

Arduino を使って OpenRTM-aist 対応 組み込みシステムを簡単に作るためのライブラリ「RTno」の開発

Development of “RTno” for development of embedded system using Arduino and OpenRTM-aist

菅佑樹 (リバスト)

Yuki Suga, Revast Co., Ltd., ysuga@ysuga.net

Abstract—A development tool for embedded system using OpenRTM-aist is shown. RT middleware is an standard platform software in which robotic elemental devices such as actuators, sensors, and so on, are treated as ‘Components’ and those components allows users to re-use them. We developed “RTno” to make it easier to make original embedded system like controllers, sensor network, etc. In this paper, the first version of the RTno is shown in detail.

Key Words: RT Middleware, embedded system, OpenRTM-aist

1. はじめに

近年、ロボット技術の広範な分野への応用が期待されている。一方で我々は、国内外の研究機関がそれぞれに異なるアーキテクチャの上でロボット開発を行っているため、研究成果の検証や再利用が難しいことが業界全体の課題であると考えている。

これに対し、再利用可能なコンポーネント (RT Component, RTC) として開発するロボット開発用基盤ソフトウェア規格として、産業技術総合研究所などが中心となってまとめた、いわゆる「RT ミドルウェア (RT middleware, RTM)」が提案されており、その実装として産業技術総合研究所が開発している OpenRTM-aist や、株式会社セックが開発している OpenRTM.NET などがある [1, 2]。本研究では、現状もっとも利用者が多いと思われ、また複数の言語から利用可能な OpenRTM-aist を積極的に利用している。

この RTM を用いたシステム (以下、RT システム) では、RTC の持つデータポートおよびサービスポートを接続し、データの入出力関係を定義することで、ブロック線図のように明快にシステム内のデータの流れを定義する。また、それぞれの RTC は active, inactive, error, および created の「状態 (state)」を持っており、さらにパラメータを調整するコンフィギュレーション (configuration) の機能を持っている。これらの状態変数をツール等から制御することで、RT システムの振る舞いを開始、停止、および調整することができる。

近年では、カメラやレーザーレンジセンサなどのセンサ系の RTC を中心に、各プロジェクトの成果物が公開され、利用可能となっている (たとえば [3, 4, 5])。しかしながら、RTC 対応製品の入手性や RTC 自体の公開が遅れている事、RTC の設計が複雑で運用性が低いことなどが原因で、現状の RTC 対応製品でロボットシステムを構築するのは未だに難しい。

そこで本研究では、研究者や学生および個人を対象として、Linux や Windows などの OS で動作する PC から利用でき、シンプルで安価ながら RTC 対応のロボット要素を最小の手間で構築する環境を提供することを目的として、市販の組み込みボード用 RTC 対応ライブラリ「RTno (アールティノ)」の開発を行ったので報告する。

2. 従来の RTC 対応組み込み機器開発

通常、組み込み機器を OpenRTM-aist の RTC 対応機器として開発する場合、以下の 3 つの方法が考えられる。

1. 組み込み機器と RTC を別々に開発
2. 汎用 I/O ボードと対応 RTC を使い、ハードのみ開発

3. 組み込み機器用ミドルウェアとブリッジによる接続

1. の方法では、組み込み機器用のソフトウェアを開発し、それを利用するための PC 上での動作を行う RTC を開発する方法である。これは手間がかかる分、システム全体の設計に関する自由度が高い。

2. の方法では、汎用 I/O ボード専用の RTC をあらかじめ用意しておき、バイト型のデータポートを利用して I/O に直接指令を送る方法である。これはソフトウェアの開発が不要となり、最小の手間で RTC から I/O を操作できるが、RTC のデータポートがバイト型などに限られる点や、データポートを使ってリアルタイム制御が必要となることがあり、リアルタイム制御が可能な OpenRTM-aist を使うための OS が限られている現状では現実的ではない。

