

無線ネットワーク環境向け アラーム管理機能を有する分散リアルタイム OS*

菊池 優輔[†]横山 孝典[‡]兪 明連[§]東京都市大学[¶]

1. はじめに

近年、無線ネットワークを用いた組み込みシステムが増えつつある。組み込み制御システムの多くはタスク単位で実装されており、タスクの管理は一般にリアルタイム OS を用いて行う。そのため、無線ネットワーク環境向けの分散リアルタイム OS が求められている。分散型のシステムはノードの違いを意識することなくタスクの起動や同期を行うために、位置透過性が要求される。位置透過性のあるシステムコールを有する分散リアルタイム OS が提案されており [1], 有線 LAN である FlexRay を用いてノード間で通信を行うことでタスクの起動や同期を行っている。しかし、無線ネットワークのように通信時間が大きく変動するネットワークに適用すると時間の変動がタスク管理に影響を与える。

本研究では、無線ネットワークにおける通信時間の変動の影響を避けるため、各ノードに時刻同期機能を持たせ、絶対時刻に基づいてタスク管理を行うこととし、そのためのノード間システムコールを有する分散リアルタイム OS を提案する。具体的には、OSEK OS [2] 仕様に基づく TOPPERS/ATK1 [3] のアラーム管理システムコールを拡張して、絶対時刻に基づく分散アラーム管理が可能なリアルタイム OS を実現する。ネットワークには ZigBee を用いる。

2. 絶対時刻とアラームによるタスク管理

2.1 絶対時刻によるタスク管理

通信時間の変動が大きい無線ネットワーク環境で直接タスクを制御する場合の問題について図 1 を用いて説明する。ノード 1 からタスクの起動要求をノード 2 とノード 3 に送信しているが、ノード 2 への通信時間とノード 3 への通信時間の違いにより起動時刻にずれが生じ、タスクの制御が困難になる。

次に絶対時刻を指定してタスクを制御する場合について図 2 を用いて説明する。各ノードには時刻同期機能を持たせる。起動時刻を指定してタスクを起動するためのアラーム設定要求をノード 1 からノード 2 とノード 3 に送信する。時刻を指定することで通信時間の変動の影響を受けずにノード 2 とノード 3 で同時にタスクを起動できる。

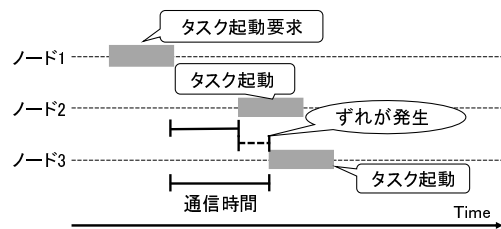


図 1: 直接タスク制御を行う場合

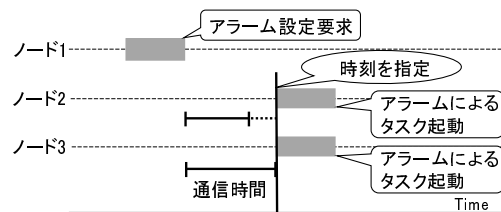


図 2: タスク制御を時刻指定で行う場合

2.2 アラーム管理システムコール

アラームは特定の時刻にタスクの起動やイベントを設定する機能である。OSEK OS が提供しているアラーム管理システムコールには、ティックの絶対値と周期を引数としてアラームの設定を行う SetAbsAlarm, ティックの相対値と周期を引数としてアラームを設定する SetRelAlarm, アラームを取り消す CancelAlarm, アラームが満了するまでのティック値を取得する GetAlarm, アラーム情報を取得する GetAlarmBase がある。

このうち SetAbsAlarm, SetRelAlarm, CancelAlarm を分散システム向けに拡張する。ただし、SetRelAlarm は時刻指定が相対時間での指定なので、それを絶対時刻に変換して SetAbsAlarm の要求として送信する。

* A distributed real-time operating system with alarm management for network environments

