

OpenBlocks IoT Family PD Handler JSON Format List



Ver.3.3.0

Plat'Home Co., Ltd.

■ About trademarks

- Company names, product names and other names in this document may be trademarks or registered trademarks of each company.
- Other proper names such as product names in this document are trademarks or registered trademarks of each company.

■ Before use

- It is prohibited to reprint the contents of this document in whole or in part without prior consent.
- The contents of this document are subject to change without notice.
- Although we make our best efforts to ensure the accuracy of this document, please contact our support center if you find typographical errors or other mistakes.
The latest version of this document can be downloaded from our website.
- Note in advance that this device is not assumed to be used in in fields with fatal dangers.
- Note in advance that we bear no responsibility for any damages and lost earnings resulting from operation of this device.

■ Notes

- This document does not contain information on devices destined for Japan that are not exported abroad.

Table of Contents

1.PD Handler BLE (Node.js)	6
1.1.Beacon.....	6
1.1.1.When setting to use as a beacon	6
1.1.2.When setting to use as BLE	6
1.2.Sensor.....	7
1.2.1TI Sensor	7
1.2.1.Fujitsu Sensor	8
1.2.1.1.When setting to use as a beacon	8
1.2.1.2.When setting to use as BLE	9
1.2.3.ALPS IoT Smart Module	10
1.2.3.1.When setting to use as a beacon	10
1.2.3.2.When setting to use as BLE	12
1.2.4.OMRON Environmental Sensor.....	16
1.2.4.1.When setting to use as a beacon	16
1.2.4.2.When setting to use as BLE	18
1.2.5.OMRON USB Type Environmental Sensor	21
1.2.5.1.When setting to use as a beacon	21
1.2.5.2.When setting to use as BLE	22
1.2.6.UNI-ELECTRONICS Wireless thermohygrometer	23
1.2.6.1.When setting to use as a beacon is done	23
1.2.6.2.When setting to use as BLE is done	23
1.2.7.UNI-ELECTRONICS Wireless Carbon Dioxide Sensor	24
1.2.7.1.When setting to use as a beacon is done	24
1.2.7.2.When setting to use as BLE is done	24
1.2.8.UNI-ELECTRONICS Wireless Water Thermometer	25
1.2.8.1.When setting to use as a beacon is done	25
1.2.8.2.When setting to use as BLE is done	25
1.2.9.RATOC systems Bluetooth Dust Sensor	26
1.2.10.RATOC systems Bluetooth Air Quality Monitor	27
1.2.11.RATOC systems Bluetooth Watt Checker.....	28
1.2.12.Elecs Industry μ PRISM.....	29
2.PD Handler BLE with Lua on C language.....	30
2.1.Beacon.....	30
2.2.Sensor.....	30

2.2.1.NAKAYO Push Botton.....	30
2.2.1.1.When setting to use as a beacon	30
2.2.1.2.When setting to use as BLE	31
3.PD Handler UART	32
3.1.EnOcean with Lua	32
3.1.1.Human Sensor (EEP : A50701).....	32
3.1.2.Door Sensor (EEP : D50001)	33
3.1.3. Temperature Sensor (EEP : A50205)	33
3.1.4.Temperature/Humidity Sensor(EEP : A50402)	34
3.1.5. Temperature/Humidity Sensor(EEP : A50403).....	34
3.1.6.Single Phase Electricity sensor (EEP : A51201)	35
3.1.7.Three Channel Current Sensor (EEP : D23202)	36
3.1.8.Atmospheric sensor (EEP : A50501).....	37
3.1.9.Illuminance Sensor(EEP : A50602)	37
3.1.10.Carbon Dioxide Concentration Sensor(EEP: A50904).....	38
3.1.11. Digital Input Sensor(EEP : A53005).....	39
3.1.12.2 Input Locker Switch(EEP : F60204)	40
3.1.13. In RAW data mode	41
3.2.Wi-SUN	42
3.2.1. Instantaneous Power (B-Route)	42
3.2.2. Accumulated Power (B-Route)	42
4.PD Handler HVSMC.....	43
4.1.RAW Data	43
4.2. Operating Status	44
4.3. Abnormal Status	44
4.4.ECHONET Lite Property Infomation.....	45
4.5.High Voltage Smart Power Meter Property Information	46
4.6. Regular Time Measuremen Value	47
4.7. Measuremen Value	48
4.8. Demand power	48
5.PD Handler Modbus	49
5.1.Modbus Client(Modbus master)	49
5.1.1.Polling operation to PLC device	49
5.1.2.On demand operation from the cloud.....	51
5.2.Modbus Server(Modbus slave).....	61
5.2.1.Write operation from PLC device.....	61

5.2.2.On demand operation from the cloud.....	63
---	----

1. PD Handler BLE (Node.js)

1.1. Beacon

1.1.1. When setting to use as a beacon

■ Data Sample

```
{
  "time": "2017-12-08T15:00:04.549+09:00",
  "deviceId": "e9c8dd35ee18",
  "appendixInfo": "G8H00012",
  "rssi": -88,
  "type": "iBeacon",
  "data": "0201040c0946434c2042656163666e31",
  "localname": "beacon",
  "status": "in"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
ex	User define	User defined	String		Value set in Web UI.

1.1.2. When setting to use as BLE

■ Data Sample

```
{
  "time": "2017-12-08T15:00:04.549+09:00",
  "deviceId": "e9c8dd35ee18",
  "memo": "BLE beacon"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
3	memo	Notes	String		Value set in Web UI.

1.2.Sensor

1.2.1TI Sensor

■Data sample

```
{
  "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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	○	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	○	ISO8601 extended format
3	humidity	Humidity	Double	△	[%]
4	temperature	Temperature	Double	△	[°C]
5	accelX	X axis acceleration	Double	△	[G]
6	accelY	Y axis acceleration	Double	△	[G]
7	accelZ	Z axis acceleration	Double	△	[G]
8	gyroX	X axis angular velocity	Double	△	[°/s]
9	gyroY	Y axis angular velocity	Double	△	[°/s]
10	gyroZ	Z axis angular velocity	Double	△	[°/s]
11	magX	X axis geomagnetic torque	Double	△	[μT]
12	magY	Y axis geomagnetic torque	Double	△	[μT]
13	magZ	Z axis geomagnetic torque	Double	△	[μT]
14	pressure	Atmospheric pressure	Double	△	[hPa]
15	objectTemp	Object temperature	Double	△	[°C]
16	ambientTemp	Ambient temperature	Double	△	[°C]
17	lux	Illuminance	Double	△	[lux]
18	memo	Notes	String		Value set in Web UI.

*Depending on the remaining battery level and sensor model, there is data not included

1.2.1.Fujitsu Sensor

1.2.1.1.When setting to use as a beacon

■Data sample

```
{
  "deviceId":"b0b448b93908",
  "time":"2016-03-14T09:12:15.225+09:00",
  "rssi":-67,
  "temperature":25.61,
  "accelX":0,
  "accelY":0,
  "accelZ":-1.0001
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	○	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	○	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	○	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
9	temperature	Temperature	Double	△	[°C]
10	accelX	X axis acceleration	Double	△	[G]
11	accelY	Y axis acceleration	Double	△	[G]
12	accelZ	Z axis acceleration	Double	△	[G]
ex	User define	User defined	String		Value set in Web UI.

1.2.1.2. When setting to use as BLE

■ Data sample

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

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	temperature	Temperature	Double	<input type="radio"/>	[°C]
4	accelX	X axis acceleration	Double	<input type="radio"/>	[G]
5	accelY	Y axis acceleration	Double	<input type="radio"/>	[G]
6	accelZ	Z axis acceleration	Double	<input type="radio"/>	[G]
7	memo	Notes	String		Value set in Web UI.

1.2.3.ALPS IoT Smart Module

1.2.3.1.When setting to use as a beacon

■Data sample(Beacon mode: Environmental format)

```
{
  "time":"2016-03-14T17:05:42.965+09:00",
  "deviceId":"34c731ffe620",
  "rssi":-87,
  "accelX":0,
  "accelY":0,
  "accelZ":-1.0002,
  "pressure":1010.42,
  "humidity":58.83,
  "temperature":29.41,
  "uv":0.0515,
  "ambientLight":50.5368
}
```

#	JSON Key	Description	Data Type	Alwa ys	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding “.” from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. (“In”or “out”)
9	accelX	X axis acceleration	Double		[G]
10	accelY	Y axis acceleration	Double		[G]
11	accelZ	Z axis acceleration	Double		[G]
12	pressure	Atmospheric pressure	Double		[hPa]
13	humidity	Humidity	Double		[%]
14	temperature	Temperature	Double		[°C]
15	uv	Ultraviolet	Double		[mW/cm2]
16	ambientLight	Illuminance	Double		[lux]
ex	User define	User defined	String		Value set in Web UI.

