

SISTEMAS OPEN HARDWARE Y OPEN SOURCE APLICADOS A LA ENSEÑANZA DE LA ELECTRÓNICA

OPEN HARDWARE AND OPEN SOURCE SYSTEMS USED FOR THE TEACHING OF ELECTRONICS

Ramiro Arango¹, Álvaro Andrés Navarro², José Bestier Padilla B.³

¹. Programa de Tecnología en Instrumentación Electrónica de la Universidad del Quindío e integrante del GIDET. (e-mail: ramy@uniquindio.edu.co).

². Ingeniero Electrónico, candidato a Magister en Robótica y Automatización de la Universidad Técnica de Dortmund (Alemania), e integrante del GIDET. (e-mail: alvaro.navarro@tu-dortmund.de)

³. Programa de Tecnología en Instrumentación Electrónica de la Universidad del Quindío y Líder del GIDET. (e-mail: jbpadilla@uniquindio.edu.co).

Recibido: 17 de febrero de 2014

Aceptado: 20 de marzo de 2014

Correspondencia del autor: ramy@uniquindio.edu.co

RESUMEN

Los sistemas open source y open hardware son una opción económica, eficiente y flexible para implementar y desarrollar tecnología en el aula. El uso de estas herramientas permite la construcción de sistemas complejos haciendo uso de software y hardware diseñado por terceros, ofreciendo un gran flexibilidad gracias a la disponibilidad del código y circuitos esquemáticos; de esta manera se logra ajustar el comportamiento de las aplicaciones obteniendo algunos beneficios como: reducción de tiempo de desarrollo, expansión de aplicaciones, modularidad y corrección de problemas de una manera práctica. Adicionalmente, estos sistemas son de extrema utilidad para el caso en el cual sea necesario reducir costos, muy común en ambientes educativos de nuestra región que cuentan con recursos muy limitados. La calidad de las herramientas para el desarrollo de las aplicaciones es muy buena y no tienen nada que envidiarle a equivalentes comerciales. Con estas aplicaciones se pretende también involucrar a los estudiantes en proyectos a nivel de trabajos de fin de materia y/o proyectos de grado en la cultura del open Source tanto de software como de Hardware.

Particularmente, en el GIDET se ha venido desarrollando aplicaciones con microcontroladores PIC's y ATMEL, para robótica y otras áreas apoyados en el IDEA (Integrated Development Enviroment for coding Algorithms), un software de ambiente gráfico desarrollado en la región, el cual permite la construcción de programas a partir de algoritmos diseñados en diagramas de flujo y/o lenguajes de alto nivel (C, C++), con el IDEA además de diseñar la aplicación al finalizar es posible descargar el programa ejecutable en sistemas basados en microcontroladores. Los sistemas con los que se han realizado las aplicaciones open hardware utilizan el mismo sistema de conexión de los módulos arduino, lo que permite la conexión simultánea de varias tarjetas para la construcción de prototipos con cierta complejidad. Por su diseño basado en arduino, se puedan utilizar con ellos los módulos que ofrecen otras empresas que utilizan la misma tecnología. Los módulos son de arquitectura abierta. En el IDEA es posible contar simultáneamente con las rutinas y planos del hardware requeridos para la construcción de la aplicación. A partir de proyectos de investigación y de aula se vienen desarrollando aplicaciones con tecnologías inalámbricas basados igualmente en sistemas open hardware.

PALABRAS CLAVES: Open source, Open Hardware, sistemas microcontrolados

ABSTRACT

The open source and open hardware systems are an economical, efficient and flexible tool to implement and develop technology in the classroom. These tools allow us the construction of complex systems by using hardware and software designed by others, providing great flexibility thanks to the availability of the code and schematic circuits, thus achieve adjust the behavior of some applications obtaining benefits: reduced development time, expanding applications, modularity and correcting problems in a practical way. Additionally, these systems are extremely useful for the case in which it is necessary to reduce costs, common in educational environments of our region with very limited resources. The quality of tools for application of development is very good and have nothing to envy to commercial equivalents. It also aims to engage students either related in projects to matter but also graduation projects in the culture of open Source both software and hardware .

