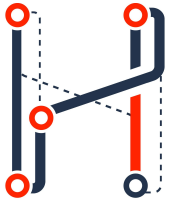


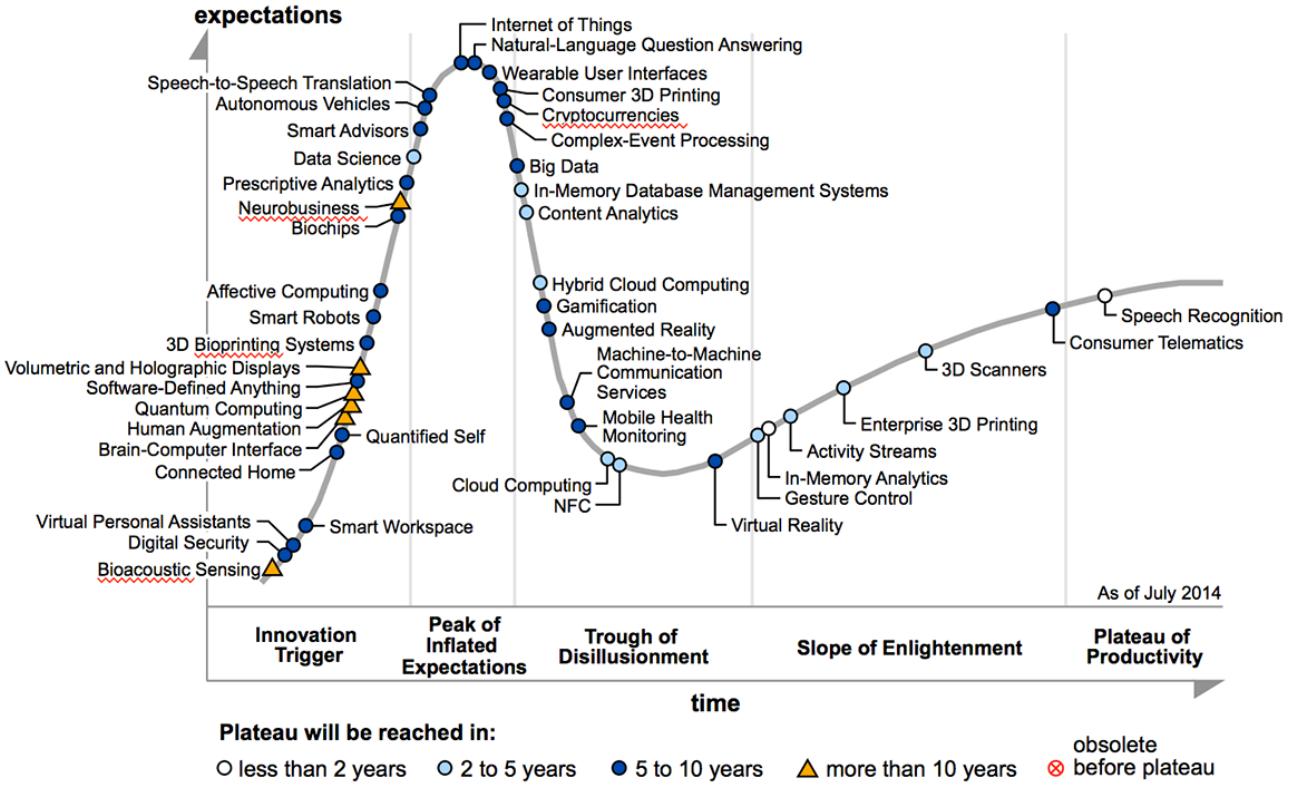
Hlutur

RallyScript

Lars Søraas @larzz
Leif Terje Fonnes @leffen


W&G

IoT - Internet of Things



Workshop

Interactive training

Different training activities

Learning goals

Practical javascript (server and frontend)

