maxon motor

maxon motor control EPOS Positioning Controller

Windows 32-Bit DLL Edition January 2007

EPOS

Positioning Controller

Documentation

Windows 32-Bit DLL

maxon document number: 573057 - 12

1 Table of contents

2 Table of figures

3 Table of tables

4 Introduction

This documentation „Windows 32-Bit DLL" provides the instructions for the implemented functions.
The library is arranged in groups of functions and helps to simplify the programming of the control software based on Windows.

This document describes the interface between a program and the Windows DLL (Dynamic Link Library). This DLL supports devices, which are attached to a **serial interface RS232** or to a **CAN board by IXXAT, Vector or National Instruments**. All other CANopen products of other manufacturers can also be used, however no motion control library is available.
The Windows DLL supports the SDO protocol from CANopen but the Windows DLL is not qualified to real-time communication.

Additional information are available in the following documents:

- „DLL Integration into MS VC++"
  The document „DLL Integration into MS VC++" describes detailed the implementation and structure for **Microsoft Visual C++**.
- „DLL Integration into MS Visual Basic"
  The document „DLL Integration into MS Visual Basic" describes detailed the implementation and structure for **Microsoft Visual Basic**.
- „DLL Integration into Borland C++"
  The document „DLL Integration into Borland C++" describes detailed the implementation and structure for **Borland C++ Builder**.
- „DLL Integration into Borland Delphi"
  The document „DLL Integration into Borland Delphi" describes detailed the implementation and structure for **Borland Delphi**.
- „DLL Integration into LabVIEW"
  The document „DLL Integration into LabVIEW" describes detailed the implementation and structure for **National Instruments LabVIEW**.
- „DLL Integration into National Instruments LabWindows/CVI"
  The document „DLL Integration into National Instruments LabWindows/CVI" describes detailed the implementation and structure for **National Instruments LabWindows/CVI**.

To several high-level languages, an appropriate example with a suitable documentation will be present.

This library is intended to cover most applications in automatisation.
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 „Windows 32-Bit DLL", additional documentation and software to the EPOS positioning controller may also be found in the internet under www.maxonmotor.com category <Service & Downloads>.

5 Third party products

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

5.1 IXXAT

CANopen boards from IXXAT operated with an universal driver VCI V2. The Windows DLL works with this universal driver VCI V2 (Version 2.14 and greater) from IXXAT.

See addresses below for ordering CANopen boards.

Distributors

- www.ixxat.de subdirectory <contact>

5.2 Vector

For use of Vector CANopen cards, the 'XL Driver Library' is needed. The latest edition of this 'XL Driver Library' may also be found in the internet under http://www.vector-informatik.de/english/ category <Hardware Products> subdirectory <Interfaces>. The library has to be installed manually in the appropriate working 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.

Distributors

- General distributors

5.3 National Instruments

CAN Interfaces of National Instruments are supported. The NI-CAN Software and Hardware has to be installed.

See address below for ordering CAN boards.

