

# Wii Fit Board と Sphero SPRK+ による直感的なロボット制御のための MROS2 の活用

1020259 中村 碧

指導教員：松原克弥

Aoi Nakamura

**概要：** ロボットソフトウェア開発において、ROS ( Robot Operating System ) の利用が増えている。ROS はクラウドサーバと連携し、分散型ロボットシステムを構築するのに役立つ。しかし、各機能モジュール ( ノード ) の配置はシステム稼働前に決定する必要がある、予測困難な状況変化で最適なノード配置が変わる可能性がある。クラウドとロボット間での CPU アーキテクチャの違いにより、稼働中のノードの再配置は技術的に難しい。この研究では、ROS ノードの動的配置機構の実現に向けた、組み込みデバイス向け ROS 2 ランタイム実装である mROS 2-Posix を評価する。mROS 2-Posix が ROS 2 と比較してマイグレーションに優位性があるかを明らかにし、動的配置機構の実現への影響を実験結果で示す。

**キーワード：** ロボティクス, mROS2

**Abstract:**

**Keywords:** Roboethics, mROS2

## 1 背景と目的

さまざまな産業向けやエンターテインメント関連のロボットシステム、自動車の自動運転技術や IoT システムの構築においても、これらのソフトウェア開発をサポートするフレームワークとして Robot Operating System ( 以下、ROS ) の普及が増加している [1]。ROS のプログラミングモデルは、システムの各機能を独立したプログラムモジュール ( ノード ) として設計することにより、汎用性と再利用性を向上させて、各機能モジュール間のデータ交換を規定することで、効率的かつ柔軟なシステム構築を可能にしている。たとえば、カメラを操作して周囲の環境を撮影するノード、画像からオブジェクトを識別するノード、オブジェクトのデータを基に動作制御を実行するノードを連携させることで、自動運転車の基本機能の一部を容易に実装できる。

ROS のプログラミングモデルは、ロボット / IoT とクラウドが協力する分散型システムにおいても有効である。ロボットシステムのソフトウェア処理は、「センサー」「知能・制御系」「動力系」の 3 要素に分けられる [2]。クラウドロボティクスにおいて [3]、主に知能・制御系のノード

を高い計算能力を持つクラウドに優先して配置することで、高度な知能・制御処理の実現を促進し、さらに、必要な情報をクラウドに集約・保存することで、複数のロボット間での情報共有と利用を容易にする。一方で、現行の ROS 実装においては、各ノードの配置をシステム起動時に設定する必要がある、先述のクラウドロボティクスのクラウドとロボット間の最適なノード配置を事前に設計する必要がある。しかし、実際の環境で動作するロボットは、ネットワークの状況やバッテリー残量の変動など、システム運用前に予測することが困難な状況変化に対応する必要がある、設定したクラウドとロボット間のノード配置が最適でなくなる可能性がある。このような状況変化に対する対応として、ノードを動的に再配置するマイグレーション機能の実現が求められているが、多くの場合でクラウドとロボット間の CPU アーキテクチャが異なり、実行中のノードをシステム運用中にマイグレーションすることは技術的に困難である。

菅らは、WebAssembly ( 以下、Wasm ) を用いることで、クラウドとロボット間での実行状態を含む稼働中ノードを動的にマイグレーションする手法を提案した [菅さんの論文]。課題として、

ROS 2 を Wasm 化したことでオーバーヘッドが増大し、マイグレーション後、ノードの実行時間が大幅に増えてしまう問題が残った。柿本ら [1] は、組み込みデバイス向け ROS 2 ランタイム実装である mROS 2-POSIX を採用し、ROS ランタイムの Wasm 化にともなうオーバーヘッド増加に対処した。しかし、採用された mROS 2-POSIX はごく小規模なアプリケーション上でしか評価実験は行われておらず、今後も mROS2 POSIX を採用していく上でアプリケーションの規模を増加させた評価実験は必要不可欠である。

本研究では、クラウドとロボット間での実行状態を含む稼働中ノードの動的マイグレーションの実現に向けた mROS 2-Posix の評価を行い、mROS 2 が ROS 2 と比べて動的配置機構の実現のソフトウェア基盤としてどの点で優位性があるのか明らかにすることを目指す。

## 2 mROS2-POSIX

そもそも mROS 2 [4] は高瀬らが研究開発を進めている ROS 2 ノードの軽量実行環境である。mROS 2 は中規模の組み込みデバイス上で実行されるプログラムについて、汎用デバイス上の ROS 2 ノードと自律的に通信する機能を提供している。このソフトウェア基盤によって、分散型のロボットシステムへの組み込み技術の導入が促進できる。組み込みデバイスは計算資源が限定的であるが、これを活用することによってリアルタイム性の向上および消費電力の削減を図ることができる。そして、mROS 2 が POSIX [2] に対応したのが mROS 2-POSIX である。本章では、組み込みデバイス向けの高効率な ROS 2 通信方式およびメモリ軽量な実行環境を提供している mROS 2-POSIX について、ソフトウェア構成を中心に解説する。

