

センササーバと ScratchX による計測システムの開発

竹田 慎*, 曾根直人**

小学校におけるプログラミング教育の必修化が検討されていることにより、使用する教材として Scratch が注目されている。Scratch はビジュアルプログラミング言語であり、マウスやキーボードの操作が不慣れた学習者でもプログラミングをすることができる。しかしながら、Scratch を中学校技術・家庭科(技術分野)の教材として使用するには、機能が不足している。技術分野の内容では、プログラミングに加え、センサを使用した計測・制御を扱う必要があるためである。本研究では、Scratch を技術分野により適した教材として改良することに加えて、学習者の創造性を重視した問題解決学習につながるシステムを構築した。

[キーワード: プログラミング教育, プログラムによる計測・制御, Scratch, Node-RED]

1. はじめに

文部科学省が小学校からのプログラミング教育の推進を提唱しているのは「プログラミング的思考」の育成を目的としているためである。「プログラミング的思考」については、「小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)」において、「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つひとつの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」と定義している[1]。「プログラミング的思考」では、「情報技術を手段として使いこなしながら、論理的・創造的に思考して課題を発見・解決し、新たな価値を創造する」ことができる子供の育成を目指している。この目標は、中学校においても同様であり、「プログラミング的思考」を考慮したプログラミング教育に適する教材の開発が必要とされている。

本論文で提案するシステムは、Scratch[2]で計測機器のデータを扱えるように計測機器の計測・制御を担当するサーバ、およびそれらを取り扱うためのブロック拡張から構成される。Scratch とは、ビジュアルプログラミング言語の一種であり、マウスやキーボードの操作が不慣れた学習者でもプログラミングできることを特長としている。そのため、小学校のプログラミング教育の教材として注目されて

いる。学習者が遠く離れた場所の状況を分析し、状況に対応したプログラミングを容易に行うことができるように提案システムを構築する。学習者に身近な生活を自動化し、生活を豊かにする方法を考えさせることを通してセンサやプログラミングの有用性を実感させることが期待される。また、身近な生活の改善をテーマに問題意識を持たせ、解決する問題解決学習につなげることもねらう。これは、「プログラミング的思考」につながる学習であり、小学校から中学校への系統的な学習として行えるだけでなく、論理的・創造的に思考して課題を発見・解決し、新たな価値を創造する体験を学習者に提供することが期待できる。

2. システム構成

本システム構成について述べる。全体のシステム構成は、図1に示す通りである。

(1) 学習者端末

学習者端末は、インターネットブラウザがインストールされているコンピュータを使用する。学習者は、ScratchX[3]のウェブサイトアクセスし、ブロックを追加するプログラムファイルをアップロードし、プログラミング環境を利用する。ScratchX とは、Scratch にブロックを追加するウェブアプリケーションであり、Scratch 2.0 と同様に学習者端末へのソフトウェアのインストールは不要でブラウザのみでプログラミング環境を利用できる。

(2) センササーバ

センササーバは、Raspberry Pi を使用する。このサーバは、計測機器の計測・制御および読み出した

* 鳴門教育大学 大学院 (修士課程) 生活・健康系コース (技術・工業・情報)

** 鳴門教育大学 大学院 自然・生活系教育部

データを WebAPI 化し、ネットワーク上に公開する。

本システムでは計測機器の接続方法として GPIO および Bluetooth を用いた。以下に述べる計測機器以外であっても、Raspberry Pi に接続でき計測機器からデータを読み出すことが可能であればシステムに組み込むことができる。WebAPI 化については、Node-RED を使用している。Node-RED は、計測機器から読み出したデータやコマンドラインに出力されたデータを容易に WebAPI 化できる。

3. 学習者端末

学習者端末における ScratchX の利用方法および ScratchX ブロック追加プログラムファイルについて説明する。

(1) ScratchX の利用方法

ScratchX とは、Scratch にブロックを追加するウェブアプリケーションである。学習者端末にソフトウェアのインストールが不要であり、ブロックを追加するプログラムファイルをアップロードすることで利用できる。

