

Let's start electronics making with Arduino compatible kit "ChibiduinoICHI"



ちびでい～の<sup>いち</sup>壺ではじめる  
はじめてのでんしこうさく

AKBROBOT

日本Androidの会秋葉原支部ロボット部・ジャパン

# もくじ

- この冊子について
- LEDを光らそう!「光の流れ」
- ちびでい〜の壺の説明
- オリジナルArduinoとの違い
- 組み立て編
- 環境構築編
- より上級者編

# この冊子について


## この冊子について

- 「ちっちゃいものくらぶ」の販売する超安い Arduino 互換ボード「ちびでい〜の壺」の解説本です。
- この本は日本Androidの会秋葉原支部ロボット部ががんばって作りました!
- 最初に「LEDを光らそう」がありますが、必要に応じて組み立て編、環境構築編を見て準備をしてください。

# 基板セットの場合

- 以下のものがついています。

- ちびでい~の壱本体
- PCに接続するインターフェース
- USBケーブル
- ブレッドボード
- ジャンパーケーブル
- 基本実験部品セット

- 
- タクトスイッチ x 3
  - LED
  - 抵抗
  - 圧電スピーカ
  - 傾きスイッチ
  - CDS
  - サーミスタ

LEDを光らせよう!

オープンソースカンファレンス OSC 2013 Tokyo/Spring  
オープンハードウェアカンファレンス 「ファミリー向け体験コーナー」

## 「ハードウェア」と「ソフトウェア」のワークショップ

# 「光の流れ」テキスト

2013年2月23日(土)

ロボット部 清水 素釘武

(日本アンドロイドの会 秋葉原支部)



メーリングリスト: <http://groups.google.com/group/robot-android-group-japan-akb>

Webサイト: <https://sites.google.com/site/akbrobot/>



(C) 2013 AKBROBOT

# 「ファミリー向け体験コーナー」の概要

## 概要

目的: 「ハードウェア」と「ソフトウェア」に触れる

概要: 光の流れをつくる (電圧は5V)

対象: 小学生以上

コース時間: 1時間

コース方式: 個別指導

## 時間

(1)コース説明 10分

(2)ハードウェア工作 10分

(3)デモ1(1つ点滅) 5分

(4)デモ2(光の流れ) 10分

(5)問題(流れを逆にするなど) 15分

(6)まとめ、質問 5分

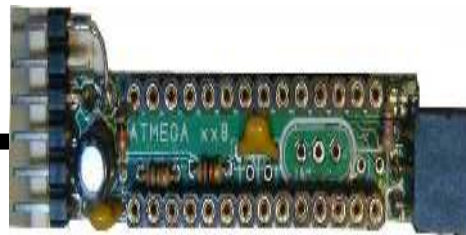
# 全体のつながり

パソコン



USB接続

電子基板



発光ダイオード  
(led)



- ・ソフトウェアを作成する
- ・ソフトウェアを電子基板に送る
- ・電気を供給する

- ・ソフトウェアを保存する
- ・ソフトウェアを実行する

- ・光る
- ・光の流れをつくる



# ハードウェア工作

①最初に先生が部品(抵抗、発光ダイオード、電線)を取付ます。よく見ていてください。

②つぎに写真を参考にして、部品を取付けてください。

抵抗

電気を流れにくくする。

ピン

ソフトウェアで指定した、電気を出力する

発光ダイオード

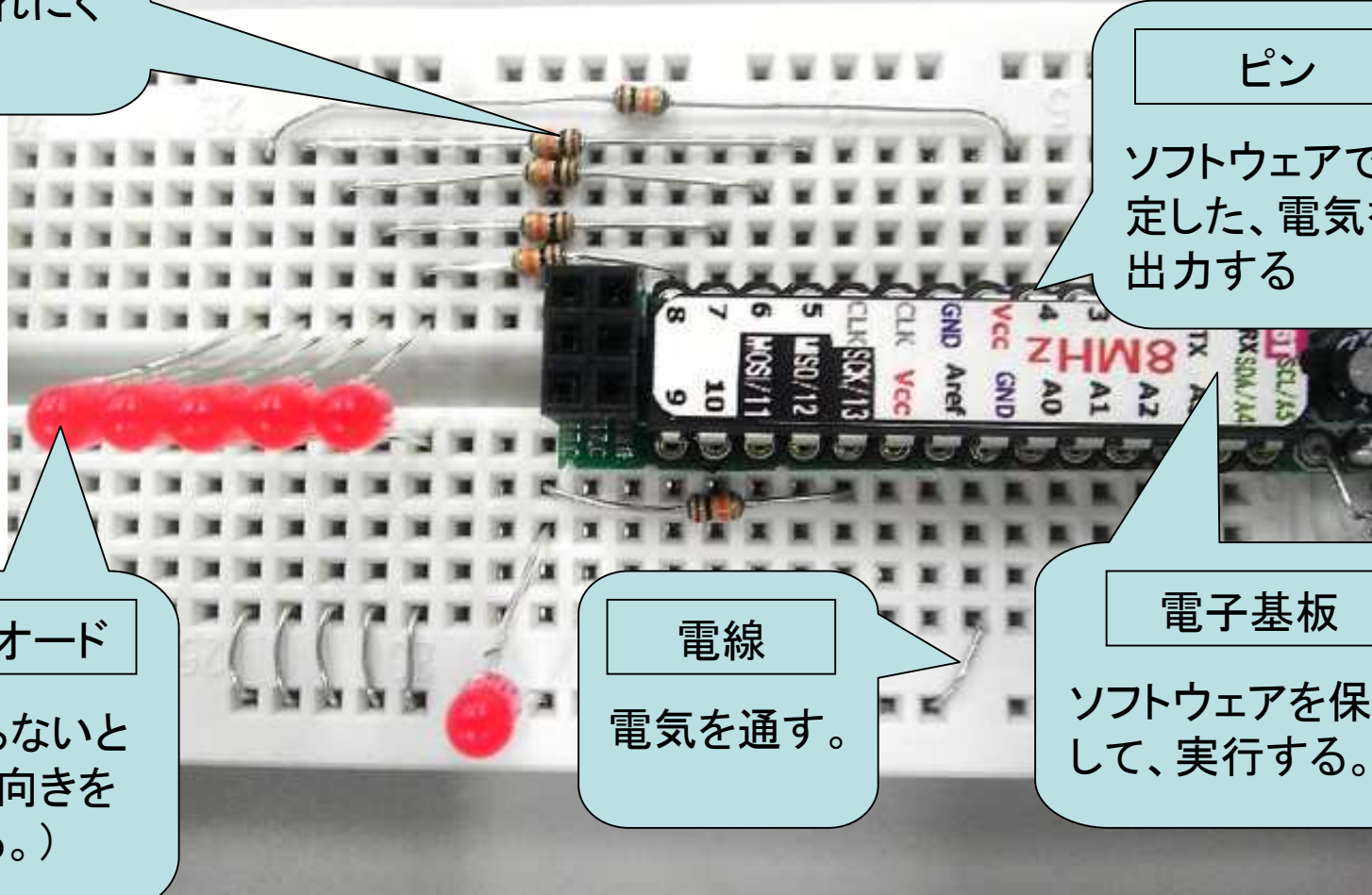
光る。(光らないときは、挿す向きを反対にする。)

