
組込みハードウェアへのmruby アプリケーション適用試行

2016年11月3日

Japan OSS Promotion Forum

株式会社日立ソリューションズ 技術統括本部 IT技術推進センタ

三好 秀徳

富士通株式会社 共通ソフトウェア開発技術本部ソフトウェア開発技術統括部
OSS技術センター

原 嘉彦

目次

1. 自己紹介とモチベーション
2. mrubyアプリケーションとシステムの紹介
3. ハードウェア・ソフトウェア詳細
4. 発生した問題とその対処
5. まとめと今後の課題

目次

1. **自己紹介とモチベーション**
2. **mrubyアプリケーションとシステムの紹介**
3. **ハードウェア・ソフトウェア詳細**
4. **発生した問題とその対処**
5. **まとめと今後の課題**

■ (株)日立ソリューションズ所属

- さまざまな業務システムをRubyを使って構築

■ 日本OSS推進フォーラムアプリケーション部会所属

- 詳細はのちほど

■ Rubyist Magazineの編集者

- <http://magazine.rubyist.net/>



るびま

Rubyist Magazine

るびま

検索

0054号 (2016-08)

巻頭言

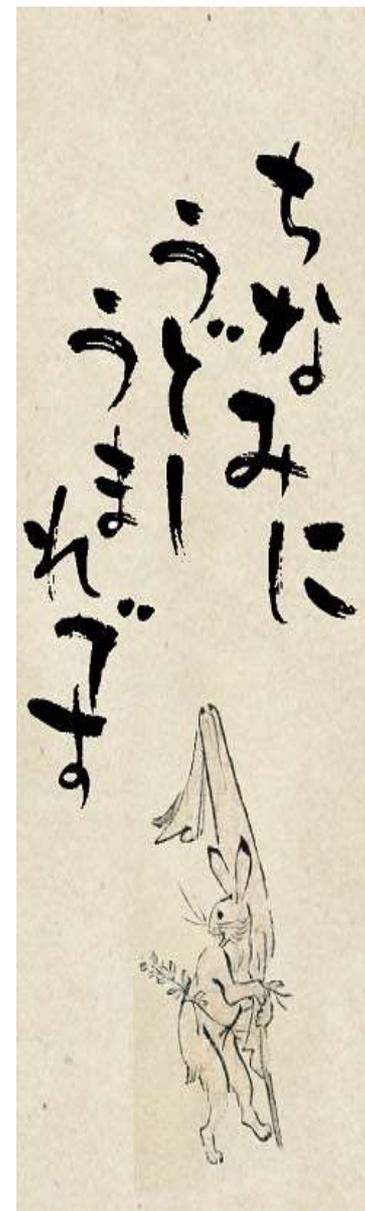
Rubyに型があると便利か
書籍紹介「プログラミングElixir」

Rubyist Magazine について

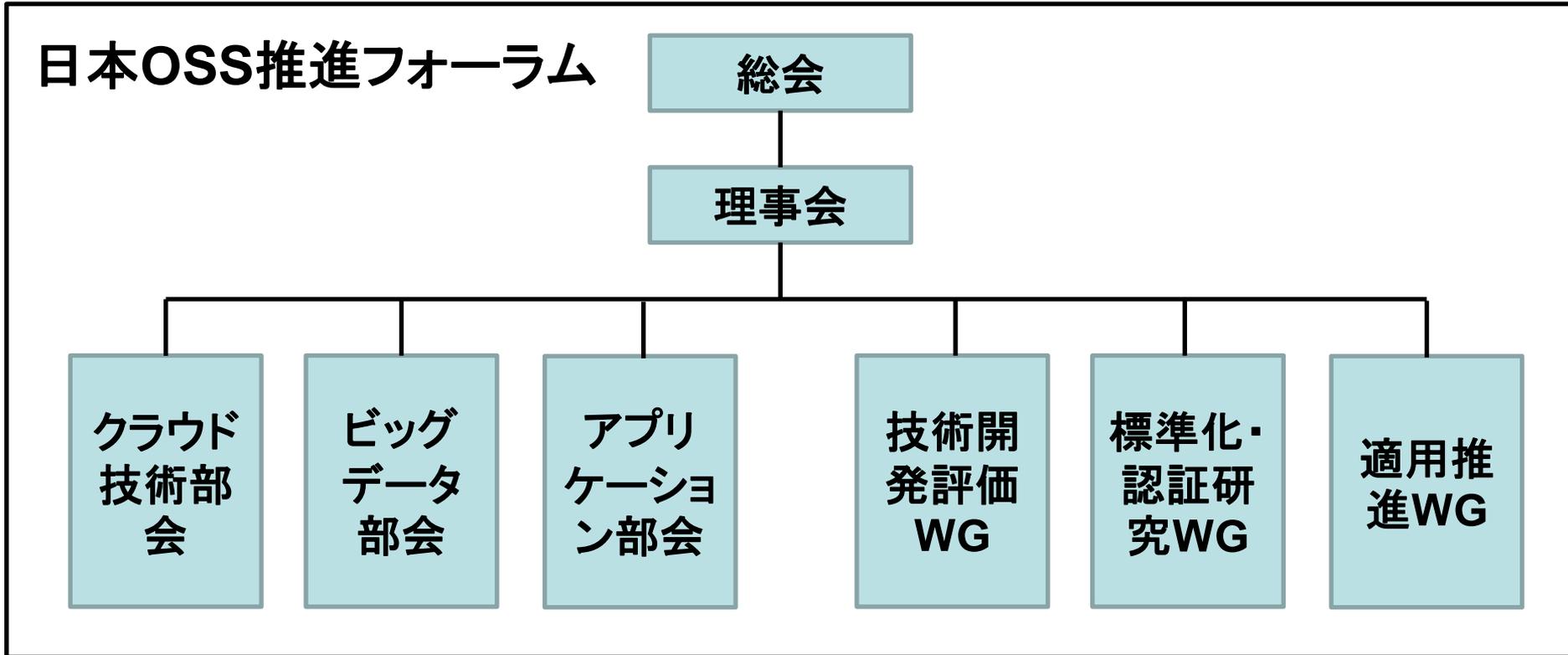
『Rubyist Magazine』、略して『るびま』は、日本 Ruby の会の有志による Rubyist の Rubyist による、Rubyist とそうでない人のためのウェブ雑誌です。

1-2.自己紹介

- 原 嘉彦 (Twitter 等ネットワーク上では @GORO_Neko と名乗っています)
- 日本OSS推進フォーラム (<http://ossforum.jp/>) アプリケーション部会に所属
- 日本 Cloud Foundry グループ (<http://cloudfoundry.gr.jp/>) にも所属 (運営委員 やってます)
- その他OSSやCloud系のコミュニティに不定期に出席
- 日頃は IT ベンダ企業 (富士通) でエンジニアとして働く
- 近頃のマイブームは「マイコンボード + Ruby」で何か作ること



■ ITビジネスの場へ、OSSの適用・普及・促進をはかる団体



- 産学官が力を合わせ、ITビジネスの場へOSSの利活用を促進
- 組織内に複数の部会やWGを設置し、OSS自体に関する技術研究やOSSのビジネスへの適用方法を実践的に探求している

1-4. 日本OSS推進フォーラム

- 北東アジア(日中韓)全体における、ITビジネスへのOSSの適用・普及・促進活動の一角も担っている



■ アプリ開発にOSSの適応普及促進をはかる部会

- Rubyを軸に、OSSをアプリケーション開発へ適用する促進活動を多角的に実施
- 活動成果は、各種カンファレンス、部会主催の勉強会等を通して、世の中に還流



- メールマガジン発行
 - 月1回発行
- アプリケーション部会主催IT勉強会
 - 年2回ペースで開催
 - フォーラム会員以外の方も聴講可
- 年次成果報告会
 - 年1回実施
 - フォーラム会員以外の方も聴講可
- 詳細は公式サイトを参照
 - <http://ossforum.jp/>



