

第24回 日本ロボット学会 学術講演会

The 24th Annual Conference of
the Robotics Society of Japan

講演概要集

会期 ▶ 2006年 9月14～16日

会場 ▶ 岡山大学津島キャンパス



RT ミドルウェアおよび組込みプラットフォームを用いた 移動型ロボット用制御システムの開発

土屋 裕[†] 水川 真[†] 末廣 尚士^{††} 安藤 慶昭^{††} 中本 啓之^{†††} 池添 明宏^{†††}
 († 芝浦工業大学, †† 産業技術総合研究所, ††† (株)SEC)

Development of Light-Weight RT-Component (LwRTC) on Embedded Processor

Yutaka TSUCHIYA*, Makoto MIZUKAWA*, Takashi SUEHIRO**,
 Noriaki ANDO**, Hiroyuki NAKAMOTO***, Akihiro IKEZOE***

(* Shibaura-I.T., ** Advanced Industrial Science and Technology (AIST), *** SEC Co.,Ltd.)

Abstract - Because since the RT middleware is only supported by ORBs implemented on PCs, application to a small-scale system and the mobile robot is assumed to be difficult. For reason of reducing power consumption and developing distributed RT-components, application to the embedded processing system is important to develop RT-services. In this paper, we report the LwRTC that is the implementation of the RT-component on embedded micro-processors, and the prototype system for the mobile robot using LwRTC.

Key words: RT-Middleware, Embedded Computer, Controller Area Network (CAN), Distributed Control System, Physical Agent System.

1. はじめに

産業技術総合研究所(AIST)にて開発されている RT ミドルウェア(以下 RTM)^[1]は次世代のロボット開発用ミドルウェアとして注目されてきている。RTM は RT コンポーネント(以下 RTC)と呼ばれるソフトウェアコンポーネントを複数組み合わせることでロボット制御プログラムを構築し、プログラム資産の再利用および処理の分散化が可能となっている。

本研究は、研究室にて開発している物理エージェントシステムに RTM を適用し、ロボット開発の効率化を図ることを目的としている。

2. 物理エージェントシステム(PAS)

我々が提唱している物理エージェントシステム(以下 PAS)^[2]は、ロボットが実社会においてどうあるべきかというテーマに対する 1 つの答えである。

PAS のシステム構成図を Fig. 1 に、また PAS 用に開発している物理エージェントロボット(以下 PAR)のシステム構成図を Fig. 2 に示す。

実社会では工場などと違いあらかじめ決められた動作ではなく、状況に応じた多様な動作が求められる。そのため PAR は機体を機能要素ごとにサブシステムと呼ぶモジュールに分け、それらを CAN バスで接続する形で構成されている。こうすることでサブシステムの追加、変更が容易になり、さまざまな要求にも柔軟に対応できる仕組みになっている。

また、開発の効率化や他のロボットとの協調動作などを目的として ORiN や RT ミドルウェアといったミドルウェアを積極的に取り入れ、開発、運用両面において非常に柔軟な構成となっている。

