

SENSORES ANALÓGICOS CON RASPBERRY PI Y MCP3008

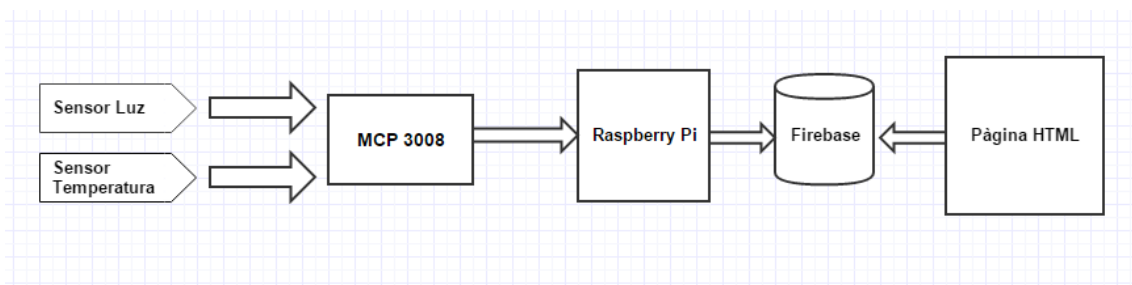


Autor: Jefferson Rivera Patiño
Email: riverajefer@gmail.com

Contents

DESCRIPCIÓN DEL PROYECTO	3
ALGO DE TEORÍA.....	3
INSTALANDO LO NECESARIO.....	4
HABILITAR LA INTERFAZ SPI.....	4
MONTAR EL CIRCUITO.....	5
CÓDIGO PYTHON.....	7
PROBAR LA APLICACIÓN WEB	8
BREVE EXPLICACIÓN DE LA APLICACIÓN WEB.....	9
USAR SU PROPIA URL DE FIREBASE.....	10

DESCRIPCIÓN DEL PROYECTO



Como se aprecia en la imagen, el sensor de luz (fotocelda) y sensor de Temperatura (LM 35), ingresan al convertidor analógico digital MCP3008, el cual envía los datos capturados a la *Raspberry Pi* por medio del protocolo SPI, luego con python se procesan estos datos y se envían a la base de datos firebase, para finalmente desde una página HTML, mostrar los valores de los sensores y cambiar una imagen de fondo de acuerdo a los datos recibidos.

ALGO DE TEORÍA

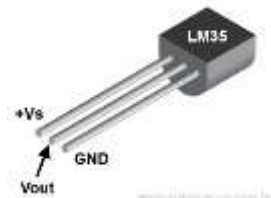
Sensor de luz (Fotocelda)

Básicamente es un dispositivo electrónico que cambia su resistencia, a medida que varía la intensidad de la luz que se aplica sobre él.



Sensor de temperatura (LM 35)

El LM35 es un sensor de temperatura con una precisión calibrada de 1 °C. Su rango de medición abarca desde -55 °C hasta 150 °C. La salida es lineal y cada grado Celsius equivale a 10 mV.¹



Convertidor Analógico Digital (MCP3008)

Es un dispositivo electrónico que convierte una señal analógica de voltaje o corriente, en una señal digital, que puede ser procesada por un ordenador.

El MCP3008 es un convertidor analógico digital, de 8 canales de 10-bit, y transmite los datos a través del protocolo serial SPI



Firebase

También se le denomina *Backend como Servicio*, que básicamente provee de una API para guardar y sincronizar datos en la nube, en tiempo real y ser consumidos desde distintos clientes² (<https://www.firebase.com>)

¹ <https://es.wikipedia.org/wiki/LM35>

² <http://www.miguel diaz rubio.com/2014/07/14/desarrollo-ios-introduccion-al-backend-firebase/#>

INSTALANDO LO NECESARIO

Actualización de repositorios

```
>> sudo apt-get update
```

Instalar Python Dev

```
>> sudo apt-get install python-dev
```

Instalar git

```
>> sudo apt-get install git
```

Instalar spidev

```
>> git clone https://github.com/doceme/py-spidev
>> cd py-spidev
>> sudo python setup.py install
```

