

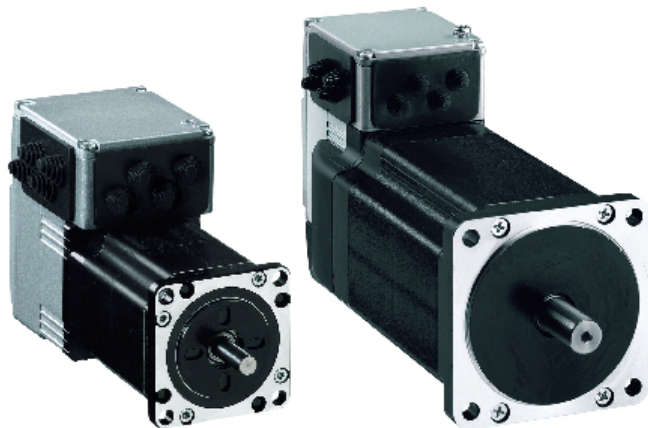
# ILx2K EtherNet/IP

Lexium Integrierter Antrieb

Feldbushandbuch PLCopen

Add-On Instructions für  
Rockwell RS Logix5000

V0.08, 11.2009



---

**BERGER LAHR**

[www.schneider-electric.com](http://www.schneider-electric.com)

**Schneider**  
Electric



## **Wichtige Hinweise**

Dieses Handbuch ist Teil der Add-On Instruction nach PLCopen für Rockwell RSLogix 5000.

Lesen und befolgen Sie dieses Handbuch.

Bewahren Sie dieses Handbuch auf.

Nicht alle Produkte sind in allen Ländern erhältlich.

Die Verfügbarkeit der Produkte entnehmen Sie bitte dem aktuellen Katalog.

Wir behalten uns das Recht vor, ohne Ankündigung technische Änderungen vorzunehmen.

Alle Angaben sind technische Daten und keine zugesicherten Eigenschaften.

Die meisten Produktbezeichnungen sind auch ohne besondere Kennzeichnung als Warenzeichen der jeweiligen Inhaber zu betrachten.



## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>4</b>
1.1	Lieferumfang	4
<b>2</b>	<b>Erstellen einer Applikation</b>	<b>5</b>
2.1	Export / Import der Add-On Instructions und Datentypen	5
2.2	Globale Datenstruktur pro Antrieb erstellen	7
2.3	I/O Konfiguration	7
2.4	I/O Konfiguration Laden und Testen	10
2.5	Adressierung der Antriebe	12
2.5.1	Drehschalter zur Einstellung der IP Adresse	12
2.5.2	Adresseinstellung mittels integriertem Web-Server	12
2.5.3	Adresseinstellung mittels Lexium CT	14
<b>3</b>	<b>Add-On Instructions</b>	<b>17</b>
3.1	Bausteinbezeichnung	17
3.2	Signaldiagramme für Ein- und Ausgänge	17
3.2.1	Steuereingänge	17
3.2.2	Steuerausgänge	18
3.2.3	Signaldiagramme für Bausteine mit Execute Steuereingang	19
3.2.4	Signaldiagramm für Bausteine mit Enable Steuereingang	20
3.3	Die Beschreibung der Bausteine	21
3.4	Verwendungszweck der Bausteine	23
3.4.1	Initialisierung	23
3.4.2	Manuellfahrt	23
3.4.3	Referenzierung	23
3.4.4	Betriebsart Punkt-zu-Punkt	24
3.4.5	Betriebsart Geschwindigkeitsprofil	24
3.4.6	Stoppen	25
3.4.7	Schnelle Positionserfassung	25
3.4.8	Parameter lesen	25
3.4.9	Parameter schreiben	26
3.4.10	Ein-/Ausgänge	26
3.4.11	Fehlerbehandlung	26
3.5	Detaillierte Bausteinbeschreibung	27
3.5.1	MC_UpdateInputData_ETH_ILx2	27
3.5.2	MC_UpdateOutputData_ETH_ILx2	28
3.5.3	MC_Power_ETH_ILx2	29
3.5.4	MC_Jog_ETH_ILx2	31
3.5.5	MC_Home_ETH_ILx2	33
3.5.6	MC_SetPosition_ETH_ILx2	36
3.5.7	MC_MoveAbsolute_ETH_ILx2	38
3.5.8	MC_MoveRelative_ETH_ILx2	40
3.5.9	MC_MoveAdditive_ETH_ILx2	42
3.5.10	MC_MoveVelocity_ETH_ILx2	44
3.5.11	MC_VelocityControl_ETH_ILx2	46
3.5.12	MC_Stop_ETH_ILx2	48
3.5.13	MC_TouchProbe_ETH_ILx2	50

---

## PLCopen AddOnInstructions ILx2 EthernetIP

3.5.14	MC_AbortTrigger_ETH_ILx2.....	53
3.5.15	MC_ReadActualPosition_ETH_ILx2 .....	54
3.5.16	MC_ReadActualVelocity_ETH_ILx2 .....	56
3.5.17	MC_ReadStatus_ETH_ILx2.....	57
3.5.18	MC_ReadParameter_ETH_ILx2 .....	60
3.5.19	MC_WriteParameter_ETH_ILx2 .....	62
3.5.20	MC_SetLimitSwitch_ETH_ILx2 .....	64
3.5.21	MC_ReadDigitalInputs_ETH_ILx2 .....	66
3.5.22	MC_Reset_ETH_ILx2 .....	67
3.5.23	MC_ReadAxisError_ETH_ILx2 .....	69
<b>4</b>	<b>Kommunikation überwachen .....</b>	<b>71</b>
4.1	I/O LED Status auslesen .....	71
4.2	Einzelne I/O Verbindungen überwachen .....	72
<b>5</b>	<b>Anhang .....</b>	<b>73</b>
5.1	Fehlernummern .....	73

## **Informationen zu dieser Ausgabe**

### **Version 0.0/1**

Handbuch erstellt.

### **Version 08**

Input Assembly von 44 bytes auf 40 bytes geändert-  
Baustein Version ab V0.201

## 1 Einführung

Mit den Bewegungsbausteinen nach PLCopen Motion Spezifikation werden Programme nach IEC 61131-3 Standard in der Programmierumgebung RSLogix 5000 der Firma Rockwell geschrieben. Die Bausteine steuern dabei einen oder mehrere Antriebe vom Typ Lexium ILx2 von Schneider Electric über den Feldbus EtherNet/IP. Die Bewegungsbausteine sind universell gehalten, die Bibliothek ist somit geeignet für die folgenden Antriebe:

- ILS2K...
- ILE2K...
- ILA2K...

### 1.1 Lieferumfang

Kontrollieren Sie den Lieferumfang auf Vollständigkeit, um eine einfache und vollständige Integration in das Programmiersystem RSLogix 5000 von Rockwell zu ermöglichen.

Die Bibliotheken tragen die Versionsnummer im Bibliotheksnamen, Vxxxx wird also durch die entsprechende Versionsnummer ersetzt, z.B. V1001.

- RSLogix Projekt mit allen verfügbaren Add-On Instructions und einem Beispielprogramm

SE\_Motion\_ILx2\_ETH\_Vxxxx.ACD

- Die Gerätebeschreibung SEILx2K\_0011E.EDS.eds
- Dieses Handbuch



## 2 Erstellen einer Applikation

Die Add-On Instructions wurden für die RSLogix 5000 Programmiersoftware erstellt. Die nachfolgenden Schritte sind notwendig, um die Lexium ILx2 Antriebe in der RSLogix 5000 Software bekannt zu machen und die Prozessdaten der Antriebe mit den Datenstrukturen der Bibliothek zu verknüpfen. Führen Sie die Schritte in der genannten Reihenfolge aus.

Das Projekt „SE\_Motion\_ILx2\_ETH\_Vxxxx.ACD“ kann als Basis für eine neue Applikation verwendet werden.

Wenn die Add-On Instructions in ein bestehendes RSLogix 5000 - Projekt eingebunden werden sollen, können diese aus dem Projekt „SE\_Motion\_ILx2\_ETH\_Vxxxx.ACD“ exportiert und in das bestehende Projekt importiert werden.

### 2.1 Export / Import der Add-On Instructions und Datentypen

Die Software RSLogix 5000 bietet keine Möglichkeit, mehrere Add-On Instructions und Datentypen gleichzeitig zu exportieren bzw. importieren. Deshalb muss jeder Baustein und jeder Datentyp einzeln exportiert und importiert werden.

- Öffnen Sie das Projekt „SE\_Motion\_ILx2\_ETH\_Vxxxx.ACD“ mit der Software RSLogix 5000.

- Exportieren Sie den Datentyp:

AXIS\_REF\_EIP\_ILx2



Bild 1) Notwendiger Datentyp

- Rechts-Klick auf den Datentyp

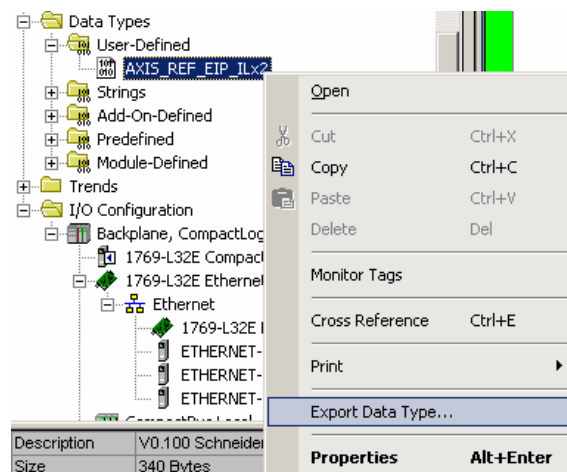


Bild 2) Menüpunkt „Export Data Type...“ aus dem Kontextmenü

- Speichern Sie die \*.L5X Dateien in einen bekannten Ordner.

- Exportieren Sie die Add-On Instructions.

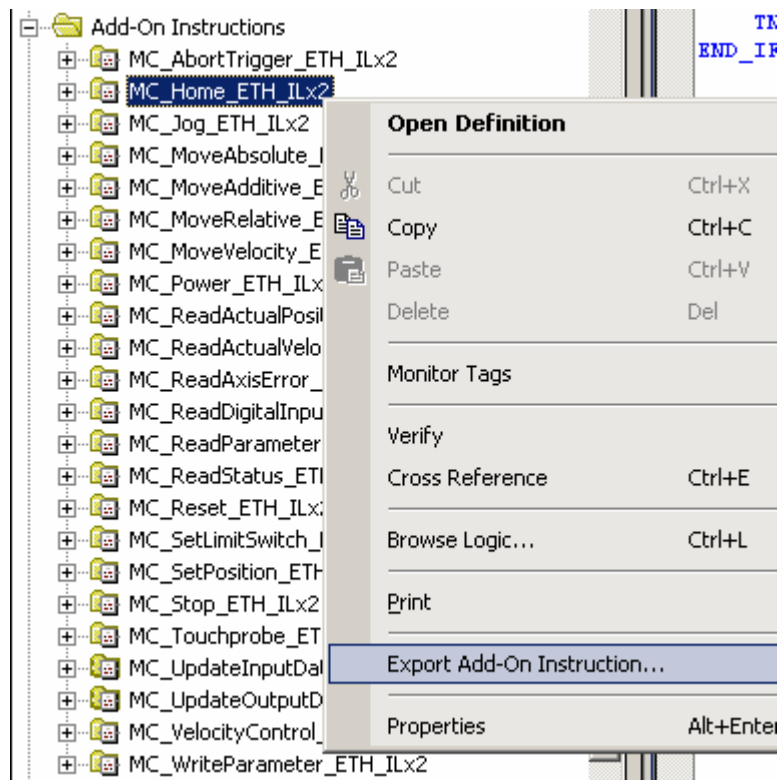


Bild 3) Menüpunkt „Export Add-On-Instruction...“ aus dem Kontextmenü

- Speichern Sie die \*.L5X Dateien in einen bekannten Ordner. Um Speicherplatz zu sparen, brauchen nicht alle verfügbaren Add-On Instructions exportiert zu werden. Welche Add-On Instructions in der Applikation benötigt werden, hängt von den benötigten Funktionen und Betriebsarten ab. Wird zum Beispiel keine „Schnelle Positionserfassung über Capture Eingang“ benötigt, brauchen die Add-On Instructions „MC\_AbortTrigger\_ETH\_ILx2“ und „MC\_Touchprobe\_ETH\_ILx2“ nicht exportiert / importiert zu werden.
- Importieren Sie nun zuerst die Datentypen und dann die Add-On Instructions aus den \*.L5X Dateien.

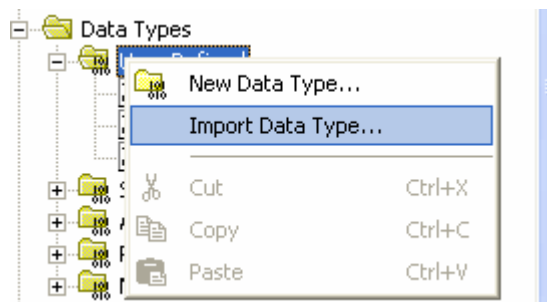


Bild 4) Menüpunkt „Import Data Type...“ aus dem Kontextmenü

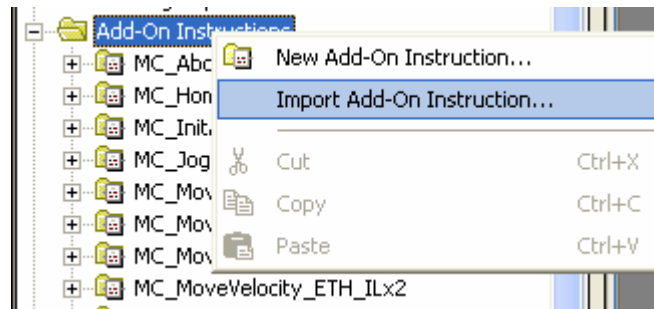


Bild 5) Menüpunkt „Import Add-On Instruction...“ aus dem Kontextmenü

## 2.2 Globale Datenstruktur pro Antrieb erstellen

- Legen Sie nun für jeden Lexium ILx2 Antrieb, den Sie am EtherNet/IP betreiben wollen, eine Datenstruktur in den „Controller Tags“ oder „Program Tags“ an.  
Die Datenstruktur muss vom Typ AXIS\_REF\_EIP\_ILx2 sein.

	Name	Alias For	Base Tag	Data Type	Style	Description
	+AxisRef_1			AXIS_REF_EIP_ILx2		V0.100 Schneider Electric Motion
	+AxisRef_2			AXIS_REF_EIP_ILx2		V0.100 Schneider Electric Motion
	+AxisRef_3			AXIS_REF_EIP_ILx2		V0.100 Schneider Electric Motion

Bild 6) Datenstruktur für jeden Antrieb (Beispiel 3 Antriebe)

## 2.3 I/O Konfiguration

In der I/O Konfiguration werden die Ein- und Ausgabegeräte verwaltet. Hier müssen die Lexium ILx2 Antriebe mit EtherNet/IP Schnittstelle hinzugefügt werden.

- Fügen Sie die Lexium ILx2 Antriebe der I/O Konfiguration hinzu. Markieren Sie dazu den EtherNet/IP Master und wählen Sie über die rechte Maustaste „New Module...“ aus.

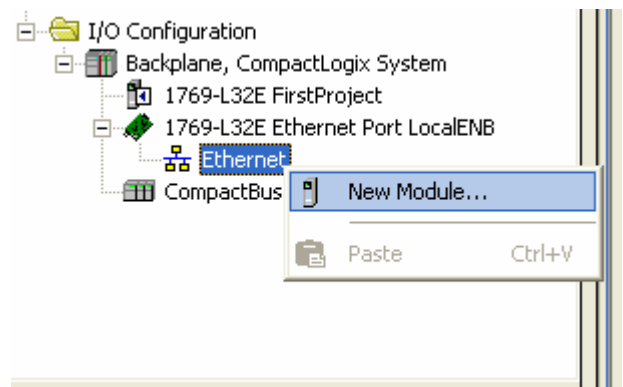


Bild 7) New Module...

## Erstellen einer Applikation PLCopen AddOnInstructions ILx2 Ethernet/IP

- Wählen Sie in dem Auswahlfenster in der Gruppe „Communications“ den Eintrag „ETHERNET-MODULE – Generic Ethernet Module“ aus. Bestätigen Sie dann mit OK.

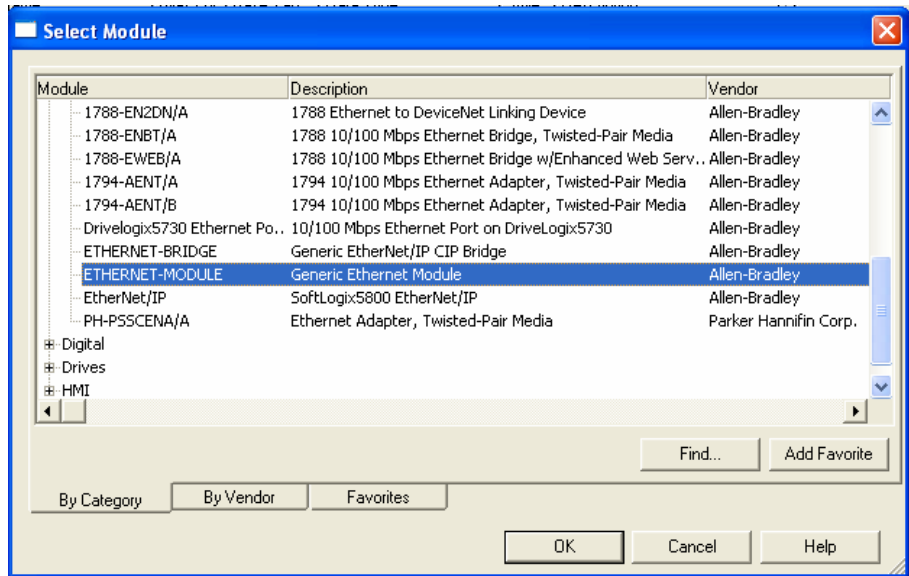


Bild 8) Einfügen eines EtherNet/IP Antriebes

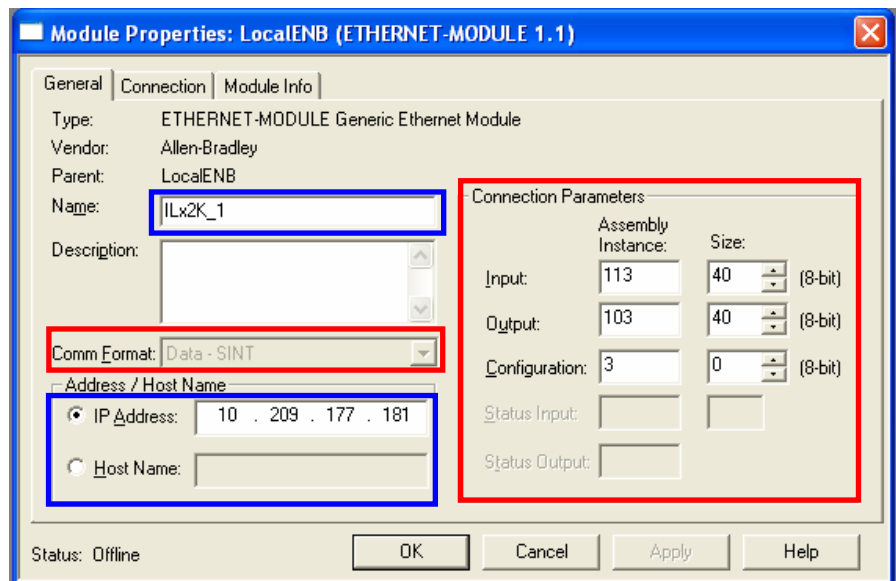
- Nach dem Bestätigen mit OK öffnet sich das Fenster „Module Properties“. Der Name des Antriebes kann frei gewählt werden. Geben Sie die Connection Parameter wie folgt an.

