



SPECTRUM
SYSTEMENTWICKLUNG MICROELECTRONIC GMBH

PAD242
200/400 MHz 8 bit transient recorder
for ISA bus

Hardware Manual
Driver Manual

© Spectrum Systementwicklung Microelectronic GmbH - 2004
Ahrensfelder Weg 13-17, 22927 Grosshansdorf, 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

Table of Contents	3
Preface	5
Product Introduction	6
General Information	6
Software	6
Additional information	6
Order information	6
Installation	7
System Requirements	7
Hardware Installation	7
DOS	7
Installation for Windows 3.x.....	8
Installation for Windows 95/98.....	8
Installation for Win NT/2000.....	9
Installation for Linux.....	10
Login	10
Load driver.....	10
Major number.....	10
Installing the device	10
End	10
Info	10
Utilities	11
SBench 5.x.....	11
DRVCONFIG.EXE	11
PCITEST.EXE	11
Hardware Description	12
Trigger Information.....	12
Option Multiple Recording	12
Block diagram.....	13
Technical data.....	13
Placement	14
50 Ohm jumper	14
Addresses	14
Interrupt line	14
DSP Link - interface.....	15
LSI 50 - Connector (P8).....	15
Timing specification:	15
Hardware Register	16
Controlregister address: 0x00	16
Posttrigger address: 0x02, 0x4	16
Channelparameter address: 0x06.....	17
Sampling frequency address: 0x08.....	17
Memory depth address: 0x0A, 0x0C.....	18
Triggerparameter address: 0x0E.....	18
Data address: 0x00 channel 0, 0x02 channel 1	19
Status address: 0x0E	19
Software Description	20
General Information	20
Header files on CD	20
DLLTYP.H.....	20
SPECTRUM.H	20
REGS.H	20
ERRORS.H.....	20
Driver functions.....	20
int16 SpcInItPCIBoards (int16* count, int16* PCIVersion).....	21
int16 SpcInItBoard (int16 nr, int16 typ).....	21
int16 SpcSetParam (int16 nr, int32 reg, int32 value)	21
int16 SpcGetParam (int16 nr, int32 reg, int32* value).....	21
int16 SpcSetData (int16 nr, int16 ch, int32 start, int32 len, dataptr data)	21
int16 SpcGetData (int16 nr, int16 ch, int32 start, int32 len, dataptr data)	22
Error Codes.....	22
Valid Board Types	22

Hints for programming the boards.....	22
Software - Register	23
Registers	23
Values for command register	23
Values for status register	23
Values for input ranges.....	23
Values for trigger.....	23

Vorwort

Diese Anleitung enthält detaillierte Informationen über die Hardware Möglichkeiten der PAD242 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.

Preface

This manual provides detailed information on the hardware features of the PAD242 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

Die PAD242 ist eine schnelle A/D-Wandler Karte für den ISA Bus. Zwei separate A/D-Wandler ermöglichen die simultane Aufnahme der beiden Kanäle ohne den bei Multiplexsystemen üblichen Zeitversatz.

Verschiedene Betriebsarten wie Speichersegmentierung, interner/externer Takt und Trigger sowie Pre- und Posttrigger erlauben eine flexible Anpassung des Meßsystems.

Der große on-board Speicher ermöglicht die Aufzeichnung extrem breitbandiger Signale.

Anwendungsbeispiele: Spektroskopie, LDA/PDA, Ultraschall, Radar

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 PAD242 away from the power supply-

The PAD242 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 PAD242 in hermetic closed systems.

Order information

PAD242 standard	PAD242 with 256 kSamples memory including drivers	PAD242
Option 1 M	Memory upgrading to 1 MSamples	PAD242-1
Option 4 M	Memory upgrading to 4 MSamples	PAD242-4
Multiple recording	Memory segmentation for fast repetition rates.	PAD242-mr
Input range	3 user specific input ranges between ± 200 mV and ± 3 V, bipolar or unipolar	PAD242-ir
DASYLab driver	Drivers for DASYLab 5.0 for Win 95/98, Win 2000 and Win NT	PAD242-dl
Agilent VEE driver	Drivers for Agilent VEE 5.0 for Win 95/98, Win 2000 and Win NT	PAD242-hp
LabVIEW driver	Drivers for LabVIEW 4.0 for Win 3.11, Win 95/98, Win 2000 and Win NT	PAD242-lv
MatLab driver	Drivers for MatLab 5.0 for Win 95/98, Win 2000 and Win NT	MATLAB

General Information

The PAD242 is a fast A/D-Converter board for ISA bus. Two separate A/D converters allow simultaneously sampling on both channels without the time offset produced by multiplex systems.

Several different operating modes like memory segmentation (multiple recording), internal and external clock and trigger enables a flexible adaption to the measurement problem.

The large on-board memory allows recording of signals with a wide bandwidth.

Application examples: Spectroscopy, LDA/PDA, Supersonics, Radar

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.

Installation

System Anforderungen

PCI basierter IBM kompatibler PC mit mindestens einem freien ISA Steckplatz in der benötigten Kartenlänge. Wenn mehr als eine Karte im System installiert werden soll, so empfehlen wir einen zusätzlichen Lüfter für die Karten einzusetzen.

