

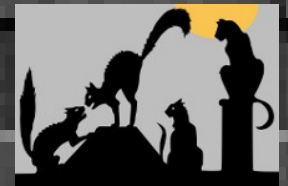
# Pyfirmata

## Cómo interactuar con arduino usando Python y Firmata

Hackmeeting – 2011

<http://lagunak.gisa-elkartea.org/projects/knrl>  
<http://lagunak.gisa-elkartea.org/svn/knrl/recetario/firmata>

**GISA Elkartea**  
Ales Zabala



## Puesta a punto

- Volcar en Arduino el sketch StandardFirmata.
- Instalar pyfirmata:

```
$ hg clone https://bitbucket.org/tino/pyfirmata
$ cd pyfirmata; python setup.py install
```

- Si necesitas servos:

```
$ svn export https://lagunak/svn/krn1/pyfirmata_servo/servo.patch .
$ patch -p0 < servo.patch
$ python setup.py install
```



# Hola mundo (blink)

- Este ejemplo hace parpadear el led enchufado al pin digital 13
- Además de *digital* también hay *analog*, y además de *write* también hay *read* :-)

```
import time
from pyfirmata import Arduino

board = Arduino('/dev/ttyUSB0')

while 1:
    board.digital[13].write(1)
    time.sleep(1)
    board.digital[13].write(0)
    time.sleep(1)
```



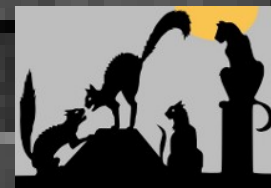
## Lecturas

- Para evitar que el Arduino nos esté mandando continuamente datos en los pines de lectura, se utiliza un Iterator:

```
it = util.Iterator(board)
it.start()
board.analog[0].enable_reporting()
board.analog[0].read()
```

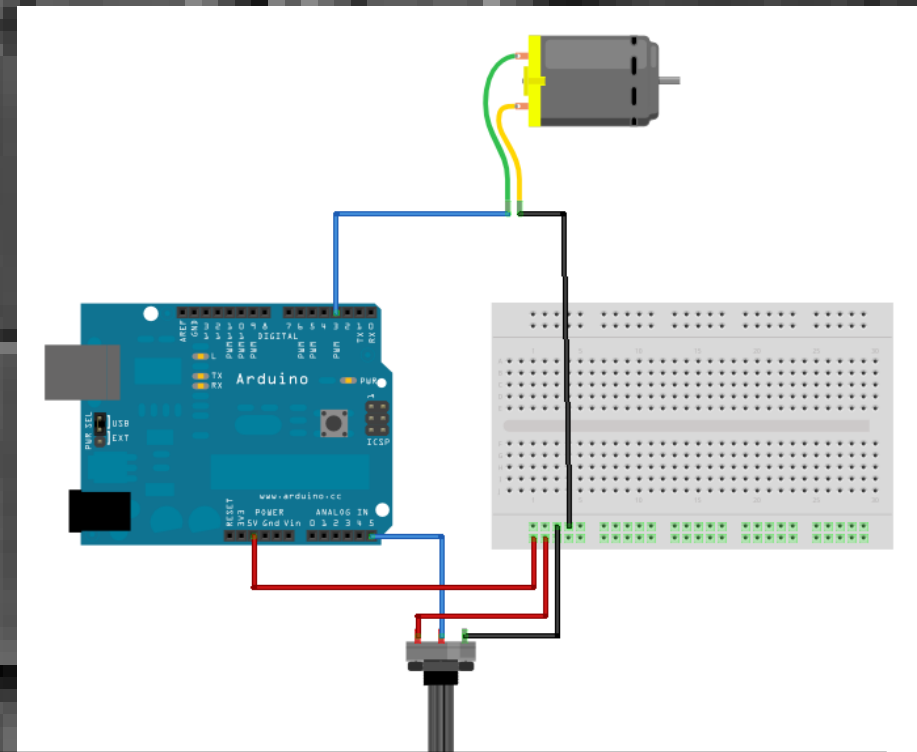
- El Iterator es un thread, por lo que hay que matarlo al salir. Lo más facil es romper la conexión con el puerto serie, el thread se suicida el solito.

```
board.exit()
```



