

Capítulo 1

Elección de las herramientas para la realización de la interfaz

La implementación de la comunicación entre el FPGA y la PC, en este trabajo, se efectúa a través de USB 2.0. Para establecer esta comunicación, como se menciona en la Sección ??, se utiliza una interfaz que ejecute las tareas relativas al protocolo, la transmisión y recepción de los datos.

La figura 1.1 muestra en líneas de trazo tres partes principales en las que este trabajo descompone el problema: La comunicación entre el FPGA y la interfaz, la configuración de la interfaz misma y la conexión entre la PC y la interfaz. El desarrollo de cada etapa cuenta con herramientas específicas que facilitan en gran medida la tarea que se realiza. En este capítulo se detalla por separado las características de cada una de estas herramientas.

1.1. Interfaz

La parte central del presente trabajo está constituida por el módulo de interfaz entre el FPGA y la PC, que es el que permite cumplir el objetivo de proveerle al sistema la comunicación, es decir, que cualquier dispositivo implementado con FPGA pueda recibir y transmitir datos a través del protocolo USB desde y hacia una PC, respectivamente.

La interfaz está constituida por el controlador EZ-USB FX2LP fabricado por Cypress Semiconductor, un circuito integrado que posee en su interior un microcontrolador 8051, con

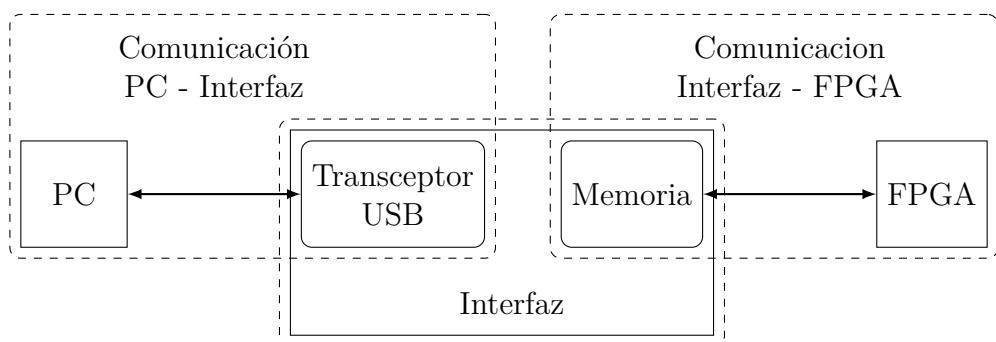


Figura 1.1: Partes en que se desglosa el trabajo



Figura 1.2: Circuito impreso principal del kit de desarrollo CY3684 EZ-USB FX2LP

algunas mejoras destinadas a satisfacer mejor los requerimientos del sistema USB; una interfaz serie, que permite ingresar datos uno tras otro y los entrega en forma paralela y viceversa; un transceptor USB encargado de todas las tareas de codificación y decodificación de paquetes USB; memoria RAM para programas y datos de 16 kB. Posee, a su vez, tres tipos de interfaces hacia periféricos externos: I²C, una memoria FIFO (Primero Entrado, Primero Salido, del inglés *First In First Out*) destinada a sistemas con poder de iniciativa para escribir y leer datos, y un sistema de propósito general que puede ser comandado a través del 8051[30].

El controlador viene montado en un circuito impreso que posee una serie de componentes adicionales que facilitan la interacción del desarrollador, tales como pulsadores, display de 7 segmentos, módulos de memoria adicional, etc. Este tipo de circuitos impresos armados con la intención de favorecer desarrollo de otros sistemas, se denomina placa de desarrollo. Una placa de desarrollo que, además, incorpora algunas herramientas extra como software, cables de conexión, fuentes, etc. toma el nombre de kit de desarrollo.

