

# 軽量Ruby普及・実用化推進ネットワーク 技術交流会

～ mruby × IoT ～

2015/10/24

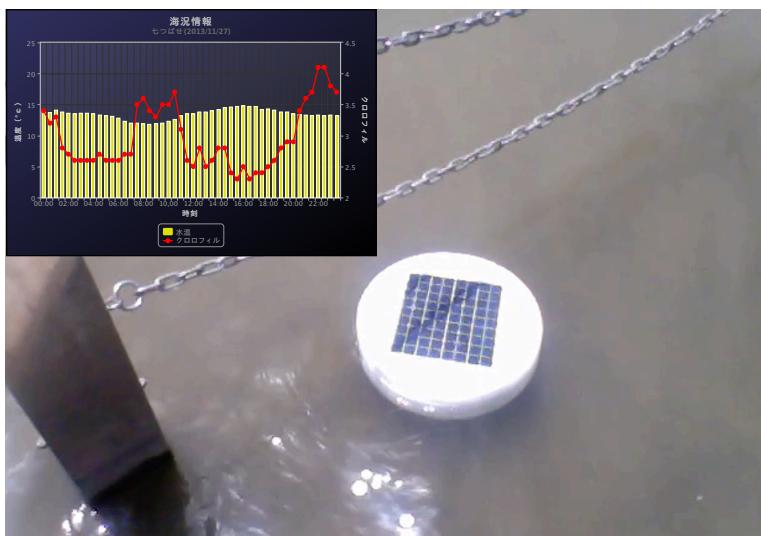
福岡県Ruby・コンテンツビジネス振興会議

# 自己紹介

2010年 Manycolors株式会社を設立後、遺伝子解析Webサービスやハードウェアの開発を手がける。

2013年 mruby プロトタイピングプラットフォーム「enzi」をSCSK九州と共同開発。

2014年 enziを使った海洋及びサービスで第6回フクオカRuby 大賞・県知事賞受賞。



hsa:3836 | Antigen

antigen.manycolors-inc.com/protein/hsa:3836

KEGG GENES Search

Antigen

hsa:3836

Ortholog Paralog Motif Hydropathy Plot Transmembrane Region

Ranking Configuration

1. KIEFLQSHENQEYQKAFDIEHYFG  
2. LLTVMDSKIVQVALNGLENI  
3. TOKFRKLISKEP  
4. KEAAWATNAT  
5. EMRRRREEEG

Hydropathy Plot

Threshold: 0.0

Hydropathy Plot

Transmembrane Region

PDB

PDB:2JDO(RCSB,NCB)

[ILE]373-B CA #5536

mol

検索: c 次を検索 前を検索 すべて強調表示 大文字/小文字を区別

This screenshot of a web-based protein analysis tool shows the entry for hsa:3836 (Antigen). It includes a sequence viewer with motifs highlighted in red, a hydropathy plot, and a 3D surface model of the protein structure with a transmembrane region highlighted in red.

## 本セミナーについて

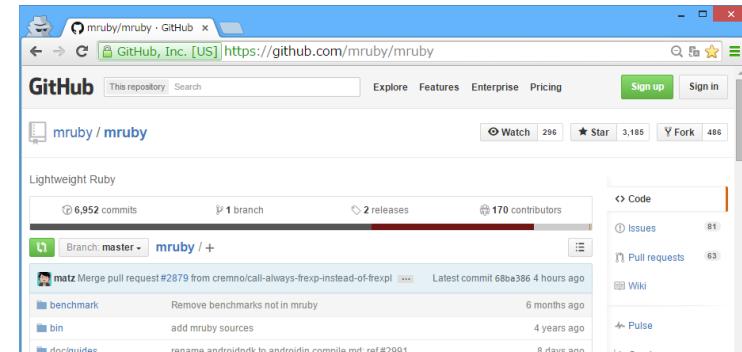
- mrubyの基礎
- 電子部品の基礎的な使い方(LED, 温度センサ)
- インターネットとデバイスの連携(IoT, M2M)
- プロトタイピング

# mrubyの基礎

「Ruby」を省リソース化した「mruby」の概要・特徴

# mrubyとは

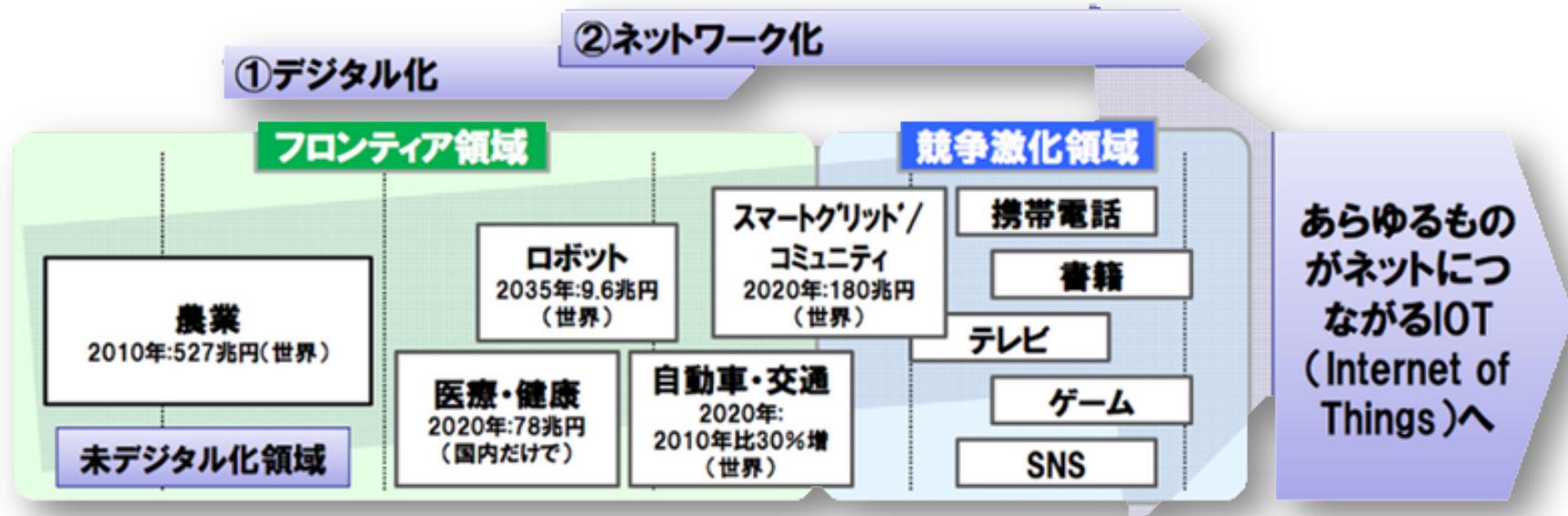
- Ruby良さをそのままに軽量化した軽量版Ruby
  - RAMサイズ100KB程度で動作可能
  - 組込み機器にも搭載可能
- 経済産業省「地域イノベーション創出研究開発事業」の研究成果として2012年4月に「mruby」を公開
- 商用利用しやすいMITライセンスのOSS



<http://github.com/mruby/mruby>  
にてソース公開

# IoT時代の新言語 - mruby

- もののインターネット (IoT)  
→ 全てのものがネットワークに繋がっていく
- 組込み分野もフロンティア領域を重点分野として新しい市場が形成されていく  
→ これに対応した組込み開発が必須になっていく



経済産業省  
組込み開発の今後 資料より

## mrubyの特徴

- ISO, JIS規格のRubyに準ずる言語仕様
  - 本家Rubyと同様に使える
- コンパクトな処理系「mruby VM」
  - mrubyコンパイラが出力するバイトコードを実行
  - VMさえ動作すればどんな環境でも動作可能  
Windows, Mac, Linux, ITRON, Android, iOS ...
- C/C++言語との高い親和性
  - 組込みシステムの資産が再利用可能
  - アプリケーションにmrubyを組込み可能