Particularly, the GIDET has been developing applications with PIC and ATMEL microcontrollers, robotics and other flat on the IDEA areas (Integrated Development Environment for coding Algorithms) software graphical environment developed in the region, which allows the construction of programs based on algorithms designed in flowcharts and / or high-level languages (C, C++) with the IDEA in addition to designing the application at the end you can download the executable program in microcontroller-based systems. The systems that have made the open hardware applications use the same system connecting the arduino modules, allowing simultaneous connection of multiple cards for prototyping with some complexity. On the arduino based on design, they can be used with modules offered by other companies using the same technology. Modules are open architecture. In the IDEA may benefit both routines and hardware drawings required to build the application. From research and classroom applications are being developed with equally open hardware systems based on wireless technologies.

KEYWORDS: Open source, Open Hardware, Systems microcontrolled.

I. INTRODUCCIÓN

De acuerdo con la organización de los estados iberoamericanos [1] (OEI), “la formación en tecnología e informática es todavía incipiente, a pesar de ser un área obligatoria del currículo de la educación media y aunque uno de sus objetivos es fomentar y facilitar las relaciones entre ciencia, tecnología y sociedad”. Por otro lado, la formación en informática en algunos casos se reduce a convertir o adecuar salas con conexión a internet, pero sin una capacitación a los docentes para preparar sus planes de estudio y sobre todo sus prácticas pedagógicas, entonces la formación que obtienen los estudiantes es limitada al manejo del navegador o paquetes ofimáticos como Word, Excel, Publisher, entre otros. En electrónica básica, la que se enseña en los colegios técnicos de Armenia - Quindío, el panorama es muy similar, aunque el problema radica en la deficiencia de las competencias definidas, de acuerdo con lo fijado en los planes educativos institucionales (PEI) [2] con respecto a las asignaturas técnicas ligadas por la ley. El contenido de estos cursos

debería incluir por lo menos conceptos de programación, pues como anota Douglas Rushkoff en su libro *Los mandamientos de la programación*, “Programa o serás programado”; las tecnologías (la programación y el acceso a las redes de información) son cada vez, un papel más importante en la creación de un nuevo mundo y su funcionamiento[3].

Si bien no existe un plan curricular establecido, en muchos de los colegios donde se orientan modalidades tecnológicas, por iniciativa propia, implementan diversos cursos de programación e inclusive se dictan principios de robótica. Para orientar estos cursos en ciertos colegios o universidades utilizan herramientas de programación de bajo nivel como lenguaje ensamblador, o nivel medio como lenguaje C. Sin embargo la sintaxis o estructura de estos lenguajes se convierten en algún momento en barreras que desmotivan a los estudiantes, si le sumamos la parte electrónica, el montaje de cada práctica o aplicación donde se considere algún nivel de complejidad, por lo que se requiere de habilidades para el diseño, siendo otro

motivo de frustración para algunos estudiantes o personas implicadas.

Por otro lado el rápido desarrollo de las tecnologías de información, la masificación del internet y las redes sociales ha incrementado notablemente el desarrollo de aplicaciones open source permitiendo diseñar, actualizar, mejorar y dar soluciones casi en tiempo real a problemas muy particulares.

Actualmente en el programa de instrumentación electrónica de la Universidad del Quindío, se ha venido implementando en el curso de microcontroladores esta iniciativa de llevar todas las prácticas por intermedio de módulos didácticos compatible con la tecnología Arduino donde el estudiante pone en práctica cada uno de los temas vistos en clase sumando la posibilidad que puedan conectar diversos módulos para aplicaciones más complejas tanto en hardware como en software.

En el presente artículo además de dar una visión de lo que significa open source, open software y open hardware damos a conocer algunos desarrollos realizados desde el grupo de investigación GIDET y a partir de los cuales queremos aportar a la comunidad open source, estos desarrollos han sido un complemento en el proceso enseñanza aprendizaje en la electrónica.

II. MARCO TEÓRICO

A. Proyectos Open-Source

Open source es un término que se originó en el contexto del desarrollo de software para referirse a la tecnología cuyo código es “abierto”, es decir, que cualquier persona puede usarla, modificarla o mejorarla.

