
Dispositivo hardware de bajo coste para bancos de pruebas de desarrollo de drivers USB en Linux



TRABAJO FIN DE GRADO

Guillermo Gascón Celdrán y Javier Rodríguez-Avello Tapias

Dirigido por: Juan Carlos Sáez Alcaide

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid
Curso 2022-2023

Dispositivo hardware de bajo coste para bancos de pruebas de desarrollo de drivers USB en Linux

Memoria de Trabajo Fin de Grado

Guillermo Gascón Celdrán y Javier Rodríguez-Avello Tapias

Dirigido por: Juan Carlos Sáez Alcaide

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid
Curso 2022-2023

Resumen

El protocolo de comunicación USB, ampliamente usado en numerosos dispositivos en todo el mundo, resulta complejo a la hora de realizar modificaciones para determinados periféricos, sobre todo a la hora de realizar cambios en la programación del dispositivo en cuestión. En este proyecto se trabaja en la realización de un firmware para un chip determinado (ATTiny85) que sea capaz de realizar distintas funciones sobre periféricos, que trabajen con un driver cargado en Linux, todo ello programado en lenguaje C.

La complejidad de la tecnología/especificación USB unido al tiempo limitado que puede dedicarse a describir aspectos de bajo nivel de entrada-salida en las titulaciones de Informática, dificulta la adquisición de conocimientos sobre desarrollo de drivers USB, muy valorados en sectores estratégicos. Asimismo, para iniciarse en el desarrollo de este tipo de drivers es preciso disponer de hardware suficientemente sencillo y de bajo coste. Un ejemplo de dispositivo para iniciación es el dispositivo Blinkstick Strip, que cuenta con 8 LEDs de colores cuyo estado puede alterarse individualmente.

En este proyecto se diseña y programa un dispositivo USB de bajo coste que pueda usarse masivamente como banco de pruebas para el desarrollo de drivers USB. Dicho dispositivo estará formado por varios componentes (LEDs, pantallas LCD, sensores, etc.), y permitirá al desarrollador familiarizarse con distintos aspectos y modos de interacción con el hardware USB, para poder afrontar la complejidad de USB de forma gradual. Además del diseño del hardware, y la construcción de un prototipo del mismo empleando un microcontrolador, como el ATTiny85, será necesario el desarrollo del firmware del dispositivo así como de un conjunto de drivers de ejemplo (preferentemente en Linux), para exponer al usuario los distintos componentes del dispositivo USB.

palabras clave: Firmware, USB, endpoint, kernel Linux, report-ID, AVR, V-USB.

Abstract

The USB communication protocol, widely used in numerous devices around the world, is complex when making changes to certain peripherals, especially when making changes to the programming of the device in question. In this project we work on the realization of a firmware for a certain chip (ATTiny85) that is capable of performing different functions on peripherals, that work with a driver loaded in Linux, all programmed in C language.

The complexity of the USB technology/specification together with the limited time that can be devoted to describing low-level input-output aspects in Computer Science degrees, makes it difficult to acquire knowledge on USB driver development, highly valued in strategic sectors. Likewise, to start developing this type of drivers it is necessary to have sufficiently simple and low-cost hardware. An example of a starter device is the Blinkstick Strip device, which has 8 colored LEDs whose state can be individually altered.

In this project, a low-cost USB device is designed and programmed that can be used massively as a test bench for the development of USB drivers. Said device will be made up of several components (LEDs, LCD screens, sensors, etc.), and will allow the developer to become familiar with different aspects and modes of interaction with USB hardware, in order to gradually face the complexity of USB. In addition to the hardware design, and the construction of a prototype of it using a microcontroller, such as the ATTiny85, it will be necessary to develop the firmware of the device as well as a set of example drivers (preferably in Linux), to expose the user to the various components of the USB device.

