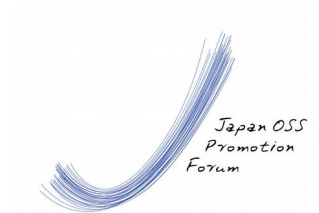


# IoT with Ruby/mruby

RubyWorld Conference 2015

2015/11/12

Computer Engineering and Consulting Ltd.  
Tetsuya Hirota



# Introduction

In the Japan OSS Promotion Forum, studied IoT with Ruby/mruby from last year, and developed the sample. This time, will present last year's result, technical challenges, and this year's activities.

- About Me
- OSS Promotion Forum
- IoT
- Overview (Last years sample)
- Technical Issues
- Countermeasures
  - Coexistence of mruby and C
  - Performance of HTTP polling communication system
- What we will do in next year
- Questions and Answers

# About Me – Ruby, Matsue and Me

I have come to Matsue, first in 2011, and this time is fifth.  
Matsue is the beautiful city of water.

And I interested in the castle, Matsue Castle is great.

And ...





# About Me - Makinohara

I'm from Makinohara of Shizuoka pref.

Makinohara has the largest tea plantations in Japan.

My home is surrounded by tea field and nature.



## 牧之原市

まきのはらし  
日本の市

牧之原市は、静岡県の市。2005年10月11日、榛原郡相良町、榛原町が合併して誕生した。キャッチフレーズは「ふれあいビタミンあいのまち・うみ・そら・みどりと共に生きる」。 [ウィキペディア](#)

面積：111.7 km<sup>2</sup>

現在の天気：温度：19°C、風向：南東、風速：3 m/s、湿度：63%

人口：4.604万 (2014年9月)

現地時間：11:55 (木曜日)

フィードバック

# About Me - My company

Introducing my company only a little.

**I am working in the CEC as an IT infrastructure engineer.**

**CEC ( <http://www.cec-ltd.co.jp/> )**

**CEC is a systems integrator in Japan.**

**Line of businesses:**

- Industry automation (e.g. Factory simulating/monitoring software)
- System Integration (e.g. Contracted software development)
- Platform Integration (e.g. Datacenter, Cloud service)



**RaLC<sup>®</sup> Facteye<sup>®</sup>**

I was in until last year a developing Web systems department.

Now, I am in a department of factory automation called **Smart Factory** unit.

# Concept of The Smart Factory

**Building Factory Information Infrastructure, and making efficient planning and production activities.**

- Building Factory Information Infrastructure. (IoT)
- The most suitable product line design support. (Simulator)
- Support of production activities. (MES) (Future)
- Traceability, Big Data analysis, etc. (Future)

## ECM (for equipment)

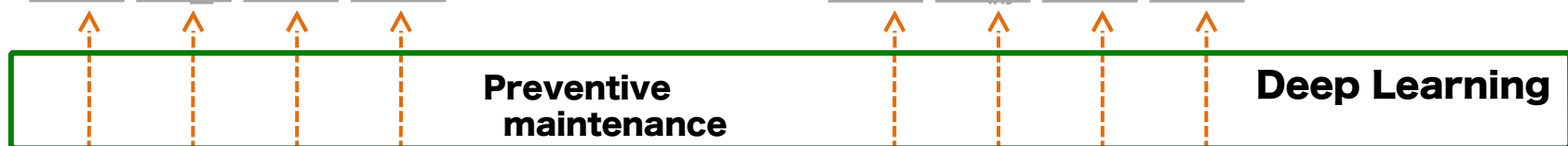


RaAP  
Virfit/RoboDiA  
SequenceEye  
Facteye

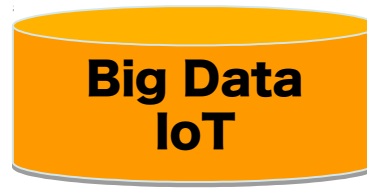
## SCM (for product)



