

PD Handler JSON format

1. PD Handler BLE (for Bluetooth Low Energy)

1-1. Beacon

●データサンプル

```
{
  "deviceId":"06091b268f10",
  "appendixInfo":"FAE00447",
  "time":"2016-03-14T17:05:42.965+09:00",
  "rssi":-67,
  "type":"iBeacon",
  "data":"1107ad7700c6a00099b2e2114c248d4a0c960cff8c0088636b7
a18be99de2c",
  "localname":"beacon",
  "status":"in"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	appendixInfo	付随情報	String		WEB UIから設定された値
3	time	データ取得日時	String	○	ISO8601拡張書式
4	rssi	受信信号強度	Integer	○	
5	type	ビーコン種別	String		WEB UIにて表示設定
6	data	ペイロードデータ	String		16進ダンプデータ
7	localname	ローカル名	String		WEB UIにて表示設定
8	status	ビーコンステータス	String		ビーコン制御タイプにより表示。("in"または"out")
ex	ユーザー設定	ユーザー設定内容	String		WEB UIから設定された値

1-2. Sensor

1-2-1. TI Sensor

●データサンプル

```

{
  "deviceId":"b0b448b93907",
  "time":"2016-03-14T09:32:15.864+09:00",
  "humidity":68.12,
  "temperature":25.51,
  "accelX":0,
  "accelY":0,
  "accelZ":-1.1001,
  "gyroX":0.3002,
  "gyroY":0.9001,
  "gyroZ":2.1003,
  "magX":-25.5004,
  "magY":48.0001,
  "magZ":-159.2002,
  "pressure":1008.22,
  "objectTemp":21,
  "ambientTemp":25.3,
  "lux":0.2
}

```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	humidity	湿度	Double	△	[%]
4	temperature	温度	Double	△	[°C]
5	accelX	X方向加速度	Double	△	[G]
6	accelY	Y方向加速度	Double	△	[G]
7	accelZ	Z方向加速度	Double	△	[G]
8	gyroX	X方向角速度	Double	△	[°/s]
9	gyroY	Y方向角速度	Double	△	[°/s]
10	gyroZ	Z方向角速度	Double	△	[°/s]
11	magX	X方向地磁気	Double	△	[μT]
12	magY	Y方向地磁気	Double	△	[μT]
13	magZ	Z方向地磁気	Double	△	[μT]
14	pressure	気圧	Double	△	[hPa]
15	objectTemp	物体温度	Double	△	[°C]
16	ambientTemp	周辺温度	Double	△	[°C]
17	lux	照度	Double	△	[lux]
18	memo	メモ	String		WEB UIから設定された値

※センサー依存のJSONキーデータは電池残量や使用モデルにより、含まれない場合があります。

1-2-2. Fujitsu Sensor

●データサンプル

```
{
  "deviceId":"b0b448b93908",
  "time":"2016-03-14T09:12:15.225+09:00",
  "temperature":25.61,
  "accelX":0,
  "accelY":0,
  "accelZ":-1.0001
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	temperature	温度	Double	○	[°C]
4	accelX	X方向加速度	Double	○	[G]
5	accelY	Y方向加速度	Double	○	[G]
6	accelZ	Z方向加速度	Double	○	[G]
7	memo	メモ	String		WEB UIから設定された値

1-2-3. ALPS IoT Smart Module ※ver.1.3.0対応

●データサンプル(接続モード：データパケット1)

```
{
  "deviceId":"34c731ffe620",
  "time":"2016-07-14T09:12:29.231+09:00",
  "dataIndex":123,
  "geoMagneticX":25.35,
  "geoMagneticY":-35.70,
  "geoMagneticZ":7.05,
  "accelX":0,
  "accelY":0,
  "accelZ":-1.0001,
  "ms":0,
  "second":28,
  "minute":12,
  "hour":9
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値

2	time	データ取得日時	String	○	ISO8601拡張書式
3	dataIndex	データインデックス	Integer	○	0~255(シーケンス番号)
4	geoMagneticX	X方向地磁気	Double		[uT]
5	geoMagneticY	Y方向地磁気	Double		[uT]
6	geoMagneticZ	Z方向地磁気	Double		[uT]
7	accelX	X方向加速度	Double		[G]
8	accelY	Y方向加速度	Double		[G]
9	accelZ	Z方向加速度	Double		[G]
10	ms	ミリ秒	Integer	○	
11	second	秒	Integer	○	
12	minute	分	Integer	○	
13	hour	時	Integer	○	
14	memo	メモ	String		WEB UIから設定された値

●データサンプル(接続モード：データパケット2)

