

小型IoTデバイス向けの開発言語 「mruby/c」

九州工業大学 田中 和明

しまねソフト研究開発センター 東 裕人

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
 - 組み込みシステム（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

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」

組み込みシステム

- 組み込みシステム
 - 機械や装置内に組み込まれるシステム
 - 例：
自動車，産業機器（ロボットなど）
家電製品，通信装置、ウェアラブル機器
- 開発に伴う困難
 - 多くの制限がある
(CPU・メモリ・時間・電力)
 - ハードとソフト両方の技術を使う

全開発費の内訳(過年度比較)



IPA SEC 組込みソフトウェア産業の動向把握等に関する調査(2017年)

組み込み開発

- 組み込みシステム開発費の50%はソフトウェア
- 組み込みシステムを効率よく開発したい
～Rubyの長所を取り入れたい
- さて、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の開発

- 2010年度～2012年度
 - 経済産業省 地域イノベーション創出研究開発事業でmrubyを開発（ネットワーク応用通信研究所、福岡CSK、九州工業大学）
- 2015年度～
 - 島根県 しまねソフト研究開発センターとの共同研究でmruby/cを開発
- 2016年度
 - NEDO 次世代スマートデバイス開発プロジェクトでmruby用IDEを開発（九州工業大学、SCSK九州、有明高専）
- 2017年度～2019年度
 - 経済産業省 戦略的基盤技術高度化支援事業（サポイン事業）でmrubyを使った省電力無線モジュールを開発（Braveridge、九州工業大学、SCSK九州、ILC）

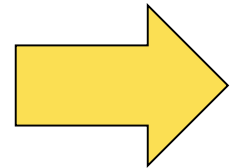
mrubyの動作

- 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

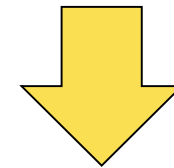
puts fib(10)
```



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

IoT, mruby/c

IoT

- 組み込みシステムの応用
- デバイスをネットに接続して活用する
- 必要な技術
 - ハードウェア（センサ）
 - ソフトウェア（データ処理）
 - ネットワーク

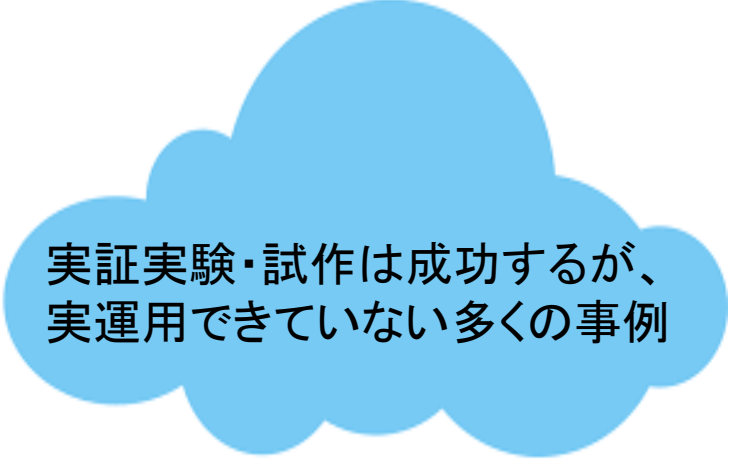
IoTで必要な処理

- センサの値を取得
- データ処理
- センサデータの送信
- データ蓄積、解析
- データの利用

} デバイスで実行する
mrubyの領域

IoTの開発

- 組み込みシステムでも、より厳しい制約
- IoTデバイス
 - 小型
 - コスト
 - 電力
 - セキュリティ



実証実験・試作は成功するが、
実運用できていない多くの事例

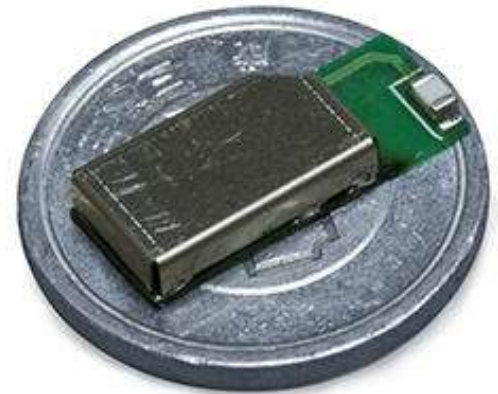
IoTデバイス

- 小型デバイスが不可欠
 - ソフトが開発できる
 - ワンチップマイコン
- 試作～実運用まで考えると、
 - ソフト開発しやすい
 - 開発したソフトがデバイスに依存しない

IoTデバイスの例



アナログを扱えるワンチップマイコン
(Cypress PSoC5LP)



BLEモジュール
(Nordic nRF52)

mrubyを搭載してみる

- nRF52モジュール
- ARM Cortex M4F
- 64KB SRAM
- 512KB Flash
- BLE 4.2
- NFC



mrubyは厳しい
↓
mrubyの小型化

mruby/c

- mrubyバイトコードと互換性を持つVM
- 相違点
 - 少ないメモリで動作する（64KB以下）
 - 複数のmrubyプログラムを動かせる（シングルコア）
 - オブジェクト実装の簡素化
 - 例外処理の簡素化
 - 処理速度よりも省メモリ

mruby/cの方針

- GCは消極的
 - 基本的にGCはしない
 - プログラム終了時に全メモリを開放する
 - リファレンスカウントを併用する
- 最小のクラスライブラリ
 - 多くのクラスはオプション
- Cインターフェースを簡単に記述できる

mruby/cの特徴

- OS無しで複数プログラムを実行する
 - 複数のセンサを利用する
 - 優先度が違うプログラムを実行する
 - （排他制御機能も持つ）
- メモリ管理
 - 静的に割り当てたメモリを利用する
 - ほとんどの場合に $O(1)$ でメモリ割り当て

消費メモリの比較

実行時に必要なメモリ量

Ruby

数MB

mruby

400KB

mruby/c

< 64KB



mruby/cが想定するデバイス

- 小型デバイス、ワンチップマイコン
 - 64KB以下のRAM
 - OS無し
- 省電力デバイス
 - すぐに起動する
- イベント・ドリブンな処理
 - プログラムは短時間の動作
(起動～実行～終了が短時間)

その他

- コンカレント処理
 - CPUやOSの支援なしにコンカレント処理
 - 割り込みが利用できるなら利用する
(パフォーマンス向上)
- メモリは静的に確保する
 - 高速な動的メモリ割り当てアルゴリズム

allocもfreeも $O(1)$
reallocは $O(n)$. . . 残念 >_<

mruby/cのクラス

- Object, True, False, Nil
- Fixnum
- Float
- String
- Symbol
- Array
- Hash
- Range

赤字はオプション

