

## Analizador de Protocolo SPI

José Enrique Dáger Pacheco <sup>(1)</sup>, Diego Xavier Villegas Redrobán <sup>(2)</sup>, Ing. Ronald Ponguillo <sup>(3)</sup>  
Facultad de Ingeniería en Electricidad y Computación <sup>(1)(2)(3)</sup>  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 vía Perimetral  
Apartado 09-01-5863. Guayaquil-Ecuador  
joendage@espol.edu.ec <sup>(1)</sup>, dixavill@espol.edu.ec <sup>(2)</sup>, rponguil@espol.edu.ec <sup>(3)</sup>

### Resumen

*En el presente proyecto “Analizador de protocolo SPI”, se creó un sistema embebido basado en el procesador NIOS II y bloques de lógica configurable que cumple con las funcionalidades necesarias para conectarse con dispositivos que usan protocolo SPI, y para recolectar tramas de comunicación de dispositivos que se comunican con éste protocolo. Para hacer éste sistema, se usó una FPGA CYCLONE IV, que viene embebida en una tarjeta de educación y desarrollo de ALTERA, la DE0-nano. La parte del hardware del sistema se la diseñó e implementó usando la herramienta de QUARTUS II 12.1, QSYS, que es una nueva herramienta de integración de sistemas, y el software se lo implementó con la herramienta NIOS II IDE, que es un entorno de software desarrollado para el procesador NIOS II. Realizamos pruebas en módulos SPI para verificar el funcionamiento del proyecto. El caso base para nuestro estudio fue para un maestro y un esclavo que usan SPI, y la implementación de las pruebas se las realizó en un ambiente de laboratorio.*

**Palabras Claves:** Analizador de protocolo SPI, Sistema embebido, FPGA CYCLONE IV, NIOS II, SPI, DE0-nano.

### Abstract

*In this project “SPI protocol analyzer”, was creates an embedded system based on the NIOS II processor and configurable logics blocks that perform the necessary functionalities to connect with devices that utilize SPI protocol, and to recollect communications frames in devices that use this protocol. To make this system we use one FPGA CYCLONE IV, that is embedded in a development and education card of ALTERA, the DE0-nano. The hardware was designed and implemented using QUARTUS II 12.1, QSYS, that is a new tool for creation of systems, and the software was implemented with NIOS II IDE, that is an environment for software development for the NIOS II processor. We made tests in SPI modules to verify the functionality of the project. The base case for our study was one master and one slave that use SPI protocol, and the implementation of the tests was made in a laboratory environment.*

**Keywords:** SPI protocol analyzer, Embedded system, FPGA CYCLONE IV, NIOS II, SPI, DE0-nano.

### 1. Introducción

En estos tiempos, las comunicaciones digitales han avanzado de tal manera que se han desarrollado innumerables protocolos de comunicación que se utilizan en distintas aplicaciones, pero con un objetivo en común, hacer más fáciles las cosas y mejorar la comunicación entre uno u otro dispositivo.

La importancia de analizar la información que se está procesando en cualquier tipo de aplicación es un problema básico en la ingeniería en general, para monitoreo y control, detectar errores, comparar valores prácticos con valores teóricos, etc.

Para esto se han creado instrumentos que nos permiten analizar la información, por ejemplo, para verificar si un amplificador de voltaje está

amplificando como se espera, utilizamos un voltímetro que nos permite medir tanto el voltaje de entrada como el de salida.

En comunicaciones, para monitorear los datos que se procesan son los correctos, utilizamos un analizador de protocolo.

En nuestro caso construimos un Analizador de Protocolo SPI. Para éste objetivo utilizamos una tarjeta de educación y desarrollo de ALTERA, la DE0-nano, que tiene embebida una FPGA que puede ser programada o configurada de acuerdo a nuestras necesidades.

## 2. Descripción del proyecto

### 2.1. Objetivo general

Construir un sistema embebido basado en el microprocesador NIOS II y bloques de lógica configurable que permita recolectar tramas en dispositivos que utilizan protocolo SPI.

