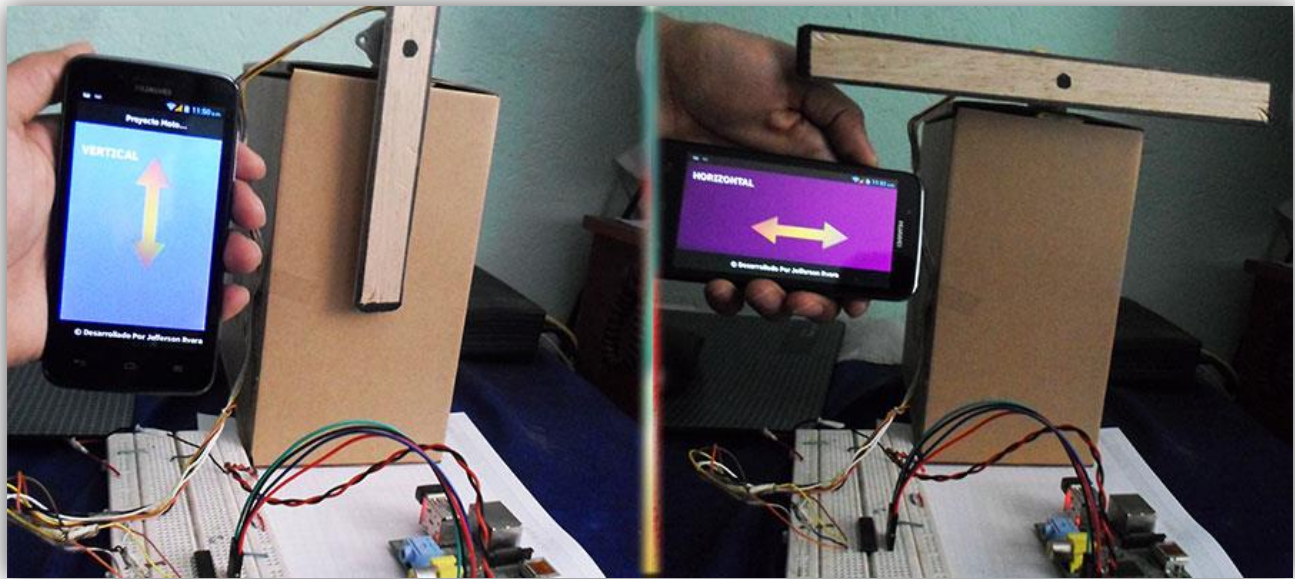


# CONTROL DE UN MOTOR A PASO, USANDO EL ACELERÓMETRO DE UN CELULAR, CON RASPBERRY PI

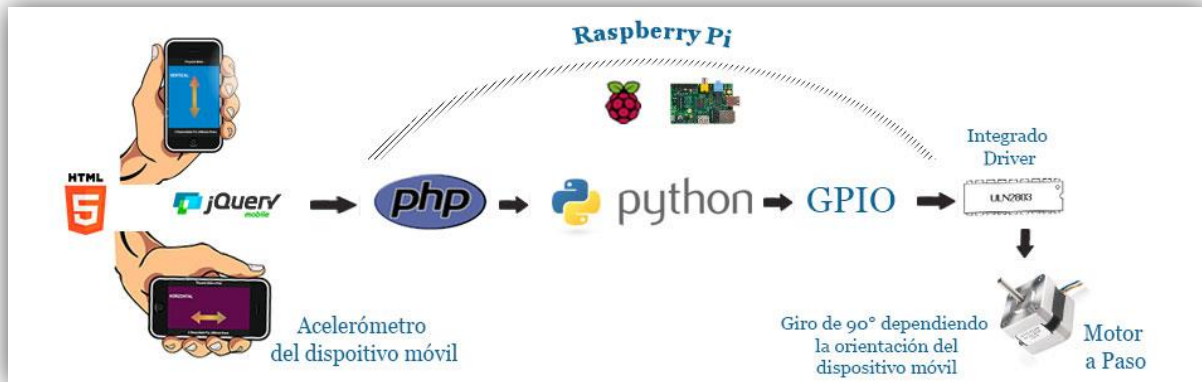


*Desarrollado Por: Jefferson Rivera Patiño  
jefersonpatino@yahoo.es*

## Contenido

DESCRIPCIÓN DEL PROYECTO .....	2
HERRAMIENTAS USADAS .....	2
CONOCIMIENTOS REQUERIDOS (BÁSICOS) .....	3
EXPLICACIÓN DEL SOFTWARE .....	3
EXPLICACIÓN DEL HARDWARE .....	7
PRUEBA DE FUNCIONAMIENTO .....	8
BIBLIOGRAFÍA .....	8
ANEXOS .....	8

## DESCRIPCIÓN DEL PROYECTO



La idea del proyecto es controlar el movimiento de un motor a paso, por medio del giro de un dispositivo móvil.