Instalar la librería GPIOZERO

```
>> sudo apt-get install python-gpiozero
```

Instalar PIP (Manejador de Paquetes)

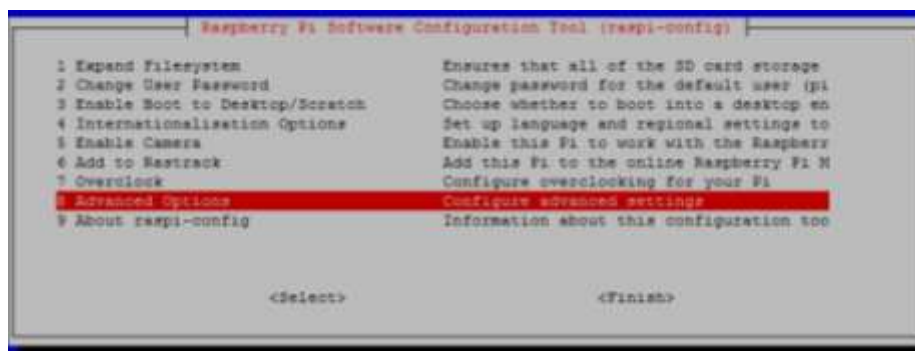
```
>> sudo apt-get install python-pip
```

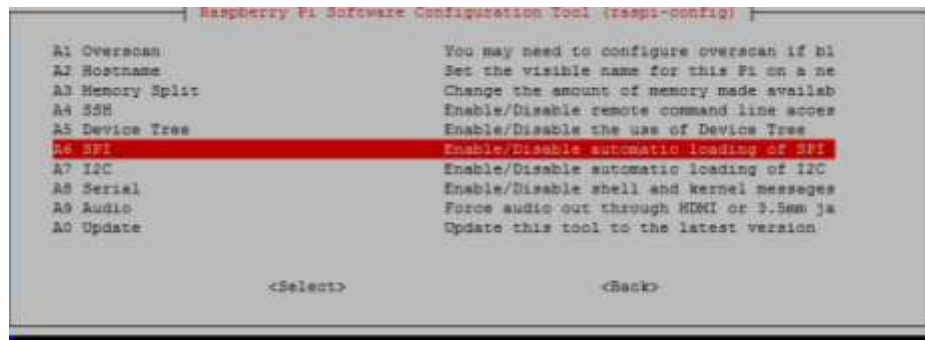
Instalar Firebase

```
>> sudo pip install requests==1.1.0
>> sudo pip install python-firebase
```

