組込みシステム向け軽量 ROS 環境に対応した LEGO SPIKE 用ファームウェアの自動生成

2020SC017 樋山一樹
指導教員：本田晋也

1　はじめに

組込みシステム開発の分野における技術教育・人材育成を目的とした ET ロボコンと呼ばれるコンテストが開催されている．このコンテストは指定の走行ロボットを制御するプログラムを開発し，ゴールを目指す競技となっている．

ET ロボコンでは走行ロボットの制御アプリケーション開発用に RasPike と呼ばれるソフトウェアプラットホームが提供されている．RasPike の構成図を図 1 に示す．RasPike は LEGO Education SPIKE Prime の Prime Hub（Hub）と Raspberry Pi(RPI) の 2 台のコンピュータ間で，独自仕様による通信を行い，動作するものとなっている．Hub 上では MicroPython で記述された Hub 制御プログラムが稼働し，センサ値の取得と RPI への送信や，RPI から受信したモータの指令値の受信と，それをもとにしたモータ出力を行っている．RPI 側では Hub から受信したセンサ値をもとにモータの指令値を計算し，Hub への送信を行っている．そして，RPI 側の制御アプリケーションを ET ロボコン参加者が開発する．

この RasPike は，Hub 上で MicroPython を使用している事から，リアルタイム性に課題を抱えている．そこで，SPIKE-RT[1] が開発された．SPIKE-RT は Hub 上で動作するリアルタイム OS（RTOS）を中心としたソフトウェアプラットホームである．これによりリアルタイム性の高いシステムが実現可能になることが期待されている．また，近年のロボット制御の分野においては ROS が利用されている．ROS は分散処理のための通信ミドルウェアであり，PC 上で動作する ROS2 や，マイコンを ROS2 に接続するための機構である micro-ROS が存在する．

本研究では，RasPike のリアルタイム性の向上のために，Hub 制御プログラムの C 言語化（RasPike-RT）を実施する事に加え，RasPike の通信に ROS2 を導入（RasPike-ROS）することで通信の汎用性を向上し，既存の ROS2 ソフトウェア資産の再利用が可能であるようなソフトウェアプラットホームの実現を目指す．また，ユーザが開発するアプリケーションの自由度の向上のため micro-ROS ファームウェアの自動生成ツールの実現も実施する．

2　背景技術

2.1　RasPike

RasPike で使用される Hub と RPI は UART により接続されている．通信の仕様には RasPike 独自のものを使用しているため，通信の汎用性に課題を抱えている．

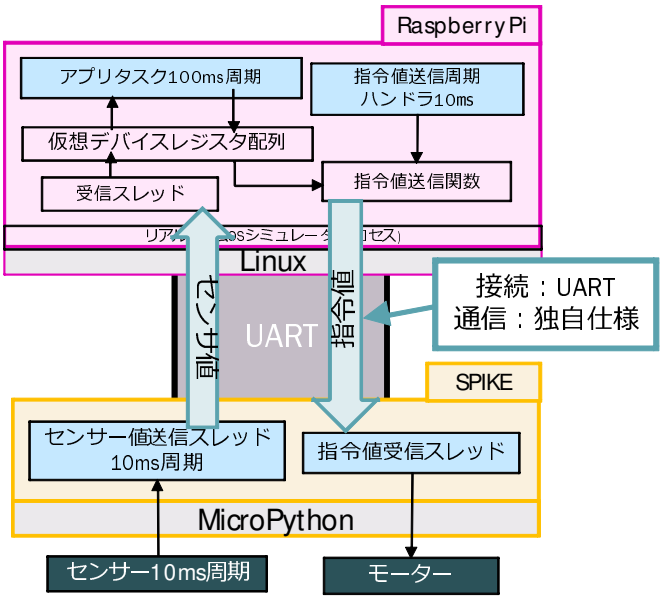

図 1　RasPike の内部構成

2.2　spike-rt

spike-rt とは ITRON 系の RTOS を中心とした Hub 向けのオープンソースのソフトウェアプラットホームである．これは Hub 上でネイティブに稼働し，C 言語でのアプリケーション開発，及びアプリケーションのリアルタイム性の保証が可能である．

2.3　ROS2/micro-ROS

ROS2 とは，ロボットや自動運転向けの分散処理フレームワークである．topic と呼ばれる単位のメッセージの pub/sub 通信を行っている．

micro-ROS（uROS）[2] とは，スペックの関係上 ROS2 を稼働することが難しいマイコンを ROS2 に接続するための機構である．uROS は処理性能を要する DDS 通信処理を汎用 PC 上に依頼する構成を取っている．

3　RasPike-ROS ランタイム

3.1　RasPike-RT

まず RasPike-RT の実現から実施する．これは RasPike の Hub 制御プログラムを MicroPython から spike-rt に変更し，MycroPython の 2 種類スレッドを C 言語の ITRON のタスクに置き換える．また，従来の RasPike と制御アプリケーションの互換性を保つため，通信は RasPike 独自の仕様に揃えて実現する．