### 2.2. Objetivos específicos

- Entender el funcionamiento del protocolo SPI.
- Crear un hardware utilizando la herramienta QSYS de QUARTUS II 12.1, que nos permita conectarnos con dispositivos que utilizan protocolo SPI.
- Crear un software utilizando la herramienta NIOS II IDE que nos permita recolectar y analizar tramas de comunicación en dispositivos que se comunican con protocolo SPI.
- Realizar pruebas para comprobar el funcionamiento del proyecto.

### 2.3. Alcance y limitaciones

El proyecto cumple con las siguientes características:

- Construimos un Analizador de Protocolo SPI que nos permite analizar tramas de comunicación en cualquier tipo de dispositivo que esté usando éste protocolo.
- Nuestro analizador tiene 4 entradas, las cuales corresponden a las líneas MOSI, MISO, selector de esclavo y reloj, éstas líneas son propias del protocolo SPI, además tiene una entrada adicional que corresponde a la conexión de tierra.
- El caso base para nuestro estudio es un maestro y un esclavo.
- La implementación de las pruebas de éste proyecto se hicieron en un ambiente de laboratorio.

### 2.4. Identificación del problema

Cuando se requiere analizar, monitorear la información que se está enviando entre dispositivos, o cuando se requiere detectar errores los cuales se evidencian cuando nuestra aplicación no responde como se espera, es importante tener algún tipo de dispositivo que nos permita visualizar la información que se está procesando.

Para nuestro caso, el problema es hacer lo anteriormente descrito, con dispositivos que utilizan protocolo SPI.

### 2.5. Descripción breve de la solución del problema

Debido a las necesidades descritas lo que se hace es utilizar un Analizador de Protocolo, en nuestro caso construimos un Analizador de Protocolo SPI con la tarjeta de desarrollo y educación de Altera, DE0-nano.

## 3. Marco teórico

### 3.1. DE0-nano

La DE0-nano de altera introduce una plataforma de desarrollo de FPGA de tamaño compacto, adecuado para diseños tales como robots y proyectos portables.

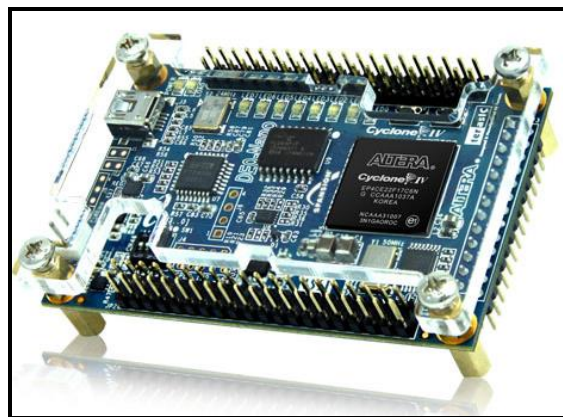


Figura 1. Tarjeta DE0-nano

Entre sus principales componentes tenemos:

- Una FPGA CYCLONE IV
- Un puerto USB-BLASTER para programar la tarjeta.
- 3 conjuntos de conectores de expansión para realizar diseños fuera de la tarjeta.
- Periféricos generales tales como LED's, botoneras, interruptores, que nos permiten realizar distintos procedimientos dentro de la tarjeta.

### 3.2. FPGA CYCLONE IV

Entre las principales características de la familia de dispositivos CYCLONE IV tenemos:

- Bajo costo y bajo consumo de potencia.
- De 6K a 150K elementos lógicos.

- Hasta 594 Kbits de bloques de memoria embebida.
- Hasta 360 multiplicadores 18x18 para aplicaciones de procesamiento DSP.

Los dispositivos CYCLONE IV E ofrecen un amplio rango de protocolos:

- Puertos Display.
- Hasta 532 puertos de E/S.
- Soporte para interfaces DDR2 SDRAM de 200 MHz o más.
- Soporte para QDRII SRAM y DDR SDRAM de 167 MHz o más.
- Ofrecido en ambientes comerciales e industriales.