## Ventilador (Digital)

- `pin.read() == 0..1`
- `pin.write([0,1])`

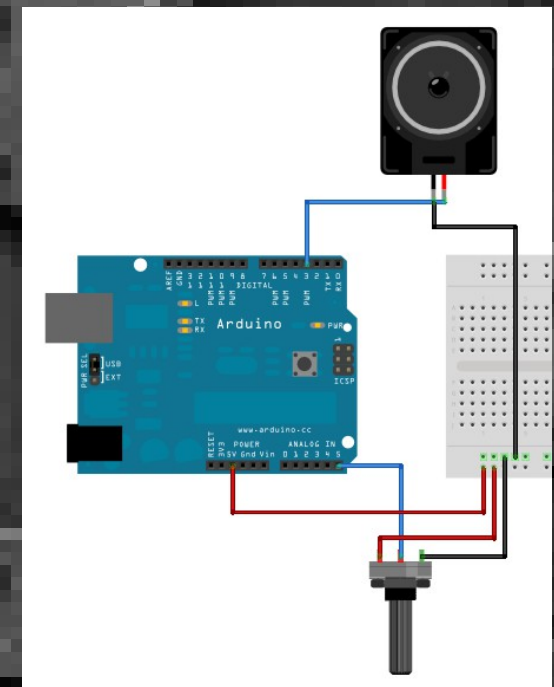


```
24 def start(self):
25     while self.loop:
26         newvalue = self.board.analog[self.pot_pin].read()
27         print newvalue
28         if self.value != newvalue:
29             if newvalue > 0.5:
30                 self.board.digital[self.motor_pin].write(1)
31             else:
32                 self.board.digital[self.motor_pin].write(0)
33         self.value = newvalue
34         time.sleep(0.5)
```



## PC Speaker (PWM)

- `pin.mode = PWM`
- `pin.write(0..1)`



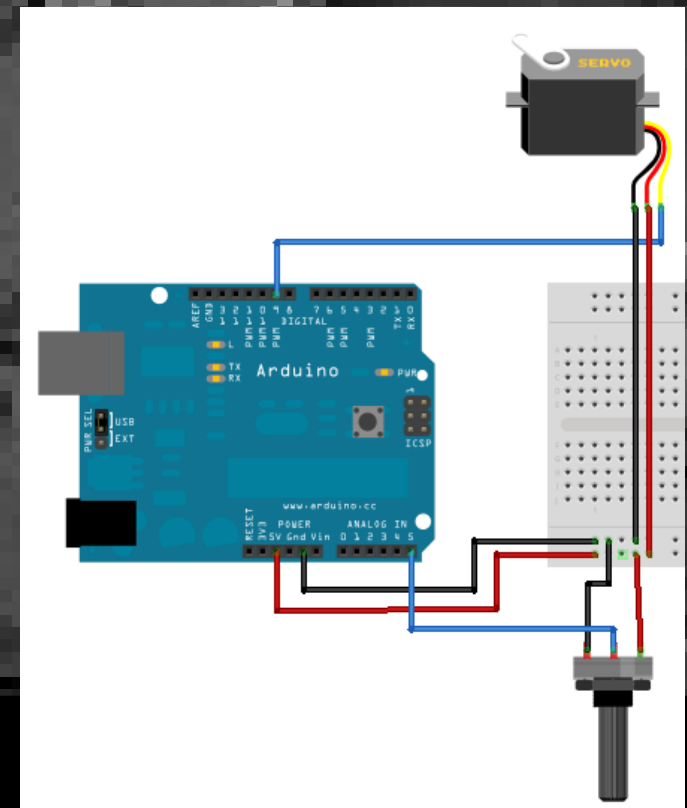
```
6     def __init__(self, port, pot_pin, motor_pin):
7         [...]
17    self.board.digital[self.motor_pin].mode = PWM
18    self.board.digital[self.motor_pin].write(0)
19    self.value = 0
20    [...]
26    def start(self):
27        while self.loop:
28            newvalue = self.board.analog[self.pot_pin].read()
29            print newvalue
30            if self.value != newvalue:
31                self.board.digital[self.motor_pin].write(newvalue)
32                self.value = newvalue
33            time.sleep(0.5)
```



