

mrubyを使った プログラミング教育

九州工業大学 情報工学部

田中和明

自己紹介

- 九州工業大学
田中 和明（たなか かずあき）
- 教育研究
 - プログラミング, アルゴリズム,
データベース、Robot Sensor Processing
 - 組込みシステム（mruby、リアルタイムOS）、
デジタルサイネージ、三次元測量（土木）
- Rubyコミュニティ活動

目次

- Ruby, mruby
- プログラミング教育

Ruby, mruby

プログラム言語Ruby

- オブジェクト指向型スクリプト言語
- 可読性が高い
- Webアプリケーション開発で広く使われている
- ISO/IEC 30170, JIS X 3017
- 日本発のプログラム言語

Rubyを使うメリット

- Rubyを使うと、、、
 - 開発しやすい
 - プログラムを再利用しやすい
 - 可読性が高い（メンテナンスしやすい）

ソフトウェアの 開発効率が高い

C言語

```
#include <stdio.h>

int main(void)
{
    int i, s;
    int ary[5] = {2,3,5,7,11};

    s = 0;
    for( i=0 ; i<5 ; i++ ){
        s = s + ary[i];
    }

    printf("%d\n", s);

    return 0;
}
```

プログラムが
読みやすい
書きやすい

Ruby

```
ary = [2,3,5,7,11]

s = 0
ary.each do |x|
    s = s + x
end

puts s
```

Rubyプログラムの実行

- インタープリタ方式
 - 実行時にコードを解析する
 - 「プログラムを書いて、すぐ実行できる」
- 動的な実行
 - 動的なメモリ管理 (GC)
 - 「開発者はメモリを管理しなくて良い」

組み込みソフトウェア開発

- 組み込みソフトウェア開発に Rubyの良さを取り入れたい
- 何が課題だろうか？

Rubyプログラムの実行

- インタープリタ方式
 - 実行時にコードを解析する
 - 「実行時に多くの資源を必要とする」
- 動的な実行
 - 動的なメモリ管理 (GC)
 - 「実行時間の予測ができない」
 - 「リアルタイム性が保証できない」

mruby = 軽量Ruby

- Rubyを軽量化した実装
 - 実行時の消費メモリが少ない
 - ある程度のリアルタイム性を持つ
- 文法はRubyと同じ
- 他のシステムに組み込んで使える
 - cプログラムからmrubyプログラムを呼び出す
 - mrubyプログラムからcの関数を呼び出す

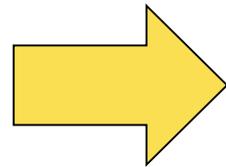
mrubyの動作

- mrubyVMによりプログラムを実行する

mrubyプログラム

```
def fib(n)
  return 1 if n<2
  return fib(n-1)+fib(n-2)
end

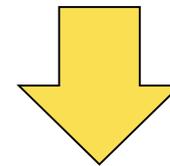
puts fib(10)
```



コンパイル

バイトコード

```
00000000 52 49 54 45 30 30 30 34 f3 be 00 00 00 f5 4d 41 |RITE0004.....MA|
00000010 54 5a 30 30 30 30 49 52 45 50 00 00 00 c0 30 30 |TZ0000IREP....00|
00000020 30 30 00 00 00 4b 00 01 00 05 00 01 00 00 00 09 |00...K.....|
00000030 00 80 00 48 01 00 00 c0 00 80 00 46 00 80 00 06 |...H.....F...|
00000040 01 00 00 06 01 c0 04 83 01 00 00 a0 00 80 40 a0 |.....@...|
00000050 00 00 00 4a 00 00 00 00 00 00 00 02 00 03 66 69 |...J.....fi|
00000060 62 00 00 04 70 75 74 73 00 00 00 00 70 00 03 00 |b...puts...p...|
00000070 08 00 00 00 00 00 11 00 02 00 00 26 01 80 40 01 |.....&...@...|
00000080 02 40 00 83 01 80 00 b3 01 c0 01 19 01 c0 00 03 |. @.....|
00000090 01 80 00 29 01 80 00 06 02 00 40 01 02 00 80 af |... ).....@.....|
000000a0 01 80 40 a0 02 00 00 06 02 80 40 01 02 80 81 2f |. @.....@...../|
000000b0 02 00 40 a0 01 80 c0 ac 01 80 00 29 00 00 00 00 |. @..... ).....|
000000c0 00 00 00 04 00 01 3c 00 00 03 66 69 62 00 00 01 |.....<...fib...|
000000d0 2d 00 00 01 2b 00 4c 56 41 52 00 00 00 17 00 00 |-...+LVAR.....|
000000e0 00 01 00 01 6e 00 00 00 01 ff ff 00 00 45 4e 44 |...n.....END|
000000f0 00 00 00 00 08 |.....|
```