# 補足： Rubyとmruby

Ruby	mruby
インタプリタ言語	インタプリタ言語／コンパイラ言語
C言語モジュールの呼び出しが可能	C言語モジュールの呼び出し、C言語からの呼び出しが可能
実行時にRubyGemsによる機能拡張が可能	ビルド時にmrbgemsによる機能拡張が可能 ※mrbgemsとRubyGemsとは互換性なし
Integer / Integer → Integer	Integer / Integer → Integer または Float ※割り切れなければ Float に変換
非常に大きな整数値はBignumで扱われる	非常に大きな整数値はFloatで扱われる
正規表現を標準装備	正規表現はオプション ※複数から選択可能
ASCII, Shift_JIS, UTF-8など様々な文字コードが利用可能	利用可能な文字コードはASCII, UTF-8のみ
バイナリ配布(ソースコードも入手可)	ソースコード配布 ※容易にカスタマイズ可能

# mrubyを支えるコミュニティ

- オープンソースコミュニティ
  - 3,000人超のフォロワー
  - 150人以上のコントリビュータ
  - 6,900件を超える改良・修正



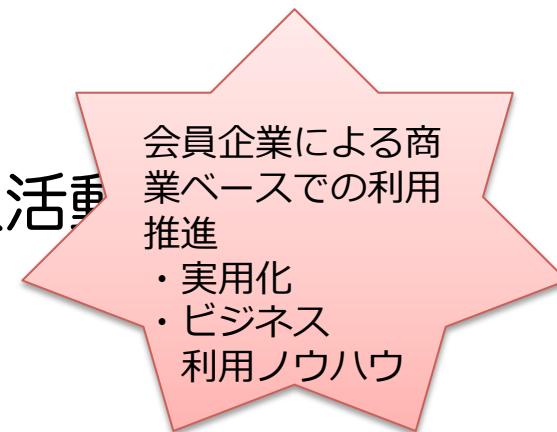
<https://github.com/mruby/mruby> (mruby公開サイト)



オープンソースとして成長  
• Creative  
• Powerful

- NPO軽量Rubyフォーラム
  - ワーキンググループによる活動
  - セミナー・講演会等でのmruby広報活動
  - 福岡県からの支援

<http://forum.mruby.org> (軽量Rubyフォーラムサイト)



会員企業による商業ベースでの利用推進  
• 実用化  
• ビジネス利用ノウハウ

**会員募集中 !!**

**enzi**

- mruby rapid prototyping platform -

enziの概要と基本的な使い方

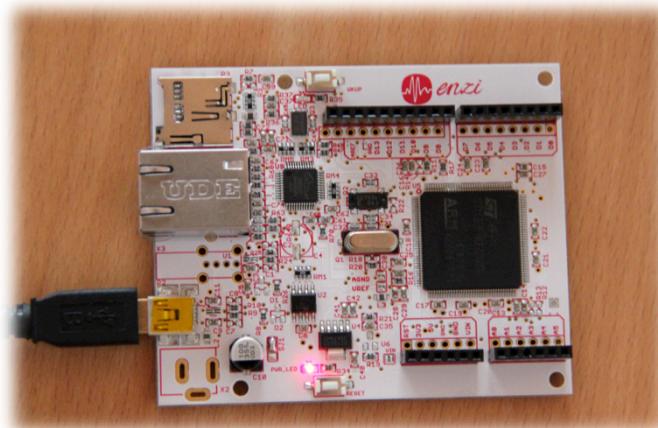
# enziについて

## enzi – mruby rapid prototyping platform –

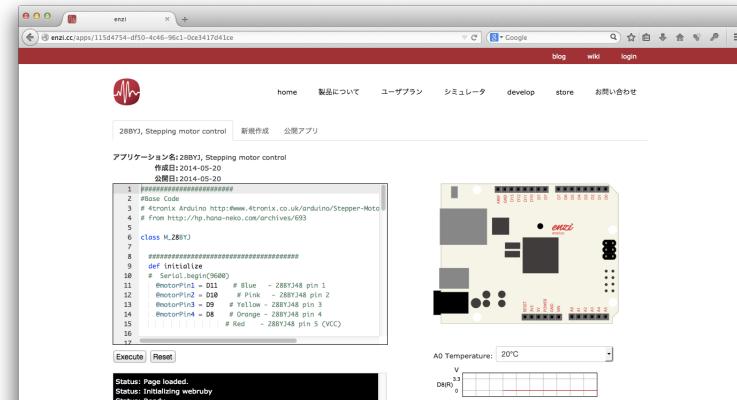
mrubyの活用を検討している企業の技術者、mrubyの習得を目的としている学生・技術者向けに

