

インターネットに計測値を送信できる計測・制御教材の提案

林 康平^{1,a)} 西ヶ谷 浩史^{2,b)} 大村 基将^{1,c)} 兼宗 進^{1,d)}

概要: インターネットに計測値を送信できる計測・制御教材を提案する。近年、IoT が普及してきているが、その仕組みについて学べる教材は少ない。そこで本教材では、Wi-Fi に接続可能なマイコンボードである Espr Developer に温度センサなどのセンサやカメラを接続し、Wi-Fi を通じてインターネット上のサーバに計測値や画像の送信、蓄積を行う。また、マイコンボードに接続するセンサ類は生徒が自身が選択し、マイコンへのプログラムも生徒自身で記述する。また、利用するコンピュータにマイコンボードの開発環境をインストールせずにインターネットからプログラムを書き換えることができる。本発表では、開発した教材の設計と実装について報告する。

HAYASHI KOHEI^{1,a)} NISHIGAYA HIROFUMI^{2,b)} OMURA MOTOMASA^{1,c)} KANEMUNE SUSUMU^{1,d)}

1. はじめに

近年、IoT が広がり、家電など身近なものがインターネットに接続し、搭載されたセンサから計測したデータを活用する世の中となってきた。このような機器が増えてきているが、その仕組みについて学ぶことができる教材は少ない。そのため、センサでの計測や、その計測値を使った制御、さらには、データの蓄積や解析といったことを学習できる教材が求められていると考えた。

そこで、Wi-Fi に接続できるマイコンである ESP8266 にカメラやセンサを接続し、生徒がカメラでの撮影やセンサでの計測をプログラミングできる仕組みを構築した。使用する生徒がマイコンボードにセンサ類を配線したり、計測・制御のためのプログラムを作成する。また、撮影した写真やセンサでの計測値をインターネット上のサーバに蓄積する仕組みを構築した。

本稿では、開発した教材の設計と実装について報告する。

2. IoT とは

近年、家電のなかにはインターネットに接続する機能を有するものが登場しはじめた。それらの中には、搭載されたセンサが計測したデータを外出中にスマートフォン等で確認したり、外出先からスマートフォンをもちいて操作ができるものも登場し、IoT 家電と呼ばれている。

IoT(Internet of Things) は、あらゆる「モノ」がセンサや無線通信を介してインターネットの一部を構成することで実現する、ユビキタスネットワークの構築形態の一つである [1]。

スマートフォンなどから色や明るさを制御可能な LED 電球である Philips Hue[2] や、プリンタなどに搭載されたセンサから消耗品の残量を計測し、少なくなった際に自動的に注文を行う Amazon Dash Replenishment Service[3] など、すでに我々の生活の中に IoT は浸透し始めている。

IoT では、まず、対象とするモノについての情報をセンサによりデジタル化することでデータを収集し、収集したデータをネットワークを介して配信、あるいは蓄積する。さらに、集めた膨大な情報を解析することで、現実世界のサービス等へと活用する。IoT の概要図を図 1 に示す。

これら IoT は、大きく 3 つの技術からの構成となる。1 つ目が、もののデジタル化を支える「センシング」技術である。光、音、温度など、自然界に存在する要素を計測し、量的データとして数値化する技術の総称をセンシング技術であり、計測する要素に応じて様々なセンサが存在する。

¹ 大阪電気通信大学
Osaka Electro-Communication University, Neyagawa, Osaka
572-8530, Japan