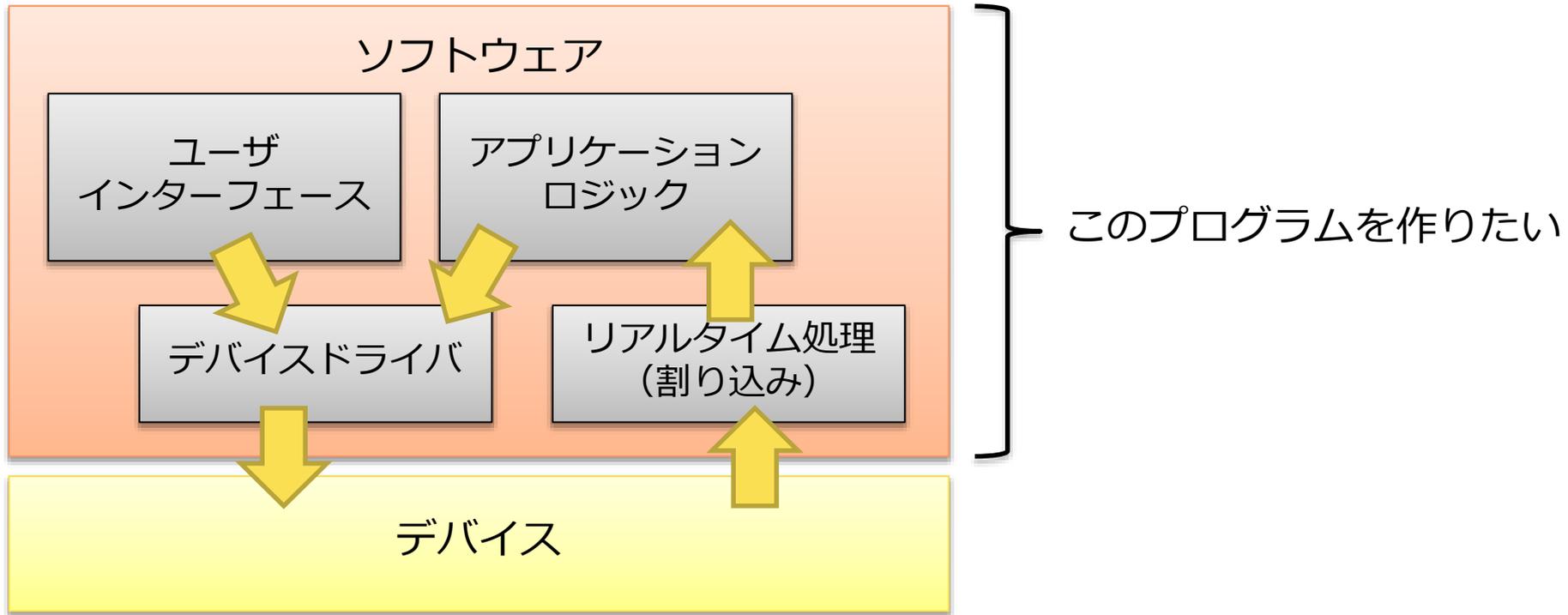


実行

mruby VM

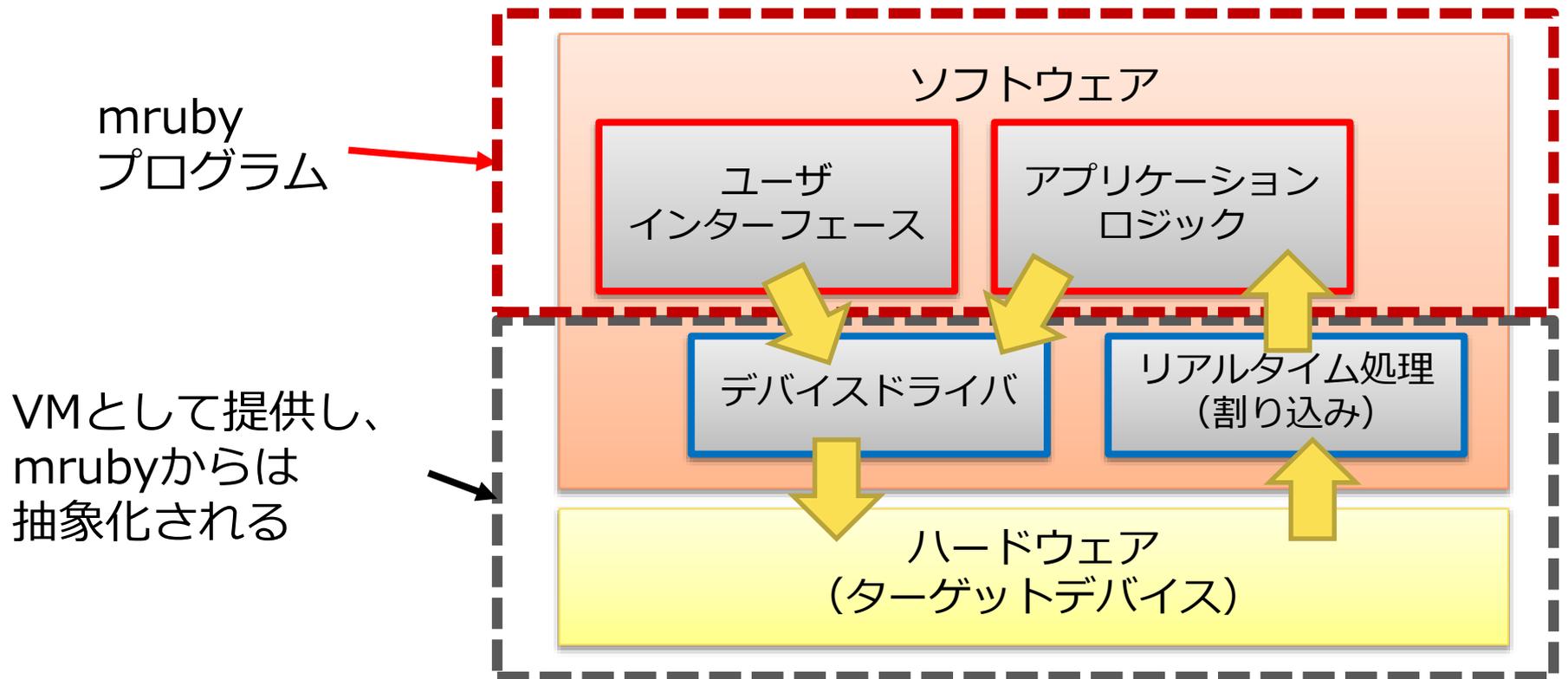
組み込みシステム開発

- 従来の組み込みソフトのイメージ



mrubyを使った組み込み開発

- mrubyの部分が開発しやすくなる



mrubyの利用事例



ネットワークルータ(IIJ社)



Apacheモジュール(OSS)



mrubyマイコンボード
(Manycolors社)



ETロボコン(アフレル社, JASA)



組み込みシステム向け
ライブラリ(ILC社)

プログラミング教育

目的は？

- Not プログラム言語の教育
- 「時代を超えて普遍的に求められる力としての『プログラミング的思考』などを育むことであり、コーディングを覚えることが目的ではない」
(小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議)

教材の要件

- 簡単に導入できる（手軽さ、環境）
- 考え方を記述できる

| | RaspberryPi | IchigoJam | Scratch(PC) |
|-----|-------------|-----------|-------------|
| 手軽さ | × | △ | △ |
| 環境 | △ | ○ | ○ |
| 記述 | なんでも可 | BASIC | GUI |

