

IoT サーバに蓄積されたプログラミング履歴から学習者の試行過程の追跡を実現した協調学習用メカトロニクス教材の開発

Development of Mechatronics Teaching Materials with Tracking of Learner's Trial Process based on Programming History Stored in IoT Server

千田 和範^{*1}, 稲守 栄^{*1}
Kazunori CHIDA^{*1}, Sakae INAMORI^{*1}

^{*1} 釧路工業高等専門学校

^{*1} National Institute of Technology, Kushiro College
Email: chida@kushiro-ct.ac.jp

あらまし： Society5.0 時代に求められる人材として、様々な情報を活用できる技術者が求められている。その人材育成の一つとして、メカトロニクス・ロボットを用いた課題探求型教育が注目されているが、試行錯誤過程を考慮した学生指導は難しいものがあつた。しかし最近 IoT 技術等を活用しプログラミング履歴を蓄積することで、学習効果を上げる事例が表れている。そこで本研究では制御コードの時系列変化の追跡・活用することで各学習者に適した指導の実現を目指したメカトロニクス学習システム開発する。
キーワード：IoT, 問題解決型学習, 学習履歴, メカトロニクス教材

1. はじめに

最近, Society5.0 時代に求められる人材育成に向けて、「主体的・対話的で深い学び」を目指した教育が推進されている。しかし、この教育の一つである問題解決型協働学習に取り組む学習者が、どのように知識や技術を深めていくのかに注視している例は少なく、効果的に協働学習を進める観点が不足しているといわざるを得ない。一方で、文科省提言にもある通り、個人の学習状況をプログラミング履歴として蓄積し、そのログを個の能力に応じて活用する新しい指導方法の取り組みが加速化しつつある。

そこで本研究では、小中高校生や高専低学年対象のグループワーク用メカトロニクス教材において、プログラミング履歴として学習者が記述したメカトロニクス教材のプログラムである制御コードを逐次データベースサーバに蓄積できるシステムを開発する。この制御コードの時系列変化の追跡・活用により、各学習者に適した効果的な指導が期待できる。

2. 学習ログを蓄積可能なメカトロニクス学習システム

2.1 基本となるメカトロニクス教材

開発する協調学習用メカトロニクス学習システムにおいて、ベースとなるロボット台車を図1に示す。実習では複数のロボットが協力して与えられた課題を達成することを目指す。本システムは制御コードを作成する PC と制御コードを遠隔転送するための送信用 micro:bit, そしてコード受信と機器制御を行う micro:bit, そして受信・制御側 micro:bit により制御される台車によって構成される。次に台車部分のハードウェア構成は制御用マイコンとして micro:bit v2.0 を搭載し車輪駆動機構としてステッピングモータを2台搭載している。またセンサー系として衝突検出用接触センサーの他、測距用超音波センサーが

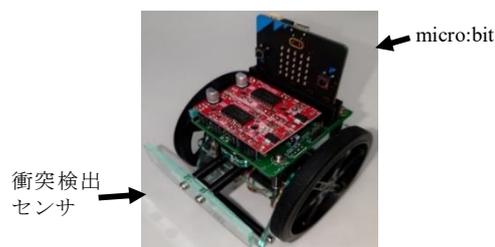


図1. メカトロニクス教材となるロボット台車外観

上位4bit	意味	上位4bit	意味
1000	繰り返し開始	0011	前進
1001	繰り返し終了	0000	後退
1101	センサ1入力確認	0001	左旋回
1110	センサ2入力確認	0010	右旋回
1111	衝突センサ入力確認	1100	停止

繰返指令	意味	直進・旋回指令	意味
0001	1回	0001	1cm/7.5度
0010	2回	0010	2cm/15度
:	:	0100	4cm/30度
1111	15回	1000	8cm/60度

図2. ビットパターンによる制御用コマンド

接続できる。またセンサーやアクチュエータの拡張用として I2C 用コネクタも用意した。

次に本教材を制御するプログラムについて説明する。本教材の制御用プログラムを micro:bit 固有の言語で直接記述すると、制御コードは intel hex 形式バイナリーデータとなり構造を解析することが非常に困難となる。そこで本教材の制御コードは micro:bit で直接制御するのではなく、ビットパターンによって制御する¹⁾。なおビットパターンは修正しやすさを考慮してテキストデータで記述する。次に制御用コマンドの1命令は8bitとし、図2の様にモータ制御とセンサー情報による分岐命令、そしてそれらを繰り返し用のコマンドに限定している。