■ Internet of Things

- 『IoTへのRuby/mruby適用試行』

- ◆ RubyWorld Conference 2015 1日目 廣田 哲也(株式会社シーイーシー)氏による発表

- 意外に少ない組み込みハードウェアへのmruby適用

■ mruby関連技術の習得

- アプリケーション開発のノウハウ習得(mrbgems/リソース等)

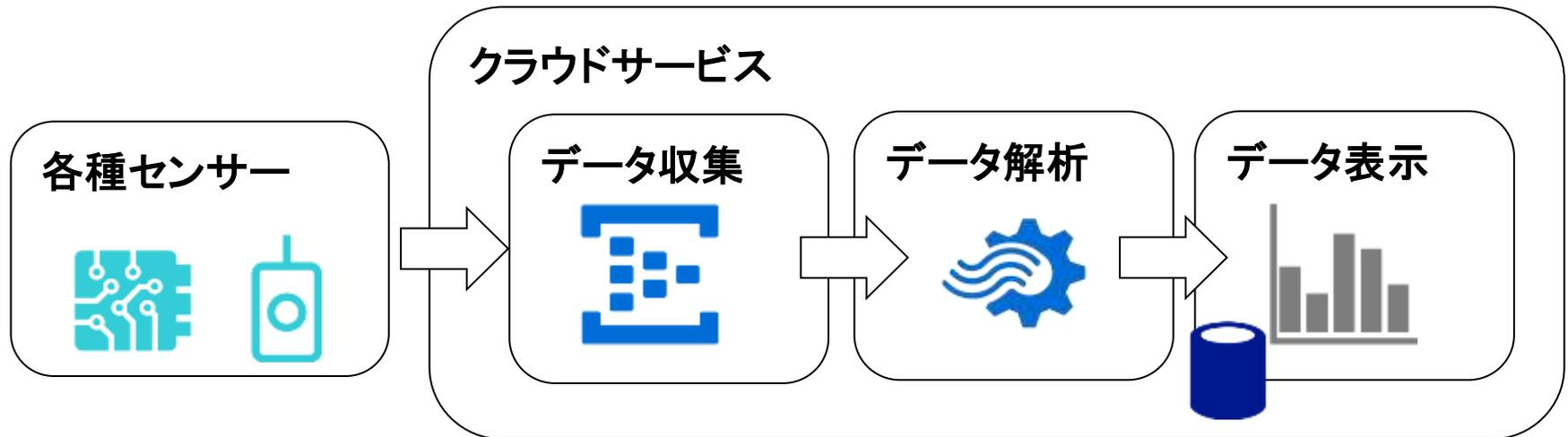
- ◆ Japan OSS Promotion Forum 2015 『mrubyとCRubyとの性能比較』富田昌宏(株式会社富士通システムズ・イースト) <http://ossforum.jp/jossfiles/JOSSPF-2015-7-3.pdf>

- 容易に準備できる環境を使うことでmrubyアプリケーション構築だけに集中する

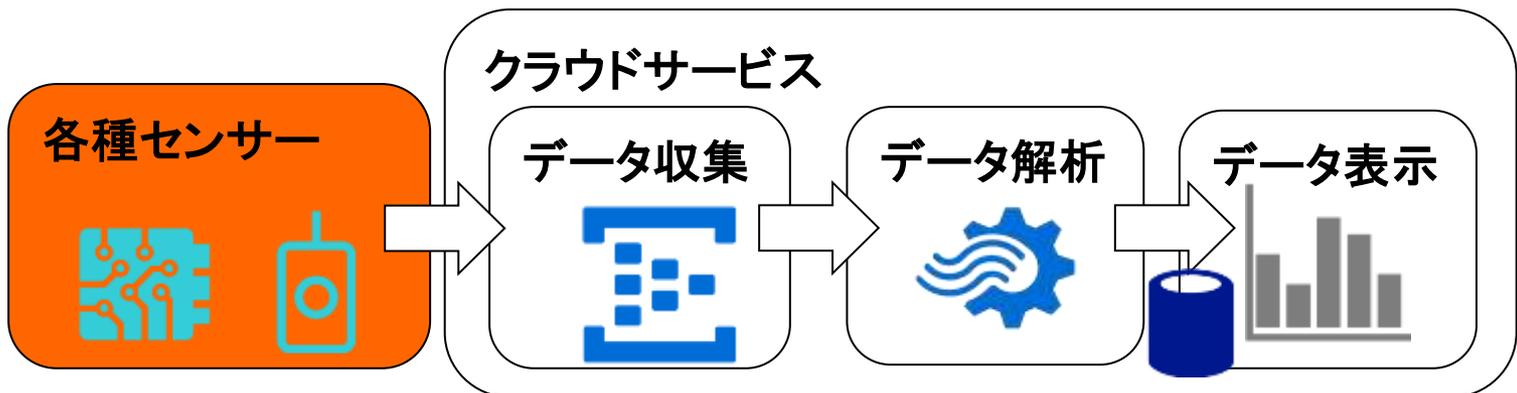
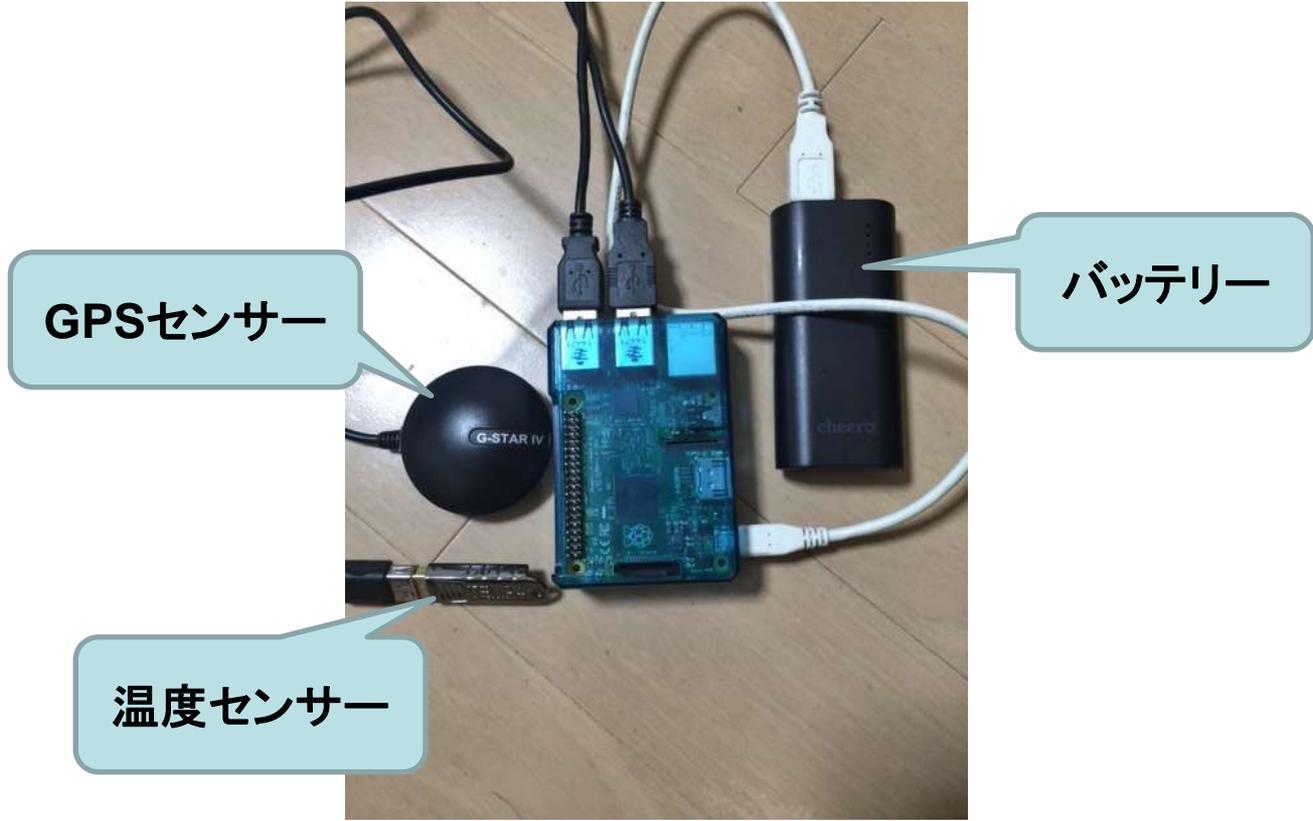
目次