電線

電気を通す。

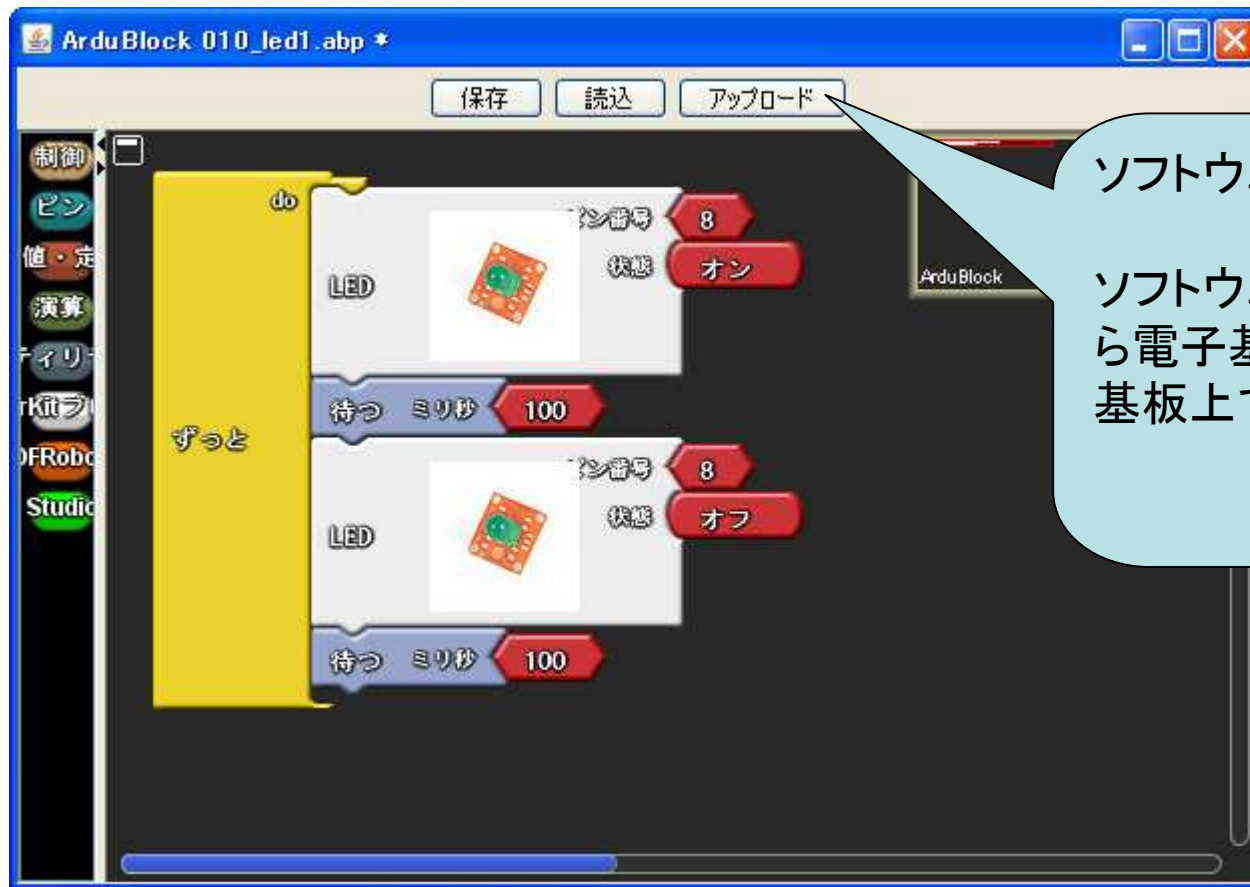
電子基板

ソフトウェアを保存して、実行する。



# 「デモ1」 1つ点滅

ソフトウェア1



# 「デモ2」光の流れ(発光ダイオード5本)

ソフトウェア2

ArduBlock 020\_led5.abp \*

保存 読込 アップロード

制御  
ピン  
値・定  
演算  
イリ  
Kitフ  
RRobc  
Studio

ずっと

繰り返し

変数 pin  
数値変数に値を設定 値 4

変数 time  
数値変数に値を設定 値 100

条件 8  $\geq$  pin

LED ピン番号 pin 状態 オン

待つ ミリ秒 time

LED ピン番号 pin 状態 オフ

待つ ミリ秒 time

変数 pin  
数値変数に値を設定 値 pin + i

時間を「100」から「30」に変更すると流れが速くなる。

# 問題に挑戦しよう。

- (1) 光の流れを逆方向にする
- (2) 光の流れを折り返す
- (3) 発光ダイオードを1つ増やす
- (4) その他

問題をとくアイデアは？

# もっと遊びたい

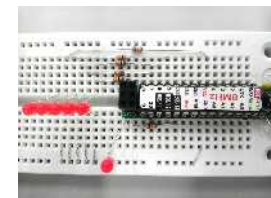
## 1. 本 (入手先:アマゾン<<http://www.amazon.co.jp/>> 秋葉原 若松通商など< <http://www.wakamatsu-net.com/biz/> >)

Arduinoをはじめよう 第2版 (Make:PROJECTS) ￥ 2,100



## 2. ハードウェア (つぎのいずれか)

- (1) 本テキストで使用した電子基板、部品  
「ちびでい~の言」ちっちゃいものくらぶ (キット) ￥ 800 <<http://tiisai.dip.jp/?p=2085>>  
(CPU基板、USBシリアル基板、USBケーブル、ブレッドボード)  
追加電子部品(発光ダイオード、抵抗など) ￥ 200程度(秋葉原 若松通商など)

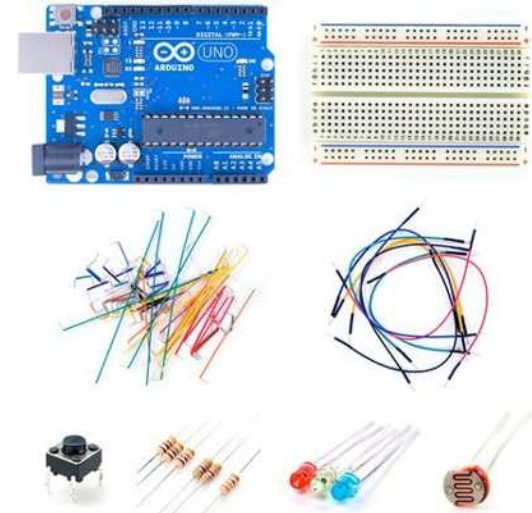
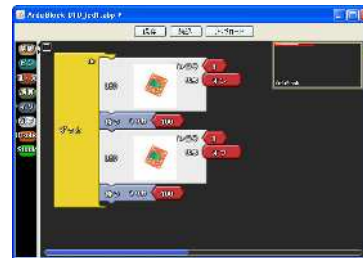


