

令和 6 年 6 月 15 日現在

機関番号：32678  
研究種目：基盤研究(C)（一般）  
研究期間：2021～2023  
課題番号：21K11815  
研究課題名（和文）メッシュ型分散処理モデルに基づくサイバーフィジカルシステム向け分散処理環境

研究課題名（英文）A Distributed Computing Environment for Cyber-Physical Systems Based on Mesh-Based Distributed Processing Models

研究代表者  
横山 孝典（Yokoyama, Takanori）  
東京都市大学・情報工学部・教授

研究者番号：60386357  
交付決定額（研究期間全体）：（直接経費） 3,000,000円

研究成果の概要（和文）：分散配置された多数のセンサ情報を利用した分散型組み込み制御システムに代表されるサイバーフィジカルシステム向けのリアルタイム分散処理環境を開発した。本分散処理環境は分散処理基盤と分散システム開発環境から成る。分散処理基盤はメッシュ型分散処理モデルに基づくもので、データの時間的整合性を保証することが可能なリアルタイムOSと分散処理ミドルウェアから成る。分散システム開発環境は分散処理基盤上で動作するアプリケーションの開発を支援するモデル変換ツール等から成る。

#### 研究成果の学術的意義や社会的意義

本研究で開発した分散処理環境は、従来の逐次型分散処理モデルに代えてメッシュ型分散処理モデルを採用するとともに、論理時間やタイムスタンプを用いることでデータの時間的整合性を保証するとともに、入力から出力までの端点間応答時間の制約を満たす動作が可能である。これにより、分散された多数のセンサ情報を収集してリアルタイム制御を行う分散制御システム等のサイバーフィジカルシステムを容易に開発可能とした。

研究成果の概要（英文）：The research has developed a real-time distributed computing environment for cyber-physical systems, such as distributed embedded control systems, which utilize distributed sensor inputs. This environment consists of a distributed computing platform and a distributed computing software development environment. The former is based on a meshed distributed computing model and comprises a real-time operating system and distributed computing middleware, which maintain temporal data integrity. The latter includes a model transformation tool that supports application development.

研究分野：組み込みシステム

キーワード：サイバーフィジカルシステム 組み込みシステム 分散処理 リアルタイム処理

### 1. 研究開始当初の背景

本研究の開発当初は、実世界のセンシングや制御を行う処理とサイバー空間での計算処理とを融合したサイバーフィジカルシステム (Cyber-Physical Systems) が増えつつあった。例えば自動車分野ではADAS (Advanced Driver-Assistance Systems) の実用化が進み、自動運転の実用化も近いと言われていた。自動車単体での制御に限らず、車々間通信や路車間通信を利用した高度な分散制御への期待も大きくなっていった。そして、より広く分散された多数のセンサ情報を収集してリアルタイム制御を行うサイバーフィジカルシステムも期待されるようになった。しかしながらそれまでのリアルタイム分散処理研究の多くは、ひとつの入力ノードからいくつかの算出ノードを経てひとつの出力ノードに至る形式の、逐次的処理を実行するモデル (逐次型分散処理モデルと呼ぶ) に基づくもので、多数のセンサ入力を扱うには十分ではなかった。

### 2. 研究の目的

本研究の目的は、広く分散された多数のセンサ情報を収集してリアルタイム制御を行う分散制御システムに代表されるサイバーフィジカルシステムを容易に実現可能な分散処理環境を提供することである。そのため従来の逐次型分散処理モデルに代えて、複数の入力ノードから複数の算出ノードを経て複数の出力ノードに至る処理を並行して実行するモデル (メッシュ型分散処理モデルと呼ぶ) を採用する。そして、メッシュ型分散処理モデルにおいて、通信時間の変動を許容しながらデータの時間的整合性を保証するとともに、入力から出力までの端点間 (End to End) 応答時間の制約を満たすことのできる分散処理基盤と、その分散処理基盤を用いた分散システムを効率よく開発するための分散システム開発環境を開発する。

### 3. 研究の方法

#### (1) メッシュ型分散処理モデル

メッシュ型分散処理の簡単な例を図1に示す。この例では、4つの入力タスク、6つの算出タスク、2つの出力タスクがメッシュ状に接続されている。制御システムでは、各タスクが周期的に処理を実行するが、1サイクルの処理は同一時刻に入力 (サンプリング) されたセンサ・データを用いる必要があるため、タスク間でやりとりするデータの時間的整合性を保証する必要がある。

