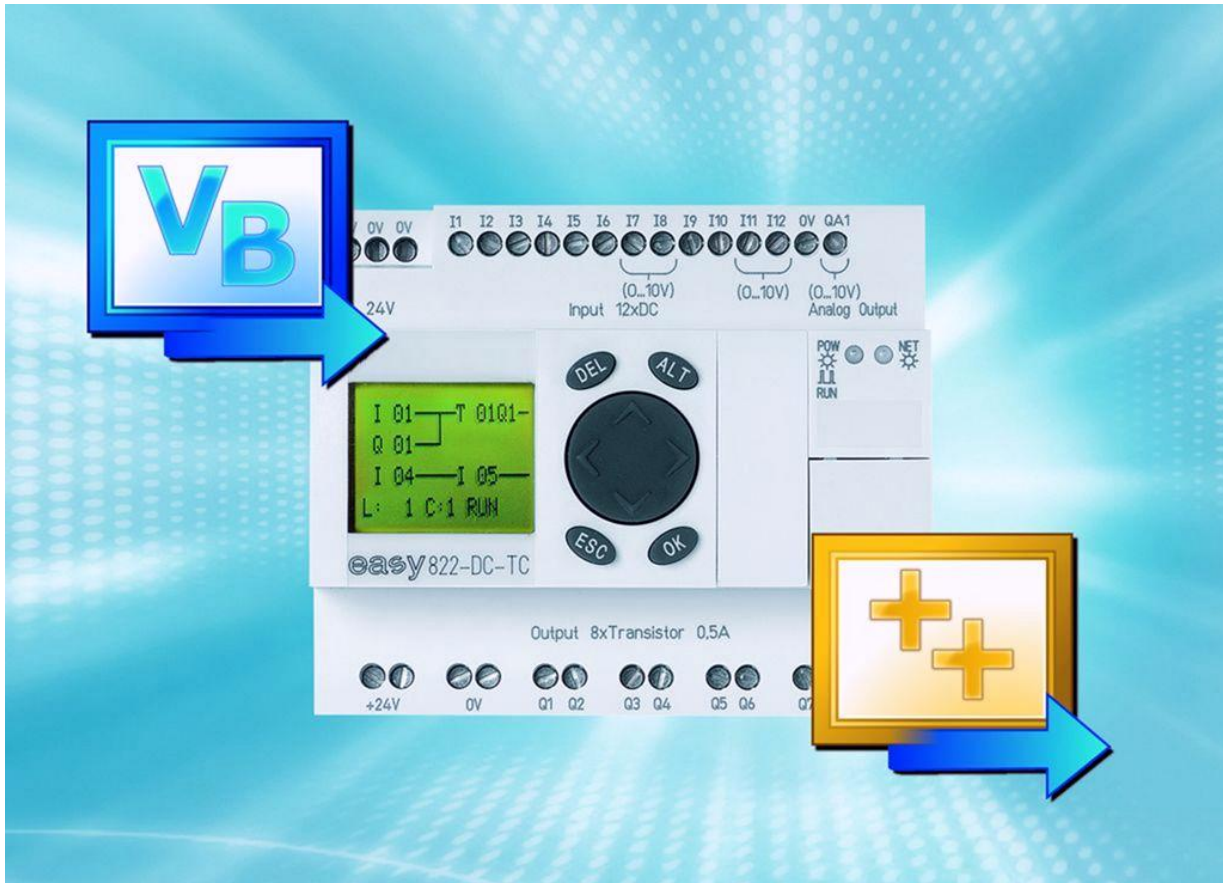


Anwendungshinweise

Funktionsbibliothek EASY_COM V2.5



11/11 AM_EASY_COM_D

(C) Eaton Industries GmbH, 53105 Bonn
Electrical Sector

Autor: E. Kastner
easy@eaton.com

Alle Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelfalter.

Alle Rechte, auch die der Übersetzung, vorbehalten.

Kein Bestandteil dieses Dokuments darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Zustimmung der Firma Eaton Industries GmbH, Bonn, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Änderungen vorbehalten.

Inhaltsverzeichnis

1. Einleitung.....	3
2. Lizenzbedingungen	4
3. Verwendung der Funktionsbibliothek	5
3.1 Systemvoraussetzungen.....	5
3.2 Verwendung unter Microsoft Visual C++	5
3.3 Verwendung unter .NET	6
3.3.1 Verwendung unter Microsoft Visual Basic 2005/2008/2010 (VB.NET).....	6
3.4 Verwendung unter Microsoft Visual Basic 6.0.....	6
3.5 Verwendung mit Java SE Development Kit 6	7
3.6 Lieferumfang.....	8
4. Beschreibung der Schnittstellenfunktionen	9
4.1 Funktionen für den Einzelverbindungsmodus.....	10
Open_ComPort.....	10
Open_EthernetPort	11
Close_ComPort	12
Close_EthernetPort.....	12
GetCurrent_Baudrate	12
Set_UserWaitingTime	13
Get_UserWaitingTime	13
GetLastSystemError	14
Start_Program.....	14
Stop_Program.....	15
Read_Clock	16
Write_Clock.....	17
Read_Object_Value.....	18
Write_Object_Value	20
Read_Channel_YearTimeSwitch.....	21
Write_Channel_YearTimeSwitch	22
Read_Channel_7DayTimeSwitch	23
Write_Channel_7DayTimeSwitch.....	24
Unlock_Device	25
Lock_Device	26
4.2 Funktionen für den Mehrfachverbindungsmodus.....	27
MC_CloseAll	27
MC_Open_ComPort	28
MC_Open_EthernetPort.....	29
MC_Close_ComPort.....	30
MC_Close_EthernetPort	30
MC_GetCurrent_Baudrate.....	30
MC_Set_UserWaitingTime	31
MC_Get_UserWaitingTime.....	31
MC_Start_Program	32
MC_Stop_Program	33
MC_Read_Clock.....	34
MC_Write_Clock	35
MC_Read_Object_Value	36
MC_Write_Object_Value	37
MC_Read_Channel_YearTimeSwitch	39
MC_Write_Channel_YearTimeSwitch.....	40
MC_Read_Channel_7DayTimeSwitch	41
MC_Write_Channel_7DayTimeSwitch.....	42
MC_Unlock_Device	43
MC_Lock_Device.....	44
5. Anhang.....	45
5.1 Fehlercodes der Geräte	45
5.2 Weiterführende Informationen.....	47
5.3 Glossar	48

1. Einleitung

Die Funktionsbibliothek 'EASY_COM.dll' enthält Funktionen zur Kommunikation mit easy500, easy700, easy800 und MFD-Titan.

Die Funktionsbibliothek ist für Programmierer gedacht, die ohne großen Aufwand eigene Windows-basierende Visualisierungslösungen für easy Geräte erstellen wollen. Sie deckt daher bewusst nicht den ganzen Funktionsumfang ab, den easySoft oder easy-OPC-Server anbieten.

Funktionsumfang:

- Verbindungsaufnahme über einen COM-Port (serielle Schnittstelle oder USB-Umsetzer) mit automatischem Test der in Frage kommenden Baudraten
- Verbindungsaufnahme über Ethernet bzw. TCP/IP
- Gleichzeitiger Betrieb von mehreren offenen Verbindungen
- Kommunizieren mit allen easyNet - Teilnehmern über eine offene Verbindung (Routing)
- Aufschließen und Zuschließen eines Gerätes mit System-Passwortschutz
- Starten und Anhalten der Programmausführung
- Auslesen und Stellen der Geräteuhr
- Auslesen des Prozessabbilds
- Schreiben in den Merkerbereich
- Auslesen und Schreiben von Jahresschaltuhr- und Wochenschaltuhr-Kanälen

Um die Handhabung der Funktionsbibliothek zu vereinfachen, werden die Kommunikationsfunktionen in zwei verschiedenen Funktionsgruppen angeboten:

Im **Einzelverbindungsmodus** wird maximal nur eine offene Verbindung innerhalb einer Anwendung bzw. eines Prozesses unterstützt. Das Öffnen einer weiteren Verbindung führt automatisch zum Schließen der vorherigen Verbindung.

Im **Mehrfachverbindungsmodus** können mehrere Verbindungen gleichzeitig geöffnet und verwendet werden, sofern diese nicht dieselbe Schnittstelle und nicht dasselbe Gerät ansprechen. Beim Öffnen einer neuen Verbindung wird ein Identifikationsmerkmal („Handle“) zurückgegeben, das beim Aufruf der anderen Kommunikationsfunktionen mit anzugeben ist. Über dieses „Handle“ erfolgt die Auswahl der Verbindung.

Die Kommunikationsfunktionen unterscheiden sich ansonsten nicht von denen des Einzelverbindungsmodus.

Hinweis: Für den gleichzeitigen Zugriff auf alle Teilnehmer eines easyNet - Strangs wird nur eine einzige Verbindung benötigt.

2. Lizenzbedingungen

Die Firma Eaton Industries GmbH räumt dem Nutzer ein unbeschränktes, nicht ausschließliches und gebührenfreies Nutzungsrecht ein, die Funktionsbibliothek 'EASY_COM.dll' zum Entwickeln einer eigenen Software zu verwenden.

Der Nutzer erklärt sich damit einverstanden,

1. Nur den ausführbaren Anteil der Funktionsbibliothek (die im Paket erhaltene Datei 'EASY_COM.dll') und nur in Verbindung mit und als Bestandteil einer eigenen entwickelten Software weiterzugeben oder zu vertreiben,
2. den Namen, das Logo oder die Marke von Eaton Industries nicht zu nutzen, um die von ihm entwickelte Software zu vermarkten,
3. keine Urheberrechts-, Marken- oder Patenthinweise zu entfernen oder unkenntlich zu machen, so wie sie in der Funktionsbibliothek 'EASY_COM.dll' erscheinen, sowie
4. die Firma Eaton Industries von allen Ansprüchen freizustellen, die aus der Nutzung oder dem Vertrieb der damit entwickelten Software entstehen, die durch Dritte Eaton Industries gegenüber geltend gemacht werden bzw. Eaton Industries hierfür zu entschädigen.

Eine Haftung der Eaton Industries GmbH – gleich aus welchem Rechtsgrund – ist ausgeschlossen. Dies gilt nicht im Falle einer zwingenden Haftung nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, der groben Fahrlässigkeit, bei arglistigem Verschweigen eines Mangels, bei Nichteinhaltung einer Beschaffheitsgarantie, wegen der Verletzung des Körpers oder der Gesundheit oder wegen der Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegt oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit gehaftet wird.

3. Verwendung der Funktionsbibliothek

3.1 Systemvoraussetzungen

Die EASY_COM.dll ist als Teil einer 32-Bit Anwendung unter folgenden Windows Versionen lauffähig:

- Windows 2000 ab SP4
- Windows XP ab SP2
- Windows Vista (32-Bit)
- Windows 7 (32-Bit und 64-Bit)

Die EASY_COM.dll unterstützt alle bislang erschienenen Gerätefamilien

- easy500
- easy700
- easy800
- MFD-Titan

Für den Zugriff auf easyNet Teilnehmer muss das easyNet bereits erfolgreich in Betrieb genommen worden sein.

Ebenso müssen für Ethernet- bzw. TCP/IP - Verbindungen über ein EASY209-SE oder eine Onboard-Ethernet-Schnittstelle die Verbindungsparameter bereits korrekt in den Geräten eingetragen worden sein. Eine Konfiguration der Geräte wird von der EASY_COM.dll nicht vorgenommen.

3.2 Verwendung unter Microsoft Visual C++

Die Funktionsbibliothek kann sowohl für Microsoft Visual C++ V6.0 SP5 als auch für Microsoft Visual Studio 2005/2008/2010 verwendet werden.

