Verfügung. Außerdem sind Treiber für LabVIEW, DASyLab, MATLAB und VEE erhältlich.

Additional information

To minimise noise keep the PCI.248 / PCI.258 away from the power supply.

The PCI.248 / PCI.258 operates with components having very high power consumption. Therefore it is highly recommended to place the board near the cooling fan. Do not use the PCI.248 / PCI.258 in hermetic closed systems.

Order information PCI.248

PCI.248 standard	PCI.248 with 32 MSamples memory including drivers	PCI248
Option 64 M	Memory upgrading to 64 MSamples	PCI248-64
Option 256 M	Memory upgrading to 256 MSamples	PCI248-256
Option 512 M	Memory upgrading to 512 MSamples	PCI248-512
Multiple Recording	Memory segmentation for fast repetition rates	PCI248-mr
Gated Sampling	Gated sampling with an external control signal	PCI248-gs
Double Mem	Channel 0 uses memory of both channels for all samplerates	PCI248-dm
DASyLab driver	Drivers for DASyLab 5.0 for Win 95/98, Win 2000 and Win NT	PCI248-dl
HP-VEE driver	Drivers for HP-VEE 5.0 for Win 95/98, Win 2000 and Win NT	PCI248-hp
LabVIEW driver	Drivers for LabVIEW 4.0 for Win 3.11, Win 95/98, Win 2000 and Win NT	PCI248-lv
MATLAB driver	Drivers for MatLab 5.0 for Win 95/98, Win 2000 and Win NT	MATLAB

General Information

The PCI.248/PCI.258 is a fast A/D-Converter for PCI based Systems. Two independent A/D-Converters makes it possible to sample signals simultaneously without the problems of multiplexed systems. This is necessary when the phase of a signal carries interesting information. Different modes, e.g. memory segmentation, internal/external clock and trigger as well as the pre- and posttrigger capability makes it easy to adapt this recorder to the measuring problem.

The large on-board memory allows the recording of signals with extremely wide bandwidth. Additionally the board has bus master capabilities, so it is able to transfer the measured data directly to the PC's memory.

By using the clock- and trigger signals the user may upgrade the board to a multichannel measuring system.

Application examples: Supersonics, LDA/PDA, Time-of-flight, Radar, Spectroscopie.

Software

Drivers for Linux, DOS and Windows 9x/ME/NT/2000/XP as well as programming examples for C/C++, Delphi and Visual Basic are delivered with the board. Comfortable programming, initialising and data display are performed by the free-of-charge Windows program SBench 5.2. Software drivers for LabVIEW, DASyLab, MATLAB and VEE are available.

Order information PCI.258

PCI.258 standard	PCI.258 with 32 MSamples memory including drivers	PCI258
Option 64 M	Memory upgrading to 64 MSamples	PCI258-64
Option 256 M	Memory upgrading to 256 MSamples	PCI258-256
Option 512 M	Memory upgrading to 512 MSamples	PCI258-512
Multiple recording	Memory segmentation for fast repetition rates	PCI258-mr
Gate	Gated sampling with an external control signal	PCI258-gs
Double Mem	Channel 0 uses memory of both channels for all samplerates	PCI258-dm
SBench 5.0	Signal processing software for Win 95/98, Win 2000, Win NT	SBENCH5
DASYLab driver	Drivers for DASYLab 5.0 for Win 95/98, Win 2000 and Win NT	PCI258-dl
HP-VEE driver	Drivers for HP-VEE 5.0 for Win 95/98, Win 2000 and Win NT	PCI258-hp
LabVIEW driver	Drivers for LabVIEW 4.0 for Win 3.11, Win 95/98, Win 2000 and Win NT	PCI258-lv
MatLab driver	Drivers for MatLab 5.0 for Win 95/98, Win 2000 and Win NT	MATLAB

Installation

System Anforderungen

PCI basierter IBM kompatibler PC mit mindestens einem freien PCI Steckplatz in voller Länge. Der PCI-Bus muß mindestens der Revision 2.1. genügen. Die Karte arbeitet nicht in einem PCI-Bus Revision 2.0 oder früher. Wenn mehr als eine Karte im System installiert werden soll, so empfehlen wir einen zusätzlichen Lüfter für die Karten einzusetzen.

Hardware Installation

- (1) Schalten Sie den PC aus.
- (2) Öffnen Sie das Gehäuse.
- (3) Wählen Sie einen freien PCI Steckplatz der benötigten Länge aus. Wenn in Ihrem System kein zusätzlicher Lüfter installiert ist, so ist die beste Wahl ein Steckplatz, in dem die Karte nicht direkt neben einer anderen Karte platziert ist. Wenn Ihr System einen oder mehrere zusätzliche Lüfter besitzt, so platzieren Sie die Karte direkt in deren Luftstrom.
- (4) Installieren Sie die Karte in dem ausgewählten Steckplatz. Achten Sie dabei besonders auf den korrekten Sitz des PCI Steckers im Steckplatz.
- (5) Schrauben Sie die Karte an der Frontblende am Gehäuse fest.
- (6) Wenn Sie eine PCI Karte in voller Baulänge erstanden haben, so liegt Ihrer Karte ein Kartenhalter bei (bei Karten mit SMB Steckern ist dieser bereits montiert). Es wird empfohlen diesen Kartenhalter zu installieren, um die Karte fest im System zu fixieren. Wenn Sie eine Karte mit 9 mm BNC Steckern haben, so ist nur die nachträgliche Montage des Bügels an der bereits installierten Karte mit einem kurzen Schraubendreher möglich.
- (7) Starten Sie das System.
- (8) Wenn Ihr System nicht bootet, überprüfen Sie bitte den korrekten Sitz der Karte in ihrem Steckplatz. Starten Sie danach das System neu.
- (9) Wenn Ihr System immer noch nicht bootet kann es jetzt ein Problem in der Zusammenarbeit mit anderen PCI Karten geben. Deinstallieren Sie bitte alle anderen PCI Karten bis auf die Grafikkarte und versuchen Sie das System in dieser Konfiguration zu starten. Wenn diese Maßnahme zum Erfolg führt, so muß vermutlich die Reihenfolge der PCI Karten in Ihrem System geändert werden.

Treiber Installation

Spectrum liefert einen Kartentreiber aus, der alle Karten unterstützt. Dieser Treiber hat auf allen Betriebssystemen ein einheitliches Interface. Mit Vorstellung der Treiberversion 3.00, die jetzt einen WDM kompatiblen Treiber enthält, mußte eine Unterteilung in PCI und ISA Karten gemacht werden. Bitte wählen Sie den passenden Treiber anhand der Tabelle aus. Wenn Sie ISA und PCI Karten von Spectrum gemischt in einem System benutzen, so nutzen Sie bitte den ISA Treiber.

System Requirements

PCI based IBM PC compatible PC with at least one free full-length PCI slot. The PCI bus version must be at least revision 2.1. The boards will not work with older PCI buses of revision 2.0. If you are installing more than one board in your PC, an additional cooling fan is strongly recommended.

Hardware Installation

- (1) Power off your PC.
- (2) Open the cover.
- (3) Select a free PCI slot of the required length. If you are using a system with no additional cooling fans, it is the best decision to put the board in a slot not adjacent to any other board. If you have a system with additional cooling fans, place the PCI board in front of a cooling fan.
- (4) Install the board in this slot. Make sure that the PCI connector is right struck into the slot.
- (5) Use a screw to fix the bracket to the PC.
- (6) If your board has full PCI length a retainer is delivered with the board (on boards with SMB connectors this retainer is already installed). It is recommended to use this retainer to fix the board in the system. If you have a board with 9 mm BNC connectors, it is not possible to install the retainer before inserting the board in the system. You need to install the retainer with a short screwdriver to the already installed board.
- (7) Reboot the system.
- (8) If your system will not boot, please check whether the board is struck correctly into the PCI connector and reboot again.
- (9) If your system will not boot after this, there may be a problem with other PCI boards. Please de-install all other PCI boards and try to boot the system without them. If this works, you may have to change the order of the PCI boards in the system.

Driver Installation

Spectrum supplies one driver that supports all boards with an unique interface for all operating systems. With introduction of the new version 3.00 which includes a WDM style driver there has been a separation made between PCI and ISA boards. Please use the matching driver that is listed in the table. If you mix ISA and PCI boards from Spectrum in your system you need to use the ISA driver.

Operating System	PCI boards only	PCI and ISA mixed	ISA boards only
Windows XP	WDM driver (Windows 98/ME/2000/XP)	Legacy driver (Windows NT)	Legacy driver (Windows NT)
Windows 2000	WDM driver (Windows 98/ME/2000/XP)	Legacy driver (Windows NT)	Legacy driver (Windows NT)
Windows ME	WDM driver (Windows 98/ME/2000/XP)	VXD driver (Windows 95)	VXD driver (Windows 95)
Windows 98	WDM driver (Windows 98/ME/2000/XP)	VXD driver (Windows 95)	VXD driver (Windows 95)
Windows NT	Legacy driver (Windows NT)	Legacy driver (Windows NT)	Legacy driver (Windows NT)
Windows 95	VXD driver (Windows 95)	VXD driver (Windows 95)	VXD driver (Windows 95)

DOS	OBJ files (DOS driver)	OBJ files (DOS driver)	OBJ files (DOS driver)
Linux	Linux Kernel Module	Linux Kernel Module	Linux Kernel Module

DOS

Der Treiber für DOS besteht aus einem Satz Objektdateien zum Linken in ein DOS Programm. Die Treiber Dateien können auf der CD im Verzeichnis \DRIVER\DOS auf der Diskette gefunden werden. Beispiele zur Nutzung von Borland C++ 3.1 sind ebenfalls vorhanden. Zur Benutzung der Treiber müssen nur die Objekt Dateien *.OBJ und die Header Dateien *.H ins Arbeitsverzeichnis kopiert werden.

Wenn die Beispielprogramme bei der Arbeit mit DOS nicht laufen, so kann es hier zu einem Problem mit den im System installierten Software Treibern gekommen sein. Starten Sie das System erneut ohne irgendwelche installierten Treiber. Wenn das Programm so läuft, fügen Sie Schritt für Schritt Ihre Treiber wieder in das System ein, um den problematischen Treiber herauszufinden.

Auf einigen Motherboards kann es zu Problemen mit älteren Versionen der Datei EMM386.EXE kommen. Die Version 6.22 läuft hier korrekt. Es kann daher nötig sein, diese Datei gegen eine neuere Version auszutauschen.

