



ComDrvS7 V6.2X

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

Additional Documentation

for software developers

Revision level: April 2012

ComDrvS7 V6.2x Documentation

MHJ-Software

Albert-Einstein-Str. 101 • 75015 Bretten • Tel: +49 (0)7252-84969 Or 87890 • Fax: 78780

1 Advanced functions for ComDrvS7 V6.2	Page 2
1.1 The function: MPI6_OpenTcplpExt	Page 2
1.2 The function: MPI6_OpenTcplpExt_2	Page 3
1.3 The function: MPI6_OpenTcplpExt_3	Page 5
1.4 The function: MPI6_EXT_BackupPlc (only in the extended version)	Page 7
1.5 The function: MPI6_EXT_RestorePlc (only in the extended version)	Page 8
1.6 The function: MPI6_EXT_DeleteHardwareCfglnPlc (only in the extended version)	Page 9
1.7 The function: MPI6_EXT_GetBlockslnPlc (only in the extended version)	Page 10
1.8 The function: MPI6_EXT_DeleteBlockPlc (only in the extended version)	Page 11
1.9 The function: MPI6_EXT_LoadBlocksFromPlc (only in the extended version)	Page 12
1.10 The function: MPI6_EXT_CompressPlc (only in the extended version)	Page 14

1 Advanced functions for ComDrvS7 V6

1.1 The function: MPI6_OpenTcplpExt

Brief Description

The **MPI6_OpenTcplpExt** function must be called to communicate with a CPU for the first time, provided that the communication link is established via TCP/IP with an Ethernet-CP or a CPU with an integrated Ethernet interface. In contrast to the **MPI6_OpenTcplp** function, then **MPI6_OpenTcplpExt** function can also be supplied with the rack number of the CPU. This is only necessary when the CPU is installed in a rack with a number that is not 0.

The function establishes a connection with an Ethernet-CP that has the specified IP address. Furthermore, the slot number as well as the rack number of the CPU is required. For S7-300 systems, this must generally be specified as slot 2 and rack number 0 (in this case, function **MPI6_OpenTcplp** may also be used).

This function creates a communication instance. Variable "Handle" supplies the "identification" for this instance. This identifier must be passed to the other functions of the DLL to ensure that the specified communication path (IP address) is used (not required for the .Net wrapper class).

Description of the Parameters

Argument	C-type	Description
Handle	INT*	This is where the handle of the newly generated communication instance is returned (not applicable for the .Net wrapper class).
IPAddress	CHAR*	Provide the IP address of the Ethernet-CP or the integrated Ethernet interface of the CPU that will be used to execute the communications. The address is entered in the form "172.16.130.84".
PlcSlotNr	INT	Specifies the slot of the CPU that you want to communicate with. For S7-300 systems, this must usually be defined as slot 2.
PlcRackNr	INT	Specifies the rack of the CPU that you want to communicate with. For S7-300 Systems, This must usually be defined as rack 0.
Error	WORD*	If The Function Returns '0', An error has occurred during execution. In this case, the Error parameter contains an error value.
Function Return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.2 The function: **MPI6_OpenTcplpExt_2**

Brief description

The **MPI6_OpenTcplpExt_2** function must be called to communicate with a CPU for the first time, provided that the communication link is established via TCP/IP with an Ethernet-CP or a CPU with an integrated Ethernet interface. In contrast to the **MPI6_OpenTcplp** function, the **MPI6_OpenTcplpExt_2** function may be supplied with the rack number of the CPU as well as the port address that is used for the communication link.

The function establishes a connection with a CPU that has an Ethernet interface or an Ethernet-CP with the specified IP address. Furthermore, the slot number as well as the rack number of the CPU is required. For S7-300 Systems, This must generally be specified as slot 2 and rack number 0 (in this case, function **MPI6_OpenTcplp** may also be used).

The default port address for S7 CPUs is 102. This value must only be changed under exceptional circumstances. This may be necessary, for example, if you want to access multiple CPUs via a WAN-IP address where the distinction is made ??by the port address. In this case, a virtual router must convert the external port address to the router's internal port address 102 and the IP address of the CPU (the Ethernet-CP).

Another exception are the S7-SoftPLCs that have variable port settings (e.g. WinPLC-Engine).

