EPOS

Positioning Controller

Documentation

EPOS Command Library

maxon document number: rel3096

maxon motor ag Brünigstrasse 220 P.O. Box 263 CH-6072 Sachseln Tel.: 041/666 15 00 Fax.: 041/666 16 50 www.maxonmotor.com

TABLE OF CONTENTS

1 Introduction

The present documentation “EPOS Command Library” provides instructions on the implemented functions of the Windows dynamic link libraries “EposCmd.dll”,“EposCmd64.dll” and Linux 32-bit shared object library “libEposCmd.so”, which can be used for devices ***EPOS*** and ***EPOS2***.
The library is arranged in groups of functions and is intended to assist in programming of the control software based on Microsoft Windows© and 32-bit Linux platform.

This document describes the interface between a program and the “EPOS Command Libraries”. This libraries support devices of the EPOS family, which are attached to a **serial interface RS232**, **USB** on the Windows and Linux platform or to a **CAN board by IXXAT, Vector or National Instruments** available for the Windows systems. All other CANopen products of other manufacturers can also be used; however no motion control library is available.

Interface \ OS		32-Bit		64-Bit
		Windows	Linux	Windows
RS-232		X	X	X
USB		X	X	X
CAN Board	IXXAT	X	-	X
	NI	X	-	-
	Vector	X	-	X

Table 1: Supported platforms

The parameters of the 32-bit and 64-bit interfaces are the same.
The “EPOS Command Libraries” support the SDO protocol by CANopen but the “EPOS Command Libraries” are not suitable for real-time communication.

Additional information is available in the following documents:

- “EposCmd.Net.chm”
  The document “EposCmd.Net.chm” describes the implementation for **.NET** applications. Examples for **VB.Net** and **C#** are available.
- „DLL Integration into MS VC++”
  The document „DLL Integration into MS VC++” describes implementation and structure for **Microsoft Visual C++** in detail.
- „DLL Integration into MS Visual Basic”
  The document „DLL Integration into MS Visual Basic” describes implementation and structure for **Microsoft Visual Basic** in detail.
- „DLL Integration into Borland C++”
  The document „DLL Integration into Borland C++” describes implementation and structure for **Borland C++ Builder** in detail.
- „DLL Integration into Borland Delphi”
  The document „DLL Integration into Borland Delphi” describes implementation and structure for **Borland Delphi** in detail.
- „DLL Integration into LabVIEW”
  The document „DLL Integration into LabVIEW” describes implementation and structure for **National Instruments LabVIEW** in detail.
- „DLL Integration into National Instruments LabWindows/CVI”
  The document „DLL Integration into National Instruments LabWindows/CVI” describes implementation and structure for **National Instruments LabWindows/CVI** in detail.
- “Linux shared library Integration into Eclipse C++/QT”
  The document „ Linux shared library Integration into Eclipse C++/QT” describes implementation and structure for **Eclipse IDE for C/C++** in detail.

For a number of high-level languages an applicable example, including respective documentation, is available.

This library is intended to cover most applications used in automation.
It is based on the experience of maxon motor control. maxon motor control certifies that the content of this library is correct according to the best of their knowledge.

BECAUSE THIS LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THIS LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM „AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THIS LIBRARY IS WITH YOU. SHOULD THIS LIBRARY PROVE DEFECTIVE OR INSUFFICIENT, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION

The latest edition of these „EPOS Command Library", additional documentation and software to the EPOS positioning controller may also be found in the internet under www.maxonmotor.com category <Products> or direct from maxon motor catalog.

2 Third Party Products

Use one of the listed PC CANopen interface cards. For all of these manufacturers motion control libraries (example and documentation) are available.
All other CANopen products of other manufacturers can also be used; however no motion control library is available.

2.1 IXXAT

All IXXAT CANopen interfaces can be operated with the hardware-independent VCI driver V3 (Virtual CAN Interface). The older version VCI driver V2 (2.16 and higher) is still supported but should not be used because of lower performance.

See addresses below for ordering CANopen boards and further details of the driver.

Distributors

- www.ixxat.de

2.2 Vector

For use of Vector CANopen cards, the 'XL Driver Library' will be required. The installed edition of this library must be installed manually in the appropriate working directory (or system directory). With this library, it is possible to write own CANopen applications based on Vector's XL hardware.

See address below for ordering CANopen boards and further details of the driver.

Distributors

- www.vector-informatik.com

2.3 National Instruments

CAN Interfaces of National Instruments are supported. The NI-CAN software and hardware must be installed.

See address below for ordering CANopen boards and further details of the driver.

Contact

- www.ni.com/can

3 How to use this Guide

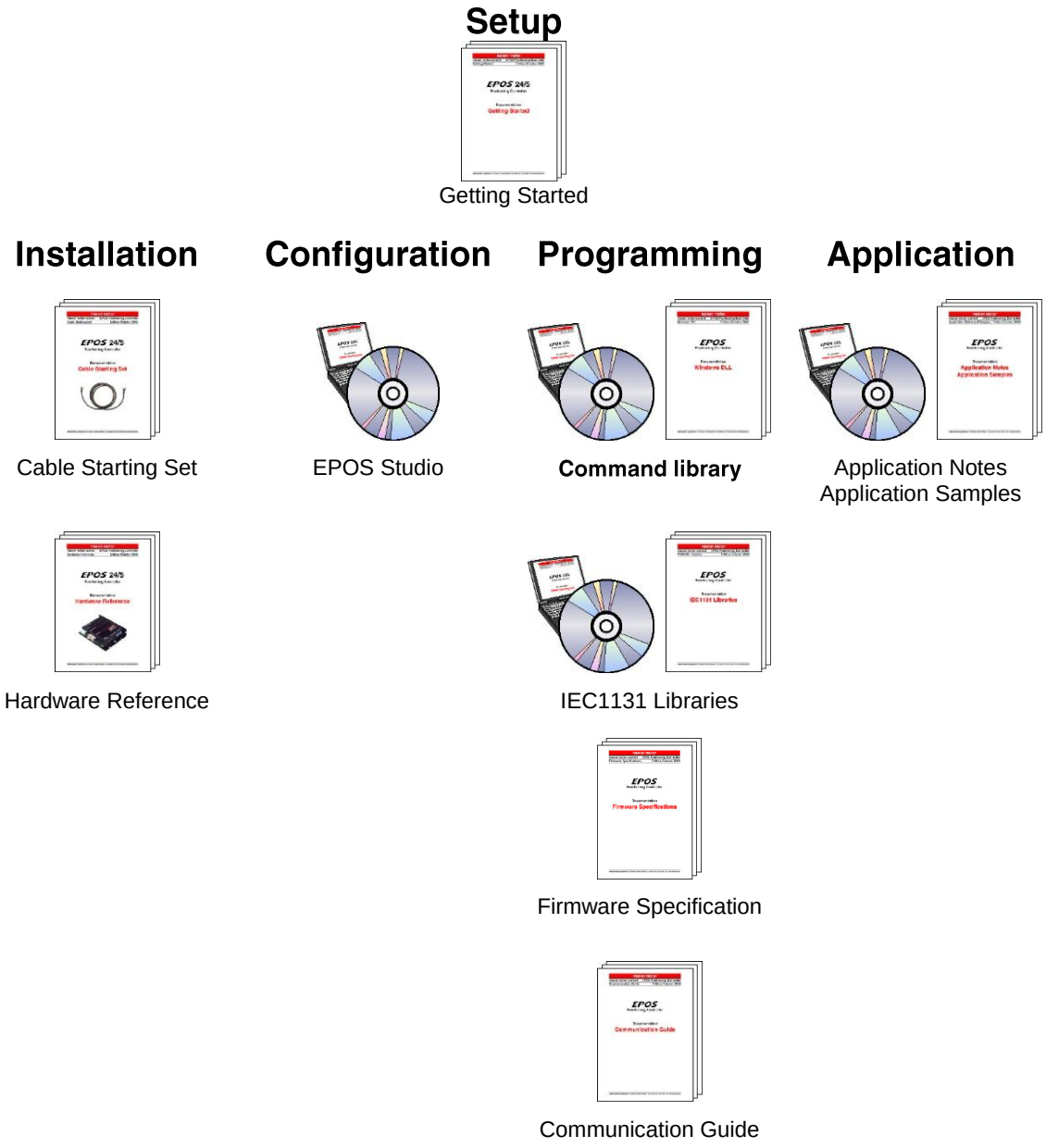

Figure 1: EPOS documentation hierarchy

3.1 General Information

- For detailed descriptions on used objects (Index-SubIndex), please refer to separate document ***Firmware Specification***.
- The units of measurement for the parameters are not being mentioned. They depend on the notation index (position, velocity and acceleration). For details please refer to separate document ***Firmware Specification***.

3.2 Communication Structure Windows

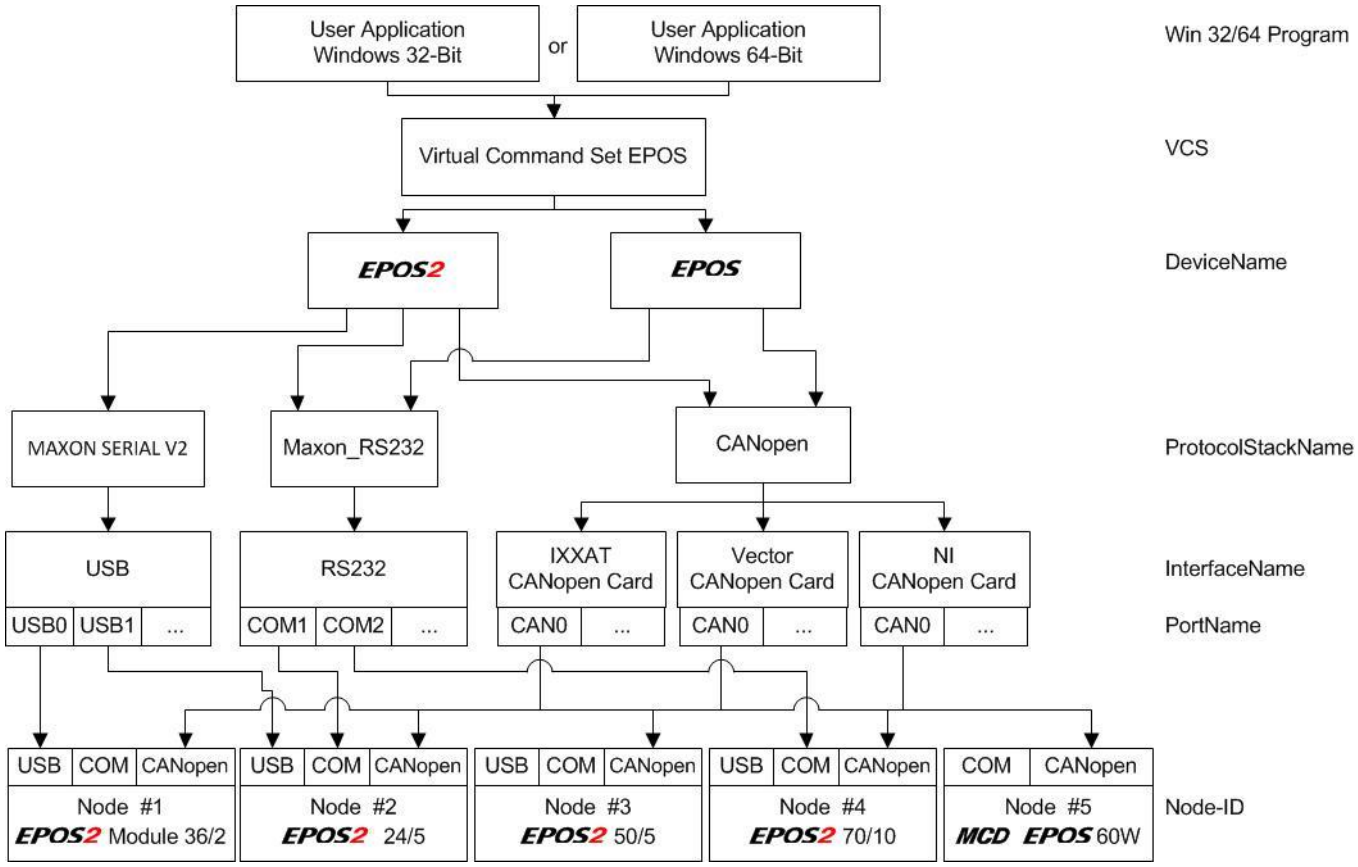

Figure 2: Example Windows communication structure

3.3 Communication Structure Linux

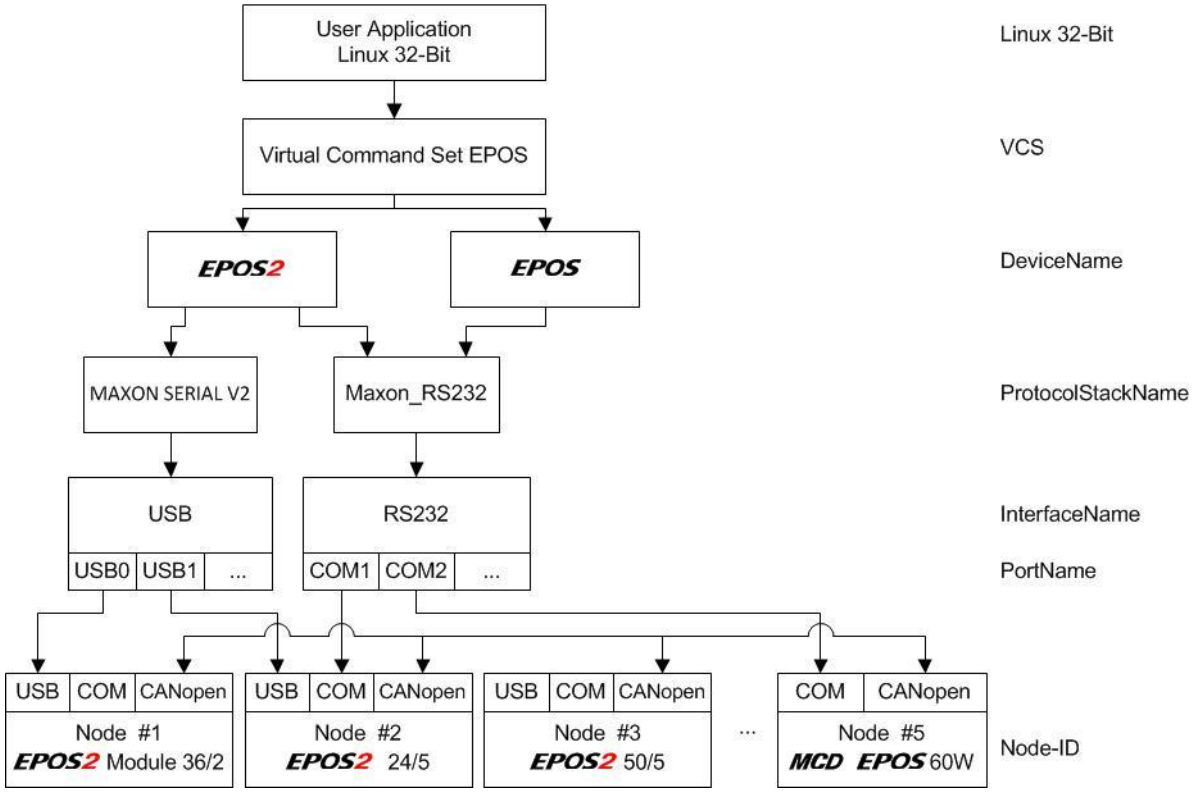

Figure 3: Example Linux communication structure

3.4 Data Type Definitions

Below is a table of all used data types.

