

PCK400
400 MHz clock generator
for ISA bus

Hardware Manual
Driver Manual

© Spectrum Systementwicklung Microelectronic GmbH - 2002
Bültbek 26, 22962 Siek, Germany

SBench is a registered trademark of Spectrum Systementwicklung Microelectronic GmbH.

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

LabVIEW is a trademark of National Instruments Corporation.

MATLAB is a registered trademark of The MathWorks Inc.

Agilent VEE is a trademark of Agilent.

FlexPro is a registered trademark of Weisang & Co.

DASYLab is a registered trademark of DATALOG GmbH.

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

Table of Contents

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
Installation for 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
Block diagram.....	12
Technical data.....	12
Placement.....	13
Addresses	13
Hardware Register	14
General information	14
Programming	14
Write address: 0x00 trigger/command	14
Write address: 0x01 output/PLL1	14
Write address: 0x02 PLL2	15
Write address: 0x03, 0x04, 0x05 divider	15
Write address: 0x08 pin high level.....	15
Write address: 0x09 pin low level.....	16
Read address: 0x00 Status	16
Software Description	17
General Information	17
Header files on CD	17
DLLTYP.H.....	17
SPECTRUM.H	17
REGS.H	17
ERRORS.H.....	17
Driver functions.....	17
int16 SpclnitPCIBoards (int16* count, int16* PCIVersion).....	18
int16 SpclnitBoard (int16 nr, int16 typ).....	18
int16 SpcSetParam (int16 nr, int32 reg, int32 value)	18
int16 SpcGetParam (int16 nr, int32 reg, int32* value).....	18
int16 SpcSetData (int16 nr, int16 ch, int32 start, int32 len, dataptr data)	18
int16 SpcGetData (int16 nr, int16 ch, int32 start, int32 len, dataptr data)	19
Error Codes.....	19
Valid Board Types	19
Hints for programming the boards.....	19
Software – Register	20
Registers	20
Values for command register	20
Values for status register	20

Values for trigger.....20

Vorwort

Diese Anleitung enthält detaillierte Informationen über die Hardware Möglichkeiten der PCK400 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 PCK400 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

Ein digitales Taktsignal mit einstellbaren Pegeln lässt sich mit der PCK400 auf einfache Art und Weise erzeugen. Damit steht eine Bandbreite von Ausgangssignalen zur Verfügung, die auch ECL, PECL, TTL und CMOS mit einschließt.

Die Frequenz kann ohne Stoppen der Ausgabe mit einer Auflösung von 0,018 dB geändert werden.

Durch den Trigger-/ Gate-Eingang besteht die Möglichkeit, die PCK400 mit externen Schaltungen zu synchronisieren.

Anwendungsbeispiele: Automatische Testsysteme, Prüfstände, Qualitätssicherung, Forschung und Entwicklung

Software

Kostenlos mitgeliefert werden Treiber für Linux, DOS, Windows 3.1x, Windows 9x/ME und Windows 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 PCK400 away from the power supply-

The PCK400 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 PCK400 in hermetic closed systems.

Order information

PCK400 standard	PCK400 including drivers	PCK400
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	PAD1616-dl
HP-VEE driver	Drivers for HP-VEE 5.0 for Win 95/98, Win 2000 and Win NT	PCK400-hp
LabVIEW driver	Drivers for LabVIEW 4.0 for Win 3.11, Win 95/98, Win 2000 and Win NT	PCK400-lv
MatLab driver	Drivers for MatLab 5.0 for Win 95/98, Win 2000 and Win NT	MATLAB

General Information

A digital clock signal with adjustable level may be easily generated by the PCK400. With this pin driver a wide range of output signals including ECL, PECL, TTL and CMOS is possible.

The frequency may be changed without stopping the output. The stepwidth of the frequency is 0.018 dB.

using the trigger and gate input offers the possibility of synchronising the PCK400 with external systems.

Application examples: Automatic testequipment, test bed, Quality management, Research and development

Software

Drivers for Linux, DOS, Windows 3.1x, Windows 9x/ME and Windows 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.

Installation für 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.

