



Artículos

Implementación de Sistemas Embebidos



Metodología Para la Transferencia Tecnológica en la Industria Electrónica Basada en Software Libre y Hardware Copyleft

Carlos I. Camargo Bareño

Universidad Nacional de Colombia, Email: cicamargoba@unal.edu.co

Abstract—Los canales tradicionales para la transferencia tecnológica en el área del diseño de sistemas embebidos no han sido exitosos en los países en vía de desarrollo donde la plataforma tecnológica no está lo suficientemente desarrollada para absorber esta nueva tecnología y conocimientos, esto se debe en gran parte a que las empresas de base tecnológica son muy pequeñas, con bajos niveles de producción y utilizan metodologías de diseño desactualizadas; por otro lado, la dependencia hacia los productos asiáticos hace muy difícil que estas pequeñas empresas puedan crecer sin la protección del estado. Así mismo, existe una sobre-oferta de profesionales afines con la industria electrónica, una gran parte de ellos provienen de entidades poco consolidadas. La unión de estos factores genera una tasa de desempleo muy alta, salarios bajos, aumento de la dependencia a los productos extranjeros y una desconfianza hacia los productos generados localmente; lo que afecta de forma considerable el número de estudiantes que ingresan a los programas de formación relacionados con la electrónica, llegando hasta el punto del cierre de programas acreditados.

Este artículo presenta una metodología para la transferencia tecnológica en el diseño de sistemas embebidos desarrollada en el Departamento de ingeniería eléctrica y electrónica de la Universidad Nacional de Colombia; esta metodología tiene como pilares el conocimiento como bien común, el movimiento de software libre y un concepto nuevo desarrollado en conjunto con un grupo de desarrolladores hardware y software: el *hardware copyleft*.

Index Terms—Sistemas Embebidos, educación en ingeniería, hardware copyleft.

I. INTRODUCCIÓN

La transferencia de tecnología ha introducido técnicas de alta productividad y en muchos casos cambios técnicos en países menos desarrollados. La adquisición de tecnología foránea contribuye a mejorar la competitividad en los mercados locales e internacionales en estos países, en los que debe ser considerada como un proceso vital. Este proceso presenta problemas cuando se pierde capacidad de absorción por parte del país receptor y la renuencia del país que transfiere a transferir tecnología real y el *know-how*. Por lo que es necesario que estos países promuevan sus capacidades tecnológicas con el fin de absorber las tecnologías foráneas de forma eficiente en función de sus necesidades locales y de esta forma generar un rápido proceso de industrialización.

La transferencia de tecnología según Van Gigch involucra la adquisición de "actividad Inventiva" por parte de usuarios secundarios. Es decir, la transferencia tecnológica no involucra necesariamente maquinaria o dispositivos físicos; el

conocimiento puede ser transferido a través de entrenamiento y educación, y puede incluir temas como manejo efectivo de procesos y cambios tecnológicos [1]. No debe confundirse la transferencia tecnológica con la apropiación de tecnología que se define como el proceso de interacción con la tecnología, la modificación de la forma como es usada y el marco social dentro del cual es usada. Un ejemplo de apropiación de tecnología lo podemos encontrar en la telefonía celular, nuestras sociedades han cambiado drásticamente su forma de comunicarse y han generado nuevas actividades alrededor de esta tecnología, los usuarios pueden generar aplicaciones que adicionan funcionalidades y servicios.

A. Tecnología

La tecnología es definida como el factor más significativo para mejorar la productividad, calidad y competitividad [2] y puede verse como un proceso de transformación que tiene como entrada recursos naturales, bienes, o productos semi-manufacturados y como salida se obtienen bienes consumibles de capital y semi-manufacturados. El *Technology Atlas team* identifica cuatro componentes de la tecnología [1]: *Techno-ware* relacionado con objetos, herramientas, equipos, máquinas, vehículos, facilidades físicas, instrumentos, dispositivos y fábricas; *Human-ware* relacionado con personas, habilidades en conocimiento experimental, sabiduría y creatividad, experiencia, competencia; *Info-ware* Relacionado con la información, incluye todo tipo de documentación y datos acumulados relacionados con especificación de procesos, procedimientos, diseños, teorías, y observaciones; *orga-ware* relacionado con la organización, acuerdos y alianzas necesarias para facilitar la integración de los componentes técnico, humano, y de información.

El uso efectivo de estos cuatro componentes requiere el cumplimiento de ciertas condiciones. El componente técnico requiere de personal con habilidades específicas para poder ser utilizado. Para mejorar la eficiencia del sistema el componente humano necesita de adaptación y motivación. A medida que la organización cambia para adaptarse a nuevas condiciones o requerimientos se debe actualizar el sector de la información. No es posible realizar operaciones de transformación ante la ausencia de uno de estos cuatro componentes. La tecnología se encuentra fuertemente relacionada con un espectro amplio de las necesidades humanas, las condiciones físicas existentes o por factores culturales derivados de las especificidades



históricas de diferentes grupos sociales [3].

B. Transferencia tecnológica

Odedra [4] define la transferencia tecnológica como el problema de transferencia de conocimiento (o know-how) sobre un número de aspectos (que incluyen el conocimiento sobre como funciona un determinado sistema, como operarlo y desarrollar sus aplicaciones, como mantenerlo y si es necesario, como producir sus componentes y implementar un sistema similar). La transferencia tecnológica se considera exitosa cuando los receptores de la tecnología asimilan los conceptos anteriores para suplir sus necesidades locales.

Según Jolly [5] La innovación tecnológica es entendida como un nuevo método, medio o capacidad del individuo para realizar una determinada actividad. El resultado de la transferencia tecnológica puede ser la aceptación de una práctica común en otros lugares, o la aplicación de una técnica diseñada para otro uso en la solución de problemas locales. La transferencia tecnológica incluye la difusión de conocimiento científico y la preocupación por la transformación del conocimiento en innovaciones útiles. El conocimiento es lo que queda al final de un proceso documentado y difundido de forma apropiada. Para que la transferencia tecnológica sea exitosa es necesario transferir los componentes de la tecnología.

1) *Tipos de Transferencia Tecnológica:* Mansfield [6] clasifica la transferencia tecnológica en *transferencia de material*: artefactos tecnológicos, materiales, productos finales, componentes, equipos; *transferencia de diseño*: diseños, proyectos, know-how para fabricar productos diseñados previamente, los productos son copiados para producirlos localmente (ingeniería inversa); *transferencia de capacidades*: proporciona know-how y software no solo para fabricar componentes existentes, sino para innovar y adaptar tecnologías existentes para generar nuevos productos. La transferencia de material no constituye una transferencia tecnológica real, ya que no genera el conocimiento necesario para transformarlos y generar nuevos productos que cumplan con las necesidades locales. La transferencia de diseños permite adquirir mayor conocimiento sobre la tecnología transferida, sin embargo, es necesario que el país receptor cuente con la plataforma tecnológica adecuada para absorber estos conocimientos, de lo contrario no se generarán nuevos productos y las actividades se limitarán al ensamblaje de productos pre-manufacturados. La transferencia de capacidades es ideal, ya que proporciona las herramientas necesarias para que la transferencia sea exitosa, está asociada a una transferencia de conocimiento, lo cual es vital para entender plenamente la tecnología, mejorando las habilidades de los profesionales del receptor, creando una demanda de bienes y servicios relacionados con el conocimiento transferido; lo que se traduce en generación de empleo y aumento del bienestar general.

C. Canales para la transferencia de tecnología

Grimpe y Hussinger [7] clasifican los mecanismos en *Formales*: acuerdos de licenciamiento, inversión extranjera,

compañías conjuntas, acuerdos de cooperación en investigación, arreglos de producción conjunta e *Informales*: No involucran acuerdos entre las partes y son difíciles de detectar y monitorear, por ejemplo, exportación de productos tecnológicos o bienes de capital, ingeniería inversa, intercambio de personal técnico y científico, conferencias de ciencia y tecnología, ferias y exposiciones, educación y entrenamiento realizado por extranjeros, visitas comerciales, literatura abierta (artículos, revistas, libros técnicos), espionaje industrial. Adicionalmente, existe una división basada en la naturaleza de la institución que proporciona los recursos para que se realice la transferencia, la institución puede ser de carácter *Abierta*: en donde la tecnología y el conocimiento son considerados bienes públicos, no existen restricciones para acceder a la información necesaria para adquirir, usar y transformar estos conocimientos en productos comerciales, y su éxito radica en obtener la máxima difusión posible para que los usuarios de este conocimiento mejoren el material existente y contribuyan a su crecimiento con experiencias personales; *Cerrada* La tecnología y el conocimiento se genera para fines privados, la utilización de este conocimiento esta sometida a acuerdos comerciales, no es posible entender las bases de la tecnología, por lo que no se pueden generar productos derivados.

Las actividades realizadas durante este estudio están enmarcadas dentro del concepto: *El conocimiento es un Bien Común*, toda la documentación necesaria para reproducir, entrenar, entender y modificar los productos generados se encuentran disponibles en servidores públicos [8] [9] y se proporciona soporte a través de listas de discusión, adicionalmente se proporciona soporte comercial para permitir la producción de estas modificaciones. A continuación se realiza una descripción de los canales más utilizados para la transferencia de tecnología y conocimiento en países en vía de desarrollo ([10] [4] [11]) indicando en cada caso sus ventajas, limitaciones y desventajas.

1) *Adquisición de IT:* La adquisición de equipo ha sido uno de los mecanismos de transferencia más importantes para los países en desarrollo. Estos equipos se entregan con el software requerido para su funcionamiento con lo que no es necesario que los usuarios generen aplicaciones, por lo que solo adquieren el conocimiento necesario para utilizar estas máquinas, y por lo tanto no saben como funcionan. Con la venta de equipos se transmite únicamente el conocimiento para operar, programar o mantener, sin embargo, este conocimiento sobre el sistema puede ayudar a concientizarse sobre la tecnología e impulsar la formación de capital humano.

La experiencia de países que lograron un rápido desarrollo económico e industrial muestra que la adquisición de una gran cantidad de tecnología foránea jugó un papel importante en este proceso. Colombia ha realizado un proceso de transformación tecnológica pero no ha diseñado políticas efectivas y eficientes para la transferencia de tecnologías de alto nivel.

2) *Educación y Entrenamiento:* Educar a las personas a través de cursos y entrenamiento en el país y enviándolas al extranjero para otros estudios es una forma de adquirir know-how sobre nuevas tecnologías. Por otro lado, muchas institu-



ciones que ofrecen carreras en ingeniería electrónica, ciencias de la computación y afines, utilizan modelos pedagógicos copiados de países desarrollados, los que no han sido adaptados plenamente a la infraestructura tecnológica local, y no es raro encontrar estudiantes que al finalizar sus estudios no están satisfechos con su profesión [4]. No se presenta una transferencia tecnológica exitosa cuando estudiantes formados en el exterior no pueden aplicar sus conocimientos en su país de origen, por lo que es necesario crear políticas que definen que áreas de estudio son prioritarias para el país.

Las multinacionales también ofrecen cursos de capacitación, sin embargo, se limitan al uso de sus productos, creando dependencia hacia sus herramientas. Adicionalmente, existen centros privados de capacitación que ofrecen cursos para el manejo de paquetes y lenguajes de programación, estos centros aprovechan la falta de centros de enseñanza tecnológica y personal calificado para cobrar altas sumas de dinero, lo cual limita el acceso a la información. Programas académicos inapropiados, acceso limitado a libros y computadores, falta de facilidades para capacitación, reduce la efectividad de la educación y capacitación como canal para la transferencia tecnológica.

3) *Asistencia Técnica:* La ventaja de contratar consultores externos radica en el ahorro de tiempo y dinero, ya que, utilizar personal local implicaría un gran esfuerzo y posiblemente se tendrían que asumir errores costosos en el proceso. Sin embargo, no es bueno confiar a consultores externos la responsabilidad de construir habilidades locales, ya que reduce el desarrollo de estas habilidades, especialmente, la del personal encargado de manejar proyectos. En algunas ocasiones los consultores no están familiarizados con las condiciones y requerimientos locales, por lo que diseñan soluciones que no se ajustan perfectamente a las necesidades, lo que significa que el sistema es sub-utilizado y la transferencia de tecnología es mínima. La falta de personal calificado hace que los consultores se encarguen de todas las tareas del proyecto, lo que aumenta su carga de trabajo y disminuye la posibilidad de entrenamiento de personal local [10].

4) *Licenciamiento:* El licenciamiento es un canal que se utiliza para transferencia de know-how sobre productos o procesos, es aplicado de forma individual o en combinación con otros instrumentos como investigación con inversión extranjera, importación de maquinaria o de técnicos. Sin embargo, esto no es efectivo si no se acompaña de habilidades administrativas y de producción. Adicionalmente, es necesario contar con una infraestructura tecnológica adecuada, capacidades locales de fabricación de hardware y software y políticas de gobierno adecuadas [11]. Un ejemplo de este tipo de práctica se presenta en el ensamblaje de dispositivos electrónicos, todos los componentes se importan completamente terminados y el dispositivo final es ensamblado probado y se carga la configuración inicial, no se producen actividades de ingeniería inversa con lo que se transmite muy poco conocimiento.

5) *Inversión Extranjera Directa:* La inversión directa de multinacionales es una forma de obtener tecnología externa. Esto asegura una rápida transferencia de información, pero

no necesariamente del entendimiento o know-how, lo que hace que la tecnología transferida a través de este canal sea mínima. Las grandes multinacionales pueden tener cierto control político en los países en vía de desarrollo, hasta tal punto que son asesores de instituciones encargadas de fijar políticas para la transferencia tecnológica [4]. El objetivo de la transferencia tecnológica debe ser el aumento de la autosuficiencia en el país receptor, unido a un uso compartido de recursos y experiencia entre los países desarrollados y en vía de desarrollo. La compra de equipo, plataformas de desarrollo o de software transfiere muy poco conocimiento sobre la tecnología, el servicio post-venta y el mantenimiento es realizado por el proveedor. Por otro lado, si las facilidades en educación y capacitación son limitadas se limita la formación de capital humano. La asistencia técnica utilizada para suplir la falta de personal especializado y ayudar con el proceso de transferencia no ha sido muy efectiva.

II. METODOLOGÍA PARA LA TRANSFERENCIA TECNOLÓGICA EN EL ÁREA DE DISEÑO DE SISTEMAS EMBEBIDOS

En esta sección se describirán los pasos de una propuesta metodológica que tiene como objetivo permitir una transferencia tecnológica exitosa en el área de diseño de sistemas embebidos. Esta metodología está compuesta por los siguientes 7 pasos y actividades:

- *Elección:* Identificación del estado de la plataforma tecnológica existente para identificar facilidades y necesidades; identificación de niveles de complejidad de la tecnología; selección de una alternativa que pueda implementarse y de resultados a mediano y corto plazo con no muy altas inversiones de capital.
- *Adquisición:* Adquisición de plataformas adecuadas de desarrollo HW y SW; identificación de herramientas de desarrollo HW y SW y su origen (abierto/cerrado).
- *Adopción:* Utilización de plataformas de desarrollo para estudio de metodologías de diseño; uso de ingeniería inversa para entender la arquitectura, funcionamiento y programación de productos comerciales; utilización de productos comerciales para adaptarlos a problemas locales (Integración); tomar conciencia de la importancia del uso de esta tecnología.
- *Absorción:* Desarrollo o adaptación de metodologías de diseño y procesos de fabricación; desarrollo de productos tecnológicos propios; enseñanza de metodologías de diseño y procesos de fabricación en centros de educación consolidados.
- *Aplicación:* Desarrollo de soluciones a problemas locales; uso de metodologías de diseño en la concepción, diseño e implementación de sistemas digitales utilizando la tecnología; utilización de procesos de fabricación adaptados al entorno local; desarrollo de proyectos académicos utilizando esta tecnología.
- *Difusión :* Vinculación de la academia para incluir los conocimientos generados en los programas académicos de las carreras relacionadas; capacitación a la industria



local sobre el uso de la tecnología, las metodologías de diseño y procesos de producción; creación de una comunidad que utilice, mejore y aumente el conocimiento generado; hacer que el conocimiento generado en los pasos anteriores este disponible a todos los interesados; dar a conocer los procesos, productos y conocimientos creados a los generadores de políticas de estado.

- **Desarrollo:** Aumento de la demanda de productos, bienes y servicios relacionados; compra de maquinaria que permita la fabricación masiva de forma local; diseño de nuevos componentes (Circuitos Integrados, IPs, software CAD); creación de políticas gubernamentales que protejan la producción local; participación activa de la academia en la solución de problemas y en la formulación de políticas de gobierno relacionadas.

A. Elección y adquisición

1) *Niveles de complejidad de la tecnología:* Existen varias alternativas para la implementación de un sistema embebido: FPGA, sistema sobre silicio (SoC), SoC + FPGA y ASIC, la utilización de FPGAs o SoCs está determinada por el cumplimiento de restricciones temporales y funcionales, mientras que el uso de ASICs depende de el número de unidades producidas, se estima que a partir de 10 mil unidades se debe utilizar un ASIC para reducir los costos de producción. Debido a que los niveles de producción de los países en vía de desarrollo no son muy grandes, se abordará el problema de la transferencia utilizando FPGAs y SoCs comerciales, sin descuidar el estudio e implementación de ASICs. Adicionalmente, la inversión necesaria para construir un circuito integrado es muy alta, y puede ser considerada como un punto final.

Un proyecto reciente promovido por la unión Europea llamado Iberchip empezó desde hace 17 años un proceso de transferencia tecnológica en el diseño circuitos integrados de aplicación específica (ASICs) hacia los países iberoamericanos; a pesar de que Colombia era uno de los nodos principales no se logró fabricar ningún circuito integrado para aplicaciones comerciales; lo que era de esperarse ya que la mayoría de empresas en Colombia son importadores y distribuidores de productos tecnológicos y carecen de Departamentos de I+D; por otro lado, las pocas empresas de base tecnológica se encuentran muy atrasadas y dependen de productos comerciales para desarrollar sus productos, esto, unido a los bajos niveles de producción hace innecesaria la fabricación de un ASIC.

2) *Diagnóstico de la Industria Local:* Para determinar el estado de la industria electrónica en Colombia, se creó la empresa **emQbit LTDA.** en asociación con profesionales en ingeniería de sistemas, ingeniería eléctrica e ingeniería electrónica. Esta empresa desarrolló una serie de proyectos y actividades que ayudaron a entender e identificar los siguientes obstáculos para el desarrollo y comercialización de sistemas digitales: Falta de proveedores de bienes y servicios relacionados con la actividad (venta de dispositivos semiconductores, fabricación de placas de circuito impreso, montaje automático de componentes, etc); desconocimiento

de la tecnología (alcances y limitaciones) debido al uso de tecnologías y metodologías de diseño obsoletas; competencia con productos asiáticos de muy bajo costo; falta de confianza en los productos nacionales; desconexión de la academia con el sistema productivo; inexistencia de reglamentación de la industria de manufactura electrónica; profesionales con pocos conocimientos en procesos de diseño y fabricación. que coincide con los resultados de estudios consultados [10] [12] [13] [14] [15] [16].

3) *Diagnóstico de la Academia:* La tendencia moderna en los programas académicos a la utilización de herramientas de alto nivel para la enseñanza en áreas afines al desarrollo de dispositivos digitales [17] ocasiona que los profesionales no adquieran las habilidades necesarias para completar la cadena concepción - diseño - implementación y operación, en la mayoría de los casos se generan habilidades para la concepción y el diseño a alto nivel y dejan los otros pasos en manos de herramientas especializadas y/o a empresas asiáticas. Esta situación resulta la más atractiva desde el punto de vista económico, ya que no es necesario adquirir maquinaria costosa ni contratar personal calificado para operarlas; sin embargo, limita la generación de empleo local a personas con un nivel de formación alto [18] generando desempleo en las personas menos capacitadas. Según John Hall presidente y CEO de Linux International “algunas facultades preparan a la gente en el uso de productos en vez de tecnologías de nivel básico” [17]. Esta situación unida al abandono de la implementación hace que la dependencia con las empresas manufactureras asiáticas aumente cada vez más.

Por otro lado, en muchas instituciones educativas se utilizan tecnologías y metodologías de diseño obsoletas (Familias 74XXX o 40XXX, lenguaje ensamblador, mapas de karnaugh), esto unido a programas académicos centrados en el análisis y no el diseño, donde el paso final es la simulación y el personal docente no tiene ninguna experiencia en el sector productivo; origina una deficiencia de habilidades necesarias para realizar el proceso completo para el diseño de dispositivos, lo que se traduce en profesionales que no disponen de las herramientas necesarias para resolver los problemas del país y al mismo tiempo competir con los productos asiáticos.

4) *Adquisición:* En la actualidad es muy fácil adquirir productos implementados con tecnología de punta, existe una gran variedad de plataformas de desarrollo y de dispositivos comerciales a los que se les puede aplicar ingeniería inversa para entender y modificar su funcionamiento. Al comienzo de este estudio se trabajó con la consola de juego GameBoy Advance, la cual posee un ASIC basado en un procesador ARM7.

B. Adopción

En el proceso de adopción se utilizaron productos desarrollados en otros países para suprir necesidades locales, en esta etapa se trabajó con las plataformas comerciales: Game Boy de Nintendo + Chamedlabs Xport (FPGA), la agenda Zaurus SL-5500, el dispositivo para redes sociales chumby y el portaretratos de Sungale ID800WT. Se utilizó la ingeniería inversa



para entender el funcionamiento y la arquitectura de un sistema embebido moderno, la metodología de diseño y como programar aplicaciones que cambien el funcionamiento de estos dispositivos. Se desarrollaron aplicaciones académicas como: Osciloscopio Digital utilizando FPAAs [19], Automatización de un Puente grúa a escala [20], Control Adaptativo Embebido [21], Control de un horno de reflujo para componentes de montaje superficial [19], Dispositivo de visualización de variables que suministra el computador de un automóvil [19], herramienta para realizar reconfiguración parcial de FPGAs [22], evolución de un arreglo de células utilizando algoritmos genéticos [23] y comerciales como: Adquisición de datos para medición de calidad de energía; plataforma robótica didáctica; registro de aceleración de vehículos para compañías de seguros; sistema de seguimiento vehicular; monitor de signos vitales; diccionario basado en Wikipedia; menú electrónico y consola de juegos.

C. Absorción y aplicación

La absorción es una actividad de aprendizaje que integra conocimiento que es nuevo para el país pero que no es nuevo para el mundo. En esta etapa se pasó de las plataformas comerciales al diseño de aplicaciones propias, para lo que se desarrollaron y/o adaptaron técnicas de fabricación al entorno local, se realizó la transferencia de los conocimientos adquiridos a la academia y se iniciaron contactos con empresas manufactureras locales y extranjeras. Se desarrollaron los productos académicos: Plataformas de desarrollo *hardware copyleft* para CPLDs, FPGAs, Procesadores ARM, MIPS, Blackfin, LM32, Microblaze, Robótica, Linux Embebido, Codiseño HW/SW [24] [25] [26], [27], [28] así como la creación de un programa académico para la enseñanza de sistemas digitales utilizando tecnología de punta que crea las habilidades necesarias para concebir, diseñar, implementar y operar dispositivos digitales modernos y la definición del concepto *hardware copyleft* y su utilización como herramienta en la enseñanza de diseño de sistemas embebidos [29]. El proyecto SIE [8] proporciona la información recolectada durante la aplicación de este programa académico en las asignaturas del área de digitales en la Universidad Nacional de Colombia; una encuesta realizada a estos estudiantes muestra que ellos encuentran la metodología adecuada para generar habilidades necesarias para el desarrollo de aplicaciones comerciales, que se proporcionan las herramientas necesarias para lograr el objetivo final, y son conscientes de que se busca que ellos generen productos novedosos y de esta forma generar empleo; adicionalmente se muestra que los estudiantes no tienen ningún problema en utilizar herramientas licenciadas de forma ilegal y que prefieren el uso de las herramientas libres aunque su utilización sea mas compleja.

Adicionalmente, se desarrollaron los siguientes productos comerciales: Control de tornos industriales, plataforma robótica didáctica, monitoreo de Temperatura, sistema de seguimiento vehicular, sistema de medición de la calidad del suministro de energía eléctrica, monitor de signos vitales (UNAL), sistema de comunicación encriptada utilizando el

canal GSM (MICROENSAMBLE), switch de 4 canales de radio frecuencia (TESAMERICA).

D. Difusión

Con las etapas previas se adquirieron los conocimientos necesarios para concebir, diseñar, implementar y operar dispositivos digitales, y la experiencia necesaria para realizar la producción a grandes escalas. La etapa de difusión (se está realizando en el momento de escribir este artículo) busca hacer llegar este conocimiento a todos los interesados, y de esta forma crear una comunidad que lo utilice como un recurso común; el cual proporciona a los centros de formación un programa académico actualizado que permite generar en los estudiantes las habilidades necesarias para innovar y generar empleo y a la industria le suministra herramientas que puede utilizar para desarrollo de nuevos productos comerciales y para la capacitación de su recurso humano. Ahorando en ambos casos el tiempo necesario para la utilización efectiva de esta tecnología (valorada en 5 años) y el dinero asociado a este estudio. Esto es una consecuencia de la filosofía del movimiento FOSS y la iniciativa *hardware copyleft*.

El proceso de difusión se realiza en varios frentes, en el plano académico se realizaron conferencias y cursos de actualización donde se presentan las plataformas *hardware copyleft* desarrolladas y el programa académico de las asignaturas asociadas con el área de la electrónica digital, hasta el momento 4 de las principales universidades públicas del país están implementando este programa; en el plano industria se realizó una alianza entre el SENA, (la entidad de formación técnica más grande del país) la Universidad Nacional de Colombia, (la universidad más importante del país) Sharism at work LTDA, (empresa constituida en HONG KONG especializada en la manufactura de dispositivos digitales) empresas de base tecnológica y universidades de diferentes regiones del país; esta alianza busca difundir los conocimientos adquiridos durante 5 años de desarrollo e investigación, para ello se creará una plataforma *hardware flexible* que cumpla con las condiciones del *hardware copyleft* y que permita la implementación de aplicaciones comerciales en diferentes áreas. Durante el desarrollo de este proyecto las empresas participantes propondrán ideas para la realización de productos comerciales que den solución a problemas de su región, se vincularán estudiantes de los centros de formación que estén realizando su trabajo de grado para formar un equipo que trabaje en la construcción de un dispositivo que implemente estas ideas basado en la plataforma diseñada previamente.

El primer paso de este proceso consiste en la capacitación de las empresas y los centros de formación en la utilización de software abierto y *hardware copyleft* como herramientas de desarrollo de sistemas embebidos, para esto se utilizará el material generado en los cursos de la Universidad Nacional; a continuación se diseñará la plataforma de desarrollo flexible, este proceso será guiado por la empresa Sharism y por la Universidad Nacional, una vez finalizado el diseño, se realizarán los primeros prototipos y las pruebas correspondientes, una vez corregidos los posibles errores se procederá a la realización



de 100 unidades de la plataforma base, esto con el fin de simular una producción a *gran escala*. Se entregará a cada grupo de trabajo varias de estas unidades para que desarrollen su aplicación con la ayuda de todos los grupos de trabajo.

Todo el proceso se documentará en un servidor de libre acceso, esto con el fin de hacerlo accesible a cualquier interesado, se suministrarán los archivos necesarios para reproducir la capacitación y la plataforma de desarrollo. Cada grupo de trabajo está encargado de llevar una *bitácora* donde se detalla el proceso de diseño y de suministrar los archivos necesarios para reproducir el producto final.

III. CONCLUSIONES Y TRABAJO FUTURO

Este artículo presenta los resultados de la aplicación de una metodología que permite una transferencia tecnológica exitosa en el área de diseño de sistemas embebidos, obteniendo como resultados de este proceso la capacitación a empresas de base tecnológica en el uso de herramientas y metodologías de diseño modernas basadas en software libre y hardware copyleft; la actualización del contenido de las asignaturas relacionadas en los centros de formación participantes; la posibilidad de generación de empleo a diferentes niveles (de servicios, técnico, profesional, comercial, administrativo); la adopción y apropiación de esta tecnología por parte del sector productivo.

A futuro se debe monitorear el cluster semilla para observar la evolución de los productos comerciales generados, así como observar los efectos que generan el cambio introducido en los programas académicos, los participantes deben estar encargados de cuidar, mejorar y aumentar el recurso común, procurando difundirlo para aumentar el número de usuarios/beneficiarios.

Una vez alcance un nivel de producción adecuado, se debe dar el siguiente paso: la construcción de un ASIC que permita reducir los costos de producción, para esto se está comenzando a trabajar con el diseño de un SoC abierto, que permita su modificación, en el momento de escribir este artículo se está trabajando con el softcore Mico 32 de lattice (lm32), se están generando IPs y se está estudiando su arquitectura con el fin de mejorarla e implementarla, la siguiente etapa de este proceso es el *desarrollo* de la tecnología, proporcionando productos nuevos que no tengan ningún tipo de patente o restricción de uso. Por lo que en un futuro no muy lejano estaremos anunciando la fabricación y disponibilidad de el primer ASIC abierto diseñado en Colombia.

IV. BIBLIOGRAFÍA

REFERENCES

- [1] F. Bar, F. Pisani, and M. Weber. Mobile technology appropriation in a distant mirror: baroque infiltration, creolization and cannibalism. *Seminario sobre Desarrollo Económico, Desarrollo Social y Comunicaciones Móviles en América Latina*, Buenos Aires, April 2007.
- [2] Goel Cohen. *Technology transfer: strategic management in developing countries*. Sage Publications inc, 2004.
- [3] K. Goel and Sayers B. Modelling Global-Oriented Energy Technology Transfer to DCs. *Sixth Global Warning International Conference, San Francisco*, 1995.
- [4] M. odedra-straub. The Myths and Illusions of Technology Transfer. *IFIP World Congress Proceedings*, August 1994.
- [5] James A. Jolly. The Technology Transfer Process: Concepts, Framework and Methodology. *The Journal of Technology Transfer*. Springer, 1977.
- [6] E. Mansfield. East-West technological transfer issues and problems, international technology transfer: Forms, resource requirements, and policies. *American Economic Review*, 65(2), 1975.
- [7] C. Grimpé and K. Hussinger. Formal and Informal Technology Transfer from Academia to Industry: Complementarity Effects and Innovation Performance. *Centre for european economical research*, 2008.
- [8] C. Camargo. Proyecto SAKC. URL:<http://en.qi-hardware.com/wiki/SAKC>.
- [9] C. Camargo. ECB_AT91 y ECBOT Plataformas Abiertas para el desarrollo de Sistemas Embebidos. URL: <http://wiki.emqbit.com/free-ecb-at91>.
- [10] M. Odedra. *Information Technology Transfer to Developing Countries: Case studies from Kenya, Zambia and Zimbabwe*. PhD thesis, London School of Economics, 1990.
- [11] M. Odedra. Information Technology Transfer to Developing Countries Is it really taking place? *The 4th IFIFTC9 International Conference on Human Choice and Computers*, North Holland, Amsterdam, Netherlands, HCC 4 held jointly with the CEC FAST Programme., 1991.
- [12] Innovation Associates Inc. Technology Transfer and Commercialization Partnerships Executive Summary.
- [13] M. Duque and A. Gauthier. Formación de Inegniers para la Innovación y el Desarrollo Tecnológico en Colombia. *Revista de la Facultad de Minas - Universidad Nacional de Colombia*, December 1999.
- [14] D Zuluaga, S Campos, M Tovar, R Rodríguez, J Sánchez, A Aguilera, L Landínez, and J Medina. Informe de Vigilancia Tecnológica: Aplicaciones de la Electrónica en el Sector Agrícola. Technical report, COLCIENCIAS, 2007.
- [15] M. Tovar and R. Rodríguez. PROSPECTIVA Y VIGILANCIA TECNOLÓGICA DE LA ELECTRÓNICA EN COLOMBIA. Master's thesis, Universidad Nacional de Colombia, 2007.
- [16] Héctor Martínez. Apropiación de conocimiento en Colombia. El caso de los contratos de importación de tecnología. *Revista Cuadernos de Economía*, 2004.
- [17] Jon Hall. POR GRANDES QUE SEAN...: ASEGURE EL FUTURO DE SU NEGOCIO. *Linux magazine*, ISSN 1576-4079(58):92, 2009.
- [18] A. Grove. How America Can Create Jobs. http://www.businessweek.com/magazine/content/10_28/b4186048358596.htm, May 2010.
- [19] C. Camargo. Implementación de Sistemas Digitales Complejos Utilizando Sistemas Embebidos. *Memorias del XI Workshop de Iberchip ISBN 959-261-105-X*, 2005.
- [20] I. Castillo, C. Camargo, and C. Perez. Automatización de un puente grúa a escala, mediante una plataforma embebida la cual soporta multiprogramación. *XII Workshop Iberchip*, 2006.
- [21] F. Pedraza, F. Segura, C. Camargo, and A. Gauthier. Control Adaptativo Embebido. *Memorias del XI workshop de Iberchip ISBN 959-261-105-X*, 2005.
- [22] C. Camargo and O. Sanchez. Linux embebido como herramienta para realizar reconfiguración parcial. *XII Workshop Iberchip*, 2006.
- [23] J. Espinosa, F. Segura, and C. Camargo. Evolución de un Arreglo de Células Utilizando Algoritmos Genéticos. *Memorias del XI Workshop de Iberchip ISBN 959-261-105-X*, 2005.
- [24] C. Camargo. ECBOT y ECB_AT91 Plataformas Abiertas para el Diseño de Sistemas Embebidos y Co-Diseño HW/SW. *VIII Jornadas de Computación Reconfigurable y Aplicaciones*, Madrid España, September 2008.
- [25] C. Camargo. ECBOT: Arquitectura Abierta para Robots Móviles. *VII conferencia Iberoamericana en Sistemas, Cibernetica e Informática*, 2008.
- [26] C. Camargo. ECBOT y ECB_AT91 Plataformas Abiertas para el Diseño de Sistemas Embebidos y Co-diseño HW-SW. *VIII Jornadas de Computación Reconfigurable y Aplicaciones*, 2008.
- [27] C. Camargo. ECBOT: Arquitectura Abierta para Robots Móviles. *IEEE Colombian Workshop on Circuits and Systems*, 2007.
- [28] C. Camargo. First Colombian Linux SBC runs Debian. URL: <http://www.linuxfordevices.com/c/a/News/First-Colombian-Linux-SBC-runs-Debian/>, 2006.
- [29] C. Camargo. Hardware copyleft como Herramienta para la Enseñanza de Sistemas Embebidos. *Simposio Argentino de Sistemas Embebidos*, 2011.

Balanceadora de Muella para Máquinas Rectificadoras con Microcontrolador

Ing. Paulo Andrés Biasutto
Especialidad en Sistemas Embebidos
Instituto Universitario Aeronáutico
Córdoba, Argentina
paulo.biasutto@gmail.com

Mg. Hector Riso
Especialidad en Sistemas Embebidos
Instituto Universitario Aeronáutico
Córdoba, Argentina
hriso@iua.edu.ar

Abstract—Este trabajo representa una solución para las pequeñas industrias que utilizan máquinas rectificadoras para realizar acabados superficiales y consiste en el desarrollo de un sistema embebido basado en un microcontrolador de 32 bits que realiza mediciones de señales de vibraciones y luego calcula el desbalance que posee la muela. El resultado del cálculo es mostrado en una interfaz visual que expresa la magnitud y la posición de la masa compensadora. Dependiendo de su configuración puede realizar cálculos de desbalance en un solo plano o en dos planos.

Keywords-Analizador de vibraciones; balanceadora; máquina rectificadora; muella; desbalance.

I. INTRODUCCIÓN

Una máquina rectificadora es utilizada no solo para conseguir mecanizados de alta precisión, sino también para acabados superficiales. La calidad de la superficie se logra principalmente cuando la muella (disco abrasivo) se encuentra perfectamente balanceada [1]. El problema surge en el momento en que la muella comienza a desequilibrarse a razón del desgaste que va sufriendo en los mecanizados que realiza. El desequilibrio ocasiona la presencia de la vibración [2] que causa el defecto de los acabados superficiales.

El método principal para detectar las vibraciones producidas en este tipo de máquina es mediante analizadores de vibraciones. Estos equipos realizan un análisispectral de las frecuencias de todas las vibraciones producidas por las partes mecánicas afectadas y son utilizados para detectar fallas y realizar mantenimientos preventivos. Se pueden utilizar analizadores como el propuesto por S.A. Ansari y R. Baig [3] basado en PC y el propuesto por G. Betta [4] basado en DSP para detectar la presencia del desbalance de la muella, pero no para obtener la magnitud y la posición de la masa necesaria para equilibrarla. Para esto se requiere de procedimientos y cálculos adicionales. Una opción para balancear la muella es la utilización de máquinas balanceadoras aunque de esta manera la muella deberá ser extraída de la máquina rectificadora cada vez que se encuentre desbalanceada. Además, si la muella no es colocada perfectamente en eje del husillo de la máquina rectificadora el problema persistirá. Como solución a este problema se pueden utilizar las balanceadoras presentadas por Li C. [5][6] y montarlas directamente en la rectificadora pero

para esto, se necesitaría colocar una PC y un monitor, lo cual implicaría un costo adicional al sistema.

El sistema embebido presentado en este trabajo consiste de una balanceadora que es utilizada directamente en la máquina rectificadora. Este sistema detecta las vibraciones como cualquier analizador, selecciona únicamente la frecuencia que corresponde al giro del husillo, realiza el cálculo de la masa necesaria para equilibrar la muella y muestra el resultado en una interfaz visual del propio sistema embebido. El sistema se instala colocando uno o dos transductores, dependiendo si se quiere balancear en uno o en dos planos, un sensor de posición colocado en unos de los extremos del eje giratorio y el sistema embebido en un lugar visible para la operación de balanceo.

II. DESCRIPCIÓN DEL SISTEMA EMBEBIDO

El sistema embebido (en adelante sistema) de la balanceadora de muella de la máquina rectificadora, fue diseñado con los componentes disponibles en el mercado local. Este sistema está constituido principalmente por dos transductores de vibración, un sensor de posición, dos pulsadores, un microcontrolador, un módulo LCD gráfico, una interfaz de conexión y un circuito de alimentación de energía. La Fig. 1 muestra un esquema general del sistema.

Los transductores convierten las vibraciones en señales eléctricas que luego son muestreadas y almacenadas en la memoria del microcontrolador. La determinación de los tipos de transductores que se necesitan para el sistema se basa en la frecuencia en que gira la muella de la rectificadora donde se va a montar el sistema. En general, en las rectificadoras sin centro, como el modelo RSC-127 [7], la velocidad de giro de la muella está definida entre 1500 rpm a 1800 rpm y por lo tanto la frecuencia de trabajo para los transductores es de 25 Hz a 30

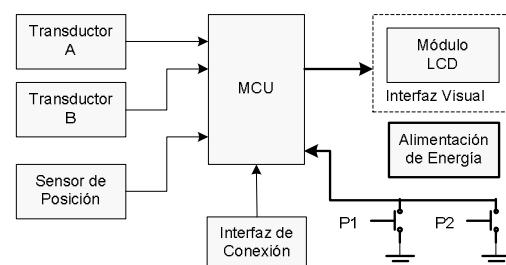


Fig. 1. Esquema general del sistema

Hz. En esta frecuencia se pueden utilizar transductores de desplazamiento, de velocidad, acelerómetros [8][10], o acelerómetros de capacidad variable con alta sensibilidad [11].

Con el sensor de posición, y a diferencia de los analizadores de vibraciones, la balanceadora conoce en qué posición se encuentra el eje y cuál es su velocidad de giro mediante el conteo del tiempo que demora en realizarlo. Esto se puede lograr utilizando un sensor óptico, cuyo circuito electrónico es simple y, con tan solo colocar una marca en el eje del husillo, se evita modificar las condiciones mecánicas de la máquina. La salida de este sensor se conecta a una entrada digital del microcontrolador que provoca una interrupción cada vez que se detecta el punto de referencia y realiza el conteo de tiempo entre vueltas.

Los pulsadores permiten seleccionar el modo de operación del sistema: estático o en un plano, dinámico o en dos planos, y calibración.

En la interfaz visual, que consiste de un módulo LCD gráfico, se muestran los resultados de las magnitudes del desbalance de la muela y las ubicaciones de las pesas compensadoras.

La interfaz de conexión se utiliza para realizar actualizaciones del firmware del sistema (como el diseño propuesto por D.G. Roca[12]) o algunas configuraciones adicionales. Debido a que no se requiere de una conexión permanente y a la posibilidad de acceder sin la necesidad de instalar algún controlador en la PC, se utiliza una interfaz RS232.

El microcontrolador realiza el procesamiento de las señales para determinar las magnitudes y las posiciones de desequilibrio y luego muestra los resultados en la interfaz visual. Se elige un microcontrolador que cuente con los siguientes requisitos:

- al menos 2 canales de conversores A/D para la lectura de transductores
- al menos 3 entradas digitales para el sensor óptico y los pulsadores
- un bus de 8 bits de datos y 5 bits de control para utilizar un módulo LCD gráfico con interfaz paralela.
- una interfaz UART para la conexión RS232
- al menos 32Kbytes de RAM para cálculo de FFT, cálculo de algoritmo, controladora de LCD y consola para la interfaz RS232

De todos los microcontroladores disponibles en el mercado, se elige trabajar con microcontroladores del fabricante Microchip Inc por estar familiarizado con los mismos, por sus bajos costos en las herramientas de desarrollos y la disponibilidad funcional completa del compilador en lenguaje C de MPLAB C. El microcontrolador que se utiliza para este sistema es el PIC32MX775F512H [13] de la familia PIC32 por su desempeño y por su costo, que está en el orden de la familia de PIC24 y dsPIC. Asimismo, utilizar este microcontrolador de 32 bits de arquitectura con capacidad de 64Kbytes de memoria RAM y 512Kbytes de memoria Flash, nos permite desarrollar un sistema con escalabilidad en futuros proyectos tales como

adicinar una interfaz Ethernet para un control del proceso de producción.

A. Firmware del Microcontrolador

El esquema general del firmware del microcontrolador está representado en la Fig. 2. Los pulsos emitidos por el sensor de referencia, que está conectado a una entrada con Notificación de Cambio (Change Notification)[14], son detectados por una rutina de interrupción que cuenta el tiempo que transcurre entre un pulso y otro utilizando como base el registro del Temporizador 1 (Timer1)[15]. Cuando se obtienen varias muestras iguales de tiempo, el eje de la máquina rectificadora se encuentra en velocidad de régimen. Luego, este tiempo es dividido por la cantidad de muestras que se desea por vuelta y, al resultado, se lo asigna al Temporizador 2:3 (Timer2:3)[15][16] para ser usado como intervalo de muestreo en el conversor A/D[16]. Si se tiene en cuenta que la máxima velocidad de giro del eje de la máquina es de 30 rps y que se obtienen 256 muestras por vuelta, el período de muestreo mínimo es $130\mu\text{s}$ por lo que no afecta a los tiempos de procesos del microcontrolador ya que el reloj del CPU opera en 80 Mhz. Además, con este método se independiza el sistema de la configuración de la frecuencia de giro.

Las muestras obtenidas por el conversor A/D, son almacenadas en un arreglo de 2048 datos para cada canal que equivale a 8 vueltas de 256 muestras. Con 8 vueltas se logra una resolución entre coeficientes espectrales de 3,5 Hz aproximadamente, pudiendo diferenciar las vibraciones generadas por otros factores tales como correa, motor o bomba hidráulica. La cantidad de 256 muestras por vuelta está determinada por la capacidad máxima de la memoria del microcontrolador. Una vez almacenadas las muestras de cada canal, se realiza la FFT y se obtienen los coeficientes espectrales de la serie discreta de Fourier[17] utilizando cada arreglo en la función *mips_fft16()* [18] provista por el compilador MPLAB C. El resultado de este cálculo es otro arreglo de vectores de números complejos en formato Q15 que representa a un espectro de frecuencias producidas por las vibraciones presentes en la máquina. Siguiendo con la idea de independizar al sistema de la velocidad de giro del husillo, se elige siempre el noveno vector o coeficiente. Dependiendo de la etapa del proceso en que se encuentra el sistema, se puede utilizar este vector para calcular el desbalance o para obtener los coeficientes de los transductores.

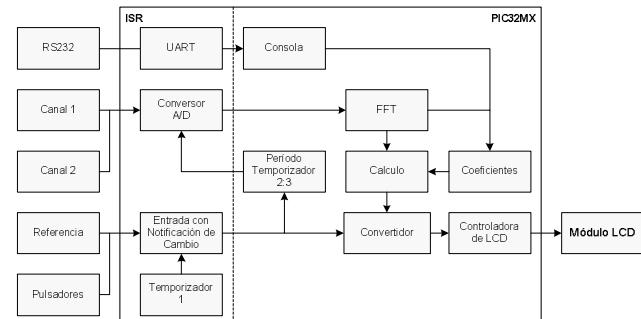


Fig. 2. Esquema general del firmware del microcontrolador

La elección del modo de cálculo dinámico depende de la estructura mecánica de la máquina y el lugar donde se colocan los transductores. Además, cuando la vibración obtenida en uno de los transductores, que se debe al desbalance de uno de los planos, es influenciado por el desbalance del otro plano, se podría utilizar el cálculo de modo dinámico [8]. Pero este método no puede ser utilizado en las rectificadoras RSC-127 o rectificadoras con una estructura mecánica similar porque no se cumple lo dicho anteriormente. A pesar de esta limitación, se implementó este modo de balanceo para que el sistema pueda ser utilizado en máquinas que poseen la mecánica adecuada.

Para conocer el valor y la posición de la masa compensadora, es necesario obtener previamente los coeficientes que determinarán las proporciones y las fases de las vibraciones leídas en la señal de los transductores. Estos coeficientes son vectores de número complejo obtenidos después de la FFT y de haber colocado una masa de prueba de peso conocido en un determinado lugar de la muela (procedimiento que se realiza durante la fase de calibración). Luego, estos coeficientes son utilizados dentro de la ecuación 1 y 2 que se muestran mas adelante.

$$M = \frac{MT \times S_A}{S_{TAI} - S_A} \quad (1)$$

Si el sistema está configurado para realizar balanceo en un solo plano y con un solo transductor utiliza la ecuación 1 [8], que mediante un cálculo vectorial entre el vector coeficiente adquirido en el proceso de calibración (S_{TAI}) con la masa de prueba (MT), y el vector de la señal adquirido en el proceso actual (S_A), se obtiene el resultado de la masa que compensará el desequilibrio (M).

En caso de que el sistema utilice dos transductores y dos planos de balanceo se procede de manera similar a lo dicho anteriormente con la diferencia de que se coloca la masa de prueba en los dos planos de manera alternada.