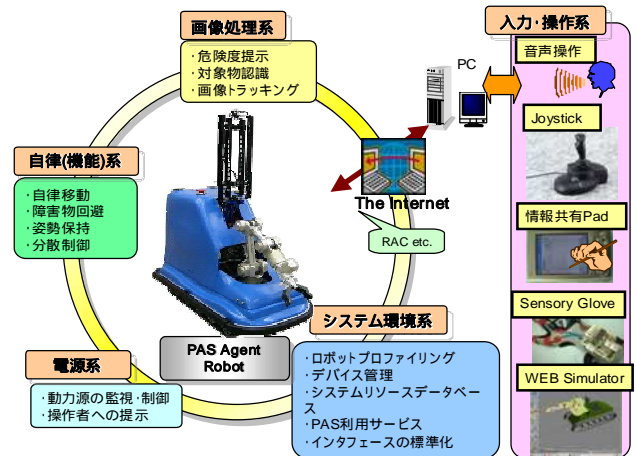


Fig. 1 Outline of Physical Agent System

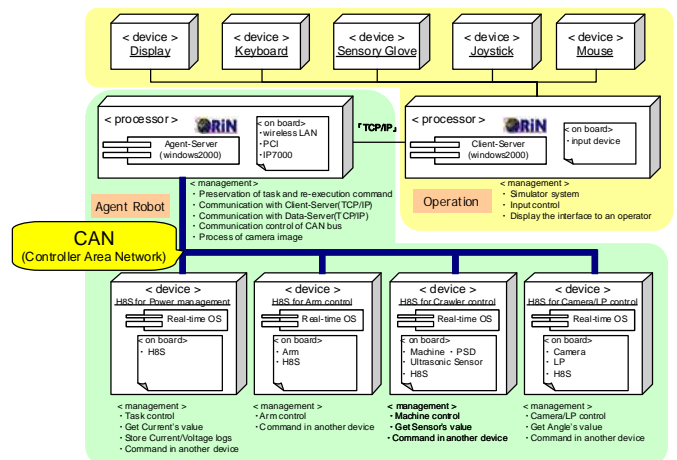


Fig. 2 Structure of Physical Agent Robot

3. 本研究のねらい

PAR04R は柔軟な構成が可能ないように作られているが、ソフトウェア的な面からは必ずしも同様のことは言えない状況である。特に複数のサブシステムを連携させ、あるまとまったサービスを提供する際の各サブシステム間の関連の設計は、大規模になればなるほど困難である。

そこでモジュール構成型ロボットの構築を、RT ミドルウェアを用いて行うことによりシステム開発コストの低減を目指す。

本研究では、PAR の構成に RTM を適用することによって柔軟な要求に対応することを目標としている。具体的な目標は、以下の3点である。

- サブシステム制御ソフトウェアの RTC 化
- 組込処理系での RTC の動作
- CAN インタフェースのサポート

この3つの目標を達成するため、我々は以下に示す RTC-CAN システムを構築した。

4. RTC-CAN システム

PAR はサブシステム間の通信に CAN を用いている。しかし RTM は Ethernet 通信のみの対応である。そのため PAR に RTM を適用するには CAN 通信を実現する必要がある。

TCP/IP の CORBA^[3]通信のみとなっている現在の産総研版の RT ミドルウェア、openRTM-aist の実装仕様に変更を加えることなく CAN デバイスを利用する仕組みを考案した。

現在、産総研では RTC-Lite という低レベル組込処理系向けの簡易版 RTC が開発されている。本研究では、この RTC-Lite の仕組みを応用して CAN 通信を実現するためのシステムを考案した。このシステムを RTC-CAN システムと呼ぶ。

RTC-CAN システムの概要図を Fig. 3 に示す。各サブシステムは CORBA が動作する程度の性能を持つ統括制御モジュールに自らのプロキシとなる RTC を配置する。CORBA を用いた外部 RTC とのデータ通信はすべてこのプロキシ RTC を通して行われる。各サブシステムには RTM で規定されている内部状態に準拠したプログラム (CAN-RTC) が配置され、プロキシ RTC 経由で受け取ったコマンドやデータに従って動作するようになっている。そして、プロキシ RTC と CAN-RTC の間を CAN にて接続することでサブシステムを RTC として見せているのである。

この仕組みでは CORBA が動作しない低レベルの組込処理系においても RTM を利用することができるというメリットがある。また、プロキシ RTC ~ CAN-RTC の接続バスを変更することでその他のバスにも対応可能である。