Name	Data type	Size bits	Size bytes	Range	Comment
char, __int8	signed integer	8	1	– 128 ... 127	
BYTE	unsigned integer	8	1	0 ... 256	
short	signed integer	16	2	– 32'768 ... 32'767	
WORD	unsigned integer	16	2	0 ... 65'535	
long	signed integer	32	4	– 2'147'483'648 ... 2'147'483'647	
DWORD	unsigned integer	32	4	0 ... 4'294'967'295	
BOOL	signed integer	32	4	TRUE = 1 FALSE = 0	
HANDLE	pointer to an object	32	4	0 ... 4'294'967'295	OS dependent
		64	8	0 … 18'446'744'073'709'551'615	

Table 2: Data type definitions

4 Initialization Functions

4.1 Communication

4.1.1 Open Device

Function
HANDLE **VCS_OpenDevice** (char *DeviceName, char *ProtocolStackName, char *InterfaceName, char *PortName, DWORD *pErrorCode)

Description
Function „VCS_OpenDevice" opens the port for sending and receiving commands. This function opens interfaces with the RS232, the USB and with CANopen boards.
For correct designations on DeviceName, ProtocolStackName, InterfaceName and PortName use the functions Get Device Name Selection, Get Protocol Stack Name Selection, Get Interface Name Selection and Get Port Name Selection.

Parameters

DeviceName	char*	Name of connected device: EPOS, EPOS2
ProtocolStackName	char*	Name of used bus system: MAXON_RS232, MAXON SERIAL V2, CANopen
InterfaceName	char*	Name of used interface: MAXON_RS232: RS232 MAXON SERIAL V2: USB CANopen: Is composed of: Manufacturer_BoardName DeviceNumber *Examples:* IXXAT_USB-to-CAN compact 0, … Vector_CANcaseXL Channel 1, … NI_PCI-CAN 0, …
PortName	char*	Name of port: RS232: COM1, COM2, … CANopen: CAN0, CAN1, … USB: USB0, USB1, …

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	HANDLE	Handle for port access. Nonzero if successful; otherwise 0

4.1.2 Open Device Dialog

Function
HANDLE **VCS_OpenDeviceDlg** (DWORD *pErrorCode)

Description
The function „VCS_OpenDeviceDlg" registers available interfaces with which the EPOS can be operated and opens the selected interface for communication.

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	HANDLE	Handle for port access. Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

4.1.3 Get Protocol Stack Settings

Function
BOOL **VCS_GetProtocolStackSettings** (HANDLE KeyHandle, DWORD *pBaudrate, DWORD *pTimeout, DWORD *pErrorCode)

Description
Function „VCS_GetProtocolStackSettings" returns the communication parameters baud rate and timeout.

Parameter

KeyHandle	HANDLE	Handle for port access

Return Parameters

pBaudrate	DWORD*	Actual baud rate from opened port [Bit/s]
pTimeout	DWORD*	Actual timeout from opened port [ms]
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

4.1.4 Set Protocol Stack Settings

Function
BOOL **VCS_SetProtocolStackSettings** (HANDLE KeyHandle, DWORD Baudrate, DWORD Timeout, DWORD *pErrorCode)

Description
With function „VCS_SetProtocolStackSettings" it is possible to write the communication parameters. For exact values of available baud rates use the function Get Baudrate Selection.

Parameters

KeyHandle	HANDLE	Handle for port access
Baudrate	DWORD	Actual baud rate from opened port [Bit/s]
Timeout	DWORD	Actual timeout from opened port [ms]

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Remark
For correct communication use the same baud rate as on the connected device.

4.1.5 Find Device Communication Settings

Function
BOOL **VCS_FindDeviceCommunicationSettings** (HANDLE KeyHandle, char *pDeviceName, char *pProtocolStackName, char *pInterfaceName, char *pPortName, WORD SizeName, DWORD *pBaudrate, DWORD *pTimeout, WORD *pNodeId, int DialogMode, DWORD *pErrorCode)

Description
With function „VCS_FindDeviceCommunicationSettings" it is possible to search the communication setting parameters.

Remark
To accelerate the process some parameters can be defined.

Parameters

KeyHandle	HANDLE	Handle for port access
pDeviceName	char*	Device name
pProtocolStackName	char*	Protocol stack name
pInterfaceName	char*	Interface name
pPortName	char*	Port name
SizeName	WORD	Reserved memory size for return parameters
DialogMode	int	0: Show Progress Dialog 1: Show Progress and Confirmation Dialog 2: Show Confirmation Dialog

		3: Don't show any Dialog
Timeout	DWORD	Actual timeout from opened port [ms]

Return Parameters

pDeviceName	char*	Device name
pProtocolStackName	char*	Protocol stack name
pInterfaceName	char*	Interface name
pPortName	char*	Port name
pBaudrate	DWORD	Baud rate
pTimeout	DWORD	Timeout
pNodeId	WORD	Node ID
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

4.1.6 Close All Devices

Function
BOOL **VCS_CloseAllDevices** (DWORD *pErrorCode)

Description
Function „VCS_CloseAllDevices" closes all opened ports and releases it for other applications.

Return Parameter

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

4.1.7 Close Device

Function
BOOL **VCS_CloseDevice** (HANDLE KeyHandle, DWORD *pErrorCode)

Description
Function „VCS_CloseDevice" closes the port and releases it for other applications.

Parameters

KeyHandle	HANDLE	Handle for port access

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

4.2 Info

4.2.1 Get Error Info

Function
BOOL **VCS_GetErrorInfo** (DWORD ErrorCodeValue, char *pErrorInfo, WORD MaxStrSize)

Description
Function „VCS_GetErrorInfo" returns the error information about the executed function from a received error code. The function returns communication and library errors but no device errors descriptions.

Parameters

ErrorCodeValue	DWORD	Received error code
MaxStrSize	WORD	Max. length of error string

Return Parameters

pErrorInfo	char*	Error string
Return Value	BOOL	Nonzero if error information found; otherwise 0

4.2.2 Get Driver Info

Function
BOOL **VCS_GetDriverInfo** (char *pLibraryName, WORD MaxStrNameSize, char *pLibraryVersion, WORD MaxStrVersionSize, DWORD *pErrorCode)

Description
Function „VCS_GetDriverInfo" returns the name and version from the "EPOS Command DLL".

Parameters

MaxStrNameSize	WORD	Reserved memory size for the name
MaxStrVersionSize	WORD	Reserved memory size for the version

Return Parameters

pLibraryName	char*	Name from DLL
pLibraryVersion	char*	Version from DLL
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

4.2.3 Get Version

Function
BOOL **VCS_GetVersion** (HANDLE KeyHandle, WORD NodeId, WORD *pHardwareVersion, WORD *pSoftwareVersion, WORD *pApplicationNumber, WORD *pApplicationVersion, DWORD *pErrorCode)

Description
„VCS_GetVersion" returns the firmware version.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pHardwareVersion	WORD*	Hardware version	Object: 0x2003-01
pSoftwareVersion	WORD*	Software version	Object: 0x2003-02
pApplicationNumber	WORD*	Application number	Object: 0x2003-03
pApplicationVersion	WORD*	Application version	Object: 0x2003-04
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

4.3 Advanced Functions

This group defines all required functions to get additional information:

4.3.1 Get Device Name Selection

Function
BOOL **VCS_GetDeviceNameSelection** (BOOL StartOfSelection, char *pDeviceNameSel, WORD MaxStrSize, BOOL *pEndOfSelection, DWORD *pErrorCode)

Description
Function „VCS_GetDeviceNameSelection" returns all available device names.

Parameters

StartOfSelection	BOOL	True: Get first selection string False: Get next selection string
MaxStrSize	WORD	Reserved memory size for the device name

Return Parameters

pDeviceNameSel	char*	Device name
pEndOfSelection	BOOL*	1: No more selection string available 0: More string available
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

4.3.2 Get Protocol Stack Name Selection

Function

BOOL **VCS_GetProtocolStackNameSelection** (char *DeviceName, BOOL StartOfSelection, char *pProtocolStackNameSel, WORD MaxStrSize, BOOL *pEndOfSelection, DWORD *pErrorCode)

Description

Function „VCS_GetProtocolStackNameSelection" returns all available protocol stack names.

Parameters

DeviceName	char*	Device name
StartOfSelection	BOOL	1: Get first selection string 0: Get next selection string
MaxStrSize	WORD	Reserved memory size for the name

Return Parameters

pProtocolStackNameSel	char*	Pointer to available protocol stack name
pEndOfSelection	BOOL*	1: No more string available 0: More string available
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

4.3.3 Get Interface Name Selection

Function

BOOL **VCS_GetInterfaceNameSelection** (char *DeviceName, char *ProtocolStackName, BOOL StartOfSelection, char *pInterfaceNameSel, WORD MaxStrSize, BOOL *pEndOfSelection, DWORD *pErrorCode)

Description

Function „VCS_GetInterfaceNameSelection" returns all available interface names.

Parameters

DeviceName	char*	Device name
ProtocolStackName	char*	Protocol stack name
StartOfSelection	BOOL	1: Get first selection string 0: Get next selection string
MaxStrSize	WORD	Reserved memory size for the interface name

Return Parameters

pInterfaceNameSel	char*	Name from interface
pEndOfSelection	BOOL*	1: No more string available 0: More string available
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

4.3.4 Get Port Name Selection

Function

BOOL **VCS_GetPortNameSelection** (char *DeviceName, char *ProtocolStackName, char *InterfaceName, BOOL StartOfSelection, char *pPortSel, WORD MaxStrSize, BOOL *pEndOfSelection, DWORD *pErrorCode)

Description

Function „VCS_GetPortNameSelection" returns all available port names.

Parameters

DeviceName	char*	Device name
ProtocolStackName	char*	Protocol stack name
InterfaceName	char*	Interface name
StartOfSelection	BOOL	1: Get first selection string 0: Get next selection string
MaxStrSize	WORD	Reserved memory size for the port name

Return Parameters

pPortSel	char*	Pointer to port name
pEndOfSelection	BOOL*	1: No more string available 0: More string available
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

4.3.5 Get Baud Rate Selection

Function

BOOL **VCS_GetBaudrateSelection** (char *DeviceName, char *ProtocolStackName, char *InterfaceName, char *PortName, BOOL StartOfSelection, DWORD *pBaudrateSel, BOOL *pEndOfSelection, DWORD *pErrorCode)

Description

Function „VCS_GetBaudrateSelection" returns all available baud rates for the connected port.

Parameters

DeviceName	char*	Name of device
ProtocolStackName	char*	Name of protocol stack
InterfaceName	char*	Interface name
PortName	char*	Port name
StartOfSelection	BOOL	1: Get first selection value 0: Get next selection value

Return Parameters

pBaudrateSel	DWORD*	Pointer to baud rate [Bit/s]
pEndOfSelection	BOOL*	1: No more value available 0: More value available
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

4.3.6 Programming Example

The example shows how to read all protocol stack names of the available interfaces.
Programming language: C++

```
const WORD maxStrSize = 100;

char *strDeviceName = "EPOS";
char *strProtocolStackName[maxStrSize];
BOOL endOfSel;
DWORD errorCode;

//get first protocol stack name
if(VCS_GetProtocolStackNameSelection(strDeviceName, TRUE,
strProtocolStackName, maxStrSize, &endOfSel, &errorCode))
{
   //get next protocol stack name (as long as endOfSel == FALSE)
   while(!endOfSel)
   {
      VCS_GetProtocolStackNameSelection (strDeviceName, FALSE,
      strProtocolStackName, maxStrSize, &endOfSel, &errorCode);
   }
}
```

Note
The same example can be used for find out all device names, protocol stack names, interface names, port names and all available baud rates.
A combination of theses names is to open the communication interface (use function Open Device).

4.3.7 Get Key Handle

Function
BOOL **VCS_GetKeyHandle** (char *DeviceName, char *ProtocolStackName, char *InterfaceName, char *PortName, HANDLE *pKeyHandle, DWORD *pErrorCode)

Description
Function „VCS_GetKeyHandle" returns the key handle from the opened interface.

Parameters

DeviceName	char*	Device name
ProtocolStackName	char*	Protocol stack name
InterfaceName	char*	Interface name
PortName	char*	Port name

Return Parameters

pKeyHandle	HANDLE*	Handle for port access, if parameters are correct; otherwise 0
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

4.3.8 Get Device Name

Function
BOOL **VCS_GetDeviceName** (HANDLE KeyHandle, char *pDeviceName, WORD MaxStrSize, DWORD *pErrorCode)

Description
Function „VCS_GetDeviceName" returns the device name to corresponding handle.

Parameters

KeyHandle	HANDLE	Handle for port access
MaxStrSize	WORD	Reserved memory size for the device name

Return Parameters

pDeviceName	char*	Device name
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

4.3.9 Get Protocol Stack Name

Function
BOOL **VCS_GetProtocolStackName** (HANDLE KeyHandle, char *pProtocolStackName, WORD MaxStrSize, DWORD *pErrorCode)

Description
Function „VCS_GetProtocolStackName" returns the protocol stack name to corresponding handle.

Parameters

KeyHandle	HANDLE	Handle for port access
MaxStrSize	WORD	Reserved memory size for the name

Return Parameters

pProtocolStackName	char*	Pointer to protocol stack name
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

4.3.10 Get Interface Name

Function
BOOL **VCS_GetInterfaceName** (HANDLE KeyHandle, char *pInterfaceName, WORD MaxStrSize, DWORD *pErrorCode)

Description
Function „VCS_GetInterfaceName" returns the interface name to corresponding handle.

Parameters

KeyHandle	HANDLE	Handle for port access
MaxStrSize	WORD	Reserved memory size for the interface name

Return Parameters

pInterfaceName	char*	Name from interface
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

4.3.11 Get Port Name

Function
BOOL **VCS_GetPortName** (HANDLE KeyHandle, char *pPortName, WORD MaxStrSize, DWORD *pErrorCode)

Description
Function „VCS_GetPortName" returns the port name to corresponding handle.

Parameters

KeyHandle	HANDLE	Handle for port access
MaxStrSize	WORD	Reserved memory size for the port name

Return Parameters

pPortName	char*	Port name
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5 Configuration Functions

For detailed information about the objects refer to associated document *Firmware Specification*.

5.1 General

5.1.1 Import Parameter

Function
BOOL **VCS_ImportParameter** (HANDLE KeyHandle, WORD NodeId, char *pParameterFileName, BOOL ShowDlg, BOOL ShowMsg, DWORD *pErrorCode)

Description
„VCS_ImportParameter" writes parameters from a file to the device.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
pParameterFileName	char*	Path of the needed file
ShowDlg	BOOL	Dialog is shown
ShowMsg	BOOL	Message box are activated

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

5.1.2 Export Parameter

Function
BOOL **VCS_ExportParameter** (HANDLE KeyHandle, WORD NodeId, char *pParameterFileName, char *pBinaryFile, char *pUserID, char *pComment, BOOL ShowDlg, BOOL ShowMsg, DWORD *pErrorCode)

Description
„VCS_ExportParameter" reads all parameters of the device and writes this into the file.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
pParameterFileName	char*	Path of the needed file
pBinaryFile	char*	Firmware file of the connected device
pUserID	char*	User name
pComment	char*	Comment
ShowDlg	BOOL	Dialog is shown
ShowMsg	BOOL	Message box are activated

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

5.1.3 Set Object

Function
BOOL **VCS_SetObject** (HANDLE KeyHandle, WORD NodeId, WORD ObjectIndex, BYTE ObjectSubIndex, void *pData, DWORD NbOfBytesToWrite, DWORD *pNbOfBytesWritten, DWORD *pErrorCode)

Description
The function „VCS_SetObject" writes an object value at the given index and sub-index from object dictionary.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
ObjectIndex	WORD	Object index
ObjectSubIndex	BYTE	Object sub-index
pData	void*	Object data
NbOfBytesToWrite	DWORD	Object length to write (number of bytes)

Return Parameters

pNbOfBytesWritten	DWORD*	Object length written (number of bytes)
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Remarks
All information about object index, object sub-index and object length can be checked in the associated document *Firmware Specification*.

