



ComDrvS7 V6.2X

für LOGO![®], S7-1200[®], S7-300[®], S7-400[®], VIPA
100V/200V/300V/300S

Zusatz-Dokumentation

für Softwareentwickler

Stand: Mai 2012

MHJ-Software

Albert-Einstein-Str. 101 • 75015 Bretten • Tel: 07252-84696 oder 87890 • Fax: 78780

Dokumentation des ComDrvS7 V6.2X

MHJ-Software

Albert-Einstein-Str. 101 • 75015 Bretten • Tel: 07252-84696 oder 87890 • Fax: 78780

1 Erweiterte Funktion für ComDrvS7 V6.2	Seite 4
1.1 Die Funktion: MPI6_OpenTcplpExt	Seite 4
1.2 Die Funktion: MPI6_OpenTcplpExt_2	Seite 5
1.3 Die Funktion: MPI6_OpenTcplpExt_3	Seite 7
1.4 Die Funktion: MPI6_EXT_BackupPlc (Nur in Extended-Version)	Seite 9
1.5 Die Funktion: MPI6_EXT_RestorePlc (Nur in Extended-Version)	Seite 10
1.6 Die Funktion: MPI6_EXT_DeleteHardwareCfgInPlc (Nur in Extended-Version)	Seite 11
1.7 Die Funktion: MPI6_EXT_GetBlocksInPlc (Nur in Extended-Version)	Seite 12
1.8 Die Funktion: MPI6_EXT_DeleteBlockPlc (Nur in Extended-Version)	Seite 13
1.9 Die Funktion: MPI6_EXT_LoadBlocksFromPlc (Nur in Extended-Version)	Seite 14
1.10 Die Funktion: MPI6_EXT_CompressPlc (Nur in Extended-Version)	Seite 16

1 Erweiterte Funktion für ComDrvS7 V6.2

1.1 Die Funktion: MPI6_OpenTcplpExt

Kurzbeschreibung

Die Funktion **MPI6_OpenTcplpExt** muss zwingend aufgerufen werden, um mit einer CPU zum ersten Mal zu kommunizieren, sofern die Kommunikation über TCP/IP hin zu einem Ethernet-CP oder einer CPU mit integrierter Ethernet-Schnittstelle aufzubauen ist.

Im Gegensatz zur Funktion **MPI6_OpenTcplp** kann bei **MPI6_OpenTcplpExt** auch die Racknummer der CPU angegeben werden. Dies ist nur notwendig, wenn die CPU in einem Rack ungleich 0 platziert ist.

Die Funktion baut eine Verbindung zu einem Ethernet-CP mit der angegebenen IP-Adresse auf. Des Weiteren ist die Slot-Nummer und die Rack-Nummer der CPU zu übergeben. Bei S7-300-Systemen ist hierbei generell der Slot 2 und die Racknummer 0 anzugeben (in diesem Fall kann auch die Funktion **MPI6_OpenTcplp** verwendet werden).

Mit dieser Funktion wird auch eine Kommunikationsinstanz angelegt. Die "Kennung" dieser Instanz wird in der Variablen "Handle" geliefert. Diese Kennung muss den anderen Funktionen der DLL übergeben werden, damit der hier angegebene Kommunikationsweg (IP-Adresse) genutzt wird (entfällt bei .Net-Wrapper-Klasse).

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT*	Hier wird das Handle der neu gebildeten Kommunikationsinstanz zurückgeliefert (entfällt bei .Net-Wrapper-Klasse).
IPAddress	CHAR*	Übergabe der IP-Adresse des Ethernet-CPs oder der integrierten Ethernet-Schnittstelle der CPU, mit dem die Kommunikation ausgeführt werden soll. Die Adresse ist in der Form "172.16.130.84" anzugeben.
PlcSlotNr	INT	Angabe des Steckplatzes der CPU mit welcher kommuniziert werden soll. Bei S7-300-Systemen ist hierbei generell der Steckplatz 2 anzugeben.
PlcRackNr	INT	Angabe des Racks der CPU mit welcher kommuniziert werden soll. Bei S7-300-Systemen ist hierbei generell das Rack 0 anzugeben.
Error	WORD*	Liefert die Funktion den Wert '0' als Rückgabewert, so ist ein Fehler bei der Ausführung aufgetreten. In diesem Fall wird in dem Parameter Error ein Error-Wert geliefert.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.2 Die Funktion: **MPI6_OpenTcplpExt_2**

Kurzbeschreibung

Die Funktion **MPI6_OpenTcplpExt_2** muss zwingend aufgerufen werden, um mit einer CPU zum ersten Mal zu kommunizieren, sofern die Kommunikation über TCP/IP hin zu einem Ethernet-CP oder einer CPU mit integrierter Ethernet-Schnittstelle aufzubauen ist.

