SCHLEIFENBAUER

Modbus API

TECHNICAL DOCUMENTATION FOR PDU5.0 – ENERTREE

VERSION 1.01

Table of contents

Chapter 1: Modbus 2
Chapter 2: Enable Modbus 2
Chapter 3: Modbus functions 2
Chapter 4: Register list data mapping 2
Chapter 5: Obtain Device IDs 3
Chapter 6: Examples 3
6.1 Read the serial number 4
6.2 Read the module firmware version 4
6.3 Read the voltage from phase 2 4
6.4 Read the actual current from outlet 15 4
6.5 Read the total kWh 4
6.6 Read the temperature 5

Chapter 1: Modbus

The Modbus implementation uses Modbus TCP/IP to read values from PDU registers. This is a Modbus variant used for communications over TCP/IP networks, connecting over a port (default 502). It doesn't require a checksum calculation, as lower layers already provide checksum protection. Modbus provides the possibility to address all devices known in the device list.

Modbus communication is unsecured, make sure that the environment is suitable before enabling Modbus.

Chapter 2: Enable Modbus

The energy management system has a Modbus interface available to use which can be activated in the 'Modbus Server' entry in the Settings menu. In this menu, the Modbus TCP server can be activated to allow Modbus clients to connect. It's also possible to enable Modbus write operations to allow Modbus clients to write values to device registers (e.g., outlet control, naming). This configuration option is disabled by default. The TCP port of Modbus server can be set which defaults to 502 if unchanged.

Chapter 3: Modbus functions

Holding registers (0x03) are used only for values that can potentially be changed by the user (all registers with read/write access, such as names).

All measurements and fixed data (phase mappings, outlet measurements etc) are in input registers (0x04) instead.

Chapter 4: Register list data mapping

Modbus only supports 16-bit integer registers, and some conventions are used to represent larger data. 32-bit integers are transferred as 2 registers with big endian word order, most significant word first.

32-bit floats are transferred as 2 registers in IEEE floating point format. Strings are transferred as 2 bytes within each Modbus register. In case of an ASCII string, read 'size/2' registers from register starting point. Also all registers must be accessed together.

Data type (from register list)	**Modbus type**	**Description**
INT, 1	1 register, integer	
INT, 2	1 register, integer	
INT, 3	2 consecutive registers, 32-bit integer format	Both registers must be accessed together
INT, 4	2 consecutive registers, 32-bit integer format	Both registers must be accessed together
FD, 2	2 consecutive registers, 32-bit float format	Both registers must be accessed together
ASCII, even sized	(Size/2) registers, string format	Both registers must be accessed together

Depending on the size and number of bytes of the register, many registers have multiple 'channels' so that measurement values for example inlets and outlets can also be accessed individually.

Chapter 5: Obtain Device IDs

A Modbus Device ID is a number that identifies a specific device on a Modbus bus. It tells the Modbus master which device should receive the command and which device should respond.

To obtain the device id for a PDU, go to the PDU in the device list of the energy management system from which the data needs to be pulled. When the PDU is clicked, go to the configuration menu and then the ‘general setup’ section. The Device ID for SNMP and Modbus is displayed there which can be used for polling data with Modbus. This value is unique for each PDU.

Chapter 6: Examples

First create a valid connection, supply the correct IP address and port number and make sure that the connection mode is set for Modbus TCP. The other setting parameters are left to their default value. To test the connection, the ping icon can be clicked. Now create the actual connection.

For Windows there are examples included for the free Modbus Doctor software. This works a bit different than the Linus Mbpoll example. In Modbus Doctor the read action is performed and then interpreted. The Linux example reads the amount of registers based on the format selected. So for the serial number in Modbus Doctor two registers must be read, and only one in Mbpoll. But in Mbpoll we have selected the format to be a 32 bit value, so it will automatically read a pair of registers. For better understanding the basic command line options for Mbpoll are included, for a full list use ‘mbpoll -h’. These are examples, any Modbus client can be used.

Mbpoll cmd option	**Values**
-m	Mode, RTU or TCP
-t	Register format (16/32-bit int/float etcetera)
-a	Device address of PDU
-0	First reference is 0 instead of 1
-B	Big endian word order for 32-bit integer & float
-r	Start register
-c	Number of values to read

6.1 Read the serial number

Modbus Doctor (windows)

The serial number starts at register 227 as can be seen in the register list. There is 1 repeat and 4 bytes, 2 Modbus registers are needed for this value. To read the serial number: read 2 registers and interpret them as a 32-bit word (swap words) starting at register 227.

