Vol.2018-CE-143 No.8
2018/2/17

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

長島 和平[1,a)]　長 慎也[2]　兼宗 進[3]　並木 美太郎[1]

概要：著者らは Web ブラウザで学習が可能なプログラミング環境 Bit Arrow を開発している．これまで Bit Arrow では，プログラミングによる作品制作を助けるライブラリの提供などを行ってきた．近年 IoT やビッグデータという用語が広まってきている背景もあり，センサデータをサーバに蓄積して，グラフを描画する実習を行った報告もある．本研究では，Bit Arrow でデータの収集やグラフ化などの統計処理をサポートするため，ラズベリーパイなどの機器から取得できるセンサデータを蓄積できる仕組みと，データをグラフ化するライブラリを実装した．センサデータは HTTP 通信を用いて Bit Arrow へ送信・収集することができる．データのグラフ化には公開されているグラフライブラリを元に，初学者でも扱いやすいような Bit Arrow 独自の API を用意して提供する．

Sensor data collection and visualization on Bit Arrow

Nagashima Kazuhei[1,a)]　Cho Shinya[2]　Kanemune Susumu[3]　Namiki Mitaro[1]

1. はじめに

次期学習指導要領では，高等学校の共通教科情報でプログラミングが必履修となることが決まっており，著者らは Web ブラウザで学習が可能なプログラミング環境 Bit Arrow を開発している．これまで Bit Arrow では，プログラミング初学者の学習を助けるためにエラー発生位置の通知や作品制作を簡単に行えるためのライブラリの提供といったサポートを行ってきた．Bit Arrow の JavaScript を用いた授業では，従来のメモ帳とブラウザを用いて行う JavaScript の授業と比較して，実行時にエラーが発生する割合が減少し，エラーが発生した場合においてもそれを修正する時間が短くなったことが確認できた [1]．また，教員側への支援として，履修している学習者を一括登録する機能や，学習者が実行したプログラムのソースコードと実行結果を時系列で確認できる機能，サンプルプログラムや課題で実装させるファイルを配布する機能などを提供しており，実際に大学の授業で利用したところクラス内で頻発しているエラーを教員側で把握することに役立ったことが分かっている [2]．これまでの実践では，Bit Arrow で提供しているグラフィックス命令を用いたゲーム制作を通じたプログラミング教育を行ってきた．

近年ではプログラミング以外にも IoT やビッグデータといった用語が広まるなど，ネットワークの仕組みに関する学習が必要になってくると考えられる．次期学習指導要領のイメージにも，必履修科目では「情報通信ネットワークとデータの利用」が項目として挙げられており，発展的な選択科目では「情報とデータサイエンス」という項目が含まれている [3]．ネットワークの仕組みを学習するためには，教室の温度データを収集したり，チャットのようなシステムを作成したりと，学習者が身近に感じるものを題材として扱うことが多くなると考えられる．また，データを分析するために統計関数が使われたり，グラフによる可視化を行う演習が取り組まれることも予想される．現行の教科書でも扱われているプログラミング言語ドリトルでは，

[1] 東京農工大学
Tokyo University of Agriculture and Technology
[2] 明星大学
Meisei University
[3] 大阪電気通信大学
Osaka Electro-Communication University
[a)] kazuakanapo@namikilab.tuat.ac.jp

2018 年に入ってデータベース，統計，グラフなどを扱うライブラリを含めた統計機能が公開されるなど，次期学習指導要領の施行に向けた動きが始まっている [4].

本研究では，プログラミング学習支援環境 Bit Arrow のサーバにセンサデータを収集する機能と，Bit Arrow の JavaScript でそのデータを分析するためのグラフライブラリ，統計ライブラリを実装した．この実装により，ラズベリーパイや Arduino といった機器から送信されたセンサデータを Bit Arrow 上に蓄積し，ライブラリを用いて分析できるようになる．

2. センサやデータ分析を題材とした教育実践

IoT やデータ分析といった内容を扱った授業の報告はまだ少ない．辰己らの調査では高校時代に統計処理について身に付いた，または身に付けたとした大学 1 年生は 15 %であるとされている [5]．細合らの報告は IoT を活用できる人材育成のための教材開発に関する内容であるが，授業の対象となったのは大学院生で，組込みハードウェアや組込みソフトウェアに関するスキルや Web アプリケーション開発，ロボット制御ソフト開発の技術の習得が到達目標としてあげられており，すでにある程度の知識をもった学習者向けのものである [6]．また，大学の経営学部向けの講義で，HTML と PHP でアンケートを作成し結果をファイルに書き込み，データを表計算ソフトに取り込んで解析するという実践報告 [7] があったが，データの分析とプログラミングで異なる環境を使わなければならない．小学校高学年向けの科学教室で計測デバイスにデータを保存し，それを PC と接続して表計算ソフトでグラフを作成させる実践の報告 [8] もあったが，プログラミングは扱われていないほか，データも計測デバイスに保存するためネットワークを通じてサーバに送信するといったことはされていない．