5.1.4 Get Object

Function
BOOL **VCS_GetObject** (HANDLE KeyHandle, WORD NodeId, WORD ObjectIndex, BYTE ObjectSubIndex, void *pData, DWORD NbOfBytesToRead, DWORD *pNbOfBytesRead, DWORD *pErrorCode)

Description
The function „VCS_GetObject" reads an object value at the given index and sub-index from object dictionary.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
ObjectIndex	WORD	Object index
ObjectSubIndex	BYTE	Object sub-index
NbOfBytesToRead	DWORD	Object length to read (number of bytes)

Return Parameters

pData	void*	Object data
pNbOfBytesRead	DWORD*	Object length read (number of bytes)
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Remarks
All information about object index, object sub-index and object length can be checked in the associated document *Firmware Specification*.

5.1.5 Restore

Function
BOOL **VCS_Restore** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
The function „VCS_Restore" restores all default parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.1.6 Store

Function
BOOL **VCS_Store** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
The function „VCS_Store" stores all parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2 Advanced Functions

5.2.1 Motor

5.2.1.1 Set Motor Type

Function
BOOL **VCS_SetMotorType**(HANDLE KeyHandle, WORD NodeId, WORD MotorType, DWORD *pErrorCode)

Description
With function „VCS_SetMotorType" the motor type is written.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
MotorType	WORD	Kind of motor (see Table 3)	Object: 0x6402-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Kind of motor

Description	Constant name	Value
brushed DC motor	MT_DC_MOTOR	1
EC motor sinus commutated	MT_EC_SINUS_COMMUTATED_MOTOR	10
EC motor block commutated	MT_EC_BLOCK_COMMUTATED_MOTOR	11

Table 3: Kind of motor

5.2.1.2 Set DC-Motor Parameter

Function
BOOL **VCS_SetDcMotorParameter** (HANDLE KeyHandle, WORD NodeId, WORD NominalCurrent, WORD MaxOutputCurrent, WORD ThermalTimeConstant, DWORD *pErrorCode)

Description
With function „VCS_SetDcMotorParameter" it is possible to write all DC motor parameters.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
NominalCurrent	WORD	Maximal continuous current	Object: 0x6410-01
MaxOutputCurrent	WORD	Maximal peak current	Object: 0x6410-02
ThermalTimeConstant	WORD	Thermal time constant winding	Object: 0x6410-05

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.1.3 Set EC-Motor Parameter

Function
BOOL **VCS_SetEcMotorParameter** (HANDLE KeyHandle, WORD NodeId, WORD NominalCurrent, WORD MaxOutputCurrent, WORD ThermalTimeConstant, BYTE NbOfPolePairs, DWORD *pErrorCode)

Description
With function „VCS_SetEcMotorParameter" it is possible to write all EC motor parameters.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
NominalCurrent	WORD	Maximal continuous current	Object: 0x6410-01
MaxOutputCurrent	WORD	Maximal peak current	Object: 0x6410-02
ThermalTimeConstant	WORD	Thermal time constant winding	Object: 0x6410-05
NbOfPolePairs	BYTE	Number of pole pairs	Object: 0x6410-03

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.1.4 Get Motor Type

Function
BOOL **VCS_GetMotorType** (HANDLE KeyHandle, WORD NodeId, WORD *pMotorType, DWORD *pErrorCode)

Description
With function „VCS_GetMotorType" it is possible to read the motor type.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

MotorType	WORD	Kind of motor (see Table 3)	Object: 0x6402-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.1.5 Get DC-Motor Parameter

Function

BOOL **VCS_GetDcMotorParameter** (HANDLE KeyHandle, WORD NodeId, WORD* pNominalCurrent, WORD* pMaxOutputCurrent, WORD *pThermalTimeConstant, DWORD *pErrorCode)

Description

With function „VCS_GetDcMotorParameter" it is possible to read all DC motor parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pNominalCurrent	WORD*	Maximal continuous current	Object: 0x6410-01
pMaxOutputCurrent	WORD*	Maximal peak current	Object: 0x6410-02
pThermalTimeConstant	WORD*	Thermal time constant winding	Object: 0x6410-05
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.1.6 Get EC-Motor Parameter

Function

BOOL **VCS_GetEcMotorParameter** (HANDLE KeyHandle, WORD NodeId, WORD* pNominalCurrent, WORD* pMaxOutputCurrent, WORD* pThermalTimeConstant, BYTE* pNbOfPolePairs, DWORD *pErrorCode)

Description

With function „VCS_GetEcMotorParameter" it is possible to read all EC motor parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pNominalCurrent	WORD*	Maximal continuous current	Object: 0x6410-01
pMaxOutputCurrent	WORD*	Maximal peak current	Object: 0x6410-02
pThermalTimeConstant	WORD*	Thermal time constant winding	Object: 0x6410-05
pNbOfPolePairs	BYTE*	Number of pole pairs	Object: 0x6410-03
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.2 Sensor

5.2.2.1 Set Sensor Type

Function
BOOL **VCS_SetSensorType** (HANDLE KeyHandle, WORD NodeId, WORD Counts, WORD SensorType, DWORD *pErrorCode)

Description
With function „VCS_SetSensorType" it is possible to write the sensor type.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
SensorType	WORD	Position Sensor Type (see Table 4)	Object: 0x2210-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Position Sensor Type

Description	Constant name	Value
Unknown sensor (undefined)	ST_UNKNOWN	0
Incremental Encoder 1 with index (3-channel)	ST_INC_ENCODER_3CHANNEL	1
Incremental Encoder 1 without index (2-channel)	ST_INC_ENCODER_2CHANNEL	2
Hall Sensors	ST_HALL_SENSORS	3
SSI Encoder binary coded	ST_SSI_ABS_ENCODER_BINARY	4
SSI Encoder Grey coded	ST_SSI_ABS_ENCODER_GREY	5

Table 4: Position Sensor Type

5.2.2.2 Set Incremental Encoder Parameter

Function
BOOL **VCS_SetIncEncoderParameter** (HANDLE KeyHandle, WORD NodeId, DWORD EncoderResolution, BOOL InvertedPolarity, DWORD *pErrorCode)

Description
With function „VCS_SetIncEncoderParameter" it is possible to write the incremental encoder parameters.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
EncoderResolution	DWORD	Encoder pulse number [pulse per turn]	Object: 0x2210-01
InvertedPolarity	BOOL	Position sensor polarity	Object: 0x2210-04

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.2.3 Set Hall Sensor Parameter

Function
BOOL **VCS_SetHallSensorParameter** (HANDLE KeyHandle, WORD NodeId, BOOL InvertedPolarity, DWORD *pErrorCode)

Description
With function „VCS_SetHallSensorParameter" it is possible to write the Hall sensor parameter.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
InvertedPolarity	BOOL	Position sensor polarity	Object: 0x2210-04

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.2.4 Set SSI Absolute Encoder Parameter

Function
BOOL **VCS_SetSsiAbsEncoderParameter** (HANDLE KeyHandle, WORD NodeId, WORD DataRate, WORD NbOfMultiTurnDataBits, WORD NbOfSingleTurnDataBits, BOOL InvertedPolarity, DWORD *pErrorCode)

Description
With function „VCS_SetSsiAbsEncoderParameter" it is possible to write all parameters for SSI absolute encoder.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
DataRate	WORD	SSI encoder data rate	Object: 0x2211-01
NbOfMultiTurnDataBits	WORD	number of bits multi turn	Object: 0x2211-02
NbOfSingleTurnDataBits	WORD	number of bits single turn	
InvertedPolarity	BOOL	Position sensor polarity	Object: 0x2210-04

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.2.5 Get Sensor Type

Function
BOOL **VCS_GetSensorType** (HANDLE KeyHandle, WORD NodeId, WORD *pSensorType, DWORD *pErrorCode)

Description
With function „VCS_GetSensorType" it is possible to read the sensor type.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pSensorType	WORD*	Position sensor type (see Table 4)	Object: 0x2210-02
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.2.6 Get Incremental Encoder Parameter

Function

BOOL **VCS_GetIncEncoderParameter** (HANDLE KeyHandle, WORD NodeId, DWORD* pEncoderResolution, BOOL* pInvertedPolarity, DWORD *pErrorCode)

Description

With function „VCS_GetIncEncoderParameter" it is possible to read the incremental encoder parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pEncoderResolution	DWORD	Encoder pulse number [pulse per turn]	Object: 0x2210-01
pInvertedPolarity	BOOL	Position sensor polarity	Object: 0x2210-04
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.2.7 Get Hall Sensor Parameter

Function

BOOL **VCS_GetHallSensorParameter** (HANDLE KeyHandle, WORD NodeId, BOOL* pInvertedPolarity, DWORD *pErrorCode)

Description

With function „VCS_GetHallSensorParameter" it is possible to read the Hall sensor parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pInvertedPolarity	BOOL	Position sensor polarity	Object: 0x2210-04
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.2.8 Get SSI Absolute Encoder Parameter

Function

BOOL **VCS_GetSsiAbsEncoderParameter** (HANDLE KeyHandle, WORD NodeId, WORD* pDataRate, WORD* pNbOfMultiTurnDataBits, WORD* pNbOfSingleTurnDataBits, BOOL* pInvertedPolarity, DWORD* pErrorCode)

Description

With function „VCS_GetSsiAbsEncoderParameter" it is possible to read all parameters from SSI absolute encoder.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pDataRate	WORD*	SSI encoder data rate	Object: 0x2211-01
pNbOfMultiTurnDataBits	WORD*	Number of bits multi turn	Object: 0x2211-02
pNbOfSingleTurnDataBits	WORD*	Number of bits single turn	
pInvertedPolarity	BOOL*	Position sensor polarity	Object: 0x2210-04
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.3 Safety

5.2.3.1 Set Maximal Following Error

Function
BOOL **VCS_SetMaxFollowingError** (HANDLE KeyHandle, WORD NodeId, DWORD MaxFollowingError, DWORD *pErrorCode)

Description
„VCS_SetMaxFollowingError" writes the maximal allowed following error parameter.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
MaxFollowingError	DWORD	Maximal allowed difference of position actual value to position demand value.	Object: 0x6065-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.3.2 Get Maximal Following Error

Function
BOOL **VCS_GetMaxFollowingError** (HANDLE KeyHandle, WORD NodeId, DWORD* pMaxFollowingError, DWORD *pErrorCode)

Description
„VCS_GetMaxFollowingError" reads the maximal allowed following error parameter.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pMaxFollowingError	DWORD*	Maximal allowed difference of position actual value to position demand value.	Object: 0x6065-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.3.3 Set Maximal Profile Velocity

Function
BOOL **VCS_SetMaxProfileVelocity** (HANDLE KeyHandle, WORD NodeId, DWORD MaxProfileVelocity, DWORD *pErrorCode)

Description
„VCS_SetMaxProfileVelocity" writes the maximal allowed velocity.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
MaxProfileVelocity	DWORD	This value is used as velocity limit in a position (or velocity) move.	Object: 0x607F-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.3.4 Get Maximal Profile Velocity

Function
BOOL **VCS_GetMaxProfileVelocity** (HANDLE KeyHandle, WORD NodeId, DWORD* pMaxProfileVelocity, DWORD *pErrorCode)

Description
„VCS_GetMaxProfileVelocity" reads the maximal allowed velocity.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	

Return Parameters

pMaxProfileVelocity	DWORD*	This value is used as velocity limit in a position (or velocity) move.	Object: 0x607F-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.3.5 Set Maximal Acceleration

Function
BOOL **VCS_SetMaxAcceleration** (HANDLE KeyHandle, WORD NodeId, DWORD MaxAcceleration, DWORD *pErrorCode)

Description
„VCS_SetMaxAcceleration" writes the maximal allowed acceleration/deceleration.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
MaxAcceleration	DWORD	This value is the limit of the other acceleration/ deceleration objects.	Object: 0x60C5-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.3.6 Get Maximal Acceleration

Function
BOOL **VCS_GetMaxAcceleration** (HANDLE KeyHandle, WORD NodeId, DWORD* pMaxAcceleration, DWORD *pErrorCode)

Description
„VCS_GetMaxAcceleration" reads the maximal allowed acceleration/deceleration.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	

Return Parameters

pMaxAcceleration	DWORD*	This value is the limit of the other acceleration/deceleration objects.	Object: 0x60C5-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.4 Position Regulator

5.2.4.1 Set Position Regulator Gain

Function
BOOL **VCS_SetPositionRegulatorGain** (HANDLE KeyHandle, WORD NodeId, WORD P, WORD I, WORD D, DWORD *pErrorCode)

Description
With function „VCS_SetPositionRegulatorGain" it is possible to write all position regulator gains. Determine the optimal parameters by using the 'Regulation Tuning' of EPOS Studio.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
P	WORD	Position regulator P-Gain	Object: 0x60FB-01
I	WORD	Position regulator I-Gain	Object: 0x60FB-02
D	WORD	Position regulator D-Gain	Object: 0x60FB-03

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.4.2 Set Position Regulator Feed Forward

Function
BOOL **VCS_SetPositionRegulatorFeedForward** (HANDLE KeyHandle, WORD NodeId, WORD VelocityFeedForward, WORD AccelerationFeedForward, DWORD *pErrorCode)

Description
„VCS_SetPositionRegulatorFeedForward" writes parameters for position regulation with feed forward.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
VelocityFeedForward	WORD	Velocity feed forward factor	Object: 0x60FB-04
AccelerationFeedForward	WORD	Acceleration feed forward factor	Object: 0x60FB-05

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.4.3 Get Position Regulator Gain

Function

BOOL **VCS_GetPositionRegulatorGain** (HANDLE KeyHandle, WORD NodeId, WORD *pP, WORD *pI, WORD *pD, DWORD *pErrorCode)

Description

With function „VCS_GetPositionRegulatorGain" it is possible to read all position regulator gains.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pP	WORD*	Position regulator P-Gain	Object: 0x60FB-01
pI	WORD*	Position regulator I-Gain	Object: 0x60FB-02
pD	WORD*	Position regulator D-Gain	Object: 0x60FB-03
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.4.4 Get Position Regulator Feed Forward

Function

BOOL **VCS_GetPositionRegulatorFeedForward** (HANDLE KeyHandle, WORD NodeId, WORD* pVelocityFeedForward, WORD* pAccelerationFeedForward, DWORD *pErrorCode)

Description

„VCS_GetPositionRegulatorFeedForward" reads parameters for position regulation feed forward.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pVelocityFeedForward	WORD*	Velocity feed forward factor	Object: 0x60FB-04
pAccelerationFeedForward	WORD*	Acceleration feed forward factor	Object: 0x60FB-05
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.5 Velocity Regulator

5.2.5.1 Set Velocity Regulator Gain

Function
BOOL **VCS_SetVelocityRegulatorGain** (HANDLE KeyHandle, WORD NodeId, WORD P, WORD I, DWORD *pErrorCode)

Description
With function „VCS_SetVelocityRegulatorGain" it is possible to write all velocity regulator gains. Determine the optimal parameters by using the 'Regulation Tuning' of EPOS Studio.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
P	WORD	Velocity regulator P-Gain	Object: 0x60F9-01
I	WORD	Velocity regulator I-Gain	Object: 0x60F9-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.5.2 Set Velocity Regulator Feed Forward

