

**Java Native GPIO Operations
Using Device I/O API**

**What is IoT, RaspberryPi
and Lab to build a Candy Vending Machine**

About me...



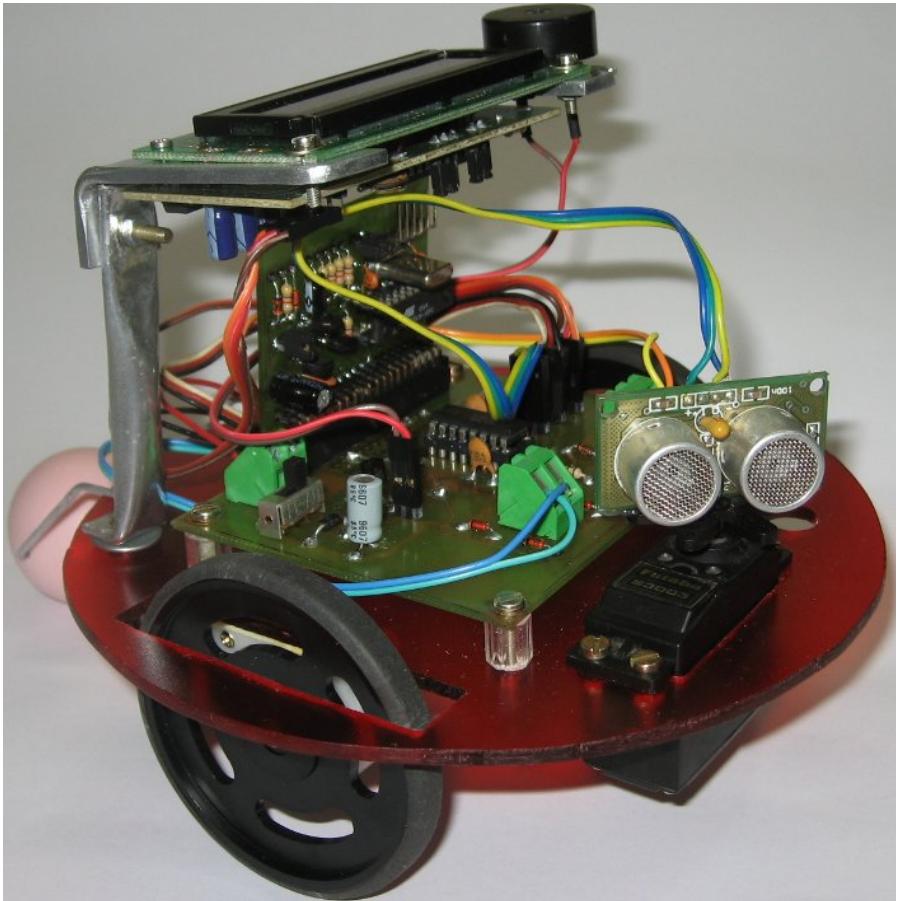
@jeffprestes
Developer Advocate
Java, PHP, JavaScript Developer

Braintree_Dev.
A **PayPal** Company

Agenda

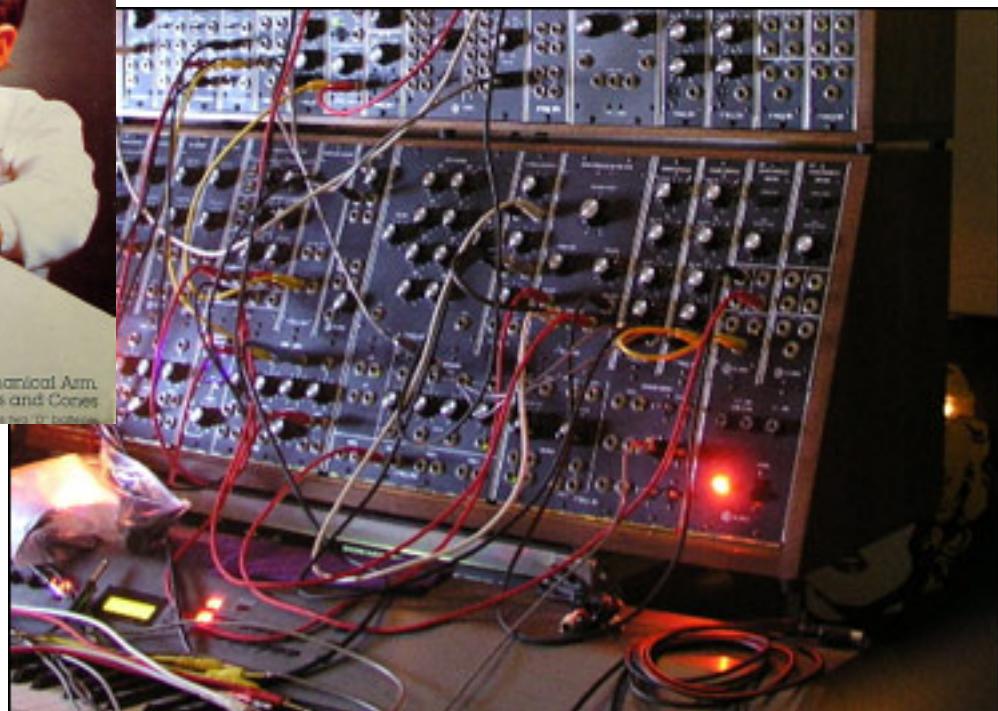
- What is not **IoT**
- What is **IoT**
- **Polymath** required skills
- Small history of **RaspberryPi**
- **RaspberryPi** Components and slots
- **Raspbian** – a Debian/Linux to RasberryPi
- Java support to SOC devices
- Java Device I/O
- Lab
- Configure **WiFi** network (static / dhcp / resolv.conf)
- Upgrade the **Raspbian**
- Install **WiringPi**
- Install **Java** for Embedded devices

This is not IoT... (IMHO)



This is so cool but this is Electronic!!

