

Bedienungsanleitung

Easy control „Windows SDK (HTTP/DLL)“

Anleitung für den Programmierer

D-ID: V1_3 – 09.15



Inhaltsverzeichnis

1	Nutzungshinweise	2
2	Einleitung	3
3	Kompatibilität	4
4	Installation	5
5	Konfiguration	7
6	Interner Aufbau	10
7	HTTP-Schnittstelle	11
8	Verwendung der DLL	12
9	Variablenbeschreibung	15
9.1	Induktionsschleifen	15
9.2	Multirelais	17
9.3	Richtungslogiken	20
9.4	Signalleuchten	22
9.5	Zeitverhalten	23
9.6	Funkfernbedienung	24
9.7	Schrankenstatus	24
9.8	Bedientaster	26
9.9	Wartung	29
9.10	Fehlermeldungen	30
9.11	Schranken-Modell	32
9.12	Weiter Variablen	32
10	Kommunikationsdienst einbinden	35
10.1	Benötigte Dateien	35
10.2	Konfigurationsdatei	37
10.3	Konfiguration und Initialisierung über DLL	41

1 Nutzungshinweise



Diese Betriebsanleitung dient dem Anwender dazu, das Produkt sachgerecht zu installieren, in Betrieb zu nehmen und zu bedienen. Vor Beginn jeglicher Arbeiten mit dem Produkt muss die Betriebsanleitung vollständig gelesen und verstanden werden.



Fertigen Sie vor der Installation dieser Software eine Sicherung der Daten auf Ihrem Computer an.



Technische Änderungen im Sinne des Fortschritts vorbehalten.

2 Einleitung

Das Windows SDK dient der Vereinfachung der Softwareentwicklung zur Ansprache der MO 64 „Plus“ und MO 24 Schrankensteuerung. Mit nur wenigen Programmzeilen Quelltext können Befehle an die Schrankensteuerung gesendet und Zustände abgefragt werden.

Die Basis des Windows SDK ist der Dienst "ELKA Communication Service". Dieser stellt die HTTP- und eine TCP-Kommunikationsschnittstelle (über eine DLL) zur Verfügung und muss auf einem zentralen PC/Server installiert werden. Verbindungen werden ganz einfach im Konfigurationsprogramm parametrisiert. Der Dienst übernimmt die Kommunikation mit der Schrankensteuerung und stellt alle Zustände und Konfigurationsparameter als Variablen zur Verfügung. Diese Variablen lassen sich per HTTP-Abfrage setzen und auslesen. Zur Vereinfachung der Implementierung stellen wir ein Kommandozeilen-Programm und eine DLL zur Verfügung.

Dieses Dokument beschreibt die Installation u. Konfiguration des Windows SDK und die Schnittstellen, die zur Verfügung gestellt werden, sowie die Einbindung in Ihr eigenes Projekt.

**HINWEIS!**

Beachten Sie für den ordnungsgemäßen Betrieb der Schrankenanlage innerhalb der CEN Staaten unbedingt auch die gültigen europäischen sicherheitsrelevanten Richtlinien und Normen. ELKA-Torantriebe GmbH u.Co. Betriebs KG übernimmt keine Gewährleistung.

3 Kompatibilität

TCP/IP- Kommunikation

Die Software ist mit allen Windows-unterstützten Ethernet-Adapterkompatibel. Derzeit sind keine Einschränkungen bekannt.

RS485-Kommunikation

Die Software ist konfiguriert für alle RS485-Adapter mit einem Windows COM-Port.

Getestet wurden folgende Adapter:

- Exsys EX 1303
- Exsys EX-1335HM
- Exsys EX-1339HMV

Betriebssystem

Getestet wurde die Software unter den folgenden Betriebssystemen:

- Windows 7 (32-Bit / 64-Bit)
- Windows 8.1 (32-Bit / 64-Bit)
- Windows 10 (32-Bit / 64-Bit)
- Windows 2008 Server (32-Bit / 64-Bit)
- Windows 2012 Server (32-Bit / 64-Bit)

Technischer Stand der Steuerungen

Diese Software ist mit folgenden Versionsständen der ELKA Schrankensteuerung kompatibel:

- 1 MO 64 "Plus" - Softwarestand 2V3
- 2 MO 24 - Softwarestand 1V2
- 3 MO 24 - Softwarestand 1V3
- 4 MO 24 - Softwarestand 1V4

Die technischen Angaben beziehen sich auf den technischen Stand der ELKA Motorteuerungen mit den o.a. Softwareversionen (Stand 25.09.2015).

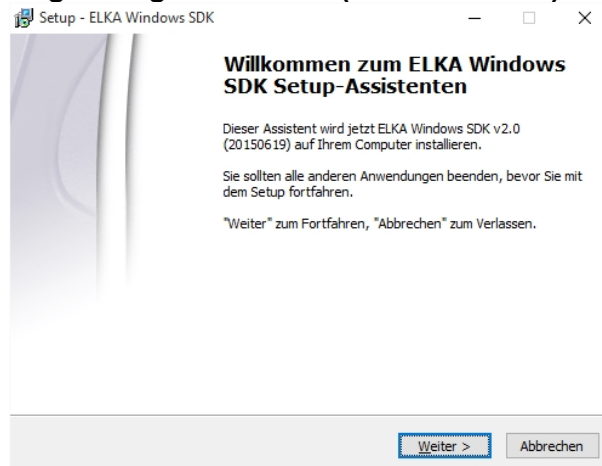
4 Installation

Installationsvoraussetzungen:

- min. 2000MB freier Festplattenplatz
- TCP/IP Protokoll installiert (Netzwerkkarte vorhanden)
- Administratorrechte vorhanden (die Datei SETUP.EXE muss als Administrator ausgeführt werden)

Installationsablauf:

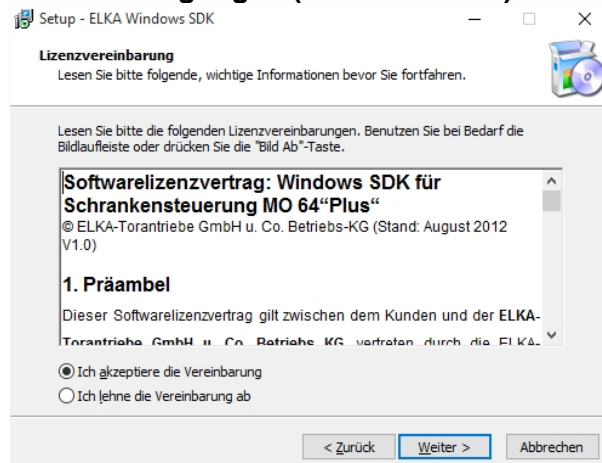
1. Begrüßungsbildschirm (Schritt 1 von 5)



Zum Fortfahren auf den Button "Next >" klicken.

Zum Abbrechen der Installation auf den Button "Cancel" klicken.

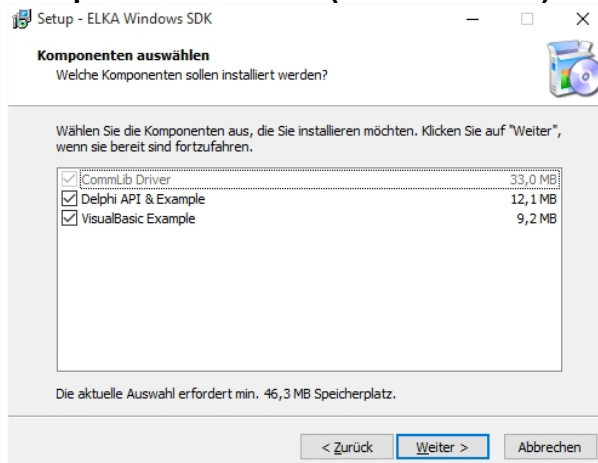
2. Lizenzbedingungen (Schritt 2 von 5)



Bitte lesen Sie die Lizenzbedingungen und bestätigen Sie danach die Auswahl bei „Ich akzeptiere die Vereinbarungen“. Zum Fortfahren auf den Button "Next >" klicken.

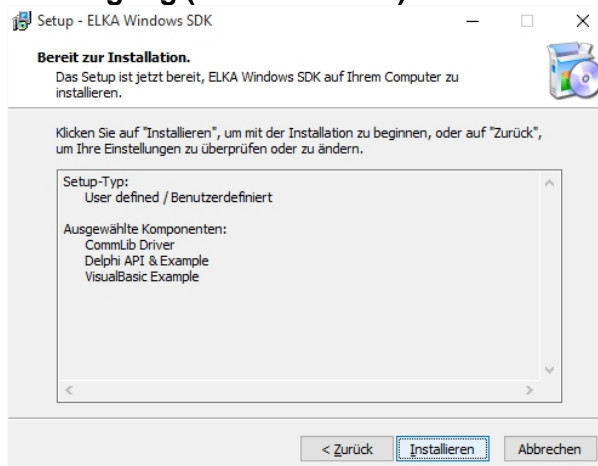
Zum Abbrechen der Installation auf den Button "Cancel" klicken.

