

回路設計支援を目的とした Arduino シミュレータ ADVIS の現状と課題

西田 龍登^{a)}・片山 徹郎^{b)}

Current Status and Issues of Arduino Simulator ADVIS to Support Circuit Design

Tatsumi NISHIDA, Tetsuro KATAYAMA

Abstract

Embedded technologies are used everywhere. One of the educational materials to learn the embedded technologies is Arduino. When a beginner designs circuit, it is possible to design the circuit that damages Arduino itself or the modules on it. Therefore, this research implements Arduino simulator ADVIS (ArDuino VIRTUAL Simulator) that runs on iPad for supporting to design a circuit. ADVIS has 4 functions: “designing a circuit on Arduino” function, “visualizing the value of voltage in the circuit” function, “detecting a risk of Arduino itself being damaged” function, and “detecting a risk of module being damaged” function. In this simulator, the circuit on iPad is analyzed to detect the value of voltage on the circuit and the risk that Arduino itself or the modules are damaged.

Keywords: *Embedded technologies, Arduino, Simulator, iPad*

1 はじめに

現在、組み込みソフトウェア技術は日常生活の至るところで使われており、我々の生活に欠かせないものとなっている¹⁾。例えば、自動車、カーナビ、テレビ、エアコンで組み込みソフトウェア技術は使われている。また、組み込みソフトウェア技術を学習できる教材の1つとして Arduino がある。Arduino は、教材として世界中で使用されている^{2,3)}。Arduino は、センサ類やジャンパワイヤ、ブレッドボードなどをまとめた学習キットも市販されており、回路作成を行ったことがない人でも容易に開発を進められるという利点がある。しかし、初学者が Arduino を用いて回路設計をする場合、Arduino 本体やモジュールが破損する回路設計をする危険性がある。

そこで本研究では、Arduino における回路設計支援を目的とした iPad 上で動作する Arduino シミュレータ ADVIS (ArDuino VIRTUAL Simulator) を実装する。今回実装した ADVIS は、以下の機能を有する。

- 回路設計機能
- 回路上の電圧値可視化
- Arduino 本体が破損する危険性の検知
- モジュールが破損する危険性の検知

2 Arduino Uno

Arduino には、Arduino Uno、Arduino Leonardo、Arduino Yun など様々な種類が存在する⁴⁾。本研究で開発する ADVIS では、Arduino Uno を対象とする。図1に、Arduino Uno の概観を示す。Arduino Uno は、8bit マイクロコンピュータを搭載したデジタル入出力装置である。また、Arduino Uno は、本体に 32KB のフラッシュメモリを搭載している。

このため、本体の電源をオフにしてもプログラムを保存することができる。Arduino Uno は、各出力ピンから 20mA の電流を流す。この仕様より、ADVIS の各出力ピンから流れる電流は、20mA とする。

Arduino Uno は、ブレッドボードと一緒に使うことでより簡単に回路設計をすることができる。図2に、ブレッドボードの概観を示す。ブレッドボードは、はんだ付けを必要とせず、ブレッドボード上の穴に、ジャンパワイヤ、もしくはモジュールを挿すことで回路設計できるボードである。ブレッドボードの通電区間は、図2に示す赤線上である。通電区間とは、電気的につながっている箇所という意味である。

Arduino は、ユーザが設計した回路によっては、Arduino 本体が破損する危険性がある。具体的には、以下の3つがある。

a) 工学専攻 機械・情報系コース 大学院生

b) 情報システム工学科 教授

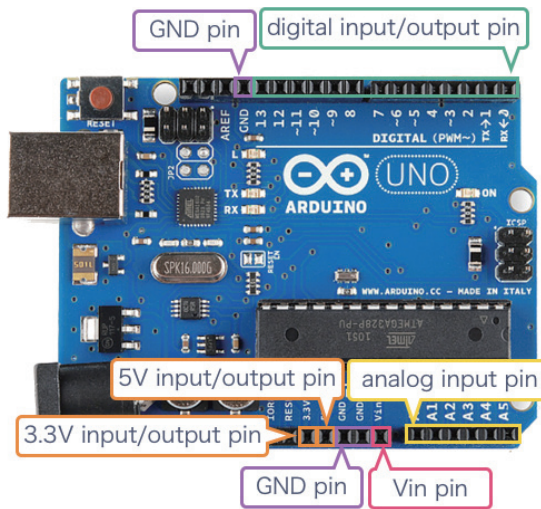


図 1. Arduino の概観

- 出力ピン同士が接続している場合に過電圧が発生する危険性
- 3V を超える電圧を入力ピンに送った場合に本体のブロッキングダイオードが破損する危険性
- 入力端子に 200mA を超える電流を受けると過電流が発生する危険性