ジッタ (周期のゆらぎ) のない分散制御システムを実現する方法として時間駆動アーキテクチャがあり、全てのタスクを同期して実行することでデータの時間的整合性も保証できる。しかしながら、時刻同期機構を有するとともに通信時間が変動しない時間駆動ネットワークを使用する必要がある。このため、通信時間が変動する無線通信を使用するとともに全ノードの同期実行が困難という特徴を有する、本研究が対象とするサイバーフィジカルシステムには適用できない。

そこで本研究では、メッセージにタイムスタンプを付加し、タイムスタンプを利用してデータの時間的整合性を保つ方法を提案する。これにより、通信時間の変動があるネットワークの利用が可能となるとともに、全てのタスクを同期実行することなく、データの時間的整合性を保つことができる。

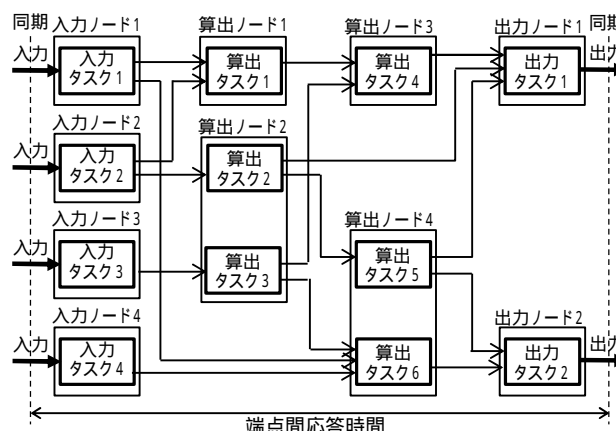


図1 メッシュ型分散処理の例

#### (2) 論理時間とタイムスタンプを導入した分散処理

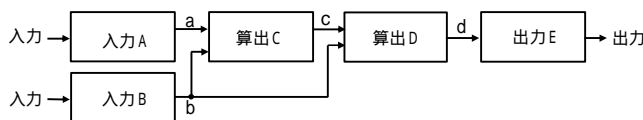
一般に制御システムはセンサからの入力処理、制御ロジックを構成する算出処理、およびアクチュエータへの出力処理から成る。図2 (a) は2つの入力処理、2つの算出処理、および1つの出力処理から成る制御ブロック図である。同図 (b) はそれらの処理をそれぞれ1つのタスクで実装し、それらを5つのノードに配置した分散制御システムの構成例である。入力タスクAは算出タスクCにメッセージaを送信し、入力タスクBは算出タスクCと算出タスクDにメッセージbを送信する。算出タスクCはメッセージaとメッセージbを受信して計算を行い、算出タスクD

にメッセージ c を送信する。算出タスクDはメッセージ b とメッセージ c を受信して計算を行い、出力タスクEにメッセージ d を送信する。出力タスクはメッセージ d を受信しアクチュエータ出力を行う。以下、このシステム構成を例に分散システムの動作方式について説明する。

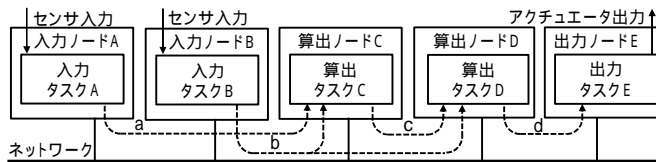
図2(b)に示したシステム構成に時間駆動アーキテクチャを適用した場合の分散システムのタイムチャートの例を図3に示す。全てのタスクは周期 T で同期実行している。時間駆動ネットワークを使用しているため通信時間の変動はなく、いずれのタスクもジッタのない動作が可能である。メッセージを送信するタスクの起動時刻とそのメッセージを受信するタスクの起動時刻の差を位相差と呼ぶ。位相差を適切に設定することで、データの時間的整合性を保つことができる。タスク X とタスク Y との位相差を X-Y 間位相差と呼ぶことにする。この例では A-C 間位相差、B-C 間位相差、C-D 間位相差、D-E 間位相差を T、A-D 間位相差を 2T とすることで、データの時間的整合性を保っている。

本研究では、メッセージにタイムスタンプを付加するとともに、仮想的な論理時間を導入することで、時間的整合性を保つ方法を提案する。時間駆動アーキテクチャと同じ制御性能を得るため、入力タスクと出力タスクのみは物理時間に同期して実行する(これを物理時間駆動と呼ぶ)。一方算出タスクは時間的整合性のあるメッセージを全て受信した時点で起動し、論理時間を用いて管理する(これを論理時間駆動と呼ぶ)。