En un esfuerzo de colaboración desinteresado, donde programadores y diseñadores van mejorando el código de un programa y comparten los cambios con todo el mundo. Esta filosofía surgió a principios de los años 90 en la comunidad tecnológica como respuesta al software patentado por las empresas. En Finlandia, Linus Torvalds, un estudiante de ciencias computacionales insatisfecho con el sistema operativo “Minix”, implementaba las primeras versiones del núcleo de Linux: muy pronto equipos enteros completaron el sistema operativo hasta hacerlo estable. A partir de ahí se desencadenó un torbellino creativo y en 1998, con el anuncio de la liberación de Netscape, se creó la Open Source Initiative, un grupo cuyo objetivo es educar y abogar por la superioridad de un proceso

de desarrollo abierto y que registró el término Open Source y empezó a promocionar esta tecnología [4].

Sin embargo, cuando hablamos de open source también nos referimos a un conjunto de valores, lo que algunos llaman el “estilo open source”: todos los proyectos, productos o iniciativas que apuestan por el intercambio abierto, la participación colaborativa, la transparencia y el desarrollo comunitario pueden considerarse open source. Algunos ejemplos de opens source: Open Office FireFox – ThunderBird.

B. Open software

Si bien existen diversas definiciones para Open software, básicamente significa que los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Algunos autores lo denominan «software libre» (no significa software gratis), el cual tiene un contexto de libertad no de precio, de hecho existe la posibilidad de cobrar por el desarrollo y por la distribución de Software Libre, y existen muchos casos y modelos de negocio relacionados. "Software libre" no significa "no comercial". Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre ha venido en aumento.

En la práctica, "Software Libre" y "código abierto" significan, expresan, cosas diferentes. Hay quien denomina al Software Libre como "código abierto" o "software de fuentes abiertas" ("open source" software). Aunque se puedan referir a programas similares, la idea que proyecta el término "abierto" no incluye los valores y derechos relacionados con la libertad. La palabra "Libre" se refiere a esa libertad y por lo tanto es un término más adecuado [5].

C. Open Hardware

Como se vio el open software básicamente ofrece al usuario cuatro libertades: libertad de uso, de estudio y modificación, de distribución, y de redistribución de las mejoras. De hecho existen licencias que las garantizan y que dan una cobertura legal, como por ejemplo la GPL [6].

El open hardware emplea la misma filosofía para aplicarlas en su campo. Es una propuesta casi tan antigua como la del open software, sin embargo su empleo

no es tan directo. Compartir diseños hardware es más complicado. Richard Stallman afirma que las ideas del software libre se pueden aplicar a los ficheros necesarios para su diseño y especificación (esquemas, PCB, etc), pero no al circuito físico en sí. Según su naturaleza los desarrollos basados en open hardware se puede clasificar en:

Hardware estático: Como bien se conoce los componentes tradicionales de un diseño de hardware son el circuito esquemático, el circuito impreso, información de diseño y documentación asociada. El producto final de los archivos de diseño da como resultado un circuito de existencia física (se puede tocar). Esta y otras diferencias con el software hacen que surgan una serie de problemas si se desean utilizar los mismos conceptos y licencias que para el software.

Algunos de los desarrollos de mediana y alta complejidad de diseño considerados open hardware son: BeagleBoard - Neo FreeRunner - FPGALibre - S2proto - Arduino - Ledtoy - Iearobotics - Simputer - The Handy Board - LART - Linux Advanced Radio Terminal - The Balloon Project - OpenHardware - Itsy - TARJETA JPS-XPC84: Entrenadora para FPGA - TARJETA CT6811.

En el presente artículo ampliaremos la información de uno de ellos, en vista de que es en el cual se han basado nuestros desarrollos: el ARDUINO.