Für die Generierung eigener Projekte wird die im Paket mitgelieferte Quellcode-Headerdatei ,easyComApi.h' sowie die Import-Bibliothekskdatei ,EASY_COM.lib' benötigt. Diese Dateien sollten lokal im Projekt hinterlegt werden.

Zur Ausführung der generierten Anwendung wird lediglich die EASY_COM.DLL benötigt, sowie das ,Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)'. Falls die Anwendung mit einer abweichenden Visual C++ Version erstellt wurde, sind evtl. noch weitere Redistributable Packages erforderlich.

Die EASY_COM.DLL sollte im Verzeichnis der Anwendung hinterlegt werden.

Die Bekanntgabe der Schnittstellenfunktionen erfolgt durch die #include-Anweisung auf die Headerdatei (ggf. muss der Include-Suchpfad in den Präprozessor-Einstellungen ergänzt werden). In den Linker-Einstellungen ,Objekt-/Bibliothekmodule' bzw. ,Zusätzliche Abhängigkeiten' ist bei Visual Studio 2005/2008/2010 außerdem die Angabe ,EASY_COM.lib' zu ergänzen.

Die EASY_COM.dll wurde erstellt mit den Compiler-Einstellungen

Plattform Win32,
Multithreaded DLL,
kein Unicode,
keine C++ Ausnahmen

3.3 Verwendung unter .NET

Die Funktionsbibliothek kann unter Microsoft Visual Studio 2005, 2008, 2010 verwendet werden.

Für die Generierung eigener .NET - Projekte wird die im Paket mitgelieferte Quellcode-Datei ‚easyComApi.cs‘ (Einzelverbindungsmodus) oder die Quellcode-Datei ‚easyComApi_MC.cs‘ (Mehrfachverbindungsmodus) benötigt. Diese Dateien müssen dem Solution Projekt als ‚vorhandenes Element‘ hinzugefügt werden. Stellen Sie als Zielplattform ‚x86‘ ein, damit die Anwendung sowohl auf 32-Bit Systemen wie auf 64-Bit-Systemen lauffähig ist.

Zur Ausführung der generierten Anwendung wird lediglich die EASY_COM.DLL benötigt, sowie das ‚Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)‘. Falls die Anwendung mit einer abweichenden Visual Studio Version erstellt wurde, sind evtl. noch weitere Redistributable Packages erforderlich.

Die EASY_COM.DLL sollte im Verzeichnis der Anwendung hinterlegt werden.

Die Quellcodedateien definieren die in C# geschriebenen Klassen ‚easy_COM_API‘ bzw. ‚easy_COM_API_MC‘ mit statischen Methoden, deren Namen und Parameter den Funktionen aus der EASY_COM.dll entsprechen.

Da es sich bei der EASY_COM.dll um eine ‚unmanaged code‘ - DLL müssen die Einsprungspunkte der DLL durch ‚DllImport‘ - Anweisungen bekannt gegeben werden. Die bereitgestellten Klassen erlauben nicht den direkten Aufruf der Einsprungspunkte, sondern kapseln diese in zusätzliche Methoden, um mögliche Ausnahmefehler wie z.B. `System.DllNotFoundException` oder

`System.EntryPointNotFoundException` abfangen zu können.

Hinsichtlich der Bedienung der neuen Methoden gelten die Konventionen, die im Kapitel „Schnittstellenbeschreibung“ beschrieben sind.

3.3.1 Verwendung unter Microsoft Visual Basic 2005/2008/2010 (VB.NET)

Für die Generierung eigener Projekte wird die im Paket mitgelieferte Quellcode-Datei ‚easyComApi.vb‘ benötigt. Diese Datei muss dem Solution Projekt als ‚vorhandenes Element‘ hinzugefügt werden. Stellen Sie als Zielplattform ‚x86‘ ein, damit die Anwendung sowohl auf 32-Bit Systemen wie auf 64-Bit-Systemen lauffähig ist.

Zur Ausführung der generierten Anwendung wird lediglich die EASY_COM.DLL benötigt, sowie das ‚Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)‘. Falls die Anwendung mit einer abweichenden Visual Basic Version erstellt wurde, sind evtl. noch weitere Redistributable Packages erforderlich.

Die EASY_COM.DLL sollte im Verzeichnis der Anwendung hinterlegt werden.

Die Quellcodedatei definiert eine Klasse namens ‚easyCOM‘ mit statischen Methoden, deren Namen und Parameter den Funktionen aus der EASY_COM.dll entsprechen.

3.4 Verwendung unter Microsoft Visual Basic 6.0

Für die Generierung eigener Projekte wird die im Paket mitgelieferte Quellcode-Datei ‚easyComApi.bas‘ benötigt. Diese Datei muss dem Visual Basic Projekt als ‚vorhandenes Modul‘ hinzugefügt werden.

Zur Ausführung der generierten Anwendung wird lediglich die EASY_COM.DLL benötigt, sowie das ‚Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)‘.

Die EASY_COM.DLL sollte im Verzeichnis der Anwendung hinterlegt werden.

3.5 Verwendung mit Java SE Development Kit 6

Für die Generierung eigener Projekte wird die im Paket mitgelieferte Quellcode-Datei ‚easyCom_API.java‘ oder ‚easyCom_API_MC.java‘ benötigt. Beide Dateien benötigen die Bibliothek ‚Java Native Access‘ (JNA) zur vereinfachten Deklaration der easyCOM-Schnittstelle (siehe <http://jna.java.net>). Der Pfad auf die JNA-Bibliotheksdateien muss sowohl in der Entwicklungsumgebung als auch in der Laufzeitumgebung zusätzlich angegeben werden:

Beispiel:

Übersetzung: `javac.exe -classpath jna.jar;platform.jar;. easyCOM_API.java`

Ausführung: `java.exe -classpath jna.jar;platform.jar;. easyCOM_API`

Zur Ausführung der generierten Anwendung wird lediglich die EASY_COM.DLL benötigt, sowie das ‚Microsoft Visual C++ 2008 SP1 Redistributable Package (x86)‘.

Die EASY_COM.DLL sollte im Verzeichnis der Anwendung hinterlegt werden.

3.6 Lieferumfang

Die Funktionsbibliothek wird als Zip-Archiv ausgeliefert. Diese Datei enthält folgende Dateien:

EASY_COM.dll	die ausführbare Funktionsbibliothek als Dynamic Link Library
EASY_COM.lib	Import-Bibliotheksdatei für MS Visual C++ V6.0 bzw. MS Visual C++ 2005/2008/2010.
Doku\easyComApi.h	Headerdatei in C/C++ mit der Deklaration der Bibliotheksschnittstelle
Doku\easyComApi.vb	Visual Basic 2005/2008/2010-Quellcodedatei mit der Deklaration der Bibliotheksschnittstelle zur Einbindung unter .NET
Doku\easyComApi.bas	Visual Basic 6-Quellcodedatei mit der Deklaration der Bibliotheksschnittstelle
Doku\easyComApi.cs Doku\easyComApi_MC.cs	C# Quellcodedateien mit den Klassendefinitionen zur Einbindung der Bibliotheksschnittstelle unter .NET
Doku\easyCOM_API.java Doku\easyCOM_API_MC.java	Java Quellcodedateien mit den Klassendefinitionen zur Einbindung der Bibliotheksschnittstelle mittels Java Native Access
Doku\AM_EASY_COM_D.pdf	Anwendungshinweise zur Bibliothek, dt. Fassung (dieses Dokument)
Doku\AM_EASY_COM_G.pdf	Anwendungshinweise zur Bibliothek, engl. Fassung
Demo.cpp	Das Verzeichnis enthält ein MS Visual Studio 2008 Projekt für eine dialogbasierte MFC-Anwendung in C++, die die Verwendung der Funktionsbibliothek demonstriert.
DemoVB.NET	Das Verzeichnis enthält ein MS Visual Basic 2008 Projekt für eine dialogbasierte .NET Anwendung, die die Verwendung der Funktionsbibliothek demonstriert. Das Projekt kann auch für MS Visual Basic 2010 (Express) verwendet werden.

4. Beschreibung der Schnittstellenfunktionen

Die Funktionsbeschreibung startet mit einer Deklaration in der Programmiersprache C.

Diese beginnt mit dem Datentyp des Rückgabewertes, gefolgt vom Funktionsnamen und der in Klammern gesetzten Parameterliste (Datentyp des Parameters gefolgt vom Parameternamen).

Steht zwischen dem Datentyp und dem Parameternamen noch ein Stern * bedeutet dies, dass es sich bei dem Parameter um einen Zeiger entweder auf ein einzelnes Datenelement oder auf ein Feld von mehreren Elementen dieses Datentyps handelt. In diesem Fall muss an der aufrufenden Stelle eine Datenstruktur in ausreichender Größe reserviert werden (Parameterbeschreibung beachten), deren Speicheradresse der Funktion zu übergeben ist.

Die folgende Aufstellung der verwendeten Datentypen soll eine Hilfestellung bei der Portierung der Schnittstelle zu anderen Programmiersprachen geben:

Datentyp in C	Bedeutung
unsigned char	8-Bit, vorzeichenlos
unsigned char*	Zeiger auf 8-Bit-Element oder Speicherblock
unsigned short	16-Bit, vorzeichenlos
long	32-Bit, vorzeichenbehaftet
long*	Zeiger auf 32-Bit-Element
unsigned long	32-Bit, vorzeichenlos
bool	Boolean (wahr, falsch), 8-Bit Kodierung
const char*	ASCII-Zeichenkette mit Nullterminierung
tEasyComHandle	untypisierter Zeiger
tEasyComHandle*	Zeiger auf Zeiger

Datentyp in C / C++	Datentyp in C#	Datentyp in VB.NET	Datentyp in Visual Basic 6	Datentyp in Java Native Access
unsigned char	byte	ByVal Byte	ByVal Byte	byte ²
unsigned char*	ref byte ¹	ByRef Byte	ByRef Byte	ByteByReference ³
unsigned short	ushort	ByVal UShort	ByVal Integer	short ²
long	int	ByVal Integer	ByVal Long	int
long*	out int	ByRef Integer	ByRef Long	IntByReference
unsigned long	uint	ByVal Integer ²	ByVal Long ²	int ²
bool	bool	ByVal Boolean	ByVal Boolean	byte
const char*	String ⁴	ByVal String	ByVal String	String
tEasyComHandle	IntPtr	ByVal Integer	ByVal Integer	Pointer
tEasyComHandle*	out IntPtr	ByRef Integer	ByRef Integer	PointerByReference

¹ bei Speicherblock: [MarshalAs(UnmanagedType.LPArray)] byte[]

² vorzeichenbehafteter Datentyp, da keine exakte Entsprechung existiert

³ bei Speicherblock: byte[]

⁴ [MarshalAs(UnmanagedType.LPStr)] String

4.1 Funktionen für den Einzelverbindungsmodus

Open_ComPort

long Open_ComPort (unsigned char com_port_nr, long baudrate)

Stellt eine Verbindung zum Gerät über den angegebenen COM-Port her.

Die Funktion öffnet die angeforderte serielle Schnittstelle und konfiguriert diese für das easy-COM Protokoll. Anstelle einer seriellen Schnittstelle kann sich auch ein virtueller COM-Port eines Umsetzer-Treibers verbergen, insbesondere der eines EASY(800)-USB-CAB Programmierkabels.

Kommt keine Verbindung zum Gerät zustande, wird ein Verbindungsaufbau mit allen in Frage kommenden Baudraten versucht. Kann die Verbindung zum Gerät hergestellt werden, wird anschließend das Gerät auf die gewünschte Baudrate umkonfiguriert. Welche Baudrate letztendlich eingestellt wird, hängt von den Fähigkeiten des Geräts sowie der verwendeten Programmierkabel ab. Bei erfolgreicher Verbindungsaufnahme kann anschließend mit der Funktion [GetCurrent Baudrate](#) die tatsächlich eingestellte Baudrate ermittelt werden.