■Data sample(Beacon mode: Motion format)

```
{
  "time":"2016-03-14T17:05:42.965+09:00",
  "deviceId":"34c731ffe620",
  "rssi":-87,
  "accelX":0,
  "accelY":0,
  "accelZ":-1.0,
  "geoMagneticX":25.35,
  "geoMagneticY":-35.70,
  "geoMagneticZ":7.05,
  "pressure":1010.42
}
```

#	JSON Key	Description	Data Type	Alwa ys	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
9	accelX	X axis acceleration	Double		[G]
10	accelY	Y axis acceleration	Double		[G]
11	accelZ	Z axis acceleration	Double		[G]
12	geoMagneticX	X axis geomagnetic torque	Double		[uT]
13	geoMagneticY	Y axis geomagnetic torque	Double		[uT]
14	geoMagneticZ	Z axis geomagnetic torque	Double		[uT]
15	pressure	Atmospheric pressure	Double		[hPa]
ex	User define	User defined	String		Value set in Web UI.

1.2.3.2. When setting to use as BLE

■ Data sample (Connection mode: Data packet 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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	dataIndex	Data index	Integer	<input type="radio"/>	0~255 (Sequence number)
4	geoMagneticX	X axis geomagnetic torque	Double	<input type="radio"/>	[uT]
5	geoMagneticY	Y axis geomagnetic torque	Double	<input type="radio"/>	[uT]
6	geoMagneticZ	Z axis geomagnetic torque	Double	<input type="radio"/>	[uT]
7	accelX	X axis acceleration	Double	<input type="radio"/>	[G]
8	accelY	Y axis acceleration	Double	<input type="radio"/>	[G]
9	accelZ	Z axis acceleration	Double	<input type="radio"/>	[G]
10	ms	Millisecond	Integer	<input type="radio"/>	
11	second	Second	Integer	<input type="radio"/>	
12	minute	Minute	Integer	<input type="radio"/>	
13	hour	Hour	Integer	<input type="radio"/>	
14	memo	Notes	String	<input type="radio"/>	Value set in Web UI.

■Data sample(Connection mode: Data packet 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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	dataIndex	Data index	Integer	<input type="radio"/>	0~255 (Sequence number)
4	pressure	Atmospheric pressure	Double		[hPa]
5	humidity	Humidity	Double		[%]
6	temperature	Temperature	Double		[°C]
7	uv	Ultraviolet	Double		[mW/cm2]
8	ambientLight	Illuminance	Double		[lux]
9	day	Day	Integer	<input type="radio"/>	
10	month	Month	Integer	<input type="radio"/>	
11	year	Year	Integer	<input type="radio"/>	
12	memo	Notes	String		Value set in Web UI.

■Data sample (beacon mode: environmental format)

```
{
  "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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	accelX	X axis acceleration	Double		[G]
4	accelY	Y axis acceleration	Double		[G]
5	accelZ	Z axis acceleration	Double		[G]
6	pressure	Atmospheric pressure	Double		[hPa]
7	humidity	Humidity	Double		[%]
8	temperature	Temperature	Double		[°C]
9	uv	Ultraviolet	Double		[mW/cm2]
10	ambientLight	Illuminance	Double		[lux]
11	memo	Notes	String		Value set in Web UI.

■Data sample (beacon mode: motion format)

```
{
  "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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	accelX	X axis acceleration	Double		[G]
4	accelY	Y axis acceleration	Double		[G]
5	accelZ	Z axis acceleration	Double		[G]
6	geoMagneticX	X axis geomagnetic torque	Double		[uT]
7	geoMagneticY	Y axis geomagnetic torque	Double		[uT]
8	geoMagneticZ	Z axis geomagnetic torque	Double		[uT]
9	pressure	Atmospheric pressure	Double		[hPa]
10	memo	Notes	String		Value set in Web UI.

1.2.4.OMRON Environmental Sensor

1.2.4.1.When setting to use as a beacon

■Data sample (beacon mode: IM)

```
{
  "time":"2016-10-14T18:23:27.739+09:00",
  "deviceId":"d11397e0d126",
  "rssi":-61,
  "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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
9	sequence	Sequence number	Integer	<input type="radio"/>	
10	temperature	Temperature	Double	<input type="radio"/>	[°C]
11	humidity	Humidity	Double	<input type="radio"/>	[%]
12	light	Illuminance	Integer	<input type="radio"/>	[lux]
13	uvi	UV index	Double	<input type="radio"/>	
14	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
15	noise	Noise	Double	<input type="radio"/>	[dB]
16	accelX	X axis acceleration	Double		[G]
17	accelY	Y axis acceleration	Double		[G]
18	accelZ	Z axis acceleration	Double		[G]
19	battery	Battery voltage	Integer	<input type="radio"/>	[mV]
ex	User define	User defined	String		Value set in Web UI.

■Data sample (beacon mode: EP)

```
{
  "time":"2016-10-14T18:05:22.375+09:00",
  "deviceId":"d11397e0d126",
  "rssi":-61,
  "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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
9	sequence	Sequence number	Integer	<input type="radio"/>	
10	temperature	Temperature	Double	<input type="radio"/>	[°C]
11	humidity	Humidity	Double	<input type="radio"/>	[%]
12	light	Illuminance	Integer	<input type="radio"/>	[lux]
13	uvi	UV index	Double	<input type="radio"/>	
14	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
15	noise	Noise	Double	<input type="radio"/>	[dB]
16	discomfortIndex	Discomfort index	Double	<input type="radio"/>	
17	heatstroke	Heat stroke risk	Double	<input type="radio"/>	[°C]
18	battery	Battery voltage	Integer	<input type="radio"/>	[mV]
ex	User define	User defined	String		Value set in Web UI.

1.2.4.2. When setting to use as BLE

■Data sample (connection mode)

```
{
  "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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	humidity	Humidity	Double	<input type="radio"/>	[%]
4	temperature	Temperature	Double	<input type="radio"/>	[°C]
5	light	Illuminance	Integer	<input type="radio"/>	[lux]
6	uvi	UV index	Double	<input type="radio"/>	
7	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
8	noise	Noise	Double	<input type="radio"/>	[dB]
9	discomfortIndex	Discomfort index	Double	<input type="radio"/>	
10	heatstroke	Heat stroke risk	Double	<input type="radio"/>	[°C]
11	battery	Battery voltage	Integer	<input type="radio"/>	[mV]
12	memo	Notes	String		Value set in Web UI.

■Data sample (beacon mode: 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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	sequence	Sequence number	Integer	<input type="radio"/>	
4	temperature	Temperature	Double	<input type="radio"/>	[°C]
5	humidity	Humidity	Double	<input type="radio"/>	[%]
6	light	Illuminance	Integer	<input type="radio"/>	[lux]
7	uvi	UV index	Double	<input type="radio"/>	
8	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
9	noise	Noise	Double	<input type="radio"/>	[dB]
10	accelX	X axis acceleration	Double		[G]
11	accelY	Y axis acceleration	Double		[G]
12	accelZ	Z axis acceleration	Double		[G]
13	battery	Battery voltage	Integer	<input type="radio"/>	[mV]
14	memo	Notes	String		Value set in Web UI.

■Data sample (beacon mode: 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 Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	sequence	Sequence number	Integer	<input type="radio"/>	
4	temperature	Temperature	Double	<input type="radio"/>	[°C]
5	humidity	Humidity	Double	<input type="radio"/>	[%]
6	light	Illuminance	Integer	<input type="radio"/>	[lux]
7	uvi	UV index	Double	<input type="radio"/>	
8	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
9	noise	Noise	Double	<input type="radio"/>	[dB]
10	discomfortIndex	Discomfort index	Double	<input type="radio"/>	
11	heatstroke	Heat stroke risk	Double	<input type="radio"/>	[°C]
12	battery	Battery voltage	Integer	<input type="radio"/>	[mV]
13	memo	Notes	String		Value set in Web UI.

1.2.5.OMRON USB Type Environmental Sensor

1.2.5.1.When setting to use as a beacon

■Data sample (beacon mode: Sensor data)

```
{
  "time":"2018-11-12T15:29:15.276+09:00",
  "deviceId":"f5961b7e0775",
  "memo":" OMRON USB ",
  "sequence":24,
  "temperature":27.11,
  "humidity":40.75,
  "light":171,
  "pressure":1008.283,
  "noise":56.5,
  "etvoc":26,
  "eco2":573
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rsi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
9	sequence	Sequence number	Integer	<input type="radio"/>	
10	temperature	Temperature	Double	<input type="radio"/>	[°C]
11	humidity	Humidity	Double	<input type="radio"/>	[%]
12	light	Illuminance	Integer	<input type="radio"/>	[lux]
13	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
14	noise	Noise	Double	<input type="radio"/>	[dB]
15	etvoc	Equivalent total volatile organic compounds	Integer		[ppb]
16	eco2	Equivalent Carbon dioxide concentration	Integer		[ppm]
ex	User define	User defined	String		Value set in Web UI.