Mbpoll (linux)

mbpoll -m tcp -t 3:int -a 3 -0 -B -r 227 -c 1 192.168.9.139

6.2 Read the module firmware version

Modbus Doctor (windows)

The module firmware version starts at register 178 as can be seen in the register list. There are 8 register positions total, one repeat and 16 bytes, 2 Modbus registers per value. To retrieve the module firmware version: read 4 registers as 32-bit word starting at register 178.

An example of two values that will be received: 84234642 (52.0 in ASCII) and 11824 (0. in ASCII). To get the correct value, the ASCII needs to be flipped to get the correct “0.25.0”

Mbpoll (linux)

mbpoll -m tcp -t 3:int -a 3 -0 -r 178 -c 2 192.168.9.139

6.3 Read the voltage from phase 2

Modbus Doctor (windows)

The input voltage registers start at register 3180 as can be seen in the register list. There are 6 register positions total, 3 repeats and 4 bytes, 2 Modbus registers per value. To read the voltage from phase 2: read 2 registers and interpret them as a 32-bit float (swap words) starting at register 3182.

Mbpoll (linux)

mbpoll -m tcp -t 3:float -a 3 -B -0 -r 3182 -c 1 192.168.9.139

6.4 Read the actual current from outlet 15

Modbus Doctor (windows)

The outlet actual current registers start at register 6468 as can be seen in the register list. There are 120 register positions total, 60 repeats and 4 bytes, 2 Modbus registers per value). To read the current from outlet 15: read 2 registers and interpret them as a 32-bit float (swap words) starting at register 6496.

Mbpoll (linux)

mbpoll -m tcp -t 3:float -a 3 -B -0 -r 6496 -c 1 192.168.9.139

6.5 Read the total kWh

Modbus Doctor (windows)

The outlet actual current registers start at register 708 as can be seen in the register list. There are 2 register positions total, 1 repeats and 4 bytes, 2 Modbus registers per value). To read the total kWh: read 2 registers and interpret them as a 32-bit float (swap words) starting at register 708.

Mbpoll (linux)

mbpoll -m tcp -t 3:float -a 3 -B -0 -r 708 -c 1 192.168.9.139

6.6 Read the temperature

Modbus Doctor (windows)

The sensor value registers start at register 16232 as can be seen in the register list. There are 160 register positions total, 32 repeats and 4 bytes, 2 Modbus registers per value). To read the temperature in case of this PDU: read 2 registers and interpret them as a 32-bit float (swap words) starting at register 16234.

Mbpoll (linux)