Interface MQTT

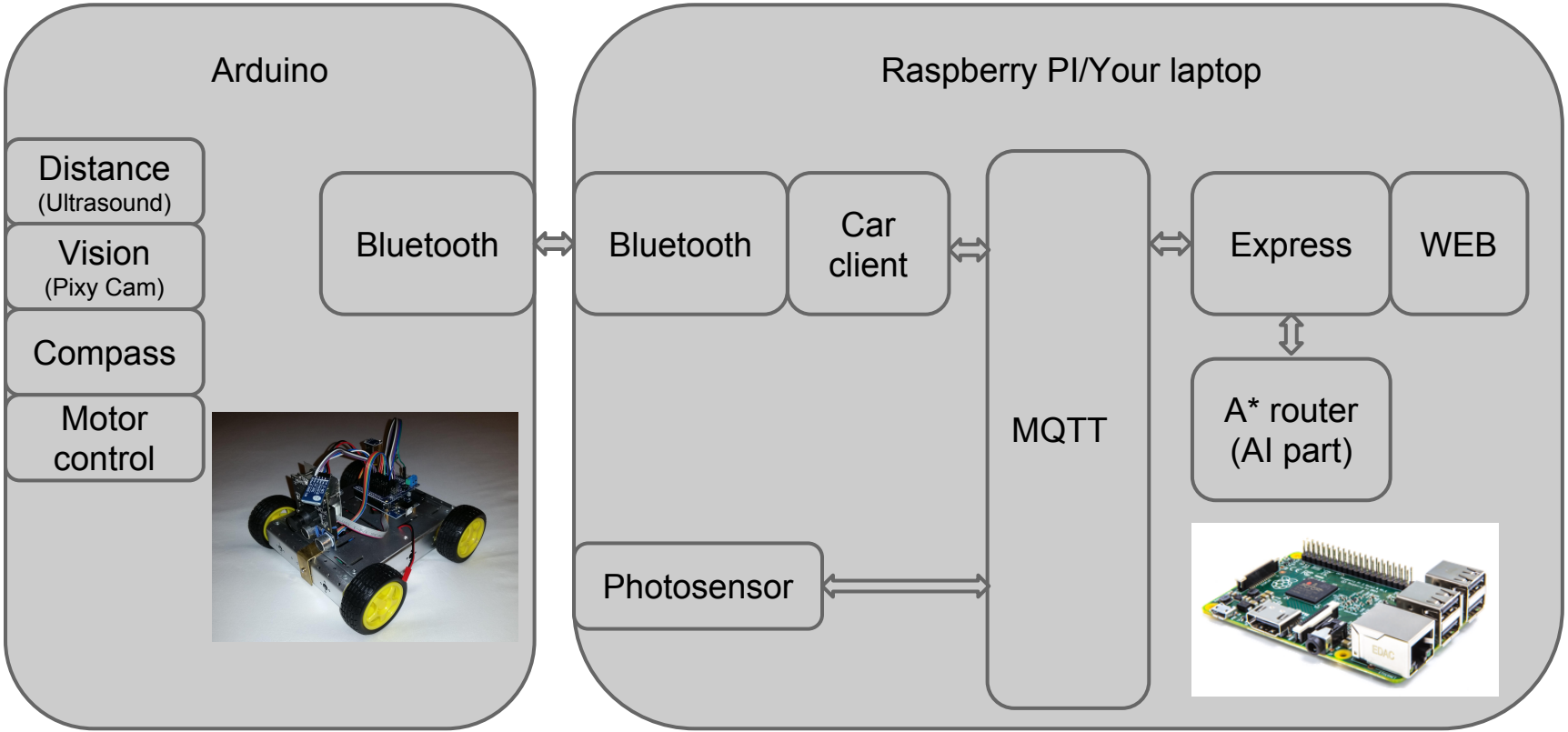
Communicating with IoT devices

Bluetooth communications

Embedded devices (Raspberry pi and arduino)

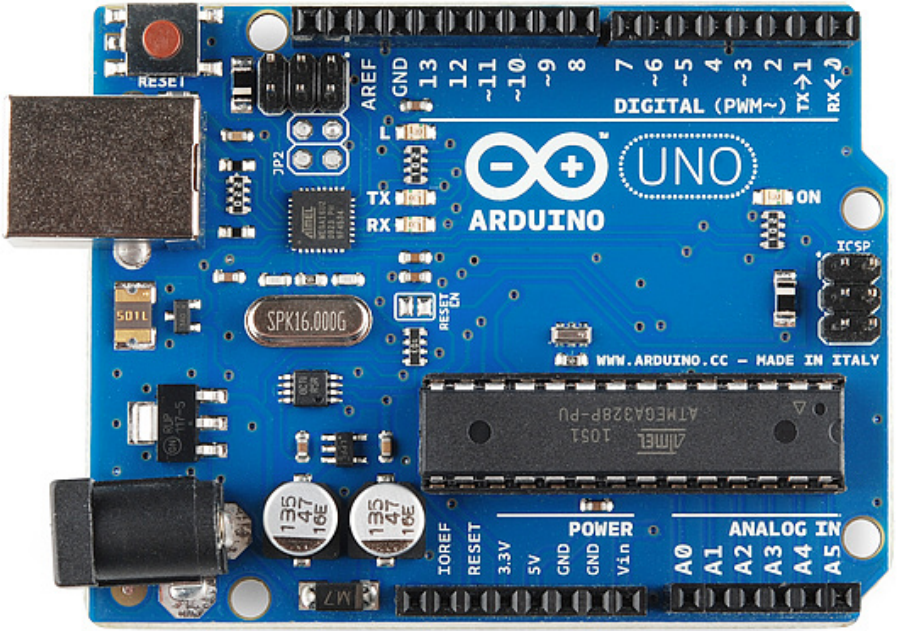
Near real time using evented systems

Architecture

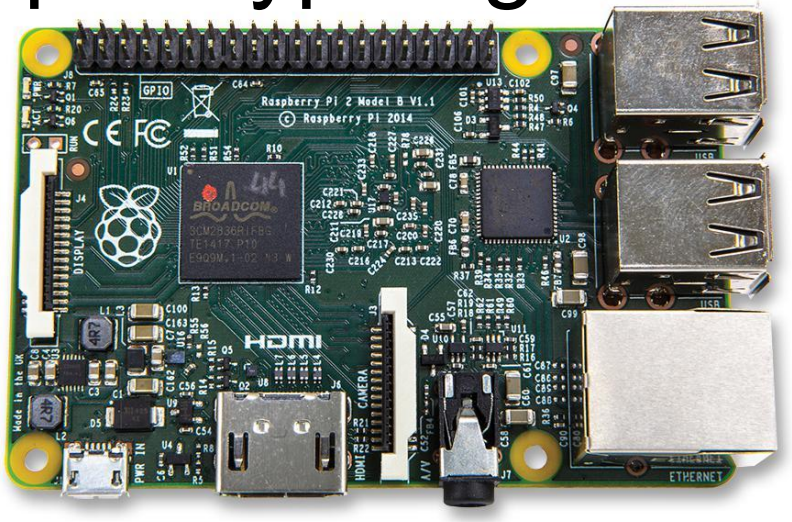


Arduino - www.arduino.cc

- Open source platform
- Atmel AVR processor
- No OS
- Lots of libraries
- Digital and analog IO
- PWM output
- Lowlevel protocols
- Simple IDE
- C-like language
- Perfect for integration of sensors and actuators



RaspberryPi - www.raspberrypi.org

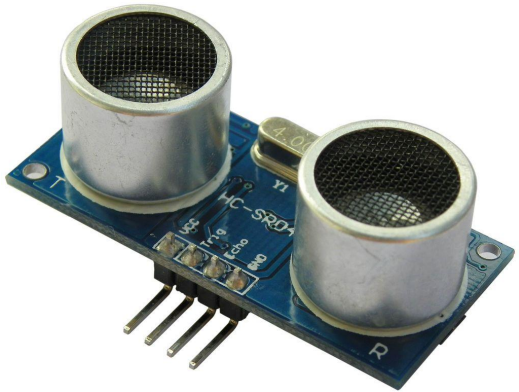


- ARM SoC (System on Chip)
- Linux
- Comparable to Pentium II
- Memory 256 MB to 1GB
- SD Card disc
- Lots of connectors:
  - USB, HDMI, AudioOut, SD, Ethernet, GPIO, Video in
- USB powered
- Created by Raspberry PI Foundation

