



Anlagenüberwachung
SUNNY WEBBOX RPC
Bedienungsanleitung

Remote-Procedure-Call Beschreibung
Schnittstellen- und API-Definition

Inhaltsverzeichnis

1	Einleitung	5
1.1	Gültigkeitsbereich	5
1.2	Nomenklatur	5
1.3	Akronyme und Abkürzungen	5
1.4	Referenzierte Dokumente und Quellen	6
2	Überblick über das System	7
2.1	Sunny WebBox RPC	7
2.2	Unterschiede zwischen Sunny WebBox und Sunny WebBox mit Bluetooth	7
3	JavaScript Object Notation	8
3.1	Beispiel	8
4	Prozedur-Konventionen	10
4.1	Prozeduraufruf (Request)	10
4.2	Rückgabewert (Response)	11
4.2.1	JSON	11
4.3	Abfrageintervall	11
5	Schnittstellen	12
5.1	RPC über UDP-Stream	12
5.2	RPC über HTTP	12
6	Objekt-Definitionen	13
6.1	Device-Objekt	13
6.2	Channel-Objekt	14

7	Service-Prozeduren	15
7.1	RPC_GET_PLANT_OVERVIEW	15
7.1.1	Version 1.0.	15
7.2	RPC_GET_DEVICES	17
7.2.1	Version 1.0.	17
7.3	RPC_GET_PROCESS_DATA_CHANNELS	19
7.3.1	Version 1.0.	19
7.4	RPC_GET_PROCESS_DATA	21
7.4.1	Version 1.0.	21
7.5	RPC_GET_PARAMETER_CHANNELS	24
7.5.1	Version 1.0.	24
7.6	RPC_GET_PARAMETER	26
7.6.1	Version 1.0.	26
7.7	RPC_SET_PARAMETER	32
7.7.1	Version 1.0.	32

1 Einleitung

1.1 Gültigkeitsbereich

Dieses Dokument gilt für die RPC-Schnittstelle der Sunny WebBox und der Sunny WebBox mit *Bluetooth*® Wireless Technology.

1.2 Nomenklatur

In diesem Dokument wird die Sunny WebBox mit *Bluetooth*® Wireless Technology als Sunny WebBox mit *Bluetooth* bezeichnet.

1.3 Akronyme und Abkürzungen

API	Application Programmers Interface
HTTP	Hypertext Transfer Protocol
JSON	Javascript Object Notation
UDP	User Data Protocol
URL	Uniform Resource Locator
RAS	Remote Access Service
RFC	Request for Comment
RPC	Remote Procedure Call

1.4 Referenzierte Dokumente und Quellen

[1]

RFC 4627: The application/json Media Type for Javascript Object Notation (JSON)

<http://www.ietf.org/rfc/rfc4627.txt?number=4627>

[2]

JSON-RPC 1.1 Working Draft August 2006

<http://json-rpc.org/wd/JSON-RPC-1-1-WD-20060807.html>

[3]

Introducing JSON

<http://www.json.org>

[4]

The MD5 Message-Digest Algorithm

<http://www.ietf.org/rfc/rfc1321.txt?number=1321>

2 Überblick über das System

2.1 Sunny WebBox RPC

Die Datenlogger Sunny WebBox und Sunny WebBox mit *Bluetooth* erfassen kontinuierlich sämtliche Daten einer Photovoltaik-Anlage. Diese werden über ein konfigurierbares Intervall gemittelt und zwischengespeichert. Zur Auswertung und Visualisierung können die Daten in regelmäßigen Intervallen an das Sunny Portal übermittelt werden.

Über die RPC-Schnittstelle der Sunny WebBox / - mit *Bluetooth* können ausgewählte Daten der PV-Anlage an eine Gegenstelle über ein RPC-Protokoll (Remote-Procedure-Call-Protokoll) übertragen werden.

Hierzu stellt die Sunny WebBox einen Pool von Service-Prozeduren (siehe Kapitel 6 „Objekt-Definitionen“, Seite 13) zur Verfügung, welche über ein Remote-Procedure-Call-Protokoll (RPC-Protokoll) von der Gegenstelle über eine Netzwerk- oder RAS-Verbindung aufgerufen werden können. Als Datenaustauschformat kommt die JavaScript Object Notation (JSON) (siehe Kapitel 3 „JavaScript Object Notation“, Seite 8) zum Einsatz.