² 焼津市立小川中学校
Kogawa Junior High School, Yaizu, Shizuoka 425-0035,
Japan

a) mm17a011@oecu.jp

b) hnishigaya@gmail.com

c) omura@osakac.ac.jp

d) kanemune@osakac.ac.jp

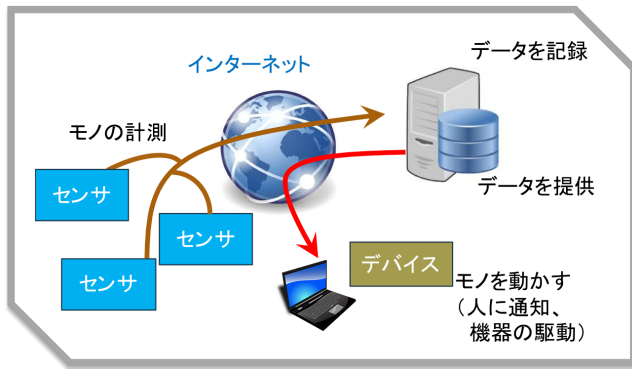


図 1 IoT 概要

さらに個々の要素だけでなく、ある状態を識別するために複数のセンサを組み合わせることもよく行われる。

2つ目がデータ蓄積を支える「データベース」技術である。膨大なデータを蓄積し、それらのデータを必要に応じて検索し、提供するための技術であり、多くの Web システムの中核の一翼を担う技術である。

3つ目がデータや機器同士を有機的に結びつける「ネットワーク」技術である。独立して存在するコンピュータやデータを結びつけ、情報のやり取りを実現する技術である。

それぞれの技術は計測・制御やネットワークを支えるコア技術であり、現代の制御技術やインターネット等の仕組みを理解する上で、欠かすことができない。これら技術を組み合わせシステムを構築する IoT 機器を開発する活動は、それぞれの技術的内容の理解を深めるとともに、単独のモノや技術にはない新たな価値が付与する仕組みを創造する活動となりうる。このような点から、技術・家庭科（技術分野）や、高校教科情報の教材とすることは非常に有効であると考えた。

3. IoT を学習できる計測・制御教材

3.1 コンセプト

本教材では、IoT のコア技術である「センシング」、「データベース」、「ネットワーク」の役割を体験的に理解し、それらの技術を組み合わせて、生徒らがプログラミングにより実際動作する IoT 装置の作成できるものを目指した。このとき、作成する IoT 機器が計測する題材としては、IoT がもつ時間や場所を問わずモノの状態を得ることができるという特徴を体感でき、かつ、計測結果を蓄積し調査や改善に活用する思考に学習者が至るような開発動機を学習者に抱かせるものが望ましい。そこで本教材は、学習者自身が時間を問わず状況を確認したいと感じ、長期間の観察活動が行われる植物・生物の育成状況の調査や気象など長時間の計測調査での利用を想定して開発を行った。

これらの内容を実現するため、本教材に求められる性質を下記と設定した。

(1) 目的に応じて、制作する IoT 機器のセンサや制御する

回路を変更できること

- (2) インターネットを通じて IoT 機器の集めた情報どこでも確認できること
- (3) 場所を問わずに利用できること
- (4) 長期にわたり観察結果を蓄積し、蓄積されたデータは誰でも利用ができること
- (5) プログラミングにより、生徒自身の課題に合わせて IoT 機器の計測内容や公開・蓄積する情報の内容を制御できること

3.2 学校内で IoT 機器を利用する際の課題

中学校では、コンピュータ室などの情報機器の管理は教育委員会単位で行われることも多く、授業を行う教員にコンピュータやネットワーク等に対する管理の権限を与えられていない場合も少なくない。アプリケーションやドライバの追加導入が禁止されたり、導入可能であっても、更新時期の制限がかけられたり、教材の導入ができない場合や、学校に合わせたカスタマイズなど迅速なサポートができない場合がある。このため、本教材の導入にあたっては、可能な限りコンピュータへのソフトウェアへのインストールや更新作業を減らす仕組みが求められる。この問題の是正のため、本教材では作成したプログラムのコンパイルを Web システム上で実施し、作成したプログラムを制御対象の IoT 機器に転送する仕組みを導入し、学校のコンピュータやネットワーク環境などへの依存度をできるだけ小さくする方法を採用した。

3.3 開発教材の概要

ここまでの教材のコンセプトと課題の検討を元に、無線通信機能を持つマイコン ESP8266 と、教育用プログラム言語ドリトル、および、プログラムのコンパイルや受け取ったデータの蓄積を行うサーバからなる IoT 制御教材を開発した。本教材の構成を図 2 に示す。

