

[Home \(/\)](#) / [S7 \(https://csharpdoc.hotexamples.com/namespace/S7/\)](https://csharpdoc.hotexamples.com/namespace/S7/) / [Net \(https://csharpdoc.hotexamples.com/namespace/S7.Net/\)](https://csharpdoc.hotexamples.com/namespace/S7.Net/) / [Plc](#)

C# Class S7.Net.Plc

Creates an instance of S7.Net driver

Inheritance: IDisposable

Show file

Open project: killnine/s7netplus

Class Usage Examples (<https://csharp.hotexamples.com/examples/S7.Net/Plc/-/php-plc-class-examples.html>)

Public Methods

Method	Description	
ClearLastError (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) () : void	Sets the LastErrorCode to NoError and LastErrorString to String.Empty	
Close (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) () : void	Disonnects from the plc and close the socket	(https://csharp.hotexamples.com/examples/S7.Net/Plc/Close/php-plc-close-method-exa)
Dispose (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) () : void	Releases all resources, disonnects from the plc and closes the socket	
Open (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) () : ErrorCode	Open a socket and connects to the plc, sending all the corrected package and returning if the connection was successful (ErrorCode.NoError) of it was wrong.	(https://csharp.hotexamples.com/examples/S7.Net/Plc/Open/php-plc-open-method-exa)

Plc https://csharpdoc.hotexamples.com/class/S7.Net/Plc# (CpuType cpu, string ip, Int16 rack, Int16 slot) : System	Creates a PLC object with all the parameters needed for connections. For S7-1200 and S7-1500, the default is rack = 0 and slot = 0. You need slot > 0 if you are connecting to external ethernet card (CP). For S7-300 and S7-400 the default is rack = 0 and slot = 2.	
Read https://csharpdoc.hotexamples.com/class/S7.Net/Plc# (DataType dataType, int db, int startByteAdr, VarType varType, int varCount) : object	Read and decode a certain number of bytes of the "VarType" provided. This can be used to read multiple consecutive variables of the same type (Word, DWord, Int, etc). If the read was not successful, check LastErrorCode or LastErrorString.	https://csharp.hotexamples.com/examples/S7.Net/Plc/Read/php-plc-read-method-ex
Read https://csharpdoc.hotexamples.com/class/S7.Net/Plc# (string variable) : object	Reads a single variable from the plc, takes in input strings like "DB1.DBX0.0", "DB20.DBD200", "MB20", "T45", etc. If the read was not successful, check LastErrorCode or LastErrorString.	https://csharp.hotexamples.com/examples/S7.Net/Plc/Read/php-plc-read-method-ex
ReadBytes https://csharpdoc.hotexamples.com/class/S7.Net/Plc# (DataType dataType, int db, int startByteAdr, int count) : byte[]	Reads a number of bytes from a DB starting from a specified index. This handles more than 200 bytes with multiple requests. If the read was not successful, check LastErrorCode or LastErrorString.	
ReadClass https://csharpdoc.hotexamples.com/class/S7.Net/Plc# (object sourceClass, int db, int startByteAdr) : void	Reads all the bytes needed to fill a class in C#, starting from a certain address, and set all the properties values to the value that are read from the plc. This reads only properties, it doesn't read private variable or public variable without {get;set;} specified.	

ReadMultipleVars (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (List dataltems) : void	Reads multiple vars in a single request. You have to create and pass a list of Dataltems and you obtain in response the same list with the values. Values are stored in the property "Value" of the dataltem and are already converted. If you don't want the conversion, just create a dataltem of bytes. Dataltems must not be more than 20 (protocol restriction) and bytes must not be more than 200 + 22 of header (protocol restriction).	
ReadStruct (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (Type structType, int db, int startByteAdr) : object	Reads all the bytes needed to fill a struct in C#, starting from a certain address, and return an object that can be casted to the struct.	
Write (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (DataType dataType, int db, int startByteAdr, object value) : ErrorCode	Takes in input an object and tries to parse it to an array of values. This can be used to write many data, all of the same type. You must specify the memory area type, memory are address, byte start address and bytes count. If the read was not successful, check LastErrorCode or LastErrorString.	(https://csharp.hotexamples.com/examples/S7.Net/Plc/Write/php-plc-write-method-ex
Write (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (string variable, object value) : ErrorCode	Writes a single variable from the plc, takes in input strings like "DB1.DBX0.0", "DB20.DBD200", "MB20", "T45", etc. If the write was not successful, check LastErrorCode or LastErrorString.	(https://csharp.hotexamples.com/examples/S7.Net/Plc/Write/php-plc-write-method-ex

