

化繁为简，以退为进

Shrink to Grow

博达

Daniel Bovensiepen

mruby

m for ...

minimal

< 100kB

< 40kB ram

m for ...

embeddedded

mod_mruby

```
LoadModule mruby_module modules/mod_mruby.so
AddHandler mruby-script .rb
```

```
backends = [
    "http://192.168.0.101:8888/",
    "http://192.168.0.102:8888/",
    "http://192.168.0.103:8888/",
    "http://192.168.0.104:8888/",
]

# write balancing algorithm here.

r = Apache::Request.new()

r.handler = "proxy-server"
r.proxyreq = Apache::PROXYREQ_REVERSE
r.filename = "proxy:" +
    backends[rand(backends.length)] +
    r.uri

Apache::return(Apache::OK)
```

sqlite3

```
$ cat test.rb
```

```
def fib(n)
  n < 2 ? n : fib(n - 1) + fib(n - 2)
end
```

```
create_function :fib do |n|
  fib(n)
end
```

```
$ LD_LIBRARY_PATH=. sqlite3
```

```
sqlite> .load libsqlite3ext_mruby.so.1.0.0;
sqlite> SELECT mrb_load('test.rb');
sqlite> SELECT fib(30);
832040
sqlite>
```

ArangoDB

mobiruby

~~Redis~~

Android

m for ...

modular

Standard Library

```
$ ./bin/mirb
```

```
mirb - Embeddable Interactive Ruby Shell
```

```
This is a very early version, please test and report errors.  
Thanks :)
```

```
> Thread
```

```
NameError: uninitialized constant Thread
```

```
> Fiber
```

```
NameError: uninitialized constant Fiber
```

```
> eval "puts 'hello world'"
```

```
NotImplementedError: eval not implemented
```

```
> print "Test"
```

```
NotImplementedError: print not available
```

```
> ObjectSpace
```

```
NameError: uninitialized constant ObjectSpace
```

```
> require 'file'
```

```
NoMethodError: undefined method 'require' for main
```

Fiber

Time

exit

sprintf

stdio

ObjectSpace

Socket

math

Struct

Random

eval

RegExp

这是如何实现的呢？

How is this possible?

C99

ISO 30170:2012

mruby

mirb

mrbc

Makefile

cmake

rake

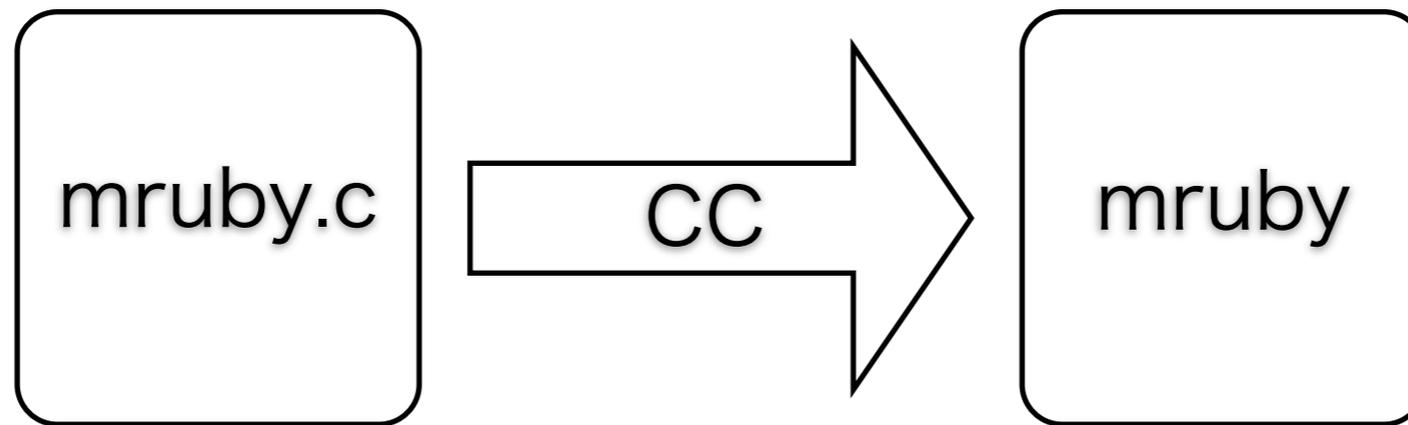
```
MRuby::Build.new do |conf|
  # load specific toolchain settings
  toolchain :gcc

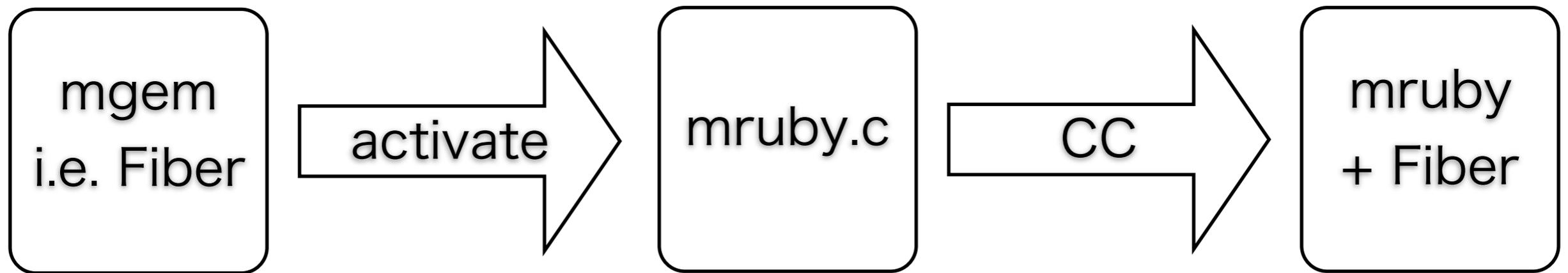
  # include the default GEMs
  conf.gembox 'default'
end

# Define cross build settings
MRuby::CrossBuild.new('32bit') do |conf|
  toolchain :gcc

  conf.cc.flags << "-m32"
  conf.linker.flags << "-m32"
end
```