2.2 Unterschiede zwischen Sunny WebBox und Sunny WebBox mit *Bluetooth*



Unterschiedliche Darstellung bei Sunny WebBox und Sunny WebBox mit *Bluetooth*

Die Darstellung der Anlagendaten, die über die RPC-Schnittstelle übermittelt werden, unterscheidet sich bei der Sunny WebBox und der Sunny WebBox mit *Bluetooth*.

In diesem Dokument sind ausschließlich Beispielantworten der Sunny WebBox aufgeführt, nicht jedoch Beispielantworten der Sunny WebBox mit *Bluetooth*.

Eine Auflistung der Messwerte und Parameter der Sunny WebBox mit *Bluetooth* finden Sie in der Technischen Beschreibung „Messwerte und Parameter“ unter www.SMA-Solar.com.

Verfügbare Service-Prozeduren	Sunny WebBox	Sunny WebBox mit <i>Bluetooth</i>
RPC_GET_PLANT_OVERVIEW	+	+
RPC_GET_DEVICES	+	+
RPC_GET_PROCESS_DATA_CHANNELS	+	+
RPC_GET_PROCESS_DATA	+	+
RPC_GET_PARAMETER_CHANNELS	+	–
RPC_GET_PARAMETER	+	–
RPC_SET_PARAMETER	+	–

3 JavaScript Object Notation

Die JavaScript Object Notation (JSON) wird auf der Website <http://www.json.org> beschrieben und erklärt.

3.1 Beispiel

Das folgende Beispiel zeigt die Darstellung einer Geräteliste. Es definiert ein Objekt, welches aus den Werten "totalDevicesReturned" und "devices" besteht.

"totalDevicesReturned" ist eine Zahl und hat den Wert 4. Das Array "devices" hat 2 Felder mit jeweils einem Geräte-Objekt (siehe Kapitel 6.1 „Device-Objekt“, Seite 13) welches wiederum verschachtelte Geräte-Objekte enthält.

```
RPC={
  "totalDevicesReturned":4,
  "devices":
  [
    {
      "key":"SCC250H9:1390148531",
      "name":"Sunny Central E1",
      "children":
      [
        {
          "key":"SCBFS016:8945",
          "name":"Sunny BFS E1",
          "children":null
        },
        {
          "key":"SMU8b004:2567",
          "name":"String Monitoring Unit E1",
          "children":null
        }
      ]
    },
    {
      "key":"SCC250H9:1390148538",
      "name":"Sunny Central E2",
      "children":
      [
```



```
{
  "key": "SCBFS016:8956",
  "name": "Sunny BFS E2",
  "children": null
},
{
  "key": "SMU8b004:2534",
  "name": "String Monitoring Unit E2",
  "children": null
}
]
}
]
```

4 Prozedur-Konventionen

Alle verwendeten Bezeichner sind case-sensitiv. Das heißt, "Power" und "power" bezeichnen beispielsweise zwei verschiedene Objekte. Alle Zeichen werden als Unicode im UTF-8 Format übertragen.

4.1 Prozeduraufruf (Request)

Jeder Request besteht aus einem serialisierten JSON-Objekt, welches folgende obligatorische Mitglieder besitzt:

- **version** – Eine Zeichenfolge, welche die zugrunde liegende RPC-Version festlegt.
- **proc** – Eine Zeichenfolge, die den Namen der aufzurufenden Prozedur enthält.
- **id** – Eine beliebige Zeichenfolge (max. 16 Zeichen), welche der Zuordnung einer Antwort zur Anfrage dient.
- **format** – Eine Zeichenfolge, die das Datenaustauschformat des Prozedurergebnisses festlegt (siehe Kapitel 4.2 „Rückgabewert (Response)“, Seite 11).
- **passwd** – Eine Zeichenfolge, die den Hashwert des Passworts für den gewünschten Zugriffs-Level (Benutzer, Installateur) enthält. Der Hashwert wird mittels MD5-Algorithmus (siehe [4]) berechnet. Die Passwortvergabe erfolgt über die Sicherheitseinstellungen der Sunny WebBox. Wenn das Objekt nicht übertragen wird, wird automatisch der Benutzer-Level angenommen.
- **params** – Ein Objekt, dessen Elemente der Prozedur als Argumente übergeben werden. Jeder Parameter muss in Form eines benannten JSON-Objekts vorliegen. Die Reihenfolge ist somit beliebig. Die Anzahl der Parameter hängt von der jeweiligen Service-Prozedur ab (siehe Kap. 7). Wenn die aufgerufene Prozedur keine Argumente erwartet entfällt der Eintrag.

