

# 「ネットワークを利用した双方向性のあるコンテンツのプログラミング」と 「計測・制御のプログラミング」の統合教材の開発

Development of Integrated Teaching Materials  
in “Programming for Interactive Contents using Network”  
and “Programming for Automatic Measurement and Control”

石 橋 直

Tadashi ISHIBASHI

教職教育院

(平成30年9月28日受付, 平成30年12月3日受理)

中学校学習指導要領の改訂を受け、技術・家庭科技術分野の「D 情報の技術」においては従前からの内容である「プログラムによる計測・制御」に加えて、「ネットワークを利用した双方向性のあるコンテンツのプログラミング」が新設された。技術分野は内容が増加したにもかかわらず授業時間数が変化していない現状に加え、この領域における効果的な教材に関する情報が不足している。本研究では、少ないコードでグラフィック描画が可能な初心者向けプログラミング言語の Processing と、多数のセンサや IO をもつ安価なデバイスである micro:bit を統合した学習内容を検討し、2つのプログラミングの内容を連続的に学習できる統合教材について提案する。

## 1. はじめに

### 1.1 学習指導要領改訂によるプログラミング教育の拡充

平成 29 年 3 月に中学校学習指導要領<sup>1)</sup>が告示され、中学校技術・家庭科技術分野（以下「技術科」）においては、「D 情報の技術」に「(2) ネットワークを利用した双方向性のあるコンテンツに関するプログラミングによる問題解決」が新設されるとともに、従前の「プログラムによる計測・制御」に引き続き、「(3) 計測・制御のプログラミングによる問題解決」が示された。また、小学校学習指導要領総則<sup>2)</sup>においては、「コンピュータ等や教材・教具の活用、コンピュータの基本的な操作やプログラミングの体験」とあり、プログラミング教育の必修化が示された。さらに、平成 30 年 3 月に告示された高等学校学習指導要領<sup>3)</sup>においては、情報科の共通必修科目「情報 I」に、「2 内容 (3) コンピュータとプログラミング」として、コンピュータで情報が処理される仕組み、プログラミングやシミュレーションによって問題を発見・解決する活動を取り入れることが示された。したがって、小学校、中学校、高等学校を通したプログラミング教育が一貫して実施されることとなった。これらの背景には、我が国が目指すべき未来社会の姿として政府が提唱する Society5.0 の実現に向けた人材育成の推進<sup>4)</sup>などがあり、産業界の要請を色濃く反映したものであると言える。

このように、今回の改訂においてプログラミング教育が学校教育で強く推進されてきており、それに伴って技術科で取り扱われるプログラミングの内容も高度化しているが、対応する教材や実践例が不足しているといった課題がある。そこで本研究では、プログラミングに関連する学習内容である「ネットワークを利用した双方向性のあるコンテンツに関するプログラミング」と「計測・制御のプログラミング」を連続的に実

施できる統合教材を提案することとした。

## 1.2 履修内容の新設に伴う新たな教材の需要

学習指導要領の改訂によって、技術科における情報技術に関する内容が倍増されたにもかかわらず、授業時間数に変化はなかった。このような中、「ネットワークを利用した双方向性のあるコンテンツのプログラミング」の新設を受けて、新たな教育方法や教材・教具の研究開発の必要性、さらに技術科教員が抱える困難の解消といった課題が指摘されている<sup>5) 6)</sup>。学習指導要領の告示から現在まで、各種書籍や論文にて、具体的な実践例とともに活用できるプログラム開発環境等の提案が進められてきた。ビジュアル・プログラミング環境については、Scratch1.4のMESH機能やApp Inventorの実践例が挙げられている<sup>7) 8) 9)</sup>。また、実社会で使用されているHTMLとJavaScriptを体験させることを目指した独自のプログラミング環境の開発に取り組む事例も報告されている<sup>10) 11)</sup>。これらの研究によって、新設内容に関する効果的な教育が期待されるが、「ネットワークを利用した双方向性のあるコンテンツのプログラミング」に特化した内容となっている。そのため、授業時間数の増加を見込めなかった技術科においては、効率的に各領域・項目の内容を取り扱うという視点からの横断的な教材の開発が今後求められる。

