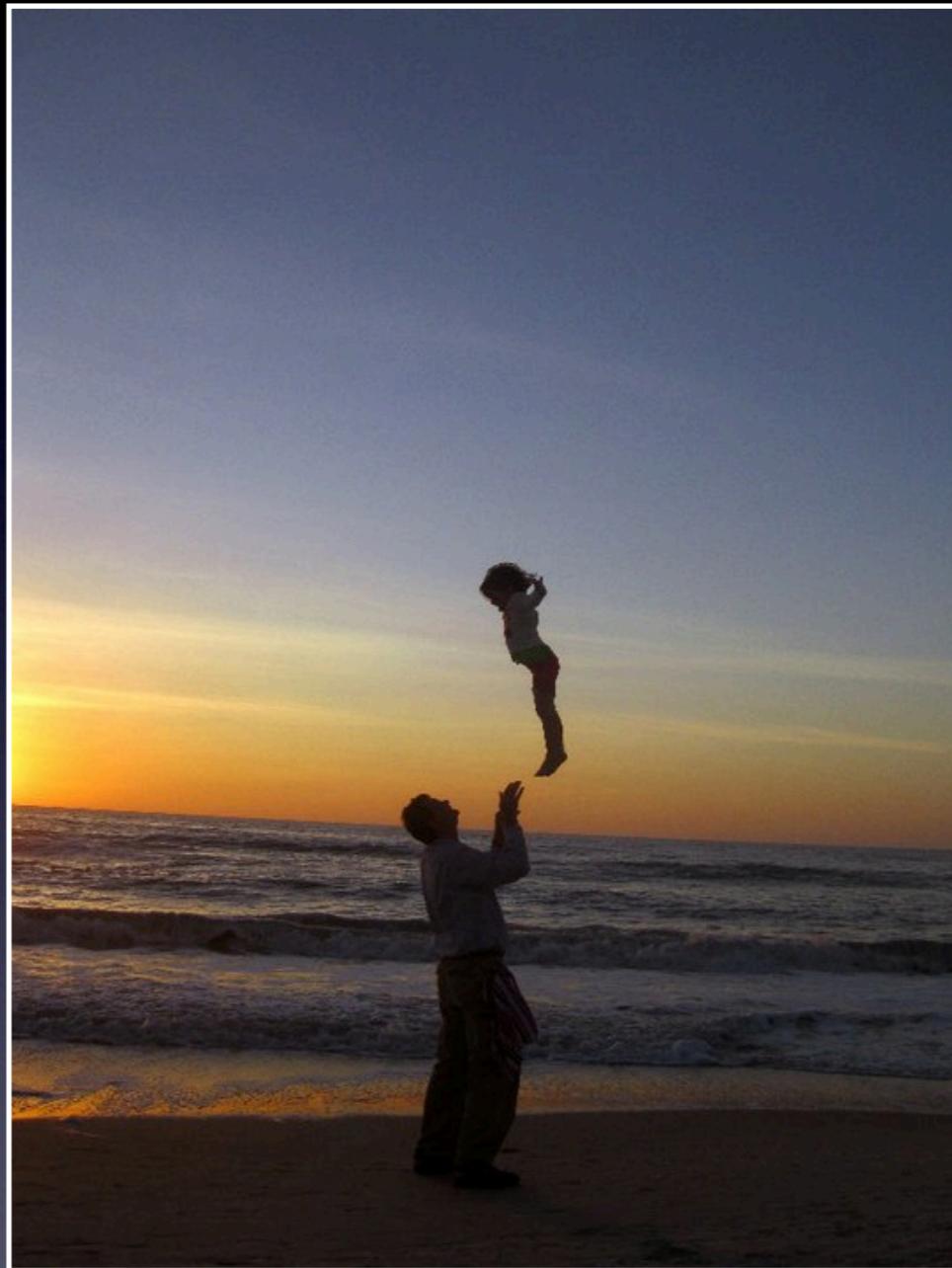


# Communicating with Perl and Arduino

Robert Blackwell  
YAPC::NA 2011

[robert@robertblackwell.com](mailto:robert@robertblackwell.com)

[twitter](#) [github](#)  
[facebook](#) [linkedin](#)



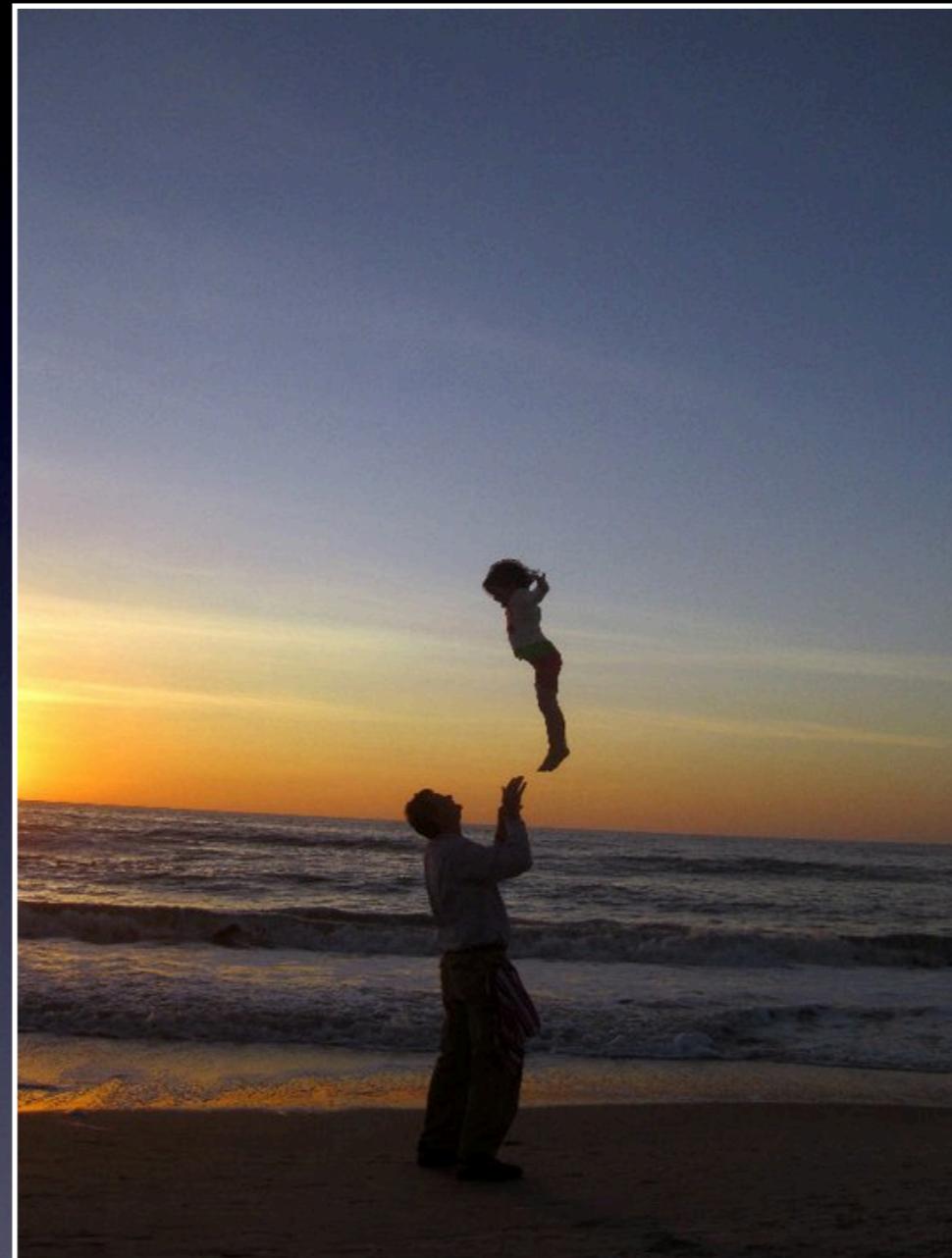
**BASIC!  
INTRO!  
GETTING STARTED!  
HELLOWORLD!**

# Communicating with Ruby and Arduino

Robert Blackwell  
YAPC::NA 2011

[robert@robertblackwell.com](mailto:robert@robertblackwell.com)

[twitter](#) [github](#)  
[facebook](#) [linkedin](#)

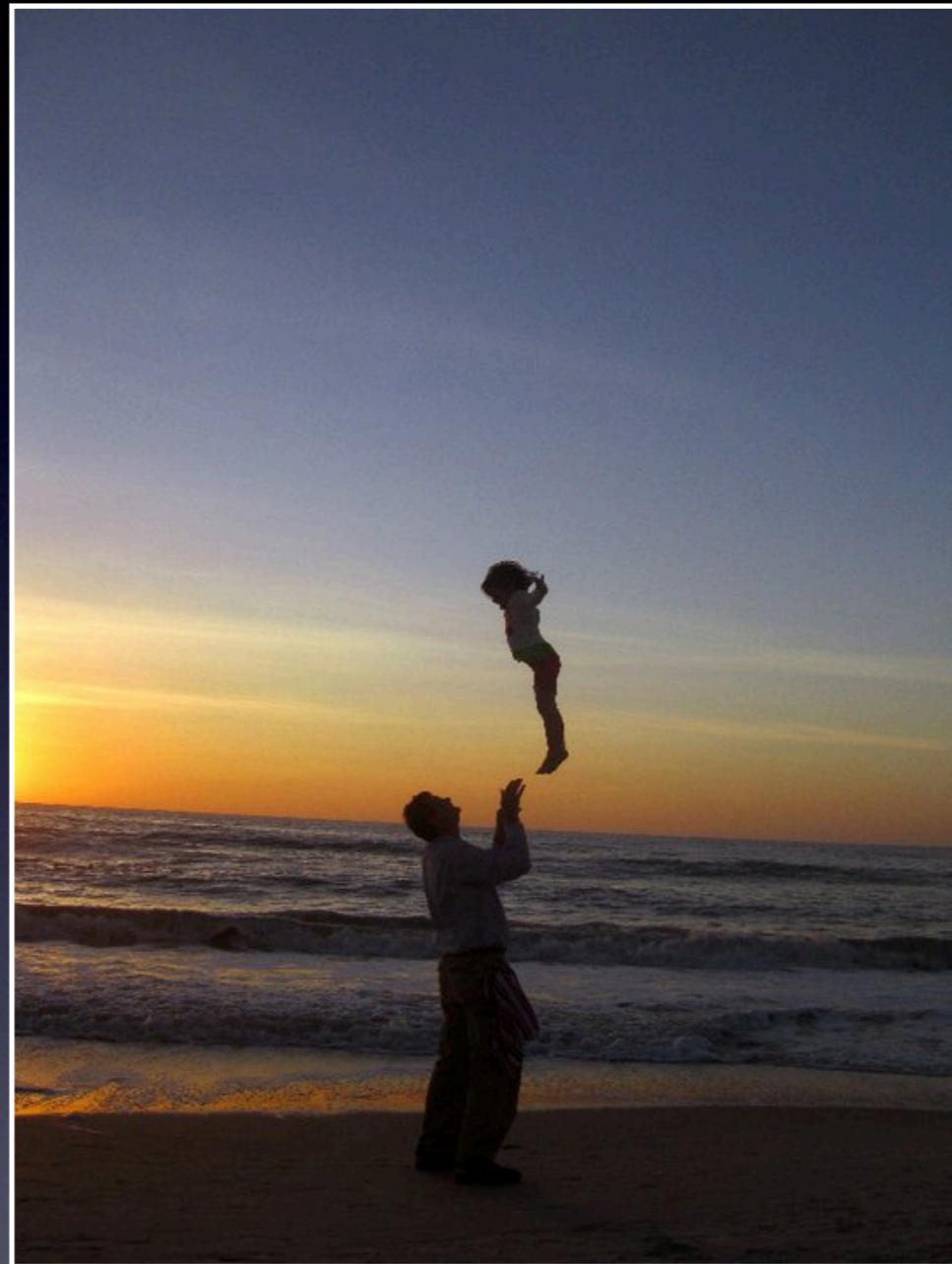


# Communicating with Python and Arduino

Robert Blackwell  
YAPC::NA 2011

[robert@robertblackwell.com](mailto:robert@robertblackwell.com)

[twitter](#) [github](#)  
[facebook](#) [linkedin](#)

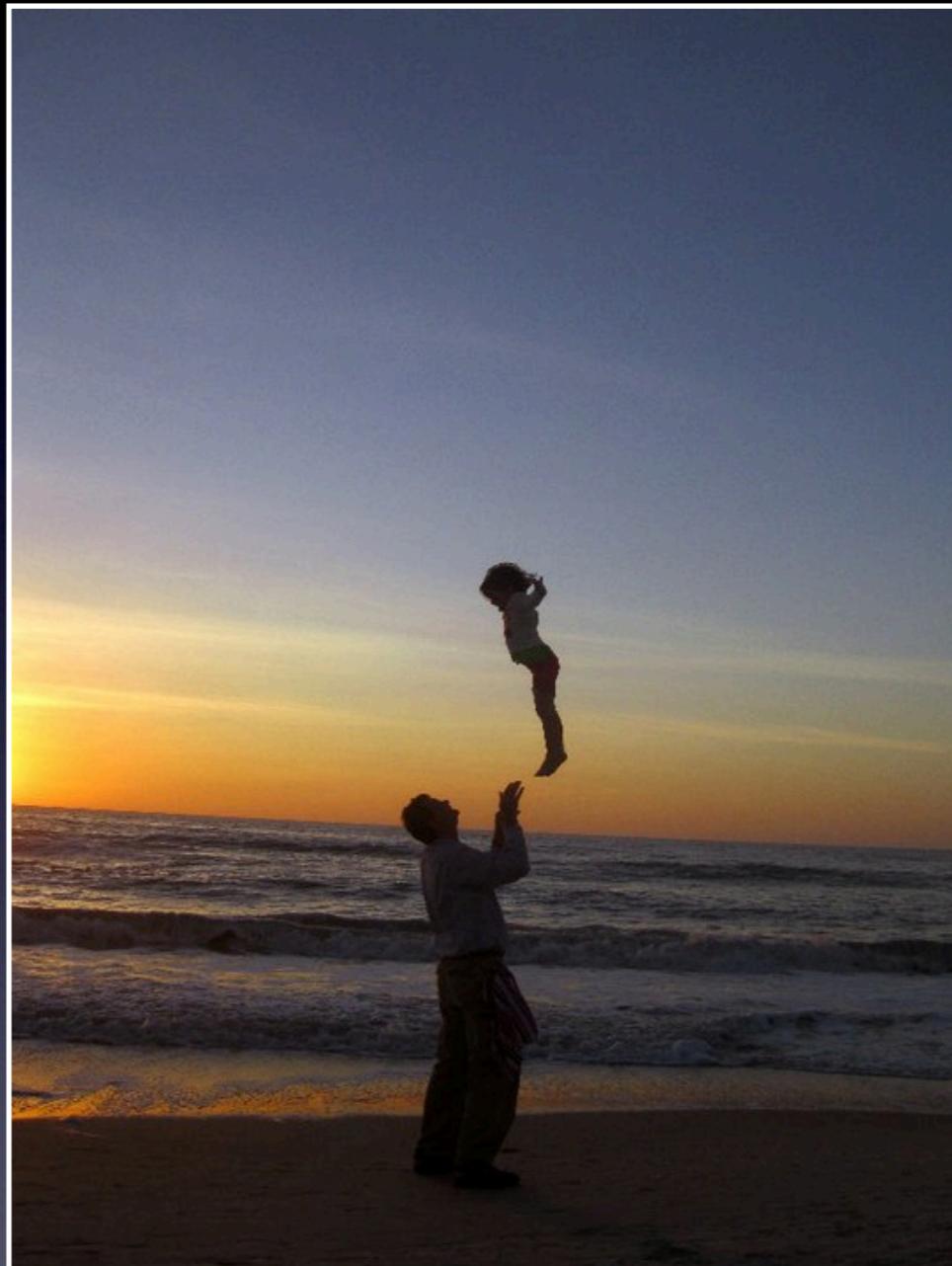


# Communicating with Erlang and Arduino

Robert Blackwell  
YAPC::NA 2011

[robert@robertblackwell.com](mailto:robert@robertblackwell.com)

