easysync デモの紹介

株式会社エージーテック

2025 年 1 月 24 日

免責事項

株式会社エージーテックは本書の使用を、利用者またはその会社に対して「現状のまま」でのみ許諾するものです。株式会社エージーテックは、いかなる場合にも本書に記載された内容に関するその他の一切の保証を、明示的にも黙示的にも行いません。本書の内容は予告なく変更される場合があります。

商標

Copyright © 2025 AG-TECH Corp. All rights reserved. 本書の全文、一部に関わりなく複製、複写、配布をすることは、前もって発行者の書面による同意がない限り禁止します。

すべての Pervasive ブランド名および製品名は、Pervasive Software Inc. の米国およびその他の国における登録商標または商標です。また、すべての Actian のブランド名は、Actian Corporation の米国およびその他の国における登録商標または商標です。

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。

easysync デモの紹介

最終更新：2025 年 1 月 24 日

Copyright © 2025 AG-TECH Corp. All rights reserved.

目次

はじめに 4
デモの概要 5
エッジデバイスの設定 6
Bluetooth アクセスライブラリのインストール 10
電力量読み込みプログラムの概要 11
電力量読み込みプログラム 13
同期先のデータベース作成 17
簡易同期コマンド easysync.exe の実行 18
最後に 20

Copyright © 2025 AG-TECH Corp. All rights reserved.

