

Matter 概要

Matter は Connectivity Standards Alliance (CSA) が策定・管理している、スマートホーム向けの共通通信規格です。

スマートホーム向けのデバイスは、ユーザーの操作を受け付けてデバイスを操作する機能や、1 つ以上のデバイスの機能を組み合わせることで自動的にデバイスを制御する機能が必要となります。こういった制御機能は、デバイス本体に実装される場合や、インターネット経由で接続したサーバーに実装される場合など様々な構成が考えられます。

執筆時点でスマートホームの制御機能を実装しているものとして最も普及しているものとしては、Amazon, Google, Apple などの各社が販売している **スマート・スピーカー**があります。これらのスマート・スピーカーを含む、スマートホーム向けデバイスの制御の中心となる **コントローラ・デバイス**と各種スマートホームに必要な機能を提供するデバイスの間の通信方法は、主にコントローラ・デバイスの販売者が定義している各種 Web API 経由での通信が必要な物がほとんどです。また API の仕様は各社が独自で決めたものであり、スマートホーム・デバイスをこれらのコントローラ・デバイスで制御するためには、それぞれの API 仕様に基づいた通信処理の実装が必要です。

こういった状況を解決するために、共通の通信規格として **Matter** が策定されました。

Matter の特徴

Matter の最も重要な特徴は、インターネット接続を経由しない、ローカルネットワーク内でのスマートホーム・デバイス間の通信規格であるという点です。従来はスマートホーム・デバイスとコントローラの間はインターネットを経由した API による接続がほとんどでしたが、Matter ではローカルネットワーク上でデバイス間で直接 IPv6 での通信を行います。

Matter デバイス間の物理的な通信手段としては、Wi-Fi、Ethernet、Thread、Bluetooth 5 の 4 つの通信規格を用います。

前述のとおり Matter では IPv6 による通信を用いて通信を行うため、IPv6 の通信を行える通信規格として Wi-Fi、Ethernet に加えて、バッテリー駆動の低消費電力デバイス向けに **Thread** を選択できます。Thread は Bluetooth5 と同じく IEEE 802.15.4 で規定される物理層・MAC 層を用いる無線通信規格ですが、Bluetooth と異なり上位の通信層として **6LoWPAN** と呼ばれる IPv6 を低消費電力無線向けに最適化したプロトコルを用います。このため Wi-Fi や Ethernet 上の IPv6 ネットワークとの相互接続が容易です。

Bluetooth 5 は動作中の主要な通信方法としては使用しませんが、コントローラとデバイス間を接続する処理に用います。

Matter で定義されるデバイス

Matter の仕様は CSA のサイトから誰でもダウンロードして読むことができます\*1。2024/05/10 時点の最新版は Matter 1.3 です。CSA のダウンロードページでダウンロードしたい仕様書を選択し、名前、会社名、メールアドレス、プライバシーポリシーへの同意を入力すると、指定したメールアドレス宛に仕様書のダウンロード元リンクが送られてきます。

代表的なデバイスの種別 (Device Type) を以下に示します。

名前	内容
On/Off Light	点灯・消灯できる照明
Dimmable Light	調光できる照明
Color Temperature Light	明るさや色を変えられる照明
On/Off Plug-in Unit	電源を ON/OFF できるプラグ (所謂スマートプラグ)
On/Off Light Switch	ON/OFF 入力して他の機器を制御するスイッチ
Contact Sensor	接触センサー。ドアの開閉検知など
Light Sensor	照度計
Occupancy Sensor	(部屋の) 使用中センサー
Temperature Sensor	温度センサー
Pressure Sensor	圧力センサー
Humidity Sensor	湿度センサー
Door Lock	ドアの鍵。所謂スマートロック
Window Covering	カーテンやブラインドの開閉
Thermostat	サーモスタット (空調)
Room Air Conditioner	ルームエアコン
Basic Video Player	ビデオプレイヤー
Refrigerator	冷蔵庫
Aggregator	複数デバイスのハブ

Matter 1.3 ではこれらに加えて、EVSE (Electric Vehicle Supply Equipment) つまり電気自動車の充電器や電力計のデバイスタイプといった、**エネルギー・マネジメント**のためのデバイスタイプが追加されています。

参考文献

- Matter Device Library Specification 1.3, Connectivity Standards Alliance, <https://csa-iot.org/developer-resource/specifications-download-request/>
  - Matter のデバイス・タイプの仕様

筆者について

- Kenta Ida
- twitter: [@ciniml](https://twitter.com/ciniml)
- GitHub: <https://github.com/ciniml/>
- 最近はだいたい ESP32 のファームウェア書いて生きてます。

Matter 概要

初版 2024 年 05 月 12 日

発行	Kenta IDA
著者	Kenta IDA (@ciniml)
連絡先	<a href="mailto:fuga@fugafuga.org">fuga@fugafuga.org</a>
Twitter	<a href="https://twitter.com/ciniml">@ciniml</a>
GitHub	<a href="https://github.com/ciniml/">https://github.com/ciniml/</a>