[twitter](#) [github](#)  
[facebook](#) [linkedin](#)

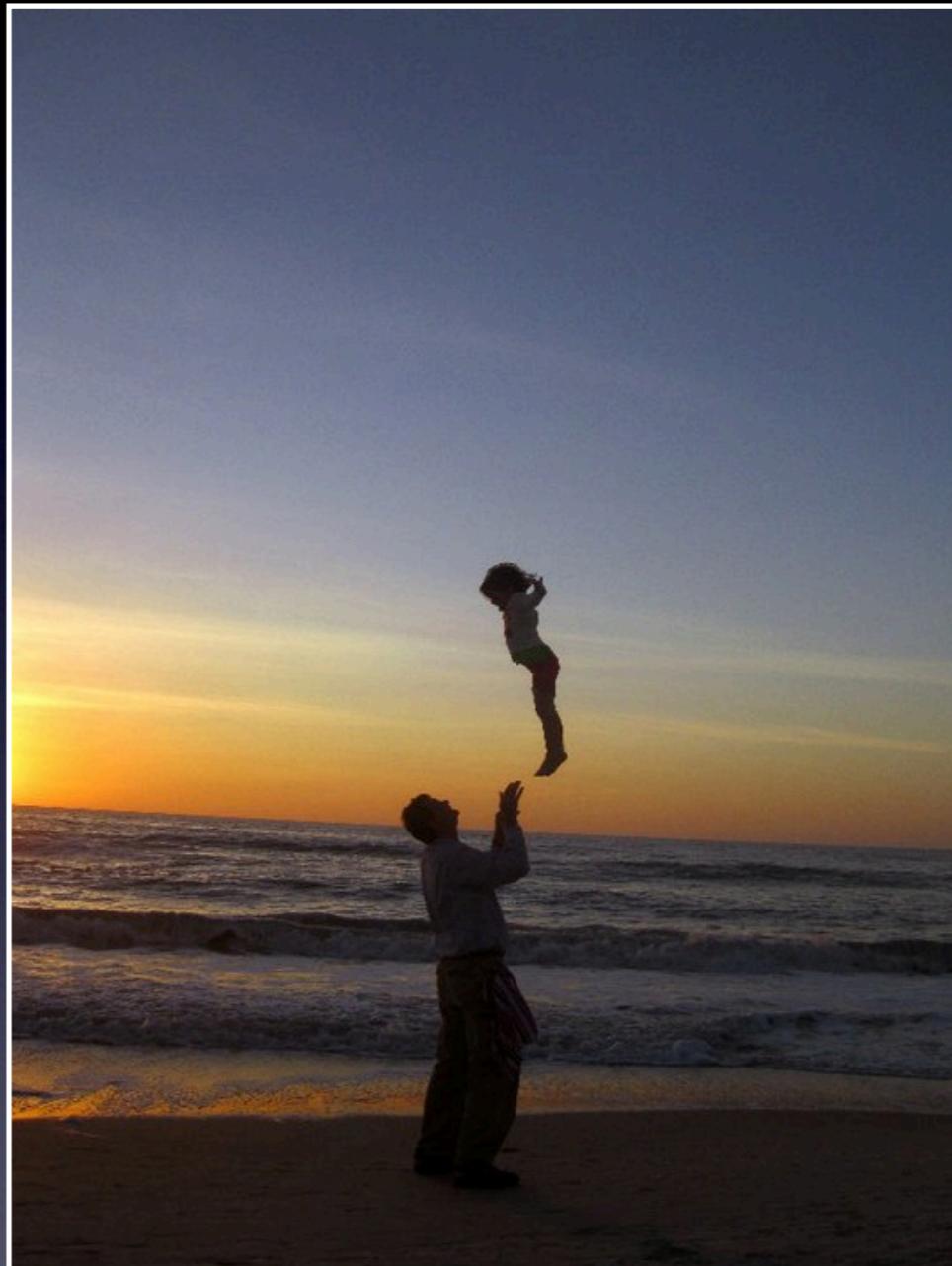


# Communicating with Bash and Arduino

Robert Blackwell  
YAPC::NA 2011

[robert@robertblackwell.com](mailto:robert@robertblackwell.com)

[twitter](#) [github](#)  
[facebook](#) [linkedin](#)

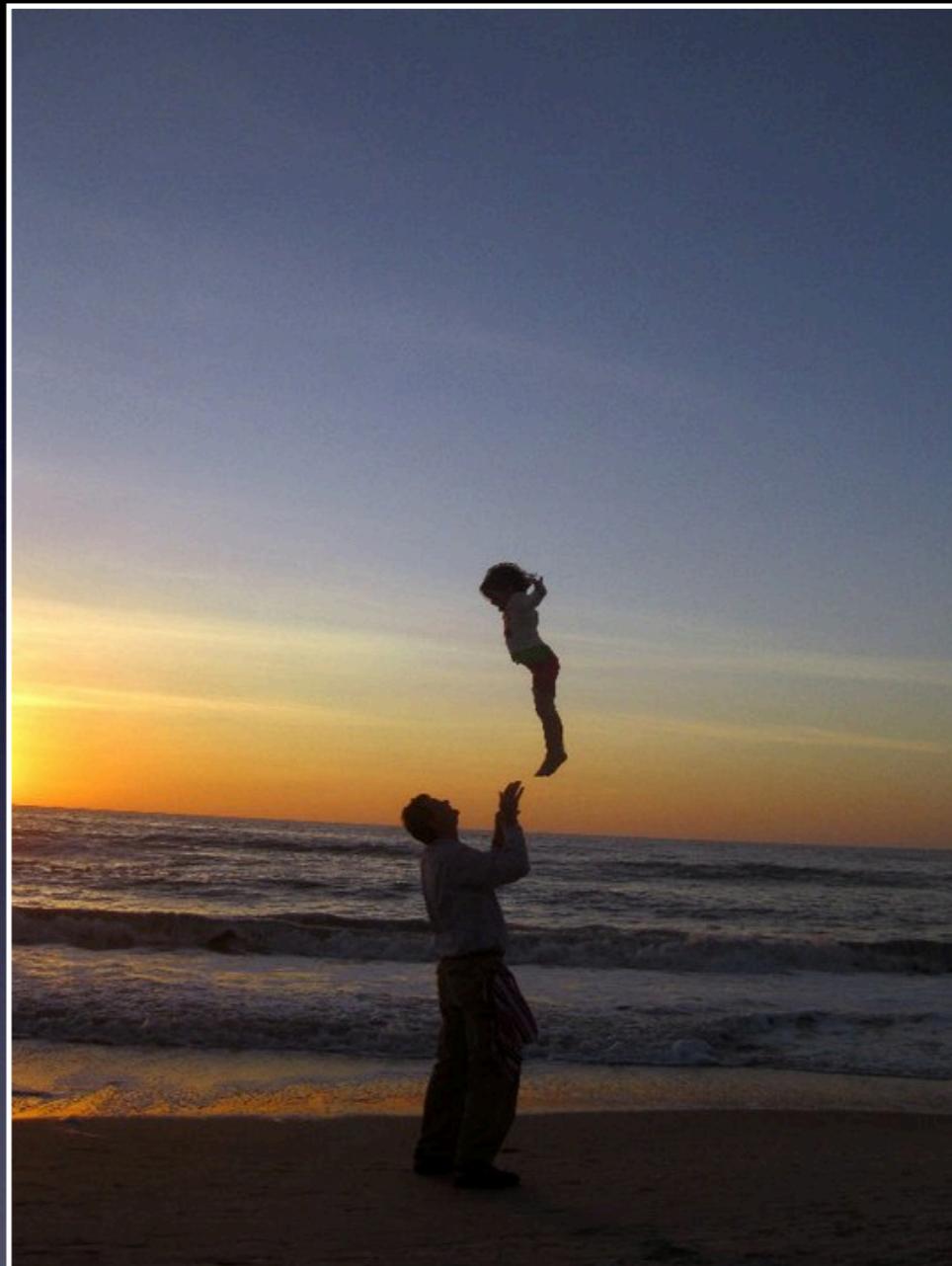


# Communicating with Haskell and Arduino

Robert Blackwell  
YAPC::NA 2011

[robert@robertblackwell.com](mailto:robert@robertblackwell.com)

[twitter](#) [github](#)  
[facebook](#) [linkedin](#)

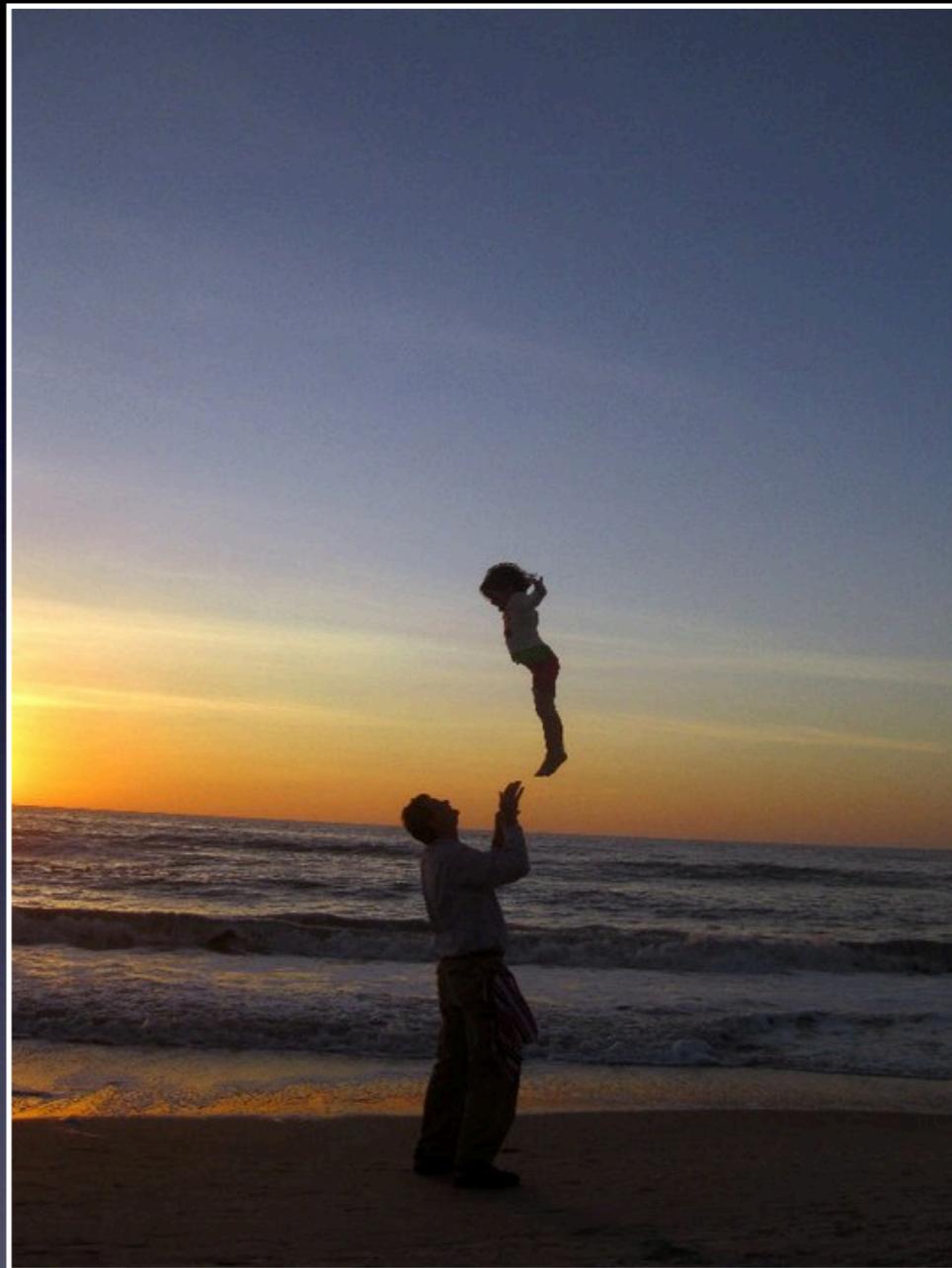


# Communicating with ???? and Arduino

Robert Blackwell  
YAPC::NA 2011

[robert@robertblackwell.com](mailto:robert@robertblackwell.com)

[twitter](#) [github](#)  
[facebook](#) [linkedin](#)



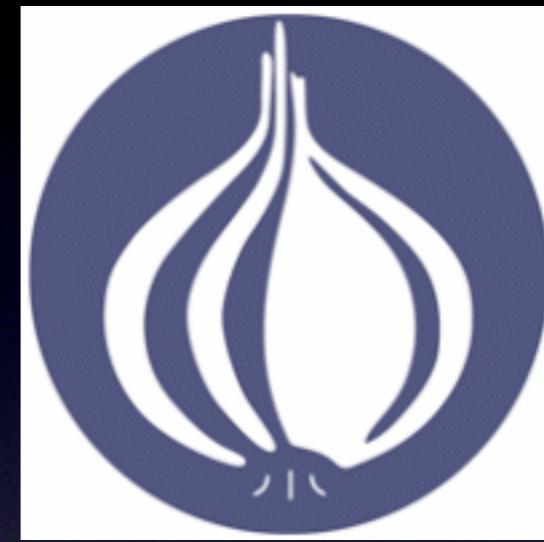


[arduino.cc](http://arduino.cc)

Can you say it?



[arduino.cc](http://arduino.cc)



[arduino.cc](http://arduino.cc)

[perl.org](http://perl.org)

# Hello World?

Hello World?  
Not just code.

Things with Stuff

# HelloWorld?

Hello World?  
Hello Arduino

Hello World?  
Hello Arduino  
Hello Stuff

Hello World?  
Hello Arduino  
Hello Stuff  
Hello Ideas

Hello World?  
Hello Arduino  
Hello Stuff  
Hello Ideas  
HELLO!