Win 98/ME/2000/XP (WDM)

Wenn das Betriebssystem Windows 98, Windows ME, Windows 2000 oder Windows XP installiert ist, wird die PCI Karte nach dem nächsten Start automatisch erkannt. Das System bietet die direkte Installation eines Treibers für die Karte an. Wählen Sie hier als Installationsquelle die mitgelieferte CD. Die Treiberdateien befinden sich im Verzeichnis \Driver\Win98_2k_XP. Die Treiber stehen sofort nach der Installation ohne Neustart des Systems zur Verfügung.

Die Treiber bestehen aus einer 32 Bit DLL, die alle Funktionen des Treibers enthält, und einem WDM-Kernel-Treiber (SYS). Die DLL kann mit allen Systemen benutzt werden, die eine Schnittstelle zu 32 Bit Windows DLLs anbieten. Beispiele für Microsoft Visual C++, Borland Delphi und Microsoft Visual Basic sind ebenfalls enthalten.

Falls Sie Visual C++ benutzen, so ist es möglich, die Library Datei SPECTRUM.LIB mit in ein Projekt zu integrieren, um die Funktionen des Treibers auf einfache Weise in das Programm einzubinden. Die Library Datei arbeitet nicht mit Borland Compilern zusammen.

Die beiden DLL's unterscheiden sich nur im Aufruf der Funktionen. Die Datei SPECTRUM.DLL exportiert die Funktionen als _cdecl (für C, C++, Delphi), die Datei SPCSTD95.DLL als _stdcall (für Visual Basic). Je nach benutztem Compiler kann eine der beiden DLL's benutzt werden.

Windows 95 (VXD)

Die Treiber für Windows 95 bestehen aus einer 32 Bit DLL, die alle Funktionen des Treibers enthält, und einem Virtual Device Driver (VXD). Die DLL kann mit allen Systemen benutzt werden, die eine Schnittstelle zu 32 Bit Windows DLLs anbieten. Beispiele für Microsoft Visual C++, für Borland Delphi und für Visual Basic sind ebenfalls enthalten.

Zur Installation des Treibers benutzen Sie bitte die auf der CD enthaltene Installationsversion im Verzeichnis /Install/Win95Drv. Hiermit werden alle Treiberdateien in die vorgesehenen Verzeichnisse installiert.

Falls Sie Visual C++ benutzen, so ist es möglich, die Library Datei SPECTRUM.LIB mit in ein Projekt zu integrieren, um die Funktionen des Treibers auf einfache Weise in das Programm

DOS

The driver consists of a set of object files ready to link to a DOS program. The driver files are found on CD in the directory \DRIVER\DOS on the driver disk. Examples for the use with Borland C++ 3.1 are included. To use the driver files, just copy the object *.OBJ and header *.H files to your working directory. If the example files are not working when using DOS operating system, there may be problems with the installed software drivers. Start the system once again without any software drivers installed. After this install the drivers step by step to find out the problematic software driver. On some motherboards, there may be problems when using older versions of EMM386.EXE. The version 6.22 works correctly. It may be necessary to update this driver to a higher version.

Win 98/ME/2000/XP (WDM)

When the operating system Windows 98, Windows ME, Windows 2000 or Windows XP is installed, the PCI board will be automatically recognised after the rebooting. The system will ask for a driver to be installed. Select the install directory from the Spectrum driver CD. The driver files are placed in the directory \Driver\Win98_2k_XP. The driver is ready to use directly after installing, no reboot is necessary.

The driver consists of a 32 bit windows DLL which includes all functions of the driver and a WDM kernel driver (SYS). The DLL can be used with all systems which accept 32 bit windows DLL's. Examples for Microsoft Visual C++ 4.x, Borland Delphi and Microsoft Visual Basic are included.

If you are using Microsoft Visual C++, you may use the delivered library file SPECTRUM.LIB to access the driver functions easily. The library file will not work with Borland compilers.

The only difference between the both DLL's is the calling convention. The file SPECTRUM.DLL uses _cdecl definition (for C, C++, Delphi), the file SPCSTD95.DLL uses _stdcall definition (for Visual Basic). Depending on the used programming language, one of the two DLL's may be used.

Windows 95 (VXD)

The driver consists of a 32 bit windows DLL which includes all functions of the driver and a virtual device driver (VXD). The DLL can be used with all systems which accept 32 bit windows DLL's. Examples for Microsoft Visual C++ 4.x, Borland Delphi and Microsoft Visual Basic are included.

You need to use the install program for driver installation. The program is located on CD in the directory /Install/Win95Drv. The program installs all driver files in the correct directory.

If you are using Microsoft Visual C++, you may use the delivered library file SPECTRUM.LIB to access the driver functions easily. The library file will not work with Borland compilers.

The only difference between the both DLL's is the calling convention. The file SPECTRUM.DLL uses _cdecl definition (for C,

einzubinden. Die Library Datei arbeitet nicht mit Borland Compilern zusammen.

Die beiden DLL's unterscheiden sich nur im Aufruf der Funktionen. Die Datei SPECTRUM.DLL exportiert die Funktionen als `_cdecl` (für C, C++, Delphi), die Datei SPCSTD95.DLL als `_stdcall` (für Visual Basic). Je nach benutztem Compiler kann eine der beiden DLL's benutzt werden.

Windows NT (Legacy)

Der Treiber besteht aus einem Kernel Mode Treiber für Windows NT und einer 32 Bit DLL, die die Funktionen des Kernel Mode Treibers benutzt. Beispiele für Microsoft Visual C++, Borland Delphi und Microsoft Visual C++ werden ebenfalls mitgeliefert.

Windows NT

- (1) Loggen Sie sich als ADMINISTRATOR oder als ein Benutzer mit dem Recht Treiber zu installieren und die Registry zu ändern in Ihr System ein.
- (2) Starten Sie das Setup Programm auf der Treiber CD. Sie finden das Installationsprogramm im Verzeichnis `\Install\WinNTDrv`.
- (3) Das Installationsprogramm installiert den Kernel Mode Treiber und die 32 Bit Windows DLL, sowie einige Hilfsprogramme im Verzeichnis 'Spectrum GmbH'. Die Registry wird ebenfalls angepaßt.
- (4) Starten Sie den Computer neu.
- (5) Die PCI Karten werden automatisch vom Kernel erkannt und eingetragen.
- (6) Falls der Geräte Treiber nicht korrekt startet (Eine Nachricht im Event Log von der Datei SPCDRV.SYS), ist der Treiber nicht korrekt konfiguriert. Bitte überprüfen Sie, ob mit dem Programm DRVCONFIG.EXE, ob die Standard Karte „PCI Board“ eingetragen ist.

Falls Sie Visual C++ benutzen, so ist es möglich, die Library Datei SPECTRUM.LIB mit in ein Projekt zu integrieren, um die Funktionen des Treibers auf einfache Weise in das Programm einzubinden. Die Library Datei arbeitet nicht mit Borland Compilern zusammen.

Es werden die beiden DLL's SPECTRUM.DLL und SPCSTDNT.DLL installiert. Die beiden DLL's unterscheiden sich nur im Aufruf der Funktionen. Die Datei SPECTRUM.DLL exportiert die Funktionen als `_cdecl` (für C, C++, Delphi), die Datei SPCSTDNT.DLL als `_stdcall` (für Visual Basic). Je nach benutztem Compiler kann eine der beiden DLL's benutzt werden.

C++, Delphi), the file SPCSTD95.DLL uses `_stdcall` definition (for Visual Basic). Depending on the used programming language, one of the two DLL's may be used.

Windows NT (Legacy)

The driver consists of a kernel mode driver for Windows NT and a 32 bit windows DLL which uses the functions of the kernel mode driver. Examples for Microsoft Visual C++, Borland Delphi and Microsoft Visual Basic are included.

Windows NT

- (1) Login as ADMINISTRATOR or with another account having the right to install drivers and to change the registry.
- (2) Start the setup program on the driver CD. The installation program is found in the directory `\Install\WinNTDrv`.
- (3) The installation routine will install the kernel mode driver, the 32 bit windows DLL and some utilities in the program folder 'Spectrum GmbH'. It will also update the registry.
- (4) Restart the computer
- (5) The PCI boards are automatically detected by the kernel driver.
- (6) If the service does not start correct (A message in the event log from the service SpcDrv.SYS), the driver is not setup correctly. Please run DRVCONFIG.EXE and check whether the standard board is correctly set to "PCI Board"

If you are using Microsoft Visual C++, you may use the delivered library file SPECTRUM.LIB to access the driver functions easily. The library file will not work with Borland compilers. The both DLL's SPECTRUM.DLL and SPCSTDNT.DLL are installed. The only difference between the both DLL's is the calling convention. The file SPECTRUM.DLL uses `_cdecl` definition (for C, C++, Delphi), the file SPCSTD95.DLL uses `_stdcall` definition (for Visual Basic). Depending on the used programming language, one of the two DLL's may be used.

Installation für Linux

Der Treiber besteht aus einem ladbaren Kernel Modul für alle Karten. Beispiele für Gnu C werden ebenfalls mitgeliefert.

Login

Loggen Sie sich als root ein oder als Benutzer mit dem Recht Module zu laden und Devices anzulegen.

Auswahl des richtigen Treibers

Die Verwendung von Linux-Kernel-Modulen hängt stark von der Kernelversion sowie der verwendeten Distribution ab. Diesem Umstand Rechnung tragend werden die Spectrum Treiber in verschiedenen Versionen ausgeliefert. Bitte wählen Sie das am besten passende Archiv für Ihre Installation aus.

Treiber laden

Der Linux Treiber wird als ladbares Kernel Modul spc.o ausgeliefert. Der Treiber enthält alle Spectrum PCI, CompactPCI und ISA Karten. Die PCI und CompactPCI Karten werden automatisch erkannt.

Laden Sie das Modul mit „insmod -f spc.o“.

Der insmod Befehl kann die Warnung generieren, daß das Kernel Modul für eine andere Kernel Version kompiliert wurde. Dies Meldung können Sie ignorieren.

Wenn das Kernel-Modul nicht in Ihre Linux Installation geladen werden kann, so ist es notwendig den Treiber auf Ihrem System neu zu kompilieren. Bitte setzen Sie sich mit Spectrum in Verbindung, um die benötigten Sourcedateien zu bekommen.

Major Number

Für den Zugriff auf den Treiber benötigen Sie die zugeteilte Major number. Sie finden diese Zahl in /proc/devices. Der Treiber trägt den Namen „spec“. Normalerweise ist diese Nummer 254, kann aber auch je nach vorher installierten Treibern davon abweichen.

Device anlegen

Als letzten Schritt muß ein Device mit dem Treiber verknüpft werden. Dieses geschieht über den Befehl mknod. Als Major number wird die in /proc/devices gefundene Zahl eingetragen. Als Minor Number der Index der Karte die angesprochen wird. Die Indexzählung beginnt bei 0.