1.2.5.2. When setting to use as BLE

■Data sample (beacon mode: Sensor data)

```
{
  "time": "2018-11-12T15:35:53.893+09:00",
  "memo": "OMRON USB",
  "deviceId": "f5961b7e0775",
  "sequence": 82,
  "temperature": 27.53,
  "humidity": 41.27,
  "light": 173,
  "pressure": 1003.256,
  "noise": 49.26,
  "etvoc": 268,
  "eco2": 583
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	sequence	Sequence number	Integer	<input type="radio"/>	
4	temperature	Temperature	Double	<input type="radio"/>	[°C]
5	humidity	Humidity	Double	<input type="radio"/>	[%]
6	light	Illuminance	Integer	<input type="radio"/>	[lux]
7	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
8	noise	Noise	Double	<input type="radio"/>	[dB]
9	etvoc	Equivalent total volatile organic compounds	Integer		[ppb]
10	eco2	Equivalent Carbon dioxide concentration	Integer		[ppm]
11	User define	User defined	String		Value set in Web UI.

1.2.6.UNI-ELECTRONICS Wireless thermohygrometer

1.2.6.1.When setting to use as a beacon is done

■Data sample (beacon mode)

```
{
  "time":"2016-10-14T11:30:41.259+09:00",
  "deviceId":"f0ab542bdca5",
  "rssi":-90,
  "temperature":27.88,
  "humidity":36.48,
  "battery":100
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
9	temperature	Temperature	Double	<input type="radio"/>	[°C]
10	humidity	Humidity	Double	<input type="radio"/>	[%]
11	battery	Battery usage rate	Integer	<input type="radio"/>	[%]
ex	User define	User defined	String		Value set in Web UI.

1.2.6.2.When setting to use as BLE is done

■Data sample (beacon mode)

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

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	temperature	Temperature	Double	<input type="radio"/>	[°C]
4	humidity	Humidity	Double	<input type="radio"/>	[%]
5	battery	Battery usage rate	Integer	<input type="radio"/>	[%]
6	memo	Notes	String		Value set in Web UI.

1.2.7.UNI-ELECTRONICS Wireless Carbon Dioxide Sensor

1.2.7.1.When setting to use as a beacon is done

■Data sample (beacon mode)

```
{
  "time":"2017-03-03T12:34:56.789+09:00",
  "deviceId":"f0ab54c2gcdf",
  "rssi":-82,
  "co2":653,
  "battery":254
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
9	co2	Carbon dioxide concentration	Integer	<input type="radio"/>	[ppm]
10	battery	Battery usage rate	Integer	<input type="radio"/>	[%]
ex	User define	User defined	String		Value set in Web UI.

1.2.7.2.When setting to use as BLE is done

■Data sample (beacon mode)

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

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	co2	Carbon dioxide concentration	Integer	<input type="radio"/>	[ppm]
4	battery	Battery usage rate	Integer	<input type="radio"/>	[%]
5	memo	Notes	String		Value set in Web UI.

1.2.8.UNI-ELECTRONICS Wireless Water Thermometer

1.2.8.1.When setting to use as a beacon is done

■Data sample (beacon mode)

```
{
  "time":"2017-12-08T12:34:56.789+09:00",
  "deviceId":"f0ab5e2bdcad",
  "rssi":-82,
  "temperature":12.34,
  "battery":100
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	<input type="radio"/>	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
8	status	Beacon status	String		Displayed by beacon control type. ("In" or "out")
9	temperature	Temperature	Double	<input type="radio"/>	[°C]
10	battery	Battery usage rate	Integer	<input type="radio"/>	[%]
ex	User define	User defined	String		Value set in Web UI.

1.2.8.2.When setting to use as BLE is done

■Data sample (beacon mode)

```
{
  "time":"2017-12-08T12:34:56.789+09:00",
  "deviceId":"f0ab5e2bdcad",
  "memo":"Logtta Water Sensor",
  "temperature":12.34,
  "battery":100
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	temperature	Temperature	Double	<input type="radio"/>	[°C]
4	battery	Battery usage rate	Integer	<input type="radio"/>	[%]
5	memo	Notes	String		Value set in Web UI.

1.2.9.RATOC systems Bluetooth Dust Sensor

■Data sample

```
{
  "deviceId":"dfb3f8c57912",
  "memo":"RATOC PM2.5",
  "time":"2017-12-07T20:55:48.173+09:00",
  "sensortime":"17-12-07T20:56:04",
  "pm25":15,
  "pm10":1,
  "pressure":999,
  "temperature":24,
  "humidity":18,
  "light":364,
  "mode":0
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	sensortime	Time measured	String	<input type="radio"/>	
4	pm25	PM2.5 concentration	Integer	<input type="radio"/>	[µg/m3]
5	pm10	PM10 concentration	Integer	<input type="radio"/>	[µg/m3]
6	pressure	Atmospheric pressure	Integer	<input type="radio"/>	[hPa]
7	temperature	Temperature	Integer	<input type="radio"/>	[°C]
8	humidity	Humidity	Integer	<input type="radio"/>	[%]
9	light	Illuminance	Integer	<input type="radio"/>	[lx]
10	mode	Measurement mode	Integer	<input type="radio"/>	0: Continuous, 1: One shot
11	memo	Notes	String		Value set in Web UI.

1.2.10.RATOC systems Bluetooth Air Quality Monitor

■Data sample

```
{
  "deviceId":"dfb308abcdef",
  "memo":"RATOC PM2.5V",
  "time":"2017-12-07T20:55:48.173+09:00",
  "sensortime":"17-12-07T20:56:04",
  "pm25":15,
  "pm10":1,
  "uvi": 0,
  "temperature":24.5,
  "humidity":18.1,
  "phumidity":18.2,
  "pressure":999.9,
  "initstate":" wait",
  "startstate":" stability",
  "light":364,
  "tvoc":123,
  "eco2":456,
  "mode":0
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	sensortime	Time measured	String	<input type="radio"/>	
4	pm25	PM2.5 concentration	Integer	<input type="radio"/>	[µg/m3]
5	pm10	PM10 concentration	Integer	<input type="radio"/>	[µg/m3]
6	uvi	UV index	Integer	<input type="radio"/>	
7	temperature	Temperature	Double	<input type="radio"/>	[°C]
8	humidity	Humidity	Double	<input type="radio"/>	[%]
9	phumidity	Humidity on atmospheric pressure sensor element.	Double	<input type="radio"/>	[%]
10	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
11	initstate	Initial stable state	String	<input type="radio"/>	"wait" or "stability"
12	startstate	Stable state at startup.	String	<input type="radio"/>	"wait" or "stability"
13	light	Illuminance	Integer	<input type="radio"/>	[lx]
14	tvoc	Total volatile organic compounds.	Integer	<input type="radio"/>	[ppb]
15	eco2	Equivalents carbon dioxide	Integer	<input type="radio"/>	[ppm]
16	mode	Measurement mode	Integer	<input type="radio"/>	0: Continuous, 1: One shot
17	memo	Notes	String		Value set in Web UI.

1.2.11.RATOC systems Bluetpph Watt Checker

■Data sample

```
{
  "deviceId":"123456abcdef",
  "memo":"RATOC WATT CHECKER",
  "time":"2017-12-07T20:55:48.173+09:00",
  "sensortime":"17-12-07T20:56:04",
  "current":17.7656,
  "voltage":104728,
  "power_consumption":400
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	sensortime	Time measured	String	<input type="radio"/>	
4	current	Current	Double	<input type="radio"/>	[mA]
5	voltage	Voltage	Integer	<input type="radio"/>	[V]
6	power_consumption	Power consumption	Integer	<input type="radio"/>	[W]
7	memo	Notes	String		Value set in Web UI.