- (2) Arduinoをはじめようキット ￥ 3,980  
(入手先:アマゾン、秋葉原 若松通商など)

## 3. ソフトウェア開発ソフト (本テキストで使用したソフト)

Arduino IDE 無料 開発環境  
ArduBlock 無料 ビジュアルプログラミング環境

ダウンロードやインストール方法は、ロボット部の「プロジェクトAKBONE」などを参考にしてください。  
<<http://sourceforge.jp/projects/akbone>>



# 参加ありがとう!

# ちびでい〜の壺の説明

## 特徴

- より安く: 徹底したコストカットで究極の低価格を実現。
- よりちっちゃく: 丸ピンICソケットに薄い基板を張り付けて、AVRの裏側に回路を詰込みました。ATMEGA328と同幅の基板なので、ブレッドボードを無駄なく使ってハード試作できます。

# こんなにすごいぞ!

- AVR書込みもできる: これ一つで、arduinoIDEベースのマイコンシステムを増産できます。
- Arduino IDEでソフト作成
- USBシリアルモジュールを接続して、Arduino IDEでソフトを作成します。
- あらかじめ、ATMega8Lにブートローダーが書込まれています(Optiboot,8MHz内臓発振)
- ブレッドボード上でハード試作、ArduinoIDEでソフト開発。
- 動作したら、AVRチップをソケットから抜いて、ターゲット基板に差替えれば、組み込みシステムが作れます。
- AVRにブートローダ書込んで、Arduino用AVRを増殖できます。
- ボードのAVR書込用ISP端子と、ArduinoIDEに標準で付属しているArduinoISPを使って、Arduino用のブートローダーを新しいAVRチップに書込めます。
- ATMEGA88/168/328チップを入手して、ISP端子からの信号を6本接続します。
- ArduinoIDEからブートローダーを書込めばArduinoで使えるチップをクローンできます。
- このチップに差換えて、次のシステムを開発できます。このようにして安価にシステム開発できます。

# オリジナルArduino との違い

## Arduinoとの互換性

- ArduinoIDEでソフトウェア開発。
- スケッチアップロード時のオートリセット機能。



# Arduinoとの非互換

- 廉価なATMEGA8L 8MHzを採用。
  - プログラムサイズ8Kバイト、PWM3チャンネル。
  - ボード設定違うので、ArduinoIDEのboards.txt書換え
- USBシリアルチップは、廉価なPL2303HX。
  - ドライバが異なる。
- 電源コネクタ無し。USB電源(5V)を使用。
- シールドコネクタ無し。
- USBシリアルの送受信確認LED無し。

# Arduinoからの拡張

- ブレッドボードに直挿して、周辺回路を組める。
- AVR書込用 ISPコネクタ。Arduino同等なチップを増殖できる。

# 組み立て編

## 組み立てのやりかた

- ちびでい～の壺の組み立て方はWebで公開されています。
- [http://tiisai.dip.jp/?page\\_id=2109](http://tiisai.dip.jp/?page_id=2109)
- ここでは補足説明を行います。
  
- キット以外に必要な道具の説明 何が必要か?予算はどれくらい?
  - こちらにまとめてます!みてね!
  - <https://sites.google.com/site/akbrobot/etc/forstarter>

# 半田付けの技術について

- 半田付けの手順を図や写真で教えてほしい。手元の写真もあればうれしい。
  - →これは別のキット用ですが参考にしてください。  
<https://twitter.com/nanbuwks/status/225618594992631809/photo/>
  - 手元写真は…次の機会に。
- こて先の種類も多いため、どの種類が扱いやすいの？
  - →あまり細いのは熱が伝わりにくいのでよろしくありません。ちょっと太めの円錐形か先が平らになっているものの方がいいでしょう。
- 半田付けした部品の取り外し方法は？
  - →とりあえずはんだ吸い取り線を使いましょう。ちょっと慣れが必要かな。

## 説明資料について

- プログラムコネクタについて
  - 資料には書いていないが、4ピン目を45°位曲げ、ボードの表に半田付けする必要がある
  - 5ピン目は使わない
  - 6ピン目はピンだけでは届かないため、ほかのパーツを付けたときに出たリード線を使用することになる
- ピンソケット 3×2の説明について
  - 垂直と斜めのどちらかに決めた方がよい。
  - →写真ではななめを推奨しているが、ななめでつけるのは面倒なので垂直に決めてしまっても良い気がする。
- 半田付け後の処置について
  - 部品を取り付けた後にリード線を切る時に、根元まで切らないとブレッドボードにきちんと挿さらないので注意。

# より上級者編

- こちらをご覧ください。
- <http://togetter.com/li/354192>
- 一般的なArduinoを元に、説明しています。

# 環境構築編

- ちびでい〜の壱用のドライバは、Webページを参考にインストールしてください。
- ArduBlockの環境構築について、MacとLinux用の資料を添付します。
- WindowsはMac/Linuxを参考にしてインストールしてください。
- 当方でのWindowsの確認は、以下のパッケージを使用しました。ArduBlockも含まれています。
- [http://www.stem.saitama-u.ac.jp/htdocs/?page\\_id=38](http://www.stem.saitama-u.ac.jp/htdocs/?page_id=38) の  
<http://www.stem.saitama-u.ac.jp/uploads/arduino-0101-portable3-stemdu-02r1.zip>
- 
- Linux
- Mac
- 
-

# ちびでい~の同人誌

Mac 環境構築メモ

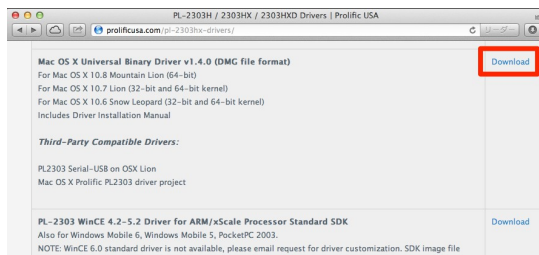
■ なかみがわ

## 1 PL2303 ドライバ導入

### ●ダウンロード

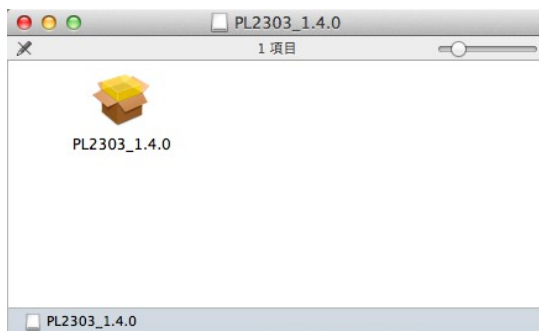
チップメーカーのサイトからドライバをダウンロードします (URL : <http://prolificusa.com/pl-2303hx-drivers/>)

