

micro:bit のブロックエディター機能の拡張と それを用いた授業の提案

Expansion of 'micro:bit' block editor function and proposal of programming lesson

石井 海渡[†] 赤澤 紀子[†] 山本 博之[‡] 中山 泰一[†]
Kaito Ishii Noriko Akazawa Hiroyuki Yamamoto Yasuichi Nakayama

1. はじめに

1.1 教科情報から考えるプログラミング学習

近年世界の多くの国は情報教育に注力したカリキュラムを取り入れ始めている [1]。日本でも 5 月の未来投資会議 (第 16 回) [2] で、Society5.0 時代を切り拓くため、学習指導要領の改訂によるプログラミング教育・統計教育の充実を掲げている。この中で、小・中・高校を通じて、情報活用能力を言語能力と同様に「学習の基盤となる資質・能力」と位置付けて育成することが以下のように挙げられた。

- 小学校においては、プログラミング教育を必修化
- 中学校においては、技術・家庭科 (技術分野) においてプログラミングに関する内容を拡充
- 高等学校においては、情報科において共通必修科目「情報 I」を新設

中でも高等学校に焦点を置くと、現行学習指導要領では、「社会と情報」および「情報の科学」から 1 科目を選択して履修するという構成であるが、平成 30 年 3 月に公示された次期学習指導要領 [3][4] では、情報と情報技術を問題の発見と解決に活用するための科学的な考え方を育成する科目として「情報 I」、発展的な内容の科目として「情報 II」が設置された。内容は以下の項目で構成されている。

- 情報 I (必修科目)
 1. 情報社会の問題解決
 2. コミュニケーションと情報デザイン
 3. コンピュータとプログラミング
 4. 情報通信ネットワークとデータの活用
- 情報 II (選択科目)
 1. 情報社会の進展と情報技術

[†] 国立大学法人電気通信大学, The University of Electro-Communications

[‡] 東京都立神代高等学校, Tokyo Metropolitan Jindai High School

2. コミュニケーションとコンテンツ

3. 情報とデータサイエンス

4. 情報システムとプログラミング

5. 情報と情報技術を活用した問題発見・解決の探究

さらに、「コンピュータやプログラミング」の身に付ける知識及び技能については「コンピュータや外部装置の仕組みや特徴、コンピュータでの情報の内部表現と計算に関する限界について理解すること」、「アルゴリズムを表現する手段、プログラミングによってコンピュータや情報通信ネットワークを活用する方法について理解し技能を身に付けること」を挙げている。平成 30 年 7 月に公示された高等学校学習指導要領解説 [5] では、AI や IoT の広がりによる新たな時代の到来より、情報社会の進展について学ぶ重要性を述べて、情報産業の創造と発展に主体的かつ協働的に取り組むこと掲げている。

このように、高等学校におけるコンピュータサイエンス教育は、現在よりも高い教育水準が求められることが予想され、それに応じた教員の訓練や学習段階に応じた教材の必要性が課題として挙げられる。

1.2 プログラミング学習ツールの機能拡張

その中新たな教材として、小学生などの初めてプログラミングに触れる方々向けのプログラミング学習ツールが注目されている。その内の 1 つである micro:bit は、イギリスの BBC が開発した教育向けマイコンボードである。イギリスではすでに、2016 年に 11 歳~12 歳の子供全員 (100 万人) に micro:bit を無償配布して、学校の授業で活用している。現在 micro:bit は、イギリスをはじめとして、ヨーロッパの教育現場でも普及している。

micro:bit は日本でも、2017 年 8 月から販売が開始された。また、Micro:bit 財団は 2020 年までに BBC micro:bit を 30 万人の日本の子どもたちに届けることを目指している [6] など、今後の日本での普及に力を入れることを宣言し、日本のプログラミング教育現場での活用が期待される。

