



汎用IoT/Edgeデータ分析クラウド SpeedBee Hive

～ カスタムコレクタを使った データ収集機能の開発 ～

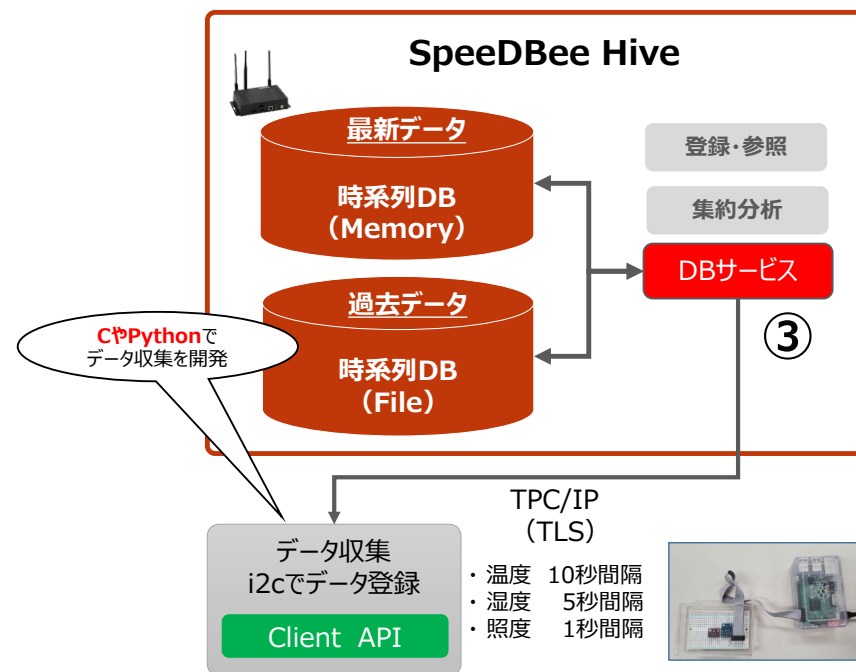
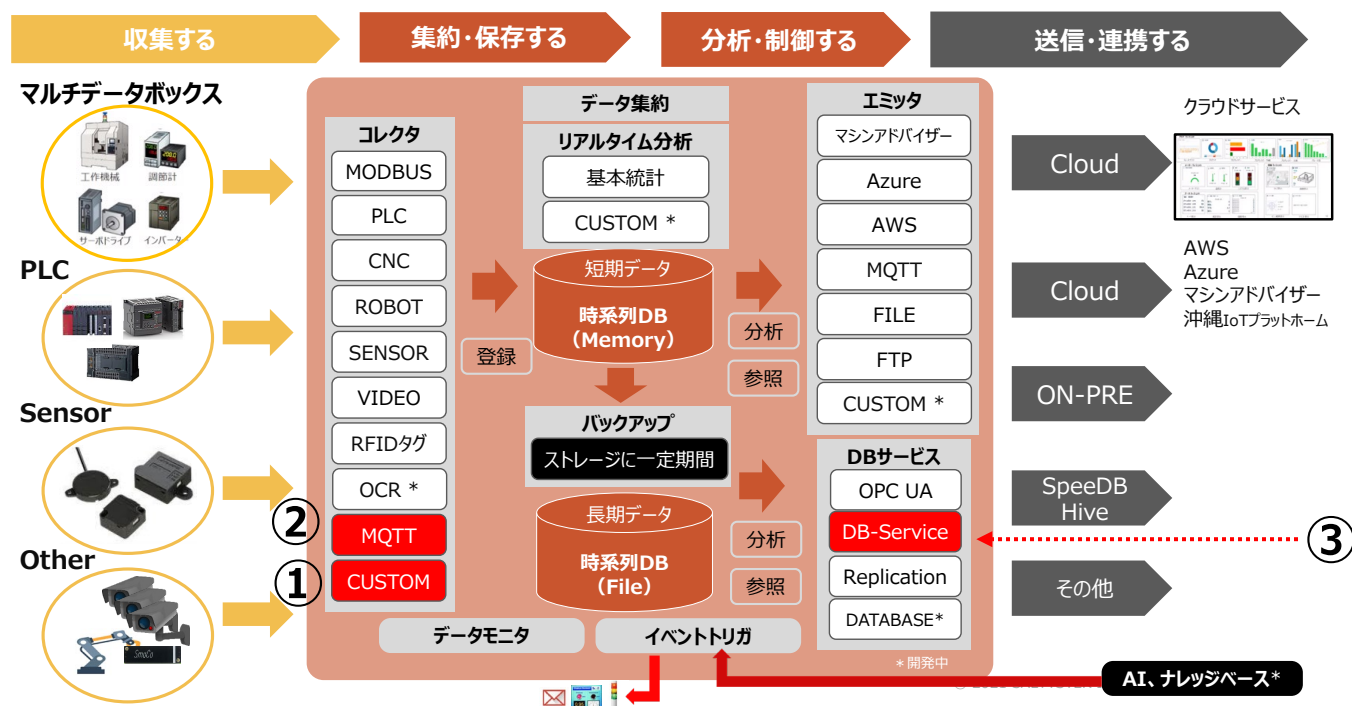
簡単・安価・エッジ・チョコ停監視・なんでも繋がるIoT
『SpeedBee Hive』