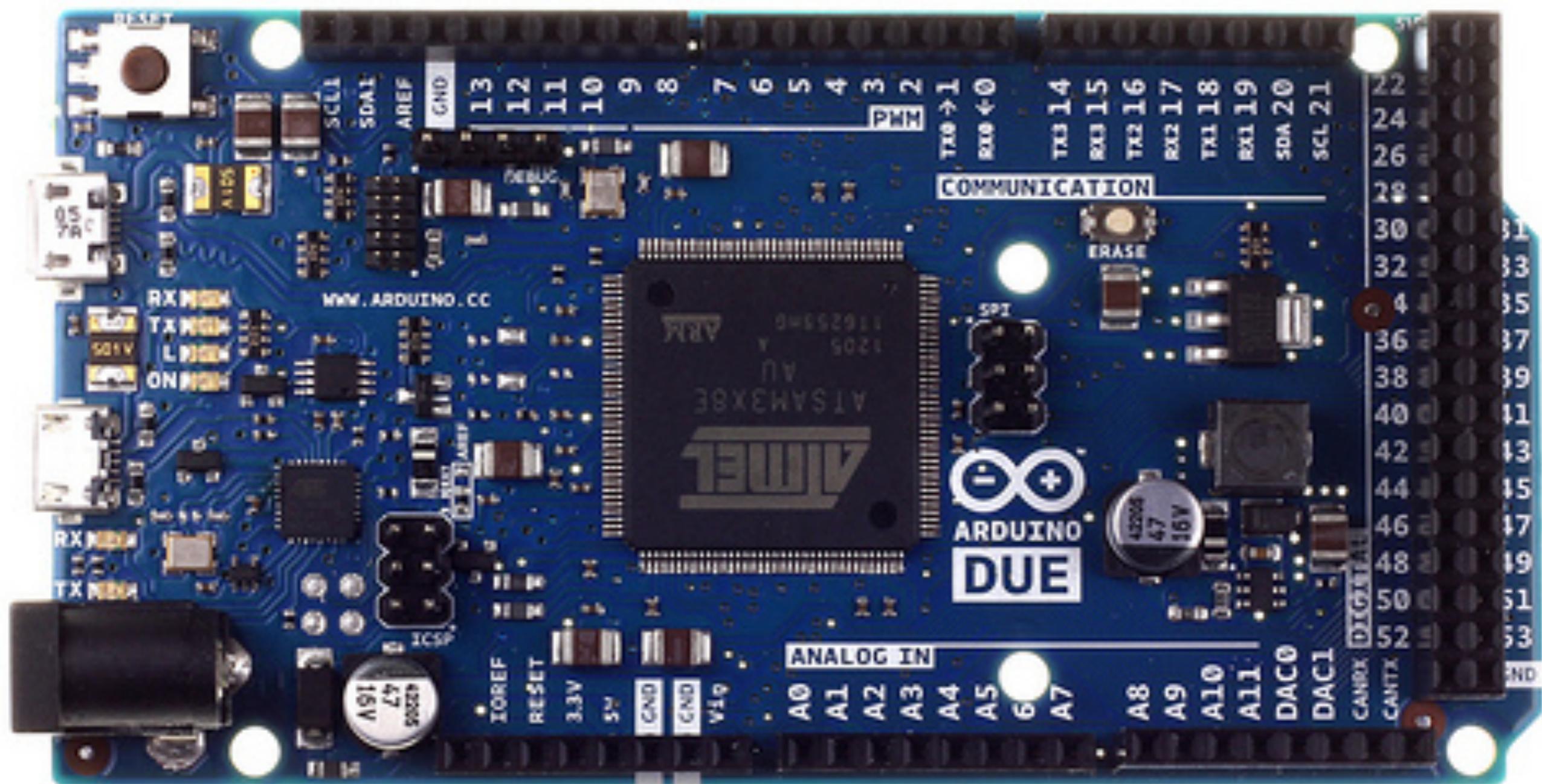
mrbgems





实用性

What can we do now?



ANALOG IN

A0
A1
A2
A3
A4
A5
6
A7

COMMUNICATION

TX0 → 1
RX0 ← 0
TX3 14
RX3 15
TX2 16
RX2 17
TX1 18
RX1 19
SDA 20
SCL 21

22
24
26
28
30
32
34
35
36
37
38
39
40
42
43
44
45
46
47
48
49
50
51
52
53
GND

ARM
1165534D
1205
AU
ATSAM3X8E
ATMEL

ARDUINO
DUE

WWW.ARDUINO.CC

16V
47
492008

16V
47
492008

ERASE

RESET

RX
TX
L
ON

RX
TX

IOREF
RESET
3.3V
5V
GND
GND
V19

SCL1
SDA1
AREF
GND

5V
GND

5V
GND

5V
GND

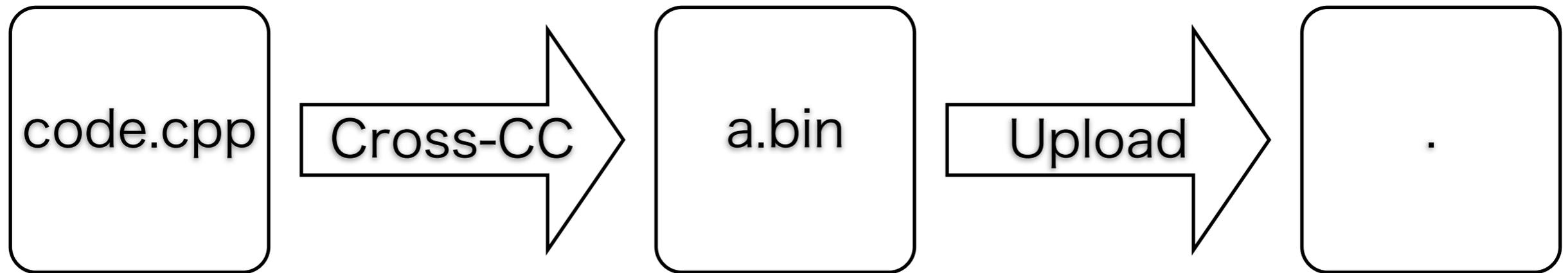
5V
GND

ARM Cortex M3

512kB Flash

96kB ram

Workflow



Arduino Code

```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

Boilerplate

```
#include "Arduino.h"
#include "../mruby_src/include/mruby.h"

mrb_state *mrbs;

/* Init phase of Microcontroller */
void
setup() { }

/* Process Loop */
void
loop() {
    mrbs = mrbs_open();

