

第31回 日本ロボット学会 学術講演会

2013年9月4日(水)～6日(金)

THE 31st
ANNUAL CONFERENCE
OF THE ROBOTICS SOCIETY OF JAPAN

講演概要集



主催：一般社団法人 日本ロボット学会

RT ルームデモシステムにおける RaspberryPi の 連続運用のための予備評価

○関山守(産総研) 鍛冶良作(産総研) 安藤慶昭(産総研) 村上青児(筑波大学)
梶谷勇(産総研) 阪口健(産総研) 神徳徹雄(産総研) 谷川民生(産総研)

1. 背景と目的

RaspberryPi財団より発売されているRaspberryPiは組み込みボードサイズの基板上にARM11プロセッサを搭載しており、Debian系LinuxであるRaspbianを始めとするARM用のLinuxが動作する低価格シングルボードコンピュータである(図1)。700MHzのクロック周波数で動作するCPUと512MバイトのRAMの他に画像・音声出力、10M/100Mbpsの有線Ethernetポート、USB2.0ポート、外部記憶用SDカードスロットといった一般のLinux用途PCが有するハードウェア資源を有している。さらに各種デバイスとのデータ授受が可能なGPIOを持っており、インターネット上では色々な応用例が報告されている。それら応用例の内容についてはUSBカメラを用いたネットワークカメラ、KinectとOpenNIを組み合わせた動作報告、OpenCVを用いた画像処理などのビジョン関係のものから、SPI, CAN, I2C等の各種デバイスのコントロールとデータ授受、SiriProxy等のソフトウェア実行など多岐にわたっている。



図1 RaspberryPi

ここで我々はRTミドルウェア[1]が動作するシステム上でのRaspberryPiの応用例とRTシステムにおけるRaspberryPi適用のガイドラインを示すことは非常に有意であると主張する。RaspberryPiの費用対効果を考えた場合、前述のような処理をRTコンポーネントで実行して安定に動作するならばシステム全体の初期導入費用を低く抑えることが可能となる。RaspberryPiによるRTシステムの連続運用における信頼性・ロバスト性の保証範囲等の検証事例を収集・分析することによってシステム運用のコストを低減する提案が可能となる。

本稿では、我々が有するRTミドルウェアをベースとしたデモシステムにおいて、実際にWindows、Linuxで実装されていたRTミドルウェア関連部分お

よびネットワークカメラやWebクライアント等についてRaspberryPiを用いて代替した事例を報告する。事例についての考察を交えてRaspberryPiによるRTシステム運用についての評価を行う。

2. RT ルームデモシステム

RTルームデモシステムは我々が有するRTミドルウェアをベースとしたデモシステムである。実験室内に住環境を作り、その環境内に温度・湿度等の環境センサや人感センサ等の各種センサ、玄関鍵や電動窓の開閉等の各種サービスを分散設置した。これらの要素の連携動作についてはシナリオ用ソフトウェアモジュールとして実装している。全ての要素はRTコンポーネント化されており、住環境内に設置されている複数のサーバ上に配置される。これらのRTコンポーネントは住環境内に設置されている様々なネットワークを経由して通信を行う(図2)。

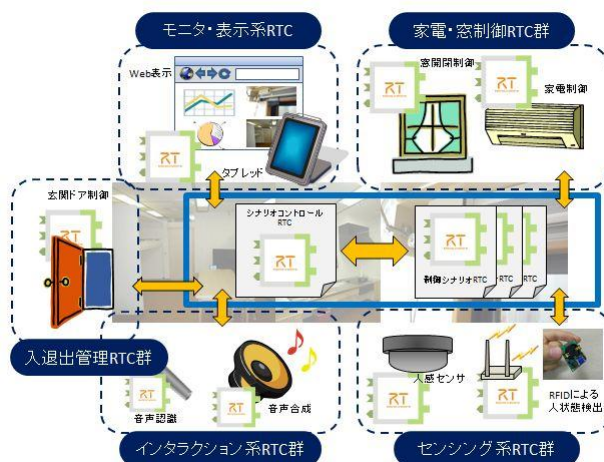


図2 RT ルームデモシステム

デモに必要なセンサ値の数値情報の提供、動作確認用の動画データのストリーミング配信などは複数サーバ上に設置されているWebサーバより行われておりRTルームデモシステムの内外を問わず産総研内部から確認する事が可能である(図3)。