1.2.12.Elecs Industry μ PRIsM

■Data sample

```
{
  "deviceId": "00089c161b39",
  "memo": "μ Prism",
  "time": "2018-09-09T09:55:18.65+09:00",
  "dataIndex": "111",
  "geoMagneticX": -10.9,
  "geoMagneticY": -24.6,
  "geoMagneticZ": -58.7,
  "accelX": 0.011,
  "accelY": 0.004,
  "accelZ": 1.042,
  "pressure": 1021.84,
  "humidity": 50.38,
  "temperature": 28.5,
  "uvi": 0.1,
  "ambientLight": 404.8,
  "ms": 65,
  "second": 18,
  "minute": 55,
  "hour": 9,
  "day": 9,
  "month": 9,
  "year": 18
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	dataIndex	Data index	Integer	<input type="radio"/>	
4	geoMagneticX	X axis geomagnetic torque	Double		[uT]
5	geoMagneticY	Y axis geomagnetic torque	Double		[uT]
6	geoMagneticZ	Z axis geomagnetic torque	Double		[uT]
7	accelX	X axis acceleration	Double		[G]
8	accelY	Y axis acceleration	Double		[G]
9	accelZ	Z axis acceleration	Double		[G]
10	pressure	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
11	humidity	Humidity	Double	<input type="radio"/>	[%RH]
12	temperature	Temperature	Double	<input type="radio"/>	[°C]
13	uvi	UV index	Double	<input type="radio"/>	
14	ambientLight	Illuminance	Double	<input type="radio"/>	[lx]
15	ms	Millisecond	Integer	<input type="radio"/>	
16	second	Second	Integer	<input type="radio"/>	
17	minute	Minute	Integer	<input type="radio"/>	
18	hour	Hour	Integer	<input type="radio"/>	
19	day	Day	Integer	<input type="radio"/>	
20	month	Month	Integer	<input type="radio"/>	
21	year	Year	Integer	<input type="radio"/>	
22	memo	Notes	String		Value set in Web UI.

2.PD Handler BLE with Lua on C language

The JSON data output to the log is in no particular order.

2.1.Beacon

Show subsection 1.1 "Beacon" of section 1 "PD Handler BLE with Node.js".

2.2.Sensor

It does not support to sensors in connection mode.

Please refer to beacon mode written in sub session 1.2 "Sensor" of session 1 "PD Handler BLE with Node.js " for supported sensor.

In addition, rssi information is also given when setting to use BLE.

Below is a description of devices operating with PD Handler BLE with Lua.

2.2.1.NAKAYO Push Botton

2.2.1.1.When setting to use as a beacon

■Data sample

```
{
  "time":"2017-12-08T12:34:56.789+09:00",
  "deviceId":" fc97c1aef545",
  "rssi":-68
  "uuid": "a903010014784824b2988e6823cfdefa",
  "major":"00c8",
  "minor":"ffe0",
  "push":0
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	○	ISO8601 extended format
2	deviceId	Device ID	String	○	Excluding ":" from the device address, lower case value.
3	appendixInfo	Appendix Information	String		Value set in Web UI.
4	rssi	Received signal strength indication	Integer	○	
5	type	Beacon type	String		Value set in Web UI.
6	data	Body of the data	String		Hexadecimal data
7	localname	Local name	String		Value set in Web UI.
5	uuid	uuid	String	○	
6	major	major	String	○	
7	minor	minor	String	○	
8	push	Pushed botton	Integer	○	Every time the button is pushed, 0 to 3 are output in order.
ex	User define	User defined	String		Value set in Web UI.

2.2.1.2. When setting to use as BLE

■Data sample

```
{
  "time": "2017-12-08T12:34:56.789+09:00",
  "deviceId": "fc97c1aef545",
  "memo": "Nakayo",
  "rssi": -68
  "uuid": "a903010014784824b2988e6823cfdefa",
  "major": "00c8",
  "minor": "ffe0",
  "push": 0
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
3	memo	Notes	String		Value set in Web UI.
4	rssi	Received signal strength indication.	Integer	<input type="radio"/>	
5	uuid	uuid	String	<input type="radio"/>	
6	major	major	String	<input type="radio"/>	
7	minor	minor	String	<input type="radio"/>	
8	push	Pushed botton	Integer	<input type="radio"/>	Every time the button is pushed, 0 to 3 are output in order.

3.PD Handler UART

*The JSON data output to the log is in no particular order.

3.1.EnOcean with Lua

Lua support is firmware 3.0 or later.

3.1.1.Human Sensor (EEP : A50701)

■Data sample

```
{
  "deviceId": "0400197A",
  "time": "2016-03-14T16:17:02.269+09:00",
  "svc": 4.764706,
  "pirs": "on",
  "EEP": "A50701",
  "memo": "Occupancy Sensor",
  "rssi": -71
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	svc	Supply voltage	Double	<input type="radio"/>	[V]
4	pirs	Detection result	String	<input type="radio"/>	"on" or "off"
5	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
6	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
7	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.2.Door Sensor (EEP : D50001)

■Data sample

```
{
  "deviceId": "04000A1B",
  "time": "2016-03-14T16:16:52.525+09:00",
  "contact": 0,
  "EEP": "D50001",
  "memo": "Contacts and Switches",
  "rssi": -65
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding "." from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	contact	Open/closed status	Integer	<input type="radio"/>	0 : Open , 1 : Closed
4	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
5	memo	Notes	String		Value set in Web UI.
6	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.3. Temperature Sensor (EEP : A50205)

■Data sample

```
{
  "deviceId": "04000C66",
  "time": "2016-03-14T16:16:59.958+09:00",
  "temperature": 25.254902,
  "EEP": "A50205",
  "memo": "Temperature Sensors",
  "rssi": -82
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding "." from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	temperature	temperature	Double	<input type="radio"/>	[°C]
4	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
5	memo	Notes	String		Value set in Web UI.
6	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.4. Temperature/Humidity Sensor(EEP : A50402)

■Data sample

```
{
  "deviceId": "0400267B",
  "time": "2017-08-31T14:26:39.283+09:00",
  "temperature": 28.160000,
  "humidity": 62.200006,
  "EEP": "A50402",
  "memo": "Temperature and Humidity Sensor",
  "rssi": -66
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	temperature	temperature	Double	<input type="radio"/>	[°C]
4	humidity	humidity	Double	<input type="radio"/>	[%]
5	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
6	memo	Notes	String		Value set in Web UI.
7	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.5. Temperature/Humidity Sensor(EEP : A50403)

■Data sample

```
{
  "deviceId": "040005C6",
  "time": "2016-03-14T16:15:58.904+09:00",
  "temperature": 25.122190,
  "humidity": 35.686275,
  "EEP": "A50403",
  "memo": "Temperature and Humidity Sensor",
  "rssi": -59
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	temperature	temperature	Double	<input type="radio"/>	[°C]
4	humidity	humidity	Double	<input type="radio"/>	[%]
5	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
6	memo	Notes	String		Value set in Web UI.
7	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.6.Single Phase Electricity sensor (EEP : A51201)

■Data sample

```
{
  "deviceId": "0400AE56",
  "time": "2016-03-14T16:15:58.904+09:00",
  "electricity": 15.0,
  "dataType": "W",
  "EEP": "A51201",
  "memo": "Automated Meter Reading (AMR)",
  "rssi": -87
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding "." from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	electricity	Power or accumulated power	Double	<input type="radio"/>	Power or accumulated power.
4	dataType	Data type	String	<input type="radio"/>	"kWh"or"W"
5	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
6	memo	Notes	String		Value set in Web UI.
7	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.7.Three Channel Current Sensor (EEP : D23202)

■Data sample

```
{
  "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",
  "rssi": -63
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding “:” from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	channel1	Current value of channel 1	Double	<input type="radio"/>	[A]
4	channel2	Current value of channel 2	Double	<input type="radio"/>	[A]
5	channel3	Current value of channel 3	Double	<input type="radio"/>	[A]
5	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
6	memo	Notes	String		Value set in Web UI.
7	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.8. Atmospheric sensor (EEP : A50501)

■Data sample

```
{
  "deviceId": "0401520B",
  "time": "2016-06-07T15:58:22.927+09:00",
  "barometer": 1010.850464,
  "telegram_type": "Hearbeat",
  "EEP": "A50501",
  "memo": "barometer",
  "rssi": -71
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding "." from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	barometer	Atmospheric pressure	Double	<input type="radio"/>	[hPa]
4	telegram_type	telegram type	String	<input type="radio"/>	"Hearbeat" or "Event triggered"
5	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
6	memo	Notes	String		Value set in Web UI.
7	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.9. Illuminance Sensor(EEP : A50602)

■Data sample

```
{
  "deviceId": "04004715",
  "time": "2016-06-07T15:58:28.150+09:00",
  "svc": 2.780000,
  "ill1": 260.000000,
  "ill2": 260.000000,
  "EEP": "A50602",
  "memo": "ill",
  "rssi": -65
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding "." from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	svc	Supply voltage	Double		[V]
4	ill1	Illuminance1	Double	<input type="radio"/>	[lux]
5	ill2	Illuminance2	Double	<input type="radio"/>	[lux]
6	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
7	memo	Notes	String		Value set in Web UI.
8	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.10. Carbon Dioxide Concentration Sensor (EEP: A50904)