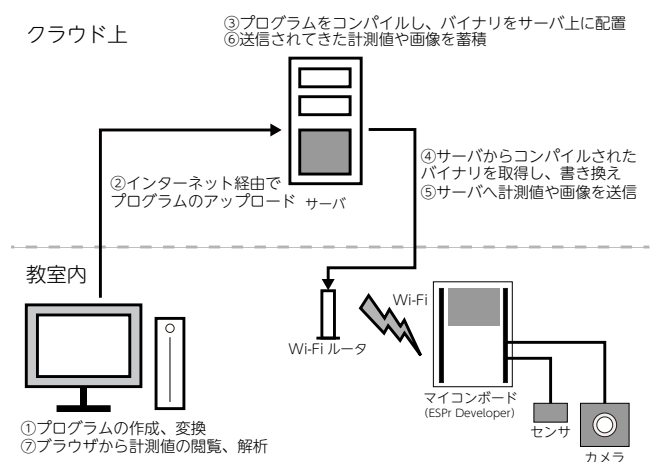


図 2 システム構成図

利用者のプログラムの記述から、マイコンボードへの書き込みまでの流れを説明する。まず、利用者がプログラムはドリトルを用いてプログラムを記述する。ドリトルでの記述後、記述したプログラムを実行することで、利用者の記述したプログラムが Arduino 言語に変換され、ファイルに書き出される (図 2 ①)。その書き出されたファイルをブラウザからアップロードする (図 2 ②) と、サーバ上でコンパイルが行われ、バイナリがサーバ上に配置される (図 2 ③)。なお、このアップロードの際にプログラムが書き込まれる対象のマイコンボードの選択を行う。コンパイルの完了後にマイコンボードの電源を入れると、マイコンがサーバにプログラムの更新の問い合わせを行い、プログラムを取得する (図 2 ④)。

次に、マイコンがサーバ上にデータを蓄積する流れについて説明する。マイコンは、接続されたカメラやセンサからデータを取得する。その取得されたデータは、マイコンボードを識別するための情報 (MAC アドレス) と共に、Wi-Fi を通じてサーバへ送信される (図 2 ⑤)。サーバは、送信されてきたデータに付加された MAC アドレスからマイコンボードを識別し、データの蓄積を行う (図 2 ⑥)。

利用者は、コンピュータやスマートフォンなどのブラウザから、蓄積されたデータの確認が行える (図 2 ⑦)。また、その蓄積されたデータをダウンロードして Excel などのソフトウェアを用いて詳細な解析を行うこともできる。

3.4 制御対象マイコンとセンサ

制御対象とするマイコンには、中国 Espressif Systems 社が製造、販売する安価なマイコンである ESP8266 を選択した [4]。ESP8266 は Wi-Fi に接続でき、インターネットへの通信が行える。周辺装置も充実し、デジタル入出力や 10bit の AD コンバータ、I²C や SPI といった各種ペリフェラルも備えることから、様々なセンサや回路を接続することが可能である。プログラムの書き換えについても、一般的なシリアル通信による書き換えとともに、ファームウェアを準備することでネットワークを利用したプログラム書き換えに対応し、本教材システムが必要とする機能をすべて満たすものであった。

中学生向けの実践授業では、ESP8266 を搭載したマイコンボード「Espr Developer」[5] を利用した。Espr Developer 本体を図 3 に示す。このボードではマイコンの各入出力端子が引き出され、ジャンパーワイヤなどを用いてセンサなどとの接続が容易となっており、本教材で取り扱うセンサ、カメラ、LED などはずべてこの入出力端子に接続することで利用可能である。

本教材で対応するセンサ、カメラを表 1 に示す。対応するセンサの選定にあたっては、配線の本数、コスト、および入手性を考慮した。なお、対応するセンサについては、これから随時追加を行っていく予定である。

