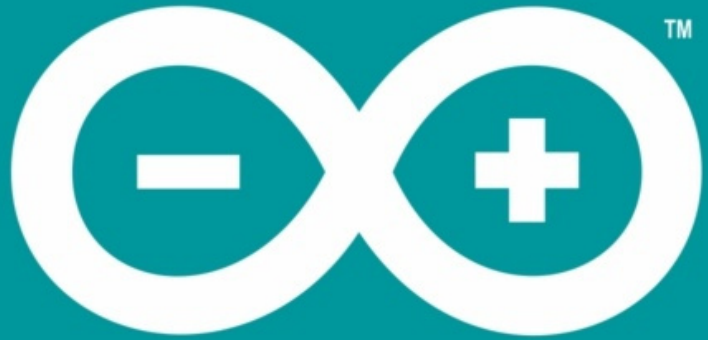
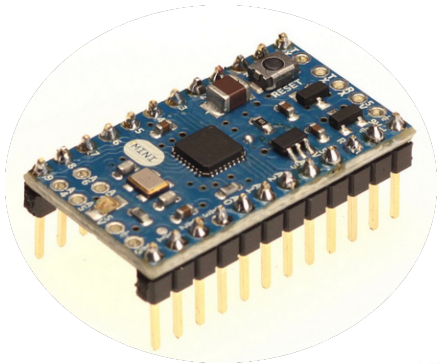

Powered by
Arduino
Python programming for Arduino
by Ramon Sorage
python
powered
print("Hello, world!")

