Vol.2021-CE-161 No.7
2021/10/23

Bit Arrowにおける組み込み機器実行機能と，収集データの共有・分析機能との連携

長 慎也[1,a)]　岸本 有生[2]　長島 和平[3]　兼宗 進[4]　並木 美太郎[3]

概要：
来年度から始まる高校教科「情報」では，「プログラム」の理解・習得だけではなく，データの「収集，整理，分析の方法」の理解・習得も目標とされており，実習環境にはこれらの活動を統合的にサポートすることが求められている．
筆者らは現在，Web ブラウザを用いてプログラミング学習が可能な環境「Bit Arrow」を，初学者向けのプログラミング学習活動に用いている．Bit Arrow はこれまで，Web ブラウザ内での実行を主軸としつつ，データの収集源として組み込み機器からのセンサデータの収集や，データ分析などの一部の処理を Web サーバ側でも行えるような実行系を備えていた．また，データ分析に必要なファイル（素材）をアップロードし，共有するための仕組みも備えていた．しかしこれまでは，データ収集が可能な組み込み機器の種類に制限がある，素材データを容易に整理・共有するための機能が不足している，という問題点もあった．
今回の発表では，Bit Arrow に新たに追加された 2 つの機能について紹介する．1 つは，組み込み機器である Raspberry Pi Pico の Python プログラムを Bit Arrow から実行する機能，もう 1 つは，データの統計的な分析の学習をブラウザやスマートフォンから手軽に行える「ConnectDB」との連携機能である．これらの 2 つの機能によって，Bit Arrow が教科「情報」における，データの「収集・整理・分析」を統合的に支援するプラットフォームになることが期待される．

Data collection from embedded-system an data sharing and analysis in Bit Arrow

CHO Shinya[1,a)]　KISHIMOTO Tomonari[2]　NAGASHIMA Kazuhei[3]　KANEMUNE Susumu[4]
NAMIKI Mitarou[3]

1. はじめに

高校教科「情報」は，情報 I と情報 II に分けられているが，情報 I においては必修科目とされており，すべての高校生が履修すべき内容となっている．

その目標には，「効果的なコミュニケーションの実現，コンピュータやデータの活用について理解を深め技能を習得するとともに，情報社会と人との関わりについて理解を深めるようにする」と挙げられている [1]．これは，次のような事柄について「理解し，技能を身に付ける」ことを目標としている [2].

- コンピュータを活用するために必要な情報が処理される仕組み
- データを活用するために必要な収集，整理，分析の方法
- プログラム
- モデル化とシミュレーション，
- ネットワーク，
- データベース

[1] 明星大学
Meisei University, Japan
[2] 大阪電気通信大学高等学校
Osaka Electro-Communicatuon University High School.
[3] 東京農工大学
Tokyo University of Agriculture and Technology, Japan
[4] 大阪電気通信大学
Osaka Electro-Communication University, Japan
[a)] cho@eplang.jp

ⓒ 2021 Information Processing Society of Japan

ここで，目標としているものには，「プログラム」の理解・習得だけではなく，データの「収集，整理，分析の方法」の理解・習得も含まれている点に着目したい．つまり，教員や生徒が活動を行うために必要な環境は，プログラミング環境を提供するだけでなく，「収集，整理，分析」を支援する環境である必要がある，と言うことができる．

筆者らが開発している Bit Arrow は，Web ブラウザ上で動作するプログラミング環境であるが，単にプログラムを書いて実行する，という機能にとどまらず，データの「収集，整理，分析」もサポートしている．

Bit Arrow におけるデータの**収集** は，「素材管理」（**図 1**）という，データを学習者や教員が自由にアップロードできる機能を備えている．アップロード先は「user」（各学習者専用フォルダ）と「class」（クラス全体で共有されるフォルダ）があり，学習者が自分のデータで独自の分析を行ったり，教員が分析用のデータを学習者に配布したりすることが可能になる．