Input Assembly Instance = 113 Size 40 (8-Bit - SINT)  
Output Assembly Instance = 103 Size 40 (8-Bit - SINT)  
Configuration Assembly Instance = 3 Size 0 (8-Bit)

Comm\_Format = Data-SINT

Tragen Sie die IP Adresse oder den Host-Namen des Antriebes ein.

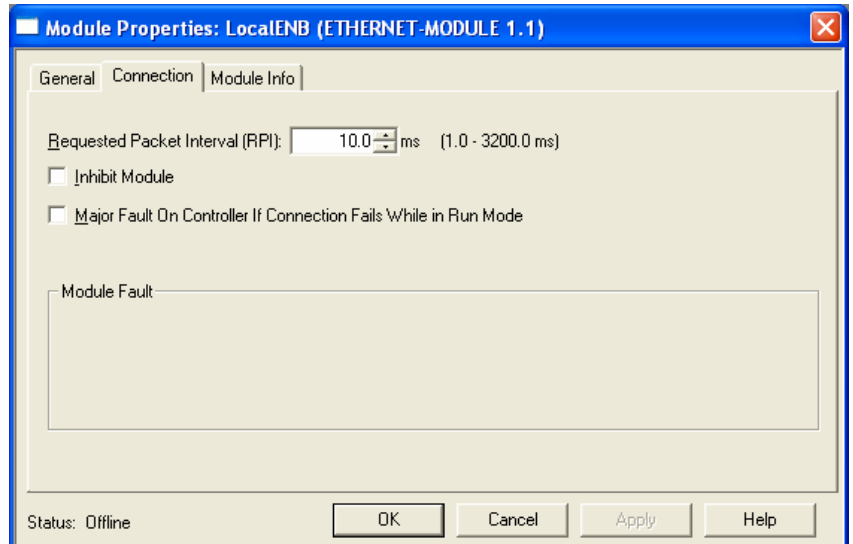
Wie Sie den Antrieb adressieren, entnehmen Sie bitte dem Produkthandbuch.



## PLCopen AddOnInstructions ILx2 EthernetIP

Bild 9) EtherNet/IP Module Properties / General

- Wählen Sie nun in der Registerkarte „Connection“ den gewünschten Datenübertragungszyklus (Requested Packet Interval).  
Defaulteinstellung ist 10ms. Beachten Sie, dass dieser Wert entsprechend der Anzahl der EtherNet/IP Teilnehmer nicht zu klein eingestellt werden darf.



- Wiederholen Sie die vorangegangenen Schritte, bis Sie alle Antriebe zum EtherNet/IP Netzwerk hinzugefügt haben.
- In den Controller Tags finden Sie nun die Prozessdaten des Antriebes.

Controller Tags - Compact(controller)

Scope: Compact Show... Show All

Name	Alias For	Base Tag	Data Type
+ ILx2K_0:C			AB:ETHERNET_MODULE:C:0
+ ILx2K_0:I			AB:ETHERNET_MODULE_SINT_40Bytes:I:0
+ ILx2K_0:O			AB:ETHERNET_MODULE_SINT_40Bytes:O:0

Bild 10) Antrieb mit seinen Prozessdaten

## 2.4 I/O Konfiguration Laden und Testen

### @ GEFAHR

#### Unbeabsichtigte Folgen des Betriebes

Beim Start der Anlage sind die angeschlossenen Antriebe in der Regel außer Sichtweite des Anwenders und können nicht unmittelbar überwacht werden.

- Starten Sie die Anlage nur, wenn sich keine Personen oder Hindernisse im Gefahrenbereich befinden.

**Nichtbeachtung dieser Vorkehrungen führt zu Tod oder schweren Verletzungen.**

### @ WARNUNG

#### Unerwartete Bewegung

Antriebe können durch falsche Verdrahtung, falsche Einstellungen, falsche Daten oder andere Fehler unerwartete Bewegungen ausführen.

Störungen (EMV) können in der Anlage unvorhergesehene Reaktionen hervorrufen.

- Führen Sie die Verdrahtung gemäß den EMV-Maßnahmen sorgfältig durch.
- Schalten Sie die Spannung an den Eingängen `STO_A` (`PWRR_A`) und `STO_B` (`PWRR_B`) ab, um einen unerwarteten Anlauf des Motors zu vermeiden, bevor Sie das Antriebssystem einschalten und konfigurieren.
- Betreiben Sie das Antriebssystem NICHT mit unbekanntem Einstellungen oder Daten.
- Führen Sie eine sorgfältige Inbetriebnahmeprüfung durch.

**Nichtbeachtung dieser Vorkehrungen kann zu Tod oder schwerwiegenden Verletzungen führen.**

## PLCopen AddOnInstructions ILx2 EthernetIP

Die folgenden Schritte übertragen das Projekt in die Steuerung und starten diese.

- Laden Sie das Projekt in die Steuerung und starten diese mit „RUN“.
- Die Status-LED I/O zeigt, ob die I/O-Konfiguration ordnungsgemäß läuft. -> Ist dies der Fall, leuchtet die LED dauerhaft.

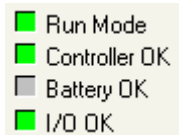


Bild 11) Zustands LEDs der Rockwell Steuerung (I/O OK)

- Ist eine oder sind mehrere Baugruppen / Feldbus-Teilnehmer nicht vorhanden oder ein Fehler liegt vor, so blinkt die LED.

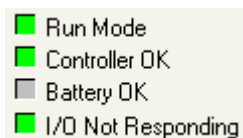


Bild 12) Zustands LEDs der Rockwell Steuerung (I/O Not Responding)

- Der Kommunikationsstatus eines einzelnen Antriebes kann in der Software RSLogix 5000 im Online Modus beobachtet werden.

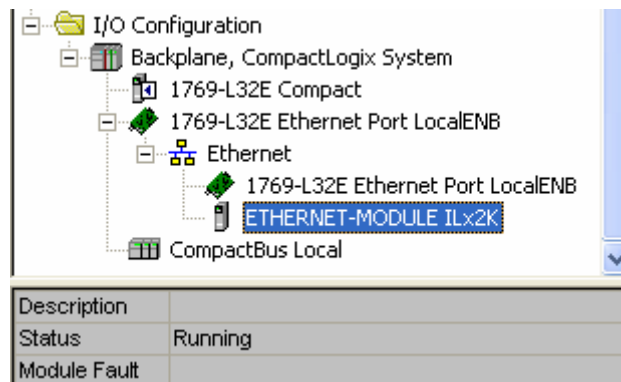


Bild 13) ILx2 Kommunikationsstatus - Running

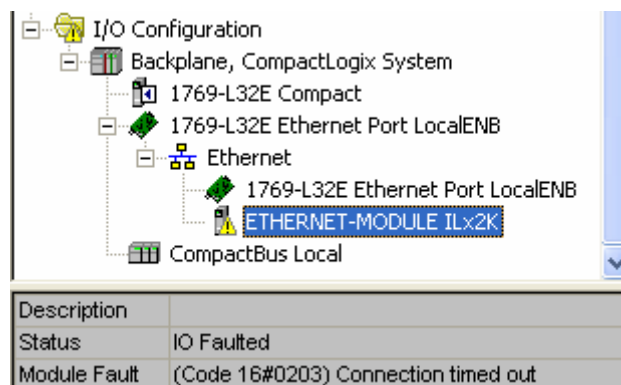


Bild 14) ILx2 Kommunikationsstatus – IO Faulted

## **2.5      Adressierung der Antriebe**

Die Antriebe können auf verschiedene Weisen adressiert werden. Eine detaillierte Beschreibung der Möglichkeiten der Adresseinstellung finden Sie im Produkthandbuch im Kapitel „Schritte zur Inbetriebnahme“.  
Im nächsten Abschnitt werden die Möglichkeiten in Kurzform beschrieben.

### **2.5.1      Drehschalter zur Einstellung der IP Adresse**

Der Antrieb verfügt über 2 Drehschalter zur Adresseinstellung.  
Sind die Drehschalter auf "Stored IP" eingestellt, so lässt sich die IP-Adresse über den integrierten Web-Server oder die Inbetriebnahmesoftware „Lexium CT“ einstellen und verändern.  
Der Antrieb hat im Auslieferungszustand die IP Adresse:  
192.168.100.10 (Default)  
Nähere Informationen finden Sie im Produkthandbuch.

### **2.5.2      Adresseinstellung mittels integriertem Web-Server**

Änderungen der IP Adresse sind nur möglich bei einer Schalterstellung „stored IP“.  
Ist ein anderer Adressierungsmodus eingestellt, dann werden die Änderungen der Kommunikationsparameter beim nächsten Bootvorgang wieder zurückgesetzt.  
Der integrierte Web-Server kann auch gestartet werden, wenn bereits ein Ethernet/IP Master mit dem Antrieb kommuniziert.

- Verbinden Sie sich unter Verwendung eines Web-Browsers mit dem Antrieb.  
Geben Sie im Adressfeld Ihres Internet Browsers die aktuelle IP-Adresse des Antriebes ein.

http:// 192.168.100.10 (Default)



## PLCopen AddOnInstructions ILx2 EthernetIP

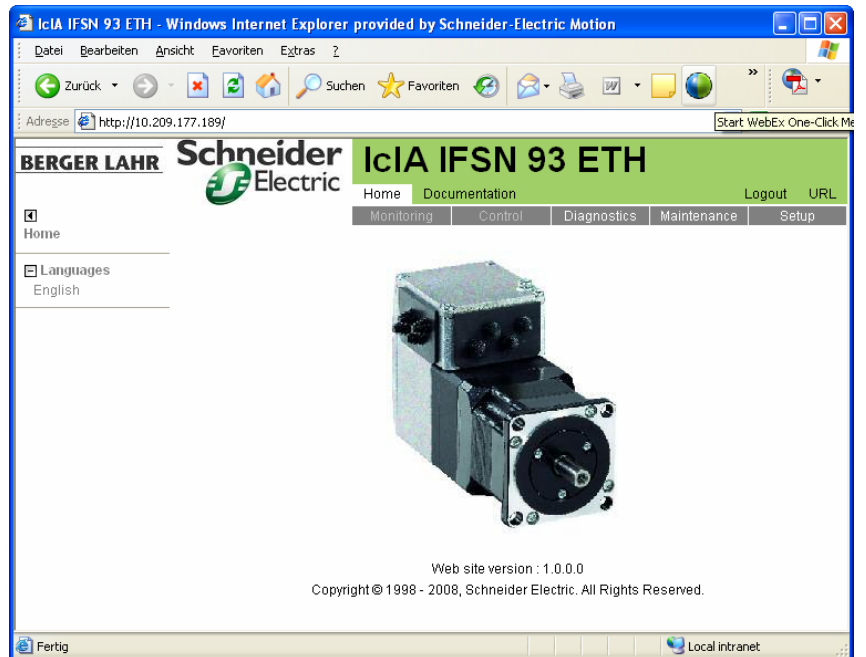


Bild 15) Startseite ILx2 – Web-Server

- Menü: Maintenance -> Communication -> Ethernet
- Benutzername: USER
- Passwort: USER



Bild 16) Einloggen

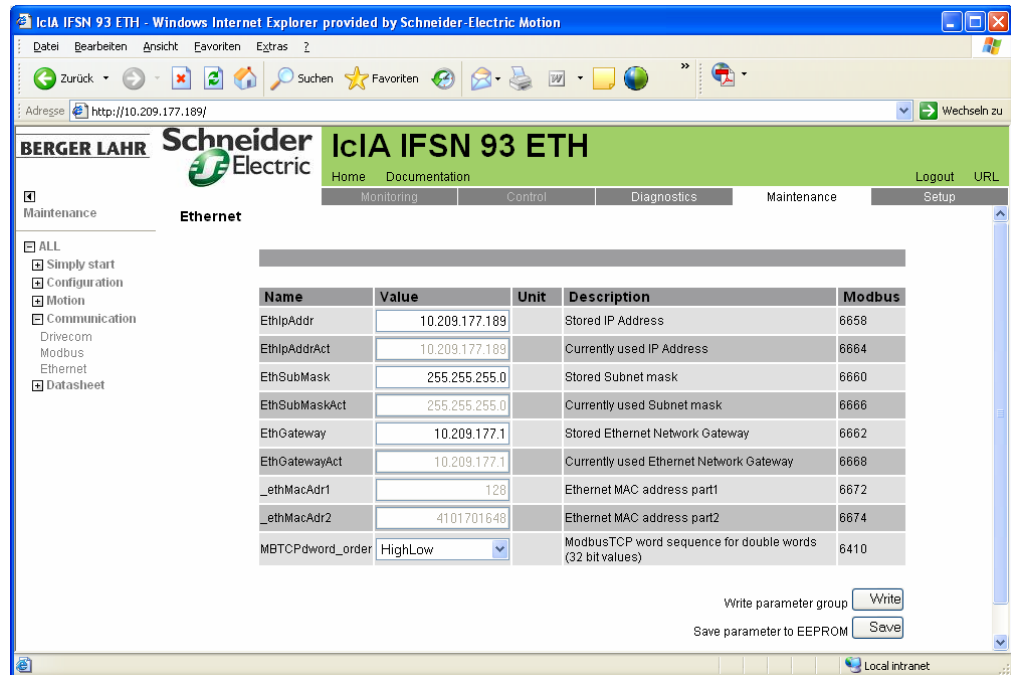


Bild 17) Kommunikationsparameter

- Auf dieser Seite können die Ethernetparameter geändert werden.
- Mit dem Button „Write“ werden die geänderten Parameter in den Antrieb übertragen.
- Mit dem Button „Save“ werden die Parameter im EEPROM gespeichert.
- Die geänderten Kommunikationsparameter werden erst beim nächsten Einschaltvorgang aktiv!

## 2.5.3 Adresseinstellung mittels Lexium CT

Die aktuelle Inbetriebnahmesoftware Lexium CT steht im Internet unter folgender Adresse zum Download bereit:

[www.schneider-electric-motion.com](http://www.schneider-electric-motion.com)

Eine komplette Beschreibung der Software finden Sie im „Lexium CT Produkthandbuch“.

## PLCopen AddOnInstructions ILx2 EthernetIP

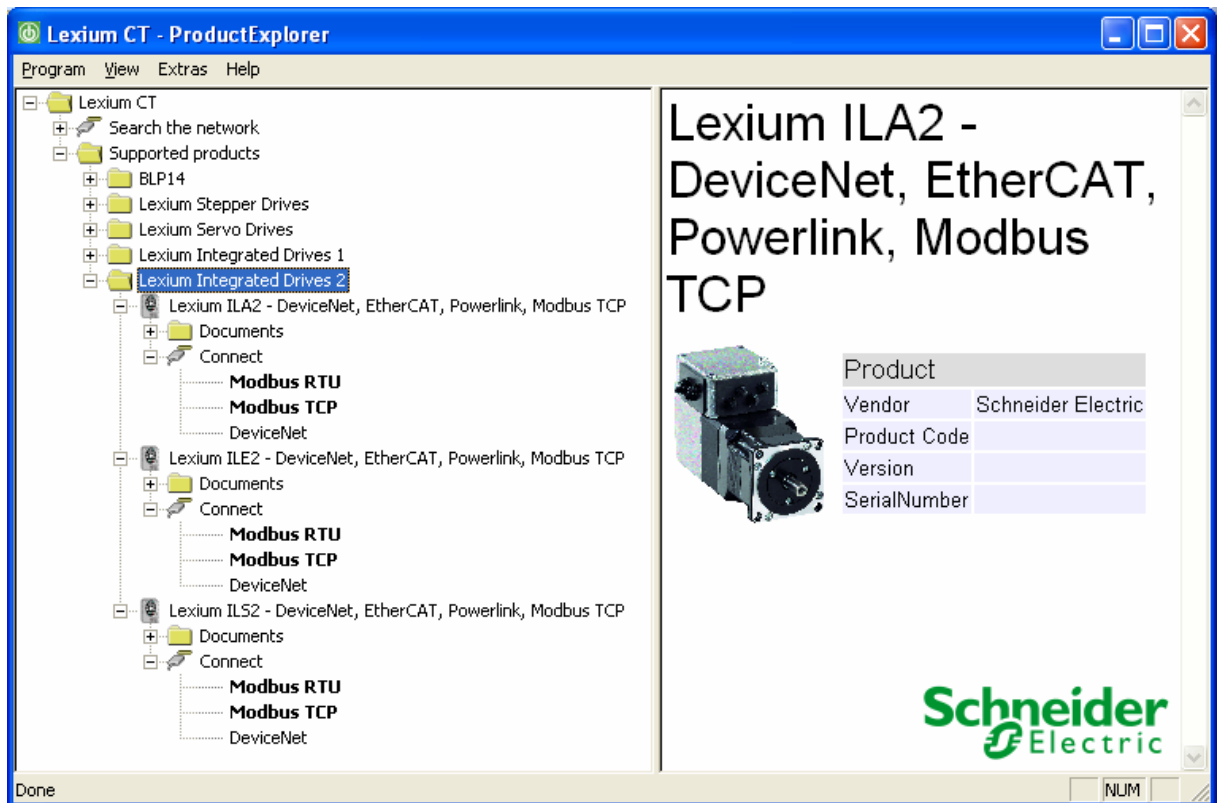


Bild 18) Lexium CT ProductExplorer

- Wählen Sie in der Gruppe Lexium Integrated Drives 2 Ihren Antriebstyp aus.  
ILA2, ILE2 oder ILS2.
- Es stehen 2 verschiedenen Arten der Kommunikation zur Verfügung:
  1. Ethernet Verbindung (Modbus TCP)
  2. Serielle RS485 Verbindung (Modbus RTU) mit einem RS 485 Interface.