Hardware Installation

- (1) Stellen Sie auf der Karte eine freie ISA-Adresse ein, wie weiter hinten beschrieben.
- (2) Schalten Sie den PC aus.
- (3) Öffnen Sie das Gehäuse.
- (4) Wählen Sie einen freien ISA 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 plziert ist. Wenn Ihr System einen oder mehrere zusätzliche Lüfter besitzt, so platzieren Sie die Karte direkt in deren Luftstrom.
- (5) Installieren Sie die Karte in dem ausgewählten Steckplatz. Achten Sie dabei besonders auf den korrekten Sitz des ISA Steckers im Steckplatz.
- (6) Schrauben Sie die Karte an der Frontblende am Gehäuse fest.
- (7) Start Sie das System
- (8) Wenn Ihr System nicht bootet, überprüfen Sie bitte den korrekten Sitz der Karte in ihrem Steckplatz. Start Sie danach das System neu.
- (9) Wenn Ihr System immer noch nicht bootet kann es jetzt ein Problem in der Zusammenarbeit mit anderen Karten geben. Stellen Sie eine andere ISA Adresse auf der Karte ein und booten Sie neu.

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.

System Requirements

PCI based IBM PC compatible PC with at least one free ISA slot with the required board length. If you are installing more than one board in your PC, a additional cooling fan is strongly recommended.

Hardware Installation

- (1) Select a free ISA address on the board using the jumpers like described below.
- (2) Power off your PC.
- (3) Open the cover.
- (4) Select a free ISA 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 board in front of a cooling fan.
- (5) Install the board in this slot. Make sure that the ISA connector is right struck into the slot.
- (6) Use a screw to fix the bracket to the PC
- (7) Reboot the system.
- (8) If your system will not boot, please check whether the board is struck correctly into the connector and reboot again.
- (9) If your system will not boot after this, there may be a problem with other boards. Please select another ISA address for your board and boot again.

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.

Installation für Windows 3.x

Der Treiber besteht aus einer 16 Bit Windows DLL, die alle Funktionen des Treibers enthält. Die DLL kann in allen Systemen genutzt werden, die 16 Bit Windows DLLs unterstützen. Die Treiber Dateien können im Verzeichnis \DRIVER\WIN31 auf der Treiber CD gefunden werden. Beispiele für Borland C++ 4.5 sind ebenfalls vorhanden. Zum Installieren des Treibers ist es nur nötig die Datei SPECTRUM.DLL in das Systemverzeichnis (Normalerweise C:\WINDOWS\SYSTEM) zu kopieren. Danach ist der Treiber zur Nutzung bereit. Wenn Sie Borland C++ benutzen, so ist es möglich die mitgelieferte Library Datei SPECTRUM.LIB zu benutzen, um die Funktionen des Treibers einfach in ein eigenes Programm einbinden zu können. Kopieren Sie dazu einfach die Library Datei in Ihr Arbeitsverzeichnis und fügen Sie die Datei in Ihr Projekt ein, wie es auch im Beispiel zu sehen ist.

Achtung, diese Library Datei arbeitet nur mit dem Borland Compiler zusammen. Sie funktioniert nicht mit dem Microsoft Compiler.

Die 16 Bit DLL kann ebenfalls mit den Betriebssystemen Windows 95 und Windows 98 benutzt werden, wir empfehlen aber hierzu die Benutzung der entsprechenden 32 Bit Treiber DLL und VXD.

Installation für Windows 95/98

Die ISA Karten von Spectrum sind nicht Plug-and-Play fähig. Aus diesem Grund werden die ISA Karten nicht automatisch vom System erkannt und die Treiber müssen von Hand installiert werden. Starten Sie das Installationsprogramm auf der mitgelieferten Treiber CD im Verzeichnis \Install\Win9xDrv. Die Treiber stehen sofort nach der Installation ohne Neustart des Systems zur Verfügung.

Die Treiber für Windows 95/98 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++ und für Inprise (Borland) Delphi 4.x 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 Inprise (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.

Installation for Windows 3.x

The driver consists of a 16 bit windows DLL which includes all functions of the driver. The DLL can be used with all systems which accept 16 bit windows DLL's. The driver files are found in the directory \DRIVER\WIN31 on CD. Examples for Borland C++ 4.5 are included. To install the driver just copy the file SPECTRUM.DLL to your system directory (normally C:\WINDOWS\SYSTEM). After this the driver is ready to use. If you are using Borland C++, you may use the library file SPECTRUM.LIB to access the driver functions in a simple way. Just copy the SPECTRUM.LIB file to your working directory and include it in your project like shown in the examples.

Attention, this library file will only work with the Borland compiler. It will not work when using the Microsoft compiler. The 16 bit driver DLL can also be used with the operating systems Windows 95 and Windows 98 but it is strongly recommended to use the corresponding 32 bit driver DLL and VXD.

Installation for Windows 95/98

The ISA boards from Spectrum are not plug-and-play compatible. For this reason the ISA boards are not automatically recognised at system start and the driver must be installed manually. Start the installation program located in directory \Install\Win9xDrv on the driver CD. 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 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 and Inprise (Borland) Delphi 4.x 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 Inprise (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.

Installation für Win NT/2000

Der Treiber besteht aus einem Kernel Mode Treiber für Windows NT / Windows 2000 und einer 32 Bit DLL, die die Funktionen des Kernel Mode Treibers benutzt. Beispiele für Microsoft Visual C++ und Inprise (Borland) Delphi werden ebenfalls mitgeliefert.