ADVIS では、上記の 1 つ目と、2 つ目の危険性を検知できる。

3 Arduino シミュレータ ADVIS

ADVIS の動作環境は、以下のとおりである。

- iPad Pro 12.9
- iOS10 以上

図 3 に、ADVIS の概観を示す。ADVIS は、以下の 3 つのエリアから構成される。

- 回路設計部
- モジュール選択部
- メニュー

3.1 回路設計部

回路設計部では、ジャンパワイヤ、抵抗器、LED を使って、回路設計を行うことができる。

3.2 モジュール選択部

モジュール選択部では、回路設計部で扱うモジュールを選択する。ADVIS で扱うことができるモジュールは、ジャンパワイヤ、LED、抵抗器のみである。LED は、抵抗値を 100Ω とし、3V 以上の電圧が通電すると緑色に点灯する。

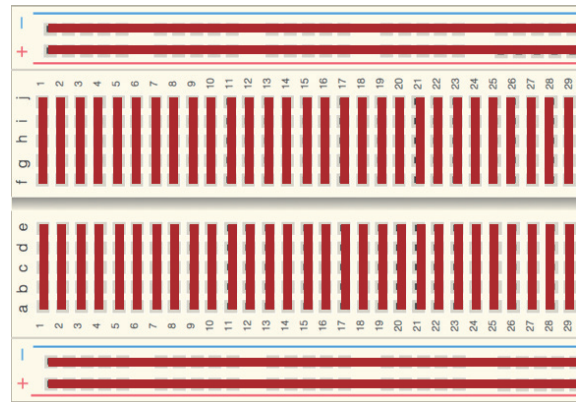


図 2. ブレッドボードの概観

抵抗器は、抵抗値を 150Ω としている。

3.3 メニュー

メニューでは、「CLEAR」機能、「BACK」機能、「RUN」機能、「VOLT RUN」機能を有している。

「CLEAR」機能では、回路設計部に配置したモジュール機能をすべて取り除き、初期状態に戻す。

「BACK」機能では、回路設計部を 1 つ前の状態に戻す。なお、「CLEAR」機能を使用した後は、「BACK」機能を使用しても「CLEAN」実行前の状態に戻すことはできない。

「RUN」機能では、回路設計部で作成した回路を実行する。また、モジュールが破損する危険性を検知した場合、モジュールの入力端子を赤色でハイライトする。本研究で破損の危険性を検知できるモジュールは LED のみである。図 4 に、「RUN」機能の例を示す。

「VOLTRUN」機能では、作成した回路上の電圧を表示する。また、Arduino 本体が破損する危険性を検知した場合、回路内の最後の出力となるピンに赤色でハイライトする。ADVIS が検知できる Arduino 本体が破損する危険性は、以下の通りである。

- 出力ピン同士が接続している場合に、過電圧が発生する危険性
- 3V を超える電圧を入力ピンに送った場合に Arduino のブロッキングダイオードが破損する危険性

図 5 に、「VOLT RUN」機能の例を示す。

3.4 ADVIS の機能

ADVIS は、2 章で述べた、Arduino 本体が破損する危険性に加えて、モジュール破損の危険性も検知可能である。ADVIS で検知可能なモジュールは、LED のみである。ADVIS が対象とする LED では、3V を超える電圧が流れると破損する危険性がある。