上記サイトの上から2 / 3位のところにある「Mac OS X Universal Binary Driver v1.4.0 (DMG file format)」という部分の「Download」をクリックしインストーラをダウンロードします。



### ●インストール

ダウンロードした ZIP ファイルを解凍し dmG ファイルをダブルクリックしてマウントし、インストーラを起動してインストールを実行します。

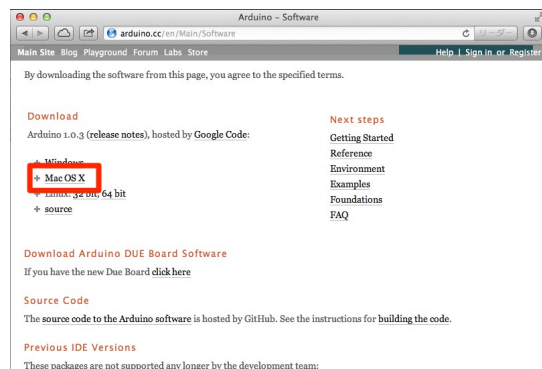


インストールは一般的なアプリケーションと変わりませんが最後に再起動が必要となりますので注意してください。

## 2 Arduino IDE 導入

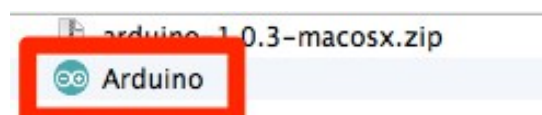
### ●ダウンロード

Arduino のプロジェクトサイトのダウンロードページ (URL : <http://arduino.cc/en/Main/Software>) よりダウンロード。



### ●インストール

ダウンロードした ZIP ファイルを解凍すると「Arduino」という名前でアプリケーションが解凍されますのでアプリケーションフォルダに移動します。

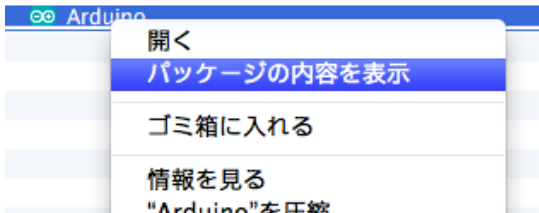


## ●ちびでい~のための設定追加

ちびでい~のは本家 Arduino と異なるチップを使用しているため設定の追加が必要となります。

Arduino の boards.txt を編集するのですが、Mac ではファイルが少々わかりにくい場所にあります。

アプリケーションフォルダへ移動した Arduino のパッケージの内容を表示。



パッケージ内の以下のファイルが編集するファイルとなります。

Contents/Resources/Java/hardware/arduino/boards.txt

以下を最後に追記してください。

```
#####  
#####  
atmega808.name=[Optiboot] ATmega8 RC8MHz
```

```
atmega808.upload.protocol=arduino  
atmega808.upload.maximum_size=7680  
atmega808.upload.speed=115200
```

```
atmega808.bootloader.low_fuses=0x94  
atmega808.bootloader.high_fuses=0xDC  
atmega808.bootloader.path=optiboot  
atmega808.bootloader.file=optiboot_atmega8-8m.hex  
atmega808.bootloader.unlock_bits=0x3F
```

```
atmega808.bootloader.lock_bits=0x0F
```

```
atmega808.build.mcu=atmega8
```

```
atmega808.build.f_cpu=8000000L
```

```
atmega808.build.core=arduino
```

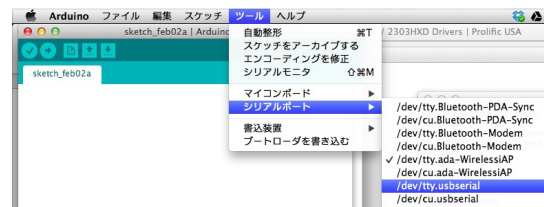
```
atmega808.build.variant=standard
```

```
#####  
#####
```

## ●動作確認

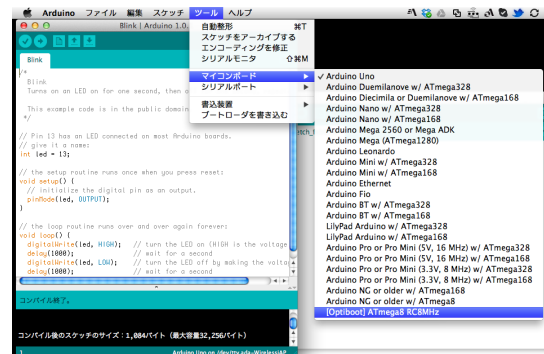
### ・確認 1

ちびでい~のを接続してシリアルポート「/dev/tty.usbserial」が増えていること。



### ・確認 2

ボードの選択肢に追加した「[Optiboot] ATmega8 RC8MHz」が増えていること。



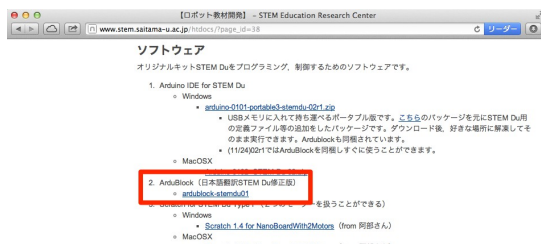
確認 1、2 が問題なければちびでい~のを Arduino IDE で使う準備は完了です。

### 3 ArduBlock 導入

#### ●ダウンロード

本家版の ArduBlock は日本語対応がいまいないため STEM 教育学習センターにて公開されている ArduBlock を使用します。

( URL : [http://www.stem.saitama-u.ac.jp/htdocs/?page\\_id=38](http://www.stem.saitama-u.ac.jp/htdocs/?page_id=38))



#### ●インストール

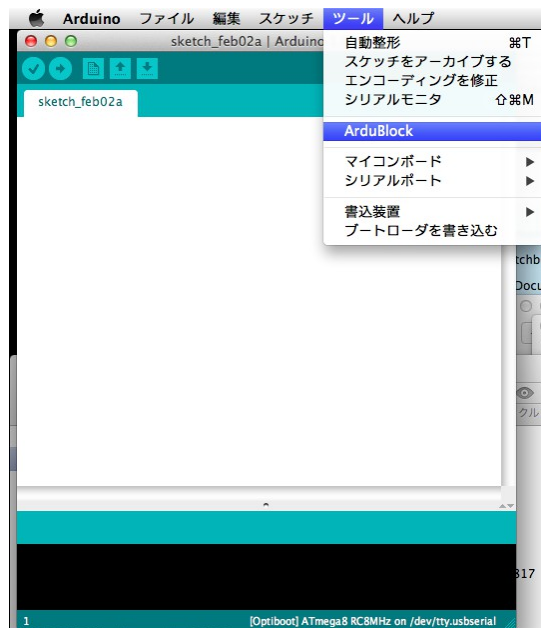
ダウンロードした ZIP ファイルを展開し、中の「tools」フォルダを ArduinoIDE のスケッチを保存するフォルダ（標準では「Users/ユーザー名/Documents/Arduino」）に移動します。