Sensors

Distance

- Ultrasound
  - Measures echo
- IO pin triggering



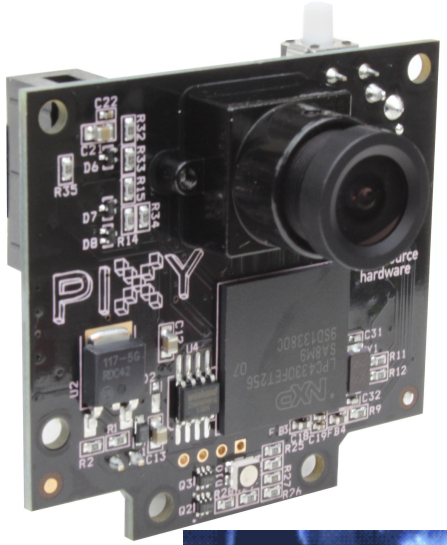
Compass/accelerometer

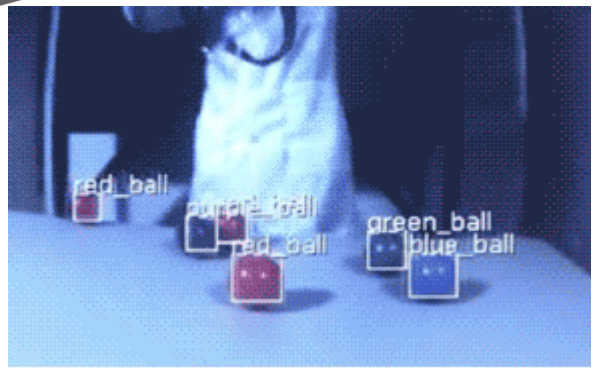
- Magneto-resistive sensors
- SPI bus

Robot vision - Pixy

- CharmedLabs
- Opensource
- Advanced image recognition
- Outputs metadata
  - What, where, size

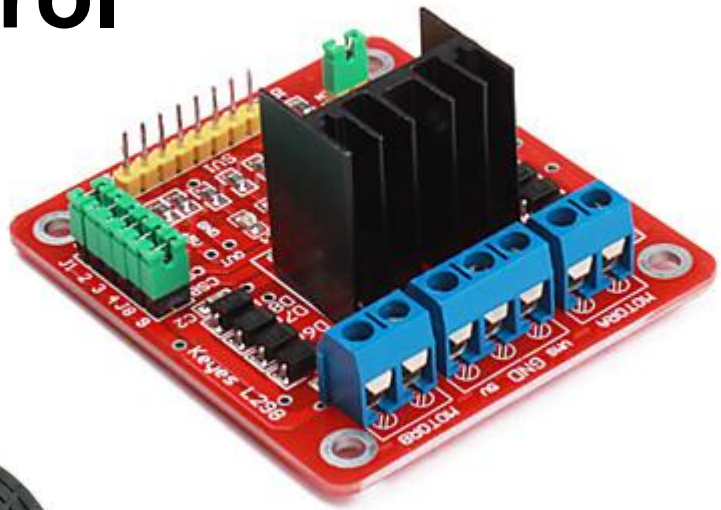
http://charmedlabs.com/default/pixy-cmucam5/