4.2 Rückgabewert (Response)

Das Datenaustauschformat der Rückgabe wird durch die im Prozeduraufruf übermittelte Zeichenfolge Format festgelegt.

Derzeit stehen folgende Formate zur Verfügung:

- JSON

4.2.1 JSON

Der Rückgabewert eines Prozeduraufrufs besteht aus einem serialisierten JSON-Objekt, welches aus den folgenden obligatorischen Objekten besteht:

- **version** – Eine Zeichenfolge, welche die zugrunde liegende RPC-Version festlegt.
- **proc** – Eine Zeichenfolge, die den Namen der ausgeführten Prozedur enthält.
- **id** – Eine Zeichenfolge, welche der Zuordnung von Anfrage und Antwort dient. Enthält die ID aus der zugehörigen Anfrage.
- **result** – Das Ergebnis der Prozedurausführung als serialisiertes JSON-Objekt. Wenn durch Auftreten eines Fehlers die Prozedur nicht erfolgreich ausgeführt werden konnte, wird an Stelle dieses Objekts das error-Objekt übermittelt.
- **error** – Ein Objekt, welches Im Fehlerfall eine Zeichenkette mit einer Beschreibung des aufgetretenen Fehlers enthält. Bei fehlerfreier Prozedurausführung wird dieses Objekt nicht übermittelt.

4.3 Abfrageintervall

Der Zeitraum zwischen zwei Abfragen sollte nicht kleiner als 30 Sekunden sein.

5 Schnittstellen

Die Sunny WebBox stellt zwei unterschiedliche Zugriffsmöglichkeiten zur Verfügung, die sich durch Implementierungsaufwand und Laufzeitressourcenverbrauch voneinander unterscheiden.

5.1 RPC über UDP-Stream

Der Prozeduraufruf wird im Nutzdatenteil des UDP-Protokolls auf Port 34268 an die Sunny WebBox übertragen. Antworten werden ebenfalls an den Port 34268 gerichtet.

Der UDP-Transport erfordert einen relativ geringen Implementierungsaufwand auf der Client-Seite und spart Laufzeitressourcen. Für die Kommunikation über die Grenzen lokaler Netzwerke hinweg ist in der Regel eine Portfreischaltung in den entsprechenden Firewalls erforderlich.

5.2 RPC über HTTP

Der Datenaustausch findet über das Hypertext Transfer Protokoll per TCP/IP-Verbindung auf dem in der Sunny WebBox konfigurierbaren Webserver-Port statt.

Die Voreinstellung ist Port 80.

Der URL für alle Anfragen lautet: `http://IP-Adresse/rpc`

IP-Adresse ist dabei die aktuell eingestellte IP-Adresse der Sunny WebBox. Die Voreinstellung lautet 192.168.0.168.

Damit sieht der Default-URL folgendermaßen aus: `http://192.168.0.168/rpc`

Der Prozeduraufruf wird per HTTP POST im Rumpf der HTTP-Anfrage als serialisiertes JSON-Objekt gemäß den in Kapitel 4.1 getroffenen Konventionen übertragen.

Sowohl der clientseitige Implementierungsaufwand, als auch der Ressourcenverbrauch sind relativ hoch. Im Regelfall erfolgt die Kommunikation über den Standard-Port 80, so dass keine Veränderungen an etwaigen Firewalls vorgenommen werden müssen.

6 Objekt-Definitionen

Dieses Kapitel definiert den Aufbau häufig verwendeter Objekte auf Basis der JSON-Syntax. Die Werte der Objektmitglieder kennzeichnen dabei in der Beschreibung deren Datentyp. Alle Definitionen gelten für andere Datenaustauschformate analog.

6.1 Device-Objekt

Beschreibt ein Gerät innerhalb der Anlage (z. B. Sunny Boy, Sunny Sensor Box).

```
RPC={  
  "key": "string",  
  "name": "string" oder null (optional),  
  "channels": [array] oder null (optional),  
  "children": [array] oder null (optional)  
}
```

- key:** Ein eindeutiger Geräteschlüssel (z. B. "SB21TL06:2000106925").
- name:** Der benutzerdefinierbare Name des Gerätes (z. B. "WR links"). Die Angabe des Elements ist optional. Wenn das Element angegeben wird, aber kein Name vergeben wurde, wird null eingetragen.
- channels:** Ein Array von Channel-Objekten des Gerätes. Die Angabe des Elements ist optional. Wenn das Element angegeben wird, aber keine Kanäle existieren, wird null eingetragen.
- children:** Ein Array von Objekten hierarchisch untergeordneter Geräte. Die Angabe des Elements ist optional. Wenn das Element angegeben wird, aber keine Untergeräte existieren, wird null eingetragen.