Im Gegensatz zur Funktion **MPI6_OpenTcplp** kann bei **MPI6_OpenTcplpExt_2** die Racknummer der CPU und die bei der Kommunikation zu verwendende Portadresse angegeben werden.

Die Funktion baut eine Verbindung zu einer CPU mit Ethernetschnittstelle oder Ethernet-CP mit der angegebenen IP-Adresse auf. Des Weiteren ist die Slot-Nummer und die Rack-Nummer der CPU zu übergeben. Bei S7-300-Systemen ist hierbei generell der Slot 2 und die Racknummer 0 anzugeben (in diesem Fall kann auch die Funktion **MPI6_OpenTcplp** verwendet werden).

Die Standard-Portadresse bei S7-CPU's ist 102. Dieser Wert muss nur in Ausnahmefällen geändert werden. Dies kann z.B. der Fall sein, wenn über eine WAN-IP-Adresse mehrere CPU's erreicht werden sollen und die Unterscheidung durch die Portadresse erfolgt. In diesem Fall muss über Virtual-Router die externe Portadresse in die interne Portadresse 102 und der IP-Adresse der CPU (des Ethernet-CP's) gewandelt werden.

Eine weitere Ausnahme stellen S7-SoftSPS'en dar, welche in der Porteinstellung variabel sind (z.B. WinPLC-Engine).

Mit der Funktion **MPI6_OpenTcplpExt_2** wird auch eine Kommunikationsinstanz angelegt. Die "Kennung" dieser Instanz wird in der Variablen "Handle" geliefert. Diese Kennung muss den anderen Funktionen der DLL übergeben werden, damit der hier angegebene Kommunikationsweg (IP-Adresse) genutzt wird (entfällt bei .Net-Wrapper-Klasse).

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT*	Hier wird das Handle der neu gebildeten Kommunikationsinstanz zurückgeliefert (entfällt bei .Net-Wrapper-Klasse).
IPAddress	CHAR*	Übergabe der IP-Adresse des Ethernet-CPs oder der integrierten Ethernet-Schnittstelle der CPU, mit dem die Kommunikation ausgeführt werden soll. Die Adresse ist in der Form "172.16.130.84" anzugeben.
PlcSlotNr	INT	Angabe des Steckplatzes der CPU mit welcher kommuniziert werden soll. Bei S7-300-Systemen ist hierbei generell der Steckplatz 2 anzugeben.
PlcRackNr	INT	Angabe des Racks der CPU mit welcher kommuniziert werden soll. Bei S7-300-Systemen ist hierbei generell das Rack 0 anzugeben.
PortAddress	WORD	Angabe der Portadresse. Die S7-Kommunikation läuft immer über die Portadresse 102. Die Portadresse muss z.B. bei einer Fernwartung geändert werden, wenn mehrere CPUs über eine WAN-Adresse anzusprechen sind. Dabei muss aber die Portadresse in eine interne Portadresse 102 gewandelt werden, da die S7-CPU's nur an dieser mithören. Ausnahmen stellen CPUs dar, welche in der Portadressen-Einstellung variabel sind, z.B. S7-SoftSPSen.
Error	WORD*	Liefert die Funktion den Wert '0' als Rückgabewert, so ist ein Fehler bei der Ausführung aufgetreten. In diesem Fall wird in dem Parameter Error ein Error-Wert geliefert.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.3 Die Funktion: MPI6_OpenTcplpExt_3

Kurzbeschreibung

Die Funktion **MPI6_OpenTcplpExt_3** muss zwingend aufgerufen werden, um mit einer CPU zum ersten Mal zu kommunizieren, sofern die Kommunikation über TCP/IP hin zu einem Ethernet-CP oder einer CPU mit integrierter Ethernet-Schnittstelle aufzubauen ist.

