

mruby-on-ev3rt+tecs menu + bluetooth

白田聖人（大阪大学）

最終更新日：2019/ 5/ 28

目次

- menu + bluetoothについて
- 環境構築
- mrubyのビルド
- 開発方法
 - EV3にアプリを取り込む
 - バイトコードをSDに保存
- 実行
 - 起動方法
 - モードの切り替え
 - Runモード
 - Deleteモード
 - Bluetooth受信モード

menu + bluetoothについて

3つのモード

1. [Run] SDに保存されたバイトコードをLCD上で選択し実行
2. [Delete] SDに保存されたバイトコードをLCD上で選択し削除
3. [Bluetooth] Bluetooth経由で飛んできたバイトコードを実行し、保存

メリット

- わざわざ何度もSDを抜き差ししなくても複数のrubyアプリケーションを実行できる。
- Bluetoothで飛んできたバイトコードは保存して次回以降も使用できる。

環境構築

- Windows10
 - (Windows7、Windows 8、Windows 8.1は動作未確認)
- Cygwinインストール
 - ruby
 - GNU Make
 - bison
- クロスコンパイラ
 - arm-none-eabi-gcc.exe (GNU Tools for ARM Embedded Processors)
https://launchpad.net/gcc-arm-embedded/4.8/4.8-2014-q3-update/+download/gcc-arm-none-eabi-4_8-2014q3-20140805-win32.exe
- mkimage
 - Windows用バイナリはパッケージに同梱
- Cygwin及びクロスコンパイラのインストールはEV3RTの開発環境構築を参照してください。
 - http://dev.toppers.jp/trac_user/ev3pf/wiki/DevEnvWin

環境構築

- Tera Term
 - Bluetoothを用いてアプリケーションを使用する場合に使用
- (Bluetooth Stack for Windows by Toshiba)

Bluetoothの接続手順

- 対応機器およびBluetooth接続手順はEV3RTの開発環境構築を参照してください。
 - Windows
 - http://dev.toppers.jp/trac_user/ev3pf/wiki/BluetoothWin

環境構築：パスの通し方

- クロスコンパイラをインストールしたディレクトリにPATHを通す
 - C:¥Program Files (x86)¥GNU Tools ARM Embedded¥4.8 2014q1¥bin
 - ※フォルダ名は、クロスコンパイラのバージョンごとに変わります。
- arm-none-eabi-gccへパスが通っているかを確認

```
takuro@Lenovo-PC:~  
$ arm-none-eabi-gcc --version  
arm-none-eabi-gcc.exe (GNU Tools for ARM Embedded Processors) 4.8.3 20140228 (release)  
RM/embedded-4_8-branch revision 208322]  
Copyright (C) 2013 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

- コンパイラバージョンが表示されない場合はPATHを確認

環境構築：ディレクトリ構造

- bin
 - Windows向け開発環境のバイナリ
 - mkimageを含む
- doc
 - mrubyリファレンス
 - ビルド手順
 - サンプルプログラムの説明
- hr-tecs
 - TOPPERS/HRP2及びEV3プラットフォーム
 - サンプルプログラム (hr-tecs/workspace/mruby_app)
- mruby
 - mruby ver1.2.0
 - build_config.rbでEV3用のクロスコンパイルを指定

mrubyのビルド

- mrubyのビルド（host用コンパイラ→ARM用クロスコンパイル）
 - パッケージを展開したディレクトリで
 - \$ cd mruby
 - \$ make
 - mrubyディレクトリでmakeを実行すると、mrubyがビルドされる。
- ※ビルドには、ruby及びbisonのインストールが必要
下記の出力がされればビルド成功

```
=====
Config Name: ARM
Output Directory: build/ARM
Included Gems:
  mruby-print - standard print/puts/p
  mruby-toplevel-ext - toplevel object (main) methods extension
  mruby-compiler - mruby compiler library
=====
```


開発方法

開発方法 menu+bluetooth

- アプリケーションをEV3に取り込む
 - パッケージを展開したディレクトリで
 - `$ cd hr-tecs/workspace/menu+bluetooth/`
 - `$ make tecs`
 - GNU Makeがtecsgenを実行
 - `$ make`
 - アプリケーションを選択実行するためのアプリがSDカードに取り込まれます。
 - エラーとなる場合はSDの指定ディレクトリが間違っている可能性があります。hr_tecs/workspace/menu+bluetooth/Makefileにおいて自身のSDのドライブ名に変更してください。

