

CONFERENCE DIGEST

ロボティクス・メカトロニクス講演会2009
2009 JSME Conference on Robotics and Mechatronics

ROBOMECH 2009 in FUKUOKA

豊かな暮らしを創生するロボティクス・メカトロニクス
Robotics and Mechatronics for Creating an Affluent Society



太宰府天満宮本殿

Sun. 24th ~ Tue. 26th May, 2009

Fukuoka International Congress Center
Fukuoka, Japan

主催 社団法人 日本機械学会 ロボティクス・メカトロニクス部門
The Japan Society of Mechanical Engineers, Robotics and Mechatronics Division



μITRON ベース組み込みシステムのための RT ミドルウェア

RT-Middleware for Embedded Systems based on μITRON Operating System

正 安藤 慶昭 (産総研) 原 功 (産総研) 正 大場 光太郎 (産総研)

Noriaki ANDO, National Institute of Advanced Industrial Science and Technology, n-ando@aist.go.jp

Isao HARA, National Institute of Advanced Industrial Science and Technology

Kotaro OHBA, National Institute of Advanced Industrial Science and Technology

Conventional RT-Middleware (RTM) / RT-Components (RTC) require PC-level computational capacity and memory. To utilize RTC on embedded CPU based devices with other generic RTCs, the RTC-Lite component, which depends on a full-spec proxy component running on a PC, is necessary. In this paper, RT-Middleware (OpenRTM-aist) is ported into μITRON OS for embedded systems, and its new architecture for embedded systems is proposed. Experimental results and performance evaluation are also shown.

Key Words: RT-Middleware, embedded system, μITRON

1. はじめに

ロボット等に搭載される種々の分散デバイス, 知能化空間における分散センサ等は, 價格的制約等から, 一般にリソースの乏しいマイクロコントローラ等によって制御される. この様なデバイスを, RT コンポーネント (RTC) 化する方法として, 組み込みデバイス用のコンポーネントフレームワーク “RTC-Lite [1]” がある. RTC-Lite は, PIC (Peripheral Interface Controller)¹ や H8²等の省資源 CPU 上で OS なしでも動作するものの, PC 上のプロキシコンポーネントを必要とし, 他の RTC との通信の全てをプロキシコンポーネント経由で行わなければならない.

モジュール化の本来の目的は, 機能要素の独立性, 自立性を高めるモジュールの再利用性を向上させる点にある. 従って, 組み込み機器単位で独立した RT コンポーネントがそのまま動作することが望ましい.

本稿では, MMU (Memory Management Unit) を搭載しない CPU でも動作可能な μITRON 系 OS 上で, RT ミドルウェア (OpenRTM-aist) を動作させる手法を示すとともに, 通信速度に関する実験および評価を行い, 組み込み機器に対する RT コンポーネントの適用可能性について議論する.

2. 組み込み CPU 用 RT ミドルウェア

OS は実行するプログラムのメモリ空間の管理方法から, スレッド型とプロセス型の 2 つに大別される. 組み込みシステムでは, スレッド型 OS を使用するか, あるいは OS 自体を使用しないことが多い. スレッド型 OS (例: μITRON, uCLinux, VxWorks 等) では通常 MMU を使用する必要がないため, 安価な MMU を持たない CPU でも動作する. 一方, プロセス型 OS (例: Linux, Windows, POSIX 系 OS) は MMU を必要とするため, 比較的高機能な CPU 上でのみ動作可能なものが多い.

従来の OpenRTM-aist は, Linux や Windows などプロセス型 OS を前提としていたが, より低機能・安価なデバイスに RTC を搭載するには, MMU を必要としないスレッド型 OS 上でも動作する必要がある. そこで, 代表的スレッド型組み込み OS である μITRON への OpenRTM-aist 移植を行う.

2.1 TOPPERS

TOPPERS (Toyohashi OPen Platform for Embedded Real-time Systems) は, NPO 法人 TOPPERS プロジェクト (リーダー: 高田広章) を中心に開発がおこなわれている μITRON4 仕様準拠のリアルタイム OS である [2].

使用する μITRON 仕様 OS として, オープンソースであり, 製品への採用実績も多数ある, TOPPERS/ASP (Advanced

¹ マイクロチップ・テクノロジー社製マイクロコントローラ

² 日立製作所 (現ルネサステクノロジ) 製マイクロプロセッサ

Standard Profile) カーネルを選択した. また, ネットワークスタックには ITRON TCP/IP API 仕様に準拠した TINET を使用した.