■ Data sample

```
{
  "deviceId": "040004FF",
  "time": "2016-06-07T15:34:15.126+09:00",
  "humidity": 52.000000,
  "temperature": 28.000000,
  "concentration": 690,
  "EEP": "A50904",
  "memo": "CO2",
  "rssi": -84
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	humidity	humidity	Double	<input type="radio"/>	[%]
4	temperature	temperature	Double	<input type="radio"/>	[°C]
5	concentration	Carbon dioxide concentration	Integer	<input type="radio"/>	[ppm]
6	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
7	memo	Notes	String		Value set in Web UI.
8	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.11. Digital Input Sensor(EEP : A53005)

■Data sample

```
{
  "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",
  "rssi": -58
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding ":" from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	vdd	Supply voltage	Double		[V]
4	signal_type	Signal type	String	<input type="radio"/>	"Normal signal"or"Heart beat signal"
5	count	Ordinal number	Integer	<input type="radio"/>	0 to 127
6	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
7	memo	Notes	String		Value set in Web UI.
8	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.12.2 Input Locker Switch(EEP : F60204)

■Data sample

```
{
  "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",
  "rssi": -70
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding "." from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	ebo	Energy state	String	<input type="radio"/>	"pressed" or "released"
4	rbi	The state "I" of the rocker switch B	String	<input type="radio"/>	"pressed" or "released"
5	rbo	The state "O" of the rocker switch B	String	<input type="radio"/>	"pressed" or "released"
6	rai	The state "I" of the rocker switch A	String	<input type="radio"/>	"pressed" or "released"
7	rao	The state "O" of the rocker switch A	String	<input type="radio"/>	"pressed" or "released"
8	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
9	memo	Notes	String		Value set in Web UI.
10	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.1.13. In RAW data mode

■Data sample

```
{
  "deviceId": "0400197A",
  "time": "2016-03-14T16:45:32.643+09:00",
  "data": "55000c020ae66200000400197a1c080b8720013da6",
  "EEP": "A50701",
  "memo": "raw data",
  "rssi": -63
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	deviceId	Device ID	String	<input type="radio"/>	Excluding “.” from the device address, lower case value.
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	data	Payload	String	<input type="radio"/>	
4	EEP	EnOcean profile	String	<input type="radio"/>	Value set in Web UI.
5	memo	Notes	String		Value set in Web UI.
6	rssi	Received signal strength indication	Integer	<input type="radio"/>	

3.2.Wi-SUN

3.2.1. Instantaneous Power (B-Route)

■Data sample

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

#	JSON Key	Description	Data Type	Always	Remarks
1	address	Address	String	<input type="radio"/>	Address of smart power meter
2	date	Date of the target data	String	<input type="radio"/>	
3	time	Time of target data	String	<input type="radio"/>	
4	inst	Instantaneous power	Integer	<input type="radio"/>	[W]

3.2.2. Accumulated Power (B-Route)

■Data sample

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

#	JSON Key	Description	Data Type	Always	Remarks
1	address	Address	String	<input type="radio"/>	Address of smart power meter
2	time	Date and time of the target data	String	<input type="radio"/>	
3	ratio	Ratio	Integer	<input type="radio"/>	0 to 999,999
4	unit	Unit	Double	<input type="radio"/>	One of the following 1 , 0.1 , 0.01 , 0.001 , 0.0001 10 , 100 , 1000 , 10000
5	cumu	Accumulated power in the positive direction	Integer	<input type="radio"/>	0 to 99,999,999. However -1 is no data
6	cumu_re	Accumulated power in the reverse direction	Integer	<input type="radio"/>	0 to 99,999,999. However -1 is no data

* Calculation method

- Accumulated power in the positive direction

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

- Accumulated power in the reverse direction

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

4.PD Handler HVSMC

* The JSON data output to the log is in no particular order.

4.1.RAW Data

■Data sample

```
{
  "8D": "0123456789AB",
  "time": "2018-11-21T12:34:56+09:00",
  "memo": "raw data",
  "datatype": "raw_data",
  "raw_data": [
    {
      "EPC": "80",
      "PDC": 1,
      "EDT": "30"
    },
    {
      "EPC": "88",
      "PDC": 1,
      "EDT": "41"
    }
  ]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	8D	Serial number	String	<input type="radio"/>	
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
4	datatype	Data type	String	<input type="radio"/>	raw_data
5	raw_data		JSON object array	<input type="radio"/>	
6	EPC	ECHONET property	String	<input type="radio"/>	
7	PDC	Property data counter	Integer	<input type="radio"/>	
8	EDT	ECHONET property value data	String	<input type="radio"/>	

4.2. Operating Status

■Data sample

```
{
  "8D": "0123456789AB",
  "time": "2018-11-21T12:34:56+09:00",
  "80": "ON",
  "memo": "pd-handler-hvsmc",
  "datatype": "op_stat"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	8D	Serial number	String	<input type="radio"/>	
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	80	Operating Status	String	<input type="radio"/>	"ON" or "OFF"
4	memo	Notes	String		Value set in Web UI.
5	datatype	Data type	String	<input type="radio"/>	op_stat

4.3. Abnormal Status

■Data sample

```
{
  "8D": "0123456789AB",
  "time": "2018-11-21T12:34:56+09:00",
  "88": "not",
  "memo": "pd-handler-hvsmc",
  "datatype": "fault_stat"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	8D	Serial number	String	<input type="radio"/>	
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	88	Abnormal Status	String	<input type="radio"/>	"occurred" or "not"
4	memo	Notes	String		Value set in Web UI.
5	datatype	Data type	String	<input type="radio"/>	fault_stat

4.4.ECHONET Lite Property Information

■Data sample

```
{
  "8D": "0123456789AB",
  "time": "2018-11-21T12:34:56+09:00",
  "82": "F",
  "9D": "808188",
  "9E": "81E1",
  "9F": "808182888A8D97989D9E9FD3D4E0E1E2E3E4E5E6E7C1C2C3C4C5C6C7CACBCCCDCE",
  "memo": "pd-handler-hvsmc",
  "datatype": "echonet_attr"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	8D	Serial number	String	<input type="radio"/>	
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	82	Standard version	String	<input type="radio"/>	
4	9D	State announcement property map	String	<input type="radio"/>	
5	9E	Set property map	String	<input type="radio"/>	
6	9F	Get property map	String	<input type="radio"/>	
7	memo	Notes	String		Value set in Web UI.
8	datatype	Data type	String	<input type="radio"/>	echonet_attr

4.5.High Voltage Smart Power Meter Property Information

■Data sample

```
{
  "8D": "0123456789AB",
  "time": "2018-11-21T12:34:56+09:00",
  "D3": 1,
  "D4": 0.1,
  "E0": 1,
  "E5": 8,
  "E6": 0.001,
  "C4": 8,
  "C5": 0.001,
  "C7": 0.001,
  "CC": 6,
  "CD": 0.01,
  "memo": "pd-handler-hvsmc",
  "datatype": "hvsm_attr"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	8D	Serial number	String	<input type="radio"/>	
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	D3	Coefficient	Integer	<input type="radio"/>	
4	D4	Scale factor	Double	<input type="radio"/>	
5	E0	Fixed date	Integer	<input type="radio"/>	
6	E5	Number of significant digits of accumulated active power	Integer	<input type="radio"/>	
7	E6	Unit of cumulated effective electric power	Double	<input type="radio"/>	[kWh]
8	C4	Number of significant digits of demand power	Integer	<input type="radio"/>	
9	C5	Unit of Demand Power	Double	<input type="radio"/>	[kWh]
10	C7	Unit of cumulative maximum demand power	Double	<input type="radio"/>	[kWh]
11	CC	Number of significant digits of cumulative reactive power amount (lag phase)	Integer	<input type="radio"/>	
12	CD	Unit of cumulative reactive power amount (lag phase)	Double	<input type="radio"/>	[kvarh]
13	memo	Notes	String		Value set in Web UI.
14	datatype	Data type	String	<input type="radio"/>	hvsm_attr

4.6. Regular Time Measuremen Value

■Data sample