mbpoll -m tcp -t 3:float -a 3 -B -0 -r 16234 -c 1 192.168.9.139

Modbus	Group	Code	Name	Unit	Datatype	Bytes	Repeats	Access	Definition
100	identificati	**iddate**	IdBirthday		string	16	1	ro	Date Of Manufacture (DOM)
108	identificati	**iddvlc**	IdDeviceLocation		string	64	1	rw	User configurable device location identifier.
140	identificati	**iddvnm**	IdDeviceName		string	64	1	rw	User configurable device name or identifier.
172	identificati	**idmaca**	IdMacAddress		int	1	6	ro	MAC address as 6-tuple of bytes.
178	identificati	**idmdfw**	IdModuleFirmware		string	16	1	ro	Firmware version number.
186	identificati	**idmdsn**	IdModuleSerialNumber		string	16	1	ro	Module serial number.
194	identificati	**idmdsw**	IdSoftwareVersion		string	16	1	ro	Software version number, only applicable for controller module
202	identificati	**idmodt**	IdModuleType		int	1	1	ro	Device category: 0 = Daisychain Module, 1 = Gateway Module, 2 = Controller Module (SF also uses -1 to meaning no module!)
203	identificati	**idmopn**	IdModulePartNo		string	16	1	ro	Module part number
211	identificati	**idonbr**	IdSalesOrderNumber		string	16	1	ro	Sales order number.
219	identificati	**idpart**	IdProduct		string	16	1	ro	Product id.
227	identificati	**idsnbr**	IdSerialNumber		int	4	1	ro	Serial number.
300	configurati	**cfamps**	ConfigMaximumLoad	A	int	1	1	ro	Maximum rated load of the device per phase (this value is also mentioned on the PDU profile), usually either 16A, 32A or 63A.
302	configurati	**cfcunm**	ConfigCustomerName		string	64	1	ro	Custom 64 character name field for the customer.
334	configurati	**cffofa**	ConfigFormFactor		string	16	1	ro	Formfactor of the device.
342	configurati	**cfleng**	ConfigModuleLength		int	2	1	ro	Length of the device in millimeter.
344	configurati	**cfnrbr**	ConfigNoBranches		int	1	1	ro	The number of branches available in the PDU, can be zero if no branches are available in the configuration.
346	configurati	**cfnrmo**	ConfigNrOutlMeasured		int	1	1	ro	Number of measured outlets available in the PDU. If numbering of outlets used is non-contiguous: the highest outlet number is assumed as amount of measured outlets.
350	configurati	**cfnrno**	ConfigNrOutletsTotal		int	1	1	ro	Total number of outlets available in the PDU, including hardwired outlets without a switchmeasure module.
352	configurati	**cfnrph**	ConfigNrPhases		int	1	1	ro	Total number of inputs: either zero (for no input metering), one or three for a single or three phase system
354	configurati	**cfnrso**	ConfigNrOutlSwitched		int	1	1	ro	Total number of switched outlets available. If the numbering of outlets used is non-contiguous: the highest outlet number is assumed as amount of switched outlets.
360	configurati	**cfvong**	ConfigVoltRating		string	16	1	ro	Voltage rating in V for this PDU.
500	daisychain	**dcbloc**	DChainBreakLocation		int	2	1	ro	Device index of the ring break location. Can be used to determine between which devices the ring is broken.
501	daisychain	**dcbsts**	DChainBusStatus		int	2	1	ro	SPBUS network architecture configuration. 0 = open ring network. 1 = closed ring network
502	daisychain	**dcnruf**	DChainNrUnitsFound		int	2	1	ro	Result of scan command, denotes the number of devices on the SPBUS network.
600	settings	**stcddt**	SetCurrentDropDetection		int	1	1	ro	Enable or disable current drop detection (0 = disabled, 1 = enabled)
601	settings	**stfodl**	SetFixedOutletDelay	milliseconds	int	2	1	ro	Minimal delay between relay switch requests in milliseconds. The minimum delay is 100ms.
602	settings	**sthilt**	SetHighlight		int	1	1	ro	Locate a unit. Set to '1' to switch on and to '0' to switch off this function.
603	settings	**stledb**	SetLedBrightness	percent	int	1	1	ro	LED brightness setting for the light ring of a PDU
604	settings	**stlock**	SetDisableOutletSwitching		int	1	1	ro	Setting this will disable all means of switching an outlet; only the powerup modes and delays will be respected (0 = disabled, 1 = enabled).
605	settings	**stopom**	SetOutletPowerupMode		int	1	1	ro	Behaviour of outlet on power-up. 0 = off. 1 = same state as at power down use default switch delay. 2 = same state, but delayed by individual delay timer. 3 = Outlets will always stay on (even at power down!)*. * this setting is only available for PDU's produced after 01-01-2018.
606	settings	**stsnsa**	SetSensorChangeAlertMode		int	1	1	ro	Enables the sensor channel change detection. 0 = off (default). 1 = on
607	settings	**stusbm**	SetConfigUsbMode		int	1	1	ro	USB Mode. 0 = Disabled, 1 = Enabled