図4に本方式に基づく分散システムのタイムチャートの例を示す。算出タスクはそれぞれ固有の論理時間を持つ。論理時間で表現した起動時刻を論理起動時刻と呼び、図中には算出タスクの起動位置に論理起動時刻を記載している。論理起動時刻は、対応する図3の時間駆動アーキテクチャにおける起動時刻



(a) 制御ブロック図



(b) システム構成

図2 複数入力分散制御システムの例

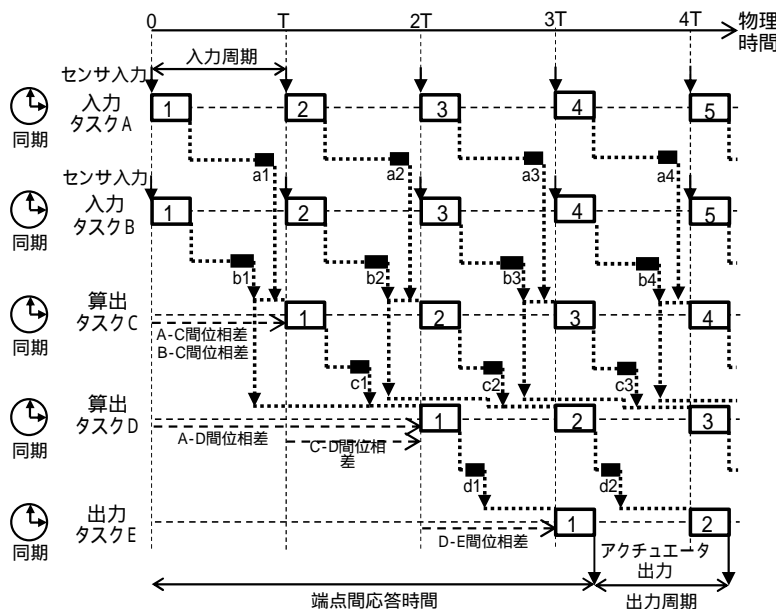


図3 時間駆動アーキテクチャに基づく分散システムのタイムチャート

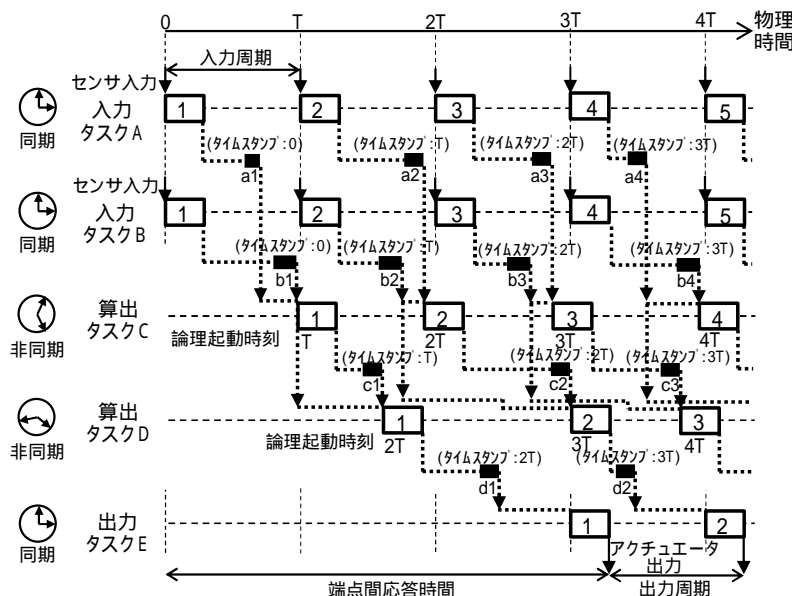


図4 タイムスタンプと論理時間を導入した分散システムのタイムチャート

と一致するように設定する。また、論理時間における位相差を論理位相差と呼び、時間駆動アーキテクチャにおける位相差と一致するように設定する。

メッセージに付加するタイムスタンプには、そのメッセージを送信するタスクの論理起動時刻を設定する。図4中には各メッセージに付加されているタイムスタンプの値を()内に記載している。タスクが受信したメッセージに付加されているタイムスタンプの時刻に論理位相差を加えた時刻がそのタスクの論理起動時刻に一致するようにしてタスクを管理する。これにより、通信時間の変動を許容しながらデータの時間的整合性を保つことができ、時間駆動アーキテクチャと等価な動作を実現できる。

