

多種多様なシステムをRSNP通信可能にする 汎用ユニット

仕様書 Ver.2.0

芝浦工業大学 知能機械システム研究室 岡野 憲, 松田 信人

本システムをご利用予定の方には別途ソフトウェアの配布を行っています。お手数ですが、下記までご連絡ください。また、改善点などのご意見がある方も、下記までご連絡ください。

連絡先：

芝浦工業大学 機械機能工学科 知能機械システム研究室

〒135-8548 東京都江東区豊洲3-7-5

機械工学専攻 修士2年 岡野 憲 Okano Satoshi

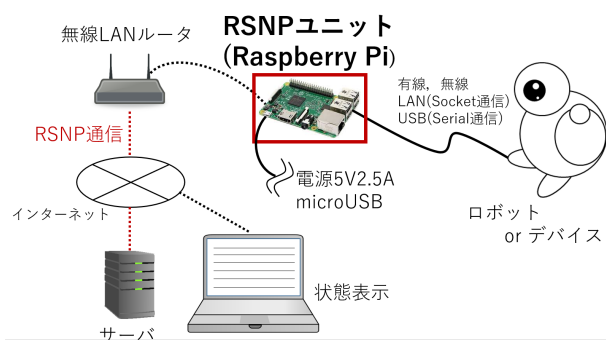
E-mail:md18020@shibaura-it.ac.jp

目次

- 1. はじめに
- 2. RSNPユニットハードウェア仕様 - ケース装着時 - バッテリ仕様 - RaspberryPi 3 Model B (RSコンポーネンツ製)
- 3. RSNPユニットソフトウェア仕様
- 4. サーバ仕様
- 4. 送信データ仕様
- 5. 通信データ仕様
- 6. 使用ソフトウェア及び入手方法, 参考資料
- 参考文献

1. はじめに

近年、労働人口の減少等から、その補完としてロボットの活用が期待されている。平成22年に発表されたNEDOの資料[1]では、ロボット市場の拡大がされている。ゆえに、今後、ロボットの台数が増加するのも必然である。そこで、増加した多数のロボットを管理、監視するためのシステムが必要になってくる。また、各ロボットからデータを取得することで、そのデータを活用した様々なサービスを期待することができる。そこで、本提案で開発した汎用ユニット(以下、「RSNPユニット」と記載)を、多種多様なロボットやデバイスに外付けで接続することで、取得したデータをRSNP(Robot Service Networking Protocol)[2]通信でインターネット経由でサーバにアップロードして蓄積し、Webブラウザ等のGUI上で各ロボットの状態を管理、監視することができる。以下の図のようにRSNPユニットをロボットに接続して使用することが可能である。



※現状、RSNPユニットは、産業技術大学院大学(品川)サーバの次のエンドポイントへ接続します。

<http://robots.aiit.ac.jp:8080/UpdateNotificationState/services>

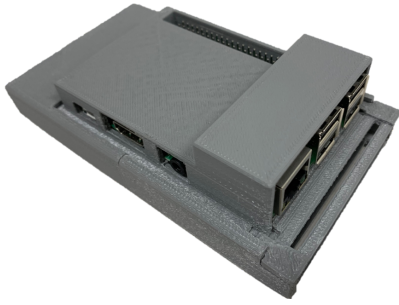
2. RSNPユニットハードウェア仕様

RSNPユニットは、マイコンとしてRSコンポーネンツ製"RaspberryPi 3 Model B"を使用している。また、バッテリーとしてTEC社製"TM-B-4K"を使用している。RSNPユニットとして、RaspberryPi、バッテリーとケースを提供しているが、javaプログラムが実行できればよいので、あくまで一例としての構成である。

ケース装着時

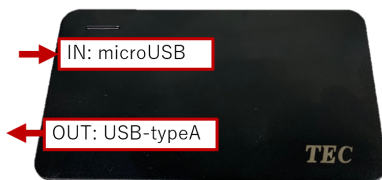
上部にRaspberryPi、下部にバッテリーを搭載し、一体型となっている。

- 寸法 : 135 x 75 x 40 mm



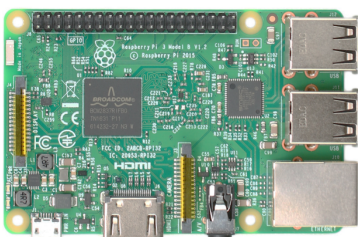
バッテリー仕様

- 容量 : 4000mAh
- 入力 : DC5V, 2A
- 出力 : DC5V, 2.5A
- 稼働時間 : 4~5時間
- 寸法 : 116 x 67 x 8 mm



