SPECTRUM
SYSTEMENTWICKLUNG MICROELECTRONIC GMBH

PCI.412

10/40 MHz 4 channel
12 bit transient recorder
for PCI bus

Hardware Manual
Driver Manual

SPECTRUM SYSTEMENTWICKLUNG MICROELECTRONIC GMBH • AHRENSFELDER WEG 13-17 • 22927 GROSSHANSDORF • GERMANY
TEL.: +49 (0)4102-6956-0 • FAX: +49 (0)4102-6956-66 • EMAIL: info@spec.de • INTERNET: http://www.spec.de
31.03.2004

Ahrensfelder Weg 13-17, 22927 Grosshansdorf, Germany

Table of Contents

Vorwort

Diese Anleitung enthält detailierte Informationen über die Hardware Möglichkeiten der PCI.412 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 Betrienbssysteme 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-aufrüstungen werden auf der Homepage http://www.spec.de bekannt gegeben. Hier kann ebenfalls die neueste Treiberverison mit den letzten Fehlerbereinigungen gefunden werden.

Preface

This manual provides detailed information on the hardware features of the PCI.412 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 actual 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

Der Transientenrekorder PCI.412 ist ein schneller, hochauflösender A/D Wandler für den PCI Bus. Seine 4 ADCs ermöglichen die simultane Erfassung mehrerer Signale ohne die Nachteile gemultiplexter Systeme. Außerdem stehen noch 16 Bit Digitaleingänge zur Verfügung, die mit dem gleichen Takt aufgezeichnet werden.
Jeder Kanal besitzt eine eigene Triggerlogik, die mit denen der anderen Kanäle verknüpft werden kann. Die verschiedenen Betriebsarten wie Speichersegmentierung, interner/externer Takt und Trigger, sowie Pre/Posttrigger ermöglichen dem Anwender eine einfache Anpassung an sein spezielles Meßproblem.
Mit einem speziellen Takt / Trigger Bus lassen sich mehrere Karten zu einem vielkanaligen System ausbauen.

Anwendungsbeispiele: Radar, Ultraschall, LDA/PDA, Qualitätssicherung, Spektroskopie

General Information

The PCI.412 transient recorder is a fast high resolution A/D-Converter for PCI based systems. Its 4 independent ADC's enable the simultaneously sampling of signals without the problems of multiplexed systems. Additionally 16 bits of digital inputs are available. These inputs are sampled with the same clock as the analogue inputs.
Each channel has its own triggerlogic, which can be linked together. The 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.
By using a special clock- and trigger bus the user may upgrade the PCI.412 to a measuring system with high channel count.

Application: Radar, Supersonics, LDA/PDA, Quality managment, Spectroscopie

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, DASY*Lab*, MATLAB und VEE erhältlich.

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, DASY*Lab*, MATLAB and VEE are available.

Additional information

To minimise noise keep the PCI.412 away from the power supply-

The PCI.412 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.412 in hermetic closed systems.

Order information

PCI.412 40 MHz	PCI.412 with 512 kSamples memory, 40 MHz, including drivers	PCI412-40
PCI.412 10 MHz	PCI.412 with 512 kSamples memory, 10 MHz, including drivers	PCI412-10
Option 2 M	Memory upgrading to 2 MSamples	PCI412-2
Multiple recording	Memory segmentation for fast repetition rates	PCI412-mr
Gate	Gated sampling with an external control signal	PCI412-gs
Cascading	Synchronisation of several PCI.412 for multi-channel-systems	PCI412-ks
Input range	3 user specific input ranges between ±500 mV and ±2.5 V bipolar or unipolar	PCI412-ir
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	PCI412-dl
HP-VEE driver	Drivers for HP-VEE 5.0 for Win 95/98, Win 2000 and Win NT	PCI412-hp
LabVIEW driver	Drivers for LabVIEW 4.0 for Win 3.11, Win 95/98, Win 2000 and Win NT	PCI412-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.

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 busses 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) 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 plaziert ist. Wenn Ihr System einen oder mehrere zusätzliche Lüfter besitzt, so plazieren 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.

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.

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.

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.

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)

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.

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)

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

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 installtion. 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 DRVCONFG.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.

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 DRVCONFG.EXE and check weather 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 Ihrere 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 gefunden 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 archieve 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 „spcioctl.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.

DRVCONFG.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 "spcioctl.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.

DRVCONFG.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.412 werden die Eingänge abgetastet und die konvertierten Daten im Speicher abgelegt (Der Speicher arbeitet als Ringbuffer, die Daten werden kontinuierlich in den Speicher geschrieben). Die Triggerereignisse werden 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 0/1/2/3) selektiert, so werden die sechs 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 Triggerereigniss 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.412 und der Status ändert sich auf ‚Ready'.