Entonces el proceso es el siguiente: por medio de Javascript (jQuery Mobile) y Html5 se detecta el cambio de giro del celular, y se envía este estado a un script PHP, el cual se encarga de ejecutar una sentencia Python para enviar una señal de pulsos que pone en marcha el motor a paso. El sentido de giro del motor depende de la orientación del dispositivo móvil.

Por ejemplo si se toma el celular de forma vertical y luego se gira 90° *sin importar el sentido*, este cambio lo detecta el navegador por medio de javascript, y envía por Ajax el estado de la orientación (Horizontal) a PHP, el cual se encarga de ejecutar un programa Python que cumple la función de girar el motor a paso 90°

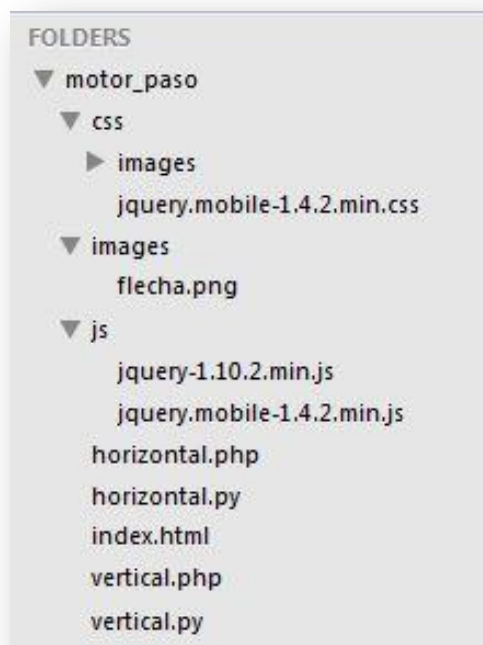
## HERRAMIENTAS USADAS

- Raspberry Pi Modelo B
- Protoboard
- Integrado driver. ULN2803
- Motor a Paso Unipolar
- Conector cable jumper
- IDE Sublime Text

## CONOCIMIENTOS REQUERIDOS (BÁSICOS)

- HTML , CSS, Javascript
- PHP
- Python
- Manejo de comandos de consola en Linux
- Electrónica básica

## EXPLICACIÓN DEL SOFTWARE



- **Css:** Hoja de estilos de jQuery-mobile
- **Js:** Librería jQuery y JQuery-mobile
- **Images:** imagen de la flecha
- **Index.html:** vista de la aplicación
- **Vertical.php:** ejecuta a vertical.py:
- **vertical.py:** envía las señales al GPIO, para mover el motor en mismo sentido de las manecillas del reloj
- **horizontal.php:** ejecuta horizontal.py
- **horizontal.py:** envía las señales al GPIO, para mover el motor en sentido contrario de las manecillas del reloj

El código fuente se puede ver detalladamente en [jeffersonrivera.com/pi/motor\\_paso.zip](http://jeffersonrivera.com/pi/motor_paso.zip)

Quiero destacar el fragmento de código que se encarga de reconocer el cambio del celular

```

<script>
$(document).on("pagecreate",function(event){
    $(window).on("orientationchange",function(event){
        var ori = event.orientation;
        if(event.orientation=='portrait'){
            $("#cambio").text('VERTICAL');
            $("#flecha").addClass('flecha_v');
            ("body").addClass('fondo1');
            ("body").removeClass('fondo2');
            // Envio del Ajax
            $.ajax({
                type: "POST",
                url: "vertical.php",
            })
        }
        else{
            $("#cambio").text('HORIZONTAL');
            $("#flecha").removeClass('flecha_v');
            $("#flecha").html('');
            ("body").addClass('fondo2');
            ("body").removeClass('fondo1');
            // Envio del Ajax
            $.ajax({
                type: "POST",
                url: "horizontal.php",
            })
        }
    });
});
</script>

```

Parte de index.html

La función *orientationchange*, es propia de jQuery-mobile, y nos retorna 'portrait', cuando el dispositivo este en posición vertical y 'landscape' cuando este en horizontal. Ya teniendo los estados, lo que se hace es ejecutar por Ajax el PHP, y trabajar un poco con estilos.