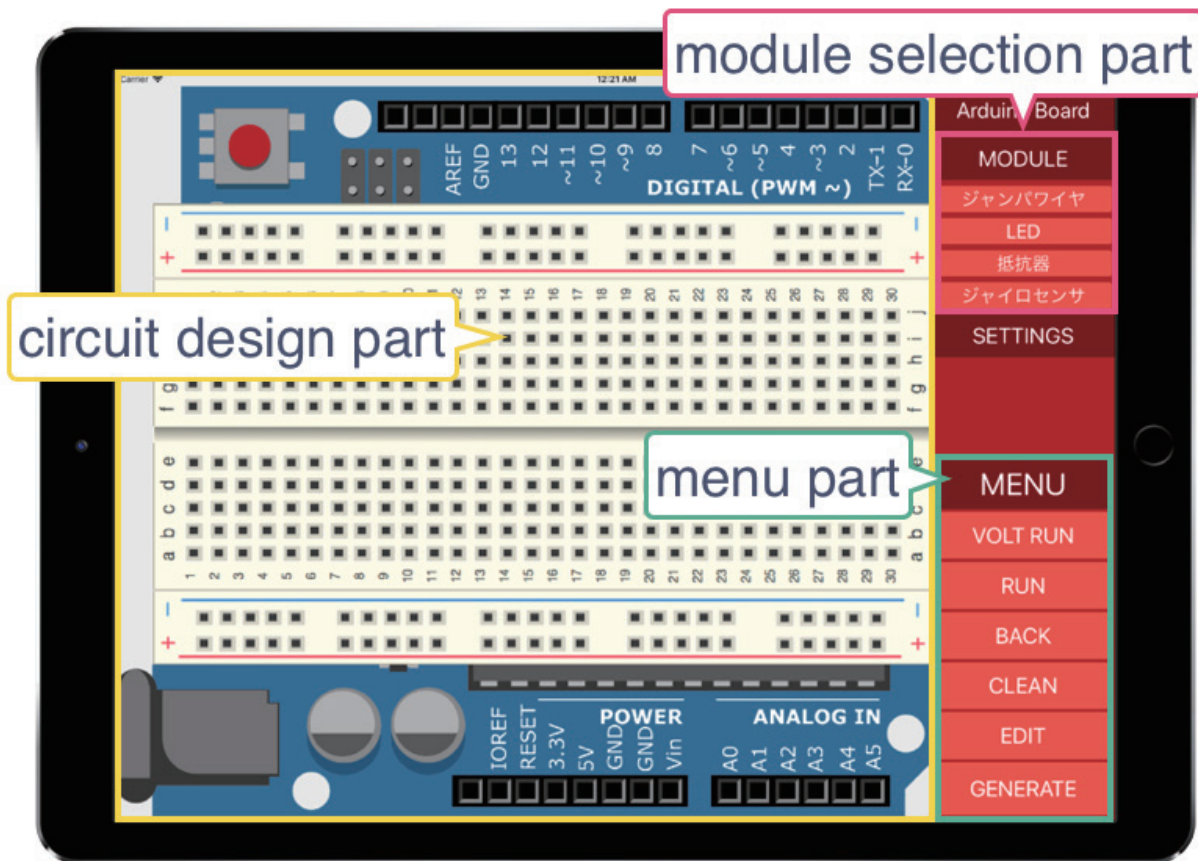


図 3. ADVIS の概観

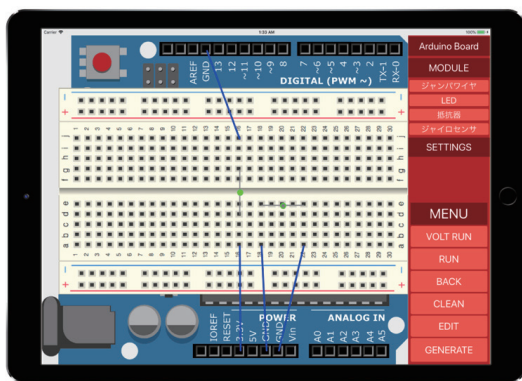


図 4. 「RUN」機能の例

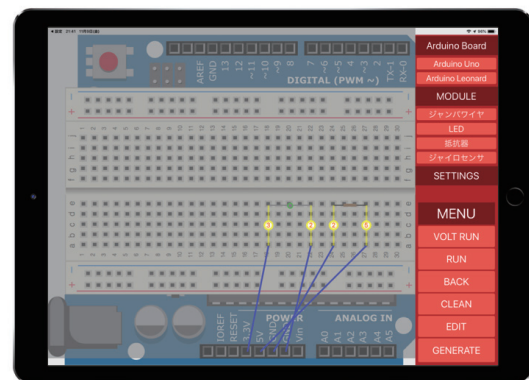


図 5. 「VOLT RUN」機能の例

4 動作検証

実装した ADVIS の動作検証を行う。動作検証として以下の動作を検証する。

- 回路にかかる電圧値の表示
- Arduino 本体が破損する危険性のある箇所を検出

- モジュール破損の危険性がある回路を検出

4.1 回路にかかる電圧値の表示

「VOLT RUN」機能を使用することで回路上の電圧値を表示する事ができる。例として、5V ピンと、Vin ピンをそれぞれブレッドボードに接続する。この回路に対して「VOLT RUN」機能を実行した結果を、図 6 に示す。

図 6 より、5V に接続した回路に対して 5V が表示し、

