UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Revista Vínculos
http://revistas.udistrital.edu.co/ojs/index.php/vinculos

Revista VÍNCULOS
Ciencia, Tecnología y Sociedad: Un Enlace hacia el Futuro

I+D Investigación y Desarrollo

Diseño e implementación de un sistema de transmisión vía TCPIP para un sensor de nivel

Design and implementation of a system for transmission via TCPIP for level sensory

Sebastian Enrique Jiménez Rojas[1] Jonathan Roberto Torres Castillo[2]

Para citar este artículo: S. E. Jiménez y J. R. Torres, "Diseño e implementación de un sistema de transmisión vía TCPIP para un sensor de nivel". *Revista Vínculos*, vol 14, no 1, enero-junio 2017, 17-26. doi: https://doi.org/10.14483/2322939X.13788.

Recibido: 23-10-2016 / Aprobado: 12-01-2017

Resumen

Esta investigación fue desarrollada para compensar la baja oferta de sensores que manejan el protocolo de comunicación TCPIP, y su alto costo en el mercado, por esto, se identifica la necesidad de realizar una nueva propuesta de un sensor industrial de menor costo. El proyecto fue desarrollado sobre una FPGA contenida en una tarjeta de desarrollo que cuenta con un puerto Ethernet y un procesador embebido instalado en ella, se establece la comunicación para transmitir los datos tomados por un sensor ultrasónico y enviarlos con el protocolo TCPIP.

Palabras clave: Instrumentación; plasma MIPS; protocolo TCP/IP; protocolo 802.3; sensores; sistemas embebidos.

Abstract

This research was developed to compensate for the low supply of sensors that handle the TCPIP communication protocol, and its high cost in the market, therefore, the need to make a new proposal for a lower cost industrial sensor is identified. The project was developed on an FPGA contained in a development card that has an Ethernet port and an embedded processor installed in it, communication is established to transmit the data taken by an ultrasonic sensor and send them with the TCPIP protocol

Keywords: Instrumentation; plasma Mip; TCP/IP protocol; protocol 802.3; sensor; ultrasonic.

1. Estudiante de Tecnología en electrónica. Universidad distrital Francisco José de Caldas. sejimenez_110@hotmail.com
2. Estudiante de Tecnología en electrónica. Universidad distrital Francisco José de Caldas. jonathanrtc@hotmail.com

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

1. Introducción

El monitoreo de máquinas, bodegas o almacenes de reserva en las empresas, juega un papel importante en la optimización de los procesos de producción en cuanto a la información oportuna de la cantidad o contenido de materia prima. La utilización de sensores para el caso del nivel de reserva de un contenedor, es la más común [1].

Partiendo de la adquisición de la señal proveniente del sensor ultrasónico SRF05 y por medio de una tarjeta de desarrollo basada en un FPGA, se procesaron los datos obtenidos mediante un sistema embebido desarrollado con el software Plasma Processor, el cual contiene las bases necesarias para enviar los datos por medio de los protocolos industriales de Ethernet y TCPIP. Para la aplicación y pruebas, se implementó el sensor en la máquina HAS-200, con el fin de controlar el nivel de reserva en uno de sus contenedores de granos y se desarrollo una interfaz de usuario, acorde a los requerimientos que se manejan en los procesos realizados por la máquina. De esta manera, se innova en un producto de menor costo y se mejora la oferta en las empresas que manejan este tipo de sensores en la automatización de sus procesos.

2. Plasma MIPS

Durante el desarrollo de los objetivos, se detectaron varios inconvenientes respecto al proceso de comunicación TCPIP desde la FPGA hacia un usuario remoto, puesto que el diseño en lenguaje VHDL de esta etapa del proyecto, se hacía demasiado extenso y además, cuando se generaba algún error, era mucho más difícil ubicar su origen.

Después de una investigación específica sobre la optimización del lenguaje y las nuevas tecnologías con FPGAs para mejorar el rendimiento de los procesos o funciones de estos dispositivos, se encontró una buena solución con la implementación de un sistema embebido.

Con la búsqueda de facilitar el desarrollo de la comunicación TCPIP, se decidió emplear estos sistemas embebidos para transmitir la información de la FPGA a un usuario remoto.

