Plat' Home
TECHNOLOGY to serve you.

OpenBlocks IoT Family向け
データーハンドリング設定
リファレンスガイド

Ver.3.3.0

ぷらっとホーム株式会社

■ 商標について

・ 文中の社名、商品名等は各社の商標または登録商標である場合があります。
・ その他記載されている製品名などの固有名詞は、各社の商標または登録商標です。

■ 使用にあたって

・ 本書の内容の一部または全部を、無断で転載することはご遠慮ください。
・ 本書の内容は予告なしに変更することがあります。
・ 本書の内容については正確を期するように努めていますが、記載の誤りなどにご指摘がございましたら弊社サポート窓口へご連絡ください。
  また、弊社公開の WEB サイトにより本書の最新版をダウンロードすることが可能です。
・ 本装置の使用にあたっては、生命に関わる危険性のある分野での利用を前提とされていないことを予めご了承ください。
・ その他、本装置の運用結果における損害や逸失利益の請求につきましては、上記にかかわらずいかなる責任も負いかねますので予めご了承ください。

目次

1. 共通事項

1.1. モジュール

データハンドリングを実現する PD Repeater や PD Handler,PD Agent,PD Broker のアプリケーションを**モジュール**と呼びます。

1.2. 設定ファイル

1.2.1. フォーマット

各モジュールの設定ファイルは、JSON フォーマット記述さます。

1.2.2. デォルト値

各モジュールは、設定ファイルで設定可能なパラメタ（以下、**キー**と称します。）について、特に指定が無い限りデフォルト値を持ちます。

1.3. モジュール間通信

1.3.1. モジュール間通信メソッド

各モジュール間のデータ転送には、Unix ドメインソケットを用います。

1.3.2. ソケットのパス名

データを受け取るソケットのパス名はbindキーで、データの送り先ソケットのパス名はpush_toキーで変更できます。
ソケットのパス名の先頭が@の場合は、Abstract name spaceと解釈されます。

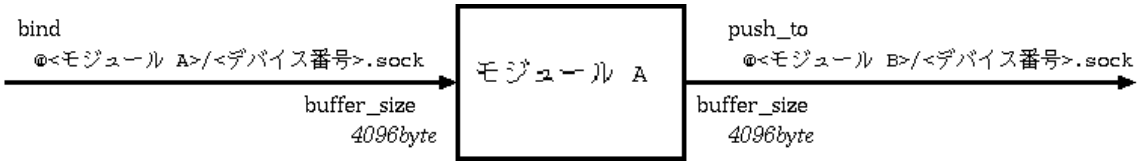
1.3.3. データーサイズ

データーのバッファサイズはデフォルト4,096 byteです。データを受け取るモジュールにおいてはbuffer size'キーで変更できます。

1.3.4. モジュール間通信に関わるモジュール共通のキー

キー	データ型	説明
bind	文字列	データを受け取るソケット名．文字列の先頭が'@' の場合はabstract namespace と解釈します． pd broker を除き空の場合は、各モジュールのデフォルト値が用いられます． @/<モジュール名＞/＜デバイス番号＞.sock の形式で指定します．
push_to	文字列	データの送り先ソケット名．文字列の先頭が'@' の場合はabstract namespace と解釈します． pd broker を除き空の場合は、各モジュールのデフォルト値が用いられます． @/<モジュール名＞/＜デバイス番号＞.sock の形式で指定します．
buffer_size	整数値	データのバッファサイズ(byte)

モジュール間通信に関わるモジュール共通のキー



2. PD Repeater

2.1. デフォルトパス

PD Repeater に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/usr/sbin/pd_repeater	常駐実行オブジェクト（デーモン）
/usr/sbin/pd_repeater_p	実行オブジェクト（ユーティリティー）
/lib/systemd/system/pd_repeater.service	Systemd Service ファイル
/etc/init.d/pd_repeater	RC ファイル
/var/webui/config/pd_repeater.conf	設定ファイル
/var/run/pd_repeater.pid	PID ファイル
/var/webui/pd-data/pd_repeater.db	キャッシュファイル
@/pd_repeater/＜デバイス番号＞.sock	入力 UNIX ドメインソケット
@/pd_handler/＜デバイス番号＞.sock	出力 UNIX ドメインソケット

PD Repeater に関連するファイルのデォルトパス

2.2. 設定ファイルの書式

2.2.1. 構文

```
{
    "db_file" : "<キャッシュファイルパス名>",
    "max_db_size" : <キャッシュサイズ>,
    "servers": {
        "<クラウドインデックスキー>" : {
            <サーバ設定用JSON オブジェクト>
        },
        "<クラウドインデックスキー>" : {
            <サーバの設定用JSON オブジェクト>
        }
      },
      "devices": [
          {
              "localname" : "<デバイスのローカル名>",
              "bind" : "<データを受け取るソケット名>",
              "push_to" : "<データの送り先ソケット名>",
              "buffer_size" : <データのバッファサイズ>,
              "truncate" : <入力データを受け付けない時間>,
              "receive" : <下流方向制御>,
              "<クラウドインデックスキー>" : {
                  <クラウド固有のデバイス設定用JSON オブジェクト>
              },
              "<クラウドインデックスキー>" : {
                  <クラウド固有のデバイス設定用JSON オブジェクト>
              }
          }
      ]
  }
```

2.2.2. ルートオブジェクト

キー	データ型	説明
db_file	文字列	キャシュファイルのパス名. デフォルト値は '/var/webui/pd-data/pd_repeater.db'. (MAXPATHLEN)
max_db_size	整数値	キャシュファイルの最大サイズ(Mbyte). デフォルト値は 16.
servers	JSON obj	servers オブジェクト
devices	JSON 配列	devices オブジェクト

ルートオブジェクト

2.2.3. servers オブジェクト

キー	データ型	説明
pd_ex	JSON obj	PD Exchenge へデータを送るための設定オブジェクト
mqtt	JSON obj	汎用の MQTT ブローカーへデータを送るための設定オブジェクト
w4d	JSON obj	IBM Watson IoT for device へデータを送るための設定オブジェクト
kinesis	JSON obj	Amazon Kinesis へデータを送るための設定オブジェクト
pgsql	JSON obj	PostgreSQL データベースへデータを送るための設定オブジェクト
mysql	JSON obj	MySQL データベースへデータを送るための設定オブジェクト
web	JSON obj	汎用の Web サーバへデータを送るための設定オブジェクト
awsiot	JSON obj	Amazon AWS IoT へデータを送るための設定オブジェクト
eventhub	JSON obj	Microsoft Azure Event Hubs へデータを送るための設定オブジェクト
w4g	JSON obj	IBM Watson IoT for gateway へデータを送るための設定オブジェクト
iothub	JSON obj	Microsoft Azure IoT Hub へデータを送るための設定オブジェクト
t4d	JSON obj	NTT docomo Toami for docomo へデータを送るための設定オブジェクト
lsocket	JSON obj	Unix ドメインソケットへデータを送るための設定オブジェクト
kddi_std	JSON obj	KDDI IoT クラウド Standard へデータを送るための設定オブジェクト
nf_dvhub	JSON obj	NIFTY IoT Device Hub へデータを送るための設定オブジェクト
pd_web	JSON obj	Plat'Home 独自仕様の Web サーバへデータを送るための設定オブジェクト
iotcore	JSON obj	Google Cloud IoT Core へデータを送るための設定オブジェクト
ltcp	JSON obj	TCP ソケットへデータを送るための設定オブジェクト
awsiot_ws	JSON obj	Amazon AWS IoT へ MQTT over WEbsocket でデータを送るための設定オブジェクト
iothub_ws	JSON obj	Microsoft Azure IoT Hub へ MQTTover Websocket でデータを送るための設定オブジェクト

server オブジェクト

クラウドインデックスキーとして同文字列に'_0' 又は'_1', '_2', '_3'を付加した文字列(例えば'pd_ex' について、'pd_ex_0','pd_ex_1','pd_ex_2','pd_ex_3') を用いることで、同一のクラウドの異る 4 つのサーバーもしくはエンドポイントを設定するこのも可能です。

2.2.3.1. PD Excheng(pd_ex)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
url	文字列	接続先URL. デフォルト値は'http://pd.plathome.com'. (MAXPATHLEN)
poll_interval	整数値	PD Exchange からコマンド(下流方向へのメッセージ) を読み出す間隔(sec). デフォルト値は30.
deid_prefix	文字列	PD Exchange のデバイス ID プリフィック. (10byte)
secretkey	文字列	接続に用いる secretkey. (16byte)

PD Exchange (pd ex) の設定オブジェクト

Proxy を使用する場合は、環境変数 http_proxy 又は https_proxy に設定します。

2.2.3.2. 汎用 MQTT ブローカー(mqtt)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
host	文字列	接続先ホストをIP アドレス又はFQDN で指定. (MAXHOSTNAMELEN)
protocol	文字列	接続プロトコル'tcp' 又は'ssl' を指定. デフォルト値は'tcp'.
port	整数値	接続先ポート番号. デフォルト値は1883(TCP 接続時) 又は 8883(SSL 接続時).
keepaliveinterval	整数値	MQTT の Keepalive 間隔(sec). デフォルト値は 10.
cleansession	論理値	MQTT の Clean session. デフォルト値は true.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
retained	論理値	MQTT の Retained. デフォルト値は false.
rcv_qos	整数値	MQTT の受信用 QoS, 0～2 を指定. デフォルト値は 1.
client_id	文字列	MQTT 接続に用いる Client ID. (64byte)
topic_prefix	文字列	MQTT の送信トピックの先頭に付加する文字列. 送信トピックは、本文字列とデバイス設定オブジェクト'unique id' に指定される文字列の組み合わせで、*topic_prefix*/*unique_id* となる. (MAXPATHLEN)
rcv_topic_prefix	文字列	MQTT の受信トピックの先頭に付加する文字列. 受信トピックは、*rcv_topic_prefix*/# で待ち受け、デバイス設定オブジェクト'unique id' に指定される文字列の組み合わせからなる *rcv_topic_prefix*/*unique_id* との完全一致するデバイスへ受信メッセージを内部へ転送する. 完全一致しない場合は、前方一致により一致する複数のデバイスへ受信メッセージを内部へ転送する. (MAXPATHLEN)
username	文字列	接続に用いるユーザー名. (32byte)
password	文字列	パスワード認証に用いるパスワード. (128byte)
truststore	文字列	SSL 接続に用いる Root CA 証明書ファイルのパス名. (MAXPATHLEN)
keystore	文字列	SSL 接続に用いる証明書ファイルのパス名. (MAXPATHLEN)
privatekey	文字列	SSL 接続に用いる秘密鍵ファイルのパス名. (MAXPATHLEN)