En este trabajo, se utiliza el kit de desarrollo CY 3684 EZ-USB FX2LP, fabricado por Cypress Semiconductor[31]. El kit posee una placa de desarrollo como la que se observa en la Figura 1.2. El componente principal del kit es el controlador EZ-USB FX2LP e incorpora un display de 7 segmentos, 4 luces led multipropósito, 6 pulsadores, de los cuales 4 son de propósito general, uno de reinicio y otro que envía una señal especial para salir de un modo de bajo consumo. También tiene dos bloques de memorias EEPROM destinadas al almacenamiento del firmware (programa que ejecuta un microcontrolador), lo que otorga la posibilidad de realizar una carga no volátil de la configuración del controlador, memoria flash de 64 kB utilizados como RAM por el programa del controlador, un puerto USB y dos puertos UART con zócalos DE-9.

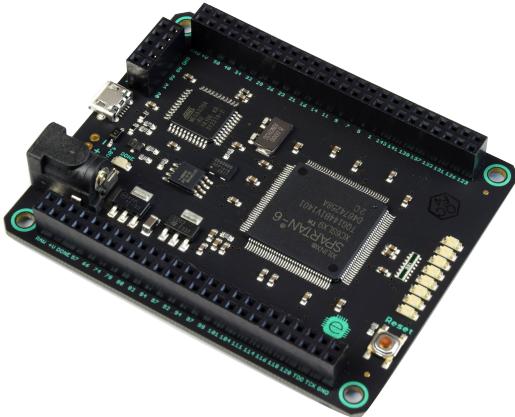


Figura 1.3: Placa de prototipado rápido MOJO v3, diseñada por Embedded Micro

Adicionalmente, cuenta con 6 puertos de 20 pines que permiten comunicarse con el controlador y 1 puerto de 40 pines, compatible con el protocolo ATA.

Se selecciona este controlador como interfaz ya que cuenta con una gran cantidad de herramientas que permiten realizar la comunicación USB, además de poseer memoria suficiente para datos y una interfaz de comunicación para periféricos simple lo que facilita el objetivo de utilizar la menor cantidad de los recursos configurables del FPGA, de forma tal que queden, estos últimos, disponibles para el desarrollo de otros sistemas.

1.2. Elección de la FPGA

En la implementación de una comunicación, para poder transmitir y recibir datos, los componentes que intervienen deben seguir un protocolo establecido, de forma tal que cada dispositivo sepa qué procedimiento efectuar. Por este motivo, una vez definido qué se utiliza una interfaz intermedia entre la PC y un FPGA y que dicha interfaz es el circuito integrado EZ-USB FX2LP de Cypress, se deberá configurar un FPGA para que reciba y envíe datos a la interfaz.

Se elige para este trabajo, la placa de desarrollo MOJO v3 desarrollada por la empresa Embedded Micro. Esta placa, la cual se observa en la Figura 1.3, posee un FPGA Spartan-6 de Xilinx. El FPGA brinda la posibilidad de elaborar sistemas digitales complejos de alta velocidad y permite, al desarrollador de sensores y sistemas de adquisición de datos, la síntesis de circuitos que resuelvan problemas en la medida de los requerimientos. Dispone también de 84 puertos digitales configurables como entrada y/o salida, 8 entradas analógicas, 8 LED's de propósito general, un botón de tipo pulsador.

La placa MOJO es una placa de desarrollo muy económica para prototipado, es decir, la fabricación de modelos funcionales. Para ello los puertos se disponen en un arreglo de pines a través de los cuales es posible acoplar el dispositivo que sea necesario. Se dispone en el mercado de otros circuitos impresos que se conectan a los pines y contienen un grupo de periféricos para propósito general. Estos impresos, se denominan shields (*escudo* traducido al castellano). se obtiene así una placa de desarrollo a la medida de las necesidades de cada proyecto. El usuario también puede diseñar sus shields o conectar las entradas y salidas de otros dispositivo mediante

cables.

Además de los shields, los diseñadores pensaron en que no sea necesario ninguna herramienta extra a la hora de programar la FPGA. Para ello, dotaron al sistema de un microcontrolador ATmega32U4 de Atmel con un programa de tipo bootloader, que se encarga de transferir la configuración del FPGA cargada desde una memoria flash incorporada, o transmitida por el usuario desde una PC, a través de un transceptor USB que contiene el microcontrolador. Luego, el controlador es colocado en modo esclavo y se configura de forma tal que dota al sistema de una comunicación entre la FPGA y una PC, vía USB y se utiliza su ADC para leer los puertos analógicos.