HABILITAR LA INTERFAZ SPI

```
>> sudo raspi-config
```

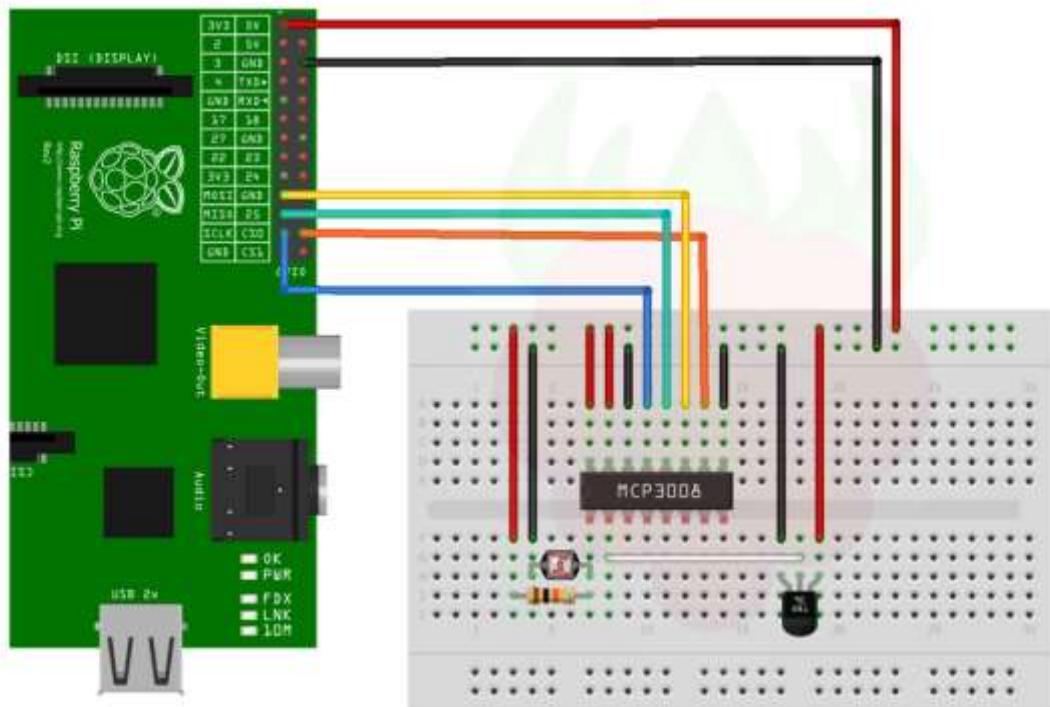




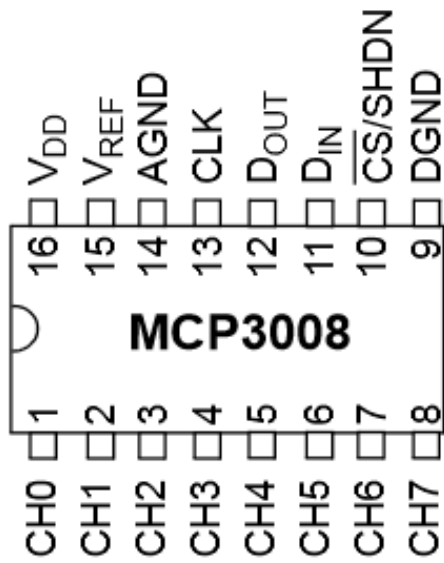
Damos en sí y reiniciamos la Raspberry Pi

```
>> sudo reboot
```