3. Komponentenauswahl (Schritt 3 von 5)



Wählen Sie die gewünschten Komponenten über die Checkboxes aus.
 Zum Fortfahren auf den Button "Next >" klicken.
 Zum Abbrechen der Installation auf den Button "Cancel" klicken.

4. Bestätigung (Schritt 4 von 5)



Es werden noch einmal alle ausgewählten Einstellungen angezeigt. Für die Installation klicken Sie jetzt auf den Button "Install".
 Zum Abbrechen der Installation auf den Button "Cancel" klicken.

5. Fertigstellen (Schritt 5 von 5)

Um die Installation abzuschließen muss der Rechner ggf. neu gestartet werden.

Nach der Installation muss der PC neu gestartet werden.

5 Konfiguration

Das Konfigurations-Programm wird bei der Windows-Anmeldung automatisch geladen. Es dient zur Konfiguration des Dienstes und der Parametrierung der Verbindungen.

Zum Anzeigen des Konfigurationsfensters, unten rechts am Bildschirm, auf das ELKA Symbol im Windows-Tray doppelklicken.



Abbildung 1

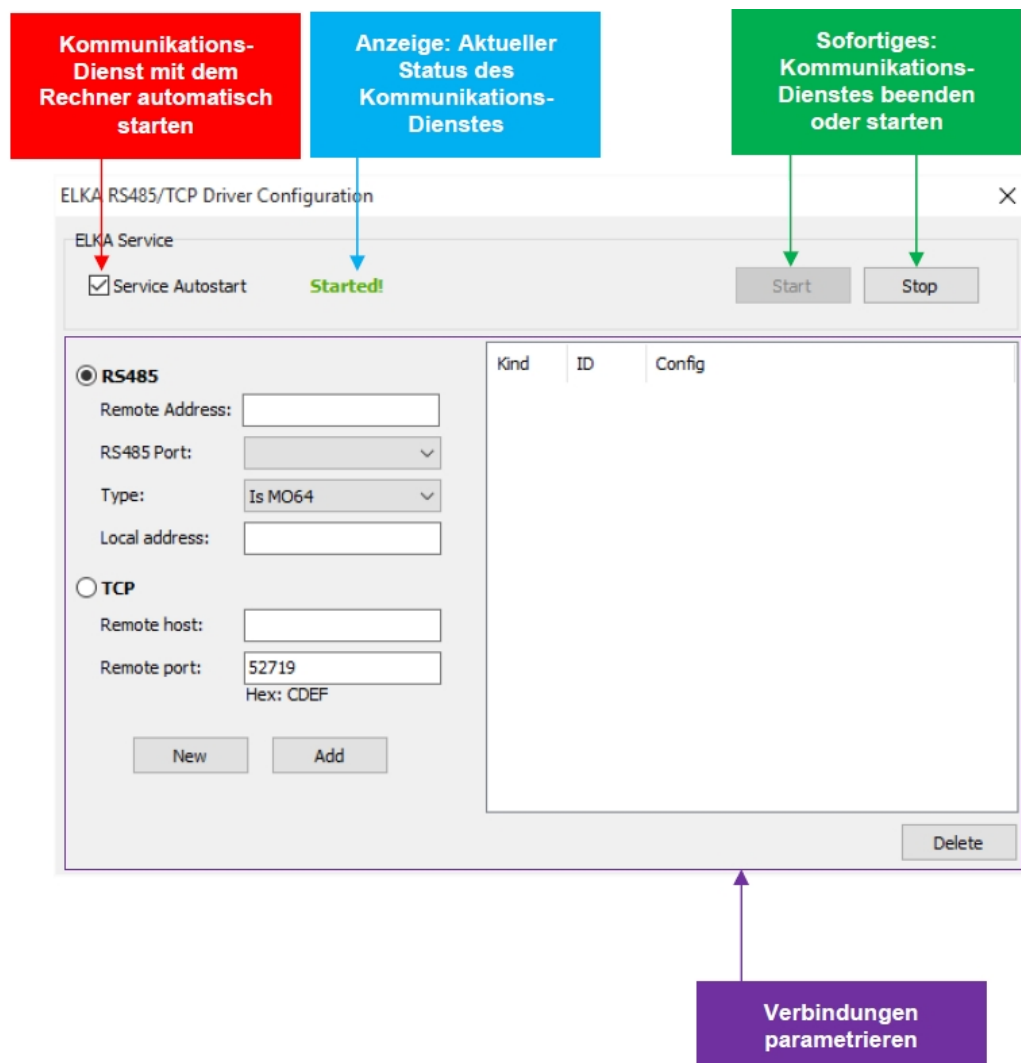


Abbildung 2

ELKA SERVICE (WINDOWS-DIENST) AUTOSTART



WICHTIG!

Diese Funktion steht nur dem „Administrator“ zur Verfügung.

Der Windows-Dienst "ELKA Service" kommuniziert mit den angeschlossenen Steuerungen. Um zu verhindern, dass dieser Dienst beim Windows-Start automatisch geladen wird, muss das Häkchen vor "Service Autostart" entfernt werden.

Um den Autostart wieder zu aktivieren, muss das Häkchen erneut gesetzt werden.

Wenn auf dem PC neben der eigenen Lösung auch der „Easy control operator“ oder „Easy control configurator“ verwendet werden soll, muss der Autostart aktiviert werden und der Dienst laufen.

ELKA SERVICE (WINDOWS-DIENST) AUTOSTART



WICHTIG!

Diese Funktion steht nur dem „Administrator“ zur Verfügung.

Zum Beenden der Kommunikation mit den Steuerungen den Windows-Dienst über das Klicken auf den Button "Stop" beenden. Zum erneuten Starten der Kommunikation auf den Button "Start" klicken. Der aktuelle Status wird links neben den Buttons angezeigt

Not started	Der Dienst wurde beendet.
Starting...	Der Dienst wird gestartet, läuft jedoch noch nicht.
Closing...	Der Dienst wird beendet, ist jedoch noch nicht beendet.
Started!	Der Dienst wurde gestartet.
Unknown	Fehlermeldung: Evtl. wurde der Dienst nicht ordnungsgemäß installiert oder deinstalliert.

Tabelle 1

Nach Konfigurationsänderungen muss der Windows-Dienst neu gestartet werden. Dafür erst auf den Button "Stop" klicken, bis der Status links neben den Buttons "Not started" anzeigt. Danach auf den Button "Start" klicken und warten bis der Status auf "Started!" wechselt.

VERBINDUNG HINZUFÜGEN

Zum Hinzufügen einer neuen Verbindung wie folgt vorgehen:

- Auf den Button "New" klicken.
- Entweder auf den Punkt "RS485" oder "TCP" unter der Verbindungsliste klicken.
 - Bei RS485:
 - Die Bus-Adresse des Bus-Teilnehmers in das Feld "Remote Address" eintragen.

- Den COM-Port an dem die Steuerung angeschlossen wurde im Feld "RS485 Port" auswählen.
- Auswählen, ob es sich bei dem Bus-Teilnehmer um einen weiteren PC "Is another PC" oder eine Steuerung "Is Barrier" handelt.
- Die Lokale Bus-Adresse im Feld "Local Address" eintragen. Beim einzigen PC im Bus die Adresse "0" eintragen.
- Bei TCP:
 - Die IP-Adresse der Steuerung in das Feld "Host" eintragen.
 - Die in der Steuerung eingestellte Port-Nummer eintragen(Normalerweise: 52719).
- Auf den Button "Add" am unteren Fensterrand klicken.

Die Verbindung wird nun in der Verbindungsliste angezeigt. Es müssen alle Steuerungen zu denen Kontakt aufgebaut werden soll und alle angeschlossenen Slave-PCs am Bus parametrieren werden. Nach Änderungen an den Verbindungen muss der Dienst neu gestartet werden.



WICHTIG!

Bei RS485 Multi-Master-Systemen (mit mehreren PCs am Bus) muss mindestens ein Folge-PC eingetragen werden. Die Master bilden einen logischen Ring. Wenn ein Master seine Anfrage gestellt hat, gibt er die Kontrolle an den nächsten Master auf dem Bus weiter. Es muss auf jedem PC der folgende Partner auf dem Bus eingetragen werden. Auf dem letzten PC muss wieder der erste Master eingetragen werden. Im Zweifelsfall kann man auf allen PCs alle anderen PCs eintragen. In diesem Fall entscheidet der Dienst automatisch über den nachfolgenden Busteilnehmer.