Como existen varios programas para el desarrollo de sistemas embebidos, se tuvo que profundizar sobre la compatibilidad, manejo de comandos y funciones prestadas por cada sistema de diseño, además de que permitiera adicionar los módulos necesarios para el manejo del puerto Ethernet de la tarjeta de desarrollo SPARTAN-3E Starter kit.

El sistema embebido que cumplió los requerimientos para el desarrollo de este objetivo fue el MIPS Plasma Processor por ser un software libre compatible con la tarjeta de desarrollo [2-4]. Plasma es un microprocesador (Figura 1) creado a base de software, diseñado inicialmente sin un control específico de sus periféricos y el cual permite la programación de sus módulos de entrada y salida según la aplicación que se le quiera dar.

Fue necesaria una basta documentación acerca del funcionamiento de procesador Plasma para asegurar un manejo correcto de las funciones y los comandos de implementación, y así empezar con las pruebas sobre la FPGA. Plasma está disponible en el sitio web de www.opencores.org, en el cual se puede encontrar información, archivos y ejemplos del procesador.

Para simplificar el proceso de programación del Plasma, el desarrollador (Steve Rhoads), diseñó una aplicación bootloader (gestor de inicio o arranque), que se encarga de dejar todo listo para la ejecución de un sistema y la cual permite programar el procesador desde un puerto serie de la tarjeta de desarrollo, simplemente transfiriendo un archivo *.bin (Extensión del archivo de programación) generado en el proceso; una vez transferido este archivo, el procesador pasa a funcionar desde la memoria externa DDR y queda listo para programar el microprocesador embebido instalado en la FPGA.

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

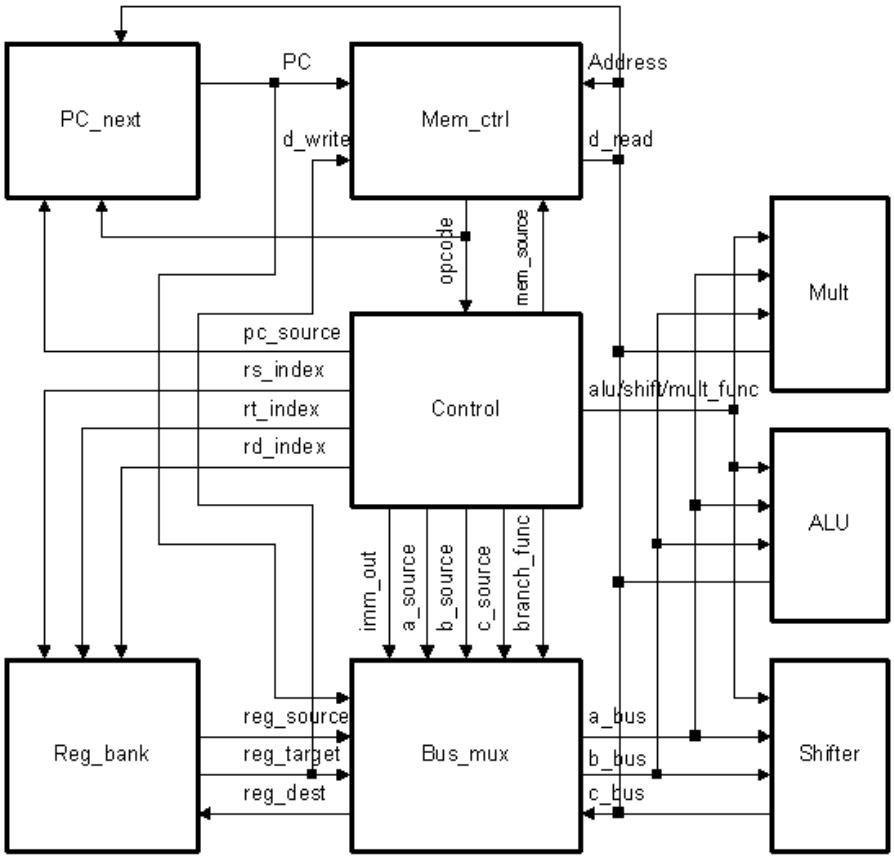

Figura 1. Esquema CPU plasma.
Fuente: elaboración propia.