ARDUINO: es una plataforma electrónica basada en el principio del hardware libre, no propietario, que permite su programación para el fin deseado. Realiza, entre otras, las funciones de comunicar e interaccionar con dispositivos y/o sistemas electrónicos (sensores, actuadores, sistemas de comunicación, etc). Arduino es libre y público y además le ampara un amplio catálogo de herramientas de programación también de tipo “Open Software”, más una importante comunidad de usuarios en la que se intercambian grandes volúmenes de información compartiendo sus proyectos y desarrollos [7][8]. En la figura 1 se ilustra una de las placas que ofrece la comunidad Arduino, sin embargo existen un sin número de aplicaciones y desarrollos con esta filosofía, los cuales pueden ser consultados en [9].

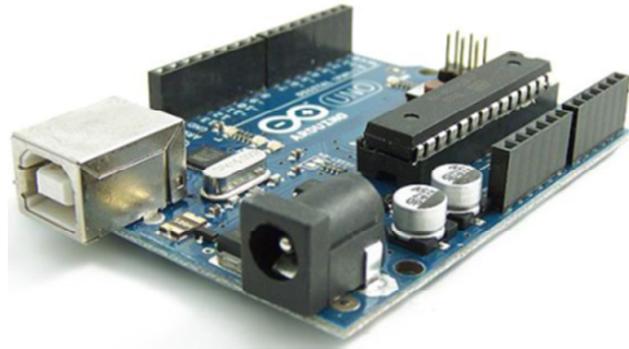


Figura 1. Arduino UNO

Hardware reconfigurable:

Es el que se describe mediante un lenguaje HDL, (Hardware Description Language, lenguaje de descripción hardware) y que permite especificar con todo detalle su estructura y funcionalidad. A partir de este código se generan unos ficheros de configuración (bitstreams) para que los dispositivos del tipo FPGA (Field Programmable Gate Arrays, matriz de compuertas programables) se reconfiguren, funcionando según lo descrito. También pueden fabricarse ASICs (Circuitos Integrado para Aplicaciones Específicas) con estos diseños, es decir, fabricar circuitos integrados hecho a la medida para un uso en particular, en vez de ser concebido para propósitos de uso general, los cuales son usados para una función específica.

Alguno de los proyectos considerados open hardware y que utilizan hardware configurable son: LEON y LEON2 processor - Opencores - The Free-IP Project - Free Model Foundry (FMF) - The Freedom CPU Project - Silicore Corporation - MPGA - Meta Programmable Gate Array - FPGALibre.

III. DESARROLLOS OPEN SOURCE - OPEN HARDWARE

Desde el grupo de investigación GIDET se han liderado algunos proyectos utilizando como apoyo proyectos open source a nivel de software y de hardware, los cuales se espera que en un futuro sean punto de partida para generar proyectos propios de este tipo.

A. DESARROLLOS BASADOS EN INICIATIVAS OPEN SOURCE

IDEA

El software Idea es un sistema que pretende servir

de apoyo a la enseñanza de la programación a nivel básico, pero sobre todo a nivel de aplicaciones con microcontrolador. A nivel de programación se pretende que el estudiante inicialmente solo utilice diagramas de flujo para programar el sistema y así omitir todos los detalles de la sintaxis asociado a un lenguaje de alto nivel, pero en la medida en que el estudiante avanza puede llegar a implementar toda una aplicación en lenguaje C. A nivel de hardware se pretende utilizar el mismo entorno para desarrollar aplicaciones para diferentes familias de microcontroladores; para lograr este propósito el sistema cuenta con archivos de configuración que definen las herramientas de compilación y enlace por cada plataforma que se utilice, también se incorporan librerías con rutinas de acceso al hardware que simplifican su acceso de manera similar a las rutinas de soporte Open Source del sistema Arduino, con la particularidad de que puede utilizarse con cualquier familia de microcontroladores para la cual se configure el sistema (claro que por cada familia se deben crear las respectivas librerías).

Luego de verificar la validez del diagrama se invoca a un traductor de los diagramas de flujo al lenguaje C, permitiendo ver en el mismo entorno el diagrama y el programa generado, con lo cual el estudiante puede relacionar bloques del diagrama con las respectivas sentencias del lenguaje C; una vez se tiene el código en lenguaje C se puede invocar los programas compiladores y enlazadores respectivos a la familia para la que se esté desarrollando la aplicación y finalmente se puede descargar la aplicación ya sea en un sistema de desarrollo embebido con el respectivo bootloader o en un quemador de microcontroladores (también se configura dentro del proyecto).

