

## micro:bit を複数用いる場合のプログラム開発について

松永 豊  
情報教育講座

## Program Development for Multiple micro:bits

Yutaka MATSUNAGA

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

## 要 約

micro:bit は優れたプログラミング教育ツールであり、様々な実践事例が報告されているが、micro:bit の特徴的な機能の一つである無線通信を用いて複数台で演習する内容で授業設計をする場合、設計段階でかなり手間がかかるかもしれない。その問題を共有したうえで原因や対策について議論する。提案した方法を用いれば設計時に発生する手数を大幅に軽減できる可能性がある。

Keywords: プログラミング教育、micro:bit、無線通信

## I はじめに

近年、コンピュータの進化は目覚ましいものがある。生成 AI をはじめとした知的情報処理技術も急速に進んでおり、今後様々な職業が AI によって変革されたり無くなったりすると考えられている。また、逆に AI 開発等の分野では人材が全く足りておらず、世界中で言わばパイの奪い合い状態が続いている。

このような背景のもと、世界中でプログラミング教育の低年齢化が急速に進んでいる[1]。日本でも学習指導要領改訂に伴い、2020 年度から小学校でプログラミング教育が必修となった[2][3][4][5]。また、中学校技術においてもプログラミング教育はほぼ倍増、高校でも情報 I でプログラミング教育が必修となり、選択の情報 II でさらに難しいプログラミングを扱うこととなる。また、大学入学共通テストにおいても、2025 年度よりプログラミングの知識を含む「情報」の科目が追加される[6][7]。

小学校でプログラミングを教える場合、低学年や中学年でアンプラグドやブロックプログラミングを扱うことが多いが[8][9][10][11][12][13]、中学校のプログラミングにおいては「計測」が含まれていることから、実機との相性、校種間の接続、などを理由として、小学校の高学年においても、ロボット等の実機を用いる学習をとることも多い[14][15]。このとき、扱いやすさやコストパフォーマンスの観点から micro:bit は最も選択される選択肢の一つと考えられている。

micro:bit を用いた実践はすでに複数あるが[7][15]、micro:bit は非常に高性能であるため、あまり使われていない機能も多々ある。そこで、本研究では無線通信など micro:bit を複数用いる場合のプログラム開発に着目し、様々な問題提起、提案、議論等を行うことを目的とする。

## II micro:bit の基本性能

micro:bit はイギリスの国営放送局 BBC が中心として開発された小型のコンピュータであり、プログラミングの学習や教育に広く使われている。5×5 の 25 個の LED により多彩な表現が可能である。また、多種多様なセンサーが標準で搭載されている。光センサー、温度センサー、加速度センサー、磁力センサー、などセンサーが豊富であり、2 つのボタンを含め様々な入力インターフェイスを利用することができる。

表 1 主な仕様

項目	内容
LED	25 個 (5×5)
ボタン スイッチ	A ボタン、B ボタン (裏面のリセットボタンはリセットにしか使えない)
センサー	光センサー 温度センサー 加速度センサー 磁力センサー
主なプロ グラミン グ言語	makecode (Block Editor) Python JavaScript
無線通信 機能	BLE
V2 専用	スピーカー マイク
バッテリ ー	3V (USB または電池)

Python や JavaScript など実用テキスト言語を用いても操作できるが、makecode などブロックを用いたビジュアルプログラミング開発環境も用意されているため、小学校のプログラミング教育でも数多くの実践事例が存在している。

25 個の LED は自由に光らせることができ、また、表示命令には敢えて 0.5 秒程度の一時停止が含まれているため、アイコン描画命令等をただ並べてもいい感じにアニメーションしてくれるため授業でもよく使われている。入力インターフェイスとしては、ボタン、ゆさぶる、明るさなどがよく使われている。また、V2 ではスピーカーが標準搭載されているので音関連のブロックも使えるが、V1.5 の場合でもワノクリップ等で端子とスピーカーを接続させて音を鳴らす授業を行っている事例も多い。

### Ⅲ 無線通信機能

micro:bit には無線通信機能 BLE が標準で搭載されており、複数の micro:bit 間で無線通信を行うことができるため、この機能を使った対戦ゲームのようなものも作成できる。

