Marstek Device Open API（Rev 1.0)

The Local API is provided "as is" for local use only. Use at your own risk. Marstek is not liable for any damages, data loss, or legal issues caused by your use of the API. You are responsible for lawful and appropriate use.

Ⅰ. Preface

Welcome!
This document provides an introduction to the Open API for Marstek devices, which is available to device owners and enables integration with third-party systems.

While Marstek offers an official mobile app and cloud services, this Open API is designed for advanced users who wish to gain greater control over their devices and seamlessly integrate them into other management platforms.

Ⅱ. General Description

Marstek devices communicate with third-party systems over a Local Area Network (LAN). Before using this API, please ensure that:

- The Marstek device is properly connected to your home network.
- The Open API feature has been enabled via the Marstek mobile app.

Please note that different Marstek models may support only a subset of the commands described in this documentation. Additionally, enabling the Open API may cause certain built-in features of the device to be disabled in order to prevent command conflicts. For detailed information about supported commands and any functional limitations for specific models, please refer to **Chapter 4** of this document.

2.1 Protocol Format

The protocol utilizes the JSON format, with commands primarily categorized into query commands, configuration commands, and certain special commands.

Command Format

Property	Type	Description
id	number or string	An identifier established by the Client.
method	string	A Structured value that holds the parameter values to be used during the invocation of the method.
params	object	Parameters that the method takes.

Example

```
{
    "id": 0,
    "method": "string",
    "params": {
        "id":0
    }
}
```

Device Response Format

Property	Type	Description
id	number or string	An identifier established by the Client.
src	string	Name of the source of the request.
result	object	This member is required on success.
error	object	This member is required on error.

Error code

Code	Message	Meaning
-32700	Parse error	Invalid JSON was received by the server. An error occurred on the server while parsing the JSON text.
-32600	Invalid Request	The JSON sent is not a valid Request object.
-32601	Method not found	The method does not exist / is not available.
-32602	Invalid params	Invalid method parameter(s).
-32603	Internal error	Internal JSON-RPC error.
-32000 to -32099	Server error	Reserved for implementation-defined server-errors.

The remainder of the space is available for application defined errors.

Example response on success

```
{
    "id": 0,
    "src": "device",
    "result": {
        "id":0
    }
}
```

Example response on error

```
{
    "id": 0,
    "src": "device",
    "error": {
        "code": -32700,
        "message": "Parse error"
    }
}
```

2.2 API over UDP

2.2.1 First-Time Use

When users first use the Open API service, they need to follow the configuration process below:

- Connect the device to power and turn it on;
- Use the Marstek APP to connect and bind the device, and configure the WiFi network for the device or connect it to the Ethernet;
- Enable the device's API feature in the APP, and set the UDP port number. The default port number is 30000, and the recommended port number is between 49152 and 65535.

After completing the above operations, the Marstek device can normally receive UDP commands from the same local area network.

2.2.2 Discovering Devices

To discover Marstek devices within the LAN, a UDP broadcast is utilized. The broadcast content is as follows:

```
{
    "id": 0,
    "method": "Marstek.GetDevice",
    "params": {
        "ble_mac":"0"
    }
}
```

If there are Marstek devices within the LAN, taking Venus C as an example, the following response will be received

```
{
    "id": 0,
    "src": "VenusC-123456789012",
    "result": {
        "device":"VenusC",
        "ver":111,
        "ble_mac":"123456789012",
        "wifi_mac":"123456789012",
        "wifi_name":"MY_HOME",
        "ip":"192.168.1.11"
    }
}
```

The device's IP address can be directly obtained from the Marstek APP or the home router. If this functionality is to be used on a long-term basis, it is recommended to configure the device with a static IP address.

Ⅲ. Components

This chapter mainly introduces the components and services supported by the Marstek device.

3.1 Marstek

Marstek contains some basic information about the product, and is mainly used for discovering devices and querying basic device information.

- Marstek.GetDevice: Locate Marstek devices on the local area network.

