V19.4.5

Drivers

eScada.Drivers.ModbusRtu

eScada.Drivers.ModbusRtu

(for every device which support it)

OS availability
Windows, Linux, RaspBian

Atomic data type
Bit or 16 bit Word oriented protocol.

Hardware and documentation reference
http://www.modbus.org/
Document specification V1.1b3

Parameters available in every section

Channel:	Serial mode	0=System (default) 1=RS232 2=RS485
	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.
	Parity	N for none, E for even, O for odd
	Data bit	Allowed values are 5, 6, 7 and 8
	Data stop bit	Allowed values are 1 and 2
	Reconnect timeout [ms]	Waiting time before a re-connection after COMM break-down
	Response timeout [ms]	Timeout interval used to wait for a response.
	Byte timeout [usec]	Timeout interval between two consecutive bytes of the same message. -1 = disabled (DEFAULT)
Device:	Node ID	Slave ID number
	Addressing mode	Addressing mode: 0 based, 1 based
	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 every device.

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

Implemented Modbus codes

Modbus code	Address syntax	Tag mode
0x02 (read input status)	**DI**x	Read only
0x01 (read coil status) 0x0F (force multiple coils) 0x05 (force single coil)	**DO**x	Read Write
0x04 (read input registers)	**RI**x	Read only
0x03 (read holding registers) 0x10 (preset multiple registers) 0x06 (preset single register)	**RO**x	Read Write

Not implemented Modbus codes
Write and read data
0x17 (write/read registers).

Useful Linux commands

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

Addressing

Variable type	Type	Address type	Items
Boolean			
Single bit	Bit	DI, DO	2000
Byte The number of items used declaring TAGs, must be a multiple of 2			
Unsigned 8 bit	UInt8	RI, RO	250
Signed 8 bit	Int8		
16 bit			
Unsigned integer 16 bit	UInt16	RI, RO	125
Signed integer 16 bit	Int16		
32 bit			
Unsigned integer 32 bit	UInt32	RI, RO	62
Signed integer 32 bit	Int32		
Single precision 32 bit - (IEEE 754)	Float		
64 bit			
Unsigned integer 64 bit	UInt64	RI, RO	31
Signed integer 64 bit	Int64		
Double precision 64 bit - (IEEE 754)	Double		
Strings The string length used declaring TAGs, must be a multiple of 2 String bytes can be interpreted as ASCII, UTF-7, UTF-8, UTF-16 or UTF-32 encoding			
Array of bytes	String	RI, RO	(A)
Array of bytes. (Siemens S7) Array of bytes. (AllenBradley style)	S7String ABString	RI, RO	(B)
(A) It depends on the strings length: e.g. if you want to read strings with a length of 20 chars each string, you can set a number of items of 250 / 20 = 12 consecutive items.			
(B) It depends on the strings length: e.g. if you want to read strings with a length of 20 chars each string, you can set a number of items of 250 / (20+2) = 11 consecutive 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

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

Consecutive items

The number of consecutive read/write items could be different, because it depends on devices and other things.