# Servos

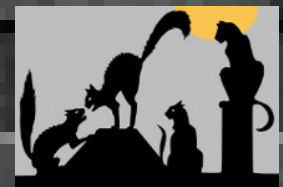
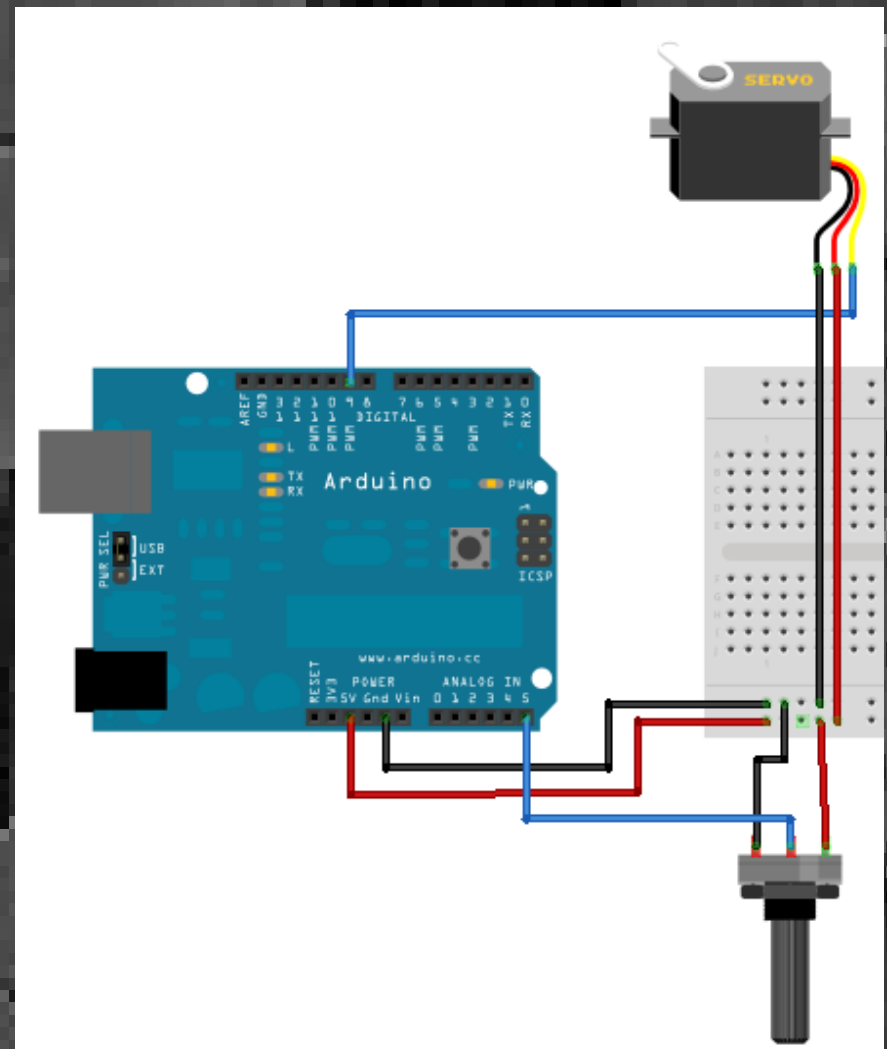
- `pin.mode = SERVO`
- `pin.write(angulo)`

```
6     def __init__(self, port,
                    pot_pin, motor_pin):
            [...]
17    self.board.digital[self.motor_pin].mode = SERVO
            [...]
26    def start(self):
27        while self.loop:
28            newvalue = self.board.analog[self.pot_pin].read()
29            if (newvalue is not None) and (self.value != newvalue):
30                angle = int(180 * newvalue)
31                print"%s => %s" % (newvalue, angle)
32                self.board.digital[self.motor_pin].write(angle)
33                self.value = newvalue
34                time.sleep(0.5)
```



## Otro ejemplo

- Representacion del estado del Arduino con una GUI en PyQt
- De igual manera podríamos usar sockets, la carga del ordenador, consultas http...





## Comentarios

- Es muy fácil de usar.
- Con la consola de python podemos hacer pruebas directamente en Arduino.
- También existe pyduino, pero parece que es un proyecto abandonado.
- En pyfirmata hay un desarrollador (un tanto esporádico), y una persona que envía parches (además del mío :-).
- Tiene sus limitaciones (DHT11).
- Todavía le faltan cosas para implementar completamente firmata, aunque lo más típico ya está.