汎用 MQTT ブローカー(mqtt)の設定オブジェクト

2.2.3.3. IBM Watson IoT for device (w4d)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
domain	文字列	接続先ドメイン名. デフォルト値は 'messag-ing.internetofthings.ibmcloud.com'. (MAXHOSTNAMELEN)
protocol	文字列	接続プロトコル'tcp' 又は'ssl' を指定. デフォルト値は'tcp'.
port	整数値	接続先ポート番号. デフォルト値は1883(TCP 接続時) 又は 8883(SSL 接続時).
keepaliveinterval	整数値	MQTT の Keepalive 間隔(sec). デフォルト値は 10.
cleansession	論理値	MQTT の Clean session. デフォルト値は true.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
retained	論理値	MQTT の Retained. デフォルト値は false.
rcv_qos	整数値	MQTT の受信用 QoS, 0～2 を指定. デフォルト値は 1.
org_id	文字列	組織 ID. デフォルト値は'quickstart'. (16byte)
event_id	文字列	イベントID. デフォルト値は'sample'. (128byte)
format_string	文字列	データーフォーマット. デフォルト値は'json'. (16byte)
truststore	文字列	SSL 接続に用いる Root CA 証明書ファイルのパス名. (MAXPATHLEN)

IBM Watson IoT for device (w4d) の設定オブジェクト

2.2.3.4. Amazon Kinesis (kinesis)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
domain	文字列	接続先ドメイン名. デフォルト値は'amazonaws.com'. (MAXHOSTNAMELEN)
region	文字列	接続先リージョン名. デフォルト値は'ap-northeast-1'. (64byte)
accessid	文字列	アクセスID. (128byte)
accesskey	文字列	アクセスキー. (128byte)
streamname	文字列	ストリーム名. (128byte)

Amazon Kinesis (kinesis) の設定オブジェクト

Proxy を使用する場合は、環境変数 http_proxy 又は https_proxy に設定します。

2.2.3.5. PostgreSQL (pgsql)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
hostaddr	文字列	接続先IP アドレス. (MAXHOSTNAMELEN)
protocol	文字列	接続プロトコル'tcp' 又は'ssl' を指定. デフォルト値は'tcp'.
port	整数値	接続先ポート番号. デフォルト値は5432.
sslmode	文字列	PostgreSQL の SSL モード. デフォルト値は'verify-full'. (12byte)
pg_type	整数値	postgresql/server/catalog/pg_type.h に定義されるOID. デフォルト値は25 (TEXTOID).
dbname	文字列	データベース名. デフォルト値は'pd repeater'. (32byte)
table	文字列	データを書き込むテーブル名. デフォルト値は'buf'. (32byte)
username	文字列	接続に用いるユーザー名. (32byte)
password	文字列	接続に持ちるパスワード. (32byte)
truststore	文字列	SSL 接続に用いる Root CA 証明書ファイルのパス名. (MAXPATHLEN)
keystore	文字列	SSL 接続に用いる証明書ファイルのパス名. (MAXPATHLEN)
privatekey	文字列	SSL 接続に用いる秘密鍵ファイルのパス名. (MAXPATHLEN)

PostgreSQL (pgsql) の設定オブジェクト

2.2.3.6. MySQL (mysql)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
hostaddr	文字列	接続先IP アドレス. (MAXHOSTNAMELEN)
protocol	文字列	接続プロトコル'tcp' 又は'ssl' を指定. デフォルト値は'tcp'.
port	整数値	接続先ポート番号. デフォルト値は3306.
dbname	文字列	データベース名. デフォルト値は'pd repeater'. (32byte)
table	文字列	データを書き込むテーブル名. デフォルト値は'buf'. (32byte)
username	文字列	接続に用いるユーザー名. (32byte)
password	文字列	接続に持ちるパスワード. (32byte)
truststore	文字列	SSL 接続に用いる Root CA 証明書ファイルのパス名. (MAXPATHLEN)
keystore	文字列	SSL 接続に用いる証明書ファイルのパス名. (MAXPATHLEN)
privatekey	文字列	SSL 接続に用いる秘密鍵ファイルのパス名. (MAXPATHLEN)

MySQL (mysql) の設定オブジェクト

2.2.3.7. 汎用の Web サーバ(web)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
url	文字列	接続先URL. (MAXPATHLEN)
username	文字列	BASIC 認証用ユーザー名. 空の場合、認証は行わない. (32byte)
password	文字列	BASIC 認証用パスワード. 空の場合、認証は行わない. (32byte)
max_post_msize	整数値	最大ポストサイズ(Mbyte). デフォルト値は 1.
content_type	文字列	'Content-Type' を指定する. 'text/plain' もしくは'application/json' が指定された場合はペイロードのURL セーフエンコードを行わない. デフォルト値は無指定. (64byte)

汎用の Web サーバ(web) の設定オブジェクト

Proxy を使用する場合は、環境変数 http_proxy 又は https_proxy に設定します。

2.2.3.8. Amazon AWS IoT (awsiot)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
host	文字列	接続先ホスト名. (MAXHOSTNAMELEN)
protocol	文字列	接続プロトコル'tcp' 又は'ssl' を指定. デフォルト値は'tcp'.
port	整数値	接続先ポート番号. デフォルト値は8883.
keepaliveinterval	整数値	MQTT の Keepalive 間隔(sec). デフォルト値は 10.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
retained	論理値	MQTT の Retained. デフォルト値は false.
rcv_qos	整数値	MQTT の受信用 QoS, 0～2 を指定. デフォルト値は 1.
rootCA	文字列	SSL 接続に用いる Root CA 証明書ファイルのパス名. (MAXPATHLEN)

Amazon AWS IoT (awsiot) の設定オブジェクト

2.2.3.9. Microsoft Azure Event Hubs (eventhub)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
domain	文字列	接続先ドメイン名. デフォルト値は'servicebus.windows.net'. (MAX-HOSTNAMELEN)
namespace	文字列	接続先名前空間. (64byte)
port	整数値	接続先ポート番号. デフォルト値は5671.

Microsoft Azure Event Hubs (eventhub)の設定オブジェクト

2.2.3.10. IBM Watson IoT for gateway (w4g)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
domain	文字列	接続先ドメイン名. デフォルト値は 'messag-ing.internetofthings.ibmcloud.com'. (MAXHOSTNAMELEN)
protocol	文字列	接続プロトコル'tcp' 又は'ssl' を指定. デフォルト値は'tcp'.
port	整数値	接続先ポート番号. デフォルト値は1883(TCP 接続時) 又は 8883(SSL 接続時).
keepaliveinterval	整数値	MQTT の Keepalive 間隔(sec). デフォルト値は 10.
cleansession	論理値	MQTT の Clean session. デフォルト値は true.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
retained	論理値	MQTT の Retained. デフォルト値は false.
rcv_qos	整数値	MQTT の受信用 QoS, 0～2 を指定. デフォルト値は 1.
org_id	文字列	組織 ID. デフォルト値は'quickstart'. (16byte)
gateway type	文字列	ゲートウェイタイプデフォルト値は'sample'. (20byte)
gateway id	文字列	ゲートウェイ ID. (20byte)
event_id	文字列	イベントID. デフォルト値は'sample'. (128byte)
format_string	文字列	データーフォーマット. デフォルト値は'json'. (16byte)
truststore	文字列	SSL 接続に用いる Root CA 証明書ファイルのパス名. (MAXPATHLEN)
keystore	文字列	SSL 接続に用いる証明書ファイルのパス名. (MAXPATHLEN)
privatekey	文字列	SSL 接続に用いる秘密鍵ファイルのパス名. (MAXPATHLEN)

IBM Watson IoT for gateway (w4g) の設定オブジェクト

2.2.3.11. Microsoft Azure IoT Hub(iothub)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
domain	文字列	接続先ドメイン名. デフォルト値は'azure-devices.net'. (MAXHOSTNAMELEN)
hub_name	文字列	接続先ハブ名. (64byte)
gw_host	文字列	ゲートウェイホスト名.(MAXHOSTNAMELEN)
tail_slash	論理値	MQTT トピックの末尾に'/'を付けるか否か. デフォルト値は true
keepaliveinterval	整数値	MQTT の Keepalive 間隔(sec). デフォルト値は 10.
cleansession	論理値	MQTT の Clean session. デフォルト値は true.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
retained	論理値	MQTT の Retained. デフォルト値は false.
rcv_qos	整数値	MQTT の受信用 QoS, 0～2 を指定. デフォルト値は 1.
rootCA	文字列	X.509 認証に用いる Root CA 証明書ファイルのパス名. (MAXHOSTNAMELEN)

Microsoft Azure IoT Hub(iothub)の設定オブジェクト

2.2.3.12. NTT docomo Toami for docomo (t4d)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
url	文字列	接続先URL. デフォルト値は'https://xxxx.to4do.com:443'. (MAX-PATHLEN)

NTT docomo Toami for docomo (t4d)の設定オブジェクト

2.2.3.13. UNIX ドメインソケット (lsocket)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
root_path	文字列	送り先Unix ドメインソケットのルートパス名. 文字列の先頭が'@' の場合はabstract namespace と解釈する. デフォルト値は'/tmp'. (MAX-PATHLEN)