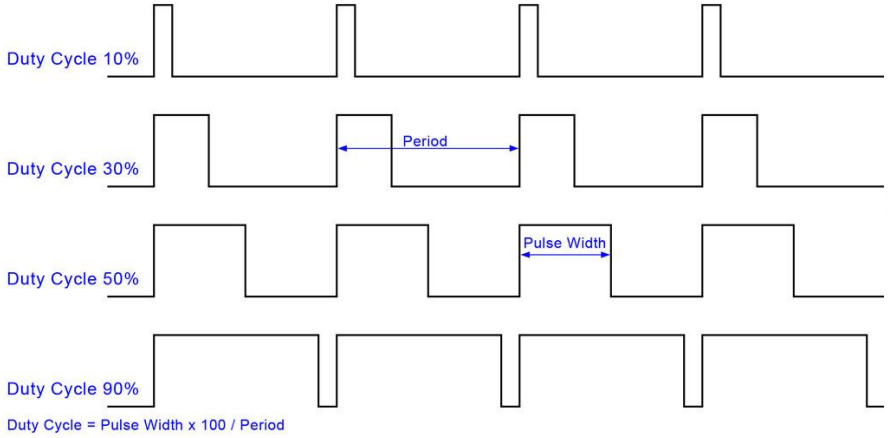


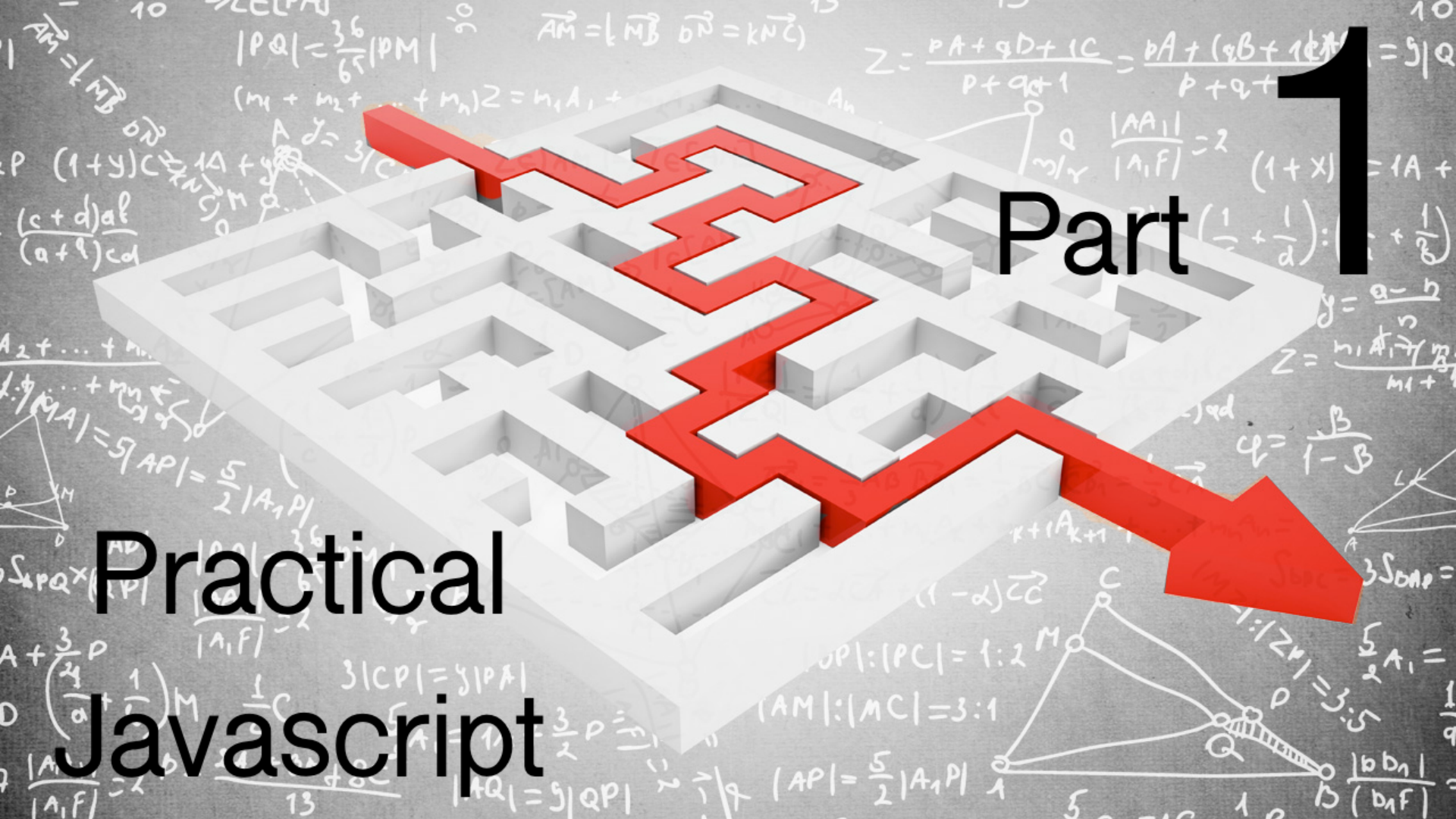


Actuators - Motor Control

- High-current controlled by low-current Arduino
- Simple DC motor controller
- PWM speedcontrol






Part 1
Practical
Javascript

Nodejs.org

Google’s V8 VM with JIT compilation

Async I/O

Lots and lots of libs

Supports quite a few frameworks.

Alternatives: jerryscript. TODO

Node JS

Single threaded

- Non blocking
- Background worker threads for I/O processing

Uses callbacks to notify completion

Install Nodejs

Check nodejs.org

I recommend to have git installed also

Lets make our first app

mkdir app1

cd app1

npm init

npm install express --save

app1.js

```
var express=require('express');
var app=express();

app.get('/',function(req,res){
  res.send('Hello RallyScript Workshop');
});

var server=app.listen(3000,function() {
  var host = server.address().address;
  var port = server.address().port;
  console.log("Ready at http://%s:%s",host,port);
});
```

Test

Open a browser and navigate to http://localhost:3000

Git

git init .

git add .

git commit -m “Initial version”

NOTE 1
To avoid to big use **.gitignore** files with the following content
node_modules

NOTE 2
git config --global user.name “Your name”
git config --global user.email “example@email.com”

Serving static files

Lets add before “var server ….”

app.use(express.static(‘public’));

Test by adding a image to public

mkdir public

cd public/

wget https://www.raspberrypi.org/wp-content/uploads/2014/07/rsz_b-.jpg

...

mv rsz_b-.jpg img1.jpg

http://ip_adress:3000/img1

index.html

<img src=”img1.jpg”>

Task

Create welcomepage for Rallyscript.

Advanced task:
Add CSS animate of Rallycar...

Adding a generator

```
$ npm install express-generator -g
```

Application 2

$ express app2 -e

create : app2

…...

create : app2/bin

create : app2/bin/www

$ cd app2 && npm install

$ DEBUG=app2 npm start

Framework setup

```
.
├── app.js
├── bin
|   └── www
├── package.json
├── public
|   ├── images
|   ├── javascripts
|   └── stylesheets
|       └── style.css
├── routes
|   ├── index.js
|   └── users.js
└── views
    ├── error.jade
    ├── index.jade
    └── layout.jade

7 directories, 9 files
```

Test it out

How can we add data to the template ?

Where do we modify the template ?