2021年9月9日
株式会社ソルティスター

SpeedBeeHiveへのデータ登録と各種機能との連携方法

標準のデータ取り込み機能で対応できない場合、ユーザーにて以下3パターンで開発が可能です

- ① カスタムコレクタ : C言語で開発、プロセス型のモジュールでHiveのDBにデータ登録
- ② MQTTコレクタ : Python/JavaScriptで開発、MQTTブローカが別途必要
- ③ DBクライアント : Python/Cで開発、DBサーバを介してクライアント/サーバ型でHiveのDBにデータ登録



SpeedBee Hiveでは、データの収集を行うモジュールのことを「コレクタ」と呼びます。
標準でいくつかの「コレクタ」が用意されていますが、ユーザーが独自にコレクタを開発することができるオプションが「カスタムコレクタ」です。

<カスタムコレクタ基本概要>

- 1.提供されているインターフェースに合わせてプログラミング（C言語）
- 2.各種プラットフォーム内でビルド、動的ライブラリを生成
- 3.SpeedBee Hiveの画面からコレクタとして登録、設定する

※上記流れを理解するには、
マニュアル「SpeedBee Hive（カスタムコレクタ）」の「2.カスタムコレクタのサンプル」の流れに従って、サンプルを動かしてみることをお勧めします。

◆ カスタムコレクタのサンプル（マニュアルに以下二つのサンプルが付属しています）

●randomsin

SIN波の振幅値(-1～+1)と、ランダムな0~100までの整数値の2つのデータを登録するコレクタのサンプルです。
データはいずれも単純なスカラ値で、前者はDOUBLE型、後者はINT32型としています。

●fixarray

円運動する物体のX-Y-Z座標値をデータとして登録するコレクタのサンプルです。
データは3要素のDOUBLE型配列としています。

環境および準備するもの

OS	Windows 10
Tool	Microsoft Visual Studio Professional 2019
SpeedBee Hive	3.0.3

◆ ビルド手順

① サンプルコレクター式をデスクトップにコピー

```
xcopy /I "C:¥Program Files¥SALTYSTER¥SpeedBeeHive¥share¥example" c:¥Users¥xxxx¥Desktop¥example
```

② コピー先フォルダへ移動して、ビルド実行

```
cd c:¥Users¥xxxx¥Desktop¥example  
nmake
```

③ 生成されたライブラリ（dll）を指定フォルダへコピー

```
xcopy cltrfixarray.dll c:¥ProgramData¥SALTYSTER¥SpeedBeeHive¥libs  
xcopy cltrrandomsin.dll c:¥ProgramData¥SALTYSTER¥SpeedBeeHive¥libs
```