Im Gegensatz zu den anderen **MPI6_OpenTcplp-Funktionen** kann bei **MPI6_OpenTcplpExt_3** die Racknummer der CPU, die bei der Kommunikation zu verwendende Portadresse und die IP-Adresse der im PC zu verwendenden Netzwerkkarte angegeben werden. Dies ist insbesondere dann wichtig, wenn **mehrere Netzwerkkarten im PC vorhanden sind**. Wird die **Verbindung zur CPU über VPN** vorgenommen, so muss ebenso diese Funktion verwendet werden. In diesem Fall ist die IP-Adresse des virtuellen VPN-Adapters am **Parameter "IPAddressNetworkAdapter"** anzugeben.

Die Funktion baut eine Verbindung zu einer CPU mit Ethernetschnittstelle oder Ethernet-CP mit der am Parameter "IPAddress" angegebenen IP-Adresse auf. Des Weiteren ist die Slot-Nummer und die Rack-Nummer der CPU zu übergeben. Bei S7-300-Systemen ist hierbei generell der Slot 2 und die Racknummer 0 anzugeben.

Die Standard-Portadresse bei S7-CPU's ist 102. Dieser Wert muss nur in Ausnahmefällen geändert werden. Dies kann z.B. der Fall sein, wenn über eine WAN-IP-Adresse mehrere CPU's erreicht werden sollen und die Unterscheidung durch die Portadresse erfolgt. In diesem Fall muss über Virtual-Router die externe Portadresse in die interne Portadresse 102 und der IP-Adresse der CPU (des Ethernet-CP's) gewandelt werden.

Eine weitere Ausnahme stellen S7-SoftSPS'en dar, welche in der Porteeinstellung variabel sind (z.B. WinPLC-Engine).

Mit der Funktion MPI6_OpenTcplpExt_3 wird auch eine Kommunikationsinstanz angelegt. Die "Kennung" dieser Instanz wird in der Variablen "Handle" geliefert. Diese Kennung muss den anderen Funktionen der DLL übergeben werden, damit der hier angegebene Kommunikationsweg (IP-Adresse) genutzt wird (entfällt bei .Net-Wrapper-Klasse).

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT*	Hier wird das Handle der neu gebildeten Kommunikationsinstanz zurückgeliefert (entfällt bei .Net-Wrapper-Klasse).
IPAddress	CHAR*	Übergabe der IP-Adresse des Ethernet-CPs oder der integrierten Ethernet-Schnittstelle der CPU, mit dem die Kommunikation ausgeführt werden soll. Die Adresse ist in der Form "172.16.130.84" anzugeben.
PlcSlotNr	INT	Angabe des Steckplatzes der CPU mit welcher kommuniziert werden soll. Bei S7-300-Systemen ist hierbei generell der Steckplatz 2 anzugeben.
PlcRackNr	INT	Angabe des Racks der CPU mit welcher kommuniziert werden soll. Bei S7-300-Systemen ist hierbei generell das Rack 0 anzugeben.
PortAddress	WORD	Angabe der Portadresse. Die S7-Kommunikation läuft immer über die Portadresse 102. Die Portadresse muss z.B. bei einer Fernwartung geändert werden, wenn mehrere CPUs über eine WAN-Adresse anzusprechen sind. Dabei muss aber die Portadresse in eine interne Portadresse 102 gewandelt werden, da die S7-CPU's nur an dieser mithören. Ausnahmen stellen CPU's dar, welche in der Portadressen-Einstellung variabel sind, z.B. S7-SoftSPSen.
IPAddressNetworkAdapter	CHAR*	Übergabe der IP-Adresse der im PC für die Kommunikation zur CPU zu verwendenden Netzwerkkarte oder des virtuellen VPN-Adapters bei Verwendung einer VPN-Verbindung.
Error	WORD*	Liefert die Funktion den Wert '0' als Rückgabewert, so ist ein Fehler bei der Ausführung aufgetreten. In diesem Fall wird in dem Parameter Error ein Error-Wert geliefert.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.4 Die Funktion: MPI6_EXT_BackupPlc (Nur in Extended-Version)

Voraussetzung zum Ausführen der Funktion

Eine der Einleitungsfunktionen (z.B. MPI6_OpenTcplp usw.) muss erfolgreich ausgeführt worden sein.

Des Weiteren muss die Funktion MPI6_ConnectToPLC oder MPI6_ConnectToPLCRouting ebenfalls erfolgreich aufgerufen worden sein.

Kurzbeschreibung

Mit dieser Funktion können alle in der CPU vorhandenen Bausteine des Typs OB, FC, FB, DB und SDB geladen und in eine noch nicht vorhandene WLD-Datei gespeichert werden.