Contact

- www.ni.com/can

6 How to use this guide

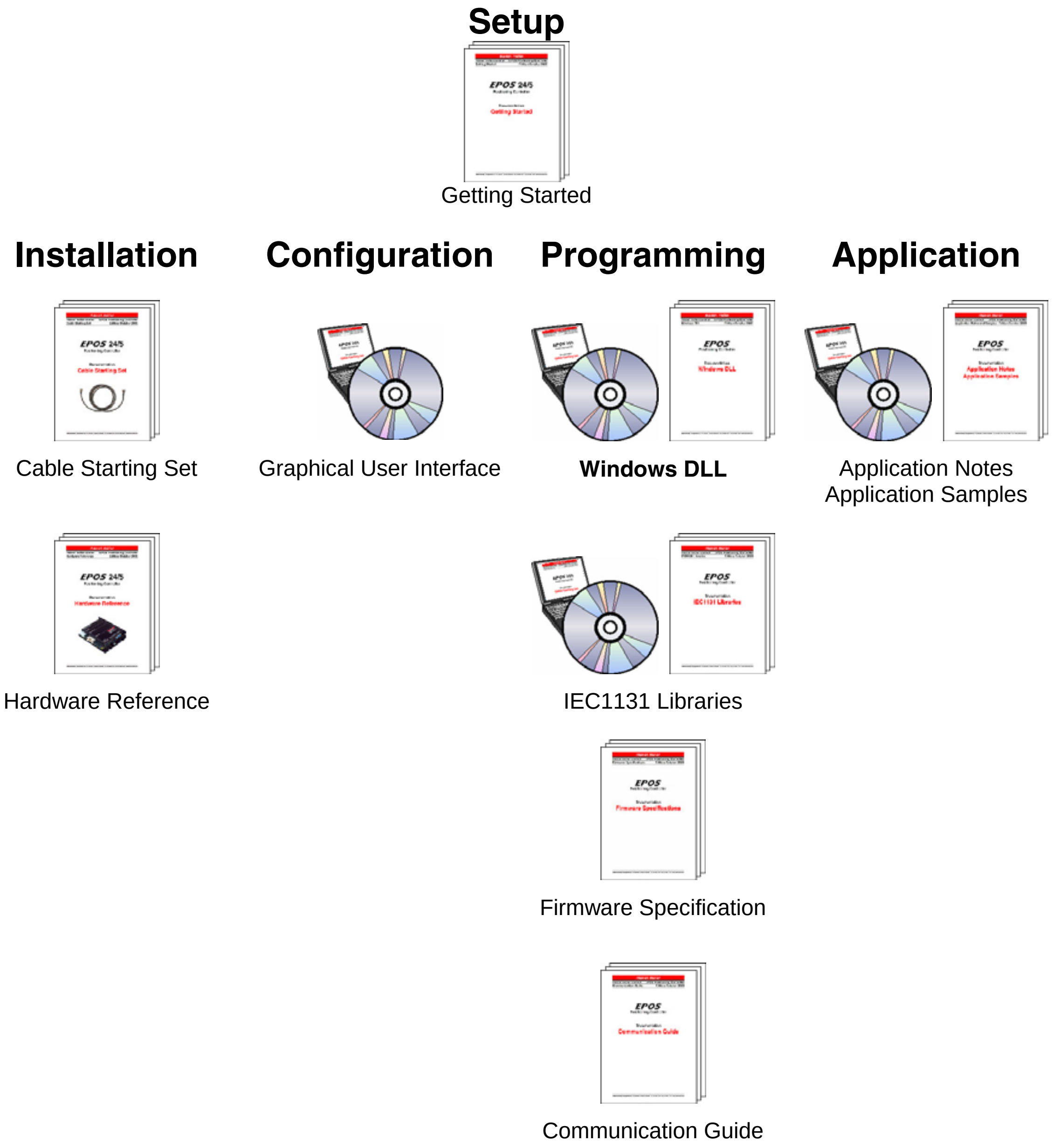

Figure 1: EPOS documentation hierarchy

7 Virtual Command Set Communication

The Virtual Command Set Communication defines following groups:

Initialisation
Help Functions

7.1 Communication Structure

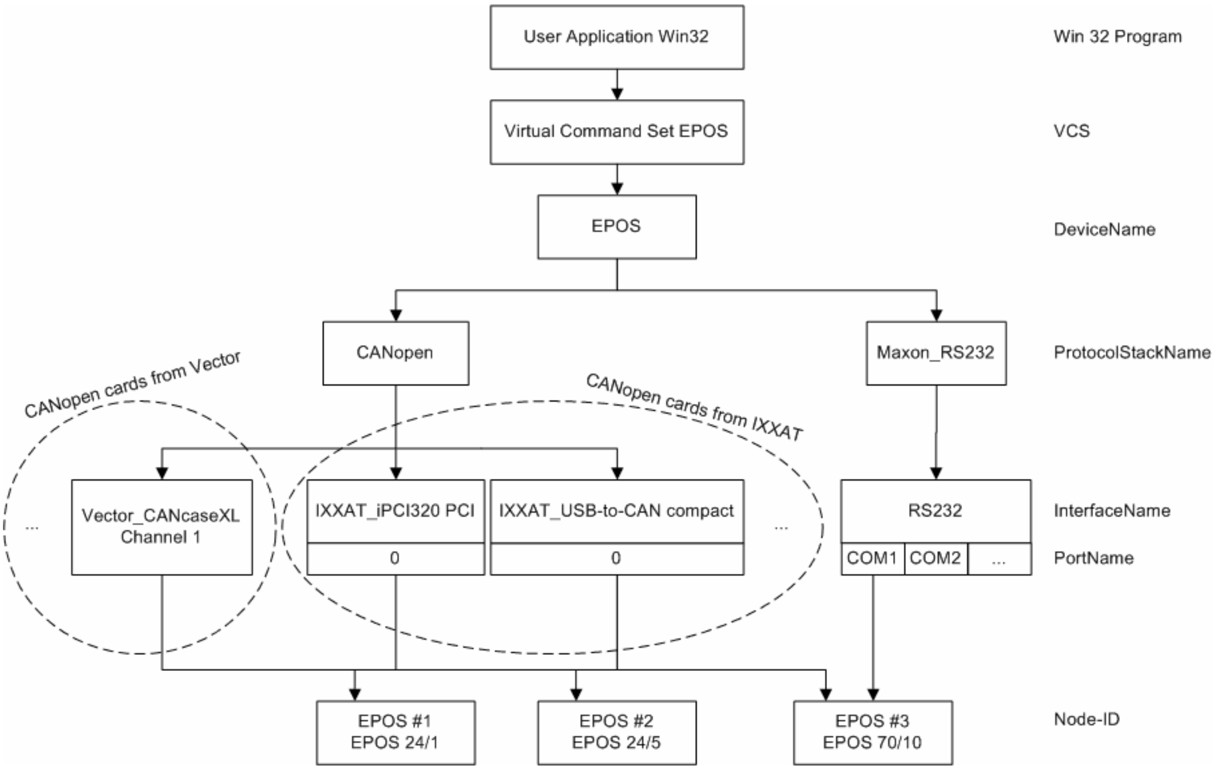

Figure 2: Communication structure

7.2 Data Type Definitions

The „Windows 32-Bit DLL" is implemented with Microsoft Visual C++. Below is a table of all used data types.

Name	Data type	Size bits	Size bytes	Range
char	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	boolean value	32	4	TRUE or FALSE
HANDLE	pointer to an object	32	4	0 ... 4'294'967'295

Table 1: Data type definitions

7.3 Initialisation

This group defines all required functions to initialize a correct communication:

Open Device
Close All Devices
Close Device
Get Protocol Stack Settings
Set Protocol Stack Settings

7.3.1 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 and with CANopen boards.
For exact names for 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
ProtocolStack-Name	char*	Name of used bus system: MAXON_RS232 or CANopen
InterfaceName	char*	Name of used interface: RS232: RS232 CANopen: Is composed of: Manufacturer_BoardName DeviceNumber *Examples:* IXXAT_USB-to-CAN compact 0, IXXAT_iPCI320 PCI 0, … Vector_CANcaseXL Channel 1, .. NI_PCI-CAN 0, …
PortName	char*	Name of port: RS232: COM1, COM2 … CANopen: 0, 1 …

Return Parameters

pErrorCode	DWORD	Error information
Return Value	HANDLE	Handle for port access. Nonzero if successful; otherwise 0

Related Functions
Open Device Dialog

7.3.2 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
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Open Device

7.3.3 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
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Open Device

7.3.4 Get Protocol Stack Settings

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

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

Parameter

KeyHandle	HANDLE	Handle for port access

Return Parameters

pBaudrate	DWORD	Actual baudrate from opened port [Bit/s]
pTimeout	DWORD	Actual timeout from opened port [ms]
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

7.3.5 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 baudrate use the function Get Baudrate Selection.