### (3) 分散処理環境

図5に示すように、本研究で開発する分散処理環境は、タイムスタンプと論理時間を導入した分散処理の実現に必要な機能を提供する分散処理基盤と、それをを用いた分散システムを容易に開発するための分散システム開発環境から成る。

分散処理基盤はリアルタイムOSと分散処理ミドルウェアから成る。分散処理ミドルウェアは、論理時間を管理

する機能やタイムスタンプ付きメッセージ通信を行う機能を有する。リアルタイムOSは論理時間に基づくタスク管理が可能なスケジューリング機能と、タスク間で整合性のあるリソース共有を可能とするリソース管理機能を有する。分散システム開発支援環境の中心的なツールはモデル変換ツールである。制御設計で作成した制御モデル(Simulinkモデル)と分散システム設計で決定したタスク配置情報をモデル変換ツールに入力することで、分散制御システムの構造や振る舞いを表現したUMLモデルを生成することができる。

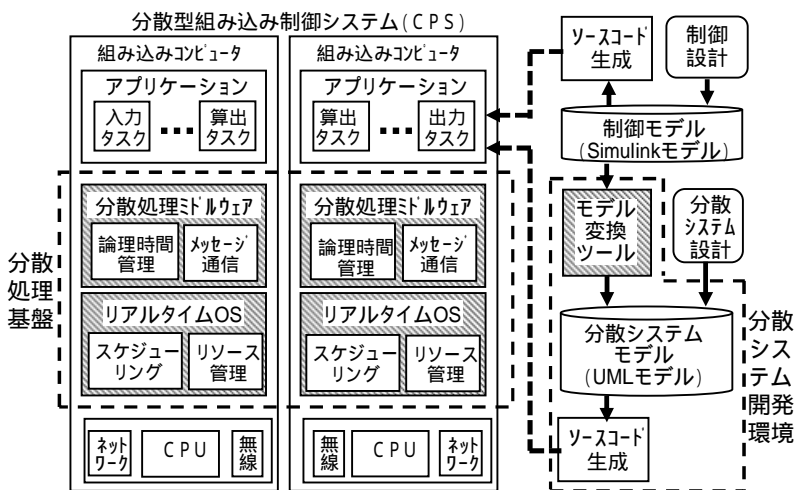


図5 分散処理環境

## 4. 研究成果

### (1) 分散処理基盤

開発した分散処理基盤のソフトウェア構成を図6に示す。分散処理ミドルウェアはイベント処理、タスク起動時処理、メッセージ受信処理、メッセージ送信処理から成る。イベント処理はメッセージ受信時に付加されたタイムスタンプに従って受信データを時系列的に記憶するとともに、リアルタイムOSに論理時間の更新を依頼する。タスク起動時処理は、アプリケーションタスクの実行前にそのタスクの論理起動時刻等の更新を行う。メッセージ受信処理は時間的整合性のあるタイムスタンプを持つ受信メッセージのデータをアプリケーションタスクに渡す。メッセージ送信処理はアプリケーションタスクより送信データを受け取り、タイムスタンプを付加したメッセージを送信する。本分散処理ミドルウェアを用いることで、アプリケーションプログラムは論理時間やタイムスタンプを意識することなく、目的とする動作を実現できる。

また、開発した分散処理ミドルウェアの性能評価を行い、実用上問題ない処理時間で実装できていることを確認した。そしてこれらの研究成果について、学会発表、学会発表、その他1件の発表