The **MPI6_OpenTcplpExt_2** function also creates a communication instance. The Variable "Handle" Supplies the "identification" for this instance. This identifier must be passed to the other functions of the DLL to ensure that the specified communication path (IP address) is used (not required for the .Net wrapper class).

Description of the parameters

Argument	C-type	Description
Handle	INT*	This is where the handle of the newly generated communication instance is returned (not applicable for the .Net wrapper class).
IPAddress	CHAR*	Provide the IP address of the Ethernet-CP or the integrated Ethernet interface of the CPU that will be used to execute the communications. The address is entered in the form "172.16.130.84".
PlcSlotNr	INT	Specifies the slot of the CPU that you want to communicate with. For S7-300 systems, this must usually be defined as slot 2.
PlcRackNr	INT	Specifies the rack of the CPU that you want to communicate with. For S7-300 Systems, This must usually be defined as rack 0.
PortAddress	WORD	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).
Error	WORD*	If The Function Returns '0', An error has occurred during execution. In this case, the Error parameter contains an error value.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.3 The function: **MPI6_OpenTcplpExt_3**

Brief Description

The **MPI6_OpenTcplpExt_3** function must be called to communicate with a CPU for the first time, provided that communications with an Ethernet-CP or a CPU with integrated Ethernet interface takes place via a TCP/IP link.

Unlike the other **MPI6_OpenTcplp-Funktionen** functions, the **MPI6_OpenTcplpExt_3** function can be supplied with the rack number of the CPU, the respective communication port address and the IP address of the network adapter that is used to communicate with the PC. This is especially important when **multiple network adapters are included in the PC**. You must also use this function to establish a **VPN connection with the CPU**. In this case, you must specify the IP address of the virtual VPN adapter in the **parameter "IPAddressNetworkAdapter"**.

The function establishes a connection with a CPU that has an Ethernet interface or an Ethernet-CP using the IP address specified in the parameter "IPAddress". Furthermore, the slot number as well as the rack number of the CPU is required. For S7-300 Systems, This must usually be defined as Slot 2 and rack number 0.

The default port address for S7 CPUs is 102. This value must only be changed under exceptional circumstances. This may be necessary, for example, if you want to access multiple CPUs via a WAN-IP address where the distinction is made by the port address. In this case, a virtual router must convert the external port address to the router's internal port address 102 and the IP address of the CPU (the Ethernet-CP).

Another exception are the S7-SoftPLCs that have variable port settings (e.g. WinPLC-Engine).

The **MPI6_OpenTcplpExt_3** function also creates a communication instance. The Variable "Handle" Supplies the "identification" for this instance. This identifier must be passed to the other functions of the DLL to ensure that the specified communication path (IP address) is used (not required for the .Net wrapper class).

Description of the Parameters

Argument	C-type	Description
Handle	INT*	This is where the handle of the newly generated communication instance is returned (not applicable for the .Net wrapper class).
IPAddress	CHAR*	Provide the IP address of the Ethernet-CP or the integrated Ethernet interface of the CPU that will be used to execute the communications. The address is entered in the form "172.16.130.84".
PlcSlotNr	INT	Specifies the slot of the CPU that you want to communicate with. For S7-300 systems, this must usually be defined as slot 2.
PlcRackNr	INT	Specifies the rack of the CPU that you want to communicate with. For S7-300 Systems, This must usually be defined as rack 0.
PortAddress	WORD	The port address. S7 communication is always conducted via port address 102. The port address must be changed, for example, during remote support when multiple CPUs are accessed via a WAN address. In this case, the port address must be converted into an internal port address 102, because the S7 CPUs can only monitor this address. An exception are the CPUs with variable port address settings, e.g. S7-SoftPLCs.
IPAddressNetworkAdapter	CHAR*	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).
Error	WORD*	If The Function Returns '0', An error has occurred during execution. In this case, the Error parameter contains an error value.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.4 The function: MPI6_EXT_BackupPlc (only in the extended version)

Condition to execute the function

The initialisation functions (e.g. MPI6_OpenTcpIp, etc.) must have been completed successfully.

In addition, the call to the function MPI6_ConnectToPLC or MPI6_ConnectToPLCRouting must also have been successful.

Brief Description

This function can be used to load all the blocks of type OB, FC, FB, DB, and SDB that exist in the CPU and store these in a WLD file that does not exist as yet.