„mknod /dev/spc0 c 254 0“ für die erste Karte
 „mknod /dev/spc1 c 254 1“ für die zweite Karte

Stellen Sie sicher, daß alle Benutzer, die mit dem Treiber arbeiten müssen Schreibrechte für das neu angelegte Device haben. Dafür können Sie allen Personen Schreibrechte für das Device erteilen: `chmod a+w /dev/spc0`.

Ende

Die Karte kann jetzt über das angelegte Device angesprochen werden. Das genaue Vorgehen kann aus den Beispielen entnommen werden.

Nach einem Neustart von Linux ist es nur nötig das Treiber Modul zu laden. Das Device muß nur geändert werden, falls die Major Number nicht mehr stimmt.

Installation for Linux

The driver consists of a loadable kernel module for all boards. Examples for Gnu C are also delivered.

Login

Login as root or login as a user who has the right to load modules and to install devices.

Select the right driver

Linux kernel modules are heavily depending on the kernel version and distribution. Therefore the kernel driver for the Spectrum boards is shipped in different versions. Please select the archive that is best matching your installed version.

Load Driver

The linux driver is shipped as the loadable kernel module spc.o. The driver includes all Spectrum PCI, CompactPCI and ISA boards. All PCI and CompactPCI boards are recognised automatically.

Load the module with “insmod -f spc.o”

The insmod command could generate a warning that the driver module was compiled for an other kernel version. You could ignore this warning.

It is not possible to use the driver module with linux versions prior to kernel version 2.0.

If the kernel module could not be loaded in your linux installation it is necessary to compile the driver directly on your system. Please contact Spectrum to get the needed source files.

Major Number

For accessing the device driver it is necessary to know the major number of the driver. This number is listed in /proc/devices. The device driver is called “spec” in this list. Normally this number is 254 but this depends on the already installed device drivers.

Installing the Device

You connect a device to the driver with the mknod command. The major number is the number found in /proc/devices. The minor number is the index of the board starting with 0.

“mknod /dev/spc0 c 254 0” for the first board
 “mknod /dev/spc1 c 254 1” for the second board

Make sure that the users that should work with the driver has write rights to access the device. Therefore you should give all persons all rights to the device: `chmod a+w /dev/spc0`

End

The board could now be accessed using the device. See the example files for more information.

After restarting linux it is only necessary to load the driver again. The device must only be changed if the major number has changed.

Der Zugriff auf das Linux Device erfolgt mit Read und Write Befehlen sowie ioctl Befehlen. Eine Umsetzung dieser Befehle in die Standard Treiber Schnittstelle von Spectrum kann über die Datei „spciocctl.inc“ realisiert werden. Das genaue Vorgehen ist aus den Beispielen ersichtlich.

Info

Informationen über die installierte Spectrum Karten können unter /proc/spectrum abgefragt werden. Für ISA Karten ist hier der Typ und die Basisadresse sichtbar. Für PCI Karten sind alle grundlegenden Informationen aus dem onboard EEPROM aufgelistet.

Hilfsprogramme

SBench 5.x

Auf der CD wird eine Vollversion von SBench 5.x mitgeliefert. Das Programm unterstützt alle aktuellen Erfassungs-, Ausgabe- und Digital I/O Karten von Spectrum. Je nach verwendeter Karte und nach Konfiguration des Programms kann SBench als Digitales Speicheroszilloskop, als Spectrumanalyser, als Logikanalyser oder einfach als Datenerfassungssystem benutzt werden. Verschiedenen Import- und Exportfunktionen erlauben die einfache Nutzung von SBench mit diversen anderen Programmen.

Eine Installationsversion ist im Verzeichnis /Install/SBench5 auf der CD zu finden. Im Verzeichnis /Manuals auf der CD ist eine kurze Anleitung zur Bedienung von SBench in Deutsch und Englisch zu finden. Eine aktuelle Version ist jederzeit aus dem Internet unter www.spec.de zu bekommen.

DRVCONFIG.EXE

Automatisch installiert im Ordner ‚Spectrum GmbH‘ bei der Installation des Windows NT Treibers. Dieses Programm erlaubt die Änderung der Treiber Konfiguration der Spectrum ISA Karten unter Windows NT. Für PCI Karten braucht das Programm nicht benutzt werden. Das Programm ändert die Eintragungen in der Registry. Die neue Konfiguration wird beim nächsten Start des Systems benutzt.

PCITEST.EXE

Zu finden auf der Treiber CD im Verzeichnis \UTILS. Dieses Hilfsprogramm sammelt alle verfügbaren Informationen über alle im System installierten Spectrum PCI Karten. Die Informationen werden aus dem on-board EEPROM ausgelesen und angezeigt. Das Programm läuft nur unter DOS oder in der DOS-Box von Windows 3.11 oder Windows 9x/ME. Das Programm läuft nicht unter Windows NT/2000/XP.

Accessing the linux device is done with read and write commands and ioctl commands. These commands could be converted to the standard Spectrum driver interface with the file “spciocctl.inc”. See the examples for this.

Info

Information about the installed boards could be found in the /proc/spectrum file. For ISA boards the board type and the base address are listed. For PCI boards the basic information from the onboard EEPROM is listed.

Utilities

SBench 5.x

A full version of SBench 5.x is delivered with the board on CD. The program supports all actual acquisition, generator and digital I/O boards from Spectrum. Depending on the used board and the software setup, one could use SBench as a digital storage oscilloscope, a spectrum analyser, a logic analyser or simply as a data recording front end. Different export and import formats allow the use of SBench together with a variety of other programs.

An install version of the program is found in the directory /Install/SBench5 on CD. There is also a short program description in german and english in the /Manuals directory. A current version could be downloaded from the internet at www.spec.de at any time.

DRVCONFIG.EXE

Installed in the folder ‚Spectrum GmbH‘ when installing the Windows NT driver. This utility manages the driver configuration of the Spectrum ISA boards for Windows NT. The program need not to be used for PCI boards. The utility changes the registry. The new configuration will only be used after the next reboot of the system.

PCITEST.EXE

Found on the driver CD in the directory \UTILS. This utility will collect some information about all installed Spectrum PCI boards. The information of the onboard EEPROM will be read out and shown. The utility will only work with DOS, Windows 3.1x, Windows 9x and Windows ME. It will not work with Windows NT/2000/XP.

Hardware Description

Trigger Informationen

Nach dem Start der PCI.248/PCI.258 werden die Eingänge abgetastet und die konvertierten Daten im Speicher abgelegt (Der Speicher arbeitet als Ringbuffer, die Daten werden kontinuierlich in den Speicher geschrieben). Wenn für die Aufnahme Pretrigger programmiert ist, werden die Triggerereignisse ignoriert bis der programmierte Speicher einmal komplett mit Daten gefüllt ist. Danach wird die Triggerverarbeitung freigeschaltet.

Wenn der Software Trigger ausgewählt wurde, wird sofort ein Triggerereignis erkannt. Wird der TTL Trigger benutzt, so wird ein Triggerereignis erkannt, wenn das TTL Signal am Eingang (Connector 4) von LOW Pegel zu HIGH Pegel wechselt (steigende Flanke) oder von HIGH Pegel zu LOW Pegel wechselt (fallende Flanke).

Ist der Kanaltrigger (Kanal 0 oder Kanal 1) selektiert, so werden die vier höchstwertigen Bits des AD-Wandlers mit dem im Triggerlevel Register programmierten Wert verglichen. Wird der Wert vom anliegenden Signal gekreuzt (Von kleiner zu größer bei steigender Flanke, oder von größer zu kleiner bei fallender Flanke), wird ein Triggerereignis erkannt.

Der Status ändert sich zu ‚Trigger found‘ und der Postcounter fängt an den programmierten Posttrigger Wert herunter zu zählen. Wenn dieser Wert Null erreicht, stoppt die PCI.248/PCI.258 und der Status ändert sich auf ‚Ready‘.

Trigger Information

After the PCI.248/PCI.258 has been started it samples the input signals and stores the converted data to the memory. (The memory operates as a circular buffer, so data are written continuously to the RAM). If pretrigger is programmed, no trigger events are processed until the programmed memory is filled one time completely with data. Afterwards the trigger sequencer will be enabled.

If Software trigger is used a trigger event is detected immediately. Using the TTL trigger will cause a trigger event if the external TTL input will go from low to high (rising edge) or from high to low (falling edge).

When the triggermode is set to channel-trigger (channel 0/1) the 4 MSB's of the ADC are compared to the programmed value in the trigger-level-register. If the value is crossed by the signal (less to greater when rising edge, or greater to less when falling edge) a trigger event is detected.

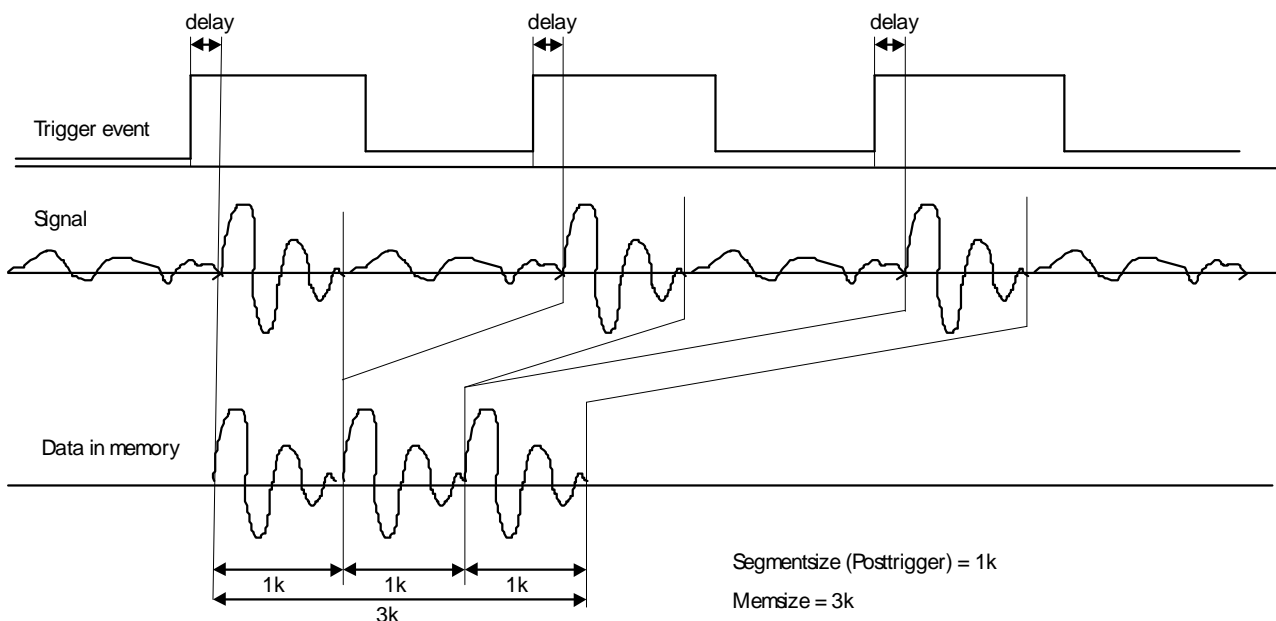
The status will be change to 'trigger found' and the postcounter starts counting down the posttrigger value. After the postcounter reaches zero the PCI.248/PCI.258 stops and signals 'ready' in the status register,

Option Multiple Recording

Die Option Multiple Recording erlaubt die Aufnahme/Ausgabe mehrerer Triggerereignisse, ohne die Hardware dazwischen neu zu starten. Der Speicher der Karte wird in mehrere gleich große Segmente unterteilt. Jedes Segment wird bei Auftreten eines Triggerereignisses mit Daten gefüllt. Im Multiple Recording Modus ist kein Pretrigger möglich.