Parameters

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

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

7.4 Help Functions

This group defines all required functions to get additional information:

Get Baudrate Selection
Get Device Name
Get Device Name Selection
Get Driver Info
Get Interface Name
Get Interface Name Selection
Get Key Handle
Get Port Name
Get Port Name Selection
Get Protocol Stack Mode
Get Protocol Stack Mode Selection
Get Protocol Stack Name
Get Protocol Stack Name Selection

7.4.1 Get Baudrate 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
ProtocolStack-Name	char*	Name of protocol stack
InterfaceName	char*	Interface name
PortName	char*	Port name
StartOfSelection	BOOL	True: Get first selection value False: Get next selection value

Return Parameters

pBaudrateSel	DWORD	Pointer to baudrate [Bit/s]
pEndOfSelection	BOOL	True: No more value available False: More value available
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

7.4.2 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
Return Value	BOOL	Nonzero if successful; otherwise 0

7.4.3 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	True: No more string available False: More string available
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

7.4.4 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 Windows DLL.

Parameters

MaxStrName-Size	WORD	Reserved memory size for the name
MaxStrVersion-Size	WORD	Reserved memory size for the version

Return Parameters

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

7.4.5 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*	Pointer to interface name
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

7.4.6 Get Error Info

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

Description
Function „VCS_GetErrorInfo" returns the error information from a received error code.

Parameters

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

Return Parameters

pErrorInfo	char*	Error string
Return Value	BOOL	Nonzero if successful; otherwise 0

7.4.7 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
ProtocolStack-Name	char*	Protocol stack name
StartOfSelection	BOOL	True: Get first selection string False: Get next selection string
MaxStrSize	WORD	Reserved memory size for the interface name

Return Parameters

pInterfaceName-Sel	char*	Pointer to interface name
pEndOfSelection	BOOL	True: No more string available False: More string available
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

7.4.8 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.

Parameters

DeviceName	char*	Device name
ProtocolStack-Name	char*	Protocol stack name
InterfaceName	char*	Interface name
PortName	char*	Port name

Return Parameters

pKeyHandle	HANDLE	Handle for port access
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

7.4.9 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
Return Value	BOOL	Nonzero if successful; otherwise 0

7.4.10 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
ProtocolStack-Name	char*	Protocol stack name
InterfaceName	char*	Interface name
StartOfSelection	BOOL	True: Get first selection string False: Get next selection string
MaxStrSize	WORD	Reserved memory size for the port name

Return Parameters

pPortSel	char*	Pointer to port name
pEndOfSelection	BOOL	True: No more string available False: More string available
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Go to Programming Example

7.4.11 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

pProtocolStack-Name	char*	Pointer to protocol stack name
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

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

Return Parameters

pProtocolStack-NameSel	char*	Pointer to available protocol stack name
pEndOfSelection	BOOL	True: No more string available False: More string available
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

7.4.13 Programming Example

The example shows how to read all protocol stack names of the available interfaces.

```
WORD wMaxStrSize = 100;
char* strDeviceName = “EPOS”;
char* strProtocolStackName = (char*)malloc(wMaxStrSize);
BOOL oEndOfSel;
DWORD dErrrorCode;

// get first protocol stack name
if(VCS_GetProtocolStackNameSelection(strDeviceName, true,
strProtocolStackName, wMaxStrSize, &oEndOfSel, &dErrrorCode))
{
   // get next protocol stack name (as long as oEndOfSel == false)
   while(!oEndOfSel)
   {
      VCS_GetProtocolStackNameSelection (strDeviceName, false,
      strProtocolStackName, wMaxStrSize, &oEndOfSel, &dErrrorCode)
   }
}
```

8 Virtual Command Set EPOS

The Virtual Command Set EPOS defines following groups:

8.1 Configuration

This group defines all required functions for device configuration:

8.1.1 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

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

Related Functions
Set Current Regulator Gain

8.1.2 Get Encoder Parameter

Function
BOOL **VCS_GetEncoderParameter** (HANDLE KeyHandle, WORD NodeId, WORD *pCounts, WORD *pPositionSensorType, DWORD *pErrorCode)

Description
With function „VCS_GetEncoderParameter" it is possible to read all encoder parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pCounts	WORD	Incremental encoder counts [pulse per turn]	Object: 0x2210-01
pPosition-SensorType	WORD	Position sensor type	Object: 0x2210-02
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Set Encoder Parameter

8.1.3 Get Motor Parameter

Function
BOOL **VCS_GetMotorParameter** (HANDLE KeyHandle, WORD NodeId, WORD *pMotorType, WORD *pContinuousCurrent, WORD *pPeakCurrent, BYTE *pPolePair, WORD *pThermalTimeConstant, DWORD *pErrorCode)