```
{
  "deviceId":"34c731ffe620",
  "time":"2016-07-14T09:12:29.456+09:00",
  "dataIndex":123,
  "pressure":1010.42,
  "humidity":58.83,
  "temperature":29.41,
  "uv":0.0515,
  "ambientLight":50.5368,
  "day":14,
  "month":7,
  "year":16
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	dataIndex	データインデックス	Integer	○	0~255(シーケンス番号)
4	pressure	気圧	Double		[hPa]
5	humidity	湿度	Double		[%]
6	temperature	温度	Double		[°C]
7	uv	紫外線	Double		[mW/cm2]
8	ambientLight	照度	Double		[lux]
9	day	日	Integer	○	
10	month	月	Integer	○	
11	year	年	Integer	○	
12	memo	メモ	String		WEB UIから設定された値

●データサンプル(ビーコンモード：環境系フォーマット)

```
{
  "time":"2016-03-14T17:05:42.965+09:00",
  "memo":"ALPS beacon env",
  "deviceId":"34c731ffe620",
}
```

```

"accelX":0,
"accelY":0,
"accelZ":-1.0002,
"pressure":1010.42,
"humidity":58.83,
"temperature":29.41,
"uv":0.0515,
"ambientLight":50.5368
}

```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	accelX	X方向加速度	Double		[G]
4	accelY	Y方向加速度	Double		[G]
5	accelZ	Z方向加速度	Double		[G]
6	pressure	気圧	Double		[hPa]
7	humidity	湿度	Double		[%]
8	temperature	温度	Double		[°C]
9	uv	紫外線	Double		[mW/cm2]
10	ambientLight	照度	Double		[lux]
11	memo	メモ	String		WEB UIから設定された値

●データサンプル(ビーコンモード：モーション系フォーマット)

```

{
  "time":"2016-03-14T17:05:42.965+09:00",
  "deviceId":"34c731ffe620",
  "memo":"ALPS beacon motion",
  "accelX":0,
  "accelY":0,
  "accelZ":-1.0,
  "geoMagneticX":25.35,
  "geoMagneticY":-35.70,
  "geoMagneticZ":7.05,
  "pressure":1010.42
}

```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	accelX	X方向加速度	Double		[G]
4	accelY	Y方向加速度	Double		[G]
5	accelZ	Z方向加速度	Double		[G]
6	geoMagneticX	X方向地磁気	Double		[uT]
7	geoMagneticY	Y方向地磁気	Double		[uT]
8	geoMagneticZ	Z方向地磁気	Double		[uT]
9	pressure	気圧	Double		[hPa]

10	memo	メモ	String		WEB UIから設定された値
----	------	----	--------	--	----------------

1-2-4. オムロン 環境センサ

●データサンプル(接続モード)

```
{
  "deviceId":"d11397e0d126",
  "memo":"OMRON Env Sensor"
  "time":"2016-10-14T09:27:52.278+09:00",
  "humidity":38.7,
  "temperature":25.42,
  "light":114,
  "uvi":0.02,
  "pressure":1018.1,
  "noise":38.17,
  "discomfortIndex":71.09,
  "heatstroke":20.05,
  "battery":2917
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから"."を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	humidity	湿度	Double	○	[%]
4	temperature	温度	Double	○	[°C]
5	light	照度	Integer	○	[lux]
6	uvi	UVインデックス	Double	○	
7	pressure	気圧	Double	○	[hPa]
8	noise	騒音	Double	○	[dB]
9	discomfortIndex	不快指数	Double	○	
10	heatstroke	熱中症危険度	Double	○	[°C]
11	battery	電池電圧	Integer	○	[mV]
12	memo	メモ	String		WEB UIから設定された値

●データサンプル(ビーコンモード : IM)