La interfaz con el usuario del programa IDEA se muestra en pantalla (Figura 2), en la parte izquierda de la pantalla se visualizan los componentes del proyecto tanto a nivel de hardware (microcontrolador, tarjetas de I/O, etc.) como a nivel de software (archivos en lenguaje c, librerías, árbol de funciones, variables globales, etc.) y en la parte derecha los diagramas de flujo de cada función implementada, y si el usuario lo requiere el respectivo código en lenguaje C generado. En la parte inferior se muestra la salida de los programas de soporte (traductor, compilador, enlazador o bootloader) que se invocan durante el proceso de creación de la aplicación. El programa se construye arrastrando los bloques de la parte superior y soltándolos en el diagrama. Las estructuras condicionales

y los ciclos se construyen de forma tal que el usuario debe proporcionar todos los elementos, minimizando la posibilidad de cometer errores.

Las pruebas iniciales con el Idea se han realizado exitosamente con compiladores basados en desarrollos Open Source del compilador GCC, adaptados a la familia 18xxx de Microchip (el compilador C18), y el compilador AVR-GCC, para la familia ATmega de Atmel y algunos ensayos con el software de soporte de Arduino.

A nivel de hardware y gracias a la arquitectura abierta del proyecto Open Source Arduino, se implementó una tarjeta de desarrollo con características similares a la tarjeta Arduino UNO pero utilizando el microcontrolador PIC 18F8550 de la familia Microchip, también gracias a que esta empresa igualmente proporciona bajo los principios del Open Source los códigos fuente para la implementación del Bootloader tanto a nivel del firmware del microcontrolador como del software del PC, más adelante se detallará esta y algunas tarjetas de propósito específico que se pueden agregar a este sistema (o al sistema Arduino), que han sido elaboradas por estudiantes con sus respectivas librerías.

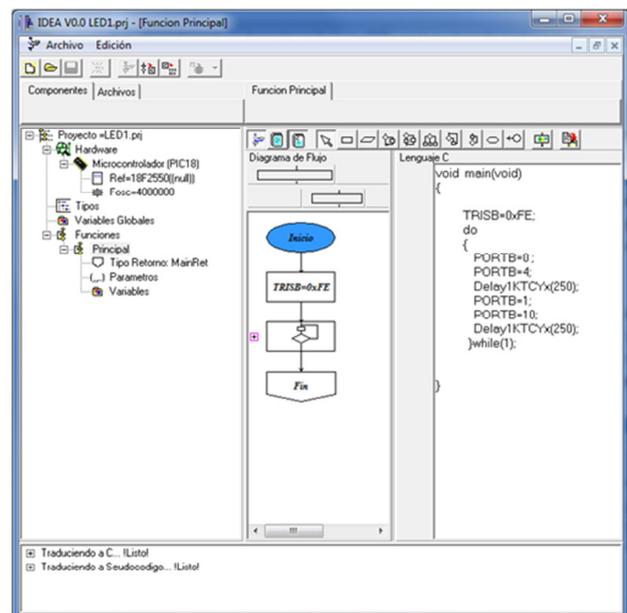


Figura 2. Interfaz del usuario del programa IDEA SISTEMA PARA LA MEDICIÓN DE VARIABLES FÍSICAS

Es un sistema conformado por cuatro módulos basados en la filosofía Arduino[9] lo cual confiere la prin-

principal ventaja del sistema, considerando criterios como la expansión y/o inclusión de diferentes sensores o aplicaciones sin tener que hacer modificaciones en el hardware, permite potencializar el análisis simultáneo de variables físicas, optimizar el desarrollo de prácticas, disminuir el tiempo empleado y los errores ocasionados por conectores (cables), entre otros.

Otro criterio, es que posee especificaciones eléctricas estables que permiten el desarrollo de nuevas aplicaciones que complementan el sistema, encajando en lo que se denomina hardware abierto.