    /* Use mruby here */

    mrbs_close(mrbs);
}
```

API

```
/* ... */
```

```
 mrb_value  
mrb_ar(mrb_state *mrb, mrb_value self) {  
  mrb_int pin;  
  mrb_get_args(mrb, "i", &pin);  
  return mrb_fixnum_value(analogRead(pin));  
}
```

```
/* ... */
```

```
/* Integrate into mruby */
```

```
RClass *am = mrb_define_module(mrb, "Arduino");  
mrb_define_module_function(mrb, am, "analog_read", mrb_ar, ARGS_REQ(1));
```

```
/* ... */
```

rake

```
MRuby::CrossBuild.new("Arduino Due") do |conf|
  toolchain :gcc

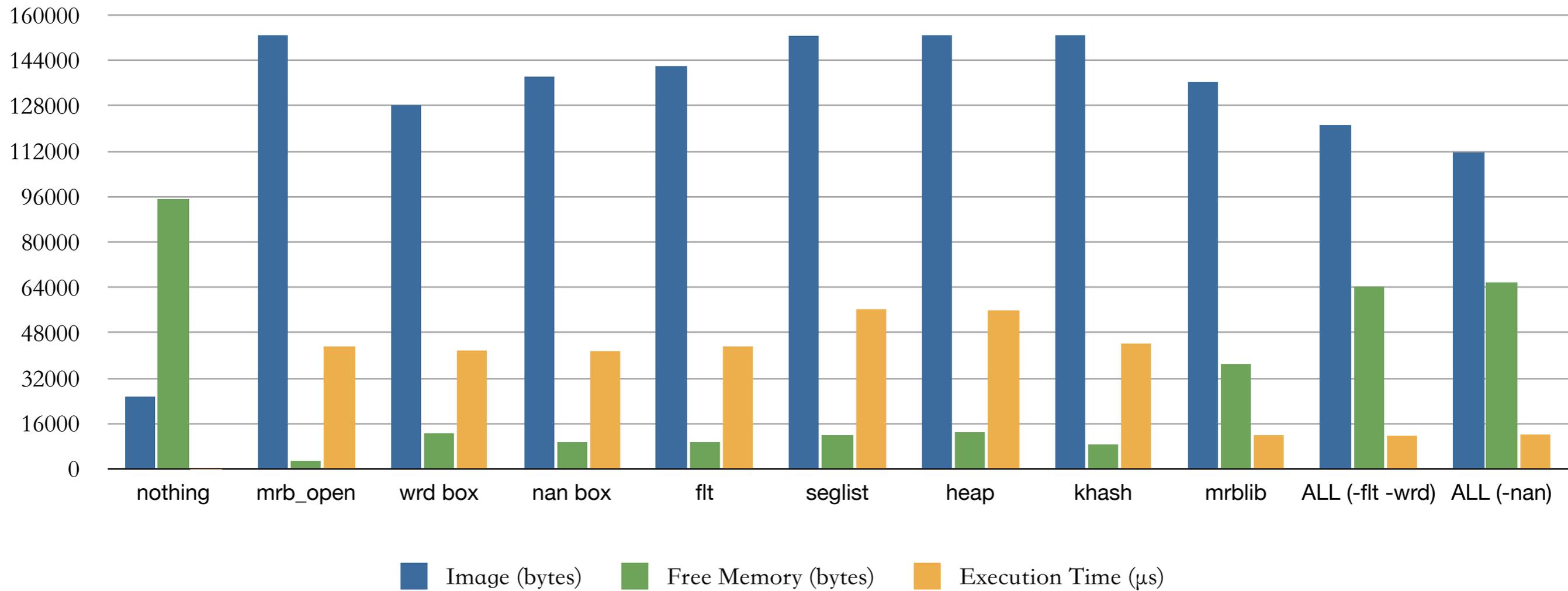
  ARDUINO_PATH = '/home/daniel/Downloads/arduino-1.5.2'
  BIN_PATH = "#{ARDUINO_PATH}/hardware/tools/g++_arm_none_eabi/bin"
  SAM_PATH = "#{ARDUINO_PATH}/hardware/arduino/sam"
  TARGET_PATH = "#{SAM_PATH}/variants/arduino_due_x"

  conf.cc do |cc|
    cc.command = "#{BIN_PATH}/arm-none-eabi-gcc"
    cc.include_paths = [ "#{SAM_PATH}/system/libsam -I#{SAM_PATH}/system/CMSIS/CMSIS/Include/",
                        "#{SAM_PATH}/system/CMSIS/Device/ATMEL/",
                        "#{SAM_PATH}/cores/arduino -I#{TARGET_PATH}",
                        "#{MRUBY_ROOT}/include" ]
    cc.flags << '-Os -w -ffunction-sections -fdata-sections -nostdlib ' +
                '--param max-inline-insns-single=500 -Dprintf=iprintf ' +
                '-mcpu=cortex-m3 -DF_CPU=84000000L -DARDUINO=152 ' +
                '-D__SAM3X8E__ -mthumb -DUSB_PID=0x003e -DUSB_VID=0x2341 -DUSBCON'
    cc.compile_options = "%{flags} -o %{outfile} -c %{infile}"
  end

  conf.archiver do |archiver|
    archiver.command = "#{BIN_PATH}/arm-none-eabi-ar"
    archiver.archive_options = 'rcs %{outfile} %{objs}'
  end