そこで本研究では、現状の micro:bit の教育現場での活用を調査するために、micro:bit を使って授業を行って

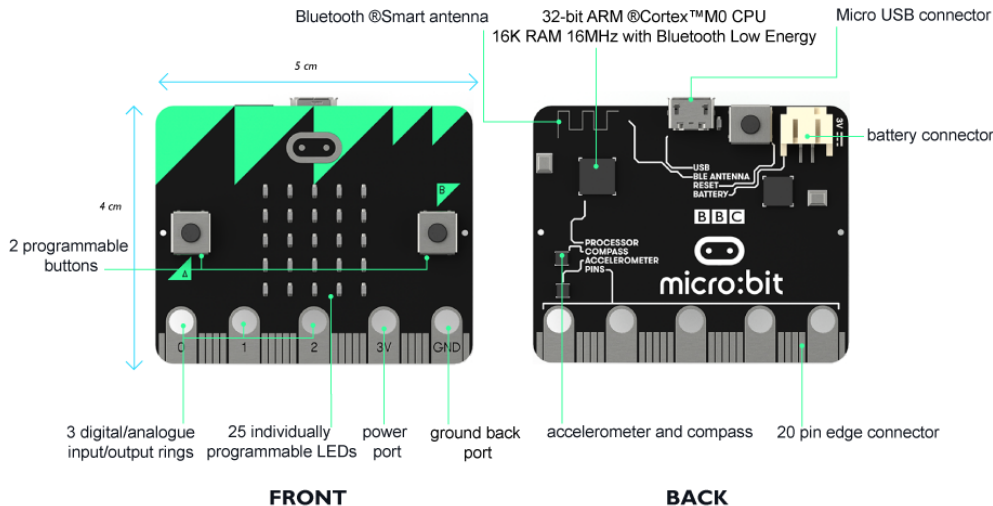


図 1: micro:bit のハードウェア機能

いる東京都立神代高校にヒアリングを実施した。その結果、micro:bit 内のブロックエディター機能には生徒がプログラミングで躓く原因となる問題がいくつか存在し、これらが生徒たちの自由な発想の実現を妨げることがわかった。

また、これらの問題を解決するために、micro:bit のブロックエディター機能の拡張及びそれを使った授業を提案する。ブロックエディター機能の拡張により生徒の躓きが解消でき、プログラミングへの理解や興味を助けることができるかを評価する。

2. micro:bit

2.1 特徴

図 1 のように、マイクロビットは ARM Cortex-M0 プロセッサによって駆動され、256K の不揮発性フラッシュ（プログラムとスタティックデータ用）と 16K の揮発性 RAM（スタック、ヒープ用）を備えている。その他、マイクロビットのハードウェアの特徴として次があげられる。

- テキストのスクロール、ゲームの作成、デジタルストーリーの作成をするための 25 個の赤色 LED
- 2 つのプログラム可能な入力ボタン
- 方向を感知する内蔵磁力計（コンパス）
- 電話やタブレットなどのデバイスや他のマイクロビットに接続する Bluetooth
- 周囲温度を測定するための温度センサー
- 光レベルを検出する LED ディスプレイ



図 2: micro:bit Web 開発ページ

これらのハードウェア機能により、プログラミングの実行結果を触覚的に感じることができる。また、Arduino、Raspberry Pi 等の他のマイコンボードの中でも比較的安価で入手しやすい、学校教師が必要とするスキルが低く訓練の必要性がない、教材が充実しているといった特徴があり学校現場での利用にも適している。プログラミングは、Web 上で動作する専用のエディタで行う [7]。図 2 のように、micro:bit の開発にはビジュアルプログラミングとコードプログラミング (Python) 両方のツールが