Description
With function „VCS_GetMotorParameter" it is possible to read all motor parameters.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pMotorType	WORD	Kind of motor	Object: 0x6402-00
pContinuous-Current	WORD	Maximal continuous current [mA]	Object: 0x6410-01
pPeakCurrent	WORD	Maximal peak current [mA]	Object: 0x6410-02
pPolePair	BYTE	Number of pole pairs	Object: 0x6410-03
pThermalTime-Constant	WORD	Thermal time constant Winding [s]	Object: 0x6410-05
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Set Motor Parameter

8.1.4 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	Identification ID of the addressed device (is given by hardware switches)

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	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Set Position Regulator Gain

8.1.5 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pP	WORD	Velocity regulator P-Gain	Object: 0x60F9-01
pI	WORD	Velocity regulator I-Gain	Object: 0x60F9-02
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Set Velocity Regulator Gain

8.1.6 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_UserInterface.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
P	WORD	Current regulator P-Gain	Object: 0x60F6-01
I	WORD	Current regulator I-Gain	Object: 0x60F6-02

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Current Regulator Gain

8.1.7 Set Encoder Parameter

Function
BOOL **VCS_SetEncoderParameter** (HANDLE KeyHandle, WORD NodeId, WORD Counts, WORD PositionSensorType, DWORD *pErrorCode)

Description
With function „VCS_SetEncoderParameter" it is possible to write all encoder parameters.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
Counts	WORD	Incremental Encoder counts [pulse per turn]	Object: 0x2210-01
PositionSensor-Type	WORD	Position Sensor Type	Object: 0x2210-02

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Encoder Parameter

8.1.8 Set Motor Parameter

Function
BOOL **VCS_SetMotorParameter** (HANDLE KeyHandle, WORD NodeId, WORD MotorType, WORD ContinuousCurrent, WORD PeakCurrent, BYTE PolePair, WORD ThermalTimeConstant, DWORD *pErrorCode)

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

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
MotorType	WORD	Kind of motor	Object: 0x6402-00
Continuous-Current	WORD	Maximal continuous current [mA]	Object: 0x6410-01
PeakCurrent	WORD	Maximal peak current [mA]	Object: 0x6410-02
PolePair	BYTE	Number of pole pairs	Object: 0x6410-03
ThermalTime-Constant	WORD	Thermal time constant Winding [s]	Object: 0x6410-05

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Motor Parameter

8.1.9 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_UserInterface.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
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
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Position Regulator Gain

8.1.10 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_UserInterface.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
P	WORD	Velocity regulator P-Gain	Object: 0x60F9-01
I	WORD	Velocity regulator I-Gain	Object: 0x60F9-02

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Velocity Regulator Gain

8.2 Current Mode

This group defines all required functions for current mode:

Get Current Must
Set Current Must

8.2.1 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 demand value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pCurrentMust	short	Current mode demand value [mA]	Object: 0x2030-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Set Current Must

8.2.2 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 demand value.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
CurrentMust	short	Current mode demand value [mA]	Object: 0x2030-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Current Must

8.3 Homing Mode

This group defines all required functions for homing mode:

Find Home
Get Homing Parameter
Set Homing Parameter
Stop Homing

8.3.1 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	Identification ID of the addressed device (is given by hardware switches)	
HomingMethod	__int8	Homing method	Object: 0x6098-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Method Number	Description
35	Actual Position
34	Index Positive Speed
33	Index Negative Speed
27	Home Switch Negative Speed
23	Home Switch Positive Speed
18	Positive Limit Switch
17	Negative Limit Switch
11	Home Switch Negative Speed & Index
7	Home Switch Positive Speed & Index
2	Positive Limit Switch & Index
1	Negative Limit Switch & Index
0	No homing operation required
-1	Current Threshold Positive Speed & Index
-2	Current Threshold Negative Speed & Index
-3	Current Threshold Positive Speed
-4	Current Threshold Negative Speed

Table 2: Homing methods

Related Functions
Set Homing Parameter
Stop Homing

8.3.2 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pHoming-Acceleration	DWORD	Acceleration for homing profile [rpm/s]	Object: 0x609A-00
pSpeedSwitch	DWORD	Speed during search for switch [rpm]	Object: 0x6099-01
pSpeedIndex	DWORD	Speed during search for index signal [rpm]	Object: 0x6099-02
pHomeOffset	long	Home offset after homing [qc]	Object: 0x607C-00
pCurrent-Threshold	WORD	Current threshold for homing method -1, -2, -3 and -4 [mA]	Object: 0x2080-00
pHomePosition	long	Home position value [qc]	Object: 0x2081-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions

Find Home
Stop Homing
Set Homing Parameter

8.3.3 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	Identification ID of the addressed device (is given by hardware switches)	
Homing-Acceleration	DWORD	Acceleration for homing profile [rpm/s]	Object: 0x609A-00
SpeedSwitch	DWORD	Speed during search for switch [rpm]	Object: 0x6099-01
SpeedIndex	DWORD	Speed during search for index signal [rpm]	Object: 0x6099-02
HomeOffset	long	Home offset after homing [qc]	Object: 0x607C-00
Current-Threshold	WORD	Current threshold for homing method -1, -2, -3 and -4 [mA]	Object: 0x2080-00
HomePosition	Long	Assign the current Homing position with this value [qc]	Object: 0x2081-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Find Home
Stop Homing
Get Homing Parameter

8.3.4 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Find Home
Set Homing Parameter

8.4 Inputs Outputs

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

[Digital Input Configuration]
[Digital Output Configuration]
[Get All Digital Inputs]
[Get All Digital Outputs]
[Get Analog Input]
[Set All Digital Outputs]

Digital Input Overview
The number of supported digital inputs depend on hardware (EPOS 24/1 and EPOS 24/5 and MCD EPOS 60 W have six digital inputs; EPOS 70/10 supports eight digital inputs). There are some predefined functions for digital inputs like home switch, limit switches, position marker and also some general purpose inputs for general purpose process inputs.