Production schedule management  
Quality management  
MES



**Factory Information Infrastructure**  
Facteye - Equipment  
RaFLOW - Person



**Multi vender and equipments**

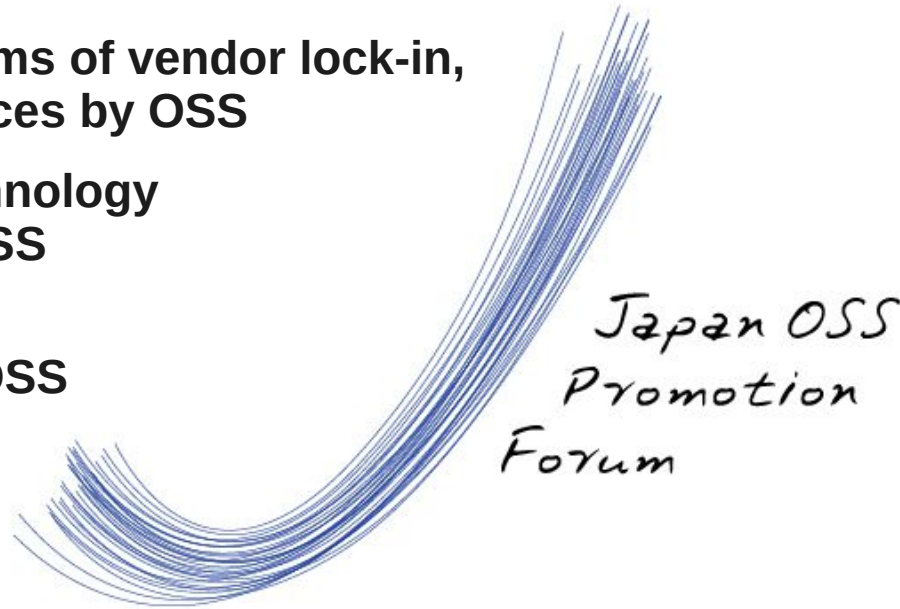


# OSS Promotion Forum

**OSS Promotion Forum is the organization which consists of the users, vendors, scholars and experts of information systems.**

**The mission of Japan OSS Promotion Forum is**

- to prevent problems of vendor lock-in, and increase choices by OSS
- promotion of technology innovation with OSS
- human resource development for OSS



I participate some studies at Application sub committee and focus on IoT with Ruby/mruby.



# OSS Promotion Forum IoT teams member



エルエスアイ開発研究所

**FUJITSU** 富士通アドバンスエンジニアリング

**HTKエンジニアリング**



*Japan OSS  
Promotion  
Forum*

**HITACHI**  
Inspire the Next  
日立ソリューションズ

**CEC**  
Computer Engineering & Consulting



## The Internet of Things (IoT)

the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity,

which enables these objects to collect and exchange data.

(Wikipedia (Oct. 15, 2015, 04:08 UTC)

[https://en.wikipedia.org/wiki/Internet\\_of\\_Things](https://en.wikipedia.org/wiki/Internet_of_Things) )

## Application

- Environmental monitoring
- Infrastructure management
- Energy management

etc.