Option Multiple Recording

The option Multiple Recording allows the recording/replay of several trigger events without restarting the hardware. The memory of the board will be divided into several segments of the same size. Each segment will be filled with data when a trigger event occurs. Pretrigger is not available when using Multiple Recording

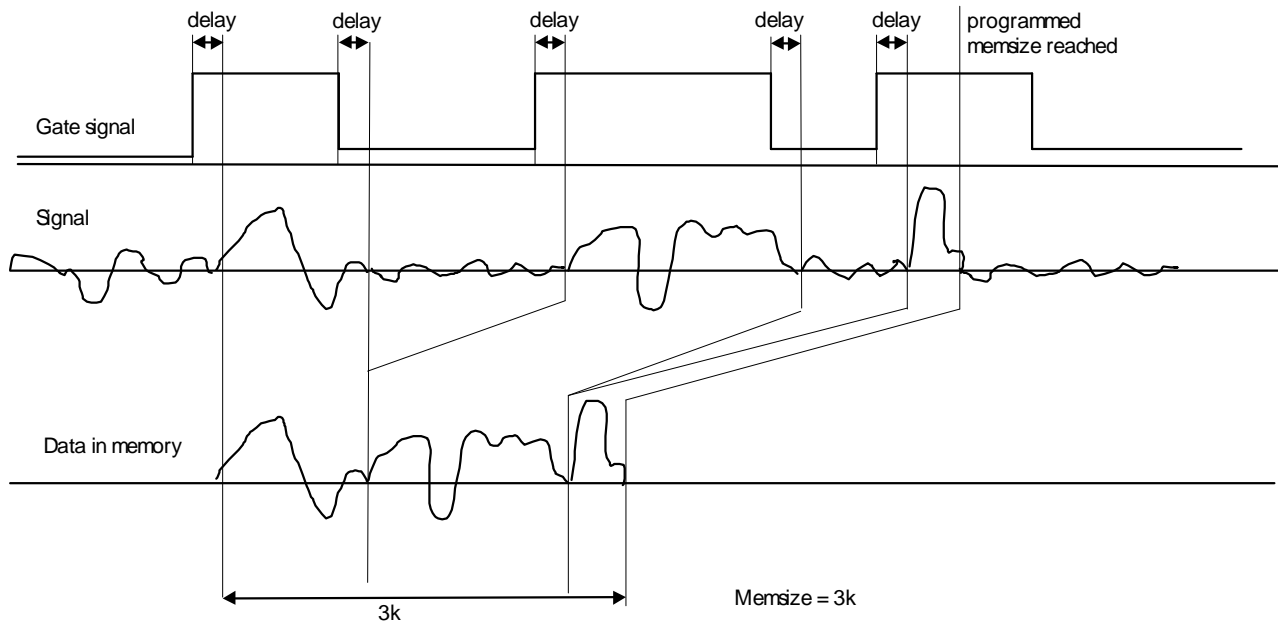


Option Gated Sampling

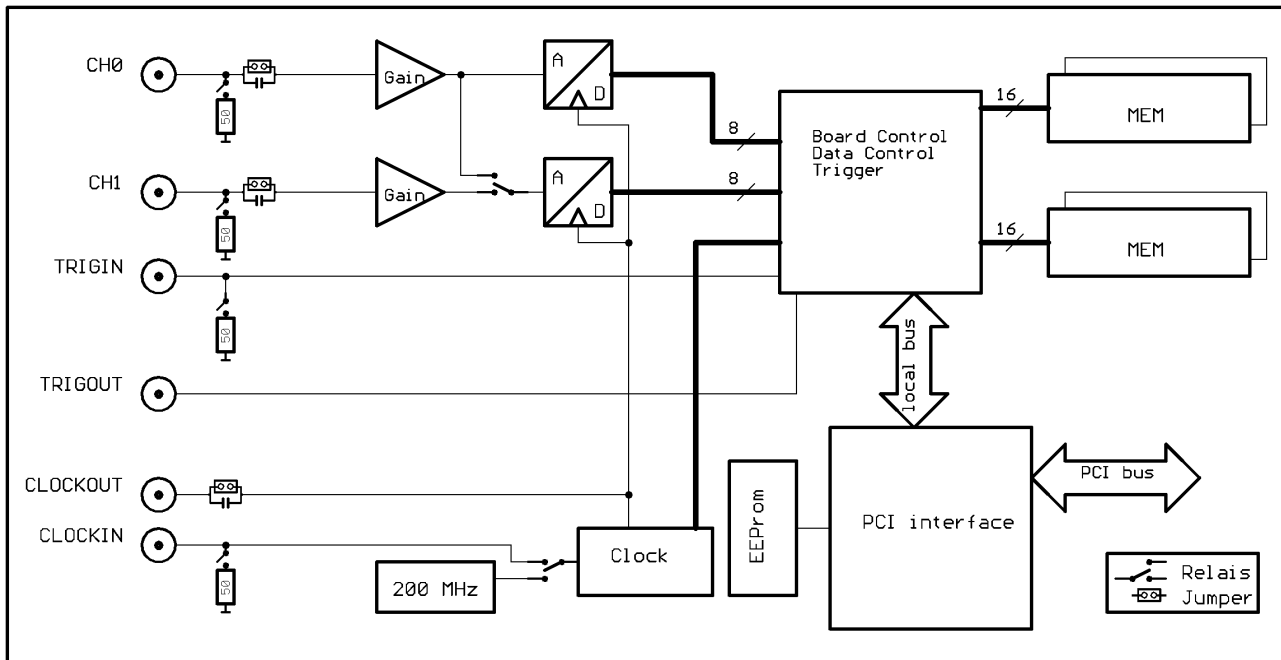
Option Gated Sampling

Die Option Gated Sampling erlaubt die Aufnahme/Wiedergabe eines Signals gesteuert über ein Gatesignal. Es werden nur Daten aufgenommen, wenn das Gatesignal einem programmierten Pegel (TTL HIGH oder TTL LOW) entspricht.

The option Gated Sampling allows recording/replay of a signal controlled by a gate signal. Data is only recorded if the gate signal is equal to a programmed level (TTL HIGH or TTL LOW).



Block diagram PCI.248 / PCI.258



Technical data PCI.248

Resolution	8 bit
Samplerate (internal clock)	2 MS/s up to 400 MS/s
Samplerate (external clock)	25 MS/s up to 200 MS/s
Bandwidth DC -3 dB	0 Hz to \geq 150 MHz
Bandwidth AC -3 dB	40 Hz to \geq 150 MHz
Differential linearity error	$\leq \pm 1.5$ LSB (ADC)
Integral linearity error	$\leq \pm 1.5$ LSB (ADC)
SNR fs= 70 MHz, fck=200 MS/s	45 dB typ. (ADC)
ENOB fs = 20 MHz, fck=200 MS/s	6.8 bit typ. (ADC)
ENOB fs = 70 MHz, fck=200 MS/s	6.8 bit typ. (ADC)
Aperture jitter	2.3 ps rms (ADC)
Input impedance	50 Ohm / 1 MOhm 25 pF
Multi: Trigger to 1 st sample delay	19 to 44 (fix)
Multi: Recovery time	\leq 20 samples
ext. clock input	ECL, AC coupled
ext. Trigger accuracy (\leq 200 MS/s)	2 Samples
ext. Trigger accuracy (400 MS/s)	4 Samples
int. Trigger accuracy	1 Sample
Ext. clock: delay to internal clock	< 3 ns

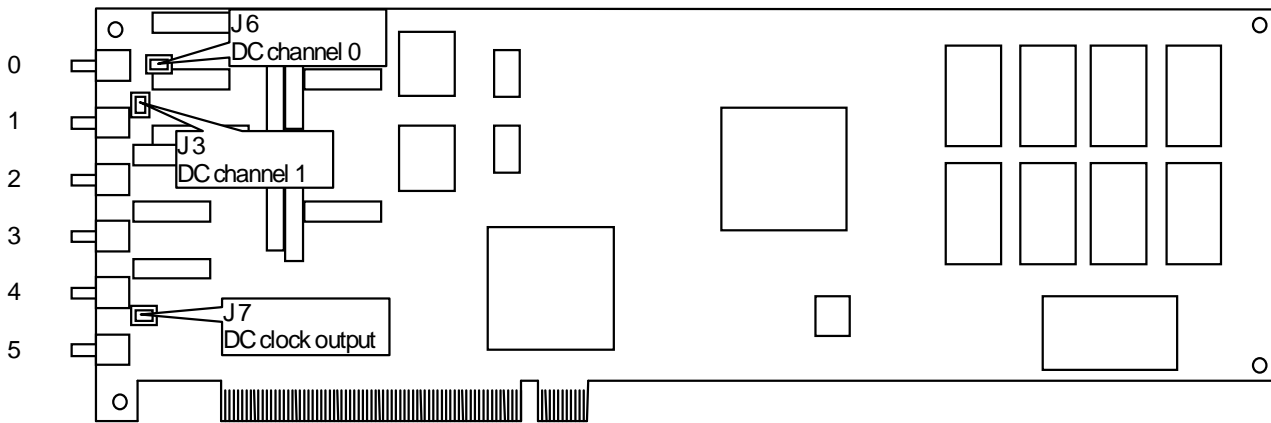
Input range	± 100 mV ... ± 500 mV	± 1 V ... ± 5 V		
Offset error	adjustable by user			
Gain error (\leq 200 MS/s)	$\leq \pm 2$ %	$\leq \pm 3$ %		
Gain error (400 MS/s)	$\leq \pm 2$ %	$\leq \pm 3$ %		
Noise (\leq 200 MS/s)	$\leq \pm 1$ LSB	$\leq \pm 1$ LSB		
Noise (400 MS/s)	$\leq \pm 2$ LSB	$\leq \pm 2$ LSB		
Crosstalk (1 MHz square)	≤ -48 dB	≤ -48 dB		
Dimension	312 mm x 107 mm			
Connector	3 mm SMB male			
Warm up time	10 minutes			
Operating temperature	0°C - 50°C			
Storage temperature	-10°C - 70°C			
Humidity	10% to 90% non condensing			
Overvoltage protection	± 20 V			
	$+3.3$ V	$+5$ V	-12 V	$+12$ V
Power consumption (A)	0 mA	2500 mA	100 mA	10 mA
Power consumption (W)	0.0 W	12.5 W	1.2 W	0.12 W

