CURSO DE RASPBERRY PI

Semana 2

Prof. Edwin Yllanes Cucho
e.yllanescucho@gmail.com

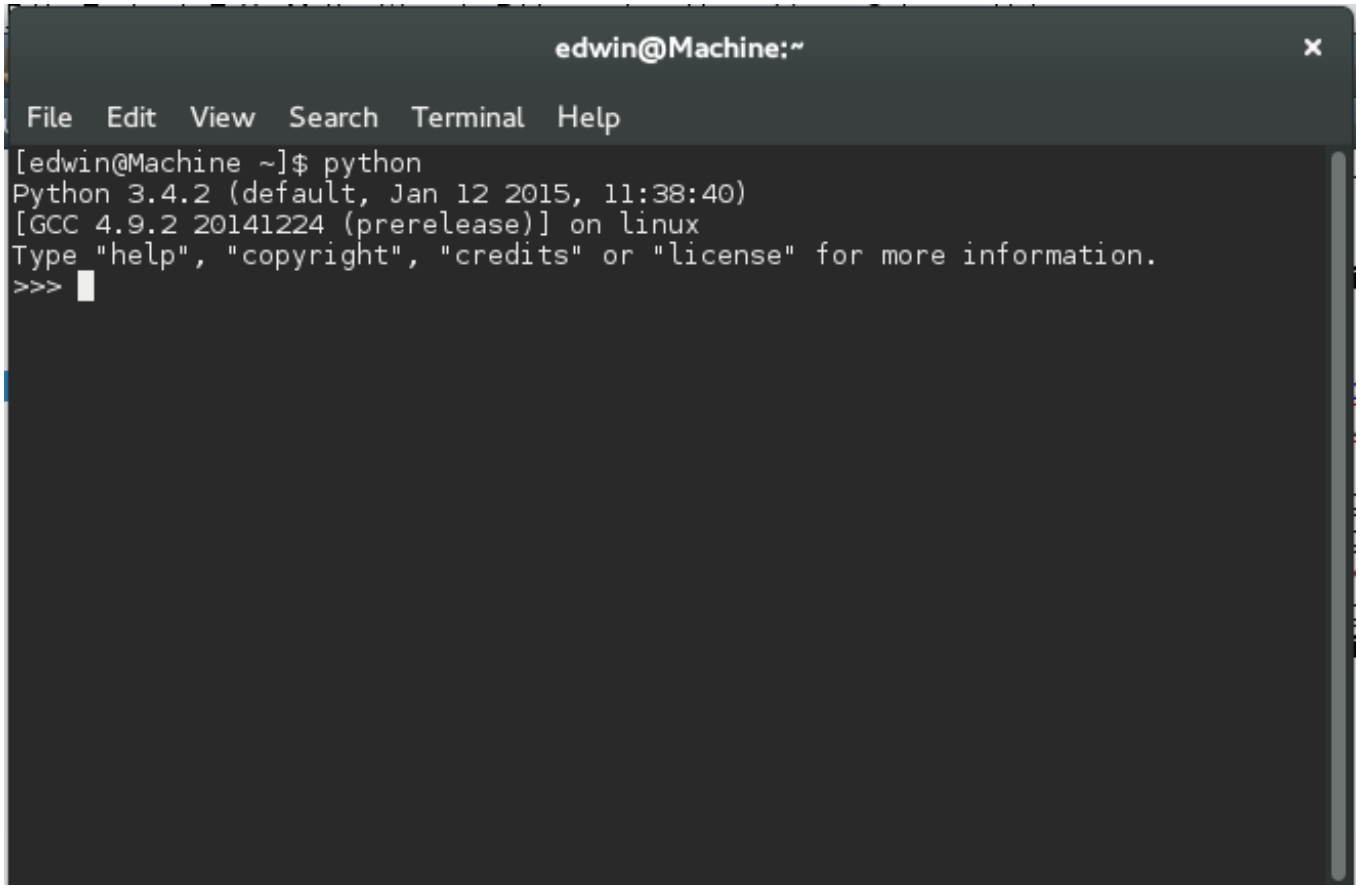
4 de mayo de 2015

SEMANA 2

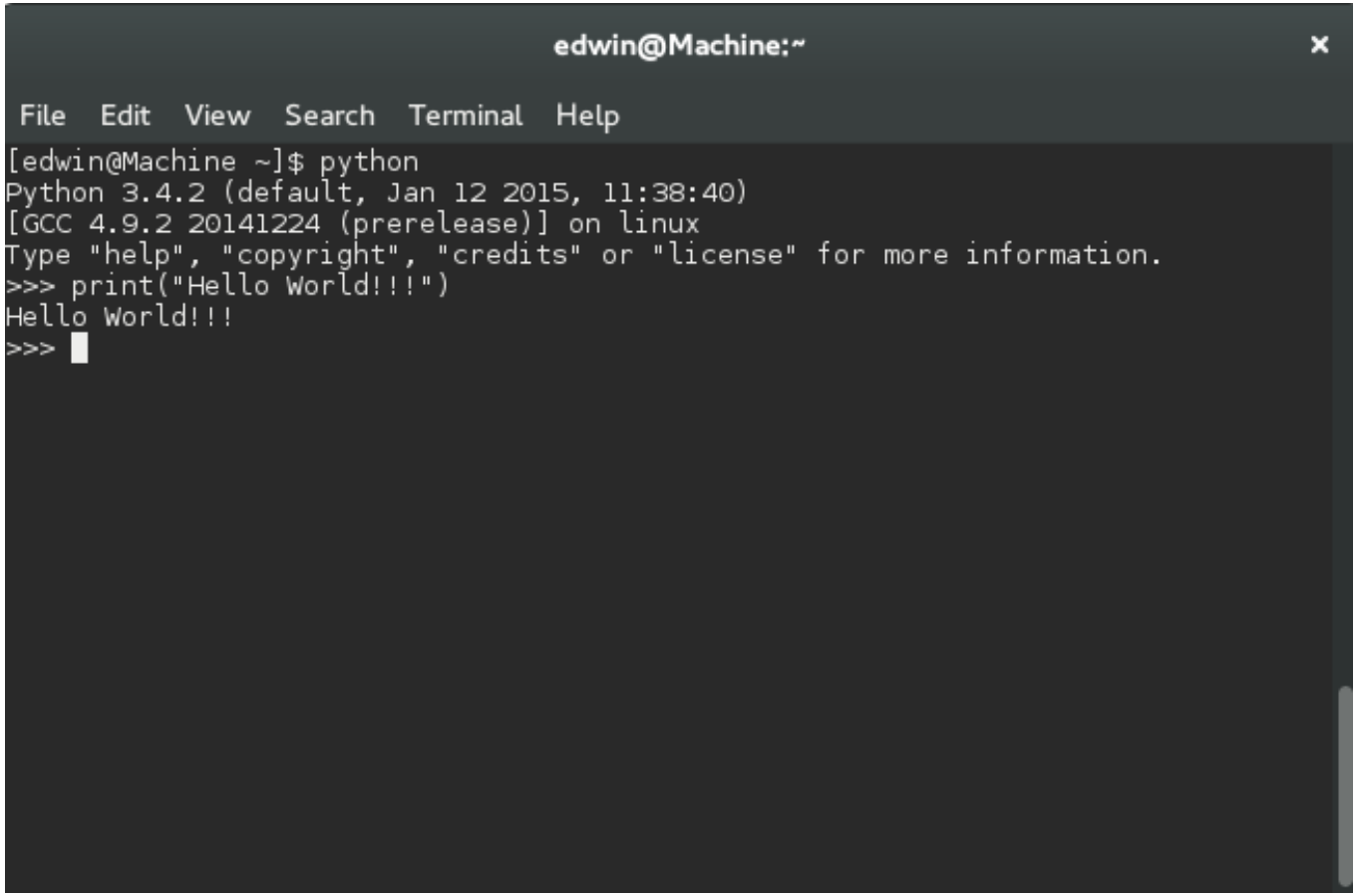
Instalación y uso de Python

$ sudo apt-get install python

$ python


edwin@Machine:~
File Edit View Search Terminal Help
[edwin@Machine ~]$ python
Python 3.4.2 (default, Jan 12 2015, 11:38:40)
[GCC 4.9.2 20141224 (prerelease)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>

