

PISTAS

Educativas

NUEVA ÉPOCA • No. 112 • NOVIEMBRE 2015 • ISSN 1405-1249

SENIE 2015

XI Semana de Ingeniería Electrónica



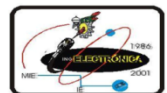
SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE CELAYA



40
Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA



TECNOLÓGICO NACIONAL DE MÉXICO

Manuel Quintero Quintero
Director

Ignacio López Valdovinos
Encargado Secretaría Académica

DIRECTORIO ITC

M.E.D. Ignacio López Valdovinos
Director

M.C. Martha Carreño Juárez
Subdirectora Académica

M.F. Martín Campos Moreno
Subdirector de Servicios Administrativos

M.C. Julián Ferrer Guerra
Subdirector de Planeación y Vinculación

MDPH Teresita de las Nieves Armengol Rico
Jefa del Departamento de Desarrollo Académico



PISTAS EDUCATIVAS

pistaseducativas@itcelaya.edu.mx

COMITÉ EDITORIAL

Juan José Martínez Nolasco

Luis Alberto López González

COMITÉ TÉCNICO

Dra. Abascal Mena Ma. del Rocío - UAM-Cuajimalpa

Dr. Alfaro Ruiz Víctor M. - Universidad de Costa Rica

Dr. Anzures Marín Juan - Universidad Michoacana de San Nicolás de Hidalgo

Dr. Arechiga Martínez René - New Mexico Tech Electrical Engineering Department, U.S.A.

Dr. Astorga Zaragoza Carlos - CENIDET-Cuernavaca

Dr. Aviles Cruz Carlos - UAM-Azcapotzalco

M. C. Borrero Guerrero Henry - Universidad de Sao Paulo

Dr. Bonilla Gutiérrez Isela - Universidad Autónoma de San Luis Potosí

Dra. Bravo Contreras Maricela - UAM-Azcapotzalco

Dr. Campos Delgado Daniel Ulises - Universidad Autónoma de San Luis Potosí

PISTAS EDUCATIVAS

• No. 112 • Noviembre 2015

Contenido

Editorial.....	3
Presentación.....	6
Simulación de la sístole y diástole utilizando el modelo del autómatas híbrido Braulio Cruz Jiménez, Jannette Contreras Rivero, Ricardo Peón Escalante.....	8-21
Diseño, construcción y control de un sistema de levitación neumática Irving Ulises Hernández Miguel, Eufracia Mendoza Monjaraz, Jesús Pacheco Mendoza, José Alberto Cruz Tolentino, Alejandro Jarillo Silva.....	22-42
On model parameter estimation methods of DC electric motors José Jiménez González, Francisco Beltrán Carbajal, Felipe González Montañez, Irvin López García.....	43-60
Modelado y control de un generador de SAGS basado en un convertidor back to back Gabriel Calderón Zavala, Jesús Darío Mina Antonio, Adolfo Rafael López Núñez, Leodan Robisel Martínez Flores.....	61-80
Close loop step test used for tuning PID controller by genetic algorithms Rubén Lagunas Jiménez, Alonzo González Aguilar, Víctor Lanz Gutiérrez De Velasco.....	81-98
Control de ganancias variables para robots de n-grados de libertad Oscar Ramos Arroyo, Fernando Reyes Cortés, María Aurora Diczorca Vargas Treviño, Sergio Vergara Limón.....	99-114
Modelado de un sistema máquina bus infinito usando transformada de Laplace Irma Martínez Carrillo, Carlos Juárez Toledo.....	115-117
Diseño y simulación de un controlador analógico para el convertidor cd-cd reductor Alberto Martínez Barbosa, Esteban Osvaldo Guerrero Ramírez.....	118-143
Control del módulo de calentamiento e instrumentación de una máquina para ensayos de torsión en caliente Gustavo Aquino Mendieta, Miguel Magos Rivera, Juana Eloina Mancilla Tolama.....	144-163
Control de la velocidad de un motor de cd a través de un convertidor cd-cd reductor alimentado mediante paneles solares Alberto Martínez Barbosa, Esteban Osvaldo Guerrero Ramírez.....	164-178
Ocupación espectral de la banda ISM de 2.4 GHz para aplicaciones de radio cognoscitivo Miguel Ángel Díaz Ibarra, Marco Aurelio Cárdenas Juárez, Ulises Pineda Rico, Enrique Stevens Navarro.....	179-194
Sistema de comunicaciones para mini UGVs/UAVs en tareas de protección civil mediante COTS Dulce Mayra Janet Martínez García, Emmanuel Ruiz López, Marco Aurelio Cárdenas Juárez, Enrique Stevens Navarro, Ulises Pineda Rico.....	198-234
Red avanzada de comunicaciones mediante Raspberry Pi para aplicaciones en vehículos aéreos no tripulados Luis Fernando Hernández Medina, Marco Aurelio Cárdenas Juárez, Enrique Stevens Navarro, Armando Arce Casas, Ulises Pineda Rico.....	215-234
Caracterización de canal para sistemas de comunicación por luz visible en interiores A. M. Ramírez Aguilera, J. M. Luna Rivera, R. Pérez Jiménez.....	235-251
Diseño de radiocomunicación de CanSat didáctico Andrés Calvillo Téllez, José Cruz Núñez Pérez, Teodoro Álvarez Sánchez, Ricardo Cárdenas Valdez, Antonio Gómez Roa, Raúl Rascón Carmona.....	252-262
Design, characterization and modeling of a tunneling break-down photodiode integrated in a standard 0.5µm CMOS technology Juan J. Ocampo Hidalgo, J. Alex Zenteno Hernández, Rosnelly Cruz Cecilio, David Flores-Montoya, Olaf E. Duran-Nava, Armando Gómez-Vieyra.....	263-281
Implementación de un circuito caótico memresistivo Moisés Torres Rivera.....	282-301
Delta-Sigma Converter Processing: aplicación de herramienta de software libre para el análisis y caracterización de convertidores $\Sigma\Delta$. Javier Alducin Castillo, Juan Jesús Ocampo Hidalgo, Iván Vázquez Álvarez.....	302-319

PISTAS EDUCATIVAS

• No. 112 • Noviembre 2015

Contenido

Filtro CMOS pasa bajas con parámetros variantes en el tiempo Edgar López Delgadillo, Luis Alejandro Flores Oropeza, Alfonso Vela Rivera.....	320-333
Configuración óptima para un enfriador termoelectrico de dos etapas usando BiTe y PbTe en cada etapa Karen Anaíd Rendón Cortés, Miguel Angel Olivares Robles.....	334-350
Análisis de la conductividad térmica efectiva en un generador termoelectrico segmentado usando Bi2Te3 y PbTe Ana Lilia Nieto Magdaleno, Miguel Ángel Olivares Robles, José Jorge Chanona Pérez.....	351-367
Análisis térmico de un enfriador termoelectrico para bebidas impulsado por rueda mecánica Gerardo Plata Medina, Miguel Ángel Olivares Robles.....	368-387
Influencia del dopante de In_2O_3 en las propiedades eléctricas de varistores a base de $SnO_2-Co_3O_4-Nb_2O_5-Cr_2O_3$ Julio Cesar Martínez Ramírez, A.N. Bondarchuk, A.B. Glot.....	388-398
Optimización geométrica de un MEMS tipo Hot-Arm María Teresa Orvañanos Guerrero, Jorge Varona Salazar, Ramiro Velázquez Guerrero.....	399-421
Virtualización tridimensional interactiva de un parque eólico con fines didácticos Karla del Rosario Trujillo González, A. Guadalupe Toledo Toledo, J. Jesús Arellano Pimentel, Miguel Ángel Hernández López, María de los Angeles Alonso Lavernia, Verónica Martínez Lazcano, Iliana Castillo Pérez, Yira Muñoz Sánchez.....	422-441
Desarrollando competencias digitales en los docentes María de los Angeles Alonso Lavernia, Verónica Martínez Lazcano, Iliana Castillo Pérez, Yira Muñoz Sánchez.....	442-463
Diseño y desarrollo de una red social de aprendizaje colaborativo para dispositivos móviles María Lucía Barrón Estrada, Ramón Zatarain Cabada, Rodrigo Beltrán Lugo.....	464-485
Videojuego didáctico empleando el kinect para la manipulación virtual de un aerogenerador Martín Hernández Reyna, Miguel Ángel Hernández López, J. Jesús Arellano Pimentel, A. Guadalupe Toledo Toledo.....	486-507
Metodología de diseño de osciladores LC con componentes discretos: una propuesta Edgar Omar Lara Alemán, F. Sandoval Ibarra, Esaú Villatoro-Tello, Carlos R. Jaimez-González.....	508-532
Diseño e implementación de Applets como material didáctico de apoyo para cursos de estructuras de datos Betzalet García-Mendoza, Pablo A. Ruiz-Mendoza, Gerardo Real-Flores, Carlos R. Jaimez-González, Esaú Villatoro-Tello.....	533-553
Experiencia de Didáctica Lúdica para Incentivar el Aprendizaje Sergio Zepeda-Hernández, Rocío Abascal-Mena, Erick López-Ornelas.....	554-572
Programación de mini robots para el desarrollo de aprendizaje significativo Margarita Elizabeth Gómez Mayorga, Luz Noé Oliva Moreno.....	573-584
Optimización de un prototipo de sistema fotovoltaico autónomo para iluminación de anuncios espectaculares Pedro Manuel Rodrigo Cruz, José Sebastián Gutiérrez Calderón, Arturo de la Torre Infante.....	585-606
Nuevo inversor fotovoltaico integrado sin transformador Jeziel Vázquez, Nimrod Vázquez, Claudia Hernández, Héctor López, Joaquín Vaquero.....	607-627
Evaluación del desempeño de un transformador monofásico mediante software Hiram Alberto Canseco García, José Jiménez González, Eduardo Campero Littlewood, Irvin López García, Juan Carlos Olivares Galván.....	628-642
Sistema de calentamiento por inducción electromagnética para pruebas experimentales en laboratorios de Físico-Química Julio Ortega Alejos, Leonel Estrada Rojo, Carlos A. Fuentes Hernández, Elías J. J. Rodríguez Segura, Israel Álvarez Ariza, José Jesús Arizaga Ibarra, Diego Díaz López.....	643-665
Sistemas de detección de modo isla en inversores conectados a la red eléctrica: análisis del estado del arte Carlos Javier Camacho Márquez, Nimrod Vázquez Nava, Claudia Verónica Hernández Gutiérrez, Héctor Juan Carlos López Tapia.....	666-683
Análisis del rendimiento de TEG's utilizando una teoría de circuitos térmicos y eléctricos Vargas Almeida Alexander, Olivares Robles Miguel Ángel.....	684-696

COMITÉ TÉCNICO

Dr. Correa Gómez Javier - Instituto Tecnológico de Morelia
M. C. Cruz Roa Ángel Alfonso - Universidad Nacional de Colombia
Dra. Cuevas Rasgado Alma Delia - Universidad Autónoma del Estado de México-Texcoco
Dr. Díaz Carmona José Javier - Instituto Tecnológico de Celaya
Dr. Espinosa Calderón Alejandro - Instituto Tecnológico de Celaya
Dr. Esquit Hernández Carlos A. - Universidad del Valle de Guatemala
Dr. Fallad Chávez Jalil - Centro Universitario de la Costa Sur
Dra.Félix Aviña Vanessa Guadalupe - Universidad Politécnica de Sinaloa
M. C. Fuentes Covarrubias Ricardo - Universidad de Colima
Dr. Gutiérrez Calderón Sebastián - Universidad Panamericana-Aguascalientes
Dr. Gutiérrez Díaz de León Carlos Adrian - Universidad Autónoma de San Luis Potosí
Dr. Herrera Alcántara Oscar - UAM-Azcapotzalco
Dr. Incera Diéguez José Alberto D. - Instituto Tecnológico Autónomo de México
Dr. Jiménez Alaniz Juan Ramón - UAM-Iztapalapa
M. T. Jiménez Orozco Gabriel - ITESM Cd de México
Dr. López Ornelas Erick - UAM-Cuajimalpa
Dr. Magos Rivera Miguel - UAM-Azcapotzalco
Dr. Martínez Pelaez Rafael - Universidad de la Sierra Sur
Dr. Mena Camare Luis Javier - Universidad Politécnica de Sinaloa
Dr. Mendoza Gutiérrez Marco Octavio - Universidad Autónoma de San Luis Potosí
Dr. Monroy Borja Raúl - ITESM Edo de México
Dr. Morales Alanis Eduardo - New England Baptist Hospital
Dr. Parra Michel Ramón - CINVESTAV-GDL
Dr. Peña Campos Fernando - CINVESTAV-GDL
Dr. Potes Cristhian M. - Philips Research North America
Dr. Ramírez Treviño Alberto - Instituto Tecnológico Superior de Cajeme
M. C. Reider Burstin Jerry - Universidad Anáhuac – México Norte
Dr. Reyes Ortiz José Alejandro - UAM-Azcapotzalco
Dr. Rivas Araiza Edgar - Universidad Autónoma de Querétaro
Dr. Rizo Domínguez Luis - Universidad del Caribe
Dr. Rodrigo Cruz Pedro - Universidad Panamericana Aguascalientes
Dr. Rodríguez García José G. - CINVESTAV-Zacatenco
Dr. Rodríguez Segura Elías José J. - Instituto Tecnológico de Celaya
M. C. Sanchez Gomez Claudia Nallely - Universidad Panamericana-Aguascalientes
M. C. Santiago Espinosa Felipe - Sanviago Tecnológica de la Mixteca
Dr. Tecpanecatí Xihuitl Jose Luis - Universidad Autónoma de San Luis Potosí
Dr. Toral Cruz Homero - Universidad de Quintana Roo
Dr. Torres Román Deni Librado - CINVESTAV-GDL
Dr. Vargas Rubio Juan Gaspar - UAM-Azcapotzalco
Dr. Vázquez Álvarez Iván - UAM-Azcapotzalco
Dr. Vázquez Castillo Javier - Universidad de Quintana Roo
Dr. Vázquez Cerón Ernesto R. - UAM-Azcapotzalco
M.C. Velázquez Buendía Francisco - Universidad Panamericana Aguascalientes
Dr. Velázquez Guerrero Ramiro - Universidad Panamericana Aguascalientes
Dr. Villegas Cortez Juan - UAM-Azcapotzalco

Pistas Educativas es una publicación impresa del Departamento de Desarrollo Académico del Instituto Tecnológico de Celaya. Los artículos y ensayos son responsabilidad de sus autores y no expresan necesariamente la ideología de la institución ni de la revista. En caso de citar la información de los contenidos se agradecerá dar los créditos a los autores, sus obras, así como a esta fuente.

Dirección: Antonio García Cubas Pte. No. 600 Esq. Av. Tecnológico, C.P. 38010, Celaya, Gto. Tel. 01-(461)-61- 175-75. Fax. 01-(461)-61-179-79. Certificado de Licitud y Contenido 6216 y 4777, respectivamente. Registro de Derecho de Autor, Expediente de Reserva: 6 98 92. ISSN 1405-1249.

Editorial

En su edición 112 la revista Pistas Educativas publica con agrado la Memoria de la Décimo Primera Semana Nacional de Ingeniería Electrónica –SENIE 2015-, que se llevó a efecto entre el 7 y el 9 de octubre bajo la organización conjunta de la División de Ciencias Básicas e Ingeniería de la Universidad Autónoma Metropolitana Azcapotzalco así como la Facultad de Ciencias de la Universidad Autónoma de San Luis Potosí, fungiendo esta última institución como anfitriona del evento.

Cada uno de los artículos que aquí se publican fueron sometidos a la consideración de un grupo evaluador constituido por dos investigadores expertos en el tema y es del Comité Técnico de SENIE 2015 la responsabilidad en lo que se refiere a su calidad técnica. Por su parte, Pistas Educativas tuvo bajo su cuidado la edición de los textos de acuerdo, como siempre, con sus normas de publicación.

A través de esta edición, el Instituto Tecnológico de Celaya, por medio de Pistas Educativas, avanza en su cometido de ser un vehículo para la difusión del conocimiento, albergando en esta ocasión las experiencias y logros de los que dan cuenta en sus artículos los estudiantes, profesores e investigadores de buena parte del sistema educativo nacional que se congregaron en SENIE 2015.

PISTAS EDUCATIVAS

• No. 112 • Noviembre 2015

Contenido

Sistema fotovoltaico multifuncional conectado a la red eléctrica controlado por medio de modos deslizantes	
Iván Méndez, Nimrod Vázquez, Claudia Hernández, Héctor López, Joaquín Vaquero.....	697-715
Análisis del proceso de balanceo de voltaje en un arreglo de supercapacitores	
Edgar Peralta Sánchez, Pedro Celestino Castellanos Morales, Nayeli Itzel Hernández Zárate, Gerardo Mino Aguilar, Beatriz E. Graniel García.....	716-734
Estudio comparativo del coeficiente de potencia y de torque presentados en diversas turbinas eólicas	
Viviana Reyes Andrade, Oscar Carranza Castillo, Jaime José Rodríguez Rivas.....	735-756
Sistema para el control y prueba de motores síncronos con base en el procesador digital de señales TMS320	
Fermin Hugo Ramírez Leyva, Mónica Edith García García, Felipe Santiago Espinosa, Gerardo Cruz González.....	757-774
Medida experimental y modelado matemático de módulos fotovoltaicos	
Raúl Castillo Meraz, Roberto Carlos Martínez Montejano, Isaac Campos Cantón, Misael Francisco Martínez Montejano.....	775-796
Modelado, control y simulación de elementos básicos que componen una micro red de CD	
Arturo De los Santos Vázquez, Elías José Juan Rodríguez Segura, Nancy Visairo Cruz.....	797-817
Tarjeta didáctica de control para tiristores	
Luis Alejandro Flores Oropeza, Eduardo Anguiano Mendieta, Edgar López Delgado.....	818-838
Medición de velocidad sin sensor en motores de inducción sumergibles conectados directamente a la línea	
Antonio Zamarrón Ramírez, Salomón Rodríguez Carrera, Daniel Zamarrón Barrón, Francisco Valdivia.....	839-852
Convertidores balanceadores de voltaje: Estado del arte	
Fidel Alejandro Castro Espinosa, Elías José Juan Rodríguez Segura, Ciro Alberto Núñez Gutiérrez.....	853-872
Desarrollo de un medidor portátil para la detección de metales pesados disueltos en medios acuosos utilizando principios de fluorescencia	
Marco Antonio González Cantellano, Luis Manuel Montaña Zetina.....	873-894
Sistema integral de software y hardware para el aprendizaje del funcionamiento y manejo de los sensores	
Javier Silvestre Zavala, José Nicolás Márquez Márquez.....	895-917
Desarrollo de un sistema de electro estimulación transcutánea de bajo costo	
Miriam C. Reyes Fernández, Ruben Posada Gómez, Albino Martínez Sibaja, Oscar O. Sandoval González, Mario Alberto García Martínez, Luis H. Sánchez Medel.....	918-934
Método para la medición de la inductancia bajo condiciones reales de operación	
Jerry Nathan Reider Burstin.....	935-953
Estimación de la humedad y resistencia a compresión final, de una mezcla de concreto fresco utilizando microondas.	
Gerardo Calva Olmos, Rafael Prieto Meléndez, Alberto Herrera Becerra, Alejandro Padrón Godínez, Mario Pacchiano de la Garza.....	954-976
Plataforma de adquisición y monitoreo para el estudio de las condiciones ambientales externas y de confort en casas prototipo para la validación de un control térmico por medio de un sistema de muro verde y aislante térmico	
Cesar Agustín Corona Patricio, Cesar Alberto Dueñas Sosa, Jesús Iván Castro Enríquez, José Luis Avendaño Juárez, Jorge Adán Sánchez Reséndiz, Eusebio JR. Ventura Ramos, Francesca Olivieri.....	977-997
Mejora de la resolución espectral de un monocromador usando un controlador de micropasos	
Ricardo Castro García, Luis Felipe Lastras Martínez.....	998-1019
Efectos de holgura de una banda trapezoidal de acoplamiento de un sistema electromecánico sobre magnitudes mecánicas y eléctricas	
Jorge Iván Ramírez López, Efraín Ramos Buenfil, Marco Antonio Rodríguez Blanco, Javier Torres Jurado.....	1020-1035

PISTAS EDUCATIVAS

• No. 112 • Noviembre 2015

Contenido

Registrador de energía fotovoltaica inyectada a la red eléctrica mediante LabView Luis Carlos Gallegos Hernández, Mariela Serna Serna, Eduardo Salinas Rodríguez, Antonio García Narvaez, Guillermo Reséndez Morales, Pedro Casillas Ríos.....	1036-1049
Monitoreo del consumo doméstico de agua potable utilizando tecnología ZigBee Pedro Alvarado Medellín, Job Daniel Rodríguez Hernández, Ruperto Ortiz Gómez, Blanca Esthela Solís Recendez.....	1050-1064
Sistema teleoperado para estimulación eléctrica transcorneal de señales múltiples Juan Jaime Salinas Lara, Eric Simancas Acevedo, Daniel Robles Camarillo, Luis Niño de Rivera y Oyarzabal.....	1065-1086
Interpolación polinomial para determinar el tiempo óptimo de tratamiento criogénico para acero AISI D2 en medio básico Irma Hernández Casco, Carlos Juárez Toledo, Irma Martínez Carrillo, Rubén Herrera Galicia, José David Arroyo Pérez, Manuel de Jesús Hernández Gutiérrez, Miriam Roxana Vázquez Flores, Tania Berenice Palacios Molina	1087-1100
Monitoreo y Control de Procesos Basado en Arduino y Raspberry Vía Internet Rubén Herrera Galicia, José David Arroyo Pérez, Manuel de Jesús Hernández Gutiérrez, Miriam Roxana Vázquez Flores, Tania Berenice Palacios Molina.....	1101-1120
Análisis para el diseño de circuitos osciladores de Colpitts con sensores de gas QCM Juan Jesús Jiménez Arellano, Severino Muñoz Aguirre, Georgina Beltrán Pérez, Juan Castillo Mixcoatl, José Lorenzo Muñoz Mata	1121-1135
Análisis cinemático y síntesis de un sistema de palancas para la sub-actuación de un dedo artificial con 3 articulaciones Jesús Alberto Ordaz Rivera, Emilio Miguel Soto García, José Eligio Moisés Gutiérrez Arias, José Eladio Flores Mena, María Monserrat Morín Castillo....	1136-1156
Una propuesta de arquitectura para el control de un robot guía Karla Lourdes Luna Gallegos, Elvia Ruth Palacios Hernández, Antonio Marín Hernández.....	1157-1173
Fault detection in VSD-fed induction motors through Park's impedance and fuzzy systems Arturo Mejía Barron, Armando Guadalupe García Ramírez, Roque Alfredo Osornio Ríos, Rene de Jesus Romero Troncoso.....	1174-1192
Diseño e implementación de tele operación de un robot móvil diferencial programado con Python desde un dispositivo Android Josué Cirilo Cruz, Arturo Zúñiga López, Juan Villegas Cortez, Carlos Avilés Cruz.....	1193-1209
Control en tiempo real de un mecanismo XY virtual por medio de un joystick Gustavo Aquino Mendieta, Miguel Magos Rivera, Ricardo Godínez Bravo.....	1210-1231
Inicialización de un robot humanoide tipo BIOLOID para tareas de caminado Antonio Benitez Ruiz, María Auxilio Medina Nieto, Jorge de la Calleja Mora, Emmanuel Baleón Flores, Carmina Cruz Ochoa, Ivonne López Cuacuas.....	1232-1253
Diseño de un sistema de desarrollo para la emulación de tráfico en una intersección vial con vehículos autónomos Andrés Ferreyra Ramírez, Arturo Zúñiga López, Roberto A. Alcántara Ramírez.....	1254-1274
Renderizado háptico para el Novint Falcon Enrique Linares Ramírez, Jorge Gudiño Lau, Saida Miriam Charre Ibarra, Miguel Duran Fonseca.....	1275-1287
Caracterización de los sonidos implicados en el ciclo cardíaco mediante un fonocardiograma Braulio Cruz Jiménez, Jannette Contreras Rivero, Luis Ricalde Castellanos.....	1288-1308
El invisible y asombroso proceso de la comunicación oral: bases sobre reconocimiento de voz Ángel David Pedroza Ramírez, José Ismael de la Rosa Vargas.....	1309-1329
Deconvolución óptima de mediciones de fluorescencia Gerardo Ibarra Vázquez, Daniel U. campos Delgado, Edgar R. Arce Santana, Javier A. Jo.....	1330-1349
Descomposición de datos multi-espectrales: interfaz gráfica para Matlab Omar Gutiérrez Navarro, Daniel U. Campos Delgado, Edgar R. Arce Santana, Javier A. Jo.....	1350-1368
Estimación de erosión de suelos utilizando sensores remotos y programación genética Cesar Augusto Puente Montejano, Gustavo Olague Caballero.....	1369-1388
Hacia la clasificación de fonocardiogramas utilizando descriptores caóticos y estadísticos Carlos Antonio Osorio Maceda, Hugo G. González-Hernández.....	1389-1408
Activación sistema de frenos ABS usando visión por computadora para la detección de la superficie de rodamiento Gabriel García Ponds, Juan Villegas Cortez, Carlos Avilés Cruz, Iván Vázquez Álvarez, Ismael Osuna Galán, Yolanda Pérez Pimentel.....	1409-1423
Implementación de una estructura neuronal celular en hardware reconfigurable Luis F. Muñoz M., Juan José Raygoza P., J. Roberto R. Barón, Susana Ortega Cisneros.....	1424-1444
Identificación de rostros por técnica de puntos de interés SURF Cesar Benavides Alvarez, Graciela Román Alonso, Juan Villegas Cortez, Carlos Avilés Cruz.....	1445-1465
Segmentación de Iris con OpenCV en Android Oscar Fuentes Salome, Aldrin Barreto Flores, Verónica Edith Bautista López.....	1466-1486
Detección de Puntos Salientes para Medición de la Simetría Facial Ricardo Irving Barrón Martínez, Ruth Mariela Aguilar Ponce, José Luis Tecpanecatí Xihuitl.....	1487-1507
Virtualización mediante MetaRouter para la implementación de una red wireless de navegación anónima tipo TOR en equipos Mikrotik Roberto David Meneses Basantes, Darwín Leonidas Aguilar Salazar, Christian Nestor Vega Muñoz, Rita Paola León Pérez.....	1508-1527

La aplicabilidad del cómputo ubicuo para el monitoreo de animales de caza Adrian Castañeda Morfin, María Andrade-Aréchiga, Pedro Damián-Reyes.....	1528-1545
Tecnología móvil para evaluar la calidad en uso de Proyectos-Productos-Servicios en eventos de innovación e invención Laura Sílvia Vargas Pérez, Agustín Francisco Gutiérrez Tornés, Edgardo Manuel Felipe Riverón, Vanessa Atenea Vargas Pérez, Jorge Peralta Escobar.....	1546-1566
Diseño y construcción de un sistema de supervisión para la evaluación de la calidad del agua en sistemas de cultivo de camarón José Juan Carbajal Hernández, Luis Pastor Sánchez Fernández, Luis Alfonso Aguilera Larrañaga.....	1567-1587
Diseño de un middleware tolerante a fallas basado en el protocolo Paxos Ricardo Adán Madrid Trejo, Ricardo Marcelín Jiménez, Orlando Muñoz Texzocotetla.....	1588-1609
Arquitectura de descubrimiento de servicios para entornos hospitalarios (módulo tiny application) José Luis Santiago López, Víctor Alberto Gómez Pérez, Adrián Josué Ramírez Díaz, Alejandro Jarillo Silva, Juan Carlos Santiago López.....	1610-1632
Detección y análisis de comunidades en redes sociales (#TodosSomosPolitécnico) Erick López Ornelas, Rocío Abascal Mena, Sergio Zepeda Hernández.....	1633-1651
Sistema de multas automático mediante la tecnología NFC para una biblioteca María Elena Acevedo Mosqueda, Marco Antonio Acevedo Mosqueda, Karen Ailed Neri Espinoza, Alexa Chávez Álvarez.....	1652-1664
Propuesta de tutorial interactivo en línea para apoyar la enseñanza del lenguaje de consulta estructurado Arturo Palma Hernández, Carlos R. Jaimez González.....	1665-1684
Identificación de influyentes en Twitter a través del análisis textual y la elaboración de grafos Dra. Rocío Abascal Mena, Dr. Erick López Ornelas, Dr. Sergio Zepeda Hernández.....	1685-1702
Proyección de los índices de criminalidad de los delitos del fuero común: una representación sobre un sistema de información geográfica para la Ciudad de México César Hernández Anaya, Héctor Rafael Orozco Aguirre.....	1703-1720
Aplicación web para la generación de material didáctico incorporando recursos educativos Reyna Carolina Medina Ramírez, Carlos Alberto López Pérez, Fausto Casco Sánchez.....	1721-1739
Repositorio de grafos para el conteo de conjuntos independientes Juan Antares Perdomo Flandez, Pedro Bello López, Meliza Contreras González, Brayan Chavez Benavides.....	1740-1759
Agrupamiento de servicios Web usando el algoritmo de colonia de hormigas Maricela Claudia Bravo Contreras, Román Anselmo Mora Gutiérrez, Roberto Alfonso Alcántara Ramírez.....	1760-1778
RespirAtorio, una aplicación de realidad aumentada para niños María Auxilio Medina Nieto, César Rafael Chío Plata, Brenda Susana Ponce Valencia, Antonio Benitez Ruiz, Jorge de la Calleja, Eduardo López Domínguez.....	1779-1799
La lógica difusa como un medio para identificar y valorar las alteraciones de conducta en niños a nivel de educación básica: primaria y secundaria Andrés Ferreyra Ramírez, David G. Maxinez, Arturo Zúñiga López, Roberto A. Alcántara Ramírez.....	1800-1819
Plataforma de entrenamiento para PIC Mauricio López Villaseñor, Miguel Ángel Ruiz Sánchez, Miguel López Guerrero.....	1820-1836
Sistema de acceso a áreas restringidas con detección de intrusos José Ignacio Vega Luna, Mario Alberto Lagos Acosta, Gerardo Salgado Guzmán, Víctor Noé Tapia Vargas.....	1837-1852
Uso del lenguaje VHDL como apoyo a la enseñanza de arquitectura de computadoras Carlos Federico Hernández Farfán, Diego Andrés Navarro Jiménez, Emmanuel Eduardo Hernández Rico, Jonathan Paris Vargas Mosqueda.....	1873-1892
Desarrollo e implementación de un algoritmo tolerante a fallas para una red ZigBee Stephany Bucio Herrera, Alberto Carlos Cañada Gómez, David Rodríguez Uribe, Ricardo Marcelín Jiménez, Mauricio López Villaseñor.....	1853-1872
Estructuras diseñadas a medida para adquisición de datos de un sensor de visión dinámica usando el protocolo AER en un FPGA Nicolás Maldonado Orozco, J. Roberto Reyes Barón, Juan José Raygoza P., Susana Ortega Cisneros, Juan Luis del Valle.....	1893-1911
Implementación y optimización del uso de DPS en FPGA en diseño de circuitos a medida para calcular determinantes de orden 4 Francisco J. Plascencia Jauregui, Juan José Raygoza P., Edwin C. Becerra A., Susana Ortega Cisneros.....	1912-1933
Sistema de alertas y control de luces de una casa, utilizando plataformas de internet y comunicación inalámbrica Roberto de Lima Hernández, Aldrin Barreto Flores, Verónica Edith Bautista López.....	1934-1946
Arquitectura genérica de una red en chip de enrutamiento unidireccional en FPGA Jorge Ernesto López Arce Delgado, Juan José Raygoza P., Susana Ortega Cisneros, Jorge Rivera D., Pablo Moreno Villalobos.....	1947-1959
Monitor de alerta de fugas de gas José Ignacio Vega Luna, Mario Alberto Lagos Acosta, Gerardo Salgado Guzmán, Víctor Noé Tapia Vargas.....	1960-1973
Desarrollo de un sistema embebido móvil de bajo costo utilizando la tarjeta beaglebone Edgar Alejandro Rivas Araiza, Alexander Rodríguez Rosales, Estefanía Desiree Avalos Rivera.....	1974-1994
Implementación de un multiplicador de punto flotante de doble precisión basado en el estándar IEEE 754-2008 José Itzcóatl Sandoval López, Juan José Raygoza Panduro, Susana Ortega Cisneros, Jorge Rivera Domínguez.....	1995-2016
Diseño y construcción de un limnógrafo electrónico Felipe Santiago Espinosa, Gabriela Álvarez Olguín, Fermín Hugo Ramírez Leyva.....	2017-2038
Desarrollo de un frecuencímetro de conteo recíproco con un FPGA para sensores basados en resonador de cuarzo José Lorenzo Muñoz Mata, Severino Muñoz Aguirre, Juan Jesús Jiménez Arellano, Georgina Beltrán Pérez, Juan Castillo Mixcóatl, Takamichi Nakamoto.....	2039-2053

PRESENTACIÓN

El avance que la industria electrónica a nivel mundial ha tenido en las dos últimas décadas ha sido impresionante, en la actualidad los dispositivos y equipos electrónicos se encuentran en prácticamente todas las áreas del quehacer humano. Así mismo, las áreas de investigación y de aplicación se multiplican cada día dando paso a nuevas ramas afines a la electrónica y sus aplicaciones. Entre estas tenemos: robótica, biónica, telemática, nano-electrónica, domótica, automatización, telecomunicaciones, electrónica de potencia, sistemas computacionales, procesamiento de señales y educación, por mencionar sólo algunas.

Los resultados de los trabajos de investigación, así como de las aplicaciones tecnológicas, que realizan los grupos de profesores y de estudiantes de las instituciones educativas y de investigación son numerosos y variados.

La Semana Nacional de Ingeniería Electrónica (SENIE) es un congreso nacional que busca servir de foro en el cual los investigadores y estudiantes de las diversas áreas de la Ingeniería Electrónica puedan intercambiar experiencias, conocimientos y establecer vínculos de colaboración, mediante actividades tales como: presentación de ponencias, conferencias magistrales, conferencias técnicas y exposición industrial.

La décimo primera edición del evento se realizó del 7 al 9 de octubre en las instalaciones de la Facultad de Ciencias de la Universidad Autónoma de San Luis Potosí. Esta institución goza de un reconocido prestigio a nivel nacional, por la calidad de enseñanza que se imparte, la investigación y divulgación del quehacer científico y la alta habilitación de su planta de profesores-investigadores.

Se recibieron 183 trabajos para su evaluación provenientes de 91 distintas instituciones de educación superior y centros de investigación tanto públicos como privados de 26 estados del país. La revisión de los trabajos fue realizada por un Comité Técnico formado por más de 70 especialistas en las distintas temáticas, varios de ellos de adscritos a instituciones en el extranjero. Como resultado del proceso de revisión, fueron aceptados para su presentación 109 artículos.

En este número especial de *Pistas Educativas*, el cual recopila los artículos expuestos en el evento, el lector encontrará una muestra representativa de los trabajos realizados en el último año por investigadores, profesores y estudiantes de diversos Centros de Investigación y Educación Superior del país.

Es de esta forma que la División de Ciencias Básicas e Ingeniería de la Universidad Autónoma Metropolitana campus Azcapotzalco, la Facultad de Ciencias de la Universidad Autónoma de San Luis Potosí y el Instituto Tecnológico de Celaya, cumplen con el objetivo de proporcionar un foro para la divulgación de los trabajos de investigación relacionados con la Ingeniería Electrónica y sus diversas ramas.

