v24.2.0

Drivers

eScada.Drivers.Kunbus

eScada.Drivers.Kunbus

(Direct access to IO memory area)

OS availability

RaspiOS

Atomic data type

Bit or byte oriented protocol

Hardware and documentation reference

https://revolution.kunbus.com/

Parameters available in every section

Channel:	none
Device:	none
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

Implemented data types

PLC data type		Single element	HMI Array
BOOL	single bit	Yes (using symbolic address)	Yes (using numeric offset address)
BYTE	8 bit	Yes	Yes
WORD16	16 bit	Yes	Yes
WORD32	32 bit	Yes	Yes
WORD64[1]	64 bit	Yes	Yes
FLOAT32[1]	32 bit	Yes	Yes
FLOAT64[1]	64 bit	Yes	Yes
STRING[1]	1 byte per character	Yes	Yes

[1] Floating point data type, 64 bit format and string elements are not directly supported by Kunbus hardware up to now.
Anyway you can use them with internal variables used as private parameters.

Addressing

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

Symbolic address must be used in order to address a single boolean variable.

Variable type	**Type**	**PLC type**	**Items**
Boolean The number of items used declaring TAGs, must be multiple of 8. Every group of booleans, must start from the first bit. Single bit can be addressed only using its symbolic address.			
Single bit	Bit	every numeric data type, BOOL	(C)
Byte			
Unsigned 8 bit	UInt8	every data type with 8 bit	(C)
Signed 8 bit	Int8		
16 bit			
Unsigned integer 16 bit	UInt16	every data type up to 16 bit	(C)
Signed integer 16 bit	Int16		
32 bit			
Unsigned integer 32 bit	UInt32	every data type up to 32 bit	(C)
Signed integer 32 bit	Int32		
Single precision 32 bit - (IEEE 754)	Float		
64 bit			
Unsigned integer 64 bit	UInt64	every data type up to 64 bit	(C)
Signed integer 64 bit	Int64		
Double precision 64 bit - (IEEE 754)	Double		
Strings 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 data type up to 64 bit	(A, C)
Array of bytes. (Siemens S7) Array of bytes. (AllenBradley style)	S7String ABString	every data type up to 64 bit	(B, C)
(A) 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) 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.			
(C) 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.