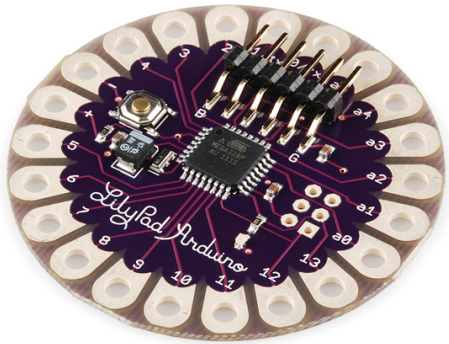
Agenda

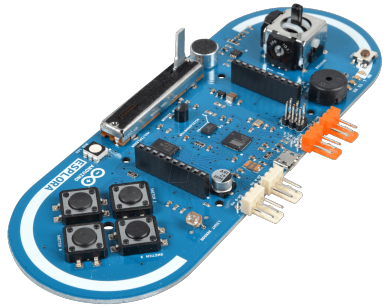


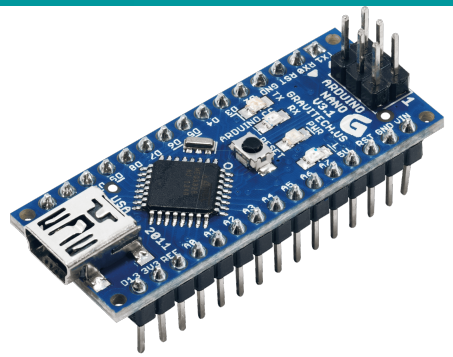
ARDUINO

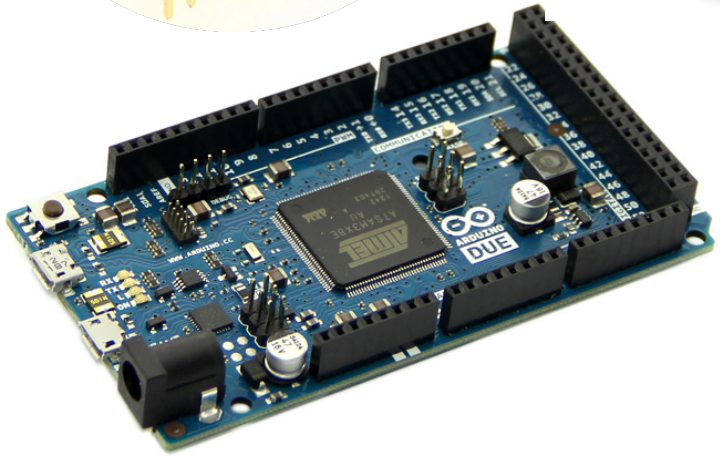
Family

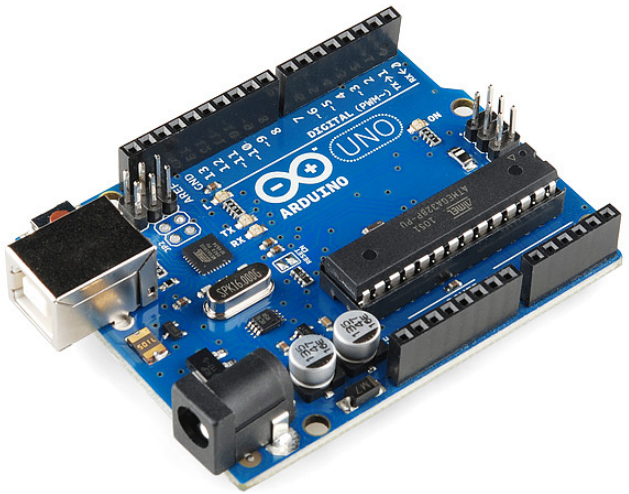












Arduino IDE

Download the Arduino Software

ARDUINO 1.6.4

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

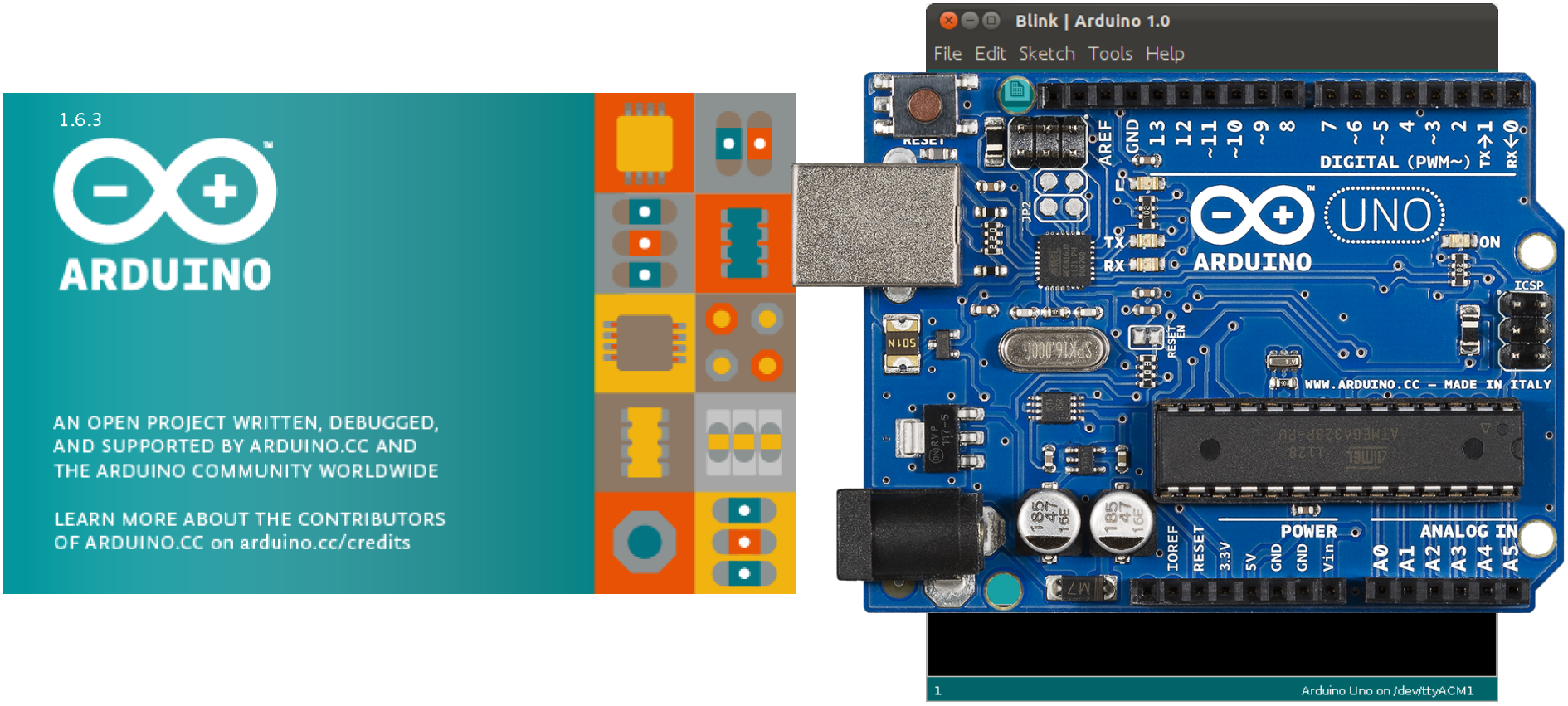
Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits

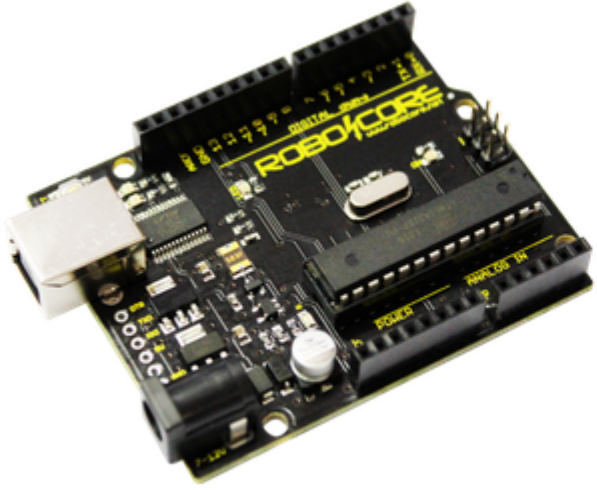
Release Notes
Source Code
Checksums

Source: http://www.arduino.cc/en/main/software

Arduino IDE


1.6.3
ARDUINO
AN OPEN PROJECT WRITTEN, DEBUGGED,
AND SUPPORTED BY ARDUINO.CC AND
THE ARDUINO COMMUNITY WORLDWIDE
LEARN MORE ABOUT THE CONTRIBUTORS
OF ARDUINO.CC on arduino.cc/credits
Blink | Arduino 1.0
File Edit Sketch Tools Help
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
WWW.ARDUINO.CC – MADE IN ITALY
1
Arduino Uno on /dev/ttyACM1

BlackBoard



BlackBoard V1.0

Quantidade: 1
(17 un. em estoque)