高校生がセンサを用いてプログラミングを行った演習としては，いくつかのセンサの値を使ってロボットを制御する体験授業 [9] や，加速度センサを用いて体感的な操作を行うプログラム作品を作る実践 [10] もあるが，どちらもセンサデータを蓄積・分析させるような内容ではない．

様々な地点のデータを収集できるプロジェクトとして Live E!というものもあり，デジタル百葉箱で収集された気温，湿度，気圧，雨量，風向，風速，二酸化炭素濃度が Live E!のサーバにアップロードされている [11]．このデータを用いて，情報理数科の生徒が PHP でデータを可視化するサイトを作成した実践の報告もある [12]．この演習は情報理数科の生徒が対象で，1 年次に Processing を学習し，3 年次には自宅で Linux サーバを立てて PHP や C のプログラムを書いている生徒もいるなど普通高校よりも情報科学に対する知識がある学習者に向けたものである．また，Live E!のデータを利用するためには Live E!プロジェクトの会員への登録が必要である．会員に提供されるのはデータのアップロードおよびダウンロードするためのライブラリで，データ分析やグラフ化といった作業は利用者自身で行う必要がある．共通教科情報に関して，IoT を活かした授業実践の例も報告されている [13]．この実践ではラズベリーパイにセンサを取り付け，収集したセンサデータをデータ蓄積サーバに蓄積している．収集したデータは表計算ソフトでグラフ化したとする記述がある．

IoT を活用した学習ではデータの収集・蓄積とデータの分析を行うことになる．このように IoT 機器から収集したデータを保存するためにはそのデータを蓄積させるためのサーバが必要となる．また，こうして蓄積されたデータを分析するためには，データをダウンロードし表計算ソフトなどを用いて分析やグラフ化を行うことが多い．このような授業において IoT 機器から収集したセンサデータをプログラミング学習環境に蓄積することで，その後の分析にプログラミングを使うことができると考えた．また，収集したデータをそのままプログラミング学習で用いた環境で分析できることで複数の環境を使う必要がなくなるため，環境ごとに使い方を学習する手間もなくすことができる．さらに，プログラミング学習環境で IoT に触れることで，プログラミングだけでなくネットワークの仕組みやデータの分析に関する理解も深めることができると考えられる．

3. ネットワーク学習とデータ分析を支援する機能の設計

3.1 ネットワーク学習とデータ分析支援の課題と目標

センサデータの収集や分析を通じてネットワークの学習を行うにあたり，まず IoT 機器などから送信されたセンサデータをサーバに蓄積する必要がある．これまでの環境では，データを蓄積させるためのサーバを教員側で用意し，ラズベリーパイなどからそのサーバにセンサデータを送信する必要があった．そこで，Bit Arrow のサーバに直接センサデータを送信できる仕組みを作ることで，教員側でデータ蓄積用サーバを用意することなく演習を行え，Bit Arrow でプログラムを用いてデータを分析できるようになる．

データを分析するための演習では，平均や相関係数などを求めるために統計関数を使うことが考えられる．平均や相関係数などは，求め方を分かっていれば学習者が自力でプログラムを組むことも可能である．しかし，このような値を計算するためのプログラムを作成することは初学者にとっては容易ではなく，プログラムの間違いが原因でデータの分析に手が回らなくなってしまう可能性もある．そのため，Bit Arrow でいくつかの統計関数をサポートすることで，データの分析に必要なプログラムを短くし，分析を円滑に進められるようになる．