### 3.3. Microcontrolador basado en NIOS II

Un microcontrolador basado en NIOS II está compuesto por:

- El núcleo procesador NIOS II.
- Memoria interna de programa y de datos.
- Periféricos integrados.
- Interfaces de entrada y salida.

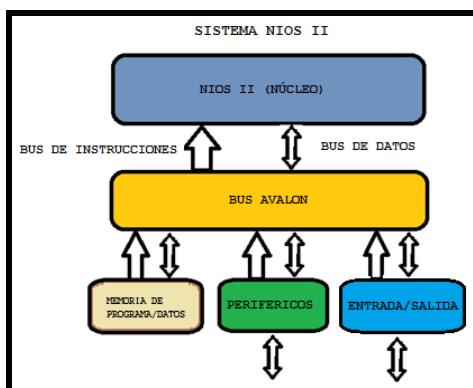


Figura 2. Microcontrolador basado en NIOS II.

NIOS II es un procesador de 32 bits de propósito general diseñado por ALTERA, es muy versátil porque puede ser usado con una variedad de otros componentes e incorpora muchas mejoras en comparación con la arquitectura NIOS II original, haciéndola más adecuada para aplicaciones embebidas computarizadas de mayor complejidad.

### 3.4. QUARTUS II 12.1

Éste software incluye el procesador NIOS II y la herramienta de creación QSYS, para facilitar la implementación de sistemas complejos.

QUARTUS II 12.1 nos permite realizar varias tareas al momento de crear un circuito lógico, desde la descripción de los componentes del circuito en lenguaje VHDL, hasta la simulación de diagramas de tiempo del comportamiento del circuito.

#### 3.4.1. QSYS

La herramienta QSYS de QUARTUS II 12.1 es una nueva herramienta de integración de sistemas, es la siguiente generación de la herramienta SOPC Builder, que nos ahorra tiempo y esfuerzo en el diseño de hardware en la FPGA.

QSYS nos brinda un mayor rendimiento, mejor reutilización de diseño y una verificación más rápida en comparación con SOPC Builder.

### 3.5. University Program

La herramienta University Program de ALTERA provee de un completo apoyo para introducir a los estudiantes a la tecnología digital. Éste tipo de apoyo consiste en hardware, software y material de enseñanza. El hardware consiste en máquinas que vienen pre configuradas para las tarjetas de desarrollo y educación de Altera (DE0-nano, DE1, DE2, DE2-115, DE4).

### 3.6. NIOS II IDE

Es una interfaz de software desarrollado para el procesador NIOS II. Todas las tareas de desarrollo de software pueden ser realizadas dentro del IDE, incluyendo edición, creación y depuración de programas.

NIOS II IDE está basado en el marco de ECLIPSE IDE y en las herramientas de desarrollo de C/C++.

### 3.7. Protocolo SPI

SPI es una interfaz serial sincrónica de propósito general. Motorola lo creó en 1980 y es usado especialmente para permitir a los microcontroladores comunicarse con sus dispositivos periféricos.

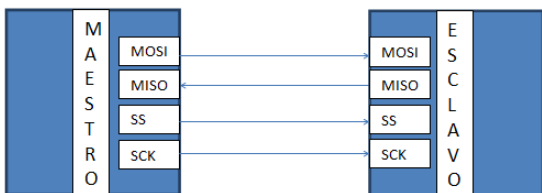
Los dispositivos se comunican utilizando una relación maestro-esclavo, en la cual el maestro inicia la trama de datos. Cuando el maestro genera el reloj y selecciona un dispositivo esclavo, los datos pueden ser enviados en una o ambas direcciones simultáneamente, es decir, es una comunicación full duplex.