※ビルドにはhiveの提供するコレクタライブラリのリンク、ヘッダファイルのインクルードが必要

カスタムコレクタで開発したモジュールの登録 ～カスタムコレクタを使用する為の設定～

SpeedBee Hive Manager

保護されていない通信 | 192.168.1.5/hive/

システム サービス コレクタ イベント エミッタ

リソース

PLC

カスタム

サイン波

コレクタ > カスタム (サイン波)

保存 削除

カスタム名 タイプ ライブラリ

サイン波 so cltrandomsin.so

パラメータ

20.0

☐ 無効にする

データ一覧

データ名	データ型	配列数	レート
sin wave	DOUBLE	1	200
random integer	INT_32	1	0

初期化 編集

Copyright 2020 SALTYSER Corp.

例) サイン波を収集するコレクタを開発

ライブラリへ渡すパラメータの指定

登録先データの指定

カスタムコレクタ開発で可能な事例

- ✓ 他システムからのデータ取込
- ✓ ICカードからのデータ取込
- ✓ RFIDタグからのデータ取込
- ✓ 複合機からのデータ取込
- ✓ ロボットからのデータ取込
- ✓ センサーからのデータ取込

カスタムコレクタの開発は、SpeedBee Hiveの提供するインターフェースに合わせてプログラミングします。
詳細は「3.カスタムコレクタのソースコードの説明」を参考にして下さい。

1) コレクタ定義

コレクタ名の名称
コレクタの説明
複数起動の許可

初期化と後処理

コンストラクタ（初期化）

デストラクタ（後処理）

メイン処理

メイン処理

```
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>

/* このヘッダファイルをインクルードします */
#include "sdcltr_collector.h"

/* コレクタメソッド. */
static const COLLECTOR_OUTPUT_TYPE *constructor(COLLECTOR *collector, const char *params);
static bool destructor(COLLECTOR *collector);
static int mainloop(COLLECTOR *collector);
static int proc(COLLECTOR *collector, void *data, int skipped);

/* コレクタ定義データ：名前、説明、パラメータ、出力データを記述する. */
const static COLLECTOR_INFO info = {
    .name          = "very simple collector", // コレクタの名称
    .description   = "collector description", // コレクタの説明
    .multiple      = false,                  // 複数インスタンスを許すかどうか
    .functions     = {
        .constructor = constructor,
        .destructor  = destructor,
        .main        = mainloop,
    },
};

/* コレクタ定義データを外部シンボルに出力するマクロ（必須）. */
EXPORT_COLLECTOR_INFO(info);
```

コンストラクタでは、コレクタが扱うデータ型定義や必要に応じてメモリ確保などを行います。
デストラクタでは、ヒープメモリの確保がある場合やリソースの解放が必要な場合の処理をコーディングします。

コレクタ定義

コレクタ名の名称
コレクタの説明
複数起動の許可

2) 初期化と後処理

コンストラクタ (初期化)

デストラクタ (後処理)

メイン処理

メイン処理

```
/* このコレクタのインスタンスが生成された際にコールされる関数。 */
static const COLLECTOR_OUTPUT_TYPE *constructor(COLLECTOR *collector, const char *params) {
    /* 必要に応じてメモリ確保 */
    collector->user_data = malloc(64);

    /* コレクタが出力するデータの定義。データ 0 はランダムな整数。最後はNONEにする。 */
    const static COLLECTOR_OUTPUT_TYPE output_types[] = {
        // type          array type      description      sampling rate
        { SDCLTR_INT32,   SDCLTR_SCALAR,  "random integer", 0 },
        { SDCLTR_NOTYPE,  SDCLTR_SCALAR,  "",               0 },
    };
    return output_types;
}

static bool destructor(COLLECTOR *collector) {
    /* メモリ確保している場合は、この関数内で解放する */
    free(collector->user_data);
    return true;
}
```