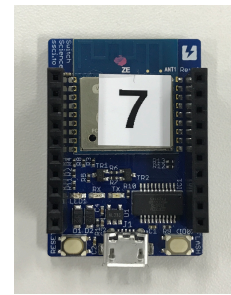


図 3 Espr Developer

表 1 選定したセンサ・カメラ一覧

種類	型番	参考価格
温湿度気圧センサ	BME280	¥1,080
水温センサ	DS18B20	¥1,404
シリアルカメラ	OV258	¥3,880

3.5 プログラミング言語

本教材のマイコンにプログラミングを行う言語として、ドリトルを選定した。ドリトルは、兼宗らが開発した教育用プログラミング言語である [6]。日本語による構文を特徴とし、プログラミング初学者への教育活動などに利用されている。

ドリトルを利用した計測・制御実習の実践例として、ロボット教材にプログラミングを行う活動 [7] や、Raspberry pi を用いた IoT 学習環境 [8] の報告がされている。これらの理由から、今回の教材におけるプログラミング言語として適切であると考えた。

4. 実装

4.1 ドリトルによるプログラミング

ESP8266 を搭載したマイコンボードのプログラム作成は、標準的なドリトルの構文にそって記述を行う。ドリトルはオブジェクト指向型のプログラム言語であることから、マイコンボードの機能などを管理するオブジェクト espr を新たに生成し、マイコンが実現できる機能を espr オブジェクトのメソッドとして割り当てた。これにより、センサを利用する際の様々な手続きを隠蔽し、命令の抽象化を実現した。使用者は、espr オブジェクトのメソッドを呼び出すだけで、マイコンの持つ様々な機能を利用することができる。

本教材を用いて、水槽の水温と魚の様子を水温センサとカメラで観測し、観測の結果をインターネット上のサーバに送信するための装置の外観とプログラムの例をそれぞれ、図 4 図 5 に示す。このプログラムは、カメラで撮影した写真と水温センサでの計測値をサーバに送信するプログラムである。

本教材のプログラムでは、ユーザはプログラムの先頭でライブラリの利用を宣言する。

システム! "espr" 使う。

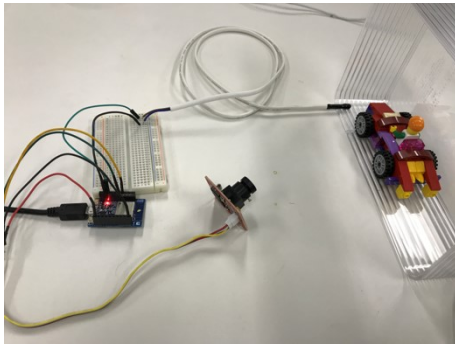


図 4 カメラと水温センサを接続したマイコン

表 2 命令一覧

命令	引数	機能
BME280	なし	BME280 センサの初期化
シリアルカメラ	なし	シリアルカメラの初期化
AD コンバータ	なし	AD コンバータの初期化
水温センサ	なし	水温センサの初期化
シリアル撮影	なし	撮影し、サーバに送信
温度?	なし	BME280 から温度を取得
湿度?	なし	BME280 から湿度を取得
気圧?	なし	BME280 から気圧を取得
水温?	なし	水温の取得
アナログ?	なし	AD コンバータの値を取得
待つ	ミリ秒	指定時間処理を停止
表示	センサ値, 文字列	確認画面に表示
ログ送信	センサ値	データ蓄積サーバへ送信
生成	なし	Arduino 言語への変換

```
システム! "espr" 使う。
最初に実行= 「
    espr!シリアルカメラ。
    espr!水温センサ。
」。
繰り返し実行= 「
    espr!シリアル撮影。
    espr!(espr!水温?) 表示。
    espr!10000 待つ。
」。
espr!生成。
```

図 5 プログラム例

これにより、マイコンの機能を抽象化した espr オブジェクトを利用することができるようになる。なお、マイコンの機能を利用する際には、espr オブジェクトを示す espr! を記述する。

次に、「最初に実行」ブロックに、センサやカメラの初期化命令を記述する。「最初に実行」ブロックは、マイコンが起動された際に最初の一度だけ実行が行われるブロックである。図 5 では、カメラと水温センサの初期化命令が記載されている。