最後に 3. は、miniRTC などの軽量の組み込みシステム用 RTM を使ったシステム構築である [6]。miniRTC では CAN バスや Zigbee ネットワークを使って組み込み機器間の通信が可能となっているが、PC 上の OpenRTM-aist と通信する場合は、専用のブリッジ機器が必要な点や、マネージャに相当する組み込み機器が必要な点から、開発するシステムの規模が大きくなる傾向がある。

3. RTno とは

RTno は、市販の組み込みマイコンボードの Arduino (アルディーノ) [7] を対象とし、Arduino で開発したコントローラやセンサ機器を簡単に OpenRTM-aist 対応の RTC から制御するためのライブラリ群である [8]。

RTno では、組み込みボード上で専用ライブラリを使ったソフトウェア開発を行うのみで、所望のデータポートを備えた RTC の開発を行う。PC との通信は RTno のライブラリでカプセル化されており、所望のデータポートを組み込みソフトウェア上で実装すると、PC 側の RTC が自動的に RTC のデータポートを追加することが出来る。これにより開発者は、組み込み機器のハードウェア開発と、モータドライバや、センシングデバイスに対応した組み込みソフトウェアを開発すると、自動的に PC 側の RTC を開発したことになる。以降は RTno の実装方法について詳しく述べる。

3.1 RTno の実装方法

上述したように、RTno は組み込みボード用の専用ライブラリと PC 上で動作するブリッジ RTC の組合せによって動作する。図 1 に RTno の実装について示す。

RTno を使った RTC 対応組み込みシステム開発では、開発者は PC との通信機能を有したマイコン上で RTno、および Arduino ラ

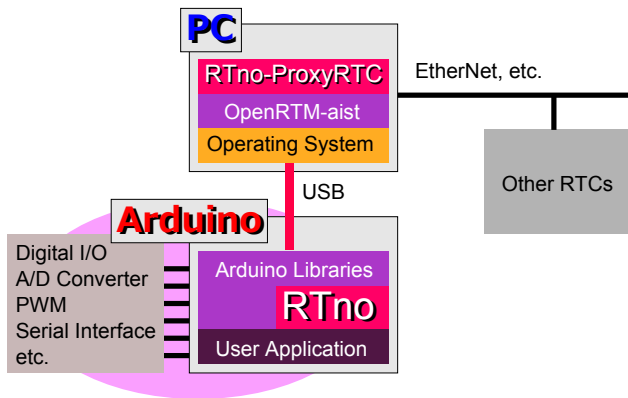


Fig.1 RTno system diagram

イブライリを使い、独自の組み込みソフトウェアの開発を行う。このとき、組み込みソフトウェア上で「addOutPort」のようなデータポート登録関数を呼び出すと、デバイスが接続された PC 上で実行されたブリッジ RTC が情報を収集し、データポートなどの構造を自動的に変更する。

また、RTC の状態変更に伴う on_activated や on_execute コールバック関数のフレームワークを実装しており、通常の RTC 開発と同様に、テンプレートを変更するだけで RTC のフレームワークを利用することが出来る。

ブリッジ RTC では、OpenRTM-aist によって、Operation System が隠ぺいされているため、同じコードを使って Windows および Linux での運用が可能になっている。

3.2 RTno の対象とするシステム

RTno は組み込みマイコンボードの Arduino での運用を前提としている。Arduino は安価でありながら、プログラムのコンパイルと書き込みが可能な統合環境を備えており、サーボモータなどを動作させるためのライブラリが豊富である。また、統合環境自体を含めたソフトウェアのみでなく、ハードウェアの設計図もオープンソースとなっており、様々な会社から Arduino を模した組み込みボードが発売されている。

本研究では、ボード自体の入手性や開発に役立つライブラリの情報が豊富である点から Arduino を最初のターゲットとしたが、RTno の仕組み自体は他のボードでも利用可能であるため、今後は対応ボードを増やす予定である。現行バージョンでの RTno は、PC と組み込みボードとの 1 対 1 の通信を前提としており、Arduino ではハードウェア層での通信は UART を利用しているが、EtherNet や CAN, Zigbee などの応用も可能である。