1. 自己紹介とモチベーション
2. **mrubyアプリケーションとシステムの紹介**
3. ハードウェア・ソフトウェア詳細
4. 発生した問題とその対処
5. まとめと今後の課題

- ランニング用リアルタイム位置情報発信アプリの開発
 - 位置情報(GPS)と気温を随時クラウドサービスに送信
 - 最低限4時間は動かすことを目標



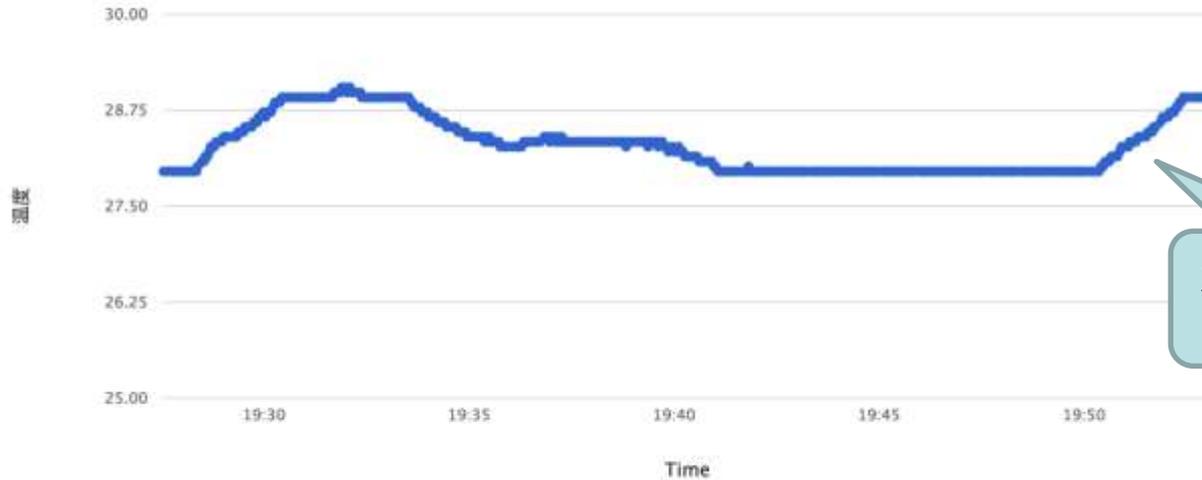
2-2.各種センサーの紹介



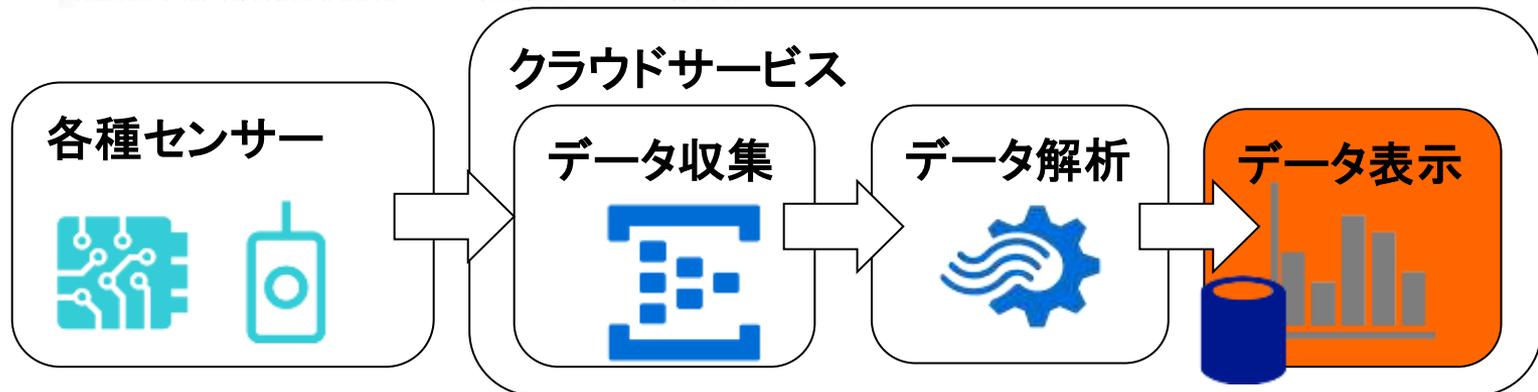
2-3. データの表示 (温度)