El protocolo SPI es usado a distancias cortas o a nivel de PCB, debido que trabaja en altas frecuencias, entonces, a altas frecuencias y largas distancias interviene el factor ruido.

### 3.7.1. Señales del protocolo SPI

SPI especifica 4 señales:

- Master Output Slave Input (MOSI): Salida de datos del maestro hacia el esclavo.
- Master Input Slave Output (MISO): Salida de datos desde el esclavo hacia el maestro.
- Serial Clock (SCK): Reloj manejado por el maestro para sincronización de datos.
- Slave Select (SS): Selector de esclavo manejada por el maestro para escoger el esclavo con el que se va a trabajar.



**Figura 3.** Comunicación SPI entre un maestro y un esclavo

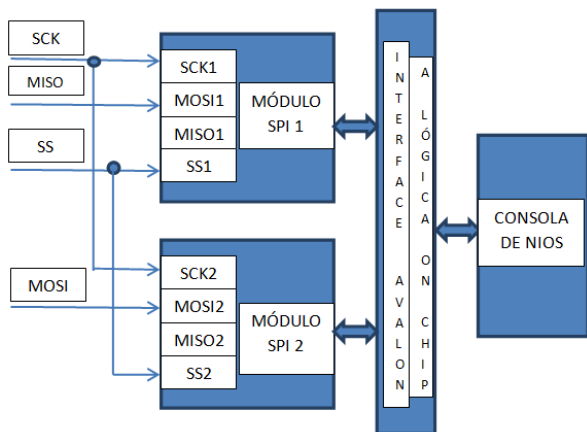
Con el selector de esclavo, el correspondiente dispositivo periférico es seleccionado. Ésta señal es activa en bajo, en el estado en que no está activo la señal MISO del esclavo está en alta impedancia.

## 4. Diseño e Implementación

### 4.1. Diseño e Implementación de Hardware

Con la herramienta de Quartus 12.1, QSYS, diseñamos en la FPGA un hardware que nos permite conectarnos con cualquier dispositivo que utilice protocolo SPI.

El siguiente diagrama de bloques muestra, de manera detallada, la configuración del hardware de nuestro analizador creado en QSYS.



**Figura 4.** Diagrama de Bloques del analizador de protocolo SPI.

La estrategia empleada para crear nuestro analizador es la siguiente:

- El analizador solo puede “escuchar” lo que se están diciendo entre maestro y esclavo, o sea, no va a enviar información, para esto necesitamos dos entradas MOSI, para conseguir esto se creó dos módulos SPI configurados como esclavo.
- Las líneas de dato (MOSI y MISO) de la configuración a analizar, deben estar conectadas a las entradas MOSI1 y MOSI2 que corresponden a los módulos SPI esclavos creados para hacer nuestro analizador
- Las dos salidas MISO del analizador no las usamos.
- Las líneas de control (relojes y los selectores de esclavo) de los 2 módulos SPI creados para hacer nuestro analizador deben estar conectados a las líneas de control (reloj y selector de esclavo) del sistema que vamos a analizar.
- La tierra de la configuración a analizar debe estar conectada a la tierra de nuestro analizador.
- Manipulamos por software los datos “escuchados” para imprimirlos en la consola de NIOS II IDE.

#### 4.1.1. Implementación de Hardware en QSYS

Para implementar nuestro hardware en QSYS, nos basamos en la computadora básica de la DE0 nano que viene preconfigurada cuando instalamos el University Program. Agregamos dos módulos SPI a la computadora básica, y luego quitamos todos los módulos que no vamos a usar, para que nos quede un hardware específico para analizar información entre dispositivos que usan protocolo SPI.

### 4.2. Diseño e Implementación de Software

Con la herramienta NIOS II IDE, diseñamos un software que nos permite analizar tramas de comunicación entre dispositivos que se comunican con protocolo SPI.

#### 4.2.1. Implementación de software en NOS II

Para implementar el software en NIOS II IDE, nos basamos en las funciones de las librerías que se crean cuando generamos la máquina en QSYS y compilamos en Quartus II.