UNIX ドメインソケット (lsocket)の設定オブジェクト

2.2.3.14. KDDI IoT クラウド Standard (kddi std)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
domain	文字列	接続先ドメイン名. デフォルト値は 'datalink.m2m-cloud-std.kddi.ne.jp'. (MAXHOSTNAMELEN)
port	整数値	接続先ポート番号. デフォルト値は443.
termid	文字列	端末ID. (16byte)
username	文字列	BASIC 認証用ユーザー名. 空の場合、認証は行わない. (32byte)
password	文字列	BASIC 認証用パスワード. 空の場合、認証は行わない. (32byte)

KDDI IoT クラウド Standard (kddi std)の設定オブジェクト

Proxy を使用する場合は、環境変数 http_proxy 又は https_proxy に設定します。

2.2.3.15. NIFTY IoT Device Hub (nf_dvhub)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
host	文字列	接続先ホスト名. デフォルト値は'iot-device.jp-east-1.mqtt.cloud.nifty.com' (MAXHOSTNAMELEN)
protocol	文字列	接続プロトコル'tcp' 又は'ssl' を指定. デフォルト値は'tcp'.
keepaliveinterval	整数値	MQTT の Keepalive 間隔(sec). デフォルト値は 10.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
rootCA	文字列	SSL 接続に用いる Root CA 証明書ファイルのパス名. (MAXPATHLEN)

NIFTY IoT Device Hub (nf_dvhub)の設定オブジェクト

NIFTY IoT Device Hub は、2018 年 11 月 30 日をもってサービスを終了しています。
また、NIFTY IoT Device Hub と DEVICEHUB(https://www.devicehub.net)は、異なるサービスであり、本設定オブジェクを用いて DEVICEHUB にデータを送ることはできません。

2.2.3.16. Plat'Home 独自仕様 Web サーバ(pd web)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
url	文字列	接続先URL. (MAXPATHLEN)
poll_interval	整数値	PD Web ではメッセージの送信と同時に下流方向へのメッセージを取得するが、poll interval には送信すべきメッセージが存在しない場合を想定し、空の接続を起こす間隔(sec) を指定する. デフォルト値は 30.
username	文字列	BASIC 認証用ユーザー名. 空の場合、認証は行わない. (32byte)
password	文字列	BASIC 認証用パスワード. 空の場合、認証は行わない. (32byte)
max_post_msize	整数値	最大ポストサイズ(Mbyte). デフォルト値は 1.

Plat'Home 独自仕様 Web サーバ(pd web)の設定オブジェクト

Proxy を使用する場合は、環境変数 http_proxy 又は https_proxy に設定します。

2.2.3.17. Google Cloud IoT Core (iotcore)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
host	文字列	接続先ホスト名(MAXHOSTNAMELEN)
port	整数値	接続先ポート番号. デフォルト値は8883.
project_id	文字列	プロジェクトID. (32bytes)
cloud_region	文字列	Cloud リージョン. (16bytes)
keepaliveinterval	整数値	MQTT の Keepalive 間隔(sec). デフォルト値は 10.
cleansession	論理値	MQTT の Clean session. デフォルト値は true.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
retained	論理値	MQTT の Retained. デフォルト値は false.
rcv_qos	整数値	MQTT の受信用 QoS, 0～2 を指定. デフォルト値は 1.
rootCA	文字列	SSL 接続に用いる Root CA 証明書ファイルのパス名. (MAXPATHLEN)

Google Cloud IoT Core (iotcore)の設定オブジェクト

2.2.3.18. TCP ソケット (ltcp)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
ip_addr	文字列	接続先IP アドレス. デフォルト値は'127.0.0.1'. (16byte)
delimiter	文字列	メッセージのセパレータコード. デフォルト値は'0x00'. 例えばCRLF コードをセパレータとする場合は'0x0d0a'. (7byte)

TCP ソケット (ltcp)の設定オブジェクト

2.2.3.19. Amazon AWS IoT [Websocket](awsiot_ws)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
host	文字列	接続先ホスト名. (MAXHOSTNAMELEN)
port	整数値	接続先ポート番号. デフォルト値は8883.
keepaliveinterval	整数値	MQTT のKeepalive 間隔(sec). デフォルト値は10.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
retained	論理値	MQTT の Retained. デフォルト値は false.
rcv_qos	整数値	MQTT の受信用 QoS, 0～2 を指定. デフォルト値は 1.
accessid	文字列	アクセス ID. (128byte)
accesskey	文字列	アクセスキー. (128byte)

Amazon AWS IoT[Websocket](awsiot_ws) の設定オブジェクト

Proxy を使用する場合は、環境変数 http_proxy 又は https_proxy に設定します。

2.2.3.20. Microsoft Azure IoT Hub[Websocket] (iothub_ws)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
interval	整数値	サーバへの送信間隔(sec). デフォルト値は 30.
expier	整数値	キャッシュのデータ保持時間(sec). デフォルト値0 の場合はデータが送信さるまで無制限.
restart	整数値	送信プロセスの再起動間隔(sec). デフォルト値は604800、0 の場合は再起動しない.
domain	文字列	接続先ドメイン名. デフォルト値は'azure-devices.net'. (MAXHOSTNAMELEN)
hub_name	文字列	接続先ハブ名. (64byte)
keepaliveinterval	整数値	MQTT のKeepalive 間隔(sec). デフォルト値は10.
cleansession	論理値	MQTT の Clean session. デフォルト値は true.
reliable	論理値	MQTT の Reliable. デフォルト値は true.
qos	整数値	MQTT の送信用 QoS, 0～2 を指定. デフォルト値は 0.
retained	論理値	MQTT の Retained. デフォルト値は false.
rcv_qos	整数値	MQTT の受信用 QoS, 0～2 を指定. デフォルト値は 1.

Microsoft Azure IoT Hub[Websocket](iothub_ws)の設定オブジェクト

Proxy を使用する場合は、環境変数 http_proxy 又は https_proxy に設定します。

2.2.4. device オブジェクト

キー	データ型	説明
localname	文字列	デバイスのローカル名(デバイス番号). (32byte)
bind	文字列	データを受け取るソケット名. 文字列の先頭が'@' の場合はabstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_repeater/＜デバイス番号＞.sock が設定されます.
push_to	文字列	制御メッセージの送り先ソケット名. 文字列の先頭が'@' の場合は abstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_handler/＜デバイス番号＞.sock が設定されます.
buffer_size	整数値	データのバッファサイズ(byte). デフォルト値は4096
truncate	実数値	データを間引くための入力データを受け付けない時間(msec). デフォルト値0 の場合は間引かない.
receive	論理値	クラウドからメッセージを受け取る下流方向制御) か否か. デフォルト値はfalse
pd_ex	JSON obj	PD Exchenge 固有の設定オブジェクト
mqtt	JSON obj	汎用の MQTT ブローカー固有の設定オブジェクト
w4d	JSON obj	IBM Watson IoT for device 固有の設定オブジェクト
kinesis	JSON obj	Amazon Kinesis 固有の設定オブジェクト
pgsql	JSON obj	PostgreSQL データベース固有の設定オブジェクト
mysql	JSON obj	MySQL データベース固有の設定オブジェクト
web	JSON obj	汎用の Web サーバ固有の設定オブジェクト
awsiot	JSON obj	Amazon AWS IoT 固有の設定オブジェクト
eventhub	JSON obj	Microsoft Azure Event Hubs 固有の設定オブジェクト
w4g	JSON obj	IBM Watson IoT for gateway 固有の設定オブジェクト
iothub	JSON obj	Microsoft Azure IoT Hub 固有の設定オブジェクト
t4d	JSON obj	NTT docomo Toami for docomo 固有の設定オブジェクト
lsocket	JSON obj	Unix ドメインソケットへデータを送るための固有の設定オブジェクト
kddi_std	JSON obj	KDDI IoT クラウド Standard 固有の設定オブジェクト
nf_dvhub	JSON obj	NIFTY IoT Device Hub 固有の設定オブジェクト
pd_web	JSON obj	Plat'Home 独自仕様の Web サーバ固有の設定オブジェクト
iotcore	JSON obj	Google Cloud IoT Core 固有の設定オブジェクト
ltcp	JSON obj	TCP ソケット固有の設定オブジェクト
awsiot_ws	JSON obj	Amazon AWS IoT へ MQTT over Websocket でデータを送るための固有の設定オブジェクト
iothub_ws	JSON obj	Microsoft Azure IoT Hub へ MQTT over Websocket でデータを送るための固有の設定オブジェクト

devices オブジェクト

クラウドインデックスキーについては servers オブジェクトと同様に同文字列に'_0' 又は '_1', '_2', '_3' を付加した文字列(例えば'pd_ex' について、'pd_ex_0','pd_ex_1','pd_ex_2', 'pd_ex_3) を用いることが可能です。

2.2.4.1. PD Exchenge(pd_ex)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
deid suffix	文字列	PD Exchange のデバイス ID サフックス. (9byte)

PD Exchenge(pd_ex)固有の設定オブジェクト

2.2.4.2. 汎用の MQTT ブローカー(mqtt)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
unique_id	文字列	ユニーク ID. (20byte)

汎用の MQTT ブローカー(mqtt)固有の設定オブジェクト

2.2.4.3. IBM Watson IoT for device (w4d)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
device_id	文字列	デバイス ID. (20byte)
device_type	文字列	デバイスタイプ. (20byte)
password	文字列	パスワード. (128byte)
keystore	文字列	SSL 接続に用いる証明書ファイルのパス名. (MAXPATHLEN)
privatekey	文字列	SSL 接続に用いる秘密鍵ファイルのパス名. (MAXPATHLEN)

IBM Watson IoT for device (w4d)固有の設定オブジェクト

2.2.4.4. Amazon Kinesis (kinesis)

キー	データ型	説明
enable	論理値	デフォルト値はfalse

Amazon Kinesis (kinesis)固有の設定オブジェクト

2.2.4.5. PostgreSQL (pgsql)

キー	データ型	説明
enable	論理値	デフォルト値はfalse

PostgreSQL (pgsql)固有の設定オブジェクト

2.2.4.6. MySQL (mysql)

キー	データ型	説明
enable	論理値	デフォルト値はfalse

MySQL (mysql)固有の設定オブジェクト

2.2.4.7. 汎用の Web サーバ(web)