Runlog RaspberryPi2 Download GPX file

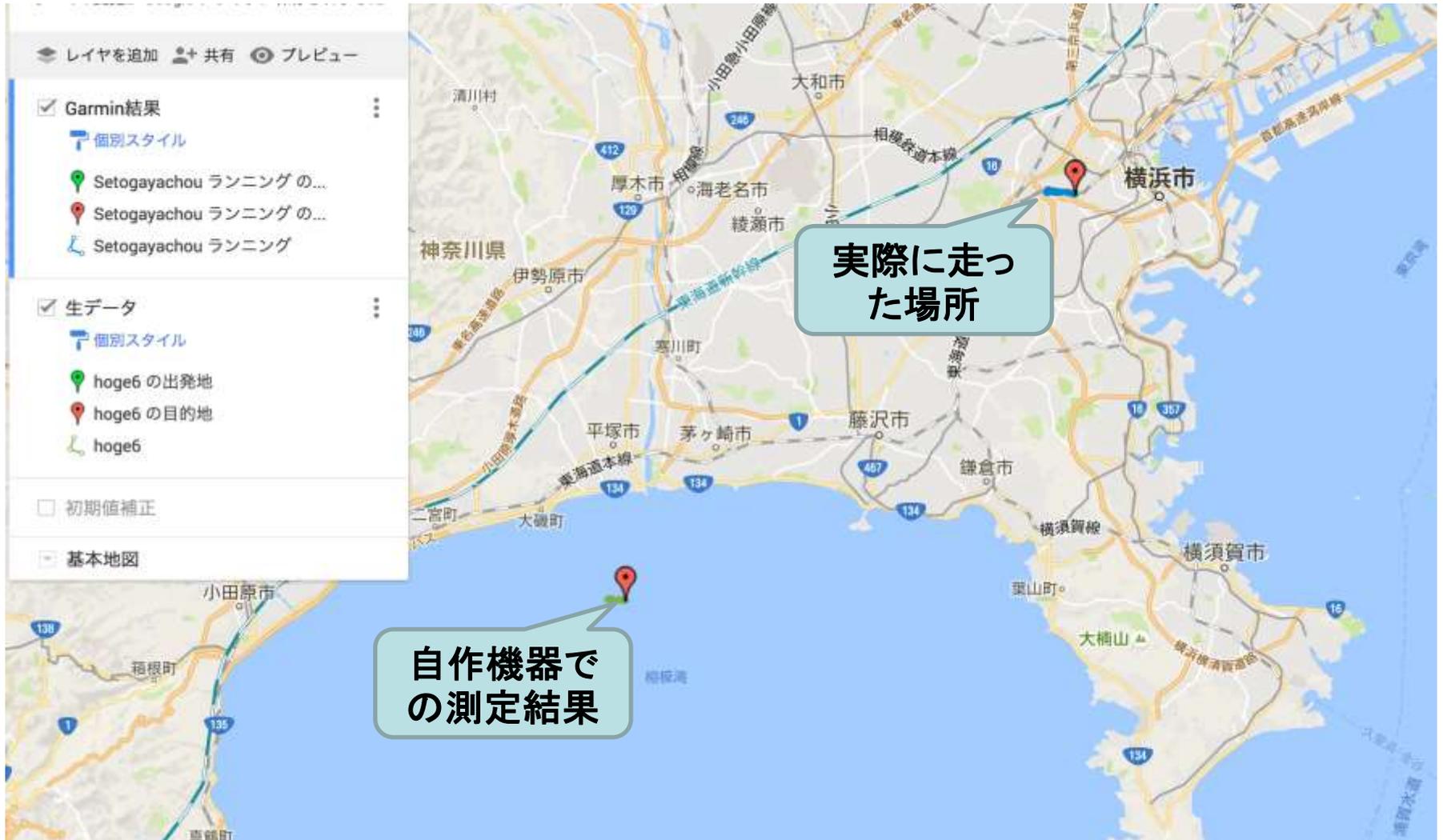
GPX download



温度の時系列
変化



2-4.GPSデータを地図上にマッピング



2-5.GPSデータ(初期値を補正)



目次

1. 自己紹介とモチベーション
2. mrubyアプリケーションとシステムの紹介
3. **ハードウェア・ソフトウェア詳細**
4. 発生した問題とその対処
5. まとめと今後の課題

■ mrubyアプリの開発に集中できるようにする

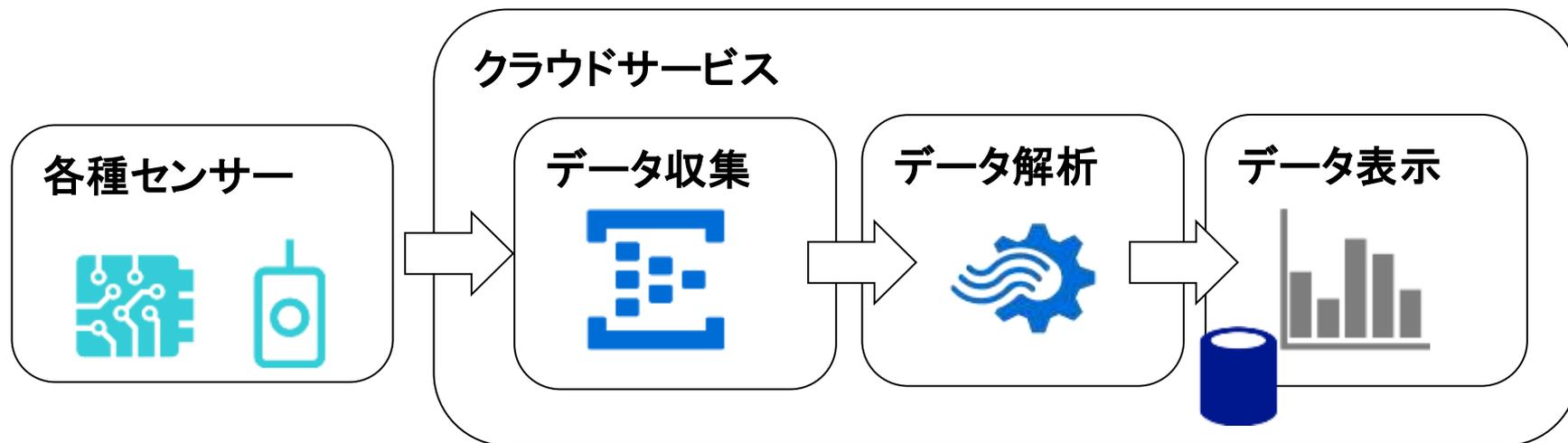
- 取り回しが容易なハードウェア

 - ◆ハンダ付け作業を行わずともセンサー類の取り付けが可能

 - ◆mrubyの動作実績あり

- 既存の基盤技術を活用する

 - ◆クライアントから送信されるデータの処理基盤はクラウドサービスを活用



3-2. 準備したハードウェア

■ Raspberry Pi 2

- サイズ : 85mm × 56mm × 17mm
- センサー類の接続がUSBやGPIOで実施できる
- 使い慣れたLinux OS (Raspbian) が使える



■ 温度センサー

- PCsensor.com TEMPper gold
- USBシリアル通信でデータの取得が可能



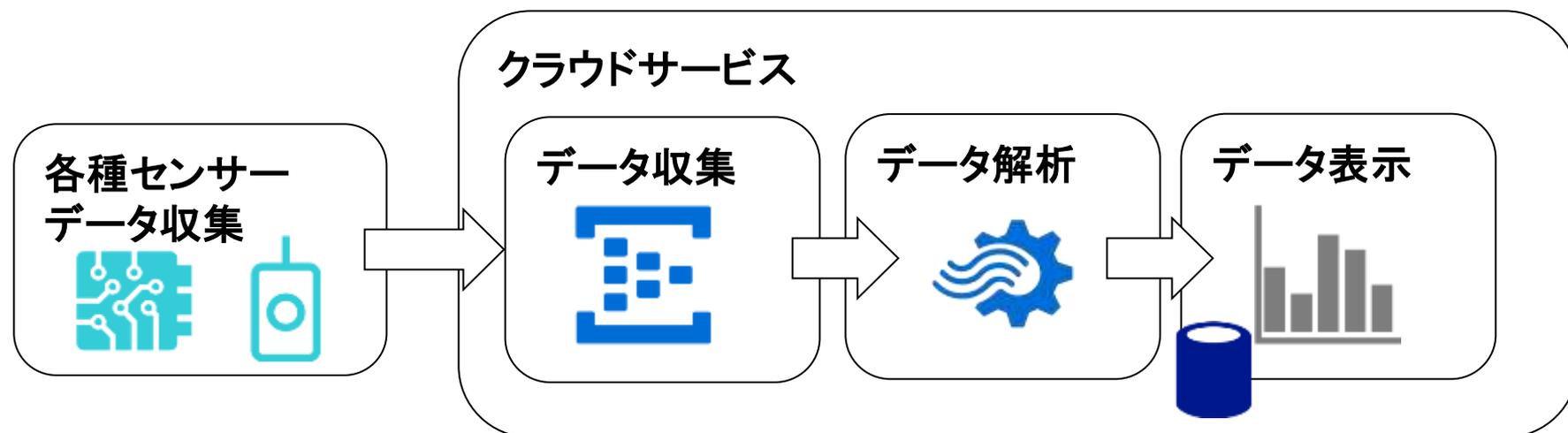
■ GPSレシーバー

- GLOBALSTAT BU-353 S4 GPSレシーバー
- USBシリアル通信でデータの取得が可能



3-3.ソフトウェア詳細

大分類	小分類	使用したもの
各種センサーデータ収集	OS	Raspbian
	言語	mruby 1.2.0
クラウドサービス	データ収集	Microsoft Azure IoT Hub
	データ分析	Microsoft Azure Stream Analytics
	データ表示	Microsoft Azure App Service
	データ表示 アプリ	JRuby 9.1.5.0 + Ruby on Rails 4.2.