VERBINDUNG ENTFERNEN

Zum Entfernen einer parametrieren Verbindung wie folgt vorgehen:

- Den Verbindungseintrag in der Liste durch Anklicken wählen.
- Auf den Button "Delete" am unteren rechten Fensterrand klicken.

Der Eintrag wird nun aus der Verbindungsliste gelöscht. Nach Änderungen an den Verbindungen muss der Dienst neu gestartet werden.

VERBINDUNG VERÄNDERN

Zum Ändern einer parametrieren Verbindung wie folgt vorgehen:

- Den Verbindungseintrag in der Liste durch Anklicken wählen. Der Button "Add" wird in "Save" umbenannt.
- Die gewünschten Einstellungen anpassen.
- Auf den Button "Save" am unteren rechten Fensterrand klicken.

Der Eintrag wird nun in der Verbindungsliste verändert. Nach Änderungen an den Verbindungen muss der Dienst neu gestartet werden.

6 Interner Aufbau

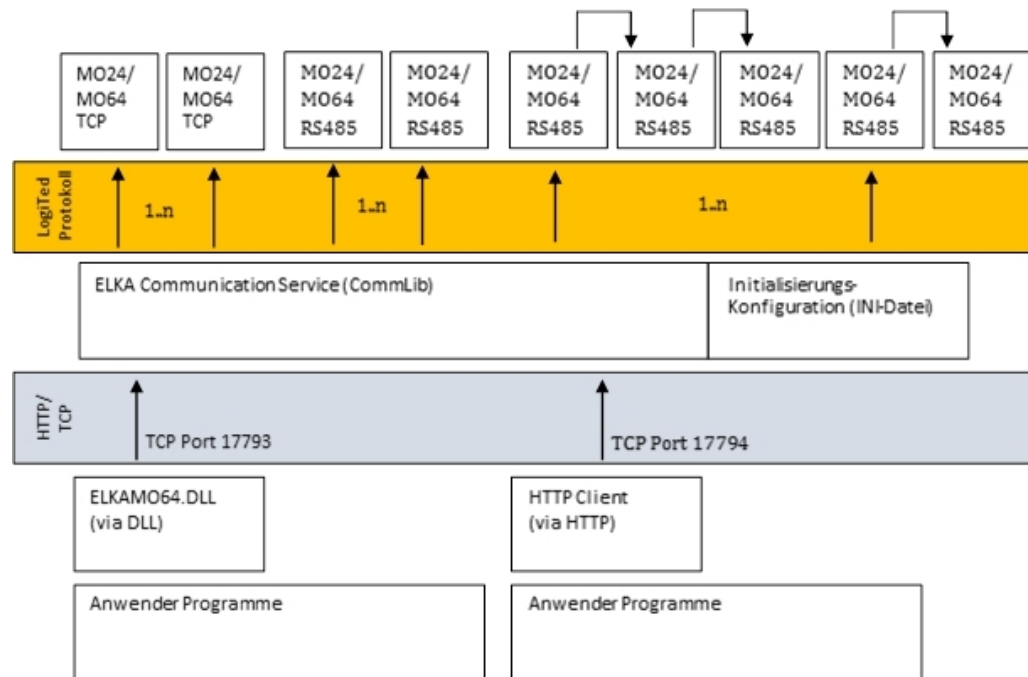


Abbildung 3

Die Steuerungen nutzen das LogiTed-Protokoll zur Kommunikation via RS485 oder TCP. Für die TCP-Kommunikation wird i.d.R. der Ziel-TCP-Port 52719 verwendet. Die Beschreibung des LogiTed Protokolls entnehmen Sie bitte der Montageanleitung "Protokollbeschreibung".

Der CommLib-Dienst ist ein TCP-Server der als Windows-Dienst gestartet wird. Zur Kommunikation mit Fremdanwendungen über die ELKAMO64.DLL wird der TCP-Port 17793 verwendet. Zur Kommunikation mit Fremdanwendungen über das HTTP-Protokoll wird der TCP-Port 17794 verwendet.

Jede Steuerung akzeptiert nur eine TCP-Verbindung auf der LogiTed-LogiTed-Protokollebene. Das Windows SDK bietet die Möglichkeit mehrere Clients mit einer Steuerung kommunizieren zu lassen. Dazu muss der Dienst auf einem Rechner installiert werden, der von allen Clients erreicht werden kann.

7 HTTP-Schnittstelle

Nahezu jede Programmiersprache bietet die Möglichkeit per HTTP Anfragen an einen sogenannten Server zu stellen. Der Dienst bietet einen HTTP-Server zum Lesen und Schreiben der Variablen.

Variablen werden immer mit Typ-Prefix übermittelt und müssen beim Schreiben auch einen solchen erhalten. Mögliche Typen:

- **N** Numerisch (Ganzzahl)
- **F** Floating-Point (Achtung: Dezimalpunkt verwenden!)
- **S** Textfeld
- **B** Ein/Aus bzw. Wahr/Falsch (Achtung: Ein = -1 und Aus = 0)

In unseren Beispielen verwenden wir „localhost“ als Rechnernamen und 17794 als TCP-Port. Diese können frei geändert werden.

EINZEL-VARIABLE LESEN

HTTP-Anfrage:

<http://localhost:17794/GetVar/BarrierID/VarName>

(Die BarrierID muss durch die in der Konfiguration parametrisierte ID ersetzt werden. Der VarName muss durch den gewünschten Variablennamen ersetzt werden.)

Rückgabe Dokument:
Variable=Typ&Wert

Ergebnis ("OK" bzw. "ERROR" mit einer Fehlerbeschreibung.)

ALLE VARIABLE LESEN

HTTP-Anfrage:

http://localhost:17794/GetVar/BarrierID/*

Die BarrierID muss durch die in der Konfiguration parametrisierte ID ersetzt werden.

Rückgabe Dokument:
Variable1=Typ&Wert

...

VariableN=Typ&Wert

Ergebnis ("OK" bzw. "ERROR" mit einer Fehlerbeschreibung.)

Das Zeilentrennzeichen: ASCII #10

VARIABLE SCHREIBEN

HTTP-Anfrage:

<http://localhost:17794/SetVar/BarrierID/VarName/TypWert>

(Die BarrierID muss durch die in der Konfiguration parametrisierte ID ersetzt werden. Der VarName muss durch den gewünschten Variablennamen; TypWert durch den zu schreibenden Variableninhalt und Typ ersetzt werden.)

Rückgabe Dokument:

Ergebnis ("OK" bzw. "ERROR" mit einer Fehlerbeschreibung.)

8 Verwendung der DLL

Die DLL erleichtert das Kommunizieren mit dem Dienst. Es werden Funktionen zum Lesen und Schreiben von Variablen zur Verfügung gestellt. Bei der ELKAMO64.DLL handelt es sich um eine Windows 32bit DLL. Diese ist threadsicher und kann somit von mehreren Threads gleichzeitig verwendet werden.

Variablen werden immer mit Typ-Prefix übermittelt und müssen beim Schreiben auch einen solchen erhalten. Mögliche Typen:

- **N** Numerisch (Ganzzahl)
- **F** Floating-Point (Achtung: Dezimalpunkt verwenden!)
- **S** Textfeld
- **B** Ein/Aus bzw. Wahr/Falsch (Achtung: Ein = -1 und Aus = 0)

EINZEL-VARIABLE LESEN

DLL-Funktion:

```
function ReadVar(
  AHost: PWideChar;
  APort: Word;
  AInterfaceAddr: Integer;
  AVarName: PWideChar;
  var AVarKindValue: PWideChar;
  AVarKindValueSize: DWord;
  var AConnected: Boolean
): Boolean; stdcall;
```

Folgende Parameter müssen an die DLL-Funktion übergeben werden:

AHost	Rechnername oder IP auf dem der Dienst läuft.
APort	TCP-Port des Dienstes. (I.d.R. die Voreinstellung 17793)
AInterfaceAddr	ID Nummer der Schranke von der gelesen werden soll.
AVarName	Der Name der Variablen die gelesen werden soll.
AVarKindValueSize	Maximaler Speicher in Byte der für die Rückgabe in AVarKindValue reserviert wurde.

Tabelle 2

Parameter die von der DLL-Funktion zurückgegeben werden:

AVarKindValue	<p>Rückgabe der Variable im Format: Variablenname=Typ&Wert</p> <p>Beispiel für eine Gleitkomma-Variable: Variablenname =F123.456</p> <p>Beispiel für eine Text-Variable:</p>
---------------	--

	Variablenname =Dies ist ein Test
AConnected	Die Verbindung zum Dienst ist aufgebaut.
Result	Die Variable wurde erfolgreich gelesen.

