

Scratch 3.0を用いたプログラミング教育用ロボットカーの 制御システムの開発

福井 真二

情報教育講座

Development of Control System of Robot Car for Programming Education Using Scratch 3.0

Shinji FUKUI

Department of Information Sciences, Aichi University of Education, Kariya 448-8542, Japan

1. はじめに

小学校におけるプログラミング教育の実施が平成29・30・31年改訂学習指導要領 [1] に明記されたことに伴い、令和2年度から小学校でプログラミング教育が始まった。また、中学校では技術・家庭科の技術分野の授業で取り扱われていたプログラミングに関する内容が更に充実され、高等学校では令和4年度から「情報I」でプログラミング教育が実施される予定になっている。これらのことから、学校教育においてプログラミングが重視されてきていることが分かる。

プログラミング教育が注目される中、様々なプログラミング教育用教材が開発されている [2]。筆者らは、ロボットカーを用いたプログラミング教育用教材を開発した [3]。この教材は、ロボットカーを操作するプログラムをScratch [4] によって作成することを通してプログラミングを学習するというものである。文献 [3] で使用しているロボットカーを図1に示す。このロボットカーは文献 [5] を参考に作成されており、Raspberry Pi、モータードライバー及び市販されている安価な部品によって簡単に作成することが可能である。

この教材を使用する場合には、教員が予めロボットカーを作成しておき、児童生徒はロボットカーの動作をScratchで作成したプログラムによって制御することでプログラミングを学習する。作成したプログラムによって実際に動作させることのできるロボットカーがあることで、より興味を持ってプログラミングの学習に取り組むことができる。

文献 [3] では、ロボット制御用プログラムを作成する環境としてScratch 2.0を利用している。最新のScratchのバージョンは3.0であり、Scratch 2.0は古い。本稿執筆時にはまだScratch 2.0を利用すること

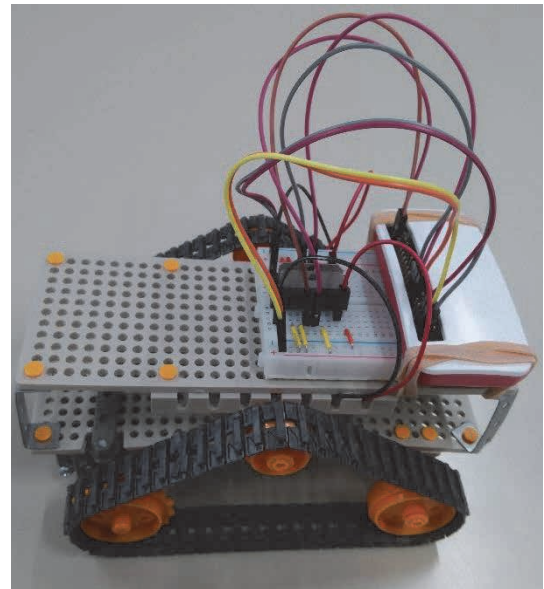


図1 ロボットカー

ができており、しばらくはScratch 2.0でも問題ないと思われる。しかし、Scratch 2.0がいつまで利用可能か分からないし、いつまでメンテナンスされるかも不明である。また、Scratch 2.0を使用しているScratch 3.0で利用できるようになった機能を利用できない。これらの理由から、最新バージョンのScratchでロボットカーを制御できた方がよいと考えられる。

そこで本研究は、Scratch 3.0を用いて作成したプログラムによってロボットカーを制御できるようにロボットカー制御システムを改良することを目的とする。そのために、Scratch 3.0用Webサーバーを導入し、そのWebサーバーで提供されるScratch 3.0にロボットカーを制御するための拡張機能を追加する。そして、その拡張機能を利用できるようにしたScratch 3.0を

使って作成したプログラムによってロボットカーを制御できるようにする。

以降の章では、II. でロボットカー制御システムの概要を、III. で提案するロボットカー制御システムをそれぞれ説明した後、IV. でロボットカー制御システムを使ったロボットカー制御実験について述べ、最後にV. で本論文をまとめる。