```
{
  "8D": "0123456789AB",
  "time": "2018-11-11T12:30:00+09:00",
  "E3": 1234567.8,
  "C3": 1234567.8,
  "CB": 1234567.8,
  "memo": "pd-handler-hvsmc",
  "datatype": "fixed_measured"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	8D	Serial number	String	<input type="radio"/>	
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	E3	Predetermined accumulated active power amount measurement value	Double	<input type="radio"/>	
4	C3	Scheduled demand power	Double	<input type="radio"/>	
5	CB	Regular power measurement Integrated reactive power (lag phase) measurement value	Double	<input type="radio"/>	
6	memo	Notes	String		Value set in Web UI.
7	datatype	Data type	String	<input type="radio"/>	fixed_measured

4.7. Measuremen Value

■Data sample

```
{
  "8D": "0123456789AB",
  "time": "2018-11-11T12:34:56+09:00",
  "E2": 1234567.8,
  "E4": 1234567.8,
  "CA": 1234567.8,
  "memo": "pd-handler-hvsmc",
  "datatype": "measured"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	8D	Serial number	String	<input type="radio"/>	
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	E2	Cumulated effective power amount measurement value	Double	<input type="radio"/>	
4	E4	Measured cumulative active power amount	Double	<input type="radio"/>	
5	CA	Measurement integrated value reactive power amount (lag phase) measured value	Double	<input type="radio"/>	
6	memo	Notes	String		Value set in Web UI.
7	datatype	Data type	String	<input type="radio"/>	measured

4.8. Demand power

■Data sample

```
{
  "8D": "0123456789AB",
  "time": "2018-11-21T12:34:56+09:00",
  "C1": 1234567.8,
  "C2": 1234567.8,
  "memo": "pd-handler-hvsmc",
  "datatype": "demand",
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	8D	Serial number	String	<input type="radio"/>	
2	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
3	C1	Maximum demand power per month	Double	<input type="radio"/>	
4	C2	Cumulative maximum demand power	Double	<input type="radio"/>	
5	memo	Notes	String		Value set in Web UI.
6	datatype	Data type	String	<input type="radio"/>	demand

5.PD Handler Modbus

5.1.Modbus Client(Modbus master)

5.1.1.Polling operation to PLC device

■Data sample (reading register output or register input by TCP protocol)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"tcp",
  "node":"172.16.7.250",
  "port":1502,
  "unit":255,
  "memo":"PLC01",
  "address":31,
  "function":3,
  "data_type":"uint16_t",
  "values":[2,0,1234,5678,9876]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu"
3	node	IP address of PLC device	String	<input type="radio"/>	Value set in Web UI. "tcp"only
4	port	TCP port number	Integer	<input type="radio"/>	Value set in Web UI. "tcp"only
5	device	Device file name	String		Value set in Web UI. "rtu"only
6	unit	Modbus Unit ID	Integer	<input type="radio"/>	Value set in Web UI.
7	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
8	address	Registers address to be read data.	Integer	<input type="radio"/>	Value set in Web UI.
9	function	Modbus function code	integer	<input type="radio"/>	Value set in Web UI. 3: Register output 4: Register input
10	data_type	Data type	String	<input type="radio"/>	Value set in Web UI. "uint16_t" : Unsigned 16bits "int16_t": Signed16bits "uint32lsb_t": Unsigned 32bits LSB "uint32msb_t": Unsigned 32bits MSB "int32lsb_t": Signed 32bits LSB "int32msb_t": Signed 32bits MSB
11	values	Body of the data	Integer array	<input type="radio"/>	The number of arrays is variable according to the setting of the number of read registers.

■Data sample (reading of digital output or digital input by RTU protocol)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"rtu",
  "device":"/dev/ttyRS485",
  "unit":21,
  "memo":"PLC04",
  "address":37,
  "function":2,
  "values":[1,0,0,0,0,0,1,0,1,0,1,1,0,1,0,1,0,0,1,1,1]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu"
3	node	IP address of PLC device.	String	<input type="radio"/>	Value set in Web UI. "tcp" only
4	port	TCP port number	Integer	<input type="radio"/>	Value set in Web UI. "tcp" only
5	device	Device file name	String		Value set in Web UI. "rtu" only
6	unit	Modbus Unit ID	Integer	<input type="radio"/>	Value set in Web UI.
7	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
8	address	Registers address from which data was read.	Integer	<input type="radio"/>	Value set in Web UI.
9	function	Modbus function code	Integer	<input type="radio"/>	Value set in Web UI. 1: Digital output 2: Digital input
10	values	Body of the data	Integer array	<input type="radio"/>	0 or 1. The number of arrays is variable according to the setting of the number of read registers.

5.1.2.On demand operation from the cloud

■Request message sample (reading register output or register input by TCP protocol)

```
{
  "protocol": "tcp",
  "node": "172.16.7.250",
  "port": 1502,
  "unit": 255,
  "address": 31,
  "function": 3,
  "number": 5,
  "data_type": "uint16_t"
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	protocol	Protocol	String	○	"tcp" or "rtu"
2	node	IP address of PLC device	String	△	IP address of PLC device, required for "tcp".
3	port	TCP port number	Integer	△	TCP port number of PLC device, required for "tcp".
4	device	Device file name	String	△	Serial port connecting PLC equipment, required for "rtu".
5	unit	Modbus Unit ID	Integer	△	Must be for "rtu". If omitted at "tcp", 255.
6	address	Registers address to be read data.	Integer *1		If omitted, 0.
7	function	Modbus function code	Integer *1	○	3: read holding registers 4: read input registers
8	number	Number of register to be read.	Integer *1		If omitted, 1.
9	data_type	Data type	String		"uint16_t" : Unsigned 16bits "int16_t": Signed 16bits "uint32lsb_t": Unsigned 32bits LSB "uint32msb_t": Unsigned 32bits MSB "int32lsb_t": Signed 32bits LSB "int32msb_t": Signed 32bits MSB If omitted, "uint16_t".

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

■Response message sample (Register output by TCP protocol or reading register input)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "reply_to":"84bfb66e5a0841732e28463bb91c297c",
  "result":"done",
  "protocol":"tcp",
  "node":"172.16.7.250",
  "port":1502,
  "unit":255,
  "memo":"PLC01",
  "address":31,
  "function":3,
  "data_type":"uint16_t",
  "values":[2,0,1234,5678,9876]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Requested value. "tcp" or "rtu"
5	node	IP address of PLC device	String		Requested value. "tcp" only
6	port	TCP port number	Integer		Requested value. "tcp" only
7	device	Device file name	String		Requested value. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	Requested value.
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	address	Registers address from which data was read.	Integer	<input type="radio"/>	Requested value.
11	function	Modbus function code	integer	<input type="radio"/>	Requested value.
12	data_type	Data type	String	<input type="radio"/>	Requested value.
13	values	Body of the data	Integer array	<input type="radio"/>	The number of arrays is variable according to the number of registers requested with the number key.

■ Request message sample (reading of digital output or digital input by RTU protocol)

```
{
  "protocol": "rtu",
  "device": "/dev/ttyRS485",
  "unit": 21,
  "address": 37,
  "function": 2,
  "number": 20
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	protocol	Protocol	String	○	"tcp" or "rtu"
2	node	IP address of PLC device	String	△	IP address of PLC device, required for "tcp".
3	port	TCP port number	Integer	△	TCP port number of PLC device, required for "tcp".
4	device	Device file name	String	△	Serial port connecting PLC equipment, required for "rtu".
5	unit	Modbus Unit ID	Integer	△	Must be for "rtu". If omitted at "tcp", 255.
6	address	Registers address to be read data.	Integer *1		If omitted, 0.
7	function	Modbus function code	Integer *1	○	1: read coils 2: read discrete inputs
8	number	Number of bit to be read.	Integer *1		If omitted, 1.

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

■Response message sample (reading of digital output or digital input by RTU protocol)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "reply_to":"5762a76a3235c71c5759029f078a8ca2",
  "result":"done",
  "protocol":"rtu",
  "device":"/dev/ttyRS485",
  "unit":21,
  "memo":"PLC04",
  "address":37,
  "function":2,
  "values":[1,0,0,0,0,0,1,0,1,0,1,1,0,1,0,1,0,0,1,1]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Requested value. "tcp" or "rtu"
5	node	IP address of PLC device	String		Requested value. "tcp" only
6	port	TCP port number	Integer		Requested value. "tcp" only
7	device	Device file name	String		Requested value. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	Requested value.
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	address	Registers address from which data was read.	Integer	<input type="radio"/>	Requested value.
11	function	Modbus function code	integer	<input type="radio"/>	Requested value.
12	values	Body of the data	Integer array	<input type="radio"/>	0 or 1. The number of arrays is variable according to the number of registers requested with the number key.