メイン処理では、実際にデータを収集して、SpeedBee Hiveへデータを登録する処理をコーディングします。

コレクタ定義

コレクタ名の名称
コレクタの説明
複数起動の許可

初期化と後処理

コンストラクタ（初期化）

デストラクタ（後処理）

3) メイン処理

メイン処理

```
/* このコレクタのインスタンス実行中、指定した周期で定期的にコールされる関数。 */
static int proc(Collector *collector, void *data, int skipped) {

    int v0 = rand() % 100;
    HIVE_INFO("very simple collector out %d\n", v0);

    /* コンストラクタで宣言した最初のデータへv0の値を登録する。 */
    sdcltr_insert_data(collector, 0, &v0);

    return SDCLTR_EXIT_SUCCESS;
}

/* メインループ。 */
/* 処理の周期をhiveに任せる場合はsdcltr_interval_callを使用します。 */
static int mainloop(Collector *collector) {
    int code = sdcltr_interval_call(collector, proc, 1000*1000, NULL);
    sdcltr_exit(collector, code);
}
```


収集する

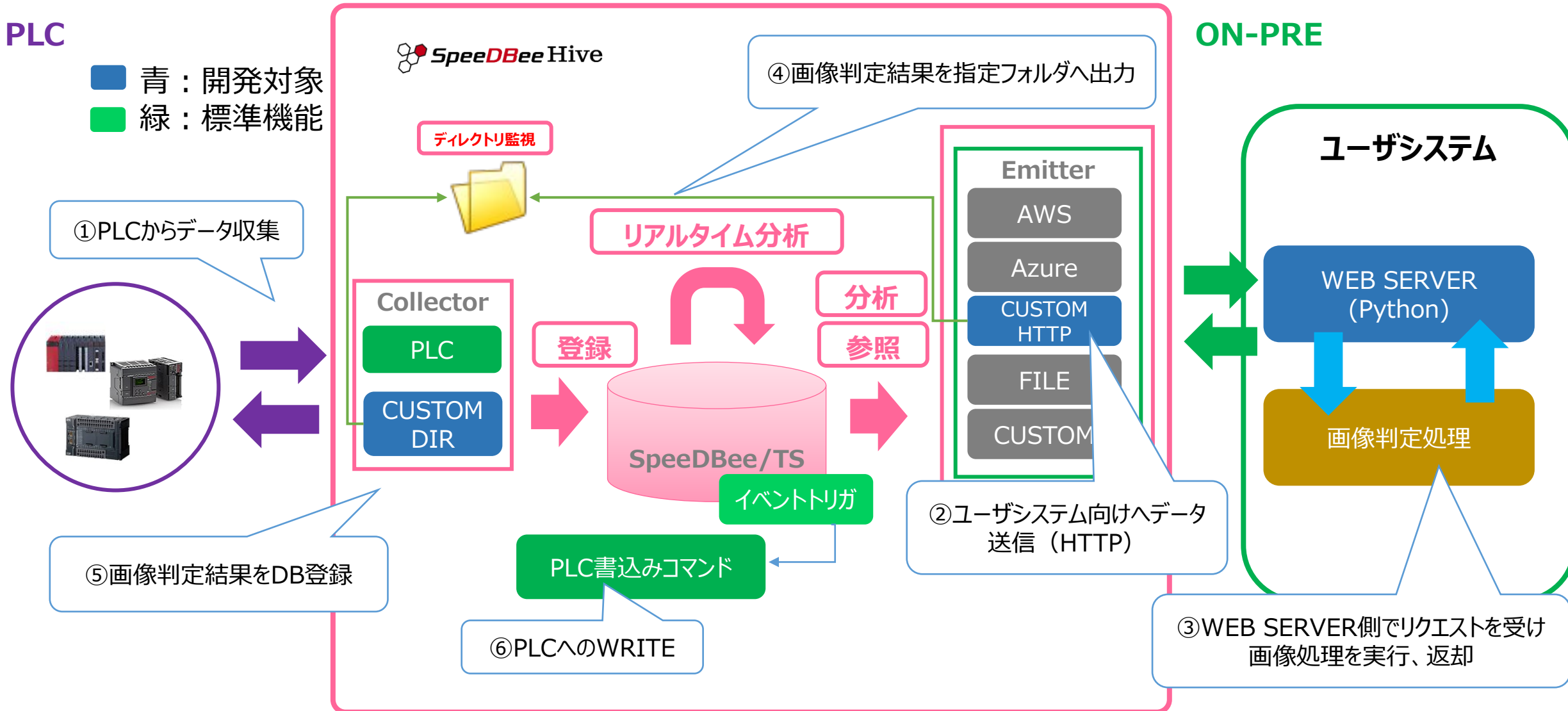
保存する

分析する

連携する

PLC

- 青：開発対象
- 緑：標準機能



✓ 収集データの数値変換

コレクタで収集したデータをそのまま登録するのではなく、計算式を適用して計算後の結果を登録する機能です。

- 対象
アナログ値や各コレクタの実データから取得するデータに対して、数値変換の公式を当てはめることで、目的の値としてデータ登録できます。
- 制限事項
カスタムコレクタ、PLCコレクタのみに対応

✓ 収集データの登録判定

コレクタで収集したデータに対して、登録前に条件判定を行い、条件合致するデータのみをデータ登録する機能です。

- 対象
閾値を下回る場合は、データ登録したくないなどの要求に対応できます。
- 制限事項
カスタムコレクタ、（PLCコレクタには類似機能があり）

注)
※ 両機能を指定した場合、数値変換->登録条件判定の順に処理
