

# LED API

**Diego Martín Arroyo**

Viernes, 8 de mayo de 2015

## Índice

<b>Introducción</b>	<b>3</b>
<b>Hardware</b>	<b>3</b>
<b>Software</b>	<b>4</b>

## Introducción

Uno de los componentes clave del sistema es su habilidad para facilitar el estudio de los sistemas distribuidos y la comprensión de los conceptos más básicos de este tipo de arquitecturas, así facilitar el desarrollo de aplicaciones distribuidas gracias a la intuitividad del sistema.

En este contexto surge la API de LEDs, un pequeño conjunto de sistemas *software* y *hardware* que permiten identificar de forma sencilla el comportamiento de un algoritmo distribuido en el sistema, apoyándose en la conmutación de una serie de diodos situados en cada placa. Dichos diodos están disponibles para el programador y pueden ser utilizados en una gran variedad de contextos.

## Hardware

El conjunto de LEDs se sitúa en un circuito impreso que se conecta a la **Raspberry Pi** a través del puerto *GPIO* (*General Purpose Input-Output*), el cual provee de energía a los diferentes componentes.

Raspberry Pi2 GPIO Header					
Pin#	NAME		NAME	Pin#	
01	3.3v DC Power	●	DC Power 5v	02	
03	GPIO02 (SDA1, I <sup>2</sup> C)	●	DC Power 5v	04	
05	GPIO03 (SCL1, I <sup>2</sup> C)	●	Ground	06	
07	GPIO04 (GPIO_GCLK)	●	(TXD0) GPIO14	08	
09	Ground	●	(RXD0) GPIO15	10	
11	GPIO17 (GPIO_GEN0)	●	(GPIO_GEN1) GPIO18	12	
13	GPIO27 (GPIO_GEN2)	●	Ground	14	
15	GPIO22 (GPIO_GEN3)	●	(GPIO_GEN4) GPIO23	16	
17	3.3v DC Power	●	(GPIO_GEN5) GPIO24	18	
19	GPIO10 (SPI_MOSI)	●	Ground	20	
21	GPIO09 (SPI_MISO)	●	(GPIO_GEN6) GPIO25	22	
23	GPIO11 (SPI_CLK)	●	(SPI_CE0_N) GPIO08	24	
25	Ground	●	(SPI_CE1_N) GPIO07	26	
27	ID_SD (PC ID EEPROM)	●	(PC ID EEPROM) ID_SC	28	
29	GPIO05	●	Ground	30	
31	GPIO06	●	GPIO12	32	
33	GPIO13	●	Ground	34	
35	GPIO19	●	GPIO16	36	
37	GPIO26	●	GPIO20	38	
39	Ground	●	GPIO21	40	

Figura 1: Puertos disponibles en la placa. El total de puertos asciende a 40, de los cuales 17 son de utilidad para la API

La estructura de la placa divide claramente el conjunto de diodos disponibles en tres secciones, a detallarse en la sección *software*:

- **LEDs de estado** Accesibles únicamente por el administrador, permiten conocer el estado de la Raspberry (4 LEDs).
- **LEDs globales** Accesibles por el usuario, a través de un conjunto de aplicaciones definidas por el administrador, generalmente servicios globales (4 LEDs).
- **LEDs de usuario** Disponibles por completo para el programador una vez que ha conseguido acceso a los mismos (9 LEDs).

La placa se conecta directamente al puerto GPIO

## Software

El software utiliza una API estándar para el manejo del encendido y apagado, limitándose a la gestión de permisos de acceso. Dicha acción se controla con los siguientes parámetros:

- Dueño de la aplicación (root)
- Acceso garantizado a una serie de LEDs
- Acceso mediante *tokens*

## Tokens

La estructura de acceso a los LEDs de usuario se basa en la utilización de *tokens* o testigos, objetos que otorgan el privilegio de uso de un recurso. En este caso se añade una restricción atípica: el tiempo de uso. La API otorga tokens siguiendo una estructura **FIFO**, otorgando acceso a una aplicación, mientras que el resto de las mismas se bloquearán a la espera de que el token sea entregado.

Sin embargo, este mecanismo no describe cómo se realiza la sincronización entre diferentes nodos de la infraestructura. Es deseable que cuando una aplicación obtenga un token de autenticación sea capaz de utilizar los LEDs en toda la infraestructura. Para ello se aprovecha **MarcoPolo**.

Una secuencia de interacción válida es la siguiente:

1. Nodo 1 solicita acceso a LEDs
2. Nodo 1 envía dicha solicitud a todos los nodos partícipes (o a un subconjunto definido, según el deseo del usuario), indicando la reserva del recurso, utilizando en la detección *MarcoPolo*.
3. Nodo 2 recibe el mensaje. Sin embargo, está ocupado atendiendo otra petición, por lo que la añadirá a una cola de espera. Nodo 1 bloquea su ejecución hasta que Nodo 2 libere el recurso. Dichas solicitudes se realizan en paralelo, por lo que el bloqueo se produce únicamente en uno de los hilos, no en el resto.
4. Una vez que todos los nodos han cedido el acceso a la aplicación dada, se procede a la ejecución, que comienza a aprovechar la funcionalidad de los nodos.
5. Cuando la aplicación termina, el privilegio de acceso se cede a la siguiente aplicación, repitiendo el proceso.

La reserva se realiza en base a un LED en concreto, no al conjunto de todos ellos, por lo que en el caso de que cada usuario requiera únicamente un LED, podrían coexistir hasta 9 usuarios simultáneamente. Dicho sistema garantiza la alternancia de varios usuarios, permitiendo establecer un sistema de turnos justo (basado en **FIFO**).

## Referencias