2.2 CPU ボード

TOPPERS を搭載する CPU ボードとして, ネットワークインターフェースを持つ, Armadillo210 (アットマークテクノ製) (図 1, 表 1) を使用する. なお, 搭載 CPU である ARM9 (ARM920T) は MMU を持つ CPU であるが, TOPPERS 自体は MMU を使用しない.

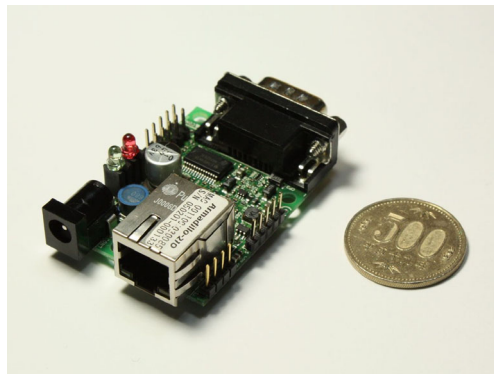


Fig.1 Appearance of target CPU board (Armadillo 210)

Table 1 Specification of target CPU board (Armadillo210)

Armadillo 210	
プロセッサ	EP9307 (Cirrus Logic 社製)
CPU コア	ARM920T (200MHz)
SDRAM	32MB
フラッシュメモリ	4MB
Ethernet	10BASE-T/100BASE-TX
シリアルポート	2 ポート
汎用入出力 (GPIO)	8 ビット
基板サイズ	37.5 x 50.0 [mm]
消費電力	1.2W (Typ.)

3. TOPPERS 版 OpenRTM-aist

本節では, TOPPERS 版 OpenRTM-aist のアーキテクチャについて議論する.

3.1 RtORB

OpenRTM-aist では, いくつかの CORBA 実装 (omniORB, TAO, MICO) が利用可能である. しかしながら, いずれの CORBA 実装も Linux や Windows 等の非組み込み OS を前提とした設計のため, 組み込み OS 上で動作させるのは困難である.

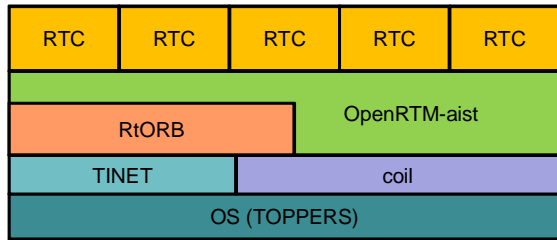


Fig.2 Architecture of OpenRTM-aist on TOPPERS

そこで、RtORB (Robotic Technology Object Request Broker) を新たに実装しこれを採用した [3]。RtORB は ORB のコア部分が C 言語で記述されており、スレッド、セマフォ/ミューテックス、メモリ使用量が非常に少ないという特徴がある³。C++でラップされたバージョンも存在し、C++で記述された OpenRTM-aist から也可以使用できるようになっている。

3.2 coil

coil (Common Operating-system Infrastructure Layer) は、OS 依存部分を吸収するために、OpenRTM-aist-1.0 用に新たに開発されたライブラリである。これまで、OpenRTM-aist で使用していた OS 抽象化ライブラリ ACE (Adaptive Communication Environment) は非常に多くの種類の OS に対応しているものの、TRON 系の OS は非対応である。また、機能が非常に豊富なためライブラリサイズが大きい上、OS 毎のソースコードが一体となっているため、新たな OS への拡張を施すのが非常に困難である。

以上の理由から、OpenRTM-aist に必要な機能のみを抽出した OS 抽象化層 coil を新たに開発した。

3.3 アーキテクチャ

以上、OS として TOPPERS、OS 抽象化層として coil、CORBA 実装として RtORB を利用した RT ミドルウェア TOPPERS 版 OpenRTM-aist を開発した。全体の構成は図 2 のようになる。

4. 性能評価

TOPPERS 版 RT コンポーネントを使用して、通信性能の性能評価を行った。

4.1 システム構成

性能評価のため、2つの CPU ボード (Armadillo210) および PC がネットワークに接続された図 3 に示すシステムを構築した。

使用した RTC は以下の 3 種類である。

SW-RTC CPU ボードのデジタル IO にプッシュスイッチを 2 個接続し、スイッチの状態を TimedLong 型の OutPort から出力する RTC