図 1　素材管理

Bit Arrow には「素材管理」の他に，KVS 方式の簡易データベースも備えており，学習者が JavaScript(教育用 JavaScript) のプログラムを作成することで，データ入力用のユーザインタフェース（Web フォーム）を通じてして入力することもできる．簡易データベースに書き込むための Web API も提供しており，ネットワーク接続機能を持った組み込み機器であれば，センサーデータを Bit Arrow に送信することも可能である [3]

データの**整理** については，先述した素材管理にアップロードされているファイルを，Bit Arrow 上で書かれた Python，C，教育用 JavaScript などのプログラムからアクセスする機能も提供されているため，収集したデータを表計算，テキスト（tsv/csv/JSON）など，プログラムで処理できる形式に変換・保存することも可能である．

データの**分析**については，Python では統計分析ライブラリ（numpy, scipy， pandas など）を用いることができ，scikit-learn などの機械学習のライブラリも備えている．データを可視化する手段として，Python の matplotlib などのグラフライブラリを備えている．また教育用 JavaScript においてもグラフを簡単に表示できるライブラリを用意している．教育用 JavaScript はブラウザ上で動作するため，アニメーションを用いたデータの可視化も行うことができる．

2. Bit Arrow の課題と解決方法

このように，データの「収集，整理，分析」を支援してきた Bit Arrow であるが，これまでは次のような機能が不足していた．

- **ネットワーク機能のない組み込み機器からのデータ入力**
  以前の実践 [3] では，組み込み機器からのセンサデータ入力を HTTP 通信を通じて行っていたため，ネットワーク通信機能をもたない機器からのデータ受け取ることができなかった．
- **データの内容を簡便に確認・整理・共有する手段**
  「素材管理」にアップロードしたファイルの中身については，ダウンロードすることで確認することはできるが，それらのファイルを開くためには，そのファイル形式に併せたアプリケーションが別途必要である．処理対象のデータは主に表形式のものが多く，表計算ソフトなどを使って確認することができるが，ダウンロードした表を変更した場合はアップロードをやり直す必要がある．また，複数人で同じ表を更新するような場合に，同時書き換えが起きたときの対処は行っていない．
  なお，KVS 方式の簡易データベースについては現状，中身を読み書きするには，プログラムを書くことが必須になっているため，手軽に確認・整理できるとは言い難い．

これらの問題に「Web ブラウザのみで実行可能な環境」という Bit Arrow の特徴を保ったまま解決を行う手法として，次の方式を採用した．

- **WebUSB を用いた，組み込み機器の直接操作**
  WebUSB[*1]は，Web ブラウザから一定の制限のもと，ローカルコンピューターの USB デバイスに対する接続する機能である．接続されている USB デバイスに対して仮想シリアルポートを作成して，Web ブラウザの JavaScript からシリアル接続を実現することが可能である．これによって，ネットワーク機能を有していない組み込み機器であっても，シリアル接続経由でデータを収集し，Web ブラウザの通信機能でサーバにデータを送信することが可能になる．
- **WebAPI を用いた，データ共有システムとの接続**
  Bit Arrow の Python では，WebAPI の呼出をサポートしており，他の Web システムとの連携が可能である．これを用いて，データベースの登録や共有を行う他のシステムと連携してプログラムを書くことが可能

[*1] https://wicg.github.io/webusb/

ⓒ 2021 Information Processing Society of Japan

になる．

今回は，WebUSB で接続可能な組み込み機器として Raspberry Pi Pico を，WebAPI を用いたデータベースシステムの接続先として Connect DB を採用し，Bit Arrow から使用可能にした．全体の構成を図 **2** に示す．次の 3, 4 節でそれぞれの機能について解説していく．