```
{
  "time":"2016-10-14T18:23:27.739+09:00",
  "memo":"OMRON Env Sensor IM"
  "deviceId":"d11397e0d126",
  "sequence":36349,
  "temperature":24.39,
  "humidity":39.23,
  "light":93,
  "uvi":0.18,
  "pressure":1013.5,
  "noise":39.26,
  "accelX":-0.3,
}
```

```

"accelY":0.1,
"accelZ":1.2,
"battery":2930
}

```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	sequence	シーケンス番号	Integer	○	
4	temperature	温度	Double	○	[°C]
5	humidity	湿度	Double	○	[%]
6	light	照度	Integer	○	[lux]
7	uvi	UVインデックス	Double	○	
8	pressure	気圧	Double	○	[hPa]
9	noise	騒音	Double	○	[dB]
10	accelX	X方向加速度	Double		[G]
11	accelY	Y方向加速度	Double		[G]
12	accelZ	Z方向加速度	Double		[G]
13	battery	電池電圧	Integer	○	[mV]
14	memo	メモ	String		WEB UIから設定された値

●データサンプル(ビーコンモード : EP)

```

{
  "time":"2016-10-14T18:05:22.375+09:00",
  "memo":"OMRON Env Sensor EP"
  "deviceId":"d11397e0d126",
  "sequence":36381,
  "temperature":24.46,
  "humidity":39.73,
  "light":97,
  "uvi":0.03,
  "pressure":1013.2,
  "noise":39.42,
  "discomfortIndex":70.33,
  "heatstroke":19.77,
  "battery":2910
}

```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	sequence	シーケンス番号	Integer	○	
4	temperature	温度	Double	○	[°C]
5	humidity	湿度	Double	○	[%]
6	light	照度	Integer	○	[lux]
7	uvi	UVインデックス	Double	○	
8	pressure	気圧	Double	○	[hPa]

9	noise	騒音	Double	○	[dB]
10	discomfortIndex	不快指数	Double	○	
11	heatstroke	熱中症危険度	Double	○	[°C]
12	battery	電池電圧	Integer	○	[mV]
13	memo	メモ	String		WEB UIから設定された値

1-2-5. ユニ電子 BLE 温湿度センサー(Logtta)

●データサンプル(接続モード)

```
{
  "deviceId":"f0ab542bdca5",
  "memo":"Logtta TH Sensor",
  "time":"2016-10-13T02:29:38.406+09:00",
  "temperature":28.23,
  "humidity":35.69
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから"."を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	temperature	温度	Double	○	[°C]
4	humidity	湿度	Double	○	[%]
5	memo	メモ	String		WEB UIから設定された値

●データサンプル(ビーコンモード)

```
{
  "time":"2016-10-14T11:30:41.259+09:00",
  "deviceId":"f0ab542bdca5",
  "memo":"Logtta TH Sensor",
  "temperature":27.88,
  "humidity":36.48,
  "battery":100
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから"."を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	temperature	温度	Double	○	[°C]
4	humidity	湿度	Double	○	[%]
5	battery	バッテリーレベル	Integer	○	[%]
6	memo	メモ	String		WEB UIから設定された値

1-2-6. ユニ電子 BLECO2センサー(Logtta CO2)

●データサンプル(ビーコンモード)

```
{
  "time":"2017-03-03T12:34:56.789+09:00",
  "deviceId":"f0ab54c2gcdf",
  "memo":"Logtta CO2 Sensor",
  "co2":653,
  "battery":254
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	co2	CO2濃度	Integer	○	[ppm]
4	battery	バッテリーレベル	Integer	○	[%]
5	memo	メモ	String		WEB UIから設定された値

