7K-01

小型デバイス向けデータフロー型ビジュアルプログラミング環境構築に向けたマイコンボードへの mruby/c コード書き込みツールの開発

河原　美優†　　尾倉　颯太‡　　杉山耕一朗†　　田中　和明*

松江工業高等専門学校情報工学科†　九州工業大学情報工学部‡　九州工業大学情報工学研究院*

1. はじめに

近年，内閣府より科学技術政策 Society 5.0 が定められ，IoT といったデータの活用が求められている．IoT 分野ではデータの流れの理解が重要であり，その理解にはデータフロー型のビジュアルプログラミング言語が有効と考えられる．そこで我々は，マイコンボードを利用した IoT 開発に対し，2 節で詳述するようなデータフロー型ビジュアルプログラミング環境を構築してきた[1]．しかしながら，これまでに開発した環境はプログラムの作成から実行までがブラウザ上で完結せず，デスクトップアプリケーションの併用が必要となるといったユーザの利用しづらさがあった．

近年，Web Serial API [2]によって Web ページとマイコンとのシリアル通信が可能となった．Web Serial API は 2021 年から「Chrome 89」に標準搭載されている．我々はこの技術を活用することで，これまで構築してきたデータフロー型ビジュアルプログラミング環境をブラウザ上で完結できる可能性があると考えた．そこで本研究では，Web Serial API を用いてバイトコードをマイコンボードへ書き込むツールを開発することを目的とする．

2. マイコンでのフローの実行手順

データフロー型プログラミング環境を用いて作成したフローをマイコンで実行するには，フローをマイコンの解釈可能な言語のプログラムに変換することや，そのプログラムをマイコンボードに書き込むという手順が必要となる．我々のこれまでの研究[1]における実行手順を図 1 に示す．PIC マイコンを搭載したマイコンボード RBoard [3]をターゲットとし，データフロー型ビジュアルプログラミング開発ツール Node-RED に RBoard 用のノードを追加した(以下，この拡張を加えた Noe-RED を「RBoard 用 Node-RED」と称す)．加えて，RBoard 用 Node-RED が生成する JSON コードを軽量 Ruby 言語 mruby/c のコードに変換するための Ruby 生成器を開発した．mruby/c コードをコンパイルしてバイトコードへ変換し，それをシリアル通信で PIC マイコンに書き込む部分については既存の mruby/c IDE [4]を用いている．

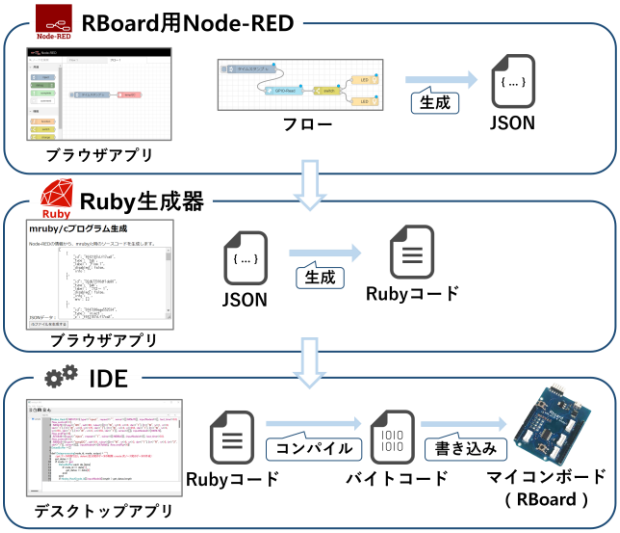

図 1 これまでの研究[1]における
プログラム実行手順

3. 研究内容

3.1 プログラム書き込みツールの概要

本研究で開発する書き込みツールは，2 節で述べた mruby/c IDE の担っているバイトコードをマイコンボードへ書き込む部分を置換するものである．この書き込みツールはブラウザ上で動作するクライアントサイドプログラムであり，HTML，JavaScript，CSS を用いて構成されている．

本書き込みツールの画面を図 2 に示す．画面上のボタンをクリックすることで必要な操作を実行することができる．まず「接続」ボタンをクリックすることで，パソコンと RBoard とのシリアル通信を確立することができる．次に，「ファイルを選択」ボタンをクリックすることで，バイナリファイルをアップロードすることができる．図には示さないが，バイナリファイルをアップロードすると「ファイルを選択」ボタンの表示がファイル名に変化する．最後に