Die Gerätefamilien easy500 und easy700 unterstützen generell nur 4800 Baud.

Wurde bereits zuvor eine Verbindung geöffnet, wird diese zuerst geschlossen, da innerhalb einer Anwendung nicht gleichzeitig mehrere offene Verbindungen unterstützt werden.

Wenn die Schnittstelle des Geräts mit einem System-Passwort geschützt ist, liefert die Funktion den Fehlercode 4 zurück. Die Verbindung ist zwar zum Gerät hergestellt worden, aber es muss anschließend die Funktion [Unlock Device](#) aufgerufen werden, um weiter mit dem Gerät kommunizieren zu können.

Parameter:

com_port_nr Nummer des COM-Ports, der geöffnet werden soll, z.B. 3 für 'COM3' { 1...255 }
baudrate gewünschte Baudrate { 4800, 9600, 19200, 38400, 57600 }

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 4 Die Schnittstelle des Geräts ist mit einem Passwort geschützt. [Unlock Device](#) aufrufen!
- 5 Gerätetyp ist unbekannt.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastSystemError](#) abfragen.
- 8 Der COM-Port ist nicht vorhanden oder derzeit nicht im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll

Open_EthernetPort

long Open_EthernetPort (const char * szIpAddress, long IpPort, long baudrate, bool no_baudrate_scan)

Stellt über Ethernet bzw. Internet eine Verbindung zum Gerät her.

Die Funktion öffnet die angeforderte TCP/IP-Verbindung und konfiguriert diese für das easy-COM Protokoll. Die Konfiguration der TCP/IP-Verbindung selbst muss also zuvor bereits vorgenommen worden sein (siehe [Systemvoraussetzungen](#)).

Kommt keine Verbindung zum Gerät zustande, wird ein Verbindungsaufbau mit allen in Frage kommenden Baudraten versucht. Kann die Verbindung zum Gerät hergestellt werden, wird anschließend das Gerät auf die gewünschte Baudrate umkonfiguriert. Welche Baudrate letztendlich eingestellt wird, hängt von den Fähigkeiten des Geräts sowie der verwendeten Programmierkabel ab. Bei erfolgreicher Verbindungsaufnahme kann anschließend mit der Funktion [GetCurrent Baudrate](#) die tatsächlich eingestellte Baudrate ermittelt werden.

Der Baudraten-Test kann aufgrund der längeren TCP/IP-Wartezeiten bis zu einer Minute dauern. Daher sollte für easy500 und easy700 der Parameter 'no_baudrate_scan' auf 'true' gesetzt werden, um den Baudraten-Test abzuschalten (die Geräte unterstützen generell nur 4800 baud).

Dasselbe gilt auch, wenn die Verbindung nicht über das Ethernet-Gateway EASY209-SE hergestellt werden soll. In diesem Fall müssen Gerät und Umsetzer bereits auf die angegebene Baudrate konfiguriert worden sein oder das Gerät besitzt eine Ethernet-Schnittstelle onboard, so dass die Baudrate für die Verbindungsaufnahme bedeutungslos ist.

Die in Frage kommenden TCP/IP-Portnummern sind geräteabhängig: Für easy und MFD-Titan, die über ein EASY209-SE Gateway angebunden sind, ist der Wertebereich 10001 bis 10999 möglich.

Wurde bereits zuvor eine Verbindung geöffnet, wird diese zuerst geschlossen, da innerhalb einer Anwendung nicht gleichzeitig mehrere offene Verbindungen unterstützt werden.

Parameter:

<i>szIpAddress</i>	IPv4-Adresse als 0-terminierte Zeichenkette, in der Regel 4-stellig z.B. "10.1.41.31"
<i>IpPort</i>	IP-Portnummer { 1200, 10001...10999 }
<i>baudrate</i>	eingestellte Baudrate im Gerät { 4800, 9600, 19200, 38400, 57600 }
<i>no_baudrate_scan</i>	Schaltet den Baudraten-Test ab. 'true' => nur die angegebene Baudrate wird getestet

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 4 Die Schnittstelle des Geräts ist mit einem Passwort geschützt. [Unlock Device](#) aufrufen!
- 5 Gerätetyp ist unbekannt.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastSystemError](#) abfragen.
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 14 TCP/IP Teilnehmer antwortet nicht oder der TCP/IP Port wird bereits von einer anderen offenen Verbindung belegt
- 15 Baudraten-Test ist nicht möglich

Siehe auch:

[GetCurrent Baudrate](#)
[GetLastSystemError](#)

Close_ComPort

long Close_ComPort ()

Beendet die Verbindung zum Gerät.

Die Funktion beendet die Verbindung zum Gerät und gibt die serielle Schnittstelle wieder frei.

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.

Close_EthernetPort

long Close_EthernetPort ()

Beendet die Verbindung zum Gerät.

Die Funktion beendet die Verbindung zum Gerät und gibt die Ethernet-Verbindung wieder frei.

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.

GetCurrent_Baudrate

long GetCurrent_Baudrate (long * baudrate)

Liefert die eingestellte Baudrate der aktuellen Verbindung zurück.

Die Funktion liefert nach Aufruf von [Open_ComPort](#) oder [Open_EthemetPort](#) die tatsächlich eingestellte Baudrate zurück. Welche Baudrate beim Verbindungsaufbau eingestellt wird, hängt von den Fähigkeiten des Geräts, der verwendeten Programmierkabel, der Verbindungsart und ggf. von den Fähigkeiten von zwischengeschalteten Umsetzern ab.

Nach Aufruf von Open_EthernetPort ohne Baudratentest ist der Rückgabewert der Funktion undefiniert.

Parameter:

baudrate

Zeiger auf eine Variable zur Ablage der aktuellen Baudrate { 4800, 9600, 19200, 38400, 57600 }

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.

1 Ein Parameter enthielt einen ungültigen Wert.

7 Es ist keine Verbindung offen.

Set_UserWaitingTime

long Set_UserWaitingTime (long timeout_delay)

Stellt eine zusätzliche Timeout-Verzögerung ein.

Die Funktion stellt eine zusätzliche Verzögerungszeit ein, auf die bei jeder Geräteanfrage gewartet werden soll, bevor ein Verbindungsabbruch gemeldet wird.

Da Funk oder Modem-Verbindungen einen längeren Übertragungsweg haben, müssen auf der PC-Seite längere Wartezeiten eingestellt werden. Die Funktion sollte bereits vor dem Funktionsaufruf zum Öffnen einer Verbindung aufgerufen werden.

Ein Wert größer 0 führt aber zwangsläufig auch zu einer Verzögerung, bis ein tatsächlich eingetretener Verbindungsabbruch erkannt wird. Daher sollte der Wert nicht größer als unbedingt nötig gewählt werden.

Je nach Übertragungsweg und verwendetem Protokollumsetzer treten unterschiedliche Verzögerungen ein, so dass kein allgemeiner Richtwert für die Verzögerungszeit genannt werden kann. Ein erster Testlauf kann z.B. mit 800ms gestartet werden, treten dabei noch Verbindungsabbrüche auf, sollte der Wert schrittweise erhöht werden.

Als Startwert ist 0 für direkte serielle Verbindungen zwischen PC und Gerät voreingestellt.

Parameter:

timeout_delay

Verzögerungszeit in Millisekunden { 0...90000 }

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.

1 Ein Parameter enthielt einen ungültigen Wert.

Get_UserWaitingTime

long Get_UserWaitingTime (long* timeout_delay)

Liefert die zuletzt eingestellte zusätzliche Timeout-Verzögerung zurück.

Parameter:

timeout_delay

Zeiger auf eine Variable zur Ablage der Verzögerungszeit { 0...90000 }

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.

1 Ein Parameter enthielt einen ungültigen Wert.

GetLastSystemError

unsigned long GetLastSystemError ()

Liefert den zuletzt von Windows übermittelten Fehlercode zurück.

Liefert eine der Kommunikationsfunktionen den Fehlercode 6 zurück, so kann anschließend mit dieser Funktion der präzise Fehlercode von Windows ermittelt werden. Dieser Fehlercode entspricht dem Rückgabewert der Windows Systemfunktion GetLastError. Eine Beschreibung der ‚System Error Codes‘ ist z.B. in der MSDN enthalten:

<http://msdn2.microsoft.com/en-us/library/ms681381.aspx>

Rückgabewerte:

Windows System Errorcode

Start_Program

long Start_Program (unsigned char net_id, unsigned char * errorcode)

Schaltet das Gerät in 'RUN'.

Die Funktion schaltet das Gerät in den RUN Modus. Sofern die Anzeige am Gerät sich noch nicht in der Grundstellung befindet, wird diese zuvor automatisch zurückgesetzt (noch nicht übernommene Benutzereingaben gehen verloren).

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1 bis 8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'errorcode' (siehe [Fehlercodes der Geräte](#)).

Achtung:

Für den Wechsel nach RUN kann das Gerät mehrere Sekunden benötigen!

Ist das Gerät ein easyNet Master (Teilnehmer NT1), so können aufgrund der 'Remote RUN' - Option weitere Geräte nach RUN umschalten.

Parameter:

net_id Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
errorcode Zeiger auf eine Bytevariable zur Ablage des vom Gerät gesendeten Fehlercodes.

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastSystemError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll

Stop_Program

long Stop_Program (unsigned char net_id, unsigned char * errorcode)

Schaltet das Gerät in 'STOP'.

Die Funktion schaltet das Gerät in den STOP Modus. Sofern die Anzeige am Gerät sich noch nicht in der Grundstellung befindet, wird diese zuvor automatisch zurückgesetzt (noch nicht übernommene Benutzer-eingaben gehen verloren).

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'errorcode' (siehe [Fehlercodes der Geräte](#)).

Achtung:

Ist das Gerät ein easyNet Master (Teilnehmer NT1), so können aufgrund der 'Remote RUN' - Option weitere Geräte nach 'STOP' umschalten.

Parameter:

net_id Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
errorcode Zeiger auf eine Bytevariable zur Ablage des vom Gerät gesendeten Fehlercodes.

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll

Read_Clock

```
long Read_Clock (  
    unsigned char net_id, unsigned char* year, unsigned char* month, unsigned char* day,  
    unsigned char* hour, unsigned char* min)
```

Auslesen der Geräteuhr.

Die Funktion liest das Datum und die Uhrzeit aus dem Gerät aus.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ‚year‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>year</i>	Zeiger auf eine Bytevariable zur Ablage der Jahreszahl {0...99}, 0 entspricht Jahr 2000
<i>month</i>	Zeiger auf eine Bytevariable zur Ablage des Monats {1...12}
<i>day</i>	Zeiger auf eine Bytevariable zur Ablage des Kalendertags {1...31}
<i>hour</i>	Zeiger auf eine Bytevariable zur Ablage der Stunde {0...23}, 24h-Format
<i>min</i>	Zeiger auf eine Bytevariable zur Ablage der Minute {0...59}

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll

Write_Clock

```
long Write_Clock (  
    unsigned char net_id, unsigned char* year, unsigned char* month, unsigned char* day,  
    unsigned char* hour, unsigned char* min)
```

Stellen der Geräteuhr.

Die Funktion schreibt das Datum und die Uhrzeit in das Gerät.

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'year' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>year</i>	Zeiger auf eine Bytevariable zur Ablage der Jahreszahl {0...99}, 0 entspricht Jahr 2000
<i>month</i>	Zeiger auf eine Bytevariable zur Ablage des Monats {1...12}
<i>day</i>	Zeiger auf eine Bytevariable zur Ablage des Kalendertags {1...31}
<i>hour</i>	Zeiger auf eine Bytevariable zur Ablage der Stunde {0...23}, 24h-Format
<i>min</i>	Zeiger auf eine Bytevariable zur Ablage der Minute {0...59}

Rückgabewerte:

0	Funktionsaufruf war erfolgreich.
1	Ein Parameter enthielt einen ungültigen Wert.
2	Gerät übermittelte eine Fehlermeldung.
3	Gerät antwortet nicht.
6	Fehlermeldung von Windows, Fehlercode mit GetLastError abfragen.
7	Es ist keine Verbindung offen.
8	Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
9	Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
10	Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
11	Interner Fehler der EASY_COM.dll

Read_Object_Value

```
long Read_Object_Value(
    unsigned char net_id, unsigned char object, unsigned short index, unsigned char * data)
```

Auslesen des Prozessabbilds.