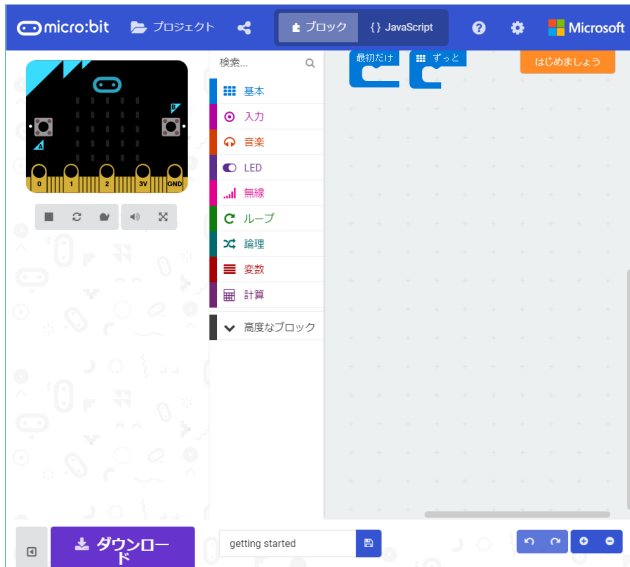


図 3: micro:bit 開発環境

用意されている。本研究では、ビジュアルプログラミング言語の JavaScript Blocks Editor の日本語版を取り扱うこととする。開発環境の様子を図 3 に示す。

2.2 ビジュアルプログラミング言語

ビジュアルプログラミング言語は、ブロックパネルのような視覚的なオブジェクトを組み合わせてプログラミングする言語である。ビジュアルプログラミング言語は初学者にとって特徴として次があげられる。

- ブロックの形や色により、視覚的にわかりやすい
- ブロックピースの組み合わせなので、文法エラーが起きない
- ドラッグアンドドロップなどのマウス操作が主で、タイピングスキルを必要としない
- Web ブラウザ上で動くものが多く、環境構築の手間がない

これらの理由より、小さな子供から大人まで簡単にプログラミング可能で、初学者に向けたプログラミング言語である。

現在、日本でよく使われるビジュアルプログラミング言語として Scratch[8] や Viscuit[9][10] などが挙げられる。micro:bit も JavaScript Blocks Editor では図 4 のような、ビジュアルプログラミング言語を採用しており、プログラミングに慣れていない子供や教師でも直感的にプログラミングを体験できるようにしている。



図 4: micro:bit のサンプルプログラム

3. 関連研究

3.1 Microsoft Touch Develop and the BBC micro:bit

Thomas Ball ら [11] は、Microsoft のスマートフォンやタブレットやマイクロビットなどの最新デバイスにプログラミングをして、楽しさを学習する Touch Develop プロジェクトがどのように機能するかを評価した。そこで、ソフトウェア工学のハードルとアーキテクチャについて、子供や教師ができるだけアクセスしやすいように設計しなければならないことを述べた。ビジュアル言語からテキストベース言語へのシームレスな移行を利点を挙げ、そのような独立されたプラットフォームはプログラミングに慣れていない子供や教師にとって不可欠なものであることも述べた。

3.2 .NET Gadeteer: A New Platform for K-12 Computer Science Education

Gadeteer システム [12] は、Microsoft .NET Gadeteer と呼ばれるプロトタイピング用ツールキットを、高等学校の情報の授業に使用した経験を紹介した。そこで、学生や子供たちが画面上でコードを見ることに加えて、「コンピュータ」を見たり、触ったり、変更したりすることができる活動に魅了された証拠を提示し、小型コンピュータに対する関心の高まりを示した。

3.3 教員のためのプログラミング教室の実践報告

赤澤ら [13] は、初めてプログラミングを習う小学生を対象に公立小学校でのプログラミング教室と子供にプログラミングを教える教員に向けた講義を行った。そこで



図 5: micro:bit の「無線で送信」ブロックを使ったプログラム例

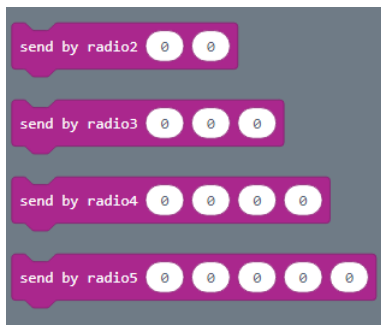


図 6: 解決案として提示するブロック例 1

micro:bit を含む複数の初学者向けプログラミング学習ツールを紹介した。アンケートの結果、これらのツールは楽しく、現場でも使ってみたいとの評価を得た。

4. ヒアリング結果

4.1 学校現場での活用をヒアリング

実際に micro:bit を使った授業を行った学校（東京都立神代高等学校）の情報教員に対してヒアリングを行った。ヒアリングを行った高校では、micro:bit の基礎的な使い方を学習した後に、生徒たちによる自由作成の課題を行っている。その課題授業後のアンケートでは以下のようなコメントが見受けられた。