Vertical.php

```
<?php
exec('sudo python /var/www/pi/motor_paso/vertical.py');
?>
```

[Vertical.py](#)

Horizontal.php

```
<?php
exec('sudo python /var/www/pi/motor_paso/horizontal.py');
?>
```

[Horizontal.php1](#)

Vertical.py

```
1  import RPi.GPIO as GPIO
2  import time
3
4  # variables
5  delay = 85
6  steps = 4
7
8  # Pines de Salida
9  salida_a = 12
10 salida_b = 16
11 salida_c = 18
12 salida_d = 22
13
14 # Setup Pines de Salida
15 GPIO.setup(salida_a, GPIO.OUT)
16 GPIO.setup(salida_b, GPIO.OUT)
17 GPIO.setup(salida_c, GPIO.OUT)
18 GPIO.setup(salida_d, GPIO.OUT)
19
20 # Funcion atras()
21 def atras(delay, steps):
22     for i in range(0, steps):
23         setPaso(0, 0, 0, 1)
24         time.sleep(delay)
25         setPaso(0, 0, 1, 0)
26         time.sleep(delay)
27         setPaso(0, 1, 0, 0)
28         time.sleep(delay)
29         setPaso(1, 0, 0, 0)
30         time.sleep(delay)
```

[Vertical.py. Parte 1](#)

```

31
32 # Funcion setPaso()
33 def setPaso(a1, a2, a3, a4):
34     GPIO.output(salida_a, a1)
35     GPIO.output(salida_b, a2)
36     GPIO.output(salida_c, a3)
37     GPIO.output(salida_d, a4)
38
39     count = 0
40     while (count < 1):
41         atras(int(delay) / 1000.0, int(steps))
42         print 'Cuenta: ', count
43         count = count + 1
44
45     setPaso(0, 0, 0, 0)
46

```

Vertical.py. Parte 2

Horizontal.py

```

1  import RPi.GPIO as GPIO
2  import time
3
4  # variables
5  delay = 85
6  steps = 4
7
8  # Pines de Salida
9  salida_a = 12
10 salida_b = 16
11 salida_c = 18
12 salida_d = 22
13
14 # Setup Pines de Salida
15 GPIO.setup(salida_a, GPIO.OUT)
16 GPIO.setup(salida_b, GPIO.OUT)
17 GPIO.setup(salida_c, GPIO.OUT)
18 GPIO.setup(salida_d, GPIO.OUT)
19
20 # Funcion adelante()
21 def adelante(delay, steps):
22     for i in range(0, steps):
23         setPaso(1, 0, 0, 0)
24         time.sleep(delay)
25         setPaso(0, 1, 0, 0)
26         time.sleep(delay)
27         setPaso(0, 0, 1, 0)
28         time.sleep(delay)
29         setPaso(0, 0, 0, 1)
30         time.sleep(delay)

```

Horizontal.py Parte 1



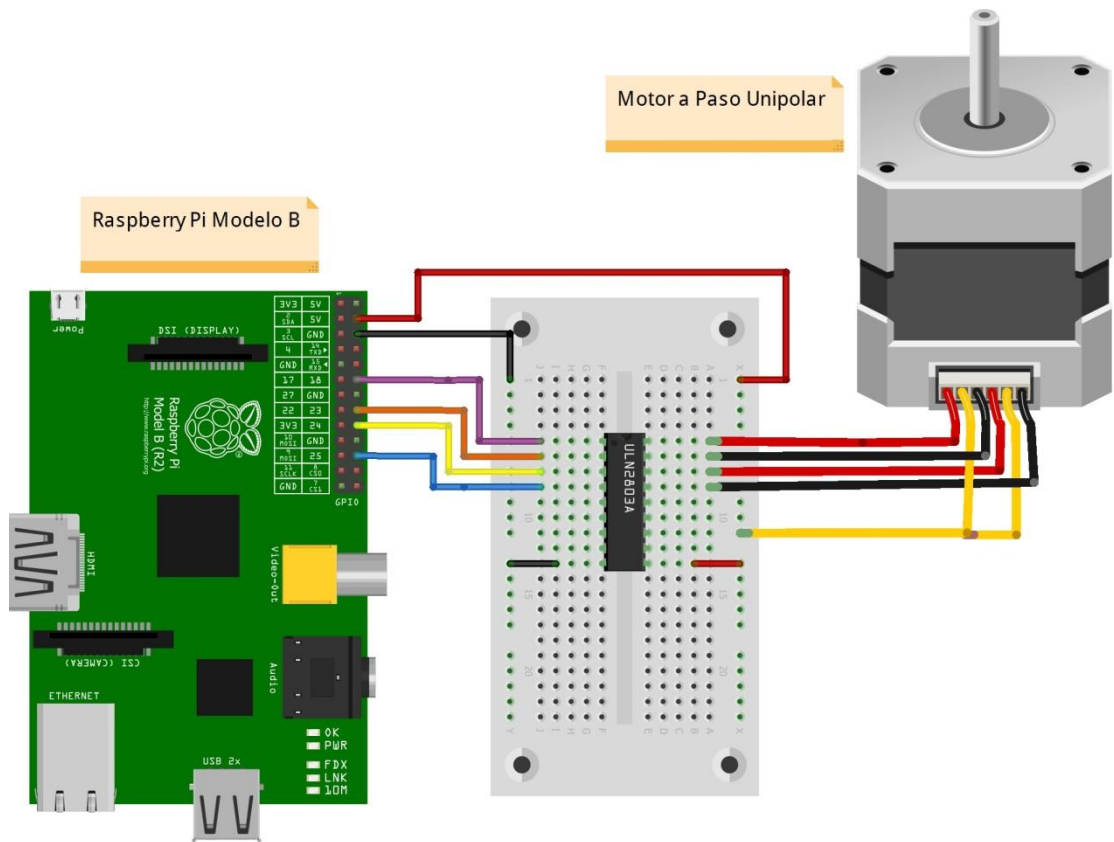
```

31
32 # Funcion setPaso()
33 def setPaso(a1, a2, a3, a4):
34     GPIO.output(salida_a, a1)
35     GPIO.output(salida_b, a2)
36     GPIO.output(salida_c, a3)
37     GPIO.output(salida_d, a4)
38
39 count = 0
40 while (count < 1):
41     adelante(int(delay) / 1000.0, int(steps))
42     print 'Cuenta: ', count
43     count = count + 1
44
45 setPaso(0, 0, 0, 0)
46

