SPECTRUM
SYSTEMENTWICKLUNG MICROELECTRONIC GMBH

MATLAB driver

for Spectrum boards

User Manual

deutsch/english

SPECTRUM SYSTEMENTWICKLUNG MICROELECTRONIC GMBH • AHRENSFELDER WEG 13-17 • 22927 GROSSHANSDORF • GERMANY
TEL.: +49 (0)4102-6956-0 • FAX: +49 (0)4102-6956-66 • EMAIL: info@spec.de • INTERNET: http://www.spec.de
13.03.2007

Ahrensfelder Weg 13-17, 22927 Grosshansdorf, Germany

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.

Treiber – Struktur

Der vorliegende MATLAB-Treiber setzt auf den beiliegenden Spectrum Treibern für Windows 9x/ME und Windows NT/2000/XP auf. Der MATLAB-Treiber besteht aus einem Satz DLLs, die die Funktionen des Spectrum Treibers unter MATLAB zur Verfügung stellen. Die DLLs funktionieren nur als Bindeglied zwischen MATLAB und den vorhandenen Spectrum Treibern. Für jede Funktion der Treiber ist eine eigene DLL vorhanden.
Ferner wird für jede unterstützte Karte ein Programmierbeispiel mitgeliefert.

Driver Structure

The MATLAB driver is based on the also delivered Spectrum driver for Windows 9x/ME and Windows NT/2000/XP. The MATLAB driver consists of a set of DLL's translating the functions of the Spectrum driver to MATLAB. The DLL's are only working as links between MATLAB and the normal Spectrum drivers. Each function of the driver is represented by one DLL.
There is also a programming example for every supported board on the disk.

Installation

1. Installieren Sie die Spectrum Treiber von der mitgelieferten CD wie im Hardware Handbuch beschrieben.
2. Führen Sie den Installer aus dem Verzeichnis D: \Install\matlab_drv\ aus.
3. Bitte halten Sie für die Installation die mitgelieferte MATLAB Lizenz bereit.
4. Die Treiber und Beispiele werden je nach gewählter MATLAB Version in folgende Ordner im MATLAB Installationsverzeichnis abgelegt:
   - Treiber: [MATLAB Install Path]\spectrum\spectrumDrv
   - Beispiele: [MATLAB Install Path]\ spectrum\examples
5. Starten Sie MATLAB
6. Fügen Sie das Treiber Verzeichnis dem Pfad von MATLAB hinzu. Klicken Sie dazu im Menü ‚File', ‚Set Path' und klicken Sie auf den ‚Add To Path' button.
7. Die neuen Funktionen stehen Ihnen nun unter MATLAB zur Verfügung.

Installation:

1. Install the Spectrum driver from the delivered driver CD as described in the hardware manual.
2. Start the installer, that in located in the directory: D: \Install\matlab_drv\
3. For the installation of the driver, you will need the provided MATLAB license.
4. The driver and the examples will be installed in the following subdirectories in the MATLAB install folder, depending on your choice of the installed MATLAB version:
   - Driver: [MATLAB Install Path]\spectrum\spectrumDrv
   - Examples: [MATLAB Install Path]\spectrum\examples
5. Start MATLAB.
6. Add the path of the spectrum driver directory to MATLAB. Therefore select ‚File' and ‚Set Path' in the menu and click on the ‚Add to Path' button..
7. The functions are now available under MATLAB.

Programmierung

Die Programmierung der Karten unter MATLAB folgt analog der Programmierung unter Microsoft Visual C++ oder Borland Delphi. Der Zugriff auf die einzelnen Boards funktioniert über eine Board-Nummer, die bei der Initialisierung angegeben wird. Eine genaue Beschreibung der Software Register und der erlaubten Werte ist in der Hardware Beschreibung zu finden. Beispiele zur Programmierung sind in dem jeweiligen Beispielprogramm zu finden.

Programming

The programming of the boards using MATLAB is carried out in the same way as programming with Microsoft Visual C++ or Borland Delphi. Access to one board is realized with a board number which is defined on the initialising of the boards. A description of the software registers and the accepted values is found in the hardware manual.
Examples for programming can be found in the "spectrum" directory.

Function SpcInitBoard: [ErrorCode] = SpcInit (Nr, Typ)

ErrorCode:	Errorcode of function as listed in the hardware manual
Nr:	Number of the board to be defined. In the reange 0-15
Typ:	Type of the defined board. See Manual for a description of valid board types

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 uses just the board number to access the board. After initialising the board, all parameters will be set to default values.

