



Version: 1.0

EM

EM (Energy Meter) component handles the data collection and processing from energy meter devices like the [ShellyPro3EM](#).

- [EM.SetConfig](#) to update the component's [configuration](#)
- [EM.GetConfig](#) to obtain the component's [configuration](#)
- [EM.GetStatus](#) to obtain the component's [status](#)
- [EM.PhaseToPhaseCalib](#) [calibrate](#) a phase CT from another phase's CT (if applicable)
- [EM.PhaseToPhaseCalibReset](#) [reset](#) a **user** calibrated CT to factory defaults (if applicable)
- [EM.GetCTTypes](#) to obtain list of supported current transformer types

Methods

EM.SetConfig

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the EM component instance
<code>config</code>	<i>object</i>	Configuration that the method takes

Find more about the config properties in [config section](#)

EM.GetConfig

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the EM component instance

Find the `EM.GetConfig` response properties in [config section](#)

EM.GetStatus

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the EM component instance

Find more about the status response properties in [status section](#)

EM.PhaseToPhaseCalib

Calibrate a phase CT from another phase's CT (if applicable).

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the EM component instance
<code>from</code>	<i>string</i>	Select the phase from witch the calibration data is taken
<code>to</code>	<i>string</i>	Select the phase to witch the calibration is due

Response:

`restart_required:true` on success; error if request can't be executed or failed

! INFO

In order to be able to calibrate correctly the phase voltages need to be equal i.e. if we calibrate a CT from phase **c** to a CT on phase **a**, the voltage of phase **c** and phase **a** needs to be the same (phases are connected on the same voltage line). The minimum power allowed for the calibration to take place is **500W**. The method takes around 5 seconds to complete, and the response is delayed with the request answer - **restart_required: true** in case of success or error message in case of **fail**. The reasons for the calibration to **fail** may be - deviation in measurement after calibration on the **from** and **to** phases that indicating incorrect input phases setup, incorrect CTs or other problem.

EM.PhaseToPhaseCalibReset

Reset a user calibrated CT to factory defaults (if applicable)

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the EM component instance
<code>phase</code>	<i>string</i>	Phase for which user calibration is going to be reset

Response:

`restart_required:true` on success; error if request can't be executed or failed

EM.GetCTTypes

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the EM component instance

Response:

Property	Type	Description
<code>supported</code>	<i>array</i>	Array of strings of all supported CT types

Configuration

The configuration of the Energy Meter component shows the energy metering device's name, blink mode selector, phase selector and phase sequence detection enabled. To Get/Set the configuration of the EM component its `id` must be specified. Name setting does not require restart however, setting `blink_mode_selector`, `phase_selector` and/or `ct_type` requires restart.

Properties:

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the EM component instance
<code>name</code>	<i>string or null</i>	Name of the energy meter instance
<code>blink_mode_selector</code>	<i>string</i>	Select the electrical quantity that drives the LED. Range of values: <code>active_energy</code> , <code>apparent_energy</code>
<code>phase_selector</code>	<i>string</i>	Select witch phase controls the LED. Range of values: <code>a</code> , <code>b</code> , <code>c</code> , <code>all</code>
<code>monitor_phase_sequence</code>	<i>boolean</i>	Set this to show <code>phase_sequence</code> error in GetStatus if three-phase power system is used and the wires are not connected correctly to the device.

Property	Type	Description												
reverse	object	Reverse CT measurement direction												
		<table><tr><th>Property</th><th>Type</th><th>Description</th></tr><tr><td>a</td><td>boolean</td><td>When set to true reverse CT measurement direction of active power and energy for phase A. <i>shown if true</i></td></tr><tr><td>b</td><td>boolean</td><td>When set to true reverse CT measurement direction of active power and energy for phase B. <i>shown if true</i></td></tr><tr><td>c</td><td>boolean</td><td>When set to true reverse CT measurement direction of active power and energy for phase C. <i>shown if true</i></td></tr></table>	Property	Type	Description	a	boolean	When set to true reverse CT measurement direction of active power and energy for phase A. <i>shown if true</i>	b	boolean	When set to true reverse CT measurement direction of active power and energy for phase B. <i>shown if true</i>	c	boolean	When set to true reverse CT measurement direction of active power and energy for phase C. <i>shown if true</i>
		Property	Type	Description										
		a	boolean	When set to true reverse CT measurement direction of active power and energy for phase A. <i>shown if true</i>										
		b	boolean	When set to true reverse CT measurement direction of active power and energy for phase B. <i>shown if true</i>										
c	boolean	When set to true reverse CT measurement direction of active power and energy for phase C. <i>shown if true</i>												
setting the reverse option requires restart														
ct_type	string	Select the type of Shelly current transformer attached to the device.												