Dr. Miguel Magos Rivera
Presidente del Comité Organizador
XI Semana Nacional de Ingeniería Electrónica 2015

Sistema de alertas y control de luces de una casa, utilizando plataformas de internet y comunicación inalámbrica

Roberto de Lima Hernández

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica, Ciudad Universitaria, Av. San Claudio y 18 sur, Col. San Manuel, C.P. 72570, Puebla, Puebla, México
delima_87@hotmail.com

Aldrin Barreto Flores

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica, Ciudad Universitaria, Av. San Claudio y 18 sur, Col. San Manuel, C.P. 72570, Puebla, Puebla, México,
Teléfono: (222) 2295500
abarreto@ece.buap.mx

Verónica Edith Bautista López

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Ciudad Universitaria, Av. San Claudio y 18 sur, Col. San Manuel, C.P. 72570, Puebla, Puebla, México,
Teléfono: (222) 2295500
vbautista@cs.buap.mx

Resumen

En este trabajo se presenta la implementación de un sistema de domótica enfocado al control de luces y vigilancia, haciendo uso de plataformas de internet y dispositivos embebidos. El primer enfoque se basa en controlar el encendido y apagado de las luces de un hogar desde un sitio de internet, minimizando el costo y haciendo uso de comunicación de radiofrecuencia con el módulo NRF24L01. Por otro lado, el segundo enfoque se centra en la implementación de un sistema de notificaciones mediante herramientas de internet, tales como whatsapp, twitter y dropbox cuando alguien toca el timbre en el hogar o se activa un sensor (PIR o magnético). En la implementación se

hizo uso de un microcontrolador con características de WIFI denominado electric imp, junto con la integración de una Raspberry Pi para el manejo de bibliotecas. Finalmente, se realiza la integración del sistema y se realizan diferentes experimentos en diferentes escenarios, verificando el correcto funcionamiento del sistema.

Palabra(s) Clave(s): domótica, internet de las cosas, sistemas embebidos.

1. Introducción

La forma en la que los seres humanos nos comunicamos ha cambiado vertiginosamente, uno de los factores principales que influyen en lo anterior es la evolución del internet. Por tal motivo, Kevin Ashton en [1] introduce el término de “Internet de las Cosas” (IoT); actualmente IoT es un tema que se ha inmerso en la sociedad y está siendo desarrollado desde diferentes vertientes debido a la evolución de los sistemas embebidos. De la misma manera, otro ende que ha revolucionado los medios de comunicación son las plataformas de internet, tales como aplicaciones de mensajería de texto, redes sociales, y almacenamiento en la nube, por mencionar algunas.

Por otro lado, la evolución de los dispositivos empotrados y la flexibilidad de los lenguajes de programación de alto nivel han contribuido a la evolución de la domótica. Éste último término se refiere al uso de sistemas de control y tecnologías de la información para automatizar una vivienda [6]. Los primeros en involucrarse en el desarrollo de éstos sistemas fue la empresa Pico Electronics, al crear el protocolo X10 [2] el cual nos permite controlar remotamente cualquier dispositivo electrónico que esté conectado dentro de éste estándar.

Por lo tanto, uno de los retos de la actualidad es crear aplicaciones de domótica que conjunten las tecnologías emergentes mencionadas: dispositivos embebidos, IoT, y plataformas de internet. En este sentido, se han realizado trabajos que hacen uso de algunas de éstas, como el diseño de un “Refrigerador Inteligente” [3] el cual se basa en el monitoreo de dicho electrodoméstico, de tal manera que el sistema detecta la falta de determinados alimentos y con base en dicha información se envían alertas al usuario por medio de su celular usando comunicación bluetooth. Sin embargo, el proceso anterior se realiza mediante una computadora de altos recursos y un microcontrolador, lo cual representa una desventaja debido a que el sistema de procesamiento es muy amplio para la aplicación, además éste no es práctico en términos de movilidad. Así mismo, en [4] se desarrolló un sistema de control de dispositivos electrónicos usando módulos zigbee de radiofrecuencia e internet. De tal forma que desde una aplicación

android conectada a internet, se controlan las luces o electrodomésticos de un hogar; a pesar de su buen desempeño respecto a la comunicación entre dispositivos, el usar módulo zigbee hace que se eleve el costo del sistema y en consecuencia se limite el número de dispositivos a controlar. De manera similar en [5] se implementa una red piconet de módulos bluetooth los cuales reciben y envían información entre ellos; éste último trabajo se centra en el rendimiento de la red, por lo tanto no implementan alguna etapa de potencia o comunicación con dispositivos electrónicos.

En el presente trabajo se crea una sinergia de las tecnologías mencionadas (dispositivos embebidos, IoT, y plataformas de internet) para implementar un sistema de domótica enfocado al control de luces y vigilancia. Por un lado, el uso de los dispositivos empotrados nos permite implementar un sistema inalámbrico usando protocolos de comunicación wifi y de radiofrecuencia, además facilita la manipulación de cámaras para la vigilancia del hogar. Por otro lado, las plataformas de internet facilitan la interacción entre el usuario y su hogar de manera remota.

La organización del presente artículo es de la siguiente manera. En la sección 2 se describe el esquema general del trabajo y se detalla en la metodología que se llevó a cabo. En la siguiente sección se presentan los resultados obtenidos. Finalmente las conclusiones y trabajo futuro se presentan en la sección 4.

2. Desarrollo

Como ya se mencionó, el trabajo se enfoca en dos vertientes: sistema de vigilancia y control de luces. El primero consiste básicamente de sensores (presencia, timbre y magnético) que al ser activados, el sistema envía una notificación de texto, imagen y video al usuario de la casa mediante plataformas de internet, esto es: whatsapp, twitter y dropbox. De la misma manera, el usuario puede activar una emisión de video de la casa en tiempo real y ésta puede ser visualizada en cualquier reproductor multimedia con soporte del protocolo rtsp. La segunda vertiente está relacionada al control de luces, permite al usuario controlar el encendido y apagado de éstas, mediante un sitio web. Con el objetivo de crear un sistema práctico y de fácil instalación, los dispositivos están comunicados de manera inalámbrica mediante radiofrecuencia. Así mismo, la interfaz entre el usuario-casa, se realizó mediante una página web compatible con cualquier dispositivo móvil.

Para ilustrar lo anterior, la figura 1 muestra el diagrama general del trabajo. Como podemos observar las partes principales que conforman el método propuesto, son el nodo central, el nodo cliente y la interfaz con el usuario. Dichos elementos se describen

enseguida.

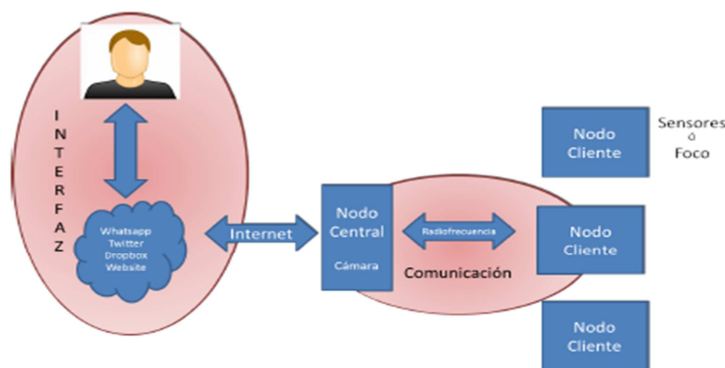


Fig. 1. Diagrama a bloques del sistema en general.

2.1. Nodo central

El nodo central es el dispositivo que se encarga de la toma de decisiones del sistema, las funciones que realiza son las siguientes:

- Recibir y enviar información de los nodos cliente.
- Recibir instrucciones del usuario por medio de internet.
- Enviar notificaciones al usuario mediante la plataforma de internet.

Teniendo en cuenta lo anterior, el nodo central se compone de los siguientes dispositivos: raspberry pi A+, electric imp y el módulo de radiofrecuencia NRF24L01. La primera función del nodo cliente se realiza mediante el módulo de radiofrecuencia y un microcontrolador PIC 16f8777a usando el protocolo de comunicación SPI; la información recibida es la lectura de los sensores y la información enviada son instrucciones lógicas que permiten encender y apagar determinados focos. El dispositivo electric imp que es un microcontrolador de 32 bits con WIFI integrado que se encarga de recibir las instrucciones desde el sitio de internet, con base en éstas se controla el encendido de los focos y se activa la emisión de video en tiempo real. Finalmente la raspberry pi se encarga del sistema de notificaciones, el cual consiste en los siguientes: cada vez que se active un sensor (PIR, timbre o magnético) el sistema envía un mensaje de whatsapp y actualiza un tweet indicando al usuario cuál de los sensores fue activado mediante una imagen y texto, posteriormente el sistema captura una imagen y se graba un video corto de la casa y ambos se almacenan en una cuenta de dropbox. La figura 2 muestra el dispositivo central con los dispositivos mencionados y sus respectivas conexiones, incluyendo una etapa de acoplamiento de voltajes.

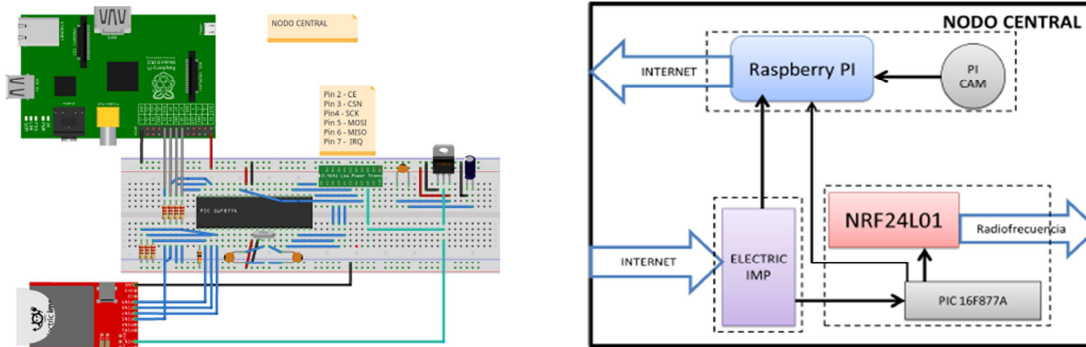


Fig. 2. Diagrama a bloques e implementación del nodo central.

2.2. Nodo cliente

El nodo cliente tiene dos funciones, la primera es recibir las instrucciones del nodo central y dependiendo de esta encender o apagar un foco, la segunda función consiste en detectar las señales de los sensores y posteriormente enviar dicho dato al nodo central, el cual activa el sistema de notificaciones. En otras palabras, el nodo cliente realiza las acciones dentro de la casa y al mismo tiempo se encarga del monitoreo mediante sensores.

Debido a la característica full dúplex del módulo NRF24L01, este es el principal componente del nodo cliente. También, se agregan dos elementos: la entrada de los sensores y una etapa de potencia en la salida. Por un lado la lectura de los sensores se realiza por medio de interrupciones, es decir, cada vez que se activa algún sensor, el microcontrolador recibe un estado lógico. Por otro lado, la etapa de potencia se conforma de un TRIAC y un opto-acoplador. Finalmente, es importante mencionar que se pueden manipular hasta 32 nodos cliente. En resumen la figura 3 muestra el esquema de conexiones del nodo cliente. La figura 6 muestra la implementación física del nodo cliente y central.

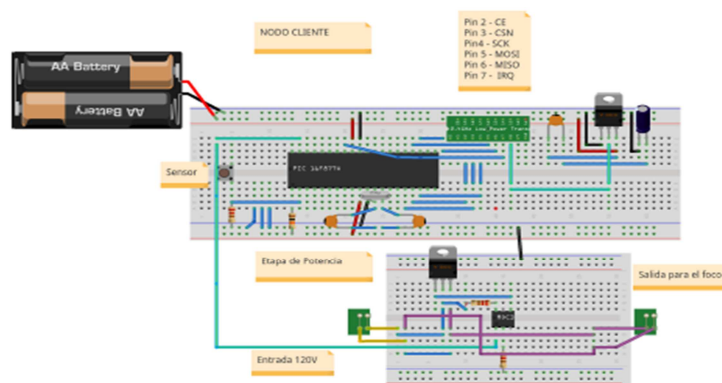


Fig. 3. Implementación del nodo cliente.

2.3. Interfaz

La interfaz entre el usuario y el hogar debe cumplir una característica fundamental, compatibilidad. Al tratarse de un sistema de notificaciones, el usuario debe monitorear su hogar y enviar órdenes a éste desde cualquier lugar, por ende, la interfaz debe ser compatible con cualquier dispositivo móvil. Con este fin, se realizó una página web programada en HTML5, la cual consta de dos partes: control de luces y sistema de seguridad. La primera de ellas permite al usuario encender o apagar tres focos de su hogar de la manera más intuitiva posible. La segunda sección de la página web, permite al usuario ver un video en tiempo real de lo sucedido en la casa y capturar una imagen. La figura 4 muestra el sitio web con las dos secciones mencionadas.



Fig. 4. Sitio web, control de luces y sistema de seguridad.

Uno de los aspectos que se tuvieron que resolver en el desarrollo del sistema fue la programación de cada uno de los dispositivos. Lo anterior debido a que diferentes sistemas de programación son involucrados. La tabla 1 describe los lenguajes de programación que se utilizaron para cada una de las aplicaciones involucradas.

Aplicación	Lenguaje de Programación	Librerías adicionales	Dispositivo
Sistema de notificaciones	Python	Tweepy, yowsup, Dropbox - uploader, raspicam	Raspberry pi
Recepción de instrucciones de internet	Squirrel		Electric Imp
Envío y recepción de datos por radiofrecuencia	C	nRF24L01.h	PIC 16f877a
Envío de instrucciones por internet	Javascript		Celular, Tablet, Laptop, PC.
Página Web (Interfaz)	HTML5		Celular, Tablet, Laptop, PC.

Tabla 1. Descripción de lenguajes de programación utilizados.

3. Resultados

Para validar el funcionamiento del sistema se realizaron diferentes experimentos que cubren tres aspectos, confiabilidad del sistema de notificaciones, eficiencia de la comunicación inalámbrica, y respuesta del sistema a las órdenes del usuario mediante la página web. Para el primer punto, se experimentó activando el sistema de notificaciones bajo diferentes condiciones, en las cuales se analizó el tiempo de respuesta y la veracidad de cada una de las notificaciones; para esto se activó cada uno de los sensores y posteriormente se intercalaron, la tabla 2 muestra los resultados obtenidos. De manera similar, para el segundo experimento se analizó la eficiencia de la comunicación entre el nodo central y el nodo cliente en un determinado tiempo y con base en lo siguiente, la distancia entre los dispositivos y la veracidad de la respuesta en diferentes escenarios y horarios. Los diferentes escenarios se seleccionaron con el fin de validar la robustez del sistema, es decir, que funcione en diferentes horarios y no presente problemas de interferencia con otras señales de radiofrecuencia tal como se muestra en la tabla 3. Finalmente, para comprobar el funcionamiento del tercer aspecto en términos de compatibilidad, se enviaron ordenes al sistema en diferentes

dispositivos móviles mediante red satelital y wifi durante diferentes intervalos de tiempo y se anotaron cuales acciones se pudieron ejecutar; la tabla 4 muestra los resultados obtenidos. Por otro lado, la figura 5 muestra la implementación de la interfaz en dos dispositivos móviles.

Condición	Estado de Twitter	Mensaje de whatsapp	Actualización de archivos en Dropbox	Mensaje correcto
Sensor de presencia activo	2s	10s	180s	Si
Sensor magnético activo	3s	12s	185s	Si
Timbre Activo	2s	9s	181s	Si
Primero se activa sensor de presencia y 4 minutos después el timbre	1er mensaje: 2s 2do mensaje: 3s	1er mensaje: 11s 2do mensaje: 10s	1er mensaje: 179s 2do mensaje: 185s	Ambos mensajes correctos
Primero se activa el timbre y 2 minutos después el magnético	1er mensaje: 3s 2do mensaje: No recibido	1er mensaje: 9 2do mensaje: No recibido	1er mensaje: 183s 2do mensaje: No recibido	Solo primer mensaje correcto, el segundo no fue recibido

Tabla 2. Resultados obtenidos de tiempo de ejecución y veracidad de respuesta del sistema de notificaciones.

Distancia entre módulos	Distancia entre nodo central y aparato rf	Distancia entre nodo cliente y aparato rf	Horario	Veracidad de datos (r= recibido, n=no funcionó)	Tiempo de respuesta (Promedio)
1m	2,10,20 (cm)	2,10,20(cm)	10 hrs.	n,n,n	2s
2m	2,10,20 (cm)	2,10,20(cm)	12 hrs.	n,r,r	3s
5m	2,10,20 (cm)	2,10,20(cm)	14 hrs.	r,r,r	1.5s
9m	2,10,20 (cm)	2,10,20(cm)	16 hrs.	r,r,r	3s
12m	2,10,20 (cm)	2,10,20(cm)	18 hrs.	r,r,r	2.5s
14m	2,10,20 (cm)	2,10,20(cm)	20 hrs.	r,r,r	1.2s
16m	2,10,20 (cm)	2,10,20(cm)	21 hrs.	n,n,n	1.3s
20m	2,10,20 (cm)	2,10,20(cm)	23 hrs.	n,n,n	1.4s

Tabla 3. Validación de la comunicación inalámbrica entre nodo central y cliente, en términos de veracidad y tiempo de respuesta.

Dispositivo	Red Celular	Señal Wifi	Video en tiempo real	Control de luces	Captura de imagen
Laptop	No	Si	Correcto	Correcto	Correcto
Alcatel one touch	Si	No	Incorrecto	Correcto	Correcto
Samsung Galaxy s3	Si	No	Correcto	Correcto	Correcto
Tablet Acer Iconia 7	No	Si	Correcto	Correcto	Correcto

Tabla 4. Validación de la interfaz del sistema en términos de compatibilidad.

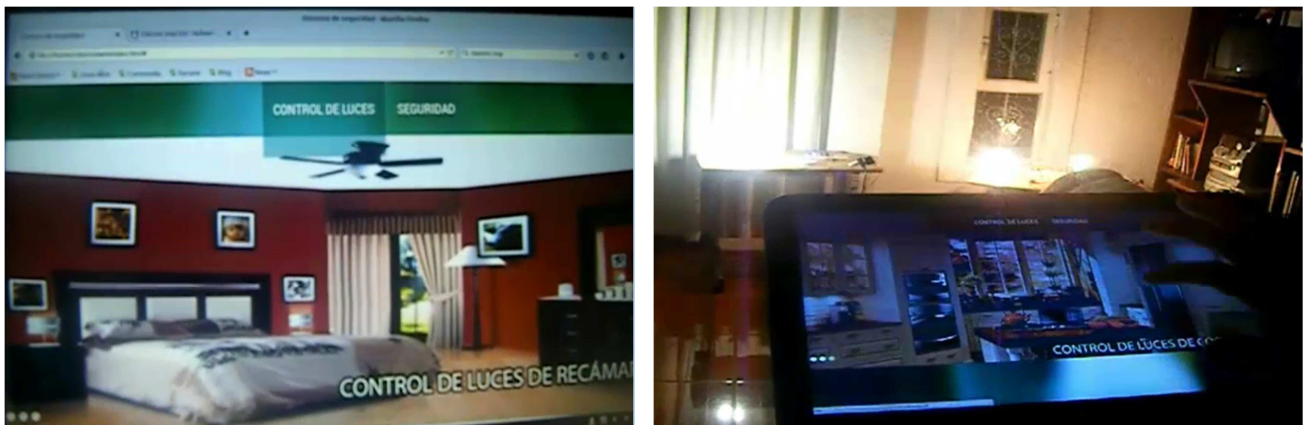


Fig. 5. Compatibilidad del sistema con diferentes dispositivos móviles, la imagen izquierda en una laptop y la derecha en una tablet.

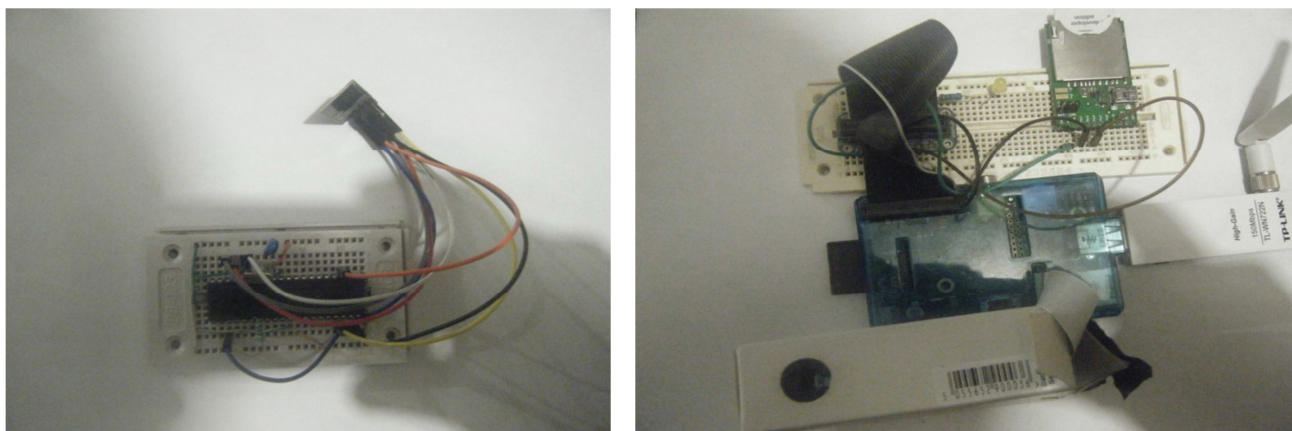


Fig. 6. Nodo Cliente y nodo central respectivamente.

4. Conclusiones

Se implementó un sistema de domótica mediante la sinergia entre plataformas de internet, sistemas embebidos y comunicación inalámbrica. El sistema está orientado al control de luces y sistema de vigilancia usando una interfaz web compatible con cualquier dispositivo móvil. La validación del sistema se realizó en términos de veracidad de la respuesta, tiempo de ejecución y robustez. En este sentido, los resultados muestran que la implementación del sistema es viable en el interior de una casa donde la distancia entre el nodo central y clientes no supere los 14 metros. Así mismo, otra restricción del sistema es que no se puede enviar información simultánea entre los nodos, es decir, se puede encender un foco a la vez y no se pueden recibir dos notificaciones de sensores diferentes al mismo tiempo. Sin embargo, se demuestra que con base a la estrategia de implementación presentada, se puede realizar un sistema de domótica robusto y de bajo costo.

Debido a que el sistema propuesto depende de la conexión de internet, el trabajo futuro incluye agregar una etapa de verificación de conexión a la red, es decir, cuando el sistema no esté conectado a internet, tanto las imágenes como los mensajes de texto se almacenen y se envíen al usuario una vez que se restablezca la conexión. De igual

manera, el sistema puede modificarse para que no solo sea una aplicación de vigilancia de hogares, sino que éste pueda funcionar como monitoreo inteligente de viveros o campos de cultivo.

5. Referencias

- [1] K. Ashton, "Internet of Things". RFID Journal. 22 de Junio 2009.
- [2] S. Goodwin, Appliance Control: Making Things do Stuff. In Smart Home Automation with Linux and Raspberry pi. 2da edición. 2013. Appress. Nueva York. 297 p.
- [3] C. Hsu, S.Yang, "Intelligent Home Appliance control system with refrigerator" Expert Systems with Applications. 2010.
- [4] S. Ferdoush, "Wireless Sensor Network System Design with Raspberry Pi and Arduino Environmental Monitoring Applications". Procedia Computer Science. 2014.
- [5] N. Sriskanthan. "Bluetooth based home automation system". Microprocessors and Microsystems. 2002.
- [6] L. De Silva, C. Morikawa, "State of art of smart homes". Engineering Applications of Artificial Intelligence. 2012.

6. Autores

Roberto de Lima Hernández, estudiante de la licenciatura en Electrónica en la Benemérita Universidad Autónoma de Puebla.

Dr. Aldrin Barreto Flores, estudió en el Instituto Tecnológico de Veracruz obteniendo el grado de Ingeniero en Electrónica en el año de 1998. Realizó estudios de Maestría en Ciencias de la Electrónica en el Instituto Nacional de Astrofísica, Óptica y Electrónica en el año 2000, posteriormente en la misma Institución obtuvo el grado de Doctor en Ciencias en el área de Ciencias de la Computación en el año 2005. Sus áreas de interés son visión por computadora, sistemas embebidos y aplicaciones en la industria.

M.C. Verónica Edith Bautista López, estudió en la Facultad de Ciencias de la Computación de la Benémerita Universidad Autónoma de Puebla, obteniendo el grado de Lic. en Ciencias de la computación en 2001, con la especialidad de programación de sistemas. En 2007 obtuvo el grado de Maestra en Ciencias de la Computación, con especialidad en sistemas distribuidos en la misma Institución. Actualmente es candidata a Dra. en Educación de las Ciencias, Ingenierías y Tecnologías por la Universidad de las Américas Puebla y sus áreas de interés son el desarrollo de aplicaciones móviles y criptográficas, sistemas embebidos, así como la evaluación del aprendizaje e innovación educativa.

Arquitectura genérica de una red en chip de enrutamiento unidireccional en FPGA

Jorge Ernesto López Arce Delgado

CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
je.lopezarce.d@gmail.com

Juan José Raygoza P.

CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
juan.raygoza@ucei.udg.mx,

Susana Ortega Cisneros

Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
susana.ortega@gdl.cinvestav.mx

Jorge Rivera D.

Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
riveraj@gdl.cinvestav.mx

Pablo Moreno Villalobos

Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
pmoreno@gdl.cinvestav.mx

Resumen

El constante aumento de los componentes que contiene un sistema *on-chip* ha incrementado la complejidad de comunicación entre los elementos de procesamiento (EPs) del sistema. Un recurso utilizado para disminuir la complejidad es el diseño de enrutamiento de conexiones (cableado), el cual ha sido suficiente para interconectar algunos EPs, dicho diseño se conoce como redes en chip o por sus siglas en inglés NoC (*Network on Chip*), de manera alternativa, enrutar paquetes permite una mayor escalabilidad de las redes, tener una latencia aceptable y una utilización moderada de

área. Sin embargo, las redes en chip (NoC) suelen ser implementadas en tecnologías rígidas y deterministas como los ASIC (Circuito Integrado de Aplicación Específica), limitando la flexibilidad, arquitectura y modularidad que ofrece una NoC de enrutamiento de paquetes. Este trabajo propone una arquitectura de una red en chip de switcheo o enrutamiento unidireccional utilizando un router genérico para topología de mariposa, de enrutamiento de paquetes, implementado en una FPGA de la familia Xilinx. Donde el diseño permite enviar paquetes desde 16 puntos de origen, hacia 16 puntos de destino, así como la flexibilidad de enviar paquetes de diferentes tamaños, divididos en flits. Este diseño tiene como resultado una arquitectura compacta, permitiendo dejar el mayor espacio posible para los EPs.

Palabra(s) Clave(s): arquitectura de router, control de flujo, FPGA, NoC, redes en chip.

1. Introducción

La característica principal de la tecnología de las FPGA (Field programmable Gate Array) es la posibilidad de ser reconfiguradas electrónicamente, contrario a otros circuitos integrados, como los ASICs, donde el hardware tiene funciones rígidas y deterministas. Esta flexibilidad permite a las FPGAs ser configuradas para aplicaciones computacionales como son el procesamiento de señales, el procesamiento de imágenes y la criptografía, típicamente tienen diseños con mejor desempeño en comparación con CPUs (*Central Processing Unit*) tradicionales[1,2]. La tecnología ha tenido una evolución tal, que en un futuro las FPGAs podrían llegar a tener millones de LUTs (*Look Up Tables*) dando la posibilidad de diseñar arquitecturas con muchos procesadores trabajando en paralelo, generando más lógica de circuitos, y a medida que la complejidad incrementa, la lógica de circuitos entre los procesadores hace que los diseños RTL (*Register File Level*) tradicionales sean ineficientes[1]. Para lograr aprovechar la evolución de las FPGAs es necesario dar énfasis al diseño y manejo de los recursos en un nivel de abstracción mayor, ejemplo de ello, sería considerar el diseño a nivel de función, es decir, si se desea implementar procesadores en paralelo,

pensar en su programabilidad para operar con otros simultáneamente, en vez de solo poder programar conexiones entre ellos. Un nivel de abstracción mayor presenta limitaciones en el uso de buses tradicionales y esquemas de interconexión punto a punto, en términos de la escalabilidad y complejidad del chip [3]. Como consecuencia, hoy en día se observa una tendencia de crear sistemas con arquitecturas complejas, velocidades de reloj mejoradas y aumento de la densidad de lógica en los chips[1]. Un nuevo paradigma de diseño que emerge son las Redes en Chip o NoC, por sus siglas en inglés, las NoC son una solución eficiente y económica para los requerimientos de comunicación aplicados en un chip. Las NoCs pueden ser diseñadas y construidas para ocupar menos área que un diseño conectado espacialmente y ofrece mejor eficiencia que un bus[4]. La red en si es una colección de interruptores conectados entre ellos usando canales cableados. Permitiendo a los EPs enviar tráfico de información a la red y llegar a su destino a través de *switches* [5].

El diseño de una red está definida por la forma en que se implementan tres elementos fundamentales que son la topología, enrutamiento y control de flujo de datos. [6]

La interconexión de red se implementa con una colección de nodos enrutadores distribuidos mediante canales compartidos. El patrón de conexión de estos nodos definen la topología de la red, y la topología de una interconexión de red es parecido a una carretera de autos, donde en los canales (como carreteras) transportan paquetes (los autos) de un nodo enrutador a otro (intersecciones), es decir es el camino por el cual los paquetes viajan. Es necesario definir el tipo de topología antes de elegir qué tipo de enrutamiento será seleccionado. El mensaje enviado por la red es entregado entre las terminales haciendo “saltos” a través de los canales y nodos compartidos desde su terminal-origen hasta su terminal-destino. Una vez que la topología es definida, aún falta seleccionar el camino que el mensaje va a tomar (la secuencia de salto en los nodos) en la red para llegar a su destino. El enrutamiento determina cuál es el camino que toma el mensaje, donde la extensión del camino impacta directamente en la latencia del mensaje a través de la red. El control de flujo de datos determina como

serán asignados los diferentes recursos de la red, ancho de banda del canal y estado de control para que los paquetes transiten. Un buen método de control de flujo debe repartir estos recursos de una manera eficiente facultando a la red de tener un alto porcentaje de su ancho de banda ideal y entrega de paquetes con una baja latencia predecible [5,6,7].

2. Desarrollo

Este trabajo presenta la arquitectura de una NoC con topología tipo mariposa implementada en una FPGA, la arquitectura se ha diseñado en lenguaje Verilog bajo el estándar IEEE 1364-2001. La tarjeta de desarrollo que se utilizó para evaluar este diseño es una kit Zynq-700 SoC ZC702, la cual cuenta con una FPGA XC7Z020, arquitectura basada en los SoCs de la familia de Xilinx, la cual tiene como características principales 106,400 Flip-Flops, 53,200 LUTs (*Look Up Tables*), 200 pines de I/O, 32 BUFG (Buffers globales), 140 bloques de RAM36/FIFO, 280 bloques de RAMB16, 220 DSPs. Esta tarjeta puede integrar diferentes elementos al diseño, muestra de ello es el elemento principal, pues esta tarjeta contiene dos CPUs de arquitectura ARM Cortex-A9 ubicado en la sección del sistema de procesamiento, la cual se puede habilitar para integrarse con módulos diseñados en la sección de lógica programable[8].

La arquitectura que se propone cuenta con 16 terminales-origen y 16 terminales-destino, en esta topología, desde un punto de vista de la terminal-origen hacia los posibles destinos se aprecia forma de árbol, donde en cada nivel de dicho árbol se encuentran nodos de *switcheo* (ver Fig. 1), estos nodos no envían ni reciben paquetes, solo los transmiten, hasta que lleguen a las terminales-destino. Los canales van en sentido de izquierda a derecha, como lo indican las flechas, donde los elementos de entrada se encuentran a la izquierda. En la Fig. 1 se observa que el diseño tiene 2 niveles o etapas de nodos de *switcheo*, y cada nodo tiene 4 canales de

entrada y 4 canales de salida de un ancho de 18 bits cada canal, siendo dos bits de cabecera.

El algoritmo de enrutamiento (routing) que se emplea en esta red en chip de switcheo es de tipo “etiqueta destino”. Donde los bits de destino son usados para seleccionar la salida de cada nodo. Se tiene dos niveles/ etapas de nodos, por lo tanto la dirección está constituida por cuatro bits, siendo el primer par los bits más significativos los que definen la salida del primer nivel de nodos (de izquierda a derecha), una vez que se “usan/leen” los dos bits de dirección en la primera etapa de nodos, se hace un corrimiento a la izquierda de dos posiciones en los bits de la dirección, quedando el segundo par de bits colocados para ser “usados/leídos”.

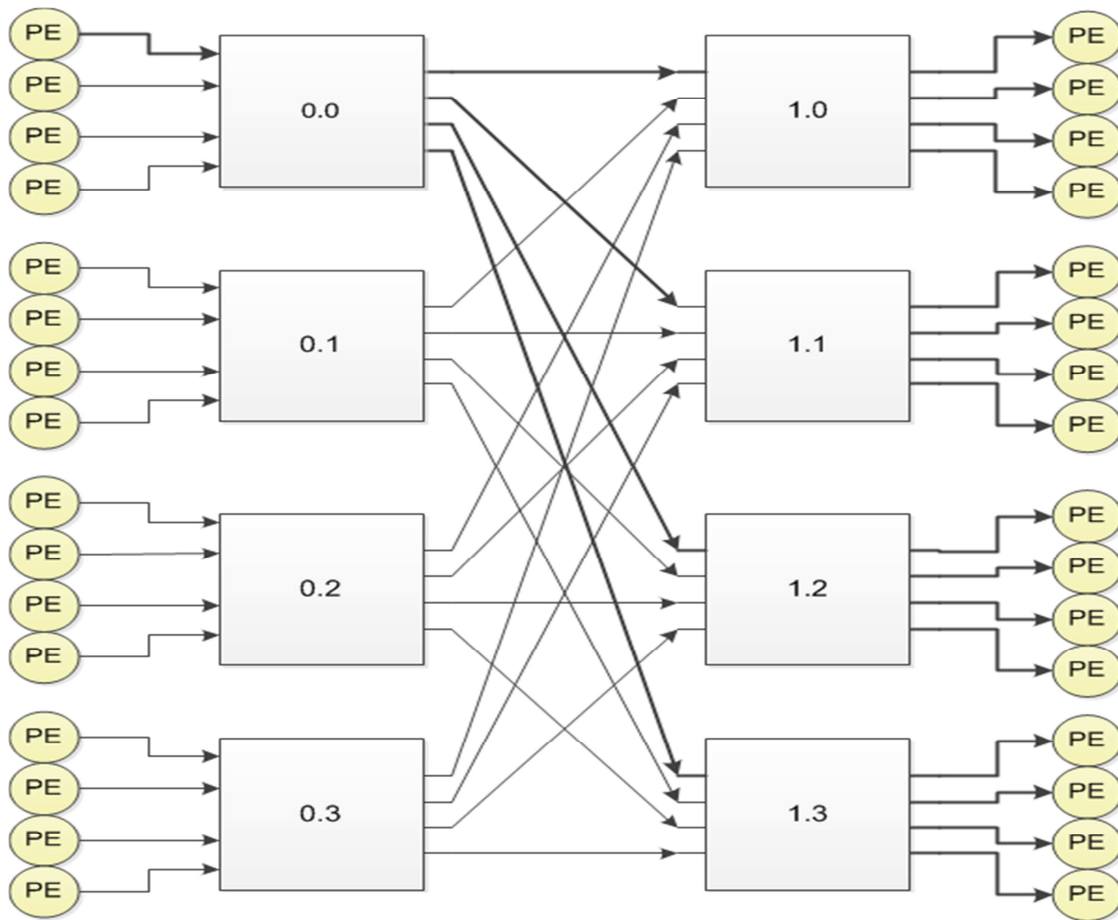


Fig. 11. Topología tipo mariposa.