1-2-7. ラトックシステム Bluetooth ホコリセンサー

●データサンプル

```
{
  "deviceId":"dfb3f8c57912",
  "memo":"RATOC PM2.5",
  "time":"2017-01-13T11:02:06.283+09:00",
  "minute":2,
  "hour":11,
  "day":13,
  "month":1,
  "year":17,
  "pm25":15,
  "pm10":1,
  "pressure":999,
  "temperature":24,
  "humidity":18,
  "light":364,
  "mode":0
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	デバイスID	String	○	デバイスアドレスから":"を除き、小文字化した値
2	time	データ取得日時	String	○	ISO8601拡張書式
3	minute	計測日時(分)	Integer	○	

4	hour	計測日時(時)	Integer	○	
5	day	計測日時(日)	Integer	○	
6	month	計測日時(月)	Integer	○	
7	year	計測日時(年)	Integer	○	
8	pm25	PM2.5濃度	Integer	○	[$\mu\text{g}/\text{m}^3$]
9	pm10	PM10濃度	Integer	○	[$\mu\text{g}/\text{m}^3$]
10	pressure	気圧	Integer	○	[hPa]
11	temperature	温度	Integer	○	[$^{\circ}\text{C}$]
12	humidity	湿度	Integer	○	[%]
13	light	照度	Integer	○	[lx]
14	mode	計測データモード	Integer	○	0:連続計測、1:ワンショット計測
15	memo	メモ	String		WEB UIから設定された値

2. PD Handler UART (for UART extend device)

2-1. EnOcean

2-1-1. 人感センサー(EEP : A50701)

●データサンプル

```
{  
  "deviceId": "0400197A",  
  "time": "2016-03-14T16:17:02.269+09:00",  
  "svc": 4.764706,  
  "pirs": "on",  
  "EEP": "A50701",  
  "memo": "Occupancy Sensor"  
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	svc	供給電圧	Double		[V]
4	pirs	検知結果	String	○	“on”または“off”
5	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
6	memo	メモ	String		WEB UIにて設定した値

2-1-2. あけしめセンサー(EEP : D50001)

●データサンプル

```
{  
  "deviceId": "04000A1B",  
  "time": "2016-03-14T16:16:52.525+09:00",  
  "contact": 0,  
  "EEP": "D50001",  
  "memo": "Contacts and Switches"  
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	contact	開閉ステータス	Integer	○	0 : Open , 1 : Closed
4	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
5	memo	メモ	String		WEB UIにて設定した値

2-1-3.温度センサー(EEP : A50205)

●データサンプル

```
{
  "deviceId": "04000C66",
  "time": "2016-03-14T16:16:59.958+09:00",
  "temperature": 25.254902,
  "EEP": "A50205",
  "memo": "Temperature Sensors"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	temperature	温度	Double	○	[°C]
4	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
5	memo	メモ	String		WEB UIにて設定した値

2-1-4.温湿度センサー(EEP : A50402)

●データサンプル

```
{
  "deviceId": "0400267B",
  "time": "2017-08-31T14:26:39.283+09:00",
  "temperature": 28.160000,
  "humidity": 62.200006,
  "EEP": "A50402",
  "memo": "Temperature and Humidity Sensor"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	temperature	温度	Double	○	[°C]
4	humidity	湿度	Double	○	[%]
5	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
6	memo	メモ	String		WEB UIにて設定した値

2-1-5.温湿度センサー(EEP : A50403)

●データサンプル

```
{
  "deviceId": "040005C6",
  "time": "2016-03-14T16:15:58.904+09:00",
  "temperature": 25.122190,
  "humidity": 35.686275,
  "EEP": "A50403",
  "memo": "Temperature and Humidity Sensor"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	temperature	温度	Double	○	[°C]
4	humidity	湿度	Double	○	[%]
5	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
6	memo	メモ	String		WEB UIにて設定した値

2-1-6. 2相式CTセンサー(EEP : A51201)

●データサンプル

```
{
  "deviceId": "0400AE56",
  "time": "2016-03-14T16:15:58.904+09:00",
  "electricity": 15.0,
  "dataType": "W",
  "EEP": "A51201",
  "memo": "Automated Meter Reading (AMR)"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	electricity	電流/電圧/電力値	Double	○	現在値または累積値
4	dataType	データタイプ	String	○	“kWh”または“W”
5	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
6	memo	メモ	String		WEB UIにて設定した値

2-1-7. 3相式CTセンサー(EEP : D23202)

●データサンプル