Technical data PCI.258

Resolution	8 bit
Samplerate (internal clock)	2.5 MS/s up to 500 MS/s
Samplerate (external clock)	25 MS/s up to 250 MS/s
Bandwidth DC -3 dB	0 Hz to \geq 190 MHz
Bandwidth AC -3 dB	40 Hz to \geq 190 MHz
Differential linearity error	$\leq \pm 1.5$ LSB (ADC)
Integral linearity error	$\leq \pm 1.5$ LSB (ADC)
SNR $f_s = 70$ MHz, $f_{ck} = 250$ MS/s	44 dB typ. (ADC)
ENOB $f_s = 20$ MHz, $f_{ck} = 250$ MS/s	6.7 bit typ. (ADC)
ENOB $f_s = 70$ MHz, $f_{ck} = 250$ MS/s	6.7 bit typ. (ADC)
Aperture jitter	2.3 ps rms (ADC)
Input impedance	50 Ohm / 1 MOhm 25 pF
Multi: Trigger to 1 st sample delay	19 to 44 (fix)
Multi: Recovery time	\leq 20 samples
ext. clock input	ECL, AC coupled
ext. Trigger accuracy (≤ 250 MS/s)	2 Samples
ext. Trigger accuracy (500 MS/s)	4 Samples
int. Trigger accuracy	1 Sample
Ext. clock: delay to internal clock	$<$ 3 ns

Input range	± 100 mV ... ± 500 mV	± 1 V ... ± 5 V		
Offset error	adjustable by user			
Gain error (≤ 250 MS/s)	$\leq \pm 2$ %	$\leq \pm 3$ %		
Gain error (500 MS/s)	$\leq \pm 2$ %	$\leq \pm 3$ %		
Noise (≤ 250 MS/s)	$\leq \pm 1$ LSB	$\leq \pm 1$ LSB		
Noise (500 MS/s)	$\leq \pm 2$ LSB	$\leq \pm 2$ LSB		
Crosstalk (1 MHz square)	≤ -48 dB	≤ -48 dB		
Dimension	312 mm x 107 mm			
Connector	3 mm SMB male			
Warm up time	10 minutes			
Operating temperature	0°C - 50°C			
Storage temperature	-10°C - 70°C			
Humidity	10% to 90% non condensing			
Overvoltage protection	± 20 V			
Power consumption (A)	+3.3 V	+5 V	-12 V	+12 V
Power consumption (W)	0 mA	2700 mA	100 mA	10 mA
	0.0 W	13.5 W	1.2 W	0.12 W

Placement PCI.248 / PCI.258



Connectors

The PCI.248/PCI.258 has six 3 mm SMB connectors.

- Connector 0: analogue channel 0.
- Connector 1: analogue channel 1.
- Connector 2: triggermode = TTLPOS or TTLNEG: external trigger input
 triggermode = GATELOW or GATEHIGH: external gate input
- Connector 3: trigger output must be terminated with 50 Ohm to GND.
- Connector 4: clock input 25 MHz up to 200 MHz
- Connector 5: clock output must be terminated with 50 Ohm to GND.
 clock = 200 MHz/400 MHz: clock output internal 200 MHz
 clock = $<$ 200 MHz: clock output internal clock at run time only
 clock = external: clock output internal 200 MHz

Software Description

Allgemeine Information

Der Spectrum Treiber besteht aus einem Satz Funktionen zur Manipulation der Register auf der Karte und zum Daten Transfer in beide Richtungen. Es gibt nur einen Treiber für alle Karten von Spectrum. Abhängig von der Funktionalität der Karte und dem benutzten Bus werden nicht alle Funktionen des Treibers von allen Karten unterstützt. Die unterschiedliche Funktionalität der Karten ist mit Hilfe von kartenspezifischen Registern realisiert. Der Treiber ist für verschiedene Betriebssysteme erhältlich und wird unter allen Betriebssystemen auf die gleiche Art und Weise programmiert.

Header Dateien auf CD

DLLTYP.H

Enthält alle Plattform spezifischen Definitionen der Datentypen und der Funktionsdeklarationen. Alle Datentypen basieren auf diesen Definitionen.

SPECTRUM.H

Definiert die sechs Funktionen des Treibers. Alle Definitionen sind aus der Datei DLLTYP.H entnommen. Die Funktionen selbst werden weiter unten beschrieben.

REGS.H

Definiert alle Register und Kommandos, die im Spectrum Treiber für die verschiedenen Karten benutzt werden. Die Register, die von einer Karte benutzt werden sind weiter unten im kartenspezifischen Teil beschrieben.

ERRORS.H

Listet alle möglichen Errorcodes der Funktionen auf.

Funktionen des Treibers

Der Spectrum Treiber besteht aus den folgenden sechs Funktionen. Die Funktionen sind in der Header-Datei SPECTRUM.H definiert. Abhängig von dem Funktionsumfang der Karte und dem verwendeten Bussystem sind nur einige der Funktionen für die spezielle Karte notwendig. Bei einigen Karten werden nicht alle Parameter der Funktion unterstützt.

General Information

The SPECTRUM driver consists of a set of functions to manipulate registers on the board and to transfer data from or to the board. There is only one driver for all the SPECTRUM boards. Depending on the functionality of the board and the used bus not all functions will be implemented for all boards. The different functionality of the boards is implemented with the help of board specific registers. The driver is available for different operating systems but will be programmed the same way on all operating systems.

Header files on CD

DLLTYP.H

Includes the platform specific definitions for data types and function declarations. All data types are based on this definitions.

SPECTRUM.H

Defines the six functions of the driver. All definitions are taken from the file DLLTYP.H. The functions itself are described below.

REGS.H

Defines all registers and commands which are used in the SPECTRUM driver for the different boards. The registers a board uses are described in the board specific part of the documentation.

ERRORS.H

Lists all possible error codes of the functions.

Driver functions

The SPECTRUM driver consists of the following six functions. The functions are declared in the header file SPECTRUM.H. Depending on the functionality of the board and the used bus only some of the functions are used for the specific board. Not all board specific drivers will interpret all parameters of a function.

	PAD52	PAD82a/b	PAD242	PCI.412	PCI.212	PCI.208	CPCI.208	PCI.248	PCI.258	PCI.DIO32	PAD1232	PAD1616	PAD164	DAPI16	PCK400	TRS582	PADCO06	MI.30xx	MI.31xx	MI.40xx	MI.45xx	MI.60xx	MI.70xx
SpcInitPCIBoards	-	-	-	+	+	+	+	+	+	+	-	-	-	-	-	-	-	+	+	+	+	+	+
SpcInitBoard	+	+	+	-	-	-	-	-	-	-	+	+	+	+	+	+	+	-	-	-	-	-	-
SpcSetParam	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
SpcGetParam	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+
SpcSetData	-	-	-	-	-	-	-	-	-	+	-	-	-	+	-	+	-	-	-	-	-	+	+
SpcGetData	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	+	+	+	+	-	+

int16 SpcInitPCIBoards (int16* count, int16* PCIVersion)

<i>count</i>	adr of 16 bit integer	number of found PCI boards
<i>PCIVersion</i>	adr of 16 bit integer	found PCI version
<i>return</i>	16 bit integer	error code of function like listed below

Initialises all installed PCI boards. The board numbers will start with zero. The number of PCI boards will be given back in the value *Count*. All installation parameters will be read from the hardware.

Using Windows NT the boards are already installed in the registry. This function just gives back the values of the kernel driver.

Linux initialises the boards while loading the kernel module. This function is not available under Linux.

int16 SpcInitBoard (int16 nr, int16 typ)

<i>nr</i>	16 bit integer	number of the board to be defined in range 0-15
<i>typ</i>	16 bit integer	type of the defined board listed in REGS.H
<i>return</i>	16 bit integer	error code of function like listed below

Defines a board for the driver. The driver supports up to 16 boards at the same time. For all ISA boards the type of installed board must be defined before using the driver the first time. All other functions just use the board number to access the board. After initialising the board all parameters will be set to default values.

Using Windows NT the board is already installed in the registry. This function will then just compare the board type with the already installed one.

Linux initialises the boards while loading the kernel module. This function is not available under Linux.

int16 SpcSetParam (int16 nr, int32 reg, int32 value)

<i>nr</i>	16 bit integer	number of the board as defined by SpcInit...
<i>reg</i>	32 bit integer	register to be changed
<i>value</i>	32 bit integer	value for the register
<i>return</i>	16 bit integer	error code of function like listed below

Sets a register to a defined value or executes a command. The board must be initialised before. When using ISA boards, all installation parameters must be set before (address, installed memory, ...). The allowed registers for the driver are listed in the board specific part of the documentation.

When using Windows NT the installation parameters may not be changed, they are set in the registry using the driver configuration utility.

int16 SpcGetParam (int16 nr, int32 reg, int32* value)

<i>nr</i>	16 bit integer	number of the board as defined by SpcInit...
<i>reg</i>	32 bit integer	register to be read
<i>value</i>	adr of 32 bit integer	value from the register
<i>return</i>	16 bit integer	error code of function like listed below

Reads a register or a status information of the board. The board must be initialised before. When using ISA boards, the installation address must be set before. The allowed registers for the driver are listed in the board specific part of the documentation.

int16 SpcSetData (int16 nr, int16 ch, int32 start, int32 len, dataptr data)

<i>nr</i>	16 bit integer	number of the board as defined by SpcInit...
<i>ch</i>	16 bit integer	channel to be written to
<i>start</i>	32 bit integer	startvalue to be written
<i>len</i>	32 bit integer	number of values to be written
<i>data</i>	huge ptr to data	data to be written
<i>return</i>	16 bit integer	error code of function like listed below

Writes data to the board for a specific channel. The board must be initialised before. When using ISA boards, all installation parameters must be set before (address, installed memory, ...). The Start and Len parameter are implemented on all PCI boards. On ISA boards the whole data will be written in one turn. The data must be in two's complement format (standard integer format).

int16 SpcGetData (int16 nr, int16 ch, int32 start, int32 len, dataptr data)

<i>nr</i>	16 bit integer	number of the board as defined by SpcInit...
<i>ch</i>	16 bit integer	channel to be read
<i>start</i>	32 bit integer	startvalue to be read
<i>len</i>	32 bit integer	number of values to be read
<i>data</i>	huge ptr to data	data space for read values
<i>return</i>	16 bit integer	error code of function like listed below

Reads data from the board from a specific channel. The board must be initialised before. When using ISA boards, all installation parameters must be set before (address, installed memory, ...). The Start and Len parameter are implemented on all PCI boards. On ISA boards the whole data will be read in one turn. The read out data is in the two's complement format and could be directly used for data processing as standard integer values.