$ print(”Hello World!!!”)



Uso de GPIO y Python

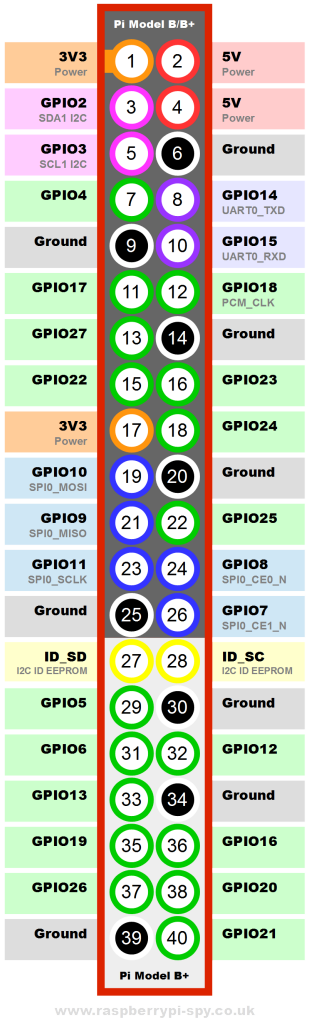


Figura : Pinout B+

Instalación

- GPIO
  - $ sudo apt-get install python-dev
  - $ sudo apt-get install python-rpi.gpio
- I^2C [opcional]
  - $ sudo apt-get install python-smbus
- SPI [opcional]
  - $ git clone git://github.com/doceme/py-spidev
  - $ cd py-spidev/
  - $ sudo python setup.py install
- UART [opcional]
  - $ sudo apt-get install python-serial