3.1.1 Marstek.GetDevice

Sending:

Property(params)	Type	Description
ble_mac	string	Valid mac，can be used to identify a specific device.

Response:

Property(result)	Type	Description
device	string	Device model
ver	number	Device firmware version
ble_mac	string	Bluetooth MAC
wifi_mac	string	WiFi MAC
wifi_name	string	WiFi name
ip	string	Device IP

Example:

Sending:

```
{
    "id": 0,
    "method": "Marstek.GetDevice",
    "params": {
        "ble_mac":"123456789012"
    }
}
```

Response:

```
{
    "id": 0,
    "src": "VenusC-123456789012",
    "result": {
        "device":"VenusC",
        "ver":111,
        "ble_mac":"123456789012",
        "wifi_mac":"012123456789",
        "wifi_name": "MY_HOME",
        "ip":"192.168.1.11"
    }
}
```


3.2 WiFi

The WiFi component is mainly used for configuring the device's WiFi and obtaining the device's basic network information.

- Wifi.GetStatus: Obtain the device's basic network information.

3.2.1 Wifi.GetStatus

Sending:

Property (params)	Type	Description
id	*number*	ID of Instance

Response:

Property (result)	Type	Description
id	*number*	ID of Instance
wifi_mac	*string*	WiFi MAC
ssid	*string or null*	WiFi name
rssi	*number*	WiFi signal strength
sta_ip	*string or null*	Device IP
sta_gate	*string or null*	Gateway
sta_mask	*string or null*	Subnet mask
sta_dns	*string or null*	DNS

Example:

Sending:

```
{
    "id": 1,
    "method": "Wifi.GetStatus",
    "params": {
        "id": 0
    }
}
```

Response:

```
{
    "id": 1,
    "src": "VenusC-mac",
    "result": {
        "id": 0,
        "ssid":"Hame",
        "rssi": -59,
        "sta_ip":"192.168.137.41",
        "sta_gate":"192.168.137.1",
        "sta_mask":"255.255.255.0",
        "sta_dns":"192.168.137.1"
    }
}
```

3.3 Bluetooth

The BLE (Bluetooth) component can view the Bluetooth-related information of the device.

- BLE.GetStatus: Check the Bluetooth connection status of the device.

3.3.1 BLE.GetStatus

Sending:

Property (params)	Type	Description
id	*number*	ID of Instance

Response:

Property (result)	Type	Description
state	*string*	Bluetooth state
ble_mac	*string*	Bluetooth MAC

Example:

Sending:

```
{
    "id": 1,
    "method": "BLE.GetStatus",
    "params": {
        "id": 0
    }
}
```

Response:

```
{
    "id": 1,
    "src": "VenusC-123456789012",
    "result": {
        "id": 0,
        "state":"connect",
        "ble_mac":"123456789012"
    }
}
```

3.4 Battery

The Bat (Battery) component contains basic information about the device's battery.

- Bat.GetStatus: Query the device's battery information and operating status.

3.4.1 Bat.GetStatus

Sending:

Property (params)	Type	Description
id	*number*	ID of Instance

Response:

Property (result)	Type	Description
id	*number*	ID of Instance
soc	*string*	soc
charg_flag	*boolean*	Charging permission flag
dischrg_flag	*boolean*	Discharge permission flag
bat_temp	*number or null*	Battery temperature, [°C]
bat_capacity	*number or null*	Battery remaining capacity, [Wh]
rated_capacity	*number or null*	Battery rated capacity, [Wh]

Example:

Sending:

```
{
    "id": 1,
    "method": "Bat.GetStatus",
    "params": {
        "id": 0
    }
}
```

Response:

```
{
    "id": 1,
    "src": "VenusC-mac",
    "result": {
        "id": 0,
        "soc": 98,
        "charg_flag": true,
        "dischrg_flag": true,
        "bat_temp": 25.0,
        "bat_capacity": 2508.0,
        "rated_capacity": 2560.0
    }
}
```

3.5 PV

The PV (Photovoltaic) component contains the photovoltaic information connected to the device.