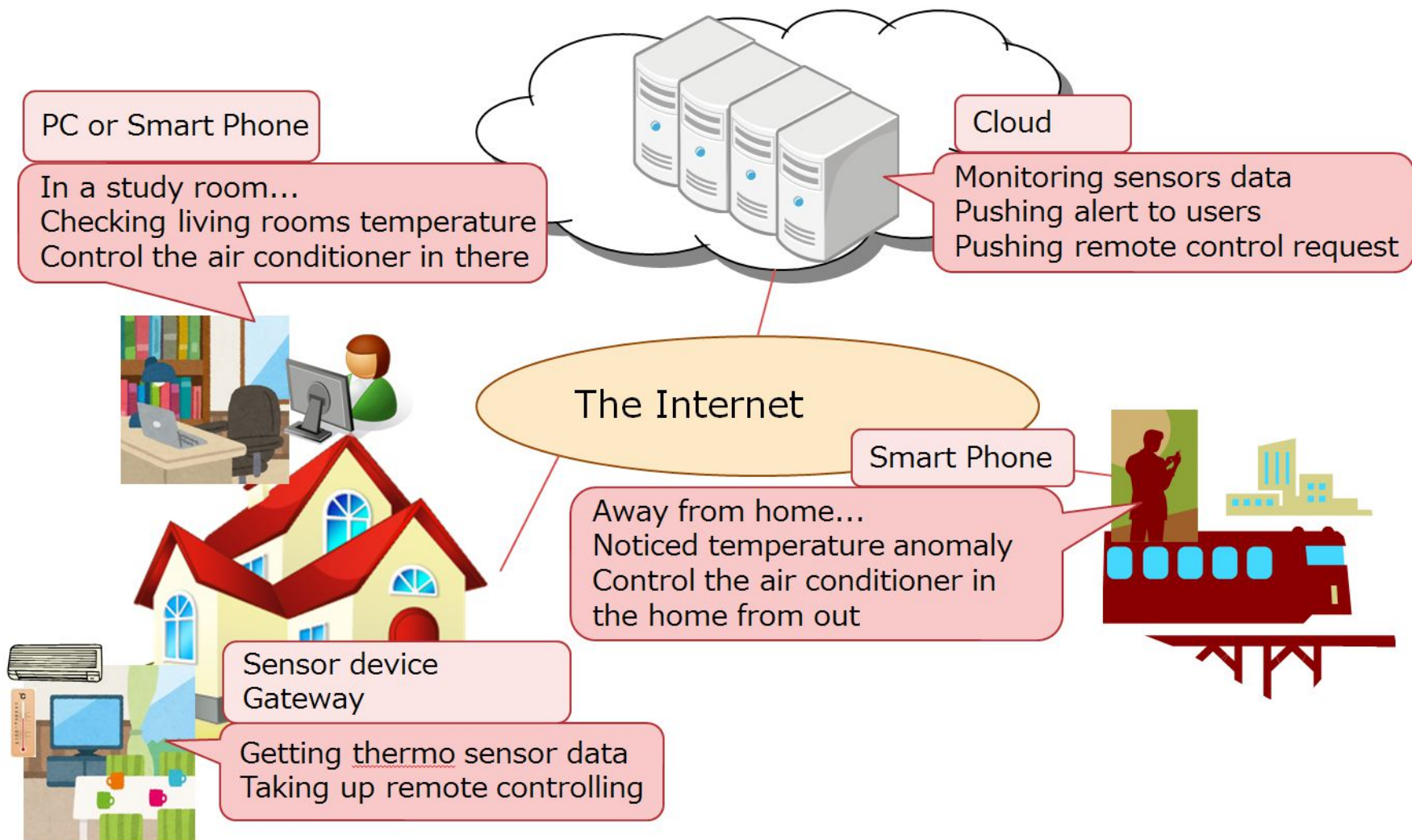
# IoT

## Hype Cycle

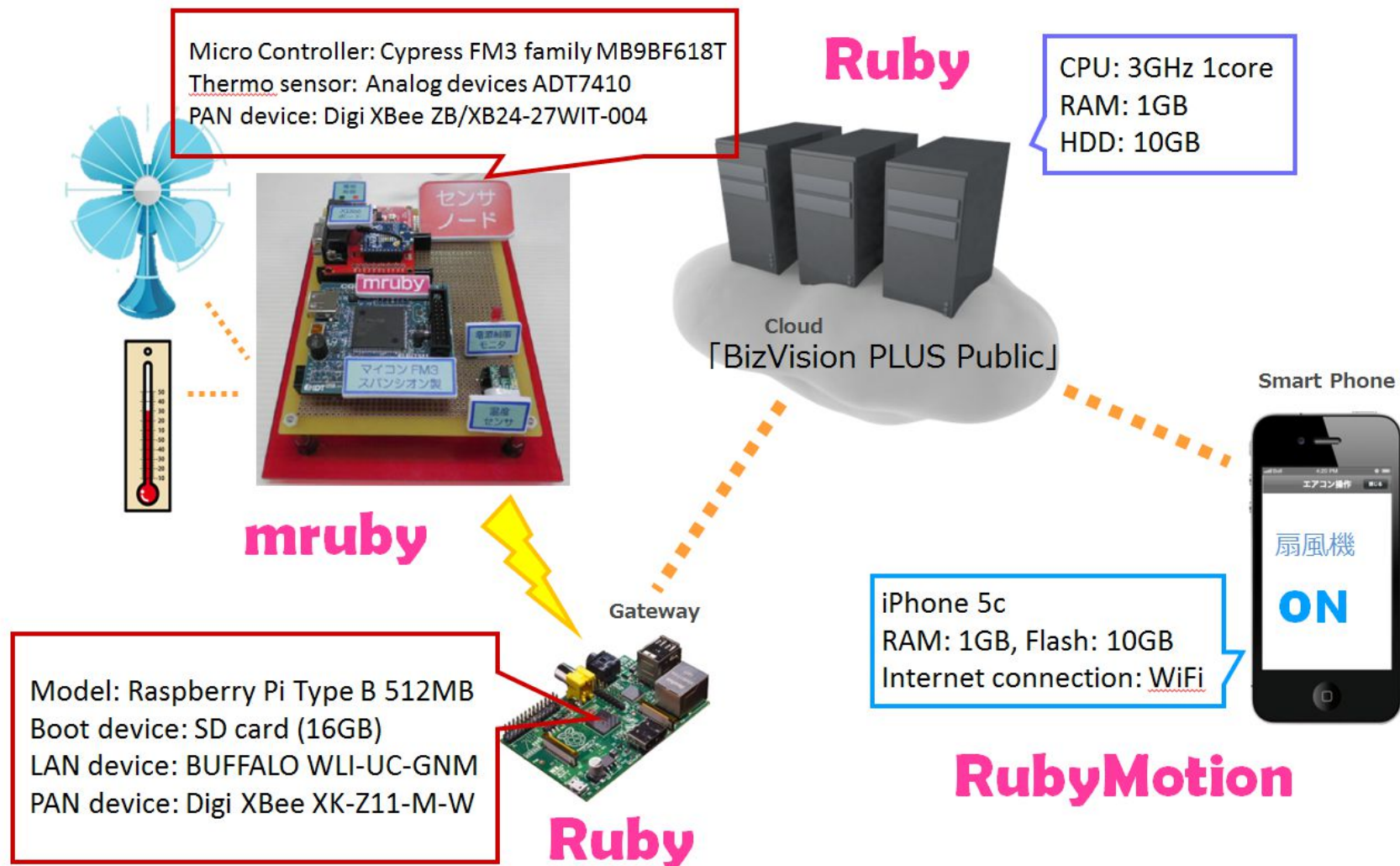


Press Release: Gartner's 2015 Hype Cycle for Emerging Technologies Identifies the Computing Innovations That Organizations Should Monitor  
<http://www.gartner.com/newsroom/id/3114217>