And this is not new...



(who has gray hair can say... P)

What is IoT?

Internet

Million of Web
Services

(PayPal, Twitter,
Google, Netflix,
Facebook)

+

Things

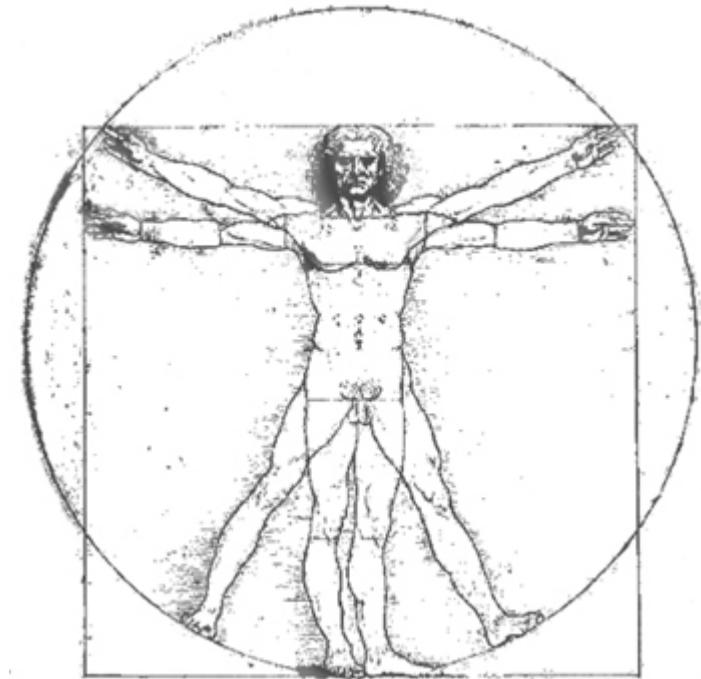
Billions of sensors,
motors, displays,
appliances, toys,
cars, stores, robots

And to work with IoT you're going to need more skills

You're going to need to a Polimath

A polymath (Greek: πολυμαθής, *polymathēs*, "having learned much")^[1] is a person whose expertise spans a significant number of different subject areas; such a person is known to draw on complex bodies of knowledge to solve specific problems. The term was first used in the seventeenth century; the related term, polyhistor, is an ancient term with similar meaning.

(Source: Wikipedia - <http://en.wikipedia.org/wiki/Polymath>)



L da Vinci



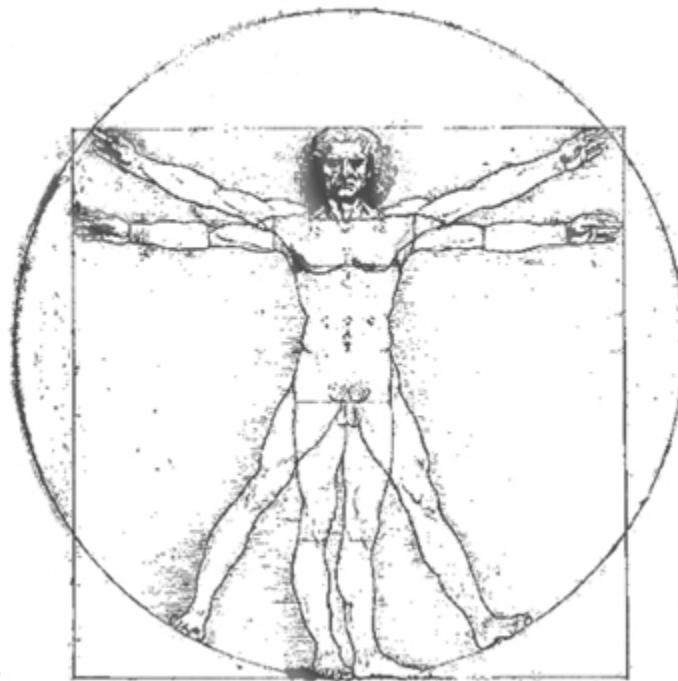
Italian Polymath. What was da Vinci? Scientist?
Engineer? Mathematic? Painter? Sculptor? Musician?
Botanic? Anatomist?