- mrubyが組み込まれたボード（enziボード）
- mrubyのアプリ開発・簡易動作シミュレーションの出来るWeb環境（enziシミュレータ）

というmruby教育・検証の為の2つの機能を統合した新しい形のシステムです。



enziボード

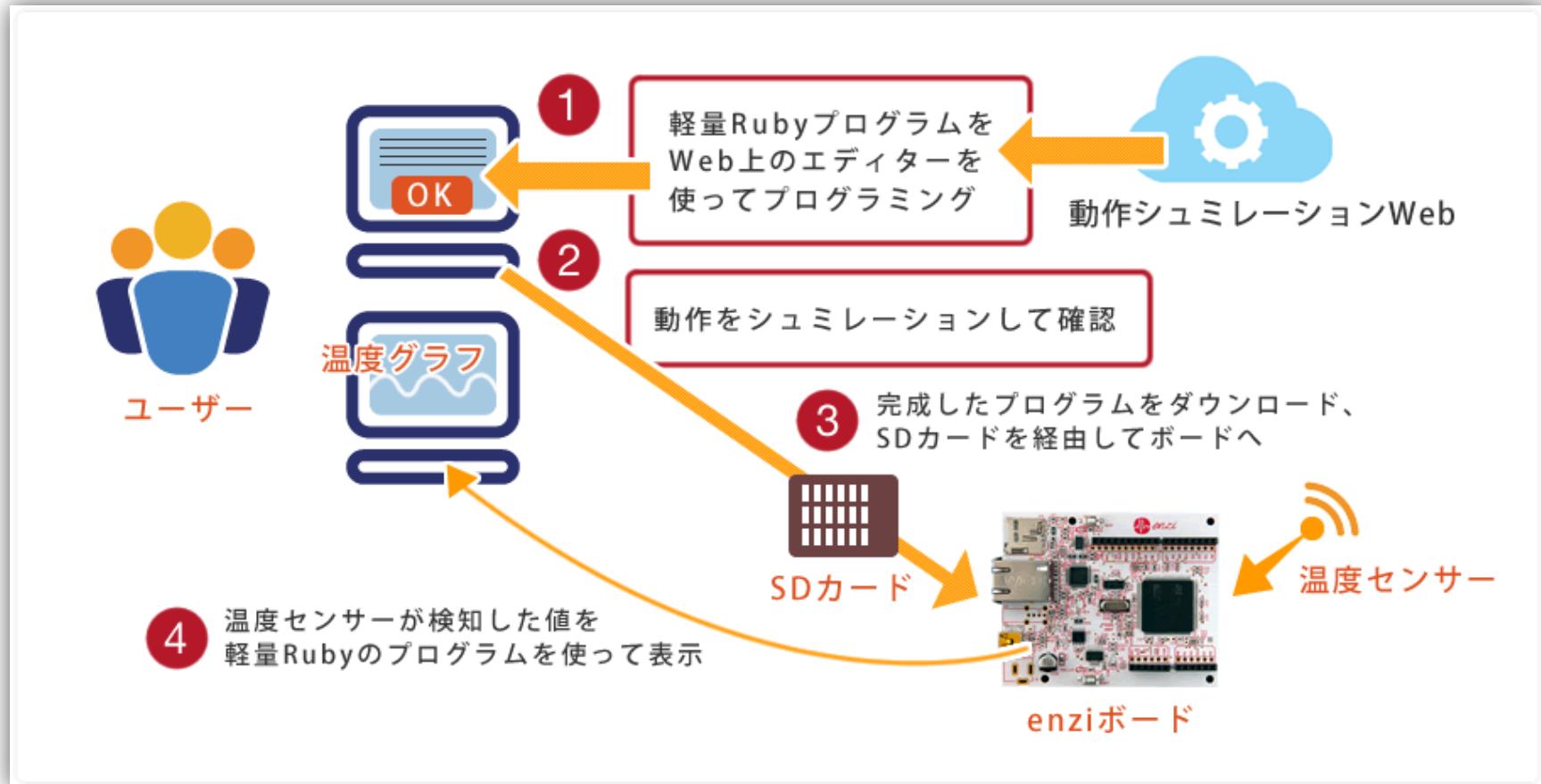


enziシミュレータ

## enziの特徴・利点

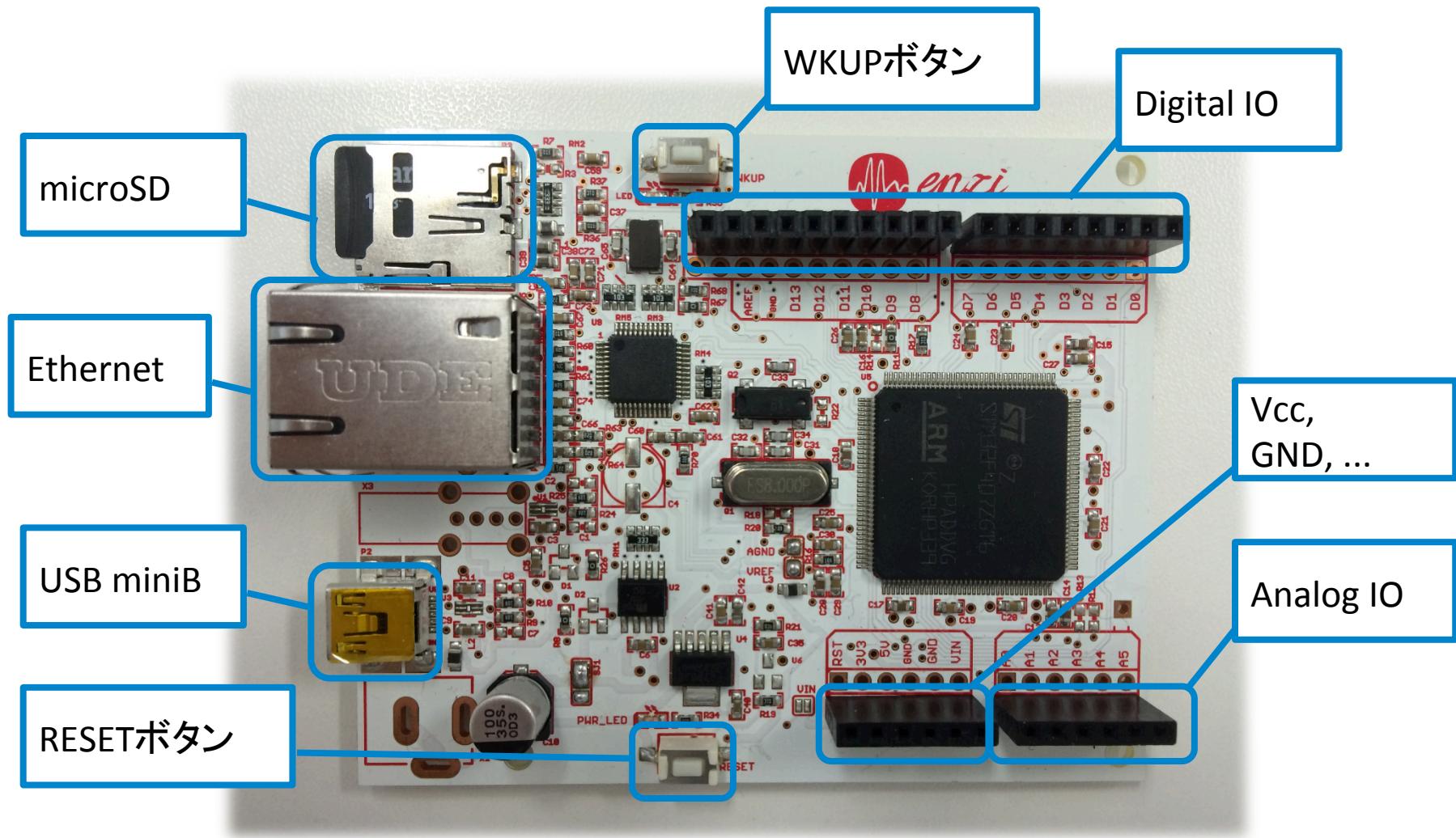
- mrubyを簡単に試せる
  - Webブラウザでのmrubyプログラミング
  - コンパイル結果はmrubyボードで動作
  - mirbによる対話型でのmruby実行も可能
- mrubyでものを動かせる
  - GPIO, SPI, I<sup>2</sup>C, UART等をmrubyで制御
  - Ethernetポートを利用したネットワーク機能

# enziの利用イメージ



enziボードに接続した温度センサの値を読むmrubyプログラムを動作させる手順例  
(enzi.cc より引用)

# enziボード外観



## enziのセットアップ

- 情報は

<http://enzi.cc/iot.html>

にもまとめてあります

## enziのセットアップ

- ・ ソフトウェアインストール (初回のみ)
- ・ enziサイトへのログイン
- ・ ライセンス登録 (初回のみ)
- ・ enziシミュレータでのシミュレーション
- ・ プログラムのコンパイル
- ・ enziボードでのプログラム実行

## ソフトウェアのインストール

- VCP（仮想COMポート）ドライバのインストール  
<https://goo.gl/mIiikE>

- ターミナルソフト（CoolTerm）のインストール  
<https://goo.gl/VZI9WM>

<注意>

- ※ 上記URLは**大文字／小文字の区別あり**
- ※ 使用環境に合ったものをインストールすること
- ※ Windows用のVCPドライバは Readme を参照のこと

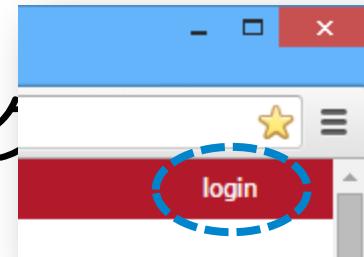
## enziサイトへのログイン

1. enziサイトを開く

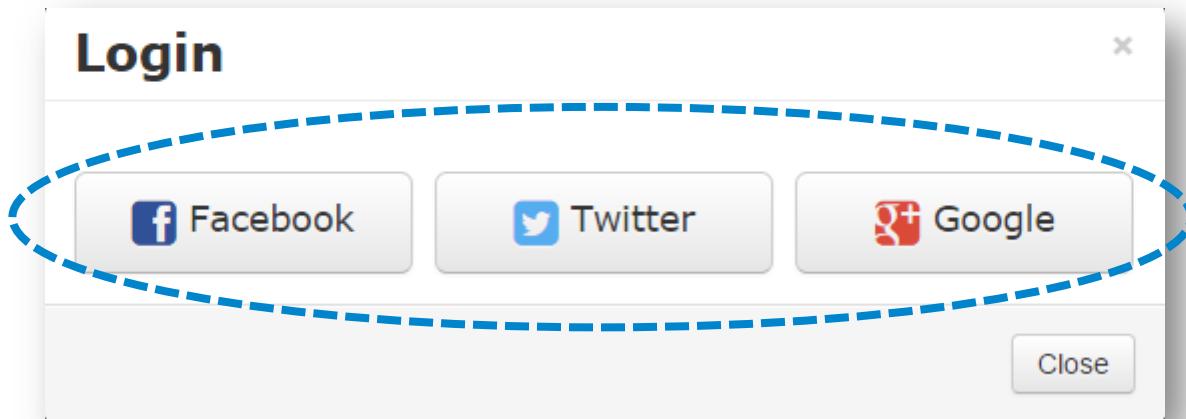
<http://enzi.cc>

※ Internet Explorer以外のブラウザを推奨

2. 画面左上のloginのリンクをクリック



3. お持ちのアカウントを使ってログイン



# enziライセンス登録

4. 画面右上の「登録」をクリック



5. enzi同梱の「Quick Start Guide」に印字されているライセンスキーを入力し、「送信する」をクリック

enziボード登録

S/N: XXXXXXXXXXXXXXXX  
MAC: 04CF25XXXXXX  
KEY: XXXX-XXXX-XXXX-XXXX

ライセンスキー:

送信する

以上で、ライセンス登録は完了です。

# enziシミュレータでのシミュレーション

画面上部の「シミュレータ」のリンクをクリック

The screenshot shows the enzi Simulator interface. At the top, there is a navigation bar with links: 製品について, ユーザプラン, シミュレータ (which is highlighted with a blue dashed oval and has a large blue arrow pointing down to it), ドキュメント, スタートアップガイド, and Q&A.