2.1 Proceso de configuración de Plasma MIPS

Para la creación del bootloader, se cuenta con una estructura de archivos (plasma_latest.tar.gz y gcc_mipself), los cuales se pueden descargar de la página de opencores, contienen los programas necesarios para ensamblar, compilar, desensamblar, y generar los ejecutables para el procesador Plasma.

Las herramientas básicas dadas por Rhoads, se encuentran en el archivo gccmipself.zip mencionada anteriormente. Este archivo debe ser descomprimido en la carpeta Trunk y luego copiar los archivos convert_bin y ram_image que se encuentran en la carpeta tools_zip a la carpeta…trunk\gccmips_elf, logrando tener acceso por ventana de comandos a cada aplicación (Figura 2).

Ya que no se contaba con la información necesaria para el manejo de estas aplicaciones y creación del embebido, se desarrolló después de profundizar en la parte práctica, y de realizar varias pruebas con dichas aplicaciones, un tutorial paso a paso para un fácil entendimiento de estos procesos.

El proceso completo de creación de este bootloader se encuentra en el tutorial sobre Plasma MIPS, anexo a este documento.

3. Comunicación entre el sensor y la FPGA

La utilización de FPGAs es muy útil en cuanto al manejo de procesos, por su gran flexibilidad para trabajar en diferentes entornos y situaciones.

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

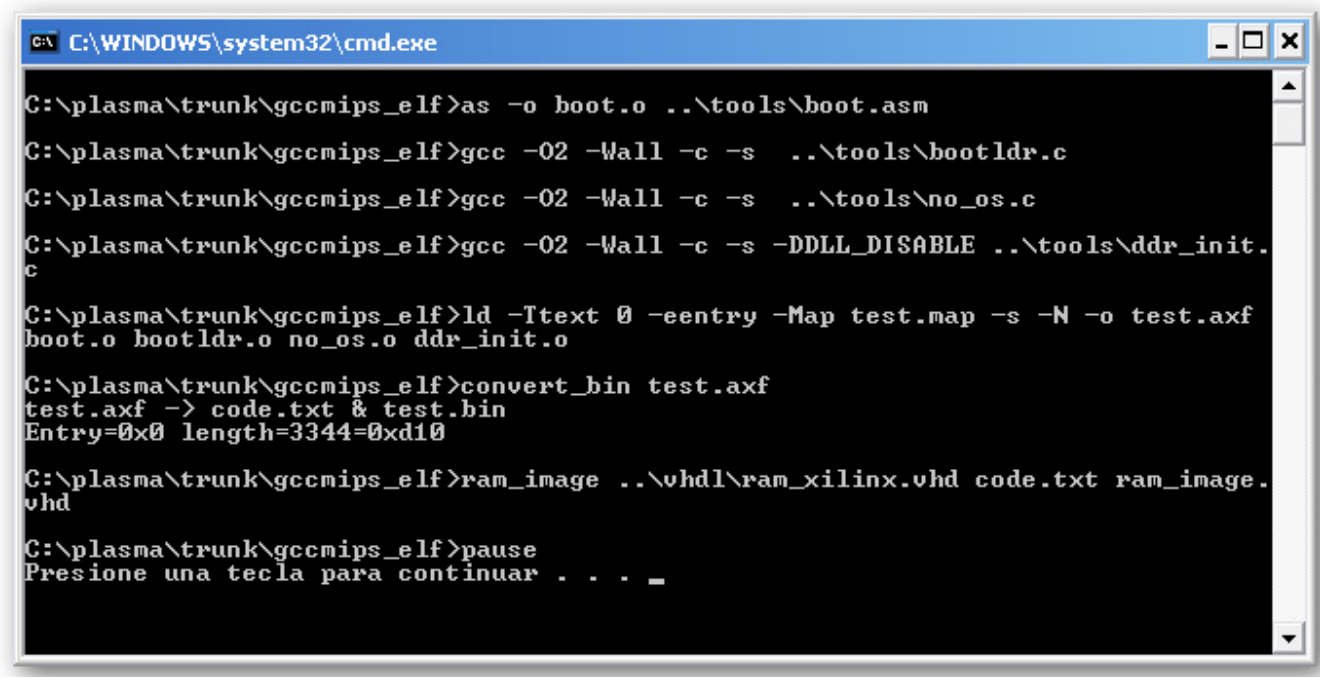

Figura 2. Ventana DOS, creación de archivos Plasma Mips.
Fuente: elaboración propia.