\*1 <https://csa-iot.org/developer-resource/specifications-download-request/>

## chip-tool 基本コマンド集

Matter の開発では Matter デバイスの動作確認のために connect-edhomeip に付属している **chip-tool** を使います。知っておくと便利なコマンドをいくつか示します。

変数名	内容
node-id	ノード ID. 普通は 1 でいい
ssid	Wi-Fi SSID
pass	Wi-Fi Password
dataset	Thread データセット
PIN	PIN コード (MPC ではない)
discriminator	ディスクリミネーター
MPC or QR	MPC または QR コードの文字列
vendor-id	ベンダー ID
product-id	プロダクト ID

### QR コード文字列を生成

```
1 chip-tool payload generate-qr-code --discriminator <discriminator> --setup-pin-code <PIN> --vendor-id <vendor-id> --product-id <product-id>
2 # chip-tool payload generate-qr-code --discriminator 3841 --setup-pin-code 20202020 --vendor-id 0 xfff1 --product-id 0x8000
3 # QR Code: MT:Y.K90619012Z.548G00
```

### QR コード文字列・MPC を解析

```
1 chip-tool payload parse-setup-payload <MPC or QR>
2 # chip-tool payload parse-setup-payload MT:Y.K90619012Z.548G00
3 # Parsing base38Representation: MT:Y.K90619012Z.548G00
4 # Version: 0
5 # VendorID: 65521
6 # ProductID: 32768
7 # Custom flow: 0 (STANDARD)
8 # Discovery Bitmask: 0x00 (NONE)
9 # Long discriminator: 3841 (0xf01)
10 # Passcode: 20202020
```

### PIN コード・ディスクリミネーターでコミッショニング

Wi-Fi 接続を設定する, Thread 接続を設定する, もしくは IP ネットワーク接続済みの場合

```
1 chip-tool pairing ble-wifi <node-id> <ssid> <pass> <PIN> <discriminator>
2 chip-tool pairing ble-thread <node-id> <dataset> <PIN> <discriminator>
3 chip-tool pairing onnetwork <node-id> <PIN>
```

### MPC や QR コードでコミッショニング

Wi-Fi 接続を設定する, Thread 接続を設定する, もしくは IP ネットワーク接続済みの場合

```
1 chip-tool pairing code-wifi <node-id> <ssid> <pass> <MPC or QR>
2 chip-tool pairing code-thread <node-id> <dataset> <MPC or QR>
3 chip-tool pairing code <node-id> <MPC or QR>
```

### エンドポイント一覧取得

```
1 chip-tool descriptor parts-list <node-id> 0
2 # エンドポイントが2つ (1と2) ある場合
```

```
3 # Endpoint: 0 Cluster: 0x0000_001D Attribute 0
   x0000_0003 DataVersion: 1997397239
4 # PartsList: 2 entries
5 # [1]: 1
6 # [2]: 2
```

### デバイスタイプ一覧

全エンドポイント (0xffff) のデバイスタイプ一覧を取得

```
1 chip-tool descriptor device-type-list <node-id> 0
   xffff
2 # endpoint0の物だけ記載
3 # Endpoint: 0 Cluster: 0x0000_001D Attribute 0
   x0000_0000 DataVersion: 1997397239
4 # DeviceTypeList: 1 entries
5 # [1]: {
6 # DeviceType: 22 # (0x16) Root Node
7 # Revision: 1
8 # }
```

### クラスター一覧

全エンドポイント (0xffff) のクラスター一覧を取得

```
1 chip-tool descriptor server-list <node-id> 0xffff
2 # endpoint2の物だけ記載
3 # Endpoint: 2 Cluster: 0x0000_001D Attribute 0
   x0000_0001 DataVersion: 1876874978
4 # ServerList: 3 entries
5 # [1]: 29 # (0x1d) Descriptor
6 # [2]: 3 # (0x03) Identify
7 # [3]: 69 # (0x45) Boolean State
```

### コミッショニング・ウィンドウを開く

既にコミッショニング済みの Matter デバイスを他のコントローラに繋ぐ場合に使います。Administrator Commissioning クラスターに関連コマンドがあります。

- <timeout-sec> - コミッショニング・ウィンドウを閉じるまでのタイムアウト秒数。
- <dummy> - ダミー
- <timeout-ms> - Open Commissioning Window コマンドのタイムアウトまでの時間 (ミリ秒)

```
1 chip-tool administratorcommissioning open-basic-commissioning-window <timeout-sec> <node-id> <dummy> --timedInteractionTimeoutMs <timeout-ms>
2 # chip-tool administratorcommissioning open-basic-commissioning-window 900 1 0 --timedInteractionTimeoutMs 10000
```

### インタラクティブモード

アトリビュートのリード・ライトやコマンドの実行は単発実行が可能です。サブスクリプションを行うにはセッションを維持する必要があるため、インタラクティブモードが必須です。

```
1 chip-tool interactive start
```

起動後は >>> のプロンプトが表示され、chip-tool の引数で指定できるコマンドが直接実行できます。

```
1 >>> booleanstate subscribe state-value 1 10 1 2
```