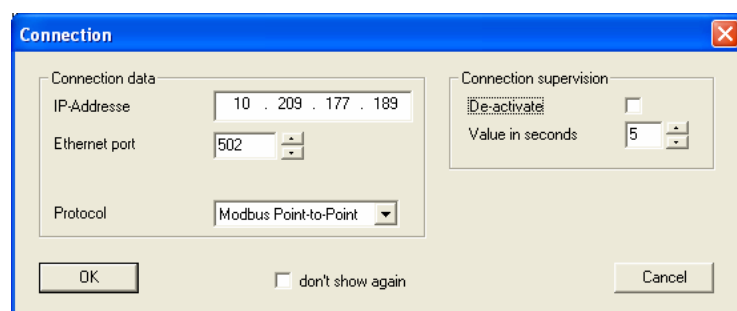


Bild 19) Modbus TCP - Verbindung

## Erstellen einer Applikation PLCopen AddOnInstructions ILx2 Ethernet/IP

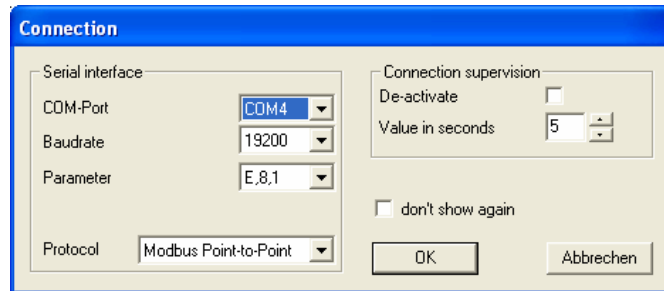


Bild 20) Modbus RTU – Verbindung (RS485 Interface erforderlich)

- Beim Verbinden lädt die Software alle Parameter aus dem Antrieb.
- In der Parametergruppe Kommunikation -> Ethernet finden Sie alle Ethernet Parameter.

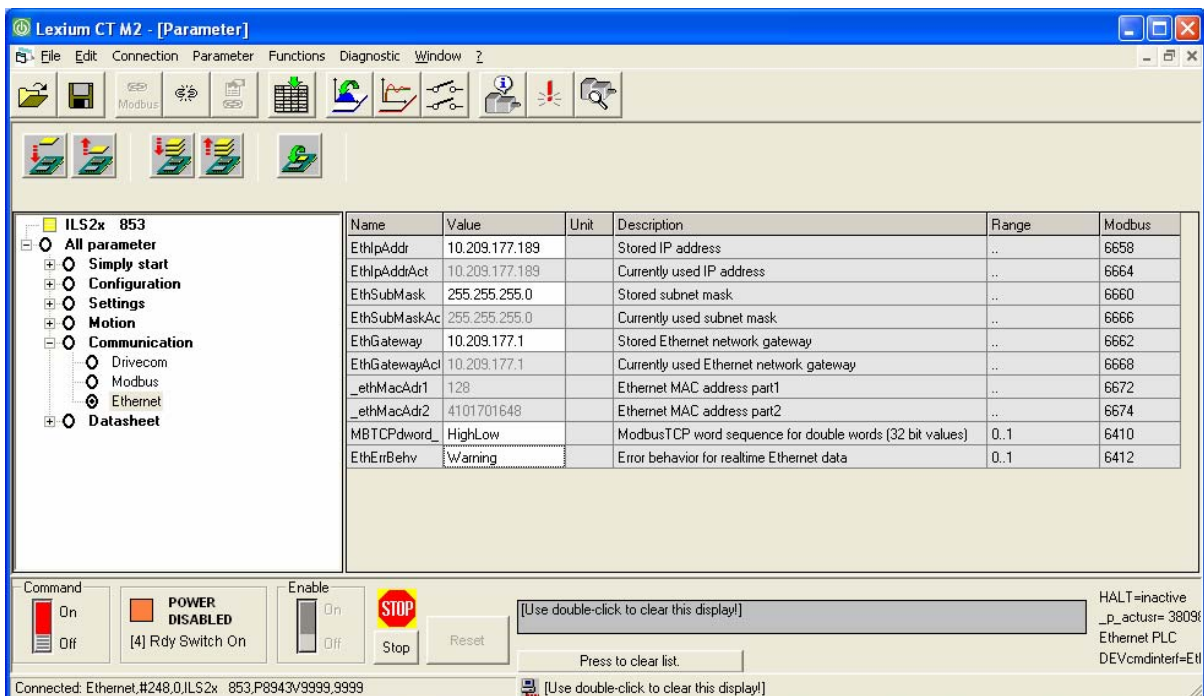


Bild 21) Lexium CT: Kommunikationsparameter -> Ethernet

- Nach Parameteränderung müssen die geänderten Parameter in den Antrieb übertragen werden. Danach müssen die Parameter im EEPROM gespeichert werden.
- Die geänderten Kommunikationsparameter werden erst beim nächsten Einschalten aktiviert.

### 3 Add-On Instructions

Die Add-On Instructions steuern jeweils nur einen Antrieb vom Typ ILx2. Es sind keine Bausteine enthalten, die mehrere Antriebe gleichzeitig steuern oder Antriebe in Abhängigkeit eines anderen Antriebes ansteuern. Da es sich jedoch um Add-On Instructions handelt, können in der Applikation mehrere Instanzen des Bausteins angelegt werden, wobei jede Instanz einen anderen Antrieb unabhängig steuert.

#### 3.1 Bausteinbezeichnung

Bausteine, deren Name mit dem Präfix *MC\_* beginnen, folgen der Spezifikation, die von der [Nutzerorganisation PLCopen](#) erarbeitet wurde. Sie erfüllen damit einen weltweiten Standard für die Programmierung von Motion Control Applikationen.

Bausteine, deren Name den Präfix *MC\_* nicht enthält, halten sich auch an diesen Standard, jedoch sind diese Bausteine in den Spezifikationen noch nicht erfasst.

Zur Unterscheidung, für welche Antriebe und welche Feldbusse die Bausteine verwendet werden, wurden die Postfix *\_ETH* für den Feldbus EtherNet/IP und *\_ILx2* für die Lexium Antriebe verwendet. Somit ist eine eindeutige Zuordnung gegeben.

Zwei typische Beispiele für die Bausteinbezeichnung:

- *MC\_Power\_ETH\_ILx2*
- *MC\_MoveAbsolute\_ETH\_ILx2*

#### 3.2 Signaldiagramme für Ein- und Ausgänge

Das Signalverhalten aller Bausteine ist einheitlich. Es gibt zwei Bausteinarten:

- Bausteine, die über den Eingang *Execute* flankengesteuert werden.
- Bausteine, die über den Eingang *Enable* pegelgesteuert werden.

##### 3.2.1 Steuereingänge

*Der Steuereingang Execute*

Der Steuereingang *Execute* hat zwei Funktionen:

- Eine steigende Flanke startet die Ausführung des Bausteins. Die Eingangsparameter wie z.B. Position und Geschwindigkeit werden übernommen und die Betriebsart gestartet.
- Die Steuerausgänge freizuschalten bzw. abzuschalten: Solange *Execute* = TRUE bleibt, signalisieren die Steuerausgänge den aktuellen Zustand des Bausteins. Das heißt, solange *Execute* = TRUE ist, bleibt genau einer der Ausgänge *Done*, *Busy*,

CommandAborted oder Error ebenfalls TRUE.

Andererseits, wenn Execute = FALSE wird, bevor die Ausführung des Bausteins beendet ist, bleibt Busy = TRUE bis die Ausführung beendet ist. Nach der Ausführung wird einer der Ausgänge Done, CommandAborted oder Error für einen Aufruf TRUE und wird dann FALSE.

#### Der Steuereingang Enable

Der Steuereingang Enable startet bzw. stoppt die Ausführung des Bausteins (Ausnahme: MC\_Power\_ETH\_ILx2). Mit TRUE wird der Baustein wiederholt ausgeführt. Mit FALSE wird die Ausführung sofort beendet und die Steuerausgänge Valid, Busy, CommandAborted und Error werden sofort FALSE.

### 3.2.2 Steuerausgänge

#### Steuerausgang Done

Der Steuerausgang Done signalisiert das Ende der Ausführung. Das ist zum Beispiel bei Erreichen der Zielposition einer Punkt-zu-Punkt Bewegung der Fall.

Bei einigen Bausteinen hat der Ausgang *Done* einen abweichenden Namen, z.B. *InVelocity* beim Baustein für die Betriebsart Geschwindigkeitsprofil. *InVelocity* signalisiert das Erreichen der gewünschten Geschwindigkeit. Die Ausführung des Bausteins ist damit nicht beendet, die Bewegung wird weiterhin überwacht.

#### Steuerausgang Valid

Der Steuerausgang Valid signalisiert, dass die Ausgänge des Bausteins gültige Werte enthalten. Beispielsweise sind am Baustein MC\_ReadParameter\_ETH\_ILx2 die Ausgänge *Value* und *Length* gültig, sobald *Valid* TRUE wird.

#### Steuerausgang Busy

Der Steuerausgang Busy signalisiert, dass der Baustein ausgeführt wird. Erst wenn *Busy* = FALSE wird, ist die Ausführung des Bausteins beendet und erst dann signalisiert einer der Ausgänge *Done*, *Error* oder *CommandAborted*, ob der Baustein ohne bzw. mit Fehler beendet wurde, oder ob er abgebrochen wurde.

#### Steuerausgang CommandAborted

Der Steuerausgang CommandAborted signalisiert, dass ein anderer Baustein die Kontrolle über die Achse übernommen hat. Die Ausführung des *CommandAborted* signalisierenden Bausteins ist damit beendet. Der Baustein MC\_MoveVelocity\_ETH\_ILx2 lässt sich beispielsweise nur durch Abbruch beenden.

### Steuerausgang Error

Der Steuerausgang Error signalisiert das Ende der Ausführung, wenn ein Fehler aufgetreten ist.

### 3.2.3 Signaldiagramme für Bausteine mit Execute Steuereingang

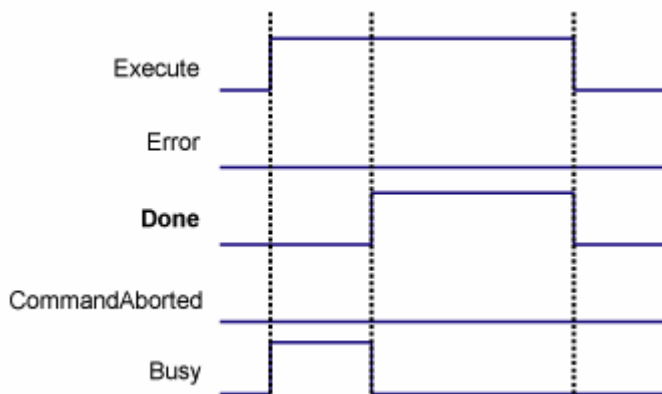


Bild 22) Ausführung ohne Fehler beendet

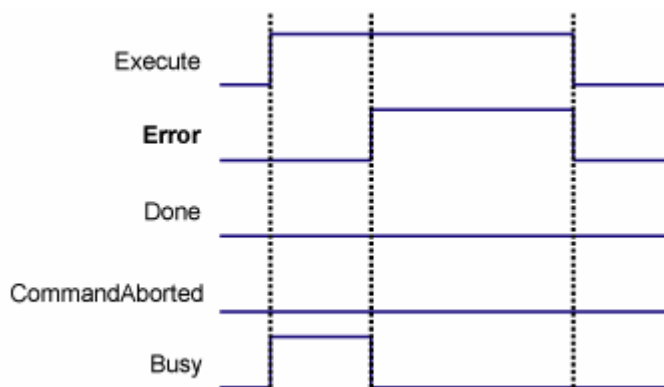


Bild 23) Ausführung mit Fehler beendet

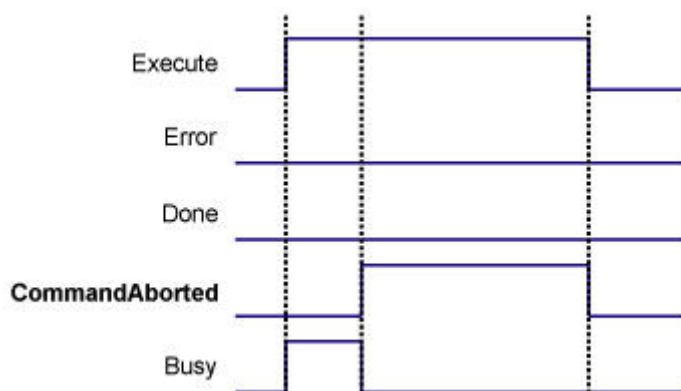


Bild 24) Abbruch der Ausführung durch Übernahme der Achskontrolle in einem anderen Baustein

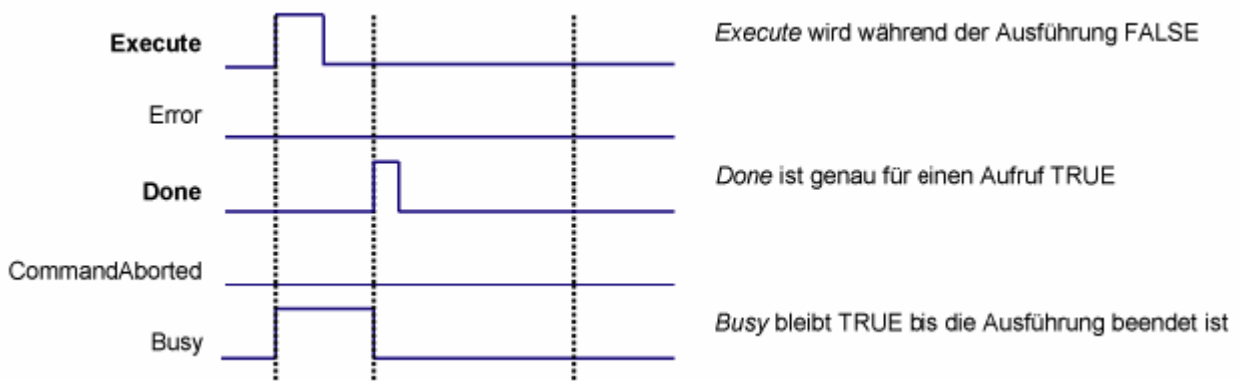


Bild 25) Ausführung ohne Fehler beendet nachdem *Execute* schon während der Ausführung auf FALSE gesetzt wurde

### 3.2.4 Signaldiagramm für Bausteine mit Enable Steuereingang

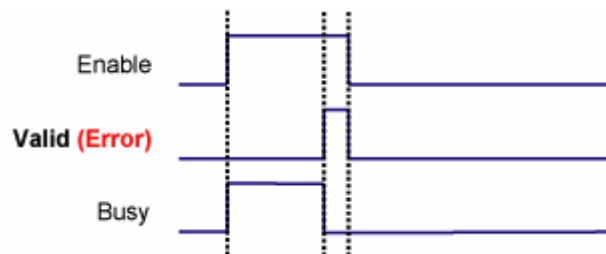


Bild 26) Einmalige Ausführung ohne (mit) Fehler [Ausführung benötigt mehr als einen Aufruf]

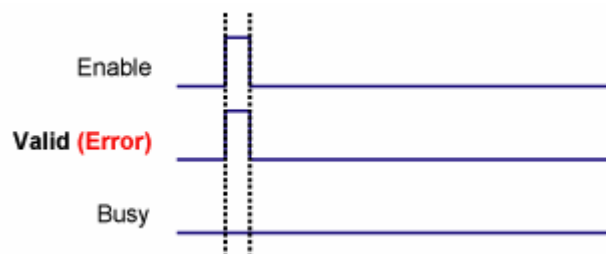


Bild 27) Einmalige Ausführung ohne (mit) Fehler [Ausführung benötigt nur einen Aufruf]

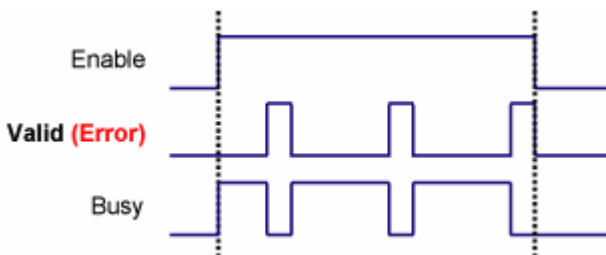


Bild 28) Wiederholte Ausführung ohne (mit) Fehler [Ausführung benötigt mehr als einen Aufruf]



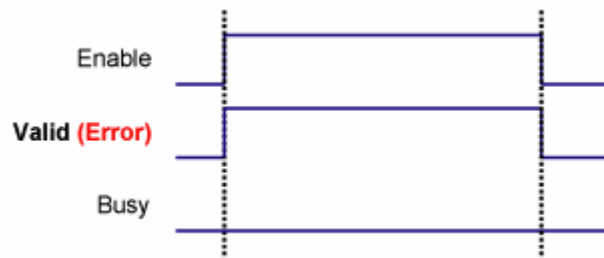


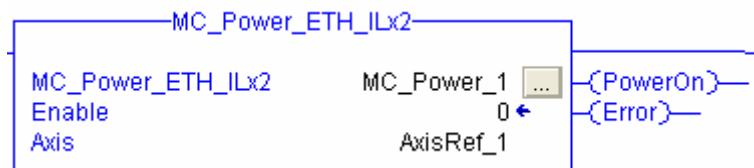
Bild 29) Wiederholte Ausführung ohne (mit) Fehler [Ausführung benötigt nur einen Aufruf]

### 3.3 Die Beschreibung der Bausteine

Im nachfolgenden Kapitel finden Sie eine nach Verwendungszweck gegliederte Beschreibung der Bausteine. Im Anschluss daran finden Sie dann die detaillierte Beschreibung der Bausteine, gegliedert nach den Namen der Bausteine.

