



Python - Internet Of Things

PyConES 14

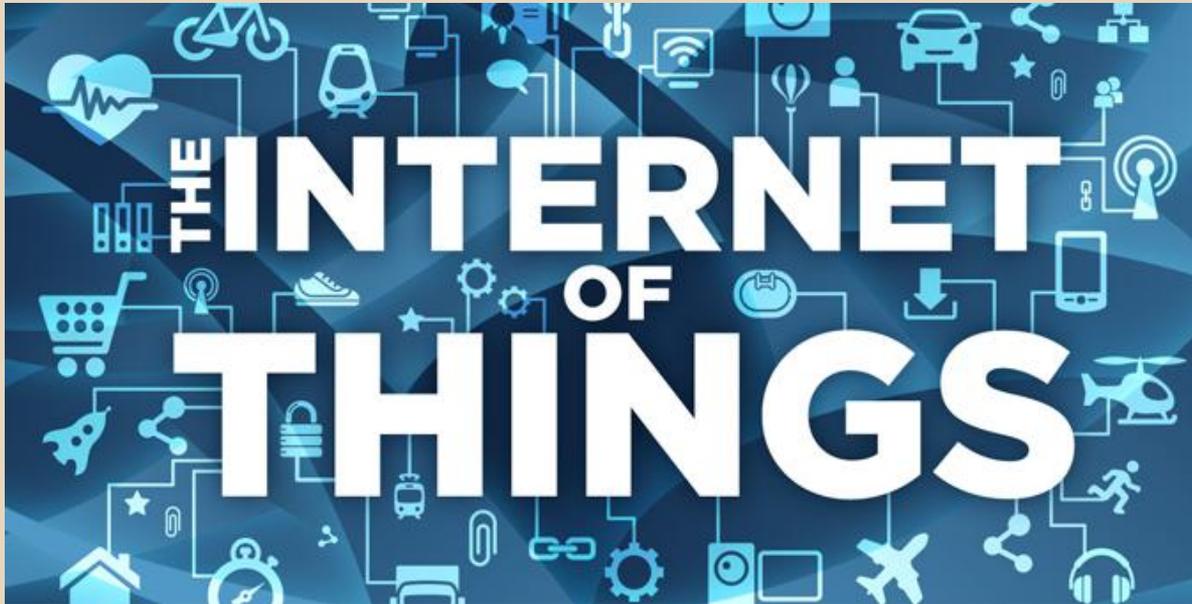
Samuel de Ancos

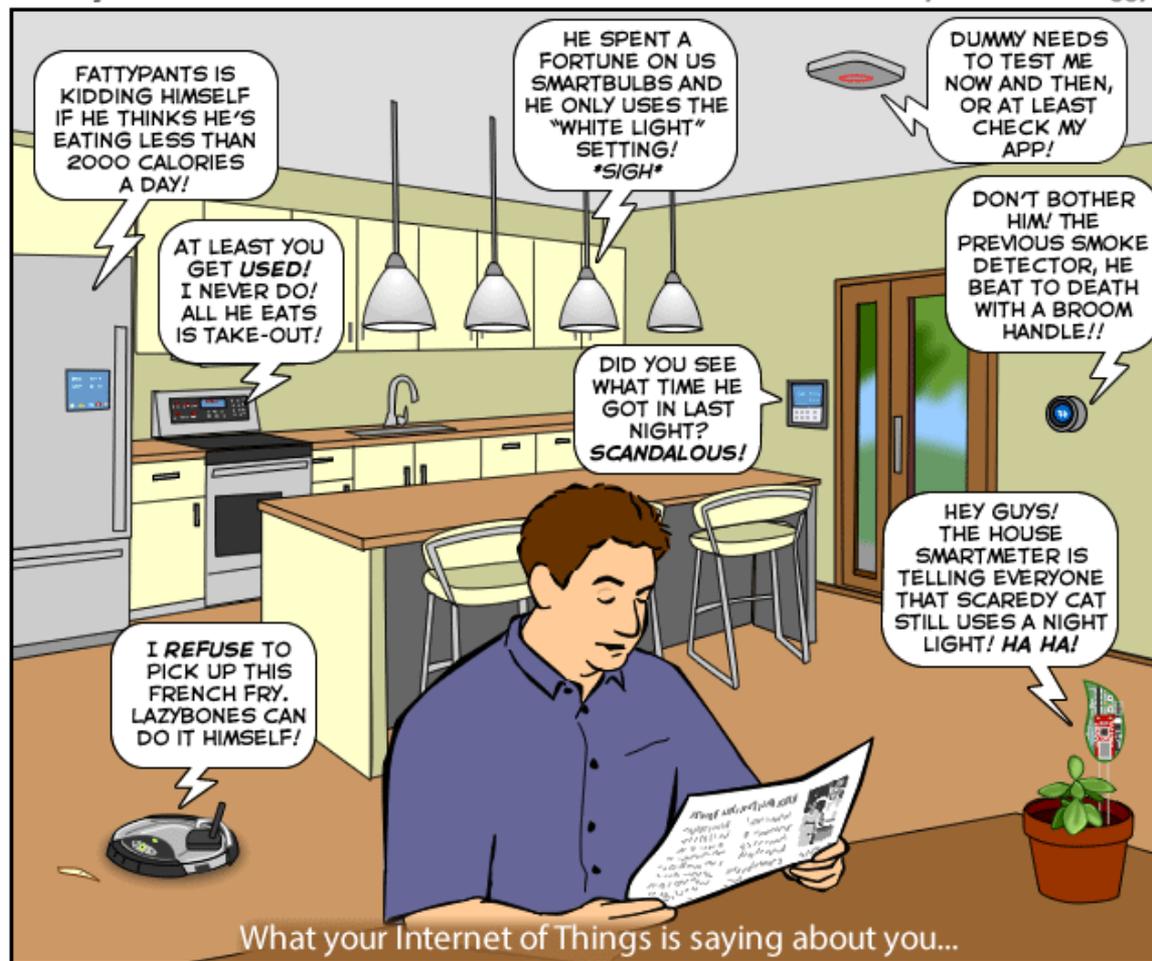
Alejandro Guirao

<http://bit.ly/13UV71m>

Introducción a IoT

<http://iotcomicbook.org>



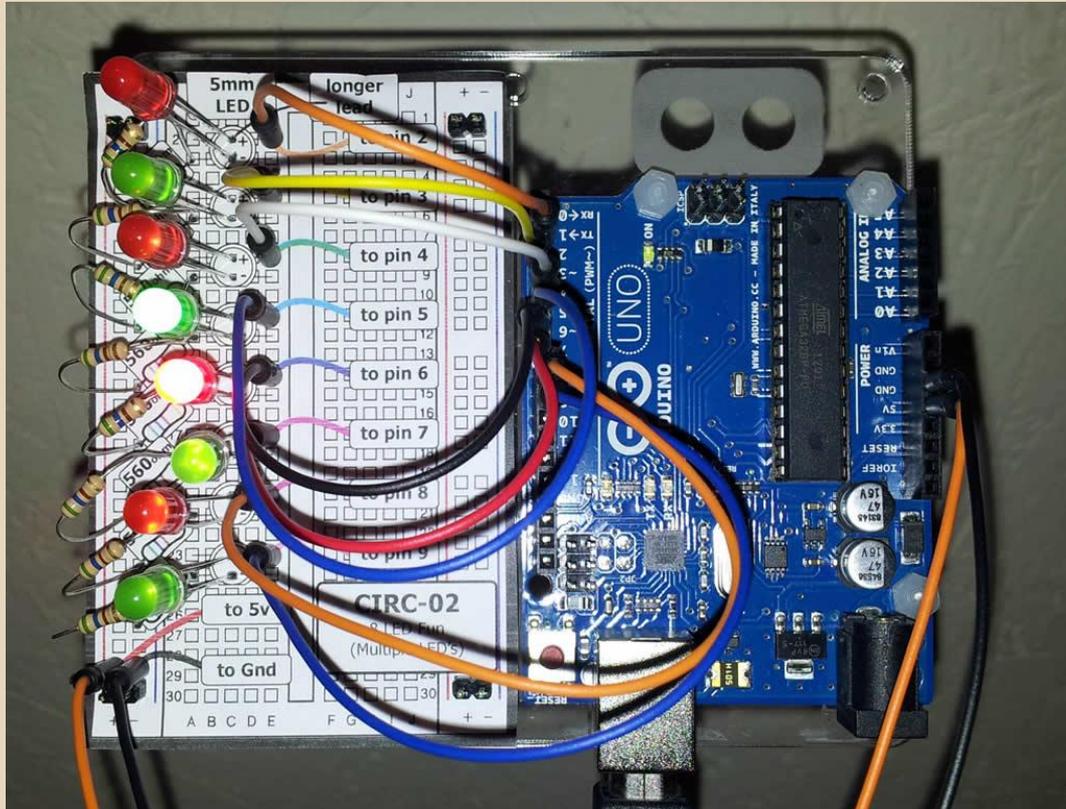


What your Internet of Things is saying about you...

Desarrollo del taller

1. Elegir un proyecto y variantes
2. Montar el circuito
3. Programar
4. ¡Enseñarnos el resultado!

Proyectos y variantes



A: Twitter Temperature Bot

- Medición de temperatura mediante termistor
- El dispositivo tiene una cuenta de Twitter asociada en la que va **tuiteando su temperatura de manera periódica**

Son las 14:15 y la temperatura es de 24.8 °C

Variante A1

- En lugar de tuitear de manera periódica, **responde a menciones** en las que aparezca un **hashtag** determinado

@bot Dame la #temperatura :-)

Variante A2

- En lugar de tuitear, tiene un umbral de temperatura **activable y configurable mediante mención con hashtag**

@bot Pon el #umbral a 24

Variante A2

- Cuando llega a ese umbral, **avisa con una mención** al usuario de twitter que lo activó. Adicionalmente, puede encenderse un **led** en la placa

@usuario ¡Se ha rebasado el umbral , la temperatura es de 25°C !