II. ロボットカー制御システムの概要

文献 [3] で提案されたロボットカー制御システムの概要を図 2 に示す。

児童生徒がロボットカーを制御するためのプログラムを作成するパソコン (図 2 左側のコンピュータ) に Scratch 2.0 オフラインエディタがインストールされている。その Scratch 2.0 にはロボットカーの動作を制御するための制御用ブロック (以下、制御用ブロックという) 等を追加するための Extension を作成して追加している。これにより、プログラム作成用パソコンで動作する Scratch 2.0 で制御用ブロックが使用できるようになる。制御用ブロックとしては、前進用ブロック、後退用ブロック、右旋回用ブロック、左旋回用ブロックを追加している。それらに加え、後述する helper app と呼ばれる外部プログラムがロボットカー (図 2 右側のロボット) 上の Raspberry Pi と通信するために必要な Raspberry Pi の IP アドレスを設定するためのブロックを Scratch 2.0 に追加している。

児童生徒が作成した Scratch のプログラムを実行している時に制御用ブロックが実行された際には、児童生徒が操作するパソコン上で動作する helper app に対して HTTP 通信を行うようになっている。文献 [3] のシステムで使用している helper app は HTTP サーバーとして動作するよう作成されており、各制御用ブロックが実行された際に発生した HTTP 通信を受け付ける。helper app に対する HTTP 通信は、ブロック毎に異なる URL に対して行われるようになっている。これにより、helper app が HTTP 通信を受信した際に 4 つの制御用ブロックのうちのどのブロックが

実行された際の通信を受信したのか識別できる。

helper app が制御用ブロックからの HTTP 通信を受信した場合、helper app はアクセスされた URL に応じて異なる URL に対して HTTP 通信を行う。この通信は Raspberry Pi 上で動作する HTTP サーバーに対して行われる。Raspberry Pi 上で動作する HTTP サーバーは、アクセスされた URL に応じた制御信号をモータードライバー (TA7291P) に送信することでロボットの動作を制御する。

helper app 及び Raspberry Pi 上で動作する HTTP サーバーは Python で作成されている。Python はプログラミングにあまりなじみのない人にも比較的理解しやすい言語であり、かつ、Python であれば簡易 HTTP サーバーを簡単に構築できるという理由から、Python により 2 つのプログラムが作成されている。

このようなロボットカー制御システムを用いたプログラミング教材を使用すれば、1 人につき 1 台のロボットカーを用意しなくてもよく、安価にプログラミング教育用環境を用意できる。また、ネットワークを利用して制御命令を送信することから、ネットワークに関する授業を行うことも可能である。更に、児童生徒にロボットカーを作成させることにより、プログラミングだけでなく簡単な電気回路の学習もできる。

III. Scratch 3.0 を用いたプログラミング教育用ロボットカー制御システム

文献 [3] で提案されたロボットカーの制御システムは、ロボットカーを動作させるプログラムを Scratch 2.0 で作成している。Scratch の最新バージョンは 3.0 であるので、Scratch 3.0 でプログラムを作成できるようにロボットカー制御システムを改良する。

Scratch 3.0 を利用したロボットカー制御システムの概要を図 3 に示す。提案するロボットカー制御システムは Scratch 3.0 用 Web サーバー (図 3 中の左端のコンピュータ内で動作)、複数台の Scratch 3.0 実行用コンピュータ (図 3 中の中央のコンピュータ)、ロボットカーから構成されている。以下に Scratch 3.0 用 Web サーバー、Scratch 3.0 実行用コンピュータ、ロボットカーについて説明する。

1. Scratch 3.0 用 Web サーバー

Scratch 3.0 にはオンライン版とオフライン版がある。Scratch 3.0 オフライン版は Scratch アプリと呼ばれ、Scratch 3.0 を実行するコンピュータにインストールして使用する。Scratch 3.0 オンライン版は、Scratch 3.0 用 Web サーバーに Web ブラウザでアクセスし、Web ブラウザ上でプログラムを作成し、実行する。