i NOTE

If `ct_type` is not set, an error `ct_type_not_set` is present in component status.

Supported `ct_type`s can be obtained with `EM.GetCTTypes`.

Webhook events

EM component supports conditional webhooks.

Events related to the EM component that can trigger webhooks:

- `voltage_change` - when voltage has changed with at least 1V and 5% from the last reported value.
 - `voltage_change` event supports attributes, that can be used to compose conditional [webhooks](#):

Property	Type	Description
<code>phase</code>	<i>string</i>	Phase which generates the event, one of 'a', 'b' or 'c'.
<code>voltage</code>	<i>number</i>	New phase voltage in Volts

- `current_change` - when current has changed with at least 0.05A and 5% from the last reported value.
 - `current_change` event supports attributes, that can be used to compose conditional [webhooks](#):

Property	Type	Description
<code>phase</code>	<i>string</i>	Phase which generates the event, one of 'a', 'b', 'c' or 'n'.
<code>current</code>	<i>number</i>	New phase current in Amps

- `active_power_change` - when active power has changed with at least 10W and 5% from the last reported value.
 - `active_power_change` event supports attributes, that can be used to compose conditional [webhooks](#):

Property	Type	Description
<code>phase</code>	<i>string</i>	Phase which generates the event, one of 'a', 'b' or 'c'.
<code>act_power</code>	<i>number</i>	New phase active power in Watts

- `total_current_change` - when the total current has changed with at least 3A and 5% from the last reported value.
 - `total_current_change` event supports attributes, that can be used to compose conditional [webhooks](#):

Property	Type	Description
<code>total_current</code>	<i>number</i>	New total current in Amps

- `total_active_power_change` - when the total active power has changed with at least 100W and 5% from the last reported value.
 - `total_active_power_change` event supports attributes, that can be used to compose conditional [webhooks](#):

Property	Type	Description
<code>total_active_power</code>	<i>number</i>	New total active power in Watts

- `total_apparent_power_change` - when the total apparent power has changed with at least 100W and 5% from the last reported value.
 - `total_apparent_power_change` event supports attributes, that can be used to compose conditional [webhooks](#):

Property	Type	Description
<code>total_apparent_power</code>	<i>number</i>	New total apparent power in VAs

Status

The status of the Energy Meter Component contains information from the current measurements, for all active phase(es) of the device. To obtain information for the components status its `id` must be specified. The information for a phase contains information for:

- The momentary values of the `current` in A (Amps), `voltage` in V (Volts), `active_power` in W (Watts), `apparent_power` in VA (Volt-Ampere), `pf` - power factor is dimensionless, `freq` - network frequency in Hz.
- `neutral` is the reporting for the `neutral` current measurements.

Property	Type	Description
<code>id</code>	<i>number</i>	Id of the EM component instance
<code>a_current</code>	<i>number or null</i>	Phase A current measurement value, [A]
<code>a_voltage</code>	<i>number or null</i>	Phase A voltage measurement value, [V]
<code>a_act_power</code>	<i>number or null</i>	Phase A active power measurement value, [W]
<code>a_aprt_power</code>	<i>number or null</i>	Phase A apparent power measurement value, [VA]
<code>a_pf</code>	<i>number or null</i>	Phase A power factor measurement value
<code>a_freq</code>	<i>number or null</i>	Phase A network frequency measurement value
<code>a_errors</code>	<i>array of type</i>	Phase A error conditions occurred. May contain <code>out_of_range:active_power</code> ,