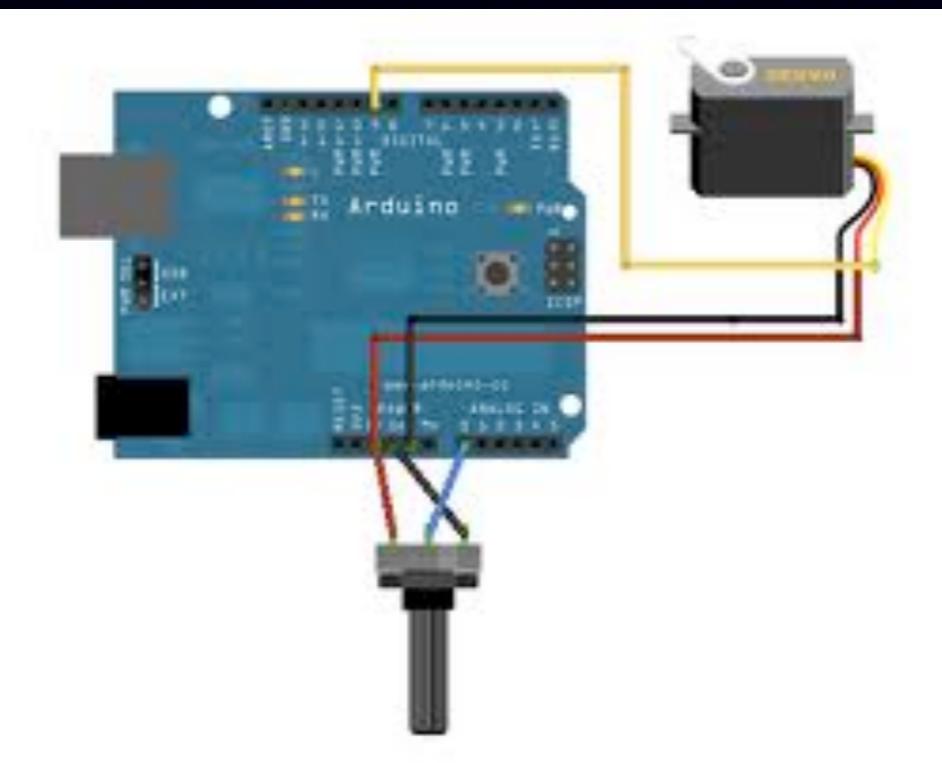
Where to get stuff

How to do some stuff

Ideas



# Apple ][+ hacking in Hardy, AR



## Knob | Arduino 0022



### Knob



```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 179); // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val); // sets the servo position according to the scaled value
  delay(15); // waits for the servo to get there
}
```

```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

#include <Servo.h>

Servo myservo;      // create servo object to control a servo

int potpin = 0;     // analog pin used to connect the potentiometer
int val;           // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  // reads the value of the potentiometer (value between 0 and 1023)
  val = analogRead(potpin);

  // scale it to use it with the servo (value between 0 and 180)
  val = map(val, 0, 1023, 0, 179);

  // sets the servo position according to the scaled value
  myservo.write(val);

  // waits for the servo to get there
  delay(15);
}
```



# Yes I have come a Long Way



# Now

Pittsburgh Perl Workshop

YAPC::NA

Ignite Pittsburgh

Maker Faire Pittsburgh

~FOSS Tech Evangelist

Other stuff ....

# Now

Pittsburgh Perl Workshop

YAPC::NA

Ignite Pittsburgh

Maker Faire Pittsburgh

~FOSS Tech & Product Evangelist

Other stuff ....

FOSS has some of the best Technology  
just not always the best product.

FOSS has some of the best Technology  
just not always the best marketing.





We often have cats on our slides.



This is cat is not Arduino controlled!

# Arduino

Anyone here playing with them yet?

# Arduino

Anyone here playing with them yet?

You will be ... or you should be (at least I think so)

# Arduino

What is it?



PSA  
Please don't schedule a  
NFL playoff came  
opposite of a Perl Oasis  
keynote.

The Arduino language is based on C/C++. It links against [AVR Libc](#) and allows the use of any of its functions;

# Arduino

## Microcontroller

# Arduino

Microcontroller  
IDE

# Arduino

Microcontroller

IDE

Culture

# Arduino

Microcontroller

IDE

Culture (Maker Culture - Not Perl Culture)

# Arduino

Micro controller

IDE

Culture

Open Source

# Arduino

Micro controller

IDE

Culture

Open Source

Amazing Magic

IDE?  
But I like make!

vim > emacs

vim ~ emacs

Yes you can use your  
own makefile!

But I would wait.

Yes you can use your  
own makefile!

But I would or  
contribute

# Google

# OSHW Definition v1.0

<http://www.openhardwaresummit.org/>

# Arduino Boards

Nano

Mini

LiliPad

Duemilanove

Mega

# Arduino Boards

Nano

Mini

LiliPad

Duemilanove is now UNO (Americans)