Source: Wikipedia - http://en.wikipedia.org/wiki/Leonardo_da_Vinci



Server Side

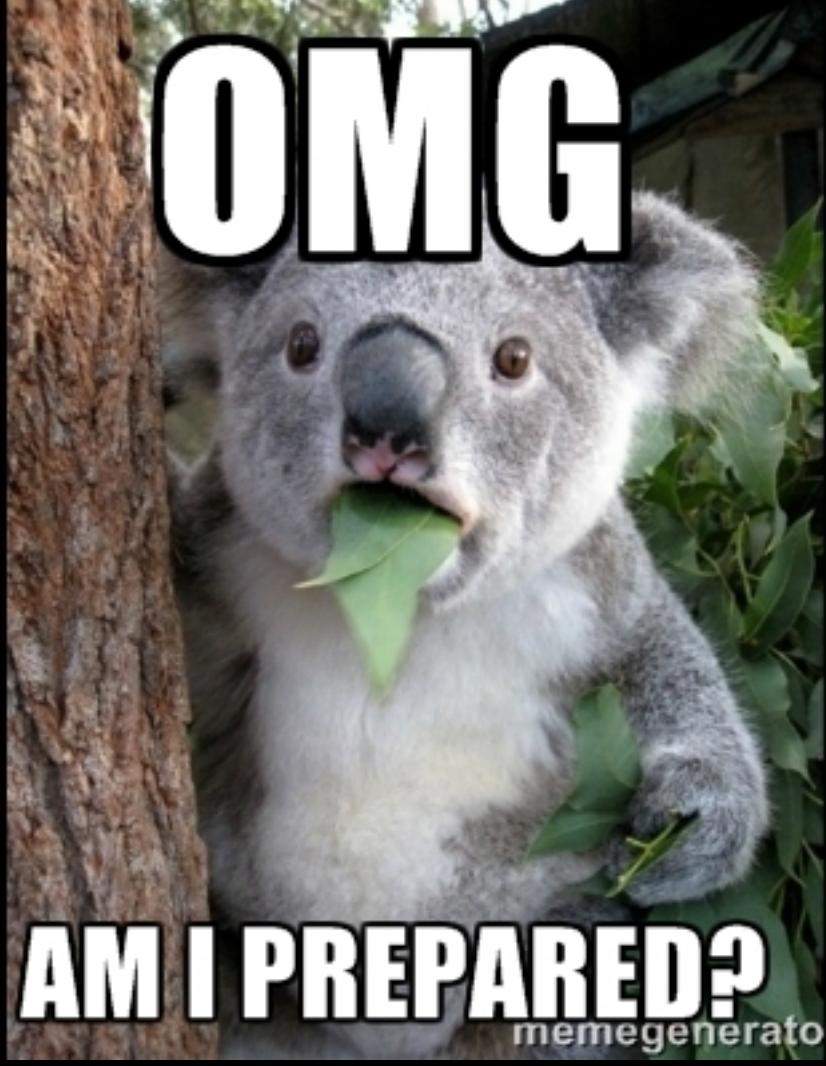
Client/Desktop



Eletronics
(sometimes mechanic
and sculptor on wood or iron)

Mobile
(Beacons e Weareables)

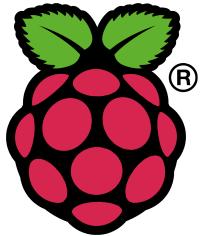
Operation Systems



OMG

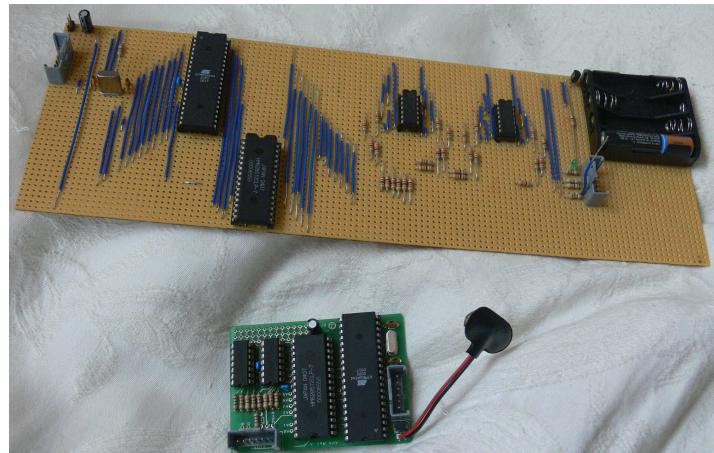
AM I PREPARED?

memegenerator.net



History

- Designed in UK, **University of Cambridge, 2006**, to be a chip alternative to computers to students. Also a way to students rediscover how cool is to work with Robotic.
It has been projected to educational purposes but can used in Commercial ones too.



- Team that have conceptualized it: Eben Upton, Rob Mullins, Jack Lang and Alan Mycroft
- Raspberry Pi Foundation was created and first alpha boards were tested, 2011
- Sales begins April, 2012
- Model B+ is released in November, 2012
- More than 5 Million already have been sold

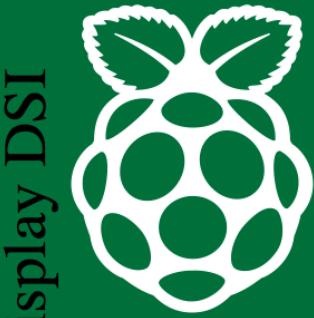
40pins: 28x GPIO, I2C, SPI, UART

Status LED's
ACT PWR

microSD slot
on bottom side

Power in

Micro
USB



Display DSI

3.3V
&
1.8V
Regulator

1

Raspberry Pi Model B+ V1.2
(C) Raspberry Pi 2014

CPU/GPU
Broadcom
BCM2835
512MB SDRAM

current limiter
polarity protection
power good

HDMI

HDMI out

Camera CSI

4 poles jack

3 . 5mm out
Composite
Video+audio

Ethernet

Ethernet
RJ45

2x USB 2.0

2x USB 2.0



Raspbian is an unofficial port of Debian Wheezy armhf with compilation settings adjusted to produce optimized "hard float" code that will run on the Raspberry Pi.