6.2 Channel-Objekt

Beschreibt einen Prozeßdaten- oder Parameter-Kanal eines Gerätes.

```
RPC={  
    "meta": "string",  
    "name": "string" (optional),  
    "value": "string",  
    "unit": "string" (optional),  
    "min": "string" (optional),  
    "max": "string" (optional),  
    "options": [array] (optional)  
}
```

- meta: Der Meta Name der den Kanal eindeutig identifiziert (z. B. "ExtSollrr").
- name: Der übersetzte Anzeigename (z. B. "Einstrahlung extern"). Die Angabe des Elements ist optional.
- value: Der Wert des Kanals (z. B. "843"). Die Angabe des Elements ist optional.
- unit: Die Einheit des Kanals (z. B. "W/m²"). Bei Kanälen die keine Einheit besitzen wird ein Leerstring eingetragen.
- min: Der minimale Wert den ein Kanal annehmen kann. Die Angabe des Elements ist optional.
- max: Der maximale Wert den ein Kanal annehmen kann. Die Angabe des Elements ist optional.
- options: Eine Liste mit möglichen Werten die ein Parameter-Kanal annehmen kann. Die Angabe des Elements ist optional.

7 Service-Prozeduren

Dieses Kapitel beschreibt den Aufbau der zur Verfügung stehenden Service-Prozeduren.

Es wird für jede Prozedur eine Kurzbeschreibung ihrer Aufgabe gegeben. Anschließend ist der Aufbau der Anfrage dargestellt. Variable Elemente werden durch Platzhalter in Großbuchstaben repräsentiert. Auf die Beschreibung der Platzhalter VERSION, FORMAT und ID für jede Prozedur wird an dieser Stelle verzichtet, weil deren Bedeutung bereits in Kapitel 4 erläutert wurde und hier selten Unterschiede zwischen den Prozeduren gemacht werden müssen.

7.1 RPC_GET_PLANT_OVERVIEW

7.1.1 Version 1.0

Liefert ein Objekt mit den folgenden Anlagendaten:

- PAC
- E-HEUTE
- E-TOTAL
- STATUS
- FEHLER

Aufbau:

```
RPC={  
  "version": "1.0",  
  "proc": "GetPlantOverview",  
  "id": "ID",  
  "format": "FORMAT"  
}
```

Beispielanfrage:

```
RPC={  
  "version": "1.0",  
  "proc": "GetPlantOverview",  
  "id": "1",  
  "format": "JSON"  
}
```

Beispielantwort:

```

RPC={
  "version": "1.0",
  "proc": "GetPlantOverview",
  "id": "1",
  "result":
  {
    "overview":
    [
      {
        "meta": "GriPwr",
        "name": "Momentanleistung",
        "value": "4250",
        "unit": "W"
      },
      {
        "meta": "GriEgyTdy",
        "name": "Tagesenergie",
        "value": "45.23",
        "unit": "kWh"
      },
      {
        "meta": "GriEgyTot",
        "name": "Gesamtenergie",
        "value": "7821",
        "unit": "kWh"
      },
      {
        "meta": "OpStt",
        "name": "Status",
        "value": "MPP",
        "unit": null
      },
      {
        "meta": "Msg",

```



```
"name": "Fehler",  
"value": null,  
"unit": null  
}  
]  
}  
}
```

Folgende Daten wurden übermittelt:

Pac = 4250 W,
E-Heute = 45.23 kWh,
E-Total = 7821 kWh,
Staus = MPP,
kein Fehler

7.2 RPC_GET_DEVICES

7.2.1 Version 1.0

Liefert eine hierarchische Liste aller erfassten Anlagengeräte.