# Overview (Last years sample)

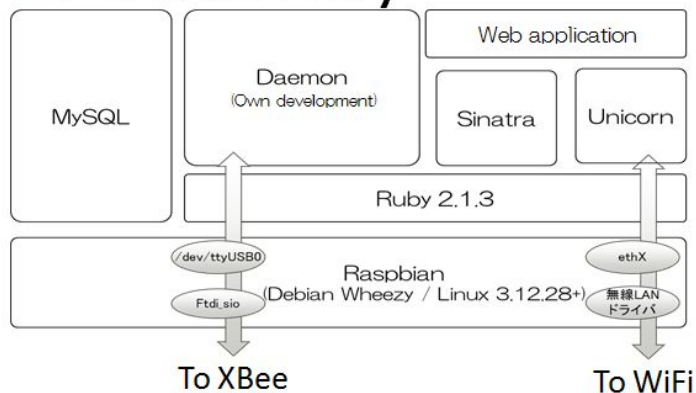


# Hardware configuration (Last years sample)

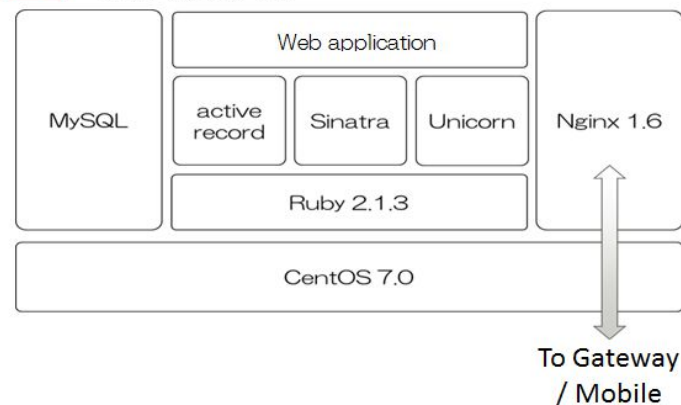


# Packages Layout (Last years sample)

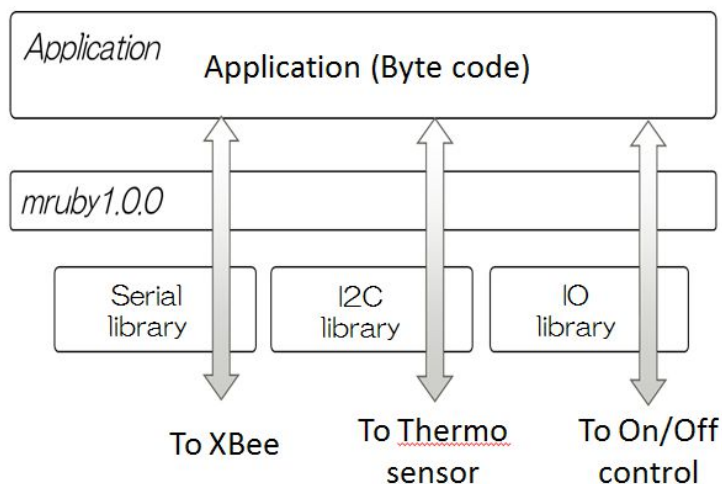
## □ Gateway



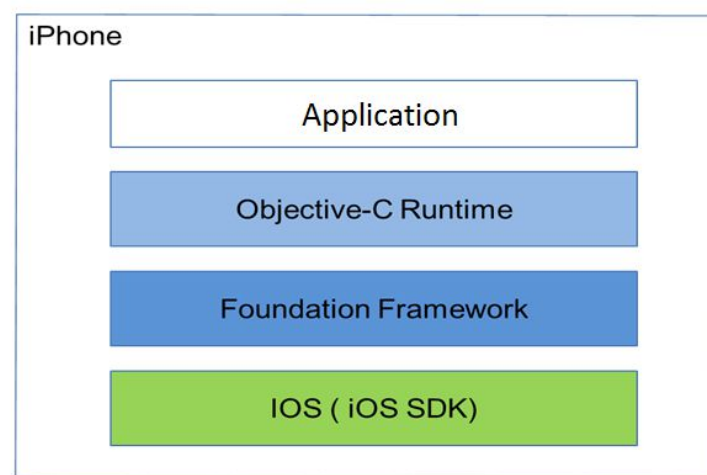
## □ Cloud



## □ Sensor



## □ Mobile





## Technical Issues - Sensor

- **What should be made with mruby and what should be made with C. (C library also uses memory area.)**
- **Isn't it possible to operate on more low-price micro controller. (e.g. Cortex-M0 based board)**
- **The difference in implementation by type of the micro controller**