```
#define MRBC_USE_FLOAT 1  
#define MRBC_USE_STRING 1
```

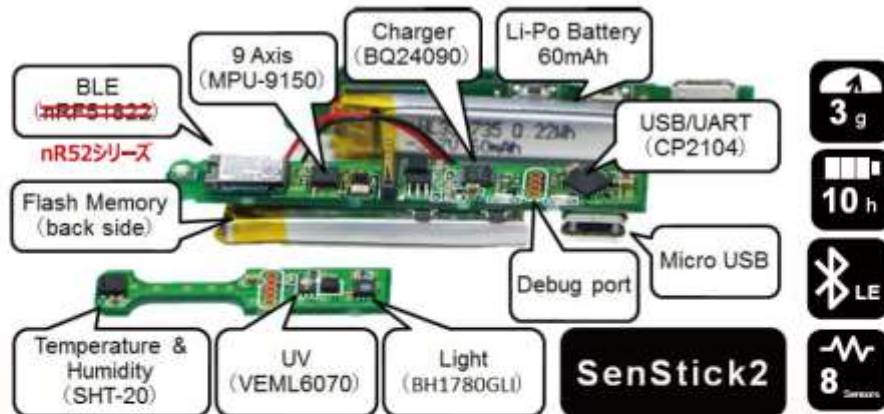
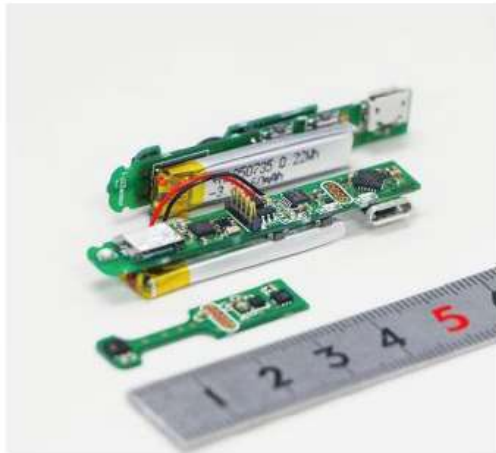
mruby/cの独自拡張

- コンカレント実行するプログラムは独立したメモリ空間を持つ
- ただし、以下は共有する
 - クラス
 - メソッド
 - グローバル変数（\$で始まる変数）

mruby/cの事例紹介

SenStick

- 小型センサデバイス（重さ 3.5g）



Ruby協からIoT/ウェアラブル開発者向け
教育キット ★ 発売決定！

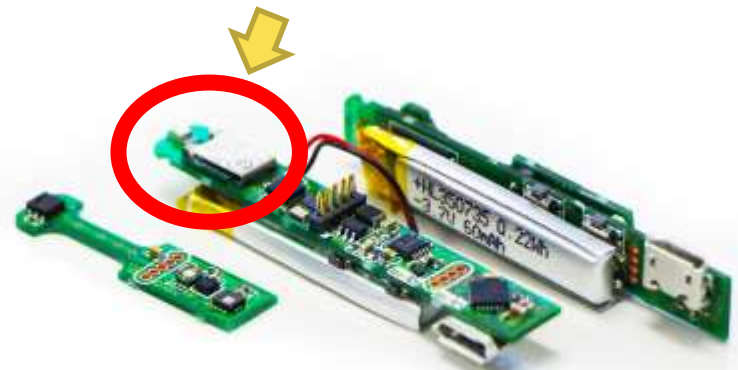
SenStick3 mruby/c

奈良先端科学技術大学院大学、Rubyビジネス推進協議会と共同開発

SenStick

- 通信モジュール
+
センサ
- 搭載するセンサ
 - 9軸
 - 温度
 - 湿度
 - 気圧
 - 照度（可視光、UV）

nRF52モジュール
ここに **mruby/c** が入る



mruby, mruby/cの入手

GitHubで公開中

<https://github.com/mruby/mruby>

<https://github.com/mrubyc/mrubyc>

mrubyの情報： 軽量Rubyフォーラム

<http://forum.mruby.org>

mruby/cの情報： しまねソフト研究開発センター

<http://www.s-itoc.jp>