Proyecto de Introducción a la Ingeniería

Domótica accesible

Prendido/apagado de un electrodoméstico a través de una página web.

Damian Galdames – 201312040.7
Esteban Jara - 201330002-2
Gerson Pincheira - 201330022-7
Bryan Rosales - 201330040-5
Joaquin Vasquez -
11/09/2013

Índice de contenido

COMPONENTES A UTILIZAR

1. Raspberry pi
2. Arduino (en nuestro caso, un Arduino Nano)
3. Relé o relevador
4. Protoboard
5. Transistor
6. Diodo
7. 2 Resistencias
8. Batería (9V)
9. Lámpara (objeto electrónico conectado a la red eléctrica (220V) sobre el cual se va a operar)

CONFIGURAR RASPBERRYPI

- Asumiendo que nuestra RaspberryPi es nueva “de paquete”, y que ya hemos instalado RaspBian en ella, la conectamos a un monitor y a un teclado y la prendemos.

- Necesitarás hacer labores de limpieza, actualización e instalación. Primero, actualizaremos el reloj, las fuentes, y los paquetes pre-instalados. Usa el siguiente código para hacer esto:

    -sudo dpkg-reconfigure tzdata

    -sudo apt-get update

    -sudo apt-get upgrade

- Ahora, queremos instalar la herramienta de actualización Hexxeh's RPI para mantener el Raspberry Pi actualizado. Para hacer esto, ejecuta:

    -sudo apt-get install ca-certificates

    -sudo apt-get install git-core

    -sudo wget http://goo.gl/1BOfJ -O /usr/bin/rpi-update && sudo chmod +x /usr/bin/rpi-update

    -sudo rpi-update

    -sudo shutdown -r now

- Si es que no lo hiciste en la configuración inicial de RaspBian, habilitaremos el SSH para poder hacer todo desde una computadora diferente. Para hacer esto, primero anota la dirección IP del Raspberry Pi (puedes usar el comando *ifconfig*), luego ejecuta lo siguiente:

    -sudo mv /boot/boot_enable_ssh.rc /boot/boot.rc

    -sudo shutdown -r now

- Ahora puedes desconectar los cables para tu teclado USB y tu monitor. Ya no los necesitas, todo se hará en SSH.
- ¡OJO! Para poder comunicarte por SSH con la Rasp debes estar conectado al mismo router que ésta (deben estar en una misma red local).

- Abre tu cliente SSH y conectate a la Raspberry Pi , ejecutando en la terminal:

    -ssh pi@*IP_de_la_raspbierry*

- O bien, si deseas conectarte desde Windows, recomendamos el cliente PuTTy. Una vez abierto PuTTy, ingresas pi@*IP_de_la_raspbierry* y 22 donde pide el puerto.
- También debemos instalar web.py y python-serial, que son módulos del lenguaje de programación Python. El primero nos habilita para utilizar las funciones orientadas al desarrollo de páginas web, y el segundo nos permite manipular los puertos seriales (USB) de la Rasp y asignarlos como variables. Para instalar web.py ingresamos el siguiente código:

    -wget http://webpy.org/static/web.py-0.37.tar.gz

- con esto bajamos un archivo en compresión tar.gz que luego deberemos descomprimir y entrar en la carpeta resien extraida para comenzar la instalacion. Para estos pasos usaremos los siguientes codigos en la terminal:

    -tar xvf (nombre del archivo en este caso web.py-0.37.tar.gz)
    -cd web.py-0.37(con esto entramos en la carpeta)
    -python setup.py install

- Para instalar python-serial, ejecutamos:

    -apt-get install python python-serial

INSTALACIÓN DEL SERVIDOR

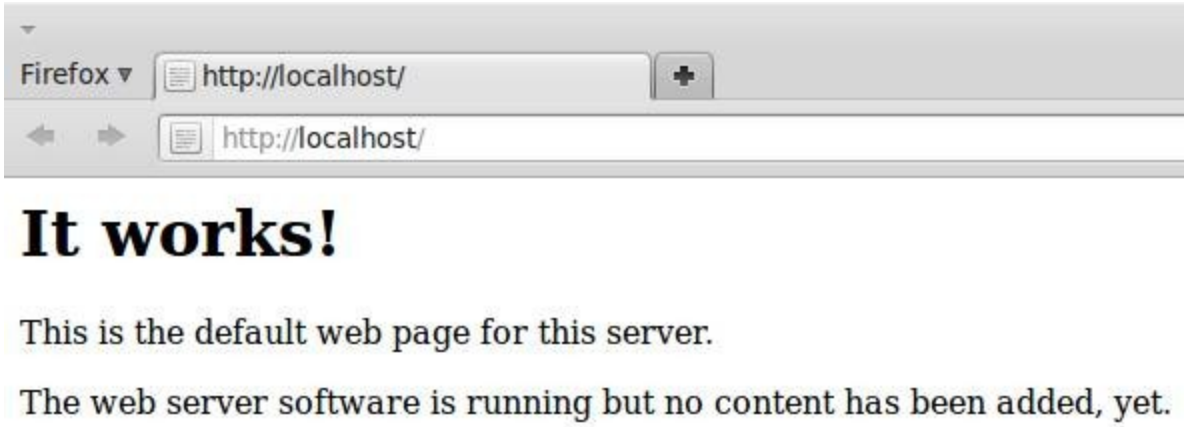
- Instala Apache (servidor web) y PHP (lenguaje de programación orientado a la creación de páginas web dinámicas), esto nos habilitará para subir nuestra página la red o a la WWW. Ejecuta este comando:

    -sudo apt-get install apache2 php5 libapache2-mod-php5

- Ahora debes reiniciar el servicio:

    -sudo service apache2 restart

- Ahora introduce la dirección IP de tu Raspberry Pi en tu navegador web y verás una simple página que dice "It Works!"



IMPLEMENTAR EL SISTEMA

- Ahora debemos implementar el sistema que manejará el Arduino. Éste consta principalmente de un relé, acompañado de varios componentes que habilitan su funcionamiento insertos en una protoboard:

  Relé o relevador: El relé o relevador es un dispositivo electromecánico. Funciona como un interruptor controlado por un circuito eléctrico (en nuestro caso, la señal de 5V del Arduino asistida por la batería) en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.
  Transistor: Aumenta el amperaje de la señal del Arduino.
  Diodo: Impide que se queme el transistor cuando la corriente se interrumpe en la bobina del relé.
  LED: Conectado en paralelo a los nodos que alimentan al relé para indicar si pasa voltaje o no
  2 Resistencias: Una resistencia que impide que la corriente queme al Arduino entre la base del transistor y el pin del arduino, y otra para no quemar el LED.
  Batería: Da el voltaje necesario para activar al relé.
  Protoboard: Tablero con orificios conectados eléctricamente entre sí, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares

- En nuestro caso, buscamos ayuda de alguien con experiencia en manipulación de circuitos para implementar estos componentes.
- Nuestro relé interrumpe el cable de corriente de una lámpara convencional

CONFIGURAR ARDUINO

- ¿Qué es un Arduino? Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares
- Debemos crear el programa con las órdenes que seguirá el Arduino dependiendo de los comandos que le enviemos:

```
int rele = 13;

void setup () {
  pinMode(rele, OUTPUT); //RELE 13 como salida
  Serial.begin(9600); //Inicializo el puerto serial a 9600 baudios
}


void loop () {
  if (Serial.available()) { //Si está disponible
    char c = Serial.read(); //Guardamos la lectura en una variable char
    if (c == 'H') { //Si es una 'H', acciono el relé
      digitalWrite(rele, HIGH);
    }
      else if (c == 'L') { //Si es una 'L', apago el relé
       digitalWrite(rele, LOW);
    }
  }
}
```

- Éste programa[1] declara la variable “led” como entero, a la que le asignaremos el valor 13, que es el pin donde está conectado. Tras esto, inicializaremos la comunicación y, en el ‘loop()’, que es la función principal, leeremos constantemente los valores que nos estén llegando desde Python.