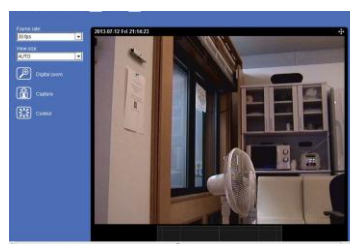


図3 デモ動作確認用動画データ配信

現在では RT ルームデモシステムを展示・開発・評価を用途として長期間連続運用を行っている。RT コンポーネントのサーバとして Windows 系 OS を一部に用いている関係から週に一度以上は全体の再起動を行っているが、一日 24 時間・年間約 350 日を超える稼働率での運用を続けている。現時点では 40 個程度の RT コンポーネントが常時稼働している(図 4)。

3. RaspberryPi による機能代替事例

ここでは上述した RT ルームデモシステムにおいて、まず RT ミドルウェアに関連した部分での代替事例をまとめて報告する。次に現時点では RT ミドルウェアとあまり関係しない部分についてもまとめて報告する。現時点で RT コンポーネント化されていないような事柄についてもいずれは RT コンポーネント化される可能性があるため、RaspberryPi の性能の限界を理解しておくことは必ずしも無駄なことではないと考える。

RT ルームデモシステムにおける代替事例を報告する前に、RaspberryPi を連続動作させる際の注意点について以下に簡単に説明する。

- RaspberryPi をケースに収納すると HDMI ケーブルの接触不良が発生しやすい。HDMI ケーブルを用いる場合は蓋を外して運用する。

- RaspberryPi の有線 LAN ポートを連続使用すると、ポートを制御している IC 上で Ethernet からの信号を検知できなくなるエラーが起きる。/root/cmdline.txt の記述修正で有線 LAN ポート通信方式を半二重に固定する事で回避可能である。

- 無線 LAN 用 WiFi ドングルを連続使用すると、1 日～3 日に一回程度の頻度で通信遮断が起きる。遮断からは 10 秒程度で復帰する。

- ftdi_sio ドライバを用いた USB シリアル変換を使用すると RaspberryPi がハングアップする。/root/cmdline.txt 内の記述を修正することでハングアップすることは無くなるが、すべての USB デバイスの通信速度が固定となり不具合が生じる。

- USB ポートに電気使用量の多いデバイスを接続すると電圧降下によるハングアップ・リポートが起きる。この場合には SD カード上のファイルシステムが破壊されることがある。