## 2. 教材の概要

今回の改訂によるこれら2つの学習は、いずれも「生活や社会における問題を…（中略）…プログラミングによって解決する活動」として位置付けられている。「ネットワークを利用した双方向性のあるコンテンツのプログラミング」と「計測・制御のプログラミング」を統合的に取り扱える身近なテーマの一つにIoT（Internet of Things）がある。本研究では、IoTを志向した教材の構成を考えた。また、これからの技術科におけるプログラミング教育は小学校と高等学校の接続を意識した内容として構築する必要がある。現在の小学校における実践は、ビジュアル・プログラミング環境によるものが多い<sup>12) 13)</sup>。高等学校においては、現行の「情報の科学」の教科書によれば、HTMLやCSSの他、JavaScript<sup>14) 15)</sup>、VBA<sup>16) 17)</sup>などのテキスト型プログラミング言語が多く見られる。技術科の学習指導要領解説には、「小学校における学習を発展させるとともに…（中略）…高等学校における情報関係の科目との連携・接続に配慮する。」としているため、中学校段階ではビジュアル型言語からテキスト型言語への橋渡しとして機能する教材の選定が効果的であると考えられる。そこで、ビジュアル・プログラミングによって多様な計測・制御の学習が可能となるmicro:bitと、Javaをベースとしたテキスト型の初心者向けプログラミング言語であるProcessingを組み合わせた統合教材について検討した。教材の構成を図1に示す。

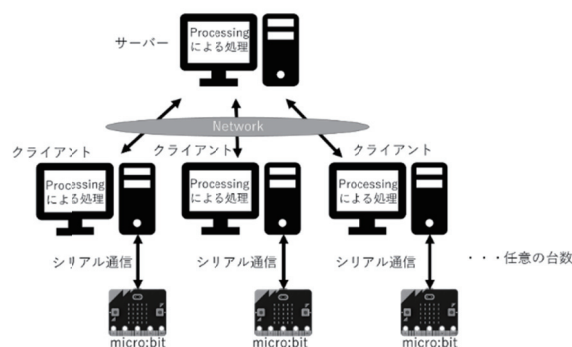


図1 教材の構成

## 3. Processingによるネットワークを利用した双方向性のあるコンテンツのプログラミング

### 3.1 Processingについて

Processing<sup>18)</sup>とは、Javaを単純化したオブジェクト指向のプログラミング言語である。グラフィック機能に特化しており、少ないコードで視覚的なフィードバックを即座に得られることから、初心者向けの言語として知られている。また、シリアル通信やネットワーク等のライブラリも豊富で拡張性が高いことが特徴

```

1 void setup(){
2   size(480,120);
3 }
4
5 void draw(){
6   ellipse(mouseX, mouseY, 80, 80);
7 }

```

図2 Processing のプログラム例

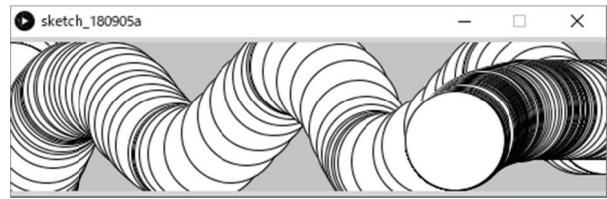


図3 実行結果

である。オープンソースであり、プログラムをデスクトップ等にダウンロードして展開するだけで使用可能となる。例えば、図2のようなシンプルなコードで、図3のような複雑なグラフィックを記述することができる。これは、480 × 120 ピクセルのウィンドウに、幅 80、高さ 80 ピクセルの円をマウスの軌跡に沿って連続的に描画するものである<sup>19)</sup>。

### 3.2 Processing によるチャットプログラム

学習指導要領解説によると、互いにコメントなどを送受信できる簡易なチャットを教室内で再現する活動を通して、利便性や安全性を高めるための機能を追加していくといった問題解決活動が例示されている。なお、必要に応じて、参考となるプログラムを用意したり、あらかじめ教師が実装しておいたりするなどの配慮についても述べられている。このプログラムについて、Processing によって制作する学習活動を検討した。