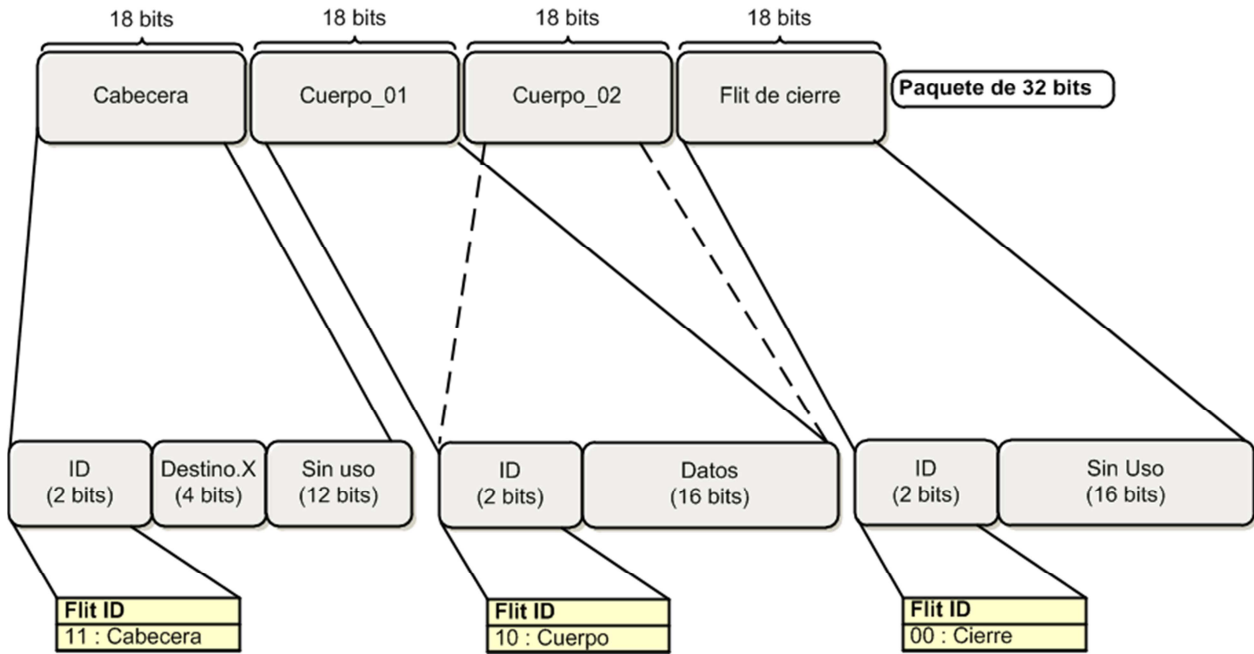


Fig. 12. Formato del paquete y formato de flit.

El control de flujo de datos en esta red de switcheo permite enviar entre 32 y 240 bits de datos. Se tiene un envío de 16 bits físicos por ciclo. Para poder enviar paquetes de ese tamaño se utiliza un protocolo, que permite dividir los paquetes en elementos más pequeños llamados flits, estos flits son de 18 bits, al igual que los canales de la red. Al ser un protocolo, se establecen diferentes tipos de flits para poder enviar no solo los datos del paquete sino también la información del ruteo, como bits de identificación del flit y la dirección de envío. Existen tres tipos de flits: de cabecera, de cuerpo y flits nulos (*Idle*). Los flits de cabecera tienen los primeros dos bits (bits más significativos) como bits de identificación,. Se puede observar en la Fig. 2, que el flit de cabecera contiene la información del destino, los flits de cuerpo los datos que conforman el paquete, y el flit nulo indicando el fin del envío del paquete así como el cese del uso del canal.

Diseño del router

Cada nodo de esta red es un router capaz de recibir paquetes en sus entradas, determinar su destino basado en su algoritmo de enrutamiento, y pasar los paquetes a

la salida correspondiente. La ruta de los datos dentro del router consiste en cuatro entradas de 18 bits, cuatro multiplexores 4:1 de 18 bits, cuatro módulos de desplazamiento (que realiza el desplazamiento de la dirección destino en cada nivel) y cuatro registros de salida de 18 bits (ver Fig. 3).

Los flits llegan en cada ciclo de reloj al registro de entrada y son enrutados a los cuatro multiplexores. Cada multiplexor está asociado a un módulo llamado “asignador”, donde se examinan los bits que definen el tipo de flit recibido y los siguientes dos bits, en el caso de que sea flit de cabecera, establece el *switcheo* correspondiente, así como envía la señal de habilitación al módulo de corrimiento para que modifique el campo de la dirección de destino y quede listo para el siguiente nivel disponible. Los flits de cuerpo pasan sin cambios.

El control del router está esencialmente en los cuatro “asignadores” asociados a cada salida, y a su vez, cada uno controla el selector del multiplexor y el módulo de corrimiento correspondiente. Cada “asignador”, destina un puerto de salida con un puerto de entrada. Un “asignador” consta internamente de cuatro partes iguales, cada uno dividido en tres secciones, en la primera se realiza la decodificación, en la segunda el arbitraje y en la tercera se define si se mantiene el canal o no. En la decodificación se toman los cuatro bits más significativos de entrada (dos del tipo de flit y dos de dirección) , cada decodificador genera dos señales, una se activa cuando el flit es de cabecera y el otro cuando los siguientes dos bits coinciden en representar el número de puerto de salida.

Estas señales indican que el flit de entrada hace la petición de usar el canal, así como cierto puerto de salida para pasar por ahí un paquete de datos. Cuando llega un flit de cuerpo el decodificador activa y envía una señal de retención a la tercera sección, indicando que el canal se mantiene pues el paquete aún no termina de pasar. La segunda sección es donde actúa el módulo de arbitraje de cuatro entradas y de prioridad fija, prioridad que otorga al canal de menor valor, es decir que ‘entrada_0’ es la de mayor prioridad y la ‘entrada_3’ la de menor prioridad.

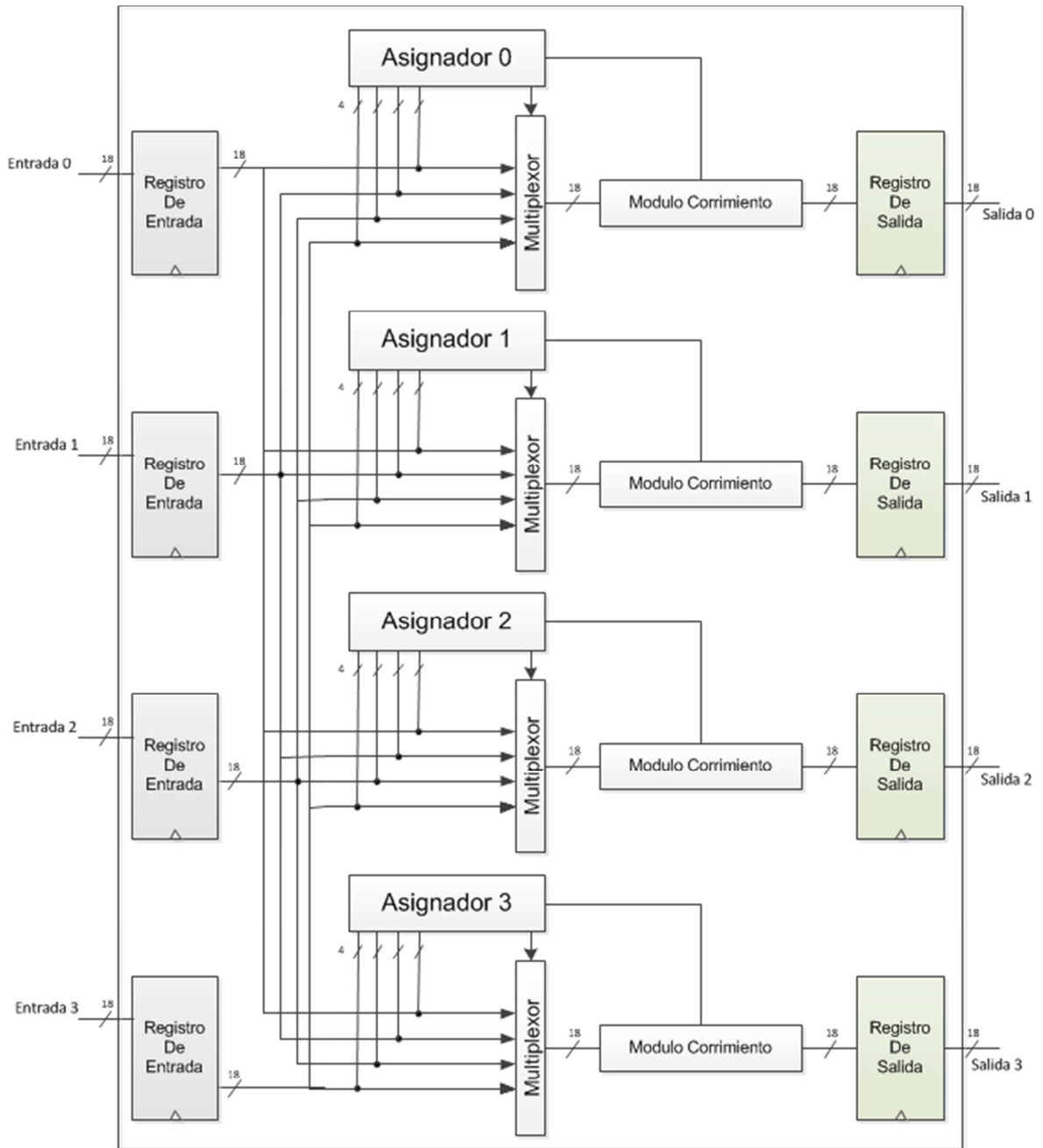


Fig. 13. Módulos internos del router.

El árbitro permite cuatro peticiones y genera cuatro señales de concesión de aprobación. Si el puerto de salida que se pide está disponible, el árbitro activa la señal de concesión al puerto de entrada que haya hecho primero la petición y tenga prioridad

mayor. Una vez que se asegura la señal de asignación, provoca que se garantice la señal de selección que indica la salida que tendrá el multiplexor. Si una señal de asignación se ratifica, indica que un flit de cabecera está pasando por el nodo, por lo tanto tiene que ser modificado por el módulo de corrimiento y envía la señal de habilitación hacia ese módulo. La última sección del “asignador” es el de retención del canal, el cual fija el camino entre el puerto de salida y entrada hasta que pase el paquete completo. También se tiene una señal de disponibilidad, la cual es deshabilitada cuando se tiene un canal activo en el ciclo anterior, indicando que aún no termina de pasar el paquete, y evita que se asigne el puerto a una nueva cabecera.

3. Resultados

En el análisis de desempeño, se consideran tres parámetros, latencia, costo de ocupación y rendimiento. La latencia es el tiempo que tarda un paquete en pasar a través de la red, el rendimiento es la cantidad de bits por segundo que la red puede transportar desde la entrada a la salida y el costo de ocupación se mide en el total de recursos de hardware utilizados por el diseño final en la FPGA.

Para evaluar el diseño se realizó un análisis de envío de paquetes de 32, 64, y 240 bits, se utilizan memorias precargadas con los datos que conforman los paquetes, se empleo un módulo externo a la arquitectura de la NoC el cual se encuentran las memorias (EPs) y se encarga de leer en cada ciclo de reloj un registro de memoria y es enviado a una entrada de la red, las memorias tienen paquetes armados para ser enviados a la red, dichos paquetes tienen la información de direccionamiento para que tomen rutas diferentes, el orden de envío es ascendente en los nodos de entrada, iniciando en la entrada 0 y descendente en el orden de salida, iniciando en la entrada 15, esto es que los paquetes enviados por la entrada 0 se envían a la salida 15, los que son enviados por la entrada 1 llegan al nodo de salida 14, así sucesivamente.

En la Fig. 4 se muestra la latencia del envío de 16 paquetes a distintas salidas (oR) con un ciclo de reloj de 2ns, usando todos los nodos de entrada, salida y canales de la red.

La Fig. 5 muestra el intervalo de 6 ns que hay entre la recepción de un paquete y otro en diferentes nodos de salida, revelando el tiempo de latencia en condiciones ideales donde siempre hay tráfico de paquetes en la red.

La intención es mostrar la fiabilidad de envío y recepción de paquetes en el mejor y peor caso, es decir, enviar y recibir los paquetes con la menor cantidad de bits (32 bits) en el periodo de tiempo más largo (50ns), así como envío y recepción de los paquetes de datos más grandes (240 bits) en el periodo de tiempo más corto (2ns) que permite la red. Los periodos de tiempo probados fueron de 50ns, 25ns, 10ns, 4ns, 2ns hasta 1ns con un ciclo de trabajo de 50/50.

El tiempo total al enviar 16 paquetes de 240 bits cada uno, por entradas diferentes a salidas diferentes es de 580ns. La latencia de cada paquete es de 40ns desde el momento en que se envía el dato a la entrada hasta la salida, donde 10ns es lo que tarda la red en hacer la asignación del canal y enviar un paquete desde su origen a su destino, al enviar los datos en una condición óptima donde la red está siempre ocupada, se aprovecha el rendimiento de envío de datos, observando a la salida entre un paquete y otro de 6ns (ver Fig. 5). Tomando los 40ns como latencia total del envío de paquetes de 240 bits, se tiene una velocidad de transmisión de $240\text{bits}/40\text{ns} = 750\text{MB/s}$.

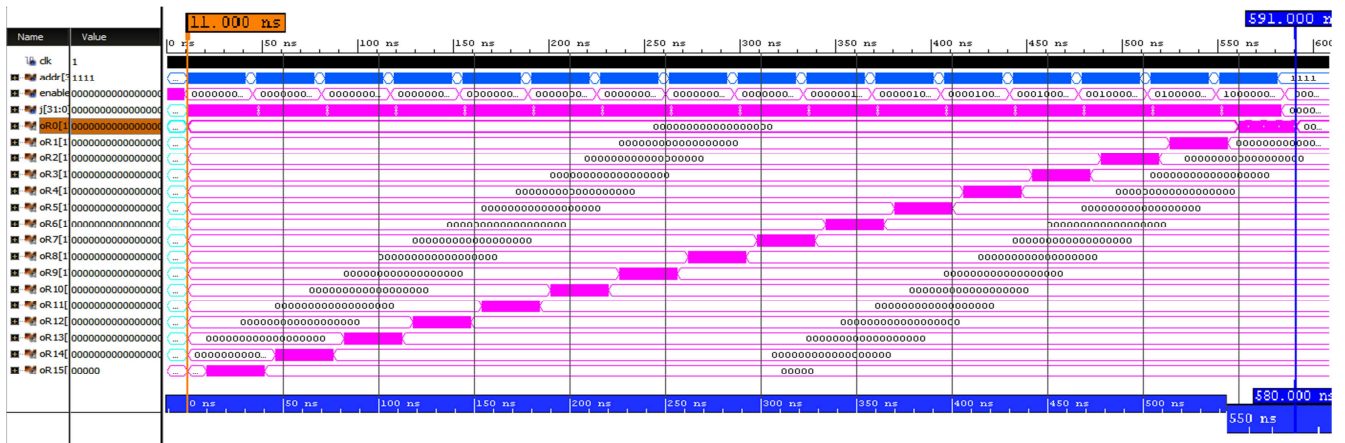


Fig. 14. Envío de 16 paquetes a 16 salidas con periodo de 2ns.

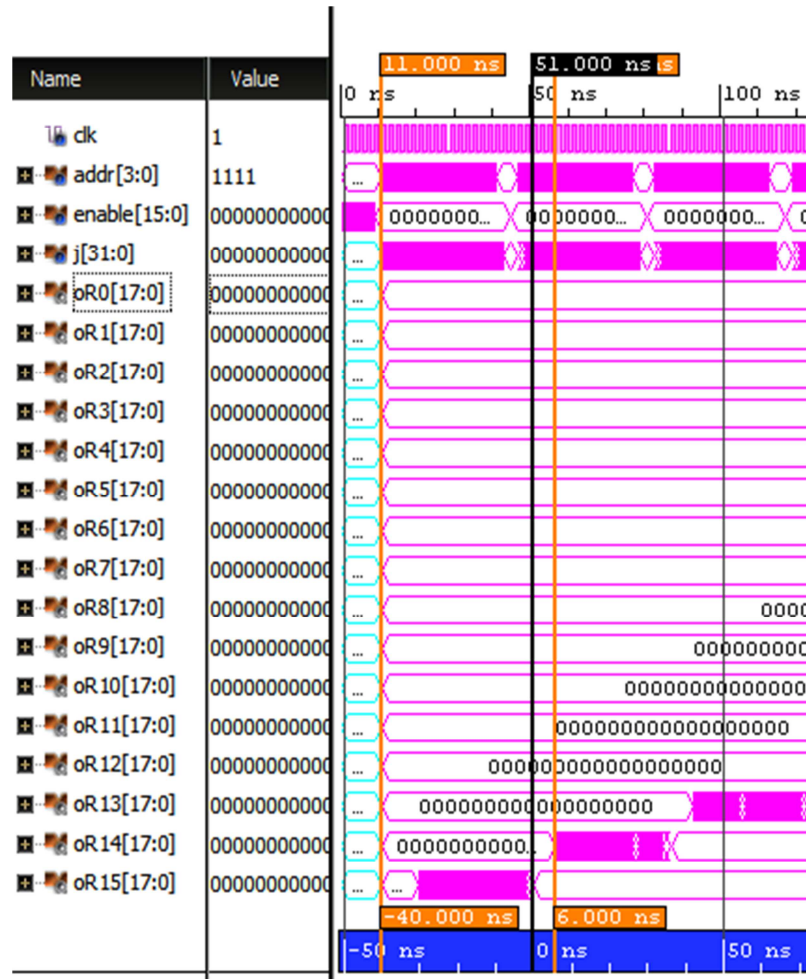


Fig. 15. Retardo de 6ns en el envío entre un paquete y otro.

En la Fig. 6 se muestra la correcta entrega de paquetes a la salida, se observa que 10ns después (línea vertical azul) de que la señal ‘addr’ (fila azul) envía “0000”, que representa la primera dirección a leer de la memoria, se recibe el flit de cabecera en la terminal-destino ‘oR0’ (fila amarilla), donde después de haber recorrido los bits de destino en dicho flit llega al destino solo con los bits que lo identifican que es un flit de cabecera, el siguiente flit contiene el identificador del flit “10” y una secuencia de 4 veces “1100”, con ello se corrobora la asignación adecuada del canal para dicho paquete de datos, y que se reciben de forma completa y sin errores.

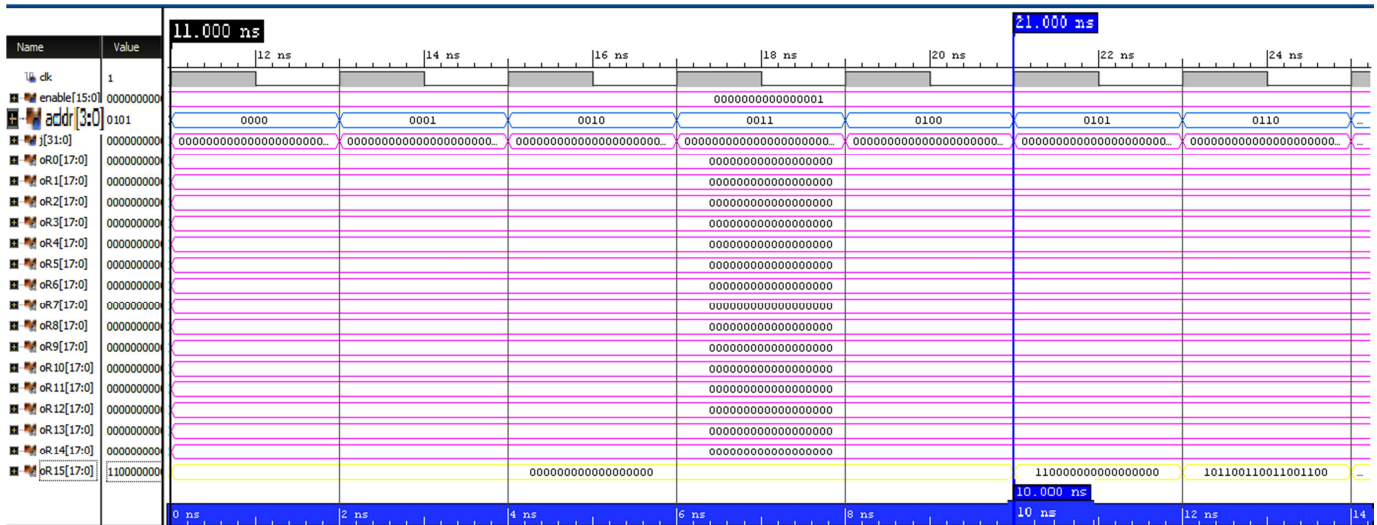


Fig. 16. Recepción correcta de los flits.

Se anexa una síntesis de los resultados de las pruebas realizadas, las que consisten en enviar los diferentes tamaños de paquetes a través de la red utilizando distintos periodos de tiempo, obteniendo en los casos de periodos de tiempo de 50ns hasta 2ns la recepción íntegra de la información enviada. Esta información presenta el comparativo de la latencia usando diferentes periodos de tiempo con diferentes tamaños de paquetes, se anexan los resultados en la Fig 7 y Tabla 1.

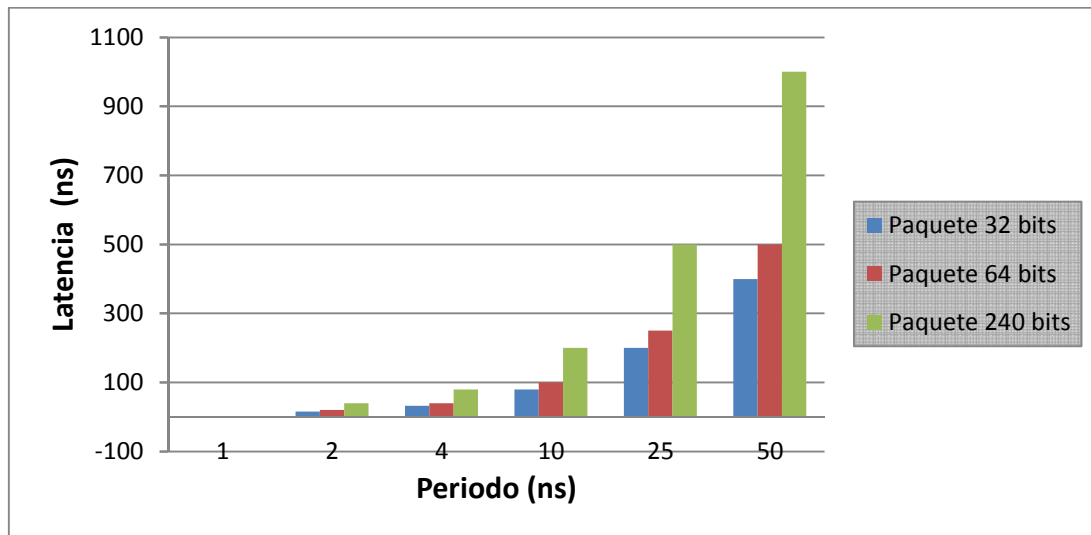


Fig. 17. Relación tamaño de los paquetes enviados contra latencia.

Periodo	Paquete (bits)	Latencia por paquete	Costo de Ocupación	Envió completo
50ns	240	1000ns	LUTs-> 816(1.53%)	Si
	64	500ns	FF -> 18 (0.67%)	Si
	32	400ns	BFG -> 1(3.12%)	Si
25ns	240	500ns	LUTs-> 816(1.53%)	Si
	64	250ns	FF -> 18 (0.67%)	Si
	32	200ns	BFG -> 1(3.12%)	Si
10ns	240	200ns	LUTs-> 816(1.53%)	Si
	64	100ns	FF -> 18 (0.67%)	Si
	32	80ns	BFG -> 1(3.12%)	Si
4ns	240	80ns	LUTs-> 816(1.53%)	Si
	64	40	FF -> 18 (0.67%)	Si
	32	32ns	BFG -> 1(3.12%)	Si
2ns	240	40ns	LUTs-> 816(1.53%)	Si
	64	20ns	FF -> 18 (0.67%)	Si
	32	16ns	BFG -> 1(3.12%)	Si
1ns	240	N/A	N/A	No
	64			No
	32			No

Tabla 1. Resultados del análisis y evaluación de la red.

4. Conclusiones

El diseño presentado en este trabajo tiene como resultado la arquitectura de una NoC con muy poco nivel de ocupación en el dispositivo FPGA, menor al 2%, lo que deja la mayor lógica programable a disposición del diseño de los elementos de procesamiento, los cuales pueden ser conectados a la red e intercomunicarse entre ellos hasta una tasa de transferencia de 750 MB/s. Esta arquitectura ha sido diseñada de forma genérica, esto es que no tiene elementos particulares de la tarjeta donde se ha implementado permitiendo que el diseño pueda ser exportado a cualquier dispositivo FPGA, así como facilitar la escalabilidad a sistemas con más nodos de entrada, salida, y continuar explotando las características de las FPGAs. El diseño propuesto también muestra la versatilidad del enrutamiento de paquetes, iniciando la propagación del paquete en los nodos que han liberado el canal a pesar que no haya concluido su llegada a la terminal-destino. También, es posible notar que al establecer un periodo de 1ns, no hay resultados de latencia, esto es debido a que las características del sistema implementado no soporta velocidades tan altas, para poder lograr una tasa de transferencia mayor a las ya obtenidas, una opción puede ser aumentar el tamaño de los flits que se transmiten, teniendo un impacto directo en la ocupación que el diseño tiene en la FPGA.

6. Referencias

- [1] M. Lanzagorta, S. Bique, R. Rosenberg, R. O. Rosenberg, "Introduction to Reconfigurable Supercomputing". Morgan & Claypool Publishers. Vol. 4. No. 1. Enero 2009.
- [2] N. Kapre, N. Mehta, M. DeLorimier, R. Rubin, H. Barnor, M. Wilson, M. Wrighton, and A. DeHon, "Packet Switched vs. Time Multiplexed FPGA Overlay Networks", 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. Apr. 2006. 205-216 pp.
- [3] Ye Lu, J. McCanny, S. Sezer, "Generic Low-Latency NoC Router Architecture for FPGA Computing Systems". Field Programmable Logic and Applications (FPL). 2011. 82-89 pp.

- [4] W. J. Dally, "Route packets, not wires: on-chip interconnection networks". Proceedings of the 38th Design Automation Conference IEEE. No. 01CH37232.
- [5] N. Kapre, "Packet-switched on-chip FPGA overlay networks". PhD. Dissertation. 2006. 6-30 pp.
- [6] W.Dally, B.Towles, Principles and Practices of Interconection Networks. 1º edición. 2004. Morgan Kaufman Publishers. San Fransico. 13-36 pp.
- [7] G. Dimitrakopoulos, A. Psarras, I. Seitanidis, Microarchitecture of Networks on Chip Routers a designer's perspective. 1º edicion. 2015. Springer. Londres. 11-18, 61-71 pp.
- [8] Zynq-700 All Programmable SoC Overview, Preliminary Product Specification. Xilinx. Estados Unidos de America. 2014.

Monitor de alerta de fugas de gas

José Ignacio Vega Luna

Universidad Autónoma Metropolitana-Azcapotzalco, Av. San pablo 180, Col. Reynosa Tamaulipas,
México D.F., Teléfono: 55-53189552
vji@correo.azc.uam.mx

Mario Alberto Lagos Acosta

Universidad Autónoma Metropolitana-Azcapotzalco, Av. San pablo 180, Col. Reynosa Tamaulipas,
México D.F., Teléfono: 55-53189552

Gerardo Salgado Guzmán

Universidad Autónoma Metropolitana-Azcapotzalco, Av. San pablo 180, Col. Reynosa Tamaulipas,
México D.F., Teléfono: 55-53189552

Víctor Noé Tapia Vargas

Universidad Autónoma Metropolitana-Azcapotzalco, Av. San pablo 180, Col. Reynosa Tamaulipas,
México D.F., Teléfono: 55-53189552

Resumen

En este trabajo se presenta el diseño e implantación de un sistema de alerta de fugas de gas. Esta aplicación está diseñada principalmente para unidades habitacionales. Usando un sensor de gas se toman muestras del aire del ambiente en un departamento y se transmiten inalámbricamente usando transceptores ZigBee al microcontrolador de un módulo central supervisor de la unidad habitacional. Si la concentración de gas en el aire es mayor al umbral establecido por el usuario, se genera una alerta audible. Las alertas se almacenan históricamente en una memoria del tipo Micro SD del módulo central. Las alertas se muestran en una pantalla LCD del módulo supervisor. La implantación y operación del sistema aumentó la seguridad familiar de los habitantes en el inmueble y la confianza para la prevención de desastres.

Palabra(s) Clave(s): fugas de gas, micro SD, ZigBee.

1. Introducción

El gas natural y el gas L.P. (gas licuado de petróleo) son una fuente de energía que es comúnmente usada en hogares. El gas natural es una mezcla de hidrocarburos simples que se encuentra en estado gaseoso, bajo condiciones ambientales normales de presión y temperatura [1]. El gas natural es más ligero que el aire (su densidad relativa es 0.61, aire=1.0) y a pesar de sus altos niveles de inflamabilidad y explosividad las fugas o emisiones se disipan rápidamente en las capas superiores de la atmósfera, dificultando la formación de mezclas explosivas en el aire. Esta característica permite su preferencia y explica su uso cada vez más generalizado en instalaciones domésticas e industriales y como carburante en motores de combustión interna. Presenta además ventajas ecológicas ya que al quemarse produce bajos índices de contaminación, en comparación con otros combustibles.

El nombre químico del gas natural es metano (CH_4), ya que su principal componente es el metano, quien es la molécula más simple de los hidrocarburos. Puede contener pequeñas cantidades de etano (C_2H_6), propano (C_3H_8) y otros hidrocarburos más pesados, aunque también se pueden encontrar trazas de nitrógeno, bióxido de carbono, ácido sulfhídrico y agua. El gas natural no tiene color, sabor, ni olor, por lo que es necesario administrar un odorífico para advertir su presencia en caso de fuga. Debe mantenerse alejado de fuentes de ignición, chispas, flama y calor. Las conexiones eléctricas domésticas o carentes de clasificación son las fuentes de ignición más comunes. Se debe evitar el manejo del gas natural en espacios confinados ya que desplaza al oxígeno disponible para respirar. Su olor característico, por el odorífico utilizado, puede advertir de la presencia de gas en el ambiente; sin embargo, el sentido del olfato se perturba, a tal grado, que es incapaz de alertar cuando existen concentraciones peligrosas [2].

Ya que el gas natural está compuesto principalmente de gas metano y el metano es un compuesto químico altamente inflamable que consiste de una molécula de carbón rodeada por cuatro átomos de hidrógeno, una fuga de gas natural es peligrosa debido a aumenta el riesgo de incendio o explosión, y ser más ligero que el aire, tiende a subir y almacenarse en zonas con poca ventilación. En la Tabla 1 se indican los componentes principales del gas natural.

Componente	%	Número CAS (Chemical Abstracts Service)	LEP (Límite de Exposición Permisible)
Gas Natural (Metano)	88	74-82-8	Asfixiante simple
Etano	9		
Propano	3		
Etíl Mercaptano	17-28 ppm		Odorífico

Tabla 1. Componentes principales del gas natural.

Dado que el metano y, por lo tanto, el gas natural no tienen ningún olor, la compañía de gas adiciona una "advertencia" con olor a huevo podrido (mercaptano, un compuesto hecho básicamente con sulfuro) que puede ser detectado fácilmente por la mayoría de las personas. Sin embargo, existen personas que tienen un sentido del olfato disminuido y por tanto no pueden confiar en este mecanismo de seguridad. Un detector de gas es una importante herramienta que ayuda a detectar de fugas [3]. Se pueden usar varios sensores en un hogar, oficina o empresa, de tal forma que al detectarse una fuga, se envíe una señal a un dispositivo de monitoreo que alerte de la misma. En la Fig. 1 se muestra la instalación recomendada de una salida de ventilación en lugares donde existe suministro de gas natural.

Una de las tecnologías de comunicación más comúnmente usada en la actualidad es ZigBee. Esta tecnología es inalámbrica, lo cual implica que no se instale cableado adicional desde el sensor al dispositivo de monitoreo disminuyendo costos y aumentando la facilidad de instalación [4].

ZigBee es una tecnología de comunicación inalámbrica basada en el estándar IEEE 802.15.4 [5]. Fue diseñada para realizar redes de malla con capacidades de ruteo, por lo que se pueden implantar Redes de Área Personal Inalámbrica (Wireless Personal Area Networks-WPAN) y Redes Inalámbricas de Sensores (Wireless Sensor Network-WSN).

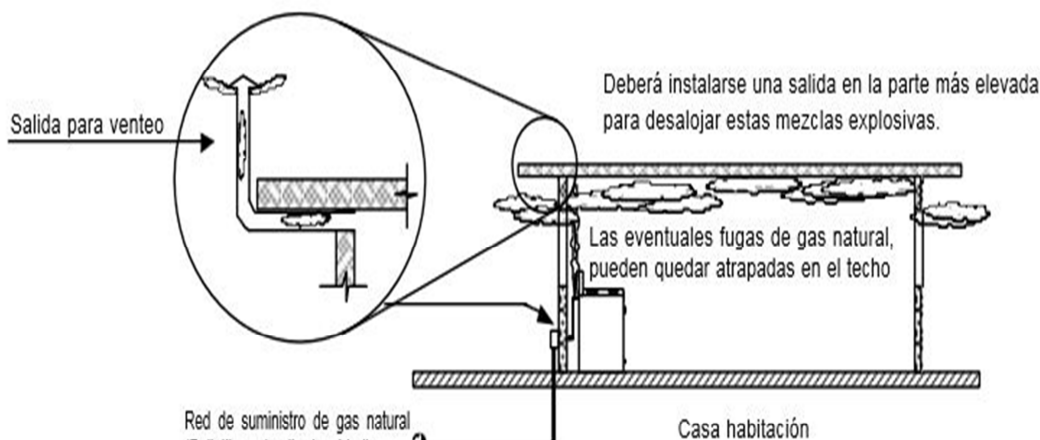


Fig. 1. Almacenamiento de gases.