- (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) Das Installationsprogramm DRVCONFIG.EXE startet automatisch. Tragen Sie die ISA Karten **hinter** den den PCI Karten in die Liste ein. Bei zwei PCI Karten und zwei ISA Karten müssen die ISA Karten auf Position 3 und 4 eingetragen werden.
- (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 alle Karten korrekt eingetragen wurden.

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 Inprise (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.

Installation for Win NT/2000

The driver consists of a kernel mode driver for Windows NT 4.0 / Windows 2000 and a 32 bit windows DLL which uses the functions of the kernel mode driver. Examples for Microsoft Visual C++ 4.x and Inprise (Borland) Delphi 4.x are included.

- (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 configuration utility DRVCONFIG.EXE starts automatically. Fill in the ISA boards **behind** the PCI boards in the list. If using two PCI boards and two ISA boards, the ISA boards must be set-up on position 3 and 4.
- (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 the setup of all boards.

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 Inprise (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.

Treiber laden

Der Linux Treiber wird als ladbares Kernel Modul spc.o ausgeliefert. Der Treiber enthält alle Spectrum PCI, CompactPCI und ISA Karten. Die ISA Karten müssen beim Laden des Moduls definiert werden.

Laden Sie das Modul mit `insmod -f spc.o type=0x123 io=0x340`. Setzen Sie für den Typ den entsprechenden Typ der ISA Karte ein die installiert werden soll. Eine Übersicht der Typen ist weiter hinten in der Beschreibung zu finden. Als I/O Adresse muß der auf der Karte mit Jumpers eingestellte Wert angegeben werden. Bei PCI und ISA Karten in einem System werden automatisch die PCI Karten an den Anfang der Indizierung gesetzt.

Der `insmod` Befehl generiert die Warnung das das Kernel Modul für eine andere Kernel Version kompiliert wurde. Wenn Sie einen Linux Kernel ≥ 2.0 benutzen können Sie diese Warnung ignorieren.

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 Indizierung beginnt bei 0.

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

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.

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 in den Beispielen ersichtlich.

Info

Informationen über die installierte Spectrum Karten können unter `/proc/spectrum` abgefragt werden. Es werden Typ und Basisadresse angezeigt

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.

Load driver

The linux driver is shipped as the loadable kernel module spc.o. The driver includes all Spectrum PCI, CompactPCI and ISA boards. ISA boards must be defined when the driver is loaded.

ISA boards: Load the module with `insmod -f spc.o type=0x123 io=0x340`. Define the type of ISA board as listed in the manual and select the I/O base address that is set by jumpers on the board. If ISA and PCI boards are mixed in the system the PCI boards are set to the start of the index.

The `insmod` command could generate a warning that the driver module was compiled for an other kernel version. if you are using a linux kernel ≥ 2.0 you could ignore this warning

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

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.

Accessing the linux device is done with read and write commands and `ioctl` commands. This 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. The board type and the base address are listed.

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.

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

Nachdem die PAD242 durch Setzen von Bit 7 im Kontrollregister gestartet wurde, werden die Eingänge abgetastet und die Daten in den Speicher geschrieben. Der Speicher wird als Ringbuffer benutzt, die Daten werden kontinuierlich in den Speicher geschrieben.

Es wird keine Triggerverarbeitung vorgenommen bis der Speicher einmal komplett mit Daten gefüllt ist. Danach wird die Triggererkennung freigeschaltet. Die vier obersten Bits der ADC-Daten werden mit dem programmierten Triggerlevel verglichen. Wenn der Wert von den Signalwerten gekreuzt wird (Von kleiner zu größer bei positiver Flanke, von größer zu kleiner bei negativer Flanke) ist das Triggerereignis detektiert und der Postcounter startet. Wenn der Postcounter die Null erreicht hat, wird die Aufzeichnung der PAD242 gestoppt und die Karte signalisiert „Ready“ (Bit D7 im Statusregister)

Option Multiple Recording

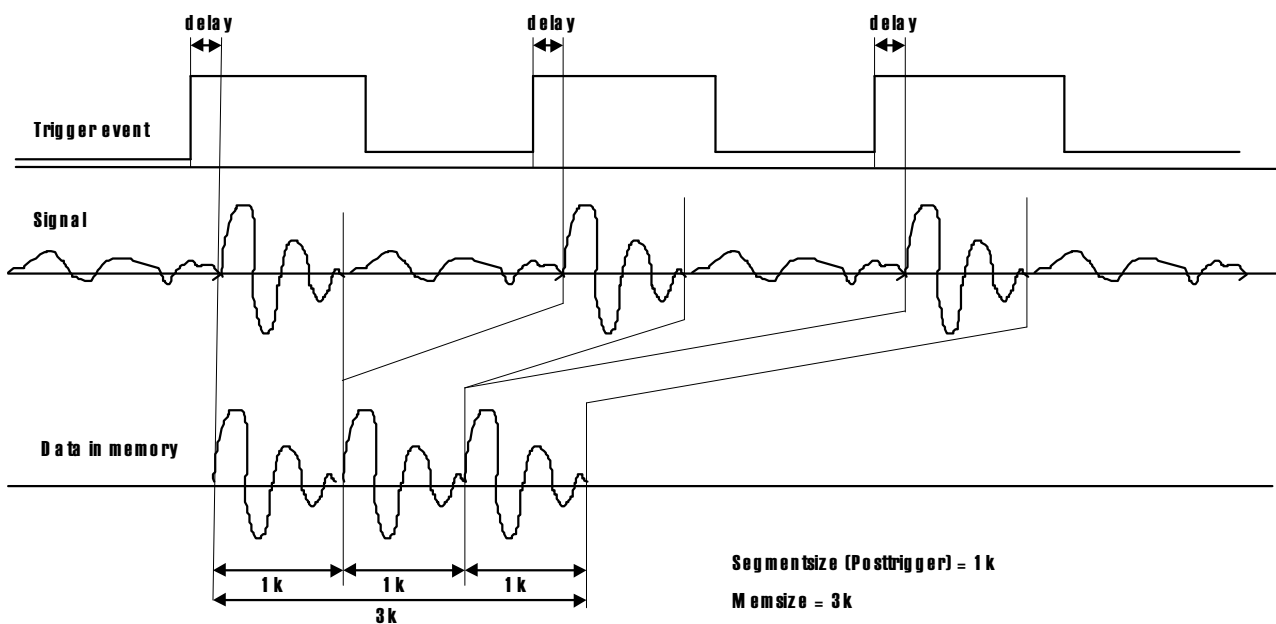
Die Option Multiple Recording erlaubt die Aufnahme/Ausgabe mehrerer Triggerereignisse, ohne die Hardware dazwischen neu zu starten. Der Speicher der Karte wird in mehrere gleich große Segmente unterteilt. Jedes Segment wird bei Auftreten eines Triggerereignisses mit Daten gefüllt. Im Multiple Recording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

