



ユーザーズマニュアル

RTC: MeterRecognition

Ver. 1.0

発行日 2020年3月27日
公立大学法人会津大学
アクアクルー株式会社

1. 概要

RTC : MeterRecognition は、カメラによって取得した画像から複数のメーター領域を抽出し、結果として何 A あるかを出力するコンポーネントです。データポート(出力用)1つ、サービスポート(Provider 側)1つを持ちます。

コンポーネントは初回起動時、ランプのメーターの値判定を行うための領域、閾値設定等を行い設定ファイルを作成します。その際 GUI 環境による設定が必須となるため、注意してください。

また、サービスポートを有するためコンシューマ側から設定値をサービス越し渡すことで設定値を変更することが可能です。

2. 動作環境

動作確認済み環境:

OS: Ubuntu16.04

RTM: OpenRTM-aist-1.1.2 (Python ver.)

Python version: 2.7.10

必要なライブラリ:

¥ OpenCV 2.4.10

¥ PyQt4

¥ PIL

3. RTC 構成

3.1. モジュール名

meterRecognition

3.2. 機能概要

USB カメラを用いて撮影された盤面のメーター位置を設定すると、起動時に入力されたインターバル事に各メーターの針を認識し、OutPort Res から結果を出力します。

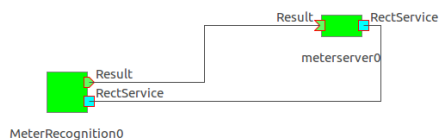
結果はメーター番号、メーター値、左上点 x 座標、y 座標の順に Short 型一次元配列に格納され、結果が出力されます。

ServicePort: rect_idl を扱う **プロバイダ側** サービスポートです。サービスの実装は同フォルダ内にある **rect_idl_example.py** で定義されています。

LampRecognition がサービス本体を import し起動時にサービスを立ち上げます。サービスはこの時点で立ち上がるため、コンシューマ側 RTC を接続する場合先にサービスへアクセスするとエラーが発生します。

また、直線検出する際に一緒に梱包されている Houhgchek.py で直線検出をお願いします。

3.3. コンポーネント図



3.4. ポート情報

3.4.1. Import

無し

3.4.2. Provider 側:OutPort(MeterRecognition)

名称	データ型	概要
ResultOut	TimedShortSeq	判定結果の出力

3.4.3 Consumer 側:InPort(MeterServer)

名称	データ型	概要
ResultOut	TimedShortSeq	判定結果の入力

3.4.4 ServicePort

IDL : rect.idl

実装は以下の IDL 定義に従います。

IDL を定義する際、配列は定義されていないため、interface で使用する際はあらかじめ定義が必要です。

```

module LampRecog
{
    typedef sequence<long> LongList; #一次元配列
    typedef sequence<float> FloatList;
    typedef sequence<LongList> WLongList; #二次元配列

    interface lampRect
    {
    
```

```
void setParameter(      in LongList n_img,
                      in WLongList n_calib,
                      in WLongList n_thresh,
                      in LongList n_rects);

LongList getSize();
WLongList getCalib();
WLongList getThresh();
LongList getRects();
short isNew();
};
};
```

--

3.4.5 Service の利用

MeterRecognition からサービスを利用するためには、起動時に定義される “self._myService” からアクセスします。本 RTC では execute ループ時、サービスに新しい設定値が来ていないか確認するため

```
if ( self._myService.isNew() > interval):
```

を実行しています。このコードでは LampRecognition 本体からサービスがもつ関数 short isNew() を呼び出しています。また rect_idl_example.py で任意のクラス内変数を宣言している場合、同様の手順で変数へ直接アクセスすることが可能です。その場合も **self._myService**.<variable_name> の形でアクセスが可能です。

又、Consumer からサービスを呼び出す場合、self._myService._ptr().setParameter() という形でサービスへのインスタンスへアクセスする。”_ptr()” は provider ではない。