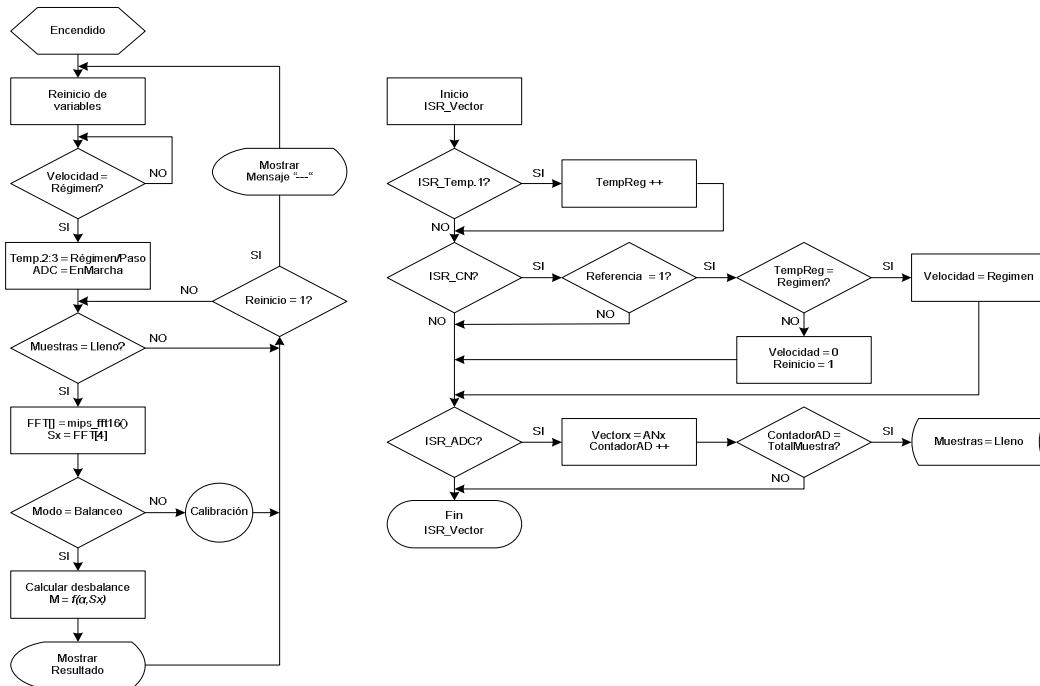


Fig. 3. Diagrama de flujo de datos del proceso de balanceo

$$\begin{Bmatrix} S_A \\ S_B \end{Bmatrix} = - \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{Bmatrix} M_D \\ M_I \end{Bmatrix} \quad (2)$$

$$\begin{aligned} \alpha_{11} &= (S_{A1} - S_{A0}) / MT_D & \alpha_{21} &= (S_{B1} - S_{B0}) / MT_D \\ \alpha_{12} &= (S_{A2} - S_{A0}) / MT_I & \alpha_{22} &= (S_{B2} - S_{B0}) / MT_I \end{aligned} \quad (3)$$

Donde (S_{A1}) y (S_{B1}) son las señales obtenidas en los transductores A y B cuando la masa de prueba (MT_I) es colocada en el lado izquierdo de la muela y, (S_{A2}) y (S_{B2}) cuando (MT_D) es colocado en el lado derecho. En las ecuaciones 3, (α) representa a los coeficientes de influencia que son utilizados luego para calcular los desbalances correspondientes. De la ecuación 2 [9], que se basa en el principio de balanceo de rotores rígidos, se despeja la matriz ($M_D M_I$) para obtener los resultados vectoriales de ambas masas compensadoras.

En ambos casos, cuando se calcula el valor de la masa compensadora, el resultado representa una proporción del valor de la masa utilizada en el momento de calibrar el sistema y es independiente a la unidad en que se trabaja.

El convertidor adapta el resultado del desbalance a la controladora del display LCD, quien luego se comunica con el módulo LCD gráfico formando la imagen correspondiente en la pantalla.

Finalmente, el sistema cuenta con una consola o terminal RS232 para el acceso desde una PC.

B. Proceso de Balanceado de Muela

Durante el funcionamiento normal de la máquina rectificadora, el sistema visualiza en forma permanente el resultado del desbalance de la muela. En la Fig. 3 se muestra, en forma simplificada, el proceso que realiza el firmware del sistema para visualizar el resultado del desbalance.



Una vez encendido el sistema y de haber reiniciado todas sus variables, el proceso principal queda en un bucle esperando que el giro de la muela haya obtenido una velocidad de régimen de trabajo mayor a cero. En el servicio de interrupciones del microcontrolador (ISR), el Temporizador 1 [15] incrementa el registro TempReg cada un tiempo definido de al menos 1000 veces menor que al de una vuelta. Cuando el eje de la muela se encuentra en la posición de referencia, se activa el ISR_CN de Notificación de Cambio (Change Notice) [14] y compara el valor del TempReg con el valor anterior y, si ambos son iguales, la velocidad está en régimen.

Una vez obtenida la velocidad régimen, el proceso principal calcula el período para el muestreo de los canales A/D escribiendo el resultado en el registro del Temporizador 2:3 (Timer2:3) [15][16] y luego activa el servicio de interrupciones del ADC. En este momento el proceso queda a la espera de que se almacene el total de muestras, mientras que el ISR_ADC captura cada una de ellas. Cuando las conversiones hayan completado el total de muestras, se procede a calcular la FFT de cada canal y obtener los vectores correspondientes a la frecuencia de giro de la muela. Cada vector es luego utilizado para obtener el desbalance y finalmente el resultado es ingresado a una función que lo muestra.

Es importante que el desbalance sea considerado cuando la muela no esté realizando el mecanizado para evitar lecturas erróneas por el contacto con la pieza o por cambios en la velocidad de giro. Si este último sucediera, el sistema se recuperaría al estado inicial desactivando el ISR_ADC y mostrando en la interfaz visual un mensaje nulo.

Cuando se detiene el giro de la muela, el sistema muestra el resultado del último proceso realizado para que la muela sea balanceada. El usuario deberá girar el eje hasta obtener una imagen “en posición” en la interfaz visual, luego colocará la masa correspondiente en la posición marcada y finalmente procederá con la operación normal de la máquina con la muela balanceada.

C. Calibración del Sistema

La calibración del sistema se debe realizar para obtener los coeficientes de los transductores de vibración con respecto a una masa conocida de prueba. En el diagrama de la Fig. 3 se indica un proceso de calibración que se activa cuando se selecciona el modo de calibración y consiste en ejecutar uno o dos pasos adicionales al proceso de balanceo dependiendo si se trabaja en uno o dos planos.

Una vez iniciado el modo de calibración, el sistema realiza el mismo proceso que se utiliza para el balanceo hasta obtener los vectores de las señales de los transductores. Para ello, antes de poner en marcha el giro de la muela, el usuario deberá colocar la masa de prueba en el plano izquierdo en la misma posición que marca la referencia del eje, encender el giro y esperar a que el sistema muestre el mensaje correspondiente en la interfaz visual. Luego el usuario procede a realizar el siguiente paso de la misma forma pero moviendo la masa de prueba al plano derecho y desfasada a 180 grados. Finalmente se extrae la masa de prueba y se enciende el giro de la muela. El sistema obtiene los vectores actuales de los transductores, calcula la matriz de coeficientes y comuta al modo de balanceo de muela mostrando el resultado en la interfaz visual.

Todas las muelas utilizadas en una máquina rectificadora utilizan las mismas bridas para sujetarse al eje del husillo, por lo tanto si se cambia de muela una vez calibrado el sistema, el sistema permanecerá calibrado debido a que las condiciones físicas para el balanceo no se modificaron.

III. EVALUACIÓN EXPERIMENTAL DEL SISTEMA

Antes de completar el diseño se hicieron evaluaciones experimentales para comprobar la funcionalidad del proceso de muestreo y la veracidad de los resultados en los procesos de cálculos. Estas evaluaciones fueron realizadas en 3 etapas:

A. Muestreo en sistema embebido y simulación de los algoritmos de cálculos

En esta etapa se utilizó el mismo sistema embebido para realizar los muestreos necesarios de las señales de los transductores. Esto consistió en implementar el diagrama de flujo de la Fig. 3 hasta la condición “Muestras = Lleno”, luego las muestras almacenadas de ambos transductores se transmitieron por RS232 a una PC. Finalmente, el sistema embebido fue colocado y puesto en marcha en la rectificadora junto con el sensor óptico, un transductor y la PC.

Se realizaron 10 capturas de 2048 muestras para las diferentes situaciones simuladas con una masa de prueba. Mediante un software implementado en JavaTM se empaquetaron en archivos con formato soportados para MATLAB® las muestras recibidas desde el sistema embebido.

Para comprobar la funcionalidad del proceso de muestreo, mediante MATLAB® y luego de implementar la Transformada de Fourier, se compararon los resultados de las 10 capturas de cada situación y se verificó la repetibilidad, dejando como consecuencia la aprobación de este proceso. Seguido a esto, se realizaron las simulaciones de los algoritmos de cálculos implementando la ecuación 1.

En la simulación del balanceo en un solo plano se utilizaron las muestras de 3 situaciones: sin masa de prueba para obtener el desbalance de la muela, masa de prueba colocada a 0° para obtener la posición y los gramos de la masa compensadora y masa de compensación de 15 gr colocada en la posición calculada para balancear la muela. El resultado del desbalance de la muela sin masa y con masa compensadora se observa en la Tabla 1.

De la misma manera se realizaron capturas de muestras sobre otra estructura mecánica que permitía obtener información con 2 transductores y con 2 planos de balanceo. En ambos casos los resultados de las simulaciones quedaron aprobados para continuar con las siguientes etapas.

B. Transformada de Fourier en sistema embebido

En esta etapa simplemente se adicionó en el firmware del sistema embebido el proceso de la Transformada de Fourier que continúa luego de la condición “Muestras = Lleno” de la Fig. 3 y, posteriormente, se transmitieron los coeficientes

TABLA I. SIMULACIÓN DE BALANCEO EN UN SOLO PLANO

Sin masa compensadora	16.2603 gr < 232.78°
Con masa compensadora 15 gr	1.5254 gr < 197.05°



TABLA II. SIMULACIÓN DE BALANCEO EN UN SOLO PLANO DESPUÉS DE APLICAR FFT

Sin masa compensadora	16.3545 gr < 235.08°
Con masa compensadora 15 gr	1.3897 gr < 224.75°

correspondientes por la interfaz RS232. Por último, los coeficientes fueron procesados y simulados de la misma manera que en la etapa anterior, dando como resultado la Tabla 2 donde el valor residual, al colocar la masa compensadora, es levemente diferente debido a la pérdida de precisión que se tuvo cuando se transformaron los datos en cadena de caracteres para transmitirlos a la PC. Este proceso quedó aprobado.

C. Implementación de algoritmos de cálculos en sistema embebido

En esta otra etapa fueron implementados los algoritmos simulados anteriormente dentro del sistema embebido. La comprobación de la veracidad de los resultados se llevó a cabo insertando en el código del firmware los mismos coeficientes de la FFT anteriormente calculados, luego el sistema realizó el cálculo correspondiente y se compararon los resultados dejando como consecuencia valores similares. Esto se muestra en la Tabla 3.

IV. CONCLUSIONES

Con el desarrollo de este sistema embebido se logró automatizar el balanceado de la muela directamente en la máquina minimizando las vibraciones presentes en la superficie de la pieza rectificada durante el proceso de mecanizado. De esta manera se resolvió una necesidad concreta para las pequeñas y medianas industrias, teniendo en cuenta las limitaciones del entorno local.

Luego de evaluar las diferentes alternativas que existen se observó que para realizar un balanceo mediante analizadores de vibraciones se requiere de conocimientos adicionales, tales como el significado de cada componente de frecuencia o el uso de cálculos vectoriales para determinar el peso de la masa que se colocará en la muela. Asimismo, si se desea utilizar alguna máquina balanceadora, se necesita extraer la muela desde la máquina rectificadora, balancearla y nuevamente colocarla para continuar con el trabajo, aunque este procedimiento demandaría más tiempo en la producción. Por lo tanto, se puede observar que el sistema embebido presentado en este trabajo permite que su uso sea sencillo y que, una vez calibrado, el operario de la máquina solo necesite colocar la pesa en el lugar adecuado. Por otra parte, con el ejemplo descrito en la evaluación experimental, se pudo determinar la veracidad de los resultados tanto en lo analítico como en lo práctico, es así como este trabajo alcanza la solución al problema planteado.

Una característica importante que posee este sistema

TABLA III. IMPLEMENTACIÓN DE ALGORITMOS DE CÁLCULOS EN SISTEMA EMBEBIDO

	Resultados en Simulación	Resultados en Firmware
Sin masa compensadora	- 9.3608 - 13.4106i	- 9.3587 - 13.4106i
	16.3545 gr < 235.08°	16.3534 gr < 236°
Con masa compens. 15 gr	- 0.9869 - 0.9784i	- 0.9867 - 0.9784i
	1.3897 gr < 224.75°	1.3896 gr < 225°

embebido es que puede ser utilizado en otras máquinas rotativas que requieran balancear un disco, una rueda u otra máquina con una mecánica similar con tan solo elegir el transductor adecuado y adaptarlo a la interfaz del conversor analógico/digital.

La interfaz RS232, además de ser utilizada para acceder a configuraciones extras y obtener datos desde una consola, fue pensada para darle escalabilidad al sistema con la posibilidad de actualizar el firmware mediante un puerto de comunicaciones serial de una PC.

Una alternativa de mejora para este diseño en el futuro sería extender la funcionalidad del sistema agregando una interfaz que permita establecer los parámetros dimensionales del elemento a balancear y, mediante un algoritmo de cálculo adecuado, poder balancear diferentes tipos de discos o ruedas. Del mismo modo y, si la mecánica de la máquina permitiera adaptar otro tipo de sensor de posición, se podría mejorar la precisión de cálculo implementando la detección de las posiciones del eje giratorio, lo que a su vez facilitaría al usuario la colocación exacta de la masa compensadora.

REFERENCIAS

- [1] W. B. Rowe, "Principles of Modern Grinding Technology", Elsevier, 2009.
- [2] V. Wowk, "Machinery Vibration, Balancing", McGraw-Hill Professional, 1998.
- [3] S. A. Ansari, R. Baig, "A PC-Based Vibration Analyzer for Condition Monitoring of Process Machinery", IEEE Trans. Instrum. Meas., 1998, vol. 47, pp. 378-383.
- [4] G. Betta, C. Liguori, A. Paolillo, A. Pietrosanto, "A DSP-Based FFT-Analyzer for the Fault Diagnosis of Rotating Machine Based on Vibration Analysis", IEEE Trans. Instrum. Meas., 2002, vol. 51, pp. 1316-1322.
- [5] C. Li, Y. Wang, "A Novel Design of Electric Measurement System for Balancing Machine", IEEE, 2010.
- [6] C. Li, T. Z. Qi, Z. Zhang, J. Ni, "A Novel Design of a Hard-bearing Dynamic Balancing Machine", 15th International conference on Mechatronics and Machine Vision in Practice, 2008.
- [7] Alercia Hnos., "Rectificadoras sin centros. Características Técnicas Modelo RSC-127", <http://www.alerciahnos.com.ar>.
- [8] C. Scheffer, P. Girdhar, "Practical Machinery Vibration Analysis and Predictive Maintenance", Newnes, Elsevier, 2004.
- [9] R. Tiwari, "Dynamic Balancing Of Rotors. Theory & Practice of Rotor Dynamics", 2008
- [10] S. Goldman, "Vibration spectrum analysis: a practical approach", Industrial Press Inc., 1999.
- [11] B. Lent, "Simple Steps to Selecting the Right Accelerometer", Endevco Corp., 2009.
- [12] G. D. Roca, "Diseño de un sistema de actualización de firmware para un sistema embebido", CASE 2011, pp. 140-145.
- [13] Microchip Technology, "PIC32MX5XX/6XX/7XX Family Data Sheet", DS61156D Rev. D, 2010.
- [14] Microchip Technology, "PIC32 Family Reference Manual, Sect. 12 IO Ports", DS61120D Rev. D, 2008.
- [15] Microchip Technology, "PIC32 Family Reference Manual, Sect. 14 Timers", DS61105D Rev. D, 2008.
- [16] Microchip Technology, "PIC32 Family Reference Manual, Sect. 17 10-Bit A/D Converter", DS61104D Rev. D, 2008.
- [17] A. V. Oppenheim, A. S. Willsky, S. Hamid Nawab, "Señales y Sistemas", Prentice Hall, Segunda Edición, pag.213, 1997.
- [18] Microchip Technology, "32-Bit Language Tools Libraries", DS51685D Rev. D, pag.111, 2009.



Sistema embebido de bajo costo como instrumento auxiliar de laboratorio para operación remota

Ing. Alfredo Taddei, Mg. Ing. Héctor Riso, Dr. Pedro E. Colla

Especialidad en Sistemas Embebidos – Instituto Universitario Aeronáutico

Av. Fuerza Aerea Km 8 ½, Córdoba, Argentina.

{ataddei,hriso,pcolla}@iua.edu.ar

Abstract- Se hace un análisis sobre la utilización de sistemas embebidos como instrumentos de laboratorio tanto para realizar tareas que ya eran ejecutadas como para nuevos usos auxiliares. Se hace hincapié en la capacidad de trabajo remoto provista por la conexión de red. Se describirá el hardware objeto de análisis y se demostrará a través de dos sencillas aplicaciones su potencialidad como instrumento auxiliar de laboratorio.

Keywords: *instrumentos, laboratorio, sistemas embebidos.*

I. PROBLEMA

En el ámbito de un laboratorio de comunicaciones ópticas donde se testea un circuito ASIC (Application Specific Integrated Circuit) transceptor de señales, una de las pruebas que se realiza consiste en hacer funcionar el integrado por un tiempo ya sea para caracterizar su performance en sistemas con tasas de error muy precisas o su integridad física [1]. Es deseable durante estas pruebas sensar variables del entorno y registrarlas para poder detectar problemas en caso de fallos e identificar el estado del entorno en dicho momento sin un personal técnico presente.

Para auxiliar a esta operación se necesita de un instrumento encargado de adquirir estas señales, registrarlas, almacenarlas y tenerlas disponibles remotamente para consulta. Las variables de entorno a recolectar pueden ser analógicas o digitales tales como tensión proporcional a la temperatura del integrado, tensiones de alimentación al integrado, la posición de las llaves de configuración en la placa o registros con información provenientes del integrado o de la placa de evaluación. Un instrumento de tamaño reducido, ciertamente facilita su traslado a distintas ubicaciones donde se requieren sus funcionalidades para el tipo de prueba requerida.

En otro orden, cuando el integrado se encuentra en fase de validación y verificación es necesario trabajar con la última versión de firmware; esto se asegura típicamente mediante la utilización de una PC con conexión un versionador (SVN x ej). Resulta de interés entonces reemplazar esta PC para la función de programación de firmware por un instrumento con conexión Ethernet e interfaz de comunicación con el integrado mediante protocolos como el I²C (Inter Integrated Circuit) o SPI (Serial Peripheral Interface). De esta manera se podría lograr un programador transportable capaz de actualizarse remotamente.

En resumen, se identifica la clara necesidad de tener un instrumento de laboratorio suficientemente flexible y de bajo costo para reemplazar el uso de PC. Este trabajo describe un sistema de esta índole basado en sistemas embebidos.

II. PROPUESTA

Los sistemas embebidos han ido desarrollándose a tal punto que en muchos casos la totalidad de las funciones de un instrumento (adquisición, control, almacenamiento, análisis y presentación de mediciones) pueden ser cubiertas por un sistema de estas características. Como solución a lo planteado en la sección anterior se diseñará una prueba de concepto de un instrumento para esta situación real de laboratorio y se explicará como mejora aspectos de trabajo y agrega posibilidades que no eran contempladas anteriormente, como obtener.

Con esta propuesta, que va mas allá de un sensor autónomo, muchas de las funciones antes residentes en un instrumento o en una plataforma PC pueden ser realizadas en un sistema embebido de bajo costo; aún teniendo en consideración las limitaciones de performance. Su valor agregado, consiste en el acceso remoto a mediciones auxiliares en circuitos donde no se había contemplado la posibilidad de medición de estas variables en el diseño del hardware de integración del ASIC.

Se agrega a esta propuesta, la capacidad de administrar el firmware que configura los registros del ASIC. Mediante una conexión a un servidor se descarga la última versión de firmware y luego se la envía al dispositivo bajo testeо. Esta aplicación de programación remota, potencia los futuros usos del instrumento ya que permite tomar control del dispositivo y cambiar sus condiciones internas.

En la Figura 1, se describe esquemáticamente la arquitectura de la interacción entre el instrumento y el hardware del integrado a testear. En la misma se puede observar el ASIC con su entorno de testeо y el instrumento auxiliar con sus componentes de comunicación (SPI, USB,I2C, etc) así como el registro de variables (ADC, GPIO), el almacenamiento (SD Card) y los medios de comunicación remota (Ethernet, Web Server).

En resumen, se propone un instrumento novedoso que toma mediciones auxiliares del entorno de prueba, las pone disponibles para su consulta remota y accede a los registros y programa el firmware del ASIC a testeо. Tantas funciones solo podrían integrarse en una PC con hardware adicional o un instrumento dedicado. Ambas alternativas requerirían diferentes costos que esta propuesta supera, como se describe en la sección III, combinando lo mejor de ambas.

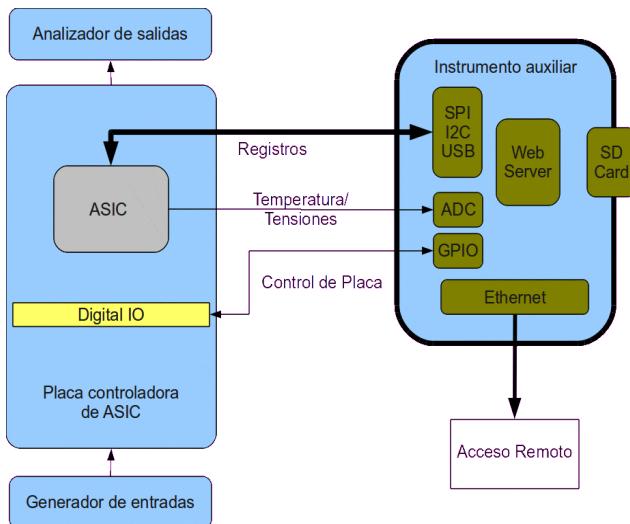


Figura 1. Arquitectura de Instrumento de Medición

III. VENTAJAS

Actualmente conviven dos tendencias en el área de instrumentación. La primera consiste en usar a la PC como "instrumento virtual" añadiéndole el hardware necesario para la adquisición. La segunda consiste en realizar el instrumento íntegramente en un sistema embebido con un microcomputador, microprocesador o microcontrolador, conformando lo que comúnmente se conoce como sistema autónomo de adquisición y registro [2].

Los "instrumentos virtuales" son periféricos, o placas de adquisición, conectadas a una computadora la cual configura, controla y accede a la placa y sus mediciones a través del sistema operativo. Esto dota a una computadora la habilidad de ser un dispositivo modular al cual se añade un hardware con el cual se comunica para realizar las funciones de instrumento propuestas (en este caso el sensado y comunicación de la medición). Bajo este paradigma, la medición se realiza en el periférico externo y el análisis, almacenamiento, comunicación, exposición de los datos y control en el sistema operativo o software dedicado. Ejemplos de este tipo de instrumental encontramos en laboratorios virtuales de universidades, donde todo la configuración de prueba se hace operable remotamente [3]. También existen aplicaciones de monitoreo en varios ámbitos cuya comparativa con un instrumento específico para su propósito resulta mucho más económica y flexible [4].

Los microcontroladores hoy son la alternativa más económica ya que integran el procesamiento y los periféricos. Su utilización en varios mercados y en particular para el consumo masivo ha conseguido que sus precios se reduzcan. Con un sistema embebido, la totalidad de las funciones de un instrumento pueden ser realizadas en el mismo hardware. La adquisición a través de sus periféricos (dentro del micro controlador o en la misma placa de desarrollo), el almacenamiento en memorias accedidas por el micro, la comunicación y el control a través de sus interfaces, y la exposición mediante los servicios que brinda el sistema

operativo, ya sea un servidor Web o una gráfica en pantalla. Los sistemas autónomos de recolección de datos, como el propuesto para este trabajo, tienen en su núcleo un microprocesador o un microcontrolador, siendo este último la opción preferencial ya que provee las funcionalidades necesarias para los periféricos en el chip mismo, reduciendo espacio y costos [5]. Asimismo, al considerar la propuesta como un "computador reducido" podemos migrar a futuro todas las funcionalidades que actualmente se resuelven en una PC de escritorio dotando a esta propuesta la capacidad de ser integradora de mas actividades y capaz de ser versátil para ajustarse a estos nuevos usos. Algo que instrumentos sensores autonomos con GPIB no pueden llegar a ser. Por último el sistema embebido es mas eficiente en consumo de energía siendo esta otra ventaja económica ya que el costo operativo de un "instrumento virtual" tiene asociado el consumo de una PC de escritorio.

En resumen, las ventajas a obtener de esta propuesta son:

- Costo de adquisición y operación.
- Acceso remoto a los datos.
- Facilidad de transporte en uso de campo.
- Versatilidad.
- Capacidad de integración a futuro.
- Funcionalmente satisfacer la problemática expuesta.

IV. DISEÑO

Con el fin de satisfacer las necesidades anteriormente planteadas se implementaron en este instrumento dos funcionalidades.

La primera, será funcionalidad de sensor. Consiste en adquisición de variables del entorno del laboratorio descriptas anteriormente, así como su registro y almacenamiento, todo esto con la posibilidad de comando y consulta de manera remota.

Como segunda, será funcionalidad de programación remota. El instrumento deberá conectarse a un servidor desde el cual descargará un firmware y una vez descargado procederá a enviarlo al dispositivo bajo testeо. De esta manera remotamente se puede ajustar la programación del integrado y funcionar como un programador que obtenga la ultima revisión de configuración. A futuro esta función facilitaría la posibilidad de hacer pruebas reactivas, es decir que luego de procesar variables recogidas se detecte una condición que dispare un cambio en la configuración del integrado mediante la funcionalidad descripta.

A. Hardware

Por cuestiones de conveniencia logística para el trabajo se utilizó la placa de desarrollo MINI2440 (Figura 2). Consiste en un integrado Samsung S3C2440A que implementa un microprocesador ARM920T [6]. El propósito de este trabajo es brindar la solución optando por este tipo de sistema como alternativa a los Instrumentos Virtuales por las ventajas descriptas en la sección III.

Las prestaciones de la placa de desarrollo del microcontrolador ARM[7] significativas y suficientes para el propósito de este trabajo son:

1. Interfaces SPI e I²C.
2. 8 entradas analógicas multiplexadas a 500KspS de 10 bit de resolución.
3. 130 entradas de propósitos generales (digitales) GPIO.
4. Controlador de tarjeta de almacenamiento SD.
5. Puerto Ethernet RJ-45 10/100M (Integrado DM9000)
6. Sistema operativo Linux con servicios de FTP y WebServer (BOA).

B. Análisis de funcionalidad de sensor

El flujo de actividades involucradas en el proceso se describen en la Figura 3. En ella quedan reflejadas las distintas tareas a codificar y orquestar en el programa. En 1) El inicio del programa configura el entorno, declara los procesos y sus variables y registra las variables externas tomadas del argumento con el que se llama al programa. En 2) se da inicio a los procesos concurrentes que sensarán la variable analógica en 2.1) y las banderas digitales en 2.2). Siguiendo la secuencia, 3.1) realiza una lectura periódica, cuyo periodo es programable externamente, del periférico conversor analógico digital y hace accesible al programa principal el valor. Al mismo tiempo, 3.2) sensa el estado de las entradas digitales externas y registras los cambios de estado para que también sean accesibles desde el orquestador. Una vez que estos datos están disponibles 4) los imprime en un archivo ubicado en el espacio de la tarjeta SD. El flujo recircula nuevamente desde las actividades del nivel 3) volviendo a tomar las muestras periódicas y los cambios de estado. La condición de salida del programa puede ser una señal externa que interrumpe el flujo o el fin del tiempo estimado de la prueba del laboratorio.

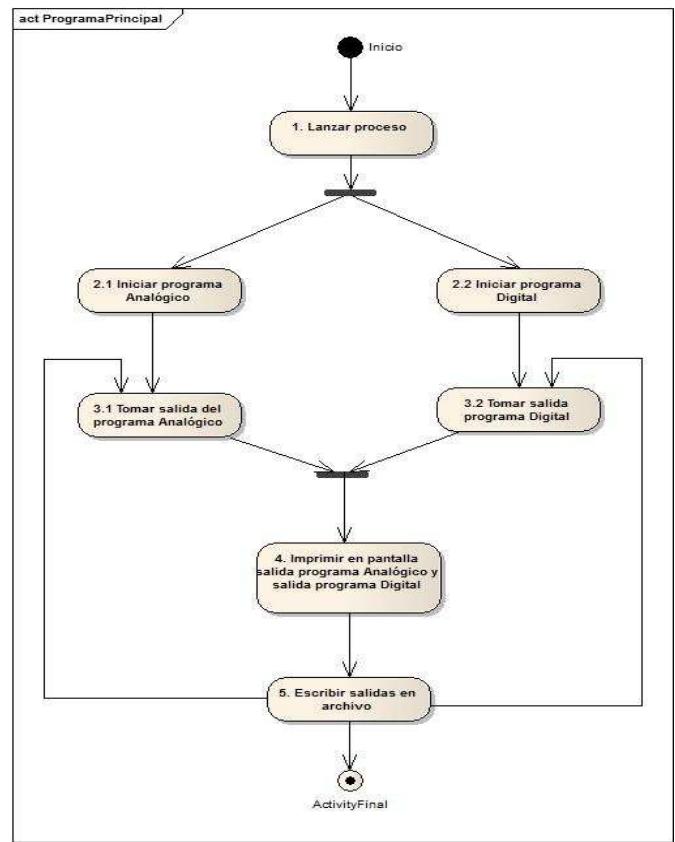


Figura 3 Diagrama de actividades de sensor

C. Análisis de funcionalidad de programación remota.

Para esta segunda funcionalidad, se desarrolló un código para un servidor TCP que ofrece el último firmware a configurar en el ASIC a testear, y un código para el cliente que descarga del servidor este firmware y lo envía mediante el bus I²C. Se basa en un protocolo simple donde el servidor envía el Firmware ante un requerimiento del cliente mediante un comando. Una vez que el cliente recibe el dato del firmware, cierra la conexión y procede con su segunda tarea de enviarlo a la memoria donde se alojan los registros. La elección del protocolo TCP por sobre UDP para el envío de los datos se debió a la necesidad de contar con un control de transmisión que garantice que el firmware a testear es el enviado desde el servidor. En la Figura 4 se muestra el diagrama de secuencia de esta funcionalidad.

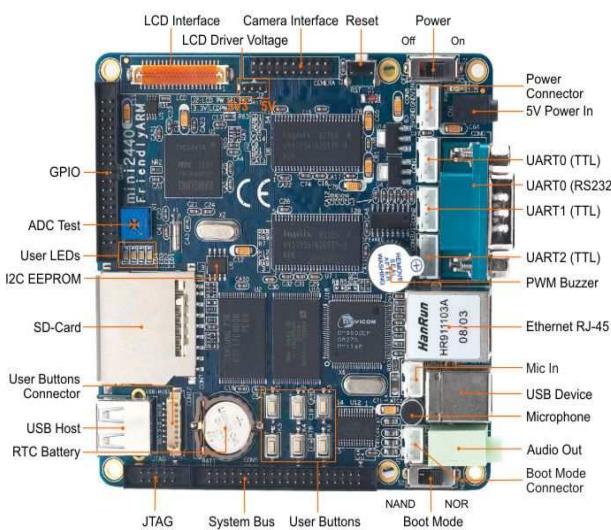


Figura 2 Fotografía de placa MINI2440

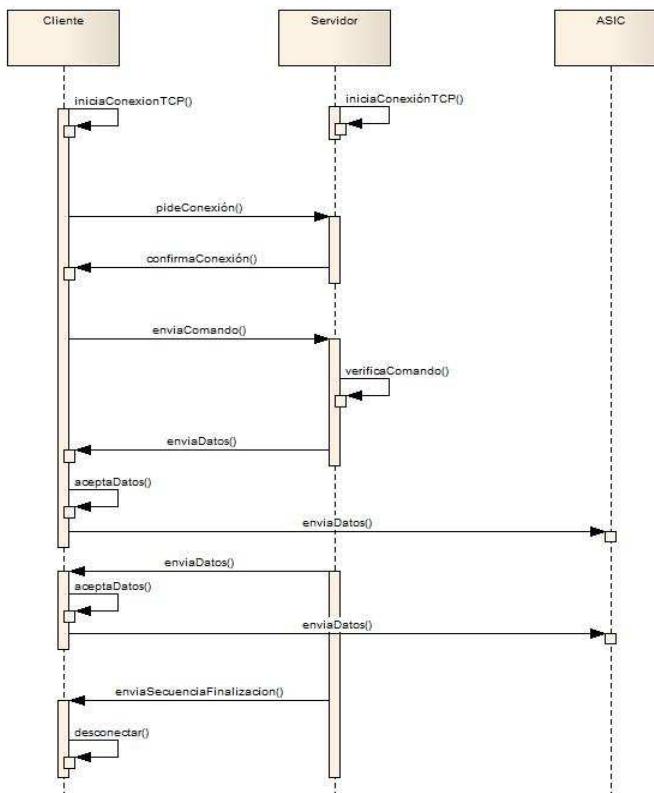


Figura 4 Funcionalidad de Programación Remota

V. IMPLEMENTACIÓN

Se implementaron ambas funcionalidades en la placa MINI2440 a un nivel de prueba de concepto que permite verificar la factibilidad funcional de lo propuesto. Las tareas a realizar del microprocesador fueron codificadas en lenguaje ANSI C y compiladas para un sistema operativo Linux embebido. Toda la comunicación remota con el instrumento se hizo mediante el servicio *Telnet* prescindiendo de una interfaz gráfica. El acceso remoto puede lograrse mediante un sitio web embebido con el WebServer BOA que provee el software original de la placa y *scripts CGI* (Common Gateway Interface) que lanzan los ejecutables.

Para la funcionalidad del sensor el programa principal crea dos procesos en Linux *pthreads* que se ejecutan concurrentemente.

```

//Creating buttons thread
pthread_create(&breader,NULL,
               (void*)&buttons_reader_function,NULL);

//Creating ADC Read thread
pthread_create(&areader,NULL,
               (void*)&adc_reader_function,NULL);
  
```

La función *buttons_reader_function*, que implementa el proceso de adquisición digital, lee el estado de los puertos de interrupción donde están conectados los botones de prueba de la placa y emulan el puerto GPIO a implementar. Inicia abriendo el archivo (*handler*) de donde leerá los valores

```

//Opening buttons handler
buttons_fd = open("/dev/buttons", 0);
  
```

Luego ejecuta el lazo del proceso testeando la integridad de la lectura y notificando los valores recibidos. El lazo y la función de lectura se representan en este código abreviado:

```

while(1)
{
    if (read(buttons_fd, current_buttons, sizeof
            current_buttons) != sizeof current_buttons)
        //ERROR EN LECTURA
        perror("read buttons:");
    }
    //Código que detecta los cambios de
    //estado y registra los valores leídos
} //end while(1)
  
```

La función *read()* que lee “/dev/buttons” es bloqueante, lo cual determinó que el programa principal fuera hecho con hilos concurrentes (*pthreads*) para que los demás procesos pudieran ejecutarse concurrentemente.

La función *adc_reader_function* que implementa la adquisición analógica abre el archivo de lectura (*handler*) donde el driver guarda el valor actual e inicia el *loop* donde lee un valor de tensión y lo registra.

```

int fd = open("/dev/adc", 0);
while(1)
{
    char buffer[buffer_size];
    int len = read(fd, buffer, sizeof buffer -1);    if (len
    > 0)
    {
        buffer[len] = '\0';
        sscanf(buffer, "%d", &adcvalue);
    }
    else{
        perror("read ADC device:");
    }
    usleep(delay);
}
  
```

Luego de la lectura en la variable *buffer* y su posterior asignación en la variable global *adcvalue*, el loop espera un retardo programable con la variable *delay* en la función *usleep*. Finalmente, se implementa una función que recoge las variables globales y las registra en un archivo dentro de la tarjeta SD.

Para la funcionalidad de programación remota se implementa una arquitectura cliente-servidor. El componente servidor abre una conexión (*stream*) TCP y configura su puerto con un argumento de entrada chequeando el éxito de la operación y aguardando peticiones.

```

sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0)
    error("ERROR opening socket");
bzero((char *) &serv_addr, sizeof(serv_addr));
portno = atoi(argv[1]); //port number argument
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = INADDR_ANY;
serv_addr.sin_port = htons(portno);
if (bind(sockfd,(struct sockaddr*)&serv_addr,sizeof(serv_addr)) < 0)
    error("ERROR on binding");
listen(sockfd,queue_limit);
  
```

Posteriormente, luego de que detecta la cadena de texto BOOT enviada por el programa cliente, procede a enviar el firmware con la función *write()*.

```

n=read(newsockfd,buffer,buffer_size);
//código de chequeo de integridad de datos leídos
  
```



```
//Parsing cadena BOOT
if((strstr(buffer,"BOOT")!=NULL))
n=write(newsockfd,minibuffer,buffer_size);
```

Por su parte el componente cliente se implementa de modo que éste abre una conexión (*stream*) TCP y configura la dirección IP (*argv[1]*) y el puerto del servidor (*argv[2]*) a través de argumentos de entrada chequeando el éxito de la conexión y abortando en caso de falla.

```
portno = atoi(argv[2]);
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0)
    error("ERROR opening socket");
server = gethostbyname(argv[1]);
if (server == NULL) {
    fprintf(stderr,"ERROR, no such host\n");
exit(0);
}
bzero((char *)&serv_addr, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
bcopy((char *)server->h_addr, (char *)
&serv_addr.sin_addr.s_addr, server->h_length);
serv_addr.sin_port = htons(portno);
if(connect(sockfd,(struct sockaddr*)&serv_addr,sizeof(serv_addr)) < 0)
error("ERROR connecting");
```

Una vez iniciada la comunicación y enviado el comando “BOOT” por el cual el protocolo del servidor inicia el envío del firmware, el cliente a medida que recibe los datos los envía a una memoria conectada al bus I²C con los drivers provistos por el fabricante de la placa. La comunicación finaliza cuando el servidor envía la cadena “OUT”.

```
n=read(sockfd,buffer,buffer_size);
if (n < 0)
    error("ERROR reading from socket");
while(strstr(buffer,"OUT")==NULL)
{
data=atoi(buffer);//parse to int
//Check valid data code
eprom_write_byte(&e, addr,data);
//Write in EEPROM via I2C and verification
n = read(sockfd,buffer,buffer_size);//New read
if (n < 0)
    error("ERROR reading from socket");
}//end while
```

VI. VERIFICACIÓN Y VALIDACIÓN

Es necesario realizar la verificación de forma que se revise la integridad de lo implementado (¿hemos implementado correctamente?) así como la validación de forma de corroborar que la funcionalidad implementada satisface las necesidades planteadas (¿hemos implementado el dispositivo correcto?).

En el caso de la funcionalidad de sensor su verificación fue ejecutada en la misma placa de desarrollo con componentes conectados al microcontrolador destinados a probar periféricos (ver en Figura 2 los elementos ADC Test y User Buttons). Se comprobó el correcto funcionamiento de la adquisición digital y analógica utilizando períodos de registro distintos. Se accedió remotamente a los datos registrados y guardados en el archivo mientras se realizaba la prueba utilizando el servicio FTP que provee Linux (*ftpd*) y se comprobó que el contenido del archivo correspondía a los estímulos ejercidos manualmente en el entorno.

En el caso de la funcionalidad de programación remota, se probó emulando un servidor TCP en una computadora portátil que envía bajo petición el firmware al instrumento y el mismo,

luego de recibido, es programado en una memoria mediante el bus I²C. Se verificó la integridad de los datos recibidos comparando la secuencia enviada con la lectura de los datos de la memoria EEPROM conectada al bus I²C.

Si bien abundan las alternativas dentro de los sistemas embebidos que consisten en una placa de desarrollo de un microcontrolador, el hardware que se propone se ajusta a las necesidades del instrumento a desarrollar en cuanto a funcionalidad y costo delimitadas para este trabajo. El aumento de prestaciones y rendimiento impactan en el costo de la solución. La placa de desarrollo MINI2440 del microcontrolador S3C2440 que se utiliza en este trabajo tiene un costo de US\$89 o US\$109 según se ordene con pantalla LCD touch de 3.5” o no [8]. A comparación con las placas de adquisición de datos que requieren de una PC o incluso las placas de computadores industriales (*Single Board Computers*) u otros sensores autónomos, cuyos valores van desde los US\$200 a US\$1000 observamos la ventaja de usar esta plataforma para desarrollar un instrumento nuevo [9]. Se aprecia en la Tabla 1 la diferencia de costos entre posibles soluciones comerciales [10][11].

TABLA 1 COMPARACIÓN ENTRE DISTINTAS ALTERNATIVAS

Nombre	Tipo	Costo	Consumo
MINI2440	Embebido económico con microcontrolador ARM	US\$89	1.5W
iUSBDAQ-U120816	Tarjeta de Adquisición de datos USB para PC	US\$99 + Computadora	>30W (PC)
DI-710-EHS	Sensor autónomo comercial, portable con Ethernet y tarjeta SD	US\$799	2W

El costo de operación fue asociado a la potencia de consumo del instrumento completo. En la Tabla 1 se hace una comparación de un ejemplo de cada alternativa.

Otro aspecto valioso de esta propuesta consiste en la posibilidad de desmontar el instrumento del escritorio de trabajo fácilmente y transportarlo con comodidad por su tamaño menor (10cm x10cm) con respecto a una computadora. Al ser una prueba de carácter no frecuente, la posibilidad de desmontar el instrumento fácilmente y llevarlo a otro sitio donde se necesite es una mejora si se lo compara con la alternativa de implementar todas estas funcionalidades en todas las placas bajo testeo [12].

VII. CONCLUSIONES

En este trabajo se desarrolló una aplicación de adquisición, registro, monitoreo y acceso de variables de entorno en un laboratorio de testeo de ASIC de comunicaciones ópticas. Siendo la tendencia de esta investigación reemplazar la PC que coordina la prueba por un sistema embebido, se desarrolló una segunda aplicación que descarga de un servidor la última versión de firmware y su posterior aplicación al ASIC bajo testeo.

Se comparó la alternativa de sistema embebido basado en un microcontrolador a usar placas de adquisición como “instrumentos virtuales”, resultando más económica y versátil



al momento de hacer un instrumento transportable para la prueba objeto de este trabajo.

Se seleccionó un hardware que cubre las expectativas de prestaciones para este trabajo y posteriormente se describieron las implementaciones de las dos aplicaciones propuestas. La verificación de lo propuesto se hizo emulando las condiciones externas con hardware de la misma placa de desarrollo del sistema embebido elegido.

Como conclusión se puede afirmar que estas aplicaciones simples desarrolladas pueden ser cubiertas rápidamente con este hardware de sistema embebido económico y en pocas líneas de código obtener una aplicación útil que cumpla con lo requerido a menor costo y portable. A medida que los requerimientos de estas funcionalidades se hagan más estrictos, se deberán tener en cuenta muchos más aspectos de implementación de estos programas que podían ser soslayados anteriormente (por ej.: muestreo uniforme, concurrencia de tareas). A futuro debería hacerse una revisión minuciosa de la robustez del código programado.

Mejoras posibles incluyen desarrollar un servidor Web para comandar mas amigablemente el instrumento e ir incorporando actividades que hasta el día de hoy las realiza una computadora de escritorio. Una de las tareas centrales es la de coordinar el funcionamiento de los instrumentos que proveen las señales de entrada, analizan señales de salida y estresan las condiciones del entorno. El estándar utilizado es GPIB con tendencia a migrar las comunicaciones a LAN Ethernet, lo cual sería beneficioso para el tipo de hardware propuesto pues no habría ningún desarrollo de bus de comunicaciones que realizar. Se obtendría un instrumento capaz de comandar a cualquier instrumento con GPIB tal cual lo hace una PC y de esa manera realizar cualquier tipo de tests como test de regresion, test de humo o test reactivo. En [13], el fabricante de hardware de instrumentos Agilent, explica con una nota de aplicación esta posibilidad en Linux.

REFERENCIAS

- [1] Birolini, Alessandro: Reliability Engineering, Springer (2007)
- [2] Mackay S., Park, J: Practical Data Acquisition and Control Systems. Elsevier p. 204 (2003)
- [3] Ko C.C., Chen B.M., Hu S., Ramakrishnan V., Cheng C.D., Zhuang Y., Chen J: A Web-Based Virtual Laboratory on a Frequency Modulation Experiment. IEEE Transactions on systems, man, cybernetics Part C: Applications and reviews, Vol. 31, No. 3, (August 2001)
- [4] Batista J., Afonso J.L., Martins J.S., Low-Cost Power Quality Monitor Based on a PC, IEEE International Symposium on Industrial Electronics (2003)
- [5] Mackay S., Park, J: Practical Data Acquisition and Control Systems. Elsevier p. 210 (2003)
- [6] ARM Information Center, <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0151c/index.html>
- [7] Mini2440 Hardware Overview, http://www.friendlyarm.net/dl.php?file=mini2440_overview.pdf
- [8] Friendly ARM MINI2440, <http://www.andahammer.com/mini2440-sdk/>
- [9] Omega Press, Transactions in measurement and control, Vol II, Data Acquisition, Chapter 5. <http://www.omega.com/literature/transactions/volume2/trantocvol2.html>
- [10] iUSBDAQ – U120816, Lowest cost USB data acquisition module with 8 channels 12bit analog inputs, 16 bi-directional DIOs, 2 PWM outputs,

one 16bit counter. <http://www.hytetautomation.ca/U120816.aspx?productId=1>

- [11] DI-710 Data Logger products operate Stand-alone or PC-connected. <http://www.dataq.com/products/hardware/di710.htm#tblfn>
- [12] Mackay S., Park, J: Practical Data Acquisition and Control Systems. Elsevier p. 232 (2003)
- [13] Agilent Application Note 1465-29, Using Linux to Control LXI Instruments Through TCP.



Embedded speaker verification in low cost microcontroller

Maximiliano Lizondo, Pablo D. Agüero, Alejandro J. Uriz, Juan C. Tulli and Esteban L. Gonzalez

Facultad de Ingeniería
Universidad Nacional de Mar del Plata
Mar del Plata, Argentina
Email: mlizondo@fimdp.edu.ar

Abstract—Automatic speaker recognition is the use of a machine to identify or verify the identity of an individual from a spoken sentence. This paper describes an implementation of an embedded speaker verification system that may be used for an electronic door lock, among other possible applications. The system is built in a low cost dsPIC from Microchip, which combines features of microcontrollers with characteristics of DSPs in a single 16-bit high-performance core. Some special programming techniques used for the implementation are described. The aim was optimizing the code for speed and memory usage. Experimental results in MATLAB of the embedded speaker verification algorithms show promising results, with a false acceptance rate of 8% for a false rejection rate of 12%.

Index Terms—Speaker verification, embedded system, dsPIC.

I. INTRODUCTION

BIOMETRIC recognition refers to the use of distinctive characteristics to identify individuals [1], [2]. These biometric identifiers are usually classified into physiological or behavioural characteristics.

Physiological biometrics, like fingerprints, face, hand geometry, retina or iris, are physical characteristics that can be measured at some particular point in time. On the other hand, behavioural biometrics like signature, voice or gait, consist of actions that extend over time. Unlike physiological biometrics, behavioural biometrics are learned or acquired over time and they can be easily and deliberately changed [2].

Speech is one of the most natural modalities of human interaction, a fact corroborated by many years of research and development in speech processing. Recent developments in speech technologies have finally provided truly functional applications. An example of these applications is the role of speech as a biometric identifier for automatic speaker recognition.

Automatic speaker recognition is the use of a machine to identify an individual from an utterance. Recently, this technology has undergone an increasing importance in applications such as access control, transaction authentication, law enforcement, forensics, and system customisation, among others.

One of the central questions addressed by this field is what conveys speaker identity in the speech signal. Traditionally, automatic speaker recognition systems have relied mostly on short-term features related to the spectrum of the voice. However, human speaker recognition relies on other additional

sources of information. Therefore, these sources may also play an important role in the automatic speaker recognition task, by adding complementary knowledge to the traditional spectrum-based recognition systems and thus improving their accuracy.

Voice is not expected to have enough distinctiveness to allow the recognition of an individual from a large database of speakers. Moreover, it is characterised by three important disadvantages: first, a speech signal can be degraded in quality by the microphone or the transmission channel; second, voice can be affected by the health of a person, stress or emotions; and finally, it has been shown that some people are extraordinarily skilled in mimicking voices [1], [2]. However, voice is a non-intrusive biometric with a high acceptability. Moreover, it is nowadays the only feasible biometric identifier in applications requiring person recognition over a telephone system [1], [3].

This paper describes an implementation of an embedded speaker verification system that may be used for an electronic door lock, among other possible applications. Our system is built in a low cost dsPIC from Microchip, which combines features of microcontrollers with characteristics of DSPs in a single 16-bit high-performance core.