Una vez llegado a este punto, el lector podría preguntar con toda razón ¿por qué es necesario realizar un sistema de comunicación USB extra, si ya cuenta con un microcontrolador que se encarga de dicho asunto? La respuesta se basa en el ancho de banda del sistema de comunicación que dispone la placa. La línea de controladores ATmega incorpora puertos USB 2.0 full-speed. Esto quiere decir que puede enviar datos a una tasa de 12 Mbit s^{-1} . Además, la comunicación entre ambos chips se realiza vía SPI (*Serial Peripheral Interface*, o en español Interfaz Serie de Periféricos), comandada por un cristal de cuarzo de 50 MHz, ofreciendo una velocidad de salida que puede resultar insuficiente a los fines de este trabajo. Se pretende dotar al sistema del mayor ancho de banda posible, utilizando la capacidad de USB 2.0 High-Speed, de hasta 480 Mbps.

La placa de desarrollo Mojo fue seleccionada debido a su bajo costo, la versatilidad que ofrece y que está dotada por un Spartan-VI de Xilinx, que es un FPGA con una buena relación entre recursos, rendimiento, velocidad y precio.

1.3. Elección de la biblioteca libusb-1.0

La tercera parte en la que se divide el trabajo es relativa a la comunicación entre la interfaz y una PC. Ya que la interfaz se encarga en gran medida de lo relativo al empaquetamiento, codificación y decodificación y que las PC, por su parte, vienen equipadas con el hardware necesario, este trabajo debe implementar el software que comande y gestione, desde el sistema operativo el correcto acceso a los datos que se envían y reciben. Para la elaboración de software que permita el manejo de los puertos USB, se utiliza la biblioteca **libusb**.

libusb es una biblioteca de código abierto, muy bien documentada, escrita en C, que brinda acceso genérico a dispositivos USB. Las características de diseño que persigue el equipo de desarrollo que mantiene la biblioteca es que sea multiplataforma, modo usuario y agnóstico de versión[32].

- **Multiplataforma:** Se apunta a que cualquier software que contenga esta biblioteca pueda ser compilado y ejecutado en la mayor cantidad de plataformas posibles, dotando al software de portabilidad, es decir, esta biblioteca puede ser ejecutada en Windows, Linux, OS X, Android y otras plataformas sin necesidad de realizar cambios en el código.
- **Modo usuario:** No se requiere acceso privilegiado de ningún tipo para poder ejecutar programas escritos con esta biblioteca.
- **Agnóstico de versión:** Sin importar la versión de la norma USB que se utilice, el programa se podrá comunicar siempre con el dispositivo USB que se requiera.

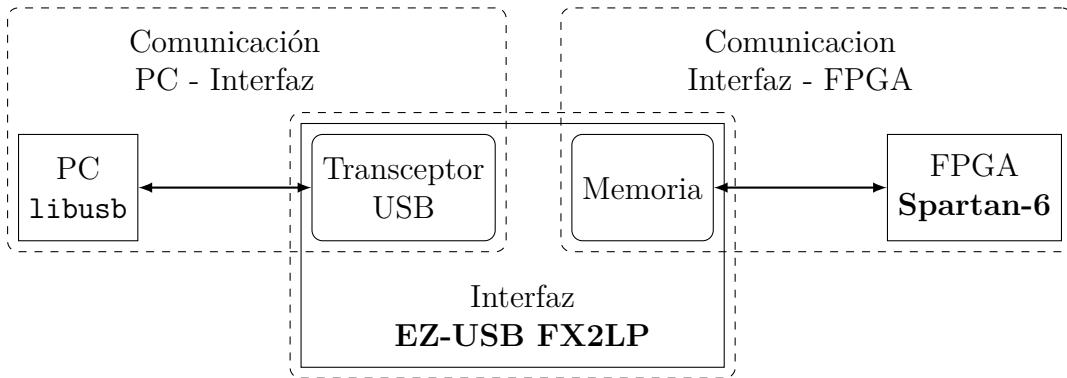


Figura 1.4: Partes en que se desglosa el trabajo con sus componentes

La biblioteca `libusb` no posee un autor formal. Es decir, no hay una persona, empresa u organización formal que se encargue de la creación y el mantenimiento del software. Existe una comunidad de más de 130 desarrolladores que en forma voluntaria cooperan en el mantenimiento y desarrollo de esta biblioteca. Se garantiza así que el proyecto esté documentado en forma detallada, existiendo amplios ejemplos y tutoriales de su uso.