キー	データ型	説明
enable	論理値	デフォルト値はfalse

汎用の Web サーバ(web)固有の設定オブジェクト

2.2.4.8. Amazon AWS IoT (awsiot)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
client_id	文字列	クライアント ID. (20byte)
thing_name	文字列	Device Shadows に用いる thingName. (128byte)
topic	文字列	MQTT の送信トピック. (MAXPATHLEN)
rcv_topic	文字列	MQTT の受信トピック. (MAXPATHLEN)
cert	文字列	SSL 接続に用いる証明書ファイルのパス名. (MAXPATHLEN)
privatekey	文字列	SSL 接続に用いる秘密鍵ファイルのパス名. (MAXPATHLEN)

Amazon AWS IoT (awsiot)固有の設定オブジェクト

2.2.4.9. Microsoft Azure Event Hubs (eventhub)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
hub_name	文字列	Event Hubs 名. (64byte)
sas_policy	文字列	SAS ポリシー. (64byte)
sas_key	文字列	SAS キー. (64byte)

Microsoft Azure Event Hubs (eventhub)固有の設定オブジェクト

2.2.4.10. IBM Watson IoT for gateway (w4g)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
device_id	文字列	デバイス ID. (20byte)
device_type	文字列	デバイスタイプ. (20byte)

IBM Watson IoT for gateway (w4g)固有の設定オブジェクト

2.2.4.11. Microsoft Azure IoT Hub(iothub)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
device_id	文字列	デバイス ID. (128byte)
device_key	文字列	デバイスキー. (64byte)
transparent	論理値	IoT Edge を透過モードで動作させるか否か. デフォルト値は false
module_id	文字列	モジュール ID. (64byte). 透過モード時は無視.
cert	文字列	X.509 認証に用いる証明書ファイルのパス名. (MAXPATHLEN)
privatekey	文字列	X.509 認証に用いる秘密鍵ファイルのパス名. (MAXPATHLEN)

Microsoft Azure IoT Hub(iothub)固有の設定オブジェクト

2.2.4.12. NTT docomo Toami for docomo (t4d)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
gwname	文字列	GatewayName. (32byte)
appkey	文字列	AppKey. (64byte)

NTT docomo Toami for docomo (t4d)固有の設定オブジェクト

2.2.4.13. UNIX ドメインソケット (lsocket)

キー	データ型	説明
enable	論理値	デフォルト値はfalse

UNIX ドメインソケット (lsocket)固有の設定オブジェクト

2.2.4.14. KDDI IoT クラウド Standard (kddi std)

キー	データ型	説明
enable	論理値	デフォルト値はfalse

KDDI IoT クラウド Standard (kddi std)固有の設定オブジェクト

2.2.4.15. NIFTY IoT Device Hub (nf_dvhub)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
event_type	文字列	イベントタイプ. (20byte)
device_id	文字列	デバイス ID. (20byte)
device_api_key	文字列	デバイス API キー(169byte)

NIFTY IoT Device Hub (nf_dvhub)固有の設定オブジェクト

NIFTY IoT Device Hub は、2018 年 11 月 30 日をもってサービスを終了しています。
また、NIFTY IoT Device Hub と DEVICEHUB(https://www.devicehub.net)は、異なるサービスであり、本設定オブジェクを用いて DEVICEHUB にデータを送ることはできません。

2.2.4.16. Plat'Home 独自仕様 Web サーバ(pd web)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
id	文字列	クライアント ID. (128byte)
key	文字列	トークン認証に用いる文字列(アクセスキー). (128byte)

Plat'Home 独自仕様 Web サーバ(pd web)固有の設定オブジェクト

2.2.4.17. Google Cloud IoT Core (iotcore)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
registry_id	文字列	レジストリ ID. (32byte)
device_id	文字列	デバイス ID. (32byte)
jwt_algorithm	文字列	JWT アルゴリズム. 'RS256', 'ES256' のいずれか.(6byte)
cert	文字列	SSL 接続に用いる証明書ファイルのパス名. (MAXPATHLEN)
privatekey	文字列	SSL 接続に用いる秘密鍵ファイルのパス名. (MAXPATHLEN)

Google Cloud IoT Core (iotcore)固有の設定オブジェクト

2.2.4.18. TCP ソケット (ltcp)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
port	文字列	接続先ポート番号. デフォルト値は 49152(TCP 接続時).

TCP ソケット (ltcp)固有の設定オブジェクト

2.2.4.19. Amazon AWS IoT[Websocket](awsiot_ws)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
client_id	文字列	クライアント ID. (20byte)
thing_name	文字列	Device Shadows に用いる thingName. (128byte)
topic	文字列	MQTT の送信トピック. (MAXPATHLEN)
rcv_topic	文字列	MQTT の受信トピック. (MAXPATHLEN)

Amazon AWS IoT[Websocket](awsiot_ws)固有の設定オブジェクト

2.2.4.20. Microsoft Azure IoT Hub[Websocket] (iothub_ws)

キー	データ型	説明
enable	論理値	デフォルト値はfalse
device_id	文字列	デバイス ID. (128byte)
device_key	文字列	デバイスキー. (64byte)

Microsoft Azure IoT Hub[Websocket](iothub_ws)固有の設定オブジェクト

2.3. 下流方向メッセージ

PD Repeater が下流のモジュールに転送するメッセージのフォーマットを示します。

Cloud ID (1Byte)	Sub ID (1Byte)	Header Size (2Bytes)	MD5 (16Bytes)	Header	Payload

フィールド	バイト数	説明
Cloud ID	1	便宜上割り当てている送受信先プロトコル（クラウド）の番号
Sub ID	1	便宜上割り当てている送受信先のサブ番号(0x00 又は 0x01)
Header Size	2	PD Reperater で付加された Header のサイズ
MD5	16	クラウドから送られて来た制御メッセージのハッシュ値(MD5)
Header	可変	PD Reperater で付加された MQTT のトピックや HTTP の応答ヘッダ
Payload	可変	クラウドから送られて来た制御メッセージ

下流方向メッセージのフォーマット

Cloud ID とヘッダーの内容を示します。

クラウド	Cloud ID	下流方向制御	ヘッダーの内容
PD Exchange	0x00	対応	Command ID, ApplicationID
MQTT サーバー	0x01	対応	MQTT 受信トピック
Watson IoT for Device	0x02	対応	MQTT 受信トピック
Amazon Kinesis	0x03		
PostgreSQL サーバ	0x04		
MySQL サーバ	0x05		
WEB サーバー	0x06		
AWS IoT	0x07	対応	MQTT 受信トピック
MS Azure Event hubs	0x08		
Watson IoT for Gateway	0x09	対応	MQTT 受信トピック
MS Azure IoT Hub	0x0a	対応	MQTT 受信トピック
Toami for DOCOMO	0x0b		
ドメインソケット	0x0c		
KDDI IoT クラウド Standard	0x0d		
IoT デバイスハブ	0x0e	対応	MQTT 受信トピック
Plat'Home 独自仕様 WEB サーバ	0x0f	対応	HTTP 応答ヘッダー
Google IoT Core	0x10	対応	MQTT 受信トピック
TCP	0x11	対応	接続先 IP アドレスとポート番号
AWS IoT [Websocket]	0x12	対応	MQTT 受信トピック
MS Azure IoT Hub[Websocket]	0x13	対応	MQTT 受信トピック

クラウド ID とヘッダの内容

Watson IoT については、ペイロードを"d"キーの JSON オブジェクトとしているため、ハッシュ値（MD5）を得た後、デコードし"d"キーの JSON オブジェクトのみを制御メッセージとして出力しています。

IoT デバイスハブについては、ペイロードを"parameters"キーの JSON オブジェクトとしているため、ハッシュ値（MD5）を得た後、デコードし"parameters"キーの JSON オブジェクトのみを制御メッセージとして出力しています。

TCP の接続先 IP アドレスとポート番号は、次の JSON 文字列で付加されます。

{"ip_addr":"<*IP address*>","port":<*port*>}

2.4. Plat'Home 独自仕様 WEB サーバ

2.4.1. HTTP ヘッダー

HTTP ヘッダー	内容
X-Pd-Web-Version	PD Web の仕様のバージョン番号
X-Pd-Web-Id	クライアントの ID
X-Pd-Web-Time	RFC3339 準拠のタイムスタンプ
X-Pd-Web-Md5	ペイロードのハッシュ値
X-Pd-Web-Signature	ヘッダ情報と鍵(Key)から作成されたハッシュ値
Content-Type	application/json;charset=UTF-8

2.4.2. トークン

PD Repeater で作成されるリクエストヘッダのトークン(X-Pd-Web-Sinature)は、Web サーバーにおいて次の PHP スクリプトにより再生できます。

```
$hash_hmac_data =
$_SERVER['HTTP_X_PD_WEB_VERSION'] .
$_SERVER['HTTP_X_PD_WEB_ID'] .
$_SERVER['HTTP_X_PD_WEB_TIME'] .
$_SERVER['HTTP_X_PD_WEB_MD5'];
$signature = hash_hmac ('sha256', $hash_hmac_data, $key, false);
```

ここで$keyは、は$_SERVER['HTTP X PD WEB ID'] と対をなす予めWebサーバーに保存された鍵(Key)となります。再生した$signature と$_SERVER['HTTP X PD WEB SIGNATURE'] を比較することで認証します。

応答ヘッダーのトークン(X-Pd-Web-Sinature)は、Web サーバーにおいて次の PHP スクリプトにより作成します。

```
$tm = localtime();
$timestamp = sprintf("%04d-%02d-%02dT%02d:%02d:%02d.000+09:00",
$tm[5]+1900,$tm[4]+1,$tm[3],$tm[2],$tm[1],$tm[0]);
$hash_hmac_data = '1.0' . $_SERVER['HTTP_X_PD_WEB_ID'] . $timestamp .
md5($payload) . $_SERVER['HTTP_X_PD_WEB_SIGNATURE'];
$signature = hash_hmac ('sha256', $hash_hmac_data, $key, false);
```