(2) ScratchX ブロック追加プログラムファイル

ブロック追加プログラムファイルは、JavaScript で記述される。同ファイルの一例を図 2 に示す。

図 2 は、レポーターブロックの追加プログラム例である。図 3 は、図 2 を ScratchX に読み込ませた時の表示例である。レポーターブロックは、値を格納するブロックであり、その値は演算や条件判定に使用できる。

descriptor 配列は、表示されるブロックの種類や表示内容、実行する関数名を設定している。ブロックの種類として、コマンドブロック、レポーターブロックがある。コマンドブロックは、値を格納せず、処理のみを行う場合に使用するブロックである(図 4)。コマンドブロックは、加速度、角速度および磁気センサで使用している。これらのセンサは、xyz 座標の値が取得できるため、1 つしか値を格納できないレポーターブロックでは使用できない。そのため、コマンドブロックにより、複数の値を ScratchX が起動している HTML ファイルに input タグの type 属性 hidden として挿入し、レポーターブロックにより、input タグの各センサの値を取得するように実装した。この時、複数種類のサーバを識別するために input タグの name 属性に識別名とセンサ値名を追加し、タグの挿入を行っている。

それぞれのブロックには、同期処理または非同期処理が設定できる。同期処理は、計算量の少ない処

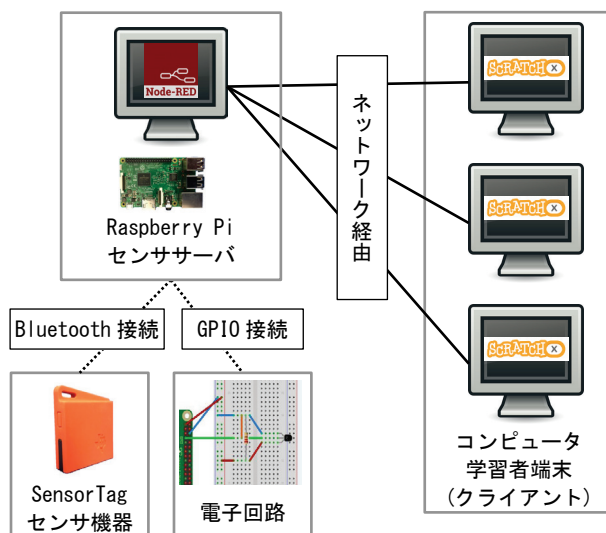


図 1 システム構成図

```
(function(ext) {
  var descriptor = {
    "blocks": [
      /* 温度 */
      ["R", "%s の温度", "temp", "192.168.3.2"],
    ],
    "menus": {}
  };

  ext._shutdown = function() {};

  ext._getStatus = function() {
    return {status: 2, msg: 'Ready'};
  };

  /* 温度 */
  ext.temp = function(ip, callback) {
    $.ajax({
      url: 'http://'+ip+":1880/temp.json",
      dataType: 'jsonp',
      jsonpCallback: 'temp',
      timeout: 5000,
      success: function(data) {
        callback(data['temp']);
      },
      error: function() {},
      complete: function(data) {}
    });
  };

  ScratchExtensions.register("ブロック群のタイトル",
    descriptor, ext);
})({});
```