本研究では、プログラミング教育用環境の準備の簡

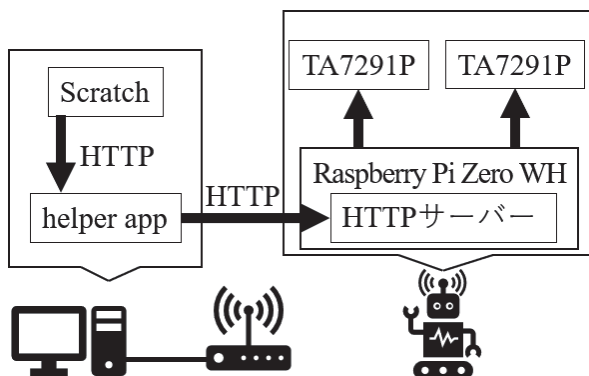


図 2 ロボットカー制御システム [3] の概要

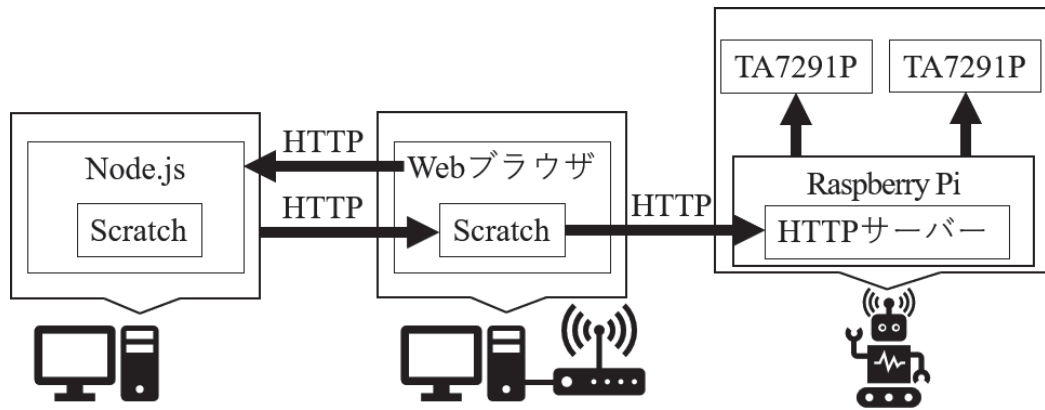


図3 Scratch 3.0 を利用したロボットカー制御システムの概要

便さからScratch 3.0オンライン版を利用する。Scratch アプリを使用する場合、Scratch 3.0実行用コンピュータすべてにScratchアプリをインストールする必要がある。それに対してScratch 3.0オンライン版であればScratch 3.0用Webサーバー1台に対して複数のScratch 3.0実行用コンピュータからアクセスしてScratchを利用することができる。Scratch 3.0はScratch 3.0実行用コンピュータで実行されるため、Scratch 3.0用Webサーバーに対するアクセスを同時刻に集中させなければScratch 3.0用Webサーバーを1台用意すればクラスの全児童生徒用のプログラミング環境を用意することもできる。このように、Scratch 3.0オンライン版であればプログラミング教育環境を簡単に用意できるため、本研究ではScratch 3.0オンライン版を利用する。

標準的なScratch 3.0を利用するのであれば文献 [4] のWebサイトを利用すればよいが、制御用ブロックのようなオリジナルブロックをScratch 3.0に追加して利用する場合、Scratch 3.0用Webサーバーを独自で用意し、ロボットカー制御用拡張機能を開発してそのサーバーが提供するScratch 3.0から利用できるようにする必要がある。それには、Scratch 3.0用Webサーバーが動作するコンピュータにScratch 3.0の開発環境をインストールし、そのサーバーが提供するScratch 3.0から開発した拡張機能を利用できるように設定する必要がある。

