

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

2019SC041 永基晟

指導教員：本田晋也

1 はじめに

レゴマインドストームと呼ばれるロボットプラットフォームを用いて、ソフトウェアを開発して競い合う ET ロボコンというコンテストがある。ET ロボコンは、ロボットを制御する必要があるため、そのコンピュータシステムやソフトウェアプラットフォームにはリアルタイム性が不可欠である。レゴマインドストームの標準のソフトウェア開発環境は Scratch 等のブロックベースの開発環境であり、OS にリアルタイム OS (RTOS) ではなく Linux や MicroPython を使用しているためリアルタイム性に問題がある。この問題を解決するために、1 世代前の EV3 では ET ロボコン用に、RTOS を導入し C 言語による開発が可能な環境が構築されてきた [1]。

レゴマインドストームは定期的にバージョンアップしており、最新のバージョンは LEGO SPIKE (SPIKE) である。SPIKE は記述されたプログラムを MicroPython のコードに変換し実行する。SPIKE 用の RTOS ベースの C 言語開発環境は有志により現在開発中であり、一部の機能のみサポートされた状態である。そのため、ET ロボコンでは RaspberryPi と SPIKE を組み合わせた RasPike が構築され、今年度から使用されている。RasPike では、Linux が動作する RaspberryPi で RTOS シミュレータを動作させ、その中でユーザーアプリケーションを実行する。そして、SPIKE からアクチュエータへの指令値やセンサー値を UART 経由で受け取る構成となっている。このように、EV3 上で実現された RTOS ベースのソフトウェアプラットフォームと異なり、リアルタイム性の確保が困難な MicroPython や Linux を使用している。しかしながら、RasPike の詳細な構成や、そのリアルタイム性について私が調査した限りでは評価がなされたことはない。

そのため、本研究では RasPike のリアルタイム性について評価を行う。具体的には、各周期処理の周期が正確であることや、負荷が高い場合のジッタがどの程度発生するか、センサーからモーターまでの応答時間について評価する。

2 背景技術

2.1 RasPike

RasPike の構成図を図 1 に示す。SPIKE は Python3 をもとに作られた MicroPython をベースに動いている。MicroPython は、Python ベースであるため、習得しやすいという特徴を持つが、実行速度が遅く、ガベージコレクション (GC) によるリアルタイム性の悪化も懸念点として挙げられる。

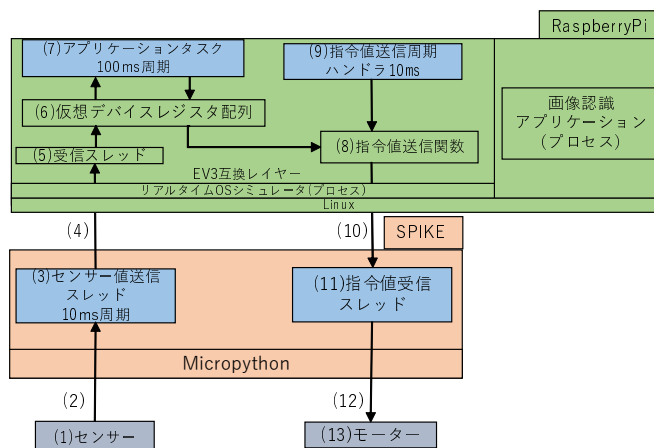


図 1 RasPike の構成図

センサーは SPIKE に取り付けられており、センサーで取得したセンサー値を指令値送信関数によって 10ms ごとに RaspberryPi に送信される。送信されたセンサー値は RaspberryPi 側の受信スレッドで受け取られ仮想デバイスレジスタ配列に格納される。アプリケーションタスクによって配列を読み込み、モーター指令値を配列に返す。そして RaspberryPi の送信関数によって 10ms ごとに SPIKE 側の受信スレッドに送信される。この時、周期ハンドラによる割り込みが考慮される。SPIKE 側の受信スレッドで受け取ったモーター指令値によってモーターが稼働される。

2.2 リアルタイム性

リアルタイム性とは、要求された時間以内に決められた処理の実行を終わらせることができ、実行時間や実行タイミングなどの予測ができる予測可能性を指す。

RTOS は、優先度の高い順にタスクを実行する優先度ベーススケジューリングや短い割り込み禁止区間により予測可能性が高い実行が可能となる。システムのリアルタイム性の実現を容易にする目的で、RTOS が用いられる。

3 RasPike の課題

RasPike は EV3 と異なり、単独で動かすのではなく RaspberryPi と SPIKE を組み合わせて動作している。また、複雑かつ多くのプログラムが同時に動作している Linux 上でアプリケーションを動作させるため、それらの影響をうけ要求された時間内に処理を終わらせることができない可能性がある。さらに、SPIKE 側は MicroPython で動作しているため、GC 等が動作することにより実行時間にジッタが生じる可能性があることが課題として挙げられる [2]。

