



V19.4.5  
Drivers

eScada.Drivers.OmronEIP

## eScada.Drivers.OmronEIP

( Ethernet IP – Connected CIP Transport )

### OS availability

Windows, Linux, RaspBian

### Atomic data type

Following CIP specifications for implemented data types.

### Hardware and documentation reference

[www.ia.omron.com](http://www.ia.omron.com)

[www.odva.org](http://www.odva.org)

### Parameters available in every section

Channel:	none	
Device:	IP address	It can be IPV4 Multiple addresses can be expressed separated using , (comma) e.g. 192.168.1.10,192.168.1.11
	TCP Port	A valid TCP port number.
	Session Serial Number	It shall be a unique number for the connected device
	EIP Mode	0=Connected, 1=Unconnected
	Request Packet Interval	It must be greater than the maximum polling time
	Reconnect timeout [ms]	Waiting time before a reconnection after COMM break-down
Group:	none	
Tag:	none	

### Remarks for devices

The following attributes can be expressed for every device.

Bytes order actions	None, Swap bytes (little endians ↔ big endians adjustment)
String actions	None, Swap bytes in words

### Implemented data types

PLC data type		Single element	HMI Array
BOOL	single bit	Yes	Yes
SINT	8 bit	Yes	Yes
INT	16 bit	Yes	Yes
DINT	32 bit	Yes	Yes
LINT	64 bit	Yes	Yes
USINT	8 bit	Yes	Yes
UINT	16 bit	Yes	Yes
UDINT	32 bit	Yes	Yes
ULINT	64 bit	Yes	Yes
REAL	floating point 32 bit	Yes	Yes
LREAL	floating point 64 bit	Yes	Yes
BYTE	8 bit	Yes	Yes
WORD	16 bit	Yes	Yes
DWORD	32 bit	Yes	Yes
LWORD	64 bit	Yes	Yes
STRING	1 byte per character	Yes	No

## Addressing

You can address every variable with a basic data type, using its symbol name.

Basic data in a user defined structure can be addressed.

Single item belonging to an array can be addressed using its index within square brackets.

### Examples

variable      myVariable

structure     structure.element.data – libab\_TIMERS[0].PRE – libab\_COUNTERS[1].CU

item array    myVariable[2] – srtucture.element[0]

Variable type	Type	PLC type	Items
<b>Boolean</b> The number of items used declaring TAGs, must be multiple of source PLC data size. Every group of booleans, must start from the first bit.			
Single bit	Bit	every numeric data type, BOOL	(C)
<b>Byte</b>			
Unsigned 8 bit	UInt8	every numeric data type with 8 bit	(C)
Signed 8 bit	Int8		
<b>16 bit</b>			
Unsigned integer 16 bit	UInt16	every numeric data type up to 16 bit	(C)
Signed integer 16 bit	Int16		
<b>32 bit</b>			
Unsigned integer 32 bit	UInt32	every numeric data type up to 32 bit	(C)
Signed integer 32 bit	Int32		
Single precision 32 bit - ( IEEE 754 )	Float		
<b>64 bit</b>			
Unsigned integer 64 bit	UInt64	every numeric data type up to 64 bit	(C)
Signed integer 64 bit	Int64		
Double precision 64 bit - ( IEEE 754 )	Double		
<b>Strings</b> The number of items used declaring TAGs, must be a multiple of source PLC data size String bytes can be interpreted as ASCII, UTF-7, UTF-8, UTF-16 or UTF-32 encoding			
Array of bytes	String	every numeric data type up to 64 bit	(A, C)
Array of bytes. (Siemens S7)	S7String	every numeric data type up to 64 bit	(B, C)
Array of bytes. (AllenBradley style)	ABString	every numeric data type up to 64 bit, STRING	
<b>(A)</b> It depends on the strings length: e.g. if you want to read strings with a length of 10 chars each string, you can set a number of items of $74 / 10 = 7$ consecutive items.			
<b>(B)</b> It depends on the strings length: e.g. if you want to read strings with a length of 10 chars each string, you can set a number of items of $74 / (10+2) = 6$ consecutive items.			
<b>(C)</b> It depends on PLC model. The best way is to try with the maximum items you need.			

**S7 strings format**

They have got two bytes at the beginning.

The first byte is for max allowed string length, the second one is for the real string length.

These types of strings can be declared with a length of 255 bytes max.

**AB Strings format**

They have got one word (16 bit) at the beginning, it contains the string length.

**Consecutive items**

The number of consecutive read/write items, depends on the PLC model.

Please have a look at 'Implemented data types' to understand which type of basic object can be addressed using array of items.