Tabelle 3

ALLE VARIABLE LESEN

DLL-Funktion:

```
function ReadVars(
  AHost: PWideChar;
  APort: Word;
  AInterfaceAddr: Integer;
  AVarName: PWideChar;
  var AVarKindValueList: PWideChar;
  AVarKindValueSize: DWord;
  var AConnected: Boolean
): Boolean; stdcall;
```

AHost	Rechnername oder IP auf dem der Dienst läuft.
APort	TCP-Port des Dienstes. (I.d.R. die Voreinstellung 17793)
AInterfaceAddr	ID Nummer der Schranke von der gelesen werden soll.
AVarKindValueListSize	Maximaler Speicher in Byte der für die Rückgabe in AVarKindValueList reserviert wurde.

Tabelle 4

Parameter die von der DLL-Funktion zurückgegeben werden:

AVarKindValueList	Rückgabe der Variable im Format: Variable1=Typ&Wert ... VariableN=Typ&Wert Das Zeilentrennzeichen: ASCII #10
AConnected	Die Verbindung zum Dienst ist aufgebaut.
Result	Die Variable wurde erfolgreich gelesen.

Tabelle 5

VARIABLEN SCHREIBEN

DLL-Funktion:

```
function WriteVar(
  AHost: PWideChar;
  APort: Word;
  AInterfaceAddr: Integer;
  AVarName: PWideChar;
  AVarVaulue: PWideChar;
): Boolean; stdcall;
```

Folgende Parameter müssen an die DLL-Funktion übergeben werden:

AHost	Rechnername oder IP auf dem der Dienst läuft.
APort	TCP-Port des Dienstes. (I.d.R. die Voreinstellung 17793)
AInterfaceAddr	ID Nummer der Schranke von der gelesen werden soll.
AVarName	Der Name der Variable die geschrieben werden soll.
AVarValue	Variable im Format: Typ&Wert Beispiel für eine Gleitkomma-Variable: F123.456 Beispiel für eine Text-Variable: Dies ist ein Test

Tabelle 6

Parameter die von der DLL-Funktion zurückgegeben werden:

Result	Die Variable wurde erfolgreich an den Dienst übermittelt.
--------	---

Tabelle 7

9 Variablenbeschreibung

Der Dienst übernimmt die Kommunikation mit der Schrankensteuerung und stellt alle Zustände und Konfigurationsparameter als Variablen zur Verfügung. Die einzelnen Variablen verfügen über unterschiedliche Zugriffsrechte. Es gilt: Variablen mit dem Präfix „Config.“ können gelesen und geschrieben werden. Variablen mit dem Präfix „Command.“ können nur geschrieben werden und Variablen mit dem Präfix „Status.“ oder „Alarms.“ können nur gelesen werden. Des Weiteren verfügt jede Variablen über einen Typ. Dieser definiert den möglichen Inhalt der Variablen. Mögliche Variablentypen:

- **N** Numerisch (Ganzzahl)
Beispiel: 1000
- **F** Floating-Point (Achtung: Dezimalpunkt verwenden!)
Beispiel: 1000.50
- **S** Textfeld (Unicode)
Beispiel: Dies ist ein Test
- **B** Ein/Aus bzw. Wahr/Falsch (Achtung: Ein = -1 und Aus = 0)
Beispiel: -1

9.1 Induktionsschleifen

Die drei Schleifen können über die folgenden Variablen abgefragt bzw. konfiguriert werden. Variablen die für die Schleifen „a“, „b“ und „c“ existieren sind mit „{a|b|c}“ markiert. Dieser Text ist beim Lesen / Schreiben durch die entsprechende Schleife zu ersetzen.

Variable	Typ	Beschreibung	
Config.InductionLoop_{a b c}. Active	B	Induktionsschleifen einschalten	
		-1	Die Schleife ist eingeschaltet
		0	Die Schleife ist ausgeschaltet
Config.InductionLoop_{a b c}. Sensitivity	N	Empfindlichkeit	
		0	Niedrigste
	
		7	Höchste
Config.InductionLoop_{a b c}. HoldingTime	N	Haltezeit	
		0	Kürzeste
	
		3	Längste
Config.InductionLoop_{a b c}. Behaviour.DriveOnAndSave	N	Beim Befahren der Schleife	
		0	Keine Funktion
		1	Sichern
		2	Öffnen

		3	Öffnen u. Sichern
		4	Schließen
Config.InductionLoop_{a b c}. Behaviour.Leave	B	Beim Verlassen der Schleife	
		0	Keine Funktion
		1	Schließen
Config.InductionLoop_{a b c}. Behaviour.Presence	N	Anwesenheit	
		0	Keine Funktion
		1	Anwesenheit melden, wenn die Schleife belegt ist
		2	Anwesenheitsmeldung sperrern, wenn die Schleife belegt ist
Config.InductionLoops.SaveMode	B	Induktionsschleifen Sichern-Modus	
		0	Stopp u. Öffnen
		1	Stopp
Command.AdjustLoop_{a b c}	B	Schleifenabgleich	
		-1	Abgleich von Schleife auslösen
		0	Keine Funktion
Command.ClearAdjustment Counter_{a b c}	B	Schleifenabgleich-Zähler löschen	
		-1	Löschen des Schleifenabgleich- Zählers
		0	Keine Funktion
Status.InductionLoop_{a b c}. Learning	B	Schleifenabgleich aktiv	
		-1	Ja
		0	Nein
Status.InductionLoop_{a b c}. LearnCounter	N	Schleifenabgleich-Zähler	
			Der Schleifenabgleich- Zähler zählt jeden abgeschlossenen Abgleichvorgang der Induktionsschleife. Es wird bis max. 9999 gezählt. Danach bleibt der Zähler stehen.
Status.InductionLoop_{ a b c}. Active	B	Schleife belegt	
		-1	Die Schleife ist eingeschaltet und belegt.
		0	Die Schleife ist nicht

			belegt bzw. eingeschaltet.
Status.InductionLoop_{ a b c}. Frequency Current	F	Schleifenfrequenz	
			Aktuelle Schleifenfrequenz in Hz
Status.InductionLoop_{ a b c}. Frequency Average	F	Schleifenfrequenz (Mittelwert)	
			Mittlere Schleifenfrequenz in Hz

Tabelle 8

9.2 Multirelais

Steuerung MO 64 "Plus":

Die Steuerung besitzt vier potentialfreie Multifunktionsrelais (Multi1 bis Multi4), die maximal 24Vdc/1A, sowie zwei Multifunktionsrelais (Multi5 und Multi6), die maximal 230Vac/120W schalten können.

Steuerung MO 24:

Die Steuerung besitzt sechs potentialfreie Multifunktionsrelais (Multi1 bis Multi6), die maximal 24Vdc/1A schalten können. Acht weitere Multifunktionsrelais können über eine optionale Erweiterungskarte angesteuert werden.

Variablen die für die Relais „1“ bis „14“ existieren sind mit „{1..14}“ gekennzeichnet. Dieser Text ist beim Lesen/Schreiben durch das entsprechende Relais zu ersetzen.

Variable	Typ	Beschreibung	
MO 64 Plus: Config.MultiRelay_{1..6} MO 24: Config.MultiRelay_{1..14}	N	Konfigurierter Ausgabewert	
		0	Das Multirelais ist deaktiviert.
		1	Störung. Das Multirelais wird aktiviert, wenn eine Fehlerflagge gesetzt wird.
		2	Wartung. Das Multirelais wird aktiviert, wenn der Wartungszähler größer als das (eingestellte) Wartungsintervall ist.
		3	Baum-Ab-Meldung
		4	Tandembetrieb. Das Relais ist aktiviert, wenn die Schranke nicht in der Endlage ZU steht. Bei der Vorwarnung vor dem Öffnen ist es bereits eingeschaltet.
		5	Rotampel
		6	Grünampel
	7	Warnlicht	

		8	LED-Baumbeleuchtung. In Endlage ZU ist die LED-Baumbeleuchtung eingeschaltet. In der Endlage AUF ist sie ausgeschaltet. Sonst blinkt sie.
		9	Das Relais ist eingeschaltet, wenn die Schranke geöffnet ist. Während der Vorwarnzeit (Räumzeit) vor dem Schließen ist das Relais bereits ausgeschaltet.
		10	Das Relais ist eingeschaltet, wenn die Schranke geschlossen ist. Während der Vorwarnung vor dem Öffnen ist das Relais bereits ausgeschaltet. Diese Betriebsart kann für den Haftmagnet verwendet werden.
		11	Das Relais wird über den Bus gesteuert (Ein-, Ausschalten, Impuls).
		12	Lichtschrankentest
		13	MO 64 Plus: Meldung "Schranke ist notentriegelt". Das Relais ist eingeschaltet, wenn der Kontakt BTS2 geöffnet ist. MO 24: Das Relais ist eingeschaltet, wenn der Schrankenbaum manuell um mehr als den halben Öffnungswinkel bewegt wurde.
		14	Anwesenheit. Relais ist eingeschaltet, wenn die Induktionsschleifenauswertung "Anwesenheit" meldet.
		15	Relais ist eingeschaltet, wenn Schleife A belegt ist.
		16	Relais ist eingeschaltet, wenn Schleife A eingeschaltet, nicht defekt und nicht belegt ist.
		17	Relais gibt einen Impuls aus, wenn die Schleife A belegt wird.
		18	Relais gibt einen Impuls aus, wenn die Schleife A frei wird.

