Siemens M-PROT Driver

Filename	MPROT.DLL
Manufacturer	Siemens, Vipa
Devices	PLCs S7-200, S7-300, and S7-400 Siemens' PLCs series; Vipa's Speed7 and other devices compatible with any protocol of the Driver
Protocol	PPI and MPI (Serial); MPI encapsulated in Ethernet and ISO over TCP (RFC1006 or S7-TCP/IP on Ethernet interface)
Version	3.1.2
Latest Update	05/12/2015
Platform	Win32
Dependencies	IOKit v2.00
Superblocks	Yes
Level	0

Introduction

The Siemens multi-protocol Driver (M-Prot) communicates with Siemens S7-200, S7-300, S7-400, and S7-1200 PLCs, as well as the VIPA's Speed7 device using the Siemens PPI, MPI, ISOTCP, and MPI protocols encapsulated in Ethernet MPI (IBHLink).

The PPI protocol must be used only with the S7-200 series, by using the RS232-PPI/MPI converter cable provided by Siemens.

The MPI protocol can be used with the S7-300 and S7-400 series via an RS232-PPI/MPI converter cable provided by Siemens, or also with the VIPA's Speed7 series on the MPI port using a common RS-232 cable.

The ISOTCP protocol (which is also known as ISO over TCP, RFC1006, or S7-TCP/IP on several hardware vendor brochures) can be used with the Siemens S7-300 and S7-400 series by using a CP-3XX, CP-433, or CP-443 Ethernet card; for the S7-1200 model, and also for VIPA's Speed7 series, directly on the CPU's Ethernet port. For the S7-200 model, there is a special variation of the ISOTCP protocol for use with the CP-243 interface. This protocol is called ISOTCP243.

For PLCs that do not have an Ethernet port, an alternative can be the Ethernet/MPI IBHLink converter provided by IBH Softec or Hilscher, which works on the FDL level. By using this converter, the advantage is a faster nominal speed, up to 187 kbps on an MPI network, while using a serial converter the speed is only 38.4 kbps. Using this converter is an alternative to the CP5611 or similar boards.

Another similar alternative is the NETLink PRO Eth converter cable provided by Softing, which converts from ISOTCP to MPI.

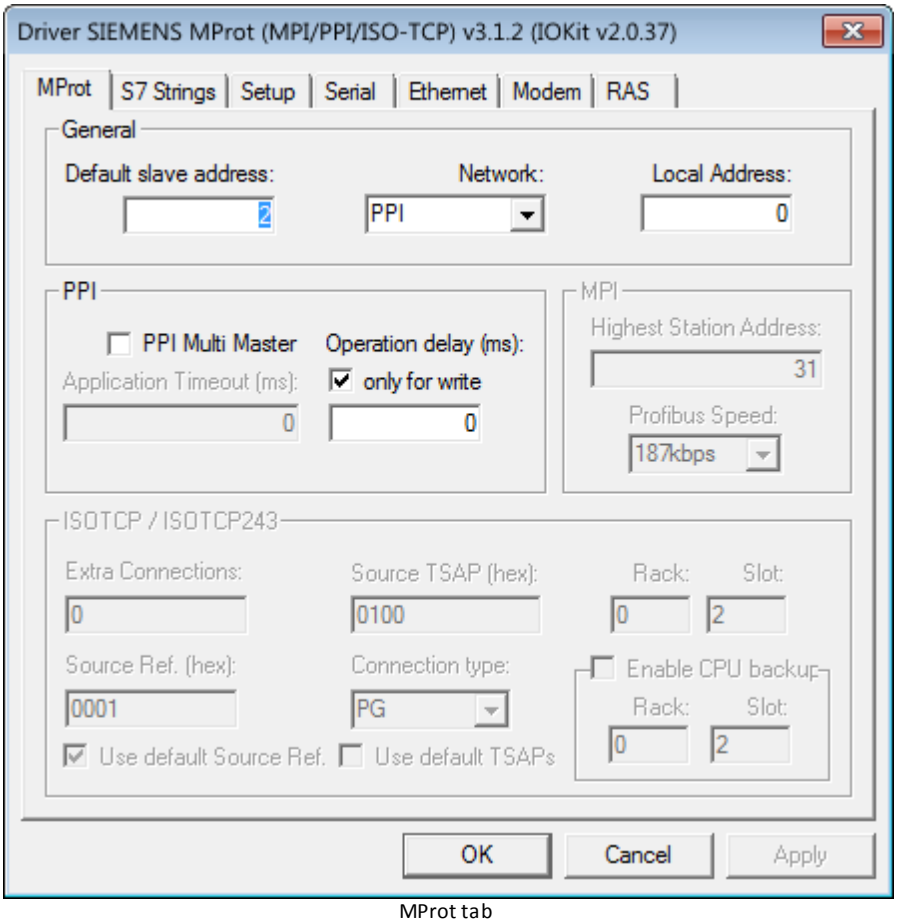
This Driver does not support using Siemens PPI/MPI adapters via USB interface.

This Driver does not support CP5611 or similar interfaces to access an MPI network. Use the S7Functions or Siemens SIMATIC.NET Drivers to communicate with these boards, by using the included OPC Server.