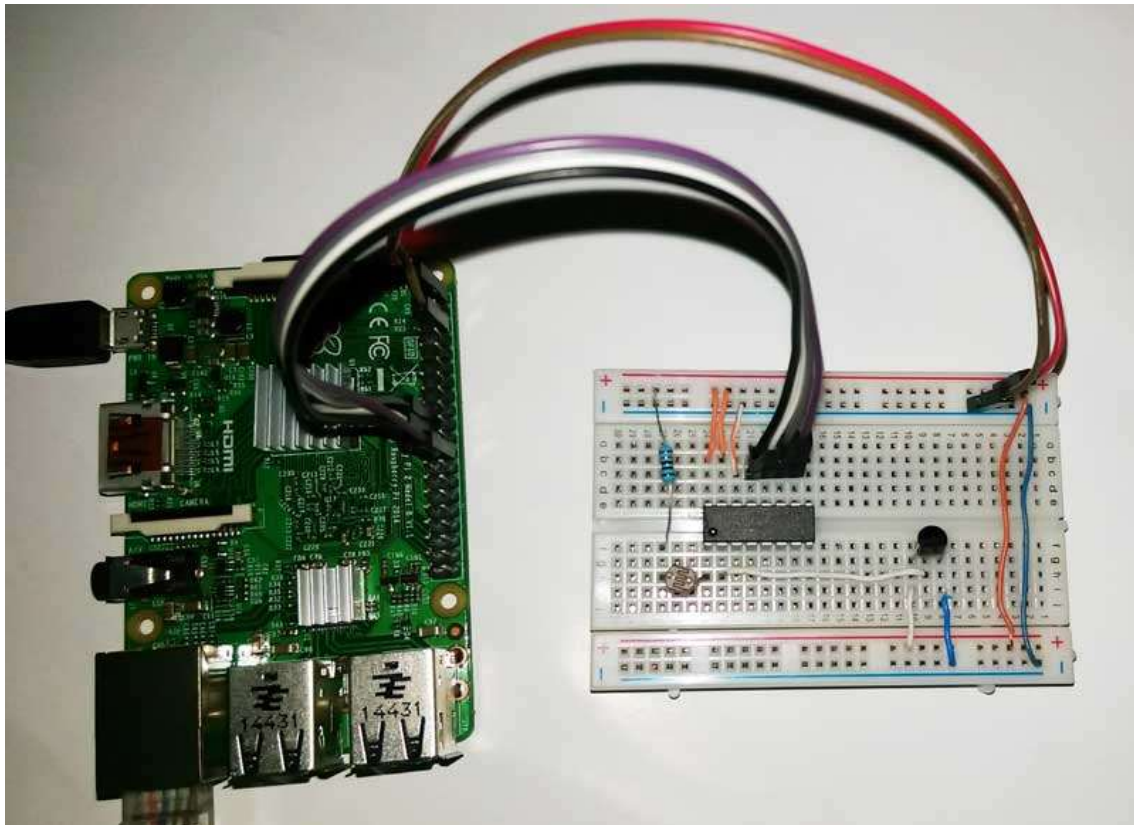
MONTAR EL CIRCUITO



(Tomado de: <http://www.rpi-spy.co.uk/2013/10/analogue-sensors-on-the-raspberry-pi-using-an-mcp3008/>)



Chip	Raspberry
VDD	3.3V
VREF	3.3V
AGND	GROUND
CLK	GPIO11 (P1-23)
DOUT	GPIO9 (P1-21)
DIN	GPIO10 (P1-19)
CS	GPIO8 (P1-24)
DGND	GROUND



CÓDIGO PYTHON

Descargar el código (python) que procesa la señal de los sensores

```
>>> sudo wget https://raw.githubusercontent.com/rpi-jefer/sensores/master/mcp3008.py
```

Ejecutar el script python

```
>>> sudo python mcp3008.py
```

Nota: Terminar el proceso con Crt+C

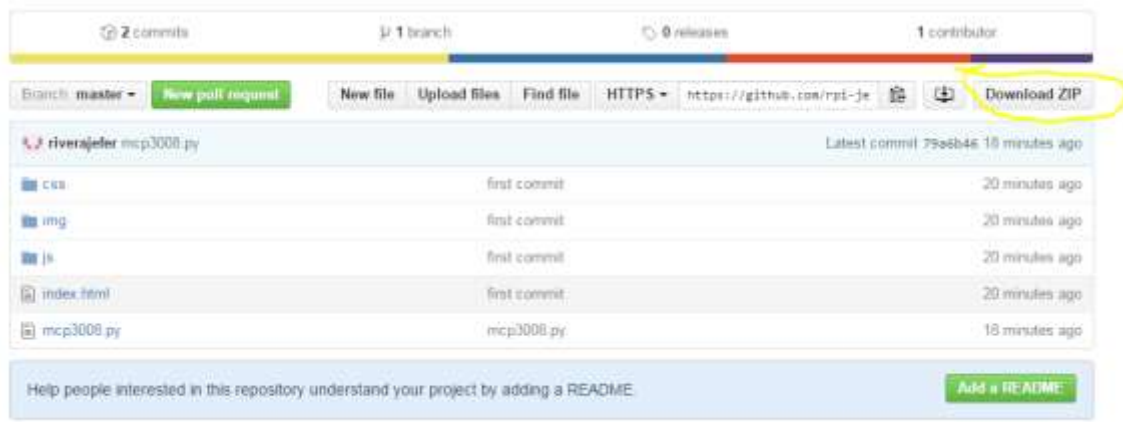
Un vistazo al código

```
# -*- coding: utf-8 -*-  
  
# Desarrollado por Jefferson Rivera  
  
from gpiozero import MCP3008  
from firebase import firebase  
from time import sleep  
  
# conexión al firebase, (acá colocas tu la URL que te genera firebase)  
fire = firebase.FirebaseApplication('https://proyectorpi.firebaseio.com', None)  
  
sensorLuz = MCP3008(0) # canal 0  
sensorTemp = MCP3008(1) # canal 1  
  
while True:  
    try:  
        # Rango de 0 a 10, donde 0 es ausencia de luz y 10 la máxima luz  
        ValorSensorLuz = sensorLuz.value*10  
        ValorSensorTemp = sensorTemp.value*500 # Convertimos el voltaje en temperatura  
        ValorSensorLuz = round(ValorSensorLuz, 2) # Redondeamos y dejamos 4 decimales  
        ValorSensorTemp = round(ValorSensorTemp, 2)  
        print "*****"  
        print "Sensor Luz: ", ValorSensorLuz  
        print "Sensor Temp:", ValorSensorTemp  
        print "*****\n"  
  
        #Actualizamos el valor de los sensores en firebase, para cada caso  
        fire.put('/sensor/', 'temp', ValorSensorTemp)  
        fire.put('/sensor/', 'luz', ValorSensorLuz)  
  
        # Esperamos 3 Segundos  
        sleep(3)  
  
    except KeyboardInterrupt:  
        print "\nSalida"  
        break
```