Las principales topologías de redes de dispositivos ZigBee que se pueden configurar son la topología de estrella, la topología de cluster de árbol y la topología de malla. En todas ellas existe un dispositivo que realiza la función de coordinador. Las principales ventajas de la tecnología ZigBee son las siguientes: arquitectura de red flexible, dispositivos de bajo costo y bajo consumo de energía, la cantidad de nodos que pueden formar una red es muy grande ($\leq 65,536$). Los nodos ZigBee pueden trabajar por periodos muy grandes de tiempo, incluso años, sin tener que cambiar la batería y no requieren licencias para el uso de frecuencias, ya que trabajan en bandas de radio centradas en los 868, 915 y 2400 MHz. disponibles en casi cualquier región del planeta,

siendo la más usada la de 2400 MHz. por ser en la que se transmite a mayor velocidad (250 Kbps.). Las principales desventajas que tiene ZigBee son las siguientes: velocidad de transmisión baja, las distancias alcanzadas son cortas y la existencia de un punto de falla representado por el coordinador de la red ZigBee, razones por las cuales en ciertas aplicaciones se usa otro tipo de conexión alámbrica o inalámbrica [6].

Los transceptores o nodos ZigBee pueden recibir comandos desde un controlador, lo que permite realización de aplicaciones de monitoreo y redes de gran tamaño ZigBee para la comunicación de sensores y actuadores remotos con un sistema de control central donde un operador pueda visualizar la información transmitida. Debido a que ZigBee usa un protocolo seguro de enrutamiento de paquetes, se pueden implantar técnicas de enrutamiento de tráfico para mejorar el rendimiento de las redes ZigBee, así como usar algoritmos de encriptación y autenticación.

En el desarrollo de este trabajo se monitorea el nivel de gas en cinco departamentos de un edificio. En estos departamentos se ubican los sensores en un área donde se tiene mayor probabilidad a una fuga de gas, como se muestra en la Fig. 2. La información leída periódicamente por los sensores de gas es enviada a un controlador central donde puede ser desplegada, almacenada y procesada. Esta información es comparada periódicamente contra el valor de un umbral pre-establecido para alertar al usuario sobre algún problema que podría ser provocado por una fuga de gas.

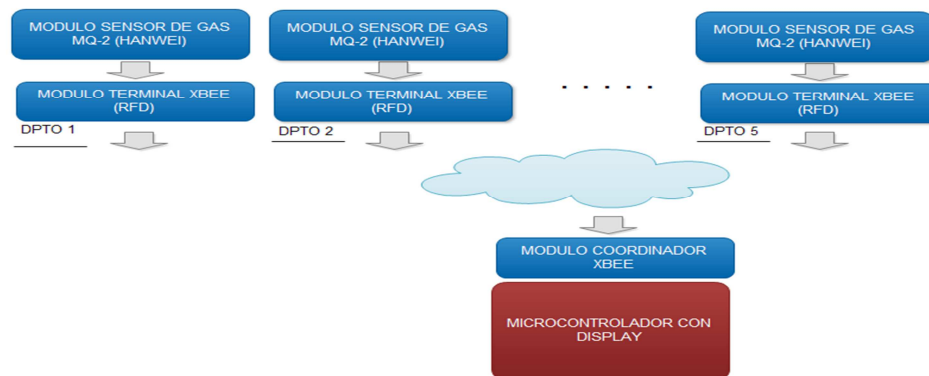


Fig. 2. Diagrama de bloques del sistema completo.

2. Desarrollo

Para el desarrollo del sistema se siguió una metodología que consistió en dividir el trabajo en tres módulos funcionales: un sensor remoto de gas, un controlador central y una interface de usuario. La Fig. 3 muestra el diagrama de bloques del sistema completo. A continuación se explicará el funcionamiento de cada módulo y posteriormente la programación del microcontrolador.

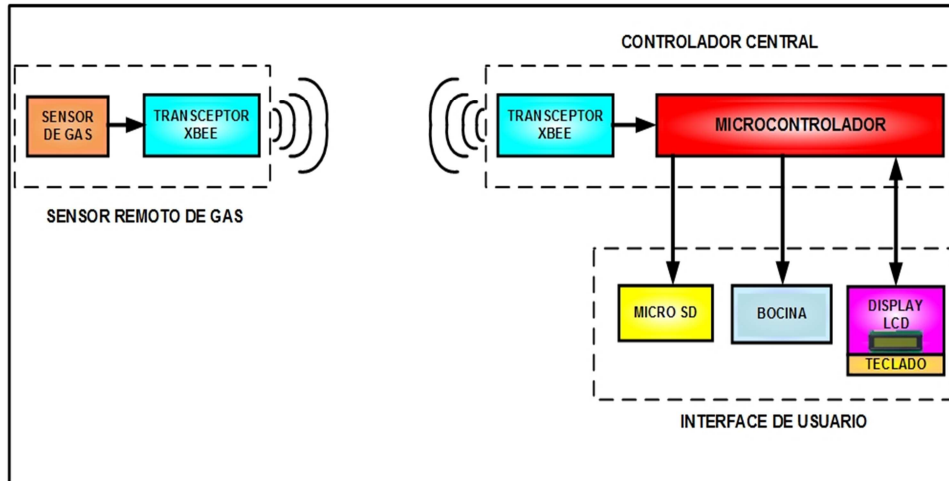


Fig. 3. Diagrama de bloques del sistema completo.

El sensor remoto de gas

Este módulo del sistema está compuesto por un sensor de gas y un transceptor ZigBee.

El sensor de gas usado es un MQ-2 de Hanwei Electronics Co., el cual ofrece un amplio rango de detección, un circuito de larga vida y estable y respuesta rápida. Es usado en aplicaciones industriales para detección de gas propano, butano, metano, alcohol, hidrogeno y humo [7]. En la Fig. 4 se muestra una vista frontal del sensor MQ-2.

El material con el que está construido el sensor es SnO₂ (óxido de estaño), el cual presenta baja conductividad en ambientes de aire limpio. Al encontrar gas combustible en el ambiente, la conductividad del sensor aumenta conforme aumenta la

concentración de gas, por lo que a su salida debe usarse una resistencia para convertir el cambio de conductividad en una señal analógica de salida de 0 a 5 V DC, que corresponde a la concentración de gas.



Fig. 4. El sensor MQ-2.

El MQ-2 puede detectar concentraciones humo y gas combustible de 300 a 10000 ppm (partes por millón), tiene una alta sensibilidad y un tiempo de respuesta rápido. Se alimenta con 5V DC, trabaja en el rango de temperatura de -10 a 50 °C y su consumo de potencia es menos de 750 mW.

En este módulo del sistema se conectó la salida analógica del sensor a la entrada analógica AD0 del transceptor ZigBee. En la Fig. 5 se muestra una vista trasera del sensor MQ-2 donde se pueden las terminales del mismo.



Fig. 5. Distribución de terminales del sensor MQ-2.

El transceptor ZigBee usado en este módulo fue un circuito XBee Serie 2. Los circuitos XBee Serie 2 se pueden configurar en uno de dos modos de operación según su

función en la red: modo coordinador o modo ruteador y en uno de dos modos para transmitir información: modo transparente o AT y modo API. Una red ZigBee debe contar al menos con un nodo coordinador y un nodo ruteador, Este circuito trabaja en la banda de 2.4 GHz con protocolo de comunicación 802.15.4 y puede ajustarse para usarse en redes de configuración punto a punto y punto a multipunto [8]. Son utilizados en automatización de hogares, sistemas de seguridad, monitoreo de sistemas remotos y aparatos domésticos. En este trabajo, el circuito XBee se configuró en modo transparente y como coordinador y su función es convertir la señal analógica recibida en su entrada AD0, a un valor digital y transmitirla inalámbricamente al controlador central del sistema. En la Fig. 6 se muestra el transceptor XBee Serie2 usado en este módulo.



Fig. 6. Circuito XBee Serie 2.

La información enviada al controlador central se empaqueta en una trama RF que contiene la dirección origen y la dirección destino de los transceptores XBee. Las tramas RF siguen la estructura de la especificación 802.15.4. La configuración de los circuitos XBee se puede realizar de manera fácil y sencilla desde una computadora personal como lo muestra la imagen de la Fig. 7.

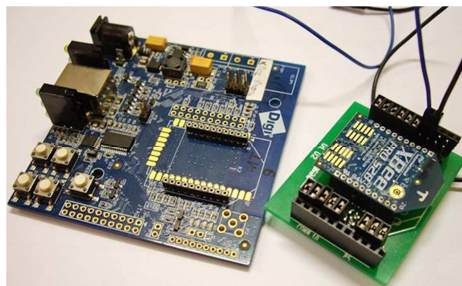


Fig. 7. Configuración de los circuitos XBee.

El controlador central

Este módulo está compuesto por un transceptor ZigBee y un microcontrolador.

El transceptor ZigBee usado en este módulo fue un circuito XBee Serie 2 configurado en modo transparente y como ruteador. Su función es convertir la señal de RF recibida desde el sensor remoto de gas a una señal serie digital y entregarla al microcontrolador. La salida DOUT del Puerto Serie del Circuito XBee se conectó a la entrada RXD del Puerto Serie del microcontrolador.

El microcontrolador usado fue un PIC 16F887, el cual cuenta con los siguientes recursos principales: CPU de 8 bits, memoria de programa flash de 8 KBytes, 4 puertos paralelo, un puerto serie EUSART, un módulo PWM de 10 bits y tres temporizadores [9]. Estos recursos fueron suficientes para la programación realizada y conectar al microcontrolador el transceptor ZigBee y la interface de usuario. Su costo es relativamente bajo comparado con otros microcontroladores.

La programación del microcontrolador se realizó de manera tal que continuamente se encuentra realizando dos tareas principales: lee periódicamente la información de su Puerto Serie y atiende la interface de usuario.

Al leer el valor de concentración de gas enviado por el sensor, lo compara con un umbral establecido por el usuario, Al leer el valor de concentración de gas enviado por el sensor, lo compara con un umbral establecido por el usuario, generando una alerta audible y almacenando en una memoria del tipo MicroSD el valor de concentración,

fecha y hora. El umbral recibido desde la interface de usuario es almacenado en la memoria EEPROM del microcontrolador.

La programación del microcontrolador también atiende a la interface de usuario usando un menú de opciones explicadas a continuación

La interface de usuario

Este módulo está compuesto por un teclado, un display LCD de 4x16, una bocina y un circuito de lectura/escritura a una memoria Micro SD.

En el display LCD se le muestra al usuario un menú de opciones mediante el cual puede realizar dos tareas básicas: establecer el nivel de umbral de la concentración de gas y revisar las alertas registradas históricamente en memoria Micro SD.

Finalmente, el circuito de lectura/escritura a una memoria Micro SD utilizado fue el módulo microSD Card, como el que muestra en la Fig. 8, el cual se conectó al puerto SPI (Serial Peripheral Interface) del microcontrolador.

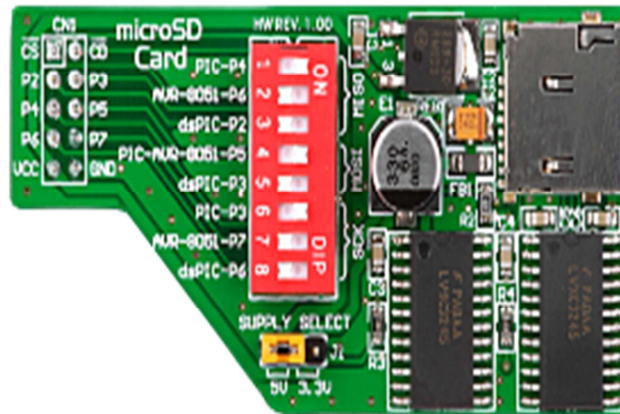


Fig. 8. Módulo microSD Card.

3. Resultados

En la implantación de este trabajo se realizaron cinco módulos sensores remotos de gas. Se llevaron a cabo un conjunto de pruebas de funcionalidad las cuales consistieron en lo siguiente: se localizaron los módulos sensores en diferentes posiciones de cada departamento del edificio para variar la distancia al controlador central; se establecieron diferentes umbrales de concentración permisible de gas y se configuraron los circuitos XBee Serie para que usaran diferentes velocidades de transmisión serie. Los resultados obtenidos fueron los siguientes: la distancia del controlador central al módulo sensor más lejano fue de 65 metros sin presentarse problemas de comunicación, considerando como obstáculos entre ellos las paredes del edificio. Esta distancia fue un poco menor a los 70 metros especificados por el fabricante del circuito XBee Serie 2. Para probar que se genera una alarma al rebasar los diferentes umbrales de concentración de gas, se abrió la llave de paso del gas para permitir que este sea detectado por los sensores, generándose oportunamente la alarma, este tipo de pruebas fue de riesgo y peligrosa y tuvo que hacerse rápidamente para evitar un accidente. Las velocidades configuradas en las pruebas de transmisión inalámbrica de información fueron de 9600 bps y 250 Kpbs., ésta última es la máxima especificada por el fabricante.

4. Discusión

Para un óptimo funcionamiento del sistema es necesario realizar constantes mantenimientos y pruebas de comunicación entre los miembros del sistema para detectar fallas y corregirlos a tiempo. Una posible extensión de este trabajo es conectar la salida serie del circuito XBee del controlador central a un gateway XBee, esto permitiría enviar la información enviada por el sensor de gas a una red Ethernet conectada a la Internet y desde una computadora personal ubicada remotamente monitorear el nivel de gas medido por el sensor, esto necesitaría realizar una interface de usuario en la computadora para recibir y mostrar la información transmitida por el circuito XBee.

5. Conclusiones

En este trabajo se alcanzó el objetivo planteado, el cual fue crear un dispositivo para monitoreo de gas en el ambiente causado por una fuga o descuido brindando seguridad en el lugar donde se instale el sistema, contando con un sistema de alerta efectivo bajo el estándar IEEE 802.15.4, comercialmente conocido como ZigBee, proporcionando una solución de bajo costo, compacta, de baja complejidad y mínimo consumo de energía para redes con bajas tasas de transmisión en comunicaciones inalámbricas. ZigBee opera en la banda de frecuencia ISM de 2.4 GHz, misma que se encuentra saturada por la utilización de otros dispositivos inalámbricos; sin embargo, las aplicaciones desarrolladas bajo esta tecnología no requieren una comunicación constante, por lo cual los 16 canales con los que cuenta son suficientes para comunicarse apropiadamente.

6. Referencias

- [1] D. Ching-Liang, L. Mao-Chen, "Nanoparticle "SnO₂ Gas Sensor with Circuit and Micro Heater on Chip Fabricated Using CMOS-MEMS Technique". NEMS '07. 2nd". IEEE International Conference on Nano/Micro Engineered and Molecular Systems. Conference Location: Bangkok. No. DOI: 10.1109/NEMS.2007.352177. 16-19 Jan 2007. 959-963 pp.

- [2] V. Jelcic, M. Magno, G. Paci, D. Brunelli, L. Benini, "Design, characterization and management of a wireless sensor network for smart gas monitoring". 4th. IEEE International Workshop on Advances in Sensors and Interfaces. Conference Location: Savelletri di Fasano. No. DOI: 10.1109/IWASI.2011.6004699. 28-29 June 2011. 115-120 pp.

- [3] L. Haitao, M. Xiaoyi, Y. Yuning, A. J. Mason, "Low Power Multimode Electrochemical Gas Sensor Array System for Wearable Health and Safety Monitoring Sensors". IEEE Journal. Vol. 14. No. 10. 2014. 3391-3399 pp.

- [4] M. Shan, X. Li, Ch. Zhu, J. Zhang, "Gas concentration detection using ultrasonic based on wireless sensor networks". 2nd. International Conference on Information Science and Engineering (ICISE). Conference Location: Hangzhou, China. No. DOI: 10.1109/ICISE.2010.5691425. 4-6 Dec. 2010. 2101-2106 pp.
- [5] C. Kyung, Y. Minjung, "An enhanced key management using ZigBee Pro for wireless sensor networks". International Conference on Information Networking (ICOIN). 2012. 399-403 pp.
- [6] D. A. Vişan, I. Liţă, "Wireless measurement system based on ZigBee transmission technology". 33rd International Spring Seminar on Electronics Technology (ISSE). 2010. 464-467 pp.
- [7] MQ-2 Gas Sensor Data Sheet. Hanwei Electronics Co., Ltd. <http://www.hwsensor.com>. 2015.
- [8] ZigBee RF Modules Documentation. Digi International Inc. <http://www.digi.com>. 2015.
- [9] PIC16F887 Data Sheet. Microchip Technology Inc. <http://www.microchip.com>. 2007.

7. Autores

M. en C. José Ignacio Vega Luna obtuvo su título de Maestría en Ciencias de la Computación en la UAM-Azcapotzalco. Sus áreas de especialización y trabajo son: aplicaciones de microcontroladores, UNIX y sistemas de alta disponibilidad. Desde 1985 es Profesor Titular de carrera en la UAM-Azcapotzalco.

Ing. Mario Alberto Lagos Acosta es Ingeniero en Electrónica por la UAM-Azcapotzalco. Sus áreas de especialización y trabajo son: aplicaciones con microprocesadores y microcontroladores. Actualmente es Profesor Asociado en la UAM-Azcapotzalco.

Ing. Gerardo Salgado es Ingeniero en Electrónica por la UAM-Azcapotzalco. Sus áreas de especialización y trabajo son: aplicaciones con microprocesadores y microcontroladores. Actualmente es Profesor Titular en la UAM-Azcapotzalco.

Ing. Víctor Noé Tapia Vargas es Ingeniero en Electrónica por la UAM-Azacapotzalco, cuenta con el 100% de créditos de la Maestría en Ciencias de la Computación en la UAM-Azacapotzalco. Sus áreas de especialización y trabajo son: aplicaciones con microprocesadores, microcontroladores y robótica. Actualmente es Profesor Titular en la UAM-Azacapotzalco.

Desarrollo de un sistema embebido móvil de bajo costo utilizando la tarjeta beaglebone black y programas de código abierto

Edgar Alejandro Rivas Araiza

Universidad Autónoma de Querétaro, Facultad de Ingeniería, Teléfono: 442-274-4637
erivas@uaq.mx

Alexander Rodríguez Rosales

Universidad Autónoma de Querétaro, Facultad de Ingeniería, Teléfono: 442-271-2588
arodriguez113@alumnos.uaq.mx

Estefanía Desiree Avalos Rivera

Universidad Autónoma de Querétaro, Facultad de Ingeniería, Teléfono: 993-300-8885
estefania.desiree.avalos.rivera@gmail.com

Resumen

El presente trabajo describe la implementación del sistema operativo libre Android, sobre una tarjeta embebida de bajo costo llamada Beaglebone Black. El objetivo es desarrollar un sistema embebido, que funcione de plataforma para probar aplicaciones móviles, interfaces gráficas, controladores de dispositivos, entre otros proyectos que se están llevando a cabo en la facultad de ingeniería. Se presenta las funcionalidades de la capa de interface de radio (*RIL*), y cómo se puede configurar en el sistema operativo. Se explica también como manejar el modem GSM Enabler III de Enfora, a través del *RIL* y la interface serial. Finalmente se presentan los resultados obtenidos, que fueron la posibilidad de hacer llamadas y enviar mensajes de texto a través del sistema, haciendo uso de los programas desarrollados en la facultad para sistemas Android.

Palabras Claves: Android, beaglebone, código abierto, ril, sistemas embebidos.

1. Introducción

En México, de acuerdo a un estudio sobre los hábitos de los usuarios de internet realizado por la AMIPCI (Asociación Mexicana de Internet) en el año 2014, se reporta que 5 de cada 10 internautas se conectan a través de su teléfono inteligente (*smartphone*), siendo el dispositivo de conexión a Internet más usado después de la laptop y la PC [1]. Debido a esta nueva tendencia tecnológica de tener acceso a Internet desde cualquier lugar, equipos como las tabletas, teléfonos inteligentes, equipos de navegación global, consolas de videojuegos portátiles, han causado un gran impacto en el mercado de consumo de equipos electrónicos [2]. Tecnologías como el servicio de datos celular 3.5G y 4G, Wifi, WiMAX, Bluetooth, NFC, y GPS son soportados por casi cualquier equipo móvil ya sea para tener acceso a la Internet, compartir ficheros, fotos, música, o incluso tener acceso a servicios empresariales que permiten a los trabajadores tener acceso a Intranets desde donde quiera que se este se encuentre [3]. La demanda de este tipo de productos ha permitido el desarrollo comercial de los mismos, así como también la investigación de nuevas tecnologías que permitan brindar novedosas soluciones, así como funcionalidades cada vez más integrales, como por ejemplo la interacción con el Internet de las cosas [4].

Debido a este crecimiento ágil de la tecnología móvil, ha crecido el interés en impulsar la línea de investigación en sistemas embebidos en la UAQ. Actualmente se trabaja en varios proyectos, entre los cuales está el desarrollo de un dispositivo móvil con fines académicos. La finalidad del mismo es crear una plataforma, sobre la cual no solo se pueda trabajar aplicaciones y funcionalidades típicas de un equipo móvil como un teléfono inteligente, o tableta, sino que también permita el aprender más sobre los sistemas embebidos, y así aplicarlos en otras áreas como las redes de sensores inalámbricos [5], edificios inteligentes [6], internet de las cosas, telemetría [7], gestión de la energía [8], por mencionar algunas.

Un aspecto importante del proyecto es el interés en utilizar programas de código abierto [9] también conocidos como *freeware* [10], u *opensource* [11]. La capacidad y funcionalidad de los mismos ha sido ampliamente demostrada [12], por lo que el interés tanto de parte de usuarios finales, como empresas [13], oficinas gubernamentales [14], y centros de investigación [15] ha aumentado significativamente en los últimos años [16]. El sistema operativo seleccionado como base del proyecto es el sistema Android 4.2 (*Jelly Bean*), por varias razones, como el amplio soporte para plataformas embebidas [17], variedad de aplicaciones disponibles libremente [18], así como también la disponibilidad de herramientas para programación de forma gratuita.

En la parte de sistemas embebidos se optó por la tarjeta de desarrollo beaglebone black, la cual es conocida por su relación de precio versus funcionalidades. Las características del sistema se explicarán de forma más amplia en una sección posterior. Como referencia se aclara que este proyecto fue comenzado sobre otra plataforma conocida como AM335x Starter Kit [19] de Texas Instrument, la cual posee más periféricos que la beaglebone black, sin embargo tiene menos potencia de procesamiento, es mucho más costosa, y su diseño electrónico es complejo. Además de esto, tiene ciertas restricciones para trabajar con tarjetas externas, debido a que fue diseñada para trabajar principalmente con los periféricos que trae de fábrica. Por estas razones se decidió cambiar la plataforma por una con arquitectura abierta, bajo costo, con una comunidad de usuarios numerosa y creciente, y que tuviera un sistema electrónico más simple, por lo que en base a estos requerimientos llegamos a la plataforma actual.

El estado del arte en sistemas embebidos es diverso, debido a su portabilidad, bajo consumo, y flexibilidad en su diseño [21]. Estas características en conjunto con restricciones como la cantidad de memoria del sistema, baja capacidad de procesamiento, y sistema de alimentación por pilas, permiten el desarrollo de diversas investigaciones y desarrollos tecnológicos dirigidos a optimizar los recursos de estos dispositivos. Líneas de investigación como criptografía [20], implementan algoritmos

para procesamiento en paralelo haciendo uso de la computación en la nube, para mejorar la seguridad de los datos en los dispositivos móviles. Otro caso de estudio interesante es el de realizar una descripción del terreno utilizando algoritmos implementados en sistemas móviles [22]. Como es notable, la oportunidad de desarrollo de nuevas aplicaciones para sistemas embebidos móviles es muy amplia, por eso uno de los objetivos es implementar una plataforma sobre la cual se puedan realizar desarrollo de programas, de modo que los estudiantes puedan trabajar en proyectos que tengan el potencial de convertirse en productos finales, gracias a todas las ventajas de la ideología del conocimiento libre.

1.1. Vistazo general al sistema Android

En este proyecto se implementó el sistema operativo Android, basado en Linux y desarrollado por Google. Este ha logrado convertirse desde su lanzamiento beta en 2007 [23], en una de las principales plataformas móviles en el mercado y es cada vez más popular en los teléfonos inteligentes y tablets [24]. En términos prácticos, Android es una aplicación desarrollada sobre el núcleo (*kernel*) Linux, lo que facilita su rápido despliegue en diferentes plataformas. Además de los dispositivos móviles, Linux domina el mercado de los sistemas embebidos, sin embargo la interfaz de usuario para los desarrolladores antes de la llegada de Android era casi inexistente, y con las que se contaban eran limitadas. Esto es más que una funcionalidad técnica, ya que los desarrolladores que crean dispositivos orientados a encarar al usuario, también deben lidiar con factores de interacción humano-computadora. Aquí es donde Android, que provee un sistema operativo robusto y rápido, ayuda a los diseñadores que deben presentar al usuario una experiencia eficiente y fácil de usar, con funcionalidades como la interfaz touchscreen que brinda un fácil acceso a las aplicaciones multimedia mientras usa un mínimo de recursos [25].

La clave del éxito de Android es que su código fuente es abierto y es gratis para su uso en cualquier proyecto ya sea comercial o no. Para que los desarrolladores implementen

sus aplicaciones embebidas, Android provee un paquete completo de componentes de programación en la parte superior del núcleo Linux. Este último provee las funciones básicas de gestión del sistema para controlar procesos, periféricos, memoria y seguridad. El núcleo maneja la creación de redes y los controladores de dispositivos, lo que facilita la interfaz con el periférico. El entorno de desarrollo, que normalmente trabaja en GNU Linux, permite ejecutar código desde el dispositivo, el cual puede estar conectado vía USB [26]. Vale aclarar que al hablar de Linux [27], se trata del núcleo básico para sistemas operativos como lo es en el caso de Android. GNU Linux por otra parte, es un sistema operativo de código abierto, el cual al igual que Android, usa a Linux como núcleo base.

En los últimos años diferentes versiones de Android OS han sido liberadas. Al inicio del proyecto de la plataforma móvil, se trabajó con la versión Ice Cream Sandwich 4.0.3, sin embargo se avanzó hasta la distribución Android JellyBean 4.2.2, o *API* nivel 17. Los niveles *API* (Interfaz de programación de aplicaciones) describen como es la comunicación entre las funciones del dispositivo y su funcionalidad, y se incrementan conforme a su funcionalidad. En los niveles del 3 al 10, Android fue hecho específicamente para teléfonos inteligentes, y del 11 al 13 fueron hechos solo para tablets. A partir del nivel 14, que se inicia con Ice Cream, se unieron las funcionalidades para ambos teléfonos y tablets [28].

Existen varios proyectos de portar el soporte de Android a las diferentes plataformas embebidas. El que se seleccionó para el presente proyecto es el Rowboat de Texas Instrument [29], primeramente por tener el código fuente completo, y por soportar la compatibilidad con el procesador y periféricos de la tarjeta beaglebone black. Además estas fuentes han sido utilizadas desde el inicio del proyecto, por lo que se tiene documentación interna relevante para procesos como la compilación, edición del código, configuración del entorno de programación, y otros detalles.

1.2. Características de la BeagleBone Black

La tarjeta BeagleBone Black (BBB) es una plataforma embebida de bajo costo (al momento de escribir el documento tiene un costo de USD 45.00 en adelante), y está diseñada para probar aplicaciones propias de manera rápida y fácil directamente en el dispositivo. Es producida por *CircuitCo* para la organización sin fines de lucro BeagleBoard.org, y fue producida por primera vez en 2013. Es una computadora con un solo procesador de la línea ARM Cortex A8 @ 1GHz. La BBB es compacta, posee soporte para diversas interfaces de comunicación como I2C, SPI, UART, HDMI, USB, Ethernet, JTAG, posee, 512MB de memoria RAM, tiene soporte para almacenamiento de memoria externo por microSD, además de un acelerador gráfico 3D. La descripción completa de las características de la BBB se encuentra en la tabla 1.

	Feature
Processor	Sitara AM3358BZCZ100 1GHz, 2000 MIPS
Graphics Engine	SGX530 3D, 20M Polygons/S
SDRAM Memory	512MB DDR3L 800MHZ
Onboard Flash	4GB, 8bit Embedded MMC
PMIC	TPS65217C PMIC regulator and one additional LDO.
Debug Support	Optional Onboard 20-pin CTI JTAG, Serial Header
Power Source	miniUSB USB or DC Jack 5VDC External Via Expansion Header
PCB	3.4" x 2.1" 6 layers
Indicators	1-Power, 2-Ethernet, 4-User Controllable LEDs
HS USB 2.0 Client Port	Access to USB0, Client mode via miniUSB
HS USB 2.0 Host Port	Access to USB1, Type A Socket, 500mA LS/FS/HS
Serial Port	UART0 access via 6 pin 3.3V TTL Header. Header is populated
Ethernet	10/100, RJ45
SD/MMC Connector	microSD , 3.3V
User Input	Reset Button Boot Button Power Button
Video Out	16b HDMI, 1280x1024 (MAX) 1024x768,1280x720,1440x900 ,1920x1080@24Hz w/EDID Support
Audio	Via HDMI Interface, Stereo
Expansion Connectors	Power 5V, 3.3V , VDD_ADC(1.8V) 3.3V I/O on all signals McASP0, SPI1, I2C, GPIO(69 max), LCD, GPMC, MMC1, MMC2, 7 AIN(1.8V MAX), 4 Timers, 4 Serial Ports, CAN0, EHRPWM(0,2),XDMA Interrupt, Power button, Expansion Board ID (Up to 4 can be stacked)
Weight	1.4 oz (39.68 grams)

Tabla 1. Especificaciones de la BBB [30].

La BBB tiene soporte para varios sistemas operativos, entre ellos Android, Linux (Debian, Ubuntu, Arch), FreeBSD, entre otros [30]. Una de las características claves de esta tarjeta es que ha sido diseñada con la filosofía de electrónica abierta o libre (*open hardware*) que es la contraparte de código abierto pero implementado en diseño electrónico. Esto quiere decir que desde la lista de partes, el diseño esquemático completo, ficheros para maquinar el circuito, y demás detalles están disponibles sin restricción en Internet, desde la página de beagleboard.org [31]. Esto ha dado pie a que diversas empresas diseñen tarjetas externas compatibles con la tarjeta, llamadas *cape* las cuales se conectan directamente a la tarjeta por medio de conectores diseñados para tal fin (vease fig. 1). Entre esos *cape* se encuentra también pantallas LCD, que se pueden conectar directamente, para formar un dispositivo móvil de forma fácil y segura. Una de las pantallas soportadas es la CS-BBB-EXP50C, de la cual se hablará en la siguiente sección.

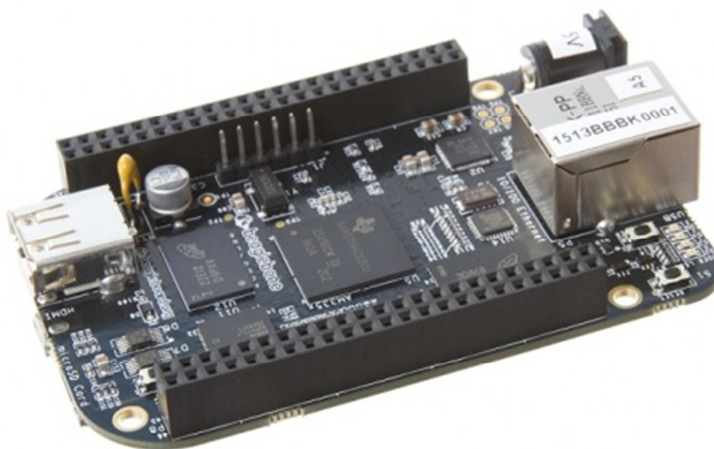


Fig. 1. Tarjeta BeagleBone Black [30].

1.3. Pantalla capacitiva CS-BBB-EXP50C

La pantalla seleccionada para el proyecto es la CS-BBB-EXP50C fabricada por Chipsee. Esta tiene un tamaño de 5", una resolución de 800x480 pixeles, soporte multitáctil para 5 puntos, trae 5 botones de pulsación de uso general, y trae una interface RS232 (véase fig. 2). Tiene soporte para sistemas Android, GNU Linux, y

Windows CE [32]. La pantalla capacitiva tiene un rango de sensibilidad mayor a la resistiva, por lo que permite una mejor experiencia para el usuario al momento de utilizarla. El proceso de compilación de los controladores se describe en una sección posterior.

1.4. Capa de Interface de Radio (RIL)

Dentro de la arquitectura de Android (véase fig. 3) [33], se cuenta con aplicaciones para realizar llamadas, mandar SMS, manejo de Wifi, y demás funciones básicas con las que cuenta cualquier teléfono.



Fig. 2. Pantalla LCD capacitiva multitáctil.

El compendio de servicios de telefonía, es inicializado y puesto en marcha durante el inicio del sistema y todas las peticiones realizadas desde las aplicaciones a través del *API* serán redirigidas al *RIL* por dichos servicios. El *RIL* (Radio Interface Layer) es el puente entre los servicios de telefonía de Android y los dispositivos e incluye soporte para radios basados en *GSM* (sistema global para la comunicación móvil). El *RIL* consiste en 2 componentes primarios: el *RIL daemon*, y el *vendor RIL*.

RIL Daemon (rild): Arranca durante el inicio de sistema de Android y viene incluido en el sistema en "hardware/ril/" como "rild", y trae una referencia para implementación

"reference-ril". Procesa toda comunicación desde los servicios de telefonía Android y despacha llamadas al *vendor* como comandos solicitados. RILD se encarga de encontrar la librería adecuada para inicializar el *vendor RIL* y desplegar todas las funciones de *Vendor* a la capa superior.

Vendor RIL (ril.h): es una librería específica para cada modem, la cual procesa toda la comunicación con el dispositivo de radio y despacha llamadas al *rild* mediante comandos no solicitados.

La librería *ril.h*, define los estados y variables del *RIL* mediante funciones. Para los comandos solicitados del *RIL*, el *vendor* debe proveer las funciones para manejarlos, se encuentran definidos con el prefijo *RIL_REQUEST*.

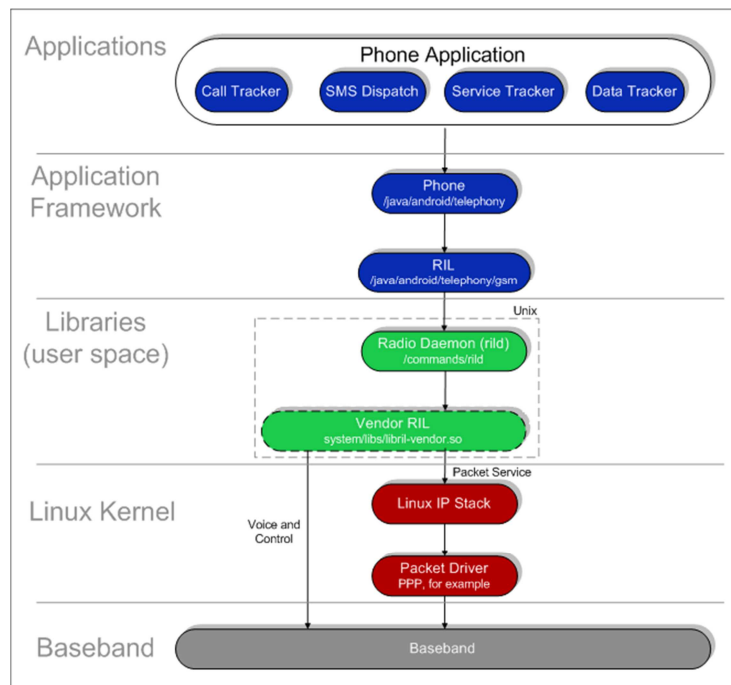


Fig. 3. Arquitectura de comunicación de Android e interacción con el RIL.