Task

- add page
- add navigation

nodemon

http://nodemon.io/

Automatic reload of server

```
$ npm install -g nodemon
```

Update package.json

```
"start": "nodemon ./bin/www"
```

Try to make a few changes

Try to make changes to index.js and refresh in browser.

browsersync / LiveReload

http://www.browsersync.io/

Tool for automatically reload browser changes

$ npm install browser-sync --save-dev

Modify bin/www

add in top section:

```
var browserSync = require('browser-sync');
```

and inside the onListening function:

```
 browserSync({
    proxy: 'localhost:' + addr.port,
    files: ['public/**/*.{js,css}']
  });
```

Try it out

Do some stylesheet changes

Add some text to the templates

IoT scenarios

Real time event data

One to many distribution model

Small devices

MQTT

A practical protocol for the internet of things

Small and lightweight

Have Quality of Service implemented

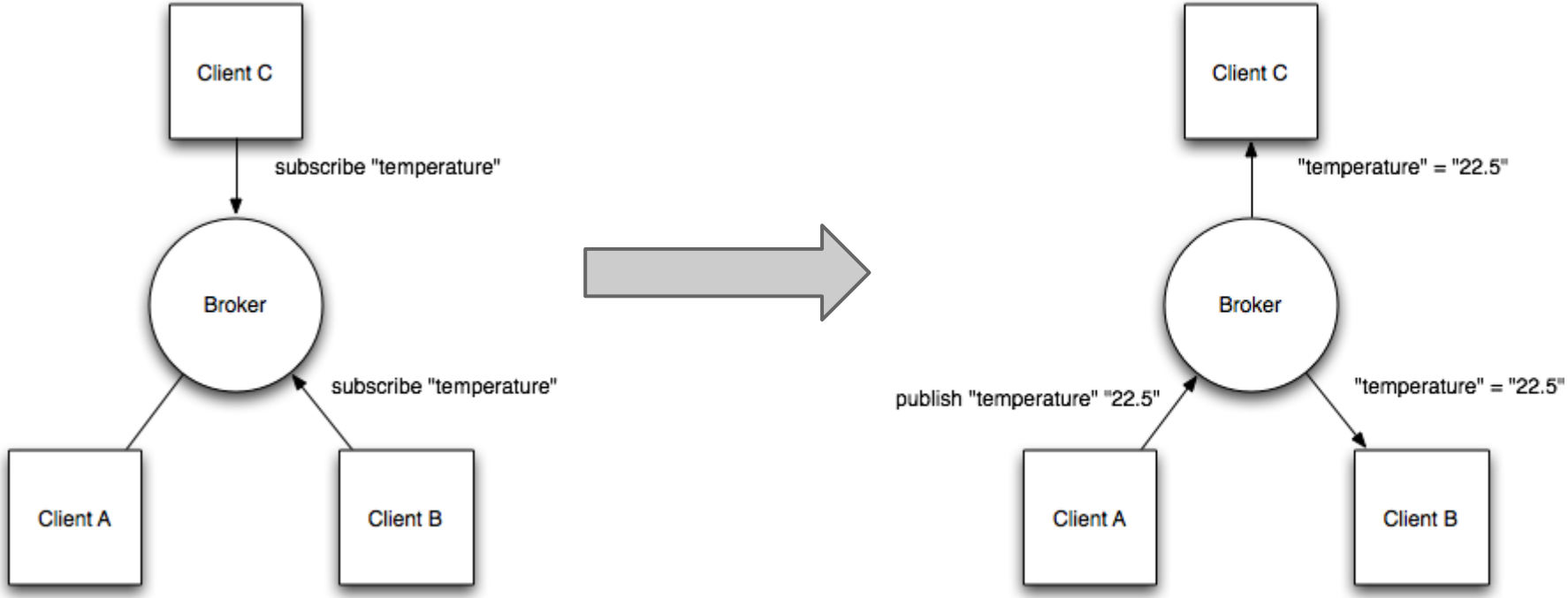
Easy model: connect, subscribe and publish

Open spec

Small clients

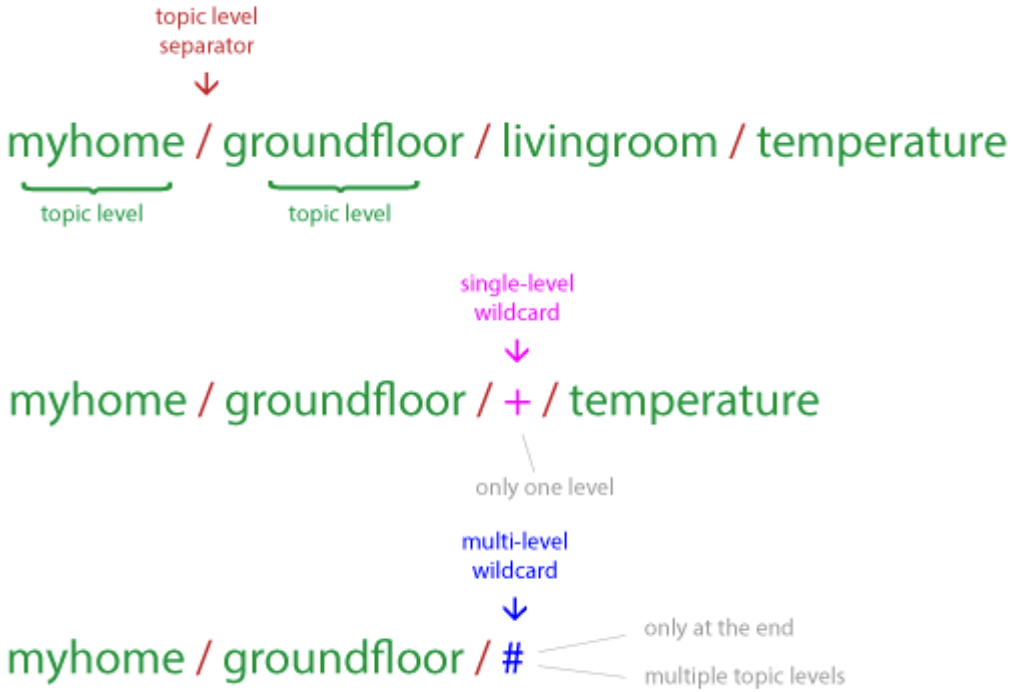
Alternative: CoAP (Compact binary REST)

MQTT subscription



http://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php

The MQTT topics



http://www.hivemq.com/

MQTT packet

Publish	
packetId	Ex: 432, always 0 for qos 0
topicName	home/temperature/kitchen
qos	1 (<= 1), 2 (>= 1), or 3 (== 1)
retainFlag	true|false, last good message saved by broker
payload	Ex: “{temperature:22}”
dupFlag	true|false, true if message is resent duplicate (qos > 0 only)