- 最初にCRubyで実装。その後、mrubyに移植
 - システムの構築に必要なものを洗い出す
 - ◆ HTTPS形式でのデータ送信
 - ◆ HMAC-SHA256でハッシュ化
 - ◆ BASE64エンコード化
 - ◆ シリアルポートの読み込み
 - 慣れた言語で実装可能性を検証。正解を知る。
- 極力CRuby用のgemは使わない。

■ 移植時に直面した課題

- gemを使わないCRubyのスクリプトもmrubyでは動かない

```
[~/mruby-1.2.0]$ head -3 runinfo_sender.rb
DEVICE_NAME = ENV["DEVICE_NAME"]
SHARED_ACCESS_KEY = ENV["SHARED_ACCESS_KEY"]
HOST_NAME = ENV["HOST_NAME"]
[~/mruby-1.2.0]$ bin/mruby runinfo_sender.rb
runinfo_sender.rb:1: uninitialized constant ENV
(NameError)
[~/work/mruby-1.2.0]$
```

■ 移植時に直面した課題

- gemを使わないCRubyのスクリプトもmrubyでは動かない
 - ◆ 環境変数が取れない
 - ◆ 日時データのフォーマット化
 - ◆ etc...
- mrubyは本体を小さくし、さまざまな機能はmrbgemsで実現する方針
- mruby用のgemである“mrbgems”を探す

3-7.mrbgemsの探し方

■ 情報元は主に以下URLから

- <https://github.com/mruby/mruby/wiki/Related-Projects>
 - ◆ mruby GitHubリポジトリのWikiページ
- <http://forum.mruby.org/download/index4.html>
 - ◆ 軽量Rubyフォーラム mruby 1.2.0 リリースアナウンスページ
 - ◆ Windows/Mac/Linuxの動作検証結果が掲載

特定非営利活動法人
mruby 軽量Rubyフォーラム specified non-profit corporation mruby Forum

Home | about Forum | Join Forum | Seminar/Event | Members | Download | Docs

Download Stable v1.2.0

Satbel v 1.0 download (Jan. 2014) | Satbel v 1.1 download (Nov. 2014)

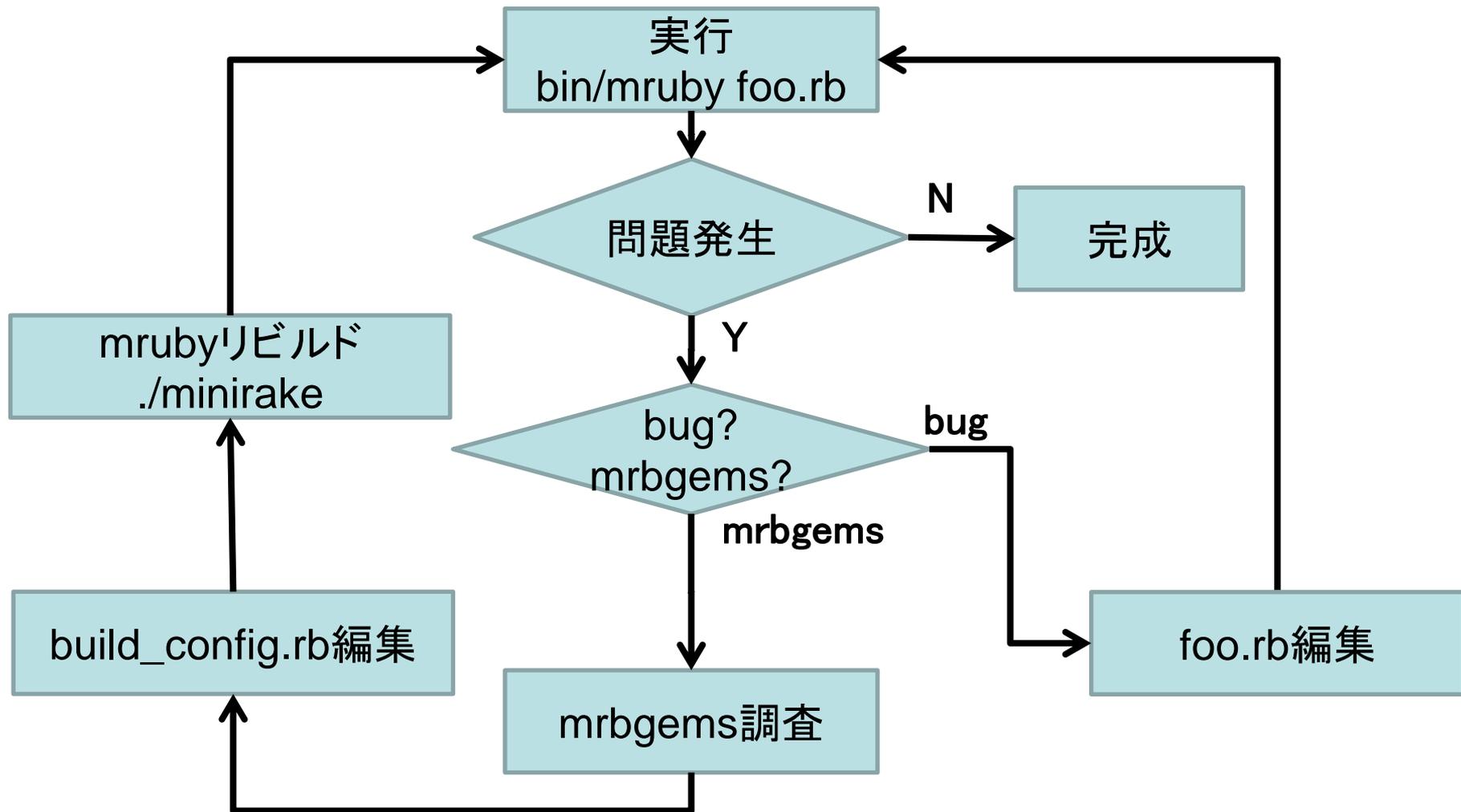
mruby v1.2.0 リリース

* mruby Stable版V1.2.0 (2015.11.16) はこちらからダウンロード出来ます。
リリースノートは[こちらから](#)

Linux/Ma/Windows安定版で評価済外部mrbgemsリスト

mgem name	Description	Mac	Ubuntu	Windows		
				Visual Basic	msys	cygwin
mruby-bcrypt	OpenBSD-style Blowfish-based password hashing.	○	○			
mruby-pjson	pure mruby JSON parser	○	○	○	○	○
mruby-md5	MD5 digest function	○	○	○	○	○
mruby-io		○	○			
mruby-kmp	IO and File class for mruby	○	○	○	○	○
mruby-updategems	update all git based mrbgems	○	○	○	○	○
mruby-ipaddr	IPAddr class for mruby	○	○			
mruby-bin-mirb-hostbased	Hostbased mirb for serial communication.	○	○			