- 「自分がしてほしいプログラミングの指示のボタンがない」
- 「やりたいことが先生に難しいと言われたのであきらめて終わりにした」
- 「アイデアは浮かんだけど実行するのが難しそう」
- 「『こうしたらこうなるんじゃないか？』っていう考えはたくさんあるのに、それがプログラミングになると上手に受送信ができなくてもうつらい」

これらのコメントはアンケート全体の一部であり、類似したコメントは他にも複数見られた。これらコメント

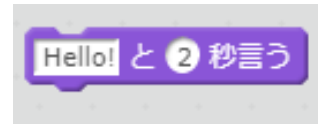


図 7: Scratch での「表示する」ブロック



図 8: 解決案として提示するブロック例 2

と情報教員からのヒアリングにより、micro:bit の一部機能にブロックエディター特有の問題があり、生徒たちの自由な発想の実現を妨げることがあることがわかった。これら問題は、工夫をこらすことで突破することもできるが、高校生の理解度と授業時間数の都合を考えるとこれらを指導するのは大変困難である。また、ヒアリングを行った高校では、micro:bit の授業よりも前に、同じビジュアルプログラミング言語である Scratch の学習を行っている。生徒はアイデアを Scratch では難なく実現できるが、micro:bit では躓いてしまう場合があるという結果も見られた。

4.2 問題の具体例と解決案

ヒアリングにより、分かった問題を以下に大きく 4 つにまとめた。それぞれをビジュアルプログラミング言語で代表的である Scratch を例に出すなどして解決案を考察する。

4.2.1 無線機能で複数の値を同時に送受信できない

micro:bit 内の「無線で送信」ブロックでは図 5 のように、送信側が送れる情報が、1 つの変数、1 つの文字列、1 つの変数と 1 つの文字列、の 3 通りしか存在しない。複数の変数を送受信したい場合、例えば x 座標と y 座標を送りたい場合には、送信側が $x * 10 + y = z$ のように、2 つの情報を新たな変数に格納して送り、受信側がその格納された変数を分解して、元の情報を読み出す手続きが必要になる。

