

中学校技術科「計測・制御」関連単元向け STEAM 指向教材の 授業での利用を意識した改良

－コンパイル時間の短縮と計測結果の蓄積機能の追加－

Improving STEAM Teaching Materials for 'Measurement and Control' in Junior High School Technology Classes

- Reduction of Compilation Time and Addition of Data Storage Functions -

加葉田 悠斗^{*1}, 各務 正太郎^{*1}, 永井 孝^{*2,3}, 館 伸幸^{*2}, 香山 瑞恵^{*2}
Yuto KABATA^{*1}, Seitaro KAKUMU^{*1}, Takashi NAGAI^{*2,3}, Nobuyuki TACHI^{*2}, Mizue KAYAMA^{*2}

^{*1}信州大学大学院総合理工学研究科

^{*2}Graduate School of Science & Technology, Shinshu University

^{*2}信州大学工学部

^{*1}Shinshu University, Faculty of Engineering

^{*3}ものづくり大学技能工芸学部

^{*3}Institute of Technologists, Faculty of Technologists

Email: 25w6025e@shinshu-u.ac.jp

あらまし：本研究の目的は、中学校技術科の「計測・制御」単元に則した教材の開発である。これまでの研究では、センサによって計測したデータに応じてアクチュエータを制御するプログラミング学習のフレームワークを提案してきた。本研究では、授業での利用を実現するため、コンパイルサーバによるコンパイル時間の短縮、および計測データのデータベースへの蓄積と再利用を実現した。

キーワード：中学校技術科, 計測・制御, STEAM, 植物工場, コンパイル

1. はじめに

内閣府の Society5.0 では、IoT と AI により人とモノがつながり、社会課題の解決や新たな価値創造が可能になると示唆されている^(1,2)。これに伴い、中学校技術・家庭科の「情報の技術」単元⁽³⁾も強化され、「プログラムによる計測・制御」(以下、「計測・制御」)では、生活支援ロボットのモデル化などが例示されている。しかし、プログラミング初心者にとってロボットのモデル化や制御プログラムの作成は難しい課題である。

本研究では、プログラミング言語を習得せずに生活支援ロボット用プログラムを作成できる環境を構築し、学習者が「計測・制御」の学習に取り組みやすくすることを目的とする。具体的には、先行研究⁽⁴⁾を基に、モデルから実行コードへのコンパイル時間の短縮と、センサーデータの蓄積・再利用機能を開発する。本稿では、先行研究の課題と改善点を示

し、新機能を備えた教材の設計と使用例を紹介する。

2. 先行教材の概要と問題点

先行研究では、中学技術の計測・制御向け STEAM 教材プロトタイプが提案された。図 1 に STEAM 教材プロトタイプの構成を示す。教材は運用部と開発部で構成され、運用部ではセンサ①で取得したデータが計測用デバイス②を通じて Firebase③に送信され、制御用デバイス④に転送後、アクチュエータ⑤の動作制御に利用される。開発部では、モデリング学習環境⑥で作成したプログラムを、書き込みツール⑦を用いて④に書き込み、学習者は独自の計測・制御システムを構築する。

先行教材の課題は以下の通りである。

- 開発部で、⑥から④用の実行コード生成に時間がかかること。
 - 運用部で、①の計測結果を可視化できないこと。
- コンパイル速度の改善は、ソフトウェアの時間効率性として検証する。計測結果の可視化は、学習者による制御機能の適用場面の多様性で評価する。

3. コンパイル時間の短縮

コンパイル時間を短縮するために、コンパイルサーバの Docker 化を図る。先行教材でのコンパイル環境は、Raspberry Pi 4ModelB 内の Node.js 上で Platform IO CLI(以降 PIO)を使用していた。この環境の課題は以下の 3 点である。

- PIO でのコンパイルが逐次処理であること
- PIO での初回コンパイル時間が、2 回目以降よ

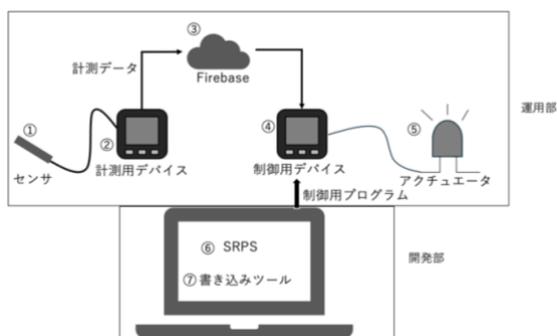


図 1 STEAM 教材プロトタイプの構成

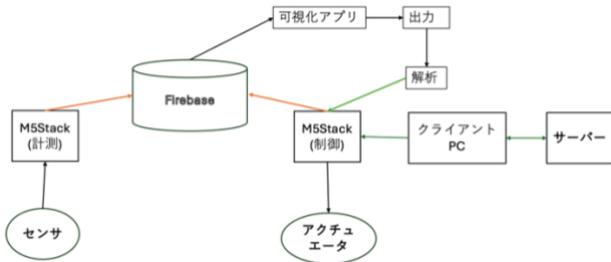


図2 提案教材のシステム概念図

りも必ず長くなること

- 1ファイルあたりコンパイルに3分要すること
これらの課題を解決するために、以下の二つの機能を、Dockerを用いて実装する。
- ファイルを並列にPIOでコンパイルする
- PIO起動時にダミーファイルをコンパイルする
その結果、10ファイルをコンパイルする場面において、提案環境は従来環境と比べ、時間効率性としてコンパイル処理が1ファイルあたり約100秒短く、総コンパイル時間は約1500秒短くなった。