Property	Type	Description
	<i>string</i>	<code>out_of_range:apparent_power</code> , <code>out_of_range:voltage</code> , <code>out_of_range:current</code> , (shown if at least one error is present)
<code>b_current</code>	<i>number or null</i>	Phase B current measurement value, [A]
<code>b_voltage</code>	<i>number or null</i>	Phase B voltage measurement value, [V]
<code>b_act_power</code>	<i>number or null</i>	Phase B active power measurement value, [W]
<code>b_aprt_power</code>	<i>number or null</i>	Phase B apparent power measurement value, [VA]
<code>b_pf</code>	<i>number or null</i>	Phase B power factor measurement value
<code>b_freq</code>	<i>number or null</i>	Phase B network frequency measurement value
<code>b_errors</code>	<i>array of type string</i>	Phase B error conditions occurred. May contain <code>out_of_range:active_power</code> , <code>out_of_range:apparent_power</code> , <code>out_of_range:voltage</code> , <code>out_of_range:current</code> , (shown if at least one error is present)

Property	Type	Description
<code>c_current</code>	<i>number or null</i>	Phase C current measurement value, [A]
<code>c_voltage</code>	<i>number or null</i>	Phase C voltage measurement value, [V]
<code>c_act_power</code>	<i>number or null</i>	Phase C active power measurement value, [W]
<code>c_aprt_power</code>	<i>number or null</i>	Phase C apparent power measurement value, [VA]
<code>c_pf</code>	<i>number or null</i>	Phase C power factor measurement value
<code>c_freq</code>	<i>number or null</i>	Phase C network frequency measurement value
<code>c_errors</code>	<i>array of type string</i>	Phase C error conditions occurred. May contain <code>out_of_range:active_power</code> , <code>out_of_range:apparent_power</code> , <code>out_of_range:voltage</code> , <code>out_of_range:current</code> (shown if at least one error is present)
<code>n_current</code>	<i>number or null</i>	Neutral current measurement value, [A] (if supported)

Property	Type	Description
<code>n_errors</code>	<i>array of type string</i>	Neutral error conditions occurred. May contain <code>out_of_range:current</code> , (shown if error is present)
<code>total_current</code>	<i>number or null</i>	Sum of the current on all phases(excluding neutral readings if available)
<code>total_act_power</code>	<i>number or null</i>	Sum of the active power on all phases
<code>total_aprt_power</code>	<i>number or null</i>	Sum of the apparent power on all phases
<code>user_calibrated_phase</code>	<i>array of type string</i>	Indicates which phase was user calibrated
<code>errors</code>	<i>array of type string</i>	EM component error conditions. May contain <code>power_meter_failure</code> , <code>phase_sequence</code> or <code>ct_type_not_set</code> . Present in status only if not empty.

! INFO

* `phase_sequence` is an error indicating if the sequence of zero-crossing events is Phase A followed by Phase C followed by Phase B. The regular succession of these zero-crossing events is Phase A followed by Phase B followed by Phase C.

Notifications

StatusChange

[Statuschange](#) notification on a regular interval and an error condition change.

Modbus registers

Address	Type	Description
31000	uint32	Timestamp of the last update
31002	boolean	Phase A meter error
31003	boolean	Phase B meter error
31004	boolean	Phase C meter error
31005	boolean	Neutral meter error
31006	boolean	Phase sequence error
31007	float	Neutral current, A
31009	boolean	Neutral current mismatch
31010	boolean	Neutral overcurrent error
31011	float	Total current, A *
31013	float	Total active power, W *
31015	float	Total apparent power, VA *
31017		3 registers reserved
31020	float	Phase A voltage, V
31022	float	Phase A current, A