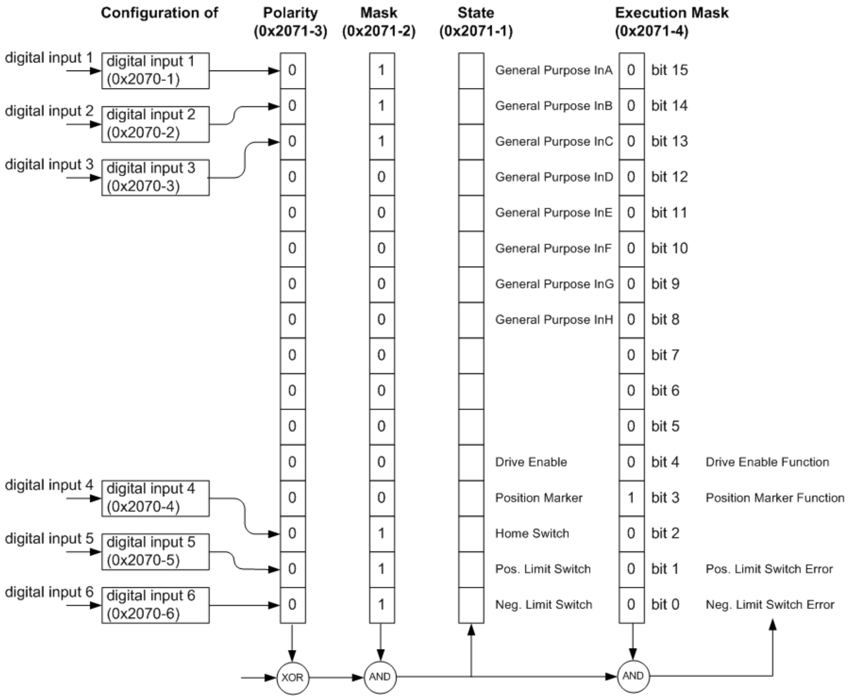

Figure 3: Digital Input Functionality EPOS 24/1 and EPOS 24/5 Overview (default configuration)

Remark
More information about the inputs from the other devices (i. e. EPOS 70/10 and MCD EPOS) are available in the document EPOS Firmware Specification!

Digital Outputs Overview
There is a predefined function for digital output: Ready/Fault. If an output is configured with this function then a hardware signal is available if a fault occurs or not. There are also some general purpose outputs for general process controlling for example lighting a lamp. The configuration is similar to the digital inputs configuration.

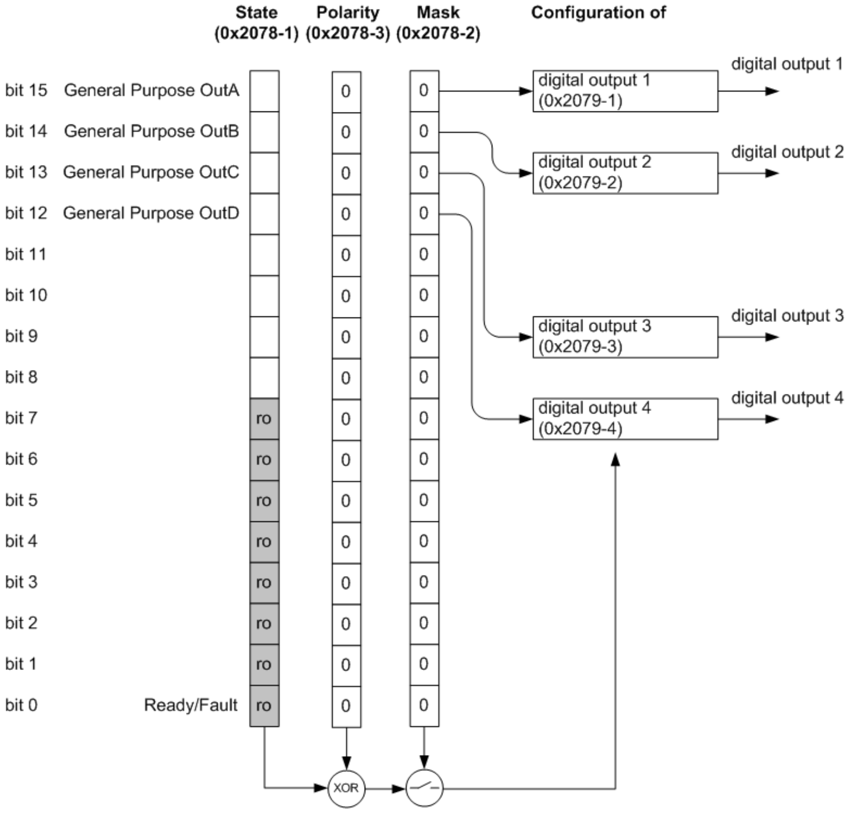
State (0x2078-1)
Polarity (0x2078-3)
Mask (0x2078-2)
Configuration of
bit 15 General Purpose OutA
bit 14 General Purpose OutB
bit 13 General Purpose OutC
bit 12 General Purpose OutD
bit 11
bit 10
bit 9
bit 8
bit 7
bit 6
bit 5
bit 4
bit 3
bit 2
bit 1
bit 0 Ready/Fault
ro
0
digital output 1 (0x2079-1)
digital output 2 (0x2079-2)
digital output 3 (0x2079-3)
digital output 4 (0x2079-4)
digital output 1
digital output 2
digital output 3
digital output 4
XOR

Figure 4: Digital Output Functionality EPOS 24/5 and EPOS 70/10 Overview (default configuration)

Remark
More information about the inputs from the other devices (i. e. EPOS 24/1 and MCD EPOS) are available in the document EPOS Firmware Specification!