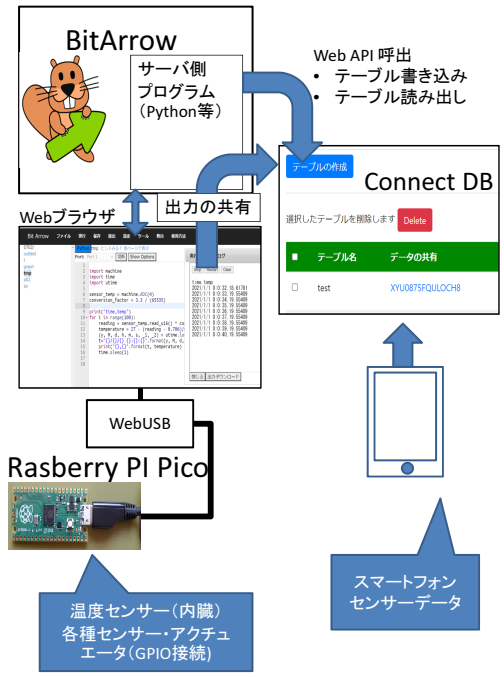

図 **2**　全体構成

3. Bit Arrow の Raspberry Pi Pico 実行モード

Raspberry Pi Pico は，温度センサーと LED を内蔵しており，GPIO ピンに各種センサーやアクチュエータを接続することが可能である．また，MicroPython インタプリタを使っての開発が可能である．

価格が 500 円程度と非常に安価であることが特徴であるが，その代わりネットワーク機能がないため，USB 通じて PC に接続し，PC から USB シリアル接続で Python プログラムを書き込んで実行する方式で開発する．

Web ブラウザの JavaScript プログラムから USB 機器にアクセスする手法として WebUSB が策定されており，現状，Google Chrome や Edge など，Chromium 系列の Web ブラウザであれば対応している[*2]．

Bit Arrow にはすでに Python のプログラミング機能があり [4]，WebUSB を通じて Python プログラムを Raspberry Pi Pico に書き込む機能を追加した．既存の Bit Arrow の Python のプロジェクトに Raspberry Pi Pico のプログラムを書き込むことで実行可能であり，matplotlib など，既存の Python による統計分析機能と同じプロジェクトで実行可能になっている．

実際に Raspberry Pi Pico 上プログラムを実行するには，従来通り Python のプロジェクトを作成し，メニューの「ツール」→「Raspberry Pi Pico 接続」を選ぶ必要がある．すると，Raspberry Pi Pico への接続用の UI が表示されるとともに，メニューの「実行」に「Raspberry Pi Pico で実行」というメニューが追加される．これは，従来通り PC 上での Python のプログラミングを実習している際に，余分な UI やメニューが表示されて混乱を招かないようにするためである．

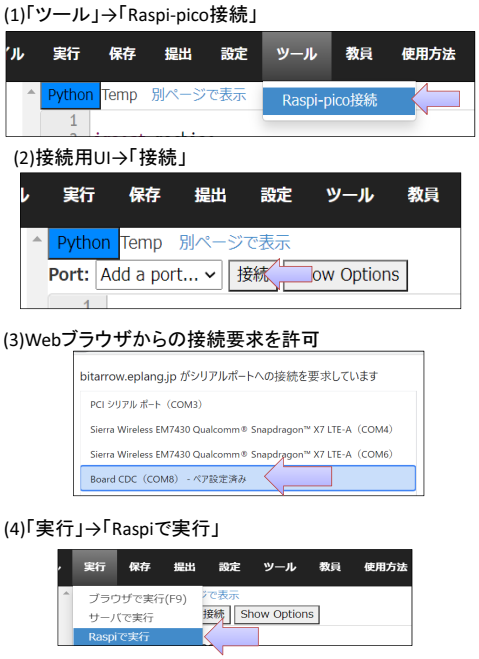

図 **3**　Bit Arrow での Raspberry Pi Pico プログラムの実行手順

4. Connect DB との接続

Connect DB は，著者らが開発している教育用データベース・統計分析システムであり，次のような特徴をもつ [5]．

- PC やスマートフォンの Web ブラウザで動作する
- ログインなしでも，サンプルデータや計測データを利用可能．ログインするとでデータをクラウド上に保存可能
- ログインすることで，クラス単位でのユーザ管理，教員によるデータの配布や共有に対応
- グラフ作成が簡単に可能
- スマートフォンの内蔵センサ（加速度・ジャイロ・GPS）を利用してセンサーデータを作成可能
- 認証不要の Web API(読み取り・書き出し両対応)