```
{
  "deviceId": "04015100",
  "time": "2016-03-14T16:15:58.904+09:00",
  "channel1": 30.0,
  "channel2": 15.0,
  "channel3": 10.0,
  "EEP": "D23202",
  "memo": "A.C. Current Clamp"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	channel1	チャンネル1電流値	Double	○	[A] ※現在値
4	channel2	チャンネル2電流値	Double	○	[A] ※現在値
5	channel3	チャンネル3電流値	Double	○	[A] ※現在値
6	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
7	memo	メモ	String		WEB UIにて設定した値

2-1-8. 大気圧センサー(EEP : A50501) ※ver.1.0.1対応

●データサンプル

```
{
  "deviceId": "0401520B",
  "time": "2016-06-07T15:58:22.927+09:00",
  "barometer": 1010.850464,
  "telegram_type": "Hearbeat",
  "EEP": "A50501",
  "memo": "barometer"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	barometer	気圧	Double	○	[hPa]
4	telegram_type	電文タイプ	String	○	“Hearbeat”または“Event triggered”
5	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
6	memo	メモ	String		WEB UIにて設定した値

2-1-9. 照度センサー(EEP : A50602) ※ver.1.0.1対応

●データサンプル

```
{
  "deviceId": "04004715",
  "time": "2016-06-07T15:58:28.150+09:00",
  "svc": 2.780000,
  "ill1": 260.000000,
  "ill2": 260.000000,
  "EEP": "A50602",
  "memo": "ill"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	svc	供給電圧	Double	○	[V]
4	ill1	照度1	Double	○	[lux]
5	ill2	照度2	Double	○	[lux]
6	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
7	memo	メモ	String		WEB UIにて設定した値

2-1-10. CO2センサー(EEP : A50904) ※ver.1.0.1対応

●データサンプル

```
{
  "deviceId": "040004FF",
  "time": "2016-06-07T15:34:15.126+09:00",
  "humidity": 52.000000,
  "temperature": 28.000000,
  "concentration": 690,
  "EEP": "A50904",
  "memo": "CO2"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	humidity	湿度	Double	○	[%]
4	temperature	温度	Double	○	[°C]
5	concentration	CO2濃度	Integer	○	[ppm]
6	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値

7	memo	メモ	String		WEB UIにて設定した値
---	------	----	--------	--	---------------

2-1-11. デジタル入力センサー(EEP : A53005) ※ver.1.0.1対応

●データサンプル

```
{
  "deviceId": "04002D68",
  "time": "2016-06-07T15:44:09.621+09:00",
  "vdd": 3.157647,
  "signal_type": "Heart beat signal",
  "count": 127,
  "EEP": "A53005",
  "memo": "button"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	vdd	供給電圧	Double	○	[V]
4	signal_type	信号種別	String	○	“Normal signal”または “Heart beat signal”
5	count	序数	Integer	○	0~127
6	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
7	memo	メモ	String		WEB UIにて設定した値

2-1-12. 2ロックースイッチ(EEP : F60204) ※ver.1.1.3対応

●データサンプル

```
{
  "deviceId": "002BC9C8",
  "time": "2016-07-26T10:45:09.625+09:00",
  "ebo": "pressed",
  "rbi": "released",
  "rbo": "released",
  "rai": "pressed",
  "rao": "released",
  "EEP": "F60204",
  "memo": "2 rocker switch"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式

3	ebo	エネルギー状態	String	○	“pressed”または “released”
4	rbi	ロッカースイッチBの状態I	String	○	“pressed”または “released”
5	rbo	ロッカースイッチBの状態O	String	○	“pressed”または “released”
6	rai	ロッカースイッチAの状態I	String	○	“pressed”または “released”
7	rao	ロッカースイッチAの状態O	String	○	“pressed”または “released”
8	EEP	EnOceanプロファイル	String	○	WEB UIにて設定した値
9	memo	メモ	String		WEB UIにて設定した値

2-1-13. RAWデータ時

●データサンプル