  conf.bins = []
end
```

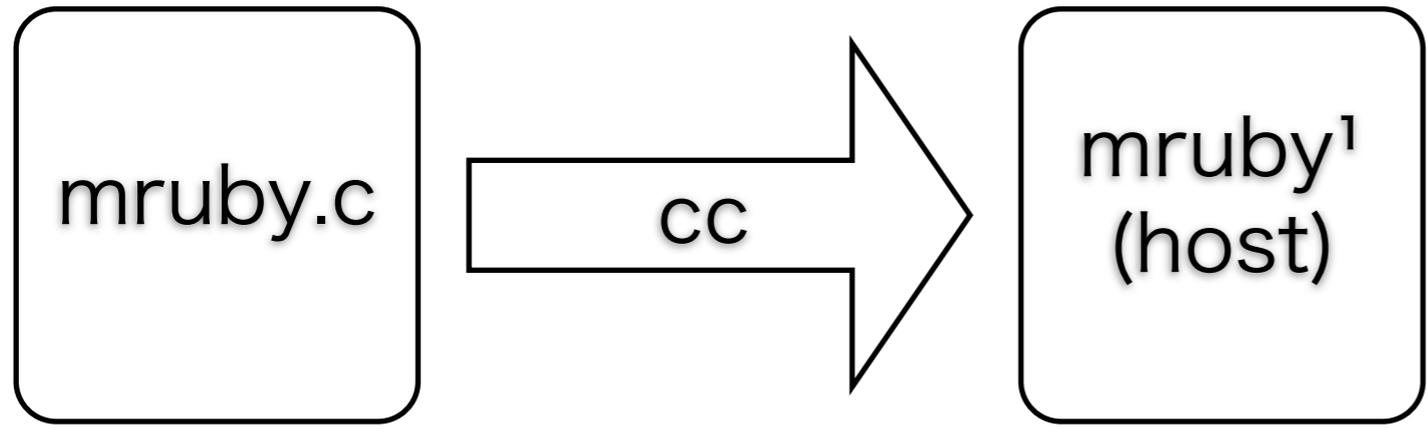
Shrink

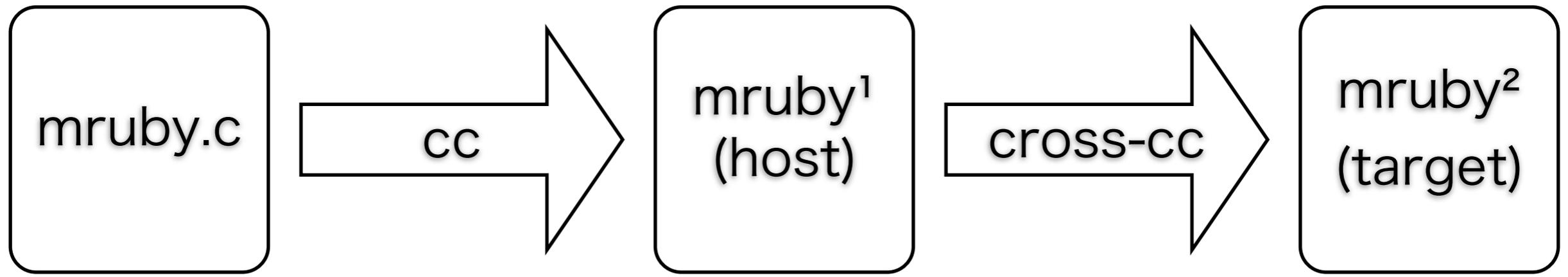


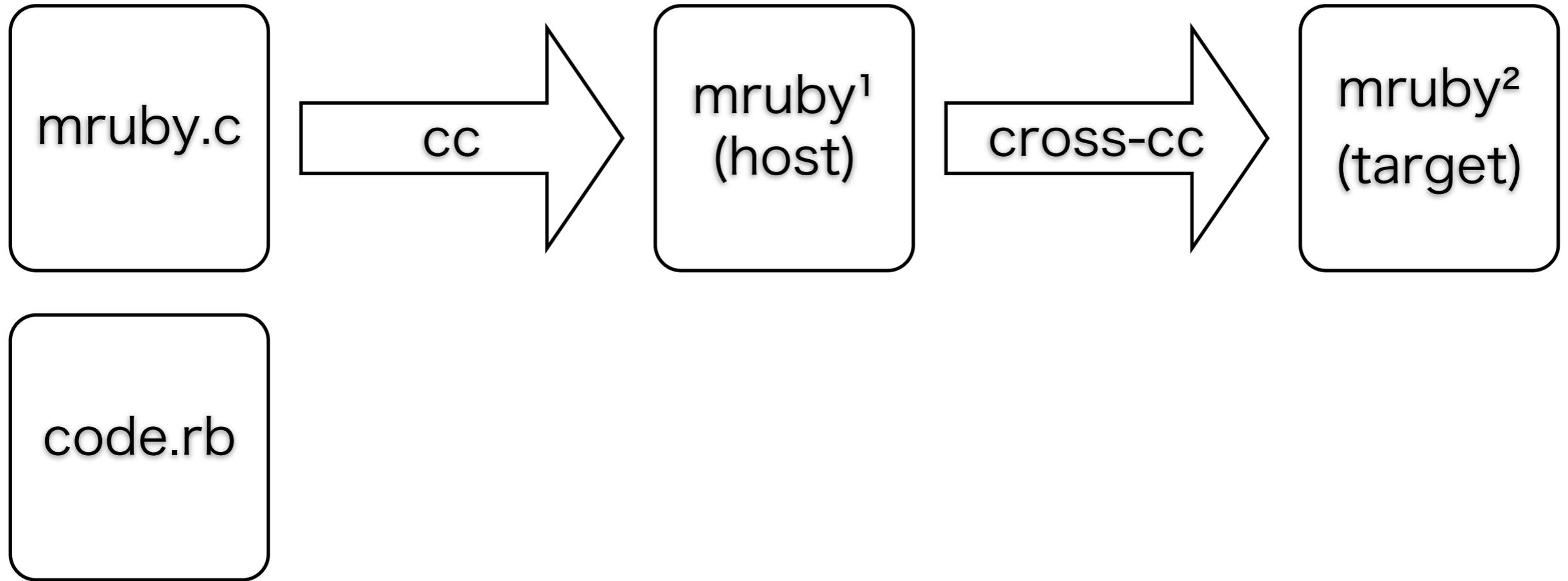
512kB Flash
~86kB used

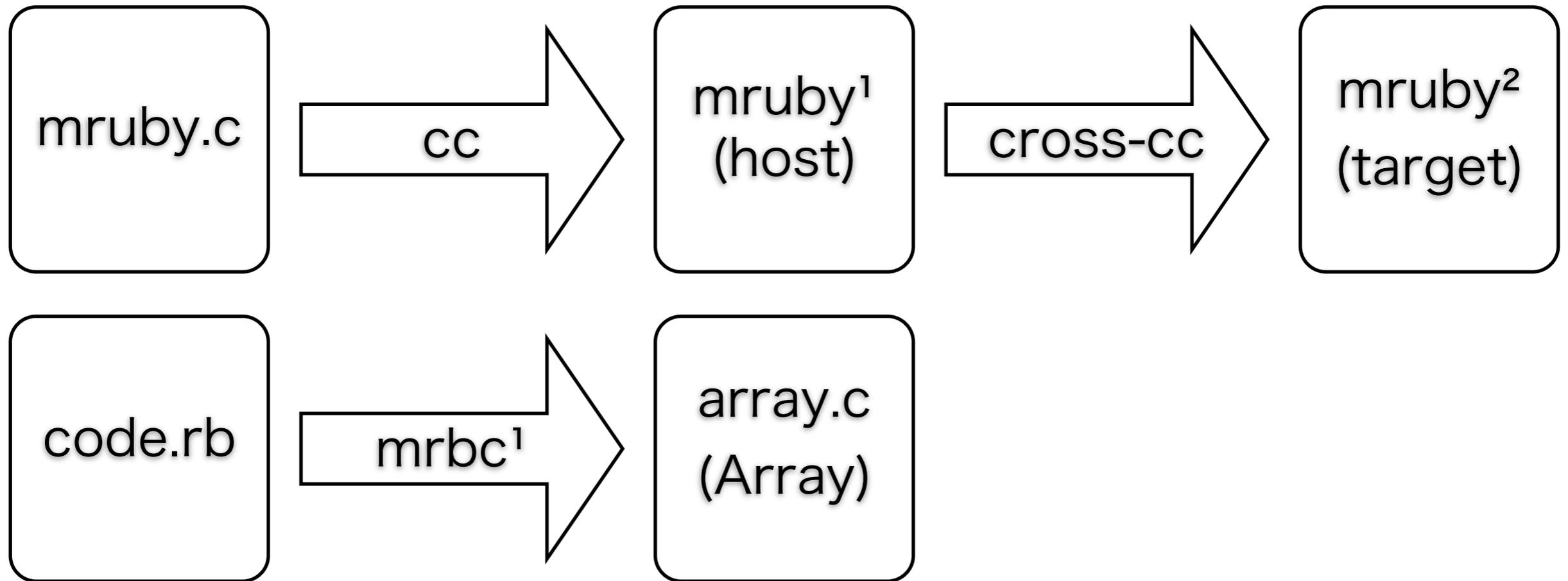
96kB ram
~31kB used