NOTE: M-Prot is a name created by Elipse Software to specify a Driver that supports multiple protocols. There is no relationship whatsoever with device names, protocols, or standards defined by the aforementioned manufacturers.

Driver Settings

The Driver's [P] parameters for configuration are not used. All configurations are performed on Driver's configuration window, shown on the next figure.



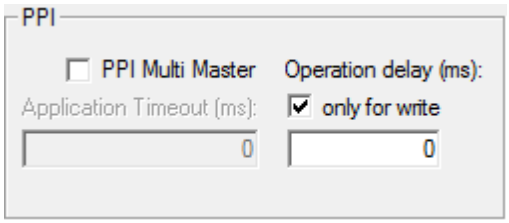
MProt tab

The available options for the **General** group are described on the next table.

Available options on the General group

OPTION	DESCRIPTION
Default Slave Address	This can be used as the default address for any Tag, by leaving the *N1* parameter in 0 (zero) so that it is replaced by the default address.
Network	Protocol selection: **PPI**, **MPI**, **ISOTCP**, **ISOTCP for CPU 243**, or **MPI for IBHLink converter**. The **PPI**, **MPI**, or **ISOTCP/ISOTCP243** groups on this tab are enabled or disabled according to the selected protocol.
Local Address	Driver's address on the network. It can be selected arbitrarily.

The available options for the **PPI** group are described on the next table.



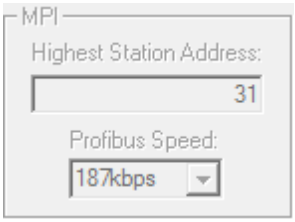
PPI group

Available options on the PPI group

OPTION	DESCRIPTION
PPI Multi Master	Informs the Driver that there are other Masters on the network.
Application Timeout (ms)	Maximum communication time for each Tag, in milliseconds. Available only when it is a multimaster.
Operation delay (ms)	Stop time for an interval between communication operations, in milliseconds. Select the **only for write** option to indicate the application of this interval only for writing operations (please check the next note).

NOTE: The **Operation delay** option adds a minimum waiting time that must occur between the ending of a reading or writing operation and the beginning of the next one. Use a value different from 0 (zero) on this configuration only if facing communication failures caused by PLC's processing inertia. Writing operations are the most affected ones, because they are usually random. That is the purpose of the **only for write** option. If this option is not selected, the waiting time only applies to reading and writing operations. If it is selected, it only applies to writing operations (recommended). Notice that adding a waiting time may slow down application's performance.

The available options for the **MPI** group are described on the next table.

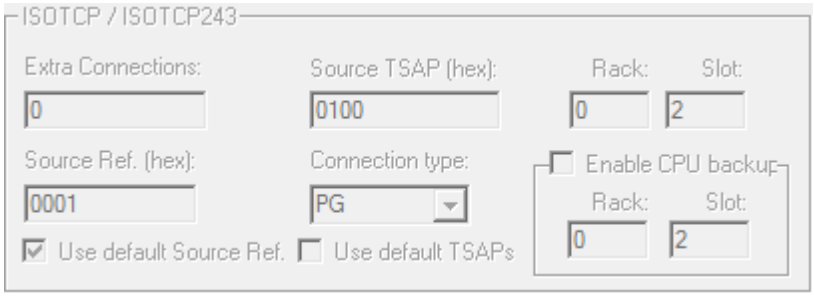


MPI group

Available options on the MPI group

OPTION	DESCRIPTION
Highest Station Address	Indicates the greatest available address on the network, so that in **PPI** and **MPI** modes the Driver discovers other possible Masters on the network. Only the **15**, **31**, or **63** options must be added.
Profibus Speed	Nominal speed of the Profibus network.

The available options for the **ISOTCP / ISOTCP243** group are described on the next table.



ISOTCP / ISOTCP243 group

Available options on the ISOTCP / ISOTCP243 group

OPTION	DESCRIPTION
Extra Connections	Number of additional TCP connections that can be created to improve communication performance.
Source Ref. (hex)	A number formed by a **Word** in hexadecimal that identifies the protocol's source reference. It is only enabled when the **Use default Source Ref** option is not selected.
Source TSAP (hex)	A number formed by a **Word** in hexadecimal that identifies the protocol's local TSAP. It is only enabled when the **Use default TSAPs** option is not selected.
Connection type	Connection type: **PG**, **OP**, or **PC**. It must be selected according to CPU configuration.
Rack	Destination CPU's rack.
Slot	Destination CPU's slot.
Enable CPU backup	Enables typing rack and slot values of the backup CPU, for use in redundancy systems that have different values from the main CPU.

For this Driver's communication to work with the Siemens S7-1200 PLC series, users must select the **ISOTCP** option, deselect the **Use default TSAPs** option, configure the **Source TSAP (hex)** property to "0100", and define the **Connection type** option as "PG", **Rack** with 0 (zero), and **Slot** with 1 (one).