	19	Relais ist eingeschaltet, wenn Schleife B belegt ist.
	20	Relais ist eingeschaltet, wenn Schleife B eingeschaltet, nicht defekt und nicht belegt ist.
	21	Relais gibt einen Impuls aus, wenn die Schleife B belegt wird.
	22	Relais gibt einen Impuls aus, wenn die Schleife B frei wird.
	23	Relais ist eingeschaltet, wenn Schleife C belegt ist.
	24	Relais ist eingeschaltet, wenn Schleife C eingeschaltet, nicht defekt und nicht belegt ist.
	25	Relais gibt einen Impuls aus, wenn die Schleife C frei wird.
	26	Relais gibt einen Impuls aus, wenn die Schleife C frei wird.
	27	Nur MO 24: Relais gibt einen Puls aus, wenn der Fahrzeugzähler inkrementiert wird.
	28	Nur MO 24: Relais gibt einen Puls aus, wenn der Fahrzeugzähler dekrementiert wird.
	29	Nur MO 24: Relais gibt einen Puls aus, wenn die Richtungslogik eine Durchfahrt von rechts erkannt hat.
	30	Nur MO 24: Relais gibt einen Puls aus, wenn die Richtungslogik eine Durchfahrt von links erkannt hat.
	31	Nur MO 24: Das Relais ist eingeschaltet, wenn die Schranke geschlossen ist (Haftmagnet).
	32	Nur MO 24: Das Relais ist eingeschaltet, wenn eine TCP-SOCKET-Verbindung aufgebaut ist.
	33	Nur MO 24: Das Relais ist eingeschaltet, wenn die Netzausfallerkennung angeschlossen ist und die Netzspannung nicht

			ausgefallen ist.
		34	Nur MO 24 (Version 1V3 u. 1V4): Das Relais ist eingeschaltet, wenn der akustische Signalgeber entsprechend der Richtlinie UL 325 eingeschaltet werden soll.
MO 64 Plus: Command.SetMulti{1..6}	N	Multirelais über Bus einschalten	
MO 24: Command.SetMulti{1..14}		0	Setzen/Rücksetzen (Impuls)
		1	Setzen
		2	Rücksetzen
MO 64 Plus: Status.MultiRelay_{1..6}	B	Eingeschaltet aufgrund von Multirelais-Konfiguration	
MO 24: Status.MultiRelay_{1..14}		-1	Eingeschaltet
		0	Nicht eingeschaltet
MO 64 Plus: Status.BusCommand. MultiRelay_{1..6}	B	Eingeschaltet über Bus	
MO 24: Status.BusCommand. MultiRelay_{1..14}		-1	Eingeschaltet
		0	Nicht eingeschaltet

Tabelle9

9.3 Richtungslogiken

Die zwei Richtungslogiken können über die folgenden Variablen abgefragt bzw. konfiguriert werden. Variablen die für die Richtungslogiken „1“ und „2“ existieren sind mit „{1|2}“ gekennzeichnet. Dieser Text ist beim Lesen / Schreiben durch die entsprechende Richtungslogik zu ersetzen.

Variable	Typ	Beschreibung	
Config.DirectionLogic_{1 2}.Channel	N	Schleifenauswahl	
		0	Deaktiviert
		1	Eingangsschleife A Ausgangsschleife B
		2	Eingangsschleife B Ausgangsschleife A
		3	Eingangsschleife A Ausgangsschleife C
		4	Eingangsschleife C Ausgangsschleife A
		5	Eingangsschleife B Ausgangsschleife C
6	Eingangsschleife C Ausgangsschleife B		

Config.DirectionLogic_{1 2}.Mode. CrossFromLeft	N	Öffnen oder Schließen bei Durchfahrt von links	
		0	Keine Funktion
		1	Öffnen
		2	Schließen
Config.DirectionLogic_{1 2}.Mode. CrossFromRight	N	Öffnen oder Schließen bei Durchfahrt von rechts	
		0	Keine Funktion
		1	Öffnen
		2	Schließen
Config.DirectionLogic_{1 2}.Count. CrossFromLeft	N	Fahrzeuge zählen bei der Durchfahrt von links	
		0	Keine Funktion
		1	Zähler inkrementieren
		2	Zähler dekrementieren
Config.DirectionLogic_{1 2}.Count. CrossFromRight	N	Fahrzeuge zählen bei der Durchfahrt von rechts	
		0	Keine Funktion
		1	Zähler inkrementieren
		2	Zähler dekrementieren
Config.DirectionLogic_{1 2}.Mode. DriveOnFromLeft	N	Öffnen o. Schließen bei der Drauffahrt von links	
		0	Keine Funktion
		1	Öffnen
		2	Schließen
Config.DirectionLogic_{1 2}.Mode. DriveOnFromRight	N	Öffnen o. Schließen bei der Drauffahrt von rechts	
		0	Keine Funktion
		1	Öffnen
		2	Schließen
Config.Direction.Logic_{1 2}.Presence. DriveOnFromLeft	N	Anwesenheit melden oder sperrern bei der Drauffahrt von links	
		0	Keine Funktion
		1	Anwesenheit melden
		2	Anwesenheitsmeldung sperrern
Config.Direction.Logic_{1 2}.Presence. DriveOnFromRight	N	Anwesenheit melden oder Sperrern bei der Drauffahrt von rechts	

		0	Keine Funktion
		1	Anwesenheit
		2	Anwesenheitsmeldung sperren
Command.SetCounterVehicles	N	Fahrzeugzähler setzen	
		Der Fahrzeugzähler kann mit einem beliebigen Wert geladen werden und dann durch die Richtungslogik der Induktionsschleifen Fahrzeuge zählen. Kleinster Wert: -2147483648 Größter Wert: 2147483647	
Status.Counter.Vehicles	N	Fahrzeugzähler	
		Mit der Richtungslogik der Induktionsschleifen können Fahrzeuge gezählt werden. Der Fahrzeugzähler steht nur im RAM der Motorsteuerung. Nach dem Einschalten der Motorsteuerung hat der Zähler den Wert 0. In dieser Variable steht der aktuelle Fahrzeugzähler der Richtungslogik.	

Tabelle 10

9.4 Signalleuchten

Die zwei Signalleuchten können über die folgenden Variablen abgefragt bzw. konfiguriert werden. Variablen die für die Signalleuchten „Rot-Ampel“, „Warnung“ und „Grün-Ampel“ existieren, sind mit „{typ}“ gekennzeichnet. Dieser Text ist beim Lesen / Schreiben durch die entsprechende Signalleuchte zu ersetzen. Dabei ist zu beachten dass für die Rot-Ampel „Red“, für die Grün-Ampel „Green“ und für die Warnung „Warn“ verwendet wird.

Variable	Typ	Beschreibung	
Config.SignalLight.{typ}.InClosedEndPosition	N	Bei geschlossener Schranke (Endlage ZU)	
		0	= aus
		1	= ein
		2	= blinken
Config.SignalLight.{typ}.OnPreWarningBeforeOpen	N	Vorwarnung vor dem Öffnen	
		0	= aus
		1	= ein
		2	= blinken
Config.SignalLight.{typ}.	N	Beim Verlassen vor dem Schließen	

OnLeavingBeforeClose		0	= aus
		1	= ein
		2	= blinken
Config.SignalLight.{typ}. WhileOpening	N	Während des Öffnens der Schranke	
		0	= aus
		1	= ein
		2	= blinken
Config.SignalLight.{typ}. WhileClosing	N	Während des Schließens der Schranke	
		0	= aus
		1	= ein
		2	= blinken
Config.SignalLight.{typ}. OnIntermediateStop	N	Bei einem Zwischenstopp	
		0	= aus
		1	= ein
		2	= blinken

Tabelle 11

9.5 Zeitverhalten

Die folgenden Variablen steuern das Zeitverhalten der Schranke.

Variable	Typ	Beschreibung	
Config.Timing.HoldOpen	N	Offenhaltezeit	
		0	Keine Offenhaltezeit
		> 0	Die Offenhaltezeit als ein Vielfaches von 100ms
Config.Timing.Prewarning.Open	N	Vorwarnung vor dem Öffnen	
		0	Keine Vorwarnung vor dem Öffnen
		> 0	Vorwarnung als Vielfaches von 100ms
Config.Timing.Prewarning.Close	N	Vorwarnung vor dem Schließen	
		0	Keine Vorwarnung vor dem Schließen

		> 0	Vorwarnzeit als ein Vielfaches von 100ms
--	--	-----	--

Tabelle 12

9.6 Funkfernbedienung

Über die folgenden Variablen kann die Funkfernbedienung aktiviert bzw. deaktiviert und konfiguriert werden.