Las librerías que usamos son:

- system.h
- io.h
- altera\_avalon\_spi.h
- altera\_avalon\_spi\_regs.h

Y las librerías estándar de c:

- stdio.h
- unistd.h
- ctype.h
- string.h

Con estas librerías creamos un software específico que nos permite recolectar los datos que se envía y/o reciben en dispositivos que utilizan protocolo SPI.

Para programar los dos módulos SPI de nuestro analizador seguimos la siguiente estrategia:

- Enceramos el registro STATUS.
- Leemos el registro STATUS.
- Chequeo si RRDY es 0, lo que significa que el registro RXDATA está vacío y está listo para leer datos.
- Leo el dato recibido.
- Imprimo en consola el dato recibido.

## 5. Resultados

En esta sección se muestran los resultados de la implementación del analizador de protocolo SPI. Los datos que son analizados pueden ser visualizados en la consola de NIOS II.

Se muestran los cuatro escenarios utilizados para realizar las pruebas y comprobar su funcionamiento.

### 5.1. Escenario A: Análisis de comunicación SPI en DE0 nano

Analizamos una comunicación SPI en una DE0 nano que tiene embebido un maestro y un esclavo que se comunican con protocolo SPI. Se espera que el analizador lea lo que se están enviando entre maestro y esclavo.

Se hace una comparación entre los resultados que se obtienen cuando se conecta el analizador a la tierra de la configuración con los resultados que se obtienen cuando no se hace esto, se espera que las lecturas sean más precisas cuando la tierra de la configuración a analizar está conectada a la tierra de nuestro analizador.

**Tabla 1.** Resultados del escenario A

	<b>Porcentaje promedio de error</b>
<b>Analizador no conectado a la tierra de la configuración analizada</b>	23.66%
<b>Analizador conectado a la tierra de la configuración analizada</b>	1.13%

### 5.2. Escenario B: Análisis de comunicación SPI en DE2-115

En éste escenario analizamos una comunicación SPI en una DE2-115 que tiene embebido un maestro y un esclavo que se comunican con protocolo SPI

**Tabla 2.** Resultados del escenario B

	<b>Porcentaje promedio de error</b>
<b>Analizador no conectado a la tierra de la configuración analizada</b>	25.33%
<b>Analizador conectado a la tierra de la configuración analizada</b>	4.66%

### 5.3. Escenario C: Análisis de comunicación SPI entre DE2-115 y DE0 nano

En éste escenario se analiza una comunicación SPI entre una DE2-115 configurada como maestro, con una DE0 nano configurada como esclavo

Se hace una comparación entre los resultados que se obtienen cuando tenemos una comunicación SPI a corta distancia con una comunicación SPI a larga distancia, se espera que las lecturas sean más precisas cuando tenemos una comunicación a corta distancia, dado que el protocolo SPI es más eficiente a cortas distancias.

**Tabla 3.** Resultados del escenario C

	<b>Porcentaje promedio de error</b>
<b>Analizador conectado a 40 cm</b>	7.66%
<b>Analizador conectado a 60 cm</b>	18.33%
<b>Analizador conectado a 80 cm</b>	25%

#### **5.4. Escenario D: Análisis de comunicación SPI entre PIC 16f887 y DE0 nano**

Analizamos una comunicación SPI entre un PIC 16F887 configurado como maestro y una DE0 nano que tiene embebido esclavo SPI.

**Tabla 4.** Resultados del escenario D

	<b>Porcentaje promedio de error</b>
<b>Analizador conectado a 30 cm</b>	7.33%
<b>Analizador conectado a 50 cm</b>	26.66%
<b>Analizador conectado a 70 cm</b>	33.33%

#### **6. Conclusiones**

- En base a los objetivos planteados, se pudo generar un sistema embebido, que cumple con las funcionalidades necesarias para analizar dispositivos que se comunican con protocolo SPI.
- Con el uso de la herramienta QSYS de QUARTUS II 12.1, se pudo desarrollar un hardware que nos permite conectarnos con dispositivos que se comunican con protocolo SPI, demostrando la eficacia de ésta herramienta para construir hardwares específicos para un proyecto determinado.