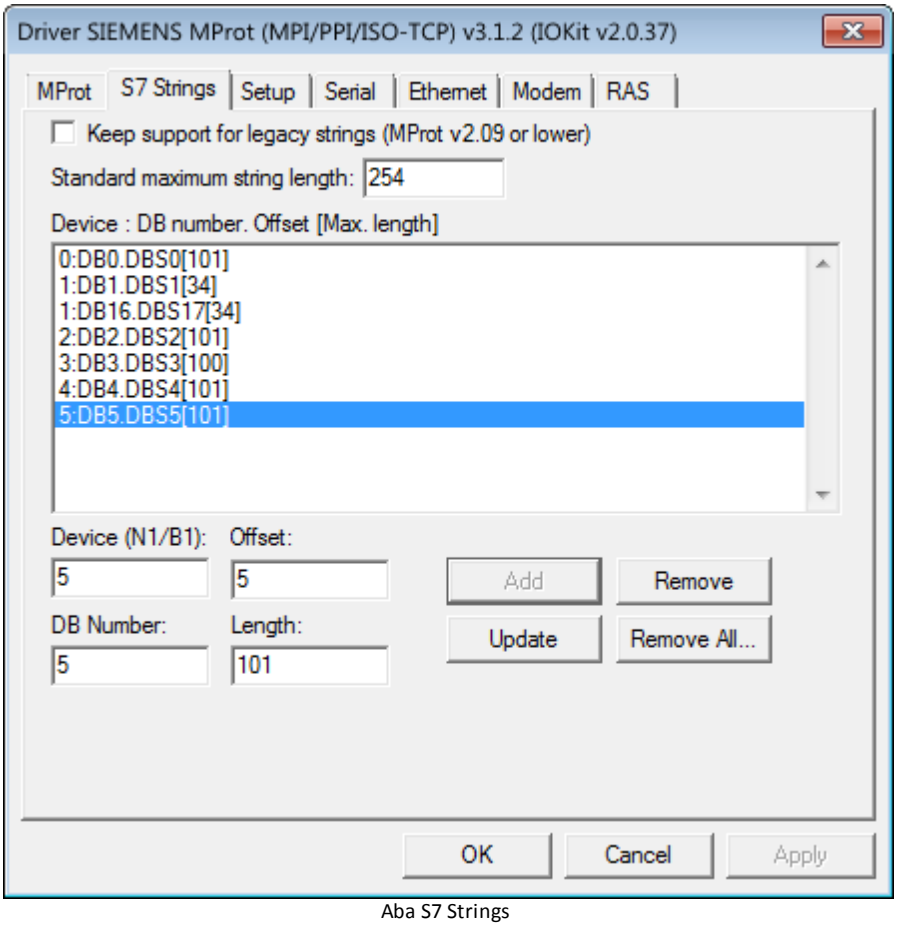
NOTES:

- When selecting the **ISOTCP** or **ISOTCP243** protocols, all Tags in the Driver object must have the *N1* (or *B1*) parameter in 0 (zero) and the **Default Slave Address** parameter also in 0 (zero).
- The **Source Ref** and **Source TSAP** parameters must only be used in very specific cases. Due to successful executions in a wide range of topologies, it is strongly recommended to keep the **Use default Source Ref** always selected and **Source TSAP** value always as "0100".
- When the **Use Default TSAPs** option is selected together with the **ISOTCP** protocol, the **Source TSAP** value is "0100" and the **Destination TSAP** value is "0202".
- **TSAP** stands for *Transport Service Access Point*, which is a terminology used by the ISO protocol.
- When using PC - PPI/MPI serial adapters, it is very common the need to configure the handshaking on the **Serial** tab of Driver's configuration window. Only the RTS control must be configured to **ON**. If there is any unsuccessful communication during this Driver's initial tests, it is advisable to try that change (**RTS Control** configured to **ON**) and run the test again.

Configuration Parameters for Strings

This tab is useful only if users need to declare **Strings** with a defined maximum length, individually or

generically.



Aba S7 Strings

The available options on the **S7 Strings** tab are described on the next table.

Available options on the S7 Strings tab

OPTION	DESCRIPTION
Keep support for legacy strings	Keeps support for old **Strings**, prior to version 2.10. By selecting this option, the old **String** format implemented on prior versions is kept, avoiding problems when updating Driver's version. It is advisable to select this option only when migrating a project whose Driver's version is 2.09 or earlier. If the project uses **Strings** after performing a version update, **String**-type Tags return reading errors from the PLC. The legacy **String** format contains a 32-byte reserved space starting from the configured offset. When working with a brand new project, leave this option deselected.
Standard maximum string length	Standard maximum length of **Strings**. Fill it in with a default value configured in the PLC memory for **Strings** without a declared maximum length. For example, in S7-200 PLCs this value is equal to "254". This means that requests for **Strings** with undeclared lengths contain and indicate a fixed length of 254 characters.

List of Strings' maximum lengths

This tab also displays a selectable list with declared **Strings** with pre-determined lengths. This list appears empty if there are no configured **Strings**. These **Strings** can be declared in the PLC memory in two ways:

- Without specifying a maximum length on declaration. Example:

```
STRING var;
```

The **String** is allocated automatically with PLC's standard maximum length.