[*2] https://caniuse.com/?search=WebUSB

ⓒ 2021 Information Processing Society of Japan

Connect DB 上で作成したテーブルの例を図 **4** に示す．テーブルそれぞれに API キーが割り当てられているのが特徴で，`https://cdb.eplang.jp/api/get?key=`キーなどの URL を用いて，データの読み書きを認証を行うことなく実行することができる．各テーブルはあらかじめスキーマを定義する必要がなく，任意の属性をもつオブジェクトの配列を JSON 形式で送信すれば，必要に応じて属性を増やして格納することができる．

Connect DB の WebAPI は現状，Bit Arrow の Python の「サーバ実行」を通じて，`urllib.request` ライブラリを使って接続することが可能である．

図 **4**　Connect DB のテーブル一覧

また，Bit Arrow 上で実行したプログラムの出力結果を Connect DB に送信する機能を追加した．次のような手順で送信可能である．

- Bit Arrow 上でプログラムを実行する．print などで出力する内容は，カンマやスペースで区切ったデータ形式にする
- プログラムを実行後，図 **5** のように「出力を共有」を選択する．
- 出力結果が表形式に変換されるので，Connect DB の API キーを入力することで，Connect DB に送信される．

なお，図 5 のダイアログは，Bit Arrow でサポートするすべての言語の出力結果（テキスト出力に限る）を送信可能であり，ここで挙げた方法とほぼ同じ手順で，図 1 の「素材管理」へのテキストファイルの送信にも対応している．

5.　実行例

ここでは，現時点で Bit Arrow 上での動作を確認している，Raspberry Pi Pico および Connect DB と連携したプログラムの例を示す．

5.1　Connect DB で収集した加速度センサー値の Bit Arrow でのグラフ化

Connect DB は，スマートフォンの加速度センサー，ジャイロセンサー，GPS の値を収集する機能が備わっている．スマートフォンで Connect DB にアクセスすると，図 **6** のようなセンサデータを収集するページがあり，記録を行うことで，図 **7** のようにテーブルに値が追加されるようになっている．