Trigger Information

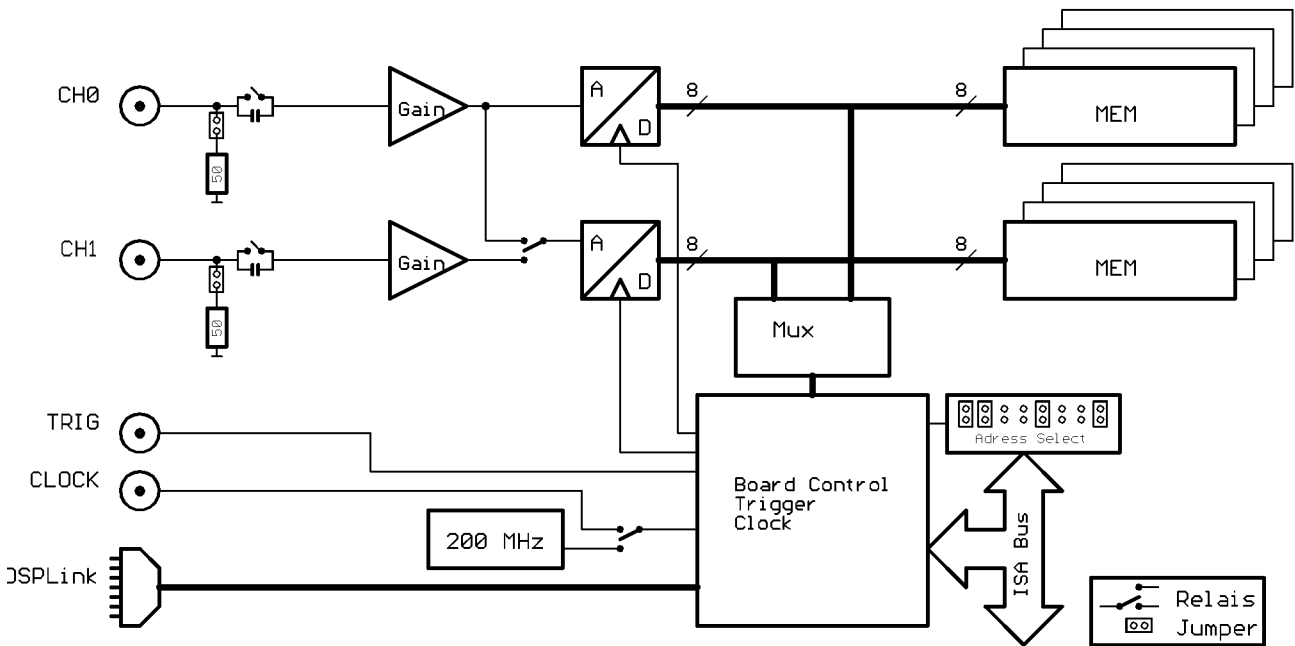
After the PAD242 is started by setting bit 7 in the control register to 1 it samples the input signals and stores the converted data to the memory. (The memory operates as a circular buffer, so data is written continuously to the RAM). No trigger events are processed until the programmed memory is once filled completely with data. Afterwards the trigger sequencer will be enabled. Now the 4 MSBs of the ADC are compared to the programmed value in the trigger-level-register. If this value is crossed by the signal (less to greater when rising edge, or greater to less when falling edge), a trigger event is detected and the postcounter starts counting. After the postcounter reaches zero the PAD242 stops and signals ready (D7 in the status register is cleared to be zero).

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.



