ESP32_Relay Thermostat Protocol

Request designation	Request	Description
JSON_RPC_START_BROAD	startbrd	Start broadcast status
JSON_RPC_STOP_BROAD	stopbrd	Stop broadcast status
JSON_RPC_IS_BROAD	isbrd	Get broadcast status alert
JSON_RPC_STATUS	status	Get description of the current state of the thermostat
JSON_RPC_GET_T	gett	Get temperature value from the thermal sensor
JSON_RPC_GET_EXT_POW	getextp	Get message about external power supply
JSON_RPC_GET_HYST	gethyst	Get hysteresis value
JSON_RPC_CUR_MODE	curmode	Get active thermostat mode
JSON_RPC_SET_HYST	sethyst	Set hysteresis
JSON_RPC_RUN_MODE	runmode	Set the active mode of the thermostat
JSON_RPC_MODE_LIST	lstmodes	Get a list of modes with an indication of the limit temperature
JSON_RPC_SET_MODE	setmode	Set the temperature value for the selected mode
JSON_RPC_GET_MODE	getmode	Get the temperature value for the selected mode

Request and response fields

Field designation	Field	Type	Description
JSON_RPC_T	t	double	Temperature
JSON_RPC_TARGET_T	trgt	double	Target temperature value for the mode
JSON_RPC_HYST	hyst	double	Hysteresis
JSON_RPC_EXT_POW	extp	bool	External power supply **true**-yes mains power **false** – no mains power, battery
JSON_RPC_HEAT	heat	bool	Relay status (open/closed) **true**-the relay is closed **false**-the relay is open
JSON_RPC_MODE	mode	obj	Mode
JSON_RPC_MODE_VALUE	value	double	Value (field of the

			JSON_RPC_MODE object)
JSON_RPC_MODE_ID	id	int	id (field of the JSON_RPC_MODE object)
JSON_RPC_MODE_VALUES	values	array	Array of values

JSON_RPC_START_BROAD request

Launch a broadcast status alert.
Enabling a broadcast alert causes the relay to send its current state to the server with a frequency of no more than 1 minute, but no less than 1 hour. The JSON_RPC_STATUS message is generated by the device if (disjunction):
1. An hour has passed since the previous message.
2. The measured temperature has changed by 0.5 °C since the previous message.
3. The target temperature has changed since the previous message.
4. The external power status has changed since the previous message.
The request fields can be empty, and the passed values are ignored.
Request example:

```
{"msg":"startbrd", "target":"relay_device1"}
```

Response fields:

Field	Description
JSON_RPC_IS_BROAD	Broadcast alert status **true** – alert started **false**-alert stopped

Sample response:

```
{"msg":"isbrd", "target":"board0", "params": {"isbrd":true}}
```

JSON_RPC_STOP_BROAD request

Stop the broadcast status notification.
The request fields can be empty, and the passed values are ignored.
Request example:

```
{"msg":"stopbrd", "target":"relay_device1"}
```

Response fields:

Field	Description
JSON_RPC_IS_BROAD	Broadcast alert status **true** – alert started **false**-alert stopped

Sample response:

```
{"msg":"isbrd", "target":"board0", "params": {"isbrd":false}}
```

JSON_RPC_IS_BROAD request

Get the status of a broadcast alert.
The request fields can be empty, and the passed values are ignored.
Request example:

```
{"msg":"isbrd", "target":"relay_device1"}
```

Response fields:

Field	Description
JSON_RPC_IS_BROAD	Broadcast alert status **true** – alert started **false**-alert stopped

Sample response:

```
{"msg":"isbrd", "target":"board0", "params": {"isbrd":false}}
```

JSON_RPC_STATUS request

Get a description of the current state of the thermostat.
The request fields can be empty, and the passed values are ignored.
Request example:

```
{"msg":"status", "target":"relay_device1"}
```

Response fields:

Field	Description
JSON_RPC_T	Last temperature value measured by the thermal sensor
JSON_RPC_TARGET_T	Target temperature value for active mode
JSON_RPC_MODE_ID	Id of active thermostat mode
JSON_RPC_HYST	Current hysteresis value
JSON_RPC_EXT_POW	Presence/absence of external power supply
JSON_RPC_HEAT	Relay status (open/closed)

Sample response:

```
{"msg":"status", "target":"board0", "params":
{"t":19.0,"trgt":23.0,"id":2,"hyst":1.0,"extp":true,"heat":false}}
```

JSON_RPC_GET_T request

Get the temperature value from the thermal sensor.
The request fields can be empty, and the passed values are ignored.

