

タブレット端末で動作する，インタプリタ型言語搭載 マイコンのプログラミング環境の開発

光永 法明^{1,a)}

概要：マイコンは身近な家電を含め多くの工業製品で活用されるだけでなく，計算機と実世界との接点であるセンサやアクチュエータとの接続が容易なものも多く，インタラクションデザインの教育・研究や中学校の技術科といった工学系以外の分野の教育での利用が拡大している．一方で，タブレット端末やスマートフォンなど小型で安価な端末が普及しつつある．そういった端末でマイコンのプログラミングが容易に出来れば，より利用の敷居が下がると考えられる．ところで，そういった端末は計算機資源が限られている．しかし，インタプリタをマイコンへ載せ，端末側にはコンパイラを持たず，エディタやデバッグに必要なもののみとすれば，計算機資源が限られていても問題がない．そこで本研究では，タブレット端末で動作する，インタプリタ型言語搭載マイコンのプログラミング環境を開発したので報告する．

1. はじめに

電子計算機が様々な場面で活用されるようになり，これからの活用の場面が広がりを見せると考えられている．なかでも実世界におけるユーザインタフェースへの電子計算機の活用が増えていくと考えられる．実世界との接点となるセンサ，アクチュエータと接続しやすい電子計算機としては周辺回路を内蔵したマイコン（マイクロコントローラ/マイクロコンピュータ）があり，多くの製品で使われている．また製品や工学系の教育においてだけでなく実世界との関わりを重視する，インタラクションデザインの教育・研究や中学校の技術科 [1] といった分野での教育での利用が拡大している．そういった用途でのマイコンの利用方法は，いわゆる試作に近い．一方で作品の制作者はマイコンを使った電子回路やプログラム開発に慣れてはいない．

初心者にとっては，マイコンの能力を最大限に生かせることよりも，目的の動作をさせるためのステップが少ない方がよい．そういった目的で BASIC 言語処理系 [2], [3], [4], [5] や Arduino [6] などが開発・市販されている．とくに利用するマイコンを限定し周辺回路の準備のステップを少なくした Arduino は利用者が多い．

そういった環境を用いても，多くの場合に，回路やプログラムが意図通りに動作していないときに，動作を可視化するツールとしては，デバッガや回路計（テスター），オシ

ロスコープといった従来からのツールが利用されている．しかし，それらのツールが利用可能であっても，操作の習得に時間がかかり，また慣れない接続でミスをすると余計に混乱するといった問題がある．

そういった問題は，特別な操作の習得が不要なマイコン用の開発環境にプログラムと回路の動作を可視化するツールを備えれば，解決すると考えられる．また対話的に動作を確かめられるインタプリタ型言語が有効な場面もあると考えられる．

ところで，タブレット端末やスマートフォンなど小型で安価な端末が普及しつつあり，そういった端末でマイコンのプログラミングが出来れば，開発環境を用意する敷居が下がると考えられる．安価な端末では計算機資源が限られているが，インタプリタをマイコンへ載せ，端末側にはコンパイラを持たず，エディタやデバッグに必要なもののみとすれば問題は少ない．

そこで本研究では，マイコン上で動作するインタプリタ [7] 用にタブレット端末 (Android OS) で動作する，インタプリタ型言語搭載マイコンのプログラミング環境を開発したので報告する．以下では，まずインタプリタと Windows 上でのプログラミング環境を紹介する．次にタブレット端末での実現を紹介し，最後に議論とまとめを述べる．

2. マイコン上で動作するインタプリタ [7]

iArduino は，Arduino のボードに搭載されたアトメルの AVR マイコン上で動作する，Arduino 言語 (C 言語ベース)

¹ 大阪教育大学
4-698-1 Asahigaoka, Kashiwara, Osaka 582-8582, Japan
^{a)} mitunaga@cc.osaka-kyoiku.ac.jp