Best OS to Raspberry Pi nowadays. If you use Ubuntu, you're going to feelin in home. All basic Unix commands works on it and almost all basic server-side too:
Apache, Nginx, PHP, Java, Python, MySQL

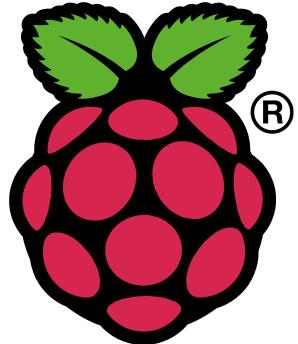
To install it in your MicroSD card (Recommend 8Gb or more)

<http://www.raspberrypi.org/documentation/installation/installing-images>

Default user: pi / Default password: raspberry

And never forget: **sudo apt-get install** and be happy :D

Note: Raspbian is not affiliated with the Raspberry Pi Foundation. Raspbian was created by a small, dedicated team of developers that are fans of the Raspberry Pi hardware, the educational goals of the Raspberry Pi Foundation and, of course, the Debian Project.



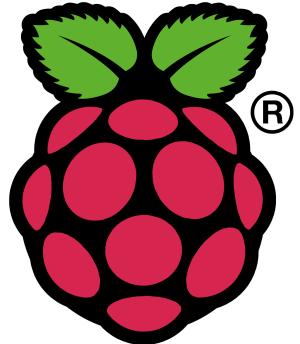
JDK SE for Embedded Devices

Since JDK 7, Java SE SDK for Desktop was ported to ARM chips and **JDK 8** is already available.

As **Raspberry Pi** is based in a ARM chip and runs over a Linux distribution you're able to use Java SE

You can run your favorite Java Application Server or create **JavaFX** applications

(you don't need to learn JME or Python anymore :D!)

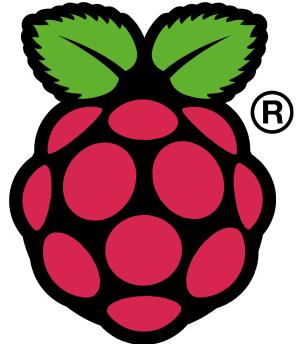


Device I/O API

Originally part of Java ME library, The **Device I/O** project is an open source, Java-level API for accessing generic device peripherals on embedded devices based on JavaSE. It's under OpenJDK project.

<https://wiki.openjdk.java.net/display/dio/Main>

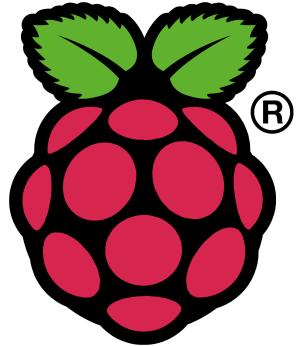
<http://docs.oracle.com/javame/8.0/api/dio/api/index.html>



Differences between Device I/O and PI4J

PI4J is an excellent API, however it's only for RaspberryPi if you would need to run your APP into a BeagleBone board will have to rewrite all GPIO access code.

Device I/O, as other Java API, brings abstraction layer over GPIO access functionallity saving your time and complexity. *Write once run anywhere* :D

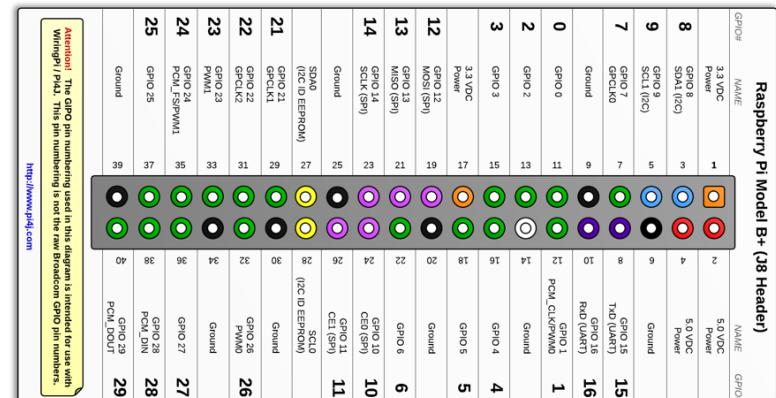


Differences between Device I/O and PI4J

Also, GPIO map is different in two APIs, check:



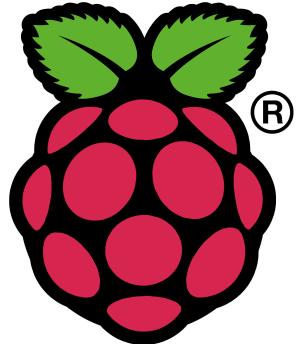
Device I/O



PI4J

<http://docs.oracle.com/javame/8.0/get-started-rpi/piportsapdx.htm>

<http://pi4j.com/pins/model-b-plus.html>



Device I/O API

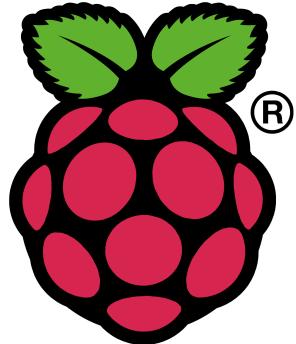
Device I/O has a configuration file for each board

The Raspberry PI file is

dio.properties-raspberrypi

It has definitions for each GPIO ports

Thus Device I/O can works with different SoC Computers



Device I/O API

Device I/O requires special JDK permissions to access GPIO

Device I/O comes with **gpio.policy** file

You can map each Pin you're planning to use or only:

```
grant {  
    permission jdk.dio.gpio.GPIOPinPermission "*:*";  
    permission jdk.dio.DeviceMgmtPermission "*:*", "open";  
};
```

(okay, don't do this in production kids)



QUESTIONS?