**テキストエディタ**: A code editor window containing the following Arduino-style pseudocode:

```
1 p "Hello World!"
2
3 analogWrite(D9, 100)
4 status = 0
5
6 loop do
7   (0..25).each do |i|
8     status ^= 1
9     digitalWrite(D8, status)
10    analogWrite(D8, i*10)
11    delay(1600)
12  end
13 end
14
```

**コンソール**: A terminal window displaying the following status messages:

```
Status: Page loaded.
Status: Initializing webruby
Status: Ready.
```

**enziシミュレータ画面**: This section contains three parts:

- A schematic diagram of an Arduino Uno board with various components and connections.
- A graph titled "A0 Temperature: 20°C" showing a constant value of 3.3V over time.
- A graph titled "簡易シミュレータ" showing two waveforms: D8(R) and D10(B). D8(R) is a constant 3.3V, and D10(B) is a sawtooth wave starting at 0V and decreasing linearly from -10 to 0.

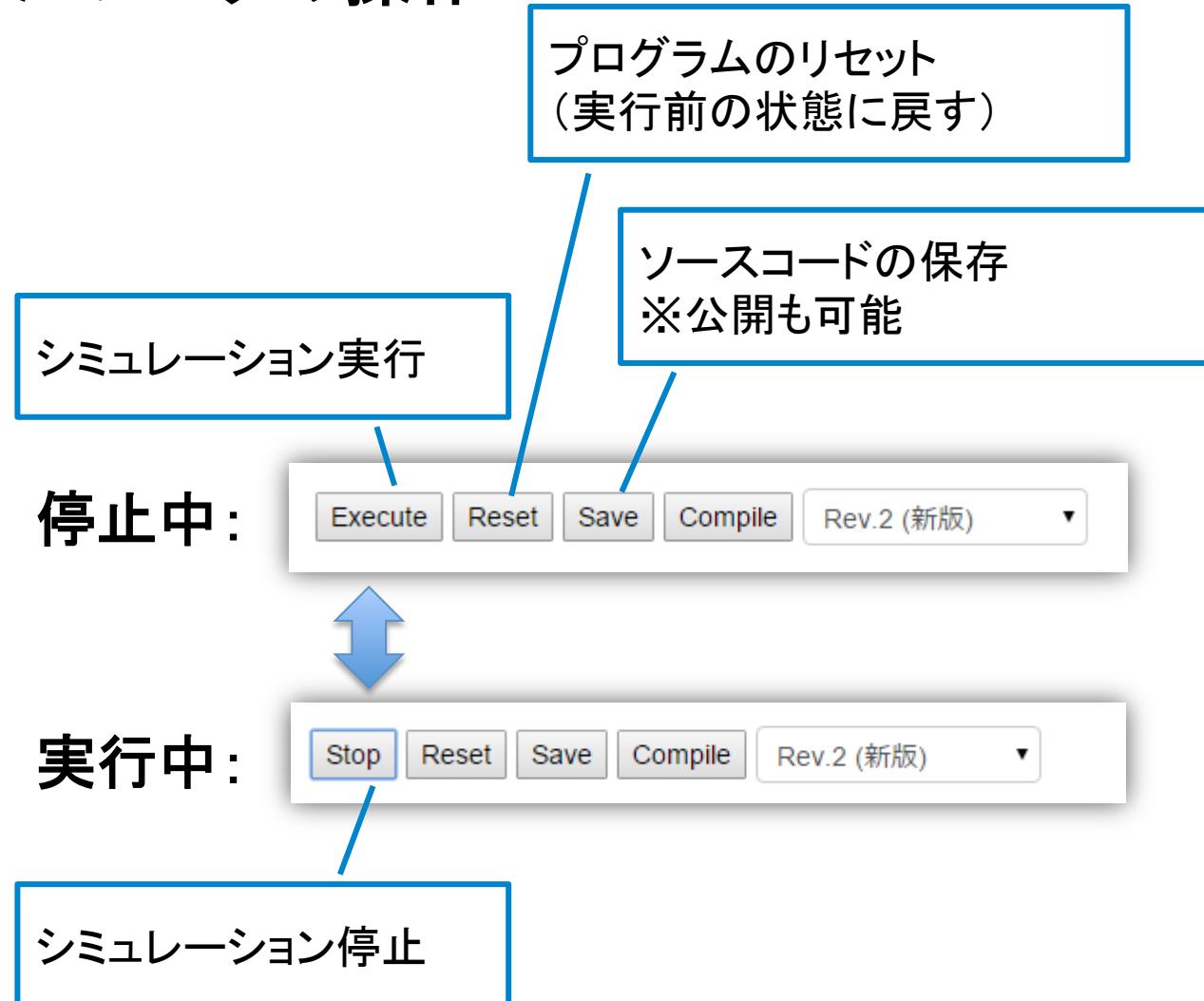
# enziシミュレータでのシミュレーション

The screenshot shows the enzi simulator interface with several key components:

- Code Editor (Top Left):** Displays mruby code for a "Hello World" application with analog and digital output logic.
- Execute Button (Bottom Left):** A red arrow points from this button to a callout box explaining the execution process.
- Output Terminal (Bottom Left):** Shows status messages indicating page load, webruby initialization, and system readiness.
- Simulator Board (Top Right):** A schematic diagram of the enzi board with various pins labeled (e.g., AREF, GND, D13, D12, D11, D10, D8, D7, D6, D5, D4, D3, D2, D1, D0).
- Execution Steps (Callout Boxes):**
  - ① mrubyのコードを書く  
ここではサンプルコードをそのまま使用
  - ② Executeボタンでシミュレーション実行
- Waveform Plot (Bottom Right):** Displays three waveforms for pins D8(R), D9(G), and D10(B) over time t, with an A0 Temperature setting of 20°C. The waveforms show constant values of 0V, 3.3V, and 5V respectively.
- Output Results (Callout Boxes):**
  - p, puts, print など出力結果が表示される。
  - D8, D9, D10への出力が波形で表示される。

# enziシミュレータでのシミュレーション

## enziシミュレータの操作



# サンプルコードの説明

```
1 p "Hello World!" — 文字列を表示
2
3 analogWrite(D9, 100) — D9ポートにアナログ値100を出力
4 status = 0
5
6 loop do
7   (0..25).each do |i|
8     status ^= 1
9     digitalWrite(D10, status)
10    analogWrite(D8, i*10)
11    delay(1600)
12  end
13 end
```