Function
BOOL **VCS_SetVelocityRegulatorFeedForward** (HANDLE KeyHandle, WORD NodeId, WORD VelocityFeedForward, WORD AccelerationFeedForward, DWORD *pErrorCode)

Description
„VCS_SetVelocityRegulatorFeedForward" writes parameters for velocity regulation with feed forward.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
VelocityFeedForward	WORD	Velocity feed forward factor	Object: 0x60F9-04
AccelerationFeedForward	WORD	Acceleration feed forward factor	Object: 0x60F9-05

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.5.3 Get Velocity Regulator Gain

Function
BOOL **VCS_GetVelocityRegulatorGain** (HANDLE KeyHandle, WORD NodeId, WORD *pP, WORD *pI, DWORD *pErrorCode)

Description
With function „VCS_GetVelocityRegulatorGain" it is possible to read all velocity regulator gains.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pP	WORD*	Velocity regulator P-Gain	Object: 0x60F9-01

pI	WORD*	Velocity regulator I-Gain	Object: 0x60F9-02
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.5.4 Get Velocity Regulator Feed Forward

Function
BOOL **VCS_GetVelocityRegulatorFeedForward** (HANDLE KeyHandle, WORD NodeId, WORD* pVelocityFeedForward, WORD* pAccelerationFeedForward, DWORD *pErrorCode)

Description
„VCS_GetVelocityRegulatorFeedForward" reads parameters for velocity regulation feed forward.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pVelocityFeedForward	WORD*	Velocity feed forward factor	Object: 0x60F9-04
pAccelerationFeedForward	WORD*	Acceleration feed forward factor	Object: 0x60F9-05
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.6 Current Regulator

5.2.6.1 Set Current Regulator Gain

Function
BOOL **VCS_SetCurrentRegulatorGain** (HANDLE KeyHandle, WORD NodeId, WORD P, WORD I, DWORD *pErrorCode)

Description
With function „VCS_SetCurrentRegulatorGain" it is possible to write all current regulator gains. Determine the optimal parameters by using the 'Regulation Tuning' of EPOS Studio.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
P	WORD	Current regulator P-Gain	Object: 0x60F6-01
I	WORD	Current regulator I-Gain	Object: 0x60F6-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.6.2 Get Current Regulator Gain

Function
BOOL **VCS_GetCurrentRegulatorGain** (HANDLE KeyHandle, WORD NodeId, WORD *pP, WORD *pI, DWORD *pErrorCode)

Description
With function „VCS_GetCurrentRegulatorGain" it is possible to read all current regulator gains.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pP	WORD*	Current regulator P-Gain	Object: 0x60F6-01
pI	WORD*	Current regulator I-Gain	Object: 0x60F6-02
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

5.2.7 Inputs/Outputs

5.2.7.1 Digital Input Configuration

Function

BOOL **VCS_DigitalInputConfiguration** (HANDLE KeyHandle, WORD NodeId, WORD DigitalInputNb, WORD Configuration, BOOL Mask, BOOL Polarity, BOOL ExecutionMask, DWORD *pErrorCode)

Description

„VCS_DigitalInputConfiguration" sets the parameter for one digital input.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
DigitalInputNb	WORD	Number of digital input (Sub-index of object)	Object: 0x2070-0x
Configuration	WORD	Configures which functionality will be assigned to digital input (bit number) For values see Table 5.	
Mask	BOOL	1: Functionality state will be displayed 0: not displayed	Object: 0x2071-02
Polarity	BOOL	1: Low active 0: High active	Object: 0x2071-03
ExecutionMask	BOOL	1: Set the error routine. Only for positive and negative switch.	Object: 0x2071-04

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Digital Input Configuration

Description	Constant name	Value
General purpose A	DIC_GENERAL_PURPOSE_A	15
General purpose B	DIC_GENERAL_PURPOSE_B	14
General purpose C	DIC_GENERAL_PURPOSE_C	13
General purpose D	DIC_GENERAL_PURPOSE_D	12
General purpose E	DIC_GENERAL_PURPOSE_E	11
General purpose F	DIC_GENERAL_PURPOSE_F	10
General purpose G	DIC_GENERAL_PURPOSE_G	9
General purpose H	DIC_GENERAL_PURPOSE_H	8
General purpose I	DIC_GENERAL_PURPOSE_I	7
General purpose J	DIC_GENERAL_PURPOSE_J	6
Quick stop	DIC_QUICK_STOP	5
Device enable	DIC_DRIVE_ENABLE	4
Position marker	DIC_POSITION_MARKER	3
Home switch	DIC_HOME_SWITCH	2
Positive limit switch	DIC_POSITIVE_LIMIT_SWITCH	1
Negative limit switch	DIC_NEGATIVE_LIMIT_SWITCH	0

Table 5: Digital Input Configuration

5.2.7.2 Digital Output Configuration

Function

BOOL **VCS_DigitalOutputConfiguration** (HANDLE KeyHandle, WORD NodeId, WORD DigitalOutputNb, WORD Configuration, BOOL State, BOOL Mask, BOOL Polarity, DWORD *pErrorCode)

Description

„VCS_DigitalOutputConfiguration" sets parameter for one digital output.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
DigitalOutputNb	WORD	Number of digital output (Sub-index of object)	Object: 0x2079-0x
Configuration	WORD	Configures which functionality will be assigned to digital output (bit number). For values see Table 6.	
State	BOOL	State of digital output	Object: 0x2078-01
Mask	BOOL	1: Register will be modified	Object: 0x2078-02
Polarity	BOOL	1: Output will be inverted	Object: 0x2078-03

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Digital Output Configuration

Description	Constant name	Value
General purpose A	DOC_GENERAL_PURPOSE_A	15
General purpose B	DOC_GENERAL_PURPOSE_B	14
General purpose C	DOC_GENERAL_PURPOSE_C	13
General purpose D	DOC_GENERAL_PURPOSE_D	12
General purpose E	DOC_GENERAL_PURPOSE_E	11
Position compare	DOC_POSITION_COMPARE	1
Ready / Fault	DOC_READY_FAULT	0

Table 6: Digital Output Configuration

5.2.7.3 Analog Input Configuration

Function

BOOL **VCS_AnalogInputConfiguration** (HANDLE KeyHandle, WORD NodeId, WORD AnalogInputNb, WORD Configuration, BOOL ExecutionMask, DWORD *pErrorCode)

Description

„VCS_AnalogInputConfiguration" sets parameter for one analog input.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
AnalogInputNb	WORD	Number of analog input (Sub-index of object)	Object: 0x2079
Configuration	WORD	Configures which functionality will be assigned to analog input (bit number) For values see Table 7.	
ExecutionMask	BOOL	1: Register will be modified	Object: 0x2078-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Analog Input Configuration

Description	Constant name	Value
Analog position set point	AIC_ANALOG_POSITION_SETPOINT	2
Analog velocity set point	AIC_ANALOG_VELOCITY_SETPOINT	1
Analog current set point	AIC_ANALOG_CURRENT_SETPOINT	0

Table 7: Analog Input Configuration

5.2.8 Units

5.2.8.1 Set Velocity Units

Function

BOOL **VCS_SetVelocityUnits** (HANDLE KeyHandle, WORD NodeId, BYTE VelDimension, char VelNotation, DWORD *pErrorCode)

Description

„VCS_SetVelocityUnits" writes velocity unit parameters.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
VelDimension	BYTE	Velocity dimension index VD_RPM = 0xA4	Object: 0x608C-00
VelNotation	char	Velocity notation index VN_STANDARD = 0 VN_DECI = -1 VN_CENTI = -2 VN_MILLI = -3	Object: 0x608B-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

5.2.8.2 Get Velocity Units

Function

BOOL **VCS_GetVelocityUnits** (HANDLE KeyHandle, WORD NodeId, BYTE* pVelDimension, char* pVelNotation, DWORD *pErrorCode)

Description

„VCS_GetVelocityUnits" reads velocity unit parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pVelDimension	BYTE*	Velocity dimension index VD_RPM = 0xA4	Object: 0x608C-00
pVelNotation	char*	Velocity notation index VN_STANDARD = 0; VN_DECI = -1; VN_CENTI = -2; VN_MILLI = -3	Object: 0x608B-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6 Operation Functions

6.1 Operation Mode

6.1.1 Set Operation Mode

Function
BOOL **VCS_SetOperationMode** (HANDLE KeyHandle, WORD NodeId, __int8 Mode, DWORD *pErrorCode)

Description
„VCS_SetOperationMode" sets the operation mode.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
Mode	__int8	Operation mode (see Table 8)	Object: 0x6060-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.1.2 Get Operation Mode

Function
BOOL **VCS_GetOperationMode** (HANDLE KeyHandle, WORD NodeId, __int8 *pMode, DWORD *pErrorCode)

Description
Function „VCS_GetOperationMode" returns the activated operation mode.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pMode	__int8*	Operation mode (see Table 8	Object: 0x6061-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Operational Modes

Description	Value	Constant name
Position Profile Mode	1	OMD_PROFILE_POSITION_MODE
Position Velocity Mode	3	OMD_PROFILE_VELOCITY_MODE
Homing Mode	6	OMD_HOMING_MODE
Interpolated Position Mode	7	OMD_INTERPOLATED_POSITION_MODE
Position Mode	-1	OMD_POSITION_MODE
Velocity Mode	-2	OMD_VELOCITY_MODE
Current Mode	-3	OMD_CURRENT_MODE
Master Encoder Mode	-5	OMD_MASTER_ENCODER_MODE
Step Direction Mode	-6	OMD_STEP_DIRECTION_MODE

Table 8: Operational modes

6.2 State Machine

For detailed information about the state machine refer to associated document *Firmware Specification*.

6.2.1 Reset Device

Function
BOOL **VCS_ResetDevice** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
The function „VCS_ResetDevice" is used to send the NMT service 'Reset Node'. It is a command without acknowledge.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.2.2 Set State

Function
BOOL **VCS_SetState** (HANDLE KeyHandle, WORD NodeId, WORD State, DWORD *pErrorCode)

Description
„VCS_SetState" reads the actual state machine state.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
State	WORD	Value of state machine (see Table 9)	Object: 0x6040-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

States

Description	Value	Constant name
Get/Set Disable State	0x0000	ST_DISABLED
Get/Set Enable State	0x0001	ST_ENABLED
Get/Set Quickstop State	0x0002	ST_QUICKSTOP
Get Fault State	0x0003	ST_FAULT

Table 9: State modes

6.2.3 Set Enable State

Function
BOOL **VCS_SetEnableState** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_SetEnableState" the device changes to enable state.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.2.4 Set Disable State

Function
BOOL **VCS_SetDisableState** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With function „VCS_SetDisableState" changes the device to disable state.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.2.5 Set Quick Stop State

Function
BOOL **VCS_SetQuickStopState** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With function „VCS_SetQuickStopState" the device changes to quick stop state.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.2.6 Clear Fault

Function
BOOL **VCS_ClearFault** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With function „VCS_ClearFault" the device changes from fault state to disable state.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.2.7 Get State

Function
BOOL **VCS_GetState**(HANDLE KeyHandle, WORD NodeId, WORD* pState, DWORD *pErrorCode)

Description
„VCS_GetState" reads the new state for the state machine.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pState	WORD*	Control word value (see Table 9)	Object: 0x6040-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.2.8 Get Enable State

Function
BOOL **VCS_GetEnableState**(HANDLE KeyHandle, WORD NodeId, BOOL *pIsEnabled, DWORD *pErrorCode)

Description
„VCS_GetEnableState" checks if the device is enabled.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pIsEnabled	BOOL*	1: Device enabled 0: Device not enabled
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.2.9 Get Disable State

Function
BOOL **VCS_GetDisableState**(HANDLE KeyHandle, WORD NodeId, BOOL *pIsDisabled, DWORD *pErrorCode)

Description
„VCS_GetDisableState" checks if the device is disabled.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pIsDisabled	BOOL*	1: Device disabled 0: Device not disabled
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.2.10 Get Quick Stop State

Function
BOOL **VCS_GetQuickStopState** (HANDLE KeyHandle, WORD NodeId, BOOL *pIsQuickStopped, DWORD *pErrorCode)

Description
The function „VCS_GetQuickStopState" returns the device state quick stop.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pIsQuickStopped	BOOL*	1: Device is in quick stop state 0: Device is not in quick stop state
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.2.11 Get Fault State

Function
BOOL **VCS_GetFaultState** (HANDLE KeyHandle, WORD NodeId, BOOL *pIsInFault, DWORD *pErrorCode)

Description
The function „VCS_GetFaultState" returns the device state fault (pIsInFault = TRUE).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pIsInFault	BOOL*	1: Device is in fault state 0: Device is not in fault state
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Remarks
Get error information if the device is in fault state see chapter Error Handling.

6.3 Error Handling

6.3.1 Get Number of Device Error

Function
BOOL **VCS_GetNbOfDeviceError** (HANDLE KeyHandle, WORD NodeId, BYTE *pNbDeviceError, DWORD *pErrorCode)

Description
„VCS_GetNbOfDeviceError" returns the number of actual errors that are recorded.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pNbDeviceError	BYTE*	Number of occurred device errors	Object: 0x1003-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Programming Example

6.3.2 Get Device Error Code

Function

BOOL **VCS_GetDeviceErrorCode** (HANDLE KeyHandle, WORD NodeId, BYTE ErrorNumber, DWORD *pDeviceErrorCode, DWORD *pErrorCode)

Description

„VCS_GetDeviceErrorCode" returns the error code of selected error number .