3-8. mrubyアプリケーション実装の進め方



目次

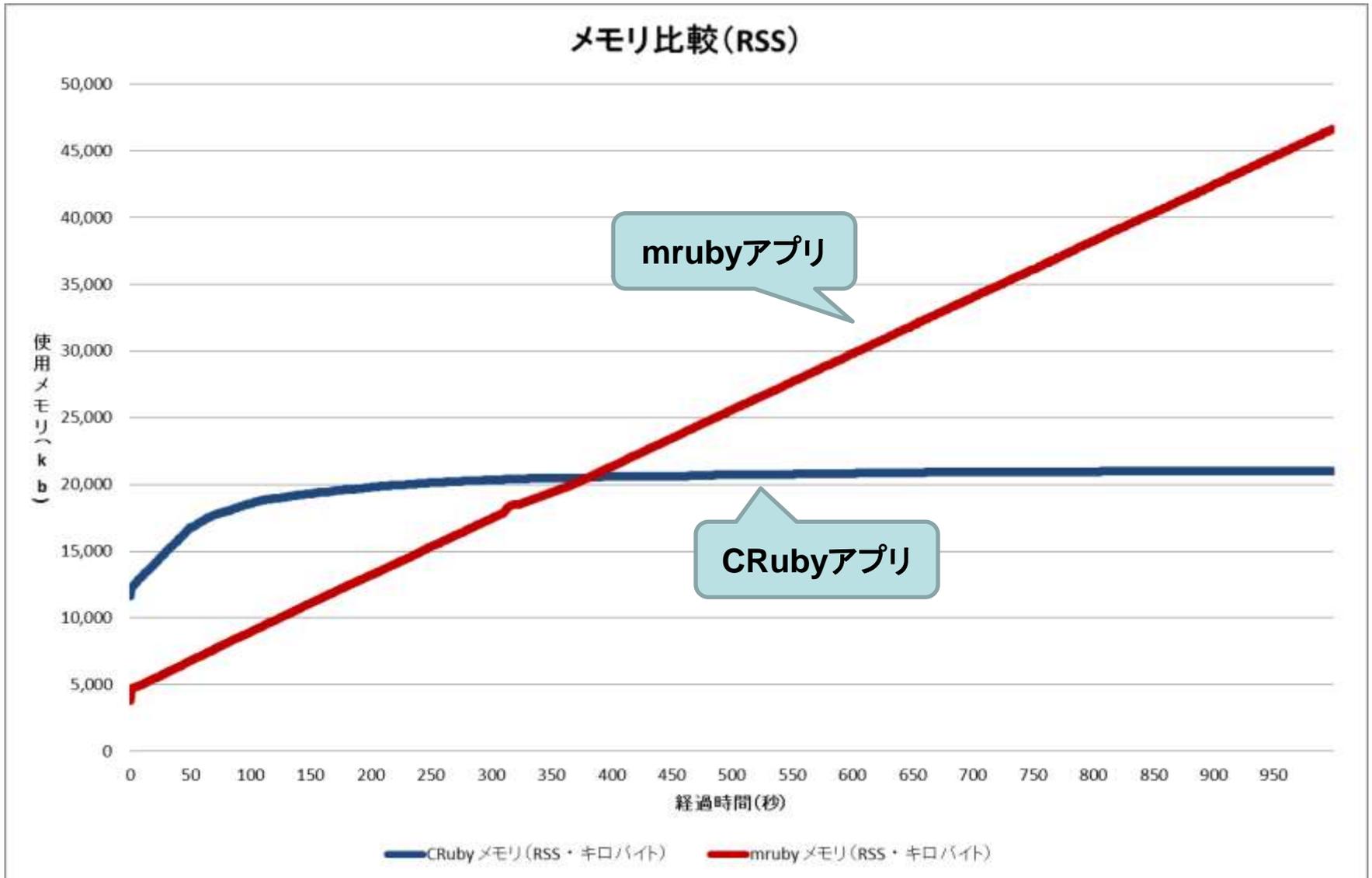
1. 自己紹介とモチベーション
2. mrubyアプリケーションとシステムの紹介
3. ハードウェア・ソフトウェア詳細
4. 発生した問題とその対処
5. まとめと今後の課題

- 1時間ほど経過すると、mrubyアプリが異常終了する
 - 目標稼働時間(4時間)持たない
- さまざまなログを取って検証を実施

```
{ "Time": "2016/9/15 23:44:24", "Mem": 236464, ... }  
{ "Time": "2016/9/15 23:44:25", "Mem": 236496, ... }  
{ "Time": "2016/9/15 23:44:26", "Mem": 236528, ... }  
trace:  
[5] /home/dev1/work/mruby-1.2.0/build/mrbgems/mruby-  
io/mrblib/io.rb:33:in IO#open  
(中略)  
[0] runinfo.rb:37  
/home/dev1/work/mruby-1.2.0/build/mrbgems/mruby-  
io/mrblib/io.rb:33: pipe_open failed. (RuntimeError)
```

使用メモリが単調増加
している？

4-2. 使用メモリ量の変化



- 時間が経過するにつれて使用メモリが増加
 - 無限ループしている箇所に着目

```
##(前略)
```

```
http = HttpRequest.new
```

```
loop do ## 無限ループで処理させる
```

```
# (中略) 送信するデータ(payload)の作成処理など
```

```
## Azure IoT Hubにデータ送信
```

```
http.post(uri_with_port, payload, {  
  "Content-Type" => "application/json",  
  "Content-Length" => payload.length.to_s,  
  "Authorization" => sas_header  
})
```

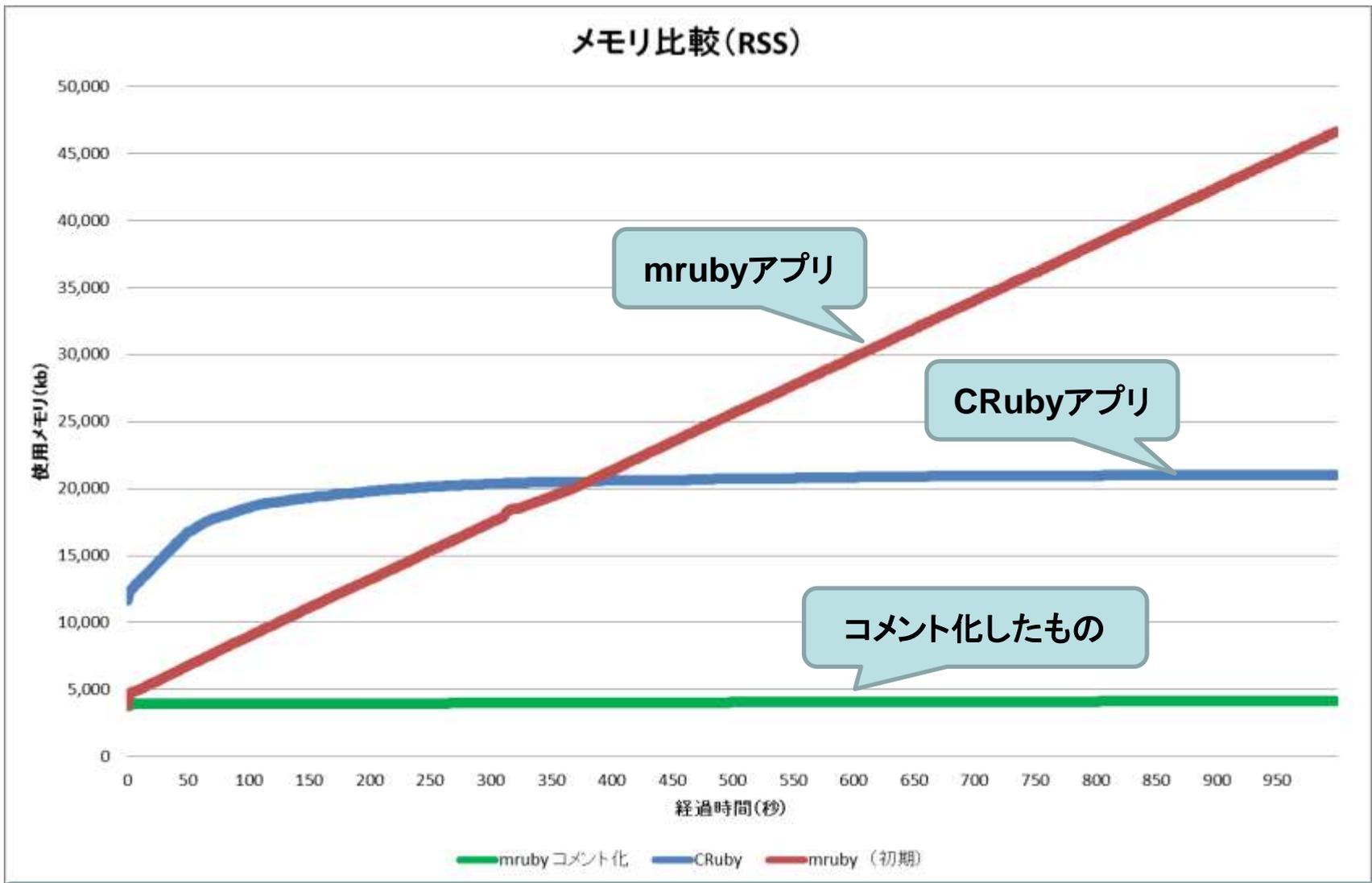
```
Sleep::sleep(1) ## 間隔をあげる
```

```
end
```