小学校のプログラミング教育等でこの無線通信を使った授業を行う場合、授業設計の段階から様々なことを計画しておかなくてはならない。

まず、「接続させるのは 2 台 (ペア) なのか 3 台以上なのか」でプログラムの作り方が大きく異なる。micro:bit では無線グループが設定できるため、同じ数字のグループでは通信が可能となる。ペアで実習する場合は例えば 40 人クラスならば、無線グループを 1~20 で 2 人ずつ割り当てればペアの相手以外とは通信できなくなる。これにより、「ペアの場合は返事があれば相手」と判断できるためプログラムが簡単になる。一方、無線グループに 3 台以上割り当てた場合も接続自体はすぐにできる。ただし、「3 台以上の場合返事がきても誰が返事をしてきたか確認できる機構が必要」となるためアルゴリズムレベルにおいて大きくことなるプログラムとなる。

また、「自分を識別する数値として変数 id を用意した場合、その値は誰がセットするか」によっても処理が分かれる。方法としては、ソースコードの段階で児童自身にそれぞれ入力してもらう方法と、何らかの条件で HOST と CLIENT が自動的に割り振られる方法の 2 つの方法が挙げられる。前者はプログラミングの難易度が軽減するメリットはあるが (図 1)、開発段階においてはやや厄介な問題が発生する (後述)。一方、例えばボタンを早く推したほうが HOST になるなど、自動的に割り振るプログラムの場合は同一のプログラムでよい (ように作る) ことがメリットになる。しかしながら、ソースコードに直接数値を入れてもらう方式よりは複雑になるし、ペアならばまだしも、3 名以上の場合かなり複雑なプログラムになってしまうデメリットが存在する。

以上のことから、micro:bit を複数用いてプログラムを開発する場合は、以下の手順を提案する。なお、ここでは 4 台の micro:bit を同時に扱うものとし、USB 接続によって認識されるドライブレターは、D, E, F, G になったと仮定する。また、後述のシリアル通信においては、COM5~COM8 が割り当てられたと仮定する。