PLC family S7-1200® and LOGO!®

This function is not possible.

Description of the parameters

Argument	C-type	Description
Handle	INT	The handle of the communication instance, which is being addressed. (omitted for the .Net wrapper class)
WldFilePath	CHAR*	Definition of the WLD file with path. This file must not exist.
WarningsArray	WORD*	Any warnings that result from the action and that do not lead to a termination of the action are gathered in this array.
WarningsNumber	WORD*	Number of warnings in the array "WarningsArray".
Error	WORD*	When a severe error occurs, the function returns a value 0. In this case, the error is recorded in FatalError.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.5 The function: MPI6_EXT_RestorePlc (only in the extended version)

Condition to execute the function

The initialisation functions (e.g. MPI6_OpenTcplp, etc.) must have been completed successfully.

In additions, the call to the function MPI6_ConnectToPLC or MPI6_ConnectToPLCRouting must also have been successful.

Brief Description

This function can be used to transfer all the OB, FC, FB, DB and SDB type blocks that exist in a WLD file to a CPU. It is important to ensure that the hardware configuration in the SDBs match the CPU. In addition, the operands existing in the blocks and the block numbers must be admissible for the CPU. Otherwise, an error will occur during the transfer or subsequent operation of the CPU.

Important:

During the transfer, the CPU must be in STOP mode!

PLC family S7-1200® and LOGO!®

This function is not possible.

Description of the parameters

Argument	C-type	Description
Handle	INT	The handle of the communication instance, which is being addressed. (omitted for the .Net wrapper class)
WldFilePath	CHAR*	Definition of the WLD file with path.
ErrorBstStr	CHAR*	If a fault occurs when a block is being transferred, this argument contains the block where the error occurred.
CountBlocksSend	WORD*	Number of blocks transferred.
Error	WORD*	When a severe error occurs, the function returns a value 0. In this case, the error is recorded in Error.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.6 The function: MPI6_EXT_DeleteHardwareCfgInPlc (only in the extended version)

Condition to execute the function

The initialisation functions (e.g. MPI6_OpenTcpIp, etc.) must have been completed successfully.

In additions, the call to the function MPI6_ConnectToPLC or MPI6_ConnectToPLCRouting must also have been successful.

Brief Description

This function erases the hardware configuration in a CPU. It is important to note that the respective CPU may only be accessible via the MPI interface.

Important:

The CPU must be in STOP mode!

PLC family S7-1200® and LOGO!®

This function is not possible.

Description of the parameters

Argument	C-type	Description
Handle	INT	The handle of the communication instance, which is being addressed. (omitted for the .Net wrapper class)
Error	WORD*	When a severe error occurs, the function returns a value 0. In this case, the error is recorded in Error.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.7 The function: MPI6_EXT_GetBlocksInPlc (only in the extended version)

Condition to execute the function

The initialisation functions (e.g. MPI6_OpenTcplp, etc.) must have been completed successfully.

In additions, the call to the function MPI6_ConnectToPLC or MPI6_ConnectToPLCRouting must also have been successful.

Brief Description

This function can be used to determine which of the blocks of type OB, FC, FB, DB, SDB, SFC and SFB exist in the CPU. The existing numbers are supplied in an array.

PLC family S7-1200® and LOGO!®

This function is not possible.

Description of the parameters

Argument	C-type	Description
Handle	INT	The handle of the communication instance, which is being addressed. (omitted for the .Net wrapper class)
BlockType	BYTE	Defines the block type that you wish to search for. 1=OB, 2=FC, 3=FB, 4=DB, 5=SDB, 6=SFB, 7=SFC
BlockArray	WORD*	Array containing the block numbers of the respective type that exist in the CPU.
Number	WORD*	Number of blocks that was entered into the array.
Error	WORD*	When a severe error occurs, the function returns a value 0. In this case, the error is recorded in Error.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.8 The function: MPI6_EXT_DeleteBlockPlc (only in the extended version)

Condition to execute the function

The initialisation functions (e.g. MPI6_OpenTcplp, etc.) must have been completed successfully.

In additions, the call to the function MPI6_ConnectToPLC or MPI6_ConnectToPLCRouting must also have been successful.

Brief Description

This function can be used to erase a block of the type OB, FC, FB, DB and SDB in the CPU. The block must already be available in the CPU.