Installation for 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 Indexzählung 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

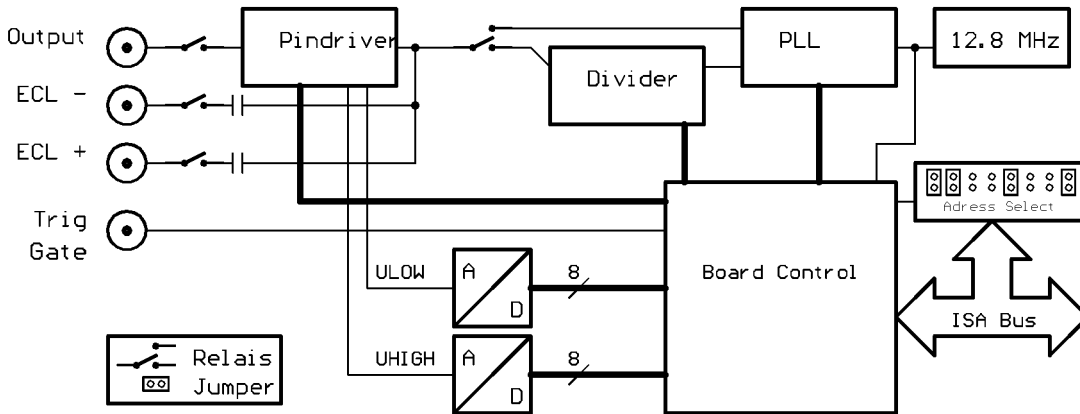
Installed in the folder ‚Spectrum GmbH‘ when installing the Windows NT driver. This utility manages the driver configuration of the Spectrum ISA boards for Windows NT. The program need not to be used for PCI boards. The utility changes the registry. The new configuration will only be used after the next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

Block diagram

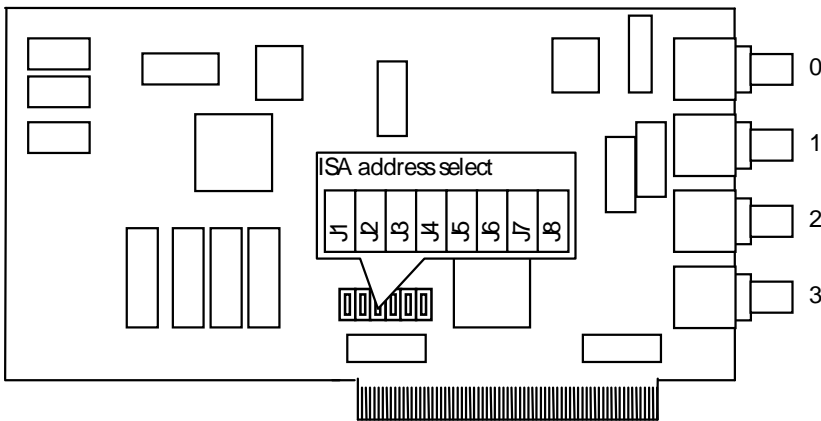


Technical data

clock frequency in 50 Ohm	0.75 Hz up to 400 MHz	Dimension	160 mm x 113 mm
Clock jitter	±25 ps rms	Connector	9 mm BNC female
Frequency stability over temperature	±50 ppm	Trig/Gate impedance	≥ 1 kOhm at TTL level
ECL bandwidth AC-coupled	-3 dB @ 11.1 kHz	Overtoltage protection	±20 V
Pindriver bandwidth (high-low) ≤ 0.8 V	0.75 Hz to 250 MHz	Warm up time	10 minutes
Pindriver bandwidth (high-low) ≤ 5.0 V	0.75 Hz to 105 MHz	Operating temperature	0°C - 50°C
Pindriver low level range	-3.33 V up to + 3.3 V	Storage temperature	-10°C - 70°C
Pindriver high level range	-1.83 V up to + 5.51 V	Humidity	10% to 90% non condensing
Pindriver high level – low level	≥ 0.3 V		
Pindriver resolution	8 bit		
Pindriver Output to ECL + delay	-1 ns		
Pindriver shortcut time	∞		
ECL signal rise time 20 % to 80 %	≤ 1 ns		
ECL signal fall time 80 % to 20 %	≤ 0.9 ns		

	-5 V	+5 V	+12 V	-12 V
Power consumption (A)	0 mA	1000 mA	100 mA	0 mA
Power consumption (W)	0.0 W	5.0 W	1.2 W	0.0 W

Placement



Connectors

The PCK400 has four 9 mm BNC connectors.

- Connector 0: Pin driver output
- Connector 1: ECL negative output, AC coupled
- Connector 2: ECL positive output, AC coupled
- Connector 3: Trigger/gate input

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
.
1	1	0	1	0	0	0x340
1	1	0	1	1	0	0x360
.
1	1	1	1	1	1	0x3F0

Hardware Register

General information

The PCK400 generates a clock signal with the help of a PLL and a divider. The PLL generates a clock between 25 MHz and 400 MHz. This clock may be divided into slower clock rates using a 24 bit divider. The output may be started with software or by using the trigger/gate input 3. The PLL and the divider may be changed while the clock signal is generated. The clock signal will be put out using a pin driver or using two ECL outputs. The pin driver is capable of driving -3.3 V up to 5.5 V in 50 Ω. The maximum output frequency for the pin driver at ECL level is 200 MHz.