Para realizar la comunicación entre el sensor y la FPGA se evaluaron las características técnicas del sensor ultrasónico, su modo de trabajo y diagrama de tiempos (Figura 3), y así establecer la manera más adecuada de acondicionar las señales entregadas por el sensor de acuerdo a los requerimientos necesarios para este proyecto.

El sensor consta de 4 pines o patillas (Figura 4), las cuales están designadas para la alimentación del circuito y el modo de trabajo del mismo, únicamente requiere de un pulso (5VDC o 1 lógico) en uno de sus pines, denominado para este proyecto como orden de "disparo'', el sensor emite varias ondas que tienen una frecuencia de 40Khz y que al chocar con algún objeto cercano (entre 1,7 cm y 4,3m) rebotarán y regresarán como el eco de las ondas; paralelamente y por medio de otro pin o patilla del sensor genera un pulso, que en este caso será denominado como ''ECO'', y cuya duración en estado alto (5VDC), dependerá de la distancia a la que se encuentre el objeto del sensor (entre 100µs para la distancia mínima y 25ms para la máxima), luego regresará a estado bajo (0VDC).

La FPGA tiene un lenguaje particular de programación llamado **VHDL** *[2]*, pero en el caso especial de este proyecto que requería el uso de un procesador embebido, dicho procesador debía ser programado en lenguaje C. El primer paso fue habilitar dos pines de entrada y salida de la tarjeta de desarrollo desde el microprocesador, entrando a mirar el código fuente del plasma Mips.

De acuerdo al método de funcionamiento del sensor, mencionado anteriormente, se genera un pulso (5VDC) de 10µs que es enviado al pin de ''disparo'' en el sensor. Posterior a este evento, se almacena en un registro de 32 bits (numero binario), la duración del pulso ''ECO'', generado por el sensor y que es recibido por uno de los pines habilitados en la Spartan 3E.

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

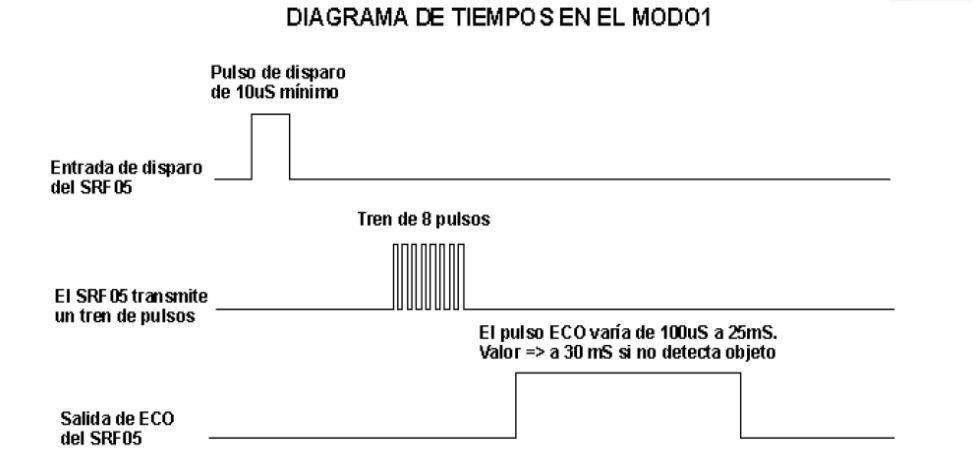

Figura 3. Diagrama de tiempos sensor SRF05.
Fuente: elaboración propia.