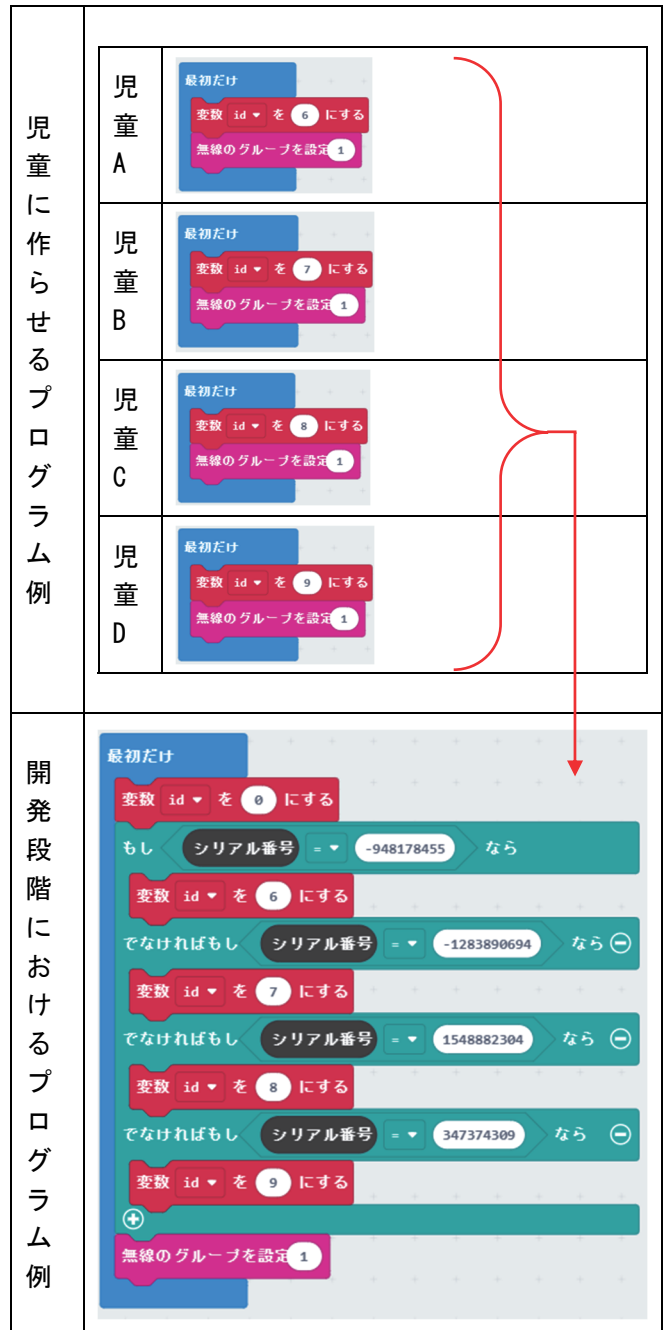


図 1 シリアル番号を使ったプログラム

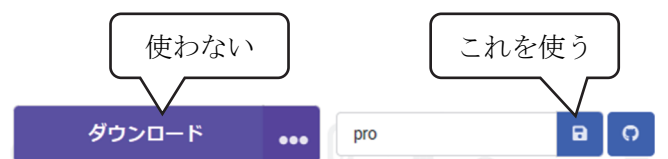


図 2 ダウンロードの方法

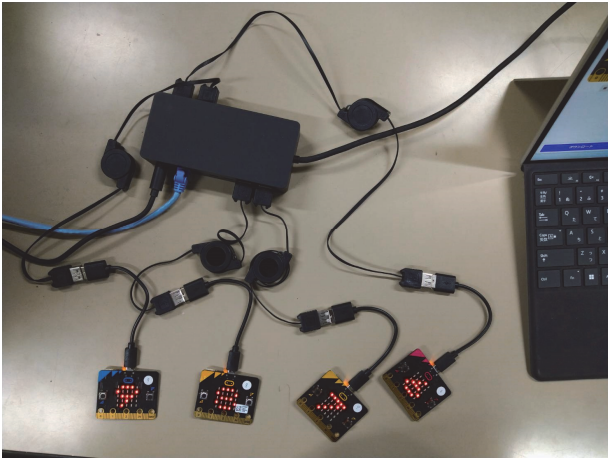


図3 micro:bit 4台同時開発の様子

手順

- micro:bit を4台とも USB 接続する。(図3)  
(ドライブレターがD, E, F, G になったとする)
- このとき、makecode の「デバイスを接続する」をしないでおく。
- シリアル番号を用いた分岐プログラムを「最初だけ」に入れておく(図1下参照)。なお、シリアル番号は後述の方法等を用いてあらかじめ調べておく必要がある。
- その他の部分(メイン部分)を作成する。
- プログラムを micro:bit に転送させたい場合は、「ダウンロード」ボタンではなく、保存ボタンを用いて HEX ファイルを一旦ハードディスクにダウンロードし、micro:bit にはこの HEX ファイルを直接コピーすればよい。(図2)

なお、HEX ファイルを micro:bit に転送するにはエクスプローラーなどでドラッグドロップすれば可能だが、これもバッチファイルや Python 辺りを使って自動化しておくとお楽である。(図4)


```
#HEX ファイル転送プログラム
import shutil
import os

file0= r' C:\Users\%xxxx%\Downloads\microbit-pro. hex'
file1= r' C:\Users\%xxxx%\Downloads\%a. hex'
if os. path. exists( file0 ):
    if os. path. exists( file1 ):
        os. remove( file1 )
    shutil. move( file0, file1 )
shutil. copy( file1, ' d: ' )
shutil. copy( file1, ' e: ' )
shutil. copy( file1, ' f: ' )
shutil. copy( file1, ' g: ' )
```