- PV.GetStatus: Query the device's connected photovoltaic information and power generation status.

3.5.1 PV.GetStatus

Sending:

property (params)	Type	Description
id	*number*	ID of Instance

Response:

Property (result)	Type	Description
id	*number*	ID of Instance
pv_power	*number*	Photovoltaic charging power, [W]
pv_voltage	*number*	Photovoltaic charging voltage, [V]
pv_current	*number*	Photovoltaic charging current, [A]

Example:

Sending:

```
{
    "id": 1,
    "method": "PV.GetStatus",
    "params": {
        "id": 0
    }
}
```

Response:

```
{
    "id": 1,
    "src": "VenusC-mac",
    "result": {
        "id": 0,
        "pv_power": 580.0,
        "pv_voltage": 40.0,
        "pv_current": 12.0
    }
}
```

3.6 ES

The ES (Energy System) component contains the device's basic power information and energy statistics, and can configure or monitor the device's operating status.

- ES.GetStatus: Query the device's basic electrical energy information.
- ES.SetMode: Configure the device's operating mode.
- ES.GetMode: Get information about the operating mode of the device.

3.6.1 ES.GetStatus

Sending:

Property (params)	Type	Description
id	*number or null*	ID of Instance

Response:

Property (result)	Type	Description
id	*number or null*	ID of Instance
bat_soc	*number or null*	Total battery SOC, [%]
bat_cap	*number or null*	Total battery capacity, [Wh]

Property (result)	Type	Description
pv_power	*number or null*	Solar charging power, [W]
ongrid_power	*number or null*	Grid-tied power, [W]
offgrid_power	*number or null*	Off-grid power, [W]
bat_power	*number or null*	Battery power, [W]
total_pv_energy	*number or null*	Total solar energy generated, [Wh]
total_grid_output_energy	*number or null*	Total grid output energy, [Wh]
total_grid_input_energy	*number or null*	Total grid input energy, [Wh]
total_load_energy	*number or null*	Total load (or off-grid) energy consumed, [Wh]

Example:

Sending:

```
{
    "id": 1,
    "method": "ES.GetStatus",
    "params": {
        "id": 0
    }
}
```

Response:

```
{
    "id": 1,
    "src": "VenusC-mac",
    "result": {
        "id": 0,
        "bat_soc": 98,
        "bat_cap": 2560,
        "pv_power": 0,
        "ongrid_power": 100,
        "offgrid_power": 0,
        "bat_power": 0,
        "total_pv_energy": 0,
        "total_grid_output_energy": 844,
        "total_grid_input_energy": 1607,
        "total_load_energy": 0
```

```
    }
}
```

3.6.2 ES.SetMode

Sending:

Property (params)	Type	Description
id	number	ID of Instance
config	object	Config Parameters

Object: config

Property(config)	Type	Description
mode	*string*	Device power generation mode, including the following modes: "Auto"; "AI"; "Manual"; "Passive".
auto_cfg	*object*	Configuration parameters for Auto mode
ai_cfg	*object*	Configuration parameters for AI mode
manual_cfg	*object*	Configuration parameters for Manual mode
passive_cfg	*object*	Configuration parameters for Passive mode

Object: auto_cfg

Property (auto_cfg)	Type	Description
enable	number	ON: 1; OFF: Set another mode

Object: ai_cfg

Property (ai_cfg)	Type	Description
enable	number	ON: 1; OFF: Set another mode

Object: manual_cfg

Property (manual_cfg)	Type	Description
time_num	number	Time period serial number, Venus C/E supports 0-9
start_time	string	Start time, hours: minutes, [hh:mm]
end_time	string	End time, hours: minutes, [hh:mm]

Property (manual_cfg)	Type	Description
week_set	number	Week, a byte 8 bits, the low 7 bits effective, the highest invalid, 0000 0001 (1) on behalf of Monday open, 0000 0011 (3) on behalf of Monday and Tuesday open, 0111 1111 (127) on behalf of a week
power	number	Setting power,[W]
enable	number	ON: 1; OFF: 0