[†] Yusuke Kikuchi

[‡] Takanori Yokoyama

[§] Myungryun Yoo

[¶] Tokyo City University

3. ノード間アラーム管理

3.1 ノード間システムコール

ノード間のアラーム機能について、タスクを起動する場合を例に図3を用いて説明する。ノード1のタスクAがシステムコールを発行すると、分散リアルタイムOS(RTOS)は対象ノードを判定する。対象ノードが自ノードの場合はノード1でシステムコールを実行し、指定時刻にアラーム1が満了してタスクBが起動する。対象ノードが他ノードであった場合、ノード1からノード2へ ZigBee 経由で要求を送信する。ノード2側では要求を受信してシステムコールを実行し、アラームを設定して、指定時刻にタスクを起動する。

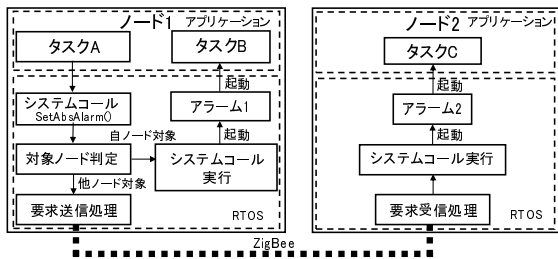


図3: ノード間システムコール処理の流れ

3.2 メッセージ送受信の流れ

ノード間の要求および戻り値の送受信の流れを図4を用いて説明する。ノード1のアプリケーションでアラーム管理システムコールを発行すると、RTOSが対象のタスクを判定し他ノード上のタスクが対象の場合は図のようにネットワーク経由要求送信処理を行いノード2に要求メッセージを送信する。要求メッセージ送信後、ノード1のタスクは待ち状態に遷移する。ノード2が要求メッセージを受信すると、それにより起動されたISR(Interrupt Service Routine)がネットワーク経由要求受信処理を実行し、メッセージから要求データを取り出してシステムコールを実行してアラームを設定する。システムコール実行後は、アラームが正常に設定されたかどうかを示す戻り値メッセージをネットワーク経由戻り値送信処理でノード1に送信する。メッセージ受信により起動されたノード1側のISRはRTOSのネットワーク経由戻り値受信処理を実行し、ノード1のタスクの待ち

状態を解除する。そして、ネットワーク経由リターン処理によりアプリケーションに復帰する。

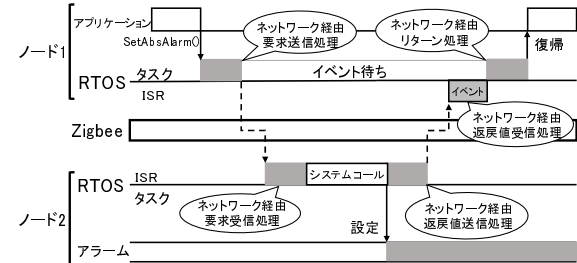


図4: ノード間におけるデータ送受信の流れ

3.3 実装

実装にはCPUとしてH8S/2638Fを搭載した評価ボードを用いる。ZigBeeデバイスにはXBEEモジュールを使用し、UARTでCPUと繋ぐ。絶対時刻を扱うための時刻同期にはGNSSによる時刻同期機構[4]を用いる。

4. おわりに

本論文では、通信時間が変動する無線ネットワーク環境において絶対時刻に基づいたタスク管理を行うことを目的に、ノード間アラーム機能によりタスク制御を行うことが可能な分散リアルタイムOSを提案した。今後の課題は、協定世界時間同期機能を用いることで、より広範囲のアプリケーションに適用可能とすることである。

謝辞

本研究で使用したTOPPERS/ATK1の開発者に感謝する。本研究はJSPS科研費JP18K11225,JP21K11815の助成を受けたものである。

参考文献

- [1] 知場貴洋, 齊藤政典, 伊丹悠一, 兪明連, 横山孝典, 位置透過性のあるシステムコールを有する組み込み制御システム向け分散リアルタイムOS, 情報処理学会論文誌, Vol.53 No.12 pp.2702-2714, 2012
- [2] OSEK/VDX, Operating System Version 2.2.3, 2005
- [3] TOPPERS Project, TOPPERS/ATK1, <http://www.toppers.jp/atk1.html>
- [4] Harayama, K., Yokoyama, T., and Yoo, M., GNSS-Based System Time Synchronization Mechanism for Cyber-Physical Systems, International Journal of Electrical and Electronic Engineering & Telecommunications, Vol.9, No.2, pp.94-99, 2020