ここで、拡張機能を利用可能なScratch 3.0を提供するScratch 3.0用Webサーバーの構築方法を簡単に説明する（詳しくは文献 [6] を参照）。Scratch 3.0用WebサーバーはNode.js [7] で動作しているため、まずScratch 3.0用Webサーバーを動作させるコンピュータにNode.jsをインストールして使用できるようにする。次に、GitHubで公開されているscratch-vm [8] とscratch-gui [9] のリポジトリのクローンを作成してビルドする。これでScratch 3.0の開発環境をセットアップできる。次に、scratch-vm内にオリジナル拡張機能プログラムを作成し、それをScratch 3.0

で利用できるようにscratch-vmとscratch-guiを設定する。その後、scratch-guiを起動すれば、Scratch 3.0用Webサーバーで提供するScratch 3.0でオリジナル拡張機能を利用できるようになる。

本研究では、ロボットカー制御用拡張機能を開発し、Scratch 3.0で利用できるようにする。この拡張機能は、制御用ブロックをScratch 3.0で利用できるようにするためのものである。本研究で作成する制御用ブロックは、文献 [3] と同じく前進、後退、右旋回、左旋回をロボットに指定時間間わせるための4つのブロックである。制御用ブロックがScratch 3.0で作成されたプログラム内で実行された際、実行されたブロック毎に異なるURLでRaspberry Pi上で動作するHTTPサーバーに対してHTTP通信を行うように制御用ブロックは作成されている。アクセス先のURLは文献 [3] のhelper appがアクセスするURLと同じものになっている。これにより、Raspberry Pi上で動作するWebサーバーとして文献 [3] と同じプログラムを利用できる。

2. Scratch 3.0実行用コンピュータ

児童生徒はScratch 3.0実行用コンピュータを利用してプログラムを作成する。

Scratch 3.0用Webサーバーを用意すれば、Scratch 3.0実行用コンピュータではWebブラウザでScratch 3.0用WebサーバーにアクセスするだけでScratch 3.0をWebブラウザ上で実行できる。そのため、Scratch 3.0実行用コンピュータではScratch 3.0を実行可能なWebブラウザさえ利用できればよく、設定変更や他のソフトウェアのインストールをする必要はない。

児童生徒がロボットカー制御用プログラムを作成して実行するまでの手順を説明する。児童生徒はまずScratch 3.0実行用コンピュータでWebブラウザを起動し、Scratch 3.0用サーバーにアクセスしてScratch 3.0オンライン版の画面を表示する。次に、画面左下にある「拡張機能を追加」をクリックしてロボットカー制御用拡張機能を追加する。これにより制御用プロッ

クをScratch 3.0で利用できるようになる。その後、ロボットカー制御プログラムを作成し、そのプログラムを実行する。

3. ロボットカー

本システムで使用するロボットカーは、搭載するRaspberry Piを除いて文献 [3] で使用しているものと同じものである。文献 [3] では、ロボットカーに搭載するRaspberry PiとしてRaspberry Pi ZERO WHを利用していた。これは、Raspberry Pi ZERO WHが安価であるという理由からである。本システムでは、ロボットカーに搭載するRaspberry PiとしてRaspberry Pi 3 Model Bを利用する。その理由は、プログラミング教育用環境の準備が簡単になったからである。その理由を以下に説明する。

ロボットカー制御用拡張機能を利用できるScratch 3.0を提供可能なScratch 3.0用WebサーバーをRaspberry Piで動作させることができれば、Scratch 3.0用Webサーバーを動作させるためのコンピュータを用意する必要がなくなる。そうすれば、Scratch 3.0実行用コンピュータでは何も事前に準備しておく必要がないため、ロボットカーを用意するだけでプログラミング教育用環境を用意することができる。このため、Raspberry Pi上でScratch 3.0用Webサーバーを動作させることを考える。