図 2 ブロック追加プログラムファイル例

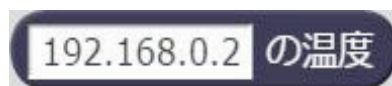


図 3 レポーターブロック表示例



図 4 コマンドブロック表示例

理や動作が保障されている場合に使用する。非同期処理は、計算量の多い処理や通信エラーが起こる可能性がある処理に使用する。

ext オブジェクトでは、ブロックの実行関数を設定している。実行関数は Ajax 通信により、センササーバの WebAPI にリクエストし、レスポンスされた値をブロックの値として格納している。Ajax 通信は、json 形式で値の取得を行っている。ブロックを非同期処理設定にしている場合は、callback 関数の呼び出すことで終了するため、Ajax 通信の success 関数と error 関数において callback 関数を呼び出す必要がある。また、Ajax 通信時、クロスドメイン制約をうけるため、センササーバ側でクロスドメイン制約回避のヘッダを付与する必要がある。クロスドメイン制約については、センササーバの Node-RED の説明において詳述する。

4. センササーバ

センササーバを構成する Raspberry Pi と計測機器の接続方法および WebAPI を作成するための Node-RED[4]について説明する。

(1) Raspberry Pi

本システムでは、Raspberry Pi に計測機器を接続し、WebAPI を提供する。Raspberry Pi のモデルは、Raspberry Pi 3 Model B を使用している(図 5)。採用理由は、サーバとして運用できる処理能力と無線 LAN を搭載しているとともに廉価で世界的に提供され入手性が高いことが挙げられる。Raspberry Pi 3 Model B は、授業時に複数の学習者がサーバにアクセスすることによる負荷に十分対応できると推測される。また、無線 LAN を利用できるため教材の利用場所の制約が少ないという利点がある。廉価でシステムを構築できるため中学校教材として導入しやすく、故障などがあっても対応しやすいというメリットがある。

(2) 計測機器の接続方法とプログラム

Raspberry Pi と計測機器の接続方法およびそれに必要なプログラムについて述べる。

1) GPIO

GPIO とは、Raspberry Pi で利用できる I/O ポートの名称である。電気回路と組み合わせて、多様なセンサを接続できる。本システムでは、温度センサ(DS18B20)および超音波距離センサ(HC-SR04)を用いた。各センサの接続方法およびセンサを使うためのプログラムについて説明する。

① 温度センサ(DS18B20)

温度センサである DS18B20 は、1-Wire 通信を採用したセンサである(図 6)。Raspberry Pi は 1-Wire 通信に対応しているため、DS18B20 と抵抗器による回路を介して GPIO に接続することによって、デバイスとして認識する(図 7)。認識後、コマンドラインから cat コマンドを使用して、センサの値を呼び出すことができる(図 8)。図 8 に例示にしたセンサ値は、表示部分の「t=25750」にあたり、この値の 1000 分の 1 が温度(°C)を示す。

② 超音波距離センサ(HC-SR04)

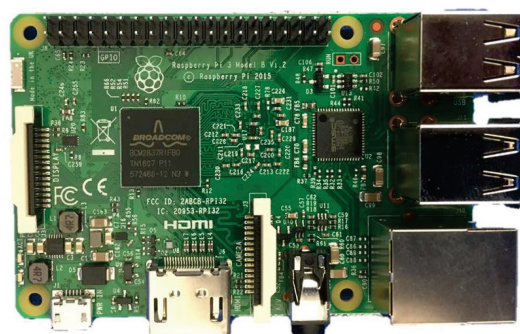


図 5 Raspberry Pi 3 Model B

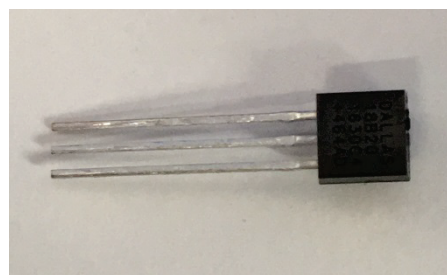


図 6 温度センサ(DS18B20)