```

Horizontal.py Parte 2

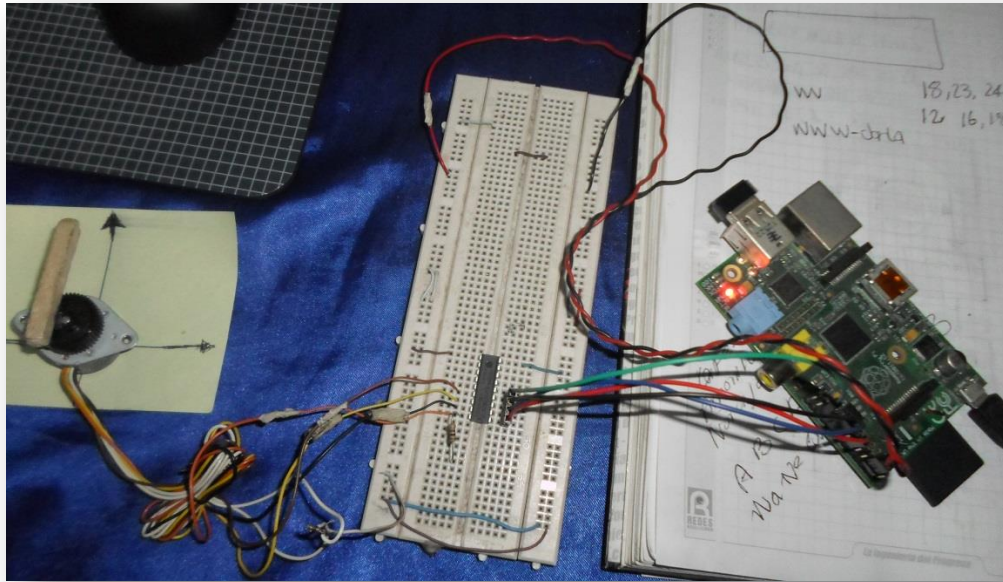
## EXPLICACIÓN DEL HARDWARE



fritzing

Diagrama Esquemático

Del GPIO de la Raspberry Pi, salen 6 cables, 4 van a las entradas del integrado ULN2803, y las dos cables restantes son de polarización; en las salidas del integrado se conecta las bobinas del motor a paso, el común del motor, va a positivo.



## PRUEBA DE FUNCIONAMIENTO

Ver Video

## BIBLIOGRAFÍA

- <https://www.youtube.com/watch?v=KZVgKu6v808>
- <http://www.w3schools.com/jquerymobile/default.asp>
- <http://www.docentes.unal.edu.co/hfvelascop/docs/CLASES/DIGITALES2/LABORATORIO/Motor%20Paso%20a%20Paso.pdf>
- <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-10-stepper-motors>
- <http://www.w3.org/TR/orientation-event/>
- 

## ANEXOS

- [http://jeffersonrivera.com/pi/motor\\_paso.zip](http://jeffersonrivera.com/pi/motor_paso.zip)



# ULN2803

