

ブロックでプログラミングできる IoT 学習環境の改良

渡部敬太^{*1}, 中西通雄^{*2}

^{*1} 大阪工業大学大学院, ^{*2} 追手門学院大学

Improvement of IoT learning environment that provides block programming

Keita Watanabe^{*1}, Michio Nakanishi^{*2}

^{*1} Osaka Institute of Technology, ^{*2} Otemon Gakuin University

マイコン基板 IchigoLatte を用いて, IoT (Internet of Things) を学ばせる教材を開発している. この教材ではパソコン上でブロックを組み合わせてプログラムを作り, マイコン基板に書き込む. 従来の教材は, マイコン基板で取得したセンサ値データをインターネットを介してサーバに送信し, サーバに蓄積されたデータをスマートフォンで観測するものであった. 今回, スマートフォンからインターネットを介してマイコン基板に対して指令を送れるようにした. これにより, IoT についてより理解を深めることができると考える. また, ブロックとそれに対応するソースコードを同じ色で表示する機能や, よりソースコードに近い表記のブロックを使用できる機能を追加した. この改良により, ブロックプログラミングからソースコードプログラミングへ移行しやすくなると考える.

キーワード: プログラミング, IoT 学習, IchigoLatte, MixJuice, Blockly

1. はじめに

本研究では, 小学 5 年生以上を対象として, プログラミングの入門教育と IoT (Internet of Things) の入門教育を目的とする. IoT とは, モノがインターネット経由で通信することで, センサを用いた計測やマイコン基板の制御等を行うことである. 近年, IoT を基盤としたサービスは増加している. IoT の使用例のイメージを図 1 に示す. 図 1 は, ビニールハウスで温度や湿度を計測し, インターネットを介してスマートフォンで観測する. 観測結果から散水するかを決め, スマートフォンからインターネットを介して指令を行う.

IoT システムを理解することで, 身近にある IoT 製品に対する見方が変わり, 視野が広がる (物事を様々な角度から考える力がつく) と考える. また, プログラミングの基礎とともに学ぶことで, IoT 製品に限らず様々な製品は全てプログラムで動いていることが理解できるであろう. 筆者らは以前に, マイコン基板 IchigoLatte を用いた IoT 学習環境を開発した. マイコン基板で測定したデータをインターネット経由でサーバコンピュータに送り, サーバコンピュータに蓄積されたデータをグラフに変換して観測することで, IoT のしくみについて学習できる⁽¹⁾. 本研究では, IoT