Error Codes

error name	value (hex)	value (dec.)	description
ERR_OK	0	0	Execution OK, no error.
ERR_INIT	1	1	The board number is not in the range of 0 to 15. When initialisation is executed: the board number is yet initialised, the old definition will be used.
ERR_NR	2	2	The board is not initialised yet. Use the function <i>SpcInitBoard</i> or <i>SpcInitPCIBoards</i> first.-
ERR_TYP	3	3	Initialisation only: The type of board is unknown.
ERR_FNCNOTSUPPORTED	4	4	This function is not supported by the hardware version.
ERR_LASTERR	10	16	Old Error waiting to be read.
ERR_ABORT	20	32	Abort of wait function
ERR_BOARDLOCKED	30	48	Access to the driver already locked by another program. Stop the other program before starting this one.
ERR_REG	100	256	The register is not valid for this type of board.
ERR_VALUE	101	257	The value for this register is not in a valid range, the allowed values and ranges are listed in the board specific documentation.
ERR_FEATURE	102	258	Feature is not installed on this board
ERR_SEQUENCE	103	259	Channel sequence is not allowed.
ERR_READABORT	104	260	Data read is not allowed after aborting the data acquisition.
ERR_NOACCESS	105	261	Access to this register denied. No access for user allowed.
ERR_POWERDOWN	106	262	Not allowed if powerdown mode is activated.
ERR_CHANNEL	110	272	The channel number may not be accessed on the board: Either it is not a valid channel number or the channel is not accessible due to the actual setup (e.g. Only channel 0 is accessible in interlace mode)
ERR_RUNNING	120	288	The board is still running, this function is not available now or this register is not accessible now.
ERR_ADJUST	130	304	Automatic adjustment has reported an error. Please check the boards inputs.
ERR_NOPCI	200	512	No PCI BIOS is found on the system.
ERR_PCIVERSION	201	513	The PCI bus has the wrong version. SPECTRUM PCI boards require PCI revision 2.1 or higher.
ERR_PCINOBORDS	202	514	No SPECTRUM PCI boards found.
ERR_PCICHECKSUM	203	515	The checksum of the board information has failed.
ERR_DMALOCKED	204	516	DMA buffer not available now.
ERR_MEMALLOC	205	517	Internal memory allocation failed.
ERR_FIFOBUFOVERRUN	300	768	Driver buffer overrun in FIFO mode.
ERR_FIFOHWOVERRUN	301	769	Hardware buffer overrun in FIFO mode.
ERR_FIFOFINISHED	302	770	FIFO transfer has been finished, programmed number of buffers has been transferred.
ERR_FIFOSSETUP	309	777	FIFO setup not possible, transfer rate to high (max 250 MB/s)
ERR_TIMESTAMP_SYNC	310	784	Synchronisation to external reference clock failed.

Valid Board Types

board	type(hex)	type (dec)
PAD52	600	1536
PAD82	200	512
PAD82a	210	528
PAD82b	220	544
PAD242	700	1792
PAD1232-10	400	1024
PAD1232-30	410	1040
PAD1232-40	420	1056

board	type(hex)	type (dec)
PAD1616a	500	1280
PAD1616b	510	1296
PAD164/2	900	2304
PAD164/5	910	2320
PADCO-06	1400	5120
PCK400	800	2048
DAP116	100	256
TRS582	1500	5376

board	type(hex)	type (dec)
PCI.212	300	384
PCI.208	1000	4096
PCI.412	1100	4352
PCI.DIO32	1200	4608
PCI.248	1300	4864
PCI.258	1600	5632
MI.3010	3010	12304
...

Hints for programming the boards

Programming an ISA board is done in the following steps:

- * initialise and define boards with function *SpcInitBoard* (Windows NT: utility DRVCONFIG.EXE)
- * set installation parameters like address, installed memory, version with function *SpcSetParam*
- * set user specific parameters and start board (loop)

Programming an PCI board is done by the following steps:

- * initialise PCI boards automatically with function *SpclnitPCIBoards*
- * read out installation parameters for all found PCI boards like version, installed memory
- * set user specific parameters and start board (loop)

If you are using ISA and PCI boards in one system at the same time, use the function *SpclnitPCIBoards* first and initialise the ISA boards after this. The function *SpclnitPCIBoards* uses the first board numbers and will overwrite other definitions.

It is only necessary to define the boards once for the driver with the functions *SpclnitPCIBoards* and *SpclnitBoard*. If you are defining the boards again, you will get an error code from the function and the old definition is still used. You may ignore this error.

Software - Register

These software register are to be used for the functions *SpcSetParam* and *SpcGetParam* of the software driver. All constants are found in the header file REGS.H.

PCI register

These Registers are set by the driver after PCI initialisation. The information is found in the on-board ROM. The program PCITEST.EXE on the driver disk will give the same information's.

register name	reg no.	r/w	
SPC_PCITYP	2000	r	type of board as listed above
SPC_PCIVERSION	2010	r	board revision: high part in bit 8..15, Low part in bit 0..7
SPC_PCIDATE	2020	r	production date: month in bit 0..7, year in bit 16..31
SPC_PCISERIALNO	2030	r	serial number of the PCI.248/PCI.258
SPC_PCISAMPLERATE	2100	r	max. samplerate as 32 bit integer value
SPC_PCIMEMSIZE	2110	r	installed memory in bytes as 32 bit integer value
SPC_PCIFEATURES	2120	r	installed features as a bitfield. See description beneath.
SPC_PCIINTERRUPT	2300	r	interrupt line of the board

PCI Features register

Bit 31 not used
 ...
 Bit 9 not used
 Bit 8 option double mem installed.
 Bit 7 not used
 Bit 6 not used
 Bit 5 option gated sampling installed.
 Bit 4 not used
 Bit 3 not used
 Bit 2 not used
 Bit 1 not used
 Bit 0 option multiple recording installed.

Error registers

If one action caused an error in the driver this error and the register and value where it occurs will be saved. The driver is then locked until the error is read out using the SPC_LASTERRORCODE function. All other functions will lead to the same errorcode unless the error is cleared by reading SPC_LASTERRORCODE.

name	value (dec)	r/w	
SPC_LASTERRORCODE	999999	r	errorcode of the last error as defined in errors.h
SPC_LASTERRORREG	999998	r	software register which causes the error
SPC_LASTERRORVALUE	999997	r	value which causes the error
SPC_LASTERRORTEXT	999996	r	Copies a short explanation of the error to a string. The argument value must be a pointer to a string with at least ERRORTXTLEN characters.

Status register

Status information can be read at any time. The other parameters can only be written and the data can be read if the board is stopped.

register name	reg no.	r/w	
SPC_STATUS	10	r	status register, values listed below.

status code	value	
SPC_RUN	0	board is running.
SPC_TRIGGER	10	trigger has been found.
SPC_READY	20	recording has stopped.

Command register

The command register executes commands like start and stop or synchronises the board with other boards.

register name	reg no.	r/w	
SPC_COMMAND	0	r/w	command register, allowed values listed below.

status code	value	
SPC_START	10	starts the board with the current register settings. If the settings of the input range has been set since the last start of the board, the driver will hold a 200 ms delay before starting to wait the relais settling time.
SPC_STOP	20	stops the board, data in memory is undefined.

Memory register

This register holds the number of samples, not the number of bytes. The possible values have to be doubled in the interlace mode (200 MHz).

register name	reg no.	r/w	
SPC_MEMSIZE	10000	r/w	memory size for recording: 32 samples up to <i>installed mem</i> /2 samples with steps of 32 samples. When using 400/500 MS/s mode: 64 samples up to <i>installed mem</i> with steps of 64.

Posttrigger register

Sets the number of samples to be recorded AFTER the triggerevent has been found. The corresponding pretrigger is calculated by the formula: pretrigger = memsize - posttrigger

If the posttrigger value is higher than the programmed memsize, the triggerevent is not visible.

If the option Multiple Recording is used, this register holds the segmentsize.

register name	reg no.	r/w	
SPC_POSTTRIGGER	10100	r/w	posttrigger value in the range 16 samples up to 256 MSamples with steps of 16. When using 400/500 MS/s mode :32 samples up to 512 MSamples with steps of 32.

Features register

All of this features Registers may be set by writing a 1 or cleared by writing a 0. Some features may only be used if this features is installed on the board (see PCI Features above).

register name	reg no.	r/w	
SPC_EXTERNALCLOCK	20100	r/w	the external clock will be used for recording. The external clock is allowed in the range of 25 MHz to 200 MHz / 250 MHz.
SPC_50OHM0	30030	r/w	set the channel 0 to 50 Ω input resistance (default is 1 MΩ)
SPC_50OHM1	30130	r/w	set the channel 1 to 50 Ω input resistance (default is 1 MΩ)
SPC_TRIGGER50OHM	40110	r/w	set the trigger input to 50 Ω input resistance (default is 1 MΩ)
SPC_CLOCK50OHM	20120	r/w	set the clock input to 50 Ω input resistance (default is 1MΩ)

Samplerate register

Sets the samplerate for recording.

register name	reg no.	r/w	
SPC_SAMPLERATE	20000	r/w	one of the possible internal samplerates listed below.

PCI.248	2 MS/s	5 MS/s	10 MS/s	20 MS/s	25 MS/s	50 MS/s	100 MS/s	200 MS/s	400 MS/s
PCI.258	2.5 MS/s	6.25 MS/s	12.5 MS/s	25 MS/s	31.25 MS/s	62.5 MS/s	125 MS/s	250 MS/s	500 MS/s

Sets the output frequency on Connector 5 fix to 10 MHz on the PCI.248 / 12.5 MHz on the PCI.258.

register name	reg no.	r/w	
SPC_SPECIALCLOCK	295100	r/w	enables the special clock output mode. 0 = disable, 1 = enable.