Die detaillierte Beschreibung der Bausteine gliedert sich wie folgt:

#### Grafische Darstellung



#### Ein-/Ausgänge

Die Beschreibung der Ein- und Ausgänge, mit Angabe des Typs, des zulässigen Wertebereichs und der Bedeutung der Werte. Zum Beispiel:

#### Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Enable	Typ BOOL (Wertebereich TRUE, FALSE) Initialwert: FALSE TRUE: Motorstrom einschalten. FALSE: Motorstrom ausschalten.
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

*Ausgangsvariablen*

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
Status	Typ: BOOL (Wertebereich TRUE, FALSE) Initialwert: FALSE Zeigt den Zustand des Motorstroms an. Der Zustand wird vom Antrieb bei jeder Änderung gemeldet. FALSE: Motorstrom ist ausgeschaltet. TRUE: Motorstrom ist eingeschaltet.
Error	Typ: BOOL (Wertebereich:TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

*Aufgabe des Bausteins*

Ein-/Ausschalten des Motorstroms. Mit TRUE am Eingang *Enable* wird der Motorstrom eingeschaltet. Sobald der Motorstrom eingeschaltet ist, wird der Ausgang *Status* gesetzt. Mit FALSE am Eingang *Enable* wird der Motorstrom ausgeschaltet. Sobald der Motorstromlos ist, wird der Ausgang *Status* zurückgesetzt. Treten bei der Ausführung Fehler auf, wird der Ausgang *Error* gesetzt

*Hinweis*

Mit dem Abschalten der Stromversorgung des Antriebs wird auch der Motorstrom abgeschaltet. Da der Antrieb diesen Zustandswechsel nicht mehr melden kann, bleibt *Status* = TRUE. Beim Wiedereinschalten der Stromversorgung des Antriebs wird der aktuelle stromlose Zustand des Motors gemeldet, der Ausgang *Status* wird daraufhin FALSE.

Das Einschalten des Motorstroms führt in einen der Betriebszustände *StandStill* oder *ErrorStop*, je nach dem, ob ein Fehler erkannt wurde oder nicht. Der Betriebszustand wird mit `MC_ReadStatus_ETH_ILx2` gelesen.

Das Ausschalten des Motorstroms ist aus jedem Zustand möglich. Ein zu diesem Zeitpunkt aktiver Bewegungsbaustein wird abgebrochen.

### 3.4 Verwendungszweck der Bausteine

#### 3.4.1 Initialisierung

Der Baustein zur Initialisierung schaltet den Antrieb in einen Zustand, der die Benutzung weiterer Bausteine erlaubt. Aktiviert und deaktiviert die Endstufe des Antriebs.

Baustein	Aufgabe
MC_Power_ETH_ILx2	Aktiviert und deaktiviert die Endstufe des Antriebs

#### 3.4.2 Manuellfahrt

Der Baustein für die Manuellfahrt bewegt den Antrieb im Tippbetrieb. Ein kurzes Signal an den Eingängen verfährt den Antrieb in positiver oder negativer Richtung um ein kleines Stück. Bei einem dauerhaften Signal geht der Antrieb in eine kontinuierliche Bewegung über.

Baustein	Aufgabe
MC_Jog_ETH_ILx2	Manuellfahrt in positiver oder negativer Richtung

#### 3.4.3 Referenzierung

Mit der Betriebsart Referenzierung wird ein absoluter Maßbezug der Motorposition zu einer definierten Achsposition hergestellt. Eine Referenzierung ist möglich durch:

- Referenzfahrt oder
- Maßsetzen

Die Referenzfahrt legt den Maßbezug durch Anfahren eines Schalters fest. Mit der Referenzfahrt wird eine definierte Position auf der Achse angefahren. Die definierte Position ist durch einen mechanischen Schalter festgelegt. Abhängig von der Motorvariante, die Sie einsetzen, sind folgende Schalter möglich:

- Endschanter in positiver und negativer Richtung
- Referenzschalter
- Indexpuls des im Motor verbauten Encodersystems

Zusätzlich können Endschanter und Indexpuls noch kombiniert werden. Welche Referenzfahrten von ihrem Lexium ILx2 Antrieb unterstützt werden, entnehmen Sie der zum Antrieb gehörenden Produkthandbuch.

Nach Erreichen der Position wird automatisch ein Maßbezug hergestellt. Dadurch wird die erreichte Position zur absoluten Anwenderposition.

Für die Referenzfahrt sind Such- und Freifahrtgeschwindigkeit sowie Abstand zur Schaltkante und Freifahrtweg einstellbar. Eine

Referenzfahrt muss vollständig durchgeführt werden, damit der neue Referenzpunkt gültig ist. Wenn eine Referenzfahrt abgebrochen wurde, muss Sie neu gestartet werden.

Das Maßsetzen legt den Maßbezug relativ zur aktuellen Motorposition fest. Maßsetzen bietet die Möglichkeit, einen Punkt auf der Achse als Bezugspunkt festzulegen, auf den sich folgende Positionen beziehen. Der Bezugspunkt für Sollpositionen wird auf die neue Maßsetzposition verschoben.

Maßsetzen kann nur im Stillstand des Motors ausgeführt werden. Das Maßsetzen kann eingesetzt werden, um kontinuierliche Relativ-Positionierungen ohne Überschreiten der Bereichsüberlauf durchzuführen.

Baustein	Aufgabe
MC_Home_ETH_ILx2	Referenzfahrt auslösen
MC_SetPosition_ETH_ILx2	Maßsetzen

### 3.4.4 Betriebsart Punkt-zu-Punkt

In der Betriebsart Punkt-zu-Punkt wird der Motor mit einem Bewegungsbaustein von einem Punkt A auf einen Punkt B positioniert. Der Positionierweg wird absolut mit Bezug auf den Nullpunkt der Achse oder relativ, bezogen auf die ursprüngliche Zielposition oder aktuelle Motorposition, angegeben. Vor einer Positionierung muss der Referenzpunkt durch eine Referenzierung definiert werden.

Baustein	Aufgabe
MC_MoveAbsolute_ETH_ILx2	Absolut positionieren
MC_MoveRelative_ETH_ILx2	Relativ zur aktuellen Motorposition positionieren
MC_MoveAdditive_ETH_ILx2	Relativ zur ursprünglichen Zielposition positionieren

### 3.4.5 Betriebsart Geschwindigkeitsprofil

In der Betriebsart Geschwindigkeitsprofil wird dem Motor eine Sollgeschwindigkeit vorgegeben und eine Bewegung ohne Zielposition gestartet. Der Motor bewegt sich mit dieser Geschwindigkeit, bis eine andere Sollgeschwindigkeit übergeben oder die Betriebsart durch Ausführung eines anderen Bausteins unterbrochen wird.

Übergänge zwischen zwei Drehzahlen werden über den Profilgenerator im Antrieb definiert. Aus den Parametern für Beschleunigung, Verzögerung, Sollgeschwindigkeit und Istgeschwindigkeit errechnet der Profilgenerator die Überleitung in die neue Geschwindigkeit.

Baustein	Aufgabe
MC_MoveVelocity_ETH_ILx2	Betriebsart Geschwindigkeitsprofil einschalten

### 3.4.6 Stoppen

Jede Betriebsart kann durch Stoppen des Antriebs abgebrochen werden. Dies erzeugt keinen Fehler. Der abgebrochene Bewegungsbaustein beendet die Ausführung indem der Ausgang *CommandAborted* = TRUE gesetzt wird. Der Antrieb wechselt in den Status „Stopping“ und nach dem Stillstand und dem Zurücksetzen des Eingangs *Execute* in „StandStill“. Erst dann können neue Bewegungen ausgeführt werden.

<b>Baustein</b>	<b>Aufgabe</b>
MC_Stop_ETH_ILx2	Antrieb stoppen

### 3.4.7 Schnelle Positionserfassung

Über 2 parametrierbare Kanäle lässt sich die Motorposition mit einer vom Motortyp abhängigen Genauigkeit erfassen. Zu den Details der schnellen Positionserfassung lesen Sie den Abschnitt im Produkthandbuch. Nur die Integrierten Antriebe Lexium ILA2E und ILS2E besitzen die Funktion der schnellen Positionserfassung.

Mögliche auslösende Ereignisse:

- Steigende Flanke oder fallende Flanke am Signaleingang CAP1
- Steigende Flanke oder fallende Flanke am Signaleingang CAP2

<b>Baustein</b>	<b>Aufgabe</b>
MC_TouchProbe_ETH_ILx2	Positionserfassung einstellen und starten
MC_AbortTrigger_ETH_ILx2	Positionserfassung abbrechen

### 3.4.8 Parameter lesen

Mit folgenden Bausteinen können spezielle Parameter sehr einfach gelesen werden. Zusätzlich gibt es einen universellen Baustein, der einen Lesezugriff auf alle Parameter des Lexium ILx2E Antrieb ermöglicht. Eine genaue Beschreibung aller Parameter des Antriebes finden Sie im Produkthandbuch.

Die Bausteine zum Auslesen der Referenzposition und Referenzgeschwindigkeit direkt aus dem Profildgenerator liefern nur beim Lexium ILA2E andere Werte als die anderen Bausteine zum Lesen von Position und Geschwindigkeit. Denn beim Servomotor werden die aktuellen Positionen und Geschwindigkeiten aus dem Encoder im Motor abgeleitet.

<b>Baustein</b>	<b>Aufgabe</b>
MC_ReadActualPosition_ETH_ILx2	Liest die aktuelle Position in Anwendereinheiten
MC_ReadActualVelocity_ETH_ILx2	Liest die aktuelle Geschwindigkeit in Anwendereinheiten
MC_ReadStatus_ETH_ILx2	Liest den aktuellen Status des Antriebs
MC_ReadParameter_ETH_ILx2	Liest den angegebenen Parameter aus dem Antrieb

### 3.4.9 Parameter schreiben

Mit folgenden Bausteinen können spezielle Parameter sehr einfach geschrieben werden. Auch lassen sich die Parameter auf die Werkseinstellungen zurücksetzen oder dauerhaft im Antrieb speichern. Zusätzlich gibt es einen universellen Baustein, der einen Schreibzugriff auf alle Parameter des Lexium ILx2E Antrieb ermöglicht. Eine genaue Beschreibung aller Parameter des Antriebes finden Sie im Produkthandbuch.

Baustein	Aufgabe
MC_WriteParameter_ETH_ILx2	Parameter einstellen

### 3.4.10 Ein-/Ausgänge

Mit folgenden Bausteinen können die digitalen Ein- und Ausgänge des Lexium ILx2E gelesen werden.

Baustein	Aufgabe
MC_ReadDigitalInput_ETH_ILx2	Eingänge lesen

### 3.4.11 Fehlerbehandlung

Zur Fehlerbehandlung hat jeder Baustein einen Ausgang *Error*, der gesetzt wird, wenn ein Fehler auftritt. Zur genaueren Analyse der Fehlerursache wird der Baustein MC\_ReadAxisError\_ETH\_ILx2 aufgerufen. Mit MC\_Reset\_ETH\_ILx2 wird die Fehlerzelle wieder gelöscht und damit frei für zukünftige Fehlermeldungen.

Die Fehlerzelle enthält den Fehlercode und die Fehlerklasse eines aufgetretenen Fehlers. Ein neu aufgetretener Fehler wird eingetragen wenn die Fehlerzelle frei war. Ist die Fehlerzelle belegt, dann wird die frühere Fehlermeldung nicht überschrieben, sondern die neue Fehlermeldung wird ignoriert.

Mögliche Fehlerquellen sind:

- Bausteinfehler
- Antriebsfehler

Bausteinfehler werden z.B. durch unerlaubte Eingangswerte erzeugt. Antriebsfehler werden meist durch Ereignisse wie z.B. Erreichen eines Endschalters ausgelöst. MC\_Reset\_ETH\_ILx2 setzt auch den Fehler im Antrieb zurück.

Baustein	Aufgabe
MC_ReadAxisError_ETH_ILx2	Fehler auslesen
MC_Reset_ETH_ILx2	Fehler löschen

# PLCopen AddOnInstructions ILx2 EthernetIP

## 3.5 Detaillierte Bausteinbeschreibung

### 3.5.1 MC\_UpdateInputData\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
CommunicationOK	Typ: BOOL (Wertebereich: FALSE,TRUE Initialwert: FALSE Der Anwender muss den Kommunikationsstatus zum Slave mit Hilfe von GSV (GetSystemValue) außerhalb der AddOnInstructions überwachen. Der Status „Running“ kann an diesem Eingang mit einer Variable vom Typ BOOL übergeben werden. Wird der Kommunikationsstatus nicht überwacht, so muss an diesem Eingang statisch ein „TRUE“ eingetragen werden.
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse> Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.
ScannerIn	Typ: ETHERNET MODULE INPUT DATA Initialwert: leer Datenarray der Ethernet Eingangsdaten aus der Controller Tag Liste.

Ausgangsvariablen

Dieser Baustein hat keine Ausgangsvariable.

Aufgabe des Bausteins

Dieser Baustein muss aufgerufen werden, bevor eine andere Add-On Instruction für den Antrieb aufgerufen wird. Dieser Baustein kopiert die EthernetIP Eingangsdaten in die Axis\_Ref Struktur, welche diese

Eingangsdaten allen AddOnInstructions zur Verfügung stellt.

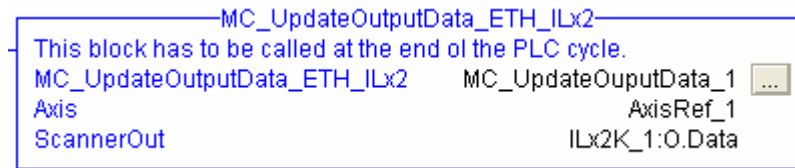
Die Axis\_Ref Struktur dient dazu, dass der Zugriff auf den Antrieb durch mehrere Add-On Instructions funktioniert.

*Hinweis*

Diese Funktion muss zyklisch aufgerufen werden.

**3.5.2 MC\_UpdateOutputData\_ETH\_ILx2**

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

*Ausgangsvariablen*

Variable	Mögliche Werte, Bedeutung
ScannerOut	Typ: ETHERNET MODULE OUTPUT DATA Initialwert: leer Datenarray der Ethernet Ausgangsdaten aus der Controller Tag Liste.

*Aufgabe des Bausteins*

Dieser kopiert die Ausgangsdaten zum Antrieb, das heißt die Daten werden aus der AxisRef Struktur in die Controller Tags kopiert.

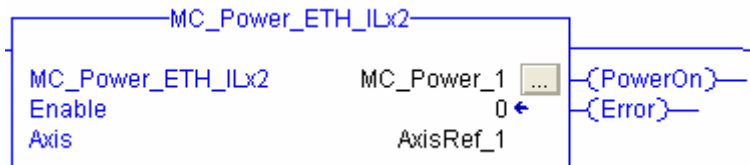
*Hinweis*

Diese Funktion muss zyklisch aufgerufen werden.



## 3.5.3 MC\_Power\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Enable	Typ BOOL (Wertebereich TRUE, FALSE) Initialwert: FALSE TRUE: Motorstrom einschalten. FALSE: Motorstrom ausschalten.
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Status	Typ: BOOL (Wertebereich TRUE, FALSE) Initialwert: FALSE zeigt den Zustand des Motorstroms an. Der Zustand wird vom Antrieb bei jeder Änderung gemeldet. FALSE: Motorstrom ist ausgeschaltet. TRUE: Motorstrom ist eingeschaltet.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

Aufgabe des Bausteins

Ein-/Ausschalten des Motorstroms. Mit TRUE am Eingang *Enable*

wird der Motorstrom eingeschaltet. Sobald der Motorstrom eingeschaltet ist, wird der Ausgang *Status* gesetzt. Mit FALSE am Eingang *Enable* wird der Motorstrom ausgeschaltet. Sobald der Motor stromlos ist, wird der Ausgang *Status* zurückgesetzt. Treten bei der Ausführung Fehler auf, wird der Ausgang *Error* gesetzt

#### *Hinweis*

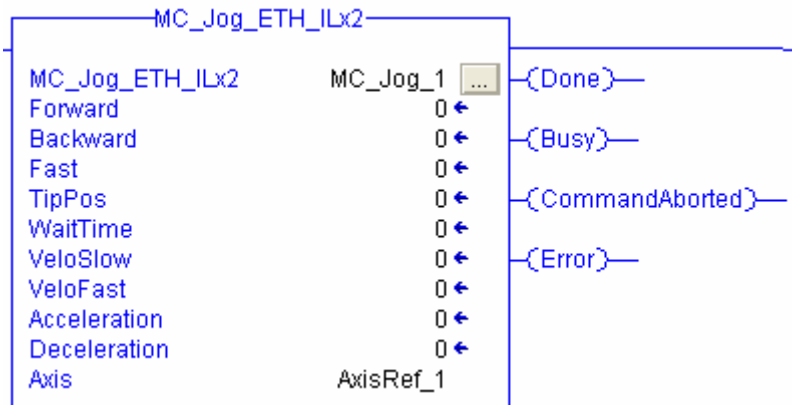
Mit dem Abschalten der Stromversorgung des Antriebs wird auch der Motorstrom abgeschaltet. Da der Antrieb diesen Zustandswechsel nicht mehr melden kann, bleibt *Status* = TRUE. Beim Wiedereinschalten der Stromversorgung des Antriebs wird der aktuelle stromlose Zustand des Motors gemeldet, der Ausgang *Status* wird daraufhin FALSE.

Das Einschalten des Motorstroms führt in einen der Betriebszustände *StandStill* oder *ErrorStop*, je nach dem, ob ein Fehler erkannt wurde oder nicht. Der Betriebszustand wird mit *MC\_ReadStatus\_ETH\_ILx2* gelesen.

Das Ausschalten des Motorstroms ist aus jedem Zustand möglich. Ein zu diesem Zeitpunkt aktiver Bewegungsbaustein wird abgebrochen.

## 3.5.4 MC\_Jog\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Forward	Typ BOOL (Wertebereich TRUE, FALSE) Initialwert: FALSE FALSE: keine Bewegung in positive Richtung TRUE: Start der Bewegung in positive Richtung
Backward	Typ BOOL (Wertebereich TRUE, FALSE) Initialwert: FALSE FALSE: keine Bewegung in negative Richtung TRUE: Start der Bewegung in negative Richtung
Fast	Typ BOOL (Wertebereich TRUE, FALSE) Initialwert: FALSE Die Geschwindigkeit kann auch während der Fahrt zwischen zwei Werten ausgewählt werden: FALSE: Geschwindigkeit VeloSlow ist ausgewählt TRUE: Geschwindigkeit VeloFast ist ausgewählt
TipPos	Typ DINT (Wertebereich: 0..2147483647) Initialwert: 20 0: der Motor geht sofort in eine kontinuierliche Bewegung über.