BATTLEHACK

Braintree_Dev.
A PayPal Company



ARE YOU THE ULTIMATE
HACKER FOR GOOD?

14 hacks. 14 cities.

\$100,000 USD at our World Finals!

Melbourne	Berlin	Raleigh
Los Angeles	London	Toronto
Singapore	Tel Aviv	Athens
Tokyo	Chicago	Stockholm
Venice	NYC	

Braintree_Dev.

A PayPal Company



BATTLEHACK

Braintree_Dev.
A PayPal Company



Hack to win an **epic axe trophy**
& a chance at **\$100,000 USD**

Massages, beer & gourmet
food awaits!

BattleHack Raleigh

April 18th - 19th
214 Martin Street

www.battlehack.org

Braintree_Dev.
A **PayPal** Company





Thanks.

Jeff Prestes

@jeffprestes
Slideshare.com/jeffprestes
Github.com/jeffprestes

@paypaldev
developer.paypal.com
developers.braintreepayments.com

YES!!!!

LAB TIME



Configure your wifi credentials:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

(file content below)

```
network={  
    ssid="iPhone"  
    psk="43070720"  
}
```

```
network={  
    ssid="<<your-wifi-name-here>>"  
    psk="<<your-wifi-password>>"  
}
```

```
network={  
    ssid="BoxNet"  
    psk="hackkitty"  
}
```



Configure your wifi network using **static IP**:

```
sudo nano /etc/network/interfaces
```

(file content below)

```
auto lo
```

```
iface lo inet loopback
iface eth0 inet dhcp
```

```
auto wlan0
allow-hotplug wlan0
iface wlan0 inet static
address 192.168.20.218
netmask 255.255.255.0
gateway 192.168.20.1
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

```
iface default inet static
```



Configure your wifi network using **DHCP** (most used):

```
sudo nano /etc/network/interfaces
```

(file content below)

```
auto lo
```

```
iface lo inet loopback
iface eth0 inet dhcp
```

```
auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

```
iface default inet dhcp
```



```
sudo apt-get update  
sudo apt-get dist-upgrade  
//Upgrades Raspbian
```

```
sudo reboot  
//Use Reboot to restart Raspbian safely
```

```
sudo halt  
//Use halt to turn off Raspbian safely
```

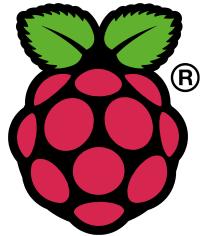


Configure the DNS Servers

`sudo nano /etc/resolv.conf`
(file content below)

`nameserver 8.8.8.8`
`nameserver 8.8.4.4`

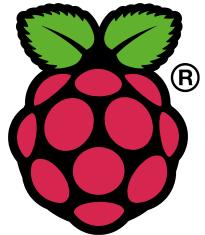
`//Google DNS servers`



Wiring Pi

It is GPIO Interface library for the Raspberry Pi. It's written in C for the BCM2835 used in the Raspberry Pi.

WiringPi includes a command-line utility **gpio** which can be used to program and setup the GPIO pins.



Wiring Pi

Installation

```
$ sudo apt-get install git-core  
$ git clone git://git.drogon.net/wiringPi  
$ cd wiringPi  
$ git pull origin  
$ ./build
```

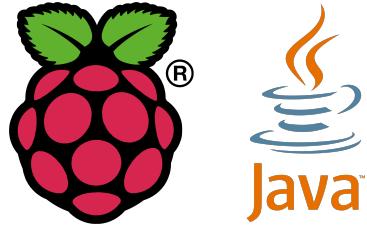
Test

```
$ gpio -v  
$ gpio readall
```



Get back to your user home directory
(in this case you're using pi user, remember?)

```
$ cd ~/
```



Downloading JDK SE 8 for Embedded

I will have to download it from your computer as you will have to accept Oracle's terms and conditions.

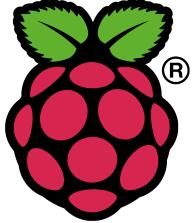
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-arm-downloads-2187472.html>

Mac and Linux users: use scp to copy the file

```
$ scp jdk-<version>-linux-arm-vfp-hflt.gz pi@<your-device-ip>:/home/pi/jdk-<version>-linux-arm-vfp-hflt.tar.gz
```

(note that I've changed the extention
because the downloaded file came only with .gz extention)

Windows users: **use WinSCP to do this**



Installing JDK SE 8 for Embedded

At PI home directory execute:

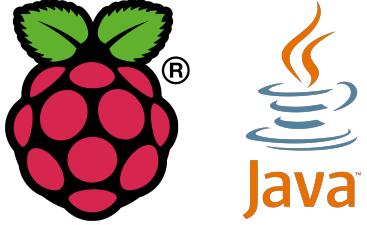
```
$ sudo mkdir -p /opt/java  
$ sudo chown root:root /opt/java  
$ cd /opt/java  
$ sudo tar xvzf ~/jdk-<version>-linux-arm-vfp-hflt.tar.gz
```

Set default java and javac to the new installed JDK.

```
$ sudo update-alternatives --install /usr/bin/javac javac /opt/jdk-<version>/bin/javac 1  
$ sudo update-alternatives --install /usr/bin/java java /opt/jdk-<version>/bin/java 1  
$ sudo update-alternatives --config javac  
$ sudo update-alternatives --config java
```

After all, verify with the commands with -verion option.

```
$ java -version  
$ javac -version
```

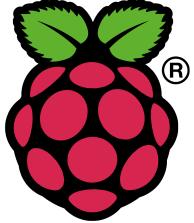


Installing JDK SE 8 for Embedded

Double check if java version is correct for root too:

```
$ sudo java -version  
$ sudo javac –version
```

This is important because will you need to run your Java application that access GPIO with root privileges.