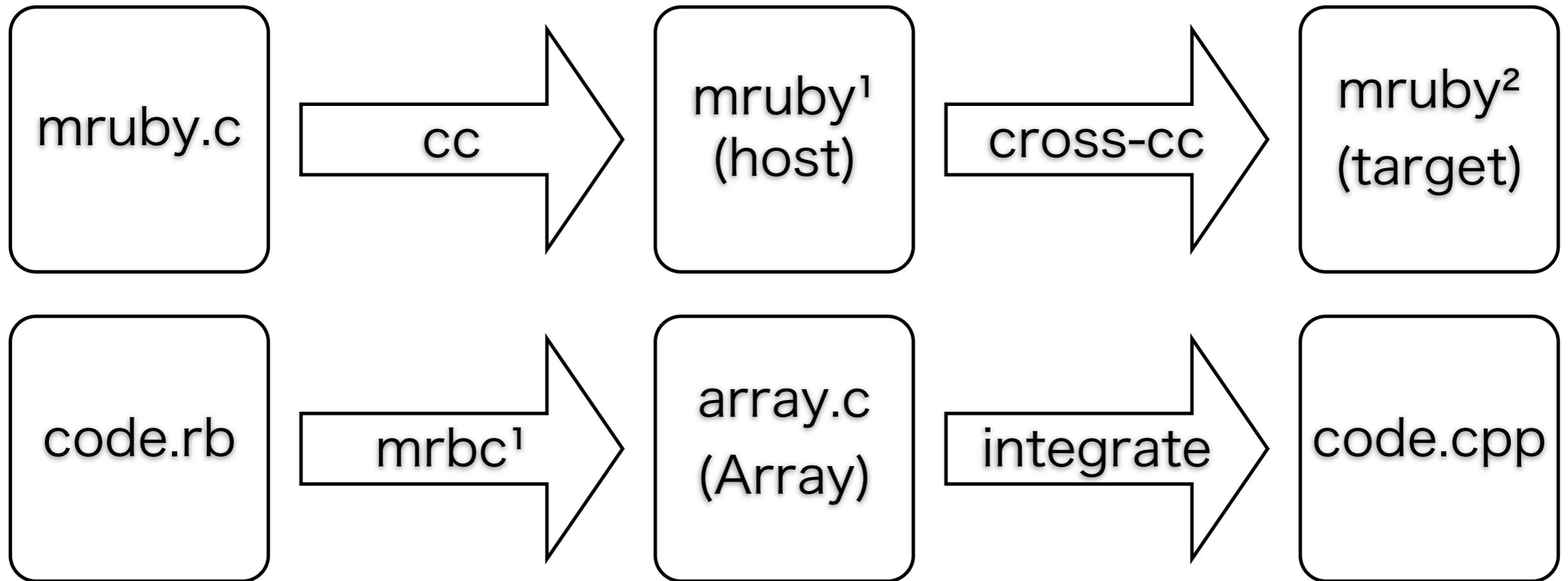
mruby.c

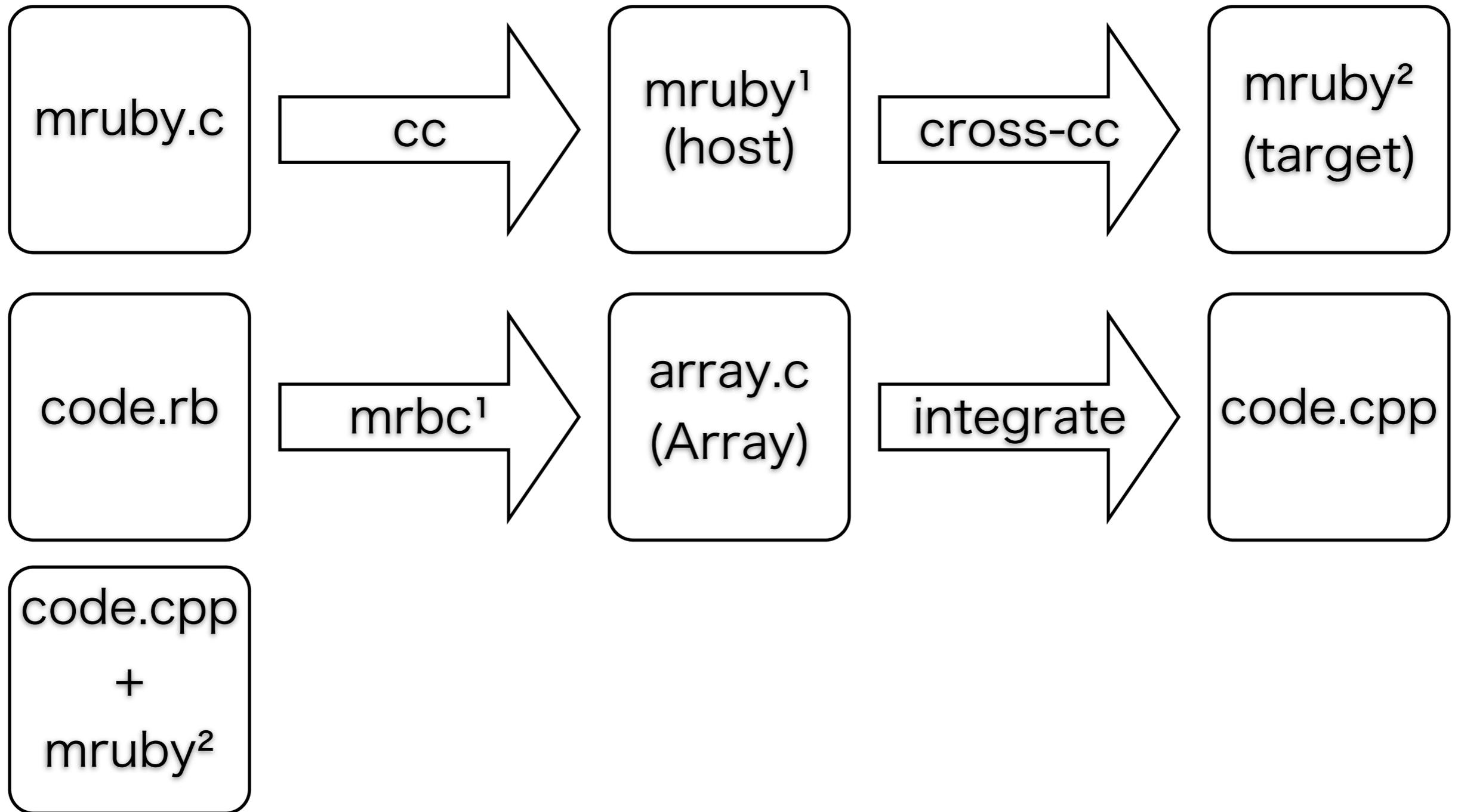


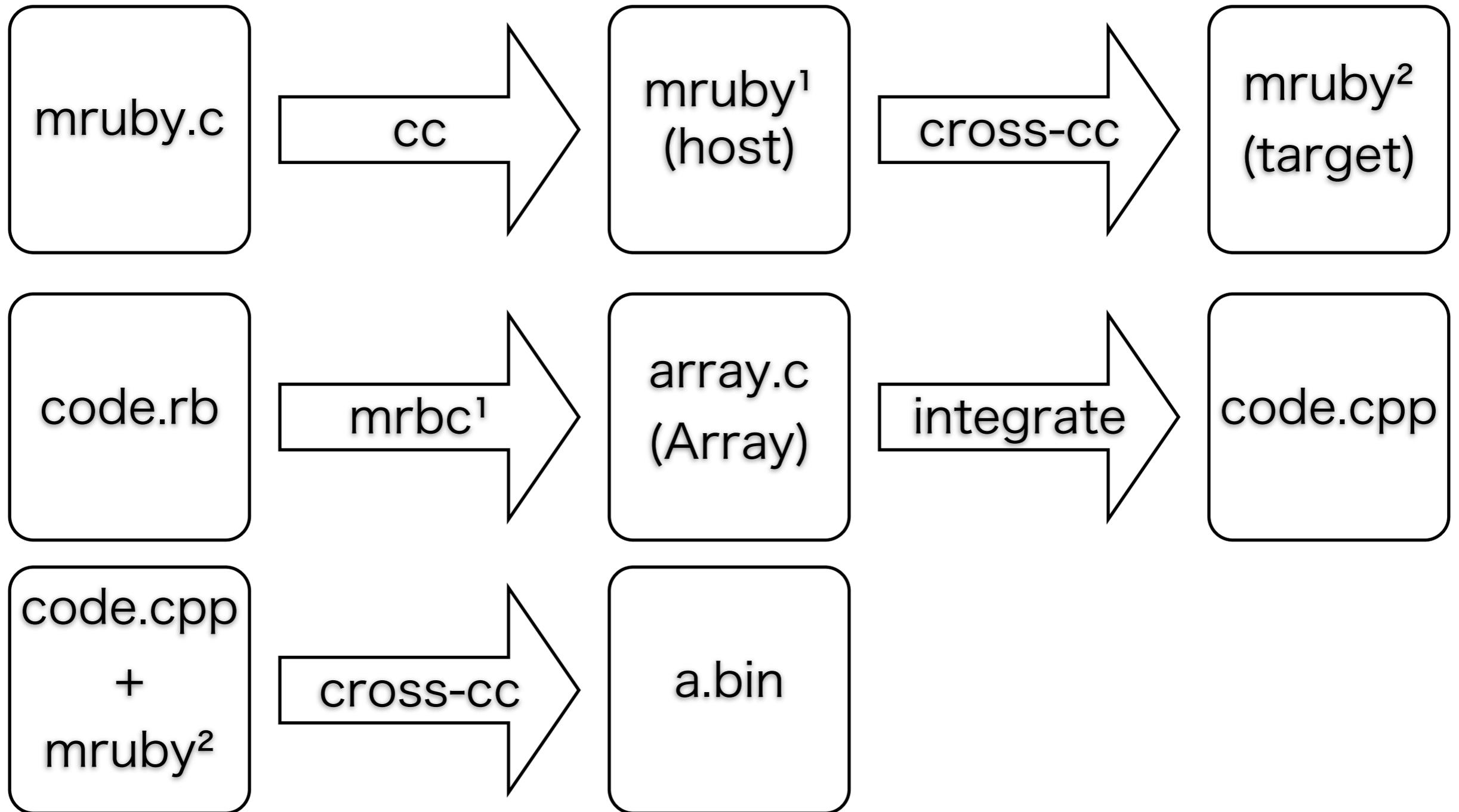


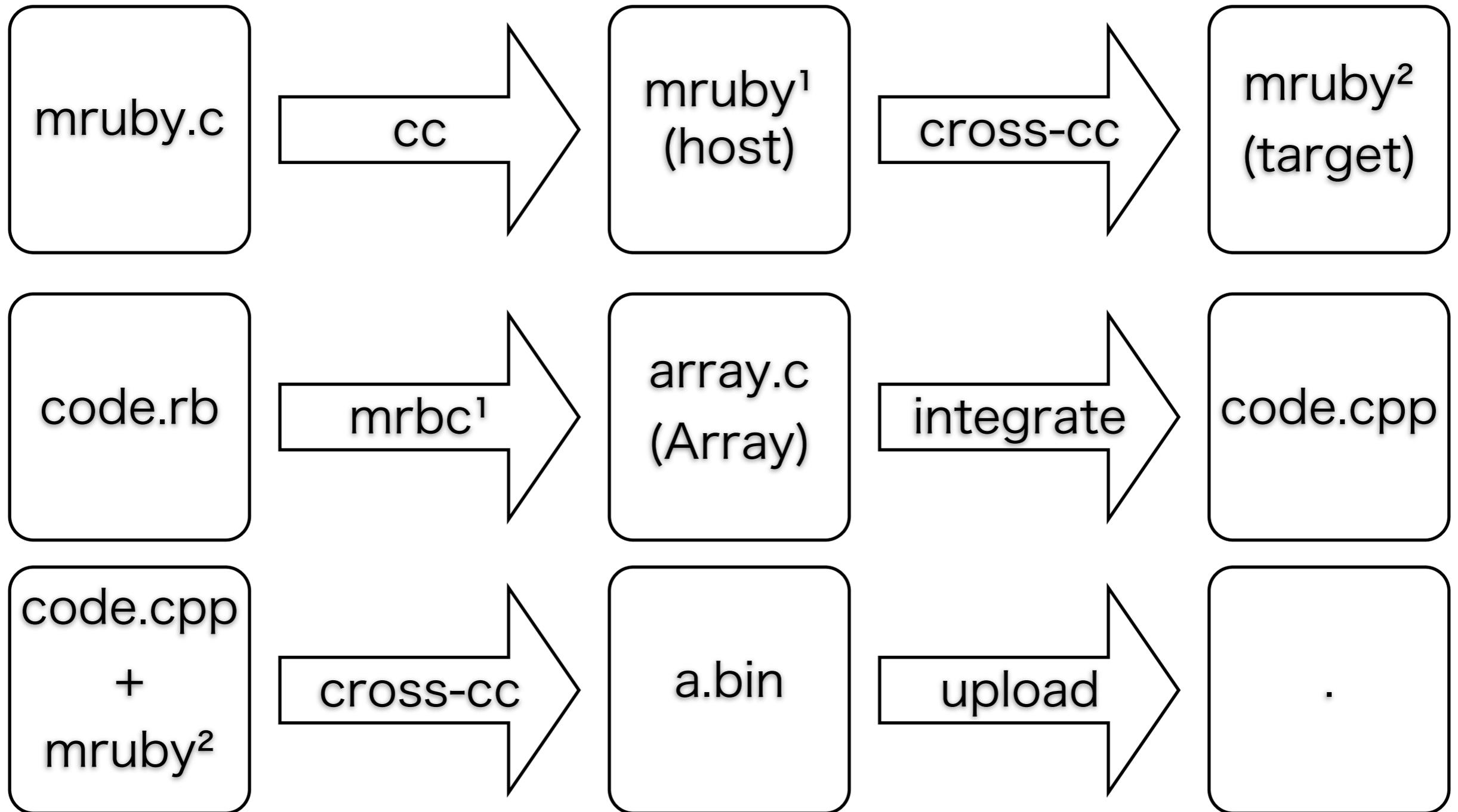






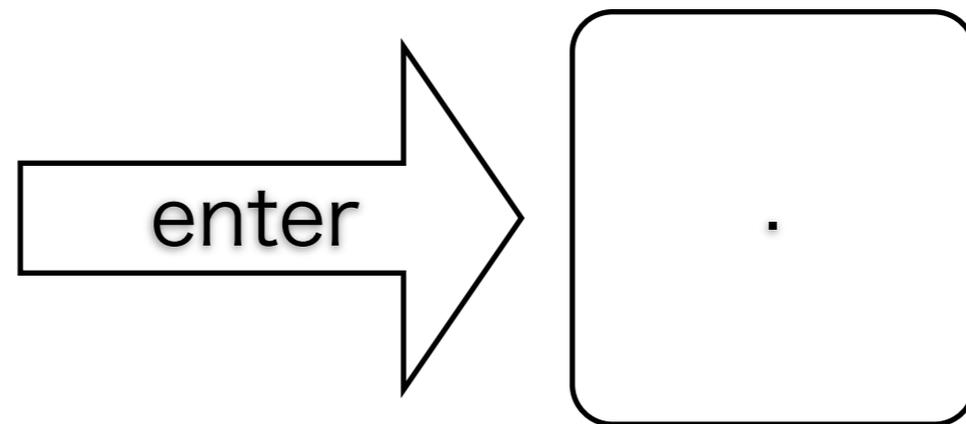






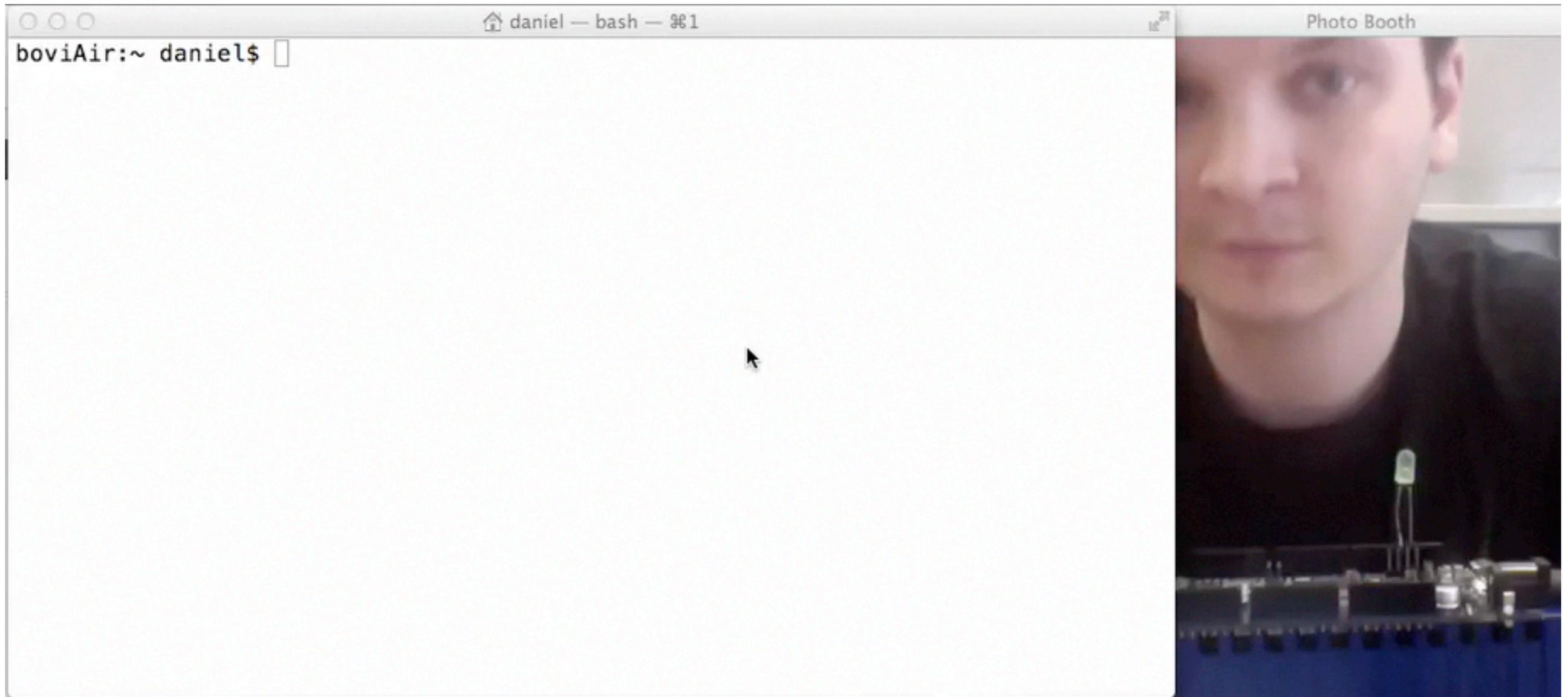