The paper is organized as follows. Section II briefly describes speaker verification systems, their architecture, feature extraction and statistical models. Section III depicts the proposed embedded speaker verification system, with details of the implementation. Finally, Section IV shows the conclusions and some future directions.

II. SPEAKER VERIFICATION SYSTEM

Depending on the application, an automatic biometric recognition system can run in two modes: identification and verification [4].

In identification mode, the aim is to determine which speaker, in a set of known users (whose models are stored in the database), matches the unknown user. In the verification mode, the aim of a system is to determine whether an unknown user is who he/she claims to be or an impostor. Applications of the later mode are mainly related to access restriction in secured areas, and it is of interest in this paper.

In verification systems (see Fig. 1), a user is claiming an identity. A model corresponding to that identity must be stored in the database, which must contain an impostor model as well. The biometric features of the claimed user are compared to

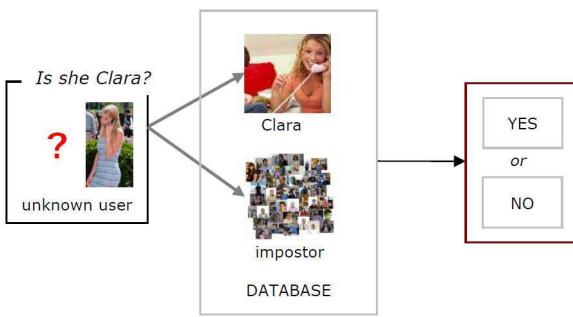


Fig. 1. Automatic speaker verification system.

the model of the claimed identity and to the impostor model. If a user seems to be closer to the claimed identity, he/she will be accepted as a known user. Otherwise, the user will be rejected and treated as an impostor.

After having computed a score of similarity between the input user and the corresponding templates stored in the database, a decision is taken whether the user must be accepted or rejected by the system. However, such decision can be correct or not. If the decision is incorrect, two different errors can occur [3]:

- False rejection: the system rejects a valid identity claim.
- False acceptance: the system accepts an identity claim from an impostor.

Both types of errors give rise to two types of error rates, which are commonly used to measure the performance of a system:

- False rejection rate (FRR): percentage of incorrectly rejected clients.
- False acceptance rate (FAR): percentage of incorrectly accepted impostors.

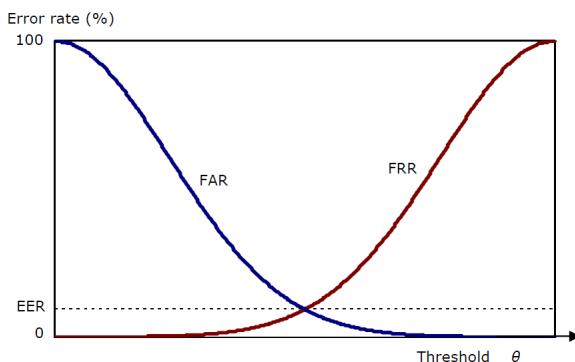


Fig. 2. False rejection rate and False acceptance rate as a function of the threshold θ .

Therefore, when designing a biometric verification system, the decision threshold (see Fig. 3) must be adjusted so that both errors are as low as possible, or one of the errors must be always below a certain threshold when a specific application requires this property.

A. Architecture of a speaker recognition system

A typical biometric recognition system consists of two phases (see Fig. 3): the training phase (enrollment) and the testing phase (recognition). In the training phase, biometric measurements from the users are captured by means of biometric sensors or readers. Then, relevant information is extracted from the biometric measurements (feature extraction) to build a user model, which will be stored in a database.

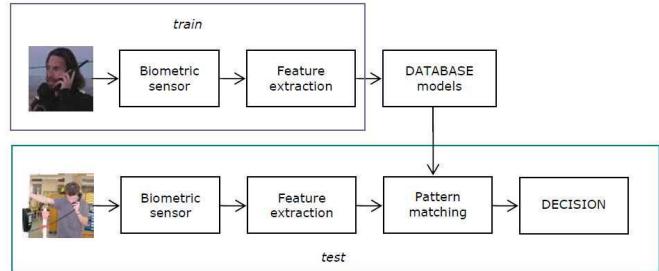


Fig. 3. Architecture of a typical biometric recognition system.

In the recognition phase, biometric readers are also used to capture biometric information of the user to be recognised. Relevant information is extracted from the data provided by the biometric sensors in the feature extraction step. This information is compared with the stored user models of the database, computing the degree of similarity (the term score is also used). This similarity measure will be used to determine whether the user corresponds to one of the users whose model is stored in the database or not. Finally, a decision is taken based on the computed similarity scores.

B. Feature extraction

Feature extraction or speech parameterisation in the speaker verification field consists in transforming the speech signal into a set of feature vectors [3]. The aim of this transformation is to obtain a relatively low-dimensional representation, more suitable for statistical modeling, the computation of a distance, or any other kind of score (in order to enable comparisons using simple similarity measures), while preserving the information related to the identity of the speaker.

The most commonly used parameters in state-of-the-art speaker and speech recognition technologies are the Mel-Frequency Cepstral Coefficients (MFCC) [5], [6]. They are a representation of the short-term power spectrum of a sound, based on the linear cosine transform of the log power spectrum on a nonlinear Mel scale of frequency (see Fig. 4).

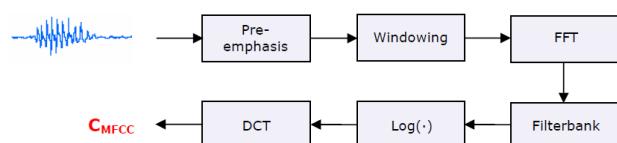


Fig. 4. Mel-frequency cepstral coefficients computation steps



Commonly, speech processing in the speaker verification task begins with a first order high-pass filtering of the speech signal to emphasise high frequency components. Then, the signal is segmented in temporal frames and typically windowed with a Hamming window to reduce the discontinuities in the boundaries of the segmentation. This procedure is usually used for short-term analysis of speech.

The first step for the computation of Mel-frequency cepstral coefficients is performing the Discrete Fourier Transform (DFT) of the speech frame. Usually a Fast Fourier Transform is used to reduce the computation time.

The resulting values are then passed through a filterbank distributed along the frequency domain according to a Mel scale. A vector of energy values is obtained with this step: the Filter Bank Energies (FBE). The Mel scale, proposed by Stevens [7] is based on the manner how the speech perception works in the human ear. The human auditory system non-linearly resolves frequencies across the audio spectrum. Empirical evidence suggests that a system that operates in a similar nonlinear way, obtaining the desired non-linear frequency resolution, provides a better recognition performance.

The Mel scale filterbank is a series of Q triangular bandpass filters that have been designed to simulate the bandpass filtering by mimicking the human auditory response. The series of constant bandwidth triangular filters are 50% overlapped and spaced on a Mel frequency scale. On a linear frequency scale, the filter spacing is approximately linear in the range from 0 to 1000Hz, and logarithmic at higher values of frequency. The triangles are all normalised to have unit area.

After applying this filterbank, the number of coefficients is reduced, and hence the information is compacted. The variance is also reduced when averaging the samples of the DFT in each filter. Finally, a logarithmic compression and the Discrete Cosine Transform (DCT) is applied to the vector of FBE in order to obtain the MFCC.

The DCT serves two purposes. First, the DCT performs the final part of a cepstral transformation which separates the slowly varying spectral envelope (or vocal tract) information from the faster varying speech excitation. MFCC only retains the low order coefficients related to vocal tract.

The second purpose of the DCT is to decorrelate the elements of the feature vector. Elements of the log filterbank vector exhibit correlation due to both the spectral characteristics of speech and the overlapping nature of the filterbank. Such process makes the resulting decorrelated coefficients suitable for the use of diagonal covariance matrices in statistical classifiers.

Research on additional information sources in speaker recognition has been mainly focused on the use of the fundamental frequency as a complement to the vocal tract information provided by MFCC. One of the reasons is the robustness to acoustic degradations from channel and noise effects [8], [9]. Arcienega et al. [10], for example, suggest the use of F0-dependent speaker models. In the works of Sönmez et al. [11] and Adami et al. [12], the variation of fundamental frequency over time is modeled for its use in

a speaker recognition task, together with the signal energy variation.

C. Statistical models

Gaussian mixture models (GMM) are commonly used as a modeling technique in speaker verification systems. A GMM is a weighted sum of gaussian density functions that models the distribution of the feature vectors extracted from the speech signal [13], [14]. Given a D-dimensional feature vector x , the Gaussian mixture model λ_i corresponding to the speaker S_i is defined by the expression in Equation 1.

$$P(x|\lambda_i) = \sum_{m=1}^M \omega_m N(x, \mu_m^i, \Sigma_m^i) \quad (1)$$

$N(x, \mu, \Sigma)$ is a gaussian function defined as shown in Equation 2, where μ is the vector of means and Σ the covariance matrix. D is the number of elements in the D-dimensional feature vector x . M is the number of mixtures, and ω_m are the weights of each mixture, that must sum up one.

$$N(x, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} \sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (2)$$

By means of dynamic programming algorithms, the probability that a sequence of speech frames was generated by this model can be determined [15], [16]. This probability -or likelihood- is used as a score for L frames of input speech given the model [15], [16], [4].

In the recognition step, given a sequence of test feature vectors $X = [x_1, x_2, \dots, x_T]$ extracted from an unknown user's speech, the probability of the unknown speaker being the speaker S_i (assuming that vectors x_t are independent) is determined by the following expression in Equation 3, which will be used as a similarity score.

$$p(X|\lambda_i) = \prod_{t=1}^T P(x_t|\lambda_i) \quad (3)$$

D. Speaker verification system implementation

The speaker verification system implemented in this paper consists of state-of-the-art approaches. Two main speech features are used in the decision about the identity of the speaker: MFCC (13 coefficients calculated from windows with 256 samples, without window overlap) and fundamental frequency (F_0).

The proposed system performs cepstral mean subtraction (CMS) to remove channel effects in MFCC parameters. Fundamental frequency and two derived parameters, relative jitter and relative shimmer, are also used in the likelihood computation of the speaker under analysis. Jitter is a measure of the periodic deviation in the voice signal, or the pitch perturbation of the signal. Shimmer (amplitude perturbation) is similar to jitter, but instead of looking at periodicity, it measures the difference in amplitude from cycle to cycle.

Frame	F0	JR	SR	LL
1	F0 ₁	JR ₁	SR ₁	LL ₁
2	F0 ₁	JR ₁	SR ₁	LL ₁
3	F0 ₁	JR ₁	SR ₁	LL ₁
...
M	F0 _M	JR _M	SR _M	LL _M
Mean	μ_{F0}	μ_{JR}	μ_{SR}	μ_{LL}

TABLE I
MEAN FEATURE VECTOR CALCULUS

Therefore, four parameters are extracted from each speech frame to evaluate the identity of the speaker: fundamental frequency, relative jitter, relative shimmer, and log-likelihood. The later is calculated as the difference between the log-likelihood of the feature vector (MFCC) for the claimed identity model, and the log-likelihood of the feature vector for the Universal Background Model (UBM). This UBM is the impostor model. Such model must contain every possible alternative to the speaker S_i .

In this paper was used the approach that consists in training the impostor model as a single model using several speakers [17], [18], [19]. This model is usually called universal background model (UBM), and when using Gaussian mixture models, the method is known as the GMM-UBM approach.

The similarity score between the unknown speaker and the claimed identity is performed comparing the mean vector of speech features (fundamental frequency (F0), relative jitter (JR), relative shimmer (SR), and log-likelihood (LL)) over all frames of the spoken utterance, as shown in Table I.

The similarity score between the input feature vector μ and the template S_i is given by a distance $d(\mu, \mu_{S_i})$. The distance measure between these two vectors can be expressed as $d(\mu, \mu_{S_i}) = (\mu - \mu_{S_i})\Sigma(\mu - \mu_{S_i})$. The Σ weighting matrix is the inverse covariance matrix corresponding to mean μ_{S_i} . This distance is known as the Mahalanobis distance. A threshold value θ , the maximum allowable value for distance $d(\mu, \mu_{S_i})$, is used to take the decision about the identity: known speaker or impostor.

III. IMPLEMENTATION OF AN EMBEDDED SPEAKER VERIFICATION

The diagram of Fig. 5 shows the different components included in the embedded speaker verification system proposed in this paper.

The biometric sensor of the system is a microphone. The electric signal of the microphone is amplified and filtered to increase the Signal-To-Noise Ratio and to prevent aliasing during sampling.

The electric signal is sampled with the internal Analog-to-Digital Converter included in the microcontroller (dsPIC 33FJ128GP802-E/SP). The sampling rate is 16KHz and the resolution is 12 bits. The signal is segmented into frames of 256 samples, obtained using a double buffer DMA technique.

Each frame is processed to obtain different acoustic parameters, such as energy, spectrum, MFCC, fundamental frequency,

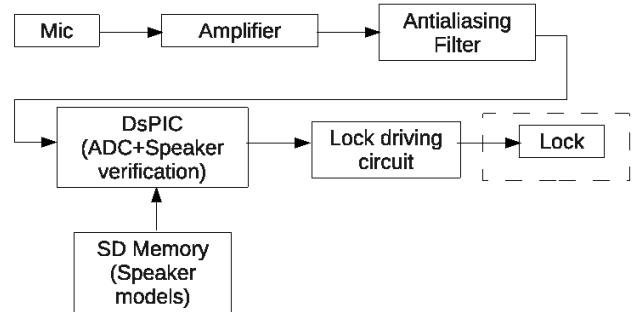


Fig. 5. Embedded speaker verification system.

jitter of fundamental frequency, and shimmer of amplitude values.

These features are analyzed using the reference models, that are loaded into the dsPIC from an SD card. These models can be changed and reloaded into the SD card using the training software that creates a custom model for the specific user.

The system proposed in this paper was implemented using a device dsPIC 33FJ128GP802-E/SP from Microchip. The main features of this device are:

- 128KB of program memory. This makes it suitable for use with cross compilers.
- 16KB of RAM. Of which 2KB are shared with direct memory access (DMA) buffer as dual ported RAM.
- Up to 40 MIPS operation.
- Low cost. As mentioned before, its price is 4 US\$, much lower than a classic DSP. This is a great advantage compared with commercial DSP devices.
- 16-bit wide data path.
- 12-bit@500ksps integrated analog-to-digital converter (ADC).
- Double-buffered input/output registers. This allows for faster operations on the ports (read-write), and also gives more flexibility on the handling of them.
- In-Circuit Serial Programming and Debugger. The device can be programmed and configured in the end application circuit.
- 28-pin SOIC packaging available. Allows for great levels of integration.

A. Limitations in RAM memory of dsPIC

One of the main limitations to implement a speaker verification in dsPIC is the limited amount of RAM memory. In order to minimize the impact of such restriction, a number of aspects were taken into account to optimize the use of memory.

Without any optimization in the use of variables in RAM memory, the total amount of memory usage is shown in Table II.

Variables Buffer and Data are the integer and floating point representation of the frame under analysis. HammingWindow has the precalculated values of a 256 point Hamming window. The variables Sinus and Cosinus have the precalculated values

TABLE II
INITIAL MEMORY USAGE

Variable	#elements	type	Memory (bytes)
Buffer	256	int	512
Data	256	float	1024
HammingWindow	256	float	1024
Sinus	256	float	1024
Cosinus	256	float	1024
Butterfly	256	char	256
RealFFT	256	float	1024
ImagFFT	256	float	1024
MFCCFilters	24x53	float	5088
DCT	24x13	float	1248
Autocorr	256	float	1024
MFCCCoefs	13	float	52
InvCovars	13x2x16	float	3328
Means	13x2x16	float	3328
Priors	2x16	float	256
Determinants	2x16	float	256
Total			21492 > 16KB

of a 256 point sinus and cosinus, and Butterfly has the necessary information for the correct calculation of the Fast Fourier Transform (FFT). The result of the FFT is put into the variables RealFFT and ImagFFT.

The Mel-frequency cepstral coefficients are calculated with the resulting spectrum of the FFT. The precalculated Mel scale filterbanks can be found in the MFCCFilters variable, and the final discrete cosine transform is computed using the DCT variable (precalculated cosinus).

Gaussian mixture models are loaded into the RAM memory of dsPIC from SD card into the variables InvCovars (inverse of the covariance matrices), Means (mean vectors), Priors (prior values of each gaussian mixture) and Determinants (determinant of each covariance matrix).

As shown in the last line, the total amount of memory necessary for our application is higher than the available memory.

A careful analysis of the variables shows that HammingWindow, MFCCFilters, DCT, and the routines that use them can be hard coded into Program memory. This decision saves 7360 bytes of memory that may be used for better Gaussian Mixture Models, that have a severe impact in the final performance of the speaker verification task.

The Fast Fourier Transform is now calculated using Microchip's subroutine FFTReal32bIP. This routine needs more memory, but the real benefit is obtained in speed.

The variable Autocorr is only used to calculate the fundamental frequency, which is then used to estimate jitter and shimmer parameters too. Therefore, Autocorr variable may point to the same memory space as pwrspect (Power Spectrum) variable , saving even more RAM memory. Actual memory usage of the system is shown in Table III.

B. Limitations in MIPS of dsPIC

Another important issue for the implementation of a embedded speaker verification system are the number of instruction per second executed in a dsPIC. The 40 MHz clock only allows 40 MIPS. Therefore, a careful selection in the implementation of the different routines is essential.

TABLE III
ACTUAL MEMORY USAGE

Variable	#elements	type	Memory (bytes)
Buffer	256	int	512
Data	256	float	1024
HammingWindow	Program	memory	0
sigReal	256	long	1024
twdfIFctr32b	768	long	3072
pwrspect/Autocorr	256	long	1024
MFCCFilters	Program	memory	0
DCT	Program	memory	0
MFCCCoefs	13	float	52
InvCovars	13x2x16	float	3328
Means	13x2x16	float	3328
Priors	2x16	float	256
Determinants	2x16	float	256
Total			13876 < 16KB

The original subroutine of Hamming window had this code:

```
int i;
for (i=0;i<bfrsize;i++)
{
    data[i]=data[i]*win[i];
}
```

The new subroutine of Hamming window has all the values hard-coded, with a benefit in memory usage and speed. The sample values of the Hamming window are literals (reduction in the usage of RAM memory), and the comparison and increment of for loop are not necessary (increment in speed).

```
data[0]*=0.08f;
data[1]*=0.0801396318543067f;
data[2]*=0.0805584426474237f;
...
```

The same approach was used to implement the subroutines to calculate Mel-frequency cepstral coefficients, with a benefit RAM usage and speed. The original subroutines took 18.7ms to calculate the MFCC for a frame, while the new hard coded subroutine only takes 5.36ms.

The use of Microchip's subroutine for Fast Fourier Transform (FFTReal32bIP) has an important benefit in speed. The original subroutine coded in C had a duration of 64ms to obtain the power spectrum of a frame, while the new subroutine that includes the code proposed by Microchip only takes 5.36ms.

The integer implementation of autocorrelation was the final improvement in the code to process a frame and calculate all the necessary parameters to take a decision in the speaker verification task. The total time to process a frame is 59ms. The total time for the decision of the system depends on the duration of the utterance (without initial and ending silences). It may be estimated as 3.6 times the duration of the vocalization.

C. Speaker verification results

Several experiments were conducted using MATLAB and train-validation-test sets to study different algorithms for speaker verification systems, mainly focused in the optimization of the number of gaussian mixtures. GMM were

trained using the standard Baum-Welch algorithm and diagonal covariance matrices.

The source code was written using both MATLAB and C30 sintaxis, and possible differences in the calculation of the algorithms was checked. The differences remain small and negligible.

Available speech from 168 speakers was divided into train and test sets. Both sets have different words, in order to build and test a text-independent speaker verification system.

The chosen architecture was tested using n-fold cross validation to obtain the ROC curve. The receiver operating characteristic (ROC), or simply ROC curve, is a graphical plot of the sensitivity, or true positive rate versus false positive rate for a binary classifier system as its discrimination threshold is varied.

The ROC for the proposed system is shown in Fig. 6. As explained in Section II, when designing a biometric verification system, the decision threshold must be adjusted to minimize false positives and false negatives. In our task, false positives should remain below threshold to prevent an impostor acceptance.

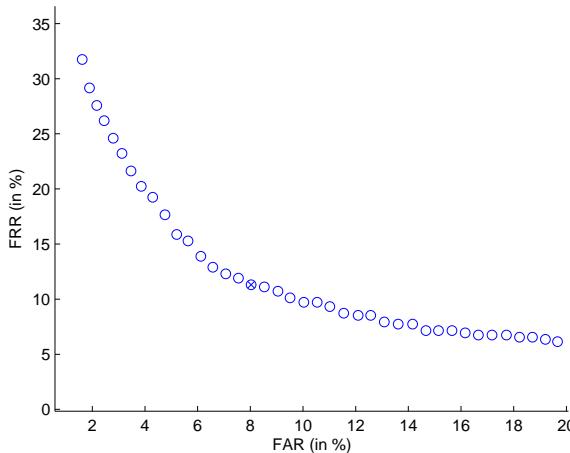


Fig. 6. ROC curve for the embedded speaker verification system.

IV. CONCLUSIONS

In this paper was described an embedded speaker verification system. The system is built in a low cost dsPIC from Microchip, which has characteristics of DSPs in a single 16-bit high-performance core. The speaker verification system is intended to operate an electrical door lock, through an electric lock driving circuit.

Experiments were performed using MATLAB to find the ROC curve of the proposed system, to get an approximation of the expected performance. Experimental results show that the system may reject impostor at the expense of also rejecting a user with the correct claimed identity. For example, the speaker verification system has a false acceptance rate of 8% for a false rejection rate of 12%. Actually, the proposed system can only

be used as an auxiliary identification technique, and not as a primary identification technology, due to these low results.

Future work will focus in two main aspects: response time and better identification performance. Response time is an important issue in this 40 MIPS speaker verification system, and improvements can be achieved through the use of inline assembler inside C language.

Better identification performance will be faced with new speaker verification techniques, which may involved additional features, different modeling techniques, and signal conditioning algorithms.

REFERENCES

- [1] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. Springer, New York, 2003.
- [2] R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior, *Guide to Biometrics*. Springer, New York, 2004.
- [3] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. Reynolds, "A tutorial on text-independent speaker verification," in *EURASIP Journal on Applied Signal Processing*, 2004, pp. 430–451.
- [4] J. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, pp. 1437–1462, 1997.
- [5] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 28, pp. 357–366, 1980.
- [6] A. Oppenheim, "From frequency to quefrency: A history of the cepstrum," *IEEE Signal Processing Magazine*, pp. 95–99, 2004.
- [7] S. Stevens, "The mel scale equates the magnitude of perceived differences in pitch at different frequencies," *Journal of the Acoustical Society of America*, vol. 8, pp. 185–190, 1937.
- [8] B. Atal, "Automatic speaker recognition based on pitch contours," *Journal of the Acoustical Society of America*, vol. 52, pp. 1687–1697, 1972.
- [9] M. Carey, E. Parris, H. Lloyd-Thomas, and S. Bennett, "Robust prosodic features for speaker identification," in *Proceedings of the ICSLP*, 1996, pp. 1800–1803.
- [10] M. Arcienega and A. Drygaljo, "Pitch-dependent gmms for text-independent speaker recognition systems," in *Proceedings of the Eurospeech*, 2001, pp. 2821–2825.
- [11] K. Sönmez, E. Shriberg, L. Heck, and M. Weintraub, "Modeling dynamic prosodic variation for speaker verification," in *Proceedings of the ICMLP*, 1998, pp. 3189–3192.
- [12] A. Adami and H. Hermansky, "Segmentation of speech for speaker and language recognition," in *Proceedings of the Eurospeech*, 2003, pp. 841–844.
- [13] D. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Communication*, vol. 17, pp. 91–108, 1995.
- [14] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 72–83, 1995.
- [15] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, pp. 4–16, 1986.
- [16] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.
- [17] T. Matsui and S. Furui, "Likelihood normalization for speaker verification using a phoneme and speaker-independent model," *Speech Communication*, vol. 17, pp. 109–116, 1995.
- [18] A. Rosenberg and S. Parthasarathy, "Speaker background models for connected digit password speaker verification," in *Proceedings of the ICASSP*, 1996, pp. 81–84.
- [19] D. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," in *Proceedings of the Eurospeech*, 1997, pp. 963–966.



Codificador de señales para Cohetería

Autores: Ing. Sergio Rago, Ing. Sergio Lara, Téc. Alejandro Acosta
con la colaboración del Ing. Javier Garayzar.

División Telemetría. Departamento de Electrónica Aplicada.
CITEDEF.

Villa Martelli. Provincia de Buenos Aires. Argentina.

srago@citedef.gob.ar, slara@citedef.gob.ar, aacosta@citedef.gob.ar, jgarayzar@citedef.gob.ar

Resumen—Este documento expone un desarrollo íntegramente diseñado y realizado en la División Telemetría (D.E.A) del Instituto CITEDEF (Ministerio de Defensa) para satisfacer los requerimientos de los Proyectos Cohete GRADICOM 2009 y 2011. Dicho desarrollo corresponde a un codificador de señales moduladas por código de pulso (*PCM*), elemento esencial para las telemetrías de cohetes.

Palabras claves- *PCM; trama; conversor; filtro; norma; analógico; digital; telemetría; frecuencia; oscilación; alimentación; ruido; resolución.*

I. INTRODUCCION

Nuestro país tiene una rica historia en lo que respecta a la cohetería. Con el transcurrir de los últimos años los lanzamientos se fueron haciendo cada vez más espaciados, lo que trajo aparejado una detención en el desarrollo de la tecnología inherente a este tipo de aplicaciones.

En lo últimos cinco años volvió a activarse la actividad aeroespacial fundamentalmente en el desarrollo de cohetes, pero en lo que respecta a la tecnología electrónica el atraso de nuestro país con respecto a las empresas líderes internacionales se hizo evidente.

Específicamente en Telemetría, los dispositivos y elementos esenciales para realizar mediciones a distancia como ser codificadores *PCM*, transmisores, etc. se adquirían exclusivamente en el exterior. En estos últimos años se hizo muy compleja la operación de importar estos dispositivos denominados “sensibles”, ya que además del elevado costo de los mismos se requiere de los respectivos permisos de los países exportadores (ej.: EEUU) para contar con estos dispositivos y poder utilizarlos en nuestras aplicaciones.

Es por este motivo que CITEDEF tomó la decisión de diseñar y desarrollar sus propios dispositivos para llevar a cabo las telemetrías de los diferentes proyectos, logrando un mayor dominio de la tecnología asociada, hecho que fue un excelente comienzo para poder independizarse a futuro de la compra de dichos productos en el exterior.

Aquí se podrá tener un acercamiento a los pasos seguidos por la División Telemetría para poder lograr un Codificador *PCM* que si bien tiene menores prestaciones que las que pueden proveer los sistemas existentes en el mercado

internacional, sus características de confiabilidad, costo y capacidad de reconfiguración lo hacen un componente indispensable en aplicaciones del ámbito civil como así también del militar.

En un sistema de Telemetría, a menos que la implementación de las mediciones de los parámetros a sensar sea demasiado simple, hay que encontrar una manera de transmitir simultáneamente varios canales que contengan la información provista por los sensores de abordo.

Las mediciones deberán transmitirse en uno de los siguientes formatos: como datos analógicos o digitales. Las telemetrías tienen una larga historia de transmisión de datos en el formato analógico original mediante la técnica de multiplexación por división de frecuencias (*FDM*).

Con el advenimiento de la computadora basada en el procesamiento masivo de la información, los datos digitales por lo general son considerados más útiles para el procesamiento de señales (ausencia de distorsión en la transmisión, mayor inmunidad al ruido externo y capacidad de detección y corrección de errores). El proceso de muestreo y conversión de una señal analógica a una representación digital y posterior codificación seguida de la transmisión de dichos datos se llama modulación por código de pulsos (*PCM*).

Mediante una modulación *PCM* un valor digital codificado de la señal medida es transmitido al receptor. Cuando las mediciones son más complejas y la transmisión de la información proviene de sensores múltiples, hay dos métodos básicos de multiplexación: dividir el canal de transmisión en franjas, en el dominio del tiempo (*TDM*) o en el dominio de la frecuencia (*FDM*) y asignar cada una de ellas la información del parámetro sensado.

Los sistemas *TDM* generalmente son utilizados en entornos *PCM* mientras que los *FDM* se utilizan en los sistemas de transmisión analógicos.

En cualquier sistema *TDM* el tema a tener en cuenta es la sincronización que debe haber entre transmisor y receptor, para poder decodificar correctamente la información que tiene cada canal.

PCM es una extensión de *PAM* (modulación por amplitud de pulso), donde cada valor de muestra de un canal analógico es cuantizado en un valor discreto para representarlo como una palabra de código digital [1].

II. DESARROLLO

Para detallar las cuestiones que tienen que ver con el codificador desarrollado partiremos de la siguiente figura:

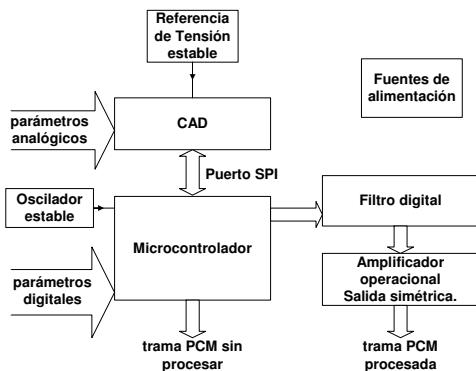


Figura 1. Diagrama en bloques del Codificador.

En el Codificador *PCM* el encargado de controlar y ejecutar las acciones es un microcontrolador.

Dicho Codificador es capaz de recibir señales digitales o analógicas. El microcontrolador, con una estructura interna basada en registros y *buses* de 8 bits, recibirá las muestras del sensado de las señales analógicas en un formato serie digital, ya que un conversor analógico-digital (CAD) se encargará de dicha tarea. Mientras que las señales digitales ingresan por los puertos paralelos propios del microcontrolador.

Luego el microcontrolador será el encargado de empaquetar cada una de las muestras y armar la trama *PCM* con las palabras de sincronismo correspondientes (Fig. 2).

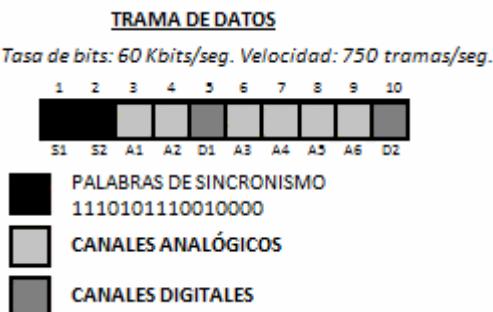


Figura 2. Ejemplo de una trama PCM.

El *CAD* es programable por medio de un puerto de interfaz periférica serie (*SPI*) y capaz de convertir 16 canales analógicos con una frecuencia de muestreo máxima de 1 MSPS y resolución máxima de 12 bits [3]. Esta basado en la técnica de cuantización por aproximación sucesiva (*SAR*).

Se tomó la decisión de utilizar un *CAD* externo (y no el propio del microcontrolador) por su mayor resolución de bits, linealidad y para no incrementar el tiempo de procesamiento al microcontrolador ya que la principal exigencia en esta aplicación es cumplir con tiempos de eventos propios de la aplicación en forma rigurosa, de lo contrario los sistemas

decodificadores de la telemetría en tierra no podrían sincronizarse con la señal transmitida. Es por este motivo que el diseño de código de programa del microcontrolador debió realizarse en lenguaje *assembler* ya que posee un exacto control de tiempos de ejecución (previamente se realizó dicho programa en lenguaje C y se produjeron dificultades).

El código puede ser dividido en dos partes, el cuerpo principal y una interrupción de alta prioridad. Dentro de la interrupción se va generando bit a bit el total de la trama *PCM* de salida (Fig. 3). Por lo tanto en esta rutina se debió realizar una compensación de tiempos para que todos los caminos por donde pudiera ir ejecutándose el programa tengan la misma demora y así, lograr un ancho de bit único.

No menos importante fue la implementación de un oscilador para el funcionamiento del microcontrolador de muy buena calidad y con una muy baja deriva por temperatura ($\pm 10 \text{ PPM}$ entre -20°C y 80°C). Esta última cualidad del oscilador fue comprobada prácticamente por ensayo en horno en el laboratorio de la División Ensayos Ambientales del Instituto.

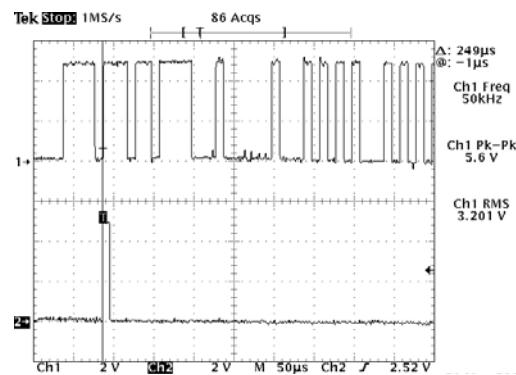


Figura 3. Trama PCM y bit de inicio de trama.

Como el Codificador *PCM* funciona en conjunto con otros dispositivos externos de los que se obtienen las correspondientes muestras, se provee para ellos una señal de inicio de trama (Fig. 3), una señal de inicio de palabra (Fig. 4) y una señal de bit presentes en el conector (Fig. 5) con el objetivo de sincronizar el conjunto y no realizar lecturas erróneas de las muestras de las señales que ingresan al Codificador.

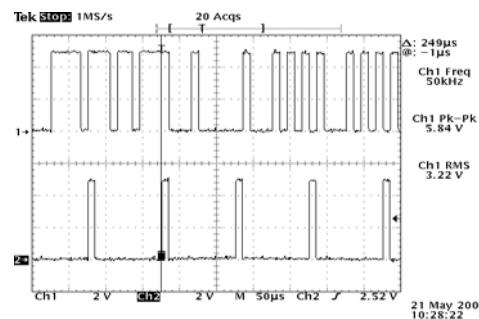
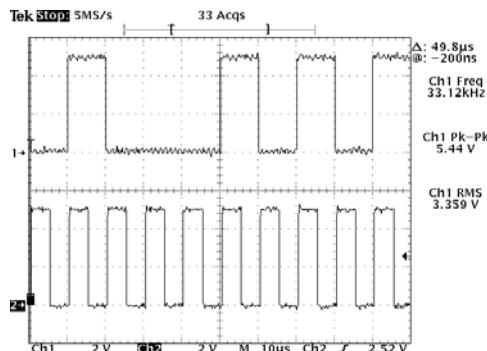


Figura 4. Señal de inicio de palabra.

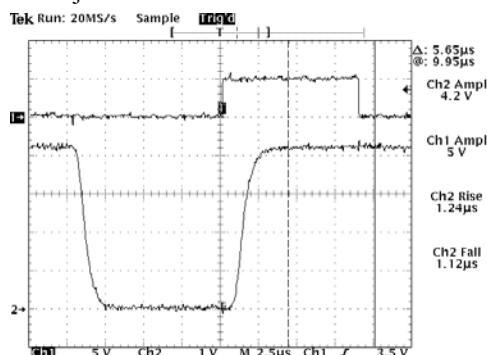


Como pudo verse en la Fig. 2 la trama *PCM* se inicia con las palabras de sincronismo, que al ser fijas y conocidas, le permitirán al sistema de recepción de telemetría reconocerlas para el posterior procesamiento de las muestras de parámetros posteriores a dicho sincronismo. El ejemplo (Fig. 2) muestra una posible configuración de la trama *PCM*, que tiene 10 palabras de longitud, cada una de ellas de 8 bits de longitud.

La definición de la palabra de sincronismo está determinada por la norma *IRIG 106* de Telemetría [2], cuya idea es obtener una palabra con la más baja probabilidad de ocurrencia en cualquiera de las palabras de información de parámetros, circunstancia que traería aparejado problemas en la posterior decodificación.

La formación de esta trama es configurable por software según el requerimiento (en los Proyectos GRADICOM 2009 y 2011 la cantidad de dispositivos sensores y por tanto la cantidad de parámetros a incluir en la trama fue diferente Fig. 11 y Fig. 12). Por lo tanto se puede diseñar una trama de datos con diferente cantidad de canales analógicos y digitales (de longitud variable en cuanto a cantidad de bits), cada uno de los cuales podrá ser sub-muestreado o sobre-muestreado según la importancia y el ancho de banda de la señal en cuestión. También es variable la tasa de bits dentro de ciertos límites.

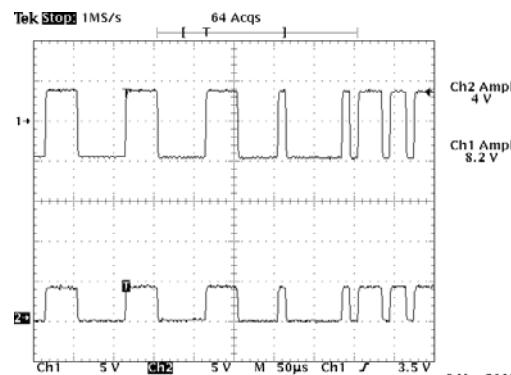
Una de las dos salidas de datos disponibles será acondicionada con un filtro digital (Fig. 6) y un amplificador de simetría con el fin de excitar al transmisor de RF de la carga telemétrica alojada en el cohete.



Se eligió según Norma *IRIG 106* un filtro de tipo elíptico de 6 polos [5], cuya frecuencia de corte va a estar definida por:

$$F_c = 0.7 * \text{Tasa de bits.} \quad (1)$$

Una vez filtrada la señal ingresa a un amplificador operacional de gran ancho de banda, que le da características de simetría y niveles de amplitud adecuados (Fig. 7).



Este dispositivo posee características de poder ser programado de manera serial y estando conectado al circuito de aplicación (*JCSPI*). Por este motivo se debió tener en cuenta el acceso a los pines de programación desde el exterior del Codificador.

No se debe olvidar en el diseño, que el Codificador *PCM* no es un dispositivo de funcionamiento aislado, sino que forma parte de un sistema y de una estrategia de transmisión y recepción de datos, en donde hay una íntima relación entre las partes que lo conforman. Ej.: el nivel proporcionado por el amplificador operacional de salida tiene que ver con el índice de modulación del transmisor de frecuencia modulada de la carga telemétrica, que a su vez tendrá que ser compatible con la desviación seteada en el demodulador del Receptor de Telemetría en tierra.

Este trabajo abarcó varias capacidades presentes en la División Telemetría como ser, diseño de circuitos electrónicos, desarrollo de software, simulaciones, diseño de circuitos impresos, filtrado digital de señales y desarrollo mecánico compatible con normas militares, cuyas consideraciones especiales en los respectivos diseños prevén que la aplicación es de alta exigencia y los conjuntos electrónicos para ésta tienen que superar los ensayos pertinentes para cumplir con normas *MIL-STD-810A* [4].

Esta norma militar detalla el procedimiento a realizar según sea la clase de elemento, el método de montaje usado y cuales de todos los tipos de ensayos son los que se van a practicar.



El Codificador PCM se incluye en la clase de equipamiento que se va a montar en un misil (para nuestro caso ‘cohete’) lanzado desde tierra y sin ningún tipo de aislación vibratoria.

El ensayo de mayor interés, además del de vacío y temperatura, fue el de vibración ya que es el más aplicable a la exigencia de cohetería (Fig. 13). El lugar donde se realizó fue el Laboratorio de Ensayos Ambientales de CITEDEF que tiene un sistema de vibración controlado permitiendo mediante software ingresarle el tipo de curva de ensayo que cumple con la secuencias de niveles de aceleración y frecuencia indicada.

En cuanto al diseño del circuito electrónico y su correspondiente impreso (Fig. 8) se tuvieron algunas de las siguientes consideraciones:

- Dimensiones lo más reducidas posibles ya que en una carga telemétrica hay poco espacio y debe además, compartirse con los demás dispositivos de sensado, alimentación, etc.
- Placa de impreso doble faz.
- Alimentación diferenciada para circuitos analógicos y digitales.
- Máxima aislación electromagnética entre circuitos digitales y analógicos.
- Puntos de testeo para ir verificando el funcionamiento sectorizado en el armado realizado por etapas.
- Separar además el circuito en dos categorías: baja y alta velocidad.
- Considerar sub-categorías. digital puro / analógico puro / señales mezcladas.
- Análisis de integridad de señal por *software*. Pines de los componentes que transportan señales de alta frecuencia (clocks, puertos digitales, etc.) lo más próximos al conector de salida (longitudes menores a los 30 mm.) para disminuir efecto de señales reflejadas.
- Evitar cruzar pistas que transportan señales analógicas con las que transportan señales digitales en capas opuestas.
- Planos de tierra con máxima cobertura de superficie en ambas capas.
- Planos de alimentación con capacitores de bypass a través del recorrido de la pista y a la entrada de los integrados.
- Topología de las pistas de tensión de alimentación tipo estrella, impedancia de fuente mínima.
- Pistas de señales analógicas menores a 30 mm. de longitud y espesor máximo de 0.381 mm. con el fin de minimizar fenómenos de auto inductancia.
- Pistas de señal de datos con guarda de plano de tierra.
- Capacitores de bypass de menor capacidad (cerámico) más próximo al regulador de tensión con el fin de evitar auto oscilaciones.
- Entradas analógicas con máxima distancia entre pistas (evita inductancia mutua).
- Filtro tipo pi en la entrada de alimentación.

- Uso de dos conectores *DB25* para entradas digitales, analógicas, señales de sincronización, tramas *PCM* con o sin procesamiento y alimentación (Fig. 9). Uso de pines paralelos para incrementar el tiempo medio entre fallas (*MTBF*) en contactos.
- Pines de programación del microcontrolador accesibles por uno de los conectores *DB25*.
- Ajuste del nivel de salida de la trama *PCM* disponible por hardware desde el exterior.
- Considerar que el amplificador de salida cumpla con los requerimientos de una salida tipo *TTL* sobre una resistencia de carga de 100 ohms, tanto para la salida *PCM* como para las señales de sincronismo.



Figura 8. Vista del circuito impreso (capa superior).



Figura 9. Placa armada con componentes de montaje superficial.

A la placa de la Fig. 9 se le aplicaron productos aislantes con base acrílica o de resina Epoxi, que a su vez le dan una mayor rigidez mecánica para soportar las exigencias de normas militares mencionadas.

Otra área que requirió diseño fue la mecánica, ya que la placa debe funcionar en un entorno que puede producir efectos indeseados por interferencias electromagnéticas. Por lo tanto se diseñó mediante software de CAD mecánico una caja construida en una aleación maquinable de aluminio con tratamiento superficial de níquel (Fig. 10).



III. RESULTADOS OBTENIDOS

TABLE I. ESPECIFICACIONES TÉCNICAS (GRADICOM 2009).

Nº de tramas por segundo	500	
Cantidad de palabras	20	
Longitud de palabras	10 bits	10µs/bit
Sincronismo	2 palabras de sincronismo de 10 bits c/u, según Norma IRIG 106	111011011100010 0000b (EDE20h)
Canales analógicos	16	Resolución 10 bits
Canales digitales	2 x 10 bits c/u	
Velocidad de datos	100 Kbits / segundo	
Alimentación	12 volts 60 mA.	
1 ^{ra} salida de datos	NRZ-L niveles TTL sin filtrar	
2 ^{da} salida de datos (filtrada)	NRZ-L niveles +/- 1 volt sobre una R ≥ 100 ohms.	Filtro de fase lineal. frecuencia de corte = 0,7 * bit rate
Estructura de la palabra	El MSB primero.	
Clock de trama	TTL	Con una carga R≥100Ω
Clock de bit	TTL (test point)	Lectura por flanco descendente en medio del bit.
Clock de palabra	TTL (test point)	
Excusión de entrada canales analógicos	0 a 5 volts	
Entrada canales digitales	TTL (0 y 5 volts)	Sin conectar, salida no especificada.
Rango de nivel de parámetros de entrada máximo sin peligro de daño a las partes.	- 1,47 volts a 6,47 volts	Imáx. de entrada diodo del clamping 10 mA.
Protección canales analógicos y digitales.	Diodos de clamping rápidos + resistencia de 82 Ω limitadora de corriente	
Impedancia de entrada canales analógicos	482 Ω.-34 pF	
Máxima impedancia de fuente analógica	200 Ω	
Corriente máxima de entrada canales digitales	Imáx = 25 mA.	
Drift por T° Oscilador	10 ppm máx.	Δ F / F0 (entre -20 y 80 ° C)
Linealidad	± 1 LSB desde la línea recta ideal de funcionamiento.	
Cross-talk	- 82 dB típico	
Conectores de conexión	2 DB25 hembra. Marca ITT	
Ensayos realizados	vibración, vacío y temperatura según Norma MIL-STD-810A	✓



Figura 10. Producto final.

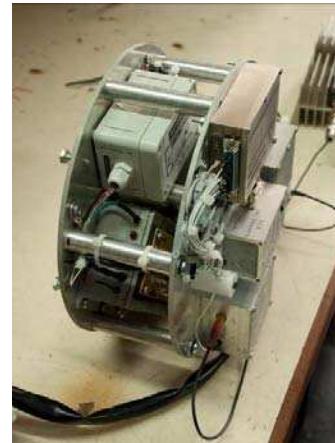


Figura 11. Carga telemétrica cohete GRADICOM 2009. Codificador PCM en piso superior.



Figura 12. Carga telemétrica cohete GRADICOM 2011. Codificador PCM en el tercer piso.



Figura 13. Lanzamiento GRADICOM 2011. Uno de los momentos más exigentes en cuanto al nivel de aceleración a soportar por el montaje electrónico.

IV. FUTURAS MEJORAS

En base a la experiencia acumulada por la División Telemetría en el desarrollo de Codificadores PCM se decidió intentar dar otro paso adelante, pero esta vez, más ambicioso. Acercarse más a la prestación de los codificadores del mercado internacional en cuanto a la tasa de bits y capacidades en lo que se refiere a conectividad.

Por tal motivo se presentó la propuesta de realizar un PIDDEF denominado “Diseño de un Codificador PCM de altas prestaciones” (tasa de bits=10 Mb/s), el cual en este momento se encuentra esperando la aprobación de las autoridades correspondientes.

En caso de ser aprobado es intención de la División Telemetría integrar al proyecto estudiantes avanzados o ingenieros recién recibidos de universidades nacionales ya que la complejidad del proyecto requiere la dedicación de un grupo de trabajo coordinado para obtener un Codificador que cumpla con las normas estandarizadas internacionales. Este hecho favorecerá además un ámbito multidisciplinario desde el punto de vista tecnológico, para que los estudiantes puedan realizar sus tesis o desarrollarse profesionalmente.

V. CONCLUSIONES

En cuanto a lenguajes de programación se refiere el C es uno de los preferidos ya que al ser de más alto nivel permite olvidar consideraciones de detalle, que ya vienen resueltas por el uso de funciones ya desarrolladas por otros programadores. Sin embargo, uno pierde precisiones en cuanto a tiempos de procesamientos que pueden verse reflejados en el no cumplimiento de los estándares, haciendo que no pueda integrarse el desarrollo con otros dispositivos internacionales que si los cumplen. Es por este motivo que el lenguaje assembler fue el elegido para programar el microcontrolador.

Cuando se intenta realizar un desarrollo tecnológico propio pero cumpliendo con normas estandarizadas internacionales, el proyecto que parecía sencillo se vuelve complejo.

Dichas normas limitan las variables puestas en juego a la hora de idear las características del dispositivo final. Además debe haber un estudio profundo de los ensayos, calibraciones y mediciones a realizar.

A esto nos referimos cuando hablamos de “un ámbito favorable desde el punto de vista tecnológico” ya que el cumplimiento de los estándares internacionales trae aparejado un enriquecimiento y profundización de los conocimientos inherentes al proyecto que especializan efectivamente al profesional o al estudiante avanzado.