■Request message sample (write to register input by TCP protocol)

```
{
  "protocol": "tcp",
  "node": "172.16.7.250",
  "port": 1502,
  "unit": 255,
  "address": "0x0ab",
  "function": 16,
  "data_type": "uint32lsb_t",
  "values": [42949672951, 21474836471]
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	protocol	Protocol	String	○	"tcp" or "rtu"
2	node	IP address of PLC device	String	△	IP address of PLC device, required for "tcp".
3	port	TCP port number	Integer	△	TCP port number of PLC device, required for "tcp".
4	device	Device file name	String	△	Serial port connecting PLC equipment, required for "rtu".
5	unit	Modbus Unit ID	Integer	△	Must be for "rtu". If omitted at "tcp", 255.
6	address	Registers address to be write data.	Integer *1		If omitted, 0.
7	function	Modbus function code	Integer *1	○	6:write_single_register 16:write_multiple_registers 23:write_and_read_registers
8	data_type	Data type	String		"uint16_t": Unsigned 16bits "int16_t": Signed 16bits "uint32lsb_t": Unsigned 32bits LSB "uint32msb_t": Unsigned 32bits MSB "int32lsb_t": Signed 32bits LSB "int32msb_t": Signed 32bits MSB If omitted, "uint16_t"
9	values	Body of the write data	Integer array	○	If the function key is 6, write the first 1 register.

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

■Response message sample (write to register input by TCP protocol)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "reply_to":"73771103b4765ed0ce859ac912321c04",
  "result":"done",
  "protocol":"tcp",
  "node":"172.16.7.250",
  "port":1502,
  "unit":255,
  "address":"0x0ab",
  "function":16,
  "data_type":"uint32lsb_t",
  "values":[42949672951,21474836471]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Requested value. "tcp" or "rtu"
5	node	IP address of PLC device	String		Requested value. "tcp" only
6	port	TCP port number	Integer		Requested value. "tcp" only
7	device	Device file name	String		Requested value. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	Requested value.
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	address	Registers address from which data was wrtten	Integer	<input type="radio"/>	Requested value.
11	function	Modbus function code	Integer	<input type="radio"/>	Requested value.
12	data_type	Data type	String	<input type="radio"/>	Requested value.
13	values	Body of the written data.	Integer array	<input type="radio"/>	Requested value.

■Request message sample (writing to digital input by RTU protocol)

```
{
  "protocol": "rtu",
  "device": "/dev/ttyRS485",
  "unit": 21,
  "address": "0x0ce",
  "function": 15,
  "values": [0,0,0,1,1,1,0,1,1,0,1,1,0,1,1]
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	protocol	Protocol	String	○	"tcp" or "rtu"
2	node	IP address of PLC device	String	△	IP address of PLC device, required for "tcp".
3	port	TCP port number	Integer	△	TCP port number of PLC device, required for "tcp".
4	device	Device file name	String	△	Serial port connecting PLC equipment, required for "rtu".
5	unit	Modbus Unit ID	Integer	△	Must be for "rtu". If omitted at "tcp", 255.
6	address	Registers address to be write data.	Integer *1		If omitted, 0.
7	function	Modbus function code	Integer *1	○	5:write_single_coil 15:write_multiple_coils
8	values	Body of the write data	Integer array	○	0 or 1. If the function key is 5, write the first 1 bit.

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

■Response message sample (writing to digital input by RTU protocol)

```
{
  "time": "2017-02-03T14:44:37.020+09:00",
  "reply_to": "0408f69db38b4d89f25d026d6d9449b7",
  "result": "done",
  "protocol": "rtu",
  "device": "/dev/ttyRS485",
  "unit": 21,
  "address": "0x0ce",
  "function": 15,
  "values": [0,0,0,1,1,1,0,1,1,0,1,1,0,1,1]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Requested value. "tcp" or "rtu"
5	node	IP address of PLC device	String		Requested value. "tcp" only
6	port	TCP port number	Integer		Requested value. "tcp" only
7	device	Device file name	String		Requested value. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	Requested value.
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	address	Registers address from which data was written	Integer	<input type="radio"/>	Requested value.
11	function	Modbus function code	Integer	<input type="radio"/>	Requested value.
12	values	Body of the written data.	Integer array	<input type="radio"/>	Requested value.

■Request message sample (reading slave ID)

```
{
  "protocol": "rtu",
  "device": "/dev/ttyRS485",
  "unit": 21,
  "function": 17
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	protocol	Protocol	String	<input type="radio"/>	"rtu" only
2	device	Device file name	String	<input type="radio"/>	
3	unit	Modbus Unit ID	Integer	<input type="radio"/>	
4	function	Modbus function code	Integer *1	<input type="radio"/>	17: report_slave_id

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

■Reply message sample (read slave ID)

```
{
  "time": "2017-02-03T14:44:37.020+09:00",
  "reply_to": "72cf056269d6bcd150df8125fbe04710",
  "result": "done",
  "protocol": "rtu",
  "device": "/dev/ttyRS485",
  "unit": 21,
  "function": 17,
  "values": [ 7, 12 ]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Requested value.
7	device	Device file name	String		Requested value.
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	Requested value.
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
11	function	Modbus function code	Integer	<input type="radio"/>	Requested value.
12	values	List of Modbus Unit IDs that connected.	Integer array	<input type="radio"/>	

■Response message sample (on error)

```
{
  "time": "2017-02-03T14:44:37.020+09:00",
  "reply_to": "7408f69d838b4d89f257036d6d9449b7",
  "result": "not queuing",
  "reason": "not specified 'function' at least"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data.	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5)	String	<input type="radio"/>	
3	result	Status	String	<input type="radio"/>	"not queuing": Incorrect request message. "failed": Failed to connect to PLC equipment, etc.
4	reason	Reason for error.	String	<input type="radio"/>	

5.2.Modbus Server(Modbus slave)

5.2.1.Write operation from PLC device

■Data sample (write to register input by TCP protocol)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"tcp",
  "node":"172.16.7.240",
  "port":502,
  "unit":255,
  "memo":"PLC Server 01",
  "address":31,
  "function":6,
  "values":[5678]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data.	String	<input type="radio"/>	ISO8601 extended format
2	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu"..
3	node	IP address of listen.	String		Value set in Web UI. "tcp"only
4	port	TCP port number	Integer		fixed 502, tcp"only.
5	device	Device file name	String		Value set in Web UI. "rtu"only
6	unit	Modbus Unit ID	Integer	<input type="radio"/>	fixed 255 for "tcp", Value set from Web UI for "rtu".
7	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
8	address	Registers address from which data was wrtn.	Integer	<input type="radio"/>	0 to (2048 - registers)
9	function	Modbus function code	Integer	<input type="radio"/>	6: write single register 16:write multiple registres
10	values	Body of the written data.	Integer aray	<input type="radio"/>	Unsigned 16bits. The number of arrays varies according to the number of registers written.

■Data sample (writing to digital input by RTU protocol)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "protocol":"rtu",
  "device":"/dev/ttyRS485",
  "unit":21,
  "memo":"PLC Server 01",
  "address":37,
  "function":5,
  "values":[1]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data.	String	<input type="radio"/>	ISO8601 extended format
2	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu".
3	node	IP address of listen.	String		Value set in Web UI. "tcp" only
4	port	TCP port number	Integer		fixed 502, "tcp" only.
5	device	Device file name	String		Value set in Web UI. "rtu" only
6	unit	Modbus Unit ID	Integer	<input type="radio"/>	fixed 255 for "tcp", Value set from Web UI for "rtu".
7	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
8	address	Registers address from which data was wrtn.	Integer	<input type="radio"/>	0 to (2048 - bits)
9	function	Modbus function code	Integer	<input type="radio"/>	6: write single register 16:write multiple registres
10	values	Body of the written data.	Integer aray	<input type="radio"/>	0 or 1. The number of arrays varies according to the number of bits written.