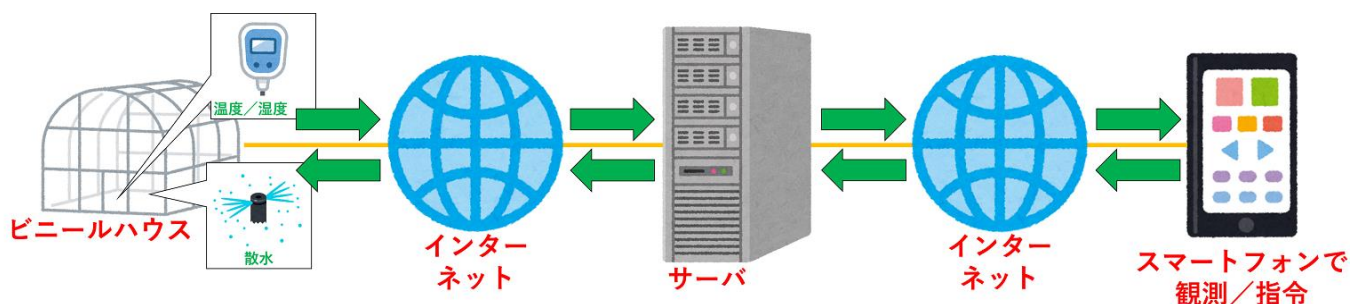


図 1 IoT の使用例のイメージ

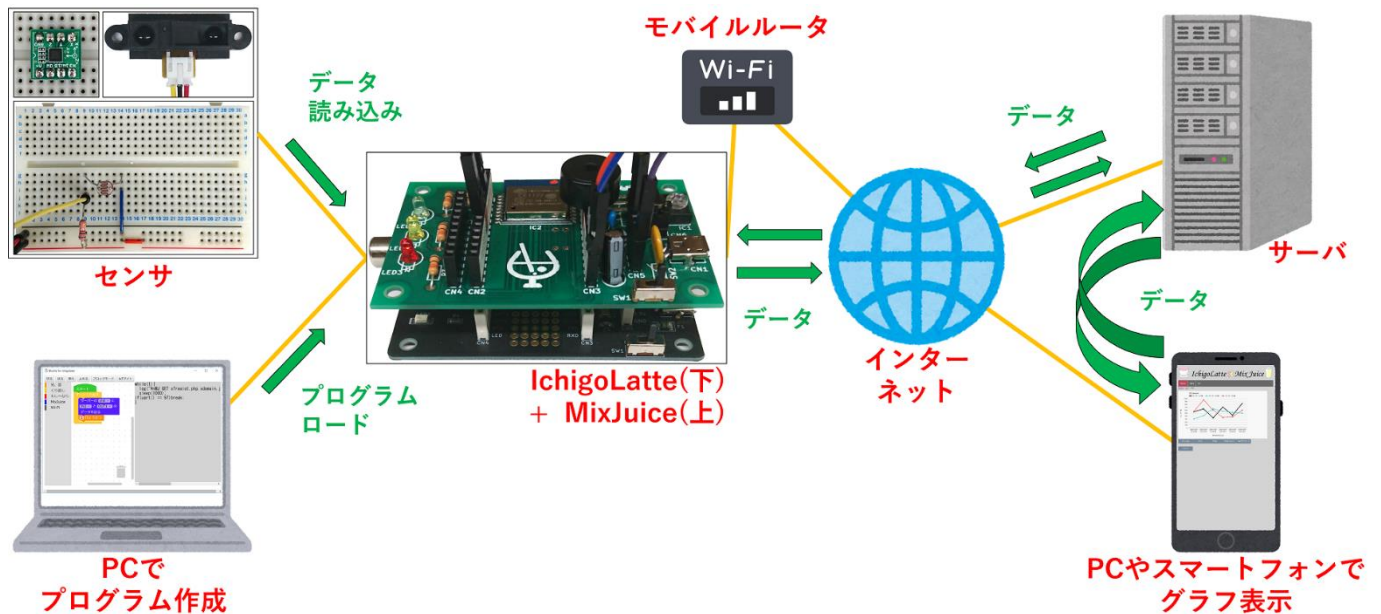


図 2 システム構成

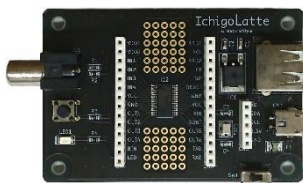


図 3 マイコン基板 IchigoLatte

学習環境に使用する IchigoLatte 用ツールと IoT サイトの改良を行った。

2. システム構成

本研究の IoT 学習環境の構成を図 2 に示す。まず、マイコン基板 IchigoLatte, WiFi 通信を行う MixJuice, センサ類を PC に USB シリアル変換モジュールで接続する。

2.1 IchigoLatte と MixJuice

IchigoLatte (図 3) には OS が搭載されており、JavaScript をベースにした MiniScript でプログラミングできる⁽⁵⁾。IchigoLatte には LED が 1 つ付いている。MixJuice はインターネットに接続するための無線 LAN 基板である。IchigoLatte, MixJuice 両基板には、スピーカやセンサ等を付けることができる。

2.2 サーバ

レンタルサーバである XFREE を利用して、サーバを用意した。サーバは、チャンネル番号とセンサ値データを受信すると、チャンネル番号に対応したテキス

トファイルにセンサ値データを書き込む。チャンネルとは 12 個のグラフから選択するための数字である。本研究では、IoT サイトのスイッチ (図 9) が ON か OFF によってテキストファイルに 1 か 0 を書き込む機能を追加した。テキストファイルに書き込みを行うソースコードは PHP で約 500 行である。