608	settings	**stnfcm**	SetNFCMode		int	1	1	ro	NFC Activated. 0 = Disabled, 1 = Enabled
700	totals	**toaeng**	TotalApparentEnergy	VAh	float	4	1	ro	The total apparent energy in VAh. This value is not resetable.
702	totals	**toapwr**	TotalApparentPower	VA	float	4	1	ro	Totals VA (volt-amps, in VA), sum of phases
704	totals	**tocrac**	TotalActualCurrent	A	float	4	1	ro	System total actual apparent, RMS current.
706	totals	**tocrpk**	TotalPeakCurrent	A	float	4	1	ro	System total peak apparent, RMS current; highest value since last reset of the peaks.
708	totals	**tokwht**	TotalEnergy	kWh	float	4	1	ro	Total energy measurement in kWh. This value is not resetable.
710	totals	**tonepk**	TotalNeutralPeak	A	float	4	1	ro	Peak neutral current.
712	totals	**toneut**	TotalNeutralCurrent	A	float	4	1	ro	Neutral current measurement. This is the sum of the neutral measurents channels: (only in three phase configs, one channel for all except 63A, then there will be two).
714	totals	**torcpk**	TotalResidualPeak	A	float	4	1	ro	Peak residual current.
716	totals	**torcsc**	TotalResidualCurrent	A	float	4	1	ro	Residual current measurement.
718	totals	**tordcc**	TotalResidualDcCurrent	A	float	4	1	ro	DC residual current measurement.
720	totals	**torpwr**	TotalActivePower	W	float	4	1	ro	Totals power (in W), sum of phases. power = current * voltage * powerfactor
722	totals	**toxpwr**	TotalReactivePower	VAR	float	4	1	ro	Totals reactive power (volt-amps reactive, in VAR), sum of phases. VAR = SQRT(SQR(VA) - SQR(W))
724	totals	**toracc**	TotalResidualAcCurrent	A	float	4	1	ro	AC residual current measurement.
726	totals	**torrpk**	TotalPeakPower	W	float	4	1	ro	Total peak power in Watt
730	totals	**tovoac**	TotalActualVoltage	V	float	4	1	ro	Total actual voltage in Volt
734	totals	**tofreq**	TotalFrequency	Hz	float	4	1	ro	Total frequency in Hz
3000	inputs	**inaeng**	InputApparentEnergy	VAh	float	4	3	ro	The apparent accumulated energy VAh of each phase
3006	inputs	**inapwr**	InputApparentPower	VA	float	4	3	ro	Input VA (volt-amps, in VA), MEASURED by the device using current and voltage measurements of a phase (phase is denoted by repeat/channel). VA = current * voltage
3012	inputs	**incfac**	InputCrestFactor		float	4	3	ro	The crest factor per input. the ratio of peak values to the effective value
3018	inputs	**incrac**	InputActualCurrent	A	float	4	3	ro	Actual apparent, RMS current of an input.
3024	inputs	**incrpk**	InputPeakCurrent	A	float	4	3	ro	Input peak apparent, RMS current; highest value since last reset of the peaks.
3030	inputs	**infreq**	InputFrequency	Hz	float	4	3	ro	Frequency in Hz per input.
3042	inputs	**inkwht**	InputEnergyTotal	kWh	float	4	3	ro	Either the only phase in a single phase measurement; or one of the three phases in a multiphase measurement. This value is not resetable.
3048	inputs	**inname**	InputName		string	64	3	rw	User configurable 64 character naming of the inputs or phases.
3144	inputs	**inpfac**	InputPowerFactor		float	4	3	ro	The effective power factor for each input.
3150	inputs	**inphse**	InputPhase		int	1	3	ro	A mapping to which phase each input is connected to
3153	inputs	**inrpwr**	InputActivePower	W	float	4	3	ro	Input power (in W), MEASURED by the device using current, voltage, and power factor measurements of a phase (phase is denoted by repeat/channel). power = current * voltage * powerfactor
3159	inputs	**inrtng**	InputMaxRatedAmps	A	float	4	3	ro	Maximum rated current in A per input.
3165	inputs	**inthdp**	InputTHD	%	float	4	3	ro	Total harmonic distortion in percent per phase
3171	inputs	**intype**	InputType		int	1	3	ro	Type of input, to be determined
3174	inputs	**invcal**	InputVoltCalibration	V	float	4	3	ro	Input calibration factor for voltage, per input, for use in endtest
3180	inputs	**invoac**	InputActualVoltage	V	float	4	3	ro	The actual voltage per input.
3186	inputs	**invodp**	InputMinVoltage	V	float	4	3	ro	RMS voltage dip; lowest value since reset of dips.
3192	inputs	**invopk**	InputMaxVoltage	V	float	4	3	ro	RMS voltage peak; highest value since reset of peaks.
3198	inputs	**inxpwr**	InputReactivePower	VAR	float	4	3	ro	Reactive power (volt-amps reactive, in VAR), MEASURED by the device (phase number is denoted by repeat/channel). VAR = SQRT(SQR(VA) - SQR(W))
3204	inputs	**inrppk**	InputPeakPower	W	float	4	3	ro	Input peak power in Watt
4000	branches	**braeng**	BranchApparentEnergy	VAh	float	4	18	ro	The apparent accumulated energy VAh of each branch