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ErrorNumber	BYTE	Number (Object sub index) of device error (>= 1)	Object: 0x1003-0x

Return Parameters

pDeviceErrorCode	BYTE*	Actual error code from error history	Object: 0x1003-0x
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.3.3 Programming Example

Description

The example shows how to read the error history from a device. It is written in programming language C++.

```
//Global parameters
HANDLE KeyHandle = 1; //handle from opened interface
WORD NodeId = 1;      //node ID from connected device

//Functional parameters
BYTE nbOfDeviceError = 0;     //number of actual errors
DWORD functionErrorCode = 0;  //error code from function
DWORD deviceErrorCode = 0;    //error code from device

//get number of device errors
if(VCS_GetNbOfDeviceError (KeyHandle, NodeId, &nbOfDeviceError,
                                                    &functionErrorCode))
{
   //read device error code
   for(BYTE errorNumber = 1; subIndex <= nbOfDeviceError; errorNumber++)
   {
      if(!VCS_GetDeviceErrorCode(KeyHandle, NodeId, errorNumber,
                                    &deviceErrorCode, &functionErrorCode))
      {
        break;
      }
   }

}
```

6.4 Motion Info

This group defines all required functions for motion information:

6.4.1 Get Movement State

Function
BOOL **VCS_GetMovementState** (HANDLE KeyHandle, WORD NodeId, BOOL *pTargetReached, DWORD *pErrorCode)

Description
With „VCS_GetMovementState" it is possible to check, if drive has reached the target.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pTargetReached	BOOL*	The drive has reached the target. The function reads actual state of bit 10 from the status word.	Object: 0x6041-00 Bit 10
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.4.2 Get Position Is

Function
BOOL **VCS_GetPositionIs** (HANDLE KeyHandle, WORD NodeId, long *pPositionIs, DWORD *pErrorCode)

Description
„VCS_GetPositionIs" returns the position actual value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pPositionIs	long*	Position actual value	Object: 0x6064-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.4.3 Get Velocity Is

Function
BOOL **VCS_GetVelocityIs** (HANDLE KeyHandle, WORD NodeId, long *pVelocityIs, DWORD *pErrorCode)

Description
„VCS_GetVelocityIs" reads the velocity actual value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pVelocityIs	long*	Velocity actual value	Object: 0x606C-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.4.4 Get Velocity Is Averaged

Function
BOOL **VCS_GetVelocityIsAveraged** (HANDLE KeyHandle, WORD NodeId, long *pVelocityIsAveraged, DWORD *pErrorCode)

Description
„VCS_GetVelocityIsAveraged" reads the velocity actual averaged value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pVelocityIsAveraged	long*	Velocity actual value averaged	Object: 0x2028-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.4.5 Get Current Is

Function
BOOL **VCS_GetCurrentIs** (HANDLE KeyHandle, WORD NodeId, short *pCurrentIs, DWORD *pErrorCode)

Description
„VCS_GetCurrentIs" returns the current actual value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pCurrentIs	short*	Current actual value	Object: 0x6078-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.4.6 Get Current Is Averaged

Function
BOOL **VCS_GetCurrentIsAveraged** (HANDLE KeyHandle, WORD NodeId, short *pCurrentIsAveraged, DWORD *pErrorCode)

Description
„VCS_GetCurrentIsAveraged" returns the current actual averaged value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pCurrentIsAveraged	short*	Current actual value	Object: 0x2027-00
pErrorCode	DWORD*	Error information about the executed function	

Return Value	BOOL	Nonzero if successful; otherwise 0

6.4.7 Wait For Target Reached

Function
BOOL **VCS_WaitForTargetReached** (HANDLE KeyHandle, WORD NodeId, DWORD Timeout, DWORD *pErrorCode)

Description
„VCS_WaitForTargetReached" is waiting until the state is changed to target reached or the time is up.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
Timeout	DWORD	Max. wait time [ms] until target reached

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.5 Profile Position Mode

This group defines all required functions for profile position mode:

6.5.1 Activate Profile Position Mode

Function
BOOL **VCS_ActivateProfilePositionMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_ActivateProfilePositionMode" the device changes to profile position mode (PPM).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.5.2 Set Position Profile

Function
BOOL **VCS_SetPositionProfile** (HANDLE KeyHandle, WORD NodeId, DWORD ProfileVelocity, DWORD ProfileAcceleration, DWORD ProfileDeceleration, DWORD *pErrorCode)

Description
„VCS_SetPositionProfile" sets the position profile parameters.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ProfileVelocity	DWORD	Position profile velocity	Object: 0x6081-00
ProfileAcceleration	DWORD	Position profile acceleration	Object: 0x6083-00
ProfileDeceleration	DWORD	Position profile deceleration	Object: 0x6084-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.5.3 Get Position Profile

Function

BOOL **VCS_GetPositionProfile** (HANDLE KeyHandle, WORD NodeId,
DWORD *pProfileVelocity, DWORD *pProfileAcceleration, DWORD *pProfileDeceleration,
DWORD *pErrorCode)

Description

„VCS_GetPositionProfile" returns the position profile parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pProfileVelocity	DWORD*	Position profile velocity	Object: 0x6081-00
pProfileAcceleration	DWORD*	Position profile acceleration	Object: 0x6083-00
pProfileDeceleration	DWORD*	Position profile deceleration	Object: 0x6084-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.5.4 Move To Position

Function
BOOL **VCS_MoveToPosition** (HANDLE KeyHandle, WORD NodeId, long TargetPosition, BOOL Absolute, BOOL Immediately, DWORD *pErrorCode)

Description
With function „VCS_MoveToPosition" device movement starts with position profile to target position.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
TargetPosition	long	Target position	Object: 0x607A-00
Absolute	BOOL	TRUE starts an absolute, FALSE a relative movement	Object: 0x6040-00 Bit 6
Immediately	BOOL	TRUE starts immediately, FALSE waits to end of last positioning	Object: 0x6040-00 Bit 5

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.5.5 Get Target Position

Function
BOOL **VCS_GetTargetPosition** (HANDLE KeyHandle, WORD NodeId, long *pTargetPosition, DWORD *pErrorCode)

Description
The function „VCS_GetTargetPosition" returns the profile position mode target value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pTargetPosition	long*	Target position	Object: 0x607A-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.5.6 Halt Position Movement

Function
BOOL **VCS_HaltPositionMovement** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With function „VCS_HaltPositionMovement" movement stops with profile deceleration.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.5.7 Advanced Functions

6.5.7.1 Enable Position Window

Function
BOOL **VCS_EnablePositionWindow** (HANDLE KeyHandle, WORD NodeId, DWORD PositionWindow, WORD PositionWindowTime, DWORD *pErrorCode)

Description
With function „VCS_EnablePositionWindow" the position window is activated.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
PositionWindow	DWORD	Position window value	Object: 0x6067-00
PositionWindowTime	WORD	Position window time value	Object: 0x6068-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.5.7.2 Disable Position Window

Function
BOOL **VCS_DisablePositionWindow** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With function „VCS_DisablePositionWindow" the position window is deactivated.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.6 Profile Velocity Mode

This group defines all required functions for profile velocity mode:

6.6.1 Activate Profile Velocity Mode

Function
BOOL **VCS_ActivateProfileVelocityMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_ActivateProfileVelocityMode" the device changes to profile velocity mode (PVM).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.6.2 Set Velocity Profile

Function
BOOL **VCS_SetVelocityProfile** (HANDLE KeyHandle, WORD NodeId,
DWORD ProfileAcceleration, DWORD ProfileDeceleration, DWORD *pErrorCode)

Description
„VCS_SetVelocityProfile" sets the velocity profile parameters.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ProfileAcceleration	DWORD	Velocity profile acceleration	Object: 0x6083-00
ProfileDeceleration	DWORD	Velocity profile deceleration	Object: 0x6084-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.6.3 Get Velocity Profile

Function
BOOL **VCS_GetVelocityProfile** (HANDLE KeyHandle, WORD NodeId,
DWORD *pProfileAcceleration, DWORD *pProfileDeceleration, DWORD *pErrorCode)

Description
„VCS_GetVelocityProfile" returns the velocity profile parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pProfileAcceleration	DWORD*	Velocity profile acceleration	Object: 0x6083-00
pProfileDeceleration	DWORD*	Velocity profile deceleration	Object: 0x6084-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.6.4 Move With Velocity

Function
BOOL **VCS_MoveWithVelocity** (HANDLE KeyHandle, WORD NodeId, long TargetVelocity,
DWORD *pErrorCode)

Description
With function „VCS_MoveWithVelocity" device movement starts with velocity profile to target velocity.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
TargetVelocity	long	Target velocity	Object: 0x60FF-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.6.5 Get Target Velocity

Function
BOOL **VCS_GetTargetVelocity** (HANDLE KeyHandle, WORD NodeId, long *pTargetVelocity, DWORD *pErrorCode)

Description
The function „VCS_GetTargetVelocity" returns the profile velocity mode target value.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	

Return Parameters

pTargetVelocity	long*	Target velocity	Object: 0x60FF-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.6.6 Halt Velocity Movement

Function
BOOL **VCS_HaltVelocityMovement** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With function „VCS_HaltVelocityMovement" movement stops with profile deceleration.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.6.7 Advanced Functions

6.6.7.1 Enable Velocity Window

Function
BOOL **VCS_Enable Velocity Window** (HANDLE KeyHandle, WORD NodeId, DWORD VelocityWindow, WORD VelocityWindowTime, DWORD *pErrorCode)

Description
With function „VCS_EnableVelocityWindow" the velocity window is activated.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
VelocityWindow	DWORD	Velocity window value	Object: 0x606D-00
VelocityWindowTime	WORD	Velocity window time value	Object: 0x606E-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.6.7.2 Disable Velocity Window

Function
BOOL **VCS_DisableVelocityWindow** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With function „VCS_DisableVelocityWindow" the velocity window is deactivated.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.7 Homing Mode

This group defines all required functions for homing mode:

6.7.1 Activate Homing Mode

Function
BOOL **VCS_ActivateHomingMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_ActivateHomingMode" the device changes to homing mode (HM).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.7.2 Set Homing Parameter

Function
BOOL **VCS_SetHomingParameter** (HANDLE KeyHandle, WORD NodeId, DWORD HomingAcceleration, DWORD SpeedSwitch, DWORD SpeedIndex, long HomeOffset, WORD CurrentThreshold, long HomePosition, DWORD *pErrorCode)

Description
With function „VCS_SetHomingParameter" it is possible to write all homing parameters.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
HomingAcceleration	DWORD	Acceleration for homing profile	Object: 0x609A-00
SpeedSwitch	DWORD	Speed during search for switch	Object: 0x6099-01
SpeedIndex	DWORD	Speed during search for index signal	Object: 0x6099-02
HomeOffset	long	Home offset after homing	Object: 0x607C-00
CurrentThreshold	WORD	Current threshold for	Object: 0x2080-00

		homing method -1, -2, -3 and -4	
HomePosition	long	Assign the current Homing position with this value	Object: 0x2081-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Remarks
Parameter units depend on (position, velocity, acceleration) notation index.

6.7.3 Get Homing Parameter

Function
BOOL **VCS_GetHomingParameter** (HANDLE KeyHandle, WORD NodeId, DWORD *pHomingAcceleration, DWORD *pSpeedSwitch, DWORD *pSpeedIndex, long *pHomeOffset, WORD *pCurrentThreshold, long *pHomePosition, DWORD *pErrorCode)

Description
With function „VCS_GetHomingParameter" it is possible to read all homing parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pHomingAcceleration	DWORD*	Acceleration for homing profile	Object: 0x609A-00
pSpeedSwitch	DWORD*	Speed during search for switch	Object: 0x6099-01
pSpeedIndex	DWORD*	Speed during search for index signal	Object: 0x6099-02
pHomeOffset	long*	Home offset after homing	Object: 0x607C-00
pCurrentThreshold	WORD*	Current threshold for homing method -1, -2, -3 and -4	Object: 0x2080-00
pHomePosition	long*	Home position value	Object: 0x2081-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Remarks
Parameter units depend on (position, velocity, acceleration) notation index.

6.7.4 Find Home

Function
BOOL **VCS_FindHome** (HANDLE KeyHandle, WORD NodeId, __int8 HomingMethod, DWORD *pErrorCode)

Description
With function „VCS_FindHome" and the parameter „HomingMethod" it is possible to find the system home. For example a home switch.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
HomingMethod	__int8	Homing method	Object: 0x6098-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Homing method

Method Number	Constant Value	Description
35	HM_ACTUAL_POSITION	Actual Position
34	HM_INDEX_POSITIVE_SPEED	Index Positive Speed
33	HM_INDEX_NEGATIVE_SPEED	Index Negative Speed
27	HM_HOME_SWITCH_NEGATIVE_SPEED	Home Switch Negative Speed
23	HM_HOME_SWITCH_POSITIVE_SPEED	Home Switch Positive Speed
18	HM_POSITIVE_LIMIT_SWITCH	Positive Limit Switch
17	HM_NEGATIVE_LIMIT_SWITCH	Negative Limit Switch
11	HM_HOME_SWITCH_NEGATIVE_SPEED_ AND_INDEX	Home Switch Negative Speed & Index
7	HM_HOME_SWITCH_POSITIVE_SPEED_ AND_INDEX	Home Switch Positive Speed & Index
2	HM_POSITIVE_LIMIT_SWITCH_AND_INDEX	Positive Limit Switch & Index
1	HM_NEGATIVE_LIMIT_SWITCH_AND_INDEX	Negative Limit Switch & Index
0		No homing operation required
-1	HM_CURRENT_THRESHOLD_POSITIVE_ SPEED_AND_INDEX	Current Threshold Positive Speed & Index
-2	HM_CURRENT_THRESHOLD_NEGATIVE_ SPEED_AND_INDEX	Current Threshold Negative Speed & Index
-3	HM_CURRENT_THRESHOLD_POSITIVE_SPEED	Current Threshold Positive Speed
-4	HM_CURRENT_THRESHOLD_NEGATIVE_SPEED	Current Threshold Negative Speed

Table 10: Homing methods

6.7.5 Stop Homing

Function

BOOL **VCS_StopHoming** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description

The function „VCS_StopHoming" interrupts homing.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.7.6 Define Position

Function
BOOL **VCS_DefinePosition** (HANDLE KeyHandle, WORD NodeId, long HomePosition, DWORD *pErrorCode)

Description
The function „VCS_DefinePosition" is using homing method 35 (Actual Position) to set a new home position.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
HomePosition	long	Assign the homing position with this value	Object: 0x2081-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.7.7 Get Homing State

Function
BOOL **VCS_GetHomingState** (HANDLE KeyHandle, WORD NodeId, BOOL *pHomingAttained, BOOL *pHomingError, DWORD *pErrorCode)

Description
The function „VCS_GetHomingState" returns the states if the homing position is attained and if an homing error is occurred.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pHomingAttained	BOOL	0: Homing mode not yet completed 1: Homing mode successfully terminated
pHomingError	BOOL	0: No homing error 1: Homing error occurred
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.7.8 Wait For Homing Attained

Function
BOOL **VCS_WaitForHomingAttained** (HANDLE KeyHandle, WORD NodeId, DWORD Timeout, DWORD *pErrorCode)

Description
„VCS_WaitForHomingAttained" is waiting until the homing mode is successfully terminated or the time is up.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
Timeout	DWORD	Max. wait time [ms] until target reached

Return Parameters

pErrorCode	DWORD*	Error information about the executed function

Return Value	BOOL	Nonzero if successful; otherwise 0

6.8 Interpolated Position Mode (IPM)

6.8.1 Activate Interpolated Position Mode

Function
BOOL **VCS_ActivateInterpolatedPositionMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_ActivateInterpolatedPositionMode" the device changes to interpolated position mode (IPM).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.8.2 Set IPM Buffer Parameter

Function
BOOL **VCS_SetIpmBufferParameter** (HANDLE KeyHandle, WORD NodeId, WORD UnderflowWarningLimit, WORD OverflowWarningLimit, DWORD *pErrorCode)

Description
The function „VCS_SetIpmBufferParameter" set warning borders of the data input.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
UnderflowWarningLimit	WORD	This object gives lower signalization level of the data input FIFO.	Object: 0x20C4-02
OverflowWarningLimit	WORD	This object gives the higher signalization level of the data input FIFO.	Object: 0x20C4-03

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.8.3 Get IPM Buffer Parameter

Function
BOOL **VCS_GetIpmBufferParameter** (HANDLE KeyHandle, WORD NodeId, WORD* pUnderflowWarningLimit, WORD* pOverflowWarningLimit, DWORD* pMaxBufferSize, DWORD *pErrorCode)

Description
The function „VCS_GetIpmBufferParameter" read warning borders and the max. buffer size of the data input.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services

		(LSS).

Return Parameters

pUnderflowWarningLimit	WORD*	This object gives lower signalization level of the data input FIFO.	Object: 0x20C4-02
pOverflowWarningLimit	WORD*	This object gives the higher signalization level of the data input FIFO.	Object: 0x20C4-03
pMaxBufferSize	DWORD*	This object provides the maximal buffer size	Object: 0x60C4-01
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.8.4 Clear IPM Buffer

Function
BOOL **VCS_ClearIpmBuffer** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
The function „VCS_ClearIpmBuffer" clears input buffer and enables the access to the input buffer for the drive functions.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.8.5 Get Free IPM Buffer Size

Function
BOOL **VCS_GetFreeIpmBufferSize** (HANDLE KeyHandle, WORD NodeId, DWORD *pBufferSize, DWORD *pErrorCode)

Description
The function „VCS_GetFreeIpmBufferSize" read the actual free buffer size.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services.