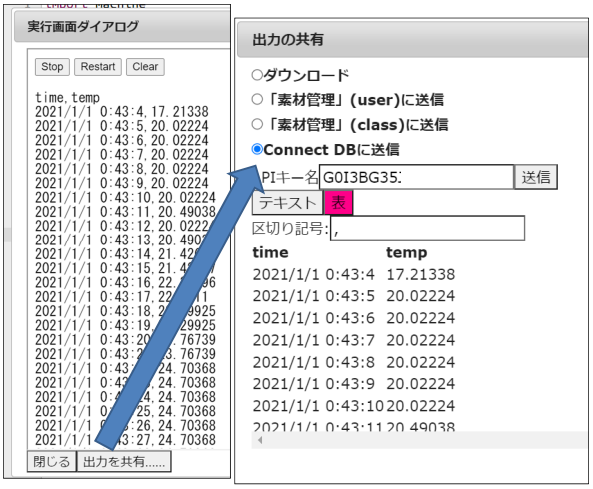

図 **5**　出力の Connect DB への送信

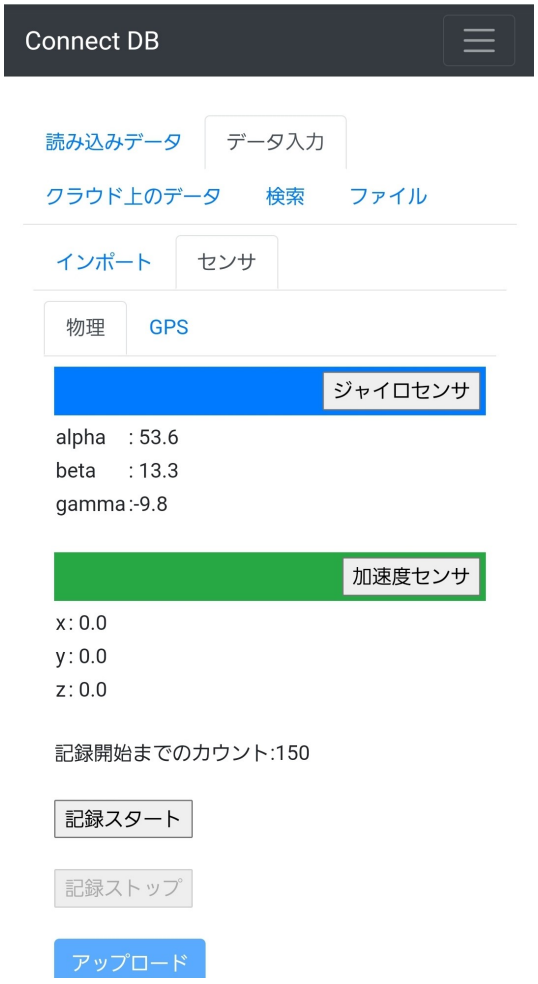

図 **6**　センサーデータ収集ページ

図 **8** は，Connect DB で収集したセンサーデータを Bit Arrow の Python（サーバーで実行）から取得し，グラフとして表示するプログラムである．`key="XXXXXXXXX"`の部分は，図 4 の該当する API キーを指定することで，任意のユーザが任意のテーブルのデータを取得可能である．この

ⓒ 2021 Information Processing Society of Japan

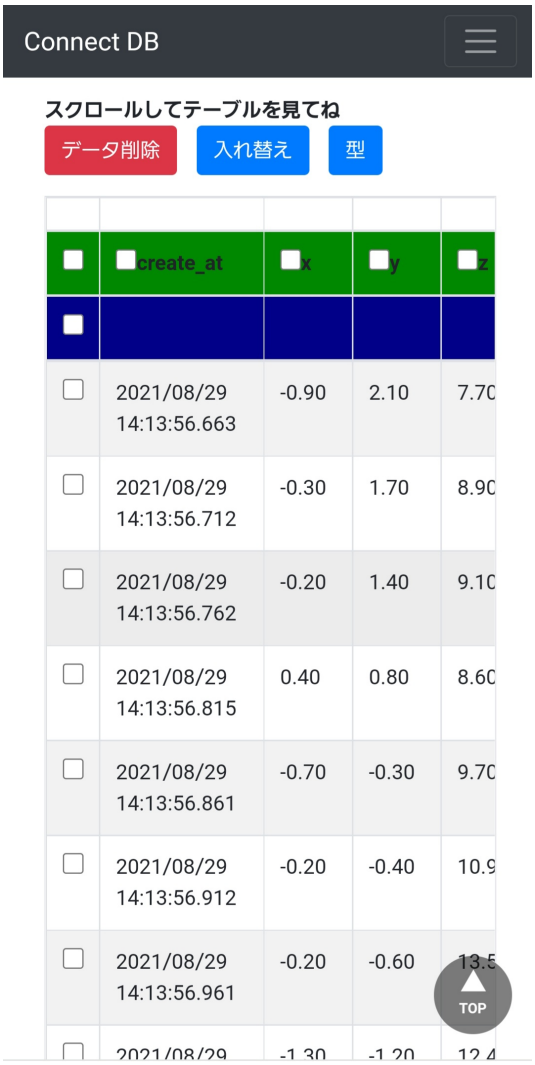

図 7　収集した加速度センサーデータ

プログラムを用いて，図 7 のテーブルの y の値をグラフ化した結果が図 **9** である．

5.2　Raspberry Pi Pico で収集した温度データ Connect DB への送信とグラフ化

図 **10** は，Bit Arrow の Python プロジェクトから Raspberry Pi Pico で実行可能なプログラムである．このプログラムでは，Raspberry Pi Pico の内蔵センサーである温度センサーに接続し (`sensor_temp`)，値を読み出して温度（℃）の値に変換して，タイムスタンプとともに表示している．表示する際には，最初にヘッダ部分として `time,temp` を表示し，その後タイムスタンプと温度をカンマ区切りで 1 秒おきに表示している．この出力結果を「出力を共有」を用いて図 5 のように送信することができる．