Address	Type	Description
31024	float	Phase A active power, W
31026	float	Phase A apparent power, VA
31028	float	Phase A power factor
31030	boolean	Phase A overpower error
31031	boolean	Phase A overvoltage error
31032	boolean	Phase A overcurrent error
31033	float	Phase A frequency, Hz
31035		5 registers reserved
31040	float	Phase B voltage, V
31042	float	Phase B current, A
31044	float	Phase B active power, W
31046	float	Phase B apparent power, VA
31048	float	Phase B power factor
31050	boolean	Phase B overpower error
31051	boolean	Phase B overvoltage error
31052	boolean	Phase B overcurrent error
31053	float	Phase B frequency, Hz
31055		5 registers reserved
31060	float	Phase C voltage, V

Address	Type	Description
31062	float	Phase C current, A
31064	float	Phase C active power, W
31066	float	Phase C apparent power, VA
31068	float	Phase C power factor
31070	boolean	Phase C overpower error
31071	boolean	Phase C overvoltage error
31072	boolean	Phase C overcurrent error
31073	float	Phase C frequency, Hz
31075		5 registers reserved

If there is a second EM component on the device, its corresponding registers addresses are calculated by adding 80 to the address in the table above.

* sum of all available phases

Examples

EM.SetConfig example

Example:

**EM.SetConfig HTTP GET
Request**

**EM.SetConfig Curl
Request**

**EM.SetConfig Mos
Request**

```
http://192.168.33.1/rpc/EM.SetConfig?id=0&config=
{"name":"abc","blink_mode_selector":"apparent_energy","phase_selector":"all","mon
```

Response

```
{
  "restart_required": true
}
```

EM.GetConfig example

Example:

EM.GetConfig HTTP GET Request

EM.GetConfig Curl Request

EM.GetConfig Mos Request

```
http://192.168.33.1/rpc/EM.GetConfig?id=0
```

Response

```
{
  "id": 0,
  "name": null,
  "blink_mode_selector": "active_energy",
  "phase_selector": "a",
  "monitor_phase_sequence": true,
  "reverse": {
    "b": true
  },
  "ct_type": "120A"
}
```

EM.GetStatus example

EM.GetStatus HTTP GET Request

EM.GetStatus Curl Request

EM.GetStatus Mos Request

```
http://192.168.33.1/rpc/EM.GetStatus?id=0
```

Response

```
{
  "id": 0,
  "a_current": 4.029,
  "a_voltage": 236.1,
  "a_act_power": 951.2,
  "a_aprt_power": 951.9,
  "a_pf": 1,
  "a_freq": 50,
  "b_current": 4.027,
  "b_voltage": 236.201,
  "b_act_power": -951.1,
  "b_aprt_power": 951.8,
  "b_pf": 1,
  "b_freq": 50,
  "c_current": 3.03,
  "c_voltage": 236.402,
  "c_active_power": 715.4,
  "c_aprt_power": 716.2,
  "c_pf": 1,
  "c_freq": 50,
  "n_current": 11.029,
  "total_current": 11.083,
  "total_act_power": 2484.782,
  "total_aprt_power": 2486.7,
  "user_calibrated_phase": [],
  "errors": [
    "phase_sequence"
  ]
}
```

EM.PhaseToPhaseCalib example

EM.PhaseToPhaseCalib HTTP GET Request

EM.PhaseToPhaseCalib Curl Request

EM.PhaseToPhaseCalib Mos Request

```
http://192.168.33.1/rpc/EM.PhaseToPhaseCalib?id=0&from="a"&to="n"
```

Response

```
{
  "restart_required": true
}
```



```
}
```

EM.PhaseToPhaseCalibReset example

EM.PhaseToPhaseCalibReset
HTTP GET Request

EM.PhaseToPhaseCalibReset
Curl Request

EM.PhaseToPhaseCalibReset
Mos Request

```
http://192.168.33.1/rpc/EM.PhaseToPhaseCalibReset?id=0&phase="a"
```

Response

```
{
  "restart_required": true
}
```

EM.GetCTTypes example

Example:

EM.GetCTTypes HTTP GET
Request

EM.GetCTTypes Curl
Request

EM.GetCTTypes Mos
Request

```
http://192.168.33.1/rpc/EM.GetCTTypes?id=0
```

Response

```
{
  "supported": [
    "120A",
    "400A"
  ]
}
```