また，例えば温度データを収集しその移り変わりを視覚的に見るためには蓄積したデータをグラフ化すると分かり

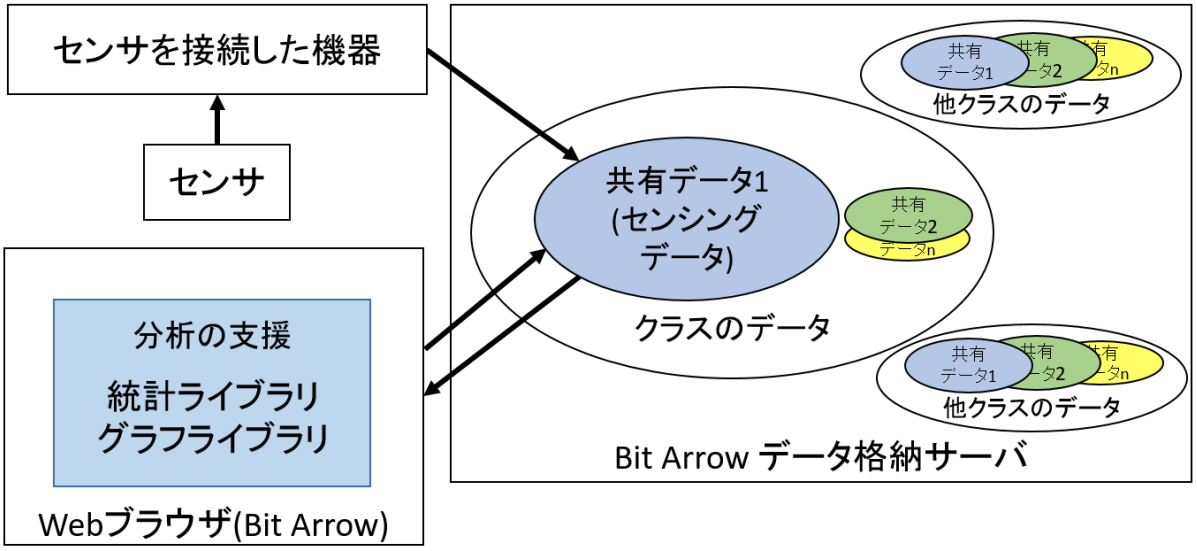

図 1　Bit Arrow を用いたデータ収集と分析の流れ

やすい．JavaScript では Canvas を使うことで自由に線や図形を記述することができるほか，Bit Arrow では図形描画のライブラリを提供しているが，これらの機能を学習者に使わせてグラフを描画させることは難しい．グラフを描画するための公開されたライブラリもあるが，JavaScript を使い慣れていないと使いにくい構文や，描画に関わる細かい設定が多いことからライブラリをそのまま使わせることは難しい．Bit Arrow でグラフを描画するためのライブラリを用意することで，描画をより簡単に行わせることができる．

また，保存するデータについてはクラスごとに管理を行う．クラスごとに仮想共有空間を作ることで同じデータにアクセスしたり，他のクラスからのアクセスを制限することができる．

センサが接続された機器から Bit Arrow のサーバへのデータ送信，および Bit Arrow を用いた分析の流れを図 1 に示す．

3.2　ネットワークを通じたデータの収集と取得

ラズベリーパイなどのセンサデータを取得できる機器から Bit Arrow のサーバへデータを送信するために，HTTP 通信を用いる．サーバにデータを格納するための API を用意し，センサを接続している機器のプログラムからネットワークを経由してデータを登録する．このデータの送信は HTTP 通信で行うため，HTTP 通信機能を持った機器であれば Bit Arrow にデータを送信することができる．また，Bit Arrow からでも同様にデータを送信できるようにすることで，チャットのプログラムなどに活用することができる．収集するデータはクラスごとに分けて管理ができる．また，クラスごとに管理するだけでなくグループを指定できるようにすることで，クラス内でグループを分けた演習などでそれぞれが異なるデータを扱うことができる．

3.3　統計関数

センサなどから収集したデータを Bit Arrow で分析するときには，なるべくデータを加工させる手間をかけさせたくない．したがって，Bit Arrow で用意する統計関数は，3.2 の方法で集めたデータをそのまま分析できるようなライブラリとする．プロトタイプで搭載する統計関数は，相関係数を計算するのに使用されるものとして，平均，分散，共分散，偏差，標準偏差，相関係数を選んだ．その他には合計，最大値，最小値も実装する．

3.4　グラフ化

3.3 と同様に，センサなどから収集したデータをなるべく加工させる手間をかけさせずにグラフ化させたい．したがって，グラフ化のライブラリも，3.2 の方法で集めたデータをそのまま分析できるようにする．プロトタイプで対応するグラフは，棒グラフ，折れ線グラフ，散布図を選択した．また，グラフの軸のタイトルの設定や，描画する範囲の設定，さらに散布図の描画時には，回帰直線の描画なども必要であると考える．

以上の機能を提供することで，センサデータをネットワークを介してサーバへ送信し，そのデータを Bit Arrow で分析する演習を行うことができる．計測したセンサデータが Web サーバに送信・蓄積され，そのデータをコンピュータで取得できる演習を通じて，ネットワーク通信の仕組みについて学習することができるようになる．また，そのデータを分析するための統計関数やグラフ化のライブラリを用いた演習は，統計処理の手法やデータごとに使われるべきグラフの種類などの学習になる．

