

Memòria pràctica 7:

Disseny web del control remot d'un led

Sergi Carol i Enric Lenard

May 9, 2015

Introducció

L'objectiu d'aquesta pràctica és aconseguir controlar un led , en el nostre cas el de un arduino, a través de una interfície web. Aquesta web en permet saber l'estat del led, encender-lo i apagar-lo. La forma de fer aquesta pràctica serà la següent:

Disposarem de dos servidors, un el servidor apache que ja hem utilitzat anteriorment, i un servidor python que serà l'encarregat de gestionar l'arduino i enviar respostes al servidor apache. El servidor apache enviarà **on load** una petició GET al servidor python demanant per l'estat del led, a més a més farà aquesta petició cada 3 segons. Per la seva banda, el servidor python informará al servidor apache del estat del led quant rebi una consulta i encendra i apagarà el led també depenen del servidor apache

1 Configuració del servidor apache

La configuració del servidor apache es senzilla. Afegirem tres proxys al arxiu **practweb.conf** per tal de que poguem redirigir, i per tan per una petició, al servidor python per tal de poder realitzar tres operacions, **consultar**, **encendre** i **apagar** el led.

```
ProxyPass /led http://mare.g2.asi.itic.cat:8080/led_ask/  
ProxyPassReverse /led http://mare.g2.asi.itic.cat:8080/led_ask/  
  
ProxyPass /led_on http://mare.g2.asi.itic.cat:8080/led_on/  
ProxyPassReverse /led_on http://mare.g2.asi.itic.cat:8080/led_on/  
  
ProxyPass /led_off http://mare.g2.asi.itic.cat:8080/led_off/  
ProxyPassReverse /led_off http://mare.g2.asi.itic.cat:8080/led_off/
```

Podem veure els tres proxys mencionats anteriorment.

A més a més hem creat un nou arxiu .html anomenat **led.html** que s'encarrega de gestionar l'estat del led dintre el servidor apache. Per exemple s'encarrega de demanar al servidor python el estat del led i dir si vol encendre o apagar el led. Per fer-ho hem creat 4 funcions amb javascript que s'encarreguen de enviar peticions al servidor python i canviar una imatge en funcio de la resposta del servidor python. Basicament el seu funcionament és el següent:

```
<div id="myDiv">
    <br></td>
</div>
```

Aqui podem veure la imatge que indica si el led esta ences o apagat, com podem veure la source de la imatge es desconeguda en un principi. Es important veure que l'identificador de la imatge es *img*.

```
window.onload = function load()
{
    var xmlhttp;
    var objecte_div;
    objecte_div=document.getElementById("img");
```

Un cop ja en el entorn de javascript creem una funcio, en aquest cas una que s'executa al carregar la pagina, que busca un element dintre el document html amb l'identificador *img*

```
xmlhttp.open("GET","/led",true);
xmlhttp.send();
```

A continuació enviem una petició **GET** a la url **/led** això en portara a **g2.asi.itic.cat/led**, que com hem vist abans, es una url de un proxy que porta a **mare.g2.asi.itic.cat:8080/led_ask/** la qual demana el estat del led.

```
if (xmlhttp.responseText=="Off")
{
    objecte_div.src = "ledoff.png";
    console.log("Off")
}
if (xmlhttp.responseText=="On")
{
    console.log("On")
    objecte_div.src="ledon.png";
}
```

Finalment comprobem la resposta que en dona el servidor python. Podem veure com podem comparar la resposta del servidor gràcies a `xmlhttp.responseText`. En el cas de que la resposta sigui *Off* canviarem la *sources* de la imatge que apunta a `objecte_div` per *ledoff.png*, en el cas contrari la canviarem per *ledon.png*.

D'aquesta forma podem demanar l'estat del led al carregar la pàgina. Encara hi han tres funcions més, una que demana cada 3 segons l'estat, una altre que al apretar el boto de encendre envia una petició per encendre el led, i una altre que apaga el led.

Per tal de poguer cridar funcions al apretar el led ho hem fet amb la següent funció:

```
<button onclick="on()">ON</button>
```

En aquest cas aquest boto executa la funció `on()` que envia una petició per encendre el led. Per últim per tal de poguer executar un script cada 3 segons fem:

```
setInterval("loadXMLDoc()",3000);
```

2 Configuració servidor python

2.1 Creació del servidor python

En aquest apartat parlarem de la creació del servidor python. La forma de fer-lo ha sigut seguint el tutorial de l'enunciat de la pràctica, amb una calsse.

Existeixen 4 possibles paths en el servidor, 5 si contem que els altres donaran error 404. Els paths que s'utilitzaran seran `/led_ask/`, que demanarà l'estat del led, `/led_on/` que encendra el led i finalment `/led_off/` que apagarà el led. Tots aquests paths al final envien una resposta al servidor d'on han rebut la petició, en el cas de `/led_on/` i `/led_off/` restornen **On** i **Off** respectivament, mentre que `/led_ask/` retorna On o Off depenent del estat led. Com a exemple podem veure `/led_ask/`:

```
def _LED_ASK(self):
    if led.estat(ser) == '1':
        resposta="On"
    else:
        resposta = "Off"
    self._head_html()
    self.wfile.write(resposta)
```

La funció `led.estat(ser)` demana al arduino l'estat, però això en parlarem més endavant. Podem veure com depenen de lo que retorna l'arduino enviem On o Off.

2.1.1 Controlador del Arduino

Finalment pararem sobre el scrip que controla el Arduino. Per tal de realitzar aquest control s'han creat dos codis, un encarregat de controlar l'arduino desde dintre, un codi .c, i un scrip de python que parla pel port serie cap al arduino per tal de enviar-li ordres.

Primer parlarem sobre el scrip python. El scrip és senzill, primer de tot obrim el port serie mitjançant **ser=serial.Serial('/dev/ttyACM0',9600)**, un cop el tenim obert podem enviar comandes atraves d'aquest amb les funcions **on** , **off** i **estat**, a continuació les funcions mencionades.

```
import serial
def on(serial):
    serial.write("O")

def off(serial):
    serial.write("F")

def estat(serial):
    serial.write("C")
    a=serial.read()
    return a
```

Per últim tenim el codi C encarregat de controlar l'arduino desde dintre. El codi es senzill, el microcontrolador entra en un bucle esperant rebre algun caracter desde el port serie i quant el rep comprova quin caracter és i depenent del que sigui relitza una acció o un altre.

```
void lectura(void){
    uint8_t a;
    bool s;
    if (serial_can_read()){
        a=serial_get();
        if (a == 'O') pin_w(led,true);
        else if (a == 'F') pin_w(led,false);
        else if (a == 'C') {
            s = pin_r(led);
            if (s == true) serial_put('1');
            else if (s == false) serial_put('0');
            pin_w(led,s);
        }
    }
}
```