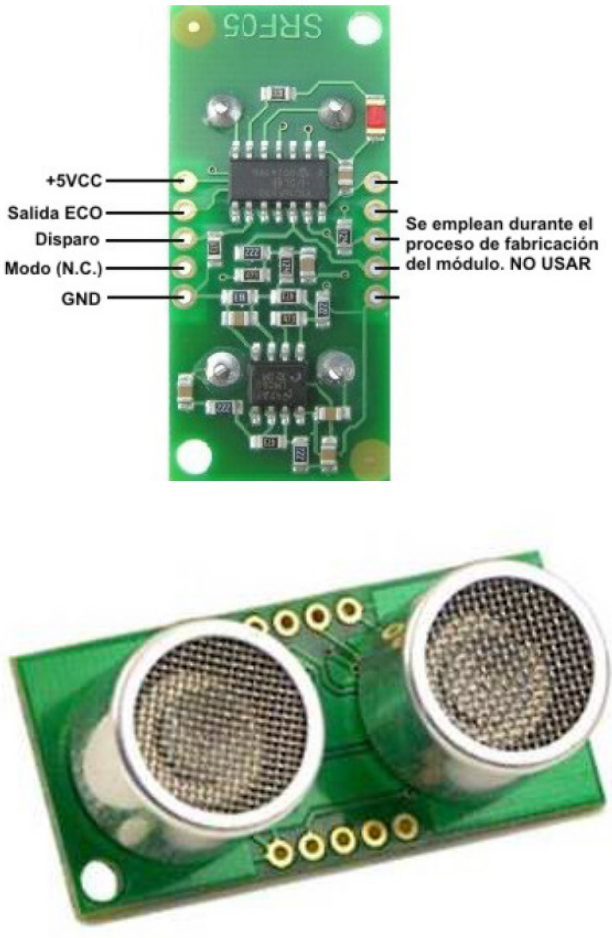

Figura 4. Sensor SRF05.
Fuente: elaboración propia.

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

3.1 Transmisión de la información de la FPGA a un usuario remoto por medio del protocolo TCP/IP

Los protocolos TCP/IP son estándares de protocolos abiertos y gratuitos. Su desarrollo y modificaciones se realizan por consenso, no a voluntad de un determinado fabricante. Cualquiera puede desarrollar productos que cumplan sus especificaciones por su Independencia a nivel software y hardware, su amplio uso los hace especialmente idóneos para interconectar equipos de diferentes fabricantes, no solo a Internet sino también formando redes locales (Figura 5).

3.2 Sistema de transmisión TCPIP para el sensor de nivel

Para programar en la comunicación Ethernet (TCP/IP), hay que entender el concepto de cliente y de servidor, el cliente es el que se conecta al servidor, es decir, para que exista comunicación, el servidor se "levanta" primero y luego es capaz de aceptar 1 o más clientes a la vez según esté programado.