Triggermode register

register name	reg no.	r/w	
SPC_TRIGGERMODE	40000	r/w	triggermode for recording.

triggermodes	value	
TM_SOFTWARE	0	recording will start immediately.
TM_CH0POS	10000	wait for rising edge on channel 0 at triggerlevel
TM_CH0NEG	10010	wait for falling edge on channel 0 at triggerlevel
TM_CH1POS	10100	wait for rising edge on channel 1 at triggerlevel
TM_CH1NEG	10110	wait for falling edge on channel 1 at triggerlevel
TM_TTLPOS	20000	wait for external TTL trigger rising edge.
TM_TTLNEG	20010	wait for external TTL trigger falling edge.
TM_GATELOW	30000	gated sampling (option). Gate is TTL low level.
TM_GATEHIGH	30010	gated sampling (option). Gate is TTL high level.

Triggerlevel register

The triggerlevel is an 4 bit value which will be compared with the upper 4 bit of the ADC data.

register name	reg no.	r/w	
SPC_TRIGGERLEVEL	42000	r/w	triggerlevel for the triggermodes TM_CH0/1POS/NEG. Triggerlevel is in the range -8 to +7 compared to the upper 4 bits of the data.

triggerlevel	±100 mV	±200 mV	±500 mV	±1 V	±2 V	±5 V
7	+87.5 mV	+175.0 mV	+437.5 mV	+0.875 V	+1.750 V	+4.375 V
6	+75.0 mV	+150.0 mV	+375.0 mV	+0.750 V	+1.500 V	+3.750 V
...						
1	+12.5 mV	+25.0 mV	+62.5 mV	+0.125 V	+0.250 V	+0.625 V
0	0.0 mV	0.0 mV	0.0 mV	0 V	0 V	0 V
-1	-12.5 mV	-25.0 mV	-62.5 mV	-0.125 V	-0.250 V	-0.625 V
...						
-7	-87.5 mV	-175.0 mV	-437.5 mV	-0.875 V	-1.750 V	-4.375 V
-8	-100.0 mV	-200.0 mV	-500.0 mV	-1.000 V	-2.000 V	-5.000 V
stepsize	12.5 mV	25.0 mV	62.5 mV	0.125 V	0.250 V	0.6250 V

Gated Sampling (Option)

See the hardware description part of the manual for basic information about gated sampling.

register name	reg no.	r/w	
SPC_TRIGGERMODE	40000	r/w	trigger mode set to TM_GATELOW or TM_GHATEHIGH

The gated sampling mode is enabled by settings the triggermode to TM_GATELOW or TM_GATEHIGH. The register memsize holds the total amount of memory to be recorded. The register posttrigger has no function at gated sampling.

The sampling of data starts with the first edge of the external gate signal. At trigger mode TM_GATELOW this is the falling edge. At trigger mode TM_GATEHIGH this is the rising edge. Data is not recorded before the first occurrence of the correct edge even if the programmed gate level is present at the input connector at start time.

The delay between the external trigger event and the first sampled data corresponds to the used samplerate and the synchronisation and is fix for each recording with this settings. The delay is necessary for this board because it works with dynamic RAM and needs refresh cycles to let the data stay in memory when the board is not recording.

Samplerate PCI.248	Samplerate PCI.258	Double Mem	Synchronisation	Triggermode	Delay trigger to first sample
≤100 MS/s	≤125 MS/s	No	No	Gate LOW or Gate HIGH	22 Samples
≤100 MS/s	≤125 MS/s	Yes	No	Gate LOW or Gate HIGH	44 Samples
200 MS/s	250 MS/s	No	No	Gate LOW or Gate HIGH	19 Samples
200 MS/s	250 MS/s	Yes	No	Gate LOW or Gate HIGH	38 Samples
400 MS/s	500 MS/s	No	No	Gate LOW or Gate HIGH	38 Samples

Recording will pause at the end of a gate intervall (rising edge on trigger mode TM_GATELOW or falling edge on trigger mode TM_GATEHIGH). Due to the structure of the on board memory, recording may only stop at a 16 samples (32 samples for 400/500 MS/s mode and Double Mem mode) alignment. So there will be 1 to 16 (1 to 32 on 400/500 MS/s mode) additional samples recorded after the end of the gate intervall.

Example (trigger mode TM_GATEHIGH, no Synchronisation, samplerate ≤ 200 MS/s):

sample	ADC data	gate signal	data	comments
...	recording is running
-	-	1	-	
0	y0	1	y0	
1	y1	1	y1	
2	y2	1	y2	
...	
60	y60	1	y60	
61	y61	0	y61	end of gate interval
62	y62	0	y62	
63	y63	0	y63	end of recording at 16 byte boundary.
-	-	0	-	
...	
-	-	0	-	
-	-	1	-	start of next gate interval
...	
64	y64	1	y64	first recorded sample after 19 samples delay
65	y65	1	y65	
...	

Gated sampling may not be used together with the option Multiple Recording.

Double Mem (Option)

The option Double Mem allows it to use the complete on-board memory for the recording of channel 0. Normally each channel uses half of the on-board memory and only the 400/500 MS/s mode allows it to use the complete memory for one channel. With the option Double Mem it is possible to use the full memory for one channel with all samplerrates.

register name	reg no.	r/w	
SPC_DOUBLEMEM	220100	r/w	enables DoubleMem mode. 0 = disable, 1 = enable.

Using the Double Mem mode together with the 400/500 MS/s samplerrate is not allowed. When using this option the minimum and maximum values for the memsize and posttrigger register are doubled. Only channel 0 is available when the Double Mem option is used.

Multiple Recording (Option)

See the hardware description part of the manual for basic information about multiple recording.

register name	reg no.	r/w	
SPC_MULTI	220000	r/w	enables Multiple Recording for the board

The register memsize holds the total amount of memory to be recorded. The register posttrigger will hold the size of one segment. Recording is started with a fixed delay after the triggerevent is found. There is no pretrigger possible in Multiple Recording mode.

The delay between the external triggerevent and the first sampled data corresponds to the used samplerrate, the synchronisation and the uses triggermode and is fix for each recording with this settings. The delay is necessary for this board because it works with dynamic RAM and needs refresh cycles to let the data stay in memory when the board is not recording.

Samplerrate PCI.248	Samplerrate PCI.258	Double Mem	Synchronisation	Triggermode	Delay trigger to first sample
≤100 MS/s	≤125 MS/s	No	No	TTL Pos. or TTL Neg.	22 Samples
≤100 MS/s	≤125 MS/s	Yes	No	TTL Pos. or TTL Neg.	44 Samples
≤100 MS/s	≤125 MS/s	No	No	Ch0 or Ch0 Pos. or Neg.	28 Samples
≤100 MS/s	≤125 MS/s	Yes	No	Ch0 or Ch0 Pos. or Neg.	56 Samples
200 MS/s	250 MS/s	No	No	TTL Pos. or TTL Neg.	19 Samples
200 MS/s	250 MS/s	Yes	No	TTL Pos. or TTL Neg.	38 Samples
200 MS/s	250 MS/s	No	No	Ch0 or Ch1 Pos. or Neg.	30 Samples
200 MS/s	250 MS/s	Yes	No	Ch0 or Ch1 Pos. or Neg.	60 Samples
400 MS/s	500 MS/s	No	No	TTL Pos. or TTL Neg.	38 Samples
400 MS/s	500 MS/s	No	No	Ch0 or Ch1 Pos. or Neg.	60 Samples

Triggeroffset register (From version 2.3 on)

Due to the internal structure of the memory sub-system. sampling could only be stopped every 2nd. sample (in 400/500 MS/s mode every 4th. sample). For this there may be an 2 samples (4 samples) jitter in the trigger. If it is necessary to know the exact trigger sample, this offset could be read out and data could be synchronized.

The trigger offset could only be read out when using the internal channel trigger (TM_CH0POS, TM_CH0NEG, TM_CH1POS, TM_CH1NEG)

register name	reg no.	r/w	
SPC_TRIGGEROFFSET	45000	r	Triggeroffset in samples. Positive values mean that the real triggerevent is xx samples after the found triggerevent. Negative values mean that the real triggerevent is xx samples before the found triggerevent.

PulseWidth register (From version 2.3 on)

After the programmed trigger event has been detected the pulsewidth counter starts to count down. If the counter reaches zero the trigger event is performed and the posttrigger counter starts. While the pulsewidth counter is working the trigger input is still observed. If an opposite edge occurs (indicating a smaller pulse than programmed) the pulsewidth counter is stopped and nothing happens.

Due to the internal trigger structure there is a two/four samples jitter to the pulsewidth detection.

The pulsewidth counter could be used for all modes including Multiple Recording, Gated sampling and DoubleMem. The puslewidth samples must be added to the posttrigger value for normal mode and DoubleMem mode.

register name	reg no.	r/w	
SPC_PULSEWIDTH	44000	r/w	Minimum pulsewidth in samples for detecting a trigger event. Values from 0 to 127 (511 from version 2.4 on) are allowed. For two channel modes the value is internally multiplied by 2, for one channel mode (400/500 MS/s or DoubleMem) the value is internally multiplied by 4.

Offset and Gain settings (from hardware version 2.2 on)

The offset and gain of the amplifiers is set using an internal digital-to-analog converter. Therefore it is possible to compensate the offset error of the board easily with the help of the DAC's. The settings are saved in the hardware in an on-board EEPROM and are automatically used by the driver.

Offset and gain are adjusted in factory as a default settings. The gain adjustment could not be changed by the user. The default settings could not be overwritten by the user. There are 4 sets of user settings that could be changed for any environment.

register name	reg no.	r/w	
SPC_ADJ_LOAD	50000	w	Loads the DAC setting from the specified file in the EEPROM. Available settings files are listed below. The default settings are automatically loaded when the driver is started.
		r	Read out the actual used settings.
SPC_ADJ_SAVE	50010	w	Saves the current DAC settings to the specified file in the EEPROM. Available settings files are listed below. The default settings could not be overwritten.
		r	Read out the actual used settings.
SPC_ADJ_AUTOADJ	50020	w	Perform an automatic offset compensation in the driver. No signal is allowed to be connected to the inputs. The automatic offset compensation needs about 1 second to finish. All internal settings of the driver could be changed while offset compensation is in progress. The value defines the type of offset compensation. See the table below for a listing of the possible values.
SPC_ADJ_OFFSET0	51000	r/w	Offset DAC 0. See table below for an explanation of the DAC index. Valid values are in the range from 0x0000 to 0x0FFF.
...
SPC_ADJ_OFFSET23	51023	r/w	Offset DAC 23. See table below for an explanation of the DAC index. Valid values are in the range from 0x0000 to 0x0FFF.
SPC_ADJ_GAIN0	52000	r/w	Gain DAC 0. See table below for an explanation of the DAC index. Valid values are in the range from 0x0000 to 0x0FFF.
...
SPC_ADJ_GAIN23	52023	r/w	Gain DAC 23. See table below for an explanation of the DAC index. Valid values are in the range from 0x0000 to 0x0FFF.