4. 各機能の実装

3 で述べた設計を元に，センサデータを Bit Arrow に送信するための API，Bit Arrow で蓄積されたデータを取得する API，統計関数とグラフ化の関数を実装した．Bit Arrow ではこれまで初学者向けに簡単に記述ができるような命令を用意することを考えて実装を行ってきた．特に，オブジェクト指向についてはプログラミング教育でも終盤に学習する項目であることから，初学者がオブジェクト指向についての知識を習得する必要なく記述ができるようにライブラリを実装していた．そのため，本研究で実装するライブラリについても，オブジェクト指向の概念を考えずに記述できるような実装を検討した．しかし，グラフ化については，軸のテキストや描画範囲など個別に設定する項目があることや，複数のグラフを描画したいときにそれぞれの異なる設定で記述することも考えられる．これらの設定項目とデータ，グラフ化したいフィールドの情報などを一度に引数に与えると 1 つの命令がとても長くなってしまうことから，グラフについてはグラフオブジェクトを用意することとした。各機能の実装と使い方についてを次に説明する．

4.1 データの収集と取得の実装

ラズベリーパイなどの機器で取得したセンサデータを Bit Arrow へ送信するために，HTTP 通信の GET を用いることとした．Bit Arrow で用意した API の URL の後ろに，クエリ文字列を連結することでデータを Bit Arrow のサーバへ送信することができる．また，Bit Arrow ではセンサの値やチャットのデータなどはクラスごとに収集され，データの保存方式には上書き型の KVS(Key Value Store) と追記型のログ形式の 2 種類である．データを送信するときにはキーと値をセットで書き込み，そのデータを Bit Arrow のプログラムで取得するときにはキーで検索して値を受け取る．データが送信されたとき，Bit Arrow のサーバへ登録されるデータにはキーと測定値以外に，送信された時間の情報が自動的に付与される．また，送信する値も複数設定することができる．

Bit Arrow のサーバに送信されたデータは送信時に設定したキーを元に取得する．また，設定していればキー以外にもグループを指定することもできる．データを Bit Arrow のプログラムから取得する時には一つのデータを一つのオブジェクトとして，取得したすべてのオブジェクトを配列に格納する．また，送信時に自動的に付与される送信された時間は取得したオブジェクトに time というキーで保持される．

Bit Arrow に用意したデータの送信と取得に関わる命令を**表 1** に示す．Bit Arrow からデータを送る場合には addLog という命令を用意した．センサデータを送信していたのと同様に addLog にキーと値を引数で与えることで Bit Arrow のサーバにデータを送信することができる．引数に複数の値を入れることで，複数の値を送信できる．

センサデータとサーバで収集したデータは findLog を用いて取得できる．HTTP 通信の GET で集めたデータや addLog で集めたデータのキーを指定することでこれまでに蓄積されたデータを取得することができる．ラズベリーパイから取得した温度データを画面に表示するプログラムの実行結果を**図 2** に示す．

また，このように収集したデータを表示できることから，Bit Arrow で送信と表示のプログラムを記述することでチャットプログラムも作成することができる．Bit Arrow はスマートフォンでの実行にも対応しているため，チャットプログラムをスマートフォンで実行してみることで身近に利用しているシステムと関連付けてネットワークの仕組みを考えさせることができると考えられる．チャットプログラムを実行した画面を**図 3** に示す．