Installing JDK SE 8 for Embedded

Define JAVA_HOME environment variable:

```
$ sudo nano /etc/environment
```

And add to it:

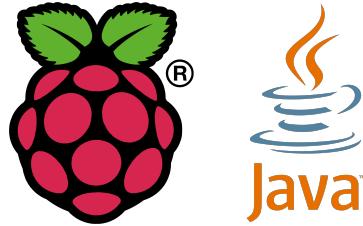
```
JAVA_HOME="/opt/jdk-<version>"
```

Do the same in your bash profile

```
$ nano ~/.bashrc
```

And add to it:

```
export JAVA_HOME="/opt/java/jdk-<version>"  
export PATH=$PATH:$JAVA_HOME/bin
```



Installing Java Device I/O

You'll need to download and compile the library in your Raspberry PI.

Device I/O like other OpenJDK projects uses Mercurial to manage their source code. You must install it into your Raspberry PI

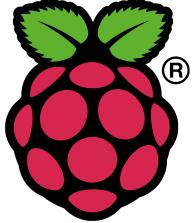
```
$ sudo apt-get install mercurial
```

Create a directory to store the source code and compile it

```
$ mkdir deviceio
```

Download it

```
$ cd deviceio  
$ hg clone http://hg.openjdk.java.net/dio/dev
```



Installing Java Device I/O

Let's compile it:

```
$ export PI_TOOLS=/usr  
$ cd dev  
$ make
```

After this, let's install your Device I/O files in your JRE

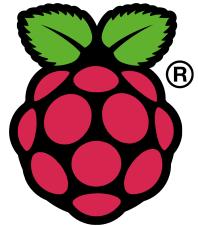
```
$ cp -r build/deviceio/lib/*  
$JAVA_HOME/jre/lib
```

Now let's get started to talk about good things...