5.2.2. On demand operation from the cloud

■ Request message sample (reading register output or register input)

```
{  
  "function":3,  
  "address":31,  
  "number":5,  
  "data_type":"uint16_t"  
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	function	Modbus function code	Integer *1	○	3: read holding registers 4: read input registers
2	address	Registers address to be read data.	Integer *1		If omitted, 0.
3	number	Number of register to be read.	Integer *1		If omitted, 1.
4	data_type	Data type	String		"uint16_t" : Unsigned 16bits "int16_t": Signed 16bits "uint32lsb_t": Unsigned 32bits LSB "uint32msb_t": Unsigned 32bits MSB "int32lsb_t": Signed 32bits LSB "int32msb_t": Signed 32bits MSB If omitted, "uint16_t".

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

■Response message sample (reading register output or register input)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "reply_to":"e4f87480e871555105cc81aac50e5e54",
  "result":"done",
  "protocol":"tcp",
  "node":"172.16.7.249",
  "port":502,
  "unit":255,
  "memo":"PLC Server 01",
  "address":31,
  "function":3,
  "data_type":"uint16_t",
  "values":[2,0,1234,5678,9876]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu"
5	node	IP address of OpenBlocks	String		"tcp" only
6	port	TCP port number	Integer		fixed 502, tcp" only.
7	device	Device file name	String		Value set in Web UI. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	fixed 255 for "tcp", Value set from Web UI for "rtu".
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	address	Registers address from which data was read.	Integer	<input type="radio"/>	Requested value.
11	function	Modbus function code	integer	<input type="radio"/>	Requested value.
12	data_type	Data type	String	<input type="radio"/>	Requested value.
13	values	Body of the data.	Integer array	<input type="radio"/>	The number of arrays is variable according to the number of registers requested with the number key.

* The values of protocol, node, port, device, unit, memo are the values set for the device having the device number of the UNIX domain socket that received the request message.

■Request message sample (reading digital output or digital input)

```
{
  "function":2
  "address":37,
  "number":20
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	function	Modbus function code	Integer *1	<input type="radio"/>	1: read coils 2: read discrete inputs
2	address	Registers address to be read data.	Integer *1		If omitted, 0.
3	number	Number of bits to be read.	Integer *1		If omitted, 1.

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

■Response message sample (reading digital output or digital input)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "reply_to":"e5910e15403f5e2158a5776cd7136eeb",
  "result":"done"
  "protocol":"rtu",
  "device":"/dev/ttyRS485",
  "unit":21,
  "memo":"PLC04",
  "address":37,
  "function":2,
  "values":[1,0,0,0,0,0,1,0,1,0,1,1,0,1,0,1,0,0,1,1]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu"
5	node	IP address of OpenBlocks	String		"tcp" only
6	port	TCP port number	Integer		fixed 502, tcp" only.
7	device	Device file name	String		Value set in Web UI. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	fixed 255 for "tcp", Value set from Web UI for "rtu".
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	address	Registers address from which data was read.	Integer	<input type="radio"/>	Requested value.
11	function	Modbus function code	integer	<input type="radio"/>	Requested value.
12	values	Body of the data.	Integer array	<input type="radio"/>	0 or 1. The number of arrays is variable according to the number of bits requested with the number key.

* The values of protocol, node, port, device, unit, memo are the values set for the device having the device number of the UNIX domain socket that received the request message.

■Request message sample (write to register output or register input)

```
{
  "function":16,
  "address":"0x0ab",
  "function":16,
  "data_type":"uint32lsb_t",
  "values":[42949672951,21474836471]
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	function	Modbus function code	Integer *1	○	6:write_single_register 10:write_single_input_registers 16:write_multiple_registers 20:write_multiple_input_registers 23:write_and_read_registers
2	address	Registers address to be write data.	Integer *1		If omitted, 0.
3	data_type	Data type	String		"uint16_t" : Unsigned 16bits "int16_t": Signed16bits "uint32lsb_t": Unsigned 32bits LSB "uint32msb_t": Unsigned 32bits MSB "int32lsb_t": Signed 32bits LSB "int32msb_t": Signed 32bits MSB If omitted,"uint16_t".
4	values	Body of the write data	Integer array	○	If the function key is 6, write the first 1 register.

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

*Of the functions, 10:write_single_input_registers and 20:write_multiple_input_registes is a function that does not exist in the original Modubus protocol.

■Response message sample (write to register output or register input)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "reply_to":" 35cf8fa6243d87e0ebb0c2aaaf8eeecf",
  "result":"done",
  "protocol":"tcp",
  "node":"172.16.7.249",
  "port":502,
  "unit":255,
  "address":"0x0ab",
  "function":16,
  "data_type":"uint32lsb_t",
  "values":[42949672951,21474836471]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done".
4	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu"
5	node	IP address of OpenBlocks	String		"tcp" only
6	port	TCP port number	Integer		fixed 502, tcp"only.
7	device	Device file name	String		Value set in Web UI. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	fixed 255 for "tcp", Value set from Web UI for "rtu".
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	address	Registers address from which data was wrtten	Integer	<input type="radio"/>	Requested value.
11	function	Modbus function code	Integer	<input type="radio"/>	Requested value.
12	data_type	Data type	String	<input type="radio"/>	Requested value.
13	values	Body of the written data.	Integer array	<input type="radio"/>	Requested value.

* The values of protocol, node, port, device, unit, memo are the values set for the device having the device number of the UNIX domain socket that received the request message.

■Request message sample (writing to digital output or digital input)

```
{
  "function":15,
  "address":"0x0ce",
  "values":[0,0,0,1,1,1,0,1,1,0,1,1,0,1,1]
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	function	Modbus function code	Integer *1	<input type="radio"/>	5:write_single_coil 9:write_single_discrete_input 15:write_multiple_coils 19:write_multiple_discrete_input
2	address	Registers address to be write data.	Integer *1	<input type="checkbox"/>	If omitted, 0.
3	values	Body of the write data	Integer array	<input type="radio"/>	0 or 1. If the function key is 5, write the first 1 bit.

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

*Of the functions, 9:write_single_discrete_input and 19 write_multiple_discrete_input is a function that does not exist in the original Modubus protocol.

■Response message sample (writing to digital output or digital input)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "reply_to":"c4348e30643dac56cb61bac9743729e7",
  "result":"done",
  "protocol":"rtu",
  "device":"/dev/ttyRS485",
  "unit":21,
  "address":0x0ce,
  "function":15,
  "values":[ 0,0,0,1,1,1,0,1,1,0,1,1,0,1,1]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu"
5	node	IP address of OpenBlocks	String		"tcp" only
6	port	TCP port number	Integer		fixed 502, tcp"only.
7	device	Device file name	String		Value set in Web UI. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	fixed 255 for "tcp", Value set from Web UI for "rtu".
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	address	Registers address from which data was written	Integer	<input type="radio"/>	Requested value.
11	function	Modbus function code	Integer	<input type="radio"/>	Requested value.
12	values	Body of the written data.	Integer array	<input type="radio"/>	Requested value.

* The values of protocol, node, port, device, unit, memo are the values set for the device having the device number of the UNIX domain socket that received the request message.

■Request message sample (reading slave ID)

```
{
  "function":17
}
```

#	JSON Key	Description	Data Type	Required	Remarks
1	function	Modbus function code	Integer *1	<input type="radio"/>	7:report_slave_id

*1 Hexadecimal notation starting with "0x" in string data type is also possible.

■Response message sample (writing to digital output or digital input)

```
{
  "time":"2017-02-03T14:44:37.020+09:00",
  "reply_to":"e553cae505e64e305373c73d7dd6cd31",
  "result":"done",
  "protocol":"rtu",
  "device":"/dev/ttyRS485",
  "unit":21,
  "function":17,
  "values":[ 21,255]
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5).	String	<input type="radio"/>	
3	result	Acquisition status	String	<input type="radio"/>	If it succeeds, "done"
4	protocol	Protocol	String	<input type="radio"/>	Value set in Web UI. "tcp" or "rtu"
5	node	IP address of OpenBlocks	String		"tcp" only
6	port	TCP port number	Integer		fixed 502, tcp"only.
7	device	Device file name	String		Value set in Web UI. "rtu" only
8	unit	Modbus Unit ID	Integer	<input type="radio"/>	fixed 255 for "tcp", Value set from Web UI for "rtu".
9	memo	Notes	String	<input type="radio"/>	Value set in Web UI.
10	function	Modbus function code	Integer	<input type="radio"/>	Requested value.
11	values	List of Modbus Unit IDs that connected.	Integer array	<input type="radio"/>	

* The values of protocol, node, port, device, unit, memo are the values set for the device having the device number of the UNIX domain socket that received the request message.

■Response message sample (on error)

```
{
  "time": "2017-02-03T14:44:37.020+09:00",
  "reply_to": "7408f69d838b4d89f257036d6d9449b7",
  "result": "not queuing",
  "reason": "not specified 'function' at least"
}
```

#	JSON Key	Description	Data Type	Always	Remarks
1	time	Timestamp of data.	String	<input type="radio"/>	ISO8601 extended format
2	reply_to	Hash value of request message (MD5)	String	<input type="radio"/>	
3	result	Status	String	<input type="radio"/>	"not queuing": Incorrect request message. "failed": Failed to connect to PLC equipment, etc.
4	reason	Reason for error.	String	<input type="radio"/>	

Plat'Home Co., Ltd.
NIHON BUILDING KUDANBEKKAN,3F
4-2-3,Kudankita,Chiyoda-ku,TOKYO 102-0073 JAPAN