図 1(a) は、mROS 2-POSIX のソフトウェア構成を示している。上層から順に、まず mROS 2-POSIX アプリケーション層は、ユーザが実装する ROS 2 ノードに相当する。mROS 2-POSIX API 層および通信ライブラリ層は、ROS 2 の topic に相当する API および通信機能を提供する階層である。本階層は、ROS 2 のネイティブなクライアントライブラリである rclcpp と互換性を保つように設計している。なお、mROS 2 通信ライブラリでは、rclcpp のうち pub/sub 通信の基本的な機能のみ実装されている。利用可

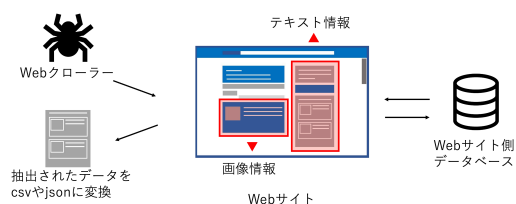


図 1 mROS 2 と mROS 2-POSIX の内部構造

能な機能は制限されているものの、組み込み技術を導入する ROS 2 開発者は、汎用 OS 向けのプログラミングスタイルを踏襲しながら C++ によって mROS 2 のアプリケーションを実装することができる。

Real Time Publish-Subscribe (以下、RTPS) プロトコルスタックには C++ 実装の embeddedRTPS [3] が採用されている。メモリ領域の静的確保など組み込みデバイスでの稼働を想定して設計されており、RTPS における Simple Participant Discover Protocol (以下、SPDP) および Simple Endpoint Discover Protocol (以下、SEDP) が実装されているため、通信の自立性が実現できる。UDP については組み込み向けの C 実装である lwIP [5] が採用されている。また、これらの階層は CMSIS-POSIX (Cortex Microcontroller System Interface Standard Portable Operating System Interface) API に依存している。通信層の embeddedRTPS および lwIP は CMSAIS-POSIX に依存しており、図 1(b) に示す mROS 2 の CMSIS-RTOS [6] を互換した層になっている。最下層にはハードウェアを抽象化したライブラリがある。

mROS 2-POSIX の実行方式は、mROS 2 が対応しているリアルタイム OS との相違がある。リアルタイム OS では、組み込みマイコンを実行資源の管理対象として、タスク単位でアプリケーションが実行される。POSIX においてはタスクに相当する概念はプロセスであり、そこから生成されるスレッドを実行単位として処理が進行している。

したがって、mROS2-POSIX は図 2 に示す実行方式を採用している。mROS 2-POSIX の実行単位であるノードは POSIX のスレッドに対応付けられており、mROS 2 の通信ライブラリ機能を

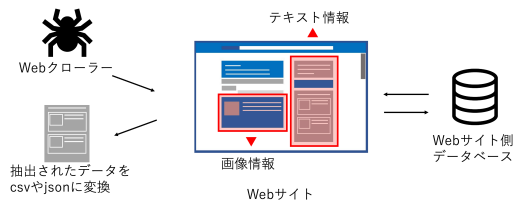


図 2 mROS 2-POSIX の実行方式

担う実行資源は POSIX のプロセスとして捉え、これを管理対象としている。組込みマイコンでの通信処理におけるイベント割込みについては、POSIX 準拠 OS におけるブロッキング API の発行に相当させて処理されている。

mROS 2 と mROS 2-POSIX の機能構造の対応として、タスク管理機能は、スレッドの生成や開始などの POSIX 資源の管理機能に対応付けられている。メッセージキューおよびミューテックスによるスレッドの同期・通信および排他制御は、POSIX では Pthread によって対応している。メモリ管理機能は Alloc/Free で、時間管理は OS 時刻管理機能で対応している。lwIP における UDP パケット処理は、POSIX における OS 資源の操作に関する機能で実現されている。

### 3 評価方針

mROS 2-POSIX を評価するにあたって、mROS2 と同様のアプリケーションを ROS 2 に実装し、比較評価する。mROS 2 は、ROS 2 のネイティブなクライアントライブラリである rclcpp と互換性を保つように設計されており、ROS2 に同様のアプリケーションを実装した場合でも期待される動作をすることができる。評価項目としては、mROS 2-POSIX が ROS 2 と比べてどの点で優位性があるのかを明らかにするため、以下の項目を設定した。

- メモリ使用量
- 実行時間
- マイグレーション時間

### 4 評価アプリケーションの実装

### 5 進捗と計画

### 6 結言

### 7 関連研究

### 8 知能システムコースにおける本研究の位置づけ

### 参考文献

- [1] 立教大学, "立教大学シラバス検索システム", 立教大学. [Online]. Available from: [https://www.rep-rikkyo.com/class\\_rooms?day=c&hour=4](https://www.rep-rikkyo.com/class_rooms?day=c&hour=4). Accessed: 2023-10-25.
- [2] 東京学芸大学 (非公式), "東京学芸大学教室検索システム", 東京学芸大学. [Online]. Available from: <https://u-gakugei-uo.a.pages.dev/akitan/>. Accessed: 2023-10-25.
- [3] SeleniumHQ, "Selenium Documentation", Selenium, 2023. [Online]. Available from: <https://www.selenium.dev/documentation/en/>. Accessed: 2023-10-25.