Trigger Information

After the PCI.412 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). No triggerevents 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 triggerevent is detected immediately. Using the TTL trigger will cause a triggerevent 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 0/1/2/3) the 6 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 triggerevent is detected. It is possible to combinate the triggerevents of up to four channels with AND or OR. Using the AND-combination, all triggerevents have to be recognized. Using the OR-combination, any triggerevent will trigger the board.
The status will be change to 'trigger found' and the postcounter starts counting down the posttrigger value. After the postcounter reaches zero the PCI.412 stops and signals 'ready' in the status register.

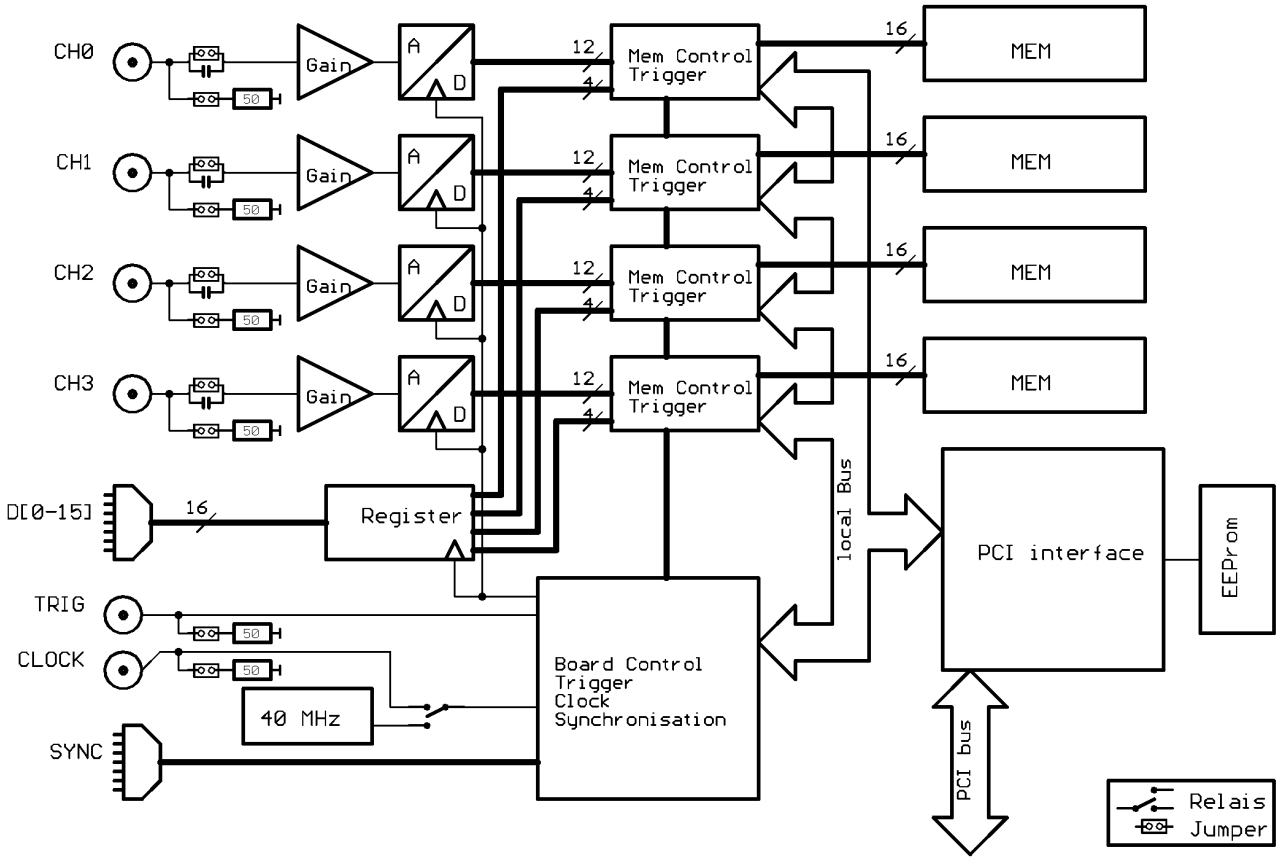
Option Synchronisierung

Diese Option erlaubt es, mehrere Karten dieses Typs miteinander intern zu synchronisieren, um auf einfache Art und Weise synchrone Mehrkanalsystem aufzubauen.
Eine Karte wird als Master konfiguriert und generiert den Takt und die Triggerinformation für die anderen (Slave) Karten. Alle Karten laufen synchron mit exakt dem gleichen Abtasttakt.

Option Synchronisation

This option allows it to connect several boards from Spectrum to generate a synchronously multi-channel system.
One board is dedicated as the master board and generates clock and trigger signals for the other (slave) boards. All boards are running synchronously.