A nuestro entender consideraremos que para lograr independizarse tecnológicamente de las empresas internacionales y generar desarrollos propios es necesario comenzar por segmentar los sistemas (ej.: Telemetría de un cohete), estudiar, ensayar y profundizar sobre uno de los elementos en cuestión (ej.: Codificador PCM) y luego diseñar el propio. Es allí donde surgirán los obstáculos teórico-prácticos que mediante esfuerzo y constancia podrán ser solucionados, lo que terminará incrementando la especialización de los diseñadores y desarrolladores y generando una verdadera independencia tecnológica puntualmente en dicho elemento integrante de un sistema.

Así continuando con esta metodología se podrá llegar en un futuro (mediano plazo), a tener la capacidad de diseñar un “Sistema Propio” en su totalidad.

REFERENCIAS

- [1] Stephen Horan, “Introduction to PCM telemetering systems” 2nd. Edition, 2002.
- [2] Telemetry Standard. IRIG 106.
- [3] Application Note AN-877 “Interfacing to high speed ADCs via SPI”.
- [4] Military standard environment test methods for Aerospace and Ground Equipments. MIL-STD-810 A.
- [5] D. Schlichthärle “Digital Filters. Basics and Design”. 2nd Edition, 2010.

Quadpector: un UAV de bajo costo para observación remota

Fernando Chouza, Michael Erusalimsky, Enzo Lanzelotti, Alberto Rodriguez

Universidad de Buenos Aires (UBA)

Buenos Aires, Argentina

fchouza@gmail.com

Resumen — Este trabajo describe el diseño y la construcción del Quadpector. El Quadpector es un vehículo aéreo no tripulado (*UAV*) de aterrizaje y despegue vertical (*VTOL*). Estas dos características lo hacen especialmente útil para su uso en operaciones de búsqueda y rescate, monitoreo de incendios, fotografía aérea y medición de parámetros atmosféricos, entre otras cosas. La propulsión se realiza mediante cuatro hélices, lo cual le da a esta clase de dispositivos el nombre de cuadricópteros. La sencillez mecánica en comparación con los helicópteros, transforma a los cuadricópteros en dispositivos altamente confiables y de bajo mantenimiento. A pesar de encontrarse todavía en proceso de desarrollo, las pruebas realizadas con el prototipo construido, tanto en un banco de pruebas como en vuelo libre, fueron satisfactorias. Por último, cabe destacar el bajo costo de construcción y mantenimiento de esta solución si se la compara con otras soluciones similares existentes.

Palabras clave – cuadricóptero; UAV; observación remota; filtro complementario.

I. INTRODUCCIÓN

Se presenta en este trabajo el proceso de diseño y construcción del Quadpector, así como mediciones que verifican el adecuado funcionamiento de los sistemas de control. El Quadpector es un vehículo aéreo no tripulado (*UAV*) capaz de ser comandado a distancia con el objeto de obtener imágenes, o cualquier otro tipo de datos capturados mediante sus sensores, en tiempo real. Asimismo, la capacidad de este tipo de vehículos de despegar y aterrizar de forma vertical, facilita su uso en zonas urbanas en donde un avión no podría operar. Estas dos características transforman al Quadpector en un vehículo ideal para asistencia en operaciones de búsqueda y rescate, monitoreo de incendios, fotografía aérea, medición de parámetros atmosféricos, vigilancia, monitoreo de lugares poco accesibles a las personas, y todo tipo de tareas en las cuales un vehículo tripulado representa un peligro muy elevado para el piloto o posee un tamaño demasiado grande para operar en el lugar [1][2]. Si bien existen otros vehículos, como los helicópteros, que pueden reunir estas características, existen algunas ventajas sobre estos [3].

Una de las principales ventajas de los cuadricópteros sobre los helicópteros es su simplicidad mecánica. Los helicópteros necesitan variar el ángulo de incidencia de las

palas del rotor de acuerdo a la posición del mismo para modificar su orientación. Esto se logra mediante un complejo sistema mecánico. El cuadricóptero, en cambio, utiliza hélices de incidencia fija (como las utilizadas en aviones) y basa la modificación de su orientación en la variación del empuje ejercido por cada una de las cuatro hélices. De este modo se logra un sistema con una menor cantidad de partes móviles, más confiable y con menor mantenimiento.

El diseño de este sistema fue realizado en el marco de la materia Trabajo Profesional de la FIUBA [4]. Los principales requisitos planteados fueron:

- Carga útil > 1 kg.
- Autonomía de vuelo > 10 minutos.
- Costo competitivo (< 1500 u\$s).
- Control a distancia
- Obtención de datos desde el cuadricóptero en tiempo real

La determinación de los mismos se realizó en base al análisis de productos que buscan satisfacer necesidades similares, como el MK Quadro XL o el GUAI 500X RTF.

El trabajo se divide en cinco secciones. La presente introducción, un breve resumen de los principios de funcionamiento, una explicación de la implementación adoptada para cada uno de los subsistemas que componen el Quadpector, un resumen de las mediciones realizadas en banco de pruebas y vuelo libre, y por último, las conclusiones que se desprenden del trabajo.

II. PRINCIPIOS DE FUNCIONAMIENTO

Los cuadricópteros poseen dos juegos de hélices contrarrotativas de incidencia fija. Las mismas se ubican sobre una estructura en forma de cruz, tal como se muestra en la Figura 1.

Modificando el torque aplicado a cada una de las hélices, se controla la orientación (ángulos de *pitch*, *roll* y *yaw*) del cuadricóptero respecto a una copia del sistema de referencia de navegación con su origen trasladado al centro de masa y la altitud respecto al sistema de referencia de navegación. Suponiendo que la hélice 1 corresponde a la parte frontal del

vehículo, una modificación en el ángulo *pitch* se logra modificando el torque relativo entre las hélices 1 y 3.

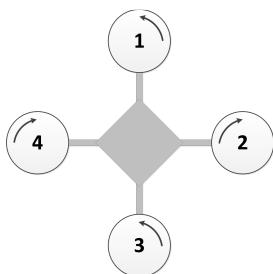


Figura 1. Ubicación y dirección de giro de las hélices

El ángulo de *roll* se modifica variando el torque relativo entre las hélices 2 y 4. El ángulo *yaw* se controla variando la velocidad relativa de las hélices 1 y 3, respecto a las 2 y 4, manteniendo constante el empuje total. Para modificar la altitud, basta con modificar el torque aplicado a las cuatro hélices simultáneamente [5].

De lo anteriormente expuesto se desprende que, para una adecuada estabilización del cuadricóptero, es necesario disponer de un control más rápido y preciso en las velocidades de giro de cada una de las hélices que lo necesario en otros dispositivos voladores, como aviones y helicópteros. El avance en los sistemas embebidos, en la tecnología de baterías y en la miniaturización de motores ha dado un gran empuje al desarrollo de esta tipo de vehículos en sus variantes no tripuladas.

III. CONSTRUCCIÓN DEL DISPOSITIVO

A. Estructura mecánica

La estructura del Quadpector está construida en aluminio y plástico. La distancia entre punta y punta de la cruz es 0,8 m, y la altura respecto del suelo es de 25 cm, de modo de permitir la sujeción de diversos sistemas. En este caso particular, una cámara IP. El peso total de la misma es de 0,9 kg. Para las primeras pruebas se agregó a la estructura una segunda estructura de protección para evitar que un eventual choque lateral rompa alguna de las hélices.

El peso total del cuadricóptero, obtenido en base al peso de la estructura, los sistemas de propulsión y la aviónica es igual a 2,1 kg.

B. Alimentación

La alimentación del sistema de propulsión está a cargo de cuatro baterías marca Zippy de tipo LiPo de 2200 mAh y 11,1V. La aviónica del Quadpector controla la tensión de cada una de las baterías a fin de verificar la carga restante en las mismas y controlar que la tensión no caiga por debajo del nivel mínimo admitido para este tipo de baterías (3V por celda). Cada batería tiene un peso de 180 g. Si bien aún no se ha implementado la función, en caso de llegar al nivel de carga mínimo admitido, el sistema emprenderá un aterrizaje

automático para evitar que los motores se detengan en pleno vuelo por el agotamiento de las baterías.

C. Propulsión

Los cuatro motores que proveen el empuje al Quadpector son de tipo *brushless outrunner*, modelo Turnigy 2217-16T. Este tipo de motores son altamente eficientes sobre un amplio rango de velocidades, esto permite prescindir de una caja reductora. Otra gran ventaja, si se lo compara con los motores de tipo *brushed*, es el bajo mantenimiento necesario. El peso del motor seleccionado es 71 g.

Los motores de tipo brushless necesitan de un controlador para su funcionamiento. El controlador utilizado es un Turnigy Plush 30 A, el mismo posee un peso de 25 g. La elección de un controlador adecuado es muy importante, porque determinará la velocidad de respuesta del sistema. La mayoría de los controladores disponibles en el mercado poseen una frecuencia de actualización de 50 Hz debido a que el protocolo de comunicación utilizado se basa en una señal de período constante e igual a 20 ms, con un ciclo de trabajo variable entre 1 y 2 ms [6]. Esta frecuencia de actualización resulta insuficiente para una adecuada estabilización de un sistema de estas características [7][8]. El controlador utilizado es capaz de aceptar frecuencias de actualización mayores. Se determinó, mediante mediciones, que la misma puede superar los 200 Hz, transformándolo en una opción adecuada para esta solución.

Estos motores mueven dos juegos de hélices contrarrotativas, tipo GWS 10x6". El empuje máximo de cada motor junto con esta hélice es 1,1 kg, con una corriente de 19,6 A aproximadamente. La relación entre consumo, empuje y eficiencia que se presenta en la Figura 2 fue obtenida en base a bases de datos disponibles en la red [9], y posteriormente verificada.

Cada conjunto de motor, hélice, controlador y batería tiene un peso de 276 g, totalizando 1,1 kg para los cuatro conjuntos. El empuje máximo total que puede proveer el sistema de propulsión en su conjunto es de 4,4 kg a máxima potencia. La diferencia entre el empuje máximo y el peso nos permite determinar la carga útil máxima, en este caso, igual a 2,3 kg. Actualmente la carga del Quadpector consiste en una cámara IP de aproximadamente 0,5 kg.

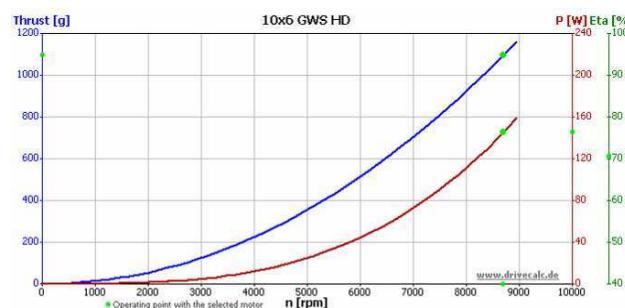


Figura 2. Características del conjunto de propulsión utilizado

D. Aviónica

La aviónica del Quadpector se compone de un conjunto de sensores, un módulo de comunicación, y un microcontrolador que realiza el procesamiento de datos y envía las señales de control al sistema de propulsión. En la Figura 3, se muestra un diagrama en bloques del Quadpector.

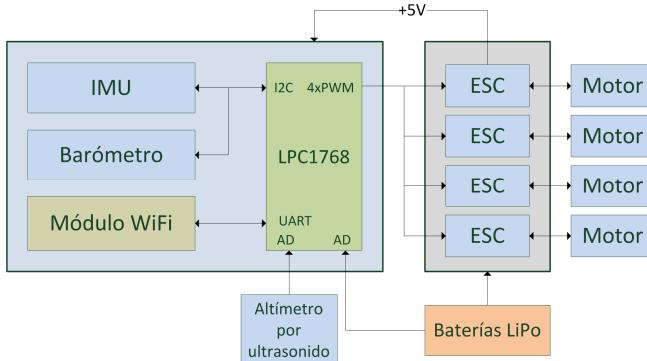


Figura 3. Diagrama en bloques de la aviónica

El conjunto de sensores está formado por una IMU (*Inertial Measurement Unit*), un barómetro BMP085 y un sensor de distancia ultrasónico XL-Maxsonar EZ3.

La IMU incluye un acelerómetro ADXL345 de 3 ejes, un giróscopo ITG3200 de 3 ejes y un magnetómetro HMC5883L. Los mismos se comunican con el microcontrolador por medio de un bus I2C funcionando a 100 kHz. Combinando los datos obtenidos por medio de estos sensores, es posible determinar la orientación espacial del cuadricóptero.

La comunicación con la estación de control se realiza por medio de una red WiFi. Para ello, el cuadricóptero dispone de un módulo de comunicación RN-XV WiFly que opera con protocolo UDP para mantener un caudal constante de datos de control, evitando demoras causadas por el reenvío de paquetes perdidos. El mismo se comunica con el microcontrolador mediante una UART operando a 57600 baud.

El microcontrolador utilizado es un LPC1768 corriendo a 100MHz.

El peso de la aviónica totaliza 154g. Distribuidos del siguiente modo: IMU 4g, RN-XV WiFly 20g, BMP085 5g, XL-Maxsonar EZ3 5g, LPC1768 50g, placa principal 70 g.

E. Software

Para poder estabilizar adecuadamente al cuadricóptero, es necesario poder determinar con precisión la orientación del cuadricóptero en cada momento. La orientación del mismo se obtiene en base a las mediciones obtenidas por los sensores de la IMU.

Existen dos sistemas de referencia importantes al momento de analizar la navegación del Quadpector. Uno no inercial, fijo a la estructura del cuadricóptero (SRE) con su origen ubicado en el centro de gravedad de la nave, los ejes x e y alineados con los brazos, y el eje z apuntando hacia

abajo. El otro, llamado sistema de referencia de navegación (SRN), se encuentra fijo a la tierra siguiendo la orientación del cuadricóptero antes de despegar. Debido a que los tiempos de vuelo y distancias son pequeñas, puede suponerse a la tierra como plana y estacionaria y, por lo tanto, al SRN como inercial ($\omega_n^{inercial} = 0$). Ambos se muestran esquemáticamente en la Figura 4.

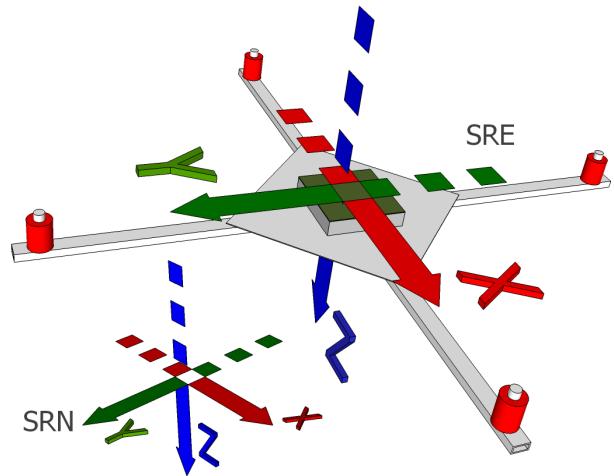


Figura 4. Sistemas de referencia utilizados en la navegación

Puesto que la IMU se encuentra sujetada de forma rígida a la estructura del cuadricóptero, los valores que esta entrega en cada instante se hallan referenciados a esta estructura. Sin embargo, nuestro interés es conocer la orientación respecto al sistema de referencia de navegación. Para ello, debemos realizar una transformación entre estos dos sistemas de referencia. Esto puede llevarse adelante mediante el uso de una DCM (Direction Cosine Matrix).

La DCM, expresada como $C_b^n(t)$, está compuesta por las proyecciones de los vectores unitarios del sistema de referencia fijo a la estructura sobre el sistema de referencia de navegación. Es posible propagar la DCM a través del tiempo mediante una serie de expresiones ampliamente utilizadas en el ámbito de los sistemas de navegación [10]. El resultado de aplicar estas ecuaciones a nuestro caso se observan en (1).

$$C_b^n(t + \Delta t) \approx C_b^n(t) \cdot \begin{bmatrix} 1 & -\omega_z \Delta t & \omega_y \Delta t \\ \omega_z \Delta t & 1 & -\omega_x \Delta t \\ -\omega_y \Delta t & \omega_x \Delta t & 1 \end{bmatrix} \quad (1)$$

Los valores $\omega_x, \omega_y, \omega_z$ son las velocidades angulares medidas por el giróscopo sobre los tres ejes de la estructura del cuadricóptero, mientras que el intervalo de tiempo Δt es el existente entre mediciones. En el caso del Quadpector, el tiempo entre muestras es 4ms.

El sistema se inicializa con $C_b^n(0) = I$, y se propaga a lo largo del tiempo aplicando (1). Una vez obtenida la DCM correspondiente a la posición del cuadricóptero, se pueden extraer de la misma los ángulos de Euler que representan la orientación de la estructura respecto al sistema de referencia de navegación, mediante (1), (2), (3) y (4):

$$\phi_{giro} = \arctan\left(\frac{C_b^n[3,2]}{C_b^n[3,3]}\right) \quad (2)$$

$$\theta_{giro} = -\arcsin(C_b^n[3,1]) \quad (3)$$

$$\psi_{giro} = \arctan 2\left(\frac{C_b^n[2,1]}{C_b^n[1,1]}\right) \quad (4)$$

Hasta este punto, solo se utilizan los valores obtenidos por el giróscopo. El problema de esta solución reside en la elevada deriva que poseen las mediciones angulares basadas en la integración de la velocidad angular. Si bien esta desviación es pequeña en intervalos cortos, se hace intolerable a medida que transcurre el tiempo. Esto hace necesario el uso de otros sensores que permitan cancelar este efecto. En contrapartida, una medición basada en el acelerómetro posee un elevado ruido, pero poseen una deriva varios órdenes de magnitud inferior [11].

En resumen, se necesita complementar ambos tipos de mediciones. En intervalos de tiempo corto, la medición basada en el giróscopo posee mayor precisión, mientras que la medición del acelerómetro es más confiable a largo plazo. El Quadpector solo compensa la deriva de los ángulos de *pitch* y *roll* mediante el uso del acelerómetro. La compensación del ángulo de *yaw*, puesto que no es crítica, no ha sido realizada hasta el momento. En versiones futuras se implementará de modo similar, pero en base a las mediciones obtenidas por el magnetómetro. La fusión de mediciones se realiza a través de un filtro complementario [12] [13].

Es posible obtener una aproximación de los ángulos de *pitch* y *roll* mediante una simple descomposición trigonométrica de la lectura correspondiente a la aceleración de la gravedad en los ejes X e Y del acelerómetro. Debido a que ambos ángulos son pequeños, es posible realizar las aproximaciones (5) y (6).

$$\phi_a = -\arcsin(a_x) \approx -a_x \quad (5)$$

$$\theta_a = \arcsin(a_y) \approx a_y \quad (6)$$

Estas mediciones poseen un error elevado en el corto plazo, puesto que consideran que toda la lectura del acelerómetro corresponde a la aceleración de la gravedad, descartando la aceleración correspondiente al movimiento del Quadpector. Sin embargo, debido a que la mayor parte del tiempo el cuadricóptero se encuentra en equilibrio ($a_x = 0$ y $a_y = 0$), el efecto del movimiento es descartado por el filtro complementario.

El filtro complementario es implementado mediante (7), (8) y (9).

$$\phi = k\phi_g + (1-k)\phi_a \quad (7)$$

$$\theta = k\theta_g - (1-k)\theta_a \quad (8)$$

$$\psi = \psi_g \quad (9)$$

El valor de la constante k , se ajustó experimentalmente para minimizar el ruido del acelerómetro y la deriva introducida por las mediciones basadas en el giróscopo, obteniendo un valor $k=0.997$.

Una vez obtenidos los ángulos compensados por el filtro complementario, se reconstruye la matriz C_b^n y se pasa a la sección en la cual se implementa el control del sistema, en base a un algoritmo PID. Las señales de control se obtienen mediante (10), (11) y (12).

$$C_\phi = k_{p-\phi}(\phi - \phi_o) + k_{D-\phi} \cdot \omega_x + k_{I-\phi} \cdot \int (\phi - \phi_o) \quad (10)$$

$$C_\theta = k_{p-\theta}(\theta - \theta_o) + k_{D-\theta} \cdot \omega_y + k_{I-\theta} \cdot \int (\theta - \theta_o) \quad (11)$$

$$C_\psi = k_{p-\psi}(\psi - \psi_o) + k_{D-\psi} \cdot \omega_z \quad (12)$$

Nuevamente, las constantes fueron establecidas de modo experimental, haciendo uso de un banco de pruebas.

Las señales de control, previamente a ser aplicadas a los motores, son combinadas tal como lo indica 13, 14, 15 y 16:

$$T1 = (-2C_\theta - C_\psi + C_A)/4 \quad (13)$$

$$T2 = (2C_\phi + C_\psi + C_A)/4 \quad (14)$$

$$T3 = (2C_\theta - C_\psi + C_A)/4 \quad (15)$$

$$T4 = (-2C_\phi + C_\psi + C_A)/4 \quad (16)$$

Cada una de las variables que representan la velocidad de los motores, se encuentran en el rango comprendido entre 1 y -1. Esta variable es transformada en una señal PWM de 4ms de período, y un *duty cycle* variable entre 1ms y 2 ms. El primero de los valores corresponde al motor apagado y a la variable de control en -1, mientras que 2 ms corresponde al motor en máxima potencia y a la variable de control en 1.

La distribución de los motores se muestra en la Figura 1. Siendo la punta del Quadpector aquella en la que se sitúa en motor 1.

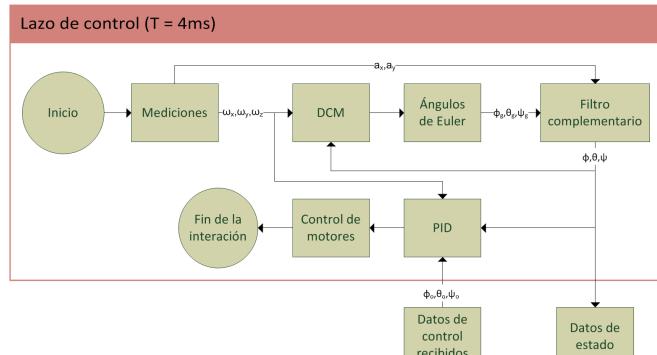


Figura 5. Diagrama en bloques del algoritmo de control

La señal de control denominada C_A , establece la potencia de referencia sobre la cual se aplicaran las señales de control propias del sistema de estabilización. Actualmente esta señal es controlada por el operador. En versiones futuras, será controlada en base a los datos obtenidos por el sensor ultrasónico y el barómetro a fin de establecer un control automático de altura.

El lazo de control se resume de forma esquemática en Figura 5.

F. Estructura de la red de comunicaciones

La conexión del Quadpsector con la estación de control se realiza por medio de una red WiFi. A esta red se conectarán el cuadricóptero, la estación de control y cualquier otro dispositivo que pueda llegar a ser necesario. En este primer prototipo, como se muestra en la Figura 6, se montó sobre el Quadpsector una cámara de video IP orientable. La principal ventaja de las redes WiFi, y la principal razón por la cual se utilizó este tipo de conexión, es la gran multiplicidad de dispositivos que pueden conectarse utilizando este medio. Gracias a esta elección, cualquier teléfono, netbook, notebook o PC puede ser utilizada, con un software apropiado, como estación de control.

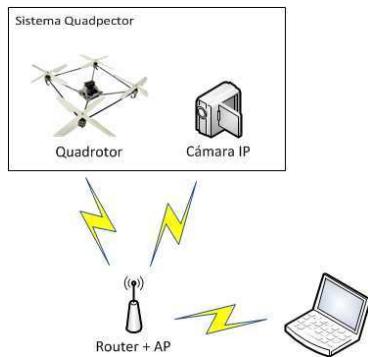


Figura 6. Diagrama del sistema de comunicación entre el Quadpsector y la estación de control

IV. RESULTADOS

En la presente sección se presentan resultados obtenidos en base al prototipo construido, el mismo puede observarse en la Figura 7. En primer lugar se presentan las pruebas realizadas bajo situación controlada, con el Quadpsector sujeto a un banco de pruebas. Luego, se muestran datos correspondientes a un primer vuelo libre realizado.

A. Medición en un banco de pruebas

Para probar el correcto funcionamiento del sistema de control, se realizaron pruebas sobre un banco de mediciones. El mismo permite el giro del cuadricóptero sobre un solo grado de libertad (*pitch* o *roll*). Las pruebas consistieron en fijar diferentes *set points*, con el objeto de ver la respuesta

del sistema de control ante cambios graduales y abruptos. Los resultados se pueden observar en las Figuras 8 y 9.



Figura 7. Prototipo construido

Puede verse que la respuesta ante variaciones abruptas es rápida y las variaciones entorno al *set point*, una vez alcanzado el punto de equilibrio, se mantenían en torno a $\pm 1^\circ$ aproximadamente.

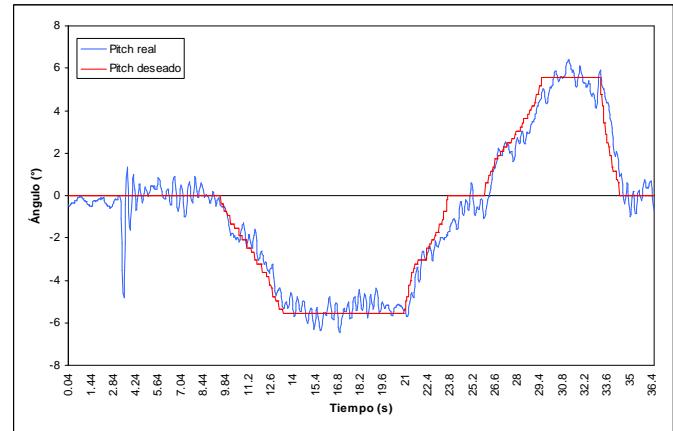


Figura 8. Variación del *pitch* para una señal de control suave en función del tiempo

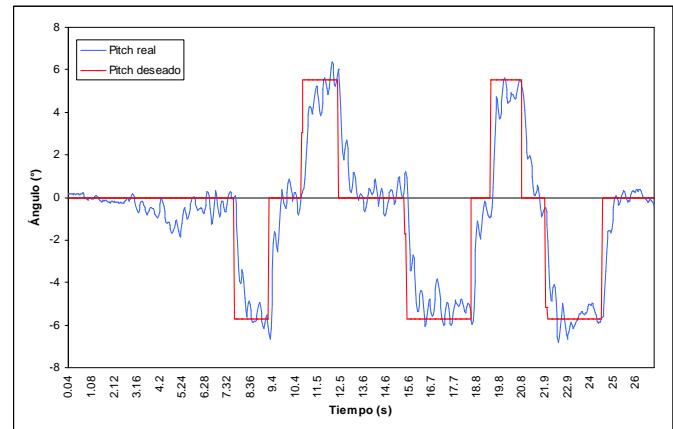


Figura 9. Variación del *pitch* para una señal de control abrupta en función del tiempo

B. Mediciones para vuelo libre

Se probó al Quadpector en exteriores, en condición de vuelo libre. Los resultados que se presentan en las Figuras 10 y 11, corresponden a los ángulos de *pitch* y *roll* durante el vuelo. Si bien el vuelo duró algo menos de 30 segundos, se pudo probar la estabilidad general del sistema.

Se observa en un tiempo cercano a los 13 segundos, una perturbación en la posición del Quadpector, esto fue producto de una fuerte ráfaga de viento. Puede verse al mismo tiempo, la rápida recuperación de la orientación comandada. Las lecturas posteriores a t=26 s, corresponden a datos posteriores al aterrizaje.

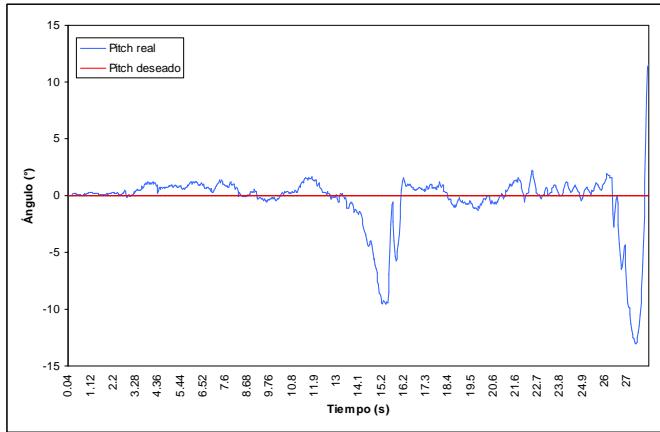


Figura 10. *Pitch* medido y objetivo para un vuelo libre

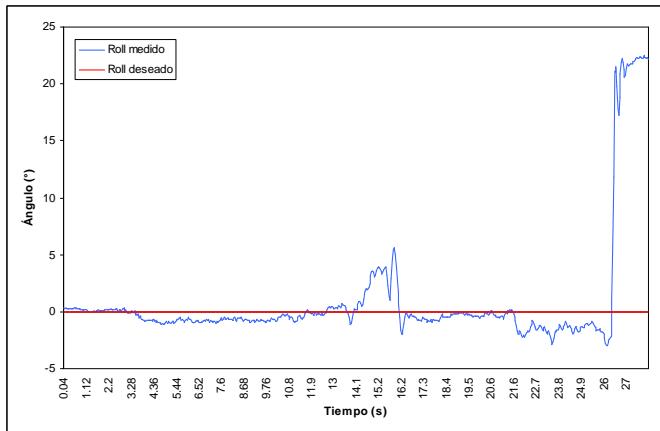


Figura 11. *Roll* medido y objetivo para un vuelo libre

En la Figura 12, se muestra una captura del Quadpector en pleno vuelo.

V. CONCLUSIONES

Si bien el Quadpector todavía se haya en proceso de desarrollo, se ha logrado que el mismo realice algunos vuelos básicos [14]. Resta por implementar un control de altura automático que permita mantener una altura constante, reduciendo de ese modo, el entrenamiento necesario por parte del usuario para el uso del sistema. Resta además realizar el algoritmo que permita el aterrizaje automático en

caso de pérdida de comunicación o agotamiento de las baterías.



Figura 12. Fotografía del Quadpector en vuelo

Asimismo, una futura versión agregará la capacidad de realizar un vuelo completamente autónomo basado en un recorrido programado utilizando la información entregada por un módulo GPS.

Se ha cumplido el requisito de costo establecido como criterio de diseño, las partes del sistema totalizan u\$s 650 (incluyendo la cámara IP), dando lugar a un sistema de observación remota económico y de alta flexibilidad.

REFERENCIAS

- [1] D. Greer, P. McKerrow y J. Abrantes. Robots in Urban Search and Rescue Operations. Australasian Conference on Robotics and Automation. Auckland, 2002.
- [2] Z. Sarris. Survey Of UAV Applications In Civil Markets (June 2001). Technical University of Crete, 2001.
- [3] S. Bouabdallah. Design And Control Of Quadrotors With Application To Autonomous Flying. EPFL. Lausanne, 2007.
- [4] 66.99 – Trabajo Profesional. 7 de Abril de 2012. <http://materias.fi.uba.ar/6699/>
- [5] M. J. Stepaniak. A Quadrotor Sensor Platform. Ohio University. Ohio, 2008, pp. 27-31.
- [6] How It Works — The PPM Radio Control System. The Model Electronics Company. Berkshire, 2001.
- [7] P. Pounds, R. Mahony y P. Corke. Modelling and Control of a Large Quadrotor Robot. Control Engineering Practice, 18(7), pp. 691-699. 2010.
- [8] William Etter y Paul Martin. AirHacks. 7 de Abril de 2012. http://airhacks.org/?page_id=219
- [9] Christian Persson. Drive Calculator. <http://www.drivecalc.de/>
- [10] Kenneth R. Britting. Inertial Navigation Systems Analysis. Wiley-Interscience, New York, 1971.
- [11] D. Bonner, G. Gede y J. Kim. MEMS Sensor Characterization.
- [12] Shane Colton. The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform. Mentor, FRC 97. 2007.
- [13] David Gaydou, Javier Redolfi y Agustín Henze. Filtro complementario para estimación de actitud aplicado al controlador embebido de un cuatrirrotor. CASE 2011 - Libro de trabajos. Buenos Aires, 2011.
- [14] <http://www.youtube.com/user/fchouza87>



Diseño e Implementación de un Nodo Compatible con 802.15.4 para Redes Inalámbricas de Sensores

Pablo Ridolfi¹, Sergio Scaglia², Ariel Lutenberg³, Pedro Martos⁴

Laboratorio de Sistemas Embebidos

Facultad de Ingeniería, UBA

Ciudad de Buenos Aires, Argentina

¹pridolfi@fi.uba.ar; ²sergio.scaglia@nxp.com;

³lse@fi.uba.ar; ⁴pmartos@fi.uba.ar

Fabian Vargas

Departamento de Ingeniería Eléctrica

Universidad Católica – PUCRS

Porto Alegre, Brasil

vargas@pucrs.br

Abstract—En los últimos años han tomado especial relevancia aquellos dispositivos electrónicos diseñados para tomar mediciones del medio que los rodea. Un enfoque clásico de esta temática implica que el sistema adquiere datos de sensores distribuidos en el medio, lo cual requiere en la mayoría de los casos un cableado especial que implica un alto coste de instalación. Al mismo tiempo, este enfoque requiere de un procesamiento centralizado, el cual complica aún más su implementación conforme aumenta el número de sensores o nodos a monitorear. Se presentará en este artículo el diseño e implementación de un nodo acorde con la norma de comunicaciones inalámbricas IEEE 802.15.4. Sus principales características son su tamaño reducido, el funcionamiento a batería de Li-ion, bajo consumo, bajo costo y capacidad de procesamiento *in situ*, utilizando un framework de software desarrollado especialmente para brindar servicios de red ad-hoc y preprocesamiento de los datos obtenidos para optimizar la estructura descentralizada de la red.

Keywords-IEEE802.15.4; wireless sensor networks; hardware implementation; low-power consumption; GPS modules

I. INTRODUCCIÓN

Una red inalámbrica de sensores (WSN – Wireless Sensor Network) [1] es un conjunto de dispositivos distribuidos espacialmente en una determinada región y con capacidad de realizar el sensado de una o varias señales relacionadas con información del entorno que los rodea, tal como temperatura, sonido, vibraciones, presión, contaminantes, etc. Estos dispositivos, además de contener una unidad de sensado, constan en general de una unidad de cómputo, una fuente de energía y un equipo transceptor de naturaleza inalámbrica [2]. En base a esta descripción general, existen varias formas de disposición o realización de los dispositivos en función de la finalidad última de la red. En particular, las topologías de tipo ad-hoc o multi-hop no son centralizadas, es decir que cada nodo puede reenviar un paquete de datos que no fue destinado a él. Esto habilita a los nodos que requieren información de otros que no están en su alcance, es decir que exceden el radio de alcance de la señal inalámbrica, a enviar el requerimiento a los vecinos hasta que se alcance al nodo interesado [3].

Como resultado de esta disposición el enlace de datos se lleva a cabo por la unidad de cómputo de cada nodo, lo que distribuye el procesamiento de la información y por lo tanto evita la necesidad de un enrutador o concentrador. La complejidad de este tipo de redes se ve aumentada en la

capacidad de procesamiento de los nodos, quienes deben ser capaces de funcionar como enrutador. Actualmente las tecnologías disponibles permiten trabajar en el firmware (programa de computadora embebido) de cada nodo a fin de brindarles tal habilidad.

En las secciones subsiguientes se desarrollarán las diferentes etapas del diseño e implementación de un nodo compatible con el protocolo de comunicaciones inalámbricas IEEE802.15.4. Se hará una comparativa de los existentes en el mercado y a partir de ello se definirán las características del hardware a implementar, así como el framework de software que brindará los servicios de red. Luego se detallará el diagrama esquemático del diseño implementado y su correspondiente circuito impreso, para finalmente hacer un análisis de consumo, autonomía y costos.

II. ESTADO DEL ARTE

Actualmente hay disponibles soluciones comerciales que implementan nodos 802.15.4. Las mismas están basadas en un transceiver que implementa la capa física del estándar, junto con un microcontrolador para implementar el resto de protocolos necesarios para establecer una red inalámbrica.

Estas soluciones comerciales se agrupan en dos grandes categorías: las que utilizan un procesador de 8 bits y las que utilizan un procesador de 32 bits. Los nodos que utilizan un procesador de 8 bits apuntan a implementar una solución de enlace inalámbrico de tal manera que el nodo se integre a una aplicación como una interface de transmisión de datos, a través de un protocolo serie (normalmente RS-232). En general utilizan protocolos inalámbricos propietarios o implementan un subconjunto de un estándar como ZigBee que no contempla la capacidad de ruteo “mesh”, es decir, que un nodo pueda ser simultáneamente generador de datos y “router” de otros nodos de la red [4]. Esto es debido a que casi toda la potencia de computo del procesador se emplea en la implementación de los protocolos inalámbricos.

Por otra parte los nodos que utilizan procesadores de 32 bits están orientados a soluciones en las cuales el nodo forma un System-On-Module (SoM) y sera el responsable del procesamiento en la aplicación. Estos nodos pueden implementar protocolos propietarios o un estándar como ZigBee en forma completa, por lo que son aptos para redes “mesh”. El uso de procesadores de 32 bits hace que haya disponible potencia de computo libre para ser utilizada en la aplicación. Estos nodos también disponen de mas interfaces, tales como GPIO (General Purpose Input Output), y mas memoria; lo que los hace aptos para implementar SoMs.

TABLA I. COMPARATIVA DE MÓDULOS 802.15.4

Fabricante	Modelo	Precio (US\$)	Transceiver	Procesador	RAM / Flash (kB)	Consumo Tx/Rx(mA)	Pout (dBm)	Sensibilidad (dBm)	Link Budget (dB)	Interfaces	Protocolos
<i>Nodos con arquitectura de 8-bits</i>											
Atmel	ATZB-24-B0	28	AT86RF230	Atmega1281	8/128	50/23	3	-101	104	UART,SPI,I2C	ZigBee,Propietarios
CEL	zicm2410p0	17	ZIC2410	8051	8/96	44/35	6	-97	103	UART,SPI	ZigBee,Propietarios
RadioCrafts	rc2400	N/A	CC2420	8051	256/8	24/34	20	-99	119	UART,SPI,Pines	ZigBee,Propietarios
RFMonolíticos	lpr2430	22	CC2430	8051	8/128	28/27	0	-92	92	UART,SPI	ZigBee,Propietarios
DIGI	Xbee	15	MC13193	HC08	4/60	45/50	0	-92	92	UART	ZigBee,Propietarios
<i>Nodos con arquitectura de 32-bits</i>											
CEL	z fsm-201	22	MC13224	ARM7	96/128	193/30	20	-94	114	UART,SPI	FreeScale BeeStack
NIVIS	versanode 210	N/A	MC13224	ARM7	96/128	60/21	10	-98	108	UART,SPI	HART, Propietario
RadioPulse	mg2410	N/A	MG2400	Cortex-M3	128/128	46/19	10	-100	110	UART	ZigBee
Panasonic	PAN4555	22	PAN1555	Cortex-M3	256/48	30/37	0	-92	92	UART	Propietario
Jennic	jn5148	24	JN5138	RISC Gen érico	128/128	15/17.5	2.5	-95	97.5	UART,SPI	ZigBee,Propietarios
FIUBA	gmot-1.0	38 ~ 91	CC2520/91	Cortex-M3	8/32	162/39	22	-104	126	UART,GPIO,...	Propietario

Cabe destacar que, si bien el uso de procesadores de 32 bits genera un mayor consumo en operación, las técnicas avanzadas de reducción de consumo que éstos implementan hace que su consumo de corriente promedio sea comparable al de los nodos que utilizan procesadores de 8 bits. Esto hace que el principal factor de consumo de los nodos sea la potencia de transmisión, la cual depende del transceiver y de la configuración del nodo. Asimismo, todos los nodos se alimentan a 3.6V, por ser una tensión estándar de baterías comerciales. La Tabla I resume las características de diferentes SoM comerciales [5]–[13], y al mismo tiempo se comparan con la implementación presentada en este artículo. Es importante destacar que nuestro diseño agrega nuevas funcionalidades que no se encuentran en los nodos restantes (ver Sección V.).

III. DISEÑO PROPUESTO

A. Hardware

Luego del análisis de los módulos existentes en el mercado, se definieron los aspectos fundamentales respecto al diseño del nodo:

- Debe ser programable. Esto permitirá experimentar con diferentes algoritmos y procesar los datos provistos por los sensores y otros periféricos antes de transmitirlos por la red.
- Debe disponer de un transceiver 802.15.4. Esto permitirá al procesador controlar todos los aspectos de la transmisión y recepción, como el tipo de modulación, selección de canal con menor ruido y potencia de transmisión y al mismo tiempo, teniendo en cuenta el perfil de energía deseado, deberá existir la posibilidad de desactivar el transceiver, aumentando la autonomía del dispositivo.
- Debe incluir sensores básicos. Se resolvió incluir en el nodo un receptor GPS, un sensor de temperatura y otro de luz ambiental, así como pulsadores y leds indicadores. Particularmente, la idea de incluir un receptor GPS radica en que una de las líneas de investigación del grupo de trabajo sobre WSN se centra en localización mediante RSSI (Received Signal Strength Indication, se refiere a la potencia de portadora medida al recibir un paquete). Este periférico permitirá comparar resultados del método de localización mencionado y aprovechar a los nodos que posean GPS como referencias de posición absoluta durante la experimentación.
- Debe funcionar en forma autónoma. Dispondrá de un circuito de carga compatible con baterías de Li-ion o

Li-pol a través de una interfaz USB para facilitar su utilización.

- Debe consumir la menor energía posible. Para ello el procesador es capaz de apagar secciones del hardware que no se están utilizando y dispone de modos de ahorro de energía.
- El procesador debe tener una buena relación costo/tamaño/ performance. Para esto se seleccionó la arquitectura de procesador que mejor se adapta a las necesidades y aplicaciones del hardware.
- Debe poseer conectores de expansión para incorporar otros sensores o periféricos.

B. Software

De acuerdo a los objetivos planteados para el nodo, la arquitectura del software contempla un framework consistente en un stack de comunicaciones, y módulos de Nombres y Direcciónamientos, Sincronización Horaria, Localización y Posicionamiento, y Calidad de Servicio (Fig. 1). A nivel de red, se implementan servicios “Centrados en Datos” y “Procesamiento en Red”, proveyendo interfaces adecuadas para su uso.

Se incluye en el framework la implementación de las distintas capas de protocolos, módulos y servicios necesarios:

- Capa física y de enlace de datos que comprende MAC (control de acceso al medio) y LLC (control de enlace lógico) [14]. En nuestro caso, el hardware seleccionado implementa el standard IEEE 802.15.4 el cual abarca a las capas físicas y MAC del modelo OSI de comunicaciones. Este modelo en particular, opera en la banda ISM (Industrial, Scientific and Medical) de 2.4GHz utilizando 16 canales (se debe seleccionar uno

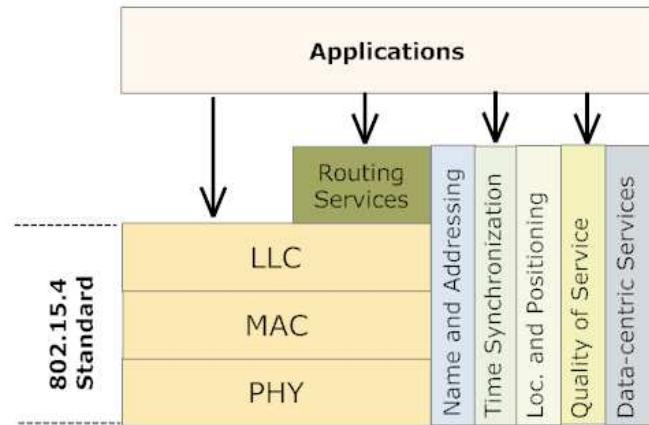


Figura 1. Arquitectura del framework de software.

de estos canales) bajo un esquema DSSS (Direct Sequence Spread Spectrum) de hasta 250Kbps de velocidad de transferencia [15]. El control de enlace lógico es provisto por los drivers del hardware seleccionado.

- Capa de red y servicio de ruteo basada en arquitectura multi-hop [16] por inundación, en la que cada nodo reenvía la información que no fue destinada para él. Si bien esta técnica es simple, resulta bastante ineficiente desde el punto de vista energético ya que todos los nodos de la red consumirán energía para retransmitir información redundante. Por eso se ha planificado implementar un modelo de ruteo alternativo basado en tablas de direccionamiento.
- Modelo de red “centrado en datos”, es decir que el requerimiento de información no se hace a un nodo en particular (para lo cual sería necesario conocer su dirección), sino que los nodos actúan en conjunto para resolver dicho requerimiento y devolver el resultado.
- Capacidad para procesamiento distribuido (“Procesamiento en Red” [16]). Dado que una red inalámbrica de sensores consiste en un conjunto de nodos actuando en colaboración con un objetivo en común, resulta razonable que el procesamiento necesario sea distribuido a lo largo de la red, en lugar de estar centralizado en solo algunos nodos en particular.
- Servicios de nombre y direccionamiento orientados hacia el dato requerido. En el caso de redes inalámbricas de sensores, donde la filosofía colaborativa del grupo de nodos hace que las respuestas sean generadas por la red en conjunto y no por algún nodo en forma individual, surge la necesidad de ofrecer un servicio de direcciones basado en el dato más que en el nodo en sí. Esto significa que el usuario solicitando alguna información deberá direccionar el dato requerido y la red se encargara de proveer este dato, cualquiera sea el nodo que pueda suministrarlo. De esto resulta lógico asignar nombres a los datos, en lugar de una dirección específica, la cual estaría asociando al dato con algún nodo específico, lo cual no condice con la filosofía del trabajo en red.

Habiendo definido la arquitectura del software como una colección de módulos, no todos deben estar siempre disponibles. La estructura mínima del nodo debería contar con protocolos de la capa física, MAC y control de enlaces. A partir de esta mínima funcionalidad, algunas redes requerirán implementar los módulos de ruteo y en algunos casos, de transporte. Como módulos adicionales, y cuando la aplicación lo requiera, se pueden proveer los servicios restantes arriba descriptos.

Por último, dado que la finalidad del nodo es la de contar con una infraestructura que permita realizar el ensayo de distintos protocolos, tanto para verificación de estudios existentes como así también la validación de nuevos protocolos y técnicas a desarrollar, resulta conveniente que los módulos presenten una clara interfaz de modo que pueda ser fácilmente reemplazado por un modulo con similar funcionalidad, pero que implemente una técnica distinta. Contando con una estructura que permite configurar los módulos a incluir, se puede lograr la flexibilidad deseada. De esto modo, el framework puede ser aprovechado por distintos usuarios con

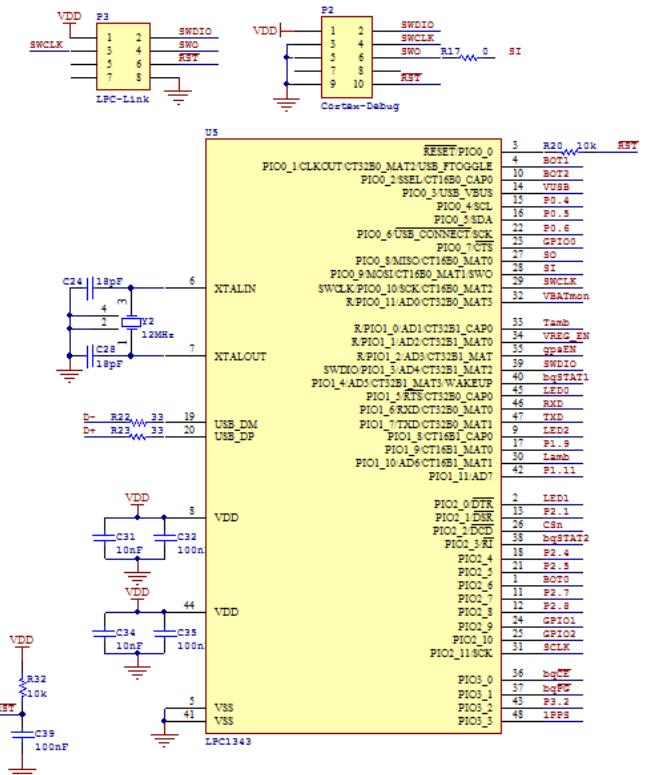


Figura 2. Microcontrolador

diferentes objetivos dentro del campo de las redes inalámbricas de sensores.