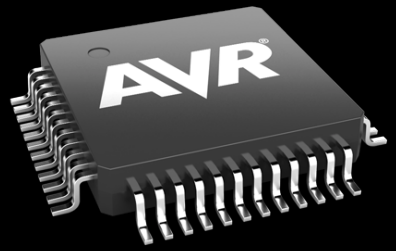
R$ 75,00 ADICIONAR AO CARRINHO

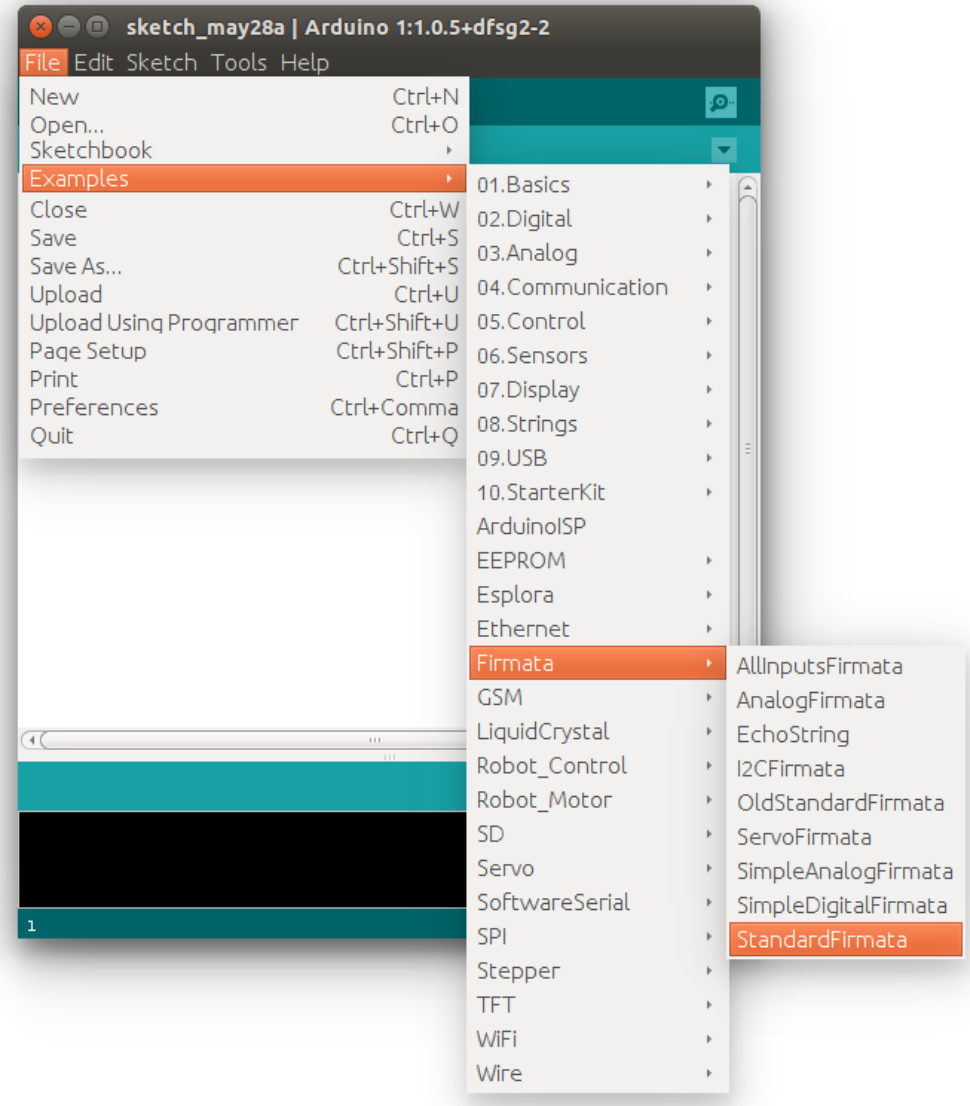
Parcele em até 12X!

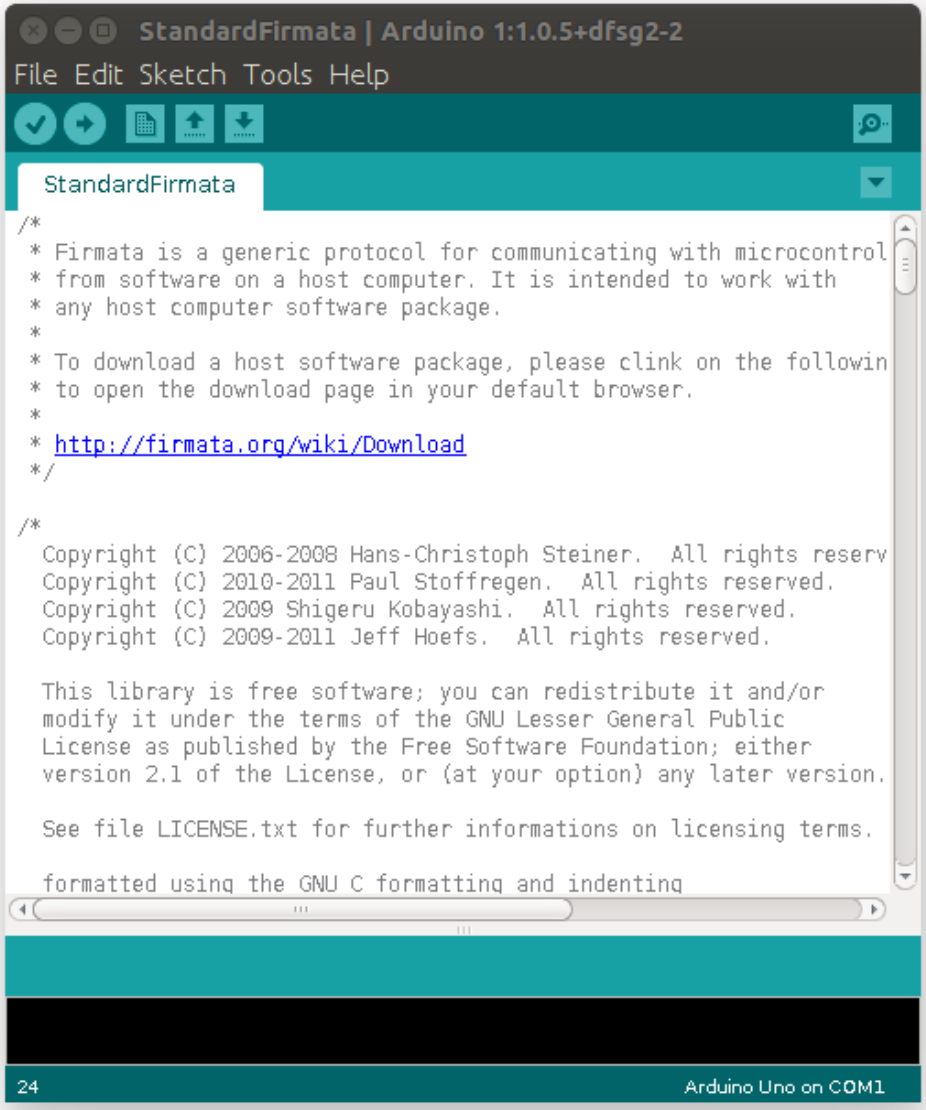
2x R$39,19	3x R$26,51	4x R$20,17	5x R$16,37
6x R$13,84	7x R$12,03	8x R$10,68	9x R$9,63
	10x R$8,79	11x R$8,10	12x R$7,53