Block diagram PCI.412



Technical data

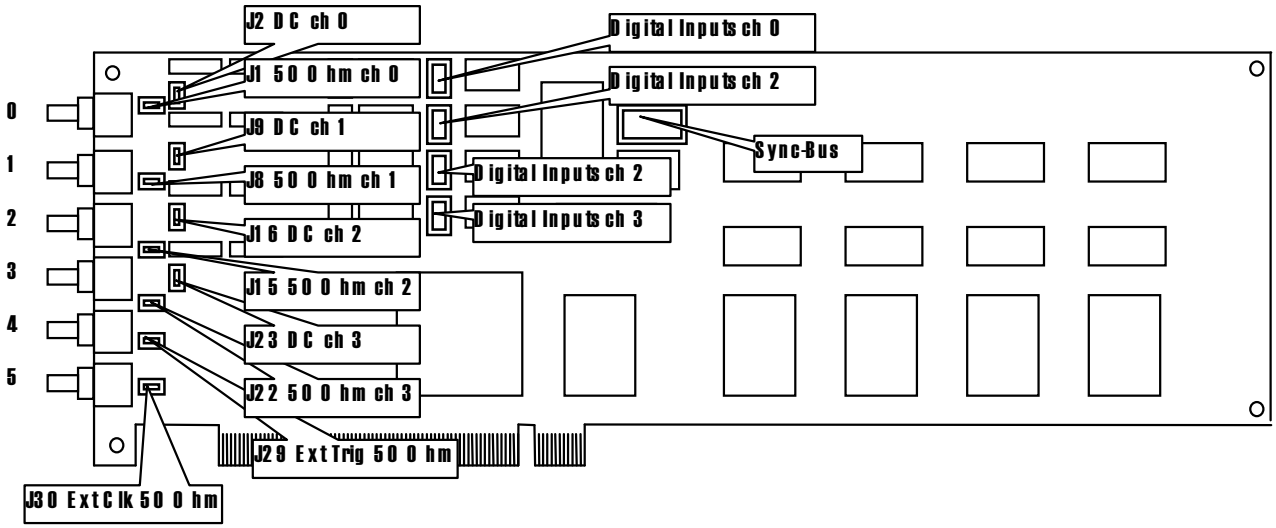
Resolution	12 bit
Samplerate	160 kHz up to 10 MHz / 40 MHz
Bandwidth DC –3 dB	0 Hz to 14 MHz / 21 MHz
Bandwidth AC –3 dB	20 Hz to 14 MHz / 21 MHz
Differential linearity error	±0.5 LSB typ. (ADC)
Integral linearity error	±2 LSB typ. (ADC)
SFDR fs = 1 MHz fck = 40 MHz	69 dBFS typ. (ADC)
ENOB fs = 1 MHz, fck=40 MHz	10.8 bit typ. (ADC)
ENOB fs=10 MHz, fck=40 MHz	10.8 bit typ. (ADC)
Aperture jitter	1.2 ps rms (ADC)
Multi: Trigger to 1st sample delay	0 to 4 samples (fix)
Multi: Recovery time	≤ 3 samples
Trigger output delay	6 to 14 samples (fix)
Trigger accuracy	1 sample
Digital input to analog input delay	1 to 8 samples (fix)
Ext. clock: output delay	5.5 ns
Ext. clock: delay to internal clock	10 ns
Sync: board to board trigger jitter	0 samples
Sync: board to board clock delay	≤ 1.5 ns

Input range	±500 mV	±1 V	±2 V
Offset error	≤ 5 LSB	≤ 4 LSB	≤ 3 LSB
Gain error	≤ 2 %	≤ 1.5 %	≤ 1.0 %
Noise	≤ 3 LSB	≤ 3 LSB	≤ 3 LSB
Crosstalk	-60 dB	-63 dB	-66 dB

Dimension	312 mm x 109 mm
Connector	3 mm BNC female
Input impedance	50 Ohm or 1 MOhm || 15 pF
Overvoltage protection	±20 V
Warm up time	10 minutes
Operating temperatur	0°C - 50°C
Storage temperatur	-10°C - 70°C
Humidity	10% to 90% non condensing

	+3.3 V	+5 V	+12 V	-12 V
Power comsumption (A)	0 mA	2650 mA	35 mA	0 mA
Power consumption (W)	0.0 W	13.3 W	0.4 W	0.0 W

Placement PCI.412



Connectors

The PCI.412 has six 3 mm BNC connectors.

Connector 0:	analogue channel 0.	
Connector 1:	analogue channel 1.	
Connector 2:	analogue channel 2	
Connector 3:	analogue channel 3	
Connector 4:	triggermode = TTLPOS or TTLNEG:	external trigger input
	triggermode ≠ TTLPOS and TTLNeg	trigger output
Connector 5:	EXTERNALCLOCK = 1	clock input
	EXTERNALCLOCK = 0	clock output

AC/DC jumper

channel 0		channel 1		channel 2		channel 3	
J2 set	ch0 DC	J9 set	ch1 DC	J16 set	ch2 DC	J23 set	ch3 DC
J2 clear	ch0 AC	J9 clear	ch1 AC	J16 clear	ch2 AC	J23 clear	ch3 AC