Código

Descargar el código fuente desde la siguiente página:
https://github.com/eyllanesc/Raspberry-Course

LED

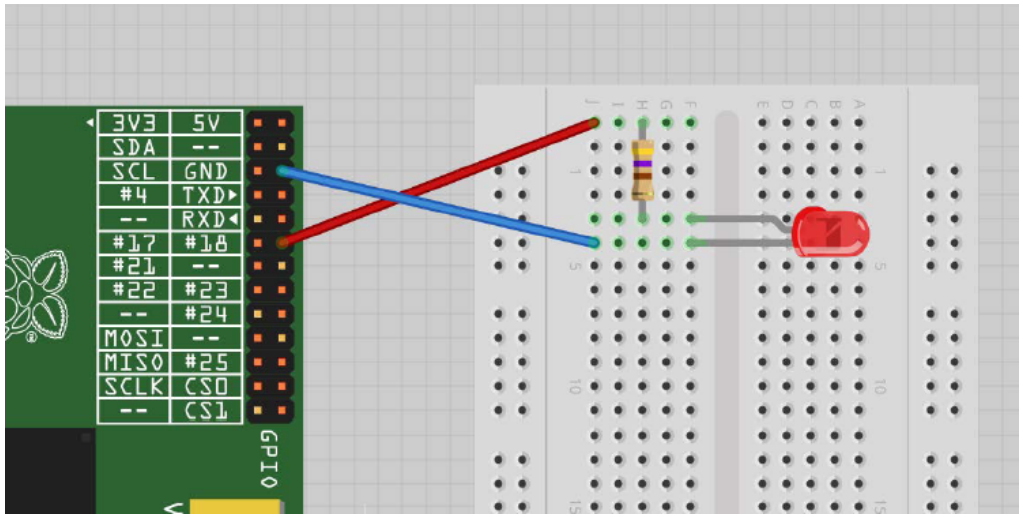


Figura : Circuito para el encendido de Led

```
import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)

state = True

while True:
    GPIO.output(18,True)
    time.sleep(1)
    GPIO.output(18,False)
    time.sleep(1)
```

```
import RPi.GPIO as GPIO
led_pin = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)
pwm_led = GPIO.PWM(led_pin, 500)
pwm_led.start(100)
while True:
    duty_s = raw_input("Enter Brightness (0 to 100):")
    duty = int(duty_s)
    pwm_led.ChangeDutyCycle(duty)
```

