

ビジュアルプログラミング環境 AT における Arduino 制御機能の開発

Development of the Functionalities for Controlling Arduino from a Visual Programming Environment: AT

國宗 永佳^{*1}, 河野 直^{*2}, 新村 正明^{*1}

Hisayoshi KUNIMUNE^{*1}, Nao KONO^{*2}, Masaaki NIIMURA^{*1}

^{*1}信州大学学術研究院工学系, ^{*2}信州大学大学院理工学系研究科

^{*1}Institution of Engineering, Shinshu University, ^{*2}Division of Science and Technology, Shinshu University
Email: kunimune@shinshu-u.ac.jp

あらまし: 我々はビジュアルプログラミング環境 AT を開発している. AT 上で作成したプログラムの動作は, 変数の値の変化とプログラムが出力するデータを画面内に示すことで, 学習者に伝えられていた. 本研究では, プログラムの動作を外界に反映させ, 直感的な理解を促進するために, AT における Arduino の制御機能を開発している. 本稿では AT における Arduino 制御機能の実装について述べる.

キーワード: プログラミング教育, 初中等教育, Arduino, ビジュアルプログラミング環境

1. はじめに

我々はプログラミング言語の文法を修得する必要なく, アルゴリズム的思考 (あるパラダイムにおける問題解決手順の分解と整理を行う思考) を学習するためのビジュアルプログラミング環境 AT を開発している⁽¹⁾. AT は使用可能なプログラム構成要素 (ブロック) の制御や課題の管理, 解答の分析など, 教授・学習活動の支援機能を備えるという特徴をもつ. これまでに高等教育機関 (大学・高専) のプログラミングに関する授業において, AT を用いたプログラミング教育が実践されている⁽²⁾⁽³⁾⁽⁴⁾.

AT ではプログラムの動作を, 変数の値 (内部状態) などを画面内に表示することで学習者に提示している. この方法によるプログラムの動作の提示は, コンピュータ単体で実現可能であるという利点があるが, 比較的抽象度が高い. 本研究では, プログラムの動作をコンピュータの外部に接続した機器に反映させ, プログラムの内部状態とともに機器の動作を確認することによって, より直感的な動作理解を促進することを目的として, AT に Arduino⁽⁵⁾を制御する機能を開発している. Arduino は, マイコンと入出力ポートを備える安価な小型の基板で, 入出力ポートにモータやセンサなどの様々な機器を接続して利用することが可能である. 本稿では, AT と Arduino 制御機能の概要について述べる.

2. 関連研究

Arduino を制御するプログラムを作成することが可能なビジュアルプログラミング環境として, BlocklyDuino⁽⁶⁾, Ardublock⁽⁷⁾, aiBlocks⁽⁸⁾などがある. BlocklyDuino と Ardublock では, ブロックの組み合わせに基づいて Arduino 言語で記述されたプログラムを生成する. このプログラムを統合開発環境 Arduino IDE でコンパイルし, Arduino に転送することで実行することができる. このプログラムは Arduino 上で動作するため, 動作状況を機器の動作

から確認することは可能であるが, 内部状態を確認することはできない.

aiBlocks では, 作成したプログラムを Arduino 上で動作するインタプリタで実行し, インタプリタのデバッグ用インタフェースを介して, 実行中のプログラムに対応する aiBlocks 上のブロックを強調表示することによって動作状況を示している.

上述したいずれの環境においても, プログラムの動作のうち, 変数の値については示すことができない. また, 教授・学習活動を支援する機能についても備えていない. 本稿で述べる AT では, Arduino を制御するプログラムを実行中に, プログラムの内部状態を学習者に提示するとともに, Arduino に接続された機器を動作させることによってプログラムの動作を提示することができる.

3. AT の概要

AT は web ベースのビジュアルプログラミング環境であり, 課題の管理・提示・採点や解答の分析, 使用可能なブロックの制御など, 教授・学習活動を支援する機能を有している.

AT のプログラム作成画面 (図 1) は, エディタ部と情報提示部とに分かれている. エディタ部では, ブロックを組み合わせることでプログラムを作成する. 情報提示部では課題の提出・保存, プログラムの実行・停止など各種操作を行うとともに, プログラムの内部状態や出力などを提示する. AT では, 作成したプログラムの動作を任意のブロックで中断するステップ実行機能を提供している. また, プログラムの動作状況を, 動作中のブロックの強調表示と, 各変数の値や出力の表示によって, 学習者に提示している. AT ではステップ実行と内部状態の表示を実現するために, ブロックを組み合わせたプログラムから javascript のプログラムを生成し, ブラウザのインタプリタ上で動作する JS-Interpreter⁽⁹⁾というインタプリタで実行している.