Processing にネットワークのライブラリをインポートすることによって、簡易的なチャットプログラムを作成できる。このプログラムは、サーバー側プログラムを先に起動し、次にクライアント側を起動することで動作する（図4・5）。クライアント側はサーバーの IP アドレスを指定し、マウスのクリックによって図5の19行目の文字列をサーバーに送信する。サーバーは受信した文字列を表示するとともに、接続されているクライアント全てに受信した文字列をそのままブロードキャストする。クライアント側はサーバー側と同様の画面を表示する。このチャットプログラムによって文字列の送受信はできるが、ソースコード内の文字列をそのまま出力するため、プログラム実行中に任意の文字列を自由に入力・送信できないことや、最新の発言のみを表示するだけで過去の発言履歴を閲覧できないといった「利便性」の低いものとなっている。また、誰が発言したのか分からないといった「安全性」にも課題がある。そこで、図9のように、入力した文字列が送信者の IP アドレスとともに10件ずつ更新表示されるように改善・修正することを通して、課題の解決に取り組ませることができる。サーバー側のプログラムでは、直近に受信した文字列10件分を

```

1  /*Simple Chat Server */
2  import processing.net.*;
3  Server s;
4  Client c;
5  String str="";
6
7  void setup()
8  {
9    size(300, 200);
10   s = new Server(this, 10001);
11   background(0);
12 }
13
14 void draw()
15 {
16   c = s.available();
17   if (c != null && c.available() > 0){
18     background(0);
19     str = c.readString();
20     text(str, 50,50);
21     s.write(str+"\n");
22   }
23 }

```

図4 簡易チャット（サーバー側）

```

1  /*Simple Chat Client */
2  import processing.net.*;
3  Client c;
4
5  void setup() {
6    size(300, 200);
7    c = new Client(this,
8      "192.168.1.10", 10001);
9    background(0);
10 }
11
12 void draw() {
13   if (c != null && c.available() > 0){
14     background(0);
15     text(c.readString(), 50,50);
16   }
17 }
18
19 void mouseClicked() {
20   c.write("hello");
21 }

```

図5 簡易チャット（クライアント側）

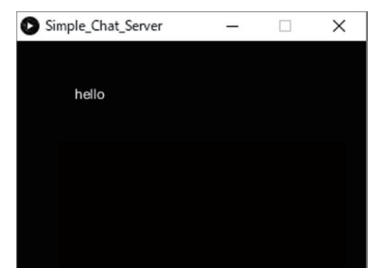


図6 サーバー側実行画面

```

1  /* Chat Server */
2  import processing.net.*;
3  Server s;
4  Client c;
5  int textLine = 10;
6  String[] str = new String[textLine];
7  int i=0;
8
9  void setup()
10 {
11   size(600, 300);
12   textSize(16);
13   s = new Server(this, 10001);
14   background(0);
15 }
16
17 void draw()
18 {
19   c = s.available();
20   if (c != null && c.available() > 0){
21     background(0);
22     str[i]=nf(hour(), 2)+"-"+nf(minute(), 2)+"-"+
23       nf(second(), 2)+"("+c.ip()+ "): "+ c.readString();
24     for (int j=0; j<=i; j++) {
25       text(str[j], 16, j*25 + 20);
26       s.write(str[j]+"¥n");
27     }
28     i++;
29     if (i==textLine) {
30       for (int j=0; j<textLine-1; j++){
31         str[j] = str[j+1];
32       }
33       i=textLine-1;
34     }
35 }

```

図7 改良版チャット（サーバー側）

```

1  /* Chat Client */
2  import processing.net.*;
3  Client c;
4  String tempStr;
5
6  void setup() {
7    size(600, 300);
8    textSize(16);
9    c = new Client(this, "192.168.1.10", 10001);
10   tempStr="";
11   background(0);
12 }
13
14 void draw() {
15   if (c != null && c.available() > 0){
16     background(0);
17     text(c.readString(), 16, 20);
18   }
19 }
20
21 void keyPressed() {
22   if (keyCode == 10) { //10:[Enter]
23     c.write(tempStr);
24     tempStr = "";
25     return;
26   } else if (keyCode == 8 && tempStr.length() > 0) {
27     tempStr = tempStr.substring(0, tempStr.length() - 1);
28   } else if (keyCode != 13) { //13:[BackSpace]
29     tempStr+=key;
30   }
31   println("Message = " + tempStr);
32 }