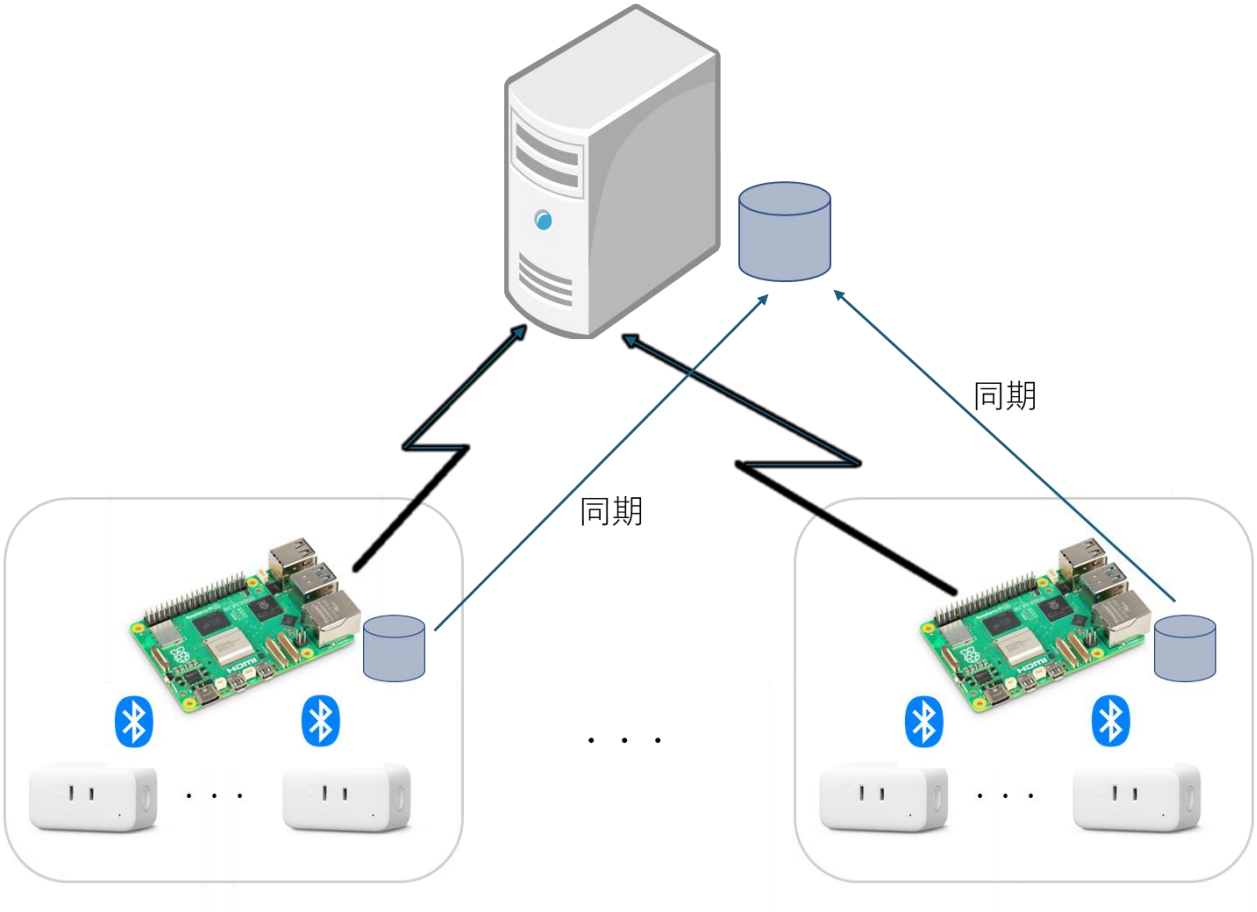
はじめに

本書では、Actian Zen v16 で追加された簡易同期機能 easysync について、電力量読み込みプログラムで生成したデータを用いて使い方を紹介します。
easysync を使用することで、定期的なバックアップだけでなく、エッジデバイスからセンターへのデータ集約を行うことも容易にできます。
今回紹介するデモでは、エッジデバイスからセンターへのデータ集約を行う構成で easysync の使い方を紹介いたします。

Copyright © 2025 AG-TECH Corp. All rights reserved.

デモの概要

本デモでは、スマートプラグ（SwitchBot プラグミニ）から Bluetooth 通信で読み込んだ消費電力をエッジデバイス（Raspberry Pi）のデータベースに保存し、そのデータを easysync を使用して定期的に PC に同期を行います。
複数のエッジデバイスから 1 台の PC に同期を行うため、テーブルのデータ型は一部異なりますが、同期を行うためにプログラムを作成する必要はありません。
1 対 1 で同期を行う場合には、テーブルのデータ型は同一で問題ありません。



本デモでは、Python のプログラムから消費電力を読み取り Btrieve 2 API でデータの書き込みを行います。
このため、Python のインストール、SWIG を使用した Python 用インターフェイスの生成についても説明します。

Copyright © 2025 AG-TECH Corp. All rights reserved.

エッジデバイスの設定

◆ データベース作成

まずは、データベースの作成から行います。
データベースの作成は isql で DEMODATA に接続し、次の SQL 文の実行で行います。

```
create database syncdemoSmartAG DICTIONARY_PATH '/usr/local/syncdemo/db'
V2_METADATA ENCODING 'UTF-8'
```

◆ テーブルの作成

エッジデバイス側のテーブルは、次の SQL 文にて作成します。
電力消費量を記録する pluglog テーブルとデータを受信するスマートプラグの情報を記録した pugcfg テーブルの 2 つを使用します。
isql で作成したデータベースに接続して SQL を実行します。

```
SET LEGACYTYPESALLOWED = ON;

CREATE TABLE "pluglog" USING 'pluglog.mkd' PAGESIZE=4096 SYSDATA_KEY_2 (
 "ts" AUTOTIMESTAMP DEFAULT '0' CONSTRAINT "DK_ts" NOT UNIQUE USING 0,
 "MacAdd" CHAR(17) NOT NULL,
 "wat" DOUBLE NOT NULL );
CREATE INDEX "pluglog-kMacAdd" USING 3  ON "pluglog" ( "MacAdd", "ts" );
CREATE INDEX "pluglog-kts" USING 2  ON "pluglog" ( "ts", "MacAdd" );
CREATE INDEX "pluglog-MacAdd" USING 1  ON "pluglog" ( "MacAdd" );

CREATE TABLE "plugcfg" USING 'plugcfg.mkd' PAGESIZE=4096 (
 "rno" IDENTITY DEFAULT '0' CONSTRAINT "UK_rno" UNIQUE USING 0,
 "macaddr" CHAR(17) NOT NULL,
 "flg" INTEGER NOT NULL );
CREATE INDEX "plugcfg_k1" USING 1  ON "plugcfg" ( "flg", "rno" );

SET LEGACYTYPESALLOWED = OFF;
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

◆ Python のインストール

次のコマンドで Python のインストールを行います。

```
sudo apt-get install python3 python3-dev
```

◆ SWIG のインストール

次のコマンドで SWIG のインストールを行います。

```
sudo apt-get install swig
```

◆ Btrieve 2 API SDK をインストールして Python 用のインターフェイス モジュールを作成

1. Btrieve2 API SDK をダウンロード
   次の WEB ページから Btrieve2 API SDK をダウンロードします。
   https://www.agtech.co.jp/actian/support/reference/sdk/access_methods/btrieve/
   Raspberry Pi OS 環境では、Linux 用の SDK を使用します。
   Zen-SDK-Btrieve2API-linux-noarch-yy.yy.zzz.zzz.tar.gz
   ※この手順では、ファイルを /usr/local にダウンロードしています。

2. SDK フォルダーを、以下のコマンドで作成して下さい。