Variable	Mögliche Werte, Bedeutung
	>0: Wegs [usr], den der Motor beim Start zuerst zurücklegt, bevor er nach der Verzögerungszeit (WaitTime) in eine kontinuierliche Bewegung übergeht.
WaitTime	Typ UINT (Wertebereich: 1ms..32767ms) Initialwert: 500ms  Verzögerungszeit [ms], die beginnt, nachdem der Motor einen definierten Weg (TipPos) zurückgelegt hat und nach deren Ablauf der Motor in eine kontinuierliche Bewegung übergeht.
VeloSlow	Typ DINT (Wertebereich: 1..13200) Initialwert: 60  Geschwindigkeit [Min-1] für die Fahrt, wenn Fast = FALSE ist.
VeloFast	Typ: DINT (Wertebereich: 1..13200) Initialwert: 180  Geschwindigkeit [Min-1] für die Fahrt, wenn Fast = TRUE ist.
Acceleration	Typ: DINT (Wertebereich: 1..3000000) Initialwert: 600  Wert für die Steilheit der Beschleunigungsrampe [1 Min-1/s]
Deceleration	Typ: DINT (Wertebereich: 200..3000000) Initialwert: 750  Wert für die Steilheit der Verzögerungsrampe [1 Min-1/s]
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer  Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

*Ausgangsvariablen*

Variable	Mögliche Werte, Bedeutung
Done	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung ist beendet.  FALSE: die Ausführung ist (noch) nicht

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	beendet.
Busy	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
CommandAborted	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung wurde durch einen anderen Baustein abgebrochen. FALSE: die Ausführung ist (noch) nicht abgebrochen worden.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

### Aufgabe des Bausteins

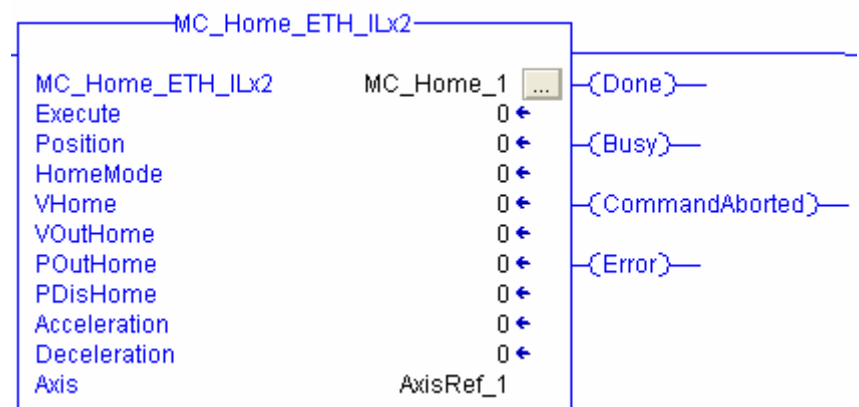
Mit TRUE an *Forward* oder *Backward* wird die Manuellfahrt gestartet.

Sind *Forward* und *Backward* = FALSE, wird die Betriebsart beendet und *Done* gesetzt.

Sind *Forward* und *Backward* = TRUE, bleibt die Betriebsart aktiv, die Manuellfahrt wird gestoppt, *Busy* bleibt gesetzt.

### 3.5.5 MC\_Home\_ETH\_ILx2

#### Grafische Darstellung



## Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Execute	<p>Typ BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>Steigende Flanke: FALSE-&gt;TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.</p> <p>Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:</p> <p>FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.</p> <p>TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.</p>
Position	<p>Typ DINT</p> <p>(Wertebereich:-2147483648..2147483647) Initialwert: 0</p> <p>Maßsetzen auf Maßsetzposition (Absolutposition setzen) in Anwendereinheiten. Position ist die aktuelle Motorposition nach erfolgreicher Referenzierung</p>
HomeMode	<p>Typ: UINT</p> <p>(Wertebereich 1..35) Initialwert: 1</p> <p>Welche Referenzierungsarten vom Antrieb unterstützt werden, entnehmen Sie dem Produkthandbuch.</p>
Vhome	<p>Typ: UINT</p> <p>(Wertebereich: 1..13200) Initialwert: 60</p> <p>Geschwindigkeit für die Suche des Referenzschalters oder eines Endschalters [Min-1]. Nach dem Erkennen der Schaltkante stoppt der Antrieb.</p>
VoutHome	<p>Typ: UINT</p> <p>(Wertebereich: 1..3000) Initialwert: 6</p> <p>Geschwindigkeit für das Freifahren zurück zur Schaltkante [Min-1]. Die maximale Fahrstrecke zum Suchen der Schaltkante kann mit dem Parameter POutHome begrenzt werden</p>
PoutHome	<p>Typ: DINT</p> <p>(Wertebereich: 0..2147483647) Initialwert: 0</p>

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	<p>0: Ausfahrkontrolle abgeschaltet.</p> <p>&gt; 0: Ausfahrweg [usr], maximale Fahrstrecke zum Suchen der Schaltkante. Wenn innerhalb dieser Strecke die Schaltkante nicht gefunden wird, erfolgt ein Abbruch der Referenzfahrt mit Fehler.</p>
PdisHome	<p>Typ: DINT</p> <p>(Wertebereich: 1..2147483647) Initialwert: 200</p> <p>Abstand von der Schaltkante zum Referenzpunkt in Anwendereinheiten.</p> <p>Der Antrieb fährt nach Erreichen der Schaltkante weiter, bis der Abstand erreicht ist.</p>
Acceleration	<p>Typ: DINT</p> <p>(Wertebereich: 1..3000000) Initialwert: 600</p> <p>Wert für die Steilheit der Beschleunigungsrampe [1 Min-1/s]</p>
Deceleration	<p>Typ: DINT</p> <p>(Wertebereich: 200..3000000) Initialwert: 750</p> <p>Wert für die Steilheit der Verzögerungsrampe [1 Min-1/s]</p>
Axis	<p>Typ: AXIS_REF_EIP_ILx2</p> <p>(Wertebereich: &lt;Name der Achse&gt;) Initialwert: leer</p> <p>Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.</p>

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Done	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>TRUE: die Ausführung ist beendet.</p> <p>FALSE: die Ausführung ist (noch) nicht beendet.</p>
Busy	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>TRUE: der Baustein wird ausgeführt.</p> <p>FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.</p>
CommandAborted	<p>Typ: BOOL</p>

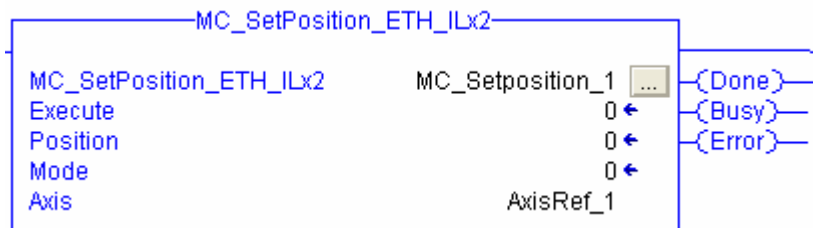
Variable	Mögliche Werte, Bedeutung
	(Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung wurde durch einen anderen Baustein abgebrochen.  FALSE: die Ausführung ist (noch) nicht abgebrochen worden.
Error	Typ: BOOL  (Wertebereich:TRUE; FALSE) Initialwert: FALSE  TRUE: die Ausführung wurde mit Fehler beendet.  FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

*Aufgabe des Bausteins*

Konfigurieren und Starten der Referenzfahrt.

**3.5.6 MC\_SetPosition\_ETH\_ILx2**

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  Steigende Flanke: FALSE->TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.  Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:  FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.  TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit



## PLCopen AddOnInstructions ILx2 EthernetIP

---

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
	Execute = FALSE aufgerufen wird.
Position	Typ: DINT (Wertebereich: -2147483648..2147483647) Initialwert: 0  Maßsetzen auf Maßsetzposition in Anwendereinheiten.
Mode	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  FALSE: aktuelle Motorposition auf Position setzen.  TRUE: Position zur aktuellen Motorposition addieren.
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer  Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

---

### *Ausgangsvariablen*

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
Done	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung ist beendet.  FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Baustein wird ausgeführt.  FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE  TRUE: die Ausführung wurde mit Fehler beendet.  FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

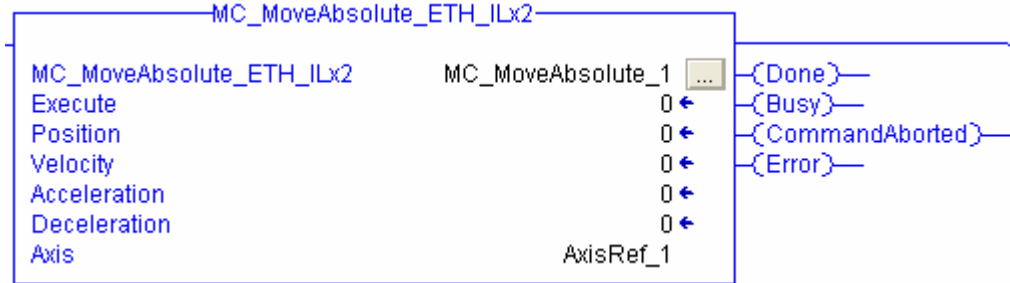
---

### *Aufgabe des Bausteins*

Maßsetzen

3.5.7 MC\_MoveAbsolute\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  Steigende Flanke: FALSE->TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.  Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:  FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.  TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.
Position	Typ: DINT (Wertebereich: -2147483648..2147483647) Initialwert: 0  Wert für die absolute Zielposition in Anwendereinheiten.
Velocity	Typ: INT (Wertebereich: 1..13200) Initialwert: 60 )  Wert für die Sollgeschwindigkeit der Bewegung [Min-1].
Acceleration	Typ: DINT (Wertebereich: 1..3000000) Initialwert: 600  Wert für die Steilheit der

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	Beschleunigungsrampe [1 Min-1/s]
Deceleration	Typ: DINT (Wertebereich: 200..3000000) Initialwert: 750 Wert für die Steilheit der Verzögerungsrampe [1 Min-1/s]
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Done	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
CommandAborted	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung wurde durch einen anderen Baustein abgebrochen. FALSE: die Ausführung ist (noch) nicht abgebrochen worden.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

### Aufgabe des Bausteins

Positionierung auf die absolute Zielposition *Position* mit der

Geschwindigkeit *Velocity*

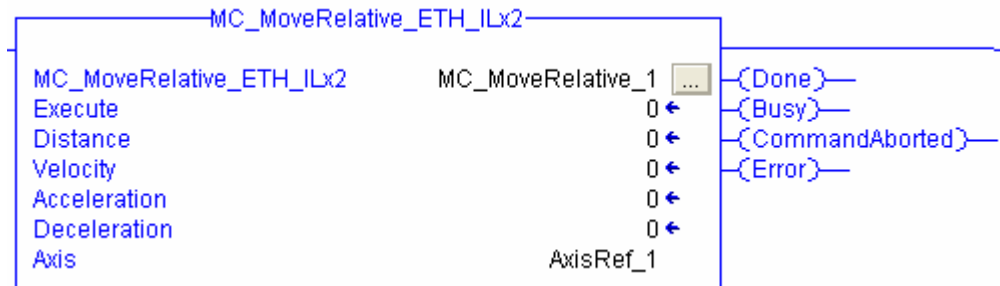
*Hinweis*

Positionsüberlauf

Eine Absolut-Positionierung kann nicht nach einem Positionsüberlauf gestartet werden, da durch den Positionsüberlauf der absolute Positionsbezug verloren geht. Ob der absolute Positionsbezug noch gegeben ist, kann am Ausgang *Referenced* des Bausteins MC\_ReadStatus\_ETH\_ILx2 gelesen werden

**3.5.8 MC\_MoveRelative\_ETH\_ILx2**

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Execute	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>Steigende Flanke: FALSE-&gt;TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.</p> <p>Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:</p> <p>FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.</p> <p>TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.</p>
Distance	<p>Typ: DINT</p> <p>(Wertebereich: -2147483648..2147483647) Initialwert: 0</p> <p>Wert für den Weg relativ zur aktuellen Motorposition in Anwindereinheiten.</p>

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
Velocity	Typ: INT (Wertebereich: 1..13200) Initialwert: 60 )  Wert für die Sollgeschwindigkeit der Bewegung [Min-1].
Acceleration	Typ: DINT (Wertebereich: 1..3000000) Initialwert: 600  Wert für die Steilheit der Beschleunigungsrampe [1 Min-1/s]
Deceleration	Typ: DINT (Wertebereich: 200..3000000) Initialwert: 750  Wert für die Steilheit der Verzögerungsrampe [1 Min-1/s]
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer  Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Done	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
CommandAborted	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung wurde durch einen anderen Baustein abgebrochen. FALSE: die Ausführung ist (noch) nicht abgebrochen worden.

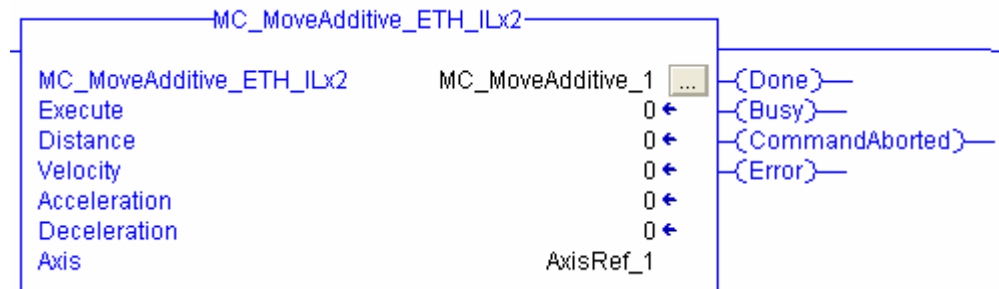
Variable	Mögliche Werte, Bedeutung
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

Aufgabe des Bausteins

Positionierung um die Weg *Distance* mit der Geschwindigkeit *Velocity*

### 3.5.9 MC\_MoveAdditive\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Execute	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE Steigende Flanke: FALSE->TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird. Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge: FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE. TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.
Distance	Typ: DINT (Wertebereich: -2147483648..2147483647) Initialwert: 0

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	Wert für den zur ursprünglichen Zielposition zusätzliche relativen Weg in Anwindereinheiten.
Velocity	Typ: INT (Wertebereich: 1..13200) Initialwert: 60 )
	Wert für die Sollgeschwindigkeit der Bewegung [Min-1].
Acceleration	Typ: DINT (Wertebereich: 1..3000000) Initialwert: 600
	Wert für die Steilheit der Beschleunigungsrampe [1 Min-1/s]
Deceleration	Typ: DINT (Wertebereich: 200..3000000) Initialwert: 750
	Wert für die Steilheit der Verzögerungsrampe [1 Min-1/s]
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer
	Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Done	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
CommandAborted	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung wurde durch einen

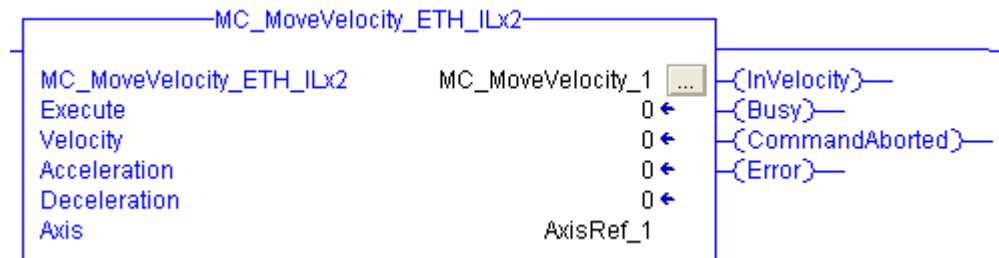
Variable	Mögliche Werte, Bedeutung
	anderen Baustein abgebrochen.  FALSE: die Ausführung ist (noch) nicht abgebrochen worden.
Error	Typ: BOOL  (Wertebereich: TRUE; FALSE) Initialwert: FALSE  TRUE: die Ausführung wurde mit Fehler beendet.  FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

*Aufgabe des Bausteins*

Positionierung um den Weg *Distance* zusätzlich zur ursprünglichen Zielposition mit der Geschwindigkeit *Velocity*.

**3.5.10 MC\_MoveVelocity\_ETH\_ILx2**

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  Steigende Flanke: FALSE->TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.  Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:  FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.  TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.



## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
Velocity	<p>Typ: INT</p> <p>(Wertebereich: -13200..13200) Initialwert: 0 )</p> <p>Wert für die Sollgeschwindigkeit der Bewegung [Min-1].</p>
Acceleration	<p>Typ: DINT</p> <p>(Wertebereich: 1..3000000) Initialwert: 600</p> <p>Wert für die Steilheit der Beschleunigungsrampe [1 Min-1/s]</p>
Deceleration	<p>Typ: DINT</p> <p>(Wertebereich: 200..3000000) Initialwert: 750</p> <p>Wert für die Steilheit der Verzögerungsrampe [1 Min-1/s]</p>
Axis	<p>Typ: AXIS_REF_EIP_ILx2</p> <p>(Wertebereich: &lt;Name der Achse&gt;) Initialwert: leer</p> <p>Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.</p>

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
InVelocity	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>FALSE: Zielgeschwindigkeit ist (noch) nicht erreicht. TRUE: Zielgeschwindigkeit ist erreicht.</p>
Busy	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.</p>
CommandAborted	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>TRUE: die Ausführung wurde durch einen anderen Baustein abgebrochen. FALSE: die Ausführung ist (noch) nicht abgebrochen worden.</p>

Variable	Mögliche Werte, Bedeutung
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

*Aufgabe des Bausteins*

Betriebsart Geschwindigkeitsprofil mit der Geschwindigkeit *Velocity* starten. Wenn die Zielgeschwindigkeit erreicht ist, wird *InVelocity* gesetzt.

*Hinweis*

Ändern der Geschwindigkeit *Velocity* auf „0“

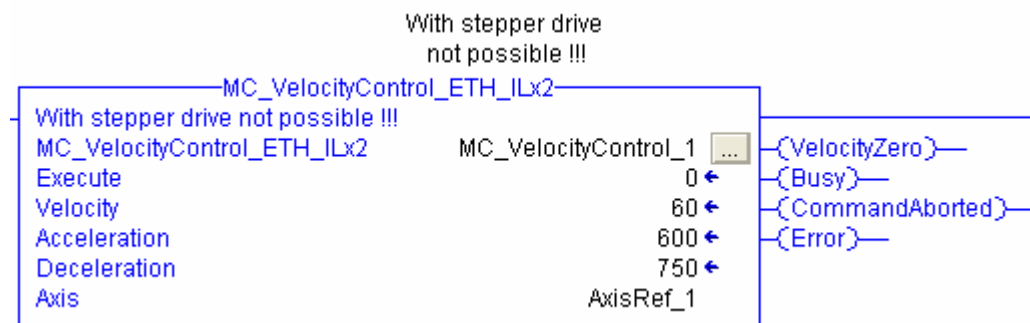
Hiermit kann der Antrieb jederzeit mit der normalen Verzögerung angehalten werden. Wenn die Geschwindigkeit auf „0“ gesetzt wird, dann wird der Antrieb nur temporär angehalten! Das heißt, sobald die Geschwindigkeit wieder auf einen Wert ungleich „0“ gesetzt wird, läuft der Antrieb sofort weiter.