Programming

Programming of the PCK400 is carried out by using eight 8 bit wide registers.

address	read	write
base + 0x00	status	trigger/command
base + 0x01	–	output/PLL1
base + 0x02	–	PLL2
base + 0x03	–	divider 0..7
base + 0x04	–	divider 8..15
base + 0x05	–	divider 16..23
base + 0x06	–	reserved
base + 0x07	–	reserved
base + 0x08	–	Pin high
base + 0x09	–	Pin low

Write address: 0x00 trigger/command

D0..D2 triggermode
 D3 stop level
 D4 reserved
 D5 divider enable
 D6 program PLL
 D7 program divider

D0..D2 Triggermode

D2	D1	D0	
0	0	0	gated output at high level on input3
0	0	1	gated output at low level on input3
0	1	0	output started with negative edge, stopped with positive edge on input3
0	1	1	output started with positive edge, stopped with negative edge on input3
1	x	0	output stopped with software
1	x	1	output started with software

D3 Stop level

„0“ stop level is low
 „1“ stop level is high

This value must be set before the output relais is set on.
 The stop level will be set on the output if the board is stopped.

D5 Divider enable

„0“ divider disabled, the PLL output is directly used
 „1“ divider enabled, the PLL output is divided by the divider value in register 0x03 - 0x05. The divider may only be enabled if the internal PLL divider „N“ is set to the value „16“ in the register 0x02.

D6 Program PLL

„1“ The PLL values in register 0x01 and 0x02 are put in the PLL, the new clock is given out

D7 Program divider

„1“ The divider values in register 0x03 - 0x05 are put in the divider the new clock is given out

Write address: 0x01 output/PLL1

D0 relais ECL
 D1 relais pin driver
 D2..D7 PLL M0..M5

D0 Relais ECL

„0“ ECL outputs disabled
 „1“ ECL outputs enabled

D1 Relais pin driver

„0“ pin driver output disabled
 „1“ pin driver output enabled

D2..D7 PLL M0..M5

reference clock multiplier for PLL bit 0..5

Write address: 0x02 PLL2

D0..D2 PLL M6..M8
 D3 PLL N0
 D4 PLL N1
 D5..D7 reserved

D0..D2 PLL M6..M8

reference clock multiplier for PLL bit 6..8

D3..D4 PLL N0, N1

PLL divider

D4	D5	
0	0	N = 2
0	1	N = 4
1	0	N = 8
1	1	N = 16

The PLL output clock is calculated by the formula:

$$\text{PLL output} = 12.75 \text{ MHz}/8 * M/N$$

The value M may be set in the range from 251 to 502

Write address: 0x03, 0x04, 0x05 divider

address 0x03 D0..D7 divider bit 0..7
 address 0x04 D0..D7 divider bit 8..15
 address 0x05 D0..D7 divider bit 16..23

If the divider is enabled (adr 0x00 D5) it will divide the PLL clock with the set value.
 The output clock is calculated by the formula

$$\text{output} = 12.75 \text{ MHz}/8 * M/N/D$$

$$D = (2 * (\text{divider} + 1))$$

Examples for clock output (both relais are disabled)

M	N	D	divider enable	adr 0x01	adr 0x02	adr 0x03	adr 0x04	adr 0x05	clock
502	2	x	0	0xD8	0x07	0x00	0x00	0x00	400.02 MHz
501	2	x	0	0xD4	0x07	0x00	0x00	0x00	399.23 MHz
501	16	2	1	0xEC	0x07	0x00	0x00	0x00	24.95 MHz
501	16	1000	1	0xEC	0x07	0xF3	0x01	0x00	49.90 kHz
300	8	x	0	0xB0	0x14	0x00	0x00	0x00	59.76 MHz
251	16	x	0	0xEC	0x1B	0x00	0x00	0x00	25.00 MHz
251	16	2*2^24	1	0xEC	0x1B	0xFF	0xFF	0xFF	0.75 Hz

Write address: 0x08 pin high level

D0..D7 DAC value for high level

The high level for the pin driver is set with the help of a DAC. The high value is set in the range -1.83 V up to +5.51 V. The step size is 28.78 mV

value	high level
0x00	-1.83 V
0x01	-1.80 V
0x02	-1.77 V
...	...
0x40	0 V
...	...
0xFD	+5.45 V
0xFE	+5.48 V
0xFF	+5.51 V

Write address: 0x09 pin low level

D0..D7 DAC value for low level

The low level for the pin driver is set with the help of a DAC. The high value is set in the range -3.33 V up to +3.30 V. The step size is 26 mV

value	low level
0x00	-3.33 V
0x01	-3.07 V
0x02	-2.81 V
...	...
0x80	0 V
...	...
0xFD	+2.78 V
0xFE	+3.04 V
0xFF	+3.30 V

There must be a difference of at least 300 mV between the high level and the low level must. The high level must not be lower than the low level.