Autorización de difusión y utilización

Los abajo firmantes, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado “Dispositivo hardware de bajo coste para bancos de pruebas de desarrollo de drivers USB en Linux”, realizado durante el curso académico 2022-2023 bajo la dirección de Juan Carlos Sáez Alcaide en el Departamento de Arquitectura de Computadores y Automática, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Guillermo Gascón Celdrán y Javier Rodríguez-Avello Tapias

Juan Carlos Sáez Alcaide

Índice general

Resumen	iii
Abstract	v
Autorización de difusión y utilización	i
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Plan de trabajo	2
1.4 Organización de la memoria	2
2 Introducción	3
Epílogo	5
Bibliografía	7

Índice de cuadros

Índice de figuras

2.1	Esto es el título	3
-----	-----------------------------	---

Capítulo 1

Introducción

En los siguientes apartados se explican los motivos de la realización del proyecto, los objetivos y la planificación de las tareas para llevarlo a cabo.

1.1 Motivación

El protocolo USB es ampliamente usado en todo el mundo para la comunicación con casi cualquier periférico. De forma sencilla, cualquier usuario con un ordenador puede conectar un dispositivo, da igual la funcionalidad que tenga, al puerto USB de su ordenador y éste reconocerlo para que pueda comunicarse con la CPU y realizar la función para la que ha sido desarrollado.

Para que todo esto se pueda dar, el protocolo USB tiene una complejidad enorme en cuanto a paquetes de comunicación entre la CPU y el dispositivo o la controladora USB, a parte de necesitar un driver instalado en el sistema operativo para que pueda interpretarlo. A pesar de lo sencillo que pueda parecer el hardware (2 cables de datos), hay muchos elementos en la comunicación USB que tienen lugar para que el dispositivo pueda ser reconocido.

Para poder estudiar todo el protocolo USB y la interacción entre el dispositivo y la CPU, en este proyecto hemos desarrollado un firmware en C que realiza una serie de funciones sobre una pantalla LCD y un sensor de temperatura, utilizando un microchip ATTiny85 sobre una placa Digispark, que contiene el propio controlador USB.

1.2 Objetivos

Uso del firmware como banco masivo de pruebas para otros proyectos que puedan utilizar la librería V-USB.

1.3 Plan de trabajo

Para el desarrollo de este proyecto, se han mantenido distintas reuniones con los directores del proyecto y entre los desarrolladores. En la parte inicial del proyecto, se ha estudiado el funcionamiento de la librería V-USB con distintos proyectos que utilizan otros microchips, para ver qué uso se hacen de los report-IDs y las distintas funciones empleadas en el código. Se ha utilizado el software Wireshark para estudiar los endpoints usados y los paquetes URBs en la comunicación, y así estudiar la configuración del software y la posibilidad de utilizar, por ejemplo, interrupciones.

Para el desarrollo de los periféricos que se van a usar conjuntamente en la placa Digispark, Guillermo se ha encargado de estudiar y desarrollar el código correspondiente al sensor de temperatura y Javier al correspondiente con la pantalla LCD 2x16.

Para la integración del código, se ha utilizado GitHub.

1.4 Organización de la memoria

Capítulo 2

Introducción

Esto es solo un ejemplo de capítulo. Vamos a citar algo ??? [1], [2].



Figura 2.1: Esto es el título

- UNO
- DOS
- TRES

OK	A	B
AHSAJHD	ASHJDHAJSH ^o	SAHJDHJASH
SADHJHJ	ASHJDFHJASH	ASDASD

```
#ifdef CONFIG_PMC_PERF
    int safety_control;
    unsigned char prof_enabled;
#endif
```


Epílogo

Esto es solo un ejemplo de apéndice.

Bibliografía

[1] «V-USB open-source Project example projects for LCD Display». <https://www.obdev.at/products/vusb/projects.html>

[2] «Source code for 4x20 LCD Display». <https://github.com/harbaum/LCD2USB>

[1] W. Maurer, *Professional Linux Kernel Architecture*. Birmingham, UK, UK: Wrox Press Ltd., 2008.

[2] ARM, «AMBA(R) AXI(TM) and ACE(TM) Protocol Specification». <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ih0022e/index.html>, 2011.