図4 HEX ファイル転送用 Python プログラム

なお、シリアル番号を取得するには「数を表示」ブロックで LED に表示させてメモを取る方法でも構わないが、10桁ほどの数字がスクロールしながら表示されるので意外と面倒である。そこで、シリアル通信で PC に転送する方法が楽である。具体的には、micro:bit 側でシリアル通信 1 行書き出すプログラムを用意し、シリアル番号を送ればよい(図5)。受信する PC 側は TeraTerm[16]などで確認しても構わないが(図6)、Python 辺りでも簡単に作ることができる(図5)。ただし、システムによっては COM ポートが異なるかもしれないので、それを考慮したプログラムに書き換えるか、うまくいかない場合は結局 TeraTerm を使うことになるかもしれない。

micro:bit 側



# シリアル番号取得プログラム (PC 側)

```
import serial
import time

# COM5~COM8 は状況で異なる
for com in [ 'COM5', 'COM6', 'COM7', 'COM8' ]:
    print( com )
    ser= serial. Serial( com, 115200, timeout=3 )
    for i in range( 5 ):
        line= ser. readline()
        print( line )
        time. sleep( 1 )
    ser. close()
```

図5 シリアル番号取得プログラム

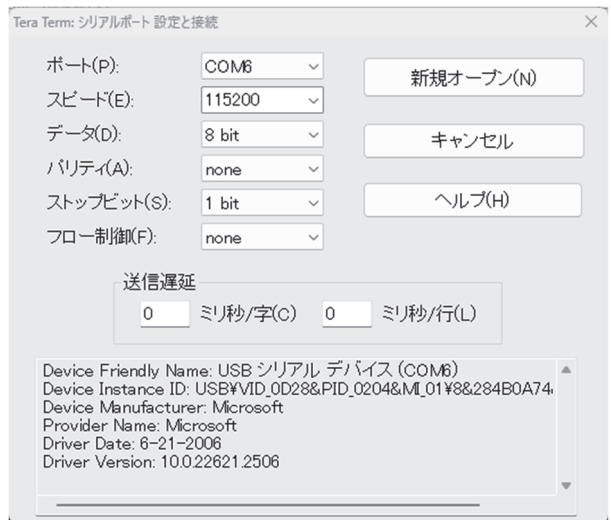


図6 TeraTerm のシリアルポート設定画面

#### IV ディスカッション

今回提案した手順についてより深く考えてみよう。それには、「今回提案した手順を用いない場合」を想像してみると分かりやすい。具体的には、変数 id を児童ごと (micro:bit ごと) に異なる数値を使いたい場合である。

まず、実践の際 (授業時)、変数 id にバラバラの数値を使わせるよう児童に指示することに関しては特に問題ない。「君は何番を使ってね」「君は何番を使ってね」とそれぞれ指示するだけでよい。

一方、教師が授業構築する際には手順の違いが大幅に影響する。そこで、

- ① PC4 台を用いて作成する方法
- ② PC は 1 台だが児童と同じように変数 id に異なる数値を入力させる方法
- ③ 今回提案した方法

の 3 つで比較することにする。

まず、①であるが、この方法ではそれぞれの micro:bit に対して makecode の「デバイスを接続する」が使えるため、「ダウンロード」ボタンが使えるというメリットがある。しかしながら、開発中はしょっちゅうプログラムを書き換えるため、その都度他の 3 台のマシンでも同じようにプログラムを書き換える必要が生じるためかなり面倒である。特に 3 台以上の micro:bit を使うプログラムにおいては顕著になる。makecode には優れた micro:bit エミュレータが搭載されているため、エミュレータ上で動作確認を済ませてしまう場合も多い。micro:bit が 2 台までであればエミュレータ上である程度は開発できるため実機への転送の回数は軽減する。しかしながら、3 台以上で動作確認する場合にエミュレータは使えないため、必ず実機に転送してから確認することになる。このことは②③でも同様である。

次に②であるが、異なる micro:bit では数値が異なるわけだからダウンロードごとにソースコードを書き換える必要がある。また、1 台しかコネクต์できない以上、HEX ファイル経由で転送しなくてはならない点は①より能率が悪い。

最後に③であるが、シリアル番号ごとに自動的に変数 id の値を変えているので、同一のプログラムで問題ない。つまり、HEX ファイルを一つだけダウンロードして 4 台の micro:bit に転送すればよいので、プログラムの書き換え等が不要となる。また、HEX ファイル転送プログラムを併用することで 4 回発生するコピー作業も 1 度の操作で自動的に行われることになるので、負担はさらに軽減する。また、開発が完了してしまえば、シリアル番号による分岐処理は不要となる。すなわち、児童にプログラミング (先生が作ったひな形の改造の場合も含め) をさせる際にはシリアル番号による分岐処理の部分は全て排除して児童それぞれに数値を入力させることになるので、難易度を抑えることもできるし、プログラムもスッキリする。