8.4.1 Digital Input Configuration

Function

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

Description

„VCS_DigitalInputConfiguration" sets the parameter for a digital input.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
DigInputNb	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)	
Mask	BOOL	1: Functionality state will be displayed 0: not displayed	Object: 0x2071-02
Polarity	BOOL	1: Low level 0: High level	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
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions

Get All Digital Inputs

8.4.2 Digital Output Configuration

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

Description
„VCS_DigitalOutputConfiguration" sets parameter for a digital output.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
DigOutputNb	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)	
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
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get All Digital Outputs
Set All Digital Outputs

8.4.3 Get All Digital Inputs

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

Description
„VCS_GetAllDigitalInputs" returns state of all digital inputs.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pInput	WORD	State of all digital inputs	Object: 0x2071-01
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Digital Input Configuration

8.4.4 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pOutputs	WORD	State of all digital outputs	Object: 0x2078-01
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Digital Output Configuration
Set All Digital Outputs

8.4.5 Get Analog Input

Function
BOOL **VCS_GetAnalogInput** (HANDLE KeyHandle, WORD NodeId, WORD Number, WORD *pAnalog, DWORD *pErrorCode)

Description
„VCS_GetAnalogInput" returns the analog value from input 1 or 2.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)
Number	WORD	Analog input number 1 or 2

Return Parameters

pAnalog	WORD	Analog value from input [mV]	Object: 0x207C-01 or Object: 0x207C-02
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

8.4.6 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	Identification ID of the addressed device (is given by hardware switches)	
Outputs	WORD	State of all digital outputs	Object: 0x2078-01

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Digital Output Configuration
Get All Digital Outputs

8.5 Motion Info

This group defines all required functions for motion information:

Get Current Is
Get Movement State
Get Position Is
Get Velocity Is

8.5.1 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pCurrentIs	short	Current actual value [mA]	Object: 0x6078-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Movement State
Get Position Is
Get Velocity Is

8.5.2 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pTarget-Reached	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	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Current Is
Get Position Is
Get Velocity Is

8.5.3 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pPositionIs	long	Position actual value [qc]	Object: 0x6064-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Current Is
Get Movement State
Get Velocity Is
Get Position Must
Set Position Must

8.5.4 Get Velocity Is

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

Description
„VCS_GetVelocityIs" read the velocity actual value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pVelocityIs	long	Velocity actual value [rpm]	Object: 0x606C-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Current Is
Get Movement State
Get Position Is

8.6 Position Mode

This group defines all required functions for position mode:

Get Position Must
Set Position Must

8.6.1 Get Position Must

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

Description
„VCS_GetPositionMust" returns the position demand value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pPositionMust	long	Position demand value [qc]	Object: 0x2062-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Position Is

8.6.2 Set Position Must

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

Description
„VCS_SetPositionMust" sets the position demand value.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
PositionMust	long	Position demand value [qc]	Object: 0x2062-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Position Is

8.7 Profile Position Mode

This group defines all required functions for profile position mode:

Get Position Profile
Get Target Position
Halt Position Movement
Move To Position
Set Position Profile

8.7.1 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pProfileVelocity	DWORD	Position profile velocity [rpm]	Object: 0x6081-00
pProfile-Acceleration	DWORD	Position profile acceleration [rpm/s]	Object: 0x6083-00
pProfile-Deceleration	DWORD	Position profile deceleration [rpm/s]	Object: 0x6084-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Target Position
Halt Position Movement
Move To Position
Set Position Profile

8.7.2 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pTargetPosition	long	Target position [qc]	Object: 0x607A-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Position Profile
Halt Position Movement
Move To Position
Set Position Profile

8.7.3 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Position Profile
Get Target Position
Move To Position
Set Position Profile

8.7.4 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	Identification ID of the addressed device (is given by hardware switches)	
TargetPosition	long	Target position [qc]	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
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Position Profile
Get Target Position
Halt Position Movement
Set Position Profile

8.7.5 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	Identification ID of the addressed device (is given by hardware switches)	
ProfileVelocity	DWORD	Position profile velocity [rpm]	Object: 0x6081-00
Profile-Acceleration	DWORD	Position profile acceleration [rpm/s]	Object: 0x6083-00
Profile-Deceleration	DWORD	Position profile deceleration [rpm/s]	Object: 0x6084-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Position Profile
Get Target Position
Halt Position Movement
Move To Position

8.8 Profile Velocity Mode

This group defines all required functions for profile velocity mode:

Get Target Velocity
Get Velocity Profile
Halt Velocity Movement
Move With Velocity
Set Velocity Profile

8.8.1 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pTargetVelocity	long	Target velocity [rpm]	Object: 0x60FF-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Velocity Profile
Halt Velocity Movement
Move With Velocity
Set Velocity Profile

8.8.2 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pProfile-Acceleration	DWORD	Velocity profile acceleration [rpm/s]	Object: 0x6083-00
pProfile-Deceleration	DWORD	Velocity profile deceleration [rpm/s]	Object: 0x6084-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Target Velocity
Halt Velocity Movement
Move With Velocity
Set Velocity Profile

8.8.3 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Target Velocity
Get Velocity Profile
Move With Velocity
Set Velocity Profile

8.8.4 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	Identification ID of the addressed device (is given by hardware switches)	
TargetVelocity	long	Target velocity [rpm]	Object: 0x60FF-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Target Velocity
Get Velocity Profile
Halt Velocity Movement
Set Velocity Profile

8.8.5 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	Identification ID of the addressed device (is given by hardware switches)	
Profile-Acceleration	DWORD	Velocity profile acceleration [rpm/s]	Object: 0x6083-00
Profile-Deceleration	DWORD	Velocity profile deceleration [rpm/s]	Object: 0x6084-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Target Velocity
Get Velocity Profile
Halt Velocity Movement
Move With Velocity

8.9 State Machine

For detailed information how the state machine functions refer to document „EPOS Firmware Specification”.