無限ループ

0~25の範囲でループ

D10ポートにstatus(0または1)を出力

D8ポートに i の10倍の値を出力

1600ms(1.6秒)処理を停止する

# プログラムのコンパイル

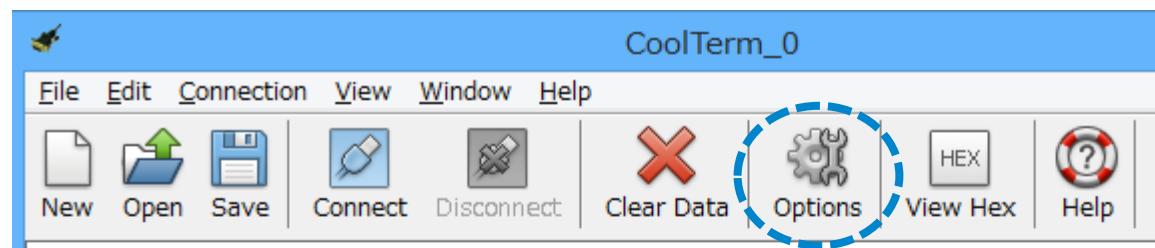
## 1. Compileボタンをクリック



2. ダウンロードされる実行モジュール (enzi.ezb)  
を  
PCに保存
3. enzi.ezbをmicroSDカードにコピー  
※ ファイル名は (enzi.ezb) は変更しないこと  
※ microSDのルートディレクトリに保存すること

## enziボードでのプログラム実行

4. microSDカードをenziボードに装着  
※ カチッと音がするまで差し込む
5. CoolTermを起動
6. enziボードとPCをUSBケーブルで接続
7. CoolTermの「Option」をクリック

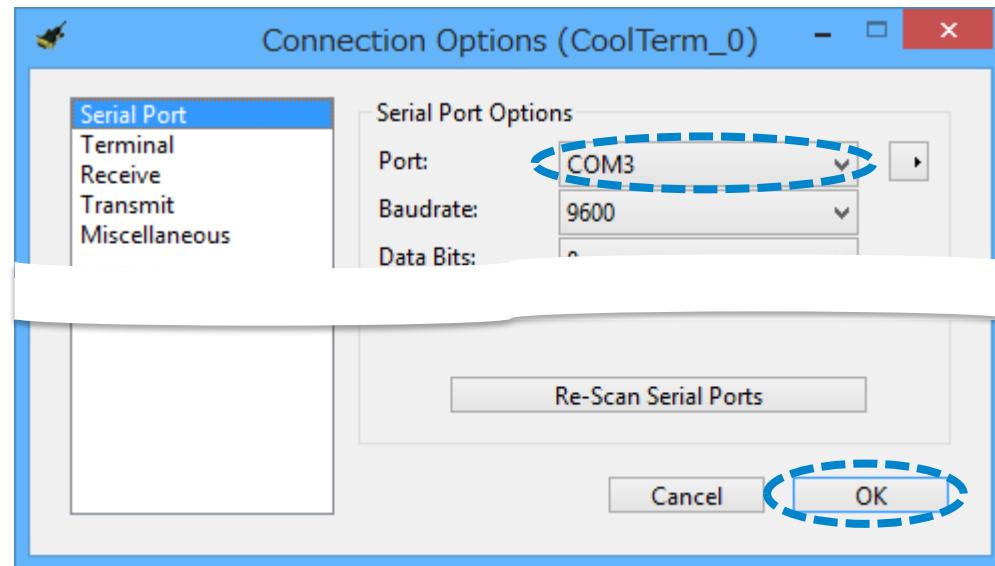


# enziボードでのプログラム実行

## 8. Portを選択してOKをクリック

Windowsの場合："COMx"

Macの場合："usbmodemXXXX"