Variable	Typ	Beschreibung	
Config.Radio.Enabled	B	Funkfernbedienung aktivieren	
		-1	Aktiviert
		0	Deaktiviert
Config.Radio.Code	S	Funkcode	
		Der eingelernte Funkcode als String. Der Funkcode wird durch 9 Codierschalter A9...A1 am Sender eingestellt. Jeder Codierschalter kann die Zustände „+“, „-“ oder „0“ haben. Diese werden hier zu einem String von 9 Zeichen zusammengesetzt.	
Status.RadioControl	B	Funkcode empfangen	
		-1	Der gelernte Funkcode wird gerade empfangen.
		0	Kein Funkcode empfangen

Tabelle13

9.7 Schrankenstatus

Über die folgenden Variablen kann der aktuelle Schrankenstatus abgefragt werden und das allgemeine Schrankenverhalten konfiguriert werden.

Variable	Typ	Beschreibung	
Config.LightBarrier.Mode	N	Lichtschrankenmodus	
		0	aus
		1	ein
Config.LightBarrier.Observation	B	Lichtschranken -Test	
		0	aus
		1	ein
Config.LightBarrier.CloseAutomatically	B	Lichtschranken-Schließautomatik	
		0	aus

		1	ein
Config.CloseImmediately	B	Sofort Schließen	
		0	aus
		1	ein
Config.DeadmanOperation	B	Totmannbetrieb	
		0	aus
		1	ein
Config.LockAfterWatchdog	B	Nur MO 64 Plus: Verriegelung nach Laufzeitbegrenzung	
		0	aus
		1	ein
Config.BoomMissingFunction	B	Baum-Ab-Funktion	
		0	Kein Stopp
		1	Stopp bei Baum-Ab-Meldung
Command.ClearFlagMD		Löschen der MD-Flagge	
		-1	Kraftumsteuerung war aufgetreten - quittieren
		0	Keine Funktion
Status.LightBarrier	B	Lichtschrankenkontakt	
		-1	Lichtschrankenkontakt ist unterbrochen
		0	Lichtschrankenkontakt ist nicht unterbrochen
Status.BoomMissing	B	Baum-Ab-Kontakt	
		-1	Betätigt
		0	Nicht betätigt
Status.SLZ	B	Sicherheitskontaktprofil	
		-1	Betätigt
		0	Nicht betätigt
Status.Barrier.State	N	Schrankenstellung	
		0	Schranke öffnet
		1	Schranke schließt
		2	Vorwarnung vor dem Öffnen
		3	Vorwarnung vor dem Schließen
4	Schranke ist geöffnet		

		5	Schranke ist geschlossen
		6	Schranke in Zwischenstellung
Status.Barrier.Position	N	Position des Schrankenbaumes	
		-1	Schrankenposition nicht bekannt
		0	Endlage ZU
		...	Zwischenstellung
		100	Endlage AUF
Status.Barrier.Mode	B	Schrankenmodus	
		0	Stopp u. Öffnen
		1	Stopp
Status.TorqueObservation	B	Kraftumsteuerung	
		-1	Eine Kraftumsteuerung ist beim letzten Schließen aufgetreten.
		0	Keine Kraftumsteuerung aufgetreten.
Status.EndPosition.Opened	B	Nur MO 64 Plus: Endlage AUF (SEA)	
		-1	SEA ist betätigt
		0	Nicht betätigt
Status.EndPosition.Closed	B	Nur MO 64 Plus: Endlage ZU (SEZ)	
		-1	SEZ ist betätigt
		0	Nicht betätigt

Tabelle14

9.8 Bedientaster

Über folgende Variablen kann der Bedientaster ausgewählt und der Zustand der unterschiedlichen Taster abgefragt werden. Des Weiteren können die unterschiedlichen Taster simuliert bzw. festgehalten werden um z.B. einen „DAUERAUF“ Befehl zu geben.

Variable	Typ	Beschreibung	
Config.ButtonSelection	N	BT-Kombi-Auswahl	
		0	BT
		1	BTA3

		2	BTZ1B
Command.SetBT	B	Bedientaster (BT) über BUS aktivieren / deaktivieren	
		-1	Aktivieren (Festhalten)
		0	Deaktivieren (Loslassen)
Command.ImpBT	B	Bedientaster (BT) über BUS bedienen	
		-1	Aktivieren und automatisch deaktivieren (Impuls)
		0	Keine Funktion
Command.SetBA	B	Bedientaster (BA/AUF) über BUS aktivieren / deaktivieren	
		-1	Aktivieren (Festhalten)
		0	Deaktivieren (Loslassen)
Command.ImpBA	B	Bedientaster (BA / AUF) über BUS bedienen	
		-1	Aktivieren u. automatisch deaktivieren (Impuls)
		0	Keine Funktion
Command.SetBZ	B	Bedientaster (BZ / ZU) über BUS aktivieren/deaktivieren	
		-1	Aktivieren (Festhalten)
		0	Deaktivieren (Loslassen)
Command.ImpBZ	B	Bedientaster (BZ / ZU) über BUS bedienen	
		-1	Aktivieren u. automatisch deaktivieren (Impuls)
		0	Keine Funktion
Command.SetBS	B	Bedientaster (BS / STOPP) über BUS aktivieren / deaktivieren	
		-1	Aktivieren (Festhalten)
		0	Deaktivieren

			(Loslassen)
Command.ImpBS	B	Bedientaster (BS / STOPP) über BUS bedienen	
		-1	Aktivieren u. automatisch deaktivieren (Impuls)
		0	Keine Funktion
Status.Button.BT	B	Bedientaster / Eingang (BT)	
		-1	Der Eingang BT-BTA3-BTZ1B ist als BT konfiguriert und BT ist betätigt.
		0	Nicht betätigt
Status.Button.BTA1	B	Bedientaster / Eingang (BTA1 / AUF)	
		-1	BTA1 ist betätigt
		0	Nicht betätigt
Status.Button.BTA2	B	Bedientaster / Eingang (BTA2 / AUF)	
		-1	BTA2 ist betätigt
		0	Nicht betätigt
Status.Button.BTA3	B	Bedientaster / Eingang (BTA3 / AUF)	
		-1	Der Eingang BT-BTA3-BTZ1B ist als BTA3 konfiguriert und BTA3 ist betätigt.
		0	Nicht betätigt
Status.Button.BTZ1A	B	Bedientaster / Eingang (BTZ1A / ZU)	
		-1	BTZ1A ist betätigt
		0	Nicht betätigt
Status.Button.BTZ1B	B	Bedientaster / Eingang (BTZ1B / ZU)	
		-1	Der Eingang BT-BTA3-BTZ1B ist als BTZ1B konfiguriert und BTZ1B ist betätigt.
		0	Nicht betätigt
Status.Button.BTZ2	B	Bedientaster / Eingang (BTZ2 / ZU)	
		-1	BTZ2 ist betätigt
		0	Nicht betätigt

Status.Button.BTS1	B	Nur MO 64 Plus: Bedientaster / Eingang (BTS1 / STOPP)	
		-1	BTS1 ist betätigt
		0	Nicht betätigt
Status.Button.BTS2	B	Nur MO 64 Plus: Bedientaster / Eingang (BTS2 / STOPP)	
		-1	BTS2 ist betätigt
		0	Nicht betätigt
Status.Button.BTS	B	Nur MO 24: Bedientaster / Eingang (BTS / STOPP)	
		-1	BTS ist betätigt
		0	Nicht betätigt
Status.BusCommand.BT	B	BUS-Befehl (BT)	
		-1	Der Bedienbefehl BUS-BT ist aktiviert.
		0	Nicht aktiviert
Status.BusCommand.BA	B	BUS-Befehl (BA / AUF)	
		-1	Der Bedienbefehl BUS-BA ist aktiviert.
		0	Nicht aktiviert
Status.BusCommand.BZ	B	BUS-Befehl (BZ / ZU)	
		-1	Der Bedienbefehl BUS-BZ ist aktiviert.
		0	Nicht aktiviert
Status.BusCommand.BS	B	BUS-Befehl (BS / STOPP)	
		-1	Der Bedienbefehl BUS-BS ist aktiviert.
		0	Nicht aktiviert

Tabelle 15

9.9 **Wartung**

Über die folgenden Variablen kann das Serviceintervall eingestellt bzw. der Servicezähler abgefragt werden. Außerdem lassen sich der Fahrzeugzähler und der remanente (nicht rücksetzbare) Servicezähler auslesen.