と互換性のあるインタプリタ型言語である。マイコン上で動作するインタプリタは、シリアルポートで操作を受け付け、一般のシリアルターミナルでもプログラミングを含めた操作ができる。さらに、専用の Windows 上で動作するターミナルソフト iArduinoTerminal(図 1) を開発環境として使うことで、プログラムの実行中でも、変数の値が表示され、入出力ポートを GUI により操作できる。また、入出力ピンの値をグラフにより表示し、エディタ上で動作中の文をハイライト表示する。

iArduino はマイコン上で動作するインタプリタであるため、同一マイコン用にコンパイラ型言語によりプログラムを開発する場合と比べて動作速度やメモリの制約は大きい。そういった制約に到達した場合には、コンパイラ型言語である Arduino 言語に移行し、Arduino IDE を開発環境に利用できる。またコンパイラ型言語に移行した際にも、数行追加することで、図 1 の入出力ポートの操作とピンのグラフによる可視化が可能である。

3. タブレット端末で動作するプログラミング環境

現在、安価で容易に入手出来るタブレット端末の OS は iOS と Android の二つに分かれる。iOS 端末では外部接続用に自由に利用できる USB 端子はない。Android 端末の場合には機種によるが、端末が USB ホストになる USB ポートを備えたものがある。そこで Android 3.1 以降の OS で動作し、USB ホストモードの USB ポートを持つタブレット端末を対象にし、iArduinoTerminal for Android を作成した*1。開発に当っては Android OS 用の USB シリアルライブラリ FTDriver [8] を用いている。動作は acer の ICONIA TAB A200 (Android 4.0.3, 画面サイズ 10.1 インチ, 解像度 1280 × 800 ピクセル) で確認している。

マイコン上の iArduino とタブレット端末上の iArduinoTerminal for Android は USB シリアル (115.2kbps) を通じて通信する。プロトコルを表 1 に示す。プロトコルは Windows 上の iArduinoTerminal と iArduino が通信する場合と同じである。プログラムの実行文の範囲をのぞき、コマンドは端末側から要求を出し、マイコン側が応答する。各要求/応答を送る際には 0x7f を先頭につけ、プロトコル番号とデータを送る。データ長は可変である。チェックサムなどの誤り検出・訂正はしていない。

iArduino の動作する Arduino をタブレット端末に接続すると、起動アプリをたずねるダイアログが出るので、iArduinoTerminal for Android を選択すると図 2 が表示される。画面の左側にシリアルターミナルと変数の値を表示している。上のテキストボックスに文字を入力し、Send をタップするとシリアルから送信される。下のボタンを使っ

番号	要求/応答
0x00	バージョン
0x01	変数値
0x04	入出力ピンの値
0x0d	プログラムリスト
0x0e	プログラム実行時に実行位置を送る (応答なし, 以降プログラム実行中に実行文の範囲を送る)
0x0f	Arduino 起動時からの経過時間 (ms 単位)
0x10	ピンの入出力設定 (応答なし)
0x11	デジタル出力 (応答なし)
0x12	PWM 出力 (応答なし)
0x13	ラジコンサーボ出力 (応答なし)
0x14	ラジコンサーボ出力の設定 (応答なし)

て決まったコマンドを入力することも出来る。画面の右側はピンのモードを設定するプルダウンメニューと、ピンの値を操作するスライダが並んでいる。その右にはスライダを操作したときに、操作と等価になる文を表示する。

Editor/GUI/Graph ボタンをタップすると図 3 に表示が切り替わる。画面の右側にはプログラムを書くテキストエリアになる。プログラムの動作中には実行中の文をハイライト表示 (選択状態) する。下のボタン read によりマイコンからのプログラム読み出し、prog によりマイコンへのプログラム書き込み (揮発性の RAM 上)、save により EEPROM への記憶、autorun によりマイコンのリセット時にプログラムを自動実行するよう設定、nouto により自動実行の解除ができる。さらに Editor/GUI/Graph ボタンをタップすると図 4 に表示が切り替わり、画面の右側がグラフ表示になる。またグラフ上をタップすると、その時刻の値を表示する。

4. 議論

iArduinoTerminal for Android を作成にあたっていくつかの問題が見つかった。まず、Android 端末の場合には画面サイズと画素数の組み合わせが多いことである。アプリケーションを作成する際に、どのような画面を想定するかによって視認性や操作性の悪い端末が出てくる場合がある。当然であるが複数のサイズの画面に対応するのは開発者に負担がかかる。関連してウィジェットの配置について相対的なものが多く、実行時の自動配置も用意されている。一方で図 1 にあるような、Arduino を模した図の上にスライダを置くといったことが難しい。また PC 上の GUI と違って複数のウィンドウを並べることが想定されていないため、図 2~図 4 で見られるような画面の分割の仕組みをアプリケーション側で用意する必要があり、利用者が自由に並べることができない。しかしながら、Android OS の仕様は変化が早く、それほど時間を経ずに解決する問題もあると考えられる。