Se elige esta biblioteca para la realización del software que gestionara el envío y la recepción de datos debido a su amplio soporte, la factibilidad de ejecutarlo en diferentes sistemas operativos y por ser totalmente gratuito.

1.4. Sumario del capítulo

En este capítulo se resumen las partes y que herramienta compone cada una de ellas. Tal como se observa en la Figura 1.4, el programa de PC que gestionará la comunicación desde ese extremo se realiza en lenguaje C, con la biblioteca `libusb`. Como interfaz, se utiliza el controlador EZ-USB FX2LP, que viene montado en la placa de desarrollo CY3684 EZ-USB FX2LP de Cypress Semiconductor. Finalmente, desde el extremo del FPGA, se utilizará un Spartan-6 de Xilinx, a través de la placa de desarrollo Mojo v3.

Bibliografía

- [1] R. Pallàs-Areny and J. G. Webster, *Sensors and signal conditioning*. Wiley-Interscience, 2001.
- [2] D. M. Considine, *Encyclopedia of instrumentation and control*. McGraw-Hill, Inc., 1971.
- [3] A. Perez Garcia, “Curso de instrumentación,” p. 261, 2008.
- [4] J. Fraden, *Handbook of modern sensors: physics, designs, and applications*. New York, NY: Springer New York, 2010.
- [5] E. Slawiński and V. Mut, *Humanos y máquinas inteligentes: conocimiento educativo sobre el comportamiento interno de robots que actúan juno y para el hombre*. Saarbrücken, Alemania: Editorial Académica Española, 2011.
- [6] K. Ogata, *Modern control engineering*. Aeeizh, 2002.
- [7] G. Binnig and H. Rohrer, “Scanning tunneling microscopy,” *Surface Science*, vol. 126, pp. 236–244, mar 1983.
- [8] R. Turchetta, K. R. Spring, and M. W. Davidson, “Digital Imaging in Optical Microscopy - Introduction to CMOS Image Sensors,” (accessed in July 2019).
- [9] S. Mendis, S. Kemeny, and E. Fossum, “CMOS active pixel image sensor,” *IEEE Transactions on Electron Devices*, vol. 41, pp. 452–453, mar 1994.
- [10] C. Hu-Guo, J. Baudot, G. Bertolone, A. Besson, A. S. Brogna, C. Colledani, G. Claus, R. D. Masi, Y. Degerli, A. Dorokhov, G. Doziere, W. Dulinski, X. Fang, M. Gelin, M. Goffe, F. Guilloux, A. Himmi, K. Jaaskelainen, M. Koziel, F. Morel, F. Orsini, M. Specht, Q. Sun, I. Valin, and M. Winter, “CMOS pixel sensor development: a fast read-out architecture with integrated zero suppression,” *Journal of Instrumentation*, vol. 4, pp. P04012–P04012, apr 2009.
- [11] J. Baudot, G. Bertolone, A. Brogna, G. Claus, C. Colledani, Y. Değerli, R. De Masi, A. Dorokhov, G. Dozière, W. Dulinski, M. Gelin, M. Goffe, A. Himmi, F. Guilloux, C. Hu-Guo, K. Jaaskelainen, M. Koziel, F. Morel, F. Orsini, M. Specht, I. Valin, G. Voutsinas, and M. Winter, “First test results of MIMOSA-26, a fast CMOS sensor with integrated zero suppression and digitized output,” *IEEE Nuclear Science Symposium Conference Record*, pp. 1169–1173, 2009.