Die Funktion liest aus dem Gerät einen Bereich des Prozessabbilds.

Der Übergabeparameter 'object' legt den Bereich des Prozessabbilds und damit auch die ausgelesene Datenmenge fest:

0	Digital-Eingänge Basisgerät	I1...I16	2 Byte
1	Digital-Ausgänge Basisgerät	Q1...Q8	2 Byte
2	Digital-Eingänge Erweiterungsmodul	R1...R16	2 Byte
3	Digital-Ausgänge Erweiterungsmodul	S1...S8	2 Byte
4	Bit-Merker bei easy800 und MFD-Titan	M1...M96	12 Byte
4	Bit-Merker bei easy500 und easy700	M1...M16	2 Byte
4	Bit-Merker bei easy800-DC-SWD	M1...M128	16 Byte
5	P-Tasten	P1...P4	1 Byte
6	easyNet-Eingänge	xI1...xI16 und xR1...xR16	4 Byte
7	easyNet-Ausgänge	xQ1...xQ8 und xS1...xS8	4 Byte
8	Analog-Eingänge Basisgerät	IA1...IA4	8 Byte
9	Analog-Ausgang Basisgerät	QA1	2 Byte
10	8 Doppelwort-Merker	MDx...MDx+7	32 Byte
11	easyNet Empfangsdaten	xRN1...xRN32	32 Byte
12	easyNet Sendedaten	xSN1...xSN32	32 Byte
13	Diagnosebits	ID1...ID16	2 Byte
14	zusätzliche Diagnosebits bei MFD-Titan	ID17...ID32	2 Byte
15	programmierbare LEDs bei MFD-Titan	LE1...LE3	1 Byte
16	zusätzliche Bit-Merker bei easy500/700	N1...N16	2 Byte
17	32 Merkerbyte	MBx...MBx+31	32 Byte
18	Digital-Eingänge Basisgerät und zugeordnete SWD-Eingänge	I1...I128	16 Byte
19	Digital-Ausgänge Basisgerät und zugeordnete SWD-Ausgänge	Q1...Q128	16 Byte

Weitere Informationen entnehmen Sie bitte dem Bedienungshandbuch zum jeweiligen Gerät.

Die Zustandswerte werden im Little-Endian-Format (Intel-Format) zurückgegeben. Der Zustandswert des Operanden mit der kleinsten Operandennummer steht im geringwertigsten Bit (LSB).

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im ersten Byte des Übergabeparameters 'data' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>object</i>	Nummer des Objekts also des Prozessabbildbereichs
<i>index</i>	Teilnehmernummer bei Objekt 6,7,11,12 {1...8} Operandennummer bei Objekt 10 {1...96} bzw. {1...128} Bytenummer bei Objekt 17 {1...384} bzw. {1...512} Sonst ohne Funktion.
<i>data</i>	Zeiger auf einen Speicherbereich zur Ablage der Zustandswerte

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll

Achtung:

Die Größe des Datenfeldes, das mit 'data' übergeben wird, muss ausreichend groß dimensioniert sein, da sonst über die Speichergrenze hinausgeschrieben wird.

Bei Objekt 10 werden generell 8 Doppelworte gelesen, außer es wird versucht, über das Ende des Merkerbereichs hinaus zu lesen (dann werden entsprechend weniger Byte eingetragen).

Das Objekt 17 erlaubt den byteweisen Zugriff auf den gesamten Merkerbereich, auch wenn das Gerät nur für den unteren Merkerbereich einen Merkerbyte-Zugriff kennt. Es werden generell 32 Byte gelesen, außer es wird versucht, über das Ende des Merkerbereichs hinaus zu lesen (dann werden entsprechend weniger Byte eingetragen).

Die Funktion [MC_Read_Object_Value](#) ermöglicht für Objekt 10 und 17 eine wählbare Datenmenge auszulesen.

Write_Object_Value

long Write_Object_Value (
 unsigned char net_id, unsigned char object, unsigned short index, unsigned char length, unsigned char * data)

Schreibt in den Merkerbereich.

Die Funktion schreibt Werte in den Merker-Bereich des Geräts.

Der Übergabeparameter ‚object‘ legt den Typ des Merker-Bereichs fest:

4	einzelner Bit-Merker Mx	1 Byte
10	1 bis 80 Byte ab Doppelwort-Merker MDx	1...80 Byte
16	einzelner Bit-Merker Nx bei easy500 und easy700	1 Byte
17	1 bis 80 Merkerbyte	1...80 Byte

Die Zustandswerte werden im Little-Endian-Format (Intel-Format) interpretiert. Der zu setzende Zustandswert des Bitmerkers muss im geringwertigsten Bit (LSB) eingetragen werden.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1 bis 8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im ersten Byte des Übergabeparameters ‚data‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>object</i>	Nummer des Objekts also des Prozessabbildbereichs
<i>index</i>	Operandennummer bei Objekt 4, 10, 16 {1...16} bzw. {1...96} bzw. {1...128} Bytenummer bei Objekt 17 {1...384} bzw. {1...512}
<i>length</i>	Gibt bei Objekt 10 und 17 die Anzahl der zu schreibenden Byte {1...80} an. Sonst ohne Funktion.
<i>data</i>	Zeiger auf einen Speicherbereich zur Ablage der Zustandswerte

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll

Achtung:

Die Größe des Datenfeldes, das mit dem Parameter ‚data‘ übergeben wird, muss ausreichend groß dimensioniert sein, da sonst über die Speichergrenze hinaus gelesen wird.

Der Zustandswert eines geschriebenen Merkers bleibt nur erhalten, wenn das Gerät kein Programm ausführt, das seinerseits auf diesen Merker schreibend zugreift.

Das Objekt 17 erlaubt den byteweisen Zugriff auf den gesamten Merkerbereich, auch wenn das Gerät nur für den unteren Merkerbereich einen Merkerbyte-Zugriff kennt.

Read_Channel_YearTimeSwitch

```
long Read_Channel_YearTimeSwitch(  
    unsigned char net_id, unsigned char index, unsigned char channel,  
    unsigned char * on_year, unsigned char * on_month, unsigned char * on_day,  
    unsigned char * off_year, unsigned char * off_month, unsigned char * off_day)
```

Parameter eines Jahresschaltuhrkanals auslesen.

Die Funktion liest aus dem angegebenen Jahresschaltuhr-Baustein den Parametersatz des selektierten Kanals aus. Der Parametersatz besteht aus den Bestandteilen ‚Kalendertag‘, ‚Monat‘ und ‚Jahr‘ jeweils getrennt nach Einschaltzeitpunkt und Ausschaltzeitpunkt.

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'on_year' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>index</i>	Gibt die Bausteinnummer an {1...4} bei easy500/700 bzw. {1...32} bei easy800/MFD-Titan.
<i>channel</i>	Gibt den Kanal an {0...3}, 0 entspricht Kanal A, 3 entspricht Kanal D
<i>on_year</i>	Zeiger auf eine Bytevariable zur Ablage der <i>Einschaltzeitpunkt</i> - Jahreszahl {0, 1...99}, 0 entspricht "nicht belegt"
<i>on_month</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Einschaltzeitpunkt</i> - Monats {0, 1...12}, 0 entspricht "nicht belegt"
<i>on_day</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Einschaltzeitpunkt</i> - Kalendertags {0, 1...31}, 0 entspricht "nicht belegt"
<i>off_year</i>	Zeiger auf eine Bytevariable zur Ablage der <i>Ausschaltzeitpunkt</i> - Jahreszahl {0, 1...99}, 0 entspricht "nicht belegt"
<i>off_month</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Ausschaltzeitpunkt</i> - Monats {0, 1...12}, 0 entspricht "nicht belegt"
<i>off_day</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Ausschaltzeitpunkt</i> - Kalendertags {0, 1...31}, 0 entspricht "nicht belegt"

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es wurde kein COM-Port geöffnet.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 12 Der Baustein ist im Programm nicht vorhanden.

Achtung:

Bei easy800 und MFD-Titan liefert das Gerät im STOP generell nur Nullwerte.

Write_Channel_YearTimeSwitch

```
long Write_Channel_YearTimeSwitch(  
    unsigned char net_id, unsigned char index, unsigned char channel,  
    unsigned char * on_year, unsigned char * on_month, unsigned char * on_day,  
    unsigned char * off_year, unsigned char * off_month, unsigned char * off_day)
```

Parameter eines Jahresschaltuhrkanals überschreiben.

Die Funktion überschreibt in dem angegebenen Jahresschaltuhr-Baustein den Parametersatz des selektierten Kanals mit neuen Werten. Der Parametersatz besteht aus den Bestandteilen ‚Kalendertag‘, ‚Monat‘ und ‚Jahr‘ jeweils getrennt nach Einschaltzeitpunkt und Ausschaltzeitpunkt.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ‚on_year‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>index</i>	Gibt die Bausteinnummer an {1...4} bei easy500/700 bzw. {1...32} bei easy800/MFD-Titan.
<i>channel</i>	Gibt den Kanal an {0...3}, 0 entspricht Kanal A und 3 entspricht Kanal D
<i>on_year</i>	Zeiger auf eine Bytevariable zur Ablage der <i>Einschaltzeitpunkt</i> - Jahreszahl {0, 1...99}, 0 entspricht "nicht belegt"
<i>on_month</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Einschaltzeitpunkt</i> - Monats {0, 1...12}, 0 entspricht "nicht belegt"
<i>on_day</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Einschaltzeitpunkt</i> - Kalendertags {0, 1...31}, 0 entspricht "nicht belegt"
<i>off_year</i>	Zeiger auf eine Bytevariable zur Ablage der <i>Ausschaltzeitpunkt</i> - Jahreszahl {0, 1...99}, 0 entspricht "nicht belegt"
<i>off_month</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Ausschaltzeitpunkt</i> - Monats {0, 1...12}, 0 entspricht "nicht belegt"
<i>off_day</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Ausschaltzeitpunkt</i> - Kalendertags {0, 1...31}, 0 entspricht "nicht belegt"

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es wurde kein COM-Port geöffnet.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 12 Der Baustein ist im Programm nicht vorhanden.
- 13 Der Kanal kann nicht überschrieben werden, da kein Parameter hinterlegt ist.

Achtung:

Bei easy800 und MFD-Titan muss im angegebenen Kanal mindestens ein Bestandteil bereits mit einem Wert belegt worden sein, andernfalls gilt der Kanal als nicht verwendet.

Read_Channel_7DayTimeSwitch

```
long Read_Channel_7DayTimeSwitch(  
    unsigned char net_id, unsigned char index, unsigned char channel,  
    unsigned char * DY1, unsigned char * DY2,  
    unsigned char * on_hour, unsigned char * on_minute,  
    unsigned char * off_hour, unsigned char * off_minute)
```

Parameter eines Wochenschaltuhrkanals auslesen.