PROBAR LA APLICACIÓN WEB

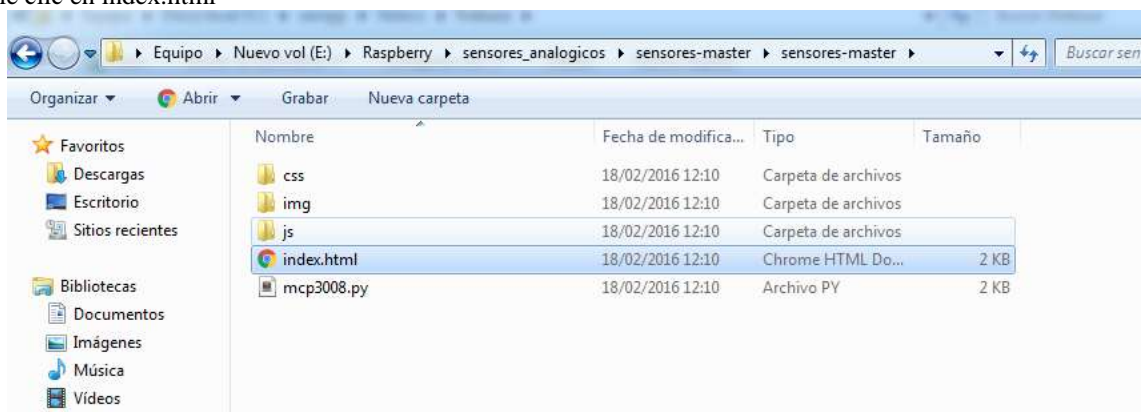
Entrar a <https://github.com/rpi-jefer/sensores>

Descargar el .zip

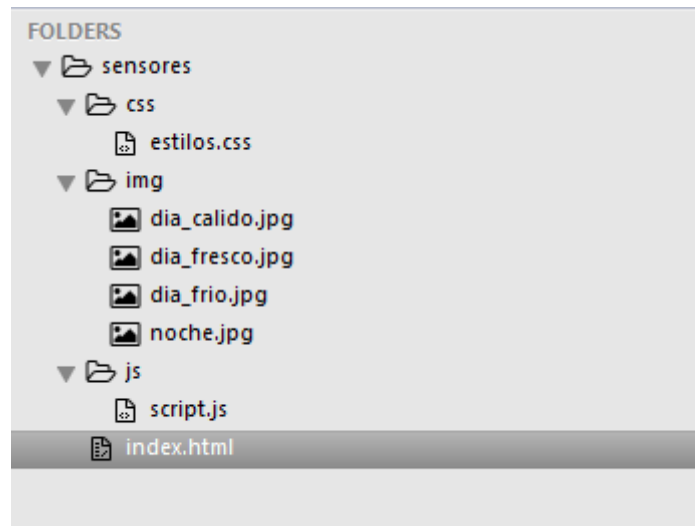


Extraer el .zip

Doble clic en index.html



BREVE EXPLICACIÓN DE LA APLICACIÓN WEB



El proyecto tiene la siguiente estructura:

Css/estilos.css: contiene la parte gráfica, por ejemplo tamaño de fuente, tipo, color etc.