En la figura 3 se observa el esquema del sistema desarrollado. Si bien el sistema permite monitorear y caracterizar las variables de peso, aceleración, temperatura y humedad, gracias a su filosofía arduino se pueden ampliar el número de variables sin modificar el hardware existente, esto es posible solo mediante el desarrollo y la inclusión de nuevas librerías previamente diseñadas.

MÓDULO PRINCIPAL

Este módulo contiene el micro controlador Pic18f4550 de la empresa Microchip, que tiene programado en su firmware un sistema bootloader utilizando el puerto USB. Desde el programa Idea se invoca la aplicación que descarga el código binario correspondiente al programa que se está diseñando y luego se puede ejecutar directamente, sin necesidad de desconectarlo del PC. Para el diseño de aplicaciones la tarjeta cuenta con un bus de expansión compatible pin a pin con el sistema Arduino y por medio del cual se puede conectar con los otros módulos de propósito específico. Igualmente se puede utilizar el sistema utilizando otras herramientas de desarrollo como el sistema MPLab de Microchip y/o utilizar la misma tarjeta con otras tarjetas comerciales compatibles con el sistema Arduino.

MÓDULO TEMPERATURA

Este módulo cuenta con dos sensores lm35, a los cuales se les hace una etapa de amplificación para simplificar el manejo de conversión de unidades de voltios a grados centígrados, antes de conectarla a dos entradas análogas. A nivel de software se implementó la función AnalogRead para hacerlo compatible con el sistema Arduino y se crearon otras funciones propias para la conversión de unidades de voltios a grados.

MÓDULO DE VISUALIZACIÓN

Este módulo permite visualizar en displays siete segmentos cualquier cantidad hasta de 4 cifras, las rutinas para su manejo están incorporadas como librería en el Idea.

MÓDULO ACELEROMETRO CON XBEE

Este módulo tiene un sensor de aceleración en el cual al estar alimentado, varia en sus 3 ejes dependiendo de cómo se manipule, el XBEE se utilizó como elemento adicional permitiendo enviarle la información al pc de forma inalámbrica y ver detalladamente el comportamiento de este sensor en Matlab mediante una gráfica, además para este módulo se realizó una librería y se compilo en el software idea, el cual le permite al usuario seleccionar que variable desea ver el comportamiento si es la de aceleración o temperatura si es el caso.

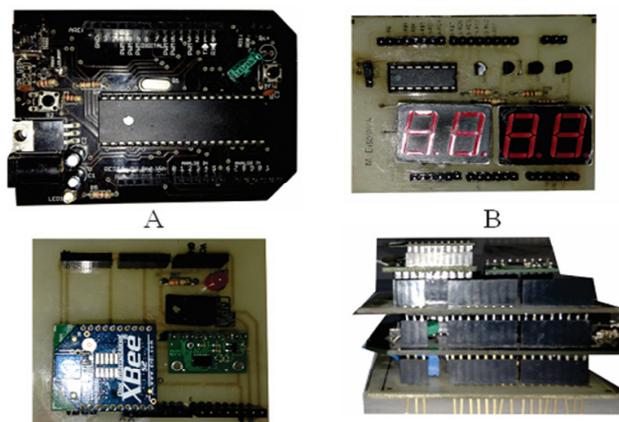


Figura 3. Módulos Hardware construidos por los estudiantes compatibles con el sistema Arduino. A) Tarjeta principal. B) Tarjeta de visualización. C) Tarjeta de sensores de aceleración y transmisión inalámbrica. D) Sistema de interconexión que permite el manejo de varias tarjetas simultáneamente.

PLATAFORMA PARA EL DESARROLLO DE APLICACIONES INALÁMBRICAS

Este sistema permitir que estudiantes y profesionales del campo de la Electrónica y afines puedan llevar a cabo la implementación de desarrollos y aplicaciones basadas en los estándar Bluetooth y Zigbee, adicionalmente el sistema permite implementar aplicaciones USB complementando y facilitando la comunicación y programación del mismo, con este módulo se pretende iniciar investigaciones en estas tecnologías

buscando el desarrollo de aplicaciones en el campo de las comunicaciones y la electrónica. En la figura 4 se muestra el esquema general del módulo desarrollado. El manejo de los estándares inalámbricos sumando la propuesta de la comunicación USB se encuentran en una misma tarjeta de aplicación separados por jumper para decidir que módulo utilizar.