EEPROM DAC settings	Value	
ADJ_DEFAULT	0	Default settings. Could not be written by the user
ADJ_USER0	1	User settings 0
ADJ_USER1	2	User settings 1
ADJ_USER2	3	User settings 2
ADJ_USER3	4	User settings 3

Automatic Offset Adjustment	Value	
ADJ_ALL	0	Offset adjustment for all ranges and settings.
ADJ_CURRENT	1	Offset adjustment only for the current setting.

DAC number	±100 mV	±200 mV	±500 mV	±1 V	±2 V	±5V
Channel 0 200/250 MSPS	0	1	2	3	4	5
Channel 1 200/250 MSPS	6	7	8	9	10	11
Channel 0 400/500 MSPS	12	13	14	15	16	17
Channel 1 400/500 MSPS	18	19	20	21	22	23

Input range register

register name	reg no.	r/w	
SPC_AMP0	30010	w	input range channel 0
SPC_AMP1	30110	w	input range channel 1

The input range can be set with in six steps with the help of relays. The input ranges are defined in factory and can be read out. Standard input ranges are:

value	calibrated input range
100	± 100 mV
200	± 200 mV
500	± 500 mV
1000	± 1 V
2000	± 2 V
5000	± 5 V

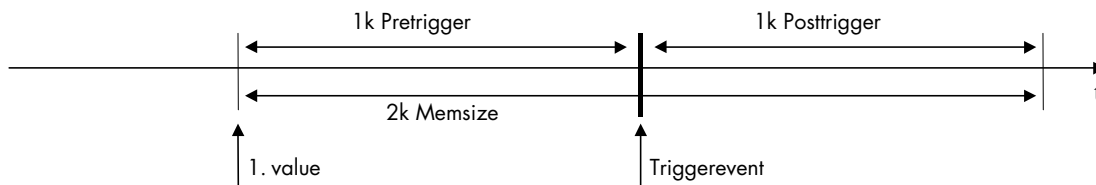
The available input ranges could be read out from the driver.

register name	reg no.	r/w	
SPC_READIRCOUNT	3000	r	number of calibrated input ranges.
SPC_READRANGECH0_0	3200	r	range in mV of calibrated input range 0. The values of the other entries may be read out from the following registers. (3201 for range1, 3202 for range2, ...)
SPC_READRANGECH1_0	3300	r	range in mV of calibrated input range 0. The values of the other entries may be read out from the following registers. (3301 for range1, 3302 for range2, ...)

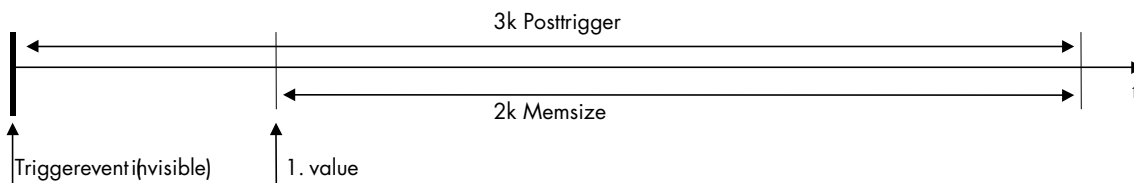
Data (Read)

Data can be read after the board has stopped. The data can be random accessed in blocks of variable length. The trigger event is found at the position memsize - posttrigger.

Example: 2k memsize, 1k posttrigger



Example: 2k memsize, 3k posttrigger



Example of Driver use

This example is written for a C or C++ compiler. It reads out the information of the PCI.248, sets the parameters and starts the board once. This file is also found in the example directory on the driver disk. The programming of the CPCI.248 is similar to this.

```
// -----
// PCI.248 software driver example      (c) Spectrum GmbH 04/1998
// -----
// This example will only work correct, if just one PCI.248 is installed
// in the system and no other Spectrum boards
// -----

#include <stdio.h>

#include "DLLTYP.H"
#include "SPECTRUM.H"
#include "ERRORS.H"
#include "REGS.H"

main()
{
    int16 Count, PCIVersion;
    int32 Typ, Version, Date, Status;
    int8  Data0[10241], Data1[10241];

    // ----- Initialisation of PCI Bus -----
    if (SpcInitPCIBoards (&Count, &PCIVersion) != ERR_OK) return 0;
    if (Count == 0) return 0;

    // ----- Test for PCI.248: 1.st board is board number 0 -----
    SpcGetParam (0, SPC_PCITYP, &Typ);
    if (Typ != TYP_PCI248)
        return 0;

    // ----- Get some PCI Parameters from driver and print them -----
    SpcGetParam (0, SPC_PCIVERSION, &Version);
    SpcGetParam (0, SPC_PCIDATE, &Date);
    printf ("PCI.248   V %x.%x   produced %02d.%04d\n", (int16) ((Version>>8)&0x000000FF),
           (int16) (Version&0x000000FF), (int16) (Date&0x000000FF), (int16) (Date>>16));

    // ----- Set Parameters for Recording -----
    SpcSetParam (0, SPC_SAMPLERATE, 2000000001); // samplerate 200 MHz
    SpcSetParam (0, SPC_MEMSIZE, 10241); // memsize 1 kSample
    SpcSetParam (0, SPC_POSTTRIGGER, 5121); // posttrigger 512 Sample
    SpcSetParam (0, SPC_EXTERNALCLOCK, 0); // no external clock
    SpcSetParam (0, SPC_AMP0, 10001); // channel 0 ±1 V input range
    SpcSetParam (0, SPC_AMP1, 10001); // channel 1 ±1 V input range
    SpcSetParam (0, SPC_50OHM0, 1); // channel 0 50 Ω input resistance
    SpcSetParam (0, SPC_50OHM1, 1); // channel 1 50 Ω input resistance
    SpcSetParam (0, SPC_TRIGGERMODE, TM_SOFTWARE); // software trigger
    SpcSetParam (0, SPC_MULT1, 01); // no multiple recording

    // ----- start the board -----
    SpcSetParam (0, SPC_COMMAND, SPC_START);

    // ----- wait for status ready -----
    do
    {
        SpcGetParam (0, SPC_STATUS, &Status);
    }
    while (Status != SPC_READY);

    // ----- read data -----
    SpcGetData (0, 0, 0, 10241, (dataptr) data0);
    SpcGetData (0, 1, 0, 10241, (dataptr) data1);

    return 0;
}
```

Transfer speed

All values measured using the actual version of the Spectrum driver. The test system was a Pentium II 450 MHz with 128 MB RAM. Board configuration is: Memsize as listed in the table, posttrigger=memsize/2, all channels recorded, software trigger

Memsize	Bytes	DOS			Win 3.11		
		Store	Add	Transfer	Store	Add	Transfer
256	512	13.2 kHz	2.8 kHz	12.8 MB/s	7.8 kHz	5.6 kHz	5.5 MB/s
512	1k	11.3 kHz	1.5 kHz	19.5 MB/s	7.0 kHz	3.8 kHz	9.8 MB/s
1k	2k	8.2 kHz	800 Hz	25.3 MB/s	6.0 kHz	2.3 kHz	16.1 MB/s
2k	4k	5.5 kHz	400 Hz	36.3 MB/s	4.0 kHz	1.3 kHz	24.8 MB/s
4k	8k	3.1 kHz	200 Hz	40.8 MB/s	2.4 kHz	700 Hz	27.6 MB/s
8k	16k	1.8 kHz	100 Hz	45.5 MB/s	1.3 kHz	360 Hz	30.0 MB/s
16k	32k	900 Hz	50 Hz	45.5 MB/s	700 Hz	180 Hz	33.9 MB/s
32k	64k	440 Hz	25 Hz	45.5 MB/s	370 Hz	90 Hz	36.0 MB/s
64k	128k	240 Hz	12.5 Hz	45.5 MB/s	180 Hz	45 Hz	36.0 MB/s
128k	256k			51.3 MB/s	90 Hz	22 Hz	33.8 MB/s
256k	512k				45 Hz	11 Hz	33.0 MB/s
512k	1M				23 Hz	5 Hz	33.0 MB/s
1M	2M				11 Hz	3 Hz	33.3 MB/s

Memsize	Bytes	Win 95/98			Win NT		
		Store	Add	Transfer	Store	Add	Transfer
256	512	6.8 kHz	6.8 kHz	5.0 MB/s	2.5 kHz	2.5 kHz	2.5 MB/s
512	1k	6.3 kHz	6.3 kHz	8.5 MB/s	2.5 kHz	2.5 kHz	4.8 MB/s
1k	2k	5.4 kHz	5.4 kHz	13.8 MB/s	2.3 kHz	2.3 kHz	8.7 MB/s
2k	4k	3.7 kHz	3.7 kHz	19.5 MB/s	2 kHz	2 kHz	14.0 MB/s
4k	8k	2.3 kHz	2.3 kHz	27.0 MB/s	1.5 kHz	1.5 kHz	20.5 MB/s
8k	16k	1.3 kHz	1.3 kHz	32.0 MB/s	1 kHz	1 kHz	26.5 MB/s
16k	32k	700 Hz	680 Hz	34.8 MB/s	600 Hz	600 Hz	30.7 MB/s
32k	64k	360 Hz	320 Hz	36.4 MB/s	330 Hz	300 Hz	33.8 MB/s
64k	128k	180 Hz	150 Hz	36.0 MB/s	170 Hz	140 Hz	34.5 MB/s
128k	256k	85 Hz	70 Hz	31.9 MB/s	85 Hz	65 Hz	31.5 MB/s
256k	512k	44 Hz	35 Hz	32.0 MB/s	43 Hz	35 Hz	31.4 MB/s
512k	1M	22 Hz	18 Hz	32.2 MB/s	21 Hz	17z	31.7 MB/s
1M	2M	11 Hz	9 Hz	31.5 MB/s	11 Hz	9 Hz	32.5 MB/s

Store: Set all Parameters to the board

Loop Start

Start the board
Wait for Ready
Read Data

Loop End

Add: Set all Parameters to the board

Loop Start

Start the board
Wait for Ready
Read Data
Add Data to existing Data

Loop End

Transfer: Set all Parameters to the board

Star the board
Wait for Ready

Loop Start

Read Data

Loop End