This group defines all required function blocks for device state machine:

Clear Fault
Get Disable State
Get Enable State
Get Fault State
Get Operation Mode
Get Quick Stop State
Set Disable State
Set Enable State
Set Operation Mode
Set Quick Stop State
Send NMT Service

8.9.1 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Disable State
Get Enable State
Get Fault State
Get Quick Stop State
Set Disable State
Set Enable State
Set Quick Stop State

8.9.2 Get Disable State

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

Description
The function „VCS_GetDisableState" returns the device state disable (pIsDisabled = TRUE).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pIsDisabled	BOOL	Device disable state
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Clear Fault
Get Enable State
Get Fault State
Get Quick Stop State
Set Disable State
Set Enable State
Set Quick Stop State

8.9.3 Get Enable State

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

Description
The function „VCS_GetEnableState" returns the device state enable (pIsEnabled = TRUE).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pIsEnabled	BOOL	Device enable state
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Clear Fault
Get Disable State
Get Fault State
Get Quick Stop State
Set Disable State
Set Enable State
Set Quick Stop State

8.9.4 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pIsInFault	BOOL	Device fault state
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Remarks
Get error information by read out the error history (use Get Object with index 0x1005, SubIndex 0x01 et sqq.)

Related Functions
Clear Fault
Get Disable State
Get Enable State
Get Quick Stop State
Set Disable State
Set Enable State
Set Quick Stop State

8.9.5 Get Operation Mode

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

Description
„VCS_GetOperationMode" returns the operation mode:

Value	***Mode***
6 (06h)	Homing Mode
3 (03h)	Profile Velocity Mode
1 (01h)	Profile Position Mode
-1 (FFh)	Position Mode
-2 (FEh)	Velocity Mode
-3 (FDh)	Current Mode
-5 (FBh)	Master Encoder Mode
-6 (FAh)	Step/Direction Mode

Table 3: Operation modes

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pMode	__int8	Operation Mode	Object: 0x6061-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Set Operation Mode

8.9.6 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 (pIsQuickStopped = TRUE).

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pIsQuick-Stopped	BOOL	Device quick stop state
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Clear Fault
Get Disable State
Get Enable State
Get Fault State
Set Disable State
Set Enable State
Set Quick Stop State

8.9.7 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Clear Fault
Get Disable State
Get Enable State
Get Fault State
Get Quick Stop State
Set Enable State
Set Quick Stop State

8.9.8 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Clear Fault
Get Disable State
Get Enable State
Get Fault State
Get Quick Stop State
Set Disable State
Set Quick Stop State

8.9.9 Set Operation Mode

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

Description
„VCS_SetOperationMode" sets the operation mode. Mode can have the following values:

Mode	***Value***
Homing Mode	6 (06h)
Profile Velocity Mode	3 (03h)
Profile Position Mode	1 (01h)
Position Mode	-1 (FFh)
Velocity Mode	-2 (FEh)
Current Mode	-3 (FDh)
Master Encoder Mode	-5 (FBh)
Step/Direction Mode	-6 (FAh)

Table 4: Operation modes

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
Mode	__int8	Operation Mode	Object: 0x6060-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Operation Mode

8.9.10 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Clear Fault
Get Disable State
Get Enable State
Get Fault State
Get Quick Stop State
Set Disable State
Set Enable State

8.9.11 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	Identification ID of the addressed device (is given by hardware switches)
Command-Specifier	WORD	Following NMT services are available: 1 Start Remote Node 2 Stop Remote Node 128 Enter Pre-Operational 129 Reset Node 130 Reset Communication

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

8.10 Utilities

This group defines all functions, which do not fall in the other groups:

Get Object
Get Version
Restore
Set Object
Store

8.10.1 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" read an object value at the given index and sub index from object dictionary.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)
ObjectIndex	WORD	Object index
ObjectSubIndex	BYTE	Object sub index
NbOfBytesTo-Read	DWORD	Object length to read (number of bytes)

Return Parameters

pData	void	Object data
pNbOfBytes-Read	DWORD	Object length read (number of bytes)
pErrorCode	DWORD	Error information
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 document EPOS Firmware Specification.

Related Functions
Get Version
Restore
Set Object
Store

8.10.2 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pHardware-Version	WORD	Hardware version	Object: 0x2003-01
pSoftware-Version	WORD	Software version	Object: 0x2003-02
pApplication-Number	WORD	Application number	Object: 0x2003-03
pApplication-Version	WORD	Application version	Object: 0x2003-04
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Object
Restore
Set Object
Store

8.10.3 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Object
Get Version
Set Object
Store

8.10.4 Set Object

Function
BOOL **VCS_SetObject** (HANDLE KeyHandle, WORD NodeId, WORD ObjectIndex, BYTE ObjectSubIndex, void Data, DWORD NbOfBytesToWrite, DWORD NbOfBytesWritten, 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	Identification ID of the addressed device (is given by hardware switches)
ObjectIndex	WORD	Object index
ObjectSubIndex	BYTE	Object sub index
Data	void	Object data
NbOfBytesTo-Write	DWORD	Object length to write (number of bytes)

Return Parameters

pNbOfBytes-Written	DWORD	Object length written (number of bytes)
pErrorCode	DWORD	Error information
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 document EPOS Firmware Specification.