Source: https://www.robocore.net/modules.php?name=GR_LojaVirtual&prod=530

Firmata Protocol




sketch_may28a | Arduino 1:1.0.5+dfsg2-2
File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
01.Basics
02.Digital
03.Analog
04.Communication
05.Control
06.Sensors
07.Display
08.Strings
09.USB
10.StarterKit
ArduinoISP
EEPROM
Esplora
Ethernet
Firmata
GSM
LiquidCrystal
Robot_Control
Robot_Motor
SD
Servo
SoftwareSerial
SPI
Stepper
TFT
WiFi
Wire
AllInputsFirmata
AnalogFirmata
EchoString
I2CFirmata
OldStandardFirmata
ServoFirmata
SimpleAnalogFirmata
SimpleDigitalFirmata
StandardFirmata
1


StandardFirmata | Arduino 1:1.0.5+dfsg2-2
File Edit Sketch Tools Help
StandardFirmata
/*
* Firmata is a generic protocol for communicating with microcontrol
* from software on a host computer. It is intended to work with
* any host computer software package.
*
* To download a host software package, please clink on the followin
* to open the download page in your default browser.
*
* http://firmata.org/wiki/Download
*/
/*
Copyright (C) 2006-2008 Hans-Christoph Steiner. All rights reserv
Copyright (C) 2010-2011 Paul Stoffregen. All rights reserved.
Copyright (C) 2009 Shigeru Kobayashi. All rights reserved.
Copyright (C) 2009-2011 Jeff Hoefs. All rights reserved.
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
See file LICENSE.txt for further informations on licensing terms.
formatted using the GNU C formatting and indenting
24
Arduino Uno on COM1

Debugging / Testing

For Linux (32 bit)

http://www.pjrc.com/teensy/firmata_test/firmata_test

For Linux (64 bit)

http://www.pjrc.com/teensy/firmata_test/firmata_test.64bit

http://www.pjrc.com/teensy/firmata_test/firmata_test.64bit_2 (Ubuntu 12.04)

http://www.pjrc.com/teensy/firmata_test/firmata_test.64bit_2_log (Ubuntu 12.04, with log window)

For Mac OS-X

http://www.pjrc.com/teensy/firmata_test/firmata_test.dmg

For Windows

http://www.pjrc.com/teensy/firmata_test/firmata_test.exe

Source code

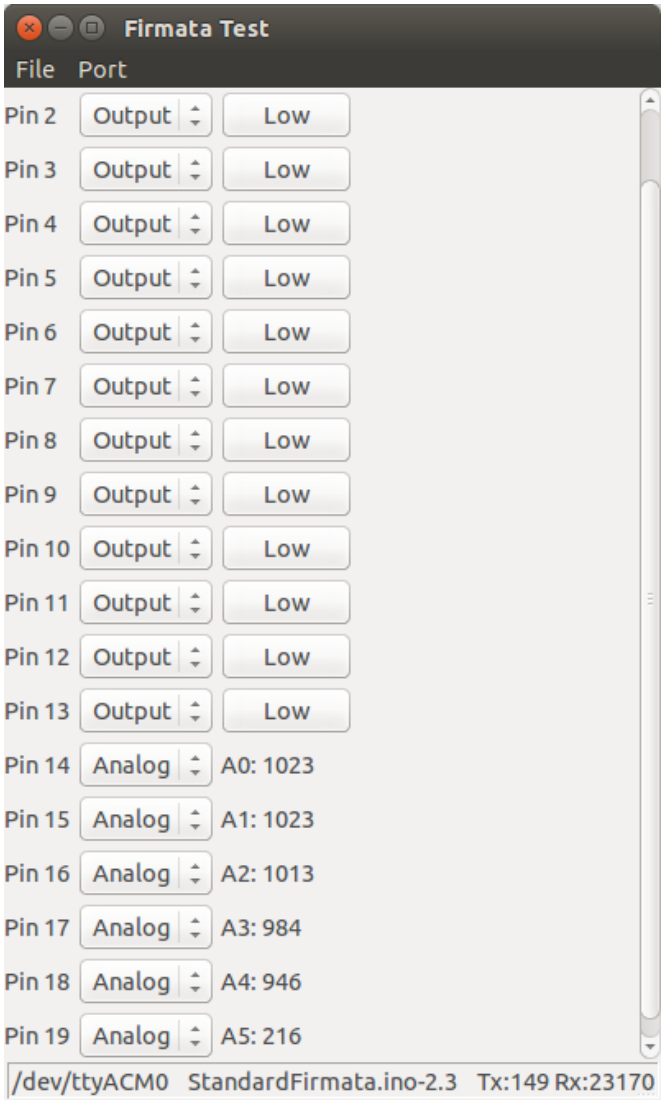
http://www.pjrc.com/teensy/firmata_test/firmata_test.tar.gz