※ 両機能は、随時他コレクタ（センサー、CNCなど）にも機能適用予定

コレクタ > カスタム1 > データ

更新戻る

データ名
normal distribution

データ型
DOUBLE

配列数
1

サンプルレート
0

☒ 数値変換

変換後の型
DOUBLE

変換式
(\$VALUE + 10) * ABS(5)

☒ 登録条件
(\$VALUE + 10) * ABS(5) < 100

入力可能な式について

例 : (\$VALUE + 10) * ABS(5) < 100
使用可能な比較演算子: =, <>, !=, <, >, <=, >=
使用可能な算術演算子: (), +, -, *, /, %

使用可能な関数

関数名	説明
ABS(x)	絶対値（整数）
ACOS(x)	三角関数 逆余弦
ASIN(x)	三角関数 反正弦
ATAN(x)	三角関数 反正接
CEIL(x)	小数点以下切り上げ
COS(x)	三角関数 余弦
COSH(x)	双曲線余弦
EXP(x)	底がネイピア数 e である指数関数
FABS(x)	絶対値（倍精度浮動小数）
FLOOR(x)	小数点以下切り捨て
FMOD(x, y)	剰余
LOG(x)	自然対数
LOG10(x)	10を底とする対数
POW(x, y)	べき乗
ROUND(x)	小数点以下四捨五入

使用可能な変数

変数名	説明
\$VALUE	コレクタから受け取った値、もしくは、変換式による計算結果
\$PVALUE	前回DBに登録された値

使用可能な定数

定数名	説明
M_E	ネイピア数、自然対数の底(e)
M_LOG2E	log_e(e)
M_LOG10E	log_10(e)
M_LN10	log_e(2)
M_LN10	log_10(e)
M_PI	円周率（Pi）
M_PI_2	Pi/2
M_PI_4	Pi/4
M_1_Pi	1/Pi
M_2_SQRTPI	2/Pi

簡単・安価・エッジ・チョコ停監視・なんでも繋がるIoT “SpeedBee Hive”



SALTYS TER

SPEED AND SMART WITH SPIRIT

株式会社ソルティスター

〒399-0737 長野県塩尻市大門八番町1 番2 号
塩尻インキュベーションプラザ109 号室

TEL 0263-51-9296

FAX 0263-51-9297

拠点：東京 / 沖縄開発センター / 長野R&Dセンター
ホームページ <https://www.saltyster.com>

お問い合わせ先：sales@saltyster.com