WriteBytes (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (<i>DataType</i> dataType, int db, int startByteAdr, byte value) : <i>ErrorCode</i>	Write a number of bytes from a DB starting from a specified index. This handles more than 200 bytes with multiple requests. If the write was not successful, check <i>LastErrorCode</i> or <i>LastErrorString</i> .
WriteClass (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (<i>object</i> classValue, int db, int startByteAdr) : <i>ErrorCode</i>	Writes a C# class to a DB in the plc
WriteStruct (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (<i>object</i> structValue, int db, int startByteAdr) : <i>ErrorCode</i>	Writes a C# struct to a DB in the plc

Private Methods

Method	Description	
CreateReadDataRequestPackage (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (<i>DataType</i> dataType, int db, int startByteAdr, int count = 1) : <i>ByteArray</i>	Create the bytes-package to request data from the plc. You have to specify the memory type (dataType), the address of the memory, the address of the byte and the bytes count.	(https://csharpsrc.hotexamples.com/method/S7.Net/Plc/CreateReadDataRequestPackage/-)
ParseBytes (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (<i>VarType</i> varType, byte bytes, int varCount) : <i>object</i>	Given a S7 variable type (Bool, Word, DWord, etc.), it converts the bytes in the appropriate C# format.	(https://csharpsrc.hotexamples.com/method/S7.Net/Plc/ParseBytes/-)
ReadBytesWithASingleRequest (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (<i>DataType</i> dataType, int db, int startByteAdr, int count) : <i>byte[]</i>		(https://csharpsrc.hotexamples.com/method/S7.Net/Plc/ReadBytesWithASingleRequest/-)
ReadHeaderPackage (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (<i>int</i> amount = 1) : <i>ByteArray</i>	Creates the header to read bytes from the plc	(https://csharpsrc.hotexamples.com/method/S7.Net/Plc/ReadHeaderPackage/-)
VarTypeToByteLength (https://csharpdoc.hotexamples.com/class/S7.Net/Plc#) (<i>VarType</i> varType, int varCount = 1) : <i>int</i>	Given a S7 variable type (Bool, Word, DWord, etc.), it returns how many bytes to read.	(https://csharpsrc.hotexamples.com/method/S7.Net/Plc/VarTypeToByteLength/-)

WriteBytesWithASingleRequest
(<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>)
(DataType dataType, int db, int startByteAdr, byte value) : ErrorCode

Writes up to 200 bytes to the plc and returns NoError if successful. You must specify the memory area type, memory are address, byte start address and bytes count. If the write was not successful, check LastErrorCode or LastErrorString.

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/WriteBytesWithASingleRequest/->)

Method Details

ClearLastError() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/ClearLastError/->)

(<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>)

Sets the LastErrorCode to NoError and LastErrorString to String.Empty

public **ClearLastError** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>) () : void

returnvoid

Close() public method

(<https://csharp.hotexamples.com/examples/S7.Net/Plc/Close/php-plc-close-method-examples.html>)

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/Close/->)

Disonnects from the plc and close the socket

public **Close** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>) () : void

returnvoid

Dispose() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/Dispose/->)

(<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>)

Releases all resources, disonnects from the plc and closes the socket

public **Dispose** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>) () : void

returnvoid

Open() public method

(<https://csharp.hotexamples.com/examples/S7.Net/Plc/Open/php-plc-open-method-examples.html>)

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/Open/->)

Open a socket and connects to the plc, sending all the corrected package and returning if the connection was successful (ErrorCode.NoError) of it was wrong.

public **Open** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>) () : ErrorCode

returnErrorCode

Plc() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/Plc/->)

(<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>)

Creates a PLC object with all the parameters needed for connections. For S7-1200 and S7-1500, the default is rack = 0 and slot = 0. You need slot > 0 if you are connecting to external ethernet card (CP). For S7-300 and S7-400 the default is rack = 0 and slot = 2.