Mega

<http://sparkfun.com>

<http://adafruit.com>

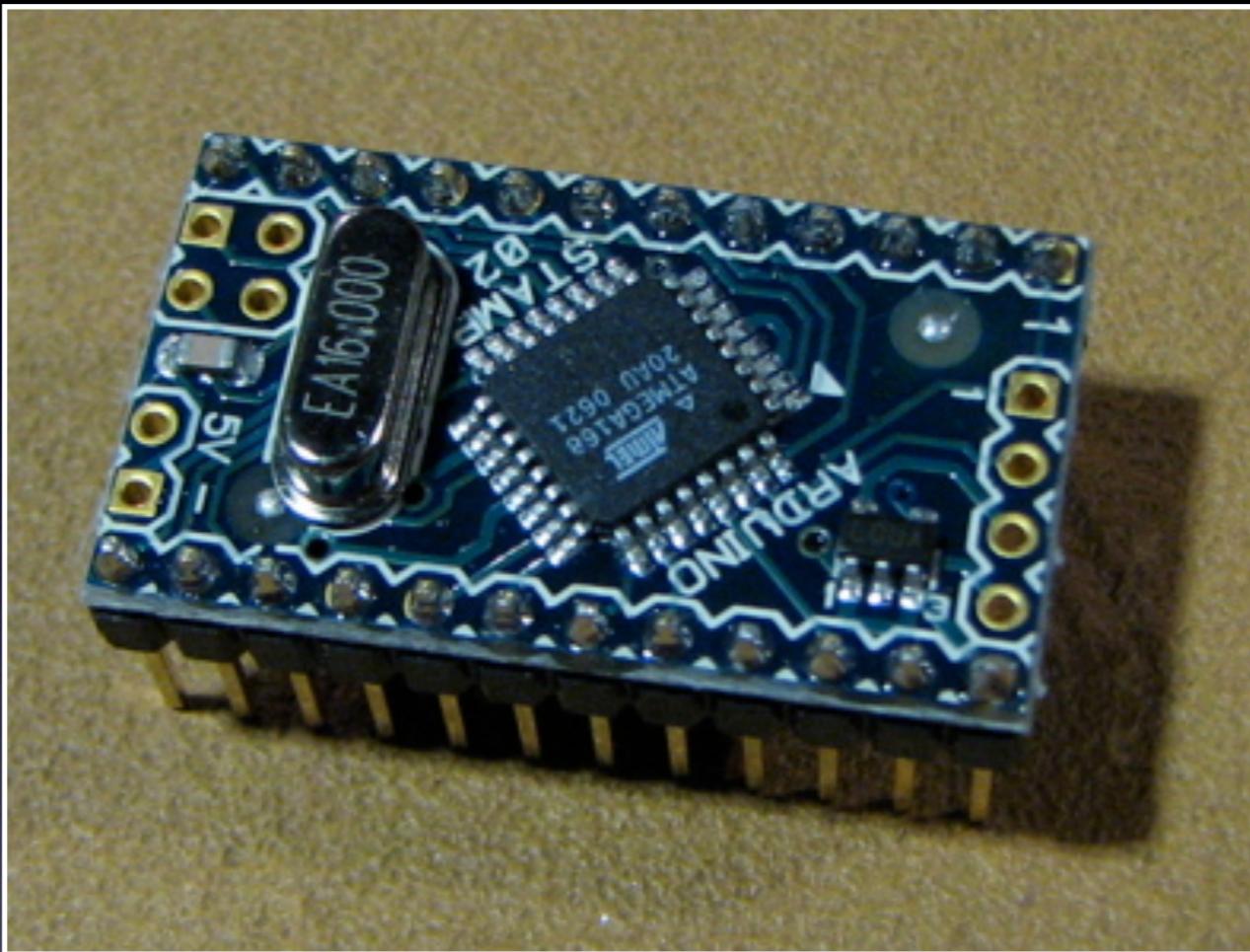
<http://makershed.com>

<http://shieldlist.org/>

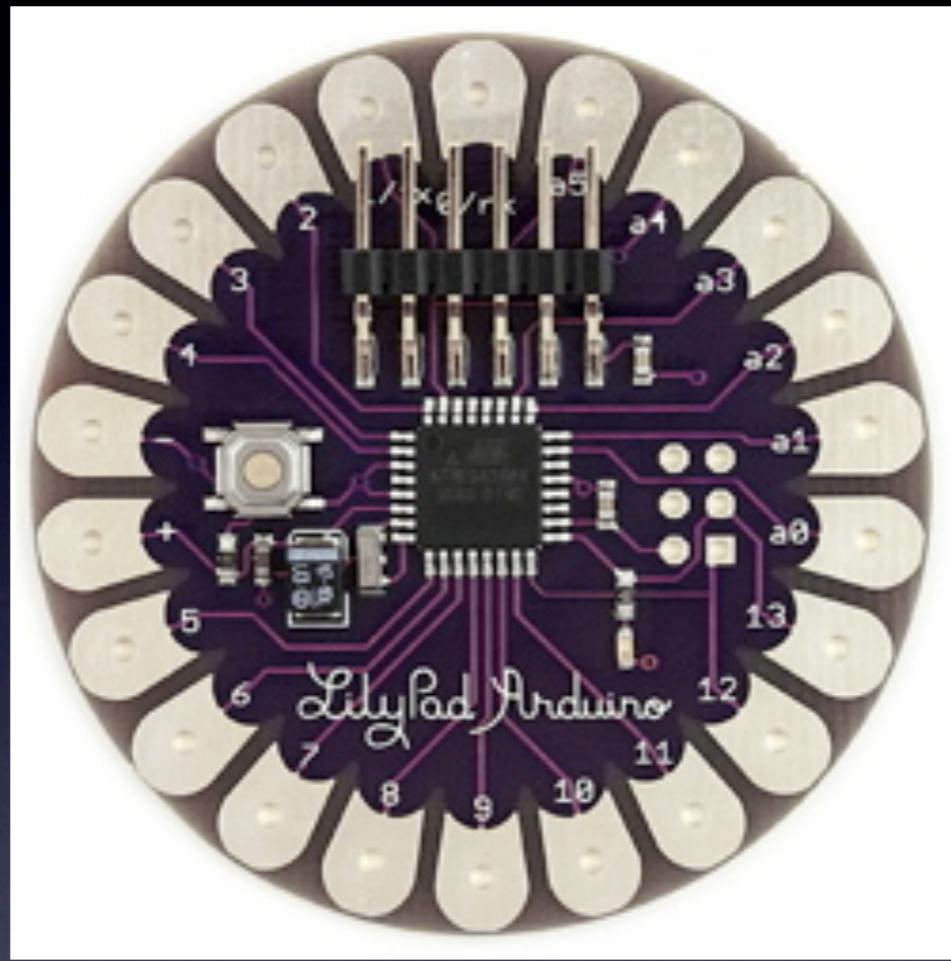
Lots more



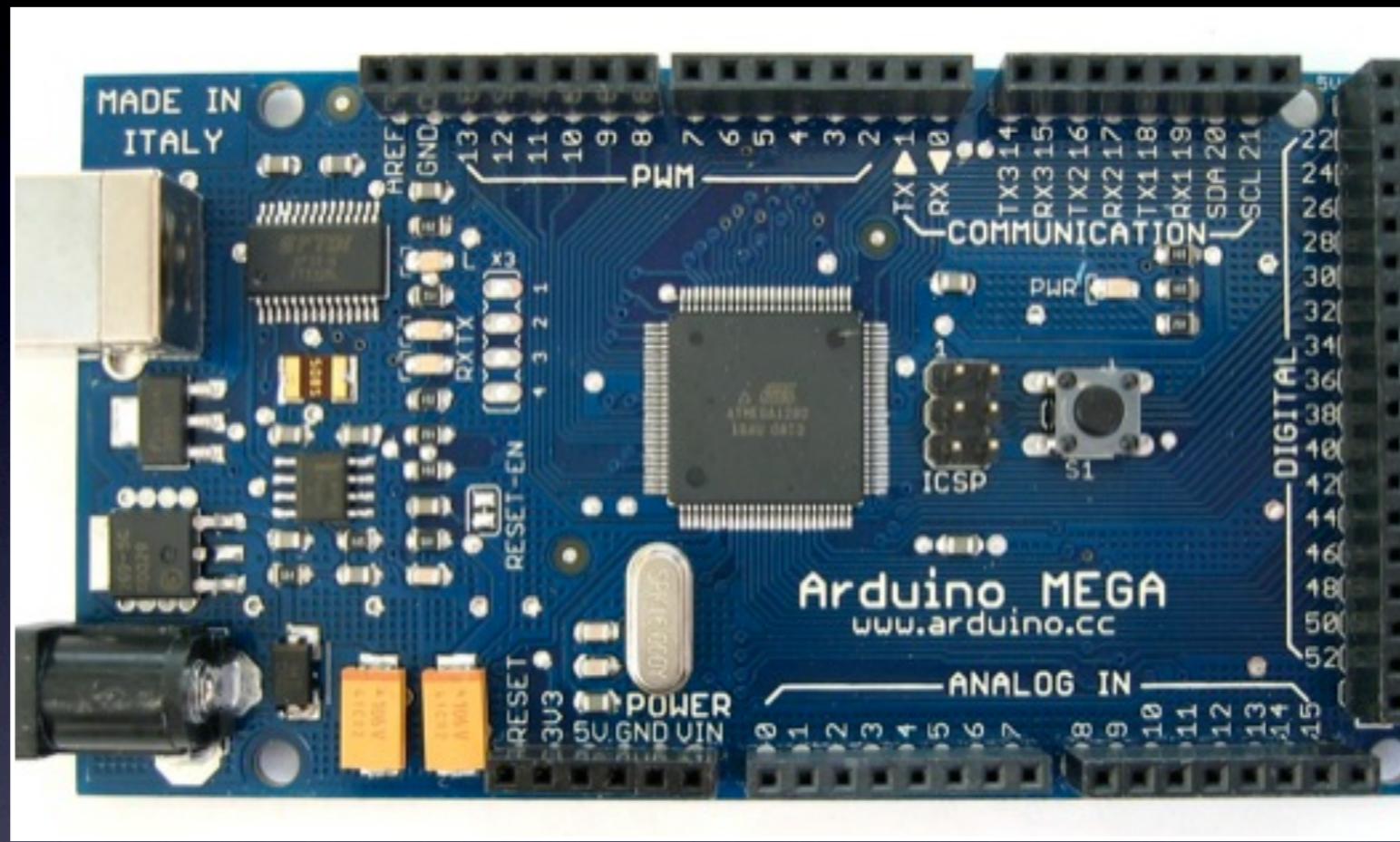
# Nano



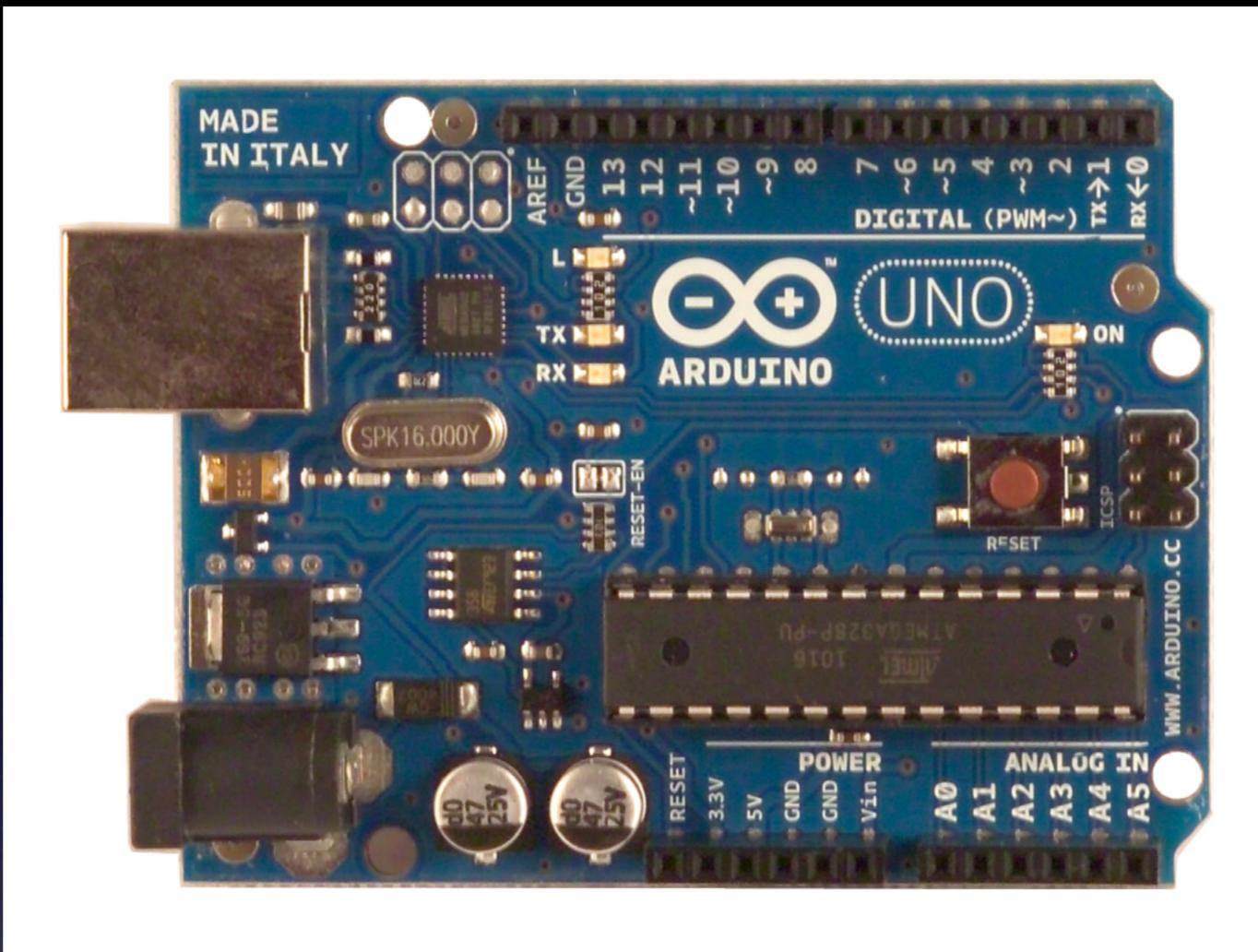
# Mini



# Lillypad

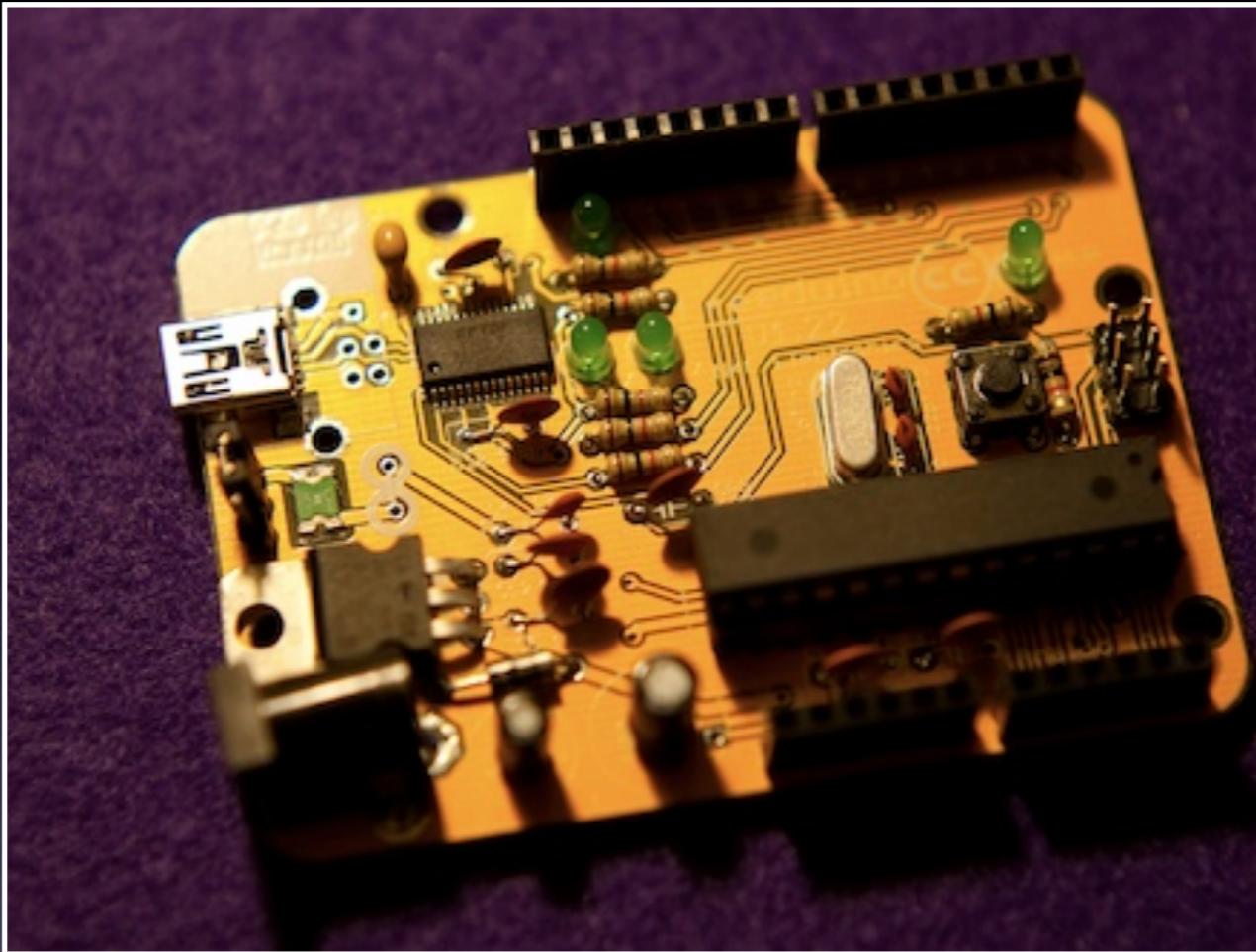


# Mega



# UNO

\$30.00



**concurrency.cc**

Parallel programming for makers and artists

**Occam on the Arduino.**

# Arduino Sheilds

Ethernet

TellyMate

Motor Shields

GPS

Breathalyzer

Custom Built Shields

Video Game Shield

etc

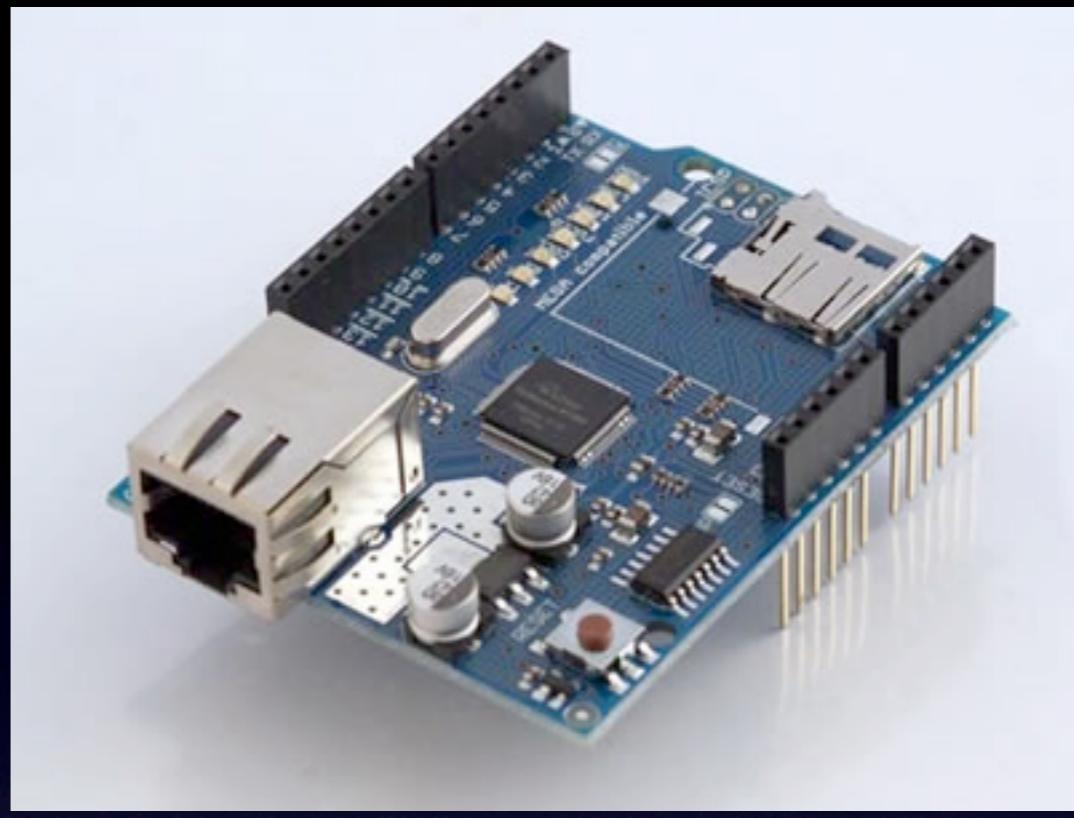
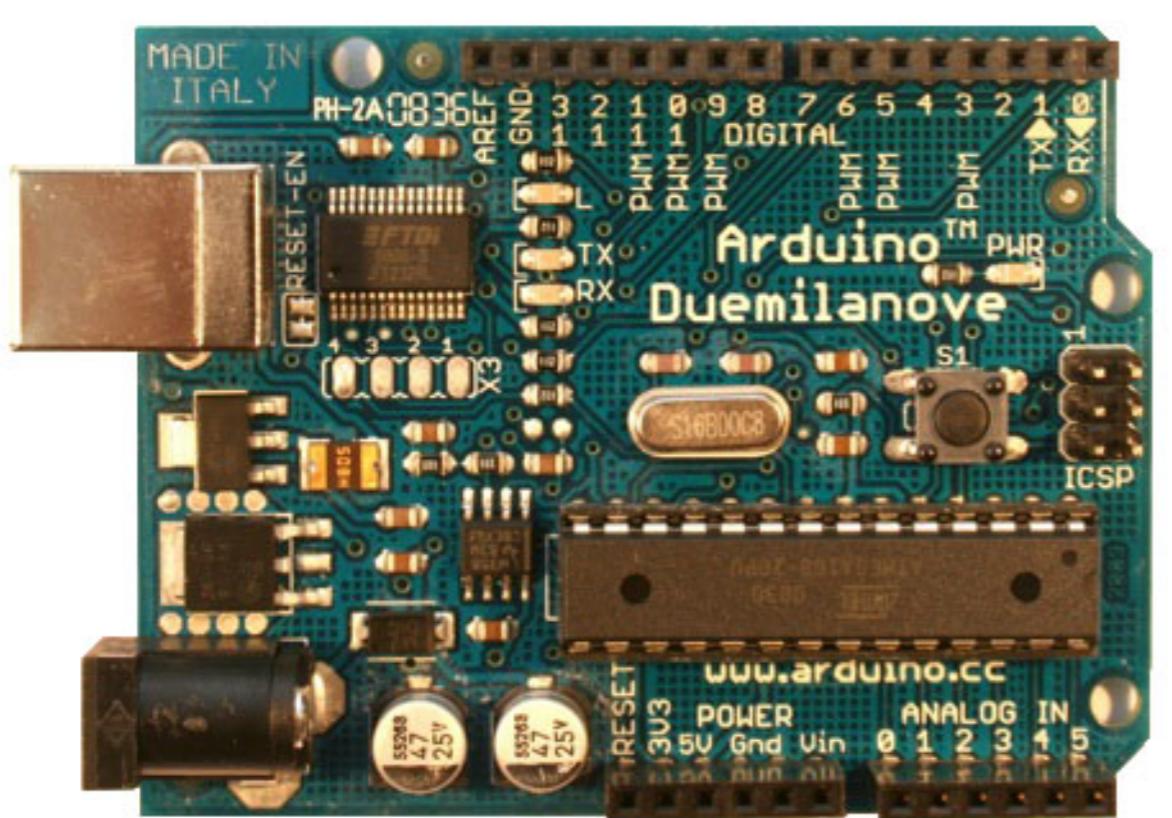


# Tellymate

```
/* Example Sketch for a TellyMate Shield */
/* Hello World */
void setup()
{
    Serial.begin( 56700 ) ;
    Serial.println( "Hello World!" ) ;
}

void loop()
{
    // do nothing!
}
```

[batsocks.co.uk/downloads/tms\\_hello\\_001.pde](http://batsocks.co.uk/downloads/tms_hello_001.pde)



# Duemilanove + Ethernet

~\$30.00 + ~\$40.00



# dd-wrt

# USB HUB!

USB HUB!  
Anyone ever set their  
monitor on fire ...

# Projects

Network testing

Power reset for testing

Halloween tour

Toys

DBD::"Internet of Things?"

jQuery Mobile with RESTduino

...