Img/...: contienen las imágenes que se muestran en la pagina

Js/script: Es el archivo javascript (jQuery) que se encarga de la lógica, se conecta con firebase para obtener los datos de los sensores y de acuerdo a esto, mostrar una imagen apropiada, ejemplo: *si el sensor de luz es mayor o igual 7 y la temperatura es mayor a 24 grados Celsius, entonces muestre una imagen de día cálido.*

Index.html: donde se muestran las imágenes y los datos de los sensores.

Descargar el proyecto desde aquí: <https://github.com/rpi-jefer/sensores>

Como se van almacenando los datos en firebase



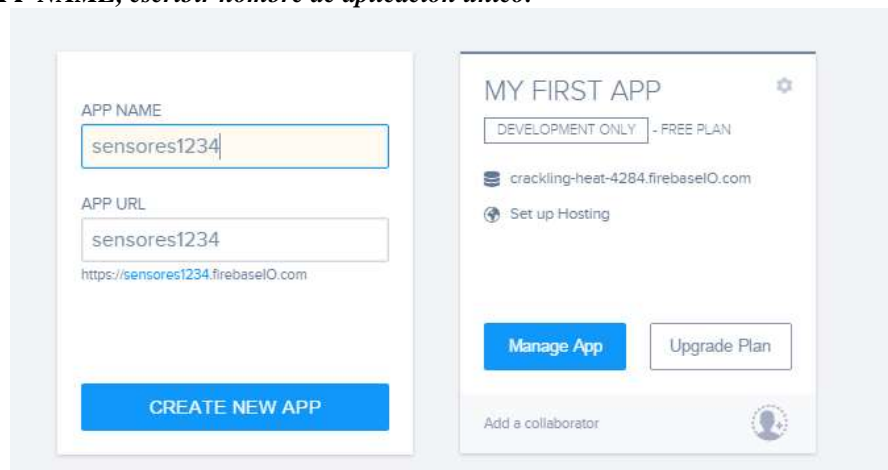
USAR SU PROPIA URL DE FIREBASE

Crear una cuenta en firebase

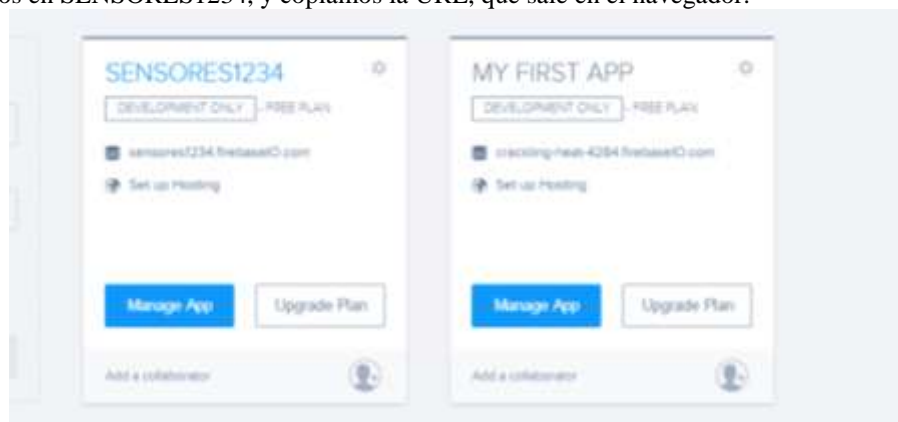
<https://www.firebase.com/login/>



En el APP NAME, escribir nombre de aplicación único.



Entramos en SENSORES1234, y copiamos la URL, que sale en el navegador.



<https://sensores1234.firebaseio.com/>

Esta es la URL que vamos a utilizar en Python y en jQuery