Arduino Segunda Parte

Introducción

En este segundo post vamos a realizar una pequeña aplicación que controla dos leds (uno rojo y uno verde) conectados a Arduino. Esta placa estará conectada a un pc con un servidor servidor web. En dicho servidor web alojaremos un archivo PHP que se encargará de enviar valores a través de USB al arduino. Con un pequeño programa en Android atacaremos mediante Post a ese fichero. De esta manera podremos controlar los leds desde nuestro terminal, siempre y cuando estemos en la misma red local que el servidor web.

- GNU\Linux, venditos ficheros



GNU\Linux es un sistema operativo que utiliza el núcleo Linux junto con bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software.
Bien, en GNU\Linux el hardware esta asignado a un fichero. Así pues, solo tendremos que esribir en el fichero que identifica al puerto USB donde tenemos conectado nuestro Arduino, ni drivers ni complicaciones. Abrir en modo escritura, escribir i cerrar el fichero, así de fácil.

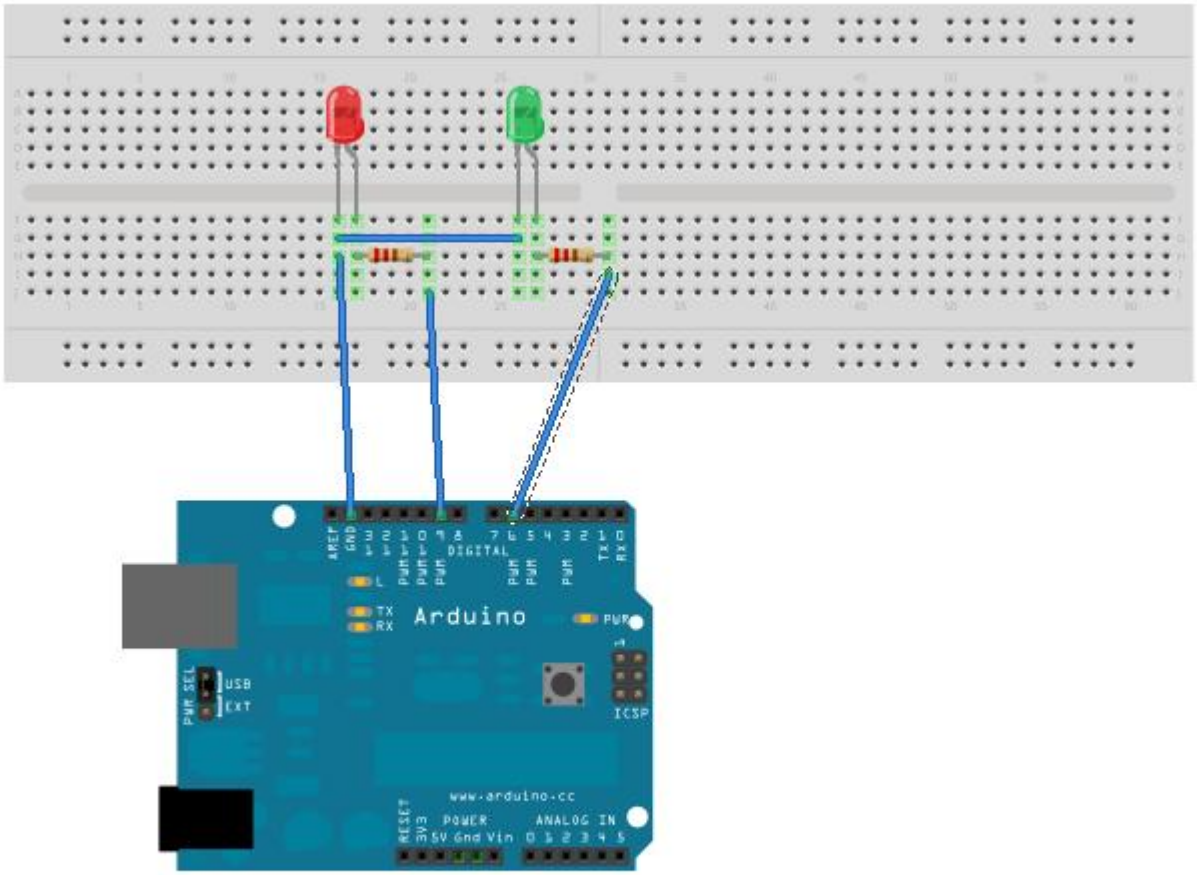
- Android

Android es un sistema operativo basado en el núcleo Linux diseñado originalmente para dispositivos móviles, tales como teléfonos inteligentes, pero que posteriormente se expandió su desarrollo para soportar otros dispositivos tales como tablet, reproductores MP3, netbook, PC, televisores, lectores de e-book e incluso, se han llegado a ver en el CES, microondas y lavadoras. Crearemos una pequeña aplicación que enviará un post a un fichero php que se encargará de escribir en el puerto USB correspondiente. Por la simplicidad de la aplicación y no ser Android objetivo del post sólo pondremos el código relativo a la ejecución del post.

Instalación de paquetes necesarios

Deberemos instalar las librerías y el IDE. Para ello podeis consultar la primera parte de la serie Arduino Arduino Primera Parte

Diseño electrico del Arduino



Código Arduino

```
int ledPin = 9;
int ledPin2 = 6;
int number_in = 0;

void setup() {
pinMode(ledPin, OUTPUT);
pinMode(ledPin2, OUTPUT);

Serial.begin(115200);
}

void loop() {
if (Serial.available() > 0) {
```

```
number_in = Serial.read();
Serial.print("He rebut: ");
Serial.println(number_in, DEC);
}

while (number_in == 49) {
digitalWrite(ledPin, HIGH);
delay(800);
digitalWrite(ledPin, LOW);
delay(800);
if (Serial.available() > 0) {
number_in = Serial.read();
Serial.print("He rebut: ");
Serial.println(number_in, DEC);
}
}