以下より代替事例を報告する。

3.1 RT ミドルウェア関連部分での代替事例

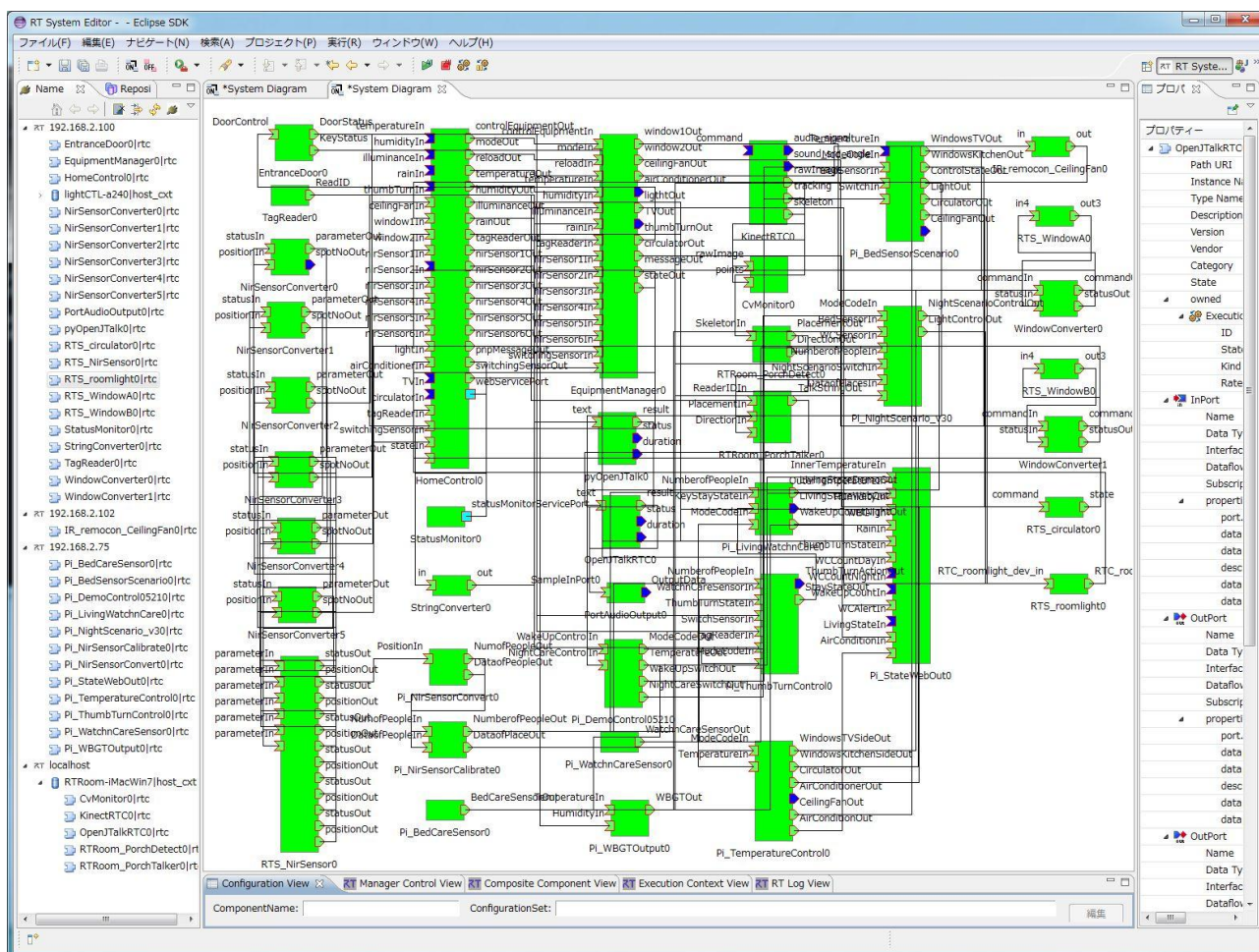


図 4 RT ルームデモシステムで常時稼働している RT コンポーネント

RaspberryPi に RT ミドルウェアを導入する場合、OpenRTM-aist の公式サイト上に RaspberryPi 用のパッケージが準備されているため apt-get を用いて簡単に導入することができる。RaspberryPi には RT ミドルウェアのコマンドラインツールである rctree, rtsprofile, rtshell を導入できるので RT コンポーネントの管理を CUI で行う事ができる。

代替事例では RaspberryPi 上にネームサーバと以下 12 個の RT コンポーネントを動作させている。

- ・エリア型赤外線人感センサによる人検出データ受信コンポーネント (実行周期: 5msec)
- ・上記コンポーネントの出力データによるエリア内人数把握コンポーネント (実行周期: 200msec)
- ・温度・湿度センサの値から WBGT 値を出力するコンポーネント (実行周期: 500msec)
- ・複数デモシナリオ間の調停, 人間からの入力を Web サーバから受信するコンポーネント (実行周期: 333msec) (図 5)