Bereichsüberlauf der Position

In der Betriebsart Geschwindigkeitsprofil kann ein Bereichsüberlauf vorkommen. Dies ist für den Antrieb kein Fehler, die Betriebsart wird unverändert fortgesetzt. Ob der absolute Positionsbezug noch gegeben ist, kann am Ausgang *Referenced* des Bausteins *MC\_ReadStatus\_ETH\_ILx2* gelesen werden.

**3.5.11 MC\_VelocityControl\_ETH\_ILx2**

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	<p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>Steigende Flanke: FALSE-&gt;TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.</p> <p>Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:</p> <p>FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.</p> <p>TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.</p>
Velocity	<p>Typ: INT</p> <p>(Wertebereich: -13200..13200) Initialwert: 0 )</p> <p>Wert für die Sollgeschwindigkeit der Bewegung [Min-1].</p>
Acceleration	<p>Typ: DINT</p> <p>(Wertebereich: 1..3000000) Initialwert: 600</p> <p>Wert für die Steilheit der Beschleunigungsrampe [1 Min-1/s]</p>
Deceleration	<p>Typ: DINT</p> <p>(Wertebereich: 200..3000000) Initialwert: 750</p> <p>Wert für die Steilheit der Verzögerungsrampe [1 Min-1/s]</p>
Axis	<p>Typ: AXIS_REF_EIP_ILx2</p> <p>(Wertebereich: &lt;Name der Achse&gt;) Initialwert: leer</p> <p>Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.</p>

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
VelocityZero	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>FALSE: Aktuelle Geschwindigkeit ist ungleich 0 TRUE: Aktuelle Geschwindigkeit ist 0. Antrieb</p>

Variable	Mögliche Werte, Bedeutung steht.
Busy	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
CommandAborted	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung wurde durch einen anderen Baustein abgebrochen. FALSE: die Ausführung ist (noch) nicht abgebrochen worden.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

*Aufgabe des Bausteins*

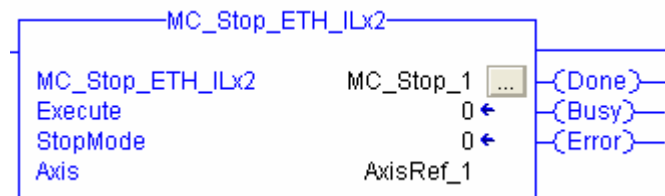
Betriebsart Drehzahlregelung mit der Geschwindigkeit *Velocity* starten. Wenn der Antrieb steht, wird *VelocityZero* gesetzt.

*Hinweis*

Diese Betriebsart wird nur von den Antrieben ILA2 und ILE2 unterstützt und ist im Schrittmotorantrieb ILS2 nicht implementiert.

**3.5.12 MC\_Stop\_ETH\_ILx2**

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL (Wertebereich: FALSE, TRUE)

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	<p>Initialwert: FALSE</p> <p>Steigende Flanke: FALSE-&gt;TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.</p> <p>Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:</p> <p>FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.</p> <p>TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.</p>
StopMode	<p>Typ: DINT</p> <p>(Wertebereich: 0..1)</p> <p>Initialwert: 0</p> <p>0 = HALT 1 = QuickStop</p> <p>HALT / QuickStop Konfiguration Siehe Produkthandbuch.</p>
Axis	<p>Typ: AXIS_REF_EIP_ILx2</p> <p>(Wertebereich: &lt;Name der Achse&gt;)</p> <p>Initialwert: leer</p> <p>Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.</p>

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Done	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE)</p> <p>Initialwert: FALSE</p> <p>TRUE: die Ausführung ist beendet.</p> <p>FALSE: die Ausführung ist (noch) nicht beendet.</p>
Busy	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE)</p> <p>Initialwert: FALSE</p> <p>TRUE: der Baustein wird ausgeführt.</p> <p>FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.</p>
Error	<p>Typ: BOOL</p> <p>(Wertebereich: TRUE; FALSE)</p>

Variable	Mögliche Werte, Bedeutung
	Initialwert: FALSE
	TRUE: die Ausführung wurde mit Fehler beendet.
	FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

*Aufgabe des Bausteins*

Stoppen des Antriebs mit Momentenrampe

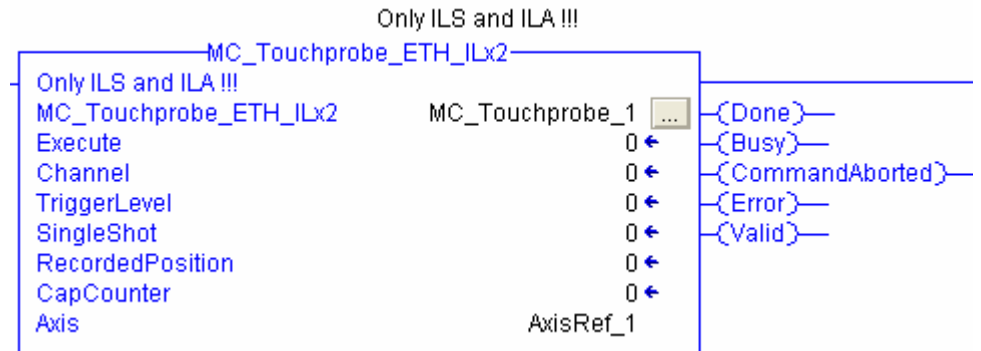
*Hinweis*

Diese Funktion kann durch andere Bewegungsbausteine nicht abgebrochen werden. Solange Execute=TRUE ist, kann kein anderer Bewegungsbaustein gestartet werden. Der Antrieb bleibt auch nach dem Stillstand blockiert.

Der Baustein bremst den Motor mit einer Momentenrampe. Der Parameter LIM\_I\_maxHalt (siehe Produkthandbuch) spezifiziert den Strom für die Momentenrampe. Nach Antriebsstillstand erfolgt ein interner Positionsabgleich, die Lageregelung wird aktiviert und der Motor wird bei aktiver Endstufe gehalten.

3.5.13 MC\_TouchProbe\_ETH\_ILx2

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE
	Steigende Flanke: FALSE->TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.
	Nach Beendigung der Ausführung bestimmt

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	<p>Execute das Verhalten der Ausgänge:</p> <p>FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.</p> <p>TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.</p>
Channel	<p>Typ: UINT</p> <p>(Wertebereich: 1..2)</p> <p>Initialwert 1</p> <p>Kanalnummer: Auswahl des Kanals, auf den sich die übrigen Parameter beziehen.</p>
TriggerLevel	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE)</p> <p>Initialwert: FALSE</p> <p>Auslösende Signalflanke:</p> <p>FALSE: fallende Flanke</p> <p>TRUE: steigende Flanke</p>
SingleShot	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE)</p> <p>Initialwert: TRUE</p> <p>FALSE: bei wiederholtem Eintreten des auslösenden Ereignisses wird die erfasste Position durch neue überschrieben.</p> <p>TRUE: die Positionserfassung wird nach Eintreten des auslösenden Ereignisses ausgeschaltet, sodass die erfasste Position nicht überschrieben werden kann.</p>
Axis	<p>Typ: AXIS_REF_EIP_ILx2</p> <p>(Wertebereich: &lt;Name der Achse&gt;)</p> <p>Initialwert: leer</p> <p>Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.</p>

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Done	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE)</p> <p>Initialwert: FALSE</p> <p>TRUE: die Ausführung ist beendet.</p> <p>FALSE: die Ausführung ist (noch) nicht beendet.</p>
Busy	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE)</p>

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
	Initialwert: FALSE  TRUE: der Baustein wird ausgeführt.  FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
CommandAborted	Typ BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung wurde durch einen anderen Baustein abgebrochen.  FALSE: die Ausführung ist (noch) nicht abgebrochen worden.
Error	Typ: BOOL  (Wertebereich: TRUE; FALSE) Initialwert: FALSE  TRUE: die Ausführung wurde mit Fehler beendet.  FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.
Valid	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung ist beendet. Ein gelesener Wert am Parameterausgang RecordedPosition ist gültig.  FALSE: die Ausführung ist (noch) nicht beendet. Ein Wert am Parameterausgang RecordedPosition ist (noch) nicht gültig.
RecordedPosition	Typ DINT  (Wertebereich: -2147483648..2147483647) Initialwert: 0  bei Eintreten des auslösenden Ereignisses erfasste Motorposition

*Aufgabe des Bausteins*

Positionserfassung einstellen und starten.

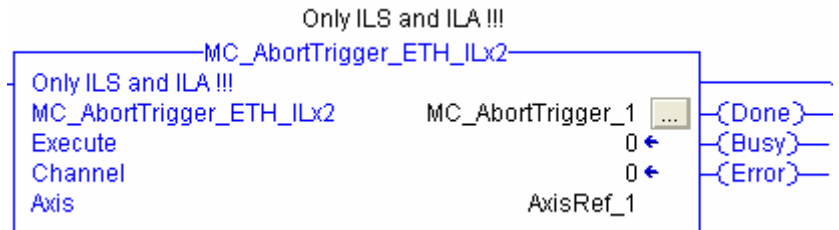
*Hinweis*

Nur die Antriebe Lexium ILA2K und ILS2K besitzen die Funktion der schnellen Positionserfassung.



## 3.5.14 MC\_AbortTrigger\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Execute	<p>Typ BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>Steigende Flanke: FALSE-&gt;TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.</p> <p>Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:</p> <p>FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.</p> <p>TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.</p>
Channel	<p>Typ: UINT</p> <p>(Wertebereich: 1..2) Initialwert 1</p> <p>1: Positionserfassung auf Kanal 1 (CAP1) abbrechen.</p> <p>2: Positionserfassung auf Kanal 2 (CAP2) abbrechen.</p>
Axis	<p>Typ: AXIS_REF_EIP_ILx2</p> <p>(Wertebereich: &lt;Name der Achse&gt;) Initialwert: leer</p> <p>Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.</p>

*Ausgangsvariablen*

Variable	Mögliche Werte, Bedeutung
Done	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

*Aufgabe des Bausteins*

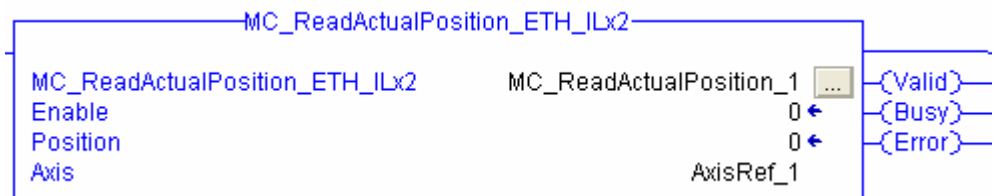
Abbrechen einer Positionserfassung.

*Hinweis*

Nur die Antriebe Lexium ILA2K und ILS2K besitzen die Funktion der schnellen Positionserfassung.

**3.5.15 MC\_ReadActualPosition\_ETH\_ILx2**

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Enable	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE

## PLCopen AddOnInstructions ILx2 EthernetIP

---

Variable	Mögliche Werte, Bedeutung
	pegelsensitiv; startet bzw. stoppt die Ausführung des Bausteins  TRUE: der Baustein wird wiederholt ausgeführt.  FALSE: die Ausführung wird sofort beendet; die Steuerausgänge Valid, Busy und Error werden sofort FALSE
Axis	Typ: AXIS_REF_EIP_ILx2  (Wertebereich: <Name der Achse>) Initialwert: leer  Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

### Ausgangsvariablen

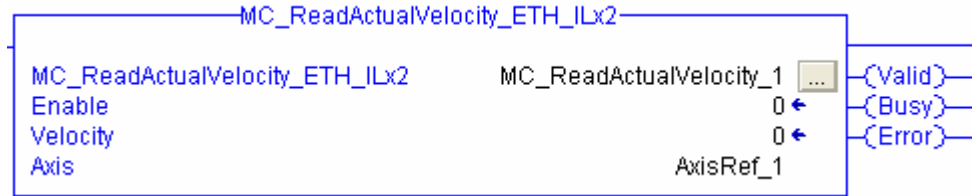
Variable	Mögliche Werte, Bedeutung
Valid	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung ist beendet.  FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Baustein wird ausgeführt.  FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL  (Wertebereich: TRUE; FALSE) Initialwert: FALSE  TRUE: die Ausführung wurde mit Fehler beendet.  FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.
Position	Typ: DINT  (Wertebereich:)  Istposition des Motors in Anwendereinheiten

### Aufgabe des Bausteins

Lesen der Istposition des Motors in Anwendereinheiten

3.5.16 MC\_ReadActualVelocity\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Enable	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE pegelsensitiv; startet bzw. stoppt die Ausführung des Bausteins TRUE: der Baustein wird wiederholt ausgeführt. FALSE: die Ausführung wird sofort beendet; die Steuerausgänge Valid, Busy und Error werden sofort FALSE
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Valid	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL

## PLCopen AddOnInstructions ILx2 EthernetIP

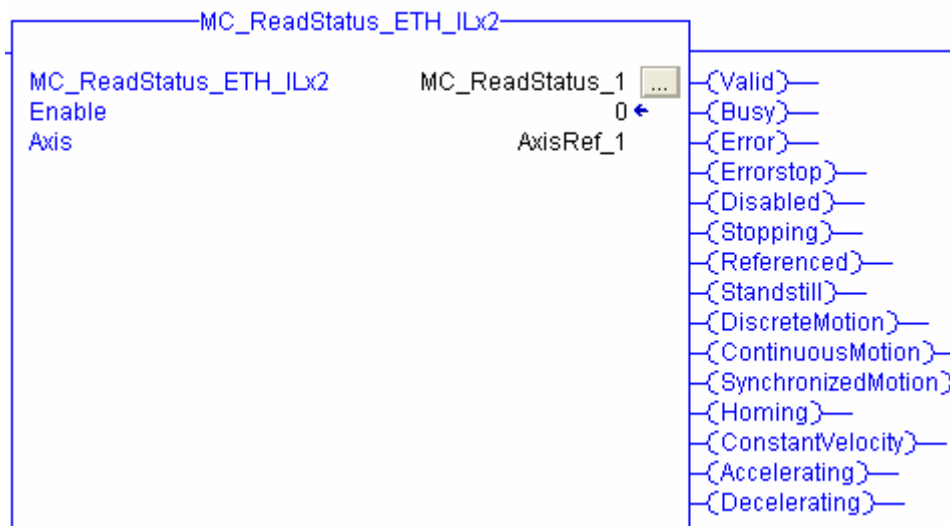
Variable	Mögliche Werte, Bedeutung
	(Wertebereich: TRUE; FALSE) Initialwert: FALSE  TRUE: die Ausführung wurde mit Fehler beendet.  FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.
Velocity	Typ: INT  (Wertebereich: -13200..+13200)  Aktuelle Motordrehzahl [Min-1]

*Aufgabe des Bausteins*

Lesen der aktuellen Motordrehzahl [Min-1]

### 3.5.17 MC\_ReadStatus\_ETH\_ILx2

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Enable	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  pegelsensitiv; startet bzw. stoppt die Ausführung des Bausteins  TRUE: der Baustein wird wiederholt ausgeführt.  FALSE: die Ausführung wird sofort beendet; die Steuerausgänge Valid, Busy und Error werden sofort FALSE

Variable	Mögliche Werte, Bedeutung
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

*Ausgangsvariablen*

Variable	Mögliche Werte, Bedeutung
Valid	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.
ErrorStop	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Bewegung wurde wegen eines Fehlers gestoppt
Disabled	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: Motorstrom ist ausgeschaltet. FALSE: Motorstrom ist eingeschaltet
Stopping	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein MC_Stop_ETH_ILx2 wird ausgeführt; die Bewegung wird gerade

## PLCopen AddOnInstructions ILx2 EthernetIP

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
	gestoppt.
Referenced	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Antrieb ist referenziert; der Positionsbezug zur Mechanik ist bekannt.
StandStill	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Antrieb steht
DiscreteMotion	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Antrieb befindet sich in der Punkt-zu-Punkt-Betriebsart.
ContinuousMotion	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Antrieb befindet sich in einer Betriebsart mit kontinuierlicher Bewegung, dem Geschwindigkeitsprofil
SynchronizedMotion	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Antrieb befindet sich in einer Synchronisationsbetriebsart, wie z.B. Betriebsart elektronisches Getriebe
Homing	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE )  TRUE: der Antrieb befindet sich in der Referenzierungsbetriebsart.
ConstantVelocity	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Antrieb dreht aktuell mit konstanter Geschwindigkeit.
Accelerating	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Antrieb beschleunigt.
Decelerating	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Antrieb verzögert

Aufgabe des Bausteins

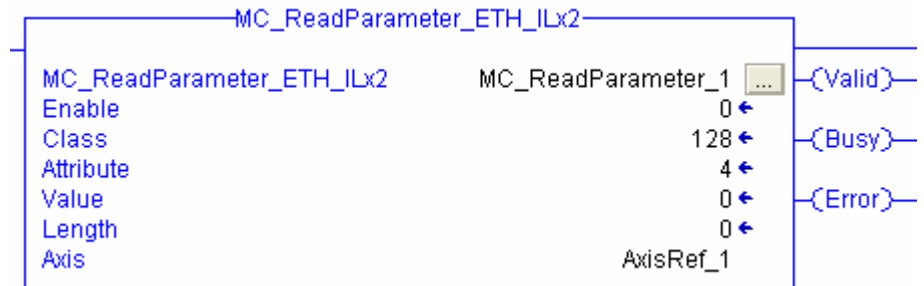
Der aktuelle Status des Antriebs wird ausgegeben.

Hinweis

Der Antrieb befindet sich zu jedem Zeitpunkt in genau einem der Zustände *StandStill*, *Homing*, *DiscreteMotion*, *ContinuousMotion*, *SynchronizedMotion*, *Stopping*, *Disabled* oder *ErrorStop*. Der gleichnamige Ausgang des Bausteins ist dann TRUE.