Variante A3

- Enviar los datos a un repositorio externo y **visualizarlos** en una gráfica

- Por ejemplo:
 - Plotly
 - Carriots
 - Xively

B: Twitter LDR bot

- Medición de luz mediante LDR
- El dispositivo tiene una cuenta de Twitter asociada en la que va **tuiteando el estado cuando advierte un cambio**

Son las 14:15 y alguien ha encendido/apagado la luz

Variante B1

- En lugar de tuitear al cambio de estado, **responde a menciones** en las que aparezca un **hashtag** determinado

@bot Dame el #estadoluz :-)

Variante B2

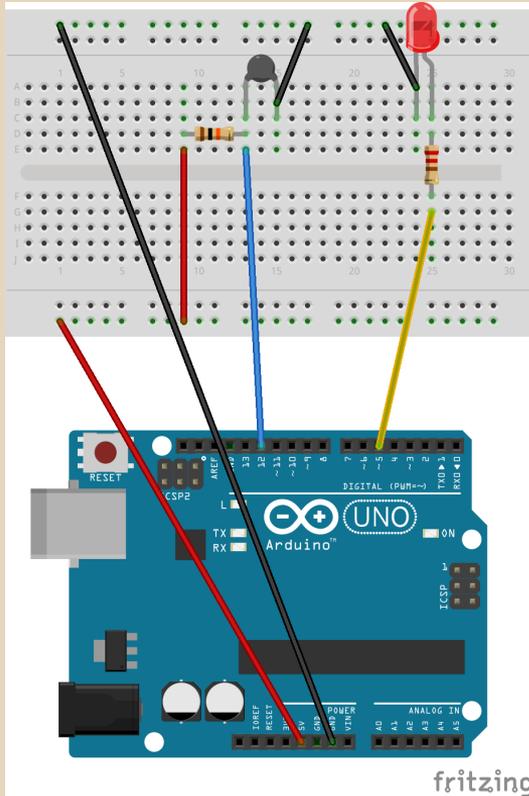
- Podemos añadir un botón para utilizarlo como activador y mandar el estado en ese momento a twitter
- También se puede añadir un LED avisador

Variante B3

- Enviar los datos a un repositorio externo y **visualizarlos** en una gráfica

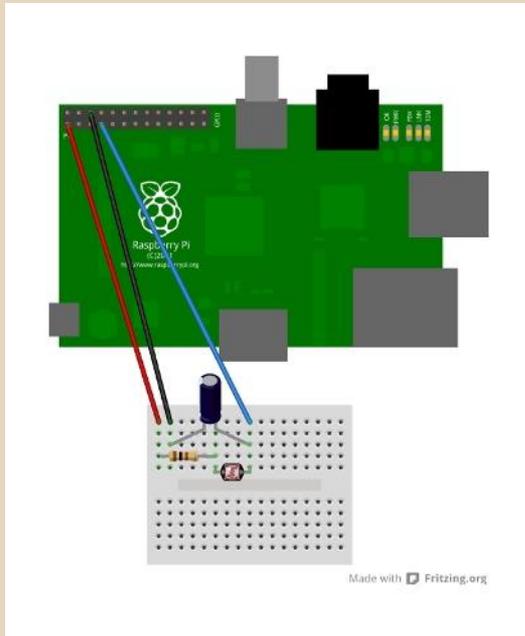
- Por ejemplo:
 - Plotly
 - Carriots
 - Xively

Circuito de proyecto A



- El LED puede servir de indicador de que se sobrepasa el umbral de temperatura
- Es aconsejable promediar varias medidas

Circuito de proyecto B



- Gracias a un filtro conseguimos simular una lectura analógica desde una entrada digital.
- Se puede añadir un LED avisador como en el anterior esquema.

Cálculo de la temperatura (°C)

$$res = res_ref / (1023 / value_read - 1)$$

$$temp = 1 / (1 / T1 + (1 / B) * \log(res / R1)) - 273.15$$

donde:

- T1 = 298.15 (temperatura de referencia, 25°C)
- B = 3977 (de la hoja de referencia del termistor)
- R1 = 10000 (resistencia del termistor a T1)
- res_ref = resistencia del divisor de tensión
- value_read = lectura analógica entre 0 y 1023

Control de Arduino por USB

- “Manualmente”:
 - pyserial para enviar y recibir datos desde Python
 - <http://arduino.cc/en/reference/serial>
- Opciones más complejas y flexibles:
 - tino/pyFirmata
 - lekum/pyduino
 - ...

Twitter

- Opciones:
 - Twython
 - Tweepy (sólo python 2.X)
 - “Manual” (requests, OAuth, ...)
- Hay que crear una serie de credenciales (APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET) en <https://dev.twitter.com/> para la aplicación
- Usar el Streaming API para recibir eventos de menciones y el API REST para actualizar el estado
 - Ejemplos con Twython en <http://twython.readthedocs.org/en/latest/>

Plotly

<https://plot.ly/python/streaming-tutorial/>