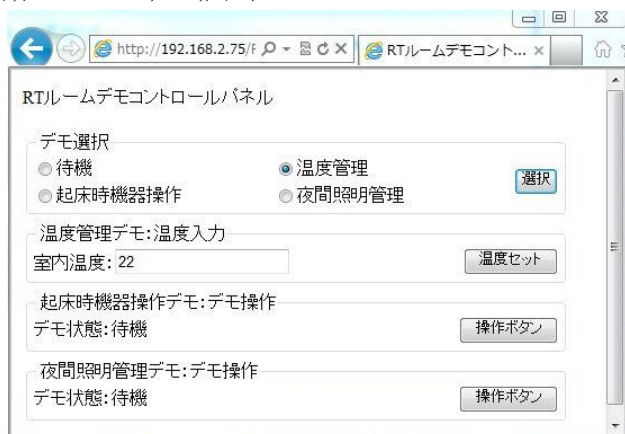


図 5 デモコントロールコンポーネントの GUI

- ・各種センサ値, トイレ転倒のアラート画面を Web 出力するコンポーネント (実行周期: 1sec) (図 6)



図 6 Web 出力コンポーネントの GUI

- ・ベッド設置 RFID 人感センサの信号をデータ出力するコンポーネント (実行周期: 1sec)
- ・トイレ設置 RFID 人感センサの信号をデータ出力するコンポーネント (実行周期: 1sec)
- ・ドアの鍵, 開閉状態, 各種人感センサ, Felica 用

RF タグリーダの値から施錠等を行うセキュリティシナリオコンポーネント (実行周期: 1sec)

- ・ドアの鍵, 各種人感センサ, Felica 用 RF タグリーダの値から人間の在室状況, 夜間起床回数等のデータを出力するコンポーネント (実行周期: 500msec)
- ・温度センサ値から, 窓・空調用家電機器を操作する空調シナリオコンポーネント (実行周期: 1sec)
- ・人感センサ値から起床時の電灯, 機器等の操作を行うシナリオコンポーネント (実行周期: 333msec)
- ・各種人感センサ値から夜間転倒予防と電灯の点消灯を行う見守りシナリオコンポーネント (実行周期: 125msec)

また Web サーバを介したデータ入出力を行うコンポーネント用に Web サーバとして Apache2 が動作している。Web サーバと RT コンポーネントの間のデータ授受は Form と PHP を用いて行われる。

登録用ネームサーバとこれらの RT コンポーネントは起動時に実行されるシェルスクリプト内で起動される。またそのシェルスクリプト内で rtshell のコマンドを用いて自動的に結線と活性化が行われるようになっている。

我々はこれらの RT コンポーネントを Windows 上で動作する RTC ビルダーを用いて生成した。その後 Windows 用のエディター上でコーディングを行い, RaspberryPi 上でビルドした。一部の RT コンポーネントについては Windows 上の RT コンポーネントをそのままビルドした。温度センサ値を用いる空調シナリオコンポーネントのコードは Windows 用状態遷移モデル設計支援ツール ZIPC[2]が出力した C 言語のコードをそのまま用いている。

上記のとおり動作周期が各々異なる RT コンポーネント 12 個, Web サーバ Apache2, ネームサーバが同時に動作している状態での RaspberryPi の CPU 負荷は最大で 30%程度であった。

これらの結果をふまえると RaspberryPi の RT コンポーネントサーバとしての能力は低くないので, 今後は画像処理のような重い処理を RaspberryPi 上で RT コンポーネント化して動作を確認する事を考えている。現在は OpenCV, OpenNI, SiriProxy の RaspberryPi 上での動作検証と RT コンポーネント化を進めている。また RaspberryPi が持つ GPIO を活用して各種デバイスの RT コンポーネント化の検討も行っている。現在は I2C デバイスセンサや XBee モジュール及び SPI-CAN 変換チップの動作検証と RT コンポーネント化を行っている。

3.2 それ以外での代替事例

RT ルームデモシステムではセンサ値の詳細な数値情報やデモ動作確認用動画データの配信を Web サーバから行っており, ネットワークを通してデモの実行と内容の確認が可能である。今回は以下二つの事項について RaspberryPi による代替を検証した。