El *vendor RIL* usa funciones de respuesta para invocar comandos no solicitados. El *RIL* se maneja por comandos *AT*, que son instrucciones codificadas enviados desde el equipo que conforman un lenguaje de comunicación entre el hombre y el modem.

El interés de trabajar con el RIL en el presente proyecto es el de implementar una comunicación por GSM para realizar llamadas y envío de mensajes de texto. Para tal fin se empleó el módulo ENFORA GSM0308-11 Enabler III, el cual soporta también comunicación de datos por GPRS, sin embargo esta última funcionalidad estará implementada en la siguiente fase del proyecto. [24]

2. Desarrollo

A continuación se presenta la información sobre el desarrollo del sistema propuesto.

2.1. Entorno de compilación

Para llevar a cabo las tareas de compilación, se procede a configurar el entorno tal como se describe en [34]. En este proyecto se trabajó en un servidor de compilación con el sistema operativo CentOS 7, y una máquina virtual con Ubuntu 12.04 de 64 bit. Para la virtualización se utilizó la herramienta *docker* [35], la cual permite tener máquinas virtuales mínimas llamadas contenedores (*containers*), quienes utilizan parte del núcleo del sistema operativo anfitrión para trabajar de forma similar a una aplicación nativa. El *docker* posee muchas ventajas por sobre otros virtualizadores, como por ejemplo el uso de *dockerfiles*, que permite compartir la configuración de una máquina virtual, para que sea clonada en cualquier otro equipo, bajando la información desde el Internet. Un ejemplo de *dockerfile* se muestra a continuación, donde se puede apreciar la configuración del entorno de compilación de forma automática, la creación de usuarios, contraseñas, selección de la distribución y versión del sistema operativo, entre otros detalles:

```
FROM ubuntu:precise-20150320
MAINTAINER Alexander Rodriguez R <arodriguez113@alumnos.uaq.mx>
# Setup for Java
RUN echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main" \
    >> /etc/apt/sources.list.d/webupd8.list && \
    apt-key adv --keyserver keyserver.ubuntu.com --recv-keys EEA14886 && \
    echo oracle-java6-installer shared/accepted-oracle-license-v1-1 select true | debconf-set-
selections

# /bin/sh points to Dash by default, reconfigure to use bash until Android
# build becomes POSIX compliant
RUN echo "dash dash/sh boolean false" | debconf-set-selections && \
```

```
dpkg-reconfigure -p critical dash

# Keep the dependency list as short as reasonable
RUN apt-get update && \
  apt-get install -y bc bison bsdmainutils build-essential curl \
    flex g++-multilib gcc-multilib git gnupg gperf lib32ncurses5-dev \
    lib32readline-gplv2-dev lib32z1-dev libesd0-dev libncurses5-dev \
    libsdl1.2-dev libwxgtk2.8-dev libxml2-utils lzop \
    oracle-java6-installer oracle-java6-set-default \
    pngcrush schedtool xsltproc zip zlib1g-dev \
    uboot-mkimage tofrodos sudo vim && \
  apt-get clean && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*

ADD https://commondatastorage.googleapis.com/git-repo-downloads/repo /usr/local/bin/
RUN chmod 755 /usr/local/bin/*
# RUN cpan App::cpanminus && cpanm Switch

# The persistent data will be in these two directories, everything else is
# considered to be ephemeral
VOLUME ["/tmp/ccache", "/home/usuario"]

#
# user account
#
ENV MYNAME usuario
ENV MYPASS contraseña
ENV HOME /home/$MYNAME

RUN echo "User: $MYNAME Pass: $MYPASS"
RUN useradd --create-home -d $HOME --shell /bin/bash --user-group --groups adm,sudo $MYNAME
RUN echo "$MYNAME:$MYPASS" | chpasswd

USER $MYNAME
WORKDIR $HOME

# Improve rebuild performance by enabling compiler cache
ENV USE_CCACHE 1
ENV CCACHE_DIR /tmp/ccache
```

Se recomienda trabajar con *docker* si se trabaja en diferentes proyectos al mismo tiempo, con un grupo de desarrolladores, y que necesitan diferentes entornos de compilación. En el caso presente, fue una herramienta muy útil y versátil para poder aislar los entornos de los diferentes sistemas operativos, además de la facilidad con que se puede compartir, o reproducir los entornos de trabajo, logrando una respuesta aceptable por parte del servidor ante la demanda de servicio.

2.2. Habilitación del soporte para la LCD capacitiva

Una vez que se cree el entorno de compilación se puede trabajar con las fuentes del *Rowboat*, y del Android distribuido con la pantalla LCD, para poder habilitarle el soporte. Como ya mencionamos anteriormente, la pantalla necesita de controladores tanto para la pantalla LCD como para la parte táctil. Las versiones de Android en ambos casos era

exactamente la misma, Android JB 4.2.2 DevKit 4.1.1, permitiendo que la habilitación fuera casi transparente entre ambos repositorios.

El primer problema a vencer fue que a pesar de que ambas distribuciones eran iguales, la versión del fabricante del LCD resultó ser una versión minimalista del JB, quiere decir que el repositorio no incluía todo el código fuente. De esta manera se volvió más difícil encontrar cuales podrían ser las diferencias entre ambos, por lo que se tuvo que idear una forma de poder encontrar cuales eran los ficheros que se encontraban en el repositorio del JB mínimo, y que no estaban en el *Rowboat*, además de eso cuales ficheros existían en ambos repositorios pero no eran iguales. En esta fase del proyecto no se trabajó con *git*, que es una herramienta de control de versiones que contiene diferentes herramientas para realizar dicha tarea, debido a la complejidad que añadía el manejo de dicha herramienta pues aunque es útil, no es tan intuitiva para el usuario. De este modo lo que se hizo fue desarrollar una serie de *scripts* en *Bash* utilizando la herramienta *diff* como reemplazo del *git*. El primer *script* permite conocer cuales ficheros están en el JB mínimo y no en el *Rowboat*, además permite saber que ficheros son diferentes entre repositorios. El código del script se muestra a continuación:

```
#!/bin/sh
dir1=$1
dir2=$2
IFS=$'\n'
#for file in $(grep -Ilsr -m 1 '.' "$dir2"); do
# diff -q "${file}/${dir1}/${dir2}" "$file"
#done
for file in $(grep -Ilsr -m 1 '.' "$dir1"); do
diff -q "$file" "${file}/${dir1}/${dir2}"
done
```

Para correr este *script* se puede realizar de la siguiente manera:

```
script.sh jb-minimo/ rowboat/ > ficheros_diff.txt 2> ficheros_ausentes.txt
```

En donde *jb-minimo* y *rowboat* son las carpetas de cada repositorio y *script.sh* es el nombre del *script*. Como resultado se obtendrán dos archivos de texto, *ficheros_diff.txt* que es una lista de ficheros que están en ambos repositorios pero no son iguales, y en *ficheros_ausentes.txt* están los que aparecen en *jb-minimo* pero no en *rowboat*. Este

último arrojó un listado en donde se reconocieron todos los ficheros correspondientes al controlador tanto de la pantalla LCD como del multitáctil, por lo que solo se necesita copiarlos al repositorio del *Rowboat*. Luego de filtrar adecuadamente el listado de ficheros diferentes, se puede pasar a utilizar otro *script* el cual creará un grupo de parches, uno por cada fichero, que luego aplicaremos al repositorio del *Rowboat*. Esto permitirá que los ficheros de compilación incluyan los controladores de la pantalla, al momento de empezar con el proceso. El script se presenta a continuación:

```
#!/bin/sh
LINEAS=`wc -l $1 | cut -d" " -f1`
for i in $( seq 1 $LINEAS );
do
    FICHERO=`awk NR==$i $1`
    diff -u $FICHERO > parche$i.patch
done
```

Una vez ya se tengan los parches listos, se pueden aplicar utilizando el siguiente *script*.

```
#!/bin/sh
LINEAS=`wc -l $1 | cut -d" " -f1`
for i in $( seq 1 $LINEAS );
do
    FICHERO=`awk NR==$i $1|cut -f1`
    patch $FICHERO < parche$i.patch && echo "parche $i listo"
    #echo $FICHERO
done
```

Una vez se apliquen los parches, se procede al proceso de compilación tal como se describe en [34], se crea la imagen en la memoria microSD y se procede a activar los interruptores necesario en el cape del LCD. Luego del arranque de la BBB, esta se conecta a una estación de trabajo mediante la interface USB. En el dispositivo activamos el modo de depuración para USB (*USB Debugging mode*) en ajustes, para que se pueda realizar cualquier tarea de mantenimiento o depuración en el dispositivo.

2.3. Configuración del *RIL*

Para activar la configuración del *RIL*, se utilizó información explicada en [34] y también se añadieron algunos puntos específicos para el modem de Enfora. Este modem tiene una distribución de *vendor RIL* de código abierto [36], sin embargo está diseñada para las primeras versiones de Android, por lo que existen problemas en la implementación

directa en el código fuente. La solución fue adaptar el *reference ril*, que es una distribución de ejemplo que viene con el código fuente del *Rowboat*, para que cumpliera con los requerimientos de comunicación del modem.

Tal como se explicó anteriormente, la función del *RIL* es gestionar el control de la comunicación entre diferentes periféricos y las aplicaciones del Android, en este caso del modem GSM. Esto se logra a través del envío de comandos *AT*, por lo que se debe consultar el manual de comandos *AT* soportados por el modem [37], para poder adaptar el *reference ril* para que lo controle adecuadamente. Esta parte es esencial para el correcto funcionamiento de llamadas y envíos de mensajes, pues aunque la aplicación de llamadas funcione, si no puede comunicarse con el modem adecuadamente a través del *RIL*, estas tareas no se cumplirán. Incluso puede caer en un ciclo infinito en que el sistema intenta establecer la comunicación reiniciando el *rild*, más sin embargo esto no causará más que un alza del consumo energético de la tarjeta por un uso continuo e innecesario del procesador.

Referente a la configuración del *Rowboat*, es necesario habilitar la funcionalidad de telefonía, deshabilitar inicialmente lo relacionado con *APN* si no deseas activar el servicio de GPRS, y aplicar estos cambios en diversos ficheros. Los más importantes se presentan a continuación:

- `build/target/product/full_base.mk`
- `build/target/product/full_base_telephony.mk`
- `build/target/product/generic_no_telephony.mk`
- `build/target/product/telephony.mk`
- `kernel/arch/arm/mach-omap2/board-am335xevm.c`
- `system/core/init/property_service.c`
- `system/core/rootdir/ueventd.rc`
- `device/ti/am335xevm/device.mk`
- `device/ti/am335xevm/am335xevm.mk`
- `device/ti/beagleboneblack/overlay/frameworks/base/core/res/res/values/config.xml`
- `device/ti/beagleboneblack/init.am335xevm.rc`

3. Resultados

Terminada esta fase del proyecto, obtuvimos un sistema embebido móvil completamente funcional, desde donde se podían hacer llamadas, enviar SMS, instalar aplicaciones de terceros, con una pantalla LCD capacitiva multitáctil, con opción para reproducción de video y navegación por Internet. La conexión entre el dispositivo y el modem se logró a través de un convertidor USB-serial, y se utilizó la conexión a auriculares y micrófono del propio módulo del modem para hacer las llamadas, debido a que la BBB no trae internamente salida o entrada de audio. En la fig. 4 se puede observar el sistema interconectado con el modem, y la aplicación de telefonía trabajando.

El sistema se ha logrado desarrollar completamente con programas de código abierto, desde el entorno de compilación, el sistema operativo, incluso la electrónica es abierta, minimizando gastos, y explotando las capacidades que nos permite tener acceso al código fuente de proyectos complejos lo es el sistema operativo Android.

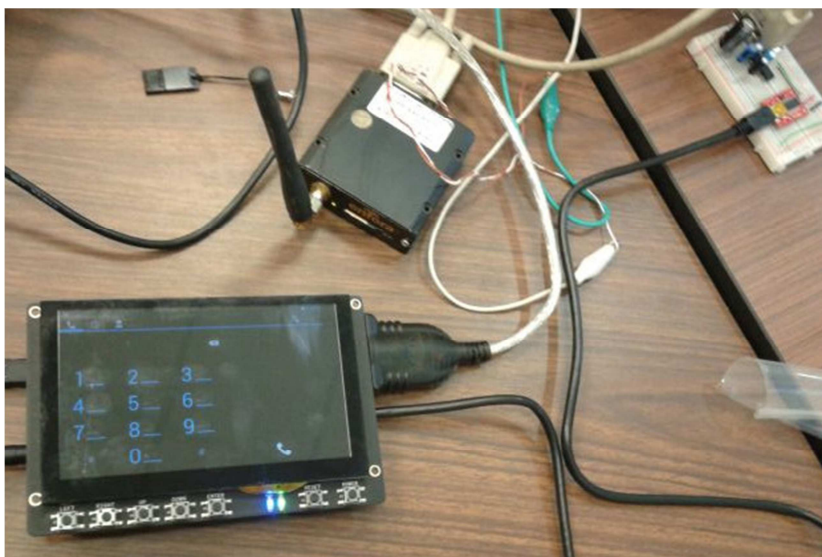


Fig. 4. Muestra del sistema embebido móvil en funcionamiento.

4. Discusión

Entre los puntos importantes que se lograron con el sistema obtenido está el obtener una plataforma más robusta en cuanto a potencia de procesamiento; se mantuvo un consumo bajo de energía dado que la tarjeta tiene un mínimo de periféricos incluidos; se obtuvo un dispositivo móvil que incluye una pantalla LCD de 5" que puede ser utilizada para cualquier utilidad como un teléfono inteligente, un equipo de medición en campo, sistema de posicionamiento, o cualquier otro tipo de uso; además se logró tener acceso a la red celular por medio de un modem GSM.

5. Conclusiones y trabajos futuros

Queda demostrado que se puede desarrollar sistemas embebidos con tecnología actual, haciendo uso de las herramientas disponibles de código y electrónica abierta. No es obligatorio ya tener que pagar por costosas licencias, ni quedarse solo con las funcionalidades que los programas traen por defecto. Hoy día se puede no solo descargar aplicaciones de libre distribución, sino que también se tiene acceso a sus fuentes, para poderlo modificar según sea la necesidad.

Se puede utilizar un modem GSM de bajo costo en conjunto con un sistema embebido para tener acceso a la red celular, ya sea para realizar llamadas, enviar mensajes de texto, o también para comunicarse por la red de datos móvil. La diversidad de interfaces que tiene la BBB permite manejar múltiples periféricos simultáneamente, convirtiéndola en una solución adecuada para diversas aplicaciones.

Como trabajos a futuro está la habilitación del soporte para GPRS a través del modem Enfora, diseñar y construir un *cape* propio para manejar las entradas y salidas de audio y para montar el modem de forma directa a una placa electrónica. También está el implementar *git* para gestionar el control de versiones del proyecto, diseñar y crear una carcasa para el dispositivo móvil, y como un producto derivado está el desarrollar de

cero una plataforma propia utilizando el procesador ARM Cortex A8, para un uso determinado según se requiera.

Finalmente este ejercicio arroja como resultado obtener una plataforma lista para aplicaciones embebidas, utilizando Android como sistema operativo base, lo que abre las puertas a futuras implementaciones mucho más complejas, en donde se podrá implementar soluciones propias haciendo uso de las funcionalidades que este sistema operativo brinda a los desarrolladores, todo a través del código abierto y la liberación del conocimiento.

6. Referencias

- [1] Estudio sobre los hábitos de los usuarios de Internet en México 2014. https://www.amipci.org.mx/estudios/habitos_de_internet/Estudio_Habitos_del_Internauta_Mexicano_2014_V_MD.pdf. Acceso: 6 de junio de 2015.
- [2] Smartphones en México: más marcas, más modelos, más competencia. <http://www.elfinanciero.com.mx/empresas/marcas-y-modelos-de-smartphones-multiplican-su-presencia-en-mexico.html>. Acceso: 5 de junio de 2015.
- [3] J. Grover, "Android forensics: Automated data collection and reporting from a mobile device". *J. Digital Investigation*. Vol. 10. Suplemento. Agosto 2013. S12–S20 pp.
- [4] F. Salim, U. Haque, "Urban computing in the wild: A survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and Internet of Things". *Int. J. Human-Computer Studies*. Vol. 81. Septiembre 2015. 31–48 pp.
- [5] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey". *J. Computer Networks*. Vol. 38. 2002. 393-422 pp.

- [6] P. De Mil, T. Allemeersch, I. Moerman, P. Demeester and W. De Kimpe, "A scalable low-power WSN solution for large-scale building automation". IEEE International Conference on Communications. Mayo 19-23, 2008. 3130-3135 pp.
- [7] G. Bruzzone, M. Caccia, G. Ravera, A. Bertone, "Standard Linux for embedded real-time robotics and manufacturing control systems". J. Robotics and Computer-Integrated Manufacturing. Vol. 25. No. 1. Febrero 2009. 178–190 pp.
- [8] F. Shrouf, G. Miragliotta, "Energy management based on Internet of Things: practices and framework for adoption in production management". J. of Cleaner Production. Vol. 100. 2015. 235–246 pp.
- [9] M. D. Gallego, S. Bueno, F. J. Racero, J. Noyes, "Open source software: The effects of training on acceptance". Computers in Human Behavior. Vol. 49. 2015. 390–399 pp.
- [10] ¿Qué es el software libre? <http://www.gnu.org/philosophy/free-sw.html>. Acceso: 5 de Junio de 2015.
- [11] The open source definition. <http://opensource.org/definition>. Acceso: 5 de junio de 2015.
- [12] M. Sarrab, O. M. Hussain Rehman, "Empirical study of open source software selection for adoption, based on software quality characteristics". Advances in Engineering Software. Vol. 69. Marzo 2014. 1–11 pp.
- [13] Ø. Hauge, C. Ayala, R. Conradi, "Adoption of open source software in software-intensive organizations – A systematic literature review". Information and Software Technology. Vol. 52. 2010. 1133-1154 pp.
- [14] A. van Loon, D. Toshkov, "Adopting open source software in public administration: The importance of boundary spanners and political commitment". Government Information Quarterly. Vol. 32. 2015. 207-215 pp.

- [15] J. S. Horsburgh, S. L. Reeder, A. Spackman Jones, J. Meline, "Open source software for visualization and quality control of continuous hydrologic and water quality sensor data". *Environmental Modelling & Software*. Vol. 70. 2015. 32-44 pp.
- [16] Y. Kuwataa, K. Takedab, H. Miura, "A Study on Maturity Model of Open Source Software Community to Estimate the Quality of Products". *Procedia Computer Science*. Vol. 35. 2014. 1711-1717 pp.
- [17] Android 5.0 compatibility definition. <http://static.googleusercontent.com/media/source.android.com/es//compatibility/android-cdd.pdf>. Acceso: 5 de Junio de 2015.
- [18] Distribution of free and paid Android apps in the Google Play Store from 2009 to 2015. <http://www.statista.com/statistics/266211/distribution-of-free-and-paid-android-apps/>. Acceso: 5 de Junio de 2015.
- [19] AM335x Starter Kit. <http://www.ti.com/tool/tmdssk3358>. Acceso: 5 de Junio de 2015.
- [20] S. Tang, B. Lv, G. Chen, Z. Peng, A. Diene, X. Chen, "Efficient hardware implementation of PMI+ for low-resource devices in mobile cloud computing". *Future Generation Computer Systems, Special Section: Cloud Computing: Security, Privacy and Practice*. Vol. 52. 2015. 116-124 pp.
- [21] L. Oneto, A. Ghio, S. Ridella, D. Anguita, "Learning Resource-Aware Classifiers for Mobile Devices: From Regularization to Energy Efficiency". *Neurocomputing, Industrial Data Processing and Analysis, 11th World Congress on Intelligent Control and Automation*. Vol. 169. 2015. 225-235 pp.
- [22] J. P. Suárez, A. Trujillo, J. M. Santana, M. de la Calle, D. Gómez-Deck, "An efficient terrain Level of Detail implementation for mobile devices and

- performance study". *Computers, Environment and Urban Systems*. Vol. 52. 2015. 21-33 pp.
- [23] A. Henderson, A. Prakash, *Android for the beaglebone black*. 1ra Ed. 2015. Packt Publishing Ltd. Birmingham, UK. 7-19 pp.
- [24] Enabler IIIG Quad-Band OEM Cellular Modem. http://www.nvtl.com/files/1713/6439/6537/Enabler_IIIG_Quad_reduced.pdf. Acceso: 5 de Junio de 2015.
- [25] K. Yagmour, *Embedded Android*. 1ra Ed. 2013. O'Reilly Media, Inc. United States of America. 1-22 pp.
- [26] C. Walls, *Embedded Software*. 2da Ed. 2012. Elsevier. UK. 337–363 pp.
- [27] What is Linux?. <https://www.kernel.org/linux.html>. Acceso: 5 de Junio de 2015.
- [28] Codenames, Tags, and Build Numbers. <http://source.android.com/source/build-numbers.html>. Acceso: 5 de Junio de 2015.
- [29] Enables Android for Texas Instruments devices. <https://code.google.com/p/rowboat/>. Acceso: 5 de Junio de 2015.
- [30] Beagleboard:BeagleBoneBlack. <http://elinux.org/Beagleboard:BeagleBoneBlack> Acceso: 5 de Junio de 2015.
- [31] BeagleBoard Hardware Design. <http://beagleboard.org/hardware/design>. Acceso: 5 de Junio de 2015.
- [32] CS-BBB-EXP50C. <http://www.chipsees.com/cs-bbb-exp50c.html>. Acceso: 5 de Junio de 2015.
- [33] Radio Layer Interface. <http://www.kandroid.org/online-pdk/guide/telephony.html>. Acceso: 5 de Junio de 2015.

- [34] J.M. Ramos-Arreguín, P. Richter, A. Takacs, M. Toledano-Ayala, E.A. Rivas Araiza, J.C. Moya- Morales, J.E. Rivas-Araiza, C.O. Mendoza-Herbert, C. Vargas-Cabrera, "Adaptación del SO Android en un AM335x para usar un módulo GSM". XI Congreso Internacional sobre Innovación y Desarrollo Tecnológico, CIINDET. 25-27 de Marzo 2015.
- [35] What is docker? <https://www.docker.com/whatisdocker/>. Acceso: 5 de Junio de 2015.
- [36] Enfora EDG308 RIL Plugin. <http://omapzoom.org/?p=platform/hardware/enfora/edg308-ril.git;a=summary>. Acceso: 5 de Junio de 2015.
- [37] Enfora Enabler III GSM/GPRS/EDGE Radio Modem AT Command Set Reference. <http://www.farnell.com/datasheets/876256.pdf>. Acceso: 5 de Junio de 2015.

7. Autores

Dr. Edgar Alejandro Rivas Araiza es Ingeniero en Automatización y Control de Procesos, M. en C. en Control Automático y Dr. en Ingeniería por la Universidad Autónoma de Querétaro. Es investigador y docente de la Facultad de Ingeniería desde el 2005. De 2008 al 2011 fue coordinador del Centro de Diseño e Innovación Tecnológica (CEDIT) y actualmente es coordinador de la Maestría en Instrumentación y Control en la Facultad de Ingeniería.

Ing. Alexander Rodríguez Rosales es Ingeniero en Electrónica y Telecomunicaciones por la Universidad Tecnológica de Panamá, y es actualmente estudiante de la M. C. en Instrumentación y Control con Línea Terminal en Electrónica. Es experto en sistemas operativos GNU Linux, y su línea de investigación son los sistemas embebidos.

Ing. Estefanía Desiree Avalos Rivera es Ingeniera Mecatrónica por la Universidad Politécnica del Centro, y es actualmente estudiante de la M. C. en Instrumentación y Control con Línea Terminal en Electrónica. Su línea de investigación son los sistemas embebidos.

Implementación de un multiplicador de punto flotante de doble precisión basado en el estándar IEEE 754-2008

José Itzcóatl Sandoval López

CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
Itzcoatl6@hotmail.com

Juan José Raygoza Panduro

CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
juan.raygoza@ucei.udg.mx

Susana Ortega Cisneros

Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
susana.ortega@gdl.cinvestav.mx

Jorge Rivera Domínguez

Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
riveraj@gdl.cinvestav.mx

Resumen

Este artículo presenta la síntesis, a partir de la descripción en VHDL de un multiplicador de punto flotante, basado en el estándar para Aritmética de Punto Flotante de IEEE (754™-2008) para microprocesadores, del cual se utiliza el formato binario para precisión doble de 64 bits. El estándar define formatos para representar diferentes tipos

de datos los cuales son: normal, subnormal, cero positivo, cero negativo, infinito positivo, infinito negativo y un no número (NaN). Muchas aplicaciones basadas en procesadores embebidos requieren la capacidad para realizar operaciones aritméticas de punto flotante, lo cual es fundamental para una mejor precisión y desempeño del sistema en el procesamiento de los datos. Además de definir una solución de hardware, también, se implementa el diseño de un multiplicador de punto fijo, mediante el algoritmo de Booth.

Palabra(s) Clave(s): FPGAS, IEEE-754, multiplicación, punto flotante, VHDL.

1. Introducción

La multiplicación es una operación necesaria para realizar operaciones complejas, a la vez es considerada como la que mayor tiempo necesita para ejecutarse. Por este motivo se requiere de una implementación eficiente para que su desempeño sea adecuado.

En algunas ocasiones, al realizar cálculos complejos, como en ingeniería, astronomía, aeronáutica, etc., los números enteros ya no son suficientes para aplicaciones donde se utilizan números muy pequeños o demasiado grandes y es indispensable incluir una parte fraccionaria o una notación científica.

El estándar de la IEEE para aritmética en punto flotante (*IEEE Standard for Floating-Point Arithmetic* [1]), especifica formatos para su uso en los sistemas informáticos, también define la precisión simple, doble y extendida así como los métodos para el intercambio de datos; este estándar es el más utilizado para resolver operaciones de alta precisión.

En este trabajo se utiliza el formato de precisión doble de 64 bits donde un dato está compuesto de signo, exponente y mantisa debidamente ordenados; el código está descrito, sintetizado y simulado en el software ISE™ 14.6 de Xilinx®.

2. Desarrollo

En esta sección se presentan los conceptos básicos que explican el comportamiento del multiplicador de punto flotante.

2.1. Multiplicación

Para realizar la multiplicación de punto flotante es necesario realizar una multiplicación de punto fijo. El rendimiento del sistema depende en gran medida del rendimiento de este multiplicador de punto fijo. Existen varios algoritmos para resolver la multiplicación, como El algoritmo de Booth, el algoritmo de Wallace, etc. El algoritmo de Booth es un método rápido y sencillo para implementar el producto de dos números binario.

2.2. Algoritmo de Booth

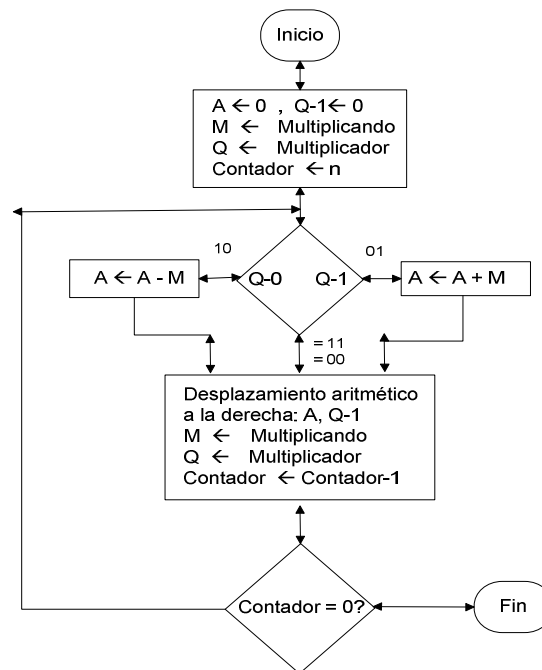


Fig. 1. Algoritmo de Booth para la multiplicación [2].

El algoritmo de Booth pretende minimizar el número de sumas que se deben realizar durante el proceso de multiplicación para reducir el tiempo en el que se realiza. En el

estado inicial se mandan a cero los bits del vector A, y al multiplicador Q se le agrega un 0 como bit menos significativo para realizar la primera comparación.

Se tiene también un contador de ciclos inicializado en n = número de bits de los multiplicandos. En cada ciclo se leen los últimos dos bits de Q y se decide si se hace la suma o resta de A y M, o simplemente se omite esa operación.

Posteriormente se hace un desplazamiento a la derecha del vector que contiene A y Q, el cual contendrá el resultado final, y se disminuye el contador. Finalmente se checa si ya se realizaron todos los ciclos, en caso de que no, se vuelve a realizar otro, y en el caso de que el contador haya llegado a cero, ya se ha terminado la multiplicación.

2.3. Punto flotante

Existen varias formas de representar números no enteros. Una de ellas es usando un punto fijo. Este tipo de representación ubica siempre el punto o coma en alguna posición a la derecha del dígito menos significativo (ver Tabla. 1). Otra alternativa comúnmente usada es la que se conoce como representación en punto flotante. Los números de punto flotante decimales normalmente se expresan en notación científica con un punto explícito siempre entre el primer y el segundo dígitos. El exponente o bien se escribe explícitamente incluyendo la base, o se usa una “e” para separarlo de la mantisa.

Mantisa	Exponente	Notación científica	Valor en punto fijo
1.5	4	1.5×10^4	15000
-2.001	2	-2.001×10^2	-200.1
5	-3	5×10^{-3}	0.005
6.667	-11	$6.667 \times e^{-11}$	0.0000000000667

Tabla 1. Ejemplos de punto flotante decimal expresado como notación científica.

Dado que un número en punto flotante puede expresarse de distintas formas que son equivalentes, es necesario establecer una única representación. Es por ello que se trabaja con números normalizados. Decimos que un número está normalizado si el dígito a la izquierda del punto está entre 0 y la base ($0 < \text{dígito a la izquierda del punto} < b$).

$$1.00 \times 10^2 \text{ Normalizado} \quad (1)$$

$$0.01 \times 10^{-1} \text{ Desnormalizado} \quad (2)$$

En particular, decimos que un número binario está normalizado si el dígito a la izquierda del punto es igual a 1.

2.4. Formato de punto flotante (Estándar IEEE 754-2008)

En esta sección se describe el formato de precisión sencilla y precisión doble para punto flotante.

2.4.1. Presión sencilla

El formato con punto flotante de precisión sencilla [1] se representa en la aritmética binaria con un bit de signo (31), (S_x), 8 bits de exponente (30 al 23), (E_x), para nuestro caso b es la base=2 y 23 bits de mantisa (22 al 0), (M_x) (ver Fig. 1), (ver Ecuación 3). Para obtener una mayor precisión, se omite el bit correspondiente a la parte entera de la mantisa y se supone "1" cuando es un número normal y "0" cuando es un subnormal, obteniendo una precisión real de 24 bits de mantisa.

$$(-1)^{S_x} \times M'_x \times b^{E_x} \quad (3)$$

Signo	Exponente	Mantisa
31	30.....23	22.....0

Fig. 2. Muestra el registro del formato de presión sencilla para punto flotante.

Para representar el exponente en punto flotante se utiliza una representación en exceso N de forma que el exponente más negativo posible para precisión sencilla, quede en 0000 0001 y el más grande de los positivos en 1111 1110 .

El estándar IEEE 754 usa como exceso 127 en precisión sencilla, el exponente está situado en un rango de -126 a +127 y es desplazado mediante la suma de 127 para obtener un valor en el rango 1 a 254, recordando que su campo tiene 8 bits (0 y 255 tienen valores especiales descritos más adelante). Cuando se interpreta el valor en coma flotante, el número es desplazado de nuevo para obtener el exponente real [4].

Por ejemplo, para representar el número 0.50_{10} en punto flotante.

- 0.50 (4)
- $(0.50 - 0) * 2 = 1 \quad d0=0$ (5)
- $(1.00 - 1) * 2 = 0 \quad d1=1$ (6)
- $0.50_{10} = 0.1_2 = 1.0_2 \times 2^{-1}$ (7)
- Exponente en exceso= $-1 + 127 = 126$ decimal = 0111 1110 binario (8)

Signo	Exponente	Mantisa
31	30.....23	22.....0
0	01111110	000000000000000000000000

Fig. 3. Representación de 0.50_{10} en punto flotante representación sencilla.

2.4.2. Formato de doble precisión para punto flotante

La representación de un número en precisión doble en el formato IEEE-754 consta de las siguientes partes:

El bit de signo está situado en la posición del registro 63. Un "1" significa un número negativo, y un "0", es un número positivo [8].

El exponente está situado entre los bits 62-52. El valor en este campo de es de 11 bits con un exceso de 1023, La mantisa es de 52 bits de longitud y ocupa los bits de 51-0. Hay un bit '1' que no está incluido en la mantisa, este bit pertenece a la parte entera del registro (ver Fig. 3).

Signo	Exponente	Mantisa
63	62.....52	51.....0

Fig. 4. Muestra el registro del formato de doble presión para punto flotante.

Por ejemplo, para representar el número 3.5 en formato de doble precisión de punto flotante El bit de signo es 0 situado en la posición 63 para representar un número positivo.

El exponente será 1024. Esto se calcula separando 3.5 como:

$$3.5 = 1.75 \times 2^1 \tag{9}$$

El desplazamiento del exponente es 1023, por lo que al añadir 1023 + 1 para calcular el valor para el campo exponente. Por lo tanto, los bits 62 a 52 serán "1000000000". La mantisa corresponde a el 1,75, que se multiplica por la potencia de 2 (2^1) para obtener 3,5. La parte entera del registro, '1' está implícito en la mantisa pero en realidad no incluido en el formato de 64 bits. Así 0.75 está representado por la mantisa. EL bit 51, es el bit más significativo de la mantisa, corresponde a $2^{(-1)}$. El bit 50 corresponde al $2^{(-2)}$ y esto continúa hasta el bit 0 lo que corresponde a $2^{(-52)}$. Representar a 0,75, los bits 51 y 50 son de 1, y el resto de los bits son 0 de. Así 3.5 como un número de coma flotante de doble precisión es:

Tipo De Dato	Signo	Exponente	Mantisa
Cero Positivo	0	$E. = 0$	$M = 0$
Cero Negativo	1	$E. = 0$	$M = 0$
Subnormal	S	$E. = 0$	$M > 0$
Normal	S	$0 < E. < 2047$	$M = X$
Infinito	S	$E = 2047$	$M = 0$
No un Número (NaN)	S	$E. = 2047$	$M > 0$

Tabla 3. Representación de tipos de datos y sus valores en precisión doble.

2.6. Consideraciones

Como consideraciones podemos asumir lo siguiente:

Hay dos ceros. $+0$ (S es 0) y -0 (S es 1).

Hay dos infinitos $+\infty$ (S es 0) y $-\infty$ (S es 1).