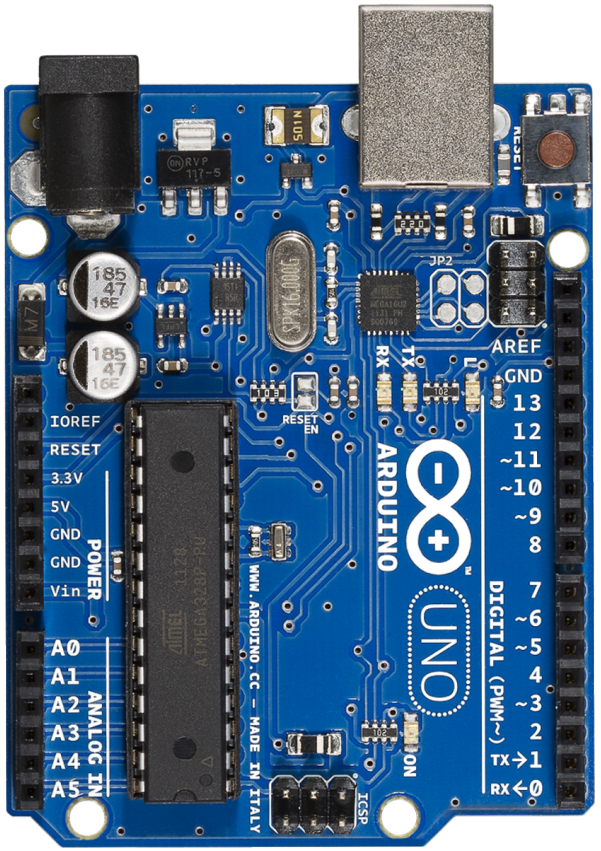
HOWTO Compile Firmata test program under Windows using GCC and WxWidgets

http://www.pjrc.com/teensy/firmata_test/firmata_test_OSL.tgz (unicode compatible, by Orion Lawlor)

Source: http://www.firmata.org/wiki/Main_Page#Firmata_Test_Program

Debugging / Testing





Client Libraries

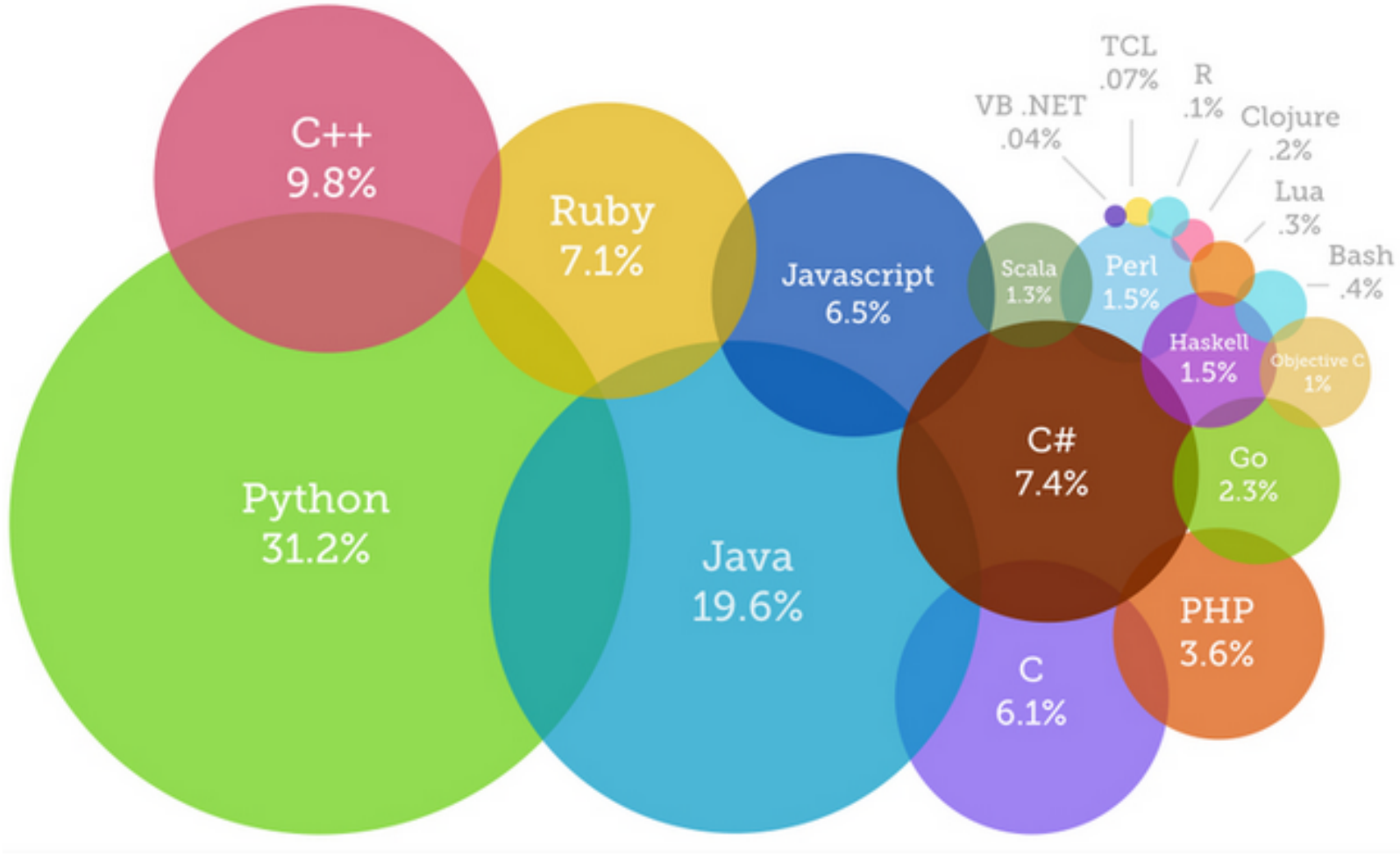
- procesing
  - [https://github.com/firmata/processing]
  - [http://funnel.cc]
- python
  - [https://github.com/firmata/pyduino]
  - [https://github.com/lupeke/python-firmata]
  - [https://github.com/tino/pyFirmata]
  - [https://github.com/MrYsLab/PyMata]
- perl
  - [https://github.com/ntruchsess/perl-firmata]
  - [https://github.com/rcaputo/rx-firmata]
- ruby
  - [https://github.com/hardbap/firmata]
  - [https://github.com/PlasticLizard/rufinol]
  - [http://funnel.cc]
- clojure
  - [https://github.com/nakkaya/clodiuno]
  - [https://github.com/peterschwarz/clj-firmata]
- javascript
  - [https://github.com/jgautier/firmata]
  - [http://breakoutjs.com]
  - [https://github.com/rwldrn/johnny-five]
- java
  - [https://github.com/4ntoine/Firmata]
  - [https://github.com/shigeodayo/Javarduino]
  - [https://github.com/kurbatov/firmata4j]
- .NET
  - [https://github.com/SolidSoils/Arduino]
  - [http://www.imagitronics.org/projects/firmatanet/]
- Flash/AS3
  - [http://funnel.cc]
  - [http://code.google.com/p/as3glue/]
- PHP
  - [https://bitbucket.org/ThomasWeinert/carica-firmata]
  - [https://github.com/oasynnoum/phpmake_firmata]
- Haskell
  - [http://hackage.haskell.org/package/hArduino]
- iOS
  - [https://github.com/jacobrosenthal/iosfirmata]
- Dart
  - [https://github.com/nfrancois/firmata]
- Max/MSP
  - [http://www.maxuino.org/]