# RESTduino

<http://192.168.1.177/9>  
{"9":"LOW"}

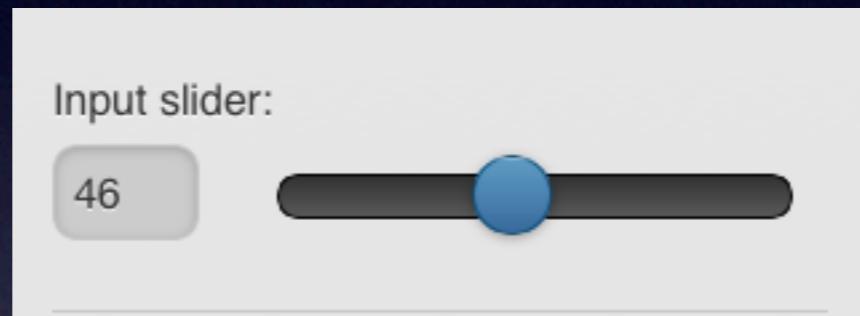
# RESTduino

<http://192.168.1.177/9>

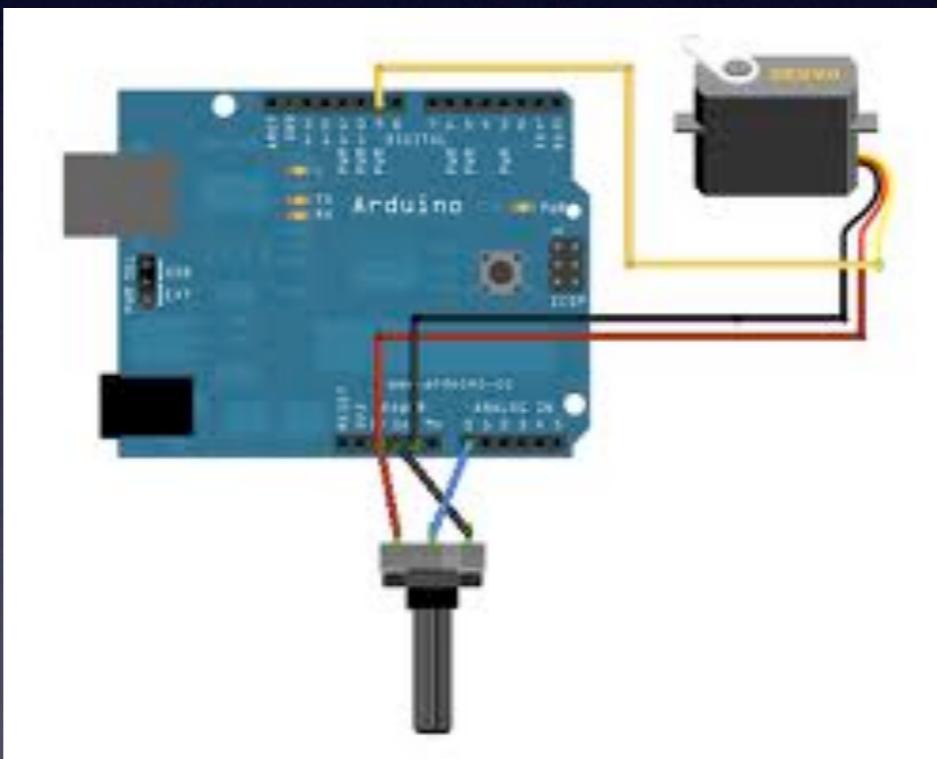
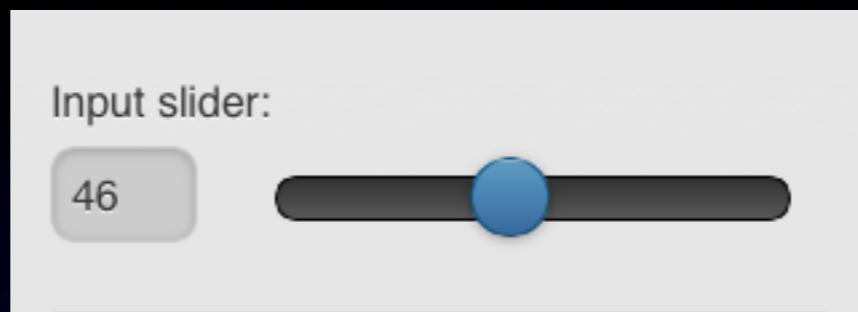
{“9”：“LOW”, “SERVER”, “192.168.1.177”}

curl <http://192.168.1.177/9> | jcut -o 9

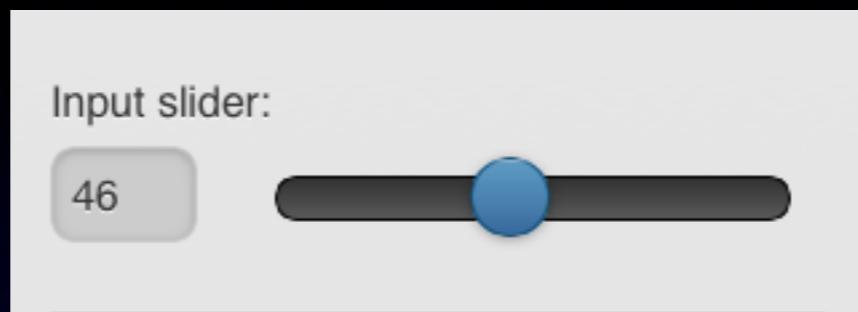
# jQuery Mobile



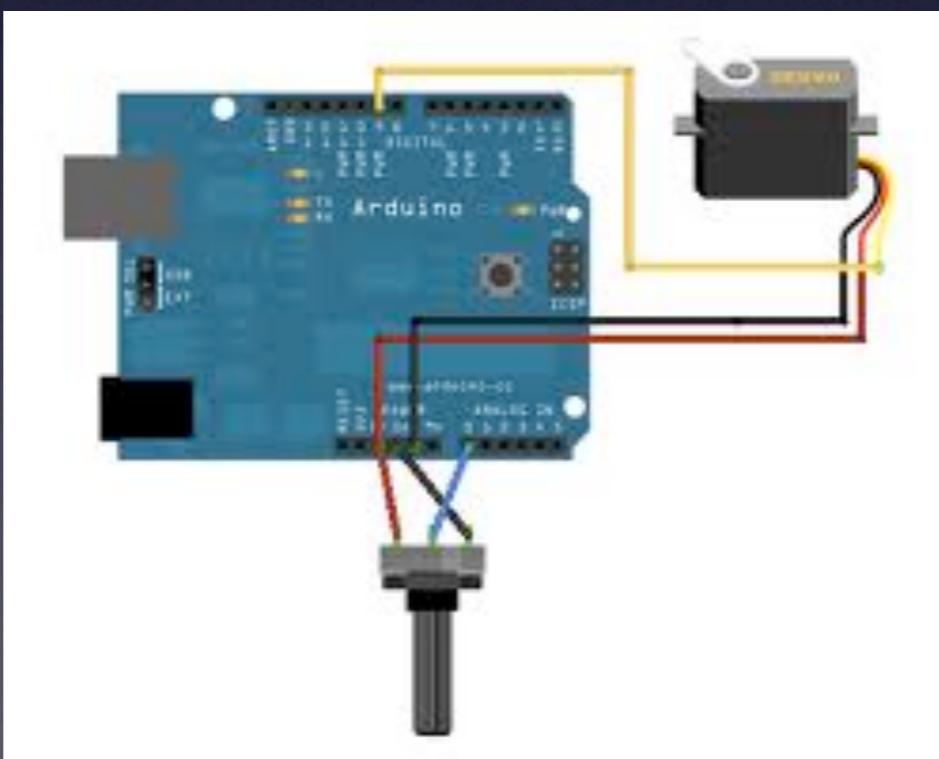
# jQuery Mobile



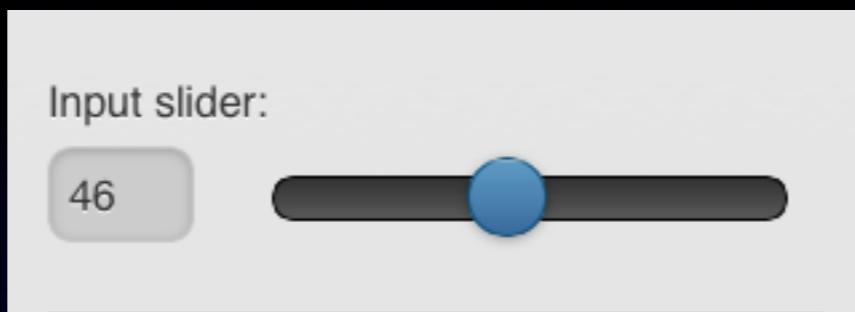
# jQuery Mobile



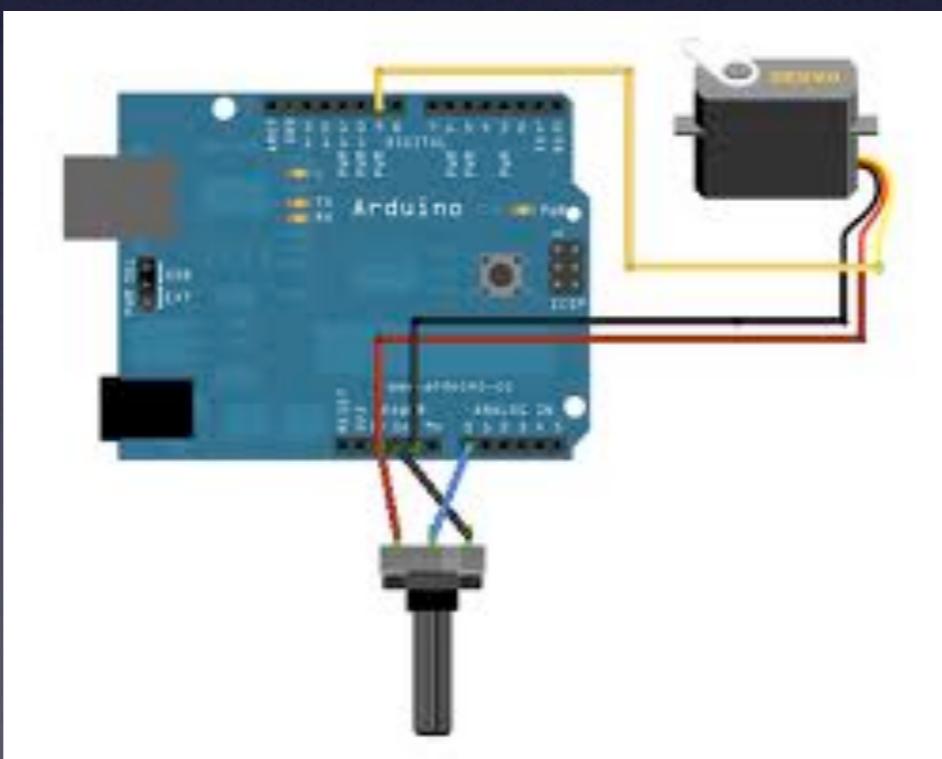
Now Wireless!



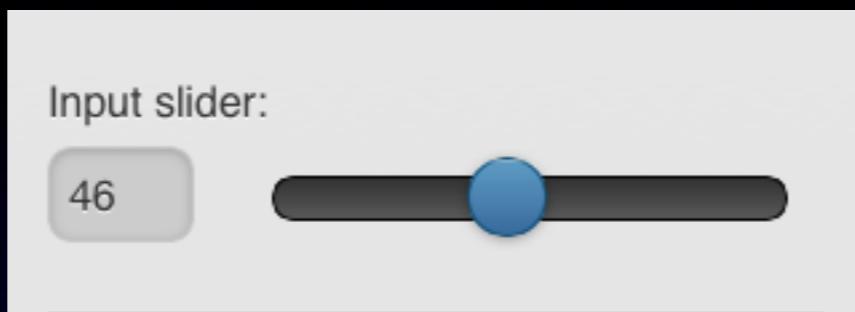
# jQuery Mobile



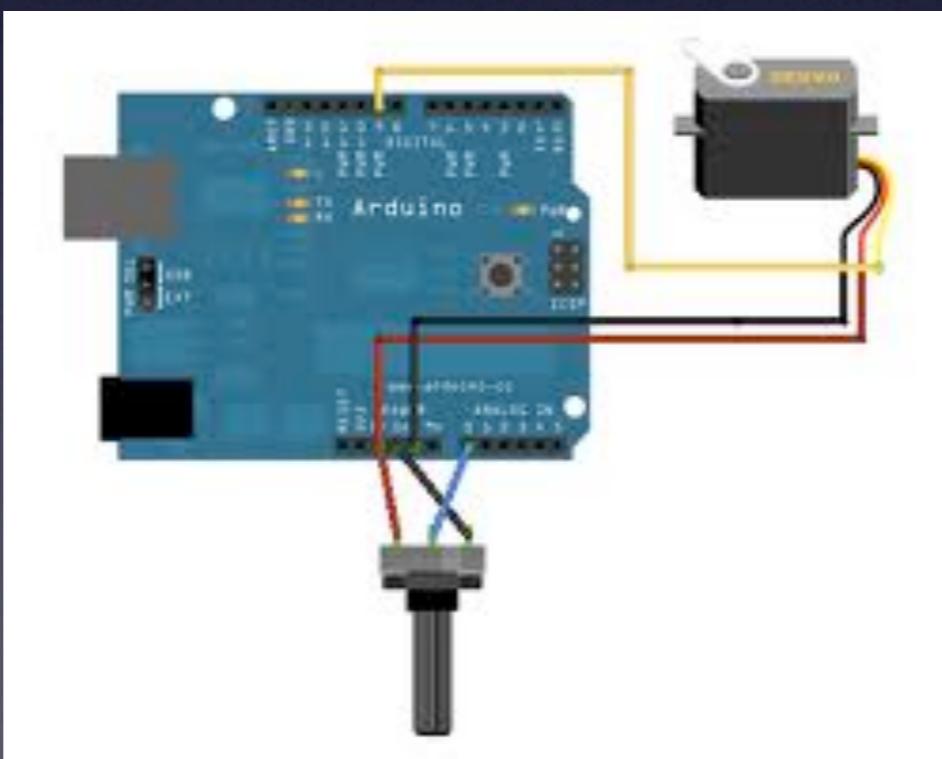
Now Wireless!  
AND EASY



# jQuery Mobile



Now Wireless!  
AND EASY(ISH)



# **Basement Doors**



Thursday, June 30, 2011



# Basement doors

Alert me if both basement doors are open.

Fine if both closed.

Fine if one closed.

Time based? Event based? Use cases!

Alarm System

Programs are Sketches  
in Arduino.