The maximum output swing of the pin driver depends on the output frequency. At low clock rates an output swing of 8.84 Vpp is possible. Up to 105 MHz an output swing of about 5 Vpp is possible. For frequencies up to 250 MHz only ECL output swing (0.8 Vss) is allowed.

Read address: 0x00 Status

D0..D5 Reserved
 D6 Pin driver temperature
 D7 input3

D6 Pin driver temperature

„0“ The pin driver has stopped the output because of heating problems. The pin driver relais is set off.
 „1“ Normal output

D7 input3

Shows the state of the external input3

Software Description

Allgemeine Information

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

Header Dateien auf CD

DLLTYP.H

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

SPECTRUM.H

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

REGS.H

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

ERRORS.H

Listet alle möglichen Errorcodes der Funktionen auf.

Funktionen des Treibers

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

General Information

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

Header files on CD

DLLTYP.H

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

SPECTRUM.H

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

REGS.H

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

ERRORS.H

Lists all possible error codes of the functions.

Driver functions

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

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

int16 SpcInitPCIBoards (int16* count, int16* PCIVersion)

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

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

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

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

int16 SpcInitBoard (int16 nr, int16 typ)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Error Codes

error name	value (hex)	value (dec.)	description
ERR_OK	0	0	Execution OK, no error.
ERR_INIT	1	1	The board number is not in the range of 0 to 15. When initialisation is executed: the board number is yet initialised, the old definition will be used.
ERR_NR	2	2	The board is not initialised yet. Use the function <i>SpcInitBoard</i> or <i>SpcInitPCIBoards</i> first.-
ERR_TYP	3	3	Initialisation only: The type of board is unknown.
ERR_FNCNOTSUPPORTED	4	4	This function is not supported by the hardware version.
ERR_LASTERR	10	16	Old Error waiting to be read.
ERR_ABORT	20	32	Abort of wait function
ERR_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 ios 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_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_FIFOBUFFEROVERRUN	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_TIMESTAMP_SYNC	310	784	Synchronisation to external reference clock failed.

Valid Board Types

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

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

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

Hints for programming the boards

Programming an ISA board is done in the following steps:

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

Programming an PCI board is done by the following steps:

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

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

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

Software - Register

The PCK400 must be initialised with the function *SpclnitBoard* first. The functions *SpcSetParam*, *SpcGetParam* and *SpcGetData* are implemented. The function *SpcGetData* will ignore the parameters *Start* and *Len*. The Register *SPC_ISAADR* 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_TRIGGERMODE	40000	w	trigger mode for board: one of the below listed values
SPC_FREQUENCE	300000	r/w	Set frequency for PCK400 in Hz in the range 1 HZ to 400 MHz. The value will be rounded to the next matching value. The new frequency could be read out again with this register.
SPC_DELTAFREQUENCE	300010	w	Change frequency with a delta value. The value will be added to the actual frequency. The new frequency could be read out with SPC_FREQUENCE.
SPC_PINLOW	300100	r/w	Set low level for pin driver in the range -3330 mV to 3300 mV. The value will be rounded to the next matching value. The new low level could be read out again with this register.
SPC_PINHIGH	300110	r/w	Set high level for pin driver in the range -1830 mV to 5510 mV. The value will be rounded to the next matching value. The new high level could be read out again with this register.
SPC_PINDELTA	300120	w	Change low and high level of pin driver with a delta value in mV. The value will be added to the low and to the high level of the pin driver. The new values could be read out in the registers SPC_PINLOW and SPC_PINHIGH.
SPC_STOPLEVEL	300200	w	Stop level PCK400: 0x00 for low level, 0x01 for high level.
SPC_EXTERNLEVEL	300300	r	Read out extern input 3. 0x00 for low level, 0x01 for high level.
SPC_PINRELAIS	300210	r	Read out temperature relais: 0x01 for ok, 0x00 for relais set off because of heating problems.

Values for command register

name	value (dec)	
SPC_START	10	Starts the board with the actual setup. Relais for output will be set on.
SPC_STOP	20	Stops the board.
SPC_RELAISON	200	Set output relais on.
SPC_RELAISSOFF	210	Set output relais off.

Values for status register

name	value (dec)	
SPC_RUN	0	Hardware is running.
SPC_READY	20	No clock is generated. Board is waiting for start.

Values for trigger

name	value (dec)	
TM_SOFTWARE	0	start immediately with software
TM_TTLPOS	20000	external trigger TTL positive edge
TM_TTLNEG	20010	external trigger TTL negative edge
TM_GATELOW	30000	external gate for output: LOW
TM_GATEHIGH	30010	external gate for output: HIGH