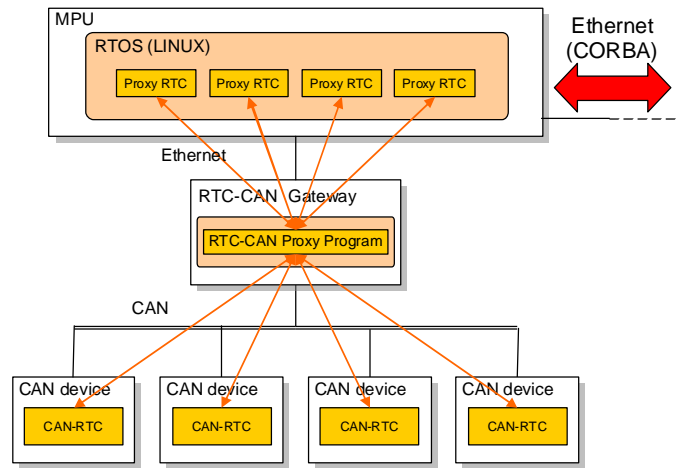


Fig. 3 RTC-CAN System Over View

4.1. サブシステムの RTC 化

サブシステムの RTC 化では RTC を、CORBA 通信を受け持つプロキシ RTC とその実装である CAN-RTC の2つに分ける。プロキシ RTC は CORBA が動作する SH4 に配置し、CAN-RTC をサブシステム制御に用いている H8S に配置する。そしてこの2つの間を CAN にて接続した。

4.2. SH4 へのプロキシ RTC の配置

サブシステムのプロキシ RTC は統括制御サブシステムに置くこととした。こうすることで整備された環境外においても外部にサーバを必要とせず、ロボット単体で柔軟な要求対応が可能となる。

RTC-Lite のシステムを用いることで特にロボット内に制御サブシステムを持たせずにロボットを構築することも可能だが、その場合は制御のための情報を常にネットワークに流すことになるため、ネットワーク負荷が著しく増大するおそれがある。

ロボット内の制御通信はロボット内で完結させることで外部ネットワークの負荷を軽減させるべきであると言える。以上からロボット内に RTC 動作可能な処理系を搭載する必要があると言える。

しかしロボットに一般的な PC を搭載することは消費電力、スペースの面から不利な面が大きい。そこで組込処理系に対し RTC を実装することとした。

統括制御サブシステムのプラットフォームとして SH4 を選択した。SH4 は Linux の動作実績があり、RTC を動作させる組込処理系として適している。

4.3. CAN 通信プロトコル

プロキシ RTC ~ CAN-RTC 間の通信プロトコルは今回、RTM の仕様にあわせて新たに規定したものを使用している。現在、データ通信、コマンド通信をサポートしている。

5. RTC-CAN テストシステム ～ Crawler 制御 ～

RTC-CANシステムの動作を確認するために、テストシステムとしてクローラを制御するシステムを構築した。

5.1. システム構成

システム構成図を Fig. 5 に示す。また、使用したハードウェア、ソフトウェアを Table 1 に示す。現在ある PAR04R のクローラサブシステムをほぼそのままの仕様で RTC 化した。

Ethernet と CAN を両方とも装備した CORBA 動作可能な組込 CPU ボードが存在しなかったため、Ethernet と CAN を変換する機能を持つハードウェア (RTC-CAN Gateway と呼ぶ) を設け、CAN 通信を実現した。

5.2. 動作実験

動作実験ではクローラの基本的動作(前進後退左右旋回)の確認、および通信時間の計測を行った。

?? に実験の様子を示す。クローラ台車上にはバッテリー、モータドライバ、クローラ制御用 H8S が搭載されている。

5.3. 実験結果

この実験によりクローラサブシステムが RTC として認識されることを確認、また基本動作(前進後退左右旋回)が可能であることを確認した。しかしデータ通信にかかる時間が最短で片道 20ms が限界であった。これはロボット制御用通信に要する時間として非常に遅い。

ゲートウェイの限界はおよそ 16ms 周期でデータ変換処理を行った時である。これは Gateway に使用している H8S の能力不足が原因であると考えられる。このとき、Gateway はデータ通信のみで処理の限界に達したため、RTC 制御コマンドの受付が不可能になった。