そして、「繰り返し実行」ブロックに、マイコンが起動している間永続的に繰り返す命令を記載する。例えば、センサでの計測といったものである。図 5 では、カメラで撮影した後水温を計測、送信、そして 10,000 ミリ秒 (10 秒) 動作を停止する命令が記載されている。最後に、プログラムの変換を行う命令を記述する。

espr!生成。

その後、プログラムを実行することでドリトルから Arduino 言語への変換が行われる。

今回、対応した機能を示す命令の一覧を表 2 に示す。命令には、センサなどの初期化を行う命令、センサなどから値を取得する命令、計測値を Web 上で表示させたり、データ蓄積サーバ [9] への送信を行ったりする命令が存在する。

4.2 インターネットを介したプログラムの書き込み

ドリトル言語から生成された Arduino 言語のプログラム

をインターネット上にアップロードして、サーバ上でコンパイルを行い、インターネット経由でプログラムの書き換えが行える仕組みを構築した。この仕組みを用いてマイコンのプログラムを書き換える際の流れを以下に示す。

- (1) 変換されたプログラムをサーバにアップロードする
- (2) サーバ側でコンパイルが行われ、配置される
- (3) マイコンを起動する
- (4) マイコンのプログラムとサーバのプログラムのハッシュからプログラムの更新を確認する
- (5) 更新されていた場合、マイコンのプログラムが書き換わる

書き込みを行うマイコンの指定と更新の確認の際のマイコンの特定には、マイコンごとに固有に設定されている MAC アドレスを利用する。そのため、サーバに予め利用するマイコンの MAC アドレスと対応する名前をサーバに登録するようにした。マイコンは、起動するごとにサーバに自身のプログラムのハッシュ値と MAC アドレスの送信を行う。サーバ側では、送信されたデータをもとにプログラムの更新を確認し、更新されていれば新しいプログラムをダウンロードさせる。プログラムのアップロード画面を図 6 に示す。生徒は、自身の利用しているマイコンボードに割り当てられた名前を選択し、変換されたプログラムのアップロードを行う。

4.3 計測値確認ページ

生徒が作成されたプログラムが正常に動作しているかを確認するために、最新のデータをブラウザから確認する計測値確認ページを作成した。確認ページのキャプチャを図 7 に示す。

このページには、ドリトルでの「espr!表示」命令でセンサの計測値を送信したり、「espr!シリアル撮影」命令で画像を送信したりしたものがブラウザで確認でき、利用



図 6 プログラムアップロード画面

者は、ページにアクセスして自身のマイコンボードの名前を入力することで、そのマイコンボードから送信された画像や計測値の確認が行える。また、このページの右下には画像と計測値のそれぞれがサーバ上でいつ更新されたかを表示するようにした。そうすることで、データが正常に送信されているかといったことや、プログラムが正常に動作しているかなどの死活監視が行える。

アップロードされた画像/計測値がどのマイコンボードから送信されたものかの判別には、前節のプログラムの書き込みと同様にサーバに登録されている MAC アドレスをもとに行う。そのため、マイコンは画像や計測値を送信すると同時に、MAC アドレスをサーバへ送信する。