PULSADORES

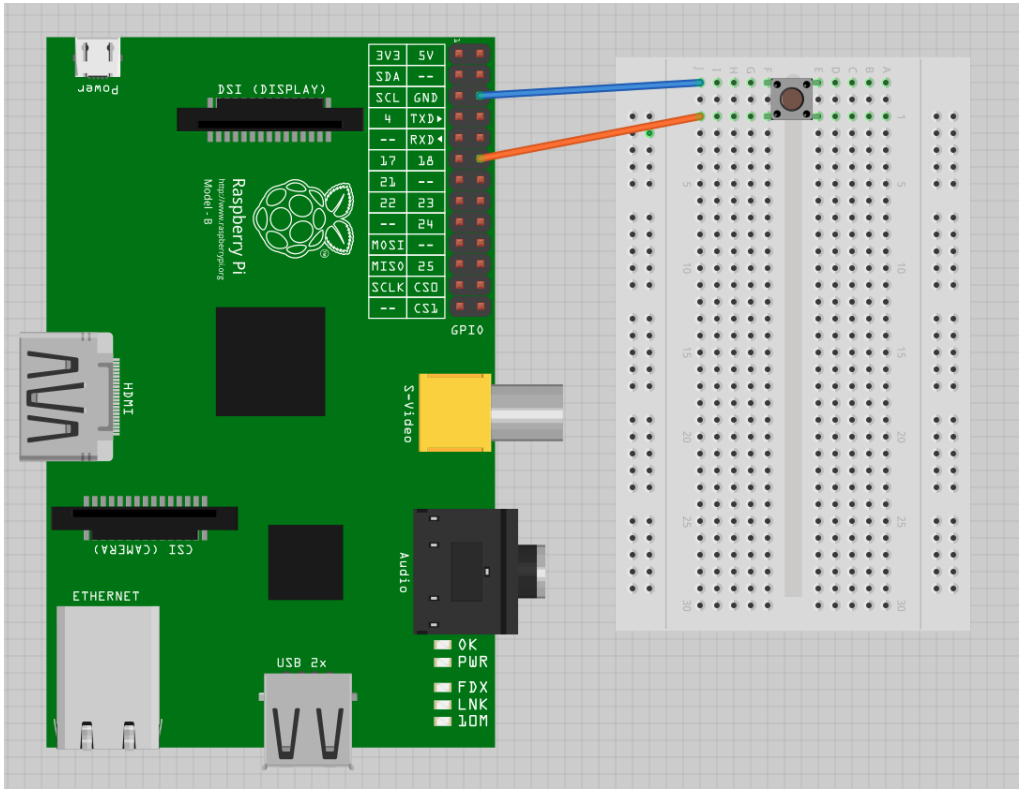


Figura : Circuito para el uso de pulsadores

```
import RPi.GPIO as GPIO #import GPIO library
import time #import time for managed time

GPIO.setmode(GPIO.BCM)

GPIO.setup(18, GPIO.IN, pull_up_down=GPIO.PUD_UP)#GPIO 18 como entrada

while True:
    input_state = GPIO.input(18) #leemos la entrada
    if input_state == False:  #si esta en nivel bajo
        print('Button Pressed') #imprime esto
    time.sleep(0.2) #tiempo muerto
```

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
switch_pin = 18
led_pin = 23
GPIO.setup(switch_pin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(led_pin, GPIO.OUT)
led_state = False
old_input_state = True # pulled-up
while True:
    new_input_state = GPIO.input(switch_pin)
    if new_input_state == False and old_input_state == True:
        led_state = not led_state
    old_input_state = new_input_state
    GPIO.output(led_pin, led_state)
```

ULTRASONIDO

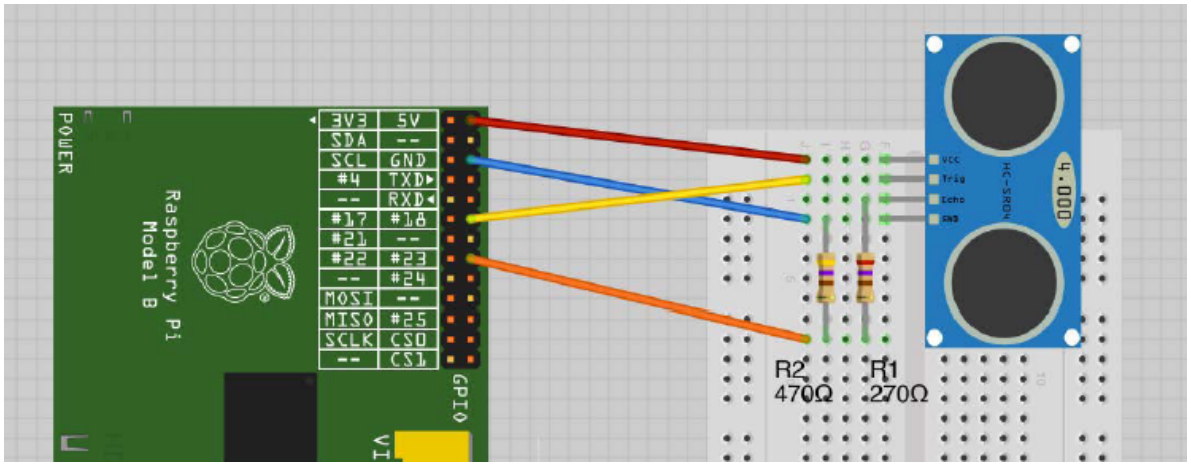


Figura : Conexión

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
TRIG = 23
ECHO = 24
print "Distance Measurement In Progress"
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)
GPIO.output(TRIG, False)
print "Waiting For Sensor To Settle"
time.sleep(2)
GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)
while GPIO.input(ECHO)==0:
    pulse_start = time.time()
while GPIO.input(ECHO)==1:
    pulse_end = time.time()
pulse_duration = pulse_end - pulse_start
distance = pulse_duration * 17150
distance = round(distance, 2)
print "Distance:",distance,"cm"
GPIO.cleanup()
```

SERVOMOTOR

```
from Tkinter import *
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)
pwm = GPIO.PWM(18, 100)
pwm.start(5)
class App:

    def __init__(self, master):
        frame = Frame(master)
        frame.pack()
        scale = Scale(frame, from_=0, to=180, orient=HORIZONTAL, command=self.update)
        scale.grid(row=0)

    def update(self, angle):
        duty = float(angle) / 10.0 + 2.5
        pwm.ChangeDutyCycle(duty)

root = Tk()
root.wm_title('Servo Control')
app = App(root)
root.geometry("200x50+0+0")
root.mainloop()
```