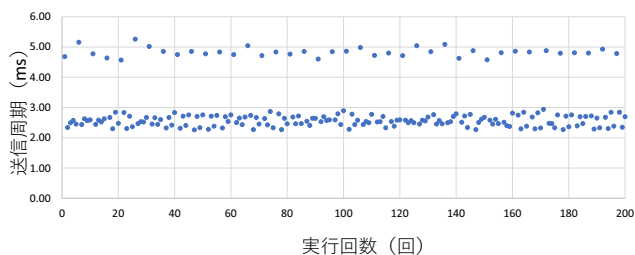


図2 SPIKE から RaspberryPi への UART の送信周期

4 周期処理のジッタ評価

RasPike の以下の 4 種類の周期処理を計測する。

1. アプリケーションタスク (100ms)
2. 指令値送信周期ハンドラ (10ms)
3. RaspberryPi から SPIKE への UART 送信 (10ms)
4. SPIKE から RaspberryPi への UART 送信 (10ms)

計測の結果, 1, 2 については, 実行直後で遅れが発生し, その後の実行で遅れた分だけ速い実行周期となっている。上記以降は全て設定時間通りで実行され続けたため, 平均値は設定通りとなった。実行直後に周期が遅れ, その後の周期が早くなっているのは, 初期化やキャッシュが原因であると考えられる。3 については誤差はあるものの設定通りの送信周期が確認できた。4 の実行結果を図 2 に示す。図 2 より, 10ms 周期に UART が送信される設定であるのに対し, 最大で約 5ms, 最小で約 2.3ms が計測された。設定されている時間内で送信がされているが, 最大値と最小値で差があることが分かる。原因は GC が動作している可能性があると考えられる。しかし, 明示的に GC を呼び出して実行すると, 最大値が約 10ms に変化したが, 最大値と最小値の差は縮まらなかった。また, 1 周期の時間を長くして同様の評価を行ったが最大値と最小値の差は縮まらなかった。このことから, GC 以外に何らかの原因があることが確認できた。

5 End-to-End 応答時間評価

End-to-End 応答時間とは, センサー値の変化を検知し, アプリケーションタスクにその変化が通知され, アプリケーションタスクがモーターの指令値を変化させ, モーターに伝わるまでの時間のことを指す。この章ではカラーセンサーがセンサー値の変化を検知してからモーターが動作するまでの応答時間について評価する。応答時間の開始時刻はセンサーに検知させる色が黒から白に変化したタイミングを契機とし, 終了時刻はモーターの駆動信号が流れたタイミングを契機とする。

5.1 計測方法

カラーセンサーが黒を検知した場合にモーターを停止し, 白を検知した場合はモーターを動かすプログラムを作成する。プログラムを実行し, 手動でカラーセンサーを黒と白の上を交互に移動させる。この時のモーターの PWM

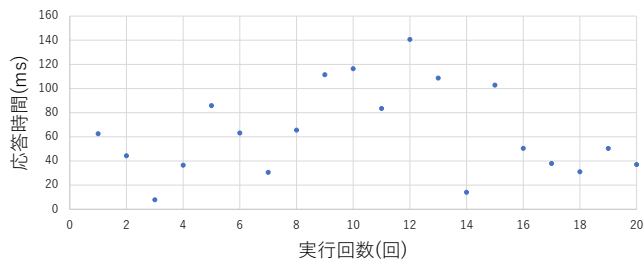


図3 End-to-End 応答時間計測結果

制御とカラーセンサーからの UART 送信の波形をロジックアナライザで計測する。計測した波形データのカラーセンサーのセンサー値が変化した箇所からモーターが動作するまでの時間を求める。

5.2 評価結果

センサー値が変化してからモーター速度が変化するまでの応答時間を図 3 に示す。図 3 より, ばらつきがあることが確認でき, 最大で 140.68ms, 最小で 7.8ms となった。最大値が 100ms を超えている理由は, 100ms 周期で動作するアプリケーションタスクが実行された直後にセンサー値が変化したためだと考えられる。RasPike のアプリケーションタスクやハンドラ, スレッドは全て個々に動作しているためアプリケーションタスクが実行された直後に変化したセンサー値が送信されても次のタスクの実行でしか受信がされないためである。反対に, 最小値はアプリケーションタスクが実行される直前にセンサー値の送信, 受信が行われたため, 時間差が大幅に短くなっていると考えられる。

6 おわりに

今回は RasPike における周期処理のジッタ評価と応答時間について評価した。評価項目 1, 2 では実行直後に周期の遅れが発生したがその後は全て設定通りの周期になることが確認でき, 3 では全て設定通りの送信周期が確認できた。また, 4 の SPIKE から RaspberryPi への UART の送信周期は全体を通して設定である 10ms は下回っているが最小値と最大値の差に幅があることが確認された。End-To-End 応答時間評価では, 最大で 140ms 以上となる場合があることが確認できた。今後の課題としては, アプリケーションタスクの周期を短くした場合の評価が挙げられる。

参考文献

- [1] Yixiao Li, Yutaka Matsubara, Hiroaki Takada : EV3RT: A Real-time Software Platform for LEGO Mindstorms EV3, コンピュータソフトウェア, Vol.34-4, pp. 4.91-4.115 (2017)
- [2] 板田怜子, 藤田一希, 横山哲郎, 本田晋也 : マイクロコントローラ向けスクリプト言語 MicroPython の組み込みシステムへの適用性評価, 情報処理学会研究報告, Vol.2021-EMB-58, No.4, pp. 1-8, (2021).