教材の要件

- 簡単に導入できる（手軽さ、環境）
- 考え方を記述できる
- キーボードが使えるようになる
- 物が動く（組込み）

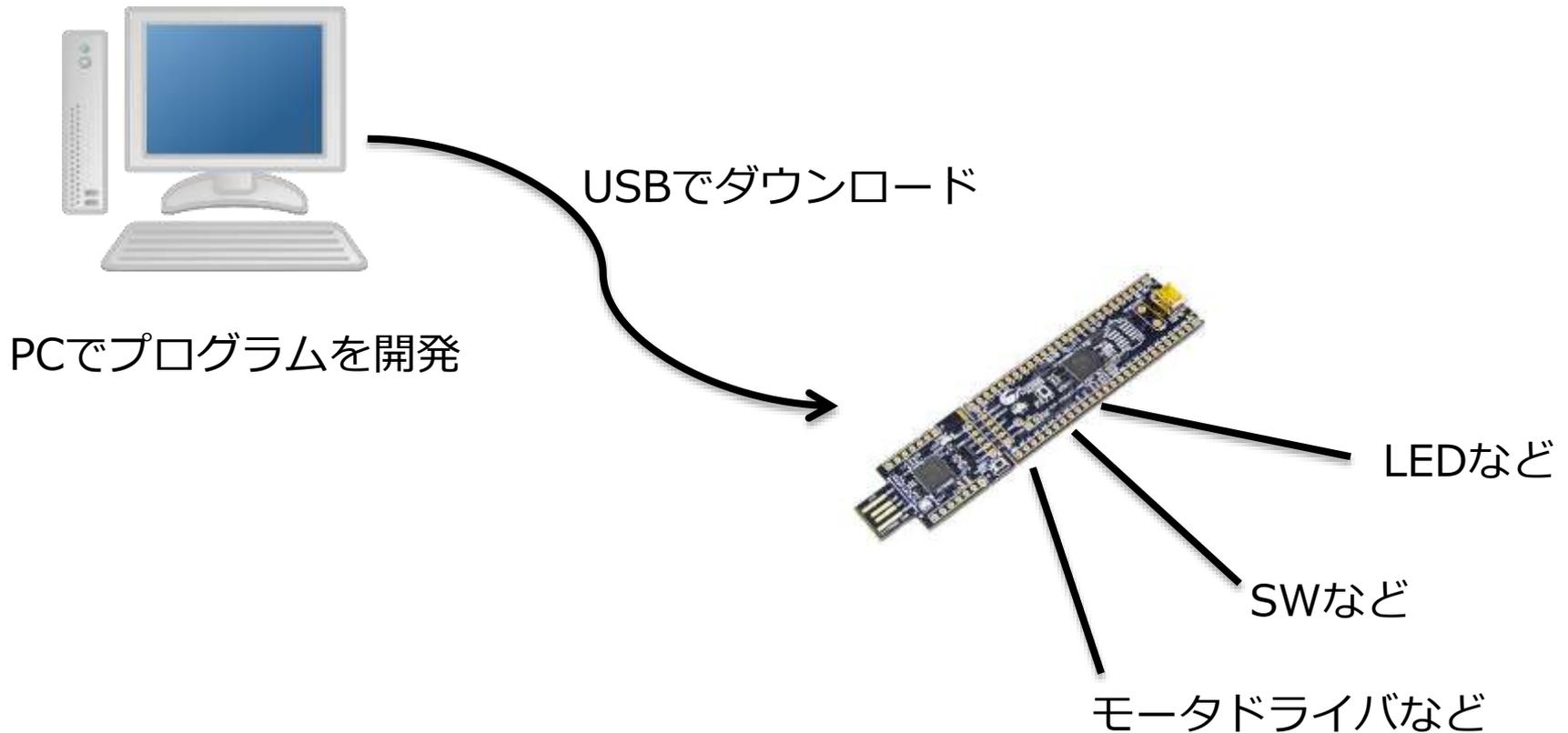
| | RaspberryPi | IchigoJam | Scratch(PC) |
|-------|-------------|-----------|-------------|
| 手軽さ | × | △ | ○ |
| 環境 | △ | ○ | ○ |
| 記述 | △ | BASIC | GUI |
| キーボード | ○ | ○ | × |
| 物が動く | △ | ○ | × |

mrubyを使った教材

- mrubyを使ったプログラム開発環境

| | mruby |
|-------|-------------------------------|
| 手軽さ | 簡単に動かせる仕組み (プログラムをすぐに動かせる) |
| 環境 | PCで開発 |
| 記述 | Ruby |
| キーボード | ○ |
| 物が動く | ○ ハードウェアアクセスを提供 |

開発の流れ



もう一つの要件

- 開発環境は共通化したい
 - テキスト、カリキュラムの継続性
- デバイスは変化するかもしれない
 - 入手可能なデバイス
 - コスト
 - ローカルなアレンジ

ハードウェアの抽象化

- VMでハードウェアを抽象化する
- 提供する機能
 - LED～デジタル出力
 - スイッチ～デジタル入力
 - モータ～PWM
 - センサ～I2C ?

DEMO

- mrubyを使った教育教材の試作
- output <ピン番号>, <値>
- input <ピン番号>
- pwm <値>