Source: https://github.com/firmata/arduino

open source

python

Most Popular Coding Languages of 2015



python™ But why?

SQLAlchemy

Tornado

Pyramid™

Flask
web development,
one drop at a time

NumPy

django

Pylons™

pygame

matplotlib

Google app engine

python™

But why?

globo.com

TU Delft
Delft University of Technology

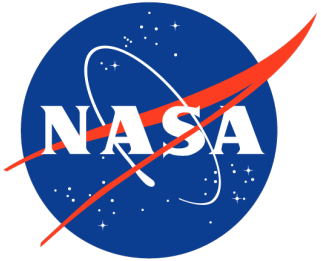
ABN·AMRO

Walt Disney
ANIMATION STUDIOS

NOKIA

Google

IBM



redhat

ALTIS
INVESTMENT MANAGEMENT

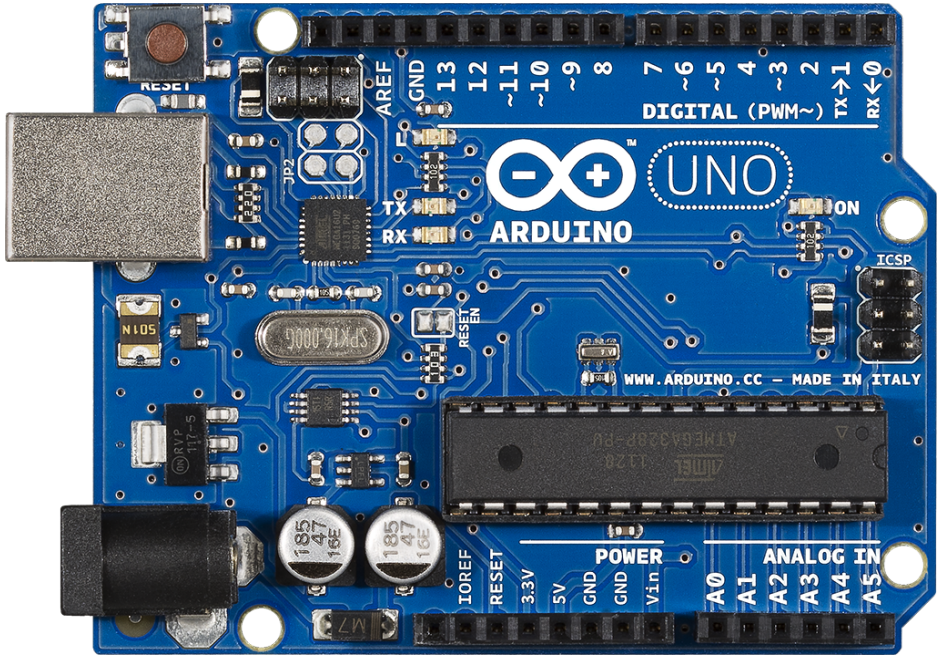

BATTLEFIELD
BAD COMPANY 2


SID MEIER'S
CIVILIZATION V

Putting it all together

pip install pyfirmata

```
from pyfirmata import Arduino, util

port = '/dev/ttyACM0'
board = Arduino(port)

d_12_out = board.get_pin('d:12:o')
d_11_pwm = board.get_pin('d:11:p')
d_7_in = board.get_pin('d:7:i')
a_1_in = board.get_pin('a:5:i')

it = util.Iterator(board)
it.start()
```