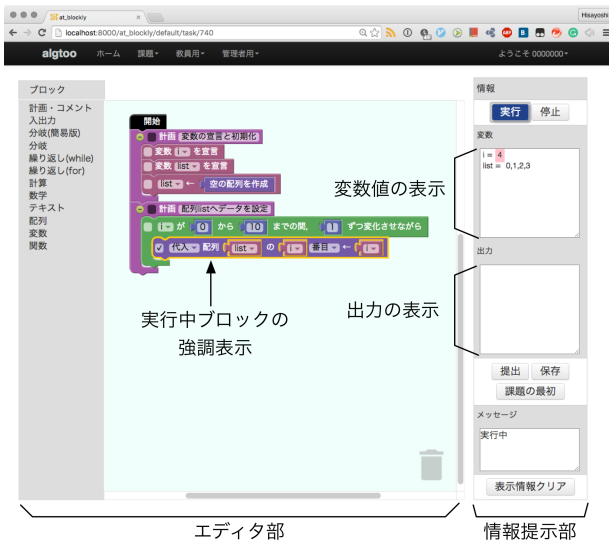


図 1 AT のプログラム作成画面

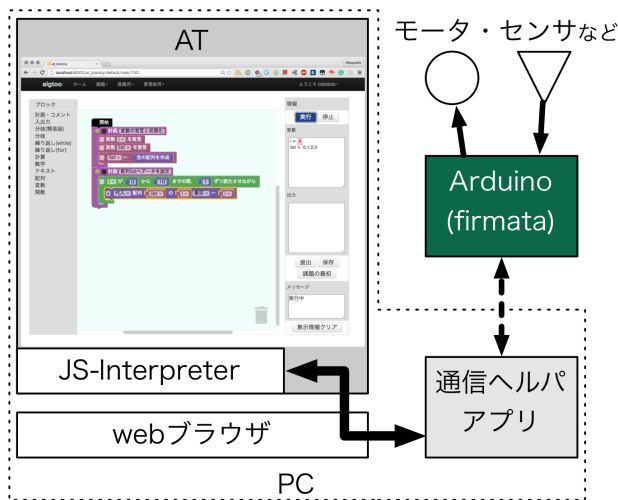


図 2 AT による Arduino の制御

4. AT への Arduino 制御機能の実装

本研究では AT でモータやセンサなどの機器を制御し、プログラムの動作を画面外で具体的に提示するために、Arduino を制御するプログラムを作成する機能を実装した。AT が提供している実行中ブロックの強調表示や、変数値や出力の表示を行いつつ、Arduino に接続した機器の制御を実現するために、次の要件を設定した。(1) プログラムの実行は AT 上で行う、(2) ブラウザ上で動作する AT から Arduino の制御を可能とする、(3) Arduino から取得した情報を AT に反映する。これらの要件を満たすために、図 2 に示す構成で実装を行った。

AT と Arduino の通信を実現するために、通信ヘルパアプリとよぶプログラムを開発した。通信ヘルパは、AT が動作する web ブラウザと同一の PC 上で動作し、web ブラウザとソケット通信を行っている。Arduino は PC と USB で接続され、通信ヘルパとの

間で Firmata プロトコル⁽¹⁰⁾による通信を行っている。Firmata は、PC とマイクロコントローラとの通信を行うためのプロトコルで、Arduino で専用のプログラムを実行することで、制御命令や取得した情報の通信を行うことができる。

また、AT には Arduino の制御に関するブロックを、BlocklyDuino のコードを流用することで追加した。このブロックを使用すると、AT では Arduino の制御命令を含むプログラムを生成する。JS-Interpreter では Arduino の制御に関するプログラムを、ブラウザのインタプリタを通して通信ヘルパアプリへ転送する。通信ヘルパアプリからは、Arduino へ命令を転送する。その結果、Arduino で制御命令が実行され、接続された機器の制御や情報の取得が行われる。取得した情報は、反対の経路で AT に転送され、画面上に反映される。

5. まとめ

本稿ではビジュアルプログラミング環境 AT において Arduino を制御するプログラムを作成する機能の開発について述べた。現時点では基本的な出力とデータの取得について実装を行った。今後、この機能を用いて教育を実践するための教材の整備を行う。

謝辞 本研究は JSPS 科研費 15K01023, 26350284 の助成を受けたものです。

参考文献

- (1) 國宗永佳, 大浦真暉, 香山瑞恵, 新村正明: “授業支援機能を有するビジュアルプログラミング環境 AT の開発”, 教育システム情報学会第 39 回全国大会講演論文集, 11-09, pp.17-18 (2014)
- (2) 大浦真暉, 國宗永佳, 山本樹, 新村正明: “プログラミング学習の導入を支援するビジュアルプログラミング環境の開発と評価”, 信学技報, Vol.114, No.260, ET2014-51, pp.73-78 (2014)
- (3) 萱津理佳, 香山瑞恵, 國宗永佳, 永井孝, 不破泰: “アルゴリズム的思考法に関する学習を取り入れた C 言語によるプログラミング教育の実践と評価 - 2010 年度から 2013 年度の 4 年間の実践より -”, 長野県短期大学紀要, Vol.69, pp.69-78 (2015)
- (4) 山本樹, 片桐恭弘, 國宗永佳, 須藤智: “文系学生を対象としたビジュアルプログラミングを用いたプログラミング初学者に対する教育の実践と評価”, 教育システム情報学会研究報告, Vol.30, No.7, pp.163-170 (2016)
- (5) “Arduino”, <https://www.arduino.cc/>
- (6) Fred Lin: “BlocklyDuino”, <https://github.com/BlocklyDuino/BlocklyDuino>
- (7) “Ardublock”, <http://blog.ardublock.com/>
- (8) 光永法明, 吉田図夢, 井芹威晴: “タブレット端末で動作する Arduino 用プログラミング環境 aiBlocks の初心者向け作例集の試作と評価”, 人工知能学会研究会資料, SIG-Challenge-042-04, pp.16-20 (2015)
- (9) Neil Fraser: “JS-Interpreter”, <https://github.com/NeilFraser/JS-Interpreter/>
- (10) “Firmata”, <https://github.com/firmata/protocol/>
(文献(5)(6)(7)(9)(10)には2016年6月6日にアクセスした)