```
mkdir /usr/local/actianzen/SDK
```

3. ダウンロードした SDK を展開します。
   /usr/local/actianzen/SDK をカレントにし、tar コマンドで展開します。

```
cd /usr/local/actianzen/SDK
tar -zxf /usr/local/Zen-SDK-Btrieve2API-linux-noarch-
```

※ファイルを /usr/local にダウンロードした前提で記載しております。

Copyright © 2025 AG-TECH Corp. All rights reserved.

次のフォルダーが作成されます。

```
Zen-SDK-Btrieve2API-linux-noarch-yy.yy.zzz.zzz
  │
  ├ doc
  │
  ├ intf
  │
  └ samples
```

便宜上フォルダー名を変更します。

```
mv Zen-SDK-Btrieve2API-linux-noarch-yy.yy.zzz.zzz Btrieve2API
```

4. Btrieve 2 API SDK が入っているディレクトリをカレントに変更します。

```
cd Btrieve2API
```

5. btrievePython.cpp を生成します。

```
swig -cppext cpp -outcurrentdir -python -c++ -Iinclude -o
btrievePython.cpp swig/btrievePython.swig
```

6. Btrieve 2 の Python 用インターフェイスモジュール（btrievePython.py と _btrievePython.so）をビルドします。

```
g++ -c -fPIC -Iinclude -I/usr/include/python3.8 btrievePython.cpp
g++ -shared btrievePython.o -o _btrievePython.so -L/usr/local/actian
zen/lib -lbtrieveCpp
```

上記で使用されている Python へのパスは、インストールされている Python のバージョンにより異なる場合があることに注意してください。

7. 電力量読み込みプログラム用のフォルダーを作成します。

```
mkdir /usr/local/zendemo
```

8. /usr/local/zendemo フォルダーのパーミッションで「内容の変更」に「すべて」を設定します。

Copyright © 2025 AG-TECH Corp. All rights reserved.

```
chmod 777 /usr/local/zendemo
```

9. Btrieve 2 SDK に含まれている Python サンプルと必要なファイルをそのフォルダー内にコピーします。

```
Cd /usr/local/zendemo
cp /usr/local/actianzen/SDK/Btrieve2API/samples/btest.py .
cp /usr/local/actianzen/SDK/Btrieve2API/btrievePython.py .
cp /usr/local/actianzen/SDK/Btrieve2API/_btrievePython.so .
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

Bluetooth アクセスライブラリのインストール

次のコマンドを実行して、Python から Bluetooth デバイスとの通信を行うためのライブラリ bluepy のインストールを行います。
今回は、Bluetooth のブロードキャストを簡単に読み込むことが可能な bluepy を使用していますが、ブロードキャストの読み込みが可能なライブラリであれば、他のライブラリでも代用可能です。

```
sudo apt-get update
sudo apt-get -y upgrade
sudo apt-get -y install python3-pip libglib2.0-dev
sudo pip3 install bluepy
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

電力量読み込みプログラムの概要

電力量読み込みプログラムはスマートプラグから送られるブロードキャストメッセージ（アドバタイズパケット）に含まれる電力消費量を約 5 秒おきに読み込み plaglog テーブルに書き込みます。
スマートプラグは、plugcfg テーブルに登録された複数のスマートプラグのデータを読み込みスマートプラグごとに 1 レコードのデータを登録します。
Python のプログラムから Bluetooth アクセス用のライブラリ blurpy を使用しています。
bluepy を使用して Bluetooth からのブロードキャストメッセージに含まれる消費電力量を読み込みます。

SwitchBot プラグミニのブロードキャストメッセージは、次のようなフォーマットで送信されます。

Byte: 0	UUID	0x09	UUID is 0x0969, big-endian
Byte: 1		0x69	
Byte: 2	MAC Address	0x01	Device Mac Address 0x010203040506, big-endian
Byte: 3		0x02	
Byte: 4		0x03	
Byte: 5		0x04	
Byte: 6		0x05	
Byte: 7		0x06	
Byte: 8	Sequence Number	1~255	1~255 increments, 256 overflows to 1, and the broadcast packet data is automatically incremented when there is an update

Copyright © 2025 AG-TECH Corp. All rights reserved.

Byte: 9	Plug Mini state	0x00 - power off 0x80 - power on	Plug Mini's switch status
Byte: 10	bit[0] delay	0-no delay, 1-has delay	Is there a delay
	bit[1] timer	0-no timer, 1-has timer	Is there a Timer
	bit[2] sync utc time	0-no sync time , 1-already sync time	Whether the UTC time has been synchronized
Byte: 11	wifi rssi		The rssi signal value of the currently connected wifi
	Overload	byte 12[7] overload	Whether the Plug Mini is overloaded, more than 15A current overload
Byte: 12-13	Power	byte 12[0-6] Power MSB byte 13[0-7] Power LSB	Plug Mini current power value of the load

Byte: 12-13 の値が 0.1W 単位の消費電力量となります。

UUID が 0x0969 のメッセージが SwitchBot プラグミニのメッセージと判断しています。

Copyright © 2025 AG-TECH Corp. All rights reserved.

電力量読み込みプログラム

以下に電力量読み込みプログラムのソースを示します。エディターで拡張し .py として保存することで実行可能です。

