

タッチ操作可能なブレッドボードシミュレータの開発

Development of touch-operable Breadboard Simulator

原 一彰, 吉村 忠志, 浅井 文男, 上野 秀剛

Kazuaki HARA, Tadashi YOSHIMURA, Fumio ASAI, Hidetake UWANO

奈良工業高等専門学校情報工学科

Information Engineering, Nara National College of Technology

Email:asuka.kazuaki@gmail.com

あらまし: タッチ操作が可能なブレッドボードシミュレータを Windows ストア アプリとして試作した。今後、被験者実験を実施し改良を加えて公開する予定である。

キーワード: ブレッドボードシミュレータ, Windows ストア アプリ, タブレット端末

1. はじめに

奈良高専の情報工学科では、論理回路および電子回路の学習においてブレッドボードを学生 1 人につき 1 枚ずつ用意している。ブレッドボードとは、抵抗素子や IC などの部品を用いた電子回路を半田付けなしに実装・検証することができる基板である。学生はこの基板に、論理ゲート IC などの SSI や JK-FF などの MSI, スイッチ類, ワイヤーなどを取り付け、種々の回路を実装し動作検証を行う。

2. 研究目的

学習にブレッドボードを用いる利点として、部品の取り付け・取り外しが容易であることや、部品の配置・配線を試行錯誤することで効率的な実装方法を体得できることなどが挙げられる。一方で、部品の劣化によって実装した回路が正常な動作をしない場合や、誤った配線に起因する過電流による発火などの事例が散見される。これらの問題により学習を進めにくくなったり、学生の安全が確保されなかったりする可能性がある。

そこで本研究では、タッチ操作可能なタブレット端末上でブレッドボードをシミュレートするアプリケーションソフトウェア（以下、提案アプリ）を開発することで、前述の利点を生かしつつ問題点を解決することを提案する。

3. 既存研究との比較

本研究で開発するアプリと同様にブレッドボードをコンピュータ上でシミュレーションできるソフトウェアと提案アプリの特徴を比較する。

3.1 Java Breadboard Simulator

Java Breadboard Simulator⁽¹⁾ はイギリスのヨーク大学の Nicholas Glass が開発したブレッドボードのシミュレータである。Java Breadboard Simulator は Java で開発されているため、プラットフォームによらず動作する。一方、本研究で開発するアプリは Windows ストア アプリとして開発するため、Windows 8 などでのみ動作する。また、Java Breadboard Simulator はマウスによる操作を基本とし

ているが、本研究で開発するアプリはタッチによる操作を想定している。

3.2 Virtual Breadboard

Virtual Breadboard⁽²⁾ は James Caska によって開発されたブレッドボードのシミュレータである。Virtual Breadboard は豊富な電子部品や電子回路のフィジカルコンピューティングデバイス Arduino をサポートしているという特徴がある。一方、本研究で開発するアプリの回路の保存および読み込み機能を用いることで、サンプル回路を提供することが可能である。しかし、実際のマイコンボード等との連携はできないという差がある。また、Virtual Breadboard もマウスによる操作を基本としている。

4. 提案手法

4.1 提案アプリの構成

提案アプリは Windows ストア アプリとして開発し、Windows 8 などが搭載されているタブレット端末にインストールする。利用者はインストールされた提案アプリを起動し、任意の論理回路の実装や動作検証をタッチ操作で行う。

提案アプリには、2 つの操作モードが存在する。表 1 に操作モードとそれぞれの機能を示す。一方は「編集モード」と呼ばれるモードで、画面上のブレッドボードに対して、部品の取り付けや取り外しを行うことができる。もう一方は「実行モード」と呼ばれるモードで、取り付けた部品が動作する様子を LED の点滅によって観察したり、スイッチ類の開閉操作をしたりできる。

表 1 操作モード

モード名	機能
編集モード	部品の取り付け, 取り外し
実行モード	動作の観察, スイッチ操作

提案アプリのプログラムは論理的に 2 つの部分からなる。一方はユーザが作成する論理回路の構成を保持し、シミュレーションを行うためのロジック部である。もう一方は、ロジック部が保持する回路の表示や、入力を処理しロジック部に渡すためのイン

タフェース部である。提案アプリのプログラムの論理的な構成を図1に示す。

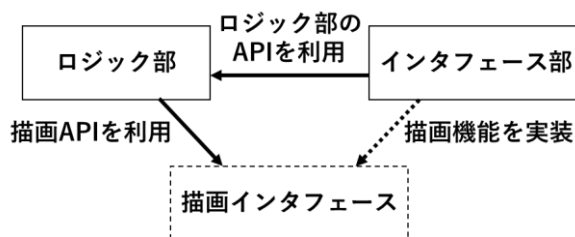


図1 提案アプリの論理的構成

ロジック部はプログラムの再利用性および適切な秘匿性を考慮し、メインプログラムと分離する。インタフェース部は回路の表示や入力処理などの必要な機能の実装を優先して行うため、随時プロトタイプを作成し、これを拡張する。また、これら2部分の結合を容易にするため、入出力機能についてはロジック部がAPIを提供し、表示機能については描画用のインタフェースクラスを作成して用いる。