Familie S7-1200® und LOGO!®

Die Funktion kann nicht bei bei CPUs der Reihe S7-1200® und LOGO!® verwendet werden.

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT	Das Handle der Kommunikationsinstanz welche angesprochen wird. (Entfällt bei .Net-Wrapper-Klasse)
WldFilePath	CHAR*	Angabe der WLD-Datei mit Pfad. Die Datei darf noch nicht vorhanden sein.
WarningsArray	WORD*	Treten bei der Aktion Warnungen auf, die nicht zum Abbruch der Aktion führen, dann werden diese in dem Array gesammelt.
WarningsNumber	WORD*	Anzahl der im Array "WarningsArray" vorhandenen Warnungen.
FatalError	WORD*	Ist ein schwerwiegender Fehler vorhanden, dann liefert die Funktion den Wert 0. In diesem Fall ist in FatalError der aufgetretene Fehler verzeichnet.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.5 Die Funktion: MPI6_EXT_RestorePlc (Nur in Extended-Version)

Voraussetzung zum Ausführen der Funktion

Eine der Einleitungsfunktionen (z.B. MPI6_OpenTcpIp usw.) muss erfolgreich ausgeführt worden sein.

Des Weiteren muss die Funktion MPI6_ConnectToPLC oder MPI6_ConnectToPLCRouting ebenfalls erfolgreich aufgerufen worden sein.

Kurzbeschreibung

Mit dieser Funktion können alle in einer WLD-Datei vorhandenen Bausteine des Typs OB, FC, FB, DB und SDB in eine CPU übertragen werden. Es ist darauf zu achten, dass insbesondere die in den SDBs vorhandene Hardwarekonfiguration zur CPU passt. Des Weiteren müssen die in den Bausteinen vorhandenen Operanden sowie die Bausteinnummern in der CPU zulässig sein. Anderenfalls tritt beim Übertragen bzw. beim späteren Betrieb der CPU ein Fehler auf.

Wichtig:

Die CPU muss sich bei der Übertragung im Zustand STOP befinden!

Familie S7-1200® und LOGO!®

Die Funktion kann nicht bei bei CPUs der Reihe S7-1200® und LOGO!® verwendet werden.

Hinweis zu Safety-CPU's und Safety-Bausteinen

Bei Safety-CPU's kann ein mit der Safety-Bibliothek erstelltes SPS-Programm nicht über die RestorePlc-Funktion in die CPU übertragen werden. Sind Safety-Bausteine in der WLD-Datei vorhanden, so wird der Vorgang ab dem ersten Safety-Baustein abgebrochen.

Nicht-Safety-Bausteine können in eine Safety-CPU übertragen werden.

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT	Das Handle der Kommunikationsinstanz welche angesprochen wird. (Entfällt bei .Net-Wrapper-Klasse)
WldFilePath	CHAR*	Angabe der WLD-Datei mit Pfad.
ErrorBstStr	CHAR*	Tritt beim Übertragen eines Bausteins ein Fehler auf, dann wird hier der Baustein angegeben, bei dem der Fehler auftrat.
CountBlocksSend	WORD*	Anzahl der übertragenen Bausteine.
Error	WORD*	Ist ein schwerwiegender Fehler vorhanden, dann liefert die Funktion den Wert 0. In diesem Fall ist in Error der aufgetretene Fehler verzeichnet.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.6 Die Funktion: MPI6_EXT_DeleteHardwareCfgInPlc (Nur in Extended-Version)

Voraussetzung zum Ausführen der Funktion

Eine der Einleitungsfunktionen (z.B. MPI6_OpenTcplp usw.) muss erfolgreich ausgeführt worden sein.

Des Weiteren muss die Funktion MPI6_ConnectToPLC oder MPI6_ConnectToPLCRouting ebenfalls erfolgreich aufgerufen worden sein.

Kurzbeschreibung

Mit dieser Funktion kann die Hardwarekonfiguration in einer CPU gelöscht werden. Es ist zu beachten, dass eine solche CPU unter Umständen nur noch über die MPI-Schnittstelle zu erreichen ist.

Wichtig:

Die CPU muss sich im Zustand STOP befinden!

Familie S7-1200® und LOGO!®

Die Funktion kann nicht bei bei CPUs der Reihe S7-1200® und LOGO!® verwendet werden.