```
#  
# SDのドライブ文字を指定  
SD_DIR = /cygdrive/e/
```

この名前をSDカードのドライブ名に変更

開発方法 cp_mrb_app①

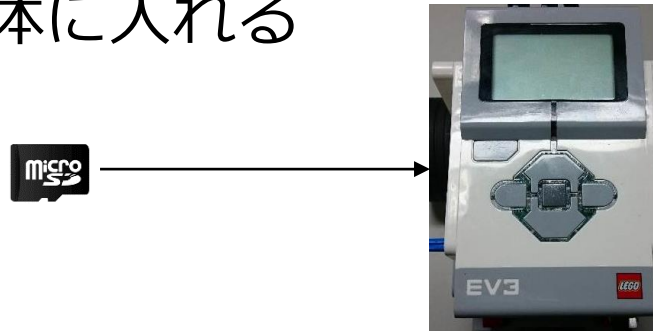
- mrubyバイトコード(.mrb)をSDに保存する
 - \$ cd hr_tecs/workspace/cp_mrb_app/
 - Makefile (hr_tecs/workspace/cp_mrb_app/Makefile)を編集
- コードを自分で作成した場合はこのディレクトリに保存し、Makefileにファイル名を追加

```
# mrubyのアプリケーションファイル名
#APP_NAME = battery_sample.rb
APP_NAME = button_sample.rb
#APP_NAME = color_sample.rb
#APP_NAME = color_sample2.rb
#APP_NAME = ev3way_sample.rb
#APP_NAME = gyro_sample.rb
#APP_NAME = lcd_sample.rb
#APP_NAME = lcd_sample2.rb
#APP_NAME = lcd_sample3.rb
#APP_NAME = led_sample.rb
#APP_NAME = motor_sample.rb
#APP_NAME = motor_sample2.rb
#APP_NAME = rtos_sample.rb
#APP_NAME = speaker_sample.rb
#APP_NAME = speaker_sample2.rb
#APP_NAME = touch_sample.rb
#APP_NAME = ultrasonic_sample.rb
```

実行したいアプリケーションを
一つ選ぶ

開発方法 cp_mrb_app②

- mrubyバイトコード(.mrb)をSDに保存する
 - \$ make
 - バイトコード (.mrb) の作成とSDへのコピーを実行
 - ここでもSDの指定ディレクトリに注意してください。
 - アプリケーションは複数コピー可能なので、他にも実行したいアプリケーションがあれば同様の手順でSDに保存しておいてください。
- SDカードをEV3本体に入れる



実行

アプリケーションの起動方法

- SDカード入れたEV3の電源を入れる

**中央 (Enter) ボタン
で電源オン**



OS起動中



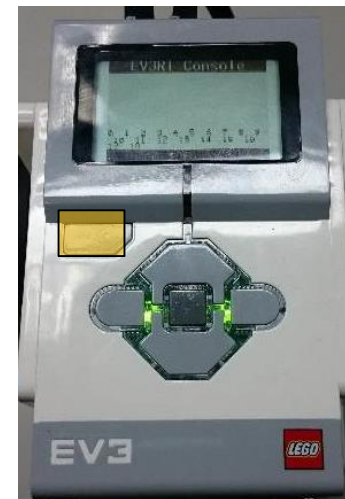