しかし求められる通信性能にはほど遠く、この構成におけるフィードバック制御など短周期制御の実現は困難であるといわざるを得ない。

6. 考察

実験結果により、RTC-CAN システムによって PAR04R への RTM 適用の可能性が示されたといえる。また、RTC-Lite の考え方をいれれば Ethernet 以外のバスにも対応可能であることが示された。

しかし実験によりシステムの動作は認められたものの、機器的な制約から性能面で十分とは言えない結果であった。これは Gateway に用いた CPU の能力不足によるものであることが判明している。

7. 課題

Gateway の機能不足によるシステム性能の弱点の克

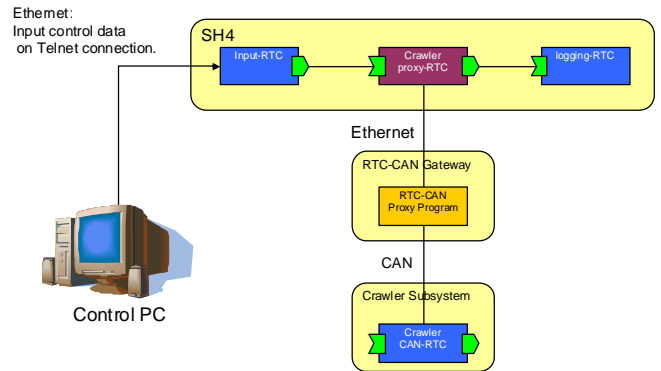


Fig. 5 RTC-CAN Test System(Crawler Control)

Table 1 Hardware List

RTC動作用SH4	ゼネラルロボティクス HRP-3P-CN
CAN I/F用H8S	北斗電子 H8S2638Fスタータキット
CANデバイス用H8S	北斗電子 H8S2636F

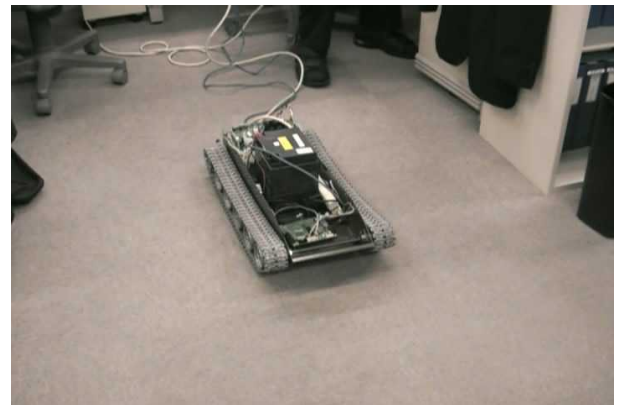


Fig. 4 Experiment of Test system

服を目指す。より短時間に通信を行うためには Gateway の機能強化が必要である。

この問題に対し私は、現在利用している Gateway を廃止し、Ethernet と CAN の両方を装備する SH4 ボードに対してこのシステムを適用することを検討中である。これによりシステム構成の簡略化、Gateway 性能不足を解消できるものと考えている。

参考文献

- [1] RTミドルウェアプロジェクト HP
<http://www.is.aist.go.jp/rt/>
- [2] 水川 真, 松原 安彦, 安藤 吉伸, 平岩 明, 町野 保: 物理エージェント(PAS)を用いた遠隔地人間協調系の基本検討 (計測自動制御学会システムインテグレーション部門学術講演論文集)
- [3] Object Management Group(OMG) HP
<http://www.omg.org/>
- [4] 安藤 慶昭(産総研), 小川秀明(慶応大メディアアートユニット h.o.主宰): RTミドルウェアのメディアアートへの応用 ~ パーソナルなコミュニケーションメディア: Small Connection ~ (ロボティクス・メカトロニクス講演会 2005)