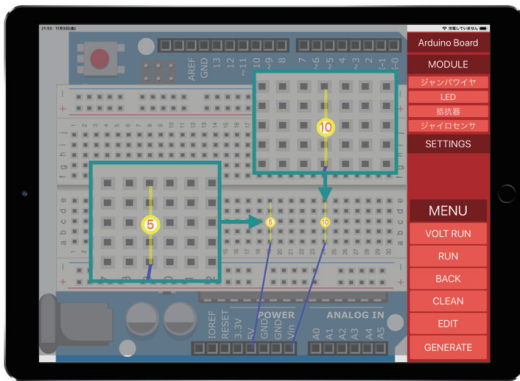


図 6. 回路上に電圧値を表示する例

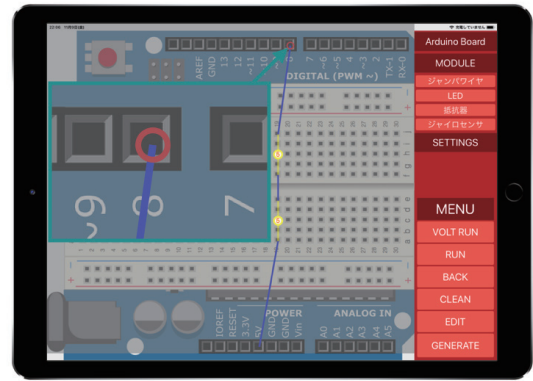


図 8. Arduino 本体が破損する危険性を検知する例

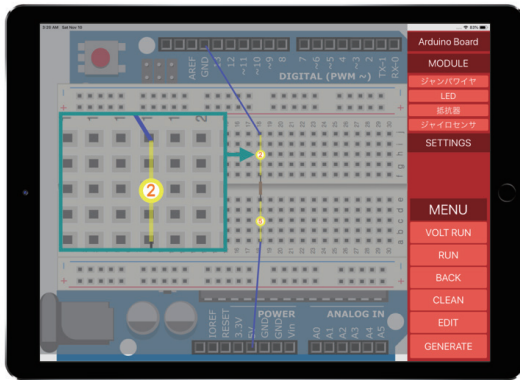


図 7. 抵抗器を使用した回路に対して電圧値を表示する例

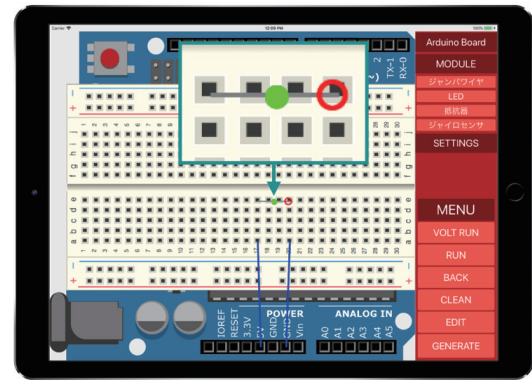


図 9. モジュールが破損する危険性を検知する例

Vin ピンに接続した回路に対して 10V を表示していることが確認できた。

次に、ブレッドボードに 5V ピンと抵抗器をブレッドボードに接続する。この回路に対して「VOLTRUN」機能を実行した結果を、図 7 に示す。図 7 より、5V ピンと接続している回路には 5V を表示している。また、抵抗器と接続した回路に対しては、オームの法則より求めた ($V = 0.02 \times 150 = 2$)、2V を表示している。

