

SunEnergyXT_Batterie_Lokale_API

Deutsche Version

1. Protokollübersicht und Kommunikationsmechanismus

Dieses Protokoll dient der Interaktion zwischen dem lokalen Hub (Host-Computer) und SunEnergyXT-Geräten.

- **Kommunikationsprotokoll:** TCP / IP
- **Datenformat:** JSON
- **Nachrichtenstruktur:** Einheitliches Umschlagformat `{"code": <Int>, "data": { ... }}`
- **Port:** 8000 (Standard)

Kernnachrichtenstruktur

Alle Interaktionen (Empfangen/Senden) folgen der unten stehenden Standard-JSON-Struktur:

JSON

```
1 {
2     "code": <Dezimalwert des Befehlsworts>,
3     "data": {
4         "<Key>": <Value>,
5         ...
6     }
7 }
```

- **code:** Entspricht dem **Dezimalwert** des "Befehlsworts" (Command Word) im Protokolldokument.
- **Beispiel:** Wenn das Befehlswort `0x6052` ist, dann ist `code = 24658`.
- **data:** Enthält die spezifischen Schlüssel-Wert-Paare der Geschäftsfelder.

2. Datenmeldung (Batteriegerät -> Lokaler Client)

Nachdem das Gerät eine Verbindung hergestellt hat, sendet es den aktuellen Status in regelmäßigen Abständen (Broadcast).

TCP-Nachrichtenfeld	Feldbeschreibung	Feldtyp	Lese-/Schreibtyp
t598	Lokaler modus	bool	RW
t700_1	Speicherlade-Modus	bool	RW
t701_1	Autolade-Modus	bool	RW
t702_1	Haushaltsgeräte-Modus	bool	RW
t728	EV AC-Aktivmodus	bool	RW
t362	System Entladegrenze	int	RW
363	System Ladegrenze	int	RW
t720	Entlade-Abschaltgrenze im Haushaltsgeräte-Modus	int	RW
t721	Entlade-Abschaltgrenze im Autolade-Modus	int	RW
t727	Lade-Abschaltgrenze im Speicherlade-Modus	int	RW
t590	System Ladeleistung	int	RW
t596	Leerlauf Abschaltzeit	int	RW
t597	Niedrigstand Abschaltzeit	int	RW
t211	Speicherlevel	int	R
t592	Kopfspeicher	int	R
t593	Erweiterungsspeicher 1	int	R
t594	Erweiterungsspeicher 2	int	R
t595	Erweiterungsspeicher 3	int	R
t1001	Erweiterungsspeicher 4	int	R
t1002	Erweiterungsspeicher 5	int	R
t1003	Erweiterungsspeicher 6	int	R
t1004	Erweiterungsspeicher 7	int	R

t507	Kopf HW-Entladegrenze	int	R
t508	Kopf HW-Ladegrenze	int	R
t509	Erweiterung 1 HW-Entladegrenze	int	R
t510	Erweiterung 1 HW-Ladegrenze	int	R
t511	Erweiterung 2 HW-Entladegrenze	int	R
t512	Erweiterung 2 HW-Ladegrenze	int	R
t513	Erweiterung 3 HW-Entladegrenze	int	R
t514	Erweiterung 3 HW-Ladegrenze	int	R
t948	Erweiterung 4 HW-Entladegrenze	int	R
t949	Erweiterung 4 HW-Ladegrenze	int	R
t950	Erweiterung 5 HW-Entladegrenze	int	R
t951	Erweiterung 5 HW-Ladegrenze	int	R
t952	Erweiterung 6 HW-Entladegrenze	int	R
t953	Erweiterung 6 HW-Ladegrenze	int	R
t954	Erweiterung 7 HW-Entladegrenze	int	R
t955	Erweiterung 7 HW-Ladegrenze	int	R

2.1 Verbindungsauflaufbau und Abhören

1. **Lokalen Modus am Gerät aktivieren:** Schalten Sie den Lokalen Modus in der App unter **Batterieeinstellungen -> Local Mode** ein und rufen Sie die IP ab. (Oder rufen Sie die Geräte-IP automatisch ab, indem Sie die mDNS-Dienstadresse im LAN scannen: `_hp-bk215`).