Aufbau:

```
RPC={  
  "version": "1.0",  
  "proc": "GetDevices",  
  "id": "ID",  
  "format": "FORMAT"  
}
```

Beispielanfrage:

```
RPC={  
  "version": "1.0",  
  "proc": "GetDevices",  
  "id": "1",  
  "format": "JSON"  
}
```

Beispielantwort:

```

RPC={
  "version": "1.0",
  "proc": "GetDevices",
  "id": "1",
  "result":
  {
    "totalDevicesReturned": 6,
    "devices":
    [
      {
        "key": "SCC250H9: 1390148531",
        "name": "Sunny Central E1",
        "children":
        [
          {
            "key": "SCBFS016:8945",
            "name": "Sunny BFS E1",
            "children": null
          },
          {
            "key": "SMU8b004:2567",
            "name": "String Monitoring Unit E1",
            "children": null
          }
        ]
      },
      {
        "key": "SCC250H9:1390148538",
        "name": "Sunny Central E2",
        "children":
        [
          {
            "key": "SCBFS016:8956",
            "name": "Sunny BFS E2",

```

```

        "children": null
    },
    {
        "key": "SMU8b004:2534",
        "name": "String Monitoring Unit E2",
        "children": null
    }
]
}
]
}
}
}

```

7.3 RPC_GET_PROCESS_DATA_CHANNELS

7.3.1 Version 1.0

Liefert eine Liste mit den Meta-Namen der verfügbaren Prozessdaten-Kanäle für einen bestimmten Gerätetyp.

Aufbau:

```

RPC={
    "version": "1.0",
    "proc": "GetProcessDataChannels",
    "id": "ID",
    "format": "FORMAT",
    "params":
    {
        DEVICE_KEY
    }
}

```

DEVICE_KEY: Der Geräteschlüssel eines Gerätes für dessen Typ die Prozessdaten-Kanäle geliefert werden sollen.

Beispielanfrage:

```
RPC={
  "version": "1.0",
  "proc": "GetProDataChannels",
  "id": "1",
  "format": "JSON",
  "params":
  {
    "device": "WR715-19:263415747"
  }
}
```

Beispielantwort:

```
RPC={
  "version": "1.0",
  "proc": "GetProcessDataChannels",
  "id": "1",
  "result":
  {
    "WR715-19:263415747":
    [
      "Upv-Soll",
      "h-Total",
      "Zac",
      "Status",
      "E-Total",
      "Upv-Ist",
      "Riso",
      "Uac",
      "Pac",
      "Fehler-Cnt",
      "Ipv",
      "Netz-Ein",
      "Seriennummer",
      "Fac",
      "Fehler"
    ]
  }
}
```

```

        "Iac-Ist"
    ]
}
}

```

7.4 RPC_GET_PROCESS_DATA

7.4.1 Version 1.0

Liefert Prozessdaten für bis bis zu 5 Geräte pro Anfrage.

Aufbau:

```

RPC={
    "version": "1.0",

    "proc": "GetProcessData",
    "id": "ID",
    "format": "FORMAT",
    "params":
    {
        "DEVICES":
        [
            {
                "key": DEVICE_KEY,
                "channels": [CHANNELS]
            }
        ]
    }
}

```

Als Parameter ist eine Liste mit den Geräteschlüsseln zu übergeben, von denen die Prozessdaten geliefert werden sollen. Jedem Gerät kann eine Auswahl von gewünschten Prozessdaten übergeben werden. Fehlt diese Auswahl, so werden alle Prozessdaten übermittelt.

DEVICES: Ein Array, welches Objekte mit den Geräteschlüsseln der Geräte enthält, von denen Prozessdaten geliefert werden sollen, sowie optionale CHANNELS.

DEVICE_KEY: Der zugehörige Geräteschlüssel (siehe Kapitel 6.1 „Device-Objekt“, Seite 13).

CHANNELS: Ein Array, welches die Meta-Namen der gewünschten Prozessdaten enthält. Die verfügbaren Meta-Namen können mit dem Kommando `RPC_GET_PROCESS_DATA_CHANNELS` ermittelt werden.

Beispielanfrage:

```
RPC={
  "version": "1.0",
  "proc": "GetProcessData",
  "id": "1",
  "format": "JSON",
  "params":
  {
    "devices":
    [
      {
        "key": "WR715-19:263415747",
        "channels": null
      },
      {
        "key": "WR715-19:263415748",
        "channels":
        [
          "Pac"
        ]
      }
    ]
  }
}
```

Beispielantwort:

```
RPC={
  "version": "1.0",
  "proc": "GetProcessData",
  "id": "1",
  "result":
  {
```

```
"devices":  
[  
  {  
    "key": "WR715-19:263415747",  
    "channels":  
    [  
      {  
        "meta": "E-Total",  
        "name": null,  
        "value": "1160.987",  
        "unit": "kWh"  
      },  
      {  
        "meta": "Fac",  
        "name": null,  
        "value": "49.98",  
        "unit": "Hz"  
      },  
      {  
        "meta": "Zac",  
        "name": null,  
        "value": "1.346",  
        "unit": "Ohm"  
      }  
    ]  
  },  
  {  
    "key": "WR715-19:263415748",  
    "channels":  
    [  
      {  
        "meta": "Pac",  
        "name": null,  
        "value": "630",  
        "unit": "W"
```

```

    }
  ]
}
]
}
}

```

7.5 RPC_GET_PARAMETER_CHANNELS

7.5.1 Version 1.0

Liefert eine Liste mit den Meta-Namen der verfügbaren Parameter-Kanäle für einen bestimmten Gerätetyp in Abhängigkeit vom Zugriffslevel. Der Level wird durch Übermittlung des MD5-Hashwertes des zugehörigen Passworts im Request-Header festgelegt.