```
{
  "deviceId": "0400197A",
  "time": "2016-03-14T16:45:32.643+09:00",
  "data": "55000c020ae66200000400197a1c080b8720013da6",
  "EEP": "A50701",
  "memo": "raw data"
}
```

#	JSONキー	内容	データ型	常駐	補足
1	deviceId	ID	String	○	
2	time	データ取得日時	String	○	ISO8601拡張書式
3	data	ペイロード	String	○	
4	EEP	EnOceanプロファイル	String		WEB UIにて設定した値
5	memo	メモ	String		WEB UIにて設定した値

2-2. Wi-SUN

2-2-1. 瞬時電力(Bルート)

●データサンプル

```
{
  "address": "0011223344556677",
  "date": "2016-1-25",
  "time": "12:34",
  "inst": 2147483645
}
```

#	JSONキー	内容	データ型	常駐	補足
1	address	アドレス	String	○	スマートメーター側のアドレス
2	date	対象データ年月日	String	○	
3	time	対象データ時間	String	○	
4	inst	瞬時電力	Integer	○	[W]

2-2-2. 積算電力(Bルート)

●データサンプル

```
{
  "address": "0011223344556677",
  "time": "2016-1-25T12:34:56",
  "ratio": 10,
  "unit": 0.01,
  "cumu": 2147483645,
  "cumu_re": -1
}
```

#	JSONキー	内容	データ型	常駐	補足
1	address	アドレス	String	○	スマートメーター側のアドレス
2	time	対象データ日時	String	○	
3	ratio	倍率	Integer	○	0 ~ 999,999
4	unit	単位	Double	○	以下の値。 1, 0.1, 0.01, 0.001, 0.0001 10, 100, 1000, 10000
5	cumu	正方向累積値	Integer	○	0 ~ 99,999,999。但し、-1は データ無し扱い。
6	cumu_re	逆方向累積値	Integer	○	0 ~ 99,999,999。但し、-1は データ無し扱い。

※計算方式

- 正方向 積算電力

$$\langle Total \rangle = \langle cumu \rangle \times \langle ratio \rangle \times \langle unit \rangle$$

- 逆方向 積算電力

$$\langle Total \rangle = \langle cumu_re \rangle \times \langle ratio \rangle \times \langle unit \rangle$$

2-3. 特定少電力モジュール

2-3-1. FCL

●データサンプル

```
{
  "timestamp": "2016-02-01T21:07:06+09:00",
  "payload":
  "YXNkYWpzYWtsZGprbGFzZGprbGFzZG5za2FsZGpzYWtsZA=="
}
```

#	JSONキー	内容	データ型	常駐	補足
1	timestamp	データ取得日時	String	○	ISO8601拡張形式
2	payload	ペイロード	String	○	ペイロードのBase64エンコード結果

2-4. PLC

2-4-1. PLCクライアント(PLCマスター)

●データサンプル(Modbus/TCP レジスタ又はレジスタ入力読み込み)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"modbus",
  "node":"172.16.7.250",
  "port":1502,
  "unit":255,
  "memo":"PLC01",
  "address":31,
  "function":3,
  "data_type":"u_int16",
  "value":[2,0,1234,5678,9876]
}
```

#	JSONキー	内容	データ型	常駐	補足
1	time	データ取得日時	String	○	ISO8601拡張書式

2	protocol	プロトコル	String	○	“modbus_tcp” (固定)
3	node	取得先IPアドレス	String	○	WEB UIから設定された値
4	port	ポート番号	Integer	○	WEB UIから設定された値
5	unit	Modbus Unit ID	Integer	○	WEB UIから設定された値
6	memo	メモ	String	○	WEB UIから設定された値
7	address	読み込みアドレス	Integer	○	WEB UIから設定された値
8	function	Modbus function code	integer	○	3: read holding registers 4: read input registers
9	data_type	データの型	String	○	“u_int16” : 符号なし16bits “Int16”: 符号付16bits “u_int32lsb”: 符号なし32bits LSB “u_int32msb”: 符号なし32bits MSB “int32lsb”: 符号付32bits LSB “int32msb”: 符号付32bits MSB
10	value	読み込み値	Integer 配列	○	配列数は読み込みレジスタ数の設定に応じて可変。