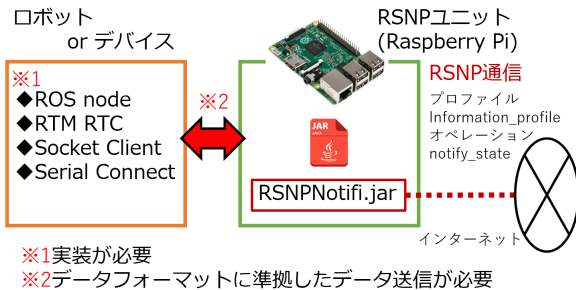
RaspberryPi 3 Model B (RSコンポーネンツ製)

- OS : Raspbian Buster
- SoC : Broadcom BCM2837 ARM Cortex-A53
- メモリ : 1 GB LPDDR2
- USB : 2.0 ポート x 4
- 映像出力 : HDMI
- 音声出力 : HDMI, 3.5 mm 4極ジャック
- ストレージ : microSDメモリーカードスロット
- ネットワーク : Ethernet 10/100 Mbps
- 無線通信 : IEEE 802.11b/g/n対応無線LAN、Bluetooth 4.1
- 低レベル周辺機器 : GPIO x 27, UART, I2C, SPI
- カメラコネクタ : 15ピン MIPIカメラシリアルインターフェース (CSI-2) コネクタ搭載
- ディスプレイコネクタ : Display Serial Interface (DSI) 15ピンフラットケーブルコネクタ
- 電源ソース : 5V/USB Micro-Bコネクタ, GPIOコネクタ
- 電源 : 5V @ 2.5A, microUSBソケット経由
- 寸法 : 85 x 56 x 17 mm



3. RSNPユニットソフトウェア仕様

RSNPユニット上ではRSNPNotifi.jarが実行されている。このアプリケーションにより、RSNPユニットが動作時に、Socket通信のサーバとRSNP通信用クライアントが立ち上がっている。ここへ、ロボットやデバイス上で立ち上げたRTMのRTC、ROSのnode、SocketのClientやSerialのクライアントを接続することで通信を行う。また、フォーマットに準拠したデータを送信することで、XML形式に変換し、プロファイルとしてInformation_profileのオペレーションとしてnotify_stateでデータ送信する。



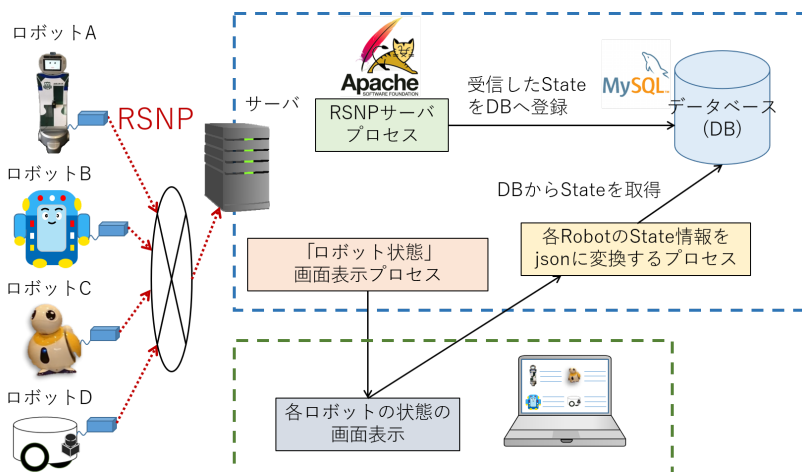
ソフトウェア要件は次の表に示すとおりである。

名称	バージョン
Linux OS	Raspbian Buster
RSNP LIB	rsnp-robot-api-2.3.0 rsnp-fjlib-api-2.3.0
Java VM	jdk-1.8.0

- 開発環境 : Windows 10, Eclipse Version Oxygen3(4.7.3)

4. サーバ仕様

サーバ内部でのプロセスは次のようになっている。各ロボットに取り付けられたRSNPユニットからnotify_stateでXML形式のデータを送信する。WebサービスフレームワークであるApach Tomcat上で、実行しているRSNPライブラリを実装したWebアプリケーションで、受信した各データをデータベースであるmySQLに登録、蓄積される。この登録されているデータから、必要なデータを取得し、json形式に変換する。ロボット画面表示用のプロセスが実行しているところへ、端末からサーバにリクエストを投げることで、json形式のデータから各ロボットの状態データをWeb表示上で見ることができる。



ソフトウェア要件は次の表に示すとおりである。

名称	バージョン
Linux OS	Cent OS 7.5
RSNP LIB	rsnp-robot-api-2.3.0 rsnp-fjlib-api-2.3.0
Java VM	jdk-1.8.0
Apach Tomcat	7.0.82
mySQL	Ver. 14.14

- 開発環境 : Windows 10, Eclipse Version 2019-03(4.11.0)