また、論理回路の保存や読み込みに対応することでサンプルの回路の提供や演習課題の添削などが可能となり、学習の効率化を図ることができる。

4.2 開発環境

提案アプリはC#およびXAMLを用いてMicrosoft Visual Studio 2012上で開発し、Windows 8上で動作させる。また、Windows Storeへの登録にはMicrosoftによる審査を受ける必要があるが、開発工程を完了するまではVisual Studioを用いてローカルコンピュータに対するデプロイを行うこととする。

4.3 システムの動作検証と評価

提案アプリの動作検証は、実際に提案アプリで論理回路を構成し、その挙動によって評価する。用いる論理回路はJK-FFを4個用いた4桁バイナリアップカウンタで、LEDの点灯状況を確認する。

提案アプリの実用性は、本来ならば論理回路を学習する授業において、学生実験により評価すべきである。しかし実際の授業に用いることは現時点では困難であるため、提案アプリを被験者に利用してもらい、アンケートの調査結果を定量的にまとめることで評価する。

評価では、被験者に対して事前に用意した実験課題を与え、その後アンケートを実施する。実験課題は2段階に分けて行う。前半では提案アプリの操作方法の説明なしに課題を与えることで、初めて提案アプリを使用するユーザを想定し行う。アンケートでは操作方法の習得度合いから改善点を抽出する。後半では操作方法の説明を自由に参照できる状態で課題を与えることで、実際に学習に利用するユーザを想定して、操作感に関する改善点を抽出する。

5. 研究結果

4.1節で述べた提案アプリの構成を満たすようにアプリケーションソフトウェアを開発した。

動作検証として4桁バイナリアップカウンタを構

成したときの、実行モード時の動作画面のブレッドボード部分を図2に示す。図中①の白色および緑色の小円はそれぞれLEDの消灯および点灯を表しており、(0000)₂から(1111)₂まで遷移することが確認された。また、図中②のディップスイッチ上の2番のスイッチはリセット動作が割り当てられており、これを操作するとカウンタがリセットされた。なお、カウンタの動作に必要なクロック信号は図中③の端子から10 [Hz]で入力されるように設計した。以上のことから、提案アプリは正常な動作を示すことが分かった。

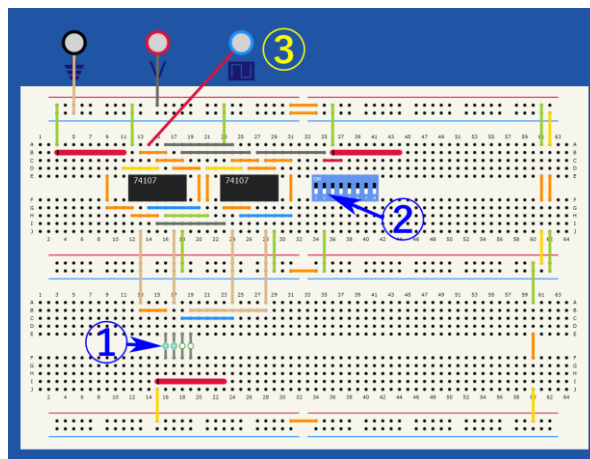


図2 提案アプリの動作画面（一部）

一方、被験者の確保および課題作成については現段階では終了していないため、被験者実験による提案アプリの客観的な評価および改善点の抽出は行っていない。

6. 今後の課題

提案アプリによって、論理回路をタッチ操作可能なデバイス上で実装および検証することができた。しかし、提案アプリの客観的な評価は行っていないことから、動作などに不具合が存在することが考えられる。また、論理回路学習に対する提案アプリの有用性についての検証もなされていない。この問題は、提案アプリを被験者に利用してもらい、不具合や改善点を抽出することで解決されるほか、実際の授業に用いることで、より現実に即した改善も可能である。さらに汎用性を高めるのであれば、論理回路だけでなくオペアンプに代表される電子部品を用いた回路のシミュレーションを可能にすることや、Windows 8以外のOSで動作するように移植することが考えられる。今後、提案アプリを実践的に利用することでこれらの課題の解決を試みたい。

参考文献

- (1) Bailey, Chris and Freeman, J, Michael : "A JAVA BREAD-BOARD SIMULATOR: DIGITAL CIRCUIT SIMULATION WITH AN OPEN-SOURCE TOOLSET", IADIS, Vol.5, No.1, pp.13-25 (2010)
- (2) 「Virtual Breadboard」, <http://virtualbreadboard.com/>, 2014年2月11日閲覧