- By specifying a maximum length on declaration. Example:

```
STRING var[50];
```

On the previous example, the **String** is allocated with a maximum length of "50". Due to that second form that this list of **String** lengths is so important.

To determine the length of a new declared **String**, users must fill in all fields correctly, as described on the next table.

Available options to configure Strings' maximum length

FIELD	DESCRIPTION
Device	PLC address. Fill it in with the same value of Tag's *N1/B1* parameter (please check the topic **Standard Addressing**).
DB Number	Type the value of the DB number where the **String** is located.
Offset	Type the value of the DB offset where the **String** is located.
Length	Type the maximum length value of the **String**, as declared in the PLC programming.

In case there is already a **String** declared on the list with the same value for **Device**, **DB Number**, and **Offset**, that one is automatically selected on the table and its values are loaded to all edit fields.

Three options help users when editing **String** data on the list:

- **Add**: Adds new parameters
- **Update**: Changes parameters already listed
- **Remove**: Completely removes a row of parameters

Click **OK** to confirm all configurations listed and close this window. Click **Remove All** to remove all data on this list.

NOTE: When choosing to declare Tags with Symbolic Addressing parameters, there is no need to fill in this list with **Strings** declarations. The length can be specified on the symbol parameter available in the Tag.

Tags Reference

This section contains information about the configuration of Tags by **Symbolic Addressing** and by **Standard Addressing** (*N/B* parameters). It also contains references to the **Extra ISOTCP Connections Interface Tags**.

Configuration by Syntactical Parameters

Use the following syntax for each field in E3 or Elipse Power:

- **Device**: Insert the device's address on the network. If it is equal to 0 (zero) and the selected protocol is different from **ISOTCP** or **ISOTCP243**, then it is replaced by the **Default Slave Address**. If the selected protocol is **ISOTCP** or **ISOTCP243**, this value must be left as 0 (zero). The **Device** field may also be left blank, as long as it is inserted in the **Item** field before the colon symbol.
- **Item**: This field must obey one of the defined syntaxes described next.

Use the following general syntax, if area is not equal to **DB**. Values inside brackets are optional:

```
<[Device:]><Area><[Type]><Address>[.Bit]
```

Where:

- **Device**: PLC address as exposed in the **Device** item, if it was not informed in that field.
- **Area**: Data area inside the PLC. The following options can be used:
  - **S**
  - **SM**
  - **AI** (*Analog Input*)
  - **AQ** (*Analog Output*)
  - **C** (*Counter*)
  - **T** (*Timer*)
  - **I** (*Digital Input*)
  - **Q** (*Digital Output*)
  - **M** (*Memory*)
  - **HC** (*High Speed Counter*)
- **Type**: Data type to read. The next table shows all possible symbols for these types.

Available options for types

TYPE	MEANING
X	Used to extract a bit from a byte
B	**Byte**
W	**Word**
D	**DWord**

TYPE	MEANING
F	**Float**
S	**String**
S5T	**S5Time**

- **Address**: Numerical address to read.
- **Bit**: Optional that informs the bit of a word to read or write (between 0 and 31).

Example:

```
(PLC 4, bit 1 of memory at address 10)
Device: Blank - Item 4:M10.1
```

If area is equal to **DB** (also called **V**), use the following syntax. Values inside brackets are optional:

```
<[Device:]>DB<DBNumber>:<Type><Address><[.Bit]>
```

- **Device**: Refers to the same optional item of the general syntax.
- **DBNumber**: Fill it in with the DB number. If the memory contains a single or unspecified DB block, fill it in with value 1 (one).
- **Address**: Numerical address (offset) to read.
- **Bit**: Optional value that informs the bit of a type to read or write (between 0 and 31).

Available options for types on DB Area

TYPE	MEANING
DBX	Used to extract a bit from a byte in a DB
DBB	Used to read or write a byte in a DB
DBW or **DW**	Used to read or write a **Word** in a DB
DBD or **DD**	Used to read or write a **Double Word** in a DB
DBF or **DF**	Used to read or write a **Floating Point** (32-bit real) in a DB
DBS or **DS**	Used to access a **String** in a DB
DBS5T	Used to access an **S5Time**-type timer in a DB

Examples:

```
(PLC 2, Word starting at address 20 of DB1)
Device: 2 - Item: DB1:DW20
(Same as the previous one, but Device was informed in the Item field)
Device: Blank - Item: 2:DB1:DW20
(PLC 7, DB 5, bit 2 of byte 7)
Device: Blank - Item: 7:DB5:DBX7.2
```

The syntax for **String** types in the DB area is the following:

```
<[Device:]>DB<DBNumber>:DBS<Address><[Maximum length]>
```

Where:

- **Device, DBNumber, and Address**: Refer to the same items of the general syntax.
- **Maximum length**: Optional that informs the maximum length declared on the **String**. If it is not informed, then it considers the maximum default length of the **String** as informed on the **Strings** configuration window.

Examples of syntax for **Strings**:

```
(PLC 2, String starting at address 16 of DB17,
using the PLC's maximum default length)
Device: 2 - Item: DB17:DBS16
(same as the previous one, but Device was informed in the Item field
and with a maximum allocated length of 25 characters)
Device: Blank - Item: 2:DB17:DBS16[25]
(PLC 4, String starting at address 100 of DB10,
with a maximum allocated length of 50 characters)
Device: Blank - Item 4:DB10:DS100[50]
```

Configuration by Numerical Parameters (N/B)

Use the default syntax described on the next table for all Tags and Blocks.