Object: passive_cfg

Property(passive_cfg)	Type	Description
power	number	Setting power,[W]
cd_time	number	Power countdown,[s]

Response:

Property (result)	Type	Description
id	number	ID of Instance
set_result	boolean	"true":succeeded in setting; "false":failed in setting

Example:

Sending:

```
/* Auto Mode Example */
{
    "id": 1,
    "method": "ES.SetMode",
    "params": {
        "id": 0,
        "config": {
            "mode": "Auto",
            "auto_cfg": {
                "enable": 1
            }
        }
    }
}

/* AI Mode Example */
{
    "id": 1,
    "method": "ES.SetMode",
    "params": {
        "id": 0,
        "config": {
```

```
            "mode": "AI",
            "ai_cfg": {
                "enable": 1
            }
        }
    }
}

/* Manual Mode Example */
{
    "id": 1,
    "method": "ES.SetMode",
    "params": {
        "id": 0,
        "config": {
            "mode": "Manual",
            "manual_cfg": {
                "time_num": 1,
                "start_time": "08:30",
                "end_time": "20:30",
                "week_set": 127,
                "power": 100,
                "enable": 1
            }
        }
    }
}

/* Passive Pattern Example */
{
    "id": 1,
    "method": "ES.SetMode",
    "params": {
        "id": 0,
        "config": {
            "mode": "Passive",
            "passive_cfg": {
                "power": 100,
                "cd_time": 300
            }
        }
    }
}
```

Response:

```
{
    "id": 1,
    "src": "Venus-mac",
    "result": {
        "id": 0,
        "set_result": ture
    }
}
```

3.6.3 ES.GetMode

Sending:

Property (params)	Type	Description
id	number or null	ID of Instance

Response:

Property (result)	Type	Description
id	number or null	ID of Instance
mode	number or null	Auto： Auto mode; AI: AI mode; Manual: manual mode; Passive: Passive control mode
ongrid_power	number or null	Grid-tied power, [W]
offgrid_power	number or null	Off-grid power, [W]
bat_soc	number or null	SOC,[%]

Example：

Sending:

```
{
    "id": 0,
    "method": "ES.GetMode",
    "params": {
        "id": 0
    }
}
```

Response:

```
{
    "id": 0,
    "src": "VenusC-mac",
    "result": {
        "id": 0,
        "mode": "Passive",
        "ongrid_power": 100,
        "offgrid_power": 0,
        "bat_soc": 98
    }
}
```

3.7 EM

The Energy Meter (EM) module contains status information and power measurement data from the energy meter, or data obtained from the current transformer (CT).

- EM.GetStatus: Queries the basic status and data information of the energy meter.

3.7.1 EM.GetStatus

Sending:

Property (params)	Type	Description
id	*number or null*	ID of Instance

Response:

Property (result)	Type	Description
id	*number or null*	ID of Instance
ct_state	*number or null*	CT (Current Transformer) status: 0: Not connected 1: Connected
a_power	*number or null*	Phase A power,[W]
b_power	*number or null*	Phase B power,[W]
c_power	*number or null*	Phase C power,[W]
total_power	*number or null*	Total power,[W]

Sending:

```
{
    "id": 1,
    "method": "EM.GetStatus",
    "params": {
        "id": 0
    }
}
```

Response:

```
{
    "id": 1,
    "src": "VenusC-mac",
    "result": {
        "id": 0,
        "ct_state": 0,
        "a_power":  0,
        "b_power":  0,
        "c_power":  0,
        "total_power":  0
    }
}
```


IV . Devices

This chapter will describe the extent of support for the components and services in this API documentation by different Marstek devices, as well as some proprietary information.

4.1 Venus C/E

- Marstek
- WiFi
- Bluetooth
- Battery
- ES
- EM

4.2 Venus D

- Marstek
- WiFi
- Bluetooth
- Battery
- PV
- ES
- EM

V . Change Logs

This chapter explains the change log for the API documentation.

● 2025-08-09 Modified: Version number Rev 1.0