

Llamada a Arduino a través de aplicación cgi

Resumen

SE describe cómo establecer comunicación con Arduino desde un formulario HTML, utilizando una aplicación cgi escrita en lenguaje C. Una aplicación cgi es un ejecutable que se coloca en un servidor web, y al que se le llama desde un formulario HTML. Cuando la llamada es de tipo POST, el cgi recibe en su entrada estándar (stdin) una cadena con los datos del formulario, y envía la respuesta por la salida estándar (stdout), creando una página HTML dinámica.

En nuestro caso básicamente este es el proceso: un formulario HTML llama a una aplicación `serie_cgi.exe` que se halla en un servidor Apache, pasándole dos parámetros (*arduinoID* y *codigo*). La aplicación cgi establece comunicación con Arduino a través del puerto serie, y los datos recibidos desde Arduino los devuelve a una página HTML.

Equipo

PC xp, servidor web AppServ, compilador C WinGW

Para utilizar aplicaciones cgi es necesario tener un servidor web. He probado varios (xitami, IIS, AppServ), y AppServ me ha resultado extremadamente fácil de instalar. Es un paquete que contiene Apache+mysql+php. Se baja el ejecutable desde <http://www.appservnetwork.com/> y se instala con pocos clics. No hace falta configurar nada! Yo había intentado instalar y configurar mysql y Apache en otros tiempos, y la configuración era lo más engorroso. En este caso, ha sido increíblemente fácil.

Hay que tener en cuenta dos directorios: el “www” donde se colocan nuestras páginas HTML, y el “cgi-bin” donde se colocan los ejecutables cgi. En mi caso han quedado así:

C : /AppServ/www

C : /AppServ/www/cgi-bin

Tal como viene configurado Apache, los programas cgi sólo se pueden ejecutar en el directorio “cgi-bin”

Elementos necesarios para la comunicación con Arduino:

La página HTML: **c : /appserv/www/arduino1_cgi.htm**

Aplicación cgi: **c : /appserv/www/cgi-bin/serie_cgi.exe**

En el ejemplo, para cargar en el navegador la página del formulario se tecllea

http://localhost/arduino1_cgi.htm

La dirección IP en este caso es **localhost**, porque está en mi propio ordenador; ahí se puede cambiar, y poner la dirección IP del equipo en la red. Por ejemplo, en una red interna, podría ser:

http://192.168.0.3/arduino1_cgi.htm

En la página html lo más importante es la etiqueta **<form>**, donde se llama a la aplicación pasándole los parámetros del mismo:

```
<form action="http://localhost/cgi-bin/serie_cgi.exe" method="post">
```

La dirección IP **localhost**, es la del propio ordenador; se puede poner la dirección IP del equipo en la red:

```
<form action="http://192.168.0.3/cgi-bin/serie_cgi.exe" method="post">
```

El ejecutable C se ha modificado un poco para que funcione como cgi en lugar de funcionar desde la consola: lee en la entrada estándar la cadena de caracteres que se el envía. En el ejemplo, al inicio de `serie_cgi.exe` lo primero que se hace es

```
fgets(szBuff, 98, stdin);
```

Y la cadena leída es algo así:

```
arduinoID=ARDUINO1&codigo=5
```

donde todos los `nombre_input=contenido_input` se encadenan con el carácter ‘&’
Existen librerías para parsear esta cadena que llega desde el formulario. En este caso, siendo tan sencilla, el parseo se hace “a pelo”, para extraer los contenidos de ‘`arduinoID`’ y ‘`codigo`’

La aplicación se comunica con Arduino por el puerto serie, y envía la respuesta por la salida estándar (`stdout`). Es importante en cgi que la primera línea que se envía sea esta:

```
printf("Content-type: text/html\n\n");
```

El resto de la salida se puede formatear creando dinámicamente una página HTML (colores, tablas, listas...), o simplemente enviar líneas separadas con `
` o `<p>`

El sketch en Arduino es el mismo que se usaba en modo consola.