

複数の OSS で構成されている組み込みシステムの性能評価

2020SC066 大野佑真

指導教員：本田晋也

1 はじめに

SPIKE プライムは、LEGO 社の STEAM 学習キットである。本体である Hub は 6 つの入出力ポート、LED ライト、スピーカーなどを備えており、各種センサやモータなどが取り付け可能である。SPIKE プライムはプログラミング言語、機械学習、AI についての学習や、授業や講義での演習など幅広い場面で使用されている。

SPIKE プライムのオリジナルの開発環境として Python があるが、これらはリアルタイム性が保証されていない [1]。また、組み込みシステムでよく使われる C 言語にも対応していないという問題点がある。そこで、C 言語で開発可能でリアルタイム性を保証することが可能なソフトウェアプラットフォームとして SPIKE-RT が開発された [2]。SPIKE-RT は開発されて間もなく、センサ値のログの出力などに利用される USB や Bluetooth などを使用した時のリアルタイム性の保証ができていない。

先行研究ではカラーセンサとモータ間の End-to-End 応答時間の評価をしているが、手動で計測しているため評価数が少ない、手動でデータの抽出をしているため多くのデータを扱えないという課題がある。

本研究では計測の自動化のためのテストベンチの作成とデータ抽出の自動化のためのアプリケーションの実装を行い、SPIKE-RT の優位性を示すとともに、SPIKE-RT の問題点の提示と改善を目的とする。

2 背景技術

2.1 SPIKE プライム

センサまたはモータと Hub は専用線により接続する。専用線の信号は、電源、UART、2 本の汎用線がある。センサの場合は、UART 経由でセンサのモードとセンサ値が送られる。モータの場合は、2 本の汎用線で PWM 信号とモータの向きを制御する。

2.2 SPIKE-RT

SPIKE-RT は、SPIKE プライム向けの RTOS + C 言語のソフトウェアプラットフォームで OSS として配布されている。ドライバ等は、Pybricks と呼ばれる MicroPython の環境のソースコードを利用している。Pybricks は、複数の OSS を組み合わせて実現されている。

3 評価対象

本研究では、モータの基本性能評価、End-to-End 応答時間の評価、USB と Bluetooth で文字を送信しているときのリアルタイム性の評価を行う。本研究では、センサが物体を検知して、アプリケーションで判断し、モータを駆動

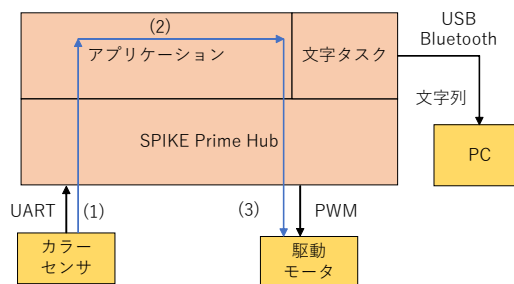


図1 計測環境の構成図

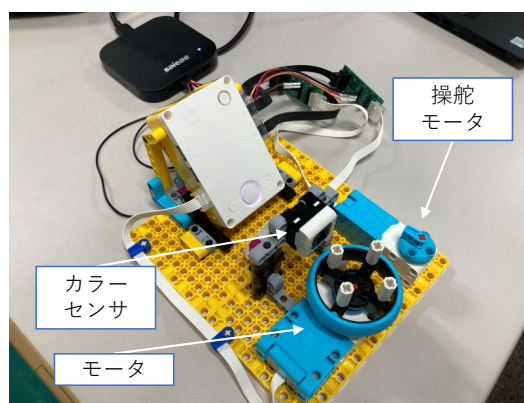


図2 テストベンチ

するまでの時間を End-to-End 応答時間として評価する。評価対象の処理の流れを図1に示す。

- (1) カラーセンサが明るさの値を UART に流す。
 - (2) アプリケーションで物体の検知を行う。検知した場合は操舵モータを駆動する。
 - (3) 操舵モータに PWM 信号を送信する。
- (1)が発生してから、(3)の PWM 信号が送られるまでの時間を End-to-End 応答時間とする。

4 評価システム

前述の End-to-End 応答時間の計測に対して、計測を自動で行う評価システムを実現する。

4.1 計測環境

評価数を多く取得するため、図2のテストベンチを実現した。このテストベンチは、モータにつながった4本の検知対象をカラーセンサの前で回転させる。モータの回転速度は最大で1000度/sであるため、検知対象はカラーセンサの前を90ms周期で横切る。カラーセンサの周波数は100Hzと10倍程度高速であるため、モータの回転速度を最大としても検知可能である。カラーセンサにより検知対象を感知している間、操舵モータを駆動する。

4.2 評価フロー

評価フローは計測フェーズと抽出フェーズに分けられる。計測フェーズでは、テストベンチを動作させ、ロジックアナライザでカラーセンサーと HUB 間の UART 通信と、操舵モータと HUB 間の PWM 信号の取得を行う。抽出フェーズでは、計測結果の CSV ファイルを抽出プログラムで解析し、End-to-End 応答時間を抽出する。