## Technical Issues – Cloud

- **Not supported more than one gateway access yet. :p**
- **Not measured cloud performance for bulk access.**
- **Using polling communication system over HTTP between cloud and gateway, cloud and mobile.  
(That load seems high.)**
- **Security measures is needed.**

# Countermeasures - Sensor

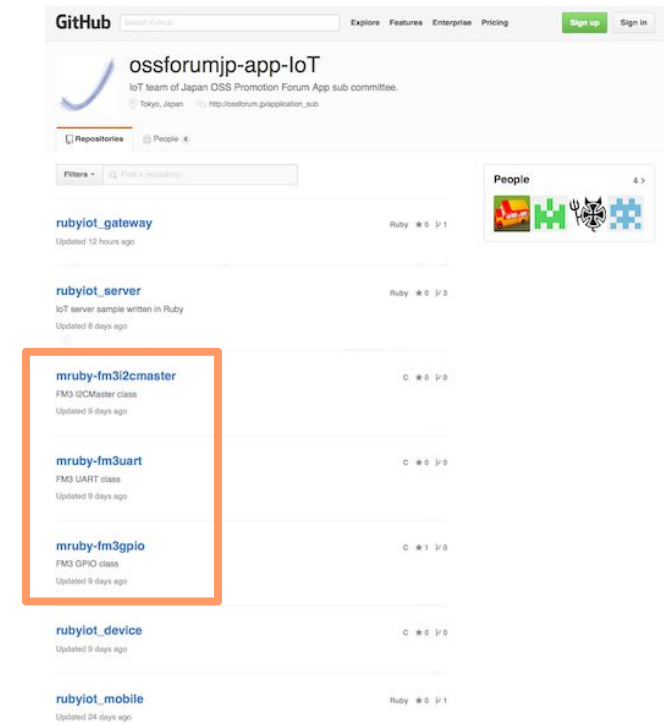
- What should be made with mruby and what should be made with C.  
(C library also uses memory area.)

⇒ We made some small mrbgems.

- 3 mrbgems for fm3
- Develop with C and operate register directly

⇒ We wrote main loop in C

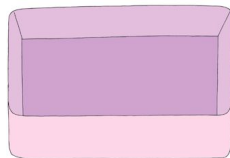
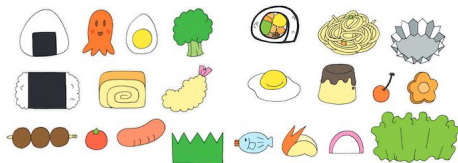
These source code are shared on Github  
<https://github.com/ossforumjp-app-LoT>



# Why create small mrbgems

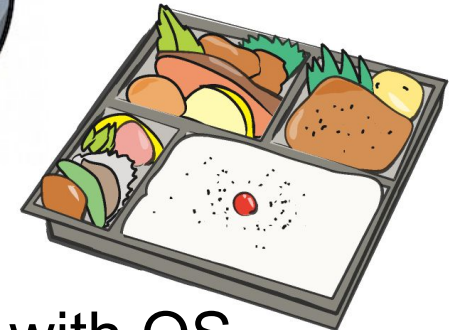
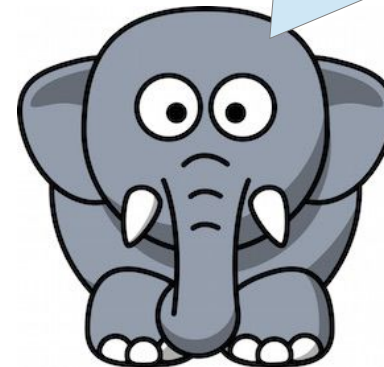
PC has more than 4000 times RAM of micro controller.  
Also Elephant weight is 4000 times of Kitten.

I want to arrange just little some dishes.



Like as micro controller with mruby and mrbgems.

I will have all in one bento.



Like as PC with OS.