Return Parameters

pBufferSize	DWORD	Actual free buffer size	Object: 0x60C4-02
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.8.6 Add PVT Value To IPM Buffer

Function
BOOL **VCS_AddPvtValueToIpmBuffer** (HANDLE KeyHandle, WORD NodeId, long Position, long Velocity, BYTE Time, DWORD *pErrorCode)

Description
The function „VCS_AddPvtValueToIpmBuffer" adds a new PVT reference point to the device.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services

		(LSS).	
Position	long	Position of the reference point	Object: 0x20C1-00
Velocity	long	Velocity of the reference point	
Time	BYTE	Time of the reference point	

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.8.7 Start IPM Trajectory

Function
BOOL **VCS_StartIpmTrajectory** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
The function „VCS_StartIpmTrajectory" starts the IPM trajectory.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.8.8 Stop IPM Trajectory

Function
BOOL **VCS_StopIpmTrajectory** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
The function „VCS_StopIpmTrajectory" stops the IPM trajectory.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.8.9 Get IPM Status

Function
BOOL **VCS_GetIpmStatus** (HANDLE KeyHandle, WORD NodeId, BOOL* pTrajectoryRunning, BOOL* pIsUnderflowWarning, BOOL* pIsOverflowWarning, BOOL* pIsVelocityWarning, BOOL* pIsAccelerationWarning, BOOL* pIsUnderflowError, BOOL* pIsOverflowError, BOOL* pIsVelocityError, BOOL* pIsAccelerationError, DWORD *pErrorCode)

Description
The function „VCS_GetIpmStatus" returns different warning and error states.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pTrajectoryRunning	BOOL*	State if IPM active	Object: 0x20C4-01
pIsUnderflowWarning	BOOL*	State if buffer underflow level is	

		reached	
pIsOverflowWarning	BOOL*	State if buffer overflow level is reached	
pIsVelocityWarning	BOOL*	State if IPM velocity greater than profile velocity	
pIsAccelerationWarning	BOOL*	State if IPM acceleration greater than profile acceleration	
pIsUnderflowError	BOOL*	State of underflow error	
pIsOverflowError	BOOL*	State of overflow error	
pIsVelocityError	BOOL*	State if IPM velocity greater than max. profile velocity	
pIsAccelerationError	BOOL*	State if IPM acceleration greater than max. profile acceleration	
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.9 Position Mode

This group defines all required functions for position mode:

6.9.1 Activate Position Mode

Function
BOOL **VCS_ActivatePositionMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_ActivatePositionMode" the device changes to position mode (PM).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.9.2 Set Position Must

Function
BOOL **VCS_SetPositionMust** (HANDLE KeyHandle, WORD NodeId, long PositionMust, DWORD *pErrorCode)

Description
„VCS_SetPositionMust" sets the position mode setting value.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
PositionMust	long	Position mode setting value	Object: 0x2062-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.9.3 Get Position Must

Function
BOOL **VCS_GetPositionMust** (HANDLE KeyHandle, WORD NodeId, long *pPositionMust, DWORD *pErrorCode)

Description
„VCS_GetPositionMust" reads the position mode setting value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pPositionMust	long*	Position mode setting value	Object: 0x2062-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.9.4 Advanced Functions

6.9.4.1 Activate Analog Position Setpoint

Function
BOOL **VCS_ActivateAnalogPositionSetpoint** (HANDLE KeyHandle, WORD NodeId, WORD AnalogInputNumber, float Scaling, long Offset, DWORD *pErrorCode)

Description
„VCS_ActivateAnalogPositionSetpoint" configures the selected analog input for analog position setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
AnalogInputNumber	WORD	Number of the used analog input	Object: 0x207B-01 or 0x207B-02
Scaling	float	The scaling factor for analog position setpoint functionality	Object: 0x2303-01
Offset	long	Offset for analog position setpoint functionality	Object: 0x2303-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.9.4.2 Deactivate Analog Position Setpoint

Function

BOOL **VCS_DeactivateAnalogPositionSetpoint** (HANDLE KeyHandle, WORD NodeId, WORD AnalogInputNumber, DWORD *pErrorCode)

Description

„VCS_DeactivateAnalogPositionSetpoint" disable the selected analog input for analog position setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
AnalogInputNumber	WORD	Number of the used analog input	Object: 0x207B-01 or 0x207B-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.9.4.3 Enable Analog Position Setpoint

Function

BOOL **VCS_EnableAnalogPositionSetpoint** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description

„VCS_EnableAnalogPositionSetpoint" enable the execution mask for analog position setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.9.4.4 Disable Analog Position Setpoint

Function

BOOL **VCS_DisableAnalogPositionSetpoint** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description

„VCS_DisableAnalogPositionSetpoint" disable the execution mask for analog position setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.10 Velocity Mode

This group defines all required functions for velocity mode:

6.10.1 Activate Velocity Mode

Function
BOOL **VCS_ActivateVelocityMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_ActivateVelocityMode" the device changes to velocity mode (VM).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.10.2 Set Velocity Must

Function
BOOL **VCS_SetVelocityMust** (HANDLE KeyHandle, WORD NodeId, long VelocityMust, DWORD *pErrorCode)

Description
The function „VCS_SetVelocityMust" sets the velocity mode setting value.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
VelocityMust	long	Velocity mode setting value	Object: 0x206B-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.10.3 Get Velocity Must

Function
BOOL **VCS_GetVelocityMust**(HANDLE KeyHandle, WORD NodeId, long *pVelocityMust, DWORD *pErrorCode)

Description
The function „VCS_GetVelocityMust" returns the velocity mode setting value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pVelocityMust	long*	Velocity mode setting value	Object: 0x206B-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.10.4 Advanced Functions

6.10.4.1 Activate Analog Velocity Setpoint

Function
BOOL **VCS_ActivateAnalogVelocitySetpoint**(HANDLE KeyHandle, WORD NodeId, WORD AnalogInputNumber, float Scaling, long Offset, DWORD *pErrorCode)

Description
„VCS_ActivateAnalogVelocitySetpoint" configures the selected analog input for analog velocity setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
AnalogInputNumber	WORD	Number of the used analog input	Object: 0x207B-01 or 0x207B-02
Scaling	float	The scaling factor for analog velocity setpoint functionality	Object: 0x2302-01
Offset	long	Offset for analog velocity setpoint functionality	Object: 0x2302-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.10.4.2 Deactivate Analog Velocity Setpoint

Function
BOOL **VCS_DeactivateAnalogVelocitySetpoint** (HANDLE KeyHandle, WORD NodeId, WORD AnalogInputNumber, DWORD *pErrorCode)

Description
„VCS_DeactivateAnalogVelocitySetpoint" disable the selected analog input for analog velocity setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
AnalogInputNumber	WORD	Number of the used analog input	Object: 0x207B-01 or 0x207B-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.10.4.3 Enable Analog Velocity Setpoint

Function
BOOL **VCS_EnableAnalogVelocitySetpoint** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
„VCS_EnableAnalogVelocitySetpoint" enable the execution mask for analog velocity setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.10.4.4 Disable Analog Velocity Setpoint

Function
BOOL **VCS_DisableAnalogVelocitySetpoint** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
„VCS_DisableAnalogVelocitySetpoint" disable the execution mask for analog velocity setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.11 Current Mode

This group defines all required functions for current mode:

6.11.1 Activate Current Mode

Function
BOOL **VCS_ActivateCurrentMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
Function „VCS_ActivateCurrentMode" changes operational mode to current mode.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.11.2 Get Current Must

Function
BOOL **VCS_GetCurrentMust** (HANDLE KeyHandle, WORD NodeId, short *pCurrentMust, DWORD *pErrorCode)

Description
With function „VCS_GetCurrentMust" it is possible to read the current mode setting value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pCurrentMust	short*	Current mode setting value	Object: 0x2030-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.11.3 Set Current Must

Function
BOOL **VCS_SetCurrentMust** (HANDLE KeyHandle, WORD NodeId, short CurrentMust, DWORD *pErrorCode)

Description
With function „VCS_SetCurrentMust" it is possible to write current mode setting value.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
CurrentMust	short	Current mode setting value	Object: 0x2030-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.11.4 Advanced Functions

6.11.4.1 Activate Analog Current Setpoint

Function
BOOL **VCS_ActivateAnalogCurrentSetpoint** (HANDLE KeyHandle, WORD NodeId, WORD AnalogInputNumber, float Scaling, short Offset, DWORD *pErrorCode)

Description
„VCS_ActivateAnalogCurrentSetpoint" configures the selected analog input for analog current setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
AnalogInputNumber	WORD	Number of the used analog input	Object: 0x207B-01 or 0x207B-02
Scaling	float	The scaling factor for analog current setpoint functionality	Object: 0x2301-01
Offset	short	Offset for analog current setpoint functionality	Object: 0x2301-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.11.4.2 Deactivate Analog Current Setpoint

Function
BOOL **VCS_DeactivateAnalogCurrentSetpoint** (HANDLE KeyHandle, WORD NodeId, WORD AnalogInputNumber, DWORD *pErrorCode)

Description
„VCS_DeactivateAnalogCurrentSetpoint" disable the selected analog input for analog current setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
AnalogInputNumber	WORD	Number of the used analog input	Object: 0x207B-01 or 0x207B-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.11.4.3 Enable Analog Current Setpoint

Function
BOOL **VCS_EnableAnalogCurrentSetpoint** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
„VCS_EnableAnalogCurrentSetpoint" enable the execution mask for analog current setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.11.4.4 Disable Analog Current Setpoint

Function
BOOL **VCS_DisableAnalogCurrentSetpoint** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
„VCS_DisableAnalogCurrentSetpoint" disable the execution mask for analog current setpoint.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.12 Master Encoder Mode

6.12.1 Activate Master Encoder Mode

Function
BOOL **VCS_ActivateMasterEncoderMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_ActivateMasterEncoderMode" the device changes to master encoder mode (MEM).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.12.2 Set Master Encoder Parameter

Function

BOOL **VCS_SetMasterEncoderParameter** (HANDLE KeyHandle, WORD NodeId, WORD ScalingNumerator, WORD ScalingDenominator, BYTE Polarity, DWORD MaxVelocity, DWORD MaxAcceleration, DWORD *pErrorCode)

Description

With function „VCS_SetMasterEncoderParameter" it is possible to write all parameters for master encoder mode.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ScalingNumerator	WORD	Scaling numerator for position calculation	Object: 0x2300-02
ScalingDenominator	WORD	Scaling denominator for position calculation	Object: 0x2300-03
Polarity	BYTE	Polarity of the direction input. 0: Positive 1: Negative	Object: 0x2300-04
MaxVelocity	DWORD	This parameter is the maximal allowed speed during a profiled move.	Object: 0x607F-01
MaxAcceleration	DWORD	Defines the maximal allowed acceleration.	Object: 0x60C5-01

6.12.3 Get Master Encoder Parameter

Function

BOOL **VCS_GetMasterEncoderParameter** (HANDLE KeyHandle, WORD NodeId, WORD* pScalingNumerator, WORD* pScalingDenominator, BYTE* pPolarity, DWORD* pMaxVelocity, DWORD* pMaxAcceleration, DWORD *pErrorCode)

Description

With function „VCS_GetMasterEncoderParameter" it is possible to read all parameters for master encoder mode.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pScalingNumerator	WORD*	Scaling numerator for position calculation	Object: 0x2300-02
pScalingDenominator	WORD*	Scaling denominator for position calculation	Object: 0x2300-03
pPolarity	BYTE*	Polarity of the direction input. 0: Positive 1: Negative	Object: 0x2300-04

pMaxVelocity	DWORD*	This parameter is the maximal allowed speed during a profiled move.	Object: 0x607F-01
pMaxAcceleration	DWORD*	Defines the maximal allowed acceleration.	Object: 0x60C5-01
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.13 Step Direction Mode

6.13.1 Activate Step Direction Mode

Function
BOOL **VCS_ActivateStepDirectionMode** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description
With the function „VCS_ActivateStepDirectionMode" the device changes to step direction mode (SDM).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.13.2 Set Step Direction Parameter

Function
BOOL **VCS_SetStepDirectionParameter** (HANDLE KeyHandle, WORD NodeId, WORD ScalingNumerator, WORD ScalingDenominator, BYTE Polarity, DWORD MaxVelocity, DWORD MaxAcceleration, DWORD *pErrorCode)

Description
With function „VCS_SetStepDirectionParameter" it is possible to write all parameters for step direction mode.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ScalingNumerator	WORD	Scaling numerator for position calculation	Object: 0x2300-02
ScalingDenominator	WORD	Scaling denominator for position calculation	Object: 0x2300-03
Polarity	BYTE	Polarity of the direction input. 0: Positive 1: Negative	Object: 0x2300-04
MaxVelocity	DWORD	This parameter is the maximal allowed speed during a profiled move.	Object: 0x607F-01
MaxAcceleration	DWORD	Defines the maximal allowed acceleration.	Object: 0x60C5-01

6.13.3 Get Step Direction Parameter

Function
BOOL **VCS_GetStepDirectionParameter** (HANDLE KeyHandle, WORD NodeId, WORD* pScalingNumerator, WORD* pScalingDenominator, BYTE* pPolarity, DWORD* pMaxVelocity, DWORD* pMaxAcceleration, DWORD *pErrorCode)

Description
With function „VCS_GetStepDirectionParameter" it is possible to read all parameters for step direction mode.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pScalingNumerator	WORD*	Scaling numerator for position calculation	Object: 0x2300-02
pScalingDenominator	WORD*	Scaling denominator for position calculation	Object: 0x2300-03
pPolarity	BYTE*	Polarity of the direction input. 0: Positive 1: Negative	Object: 0x2300-04
pMaxVelocity	DWORD*	This parameter is the maximal allowed speed during a profiled move.	Object: 0x607F-01
pMaxAcceleration	DWORD*	Defines the maximal allowed acceleration.	Object: 0x60C5-01
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.14 Inputs Outputs

This group defines all required functions for inputs and outputs information:

Remark
More information about the inputs from the other devices are available in the "Firmware Specification" documents!