Los NaNs pueden tener un signo y un significando, pero estos no tienen otro significado que el que puedan aportar en pruebas de diagnóstico; el primer bit del significando es a menudo utilizado para distinguir los NaNs señalizados (SNaN: *Signalling Not a Number*) de los NaNs silenciosos (QNaN: *Quiet Not a Number*). Los primeros son utilizados cuando el dato ingresado al sistema no tiene el formato de un número válido especificado por la norma de punto flotante. Los QNaN especifican que la operación realizada con las cifras ingresadas produce un valor numérico no válido. Los NaNs y los infinitos tienen todos los bits puestos a 1 en el campo exponente [5].

3. Errores de redondeo

La representación de punto flotante tiene un número de dígitos limitado, por consiguiente no es posible representar todos los reales de forma precisa: cuando hay más dígitos de los que permite el formato, los que sobran se omiten, y se redondea.

Existen tres razones por las que esto puede ser necesario:

A) Denominadores grandes En cualquier base, cuanto mayor es el denominador de una fracción (irreducible). Un denominador lo suficientemente grande necesitará redondeo, no importa qué base o número de dígitos disponible haya.

B) Dígitos periódicos Cualquier fracción (irreducible). Por ejemplo, $\frac{1}{3}$ no es finito, ni tampoco $\frac{2}{3}$, $\frac{1}{7}$ o $\frac{5}{6}$.

C) Los números irracionales no se pueden representar como una fracción regular, requieren un número infinito de dígitos no periódicos.

D) Los tipos de redondeo más común en punto flotante son:

Hacia el cero o truncamiento: simplemente se omiten los dígitos sobrantes. Es el método más sencillo, pero introduce más error del necesario y un sesgo hacia el cero cuando se manejan sobre todo números positivos o sobre todo negativos.

Redondeo al alza: si la fracción truncada es mayor o igual que la mitad de la base, se incrementa el último dígito. Este método es el que se enseña [4].

Redondeo mitad al par: también conocido como el redondeo del banquero - si la fracción truncada es mayor que la mitad de la base, se incrementa el último dígito. Si es igual a la mitad de la base, se incrementa solamente si el resultado es par. Esto minimiza el error y el sesgo, y por eso se prefiere en contabilidad. Es el método por defecto en el estándar IEEE 754.

Número	Hacia el cero	Al alza	La mitad al par
1.4	1	1	1
1.5	1	2	2
- 1.6	-1	-2	-2
2.6	2	3	3
2.5	2	3	2
- 2.4	-2	-2	-2

Tabla 4. Ejemplo de los tipos de redondeo que existen.

4. Implementación en FPGAs

La figura 4 ilustra el diagrama esquemático principal del circuito multiplicador de punto flotante de 64 bits, el cual está formado a base de bloques diseñados para el cálculo de cada tipo de datos y sus combinaciones. Éste es implementado en FPGAs de Xilinx ® [6].

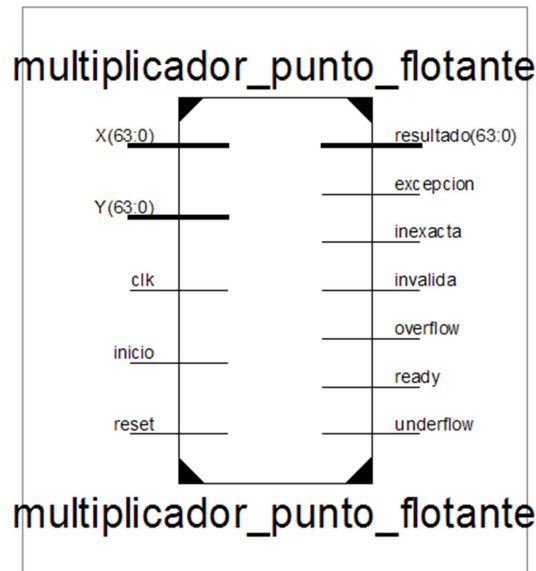


Fig. 6. Diagrama general del multiplicador de punto flotante.

Las señales de entrada al módulo son los siguientes:

1. Clk: Señal de reloj.
2. inicio: señal que indica que el dato está listo a ser procesado (en alto para iniciar la operación).
3. X: (Multiplicando de entrada, 64 bits).
4. Y: (Multiplicador de entrada, 64 bits).

Las señales de salida del módulo son los siguientes:

5. Resultado: Indica el resultado de la operación (salida de la operación, 64 bits).
6. Listo: indica que termino el cálculo (va alto cuando la salida está disponible).
7. *Underflow*: indica que el resultado no pertenece al rango mínimo de números que puede representar la unidad.
8. *Overflow*: indica que el resultado no pertenece al rango máximo de números que puede representar la unidad
9. Inexacta: si existiera más de una señal de las excepciones activas se activa.
10. Inválida: multiplicar 0 por infinito, indica que el resultado es no un número (NaN).
11. Excepción: indica alguna de las siguientes señales activas: *underflow*, *overflow*, inexacta o invalida.

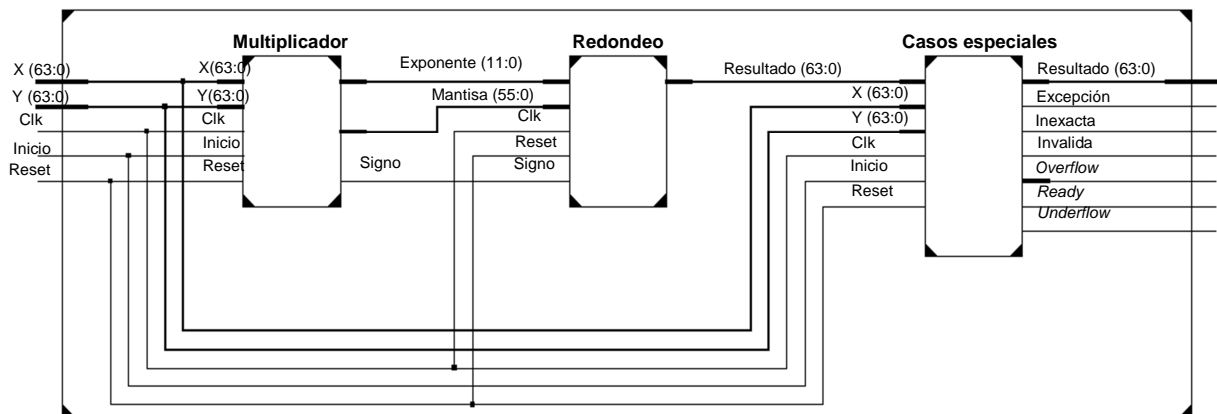


Fig. 7. Diagrama esquemático principal.

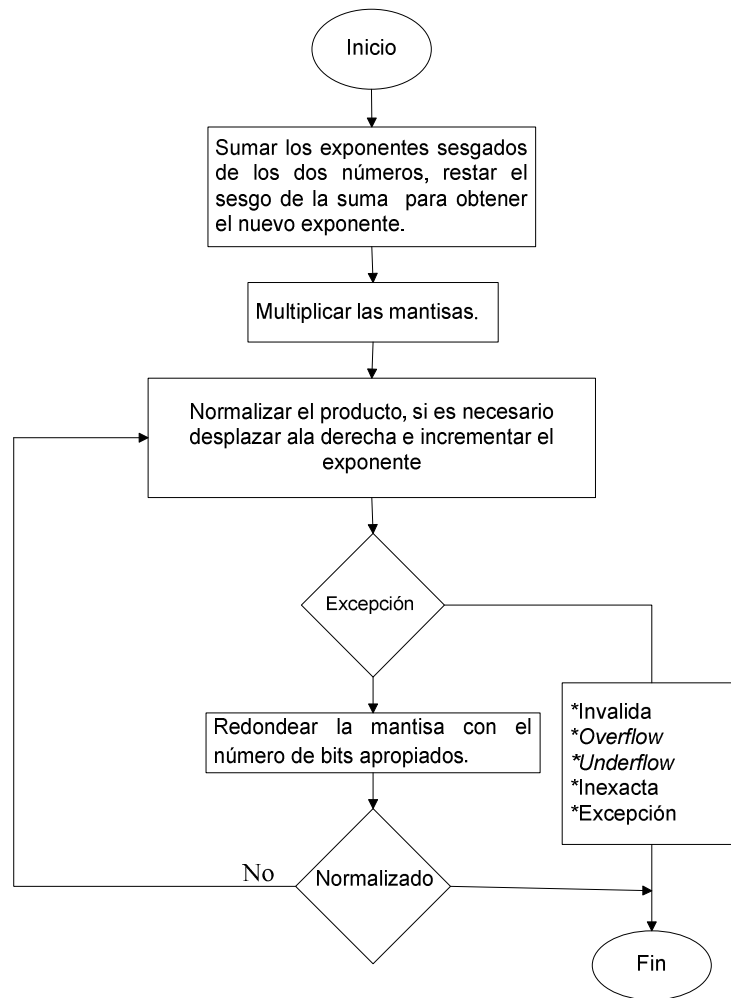


Fig. 8. Funcionamiento del multiplicador de punto flotante.

4.1. Bloque multiplicación

La multiplicación de punto flotante se realiza en el bloque "Multiplicador". Este bloque recibe las señales exteriores del módulo general del multiplicador de punto flotante, cuenta con 5 señales de entrada y 3 de salida. Las señales de entrada son las siguientes: el multiplicando "X(63:0)" de 64 bits, que admite datos en formato de punto flotante precisión doble, normalizados, el multiplicador "Y(63:0)" que admite el mismo tipo de datos que el multiplicando, una señal de reloj "Clk", una señal de inicio que

indica cuando el dato está listo para ser procesado (“0” para indicar alto, “1” para indicar que el dato está listo para procesar) y una señal de reinicio “Reset” (“1” para reiniciar).

Las señales de salida: El “signo”, que es de longitud de un bit, el “exponente (11:0)”, con una longitud 12 bits, y la “mantisa (55:0)” que tiene una longitud de 56 bits. Estos 3 datos representan el producto de la multiplicación de punto flotante, se puede observar que estos datos no se encuentran aún representados de acuerdo a la norma IEEE 754-2008.

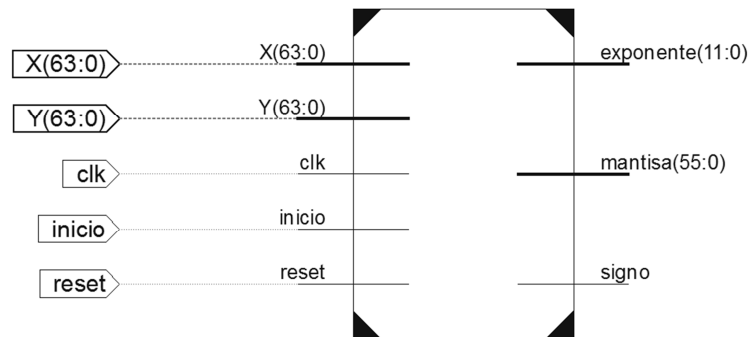


Fig. 9. Bloque multiplicación.

La salida de la mantisa tiene una longitud de 56 bits. El bit más significativo es un '0' para permitir un desbordamiento potencial en el módulo de redondeo. El primer bit '0' es seguido por '1' para los números normalizados, o "0" para los números no normalizados. Esto es la parte entera del registro. A continuación, los 52 bits de la mantisa más dos bits adicionales, que se utilizan para fines de redondeo.

4.1.1. Funcionamiento del bloque multiplicación

La mantisa del operando “X” y el bit '1' que no está incluido en la mantisa (la parte entera, para los números normalizados) se almacenan en un registro de 53 bits (mult a).

La mantisa del operando “Y” y el bit '1' que no está incluido en la mantisa (la parte entera, para los números normalizados) se almacenan en un registro de 53 bits (mult b).

Los exponentes se almacenan en registros de 12 bits. Los exponentes de X y Y se suman juntos y luego se les resta el valor de (1022) de la suma de X y Y. Si el exponente resultante es menor que 0, será necesario realizar un desplazamiento a la derecha en el producto de las mantisas. Este valor se almacena en el registro (*exponent_under*). En este caso el exponente final de la salida será 0 y el resultado será un número desnormalizado. Si *exponent_under* es mayor de 52, de la mantisa se desplaza fuera del registro de productos, y la salida será 0, y se activara una señal "underflow".

Al utilizar el multiplicador de Booth, todos los 53 bits de *mul_a* por 53 bits de *mul_b* resulta un producto de 106 bits, se almacena en el registro (*producto*). Después, se realiza un desplazamiento, de acuerdo a la cantidad de ceros a la izquierda, de este registro, si no hay un '1' en el bit más significativo del producto. El número de ceros a la izquierda en el registro (de productos) se cuentan por la señal (*product_shift*). El exponente de salida también será reducido por (*product_shift*), de acuerdo a la cantidad de ceros que se hayan encontrado.

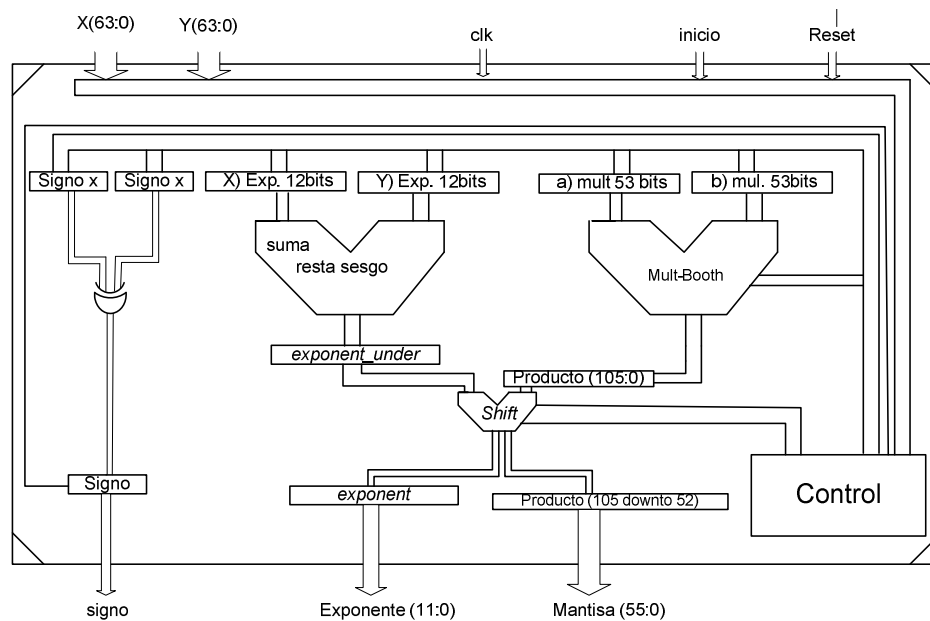


Fig. 10. Bloque de flujo de datos del multiplicador.

4.2. Bloque redondeo

La operación de redondeo se realiza en este módulo, las entradas a este bloque provienen del módulo anterior, multiplicador y son las siguientes:

Signo (1 bit), mantisa (56 bits), y exponentes (12 bits). La mantisa incluye un bit extra '0' como el bit mas significativo, seguido del bit de la parte entera, y dos bits en la parte menos significativa adicionales, y 52 bits de la mantisa. El exponente tiene un '0' en el bit mas significativo para provocar desbordamientos en caso de ser necesario. si el registro fuera de 11 bits, Un desbordamiento daría lugar a un valor de 0 en el campo de exponente, y el resultado final de la operación sería incorrecta. Cuenta con una señal de reloj (clk), un reinicio (Reset) y una salida para mostrar el resultado (resultado) con la norma IEEE 754-2008. (Ver Figura. 8).

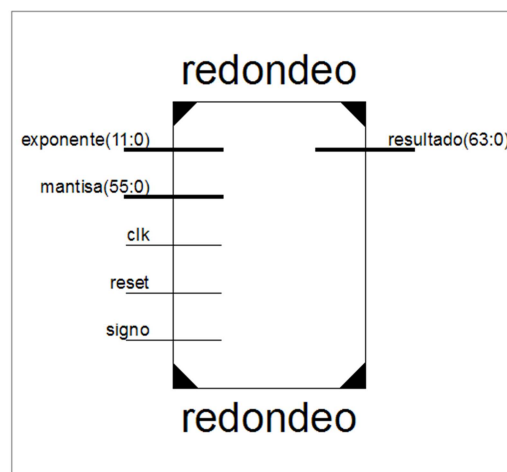


Fig. 11. Bloque redondeo.

Se redondea al modo alza de la siguiente manera: si el bit en la posición 1 y el bit en la posición 0 es un "1" de la mantisa, se redondea. Para llevar a cabo el redondeo, se suma la mantisa a la señal_redondeo. La señal_redondeo tiene un '1' en el espacio de bit que se alinea con el bit menos significativo de la mantisa de 52 bits.

Este '1' en la señal_redondeo se alinea con el 2 bit del registro mantisa (recordando que sus posiciones van de 55 a 0). Los bits 1 y 0 del registro mantisa son los bits restantes de más, y éstos son eliminados en el registro resultado.

4.3. Bloque casos

En este bloque, todos los casos especiales se comprueban, y si se encuentran, la adecuada se activa la salida: excepción, inexacta, inválida, *overflow*, etc.

Se afirma cuando se cumplen las condiciones de cada caso. Algunos casos especiales son: Multiplicar 0 por infinito: activa la señal inválida; multiplicar por un dato fuera del rango máximo o mínimo de los valores que puedan representarse será un *overflow* o *underflow*, entre otros.

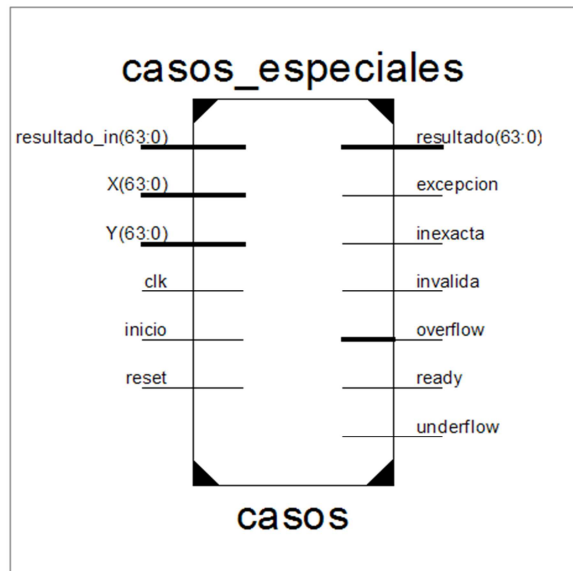


Fig. 12. Bloque casos especiales.

5. Resultados

El diseño propuesto se verificó con simulaciones del sistema completo en el software de Xilinx ISE 14.6 en donde se implementó el sistema en lenguaje VHDL. Los resultados de la simulación del circuito multiplicador de punto flotante se muestran en la figura 10

en donde el cursor en el tiempo 700ns indica el dato “X” que contiene “1111111111” de exponente y la mantisa es cero. Revisando la tabla 3, observamos que éste es un dato inválido y, de igual manera, que el dato “Y” que contiene cero.

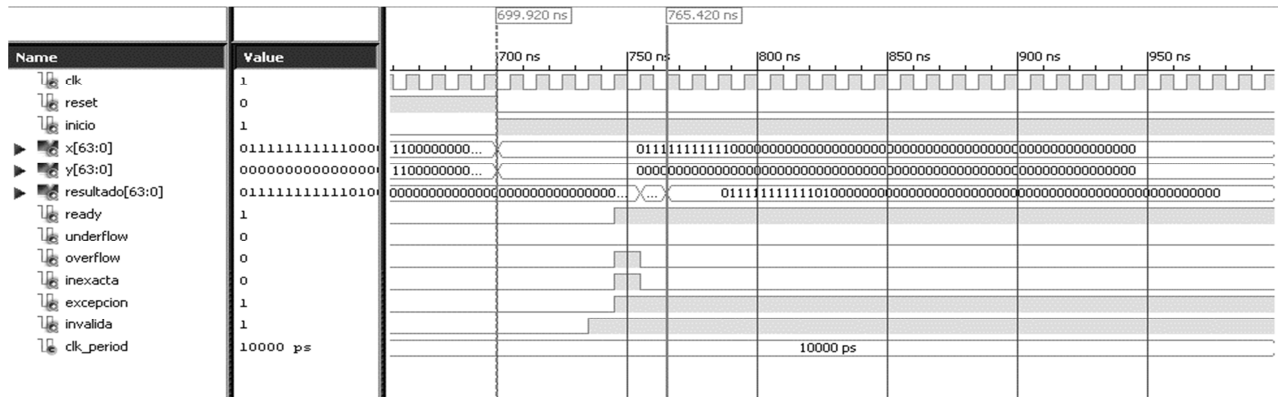


Fig. 13. Multiplicación de un número infinito x un cero. La bandera excepción se activa y la bandera inválida también se activa.

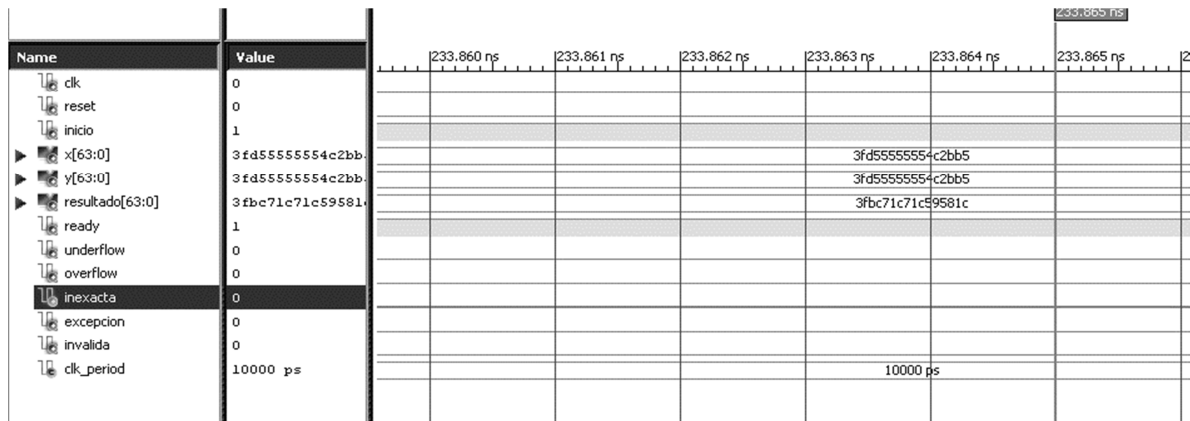


Fig. 14. Multiplicación de 0.333333333 x 0.333333333= 0.111111110888888889 (formato decimal).

Los resultados de la figura 11 se muestran en formato doble precisión y base hexadecimal.

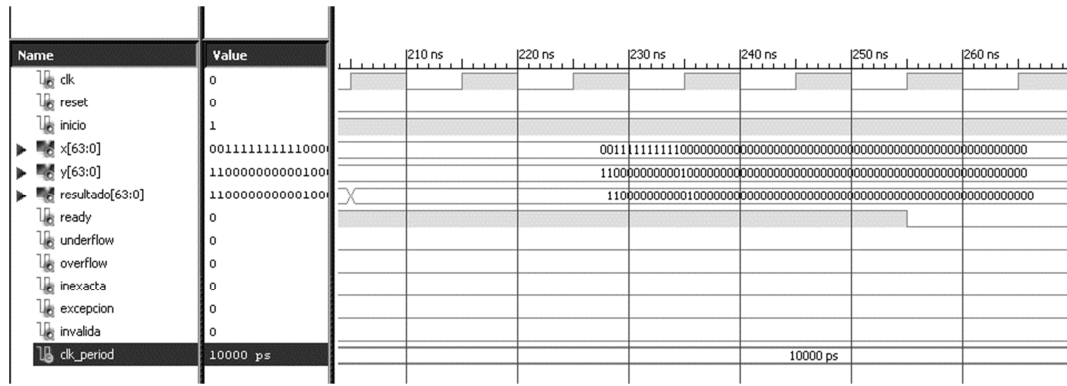


Fig. 15. Multiplicación de 1 x -3 en punto flotante doble precisión.

Se observa en la figura 12 que el multiplicador Y (63:0) es igual al producto “resultado (63:0)”. Los resultados se muestran en formato doble precisión y base binaria.

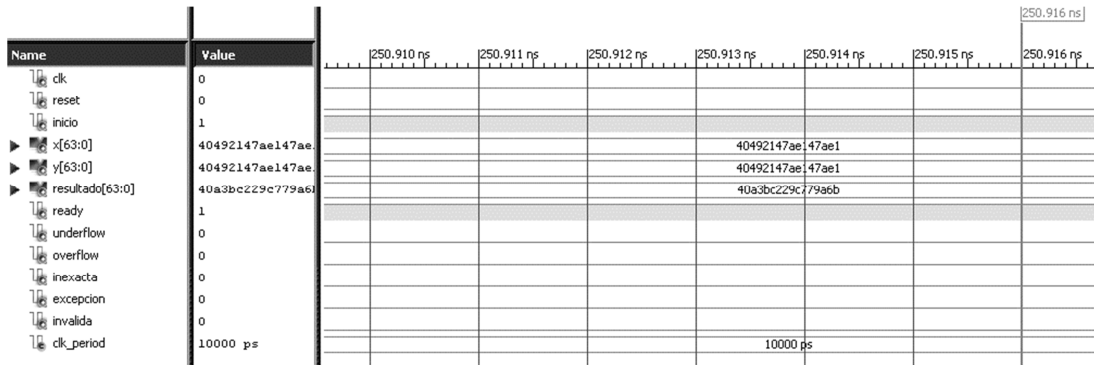


Fig. 16. Multiplicación de 50.26 x 50.26=2526.0676 (formato decimal).

Los resultados de la figura 13 se muestran en formato doble precisión y base hexadecimal.

Los resultados de ocupación de los recursos en FPGAs se muestran en la tabla 5 y tabla 6 en donde se puede observar que el circuito multiplicador puede ser implementado en las 3 FPGAs.

Dispositivo Utilizado Ocupación	Virtex 6 (XC6VLX240T)			Virtex 5(xc5vtx240t)		
	Usadas	Total	%			
Número de <i>Slice</i>	1,765	150,720	1%	968	149,760	1%
Número de <i>LUTs</i>	966	301,440	1%	3,343	149,760	2%
Número de <i>IOBs</i> utilizadas	201	400	50%	201	680	29%

Tabla 5. Tabla de ocupación para Virtex 6 y Virtex 5.

Dispositivo Utilizado Ocupación	Virtex 7 (xc7vx330t)		
	Usadas	Total	%
Número de <i>Slice</i>	966	408,000	1%
Número de <i>LUTs</i>	1,863	204,000	1%
Número de <i>IOBs</i> utilizadas	201	600	33%

Tabla 6. Tabla de ocupación para Virtex 7.

6. Discusión

De los tres módulos de operación, el módulo de la multiplicación es considerado como la etapa crítica y de mayor importancia para el funcionamiento correcto del sistema. Esto se debe a que el sistema incluye una multiplicación de punto fijo de 53x53 bits que en términos de hardware se traduce en una gran cantidad de componentes dependiendo del algoritmo utilizado. El sistema propuesto también es capaz de detectar condiciones de desbordamiento de datos (*underflow* y *overflow*) y cuando un número de entrada es igual a cero [7]. Por lo tanto, este sistema es completamente funcional.

7. Conclusiones

En este proyecto, se presentó la implementación de un multiplicador en representación de números de punto flotante. Desde la aplicación se puede concluir de la siguiente manera. Se comprobó que el circuito propuesto puede ser implementado en las 3 familias de FPGAs de Xilinx debido a su nivel de utilización de recursos de ocupación. Una de las ventajas de implementar este multiplicador en FPGAs es que tiene procesos independientes al mismo tiempo y permite agilizar los cálculos, como la suma de

exponentes y la multiplicación de mantisas. La etapa de la multiplicación es la parte más crítica. La cual conlleva más conexiones a nivel hardware y depende mucho del algoritmo utilizado. Como trabajo futuro se podría implementar en hardware un algoritmo que permita hacer más veloz la etapa de multiplicación, como por ejemplo utilizar el árbol de Wallace, o un multiplicador de Booth, para resolver las operaciones.

8. Referencias

- [1] Institute of Electrical and Electronics Engineers, Inc. IEEE Standard for Floating-Point Arithmetic. IEEE Xplore®. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=4610935. Abril 2015.
- [2] N. Goyal, K. Gupta, R. Singla. Study of Combinational and Booth Multiplier. International Journal of Scientific and Research Publications. Vol. 4. No. 5. May 2014.
- [3] R. Shankar, P. Gour, B. Bihari, "Design and Implements of Booth and Robertson's multipliers algorithm on FPGA". Soni International Journal of Engineering Research and Applications (IJERA).
- [4] J. Rapallini, S. Ledesma, F. Costantino, J. R. Osio, Matemática de PuntoFlotante. http://www.edudevices.com.ar/download/articulos/buceando/BC_MCU_42_ED.pdf. Abril 2014.
- [5] Consideraciones acerca del Estándar IEEE 754. <http://www.led.uc.edu.py/micro2/tp2/pf/pag2.htm>. Abril 2015.
- [6] Xilinx Inc. Xilinx ISE 14.6 Software Manuals and Help. http://www.xilinx.com/support/index.html/content/xilinx/en/supportNav/ip_documentation.html.
- [7] E. Vilches. Números de punto flotante. <http://www.erikavilches.com/Anterior/TC1004.01.200811/diapositivas/Punto%20Flotante%202.pdf>. Abril 2014.

- [8] S. Orley, J. Mathews. IEEE 754 Format. <http://www.oxfordmathcenter.com/drupal7/node/43>. Abril 2015.

- [9] M. Morris, *Arquitectura de Computadoras*. Tercera Edición. 1994. Pearson Educación. México. 547 pp.

Diseño y construcción de un limnógrafo electrónico

Felipe Santiago Espinosa

Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica,
Carretera a Acatlima km 2.5, Huajuapán de León Oaxaca

Teléfono: (01-953) 53 203 99 ext. 555

fsantiag@mixteco.utm.mx

Gabriela Álvarez Olguín

Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica,
Carretera a Acatlima km 2.5, Huajuapán de León Oaxaca

Teléfono: (01-953) 53 203 99 ext. 550

galvarez@mixteco.utm.mx

Fermín Hugo Ramírez Leyva

Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica,
Carretera a Acatlima km 2.5, Huajuapán de León Oaxaca

Teléfono: (01-953) 53 203 99 ext. 555

hugo@mixteco.utm.mx

Resumen

Un limnógrafo es un mecanismo que mide el nivel del agua, está formado por un flotador unido a una plumilla que marca el nivel en un papel fijado a un tambor, que gira mediante un mecanismo de relojería, generando una gráfica de niveles contra el tiempo conocida como limnograma. El limnograma es interpretado y capturado para usarse en diversos estudios hidrológicos, este procedimiento es manual y queda expuesto a errores. Buscando automatizar el registro y captura se desarrolló un limnógrafo electrónico con el microcontrolador ATmega8, el nivel se obtiene por medio de un

sensor ultrasónico y los datos se almacenan en una memoria EEPROM para preservarlos en ausencia de energía, se utiliza la versión 24PL512 de Microchip, la cual tiene una interfaz I2C. El periodo de muestreo es configurable por medio de dos botones. Mediante una pantalla LCD se muestra al usuario el estado del sistema. También se agrega un adaptador USB-TTL para la comunicación con una computadora, desde donde se puede solicitar el periodo de muestreo, el total de muestras y el conjunto de datos; así como comandar para reiniciar al contador de muestras. Todo desde una aplicación en LabVIEW, la cual también permite graficar las muestras.

Palabra(s) Clave(s): ATmega8, LabVIEW, limnógrafo, nivel de agua.

1. Introducción

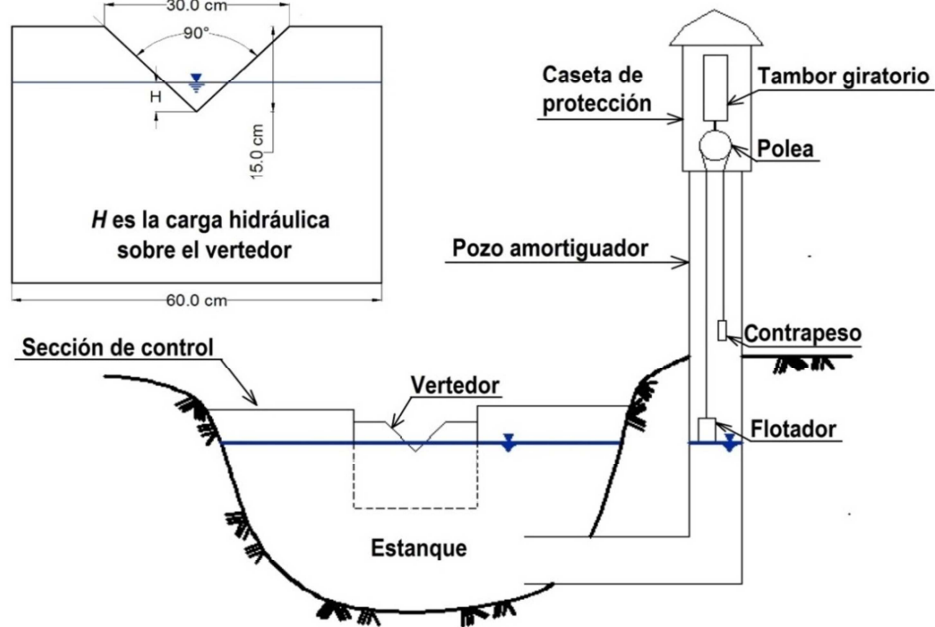
La medición de los niveles de agua en estanques, canales, ríos y presas, se utiliza para estimar el componente del ciclo hidrológico denominado escurrimiento; también conocido como gasto de un cauce, rendimiento de una cuenca o aportación líquida [1]. Los datos de escurrimiento son la base de una serie de estudios que contribuyen al bienestar social, entre los que están: la evaluación de la disponibilidad de agua superficial, la estimación de eventos de diseño en obras hidráulicas y la caracterización de sequías e inundaciones.

Las mediciones de los niveles de un cuerpo de agua se realizan en una estación de aforos formada por diferentes dispositivos e instrumentos de medición, de los cuales el limnógrafo se destaca por ser el único capaz de proporcionar registros continuos. Tradicionalmente en México el registro de datos con limnógrafos se realiza a través de un mecanismo formado por un flotador unido a una plumilla que marca los niveles del agua en un papel fijado a un tambor, que gira mediante un mecanismo de relojería, en la figura 1(a) se muestra una estación de aforo ubicada en la Universidad Tecnológica de la Mixteca, la cual es controlada y mantenida por el Instituto de Hidrología de la

misma institución, mientras que en la figura 1(b) se esquematiza a la imagen 1(a) para distinguir sus diferentes partes.



(a)



(b)

Fig. 1. (a) Estación de aforo instalada en la UTM, conformada por un limnógrafo convencional y una sección de control, (b) representación esquemática de la misma.

El resultado del limnógrafo es una gráfica de los niveles contra el tiempo conocida como limnograma [2]. Con esta información es posible determinar los gastos en función de la elevación, para ello se debe contar con una sección de control (figura 1), en la cual existe una única relación entre el gasto de descarga y la carga hidráulica, cuya gráfica es conocida como curva elevaciones-gastos, en la figura 2 se muestra un ejemplo de esta curva.