したがって、正しく表示していることが確認できる。

4.2 Arduino 本体が破損する危険性のある箇所を検出

「VOLT RUN」機能を使用することで、Arduino 本体が破損する危険性を検出できる。例として、5V ピンと、Vin ピンを入出力ピンに接続する。この回路は、3V を超える電圧を入力ピンに送っているため Arduino 本体のブロッキングダイオードが破損する危険性がある。図 8 に、「VOLTRUN」を実行した結果を示す。図 8 より、ADVISE が正しく Arduino 本体の破損を検出していることが確認できる。

4.3 モジュールが破損する危険性の検出

「RUN」機能を使用することで、モジュールが破損する危険性を検出できる。例として、LED を 5V ピンに接続する。この回路は、LED が 3V を超える電圧が流れるため、破損する危険性がある。図 9 に、「RUN」機能を実行した結果を示す。図 9 より、ADVISE が正しくモジュール破損の危険性を検出していることが確認できる。

5 関連研究

関連研究として、中村亮太氏らによって研究されている、「プログラムによる計測・制御を学ぶための LilyPad Arduino シミュレータと学習環境の開発」について述べる⁵⁾。

5.1 プログラムによる計測・制御を学ぶための LilyPad Arduino シミュレータと学習環境の開発

松浦敏雄氏らの研究グループは、手芸向けマイコンボード LilyPad Arduino を用いた教材および、そのプログラムを作成するための学習環境を開発した。図 10 に LilyPad Arduino の概観を示す。今回紹介する関連研究では、

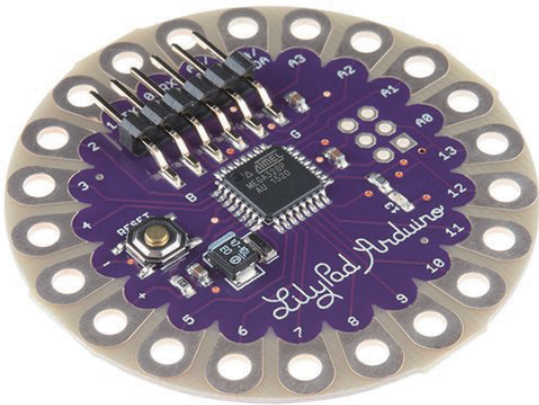


図 10. LilyPad Arduino の概観

LilyPad Arduino のシミュレータを実装し、初学者にとってプログラムの動作が分かりやすくなるようにした。

この論文では、プログラミング環境として、既に中村亮太氏らが開発している PEN を利用している。

5.2 プログラミング学習環境 PEN

PEN とは、Programming Environment for Novices の頭文字をとった名前であり、制御構造などのプログラミングの基礎を短時間で習得することを目指したプログラミング学習環境である⁶⁾。PEN で使用されるプログラミング言語は、xDNCL である。xDNCL は、大学入試センターの「情報関係基礎」で用いられる試験手順記述標準言語 DNCL に準拠しており、一部拡張したものである⁷⁾。そのため、日本語をベースにしており付加的な説明がなくても容易に理解できる。

PEN には、プログラムの動作を観察できるように一行実行機能や、低速実行機能を備えている。

PEN では、xDNCL を Arduino で用いられる言語に変換し、文法エラーが無ければ一時ファイルとして保存する。その後、PEN 内部から Arduino IDE をコマンドモードで起動し、実機の LilyPad Arduino にプログラムを書き込むことができる。しかし、実機上で作成したプログラムが、意図しない動作をした場合に、何が問題であるか、初学者が発見することは難しい。この問題点は、シミュレータ上であれば解決できる。また、実機上でセンサ値の細かな調整を必要とするプログラムの場合、調整することは難しい。この場合にもシミュレータは、有効である。

5.3 LilyPad Arduino シミュレータ

LilyPad Arduino シミュレータでは、各入出力装置が一体となった LilyPad 開発ボード⁸⁾をシミュレーションの対象としている。LilyPad 開発ボードでは、ボタンスイッチやスライドスイッチ、白色 LED、明るさセンサ、温度センサなどを搭載している。明るさセンサや、温度センサは、動作環境によって動作電圧が 0V~5V で変化する。シミュ