# Two Functions

- setup
- loop

# Basic Blinking LED

# Basic Blinking LED (Hello World)

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch\_sep04b | Arduino 0018
- Toolbar:** Includes icons for play, stop, upload, download, and other functions.
- Sketch Name:** sketch\_sep04b §
- Code Area:** Displays the `Blink` sketch code. The code initializes pin 13 as an output and alternates its state between HIGH and LOW every second using `delay(1000)`.
- Status Bar:** Shows "Done uploading." and "Binary sketch size: 888 bytes (of a 30720 byte maximum)".
- Page Number:** 18

```
/*
Blink
Turns on an LED on for one second, then off for one second,
repeatedly.
```

This example code is in the public domain.

```
*/
```

```
void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has an LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);      // set the LED on
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // set the LED off
    delay(1000);                // wait for a second
}
```

# setup

```
void setup()
{
    // sets the digital pin as output
    pinMode(13, OUTPUT);
}
```

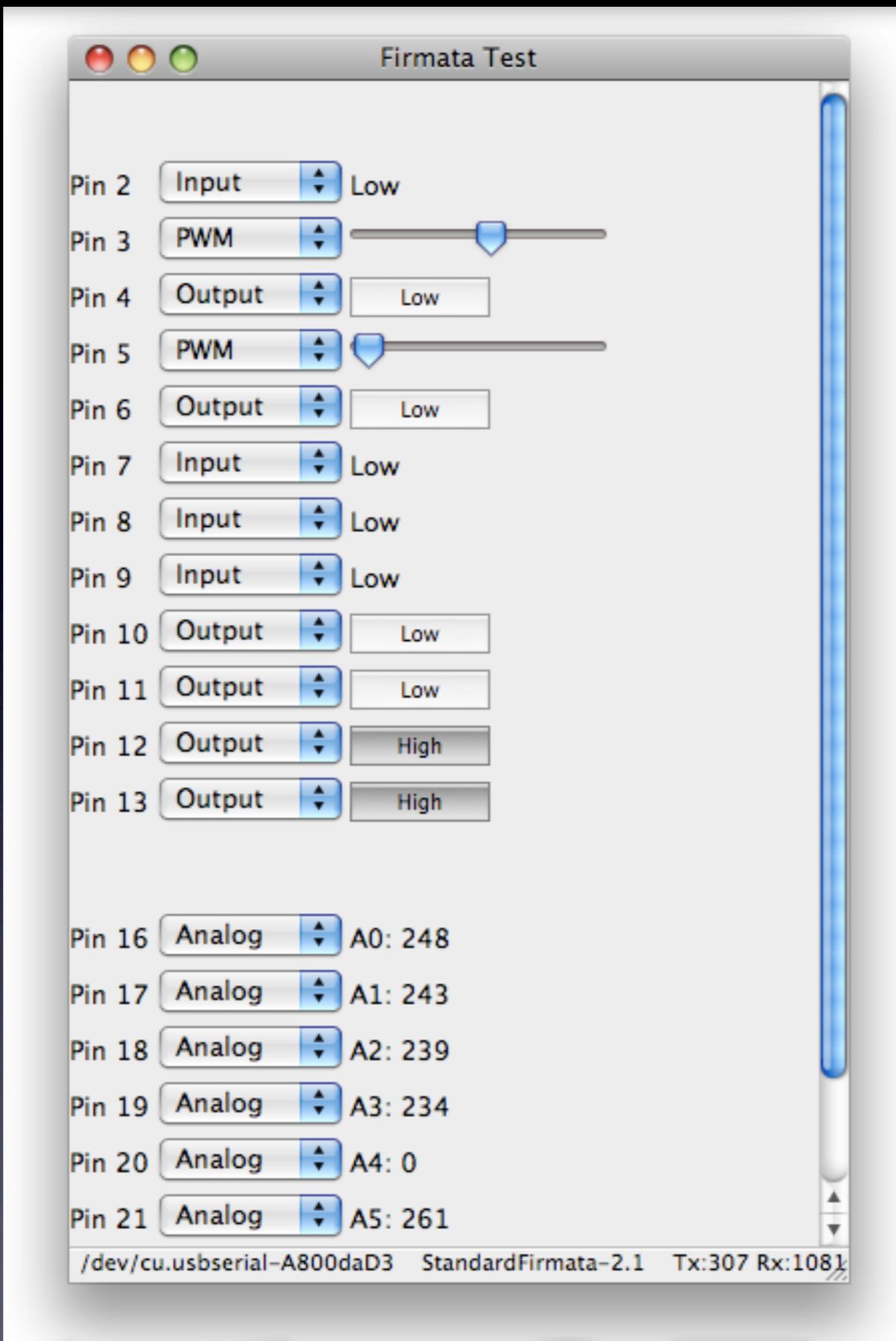
# Loop

```
void loop()
{
    digitalWrite(13, HIGH);      // sets the LED on
    delay(1000);                // waits for a second
    digitalWrite(13, LOW);       // sets the LED off
    delay(1000);                // waits for a second
}
```

# Firmata

**Firmata is a generic protocol  
for communicating with  
microcontrollers from software  
on a host computer.**

**[firmata.org](http://firmata.org)**



```
use Device::Firmata;
use Device::Firmata::Constants qw/:all/;

my $device = Device::Firmata->open('/dev/ttyUSB0');

$device->pin_mode(13=>PIN_OUTPUT);
while (1) {
    $device->digital_write(13=>1);
    sleep(1);
    $device->digital_write(13=>0);
    sleep(1);
}
```

# Device::Firmata

<https://github.com/amimoto/perl-firmata>

Not on CPAN

**Reflexive::Firmata** - an Asynchronous OO interface to serial Firmata devices

Current status: Work in progress. Committed to github and Gitorious to facilitate collaborative hacking.

Community: irc.perl.org #reflex (for now)

License: Unless otherwise specified, Same Terms as Perl Itself. This may change.

The repository contains some other-copyright files that are being used

<https://github.com/rkaputo/rx-firmata>

# Ways to connect

- Serial
- Ethernet

# Serial

# BLINK over Serial

```
int led = 13;

void setup() {
  Serial.begin(9600);
  pinMode(led,OUTPUT);
}

void loop() {
  while (Serial.available()>0) {
    if (Serial.read() =='1') {
      digitalWrite(led, HIGH);
      Serial.println ("ON");
    } else {
      digitalWrite(led, LOW);
      Serial.println ("OFF");
    }
  }
  delay(10);
}
```



```
use Device::SerialPort;  
my $port =  
    Device::SerialPort->("/dev/ttyUSB0");  
$port or die;  
$port->baudrate(9600);  
$port->databits(8);  
$port->parity("none");  
$port->stopbits(1);  
  
$port->write( "1" );
```

# Happy Loop

Do you know where your keys are?

Have you ever lost your  
keys?

# Me Too!

We can fix that with  
an ...

# Aruduino

&

A little bit of  
programming.

# A little Perl code

```
1 use Test::More tests => 2;
2 use Test::WWW::Mechanize;
3 use Getopt::Long;
4
5 my %h = ();
6 GetOptions (\%h, 'url=s');
7
8 my $mech = Test::WWW::Mechanize->new;
9
10 $mech->get_ok( $h{url} );
11
12 if ( keys_should_be_home() ) {
13     $mech->text_contains( 'my keys are home', 'Key check' );
14 } else {
15     $mech->text_contains('my keys are not home', 'Key check');
16 }
17
18 sub keys_should_be_home {
19     #check http://www.google.com/calendar/
20     return 1;
21 }
```

```
prove -v keys.t :: --url http://172.16.6.56
```

# Keys are not Home

Keys.t ..

1..2

ok 1 - GET <http://172.16.6.56>

not ok 2 - Key check

```
# Failed test 'Key check'  
# at keys.t line 12.  
#   searched: "my keys are not home."  
#   can't find: "my keys are home"  
#       LCSS: "my keys are "  
# LCSS context: "my keys are not home."  
# Looks like you failed 1 test of 2.  
Dubious, test returned 1 (wstat 256, 0x100)  
Failed 1/2 subtests
```

## Test Summary Report

---

keys.t (Wstat: 256 Tests: 2 Failed: 1)

Failed test: 2

Non-zero exit status: 1

Files=1, Tests=2, 0 wallclock secs ( 0.02 usr 0.00 sys + 0.19 cusr 0.01 csys = 0.22 CPU)

Result: FAIL

# Keys are Home

```
keys.t ..
1..2
ok 1 - GET http://172.16.6.56
ok 2 - Key check
ok
All tests successful.
Files=1, Tests=2, 0 wallclock secs ( 0.03 usr
0.00 sys + 0.19 cusr 0.01 csys = 0.23 CPU)
Result: PASS
```

```
prove -v keys.t :: --url http://172.16.6.56
```

Put this in cron or the like.

```
prove -v keys.t :: --url http://172.16.6.56
    ^A - arisdottle
Put this in cron or the like.
```

# Arduino Code

# Webserver from Example

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 10, 0, 0, 177 };

Server server(80);

void setup()
{
  Ethernet.begin(mac, ip);
  server.begin();
}

void loop()
{
  Client client = server.available();
  if (client) {
    // an http request ends with a blank line
    boolean current_line_is_blank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        // if we've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so we can send a reply
        if (c == '\n' && current_line_is_blank) {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();

          // output the value of each analog input pin
          for (int i = 0; i < 6; i++) {
            client.print("analog input ");
            client.print(i);
            client.print(" is ");
            client.print(analogRead(i));
            client.println("<br />");
          }
          break;
        }
        if (c == '\n') {
          // we're starting a new line
          current_line_is_blank = true;
        }
      }
    }
  }
}
```

# Webserver from Example

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 10, 0, 0, 177 };

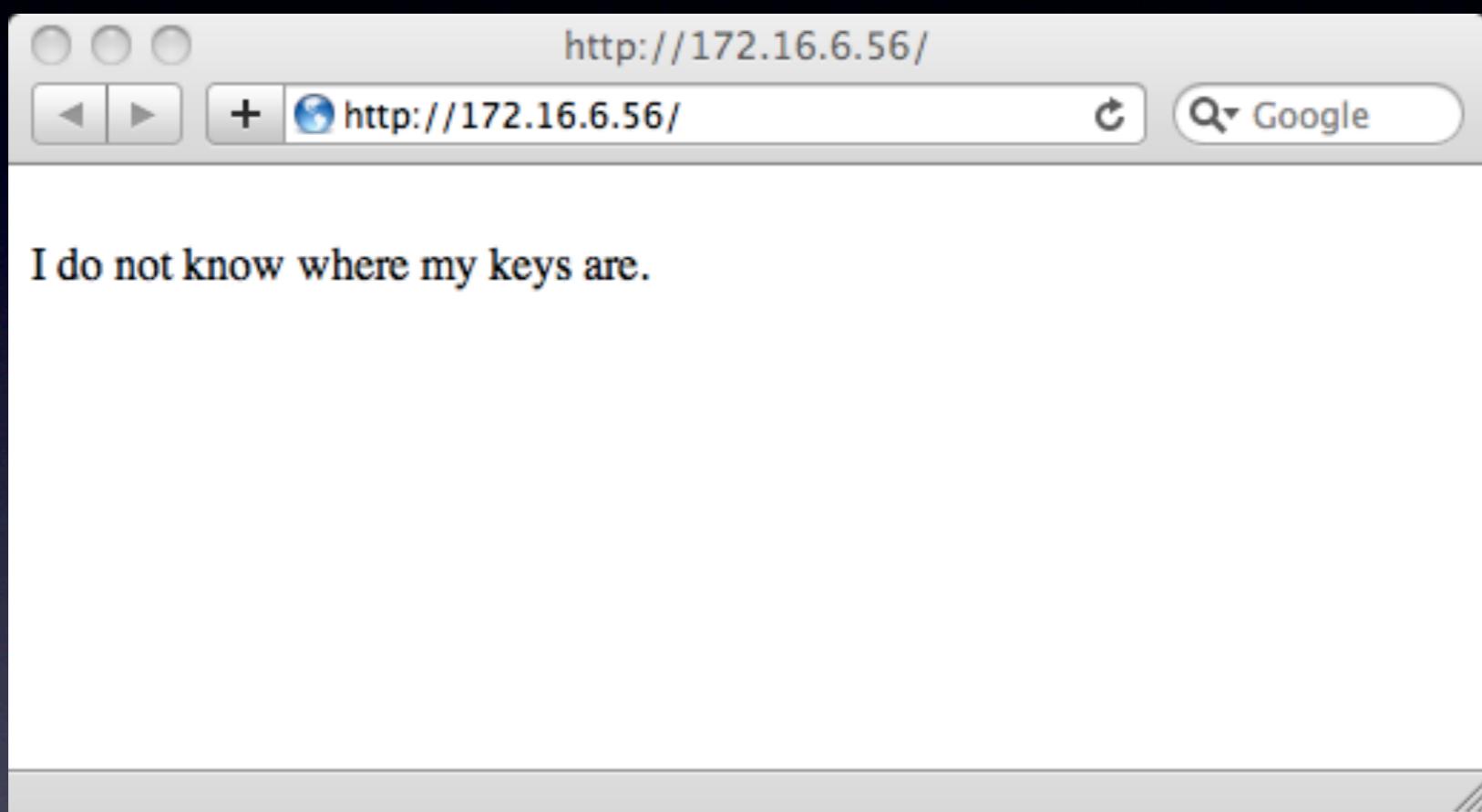
Server server(80);

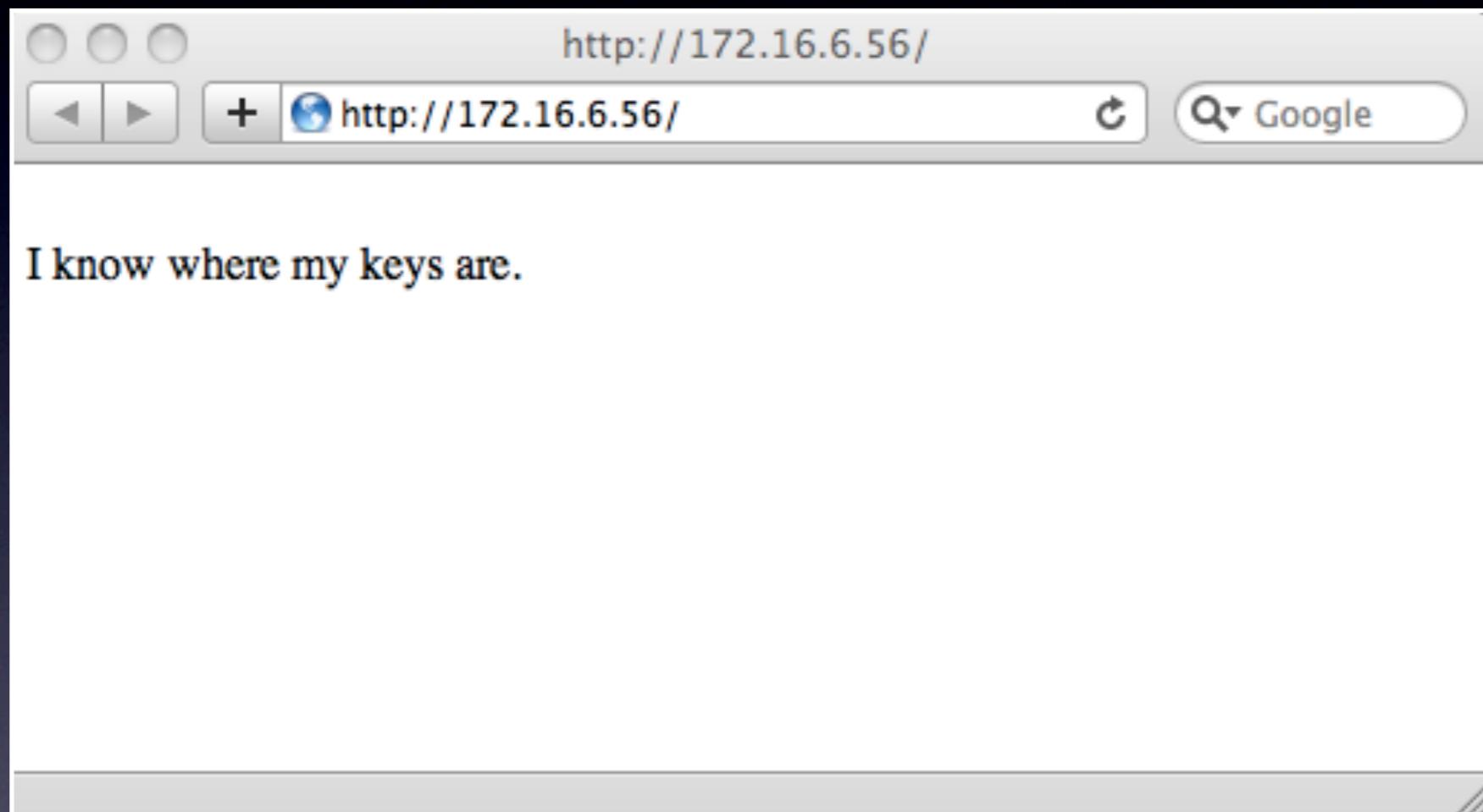
void setup()
{
  Ethernet.begin(mac, ip);
  server.begin();
}

void loop()
{
  Client client = server.available();
  if (client) {
    // an http request ends with a blank line
    boolean current_line_is_blank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        // if we've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so we can send a reply
        if (c == '\n' && current_line_is_blank) {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();

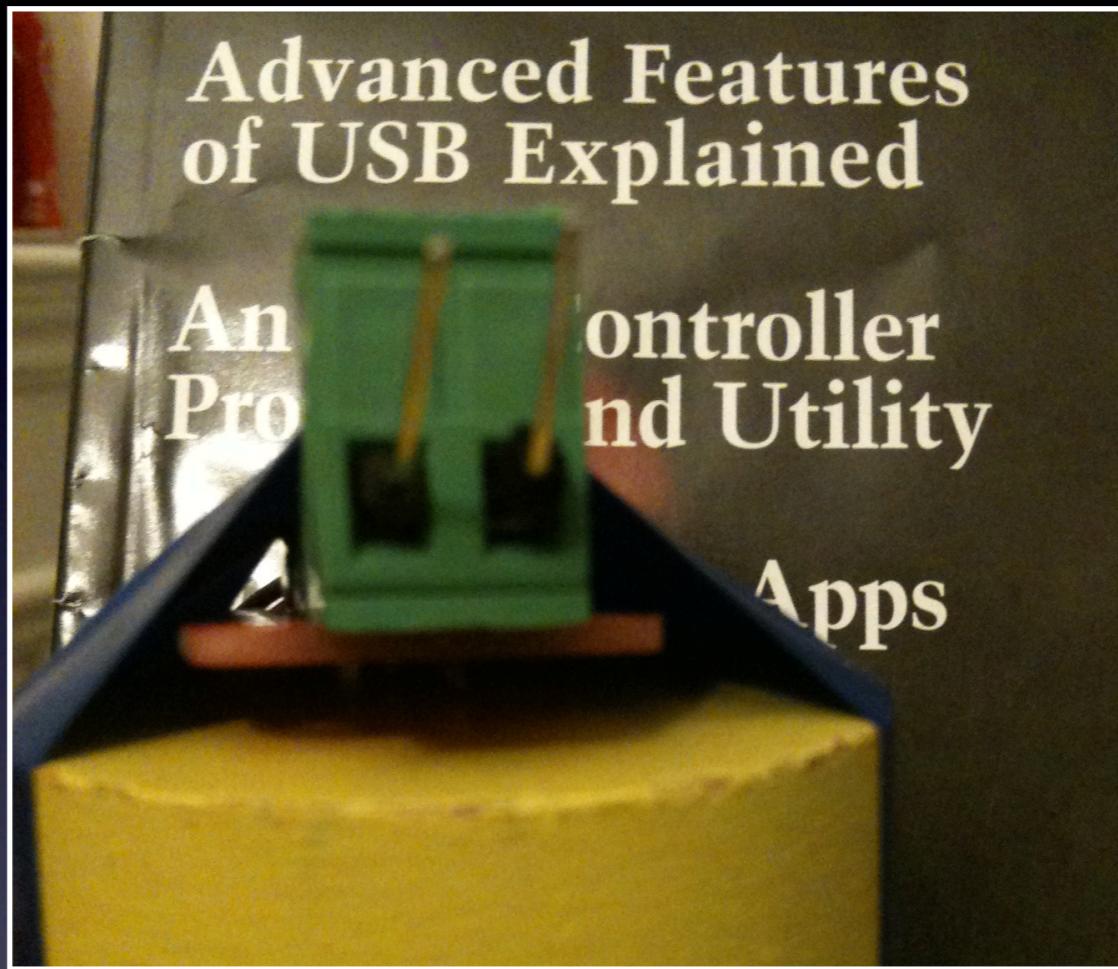
          // output the value of each analog input pin
          for (int i = 0; i < 6; i++) {
            client.print("analog input ");
            client.print(i);
            client.print(" is ");
            client.print(analogRead(i));
            client.println("<br />");
          }
          break;
        }
        if (c == '\n') {
          // we're starting a new line
          current_line_is_blank = true;
        }
      }
    }
  }
}
```

```
buttonState = digitalRead(buttonPin) ;  
  
client.println("<br />");  
if( buttonState == true ) {  
    client.println("I know where my keys are.");  
} else {  
    client.println("I do not know where my keys  
are.");  
}  
  
client.println("<br />");
```