2.3 IchigoLatte 用ツール

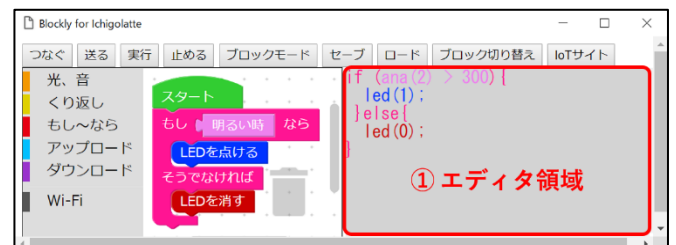


図 4 IchigoLatte 用ツール

IchigoLatte 用ツール⁽³⁾では、PC 画面上で日本語表記のブロックを組み合わせてプログラムを作成できる (図 4)。このツールを使用することで、プログラミングを学んだことがなくても簡単にプログラムを作成できる。ブロックで作成したプログラムは MiniScript のコードに変換されて、図 4 の①のエディタ領域に表示される。また、エディタ領域のソースコードを直接編集することができる。しかし、ソースコードの内容は、ブロックには反映されない。従来の IchigoLatte 用ツールは、html が約 70 行で、JavaScript が約 820 行である。そして、本研究で新しく追加したコードは、html が約 160 行、JavaScript が約 1200 行である。

2.4 IoT サイト

IoT サイトとは、IoT 学習環境として開発した Web サイトでサーバ上に用意した。サーバに蓄積されたセンサ値データは、グラフ化して表示することができる。グラフは 12 個の中から選択でき、12 個のチャンネル番号(ch0 から ch11)から選ぶことで切り替えられる。

3. システムの流れ

PC 上でプログラムを作成して、IchigoLatte で実行する例を図 5 に示す。

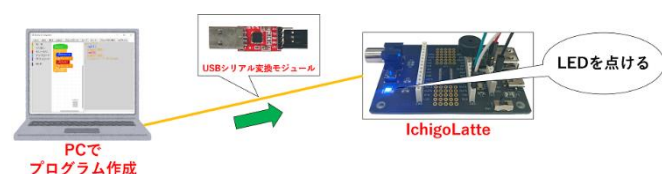


図 5 PC から IchigoLatte を実行する例

データ送信からグラフ表示については、図 6 の流れで以下の 1 から 4 の順に操作を行う。

1. 学習者は、PC 上でブロックを組み合わせてプログラムを作成し、IchigoLatte にロードする (図 5)。
2. IchigoLatte はロードしたプログラムを UART (Universal Asynchronous Receiver Transmitter) 経由で MixJuice へと渡し、プログラムに従って実行する。
3. IchigoLatte は、センサで取得した数値データを、MixJuice からインターネットを介してサーバへと送信する (図 6)。
4. 学習者は、スマートフォンで IoT サイトを開き、サーバに蓄積された数値データをグラフに変換した形で観測することで、データの可視化、分析を行える (図 6)。

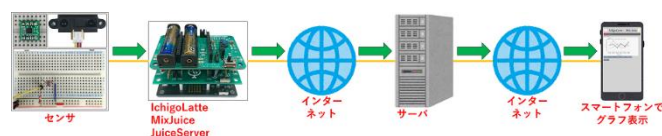


図 6 データ送信からグラフ表示までの流れ

ここまでが、筆者らが以前に作成した IoT 学習環境である。

本研究で拡張したデータの受信については、図 7 の流れで以下の 1 から 3 の順に操作を行う。

1. 学習者は、PC 上でブロックを組み合わせて、プログラムを作成し、IchigoLatte にロードする (図 5)。
このプログラムでは、1 秒毎にサーバからテキストファイルのデータを受信し、受信したデータによって LED や音の制御を行う (図 7)。
2. IchigoLatte はロードしたプログラムを UART 経由で MixJuice へと渡し、プログラムに従って実行する。
3. 学習者は、スマートフォンで IoT サイトを開き、IoT を制御するボタンをクリックすることで、IchigoLatte の LED の点灯や消灯、音を鳴らす等の制御を行う。

