IO functions

Jump to bottom

Federico Barresi edited this page on Jan 15, 2019 · 9 revisions

These functions allow the S7Client to exchange data with a PLC.

Function	Purpose
ReadArea	Reads a data area from a PLC
WriteArea	Writes a data area into a PLC
ReadMultiVars	Reads different kind of variables from a PLC simultaneously
WriteMultiVars	Writes different kind of variables into a PLC simultaneously

ReadArea

Description

This is the main function to read data from a PLC. With it you can read DB, Inputs, Outputs, Merkers, Timers and Counters.

Declaration

```
public int ReadArea(int Area, int DBNumber, int Start, int Amount, int WordLen, byte[] Bu
```

```
public int ReadArea(int Area, int DBNumber, int Start, int Amount, int  WordLen, byte[] Bu
```

Parameters

Name	Type	Note
Area	int	Area identifier

Name	Type	Note
DBNumber	int	DB Number (if Area = S7AreaDB, otherwise ignored)
Start	int	Offset to start
Amount	int	Amount of elements to read (Quantity, not the size)
WordLen	int	Word Size
Buffer	int	Buffer
BytesRead	int	Number of bytes read

Area Table

Name	Value	Meaning
S7Consts.S7AreaPE	0x81	Process inputs
S7Consts.S7AreaPA	0x82	Process outputs
S7Consts.S7AreaMK	0x83	Merkers
S7Consts.S7AreaDB	0x84	Datablocks
S7Consts.S7AreaCT	0x1C	Counters
S7Consts.S7AreaTM	0x1D	Timers

WordLen Table

Name	Value	Meaning
S7Consts.S7WLBit	0x01	Bit (inside a word)
S7Consts.S7WLByte	0x02	Byte (8 bit)
S7Consts.S7WLWord	0x04	Word (16 bit)
S7Consts.S7WLDWord	0x06	Double Word (32 bit)
S7Consts.S7WLReal	0x08	Real (32 bit float)
S7Consts.S7WLCounter	0x01C	Counter (16 bit)
S7Consts.S7WLTimer	0x01D	Timer (16 bit)

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

Remarks

As said, every data packet exchanged with a PLC must fit in a PDU, whose size is fixed and varies from 240 up to 960 bytes.

This function completely hides this concept, the data that you can transfer in a single call depends only on the size available of the data area (i.e. obviously, you cannot read 1024 bytes from a DB whose size is 300 bytes).

If this data size exceeds the PDU size, the packet is automatically split across more subsequent transfers.

If either S7AreaCT or S7AreaTM is selected, WordLen must be either S7WLCounter or S7WLTimer (However no error is raised and the values are internally fixed).

Your buffer should be large enough to receive the data.

Particularly: **Buffer size (byte) = Word size * Amount**

Where:

	Word size
S7Consts.S7WLBit	1
S7Consts.S7WLByte	1
S7Consts.S7WLWord	2
S7Consts.S7WLDWord	4
S7Consts.S7WLReal	4
S7Consts.S7WLCounter	2
S7Consts.S7WLTimer	2

Notes

- When **WordLen** = **S7WLBit** the Offset (Start) must be expressed in bits. E.g. The Start for DB4.DBX10.3 is (10*8)+3 = 83.
- Since Amount is the number of elements read, BytesRead returns the effective number of bytes.

WriteArea

Description

This is the main function to write data into a PLC. It's the complementary function of ReadArea(), the parameters and their meanings are the same.

The only difference is that the data is transferred from the byte buffer into PLC.

Declaration

```
public int WriteArea(int Area, int DBNumber, int Start, int Amount, int WordLen, byte[] Bu
```

```
public int WriteArea(int Area, int DBNumber, int Start, int Amount, int WordLen, byte[] Bu
```

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

See ReadArea for parameters and remarks.

Use S7Helper methods to insert S7 data types (int, word, real ...) into the byte buffer.

ReadMultiVars

Description

This is function allows to read different kind of variables from a PLC in a single call. With it you can read DB, Inputs, Outputs, Merkers, Timers and Counters.

Declaration

```
public int ReadMultiVars(S7DataItem[] Items, int ItemsCount)
```

Parameters

Name	Type	Note
Items	S7DataItem[]	see below
ItemsCount	int	Number of Items to read

S7DataItem struct description

Field	Type	Meaning

Field	Type	Meaning
Area	int	Area identifier
WordLen	int	Word size
Result	int	Item operation result*
DBNumber	int	DB Number (only if Area = S7AreaDB, otherwise is ignored)
Start	int	Offset to start
Amount	int	Amount of words to read
pData	IntPtr	Pointer to user's buffer

(*: Since could happen that some variables are read, some other not because maybe they don't exist in PLC. Is important to check the single item Result)

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

Remarks

To use ReadMultiVars a special class is supplied S7MultiVar that avoids the use of unsafe code (since there is a IntPtr)

Due the different kind of variables involved , there is no split feature available for this function, so **the maximum data size must not exceed the PDU size**.

The advantage of this function becomes big when you have many small noncontiguous variables to be read.

WriteMultiVars

Description

This is function allows to write different kind of variables from a PLC in a single call. With it you can read DB, Inputs, Outputs, Merkers, Timers and Counters.

Declaration

```
public int WriteMultiVars(S7DataItem[] Items, int ItemsCount)
```

See ReadMultiVars for parameters and remarks.