ここで、$payload は、ペイロードの文字列、送信すべき文字列（下流方向の制御メッセージ）が無い場合は、$payload="";とします。

リクエストヘッダのトークンとは異なり、被署名文字列に PD Repeater から送られて来たリクエストヘッダのトークン($_SERVER['HTTP X PD WEB SIGNATURE'])が、含まれる点に注意して下さい。

3. PD Broker

3.1. デフォルトパス

PD Broker に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/usr/sbin/pd_broker	常駐実行オブジェクト（デーモン）
/lib/systemd/system/pd_broker.service	Systemd Service ファイル
/etc/init.d/pd_broker	RC ファイル
/var/webui/config/pd_broker.conf	設定ファイル
/var/run/pd_broker.pid	PID ファイル

PD Broker に関連するファイルのデォルトパス

3.2. 設定ファイルの書式

3.2.1. 構文

```
{
    "brokers" : [
        {
            "enable" : <bool値>,
            "bind": "<受信ソケット名>",
            "buffer_size" : <バッファサイズ>,
            "bind": "<受信ソケット名>",
            "destinations":[
                “<送信先ソケット名>”,
                “<送信先ソケット名>”
            ]
        },
        {
            "enable" : <bool値>,
            "bind": "<受信ソケット名>",
            "buffer_size" : <バッファサイズ>,
            "destinations":[
                “<送信先ソケット名>”,
                “<送信先ソケット名>”
            ]

        }
    ]
}
```

3.2.2. brokers オブジェクト

キー	データ型	説明
enable	論理値	デフォルト値はfalse
bind	文字列	データを受け取るソケット名．文字列の先頭が'@' の場合はabstract namespace と解釈します． デフォルト値は定義されていません。@/<モジュール名>/<デバイス番号>.sock の形式で指定します.
buffer_size	整数値	データのバッファサイズ(byte)
destinations	文字列配列	データの送り先ソケット名．文字列の先頭が'@' の場合はabstract names-pace と解釈する．'@/*module name*/*localname*.sock' の形式で指定する.(MAXPATHLEN) デフォルト値は定義されていない．最大32個まで指定可能

brokers オブジェクト

4. PD Agent

4.1. デフォルトパス

PD Agent に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/usr/sbin/pd_agent	常駐実行オブジェクト（デーモン）
/lib/systemd/system/pd_agent.service	Systemd Service ファイル
/etc/init.d/pd_agent	RC ファイル
/var/webui/config/pd_agent.conf	設定ファイル
/var/run/pd_agent.pid	PID ファイル

PD Agent に関連するファイルのデォルトパス

4.2. 設定ファイルの書式

4.2.1. 構文

```
{
    "agents" : [
        {
            "enable" : <bool値>,
            “localname:”<デバイス番号>”,
            "bind": "<受信ソケット名>",
            “push_to”: “<送信先ソケット名>”,
            "buffer_size" : <バッファサイズ>,
            “channels”:[
                {
                    <channelsオブジェクト>
                },
                {
                    <channelsオブジェクト>
                }
            ]
        },
        {
            "enable" : <bool値>,
            “localname:”<デバイス番号>”,
            "bind": "<受信ソケット名>",
            “push_to”: “<送信先ソケット名>”,
            "buffer_size" : <バッファサイズ>,
            “channels”:[
                {
                    <channelsオブジェクト>
                }
            ]
        }
    ]
}
```

4.2.2. agents オブジェクト

キー	データ型	説明
enable	論理値	デフォルト値はfalse
localname	文字列	デバイスのローカル名(デバイス番号). (32byte)
bind	文字列	データを受け取るソケット名. 文字列の先頭が'@' の場合はabstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_handler/<デバイス番号>.sock が設定されます.
push_to	文字列	制御メッセージの送り先ソケット名. 文字列の先頭が'@' の場合は abstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_repeater/<デバイス番号>.sock が設定されます.
buffer_size	整数値	データのバッファサイズ(byte). デフォルト値は4096
channels	JSON 配列	channels オブジェクト. 最大32 個まで設定可能.

agents オブジェクト

4.2.3. channels オブジェクト

キー	データ型	説明
name	文字列	チャンネルの名称. (32byte)
reply	論理値	'push to' キーに設定されるソケットに応答を返すか否かの設定. デフォルト値はtrue
dynamic_link	論理値	'exec' キーに指定されるパス名を Dynamic Link モジュールとして動作します. デフォルト値は false.
index	文字列	'exec' キーに指定される実行オブェクトの起動条件として評価される、JSON 文字列データのキー(32byte)
value	文字列	'exec' キーに指定される実行オブェクトの起動条件として評価される、JSON 文字列データの値(32byte)
exec	文字列	'index' キーに設定されるキーと'value' キーに設定されるその値が、JSON文字列データに含まれている場合に実行される実行オブェクトもしくは Dynamic Link モジュールのパス名.(MAXPATHLEN)
args	文字列	'exec' キーに設定される実行オブェクトに与える引数. 'dynamic_link' キーが true の場合は無視されます.(1024byte)

channels オブジェクト

4.3. 実行オブジェクトに継承される環境変数

受信データのJSON 文字列の内、起動条件の評価に用いたキーと値を含め、値が文字列か数値の場合は、これを環境変数に設定し実行オブジェクトに継承します。
また、pd repeater から渡されたCloud ID、受信データ(ペイロード) のMD5 ハッシュ値、受信データのヘッダ情報とpd agent に設定されているデバイスのローカル名も合わせて環境変数に継承します。

環境変数名	データ型	説明
request_cloud_id	整数値	PD Repeater が便宜上割り当てている番号
request_sub_id	整数値	同一クラウドの複数サーバを利用している場合の識別子
request_header	文字列	PD Repeater から渡される受信データのヘッダ情報
request_payload	文字列	PD Repeater から渡される受信データ(ペイロード)
request_md5	文字列	PD Repeater から渡される受信データ(ペイロード) のMD5 ハッシュ値
agent_localname	文字列	'localname' キーに設定されているデバイスのローカル名(デバイス番号).
agent_bind	文字列	'bind' キーに設定されているデータを受け取ったソケット名.
agent_push_to	文字列	'push to' キーに設定されている応答先ソケット名.
agent_buffer_size	整数値	'buffer size' キーに設定されているデータのバッファサイズ.(byte)

実行オブジェクトに環境変数として継承されるペイロード以外のパラメータ

4.4. Dynamic Link モジュール

Dynamic Link モジュールのサンプルコードを示します。
Dynamic Link モジュールは、dlopen()によって PD Agent に取り込まれます。dl_exec() が、execv() によって実行されう外部の実行オブジェクトの代わりに呼び出される関数です。dl_exec() の引数には、実行オブジェクトを実行する際に環境変数として継承される全てのパラメータを含みます。 文字列ポインタ result が NULL の場合、応答メッセージの‘result’キーの値は ‘done’となります。 dl_init() は PD Agent の起動時に、dl_fini() は停止時に一度だけ呼ばれる関数であり、必要が無ければサンプルコードのままとして下さい。

```
/*
 *   Copyright (c) 2018
 *          Plat'Home CO., LTD. <support@plathome.co.jp>. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *      notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *      notice, this list of conditions and the following disclaimer in the
 *      documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the Plat'Home CO., LTD. nor the names of
 *      its contributors may be used to endorse or promote products derived
 *      from this software without specific prior written permission.
```

```
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.    IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <syslog.h>
#include <time.h>
#include <syslog.h>
#include <time.h>
#include <unistd.h>
#include <errno.h>
#include <sys/param.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <jansson.h>

#define N_CHANNEL           32

struct channel_t {
        unsigned char    reply;
        unsigned char    dynamic_link;
        char             name[32];              /* Name of chaanel                    */
        char             index[32];             /* Index of JSON string to be driven. */
        char             value[32];             /* Value of JSON string to be driven. */
        char             exec[MAXPATHLEN];      /* Pathname of dynamic link module.   */
        int              argc;                  /* Not use in dynamic link module.    */
        char             *argv[16];             /* Not use in dynamic link module.    */
        void             *dl_handle;            /* Handle on the global symbol object.*/
};

struct agent_t {
        int              num_ch;                /* Total number of channels in device.  */
        unsigned char    enable;
        char             localname[32];         /* Local Device name (number).          */
        char             bind[MAXPATHLEN];      /* Pathname of UNIX domain socket to listen.  */
        size_t           buffer_size;           /* Buffer size of UNIX domain socket.         */
        char             push_to[MAXPATHLEN];   /* Pathname of UNIX domain socket to send. */
        struct channel_t *ch[N_CHANNEL];        /* Pointer to each channel.             */
        char             *buf;                  /* message buffer for down stream.      */
};

// Your own functions

/*
        Your own initialize function : void dl_init()
          It is called only once at startup.
          Return value : none.
```

```
*/
void dl_init()
{
        return;
}

/*
        Your own finalize function : void dl_fini()
          It is called only once at termination.
          Return value : none.
*/
void dl_fini()
{
        return;
}

/*
        Your own exec function : int dl_exec()
            char *result                : Reply value for 'result' key of JSON.
            struct agent_t *agent       : See above structure declaration.
            int ch_number               : Channel number where Index and Value match.
            json_t *json_obj            : It stores all indexes and values contained in the payload of
                                          the downstream message as Jansson C objects.
            unsigned char cloud_id      : Cloud Id.
            unsigned char sub_id        : Sub Cloud Id.
                                          These are the indications from which cloud they have
                                          been sent.
            char *rcv_header            : Contain the header information of the down stream
                                          message.
            char *rcv_payload           : Contain the paylod of the down stream message.
            char *hash                  : Hash value (MD5) of paylod of the down stream message.
          Return value : must be 0 for normal, -1 for error.
*/
int dl_exec(char *result, struct agent_t *agent, int ch_number, json_t *json_obj,
        unsigned char cloud_id, unsigned char sub_id,
        char *rcv_header, char *rcv_payload, char *hash)
{
        int rc;
        char *tx_payload;
        struct channel_t *ch;

        ch = (struct channel_t *)agent->ch[ch_number];

        pid_t pid;

//      The method to obtain Index and value from json_obj is shown below.
/*
        json_t *val;
        const char *key;
        val = json_object();
        json_object_foreach(json_obj, key, val) {
                if(json_is_string(val)) {
                        printf("%s", json_string_value(val));
                }
                else if(json_is_integer(val)) {
                        printf("%ld", json_integer_value(val));
                }
                else if(json_is_real(val)) {
                        printf("%lf", json_real_value(val));
                }
```

```
        }
*/

//      Write your own process here.

        snprintf(result, sizeof(char)*agent->buffer_size,
                "{YOUR OWN JSON OBJECT}");

        return(0);
}
```