OH GOOD

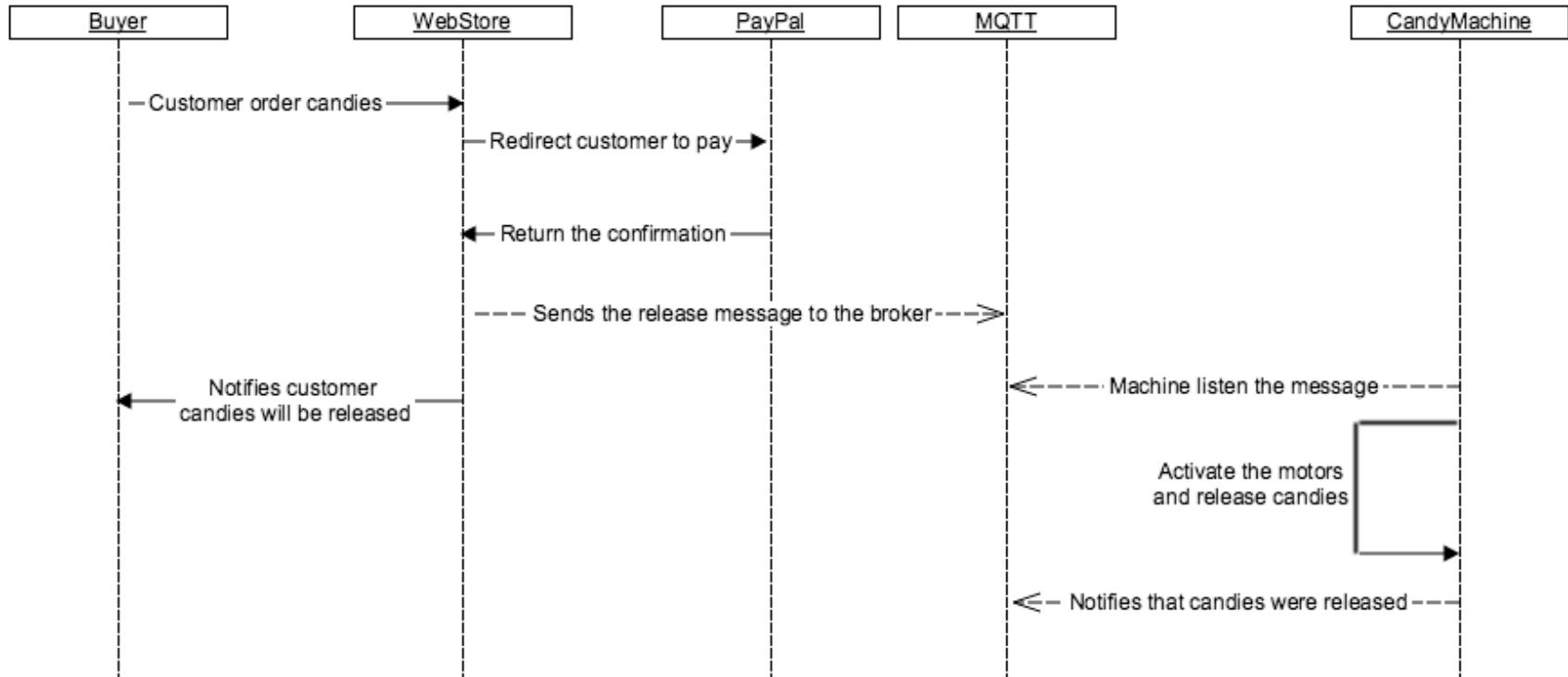
**LET'S TALK ABOUT
MONEY**

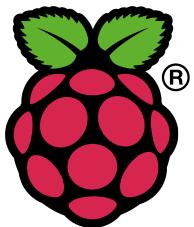
memegenerator.net



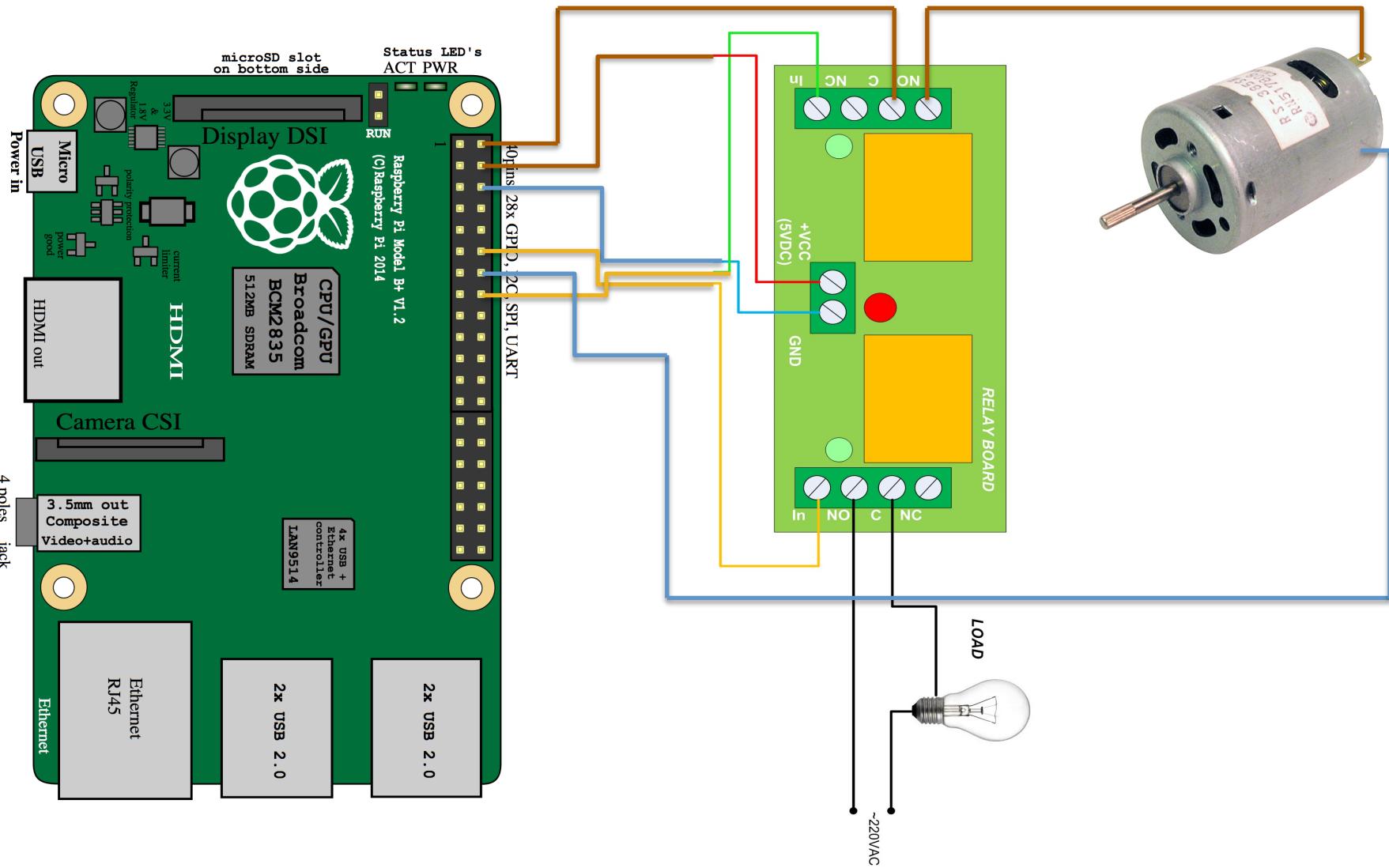
Our Lab: Create a Candy Vending Machine

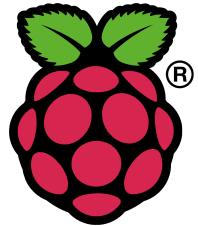
How it works





Our Lab: Create a Candy Vending Machine





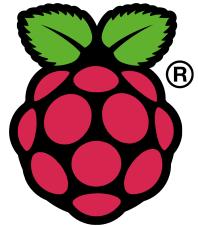
Our Lab: Create a Candy Vending Machine

Let's use Wiring Pi to test the circuits

```
$ sudo gpio mode 1 output  
$ sudo gpio mode 4 output
```

```
$ sudo gpio write 1 1  
$ sudo gpio write 1 1
```

```
$ sudo gpio write 1 0  
$ sudo gpio write 1 0
```