*1 <http://n.mtng.org/ele/arduino/iarduino-j.html> から入手できる

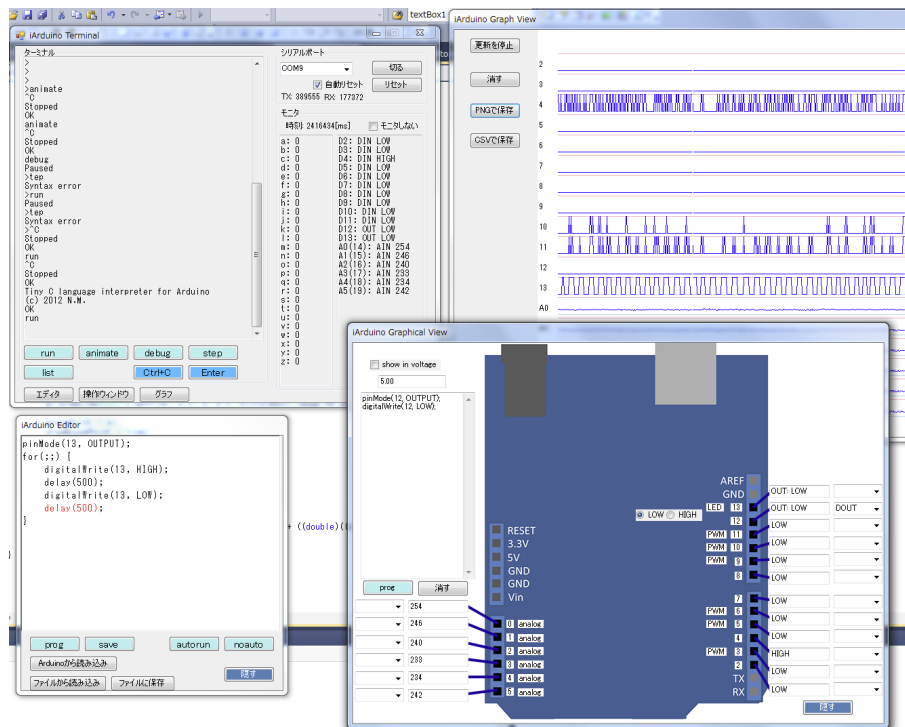


図 1 Windows 上で動作する iArduinoTerminal のウィンドウ. 左上のウィンドウが起動時に表示されるシリアルターミナルと変数, 入出力ピンモニタである. 右上のウィンドウは入出力ピンの変化をグラフで表示するウィンドウである. 左下のウィンドウはプログラム用のテキストエディタで, プログラム実行中は, 実行している行が赤で示される. 右下のウィンドウはピンの操作ウィンドウである. ピンを GUI で操作し, 回路の動作などを確かめられる. また操作をプログラムの文として表示する.

一方で, プログラムの作成時などには, 画面にソフトウェアキーボードが表示される, キーボードが画面の広い面積を占めるためプログラム全体が見にくくなる, など, ハードウェアキーボードをもたないタブレット端末に共通の問題も見られた. これにはボタンによる定型文の入力, ブロック状のアイコンによるプログラミングなど, あまりキーボードを必要としない工夫が必要になると考えられる.

5. まとめ

本論文ではインタプリタ型言語 iArduino が動作するマイコンの開発環境として, Android で動作する iArduinoTerminal for Android の開発を報告した. 本開発環境がマイコンを利用する方々, とくに初心者の方, 指導者の方の一助になれば幸いである.