3.3 RTno で提供される機能

上述のように RTno を使えば、テンプレートを変更するのみでの RTC 対応デバイスの開発が可能である。以下に簡単なデジタル I/O を制御する RTno のコードを示す。

```
#include <RTno.h>

TimedLongSeq in0;
InPort in0In("in0", in0);

TimedLongSeq out0;
OutPort out0Out("out0", out0);

int RTno::onInitialize() {
    /* Data Ports are added in this section. */
    addInPort(in0In);
    addOutPort(out0Out);

    // Some initialization (like port direction setting)
    for(int i = 0; i < 6; i++) {
        pinMode(2+i, INPUT);
    }
    for(int i = 0; i < 6; i++) {
        pinMode(8+i, OUTPUT);
    }
}
```

```
}
return RTC_OK;
}

int RTno::onActivated() {
    // Write here initialization code.

    return RTC_OK;
}

int RTno::onDeactivated() {
    // Write here finalization code.

    return RTC_OK;
}

int RTno::onExecute() {
    /* Input digital data */
    if(in0In.isNew()) {
        in0In.read();
        for(int i = 0; i < in0.data.length() && i < 6; i++) {
            digitalWrite(8+i, in0.data[i]);
        }
    }

    /* Output digital data */
    out0.data.length(6);
    for(int i = 0; i < 6; i++) {
        out0.data[i] = digitalRead(2+i);
    }
    out0Out.write();

    return RTC_OK;
}
```

Arduino 開発環境で用いるプログラミング言語は Processing と呼ばれる C++ のサブセットに近い言語で、リストに示したように、C++ による RTC 開発とほぼ同じコードで実装可能である。このような点から、本研究成果は、RTC 開発を経験していながら、組み込みマイコンを使った経験が少ない情報系のユーザ、ならびに Arduino を代表としたマイコン開発を得意とする電子系ユーザへの RTM 入門の窓口に適していると考えられる。

4. まとめと今後の展望

本稿では、組み込みマイコンを対象とした RTC 開発支援ライブラリ群の RTno を紹介した。RTno は、RTno 使ってマイコン用の組み込みソフトウェアを開発し、それを PC に接続することで、ブリッジ RTC を介して他の RTC とデータを送受信することを可能にする。また、ブリッジ RTC は開発するシステム毎に再設計する必要が無いため、RTC 対応の組み込みシステム開発の手間を軽減することが出来る。さらには、RTno を使ったコーディングは、通常の C++ 版 RTC 開発と何ら変わらないので、組み込みマイコン技術者の RTC 開発入門や、PC 上の RTC 開発経験者の I/O 制御用組み込みシステム開発に適していると考えられる。

参考文献

- [1] OpenRTM-aist Official Website: <http://www.is.aist.go.jp/rt/OpenRTM-aist/>
- [2] 長瀬雅之, 中本啓之, 池添明弘: "はじめてのコンポーネント指向ロボットアプリケーション開発 RT ミドルウェア超入門", 株式会社 毎日コミュニケーションズ, 2008 年
- [3] OpenHRI Official Website: <http://openhri.net/>
- [4] OpenINVENT: <http://www.openrtmp.jp/INVENT/>
- [5] RT middleware Contest: <http://openrtm.sakura.ne.jp/cgi-bin/wiki/wiki.cgi>
- [6] 大原 賢一, 米坂 真司, 田窪 朋仁, 前 泰志, 新井 健生: "RTC-Lite を利用した無線通信に基づくホームネットワークシステム構築", 第 11 回計測自動制御学会 システムインテグレーション部門講演会 (SI2010), IE4-4, 2010 年
- [7] Arduino Official Website: <http://www.arduino.cc/>
- [8] RTno official Website: <http://ysuga.net/robot/rtm/rtc/rtno>