4.5.応答メッセージ

'reply' キーがtrue の場合、pd agent は'push to' に設定されるソケットに実行ステータスをJSON文字列で返します。

ステータス	応答メッセージ
実行オブジェクトを起動した	{"time":"<timestamp>","repl_ to":"<md5>","result":"queuing", "reason":"matched","matched":{"<key>":"<value>"}}
実行オブジェクトの実行に失敗した	{"time":"<timestamp>","repl_ to":"<md5>","result":"faild", "reason":"matched","matched":{"<key>":"<value>"}}
実行オブジェクトの実行を終えた	{"time":"<timestamp>","repl_ to":"<md5>","result":"done", "reason":"matched","matched":{"<key>":"<value>"}}
一致するキーと値が存在しない	{"time":"<timestamp>","repl_ to":"<md5>","result":"not queuing", "reason":"key and value notmatched"}
受信データが JSON フォーマットでない	{"time":"<timestamp>","repl_ to":"<md5>","result":"not queuing", "reason":"not JSON form"}

PD Agentの応答メッセージ

ここで、<md5> はpd repeater から渡された受信データ(ペイロード) のMD5 ハッシュ値、'"<key>":"<value>"'は、一致した'index' キーと'value' キーに設定された条件です。
実行ステータスは実行オブジェクトを呼び出すexecv() の戻り値であり、実行オブジェクトの処理の結果を示すものではないため. システムに完全性を求めるのであれば実行オブジェクトで所定のUnixドメインソケットに応答を返すようにして下さい。
Dynamic Link モジュール使用時で、dl_exec() の文字列ポインタ引数 result が NULLで無い場合、実行終了時の'result' キーの値は文字列ポインタ引数resultの値(JSON 文字列) に置き換えられます。

5. PD Handler Modbus

5.1. Modbus ファンクションコード

コード	名称	機能説明
0x01	Read Coils	デジタル出力に設定されているビット値を読み込む
0x02	Read Discrete Input	デジタル入力のビット値を読み込む.
0x03	Read Holding Registers	レジスタ出力に設定されている値を読み込む.
0x04	Read Input Registers	レジスタ入力の値を読み込む.
0x05	Write Single Coil	デジタル出力 1 bits のビット値を設定する.
0x06	Write Single Register	レジスタ出力 1 レジスタの値を設定する.
0x07	Read Exception Status	Modbus サーバ/クライアント間でエラーステータスを通知する.
0x09	Write Single Discrete Input	デジタル入力 1 bits のビット値を設定する.
0x0a	Write Single Input Register	レジスタ入力 1 レジスタの値を設定する.
0x0f	Write Multiple Coils	連続する複数のデジタル出力のビット値を設定する.
0x10	Write Multiple Registers	連続する複数のレジスタ出力の値を設定する.
0x11	Report Slave ID	接続可能なスレーブ(サーバ) 機器のユニット ID の一覧.
0x13	Write Multiple Discrete Input	連続する複数のデジタル入力のビット値を設定する.
0x14	Write Multiple Input Registers	連続する複数のレジスタ入力の値を設定する.
0x16	Mask Write Registers	レジスタ出力をマスクする.
0x17	Write And Read Registers	連続する複数のレジスタ出力の値を設定し、その値を読み込む.

PD Handler Modbus に用いられる Modbus ファンクションコード

0x09, 0x0a, 0x13, 0x14 は、物理的入力を持たないpd handler modbus server のデジタル入力もしくはレジスタ入力をクラウド側から設定できるよう用意された、本来のModbus プロトコルとは異なる独自仕様です。

Modbus ファンクションコードは、設定ファイルもしくはCSV ファイルあるいはクラウド側からの制御においてJSON 文字列の'function' キーの値に設定される. '0x' から始まる 16 進表記のファンクションコードを文字列として'function' キーの値に設定することも可能ですが、16 進表記とは別に整数表記と文字列表記を用いることも可能です。

コード	名称	整数表記	文字列表記
0x01	Read Coils	1	read_coils
0x02	Read Discrete Input	2	read_discrete_input
0x03	Read Holding Registers	3	read_holding_registers
0x04	Read Input Registers	4	read_input_registers
0x05	Write Single Coil	5	write_single_coil
0x06	Write Single Register	6	write_single_register
0x07	Read Exception Status	7	read_exception_status
0x09	Write Single Discrete Input	9	write_single_discrete_input
0x0a	Write Single Input Register	10	write_single_input_register
0x0f	Write Multiple Coils	15	write_multiple_coils
0x10	Write Multiple Registers	16	write_multiple_registers
0x11	Report Slave ID	17	report_slave_id
0x13	Write Multiple Discrete Input	19	write_multiple_discrete_input
0x14	Write Multiple Input Registers	20	write_multiple_iput_registers
0x16	Mask Write Registers	22	mark_write_registers
0x17	Write And Read Registers	23	write_and_read_registers

function キーに用いるファンクションコードの別称

5.2. PD Handler Modbus Client

5.2.1. デフォルトパス

PD Handler Modbus Client に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/usr/sbin/pd_handler_modbus_client	常駐実行オブジェクト（デーモン）
/lib/systemd/system/pd_handler_modbus_client.service	Systemd Service ファイル
/etc/init.d/pd_handler_modbus_client	RC ファイル
/var/webui/config/pd_handler_modbus_client.conf	設定ファイル
/var/webui/upload_dir/ pd_handler_modbus_client.csv	CSV ファイル
/var/run/pd_handler_modbus_client.pid	PID ファイル

PD Handler Modbus Client に関連するファイルのデォルトパス

5.2.2. 設定ファイルの書式

5.2.2.1. 構文

```
{
    "csv_file" : “<CSVファイルのパス名>”,
    “clients”: [
        {
            <clientsオブジェクト>,
            “acquisitions”:[
                {
                    < acquisitionsオブジェクト>
                },
                {
                    < acquisitionsオブジェクト>
                }
            ]
        },
        {
            <clientsオブジェクト>,
            “acquisitions”:[
                {
                    < acquisitionsオブジェクト>
                },
                {
                    < acquisitionsオブジェクト>
                }
            ]
        }
    ]
}
```

5.2.2.2. ルートオブジェクト

キー	データ型	説明
csv_file	文字列	CSVファイルのパス名. デフォルト値は '/var/webui/upload_dir/pd-data/pd_handler_modbus_client.csv'. (MAXPATHLEN)
clients	JSON obj	clients オブジェクト

ルートオブジェクト

5.2.2.3. clients オブジェクト

clients オブジェクトの配列数は、最大256 個です。Modbus のプロトコル(TCP, RTC) に依存するオブジェクトとプロトコルに依存しない共通のオブジェクトがあります。

キー	対象	データ型	説明
enable	共通	論理値	デフォルト値はfalse
localname		文字列	デバイスのローカル名(デバイス番号). (32byte)
memo		文字列	出力に付加されるユーザー定義文字列. (256byte)
bind		文字列	データを受け取るソケット名. 文字列の先頭が'@' の場合はabstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_handler/<デバイス番号>.sock が設定されます.
push_to		文字列	制御メッセージの送り先ソケット名. 文字列の先頭が'@' の場合は abstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_repeater/<デバイス番号>.sock が設定されます.
buffer_size		整数値	データのバッファサイズ(byte). デフォルト値は4096
receive		論理値	クラウドからメッセージを受け取る下流方向制御) か否か. デフォルト値はfalse
protocol		文字列	接続プロトコル'tcp' 又は'rtu' を指定. デフォルト値は'tcp'.
writeout		論理値	push to に指定するソケットに出力するか否か. デフォルト値は true.
interval		整数値	データを取得する間隔(sec). デフォルト値は 60.
timeout		整数値	データを取得出来ない場合のタイムアウト(msec). デフォルト値は 5000.
time_sysnc		論理値	時刻同期モード. デフォルト値は false
base_time		文字列	時刻同期モードの基準時刻を'HH:MM' 形式で指定する. デフォルト値は'00:00'
node	TCP	文字列	TCP 接続 PLC 機器の IP アドレス. デフォルト値は'127.0.0.1'
port		整数値	TCP 接続 PLC 機器のポート番号. デフォルト値は 502
device	RTU	文字列	シリアル接続PLC 機器のデバイス名. デフォルト値は'/dev/tty00'
rtu_speed		整数値	シリアル接続 PLC 機器のビットレート. デフォルト値は 115200
rtu_bits		整数値	シリアル接続 PLC 機器のビット数. 8 又は 7. デフォルト値は 8.
rtu_parity		文字列	シリアル接続PLC 機器のパリティ. 'none','even','odd' のいずれか. デフォルト値は'none'.
rtu_stop		整数値	シリアル接続PLC 機器のストップビット. 1 又は2. デフォルト値は1.
acquisitions	共通	JSON 配列	acquisitions オブジェクト

clients オブジェクト

5.2.2.4. acquisitions オブジェクト

キー	データ型	説明
unit	整数	PLC 機器のModbus ユニットID. 1 ～ 247 又は255 (TCP プロトコルのみ)
function	文字列又は整数	Modbus プロトコルのデータ読み出しファンクション名又はファンクション番号. 文字列の場合は、'read_coils', 'read_discrete input', 'read_holding_registers','read_input_registers' 又は'0x01'～'0x04'、整数値の場合は文字列の順に1～4. デフォルト値は 'read_holding_r_gisters'.
data_type	文字列又は整数	文字列の場合は'uint16_t','int16_t','uint32lsb_t','int32lsb_t', 'uint32msb_t','int32msb_t' のいずれか整数値の場合は文字列の順に0～5. function が'read_coils' 又は'read discrete_input'の場合は、'uint16_t' に固定される. デフォルト値は'uint16 t'.
address	文字列又は整数	読み込み介しレジスタアドレスを指定する. 文字列の先頭に'0x'が付加されている場合は16 進表記と解釈される. デフォルト値は'0x0'.
number	文字列又は整数	読み込むビット数またはレジスタ数、文字列の先頭に'0x' が付加されている場合は16 進表記と解釈される. data_type が'uint32lsb_t', 'int32ls_ t','uint32msb_t','int32msb_t' の場合は、内部的に2 倍の値で扱われる. デフォルト値は1.