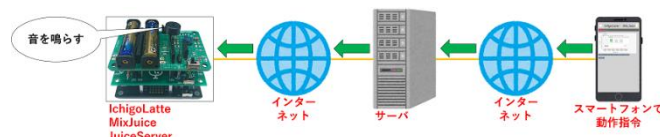


図 7 スマートフォンから IchigoLatte を操作する流れ

4. プログラミング教室の実施

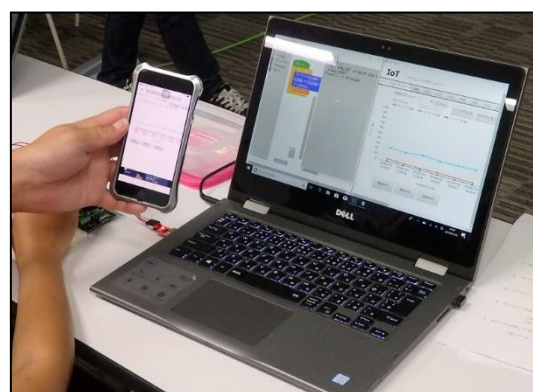


図 8 プログラミング教室の写真

4.1 プログラミング教室の実施内容

2019 年 7 月 20 日に小学 3 年生 1 人と 5 年生 2 人を対象としてプログラミング教室を実施した (図 8)。約 110 分でプログラミングの基礎、残りの約 50 分で IoT 入門の授業を行った。また、このとき使用した IoT 学習環境は筆者らが以前に作成したものである。授業では約 100 枚のスライドを用いて、以下の 1 から 4 の順番で説明を行った。

1. 順次処理：スピーカを用いて、複数の音を順番に鳴らすプログラムの作成
2. 繰り返し処理：LED の点灯と消灯を交互にくり

返し行うプログラムの作成

3. 条件分岐処理：赤外線センサを用いて、センサの値によって LED の点灯、消灯を行うプログラムの作成
4. IoT（データ送信のみ）：照度センサと赤外線センサの値をサーバに送信するプログラムの作成とサーバに送られたセンサ値データの観測

4.2 プログラミング教室実施に対する考察

授業実施後に筆記形式のアンケートと理解度テストを実施した。アンケート結果では、ケーブルのつながりが難しかったという意見が多かった。これは、パソコンと IchigoLatte を USB シリアル変換モジュールでつなげるとき、ジャンパワイヤが 4 本あり、つながりが複雑であったことが原因と考えられる。

この解決策として、JuiceServer を使用することが挙げられる。JuiceServer とは IchigoLatte に使用できるバッテリー用基板である。これを使用することで、USB シリアル変換モジュールをつなぐ際、ジャンパワイヤが 2 本で済み、つながり方もより簡単になる。また、プログラミングと IoT に対する理解度テストについて、表 1 に示す。

表 1 プログラミングと IoT に対する理解度テスト

項目	問題	正解者 (3 人中)
プログラミング	プログラムが実行する順番を答える問	3 人
IoT	マイコン基板、パソコン、インターネット等がどのようにデータをやり取りしているかを線でつなぐ問	1 人

表 1 から、モノとインターネットが通信するという IoT の基本が理解できていないことが推測できる。これは、プログラミング教室実施時に、グラフ表示をスマートフォンだけでなく、プログラミングを行う PC でも行ってしまったことが原因であると考えられる。つまり、PC と IchigoLatte を USB シリアル変換モジュールで接続していたので、センサ値を直接 PC に読み込んでグラフ表示していると誤解したと考える。

この問題の解決策に対しても、JuiceServer を用いることで解決できると考える。JuiceServer を接続すると PC とマイコン基板を切り離れた状態で実行できるからである。