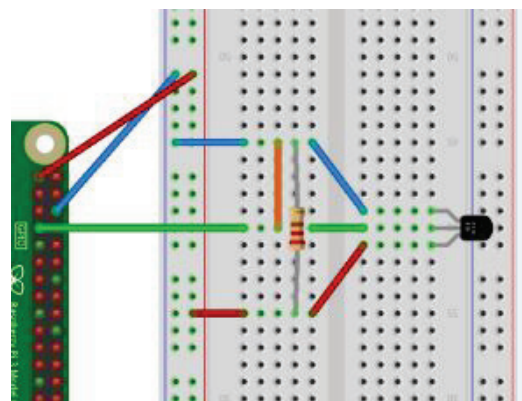


図 7 温度センサ回路の実装例

```
pi@raspberrypi:~$ cat /sys/bus/w1/devices/28-00000647d6f6/w1_slave
9c 01 4b 46 7f ff 04 10 7a : crc=7a YES
9c 01 4b 46 7f ff 04 10 7a t=25750
```

図 8 DS18B20 のセンサ値呼び出し例

超音波距離センサである HC-SR04 は、測定した距離に比例するパルス幅を出力するセンサである(図 9)。図 10 に HC-SR04 と Raspberry Pi を接続する回路の実装例を示す。この回路を用いてセンサ値を取得する場合、Raspberry Pi の GPIO を操作するプログラムを用いる必要がある(図 11)。ここではそのプログラムを Python 言語で記述した。このプログラムを実行すると、センサ値をコマンドラインに出力する。出力結果は、後述する Node-RED から読み出すことができる。

2) Bluetooth

本システムでは、Bluetooth 接続の計測機器である SensorTag[5]を用いた。ここでは、SensorTag のセンサ値を取得するためのプログラムについて説明する。

① SensorTag の概要

SensorTag は、Texas Instruments 社が開発した Bluetooth 接続の計測機器である(図 12)。SensorTag は、光、デジタル・マイク、磁気センサ、湿度、気圧、加速度計、ジャイロスコープ、磁力計、物体の温度および周囲温度を計測できる。Bluetooth 接続であることから、無線の有効範囲であれば、自由に動かすことが可能である。

② SensorTag の接続方法

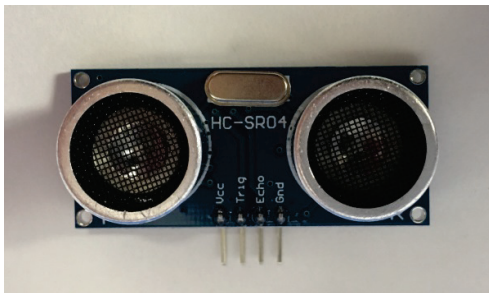


図 9 超音波距離センサ(HC-SR04)

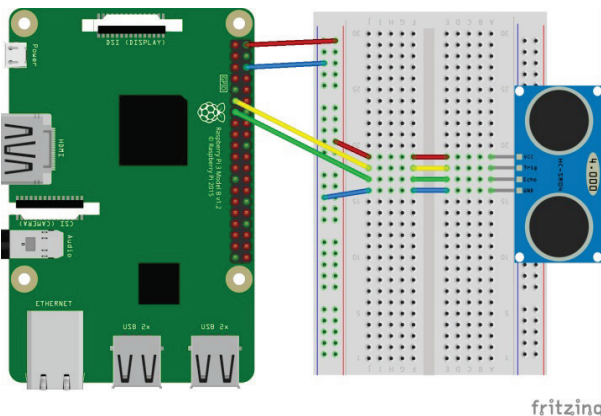


図 10 超音波距離センサ回路の実装例

Raspberry Pi が SensorTag をデバイスとして認識するためには、Bluetooth 用のドライバである BlueZ が必要である。さらに、SensorTag のセンサ値を取得プログラムを作成するために Python 言語のライブラリである Bluepy をインストールする必要がある。