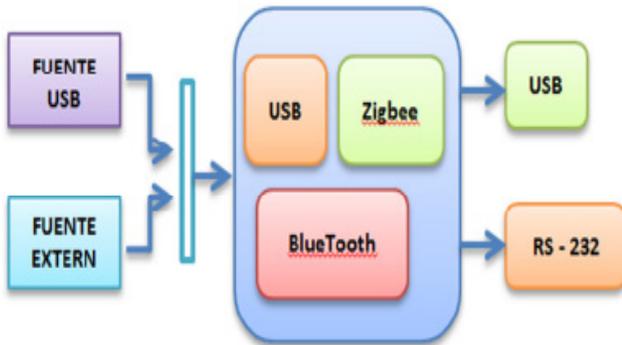


Figura 4. Diagrama de bloques sistema de aplicaciones inalámbricas

El módulo desarrollado le permite al aprendiz tener una mayor flexibilidad en el uso de cualquiera de los estándares BlueTooth y Zigbee así como en el uso del protocolo de comunicación USB para llevar a cabo diversas aplicaciones con diversas configuraciones para usar hardware con microcontrolador o hardware emulador de puerto serial. El sistema tiene la posibilidad de elegir entre utilizar una fuente externa o la fuente que proviene desde el conector USB para aplicaciones que no demanden mucho consumo de corriente y volver el manejo de la tarjeta mucho más portable. Ya sea que se utilice o no la tarjeta con el microcontrolador, se puede acceder a los pines de E/S del microcontrolador y/o de los módulos inalámbricos para utilizarlos como se requiera en una aplicación específica y de esta forma probar rápidamente los circuitos sin la necesidad de tener implementado el circuito final.

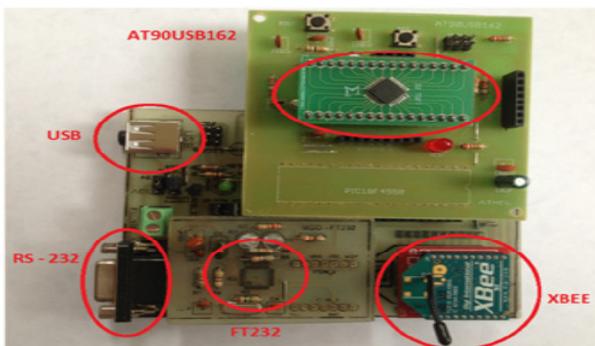


Figura 5. Sistema de desarrollo para el manejo del estándar inalámbrico BlueTooth, Zigbee (xbee) y USB

El diseño de la tarjeta permite utilizar bien sea el microcontrolador PIC18F4550 de la empresa de Microchip o el AT90USB162 de la empresa Atmel, esto con el fin de darle la posibilidad al usuario de aprender a interactuar con firmware USB de diferentes familias de microcontroladores para el desarrollo de aplicaciones de entrada y salida de información. Tanto para cuando se usa el microcontrolador de Microchip como el de Atmel se utilizan librerías Open Source que permiten realizar las operaciones de bajo nivel asociadas con el estándar USB, lo que permite disminuir el tiempo de desarrollo al solo tener que implementar las tareas propias de la aplicación. En la figura 5 se ilustra la tarjeta diseñada.

RESULTADOS Y DISCUSIÓN

Los desarrollos e implementaciones que se han realizado al interior del grupo hacen uso de proyectos open source u open hardware, en particular el IDEA en principio aprovecha los proyectos open-source (compiladores de microcontroladores basados en GCC, como AVR-GCC y C18), para invocar la compilación del código en C luego de ser traducido de diagrama de flujo. Y los sistemas open hardware desarrollados están basados en la filosofía arduino, se espera que a futuro se generen al interior del grupo proyectos Open Source.

Cabe indicar que el disponer de una plataforma Open Hardware se considera una gran ventaja para los investigadores, profesores y alumnos, porque con ella no se ven esclavizados a depender de altos costos en pago de patentes que repercuten en los productos que se compran logrando independencia y autonomía para mejoras y ampliación de aplicaciones.