Install

osx: brew install mosquitto

debian:

```
sudo apt-get update
sudo apt-get install mosquitto mosquitto-clients
```

NPM package

Global:

$npm install mqtt -g

Makes the mqtt command available on the command line

In application:

$npm install mqtt --save

First application

```
var mqtt    = require('mqtt');
var client  = mqtt.connect('mqtt:localhost');
var topic = '/myhome/kitchen/chat';

client.on('connect', function () {
  client.subscribe(topic);
  client.publish(topic, 'Hello mqtt');
});

client.on('message', function (topic, message) {
  console.log(topic +':' + message.toString());
  client.end();
});
```

Bluetooth

Pair Bluetooth to device on the server

bluetooth-serial-port

npm install bluetooth-serial-port --save

Bluetooth access 1

```
var serial = new (require('bluetooth-serial-port')).BluetoothSerialPort();

serial.listPairedDevices(function(pairedDevices) {
 pairedDevices.forEach(function(device) {
   console.log(device);
 });
});
```

Run: …. { name: 'HC-06', address: '30-14-06-09-10-16', services: [] } ….

Bluetooth access 2

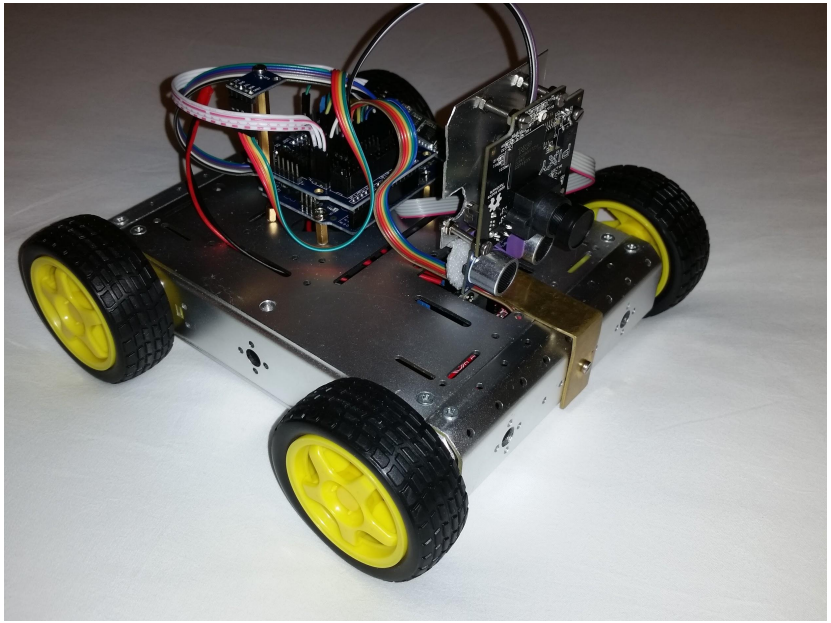
```
var btSerial = new (require('bluetooth-serial-port')).BluetoothSerialPort();

var address = '30-14-06-09-10-16',
 channel=1;

btSerial.connect(address, channel, function () {
 console.log('connected to ',address);

 btSerial.write(new Buffer('ll', 'utf-8'), function (err, bytesWritten) {if (err) console.log(err);});

 btSerial.on('data', function (buffer) {console.log(buffer.toString('utf-8')); });
}, function () {
 console.log('cannot connect');
});
```

Task

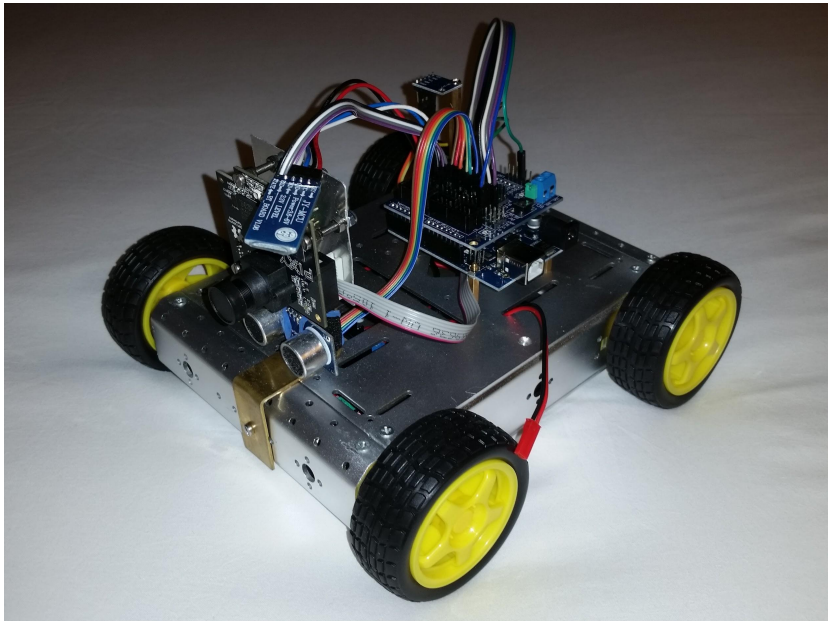
1) Create listener for car data and deliver to MQTT
2) Create listener for MQTT and deliver to car