③ SensorTag のセンサ値取得プログラム

本システムでは、Bluepy のサンプルプログラムである `sensortag.py` を改変している。Bluepy のライセンスはパブリックドメインであるため、改変が許可されている。改変した内容は、コマンドラインに表示されるセンサ値をテキストファイルに書き出している部分である。センサ値を取得するには、コマンドラインでこのプログラムを起動し、センサ値を継続的にテキストファイルに書き出す必要がある。この実装を行ったのは、計測機器の接続に遅延がある

```
#!/usr/bin/python

def reading(sensor):
    import time
    import RPi.GPIO as GPIO

    GPIO.setwarnings(False)

    GPIO.setmode(GPIO.BCM)

    if sensor == 0:
        GPIO.setup(17,GPIO.OUT)
        GPIO.setup(27,GPIO.IN)
        GPIO.output(17, GPIO.LOW)
        time.sleep(0.3)
        GPIO.output(17, True)
        time.sleep(0.00001)
        GPIO.output(17, False)
        while GPIO.input(27) == 0:
            signaloff = time.time()

        while GPIO.input(27) == 1:
            signalon = time.time()

        timepassed = signalon - signaloff

        distance = timepassed * 17000

        return distance

    GPIO.cleanup()
else:
    print "Incorrect usonic() function variable."

print str(reading(0))
```

図 11 HC-SR04 センサ値の取得プログラム



図 12 SensorTag (CC2650STK)

こと、ならびに SensorTag の電源ボタンを押さなければならぬことが主な理由である。このプログラムを用いることによって、SensorTag のセンサ値をテキストファイルに書き出し続ける機能を実装できた。

(3) Node-RED

Node-RED とは、計測機器を活用し、WebAPI などのオンラインサービスを作成するプラットフォームである。Node-RED は、Node と呼ばれるプラグインやモジュールがパッケージ化されたブロックとブロックをつなげることで行う。本システムでは、センサ値の取得および WebAPI の作成を Node-RED で行っている。以下に使用した Node を説明する。

1) http Node

Http Node は、リクエストを受ける URL を設定する Node である (図 13)。この Node の URL 設定することで、「http://(サーバ IP アドレス):1880/(設定文字列)」にリクエストを待つ URL が設定される (図 14)。

2) exec Node

exec Node は、コマンドラインにコマンドを入力し、出力された値を取得する Node である。この Node では、センサ値が書き込まれたテキストファイルの読み込み、もしくはセンサ値取得プログラムの実行を Command 設定で行っている (図 16)。

3) function Node

function Node は、プログラムを記述することができる Node である (図 17)。この Node の設定画面では、JavaScript 言語によるプログラムで行う。このプログラムでは、クロスドメイン制約回避のためのヘッダ付与、センサ値の抽出、json 形式への変換を行っ