図 11. PEN と LilyPad シミュレータの実行画面 (文献 [5] より)

レータ上では、スライドバーを操作し、電圧ではなくアナログ入力値を 0~1023 の範囲で選択できる。これにより、実機上で確認の難しい細かなセンサ値の操作も容易にできる。

LilyPad Arduino シミュレータは入出力機器やピン番号が固定となっており、実際の LilyPad Arduino のように各機器を作品に合わせて自由に配置することはできない。したがって今後、様々な教材に活用できるようにするために、自由に配置・配線できるようにする必要がある。

5.4 比較

LilyPad Arduino シミュレータでは、様々なモジュールを実装している。これに対して、ADVIS では、明るさセンサ、温度センサなどの一部のモジュールを実装していない。ADVIS 上でこれらのモジュールをシミュレートするには、UISlider (スライドバー) を使うことで、実現できると考えられる。

ADVIS では、自由にモジュールや、ジャンパワイヤを配置・配線できる。しかし、LilyPad Arduino シミュレータでは、自由にモジュールや、ジャンパワイヤが配置・配線できない。この点に関しては、ADVIS の方が有用性が高いと考えられる。

6 まとめ

本研究では、Arduino における回路設計支援を目的とした iPad 上で動作する Arduino シミュレータ ADVIS (Arduino Virtual Simulator) を実装した。本研究で実装した ADVIS は、以下の機能を有している。

- 回路設計機能
- 回路上の電圧値可視化
- Arduino 本体が破損する危険性の検知
- モジュールが破損する危険性の検知

これらの機能を使うことで、Arduino における回路設計

支援が可能である。以下に、今後の課題を示す。

- 作成した回路の保存機能
現在の ADVIS は、ADVIS 上で作成した回路を、アプリ内に保存する機能を有していない。作成した回路の保存機能を実装することで、回路作成の手間を省くことができると考えられる。
- プログラミングによる回路制御
現在の ADVIS は、プログラミングによる回路制御に対応していない。Arduino は、プログラミングによる回路制御を行うことで組み込み技術を学ぶ教材である。そのため、プログラミングによる回路制御機能を実装することで、より本物の Arduino に近づくことができると思える。
- 作成した回路をもとにプログラムを自動生成する機能
現在の ADVIS は、ADVIS 上で作成した回路をもとにプログラムを自動生成する機能を有していない。作成した回路をもとにプログラムを自動生成する機能を実装することで、回路設計にかかる手間を省くことができると考えられる。
- 様々な抵抗値をもつ抵抗器や、モジュールの実装
現在の ADVIS は、様々な抵抗値をもつ抵抗器や、LED 以外のモジュールに対応していない。Arduino は、様々なモジュールを使うことで、回路設計を学ぶ教材である。そのため、様々な抵抗値をもつ抵抗器や、モジュールを実装することで、より幅広い回路設計支援が可能であると思える。

参考文献

- 1) 平山雅之:組み込みソフトウェア開発技術:1. 組み込みソフト開発の現状, 情報処理学会誌, Vol. 45, No. 7, pp. 677-681, 2004.
- 2) P. Jamieson, Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?, Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS), pp. 289-294, 2011.
- 3) Mohammed El-Abd, A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions, International Journal of Engineering Pedagogy, Vol. 7, No. 2, pp. 79-93, 2017.
- 4) <https://www.arduino.cc/en/Main/Products> (accessed 2019/2/15)
- 5) 中村亮太, 吉田智子, 松浦敏雄: プログラムによる計測・制御を学ぶための LilyPad Arduino シミュレータと学習環境の開発, 情報処理学会第 77 回全国大会, pp. 565-566, 2015.
- 6) 西田知博, 原田章, 中村亮太, 宮本友介, 松浦敏雄: 初学者用プログラミング環境 PEN の実装と評価, 情報処理学会論文誌, Vol. 48, No. 8, pp. 2736-2747, 2007.
- 7) <http://www.media.osaka-cu.ac.jp/~k-abe/joho-kiso/xDNCL-Language-Manual.pdf> (2019/2/15 閲覧)
- 8) <https://www.switch-science.com/catalog/1071/> (2019/2/15 閲覧)