We can
do it
together

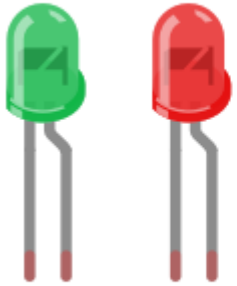
Our 1st project

Home Alarm System

Our 1st project

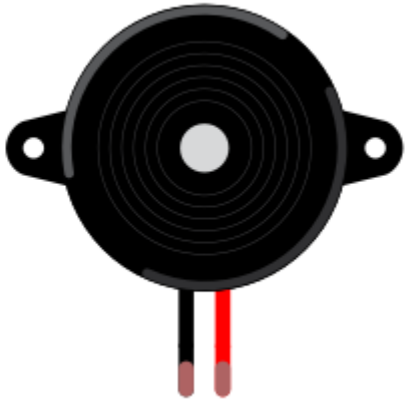
Components

LEDs



Push Button

Buzzer



PIR Sensor

Resistors

220 Ω

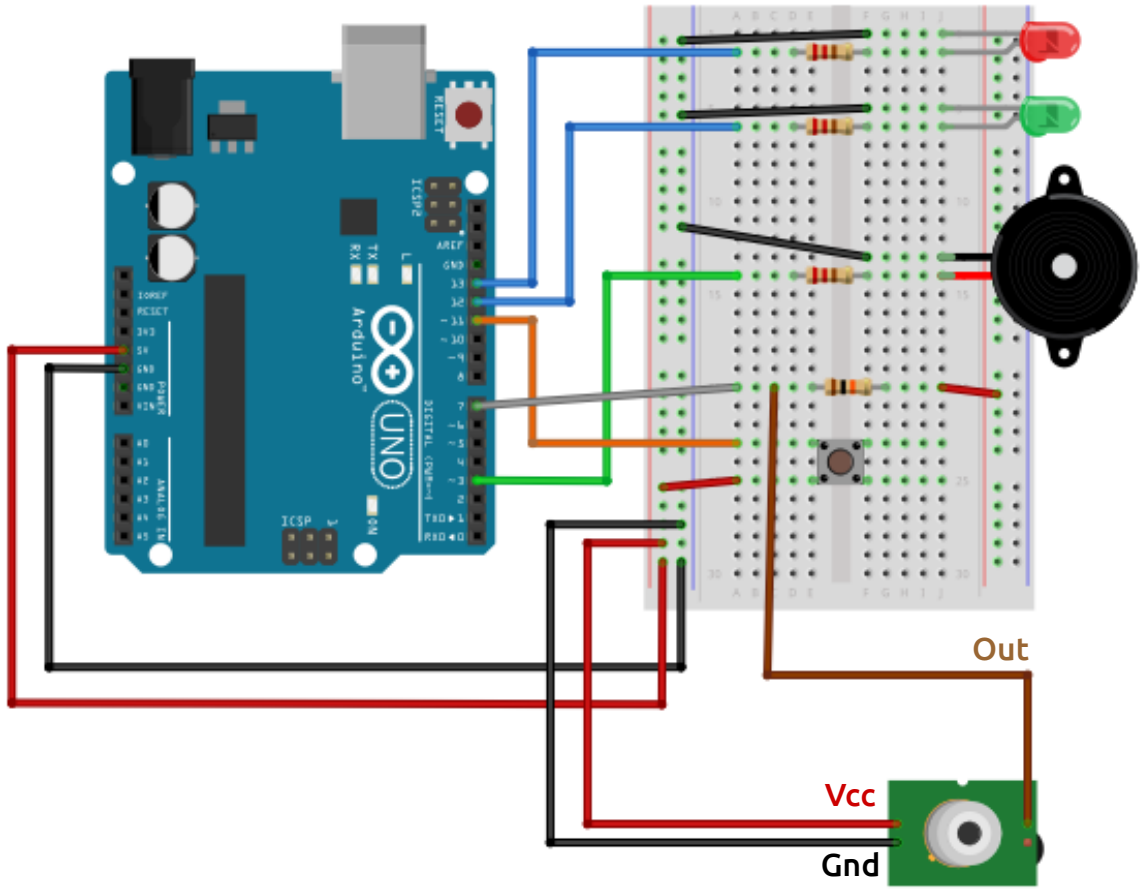
10 kΩ

Powered by Arduino

Our 1st project

Schematics

```
from pyfirmata import Arduino, util

port = '/dev/ttyACM0'
board = Arduino(port)

led_red = board.get_pin('d:13:o')
led_green = board.get_pin('d:12:o')
but_disarm = board.get_pin('d:11:i')
sensor_pir = board.get_pin('d:7:i')
act_beeper = board.get_pin('d:3:o')

it = util.Iterator(board)
it.start()

led_green.write(1)
print 'Alarm armed!'
```



Our 1st project

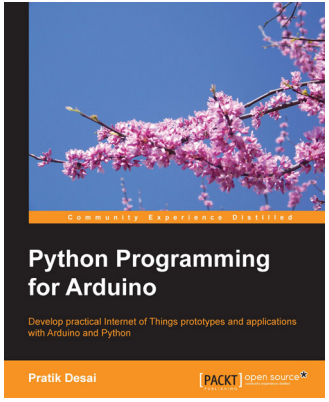
Logic

```
from time import sleep

try:
    while True:
        motion = sensor_pir.read()
        disarm = but_disarm.read()

        if motion is True:
            led_green.write(0)
            led_red.write(1)
            act_beeper.write(1)
            print 'Motion detected!'
            sleep(4)
            #mail_snd.send()
            #sleep(1)

        if disarm is True:
            led_green.write(1)
            led_red.write(0)
            act_beeper.write(0)
            print 'Alarm disarmed!'
            sleep(1)
```

```
except KeyboardInterrupt:
    print('Releasing resources ... '),
    board.exit()
    print('Bye!')
```