6.14.1 Get All Digital Inputs

Function
BOOL **VCS_GetAllDigitalInputs** (HANDLE KeyHandle, WORD NodeId, WORD *pInputs, DWORD *pErrorCode)

Description
„VCS_GetAllDigitalInputs" returns state of all digital inputs.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pInputs	WORD*	Display the state of the digital input functionalities. If a bit is read as "1", the functionality is activated.	Object: 0x2071-01
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.14.2 Get All Digital Outputs

Function
BOOL **VCS_GetAllDigitalOutputs** (HANDLE KeyHandle, WORD NodeId, WORD *pOutputs, DWORD *pErrorCode)

Description
„VCS_GetAllDigitalOutputs" returns state of all digital outputs.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	

Return Parameters

pOutputs	WORD*	State of all digital outputs. If a bit is read as "1", the state activated.	Object: 0x2078-01
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.14.3 Set All Digital Outputs

Function
BOOL **VCS_SetAllDigitalOutputs** (HANDLE KeyHandle, WORD NodeId, WORD Outputs, DWORD *pErrorCode)

Description
„VCS_SetAllDigitalOutputs" set state of all digital outputs.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
Outputs	WORD	State of all digital outputs. If a bit is written as "1", the state is activated.	Object: 0x2078-01

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.4 Get Analog Input

Function
BOOL **VCS_GetAnalogInput** (HANDLE KeyHandle, WORD NodeId, WORD InputNumber, WORD *pAnalogValue, DWORD *pErrorCode)

Description
„VCS_GetAnalogInput" returns the value from an analog input.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
InputNumber	WORD	Analog input number

Return Parameters

pAnalogValue	WORD*	Analog value from input	Object: 0x207C-0?
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.14.5 Set Analog Output

Function
BOOL **VCS_SetAnalogOutput** (HANDLE KeyHandle, WORD NodeId, WORD OutputNumber, WORD AnalogValue, DWORD *pErrorCode)

Description
„VCS_SetAnalogOutput" set the voltage level of an analog output.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
OutputNumber	WORD	Analog output number	
pAnalogValue	WORD*	Analog value for output	Object: 0x207E-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.6 Position Compare

6.14.6.1 Set Position Compare Parameter

Function
BOOL **VCS_SetPositionCompareParameter** (HANDLE KeyHandle, WORD NodeId, BYTE OperationalMode, BYTE IntervalMode, BYTE DirectionDependency, WORD IntervalWidth, WORD IntervalRepetitions, WORD PulseWidth, DWORD* pErrorCode)

Description
„VCS_SetPositionCompareParameter" write all parameters for position compare.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
OperationalMode	BYTE	Used operational mode in position sequence mode: (see Table 11)	Object: 0x207A-01
IntervalMode	BYTE	Used interval mode in position sequence mode: (see Table 12)	
DirectionDependency	BYTE	Used direction dependency in position sequence mode: (see Table 13)	
IntervalWidth	WORD	This object holds the width of the position intervals	Object: 0x207A-03
IntervalRepetitions	WORD	This object allows to configure the number of position intervals to be considered by position compare	Object: 0x207A-04
PulseWidth	WORD	This object configures the pulse width of the trigger output	Object: 0x207A-05

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

OperationalMode

Description	Value	Constant name
Single position mode	0	PCO_SINGLE_POSITION_MODE
Position sequence mode	1	PCO_POSITION_SEQUENCE_MODE

Table 11: Position Compare - Operational mode

IntervalMode

Description	Value	Constant name
Interval positions are set in negative direction relative to the position compare reference position	0	PCI_NEGATIVE_DIR_TO_REFPOS
Interval positions are set in positive direction relative to the position compare reference position	1	PCI_POSITIVE_DIR_TO_REFPOS
Interval positions are set in positive and negative direction relative to the position compare reference position	2	PCI_BOTH_DIR_TO_REFPOS

Table 12: Position Compare - Interval mode

DirectionDependency

Description	Value	Constant name
Positions are compared only if actual motor direction is negative	0	PCD_MOTOR_DIRECTION_NEGATIVE
Positions are compared only if actual motor direction is positive	1	PCD_MOTOR_DIRECTION_POSITIVE
Positions are compared regardless of the actual motor direction	2	PCD_MOTOR_DIRECTION_BOTH

Table 13: Position Compare - Direction Dependency

6.14.6.2 Get Position Compare Parameter

Function

BOOL **VCS_GetPositionCompareParameter** (HANDLE KeyHandle, WORD NodeId, BYTE* pOperationalMode, BYTE* pIntervalMode, BYTE* pDirectionDependency, WORD* pIntervalWidth, WORD* pIntervalRepetitions, WORD* pPulseWidth, DWORD* pErrorCode)

Description

„VCS_GetPositionCompareParameter" read all parameters for position compare.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pOperationalMode	BYTE*	Used operational mode in position sequence mode: (see Table 11)	Object: 0x207A-01
pIntervalMode	BYTE*	Used interval mode in position sequence mode: (see Table 12)	
pDirectionDependency	BYTE*	Used direction dependency in position sequence mode: (see Table 13)	
pIntervalWidth	WORD*	This object holds the width of the position intervals	Object: 0x207A-03
pIntervalRepetitions	WORD*	This object allows to configure the number of position intervals to be considered by position compare	Object: 0x207A-04
pPulseWidth	WORD*	This object configures the pulse width of the trigger output	Object: 0x207A-05
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.14.6.3 Activate Position Compare

Function

BOOL **VCS_ActivatePositionCompare** (HANDLE KeyHandle, WORD NodeId, WORD DigitalOutputNumber, BOOL Polarity, DWORD* pErrorCode)

Description

Function „VCS_ActivatePositionCompare" enables the output to position compare method.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
DigitalOutputNumber	WORD	Selected digital output for position compare	Object: 0x2079
Polarity	BOOL	Polarity of the selected output	Object: 0x2078-03

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.6.4 Deactivate Position Compare

Function

BOOL **VCS_DeactivatePositionCompare** (HANDLE KeyHandle, WORD NodeId, WORD DigitalOutputNumber, DWORD* pErrorCode)

Description

Function „VCS_DeactivatePositionCompare" disables the output to position compare method.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
DigitalOutputNumber	WORD	Selected digital output for position compare	Object: 0x2079

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.6.5 Enable Position Compare

Function

BOOL **VCS_EnablePositionCompare** (HANDLE KeyHandle, WORD NodeId, DWORD* pErrorCode)

Description

Function „VCS_EnablePositionCompare" enables the output mask for position compare method.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.6.6 Disable Position Compare

Function
BOOL **VCS_DisablePositionCompare** (HANDLE KeyHandle, WORD NodeId, DWORD* pErrorCode)

Description
Function „VCS_DisablePositionCompare" disables the output mask from position compare method.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.6.7 Set Position Compare Reference Position

Function
BOOL **VCS_SetPositionCompareReferencePosition** (HANDLE KeyHandle, WORD NodeId, long ReferencePosition, DWORD* pErrorCode)

Description
„VCS_SetPositionCompareReferencePosition" writes the reference position for position compare method.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ReferencePosition	long	This object holds the position that is compared with the position actual value	Object: 0x207A-02

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.7 Position Marker

6.14.7.1 Set Position Marker Parameter

Function
BOOL **VCS_SetPositionMarkerParameter** (HANDLE KeyHandle, WORD NodeId, BYTE PositionMarkerEdgeType, BYTE PositionMarkerMode, DWORD* pErrorCode)

Description
„VCS_SetPositionMarkerParameter" write all parameters for position marker method.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
PositionMarkerEdgeType	BYTE	The value of this object defines on what kind of edge the position should be captured: (see Table 14)	Object: 0x2074-02
PositionMarkerMode	BYTE	This object defines the position marker-capturing mode:	Object: 0x2074-03

		(see Table 15)	

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

PositionMarkerEdgeType

Description	Value	Constant name
Both edges	0	PET_BOTH_EDGES
Rising edge	1	PET_RISING_EDGE
Falling edge	2	PET_FALLING_EDGE

Table 14: Position marker edge types

PositionMarkerMode

Description	Value	Constant name
Continuous	0	PM_CONTINUOUS
Single	1	PM_SINGLE
Multiple	2	PM_MULTIPLE

Table 15: Position marker modes

6.14.7.2 Get Position Marker Parameter

Function

BOOL **VCS_GetPositionMarkerParameter** (HANDLE KeyHandle, WORD NodeId, BYTE* pPositionMarkerEdgeType, BYTE* pPositionMarkerMode, DWORD* pErrorCode)

Description

„VCS_GetPositionMarkerParameter" read all parameters for position marker method.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pPositionMarkerEdgeType	BYTE*	The value of this object defines on what kind of edge the position should be captured: (see Table 14)	Object: 0x2074-02
pPositionMarkerMode	BYTE*	This object defines the position marker capturing mode: (see Table 15)	Object: 0x2074-03
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.14.7.3 Activate Position Marker

Function

BOOL **VCS_ActivatePositionMarker** (HANDLE KeyHandle, WORD NodeId, WORD DigitalInputNumber, BOOL Polarity, DWORD* pErrorCode)

Description

Function „VCS_ActivatePositionMarker" enables the digital input to position marker method.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
DigitalInputNumber	WORD	Selected digital input for position marker	Object: 0x2070
Polarity	BOOL	Polarity of the selected input	Object: 0x2071-03

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.7.4 Deactivate Position Marker

Function

BOOL **VCS_DeactivatePositionMarker** (HANDLE KeyHandle, WORD NodeId, WORD DigitalInputNumber, DWORD* pErrorCode)

Description

Function „VCS_DeactivatePositionMarker" disables the digital input to position marker method.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
DigitalInputNumber	WORD	Selected digital input for position marker	Object: 0x2070

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

6.14.7.5 Read Position Marker Counter

Function

BOOL **VCS_ReadPositionMarkerCounter** (HANDLE KeyHandle, WORD NodeId, WORD *pCount, DWORD *pErrorCode)

Description

„VCS_ReadPositionMarkerCounter" returns the number of the detected edges.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	

Return Parameters

pCount	WORD*	This object counts the number of the detected edges.	Object: 0x2074-04
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.14.7.6 Read Position Marker Captured Position

Function

BOOL **VCS_ReadPositionMarkerCapturedPosition** (HANDLE KeyHandle, WORD NodeId, WORD CounterIndex, long* pCapturedPosition, DWORD *pErrorCode)

Description

„VCS_ReadPositionMarkerCapturedPosition" returns the last captured position or the position from the position marker history.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
CounterIndex	WORD	0: Read position marker captured position	Object: 0x2074-01
		1 or 2: Read position marker history	Object: 0x2074-05 or 06

Return Parameters

pCapturedPosition	long*	This object contains the captured position or the position marker history	Object: 0x2074-01 or 0x2074-05/06
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

6.14.7.7 Reset Position Marker Counter

Function

BOOL **VCS_ResetPositionMarkerCounter** (HANDLE KeyHandle, WORD NodeId, DWORD *pErrorCode)

Description

„VCS_ResetPositionMarkerCounter" clears the counter and the captured positions by writing zero to object position marker counter (0x2074-04).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

7 Data Recording Functions

7.1 Data Recorder Setup

7.1.1 Set Recorder Parameter

Function
BOOL **VCS_SetRecorderParameter** (HANDLE KeyHandle, WORD NodeId, WORD SamplingPeriod, WORD NbOfPrecedingSamples, WORD PulseWidth, DWORD* pErrorCode)

Description
„VCS_SetRecorderParameter" writes parameters for data recorder .

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
SamplingPeriod	WORD	Sampling Period as a multiple of the current regulator cycle (n-times 0.1ms)	Object: 0x2012-00
NbOfPrecedingSamples	WORD	Number of preceding samples (data history).	Object: 0x2013-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

7.1.2 Get Recorder Parameter

Function
BOOL **VCS_GetRecorderParameter** (HANDLE KeyHandle, WORD NodeId, WORD* pSamplingPeriod, WORD* pNbOfPrecedingSamples, WORD PulseWidth, DWORD* pErrorCode)

Description
„VCS_GetRecorderParameter" reads parameters for data recorder .

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pSamplingPeriod	WORD*	Sampling Period as a multiple of the current regulator cycle (n-times 0.1ms)	Object: 0x2012-00
pNbOfPrecedingSamples	WORD*	Number of preceding samples (data history).	Object: 0x2013-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

7.1.3 Enable Trigger

Function
BOOL **VCS_EnableTrigger** (HANDLE KeyHandle, WORD NodeId, BYTE TriggerType, DWORD* pErrorCode)

Description
„VCS_EnableTrigger" connects trigger(-s) for data recording.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
TriggerType	BYTE	Configuration of Auto Trigger functions. If a bit is write as one the trigger is activated: (see Table 16) It is possible to activate more than one trigger at the same time.	Object: 0x2011-00

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Trigger Types

Description	Value	Constant name
Trigger movement start	1	DR_MOVEMENT_START_TRIGGER
Error trigger	2	DR_ERROR_TRIGGER
Digital input trigger	4	DR_DIGITAL_INPUT_TRIGGER
Trigger movement end	8	DR_MOVEMENT_END_TRIGGER

Table 16: Data recorder trigger types

7.1.4 Disable all Triggers

Function
BOOL **VCS_DisableAllTrigger** (HANDLE KeyHandle, WORD NodeId, DWORD* pErrorCode)

Description
„VCS_DisableAllTrigger" sets data recorder configuration (0x2011-00) for triggers to zero.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

7.1.5 Activate Channel

Function

BOOL **VCS_ActivateChannel** (HANDLE KeyHandle, WORD NodeId, BYTE ChannelNumber, WORD ObjectIndex, BYTE ObjectSubIndex, BYTE ObjectSize, DWORD* pErrorCode)

Description

„VCS_ActivateChannel" connects object for data recording.
Start with channel number one! For every activated channel the number of sampling variables (Object 0x2014-00) will be incremented

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ChannelNumber	BYTE	Channel number [1 … 4]	
ObjectIndex	WORD	Object index for data recording	Object: 0x2015-ChannelNumber
ObjectSubIndex	BYTE	Object sub index for data recording	Object: 0x2016-ChannelNumber
ObjectSize	BYTE	Object size in bytes for data recording	

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

7.1.6 Deactivate all Channels

Function

BOOL **VCS_DeactivateAllChannel** (HANDLE KeyHandle, WORD NodeId, DWORD* pErrorCode)

Description

„VCS_DeactivateAllChannel" sets all data recording objects to zero (0x2014, 0x2015 and 0x2016).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

7.2 Data Recorder Status

Functions are not available in the Linux library!

7.2.1 Start Recorder

Function
BOOL **VCS_StartRecorder** (HANDLE KeyHandle, WORD NodeId, DWORD* pErrorCode)

Description
„VCS_StartRecorder" starts the data recording.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

7.2.2 Stop Recorder

Function
BOOL **VCS_StopRecorder** (HANDLE KeyHandle, WORD NodeId, DWORD* pErrorCode)

Description
„VCS_StopRecorder" stops the data recording.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

7.2.3 Force Trigger

Function
BOOL **VCS_ForceTrigger** (HANDLE KeyHandle, WORD NodeId, DWORD* pErrorCode)

Description
„VCS_ForceTrigger" forces the data recording triggers.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

7.2.4 Is Recorder Running

Function
BOOL **VCS_IsRecorderRunning** (HANDLE KeyHandle, WORD NodeId, BOOL* pRunning, DWORD* pErrorCode)

Description
„VCS_IsRecorderRunning" returns data recorder status running.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pRunning	BOOL	1: Data recorder running 0: Data recorder stopped	Object: 0x2017-00, (bit 0)
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Additional informations
Function is not available in the Linux library

7.2.5 Is Recorder Triggered

Function
BOOL **VCS_IsRecorderTriggered** (HANDLE KeyHandle, WORD NodeId, BOOL* pTriggered, DWORD* pErrorCode)

Description
„VCS_IsRecorderTriggered" returns data recorder status triggered.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pTriggered	BOOL	1: Data recorder triggered 0: Data recorder not triggered	Object: 0x2017-00 (bit 1)
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Additional informations
Function is not available in the Linux library

7.3 Data Recorder Data

7.3.1 Read Channel Vector Size

Function
BOOL **VCS_ReadChannelVectorSize**(HANDLE KeyHandle, WORD NodeId, DWORD* pVectorSize, DWORD* pErrorCode)

Description
„VCS_ReadChannelVectorSize" returns the maximal number of samples per variable. This parameter is dynamically calculated by the data recorder.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pVectorSize	DWORD	Maximal number of samples per variable.	Object: 0x2018-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Additional informations
Function is not available in the Linux library

7.3.2 Read Channel Data Vector

Function
BOOL **VCS_ReadChannelDataVector** (HANDLE KeyHandle, WORD NodeId, BYTE ChannelNumber, BYTE* pDataVector, DWORD VectorSize, DWORD* pErrorCode)

Description
„VCS_ReadChannelDataVector" returns the data points of a selected channel.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ChannelNumber	BYTE	Selected channel	
VectorSize	DWORD	Size of data points	Object: 0x2018-00

Return Parameters

pDataVector	BYTE	Data points of selected channel	Object: 0x201B-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Additional informations
Function is not available in the Linux library

7.3.3 Show Channel Data Dialog

Function
BOOL **VCS_ShowChannelDataDlg**(HANDLE KeyHandle, WORD NodeId, DWORD* pErrorCode)

Description
„VCS_ShowChannelDataDlg" opens the dialog to show the data channel(-s).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

7.3.4 Export Channel Data to File

Function
BOOL **VCS_ExportChannelDataToFile** (HANDLE KeyHandle, WORD NodeId, char* FileName, DWORD* pErrorCode)

Description
„VCS_ExportChannelDataToFile" saves data point in a file.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
FileName	char*	Path and file name for save data points. File endings: *.csv, *.txt or *.rda.