●データサンプル(Modbus/TCP コイル又はステータス読み込み)

```
{
  "time": "2017-02-03T14:44:37.020+09:00",
  "protocol": "modbus_tcp",
  "node": "172.16.7.250",
  "port": 1502,
  "unit": 255,
  "memo": "PLC02",
  "address": 40,
  "function": 1,
  "value": [1,0,0,1,1,0,1]
}
```

#	JSONキー	内容	データ型	常駐	補足
1	time	データ取得日時	String	○	ISO8601拡張書式
2	protocol	プロトコル	String	○	“modbus_tcp” (固定)
3	node	取得先IPアドレス	String	○	WEB UIから設定された値
4	port	ポート番号	Integer	○	WEB UIから設定された値
5	unit	Modbus Unit ID	Integer	○	WEB UIから設定された値
6	memo	メモ	String	○	WEB UIから設定された値
7	address	読み込みアドレス	Integer	○	WEB UIから設定された値
8	function	Modbus function code	Integer	○	1: read coils 2: read discrete inputs
9	value	読み込み値	Integer 配列	○	0又は1。

					配列数は読み込みレジスタ数の設定に応じて可変。
--	--	--	--	--	-------------------------

●データサンプル(Modbus/RTU レジスタ又はレジスタ入力読み込み読み込み)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"modbus_rtu",
  "device":"/dev/ttyMFD2",
  "unit":35,
  "memo":"PLC03",
  "address":12,
  "function":4,
  "data_type":"u_int32lsb",
  "value":[2,0,8910]
}
```

#	JSONキー	内容	データ型	常駐	補足
1	time	データ取得日時	String	○	ISO8601拡張書式
2	protocol	プロトコル	String	○	“modbus_rtu” (固定)
3	device	デバイスファイル名	String	○	WEB UIから設定された値
4	unit	Modbus Unit ID	Integer	○	WEB UIから設定された値
5	memo	メモ	String	○	WEB UIから設定された値
6	address	読み込みアドレス	Integer	○	WEB UIから設定された値
7	function	Modbus function code	Integer	○	3: read holding registers 4: read input registers
8	data_type	データの型	String	○	“u_int16” : 符号なし16bits “Int16”: 符号付16bits “u_int32lsb”: 符号なし32bits LSB “u_int32msb”: 符号なし32bits MSB “int32lsb”: 符号付32bits LSB “int32msb”: 符号付32bits MSB
9	value	読み込み値	Integer 配列	○	配列数は読み込みレジスタ数の設定に応じて可変。

●データサンプル(Modbus/RTUコイル又はステータス読み込み)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"modbus_rtu",
  "device":"/dev/ttyMFD2",
  "unit":21,
```

```

"memo":"PLC04",
"address":37,
"function":2,
"value":[1,0,0,0,0,0,1,0,1,0,1,1,0,1,0,1,0,0,1,1,1]
}

```

#	JSONキー	内容	データ型	常駐	補足
1	time	データ取得日時	String	○	ISO8601拡張書式
2	protocol	プロトコル	String	○	“modbus_rtu”（固定）
3	device	デバイスファイル名	String	○	WEB UIから設定された値
4	unit	Modbus Unit ID	Integer	○	WEB UIから設定された値
5	memo	メモ	String	○	WEB UIから設定された値
6	address	読み込みアドレス	Integer	○	WEB UIから設定された値
7	function	Modbus function code	Integer	○	1: read coils 2: read discrete inputs
8	value	読み込み値	Integer 配列	○	0又は1。 配列数は読み込みレジスタ数の設定に応じて可変。

2-4-1. PLCサーバー(PLCスレーブ)