Default syntax for Tags and Blocks

PARAMETER	DESCRIPTION
N1/B1	PLC address. If it is equal to 0 (zero) and the selected protocol is different from **ISOTCP** or **ISOTCP243**, then it is replaced by the **Default Slave Address**. If the selected protocol is **ISOTCP** or **ISOTCP243**, this value must be left as 0 (zero).
N2/B2	Data type and Area (please check the next tables). This value must be composed by the data type multiplied by 100 plus the area (the formula is **N2/B2 = DataType × 100 + Area**).
N3/B3	If the selected area is **V (DB)**, fill it in with the number of the DB block. Otherwise, leave it in 0 (zero). If the memory contains a single or unspecified DB block, fill it in with the value 1 (one).
N4/B4	DB block's address in the area or offset. To use data types that require more than one byte, use addresses that are multiples of two for two-byte types (signed or unsigned 16-bit) and multiples of four for four-byte types (signed or unsigned 32-bit and 32-bit floating point).

Available options for Data types

TYPE	MEANING
0	Area's default
1	BOOL (Boolean)
2	BYTE (unsigned 8-bit)

TYPE	MEANING
3	WORD (unsigned 16-bit)
4	INT (signed 16-bit)
5	DWORD (signed 32-bit)
6	DINT (signed 32-bit)
7	REAL (32-bit floating point - IEEE 754)
8	STRING (please check the **note** further on this topic)
12	S5TIME (time in seconds, 32-bit floating point - IEEE 754, please check the **note** further on this topic)

Available options for Areas

AREA	MEANING
0	S
1	SM
2	AI (Analog Input)
3	AQ (Analog Output)
4	C (Counter)
5	T (Timer)
6	I (Digital Input)
7	Q (Digital Output)
8	M (Memory)
9	V (DB)
10	HC (High Speed Counter)

NOTES:

- For **S5Time**-type data, the value to be filled in is always in seconds, as a 32-bit floating point. The range of values different from zero is between 0.01 and 9990.0 seconds. The time base is filled in or interpreted automatically.
- In the PPI protocol there is a limitation in the I/O Block for data in bytes. For reading, the maximum allowed is 224 bytes, and for writing it is 218 bytes. This means, respectively, that for **Word**-type data (16 bits), a Block cannot have more than 112 and 109 Elements. For **DWord**-type data (32 bits), a Block cannot have more than 56 and 54 Elements, and so on.
- If the Rack and Slot definition is unknown for Tag addressing in the ISOTCP protocol, please check the article *KB-39019: Rack and Slot settings*, on Elipse Knowledgebase.

Interface Tags on Extra ISOTCP Connections

By opting to use extra ISOTCP connections with the **Extra Connections** parameter on the **Driver's configuration window**, these connections can be controlled and monitored by three Interface-specific Tags: **Physical Layer Status**, **IPSelect**, and **IPSwitch**.

NOTE: These Tags cannot be used when the **Extra Connections** parameter is 0 (zero). In this case, use the corresponding IOKit Tags, with the same name, whose usage can be checked on **IOKit User's Manual**.

Physical Layer Status (MProt)

Read-Only

Configuration by numerical parameters

PARAMETER	VALUE
N1	-2
N2	0 (zero)
N3	0 (zero)
N4	2

Configuration by syntactical parameters

PARAMETER	VALUE
Item	MProt.IO.PhysicalLayerStatus

This Tag indicates the status of the physical layer connection. Its possible values are the following:

- **0**: Physical layer disconnected
- **1**: Physical layer connected

IPSelect (MProt)

Read and Write

Configuration by numerical parameters

PARAMETER	VALUE
N1	-2
N2	0 (zero)
N3	4
N4	0

Configuration by syntactical parameters

PARAMETER	VALUE
Item	MProt.IO.Ethernet.IPSelect

Indicates the active IP. Its possible values are the following:

- **0**: The main IP is selected (active)
- **1**: The alternative IP (backup) is selected (active)

If the Ethernet interface is connected, this Tag indicates which one of the two configured IPs is in use. If the interface is disconnected, this Tag indicates which IP is used first in the next connection attempt.

During the connection process, if the active IP is not available, then IOKit tries connection to the other IP. If the connection to the alternative IP succeeds, then this IP is set as the active one (automatic switchover).

To force a manual switchover, write 1 (one) or 0 (zero) to this Tag. This forces a reconnection with the specified IP (**0**: Main IP and **1**: Backup IP) if the Driver is currently connected. If the Driver is disconnected, that configures the active IP for the next connection attempt.

IPSwitch (MProt)

Write-Only

Configuration by numerical parameters

PARAMETER	VALUE
N1	-2
N2	0 (zero)
N3	4
N4	1

Configuration by syntactical parameters

PARAMETER	VALUE
Item	MProt.IO.Ethernet.IPSwitch

Writing any value to this Tag forces a manual switchover. If the main IP is active, then the backup IP is activated, and vice versa. This forces a reconnection with the specified IP if the Driver is currently connected. If the Driver is disconnected, that configures the active IP for the next connection attempt.