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT	Das Handle der Kommunikationsinstanz welche angesprochen wird. (Entfällt bei .Net-Wrapper-Klasse)
Error	WORD*	Ist ein schwerwiegender Fehler vorhanden, dann liefert die Funktion den Wert 0. In diesem Fall ist in Error der aufgetretene Fehler verzeichnet.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.7 Die Funktion: MPI6_EXT_GetBlocksInPlc (Nur in Extended-Version)

Voraussetzung zum Ausführen der Funktion

Eine der Einleitungsfunktionen (z.B. MPI6_OpenTcpIp usw.) muss erfolgreich ausgeführt worden sein.

Des Weiteren muss die Funktion MPI6_ConnectToPLC oder MPI6_ConnectToPLCRouting ebenfalls erfolgreich aufgerufen worden sein.

Kurzbeschreibung

Mit dieser Funktion können die vorhandenen Bausteine der Typen OB, FC, FB, DB, SDB, SFC und SFB in der CPU ermittelt werden. Die vorhandenen Nummern werden dabei in einem Array geliefert.

Familie S7-1200® und LOGO!®

Die Funktion kann nicht bei bei CPUs der Reihe S7-1200® und LOGO!® verwendet werden.

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT	Das Handle der Kommunikationsinstanz welche angesprochen wird. (Entfällt bei .Net-Wrapper-Klasse)
BlockType	BYTE	Angabe des Bausteintyps nach welchem gesucht werden soll. 1=OB, 2=FC, 3=FB, 4=DB, 5=SDB, 6=SFB, 7=SFC
BlockArray	WORD*	Array mit den Nummern der in der CPU vorhandenen Bausteine des Typs.
Number	WORD*	Anzahl der im Array eingetragenen Bausteine.
Error	WORD*	Ist ein schwerwiegender Fehler vorhanden, dann liefert die Funktion den Wert 0. In diesem Fall ist in Error der aufgetretene Fehler verzeichnet.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.8 Die Funktion: MPI6_EXT_DeleteBlockPlc (Nur in Extended-Version)

Voraussetzung zum Ausführen der Funktion

Eine der Einleitungsfunktionen (z.B. MPI6_OpenTcplp usw.) muss erfolgreich ausgeführt worden sein.

Des Weiteren muss die Funktion MPI6_ConnectToPLC oder MPI6_ConnectToPLCRouting ebenfalls erfolgreich aufgerufen worden sein.

Kurzbeschreibung

Mit dieser Funktion kann ein Baustein der Typen OB, FC, FB, DB und SDB in der CPU gelöscht werden. Der Baustein muss in der CPU vorhanden sein.

Achtung:

Für das Löschen der SDBs sollte die Funktion "MPI6_EXT_DeleteHardwareCfgInPlc" verwendet werden. Das Löschen von einzelnen SDBs ist nur in Einzelfällen auszuführen.

Hinweis:

Beim Ausführen dieser Funktion kann die CPU in den Zustand STOP wechseln!

Familie S7-1200® und LOGO!®

Die Funktion kann nicht bei bei CPUs der Reihe S7-1200® und LOGO!® verwendet werden.

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT	Das Handle der Kommunikationsinstanz welche angesprochen wird. (Entfällt bei .Net-Wrapper-Klasse)
BlockType	BYTE	Angabe des Bausteintyps der gelöscht werden soll. 1=OB, 2=FC, 3=FB, 4=DB, 5=SDB
BlockNr	WORD	Nummer des zu löschenden Bausteins.
Error	WORD*	Ist ein schwerwiegender Fehler vorhanden, dann liefert die Funktion den Wert 0. In diesem Fall ist in Error der aufgetretene Fehler verzeichnet.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.9 Die Funktion: **MPI6_EXT_LoadBlocksFromPlc** (Nur in Extended-Version)

Voraussetzung zum Ausführen der Funktion

Eine der Einleitungsfunktionen (z.B. `MPI6_OpenTcplp` usw.) muss erfolgreich ausgeführt worden sein.

Des Weiteren muss die Funktion `MPI6_ConnectToPLC` oder `MPI6_ConnectToPLCRouting` ebenfalls erfolgreich aufgerufen worden sein.

Kurzbeschreibung

Mit dieser Funktion können Bausteine unterschiedlichen Typs (OB, FC, FB, DB, SDB) aus der CPU geladen und in einer WLD-Datei gespeichert werden. Hat der Parameter "ErrorIfFileExist" den Wert 1, dann wird bei Vorhandensein der Datei ein Fehler gemeldet. Hat der Parameter den Wert 0, so werden die Bausteine aus der CPU an die vorhandene Datei angefügt. Dabei wird nicht überprüft, ob der Baustein bereits in der Datei vorhanden ist. Sollte ein Baustein doppelt in der WLD vorhanden sein, so wird immer nur der erste Baustein in der Datei gefunden.