```

図8 改良版チャット（クライアント側）

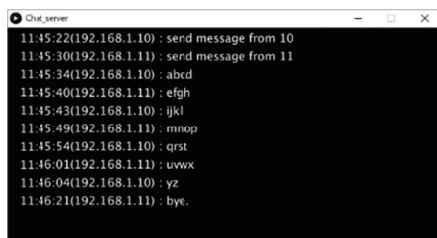


図9 実行画面（サーバー側）

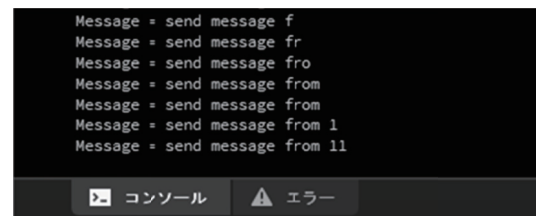


図10 実行画面（クライアント側コンソール）

配列データとして一時保管し、受信処理を行うたびにその10件分のデータを配信する。受信時にはサーバー機の時刻と送信者のIPアドレスを同時に記録する。クライアント側のプログラムでは、新たな発言があるたびにサーバーから送信される10件分のデータを随時更新しながら表示する。送信する場合はキーボードから文字列を入力し、Enterキーを押すことで送信される。文字列はBackSpaceキーで編集も可能となっており、送信待ちの文字列はコンソール画面（図10）で確認できるようになっている。この仕組みを応用すれば、座標情報等を送受信することで、共有描画ボードなどの制作も可能となる。

#### 4. micro:bit による計測・制御のプログラミング

##### 4.1 micro:bit について

micro:bit<sup>20)</sup> は英国のBBCが開発した教育用シングルボードコンピュータのことである（図11）。2017年から国内において販売が開始され、現在では小学校のプログラミング教育や中学校技術科教育、民間のプログラミング教室などに急速に普及しているデバイスである。CPUにはARM Cortex-M0を採用し、2つの押しボタン、加速度センサ、磁気センサ、温度センサに加え、光センサを兼ねた5×5LEDマトリクスディスプレイ、3つのリングコネクタと20のエッジコネクタによるIOポート、BLE（Bluetooth Low Energy）



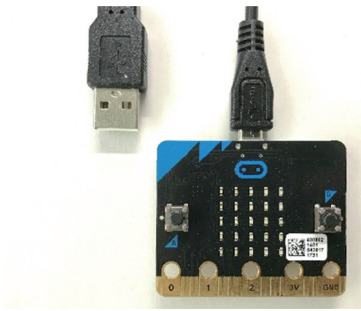


図 11 micro:bit の外観



図 12 micro:bit のプログラム開発画面

通信機能を有する。USB 給電または 3 V の外部電源で駆動し、プログラムは Web 上で作成した実行ファイルを USB 経由で転送することで動作する。Web 上でのプログラム作成方法はブロックによるビジュアル・プログラミング（図 12）となっているが、テキスト型プログラミングへの発展も考慮されており、ブロックと JavaScript のコードが一对一で対応し、互いのソースコードをワンクリックで往還できるようになっている。さらに、専用のアプリケーションを導入することで Python による開発もでき、mbed21) の開発環境を用いれば C/C++ による開発も可能である。一台 2,000 円と安価であり、学習指導要領解説に示されているセンサ、コンピュータ、アクチュエータ等の計測・制御システムの要素や、アナログ信号とデジタル信号、インタフェース等に関する内容を取り扱うことができる。