ている (図 18)。それぞれのプログラムについて説明する。

① クロスドメイン制約回避

クロスドメイン制約とは、あるドメインからダウンロードされた HTML ファイル上で動作するスクリプトはダウンロードサイトと同一ドメインのみ HTTP リ

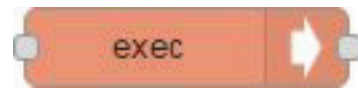


図 15 exec Node

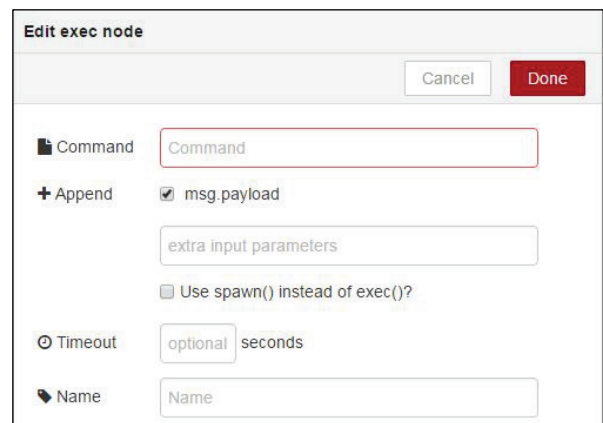


図 16 exec Node 設定画面



図 17 function Node

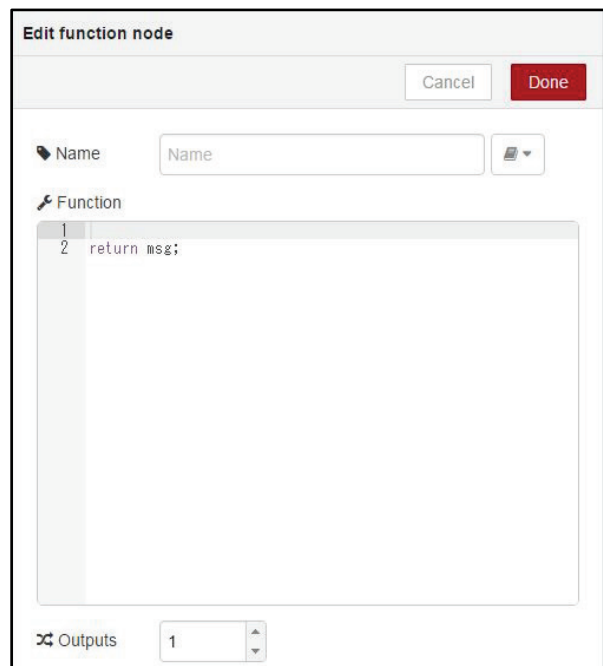


図 18 function Node 設定画面

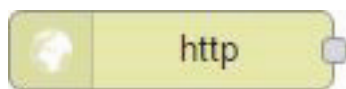


図 13 http Node

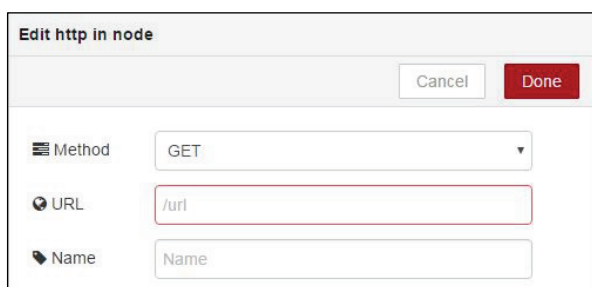


図 14 http Node 設定画面

クエストが許可され、それ以外のドメインにはリクエストが制限されるセキュリティ機構である(図 19)。

このセキュリティ機構は、クロスサイトリクエストフォージェリ対策として、ウェブブラウザが行っている。

クロスドメイン制約を回避するためには、2つの方法があり、1つ目は jsonp 形式での Ajax 通信を行うこと、2つ目はサーバ側のレスポンスヘッダにクロスドメイン制約回避のヘッダを付与することである。本システムでは、ヘッダ付与の方法を採用した。これは jsonp 方式の場合、ランダムな GET 値が付与されることにより、インターネットに WebAPI を公開するときにサーバのセキュリティレベルによっては公開できない場合があるため、jsonp 形式での実装にしていない。ヘッダの付与については、図 20 のヘッダを付与することで回避することができる。

② センサ値の抽出

センサ値の抽出は、exec Node から引き渡された文字列から必要なセンサ値のみを抽出し、変数に格納している。SensorTag の場合は、tsv 形式で受け渡されるため、split 関数で配列に分解し、必要なセンサ値を取得している。

③ json 形式への変換

json 形式への変換は、JSON.stringify メソッドを使用している。センサ値をオブジェクトの変数として宣言し、JSON.stringify メソッドを通して、json 形式に変換している。

4) http response Node

http response Node は、レスポンスを行う Node である(図 21)。この Node では、特に設定項目は無いが、レスポンスを返すために Node-RED プログラムの最後につなげる必要がある(図 22)。