IV. IMPLEMENTACIÓN

A. Diagrama esquemático

A continuación se detallarán los componentes del diagrama esquemático del nodo propuesto.

El microcontrolador elegido es el modelo LPC1343 de la firma NXP (Fig. 2) [17]. Con un núcleo ARM Cortex-M3 de 32-bits, posee un tamaño y costo aceptables (LQFP48; US\$3.30 para 100 unidades [18]). Al mismo tiempo, los distintos grupos de trabajo del LSE trabajan con esta familia de microcontroladores, evitando la necesidad de aprenderla desde cero y centrando el trabajo en la implementación del software. Para programación y depuración se colocaron los conectores P2 y P3. Corresponden al header estándar para procesadores Cortex y al header LPC-Link que se incluye con la herramienta LPCXpresso.

La Figura 3 muestra la disposición del transceiver CC2520 [19] y su front-end, CC2591 [20]. Este último permite extender el alcance del CC2520 ya que incorpora un amplificador de bajo ruido para la recepción y una etapa de potencia para la transmisión, admitiendo potencias de salida de hasta 22dBm.

Por su lado el transceiver CC2520 resuelve en hardware algunos aspectos del protocolo IEEE802.15.4 reduciendo la carga de trabajo que tendrá el microcontrolador. Los más relevantes son: envío automático de paquete ACK, aceptación/rechazo de frames automático por CRC, filtrado por campo de dirección, 6 entradas/salidas de propósito general programables para disparar comandos o activarse frente a un determinado

evento, FIFOs de transmisión y recepción, entre otras funciones.

Dado que el nodo debe funcionar en forma autónoma, se optó por implementar un circuito de alimentación a batería con su correspondiente cargador (Fig. 4). El circuito integrado elegido para esta función es el BQ24080 de la firma Texas Instruments [21], que además posee una interfaz digital que permite al CPU conocer el estado de carga así como poner en modo de ahorro de energía al cargador. La batería a utilizar será de Li-ion, 3.7V, 900mAh (valores típicos para una batería de celular). Al mismo tiempo, se utilizan reguladores LDO con muy baja caída de tensión (del orden de 120mV) [22] para lograr un aprovechamiento óptimo de la batería. El criterio para la elección de estos reguladores se basa en su tamaño reducido, su mínima cantidad de componentes externos necesarios y su alta inmunidad al ruido. Utilizar una fuente conmutada en estos casos no es recomendable debido principalmente a la complejidad de los filtros requeridos para eliminar el ruido que podría interferir en la etapa de RF. Obsérvese la utilización de un regulador independiente para el módulo GPS. Esto permite al microcontrolador desactivar este módulo, cuyo consumo ronda los 40mA en forma continua (Fig. 5). El circuito de carga se alimenta a través de un conector USB micro-B, que es el adoptado por las compañías de telefonía celular, con lo cual adquirir una fuente de alimentación de 5V resulta accesible ya que cualquier cargador de teléfono celular cumple con esta condición. Al mismo tiempo se aprovecha el controlador USB integrado en el microcontrolador LPC1343, en caso que la aplicación requiera este tipo de comunicación.

El módulo GPS (Fig. 6) incluido en este dispositivo es el modelo ET318 de la firma Globalsat [23], y constituye un hardware relativamente estándar. Si bien diferentes fabricantes de GPS optan por encapsulados distintos en cada caso, las conexiones con el resto del sistema no suelen variar de un módulo a otro: los pines de comunicación RS232 y salida PPS (*Pulse Per Second*) para tareas de sincronización fueron las utilizadas en este diseño. Los módulos GPS comerciales envían la información a través del protocolo NMEA 0183 en forma serial RS232. La salida PPS, como su nombre lo indica, genera una señal pulsante de 1Hz. La utilidad de esta señal radica en que una vez que el módulo ha adquirido la información de los satélites (*ephemeris*), todos los módulos GPS del mundo generan el pulso en el *mismo instante*, resultando en una señal muy útil para sincronizar dispositivos cuando el alcance de RF se ve afectado por la distancia.

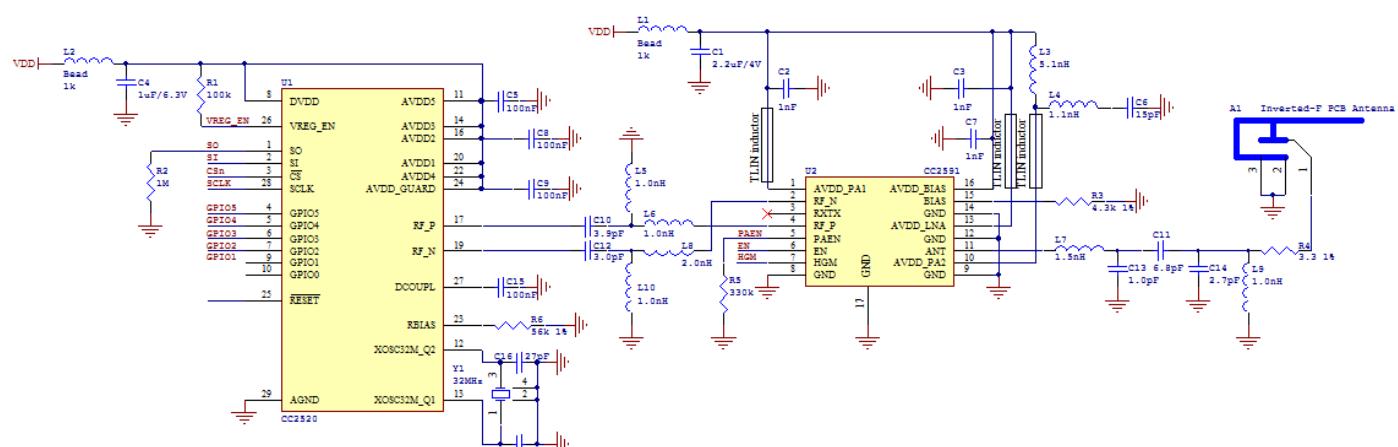


Figura 3. Etapa de RF – Transceiver y front-end

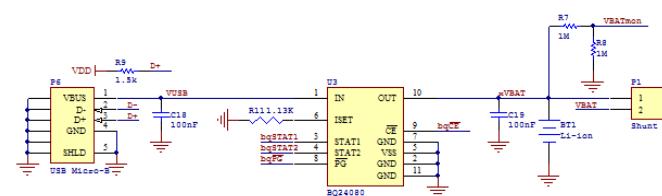


Figura 4. Circuito de conexión USB y cargador de batería.

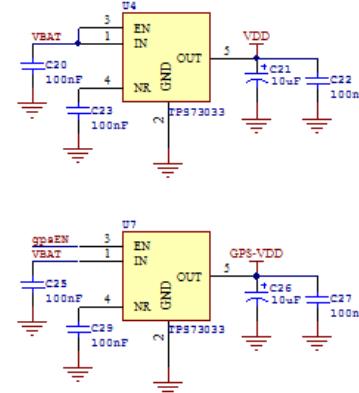


Figura 5. Reguladores de tensión. U7 provee 3.3V exclusivamente al módulo GPS. Esto permite, mediante la señal gpsEN, desconectarlo para ahorrar energía.

Se incluyen también tres pulsadores y tres LEDs de propósito general, un sensor de luz formado por un fotodiodo y un sensor de temperatura (MCP9700AT de Microchip [24]) para mediciones y pruebas sobre magnitudes reales del entorno.

Finalmente, un conector de expansión conectado a los puertos que quedaron libres del microcontrolador permitirá expandir la funcionalidad del nodo al conectar otros periféricos.

B. Modelo de PCB

Se ha diseñado la placa de circuito impreso de doble faz en función de los parámetros de fabricación recomendados por los fabricantes en Argentina, en cuanto a mínimo ancho de pistas y diámetro de perforaciones. La Figura 7 muestra el modelo resultante, cuyas dimensiones son 43 x 95 mm.

C. Características técnicas principales

Las presentes características están basadas en las especificaciones de los distintos componentes.

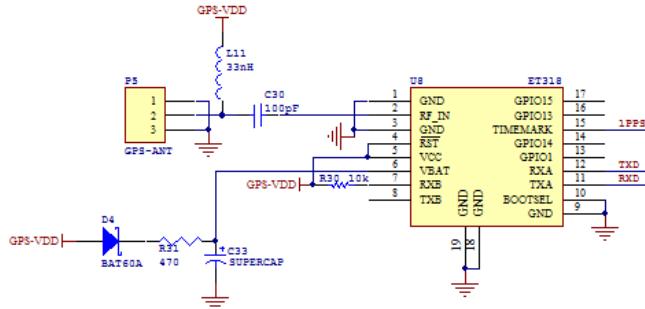


Figura 6. Módulo GPS

1) *Microcontrolador LPC1343*

- Procesador Cortex-M3 @ 72MHz.
 - 32kB de memoria Flash.
 - 8kB de memoria RAM.
 - Consumo estático @ 72MHz: 17mA.
 - Consumo estático @ 12MHz: 4mA.
 - Consumo estático en modo *deep power-down*: 220nA.
- 2) *Transceiver CC2520+CC2591*
- Consumo Tx @22dBm: 145mA.
 - Consumo Rx: 22mA.
 - Consumo en estado idle: 1.3µA.
 - Sensibilidad: -98dBm + (-6dBm) = -104dBm.
 - Link Budget: 104dBm + 22dBm = 126dB.
- 3) *GPS ET-318*
- Sensibilidad: -159dBm.
 - Canales: 20.
 - Consumo estático: 42mA.
- 4) *Generales:*
- Autonomía, considerando una batería de 900mAh, transmisión el 20% del tiempo, GPS encendido el 50% del tiempo, CPU con clock máximo: 10.63 horas.
 - Alimentación externa: 4.5 a 6.5V.

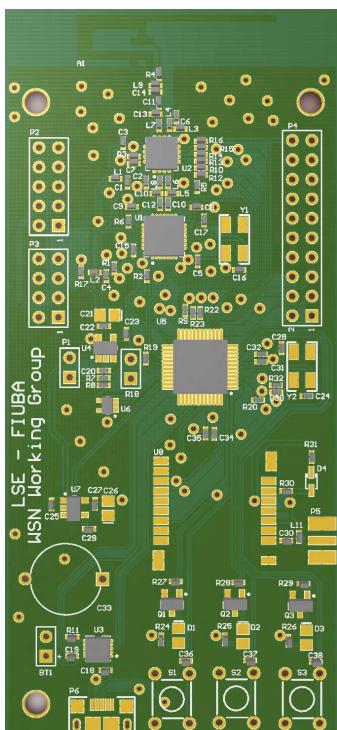


Figura 7. Modelo de placa de circuito impreso (PCB).

D. Costos estimados

La estimación de costos se realizó en función de los precios FOB de la firma Digi-Key para los componentes de montaje superficial, salvo el módulo GPS que fue presupuestado por la firma Cika Electrónica y el costo de fabricación de los PCB que fue presupuestado por la firma Eleprint. Debido a que el GPS no es necesario en todas las aplicaciones, se plantean diferentes opciones en la Tabla II. Tomando como referencia los costos de los nodos comerciales detallados en la sección II., y considerando la opción de diseño sin el módulo GPS, se observan características del hardware implementado que superan a los anteriores en materia de prestaciones, periféricos y autonomía, manteniendo un costo dentro del orden de magnitud observado en la Tabla I.

TABLA II. COSTO UNITARIO FOB POR CADA NODO EXPRESADO EN DÓLARES ESTADOUNIDENSES

	Con GPS	Sin GPS
10 unidades	91.09	59.83
100 unidades	68.76	38.03

E. Experimentación y análisis de los primeros resultados

La Figura 8 muestra un nodo parcialmente montado con sus componentes fundamentales: microcontrolador, transceiver, front-end y cargador de batería. Se construyeron tres nodos similares para realizar las pruebas de enlace e intercambio de datos a nivel de capa física, a fin de verificar el funcionamiento de la etapa de RF. Se ha comprobado que el alcance indoor entre nodos ronda los 25 metros, considerando que en el ambiente de prueba existen divisiones de durlock y una pared maciza de 15cm de ancho. Al momento no fue posible realizar pruebas al aire libre.

Al mismo tiempo se realizaron mediciones de consumo: Con el microcontrolador funcionando a su clock máximo (72MHz) y en transmisión permanente de datos, el consumo del nodo ronda los 100mA. En modo recepción permanente, el mismo consumo promedia los 50mA. Se ha planificado, luego de finalizada la etapa de testing del stack definido en la sección III.B., crear diferentes perfiles de energía que apunten a reducir el consumo del sistema a fin de aumentar su autonomía.

En esta etapa de puesta en marcha se ha utilizado un firmware reducido en funcionalidades que implementa únicamente las capas de abstracción de hardware, física y de enlace. Al mismo tiempo este stack mínimo admite la lectura

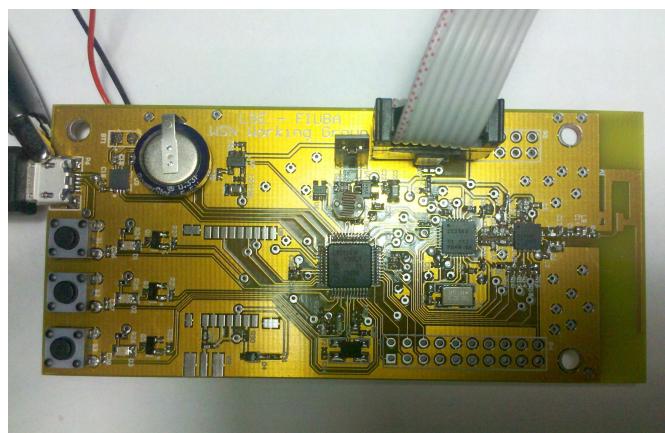


Figura 8. Fotografía de uno de los nodos construidos para las pruebas de enlace.



de los parámetros RSSI y LQI (Link Quality Indication), que vienen asociados a cada frame aceptado por el transceiver. Como se mencionó en la sección III.A. estos parámetros son fundamentales para la implementación del servicio de localización.

V. CONCLUSIÓN

La ventaja del nodo implementado radica principalmente en características que no se encontraron en los SoMs mencionados anteriormente:

- Capacidad para funcionamiento a batería.
- Posibilidad de programación directa del CPU, habilitando al módulo para ejecutar cualquier tipo de firmware y/o RTOS, aprovechando la extensa literatura existente sobre la arquitectura Cortex-M3.
- Versatilidad de implementación de protocolos basados en 802.15.4 al tener control completo del transceiver.
- Potencia de transmisión superior: 22dBm.
- Módulo GPS y periféricos integrados para medición de variables del entorno.

Disponer de estos dispositivos permitirá a los diferentes grupos de trabajo del Laboratorio de Sistemas Embebidos, así como otros grupos de investigación que deseen participar, estudiar y experimentar con diferentes topologías de red inalámbrica e intercambio de información. En este sentido vale mencionar que actualmente existen vínculos activos con diversos grupos de investigación de la región interesados en utilizar estos nodos, en Mendoza, Córdoba, Paraguay, Brasil, Costa Rica, México, etc. Hasta la fecha no se conoce ningún desarrollo similar que se haya realizado en laboratorios de investigación de países de la región.

Al mismo tiempo, el framework implementado en este hardware provee una capa de servicios de red que brinda un nivel de abstracción adecuado para llevar a cabo distintas experiencias, como ser *testbenches* sobre algoritmos de sincronización y localización, domótica y control remoto de virtualmente cualquier tipo de hardware.

REFERENCIAS

- [1] "Wireless sensor networks: technology, protocols, and applications" , Kazem Sohraby, Daniel Minoli, Taieb Znati. Wiley-Interscience (2007). Capítulo I.
- [2] "Wireless sensor networks: technology, protocols, and applications" , Kazem Sohraby, Daniel Minoli, Taieb Znati. Wiley-Interscience (2007). Sección 3.2.2.
- [3] "Wireless sensor networks: technology, protocols, and applications" , Kazem Sohraby, Daniel Minoli, Taieb Znati. Wiley-Interscience (2007). Sección 6.3.
- [4] "Wireless Broadband in a Box", B. Schrick, M.J. Riezenman, IEEE Spectrum, Jun 2002, pp. 38-43.
- [5] Módulos ATMEL:
http://www.atmel.com/tools/ZIGBIT2_4GHZMODULEWITHBALANCE_CEDRFOUTPUT.aspx
- [6] Módulos CEL: <http://www.cel.com/static.do?command=ZigBeeNewProd&group=5&rpart=ZFSM-201>
<http://www.cel.com/static.do?command=MeshConnect&group=5&rpart=ZICM2410P0>
- [7] Módulos RadioCraft: <http://www.radiocrafts.com/index.php?sideID=474>

- [8] Módulos RF Monolitics: http://rfm.com/products/spec_sheet.php?802.15.4transceivermodules_&record=LPR2430
- [9] Módulos Digi:<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/xbee-zb-module#overview>
- [10] Módulos NIVIS:
http://www.nivis.com/industrial_sensor_networks/VersaNode200.php
- [11] Módulos Jennic:
http://www.jennic.com/products/wireless_microcontrollers/
- [12] Módulos RadioPulse: http://www.radiopulse.co.kr/eng/main.html?mode=02_05
- [13] Módulos Panasonic: <http://www.panasonic.com/industrial/electronic-components/rf-modules/snap4555.aspx>
- [14] S. G. Fischer, T. A. Wysocki and H. J. Zepernick, "MAC protocol for a CDMA based wireless ATM LAN", IEEE International Conference on Communications, Montreal, Que., Canada, 8-12 June 1997.
- [15] Y. Nakagawa, H. Uchiyama, H. Kokaji, S. Takahashi, M. Suzuki and M. Kanaya, "Multi-channel adhoc wireless local area network", 48th IEEE Vehicular Technology Conference, Ottawa, Ont., Canada, 18-21 May 1998.
- [16] H. Karl and A. Willig, "Protocols and Architectures for Wireless Sensors Networks", John Wiley & Sons, 2007
- [17] NXP LPC1343:
http://ics.nxp.com/support/documents/microcontrollers/pdf/user.manual_lpc13xx.pdf
http://www.nxp.com/documents/data_sheet/LPC1311_13_42_43.pdf
- [18] LPC1343 – Costos en DigiKey:
<http://search.digikey.com/us/en/products/LPC1343FBD48,151/568-4945-ND/2180456>
- [19] TI CC2520: <http://www.ti.com/lit/ds/symlink/cc2520.pdf>
- [20] TI CC2591: <http://www.ti.com/lit/ds/symlink/cc2591.pdf>
- [21] TI BQ24080: <http://www.ti.com/product/bq24080>
- [22] TI TPS73033: <http://www.ti.com/lit/ds/symlink/tps73033.pdf>
- [23] GlobalSat ET318: http://www.globalsat.com.tw/products-page.php?menu=2&gs_en_product_id=4&gs_en_product_cnt_id=36
- [24] Microchip MCP9700A:
<http://ww1.microchip.com/downloads/en/DeviceDoc/21942e.pdf>

Recarga Fácil por Radio Frecuencia, RF²

Daniel Aicardi, Melina Rabinovich, Edgardo Vaz

daicav@gmail.com, mrabinovichm@gmail.com, edgardovaz@gmail.com

Resumen—El presente documento muestra las características de hardware y software que componen un prototipo de sistema embebido enfocado a operar con tarjetas RFID (ISO14443) como las que son utilizadas actualmente en el sistema de transporte de la ciudad de Montevideo.

Palabras clave—Tarjetas inteligentes con y sin contacto, RFID, ISO14443, Mifare, CL RC632.

I. INTRODUCCIÓN

El uso de tarjetas inteligentes es cada vez más frecuente en todos los ámbitos de nuestra vida cotidiana. Tal es el caso del sistema de transporte de la ciudad de Montevideo, donde cada pasajero utiliza una tarjeta RFID para efectuar el pago de cada viaje. Esto último implica que cada usuario debe cargar saldo en su tarjeta para su posterior uso. Es necesario entonces brindar un mecanismo simple, seguro y rápido que permita asignar saldo a cada tarjeta; la siguiente figura muestra un diagrama simplificado del sistema.

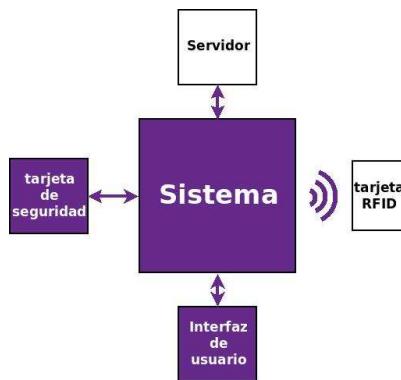


Fig. 1. Bloques que conforman el sistema

De los conceptos antes mencionados, se decidió implementar los bloques correspondientes al sistema que incluye el lector/escritor de tarjetas RFID, el lector/escritor de tarjetas de seguridad y la interfaz de usuario. No fueron implementados los bloques relativos al servidor y las tarjetas RFID.

Las partes implementadas forman un prototipo de sistema embebido que interactúa con tarjetas RFID, permitiendo consultar y/o acreditar saldo en las mismas. El mecanismo de transferir saldo en las tarjetas RFID mediante éste dispositivo, se encuentra desacoplado del sistema de pago de dinero, el cual podría efectuarse a través de una red de pagos, mensajes de texto, web, etc, esto

D. Aicardi, M. Rabinovich, E. Vaz - Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay, daicav@gmail.com, mrabinovichm@gmail.com, edgardovaz@gmail.com

último no forma parte de este proyecto por tanto no fue implementado.

II. OBJETIVO

El objetivo del proyecto es la fabricación de un prototipo de sistema embebido capaz de consultar y recargar tarjetas. Para esto, como se mencionó en la introducción, deberá lograr establecer comunicación con tarjetas como las utilizadas en el sistema de transporte de la ciudad de Montevideo (comunicación RFID a 13,56 MHz), con tarjetas de contacto (módulo de seguridad SAM), y con el usuario a través de una interfaz simple.

Esto implica entonces la fabricación de dos lectores/escritores de tarjetas, uno para tarjetas RFID (sin contacto) y otro para tarjetas con contacto (SAM), una interfaz para el usuario capaz de informar el estado de la transacción mediante mensajes adecuados, y la utilización de un sistema basado en un microprocesador para controlar los periféricos y realizar las operaciones. Esto último implica además el desarrollo del software para que todo funcione adecuadamente.

III. HARDWARE

En una primera instancia se pretendía utilizar únicamente el dispositivo OpenPCD [1], que es un elemento hardware abierto orientado a operar por acoplamiento magnético (Proximity Coupling Devices) a la frecuencia 13,56 MHz, y que es compatible con las normas ISO14443, ISO15693 y protocolos propietarios como Mifare Classic. Una vez estudiado se llegó a la conclusión que no permitía la instalación de un sistema operativo GNU/Linux, ya que el mismo precisa más de 4 MB de RAM para poder hacer algo útil. Otra desventaja encontrada fue que sólo tiene un puerto I2C como forma de conectar periféricos, además que su costo encarece el prototipo.

Surgió entonces la necesidad de usar una Single Board Computer (SBC), como dispositivo capaz de ejecutar un sistema operativo y las aplicaciones necesarias para que el dispositivo cumpla con los requerimientos exigidos.

Fue necesario entonces descartar el uso del dispositivo OpenPCD y dar lugar a un diseño propio del lector/escritor de tarjetas RFID, utilizando para esto el integrado CL RC632 de Philips.

Se pensó entonces en diseñar la arquitectura que consta de SBC, lector de tarjetas RFID, lector de tarjeta de contacto, display, buzzer, LEDs.

En la Figura 2 se muestra un diagrama de bloques correspondiente a la arquitectura seleccionada:

Los bloques oscuros serán implementados, no así los claros.

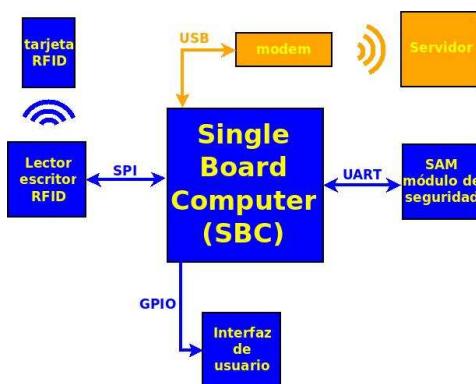


Fig. 2. Diagrama de bloques de la arquitectura seleccionada

La comunicación con un servidor no está implementada por quedar fuera del alcance del proyecto.

A. Elección de hardware

A.1 SBC

En primera instancia se confeccionó una lista con posibles candidatas de SBC disponibles en el mercado internacional, teniendo en cuenta factores como: precio, puertos de I/O, memoria RAM, memoria Flash, puertos USB, soporte para GNU/Linux, entre otros. Se definieron una serie de requisitos mínimos necesarios para seleccionar la SBC que más se adecuara a la arquitectura definida. Para la comunicación con el resto de los módulos son necesarios: una interfaz UART para el módulo de seguridad (SAM); una interfaz SPI para el módulo lector/escritor RFID (CL RC632 de Philips); 20 GPIO para display, LEDs, buzzer, otros; 1 USB host previendo una futura conexión de un modem 3G (intercambio de datos con un servidor). En cuanto a la memoria disponible debe ser de 32 MB de RAM y 8 MB de flash, como mínimo, para un funcionamiento aceptable. Es conveniente, pensando a futuro, que el procesador trabaje a una frecuencia no menor a 200 MHz. Dado el presupuesto estimado para el proyecto, el precio no debe superar los 150 dólares en origen. Como requisito adicional se exigió que existiera un foro actualizado y soporte técnico que permitiera evacuar dudas.

Finalmente, la SBC seleccionada para trabajar fue la Beagleboard rev.C4 [2].

Las características generales de la BeagleBoard son: cuenta con un procesador OMAP3530 de 720 MHz con arquitectura ARM. Posee memoria NAND-flash de 256 MB y memoria ROM de igual tamaño. Tiene una ranura adicional para extender la memoria a través de una memoria SD. Entre otras cosas cuenta con un puerto USB OTG, un puerto USB host, un bloque de expansión de 28 pines (con señales a 1,8 V), puerto JTAG, conector RS232, etc. En lo que respecta a la potencia disipada, la Beagleboard tiene un consumo de pico de 2 W, y un consumo promedio de 560 mW [3] [4].

A.2 VLT - Conversor de Voltajes

La placa de circuito impreso VLT consta básicamente de dos conectores, uno de ellos permite la conexión con la Beagleboard y el otro la conexión con el restante hardware el cual se encuentra intergrado en un PCB llamado SCUI. Ambos conectores no se encuentran directamente interconectados entre sí a través de pistas, pues para el caso particular de Beagleboard fue necesario incorporar conversores de tensión que permitieran el traslado del nivel de tensión desde 1,8 V que usa esta SBC, a las tensiones con las que operan los periféricos, ya sea 3,3 o 5 V.

A.3 Lector de tarjeta de contacto

Está compuesto por un conversor full duplex a half duplex el cual se encuentra conectado a uno de los puertos UART de la SBC a través del módulo VLT, que se describió en el punto anterior. Este conversor permite la transmisión de datos directamente entre la tarjeta y la SBC, sin necesidad de intercalar un ASIC para el manejo de tarjetas del tipo ISO7816. Cuenta también con un oscilador para alimentar la entrada de reloj de las tarjetas. La entrada de control (OE) del oscilador operada desde la SBC permite poner la salida de reloj en tercer estado, cosa muy útil a la hora de cumplir con la secuencia de inicialización de las tarjetas descritas en el estándar. Se cuenta con un zócalo para insertar la tarjeta de contacto.

A.4 Interfaz de usuario

Está compuesta por tres LEDs (verde, amarillo y rojo), buzzer y un display LCD16x2 donde son desplegados los mensajes que indican al usuario la operación que se efectúa sobre su tarjeta.

A.5 Lector de tarjetas RFID

Este módulo es el encargado de la comunicación con las tarjetas RFID que cumplen con la norma ISO14443. Consta básicamente de cuatro secciones entre las que se encuentran: el integrado CL RC632; el filtro EMC, el circuito de adaptación de impedancia (matching); y el inductor de la antena. El ASIC CL RC632 permite, por un lado la comunicación digital con un microprocesador a través de su puerto de datos y por el otro lado la transmisión de datos hacia la antena que emitirá la señal RF para la comunicación con las tarjetas ISO14443.

Lo que se llama propiamente antena RF está conformada por el circuito de adaptación de impedancia (matching) y por el inductor ubicado en el circuito impreso, que propaga el campo magnético para lograr el acoplamiento necesario entre lector y tarjeta, de aquí la sigla PCD (Proximity Coupling Device).

IV. SOFTWARE

A. Introducción

Se debe destacar que todo el desarrollo de software se basó exclusivamente en herramientas de software libre. La distribución GNU/Linux elegida para el sistema embebido se llama Angström [5]. Esta distribución es recomendada



en aplicaciones que utilizan una SBC basada en arquitectura ARM como la Beagleboard, y cuenta con una gran cantidad de bibliotecas implementadas en lenguaje C, que permiten gran escalabilidad a la hora de incorporar nuevos periféricos en la aplicación.

A.1 Sistema Operativo

El sistema operativo se compone de: sistema de archivos (fileSystem), kernel (uImage) y bootloader (MLO y u-boot). En el arranque, la Beagleboard tiene la posibilidad de buscar el bootloader y el kernel en memoria NAND, o en dispositivos extraíbles tales como memorias USB o memorias SD. Para el sistema RF², se eligió un arranque a través de una memoria SD ya que es fácil de manipular.

En la Figura 3 se puede ver como queda distribuida la memoria SD con las distintas partes que conforman el sistema operativo.

En la memoria SD se pueden distinguir dos particiones, una en formato FAT32 y otra en formato ext3. La partición en FAT32 es llamada “de arranque”, y es donde se encuentra el bootloader (MLO, u-boot.bin) y la imagen comprimida del kernel (uImage). La partición en ext3 es donde se encuentra el sistema de archivos (fileSystem) asociado con la distribución elegida.

B. Herramientas utilizadas en el desarrollo del sistema

B.1 Introducción

Para la elección de las herramientas se tomó como primer criterio de decisión el hecho que sean libres, así como las experiencias de otras personas que ya han transitado caminos comunes, consultando y participando en foros activos en Internet.

B.2 MLO, u-boot.bin y uImage

No fue necesario generar el MLO debido a su simpleza, puesto que el binario precompilado realiza bien su función.

El u-boot.bin y el uImage fueron generados con la herramienta de desarrollo y compilación OpenEmbedded-Bitbake [6] que es una fusión de dos herramientas: Open-Embedded, herramienta para construcción y mantenimiento de distribuciones, y Bitbake, herramienta de compilación similar al Make que automatiza la construcción de ejecutables entre otros. Esto es, OpenEmbedded utiliza Bitbake para su objetivo. OpenEmbedded-Bitbake es una herramienta muy potente y difícil de aprender al principio.

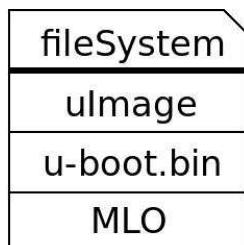


Fig. 3. Memoria SD para funcionar en Beagleboard

Luego de entender su principio de funcionamiento se hace muy simple su uso.

B.3 FileSystem

Para la generación del FileSystem de Angström, se utilizó la herramienta web Narcissus disponible en la página de Angström.

B.4 Crosscompilación

Para la crosscompilación se utilizó el kit de desarrollo (SDK) generado por Narcissus y la herramienta Make para generar los archivos necesarios.

B.5 Depuración de código

Para la depuración, se utilizó la herramienta GDB [7] del proyecto GNU.

B.6 Bibliotecas

- psc-scan [8] Esta aplicación fue utilizada para testear el lector de tarjetas de contacto.
- librfid-tool Esta herramienta fue utilizada para testear lectores/escritores de tarjetas RFID.

C. Desarrollo

C.1 MLO

Como se mencionó anteriormente, no fue necesario generar el MLO debido a que la Beagleboard viene con uno pre-instalado, y en caso que no funcione en forma correcta se puede descargar desde un repositorio de Angström.

C.2 Multiplexado de pines

El microprocesador OMAP3530 tiene muchos pines con distintas interfaces entre las que se cuentan puertos UART, SPI, GPIO, etc., pero no todos son accesibles desde la Beagleboard. Para poder acceder a algunos de estos puertos del microprocesador, existe en la placa de la Beagleboard un bloque de expansión de 28 pines.

Por defecto, en el bloque de expansión no se encuentran las señales deseadas para esta aplicación. Esto lleva a que se tenga que modificar el estado inicial de los pines. Existen dos formas de modificar los pines de modo de tener las señales que se precisan. Una de ellas es modificar el bootloader y la otra es modificar el kernel (uImage). Esto implica cambios en los archivos fuentes y posterior compilación que genere los nuevos binarios u-boot o uImage.

Para la modificación de las señales disponibles en el bloque de expansión se decidió modificar el u-boot, ya que la modificación por u-boot es más intuitiva y por experiencia se sabe que lo que más se actualiza y/o modifica es el kernel.

C.3 u-boot

Como se mencionó anteriormente, en el u-boot se realizó la configuración de los pines del bloque de expansión de la Beagleboard. Cada pin del bloque de expansión tiene varias funcionalidades asociadas, y la configuración de la funcionalidad depende de un multiplexado modificable a



nivel de software. Esto es, dependiendo del “modo de pin” elegido, la función que se obtiene en dicho pin.

Pese a que en la literatura y foros, se plantea lo contrario, no fue posible establecer los atributos “valor” y “dirección” de los pines GPIO mediante la modificación planteada. Lo que sí cambia efectivamente es el modo del pin, permitiendo obtener las interfaces adecuadas en el bloque de expansión.

C.4 uImage

La versión del kernel elegida fue la 2.6.32 ya que en el momento del desarrollo era la versión más estable. En algunos casos, no aparecen algunas de las interfaces configuradas, lo que lleva a modificar los fuentes del kernel para que esto así suceda. Este fue el caso de la interfaz SPI que no quedó mapeada en /dev pese a que había sido configurada en los fuentes del u-boot. También hubo problemas con los atributos “valor” y “dirección” de los GPIO, como se mencionó anteriormente. Adicionalmente hacia falta un módulo para simular una conexión Ethernet sobre una interfaz USB para establecer una conexión entre la Beagleboard y un PC, que emula una tarjeta de red por software. Todo esto llevó a que se tuvieran que modificar los archivos fuente del kernel.

C.5 FileSystem

Como se mencionó anteriormente, el fileSystem se generó a partir de la herramienta web Narcissus. En el fileSystem es donde se encuentran los paquetes y programas ya instalados. Cuanto más programas se instalen más grande será en tamaño el fileSystem.

C.6 Bibliotecas

Software para el manejo de GPIO

Este módulo de software fue realizado desde cero, basándose en el fileSystem virtual SYSFS para el control de GPIOs [9] [10], esto permite que el código pueda ser portado a cualquier otro sistema que use GNU/Linux y que disponga de este tipo de hardware.

Software para comunicación SAM

Como fue descrito en la sección de hardware, el lector de tarjetas de contacto tiene una interfaz serial pura para la transferencia de datos con las tarjetas. En base al diseño hardware elegido, las capas de software sobre las que se decidió trabajar son las que se detallan en la Figura 4.

A continuación se describen las capas asociadas a la Figura 4.

Controlador:

El kernel es el encargado de manipular directamente los registros del puerto serial, las interrupciones que desde éste se generan y la ISR para atender las interrupciones. La implementación del controlador del lector de tarjetas se basó en el controlador serial de GNU/Linux a través de su estructura “termios” [11]. Esta estructura permite

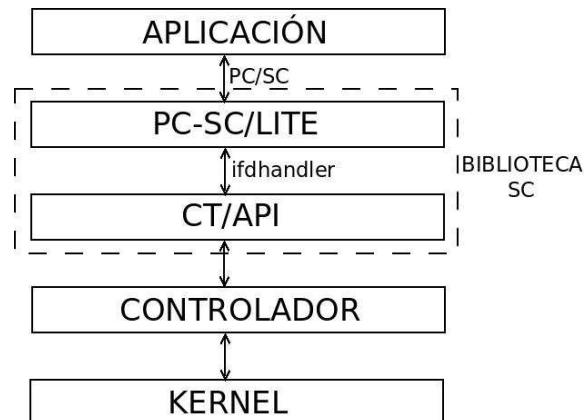


Fig. 4. Capas de software de trabajo

configurar todos los parámetros necesarios para la comunicación serial como ser, baud rate, cantidad de bits por byte, bit de paridad, bit de parada entre otros.

CT/API:

Por encima del controlador serial se encuentra CT/API (Card Terminal / Application Programming Interface) [12], una interfaz definida por varias empresas (entre las que se incluye Telekom Alemania) en la década de los noventa, que permite encapsular el controlador específico de cada lector de tarjetas, de manera que la aplicación final no se vea afectada al cambiar un lector por otro. Esta interfaz de programación está formada tan solo por tres funciones, que permiten la inicialización del lector, la transferencia de datos entre host/lector o host/tarjeta (host se refiere a la SBC o PC donde se encuentra conectado el lector de tarjetas de contacto) directamente y el cierre de la comunicación.

IFDHandler:

El siguiente componente en este stack de capas es ifdhandler [13]; no es otra cosa que un conjunto de funciones formando una API, empleada por pcsclite [15] para encapsular el manejo del hardware de lectores cuyos fabricantes quieran cumplir con las especificaciones PC/SC [14]. Una ventaja importante de esta API es que le permite a pcsclite operar tanto con lectores de puerto serial como con lectores de puerto USB. Esta capa de software podría utilizarse directamente sobre el controlador del lector, prescindiendo de CT/API, aunque se decidió mantenerla por motivos de simplicidad ya que sólo es necesario sustituir la capa de aplicación por las restantes capas superiores como se indica en la Figura 4.

Las funciones que contiene esta capa de software fueron modificadas para que contengan a su vez las funciones de CT/API, y de esta manera hacer más fácil de integrar el controlador del lector con la biblioteca PCSCLite.

PCSCLite:

Por arriba de ifdhandler se encuentra la biblioteca pc-

sclite, ésta contiene todas las funciones necesarias para establecer la comunicación con un lector y la tarjeta conectada a éste último.

Aplicación final:

Por arriba de todas las capas descritas antes, se encuentra la aplicación del prototipo que hace uso de las funciones suministradas por pcsclite y donde se encuentran definidos los comandos APDU específicos con los que opera la tarjeta de contacto.

Software RFID

librfid [16] es una biblioteca de software libre para manejo de lectores/escritores RFID. Implementa el stack de protocolos del lado del dispositivo lector/escritor ISO 14443A, ISO 14443B, ISO 15693, Mifare Ultralight y Mifare Classic.

Entre los lectores soportados están, OpenPCD y algunos modelos Omnikey, estos con interfaz de conexión USB. Además tiene soporte para cualquier otro lector con comunicación directa con el CL RC632 mediante la interfaz SPI y es por esta razón que se tuvo en cuenta.

El manejo de librfid es de bajo nivel y se comunica directamente con el kernel utilizando el módulo spidev.

Librfid-tool implementa funciones de más alto nivel que hacen uso de las funciones de bajo nivel de la librfid. La estructura de software comentada se muestra en la Figura 5.

Después de estudiadas las funciones que provee la herramienta librfid-tool, se estudió la posibilidad de su uso para la aplicación RF².

Aunque sus funciones son muy útiles, la gran mayoría no sirven completamente para la aplicación RF² debido a que fueron definidas para otros propósitos.

Es de interés, que la herramienta librfid-tool siga manteniendo sus funcionalidades y pueda convivir con la aplicación RF², por lo que no se modificó el contenido de ninguna función de la herramienta. En el caso que alguna función fuera mayormente utilizable, se procedió a crear una nueva con los cambios necesarios.

Se implementaron la mayoría de las funciones, logrando compatibilidad con la biblioteca librfid. Entre las funciones creadas están las asociadas con la tarjeta RFID:

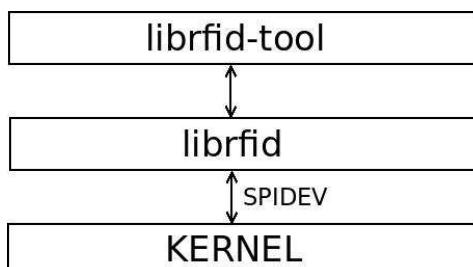


Fig. 5. Capa de software RFID

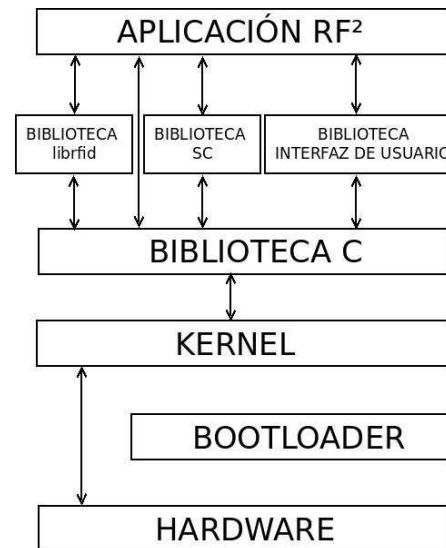


Fig. 6. Capas de software del sistema RF²

autenticación según el tipo de clave, búsqueda de tarjetas próximas al lector, lectura y escritura de bloques de memoria, obtención del UID, etc.

C.7 Aplicación final

Para el desarrollo de la aplicación RF² se decidió trabajar sobre los fuentes de la herramienta librfid-tool, ya que maneja varias funciones de utilidad y es de ayuda a la hora de compilar para el armado de una aplicación completa. Se mantuvieron todas las opciones de la herramienta ya que son útiles, y pueden ayudar en un futuro para establecer orígenes de fallas. No se modificó ninguna función de la aplicación original y cuando fue necesaria alguna modificación, se procedió a implementar una nueva.

En la Figura 6 se detallan las capas de software del sistema RF².

Modificaciones en librfid-tool

Se implementó una función de inicialización, la cual inicializa todos los periféricos al arrancar la aplicación RF².

Se definieron constantes simbólicas para hacer configurable al sistema.

Se agregó una nueva opción en el main de la aplicación librfid-tool que llama a la función principal(). Ésta, es la función principal de la aplicación RF², la cual se ejecuta en loop. De este modo no se modifica el main original de librfid-tool (sólo se agrega una opción) y se dejan las opciones por defecto.

Funcionamiento general de la aplicación

Se desarrolló una estructura de directorios de forma que el sistema sea modular, esto es, cualquier cambio requerido en algún módulo del sistema, no afecta al resto.

De las funciones, salvo dos, todas fueron realizadas por el grupo de trabajo.



V. COSTOS

El costo total del proyecto asciende aproximadamente a 1385 dólares, habiendo estimado un gasto total de 1500 dólares.

En cuanto al costo de fabricación de los PCB, se puede decir que las diferencias en precio son sustanciales, dependiendo del origen del fabricante.

El costo por unidad con los PCB hechos en Uruguay es de aproximadamente 375 dólares, si los PCB se fabricaran en China el costo sería de unos 242 dólares. Estos valores incluyen, la SBC, la fabricación de las placas, y los componentes necesarios, no incluyen el armado de las mismas.

VI. MEJORAS Y TRABAJOS A FUTURO

Entre las cosas que se deben investigar, es que el blindaje incluido en la primer antena fabricada cumpla con las regulaciones de compatibilidad electromagnética, EMC, definidas por la EN de Europa o por la FCC de Estados Unidos. A tales efectos sería necesario contar con el equipamiento adecuado para efectuar las mediciones necesarias.

Tal vez la disminución del tamaño del PCB de la antena, o la incorporación de ésta al resto del hardware, puedan ser vistas como mejoras desde el punto de vista de disminución de costos.

En cuanto al lector de tarjetas de contacto, su incorporación a la lista de lectores soportados por la biblioteca pcsclite quedó inconclusa; si bien el hardware está operativo y es posible enviar comandos APDU al módulo SAM de seguridad a través de una pequeña aplicación, sería importante alcanzar el objetivo planteado al comenzar el proyecto.

En lo que refiere al diseño industrial, faltaría diseñar y fabricar una carcasa acorde. Ésta debe permitir el anclaje rápido y seguro de las distintas partes de hardware, sin que su material interfiera con la propagación de radio frecuencia. Esto descarta la posibilidad de usar una carcasa metálica.

Algo no previsto en el prototipo, es el hecho de que ocurra un corte de energía. Puede solucionarse incluyendo un sistema con una batería, que se active al detectar la falta de energía externa, capaz de alimentar el dispositivo hasta terminar una posible transacción en curso y luego ingrese en modo fuera de línea.

Todos los elementos que forman parte del software del sistema se encuentran almacenados en la memoria SD, sin embargo este tipo de memoria es de menor calidad que la NAND Flash con la que cuenta la SBC utilizada, y por tanto aumenta la posibilidad de fallas, hecho que ocurrió con una de las SD al tener varios bloques de almacenamiento defectuosos. El inconveniente anterior, y otros asociados con temas mecánicos como vibraciones (si el dispositivo se instala en un vehículo) o suciedad en los contactos, etc., podrían evitarse si los archivos son almacenados directamente en memoria Flash.

Un detalle a mejorar en la inicialización del lector/escritor RFID es que verifique que realmente se trata

de este dispositivo y no de otro que se encuentre conectado al puerto SPI. Esto puede hacerse efectuando una lectura del identificador de producto desde la memoria EEPROM integrada CL RC632.

Migrar de la versión 2.6.32 del kernel, a la versión 3.0 sería otro de los tantos puntos a mejorar.

VII. CONCLUSIONES

El mundo de la tecnología RFID está poco explorado en nuestro país, éste tal vez sea el primer proyecto que incluye el diseño y fabricación de un lector/escritor RFID capaz de operar con tarjetas sin contacto en la banda de frecuencia de 13,56 MHz.

El aporte realizado en este campo es tan solo una primera aproximación y aún queda mucho por hacer al respecto. Sobre este proyecto en particular es necesario mejorar varios aspectos antes de pasar de la fase de prototipo a la de producción, como ya mencionamos en el punto anterior.

Repasando en particular los criterios de éxito, el proyecto ha resultado satisfactorio porque se logró construir un dispositivo capaz de consultar y recargar tarjetas RFID.

BIBLIOGRAFÍA

- [1] *OpenPCD*, <http://www.openpcd.org/>, Accedido: agosto 2011.
- [2] *Página oficial Beagleboard*, <http://beagleboard.org/>, Accedido: agosto 2011.
- [3] “Beagleboard FAQ” (Power Usage), http://elinux.org/BeagleBoardFAQ#Power_usage, *Embedded Linux Wiki*; Accedido: agosto 2011.
- [4] “Consumo de Beagleboard”, <http://www.h-online.com/news/article/14369911/read/>, *The H*, Accedido: agosto 2011.
- [5] *Angström*, <http://www.angstrom-distribution.org/>, Accedido: agosto 2011.
- [6] “Openembedded-Bitbake”, <http://wiki.openembedded.net/index.php/BitBake.%28dev%29>, Accedido: agosto 2011.
- [7] “GDB: The GNU Project Debugger”, <http://www.gnu.org/software/gdb/gdb.html>, *Sistema Operativo GNU*; Accedido: agosto 2011.
- [8] Autor: L. Rousseau, “pcsc-tools”, <http://ludovic.rousseau.free.fr/softwares/pcsc-tools/>, Accedido: agosto 2011.
- [9] “Documentation:Linux/GPIO”, http://www.avrfreaks.net/wiki/index.php/Documentation:Linux/GPIO#Interfaces_explained, *AVRFreaks Wiki*; Accedido: agosto 2011.
- [10] “Kernel GPIO”, <http://kernel.org/doc/Documentation/gpio.txt>, Accedido: agosto 2011.
- [11] “termios”, <http://pubs.opengroup.org/onlinepubs/007908799/xsh/termios.h.html>, Accedido: agosto 2011.
- [12] Application Independent CardTerminal Application Programming Interface for ICC Applications; Deutsche Telekom AG / PZ Telesec, GMD - Forschungszentrum Informationstechnik GmbH, TÜV Informationstechnik GmbH, TELETRUST Deutschland e.V., 30.10.1996.
- [13] IFDHandler 3.0 MUSCLE PC/SC IFD Driver API; David Corcoran & Ludovic Rousseau, July 28, 2004.
- [14] *PCSC Workgroup*, <http://www.pcscworkgroup.com/>, Accedido: agosto 2011.
- [15] “PCSC-Lite”, <http://pcsclite.alioth.debian.org/>, *Aliothdebian.org*; Accedido: agosto 2011.
- [16] Autor: H. Welte, “librfd - A Free Software RFID stack,” <http://openmrtd.org/projects/librfd/>, *The OpenMRTD.org Project*; Accedido: agosto 2011.