#### 4.2 計測・制御における micro:bit の活用

「計測・制御のプログラミング」の学習においては、計測・制御システムの仕組みの理解や、問題解決のための計測・制御システムを構想できること等がねらいとして示されている。計測・制御システムの取扱いにおいては、実物を動作させることを通して、順次、分岐、反復という情報処理の手順や構造を理解させる学習が適切であると考ええる。技術科の授業では、従来よりセンサを搭載した自走式車両の制御等でこの内容を取り扱っており、処理の流れを把握させるためにフローチャートが多用されてきた。そこで、従来の流れを汲み、プログラミングの初心者でも簡単なプログラムを自力で構築できるように、フローチャートと命令ブロックを対応させ組み合わせていくような段階的な課題を通して、「計測・制御のプログラミング」の学習を進められると考えた。

図 13 に課題スライドの例を示す。micro:bit の基本的な使い方を説明する過程において「順次」の意味と表し方を取り扱う。課題 01 は、左上と右下の LED を交互に 3 回点滅させるもので、ここでは「反復」の

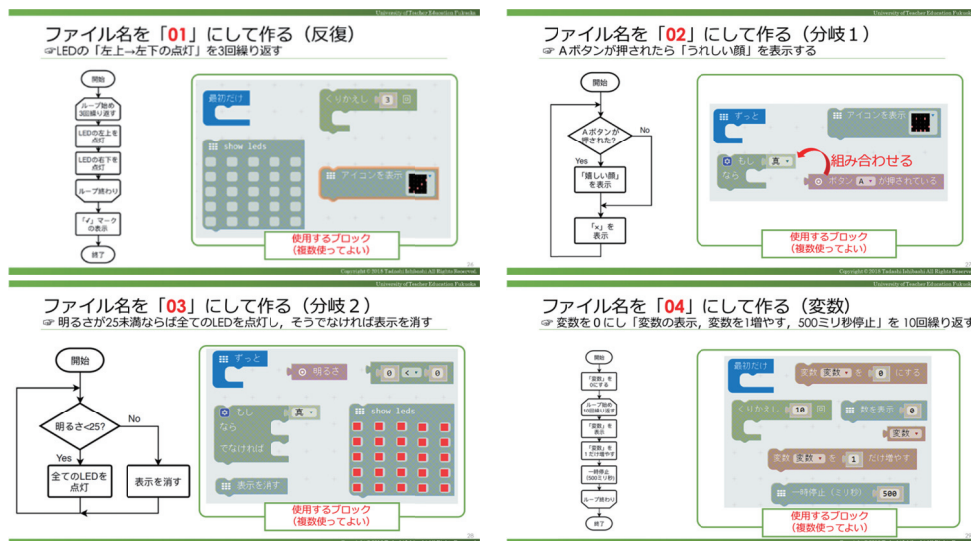


図 13 micro:bit プログラミング課題の例

意味と表し方について体験的に学ぶ。課題 02・03 では「分岐」について、課題 04 では変数について取り扱う。さらに、イベント駆動によるプログラム、各種センサ入力を用いたプログラム、I/O ピンによる外部機器の制御を行うプログラム等に取り組ませることで、計測・制御システムの理解や計測・制御システムの構想に必要な基礎的な力を身に付けさせることができる。