LED-RTC CPU ボードのデジタル IO に LED を 2 個接続し、TimedLong 型の InPort からの入力に応じて LED を点滅させる RTC

ConsoleIn-RTC コンソールから入力した数値を、TimedLong 型の OutPort から出力する RTC

SW-RTC および LED-RTC は Armadillo210 に搭載された TOPPERS 上で、ConsoleIn-RTC は PC 上で動作する RTC である。PC 上では、CORBA のネームサーバも同時に起動しており、3 種類の各 RTC は起動時に、このネームサーバに自身の参照を登録するように構成されている。

³Linux 上での同一 RTC での比較で約 1/2 程度 [3]。

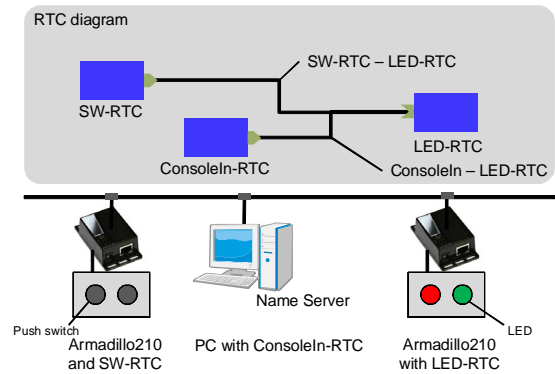


Fig.3 System configuration for evaluation

4.2 ConsoleIn-RTC 対 LED-RTC

ConsoleIn と LED-RTC を接続して、通信にかかる時間を計測した。計測にはパケット・アナライザ・ツール Wireshark を使用し、GIOP (CORBA のプロトコル) による 2 つ RTC 間の通信時間、すなわち ConsoleIn からデータが送られ、LED-RTC がデータを受け取って ACK を返しデータ転送が完了するまでの時間を計測した。

Table 2 Data transfer time from “ConsoleIn” to “RTC-LED”

最小値	最大値	平均値	標準偏差
10.26 [ms]	22.69 [ms]	15.30 [ms]	5.370 [ms]

10 回の計測を行った結果を表 2 に示す。最小 10[ms] 程度、最大 20[ms] 程度であるが、標準偏差 5.370[ms] に表れているようにばらつきが大きく、実際には約 10[ms] の場合と約 20[ms] の場合の大きく 2 つの場合に分かれている。これは、LED-RTC 側のタスク実行周期が 10[ms] であることに起因しているものと考えられる。

4.3 RTC-SW 対 RTC-LED

次に、SW-RTC のスイッチと LED-RTC の LED にオシロスコープを接続して、スイッチを押してから LED が点灯するまでの時間を計測することで通信速度の評価を行った。

計測結果は、スイッチの押下から LED 点灯まで 43[ms] であった。なお、スイッチの押下判定はチャタリング対策のために、10[ms] 間隔で 4 回の計測を行うため押下判定に 30~40[ms] かかる。したがって、実質的な通信時間は 3~13[ms] となるが、先の実験結果と総合して判断すると、TimedLong 型の通信時間は 10[ms] 程度と見積もれる。数十 ms またはそれ以下の周期でデータを取得する、分散センサのためのデバイスノードとして十分に実用的なデータ伝送速度である。

5. おわりに

本稿では、組込みシステムのための RT ミドルウェアとして、 μ ITRON 系 OS である TOPPERS、省メモリ CORBA である RtORB および、OS 抽象化層 coil を利用した OpenRTM-aist の実装例を示した。OpenRTM-aist が μ ITRON 系 OS をサポートすることにより、より省資源の CPU ボードに RTC-Lite を使用せずに RT コンポーネントを搭載することが可能となった。これにより、組込み CPU による分散デバイスの独立性がより高まり、他の RT コンポーネントとの親和性が向上した。

- [1] 安藤 慶昭, 鈴木 喬, 大原 賢一, 大場 光太郎, 谷江 和雄, “組込機器のための軽量 RT コンポーネント: RTComponent-Lite”, 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 2005 (SI2005), p.3C2-2, 2005.12, 熊本
- [2] TOPPERS プロジェクト Web ページ, <http://www.toppers.jp/index.html>
- [3] 原, 安藤, 神徳, 未廣, “軽量 CORBA RtORB による OpenRTM の実装と評価”, ROBOMECH09, 2009