The Racecars

Turtle



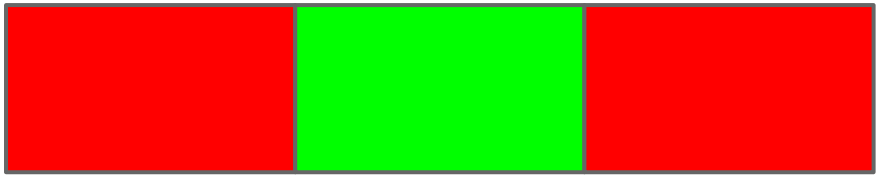
Rabbit



Pixy Patterns

Color codes

121 =



321 =

Commands

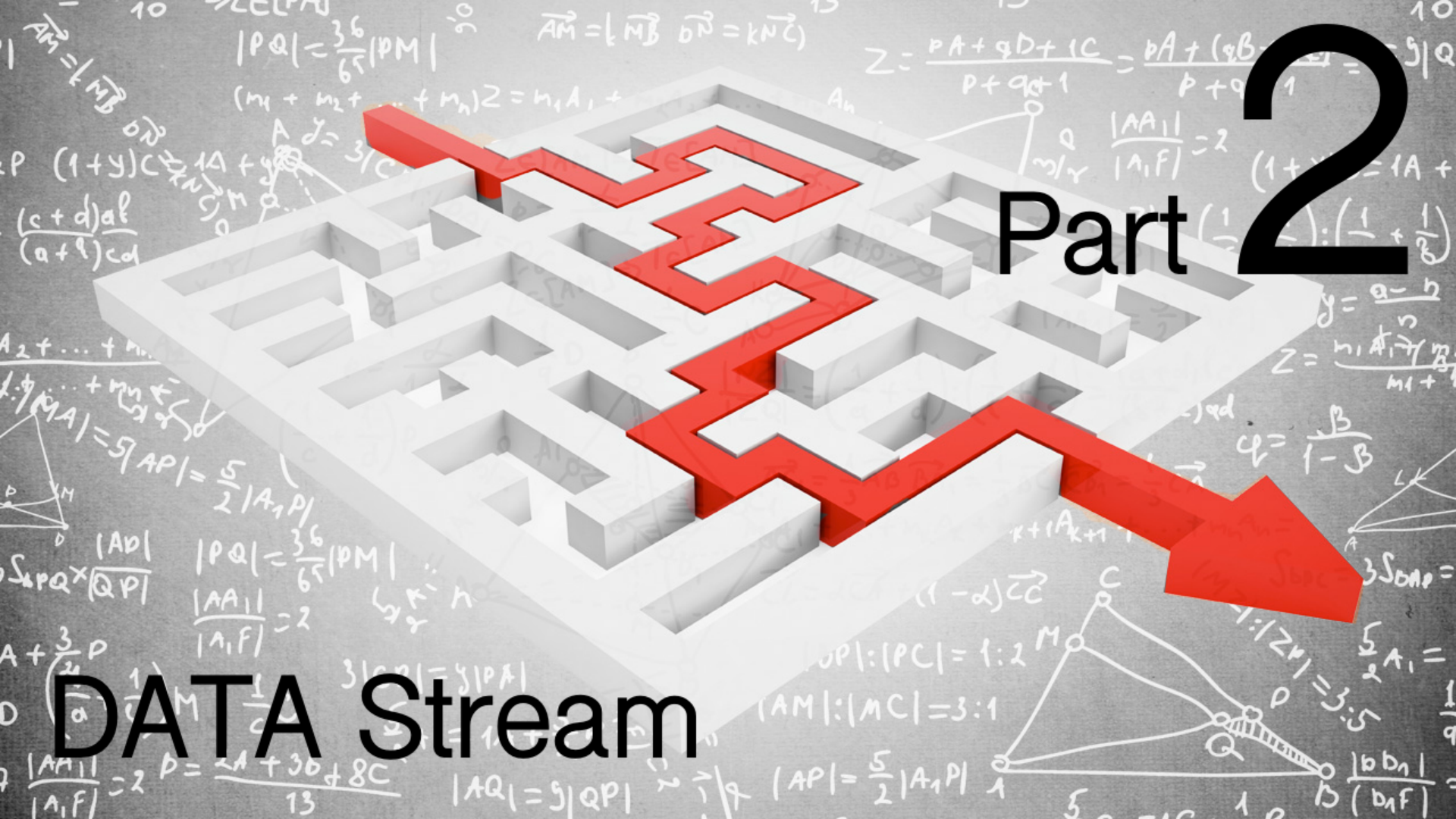
- f : fast forward
- l : Short turn left
- r : Short turn right
- s : Stop
- b : backwards
- l*deg* : *deg* turn left
- r*deg* : *deg* turn right
- v[0-255] : set top-speed

Pattern commands

- <*pat* : turn left until *pat* discovered
- >*pat* : turn right until *pat* discovered
- pat = Is pixy color pattern

MQTT friendly BLT Car datastream

fullstackfest/car/Rabbit/move|{dir:'f'}	Change in movement: forward
fullstackfest/car/Rabbit/move|{dir:'stop'}	Change in movement: stopped
fullstackfest/car/Rabbit/move|{dir:'l'}	Change in movement: turned left
fullstackfest/car/Rabbit/move|{dir:'r'}	Change in movement: turned right
fullstackfest/car/Rabbit/heading|{deg:82.75}	Heading changed, new heading reported
fullstackfest/car/Rabbit/dist|{cm:23}	Wall detected, distance reported
fullstackfest/car/Rabbit/cc|{id:121,x:100,w:120}	Recognized Color Code. Position (x) and width. 100 is apx middle.
fullstackfest/car/Rabbit/cc|{id:121,inRange:'true'}	Detected CC 121 during 'search turn'
fullstackfest/car/Rabbit/cc|{id:121,inRange:'false'}	Failed to detected CC 121 during 'search turn'
fullstackfest/car/Rabbit/heartbeat|{t:4.321}	Heartbeat, each 5 sec