acquisitions オブジェクト

5.2.2.5. 利用可能なファンクションコード

PD Handler Modbus Client で利用可能なファンクションコードを示します。

コード	名称	ローカル	クラウド
0x01	Read Coils	○	○
0x02	Read Discrete Input	○	○
0x03	Read Holding Registers	○	○
0x04	Read Input Registers	○	○
0x05	Write Single Coil		○
0x06	Write Single Register		○
0x07	Read Exception Status		
0x09	Write Single Discrete Input		
0x0a	Write Single Input Register		
0x0f	Write Multiple Coils		○
0x10	Write Multiple Registers		○
0x11	Report Slave ID		○
0x13	Write Multiple Discrete Input		
0x14	Write Multiple Input Registers		
0x16	Mask Write Registers		
0x17	Write And Read Registers		○

PD Handler Modbus Client で利用可能なファンクションコード

ローカルは、設定ファイルもしくはCSV ファイルに指定可能なファンクションコード、クラウドはクラウド側か指定可能なファンクションコードを意図します。

5.2.2.6. CSV ファイル

/var/webui/config/pd_handler_modbus_client.conf の csv_file キーに設定される CSV ファイルを置くことで、/var/webui/config/pd_handler_modbus_client.conf で設定された１つのデバイス番号に対し複数の取得 Modbus クライアントデバイスを設定することができます。
CSV ファイルの書式は、次の通りです。

デバイス番号，ユニット ID，読込方式，データタイプ，読込開始アドレス，読込レジスタ数

パラメータ	データの型式	説明
デバイス番号	半角英数字	clients オブジェクトの localname キーに設定されたデバイス番号を記載します。　設定されていないデバイス番号は無視されます。 先頭が'#'または"/"の場合は、コメント行として扱われます。
ユニット ID	半角数字	PLC 機器の Modbus ユニット ID を設定します。ユニット ID は、1～247 または 255 を記載します。
読込方式	半角英数字	読込方式として次の何れかを記載します。 デジタル出力：'read_coils'　又は '0x01' 又は '1' デジタル入力：'read_discrete_input' 又は '0x02' 又は '2' レジスタ出力：'read_holding_registers' 又は '0x03' 又は '3' レジスタ入力：'read_input_registers' 又は'0x04' 又は '4'
データタイプ	半角英数字	データタイプとして次の何れかを記載します。 符号なし 16 ビット整数：　'uint16_t' 又は '0' 符号付き 16 ビット整数：　'int16_t' 又は '1' 符号なし 32 ビット整数/リトルエンディアン：'uint32lsb_t' 又は '2' 符号付き 32 ビット整数/リトルエンディアン：'int32lsb_t' 又は '3' 符号なし 32 ビット整数/ビッグエンディアン：'uint32msb_t' 又は '4' 符号付き 32 ビット整数/ビッグエンディアン：'int32msb_t' 又は '5'
読込開始アドレス	半角英数字	読み込みたいデータが格納されている PLC 機器上の開始アドレスを設定します。　先頭が'0x'の場合は 16 進数と解釈されます。
読込レジスタ数	半角数字	読み込みたいレジスタ数を記載します。

PD Handler Modbus Client の CSV ファイルの書式

各パラメータの区切りはカンマ、先頭がシャープ’#’もしくはスラッシュ’/’の行はコメント行と見なされます。

CSV ファイルが読み込まれると clients オブジェクトに設定された Modbus クライアントデバイスは上書きされます。そのため、CSV ファイルには clients オブジェクトに設定した Modbus クライアントデバイスを含む取得したい全ての Modbus クライアントデバイスを記載して下さい。

5.2.2.7. 基準時刻制御

基準時刻制御は、特定の時刻にデータを取得する機能です。
clients オブジェクトの time_sysnc キーを true に設定し、interval キーと base_time キーで取得間隔と取得時刻を設定します。
基準時刻制御における「取得時間間隔」は 300, 600, 900, 1800, 3600, 7200, 10800, 14400, 21600, 28800, 43200 と 86400 の倍数に限られます。「取得時間間隔」としてこれら以外の値が設定されると PD Handler Modbus Client 内で次のように扱われます。

取得時間間隔の設定値	実動作値
0 ～ 599	300
600 ～ 899	600
900 ～ 1799	900
1800 ～ 3599	1800
3600 ～ 7199	3600
7200 ～ 10799	7200
10800 ～ 14399	10800
14400 ～ 21599	14400
21600 ～ 28799	21600
28800 ～ 43199	28800
43200 ～ 86399	43200
86400～	86400 の倍数

基準時刻制御における取得時間間隔の設定と実動作値

「基準時刻」とは動作の起点となる時刻で、例えば「取得時間間隔」を 300 とし、「基準時刻」を”00:01”とした場合、データの取得は 00:01, 00:06, 00:11 ... 00:56, 01:01 ... 23:56, 00:01 の定刻に行われます。
データの取得開始時刻は「基準時刻」に設定さた時刻そのものではなく、「基準時刻」と「取得時間間隔」から算定される直近の時刻となります。
例えば 08:30 に「基準時刻」”01:05”、 「取得時間間隔」10800 の設定が行われた場合、最初のデータ取得は 10:05 に行われ、以降 13:05, 16:05, 19:05, 22:05, 01:05 の順におこなわれます。

5.3. PD Handler Modbus Server

Modbus サーバーは次表に示すレジスタマップを保持し、PLC 機器からの Modbus プロトコルによる接続を待ち受け、PLC 機器の書き込み操作によりレジスタの値を更新すると共に更新されたレジスタとその値を PD Repeater を介してクラウドに送ります。
また、PD Repeater を介してクラウドから送られる制御メッセージ(JSON 文字列)に基づきレジスタマップを読み書きすることもできます。
レジスタマップは、60 秒毎に更新があればレジスタマップファイル registers.map に出力されます。

レジスタ	開始アドレス	サイズ
デジタル出力(Coils)	0x000	uint8 _t × 2048
デジタル入力(Discrete Input)	0x000	uint8_t × 2048
レジスタ出力(Holdig Registers)	0x000	uint16_t × 2048
レジスタ入力(Input Registers)	0x000	uint16_t × 2048

PD Handler Modbus Server のレジスタマップ

5.3.1. デフォルトパス

PD Handler Modbus Server に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/usr/sbin/pd_handler_modbus_server	常駐実行オブジェクト（デーモン）
/lib/systemd/system/pd_handler_modbus_server.service	Systemd Service ファイル
/etc/init.d/pd_handler_modbus_server	RC ファイル
/var/webui/config/pd_handler_modbus_server.conf	設定ファイル
/var/webui/.modbus_server/registers.map	レジスタマップファイル
/var/run/pd_handler_modbus_server.pid	PID ファイル

PD Handler Modbus Server に関連するファイルのデォルトパス

5.3.2. 設定ファイルの書式

5.3.2.1. 構文

```
{
    "registers_file" : “<レジスタマップファイルのパス名>”,
    “servers”: [
        {
            <serversオブジェクト>
        },
            <serversオブジェクト>
        }
    ]
}
```

5.3.2.2. ルートオブジェクト

キー	データ型	説明
registers_file	文字列	レジスタマップファイルのパス名. デフォルト値は '/var/webui/.modbus_server/pd-data/registers.map'. (MAXPATHLEN)
servers	JSON obj	servers オブジェクト

ルートオブジェクト

5.3.2.3. servers オブジェクト

server オブジェクトは、PD Handler Modbus Serverの動作を規定する設定オブジェクトです。 server オブジェクトの配列数は、最大8 個. Modbus のプロトコル(TCP, RTC) に依存するオブジェクトとプロトコルに依存しない共通のオブジェクトがあります。

キー	対象	データ型	説明
enable	共通	論理値	デフォルト値はfalse
localname		文字列	デバイスのローカル名(デバイス番号). (32byte)
memo		文字列	出力に付加されるユーザー定義文字列. (256byte)
bind		文字列	データを受け取るソケット名. 文字列の先頭が'@' の場合はabstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_handler/＜デバイス番号＞.sock が設定されます.
push_to		文字列	制御メッセージの送り先ソケット名. 文字列の先頭が'@' の場合は abstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_repeater/＜デバイス番号＞.sock が設定されます.
buffer_size		整数値	データのバッファサイズ(byte). デフォルト値は4096
receive		論理値	クラウドからメッセージを受け取る下流方向制御) か否か. デフォルト値はfalse
protocol		文字列	接続プロトコル'tcp' 又は'rtu' を指定. デフォルト値は'tcp'.
writeout		論理値	push to に指定するソケットに出力するか否か. デフォルト値は true.
timeout		整数値	データを取得出来ない場合のタイムアウト(msec). デフォルト値は 5000.
node	TCP	文字列	TCP 接続待ち受け IP アドレス. デフォルト値は'127.0.0.1'
port		整数値	TCP 接続待ち受けポート番号. デフォルト値は 502
device	RTU	文字列	シリアル接続のデバイス名. デフォルト値は'/dev/tty00'
rtu_speed		整数値	シリアル接続のビットレート. デフォルト値は 115200
rtu_bits		整数値	シリアル接続のビット数. 8 又は 7. デフォルト値は 8.
rtu_parity		文字列	シリアル接続のパリティ. 'none','even','odd' のいずれか. デフォルト値は'none'.
rtu_stop		整数値	シリアル接続のストップビット. 1 又は2. デフォルト値は1.
unit		整数値	自身に付与するModbus ユニットID. 整数値1 ～ 247.