Elephant: <http://free-illustrations.gatag.net/2013/11/16/050000.html>

Cat: [http://blogs.yahoo.co.jp/studio\\_robin2008/2060732.html](http://blogs.yahoo.co.jp/studio_robin2008/2060732.html)

Makunouchi Bento: <http://clipart-food.com/26packed-lunch/01-packed-lunch-dl.html>

Bento box and side dishes:

[ 幼児の学習素材館 ] <http://happyilac.net/asobi-obento.html> Illustration: がみさん

# Why main loop was written in C (1)

**We made the main loop with C.**

**That is why the problem may be occurred if it makes with mruby.**

**For example:**

main.rb (excerpt):

```
i = 0
loop do
  sensor = gpio.get_data if i % 100 == 0
  recvdata = serial.get_data
  if recvdata
    (0 .. 7).each { |x| gpio.set_data(x, 1) if recvdata[x] = 0x01 }
  end
  i += 1
  i = 0 if i > 100
  mssleep 1
end
```



# Why main loop was written in C (2)

```
> GC.start
```

```
=> nil
```

```
> p ObjectSpace.count_objects; 1000000.times { a = "a" }; ObjectSpace.count_objects
{:TOTAL=>2048, :FREE=>1083, :T_OBJECT=>3, :T_CLASS=>45, :T_MODULE=>7,
:T_ICLASS=>11, :T_SCLASS=>50, :T_PROC=>694, :T_ARRAY=>47, :T_HASH=>2,
:T_STRING=>89, :T_EXCEPTION=>1, :T_ENV=>15, :T_DATA=>1}
=> {:TOTAL=>2048, :FREE=>1120, :T_OBJECT=>3, :T_CLASS=>45, :T_MODULE=>7,
:T_ICLASS=>11, :T_SCLASS=>50, :T_PROC=>691, :T_ARRAY=>1, :T_HASH=>1,
:T_STRING=>103, :T_EXCEPTION=>1, :T_ENV=>14, :T_DATA=>1}
>
```

```
> GC.disable
```

```
=> false
```

```
> p ObjectSpace.count_objects; 1000000.times { a = "a" }; ObjectSpace.count_objects
{:TOTAL=>2048, :FREE=>859, :T_OBJECT=>3, :T_CLASS=>45, :T_MODULE=>7, :T_ICLASS=>11,
:T_SCLASS=>50, :T_PROC=>698, :T_ARRAY=>94, :T_HASH=>2, :T_STRING=>259,
:T_EXCEPTION=>1, :T_ENV=>18, :T_DATA=>1}
=> {:TOTAL=>1001472, :FREE=>153, :T_OBJECT=>3, :T_CLASS=>45, :T_MODULE=>7,
:T_ICLASS=>11, :T_SCLASS=>50, :T_PROC=>701, :T_ARRAY=>141, :T_HASH=>3,
:T_STRING=>1000336, :T_EXCEPTION=>1, :T_ENV=>20, :T_DATA=>1}
```

# Why main loop was written in C (3)

**Do you know where the problem is occurred?  
It is range.**

**The received data from serial is 8 bytes, if keeping receiving, range object is generated every 1ms.**

main.rb (excerpt):

```
i = 0
loop do
  sensor = gpio.get_data if i % 100 == 0
  recvddata = serial.get_data
  if recvddata
    (0 .. 7).each { |x| gpio.set_data(x, 1) if recvddata[x] = 0x01 }
  end
  i += 1
  i = 0 if i > 100
  mssleep 1
end
```

# Why main loop was written in C (4)

```
> p Time.now; 1000000.times { a = "a" }; Time.now
Fri Oct 30 19:09:24 2015
=> Fri Oct 30 19:09:25 2015

> GC.enable
=> true

> p Time.now; 1000000.times { a = "a"; GC.start }; Time.now
Fri Oct 30 19:10:05 2015
=> Fri Oct 30 19:11:58 2015
```

1sec

113sec

**A micro controller picks up events by loop processing.**

**If a literal string, array, hash, range etc. is used in the loop, RAM is consumed.**

**Or GC occurs frequently, and processing becomes slow.**

**So the main loop was written by C and a part equivalent to an event handler was written by mruby.**

# Mruby and C - Sensor

- **The advantage of mruby:**

- Simple and plain code. (in particular, string, network communication, etc.)
- Byte code separable.