50 Ohm jumper

channel 0		channel 1		channel 2		channel 3	
J1 set	ch0 50 Ohm	J8 set	ch1 50 Ohm	J15 set	ch2 50 Ohm	J22 set	ch3 50 Ohm
J1 clear	ch0 1 MOhm	J8 clear	ch1 1 MOhm	J15 clear	ch2 1 MOhm	J22 clear	ch3 1 MOhm

Digital inputs PCI.412

J3	ch0 bit 0	J17	ch2 bit 0
J4	ch0 bit 1	J18	ch2 bit 1
J5	ch0 bit 2	J19	ch2 bit 2
J6	ch0 bit 3	J20	ch2 bit 3
J10	ch1 bit 0	J24	ch3 bit 0
J11	ch1 bit 1	J25	ch3 bit 1
J12	ch1 bit 2	J26	ch3 bit 2
J13	ch1 bit 3	J27	ch3 bit 3

Sync Bus

Carries the signals for synchronisation of multiple PCI.412

Pin 1	Sample Clock
Pin 3	Sync Trigger
Pin 5, 7, 9	not used
Pin 2, 4, 6, 8, 10	GND

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.

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 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.

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.

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.

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	DAP116	PCK400	TRS582	PADCO-06	MI.30xx	MI.31xx	MI.40xx	MI.45xx	MI.60xx	MI.70xx
SpcInitPCIBoards	-	-	-	+	+	+	+	+	+	+	-	-	-	-	-	-	-	+	+	+	+	+	+
SpcInitBoard	+	+	+	-	-	-	-	-	-	-	+	+	+	+	+	+	+	-	-	-	-	-	-
SpcSetParam	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
SpcGetParam	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+
SpcSetData	-	-	-	-	-	-	-	-	-	+	-	-	-	+	-	+	-	-	-	-	-	+	+
SpcGetData	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	+	+	+	+	-	+

int16 SpcInitPCIBoards (int16* count, int16* PCIVersion)

count	adr of 16 bit integer	number of found PCI boards
PCIVersion	adr of 16 bit integer	found PCI version
return	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)

nr	16 bit integer	number of the board to be defined in range 0-15
typ	16 bit integer	type of the defined board listed in REGS.H
return	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)

nr	16 bit integer	number of the board as defined by SpcInit...
reg	32 bit integer	register to be changed
value	32 bit integer	value for the register
return	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)

nr	16 bit integer	number of the board as defined by *SpcInit...*
reg	32 bit integer	register to be read
value	adr of 32 bit integer	value from the register
return	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)

nr	16 bit integer	number of the board as defined by SpcInit...
ch	16 bit integer	channel to be written to
start	32 bit integer	startvalue to be written
len	32 bit integer	number of values to be written
data	huge ptr to data	data to be written
return	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)

nr	16 bit integer	number of the board as defined by SpcInit...
ch	16 bit integer	channel to be read
start	32 bit integer	startvalue to be read
len	32 bit integer	number of values to be read
data	huge ptr to data	data space for read values
return	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 *SpcInitBoard* or *SpcInitPCIBoards* 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 adjustion 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_PCINOBOARDS	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_FIFOSETUP	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 DRVCONFG.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 *SpcInitPCIBoards*
* 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 *SpcInitPCIBoards* first and initialise the ISA boards after this. The function *SpcInitPCIBoards* 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 *SpcInitPCIBoards* and *SpcInitBoard*. 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.412
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.

PCI Features register

Bit 31 not used
...
Bit 8 not used
Bit 7 option synchronisation master: board is clock-master for synchronisation.
Bit 6 option synchronisation slave : board is clock-slave for synchronisation.
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 ERRORTEXTLEN 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 event 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	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.
SPC_SYNCMASTER	100	synchronisation with internal sync bus, this board works as clock master
SPC_SYNCSLAVE	110	synchronisation with internal sync bus, this board works as clock slave
SPC_NOSYNC	120	no synchronisation
SPC_SYNCTRIGGERMASTER	101	synchronisation: board generates trigger for all boards
SPC_SYNCTRIGGERSLAVE	111	synchronisation: board receives trigger from trigger master board.

Synchronisation (Option)

This option allows it to connect several boards from Spectrum to generate a multi-channel system. It is possible to connect several PCI.412. One board is the clock master and generates the clock for the other (slave) boards. The clock master is defined in hardware and has the synchronisation master bit set in the PCI features register (see above). Only one board may be the clock master. At runtime any of the synchronised boards may be defined as a trigger master and generates trigger information for the other boards.

The boards have to be installed and connected in a defined order to work correctly. Use the label on the board to determine which board must be installed in which slot. Connect the boards with the delivered flat ribbon cable and use the sync-connector as shown in the placement. Do not change the board order, the synchronisation will only work with this installation: **Master – Slave0 – Slave1 – Slave2 ...**

If the boards are synchronised, they must be programmed in the following steps:

(1) Set all parameters for all boards except the sync information
(2) Set the sync information for the clock-master board.
(3) Set the sync information for all clock-slave boards.
(2) Start all trigger-slave boards.
(3) Start the trigger-master board.

