

TOPPERS 活用アイデア・アプリケーション開発 コンテスト

部門 : アプリケーション開発部門

作品のタイトル : SPIKE-RT — LEGO SPIKE Prime 向けソフトウェアプラットフォーム

作成者 : 朱 義文 (名古屋大学)

共同作業 :

対象者 : LEGO SPIKE Prime 使用者, TOPPERS カーネル入門者

使用する開発成果物 : TOPPERS/ASP3

目的・狙い

- LEGO SPIKE Prime におけるリアルタイムプログラミング環境の提供.
- LEGO SPIKE Prime による TOPPERS カーネルの入門教材の提供.

アイデア/アプリケーションの概要

SPIKE-RT は, 現在開発中の向け LEGO SPIKE Prime 向け RTOS ベースのソフトウェアプラットフォームである.

現状として, 一部の外部センサや Hub 本体に付属しているデバイスの制御が可能である.

1. 背景

LEGO® Education SPIKE™ Prime (以後, 単に「SPIKE Prime」と呼ぶ) は, 小学校高学年から中高生向けの STEAM 学習セット[1]である. Prime Hub というプログラミング可能なデバイスと PUP(Powered Up)デバイスと呼ばれる外部デバイスを組み合わせることで様々なアプリケーションを開発することができる.

しかし, SPIKE Prime には, リアルタイムなアプリケーションを開発するためのソフトウェアプラットフォーム (以後, 「SPF」と呼ぶ) が存在しないようである. SPIKE Prime で利用可能な SPF として LEGO Education が公式に提供しているものと, OSS コミュニティによって開発されている Pybricks[2]が利用可能である. 前者の公式の SPF では, MicroPython のランタイムをベースとしてファームウェアが開発されているようであり, アプリケーションの最悪応答性を保証することが難しい. 後者の Pybricks は, MicroPython ランタイムと Contiki[3]というノンプリエンプティブ・カーネルをベースにファームウェアが開発されている. つまり, どちらもリアルタイムプログラミングには不向きである.

また, もし, 多くの人にとって比較的親近感のある LEGO を入口とした TOPPERS カーネルの教材を提供することができれば, あまり馴染みのない方にも TOPPERS プロジェクトに触れるきっかけを提供することができるだろう.

そこで, SPIKE Prime におけるリアルタイムプログラミング環境を目標とし, SPIKE-RT を開発した.

2. SPIKE-RT

SPIKE-RT は, GitHub[4]で SPIKE-RT のソースコードを公開している. SPIKE-RT を用いることで, 複雑なデバイスドライバを実装することなく, 容易にアプリケーションを実装することができる. ユーザはアプリケーションプログラムにおいて以下の API を使用することができる.

➤ TOPPERS/ASP3

ASP3 のスタイルに従ってアプリケーションを開発することができる.

➤ Newlib

Newlib による C 言語標準ライブラリに一部対応している.

➤ SPIKE API (仮称)

SPIKE Prime のデバイスを制御するための API である. デバイスの対応状況を表 1 に示す.

	動作	API 対応
マトリックス LED	○	○
ステータスライト	○	○
ボタン	○	×
IMU	×	×
スピーカ	×	×
外部フラッシュメモリ	×	×
カラーセンサ	○	○
超音波センサ	○	○
フォースセンサ	○	○
サーボモータ	○	×

表 1. デバイス対応表

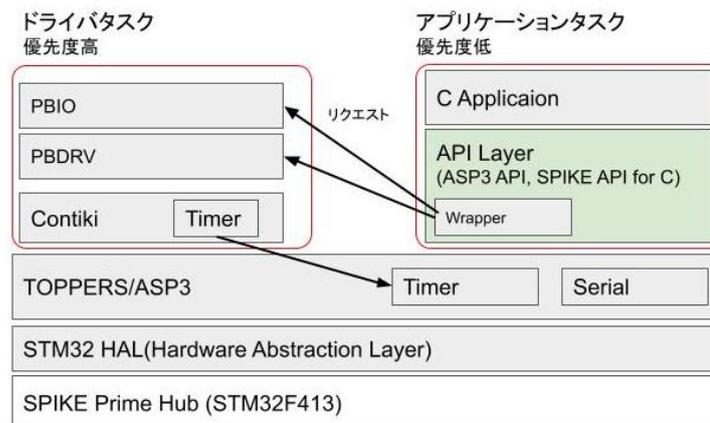


図 1. SPIKE-RT 全体構成図

3. 使用方法

現状では対応デバイスの動作確認を行うテストアプリケーションを動作させることができる。以下にテストアプリケーションを動作させるまでの手順を示す。

- ① **LEGO SPIKE Prime** の入手
[5]などから購入可能である。
- ② 開発環境の構築
[6]に従って開発環境を構築する。アプリケーションコンテナによるビルド環境の構築を想定している。
- ③ テストアプリケーションのビルドと動作確認
[7]に従って、テストアプリケーションのビルド、**Prime Hub** への書き込み、動

作確認を行う。

ただし、以下の通り PUP デバイスを Prime Hub に接続する

- Port A にモーターを接続する。
- Port B にフォースセンサを接続する。
- Port C に超音波センサを接続する。
- Port D にカラーセンサを接続する。

4. 設計

SPIKE-RT の設計について説明する。

まず全体像について説明する。構成を図 1 に示す。まず、土台となる RTOS として TOPPERS/ASP3 が動作している。その上でアプリケーションタスクとドライバタスクが動作している。アプリケーションタスクは、ユーザが独自に開発するアプリケーションプログラムである。ドライバタスクは、SPIKE Prime の一部のデバイス制御を行い Pybricks を ASP3 のタスクとして動かすことで実現される。アプリケーションタスクは、SPIKE API を通してデバイスを使用することができる。SPIKE API の一部は、ドライバタスクに対してリクエストを送る。

デバイス制御について詳しく説明する。一般に、デバイスドライバを実装するためには、

- デバイスの制御の仕組みを理解し、0 から実装する。
- Pybricks のような既存のソフトウェアから、デバイスドライバを ASP3 向けに移植する。

といった方法が思いつくだらう。前者は、言うまでもなく莫大な労力を費やす。一方、後者も、Pybricks には、10 以上のプロセスが動作しており、個々を移植するのは、少なくないコストを要する。そこで SPIKE-RT では、Pybricks の実行基盤である Contiki を ASP3 のタスク上に移植することで Pybricks のデバイスドライバをほとんど修正することなく動作させることができた。現状では 1 つのタスク上に、Pybricks の Contiki 上で動作しているデバイスドライバが動作している。

この Contiki 向けプログラムを TOPPERS/ASP3 にそのまま再利用するアプローチは、その他の Contiki 向けに開発されたプログラムに対しても概ね適用可能であると期待している。

5. 今後の展開

残りのデバイスへの API 対応を進め、さらなる機能拡張を目指す。

6. 参考文献

- [1] SPIKE Prime の公式日本語製品ページ,
<https://education.lego.com/ja-jp/products/-spike-/45678#spike%E3%83%97%E3%83%A9%E3%82%A4%E3%83%A0>
- [2] Pybricks, <https://pybricks.com/>
- [3] Contiki, <http://www.contiki-os.org/>
- [4] SPIKE-RT の GitHub ページ, <https://github.com/spike-rt/spike-rt>
- [5] Afrel による SPIKE Prime の販売ページ:
https://afrel-shop.com/view/item/000000000569?category_page_id=ct257
- [6] <https://github.com/spike-rt/spike-rt/blob/main/docs/ja/Env.md>
- [7] <https://github.com/spike-rt/spike-rt/blob/main/docs/ja/Test.md>

以上