Variable	Typ	Beschreibung
Config.ServiceInterval	N	Wartungsintervall
		Die Wartungsmeldung wird erzeugt,

		wenn der Wartungscounter größer als das Wartungsintervall ist. Die Wartungsmeldung kann über ein Multirelais signalisiert werden. Der Wartungszähler kann gelöscht werden. Minimaler Wert: 1.000 Maximaler Wert: 9.999.000 Es sind Werte zulässig die sich ganzzahlig durch 1.000 teilen lassen. Ansonsten wird abgerundet. Beispiel: Aus 1.234.567 wird 1.234.000.	
Command.ClearServiceCounter	B	Wartungszähler zurücksetzen	
		-1	Der Wartungszähler wird auf 0 gesetzt.
		0	Keine Funktion
Status.Counter.ServiceTotalizer	N	Servicecounter	
		Der Servicecounter zählt jede Schrankenbewegung. Gezählt wird immer nach dem Abschalten des Motors. Der Zähler kann nicht gelöscht werden.	
Status.Counter.Service	N	Wartungscounter	
		Der Wartungscounter zählt jede Schrankenbewegung. Gezählt wird immer nach dem Abschalten des Motors. Der Zähler kann gelöscht werden.	

Tabelle 16

9.10 Fehlermeldungen

Bei jeder Veränderung der Fehlerflaggen werden diese zusammen mit dem aktuellen Wert des Betriebsstundenzählers gespeichert. Dabei wird der aktuelle Wert in den Fehlerspeicher 0 geschrieben. Der „alte“ Fehlerspeicher 0 wandert in Fehlerspeicher 1 usw. Der Fehlerspeicher-Eintrag steht in der Variable „Alarms.Msg_{0..9}.Flags“ und der dazugehörige Zeitstempel in der Variable „Alarms.Msg_{0..9}.Timestamp“. Dabei ist im Variablennamen das „{0..9}“ durch die gewünschte Eintrags-Nummer (zwischen 0 und 9) zu ersetzen.

Variable	Typ	Beschreibung	
Command.ClearAlarmLog	B	Fehlerspeicher löschen	
		-1	Löschen des Fehlerspeichers
		0	Keine Funktion
Alarms.Msg_{0..9}.Flags	S	Fehlerspeicher 0..9	

	<p>Der Fehlerspeicher wird als String übermittelt. Je Meldung ein Bit („0“ oder „1“). Die „0“ steht für nicht gemeldet und die „1“ für gemeldet. Hier die Meldebits:</p>
1	Baum-Ab
2	SEA defekt
3	SEZ defekt
4	SEA u. SEZ gleichzeitig betätigt
5	12V Versorgung defekt
6	24V Versorgung defekt
7	Schleife A defekt
8	Schleife B defekt
9	Schleife C defekt
10	Lichtschrankentest-Fehler
11	SLZ-Test-Fehler
12	UC Lern-Fehler
13	Relais defekt
14	Triac defekt
15	Registertest-Fehler
16	Stoppredundanzfehler
17	EEPROM defekt
18	EEPROM Checksummenfehler
19	Nicht genutzt - immer "0"
20	Nicht genutzt - immer "0"
21	Nicht genutzt - immer "0"
22	Nicht genutzt - immer "0"
23	Nicht genutzt - immer "0"
24	Nicht genutzt - immer "0"
25	Nicht genutzt - immer "0"

		26	Nicht genutzt - immer "0"
		27	Nicht genutzt - immer "0"
		28	Nicht genutzt - immer "0"
		29	Nicht genutzt - immer "0"
		30	Nicht genutzt - immer "0"
		31	Nicht genutzt - immer "0"
Alarms.Msg_{0..9}.Timestamp	N	Zeitstempel des Fehlerspeicher-Eintrags Der Zeitstempel hat das gleiche Format wie der Betriebsstundenzähler.	

Tabelle 17

9.11 Schranken-Modell

Diese Konfigurationsparameter sind nur in der MO 24 vorhanden.

Config.Barrier.Series	N	Schranken-Modell	
		1	T 3000
		2	T 3500
		3	P 2500
		4	P 3000
		5	P 3500
		6	P 4000
		7	P 5000
Config.Barrier.Speed	N	Geschwindigkeit	
		0	Schnell
		1	Standard
		2	Langsam

Tabelle 18

9.12 Weiter Variablen

Variable	Typ	Beschreibung
Config.CommandCounter.LowerLimit	N	Befehlszähler (Obere Grenze) Der Befehlszähler in der Steuerung wird durch Auf-Befehle inkrementiert und durch Zu-Befehle dekrementiert. Die

		Schranke öffnet, wenn der Zähler von 0 auf 1 springt und schließt, wenn der Zähler von 1 auf 0 springt.
		-9 ... 0
		Minimaler Wert Maximaler Wert
Config.CommandCounter.UpperLimit	N	Befehlszähler (Untere Grenze)
		1 ... 9
		Minimaler Wert Maximaler Wert
Status.Counter.Commands	N	Nur MO24: Aktueller Wert des Befehlszählers
Config.Password	N	Passwort Das eingestellte Passwort für das Lernmenü der MO 64 "Plus".
		3552
		Passwort ungleich 3552 = Es kann keine Konfiguration an der Steuerung vorgenommen werden.
Command.TransferToUser	B	Kundenspeicher
		-1
		Überträgt die aktuelle Konfiguration in den Kundenspeicher
		0
		Keine Funktion
Status.Hardware.HardwareVersion	N	Hardwareversion
		5
		MO 64 V4
Status.Hardware.SoftwareVersion	N	Softwareversion
		5
		MO 64 2V2
		6
		MO 64 2V3
Status.Hardware.SerialNumber	N	Seriennummer
		Jede Steuerung hat eine einzigartige Seriennummer, die bei der Produktion vergeben wird.
Status.Hardware.MacAdresse	S	MAC-Adresse
		Jedes Netzwerkgerät muss zur Verwendung des TCP/IP-Protokolls eine einzigartige MAC-Adresse zugewiesen bekommen. Die MAC-Adresse ist in der Steuerung abgelegt

		und wird für das TCP-Modul verwendet. Das TCP-Modul selbst hat keine eigene MAC Adresse.	
Status.Hardware.WorkingHours	N	Betriebsstundenzähler	
		Dies ist der Betriebsstundenzähler der Steuerung der alle 10 Minuten inkrementiert wird.	
Config.AutoSync	B	Nur MO24: Automatischer Zulauf nach Netzeinschalten	
		-1	Aktiv
		0	Nicht aktiv
Status.ServiceMode	B	Nur MO24: Serviceschalter	
		-1	Betätigt
		0	Nicht betätigt
Status.PowerMonitor	B	Nur MO24: Netzüberwachung	
		-1	Betätigt
		0	Nicht betätigt
Config.OpenLimit	N	Nur MO24: Öffnungsbegrenzung	
		0	Vollöffnung 90°
		1	Teilöffnung 85°
		2	Teilöffnung 80°
		3	Teilöffnung 75°
		4	Teilöffnung 70°
		5	Teilöffnung 65°
		6	Teilöffnung 60°
Config.AlmostCloseFunction	N	Nur MO24: Ausblendung	
		0	Keine Ausblendung
		1	Ausblendung 10°
		2	Ausblendung 20°
		3	Ausblendung 30°
Config.Power	B	Nur MO24: Stromversorgung	
		-1	Netzausfallerkennung
		0	Normal + Akku

Tabelle19

10 Kommunikationsdienst einbinden

In diesem Kapitel werden die technischen Details der Software beschrieben um das Programm in eine eigene Software zu integrieren.