3.5.18 MC\_ReadParameter\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Enable	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE pegelsensitiv; startet bzw. stoppt die Ausführung des Bausteins TRUE: der Baustein wird wiederholt ausgeführt. FALSE: die Ausführung wird sofort beendet; die Steuerausgänge Valid, Busy und Error werden sofort FALSE
Class	Typ: UINT (Wertebereich: 0..65535) Klasse des zu lesenden Objektes; die Objekte sind mit ihrem Klasse, Instanz und Attribut im Produkthandbuch gelistet. Da der Antrieb nur mit Attribute = 1 arbeitet, ist dieser Wert intern fest auf den Wert 1 gesetzt.
Attribute	Typ: UINT; (Wertebereich: 0..255 ) Attribut des zu lesenden Objektes.
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>)



## PLCopen AddOnInstructions ILx2 EthernetIP

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
	Initialwert: leer
	Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

### *Ausgangsvariablen*

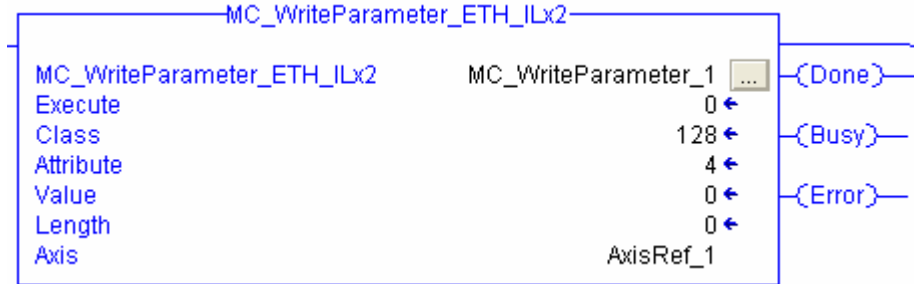
<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
Valid	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.
Value	Typ: DINT (Wertebereich: -2147483648..2147483647) Initialwert: 0 Wert des Geräteparameters.
Length	Typ: UINT (Wertebereich: 0..65535) Initialwert: 0 Länge des gelesenen Geräteparameters in Bytes

### *Aufgabe des Bausteins*

Lesen eines Objekts aus der Geräteparameterliste.

3.5.19 MC\_WriteParameter\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  Steigende Flanke: FALSE->TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.  Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:  FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.  TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.
ParameterNumber	Typ: INT (Wertebereich: 0..65535)  0: Parameter wird durch Index und Subindex ausgewählt.  >0: Nummer des Parameters, der geschrieben werden soll:  2: Position des positiven Softwareendschalters [usr]  3: Position des negativen Softwareendschalters [usr]  4: Freigabe (Bit0=1) bzw. Sperrung (Bit0=0) des positiven Softwareendschalters  5: Freigabe (Bit0=1) bzw. Sperrung (Bit0=0) des negativen Softwareendschalters  Andere Nummern werden nicht unterstützt.

## PLCopen AddOnInstructions ILx2 EthernetIP

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
Value	Typ: DINT (Wertebereich: -2147483648..2147483647) Initialwert: 0 Wert des Geräteparameters
Index	Typ: UINT (Wertebereich: 0..65535) Index des zu schreibenden Objektes; die Objekte sind mit ihrem Index und Subindex im Handbuch gelistet. Nur gültig bei ParameterNumber = 0.
Subindex	Typ: UINT (Wertebereich: 0..255) Subindex des zu schreibenden Objektes; die Objekte sind mit ihrem Index und Subindex im Handbuch gelistet. Nur gültig bei ParameterNumber = 0.
Length	Typ: UINT (Wertebereich: 0..65535) Initialwert: 0 Länge des zu schreibenden Geräteparameters in Bytes
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

### Ausgangsvariablen

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
Done	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE)

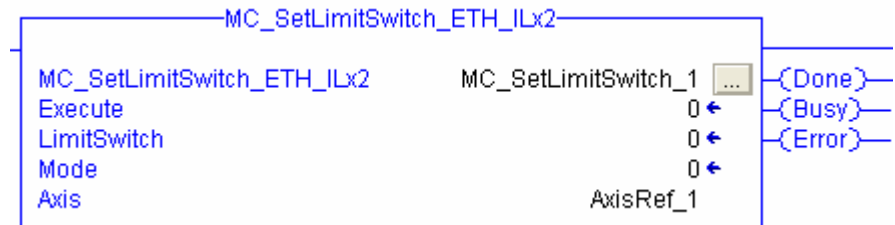
Variable	Mögliche Werte, Bedeutung
	Initialwert: FALSE
	TRUE: die Ausführung wurde mit Fehler beendet.
	FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

*Aufgabe des Bausteins*

Schreiben eines Objekts in der Geräteparameterliste.

**3.5.20 MC\_SetLimitSwitch\_ETH\_ILx2**

*Grafische Darstellung*



*Eingangsvariablen*

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE  Steigende Flanke: FALSE->TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit dem dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.  Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:  FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.  TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.
LimitSwitch	Typ INT (Wertebereich: 1..2) Initialwert: 1  1 = LimP

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	2= LimN
Mode	Typ INT  (Wertebereich: 0..2) Initialwert: 0  0 = inactiv 1 = normally closed 2 = normally open
Axis	Typ: AXIS_REF_EIP_ILx2  (Wertebereich: <Name der Achse>) Initialwert: leer  Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Done	Typ: BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: die Ausführung ist beendet. FALSE: die Ausführung ist (noch) nicht beendet.
Busy	Typ BOOL  (Wertebereich: FALSE, TRUE) Initialwert: FALSE  TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL  (Wertebereich: TRUE; FALSE) Initialwert: FALSE  TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

### Aufgabe des Bausteins

Endschalter Konfiguration.

Der positive Endschalter (LimP) und der negative Endschalter (LimN) werden mit diesem Baustein parametrieret.

Mode = 0 : Endschalter wird inaktiv geschaltet.

Mode = 1 : Endschalter wird als Öffner konfiguriert

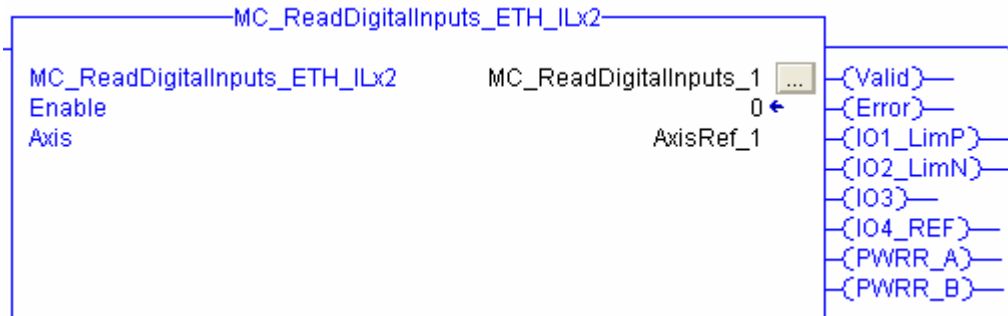
Mode = 2 : Endschalter wird als Schließer konfiguriert

Hinweis

Soll die Konfiguration nach Power Off erhalten bleiben, so muss anschließend der Befehl SaveEEPROM mit der Funktion „MC\_WriteParameter“ an den Antrieb gesendet werden.

3.5.21 MC\_ReadDigitalInputs\_ETH\_ILx2

Grafische Darstellung



Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Enable	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE pegelsensitiv; startet bzw. stoppt die Ausführung des Bausteins TRUE: der Baustein wird wiederholt ausgeführt. FALSE: die Ausführung wird sofort beendet; die Steuerausgänge Valid, Busy und Error werden sofort FALSE
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Valid	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. Der gelesene Wert am Parameterausgang Value ist gültig.

## PLCopen AddOnInstructions ILx2 EthernetIP

Variable	Mögliche Werte, Bedeutung
	FALSE: die Ausführung ist (noch) nicht beendet. Der Wert am Parameterausgang <i>Value</i> ist (noch) nicht gültig.
Busy	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: der Baustein wird ausgeführt. FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.
Error	Typ: BOOL (Wertebereich: TRUE; FALSE) Initialwert: FALSE TRUE: die Ausführung wurde mit Fehler beendet. FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.

### Aufgabe des Bausteins

Lesen des aktuellen Eingangsabbilds des Antriebs.

### Hinweis

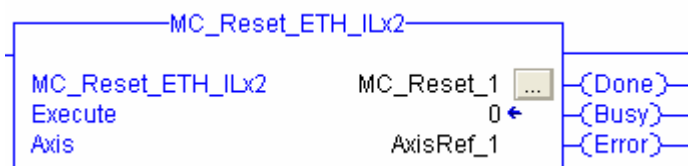
Die Bedeutung der Signale ist im Produkthandbuch beschrieben.

Der Eingang I0 (/REF) wird vom Antrieb nur während einer Referenzierung auf das Referenzsignal verwendet (siehe MC\_Home\_ETH\_ILx2). Wenn diese Funktion nicht benutzt wird, kann der Eingang für andere Zwecke frei verwendet werden.

Die Endschalterfunktion der Eingänge I1 (/LIMN) und I2 (/LIMP) kann abgeschaltet werden. Wenn die Endschalterfunktion nicht benutzt wird, können die Eingänge für andere Zwecke frei verwendet werden.

## 3.5.22 MC\_Reset\_ETH\_ILx2

### Grafische Darstellung



### Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Execute	Typ BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE

Variable	Mögliche Werte, Bedeutung
	<p>Steigende Flanke: FALSE-&gt;TRUE startet die Ausführung. Eine erneute steigende Flanke setzt die Ausführung mit den dann aktuellen Eingangsparametern fort. Die Ausführung ist beendet, sobald der Ausgang Busy = FALSE wird.</p> <p>Nach Beendigung der Ausführung bestimmt Execute das Verhalten der Ausgänge:</p> <p>FALSE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted für genau einen Aufruf TRUE.</p> <p>TRUE: gleichzeitig mit Busy = FALSE wird entweder Done, Error oder CommandAborted TRUE und bleibt TRUE, bis der Baustein mit Execute = FALSE aufgerufen wird.</p>
Axis	<p>Typ: AXIS_REF_EIP_ILx2</p> <p>(Wertebereich: &lt;Name der Achse&gt;) Initialwert: leer</p> <p>Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.</p>

*Ausgangsvariablen*

Variable	Mögliche Werte, Bedeutung
Done	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>TRUE: die Ausführung ist beendet.</p> <p>FALSE: die Ausführung ist (noch) nicht beendet.</p>
Busy	<p>Typ: BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>TRUE: der Baustein wird ausgeführt.</p> <p>FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.</p>
Error	<p>Typ: BOOL</p> <p>(Wertebereich: TRUE; FALSE) Initialwert: FALSE</p> <p>TRUE: die Ausführung wurde mit Fehler beendet.</p> <p>FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.</p>



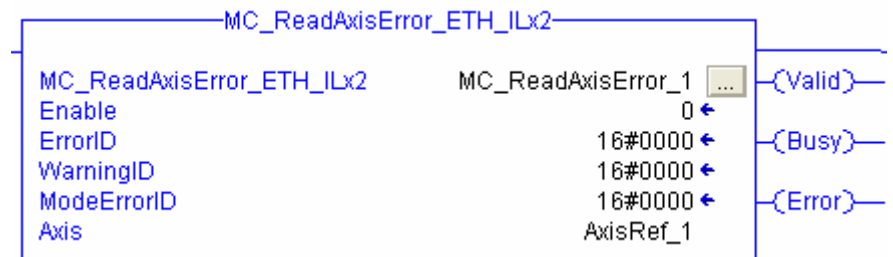
# PLCopen AddOnInstructions ILx2 EthernetIP

## Aufgabe des Bausteins

Fehlerquittierung. Die Fehlerzelle wird gelöscht und damit frei für zukünftige Fehlermeldungen. Wurde der Motor durch die automatische Fehlerreaktion gestoppt, dann wird er wieder freigegeben, wenn die Fehlerursache zum Zeitpunkt der Fehlerquittierung beseitigt ist.

### 3.5.23 MC\_ReadAxisError\_ETH\_ILx2

#### Grafische Darstellung



#### Eingangsvariablen

Variable	Mögliche Werte, Bedeutung
Enable	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE pegelsensitiv; startet bzw. stoppt die Ausführung des Bausteins TRUE: der Baustein wird wiederholt ausgeführt. FALSE: die Ausführung wird sofort beendet; die Steuerausgänge Valid, Busy und Error werden sofort FALSE
Axis	Typ: AXIS_REF_EIP_ILx2 (Wertebereich: <Name der Achse>) Initialwert: leer Name des Antriebes, für den der Baustein ausgeführt werden soll. Es muss für jeden Antrieb eine globale Datenstruktur angelegt werden. Diese Datenstruktur wird hier als Parameter übergeben.

#### Ausgangsvariablen

Variable	Mögliche Werte, Bedeutung
Valid	Typ: BOOL (Wertebereich: FALSE, TRUE) Initialwert: FALSE TRUE: die Ausführung ist beendet. Der

<b>Variable</b>	<b>Mögliche Werte, Bedeutung</b>
	<p>gelesene Wert am Parameterausgang <i>ErrorID</i> ist gültig.</p> <p>FALSE: die Ausführung ist (noch) nicht beendet. Der Wert am Parameterausgang <i>ErrorID</i> ist (noch) nicht gültig.</p>
Busy	<p>Typ BOOL</p> <p>(Wertebereich: FALSE, TRUE) Initialwert: FALSE</p> <p>TRUE: der Baustein wird ausgeführt.</p> <p>FALSE: die Ausführung ist beendet; der Baustein ist nicht aktiv.</p>
Error	<p>Typ: BOOL</p> <p>(Wertebereich:TRUE; FALSE) Initialwert: FALSE</p> <p>TRUE: die Ausführung wurde mit Fehler beendet.</p> <p>FALSE: bei der Ausführung ist (noch) kein Fehler aufgetreten.</p>
ErrorID	<p>Typ: WORD</p> <p>(Wertebereich: 0000<sub>h</sub> ... FFFF<sub>h</sub>) Initialwert: 0000<sub>h</sub></p> <p>0: kein Fehler in der Fehlerzelle</p> <p>&gt;0: Fehlernummer (siehe Liste der Fehlernummern im Anhang).</p>

*Aufgabe des Bausteins*

Fehlerabfrage eines Geräts

## 4 Kommunikation überwachen

In diesem Kapitel finden Sie einen Vorschlag zur Überwachung der Kommunikation zum Antrieb in der Applikation.

In dem Projekt „SE\_Motion\_ETH\_ILx2.ACD“ finden Sie die Beispielroutine „CheckIOState\_SingleAxis“, in der die Kommunikation zum Antrieb überwacht wird.



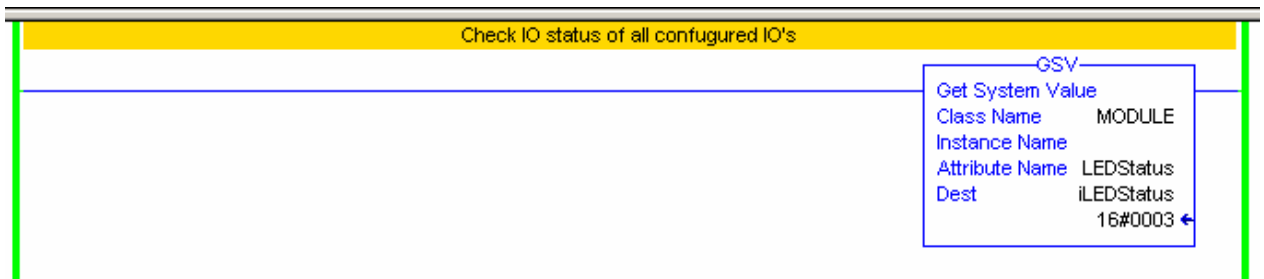
### 4.1 I/O LED Status auslesen

Mit dieser Routine lässt sich der Status der I/O LED auslesen. Hiermit kann die komplette I/O-Konfiguration überwacht werden.

- Definieren Sie eine Variable vom Typ INT.

+ iLEDStatus	16#0003	Hex	INT
--------------	---------	-----	-----

- Rufen Sie die Routine GSV (Get System Value) wie folgt auf.



- Nach dem Aufruf der Routine kann der Status der I/O LED ausgewertet werden:  
iLEDStatus:  
0 I/O LED ist AUS  
Es sind keine I/O Verbindungen konfiguriert.  
  
1 I/O LED blinkt rot  
Keine der konfigurierten I/O Verbindungen arbeitet korrekt.  
  
2 I/O LED blinkt grün  
Ein Teil der konfigurierten I/O Verbindungen arbeitet korrekt.  
Mindestens eine I/O Verbindung arbeitet nicht korrekt.  
  
3 I/O LED leuchtet statisch grün  
Alle I/O Verbindungen arbeiten ohne Fehler.

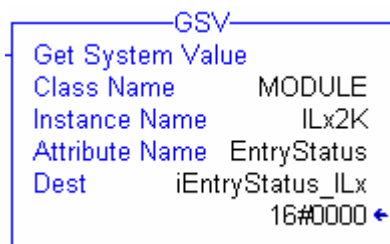
**4.2 Einzelne I/O Verbindungen überwachen**

Mit dieser Routine lässt sich der Status einzelner I/O Verbindungen auslesen.  
 Hiermit kann die EtherNet/IP Kommunikation jedes einzelnen Antriebes überwacht werden.

- Definieren Sie eine Variable vom Typ INT.



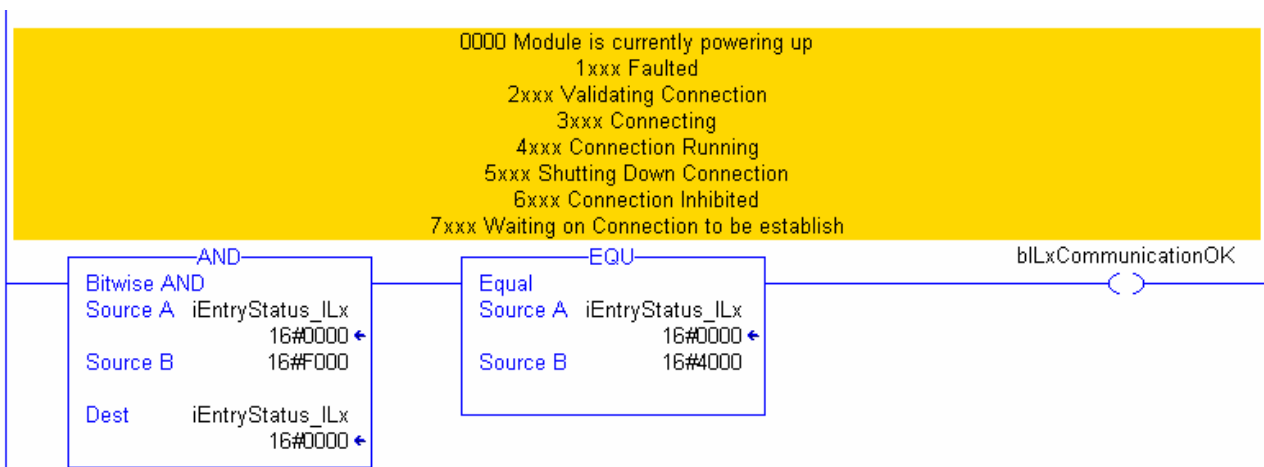
- Rufen Sie die Routine GSV (Get System Value) wie folgt auf. Der Eingang „Instance Name“ muss dem Antriebsnamen aus der I/O Konfiguration entsprechen.