mirb on ARM

mirb on ARM



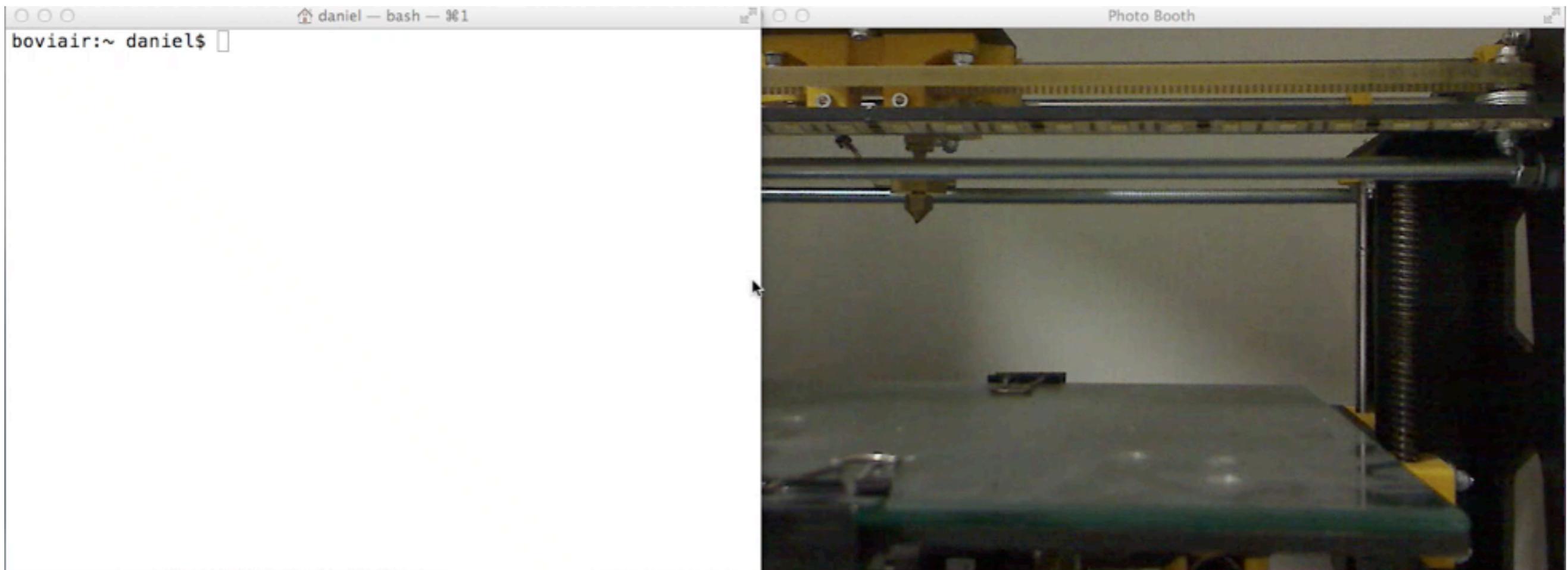
LED switch

LED switch



Motor Controller

Motor Controller



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

mod_mruby https://github.com/matsumoto-r/mod_mruby

sqlite3 <https://github.com/spiritloose/sqlite3ext-mruby>

ArangoDB <http://www.arangodb.org>

Redis <https://github.com/antirez/redis/pull/848>

mobiruby (iOS) <http://mobiruby.org>

Android <http://podtynnyi.com/2012/11/29/build-mruby-for-android/>

RegExp

Overview <http://blog.mruby.sh/201302190729.html>

Oniguruma <http://www.geocities.jp/kosako3/oniguruma/>

Henry Spencer <http://www.arglist.com/regex>

Five Ways to execute mruby <http://blog.mruby.sh/201207020720.html>

谢谢
Thanks

博达
Daniel Bovensiepen

<http://mruby.sh>