Block diagram



Technical data

Resolution	8 bit
Sample rate	781 kHz up to 400 MHz
Bandwidth -3 dB	≥ 150 MHz
Differential linearity error	±0.5 LSB
Integral linearity error	±0.5 LSB
ENOB $f_s = 1$ MHz, $f_{ck}=200$ MHz	6.4 bit typ. (ADC)
ENOB $f_s = 62$ MHz, $f_{ck}=200$ MHz	5.2 bit typ. (ADC)
Aperture jitter	9 ps (ADC)
Input impedance	50 Ohm or 1 MOhm
Overvoltage protection	±20 V
Connector	9 mm BNC female

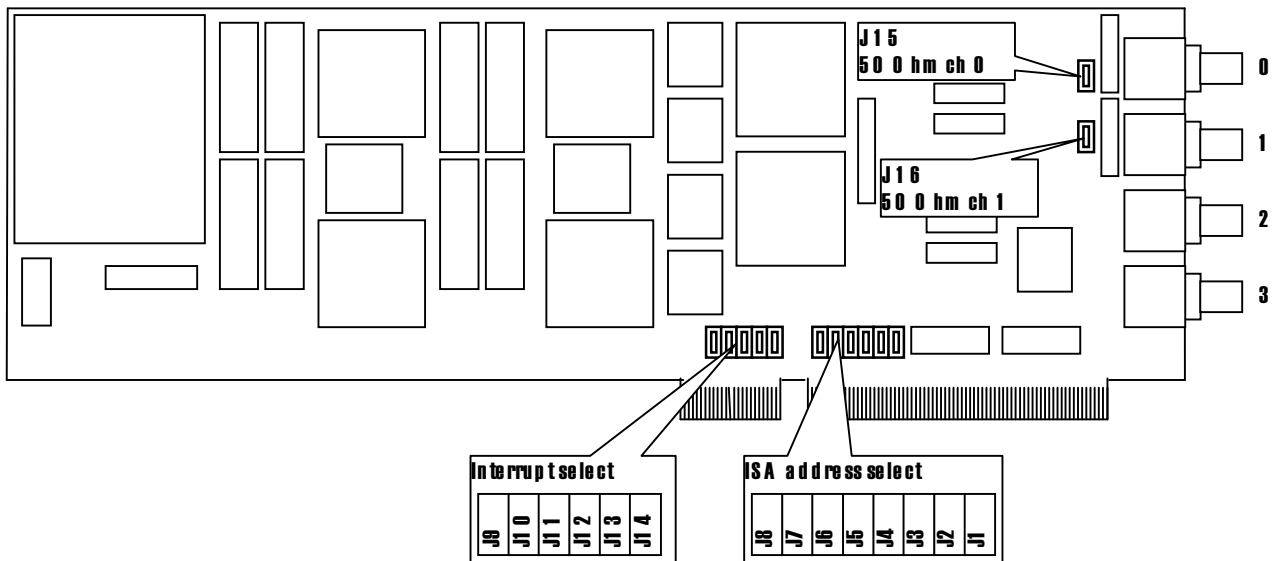
Multi: Trigger to 1 st sample delay	4 samples
Multi: Recovery time	8 samples
Trigger output delay	16 samples
Trigger accuracy (≤200 MHz)	4 samples
Trigger accuracy (400 MHz)	8 samples
Ext. clock: delay to internal clock	≤ 7 ns

Input range	±200 mV	±500 mV	±1 V
Offset error (200 MHz)	≤ 1 LSB	≤ 1 LSB	≤ 1 LSB
Offset error (400 MHz)	≤ 1 LSB	≤ 1 LSB	≤ 1 LSB
Gain error (200 MHz)	≤ 2.5 %	≤ 2.5 %	≤ 1 %
Gain error (400 MHz)	≤ 2.5 %	≤ 2.5 %	≤ 1 %
Noise (200 MHz)	≤ 2 LSB	≤ 2 LSB	≤ 1 LSB
Noise (400 MHz)	≤ 2 LSB	≤ 2 LSB	≤ 1 LSB
Crosstalk	-	-	-

Dimension	312 mm x 109 mm
Warm up time	10 minutes
Operating temperature	0°C - 50°C
Storage temperature	-10°C - 70°C
Humidity	10% to 90% non condensing

	-5 V	+5 V	+12 V	-12 V
Power consumption (A)	0 mA	3800 mA	0 mA	0 mA
Power consumption (W)	0.0 W	19.0 W	0.0 W	0.0 W

Placement



Connectors

The PAD242 has four 9 mm BNC connectors.

- Connector 0: analogue channel 0.
- Connector 1: analogue channel 1.
- Connector 2: triggermode = TTLPOS or TTLNEG: external trigger input
triggermode ≠ TTLPOS and TTLNeg: trigger output
- Connector 3: externalclock input

50 Ohm jumper

channel 0

- J15 set channel 0 50 Ohm
- J15 clear channel 0 1 MOhm

channel 1

- J16 set channel 1 50 Ohm
- J16 clear channel 1 1 MOhm

Addresses

The address is selectable in increment of 16 in the I/O addressspace. Selection is carried out by setting the jumpers J1 - J8. A fitted jumper selects the address signal to be „0“.

jumper	address	value (10)
J1	A4	16
J2	A5	32
J3	A6	64
J4	A7	128
J5	A8	256
J6	A9	512
J7	A14	option jumper should be fitted
J8	A15	option jumper should be fitted

Example:

J6	J5	J4	J3	J2	J1	address
0	0	0	0	0	0	0x0
0	0	0	0	0	1	0x10
0	0	0	0	1	0	0x20
.
1	1	0	1	0	0	0x340
1	1	0	1	1	0	0x360
.
1	1	1	1	1	1	0x3F0

Interrupt line

Selection of the interrupt channel is carried out by using the jumpers J14 - J21. Only one jumper should be fitted!