Die Funktion liest aus dem angegebenen Wochenschaltuhr-Baustein den Parametersatz des selektierten Kanals aus.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ‚DY1‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>index</i>	Gibt die Bausteinnummer an {1...4} bei easy500/700 bzw. {1...32} bei easy800/MFD-Titan.
<i>channel</i>	Gibt den Kanal an {0...3}, 0 entspricht Kanal A und 3 entspricht Kanal D
<i>DY1</i>	Zeiger auf eine Bytevariable zur Ablage des Wochentag-Parameters DY1 {0=SO...6=SA}, 255 entspricht "nicht belegt"
<i>DY2</i>	Zeiger auf eine Bytevariable zur Ablage des Wochentag-Parameters DY2 {0=SO...6=SA}, 255 entspricht "nicht belegt"
<i>on_hour</i>	Zeiger auf eine Bytevariable zur Ablage der Einschaltzeit-Stunde {0...23}, 255 entspricht "nicht belegt"
<i>on_minute</i>	Zeiger auf eine Bytevariable zur Ablage der Einschaltzeit-Minute {0...59}, 255 entspricht "nicht belegt"
<i>off_hour</i>	Zeiger auf eine Bytevariable zur Ablage der Ausschaltzeit-Stunde {0...23}, 255 entspricht "nicht belegt"
<i>off_minute</i>	Zeiger auf eine Bytevariable zur Ablage der Ausschaltzeit-Minute {0...59}, 255 entspricht "nicht belegt"

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 12 Der Baustein ist im Programm nicht vorhanden.

Achtung:

Bei easy800 und MFD-Titan liefert das Gerät im STOP generell nur Nullwerte.

Write_Channel_7DayTimeSwitch

```
long Write_Channel_7DayTimeSwitch(  
    unsigned char net_id, unsigned char index, unsigned char channel,  
    unsigned char * DY1, unsigned char * DY2,  
    unsigned char * on_hour, unsigned char * on_minute,  
    unsigned char * off_hour, unsigned char * off_minute)
```

Parameter eines Wochenschaltuhrkanals überschreiben.

Die Funktion überschreibt in dem angegebenen Wochenschaltuhr-Baustein den Parametersatz des selektierten Kanals mit neuen Werten.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ‚DY1‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>index</i>	Gibt die Bausteinnummer an {1...4} bei easy500/700 bzw. {1...32} bei easy800/MFD-Titan.
<i>channel</i>	Gibt den Kanal an {0...3}, 0 entspricht Kanal A und 3 entspricht Kanal D
<i>DY1</i>	Zeiger auf eine Bytevariable zur Ablage des Wochentag-Parameters DY1 {0=SO...6=SA}, 255 entspricht "nicht belegt"
<i>DY2</i>	Zeiger auf eine Bytevariable zur Ablage des Wochentag-Parameters DY2 {0=SO...6=SA}, 255 entspricht "nicht belegt"
<i>on_hour</i>	Zeiger auf eine Bytevariable zur Ablage der Einschaltzeit-Stunde {0...23}, 255 entspricht "nicht belegt"
<i>on_minute</i>	Zeiger auf eine Bytevariable zur Ablage der Einschaltzeit-Minute {0...59}, 255 entspricht "nicht belegt"
<i>off_hour</i>	Zeiger auf eine Bytevariable zur Ablage der Ausschaltzeit-Stunde {0...23}, 255 entspricht "nicht belegt"
<i>off_minute</i>	Zeiger auf eine Bytevariable zur Ablage der Ausschaltzeit-Minute {0...59}, 255 entspricht "nicht belegt"

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 12 Der Baustein ist im Programm nicht vorhanden.
- 13 Der Kanal kann nicht überschrieben werden, da kein Parameter hinterlegt ist.

Achtung:

Bei easy800 und MFD-Titan muss im angegebenen Kanal mindestens ein Bestandteil bereits mit einem Wert belegt worden sein, andernfalls gilt der Kanal als nicht verwendet.

Unlock_Device

long Unlock_Device (unsigned char net_id, const char * szPassword, unsigned char * errorcode)

Schließt ein Systempasswort-geschütztes Gerät auf.

Sofern im Gerät ein Passwort aktiv ist, kann mit dieser Funktion das Gerät aufgeschlossen werden. Hierzu muss das im Gerät hinterlegte Systempasswort im Parameter 'szPassword' als Dezimalwert-Zeichenkette übergeben werden. Bei easy500 und easy700 besteht das Passwort aus 4 Dezimalziffern, bei easy800 und MFD-Titan besteht es aus 6 Dezimalziffern. Das Passwort wird verschlüsselt an das Gerät übertragen und dort geprüft. Entspricht das gesendete Passwort nicht dem im Gerät hinterlegten Passwort wird eine zusätzliche Wartepause eingelegt, um einen automatischen Test aller Kombinationen zu erschweren.

Der Übergabeparameter 'net_id' identifiziert das Zielgerät.

Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Wenn in allen easyNet Teilnehmern ein Passwort aktiv ist und mit allen kommuniziert werden soll, muss Unlock_Device für jeden Teilnehmer separat aufgerufen werden.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'errorcode' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>szPassword</i>	Geräte-Systempasswort als 0-terminierte Zeichenkette { "0001"... "9999" bzw. "000001"... "999999" }
<i>errorcode</i>	Zeiger auf eine Bytevariable zur Ablage des vom Gerät gesendeten Fehlercodes.

Rückgabewerte:

0	Funktionsaufruf war erfolgreich.
1	Ein Parameter enthielt einen ungültigen Wert.
2	Gerät übermittelte eine Fehlermeldung.
3	Gerät antwortet nicht.
6	Fehlermeldung von Windows, Fehlercode mit GetLastSystemError abfragen.
7	Es ist keine Verbindung offen.
8	Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
9	Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
10	Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
11	Interner Fehler der EASY_COM.dll

Achtung:

Durch das Aufschließen des Gerätes sind auch die zuvor verriegelten Menüpunkte am Gerät aufrufbar.
Nach ca. 10 Minuten Inaktivität reaktivieren die Geräte den Passwortschutz automatisch.

Lock_Device

long Lock_Device (unsigned char net_id, unsigned char * errorcode)

Reaktiviert das Systempasswort am Gerät.

Sofern im Gerät ein Passwort eingetragen ist und das Gerät zuvor aufgeschlossen wurde, kann mit dieser Funktion der Passwortschutz wieder reaktiviert werden.

Der Übergabeparameter 'net_id' identifiziert das Zielgerät.

Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Wenn in allen easyNet Teilnehmern ein Passwort aktiv ist und alle Teilnehmer aufgeschlossen worden sind, muss die Funktion für jeden Teilnehmer separat aufgerufen werden.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'errorcode' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>errorcode</i>	Zeiger auf eine Bytevariable zur Ablage des vom Gerät gesendeten Fehlercodes.

Rückgabewerte:

- | | |
|----|---|
| 0 | Funktionsaufruf war erfolgreich. |
| 1 | Ein Parameter enthielt einen ungültigen Wert. |
| 2 | Gerät übermittelte eine Fehlermeldung. |
| 3 | Gerät antwortet nicht. |
| 6 | Fehlermeldung von Windows, Fehlercode mit GetLastError abfragen. |
| 7 | Es ist keine Verbindung offen. |
| 8 | Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch) |
| 9 | Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch). |
| 10 | Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt) |
| 11 | Interner Fehler der EASY_COM.dll |

4.2 Funktionen für den Mehrfachverbindungsmodus

Im Mehrfachverbindungsmodus können mehrere Verbindungen gleichzeitig geöffnet und verwendet werden, sofern diese nicht dieselbe Schnittstelle und nicht dasselbe easy Gerät ansprechen. Beim Öffnen einer neuen Verbindung wird ein Identifikationsmerkmal („Handle“) zurückgegeben, das beim Aufruf der anderen Kommunikationsfunktionen mit anzugeben ist. Über das „Handle“ erfolgt die Auswahl der Verbindung.

Außer dem zusätzlichen „Handle“ Parameter unterscheiden sich die Funktionen nicht von denen des Einzelverbindungsmodus. Daher beginnen die Funktionsnamen alle mit dem Präfix „MC_“.

```
typedef void* tEasyComHandle;
```

MC_CloseAll

```
long MC_CloseAll( )
```

Schließt alle easy-Com Verbindungen

Die Funktion beendet alle noch offenen Verbindungen und gibt die Systemressourcen wieder frei. Sie kann auch zum Schließen der Verbindung verwendet werden, die im Einzelverbindungsmodus geöffnet worden ist.

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.

MC_Open_ComPort

long MC_Open_ComPort(tEasyComHandle *phandle, unsigned char com_port_nr, long baudrate)

Stellt eine Verbindung zum Gerät über den angegebenen COM-Port her.

Die Funktion öffnet die angeforderte serielle Schnittstelle und konfiguriert diese für das easy-COM Protokoll. Anstelle einer seriellen Schnittstelle kann sich auch ein virtueller COM-Port eines Umsetzer-Treibers verbergen, insbesondere der eines EASY(800)-USB-CAB Programmierkabels.

Kommt keine Verbindung zum Gerät zustande, wird ein Verbindungsaufbau mit allen in Frage kommenden Baudraten versucht. Kann die Verbindung zum Gerät hergestellt werden, wird anschließend das Gerät auf die gewünschte Baudrate umkonfiguriert. Welche Baudrate letztendlich eingestellt wird, hängt von den Fähigkeiten des Geräts sowie der verwendeten Programmierkabel ab. Bei erfolgreicher Verbindungsaufnahme kann anschließend mit der Funktion [MC_GetCurrent_Baudrate](#) die tatsächlich eingestellte Baudrate ermittelt werden.

Die Gerätefamilien easy500 und easy700 unterstützen generell nur 4800 Baud.

Bei erfolgreicher Verbindungsaufnahme wird über den Parameter 'p_hPort' ein Handle-Wert zurückgeliefert, mit dem die Verbindung in den nachfolgenden Kommunikationsaufrufen selektiert wird.

Wenn die Schnittstelle des Geräts mit einem System-Passwort geschützt ist, liefert die Funktion den Fehlercode 4 zurück. Die Verbindung ist zwar zum Gerät hergestellt worden, aber es muss anschließend die Funktion [MC_Unlock_Device](#) aufgerufen werden, um weiter mit dem Gerät kommunizieren zu können.

Parameter:

phandle Zeiger auf eine Handle-Variable
com_port_nr Nummer des COM-Ports, der geöffnet werden soll, z.B. 3 für 'COM3' { 1...255 }
baudrate gewünschte Baudrate { 4800, 9600, 19200, 38400, 57600 }

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 4 Die Schnittstelle des Geräts ist mit einem Passwort geschützt. MC_Unlock_Device aufrufen!
- 5 Gerätetyp ist unbekannt.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastSystemError](#) abfragen.
- 8 Der COM-Port ist nicht vorhanden oder derzeit nicht im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll

MC_Open_EthernetPort

long MC_Open_EthernetPort(
tEasyComHandle *phandle, const char * szIpAddress, long IpPort, long baudrate, bool no_baudrate_scan)

Stellt über Ethernet bzw. Internet eine Verbindung zum Gerät her.

Die Funktion öffnet die angeforderte TCP/IP-Verbindung und konfiguriert diese für das easy-COM Protokoll. Die Konfiguration der TCP/IP-Verbindung selbst muss also zuvor bereits vorgenommen worden sein (siehe [Systemvoraussetzungen](#)).

Kommt keine Verbindung zum Gerät zustande, wird ein Verbindungsaufbau mit allen in Frage kommenden Baudraten versucht. Kann die Verbindung zum Gerät hergestellt werden, wird anschließend das Gerät auf die gewünschte Baudrate umkonfiguriert. Welche Baudrate letztendlich eingestellt wird, hängt von den Fähigkeiten des Geräts sowie der verwendeten Programmierkabel ab. Bei erfolgreicher Verbindungsaufnahme kann anschließend mit der Funktion [MC_GetCurrent_Baudrate](#) die tatsächlich eingestellte Baudrate ermittelt werden.

Der Baudraten-Test kann aufgrund der längeren TCP/IP-Wartezeiten bis zu einer Minute dauern. Daher sollte für easy500 und easy700 der Parameter 'no_baudrate_scan' auf 'true' gesetzt werden, um den Baudraten-Test abzuschalten (die Geräte unterstützen generell nur 4800 baud).