さらに，Connect DB において，グラフ化の操作を行うことで，図 **11** のような温度変化のグラフを描くことができる．

あるいは，図 **13** のように Connect DB に保存した温度データを，API を通じて再び Bit Arrow に取り込み，図 **12**

```
from urllib import request
import json
from datetime import datetime
from matplotlib import pyplot

key="XXXXXXXXX"
url='https://cdb.eplang.jp/api/get?key='+key
response=request.urlopen(url)
html = response.read().decode("utf-8")
response.close()
res=json.loads(html)
xs=[]
ys=[]
for e in res:
    ys.append(e["y"])
    fmt='%Y-%m-%dT%H:%M:%S.%f'
    x=datetime.strptime(e["create_at"],fmt)
    xs.append(x)
pyplot.plot(xs,ys)
pyplot.show()
```

図 8　加速度センサー値のグラフ化

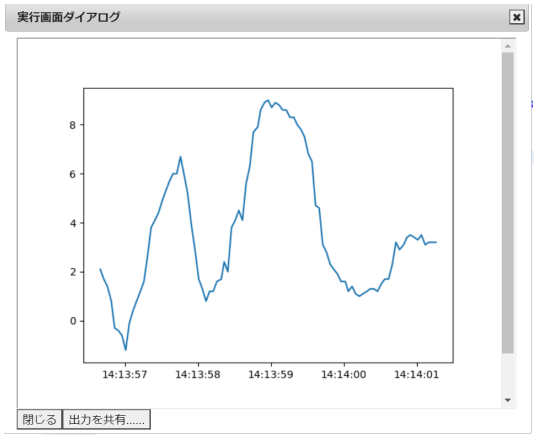

図 9　Connect DB で収集したセンサーデータの Bit Arrow 上での表示

のようにグラフ化することも可能である．このとき，図 10 の温度データ収集プログラム（Raspberry Pi Pico で実行）と，図 13 のプログラム（サーバで実行）は同じプロジェクト内に置くことが可能であり，データの収集から分析までを 1 つの画面で演習することができる．

6.　まとめと今後の予定

本発表では，Bit Arrow に Raspberry Pi Pico を接続して実行する機能と，Connect DB と連携してデータへのアクセスをおこなう機能を拡張した事例を報告し，それによって高校教科で求められている，「データの収集・整理・分析」を Bit Arrow 内で統合的に行える可能性を示した．

ただ，今回 5 節で示した例は，Raspberry Pi Pico と Bit

ⓒ 2021 Information Processing Society of Japan

```
import machine
import time
import utime

sensor_temp = machine.ADC(4)
conv = 3.3 / (65535)

print("time,temp")
for i in range(100):
    reading=sensor_temp.read_u16()*conv
    temperature=27-(reading - 0.706)/0.001721
    (y, M, d, h, m, s, _1, _2)=utime.localtime()
    t="{}/{}/{} {}:{}:{}".format(y,M,d,h,m,s)
    print("{},{}".format(t, temperature) )
    time.sleep(1)
```

図 10　温度データの収集

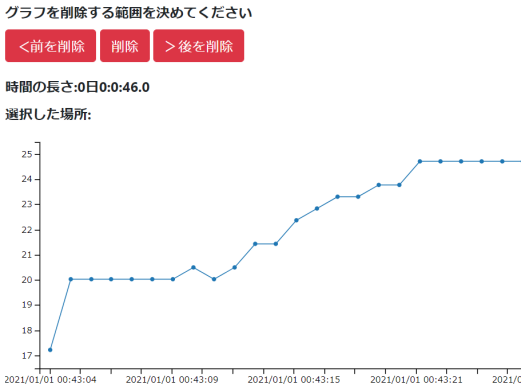

図 11　温度データのグラフ (Connect DB)