SOE Collecting

This section contains specific information about SOE's event collecting.

Preparing for SOE Collecting

Before using the SOE Collecting Tags, users must prepare the PLC by creating a DB Table (**V** area) and develop a programmable logic compatible with all SOE collecting procedures developed for this Driver.

Table of SOE Events

This table aims to configure the size of the event buffer and manage their input and output in a circular buffer routine. This table is constantly updated by both the PLC and the Siemens MProt Driver.

The Table of SOE Events must contain registers on control and storage of events, based on the data structure described on the next table.

Data structure

ADDRESS	DESCRIPTION	DATA TYPE
0.0		STRUCT
+0.0	Table Status	WORD (unsigned 16-bit)
+2.0	Recording Pointer	WORD (unsigned 16-bit)
+4.0	Acquisition Status	WORD (unsigned 16-bit)
+6.0	Maximum Limit of Items of the Circular Buffer	WORD (unsigned 16-bit)
+8.0	Circular Buffer	ARRAY[1..*n*] (limit of user-defined items)
+0.0		STRUCT
+0.0	TIMESTAMP_LOLO (Year)	WORD (unsigned 16-bit)
+2.0	TIMESTAMP_LOHI (Day and Month)	WORD (unsigned 16-bit)

ADDRESS	DESCRIPTION	DATA TYPE
+4.0	TIMESTAMP_HILO (Hour and Minute)	WORD (unsigned 16-bit)
+6.0	TIMESTAMP_HIHI (Second and Millisecond)	WORD (unsigned 16-bit)
+8.0	Value of Event Type 1	Event's data type (user-defined)
+n.0	Value of Event Type 2	Repeats the same data type
+n.0	Value of Event Type 3	Repeats the same data type
+n.0	Value of Event Type *n*	Repeats the same data type
=n.0		END_STRUCT
=n.0		END_STRUCT

Description of control registers of events

- **Table Status**: It must be kept exclusively by the PLC, indicating the number of events available for reading in the circular buffer. It must be updated by the PLC whenever new events are added to the circular buffer, or after completing the collecting of events by the application, which can be detected when **Acquisition Status** changes.

- **Recording Pointer**: It must be kept exclusively by the PLC, indicating the index, starting at zero, of the position where the next event must be inserted. The index must be incremented by the PLC whenever a new event is inserted in the circular buffer, then returning to index zero after reaching the maximum limit of the circular buffer.

- **Acquisition Status**: It must be kept by the PLC and by the MProt Driver, indicating the number of records already read at every transaction. After each collecting, the MProt Driver writes to this rerister the number of events that it could read. When detecting this change, the PLC must immediately subtract this value written by the MProt Driver from the **Table Status** and then reset the **Acquisition Status**.

- **Maximum Limit of Items of the Circular Buffer**: A constant value that specifies the maximum limit of events to store in the circular buffer before the pointer moves back to index 0 (zero). It must contain exactly the limit value of the Array resized for events of the circular buffer.

Description of storage registers of events

- **TIMESTAMP**: Time when the event occurred.

- **Event Value**: Value of the occurred event, which can be composed by one or *n* values (all with the same data type), in which they are grouped together for the same **TIMESTAMP** generated when an event occurs.

TIMESTAMP format

The **TIMESTAMP** is represented by four **WORDs**, according to the data structure described on the next table.

Data structure

WORD	CONTENT	RANGE
0	Year	Between 0 and 65535
1	Day and Month	ddddddddmmmmmmmm
2	Hour and Minute	hhhhhhhhmmmmmmmm

WORD	CONTENT	RANGE
3	Seconds and Milliseconds	ssssssmmmmmmmmm

- The first **Word** contains an integer value for a year.
- The second **Word** is divided into a high byte to represent a day and into a low byte to represent a month.
- The third **Word** is divided into a high byte to represent hours and into a low byte to represent minutes.
- The fourth **Word** uses the six highest bits to represent seconds and the 10 lowest bits to represent milliseconds.

Acquisition Procedure

The PLC must start inserting events in ascending order, starting from table's base address, referring to the beginning of the circular buffer. At every new event inserted, the recording pointer must be incremented, starting to point to the next buffer's available address.