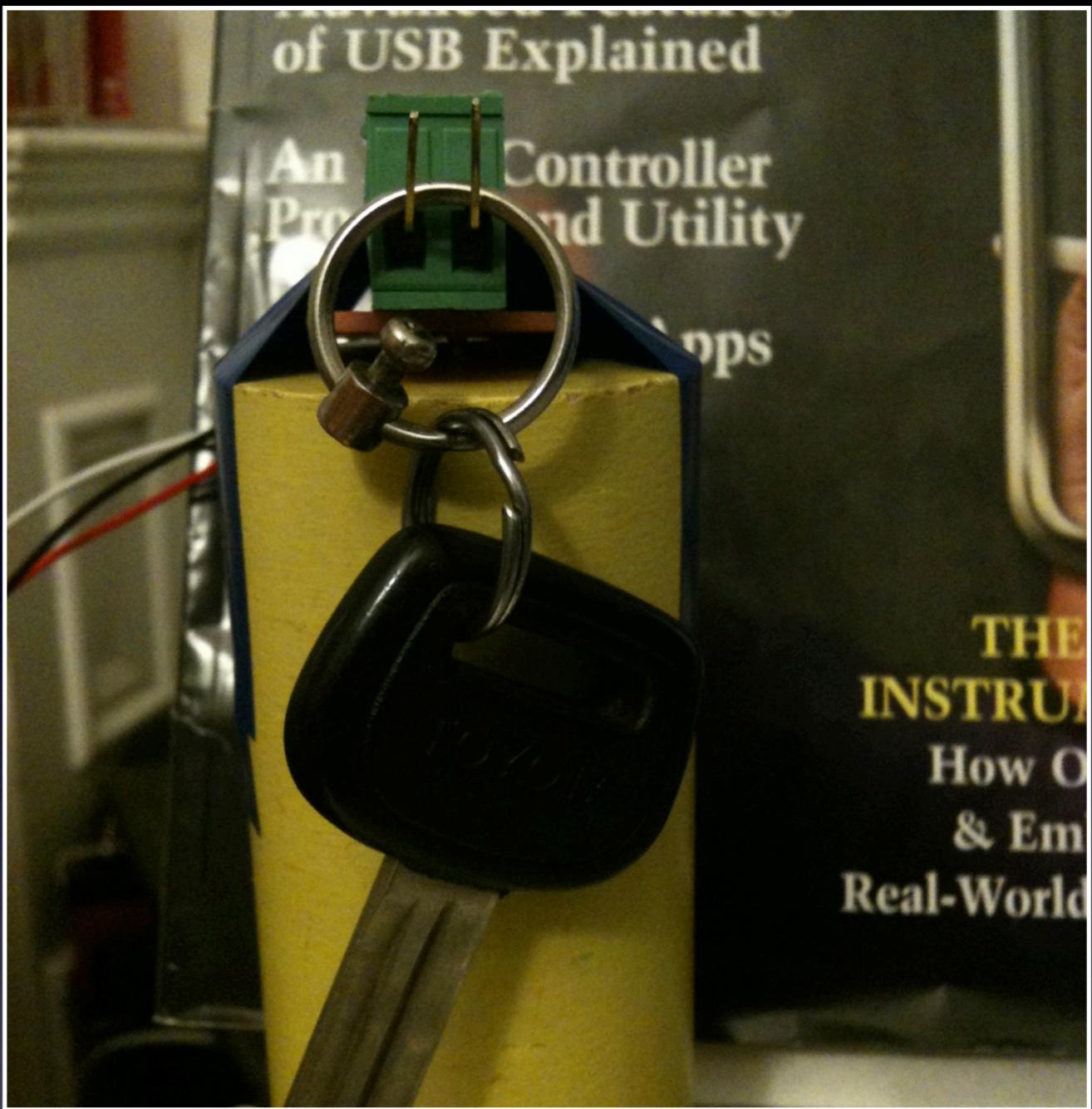




I know where my keys are.



# No Keys



# Keys

# Doorbell

In my neighborhood of  
Manchester in  
Pittsburgh everywhere  
is home.

```
1 print("Content-Type: text/txt\n\n");
2
3 print "ding dong";
4
5 notify_mail();
6
7 sub notify_mail {
8     use Email::Sender::Simple qw(sendmail);
9     use Email::Simple;
10    use Email::Simple::Creator;
11
12    my $email = Email::Simple->create(
13        header => [
14            To      => 'robert@robertblackwell.com',
15            From    => 'robert@robertblackwell.com',
16            Subject => "Ding Dong",
17        ],
18        body => "It is " . localtime() . ". Some one is
19                    at the door.?\\n",
20    );
21    sendmail($email);
22 }
```

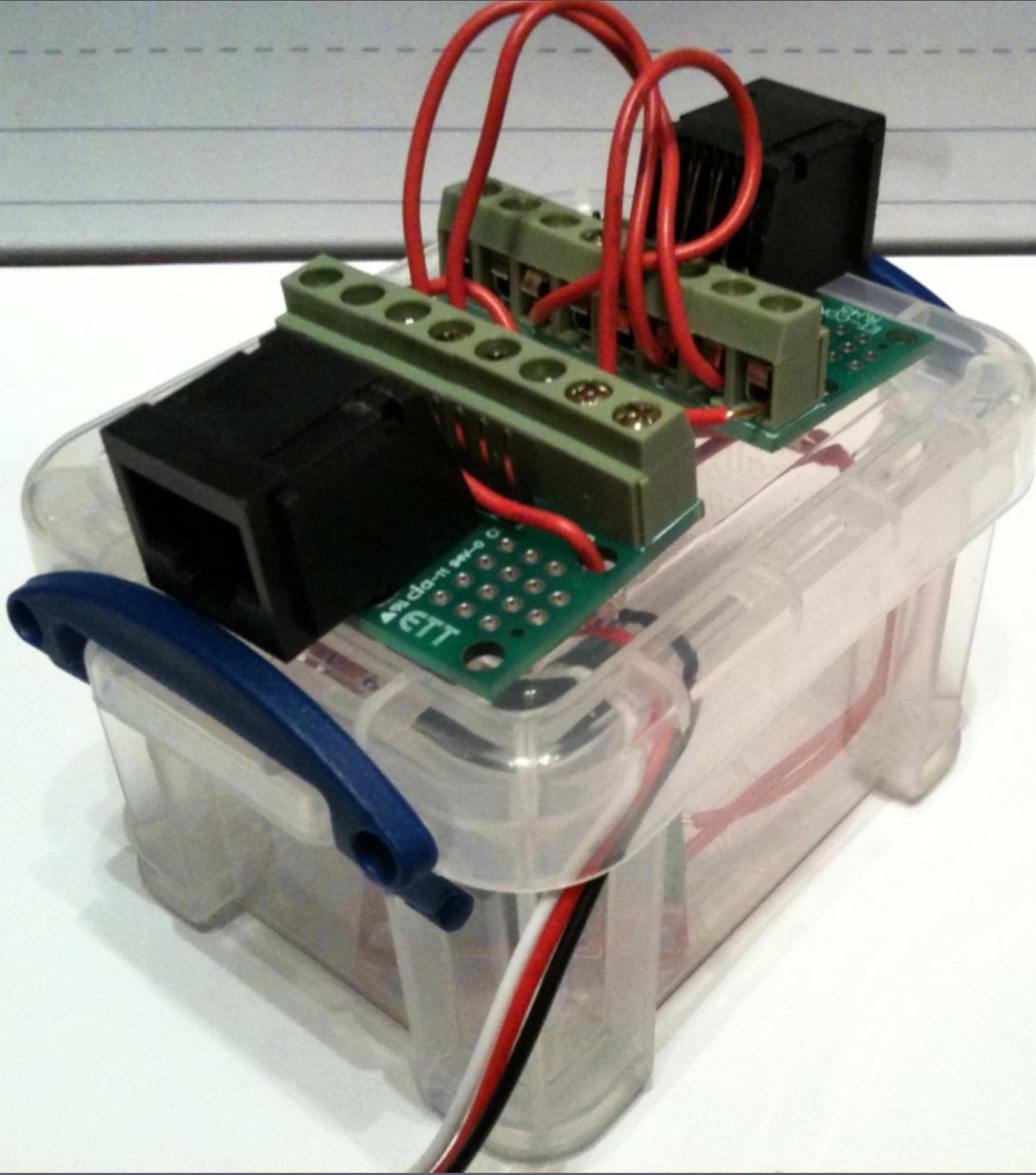
```
#include <Ethernet.h>
void ding_dong();
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 172, 16, 6, 56 };
byte server[] = { 172, 16, 6, 54 };

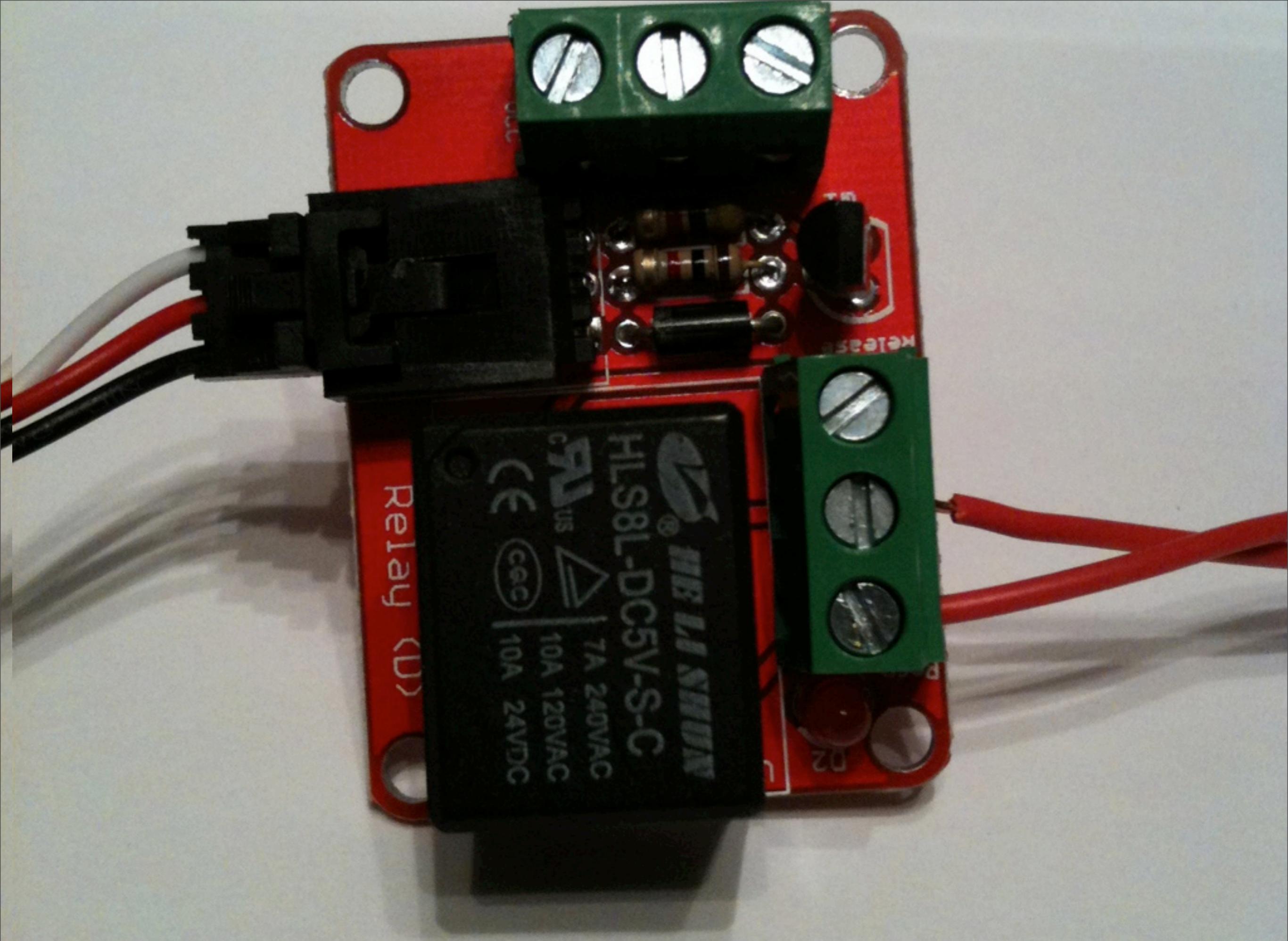
Client client(server, 80);
```

```
void setup()
{
  Ethernet.begin(mac, ip);
  Serial.begin(9600);
}
```

```
void loop()
{
    buttonState = digitalRead(buttonPin) ;
    if( buttonState == true ) {
        Serial.println("Someone is at the door.");
        ding_dong();
        delay(1000);
    } else {
        Serial.println("Nobody at the door");
    }
    delay(1000);
}
```

```
void ding_dong() {  
    Serial.println("connecting...");  
  
    if (client.connect()) {  
        Serial.println("connected");  
        client.println("GET /doorbell.cgi HTTP/1.0");  
        client.println();  
        client.stop();  
    } else {  
        Serial.println("connection failed");  
    }  
}
```