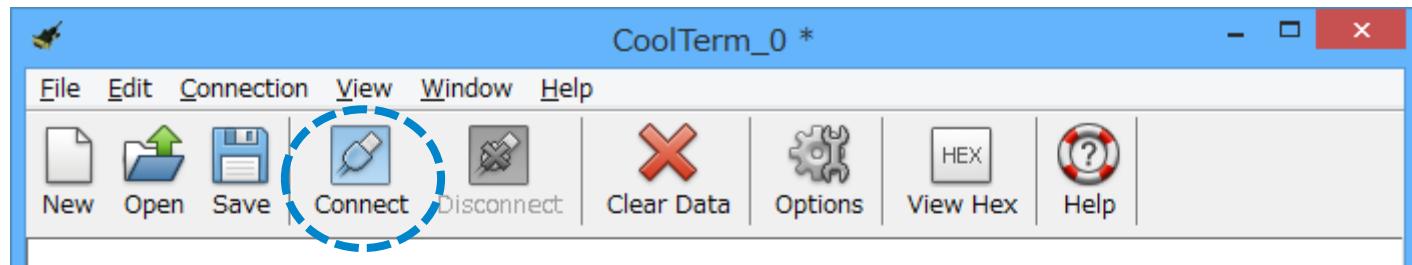


- ※ Portが選択できない場合はenziの仮想COMポートが正しく認識されていません。  
VCPドライバのインストールを確認して下さい。

## enziボードでのプログラム実行

9. enziボードのリセットボタン（赤LEDが点灯している側のボタン）を押下

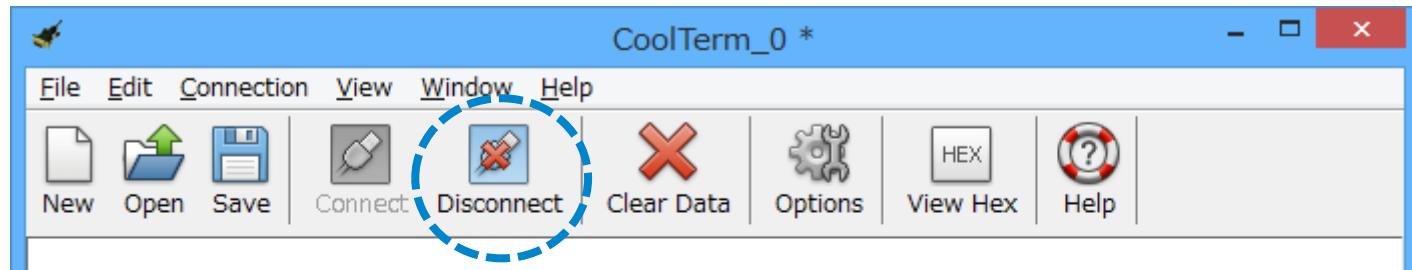
10. CoolTermの「Connect」をクリック



※ CoolTermの画面に "Hello World!" が表示されればOK

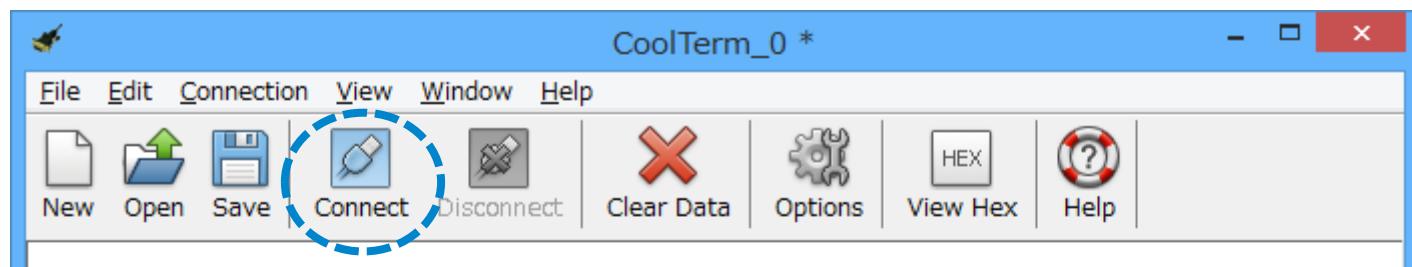
## 補足：enziボードでのプログラム再実行

1. CoolTermの「Disconnect」をクリック



2. enziボードのリセットボタンを押下

3. CoolTermの「Connect」をクリック



## 対話形式でmrubyを実行する

enziではmirb (mruby版のirb) による対話形式でのmruby実行が可能

対話モードでの動作手順

1. enziボードからmicroSDカードを抜く
2. CoolTermを起動

# 対話形式でmrubyを実行する

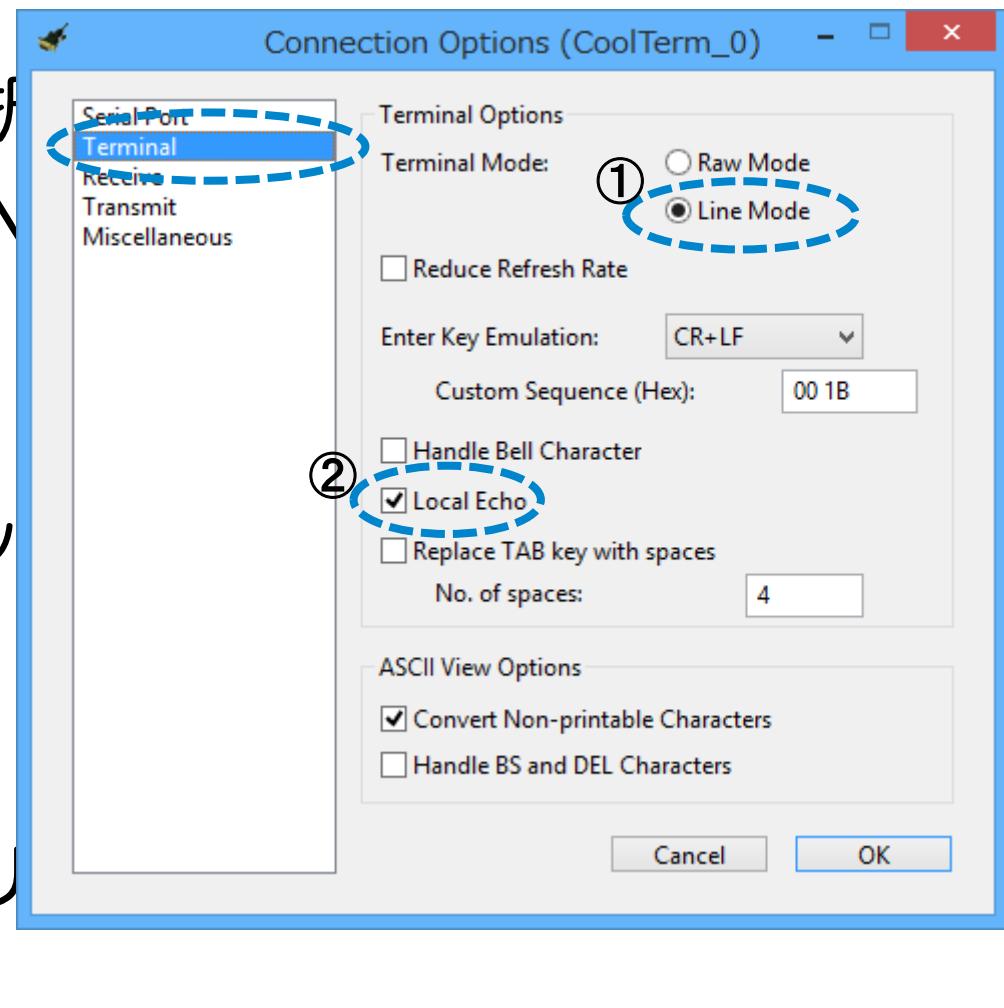
1. オプション画面で「Terminal」を選択し下記を変更

- ① Line Modeを選択
- ② Local EchoをON

2. OKをクリック

3. enziボードのリセットボタンを押下

4. CoolTermの「Connect」をクリック



# 対話形式でmrubyを実行する

## mirbの実行

The screenshot shows the CoolTerm window titled "CoolTerm\_0 \*". The menu bar includes File, Edit, Connection, View, Window, and Help. The toolbar contains New, Open, Save, Connect, Disconnect, Clear Data, Options, View Hex, and Help buttons. The main terminal window displays the following text:

```
Network Configuration Information
-----
MAC: 04:CF:25:04:86:B5
IP : 192.168.250.250
SN : 255.255.255.0
GW : 0.0.0.0
DNS: 0.0.0.0
enzi - mruby rapid prototyping platform
Copyright(c) 2013-2014 mruby developers, FUKUOKA CSK CORPORATION & Manycolors, Inc.
Firmware : v1.0.1
enzi start.
initialize mruby...OK
application(/enzi.ezb) cannot open.
mirb - Embeddable Interactive Ruby Shell

This is a very early version, please test and report errors.
Thanks :)
```

The bottom part of the terminal shows a command-line interface with the prompt ">". Below the terminal is a status bar indicating "COM3 / 9600 8-N-1" and "Connected 04:20:24". A legend at the bottom right shows the status of serial port pins: TX (green), RX (green), RTS (green), CTS (green), DTR (green), DSR (green), RI (green), and DCD (green). Two red callout boxes point to specific areas: one points to the "mirb - Embeddable Interactive Ruby Shell" message with the text "起動メッセージでmirb の起動を確認", and another points to the command line with the text "mrubyスクリプトを入力".

## 対話形式でmrubyを実行する

mirbでmrubyを実行してみましょう。

```
> 1+2+3  
=> 6  
> [4,6,-1,0,2].sort  
=> [-1, 0, 2, 4, 6]  
> 5.times { |i| puts i}  
0  
1  
2  
3  
4  
=> 5
```

## 対話形式でmrubyを実行する

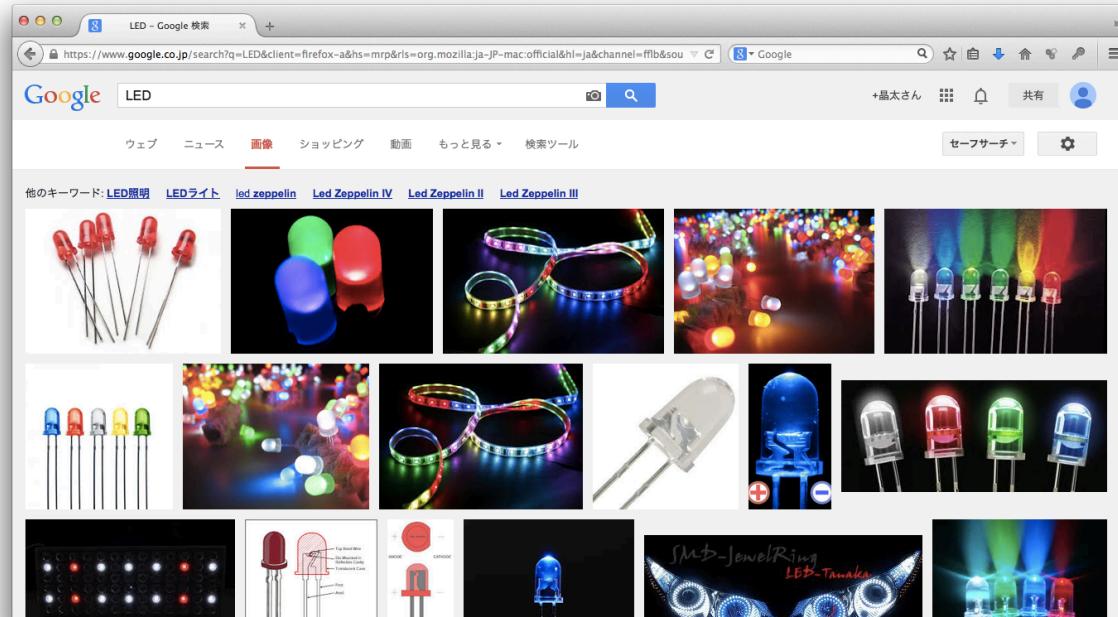
1. 長いクラス定義などはSDに書き込み
2. 実行はmirbで実行すると便利です

# IoTのための電子回路基礎

電子部品とその基礎的な使い方

# enziでLEDを光らせる①

- enziはArduinoとレイアウト及び多くの機能が同じな外部IOを用意してあります
- 適切に外部電子部品等を接続することで制御/観測が可能です
- 最も簡単な部品としてLEDを光らせてみましょう