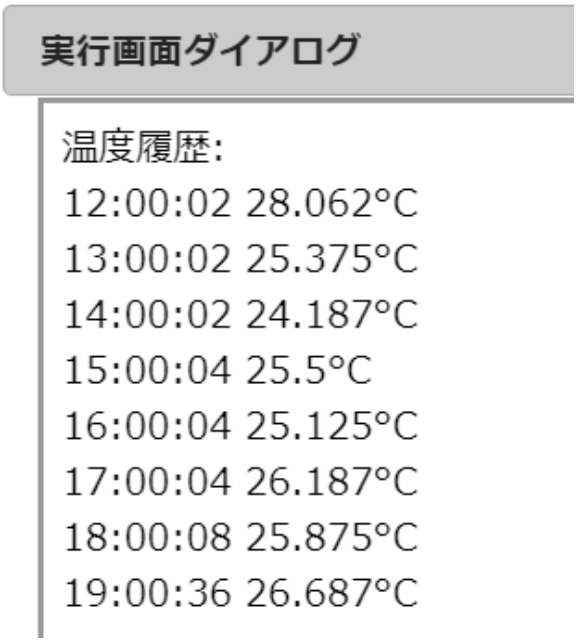

図 2 ラズベリーパイから収集した温度データ

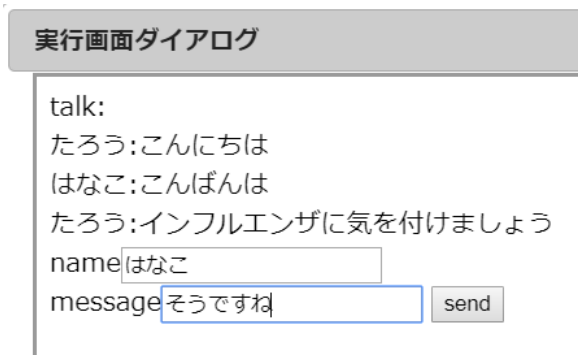

図 3 チャットプログラムの実行画面

表 1　データの送信と取得に関するライブラリ

命令	動作
setGroup(グループ名)	下の命令で送信したり取得する対象のグループを設定する．省略すると”default” になる．
addLog(キー, 値 1 [, 値 2[...[, 値 N]]])	キーを指定して値を送信することができる．値は 1 から順に”data1”,”data2”,...,”dataN” として送信される．サーバへは追記型で保存される．
findLog(キー)	addLog で蓄積されたデータから指定したキーと合致する追記型で保存されたデータを取得する．
putToServer(キー, 値)	指定されたキーと値を送信する．サーバへは KVS（上書き型）で保存される．
getFromServer(キー)	指定されたキーと合致する KVS（上書き型）で保存されたデータを取得する．

```
var cc=corrcoef(src,"ans","erate");
addText("calc","相関係数:"+cc);
```

図 4　相関係数を表示する JavaScript コード

図 5　相関係数を表示するプログラムの実行画面

4.2　統計関数の実装

実装する統計関数は，相関係数を計算するのに使用されるものから選んだ．**表 2** に一覧を示すとおり，平均，分散，共分散，偏差，標準偏差，相関係数を実装し，その他に合計，最小値，最大値も実装している．収集されたデータは Bit Arrow では，4.1 で述べた通りオブジェクトの配列として取得できる．取得したデータをそのまま分析できるように，オブジェクトの配列をそのまま引数に与えることができる仕様とした．また，オブジェクトの中から分析したいフィールド名を選択する必要があることから，オブジェクトの配列と，分析したいフィールド名を引数に取る．

どの命令も共通して第一引数にはデータオブジェクトの配列を渡す．第二引数にはそれぞれの値を計算したいフィールドを与える．共分散と相関係数を求める cov と corrcoef のみ，第三引数にもう一つのフィールドを与える．結果として計算した数値，一部の命令では数値の配列を受け取ることができる．また，統計関数ライブラリの命令名は，NumPy で用いられている命令名を参考にした．例として，プログラミングの授業における課題正解数と演習中にエラーが発生した割合のデータから，課題正解数と演習中にエラーが発生した割合の相関係数を表示するプログラムを**図 4** に示し，実行結果を**図 5** に示す．

4.3　グラフ化の実装

グラフ化を行うにあたって，公開されているグラフライブラリを導入することを考えた．公開されているライブラリは高機能なものが多く，グラフ描画を Bit Arrow に実装するよりも高性能で見やすいものを提供できる可能性が高いためである．そこで，著者らは MIT ライセンスで公開さ

```
setGroup("mygrp");
src=findLog("thermo");
var x=[];
var y=[];
for(var i=0;i<src.length;i++){
    x.push(src[i]["time"]*1000);
    y.push(src[i]["data1"]);
}
var data=[{
    x:x,
    y:y,
    type:"line",
    mode:"lines"
}];
var options={
    xaxis:{
        type:'date'
    }
};
window.Plotly.newPlot("graph",data,options);
```

図 6　収集した温度データを Plotly.js のライブラリを用いてグラフ化するコード例

れている Plotly.js というライブラリを導入した [14]．しかし，Plotly.js のグラフ描画機能をそのまま提供しようとすると，findLog で取得したデータを x 軸の配列と y 軸の配列に加工する必要があり，他にも設定項目が多くあるためプログラムが長くなってしまった．例として図 2 で示した温度データを折れ線グラフで表示するために Plotly.js を用いたプログラムを**図 6** に示す．なお，Plotly.js ではグラフの描画領域として HTML の div 要素を使用する．図 6 の一番最後の行で第一引数として与えている”graph” は HTML にあらかじめ記述した描画領域の div 要素の id である．

グラフ化のために手間がかかりすぎるのは良くないと考え，Plotly.js のライブラリをそのまま提供するのではなく，より短いプログラムで記述できるグラフオブジェクトを Bit Arrow で用意することとした．用意した命令は**表 3** に示す．取得できるセンサデータをそのままグラフ化するために，統計関数と同様にグラフ化の関数でもオブジェクトの配列を引数で与える．その他，x 軸と y 軸に適用するフィールドを設定するためにそれぞれのフィールド名を与えることとした．Plotly.js では描画領域の div 要素の id が必要であるため，グラフオブジェクトのコンストラクタで与えることとした．また，グラフ描画は同じ命令とし，折れ線グラフや棒グラフといったグラフのタイプを引数で与