5. おわりに

センササーバと ScratchX による計測システムを用いることによって、遠隔地に設置した計測機器のデータを用いるプログラミング環境を提供できた。学習者はセンサとプログラミングについて学ぶことを通して、提案したシステムを活用し生活を豊かにする方法を考えることができ、問題解決学習につながるることができる。特に、中学校技術分野の内容「プログラムによる計測・制御」において、学習者の生活に根ざした学びにすることができると考えられる。また、開発したシステムは、センサのデータを返す WebAPI を備えている。この WebAPI をインター

ネットに公開し、プログラミングできる方法も実装されている。この方法を用いて学校に配置した計測機器のみならず、計測機器を配置することができるあらゆる場所の測定データを使ったプログラミングも可能となり、地域連携や学校間連携などにも活用が広がる。

今後の課題として、授業実践を踏まえた提案システムの評価がある。学習者の反応や学習効果に関する評価結果に基づき、本システムの問題点を把握し改善していかなければならない。さらに、制御機器の操作、計測機器および制御機器への新たな対応が挙げられる。

本論文で述べたシステムは、制御機器への実装が

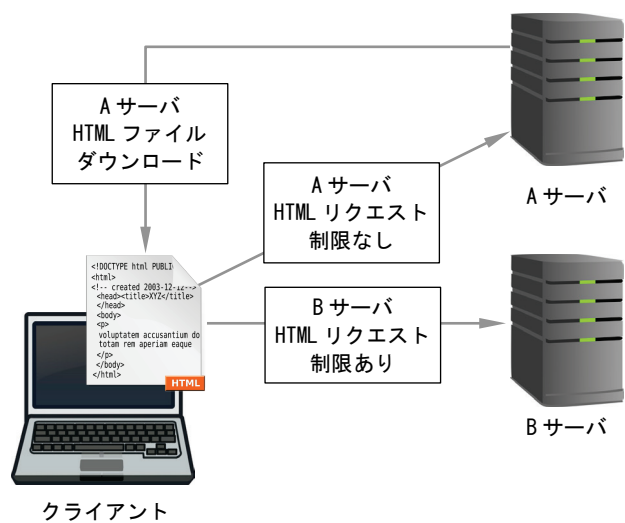


図 19 クロスドメイン制約関係図

```
msg.headers = {
  "Access-Control-Allow-Origin": "*",
  "Access-Control-Allow-Method": "GET, POST, PUT, DELETE, OPTIONS",
  "Access-Control-Allow-Headers": "Origin, X-Requested-With, Content-Type, Accept"
}
```

図 20 クロスドメイン制約回避ヘッダ



図 21 http response



図 22 http response Node 設定画面

できていないため、技術分野の内容「プログラムによる計測・制御」の「計測」に関する学習に利用できるが、「制御」に関する学習に利用できない。今後、照明やモータなどを対象とする制御機器に対応できるようにシステムを改善する必要がある。提案したシステムは、学習者の創造性を主体にしている教材である。授業実践などを踏まえてより創造性を刺激できるような様々な計測機器や制御機器に対応できるようにシステム構成を改良していきたい。

参考文献

- [1] 文部科学省教育課程部会教育課程企画特別部会 (2016) 議事要旨・議事録・配付資料, http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo/3/053/siryo/_icsFiles/afieldfile/2016/07/29/1374873_12.pdf (最終アクセス日: 2017年2月2日).
- [2] Scratch Foundation Scratch, <https://scratch.mit.edu/> (最終アクセス日: 2017年2月2日).
- [3] MIT Media Lab. ScratchX, <http://scratchx.org/> (最終アクセス日: 2017年2月2日).
- [4] JS Foundation Node-RED, <https://nodered.org/> (最終アクセス日: 2017年2月2日).
- [5] Texas Instruments SimpleLink SensorTag, <http://www.tij.co.jp/tool/jp/cc2650stk> (最終アクセス日: 2017年1月15日).
- [6] 竹田慎・曾根直人(2016) プログラムによる計測・制御における ScratchX の応用, 日本教育工学会研究報告集, pp. 201-206.