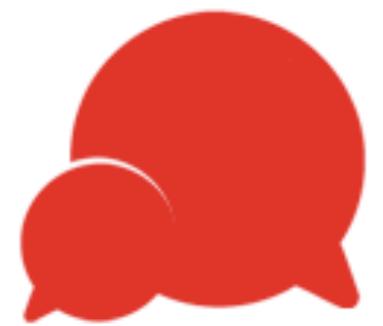
**Store, share & discover realtime sensor, energy  
and environment data from objects, devices &  
buildings around the world.**

**Pachube is a convenient, secure & scalable  
platform that helps you connect to & build the  
'internet of things'.**

**pachube.com**

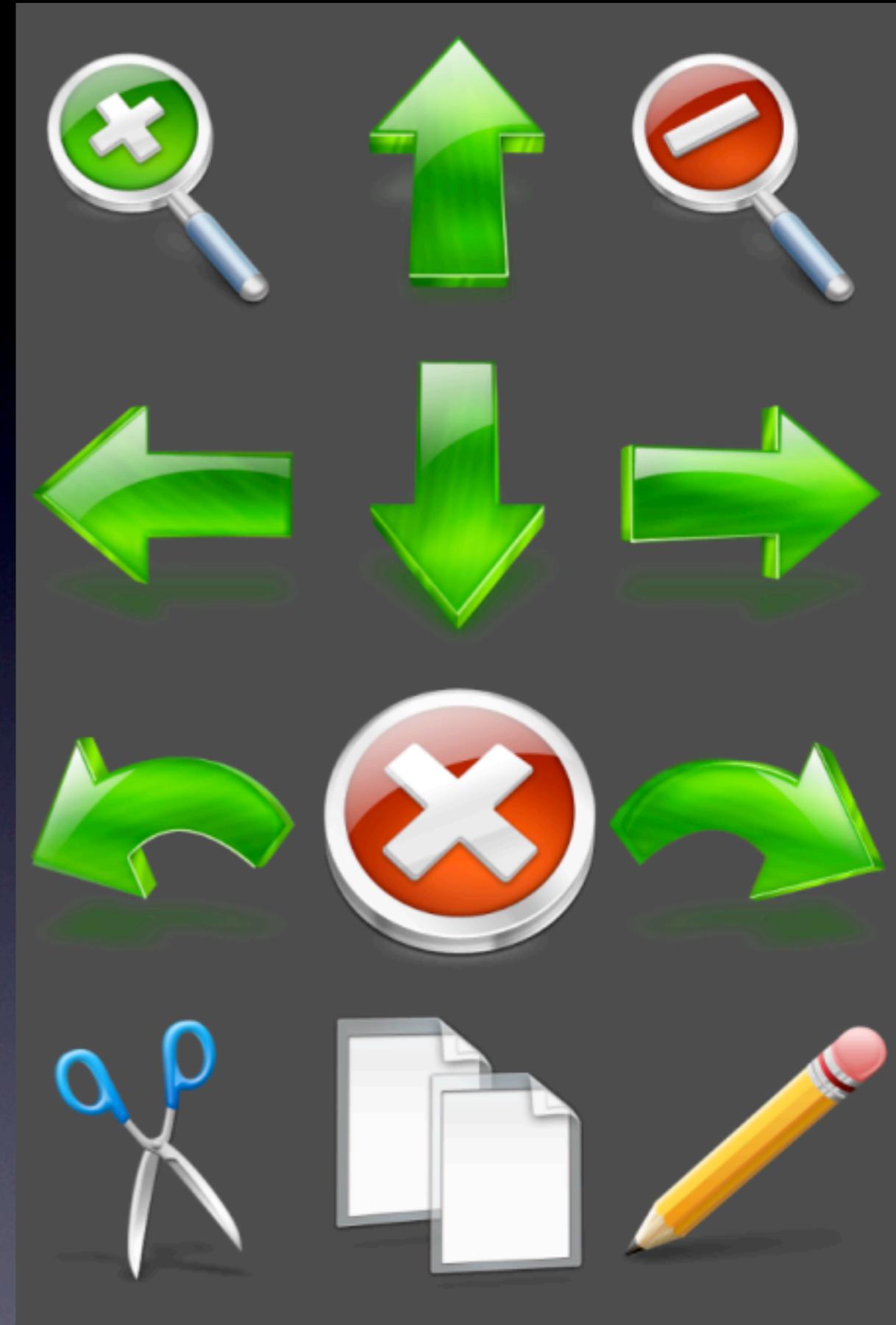
create your  
Internet of Things  
application with  
**ThingSpeak**

[ an open application platform  
designed to enable meaningful  
connections between things  
and people ]



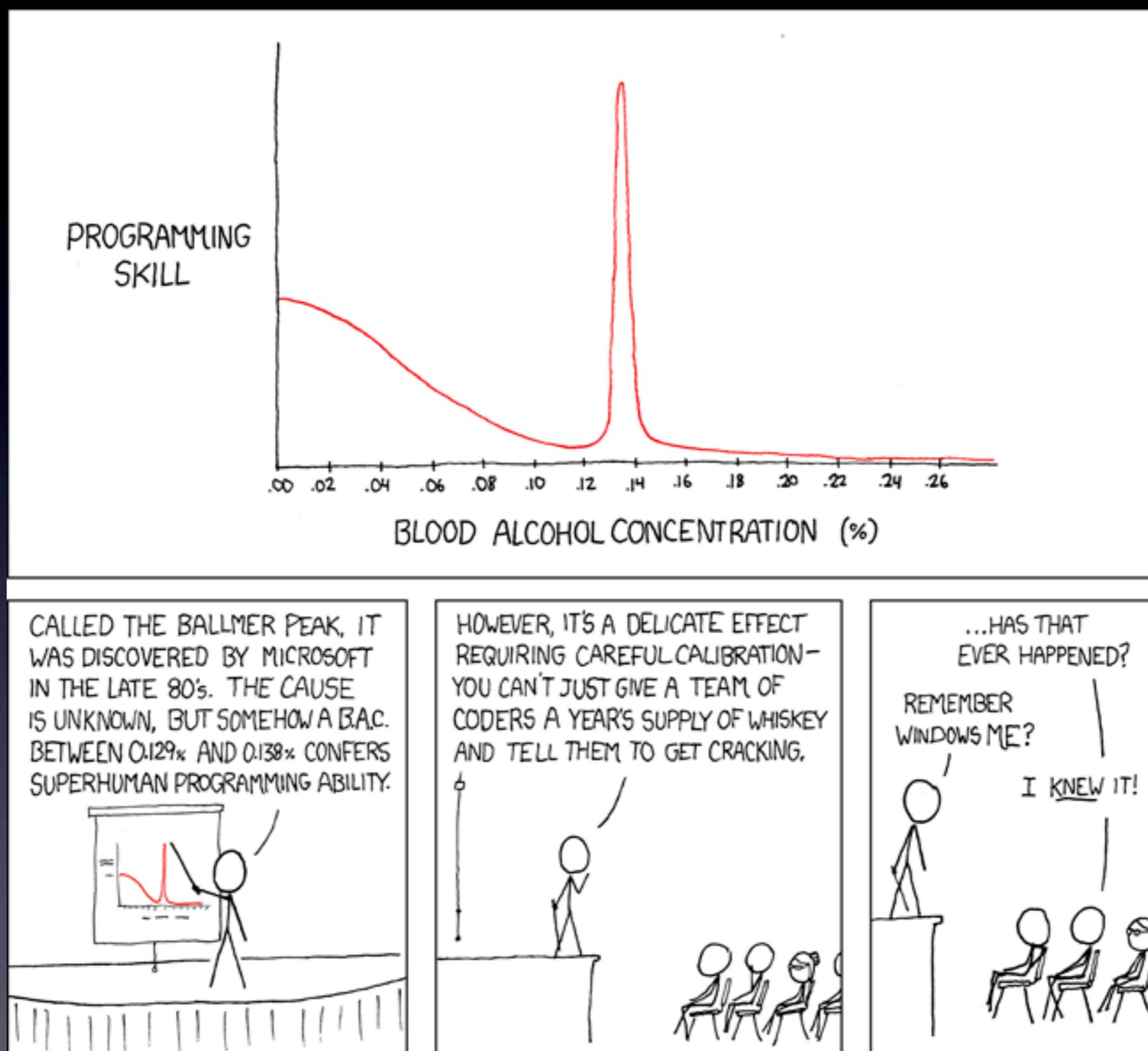
[thingspeak.com](http://thingspeak.com)

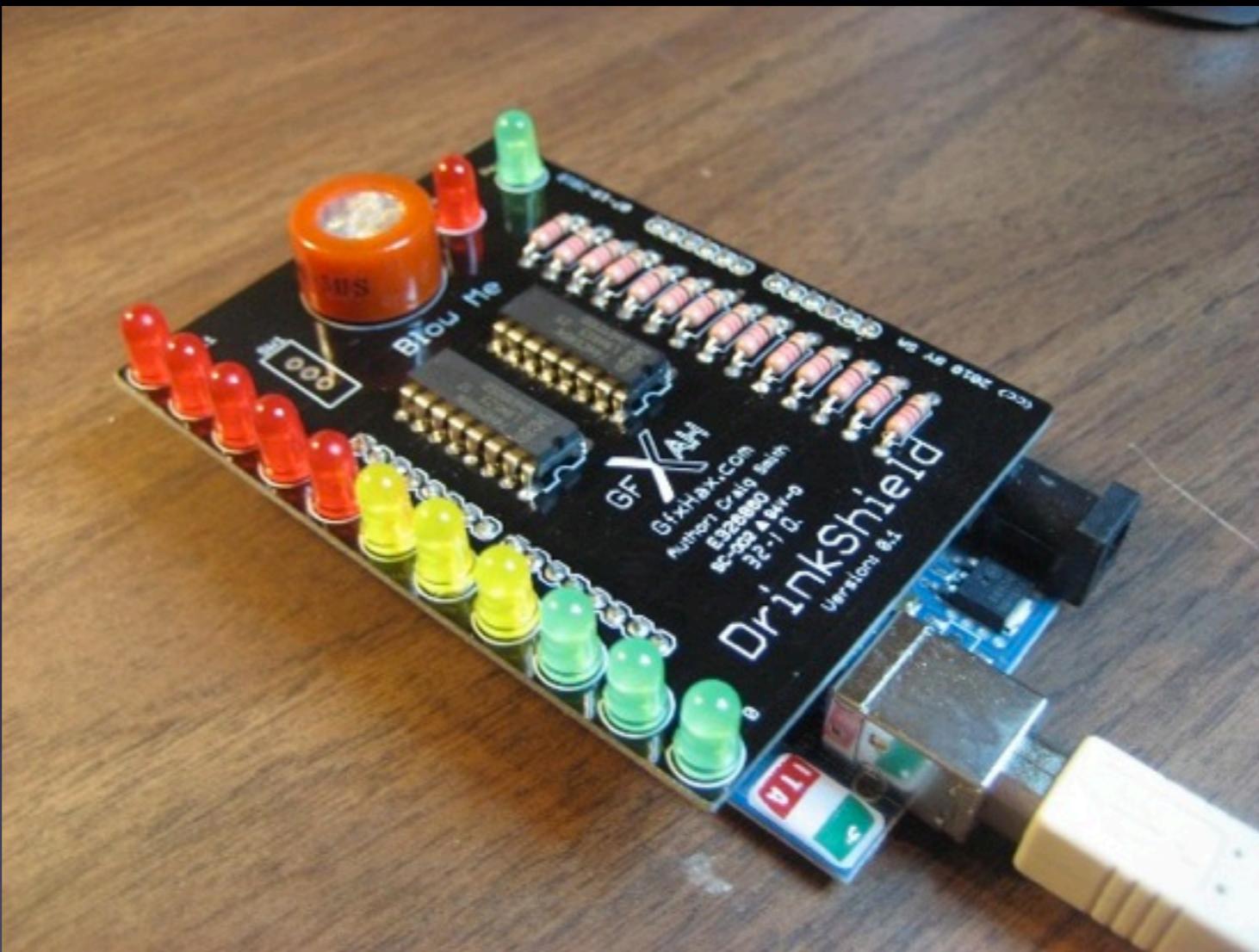
# WOW, Keys!





What will the CGI.pm  
of the Internet of  
Things be?





Arduino + Breathalyzer  
Shield + git commit  
hook = Better code