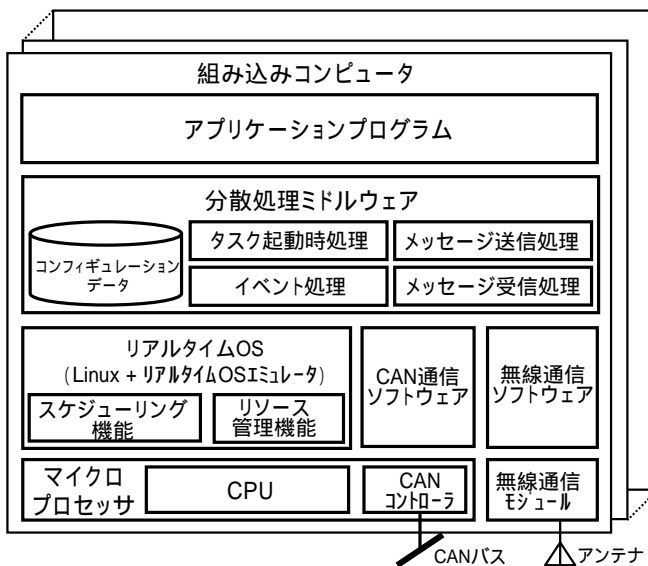


図6 分散処理基盤のソフトウェア構成

を行った。

リアルタイム OS は、自動車制御分野の標準 OS 仕様である OSEK OS や AUTOSAR OS をベースに、論理時間に基づくタスク管理を可能とするスケジューリング機能や、そのもとで動作するタスク間でリソースを共有するためのリソース管理機能を有する。スケジューリング機能は、入出力タスク向けにはジッタの小さい固定優先度スケジューリング、算出タスク向けには応答時間を短縮できる論理時間に基づく EDF (Earliest Deadline First) スケジューリングとともに、非リアルタイム (非周期) タスクを扱うための論理時間に基づく TBS (Total Bandwidth Server) スケジューリングをサポートしている。そしてリソース管理機能は、それら異なるスケジューリングのもとで動作するタスク間でも整合性のある排他制御を可能としている。CPU 上で直接動作するリアルタイム OS の他、最近増えている Linux を採用した組み込みコンピュータに対応するため、Linux 上で上記リアルタイム OS と同等の機能を提供するリアルタイム OS エミュレータも開発した。

また、開発したリアルタイム OS およびリアルタイム OS エミュレータの性能評価を行い、実用上問題ない処理時間で実装できていることを確認した。そしてそれらの研究成果について、雑誌論文、学会発表、その他 7 件の発表を行った。

## (2) 分散システム開発環境

前述のように分散システム開発環境の中心となるのはモデル変換ツールである。本ツールは制御設計で作成した Simulink モデルと分散システム設計で作成した配置図を入力し、分散システムを記述した UML モデルを出力する。配置図は UML の図のひとつで、アプリケーションを構成する各タスクを分散システム上のどのノードに配置するかを表現したものである。

図 7 を用いてモデル変換ツールの動作を説明する。まず、入力された Simulink モデルと配置図を解析して Simulink モデルデータとタスク配置データを生成する。次に、それらを参照して構造モデルデータと振る舞いモデルデータを生成する。そしてそれらをもとに、クラス図、オブジェクト図、シーケンス図、アクティビティ図から成る UML モデルを生成し、出力する。

また、実際の制御モデルを用いて本ツールの適用実験を

を行い、現実の制御システム設計に適用可能であるとの見通しを得た。そしてそれらの研究成果について、学会発表、その他 1 件の発表を行った。

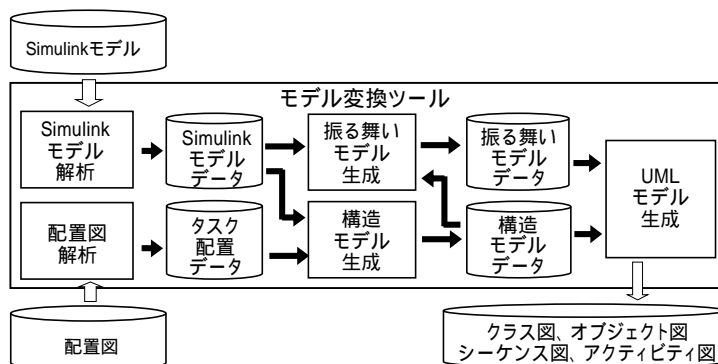


図 7 モデル変換ツールの処理の流れ

## < 引用文献 >

Shimabukuro, K., Yokoyama, T. and Yoo, M., A Logical Time-Triggered Distributed Computing Environment for Cyber-Physical Systems, 6th International Conference on Information Science and Systems, 2023

Nakajima, S., Shimabukuro, K., Takehara, E., Yokoyama, T. and Yoo, M., A Logical Time-Triggered Distributed Computing Environment for Cyber-Physical Systems with V2X Wireless Communication, 17th International Conference on Information Technology and Applications, 2023

藤島大暉、石橋航太、横山孝典、兪明連、CAN アービトレーションを用いたノード間リソース管理機能及び分散共有メモリ機能を有するリアルタイム OS、電子情報通信学会論文誌、J106-D 巻、2023、375-387