表 2　統計関数ライブラリ

命令	動作
mean(データオブジェクト配列, フィールド名)	データオブジェクト配列から指定されたフィールドの平均値を返す.
vari(データオブジェクト配列, フィールド名)	データオブジェクト配列から指定されたフィールドの分散を返す.
cov(データオブジェクト配列, フィールド名 1, フィールド名 2)	データオブジェクト配列から指定された 2 つのフィールドの共分散を返す.
dev(データオブジェクト配列, フィールド名)	データオブジェクト配列から指定されたフィールドの偏差を配列で返す.
std(データオブジェクト配列, フィールド名)	データオブジェクト配列から指定されたフィールドの標準偏差を返す.
corrcoef(データオブジェクト配列, フィールド名 1, フィールド名 2)	データオブジェクト配列から指定された 2 つのフィールドの相関係数を返す.
sum(データオブジェクト配列, フィールド名)	データオブジェクト配列から指定されたフィールドの合計値を返す.
min(データオブジェクト配列, フィールド名)	データオブジェクト配列から指定されたフィールドの最小値を返す.
max(データオブジェクト配列, フィールド名)	データオブジェクト配列から指定されたフィールドの最大値を返す.

```
g=new GraphObject("graph");
setGroup("mygrp");
res=findLog("thermo");
g.setXAxisText("時間");
g.setYAxisText("気温");
g.line(res,"time","data1");
```

図 7　収集した温度データを Bit Arrow で用意したライブラリを用いてグラフ化するコード例

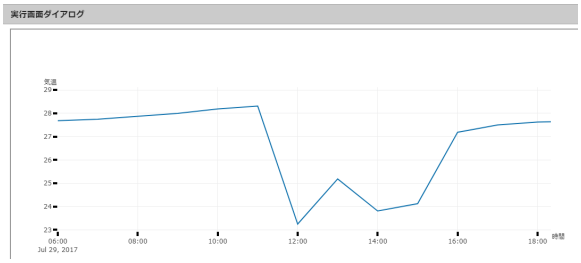

図 8　温度データのグラフ化の実行画面

える仕様 (例:draw(グラフタイプ, データオブジェクトの配列, フィールド...)) も考えたが，引数が多いと初学者に易しくないと判断し命令を細かく分けることとした．

図 6 で示した温度データをグラフ化するプログラムを表 3 のライブラリを用いて記述したときのプログラムを**図 7** に示す．図 7 のプログラムでは，setXAxisText と setYAxisText を用いて各軸のラベルを設定しているが，Plotly.js のライブラリをそのまま用いる図 6 で同様にラベルを設定しようとするとこのプログラムに約 20 行のプログラムをさらに書き足す必要がある．実行結果を**図 8** に示す．

また，散布図を記述する場合のみ，第四引数に true を渡すことで回帰直線を自動で描画する機能を付けた．これも Plotly.js では一つのグラフ上に複数のグラフを描画するための手続きを書かなければならないほか，回帰直線の式を計算し，配列の形式に格納して渡す手間がかかるため，学習者からはフラグのみで描画できるような実装とした．図 5 の表示に用いたデータから，課題正解数と演習中にエラーが発生した割合の散布図と回帰直線を描画するプログラムを**図 9** に示し，実行結果を**図 10** に示す．

```
g=new GraphObject("s");
g.scatter(src,"ans","erate",true);
```

図 9　散布図と回帰直線を描画するコード例

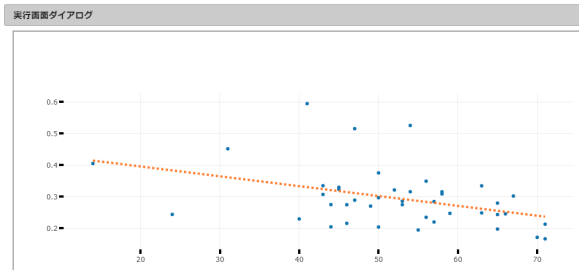

図 10　散布図と回帰直線のグラフ化の実行画面

このライブラリを用いることでグラフ化のために記述する必要があったプログラムを減らすことができた．統計関数やグラフ化を行うライブラリを用いることでプログラムを簡単に書くことができるようになり，授業での利用時にはデータの分析や考察に時間を使うことができるようになることが期待できる．本稿ではプロトタイプとして実装したライブラリを報告したが，統計関数やグラフ化には他にも多数の種類があるため，ここで実装したものの他に必要なものを選択し，追加で実装を行っていきたい．