Dasselbe gilt auch, wenn die Verbindung nicht über das Ethernet-Gateway EASY209-SE hergestellt werden soll. In diesem Fall müssen Gerät und Umsetzer bereits auf die angegebene Baudrate konfiguriert worden sein oder das Gerät besitzt eine Ethernet-Schnittstelle onboard, so dass die Baudrate für die Verbindungsaufnahme bedeutungslos ist.

Die in Frage kommenden TCP/IP-Portnummern sind geräteabhängig: Für easy und MFD-Titan, die über ein EASY209-SE Gateway angebunden sind, ist der Wertebereich 10001 bis 10999 möglich.

Bei erfolgreicher Verbindungsaufnahme wird über den Parameter 'p_hPort' ein Handle-Wert zurückgeliefert, mit dem die Verbindung in den nachfolgenden Kommunikationsaufrufen selektiert wird.

Parameter:

<i>phandle</i>	Zeiger auf eine Handle-Variable
<i>szIpAddress</i>	IPv4-Adresse als 0-terminierte Zeichenkette, in der Regel 4-stellig z.B. "10.1.41.31"
<i>IpPort</i>	IP-Portnummer { 1200, 10001...10999 }
<i>baudrate</i>	eingestellte Baudrate im Gerät { 4800, 9600, 19200, 38400, 57600 }
<i>no_baudrate_scan</i>	Schaltet den Baudraten-Test ab. 'true' => nur die angegebene Baudrate wird getestet

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 4 Die Schnittstelle des Geräts ist mit einem Passwort geschützt. [MC_Unlock_Device](#) aufrufen!
- 5 Gerätetyp ist unbekannt.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastSystemError](#) abfragen.
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 14 TCP/IP Teilnehmer antwortet nicht oder der TCP/IP Port ist bereits von einer anderen offenen Verbindung belegt
- 15 Baudraten-Test ist nicht möglich

MC_Close_ComPort

long MC_Close_ComPort (tEasyComHandle handle)

Beendet die Verbindung zum Gerät.

Die Funktion beendet die Verbindung zum Gerät und gibt die serielle Schnittstelle wieder frei.

Parameter:

handle das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.
16 das Verbindung-Handle ist nicht (mehr) gültig

MC_Close_EthernetPort

long MC_Close_EthernetPort (tEasyComHandle handle)

Beendet die Verbindung zum Gerät.

Die Funktion beendet die Verbindung zum Gerät und gibt die Ethernet-Verbindung wieder frei.

Parameter:

handle das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.
16 das Verbindung-Handle ist nicht (mehr) gültig

MC_GetCurrent_Baudrate

long MC_GetCurrent_Baudrate (tEasyComHandle handle, long * baudrate)

Liefert die eingestellte Baudrate der angegebenen Verbindung zurück.

Die Funktion liefert nach Aufruf von [MC_Open_ComPort](#) oder [MC_Open_EthernetPort](#) die tatsächlich eingestellte Baudrate zurück. Welche Baudrate beim Verbindungsaufbau eingestellt wird, hängt von den Fähigkeiten des Geräts, der verwendeten Programmierkabel, der Verbindungsart und ggf. von den Fähigkeiten von zwischengeschalteten Umsetzern ab.

Nach Aufruf von MC_Open_EthernetPort ohne Baudratentest ist der Rückgabewert der Funktion undefiniert.

Parameter:

handle das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
baudrate Zeiger auf eine Wortvariable zur Ablage der aktuellen Baudrate
 { 4800, 9600, 19200, 38400, 57600 }

Rückgabewerte:

0 Funktionsaufruf war erfolgreich.
1 Ein Parameter enthielt einen ungültigen Wert.
16 das Verbindung-Handle ist nicht (mehr) gültig.

MC_Set_UserWaitingTime

long Set_UserWaitingTime (long timeout_delay)

Stellt eine zusätzliche Timeout-Verzögerung ein.

Die Funktion stellt eine zusätzliche Verzögerungszeit ein, auf die bei jeder Geräteanfrage gewartet werden soll, bevor ein Verbindungsabbruch gemeldet wird. Diese Einstellung gilt für alle Verbindungen, die mit der DLL geöffnet werden.

Da Funk oder Modem-Verbindungen einen längeren Übertragungsweg haben, müssen auf der PC-Seite längere Wartezeiten eingestellt werden. Die Funktion sollte bereits vor dem Funktionsaufruf zum Öffnen einer Verbindung aufgerufen werden.

Ein Wert größer 0 führt aber zwangsläufig auch zu einer Verzögerung, bis ein tatsächlich eingetretener Verbindungsabbruch erkannt wird. Daher sollte der Wert nicht größer als unbedingt nötig gewählt werden.

Je nach Übertragungsweg und verwendetem Protokollumsetzer treten unterschiedliche Verzögerungen ein, so dass kein allgemeiner Richtwert für die Verzögerungszeit genannt werden kann. Ein erster Testlauf kann z.B. mit 800ms gestartet werden, treten dabei noch Verbindungsabbrüche auf, sollte der Wert schrittweise erhöht werden.

Als Startwert ist 0 für direkte serielle Verbindungen zwischen PC und Gerät voreingestellt.

Parameter:

timeout_delay
Verzögerungszeit in Millisekunden { 0...90000 }

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.

MC_Get_UserWaitingTime

long Get_UserWaitingTime (long* timeout_delay)

Liefert die zuletzt eingestellte zusätzliche Timeout-Verzögerung zurück.

Parameter:

timeout_delay
Zeiger auf eine Variable zur Ablage der Verzögerungszeit { 0...90000 }

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.

MC_Start_Program

long MC_Start_Program (tEasyComHandle handle, unsigned char net_id, unsigned char * errorcode)

Schaltet das Gerät in 'RUN'.

Die Funktion schaltet das Gerät in den RUN Modus. Sofern die Anzeige am Gerät sich noch nicht in der Grundstellung befindet, wird diese zuvor automatisch zurückgesetzt (noch nicht übernommene Benutzer-eingaben gehen verloren).

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1 bis 8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'errorcode' (siehe [Fehlercodes der Geräte](#)).

Achtung:

Für den Wechsel nach RUN kann das Gerät mehrere Sekunden benötigen!

Ist das Gerät ein easyNet Master (Teilnehmer NT1), können aufgrund der 'Remote RUN' - Option weitere Geräte nach RUN umschalten.

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>errorcode</i>	Zeiger auf eine Bytevariable zur Ablage des vom Gerät gesendeten Fehlercodes.

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 16 das Verbindung-Handle ist nicht (mehr) gültig

MC_Stop_Program

long MC_Stop_Program (tEasyComHandle handle, unsigned char net_id, unsigned char * errorcode)

Schaltet das Gerät in 'STOP'.

Die Funktion schaltet das Gerät in den STOP Modus. Sofern die Anzeige am Gerät sich noch nicht in der Grundstellung befindet, wird diese zuvor automatisch zurückgesetzt (noch nicht übernommene Benutzer-eingaben gehen verloren).

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'errorcode' (siehe [Fehlercodes der Geräte](#)).

Achtung:

Ist das Gerät ein easyNet Master (Teilnehmer NT1), so können aufgrund der 'Remote RUN' - Option weitere Geräte nach STOP schalten.

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>errorcode</i>	Zeiger auf eine Bytevariable zur Ablage des vom Gerät gesendeten Fehlercodes.

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 16 das Verbindung-Handle ist nicht (mehr) gültig

MC_Read_Clock

long MC_Read_Clock (tEasyComHandle handle, unsigned char net_id,
unsigned char* year, unsigned char* day, unsigned char* hour, unsigned char* min)

Auslesen der Geräteuhr.

Die Funktion liest das Datum und die Uhrzeit aus dem Gerät aus.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ‚year‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>year</i>	Zeiger auf eine Bytevariable zur Ablage der Jahreszahl {0...99}, 0 entspricht Jahr 2000
<i>month</i>	Zeiger auf eine Bytevariable zur Ablage des Monats {1...12}
<i>day</i>	Zeiger auf eine Bytevariable zur Ablage des Kalendertags {1...31}
<i>hour</i>	Zeiger auf eine Bytevariable zur Ablage der Stunde {0...23}, 24h-Format
<i>min</i>	Zeiger auf eine Bytevariable zur Ablage der Minute {0...59}

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 16 das Verbindung-Handle ist nicht (mehr) gültig

MC_Write_Clock

long MC_Write_Clock (tEasyComHandle handle, unsigned char net_id,
unsigned char* year, unsigned char* month, unsigned char* day, unsigned char* hour, unsigned char* min)

Stellen der Geräteuhr.

Die Funktion schreibt das Datum und die Uhrzeit in das Gerät.

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ,year' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>year</i>	Zeiger auf eine Bytevariable zur Ablage der Jahreszahl {0...99}, 0 entspricht Jahr 2000
<i>month</i>	Zeiger auf eine Bytevariable zur Ablage des Monats {1...12}
<i>day</i>	Zeiger auf eine Bytevariable zur Ablage des Kalendertags {1...31}
<i>hour</i>	Zeiger auf eine Bytevariable zur Ablage der Stunde {0...23}, 24h-Format
<i>min</i>	Zeiger auf eine Bytevariable zur Ablage der Minute {0...59}

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 16 das Verbindung-Handle ist nicht (mehr) gültig

MC_Read_Object_Value

long MC_Read_Object_Value (tEasyComHandle handle, unsigned char net_id,
unsigned char object, unsigned short index, unsigned char length, unsigned char * data)

Auslesen des Prozessabbilds oder des Merkerbereichs

Die Funktion liest aus dem Gerät einen Bereich des Prozessabbilds oder des Merkerbereichs. Der Übergabeparameter 'object' legt den Bereich des Prozessabbilds und damit auch die ausgelesene Datenmenge fest (Ausnahme: bei Objekt 10 und 17 wird mit dem Parameter 'length' die Datenmenge vorgegeben).

0	Digital-Eingänge Basisgerät	I1...I16	2 Byte
1	Digital-Ausgänge Basisgerät	Q1...Q8	2 Byte
2	Digital-Eingänge Erweiterungsmodul	R1...R16	2 Byte
3	Digital-Ausgänge Erweiterungsmodul	S1...S8	2 Byte
4	Bit-Merker bei easy800 und MFD-Titan	M1...M96	12 Byte
4	Bit-Merker bei easy500 und easy700	M1...M16	2 Byte
4	Bit-Merker bei easy800-DC-SWD	M1...M128	16 Byte
5	P-Tasten	P1...P4	1 Byte
6	easyNet-Eingänge	xI1... xI16 und xR1...xR16	4 Byte
7	easyNet-Ausgänge	xQ1...xQ8 und xS1...xS8	4 Byte
8	Analog-Eingänge Basisgerät	IA1...IA4	8 Byte
9	Analog-Ausgang Basisgerät	QA1	2 Byte
10	1 bis 20 Doppelwort-Merker MDx	MDx...MDx+n	4...80 Byte
11	easyNet Empfangsdaten	xRN1...xRN32	32 Byte
12	easyNet Sendedaten	xSN1...xSN32	32 Byte
13	Diagnosebits	ID1...ID16	2 Byte
14	zusätzliche Diagnosebits bei MFD-Titan	ID17...ID32	2 Byte
15	programmierbare LEDs bei MFD-Titan	LE1...LE3	1 Byte
16	zusätzliche Bit-Merker bei easy500/700	N1...N16	2 Byte
17	1 bis 255 Merkerbyte	MBx...MBx+n	1...255 Byte
18	Digital-Eingänge Basisgerät und zugeordnete SWD-Eingänge	I1...I128	16 Byte
19	Digital-Ausgänge Basisgerät und zugeordnete SWD-Ausgänge	Q1...Q128	16 Byte

Weitere Informationen entnehmen Sie bitte dem Bedienungshandbuch zum jeweiligen Gerät.