- Gerätestatus ermitteln:

- 0000 Module is currently powering up
- 1xxx Fehler
- 2xxx Validating Connection
- 3xxx Connecting
- 4xxx Connection Running
- 5xxx Shutting Down Connection
- 6xxx Connection Inhibited
- 7xxx Waiting on Connection to be establish

- Nachfolgend ein Beispiel zur Auswertung "Kommunikation OK".



## 5 Anhang

### 5.1 Fehlernummern

Die Fehlernummern sind der Rückgabewert des Funktionsbausteines MC\_ReadAxisError\_ETH\_ILx2.

ErrorID hex.	ErrorID dec.	Fehler- klasse	Beschreibung
Bis 00FF <sub>h</sub>	Bis 255		siehe CiA405 Fehlermeldungen
<b>Antriebsfehlermeldungen:</b>			
1100 <sub>h</sub>	4352	0	Parameter außerhalb zulässigem Wertebereich
1101 <sub>h</sub>	4353	0	Parameter existiert nicht (Index)
1102 <sub>h</sub>	4354	0	Parameter existiert nicht (Subindex)
1103 <sub>h</sub>	4355	0	Schreiben des Parameters nicht zulässig (READ-only)
1104 <sub>h</sub>	4356	0	Schreibzugriff verweigert (keine Zugriffsrechte)
1106 <sub>h</sub>	4358	0	Befehl nicht erlaubt, wenn Antrieb aktiv
1107 <sub>h</sub>	4359	0	Zugriff durch andere Schnittstelle verriegelt
1108 <sub>h</sub>	4360	0	Parameter nicht lesbar (Block Upload)
1109 <sub>h</sub>	4360	0	Powerfail Daten ungültig
110A <sub>h</sub>	4362	0	Kein Bootloader vorhanden
110B <sub>h</sub>	4363	3	Initialisierungsfehler
1300 <sub>h</sub>	4864	3	Sicherheitsfunktion Safe Torque OFF ausgelöst (STO_A und STO_B)
1301 <sub>h</sub>	4865	4	Eingänge für Sicherheitsfunktion STO_A und STO_B haben unterschiedliche Pegel
1310 <sub>h</sub>	4880	3	Frequenz der Führungssignale zu hoch
1603 <sub>h</sub>	5635	0	Aufzeichnungsspeicher durch andere Funktion belegt
1606 <sub>h</sub>	5638	0	Aufzeichnung ist noch aktiv
1607 <sub>h</sub>	5639	0	Kein Trigger Parameter für Aufzeichnung definiert
1608 <sub>h</sub>	5640	0	Triggeroption für Triggerparameter nicht zulässig
1609 <sub>h</sub>	5641	0	Kein Aufzeichnungs kanal definiert
160A <sub>h</sub>	5642	0	Keine Aufzeichnungsdaten vorhanden
160B <sub>h</sub>	5643	0	Parameter nicht aufzeichnenbar
160C <sub>h</sub>	5644	1	Autotuning: Trägheitsmoment außerhalb zulässigem Bereich
160E <sub>h</sub>	5646	1	Autotuning: Testfahrt konnte nicht gestartet werden
160F <sub>h</sub>	5647	1	Autotuning: Endstufe kann nicht aktiviert werden
1610 <sub>h</sub>	6548	1	Autotuning: Bearbeitung abgebrochen
1611 <sub>h</sub>	5649	1	Systemfehler: Autotuning interner Schreibzugriff
1612 <sub>h</sub>	5650	1	Systemfehler: Autotuning interner Lesezugriff
1613 <sub>h</sub>	5651	1	Autotuning: max. zulässiger Positionierbereich überschritten
1614 <sub>h</sub>	5652	0	Autotuning: bereits aktiv
1617 <sub>h</sub>	5655	1	Autotuning: Reib- bzw. Lastmoment zu groß
1618 <sub>h</sub>	5656	1	Autotuning: Optimierung fehlgeschlagen
1A00 <sub>h</sub>	6656	0	Systemfehler: FIFO Speicher Überlauf
1A01 <sub>h</sub>	6657	3	Motor wurde getauscht
1A02 <sub>h</sub>	6658	3	Motor wurde getauscht
1B00 <sub>h</sub>	6912	4	Systemfehler: Fehlerhafte Parameter in Motor oder Endstufe
1B01 <sub>h</sub>	6913	3	Anwenderparameter max. Drehzahl zu groß
1B02 <sub>h</sub>	6914	3	Anwenderparameter max. Strom, Haltestrom oder Quickstopstrom zu groß
2300 <sub>h</sub>	8960	3	Überstrom Endstufe
2301 <sub>h</sub>	8961	3	Überstrom Bremswiderstand

ErrorID hex.	ErrorID dec.	Fehler- klasse	Beschreibung
3100 <sub>h</sub>	12544	par.	Phasenfehler Netzversorgung
3200 <sub>h</sub>	12800	3	Überspannung DC-Bus
3201 <sub>h</sub>	12801	3	Unterspannung DC-Bus (Abschalt-Schwelle)
3202 <sub>h</sub>	12802	2	Unterspannung DC-Bus (Quickstop-Schwelle)
3203 <sub>h</sub>	12803	4	Versorgungsspannung Motor-Encoder
3206 <sub>h</sub>	12806	0	Unterspannung DC-Bus (Warnung)
4100 <sub>h</sub>	16640	3	Übertemperatur Endstufe
4101 <sub>h</sub>	16641	0	Warnung Übertemperatur Endstufe
4102 <sub>h</sub>	16642	0	Warnung Überlast (I <sup>2</sup> t) Endstufe
4200 <sub>h</sub>	16896	3	Übertemperatur Gerät
4300 <sub>h</sub>	17152	3	Übertemperatur Motor
4301 <sub>h</sub>	17153	0	Warnung Übertemperatur Motor
4302 <sub>h</sub>	17154	0	Warnung Überlast (I <sup>2</sup> t) Motor
4402 <sub>h</sub>	17410	0	Warnung Überlast (I <sup>2</sup> t) Bremswiderstand
5200 <sub>h</sub>	20992	3	keine Verbindung mit Motor-Encoder
5201 <sub>h</sub>	20993	4	fehlerhafte Kommunikation Motor-Encoder
5202 <sub>h</sub>	20994	4	Motor-Encoder wird nicht unterstützt
5203 <sub>h</sub>	20995	4	keine Verbindung mit Motor-Encoder
5204 <sub>h</sub>	20996	3	Verbindung mit Motor-Encoder verloren
5430 <sub>h</sub>	21552	0	Systemfehler: EEPROM Lesefehler
5431 <sub>h</sub>	21553	0	Systemfehler: EEPROM Schreibfehler
5435 <sub>h</sub>	21557	0	Systemfehler: EEPROM nicht formatiert
5437 <sub>h</sub>	21559	0	Systemfehler: EEPROM Prüfsummenfehler Herstellerdaten
5438 <sub>h</sub>	21560	0	Systemfehler: EEPROM Prüfsummenfehler Anwender-Parameter
5439 <sub>h</sub>	21561	0	Systemfehler: EEPROM Prüfsummenfehler CAN-Parameter
543A <sub>h</sub>	21562	0	Systemfehler: EEPROM HardwareInfo ungültig
543B <sub>h</sub>	21563	0	Systemfehler: EEPROM Herstellerdaten ungültig
543C <sub>h</sub>	21564	0	Systemfehler: EEPROM CAN-Daten ungültig
543D <sub>h</sub>	21565	0	Systemfehler: EEPROM Anwender-Parameter ungültig
5600 <sub>h</sub>	22016	3	Phasenfehler Motoranschluss
5601 <sub>h</sub>	22017	4	Unterbrechung bzw. fehlerhafte Motor-Encodersignale
5602 <sub>h</sub>	22018	4	Unterbrechung bzw. fehlerhafte Motor-Encodersignale
5603 <sub>h</sub>	22019	4	Kommutierungsfehler
6107 <sub>h</sub>	24839	0	Parameter außerhalb Wertebereich (Berechnungsfehler)
6108 <sub>h</sub>	24840	0	Funktion nicht verfügbar
610D <sub>h</sub>	24845	0	Fehler im Auswahlparameter
610F <sub>h</sub>	24847	4	Interne Zeitbasis ausgefallen (Timer 0)
7120 <sub>h</sub>	28960	4	Ungültige Motordaten
7121 <sub>h</sub>	28961	2	Systemfehler: fehlerhafte Kommunikation Motor-Encoder
7122 <sub>h</sub>	28962	4	Unzulässige Motordaten
7123 <sub>h</sub>	28963	4	Motorstrom Offset außerhalb zulässigem Bereich
7124 <sub>h</sub>	28964	4	Systemfehler: Motor-Encoder ist defekt
7200 <sub>h</sub>	29184	4	Systemfehler: Kalibrierung Analog/Digital Wandler
7201 <sub>h</sub>	29185	4	Systemfehler: Motor-Encoder Initialisierung (Quadrantenauswertung)
7327 <sub>h</sub>	29479	4	Systemfehler: Positionssensor nicht bereit
7328 <sub>h</sub>	29480	4	Motor-Encoder meldet: Positionserfassung fehlerhaft
7329 <sub>h</sub>	29481	0	Motor-Encoder meldet: Warnung
7330 <sub>h</sub>	29482	4	Systemfehler: Motor-Encoder (Hiperface)
7331 <sub>h</sub>	29483	4	Systemfehler: Motor-Encoder Initialisierung
7333 <sub>h</sub>	29485	4	Systemfehler: Abweichung bei Kalibrierung Analog/Digitalwandler
7334 <sub>h</sub>	29486	3	Systemfehler: Analog/Digitalwandler Offset zu groß
7335 <sub>h</sub>	29487	0	Kommunikation zum Motor-Encoder belegt

## PLCopen AddOnInstructions ILx2 EthernetIP

ErrorID hex.	ErrorID dec.	Fehler- klasse	Beschreibung
7336 <sub>h</sub>	29488	3	Offset bei Offsetabgleich Sincos zu groß
7337 <sub>h</sub>	29489	1	Schreiben des Offsets konnte nicht erfolgreich durchgeführt werden
7400 <sub>h</sub>	29696	0	Systemfehler: ungültiger Interrupt (XINT2)
7500 <sub>h</sub>	29952	0	Modbus: Overrun-Fehler
7501 <sub>h</sub>	29953	0	Modbus: Framing-Fehler
7502 <sub>h</sub>	29954	0	Modbus: Parity-Fehler
7503 <sub>h</sub>	29955	0	Modbus: Empfangsfehler
8110 <sub>h</sub>	33040	0	CANopen over EtherCAT: CAN Überlauf (Nachricht verloren)
8130 <sub>h</sub>	33072	2	CANopen over EtherCAT: Heartbeat oder Life Guard Fehler
8201 <sub>h</sub>	33281	0	CANopen over EtherCAT: RxPDO1 konnte nicht verarbeitet werden
8202 <sub>h</sub>	33282	0	CANopen over EtherCAT: RxPDO2 konnte nicht verarbeitet werden
8203 <sub>h</sub>	33283	0	CANopen over EtherCAT: RxPDO3 konnte nicht verarbeitet werden
8204 <sub>h</sub>	33284	0	CANopen over EtherCAT: RxPDO4 konnte nicht verarbeitet werden
8205 <sub>h</sub>	33285	0	CANopen over EtherCAT: TxPDO konnte nicht verarbeitet werden
8206 <sub>h</sub>	33286	0	CANopen over EtherCAT: Überlauf interne Queue Nachricht verloren
A060 <sub>h</sub>	41056	2	Berechnungsfehler beim elektronischen Getriebe
A061 <sub>h</sub>	41057	2	Führungsgrößenänderung beim elektronischen Getriebe zu groß
A300 <sub>h</sub>	41728	0	Momentenrampe mit HALT-Strom aktiv
A301 <sub>h</sub>	41729	0	Antrieb im Zustand ‚QuickStopActive‘
A302 <sub>h</sub>	41730	1	Unterbrechung durch LIMP
A303 <sub>h</sub>	41731	1	Unterbrechung durch LIMN
A304 <sub>h</sub>	41732	1	Unterbrechung durch REF
A305 <sub>h</sub>	41733	0	Aktivieren der Endstufe in aktuellem Betriebszustand der Zustandsmaschine nicht möglich
A306 <sub>h</sub>	41734	1	Unterbrechung durch Softwarestop durch Anwender
A307 <sub>h</sub>	41735	0	Unterbrechung durch internen Softwarestop
A308 <sub>h</sub>	41736	0	Antrieb im Zustand ‚FAULT‘
A309 <sub>h</sub>	41737	0	Antrieb nicht im Zustand ‚OperationEnable‘
A310 <sub>h</sub>	41744	0	Endstufe nicht aktiv
A312 <sub>h</sub>	41746	0	Profilgenerierung unterbrochen
A313 <sub>h</sub>	41747	0	Positionsüberlauf vorhanden (pos_over=1) hierdurch Referenzpunkt nicht mehr definiert (ref_ok=0)
A314 <sub>h</sub>	41748	0	keine Referenzposition
A315 <sub>h</sub>	41749	0	Referenzierung aktiv
A316 <sub>h</sub>	41750	0	Überlauf bei Beschleunigungsberechnung
A317 <sub>h</sub>	41751	0	Antrieb nicht im Stillstand
A318 <sub>h</sub>	41752	0	Betriebsart aktiv (x_end = 0)
A319 <sub>h</sub>	41753	1	Manual-/Autotuning: Bereichsüberschreitung Distanz
A31A <sub>h</sub>	41754	0	Manual-/Autotuning: zu hohe Amplitude/Offset eingestellt
A31B <sub>h</sub>	41755	0	HALT angefordert
A31C <sub>h</sub>	41756	0	Unzulässige Positionseinstellung bei Software-Endschalter
A31D <sub>h</sub>	41757	0	Bereichsüberschreitung Drehzahl (CTRL_n_max)
A31E <sub>h</sub>	41758	1	Unterbrechung durch pos. Software-Endschalter
A31F <sub>h</sub>	41759	1	Unterbrechung durch neg. Software-Endschalter
A320 <sub>h</sub>	41760	par.	Positions-Schleppfehler
A321 <sub>h</sub>	41761	0	RS422 Positions-Schnittstelle nicht als Eingang definiert

ErrorID hex.	ErrorID dec.	Fehler- klasse	Beschreibung
A324 <sub>h</sub>	41764	1	Fehler bei Referenzierung (Zusatzinfo = Detaillierte Fehlernummer)
A325 <sub>h</sub>	41765	1	Anzufahrender Endschalter nicht aktiviert
A326 <sub>h</sub>	41766	1	Schalter REF nicht gefunden zwischen LIMP und LIMN
A327 <sub>h</sub>	41767	1	Referenzfahrt auf REF ohne Richtungsumkehr, unzulässiger Endschalter LIM aktiviert“
A328 <sub>h</sub>	41768	1	Referenzfahrt auf REF ohne Richtungsumkehr, Überfahren von LIM oder REF nicht zulässig
A329 <sub>h</sub>	41769	1	Mehr als ein Signal LIMP, LIMN, REF aktiv
A32A <sub>h</sub>	41770	1	Ext. Überwachungssignal LIMP bei neg. Drehrichtung
A32B <sub>h</sub>	41771	1	Ext. Überwachungssignal LIMN bei pos. Drehrichtung
A32C <sub>h</sub>	41772	1	Referenzfahrtfehler bei REF (z.B. durch Prellen)
A32D <sub>h</sub>	41773	1	Referenzfahrtfehler bei LIMP (z.B. durch Prellen)
A32E <sub>h</sub>	41774	1	Referenzfahrtfehler bei LIMN (z.B. durch Prellen)
A32F <sub>h</sub>	41775	1	Indexpuls wurde nicht gefunden
A330 <sub>h</sub>	41776	0	Referenzfahrt auf Indexpuls nicht reproduzierbar. Indexpuls ist zu nahe an Schalter.
A331 <sub>h</sub>	41777	3	Keine Hochlauf-Betriebsart bei lokaler Steuerung ausgewählt
A332 <sub>h</sub>	41778	1	Fehler bei Manuellfahrt (Zusatzinfo = Detaillierte Fehlernummer)
A334 <sub>h</sub>	41780	2	Timeout bei Stillstandsfenster-Kontrolle
A335 <sub>h</sub>	41781	1	Bearbeitung nur in Feldbusbetrieb möglich
B100 <sub>h</sub>	45312	0	Modbus: unbekannter Dienst
B200 <sub>h</sub>	45568	0	Modbus: Protokollfehler
B201 <sub>h</sub>	45569	2	Modbus: Nodeguard Fehler
B202 <sub>h</sub>	45570	0	Modbus: Nodeguard Warnung
B203 <sub>h</sub>	45571	0	Modbus: Anzahl Monitorobjekte falsch
B204 <sub>h</sub>	45572	0	Modbus: Dienst zu lang
B600 <sub>h</sub>	46592		Ethernet: Initialization error
B601 <sub>h</sub>	46593		Ethernet: Realtime data error
B602 <sub>h</sub>	46594		Ethernet: Realtime data warning
B603 <sub>h</sub>	46595		Ethernet: Protocol-specific error
B604 <sub>h</sub>	46596		Ethernet: Protocol-specific warning
B605 <sub>h</sub>	46597		Ethernet: Unknown error
B606 <sub>h</sub>	46598		Ethernet: delayed parameter access to module
B607 <sub>h</sub>	46599		Ethernet: is currently processing another request
B608 <sub>h</sub>	46600		Ethernet: Realtime Hot-Reset
B609 <sub>h</sub>	46601		Ethernet: Realtime Hot-Stop
B60A <sub>h</sub>	46602		Ethernet: timeout in internal communication
B60B <sub>h</sub>	46603		Ethernet: Error in internal communication
<b>Bibliotheksfehlermeldungen:</b>			
FF10 <sub>h</sub>	65296	0	Datentyp und Parameterlänge stimmen nicht überein
FF22 <sub>h</sub>	65314	0	Versuch einen nicht unterbrechbaren Baustein zu unterbrechen (MC Power, MC Stop, MC Home, MC SetPosition)
FF23 <sub>h</sub>	65315	0	Trigger Funktion ist bereits aktiv
FF24 <sub>h</sub>	65316	0	Timeout Toggle Bit
FF27 <sub>h</sub>	65319	0	Achse ist nicht im Zustand StandStill
FF2A <sub>h</sub>	65322	0	Trigger Ereignis verloren
FF34 <sub>h</sub>	65332	0	Endstufe wechselt nicht in den Zustand Enabled
FF36 <sub>h</sub>	65334	0	Betriebsart wird vom Antrieb nicht unterstützt
FF37 <sub>h</sub>	65335	0	Endstufe ist nicht im Zustand Enabled
FF3B <sub>h</sub>	65339	0	Endstufe ist nicht im Zustand Disabled