All boards will run with the clock generated by the clock-master board. Only the trigger-master board may generate a trigger. The trigger settings for the trigger-slave boards will be ignored.

Memory register

This register holds the number of samples, not the number of bytes.

register name	reg no.	r/w	
SPC_MEMSIZE	10000	w	memory size for recording: 32 samples up to *installed mem* / 4 samples with steps of 32 samples.

Posttrigger register

Sets the number of samples to be recorded AFTER the trigger event 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 trigger event is not visible.
If the option Multiple Recording is used, this register holds the segmentsize.

register name	reg no.	r/w	
SPC_POSTTRIGGER	10100	w	posttrigger value in the range 32 samples up to 512 kSamples with steps of 32.

Features register

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

register Name	reg no.	r/w	
SPC_EXTERNALCLOCK	20100	w	The external clock will be used for recording. Be sure that the external clock is not higher than 40 MHz (10 MHz for the 10 MHz Version) and that the sampling clock is not lower than 10 kHz.
SPC_EXTERNOUT	20110	w	The sampling clock will be put out on BNC-connector 6. The output will generate a low to high edge on every sample.
SPC_TRIGGEROUT	40100	w	The triggersignal will be put out on BNC-connector 5. The output will generate a high to low edge when the triggercondition occurs..
SPC_PATTERNENABLE	110000	w	Read out the digital channels and the gate-mark-bit (option). If this register is not set, data will be expanded in hardware to 16 bit integer. At the triggermode TM_GATELOW/HIGH is the first sample after the gate occurred marked on each channel with bit 15 set to 1. The other samples are readout with bit 15 set to low.

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
SPC_AMP2	30210	w	Input range channel 2
SPC_AMP3	30310	w	Input range channel 3

The input range can be set in three steps with the help of relais. The input ranges are defined in factory and can be read out from the driver. Standard input ranges are:

value	calibrated input range
500	±500 mV
1000	±1 V
2000	±2 V

register name	reg no.	r/w	
SPC_READIRCOUNT	3000	r	number of calibrated input ranges.
SPC_READUNIPOLAR0	3010	r	channel 0: +1 = unipolar > 0 V, -1 = unipolar < 0 V, 0 = bipolar
SPC_READUNIPOLAR1	3020	r	channel 1: +1 = unipolar > 0 V, -1 = unipolar < 0 V, 0 = bipolar
SPC_READUNIPOLAR2	3030	r	channel 2: +1 = unipolar > 0 V, -1 = unipolar < 0 V, 0 = bipolar
SPC_READUNIPOLAR3	3040	r	channel 3: +1 = unipolar > 0 V, -1 = unipolar < 0 V, 0 = bipolar
SPC_READRANGECH0_0	3200	r	range in mV of calibrated input range for channel 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 for channel 1. The values of the other entries may be read out from the following registers. (3301 for range1, 3302 for range2)
SPC_READRANGECH2_0	3400	r	range in mV of calibrated input range for channel 2. The values of the other entries may be read out from the following registers. (3401 for range1, 3402 for range2)
SPC_READRANGECH3_0	3500	r	range in mV of calibrated input range for channel 3. The values of the other entries may be read out from the following registers. (3501 for range1, 3502 for range2)

Triggermode register

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

triggermodes	value	
TM_SOFTWARE	0	Recording will start immediately.
TM_TTLPOS	20000	Wait for external TTL trigger rising edge.
TM_TTLNEG	20010	Wait for external TTL trigger falling edge.
TM_GATELOW	30000	Wait for external low-level (option).
TM_GATEHIGH	30010	Wait for external high-level (option).
TM_CHOR	35000	Wait for the first triggerevent off all enabled channels.
TM_CHAND	35010	Wait for all triggerevents off all enabled channels.

register name	reg no.	r/w	
SPC_TRIGGERMODE0	40200	w	Channelmode for the triggermode TM_CHOR or TM_CHAND.
SPC_TRIGGERMODE1	40201	w	Channelmode for the triggermode TM_CHOR or TM_CHAND.
SPC_TRIGGERMODE2	40202	w	Channelmode for the triggermode TM_CHOR or TM_CHAND.
SPC_TRIGGERMODE3	40203	w	Channelmode for the triggermode TM_CHOR or TM_CHAND.

channelmode	value	
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_CH0OFF	10020	Channel 0 is disabled for triggering.
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_CH1OFF	10120	Channel 1 is disabled for triggering.
TM_CH2POS	10200	Wait for rising edge on channel 2 at triggerlevel
TM_CH2NEG	10210	Wait for falling edge on channel 2 at triggerlevel
TM_CH2OFF	10220	Channel 2 is disabled for triggering.
TM_CH3POS	10300	Wait for rising edge on channel 3 at triggerlevel
TM_CH3NEG	10310	Wait for falling edge on channel 3 at triggerlevel
TM_CH3OFF	10320	Channel 3 is disabled for triggering.