Die Zustandswerte werden im Little-Endian-Format (Intel-Format) zurückgegeben. Der Zustandswert des Operanden mit der kleinsten Operandennummer steht im geringwertigsten Bit (LSB).

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im ersten Byte des Übergabeparameters 'data' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>object</i>	Nummer des Objekts also des Prozessabbildbereichs
<i>index</i>	Teilnehmernummer bei Objekt 6, 7, 11, 12 {1...8} Operandennummer bei Objekt 10 {1...96} bzw. {1...128} Bytenummer bei Objekt 17 {1...384} bzw. {1...512} Sonst ohne Funktion
<i>length</i>	Objekt 10: Durch 4 teilbare Anzahl der zu lesenden Byte {4...80}. Objekt 17: Anzahl der zu lesenden Byte {1...255} an Sonst ohne Funktion.
<i>data</i>	Zeiger auf einen Speicherbereich zur Ablage der Zustandswerte

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 16 das Verbindung-Handle ist nicht (mehr) gültig

Achtung:

Die Größe des Datenfeldes, das mit ‚data‘ übergeben wird, muss ausreichend groß dimensioniert sein, da sonst über die Speichergrenze hinausgeschrieben wird.

Bei Objekt 10 kann generell nur ein Vielfaches von 4 Byte gelesen werden. Das Objekt 17 erlaubt den byteweisen Zugriff auf den gesamten Merkerbereich, auch wenn das Gerät nur die Operanden MB1 bis MB96 kennt. Verwenden Sie – wenn immer möglich – den effizienteren Zugriff über Objekt 10.

MC_Write_Object_Value

long MC_Write_Object_Value (tEasyComHandle handle, unsigned char net_id,
unsigned char object, unsigned short index, unsigned char length, unsigned char * data)

Schreibt in den Merkerbereich.

Die Funktion schreibt Werte in den Merker-Bereich des Geräts.

Der Übergabeparameter ‚object‘ legt den Typ des Merker-Bereichs fest:

4	einzelner Bit-Merker Mx	1 Byte
10	1 bis 80 Merkerbyte ab Doppelwort-Merker MDx	1...80 Byte
16	einzelner Bit-Merker Nx bei easy500 und easy700	1 Byte
17	1 bis 80 Merkerbyte	1...80 Byte

Die Zustandswerte werden im Little-Endian-Format (Intel-Format) interpretiert. Der zu setzende Zustandswert des Bitmerkers muss im geringwertigsten Bit (LSB) eingetragen werden.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1 bis 8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im ersten Byte des Übergabeparameters ‚data‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>object</i>	Nummer des Objekts also des Prozessabbildbereichs
<i>index</i>	Operandennummer bei Objekt 4, 10, 16 {1...16} bzw. {1...96} bzw. {1...128} Bytenummer bei Objekt 17 {1...384} bzw. {1...512}
<i>length</i>	Gibt bei Objekttyp 10 und 17 die Anzahl der zu schreibenden Byte {1...80} an. Sonst ohne Funktion.
<i>data</i>	Zeiger auf einen Speicherbereich zur Ablage der Zustandswerte

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System registriert (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht im System registriert (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 16 das Verbindung-Handle ist nicht (mehr) gültig

Achtung:

Die Größe des Datenfeldes, das mit dem Parameter ‚data‘ übergeben wird, muss ausreichend groß dimensioniert sein, da sonst über die Speichergrenze hinaus gelesen wird.

Der Zustandswert eines geschriebenen Merkers bleibt nur erhalten, wenn das Gerät kein Programm ausführt, das seinerseits auf diesen Merker schreibend zugreift.

Das Objekt 17 erlaubt den byteweisen Zugriff auf den gesamten Merkerbereich, auch wenn das Gerät nur für den unteren Merkerbereich einen Merkerbyte-Zugriff kennt. Verwenden Sie – wenn immer möglich – den effizienteren Zugriff über Objekt 10.

MC_Read_Channel_YearTimeSwitch

long MC_Read_Channel_YearTimeSwitch(tEasyComHandle handle,
unsigned char net_id, unsigned char index, unsigned char channel,
unsigned char * on_year, unsigned char * on_month, unsigned char * on_day,
unsigned char * off_year, unsigned char * off_month, unsigned char * off_day)

Parameter eines Jahresschaltuhrkanals auslesen.

Die Funktion liest aus dem angegebenen Jahresschaltuhr-Baustein den Parametersatz des selektierten Kanals aus. Der Parametersatz besteht aus den Bestandteilen ‚Kalendertag‘, ‚Monat‘ und ‚Jahr‘ jeweils getrennt nach Einschaltzeitpunkt und Ausschaltzeitpunkt.

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'on_year' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>index</i>	Gibt die Bausteinnummer an {1...4} bei easy500/700 bzw. {1...32} bei easy800/MFD-Titan.
<i>channel</i>	Gibt den Kanal an {0...3}, 0 entspricht Kanal A, 3 entspricht Kanal D
<i>on_year</i>	Zeiger auf eine Bytevariable zur Ablage der <i>Einschaltzeitpunkt</i> - Jahreszahl {0, 1...99}, 0 entspricht "nicht belegt"
<i>on_month</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Einschaltzeitpunkt</i> - Monats {0, 1...12}, 0 entspricht "nicht belegt"
<i>on_day</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Einschaltzeitpunkt</i> - Kalendertags {0, 1...31}, 0 entspricht "nicht belegt"
<i>off_year</i>	Zeiger auf eine Bytevariable zur Ablage der <i>Ausschaltzeitpunkt</i> - Jahreszahl {0, 1...99}, 0 entspricht "nicht belegt"
<i>off_month</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Ausschaltzeitpunkt</i> - Monats {0, 1...12}, 0 entspricht "nicht belegt"
<i>off_day</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Ausschaltzeitpunkt</i> - Kalendertags {0, 1...31}, 0 entspricht "nicht belegt"

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es wurde kein COM-Port geöffnet.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 12 Der Baustein ist im Programm nicht vorhanden.
- 16 das Verbindung-Handle ist nicht (mehr) gültig

Achtung:

Bei easy800 und MFD-Titan liefert das Gerät im STOP generell nur Nullwerte.

MC_Write_Channel_YearTimeSwitch

long MC_Write_Channel_YearTimeSwitch(tEasyComHandle handle,
unsigned char net_id, unsigned char index, unsigned char channel ,
unsigned char * on_year, unsigned char * on_month, unsigned char * on_day,
unsigned char * off_year, unsigned char * off_month, unsigned char * off_day)

Parameter eines Jahresschaltuhrkanals überschreiben.

Die Funktion überschreibt in dem angegebenen Jahresschaltuhr-Baustein den Parametersatz des selektierten Kanals mit neuen Werten. Der Parametersatz besteht aus den Bestandteilen ‚Kalendertag‘, ‚Monat‘ und ‚Jahr‘ jeweils getrennt nach Einschaltzeitpunkt und Ausschaltzeitpunkt.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ‚on_year‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>index</i>	Gibt die Bausteinnummer an {1...4} bei easy500/700 bzw. {1...32} bei easy800/MFD-Titan.
<i>channel</i>	Gibt den Kanal an {0...3}, 0 entspricht Kanal A und 3 entspricht Kanal D
<i>on_year</i>	Zeiger auf eine Bytevariable zur Ablage der <i>Einschaltzeitpunkt</i> - Jahreszahl {0, 1...99}, 0 entspricht "nicht belegt"
<i>on_month</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Einschaltzeitpunkt</i> - Monats {0, 1...12}, 0 entspricht "nicht belegt"
<i>on_day</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Einschaltzeitpunkt</i> - Kalendertags {0, 1...31}, 0 entspricht "nicht belegt"
<i>off_year</i>	Zeiger auf eine Bytevariable zur Ablage der <i>Ausschaltzeitpunkt</i> - Jahreszahl {0, 1...99}, 0 entspricht "nicht belegt"
<i>off_month</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Ausschaltzeitpunkt</i> - Monats {0, 1...12}, 0 entspricht "nicht belegt"
<i>off_day</i>	Zeiger auf eine Bytevariable zur Ablage des <i>Ausschaltzeitpunkt</i> - Kalendertags {0, 1...31}, 0 entspricht "nicht belegt"

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es wurde kein COM-Port geöffnet.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 12 Der Baustein ist im Programm nicht vorhanden.
- 13 Der Kanal kann nicht überschrieben werden, da kein Parameter hinterlegt ist.
- 16 das Verbindung-Handle ist nicht (mehr) gültig

Achtung:

Bei easy800 und MFD-Titan muss im angegebenen Kanal mindestens ein Bestandteil bereits mit einem Wert belegt worden sein, andernfalls gilt der Kanal als nicht verwendet.

MC_Read_Channel_7DayTimeSwitch

```
long MC_Read_Channel_7DayTimeSwitch(tEasyComHandle handle,  
    unsigned char net_id, unsigned char index, unsigned char channel,  
    unsigned char * DY1, unsigned char * DY2,  
    unsigned char * on_hour, unsigned char * on_minute,  
    unsigned char * off_hour, unsigned char * off_minute)
```

Parameter eines Wochenschaltuhrkanals auslesen.

Die Funktion liest aus dem angegebenen Wochenschaltuhr-Baustein den Parametersatz des selektierten Kanals aus.

Der Übergabeparameter ‚net_id‘ identifiziert das Zielgerät.

Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ‚DY1‘ (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>index</i>	Gibt die Bausteinnummer an {1...4} bei easy500/700 bzw. {1...32} bei easy800/MFD-Titan.
<i>channel</i>	Gibt den Kanal an {0...3}, 0 entspricht Kanal A und 3 entspricht Kanal D
<i>DY1</i>	Zeiger auf eine Bytevariable zur Ablage des Wochentag-Parameters DY1 {0=SO...6=SA}, 255 entspricht "nicht belegt"
<i>DY2</i>	Zeiger auf eine Bytevariable zur Ablage des Wochentag-Parameters DY2 {0=SO...6=SA}, 255 entspricht "nicht belegt"
<i>on_hour</i>	Zeiger auf eine Bytevariable zur Ablage der Einschaltzeit-Stunde {0...23}, 255 entspricht "nicht belegt"
<i>on_minute</i>	Zeiger auf eine Bytevariable zur Ablage der Einschaltzeit-Minute {0...59}, 255 entspricht "nicht belegt"
<i>off_hour</i>	Zeiger auf eine Bytevariable zur Ablage der Ausschaltzeit-Stunde {0...23}, 255 entspricht "nicht belegt"
<i>off_minute</i>	Zeiger auf eine Bytevariable zur Ablage der Ausschaltzeit-Minute {0...59}, 255 entspricht "nicht belegt"

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 12 Der Baustein ist im Programm nicht vorhanden.
- 16 das Verbindung-Handle ist nicht (mehr) gültig

Achtung:

Bei easy800 und MFD-Titan liefert das Gerät im STOP generell nur Nullwerte.

MC_Write_Channel_7DayTimeSwitch

```
long MC_Write_Channel_7DayTimeSwitch(tEasyComHandle handle,  
    unsigned char net_id, unsigned char index, unsigned char channel,  
    unsigned char * DY1, unsigned char * DY2,  
    unsigned char * on_hour, unsigned char * on_minute,  
    unsigned char * off_hour, unsigned char * off_minute)
```

Parameter eines Wochenschaltuhrkanals überschreiben.

Die Funktion überschreibt in dem angegebenen Wochenschaltuhr-Baustein den Parametersatz des selektierten Kanals mit neuen Werten.

Der Übergabeparameter ,net_id' identifiziert das Zielgerät.

Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter ,DY1' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>index</i>	Gibt die Bausteinnummer an {1...4} bei easy500/700 bzw. {1...32} bei easy800/MFD-Titan.
<i>channel</i>	Gibt den Kanal an {0...3}, 0 entspricht Kanal A und 3 entspricht Kanal D
<i>DY1</i>	Zeiger auf eine Bytevariable zur Ablage des Wochentag-Parameters DY1 {0=SO...6=SA}, 255 entspricht "nicht belegt"
<i>DY2</i>	Zeiger auf eine Bytevariable zur Ablage des Wochentag-Parameters DY2 {0=SO...6=SA}, 255 entspricht "nicht belegt"
<i>on_hour</i>	Zeiger auf eine Bytevariable zur Ablage der Einschaltzeit-Stunde {0...23}, 255 entspricht "nicht belegt"
<i>on_minute</i>	Zeiger auf eine Bytevariable zur Ablage der Einschaltzeit-Minute {0...59}, 255 entspricht "nicht belegt"
<i>off_hour</i>	Zeiger auf eine Bytevariable zur Ablage der Ausschaltzeit-Stunde {0...23}, 255 entspricht "nicht belegt"
<i>off_minute</i>	Zeiger auf eine Bytevariable zur Ablage der Ausschaltzeit-Minute {0...59}, 255 entspricht "nicht belegt"

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 12 Der Baustein ist im Programm nicht vorhanden.
- 13 Der Kanal kann nicht überschrieben werden, da kein Parameter hinterlegt ist.
- 16 das Verbindung-Handle ist nicht (mehr) gültig

Achtung:

Bei easy800 und MFD-Titan muss im angegebenen Kanal mindestens ein Bestandteil bereits mit einem Wert belegt worden sein, andernfalls gilt der Kanal als nicht verwendet.

MC_Unlock_Device

long MC_Unlock_Device (
tEasyComHandle handle, unsigned char net_id, const char * szPassword, unsigned char * errorcode)

Schließt ein Systempasswort-geschütztes Gerät auf.

Sofern im Gerät ein Passwort aktiv ist, kann mit dieser Funktion das Gerät aufgeschlossen werden. Hierzu muss das im Gerät hinterlegte Systempasswort im Parameter 'szPassword' als Dezimalwert-Zeichenkette übergeben werden. Bei easy500 und easy700 besteht das Passwort aus 4 Dezimalziffern, bei easy800 und MFD-Titan besteht es aus 6 Dezimalziffern.

Das Passwort wird verschlüsselt an das Gerät übertragen und dort geprüft. Entspricht das gesendete Passwort nicht dem im Gerät hinterlegten Passwort wird eine zusätzliche Wartepause eingelegt, um einen automatischen Test aller Kombinationen zu erschweren.

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Wenn in allen easyNet Teilnehmern ein Passwort aktiv ist und mit allen kommuniziert werden soll, muss Unlock_Device für jeden Teilnehmer separat aufgerufen werden.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'errorcode' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>szPassword</i>	Geräte-Systempasswort als 0-terminierte Zeichenkette { "0001"... "9999" bzw. "000001"... "999999" }
<i>errorcode</i>	Zeiger auf eine Bytevariable zur Ablage des vom Gerät gesendeten Fehlercodes.

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 16 das Verbindung-Handle ist nicht (mehr) gültig

Achtung:

Durch das Aufschließen des Gerätes sind auch die zuvor verriegelten Menüpunkte am Gerät aufrufbar.
Nach ca. 10 Minuten Inaktivität reaktivieren die Geräte den Passwortschutz automatisch.

MC_Lock_Device

long MC_Lock_Device (tEasyComHandle handle, unsigned char net_id, unsigned char * errorcode)

Reaktiviert das Systempasswort am Gerät.

Sofern im Gerät ein Passwort eingetragen ist und das Gerät zuvor aufgeschlossen wurde, kann mit dieser Funktion der Passwortschutz wieder reaktiviert werden.

Der Übergabeparameter 'net_id' identifiziert das Zielgerät. Der Wert 0 bezeichnet das Gerät, an dem das Programmierkabel angeschlossen ist. Der Wert 1...8 wählt den Teilnehmer mit der entsprechenden NET-ID aus.

Wenn in allen easyNet Teilnehmern ein Passwort aktiv ist und alle Teilnehmer aufgeschlossen worden sind, muss die Funktion für jeden Teilnehmer separat aufgerufen werden.

Liefert die Funktion den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im Übergabeparameter 'errorcode' (siehe [Fehlercodes der Geräte](#)).

Parameter:

<i>handle</i>	das beim Öffnen der Verbindung zurück gelieferte Verbindung-Handle
<i>net_id</i>	Zielgerät {0...8}; 0 für lokale Programmierschnittstelle
<i>errorcode</i>	Zeiger auf eine Bytevariable zur Ablage des vom Gerät gesendeten Fehlercodes.

Rückgabewerte:

- 0 Funktionsaufruf war erfolgreich.
- 1 Ein Parameter enthielt einen ungültigen Wert.
- 2 Gerät übermittelte eine Fehlermeldung.
- 3 Gerät antwortet nicht.
- 6 Fehlermeldung von Windows, Fehlercode mit [GetLastError](#) abfragen.
- 7 Es ist keine Verbindung offen.
- 8 Der COM-Port ist nicht mehr vorhanden oder nicht mehr im System eingetragen (Verbindungsabbruch)
- 9 Der COM-Port wird durch einen anderen Prozess blockiert oder ist nicht mehr im System eingetragen (Verbindungsabbruch).
- 10 Allgemeiner Kommunikationsfehler eingetreten (evtl. Hardware-Defekt)
- 11 Interner Fehler der EASY_COM.dll
- 16 das Verbindung-Handle ist nicht (mehr) gültig

5. Anhang

5.1 Fehlercodes der Geräte

Liefert eine der Schnittstellenfunktionen den Rückgabewert 2 zurück, wurde der Auftrag vom Gerät zurückgewiesen. Der vom Gerät übermittelte Fehlercode steht dann im ersten Byte (LSB) des Übergabeparameters ‚data‘ bzw. ‚errorcode‘.

Die Bedeutung des vom Gerät zurückgegebenen Fehlercodes ist den folgenden Tabellen zu entnehmen. Ist der Fehlercode nicht in den Tabellen aufgeführt, kann dies auf eine gestörte Datenübertragung hindeuten, wenn z.B. parallel zum PC noch weitere Kommunikationszugriffe auf das Gerät erfolgen („Terminalmodus“ mit MFD-CP4).

Fehlercodes von easy500, easy700

Code	Bedeutung
2	Funktion/Operation wird nicht unterstützt
3	Funktion/Operation wird nicht unterstützt
4	Instanznummer ist ungültig
5	Parameterwert ist ungültig
6	Schreibzugriff auf einen Baustein-Parameter, der keine Konstante enthält
12	Funktion/Operation ist nur im STOP zulässig
13	Display-Anzeige befindet sich nicht in Grundstellung (Eingabe läuft)
16	Das übermittelte Passwort stimmt nicht mit dem im Gerät hinterlegten Passwort überein.
101	Passwortschutz ist aktiv. Gerät muss erst aufgeschlossen werden.

Fehlercodes von easy800, MFD-Titan

Code	Bedeutung
6	Funktion/Operation wird nicht unterstützt
7	Funktion/Operation wird nicht unterstützt
8	Ungültige easyNet ID
9	easyNet ID ist größer 0, das Gerät ist aber nicht für easyNet konfiguriert
10	Instanznummer ist ungültig
18	Funktion/Operation ist nur im STOP zulässig
20	Schreibzugriff auf ein nur lesbares Objekt
22	Angegebene Objektgröße ist falsch
26	Angeforderter Baustein ist nicht im Programm enthalten
96	Passwortschutz ist aktiv. Gerät muss erst aufgeschlossen werden.
97	Passwortschutz ist aktiv. Gerät muss erst aufgeschlossen werden.
98	Passwortschutz ist aktiv. Gerät muss erst aufgeschlossen werden.
99	Passwortschutz ist aktiv. Gerät muss erst aufgeschlossen werden.
100	Passwortschutz ist aktiv. Gerät muss erst aufgeschlossen werden.
101	Passwortschutz ist aktiv. Gerät muss erst aufgeschlossen werden.
102	Passwortschutz ist aktiv. Gerät muss erst aufgeschlossen werden.
103	Das übermittelte Passwort stimmt nicht mit dem im Gerät hinterlegten Passwort überein.
112	Gerät ist nicht für den easyNet-Betrieb konfiguriert. Keine Kommunikation über easyNet möglich.
113	easyNet ist außer Funktion
114	Kommunikation gestört. Das Gerät kommuniziert bereits mit einer anderen Gegenstelle.
115	easyNet ist außer Funktion
116	Kommunikation gestört. Das Gerät kommuniziert bereits mit einer anderen Gegenstelle.
126	Kommunikation gestört. Das Gerät kommuniziert bereits mit einer anderen Gegenstelle.
133	Die Display-Anzeige befindet sich nicht in Grundstellung (Eingabe läuft)
135	Die Aktion kann nicht ausgeführt werden, weil ein SWD-Konfigurationsfehler vorliegt oder der Verdrahtungstest noch eingeschaltet ist.
176	Das Stellen der Geräteuhr bzw. Setzen der Sommerzeit-Konfiguration wurde wegen ungültiger Parameter zurückgewiesen
225	Kommunikation gestört. Das Gerät kommuniziert bereits mit einer anderen Gegenstelle.
226	Kommunikation gestört. Das Gerät kommuniziert bereits mit einer anderen Gegenstelle.

5.2 Weiterführende Informationen

Bedienungshandbücher

MN05013003Z...	easy500/700
MN04902001Z...	easy800
MN05002001Z...	MFD-Titan
MN05013012Z...	EASY209-SE
MN05006004Z...	Datenübertragung zwischen easy- und IEC-Steuerungen

Montageanweisungen

IL05013015Z...	easy500/700
IL05013012Z...	easy800
IL05013014Z...	MFD-Titan
IL05013021Z...	Programmierkabel EASY800-MO-CAB
IL05013019Z...	EASY209-SE

Anwendungshinweise

USBDriverInst_...	Programmierkabel EASY-USB-CAB / EASY800-USB-CAB
Onlinehilfe der easySoft	
Onlinehilfe des EASY209-SE Konfigurators V2	
AN_EASYNET_...	Inbetriebnahme des easyNet

Internetadressen

Support	http://www.moeller.net/easy
Produktkatalog	http://ecat.moeller.net/
FTP-Server	ftp://ftp.moeller.net/EASY/
easy-Forum	http://www.easy-forum.net/

5.3 Glossar

easyNet	Ein auf CAN basierendes Netzwerk für easy Steuerrelais. Innerhalb eines Strangs können bis zu 8 Geräte vernetzt werden.
easyCOM	Kommunikationsprotokoll der easy Steuerrelais für den Zugriff per Programmiergerät.
Teilnehmernummer NET-ID	Eindeutige Adresse eines easyNet-Teilnehmers (1...8)
easyNet Master	Der Teilnehmer mit der Teilnehmernummer 1. Dies ist immer das 1. Gerät in einem Strang.
easyNet Slave	Ein Teilnehmer mit der Teilnehmernummer 2 bis 8.
Remote-RUN Option	Eine Betriebsart des Slaves. Ist die Option aktiv, folgt der Slave dem RUN/STOP Zustand des Masters.
Remote-IO-Option	Eine Betriebsart des Slaves („dezentrales E/A“). Ist die Option aktiv, arbeitet der Slave kein Programm ab und erlaubt dem Master, die Ausgänge des Slaves direkt anzusteuern.
Prozessabbild	Datenbereich, der die Zustände aller Ein- und Ausgänge eines Geräts zusammenfasst.
Zeiger	Eine Variable, die die Adresse eines Datenspeichers enthält.