- J9 IRQ3
- J10 IRQ4
- J11 IRQ5
- J12 IRQ6
- J13 IRQ7
- J14 IRQ9

DSP Link - interface

LSI 50 - Connector (P8)

This connector carries signals to implement a high speed interface to external boards. The following table shows the pinout of the connector:

signalname	Pin	Pin	signalname
GND	50	49	D0
D1	48	47	D2
D3	46	45	D4
D5	44	43	D6
D7	42	41	D8
D9	40	39	D10
D11	38	37	D12
D13	36	35	D14
D15	34	33	GND
A0	32	31	A1
A2	30	29	A3
nc	28	27	nc
nc	26	25	nc
GND	24	23	W
/IOE	22	21	nc
nc	20	19	nc
nc	18	17	nc
nc	16	15	nc
nc	14	13	nc
nc	12	11	GND
nc	10	9	nc
nc	8	7	nc
nc	6	5	nc
nc	4	3	nc
GND	2	1	nc

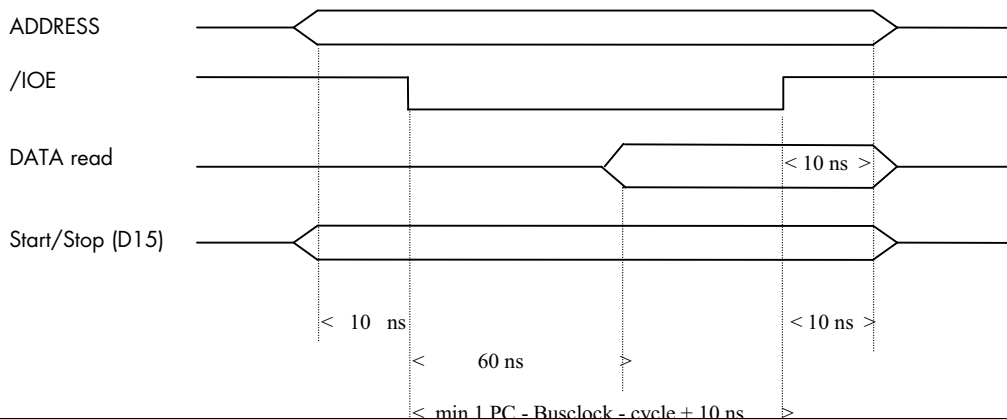
Signaldescription:

- D0 .. D15 -> data read, (D12..D15 LOW - Level) {addr. = 0x00 , 0x01 (channel 0 and channel 1)}
- D15 -> reading, 0 -> recorder activ {addr. = 0x03 (status/command)}
- reading, 1 -> recorder ready { " " }
- writing, 0 -> stop recorder { " " }
- writing, 1 -> start recorder { " " }

- A1, A0 -> channel - address
 - 0 0 -> select channel 0
 - 0 1 -> select channel 1
 - 1 1 -> status/command
- A3, A2 -> Boardaddress. Four addresses are selectable. Selection is carried out by using the jumpers J20 - J23. Only one Jumper should be fitted.
 - 0 0 (0) corresponds with J20
 - 0 1 (4) corresponds with J21
 - 1 0 (8) corresponds with J22
 - 1 1 (C) corresponds with J23
- /IOE -> valid address

Timing specification:

The following setup and hold times will work all DSP Boards.



Hardware Register

Programming of the PAD242 is carried out via eight 8 bit wide registers.

address	read	write
basis + 0x00	data channel 0	controlregister
basis + 0x02	data channel 1	posttrigger low
basis + 0x04	–	posttrigger high
basis + 0x06	–	channel parameter
basis + 0x08	–	sampling frequency
basis + 0x0a	–	memory depth low
basis + 0x0c	–	memory depth high
basis + 0x0e	status	trigger parameter

Controlregister address: 0x00

- D7 = 1 start, 0 = stop
- D6 = 1 multiple recording (option)
- D5 = 1 channel 1 is triggerchannel, 0 = Channel 0 is triggerchannel
- D4 = 1 external clock, 0 = internal clock
- D3 = 1 softwaretrigger, 0 = hardwaretrigger
- D2 = 1 BNC connector P3 **OUTPUT** , 0 = BNC connector P3 **INPUT**
- D1 = not used, write 0
- D0 = not used, write 0

(D7) start

"1" written to this bit starts the PAD242
 "0" stops the board. In this case data in memory are undefined

(D6) multi {option}

"1" written to this bit sets the PAD242 in multiple recording mode
 "0" normal mode

Multiple recording will divide the memory into several segments. Each segment is filled with data on every trigger event. If the option „multiple recording“ is selected the external trigger input has to be used. No other triggersources are available. The segment size is programmed with the posttrigger value (therefore no pretrigger is available, the recorder starts on every trigger edge). Memory size is calculated by:

$$\text{mem.size} = n \times \text{size}$$

where n is the number of segments, size is the segment size (programmed in the posttrigger counter)

(D5) channel

"1" selects channel 1 to be the triggerchannel
 "0" selects channel 0 to be the triggerchannel

(D3) h/s - trigger

"1" select software - trigger
 "0" select hardware - trigger

(D2) BNC connector P3

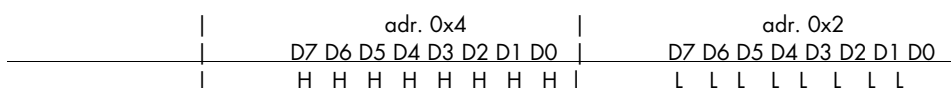
"1" select output, triggersignal
 "0" select input, triggersignal

(D1) clock

"1" selects external clock, $f_{\text{max}} = 200 \text{ MHz}$
 "0" selects internal clock

Posttrigger address: 0x02, 0x4

The posttrigger defines the number of samples stored **after** a trigger event is to be found. The corresponding pretrigger is calculated by the formula: pretrigger = memorydepth - posttrigger



H = high part, L = low part

The posttrigger can be programmed in increments of 64. Minimum is 64. Maximum is 20971152

(2 MSamples).