BIBLIOGRAFÍA

- [12] M. Pérez, J. Lipovetzky, M. Sofo Haro, I. Sidelnik, J. J. Blostein, F. Alcalde Bessia, and M. G. Berisso, “Particle detection and classification using commercial off the shelf CMOS image sensors,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 827, pp. 171–180, aug 2016.
- [13] M. Pérez, J. J. Blostein, F. A. Bessia, A. Tartaglione, I. Sidelnik, M. S. Haro, S. Suárez, M. L. Gimenez, M. G. Berisso, and J. Lipovetzky, “Thermal neutron detector based on COTS CMOS imagers and a conversion layer containing Gadolinium,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 893, pp. 157–163, jun 2018.
- [14] C. L. Galimberti, F. Alcalde Bessia, M. Perez, M. G. Berisso, M. Sofo Haro, I. Sidelnik, J. Blostein, H. Asorey, and J. Lipovetzky, “A Low Cost Environmental Ionizing Radiation Detector Based on COTS CMOS Image Sensors,” in *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, pp. 1–6, IEEE, jun 2018.
- [15] T. Hizawa, J. Matsuo, T. Ishida, H. Takao, H. Abe, K. Sawada, and M. Ishida, “ 32×32 pH image sensors for real time observation of biochemical phenomena,” *TRANSDUCERS and EUROSENSORS '07 - 4th International Conference on Solid-State Sensors, Actuators and Microsystems*, pp. 1311–1312, 2007.
- [16] ON Semiconductor, “NOIP1SN0300A Global Shutter CMOS Image Sensors,” 2014.
- [17] N. Ida, *Engineering Electromagnetics*. Cham: Springer International Publishing, 3th ed., 2015.
- [18] J. F. Wakerly, *Digital Design: principles and practices*, vol. 1. Pearson, 1999.
- [19] M. Perez, F. Alcalde, M. S. Haro, I. Sidelnik, J. J. Blostein, M. G. Berisso, and J. Lipovetzky, “Implementation of an ionizing radiation detector based on a FPGA-controlled COTS CMOS image sensor,” in *2017 XVII Workshop on Information Processing and Control (RPIC)*, pp. 1–6, IEEE, sep 2017.
- [20] R. Biswas, *An Embedded Solution for JPEG 2000 Image Compression Based Back-end for Ultrasonography System*. PhD thesis, IIT, Kharagpur, 2018.
- [21] T. Yanagisawa, T. Ikenaga, Y. Sugimoto, K. Kawatsu, M. Yoshikawa, S.-i. Okumura, and T. Ito, “New NEO search technology using small telescopes and FPGA,” in *2018 IEEE Aerospace Conference*, vol. 2018-March, pp. 1–7, IEEE, mar 2018.
- [22] H. H. Goldstine and A. Goldstine, “The Electronic Numerical Integrator and Computer (ENIAC),” *Mathematical Tables and Other Aids to Computation*, vol. 2, p. 97, jul 1946.
- [23] S. of Cable Telecommuniocations Engineers, *American National Standard ANSI/SCTE 07 2006. Digital Transmission Standard for Cable Television*. Society of Cable Telecommunications Engineers, Inc., 2006.
- [24] I. Micron Technology, “1 / 2-Inch Megapixel CMOS Digital Image Sensor MT9M001C12STM (Monochrome),” pp. 1–35, 2004.

BIBLIOGRAFÍA

- [25] IEEE Computer Society, *IEEE Standard for Ethernet*, vol. 2018. 2018.
- [26] IEEE Computer Society, *Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications IEEE Computer Society Specific requirements Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, vol. 2012. 2016.
- [27] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, and Philips, *Universal Serial Bus Specification*, vol. Revision 2.0. 2000.
- [28] “Usb hardware.” https://en.wikipedia.org/wiki/USB_hardware. Ingreso: 8 de agosto del 2019.
- [29] T. Riihonen, *Desing and analysis of duplexing Modes and Forwarding Protocols for OFDM(A) Relay Links*. PhD thesis, 2015.
- [30] Cypress Semiconductor, “EZ-USB ® Technical Reference Manual,” tech. rep., 2014.
- [31] Cypress Semiconductor, “CY3684/CY3684 EZ-USB Development Kit User Guide,” tech. rep., 2014.
- [32] libusb, “libusb 1.0 <https://libusb.info/> - acceso: 04/11/2019.”