4.送信データ仕様

XML形式のデータは実際、次のようになっている。ロボットから送信するデータは複数種類あるため、これを識別するために"アクションID"を設けている。次に、"アクション名"は、ロボットから送信するデータの名前である。"(状態)結果ID"は、基本的に"アクションID"に対応する。"(状態)結果"は、ロボットの送信する具体的なデータそのものが入る。例えば、ロボットの稼働状況や、動作回数などを入れる。"時間"は、送信時のタイムスタンプが入る。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <notification>
    <states class="java.util.ArrayList">
      <status>
        <action_id>"アクションID"</action_id>
        <action>"アクション名"</action>
        <result_id>"(状態)結果ID"</result_id>
        <result>"(状態)結果"</result>
        <comment></comment>
      </status>
    </states>
    <time>"時間"</time>
  </notification>
```

5. 通信データ仕様

RSNPユニットからロボットまたはデバイス間のデータのやり取りはSocket,Serial通信で行います。ただし、以下の5種類のデータで定義づける必要があります。

- **Action_id**
- **Action名**
- **Result_id**
- **Resultデータ**
- **コメント**

各データの意味は、次のようになっています。 **Action_id**とは、**Action名**に対する紐づけidです。

Action名とは、ロボットが行った動作名などです。

Result_idとは、**Resultデータ**に対する紐づけidです。

Resultデータとは、ロボットから得たデータ(変数)などです。

コメントとは、コメント記述を入れたい場合に用います。

例えば、挨拶を3回、人数カウントを5人としたロボットがあったとします。この場合、データの仕様は次のようになります。

データ名	データ1	データ2
Action_id	1	2
Action名	挨拶回数	人数
Result_id	1	2
Resultデータ	3	5
コメント	無し	無し

ここで、実際のデータ形式は以下のようなjson形式としています。 `{...}`内において、先頭に `"data":` があり、その次に配列のカッコ `[]`内において、1種類のデータが配列の1つの要素に入ります。ダブルクォーテーション `"` で囲んだ仕様名と値をカンマ `:` で区切ります。3点 `...` には、対応するデータ等が入ります。見やすいように改行してありますが、実際は1行でデータ送信してください。これ以外の仕様でのデータを送信するとRSNPユニット側で受信できないのでご注意ください。

```
{
  "data":
  [
    {
      "ac_id": ... ,
      "ac": ... ,
      "re_id": ... ,
      "re": ... ,
      "co": ...
    },
    {...},
    ...
  ]
}
```

データ名は以下の表のように短縮形となっているのでご注意ください。

データ名	省略形
Action_id	ac_id
Action名	ac
Result_id	re_id

データ名	省略形
Resultデータ	re
コメント	co

上記のロボットの例の場合は,

```
{"data":[{"ac_id":1,"ac":"挨拶回数","re_id":1,"re":3,"co":""},{ "ac_id":2,"ac":"人数","re_id":2,"re":5,"co":""}]}
```

となります。(コメントは無しのため、空欄(""))となっている)

複数種類の場合は、配列の成分が増加し、

```
{"data":[{"...},{...},{...},...]}
```

となる。今回は5種類まで対応している。

6. 使用ソフトウェア及び入手方法, 参考資料

- RSNPライブラリ v2.3 (株式会社富士通研究所) ・rsnp-robot-api-2.3.0_r49.jar ・rsnp-robot-fjlib-2.3.0_r49.jar http://robotservices.org/rsi_spec.html (上記サイトを参考に書面またはメールで申し込みが必要)
- Apache Tomcat v7.0 <http://tomcat.apache.org/>
- Axis2 v1.7.9 <http://axis.apache.org/axis2/java/core/download.html>
- Java SE Development Kit 8 (JDK)
<http://www.oracle.com/technetwork/opensource/index.html>
- Pleiades All in One 4.7.3a(Eclipse 4.7 Oxygen 3a) <http://mergedoc.osdn.jp/>
- apache-mime4j-0.6.jar <http://james.apache.org/mime4j/>
- commons-codec-1.3.jar <http://commons.apache.org/proper/commons-codec/>
- commons-logging-1.1.1.jar <http://commons.apache.org/proper/commons-logging/>

参考文献

[1] NEDO "2035年までのロボット産業の将来市場予測",

<http://www.nedo.go.jp/content/100080673.pdf>, 最終閲覧日2019年8月20日

[2] ロボットサービスイニシアチブ, Robot Service Network Protocol2.3 仕様書 第1.0版, 2010

[3] OpenRTM-aist, <https://openrtm.org/openrtm/ja>, 最終閲覧日2019年8月20日

[4] ROS Wiki, <http://wiki.ros.org/ja>, 最終閲覧日2019年8月20日