#### V まとめ

以上、本研究では無線通信を扱うような複数台の micro:bit を同時に開発する際の開発手順について提案した。提案した方法を用いれば、開発段階では作業効率は大幅に向上するにも関わらず、実際の児童が行う演習においてはただ数値を代入させるだけで済むため難易度上昇を軽減することができる。

micro:bit の無線通信のような比較的複雑な内容を小学校プログラミング教育で導入する場合、あらかじめ膨大な実験と検証を行う必要がある。今回は開発段階の話を中心に議論を進めたが、本論文執筆中に「科学・ものづくりフェスタ@愛教大」および「あつまれ！子どもキャンパス」に出展しており、先日実践を行ったところである。実践直後なので、現在、データ分析中であるが、詳細は後日報告する予定である。

#### 謝辞

本研究の一部は JSPS 科研費 (JP20K03207, 代表: 松永豊) の助成を受けたものである。

#### 参考文献

- [1] 松永豊, 低学年向け教育を意識した情報科学教材に関する一考察, 愛教大研究報告. 教育科学編, 68 輯, 103-108, 2019
- [2] 梅田恭子, 齋藤ひとみ, 高橋岳之, 松永豊, 野崎浩成, 福井真二, ICT活用指導力アップ, 書籍, 実教出版, 2019
- [3] 磯部征尊, プログラミング教育における教育課程基準の検討と実践, 日本教育公務員弘済会研究成果報告書, 2018
- [4] 松永豊, 小学校プログラミングの指導法に関する一考察, 愛教大研究報告. 教育科学編, 66 輯, 157-161, 2017
- [5] 松永豊, 梅田恭子, 齋藤ひとみ, 磯部征尊, 教員を目指す学生に対するプログラミング教育の指導法について, 愛教大教職キャリアセンター紀要 4, 91-96, 2019
- [6] 中西渉, DNCL2 (仮称) の学習環境の実装に関する考察, 情報教育シンポジウム, 8, 254-260, 2021
- [7] 安本太一, 磯部征尊, 梅田恭子, 鎌田敏之, 齋藤ひとみ, 松永豊, 小学校・中学校・高等学校間の接続を考慮したプログラミングの授業の提案, 情報教育シンポジウム, 8 月, 36-43, 2021
- [8] 松永豊, 松下侑樹, 鈴木秀弥, 教職大学院生による ICT を活用した実践について, 愛教大教職キャリアセンター紀要 7, 35-40, 2022
- [9] 齋藤ひとみ, 野々垣真帆, 演劇的手法を用いたアンプラグド・プログラミング教育:ものづくりフェスタでの実践, 愛教大研究報告. 教育科学編, 68 輯, 95-101, 2019

- [10] 梅田恭子, 米澤和志, 齋藤ひとみ, 松永豊, 磯部征尊, 児童のプログラミング的思考の育成を目指した設計シートの開発, 日本情報科教育学会, 2019
- [11] 松永豊, 磯部征尊, 梅田恭子, 齋藤ひとみ, 小学校プログラミング教育におけるメンター育成および実践授業について, 愛教大教職キャリアセンター紀要 3, 75-80, 2018
- [12] 齋藤ひとみ, 梅田恭子, 松永豊, 磯部征尊, プログラミング学習における児童の自己評価と他者評価, 日本教育工学会研究報告集, 71-75, 2018
- [13] 齋藤ひとみ, 大場光樹, 松永豊, 教材の違いによるプログラミング教育の学習効果の比較, 愛知教育大学研究報告, 教育科学編, 71, 116-124, 2022
- [14] 日比野浩規, 景井美帆, 松永豊, 山崎貞登, 磯部征尊, テクノロジーとエンジニアリングの教室 第1巻, 日本産業技術教育学会 実践事例集, 2021
- [15] 松永豊, 鈴木秀弥, 教職大学院生による校種間の接続を考慮したプログラミング教育の実践について, 愛教大教職キャリアセンター紀要 8, 155-160, 2023
- [16] <https://teratermproject.github.io/>