public **Plc** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc#>) (CpuType cpu, string ip, Int16 rack, Int16 slot) : System

cpu	CpuType	CpuType of the plc (select from the enum)
ip	string	Ip address of the plc
rack	System.Int16	rack of the plc, usually it's 0, but check in the hardware configuration of Step7 or TIA portal
slot	System.Int16	slot of the CPU of the plc, usually it's 2 for S7300-S7400, 0 for S7-1200 and S7-1500. /// If you use an external ethernet card, this must be set accordingly.

return System

Read() public method

(<https://csharp.hotexamples.com/examples/S7.Net/Plc/Read/php-plc-read-method-examples.html>) (<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/Read/->)

Read and decode a certain number of bytes of the "VarType" provided. This can be used to read multiple consecutive variables of the same type (Word, DWord, Int, etc). If the read was not successful, check LastErrorCode or LastErrorString.

public **Read** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (DataType dataType, int db, int startByteAdr, VarType varType, int varCount) : object

dataType	DataType	Data type of the memory area, can be DB, Timer, Counter, Merker(Memory), Input, Output.
db	int	Address of the memory area (if you want to read DB1, this is set to 1). This must be set also for other memory area types: counters, timers,etc.
startByteAdr	int	Start byte address. If you want to read DB1.DBW200, this is 200.
varType	VarType	Type of the variable/s that you are reading
varCount	int	
return	object	

Read() public method

(<https://csharp.hotexamples.com/examples/S7.Net/Plc/Read/php-plc-read-method-examples.html>) (<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/Read/->)

Reads a single variable from the plc, takes in input strings like "DB1.DBX0.0", "DB20.DBD200", "MB20", "T45", etc. If the read was not successful, check LastErrorCode or LastErrorString.

public **Read** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (string variable) : object

variable	string	Input strings like "DB1.DBX0.0", "DB20.DBD200", "MB20", "T45", etc.
return	object	

ReadBytes() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/ReadBytes/->) (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>)

Reads a number of bytes from a DB starting from a specified index. This handles more than 200 bytes with multiple requests. If the read was not successful, check LastErrorCode or LastErrorString.

public **ReadBytes** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (DataType dataType, int db, int startByteAdr, int count) : byte[]

dataType	DataType	Data type of the memory area, can be DB, Timer, Counter, Merker(Memory), Input, Output.
db	int	Address of the memory area (if you want to read DB1, this is set to 1). This must be set also for other memory area types: counters, timers,etc.
startByteAdr	int	Start byte address. If you want to read DB1.DBW200, this is 200.
count	int	Byte count, if you want to read 120 bytes, set this to 120.
return	byte[]	

ReadClass() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/ReadClass/->) (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>)

Reads all the bytes needed to fill a class in C#, starting from a certain address, and set all the properties values to the value that are read from the plc. This reads only properties, it doesn't read private variable or public variable without {get;set;} specified.

public **ReadClass** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (object sourceClass, int db, int startByteAdr) : void

sourceClass	object	Instance of the class that will store the values
db	int	Index of the DB; es.: 1 is for DB1
startByteAdr	int	Start byte address. If you want to read DB1.DBW200, this is 200.
return	void	

ReadMultipleVars() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/ReadMultipleVars/->) (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>)

Reads multiple vars in a single request. You have to create and pass a list of Dataltems and you obtain in response the same list with the values. Values are stored in the property "Value" of the dataltem and are already converted. If you don't want the conversion, just create a dataltem of bytes. Dataltems must not be more than 20 (protocol restriction) and bytes must not be more than 200 + 22 of header (protocol restriction).

public **ReadMultipleVars** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (List dataltems) : void

dataItems	List	List of dataitems that contains the list of variables that must be read. Maximum 20 dataitems are accepted.
return	void	

ReadStruct() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/ReadStruct/->)

(<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>)

Reads all the bytes needed to fill a struct in C#, starting from a certain address, and return an object that can be casted to the struct.

public **ReadStruct** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (Type structType, int db, int startByteAdr) : object

structType	System.Type	Type of the struct to be readed (es.: TypeOf(MyStruct)).
db	int	Address of the DB.
startByteAdr	int	Start byte address. If you want to read DB1.DBW200, this is 200.
return	object	

Write() public method

(<https://csharp.hotexamples.com/examples/S7.Net/Plc/Write/php-plc-write-method-examples.html>)

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/Write/->)

Takes in input an object and tries to parse it to an array of values. This can be used to write many data, all of the same type. You must specify the memory area type, memory are address, byte start address and bytes count. If the read was not successful, check LastErrorCode or LastErrorString.