servers オブジェクト

5.3.2.4. 利用可能なファンクションコード

PD Handler Modbus Server で利用可能なファンクションコードを示します。

コード	名称	ローカル	クラウド
0x01	Read Coils	○	○
0x02	Read Discrete Input	○	○
0x03	Read Holding Registers	○	○
0x04	Read Input Registers	○	○
0x05	Write Single Coil	○	○
0x06	Write Single Register	○	○
0x07	Read Exception Status	○	
0x09	Write Single Discrete Input		○
0x0a	Write Single Input Register		○
0x0f	Write Multiple Coils	○	○
0x10	Write Multiple Registers	○	○
0x11	Report Slave ID	○	○
0x13	Write Multiple Discrete Input		○
0x14	Write Multiple Input Registers		○
0x16	Mask Write Registers	○	
0x17	Write And Read Registers	○	○

PD Handler Modbus Server で利用可能なファンクションコード

ローカルは、ローカルに接続するPLC 機器からに指定可能なファンクションコード、クラウドはクラウド側か指定可能なファンクションコードを意図します。

6. PD Handler BLE (Node.js)

6.1. デフォルトパス

PD Handler BLE (Node.js)に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/opt/pd/handler/pd-handler.js	実行ファイル
/etc/init.d/pd-handler-ble-js	RC ファイル
/var/webui/config/pd-handler-ble.conf	設定ファイル
/var/run/pd-handler-ble-js.pid	PID ファイル
/var/webui/pd-logs/pd-handler-ble-js.log	ログファイル
/var/werbui/.blebackup/pd-handler-ble.conf	設定ファイルのバックアップ
/var/werbui/.blebackup/pd-handler-ble-js-restore.log	リストア用ファイル
/var/webui/.bsensor/xxx.json	local 用センサデータファイル(xxx is localname.)

PD Handler BLE (Node.js)に関連するファイルのデォルトパス

6.2. 設定ファイルの書式

6.2.1. 構文

{

“servers”: <servers オブジェクト>,

“beacon”: <beacon オブジェクト>,

“blesensor”: <blesensor オブジェクト>

}

6.2.2. ルートオブジェクト

キー	データ型	説明
servers	JSON obj	serversオブジェクト
beacon	JSON obj	beacon オブジェクト
blesensor	JSON 配列	blesensor オブジェクト

ルートオブジェクト

6.2.3. servers オブジェクト

キー	データ型	説明
json_ldir	文字列	データ用ディレクトリパス
handler_ldir	文字列	バックアップ用ディレクトリパス
blehandler_monitor_api	文字列	beacon モニタリング用 API ファイル

serevers オブジェクト

6.2.4. beacon オブジェクト

キー	データ型	説明
localname	文字列	デバイスのローカル名(デバイス番号). (32byte)
push_to	文字列	制御メッセージの送り先ソケット名. 文字列の先頭が'@' の場合は abstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_repeater/＜デバイス番号＞.sock が設定されます.
duplicate_type	文字列	“interval”, “entry”, “inout”
duplicate_interval	整数値	重複制御時間間隔. 0 ~ 3600000[msec]
appendix_info	文字列	付随情報.デフォルト値はシリアルナンバー.
payload_manage	JSON obj	ペイロード管理
data_filter_rule	JSON 配列	データフィルタ
rssi_filter	整数値	RSSI フィルタ
add_data	JSON obj	ユーザー定義情報
enable	論理値	データを送信するか否か

beacon オブジェクト

6.2.4.1. payload_manage オブジェクト

キー	データ型	説明
data	論理値	送信データにアドバタイズデータを付与するか否か
localname	論理値	送信データにローカルネームを付与するか否か
type	論理値	送信データにビーコンタイプ(iBeacon)を付与するか否か

payload_manage オブジェクト

6.2.4.2. data_filter_rule オブジェクト

キー	データ型	説明
length	整数値	prefixの長さ
prefix	文字列	前方一致のデータフィルタ

data_filter_rule オブジェクト

6.2.5. blesensor オブジェクト

キー	データ型	説明
localname	文字列	デバイスのローカル名(デバイス番号). (32byte)
push_to	文字列	制御メッセージの送り先ソケット名. 文字列の先頭が'@' の場合は abstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_repeater/＜デバイス番号＞.sock が設定されます.
uuid	文字列	Bluetooth Device Address(コロンを削除し、16 進数を英小文字で表した文字列)
memo	文字列	出力に付加されるユーザー定義文字列. (256byte)
interval	整数値	データ送信間隔.[msec]
txpower	整数値	送信出力[dBm]
local	論理値	送信先設定の本体内(local)へ送信するか否か
enable	論理値	データを送信するか否か

blesensor オブジェクト

7. PD Handler BLE (C)

7.1. デフォルトパス

PD Handler BLE (C)に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/usr/sbin/pd-handler-ble-c	常駐実行オブジェクト（デーモン）
/etc/init.d/pd-handler-ble-c	RC ファイル
/var/webui/config/pd-handler-ble.conf	設定ファイル
/var/run/pd-handler-ble-c.pid	PID ファイル
/var/webui/pd-logs/pd-handler-ble-c.log	ログファイル
/var/werbui/.blebackup/pd-handler-ble.conf	設定ファイルのバックアップ
/var/werbui/.blebackup/pd-handler-ble-c-restore.log	リストア用ファイル
/var/webui/.bsensor/xxx.json	local 用センサデータファイル(xxx is localname.)
/opt/pd/lua/ble/devices/*.lua	Lua 拡張用ファイル

PD Handler BLE (C)に関連するファイルのデォルトパス

7.2. 設定ファイルの書式

設定ファイルは PD Handler BLE (Node.js)と同じです。

詳細は 6.2.設定ファイルの書式を参照してください。

8. PD Handler UART

8.1. デフォルトパス

PD Handler UART に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/usr/sbin/pd-handler-uart	常駐実行オブジェクト（デーモン）
/etc/init.d/pd-handler-uart	RC ファイル
/var/webui/config/pd-handler-uart.conf	設定ファイル
/var/run/pd-handler-uart.pid	PID ファイル
/var/webui/pd-logs/pd-handler-uart.log	ログファイル
/opt/pd/lua/uart/enocean/devices/*.lua	Lua 拡張用ファイル

PD HandlerUART に関連するファイルのデォルトパス

8.2. 設定ファイルの書式

8.2.1. EnOcean

8.2.1.1. 構文

{

“prototype”:”enocean”,

“enocean”: <enocean オブジェクト> ,

“enocean_device”: <enocean_device オブジェクト>

}

8.2.1.2. ルートオブジェクト

キー	データ型	説明
prototype	文字列	“enocean”
enocean	JSON obj	enocean オブジェクト
enocean_device	JSON 配列	enocean_device オブジェクト

ルートオブジェクト

8.2.1.3. enocean オブジェクト

キー	データ型	説明
serial_port	文字列	デフォルトは”/dev/ttyEX2”
raw_data_mode	論理値	true なら受信データをそのまま送信

enocean オブジェクト

8.2.1.4. enocean_device オブジェクト

キー	データ型	説明
localname	文字列	デバイスのローカル名(デバイス番号). (32byte)
push_to	文字列	制御メッセージの送り先ソケット名．文字列の先頭が'@' の場合は abstract namespace と解釈します． 空の場合は、デフォルト値 @/pd_repeater/＜デバイス番号＞.sock が設定されます.
deviceid	文字列	ID
memo	文字列	出力に付加されるユーザー定義文字列. (256byte)
eep	文字列	EnOcean Equipment Profiles
enable	論理値	データを送信するか否か

enocean_device オブジェクト

9. PD Handler HVSMC

9.1. デフォルトパス

PD Handler HVSMC に関連するファイルのデォルトパスは次の通りです。

パス名	説明
/usr/sbin/pd-handler-hvsmc	常駐実行オブジェクト（デーモン）
/etc/init.d/pd-handler-hvsmc	RC ファイル
/var/webui/config/pd-handler-hvsmc.conf	設定ファイル
/var/run/pd-handler-hvsmc.pid	PID ファイル
/var/webui/pd-logs/pd-handler-hvsmc.log	ログファイル

PD Handler HVSMC に関連するファイルのデォルトパス

9.2. 設定ファイルの書式

9.2.1. 構文

9.2.1.1. 構文

{

“hvsmc”: <hvsmc オブジェクト>

}

9.2.1.2. ルートオブジェクト

キー	データ型	説明
hvsmv	JSON obj	hvsmc オブジェクト

ルートオブジェクト

9.2.1.3. hvsmc オブジェクト

キー	データ型	説明
enable	論理値	データを送信するか否か. デフォルト値は false
localname	文字列	デバイスのローカル名(デバイス番号). (32byte)
memo	文字列	出力に付加されるユーザー定義文字列. (256byte)
push_to	文字列	制御メッセージの送り先ソケット名. 文字列の先頭が'@' の場合は abstract namespace と解釈します. 空の場合は、デフォルト値 @/pd_repeater/<デバイス番号>.sock が設定されます.
nif	文字列	高圧スマート電力量メータを接続する Ethernet インタフェースのデバイス名.
raw_data_mode	論理値	取得したデータを変換せずに PD Repeater へ送るか否か. デフォルト値は false
demand_info	論理値	定時計測時に需要電量の取得を行うか否か. デフォルト値は false
measured_info	論理値	計測値の取得を行うか否か. デフォルト値は false
measured_interval	整数値	計測値の取得を行う時間間隔（min）. デフォルト値は 0

hvsmc オブジェクト

ぷらっとホーム株式会社
〒102-0073　東京都千代田区九段北 4-1-3　日本ビルディング九段別館 3F