Aufbau

```

RPC={
  "version": "1.0",
  "proc": "GetParameterChannels",
  "id": "ID",
  "format": "FORMAT",
  "passwd" : "PASSWORD",
  "params":
  {
    "key": DEVICE_KEY
  }
}

```

PASSWORD: Der MD5-kodierte Hashwert des Passwortes für den gewünschten Zugriffslevel.

DEVICE_KEY: Der Geräteschlüssel eines Gerätes für dessen Typ die Parameter-Kanäle geliefert werden sollen.

Beispielanfrage

```

RPC={
  "version": "1.0",
  "proc": "GetParameterChannels",
  "id": "1",
  "format": "JSON",
  "passwd": "a289fa4252ed5af8e3e9f9bee545c172",

```



```

"params":
{
  "device": "WR715-19:263415747"
}
}

```

Beispielantwort:

```

RPC={
  "version": "1.0",
  "proc": "GetParameterChannels",
  "id": "1",
  "result":
  {
    "WR715-19:263415747":
    [
      "Plimit",
      "SMA-Grid-Guard",
      "SMA-SN",
      "Betriebsart",
      "Control",
      "Ripple-Ctl-Frq",
      "PowerBalancer",
      "Usoll-Konst",
      "Upv-Start",
      "Default",
      "T-Start",
      "Ripple-Ctl-Lev",
      "Storage",
      "Ripple-Ctl-Rcvr",
      "Software-SRR",
      "T-Stop",
      "Software-BFR",
      "Hardware-BFS"
    ]
  }
}

```

7.6 RPC_GET_PARAMETER

7.6.1 Version 1.0

Liefert die Parameterwerte von bis zu 5 Geräten in Abhängigkeit vom Zugriffslevel. Der Level wird durch Übermittlung des MD5-Hashwertes des zugehörigen Passworts im Request-Header festgelegt.

Aufbau

```
RPC={
  "version": "1.0",
  "proc": "GetParameter",
  "id": "ID",
  "format": "FORMAT",
  "passwd": "PASSWORT",
  "params":
  {
    "DEVICES":
    [
      {
        "key": DEVICE_KEY,
        "channels": [CHANNELS]
      }
    ]
  }
}
```

Als Parameter ist eine Liste mit den Geräteobjekten zu übergeben, von denen die Parameter geliefert werden sollen. Jedem Gerät kann eine Auswahl von gewünschten Parametern übergeben werden. Fehlt diese Auswahl, so werden alle Parameter übermittelt.

PASSWORT: Der MD5-kodierte Hashwert des Passwortes für den gewünschten Zugriffslevel.

DEVICES: Ein Array, welches die Geräte-Objekte enthält, von denen Parameter geliefert werden sollen, sowie eine optionale Auswahl bestimmter Kanäle.

DEVICE_KEY: Der zugehörige Geräteschlüssel (siehe Kapitel 6.1 „Device-Objekt“, Seite 13).

CHANNELS: Ein Array, welches die Meta-Namen der gewünschten Prozessdaten enthält. Die verfügbaren Meta-Namen können mit dem Kommando `RPC_GET_PROCESS_DATA_CHANNELS` ermittelt werden.

Beispielanfrage

```
RPC={
  "version": "1.0",
  "proc": "GetParameter",
  "id": "1",
  "format": "JSON",
  "passwd": "a289fa4252ed5af8e3e9f9bee545c172",
  "params":
  {
    "devices":
    [
      {
        "key": "WR715-19:263415747"
      }
    ]
  }
}
```

Beispielantwort:

```
RPC={
  "version": "1.0",
  "id": "1",
  "format": "JSON",
  "proc": "GetParameter",
  "result":
  {
    "devices":
    [
      {
        "key": "WR21TL06:2000101000"
        "channels":
        [
          {
            "min": "0",
            "max": "7",
            "meta": "Betriebsart",