public **Write** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (DataType dataType, int db, int startByteAdr, object value) : ErrorCode

dataType	DataType	Data type of the memory area, can be DB, Timer, Counter, Merker(Memory), Input, Output.
db	int	Address of the memory area (if you want to read DB1, this is set to 1). This must be set also for other memory area types: counters, timers,etc.
startByteAdr	int	Start byte address. If you want to read DB1.DBW200, this is 200.
value	object	Bytes to write. The lenght of this parameter can't be higher than 200. If you need more, use recursion.
return	ErrorCode	

Write() public method

(<https://csharp.hotexamples.com/examples/S7.Net/Plc/Write/php-plc-write-method-examples.html>)

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/Write/->)

Writes a single variable from the plc, takes in input strings like "DB1.DBX0.0", "DB20.DBD200", "MB20", "T45", etc. If the write was not successful, check LastErrorCode or LastErrorString.

public **Write** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (string variable, object value) : ErrorCode

variable	string	Input strings like "DB1.DBX0.0", "DB20.DBD200", "MB20", "T45", etc.
value	object	Value to be written to the plc
return	ErrorCode	

WriteBytes() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/WriteBytes/->)

(<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>)

Write a number of bytes from a DB starting from a specified index. This handles more than 200 bytes with multiple requests. If the write was not successful, check LastErrorCode or LastErrorString.

public **WriteBytes** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (DataType dataType, int db, int startByteAdr, byte value) : ErrorCode

dataType	DataType	Data type of the memory area, can be DB, Timer, Counter, Merker(Memory), Input, Output.
db	int	Address of the memory area (if you want to read DB1, this is set to 1). This must be set also for other memory area types: counters, timers,etc.
startByteAdr	int	Start byte address. If you want to write DB1.DBW200, this is 200.
value	byte	Bytes to write. If more than 200, multiple requests will be made.
return	ErrorCode	

WriteClass() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/WriteClass/->)

(<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>)

Writes a C# class to a DB in the plc

public **WriteClass** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/#>) (object classValue, int db, int startByteAdr) : ErrorCode

classValue	object	The class to be written
db	int	Db address
startByteAdr	int	Start bytes on the plc
return	ErrorCode	

WriteStruct() public method

(<https://csharpsrc.hotexamples.com/method/S7.Net/Plc/WriteStruct/>) (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/>)

Writes a C# struct to a DB in the plc

public **WriteStruct** (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc/>) (object structValue, int db, int startByteAdr) : ErrorCode


structValue	object	The struct to be written
db	int	Db address
startByteAdr	int	Start bytes on the plc
return	ErrorCode	

■ ■ ■

Top Classes (<https://csharp.hotexamples.com/site/trends?type=csharp%7Cc>) | Documentation (<https://csharpdoc.hotexamples.com/doc/map>) | Privacy Policy (</site/privacy>) | Advertise with us (<https://cpp.hotexamples.com/direct-sales.html>)

PHP (<https://hotexamples.com/>) | C# (CSharp) (<https://csharp.hotexamples.com/>) | Java (<https://java.hotexamples.com/>) | Golang (<https://golang.hotexamples.com/>) | C++ (Cpp) (<https://cpp.hotexamples.com/>) | Python (<https://python.hotexamples.com/>) | JavaScript (<https://javascript.hotexamples.com/>) | TypeScript (<https://typescript.hotexamples.com/>)

EN (<https://csharpdoc.hotexamples.com/class/S7.Net/Plc>) | RU (<https://csharpdoc.hotexamples.com/ru/class/S7.Net/Plc>) | DE (<https://csharpdoc.hotexamples.com/de/class/S7.Net/Plc>) | FR (<https://csharpdoc.hotexamples.com/fr/class/S7.Net/Plc>) | ES (<https://csharpdoc.hotexamples.com/es/class/S7.Net/Plc>) | PT (<https://csharpdoc.hotexamples.com/pt/class/S7.Net/Plc>) | IT (<https://csharpdoc.hotexamples.com/it/class/S7.Net/Plc>) | JP (<https://csharpdoc.hotexamples.com/jp/class/S7.Net/Plc>) | ZH (<https://csharpdoc.hotexamples.com/zh/class/S7.Net/Plc>)

 (<http://www.liveinternet.ru/click>)