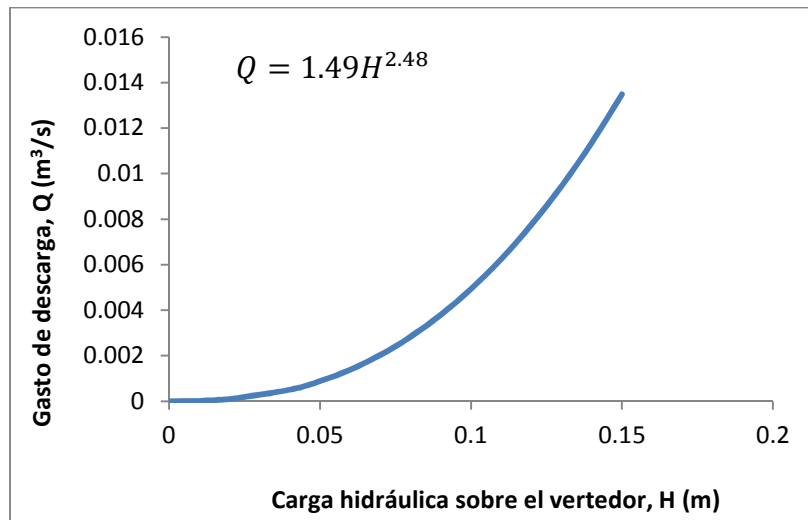


Fig. 2. Curva elevaciones-gastos, correspondiente a la sección de control de la estación de aforo instalada en la Universidad Tecnológica de la Mixteca.

El mecanismo del limnógrafo convencional representa una opción sencilla y económica, sin embargo, la interpretación visual del limnograma suele ser una tarea laboriosa debido a la gran cantidad de información contenida en una sola hoja de papel.

En las estaciones de aforo localizadas en zonas remotas, donde no siempre es posible contar con un operador permanente, se dificulta la medición constante de gastos. Para resolver este problema, en diferentes investigaciones se presentan propuestas para automatizar el monitoreo, registro y procesamiento de datos, adaptando dispositivos electrónicos a los mecanismos convencionales de medición existentes, como lo reflejan los trabajos de Vidal *et al.* [3], Morales [4] y Tamari *et al.* [5].

En la UTM se implementó un limnógrafo electrónico para las estaciones de aforo mantenidas por el instituto de Hidrología, el limnógrafo está basado en un ATmega8, un microcontrolador AVR de 8 bits de ATMEL [6], este dispositivo es suficiente para el manejo de periféricos que permiten obtener el nivel, capturar y almacenar la información, así como el envío de los datos a una computadora para su posterior procesamiento.

2. Desarrollo

Para la implementación del sistema se utilizó la Metodología Simplificada para Desarrollar Sistemas basados en Microcontroladores [7], esta metodología comprende los pasos que se irán cubriendo en los apartados de esta sección.

2.1. Planteamiento del problema

En la Figura 3 se muestran los elementos propuestos para el limnógrafo electrónico. La pantalla LCD proporcionará una interfaz visual para conocer el estado actual del sistema. El sensor ultrasónico trabaja por rebote, emite una ráfaga de pulsos y espera su retorno, conociendo el tiempo que tardó desde su emisión hasta su regreso, se estima la distancia a donde encontró un obstáculo, con este sensor se medirá la distancia al nivel del agua.



Fig. 3. Requerimientos del Limnógrafo Electrónico.

Las mediciones se irán almacenando en la memoria del sistema, el periodo de almacenamiento debe ser configurable para seleccionar la mejor opción de acuerdo con

las condiciones ambientales, por ello, se requieren dos botones para configurar el periodo de muestreo. Con un botón se dará paso a la configuración y con el otro se podrá seleccionar uno de los intervalos de tiempo preestablecidos, se consideran las siguientes opciones: 1, 5, 10, 20 y 30 minutos.

La memoria en donde se almacenen los datos debe ser no-volátil, lo que significa que debe mantener la información aún en ausencia de energía, es importante que la información no se pierda si después de algunos días de operación llega a ocurrir una falla temporal en la alimentación del sistema.

El sistema debe contar con un puerto USB para establecer la comunicación con una computadora, por este medio se enviarán los datos almacenados para su interpretación y graficado por medio de una aplicación. Desde la computadora se enviarán comandos que el sistema interpretará para que les pueda dar respuesta.

2.2. Requerimientos de hardware y software

En cuanto al Hardware, el primer elemento a seleccionar es el microcontrolador, se empleará un ATmega8 porque tiene 3 puertos que resultan suficientes para los periféricos requeridos.

Adicionalmente se requiere de una pantalla LCD, empleándose una versión alfanumérica de 2 renglones y 16 caracteres por renglón. Así como los botones de configuración.

El sensor ultrasónico a utilizar es el HC-SR04, tiene un alcance hasta de 4 metros que resulta suficiente para la aplicación y, aunque su resolución es de 3 mm, por su disponibilidad y costo se considera adecuado para el desarrollo del primer prototipo [8].

El ATmega8 no cuenta con un puerto USB pero si tiene una USART, por lo que se empleará un adaptador de USB a TTL, de manera que para la aplicación de la computadora se presentará como un puerto serial (COM) convencional.

La memoria interna del microcontrolador no le proporciona suficiente autonomía porque los datos deben mantenerse aún ante ausencias breves de energía y el ATmega8 sólo tiene 512 bytes de EEPROM. Para ello se elige al dispositivo 24LC512, se trata de una memoria EEPROM de 64 Kbytes (512 Kbits) con interfaz I2C [6]. Con esta capacidad y seleccionando el periodo de muestreo más corto (1 minuto), al sistema se le dará una autonomía de 22 días puesto que cada dato ocupa 2 bytes. En el periodo más largo (30 minutos) la autonomía es de más de 600 días. Con la interfaz I2C se tiene la ventaja de que sólo se utilizan 2 terminales del microcontrolador y dos resistores de 2.2 K Ω para *pull-up*.

El último elemento de hardware a considerar es un cristal de 32.768 KHz, con el cual se operará al temporizador 2 del ATmega8 para que genere los intervalos de tiempo de muestreo, con el pre-escalador interno se pueden manejar periodos correspondientes a fracciones o múltiplos exactos de segundos reales.

Referente al software, se requiere de una biblioteca básica para el manejo del LCD con funciones básicas para su inicialización y manejo. Para simplificar el programa principal se planea que los recursos internos del MCU sean atendidos por interrupciones.

Además del programa del ATmega8, el limnógrafo debe complementarse con una aplicación para una computadora, por medio de la cual se haga la transferencia de los datos almacenados en la memoria del sistema, se considera el uso de LabVIEW para simplificar su desarrollo.

2.3. Diseño del hardware

El microcontrolador ATmega8 tiene tres puertos, puertos B y D de 8 bits, y puerto C de 7 bits, pero como el bit PC6 también corresponde con el *reset* del sistema, no se puede emplear para un propósito general. Dado que algunos elementos de hardware serán manejados por los recursos del MCU, éstos son los primeros que deben ubicarse porque utilizan las terminales dedicadas a cada recurso.

Para el oscilador externo se tienen destinadas las terminales PB6 y PB7 (OSC1 y OSC2, respectivamente), ahí se conectará el cristal de 32.768 KHz, que será la base de tiempo para el temporizador 2 del ATmega8.

En el puerto C está la terminal de *reset* (PC6) y en ella se colocará un botón. También está disponible la interfaz I2C, ocupando los pines PC4 (SDA, para datos) y PC5 (SCL, para reloj), en esta interfaz se conectará la memoria 24LC512.

Con respecto al puerto D, el adaptador USB-TTL se conectará en las terminales PD0 (RXD) y PD1 (TXD), para ser manejado por la USART del ATmega8. Los botones de configuración se manejarán con las interrupciones externas del MCU, ocupando los pines PD.2 (INT0) y PD.3 (INT1).

Después de colocar los dispositivos que requieren terminales dedicadas del MCU sólo falta por ubicar al LCD y al sensor ultrasónico. El LCD se manejará con una interfaz de 4 bits y se puede conectar en el puerto B, en donde aún quedan disponibles las terminales de PB0 a PB5, que resultan suficientes. El sensor ultrasónico se conectará en los dos bits menos significativos del puerto C, empleando PC0 como la salida para el disparo del sensor y PC1 para la entrada del eco que se recibe como respuesta.

En la figura 4 se muestra el diseño completo del hardware, aún queda disponible la parte alta del puerto D y dos terminales del puerto C para posibles expansiones del sistema.

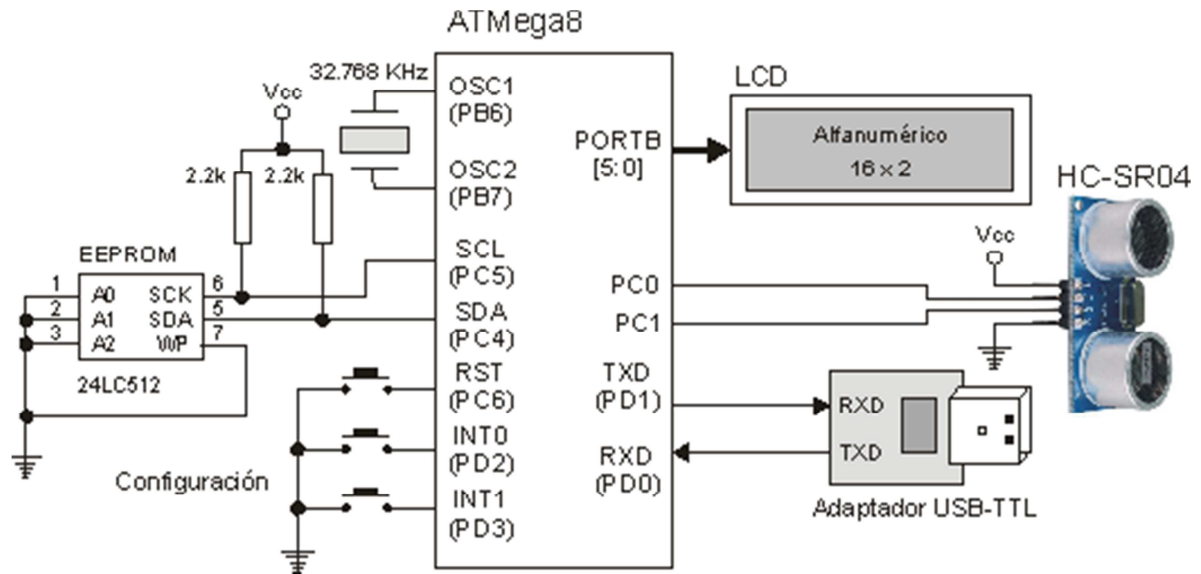


Fig. 4. Diseño del hardware para el limnógrafo electrónico.

2.4. Diseño del software

El sistema está soportado principalmente por los recursos del microcontrolador y por lo tanto, su funcionalidad está determinada por los eventos que ocurren en ellos. El programa principal básicamente seguirá el algoritmo de la figura 5, en donde se realiza la inicialización de los recursos. En las rutinas de atención a las interrupciones (ISR's) se estarán realizando las diferentes tareas del sistema.

1. Configuración de Entradas y Salidas
2. Configuración de las Interrupciones Externas
3. Inicialización del LCD
4. Lectura de la EEPROM interna del periodo de muestreo
5. Lectura de la EEPROM interna del total de datos
6. Configuración del Temporizador 2
7. Configuración de la USART
8. Habilitación de interrupciones
9. Lazo infinito ocioso

Fig. 5. Algoritmo del programa principal.

En los pasos 3 y 4 se realizan lecturas de la EEPROM interna, en ese espacio se almacenará la configuración elegida por el usuario para la toma de muestras y el total de datos almacenados, al usar una memoria no volátil para estos datos, esta información no se pierde aún en ausencia de energía. Cuando el microcontrolador se programe por primera vez se considerará un periodo de muestreo de 1 minuto. Los datos también serán almacenados en memoria EEPROM, pero en el dispositivo 24LC512, que es una memoria EEPROM de 64 Kbytes (512 Kbits) con interfaz I2C.

Los recursos que podrán generar eventos son:

- Interrupciones externas (INT0 e INT1), con la INT0 se dará paso a la configuración, mostrando el valor actual y permitiendo su modificación, los ajustes se harán con la INT1 y con la misma INT0 se aceptará y registrará el cambio, en ese momento también se respaldará en EEPROM interna. La INT1 será ignorada si con anterioridad no se presionó la INT0, con la INT1 habilitada sólo se mostrarán las diferentes opciones de configuración.
- Temporizador 2, dado que este recurso será manejado con el oscilador externo de 32.768 KHz, generará la base de tiempo para la toma de muestras. El temporizador 2 se configurará para que desborde cada 2 segundos, con un desbordamiento se incrementará un contador para luego comparar y evaluar si ya se debe almacenar de un dato. Con la finalidad de almacenar información más precisa, la distancia se medirá 10 veces y se obtendrá el promedio, este dato es el que será almacenado.
- Puerto Serie (USART), la comunicación con una computadora es esencial para el envío de la información almacenada. El microcontrolador podrá recibir comandos desde una computadora en cualquier momento, por medio de los cuales se le solicitará que realice una de las siguientes acciones:
 - Envío del total de datos (un entero formado con 2 bytes, iniciando con la parte alta).
 - Envío del periodo de muestro (un entero corto de 1 byte).
 - Envío de los datos (2 bytes por dato, enteros representando milímetros).

- Inicialización del contador de datos, con ello se pondrán ceros en las localidades destinadas para este parámetro en la memoria local. No es necesario limpiar todas las localidades de la memoria, con la puesta en ceros de este parámetro las nuevas escrituras remplazarán a las anteriores.

Para el cálculo de la distancia se considera que la velocidad de una señal de ultrasonido en el aire es: $v = 340 \text{ m/s}$, puesto que el MCU trabaja en el orden de μS y la distancia se medirá en mm, $v = 0.34 \text{ mm}/\mu\text{S}$.

Para calcular la distancia, se parte del hecho que $v=d/t$ (velocidad = distancia/ tiempo). Despejando d se obteniendo $d = v \times t$, dado que v tiene el valor constante previamente citado, únicamente se debe medir el tiempo que tardan en viajar los pulsos ultrasónicos.

El sensor HC-SR04, después de ser disparado, genera un pulso en alto cuya duración corresponde al tiempo que tardó en recibir la ráfaga ultrasónica de respuesta. Ese tiempo debe dividirse entre 2 debido a que involucra la ida y retorno, por lo tanto, conociendo a t en μs se tiene que

$$d = 0.17 \times t \text{ mm}, \quad (1)$$

con t en $\mu\text{segundos}$ la distancia se obtiene en milímetros.

En el microcontrolador se empleará al temporizador 1 para medir el ancho del pulso, este temporizador es de 16 bits, por lo que puede medir pulsos con ancho de hasta 65 535 μs , muy superior a lo requerido por la aplicación.

2.5. Implementación de hardware

En la figura 6 se muestra una fotografía con el hardware implementado en una tablilla de pruebas. Se resalta al microcontrolador ATmega8, al sensor ultrasónico, la memoria EEPROM y al adaptador USB-TTL. Al cristal se le agregan dos capacitores cerámicos para estabilización.

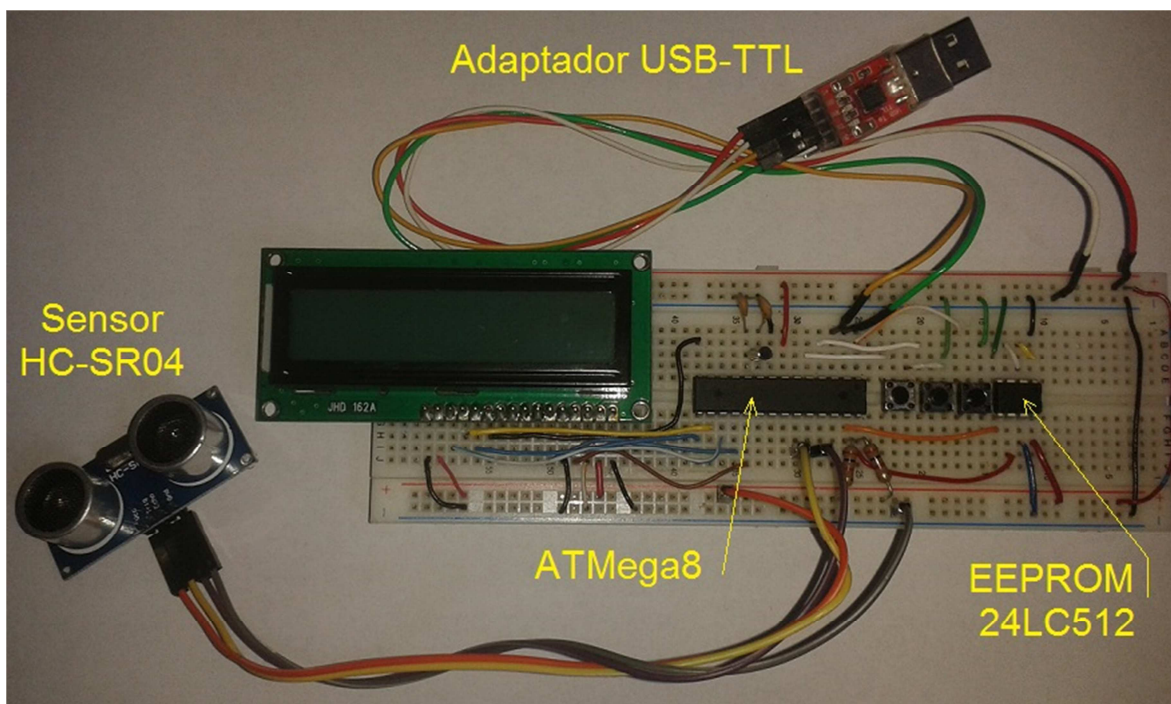


Fig. 6. Implementación del hardware.

2.6. Implementación del software

El programa fue desarrollado en lenguaje C, el programa principal sigue fielmente al algoritmo mostrado en la figura 5, como puede verse en la secuencia de código ilustrada en la figura 7.

```
int main(void){

    // Configuración de entradas y salidas
    DDRB = 0x3F;    // PORTB [5:0] - LCD, 6 y 7 para el oscilador
    DDRC = 0x01;    // PC0 - Salida de disparo y PC1 - Entrada de Eco
    DDRD = 0x02;    // Ent: RXD(0),INT0(2),INT1(3), Sal: TXD(1)
    PORTD = 0x0C;   // Pull-Up para las interrupciones

    // Configuración de las interrupciones externas
    MCUCR = 0x0A;   // Flanco de bajada
    GICR = 0xC0;    // Sólo habilita la INT0

    // Inicializa LCD
    LCD_reset();
    LCD_write_cad("    - U T M -        LIMNIGRAFO 1.0", 31);

    // Obtiene información de EEPROM interna
    Muestreo = eeprom_read_byte(&T_Muestreo);
    TDatos    = eeprom_read_word(&T_Datos);

    delay_Seg(2);
    sprintf(Cadena, "Muestreo= %2d min  Muestras = %3d",
                Muestreo, TDatos );
    LCD_write_cad(Cadena, 32);

    // Configuración del Temporizador 2
    ASSR = 0x08;    // Reloj externo
    TCCR2 = 0x06;   // División entre 256
    TIMSK = 0x40;   // Interrupción por desbordamiento del TIMER2

    // Configuración de la USART
    UBRRL = 12;     // Transmisión a 9600 baudios
    UBRRH = 0;
    UCSRA = 0x02;
    UCSRB = 0x98;   // Habilita TX y RX con interrupción por RX
    UCSRC = 0x86;   // Datos de 8 bits, 1 bit de paro y sin paridad

    delay_Seg(2);  // Sistema Operando
    LCD_write_cad("    SISTEMA EN        OPERACION", 29);

    sei();         // Habilitador global de las interrupciones

    while(1)
        asm("NOP");
}
```

Fig. 7. Programa principal.

Puede notarse que, con excepción de los mensajes iniciales, el procesamiento no se realiza en el programa principal, éste se distribuye en las rutinas que dan servicio a las interrupciones cuya funcionalidad fue descrita en la sección 2.4, el programa principal queda en un lazo ocioso, es decir, sin realizar operaciones.

2.7. Integración

El programa se editó y compiló desde el entorno AVRStudio, generando el archivo binario (.HEX) para ser descargado en el dispositivo. La evaluación del programa fue exitosa excepto porque fue necesario dar formato a la información antes de enviarla a la computadora.

3. Aplicación de LabVIEW

En LabVIEW se desarrolló una aplicación para enviar comandos al sistema y obtener información, básicamente son cuatro comandos. En la figura 8 se muestra el diagrama de secuencias ilustrando la relación entre el microcontrolador y la computadora, por medio de la aplicación en LabVIEW.

Para enlazar al microcontrolador con la computadora se utiliza la Arquitectura de Software para Instrumentos Virtuales (VISA, *Virtual Instrument Software Architecture*), que es un estándar para configuración y programación de sistemas de instrumentación por medio de las interfaces GPIB, VXI, PXI, Serial, Ethernet y/o USB [10].

Para la comunicación serial, VISA provee los módulos de LabVIEW que permiten la configuración del puerto, escrituras y lecturas, así como el cierre del puerto para liberarlo de la aplicación y posibilitarlo para ser usado en otras aplicaciones.

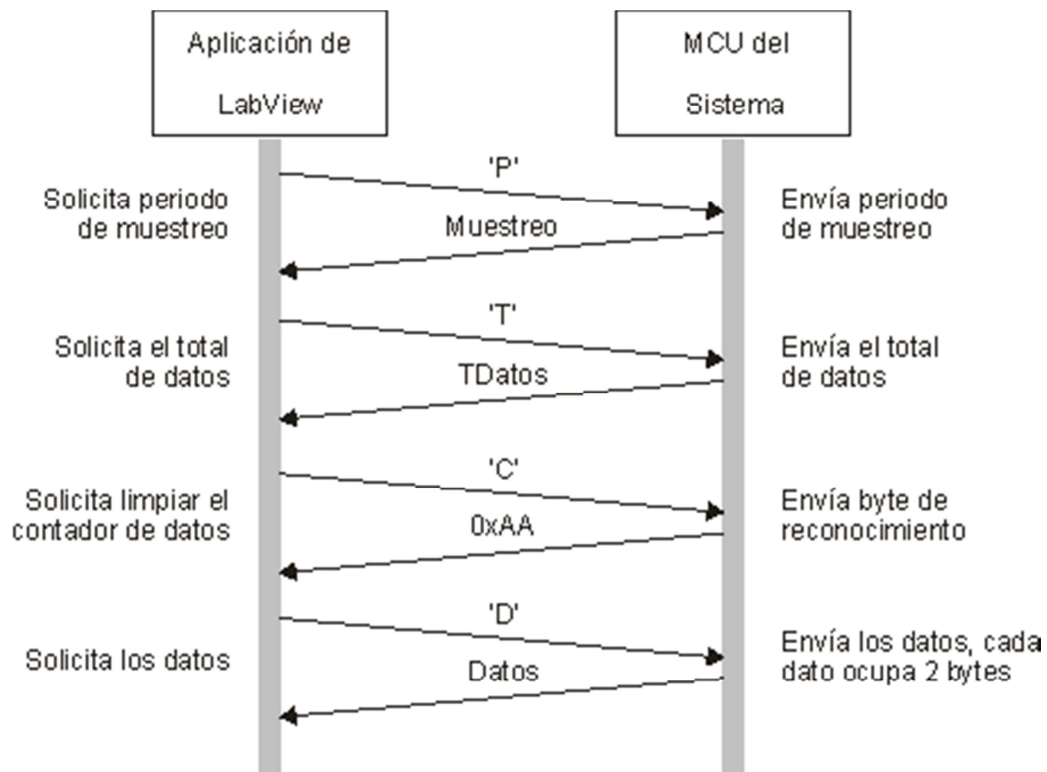


Fig. 8. Diagrama de secuencias que muestra la relación entre el sistema y una aplicación de computadora.

Los requerimientos para la aplicación serial son:

- Selección del puerto.
- Selección del comando a enviar al sistema.
- Botón para iniciar con el proceso de petición-recepción de información.
- Buffer para mostrar los datos recibidos.
- Gráfica de resultados.
- Botón para almacenar la información en un archivo.

Con estos requerimientos, se crea la interfaz mostrada en la figura 9, se agrega un botón para escalar la gráfica resultante conforme al periodo de muestreo seleccionado en el sistema.

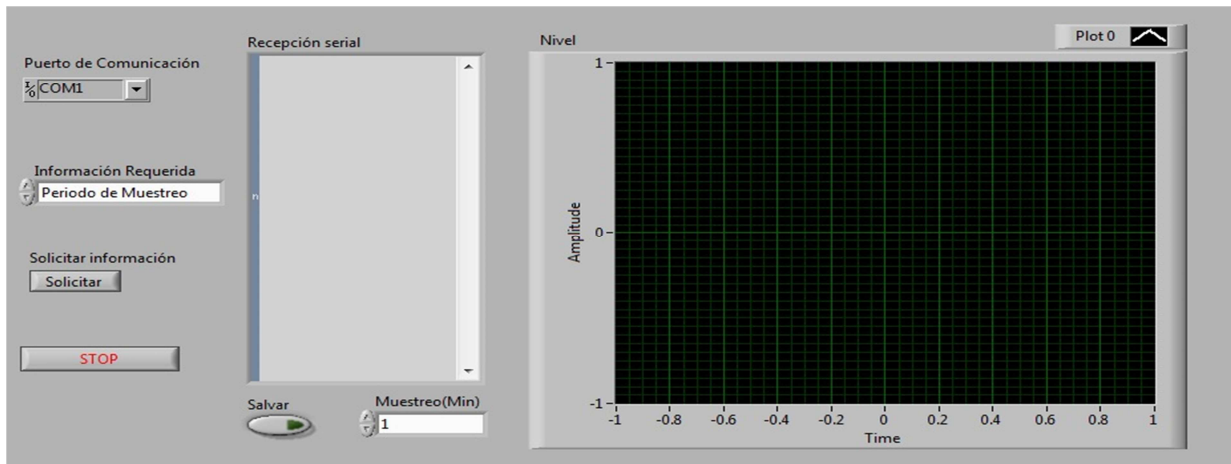


Fig. 9. Interfaz de la aplicación en LabVIEW.

El diagrama a bloques de la aplicación se muestra en la figura 10, en donde puede verse el flujo del programa. La inicialización del puerto se realiza sólo una vez y dentro de un ciclo infinito se pedirán los datos cuando se presione un botón, previamente se debió seleccionar el tipo de información. Se introduce un lapso de espera antes de realizar las lecturas. Las lecturas se realizan cuando hay datos disponibles y el resultado se muestra como una cadena para después generar su gráfica.

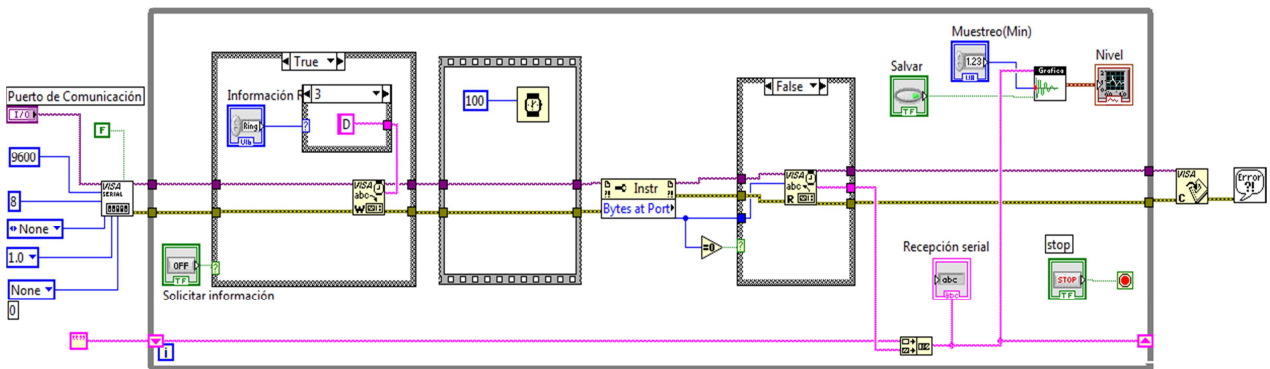


Fig. 10. Diagrama a bloques de la aplicación en LabVIEW.

El proceso se desarrollará de manera continua hasta que el botón de paro (STOP) sea presionado. El botón Salvar es para solicitar el respaldo de los datos en un archivo de texto.

4. Resultados

El programa en el microcontrolador ocupó los recursos mostrados en la tabla 1, puede notarse que la memoria de programa está muy próxima a agotarse, para ampliar la funcionalidad del sistema sería necesario migrar a un dispositivo con mayores prestaciones.

Memoria	Tamaño	Uso	Porcentaje
Programa (Flash)	8192 bytes	7586 bytes	92.6 %
Datos (SRAM)	1024 bytes	650 bytes	63.5 %
Datos (EEPROM)	512 bytes	3 bytes	0.6

Tabla 1. Porcentaje de uso de la memoria en el microcontrolador.

En la figura 11 se muestran algunas pantallas con el circuito en operación, en (a) se presenta la pantalla de bienvenida, en (b) el periodo de muestreo y el total de muestras, en (c) el modo de configuración, en (d) la toma de una muestra y en (e) el tiempo transcurrido en segundos entre la toma de una muestra y otra.



Fig. 11. Diversas pantallas del sistema en operación, (a) bienvenida, (b) periodo de muestreo y total de muestras, (c) configuración, (d) toma de una muestra y (e) tiempo transcurrido en segundos.

La figura 12 muestra al limnógrafo electrónico instalado en la estación de aforo de la UTM, se alimentó con una batería de 7.8 Volts para una operación independiente, por lo que fue necesario agregar un regulador de 5 volts.

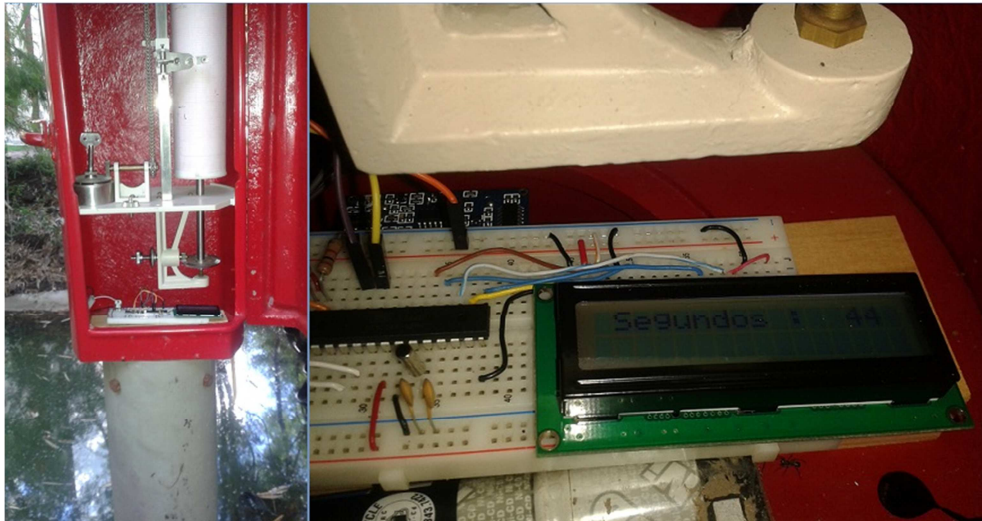


Fig. 12. Puesta en operación del Limnógrafo Electrónico.

El sistema se dejó operando por más de 15 horas con un periodo de muestreo de 1 minuto, en total se capturaron 913 datos, en la figura 13 se muestra la pantalla con la aplicación de LabVIEW en ejecución, se observa que la distancia desde la ubicación del sensor hasta el nivel del agua estuvo alrededor de 1700 mm.

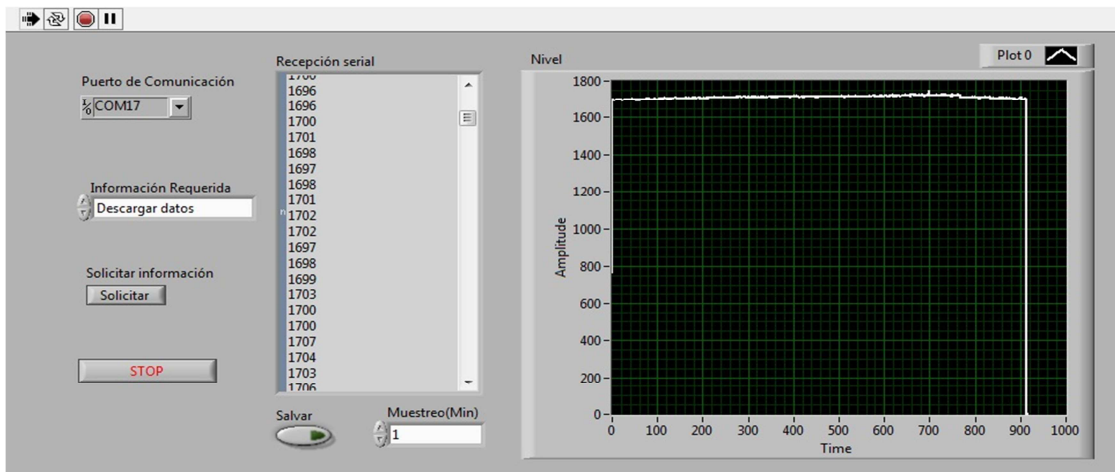
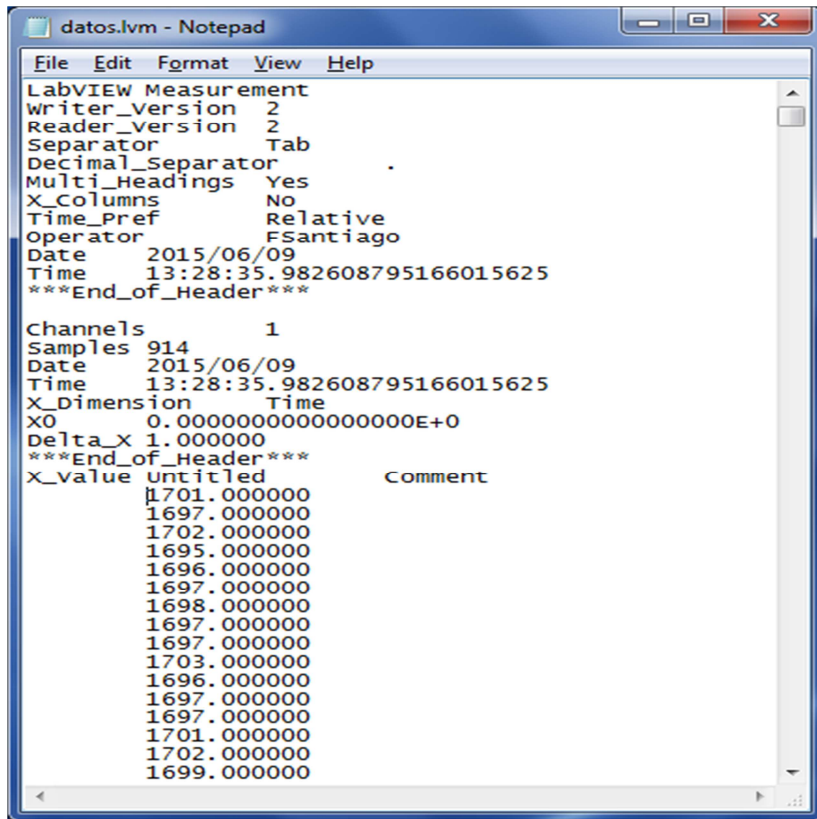


Fig. 13. Programa en LabVIEW mostrando los datos del sistema.

