v24.2.0

Drivers

eScada.Drivers.AllenBradleyEIP

eScada.Drivers.AllenBradleyEIP

(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.rockwellautomation.com
www.odva.org
Communicating with RA products Using EtherNet/IP Explicit Messaging (Rev. 1.2)

Parameters available in every section

Channel:	none	
Device:	Node network	Ethernet/CNET DH+ (*)
	DH+ Channel	* Channel A, Channel B
	IP address	It can be IPV4 Multiple addresses can be expressed using multiple rows or a comma. e.g. 192.168.1.10,192.168.1.11
	TCP Port	A valid TCP port number.
	Node path	Destination node path. (only standard path 1,0 has been tested, even though other longer paths should work)
	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 re-connection after COMM break-down
Group:	none	
Tag:	Chunk mode	None, no chunks System, tries to use a default value for chunks size. Custom, permits to set a custom size for every chunk.
	Bytes per chunk	Only with custom mode Amount of bytes, admitted by the protocol, for each communication frame to get or set data. It depends on the protocol and device you are using, please refer to the protocol documentation. 0=No data chunks used.

Remarks for devices

The following attributes can be expressed for each device.

Bytes order actions	None, Swap bytes order, Swap bytes order in DWords, Swap words order, Swap bytes order in DWords then words order
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
REAL	floating point 32 bit	Yes	Yes
STRING	String of bytes Please use ABString as data type	Yes	Yes (using chunks)
COUNTER	structure		
.PRE	DINT, 32 bit	Yes	No
.ACC	DINT, 32 bit	Yes	No
.CU	BOOL, single bit	Yes	No
.CD	BOOL, single bit	Yes	No
.DN	BOOL, single bit	Yes	No
.OV	BOOL, single bit	Yes	No
.UN	BOOL, single bit	Yes	No
TIMER	structure		
.PRE	DINT, 32 bit	Yes	No
.ACC	DINT, 32 bit	Yes	No
.EN	BOOL, single bit	Yes	No
.TT	BOOL, single bit	Yes	No
.DN	BOOL, single bit	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]

remark:
In order to address an array variable, it is important to add the first array element you want to access at the end of the variable name. Otherwise you'll get a communication error.
e.g. myarray[0] is the correct way to express the tag address.

Variable type	**Type**	**PLC type**	**chunks**	**Items**
Boolean 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	BOOL, SINT, INT, DINT	NO	492
Byte				
Unsigned 8 bit	UInt8	SINT	YES	492
Signed 8 bit	Int8			
16 bit				
Unsigned integer 16 bit	UInt16	SINT, INT	YES	246
Signed integer 16 bit	Int16			
32 bit				
Unsigned integer 32 bit	UInt32	SINT, INT, DINT, REAL	YES	123
Signed integer 32 bit	Int32			
Single precision 32 bit - (IEEE 754)	Float			
64 bit				
Unsigned integer 64 bit	UInt64	SINT, INT, DINT, REAL	YES	61
Signed integer 64 bit	Int64			
Double precision 64 bit - (IEEE 754)	Double			
Strings The number of items used declaring TAGs, must be a multiples 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	SINT, INT	YES	A
Array of bytes. (Siemens S7)	S7String	SINT, INT	YES	A
Array of bytes. (AllenBradley style)	ABString	STRING	YES	B
A It depends on the string's length				
B Multiple items are admitted only using data chunks. Without using chunks, only one element can be treated.				

remark:
When using chunks, there are no limits on the amount of items.

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
This kind of string format permits to read and write single text defined as STRING data type into the PLC.
STRING Data type is a structure with these two elements: .DATA and .LEN

Consecutive items
The number of consecutive read/write items, depends on the PLC model.
Please review ‘Implemented data types’ to better understand which types of basic object can be addressed using array of items.