2. TCP-Verbindung herstellen:

- Der **Lokale Client** agiert als **TCP-Client**.
- **Ziel-IP:** Geräte-IP (aus der App abgerufen, z. B. `192.168.0.85`).
- **Port:** 8000.

3. Datenempfang:

- Halten Sie eine dauerhafte TCP-Verbindung aufrecht und hören Sie den Port ab.

- Fangen Sie Nachrichten zeilenweise (`\r\n`) oder durch JSON-Klammerabgleich ab.

2.2 Analyse der Meldungsnachricht

Das vom Gerät gesendete Befehlswort ist `0x6052` (Hexadezimal) -> `24658` (Dezimal).

Beispiel für empfangenes JSON:

JSON

```
1 {
2     "code": 24658,
3     "data": {
4         "t362": 1,
5         "t363": 80,
6         "t590": 200,
7         "t596": 1440,
8         "t597": 1440,
9         "t598": 1,
10        "t700_1": -1,
11        "t701_1": -1,
12        "t720": -1,
13        "t721": -1,
14        "t727": -1,
15        "t728": -1,
16        "t211": 25,
17        "t592": 21,
18        "t507": 1,
19        "t508": 90,
20        "t593": 30,
21        "t509": 1,
22        "t510": 90,
23        "t594": -1,
24        "t511": -1,
25        "t512": -1,
26        "t595": -1,
27        "t513": -1,
28        "t514": -1,
29        "t1001": -1,
30        "t948": -1,
31        "t949": -1,
32        "t1002": -1,
33        "t950": -1,
34        "t951": -1,
35        "t1003": -1,
36        "t952": -1,
37        "t953": -1,
38        "t1004": -1,
39        "t954": -1,
```

```
40      "t955": -1
41  }
42 }
```

3. Befehlsübermittlung (Lokaler Client -> Batteriegerät)

3.1 Übermittlungsprozess

4. **Basis konvertieren:** Wandeln Sie die Hexadezimalzahl in eine **Dezimalzahl** um und tragen Sie sie in `code` ein. Das Befehlswort für "Einstellungsanfrage" ist `0x6056`, also `code = 24662`.
5. **JSON erstellen:** Fügen Sie die einzustellenden Parameter in `data` ein.
6. **Befehl senden:** Senden Sie den String über den aufgebauten TCP-Kanal; es wird empfohlen, am Ende `\r\n` hinzuzufügen.

3.2 Nachrichtenbeispiele

Szenario 1: Abschaltzeit bei niedrigem Akku einstellen

JSON senden:

JSON

```
1 {
2   "code": 24662,
3   "data": {
4     "t597": 1440
5   }
6 }
```

Bedeutung: Setzt die Ruhezeit bei niedrigem Akkustand auf 1440 Minuten (24 Stunden).

Szenario 2: Lokalen Modus deaktivieren

JSON senden:

JSON

```
1 {
2     "code": 24662,
3     "data": {
4         "t598": 0
5     }
6 }
```

Bedeutung: Setzt den Status des Lokalen Modus (`t598`) auf `0` (Aus).

4. Benutzerdefinierte Anzeigelogik der Plattform

In Ihrem lokalen Hub-Backend wird folgende Verarbeitung empfohlen:

- **Socket-Dienst:** Starten Sie einen TCP-Client-Prozess, um die Verbindung zum Gerät aufrechtzuerhalten.
- **Datenbereinigung:**
- Empfang von `code: 24658` -> Als **vollständige Echtzeit-Statusaktualisierung** behandeln -> In Datenbank schreiben / Frontend-Dashboard aktualisieren.
- Verwendung von `code: 24662` -> Als **Senden einer Einstellungsanfrage** behandeln.