- HTTPSを使ったデータ送信処理をコメント化してみる

```
##(前略)
http = HttpRequest.new
loop do ## 無限ループで処理させる
# (中略) 送信するデータ(payload)の作成処理など
## Azure IoT Hubにデータ送信
# http.post(uri_with_port, payload, {
#   "Content-Type" => "application/json",
#   "Content-Length" => payload.length.to_s,
#   "Authorization" => sas_header
# })
Sleep::sleep(1) ## 間隔をあげる
end
```

4-5. 使用メモリ量の変化 (コメント化版)



コメント化したものは単調増加しない

HTTPS送信処理

使用しているmrbgems

mruby-httprequest

mruby-simplehttp

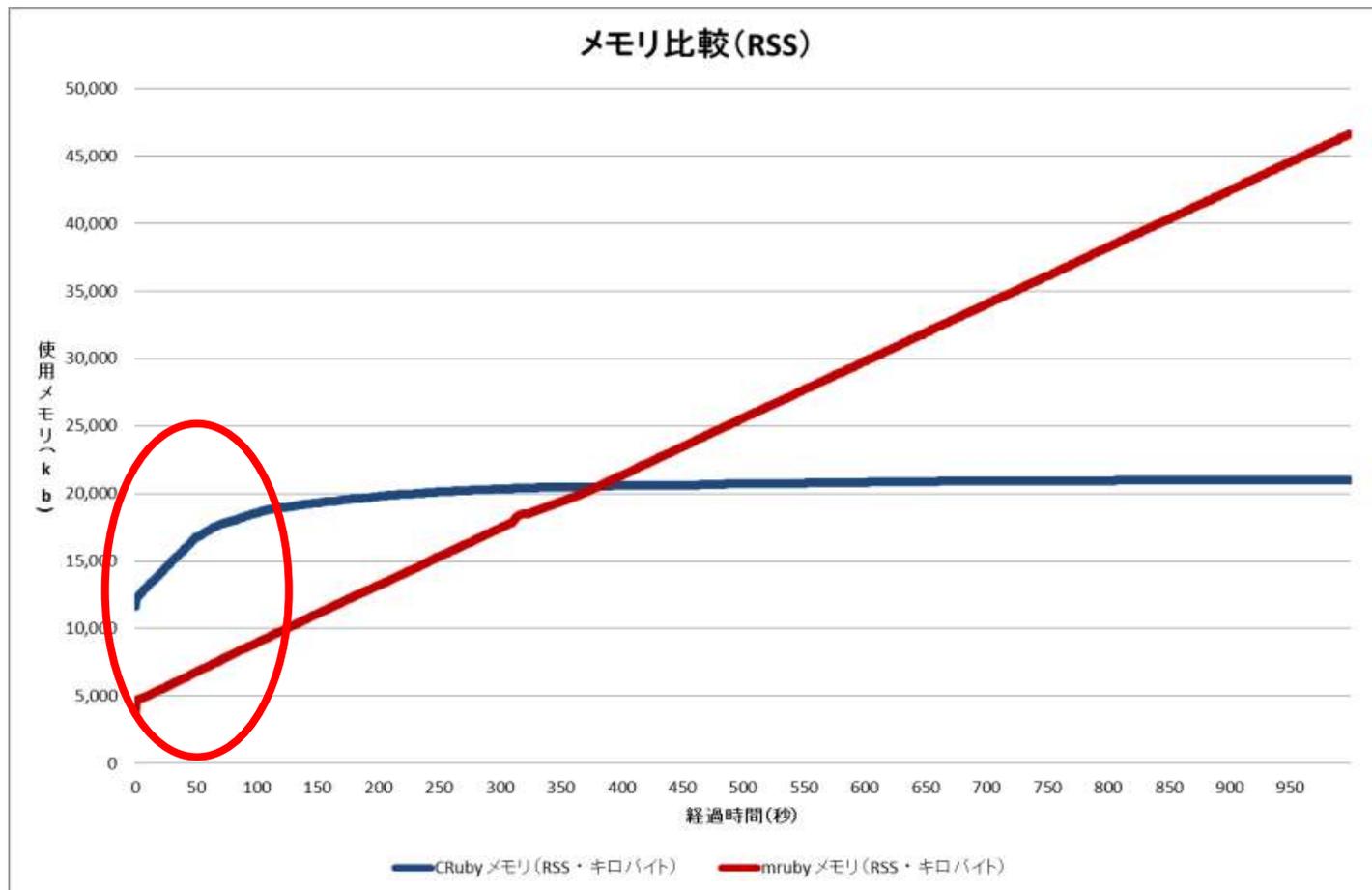
mruby-http

mruby-polarssl

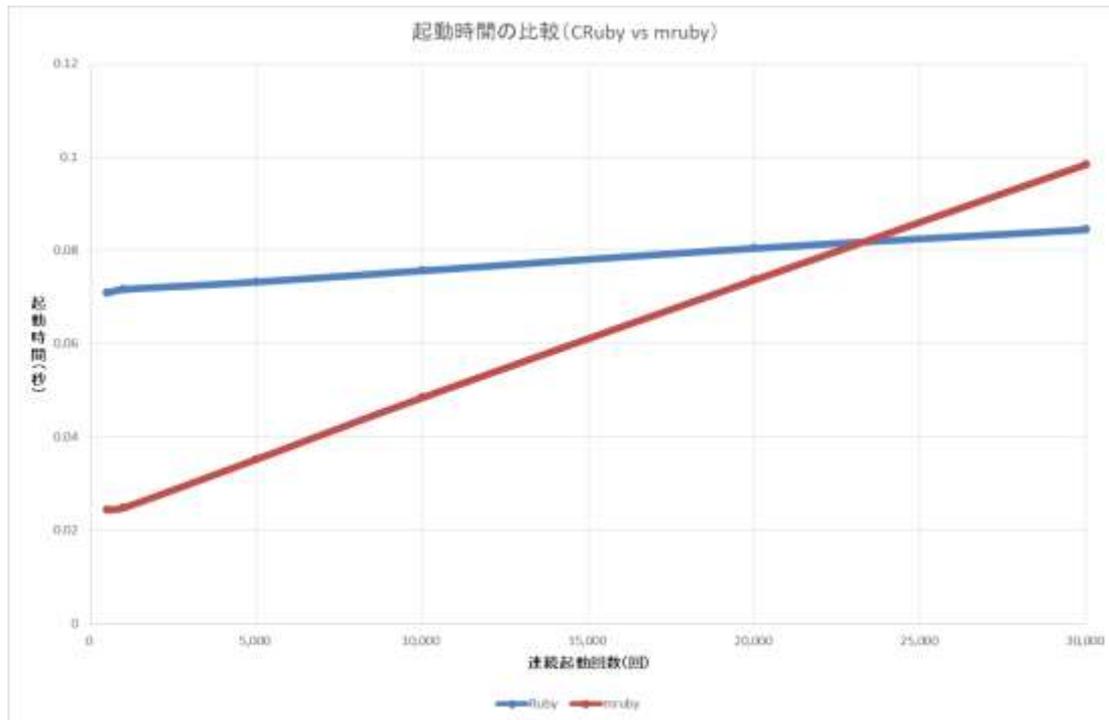
C言語

今回は原因追及と解決はできず

- mrubyアプリで無限ループすることをやめる
 - ループ回数が少ないときは、使用メモリは少ない



- mrubyアプリで無限ループすることをやめる
 - ループ回数が少ないときは、使用メモリは非常に少ない
 - mrubyアプリの起動はCRubyアプリよりも早く起動する
 - ◆ cf. mruby/c in TokyoRubyKaigi#11 (田中和明氏)



```
#!/bin/sh
```