●データサンプル(Modbus/TCP レジスタ書き込み)

```

{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"modbus_tcp",
  "node":"172.16.7.240",
  "port":502,
  "unit":255,
  "memo":"PLC05",
  "address":31,
  "function":6,
  "value":[5678]
}

```

#	JSONキー	内容	データ型	常駐	補足
1	time	データ取得日時	String	○	ISO8601拡張書式
2	protocol	プロトコル	String	○	“modbus_tcp”（固定）
3	node	書き込み元IPアドレス	String	○	
4	port	ポート番号	Integer	○	502（固定）
5	unit	Modbus Unit ID	Integer	○	255（固定）
6	memo	メモ	String	○	WEB UIから設定された値
7	address	書き込みアドレス	Integer	○	0～(2048 - registers)の範囲

8	function	Modbus function code	Integer	○	6: write single register 16:write multiple registres
9	value	書き込まれた値	Integer 配列	○	16ビット符号無し整数 値。 配列数は書き込まれたレ ジスタ数に応じて可変。

●データサンプル(Modbus/TCP コイル書き込み)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"modbus_tcp",
  "node":"172.16.7.240",
  "port":502,
  "unit":255,
  "memo":"PLC06",
  "address":40,
  "function":"write_multiple_coils",
  "value":[1,0,0,1,1,0,1]
}
```

#	JSONキー	内容	データ型	常駐	補足
1	time	データ取得日時	String	○	ISO8601拡張書式
2	protocol	プロトコル	String	○	“modbus_tcp” (固定)
3	node	書き込み元IPアド レス	String	○	
4	port	ポート番号	Integer	○	502 (固定)
5	unit	Modbus Unit ID	Integer	○	255 (固定)
6	memo	メモ	String	○	WEB UIから設定された値
7	address	書き込みアドレス	Integer	○	0 ~ (2048 - bitss)の範囲
8	function	Modbus function code	Integer	○	5: write single 15: write multiple coils
9	value	書き込まれた値	Integer 配列	○	0又は1。 配列数は書き込まれた ビット数に応じて可変。

●データサンプル(Modbus/RTUレジスタ書き込み)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"modbus_rtu",
  "device":"/dev/ttyMFD2",
  "unit":35,
  "memo":"PLC07",
  "address":12,
  "function":16,
  "value":[2,0,78910]}
}
```

#	JSONキー	内容	データ型	常駐	補足
---	--------	----	------	----	----

1	time	データ取得日時	String	○	ISO8601拡張書式
2	protocol	プロトコル	String	○	“modbus_rtu”（固定）
3	device	デバイスファイル名	String	○	WEB UIから設定された値
4	unit	Modbus Unit ID	Integer	○	WEB UIから設定された値
5	memo	メモ	String	○	WEB UIから設定された値
6	address	書き込みアドレス	Integer	○	0 ~ (2048 - registers)の範囲
7	function	Modbus function code	Integer	○	6: write single register 16: write multiple registers
8	value	書き込まれた値	Integer 配列	○	16ビット符号無し整数値。 配列数は書き込まれたレジスタ数に応じて可変。

●データサンプル(Modbus/RTU コイル書き込み)

```
{
  "time": "2017-02-03T14:44:37.020+09:00",
  "protocol": "modbus_rtu",
  "device": "/dev/ttyMFD2",
  "unit": 21,
  "memo": "PLC08",
  "address": 37,
  "function": 5,
  "value": [1]
}
```

#	JSONキー	内容	データ型	常駐	補足
1	time	データ取得日時	String	○	ISO8601拡張書式
2	protocol	プロトコル	String	○	“modbus_rtu”（固定）
3	device	デバイスファイル名	String	○	WEB UIから設定された値
4	unit	Modbus Unit ID	Integer	○	WEB UIから設定された値
5	memo	メモ	String	○	WEB UIから設定された値
6	address	書き込みアドレス	Integer	○	0 ~ (2048 - bits)の範囲
7	function	Modbus function code	Integer	○	5: write single 15: write multiple coils
8	value	書き込まれた値	Integer 配列	○	0又は1。 配列数は書き込まれたビット数に応じて可変。