

# Introduction to RubyCocoa

(CocoaHead:Thinkboy)



source: <http://www.rubycocoa.com/>

# Aganda

- Problem?
- What?
- Why?
- How?
- Where?
- Demo?

Neither true or false,  
just a matter of choice!

# Problem?

- I love Mac OSX GUI (Aqua)
- I love Cocoa
- I love Xcode & Interface Builder
- But programming in Objective-C may be a bit of 'curve' to get started

# From Objective-C

```
@interface Controller : NSObject
{
    IBOutlet NSWindow *itsWindow;
}
- (IBAction)changeTransparency:(id)sender;
@end

@implementation Controller
- (IBAction)changeTransparency:(id)sender
{
    [itsWindow setAlphaValue:[sender floatValue]];
    [itsWindow display];
}
@end
```

# To RubyCocoa

```
class Controller < OSX::NSObject
  IBOutlet :itsWindow
  def changeTransparency(sender)
    @itsWindow.setAlphaValue(sender.floatValue)
    @itsWindow.display
  end
end
```

# Differences?

- ruby vs many other programming lang
- ruby vs objective-c vs java
- more comparison
- benchmark comparison

Again, it is just a matter of choice  
and, perhaps, depending on  
requirement.

# What is Ruby?

- Created by Yukihiro "Matz" Matsumoto in 1993
- PURE object-oriented
- Dynamic typing (duck typing)
- Interpreted language (scripting language)
- Designed for programmer productivity and fun
- Follow the principle of least surprise (POLS)

"if it walks like a duck,  
and talks like a duck,  
then it might as well be a duck."

~ Dave Thomas

in short, when u look at the  
syntax, u feel like home.  
that's all.

# Ruby Learning Trail

- quick tutorial
- interactive practice
- Programming Ruby by Dave Thomas
- Ruby Pocket Reference (O'Reilly 2007)
- more recommendation

# When to use RubyCocoa?

Ruby

Ruby

Domain Specific  
Application

JRuby

RubyCocoa

Runtime Bridge

Java

Objective-C

Core/ Framework

C

C

System/Native

# What is RubyCocoa?

- created by Hisakuni Fujimoto in 2001
- a bridge that makes it possible for Ruby scripts to access Objective-C objects
- like SWIG? but better!
- automatically create Ruby proxy objects that are bridged to Objective-C classes.
- forwards Ruby messages to the instances of Objective-C classes.
- so what? mixing Ruby and Objective-C in the same source files.

Ruby and Objective C share a common ancestor in Smalltalk

tool like SWIG which reads C and C++ source files and automatically writes the glue code that wraps C functions for other languages. Objective-C uses for dynamic message

by automatically creating Ruby proxy objects that are bridged to Objective-C classes.

forwards Ruby message to the instances of these Objective-C classes.

# Why RubyCocoa?

- officially supported by Apple
- full support in Xcode3 (e.g. color syntax, auto complete, formatting)
- supports all important features of Cocoa, such as key-value coding, key-value observing, Core Data, the document architecture, notifications, and undo management.
- standard package in 10.5: ruby 1.8.6, rubycocoa, RubyGems, rake , Rails , Mongrel , Capistrano, Ferret, OpenID, sqlite3-ruby, libxml-ruby, dnssd, net-ssh and net-sftp

# RubyCocoa How-To?

1. install RubyCocoa 0.13.0
2. create Cocoa-Ruby Application in Xcode3
3. add ScriptingBridge.framework
4. create new ruby controller by subclass  
NSObject
5. create label and button
6. connect controller to `ib_outlet` & `ib_action`

# ITunesController.rb

```
require 'osx/cocoa'
include OSX

class ITunesController < NSObject
  ib_outlet :text_field

  def show_version(sender)
    iTunes =
      SBApplication.applicationWithBundleIdentifier:'com.apple.iTunes'
    @text_field.setStringValue("iTunes version:
#{iTunes.version}")
  end
  ib_action :show_version
end
```

# Quick Tutorial?

## 1. This (Shortest) Tutorial:

<http://cocoalocker.blogspot.com/2007/11/ruby-cocoa.html>

## 2. Other Tutorial from YouTube

[http://www.youtube.com/watch?v=7q\\_DD-W6-ol&eurl=http://technorati.com/videos/youtube.com%2Fwatch%3Fv%3D7q\\_DD-W6-ol](http://www.youtube.com/watch?v=7q_DD-W6-ol&eurl=http://technorati.com/videos/youtube.com%2Fwatch%3Fv%3D7q_DD-W6-ol)

# RubyCocoa ri Doc?

- `install rubycocoa-0.13.0.tar` manually.
- `ruby install config`
- `ruby install doc`
- in the folder `/RubyCocoa-0.13.0/framework/bridge-doc/`

# Where to Fish?

- Warn! apple official doc is not so accurate!!
  - doc example does not work on Xcode3
- lets learn by example:
  - /Developer/Examples/Ruby/RubyCocoa
  - Must Read: official RubyCocoa site
  - Mailing list: rubycocoa-talk

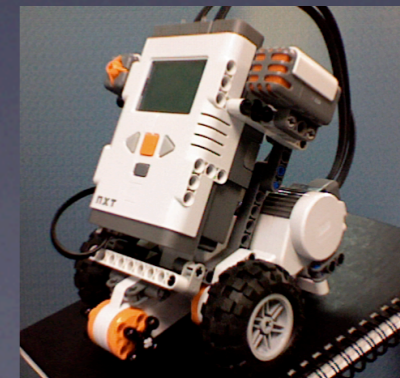
NXT Demo!