4036	branches	**brapwr**	BranchApparentPower	VA	float	4	18	ro	Branch VA (volt-amps), MEASURED by the device using current and voltage measurements of an branch (branch number is denoted by repeat/channel). VA = current * voltage
4072	branches	**brcfac**	BranchCrestFactor		float	4	18	ro	The crest factor per branch. the ratio of peak values to the effective value
4108	branches	**brcrac**	BranchActualCurrent	A	float	4	18	ro	Actual apparent, RMS current of a branch.
4144	branches	**brcrpk**	BranchPeakCurrent	A	float	4	18	ro	Peak apparent, RMS current of a branch; highest value since last reset of the peaks.
4180	branches	**brfreq**	BranchFrequency	Hz	float	4	18	ro	Frequency in Hz per branch.
4252	branches	**brkwht**	BranchEnergyTotal	kWh	float	4	18	ro	Branch energy total measurement in kWh. This value is not resetable. 24-bit little-endian integer.
4288	branches	**brname**	BranchName		string	64	18	rw	User configurable naming of a branch in 64 characters per branch. Always returns the values for all entries (= the number of repeats) regardless of PDU configuration.
4864	branches	**brpfac**	BranchPowerFactor		float	4	18	ro	The effective power factor per branch.
4900	branches	**brphse**	BranchPhase		int	1	18	ro	A mapping to which phase (input) each branch is connected to
4936	branches	**brrpwr**	BranchActivePower	W	float	4	18	ro	Branch power (in W), MEASURED by the device using current, voltage, and power factor measurements of an branch (branch number is denoted by repeat/channel). power = current * voltage * powerfactor.
4972	branches	**brrtng**	BranchMaxRatedAmps	A	float	4	18	ro	Maximum rated current in A per branch (representing the breakerfuse value). [changed mnemonic to match; was brmaxr]
5008	branches	**brstat**	BranchState		int	1	18	ro	Power state of the branch, usable for determining if a branch is currently tripped. 0 = OK, 1 = Tripped.
5026	branches	**brthdp**	BranchTHD	%	float	4	18	ro	Total harmonic distortion in percent per branch
5062	branches	**brtype**	BranchType		int	1	18	ro	Breaker type: 0: Unknown, 1: ETA (thermal, resettable), 2: HYMA (hydraulic magnetic), 3: MCB (thermal magnetic DIN), 4: MELT (small ceramic fuse), 5: REVOLVER (large ceramic fuse, DIN holder)
5098	branches	**brvoac**	BranchActualVoltage	V	float	4	18	ro	The actual voltage per branch.
5134	branches	**brvodp**	BranchMinVoltage	V	float	4	18	ro	RMS voltage dip per branch; lowest value since reset of dips.
5170	branches	**brvopk**	BranchMaxVoltage	V	float	4	18	ro	RMS voltage peak per branch; highest value since reset of peaks.
5206	branches	**brxpwr**	BranchReactivePower	VAR	float	4	18	ro	Reactive power (volt-amps reactive, in VAR) of a branch, MEASURED by the device (branch number is denoted by repeat/channel). VAR = SQRT(SQR(VA) - SQR(W))
5314	branches	**brrppk**	BranchPeakPower	W	float	4	18	ro	Branch peak power in Watt
6000	outlets	**olaeng**	OutletApparentEnergy	VAh	float	4	60	ro	The apparent accumulated energy VAh of each outlet
6120	outlets	**olapwr**	OutletApparentPower	VA	float	4	60	ro	Output VA (volt-amps, in VA), MEASURED by the device using current and voltage measurements of an outlet (outlet number is denoted by repeat/channel). VA = current * voltage .
6240	outlets	**olbrch**	OutletBranch		int	1	108	ro	A mapping to which branch each outlet is connected to
6348	outlets	**olcfac**	OutletCrestFactor		float	4	60	ro	The crest factor per outlet. the ratio of peak values to the effective value
6468	outlets	**olcrac**	OutletActualCurrent	A	float	4	60	ro	Actual apparent, RMS current per outlet.
6588	outlets	**olcrpk**	OutletPeakCurrent	A	float	4	60	ro	Peak apparent, RMS current of an outlet; highest value since last reset of the peaks.
6828	outlets	**olkwht**	OutletEnergyTotal	kWh	float	4	60	ro	Total kWh of selected outlet. This value is not resetable.
6948	outlets	**olname**	OutletName		string	64	108	rw	User configurable naming of individual outlets.
10404	outlets	**olocst**	OutletCurrentState		int	1	60	rw	The state of the outlet relay(s)