## 5. Processing と micro:bit を連携させたプログラミング（統合教材）

### 5.1 PC と micro:bit 間のデータ送受信

IoT を志向した教材にするためには、まず、PC と micro:bit 間でのデータの送受信が必要となる。ここでは、シリアル通信機能を使用した。micro:bit とシリアル通信するには、あらかじめ PC にドライバ<sup>22)</sup>をインストールする必要がある。ここでのプログラムの動作は、micro:bit から受信した明るさと傾き（ロール）の数値に応じて画面に表示する円の色や大きさを変化させるとともに、マウスのクリックによって文字「,」を PC から送信し、そのイベントによって micro:bit に「嬉しい顔」を表示するものである。micro:bit 側のプログラムと、Processing 側のプログラム、実行の様子を図 13・14・15 に示す。プログラムには、接続されている micro:bit の COM ポート番号の設定や、ボーレートの設定（デフォルトで 115200）が必要になる。PC が micro:bit からデータを受信すると、serialEvent () が呼び出され、null でなければ文字列 str にデータが格納される。この文字列 str を split () によって「,」で区切り、明るさと傾きの整数値に分離して配列として格納する。これらの明るさや傾きの数値を、円の色や大きさに対応させる。なお、コンソールには受信データの履歴が数値表示される。このプログラムを基に、micro:bit のセンサの種類を変更したり、Processing 側の表示されるグラフィックを変更したりすることによって、より発展的・創造的な計測・制御のプログラミングの学習が可能となる。

```

1  /* Serial Communication between PC and micro:bit */
2  import processing.serial.*;
3
4  Serial microbit;
5  int light = 0;
6  int roll = 0;
7
8  void setup() {
9    size(400, 400);
10   frameRate(30);
11   microbit = new Serial(this, "COM1", 115200);
12 }
13
14 void draw() {
15   background(0);
16   fill(255, light*5, light*5);
17   ellipse(width/2, height/2, roll*5, roll*5);
18 }
19
20 void serialEvent(Serial microbit) {
21   try{
22     String str = microbit.readStringUntil('\n');
23     if (str != null) {
24       trim(str);
25       int[] data = int(split(str, ','));
26       println("data[0] = "+ data[0] + ", data[1] = " + data[1]);
27       light = data[0];
28       roll = abs(data[1]);
29     }
30   }catch(Exception e){
31     println("error!");
32     e.printStackTrace();
33   }
34 }
35
36 void mousePressed() {
37   microbit.write(',');
38 }