実行するシェル

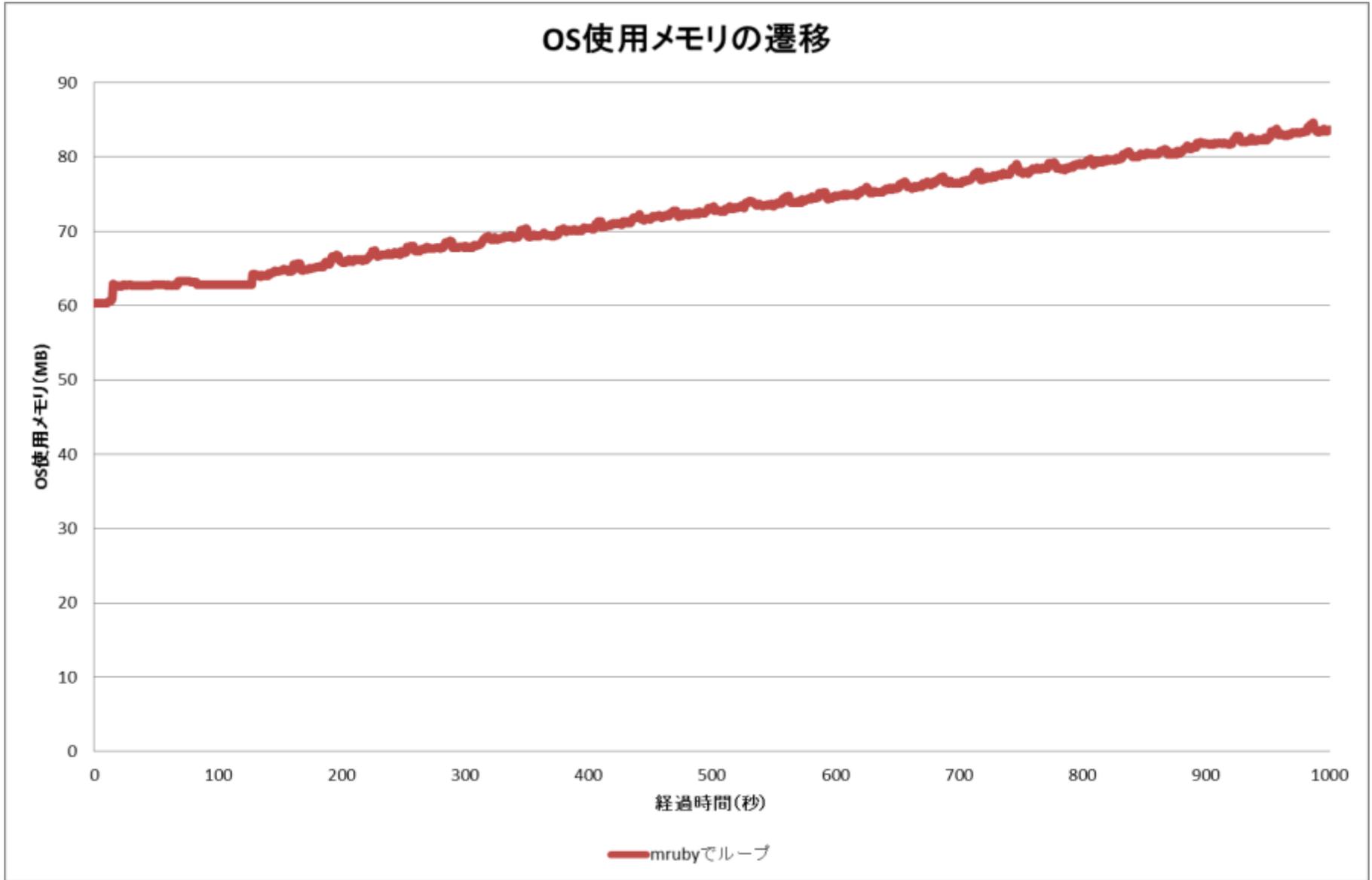
```
while :  
do  
  ./bin/mruby run_log.rb  
  sleep 1  
done
```

```
##(前略)
```

run_log.rb

```
http = HttpRequest.new  
# loop do ## シェルで実装  
# (中略) 送信するデータ(payload)の作成処理など  
## Azure IoT Hubにデータ送信  
http.post(uri_with_port, payload, {  
  "Content-Type" => "application/json",  
  "Content-Length" => payload.length.to_s,  
  "Authorization" => sas_header  
})  
# Sleep::sleep(1) ## シェルで実装  
end
```

4-9.実行結果(対策前)



4-10.実行結果(対策後)



目次

1. 自己紹介とモチベーション
2. mrubyアプリケーションとシステムの紹介
3. ハードウェア・ソフトウェア詳細
4. 発生した問題とその対処
5. **まとめと今後の課題**

- 日本OSS推進フォーラムとしてmrubyアプリを作成
 - 我々が行ったmrubyアプリの作り方を紹介
 - mrbgemsの探し方・選び方を紹介
- mrubyアプリに使用メモリが単調増加する問題発生
 - mrbgemsを含めた根本原因の解決には至らず
 - mrubyの特徴を生かして、問題を(とりあえず)解決

■ GPSデータの精度向上

- 補正データの収集(各ランドマークを基準に補正データを収集)
- 他機器のデータを使った位置の補正

■ 使用メモリが単調増加する問題の根本解決

- 『mrubyのmrb_gc_arena_save()/mrb_gc_arena_restore()の使い方』(<http://www.rubyist.net/~matz/20130731.html>)
- 『mrubyのirepテーブルを再利用する事でmod_mrubyのメモリ消費量を改善』(<http://blog.matsumoto-r.jp/?p=3147>)

■ より小さな機器で動かしてみたい

- 現状では4時間の稼働できない(消費電力大)
 - ◆ 自分が速く走る
 - ◆ ハードウェアが使う消費電力を抑える
- mruby/cの適用
 - ◆ <http://www.s-itoc.jp/activity/research/mruby/>



小型マイコン付きWiFiモジュール
ESP-WROOM-02

■ 軽量な通信プロトコルMQTTの適用

- 消費電力の低減に有効といわれている
- 現状のmruby用MQTTライブラリは認証関係の機能がない



Cloud Foundryは、Pivotal Software,Inc. の米国およびその他の国における商標または登録商標です。
Ruby on Railsは、David Heinemeier Hansson氏の米国およびその他の国における商標または登録商標です。
Windows、Microsoft Azureは米国Microsoft Corporationの米国およびその他の国における商標または登録商標です。
Macは、Apple Inc.の米国およびその他の国における商標または登録商標です。
LinuxはLinus Torvalds氏の米国、日本およびその他の国における登録商標または商標です。
Raspberry Piは、英国Raspberry Pi財団の米国およびその他の国における商標または登録商標です。
Cortexは、ARM Limitedの米国およびその他の国における商標または登録商標です。
その他、記載されている会社名、商品名は、各社の登録商標または商標です。