Diseño e Implementación de Sistema Embebido para Telemetrizar Estaciones Limnimétricas

Micolini, Orlando

Laboratorio de Arquitectura de Computadoras (LAC)
Facultad de Ciencias Exactas, Físicas y Naturales
(FCEFyN)
Universidad Nacional de Córdoba (UNC)
Córdoba, Argentina
omicolini@compuar.com

Alasia, Melisa A.

alasiameili@gmail.com

Martínez, Pablo A.

pabloa.mar@gmail.com

Pereyra, Martín

tinchommp@gmail.com

Resumen—A continuación, se describe un sistema embebido, distribuido, diseñado e implementado para lograr la telemetrización de estaciones limnimétricas; donde el emplazamiento es difícil, puesto que se trata de una zona carente de energía eléctrica y con infraestructura deficiente de comunicaciones. La solución planteada está conformada por un subsistema remoto, emplazado en la región antes mencionada, encargado de la comunicación con la estación; y uno local, ubicado en Córdoba Capital. Se logra un sistema sencillo de implementar, como así también de mantener, fácilmente escalable, robusto, tolerante a fallos y de bajo costo tanto para su construcción, como para su explotación.

Palabras Clave: Límmetro, Sistema Embebido Distribuido, Telemetrización.

I. INTRODUCCIÓN

A. Descripción General

Los niveles de agua medidos por una estación limnimétrica, posibilitan generar series temporales, útiles para la gestión del recurso, como así también para alertas de inundaciones.

Se trabajó sobre la Laguna Mar Chiquita y los Bañados del Río Dulce, una región sensible a cambios acentuados, de complicado acceso, sujeta a elevadas precipitaciones, anegamientos y sin infraestructura básica (energía eléctrica, pavimento, agua).

La telemetrización en tiempo real, permite solucionar el alto costo asociado a la recolección de los datos, realizar un seguimiento del comportamiento de los equipos, sin la consiguiente pérdida de datos. Además, permite la rápida toma de decisiones operativas y suministrar información para generar las acciones correspondientes (avisos de alerta).

La importancia del ambiente bajo estudio radica en que el Río Dulce es el principal afluente a la Laguna Mar Chiquita y conforma uno de los humedales más importantes en Argentina. En el año 1994, fue reconocida como Reserva Natural Provincial de Mar Chiquita y en 2002 fue designada por Ramsar (www.ramsar.org) como Humedal de Importancia Internacional. [2]

En la solución aquí planteada, desde un sistema remoto, se adquieren los datos de los límmetros, permitiéndose además su gestión. Estos datos, son retransmitidos a un Servidor localizado en Córdoba Capital.

La aplicación Servidor cuenta con un Sistema de Gestión de Base de Datos (SGBD), almacenando los datos recibidos desde el sistema remoto y re direcciona avisos de notificaciones ante eventos extremos.

Una aplicación Cliente permite gestionar la información almacenada. [1]

B. Objetivos General y Específicos

Analizar, Diseñar e Implementar los procesos y sistemas requeridos para la telemetrización de estaciones limnimétricas en la región de la Laguna Mar Chiquita y Bañados del Río Dulce.

- Analizar, seleccionar y desarrollar un sistema distribuido que permita la recepción y gestión de los datos.
- Lograr una comunicación robusta y tolerante a fallos.
- Permitir la gestión telemétrica de la estación.
- Incluir un mecanismo de aviso de mediciones ante eventos extremos.
- Posibilitar la inserción y administración de nuevas estaciones (escalabilidad de la red limnimétrica).
- Adoptar una solución que resulte técnica y económicamente viable y sustentable. (Instalación/Mantenimiento).

II. MARCO TEÓRICO

A. Estaciones Limnimétricas

Permiten medir tanto niveles de agua superficiales (Límnímetros), como fluctuaciones de nivel freático (Freatígrafos). Las estaciones limnimétricas OTT Thalimedes, Figura 1, están compuestas por 2 módulos: un codificador angular y una unidad de adquisición de datos.

En cuanto al codificador angular, un sistema de cable de flotador con contrapeso trasmite el cambio del nivel del agua a la rueda del flotador del sensor. La rotación resultante se transforma en una señal eléctrica que se transmite a la unidad recolectora de datos a través del cable del sensor, almacenándose allí como valor medido. Esta transmisión se realiza por medio del estándar SDI-12 (Interfaz Serie-Digital estándar a 1200 baudios para sensores basados en microprocesadores). [3]

Por su parte, la Unidad de Adquisición de Datos permite, mediante su Display, observar el valor instantáneo medido, la marca de tiempo (hora y fecha) y la tensión de la pila del data logger. Posee una memoria circular EEPROM, que permite almacenar hasta 30.000 datos. Cuenta con una Interfaz RS-232 que facilita el diseño de un sistema de comunicación, mediante el Protocolo OTT-Transmission 2.

B. Red Hidrometeorológica

Una red hidrometeorológica está compuesta por estaciones que poseen sistemas de medición y adquisición de datos provenientes de sensores hidrometeorológicos y/o ambientales.

El Laboratorio de Hidráulica de la FCEFyN prevé la consolidación de una red hidrometeorológica en la zona de estudio anteriormente mencionada. La misma quedará formada por las estaciones limnimétricas ya instaladas (Paso de Oscares, Paso de la Cina, La Rinconada y Miramar) y por dos nuevas a ser incluidas (sobre los puentes en las desembocaduras de los Ríos Suquía y Xanaes a la Laguna Mar Chiquita).

C. Tecnologías para el Envío y Gestión de la Información

1) Módem Acústico y ADSL

Se trata de la transmisión de datos a través de la Red Telefónica Básica (RTB), donde existe un único canal



Figura 1: Límnímetro OTT

compartido para voz y datos.

Por su parte, ADSL (Línea de Abonado Digital Asimétrica) modula las señales de datos en una banda de frecuencias más alta que la utilizada en las conversaciones telefónicas. Se denomina Asimétrica, debido a que la capacidad de descarga y subida de datos no coinciden. Necesita la digitalización de la línea RTB. [5]

2) Telefonía Celular

El Sistema Global para las Comunicaciones Móviles (GSM), define una completa arquitectura de red, es un estándar abierto (Asociación GSM) donde la información se digitaliza y comprime para su transferencia. Dicha transmisión, se realiza asignando a cada llamada ranuras de tiempo. Finalmente, permite el empleo del servicio de Mensajes Cortos, SMS. [6]

El Sistema General de Paquetes vía Radio (GPRS), se basa en paquetes de datos que pueden ser enviados a través de la red GSM, pero ofrece un servicio más eficiente para accesos a redes IP. La tarificación se realiza por paquete de información transferido frente al tiempo de conexión de GSM. Existe una conexión permanente con tarificación por tráfico. Los paquetes de datos tienen una longitud constante o fija correspondiente a la ranura de tiempo de GSM.

3) Telefonía Satelital

Los dispositivos se conectan directamente a un satélite de telecomunicaciones. Por lo que resulta útil en zonas donde la telefonía celular o fija no tienen cobertura, o donde la infraestructura de comunicaciones convencionales ha sufrido daños. Su principal desventaja es el elevado costo, tanto en adquisición de los equipos, como en el mantenimiento del servicio.

4) Radio Frecuencia – Packet Radio

Es un sistema de comunicación digital que emplea emisoras de radio. Consiste en el envío de señales digitales mediante la comutación de paquetes que luego son reensamblados en un mensaje completo en el destino final. Permite el establecimiento de redes multihop con soporte para TCP/IP. [4]

III. ANÁLISIS DEL PROBLEMA

En la siguiente tabla, Tabla 1, se resume el Análisis de los condicionantes y tecnologías respecto a cada lugar de emplazamiento del sistema. Las Telefonías Satelital y Celular, además de Packet Radio, son Tecnologías disponibles en todos los puntos del sistema.

TABLA I. ANÁLISIS DE CONDICIONANTES Y TECNOLOGÍAS RESPECTO A CADA LUGAR

Lugares	Energía Eléctrica	Tecnologías Disponibles
Paso de la Cina	No	Telefonía Celular (antenas Yagi)
La Rinconada	SI	RTB
Miramar	SI	RTB y ADSL

IV. SOLUCIÓN GENERAL PLANTEADA

Luego del análisis de factibilidad preliminar y en campo realizados en base a los Requerimientos de usuario (los cuales coinciden con los Objetivos Específicos mencionados) se planteó la siguiente solución, aplicable a todos los casos de estudio.

En la Figura 2, se descompone al Sistema Distribuido propuesto, para su análisis y solución, en dos subsistemas: Sistema Local y Sistema Remoto. Interconectados por un servidor de correo.

A. Sistema Remoto:

En la Figura 3 se detalla al Sistema Remoto, el cual debe:

- Establecer conexión entre la estación y un sistema que permita su manipulación y adquisición de datos.
- En casos donde la intensidad de la señal sea intermitente y/o débil, el sistema remoto debe ser respaldado por antenas direccionales de tipo Yagi-Uda.
- Incluir un mecanismo que permita la interrogación de valores instantáneos y la parametrización de la estación.
- Generar avisos de eventos extremos para mediciones que superen un umbral.
- Incorporar alimentación eléctrica por medio de un panel solar.

B. Sistema Local:

En la Figura 4 se detalla al Sistema Local, el cual debe:

- Implementar una aplicación Servidor que permita la descarga y procesamiento de la información contenida en el servidor de correo, y que almacene dicha información en un Sistema de Gestión de Bases de Datos (SGBD).
- Contar con una aplicación Cliente, que se comunique con

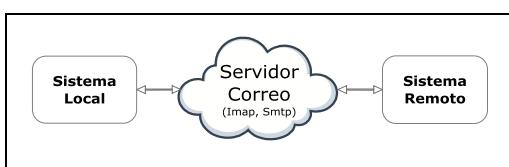


Figura 2: Sistema Solución Planteado

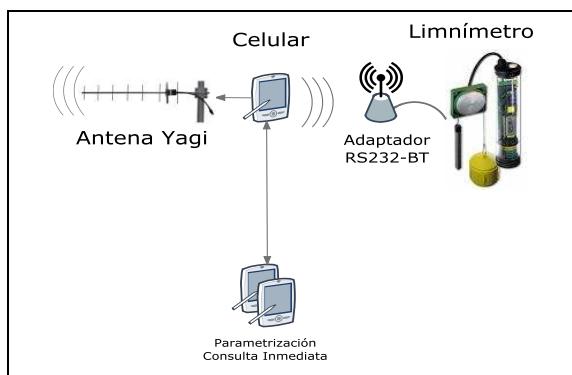


Figura 3: Detalle del Sistema Remoto.

el SGBD para leer niveles, agregar y corroborar estaciones instaladas, o administrar la lista de destinatarios de las notificaciones de eventos extremos.

- Complementar el mecanismo de Aviso de Eventos Extremos, implementando en la aplicación Servidor, el envío de correos a una Lista de Destinatarios de Avisos de Eventos Extremos.

C. Servidor de Correo

Para la recepción de los datos provenientes del sistema remoto, se necesita contar con un sistema que realice el almacenamiento de la información de forma segura y estable. Es por esto, que dicho sistema debe garantizar robustez y disponibilidad de forma continua.

Se decidió utilizar el servidor de correo de Gmail, ya que éste cuenta con las características anteriormente mencionadas; además de ser un servicio gratuito y de fácil mantenimiento.

Cabe destacar que el empleo de un servidor de correo es una técnica que otorga facilidad de reemplazo (por cualquier otro servidor que permita el uso del protocolo de acceso a mensajes de Internet IMAP y el Protocolo Simple de Transferencia de Correo SMTP) y de redundancia; permitiendo incluir nuevos servidores de correo.

Esto permite el diseño de un sistema robusto en cuanto a la comunicación.

D. Consideraciones sobre la Solución en el Sistema Remoto

En el siguiente apartado se estudian aspectos que se consideran claves para comprender la solución hasta aquí planteada y permitir realizar una mejor compresión de los subsistemas desarrollados.

1) Comunicación Inalámbrica entre la Estación y el Dispositivo Remoto

Luego de analizar soluciones posibles, inalámbricas y cableadas, para establecer la comunicación entre la estación limnímetrica y el celular que conforma el sistema remoto, se concluyen las siguientes ventajas de una solución inalámbrica:

- Permite desacoplar el módulo de comunicación (con su respectiva antena) de la estación limnímetrica; la cual suele encontrarse instalada sobre puentes, por los que

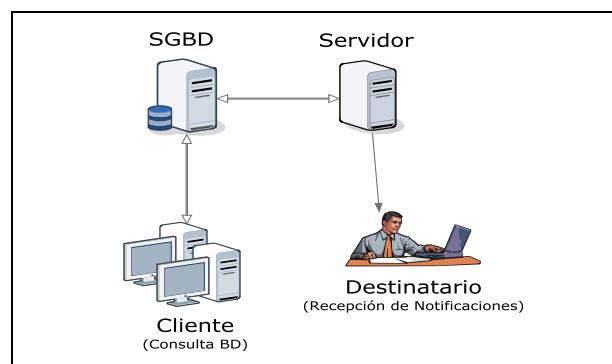


Figura 4: Detalle del Sistema Local.



circulan vehículos cuyas dimensiones pueden exceder sus estribos. Ejemplos: cosechadoras, fumigadoras, etc.

- Representa una forma de protección del equipo, ya que éste puede emplazarse en sectores no inundables, resguardados del viento, de difícil acceso, etc.
- En contraste con redes cableadas, no presenta el riesgo de quedar inutilizable ante cortes del enlace (vandalismo, corte por vientos, arrastre por inundaciones) y al no tener que colocar cables físicos, la instalación puede ser más rápida.
- Para distancias relativamente cortas, las interferencias no deberían ser un problema.
- La instalación del módulo de comunicación no implica la modificación del gabinete donde se encuentra la estación.
- El limnímetro no queda eléctricamente conectado al módulo de comunicación. Esto permite protegerlo, considerando que tiene un costo económico y funcional altamente superior a este último.

2) Evaluación del Hardware para el Dispositivo Remoto

A la hora de seleccionar el hardware necesario para llevar a cabo el sistema remoto, y recordando que éste debe disponer de interfaz bluetooth, tecnología GSM/GPRS, capacidad de procesamiento de datos; surgen distintas posibilidades. Entre ellas, el empleo de módems GSM/GPRS, microcontroladores, Single Board Computers (SBCs), Field Programmable Gate Array (FPGAs), teléfonos celulares.

Finalmente, se decidió trabajar con un teléfono celular por los motivos que se exponen a continuación:

- Modem GSM/GPRS integrado.
- Sistema Operativo Embebido, con sus drivers y APIs de acceso, que garantizan un mejor aprovechamiento del hardware y la posibilidad de programar en alto nivel.
- Interfaz Bluetooth con Serial Port Profile.
- Menor costo, tanto económico por su alta disponibilidad en el mercado a bajo precio; como en desarrollo, ya que disminuye la necesidad de integrar módulos (Modem GPRS, Microcontrolador, interfaz de alimentación, etc).

Cabe destacar que se han considerado las siguientes desventajas, aunque las mismas no resultan críticas:

- Para poder acoplar una antena Yagi-Uda a la antena interna del celular, se necesita usar un adaptador por inducción, que produce una pérdida de ganancia (aprox. 3 dB).
- El celular, a diferencia de una computadora convencional, no posee un sistema de arranque ni reinicio automático.

3) Elección del Equipo Celular

Para llevar a cabo la selección del teléfono celular, en primera instancia, se hizo un análisis de los Sistemas Operativos de los equipos disponibles en el mercado: Android, Windows CE y Symbian.

Se optó por el sistema operativo Symbian, ya que éste presenta como interfaz de programación al Framework multiplataforma Qt C++, que permite portar el software desarrollado a diferentes sistemas operativos, como Android y Windows CE.

Finalmente, se decidió utilizar un equipo Nokia 5230, ya que presentaba el menor costo económico entre los dispositivos disponibles de estas características.

V. REQUERIMIENTOS DE SISTEMA

Para implementar la solución general planteada, es necesario desarrollar tres aplicaciones: Una Cliente-Servidor ubicada en el subsistema local, y otra Celular instalada en el dispositivo remoto.

A continuación, se describen los requerimientos de las distintas aplicaciones: funciones, servicios proporcionados y restricciones operativas.

A Requerimientos de la Aplicación Cliente

Permite a los usuarios finales realizar consultas pertinentes de los valores de las estaciones. Instancias de ésta, pueden ser instaladas en computadoras locales del Laboratorio de Hidráulica, teniendo acceso concurrente a los datos, como se puede observar en la Figura 4. Los requerimientos de sistema para esta aplicación son:

- Conexión SGBD: Establecer una conexión con un SGBD a fin de realizar consultas de niveles, las cuales pueden ser por estación, por intervalo de fechas o por máximos, mínimos y promedios mensuales y anuales; como así también registros y consultas de estaciones.
- Consultas y modificaciones de destinatarios de notificaciones
- Interfaz Gráfica: Brindar una interfaz gráfica de lectura de parámetros y niveles medidos de las distintas estaciones limníméticas. También debe permitir el registro de nuevas estaciones y la administración de una lista de destinatarios de notificaciones de eventos extremos.
- Creación de un archivo de Texto: Generar un archivo de texto según la consulta de niveles deseada.

B. Requerimientos de la Aplicación Servidor

Se encarga de la gestión y almacenamiento de los datos recibidos del sistema remoto, como puede verse en la Figura 4. Se debe instalar una sola instancia de la misma, y garantizarle acceso a internet. Los requerimientos de sistema para esta aplicación son:



- Descarga Correo Entrante: Crear un canal de conexión TCP/IP con un servidor de correo, que permita la descarga de correo entrante.
- Actualización de la BD: Establecer una conexión con un Sistema de Gestión de Base de Datos para almacenar lecturas de parámetros y niveles (contenidas en el correo entrante).
- Envío de Notificación de Eventos Extremos: Establecer una comunicación con el servidor de correo a fin de enviar avisos (contenidos en el correo entrante) a los destinatarios (almacenados en base de datos).

C. Requerimientos de la Aplicación Celular

Lleva a cabo la gestión y comunicación con la estación limnímetrica y la trasmisión de valores medidos y de funcionamiento. También se encarga de la integridad de los datos y de la generación de notificaciones de eventos extremos (superan umbrales). Se muestra en la Figura 3. Los requerimientos de sistema para esta aplicación son:

- Conexión Bluetooth: Establecer una comunicación bidireccional con el limnímetro utilizando Serial Port Profile.
- Ejecución de Comandos: Permitir la ejecución de los siguientes comandos del protocolo Ott Transmission 2: Iniciar/Abortar conexión, Pedido/Escritura de parámetros del equipo.
- Verificación de Integridad de datos: Controlar la integridad de los datos transmitidos desde y hacia el limnímetro, para disminuir retransmisiones; procesando las tramas, controlando y generando los checksums pertinentes.
- Parametrización: Permitir la configuración de parámetros de funcionamiento, a través de mensajes entrantes de texto.
- Lectura Inmediata: Poder aceptar peticiones de lectura de parámetros y nivel por medio de mensajes de texto, retornando las respuestas de modo inmediato.
- Transmisión Periódica de Valores: Establecer una conexión con el servidor de correo a fin de enviar periódicamente los valores medidos y parámetros de funcionamiento.
- Notificación de Eventos Extremos: Comparar un valor de nivel leído con el valor de umbral seteado y, en caso de ser necesario, generar un aviso. El cual será enviado por mail y por mensaje de texto. [1]

VI. IMPLEMENTACIÓN Y PRUEBAS

A continuación se ilustran las secuencias que se consideran representativas de las distintas funcionalidades del sistema.

A. Recepción de Mediciones Provenientes de la Estación

En la Figura 5 se muestra cómo es almacenada en la Aplicación Servidor, una nueva secuencia de mediciones recibida desde el sistema remoto y actualizada para la visualización del usuario, quien la solicita inicialmente, en la Aplicación Cliente.

B. Lectura Inmediata de la Estación

En la Figura 6, se observa la secuencia llevada a cabo por el sistema cuando el usuario requiere una lectura del valor inmediato sensado por la estación.

C. Parametrización del Sistema Remoto

La Figura 7 muestra los pasos necesarios para que sea posible llevar a cabo la parametrización del sistema remoto, ya sea, del dispositivo celular, como así también de la estación limnímetrica.

D. Generación de Avisos de Eventos Extremos

La Figura 8, permite visualizar cómo se desencadena el mecanismo de avisos a la lista de destinatarios almacenada en

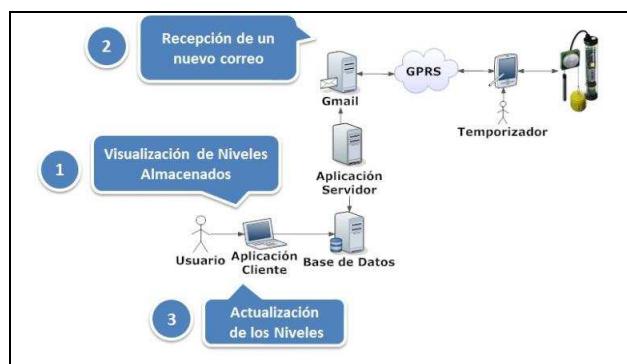


Figura 5: Recepción de mediciones provenientes de la Estación.

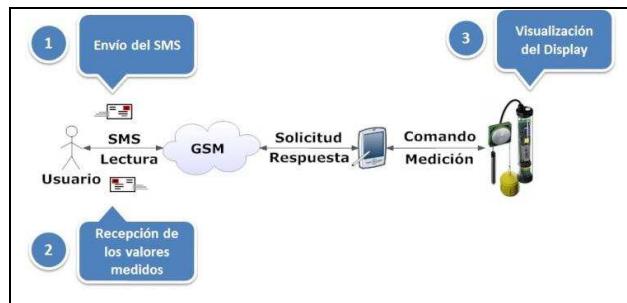


Figura 6: Lectura Inmediata de la Estación.

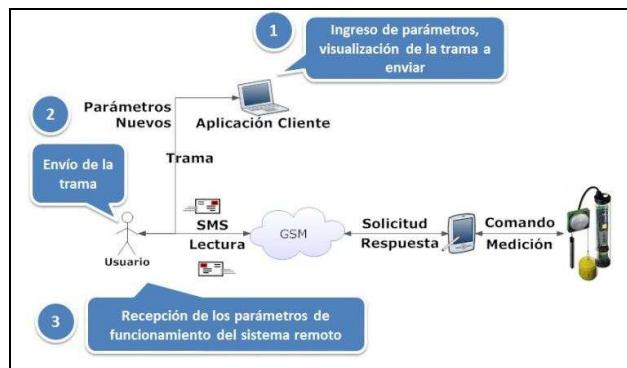


Figura 7: Parametrización del Sistema Remoto.

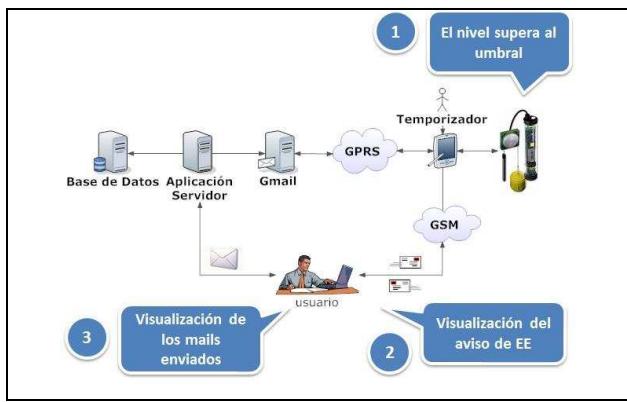


Figura 8: Generación de Avisos ante Eventos Extremos.

la Aplicación Cliente, una vez que se produce algún evento extremo (mediciones que superan los valores fijados como umbral).

En relación a la resistencia a la intemperie del sistema remoto, tanto las estaciones como los dispositivos celulares, se encuentran ubicados en gabinetes diseñados y preparados para soportar cambios bruscos de temperatura, adversidades climáticas (como fuertes vientos, lluvias o granizo y piedra), como así también ataques de vandalismo, entre otros; lo cual es facilitado por la comunicación inalámbrica (bluetooth) entre el dispositivo celular y la estación limnimétrica.

En lo referente a la alimentación eléctrica, sólo La Rinconada y Miramar poseen suministro; no así Paso de Oscares, Paso de la Cina y los puentes de Río Suquía y Río Xanaes. Este factor no fue considerado un problema en el desarrollo del proyecto, ya que el Laboratorio de Hidráulica cuenta con Módulos Fotovoltaicos de la empresa Solartec, que han sido colocados en cada estación. Los mismos son usados con baterías CP 12240 de 12V y 24 A/h. El panel KS10 en su máxima entrega, carga su batería en aproximadamente 48hs (de exposición solar); mientras que el KS20, lo hace en la mitad del tiempo.

El celular tiene un consumo a máxima potencia continua de aproximadamente 1 A/h. Comparando estos consumos, se puede concluir que el sistema de alimentación está sobredimensionado para cumplir con los requerimientos de consumo del teléfono. La autonomía observada es de 15 días, en el peor de los casos. [1]

VII. CONCLUSIONES

La importancia del desarrollo de la presente solución radica en el establecimiento del canal de comunicación entre los sistemas local y remoto, debido a las adversidades tecnológicas de la zona de estudio, como lo son: la falta de energía eléctrica, de señal celular, el complicado acceso a la zona, entre otras.

Por la característica modular y la simplicidad, tanto del software como del hardware, y la distribución del sistema, es posible ajustar la implementación a soluciones particulares según los requerimientos que la ubicación de las estaciones limnimétricas demanden; y de acuerdo a factores

(económicos, tecnológicos, necesidades temporales y de mantenimiento) que se consideren prioritarios.

Cabe aclarar que al añadir el adaptador Bluetooth al limnímetro, el dispositivo interrogador puede ser cualquier equipo que disponga de capacidad de procesamiento necesaria y Perfil de Puerto Serie Bluetooth.

Por ejemplo: en el caso de Miramar, es posible reemplazar el Celular por una PC con acceso a internet, que mediante Bluetooth se comunique con la estación. Esto es viable, ya que las aplicaciones son desarrolladas con Módulos Multiplataforma y Protocolos Estándares.

Además, el sistema de comunicación posibilita la interrogación de cualquier tipo de dispositivo final (a consultar) que disponga de Bluetooth o permita su incorporación (a través de un adaptador, como en este caso).

VIII. APORTES

Se logra un sistema flexible respecto a todas sus partes, debido a que sus componentes son estándares, como así también sus interfaces.

El software, al ser multiplataforma, corre tanto sobre PCs, celulares o sistemas embebidos.

En cuanto al hardware y los sensores, pueden ser cualquier dispositivo que posea Bluetooth, con soporte para perfil de puerto serie, o permita su incorporación (como en este caso).

En lo que respecta a la viabilidad económica, permite seleccionar y reusar los componentes más accesibles y disponibles, tanto de medición, comunicaciones y procesamiento.

En cuanto a la comunicación, es posible elegir la tecnología, según disponibilidad o conveniencia económica, como así también, el proveedor de servicios de internet y de correo.

IX. REFERENCIAS

- [1] M. A. Alasia y P. A. Martínez, «Análisis, Diseño e Implementación de la Telemetrización de Estaciones Limnímetricas: Laguna Mar Chiquita y Bañados del Río Dulce,» FCEFyN - UNC, Córdoba, Argentina., 2011.
- [2] Universidad Nacional de Córdoba, Universidad Nacional Santiago del Estero y Subsecretaría de Recursos Hídricos, «Estudio hidrológico-Ambiental del Sistema Mar Chiquita-Bañados y tramo inferior del Río Dulce,» Córdoba, Argentina, 2007.
- [3] OTT, «Instrucciones de Servicio. Cordificador Angular con Recolector de Datos Thalimedes.,» Kempten, Alemania..
- [4] Oscar Somarriba, «Redes de Paquetes por Radio Multihop.» Departamento de Sistemas Digitales y Telecomunicaciones - Universidad Nacional de Ingeniería, Apdo. 5595. MANA..
- [5] Mtpropertiesllc. [En línea]. Available: <http://mtpropertiesllc.com/youths-how-to-set-up-a-speedstream-5100-adsl-modem/>. [Último acceso: 2011].
- [6] GSM Association. [En línea]. Available: <http://www.gsmworld.com>. [Último acceso: 2011].

Sistema georeferenciador con parcelamiento virtual y adquisidor de sonidos masticatorios de rumiantes en pastoreo extensivo

Aranda, Gonzalo D., De La Cruz Arbizu, Federico
Laboratorio de Comunicaciones
Universidad nacional de Mar del Plata
Mar del Plata, Argentina
Email: fede.dlca@hotmail.com, gonzalo.d.ar@hotmail.com

González, Esteban L., Tulli, Juan C., Uribarri, Alejandro,
Agüero, Pablo D.
Laboratorio de Comunicaciones
Universidad nacional de Mar del Plata
Mar del Plata, Argentina
Email: [\(elgonzal,jctulli,ajuriz,pdagüero\)@fimdp.edu.ar](mailto:(elgonzal,jctulli,ajuriz,pdagüero)@fimdp.edu.ar)

Abstract - Este trabajo describe el hardware electrónico de un dispositivo desarrollado originalmente para el registro de sonidos masticatorios del ganado en pastoreo extendido y que durante su desarrollo fue modificado para permitir registrar también la posición del animal y/o guiarlo a determinadas áreas de pastoreo. Diversos estudios con animales estabulados han demostrado que existe una correlación entre los sonidos producidos durante el arranque, la masticación y la rumia, y la ingesta del animal. Se estima en consecuencia que a partir de estos sonidos se pueden determinar aspectos tales como el engorde o el rendimiento que se obtendrá del ganado. Para facilitar la etapa de análisis de sonidos posterior, se decidió georeferenciar al animal (asumiendo que se conocería la ubicación de las distintas pasturas). Surgió entonces que para fines de estudio sería útil poder forzar al animal a alimentarse de un determinado tipo de pastura. Una opción al parcelamiento con alambre o al uso de "boyeros" es el uso de parcelas virtuales. Esto se logra modificando el software de la etapa de georeferenciación de manera de que pueda calcular si el animal está dentro, o acercándose al borde, de un perímetro virtual. Y en este último caso, generar algún tipo de estímulo que lo persuada de permanecer adentro. Adicionalmente, en algunas regiones, las áreas en donde el ganado, principalmente ovino, puede pastar libremente se están desertificando, esto implica que el manejo de estas tierras deba ser llevado a cabo con sumo cuidado. En consecuencia, este dispositivo permitirá, por una parte, obtener información en condiciones normales de pastoreo, dando una herramienta muy útil al ingeniero agrónomo, investigador o productor para determinar la cantidad de alimento ingerido o los requerimientos de forraje necesarios para suministrarle al ganado una correcta alimentación. Por otra parte este dispositivo, sin su etapa de registro de sonido, podrá confinar al animal en áreas específicas sin necesidad de cercas físicas.

Keywords-component; Georeferenciación. Microcontroladores, Ganado, Adquisición de sonido

I. INTRODUCCIÓN

En este reporte se describe la implementación de un Sistema de georeferenciación con parcelamiento virtual y adquisición de sonidos masticatorios en rumiantes en pastoreo extensivo, basado en microcontroladores PICs. Estos sonidos y las posiciones registradas, permitirán en un posterior análisis, determinar la ingesta y el tipo de pastura que consume el animal. Estos datos son sumamente importantes para la estimación del engorde por parte del ingeniero agrónomo o productor.

El dispositivo desarrollado cuenta con dos canales de audio para la adquisición de sonido, que serán almacenados en una tarjeta de memoria SD, y un sistema de georeferenciación mediante un módulo comercial GPS. Con este módulo se registra la ubicación del animal, se implementa el sistema de parcelamiento virtual y se obtiene una referencia horaria precisa para etiquetar los audios y las posiciones. Además se añadió al desarrollo una interfaz de programación para que el usuario establezca sobre el dispositivo los parámetros básicos para el estudio buscado.

La posibilidad de utilizar dos canales de audio para la adquisición del sonido permitirá, en estudios posteriores, una reducción de ruido ambiental por medio de correlación cruzada entre canales, técnica muy utilizada en este tipo de situaciones [1]. El sistema propuesto permite a su vez, el constante registro de la posición del animal, muy importante para determinar el tipo de pastura, como también una constante verificación para establecer si el animal se encuentra dentro de la zona predefinida - parcela virtual -. En caso de estar acercándose a los límites de ésta, una serie de estímulos (primero auditivos y luego eléctricos) lo guiarán nuevamente al área permitida [2]. Los módulos receptores de señal GPS usados en este dispositivo poseen una precisión cercana a los 2 metros.

Por último, respecto al almacenamiento, se resalta la generación de archivos de audio tipo WAVE [3], es decir, sin compresión temporal ni frecuencial, y archivos de posicionamiento tipo KML que permitirán la visualización de



la ruta del animal con etiquetas horarias usando el software Google Earth [4] en su versión gratuita.

II. DEFINICIÓN DEL PROBLEMA Y ESTADO DE LA CUESTIÓN

Desde hace tiempo se han comenzado a aplicar distintos tipos de recursos tecnológicos al estudio de los comportamientos rutinarios de rumiantes, en busca de la optimización en el uso de recursos y alimentos disponibles. Algunas de estas tecnologías incluyen el uso de cámaras de video y micrófonos, con los cuales se puede realizar un registro de los ruidos producidos por el animal cuando se alimenta. Numerosas publicaciones dan cuenta de la vinculación que existe entre los sonidos producidos por el arranque, la masticación y rumia, con la ingesta del animal [5] [6] [7]. A su vez, estos estudios se realizaban principalmente sobre animales estabulados (en establos). La necesidad de usar animales estabulados se manifiesta, en principio, para lograr una correcta correlación entre el sonido y la ingesta, ya que esta última se conoce suministrándole al animal cantidades muy precisas de alimento. Pero superada esta etapa, en donde se desarrollaron fundamentos e instrumentos para deducir la ingesta del animal a partir del sonido, se hace deseable automatizar el proceso de registro, es decir, reemplazar los registros en establos por un dispositivo portátil, que brinde una optimización del proceso, pudiendo adquirir los datos en forma automática en un plazo que abarque desde algunos minutos hasta días completos en condiciones normales de pastoreo [8] [9].

En cuanto al área de pastoreo, existe un consenso generalizado entre los expertos y en los diversos organismos técnicos provinciales, nacionales e internacionales, acerca de que el fenómeno conocido como "desertificación" se debe, al menos en la Patagonia Argentina, al sobreuso de los suelos que, asociado con factores climáticos, llevó a un agotamiento del recurso natural, base de la explotación extensiva del ganado ovino por más de cien años [10].

En vista que la necesidad de una referencia horaria precisa y el conocimiento del tipo de pastura que el animal ingiere hace necesario la utilización de un sistema de georeferenciación, surge casi espontáneamente la posibilidad que el dispositivo también pueda guiar el ganado hacia determinadas áreas de pastoreo. Prototipos de estos sistemas ya han sido probados con éxito [11] [12]. El sistema de registro de sonidos puede, en consecuencia, ser potenciado considerablemente con la utilización de un sistema de posicionamiento global (GPS) para proveer información precisa a través de datos geográficos, sobre el tipo de pastura de la cual se alimenta el animal (teniendo en cuenta que la posición de esta pastura es conocida) y los horarios, dado que se dispone en el GPS de un patrón de tiempo preciso.

III. EL DISPOSITIVO

El sistema desarrollado consta de dos partes,

- Un dispositivo adquisidor de sonidos (que será portado por el animal) con georeferenciación y capacidad para producir estímulos de guía de ganado,

que a su vez está formado por dos subsistemas (subsistema I y subsistema II)

- Y un dispositivo de acceso (a ser conectado al adquisidor) o HMI (Human Machine Interface) que permitirá al usuario del sistema programar un número de parámetros en el dispositivo adquisidor

El subsistema I está construido alrededor de un PIC 18F26K20 [13], mientras que el subsistema II utilizará un PIC 18F25K20 [13].

A. Subsistema I

Esta etapa está encargada de la adquisición de audio y la georeferenciación, además de todo lo que se refiera a la generación de los archivos asociados. Lo que también implica el completo control y responsabilidad de la tarjeta de memoria SD. Esto último es realizado mediante el módulo SPI del microcontrolador y algunos terminales auxiliares que son conectados al zócalo de la tarjeta de memoria.

1) *Adquisición de audio:* En lo que a audio respecta, se disponen de dos canales, que por medio del filtro antialiasing, están conectados a las dos primeras entradas analógicas del PIC 18F26K20; la configuración programada por medio del HMI definirá si se usarán ambos o solamente alguno de ellos. Los archivos se guardan en formato WAVE.

a) *Módulo ADC:* Interfaz fundamental a la hora de trabajar con variables analógicas. El módulo ADC del PIC 18F26K20 consta con la posibilidad de trabajar con 13 canales de forma independiente entre sí. El funcionamiento de dicho módulo se basa en una retención de orden cero y un conversor por aproximaciones sucesivas. En el caso de este trabajo se usaron 256 niveles (8 bit's) que fueron considerados aceptable. Sin embargo, en caso de necesidad, el software del dispositivo es fácilmente configurable para permitir una grabación a 10 bits. El rango de 256 niveles pueden ser fácilmente programable entre las tensiones del microcontrolador como GND (0x00) y Vcc (0xFF).

b) *Frecuencia de muestreo:* El Teorema de Nyquist [14] asegura que, si una señal de banda limitada es muestreada a una frecuencia de por lo menos el doble de su máxima componente en frecuencia, entonces es posible recuperarla únicamente. En consecuencia, sabiendo que el sonido captado no posee una respuesta espectral de banda limitada, surge la necesidad de la utilización de un filtro antialiasing o antisolapamiento de modo tal de evitar la distorsión espectral consecuencia del submuestreo.

Numerosas publicaciones [5] [6] [7] indican que el contenido espectral útil para este tipo de estudios se encuentra por debajo de los 4KHz, por lo que fue elegida una frecuencia de muestreo de 10KHZ.

La solución implementada en el software para lograr un registro de tiempo preciso en el que no se produjeran corrimientos que afecten la relación temporal entre muestras, se llevó a cabo por medio de la interrupción por desbordamiento TIMER0, en la cual se definen tanto el reloj de incremento y el tamaño de pila. De esta manera se ha

logrado una adquisición pareja y sin "jitter", lo cual resultaría muy molesto a la hora del análisis.

c) Formato WAVE [3]: En este proyecto se tuvo en cuenta que la finalidad de los audios adquiridos es analizarlos espectralmente con programas específicos. Es por esto que se decidió la utilización de archivos WAVE (.wav), en los cuales las componentes espectrales no se ven afectadas (no hay compresión), y se mantiene una copia fiel de la realidad dentro del espectro de frecuencia que el muestreo permite.

El formato es muy simple y se conoce como el formato canónico de un archivo WAVE. Su composición está basada en un encabezado, una zona de formato (fmt), donde se caracteriza al archivo, y la sección de datos en donde se encuentran las muestras adquiridas a la velocidad predeterminada. En este dispositivo, el usuario tendrá la posibilidad de elegir tanto la frecuencia de muestreo como el número de canales de adquisición.

d) Identificaciones archivos .wav: La propuesta es generar la cantidad de archivos por día según el usuario lo requiera con la siguiente nomenclatura: ddmmaaa##.wav. Lo cual simplifica la idea de día, mes, año y número del archivo generado en el día.

2) Georeferenciación: La georeferenciación hace referencia a la verificación constante de la ubicación del animal en estudio o análisis, de manera de poder verificar su posición respecto de la parcela y para que el sistema pueda elaborar un registro preciso de la ingesta del animal en pastoreo extendido. La manera de llevar esto a cabo es por medio de una comunicación fluida entre ambos subsistemas (I y II). El subsistema I genera los archivos de audio georeferenciados (por medio de archivos KML compatibles con el software Google Earth), y el subsistema II tiene la comunicación directa con el módulo GPS y es el encargado de la adaptación de la información de este modulo a coordenadas UTM (Universal Transverse Mercator) [15]. El módulo GPS utilizado para la recepción fue el LS20033 de LOCOSYS [17] y usa el protocolo NMEA [16] para su comunicación. Éste permite una actualización de información de hasta 5Hz y una precisión de hasta 2 metros.

a) Archivo .kml [4]: Al igual que los archivos WAVE, el sistema almacenará archivos de formato KML en los cuales se presentará, con las etiquetas horarias correspondientes, el recorrido del animal en el transcurso de la adquisición del sonido. Este tipo de archivos está diseñado para poder abrirlo y analizarlo en el software libre Google Earth.

3) Almacenamiento: Para el almacenamiento de la información se optó por utilizar tarjetas de memoria SD. Éstas son dispositivos de almacenamiento Flash muy versátiles debido a su gran capacidad y reducido tamaño y precio. Desde el punto de vista técnico, la comunicación y proceso de grabación será establecido por el

microcontrolador PIC a través del SPI (Serial Peripheral Interface) cuya interfaz se implementa mediante el módulo MSSP (Synchronous Serial Port) interno del PIC. Si bien esto suena relativamente sencillo, el sistema operativo que posteriormente va a encargarse de leer la tarjeta exige estándares a la hora de administrar los datos que deberán ser respetados a la hora de leer el dispositivo.

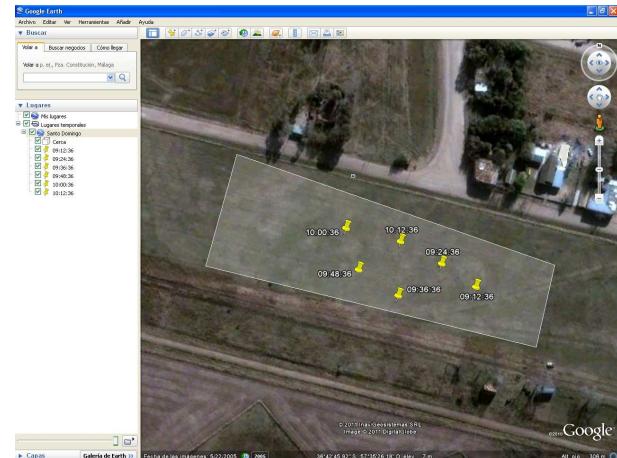


Figura 1. Captura de pantalla de una simulación de una adquisición de sonido y posición

B. Subsistema II

Esta etapa, construida alrededor de otro PIC, está encargada de la importación de datos desde el HMI, provisión de los mismos al subsistema I y el parcelamiento. Esto último implica también la conectividad con el módulo GPS, la manipulación de los datos del mismo y el protocolo de comunicación PIC a PIC.

a) Conversión de coordenadas: Toda coordenada que ingrese al dispositivo, ya sea a través del HMI o del GPS en sí, será sometido a una conversión de sistema para su mejor manipulación. La representación y control del espacio geográfico, resulta más simple si los datos procesados por el sistema de posicionamiento son representativos de un espacio rectangular, es decir, coordenadas cartesianas. Como la tierra es un geoide, la forma de expresar las posiciones terrestres por el GPS es del tipo geodésica, es decir, están representadas por variables como la latitud, la longitud, y la altura geodésica. Pero representar diferentes áreas de la superficie en un sistema polar trae complicaciones, debido a que dichos valores entregados por el GPS no solo dependen del lugar en la tierra donde se encuentre, sino que no se puede hallar una relación lineal entre dos puntos diferentes. Se podría decir entonces que las coordenadas rectangulares se han creado para facilitar la ubicación en el terreno, ya que el sistema de coordenadas geodésicas angulares tradicionales (grados, minutos y segundos) no es intuitivo y no permite realizar de una forma fácil las estimas de las distancias que separan unos puntos de los otros. Por ende, se elige convertir

el espacio geodésico en un sistema de coordenadas rectangulares definido como UTM.

Para traducir coordenadas geodésicas en UTM y viceversa existen diversos procedimientos. El método utilizando para este trabajo está basado en las denominadas fórmulas de Coticchia-Surace [18], por su facilidad para ser programado en el PIC. La precisión que se puede obtener ronda el centímetro cuando se utilizan suficientes decimales. En consecuencia, es imperativo que a la hora de programar se utilicen variables de punto flotante y doble precisión.



Figura 2: Prototipo del adquisidor a ser montado en el animal

b) *Parcelamiento:* Este equipo puede crear, manipular y controlar una cerca virtual, limitando dinámica e implicitamente el terreno. Es una forma adecuada de reducir el trabajo de tendido de un alambrado real y permitir un uso eficiente de los terrenos. Para lograr el parcelamiento debe interpretarse al espacio geográfico como un sistema en donde pueda efectuarse algún tipo de comparación que sea independiente de los parámetros geográficos ingresados. Básicamente el sistema usa cuatro coordenadas cartesianas, ya que las coordenadas geodésicas fueron transformadas a coordenadas UTM. Por lo tanto, de forma intuitiva, surge que la manera de lograr esto es uniendo dichos puntos por medio de líneas rectas, es decir, calculando cuatro pendientes que corresponden a cuatro rectas diferentes.

Las cuatro ecuaciones, representan una sección infinitesimal del espacio, formando así la parcela. Obviamente, dichas rectas poseen una pendiente determinada, dependiente de los puntos a unir.

Por otro lado se tiene un par de coordenadas obtenidas del GPS, las cuales se debe verificar que estén dentro de esa área delimitada por las cuatro rectas.

En la Figura 3, el placemark representa el valor entregado por el GPS, es decir, representa un par de puntos (E,S) en el espacio.

Datos con lo que se dispone

GPS

X = E: coordenada este;

Y = S: coordenada sur;

Pendientes

$$m1 = (s1 - s2) / (e1 - e2) \text{ pendiente 1}$$

$$m2 = (s2 - s3) / (e2 - e3) \text{ pendiente 2}$$

$$m3 = (s3 - s4) / (e3 - e4) \text{ pendiente 3}$$

$$m4 = (s4 - s1) / (e4 - e1) \text{ pendiente 4}$$

Ecuaciones:

$$Y1 = m1 * (E - e2) + s2$$

$$Y2 = m2 * (E - e3) + s3$$

$$Y3 = m3 * (E - e4) + s4$$

$$Y4 = m4 * (E - e1) + s1$$

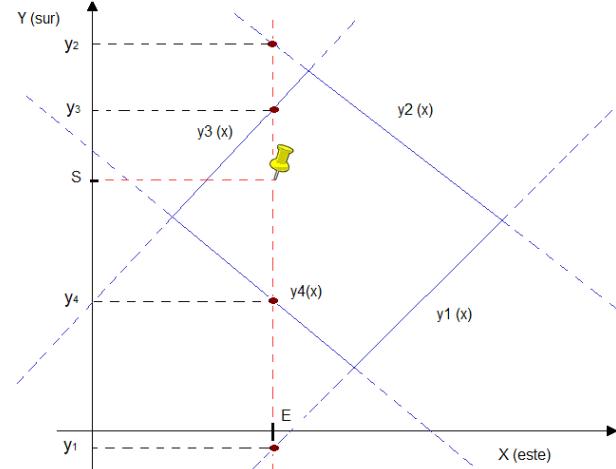


Figura 3: Placemark en proceso de parcelamiento

La variable independiente 'E' es evaluada en cada una de las rectas, obteniendo así sus correspondientes valores de ordenada, en el gráfico expresadas como y_1 , y_2 , y_3 e y_4 . Una vez que el programa calcula dicho valores, recurre a obtener las desigualdades, es decir, a comparar el valor de ordenada del GPS, S, con los valores de ordenadas arrojados por la correspondiente evaluación de E en las rectas. La desigualdad o condición de seguridad, es decir, aquella en la que el animal se encuentra dentro de la cerca, es la siguiente:

$$S > Y1 \& S < Y2 \& S < Y3 \& S > Y4$$

Una vez que el programa comprueba que dicha situación es correcta, el mismo retorna al inicio del lazo y comienza nuevamente el testeo. En caso que la condición de comparación no se cumpla, el subsistema activa la estimulación pautada, de manera de lograr que el animal retorne a la zona segura. Al mismo tiempo, se pueden establecer parcelas internas de seguridad que "flexibilizan" el momento y tipo de estímulo sobre el animal en relación a qué tan cerca del límite del área virtual se encuentre. Pero ello quedará a elección del usuario.