```

図 13 PC と micro:bit 間の通信プログラム（Processing 側）

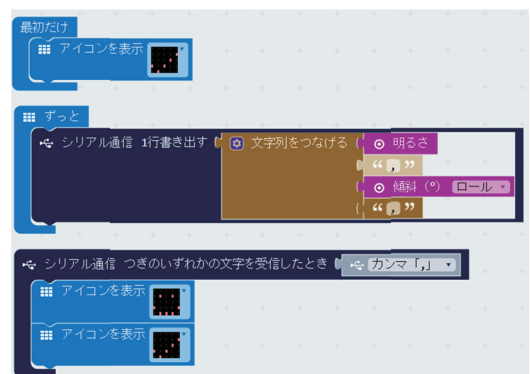


図 14 PC と micro:bit 間の通信プログラム（micro:bit 側）

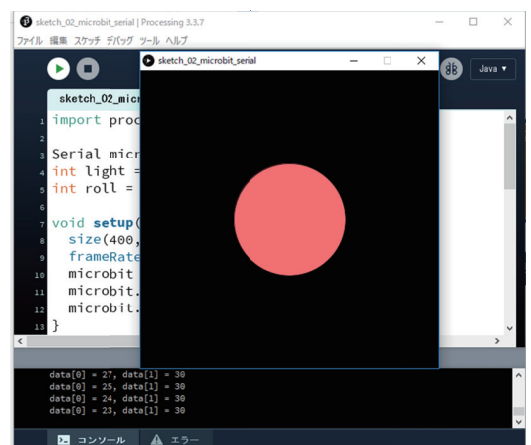


図 15 実行画面

## 5.2 ネットワーク内の PC に接続した micro:bit のセンサ値可視化プログラム

最後に、図 1 をモデルとして、サーバー 1 台、クライアント 2 台の構成を想定したプログラムを制作した (図 16・17・18)。プログラムの動作は、各々の PC が接続された micro:bit のセンサ値をサーバーで集約し、

```

1  /* Draw Graph Server */
2  import processing.net.*;
3  Server s;
4  Client c;
5
6  PrintWriter csv;
7  float time;
8  int val0=0, val1=0;
9  int x;
10
11 void setup() {
12   size(800, 520);
13   frameRate(30);
14   s = new Server(this, 10001);
15   background(0);
16   String filename = nf(year(), 4) + nf(month(), 2) +
17     nf(day(), 2) + nf(hour(), 2) + nf(minute(), 2);
18   csv = createWriter(filename + ".csv");
19   csv.println("time,IP,channel, data");
20 }
21
22 void draw() {
23   c = s.available();
24   if (c != null && c.available() > 0) {
25     String str = c.readString();
26     csv.println(nf(hour(), 2) + ':' + nf(minute(), 2) + ':' +
27       nf(second(), 2) + ',' + c.ip() + ',' + str);
28     println("received data = " + str);
29     int[] data = int(split(str, ','));
30     switch(data[0]) {
31       case 0:
32         val0 = int(data[1]);
33         break;
34       case 1:
35         val1 = int(data[1]);
36         break;
37       default:
38         break;
39     }
40     s.write(str);
41   }
42
43   fill(0);
44   stroke(0);
45   rect(10, 30, 140, 30);
46   rect(10, 30+height/2, 140, 30);
47   fill(255);
48   textSize(20);
49   text("val 0 = " + val0, 10, 50);
50   text("val 1 = " + val1, 10, 50+height/2);
51
52   stroke(0);
53   line(x, 0, x, height);
54   stroke(255);
55   line(x, height/2, x, height/2 - val0);
56   line(x, height, x, height - val1);
57   line(0, height/2, width, height/2);
58   line(x+1, 0, x+1, height);
59
60   x++;
61   if (x>width) {
62     x=0;
63   }
64 }
65
66 void dispose() {
67   csv.flush();
68   csv.close();
69 }

```

図 16 センサ情報の可視化プログラム (サーバー側)

```

1  /* Draw Graph Client */
2  import processing.serial.*;
3  import processing.net.*;
4  Client c;
5
6  int val0=0, val1=0;
7  int x;
8
9  Serial microbit;
10
11 void setup() {
12   size(800, 520);
13   frameRate(30);
14   c = new Client(this, "192.168.1.10", 10001);
15   String mPort = Serial.list()[0];
16   microbit = new Serial(this, mPort, 115200);
17   background(0);
18 }
19
20 void draw() {
21   ...
22   サーバー側プログラムの 23~61 行目と同じ
23 }
24
25 void serialEvent(Serial microbit) {
26   try {
27     String str = microbit.readStringUntil('\n');
28     if (str != null) {
29       trim(str);
30       int[] inpdata = int(split(str, ','));
31       println("data[0] = " + inpdata[0]);
32       c.write("0," + inpdata[0]);
33       if (inpdata[0] > 200) {
34         microbit.write(',');
35       }
36     }
37   }
38   catch(Exception e) {
39     println("error!");
40     e.printStackTrace();
41   }
42 }

```

図 17 センサ情報の可視化プログラム  
(クライアント側)



図 18 micro:bit 側プログラム



図 19 実行画面

	A	B	C	D	E	F	G	H
1	time	IP	channel	data				
2	13:29:17	192.168.1.10	0	34				
3	13:29:17	192.168.1.11	1	47				
4	13:29:18	192.168.1.10	0	36				
5	13:29:18	192.168.1.11	1	46				
6	13:29:19	192.168.1.10	0	32				
7	13:29:19	192.168.1.11	1	46				
8	13:29:20	192.168.1.10	0	28				
9	13:29:20	192.168.1.11	1	7				
10	13:29:21	192.168.1.10	0	24				

図 20 CSV データ

サーバーとクライアントともにデータの推移をグラフ表示するものとした（図 19）。クライアント側は、固有の識別番号（この場合は「0」と「1」）と、接続された個々の micro:bit のセンサ値をサーバーに送信する。サーバーは受け取った文字列に基づき、識別番号に応じてセンサ値を変数 val0 と val1 に振り分ける。センサ値によって画面上の数値および棒グラフを変化させることで、センサ値の変化を可視化できるようにした。このプログラムを拡張させ、クライアントの台数を増やしたり、センサの種類を変更したりすることによって、リアルタイム性のある多点観測ができる。さらに、サーバー側は受信データを CSV 形式で自動保存でき、時間、送信側の IP アドレス、識別番号、センサ値等をログに残すことができる（図 20）。

以上のように、Processing と micro:bit を組み合わせて使用することによって、新学習指導要領に示された「ネットワークを利用した双方向性のあるコンテンツのプログラミング」と「計測・制御のプログラミング」をカバーできるとともに、学習内容を系統的・連続的なものにできる。