一つは, デモ動作確認のための動画配信部分についての代替である。RT ルームデモシステムでは動画

配信はネットワークカメラ等を用いて行っている。この部分を RaspberryPi+USB カメラで代用する事を考える (図 7)。

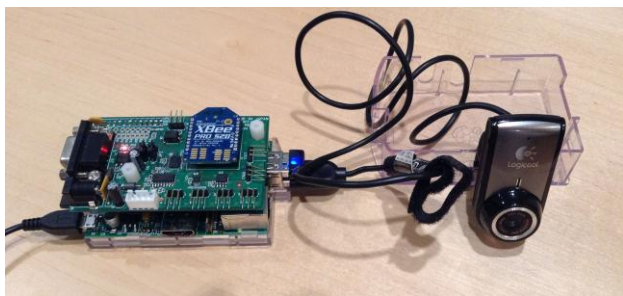


図 7 RaspberryPi+USB カメラ

RaspberryPi に mjpg-streamer を導入して実際にどのように動作するかを確認した(図 8)



図 8 RaspberryPi での mjpg-streamer 出力

確認の際には mjpg-streamer 以外のタスクを極力なくした状態で行った。フレームレートは最大でも 10fps を超えなかったが、用途と環境のバランスを取ることによって動画配信処理においては十分活用可能であると判断した。

次に、人感センサ値の Web ブラウザによる表示とリモートデスクトップによる動画確認である (図 9)。

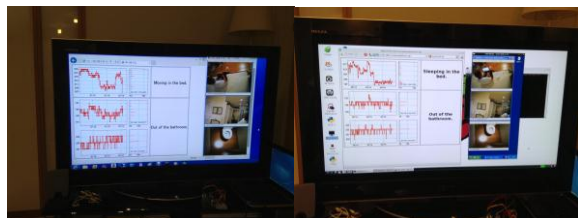


図 9 左: Windows8 右 RaspberryPi
同一デモ画面出力比較

Web ブラウザには RaspberryPi にデフォルトで導入されている midori を用いた。リモートデスクトップクライアントについては rdesktop を apt-get でインス

トールした。実際に動作させてみたところ、Web クライアントは確実に動作していることを確認できた。rdesktop については、画面変化に対する追従が悪く操作が重い。デモ用として Windows の画面を表示させるだけ、という用途であればかろうじて使うことが可能というレベルである。

これら二つのような重い処理であっても動作する RaspberryPi は使い勝手のよいシングルボードコンピュータであると我々は結論付ける。

4. まとめ

RaspberryPi は低価格 Linux シングルボードコンピュータであり、色々な応用が報告されている。RaspberryPi を RT システムに組み込むことで得られるアドバンテージが大きいと考え、RT ルームデモシステム内で RaspberryPi を評価した。

RaspberryPi 上で複数の RT コンポーネントを安定して動作させることが出来た。今後は画像処理等の重い処理や GPIO を活用した各種デバイスの RT コンポーネント化についての検討を進める。

他に動画出力と Web ブラウザ、リモートデスクトップの動作検証と評価を行った。リモートデスクトップの動作では実用は難しいレベルの動作結果しか得られなかったが、それ以外の用途においては実用レベルの性能を有しているという感触を得た。

連続運用を考えた場合には各ポートでの用法に特定のケアが必要等の問題点もあるが、費用対効果を考えた場合には非常に大きいアドバンテージがあるため今後も様々な用途への導入と動作検証を通して評価を続けていきたい。

謝 辞

本研究の一部は、NEDO 基盤ロボット技術活用型オープンイノベーション促進プロジェクト、NEDO 平成 20 年度産業技術研究助成事業、厚生労働省障害者自立支援機器等研究開発プロジェクトによる成果を利用して実施した。

参 考 文 献

- [1] Noriaki ANDO, Takashi SUEHIRO, Kosei KITAGAKI, Tetsuo KOTOKU, Woo-Keun Yoon, "RT-Middleware: Distributed Component Middleware for RT (Robot Technology)", 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2005), pp.3555-3560, 2005.08, Edmonton, Canada
- [2] “状態遷移表設計で品質向上 キャッツ ZIPC”, <http://www.zipc.com/>