A trigger for only one channel (e. g. channel 1) is easily set by using these commands:

(1) Write TM_CHOR to the SPC_TRIGGERMODE register.
(2a) Write TM_CH1POS or TM_CH1NEG to the SPC_TRIGGERMODE1 register.
(2b) Write TM_CH0OFF to the SPC_TRIGGERMODE0 register.
(2c) Write TM_CH2OFF to the SPC_TRIGGERMODE2 register.
(2d) Write TM_CH3OFF to the SPC_TRIGGERMODE3 register.
(3) Write the triggerlevel for channel 1 to the SPC_TRIGGERLEVEL1 register.

For a trigger-combination of the channels 0, 1 and 2 the following commands are needed:

(1) Write TM_CHOR or TM_CHAND to the SPC_TRIGGERMODE register.
(2a) Write TM_CH0POS or TM_CH0NEG to the SPC_TRIGGERMODE0 register.
(2b) Write TM_CH1POS or TM_CH1NEG to the SPC_TRIGGERMODE1 register.
(2c) Write TM_CH2POS or TM_CH2NEG to the SPC_TRIGGERMODE2 register.
(2d) Write TM_CH3OFF to the SPC_TRIGGERMDCH3 register.
(3a) Write the triggerlevel for channel 0 to the SPC_TRIGGERLEVEL0 register.
(3b) Write the triggerlevel for channel 1 to the SPC_TRIGGERLEVEL1 register.
(3c) Write the triggerlevel for channel 2 to the SPC_TRIGGERLEVEL2 register.

Triggerlevel register

register name	reg no.	r/w	
SPC_TRIGGERLEVEL0	42000	w	Triggerlevel for channel 0 in triggermodes TM_CHOR or TM_CHAND.
SPC_TRIGGERLEVEL1	42001	w	Triggerlevel for channel 1 in triggermodes TM_CHOR or TM_CHAND.
SPC_TRIGGERLEVEL2	42002	w	Triggerlevel for channel 2 in triggermodes TM_CHOR or TM_CHAND.
SPC_TRIGGERLEVEL3	42003	w	Triggerlevel for channel 3 in triggermodes TM_CHOR or TM_CHAND.

The triggerlevel is an 6 bit value which will be compared with the upper 6 bit of the ADC data. For this the voltage triggerlevel depends on the selected inputrange.

value (decimal)	triggerlevel	input range: ±500 mV	input range: ±1 V	input range: ±2 V
31	+ (range / 32) * 31	484 mV	969 mV	1, 94 V
30	+ (range / 32) * 30	469 mV	938 mV	1,88 V
...	...	...	...	...
2	+ (range / 32) * 2	31,3 mV	62,5 mV	125 mV
1	+ (range / 32) * 1	15,6 mV	31,3 mV	62,5 mV
0		0 mV	0 mV	0 mV
-1	- (range / 32) * 1	-15,6 mV	-31,3 mV	-62,5 mV
-2	- (range / 32) * 2	-31,3 mV	-62,5 mV	-125 mV
...	...	...	...	...
-30	- (range / 32) * 30	-469 mV	-938 mV	-1,88 V
-31	- (range / 32) * 31	-484 mV	-969 mV	-1,94 V

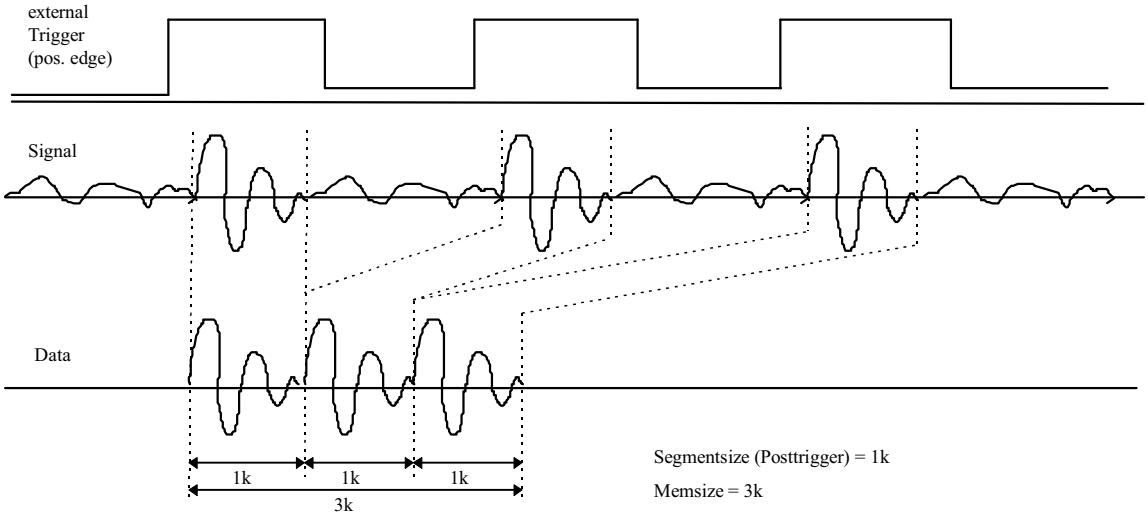
Allowed values are in the range from -31 to +31.