参考文献

- [1] 文部科学省: 中学校学習指導要領解説技術・家庭編 (平成 20 年 9 月), 2008.
- [2] Parallax Inc.: BASIC Stamp Windows Editor, 詳細 (<http://www.parallax.com/tabid/295/Default.aspx>)
- [3] MCS Electronics: BACOM-AVR, 詳細 (http://www.mcselec.com/index.php?option=com_content&task=view&id=14&Itemid=41)
- [4] mikroElektronika: mikroBASIC Pro for PIC, 詳細 (<http://www.mikroe.com/eng/categories/view/98/>)

mikrobasic/)

- [5] MicroEngineering Labs.: PICBASIC PRO, 詳細 (<http://melabs.com/>)
- [6] Arduino project: Arduino, 詳細 (<http://arduino.cc/>)
- [7] Noriaki Mitsunaga. "An interpreted language with debugging interface for a micro controller", IEEE GCCE 2012, pp.115-119, 2012.
- [8] @ksksue: FTDriver: Android USB シリアルドライバ, 詳細 (<http://ksksue.com/wiki/doku.php?id=wiki:android:hardware:usb:ftdriver>)



図 2 iArduinoTerminal for Android の起動時の画面。左半面にシリアルターミナルと変数の値表示があり、右半面で Arduino の各ピンのモードと値を表示・操作できる。

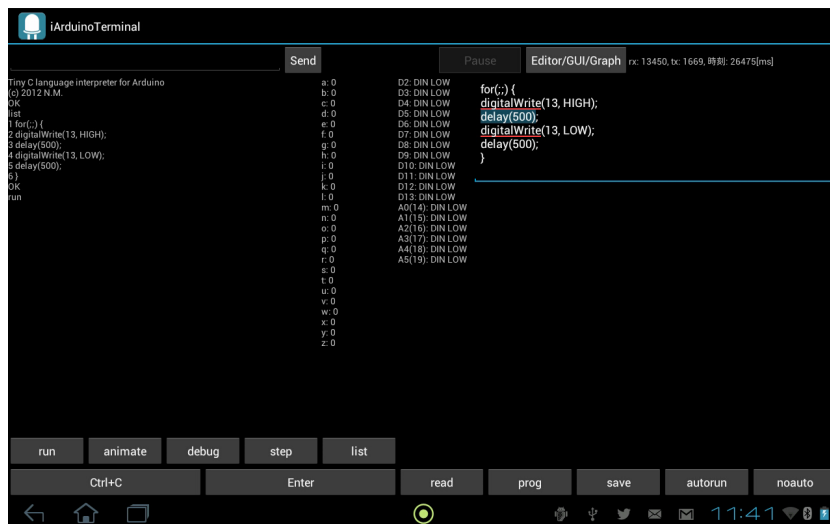


図 3 iArduinoTerminal for Android のエディタ表示画面。左半面にシリアルターミナルと変数の値表示があり、右半面にプログラムの編集と実行中の文を表示するエリアがある。

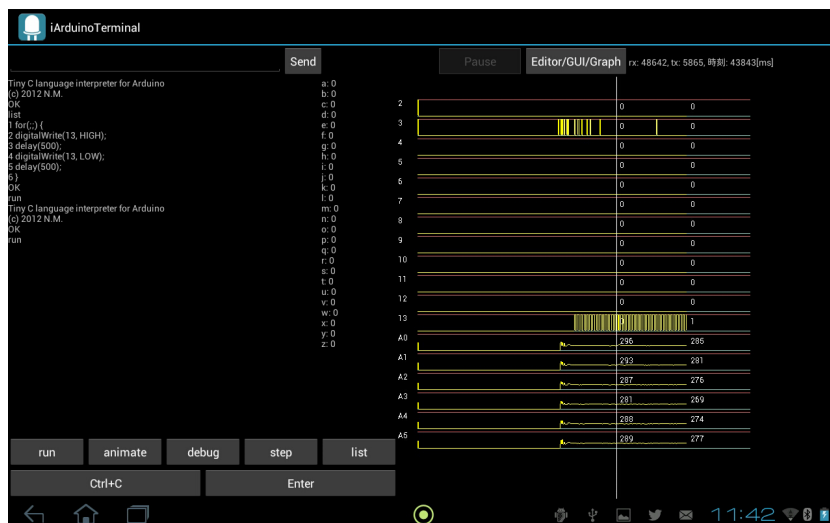


図 4 iArduinoTerminal for Android のグラフ表示画面。左半面にシリアルターミナルと変数の値表示があり、右半面に Arduino の各ピンの入出力の様子をグラフで表示する。