IV. INTERFAZ HMI

El diseño de esta interfaz surge de lo conversado directamente con los posibles usuarios o destinatarios de este proyecto, o sea los ingenieros agrónomos o productores. Ellos necesitaban “una interfaz robusta y de simple manejo para configurar al sistema principal con las características deseadas”.

Tanto software como hardware fue acondicionado para tal fin, componiéndose el HMI de una sección principal constituida por un microcontrolador PIC18F2620, un sistema de representación gráfica montado por medio de un display LCD (CCM-2040CSL de CASIL), un teclado básico con diseño intuitivo para la navegación del dispositivo, una compuerta de alimentación externa a través de un conector USB y la comunicación RS232 con el registrador.

Los parámetros a ser configurados por el usuario son:

- **Memoria utilizada:** debido a que el sistema puede trabajar con memorias de 1GB, 2GB y 4GB
- **Número de canales:** Uno o dos canales, dependiendo de las condiciones de ruido ambiental
- **Duración de los tiempos de registro de audio:** en múltiplos de media hora, el usuario podrá definir la duración estimada del audio generado por el sistema.
- **Tiempo entre archivos de audio:** se podrán definir de a múltiplos de 5 minutos
- **Número de archivos de audio:** de a múltiplos de a 5 se podrá definir la cantidad de archivos que se deseen generar. (limitado por el tamaño de la memoria)
- **Número de posiciones:** aquí, se define el número de posiciones o placemark's que se desean observar en el archivo tipo KML generado por cada archivo de audio, aumentando de a 4 el número buscado.
- **Definición de la cerca virtual:** por último, podrán ser elegidos libremente por el usuario los 4 vértices del área de parcelamiento.

El HMI brinda también la posibilidad de usar la memoria EEPROM del PIC18F2620 para guardar datos del usuario aún después de apagar el equipo. La necesidad surgió de las pruebas de un primer prototipo en donde se notaron las molestias que generaría a los usuarios tener que ingresar siempre los mismos datos al HMI, pudiendo incurrir en posibles errores o el olvido de ingreso de ciertos parámetros. Es por esto que se introduce la característica de que los datos ingresados no se eliminarán una vez desconectado el equipo, sino que serán almacenados en memoria para poder utilizarlos las veces que sean necesarias.

V. CONCLUSIÓN

A lo largo de este informe se presenta un sistema que, habiendo nacido como un simple adquisidor de sonidos, evolucionó para convertirse también en un dispositivo de control de ganado.

El dispositivo logrado no solo cumple con los requerimientos iniciales sino que agregó nuevas posibilidades y además presenta un diseño fácilmente modificable de acuerdo a los

requisitos de futuras investigaciones, tales como aumentar la cantidad de canales o la cantidad de bits por muestra. Con la configuración aquí mostrada (frecuencia de muestreo y bits por muestra) este dispositivo puede registrar en forma continua casi 48 horas de sonidos en dos canales sobre una memoria de 4 GB, tiempo que se puede ampliar considerablemente introduciendo espacios entre secciones de adquisición. Pasado este lapso se deberán descargar los datos a una PC o Notebook. En función de que esto implica una acceso al sistema en forma periódica, no se consideró la necesidad de analizar el consumo de la batería ya que esta sería fácilmente reemplazable.

En este informe se describe un prototipo, el dispositivo final no sería muy diferente en tamaño en virtud que el hardware es el mismo. Variaría desde ya, la caja de contención que tendría que ser resistente a golpes e intemperie y su ubicación en el cuello del animal deberá ser realizada mediante un collar cuyo diseño no fue considerado en este diseño.

Sin la etapa de adquisición de sonido, el sistema, mas económico, puede ser usado en mayor número solo para control de ganado, modalidad que se está afianzando en países con áreas de pastoreo sensibles a desertificación como Australia o Argentina o con geografías poco aptas para el tendido de alambres (Suiza).

El haber desarrollado prácticamente la totalidad del software del dispositivo desde cero, o sea sin incluir a librerías estándares, permitió a los proyectistas adquirir un conocimiento profundo de cada aspecto del diseño, ya sea desde la programación de los PICs y el manejo de la memorias, GPSs y la comunicación entre ellos hasta la matemática utilizada en la determinación de la posición en coordenadas UTM.

Este proyecto puede entonces, no solo ser considerado como un producto terminado, sino como el chasis para futuras mejoras o desarrollos. El desarrollo de la tecnología Wi Fi, la inserción en la red GSM y el consecuente abaratamiento de sus componentes hacen que una futura versión de este sistema prevea la comunicación inalámbrica entre el adquisidor a ser montado en el animal y la computadora del investigador o productor. Una opción con ZigBee [19] que se está desarrollando si deberá tener en cuenta aspectos tales como consumo de potencia y autonomía ya que serán sistemas que operaran sin supervisión por largos periodos de tiempo.

VI. REFERENCIAS

- [1] Implementation of a noise reduction algorithm in a hearing aid device based on dsPIC. A. Uriel, P. Agüero, J. Catíñeira, J.C. Tulli, E. González. IEEE ARGENCON 2012. Córdoba, Argentina. Junio de 2012
- [2] Virtual fences set to transform farming. http://www.csiro.au/Outcomes/Food-and-Agriculture/~/media/CSIROau/Divisions/CSIRO%20Entomology/VirtualFences_ent.pdf%20Standard.pdf



- [3] WAVE es un formato de audio digital normalmente sin compresión de datos desarrollado y propiedad de Microsoft e IBM. <http://tools.ietf.org/html/rfc2361>
- [4] <http://www.google.com/earth/index.html>
- [5] Computational method for segmentation and classification of ingestive sounds in sheep . D.H. Milone, H.L. Rufiner, J.R. Galli, E.A. Laca, C.A. Cangiano
- [6] Reconocimiento automático de sonidos ingestivos en rumiantes . D. Milonea, J. Gallib, C. Martíneza, H. Rufinera, E. Lacad, C. Cangiano. http://www.inta.gov.ar/actual/congreso/jaiio/doc/JIIA_GRO_2008_28.pdf
- [7] Automatic recognition of ingestive sounds of cattle based on hidden Markov models . Diego H. Milone1, María Soledad Padron, and Julio R. Galli, Carlos A. Cangiano, and Hugo L. Rufiner
- [8] http://infouniversidades.siu.edu.ar/noticia.php?titulo=descifran_los_sonidos_que_emiten_las_vacas_al_comer&id=135
- [9] Acoustic measurement of intake and grazing behaviour of cattle . E. A. Laca and M. F. WallisDeVries
- [10] Territorio y ganadería en la Patagonia Argentina: desertificación y rentabilidad en la Meseta Central de

- Santa Cruz, Andrade, Larry. Economía, Sociedad y Territorio, Vol. III, Núm. 012, julio-dici, 2002. El Colegio Mexiquense, A. C. México. "redalyc.uaemex.mx/pdf/111/11112309.pdf"
- [11] Automatic cattle control systems grazing without boundaries , Dave Swain, Greg Bishop-Hurley, and Jill Griffiths, Farming Ahead June 2009 No. 209 www.farmingahead.com.au
- [12] Sensor and Actuator Networks: Protecting Environmentally Sensitive Areas . Tim Wark, Chris Crossman, and Philip Valencia Dave Swain and Greg Bishop-Hurley Rebecca Handcock <http://doi.ieeecomputersociety.org/10.1109/MPRV.2009.15>
- [13] <http://ww1.microchip.com/downloads/en/DeviceDoc/41303G.pdf>
- [14] <http://www.lpi.tel.uva.es/~santi/slweb/muestreo.pdf>
- [15] <http://www.cartesia.org/data/apuntes/cartografia/cartografia-utm.pdf>
- [16] <http://www.tronico.fi/OH6NT/docs/NMEA0183.pdf>
- [17] <http://www.locosystech.com/product.php?zln=en&id=20>
- [18] http://foro.gabrielortiz.com/comparte/repositorio/Gabriel/Geograficas-UTM_Hoja_A1.pdf
- [19] <http://www.zigbee.org/>

Diseño e Implementación de un Sistema Embebido de Control Moderno: una Experiencia Práctica

Mariano Scasso*, Ricardo Garro†, Leo Ordinez‡ y Omar Alimenti§

*DIEC-UNS, †EEA Anguil "Ing. Agr. Guillermo Covas" – INTA, ‡§DIEC-UNS-CONICET

*mariano.scasso@uns.edu.ar – †rgarro@anguil.inta.gov.ar – ‡lordinez@uns.edu.ar – §iealimen@criba.edu.ar

*‡§ Bahía Blanca, Argentina †Anguil, Argentina

Resumen—Los cultivos bajo cubierta representan una forma de producción intensiva de alta eficiencia y gran calidad. El acoplamiento de tecnología electrónica a los invernaderos permite una mayor precisión en la recolección de datos y una mayor capacidad de respuesta. Además, se obtiene alta eficiencia en la gestión de la producción y se libera a las personas de tareas que pueden resultar peligrosas. El trabajo presentado aquí forma parte de uno mayor que consiste en el modelado, desde el punto de vista de los Sistemas Ciber-Físicos, de un Invernadero Robotizado. En particular, este artículo muestra la experiencia recogida en el modelado, diseño e implementación de un robot diferencial que realiza tareas dentro de un invernadero. En este sentido, se exponen los aspectos físicos considerados, el diseño del software y del hardware, sus interacciones y se presentan comparaciones entre los resultados simulados y los obtenidos de la implementación real. Asimismo, en cada parte analizada se hace referencia a teorías y técnicas prácticas involucradas, con el fin de que la experiencia adquirida pueda ser replicada en otras situaciones similares.

I. INTRODUCCIÓN

Los sistemas de producción bajo cubierta o en invernaderos constituyen, en la actualidad y en muchas de las regiones del mundo, un claro exponente de producción intensiva donde se alcanzan importantes niveles de rendimiento. En Argentina, la superficie destinada a la producción hortícola en invernaderos es aproximadamente de 5000 ha [6]. Sin embargo, los cultivos de invernadero traen aparejadas tareas tediosas y perjudiciales para la salud humana. A fin de abordar una solución a dichas problemáticas, pero manteniendo las ventajas de alto rendimiento y eficiencia en la producción, en [1], [2] se presentó el modelo y el diseño de un Invernadero Robotizado. La base teórica sobre la que se sustentó el enfoque propuesto en dichos artículos es la proporcionada por los Sistemas Ciber-Físicos (SCF). Este paradigma es relativamente nuevo [3], [4] y surge frente a la necesidad de modelar la interacción entre el mundo físico y el computacional de manera conjunta. Comúnmente esta clase de sistemas son caracterizados por la imposibilidad, por parte de un observador externo, de distinguir si el comportamiento del sistema se debe a las leyes físicas que lo rigen, a la intervención de un sistema informático embebido o a ambos. Otra manera de ver a los SCF, es como sistemas de sistemas profundamente embebidos, que son desplegados en un entorno físico real [3], [5]. Este último enfoque es el que se utilizó en la descripción de [1], [2], la cual establece al Invernadero Robotizado como un gran sistema, que incluye las condiciones climáticas a controlar, el grado de aplicación de producto

fitosanitario y diversos sistemas embebidos encargados de que las condiciones medioambientales deseadas se alcancen.

En particular, este artículo aborda uno de los sistemas embebidos que componen el Invernadero Robotizado: el Subsistema de Control de Movimiento, que se encarga de controlar la forma en que se desplaza una plataforma móvil dentro del invernadero. Este subsistema es crucial para el correcto funcionamiento de todo el SCF, ya que la mayoría de los sistemas embebidos que componen el Invernadero Robotizado lo utilizan o se montan sobre éste. En este sentido, este artículo tiene como principal objetivo exponer los aspectos experimentales que surgen de la implementación de un sistema embebido de control para una plataforma móvil, que se va a desplazar dentro de un invernadero. Asimismo, todas las cuestiones prácticas mostradas serán acompañadas con su base teórica correspondiente. A partir de esto, los autores pretenden que su experiencia pueda ser aprovechada por otros, al momento de modelar, diseñar e implementar un sistema embebido para el control de un artefacto móvil.

En la Figura 1 se puede ver captura de requerimientos realizada para el Invernadero Robotizado. En la captura se utilizó una versión simplificada del enfoque propuesto en [7], [8], haciendo hincapié en el rol del sistema relacionado con este artículo (Robot Diferencial) y dejando únicamente expresados otros dos roles por razones de completitud.

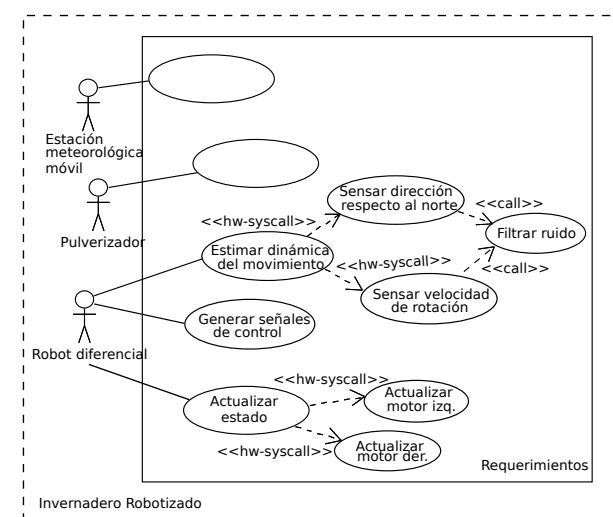


Figura 1: Requerimientos para el Invernadero Robotizado

Luego de esta introducción, el resto del artículo se organiza de la siguiente manera: la Sección II describe el modelo teórico de la plataforma móvil, tanto en su modelo cinemático como el diseño del controlador asociado y presentando una serie de simulaciones; en la Sección III, se exponen los aspectos de diseño e implementación de la plataforma móvil, distinguiendo aquellos referidos al sistema embebido general, al hardware involucrado y al software asociado. La Sección IV discute resultados experimentales obtenidos; y la Sección V, expone las conclusiones del trabajo.

II. PLATAFORMA MÓVIL: ROBOT DIFERENCIAL

Existen en la actualidad muchos tipos de robots móviles, y diferentes formas de clasificarlos: según su aplicación, según su forma de desplazamiento, según el tipo de rueda (si es que las tiene), etc. Para esta aplicación se utiliza un ROBOT DIFERENCIAL [11]. Este cuenta con dos ruedas que se controlan en forma independiente (cada una con un motor) y eventualmente una tercera rueda de rotación libre para dar estabilidad al vehículo.

Entre las ventajas de un robot diferencial se encuentran las siguientes:

- Diseño mecánico simple. Se eliminan los mecanismos relacionados con el control de dirección del robot.
- Permite un radio de giro pequeño. El robot puede girar 360° en torno a su centro de masa.
- Modelo matemático simple.

La principal desventaja es la imposibilidad de implementar un control suave para el desplazamiento en linea recta [12], [13]. Además se clasifica dentro de los robots no-holónomos, ya que no puede desplazarse en forma lateral.

II-A. Modelo cinemático

La Figura 2 muestra el esquema a partir del cual se plantea el modelo cinemático [17]. Como sistema de coordenadas se utilizan las coordenadas polares ya que permiten modelar el sistema con expresiones matemáticas lineales¹.

Sean v_d y v_i las velocidades tangenciales de la rueda izquierda y la derecha respectivamente, y ω_i y ω_d sus velocidades angulares (Figura 2). Entonces es posible definir a la velocidad lineal del vehículo como:

$$V = \dot{R} = \frac{v_d + v_i}{2} = \frac{\omega_d + \omega_i}{2} r, \quad (1)$$

donde r es el radio de la rueda y V es la velocidad lineal de avance. “R” es la distancia radial del vehículo respecto de la posición inicial.

La velocidad de rotación del vehículo será proporcional a la diferencia de las velocidad angulares de las ruedas, y se expresa como:

$$\dot{\theta} = \frac{(\omega_d - \omega_i) r}{l}, \quad (2)$$

donde l es la separación entre las ruedas.

¹Esto, en primera instancia, parece contradecir lo demostrado en [17]. Sin embargo, el sistema de coordenadas polares no está definido en el origen ($x=0$, $y=0$). No obstante, en la práctica, esta situación se resuelve considerando un entorno de dicho punto.

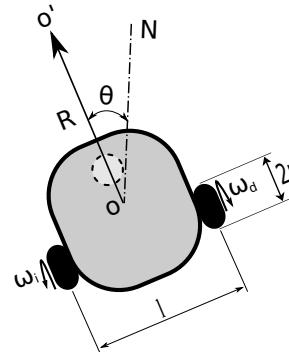


Figura 2: Esquema del robot diferencial.

Despreciando la inductancia interna del motor, se obtiene el sistema de ecuaciones diferenciales:

$$\begin{cases} \dot{\omega}_d = k_1 v_1 - k_2 \omega_d \\ \dot{\omega}_i = k_3 v_2 - k_4 \omega_i \\ \dot{R} = \frac{\omega_d + \omega_i}{2} r \\ \dot{\theta} = \frac{(\omega_d - \omega_i) r}{l} \end{cases} \quad (3)$$

En la Ecuación (3), k_1 y k_3 son constantes propias del motor que relacionan la aceleración con la tensión (v_1 y v_2) en bornes, así como k_2 y k_4 representan los coeficientes de fricción en las ruedas.

Luego seleccionando como variables de estado a w_d , w_i , y θ y a v_1 y v_2 como las entradas del sistema, se obtiene la siguiente representación matricial del espacio de estados.

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{b} \mathbf{u}, \quad (4)$$

donde:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \omega_d \\ \omega_i \\ \theta \end{bmatrix}, \mathbf{A} = \begin{bmatrix} -k_2 & 0 & 0 \\ 0 & -k_4 & 0 \\ \frac{r}{l} & -\frac{r}{l} & 0 \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} k_1 & 0 \\ 0 & k_3 \\ 0 & 0 \end{bmatrix}, \mathbf{u} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}.$$

Las constantes k_i se relacionan con las características electromecánicas de los motores, pero también se ven afectadas por cuestiones mecánicas tales como el sistema de transmisión o el coeficiente de rozamiento entre las ruedas y la superficie de desplazamiento. Por estos motivos se determinan mediante un procedimiento experimental.

II-B. Diseño del controlador

Una vez obtenido el modelo se diseña el controlador que permite llevar el sistema desde un estado inicial a otro deseado: $(\omega_i^0, \omega_d^0, \theta^0) \rightarrow (\omega_i^1, \omega_d^1, \theta^1)$. Para que cualquier estado sea alcanzable el sistema debe ser *controlable*, lo cual puede verificarse fácilmente siguiendo el método en [14].

Para conocer los estados actuales del sistema se deberían sensar cada uno de ellos. Sin embargo basta con medir la posición angular y una combinación lineal de las velocidades

angulares de las ruedas para que el sistema se vuelva *observable* [14]; es decir que es posible estimar los valores de ω_i y ω_d a partir de la medición de θ y $\dot{\theta}$.

Verificadas las propiedades del sistema se diseña la implementación de estados, es decir las ganancias del controlador. Para ello se selecciona un tiempo de establecimiento de un segundo utilizando los polos de Bessel [14], ya que no es necesaria una reacción demasiado rápida del sistema. Esta elección se sustenta en que la dinámica natural del sistema es lenta, ya que los polos a lazo abierto son $p_1 = 0$ y $p_2 = -0,6$.

Por ello seleccionar un tiempo de respuesta pequeño producirá señales de control de gran magnitud durante la operación del sistema. Esta situación es nociva desde el punto de vista de los requerimientos de potencia de los circuitos utilizados como interfaces a los motores. Además en la realidad las señales de control se saturan debido a que la potencia disponible es finita y a las limitaciones eléctricas de los motores utilizados.

En consecuencia, si bien es posible seleccionar cualquier tiempo de establecimiento desde el punto de vista de control, es la aplicación práctica la que produce las limitaciones y deben tenerse en cuenta durante la etapa de diseño.

A su vez las ganancias del observador deben asegurar un tiempo de convergencia al menos tres veces menor al tiempo de establecimiento deseado para el sistema, de manera que la estimación de los valores no afecte la acción de control diseñada [14].

Vale aclarar que los diseños anteriores se realizan en tiempo continuo, es decir con ecuaciones diferenciales, que luego se discretizan con un tiempo de muestreo adecuado obteniéndose el modelo en ecuaciones a diferencias que se implementa en un microcontrolador. Además se produce una discretización en módulo debido a la operación de los conversores A/D que tienen una definición finita. Estos aspectos introducen errores que generan diferencias en el comportamiento real del sistema con respecto a las simulaciones realizadas. Sin embargo teniendo en cuenta estos aspectos durante el diseño es posible obtener resultados aproximados [14], [15].

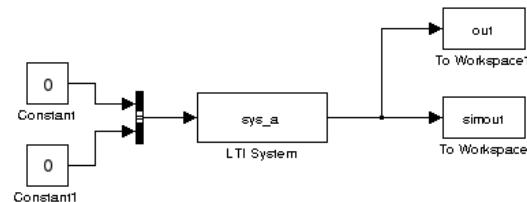
II-C. Simulaciones

Se define el estado de reposo como $(\omega_i, \omega_d, \theta) = (0, 0, 0)$ es decir que ambas ruedas están detenidas y el vehículo se encuentra en la posición inicial. A continuación se muestran algunas simulaciones realizadas sobre el modelo. La Figura 3 muestra la representación en Simulink del sistema y su respuesta ante las condiciones iniciales $(\omega_i, \omega_d, \theta) = (10, -10, 0)$, en esta situación la plataforma se encontraría girando rápidamente en torno a su centro de masa. Se observa que el sistema es estable, es decir que la respuesta es acotada y el vehículo tiende a detenerse una vez retirada la excitación. Sin embargo nunca retorna al estado $(0, 0, 0)$, es decir que la posición angular final converge a un valor cualquiera en lugar de retornar a la posición previa a la perturbación.

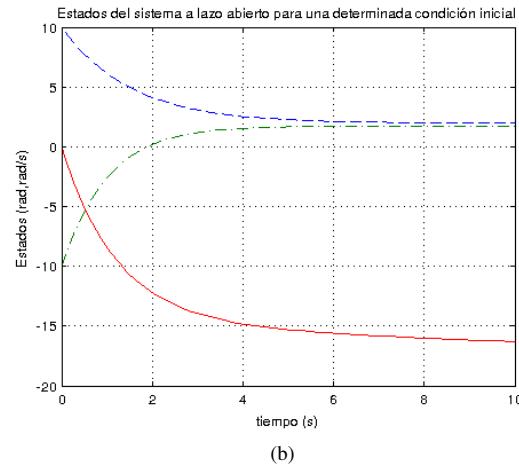
En la Figura 4 se muestra el sistema luego del diseño del controlador sometido a la misma condición inicial. En este caso se aprecia que el controlador modifica progresivamente las velocidades angulares hasta converger al estado $(0, 0, 0)$.

Es interesante observar la Figura 4-b, en la cual se muestran en forma superpuesta los estados del sistema y los estimados.

Durante la simulación se inicia el sistema en el estado correspondiente a la condición inicial, pero por defecto el observador inicia la aproximación en el estado $(0, 0, 0)$. Por ello se observa que las curvas se “despegan” hasta los 0.3 segundos y comienzan a converger a partir de allí, lo cual se condice con el tiempo de establecimiento fijado para el observador. Producto de esta respuesta transitoria las señales ω_i y ω_d presentan un gran sobrepico. Esta situación es improbable en una situación real ya que, asumiendo que el controlador se activa cuando el móvil está detenido, las condiciones iniciales del sistema y del controlador son similares. Aún así fue seleccionada para exemplificar estos aspectos y comprobar el tiempo de establecimiento del observador.



(a) Modelo a lazo abierto en Simulink



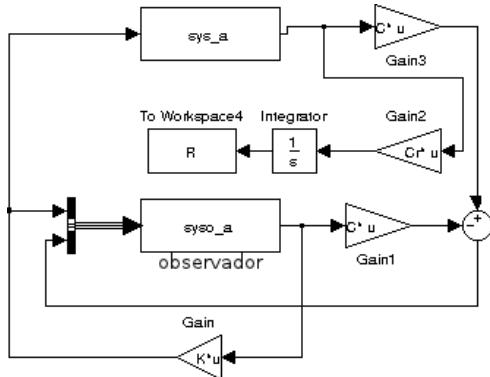
(b)

Figura 3: (b) Respuesta del sistema a lazo abierto. Con líneas a trazos se indican ω_i y ω_d , con línea llena θ .

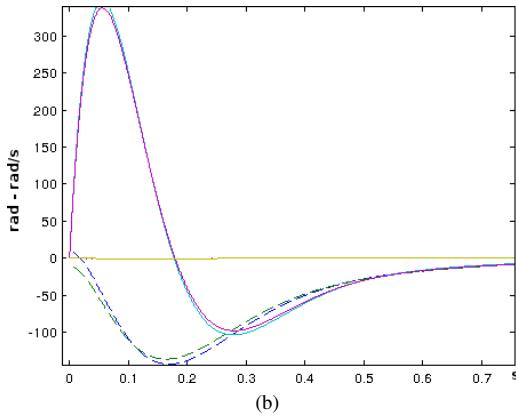
La Figura 5 muestra al sistema siguiendo una trayectoria determinada (5-a en linea punteada). Se observa que las respuestas del sistema son suaves y verifican los tiempos de establecimientos calculados. La Figura 5-b representa el desplazamiento que realizaría el robot en el plano al seguir la señal de referencia de la Figura 5-a.

III. IMPLEMENTACIÓN

La implementación del sistema embebido de control, modelado en la sección anterior, se realiza distinguiendo el diseño del hardware, la unidad de procesamiento y el software asociado. En este sentido plantear el modelo en forma de variables de estado en el domino digital es una ventaja, pues las ecuaciones a diferencias resultantes se resuelven simplemente como sumas y productos de términos. Este procedimiento se explicará en la subsección de software embebido. Por otra



(a) Modelo a lazo cerrado en SimuLink



(b)

Figura 4: (b) Respuesta del sistema a lazo cerrado. Las líneas a trazos representan los estados del sistema, las líneas llenas los estados del observador.

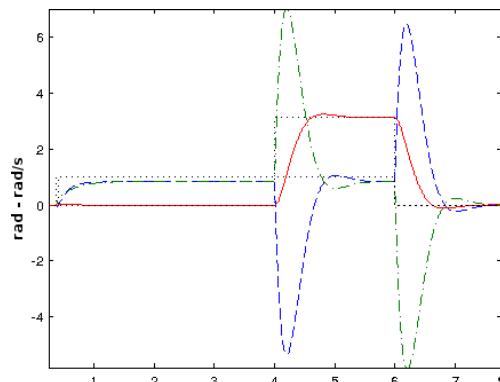
parte se requiere realimentar al sistema con información desde el exterior, y adaptar las señales de control para que actúen sobre los motores. El hardware necesario para este propósito se detalla en su subsección correspondiente. Asimismo, se describen las características de la unidad de procesamiento del sistema, de acuerdo a los requerimientos capturados tanto de hardware como de software.

III-A. Hardware

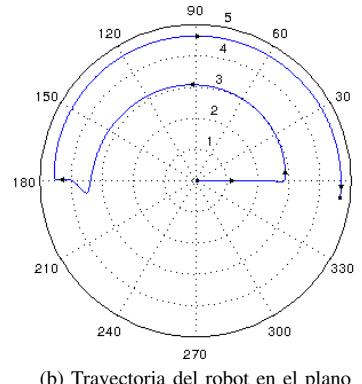
En el hardware del sistema embebido se destacan tres componentes:

1. Una unidad de procesamiento, implementada mediante un microcontrolador.
2. Un sistema de medición, compuesto por sensores y sus interfaces A/D.
3. Un sistema de actuación, conformado principalmente por los motores, sus interfaces D/A y la etapa de potencia.

A1. Unidad de procesamiento: Para procesar la información necesaria para el control se utiliza el microcontrolador PIC18F4550 [18]. La selección de esta arquitectura se sustenta en que cuenta con una gran cantidad de funciones, es de bajo costo y fácil acceso en el mercado local. Cuenta además con periféricos específicos para la modulación PWM y para la



(a) Estados del sistema siguiendo una trayectoria definida



(b) Trayectoria del robot en el plano

Figura 5: Respuesta del sistema al seguir una trayectoria de referencia. (a) En líneas a trazos ω_i y ω_d , línea llena θ . En línea punteada trayectoria a seguir.

comunicación I2C los cuales se requieren por las otras etapas del sistema.

A2. Sistema de medición: Para poder estimar los estados del sistema se requiere tomar información del exterior. Existen diferentes variables que pueden ser medidas, por ejemplo las velocidades angulares de las ruedas utilizando encoders. Sin embargo dado que el medio ambiente en el cual debe operar el dispositivo es adverso, se optó realizar el control utilizando sensores de estado sólido. Aunque usar estos sensores dificulta la obtención de los datos, evita totalmente el desgaste mecánico que sufren los encoders por tener piezas móviles, y las estructuras mecánicas para colocarlos. Además son por lo general de un costo menor, lo cual es importante pensado en un desarrollo con fines aplicados.

La velocidad angular se mide utilizando un giróscopo analógico [19] que es muestreado periódicamente utilizando un conversor A/D del microcontrolador. Sin embargo esta medición no es suficiente para cumplir con la condición de observabilidad del sistema, por lo tanto se adiciona una brújula digital que, entre otros parámetros, transmite al microcontrolador la posición angular mediante el bus I2C [20].

A3. Sistema de actuación: Para variar la tensión sobre los motores y a partir de ella la velocidad, se utiliza una modulación PWM bipolar, que se implementa en una configuración comúnmente denominada “puente H” [16]. El microcontrolador empleado cuenta con un periférico específico

para generar las señales de este tipo de modulación.

Para asegurar la aislación entre las señales de control y de potencia el circuito se encuentra optoacoplado. En la Figura 6 se muestra el circuito esquemático. Los requerimientos de potencia del sistema son importantes, ya que los motores utilizados consumen 10A durante el arranque y se estabilizan en torno a los 7A una vez en régimen. Por ello se utilizan transistores MOSFET de potencia controlados por sus correspondientes drivers que aseguran la conmutación correcta a la frecuencia de trabajo, 1.5 KHz.

Una virtud de este circuito es que permite que la potencia recircule durante el frenado desde los motores hacia la batería, almacenando parte de esa energía.

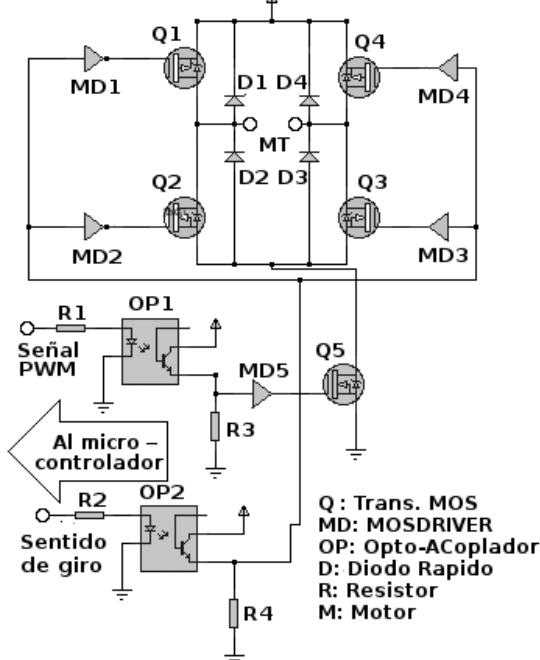


Figura 6: Circuito del sistema de actuación

III-B. Software Embebido

El software embebido en el microcontrolador realiza las tareas que hacen al control de la plataforma (Figura 1, Sección I). La descripción del software de control, se muestra en un Diagrama de Estado de UML [9] en la Figura 7.

Debido a que la salida analógica del giróscopo es ruidosa se aplica un filtro pasabajos FIR [15] a las muestras tomadas con el conversor A/D. La frecuencia de corte de dicho filtro se fija en 10 Hz, ya que la dinámica del sistema es muy lenta, utilizando como frecuencia de muestreo 100 Hz, lo cual asegura ampliamente la frecuencia de Nyquist y no genera una gran demanda al microcontrolador. A su vez para obtener una referencia de la posición del vehículo se utiliza una brújula electrónica que informa la desviación con respecto al norte magnético con una exactitud de 2°; la información se transfiere sobre el bus I2C.

El modelo del observador, que permite calcular las señales de control requeridas a partir de la información adquirida,

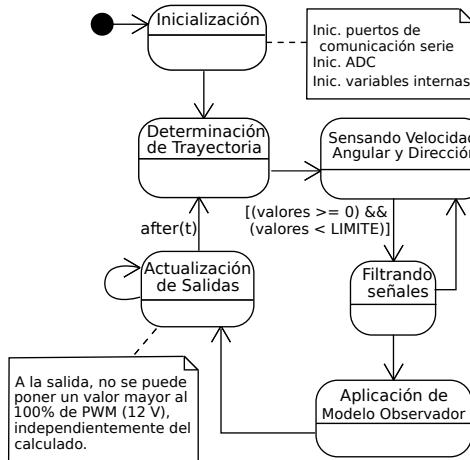


Figura 7: Diagrama de Estado

se reduce a una serie de ecuaciones a diferencias que no son más que sumas y productos en forma matricial. Una vez calculadas las dos señales de control (una para cada motor) se transforman en dos señales PWM utilizando el módulo CCP del microcontrolador. Luego, a través del circuito de potencia, se excitan los motores. La frecuencia de operación del PWM se fija en 1.5 KHz que es la frecuencia mínima permitida dada la configuración del clock, de manera de disminuir las pérdidas de conmutación en la etapa de potencia [15], ya que para esta aplicación no se requiere switching a altas frecuencias.

Finalmente, se puede observar que, desde el punto de vista de un sistema de tiempo real [10], no se diseña el software como tareas concurrentes que compiten por el procesador. Sino que se optó por construir un único proceso secuencial que realiza las acciones de sensado, procesamiento de la información y actuación. Esta decisión se basó en el estudio de la dinámica del sistema, en pruebas realizadas sobre la respuesta temporal de los dispositivos electromecánicos (motores, etapa de potencia, etc.) y en cálculos estimados del tiempo de ejecución de las actividades de software más demandantes de procesador. No obstante, en caso de disponer de otros dispositivos de hardware o requerir mayor complejidad en los cálculos, sería necesario un sistema de software en forma de tareas utilizando un Sistema Operativo de Tiempo Real.

IV. RESULTADOS EXPERIMENTALES

La implementación del robot diferencial permitió comparar y ajustar diferentes parámetros y variables del modelo matemático simulado. Un pequeño video ilustrativo de una de las pruebas realizadas puede verse en [21]. Este experimento consistió en desplazar el vehículo en línea recta por un terreno desnivelado. De ésta y otras experiencias realizadas, se desprenden las siguientes observaciones:

- El circuito de potencia que alimenta a los motores, al ser exigido al límite de sus capacidades, debe ser cuidadosamente dimensionado. Si bien sobredimensionarlo es una previsión aceptable, ésto incrementa los costos de los componentes utilizados.



- El ruido que afecta a los sensores es una aspecto sensible para conseguir un control preciso. Por tal motivo los mismos deben estar adecuadamente alimentados y protegidos del ruido electromagnético.
- Las condiciones del medio, en el cual se desempeña el robot, influyen drásticamente en el comportamiento del mismo. Esto se debe a que las diferentes perturbaciones, tales como variaciones en el coeficiente de rozamiento o inclinaciones del terreno, no son incluidas en el modelo. Para disminuir el efecto de estos factores deben seleccionarse apropiadamente dispositivos mecánicos adicionales (amortiguadores, ruedas, etc). De otro modo, un modelo matemático mas avanzado complicaría su implementación, requiriéndose un poder de cómputo mayor al de un sistema embebido monoprocesador.

V. CONCLUSIONES

La agricultura de precisión es una disciplina que busca lograr un incremento de la producción agropecuaria, mediante la inclusión de tecnología que haga más eficiente y seguro el manejo de los cultivos. En particular, los cultivos intensivos bajo cubierta o en invernaderos representan un campo de aplicación propicio para estas tecnologías. En este sentido, la construcción de un Invernadero Robotizado se plantea como una necesidad, debido a cuestiones relacionadas con la salud humana y la eficiencia de las tareas a desarrollar en él.

En este artículo se presentó la implementación de uno de los subsistemas que conforman el Invernadero Robotizado, el cual se compone de un robot móvil que recorre los pasillos del invernáculo. Así, partiendo del modelo cinemático simplificado de un robot diferencial y aplicando técnicas de control lineal, se controló su posición angular. Esto permitió que un vehículo de estas características se desplace en linea recta o pueda realizar giros de una magnitud dada. Es de remarcar, que a lo largo del trabajo, se mantuvo un enfoque práctico, pensando en la implementación de la plataforma móvil en el marco de un proyecto agro-tecnológico más amplio. En este sentido no es menor el hecho de conseguir un producto simple desde el punto de vista mecánico, y económico y robusto desde el punto de vista de la electrónica asociada a su control.

El desarrollo del sistema embebido que controla el robot se planteó en diferentes fases. En primer lugar, realizó una Ingeniería de Requerimientos del sistema. Luego, se modeló matemáticamente el aspecto cinemático del robot. En base a ésto, se diseñó un controlador acorde a dichas características. Este modelo matemático fue simulado y verificado con herramientas de software. Con lo anterior ya analizado, se pasó a la implementación. En primer lugar, se diseño el hardware necesario, el cual incluía una unidad de procesamiento y distintas interfaces para tomar información del estado del móvil y actuar en consecuencia. Una vez determinado el hardware, se continuó con el diseño del software que implementaría el controlador modelado anteriormente. Cabe remarcar que las etapas de diseño e implementación del hardware y del software no se realizaron en forma secuencial sino que fueron hechas de modo incremental e iterativo. Por último, se contrastaron experiencias recogidas de la implementación práctica con

aspectos teóricos y resultados de simulaciones, las cuales serán utilizadas para abordar futuros trabajos relacionados con otros subsistemas del Invernadero Robotizado. En particular, los resultados mostrados en este artículo servirán de base para la construcción de un sistema automático de pulverización y dosificación de producto fitosanitario.

AGRADECIMIENTOS

Los autores quieren expresar su agradecimiento a los doctores Andrés Gracia y Guillermo Calandrini por los aportes realizados a este trabajo.

REFERENCIAS

- [1] Ricardo Garro, Leo Ordinez y Omar Alimenti. (2011). "Design Patterns for Cyber-Physical Systems: The Case of a Robotic Greenhouse", Brazilian Symposium on Computing System Engineering, Workshop on Embedded Systems, Florianopolis.
- [2] Ricardo J. Garro, Leo Ordinez y Omar Alimenti. (Septiembre, 2011). "Invernadero Robotizado - Perspectiva de Diseño desde los Sistemas Ciber-Físicos", II Congreso de Microelectrónica Aplicada (uEA 2011), La Plata, Argentina.
- [3] Lee, E. A. and S. A Seshia. (2010). "Introduction to Embedded Systems-A Cyber-Physical Systems Approach." Lee & Seshia.
- [4] Krogh, B. H., E. Lee, I. Lee, A. Mok, R. Rajkumar, L. R. Sha, A. S. Vincentelli, et al. (2008). "Cyber-Physical Systems, Executive Summary." CPS Steering Group, Washington DC.
- [5] Lee, E. A. (2006). "Cyber-physical systems-are computing foundations adequate." In Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, 1:1-9.
- [6] (Marzo, 2012) WebSite Instituto Nacional de Tecnología Agropecuaria. [Online]. Disponible en: <http://www.inta.gov.ar>
- [7] Leo Ordinez, Omar Alimenti y Lucas Calles. (Octubre, 2011). "Eliciting Requirements in Small Cyber-Physical Systems", XXXVII Conferencia Latinoamericana de Informática (XXXVII CLEI), Quito, Ecuador.
- [8] Leo Ordinez, David Donari, Rodrigo Santos, Javier Orozco.(Abril, 2010). "From User Requirements to Tasks Descriptions in Real-Time Systems", 13th Workshop on Requirements Engineering WER'10, Cuenca, Ecuador.
- [9] (Marzo, 2012) WebSite Unified Modeling Language. [Online]. Disponible en: <http://www.uml.org>
- [10] Pont, M. J. (2001). "Patterns for time-triggered embedded systems: Building reliable applications with the 8051 family of microcontrollers." ACM Addison-Wesley Publishing Co.
- [11] R. Silva Ortigoza, J. R. García Sánchez, V. R. Barrientos Sotelo, M. A. Molina Vilchis, V. M. Hernández Guzmán, G. Silva Ortigoza. (2007). "Una panorámica de los robots móviles", Telemática, Universidad Rafael Belloso Chacín Zulia, Venezuela pp. 1-14.
- [12] Brockett R.W. (1983). "Asymptotic stability and feedback stabilization". Differential geometric control theory. R.W. Brockett , R.S. Millman y H.J. Sussman, Eds., Birkhauser, Boston, USA.
- [13] Canudas de Wit., B. Siciliano y G. Bastin. (1997). "Theory of Robot Control". Springer.
- [14] Vaccaro, R.J. (1995). "Digital Control – A State-Space Approach", McGraw-Hill Inc.
- [15] Oppenheim, A. Schafer, R. (1998)."Discrete-Time Signal Processing", 2da. Edición, Prentice Hall.
- [16] Ned Mohan, "Power Electronics - Converters, Applications and Desing", 2da. Edición, John Wiley & Sons INC., 1995
- [17] Yongoug Chung; Chongkug Park; Harashima, F. (Aug, 2001). "A position control differential drive wheeled mobile robot," Industrial Electronics, IEEE Transactions.
- [18] Microchip, PIC18F4550 datasheet.
- [19] (Marzo, 2012) WebSite Pololu, giroscopio. [Online]. Disponible en: <http://www.pololu.com/catalog/product/1270>
- [20] (Marzo, 2012) WebSite Pololu, Compás magnético. [Online]. Disponible en: <http://www.pololu.com/catalog/product/1265>
- [21] (Marzo, 2012) Video del Prototipo. [Online]. Disponible en: <http://www.youtube.com/watch?v=N7c1RA-Sy9k>

Procesamiento de señales de speckle dinámico

Implementación del algoritmo de Fujii en FPGA

Noelia I. Echeverría^{*}, Maximiliano Antonelli[†], Adriana G. Scandurra, Lucía I. Passoni, y Claudio M. González

Departamento de Ingeniería Electrónica

Facultad de Ingeniería. UNMdP

Mar del Pata, Argentina

* noelia.echeverria@fi.mdp.edu.ar, †maxanto@fi.mdp.edu.ar

Resumen – Este trabajo presenta el desarrollo e implementación en dispositivos lógicos programables de un sistema de almacenamiento, procesamiento y presentación de resultados de señales de speckle dinámico.

La realización práctica se resolvió con un kit de desarrollo DE2 de Altera, ya que cuenta con todas las interfaces y periféricos necesarios para esta implementación.

El resultado final es un sistema capaz de leer los datos almacenados previamente en una memoria RAM, aplicar el algoritmo de Fujii y mostrar una imagen procesada en un monitor.

Palabras clave – implementación en FPGA; speckle láser; algoritmo de Fujii; sistema embebido; tiempo real.

I. INTRODUCCIÓN

Debido a que el fenómeno de speckle dinámico ha resultado ser una técnica muy útil para obtener información del nivel de actividad de una muestra, es cada vez más necesario contar con herramientas de adquisición, medición y presentación de resultados. Para brindar esta solución se eligió un dispositivo FPGA, por su flexibilidad, velocidad y capacidad de paralelización de procesos.

Actualmente, el procesamiento de señales de speckle dinámico se realiza a través de una computadora, ya sea en modo off-line procesando un archivo que contiene los datos almacenados por un adquisidor u on-line conectada directamente al sistema de adquisición mediante una placa adquisidora. A pesar de la cantidad de usos prácticos posibles para esta tecnología, como evaluar la movilidad de bacterias [1], el secado de pinturas [2], la calidad de semillas para cultivo [3], el estado de frutas [4], etc., no se cuenta con dispositivos dedicados para solucionar aplicaciones prácticas. Pueden verse en los últimos años recientes esfuerzos para implementar un sistema embebido autónomo [5], pero el foco de estos trabajos es el de resolver un algoritmo determinado en hardware, sin tener en cuenta interfaces, adquisición y presentación de datos. El foco de este trabajo consiste en conseguir un sistema embebido autónomo que incluya un algoritmo fácilmente reemplazable.

A. Speckle Dinámico

Cuando un haz de luz coherente ilumina una superficie difusora rugosa (con una profundidad de rugosidad del orden de la longitud de onda), en la imagen adquirida se puede observar un diagrama de claros y oscuros. Cada punto luminoso es llamado grano de *speckle* y al diagrama se lo denomina patrón o diagrama de *speckle* [6] [7].

Cada punto perteneciente a una superficie difusora se comporta como fuente de luz cuando es iluminado por una fuente externa, cada uno de estos puntos (diferenciales) se denomina “dispersor”.

Si esta superficie además de difusora es rugosa cada dispersor tiene una diferencia de camino hasta llegar al observador, que puede ser el ojo humano, cámara CCD o cualquier otro tipo de detector.

Cuando la fuente de luz externa es coherente, es decir con fase constante a lo largo del frente de onda, se produce un desfasaje entre los rayos que llegan al observador debido únicamente a la diferencia de camino hasta los distintos dispersores. Esto produce un patrón de interferencia con máximos y mínimos separados en zonas (fig. 1), cada una de estas zonas se llama *speck* y al conjunto de estas zonas patrón de *speckle* o simplemente *speckle*.

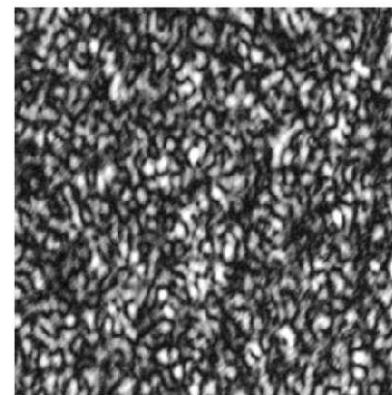


Figura 1. Patrón de speckle

Este patrón de *speckle* no guarda una relación unívoca con la superficie iluminada, es decir, no permite inferir propiedades de esta superficie excepto que es difusora (al menos levemente) y rugosa. En cambio, si existe movimiento de los dispersores, o alteraciones temporales del medio de transmisión o de la fuente de luz, el patrón observado varía en el tiempo y se lo denomina *speckle* dinámico. Si mediante un banco de medición adecuado se puede asegurar que el medio de transmisión y la fuente de luz son invariantes, se puede atribuir toda la dinámica del *speckle* a cambios de la superficie y así inferir propiedades de ella.

Un banco de prueba conveniente para registrar la evolución temporal de un patrón de *speckle* dinámico es el propuesto por Oulamara [8] (fig. 2). Con esta disposición se adquiere y almacena una serie consecutiva de imágenes de patrones de *speckle* (fig. 3).

Existen diferentes algoritmos para el análisis de patrones de *speckle* dinámico [9]. En el presente trabajo se implementó el algoritmo de Fujii [10] (ec. 1), dado que trabaja en el dominio espacio-temporal, lo cual permite que el análisis se realice en tiempo real.

$$D_{x,y} = \sum_{n=1}^{N-1} \frac{|I_{x,y}(n) - I_{x,y}(n+1)|}{I_{x,y}(n) + I_{x,y}(n+1)} \quad (1)$$

Donde $D_{x,y}$ es el valor del descriptor de Fujii en el píxel de coordenadas x,y correspondiente a la secuencia de imágenes adquiridas, $I_{x,y}(n)$ es la intensidad del píxel de la imagen n . N es la cantidad de imágenes del arreglo.