```

```

"options":
[
  "Stop",
  "Konstantspg.",
  "Mpp-Betrieb",
  "Res1",
  "Res2",
  "Res3",
  "Res4",
  "Res5"
],
"value": "Mpp-Betrieb",
"name": "Betriebsart",
"unit": ""
},
{
  "min": "2150",
  "max": "2150",
  "meta": "Plimit",
  "value": "2150",
  "name": "Plimit",
  "unit": "W"
},
{
  "min": "0",
  "max": "4294900000",
  "meta": "SMA-SN",
  "value": "2000101000",
  "name": "SMA-SN",
  "unit": ""
},
{
  "min": "125",
  "max": "600",
  "meta": "Upv-Start",

```

```
"value": "150",
"name": "Upv-Start",
"unit": "V"
},
{
  "min": "1",
  "max": "300",
  "meta": "T-Stop",
  "value": "4",
  "name": "T-Stop",
  "unit": "s"
},
{
  "min": "125",
  "max": "600",
  "meta": "Usoll-Konst",
  "value": "600",
  "name": "Usoll-Konst",
  "unit": "V"
},
{
  "min": "5",
  "max": "300",
  "meta": "T-Start",
  "value": "10",
  "name": "T-Start",
  "unit": "s"
},
{
  "min": "0",
  "max": "100",
  "meta": "Software-SRR",
  "value": "2",
  "name": "Software-SRR",
  "unit": "Version"
```

```

},
{
  "min": "0.005",
  "max": "4",
  "meta": "dFac-Max",
  "value": "0",
  "name": "dFac-Max",
  "unit": "Hz/s"
},
{
  "min": "0",
  "max": "7",
  "meta": "Storage",
  "options":
  [
    "permanent",
    "volatile",
    "Res1",
    "Res2",
    "Res3",
    "Res4",
    "Res5",
    "Res6"
  ],
  "value": "permanent",
  "name": "Storage",
  "unit": ""
},
{
  "min": "0",
  "max": "100",
  "meta": "Software-BFR",
  "value": "2",
  "name": "Software-BFR",
  "unit": "Version"
}

```

```
    },  
    {  
      "min": "0",  
      "max": "100",  
      "meta": "Hardware-BFS",  
      "value": "1",  
      "name": "Hardware-BFS",  
      "unit": "Version"  
    }  
  ]  
}  
]  
}  
}
```

7.7 RPC_SET_PARAMETER

7.7.1 Version 1.0

Setzt Parameterwerte von bis zu 5 Geräten und liefert die im Request übergebene Geräteliste mit den zugehörigen aktuellen Parameterwerten zurück. Eine Überprüfung, ob jeder Parameterwert erfolgreich gesetzt werden konnte, ist von der aufrufenden Anwendung vorzunehmen.

Aufbau

```
RPC={
  "version": "1.0",
  "proc": "SetParameter",
  "id": "ID",
  "format": "FORMAT",
  "passwd": "PASSWORT",
  "params":
  {
    "DEVICES":
    [
      {
        key,
        "channels": [CHANNELS]
      }
    ]
  }
}
```

Als Parameter ist eine Liste mit den Geräteobjekten zu übergeben, deren Parameterwerte geändert werden sollen. Jedes Geräteobjekt beinhaltet eine Liste mit den Parametern die zu setzen sind.

Das Setzen der Parameter erfolgt synchron. Somit sind die Antwortzeiten von der Anzahl der zu setzenden Parameter abhängig. Für das nachfolgende Beispiel liegt die Antwortzeit bei ca. 10 Sekunden.

PASSWORT: Der MD5-kodierte Hashwert des Passwortes für den gewünschten Zugriffslevel.

DEVICES: Ein Array, welches Objekte mit den Geräteschlüsseln der Geräte enthält, sowie ein Array mit den CHANNELS deren Parameterwerte gesetzt werden sollen.

DEVICE_KEY: Der zugehörige Geräteschlüssel (siehe Kapitel 6.1 „Device-Objekt“, Seite 13).

CHANNELS: Ein Array, welches die Channel-Objekte enthält, die für das jeweilige Gerät gesetzt werden sollen. Eine Liste der verfügbaren Parameter-Channels kann mit dem Kommando `RPC_GET_PARAMETER_CHANNELS` ermittelt werden.