Our Lab: Create a Candy Vending Machine

Copy the **dio.jar** compiled in your Raspberry Pi to your machine

```
$ scp ~/deviceio/dev/build/jar/dio.jar <your-user>@<yourmachineip>:~/dio.jar
```

Or use your WinSCP

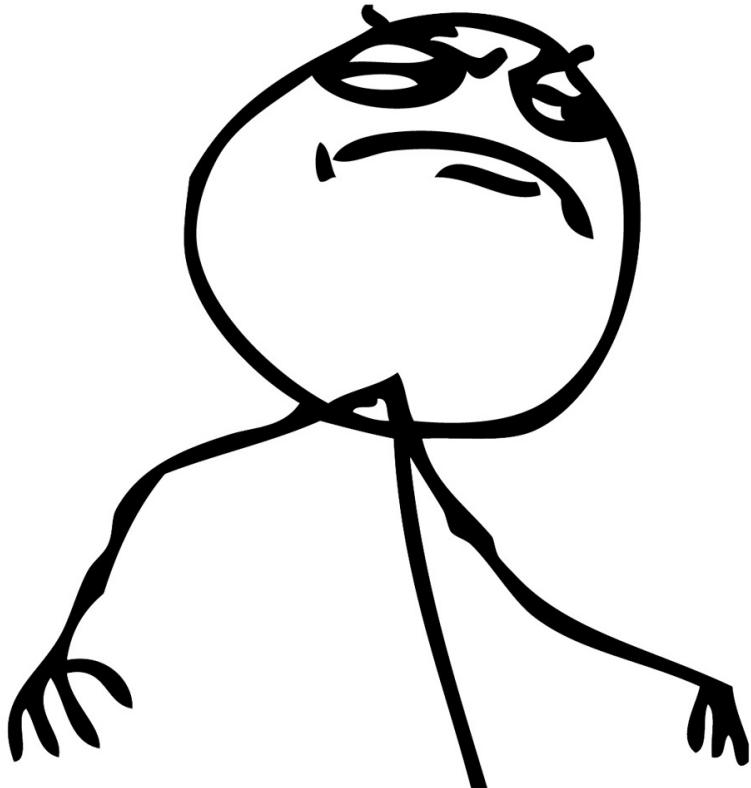
Add it to your Maven local repository

```
$ mvn install:install-file -Dfile=~/dio.jar -DgroupId=jdk.dio -DartifactId=device-io -Dversion=1.0 -Dpackaging=jar -DgeneratePom=true
```

Open the **Netbeans** and at Team->Git->Clone Repository

<https://github.com/jeffprestes/candies-client-native-java.git>

YEAH...



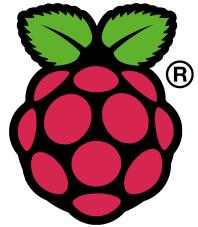
I USE NETBEANS

memegenerator.net

**SHOW ME THE
SOURCE**

YOU MUST

memegenerator.net



Our Lab: Create a Candy Vending Machine

Copy the **jar** compiled with Dependencies to home's directory of pi user.

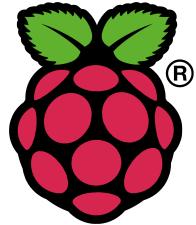
Now it's time to edit GPIO permissions in gpio.policy file

Copy it to home directory

```
$ cp ~/deviceio/dev/samples/gpio/gpio.policy ~/gpio.policy
```

And content must be like this

```
grant {  
    // permissions for using GPIO pin 18 GPIOLEDSample  
    permission jdk.dio/gpio.GPIOPinPermission ":18";  
    permission jdk.dio/gpio.GPIOPinPermission "0:18";  
    permission jdk.dio/DeviceMgmtPermission "GPIO18:18", "open";  
    permission jdk.dio/gpio.GPIOPinPermission ":24";  
    permission jdk.dio/gpio.GPIOPinPermission "0:24";  
    permission jdk.dio/DeviceMgmtPermission "GPIO24:24", "open";  
};
```



Our Lab: Create a Candy Vending Machine

Now, let's start the machine application

```
$ sudo java  
-Djdk.dio.registry=/home/pi/java/dio/config/dio.properties-raspberrypi  
-Djava.security.policy=/home/pi/java/nativegpio/gpio.policy  
-jar candies.jar iot.eclipse.org jeffprestes/candies/world machineusa 1883
```

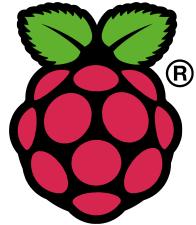
Where:

iot.eclipse.org is the MQTT server

jeffprestes/candies/world is the queue

machineusa is the client name

1883 is the MQTT broker port



Our Lab: Create a Candy Vending Machine

To buy

Create a Premier account in PayPal Sandbox

developer.paypal.com -> Dashboard -> Sandbox -> Accounts -> Create Account

Later, goto **candies.novatrix.com.br** in your mobile phone and use this user as a buyer

YES!!



IT WORKS!

memegenerator.net



Hack to win an **epic axe trophy**
& a chance at **\$100,000 USD**

Massages, beer & gourmet
food awaits!

BattleHack Raleigh

April 18th - 19th
214 Martin Street

www.battlehack.org

Braintree_Dev.
A **PayPal** Company





Thanks.

Jeff Prestes

@jeffprestes
Slideshare.com/jeffprestes
Github.com/jeffprestes

@paypaldev
developer.paypal.com
developers.braintreepayments.com