- Con el uso de la herramienta NIOS II IDE, se pudo desarrollar un software en lenguaje C, que nos permitió recolectar y analizar tramas de comunicación entre dispositivos que utilizan protocolo SPI, lo que demuestra la versatilidad de éste lenguaje de alto nivel para programar incluso sistemas embebidos.
- Comprendimos el protocolo SPI, que es un protocolo estándar en la industria, que lo utilizan microcontroladores y microprocesadores. Para hacer nuestro analizador, nos valimos de que cuando un dispositivo esclavo no está seleccionado, éste sólo “escucha” lo que se está enviando entre el maestro y el esclavo seleccionado, es decir se comporta como un espectador, y no “dice” algo, sólo “escucha”.
- El protocolo SPI es más eficiente a cortas distancias, debido a que trabaja a altas frecuencias, entonces a largas distancias y con frecuencias altas, el factor ruido interviene en la comunicación.

#### **7. Recomendaciones**

- Se recomienda tener un conocimiento de las librerías que se utilizan al momento de la programación sobre todo aquellas que interactúan con los registros del Bus Avalon, ya que es este el que permite conectar todos los elementos dentro de la FPGA y un mal manejo de estos registros puede llevar a un mal funcionamiento del proyecto al momento de ver los resultados.
- Se recomienda retirar de la máquina básica aquellos módulos que no van a ser utilizados, ya que el tamaño del circuito es un factor importante cuando se está desarrollando un prototipo, porque influye directamente en el precio al momento de la implementación.
- Se recomienda revisar los voltajes de alimentación con los que trabajan los dispositivos, ya que la alimentación que recibe cada uno de ellos puede ser diferente, lo que si bien no puede provocar daños en los dispositivos (DE0-Nano, 16F887), pero puede provocar errores de transmisión mostrando valores erróneos durante las pruebas.

## 8. Bibliografía

- [1] ALTERA Corporation, Cyclone IV FPGA Device Family, <http://www.altera.com/literature/hb/cyclone-iv/cyiv-51001.pdf>, fecha de publicación mayo 2013.
- [2] Altera Corporation, Nios II Processor Reference, [http://www.altera.com/literature/hb/nios2/n2cpu\\_ni\\_i5v1.pdf](http://www.altera.com/literature/hb/nios2/n2cpu_ni_i5v1.pdf), fecha de publicación mayo 2011.
- [3] Renesas Electronics Corporation, Serial Peripheral Interface (SPI) & Inter-IC (IC2) (SPI\_I2C), [http://documentation.renesas.com/doc/products/regi on/rtas/mpumcu/apn/spi\\_i2c.pdf](http://documentation.renesas.com/doc/products/regi on/rtas/mpumcu/apn/spi_i2c.pdf), fecha de publicación abril 2010.
- [4] Altera, Cyclone IV FPGA Family Overview, <http://www.altera.com/devices/fpga/cyclone-iv/overview/cyiv-overview.html>, Fecha de Consulta Abril 2013.
- [5] Altera, DE0-Nano User Manual, [ftp://ftp.altera.com/up/pub/Altera\\_Material/12.1/Boards/DE0-Nano/DE0\\_Nano\\_User\\_Manual.pdf](ftp://ftp.altera.com/up/pub/Altera_Material/12.1/Boards/DE0-Nano/DE0_Nano_User_Manual.pdf), Fecha de Consulta Abril 2013.
- [6] Altera Corporation, Nios II Processor Reference, [http://www.altera.com/literature/hb/nios2/n2cpu\\_ni\\_i5v1.pdf](http://www.altera.com/literature/hb/nios2/n2cpu_ni_i5v1.pdf), fecha de publicación mayo 2011.