Function SpcInitPCIBoards: [ErrorCode, Count, PCIVersion] = SpcInPCI

ErrorCode:	Errorcode of function as listed in the hardware manual
Count:	Number of found PCI boards from Spectrum
PCIVersion:	Found PCI bus version

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

Function SpcSetParam: [ErrorCode] = SpcSetPa (Nr, Reg, Value)

ErrorCode:	Errorcode of function as listed in the hardware manual.
Nr:	Number of board as defined by SpcInit or SpcInPCI.
Reg:	Software register to be changed.
Value:	New value for the softwar register.

Sets a software register or executes a command. The board must be initialized before. When using ISA boards, all installation parameters must be set before (adress, installed memory). The allowed registers for the specific board are listed in the hardware manual.

Function SpcGetParam: [ErrorCode, Value] = SpcGetPa (Nr, Reg)

ErrorCode:	Errorcode of function as listed in the hardware manual.
Nr:	Number of board as defined by SpcInit or SpcInPCI.
Reg:	Software register to be read.
Value:	Value from the softwar register.

Reads a software register or status information. The board must be initialized before. When using ISA boards, the installation adress must be set before. The allowed registers ,for this board are listed in the hardware manual.

Function SpcGetData: [ErrorCode, Data] = SpcGetDa (Nr, MemCh, Start, Len, Bits)

ErrorCode:	Errorcode of function as listed in the hardware manual.
Data:	Array of the returned values.
Nr:	Number of board as defined by SpcInit or SpcInPCI.
MemCh:	Number of memory channel to be read, each memory channel contains multiplexed data of one module
Start:	First value to be read (On ISA boards read out starts with zero)
Len:	Number of values to be read.
Bits:	Number of Bits for one sample. Valid values are 8 for 8 bit boards, 16 for 12 bit and 16 bit boards.

Reads data from a specific channel. The board must be initialized before. When using ISA boards, all installation parameters must be set before (adress, installed memory).

Function SpcGetSplittedData: [ErrorCode, DigitalData, AnalogData] = SpcGetSplittedDa (Nr, MemCh, Start, Len, BitsAnVal)

ErrorCode:	ErrorCode of function as listed in the hardware manual.
DigitalData:	Array of the returned digital values.
AnalogData:	Array of the returned analog values.
Nr:	Number of board as defined by SpcInit or SpcInPCI.
MemCh:	Number of memory channel to be read, , each memory channel contains multiplexed data of one module.
Start:	First value to be read (On ISA boards read out starts with zero).
Len:	Number of values to be read.
BitsAnVal:	Number of Bits for one analog sample. Valid values are 12 or 14 bits.

Reads combined analog and digital input data from a specific channel and splits the data into the digital and analog part. The board must be initialized before. When using ISA boards, all installation parameters must be set before (address, installed memory).

Function SpcSetData: [ErrorCode] = SpcSetDa (Nr, MemCh, Start, Len, Data, Bits)

ErrorCode:	Errorcode of function as listed in the hardware manual.
Data:	Array of the values.to be written
Nr:	Number of board as defined by SpcInit or SpcInPCI.
MemCh:	Number of memory channel to be written, each memory channel contains multiplexed data for one module
Start:	First value to be written(On ISA boards writing starts with zero)
Len:	Number of values to be written.
Bits:	Number of Bits for one sample. Valid values are 8 for 8 bit boards, 16 for 12 bit and 16 bit boards.

Writes data for a specific channel. The board must be initialized before. When using ISA boards, all installation parameters must be set before (adress, installed memory).

Function SpcSetSplittedData: [ErrorCode] = SpcSetSplittedDa (Nr, MemCh, Start. Len, DigitalData, AnalogData, BitsAnVal)

ErrorCode:	ErrorCode of function as listed in the hardware manual.
Nr:	Number of board as defined by SpcInit or SpcInPCI.
MemCh:	Number of memory channel to be written, each memory channel contains multiplexed data for one module.
Start:	First value to be written (On ISA boards writing starts with zero).
Len:	Number of values to be written.
DigitalData:	Array of digital values to be written.
AnalogData:	Array of analog values to be written.
BitsAnVal:	Number of Bits for one analog sample. Valid value is 14 bits.

Writes combined analog and digital output data for a specific channel. The data is combined from separate analog and digital data arrays and written to the on-board memory. The board must be initialized before. When using ISA boards, all installation parameters must be set before (address, installed memory).