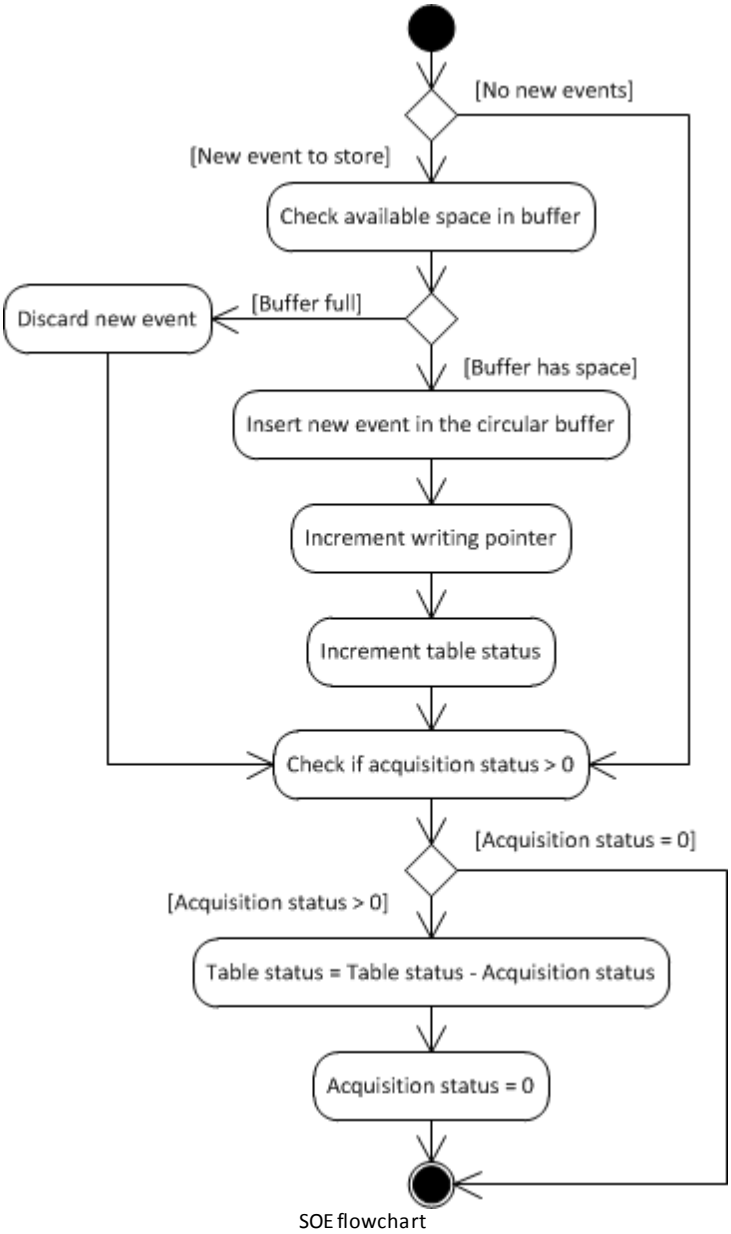
The Driver performs an event reading from the oldest to the newest. The starting address for reading is calculated by the Driver using the value of **Recording Pointer** and **Table Status**.

If the number of available events is greater than the maximum allowed in a single communication frame of the protocol, the Driver performs multiple block readings, updating the value of **Acquisition Status** at the end of the process with the total amount of events read.

When detecting that the Driver wrote a value greater than 0 (zero) to **Acquisition Status**, the PLC must immediately subtract the value of **Acquisition Status** from the value of **Table Status** and then reset **Acquisition Status**.

The PLC can insert new events on the table during the PLC's acquisition process, as long as there is no overflow in the circular buffer, then incrementing **Table Status**.

The next figure presents a flowchart, as a UML Activity Diagram, with a suggested implementation for this PLC logic.



SOE flowchart

SOE Collecting Tags

The SOE collecting of events is performed by using the Tags described next, by using an ISOTCP communication with the PLC.

Block Tag for Control Register (Read only)

- **B1**: 0 (zero)
- **B2**: 309 (Data Type = 3 and Area = 9)
- **B3**: Number of the DB block. If the memory contains a single or unspecified DB block, fill it in with value 1 (one)
- **B4**: Not used

The Block Tag to query Control Registers must contain four Elements to return the following values:

- **Element 1**: Table Status

- **Element 2**: Recording Pointer
- **Element 3**: Acquisition Status
- **Element 4**: Maximum Limit of Items of the Circular Buffer

For a description of each one of these Control Registers, please check the topic **Preparing for SOE Collecting**.

Tag Block for Data Collecting (Read only)

- **B1**: 0 (zero)
- **B2**: Data Type and Area = 90
- **B3**: Number of the DB block. If the memory contains a single or unspecified DB block, fill it in with value 1 (one)
- **B4**: Not used

The Block Tag for Data Collecting must contain a number of Elements corresponding to the number of values of *n*-event type that compose a single event. If this event is composed of a single value, resize the Block Tag for Data Collecting with a single Element. If this event is composed by two values, the Block Tag must be resized to two Elements, and so on. Use Block Tag's *B2* parameter to indicate a data type associated to event values.

NOTE: All values that compose an event must have the same data type, as well as every PLC's DB table must be filled in with the same event type.

Driver Revision History

VERSION	DATE	AUTHOR	COMMENTS
3.1.2	05/12/2015	M. Ludwig	• Fixed a denial on the option to select ISOTCP243 protocol (*Case 18675*).
3.1.1	09/19/2014	M. Ludwig	• Implemented CPU redundancy (automatic selection of backup CPU, alternative Rack/Slot, with connection to the backup IP, *Case 15782*). • Implemented configuration of Rack, Slot, and connection type on Driver's properties window (*Case 15911*). • Added Interface-specific Tags for the extra connections option (*Case 17221*).