- **The advantage of C:**

- Hardware access. (and able to the power management)
- Fixing resources.



- **Let's make small mrbgem. (having common api for many devices)**
- **Making main loop with C, logic with mruby.**



**The same code can execute on other devices by changing small mrbgems. And the logic part will code and replace easily.**

# Countermeasures - Cloud

**Cloud performance for bulk access...**

**Unfortunately, I didn't make for performance measurement.**

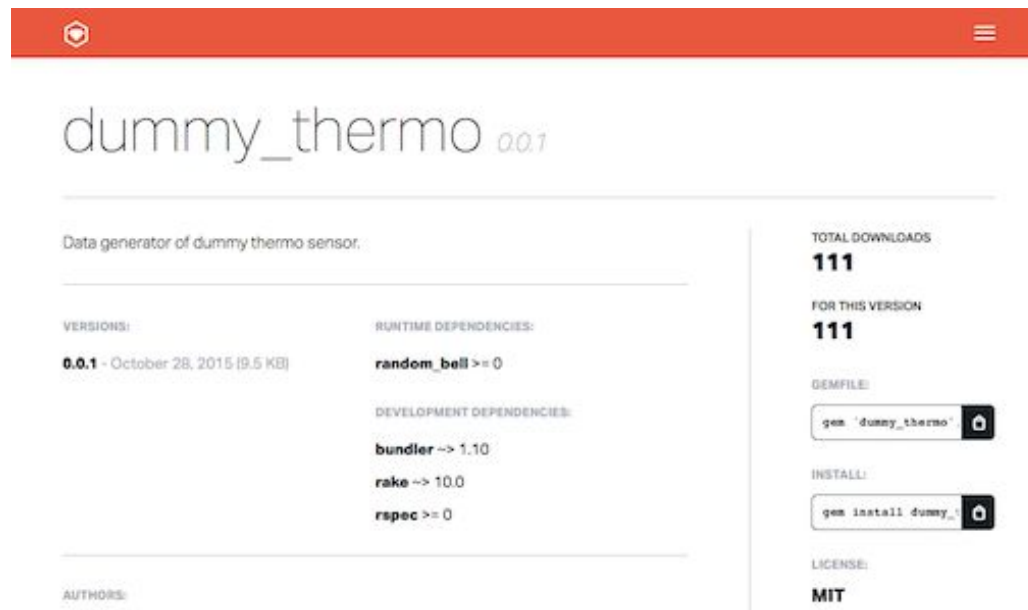
**We will report at the result report meeting of the OSS Promotion Forum in January 2016.**



# Addition

I developed sensor data generator for measurement of cloud performance.

There might be a person who uses it, so about 1 person for 1 million people, made it a gem and up to rubygems site.



## What we will do in next year

- **Now, the byte code is not separated. We will try to separate the byte code, and change the action of micro controller without compiling.**
- **We will use MQTT for server, gateway and mobile, to reduce the protocol overhead and to be more simple their codes.**

## • Contact

### **OSS Promotion Forum Application sub committee**

Web site: [http://ossforum.jp/application\\_sub](http://ossforum.jp/application_sub)

Github: <https://github.com/ossforumjp-app-IoT>

### **CEC**

Web site: <http://www.cec-ltd.co.jp/>

### **Tetsuya Hirota**

Facebook: <https://www.facebook.com/tetsuya.hirota>

Github: <https://github.com/constdrop>

# Thanks!



エルエスアイ開発研究所

**FUJITSU** 富士通アドバンスエンジニアリング

**HTKエンジニアリング**



*Japan OSS  
Promotion  
Forum*

**HITACHI**  
Inspire the Next  
日立ソリューションズ

**CEC**  
Computer Engineering & Consulting