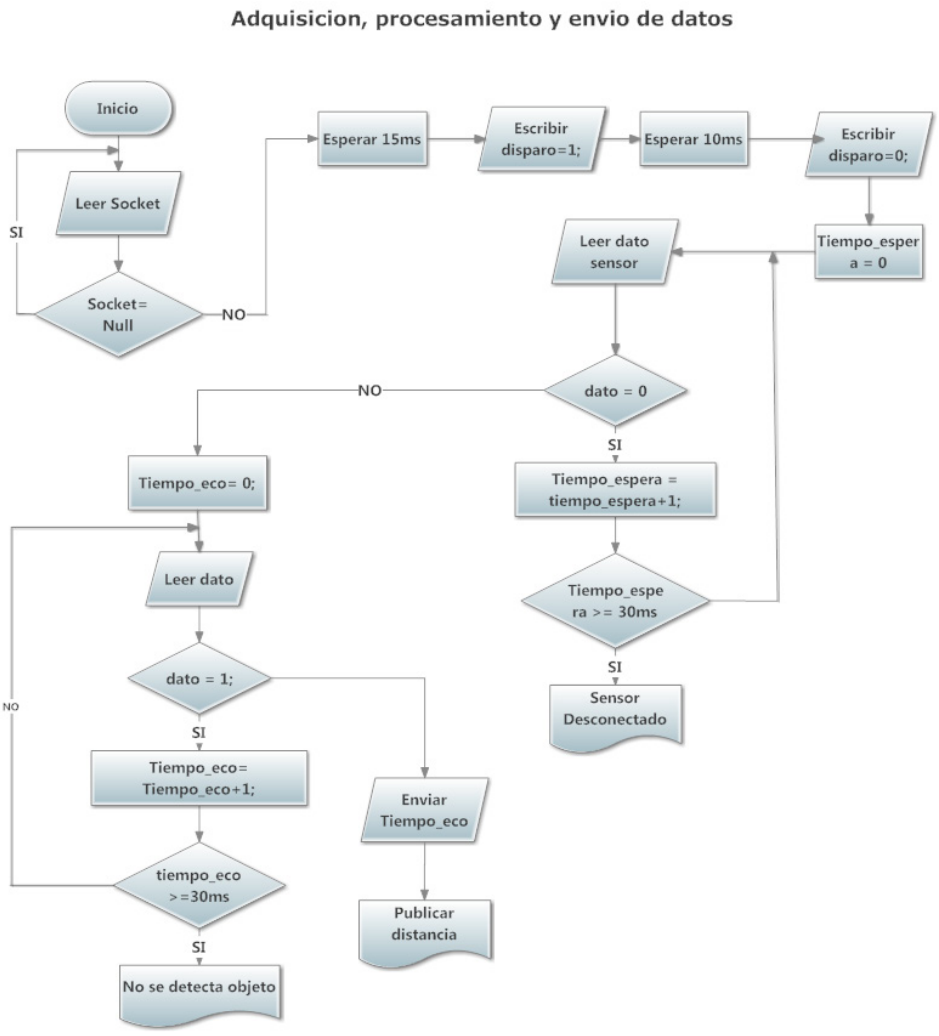

Figura 5. Diagrama de flujo para el envío de datos desde la tarjeta a la interfaz.
Fuente: elaboración propia.

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

El procesador Embebido Plasma Mips, establece una pila de información con los datos recibidos del sensor y los empaqueta en una serie de capas establecidas por el protocolo TCPIP para el envío de la información vía Ethernet al momento de recibir la orden de la interfaz grafica establecida para este proyecto y que se encuentra en un servidor remoto manejado por algún usuario (Figura 6).

El segmento TCP es la unidad de transferencia de datos de este protocolo. Los segmentos se intercambian para establecer y liberar conexiones, transferir datos, enviar acuses de recibo e informar sobre tamaños de ventana. Un acuse de ventana que se dirija del servidor al cliente, puede viajar en el mismo segmento que traslade datos a través de estos.

El protocolo TCP entrega al servidor IP el segmento que contiene los datos y una pseudo-cabecera que no se transmite, pero que permite al protocolo IP completar los datos del o los datagramas que va a generar. El formato del segmento y de la pseudo-cabecera son los que se muestran en la Figura 7.

Estableciendo la comunicación remota de la FPGA con la interfaz gráfica, se gestiona el manejo de la información desde el computador, y se organiza en una estructura ordenada y sencilla para que sea monitoreada por el operario.

4. Desarrollo de la Interfaz de usuario

La interfaz fue diseñada en lenguaje java por ser un software libre, además, cuenta con las librerías necesarias para la comunicación por Ethernet. Los denominados sockets o puertos son los que permiten enviar y recibir datos a través de una conexión de red, generalmente cuando se trabaja con comunicaciones, se usan algoritmos dentro de un bucle infinito, por lo tanto, se necesitó de los threads (hilos) para poder implementar el algoritmo de comunicación (Figura 8).

La interfaz fue diseñada con las herramientas necesarias para el adecuado monitoreo del sensor de nivel, como por ejemplo el estado del sensor, conexión o desconexión con el sensor, etc (Figura 9).

5. Resultados

Al trabajar con sistemas embebidos, se abrieron puertas al conocimiento de nuevos dispositivos y recursos que no se han profundizado todavía en la universidad, así que se diseñaron tutoriales y manuales de usuario para dejar la mayor documentación sobre este trabajo de grado, con la posibilidad de que en otro momento sean utilizados por los estudiantes de la universidad Distrital que deseen realizar proyectos similares.