Beispielanfrage:

```
RPC={
  "version": "1.0",
  "proc": "SetParameter",
  "id": "1",
  "format": "JSON",
  "passwd": "a289fa4252ed5af8e3e9f9bee545c172",
  "params":
  {
    "devices":
    [
      {
        "key": "WR21TL06:2000101000",
        "channels":
        [
          {
            "meta": "Betriebsart",
            "value": "Mpp-Betrieb"
          }
        ]
      },
      {
        "key": "WR21TL06:2000101001",
        "channels":
        [
          {
            "meta": "Betriebsart",
            "value": "Stop"
          }
        ]
      }
    ]
  }
}
```

```

    }
}

```

Beispielantwort:

```

RPC={
  "version": "1.0",
  "id": "1",
  "format": "JSON",
  "proc": "SetParameter",
  "result":
  {
    "devices":
    [
      {
        "key": "WR21TL06:2000101000",
        "channels":
        [
          {
            "min": "0",
            "max": "7",
            "meta": "Betriebsart",
            "options":
            [
              "Stop",
              "Konstantspg.",
              "Mpp-Betrieb",
              "Res1",
              "Res2",
              "Res3",
              "Res4",
              "Res5"
            ],
            "value": "Mpp-Betrieb",
            "name": "Betriebsart",
            "unit": ""
          }
        ]
      }
    ]
  }
}

```

```
]
},
{
  "key": "WR21TL06:2000101001",
  "channels":
  [
    {
      "min": "0",
      "max": "7",
      "meta": "Betriebsart",
      "options":
      [
        "Stop",
        "Konstantspg.",
        "Mpp-Betrieb",
        "Res1",
        "Res2",
        "Res3",
        "Res4",
        "Res5"
      ],
      "value": "Stop",
      "name": "Betriebsart",
      "unit": ""
    }
  ]
}
]
```


Rechtliche Bestimmungen

Die in diesen Unterlagen enthaltenen Informationen sind Eigentum der SMA Solar Technology AG. Die Veröffentlichung, ganz oder in Teilen, bedarf der schriftlichen Zustimmung der SMA Solar Technology AG. Eine innerbetriebliche Vervielfältigung, die zur Evaluierung des Produktes oder zum sachgemäßen Einsatz bestimmt ist, ist erlaubt und nicht genehmigungspflichtig.

Konformitätserklärung

Hiermit erklärt SMA Solar Technology AG, dass sich das beschriebene Gerät/die beschriebenen Geräte in Übereinstimmung mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften der Richtlinie 1999/5/EG befindet/befinden. Die vollständige CE-Konformitätserklärung finden Sie unter www.SMA-Solar.com.

Warenzeichen

Alle Warenzeichen werden anerkannt, auch wenn diese nicht gesondert gekennzeichnet sind. Fehlende Kennzeichnung bedeutet nicht, eine Ware oder ein Zeichen seien frei.

Die *Bluetooth*® Wortmarke und Logos sind eingetragene Warenzeichen der Bluetooth SIG, Inc. und jegliche Verwendung dieser Marken durch die SMA Solar Technology AG erfolgt unter Lizenz.

QR Code® ist eine eingetragene Marke der DENSO WAVE INCORPORATED.

SMA Solar Technology AG

Sonnenallee 1
34266 Niestetal
Deutschland

Tel. +49 561 9522-0
Fax +49 561 9522-100
www.SMA.de
E-Mail: info@SMA.de

© 2004 bis 2012 SMA Solar Technology AG. Alle Rechte vorbehalten.

SMA Solar Technology

www.SMA-Solar.com

SMA Solar Technology AG

www.SMA.de

SMA Australia Pty. Ltd.

www.SMA-Australia.com.au

SMA Benelux bvba/sprl

www.SMA-Benelux.com

SMA Beijing Commercial Company Ltd.

www.SMA-China.com.cn

SMA Central & Eastern Europe s.r.o.

www.SMA-Czech.com

SMA France S.A.S.

www.SMA-France.com

SMA Hellas AE

www.SMA-Hellas.com

SMA Ibérica Tecnología Solar, S.L.U.

www.SMA-Iberica.com

SMA Solar India Pvt. Ltd.

www.SMA-India.com

SMA Italia S.r.l.

www.SMA-Italia.com

SMA Japan K.K.

www.SMA-Japan.com

SMA Technology Korea Co., Ltd.

www.SMA-Korea.com

SMA Middle East LLC

www.SMA-Me.com

SMA Portugal - Niestetal Services Unipessoal Lda

www.SMA-Portugal.com

SMA Solar (Thailand) Co., Ltd.

www.SMA-Thailand.com

SMA Solar UK Ltd.

www.SMA-UK.com