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Additional informations
Function is not available in the Linux library

7.4 Advanced Functions

7.4.1 Read Data Buffer

Function
BOOL **VCS_ReadDataBuffer**(HANDLE KeyHandle, WORD NodeId, BYTE* pDataBuffer, DWORD BufferSizeToRead, DWORD* pBufferSizeRead, WORD* pVectorStartOffset, WORD* pMaxNbOfSamples, WORD* pNbOfRecordedSamples, DWORD* pErrorCode)

Description
„VCS_ReadDataBuffer" returns the buffer data points.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
BufferSizeToRead	DWORD	Buffer size

Return Parameters

pDataBuffer	BYTE*	Data points	Object: 0x201B-00
pBufferSizeRead	DWORD*	Size of read data buffer	
pVectorStartOffset	WORD*	Offset to the start of the recorded data vector within the ring buffer.	Object: 0x201A-00
pMaxNbOfSamples	WORD*	Maximal number of samples per variable.	Object: 0x2018-00
pNbOfRecordedSamples	WORD*	Number of recorded samples.	Object: 0x2019-00
pErrorCode	DWORD*	Error information about the executed function	
Return Value	BOOL	Nonzero if successful; otherwise 0	

7.4.2 Extract Channel Data Vector

Function
BOOL **VCS_ExtractChannelDataVector**(HANDLE KeyHandle, WORD NodeId, BYTE ChannelNumber, BYTE* pDataBuffer, DWORD BufferSize, BYTE* pDataVector, DWORD VectorSize, WORD VectorStartOffset, WORD MaxNbOfSamples, WORD NbOfRecordedSamples, DWORD* pErrorCode)

Description
„VCS_ExtractChannelDataVector" returns the vector of one data channel.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).	
ChannelNumber	BYTE	Selected channel	
pDataBuffer	BYTE	Data points	Object: 0x201B-00
BufferSize	DWORD	Buffer size	
VectorSize	DWORD	Vector size	
VectorStartOffset	WORD	Offset to the start of the recorded data vector within the ring buffer.	Object: 0x201A-00
MaxNbOfSamples	WORD	Maximal number of samples per variable.	Object: 0x2018-00
NbOfRecordedSamples	WORD	Number of recorded samples.	Object: 0x2019-00

Return Parameters

pDataVector	BYTE*	Data points of the channel
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

8 Low Layer Functions

8.1 Send CAN Frame

Function
BOOL **VCS_SendCANFrame** (HANDLE KeyHandle, WORD CobID, WORD Length, void *pData, DWORD *pErrorCode)

Description
„VCS_SendCANFrame" sends a general CAN frame to the CAN bus.

Parameters

KeyHandle	HANDLE	Handle for port access
CobID	WORD	CAN frame 11-bit identifier
Length	WORD	CAN frame data length
pData	void*	CAN frame data

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

8.2 Read CAN Frame

Function
BOOL **VCS_ReadCANFrame** (HANDLE KeyHandle, WORD CobID, WORD Length, void *pData, DWORD Timeout , DWORD *p ErrorCode)

Description
„VCS_ReadCANFrame" reads a general CAN frame from the CAN bus.

Parameters

KeyHandle	HANDLE	Handle for port access
CobID	WORD	CAN frame 11-bit identifier
Length	WORD	CAN frame data length
Timeout	WORD	Maximum waiting period

Return Parameters

pData	void*	CAN frame data
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

8.3 Request CAN Frame

Function
BOOL **VCS_RequestCANFrame** (HANDLE KeyHandle, WORD CobID, WORD Length, void *pData, DWORD *pErrorCode)

Description
„VCS_RequestCANFrame" requests a general CAN frame from the CAN bus using Remote Transmit Request (RTR).

Parameters

KeyHandle	HANDLE	Handle for port access
CobID	WORD	CAN frame 11-bit identifier
Length	WORD	CAN frame data length

Return Parameters

pData	void*	CAN frame data
pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

8.4 Send NMT Service

Function
BOOL **VCS_SendNMTService** (HANDLE KeyHandle, WORD NodeId,
WORD CommandSpecifier, DWORD *pErrorCode)

Description
The function „VCS_SendNMTService" is used to send a NMT protocol from a master to a slave. It is a command without acknowledge.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Node ID of the addressed device. ID is given from hardware switches or the layer setting services (LSS).
CommandSpecifier	WORD	NMT service (see Table 17)

Return Parameters

pErrorCode	DWORD*	Error information about the executed function
Return Value	BOOL	Nonzero if successful; otherwise 0

Command Specifier

Description	Value	Constant name
Start remote node	1	NCS_START_REMOTE_NODE
Stop remote node	2	NCS_STOP_REMOTE_NODE
Enter pre-operational	128	NCS_ENTER_PRE_OPERATIONAL
Reset node	129	NCS_RESET_NODE
Reset communication	130	NCS_RESET_COMMUNICATION

Table 17: Command specifier

9 Error Overview

9.1 Overview Communication Errors

Abort Code	Name	Error cause
0x0000 0000	No Communication Error	The communication was successful
0x0503 0000	Toggle Error	Toggle bit not alternated
0x0504 0000	SDO Time Out	SDO protocol timed out
0x0504 0001	Client/server specifier Error	Client/server command specifier not valid or unknown
0x0504 0002	Invalid block size	Invalid block size (block mode only)
0x0504 0003	Invalid sequence	Invalid sequence number (block mode only)
0x0504 0004	CrcError	CRC error (block mode only)
0x0504 0005	Out of Memory Error	Out of Memory
0x0601 0000	Access Error	Unsupported access to an object (e.g. write command to a read-only object)
0x0601 0001	Write Only	Read command to a write only object
0x0601 0002	Read Only	Write command to a read only object
0x0602 0000	Object does not exist Error	The last read or write command had a wrong object index or sub-index
0x0604 0041	PDO mapping Error	The object cannot be mapped to the PDO
0x0604 0042	PDO length Error	The number and length of the objects to be mapped would exceed PDO length
0x0604 0043	General parameter Error	General parameter incompatibility
0x0604 0047	General Intern Incompatibility Error	General internal incompatibility in device
0x0606 0000	Hardware Error	Access failed due to an hardware error
0x0607 0010	Service Parameter Error	Data type does not match, length or service parameter does not match
0x0607 0012	Service Parameter Error too High Error	Data type does not match, length or service parameter too high
0x0607 0013	Service Parameter Error too Low Error	Data type does not match, length or service parameter too low
0x0609 0011	Object Sub-index Error	The last read or write command had a wrong Object sub-index
0x0609 0030	Value Range Error	Value range of parameter exceeded
0x0609 0031	Value too High Error	Value of parameter written too high
0x0609 0032	Value too Low Error	Value of parameter written too low
0x0609 0036	Maximum less Minimum Error	Maximum value is less than minimum value
0x0800 0000	General Error	General error
0x0800 0020	Transfer or store Error	Data cannot be transferred or stored
0x0800 0021	Local control Error	Data cannot be transferred or stored to application because of local control
0x0800 0022	Wrong Device State	Data cannot be transferred or stored to application because of the present device state
0x0F00 FFB9	Error CAN id	Wrong CAN id
0x0F00 FFBC	Error Service Mode	The device is not in service mode
0x0F00 FFBE	Password Error	The password is wrong
0x0F00 FFBF	Illegal Command Error	The RS232 command is illegal (does not exist)
0x0F00 FFC0	Wrong NMT State Error	The device is in wrong NMT state

Table 18: Communication errors

9.2 Overview „EPOS Command Library" specified Errors

9.2.1 General Errors

Code	Name	Error cause
0x0000 0000	No Error	The function was successful
0x1000 0001	Internal Error	Internal Error
0x1000 0002	Null Pointer	Null Pointer passed to function
0x1000 0003	Handle not Valid	Handle passed to function is not valid
0x1000 0004	Bad Virtual Device Name	Virtual Device name is not valid
0x1000 0005	Bad Device Name	Device name is not valid
0x1000 0006	Bad ProtocolStack Name	ProtocolStack name is not valid
0x1000 0007	Bad Interface Name	Interface name is not valid
0x1000 0008	Bad Port Name	Port is not valid
0x1000 0009	Library not Loaded	Could not load external library
0x1000 000A	Executing Command	Command failed
0x1000 000B	Timeout	Timeout occurred during execution
0x1000 000C	Bad Parameter	Bad Parameter passed to function
0x1000 000D	Command Aborted By User	Command aborted by user

Table 19: „Command DLL" specified general errors

9.2.2 Interface Layer Errors

Code	Name	Error cause
0x2000 0001	Opening Interface	Error opening interface
0x2000 0002	Closing Interface	Error closing interface
0x2000 0003	Interface not Open	Interface is not open
0x2000 0004	Opening Port	Error opening port
0x2000 0005	Closing Port	Error closing port
0x2000 0006	Port not Open	Port is not open
0x2000 0007	Reset Port	Error resetting port
0x2000 0008	Set Port Settings	Error configuring port settings
0x2000 0009	Set Port Mode	Error configuring port mode

Table 20: „Command DLL" specified interface layer errors

9.2.3 Interface Layer 'RS232' Errors

Code	Name	Error cause
0x2100 0001	Write Data	Error writing data
0x2100 0002	Read Data	Error reading data

Table 21: „Command DLL" specified interface layer 'RS232' errors

9.2.4 Interface Layer 'CAN' Errors

Code	Name	Error cause
0x2200 0001	Receive CAN Frame	Error receiving CAN frame
0x2200 0002	Transmit CAN Frame	Error transmitting CAN frame

Table 22: „Command DLL" specified interface layer 'CAN' errors

9.2.5 Interface Layer 'USB' Errors

Code	Name	Error cause
0x2300 0001	Write Data	Error writing data
0x2300 0002	Read Data	Error reading data

Table 23: „Command DLL" specified interface layer 'USB' errors

9.2.6 Protocol Layer 'MaxonRS232' Errors

Code	Name	Error cause
0x3100 0001	NegAckReceived	Negative acknowledge received
0x3100 0002	BadCrcReceived	Bad checksum received
0x3100 0003	BadDataSizeReceived	Bad data size received

Table 24: „Command DLL" specified protocol 'MaxonRS232' errors

9.2.7 Protocol Layer 'CANopen' Errors

Code	Name	Error cause
0x3200 0001	SdoReceiveFrameNotReceived	CAN frame of SDO protocol not received
0x3200 0002	RequestedCanFrameNotReceived	Requested CAN frame not received
0x3200 0003	CanFrameNotReceived	Can frame not received

Table 25: „Command DLL" specified protocol 'CANopen' errors

9.2.8 Protocol Layer 'USB' Errors

Code	Name	Error cause
0x3300 0001	Stuffing	Failed Stuffing Data
0x3300 0002	Destuffing	Failed Destuffing Data
0x3300 0003	BadCrcReceived	Bad CRC received
0x3300 0004	BadDataSizeReceived	Bad Data received

Table 26: „Command DLL" specified protocol 'USB' errors

10 Version History

Date	DLL Version	Documentation	Description
11.11.2003	1.00	Edition November 2003	• First library version
01.12.2003	1.01	Edition December 2003	• All selection functions have been changed: VCS_GetBaudrateSelection(..), VCS_GetDeviceName(..), VCS_GetDeviceNameSelection(..), VCS_GetDriverInfo(..), VCS_GetInterfaceName(..), VCS_GetInterfaceNameSelection(..), VCS_GetPortName(..), VCS_GetPortNameSelection(..), VCS_GetProtocolStackModeSelection(..), VCS_GetProtocolStackName(..), VCS_GetProtocolStackNameSelection(..)
05.01.2004	1.02	Edition January 2004	• Insert IXXAT details
06.04.2004	2.0.0.0	Edition April 2004	• New functions documented: VCS_CloseAllDevices(..), VCS_DigitalInputConfiguration(..), VCS_DigitalOutputConfiguration(..), VCS_GetAllDigitalInputs(..), VCS_GetAllDigitalOutputs(..), VCS_GetAnalogInput(..), VCS_SetAllDigitalOutputs(..), VCS_SendNMTService(..), VCS_OpenDeviceDlg(..) • All this functions have been changed: VCS_GetBaudrateSelection(..), VCS_FindHome(..), VCS_GetHomingParameter(..), VCS_SetHomingParameter(..), VCS_MoveToPosition(..), VCS_GetOperationMode(..), VCS_SetOperationMode(..), VCS_GetObject(..), VCS_SetObject(..) • All this functions have been deleted: VCS_GetProtocolStackMode(..), VCS_GetProtocolStackModeSelection(..)
16.07.2004	2.0.3.0	Edition July 2004	• Error correction documentation • Additional information about error codes
01.03.2005	3.0.0.0	Edition March 2005	• Insert from Vector CAN cards details
01.10.2005	4.0.0.0	Edition October 2005	• Error correction documentation
03.02.2006	4.0.0.0	Edition February 2006	• Additional information about error codes
12.04.2006	4.1.0.0	Edition April 2006	• New error codes
12.04.2006	4.1.1.0	Edition April 2006	• VCS_SendCANFrame bug fixed
11.10.2006	4.2.0.0	Edition October 2006	• New function: VCS_GetErrorInfo(..)
16.10.2006	4.2.1.0	Edition October 2006	• VCS_GetDriverInfo, VCS_SetHomingParameter bug fixed
01.02.2007	4.3.0.0	Edition January 2007	• Support for National Instruments Interfaces
10.08.2007	4.4.0.0	Edition August 2007	• Support for IXXAT VCI V3
01.05.2008	4.5.0.0	Edition April 2008	• New functions for read device errors (Get Device Error) • Adaption for EPOS2
04.09.2009	4.6.0.0	Edition September 2009	• Support for EPOS2 functionality • Support for data recorder • Support for parameter export and import • VCS_ReadCANFrame
22.10.2009	4.6.1.3	Edition October 2009	• Multithreading bug fixed
30.08.2009	4.7.1.0	Edition August 2010	• New Parameter “DialogMode” for Findxxx Functions • New ProtocolStack Name “MAXON SERIAL V2” (Library is still compatible with old name “EPOS2_USB”) • VCS_WaitForTargetReached returns false, if timeout elapses

11.10.2010	4.7.2.0	Edition October 2010	• Deadlock when closing Application fixed • Communication for IXXAT VCI V3.3 fixed
28.10.2010	4.7.3.0	Edition November 2010	• Bugfix “VCS_CloseDevice”, “VCS_CloseAllDevices”
28.01.2011	4.8.1.0	Edition January 2011	• Expand to 64-Bit Windows and 32-Bit Linux • Bugfix Segmented Write
02.02.2011	4.8.2.0	Edition February 2011	• Bugfix NI-LIN device
10.04.2012	4.8.5.0	Edition April 2012	• Bugfix Sporadic CAN failure with IXXAT VCI V3.3
08.10.2012	4.8.6.0	Edition October 2012	• CANopen Vector Interface support for VN1600 series
10.10.2012	4.8.7.0	Edition October 2012	• Bugfix command Send NMT Service • New Functions: VCS_GetVelocityRegulatorFeedForward, VCS_SetVelocityRegulatorFeedForward
04.01.2013	4.9.1.0	Edition December 2012	• New Functions: VCS_GetHomingState, VCS_WaitForHomingAttained, VCS_GetVelocityIsAveraged, VCS_GetCurrentIsAveraged

LIST OF FIGURES

LIST OF TABLES