



ユーザーズマニュアル

RTC:SwitchRobotController

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 :SwitchRobotController は、stepper, servo コンポーネントを同時に接続させ、コマンドを入力する事で stepper を動作させるか、servo を回すかでコントロールする。

コンポーネントは初回起動時、入力を待ちと入力したデータを受け取り出力させ、stepper、servo の回転を決定し、pusher の on, off も制御する。

2. 動作環境

動作確認済み環境:

OS: Raspbian burster

RTM: OpenRTM-aist-1.2.0 (Python ver.)

Python version: 2.7.10(servo)

Python3 version:3.6.3(stepper)

必要なライブラリ:

adafruit motor kit

3. RTC 構成

3.1. モジュール名

SwichRobotController

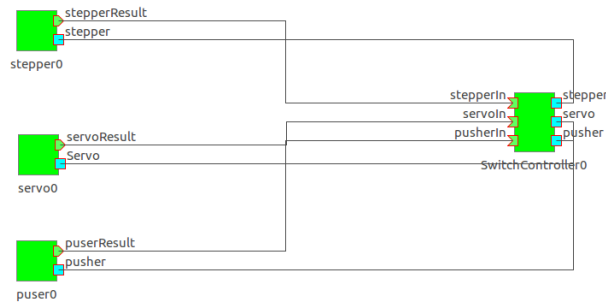
3.2. 機能概要

ServicePort: SimpleService.idl を扱うコンシューマ側サービスポートです。

Provider は SwitchController であり、各々の RTC からの出力を受け取り、stepper からのコマンドであれば、stepper が動き、servo からのコマンドであれば、servo が回る仕組みになっている。

DataPort:stepper.py servo.py

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. InPort

名称	データ型	概要
stepperIn	TimeShortSeq	Stepper の回転数
servoln	TimedshotSeq	Servo の回転数
pusherIn	TimedshotSeq	Pusher の on, off

3.4.3 OutPort

名称	データ型	概要
stepperResult	TimeShortSeq	Stepper の回転数
servoResult	TimedshortSeq	Servo の角度
pusherResult	TimedshotSeq	Puser の on, off 制御

servicePort

名称	データ型	概要
Stepper	TimeShortSeq	Stepper 動作状態の出力
Servo	TimedshotSeq	Servo 動作状態の出力
Pusher	TimedshotSeq	Pusher 動作状態の出力

ServicePort

IDL : SimpleService.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 SimpleService {
    Interface SimpleService{
        Long read (out long data);
        Long write(in long data);
        Long reset();
    };
};

```

3.4.3

Python コードについて

1. Stepper.py

①入力について stepper を動かせるステップ範囲は初期位置を 0、移動限界値を 10000 としています。つまり、移動できる範囲は(0, 10000)です。・0を入力すると、stepper の現在位置から初期位置に移動し、リセットされます。・1を入力すると、次に移動したい位置を聞かれるので、(0, 10000)の範囲で値を入力します。入力後に stepper が移動します。

・3を入力すると、プログラムが終了します。・それ以外の整数の入力をするると再入力求められます。

②プログラム仕様について

・ 値 {0(reset), 1(move), 3(exit)} の入力をする。 ・ 入力値が 0 のときポートに入力値と同じ値を送ります。

2, stepper が現在位置から初期位置 0 に移動し、リセットします。 3, stepper の現在位置を出力します。

・ 入力値が 1 のとき

1, SwitchContoroller のデータポートに入力値と同じ値を送ります。

2, 移動するステップ数 (0, 6000) を入力します。

3, stepper が現在位置から 2 で入力した位置に移動します。

4, 移動した方向と stepper の現在位置を出力します。

・ 入力値が 3 のとき

1, SwitchContoroller のデータポートに入力値と同じ値を送ります。

2, プログラムを終了します。

・ その他の入力するとき

1, 入力した値がエラーであることを出力します。

2, 再入力を求めます。

2. servo.py

・ 値 {2(move), 4(exit)} の入力をする。

・ 入力値が 2 のとき

1, SwitchContoroller のデータポートに入力値と同じ値を送ります。

2, servo が動かせる状態になります。

3, a と b と c を入力すると servo を回転させることができます。

4, Enter ~. → を入力すると servo を動かせる状態が終了します。

・ 入力値が 4 のとき

1, SwitchContoroller のデータポートに入力値と同じ値を送ります。

2, プログラムを終了します。

・ SwitchContoroller.py

① プログラムの仕様

・ stepper のデータポートの値が更新されたとき

サービスポートを使ってデータ (0, 1, 3) を取得し、stepper の動作状態を出力します。

・ servo のデータポートの値が更新されたとき サービスポートをを使ってデータ (2, 4) を取得し、servo の動作状態を出力します。

4. pusher.py

-40~60 の値を入力する事で、-40 がボタン側にピンが移動し、60 で初期状態に戻る使用になっています。

・ その他 Ctrl+C でそれぞれのプログラムを強制終了することができる。

プログラムの各 struct.unpack('<H', 変数名.to_bytes(2, 'big')) の処理は、SwitchContoroller のデータポートが short 型であるため、入力値 int 型を short 型に変換し、データポートの型に合わせるために行っている