La figura 14 muestra parte del archivo generado por LabVIEW, los datos están dados en milímetros y la información inicial es agregada automáticamente por la aplicación.



```
datos.lvm - Notepad
File Edit Format View Help
LabVIEW Measurement
Writer_Version 2
Reader_Version 2
Separator Tab
Decimal_Separator .
Multi_Headings Yes
X_Columns NO
Time_Pref Relative
Operator FSantiago
Date 2015/06/09
Time 13:28:35.982608795166015625
***End_of_Header***

Channels 1
Samples 914
Date 2015/06/09
Time 13:28:35.982608795166015625
X_dimension Time
x0 0.0000000000000000E+0
Delta_x 1.000000
***End_of_Header***
x_value untitled Comment
1701.000000
1697.000000
1702.000000
1695.000000
1696.000000
1697.000000
1698.000000
1697.000000
1697.000000
1703.000000
1696.000000
1697.000000
1697.000000
1701.000000
1702.000000
1699.000000
```

Fig. 14. Muestras consecutivas.

5. Conclusiones

Con el limnógrafo electrónico terminado y puesto en operación se tienen las siguientes conclusiones:

El sistema funcionó favorablemente durante un tiempo razonable (15 horas en forma continua), aún con eso es necesario operarlo por más tiempo para validar su operación y detectar posibles errores, después de ello ya se podrá realizar el circuito impreso y el acondicionamiento para su colocación en la estación de aforo.

El sistema proporciona directamente el resultado del caudal, con lo cual se evita una tarea laboriosa de interpretación y análisis de datos, de esta forma se está automatizando un proceso que en la UTM aún se hace en forma manual.

Por la capacidad de la memoria EEPROM es posible almacenar la información de un número considerable de días (22 con un periodo de 1 minuto), pero como esta memoria se maneja con la interfaz I2C, en el mismo bus se pueden agregar más memorias con ajustes mínimos en el programa. Con ello el sistema podrá operar independientemente durante más tiempo, lo que resulta muy conveniente para sitios de difícil acceso, un buen complemento sería un panel solar junto con una batería recargable.

Como trabajo inmediato se planea el desarrollo de un circuito de acondicionamiento para la alimentación del sistema desde un panel solar y posteriormente realizar el circuito impreso. También se considera el desarrollo de un programa propietario en Java, para no depender del uso de LabView.

6. Referencias

- [1] D. F. Campos-Aranda, Procesos del ciclo hidrológico. Universidad de San Luis Potosí. México. 1998.
- [2] F. J. Aparicio-Mijares, Fundamentos de hidrología de superficie. 1997. Editorial Limusa, México.
- [3] S. Vidal, A. Solé, R. Lázaro, A. Duran, J. Puigdefábregas, “Estaciones automáticas de aforo y muestreo de caudales, de bajo costo y mantenimiento, para zonas remotas”. In Proceedings of the IV Simposio sobre el Agua en Andalucía. Vol. 1. 1996. 193-203 pp.
- [4] S. Tamari, A. Aguilar, V. Mejía, A. López, R. Álvarez, S. Ortega, Instrumentación de la estación hidrométrica Los Aldamas. Nuevo León. Anuario IMTA 2005.

- [5] F. H. F. Morales, “Desarrollo de un caudalímetro digital para la medición de caudal en ríos”. *Revista de Investigación, Desarrollo e Innovación*. Vol. 3. No. 1. 2013.
- [6] ATmega8: Avr RISC Microcontroller, Datasheet, 2006 Atmel Corporation.
- [7] Los Microcontroladores AVR de Atmel. Editado e Impreso por la Universidad Tecnológica de la Mixteca. México. 2012.
- [8] Ultrasonic Ranging Module HC - SR04, ELECFreaks, www.Electfreaks.com.
- [9] 24AA512/24LC512/24FC512, 512K I2C™ Serial EEPROM. Microchip Technology Inc. 2010.
- [10] LabView User Manual. National Instruments Corporation. January 1998.

7. Autores

M. C. Felipe Santiago Espinosa es Maestro en Ciencias con especialidad en Electrónica por parte del INAOE, incorporado al IEM de la Universidad Tecnológica de la Mixteca, en donde es Profesor-Investigador desde 1998. Actualmente está cursando el Doctorado en Robótica en la misma institución. En el año de 2012 publicó su libro titulado “Los Microcontroladores AVR de ATMEL”.

Dra. Gabriela Álvarez Olgún obtuvo el grado de Doctora en Ingeniería por la Universidad Nacional Autónoma de México. Actualmente es profesor-investigador del Instituto de Ingeniería de la Universidad Tecnológica de la Mixteca.

Dr. Fermín Hugo Ramírez Leyva obtuvo el doctorado en Ingeniería Mecatrónica en la Universidad Popular Autónoma del Estado de Puebla y la maestría en electrónica por el Instituto Nacional de Astrofísica Óptica y Electrónica. Actualmente es profesor investigador de la UTM, adscrito al Instituto de Electrónica y Mecatrónica desde 1999.

Desarrollo de un frecuencímetro de conteo recíproco con un FPGA para sensores basados en resonador de cuarzo

José Lorenzo Muñoz Mata

Universidad Tecnológica de Puebla, División de Mecatrónica, Antiguo camino a la resurrección 1002-A,
Zona Industrial Oriente, Teléfono: 52 01 (222) 309 8833
jose.munoz@utpuebla.edu.mx

Severino Muñoz Aguirre

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico-Matemáticas, Av. San Claudio
y 18 sur, Ciudad Universitaria

Juan Jesús Jiménez Arellano

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico-Matemáticas, Av. San Claudio
y 18 sur, Ciudad Universitaria

Georgina Beltrán Pérez

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico-Matemáticas, Av. San Claudio
y 18 sur, Ciudad Universitaria

Juan Castillo Mixcóatl

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico-Matemáticas, Av. San Claudio
y 18 sur, Ciudad Universitaria

Takamichi Nakamoto

Tokyo Institute of Technology, Precision & Intelligence Laboratory, Suzukakedai Campus, Nagatsuta,
Yokohama, Japan

Resumen

Los sensores basados en resonadores de cuarzo son ampliamente utilizados en dispositivos de detección de olores conocidos como narices electrónicas (NE). Para medir la respuesta de este tipo de sensores es esencial el uso de un frecuencímetro. Existen diferentes métodos para implementar un frecuencímetro, tal como el método de conteo directo de frecuencia, el cual está restringido en la generación de datos, ya que si se desea un incremento en el número de muestras por segundo, implicaría una pérdida en la resolución del instrumento debido a su principio de operación. Por otro lado, el método de conteo recíproco de frecuencia es un método que ofrece una alternativa en la medición de respuesta de sensores de gas. Por lo tanto en este trabajo se presenta el desarrollo de un frecuencímetro aplicando el método de conteo recíproco de frecuencia para sensores de gas utilizando un FPGA. Se obtuvieron resultados utilizando como patrón de entrada un generador de funciones. Este frecuencímetro tiene la ventaja de tener una mayor tasa de muestreo y resolución que un frecuencímetro comercial. Finalmente se realizaron experimentos utilizando sensores de gas QCM a temperatura ambiente aplicando muestras de etanol para obtener datos reales y comprobar la efectividad del sistema obteniendo resultados satisfactorios, dado que los datos arrojados coinciden con el comportamiento típico de este tipo de sensores.

Palabra(s) Clave(s): frecuencímetro, FPGA, nariz electrónica, sensores de gas QCM.

1. Introducción

Los sensores de gas de microbalanza de cristal de cuarzo (QCM) son utilizados para el desarrollo de una nariz electrónica (NE) debido a su alta correlación con la nariz humana [1]. Estos dispositivos pueden medir diferentes variables, pero en el caso de la nariz electrónica estos son configurados como dispositivos sensores de cambio de masa [2]. Estos son de gran importancia en el campo científico debido a la necesidad de trabajar en la detección, discriminación y reconocimiento de patrones de olores [3].

Además, al aplicar diferentes tipos de películas sensibles en la superficie se podrá obtener diferentes características de cada sensor logrando obtener evaluaciones más acertadas de gases o vapores [4].

El principio de funcionamiento del sensor se lleva a cabo por la adsorción de moléculas en la película sensible [5]. Al incrementarse la masa, provoca un decremento en la frecuencia de resonancia por el efecto de la carga de masa [6], esta variación es proporcional a la cantidad de masa que interactúa con la capa o película sensible y se conoce como la respuesta del sensor [7].

En la Ecuación de Sauerbrey (1) se describe el cambio de la frecuencia de resonancia en la película sensible en función del cambio de masa por unidad de área en la superficie del electrodo del QCM [8].

$$\Delta f = -2.3 \times 10^{-6} \cdot F^2 \frac{\Delta m}{A} \quad (1)$$

donde Δf es la variación de la frecuencia (Hz), -2.3×10^{-6} es una constante determinada por la densidad (ρ_q) y el módulo de cizalla o shear (μ_q) del cuarzo ($m/(Hz \cdot g)$), Δm es la masa adsorbida (g), A es el área del electrodo (cm^2) y F es la frecuencia de resonancia del cristal (Hz) [9].

Por lo tanto es esencial el uso de un frecuencímetro para medir la respuesta de este tipo de sensores. Existen frecuencímetros comerciales que solo manejan uno o a lo más dos entradas, para el caso particular de la nariz electrónica es necesario un número mayor de entradas [10]. Además, los precios de estos dispositivos son muy elevados y en algunos casos con rangos de GHz lo cual no es necesario para esta aplicación. En cambio, en algunos dispositivos al aumentar la frecuencia de la señal de entrada, estos pueden perder resolución. Por otro lado, los FPGA ofrecen prestaciones muy atractivas para el desarrollo de este tipo de dispositivos, tanto en velocidad como

en características de operación. Por lo tanto, es posible agregar más canales de entrada para medir en paralelo y mantener una buena resolución del instrumento.

En este trabajo se muestra el desarrollo de un frecuencímetro de conteo recíproco para sensores de gas basados en resonador de cuarzo. Para el cual se utilizó un FPGA para su implementación. La transmisión de datos se realizó utilizando una tarjeta (USB-DAQ), utilizando un Microcontrolador PIC18F4550 [11], la cual funge como interfaz de comunicación entre el FPGA y la computadora a través del módulo de comunicación USB (Universal Serial Bus). Se realizaron pruebas utilizando como patrón un generador de funciones. Además se realizó una comparación del sistema desarrollado con un frecuencímetro comercial para comprobar el buen funcionamiento del sistema. Finalmente se realizaron mediciones utilizando sensores de gas reales aplicando estímulos de etanol.

2. Desarrollo

Debido al principio de funcionamiento del sensor QCM, el uso de un frecuencímetro es esencial para su instrumentación y obtención de la respuesta. Dadas las prestaciones que ofrecen los FPGA' s en el desarrollo de diferentes circuitos, se utilizó una tarjeta de desarrollo EDA-004 que contiene un FPGA (Cyclone III-Altera) [12]. Para comprobar el desempeño del sistema se utilizó un generador de funciones (AFG3102, Tektronix). Para la adquisición de datos se utilizó un microcontrolador vía USB. Para el desarrollo de software se utilizó LabVIEW (ver Fig.1).

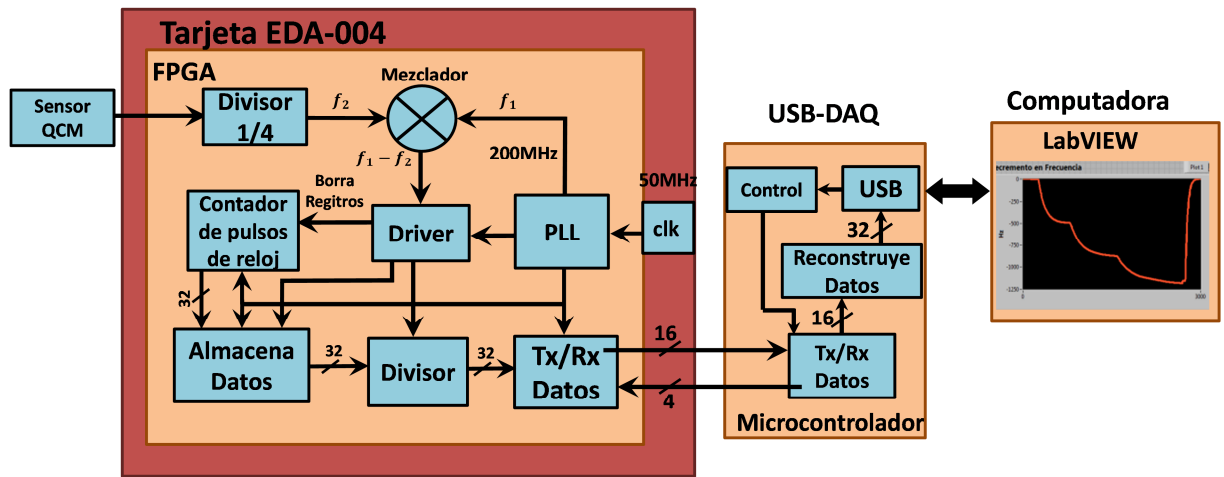


Fig. 1. Diagrama a bloques del sistema implementado.

El método de conteo de frecuencia directa es el más frecuentemente utilizado, donde el número de pulsos de la señal de entrada dentro de una ventana de tiempo determinada es contado [10]. Sin embargo, la velocidad en la generación de datos está relacionada directamente a la resolución del sistema, debido a su principio de operación, por lo que al aumentar la velocidad en la generación de datos provocaría una pérdida en resolución.

Por otro lado, existe el método de conteo recíproco de frecuencia. Este método cuenta periodos de un reloj maestro en función de los flancos ascendentes de la señal de entrada, la cual funciona como señal de inicio y paro del contador. La Ecuación (2) describe el principio de funcionamiento.

$$Frecuencia = \frac{1}{pulsos_{mclk} \times T_{mclk}} \quad (2)$$

Por ejemplo, el contador inicia con el flanco ascendente de la señal de entrada y se detiene al siguiente flanco ascendente. Ahora el contador contiene información del periodo de la señal de entrada en múltiplos del reloj maestro (ver Fig. 2). Por lo tanto este método fue el utilizado para la implementación del frecuencímetro.

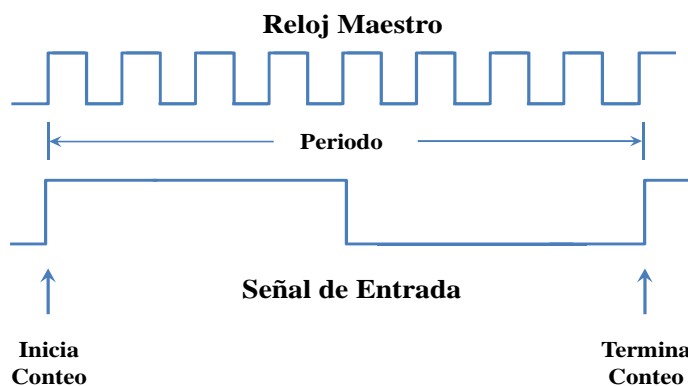


Fig. 2. Principio de funcionamiento de un frecuencímetro por el método de conteo recíproco de frecuencia.

Para esta técnica se debe dividir la frecuencia del sensor, ya que como se utilizan sensores de 12 MHz, dicha frecuencia es muy alta como para obtener buena resolución del frecuencímetro. Por lo tanto, es necesario reducir la frecuencia de entrada del sensor utilizando un circuito para dividir la frecuencia entre cuatro para obtener una frecuencia de 3 MHz. Se utiliza una señal de referencia local generada con un PLL también de 3 MHz para obtener una diferencia del orden de KHz. Estas señales son introducidas a un mezclador de frecuencia para obtener la diferencia de dichas señales (Ec. 3) y obtener una buena resolución [13].

$$f_o = f_1 - f_2 \quad (3)$$

Driver.

Se desarrolló un circuito utilizando una máquina de estados para dirigir el sistema. Una vez que se presente el flanco de subida de la señal mezclada se generó la señal para borrar los registros del conteo de pulsos de reloj maestro. En el siguiente estado, es decir, en el siguiente ciclo de reloj se genera la señal para la almacenar los datos. Posteriormente al siguiente estado se envía la señal al bloque de división para obtener el conteo recíproco.

Contador de pulsos del reloj maestro.

Se diseñó un bloque para realizar el conteo de pulsos del reloj maestro utilizando un contador síncrono ascendente, la frecuencia del reloj maestro en este caso es de 200 MHz. Los registros son borrados cada flanco ascendente de la señal mezclada.

Almacenamiento de datos.

Para el almacenamiento de datos se desarrolló un circuito de almacenamiento de datos utilizando un circuito de captura (Latch). Cada vez que se genera el flanco de subida de la señal de entrada se recibe la bandera de almacenamiento. De manera que se almacenan los datos recibidos por el contador de pulsos del reloj maestro. Estos datos son almacenados en registros de 32 bits. El software utilizado para implementar el firmware dentro del FPGA fue Quartus II.

Para el envío de datos se utilizó una estrategia de transferencia hacia la USB-DAQ. Los datos son enviados en paquetes de 16 bits a la computadora.

Desarrollo de Software.

Se implementó una interfaz basada en el software de instrumentación virtual LabVIEW, el cual permite visualizar los datos obtenidos de la respuesta del sensor, así como la diferencia del corrimiento de frecuencia (Δf) del mismo. Además es posible el almacenamiento de los datos para posteriormente poder realizar el análisis de los mismos (ver Fig. 3).

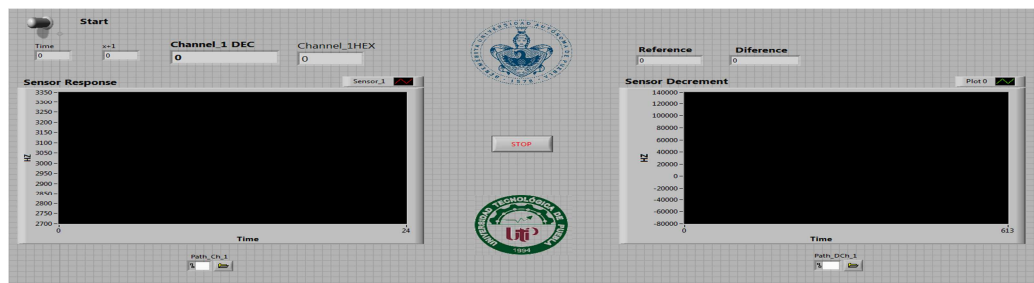


Fig. 3. Interfaz desarrollada con LabVIEW.

3. Resultados

Una vez terminado el firmware correspondiente al FPGA se realizaron las pruebas correspondientes para comprobar el funcionamiento del sistema. Por lo tanto, se utilizó un generador de funciones, un osciloscopio, la tarjeta de desarrollo EDA-004 que contiene el FPGA, la tarjeta de adquisición de datos USB-DAQ y una computadora para la visualización de la información (ver Fig. 4).

En este caso se utilizó como patrón de entrada el generador de funciones, con el cual se ajustó la frecuencia de entrada emulando la frecuencia del sensor, para observar que el sistema funciona adecuadamente.

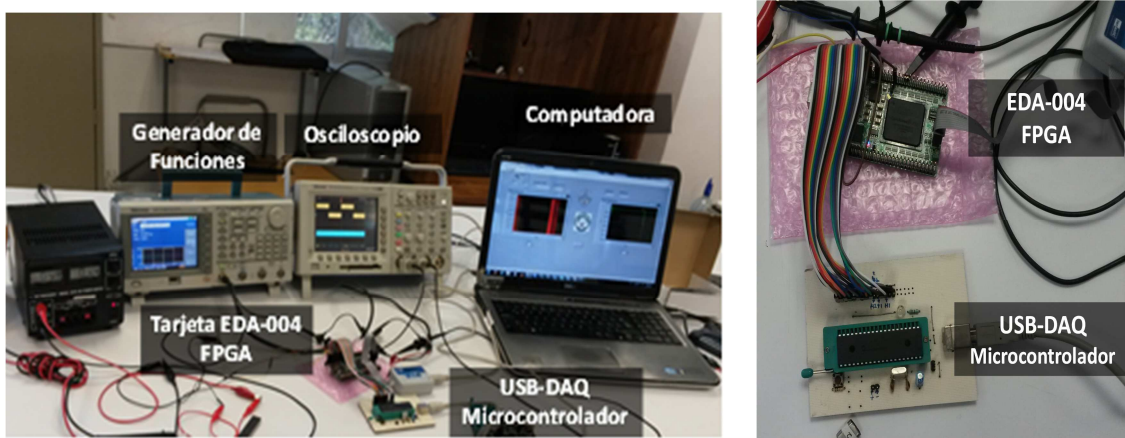


Fig. 4. Arreglo Experimental con generador de funciones.

Posteriormente se implementó un arreglo experimental, para el cual se utilizó un sensor QCM de 12 MHz, con una película sensible compuesta de etil celulosa. En este caso se utilizó el sensor a temperatura ambiente aplicando un vapor de etanol directamente al sustrato, es decir los estímulos aplicados no fueron controlados. El sensor QCM fue conectado con un circuito oscilador para excitar la frecuencia de resonancia del sensor conectado al frecuencímetro (ver Fig. 5).

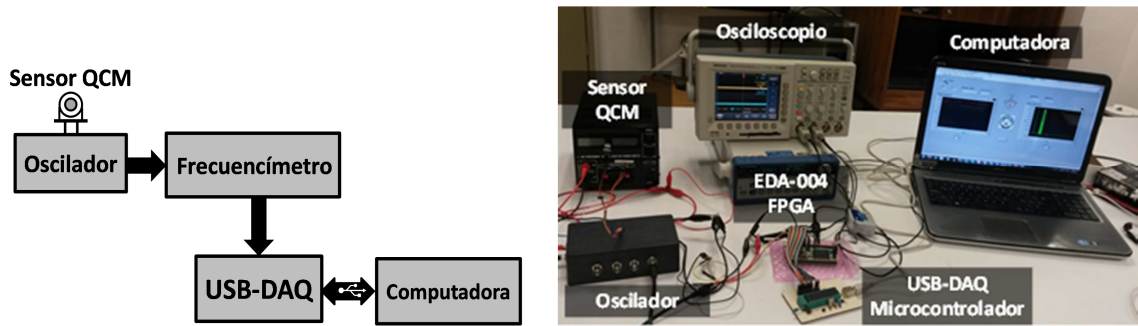


Fig. 5. Arreglo experimental utilizando sensor de gas QCM.

4. Discusión

Al integrar todo el sistema, se realizaron pruebas usando como patrón de entrada el generador de funciones, aplicando frecuencias de entrada de alrededor de 12 MHz. En este caso se aplicaron decrementos a razón de Hz y de KHz en un tiempo aproximadamente de 400 segundos por cada medición (ver Fig.6).

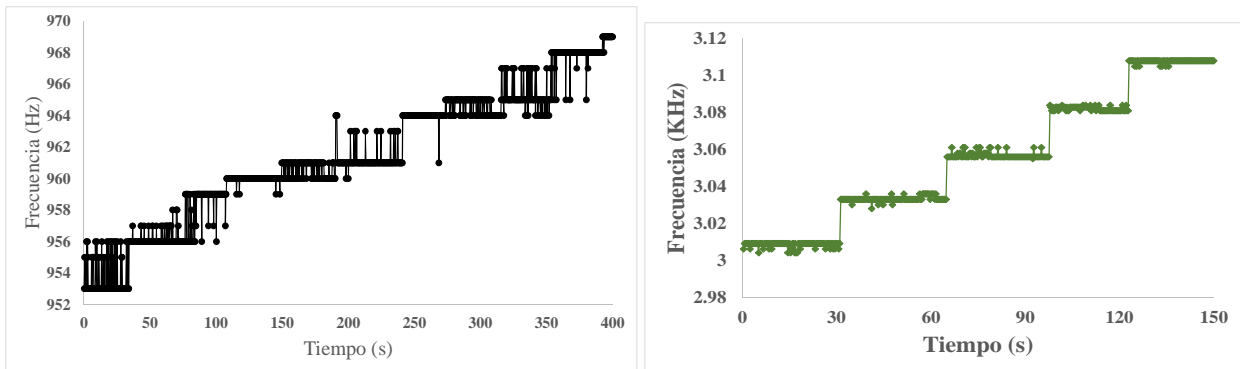


Fig. 6. Datos obtenidos utilizando generador de funciones. a) Decrementos a razón de Hz. b) Decrementos a razón de KHz.

En la Figura 6a se muestran los datos obtenidos del frecuencímetro, donde se puede apreciar que los datos enviados por el generador de funciones concuerdan con los obtenidos a través del sistema almacenados en la computadora. Como se puede observar, existe una fluctuación en los datos obtenidos, debido a las características de estabilidad del generador de funciones utilizado, por lo que se puede decir que el sistema funciona de manera correcta. En la Figura 6b se observa una estabilidad mayor

en los datos obtenidos ya que en este caso se realizan decrementos de la señal de entrada en el rango de KHz, donde se puede observar que el sistema funciona satisfactoriamente.

Hasta este punto el sistema no muestra ningún inconveniente en la dinámica de funcionamiento aunque a ciertas frecuencias se muestra una inestabilidad en el sistema debido al principio de funcionamiento del conteo recíproco de frecuencia, para lo cual es necesario realizar algunos ajustes para su optimización. Adicionalmente se realizó una prueba comparando nuestro frecuencímetro desarrollado con un frecuencímetro comercial (ver Fig. 7).

Se puede observar la alta correlación que existe entre los datos obtenidos con el frecuencímetro comercial y nuestro frecuencímetro. Por lo tanto se pudo comprobar que el instrumento diseñado funciona satisfactoriamente.

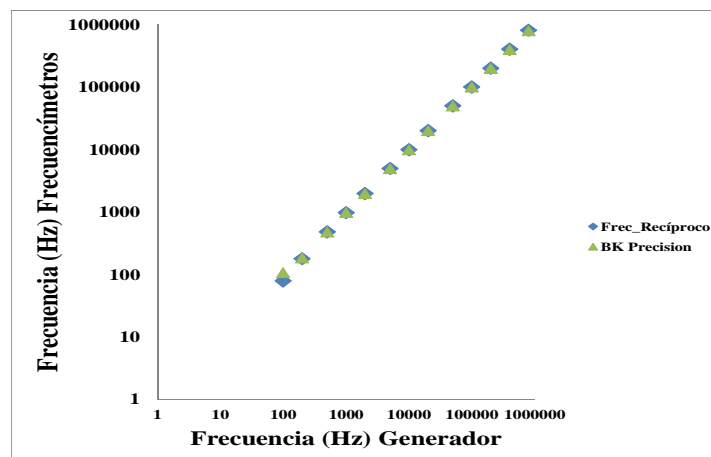


Fig. 7. Comparación entre un frecuencímetro comercial BK Precision y el frecuencímetro implementado.

Posteriormente se realizaron mediciones utilizando un sensor de gas en condiciones de temperatura ambiente, para lo cual se utilizó un sensor QCM de 12 MHz (ver Fig. 8).

Cabe destacar que al haber realizado esta prueba a temperatura ambiente las concentraciones de los estímulos o muestras aplicados no fueron controlados, de manera que solo se expuso a concentraciones de gas indeterminadas y estas pueden variar debido a las condiciones expuestas. Sin embargo, se puede observar en la Figura 8 una línea base del sensor aproximadamente de 4.05 KHz. Se aplicó el primer estímulo durante al menos un minuto, donde se puede observar corrimiento en la frecuencia del sensor (Δf) de aproximadamente 70 Hz, una vez retirado el gas se observa la recuperación del sensor a su frecuencia de resonancia base. Se aplicó un segundo estímulo durante aproximadamente un minuto, en este caso se observa un corrimiento en la frecuencia de 60 Hz.

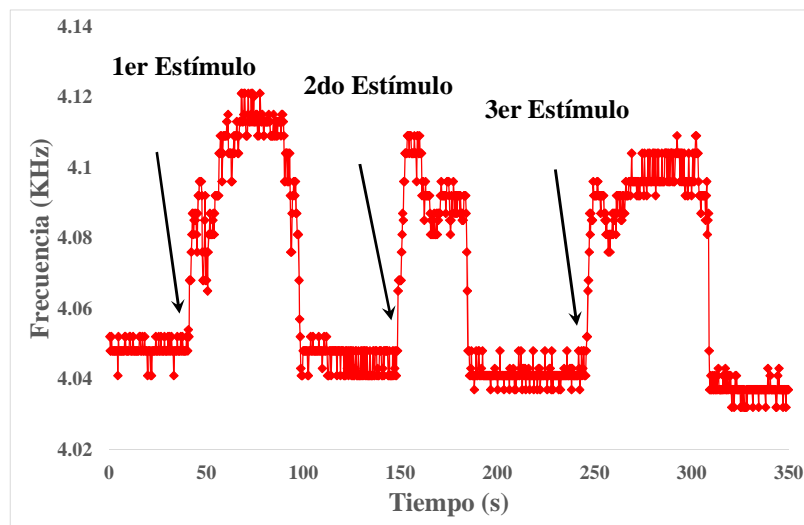


Fig. 8. Respuesta del sensor obtenida a temperatura ambiente.

Finalmente se aplicó un tercer estímulo donde se puede apreciar un corrimiento de frecuencia de aproximadamente 60 Hz y una vez retirado el estímulo el sensor se recupera a su frecuencia base. Sin embargo, cabe señalar que no se tiene control en el flujo de concentración de gas. Los datos obtenidos muestran un comportamiento típico de este tipo de sensores, por lo que se puede observar que el sistema funciona

correctamente, aunque aún se necesitan realizar un número mayor de experimentos para verificar el completo funcionamiento del sistema implementado.

5. Conclusiones

Se ha realizado el diseño de un frecuencímetro de conteo recíproco con buena resolución, utilizando un FPGA para sensores de gas QCM, así como la integración del sistema para obtener datos a través de una computadora.

Se realizaron pruebas para comprobar el funcionamiento del sistema utilizando como patrón de entrada un generador de funciones. Además se realizó una comparación con un frecuencímetro comercial y se encontró que nuestro frecuencímetro tiene mejor resolución y una mayor tasa de muestreo para un rango de 1 MHz.

Finalmente se realizaron mediciones utilizando sensores reales a temperatura ambiente obteniendo un resultado satisfactorio, ya que los resultados obtenidos concuerdan con el comportamiento típico de este tipo de sensores.

Actualmente se está trabajando en la optimización del firmware del frecuencímetro para optimizar el funcionamiento del sistema, para finalmente lograr escalar el sistema a un número mayor de canales de entrada y realizar un proceso adecuado enfocado a la caracterización de sensores QCM.

6. Referencias

- [1] H. Troy Nagle & R. Gutiérrez-Osuna, S. S. Schiffman, "The How and Why of Electronic Noses". IEEE Spectrum. Vol. 35 No. 22. September 1998.
- [2] J. W. Gardner, P. N. Bartlett, "A brief history of electronic noses". Sensors and Actuators B. Vol. 18. No. 211. 1994.

- [3] J. W. Gardner, P. N. Bartlett, *Electronic Noses*. 1999. New York: Oxford University Press.
- [4] S. Muñoz, T. Nakamoto et al, "Odor Approximation of fruit flavors using a QCM odor sensing system". *Sensor and Actuators B*. No. 105. 2005. 144-149 pp.
- [5] W. H. King. Jr., "Piezoelectric Sorption Detector". Analytical Research Division, Esso Research and Engineering Co., Linden N.J. Vol.36. No. 9 Aug 1964.
- [6] *Piezoelectric Transducers and Applications*. Berlin: Springer. 2004.
- [7] S. Muñoz-Aguirre, T. Nakamoto, et al. "Study of deposition of gas sensing films of quartz crystal microbalance using an ultrasonic atomizer". *Sensors and Actuators B*. No. 105. 2005. 144-149 pp.
- [8] SRS Application Notes, QCM- "Quartz Crystal Microbalance Theory and Calibration". Stanford Research Systems. <http://www.thinksrs.com/downloads/PDFs/ApplicationNotes/QCMTheoryapp.pdf>.
- [9] G. Sauerbrey, Verwendung von Schwingquarzen zur Wägung dünner Schichten und zur Mikrowägung *Z. Phys.* 1959. 155-206 pp.
- [10] J. L. Muñoz M, S. Muñoz, et al., "Development and implementation of a system to measure the response of quartz crystal resonator based gas sensor using an a field programmable gate array". *Meas. Sci. Technol.* Vol. 35 No. 5. Marzo 2012.
- [11] 28/40/44 Pin High Performance, Enhanced Flash USB Microcontrollers DS39617A. Microchip. USA. 2003.
- [12] Cyclone III USB-FPGA Board EDA-004, Ver 2.0. Humandata LTD. Japon. 2012
- [13] F. Yoshino, T. Nakamoto, "Odor Recognition Using Embedded Learning Vector Quantization Circuit. *Sensors and Materials*". Vol. 26. No. 3. 2014. 137-147 pp.

7. Autores

Dr. José Lorenzo Muñoz Mata obtuvo su título de Maestría en Ciencias de la Electrónica opción en Automatización por la BUAP México (2008), el de Doctorado en Ciencias por la BUAP Puebla México (2013). Actualmente es profesor investigador en la División de Mecatrónica de la Universidad Tecnológica de Puebla (UTP), México.

Dr. Severino Muñoz Aguirre obtuvo su título de Maestría en Ciencias en el CINVESTAV México (1995), el de Doctorado en Ciencias por el Instituto de Tecnología de Tokio, Japón (1999). Actualmente es profesor investigador en la Facultad de Ciencias Físico-Matemáticas de la BUAP. Su trabajo se ha centrado en el desarrollo de sistemas electrónicos para la detección de gas/olor, reconocimiento de patrones y discriminación. Su más reciente investigación involucra sensores optoelectrónicos y sistemas multisensoriales.

M. en C. Juan Jesús Jiménez Arellano obtuvo su título de Maestría en Ciencias de la Electrónica opción en Automatización por la Benemérita Universidad Autónoma de Puebla (BUAP) México (2008). Actualmente realiza sus estudios de doctorado en la Facultad de Ciencias Físico-Matemáticas de la BUAP sobre el tema de las narices electrónicas y su capacidad de sensibilidad.

Dra. Georgina Beltrán Pérez obtuvo su título de Maestría en Ciencias en el CICESE Ensenada Baja California México (1998), el de Doctorado en Ciencias por el INAOE Puebla México (2002). Actualmente es profesora investigadora en la Facultad de Ciencias Físico-Matemáticas de la BUAP. Su más reciente investigación incluye sensores de fibra óptica, rejillas de fibra de periodo largo y láser de fibra óptica.

Dr. Juan Castillo Mixcóatl obtuvo su título de Maestría en Ciencias en el CICESE Ensenada Baja California México (1998), el de Doctorado en Ciencias por el INAOE Puebla México (2003). Actualmente es profesor investigador en la Facultad de Ciencias Físico-Matemáticas de la BUAP.

Dr. Takamichi Nakamoto obtuvo su título de maestría y doctorado en el Instituto Tecnológico de Tokyo, Japón, en 1984 y 1991, respectivamente. Actualmente se desempeña como Profesor en el Precision and Intelligence Laboratory, en el Instituto Tecnológico de Tokyo, en el tema de desarrollo de Narices Electrónicas y Olfactory Display.