#### ●動作確認

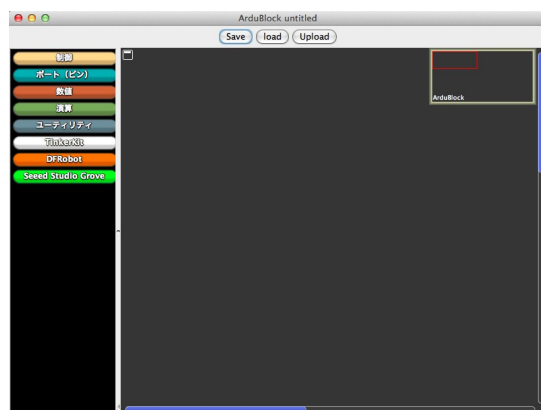
ArduinoIDE を起動してメニューのツールを開くと ArduBlock が追加されているはずで

す。ArduBlock を選択して ArduBlock が起動す

れば準備完了です。



ArduBlock の起動方法



ArduBlock の画面

### 4 ArduBlock の使い方

#### ●

作成中



matoken から共有されています

## Debian Wheezy amd64 上Arduino に ArduBlock を導入 & ちびでい〜の壺を利用できるようにする #den2moku2

更新日 2013, 2013

### Debian Wheezy amd64 に最新のArduino を導入

関連パッケージのために `arduino` パッケージを導入する.

```
$ sudo apt-get install arduino
```

以下からアーカイブをダウンロード. 今回はLinux でamd64 なので, Linux 64bit のarduino-1.0.3-linux64.tgz を入手.

"Arduino - Software" <http://arduino.cc/en/Main/Software>

展開して移動する.

```
$ tar xvzf arduino-1.0.3-linux64.tgz
```

```
$ cd arduino-1.0.3
```

※GUI 上で展開してもok

シリアルデバイス( `/dev/ttyUSB0` )を一般ユーザで動かすために, 利用ユーザをdialout グループに登録する

※以下はユーザが `mk` の場合の例.

```
$ sudo addgroup mk dialout
```

ユーザ `mk` をグループ `dialout` に追加しています...

ユーザ `mk` をグループ `dialout` に追加

完了。

OS を再起動して反映.

※ドライバは自動認識する.

※デバイスファイルはdmesg コマンドなどで確認可能. 以下の例では/dev/ttyUSB0 になる.

```
[48041.541964] usb 2-1.1: new full-speed USB device number 10 using ehci_hcd
```

```
[48041.634797] usb 2-1.1: New USB device found, idVendor=067b, idProduct=2303
```

```
[48041.634800] usb 2-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
```

```
[48041.634802] usb 2-1.1: Product: USB-Serial Controller
```

```
[48041.634803] usb 2-1.1: Manufacturer: Prolific Technology Inc.
```

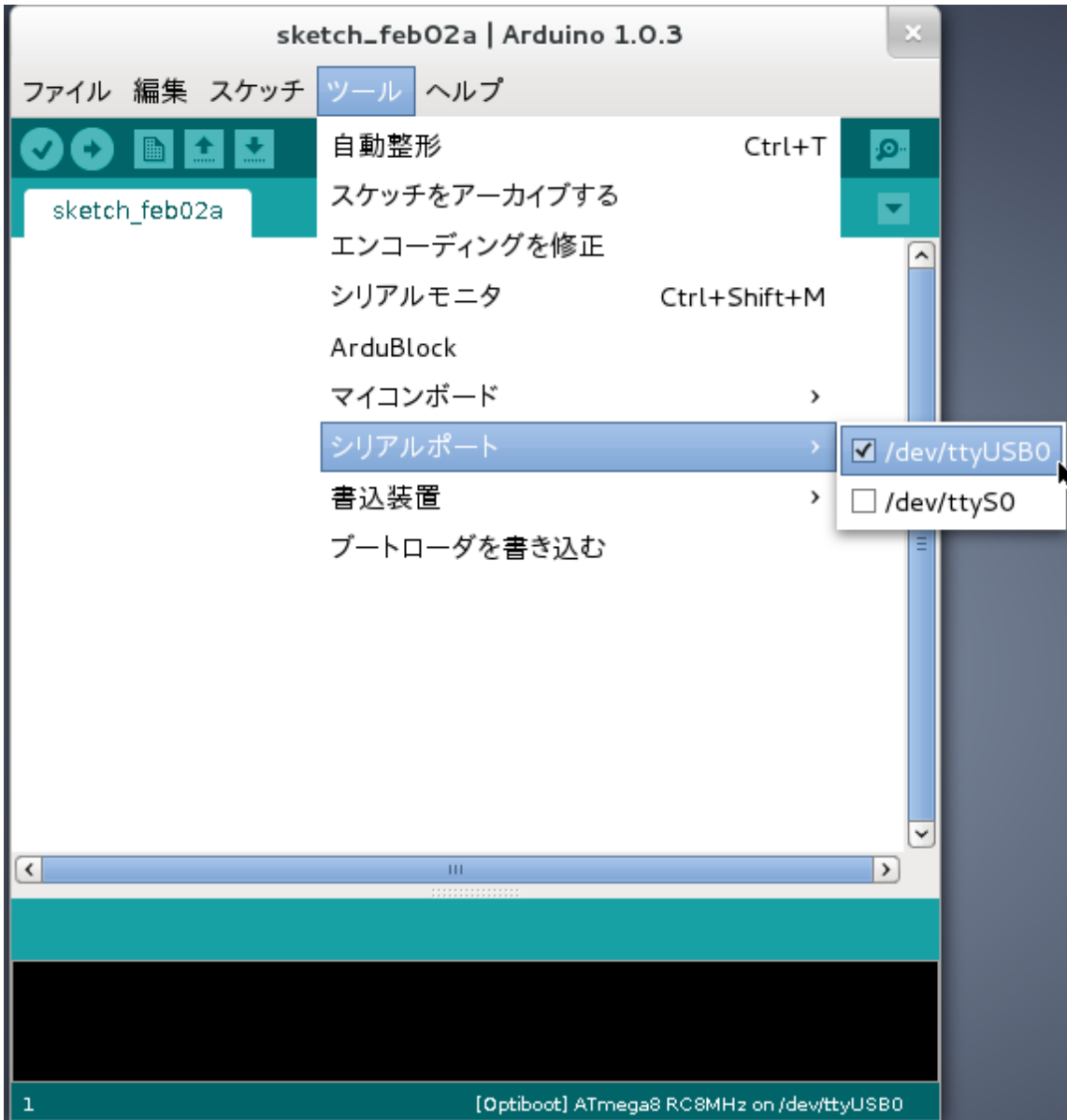
```
[48041.635146] pl2303 2-1.1:1.0: pl2303 converter detected
```

```
[48041.636674] usb 2-1.1: pl2303 converter now attached to ttyUSB0
```

