

mruby

The programming language for IoT



Kyushu Institute of Technology
Kazuaki TANAKA

Ruby Association
mruby Forum

Embedded Software Development

Hardware
+
Software

Limited Resources

- * Memory
- * Processing power
- * Time (Real-time processing)

Realtime?

NOT processing speed!

Time constraints in response from
events.

i.e. “Deadline estimation”

Ruby
for
Embedded Software
Development

mruby

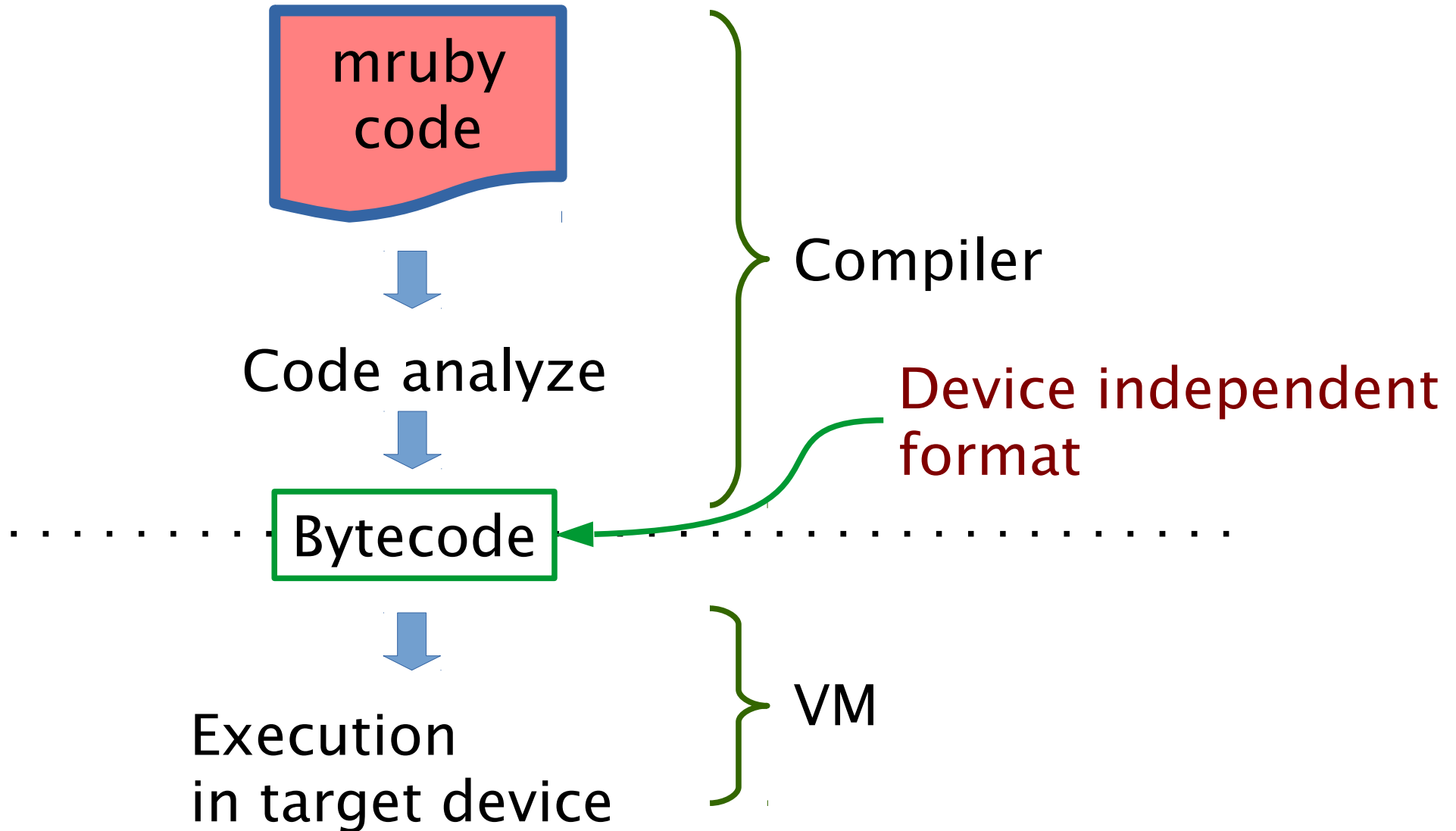
Why Ruby?

- Easy to read, easy to implement
- ISO/IEC 30170, JIS X 3017
- Open source software

mruby provides,,,

- Lightweight C interface
 - C-func. and Ruby-method mapping
- Less resources in execution
 - mruby compiler and VM
 - Small VM < 250KB
- Realtime
 - Incremental GC

Compiler and VM



Ruby and mruby

- Ruby
 - Widely used in Web application development
 - Rapid software development
- mruby
 - Reduce resources in execution
 - Compiler and VM, like Java environment

Software development using mruby

- * Build VM for target device
- * Implement mruby application
- * Test
- * Deploy

Test in early phase

- * VM for PC

- Test application on PC
with emulated driver

Using C

1. Implement C application
2. Download to target device
3. Test on target device
If errors, back to 1

Using mruby

0. Build VM for PC and for target device
1. Implement mruby application
2. Test on PC (with VM for PC)
If errors, back to 1
3. Download to target device
4. Test on target device (with VM for device)
If errors, back to 1

More advantages

- * Develop mruby application without target device
- * Separate hardware layer code and abstracted layer code
- * Hide source code

Get mruby

Latest



mruby

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

Versioned mruby
(with Tested libraries)



mruby

<http://forum.mruby.org/>

DEMO

Sample Program

```
a = 15  
p a * 2
```

```
a = 1.2  
p a * 2
```

```
a = "abc"  
p a * 2
```

```
a = [1, 2]  
p a * 2
```

Dynamic Execution

```
def func(n)
  return n+1
end

if rand(2)==1 then
  def func(n)
    return n*2
  end
end

p func(5)
```

Research Topic

mruby

```
i = 100
while i < 110 do
  puts i
  i = i + 1
end
```

```
i = 209
while i >= 200 do
  puts i
  i = i - 1
end
```