10.1 Benötigte Dateien

Installieren Sie das SDK auf einem Rechner. Es werden alle Dateien und Verzeichnisse aus dem Verzeichnis: "C:\Program Files (x86)\ELKA\SDK" bis auf das Unterverzeichnis „Development“ benötigt. Insbesondere werden folgende Dateien bzw. Verzeichnisse benötigt:

Datei / Verzeichnis	Beschreibung
Log	Protokoll-Verzeichnis. Es muss jeder Benutzer des Rechners Schreibrechte in diesem Verzeichnis erhalten.
_Register.bat	Kommunikationsdienst in Windows registrieren. Dieses Script muss als Administrator ausgeführt werden. Es wird ein Windows-Dienst mit dem Namen: „ELKA Communication Service“ und der ID: ElkaMo64 erstellt.
_ServiceStart.bat	Den Windows-Dienst starten. Dieses Script muss als Administrator ausgeführt werden.
_ServiceStop.bat	Den Windows-Dienst beenden. Dieses Script muss als Administrator ausgeführt werden.
_UnRegister.bat	Entfernen der Registrierung des Windows-Dienstes. Dieses Script muss als Administrator ausgeführt werden.
ELKAMO64.dll	DLL zur Kommunikation von Fremdanwendungen mit dem Windows-Dienst.
ELKAService.exe	Programmdatei des Windows-Dienst.
ELKAServiceInstaller.exe	Installationsprogramm für den Windows-Dienst. Dieses Programm wird von den Script-Dateien: _Register.bat und _UnRegister.bat verwendet. Dieses Script muss als Administrator ausgeführt werden. Mögliche Parameter: /INSTALL Dienst installieren so dass dieser manuell gestartet werden muss. /INSTALL_AUTO Dienst installieren so dass dieser

Datei / Verzeichnis	Beschreibung
	<p>beim Start von Windows automatisch startet.</p> <p>/UNINSTALL Dienst deinstallieren</p> <p>/KILLTRAY Alle laufenden Instanzen derELKATray.exe-Anwendung beenden</p>
ELKATray.exe	<p>Programm zur Konfiguration des Kommunikations-Dienstes. Mögliche Parameter:</p> <p>/VISIBLE Das Programm beim Start nicht gleich minimieren. Nur Administratoren haben die Möglichkeit über dieses Programm den Dienst zu starten oder zu beenden. Die Konfiguration kann von jedem Benutzer vorgenommen werden.</p>
ELKATrayServiceHandler.exe	<p>Programm zum Steuern des Dienstes. Dieses Programm wird vom Programm ELKATray.exe und den Scripten _ServiceStart.bat und _ServiceStop.bat verwendet. Mögliche Parameter:</p> <p>/START Den Dienst starten</p> <p>/STOP Den Dienst beenden</p> <p>/ENABLE Den automatischen Start des Dienstes aktivieren</p> <p>/DISABLE Den automatischen Start des Dienstes deaktivieren</p>
libeay32.dll	OpenSSL Library wird von dem Dienst und der ELKAMO64.dll verwendet.
SDK.cert.pem	Das Zertifikat wird nur benötigt, wenn über HTTPS kommuniziert werden soll.
SDK.key.pem	Der Zertifikats-Schlüssel wird nur benötigt, wenn über HTTPS kommuniziert werden soll.

Datei / Verzeichnis	Beschreibung
ssleay32.dll	OpenSSL Library wird von dem Dienst und der ELKAMO64.dll verwendet.

Tabelle 20

Des Weiteren wird die Konfiguration benötigt. Diese wird im folgenden Pfad erwartet:

„C:\Program Files (x86)\Common Files\ELKA SDK“.

Derzeit wird hierfolgende Datei erwartet:

Datei / Verzeichnis	Beschreibung
ELKAService.INI	Konfigurationsdatei. Siehe Abschnitt: „Konfiguration“ weiter unten.

10.2 Konfigurationsdatei

Die Konfiguration wird im INI-Dateiformat in der Datei „C:\Program Files(x86)\Common Files\ELKA SDK\ELKAService.INI“ gespeichert. Folgende allgemeinen-Parameter können hier eingestellt werden:

Sektion	Parameter	Beschreibung
Interface	TCP.Port	Vorgabe: 17793 Der TCP-Port über den die DLL mit dem Dienst kommuniziert (0 = deaktiviert)
Interface	HTTP.Port	Vorgabe: 17794 Der TCP-Port über den mit dem Dienst via HTTP-Protokoll kommuniziert werden kann.
Interface	HTTPS.Port	Vorgabe: 17795 Der TCP-Port über den mit dem Dienst via HTTPS-Protokoll kommuniziert werden kann.
Interface	HTTPS.CertDir	Vorgabe: Auskommentiert Pfad zum HTTPS-Zertifikat (wird im Regelfall nicht benötigt)
Interface	HTTPS.CertPass	Vorgabe: Auskommentiert Passwort der Zertifikatsdatei (wird im Regelfall nicht benötigt)
Debug	Enabled	Vorgabe: 0

Sektion	Parameter	Beschreibung
		Erweiterte Protokollierung in der Log-Datei zur Fehlersuche Aktiviert: 1 Deaktiviert: 0
RS485	RetryCount	Vorgabe: 3 Anzahl der Versuche bis ein Telegramm verworfen wird.
RS485	AutoToken	Vorgabe: 200 Einheit: Millisekunden Diese Werte werden vom LogiTed-Protokoll vorgegeben und sollten nicht verändert werden
RS485	TimeoutRX	Vorgabe 150 Einheit: Millisekunden Diese Werte werden vom LogiTed-Protokoll vorgegeben und sollten nicht verändert werden
RS485	TimeoutSYN	Vorgabe: 200 Einheit: Millisekunden Dieser Wert wird vorgegeben und sollte nicht verändert werden
RS485	IdleGapRxTx	Vorgabe: 10 Einheit: Millisekunden Dieser Wert wird vorgegeben und sollte nicht verändert werden
RS485	ForceDisconnectedRX	Vorgabe: 3000 Einheit: Millisekunden Dieser Wert wird vorgegeben und sollte nicht verändert werden
TCP	TimeoutRX	Vorgabe: 500 Einheit: Millisekunden Dieser Wert wird vorgegeben und sollte nicht verändert werden
TCP	TimeoutSYN	Vorgabe: 200 Einheit: Millisekunden Dieser Wert wird vorgegeben und sollte nicht verändert werden
TCP	ReconnectWait	Vorgabe: 2000 Einheit: Millisekunden

Sektion	Parameter	Beschreibung
		Zeit zwischen den Verbindungsversuchen
TCP	RetryCount	Vorgabe: 3 Anzahl der Versuche bis ein Telegramm verworfen wird.
TCP	ForceDisconnectedRX	Vorgabe: 3000 Einheit: Millisekunden Vorgabezeit nach der ohne Kommunikation eine Verbindung als abgebaut erachtet wird.
Try	HWND	Dieser Wert wird vom Programm ELKATray.exe geschrieben und beschreibt dessen Windows-Form-Handle (HWND)
Devices	Count	Anzahl der konfigurierten Schranken/Geräte

Tabelle 21

Zum Parametrieren der Kommunikationsteilnehmer die Konfiguration wie folgt anpassen:

Je Teilnehmer muss in der Sektion „Devices“ der Zähler „Count“ hochgezählt werden. Der Erste Teilnehmer erhält hierbei die Nummer „1“ alle weiteren werden fortlaufend nummeriert. Je Teilnehmer wird wie folgt eine Sektion erstellt: „Device.{1..n}“ wobei {1..n} durch den Zähler ersetzt wird.

Beispiel Schranke über RS485:

Sektion	Parameter	Beschreibung
Device.{Nummer}	Mode	Wert: RS485
Device.{Nummer}	Addr	Schrankenadresse wie in der Schranke konfiguriert.
Device.{Nummer}	LocalAddr	Eigene BUS-Adresse
Device.{Nummer}	Type	Wert: MO64 Achtung: auch bei MO 24 verwenden
Device.{Nummer}	COMPort	Lokaler COM-Port über den kommuniziert werden soll. Beispiel: COM1

Tabelle 22

Beispiel - weiterer (zusätzlicher) PC über RS485:

Sektion	Parameter	Beschreibung
Device.{Nummer}	Mode	Wert: RS485
Device.{Nummer}	Addr	Schrankenadresse wie in der Schranke konfiguriert.
Device.{Nummer}	LocalAddr	Entfernte BUS-Adresse.
Device.{Nummer}	Type	Wert: PC
Device.{Nummer}	COMPort	Lokaler COM-Port über den kommuniziert werden soll. Beispiel: COM1

Tabelle 23

Beispiel Schranke über TCP:

Sektion	Parameter	Beschreibung
Device.{Nummer}	Mode	Wert: TCP
Device.{Nummer}	Host	IP der Schranke Beispiel: 192.168.0.100
Device.{Nummer}	Port	Port der Schranke Vorgabe: 52719

Tabelle 24

10.3 Konfiguration und Initialisierung über DLL

Die Konfiguration muss nicht zwangsläufig über die DLL konfiguriert werden. Im laufenden Betrieb kann dieser über die folgende Funktion angepasst werden:

```
function SetConfig(  
  AHost: PWideChar;           // IP des PCs auf dem der Dienst läuft  
  APort: Word;                // Port-Nummer des Dienstes  
  AConfig: PWideChar          // Die Konfiguration [Devices] und  
                               // [Device.{1..n}]  
): Boolean; stdcall;          // TRUE = Befehl erfolgreich
```

Wenn die INI-Datei extern modifiziert wurde, z.B. über ein eigenes Konfigurationswerkzeug, dann kann der Dienst über folgenden DLL-Aufruf zum neuen Einlesen der Konfigurations-INI-Datei aufgefordert werden.

```
function Relnit(  
  AHost: PWideChar;           // IP des PCs auf dem der Dienst läuft  
  APort: Word                 // Port-Nummer des Dienstes  
): Boolean; stdcall;          // TRUE = Befehl erfolgreich
```