Bei Verwendung der Funktion sollten die in der CPU vorhandenen Bausteine über die Funktion "**MPI6_EXT_GetBlocksInPlc**" ermittelt und danach für jeden Bausteintyp getrennt die Funktion "**MPI6_EXT_LoadBlocksFromPlc**" aufgerufen werden. Ab dem zweiten Aufruf ist dann der Parameter "ErrorIfFileExist" auf den Wert 0 zu setzen.

Die Funktion "**MPI6_EXT_LoadBlocksFromPlc**" wird intern auch von der Funktion "**MPI6_EXT_BackupPlc**" verwendet.

Familie S7-1200® und LOGO!®

Die Funktion kann nicht bei bei CPUs der Reihe S7-1200® und LOGO!® verwendet werden.

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT	Das Handle der Kommunikationsinstanz welche angesprochen wird. (Entfällt bei .Net-Wrapper-Klasse)
WldFileWithPath	CHAR*	Angabe der WLD-Datei mit Pfad in welche die Bausteine aus der CPU abgelegt werden sollen. Die Datei sollte beim ersten Aufruf der Funktion nicht vorhanden sein. Über den Parameter "ErrorIfFileExist" kann beeinflusst werden, ob beim Vorhandensein der Datei die Aktion ausgeführt werden soll oder nicht.
BlockTypeArray	BYTE*	In diesem Array sind die Typen der Bausteine anzugeben, welche aus der CPU zu laden sind. 1=OB, 2=FC, 3=FB, 4=DB, 5=SDB Wird z.B. im Index 0 des "BlockTypeArray" der Wert "2" für den Bausteintyp FC angegeben und im Index 0 des Arrays "BlockNumberArray" der Wert "10", dann wird der Baustein "FC10" aus der CPU geladen.
BlockNumberArray	WORD*	In diesem Array sind die Bausteinnummern der aus der CPU zu ladenden Bausteine anzugeben. Wird z.B. im Index 2 des "BlockTypeArray" der Wert "1" für den Bausteintyp OB angegeben und im Index 2 des Arrays "BlockNumberArray" der Wert "100", dann wird der Baustein "OB100" aus der CPU geladen.
CountBlocks	WORD	Gibt an wieviele Bausteine in den Arrays "BlockTypeArray" und "BlockNumberArray" angegeben sind.
Error	WORD*	Ist ein schwerwiegender Fehler vorhanden, dann liefert die Funktion den Wert 0. In diesem Fall ist in Error der aufgetretene Fehler verzeichnet.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).

1.10 Die Funktion: MPI6_EXT_CompressPlc (Nur in Extended-Version)

Voraussetzung zum Ausführen der Funktion

Eine der Einleitungsfunktionen (z.B. MPI6_OpenTcpIp usw.) muss erfolgreich ausgeführt worden sein.

Des Weiteren muss die Funktion MPI6_ConnectToPLC oder MPI6_ConnectToPLCRouting ebenfalls erfolgreich aufgerufen worden sein.

Kurzbeschreibung

Beim Ausführen dieser Funktion werden die gelöschten Bausteine aus dem Speicher der CPU entfernt.

S7-Bausteine werden beim Löschen nur für ungültig erklärt, verbleiben aber im Speicher der CPU. Erst das Komprimieren führt zum Entfernen der Bausteine und der Speicher der CPU wird neu organisiert.

Familie S7-1200® und LOGO!®

Die Funktion kann nicht bei bei CPUs der Reihe S7-1200® und LOGO!® verwendet werden.

Beschreibung der Parameter

Argument	C-Typ	Beschreibung
Handle	INT	Das Handle der Kommunikationsinstanz welche angesprochen wird. (Entfällt bei .Net-Wrapper-Klasse)
Error	WORD*	Ist ein schwerwiegender Fehler vorhanden, dann liefert die Funktion den Wert 0. In diesem Fall ist in Error der aufgetretene Fehler verzeichnet.
Funktionswert	BOOL	Wurde die Funktion erfolgreich ausgeführt, so wird der Wert '1' (TRUE) geliefert. Bei einem Fehler ist der Rückgabewert '0' (FALSE).