Attention:

the "MPI6_EXT_DeleteHardwareCfgInPlc" function should be used for the deletion of the SDBs. Deletion of individual SDBs should only be executed in individual cases.

Note:

When this function is executed, the CPU can change to STOP mode!

PLC family S7-1200® and LOGO!®

This function is not possible.

Description of the parameters

Argument	C-type	Description
Handle	INT	The handle of the communication instance, which is being addressed. (omitted for the .Net wrapper class)
BlockType	BYTE	Defines the block type that must be erased. 1=OB, 2=FC, 3=FB, 4=DB, 5=SDB
BlockNr	WORD	The number of the block that must be erased.
Error	WORD*	When a severe error occurs, the function returns a value 0. In this case, the error is recorded in Error.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.9 The function: **MPI6_EXT_LoadBlocksFromPlc** (only in the extended version)

Condition to execute the function

The initialisation functions (e.g. **MPI6_OpenTcplp**, etc.) must have been completed successfully.

In additions, the call to the function **MPI6_ConnectToPLC** or **MPI6_ConnectToPLCRouting** must also have been successful.

Brief Description

This function can be used to load different block types (OB, FC, FB, DB, SDB) from the CPU and stored them in a WLD file. When the parameter "ErrorIfFileExist" is set to 1, an error is raised if the file exists. If the value of the parameter is 0, then the blocks from the CPU are appended to the file that exists. The function does not check whether the block is already present in the file. If the block is duplicated in the WLD, only the first block will be located in the file.

Before using the function, the blocks that exist in the CPU should be determined by means of the "**MPI6_EXT_GetBlocksInPlc**" function and then the function "**MPI6_EXT_LoadBlocksFromPlc**" should be called separately for each type of block. As of the second call, the parameter "ErrorIfFileExist" must be set to 0.

Internally, the "**MPI6_EXT_LoadBlocksFromPlc**" function is also used by the function "**MPI6_EXT_BackupPlc**".

PLC family S7-1200® and LOGO!®

This function is not possible.

Description of the parameters

Argument	C-type	Description
Handle	INT	The handle of the communication instance, which is being addressed. (omitted for the .Net wrapper class)
WldFileWithPath	CHAR*	Specification of the WLD file where the blocks retrieved from the CPU must be saved. The file should not be the first call to be present. The parameter "ErrorIfFileExist" can determine whether the action should be executed or not if the file exists.
BlockTypeArray	BYTE*	This array is used to define the block types that must be loaded from the CPU. 1=OB, 2=FC, 3=FB, 4=DB, 5=SDB If for example, index 0 of the "BlockTypeArray" is set to a value "2" for block type FC and index 0 of the array "BlockNumberArray" is set to "10", block "FC10" will be loaded from the CPU.
BlockNumberArray	WORD*	This array is used to define the block types that must be loaded from the CPU. If for example, index 2 of the "BlockTypeArray" is set to a value "1" for block type OB and index 2 of the array "BlockNumberArray" is set to "100", block "OB100" will be loaded from the CPU.
CountBlocks	WORD	Indicates how many blocks were entered into the array "BlockTypeArray" and "BlockNumberArray".
Error	WORD*	When a severe error occurs, the function returns a value 0. In this case, the error is recorded in Error.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).

1.10 The function: MPI6_EXT_CompressPlc (only in the extended version)

Condition to execute the function

The initialisation functions (e.g. MPI6_OpenTcpIp, etc.) must have been completed successfully.

In additions, the call to the function MPI6_ConnectToPLC or MPI6_ConnectToPLCRouting must also have been successful.

Brief Description

When this function is executed, the deleted blocks are removed from the memory of the CPU.

Deleting S7 blocks only marks the blocks as being invalid, but they remain in the memory of the CPU. Compression removes the blocks from memory and reorganises the memory of the CPU.

PLC family S7-1200® and LOGO!®

This function is not possible.

Description of the parameters

Argument	C-type	Description
Handle	INT	The handle of the communication instance, which is being addressed. (omitted for the .Net wrapper class)
Error	WORD*	When a severe error occurs, the function returns a value 0. In this case, the error is recorded in Error.
Function return	BOOL	If the function was executed successfully, a value '1' (TRUE) is returned. When an error has occurred, the returned value is '0' (FALSE).