

Laboratorio de Programación de Sistemas Embebidos en Red

José Luis Cánovas Sánchez
Ezequiel Santamaría Navarro

14 de enero de 2016

Resumen

Descripción detallada del desarrollo en el laboratorio de la asignatura,

Índice

1. Práctica 1. Introducción a Arduino. Descripción del hardware	2
1.1. Esquemático de conexiones	2
2. Práctica 2. Montaje de un servidor web de monitorización. Descripción del software	2
2.1. Montaje de red	2
2.2. Protocolo serie	2
2.3. Servicio web	2
3. Notas adicionales	2
3.1. Aproximación a la temperatura real	2
3.2. Lector de pulsaciones	2

1. Práctica 1. Introducción a Arduino. Descripción del hardware

1.1. Esquemático de conexiones

2. Práctica 2. Montaje de un servidor web de monitorización. Descripción del software

En la segunda parte de la práctica hemos puesto como objetivo desarrollar algo en el ámbito de internet de las cosas. Básicamente nuestro reto ha sido poner en internet (en nuestro caso usando HTTP y servicios Web) los distintos sensores y el monitor LCD utilizados en la práctica anterior.

Para ello nos hemos servido de un Arduino Due (ya que el voltaje de sus pines es compatible con la Raspberry Pi), de una Raspberry Pi y de un router Wifi que hace de punto de acceso entre la Raspberry Pi e internet.

2.1. Montaje de red

La Raspberry Pi, por su gran capacidad de cómputo, hace de centro de datos. Recibe información del Arduino Due a través de el puerto serie, y lo pone en internet a través de un puerto Ethernet.

2.2. Protocolo serie

El protocolo serie es muy simple, y se basa en transmitir unas tramas con la siguiente información (en ese orden):

- 4 bytes que codifican en ASCII hexadecimal el tamaño de la trama.
- 1 byte que codifica en ASCII el tipo del paquete. Puede ser
- Varios bytes con el payload de la trama, en ASCII. No pueden contener el byte 0.
- 1 byte nulo (valor 0) que inidica fin de la trama. Se utiliza para sincronizar en caso de error.

2.3. Código arduino

Lo primero que hemos hecho ha sido reutilizar parte del código de la primera práctica. En un bucle continuo, leemos los sensores de temperatura y luz y con la información obtenida mandamos frames por el protocolo serie a la Raspberry Pi.

Por otro lado, también nos encargamos de leer frames que puedan venir de la Rapsberry Pi. Si llega un frame de tipo 'L', le cambiamos el valor al *analog output* del led, haciendo que ilumine más o menos. O por si llega un frame de tipo 'D', en cuyo caso limpiamos el display y ponemos el nuevo texto en la pantalla.

2.4. Servicio web

El servicio web tiene dos partes, un código escrito en python que corre en la Rapsberry PI y otro código escrito en javascript que se ejecuta en un teléfono móvil, o un pc.

2.4.1. Parte servidor de python

En esta parte, se espera a que un navegador web estándar haga una petición de tipo GET para obtener el documento index.html. Además, está a la espera de recibir peticiones GET al documento 'luz' o al documento 'tmp'.

Cada vez que recibe una petición al documento luz, lee las tramas a través del puerto serie para obtener del Arduino la última lectura del sensor de luz. Después, contesta la petición con el valor del sensor. Lo mismo sucede si se recibe una petición al documento tmp, solo que se devuelve la temperatura.

Por otro lado, el servidor de python también está a la espera de las peticiones PUT 'led' y 'display'. En este caso, en vez de devolver un valor leyendo por el puerto serie, lo que hace es escribir en el puerto serie una trama de tipo 'L' (si led) o de tipo 'D' (si display) para alterar el brillo del led o el texto del monitor LCD.

2.4.2. Parte cliente de javascript

En el explorador web, se ejecuta continuamente una petición de tipo GET 'luz' y GET 'tmp', para actualizar en tiempo real la oscuridad del fondo (dependiendo del nivel de luz), o el valor de un número dependiendo de la temperatura.

También se pone una barra de desplazamiento para modificar el brillo del led, y una barra de texto para cambiar el mensaje del display. Cuando se mueve la barra de desplazamiento se ejecuta un PUT 'led', y cuando se cambia el texto en la barra se ejecuta un PUT 'display', que llegan al servidor web de python.

3. Notas adicionales

3.1. Aproximación a la temperatura real

3.2. Lector de pulsaciones