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

Lean Data I/O functions

These are utility functions that simplify the use of ReadArea and WriteArea.

They call internally ReadArea and WriteArea passing the correct Area and WordLen.

Function	Purpose
DBRead	Reads a part of a DB
DBWrite	Writes a part of a DB
ABRead	Reads a part of IPU area
ABWrite	Writes a part of IPU area
EBRead	Reads a part of IPI area
EBWrite	Writes a part of IPI area
MBRead	Reads a part of Merkers area
MBWrite	Writes a part of Merkers area
TMRead	Reads timers
TMWrite	Write timers
CTRead	Reads counters
CTWrite	Write counters

Block oriented functions

Function	Purpose
GetAgBlockInfo	Returns info about a given block in PLC memory
DBGet	Uploads a DB from the PLC
DBFill	Fills a DB into the PLC with a given value

GetAgBlockInfo

Description

Returns some information about a given block. This function is very useful if you need to read or write data in a DB which you do not know the size in advance (see MC7Size field).

This function is also internally used by DBGet.

Declaration

```
public int GetAgBlockInfo(int BlockType, int BlockNumber, ref S7BlockInfo Block)
```

Parameters

Name	Type	Note
BlockType	int	Type of Block that we need
BlockNum	int	Number of Block
Block	S7BlockInfo	S7BlockInfo struct

BlockType values

Helper constant	Type	Value
S7Consts.Block_OB	OB	0x38
S7Consts.Block_DB	DB	0x42
S7Consts.Block_SDB	SDB	0x42
S7Consts.Block_FC	FC	0x43
S7Consts.Block_SFC	SFC	0x44
S7Consts.Block_FB	FB	0x45
S7Consts.Block_SFB	SFB	0x46

S7BlockInfo struct

```
public struct S7BlockInfo {
    public int BlkType;    // Block Type (see SubBlkType table)
    public int BlkNumber;  // Block number
    public int BlkLang;    // Block Language (see LangType Table)
    public int BlkFlags;   // Block flags (bitmapped)
    public int MC7Size;    // The real size in bytes
    public int LoadSize;   // Load memory size
    public int LocalData;  // Local data
    public int SBBLength;  // SBB Length
```

```
        public int CheckSum;    // Checksum
        public int Version;     // Version (BCD 00<HI><LO>)
        public string CodeDate;
        public string IntfDate;
        public string Author;
        public string Family;
        public string Header;
    };
```

This struct is filled by the function, some fields require additional info:

SubBlockType table

	Value	Type
SubBlk_OB	0x08	OB
SubBlk_DB	0x0A	DB
SubBlk_SDB	0x0B	SDB
SubBlk_FC	0x0C	FC
SubBlk_SFC	0x0D	SFC
SubBlk_FB	0x0E	FB
SubBlk_SFB	0x0E	SFB

LangType table

	Value	Language
BlockLangAWL	0x01	AWL
BlockLangKOP	0x02	KOP
BlockLangFUP	0x03	FUP
BlockLangSCL	0x04	SCL
BlockLangDB	0x05	DB
BlockLangGRAPH	0x06	GRAPH

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

DBGet

Description

Read an entire DB from the PLC without the need of specifying its size. As output SizeRead will contain the size read.

This function is not subject to the security level set.

Declaration

```
public int DBGet(int DBNumber, byte[] usrData, ref int Size)
```

Parameters

Name	Type	Meaning
DBNumber	int	DB Number
usrData	byte[]	user buffer
Size	byte[]	In -> Buffer size supplied / Out -> Number of bytes read

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

Remarks

This function first gathers the DB size via GetAgBlockInfo then calls ReadArea if the Buffer size is greater than the DB size, otherwise returns an error.

DBFill

Description

Fills a DB in AG with a given byte pattern (int) without the need of specifying its size.

Declaration

```
public int Cli_DBFill(int DBNumber, int FillChar);
```

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

Remarks

Fillchar is an integer for efficiency reasons, only the lowest byte is used.

Date/Time functions

These functions allow to read/modify the date and time of a PLC.

Imagine a production line in which each PLC saves the data with date/time field inside, it is very important that the date be up to date.

Both CP X43 and internal PN allow to synchronize date and time but you need an NTP server, and in some cases (old hardware or CP343-1 Lean or old firmware release) this doesn't work properly.

Sharp7, since it uses the same method of S7 Manager, always works.

Function	Purpose
GetPlcDateTime	Returns the PLC date/time
SetPlcDateTime	Sets the PLC date/time with a given value
SetPlcSystemDateTime	Sets the PLC date/time with the host (PC) date/time

GetPlcDateTime

Description

Reads PLC date and time into a C# DateTime class instance.

Declaration

```
public int GetPlcDateTime(ref DateTime DT)
```

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

SetPlcDateTime

Description

Sets the PLC date and time.

Declaration

```
public int SetPlcDateTime(DateTime DT)
```

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

SetPlcSystemDateTime

Description

Sets the PLC date and time in accord to the PC system Date/Time.

Declaration

```
public int SetPlcSystemDateTime()
```

Return value

- 0 : The Client is successfully connected (or was already connected).
- Other values : see the Errors Code List.

S7MultiVar

Clone this wiki locally

```
https://github.com/fbarresi/Sharp7.wiki.git
```