Our 2nd project

Real-time Plotting



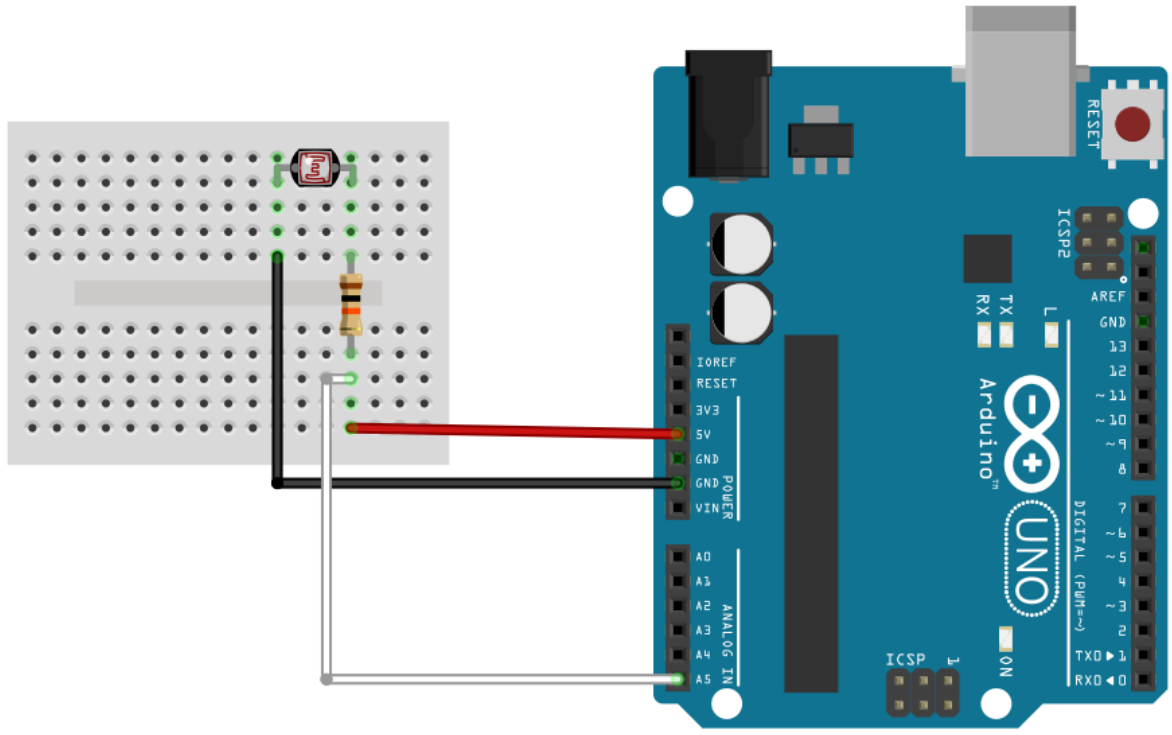
matplotlib

Powered by Arduino

Our 2nd project

Components + Schematics

```
from pyfirmata import Arduino, util

port = '/dev/ttyACM0'
board = Arduino(port)

ldr = board.get_pin('a:5:i')

it = util.Iterator(board)
it.start()
```



LDR

LDR

10 kΩ

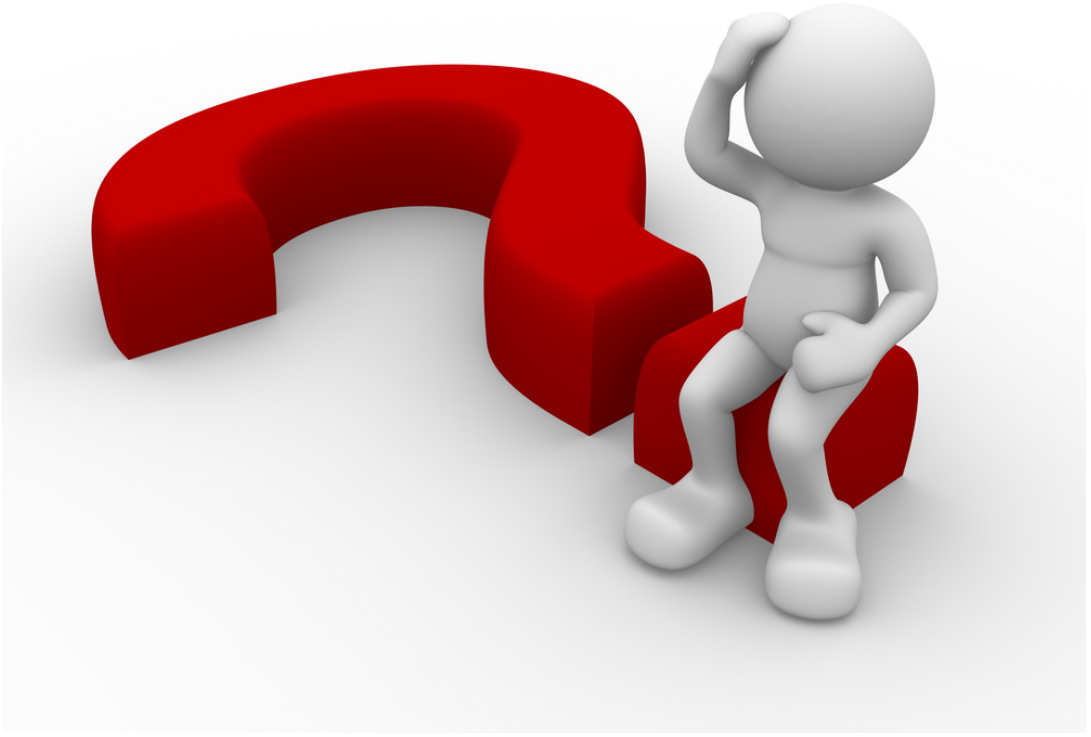
Powered by Arduino

Our 2nd project

Logic

```
from matplotlib import pyplot

pyplot.ion()
pData = [0] * 25
fig = pyplot.figure()
pyplot.title('Real-time LDR reading')
ax1 = pyplot.axes()
l1, = pyplot.plot(pData)
pyplot.ylim([0, 1])
```

```
from time import sleep

while True:
    try:
        sleep(1)
        pData.append(float(a0.read()))
        pyplot.ylim([0, 1])
        del pData[0]
        l1.set_xdata([i for i in xrange(25)])
        l1.set_ydata(pData)
        pyplot.draw()
    except:
        board.exit()
        break
```



Ramon Sorage *<rsorage@gmail.com>*

http://pt.slideshare.net/rsorage/python-programming-for-arduino