**LEDが赤
の間はOS
の起動中**

**中央 (Enter) ボタンで
mrubyプログラム開始**



LEDが緑になればOS起動完了

**アプリケーション
の実行**



**戻る (Back) ボタン
長押しでスタート画面へ**

モードの切り替え

- 左右のボタンでモードの切り替えが可能です



Runモード



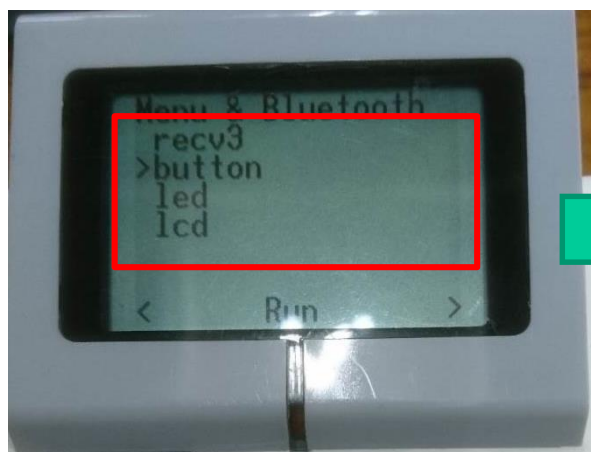
Deleteモード



Bluetoothモード

Runモード

- hr-tecs/workspace/cp_mrb_app でのmakeで保存されたバイトコードが選択実行可能



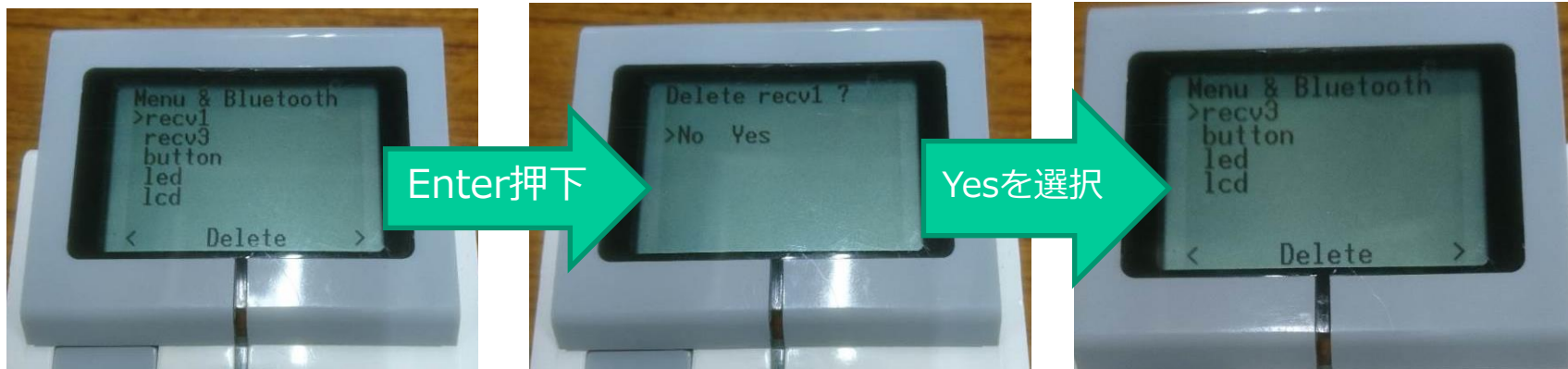
保存されたアプリケーションが
LCDに一覧表示されている



アプリを実行

Deleteモード

- SDに保存したバイトコードをEV3上で削除
 - Bluetoothで受信した不要になったファイルも消せます。



削除したいアプリケーション
を選択

確認画面

SDから削除
一覧から消える

Bluetooth受信モード

- doc/mruby_on_ev3rt+tecs_build
「開発方法：Bluetooth」
と同様にBluetooth受信が可能です
- TeraTermからバイトコードを受信し実行します
- 「recv#」という名前で保存されるので再起動時にRunモードから選択実行可能
 - #には1から順に使用されていない番号が入ります

Bluetooth受信モード

- EV3をBluetoothでPCと接続してください
 - より詳しくは `doc/mruby_on_ev3rt+tecs_build` [開発方法：Bluetooth] を見てください。



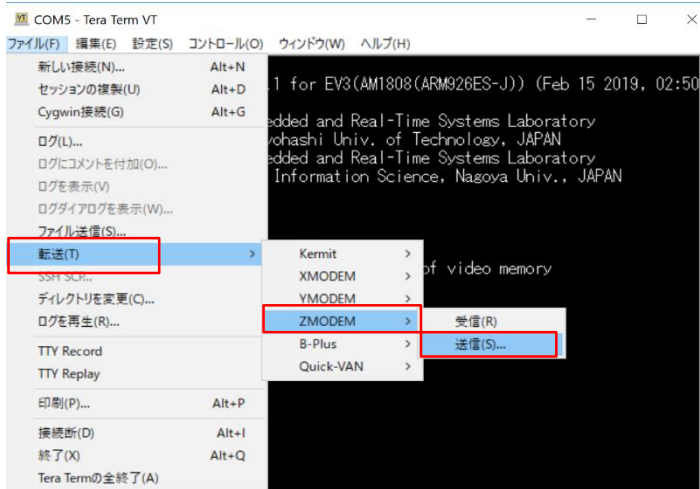
この画面でEnterを押す前に
EV3とPCをBluetoothで
接続してください



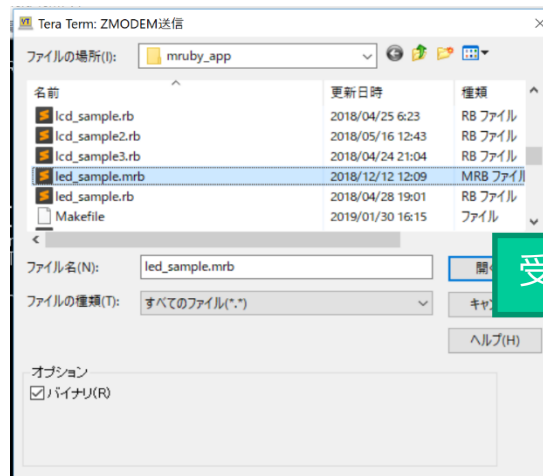
この画面になればTeraTerm
からバイトコード受信可能です

Bluetooth受信モード：TeraTerm操作

- TeraTermからバイトコードを受信します
※バイトコード受信中に電源を落とさないでください
- バイトコードは保存されているので再起動時に
recv#としてRunモードから選択実行可能



転送 → ZMODEM → 送信



送信するmrbコードを選択



受信完了

EV3で実行される