この問題の解決案としては図 6 のような、「無線で送信」ブロックの引数の数を増やしたブロックを新たに用意するなどの方法が考えられる。

4.2.2 表示する時間を調節できない

micro:bit 内の「数を表示する」ブロックでは、表示する時間が固定で定められている。表示する時間を短

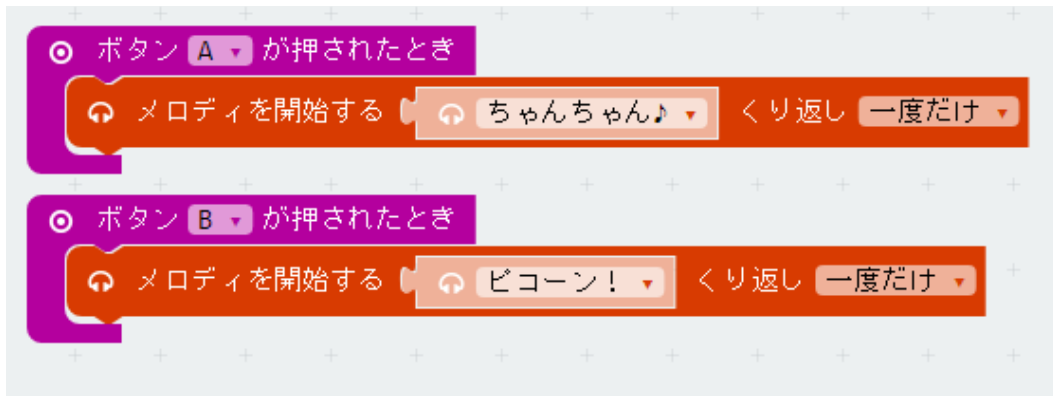


図 9: micro:bit の「音を鳴らす」ブロックを使ったプログラム例

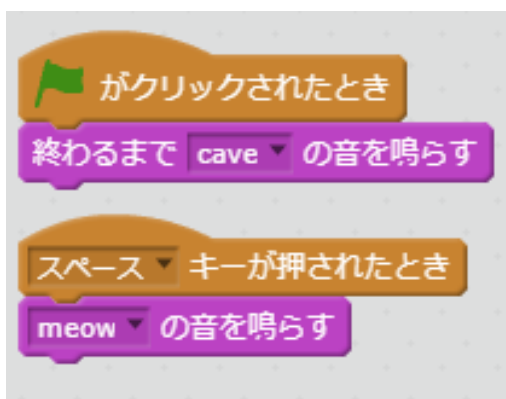


図 10: Scratch の「音を鳴らす」ブロックを使ったプログラム例

くしたい場合や長くしたい場合、例えば瞬時に変わる数をタイミングよく止めるスロットゲームのようなものを作りたい場合、実現させることができない。この問題は Scratch では解決でき、図 7 のように、表示する変数と時間を引数にとるブロックが用意されている。

この問題の解決案として図 8 のような、「数を表示する」ブロックの表示時間を変数で調節できるように変えたブロックを新たに用意するなどの方法が考えられる。

4.2.3 メインの音ブロックを再生中にサブ音ブロックを割り込めない

micro:bit 内の「音を鳴らす」ブロックは、図 9 のようなプログラムで実行するが、音の再生中に別の音を再生すると元の音が強制終了させられる。元の音を強制終了させたくない場合、例えばメインの BGM 音を鳴らしながら、サブの SE 音を鳴らしたい場合、実現させることができない。この問題は Scratch では解決でき、図 10 のような、「音を鳴らす」ブロックと別の「終わるまで音を鳴らす」ブロックが用意されている。これにより「終

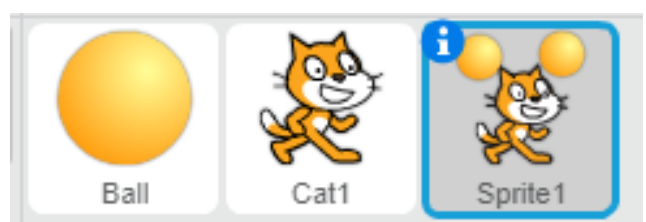


図 11: Scratch の複数スプライトの統合例

わるまで音を鳴らす」ブロックの実行中に別の音が割り込まれても終了することがなく、2つの音を平行して鳴らすことができる。

この問題の解決案は現在検討中である。この問題の解決には、ハードウェア的要因があり、micro:bit が一度に 1 つのイベントしか実行できない場合に、2 つの音を同時に切り替えて疑似的に平行して音が鳴っているようにするなどの工夫が考えられる。

4.2.4 複数ドットの塊を一つのスプライトとして扱えない

スプライトとは、LED のドットをキャラクタ化して操作する機能のことである。これによりボタンによる操作や当たり判定の判別など、ゲーム的要素を簡単に作ることができる。micro:bit 内の「スプライトを作成」ブロックでは、1 ドット分しかスプライト化することができない。複数ドットをスプライト化したい場合、例えば複数ドットをグループ化して 1 つのスプライトの様に扱いたい場合、1 ドットずつ動作を作り、それをあたかもグループで動いているかのように見せかける労力が必要である。この問題は Scratch では解決でき、図 11 のように画像 1 つをスプライトとして扱っているため、複数画像を重ね合わせたものも問題なく 1 つ画像（スプライト）として扱える。