### Arduino でシリアルの設定をする

Arduino IDE から[ツール]->[シリアルポート] で利用するArduino のシリアルポートを選択する.

```
$ arduino &
```



※デバイスファイルはdmesg コマンドなどで確認可能. 以下の例では/dev/ttyUSB0 になる.

```
[48041.541964] usb 2-1.1: new full-speed USB device number 10 using ehci_hcd
[48041.634797] usb 2-1.1: New USB device found, idVendor=067b, idProduct=2303
[48041.634800] usb 2-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[48041.634802] usb 2-1.1: Product: USB-Serial Controller
[48041.634803] usb 2-1.1: Manufacturer: Prolific Technology Inc.
[48041.635146] pl2303 2-1.1:1.0: pl2303 converter detected
[48041.636674] usb 2-1.1: pl2303 converter now attached to ttyUSB0
```

#### ちびでい〜の壺を利用する準備

"ちびでい〜の壺 の作り方 « ちっちゃいものくらぶ" [http://tiisai.dip.jp/?page\\_id=2109](http://tiisai.dip.jp/?page_id=2109)  
hardware/arduino/boards.txt の末尾に下記を追加する.

※gedit などを利用.

```
#####
atmega808.name=[Optiboot] ATmega8 RC8MHz
```

```
atmega808.upload.protocol=arduino
atmega808.upload.maximum_size=7680
```

```

atmega808.upload.speed=115200

atmega808.bootloader.low_fuses=0x94
atmega808.bootloader.high_fuses=0xDC
atmega808.bootloader.path=optiboot
atmega808.bootloader.file=optiboot_atmega8-8m.hex
atmega808.bootloader.unlock_bits=0x3F
atmega808.bootloader.lock_bits=0x0F

```

```

atmega808.build.mcu=atmega8
atmega808.build.f_cpu=8000000L
atmega808.build.core=arduino
atmega808.build.variant=standard
#####

```

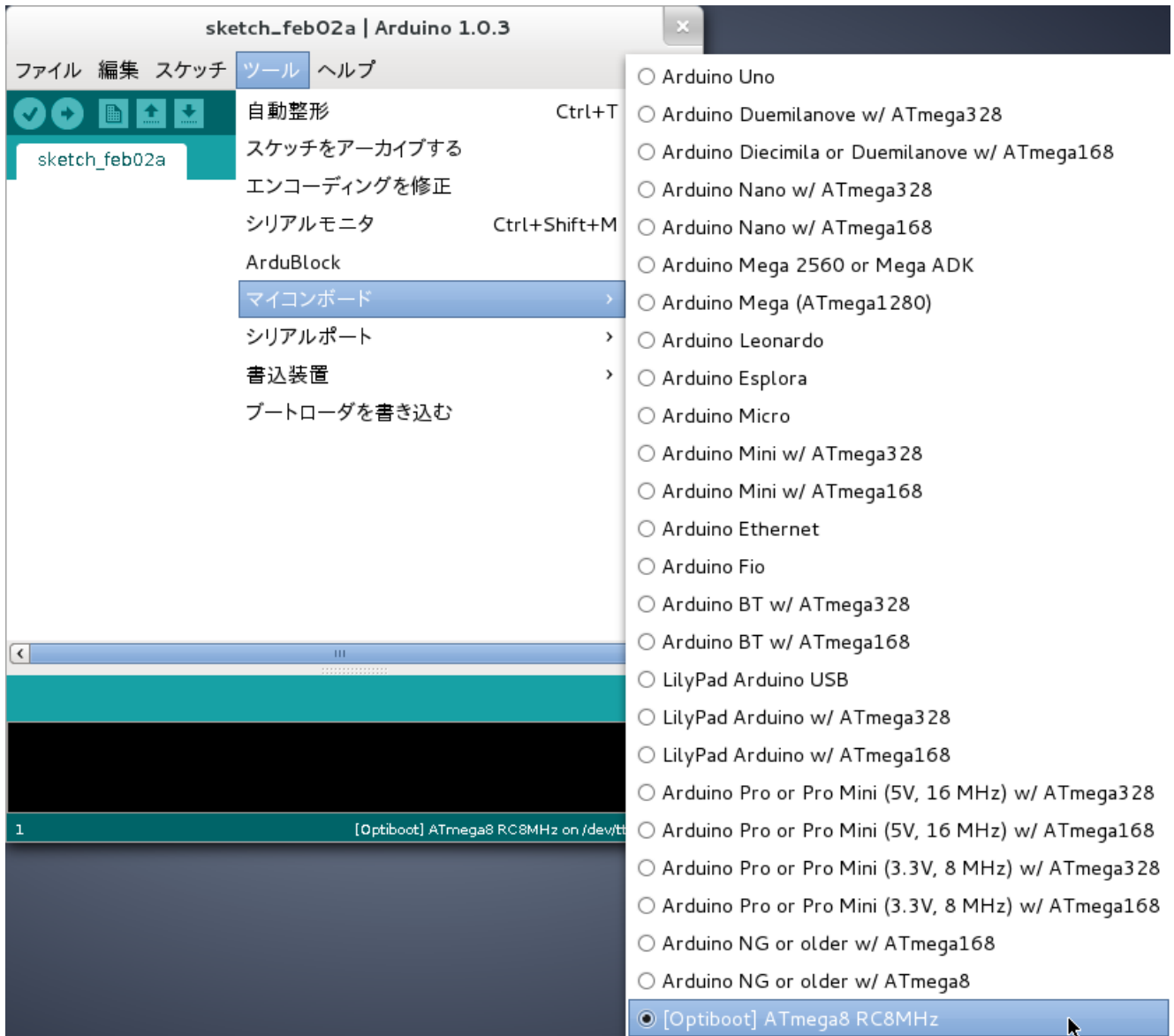
コンソールから追記する例.

```

$ cat << EOF >> hardware/arduino/boards.txt
> #####
> atmega808.name=[Optiboot] ATmega8 RC8MHz
>
> atmega808.upload.protocol=arduino
> atmega808.upload.maximum_size=7680
> atmega808.upload.speed=115200
>
> atmega808.bootloader.low_fuses=0x94
> atmega808.bootloader.high_fuses=0xDC
> atmega808.bootloader.path=optiboot
> atmega808.bootloader.file=optiboot_atmega8-8m.hex
> atmega808.bootloader.unlock_bits=0x3F
> atmega808.bootloader.lock_bits=0x0F
>
> atmega808.build.mcu=atmega8
> atmega808.build.f_cpu=8000000L
> atmega808.build.core=arduino
> atmega808.build.variant=standard
> #####
> EOF

```

Arduino IDE を起動して, [ツール] -> [マイコンボード] -> [[Optiboot] ATmega8 RC8MHz] を選択する.