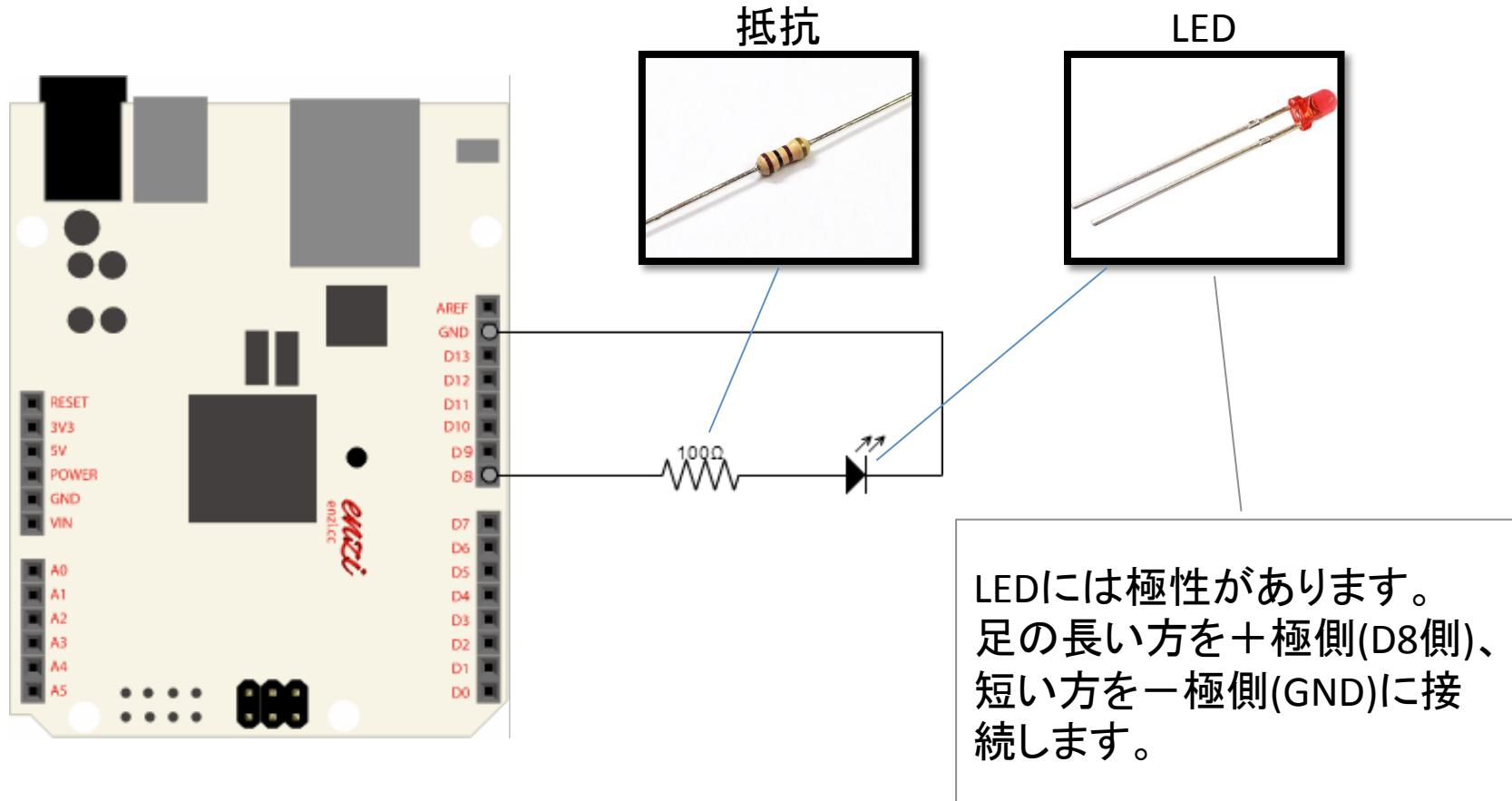
# enziでLEDを光らせる②

- LEDとは
  - 発光ダイオードの略
  - 省エネ照明などに利用
  - 片方向にしか電流が流れません
- 簡易的な回路で実験
  - 本当はトランジスタやドライバICを用いる