Gracias al uso de herramientas open source, Silicon Valley, Seattle o Nueva York ya no son los únicos centros de innovación mundial. Cada vez son más los emprendedores de todas partes que están aprovechando la independencia tecnológica que ofrece el software libre y hardware libre para poder crear productos innovadores.

El uso demostrativo en el aula y en pequeños proyectos de los estudiantes a partir de plataformas o sistemas similares a los desarrollados por el grupo presagia resultados positivos en el aprendizaje de aplicaciones con microcontrolador con un corto tiempo dedicado a la implementación lo cual resulta

altamente gratificante, y anima a los autores a explorar su uso como herramienta de desarrollo en los cursos formales, y esperamos que este sea el comienzo de un “nuevo despertar” en la vida académica: el aprovechamiento de las ofertas que nos brinda el Open Source y el Open Hardware para la formación de los futuros diseñadores de productos.

CONCLUSIONES

A partir de la experiencia que se ha adquirido en el grupo de investigación y gracias a herramientas open source se ha podido desarrollar e implementar sistemas open software y open hardware que han permitido no sólo ampliar los conocimientos sino mejorar el proceso enseñanza aprendizaje en el campo de la electrónica, a tal punto que hemos tenido estudiantes más motivados, modificación y actualización en los currículos, incremento en el número de propuestas y de trabajos de final de carrera que en algunos casos involucran los trabajos realizados o propuestas nuevas que impactan directamente la industria o algún campo en particular (ambiental, medico, etc), igualmente nos ha permitido proponer y ejecutar proyectos de investigación que fortalecen estas metodologías de desarrollos tanto en software como en hardware.

AGRADECIMIENTOS

Agradecemos a los integrantes del Grupo de Investigación en Desarrollos Tecnológicos GIDET, adscrito al programa de Tecnología en Instrumentación Electrónica de la Universidad del Quindío y aquellos que han colaborado independientemente en los desarrollos mencionados.

BIBLIOGRAFÍA

1. SITUACIÓN DEL SECTOR DE CIENCIA Y TECNOLOGÍA ANÁLISIS DEL PERÍODO 2000 – 2005. Organización de Estados Americanos. Disponible en: <http://www.oei.es/quipu/colombia>.
2. World Data on Education. Sexta edición 2006/07. Disponible en: <http://www.oei.es/quipu/colombia>.
3. Douglas Rushkoff. Program or Be Programmed: Ten Commands for a Digital Age.
4. <http://blog.megafounder.com/es/blog/open-source-entrepreneurship/> [Febrero 15 de 2014]
5. <http://www.escuelaslibres.org.ar/2013/07/fisl14-no-existe-educacion-sin-software-libre/#more-2527> [Febrero 9 de 2013].
6. <http://www.libre.org/es/libre/software-libre/definicion-de-software-libre> [Febrero 15 de 2014]
7. Licencia pública GNU. Traducción al castellano. <http://gugs.sindominio.net/gnu-gpl/gples.html>. [Febrero 15 de 2014]
8. González Iván, González Juan, Gómez Arribas Francisco. Hardware libre: clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux. Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain, <http://www.ii.uam.es>
9. José Manuel Ruiz Gutiérrez Arduino y “El Internet de las Cosas”.
10. www.arduino.com.cc
11. <http://www.learobotics.com/personal/juan/publicaciones/art4/hardware-libre.pdf>
12. http://www.inti.gob.ar/electronicaeinformatica/instrumentacion/utic/proyectos/free_hardware/index.html
13. http://www.inti.gob.ar/electronicaeinformatica/instrumentacion/utic/proyectos/free_hardware/index.html
14. Ramiro Arango. Sistema de componentes de hardware para el diseño rápido de aplicaciones con microcontroladores. XII Seminario Taller Nacional de Instrumentación y Control y XI Muestra Tecnológica (Armenia Octubre 2.009).
15. Caterine Lipnik Porras. Diseño de software didáctico para la enseñanza de la robótica SI-RM01. Tesis.