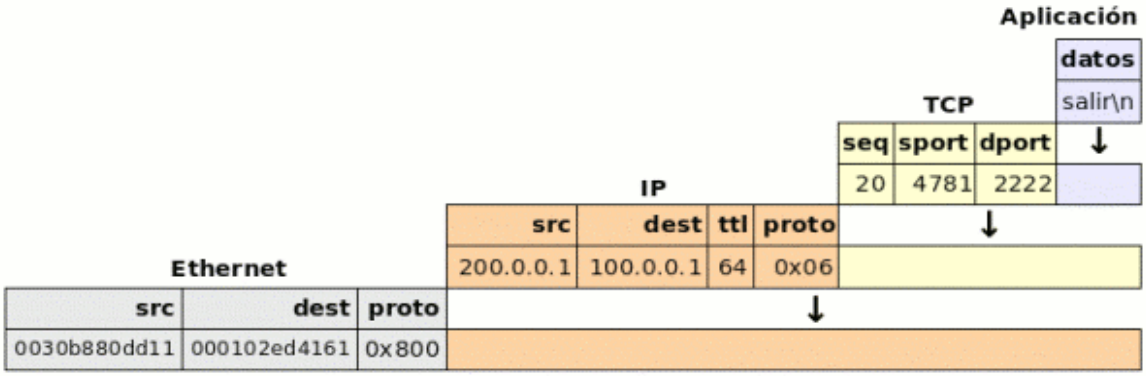

Figura 6. Pila para el envió de información.
Fuente: elaboración propia.

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

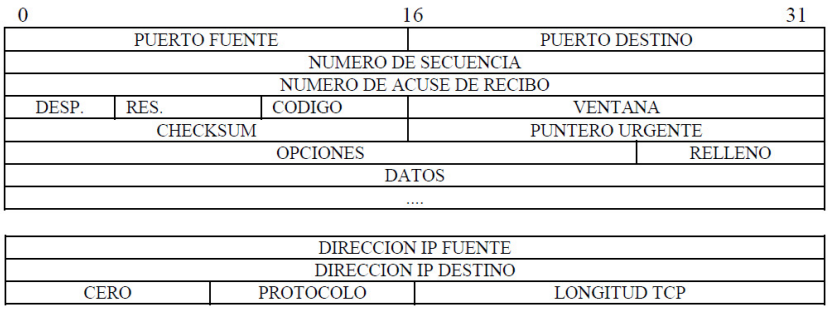

Figura 7. Datagrama de informacion.
Fuente: elaboración propia.

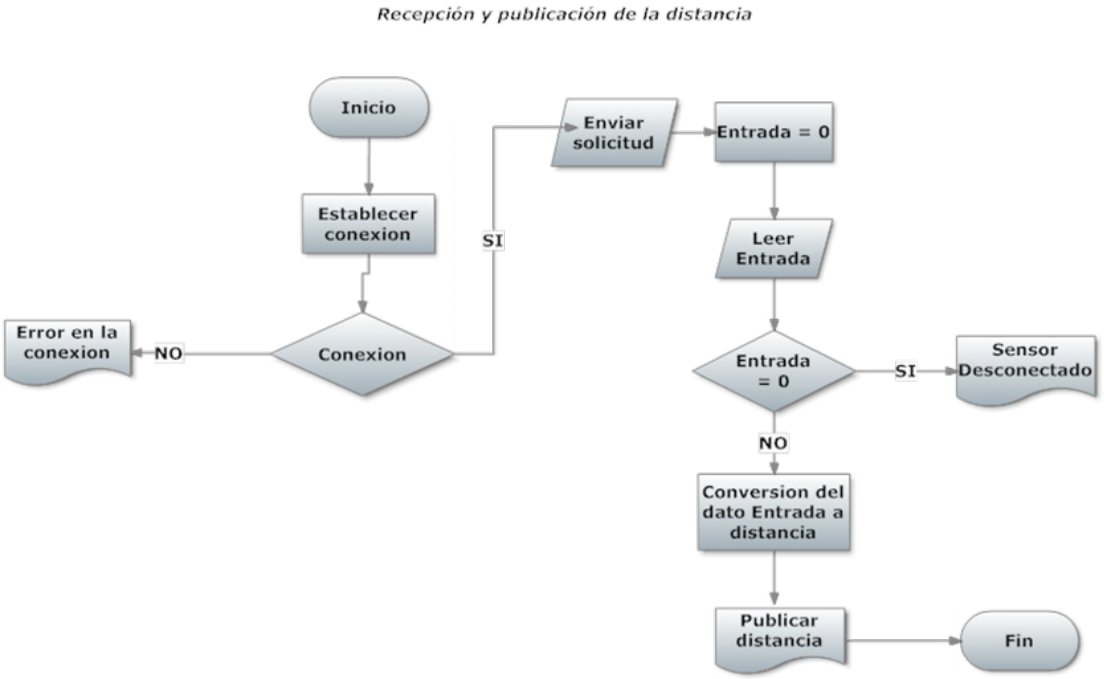

Figura 8. Diagrama de flujo interfaz grafica.
Fuente: elaboración propia.