The programmed value is calculated by the formula:

$$\text{programmed value} = (\text{intended posttrigger})/64 - 1$$

Example:	posttrigger 64 Sample:	64/64 - 1	= 0	= 0x0000
	posttrigger 2048 Sample:	2048/64 - 1	= 31	= 0x001F
	posttrigger 32 k Sample:	65536/64 - 1	= 1023	= 0x03FF
	posttrigger 1M Sample:	2 M/64 - 1	= 32767	= 0x7FFF

If the posttrigger is greater than the mem-depth the memory will be overwritten „n“-times.

Channelparameter address: 0x06

D7	=	mode
D6	=	AC coupling channel 1
D5,D4	=	input range channel 1
D3	=	reserved, write 0
D2	=	AC coupling channel 0
D1,D0	=	input range channel 0

(D7) mode

If bit D7 is set to „0“ the PAD242 is set to the 400 MHz interlace mode. In this case the register holding the value of the sampling clock must be programmed with 0x00!

Only signals of the analogue channel 0 are measured in the 400 MHz mode.

(D6;D2) coupling

AC/DC coupling channel 0 and channel 1

D6,D2 = 0 = DC coupling

D6,D2 = 1 = AC coupling

(D5;D4) (D1;D0) input range

Gain selection. D5,D4 corresponds with channel 1, D1,D0 with channel 0.

D5(D1)	D4(D0)	input range
0	0	reserved
0	1	±200 mV
1	0	±500 mV
1	1	±1 V

The input range is selected with the help of relais. When changing channelparameters do not start the PAD242 immediately. Minimum settling time for the relais circuit is 0.2s!

Sampling frequency address: 0x08

Possible programmed values are:

D7	D6	D5	D4	D3	D2	D1	D0	sampling frequency/MHz	prog. value
0	0	0	0	0	0	0	0	200/400	0x00
0	0	0	1	0	0	0	1	100	0x11
0	0	0	1	0	0	1	0	66.667	0x12
0	0	0	1	0	0	1	1	50	0x13
0	0	0	1	0	1	0	0	40	0x14
0	0	0	1	0	1	0	1	33.334	0x15
0	0	0	1	0	1	1	0	25	0x16
0	0	0	1	0	1	1	1	20	0x17
0	0	0	1	1	0	0	0	15.385	0x18
0	0	0	1	1	0	0	1	12.5	0x19
0	0	0	1	1	0	1	0	10	0x1A
0	0	0	1	1	0	1	1	6.667	0x1B
0	0	0	1	1	1	0	0	5	0x1C
0	0	0	1	1	1	0	1	2	0x1D
0	0	0	1	1	1	1	0	1	0x1E
0	0	0	1	1	1	1	1	0.781	0x1F

For other sampling freq. please contact factory.

Memory depth address: 0x0A, 0x0C

adr. 0x0C								adr. 0x0A							
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
H	H	H	H	H	H	H	H	L	L	L	L	L	L	L	L

H = high part, L = low part

Memory depth can be programmed in increments of 32.

The programmed value is calculated by the formula:

$$\text{programmed value} = 65535 - (\text{intended memory depth}/32) - 1$$

Example: intended mem. 256 Sample, max.mem = 2M :

$$65535 - 256/32 - 1 = 65528 = \text{OxFFF8}$$

$$\text{address 0x0C} = \text{OxFF} \quad \text{address 0x0A} = \text{OxF8}$$

Example: intended mem. 32 kSamples, max.mem = 2M :

$$65535 - 32768/32 - 1 = 64512 = \text{OxFC00}$$

$$\text{address 0x0C} = \text{OxFC} \quad \text{address 0x0A} = \text{Ox00}$$

Example: intended mem. 1 MSamples, max.mem = 2M :

$$65535 - 1048576/32 - 1 = 32768 = \text{Ox8000}$$

$$\text{address 0x0C} = \text{Ox80} \quad \text{address 0x0A} = \text{Ox00}$$

By programming a deeper memory than physically installed on the board (see memory options) the memory will be "n"-times overwritten (n = programmed depth/installed memory).

Triggerparameter address: 0x0E

D7 = triggerselection
D6..D5 = not used, write 0
D4 = triggeredge
D3..D0 = triggerlevel

(D7) ext./int. trigger:

D7 = 0 internal trigger
D7 = 1 external trigger

(D4) edge (D4):

D4 = 0 rising edge
D4 = 1 falling edge

(D[3..0]) level:

The programmed triggerlevel is compared to the 4 MBSs of the ADC.
Resolution depends on the programmed input range.

resolution for the input range: 2Vpp = 0,125 V 1Vpp = 0.0625 V 0.4Vpp = 0,025 V

input ±200 mV	input ±500 mV	input ±1 V	prog. value D[3..0]
-0,175 V	-0,4375 V	-0,875 V	0x0
-0,150 V	-0,3750 V	-0,750 V	0x1
-0,125 V	-0,3125 V	-0,625 V	0x2
-0,100 V	-0,2500 V	-0,500 V	0x3
-0,075 V	-0,1875 V	-0,375 V	0x4
-0,050 V	-0,1250 V	-0,250 V	0x5
-0,025 V	-0,0625 V	-0,125 V	0x6
0,000 V	0,000 V	0,00 V	0x7
0,025 V	0,0625 V	0,125 V	0x8
0,050 V	0,1250 V	0,250 V	0x9
0,075 V	0,1875 V	0,375 V	0xA
0,100 V	0,2500 V	0,500 V	0xB
0,125 V	0,3125 V	0,625 V	0xC
0,150 V	0,3750 V	0,750 V	0xD
0,175 V	0,4375 V	0,875 V	0xE

The trigger- out signal has a delay of 6 samples.