# enziでLEDを光らせる③

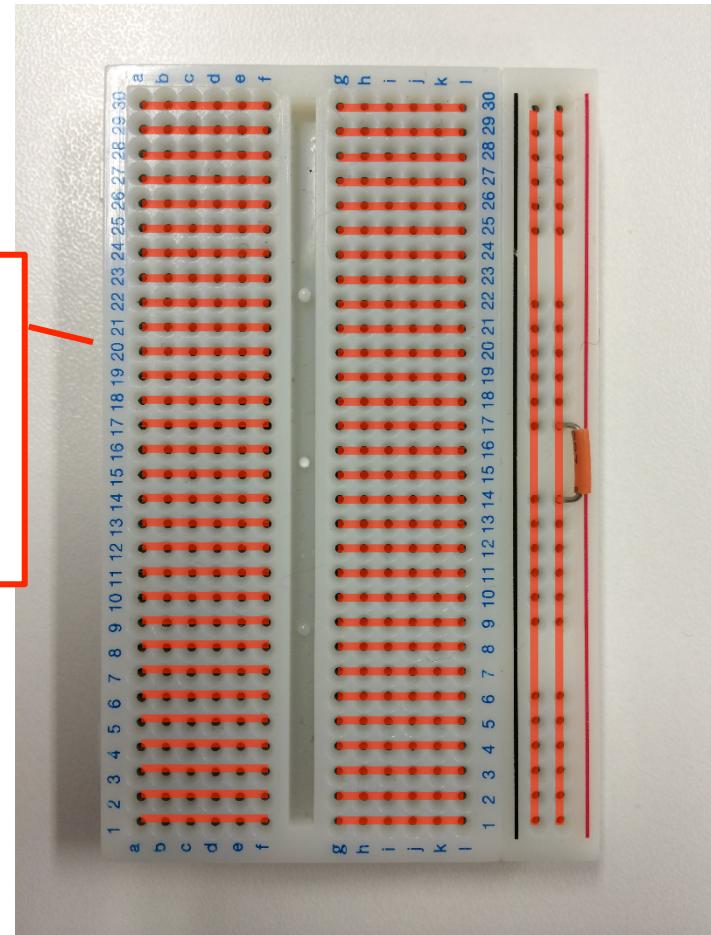
## LEDを光らせるための回路図



# enziでLEDを光らせる④

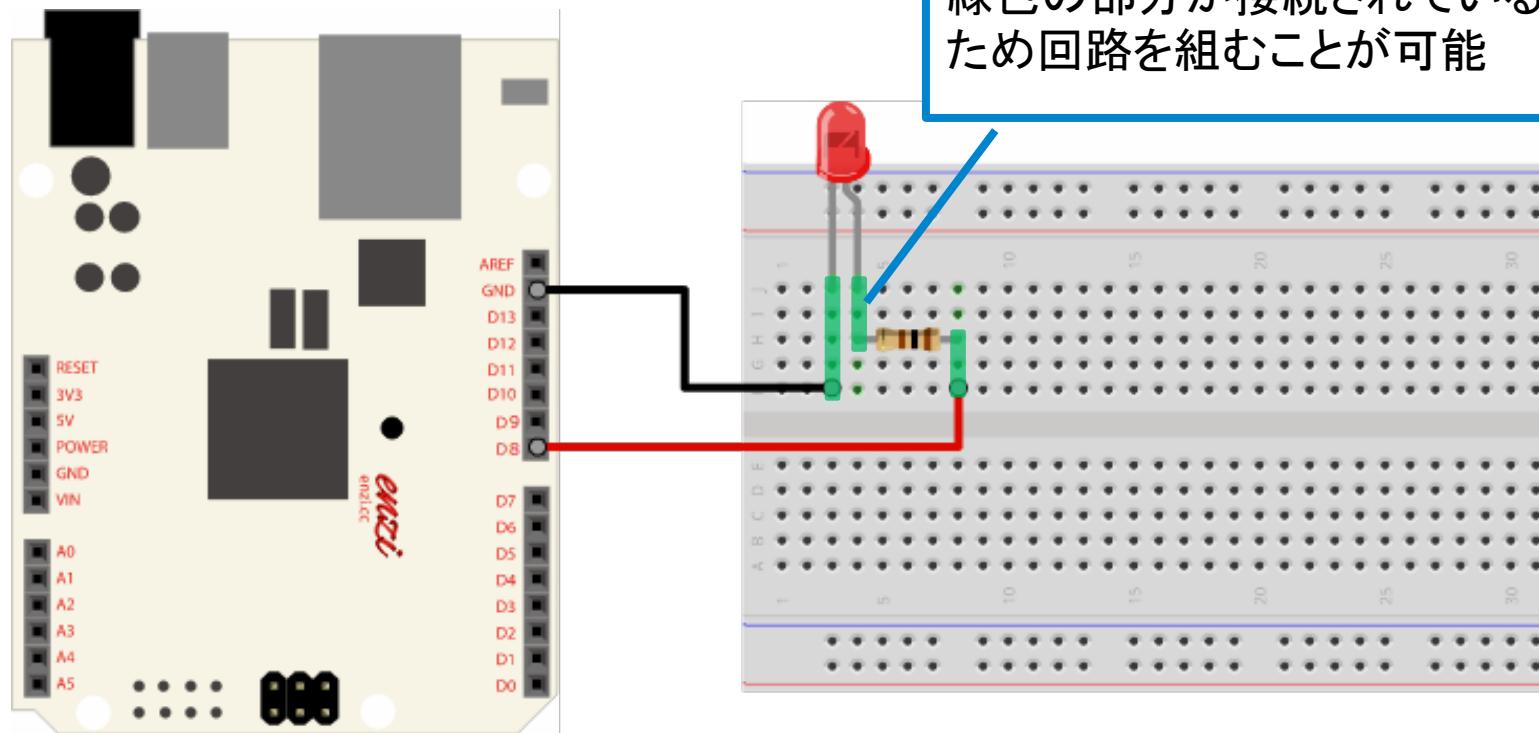
## ブレッドボードで回路を作る

ブレッドボードは赤線で示す穴同士が金属で接続されています。  
その仕組を利用して、電子部品の端子を穴に差し込むことで回路を作ることができます。



# enziでLEDを光らせる⑤

## ブレッドボードを使ったLED回路の例



## enziでLEDを光らせる⑥

1. enziサイトのサンプルプログラムを動作させる  
→ LEDが徐々に明るくなる

2. D8をD10に接続し直す  
→ LED ON/OFFを繰り返す

```
loop do
  (0..25).each do |i|
    status ^= 1
    digitalWrite(D10, status)
    analogWrite(D8, i*10)
    delay(1600)
  end
end
```

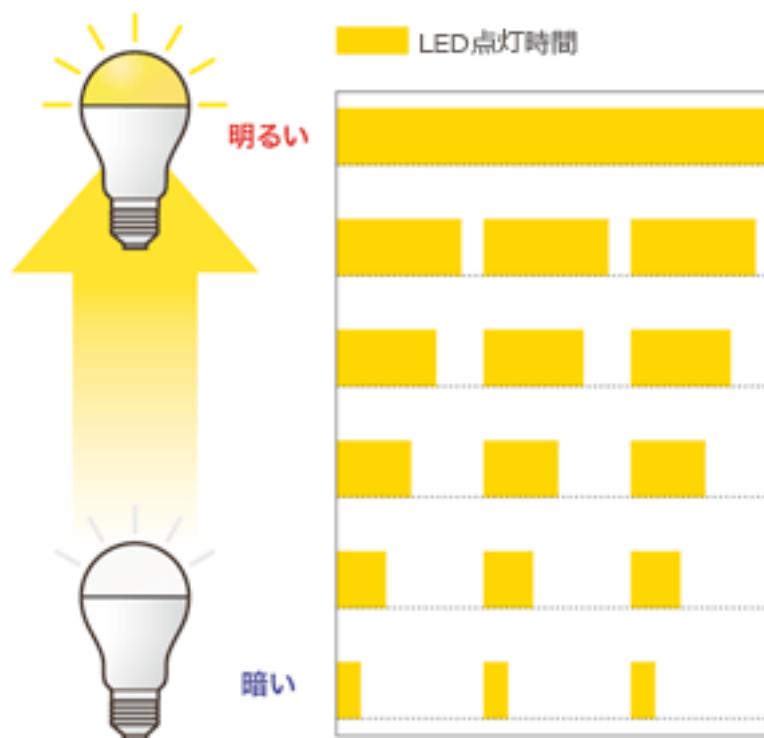
## enziでLEDを光らせる⑦

- digitalWrite: IOのON/OFF
  - digitalWrite(対象ピン, 0か1)
- 組み合わせて色を変えてみましょう
- led.rb

```
digitalWrite(D8, 1) #R, ON
```

# enziでPWM①

- PWMとは
  - 半導体を使った電力を制御する方式の1つです。オンとオフの繰り返しスイッチングを行い、出力される電力を制御します。
  - Pulse Width Modulation



## enziでPWM②

- AnalogWrite: PWM出力
  - AnalogWrite(対象ピン, 0-255)
- loopを使いなめらかに色を変えてみましょう
- pwm.rb

```
analogWrite(D8, 10) #R, 10
```

# enziでIoT

enziを使ったIoTのプロトタイピング

# enziでネットワーク接続

1. microSDにenzi.cfgというファイルを作成し、下記の内容を保存

```
ipaddr=172.16.31.XXX  
dns=8.8.8.8  
subnet=255.255.0.0  
gateway=172.16.0.1
```

個別に割り振るアドレス(1~20)を  
入力して下さい。

2. microSDをenziボードに装着し、LANケーブルでenziをHUBに接続
3. enziボードを起動  
※ CoolTermで起動時の表示（IPアドレス等）を確認
4. PCからenziに対してpingを実行（接続確認）

```
ping 172.16.31.XXX
```

## enziでhttp

- SimpleHttpを使いyahoo.co.jpに接続
- get出来ることを確認
- get\_yahoo.rbを参照

# 午前の復習

---

- mrubyの概要
- enziの基礎
- LEDを光らせる(機器の制御の基礎)
- PWMでLEDを制御
- httpアクセス

```
loop do
    処理
end

delay(50) #delay 50ms
```

## お願い

---

- [info@manycolors-inc.com](mailto:info@manycolors-inc.com)

にタイトル”mruby-iot”で空メールを送って下さい。

午後にSlackを使うためグループに追加します。  
(本日終了後改めて削除させていただきます)

## enziと温度センサ

- センサによる観測
- 実際には変換テーブルや補正が必要です
- クラスを用意してますのでtemp\_bar.rbを見て下さい  
※切り欠きがあるボードはtemp\_bar\_2.rbを見て下さい

## enziとweb api

- センサによる観測
  - 実際には変換テーブルや補正が必要です
  - web\_api.rbを見て下さい
- 
- 測定した温度をweb api経由でSlack(またはTwitter)に投稿します。
- 
- IDにはenziのIPアドレスの末尾(0-255)を入れ、識別できるようにして下さい
- 
- このweb apiのソースもGitHubで閲覧可能です

## enziを簡易Webサーバにする

- enziはサーバにもなります
- hello\_world\_ethernet.rbを見て下さい
- Webブラウザでenziのアドレスにアクセス

# Google chartを表示する

- 温度測定した結果をgoogle chartで表示します
- google\_chart.rbを見て下さい
- Webブラウザでenziのアドレスにアクセス

## enziの応用①

- 測定温度によってLEDの光り方を変えてみましょう
- ネット上の情報(天気など)と併せてSlackに投稿してみましょう
  - ex. 晴れ(23°C)など
- センサには気圧計もついているので使ってみましょう
  - ex. 気圧もgoogle chartに表示

# まとめ

---

- mrubyの基礎
- 電子部品の基礎的な制御と観測
- インターネットとデバイスの連携(IoT, M2M)
- プロトタイピング

enziユーザーグループ

<https://www.facebook.com/groups/321739777988177>