Request example:

```
{"msg":"gett", "target":"relay_device1"}
```

Response fields:

Field	Description
JSON_RPC_T	Last temperature value measured by the thermal sensor

Sample response:

```
{"msg":"status", "target":"board0", "params": {"t":19.0}}
```

JSON_RPC_GET_EXT_POW request

Get a message about the presence of external power.
The request fields can be empty, and the passed values are ignored.
Request example:

```
{"msg":"getextp", "target":"relay_device1"}
```

Response fields:

Field	Description
JSON_RPC_EXT_POW	Presence/absence of external power supply

Sample response:

```
{"msg":"status", "target":"board0", "params": {"extp":true}}
```

JSON_RPC_GET_HYST request

Get the hysteresis value.
The request fields can be empty, and the passed values are ignored.
Request example:

```
{"msg":"gethyst", "target":"relay_device1"}
```

Response fields:

Field	Description Field
JSON_RPC_HYST	Current hysteresis value

Sample response:

```
{"msg":"status", "target":"board0", "params": {"hyst":1.0}}
```

JSON_RPC_CUR_MODE request

Get the current thermostat mode.
The request fields can be empty, and the passed values are ignored.
Request example:

```
{"msg":"curmode", "target":"relay_device1"}
```

Response fields:

Field	Description
JSON_RPC_MODE_ID	ID of the active mode

Sample response:

```
{"msg":"status", "target":"board0", "params": {"id":2}}
```

JSON_RPC_RUN_MODE request

Set the active thermostat mode.

Request fields:

Field	Description
JSON_RPC_MODE_ID	New ID of the active mode (0, 1, or 2)

Request example:

```
{"msg":"runmode", "target":"relay_device1", "params": {"id":1}}
```

Response fields:

Field	Description
JSON_RPC_MODE_ID	ID of the active mode

Sample response:

```
{"msg":"status", "target":"board0", "params": {"id":1}}
```

JSON_RPC_SET_HYST request

Set the hysteresis value.

Request fields:

Field	Description
JSON_RPC_HYST	New hysteresis value (from 0.125 to 2.5)

Request example:

```
{"msg":"sethyst", "target":"relay_device1", "params":
{"hyst":2.0}}
```

Response fields:

Field	Description Field
JSON_RPC_HYST	Current hysteresis value

Sample response:

```
{"msg":"status", "target":"board0", "params": {"hyst":2.0}}
```

JSON_RPC_MODE_LIST request

Get a list of modes with an indication of the maximum temperature.
The request fields can be empty, and the passed values are ignored.
Request example:

```
{"msg":"lstmodes", "target":"relay_device1"}
```

Response fields:

Field	Description
JSON_RPC_MODE_VALUES	List of temperature limits for all available modes

Sample response:

```
{"msg":"lstmodes", "target":"board0", "params": {"values":[5.0,
20.0, 23.0]}}
```

JSON_RPC_SET_MODE request

Set the temperature value for the selected mode.
Request fields:

Field	Description
JSON_RPC_MODE_ID	Mode ID (0, 1, or 2)
JSON_RPC_MODE_VALUE	New temperature limit value for the mode (minimum 0.0, maximum 50.0)

Request example:

```
{"msg":"setmode", "target":"relay_device1", "params": {"id":1,
"value":12.0}}
```

Response fields:

Field	Description
JSON_RPC_MODE_ID	Mode ID
JSON_RPC_MODE_VALUE	Stored temperature limit value for the mode

Sample response:

```
{"msg":"mode", "target":"board0", "params": {"id":1,
"value":12.0}}
```

JSON_RPC_GET_MODE request

Get the temperature value for the selected mode.
Request fields:

Field	Description
JSON_RPC_MODE_ID	Mode ID (0, 1, or 2)

Request example:

```
{"msg":"getmode", "target":"relay_device1", "params": {"id":2}}
```

Response fields:

Field	Description
JSON_RPC_MODE_ID	Mode ID
JSON_RPC_MODE_VALUE	Limit temperature value for the mode

Sample response:

```
{"msg":"mode", "target":"board0", "params": {"id":2,
"value":23.0}}
```