Data address: 0x00 channel 0, 0x02 channel 1

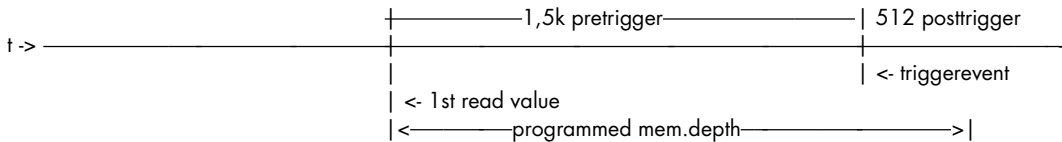
The data are read by accessing 0x00 or 0x02. An internal counter increments on each read cycle. The reading data format is binary offset.

!Once reading data from one channel has been started it must be completed (memory depth) before start reading data from the other channel!

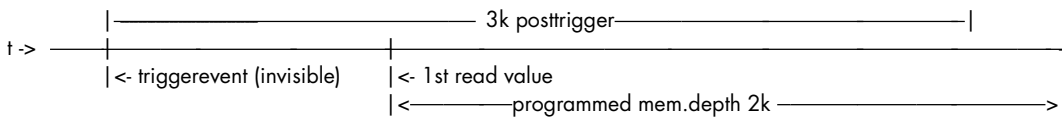
The trigger event is found at the position: (mem.depth) minus (posttrigger)

If the programmed posttrigger is larger than the memory depth only data after the trigger event will be stored.

Example 1: mem.depth = 2K-Sample, posttrigger = 512 Sample



Example 2: mem.depth = 2K-Sample, posttrigger = 3k-Sample



Status address: 0x0E

D7 = 1 PAD242 active, samples data
All other bits are undefined.

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	DAP116	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 *SpcNnitPCIBoards*
- * 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 *SpcNnitPCIBoards* first and initialise the ISA boards after this. The function *SpcNnitPCIBoards* 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 *SpcNnitPCIBoards* and *SpcNnitBoard*. 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.

The PAD242 must be initialized with the function *SpcNnitBoard* first. The functions *SpcSetParam*, *SpcGetParam* and *SpcGetData* are implemented. The function *SpcGetData* will ignore the parameters *Start* and *Len*. The Registers *SPC_ISAADR* and *SPC_INSTMEM* must be set before writing or reading any of the other registers.

Registers

name	value (dec)	r/w	
SPC_COMMAND	0	w	command register, allowed values listed below
SPC_STATUS	10	r	status register, possible values listed below
SPC_ISAADR	1010	w	installation address: 0x200 up to 0x3F0 in step of 0x10
SPC_INSTMEM	1020	w	installed memory: 256k, 1M, 4M
SPC_MEMSIZE	10000	w	memory size for recording : 32 up to instmem in steps of 32 in interlace mode, all values are doubled
SPC_POSTTRIGGER	10100	w	posttrigger: 64 up to 2M in steps of 64
SPC_SAMPLERATE	20000	w	samplerate for recording: see listed values in the hardware description
SPC_EXTERNALCLOCK	20100	w	external clock: 0x00 for disable, 0x01 for enable
SPC_AMPO	30010	w	input range for channel 0: one of the below listed values
SPC_ACDCO	30020	w	AC coupling channel 0: 0x00 for DC, 0x01 for AC
SPC_AMP1	30110	w	input range for channel 1: one of the below listed values
SPC_ACDC1	30120	w	AC coupling channel 1: 0x00 for DC, 0x01 for AC
SPC_TRIGGERMODE	40000	w	trigger mode for board: one of the below listed values
SPC_TRIGGERLEVEL	42000	w	trigger level for channel trigger: value from 0 to 14
SPC_MULT	220000	w	multiple recording (option): 0x00 for disable, 0x01 for enable

Values for command register

name	value (dec)	
SPC_START	10	starts the board with the actual setup
SPC_STOP	20	stops the board and resets the logic

Values for status register

name	value (dec)	
SPC_RUN	0	hardware is running: the board is waiting for trigger or recording data
SPC_READY	20	recording has stopped, data are available

Values for input ranges

name	value (dec)	
AMP_BI200	200	input range +/- 200 mV
AMP_BI500	500	input range +/- 500 mV
AMP_BI1000	1000	input range +/- 1 V

Values for trigger

name	value (dec)	
TM_SOFTWARE	0	trigger immediately with software
TM_CH0POS	10000	trigger channel 0 positive edge
TM_CH0NEG	10010	trigger channel 0 negative edge
TM_CH1POS	10100	trigger channel 1 positive edge
TM_CH1NEG	10110	trigger channel 1 negative edge
TM_TTLPOS	20000	external trigger TTL positive edge
TM_TTLNEG	20010	external trigger TTL negative edge