Related Functions
Get Object
Get Version
Restore
Store

8.10.5 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	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Object
Get Version
Restore
Set Object

8.11 Velocity Mode

This group defines all required functions for velocity mode:

Get Velocity Must
Set Velocity Must

8.11.1 Get Velocity Must

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

Description
The function „VCS_GetVelocityMust" returns the position demand value.

Parameters

KeyHandle	HANDLE	Handle for port access
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)

Return Parameters

pVelocityMust	long	Velocity demand value [rpm]	Object: 0x206B-00
pErrorCode	DWORD	Error information	
Return Value	BOOL	Nonzero if successful; otherwise 0	

Related Functions
Get Velocity Is
Set Velocity Must

8.11.2 Set Velocity Must

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

Description
The function „VCS_SetVelocityMust" sets the velocity demand value.

Parameters

KeyHandle	HANDLE	Handle for port access	
NodeId	WORD	Identification ID of the addressed device (is given by hardware switches)	
VelocityMust	long	Velocity demand value [rpm]	Object: 0x206B-00

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Get Velocity Is
Get Velocity Must

8.12 General CAN Commands

8.12.1 Send CAN Frame

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

Description
Function „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 Code (DLC)
pData	void	CAN Frame Data

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

8.12.2 Request CAN Frame

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

Description
Function „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 Code (DLC)

Return Parameters

pData	void	CAN Frame Data
pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

9 Virtual Dialog Set Communication

The Virtual Dialog Set Communication defines following groups:

Initialisation

9.1 Initialisation

9.1.1 Open Device Dialog

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

Description
The function „VCS_OpenDeviceDlg" registers available interfaces (CAN and RS232) with which the EPOS can be operated and opened the selected interface for communication.

Return Parameters

pErrorCode	DWORD	Error information
Return Value	BOOL	Nonzero if successful; otherwise 0

Related Functions
Open Device

10 Error Overview

10.1 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 5: Communication errors

10.2 Overview „Windows 32-Bit DLL„ specified Errors

10.2.1 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	ExecutingCommand	Command failed
0x1000 000B	Timeout	Timeout occurred during execution
0x1000 000C	BadParameter	Bad Parameter passed to function
0x1000 000D	CommandAbortedByUser	Command aborted by user

Table 6: „Windows 32-Bit DLL" specified general errors

10.2.2 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	ResetPort	Error resetting port
0x2000 0008	SetPortSettings	Error configuring port settings
0x2000 0009	SetPortMode	Error configuring port mode

Table 7: „Windows 32-Bit DLL" specified interface layer errors

10.2.3 Interface Layer 'RS232' Errors

Code	Name	Error cause
0x2100 0001	WriteData	Error writing data
0x2100 0002	ReadData	Error reading data

Table 8: „Windows 32-Bit DLL" specified interface layer 'RS232' errors

10.2.4 Interface Layer 'CAN' Errors

Code	Name	Error cause
0x2200 0001	ReceiveCanFrame	Error receiving CAN Frame
0x2200 0002	TransmitCanFrame	Error transmitting CAN Frame

Table 9: „Windows 32-Bit DLL" specified interface layer 'CAN' errors

10.2.5 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 10: „Windows 32-Bit DLL" specified protocol 'MaxonRS232' errors

10.2.6 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

Table 11: „Windows 32-Bit DLL" specified protocol 'CANopen' errors

11 Table of functions

12 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: - BOOL VCS_GetBaudrateSelection(..) - BOOL VCS_GetDeviceName(..) - BOOL VCS_GetDeviceNameSelection(..) - BOOL VCS_GetDriverInfo(..) - BOOL VCS_GetInterfaceName(..) - BOOL VCS_GetInterfaceNameSelection(..) - BOOL VCS_GetPortName(..) - BOOL VCS_GetPortNameSelection(..) - BOOL VCS_GetProtocolStackModeSelection(..) - BOOL VCS_GetProtocolStackName(..) - BOOL 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: - BOOL VCS_CloseAllDevices(..) - BOOL VCS_DigitalInputConfiguration(..) - BOOL VCS_DigitalOutputConfiguration(..) - BOOL VCS_GetAllDigitalInputs(..) - BOOL VCS_GetAllDigitalOutputs(..) - BOOL VCS_GetAnalogInput(..) - BOOL VCS_SetAllDigitalOutputs(..) - BOOL VCS_SendNMTService(..) - HANDLE VCS_OpenDeviceDlg(..) • All this functions have been changed: - BOOL VCS_GetBaudrateSelection(..) - BOOL VCS_FindHome(..) - BOOL VCS_GetHomingParameter(..) - BOOL VCS_SetHomingParameter(..) - BOOL VCS_MoveToPosition(..) - BOOL VCS_GetOperationMode(..) - BOOL VCS_SetOperationMode(..) - BOOL VCS_GetObject(..) - BOOL VCS_SetObject(..) • All this functions have been deleted: - BOOL VCS_GetProtocolStackMode(..) - BOOL 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 Vector 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
10.01.2007	4.3.0.0	Edition January 2007	• Support for National Instruments Interfaces