5. 評価

実装した統計ライブラリとグラフライブラリは実際の授業や演習で利用するには至っていない。そこで，高等学校の情報と科学で現在使用されている教科書で扱われているグラフや統計量について，今回実装を行ったライブラリがそのうちのどれだけに対応しているかを調べる．扱われているものを網羅していれば，このライブラリを用いて授業を行うことができるものであると考えられる．また，今回ライブラリを JavaScript で実装しているため，調査する対象とした教科書はプログラミングの項目で JavaScript を扱っている日本文教出版 [15]，東京書籍 [16] のものとした．

まず日本文教出版の教科書では，データを分析するための代表的な統計量として平均，中央値，度数，最頻値，分

表 3 グラフオブジェクトのライブラリ

命令	動作
GraphObject(描画領域の id)	グラフオブジェクトのコンストラクタ．Plotly.js ではグラフ描画のために HTML の div 要素の id を求めるため引数で与える．
line(データオブジェクト配列,x 軸のフィールド,y 軸のフィールド)	データオブジェクト配列から指定されたフィールドの折れ線グラフを描画する．
bar(データオブジェクト配列,x 軸のフィールド,y 軸のフィールド)	データオブジェクト配列から指定されたフィールドの棒グラフを描画する．
scatter(データオブジェクト配列,x 軸のフィールド, y 軸のフィールド [, 回帰直線描画])	データオブジェクト配列から指定されたフィールドの散布図を描画する． 第四引数に true を与えると，自動的に回帰直線を描画する．
setXAxisText(x 軸のラベル)	グラフの x 軸のラベルを設定する．
setYAxisText(y 軸のラベル)	グラフの y 軸のラベルを設定する．
setXRange(最小値, 最大値)	グラフを描画する x 軸の範囲を設定する．
setYRange(最小値, 最大値)	グラフを描画する y 軸の範囲を設定する．

散，標準偏差，相関係数が紹介されている．また，これらの分析について表計算ソフトを使って求められるといった記述も見られる．実習では平均を求める問いと，分散を求めて何が分かるかを考えさせる問いが出されている．グラフは度数分布表 (ヒストグラム)，折れ線グラフ，円グラフ，棒グラフ，帯グラフ，散布図が例題や説明の中で扱われている．実習では折れ線グラフを書かせる問いが出されている．

東京書籍の教科書は，実習編，理論編，総合実習編に分かれており，統計については総合実習編で相関係数が紹介されている．グラフについては折れ線グラフと円グラフが実習編に，折れ線グラフと棒グラフが理論編に，折れ線グラフと散布図が総合実習編に用いられている．また，これらのグラフの他に資料として帯グラフ，レーダーチャート，箱ひげ図について簡単に紹介がされている．

2 つの教科書で扱われていた統計関数とグラフを本研究で実装した機能と比較した表を**表 4** に示す．各教科書の実習で実際に使用される項目については表中に「◎」で示す．Bit Arrow のライブラリで現時点で未実装であるものは「未」と表す．

表 4 から，統計関数については日本文教出版の教科書で扱われている中央値，度数，最頻値が Bit Arrow のライブラリには不足していることが分かった．東京書籍の教科書では，相関係数が扱われているだけであったため，今回実装したライブラリで対応が可能である．また，日本文教出版の実習の中で用いられていたのは平均と分散であり，この 2 点についてはすでにライブラリで提供されているものである．このことから，実習で扱われる内容については Bit Arrow で実装したライブラリで対応可能であるといえる．

グラフライブラリについては，ヒストグラム，円グラフ，帯グラフ，レーダーチャート，箱ひげ図が不足していることが分かった．一方，実習で用いられるグラフの種類については，日本文教出版は折れ線グラフ，東京書籍は折れ線グラフと散布図であることから，どちらの教科書についても実習で扱われる内容については今回実装したライブラリでカバーすることができている．

表 4 教科書の記載と実装したライブラリの対応

		日本文教出版	東京書籍	Bit Arrow
統計関数	平均	◎	×	○
	中央値	○	×	未
	度数	○	×	未
	最頻値	○	×	未
	分散	◎	×	○
	標準偏差	○	×	○
	相関係数	○	○	○
グラフ	ヒストグラム	○	×	未
	折れ線グラフ	◎	◎	○
	円グラフ	○	○	未
	棒グラフ	○	○	○
	帯グラフ	○	○	未
	散布図	○	◎	○
	レーダーチャート	×	○	未
	箱ひげ図	×	○	未

一方で，例題や紹介されているグラフについても，実際に生徒に描かせるような演習も行われることが予想される．今後は表 4 をもとに不足している統計関数やグラフの種類への対応を行うことで授業で実際に利用できるものにしたい．

6. まとめ

本研究では，プログラミング学習支援環境 Bit Arrow のサーバに，IoT 機器から収集できるセンサデータを送信できる機能と，そのデータを Bit Arrow で取得し，統計関数やグラフ化のライブラリを用いることで分析できる環境を実装した．データはセンサからだけでなく Bit Arrow でも収集することができ，これを応用してチャットに似たプログラムを授業で作らせることもできる．データはクラスごとにグループとキーで管理され，グループごとに異なる値をやり取りすることもできる．