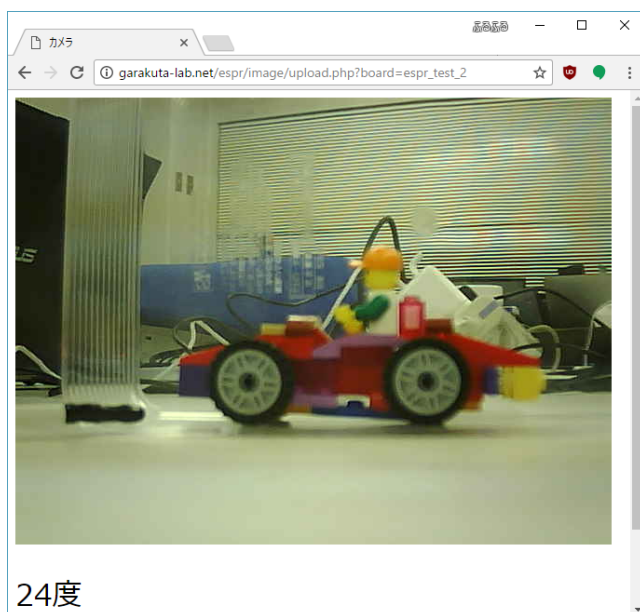


図 7 確認ページのキャプチャ

5. 中学校での実践

5.1 概要

静岡県の中学校において、本教材を利用した。対象は中学3年の4クラス、約120人である。

生徒は2人一組の班となり、センサやカメラをマイコンボードに接続する実習や、教室内の写真を撮影したり、セ

ンサでの気温計測を行うプログラムを作成したりする実習を行った。

5.2 授業の流れ

本教材を使うにあたっての入門として、温湿度気圧センサとカメラを生徒に使わせた。その際の授業の流れは次のようなものであった。

- (1) ボードやセンサのキットを班ごとに配布する
- (2) プリントをもとに、サンプルと同等のプログラムを制作する
- (3) プリントをもとに、マイコンボードにセンサやカメラの配線を行う
- (4) 動作の確認をする
- (5) プログラムの改良を行う

なお、(2)と(3)については、班のメンバーで作業の分担を行い、(5)については余裕のある生徒のみが行った。

配布された、プリントには、センサの実体配線図と図8の温湿度気圧センサから気温を計測し、計測値閲覧ページに反映させるプログラムが記載されていた。このプログラ

```
システム!"espr" 使う。  
最初に実行=「  
    espr!シリアルカメラ。  
    espr!BME280。  
」。  
繰り返し実行=「  
    espr!シリアル撮影。  
    espr!(espr!温度?) 表示。  
    espr!10000 待つ。  
」。  
espr!生成。
```

図 8 プログラム例

ムは、表示する際の単位の部分や文言(単に「一度」と出すだけでなく、「現在一度です。」とするなど)の部分や計測を行う間隔などいくつか改良の余地が残されている。

実際に温湿度気圧センサに配線する生徒の様子を図9に示す。生徒らは、配線のミスや、プログラムのミスなどで

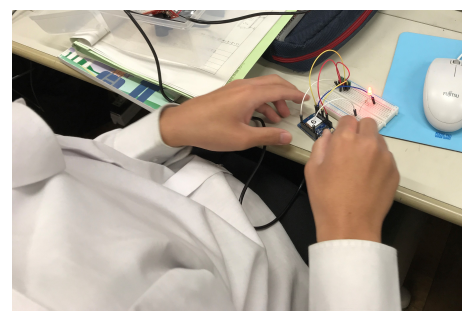


図 9 温湿度気圧センサへの配線を行う生徒

うまく動かない場合には、それぞれが教え合うことで、解

決を試みる姿もみられた。また、記載されたプログラムを改変することにより、センサの値を表示する文言を変えている生徒も存在した。例えば、図 10 の 2 名の生徒は、単位のあとに「!」を追加して個性を出していたり、湿度を計測するプログラムを追加して気温と湿度を計測するようになりしていた。また、完成が早かった生徒には、湿度

たつては、電源投入直後には電源を抜かないことを周知したり、ファームウェアを修復するためのツールを構築して提供したりすることで対応したが、広く使ってもらうためには容易にファームウェアが壊れないようにする必要がある。原因として、ファームウェアと生徒の作成したプログラムが一体であり、プログラムが書き換わる際に電源が切れることで正常な書き込みが行われず、動作しなくなるものと考えている。そのため、ファームウェア部分と生徒の作成したプログラムを分離し、突然電源が切れた際でもファームウェアに及ぶ影響を最小限に抑える必要があると考えている。

6. おわりに

Wi-Fi に接続可能な安価なマイコンである ESP8266 にドリトルを用いてプログラミングを行い、センサやカメラを利用した IoT 教材を開発した。また、学校で利用しやすいクラウド上のサーバを利用したプログラムのコンパイルと書き込みの仕組みを構築した。これにより、中学生が実際にマイコンボードにセンサなどを配線し回路を作成する実習や、計測値をサーバへ送信するプログラムを作成する実習が行えるようになった。