4. 計測データの可視化

計測結果を可視化するため、計測データをFirebase③に永続的に蓄積し、可視化アプリケーション(以下、可視化アプリ)で③から取得したデータを可視化することとした。この可視化アプリでは、④のデバイスIDを参照することで、センサ①で計測した温度や湿度、酸素濃度・二酸化炭素濃度等の計測データを時系列グラフとして可視化し、CSVファイルとしてダウンロードできる。

新規機能を組込んだ提案教材のシステム概念図を図2に示す。計測結果を可視化アプリでグラフとして確認できる。提案教材に新規に組み込んだ可視化機能の利用者3名に制御プログラムを考えさせた。その結果、全員が時間とセンサ値を組み合わせたアイデアを提案した。可視化機能を用いる前は、センサ値のみを利用するアイデアであったのに対して、利用後は時間要素を加味したアイデアに変化したことは、制御機能適用場面の多様性が増したと考える。

5. 新機能を有する提案教材の利用例

新機能の可視化効果を検証するため、しいたけ栽培キットを用いて連続的な温度データを取得した。しいたけを選んだ理由は、成長が早く、温度管理のみで育成可能なためである。2024年12月18日から育成を開始し、25日から27日に収穫を行った。図1の温度センサ①、計測用デバイス②、Firebase③、可視化アプリを使用し、19日から27日までの温度推移を記録・可視化した(図3参照)。栽培中の様子は22日に撮影した(図4参照)。

成長期(18日~24日)と収穫期(25日~27日)の可視化結果を3名の被験者に提示し、制御プログラムに関する以下のアイデアを得た。

- 30分前の湿度が90%以上ならばガラリを開け

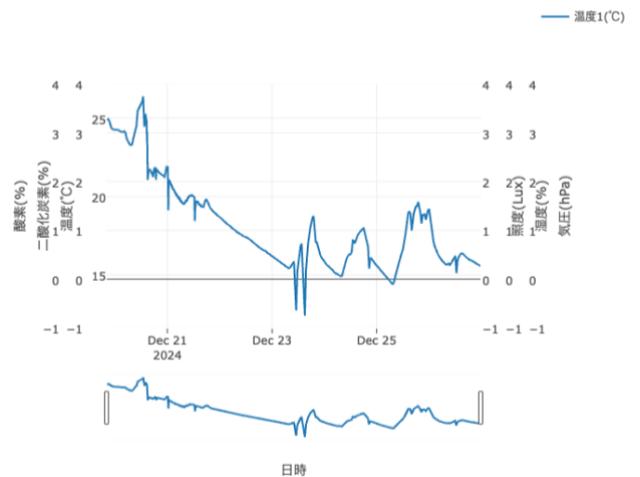


図3 可視化アプリのインターフェース



図4 栽培対象

る

- 1時間前から最新の土壌湿度の平均が50%未満ならば水やりをする
- 2時間前の平均温度が20°C未満かつ平均照度が100lux未満ならばLEDを点灯する

6. おわりに

本稿では、中学技術の計測・制御向けSTEAM教材プロトタイプの問題に対する改善方針を示し、新規機能を設計・実装し、それらの使用例を示した。今後は、中学技術での授業実現のために、植物工場を模した学習シナリオが「計測・制御」で使用可能かを調査する。そして、中学生が可視化アプリからしきい値を容易に決める支援機能や、中学生が時系列データを活用する支援方法を検討する。

参考文献

- (1) 内閣府, "Society5.0 - 科学技術政策", https://www8.cao.go.jp/cstp/society5_0/, (2025/2/3 確認).
- (2) 総務省, "平成30年版 情報通信白書 | Society5.0", <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd102300.html>, (2025/2/3 確認).
- (3) 文科省, "中学校学習指導要領(平成29年告示) 解説技術・家庭編", https://www.mext.go.jp/content/20230516-mxt_kyoikujinzai02-000033059_04.pdf, (2025/2/3 確認).
- (4) 林竜希, 豊嶋宏太, 清水峻司他, "STEAM教育用IoT教材とモデル駆動開発方法論を組み合わせた中学校技術科「計測・制御」関連単元向けフレームワークの提案", 情処研報 CE, 2024, 2024(8), 1-8.