5. IchigoLatte 用ツールと IoT サイトの機能拡張

5.1 IoT サイトのスイッチ追加とスイッチデータ受信用ブロック追加

本研究で追加した IoT サイトの Web 画面上のスイッチを図 9 に示す。

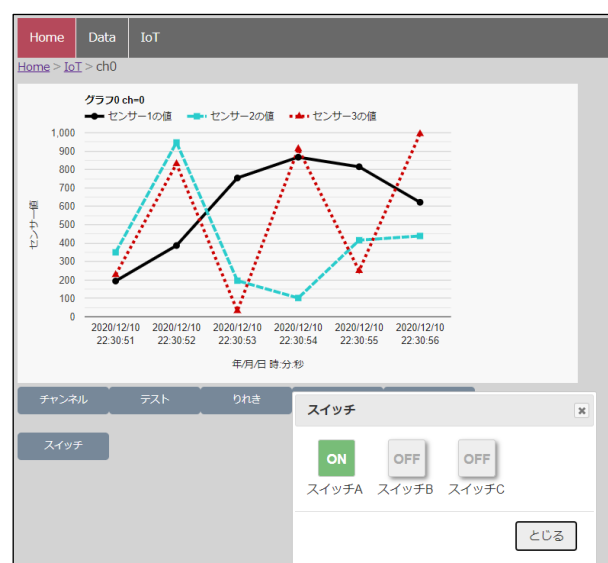


図 9 IoT サイトのスイッチ

スイッチには A, B, C の 3 種類がある。この 3 種類のスイッチは、IoT サイトの 12 個のグラフそれぞれにあるため、データ受信時にチャンネル番号 (ch0～ch11) を選択する必要がある。

本研究で追加した 3 つのスイッチデータ受信用ブロックを図 10 に示す。

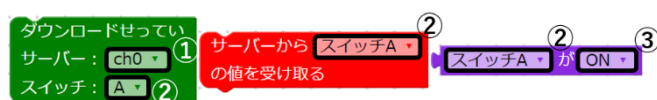


図 10 スイッチデータ受信用ブロック

左がスイッチデータを受信する準備を行うためのブロック、中央がサーバからスイッチデータを受信するブロック、右がスイッチの ON, OFF を判定するブロックである。図 10 の①は ch0 から ch11 の 12 種類から選択する。図 10 の②は、スイッチ A, スイッチ B, スイッチ C から 3 つから選択する。図 10 の③は ON

か OFF を選択する。

このブロックを使用して、スイッチデータを受信する場合、MixJuice に接続されている USB シリアル変換モジュール (IchigoLatte が PC と通信するためのモジュール) を外しておく必要がある。これは、スイッチデータを受信するために UART を使うためである (USB シリアル変換モジュールを接続した場合、データ待ち状態になり、UART が-1 になるため、UART が使用できなくなる)。

USB シリアル変換モジュールを外すと、電源供給が断たれ、PC 画面 (IchigoLatte 用ツール) からの実行ができなくなる。電源供給のため、JuiceServer を接続し、IchigoLatte 上のボタンを押しながら、電源スイッチを入れて、保存されているプログラムを実行する。これにより、PC とマイコン基板がつながっていない状態になるため、プログラミング教室で問題となっていた PC とマイコン基板のみで通信しているという誤解がなくなり、インターネット経由で通信しているということが理解しやすくなると考える。

5.2 データ保存／読み込み機能の追加

従来の IchigoLatte 用ツールでは、作成中のプログラムを中断して後から再開したいときには、ソースコードをテキストエディタにコピーしなければならないという問題点があった。これに加えて、保存できるのはソースコードだけでブロックの保存はできないという問題点もあった。今回、ブロックとソースコードを保存する機能と保存したデータを読み込む機能を追加した。これにより、テキストエディタを使う手間をなくすことができた。

6. ブロックプログラミングからソースコードプログラミングへの移行

IchigoLatte 用ツールでは、ブロックを配置するとエディタ領域に、ブロックに対応したソースコードを表示する。これにより、ブロックプログラミングからソースコードプログラミングへの橋渡しができるように設計した⁽³⁾。しかし、これだけではブロックでのプログラミングからソースコードでのプログラミングへスムーズに移行できないと考えており、今回、よりスムーズに移行することを目的として IchigoLatte 用ツールの改良を行った。