Multiple Recording (Option)

The option Multiple Recording allows the recording of several triggerevents 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 an edge on the external triggerinput occurs.

register name	reg no.	w	
SPC_MULTI	220000	w	enables Multiple Recording for the board. 0 = disable, 1 = enable.

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.



Samplerate register

Sets the samplerate for recording.

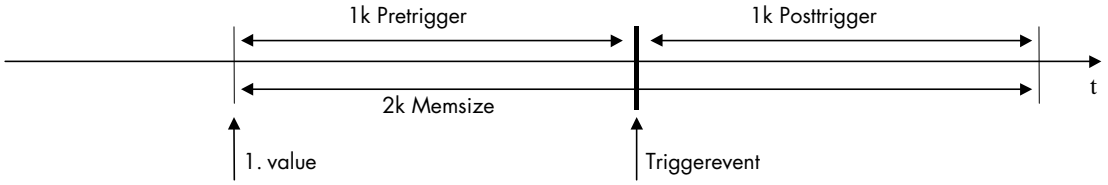
register name	reg no.	r/w	
SPC_SAMPLERATE	20000	w	Samplerate between 157 kHz and 40 MHz (10 MHz for the 10 MHz Version)

The value is a 32 bit integer in the range from 40 MHz down to 157 kHz using an 8 bit divider.

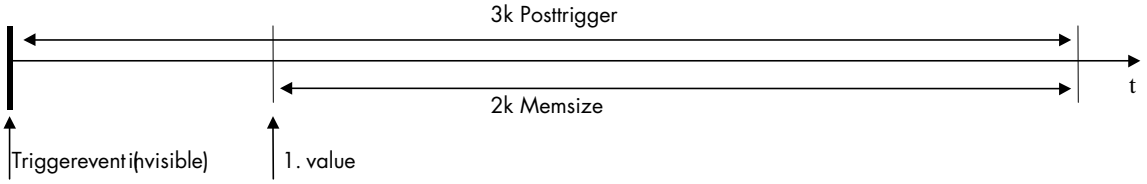
Version	÷ 1	÷ 2	÷ 3	÷ 4	÷ 5	...	÷ 254	÷ 255
40 MHz	40 MHz	20 MHz	13,33 MHz	10 MHz	8 MHz	...	157,5 kHz.	156,8 kHz
10 MHz	-	-	-	10 MHz	8 MHz	...	157,5 kHz.	156,8 kHz

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



Data format

Data will be read out in 2th complement. Normally the upper 4 bit (bit 12 ...bit 15) will be signed expanded in hardware. With this feature it is possible to use the read data directly as signed 16 bit integer values.
If the register SPC_PATTERNENABLE is set, the digital inputs will be read out in the upper four bits of the four channels. In this case it is necessary to divide the data in the analogue and the digital channels before processing the data.

			TM_GATELOW or TM_GATEHIGH
	SPC_PATTERNENABLE = 0	SPC_PATTERNENABLE = 1	SPC_PATTERNENABLE = 1
bit		channel x (x = 0, 1, 2, 3)	channel x (x = 0, 1, 2, 3)
0	AD 0 (LSB)	AD 0 (LSB)	AD 0 (LSB)
1	AD 1	AD 1	AD 1
2	AD 2	AD 2	AD 2
3	AD 3	AD 3	AD 3
4	AD 4	AD 4	AD 4
5	AD 5	AD 5	AD 5
6	AD 6	AD 6	AD 6
7	AD 7	AD 7	AD 7
8	AD 8	AD 8	AD 8
9	AD 9	AD 9	AD 9
10	AD 10	AD 10	AD 10
11	AD 11 (MSB)	AD 11 (MSB)	AD 11 (MSB)
12	AD 11	channel x Digital 0	channel x Digital 0
13	AD 11	channel x Digital 1	channel x Digital 1
14	AD 11	channel x Digital 2	channel x Digital 2
15	AD 11	channel x Digital 3	1 for the first sample after the gate is opened otherwise 0