Takado, K., Yokoyama, T. and Yoo, M., A Real-Time Operating System for Physical and Logical Time-Triggered Distributed Computing, 6th Asia Pacific Information Technology Conference, 2024

Nishimura, T., Yokoyama, T. and Yoo, M., A Simulink to UML Model Transformation Tool for Distributed Embedded Control System Development, 4th International Conference on Computer Communication and Information Systems, 2024

5. 主な発表論文等

〔雑誌論文〕 計1件（うち査読付論文 1件／うち国際共著 0件／うちオープンアクセス 0件）

1. 著者名 藤島大暉、石橋航太、横山孝典、兪明連	4. 巻 J106-D
2. 論文標題 CANアービトレーションを用いたノード間リソース管理機能及び分散共有メモリ機能を有するリアルタイムOS	5. 発行年 2023年
3. 雑誌名 電子情報通信学会論文誌	6. 最初と最後の頁 375-387
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

〔学会発表〕 計13件（うち招待講演 0件／うち国際学会 4件）

1. 発表者名 Shimabukuro, K., Yokoyama, T., Yoo, M.
2. 発表標題 A Logical Time-Triggered Distributed Computing Environment for Cyber-Physical Systems
3. 学会等名 6th International Conference on Information Science and Systems（国際学会）
4. 発表年 2023年

1. 発表者名 Nakajima, S., Shimabukuro, K., Takehara, E., Yokoyama, T., Yoo, M.
2. 発表標題 A Logical Time-Triggered Distributed Computing Environment for Cyber-Physical Systems with V2X Wireless Communication
3. 学会等名 17th International Conference on Information Technology and Applications（国際学会）
4. 発表年 2023年

1. 発表者名 Takado, K., Yokoyama, T., Yoo, M.
2. 発表標題 A Logical Time-Triggered Distributed Computing Environment for Cyber-Physical Systems with V2X Wireless Communication
3. 学会等名 6th Asia Pacific Information Technology Conference（国際学会）
4. 発表年 2024年

1. 発表者名 Nishimura, T., Yokoyama, T., Yoo, M.
2. 発表標題 A Simulink to UML Model Transformation Tool for Distributed Embedded Control System Development
3. 学会等名 4th International Conference on Computer Communication and Information Systems (国際学会)
4. 発表年 2024年

1. 発表者名 溝呂木裕規、横山孝典、兪明連
2. 発表標題 アスペクト指向プログラミングによる並列・分散アラーム管理機能を有するリアルタイムOSの実現
3. 学会等名 情報処理学会 第60回組込みシステム研究発表会
4. 発表年 2022年

1. 発表者名 川田陸、竹原永也、横山孝典、兪明連
2. 発表標題 Linux上のリアルタイムOSエミュレータ
3. 学会等名 情報処理学会 第85回全国大会
4. 発表年 2023年

1. 発表者名 小日向大祐、飯田友樹、横山孝典、兪明連
2. 発表標題 GNSSを用いた協定世界時同期機能を有するリアルタイムOS
3. 学会等名 情報処理学会 第85回全国大会
4. 発表年 2023年

1. 発表者名 小林万里、横山孝典、兪明連
2. 発表標題 物理時間駆動と論理時間駆動に対応したリアルタイムOS
3. 学会等名 情報処理学会 第85回全国大会
4. 発表年 2023年

1. 発表者名 橋本優太、横山孝典、兪明連
2. 発表標題 デッドラインに基づくスケジューリングが可能なCAN通信ソフトウェア
3. 学会等名 情報処理学会 第85回全国大会
4. 発表年 2023年

1. 発表者名 溝呂木裕規、横山孝典、兪明連
2. 発表標題 アスペクト指向プログラミングによる並列・分散アラーム管理機能を有するリアルタイムOSの実現
3. 学会等名 情報処理学会第84回全国大会
4. 発表年 2022年

1. 発表者名 菊池優輔、横山孝典、兪明連
2. 発表標題 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
3. 学会等名 情報処理学会第84回全国大会
4. 発表年 2022年



1. 発表者名 高堂航希、横山孝典、兪明連
2. 発表標題 物理時間駆動と論理時間駆動に対応したリソース管理機能を有するリアルタイムOS
3. 学会等名 情報処理学会第84回全国大会
4. 発表年 2022年

1. 発表者名 西村太凱、横山孝典、兪明連
2. 発表標題 分散処理に対応したSimulink・UMLモデル変換ツール
3. 学会等名 情報処理学会第84回全国大会
4. 発表年 2022年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------