4.2.1 計測フェーズ

1. モータを回転させカラーセンサに物体を通過させる。
カラーセンサが Hub に送るセンシングの情報を取得する (計測結果 a)。
2. 物体を検知している間操舵モータの駆動命令を出す。
モータと Hub 間の PWM 信号の変化を取得する (計測結果 b)。

4.2.2 抽出フェーズ

3. ロジックアナライザの計測結果 (a)(b) を CSV ファイル (信号 CSV) として出力する。
4. 抽出プログラムにより応答時間を求める。

信号 CSV は、1 列目に信号が変化した時刻が記載されており、2 列目にビットデータ '0' or '1' が記載されている。

4.2.3 抽出プログラム

信号 CSV を解析し、カラーセンサが物体を検知した時間と操舵モータが動き出した時間の差を算出し、End-to-End 応答時間を抽出する。カラーセンサが物体を検知したかの判断は、UART 上のセンサ値が 0 からそれ以外の値に変化したかで判断する。この判断を自動化するために、まず計測結果 (a) を変換して、時刻と数値のデータ列に変換する。次にデータ列をチェックして、数値が 0 からそれ以外の値に変化した点を検索していく。最後に、変化した時刻から計測結果 (b) をチェックし、PWM 信号が有効になった時刻を求めその差を応答時間とする。

5 評価

5.1 応答時間評価

SPIKE-RT と Python それぞれの環境で評価して、その結果を比較する。試行回数は 1000 回とする。SPIKE-RT の結果を図 3、Python の結果を図 4 に示す。評価の結果、Python より SPIKE-RT のほうがリアルタイム性、最悪応答時間において優れていることが分かる。Python の End-to-End 応答時間が周期的なのはセンサやモータの周期が影響していると考えられる。

5.2 USB/Bluetooth 通信の影響評価

USB で秒間 6400 文字、秒間 12800 文字、Bluetooth で秒間 1024 文字、秒間 2048 文字送信しているときの End-to-End 応答時間の計測、比較を行う。Bluetooth で秒間 0 文字、秒間 1024 文字、秒間 2048 文字送信しているときの

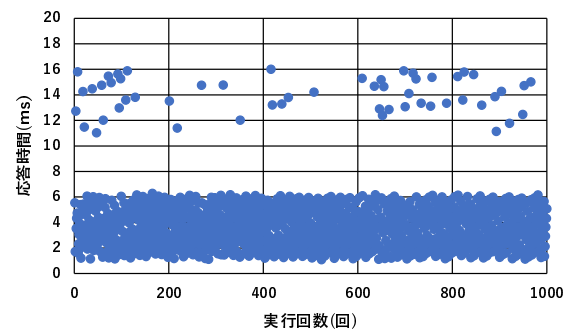


図 3 SPIKE-RT の End-to-End 応答時間

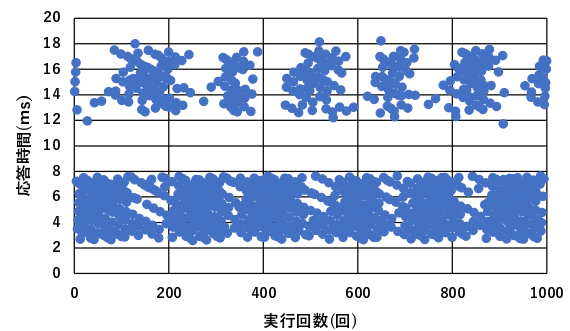


図 4 Python の End-to-End 応答時間

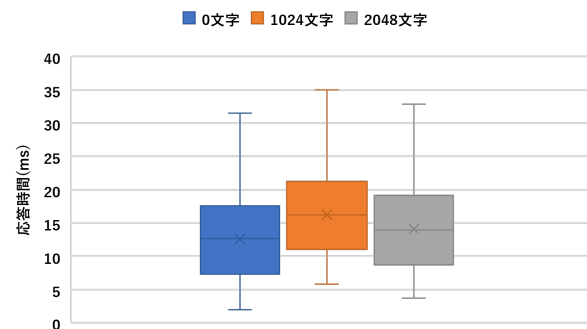


図 5 Bluetooth を使用しているときの応答時間の比較

End-to-End 応答時間を比較した箱ひげ図を図 5 に示す。評価の結果、USB で文字を送信したとき、Bluetooth で文字を送信したとき共に End-to-End 応答時間に大きな影響は与えなかった。

6 おわりに

本研究では End-to-End 応答時間の評価を自動で行う評価システムを作成し、SPIKE-RT と Python のリアルタイム性の評価を行った。その結果、Python よりも SPIKE-RT のほうが優れていることが確認できた。

参考文献

- [1] 永墓 晟：汎用コンピュータを用いたロボット制御プラットフォームのリアルタイム性評価，南山大学 理工学部 卒業論文，2022。
- [2] 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フォーム，情報処理学会研究報告，Vol.2022-EMB-61，No.11，2022。