Después de concluida la investigación, se analiza la posibilidad de implementarla en la maquina HAS-200 del laboratorio de tecnología Industrial, adaptando así la interfaz gráfica para la interacción con la tolva que almacena la materia prima. Se diseña una tapa para la tolva en la cual se coloca el sensor que censará el nivel de la materia prima y que además, permite el llenado de la tolva sin retirar el sensor de su posición para que se pueda verificar el contenido constantemente.

Se realizaron las pruebas durante el proceso automatizado de empaque y despacho de los pedidos que se registran en el software de control de la maquina, en donde se podía constatar, sin ningún inconveniente, el nivel de la materia prima en una de las estaciones de producción.

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

Figura 9. Interfaz grafica.
Fuente: elaboración propia.

6. Expectativas sobre el proyecto

Como principal expectativa, se piensa en la posibilidad de desarrollar una producción en serie de estos sensores para la venta en el mercado después de analizar las ofertas y los costos de estos sensores y sus periféricos correspondientes (Tabla 1).

Tabla 1. Costo por unidad para la producción en serie del proyecto.

Material	Costo($)
Sensor SRF05	30.000
Modulo con puerto Ethernet	46.000
Chip FPGA (XC3s500e)	54.000
Conocimiento empleado y Construcción del Sensor	70.000
Total	200.000

Fuente: elaboración propia.

Estos costos resultan muy económicos teniendo en cuenta que en Colombia no se dispone de una gran oferta en esta clase de dispositivos y que algunas empresas, como la española DOMODESK ofrece sensores desde 86 Euros (unos 240.000 pesos Colombianos) pasando por los 380 Euros y hasta mas sin incluir una interfaz de control para estos sensores, lo cual es otra ventaja para este proyecto, ya que también se diseñó una interfaz gráfica de fácil manejo para el monitoreo de este sensor de nivel. Se espera también, que surja una red de sensores industriales adaptados a cualquier planta automatizada y que sea un incentivo para el desarrollo de nuevos proyectos de grado que implementen la utilización de sistemas embebidos sobre FPGAs.

7. Conclusiones

- Manejar una codificación de instrucciones mas depurada y compleja se convirtió en el principal

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.

logro del diseño y operación de la comunicación, permitiendo sintetizar conocimientos en el manejo de lenguaje VHDL y C.

- El rebote de las ondas generadas por el sensor, depende del ángulo de trabajo del mismo por lo que se debe situar el sensor en una posición estable para evitar pérdidas en la emisión y recepción de las ondas.
- Tener en cuenta, que la ausencia de objetos dentro del límite de medición genera una señal de ''ECO'' fuera del rango de duración, por lo que se establece un límite máximo de 4 metros en la distancia de medición a la hora de implementar el sensor.
- Se ha podido comprobar que es posible desarrollar la transmisión confiable de datos sobre redes TCP/IP utilizando el kit de desarrollo Spartan 3E 500 Starter Kit y el procesador embebido Plasma Processor.

Referencias

[1] M. P Enrique, “Intrumentacion Electronica,” Marcombo., vol. 1, pp. 56-67, Septiembre, 2011.

[2] P. C. Fernando, “Lenguaje para descripción y modelado de circuito,” Universidad de Valencia, Septiembre. 2011.

[3] S. Rohads, “Plasma Processor”, [En línea] Disponible en: www.opencores.org

[4] F. Davidson, “Plasma CPU”, [En línea] Disponible en: http://www.slideshare.net/davidsonfellipe/plasma-cpu

ISSN 1794-211X • e-ISSN 2322-939X • Vol 14, No 1 (enero-junio 2017). pp. 17-26. Universidad Distrital Francisco José de Caldas-Facultad Tecnológica.