Example of Driver use

This example is written for a C or C++ compiler. It reads out the information of the PCI.412, sets the parameters and starts the board once. This file is found in the example directory on the driver disk.

```
// PCI.412 DOS Software driver example              (c) Spectrum GmbH 10/1998
// ----------------------------------------------------------------------
// This example will only work correct, if just one PCI.412 is installed in
// the system and no other Spectrum boards
// ----------------------------------------------------------------------

#include <stdio.h>
#include <string.h>
#include "DLLTYP.H"
#include "SPECTRUM.H"
#include "ERRORS.H"
#include "REGS.H"

int16 Data0[1024l], Data1[1024l], Data2[1024l], Data3[1024l];

int main ()
{ int16 Count, PCIVersion;
  int32 Typ, Version, Date, Status, IRCount, i, Range0, Range1, Range2, Range3;
  int32 ErrorCode, ErrorReg, ErrorValue;

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

  // ----- Test for PCI.412: 1.st board is board number 0 -------------------
  SpcGetParam (0, SPC_PCITYP, &Typ);
  if (Typ != TYP_PCI412) return 0;

  // ----- Get some PCI Parameters from driver and print them ---------------
  SpcGetParam (0, SPC_PCIVERSION,    &Version);
  SpcGetParam (0, SPC_PCIDATE,       &Date);
  printf ("\n\nPCI.412   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,    100000000l);   // Samplerate 10 MHz
  SpcSetParam (0, SPC_MEMSIZE,       1024l);        // Memsize 1 kSample
  SpcSetParam (0, SPC_POSTTRIGGER,   512l);         // Posttrigger 512 Sample
  SpcSetParam (0, SPC_EXTERNALCLOCK, 0);            // no external clock
  SpcSetParam (0, SPC_TRIGGEROUT,    0);            // no trigger output
  SpcSetParam (0, SPC_EXTERNOUT,     0);            // no clock output
  SpcSetParam (0, SPC_PATTERNENABLE, 0);            // no pattern input
  SpcSetParam (0, SPC_AMP0,          1000l);        // ch. 0 +/- 1 V input range
  SpcSetParam (0, SPC_AMP1,          1000l);        // ch. 1 +/- 1 V input range
  SpcSetParam (0, SPC_AMP2,          1000l);        // ch. 2 +/- 1 V input range
  SpcSetParam (0, SPC_AMP3,          1000l);        // ch. 3 +/- 1 V input range
  SpcSetParam (0, SPC_TRIGGERMODE,   TM_SOFTWARE); // Software trigger
  SpcSetParam (0, SPC_COMMAND,       SPC_START );  // ----- Start the board

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

  SpcGetData (0, 0, 0, 1024l, (dataptr) &Data0[0]); // ----- Read data
  SpcGetData (0, 1, 0, 1024l, (dataptr) &Data1[0]);
  SpcGetData (0, 2, 0, 1024l, (dataptr) &Data2[0]);
  SpcGetData (0, 3, 0, 1024l, (dataptr) &Data3[0]);

  printf ("\nPCI.412 ready, Data transferred.\n");
  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

		DOS			Win 3.11		
Memsize	Bytes	Store	Add	Transfer	Store	Add	Transfer
256	2k	6300 Hz	1700 Hz	22,0 MB/s	3700 Hz	2800 Hz	11,0 MB/s
512	4k	4400 Hz	900 Hz	35,5 MB/s	2700 Hz	1750 Hz	17,0 MB/s
1k	8k	2800 Hz	450 Hz	50,0 MB/s	1700 Hz	1000 Hz	19,0 MB/s
2k	16k	1600 Hz	250 Hz	61,0 MB/s	1000 Hz	550 Hz	24,5 MB/s
4k	32k	850 Hz	120 Hz	67,0 MB/s	490 Hz	280 Hz	24,0 MB/s
8k	64k	440 Hz	60 Hz	69,0 MB/s	260 Hz	140 Hz	25,0 MB/s
16k	128k	220 Hz	30 Hz	73,0 MB/s	130 Hz	70 Hz	26,0 MB/s
32k	256k	110 Hz	15 Hz	77,0 MB/s	70 Hz	35 Hz	27,0 MB/s
64k	512k				35 Hz	17 Hz	27,0 MB/s

		Win 95			Win NT		
Memsize	Bytes	Store	Add	Transfer	Store	Add	Transfer
256	2k	3700 Hz	3700 Hz	5 MB/s	1900 Hz	1950 Hz	7,0 MB/s
512	4k	2800 Hz	2800 Hz	9 MB/s	1600 Hz	1550 Hz	11,5 MB/s
1k	8k	1450 Hz	1450 Hz	14 MB/s	1200 Hz	1100 Hz	16,0 MB/s
2k	16k	1000 Hz	950 Hz	19 MB/s	750 Hz	700 Hz	20,0 MB/s
4k	32k	510 Hz	480 Hz	23 MB/s	450 Hz	400 Hz	23,0 MB/s
8k	64k	260 Hz	240 Hz	25 MB/s	250 Hz	210 Hz	24,0 MB/s
16k	128k	130 Hz	120 Hz	27 MB/s	120 Hz	110 Hz	26,0 MB/s
32k	256k	67 Hz	60 Hz	27 MB/s	60 Hz	60 Hz	26,5 MB/s
64k	512k	35 Hz	30 Hz	27 MB/s	30 Hz	30 Hz	27,0 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
```