```
import sys
import binascii
from bluepy.btle import Scanner, DefaultDelegate
import keyboard
import time
import struct
sys.path.append("/usr/local/actianzen/lib")

import btrievePython

macadder = []

wts = []
wMac = []
wWat = []
min_x = 0
max_x = 255
btrieveFileName = "/usr/local/syncdemo/db/pluglog.mkd"
cfgFileName = "/usr/local//syncdemo/db/plugcfg.mkd"
recordFormat = "<q17sd"
recordLength = 33
cfgFormat = "<l17sl"
cfgLength = 25
class ScanDelegate(DefaultDelegate):
  def __init__(self):
    DefaultDelegate.__init__(self)

  def handleDiscovery(self, dev, isNewDev, isNewData):
    for (adtype, desc, value) in dev.getScanData():
      if (adtype != 255): continue
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

```
      manufacturer = binascii.unhexlify(value)
      print('MAC adder : ' + dev.addr + '     ' + hex(manufacturer[0]) + ':' + hex(man
ufacturer[1]))
      if ((manufacturer[0] != 0x69 or manufacturer[1] != 0x9)): continue
      if (len(manufacturer) < 14): continue

      isState = ( manufacturer[9] & 0b10000000 ) >> 7
      Rssi = manufacturer[11]
      isOverload = ( manufacturer[12] & 0b10000000 ) >> 7
      Power = ( ( ( manufacturer[12] << 8 ) + manufacturer[13] ) & 0x7fff ) / 10

#      print('MAC adder : ' + dev.addr + '     ' + hex(manufacturer[0]) + ':' + hex(ma
nufacturer[1]))
      print( 'Power on: ' + str( bool( isState ) ) )
      print( 'AP RSSI: -' + str( Rssi ) )
      print( 'isOverload: ' + str( bool( isOverload ) ) )
      print( 'Watt: ' + str( Power ) )
      for i in range(len(macadder)):
         if (dev.addr == macadder[i]):
            wts[i] = 0
            wMac[i] = macadder[i]
            wWat[i] = Power

#      exit()

btrieveClient = btrievePython.BtrieveClient()
btrieveFile = btrievePython.BtrieveFile()
rc = btrieveClient.FileOpen(btrieveFile, cfgFileName, None, btrievePython.Btrieve.OPEN
_MODE_NORMAL)
#print('open sts:',rc)
cfgrecord = struct.pack(cfgFormat, 0, bytes('                    ','shift-jis'), 0)
rc = btrieveFile.RecordRetrieveFirst(btrievePython.Btrieve.INDEX_NONE, cfgrecord)
sts = btrieveFile.GetLastStatusCode()
print('get sts:',sts)
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

```
while True:
    updata = struct.unpack(cfgFormat,cfgrecord)
    print(updata)
    wts.append(0)
    wMac.append(updata[1].decode('cp932'))
    wWat.append(0)
    print("MACaddress",updata[1].decode('cp932'))
    rc = btrieveFile.RecordRetrieveNext(cfgrecord)
    sts = btrieveFile.GetLastStatusCode()
    if (sts != btrievePython.Btrieve.STATUS_CODE_NO_ERROR):
       break

rc = btrieveClient.FileClose(btrieveFile)
rc = btrieveClient.Reset()

while True:
    btrieveClient = btrievePython.BtrieveClient()
    btrieveFile = btrievePython.BtrieveFile()
    rc = btrieveClient.FileOpen(btrieveFile, btrieveFileName, None, btrievePython.Bt
rieve.OPEN_MODE_NORMAL)
    print('open rc:',rc)