# Demo Setup

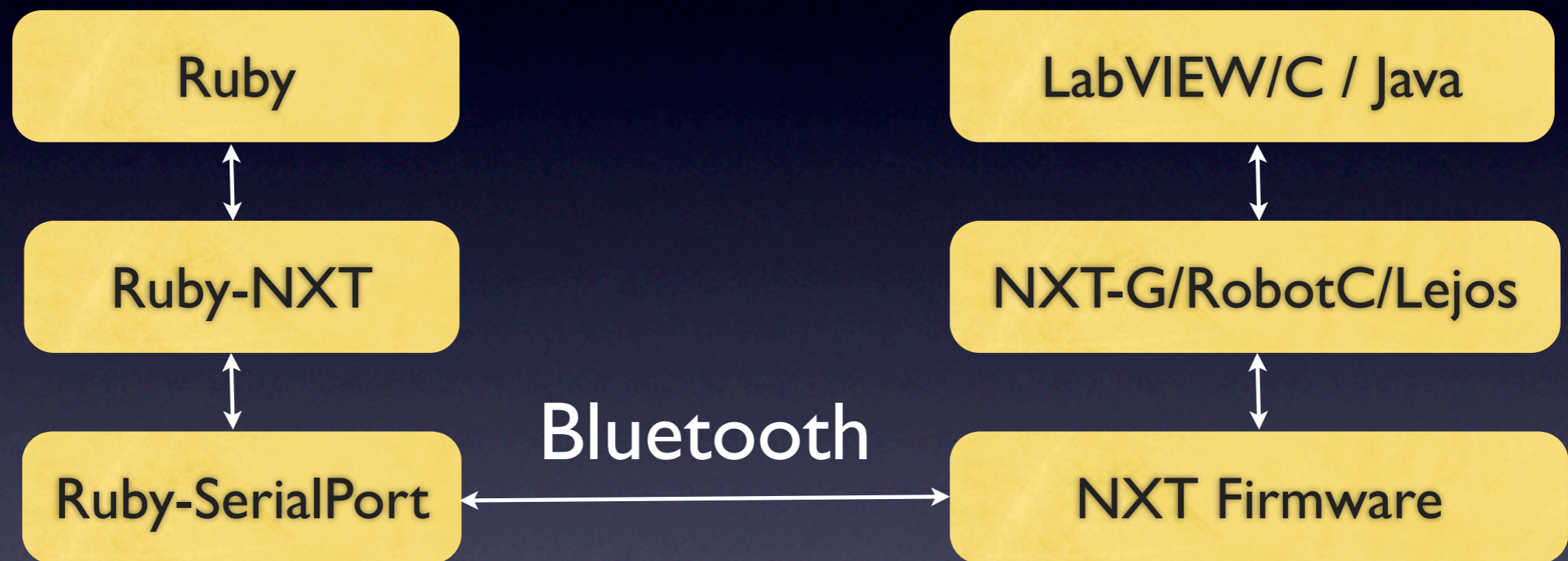
- RubyCocoa that remote control NXT
- install Lego Mindstorms NXT 1.1 software
- setup bluetooth device from Mac to NXT
- gem install ruby-nxt 0.8.1
- download and compile ruby-serialport 0.6
- create a RubyCocoa apps in Xcode3



source: [www.bluetooth.com](http://www.bluetooth.com)



# Demo Architecture



# NxtController.rb

```
require 'osx/cocoa'
class NxtController < OSX::NSObject

  ib_outlet :command_mode
  ib_outlet :run_state
  ib_outlet :tacho_count
  ib_action :move_forward

  def initialize
    @nxt = NXTComm.new($DEV)
  end

  def move_forward
    @command_mode.setStringValue("Moving forward...")
    command = Commands::Move.new(@nxt)
    command.ports = :a, :b, :c
    command.duration = {:seconds => 1}
    command.next_action = :brake
    @run_state.setStringValue("Run State: #{command.message1.inspect}")
    @tacho_count.setStringValue("Tacho Count: #{command.message2.inspect}")

    command.duration = :unlimited
    command.start
    sleep(1)
    command.stop
  end
  ....

```

\* ported from the tk example of ruby-nxt

# More About NXT

- NXT blog

<http://nxtasy.org/>

- NXT programming Software

[http://www.teamhassenplug.org/NXT/  
NXTSoftware.html](http://www.teamhassenplug.org/NXT/NXTSoftware.html)

- Interesting Projects

<http://www.youtube.com/watch?v=I8VvTENzPGI>

<http://www.youtube.com/watch?v=0sIIQ6S3yuo>

<http://www.youtube.com/watch?v=s0G35-xoRfA>



Q & A  
Email: [manchi.leung@gmail.com](mailto:manchi.leung@gmail.com)