VERSION	DATE	AUTHOR	COMMENTS
3.00	12/20/2103	M. Salvador M. Ludwig	• Implemented internal Superblocks in extra TCP connections (Case 14025). • Driver ported to IOKit 2.00 (Case 14019).
2.13	08/21/2012	M. Ludwig	• Implemented the **PDU REF** field functionality in ISOTCP protocol (Case 13299).
2.12	05/30/2012	C. Mello	• Added support for SOE Collecting of events in DB tables (Case 12483).
2.11	08/04/2011	M. Ludwig	• Included a consistency according to the MPI protocol and code improvements (Case 12392). • Added information about support for PLC Siemens S7-1200 series (Case 12292).
2.10	03/25/2011	M. Ludwig	• Implemented the **S7 String** format and a new properties window to configure **Strings** (Case 12005).
2.09	08/25/2009	M. Ludwig	• Fixed a bug when reading **Counter**-type variables (Case 10701). • Implemented advanced configurations for ISOTCP / ISOTCP243 (Case 10717).
2.08	06/19/2009	M. Ludwig	• Fixed a bug in a disconnection addressing multiple slaves in the MPI protocol (Case 10595).
2.07	06/03/2009	M. Ludwig	• Implemented the **S5Time** data type (Case 10413).
2.06	01/07/2009	M. Ludwig	• Fixed a connection failure under ISOTCP protocol (Case 10138).
2.05	11/04/2008	M. Ludwig	• Improvements on properties window layout (Case 9994). • Implemented an operation delay in PPI (Case 9968).

VERSION	DATE	AUTHOR	COMMENTS
2.04	04/01/2008	M. Ludwig	• Fixed a problem when addressing analog inputs and outputs combined with the **EnableReadGrouping** property configured to True (Case 8927). • Improvements and consistencies to avoid PLC's disconnection problems, as described on case 8968 (receiving random values in alarm variables in ISOTCP). • Fixed an unhandled exception when receiving NAK characters in MPI protocol, which caused a lock on data reception (Case 8981). • Improvements on consistency of MPI protocol reception (Case 8981). • Removed an unnecessary byte in the frame, which caused problems when writing bytes and bits under the ISOTCP protocol and the S7-400 PLC (Case 9021). • Fixed a bug in the automatic reconnection after a physical disconnection in ISOTCP (Case 9030). • Fixed the implementation of a long ACK frame reception in PPI (Case 9118). • Implemented a condition of unavailable data in PPI. When this condition is met, returns an empty list and **OK** instead of a failure (Case 9232). • Fixed a wrong attribution of Service Access Point in MPI protocol, which caused communication failures with Tecnatron adapters (Case 9238).

VERSION	DATE	AUTHOR	COMMENTS
2.03	09/13/2007	M. Ludwig	• Fixed a reconnection problem with serial adapters when the PLC is turned off (Case 8069). • Implemented addressing to multiple slaves in MPI protocol (Case 8625). • Ethernet port freely configurable (Case 8683). • Driver compiled with IOKitLib v1.14 to fix reading and writing errors before the first connection (Case 7614). • Documentation updated with information about the length of **Strings**, protocols, and compatible devices (Case 8206).
2.02	03/28/2007	M. Ludwig	• Fixed the lack of creating a blob, which caused run-time errors (Case 8015). • Fixed a problem of switching IP numbers at run time (Case 8026). • Developed support for Windows CE (Case 7504). • Added support to IBHLink converters (Case 7994). • Fixed a writing problem with **Strings** (Case 7967).
2.01	07/10/2006	M. Ludwig	• Fixed parsing of DB variables (Case 7172).
2.00	04/13/2006	M. Salvador M. Ludwig	• Fixed a failure in PPI protocol *Error: Single DLE in data field* (Case 6644). • Removed address check. Regardless of data type, any value in *N4* is allowed (Case 6644). • Fixed a bug in the configuration interface, where IBHLink and ISOTCP configurations were mixed (forcing port 1099 instead of port 102, Case 6644). • Added support for Superblocks and symbolic addressing (Case 6644).

VERSION	DATE	AUTHOR	COMMENTS
1.01	11/03/2005	M. Ludwig	• Optimization, standardization, and source code review.
1.00	05/01/2005	M. Salvador	• Original version of the Driver.

Headquarters
Rua 24 de Outubro, 353 - 10º andar
90510-002 Porto Alegre
Phone: (+55 51) 3346-4699
Fax: (+55 51) 3222-6226
E-mail: elipse-rs@elipse.com.br

USA
2501 Blue Ridge Road, Suite 250
Raleigh - NC - 27607 USA
Phone: (+1 252) 995-6885
Fax: (+1 252) 995-5686
E-mail: support@elipse.com.br

Taiwan
9F., No.12, Beiping 2nd St., Sanmin Dist.
807 Kaohsiung City - Taiwan
Phone: (+886 7) 323-8468
Fax: (+886 7) 323-9656
E-mail: evan@elipse.com.br

Check our website for information about a representative in your country.

www.elipse.com.br

kb.elipse.com.br

elipse@elipse.com.br

Microsoft Partner
Gold Independent Software Vendor (ISV)

OPC FOUNDATION MEMBER