Part 2
DATA Stream

Bower

A package manager for the web

```
npm install -g bower
```

.bowerrc

```
{
  "directory" : "public/components"
}
```

getbootstrap.com

Adding bootstrap(++) to the project

cd app2

bower init

bower install bootstrap --save

bower install underscore --save

Adding to the web page

edit views/index.ejs

Add bootstrap css just above the style.css

<link href=”/components/.. /bootstrap.css rel=”stylesheet”>

Add navigation buttons

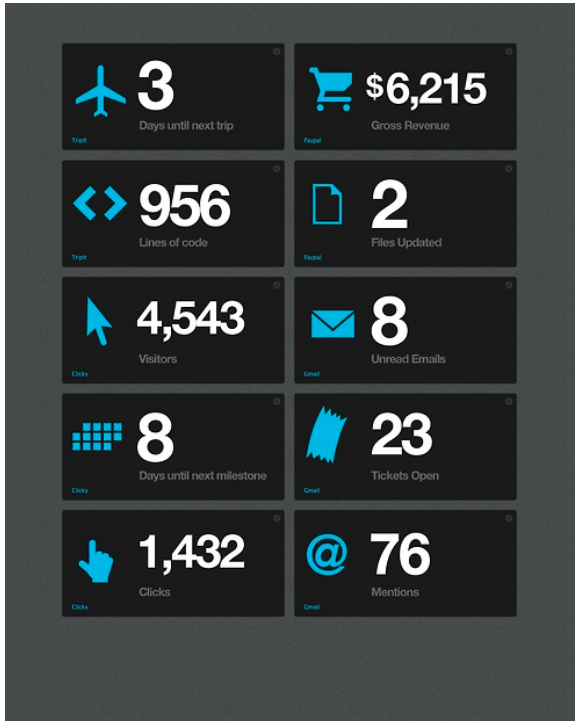
```
<div class="btn-group" role="group" aria-label="direct commands">
 <button type="button" class="btn btn-default car_action" data="f">Forward</button>
 <button type="button" class="btn btn-default car_action" data="l">Left - one step</button>
 <button type="button" class="btn btn-default car_action" data="r">Right - one step</button>
 <button type="button" class="btn btn-default car_action" data="b">B - drive in reverse</button>
 <button type="button" class="btn btn-default car_action" data="s">S - Stop</button>
</div>
```

script

```
$('.car_action').click(function(event){$(event.target).attr("data");});
```

Task

Present car state on web page



Task

Add commands to web page deliver to MQTT

Socket.io

npm install socket.io --save

Add to sourcefile:

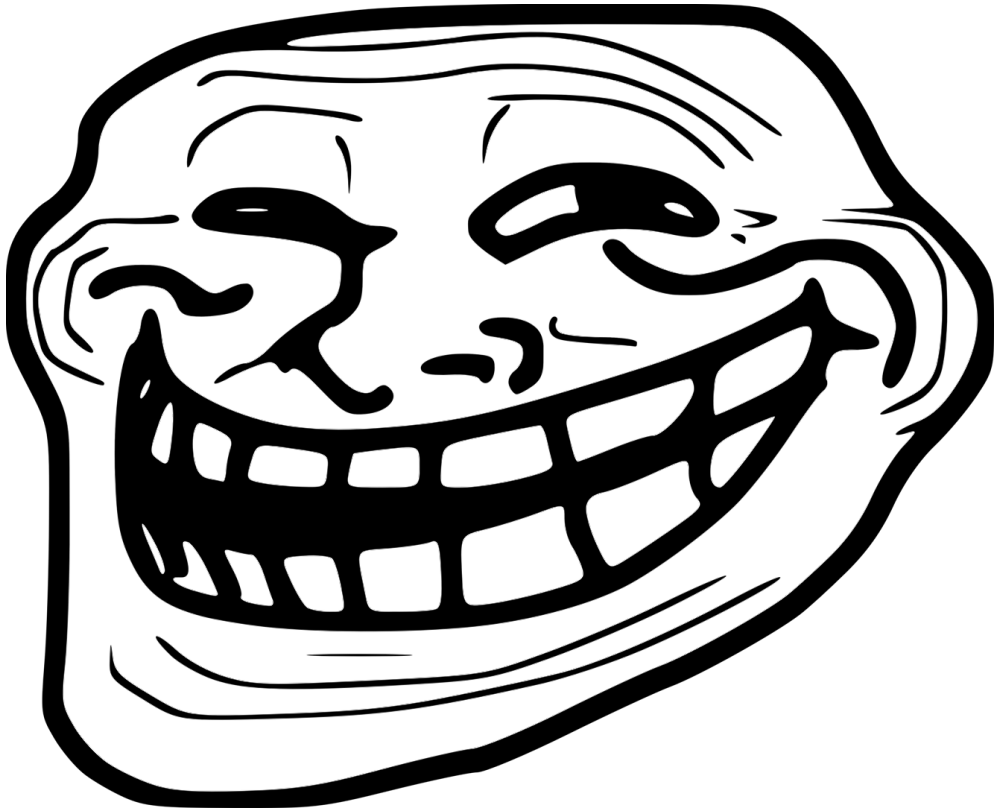
```
var io = require('socket.io')(http);

io.on('connection', function(socket){
  console.log('a user connected');
});
```

Socket.io - frontend

pixie cam

TODO

Laser fun



Part 4

Competition - TBD

Naming your PI

sudo apt-get install avahi-daemon

sudo nano /etc/hosts

** Change raspberrypi to your actual device name. F.eks my_pi

sudo nano /etc/hostname

** Change raspberrypi to my_pi (NB same name as above)

sudo /etc/init.d/hostname.sh

sudo reboot

On your workstation:
ping my_pi.local
ssh my_pi.local

Install node on the pi

- http://node-arm.herokuapp.com/

wget http://node-arm.herokuapp.com/node_latest_armhf.deb

sudo dpkg -i node_latest_armhf.deb
# Check installation
node -v

Somehow I managed to break npm and had to use :

http://node-arm.herokuapp.com/node_0.10.36_armhf.deb

Resources

http://nodejs.org/

http://expressjs.com/

http://nodemon.io/

http://www.browsersync.io/

https://www.npmjs.com/package/mqtt

http://mosquitto.org/

Other

https://www.youtube.com/watch?v=3ZWVcCHXbbY