Raspberry Pi ZERO WH上でScratch 3.0用Webサーバーを動作させようと試みたところ、Scratch 3.0用開発環境を簡単にはセットアップできなかった。調査したところ、64bit版Raspberry Pi OSを利用すればScratch 3.0用開発環境を簡単にセットアップできることが分かった。しかし、64 bit版Raspberry Pi OSはRaspberry Pi ZERO WHには対応していないため、Raspberry Pi ZERO WH上では64 bit版Raspberry Pi OSを動作させることができない。それに対してRaspberry Pi 2 Model B v1.2及びRaspberry Pi 3とRaspberry Pi 4の全モデルは64bit版Raspberry Pi OSに対応している。それらのモデルであれば64bit版Raspberry Pi OSを動作させることができ、Scratch 3.0用開発環境を簡単にセットアップできる。そのため、提案するロボットカー制御システムでは先に挙げた64bit版Raspberry Pi OSの動作するモデルをロボットカーに搭載する。先に挙げたモデルのどのモデルでよいが、本研究では本稿執筆時に筆者が所有していたRaspberry Pi 3 Model Bを利用する。

4. Raspberry Pi上でScratch 3.0用Webサーバーを動作させる場合のロボットカー制御システム

Raspberry Pi上でScratch 3.0用Webサーバーを動作させる場合のロボットカー制御システムの概要を図4に示す。通信の手順や流れは図3で示されているシステ

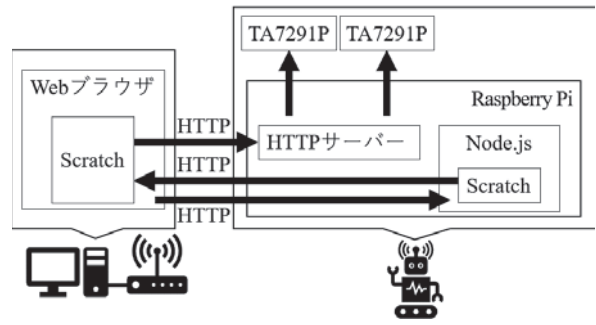


図4 Raspberry Pi 3 Model Bを用いた場合のロボットカー制御システムの概要

ムと同じであるが、ロボットカーに搭載したRaspberry Pi上でScratch 3.0用Webサーバーを動作させているため、システム全体が簡略化されており、ロボットカーをセットアップするだけでプログラミング教育用環境を用意できることが分かる。

機器構成としては文献 [3] のシステムと同じであるが、それと比較しても、本研究で提案するシステムはより簡単にプログラミング教育用環境を準備できる。文献 [3] の教材を利用する場合、児童生徒がプログラムを作成するコンピュータにロボットカー制御用Extensionを組み込んだScratch 2.0オフラインエディタをインストールし、更にhelper appを動作させておく必要があった。それに対して本研究で提案するロボットカー制御システムはロボットカーを準備するだけでよく、より簡単に環境を整えることが可能である。

また、文献 [3] のロボットカー制御システムでは、Scratch 2.0が動作しているコンピュータとロボットカーに搭載されているRaspberry Piが別であるため、helper appがRaspberry Pi上のHTTPサーバーと通信するためにRaspberry Piに設定されているIPアドレスを設定するためのブロックが必要であった。それに対してScratch 3.0用WebサーバーをRaspberry Pi上で動作させた場合、Webブラウザでアクセスした際のScratch 3.0用Webサーバーの情報を使って制御用ブロックのHTTP通信の通信先を設定できる。そのため、Raspberry PiのIPアドレスを設定するためのブロックをなくすことができ、児童生徒がプログラムを作成する際の作業を1つ削減できる。

IV. 実験

本研究で提案するロボットカー制御システムを作成し、実際にロボットカーを動作させることができるか確認するための実験を行った。実験で使用したロボットカーにはRaspberry Pi 3 Model Bを搭載し、その上でロボットカー制御用拡張機能を追加したScratch 3.0用Webサーバーを動作させた。その後、Scratch 3.0