### ArduBlock の導入

ArduBlock を以下からダウンロードする。

"ArduBlock" <http://blog.ardublock.com/>

ArduBlock はこっちのもの( ArduBlock(日本語翻訳STEM Du修正版) )を使う

"[ロボット教材開発] - STEM Education Research Center" [http://www.stem.saitama-u.ac.jp/hdocs/?page\\_id=38](http://www.stem.saitama-u.ac.jp/hdocs/?page_id=38)

Arduino ディレクトリ以下に `tools/ArduBlockTool/tool` というディレクトリを作成してその中にダウンロードしたArduBlock を移動する。

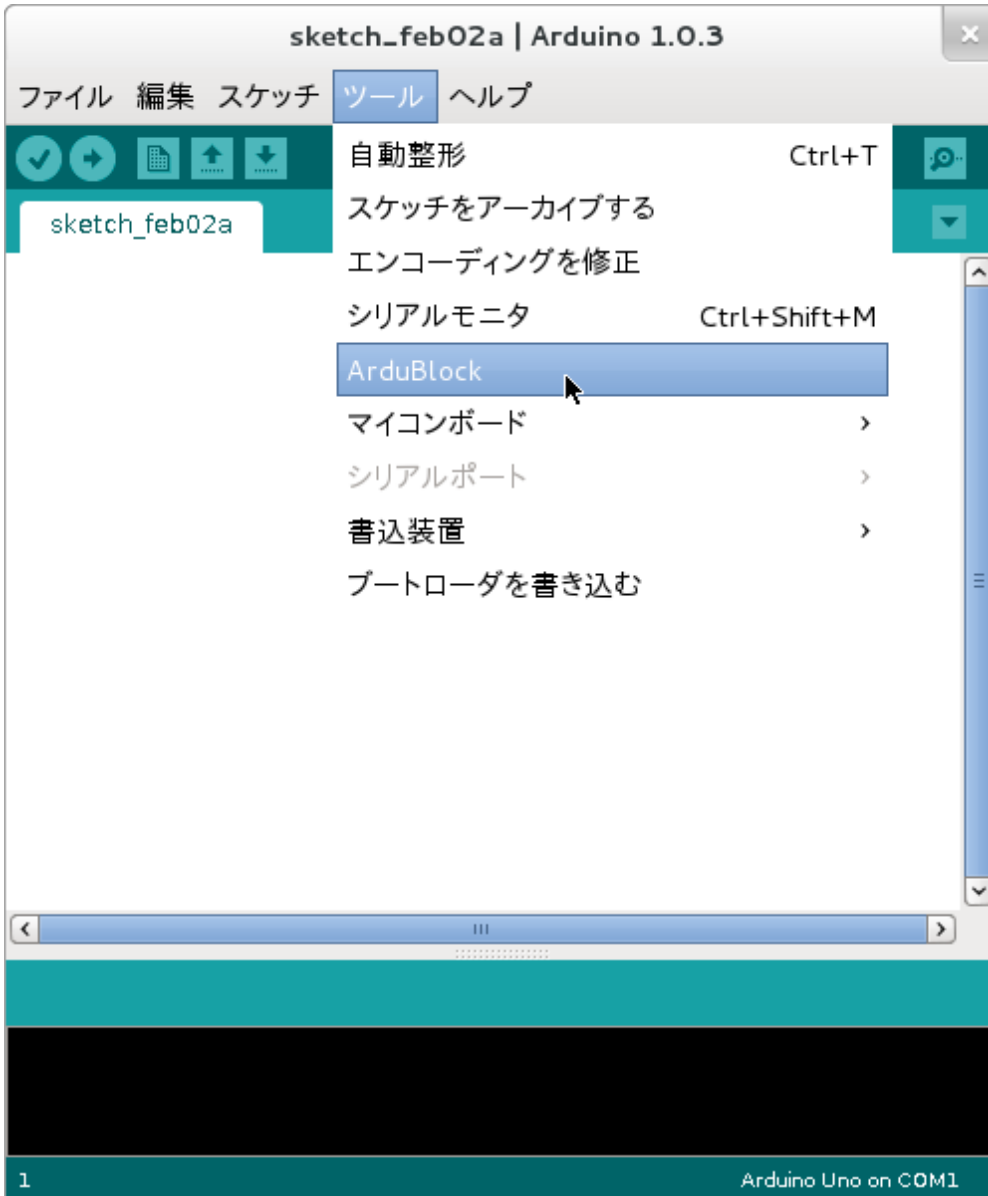
```
$ mkdir -p tools/ArduBlockTool/tool
```