Development of a mruby/c code writing tool to microcomputer board for a dataflow visual programming environment suitable for small devices
†Miu Kawahara, Ko-ichiro Sugiyama; National Institute of Technology, Matsue College
‡Sota Ogura, Kyushu Institute of Technology
*Kazuaki Tanaka, Kyushu Institute of Technology

「書き込み」ボタンをクリックすると，バイナリコードをマイコンへ転送することができる．なお，「切断」をクリックするとパソコンとマイコンとのシリアル通信が切断される．図 2 の下方にあるテキスト表示エリアは，シリアルモニタに相当するものである．RBoard の出力をこのエリアに表示させることができる．

図 2　プログラム書き込みツールの外見

3.2 プログラム書き込みツールの実装方法

書き込みツールは RBoard の通信プロトコル[5]に基づいて実装されている．このプロトコルではバイトコードの書き込みのために，改行コード，write コマンド，バイトコード，execute コマンドを順番に送信する必要がある．本書き込みツールのソースコードの骨格を図 3 に示すが，まさにこのバイナリファイルを書き込む手順が JavaScript で実装されている．

```
// シリアルデバイスへのアクセスをリクエストする
port = await navigator.serial.requestPort();

// writeボタン //
async function writeButtonClick() {
    // 書き込み可能なストリームの作成
    writer = port.writable.getWriter();

    // シリアルポートに¥r¥nを送信する
    await writer.write(encoder.encode("¥r¥n"));

    // シリアルポートにversionを送信する
    await writer.write(encoder.encode("version¥r¥n"));

    // シリアルポートにファイルを書き込む準備をする
    await writer.write(encoder.encode("write " + file_size + "¥r¥n"));

    // RBoardに.mrbファイルを転送
    await writer.write(ary);

    // RBoardにファイルを書き込み
    await writer.write(encoder.encode("¥r¥n"));

    // .mrbファイルを実行する
    await writer.write(encoder.encode("execute¥r¥n"));

    writer.releaseLock();  // 書き込み可能なストリームのロックを解除
}
```

図 3　プログラム書き込みツールのソースコードの骨格

図 3 において用いられている変数であるが，変数 port にはシリアルデバイスとの接続状態や，読み込みの状態などが格納されている．requestPort()メソッドを用いることで，ブラウザ上でシリアルデバイスをユーザに選択させている．また，変数 writer はデータをシリアルポートに書き込むためのインタフェースである．write()メソッドによって選択されたシリアルポートにデータを書き込んでいる．write()メソッドの引数は Web Serial API の規格に合わせて TextEncoder オブジェクトを用いて 8 ビット符号なし整数値の配列形式に変換している．マイコンに送信されるバイトコードも同様に 8 ビット符号なし整数値の配列形式に変換する必要があり，本書き込みツールでは変換した結果の配列を変数 ary に格納している．

4. まとめ

動作検証として本書き込みツールを用いたバイナリファイルの書き込みおよび実行を行ったところ，ブラウザから RBoard とシリアル通信が可能なこと，さらにバイトコードの書き込まれた RBoard が想定通りの動作をすることが確認できた．

今後の課題は、本書き込みツールと図 1 に示した Ruby 生成器を連動させることである．具体的には，RBoard 用 Node-RED の生成した JSON コードを本書き込みツールにアップロードするようにし，書き込みツールがバックグラウンドで Ruby 生成器を利用するようにする．現在，共同研究者が IDE の担っていた mruby/c コードのコンパイル作業を Ruby 生成器で行えるようにしており，連携のための準備は整っている．連携の結果として，データフロー型ビジュアルプログラミング環境をブラウザ上で完結させることができ，RBoard 用 Node-RED と本書き込みツールのみを利用して IoT 開発できるようになると期待される．これによりユーザの利便性が向上すると考えられる．

参考文献

[1] 村上旭人，田中和明．小型デバイス向けのデータフロー型プログラミング環境の構築．情報処理学会第 84 回全国大会，2K-08，2022.
[2] “Serial API 手引き書”．https://g200kg.github.io/web-serial-api-ja/EXPLAINER.html，(参照 2023-01-12).
[3] “RBoard”．https://www.sjcinc.co.jp/service/rboard，(参照 2023-01-12).
[4] “mruby/c IDE”．https://www.s-itoc.jp/support/technical-support/mrubyc/mrubyc-report/703，(参照 2023-01-12).
[5] “mrbwrite”，https://github.com/mrubyc/mrbwrite，(参照 2023-01-12).