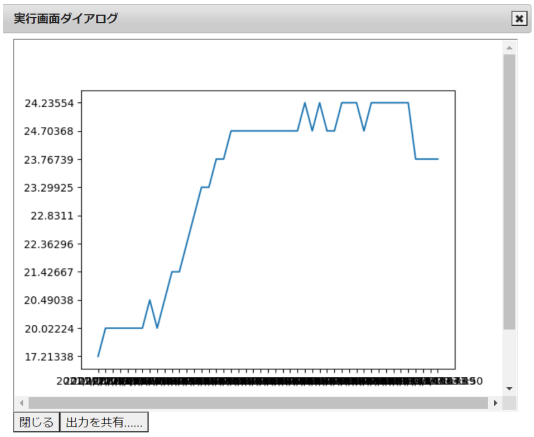

図 12　温度データのグラフ (Bit Arrow)

Arrow と Connect DB が接続できることを示すための単純な例であった．例えば，Bit Arrow でもスマートフォンのセンサーデータを集めることは可能であり [6]，センサーデータをグラフ化する機能については，Connect DB でもサポートされている．また，Raspberry Pi Pico の計測データのグラフ化についても，出力の保存先を Connect DB で

```
from urllib import request
import json
from datetime import datetime
from matplotlib import pyplot
key="XXXXXXX"
url='https://cdb.eplang.jp/api/get?key='+key
response=request.urlopen(url)
html = response.read().decode("utf-8")
response.close()
res=json.loads(html)
xs=[]
ys=[]
for e in res:
    ys.append(e["temp"])
    xs.append(e["time"])
pyplot.plot(xs,ys)
pyplot.show()
```

図 13　温度データのグラフ化プログラム

はなく，「素材管理」へのテキストファイルにし，Python のプログラムから読み込んでグラフ化すれば，Bit Arrow だけで完結することも可能である．

それでも，Bit Arrow と Connect DB を連携させる理由としては，Connect DB は，センサーデータの収集，データの内容の確認・編集，グラフ化などプログラムを構築することなく行うことができるという，Bit Arrow にない利点が挙げられるからである．また，既存の表計算ソフトと比較しても，クラス内でデータを共有する仕組みが標準で組み込まれているなど，教育現場での配慮が行き届いているため，Connect DB 単独でも行える演習も数多くあると考えられる．

一方で Bit Arrow は，プログラムを構築することで，高度なデータ分析を行うために使用することができる．今回の例を発展させることで，例えば次のような題材にも対応可能と考えられる．

- 温度データを複数のデバイスから集めて地図上に表示する
- スマートフォンのセンサーデータから特定の動作（ジェスチャーなど）を機械学習で検出する

今後，このような題材に対しても Bit Arrow と Connect DB を組み合わせて実現可能となるようにライブラリや教材などを充実させていく予定である．

謝辞　本研究は JSPS 科研費 19K03153 の助成を受けたものです。

参考文献

[1] 文部科学省：高等学校学習指導要領（平成 30 年告示）. https://www.mext.go.jp/content/1384661_6_1_

ⓒ 2021 Information Processing Society of Japan

3.pdf.
[2] 文部科学省：高等学校学習指導要領（平成 30 年告示）解説 情報編. https://www.mext.go.jp/content/1407073_11_1_2.pdf.
[3] 長島和平，長　慎也，兼宗　進，並木美太郎：プログラミング学習環境 Bit Arrow でのセンサデータ収集と可視化ライブラリ，技術報告 8，東京農工大学, 明星大学, 大阪電気通信大学, 東京農工大学 (2018).
[4] 長島和平, 長　慎也, 広樹間辺, 兼宗　進, 並木美太郎：オンラインプログラミング環境 Bit Arrow における Python 処理系, 情報教育シンポジウム論文集, Vol. 2019, pp. 122–129 (2019).
[5] 岸本有生，本多佑希，漆原宏丞，兼宗　進：スマートフォンの内蔵センサを用いたデータ分析教材の提案，情報教育シンポジウム論文集，Vol. 2021, pp. 159–163 (2021).
[6] 本多佑希，大村基将，長　慎也，久野　靖，並木美太郎，兼宗　進：Dolittle のオンラインプログラミング環境の開発，技術報告 25，大阪電気通信大学, 大阪電気通信大学／静岡大学, 明星大学, 筑波大学, 東京農工大学, 大阪電気通信大学 (2016).

ⓒ 2021 Information Processing Society of Japan