    scanner = Scanner().withDelegate( ScanDelegate() )
    scanner.scan(1)
    print('count:',len(wMac))
    for i in range(len(wMac)):
       record = struct.pack(recordFormat, wts[i],bytes(wMac[i],'shift-jis'),wWat[i])
       rc = btrieveFile.RecordCreate(record)
       print('instert rc:',rc)
    rc = btrieveClient.FileClose(btrieveFile)
    rc = btrieveClient.Reset()
    print('--------------------------------------------')
    wcnt = 0
    while wcnt < 5:
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

```
        if keyboard.is_pressed('q'):  # 'q'キーが押された場合
            exit()
        wcnt = wcnt + 1
        time.sleep(0.1)
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

同期先のデータベース作成

同期先 PC のデータベースは、複数の Edge 端末からのデータを 1 つのテーブルに同期（多対 1）するため、一部の項目のデータ型がエッジデバイス側のテーブルとは異なっています。
1 対 1 での同期の場合には、同期元と同期先のテーブルは同一のデータ型（全く同じ構造のテーブル）で使用可能です。
具体的には、同期元 AUTOTIMESTAMP 型の ts フィールドが既に値が設定されていることから、同期先では TIMESTAMP2 型のフィールドとして定義しています。

```
SET LEGACYTYPESALLOWED = ON;

CREATE TABLE "pluglog" USING 'pluglog.mkd' PAGESIZE=4096 SYSDATA_KEY_2 (
 "ts" TIMESTAMP2(9) NOT NULL,
 "MacAdd" CHAR(17) NOT NULL,
 "wat" DOUBLE NOT NULL );

CREATE INDEX "pluglog-kMacAdd" USING 3  ON "pluglog" ( "MacAdd", "ts" );
CREATE INDEX "pluglog-kts" USING 2  ON "pluglog" ( "ts", "MacAdd" );
CREATE INDEX "pluglog-MacAdd" USING 1  ON "pluglog" ( "MacAdd" );
CREATE INDEX "pluglog-ts" USING 0  ON "pluglog" ( "ts" );

SET LEGACYTYPESALLOWED = OFF;
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

簡易同期コマンド easysync.exe の実行

エッジデバイス側で簡易同期コマンドを実行します。
まず、次のような簡易同期の設定を格納した JSON 構成ファイルを作成します。

```
{
	"version":  1,
	"settings": {
		"log_file": "/usr/local/switchbot/esync.log",
		"log_level":	"verbose",
		"polling_interval_sec": 10,
		"resume_on_error":	true
	},
	"files":	[{
			"source_file":	"/usr/local/switchbot/db/pluglog.mkd",
			"destination_file":	"btrv://ユーザー名@サーバーの IP
アドレス/plug?dbfile=PLUGLOG.MKD&pwd=ユーザーのパスワード",
			"unique_key":	3,
			"create_destination":	true,
			"_last_copied_record_timestamp":	"0000000000000000",
			"_last_existing_transaction_time":	"0000000000000000",
			"_last_file_timestamp": "0000000000000000"
		}]
}
```

※ファイルパス等は実行環境に合わせて変更してください。

この構成ファイルの設定では、10 秒おきに同期を行います。
簡易同期を行うためには、システムデータ v2 が設定されていることが必要です。
レコードを特定するためにユニークなキーも必要です。
この構成ファイルでは、ユニークなキーとして 3 を指定しています。

ターミナルで次のコマンドを実行します。

easysync JSON 構成ファイル

Copyright © 2025 AG-TECH Corp. All rights reserved.

構成ファイルの設定内容や、easysync の制限事項については、次のマニュアルをご覧ください。

Zen v16 | コマンド ライン インターフェイス ユーティリティ

Copyright © 2025 AG-TECH Corp. All rights reserved.

最後に

easysync を使用することで、バックアップやデータの集約等が簡単に可能となりますが、削除したデータの同期にはレプリケーションツールの DataExchange が有用です。
ただし、DataExchange は、Windows Server 製品（Zen Enterprise Server、Cloud Server）でのみ使用が可能です。
easysync であれば、OS やエディションを問わずご使用いただけます。

今回のデモでは、単純な同期ではなく、エッジデバイスで取得したデータをサーバーに集約しています。
使い方次第で、バックアップに限らず、様々な用途でご使用いただける可能性があります。

Copyright © 2025 AG-TECH Corp. All rights reserved.