この問題の解決案は現在検討中である。

5. 解決案の実装

解決案の実装は、オープンソースで公開されている Javascript Block Editor を改造する方法で行う。Javascript Block Editor の開発環境 [14] には、ブロックの拡張に必要となる型のテンプレート [15] が用意されている。テンプレートを元に、ブロックの形状や動作についてプログラミングを行う。なお、既存のブロックやテンプレートでは、micro:bit の動作に関する部分は TypeScript、LED やセンサーなどのハードウェアの制御に関する部分は C++ で実装されており、本研究でもこれらの言語を用いて実装する。実装したブロックがテスト用ローカルサーバーで正しく機能することを確認する。実験を行う際には、ローカルサーバーにアクセスして、実装したブロックを使用する。また、実装したブロックは公式に承認されることによりパッケージとして公開することができる。一般ユーザーは検索によりパッケージ化されたブロックを自身の micro:bit 開発環境に拡張する。これを利用して実験を行うかは現在検討中である。

6. 評価方法について

ヒアリングを行った高校にて、拡張した micro:bit のブロックエディター機能を使った実験授業の実施を検討する。詳細な実験方法と授業内容は現在検討中である。生徒の中でどれだけの比率がこれらの問題に躓いているのかとその躓きを解消することで、プログラミングへの理解や興味を助けることができるかを評価する。

参考文献

- [1] 久野靖, 和田勉, 中山泰一: 初等中等段階を通じた情報教育の必要性和カリキュラム体系の提案, 情報処理学会論文誌「教育とコンピュータ」, Vol. 1, No. 3, pp. 48-61 (2015).
- [2] 首相官邸: 未来投資会議 (オンライン), 入手先 <https://www.kantei.go.jp/jp/singi/keizaisaisei/miraitoshikaigi/> (参照 2018-07-27).
- [3] 文部科学省: 高等学校学習指導要領 (平成 30 年 公示)(オンライン), 入手先 http://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm (参照 2018-07-27).
- [4] 鹿野利春: 学習指導要領の改訂と共通教科情報科, 情報処理, Vol.58, No.7, pp.626-629 (2017).
- [5] 文部科学省: 高等学校学習指導要領解説 (平成 30 年 公示)(オンライン), 入手先 http://www.mext.go.jp/a_menu/shotou/new-cs/1407074.htm (参照 2018-07-27).
- [6] Micro:bit Educational Foundation: 小型ボード型コンピュータ micro:bit が日本上陸 (オンライン), 入手先 <http://microbit.org/ja/2017-08-03-japan-launch/> (参照 2018-07-11).
- [7] Micro:bit Educational Foundation: プログラムしましょう — micro:bit (オンライン), 入手先 <https://microbit.org/ja/code/> (参照 2018-07-11).

- [8] MIT メディアラボ: Scratch - Imagine, Program, Share (オンライン), 入手先 <https://scratch.mit.edu/> (参照 2018-07-27).
- [9] デジタルポケット: ビスケット Viscuit (オンライン), 入手先 <https://www.viscuit.com/> (参照 2018-07-27).
- [10] 原田康徳, 勝沼奈緒実, 久野靖: 公立小学校の課外活動における非専門家によるプログラミング教育, 情報処理学会論文誌, Vol.55, No.8, 1765-1777 (2014).
- [11] Ball, T. Protzenko, J. Bishop, J. et al.: Microsoft Touch Develop and the BBC micro:bit, ICSE-C, (2016).
- [12] Hodgesa, S. Scotta, J. Sentanceb, S. et al.: NET Gadgeteer: A New Platform for K-12 Computer Science Education, SIGCSE '13 Proceedings of the 44th ACM Technical Symposium on Computer Science Education, 391-396, (2013).
- [13] 赤澤紀子, 石井海渡, 小早川祐一郎, 中山泰一: 教員のためのプログラミング教室の実践報告, 情報処理学会情報教育シンポジウム SSS2018 (2018).
- [14] Microsoft: Microsoft MakeCode (online), available from <https://www.microsoft.com/en-us/makecode> (accessed 2018-07-11).
- [15] Microsoft: MakeCode Playground (online), available from <https://makecode.com/playground> (accessed 2018-07-11).