実行用コンピュータ上で動作する Web ブラウザを用いて Scratch 3.0 用 Web サーバーにアクセスし、その Web ブラウザ上で動作する Scratch 3.0 にロボットカー制御用拡張機能を追加した。制御用ブロックを使用できるようにした Scratch 3.0 の画面を図 5 に示す。ロボットカー制御用拡張機能を追加したことで、前進ブロック、後進ブロック、右旋回用ブロック、左旋回用ブロックの 4 つのブロックを Scratch 3.0 に追加できることを確認した。

次に、このロボットカー制御用拡張機能を組み込んだ Scratch 3.0 を使用してプログラムを作成し、ロボットカーを制御できるか確認するための実験を行った。今回テスト用に作成したプログラムを図 6 に示す。このプログラムはロボットカーが前進、右旋回、左旋回、後進をそれぞれ 5 秒ずつ行う、という動作を 2 回行うというものである。このプログラムを実行したところ、ロボットカーが想定通りの動作をした。これにより、Scratch 3.0 で作成したプログラムによってロボットカーの動作を制御できることを確認した。



図 5 Scratch 3.0 に追加した制御用ブロック



図 6 動作テスト用プログラム

V. おわりに

本研究では、プログラミング教育用ロボットカーを Scratch 3.0 オンライン版で作成されたプログラムによって制御するためのロボットカー制御システムを開発した。

Scratch 3.0 でロボットカーを制御するプログラムを作成できるように Scratch 3.0 のためのロボットカー制御用拡張機能を開発した。これにより、Scratch の最新バージョンでロボットカーを動作させることができるようになった。また、Scratch 3.0 用 Web サーバーをロボットカーに搭載している Raspberry Pi 上で動作させた。これにより、ロボットカーを用いたプログラミング教育用環境を簡単に整備できるようになった。それに加え、ロボットカー上の Raspberry Pi に設定されている IP アドレスを設定するためのブロックをなくすことができ、文献 [3] のシステムと比較して児童生徒がプログラミングを作成する際の作業を 1 つ減らすこともできた。

本研究の目的は Scratch 3.0 で作成したプログラムによってロボットカーを制御できるようにすることであった。そのため、文献 [3] と同様に、今後の課題としてロボットカー制御用プログラムの実行途中にロボットカーの動作を停止できるようにする、左右の旋回動作を角度で指定できるようにする、マニュアルを作成する、各種センサーの付いたロボットカーによるプログラミング教育用環境を構築する等が挙げられる。

文 献

- [1] 文部科学省, “平成 29・30・31 年改訂学習指導要領”, https://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm (2021 年 8 月 3 日 閲覧)
- [2] 文部科学省, “教材情報”, 小学校を中心としたプログラミング教育ポータル, <https://miraino-manabi.mext.go.jp/teaching> (2021 年 8 月 10 日 閲覧)
- [3] 福井真二, 藤澤悠貴, “Raspberry Pi を用いたプログラミング教育用教材の開発”, 愛知教育大学研究報告自然科学編, 第 69 輯, pp. 23-27, 2020
- [4] Scratch 財団, “Scratch”, <https://scratch.mit.edu/> (2021 年 8 月 3 日 閲覧)
- [5] 日経クロステック, “ラズパイで作ろう! ゼロから学ぶロボット製作教室”, <http://itpro.nikkeibp.co.jp/atcl/column/15/040800081/> (2021 年 8 月 3 日 閲覧)
- [6] Japanese Scratch-Wiki, “Scratch_3.0 の拡張機能を作ってみよう”, https://ja.scratch-wiki.info/wiki/Scratch_3.0の拡張機能を作ってみよう (2021 年 8 月 5 日 閲覧)
- [7] Node.js, <https://nodejs.org/ja/> (2021 年 8 月 5 日 閲覧)

- [8] scratch-vm, <https://github.com/llk/scratch-vm>
(2021年8月5日 閲覧)
- [9] scratch-gui, <https://github.com/llk/scratch-gui>
(2021年8月5日 閲覧)

(2021年9月14日受理)