図3. 本システムの構成

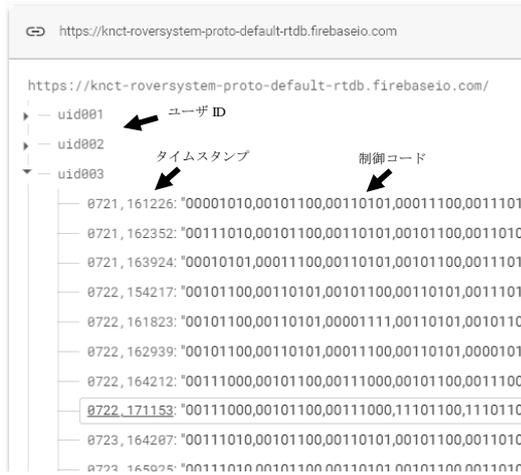


図4. 蓄積されたプログラミング履歴

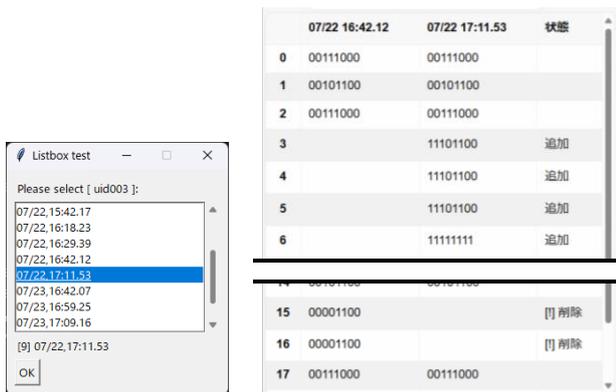


図5. データ選択画面(左)と見える化データ(右)

一連の動作をビットパターンでとなるテキストデータは転送専用のpython スクリプトにより、送信用micro:bit からロボット台車に無線送信される。

2.2 プログラミング履歴の収集の仕組み

本研究のプログラミング履歴とは学習者が記述したプログラムでもある制御コード群をいう。ここで学習者が制御コードを改変するたびに自らログを提出する方式では、ログの取りこぼしなどが懸念される。そこで、図3の様にプログラム転送に介入し、自動的にデータベースサーバに履歴を蓄積する仕組みを文献2)を元に構築する。今回データベースサーバとして、google社のFirebase realtime databaseを用いることとした。このrealtime databaseはJSON形式のデータを扱うことに特化しているため、プログラミング履歴についてもuser名をノード名とし、各

ノードに対してプログラム転送時間を表すタイムスタンプをkey, プログラムをvalueとしてrealtime databaseに蓄積することにする。この方式によって、特定の学習者のプログラミング履歴分析だけではなく、学習間のプログラミング履歴の比較分析も可能となる。図4は制御用ビットパターン群をrealtime databaseに蓄積していった様子を表している。

3. プログラミング履歴の活用

ロボットなどの制御プログラムは順次構造を入れ替えただけでも最終的な結果が異なることが容易に起こる。そのため、プログラムを指導する場合は学習者がどのような過程でプログラムを記述したのかを知ることも重要である。そこで、プログラムがどのように追加や変更されていくかを見える化する仕組みを考える。

まず、図5左に示す蓄積されたプログラムのタイムスタンプ一覧から参照したい日時を指定する。次に指定された日時のプログラムデータnと、その直前に蓄積されたプログラムデータn-1をrealtime databaseから取得する。そして二つのプログラムの差分を取り、プログラムデータnがプログラムデータn-1に対して追加された箇所と削除された箇所がわかるようにhtmlデータとして出力する。その出力結果は図5右の様になる。この図の1行目がプログラム行数、2行目がプログラムデータn-1の内容、3行目がプログラムデータnの内容、4行目が改変状態を表している。この図から3~6行目に条件分岐系のプログラムブロックが追加され、15~16行目モータ制御系のプログラムブロックが削除されたことが確認できる。

3. まとめ

本研究ではグループワーク用メカトロニクス教材に対して制御プログラムをプログラミング履歴として逐次データベースサーバに蓄積し、制御コードの時系列変化の追跡・活用とするシステムを実現した。今後は、本校の授業・実験で運用しながら、効果的な指導を実現するためユーザーインターフェースの改修などを行う予定である。

謝辞

本研究の一部は科学研究費基盤研究(C)課題番号22K02904の助成を受けたものである。ここに記して謝意を表す

参考文献

- 野口孝文他: “ロボット教材を用いたプログラミング遠隔協調学習環境”, 教育システム情報学会第49回全国大会論文集, pp.181-182 (2021)
- 千田和範他, クラウドサーバを介したIoTデバイス連携による臨場感強化型VRHMDシステムの開発, 教育システム情報学会第44回全国大会論文集, pp. 281-282 (2020)