while (number_in == 51) {
digitalWrite(ledPin2, HIGH);
delay(800);
digitalWrite(ledPin2, LOW);
delay(800);
if (Serial.available() > 0) {
number_in = Serial.read();
Serial.print("He rebut: ");
Serial.println(number_in, DEC);
}
}
}
```

Servidor Web

- Deberemos dar permisos al usuario www-data (usuario por defecto del servidor Apache) para escribir en el fichero que corresponde al puerto USB:

```
sudo adduser www-data dialout
```

- En el servidor web alojaremos el fichero que se encargará de enviar las señales a Arduino:

```
if (!empty($_POST['green'])) {
$return['msg'] = "green";
// '/dev/ttyACM1' corresponde a al puerto USB deberemos cambiarlo por el que corresponda en nuestro PC
$fp =fopen("/dev/ttyACM1", "w");
fwrite($fp, chr('51'));
fclose($fp);
}else if (!empty($_POST['red'])){
$return['msg'] = "red";
$fp =fopen("/dev/ttyACM1", "w");
fwrite($fp, chr('49'));
fclose($fp);
}else if (!empty($_POST['off'])){
$return['msg'] = "off";
$fp =fopen("/dev/ttyACM1", "w");
fwrite($fp, chr('48'));
fclose($fp);
}
```

Código Android

- Encender led Verde

```
HttpClient httpclient = new DefaultHttpClient();
HttpPost httppost = new HttpPost("http://localhost/post.php");

List postValues = new ArrayList(2);
postValues.add(new BasicNameValuePair("green", "green"));
httppost.setEntity(new UrlEncodedFormEntity(postValues));
```

```
HttpResponse response = httpclient.execute(httppost);
```

- Encender led Rojo

```
HttpClient httpclient = new DefaultHttpClient();
HttpPost httppost = new HttpPost("http://localhost/post.php");

List postValues = new ArrayList(2);
postValues.add(new BasicNameValuePair("red", "red"));
httppost.setEntity(new UrlEncodedFormEntity(postValues));

HttpResponse response = httpclient.execute(httppost);
```

- Apagar leds

```
HttpClient httpclient = new DefaultHttpClient();
HttpPost httppost = new HttpPost("http://localhost/post.php");

ListpostValues = new ArrayList(2);
postValues.add(new BasicNameValuePair("off", "off"));
httppost.setEntity(new UrlEncodedFormEntity(postValues));

HttpResponse response = httpclient.execute(httppost);
```

Observaciones

Con esta segunda parte podemos ver una aproximación a las posibilidades de arduino. De esta manera podemos controlar cualquier cosa que connectemos a Arduino (las posibilidades son infinitas) y es que con este «juguete» se sabe cuando se empieza pero no cuando se acaba.

Fuentes

- Wikipedia – Android
- Android Hello World

Ruben