実現した RasPike-RT の End to End 応答時間を 20 回計測した．ここでの End to End 応答時間はセンサが値

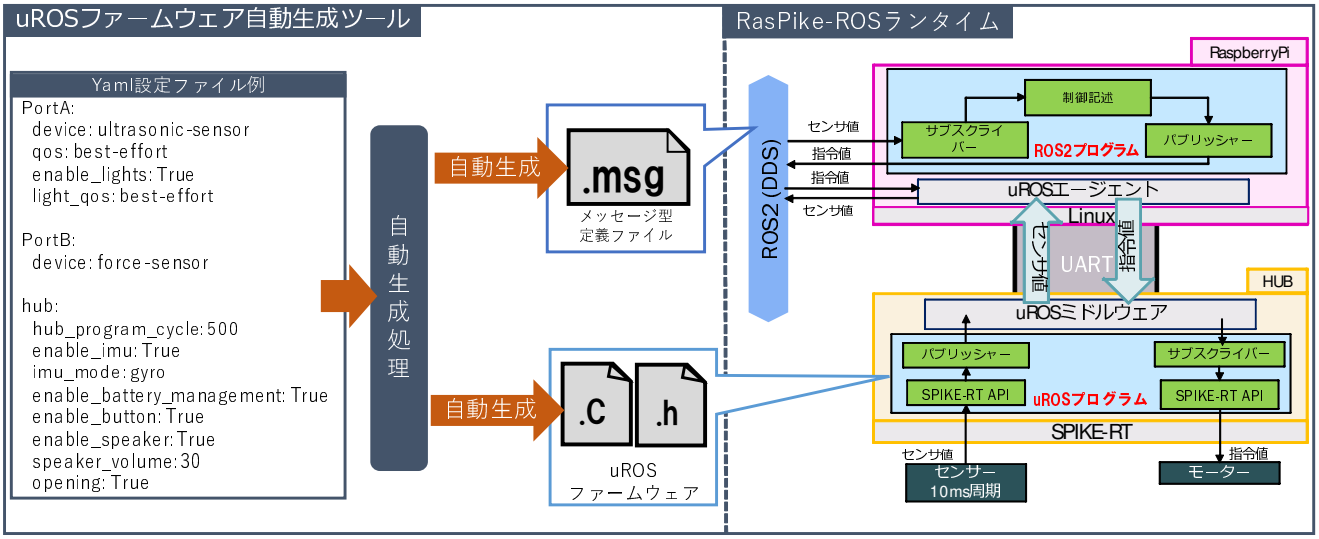

図 2　RasPike-ROS の内部構成と自動生成ツールの構成

の変化を検知してから，それに応じて変化したモータへの指令値が届くまでの時間を指す．この結果，RasPike の応答時間の最大値が 31.6ms，平均値が 23.52ms であり，RasPike-RT の最大値が 49.8ms，平均値が 34.07ms となり，応答性が向上する結果となった．

3.2　RasPike-ROS

Hub と RPI 間の通信に ROS2 を使用することで通信の汎用性の向上を図る．内部構成は図 2 の右側に示す通りである．RasPike-ROS の実装により，走行ロボットと外部機器を ROS2 通信により接続が可能になり，ROS2 のソフトウェア資産の再利用が可能であるようなファームウェアプラットホームの実現に成功した．

機能評価として RasPike-ROS を使用して走行ロボットを PID 制御によりライントレースするプログラムを作成し，実行した．これにより対象の走行ロボットを十分に制御することが可能であることを確認した．

4　uROS ファームウェア自動生成ツール

RasPike-ROS は Hub に接続する PUP デバイスの構成を固定化した仕様となっている．これでは，Hub に接続するデバイスの構成をユーザが変更する場合，ROS2 プログラムの開発に取り掛かる前に uROS のファームウェアと必要なカスタムメッセージ定義ファイルを再度作成する必要がある．これは ROS2 プログラミングの学習者にとって大きな障壁となる．そこで，図 2 に示すように，ユーザが Hub 側の構成を設定ファイルに記述するだけで必要なメッセージ定義ファイルと uROS ファームウェアを自動生成するようなツールを実現する．

実現にあたり，設定ファイルは Yaml ファイルで作成する仕様とした．ユーザから見て設定内容をわかりやすくするため，インデントを一段使用する構成を採用する．図 2 に示したように，設定ファイルに使用するポートを記述し，その下に接続するデバイスの種類と各種のデバイス向けの設定を記述する．

設定ファイルの読み取りには Yaml 向けの Python ライブラリである PyYaml を使用する．PyYaml で Yaml ファイルの内容を読み取った後，構文解析を行った上で設定内容を取得する．そして取得した情報を元に，各種ファイルを生成する構成となっている．ユーザが使用するデバイスの数に制限を設けないようにするため，トピック名やセンサ値等を格納する変数名は，自動生成時に動的に命名している．また，生成するパッケージの名前はユーザが任意に指定できる仕様とした．

このツールにより，ユーザが Hub の構成に縛られずとも ROS2 プログラミングに専念できるソフトウェアプラットホームが実現された．また，37 行の設定ファイルに対してコードの生成量は約 1400 行という結果になった．

5　おわりに

RasPike-RT の実装による RasPike の応答性の向上と RasPike-ROS の実装による通信の汎用性の向上を実現した．そして，自動生成ツールによりユーザが uROS ファームウェアを作成することなく Hub 側の構成を自由に変更して ROS2 プログラムの開発が可能なソフトウェアプラットホームの実現を行った．今後の課題として，実際に RasPike-ROS と外部機器を ROS2 により接続して走行ロボットを制御するシステムを構築することが挙げられる．

参考文献

[1] 朱 義文, 李 奕驍, 松原 豊, 本田晋也：spike-rt: LEGO SPIKE Prime 向けリアルタイムソフトウェアプラットホーム，情報処理学会研究会研究報告，(Vol.2021-EMB-58 No.4,) (2022).

[2] K. Belsare, A. C. Rodriguez, P. G. Sanchez, J. Hierro, et al. Micro-ROS. In: Anis Koubaa (ed.) Robot Operating System (ROS): The Complete Reference (Volume 7), Springer, pp. 3-55 (2023)