これから、課題点について解決し、広く使ってもらうためにパッケージ化して公開したいと考えている。

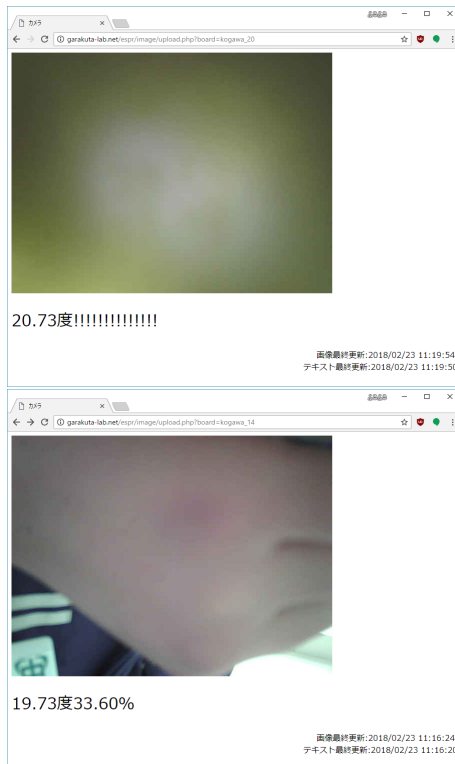


図 10 生徒の作成した計測値閲覧ページ

や気圧を計測する方法を提示し、温度、湿度、気圧が全て測れるプログラムに改良を行わせた。

5.3 課題

実際に利用すると、いくつかの課題が浮上した。まず 1 点目は、サーバ上でコンパイルを行う際の時間が長いという点である。具体的には、1 回のコンパイルにおよそ 1 分の時間がかかっている。また、複数の生徒が同時にコンパイルを行おうとした際には、1 分よりも更に長い時間がかかってしまい、授業で利用するにあたって生徒が試行錯誤を繰り返す際の障壁となってしまっている。原因として、変換後のプログラムが必要とするライブラリ類を含めたすべてのプログラムを毎回コンパイルしているためであると考えている。そのため、共通部分については予めコンパイルしておき、リンクを行うようにすることでコンパイル時間を短くできると考えている。

2 点目は、起動直後に不意に電源が切れた際に、マイコンに書き込まれたファームウェアが壊れてしまい、正常に動作しなくなる点である。今回授業で利用してもらうにあ

参考文献

- [1] 総務省: 情報通信白書平成 27 年度版, <http://www.soumu.go.jp/johotsusintokei/whitepaper/h27.html> (2017/11/1 確認)
- [2] フィリップスが提供するワイヤレスでスマートな照明 — Hue の紹介. <https://www2.meethue.com/ja-jp> (2018/5/16 閲覧)
- [3] Dash Replenishment Service - Amazon Developer. <https://developer.amazon.com/dash-replenishment-service> (2018/5/16 閲覧)
- [4] ESP-WROOM-02 Overview — Espressif Systems. <https://www.espressif.com/en/products/hardware/esp-wroom-02/overview> (2018/5/9 閲覧)
- [5] ESPr@Developer (ESP-WROOM-02 開発ボード) - スイッチサイエンス <https://www.switch-science.com/catalog/2500/> (2018/5/9 閲覧)
- [6] 兼宗進, 御手洗理恵, 中谷多哉子, 福井真吾, 久野靖: 学校教育用オブジェクト指向言語「ドリトル」の設計と実装. 情報処理学会トランザクション「プログラミング」 Vol.42, No.11, pp.78-90 (2001)
- [7] 佐々木寛, 奥本拓哉, 島袋舞子, 大村基将, 兼宗進: 高等学校共通教科情報における「ドリトル」×「ロボティスト (スタディーノ)」を利用した授業の実践報告. 研究報告コンピュータと教育 Vol.2016-CE-161, No.19, pp.1-9 (2016)
- [8] 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)
- [9] 林康平, 西川弘恭, 小林史弥, 間辺広樹, 大村基将, 兼宗進: ラズベリーパイを用いたドリトルでのデータ計測とデータ蓄積サーバの提案 Vol.2015-CE-131, No.3 pp.1-7 (2015)