10464	outlets	**olpcdl**	OutletPowerCycleDelay	seconds	int	1	60	rw	Individual power cycle timer. This is the amount, in seconds, for each outlet (denoted by the channel) to wait until the relay should be switched on again.
10524	outlets	**olpfac**	OutletPowerFactor		float	4	60	ro	Power factor per outlet.
10644	outlets	**olphse**	OutletPhase		int	1	108	ro	A mapping to which input each outlet is connected to
10752	outlets	**olrpwr**	OutletActivePower	W	float	4	60	ro	Output power (in W), MEASURED by the device using current, voltage, and power factor measurements of an outlet (outlet number is denoted by repeat/channel). power = current * voltage * powerfactor
10872	outlets	**olrtng**	OutletMaxRatedAmps	A	float	4	108	ro	Maximum rated current in A per outlet.
11088	outlets	**olschd**	OutletScheduled		int	1	60	rw	The intended (externally set) state of an outlet. 0 = Relay open, 1 = Relay closed. See [olocst] for actual state.
11148	outlets	**olswdl**	OutletSwitchDelay	seconds	int	1	60	rw	Delay before an individual outlet's relay switches on at power-up, in seconds.
11208	outlets	**olthdp**	OutletTHD	%	float	4	60	ro	Total harmonic distortion per outlet
11328	outlets	**oltype**	OutletType		int	1	108	ro	Outlet types: 0: Unknown, 1: BS, 2: C13, 3: C13 IEC Lock, 4: C19, 5: C19 IEC Lock, 6: CEE 16A 1 phase, 7: CEE 16A 3 phase, 8: CEE 32A 1 phase, 9: CEE 32A 3 phase, 10: CEE 63A 1 phase, 11: CEE 63A 3 phase, 12: COMBI IEC Lock, 13: GST 155, 14: GST 183, 15: GST 185, 16: Schuko, 17: T13, 18: T23, 19: UTE
11436	outlets	**olvoac**	OutletActualVoltage	V	float	4	60	ro	Actual voltage on outlet.
11556	outlets	**olxpwr**	OutletReactivePower	VAR	float	4	60	ro	Reactive power (volt-amps reactive, in VAR), MEASURED by the device (outlet number is denoted by repeat/channel). VAR = SQRT(SQR(VA) - SQR(W))
11676	outlets	**olrppk**	OutletPeakPower	W	float	4	60	ro	Outlet peak power in Watt
15000	sensors	**snindx**	SensorIndex		int	1	32	ro	Sensor index
15032	sensors	**snsnme**	SensorName		string	36	32	rw	User definable name for sensors.
16056	sensors	**snsril**	SensorSerialNumber		string	8	32	ro	Serial number
16424	sensors	**snssrc**	SensorSource		int	1	32	ro	Source of the sensor
16456	sensors	**snstat**	SensorState		int	1	32	ro	State of the sensor
16200	sensors	**snstyp**	SensorType		int	1	32	ro	Returns the detected sensor type, can be: Temperature, Humidity, Dry Switch Contact, Output Switch
16232	sensors	**snsval**	SensorValue		float	4	32	ro	NOTE: RELAY in separate section; sensors only return measurements
16392	sensors	**snunit**	SensorUnit		int	1	32	ro	Sensor measurement unit
18000	actuators	**acindx**	ActuatorIndex		int	1	16	ro	Actuator index
18032	actuators	**acname**	ActuatorName		string	25	16	rw	User definable name for actuators.
18432	actuators	**acocst**	ActuatorCurrentState		int	1	16	rw	The state of the actuator outlet
18464	actuators	**acschd**	ActuatorScheduled		int	1	16	ro	The intended (externally set) state of an actuator. 0 = Actuator inactive, 1 = Actuator active. See [acocst] for actual state.
18592	actuators	**acsril**	ActuatorSerialNumber		string	8	16	ro	Serial number
18624	actuators	**acssrc**	ActuatorSource		int	1	16	ro	Source of the actuator
18656	actuators	**acstat**	ActuatorState		int	1	16	ro	State of the actuator
18688	actuators	**acstyp**	ActuatorType		int	1	16	ro	Returns the detected actuator type, can be: Actuator 1, Actuator 2
19032	digital_in	**diname**	DigitalInName		string	36	32	rw	User definable name for digital inputs
19608	digital_in	**disril**	DigitalInSerial		string	8	32	ro	Serial number
19736	digital_in	**dissrc**	DigitalInSource		int	1	32	ro	Source of the digital input
19768	digital_in	**distat**	DigitalInState		int	1	32	ro	The state of the digital input
19800	digital_in	**distyp**	DigitalInType		int	1	32	ro	Returns the digital input type
19832	digital_in	**disval**	DigitalInValue		int	1	32	ro	Returns the value of the digital input