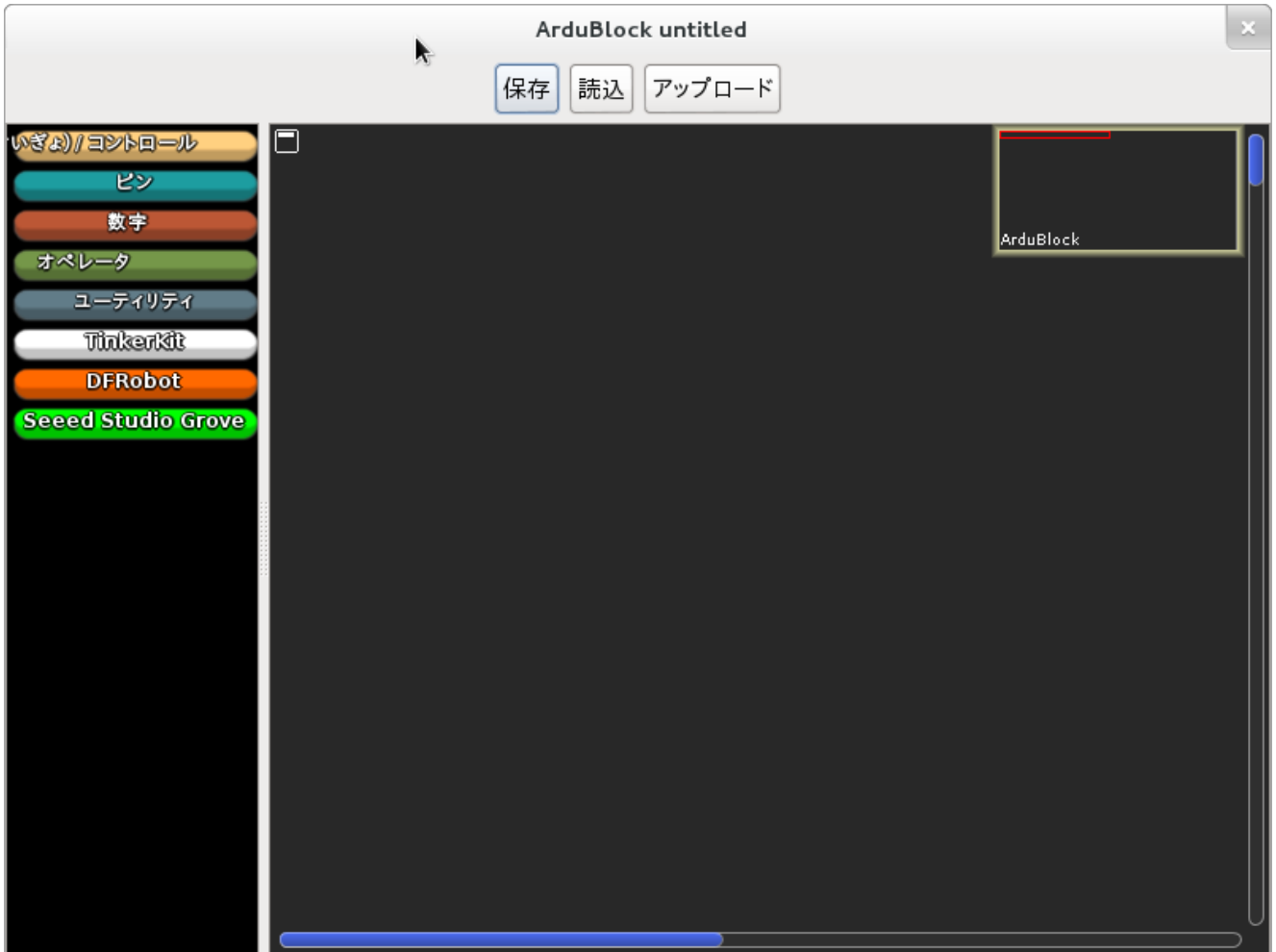
```
$ unzip ~/Downloads/ ardublock-stemdu01.zip
```

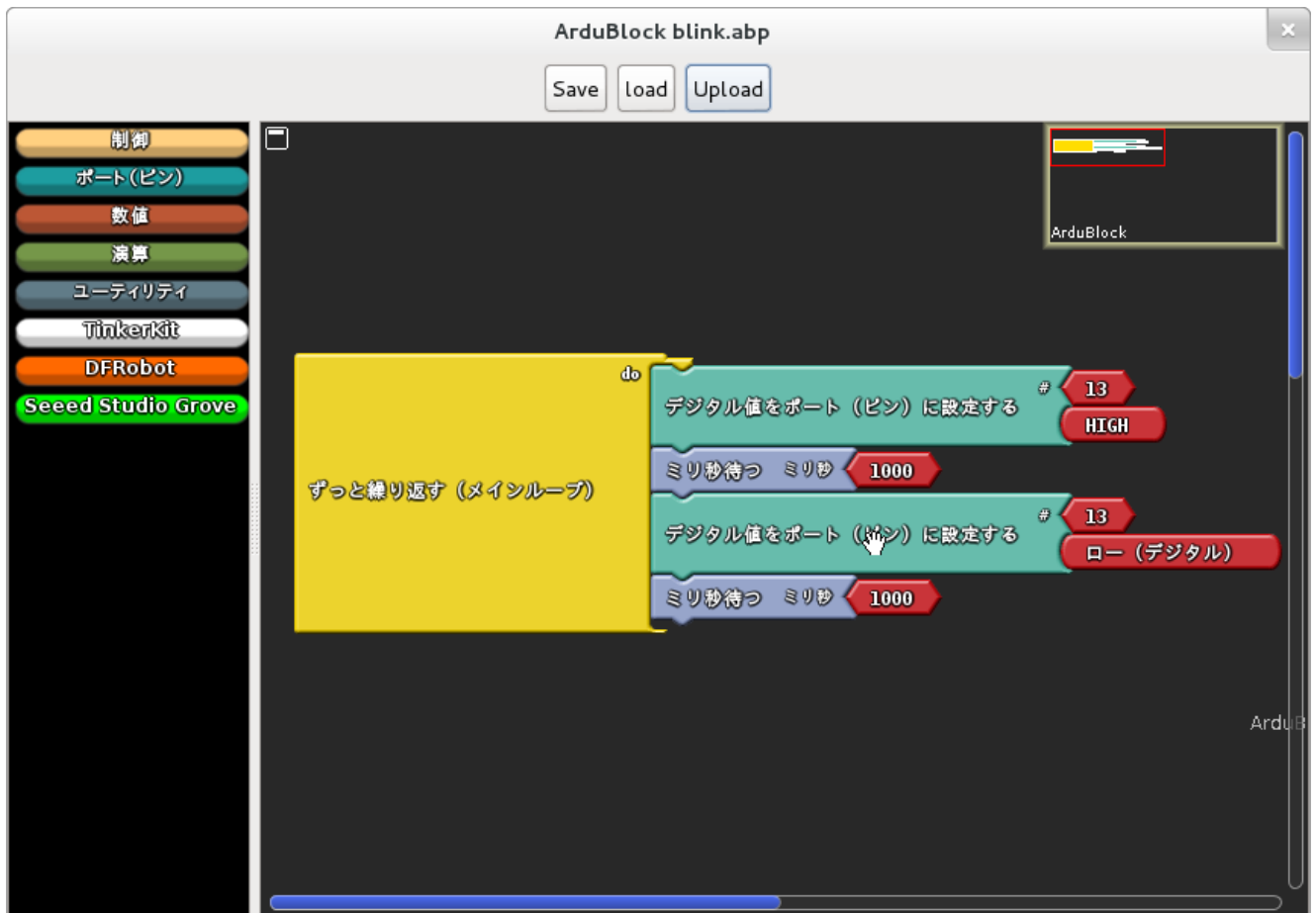
```
$ mv ~/Downloads/ardublock-all.jar tools/ArduBlockTool/tool/
```

Arduino を起動して, [ツール] -> [ArduBlock] として起動する。

```
$ ./arduino &
```







### 参考

"日本Androidの会秋葉原支部ロボット部 第11回勉強会 : ATND" <http://atnd.org/events/36354>

Evernote に保存

Evernote はコンピュータ、タブレット、携帯電話と Web ブラウザを使用して日常生活に関する様々なことを簡単に記録できるようにします。





*Let's start electronics making with Arduino compatible kit "ChibiduinoICHI"*

## ちびでい～の壺ではじめる はじめてのでんしこうさく

内容物：  
ちびでい～の壺  
書き込み装置  
ブレッドボード  
タクトスイッチ  
LED  
抵抗  
ジャンパケーブル  
サーミスタ  
CDS  
USBケーブル

AKBROBOT 発行所／日本Androidの会秋葉原支部ロボット部・ジャパン