



eScada

v24.3.0  
Drivers

eScada.Drivers.SBusCom

## eScada.Drivers.SbusCom

( SAIA-Burges S-Bus serial protocol - Data mode )

### OS availability

Windows, Linux, RaspBian

### Atomic data type

Bit, 32 bit Word oriented protocol.

### Hardware and documentation reference

Document DDEPFW680 - Design Description of the new Data-Mode

Document DDPF6162 - S-Bus Telegrams description (reduced mode)

### Parameters available in every section

Channel:	COM Port	Serial port name depending on OS type. e.g. Linux: /dev/ttyS0, /dev/ttyUSB0 e.g. Windows: COM1, COM3
	Baud rate	Communication baud-rate, eg. 9600, 38400, 19200, etc.
	Reconnect timeout [ms]	Waiting time before a reconnection after COMM break-down
	Response timeout [ms]	Timeout interval used to wait for a response.
Device:	Station ID	Station ID number
	Protocol mode	Data mode (only)
	Read - Retry value.	Retry value before getting COMM error. (0=no retry)
	Write - Retry value.	Retry value before getting COMM error. (0=no retry)
Group:	none	
Tag:	none	

### 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

### Useful Linux commands

COM List:	<code>dmesg   grep tty</code>
COM rights:	<code>sudo chmod a+rw /dev/ttyUSB0</code>
COM user info:	<code>ls -l /dev/ttyUSB0</code>
COM add user:	<code>sudo adduser username dialout</code> - (dialout is the default group)

### Implemented I/O memory area

Digital Input	<b>I</b> x	1 Bit
Digital Output	<b>O</b> x	1 Bit
Digital Flag	<b>F</b> x	1 Bit
Registers	<b>R</b> x	32 bit
Counters	<b>C</b> x	32 bit
Timers	<b>T</b> x	32 bit
Data Block	<b>D</b> x	32 bit

x = Numeric address

## BCD values

If it's necessary to use BCD codification for integers values, please precede the value address with this character @

It is possible to use such kind of character with all integers format from 16 bits to 64 bits.

e.g. @R3 (value using BCD format)

## Addressing

Variable type	Type	Address type	Items
<b>Boolean</b>			
Single bit	Bit	I, O, F	128
<b>Byte</b> The number of items used declaring TAGs, must be a multiple of 4			
Unsigned 8 bit	UInt8	R, C, T, D	128
Signed 8 bit	Int8		
<b>16 bit</b> The number of items used declaring TAGs, must be a multiple of 2			
Unsigned integer 16 bit	UInt16	R, C, T, D	64
Signed integer 16 bit	Int16		
<b>32 bit</b>			
Unsigned integer 32 bit	UInt32	R, C, T, D	32
Signed integer 32 bit	Int32		
Single precision 32 bit - ( IEEE 754 )	Float		
<b>64 bit</b>			
Unsigned integer 64 bit	UInt64	R, C, T, D	16
Signed integer 64 bit	Int64		
Double precision 64 bit - ( IEEE 754 )	Double		
<b>Strings</b> The string length used declaring TAGs, must be a multiple of 4 String bytes can be interpreted as ASCII, UTF-7, UTF-8, UTF-16 or UTF-32 encoding			
Array of bytes	String	R, D	(A)
Array of bytes. (Siemens S7) Array of bytes. (AllenBradley style)	S7String ABString	Not implemented	(B)
(A) It depends on the strings length: e.g. if you want to read strings with a length of 12 chars each string, you can set a number of items of 128 / 12 = 10 consecutive items.			
(B) Not implemented			

## Consecutive items

The number of consecutive read/write items may depend on the PLC model.