6.1 ブロックとソースコードの対応

ブロックの色とブロックに対応するソースコードの色を同じにする機能を追加した。従来の IchigoLatte 用ツールでは、図 11 の右上のように複数のブロックを並べた後、どのブロックがどのソースコードと対応しているかがわかりにくいという問題があった。そこで、図 11 の右下のように、ブロックの色とブロックに対応するソースコードの色を同じにした。これにより、ソースコードとの対応がわかりやすくなるので、ソースコードの意味を把握しやすくなると考える。

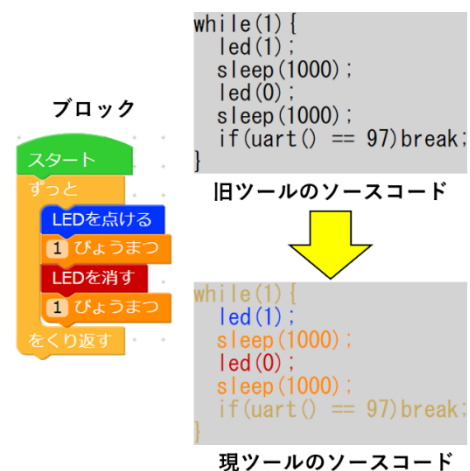


図 11 旧ツールと現ツールのソースコードの比較

6.2 ブロック切り替え機能の追加

ソースコードは英語ベースの表記であるため、従来の IchigoLatte 用ツールのブロックは、プログラミング初学者でも理解しやすくするために、日本語表記で記述している。しかし、プログラミングに慣れてきた学習者がソースコードでのプログラミングに移行するとなったとき、スムーズに移行できないのではないかと考える。



図 12 音を鳴らすブロックとソースコード

例えば図 12 のように実際のソースコードが、250Hz で 500m 秒音を鳴らすという beep(250, 500) のとき、ブロックは音を出すという表記になっている場合である。これだと、ソースコードの意味が理解しにくい。そこで、ソースコードに少し近づけたブロックを追加

した．追加したブロックの一部を表 2 に示す．プログラミングに慣れてきた人用のブロックは，IchigoLatte 用ツールのブロック切り替えボタン（図 4 の上部のボタン）で使うことが可能である．このブロック追加により，ブロックプログラミングからソースコードプログラミングへ移行していくときの助けになると考える．

表 2 プログラミング初学者用のブロックとプログラミングに慣れてきた人用のブロックの比較

プログラミング初学者用のブロック	プログラミングに慣れてきた人用のブロック
	
	

7. おわりに

本研究では IchigoLatte 用ツールと IoT サイトを改良し，より IoT について理解を深める教材にした．さらに，ブロックプログラミングからソースコードプログラミングへよりスムーズに移行できるように改良を行った．しかし，コロナ禍により改良した教材を用いて学習者に授業できていない．今後プログラミング教室を実施して評価をとり，それに基づいた改良を行っていくことが課題である．

謝辞

本研究の一部は，JSPS 科研費 JP19K12281 の助成を受けた．

参 考 文 献

- (1) 渡部敬太，中西通雄：“IchigoLatte 用ツールのブロック追加とデータを可視化するアプリの機能拡張”，PC カンファレンス，pp.63-66 (2019 年 8 月)
- (2) 渡部敬太，中西通雄：“IchigoLatte を用いてセンサ値データをサーバに送るプログラムとセンサ値データを可視化する Web アプリの開発”，教育システム情報学会

2018 年度学生研究発表会，pp.139-140 (2019 年 2 月)

- (3) 中南拓也，中西通雄：“ブロック組み立てによるプログラミングからキーボード入力によるプログラミングへの橋渡しをする IchigoLatte を用いた教材”，教育システム情報学会，pp.137-138 (2019 年 2 月)
- (4) 鴻池泰元，中西通雄：“IchigoJam 用ビジュアルブロックプログラミング環境の開発とプログラミング体験教室の実践”，情報処理学会コンピュータと教育研究会 144 回研究発表会，pp.67-70 (2018 年 3 月)
- (5) 古旗一浩，松田優一：“みんなの IchigoLatte 入門 JavaScript で楽しむゲーム作りと電子工作”，リックテレコム (2017 年 4 月)