収集したデータを分析するために，統計関数とグラフ化のライブラリを実装した．統計関数は，相関係数を算出するのに必要となるものを中心に実装を行い，収集したデー

タを特に加工しなおす必要なく分析を行うことができる仕様とした．また，グラフ化のライブラリには MIT ライセンスで公開されている Plotly.js を導入し，グラフ描画のために記述するプログラムが短くなるよう Bit Arrow でライブラリを用意した．

ネットワークを通じてセンサデータを収集する実習を行うことで，ネットワークの仕組みについての学習につながることが期待できる．また，収集したデータを分析することで，統計教育にも生かすことができると考えられる．

現在扱われている教科書中に登場する統計関数とグラフの種類を本研究で実装したライブラリと比較した結果，教科書中で実習として扱われている項目について対応できていることが分かった．しかし，教科書中で例題や説明として紹介されているがライブラリへの対応が行われていない機能もあったため，今後はそれらの機能について実装を行い，実際にこの機能を用いた授業実践などでネットワークの仕組みや統計的な分析の学習に効果があるか検証したい．

参考文献

[1] 長島 和平, 長 慎也, 間辺 広樹, 兼宗 進, 並木美太郎：Web ブラウザを用いたプログラミング学習支援環境 Bit Arrow の設計と評価，情報処理学会研究報告コンピュータと教育（CE），Vol. 2017-CE-138, No. 2, pp. 1–8 (2017).
[2] 長島 和平，堀越 将之，長 慎也，間辺 広樹，兼宗 進，並木美太郎：プログラミング学習支援環境 Bit Arrow の教員支援機能の設計と試作，情報教育シンポジウム 2017 論文集，Vol. 2017, pp. 129–136 (2017).
[3] 幼稚園、小学校、中学校、高等学校及び特別支援学校の学習指導要領等の改善及び必要な方策等について（答申）（中教審第 197 号），http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/1380731.htm.
[4] プログラミング言語「ドリトル」, http://dolittle.eplang.jp.
[5] 辰己 丈夫，江木 啓訓，瀬川大勝：大学 1 年生の情報活用能力と ICT 機器やメディアの利用状況調査，学術情報処理研究，No. 16, pp. 111–121 (2012).
[6] 細合 晋太郎，石田 繁巳，亀井 靖高，鵜林 尚靖，福田晃：自律走行ロボットを用いた IoT 開発 PBL に向けた教材開発，組込みシステムシンポジウム 2015 論文集，Vol. 2015, pp. 40–45 (2015).
[7] 福永 厚：情報教育のための Web プログラミングによるアンケート作成とデータ解析について，北海学園大学学園論集，No. 3, pp. 17–25 (2016).
[8] 福地 健太郎，茂木大佑：センサによる計測を題材とした小学校高学年向け教材の開発とその活用事例，情報処理学会研究報告ヒューマンコンピュータインタラクション（HCI），Vol. 2011-HCI-145, No. 3, pp. 1–8 (2011).
[9] 高橋 知希, 富永浩之：高校生への導入体験としての LEGO プログラミング演習の支援－高大連携の LEGO 講座における教育実践－，技術報告 18 (2013).
[10] 大見 嘉弘，滑川 敬章，永井保夫：情報系高校におけるセンサを利用したプログラミング教育の実践，情報処理学会研究報告コンピュータと教育（CE），Vol. 2012-CE-114, No. 5, pp. 1–7 (2012).
[11] 落合 秀也，松浦 知史，山内正人：センサネットワーキングの新たな展開を目指して-Live E! Workshop in APNG Camp 活動報告-，情報処理， Vol. 50, No. 1, pp. 55–63 (2009).
[12] 滑川 敬章, 落合 秀也, 山内 正人, 高岡 詠子, 中山 雅哉, 江崎浩，砂原秀樹：情報系高校における環境情報を計測・可視化する実用的なプログラミング教育の実践，情報処理学会研究報告コンピュータと教育（CE），Vol. 2012-CE-116, No. 16, pp. 1–8 (2012).
[13] 間辺 広樹，大村 基将，林 康平，兼宗進：情報科教育における IoT 学習環境の利用方法の検討，情報教育シンポジウム 2016 論文集，Vol. 2016, pp. 98–105 (2016).
[14] Plotly.js, https://github.com/plotly/plotly.js.
[15] 水越 敏行，村井 純，生田 孝至ほか：情報の科学，日本文教出版 (2016).
[16] 赤堀 侃司，永野 和男，東原 義訓ほか：情報の科学，東京書籍 (2013).