## 6. おわりに

本研究では、平成 29 年 3 月に示された中学校学習指導要領における「ネットワークを利用した双方向性のあるコンテンツに関するプログラミングによる問題解決」と「計測・制御のプログラミングによる問題解決」の内容を系統的・統合的に取り扱うことができる教材として、Processing と micro:bit を併用したシステムについて提案した。学校現場からは新たな学習内容に対して不安の声があげられているが、今回提案した教材はその課題解決の一助になるものと考えている。また、教材の選定にあたっては、教員・生徒ともに作業的・経済的な負担の小さいものが望まれており、インストール不要で安価なものが好まれる。Processing と micro:bit は、これらの条件をクリアしており、中学校現場に適しているものと考えている。なお、Processing についてはテキスト型プログラミング言語であるために、中学生にとっては難易度が高いのではと教員側にその導入を躊躇させることが予想される。しかし、小学校からのプログラミング教育の必修化によって、これまで技術科で取り扱っていた内容が段階的に小学校から実施されるようになるため、系統的な学習を行えば小学校でそのような学習を経た中学生には十分に対応可能であると考えている。今後は、技術科で育む資質・能力に照らして更に横断的な視点に立ち、「材料と加工の技術」「生物育成の技術」「エネルギー変換の技術」と「情報の技術」を融合させた教材の開発に取り組んでいく予定である。

## 謝辞

本研究の一部は、JSPS 科研費（課題番号：18K02981「STEM 教育を指向した科学概念育成のための理科教材の開発と実践」、研究代表者：伊藤克治）の助成を受けて行ったものである。ここに記して謝意を表します。

## 参考文献

- 1) 文部科学省：中学校学習指導要領，（2017）
- 2) 文部科学省：小学校学習指導要領，（2017）
- 3) 文部科学省：高等学校学習指導要領，（2018）



- 4) 文部科学省：Society5.0に向けた人材育成の推進，(2018)
- 5) 香西孝行，他：地図サービスを取り入れた双方向性のあるコンテンツのプログラミング教材の開発，鳴門教育大学情報教育ジャーナル No.15 (1) pp.37-41, (2017)
- 6) 木下崇，他：Scratch の Mesh 機能を用いた双方向性のプログラミング教材の開発～中学校技術科のネットワークを用いたコンテンツ制作の導入として～，愛知教育大学技術教育研究, 6, pp.7-12, (2018)
- 7) 古川稔：平成 29 年版 中学校新学習指導要領の展開 技術・家庭 技術分野編，明治図書，(2017)
- 8) 古川稔，他：平成 29 年改訂 中学校教育課程実践講座 技術・家庭，ぎょうせい，(2017)
- 9) 竹野英敏：中学校技術・家庭「技術分野」授業例で読み解く新学習指導要領，開隆堂，(2017)
- 10) 福井昌則，他：双方向性のある Web コンテンツ作成を支援するブロックプログラミング環境の開発，日本産業技術教育学会第 61 回全国大会講演要旨集, p.3, (2018)
- 11) 水門博一，他：双方向コンテンツのプログラミングを支援する Web アプリについて，日本産業技術教育学会第 61 回全国大会講演要旨集, p.85, (2018)
- 12) 文部科学省：プログラミング教育実践ガイド，(2015)
- 13) 中央教育研究所：小学校プログラミング教育ガイド，(2018)
- 14) 水越敏行，他：情報の科学（高等学校情報科用教科書），日本文教出版，(2013)
- 15) 赤堀侃司，他：情報の科学（高等学校情報科用教科書），東京書籍，(2013)
- 16) 岡本敏雄，他：情報の科学（高等学校情報科用教科書），実教出版，(2013)
- 17) 坂村健，他：情報の科学（高等学校情報科用教科書），数研出版，(2013)
- 18) <https://processing.org/>
- 19) Casey Reas，他：Processing をはじめよう第 2 版, p.12, オライリー・ジャパン，(2016)
- 20) <https://microbit.org/>
- 21) <https://www.mbed.com/en/>
- 22) <https://os.mbed.com/handbook/Windows-serial-configuration>