[1]

IMPLEMENTAR LA PÁGINA

En este punto es donde comenzamos a usar los módulos de python descargados en el principio.

- Primero creamos un programa en python web que sera visto desde nuestra ip en cualquier navegador de cualquier dispositivo que este dentro de la misma red local.
- Usaremos python serial para comunicar nuestra Raspberry pi con el Arduino ya configurado y darle la orden correspondiente.
- El programa tendrá la siguiente estructura :
- **estáticos**
  jquery.js
  tutorial.css
  plantillas
  tutorial.html
  app.py

  todo esto en la carpeta /var/www
- el programa quedara de la siguiente forma (app.py):

```
import web
import serial


def make_text(string):
    return string

urls = ('/', 'tutorial')
render = web.template.render('/var/www/html0/pepito/templates/')#ubicacion de templates
arduino = serial.Serial('/dev/ttyUSB0', 9600)

app = web.application(urls, globals())

my_form = web.form.Form(
                web.form.Textbox('', class_='textfield', id='textfield'),
                )

class tutorial:
    def GET(self):
        form = my_form()
        return render.tutorial(form, "Your text goes here.")

    def POST(self):
        form = my_form()
        form.validates()
        comando = form.value['textfield']
#       comando='H o L'
        if comando == 'H':
                arduino.write(comando) #Mandar un comando hacia Arduino
                return make_text('Encendido')
        if comando == 'L':
                arduino.write(comando)
                return make_text('Apagado')
        else:
                return make_text('Ingrese un comando valido (H o L)')

if __name__ == '__main__':
    app.run()

```

- luego hay que ver la plantillas (templates) en formato html nombrada en el inicio esta es la que le dará la forma a la pagina web y la cual es llamada en el programa en python(app.py).

```
$def with (form, text)

<!doctype html>

<html>

    <head>
        <title>Proyecto Introducci&oacute;n a la Ingenier&iacute;a</title>
        <link rel="stylesheet" type="text/css" href="/static/tutorial.css" />

        <script type="text/javascript" src="/static/jquery.js"></script>

        <script type="text/javascript">
                                jQuery(document).ready(function() {
                                jQuery(".button").click(function() {
                                        var input_string = $$("input#textfield").val();
                                        jQuery.ajax({
                                                type: "POST",
                                                data: {textfield : input_string},
                                                success: function(data) {
                                                jQuery('#foo').html(data).hide().fadeIn(1500);
                                                },
                                                });
                                        return false;
                                        });
                                });

                        </script>
    </head>

    <body>
        <br>
        <H1>Proyecto Introducci&oacute;n a la Ingenier&iacute;a</H1>
        <form class="form" method="post">
        $:form.render()
        <input class="button" type="submit" value="send"/>
        </form>

        <br><br>
        <span id="foo">$text</span>
    </body>

</html>
```

- Luego de tener el programa listo, verificamos que apache este encendido y nuestro arduino conectado entonces corremos en programa:
    - -python /var/www/app.py (o en alguna sub carpeta dentro de /www esto no afecta el funcionamiento final)
- una vez iniciado no aparecerá un mensaje 0.0.0.0:8080/ esto nos dice que el programa esta corriendo en el puerto 80 de nuestro servidor, lo que debemos hacer para ver si funciona correctamente es entrar desde cualquier dispositivo en nuestra red local a la dirección http://(ip de nuestra Raspberry):8080 , esto nos va dirigir al programa en si con un cuadro en donde podremos digitar H o L en nuestro caso y esa información sera enviada mediante USB al Arduino el cual la interpretara deacuardo al programa que ya revisamos anteriormente.

FUENTES:

Idea principal del proyecto donde explica los usos que se le puede dar:

http://www.puigros.es/es/

como crear un servidor en nuestra Raspberry Pi :

http://es.wikihow.com/hacer-un-servidor-web-Raspberry-Pi

conexiones con Arduino Raspberry pi:

http://www.geekytheory.com/arduino-raspberry-pi-raspduino/

crear pagina web basada en python:

http://kooneiform.wordpress.com/2010/02/28/python-and-ajax-for-beginners-with-webpy-and-jquery/