B. FPGA

Una FPGA (Field Programmable Gate Array), es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión puede ser configurada ‘in situ’ mediante un lenguaje de descripción de hardware (HDL) como VHDL [11] o Verilog [12]. Estos bloques lógicos denominados LAB (logic array blocks) en nomenclatura de Altera o CLB (Configurable Logic Block) en nomenclatura de Xilinx pueden ser interconectados después del proceso de manufactura por el usuario/diseñador mediante líneas de interconexión programables (fig. 4), lo que permite que la FPGA pueda desempeñar cualquier función lógica necesaria.

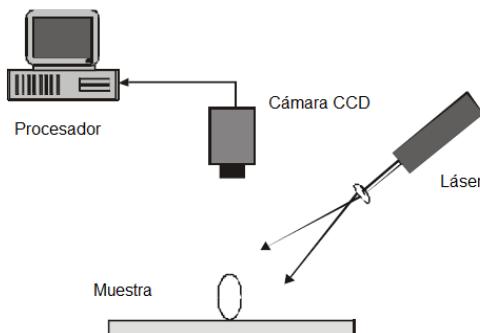


Figura 2. Banco de prueba para la adquisición de speckle dinámico

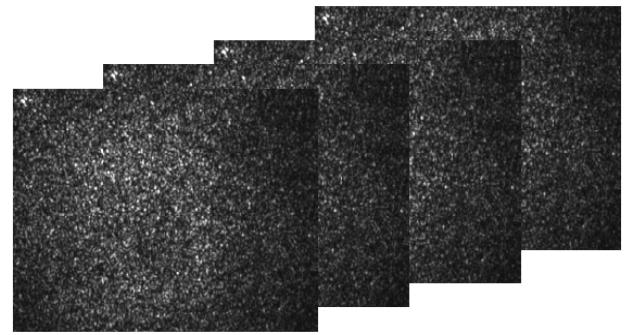


Figura 3. Arreglo de imágenes de speckle

El kit de desarrollo utilizado cuenta con una FPGA EP2C35F672C6 como componente principal, correspondiente a la familia Cyclone II de Altera. Este dispositivo tiene una arquitectura de 16 LE's (Logic Elements) por cada LAB [13] (fig. 5).

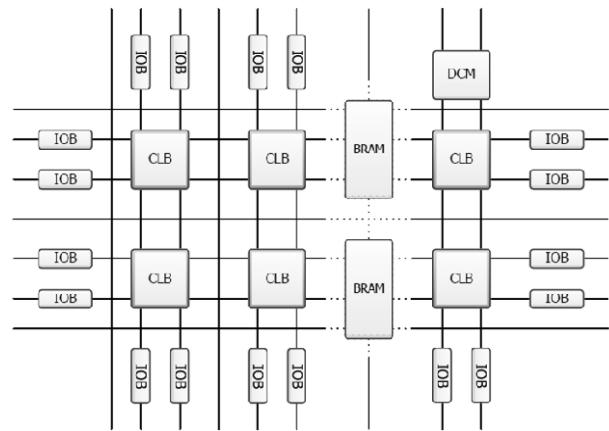


Figura 4. Bloques y líneas de una FPGA de Xilinx

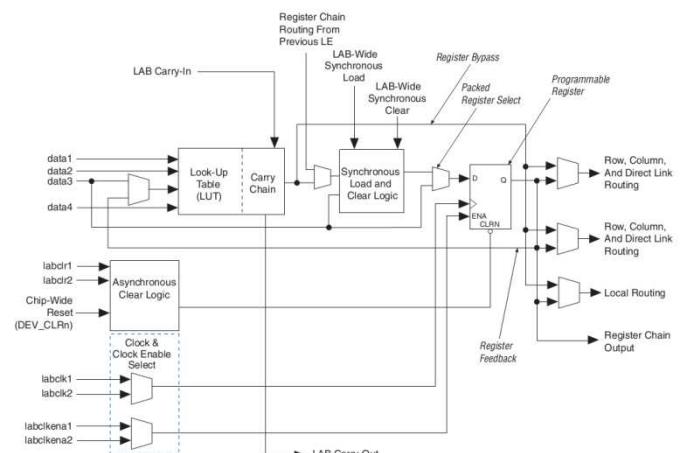


Figura 5. Diagrama de un LE del dispositivo utilizado.

visualización, en este caso, en un monitor VGA. Este sistema consta de 3 etapas (fig. 6):

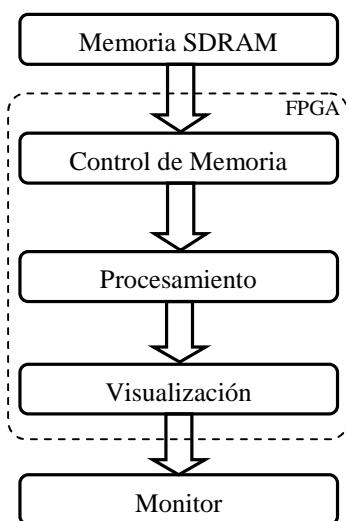


Figura 6. Diagrama del sistema

II. IMPLEMENTACIÓN

Se implementó un sistema de procesamiento y visualización de resultados de señales de speckle dinámico en un kit de desarrollo DE2 [14] de Altera [15]. Este kit fue elegido por contar con memoria SDRAM y conversor VGA/DAC, utilizados para esta aplicación.

Todos los bloques (excepto SDRAM_CONTROLLER) fueron generados en VHDL y son de producción propia. El circuito SDRAM_CONTROLLER fue encontrado en Opencores [16], está escrito en Verilog y es de uso libre. Los bloques individuales fueron generados mediante código y luego interconectados en un archivo gráfico. Todo el desarrollo y simulaciones fueron realizados mediante el paquete Quartus II® v8.0 de Altera.

Los datos de los patrones de speckle fueron obtenidos utilizando un banco de medición similar al de la figura 2 y almacenados en forma off-line en la SDRAM del kit de desarrollo. Para cargar estos datos en la memoria se utilizó el paquete DE2 Control Panel de Altera, que permite, entre otras cosas, leer y grabar datos en la memoria SDRAM. Para ingresar los datos de las 100 imágenes utilizadas, se generó un archivo en formato .hex (utilizando Matlab® 7.9.0) de $N \times 100$ filas de 16 bits (4 caracteres de 4 bytes/caracter en formato hexadecimal por fila), siendo N el numero de píxeles por imagen dividido 2 (por cada posición de memoria se almacenan 2 píxeles) y 100 el número de imágenes utilizadas. En este caso, $N= 8192$ (imágenes de 128*128 píxeles).

El sistema desarrollado lee los datos almacenados en la memoria, los procesa mediante un algoritmo específico (algoritmo de Fujii), y adapta los resultados para su posterior

- Etapa de memoria. Genera las señales necesarias para el inicio, configuración y control de la memoria SDRAM, lee los datos almacenados en ella y los presenta a la etapa siguiente en un formato adecuado.
- Etapa de procesamiento. Toma los datos provenientes de la etapa de memoria, los procesa y los almacena en una memoria RAM embebida con una longitud de palabra de 16 bits, luego, se adaptan los valores almacenados en esta memoria a un formato de 8 bits y se entregan a la etapa de visualización.
- Etapa de visualización. Genera las señales de sincronismo del monitor, se toman los datos provenientes de la etapa de procesamiento y se presentan en pantalla.

A. Control de memoria

El Kit utilizado en este desarrollo cuenta con 8 MB de memoria SDRAM [14] con una longitud de palabra de 16 bits, que es utilizada para almacenar el vector de datos de entrada al algoritmo.

Para poder leer los datos almacenados en esta memoria y entregarlos al algoritmo en un formato adecuado, fue necesario desarrollar un circuito de cuatro bloques lógicos (fig. 7).

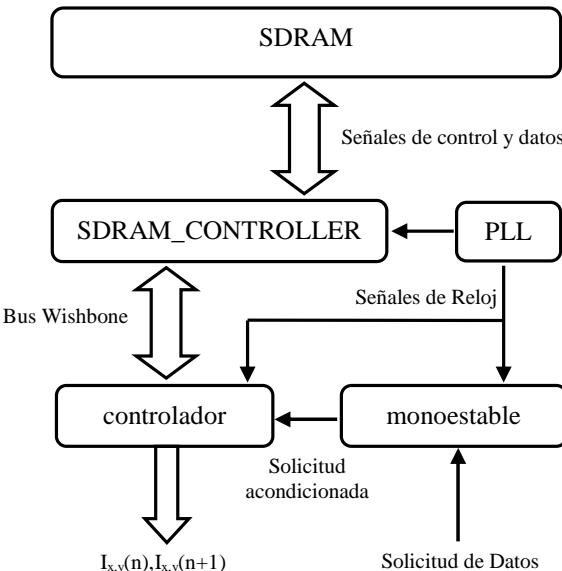


Figura 7. Circuito de lectura de memoria.

El primero de ellos (SDRAM_CONTROLLER) es el encargado de generar las señales de sincronismo y de entregar los datos y direcciones de memoria en tiempo y forma, este circuito se encuentra conectado físicamente al chip de memoria y presenta al circuito interno una interfaz *wishbone* estándar [17]. El segundo circuito (controlador) se encarga de liberar al circuito de procesamiento del manejo del bus *wishbone*, tomando solicitudes de lectura y presentando los datos de a pares al algoritmo. Un tercer bloque (monoestable) se encarga de acondicionar los tiempos de solicitud de datos y el último (PLL) es un generador de señales de reloj.

B. Procesamiento

En esta etapa, formada por 3 bloques (Fig. 8), se procesan los datos provenientes de la memoria SDRAM mediante el algoritmo de Fujii. En el primer bloque (ALGORITMO) se separan los datos provenientes de la etapa de memoria, en 4 datos que representan las intensidades de 4 píxeles de dos imágenes consecutivas. Se realizan las operaciones matemáticas que se encuentran dentro del núcleo de la sumatoria del algoritmo (Ec. 2) y se pasan los datos procesados al bloque siguiente (ACUMULADOR). En este bloque, los datos procesados se suman a los previamente almacenados en una memoria embebida de 16 bits de longitud de palabra (la primera vez, los datos almacenados son todos cero) y se vuelven a guardar en la misma posición (implementación de la sumatoria). Cuando se terminan de procesar estos píxeles, se cargan nuevamente dos datos de la SDRAM y se repite el proceso. Este procedimiento se realiza hasta que el algoritmo se aplica a todas las intensidades de los píxeles de la imagen n (junto con la imagen n+1). Cuando se procesa una imagen completa los datos almacenados en la memoria RAM de 16 bits son enviados a la etapa de visualización.

$$\frac{|I_{x,y}(n) - I_{x,y}(n+1)|}{I_{x,y}(n) + I_{x,y}(n+1)} \quad (2)$$

C. Presentación

Para la visualización de los resultados del procesamiento de las señales de speckle dinámico se utilizó un monitor VGA de tubo de rayos catódicos (TRC) por lo que se desarrolló un bloque (SYNC_VGA) para la generación de todas las señales de sincronismo necesarias para su correcto funcionamiento (fig. 9). Las salidas de este circuito son las entradas al conversor VGA/DAC ADV7123 integrado al kit de desarrollo DE2.

Para brindar flexibilidad y paralelismo al sistema, se utilizó un bloque de memoria embebida de 8 bits de longitud de palabra (MEMO_8) para almacenar los datos a ser visualizados.

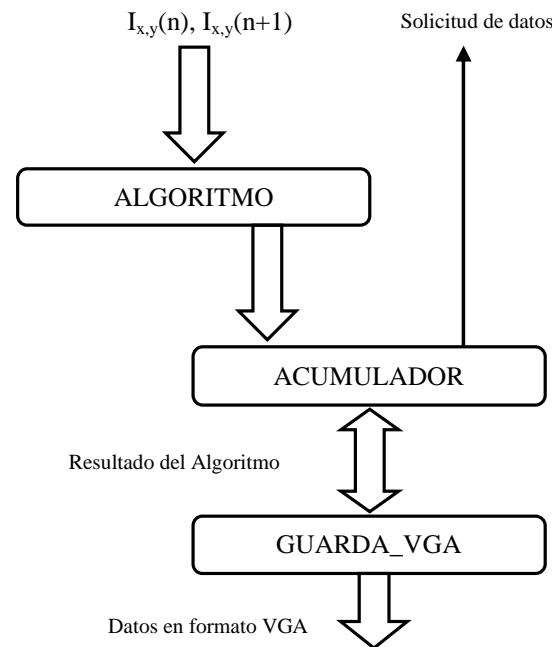


Figura 8. Etapa de procesamiento.

Se generó un bloque buffer (FRAME_BUFFER) para prealmacenar los datos procesados y así evitar problemas de sincronismo entre la lectura de la memoria embebida y la visualización en el monitor. Además se generó un bloque (FORMATO) para controlar la posición de la imagen, el color del fondo, el marco de la imagen, etc

III. CONCLUSIONES

Este trabajo es la base para un futuro sistema embebido de adquisición, análisis y presentación de resultados de speckle dinámico. Se podrá implementar a partir de este sistema, un dispositivo autónomo de procesamiento de señales de speckle dinámico, reemplazando la etapa de memoria SDRAM por una etapa de adquisición (cámara CCD conectada a un banco de medición).

El sistema generado es flexible, modular y fácilmente paralelizable, por lo cual pueden programarse otros algoritmos de procesamiento de imágenes (de speckle u otro tipo de imágenes), presentar los resultados en otro dispositivo de visualización o tomar como fuente de datos algún otro dispositivo como cámaras CCD, memorias flash, dispositivos USB, etc.

Se eligió implementar el algoritmo de Fujii dado que trabaja en el dominio espacio-temporal, y permite el procesamiento en tiempo real de las señales (también se pueden aplicar algoritmos que trabajen en el dominio de la frecuencia, pero el análisis es off-line). Para implementar el algoritmo de Fujii es necesario un acumulador dimensionado por dos factores, cantidad de muestras procesadas y cantidad

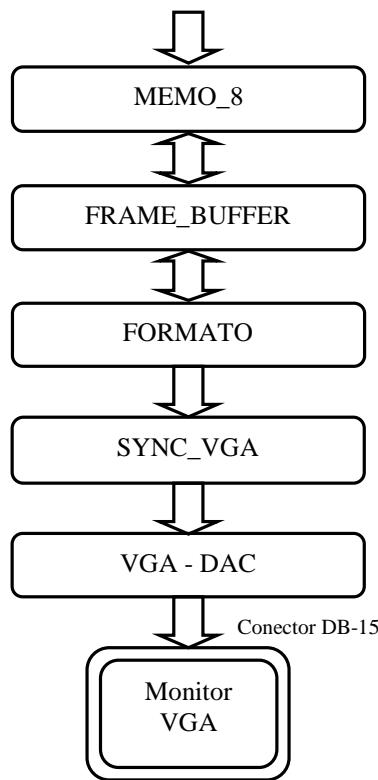


Figura 9. Etapa de visualización

de píxeles por muestra, por lo que la cantidad de memoria disponible es un factor fundamental a tener en cuenta, puede verse en los resultados de la compilación (tabla 1), que la cantidad de memoria utilizada es el 81% de la disponible en el chip EP2C35F672C6, por lo que este fue el factor limitante para el tamaño de la imagen a procesar.

Se generó el código VHDL para el manejo de la interfaz *wishbone* del controlador de la memoria SDRAM provista en el kit de desarrollo DE2 de Altera, permitiendo el control integral de la SDRAM externa mediante la adaptación de un bloque estándar de uso libre.

Se desarrolló un controlador robusto y sencillo para la generación de las señales de sincronismo de un monitor VGA.

Se logró la sincronización de la lectura de memoria RAM embebida con el barrido del monitor VGA mediante un bloque buffer, generado con un código VHDL propio.

En la figura 10 se muestran los resultados del procesamiento realizado con Matlab® para 200 iteraciones (imagen de 300 x 300 píxeles) y la presentación en pantalla del sistema implementado para 100 iteraciones (imagen de 128 x 128 píxeles fotografiada del monitor conectado a la FPGA). La imagen corresponde a un experimento de secado de pintura sobre una moneda.

Tabla 1. Resultados de la compilación

	Resultados de la compilación		
	Utilizado	Disponible	%
Logic Elements	6586	33216	20
Total registers	1500	-	-
Total pins	165	475	35
Total memory bits	393216	483840	81
Embedded multiplier	0	70	0
Total PLLs	1	4	25

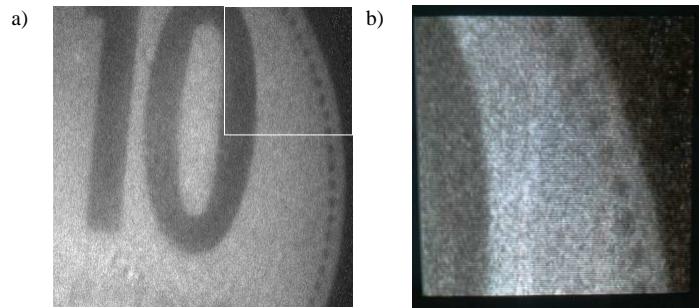


Figura 10. Comparación de resultados a) Procesado con Matlab® y b) Fotografía de la imagen resultante procesada con la FPGA. El recuadro en a) se corresponde con la imagen procesada en b).

REFERENCIAS

- [1] G.H. Sendra, S.E. Murielado, J.F. Gonzalez, R. Arizaga, M. Trivi, y H. Rabal, "Observación de movilidad en bacterias usando speckle dinámico," presentado en 91º Reunión Nacional de la Asociación Física Argentina (AFA), Merlo, San Luis, Argentina, 2006.
- [2] J. Amalvy, C. Lasquibar, R. Arizaga, H. Rabal, y M. Trivi, "Application of dynamic speckle interferometry to the drying of coatings," Progress in organic coatings 42(1-2), 89-99 (2001).
- [3] G.H. Sendra, J.C. Salerno, M.V. Kandus, C. Weber, R. Arizaga, H.J. Rabal, y M. Trivi, "Análisis de distintos tipos de maíz (*Zea mays L.*) mediante un método óptico (biospeckle laser) para diferenciar su calidad industrial," presentado en VIII Congreso Nacional de Maíz, Rosario, Santa Fe, Argentina., Proceeding de AIANBA, 396-399 (Ed. A. Portugal) (2005).
- [4] A. Mavilio Nuñez, M. Fernandez Limia, R. Arizaga, G.H. Sendra, M. Trivi, H. Rabal, "Utilización de speckle dinámico y de análisis por bandas de frecuencia para la detección de zonas dañadas en manzanas. En 13º Convención Científica de Ingeniería y Arquitectura y VII Taller Internacional de Enseñanza en Física e Ingeniería EFING'06, La Habana, Cuba (noviembre, 2006).
- [5] E. Todorovich, M. Vazquez, E. Cozzolino, F. Ferrara, G. Bioul, A.L. Dai Pra, L.I. Passoni, Speckle Signal Processing through FPGA, en uEA 2011, Libro de Memorias 33-38.
- [6] Passoni L.I., Modelos en Bioingeniería. Caracterización de imágenes estáticas y dinámicas. Tesis de Doctorado. (2005).



- [7] Sendra G. H., Análisis de Actividad en Patrones de speckle dinámico. Tesis de Doctorado. (2009).
- [8] Oularama A., Tribillion G., Duvernoy J., Biological activity measurement on botanical specimen surfaces using a temporal decorrelation effect of laser speckle. Journal of Modern Optics, v. 36, n.2, 165-179. (1989).
- [9] Rabal H.J., Braga R.A., *Dynamic Laser Speckle and Applications*. Taylor and Francis. NY. 304 pp. (2008).
- [10] Fujii H., Asakura T., Nohira K., Shintomi Y., y Ohura T., *Blood flow observed by timevarying laser speckle*, Optics Letters 10 (3), 104-106. (1985).
- [11] IEEE Standard VHDL Language Reference Manual, <http://standards.ieee.org/findstds/standard/1076-2008.html>
- [12] IEEE Standard Verilog Hardware Description Language, <http://standards.ieee.org/findstds/standard/1364-2001.html>
- [13] EP2C35F672C6 Altera datasheet, 13-16, <http://datasheet.octopart.com/EP2C35F672C6-Altera-datasheet-1735.pdf>
- [14] DE2 Manual, disponible en <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=53&No=30&PartNo=4>
- [15] Altera - <http://www.altera.com/>
- [16] Open Cores - <http://opencores.org/>
- [17] Wishbone B4, manual disponible en http://cdn.opencores.org/downloads/wbspec_b4.pdf

Generación de imágenes de alto rango dinámico con mapeo de tonos en un sistema embebido con arquitectura híbrida

Bayron Pérez Vega

Ingeniería Electrónica, ITCR - RidgeRun Engineering

Cartago, Costa Rica

E-Mail: bayron241@gmail.com

Resumen—El presente proyecto tiene como objetivo obtener imágenes de alto rango dinámico con mapeo de tonos logarítmico adaptativo sobre la arquitectura híbrida BeagleBoard-xM, además de mostrar el proceso de implementación de algoritmos de procesamiento de imágenes en arquitecturas híbridas que poseen un procesador de señales digitales (DSP) de Texas Instruments.

Con este fin, se desarrolla un sistema que permite capturar imágenes a diferentes exposiciones a partir de una señal de activación, estas imágenes en conjunto con los tiempos de exposición son las entradas al módulo generador de imágenes de alto rango dinámico.

La visualización de las imágenes de alto rango dinámico depende del hardware de los dispositivos de despliegue, debido a las limitaciones de hardware en estos dispositivos se debe aplicar a la imagen de alto rango dinámico un algoritmo de mapeo de tonos con la finalidad de reducir el contraste y la corrección gamma para compensar la no linealidad del receptor del dispositivo de visualización. La imagen resultante se almacena en el formato jpeg con el máximo factor de calidad.

La evaluación de los algoritmos se da por medio de los tiempos de ejecución e histogramas de frecuencias de las imágenes con diferentes exposiciones e imagen de alto rango dinámico de una escena estática.

Index Terms—HDR, Mapeo de tonos, DSP, codec, servidor, GStreamer, Codec Engine, iUniversal, CMEM, XDM, XDAIS, modelo de mundo gris, memoria contigua, espacio de color

I. INTRODUCCIÓN

En la actualidad existe un incremento en el desarrollo de aplicaciones en las áreas de medicina, control automático, redes, etc. en sistemas embebidos como la tarjeta BeagleBoard-xM, la cual está compuesta por un núcleo heterogéneo (dos procesadores distintos en un mismo chip, SoC), un procesador ARM cortex A8 como procesador de propósito general (GPP) y un procesador DSP C64x+ como un procesador de propósito específico [1]. Esta arquitectura se diseñó para el uso de aplicaciones multimedia como la generación de imágenes de alto rango dinámico.

Las imágenes de alto rango dinámico (o imágenes HDR por sus siglas en inglés High Dynamic Range) son un conjunto de técnicas que permiten un mejor rango dinámico de luminancia (intensidad de luz máxima menos intensidad de luz mínima) entre regiones saturadas y zonas oscuras de una imagen. El rango dinámico de las imágenes HDR está comprendido por el rango de la combinación de las imágenes digitales estándar.

La técnica de imágenes HDR surge debido a la limitación de rango dinámico que se presenta en los sensores de captura en comparación al rango que percibe el ojo humano, el cual se adapta a diferentes luminosidades a diferencia de los sensores de captura que no poseen esta capacidad.

La visualización de la imagen HDR en dispositivos que poseen un menor rango dinámico como monitores CRT, LCD, impresoras, etc. implica reducción del contraste o rango dinámico de luminancia de la imagen, esto se lleva a cabo mediante la técnica de mapeo de tonos.

En la figura 1 se muestra el diagrama de bloques general del sistema de generación de imágenes HDR con mapeo de tonos.

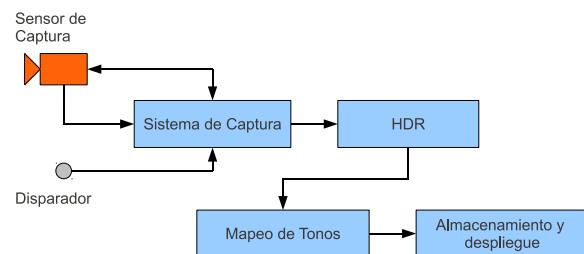


Figura 1. Diagrama general del sistema de generación HDRI con mapeo de tonos

El módulo sensor de captura permite adquirir imágenes del medio con variaciones en el tiempo de exposición, diferentes ganancias de los colores primarios, ajuste de balance de blancos, entre otros.

El módulo del sistema de captura, permite ajustar el sensor para obtener imágenes a diferentes exposiciones con sus respectivos tiempos de exposición, las imágenes son enviadas al sistema de generación de HDR para la obtención de la imagen. Luego la imagen es transformada por medio del sistema de mapeo de tonos a un formato como (jpeg o png), para que esta sea escrita en una unidad de almacenamiento y finalmente visualizar la imagen en un dispositivo de despliegue.

El presente trabajo se enfoca en mostrar la implementación de los algoritmos de imágenes HDR y mapeo de tonos en el sistema embebido BeagleBoard-xM, así como los resultados obtenidos en el proceso.

II. CONCEPTOS GENERALES

II-A. Imágenes de alto rango dinámico

Existen varias técnicas para mejorar el rango dinámico efectivo de cámaras usando imágenes de la misma escena (escena estática) de bajo rango dinámico (LDR, Low Dynamic Range) a diferentes tiempos de exposición del sensor.

Este enfoque requiere de una calibración inicial del sensor de captura para determinar la función de respuesta del sensor ante intensidades de luz. La función de respuesta del sensor corresponde a la función de mapeo de la intensidad de luz que llega a la cámara en valores de R , G y B . En [2] la respuesta del sensor se calcula mediante una calibración fotométrica, donde la respuesta del sensor se optimiza con todos los píxeles de las imágenes de entrada, asignando mayor peso a valores con exposiciones altas, lo que permite estabilidad en la respuesta de sensor ante ruido de datos de entrada.

Una vez realizada la calibración, se crea la imagen HDR con un promedio ponderado de las imágenes de entrada.

II-B. Mapeo de Tonos

En [3] se define el mapeo de tonos como una función que prepara una imagen HDR para ser visualizada en dispositivos LDR. No existe un algoritmo universal de mapeo de tonos.

La finalidad del mapeo de tonos es disminuir la brecha presente entre la percepción de una escena del mundo real y una desplegada en un dispositivo de visualización.

El algoritmo de mapeo logarítmico adaptativo es del tipo de operadores globales y fue desarrollado por Drago et ál. en [4]. El algoritmo realiza una compresión logarítmica de los valores de luminancia imitando la respuesta humana a la luz. Después de ser aplicado el mapeo de tonos a los píxeles, se debe hacer una corrección gamma para compensar la no linealidad del receptor del dispositivo de visualización.

II-C. Marco de trabajo del procesador DSP

Los paquetes de software que se utilizan son: *DSPLink*, *CMEM (Memoria Contigua)* [5], *Interfaz XDAIS*, *XDM* [6], *IUNIVERSAL* [7], *Codec Engine* y *RTSC* o componentes de software en tiempo real [8].

El desarrollo de aplicaciones en el procesador DSP requiere de tres paquetes: *Codec* (código que se ejecuta en el DSP), *Servidor* (maneja las llamadas remotas del algoritmo) y la *Aplicación* (cliente que se ejecuta en el ARM e invoca funciones del codec).

III. SISTEMA GENERACIÓN HDRI CON MAPEO DE TONOS

El módulo de cámara adaptable al sistema BeagleBoard-xM es de la familia LI-MOD, producido por Leopard Imaging en el cual se utiliza un sensor CMOS MT9V113 con resolución VGA (640×480) creado por la empresa Aptina Imaging [9].

III-A. Sistema de captura

El sistema de captura se ha encapsulado en el diseño del contenedor de GStreamer *camerabin2*.

En las funcionalidades de *camerabin2* no se incluye la posibilidad de tomar varias imágenes con un solo disparo ni

almacenar los tiempos de exposición de las imágenes, por lo que estas funcionalidades se agregan modificando el contenedor *gstwrappercamerabinsrc* (establece la comunicación con el sensor de captura).

III-B. Módulo de HDR

La etapa de HDR contempla la generación de imágenes HDR a partir de imágenes con múltiples exposiciones con el algoritmo de Robertson, Borman y Stevenson [2], además de consideraciones como conversión entre espacios de colores y utilización de elementos de GStreamer [10] para leer de archivos las imágenes provenientes del sistema de captura. La figura 2 comprende el módulo de HDR.

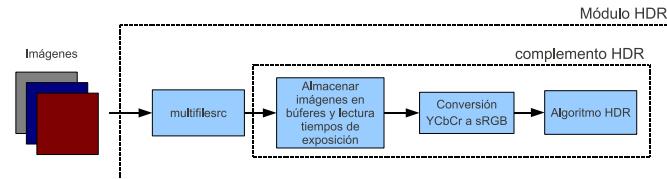


Figura 2. Diagrama de bloques del módulo HDR

La implementación del algoritmo de Robertson, Borman y Stevenson en el procesador ARM y en el DSP se basa en el paquete de software pfstools [11] desarrollado en el Instituto Max Planck [12].

La imagen HDR es el resultado de aplicar la respuesta del sensor a cada píxel de los canales R , G y B de las imágenes de entrada, para obtener las irradiancias. Si la respuesta del sensor no es conocida, se debe estimar mediante el proceso de calibración fotométrica. La respuesta del sensor es dada por el promedio de las respuestas de los canales R , G y B .

III-C. Módulo de mapeo de tonos

En la figura 3 se muestran las etapas que conforman el módulo de mapeo de tonos.



Figura 3. Diagrama de bloques del módulo de mapeo de tonos

La imagen HDR o de irradiancias tiene tres canales R , G y B . Si los valores máximos de cada canal difieren, existen problemas con regiones blancas. Caso similar ocurre con los valores promedio de cada canal, provocando problemas con el color gris en la imagen. Estos problemas son corregidos por medio de la teoría Retinex y el algoritmo de mundo gris, respectivamente.

Los algoritmos de mundo gris y de la teoría Retinex son incompatibles debido a las condiciones de las que se parten no se pueden satisfacer de forma conjunta. En el presente trabajo se utiliza por defecto el algoritmo de mundo gris.

La comprensión o reducción logarítmica de los valores de luminancia de la imagen HDR es llevaba a cabo por el operador de tonos logarítmico adaptativo.

Un convertidor del espacio de color RGB a XYZ permite tener un canal de luminancias que puede ser mapeado con el operador de tonos logarítmico adaptativo. Los valores del canal X y Z están dados por

$$\begin{aligned} f_{scale} &= \frac{l_{out}}{l_{in}} \\ X &= X f_{scale} \\ Z &= Z f_{scale} \end{aligned} \quad (1)$$

donde f_{scale} es el factor de escala.

La corrección gamma se aplica a los canales mapeados para ajustar los valores de los píxeles a la no linealidad de los dispositivos de visualización.

Las etapas del mapeo de tonos se agregan al complemento HDR como se muestra en la figura 4.

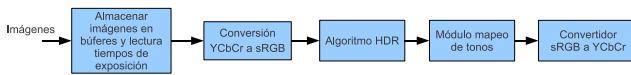


Figura 4. Diagrama de bloques del complemento HDR con la integración de las etapas del módulo mapeo de tonos

En las siguientes secciones se presenta la implementación propuesta para el complemento HDR en el ARM (gsthdriimagearm) y en el DSP (gsthdriimagedsp).

IV. IMPLEMENTACIÓN DEL COMPLEMENTO GSTHDRIMAGEARM

La imagen obtenida por el sistema de captura es leída mediante el elemento de GStreamer `multifilesrc`, enviada al complemento (almacenada en una estructura, `inbufs`) y escrita en un archivo con formato jpeg, png, etc.

La función `hdrimage_algorithm()` realiza el procesamiento de los algoritmos recibe la estructura y parámetros como: cantidad de imágenes, tiempos de exposición, factor gamma, parámetro Bias, tamaño de imagen, propiedades que indican si es necesaria la calibración, aplicar el modelo de mundo gris y el formato de entrada de las imágenes (YUV o RGB).

V. IMPLEMENTACIÓN DEL COMPLEMENTO GSTHDRIMAGEDSP

El desarrollo de la aplicación en el procesador DSP se divide en tres secciones: aplicación cliente, servidor y codec. La etapa de almacenar imágenes en búferes y la lectura de tiempos de exposición (figura 4) se implementa en la aplicación cliente, la cual mediante el servidor establece la comunicación con el codec, el cual contiene la implementación del complemento `gsthdriimagedsp`.

V-A. Aplicación cliente

En la aplicación cliente se establece la comunicación entre los procesador ARM y DSP, mediante el uso de la API de iUniversal, pasando parámetros y búferes hacia el codec.

La estructura `IHDR_Params` se extiende con los parámetros: calibrat, gwa, imgCount, luminance, biasVal y gammactf.

La función `UNIVERSAL_create()`, permite pasar la estructura de parámetros al codec, invocando las funciones `algAlloc` y `algInit` de la API de iUniversal. En la función `algInit` se inicializa la estructura parámetros del codec con los datos enviados a través de `UNIVERSAL_create()`. La función `algAlloc` reserva memoria para el objeto de la estructura `hdr` del codec.

Para el complemento `gsthdriimagedsp` los búferes de entrada son almacenados en búferes del tipo XDM, mediante la función `Memory_contigAlloc()`.

La función `Memory_contigAlloc()` reserva memoria de forma dinámica en la memoria compartida del procesador ARM y DSP. Los búferes XDM son pasados como parámetros al codec por medio de la función `process()` de la API de iUniversal.

Los tiempos de exposición de las imágenes de entrada se pasan al codec por medio de un búfer de entrada de XDM. El búfer de salida con la imagen HDR con mapeo de tonos se asigna a un búfer de XDM de salida. Esta a su vez se copia a un búfer de GStreamer `outbuf`, el cual es la salida de la aplicación cliente.

La respuesta del sensor se asigna a un búfer de entrada-salida de XDM. Cuando la propiedad indica lectura el búfer de XDM se comporta como un búfer de entrada y en el caso de escritura, el comportamiento es de un búfer de salida de XDM. La lectura/escritura de la respuesta del sensor en el archivo `respCurve.txt` se lleva a cabo en la aplicación cliente.

V-B. Servidor

Permite enlazar bibliotecas con el código que es ejecutado en el procesador DSP.

V-C. Codec

En el codec se implementa los bloques mostrados en la figura 4 considerando: la estructura `IHDR_Process`, el manejo de archivos desde la aplicación cliente, los algoritmos son en coma flotante y los búferes de entrada y salida son del tipo XDM, además de la arquitectura del procesador de DSP (coma fija con un emulador de coma flotante).

La implementación de los algoritmos en coma flotante tiene una consecuencia: la reducción del rendimiento del procesador, debido a la carga de operaciones de coma flotante a realizar en el emulador.

En el presente proyecto se portan ciertas funciones de manera local a coma fija con la biblioteca TMS320C64x+ IQmath [13] con la finalidad de analizar el rendimiento en la arquitectura de DSP.

V-D. Aspectos a considerar en el uso de la memoria

Las formas de reservar memoria de acuerdo a la API de iUniversal utilizan la funciones:

- *memtab*: tabla de memoria (memory table). Permite reservar memoria para búferes unidimensionales en el segmento de memoria DDR2 del procesador de DSP y se utiliza en la función *ialgAlloc* del codec.
- *Memory_contigAlloc*: permite reservar en el segmento de memoria compartida. La reserva de memoria se realiza desde la aplicación cliente y los datos son enviados al codec mediante los búferes de XDM.

Los búferes de XDM son unidimensionales, por lo que las matrices utilizadas en el procesamiento de los algoritmos de HDR y mapeo de tonos se descomponen en búferes unidimensionales para utilizar la función *memtab* o *Memory_contigAlloc()* en la reserva de memoria. Las limitaciones de las funciones para reservar memoria son:

- *memtab*: utiliza un segmento de memoria limitado de 6MiB. Con $1MiB = 2^{20}$ bytes.
- *Memory_contigAlloc()*: utiliza el segmento de memoria compartida de 30MiB. Cantidad de búferes de entrada y entrada-salida de XDM limitado a un máximo de 32.

La cantidad de memoria requerida por los búferes y matrices de procesamiento para los algoritmos considerando que la resolución de imágenes de entrada es de 640×480 (614400 bytes) y la cantidad de niveles de la respuesta del sensor (*input_level*) de $2^8 = 256$ es 19,01MiB. La cantidad de memoria que permite la función *memtab* no es suficiente para los búferes y matrices.

Incrementar en uno la cantidad de imágenes de entrada manteniendo la resolución requiere un adicional de memoria de 3,52MiB. En el caso de un aumento de la resolución a 1024×768 manteniendo la cantidad de imágenes de entrada requiere 31,07MiB de memoria adicional. Con la resolución de 1024×768 , un incremento de una imagen de entrada equivale a 16,45MiB de memoria adicional. Debido a lo anterior el segmento de memoria compartida se amplia de 30MiB a 160MiB.

Veintiseis son los búferes XDM requeridos para pasar los datos de los búferes y matrices de procesamiento, considerando tres imágenes como la cantidad de entrada del sistema. Adicionalmente uno para la respuesta del sensor y uno para los tiempos de exposición, para un total de 31 búferes. Esto queda próximo a los 32 búferes de XDM disponibles. En el caso de aumentar las imágenes de entrada en uno, se requieren adicionalmente tres búferes XDM.

Un objeto que contiene la estructura con los búferes y matrices de procesamiento de los algoritmos permite eliminar la limitación dada por la cantidad de búferes XDM. El puntero de la estructura *IHDR_Process* se pasa por medio de un búfer de XDM, coercionando al tipo *XDAS_Int8* y en el codec se coerciona al tipo de la estructura para recuperar los datos.

Consideraciones en la creación de la estructura (reserva de memoria) en la aplicación cliente son:

- El procesador DSP ve el segmento de memoria como direcciones físicas.
- El procesador ARM ve CMEM como direcciones virtuales.
- Los búferes de XDM hacen la traducción de dirección virtual a física.

El puntero de la estructura es mapeado a una dirección física por medio del búfer XDM. Los miembros de la estructura mantienen su dirección virtual, lo que impide que el procesador de DSP pueda utilizar los datos de los miembros de la estructura.

Mediante la función *CMEM_getPhys()* se traducen direcciones virtuales a físicas. Los búferes de la estructura *IHDR_Process* se reservan mediante la función *Memory_contigAlloc()*, la cual da un puntero a memoria virtual, este puntero a su vez es traducido por la función *CMEM_getPhys()* a una dirección física.

Finalmente, en la etapa de almacenamiento y visualización la imagen HDR con mapeo de tonos se escribe en un archivo y se convierte a un formato donde la imagen sea observable mediante algún visor de imágenes mediante una tubería o línea de proceso de GStreamer donde se convierte del formato *x-raw-yuv* a *jpeg*. El factor de calidad es el máximo permitido en el elemento (valor de 97).

En la siguientes secciones se muestran los resultados del sistema generación de HDRI con mapeo de tonos.

VI. SISTEMA DE CAPTURA

Se capturan tres imágenes con diferentes tiempos de exposición, una a exposición normal (1,409279 ms), imagen sub-expuesta a -2EV (0,352320 ms) y imagen sobre-expuesta a +2EV (4,580338 ms), como se muestra en la figura 5.



Figura 5. Escena estática obtenida por el sistema de captura. a). Imagen sub-expuesta. b). Imagen a exposición normal. c). Imagen sobre-expuesta

La captura de la imagen a exposición normal se da bajo los requisitos de habilitación de AE (Auto Exposición) y AWB (Auto Balance de Blancos), esto con la finalidad de que el sensor se ajuste al medio en que se encuentra.

VII. HDR Y MAPEO DE TONOS

VII-A. Algoritmos coma flotante, comparación ARM y DSP

La figura 6 muestra la imagen HDR con mapeo de tonos obtenida a partir de la secuencia de imágenes de la figura 5.

La función *process* comprende la generación de la imagen HDR sin la calibración del sensor de captura. Los tiempos de ejecución de las funciones que comprenden la función *process* obtenidos en el procesador ARM y en el procesador de DSP, se muestran en la tabla I.

El tiempo de ejecución total (función *process*) del complemento *gsthdriagedsp* es 74,94 % mayor que el complemento *gsthdriimagearm* (tabla I). La causa de este resultado es la implementación de una aplicación de coma flotante en una arquitectura de coma fija (procesador DSP), donde las operaciones son llevadas acabo mediante un emulador.



Figura 6. Imagen HDR con mapeo de tonos

Tabla I
TIEMPOS DE EJECUCIÓN EN EL PROCESADOR ARM Y DSP DE LAS FUNCIONES DEL COMPLEMENTO

Función	Parámetro	DSP (s)	ARM (s)	Porcentaje
getResponse	Promedio Desv. estánd	42,125230 0,005568	5,514449 0,005568	86,91
applyResponse	Promedio	4,425193	1,293702	70,77
toneMapOperator	Promedio Desv. estánd	33,449203 0,000605	14,086347 0,000005	57,89
process	Promedio Desv. estánd	85,956699 0,169723	21,542142 0,004891	74,94

VII-B. Comparación de los algoritmos en coma flotante y coma fija en el procesador de DSP

Las funciones *applyResponse* y *responseLinear* se portan a coma fija mediante la biblioteca IQMath.

En la tabla II se muestra que el tiempo de la función *applyResponse* en coma fija se reduce en 77,59 % con respecto a la implementación en coma flotante y la función *responseLinear* en 95,47 %. La reducción de estas funciones se refleja en la reducción de 4,28 % del tiempo de ejecución total de la aplicación (función *process*).

La reducción del tiempo de ejecución se debe a la optimización del código mediante la biblioteca IQMath y la arquitectura de coma fija del procesador DSP (se evita el uso del emulador de coma flotante).

VII-C. Comparación de los algoritmos en coma flotante en el procesador ARM y coma fija en el procesador de DSP

En la tabla III se muestran los tiempos de ejecución de las funciones *applyResponse* y *responseLinear* en los complementos *gsthdriimagearm* y *gsthdriagedsp*, considerando los tipos de arquitectura.

La implementación en coma fija del DSP muestra una disminución en el tiempo de ejecución con respecto a la de coma flotante en el ARM. El tiempo de ejecución de la función *applyResponse* se reduce en 23,34 % y la función *responseLinear* en 38,60 %.

Tabla II
TIEMPOS DE EJECUCIÓN EN EL PROCESADOR DSP PARA LAS FUNCIONES *applyResponse* Y *responseLinear* EN COMA FLOTANTE Y COMA FIJA

Función	Parámetro	Flotante (s)	Fija (s)	% Reduc
applyResponse	Promedio	4,425193	0,991735	77,59
ResponseLinear	Promedio	0,000773	0,000035	95,47
process	Promedio Desv. estánd	85,956699 0,169723	82,281642 0,301745	4,28

Tabla III
TIEMPOS DE EJECUCIÓN EN EL PROCESADOR ARM EN COMA FLOTANTE Y EN EL PROCESADOR DE DSP EN COMA FIJA PARA LAS FUNCIONES *applyResponse* Y *responseLinear*

Función	Parámetro	ARM (s)	DSP (s)	% Reduc
applyResponse	Promedio	1,293702	0,991735	23,34
ResponseLinear	Promedio	0,000171	0,000105	38,60
process	Promedio Desv. estánd	21,542142 0,069935	82,281642 0,549313	281,96

VII-D. Análisis de imágenes HDR con mapeo de tonos por medio de histogramas

En la presente sección se analiza la relación entre histograma de frecuencia e imágenes con diferentes exposiciones (imágenes de entrada del sistema de captura) y HDR con mapeo de tonos (obtenida en el procesador DSP).

En la figura 7 se muestra una imagen a exposición normal con el respectivo histograma de frecuencias. La distribución de frecuencias de la imagen a exposición normal presenta dos picos pronunciados que corresponden a zonas oscuras (aproximadamente 7000 píxeles) y zonas con saturación (aproximadamente 26000 píxeles).

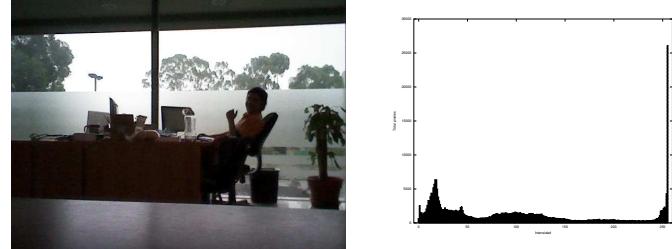


Figura 7. Imagen exposición normal con respectivo histograma de frecuencias

En la figura 8 se muestra una imagen sub-expuesta a -2EV con respecto a la imagen a exposición normal (figura 7) y el respectivo histograma de frecuencias.

Al sub-exponer en -2 EV se reduce la cantidad de luz que incide en el sensor en un factor de 4, reduciendo las zonas de saturación a consecuencia de un aumento de las zonas oscuras. La imagen sub-expuesta presenta una distribución de frecuencias de píxeles cercanas a cero. La cantidad de píxeles en la región oscura es aproximadamente 17000.

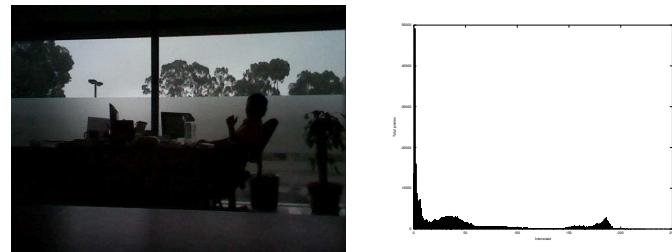


Figura 8. Imagen sub-expuesta con respectivo histograma de frecuencias

En la figura 9 se muestra una imagen sobre-expuesta a +2EV con respecto a la imagen a exposición normal (figura 7) y el respectivo histograma de frecuencias.

Al sobre-exponer en +2 EV se aumenta la cantidad de luz que incide en el sensor en un factor de 4, aumentando las zonas



Figura 9. Imagen sobre-expuesta con respectivo histograma de frecuencias

de saturación a consecuencia de una reducción de las zonas oscuras. La imagen sobre-expuesta presenta una distribución de frecuencias de píxeles cercanas a 255. La cantidad de píxeles en la región de saturación es aproximadamente 66000.

La figura 10 se muestra la imagen HDR con mapeo de tonos obtenida a partir de la secuencia de imágenes de la figuras 7, 8 y 9. Además, el respectivo histograma de frecuencias.



Figura 10. Imagen HDR con mapeo de tonos y respectivo histograma

La imagen HDR con mapeo de tonos presenta un histograma de frecuencias uniformemente distribuido en el rango de valores entre 0 y 255. Los extremos (zonas oscuras y con saturación) tienden a cero evitando la pérdida de información en las regiones oscuras y de saturación.

VIII. CONCLUSIONES

En el presente trabajo se muestra el proceso de diseño e implementación de la aplicación de generación de HDRI con mapeo de tonos logarítmico adaptativo en la plataforma híbrida BeagleBoard-xM. Para ello se utilizó el sensor de captura MT9V113 y la interfaz de programación GStreamer.

Se ha diseñado un módulo de captura el cual permite obtener imágenes a diferentes exposiciones considerando la imagen a exposición normal como referencia. La señal de activación que inicia el proceso de captura de las imágenes con diferentes exposiciones se implementó mediante el evento del botón de usuario en la tarjeta BeagleBoard-xM.

En la implementación del complemento *hdrimagedsp* se muestra el proceso de pasar una estructura con miembros del tipo punteros como parámetro a través de un búfer de XDM, haciendo uso de la función *Memory_contigAlloc()* la cual permite reservar memoria en el segmento de memoria contigua y *CMEM_getPhys()* que permite obtener la dirección física mediante la traducción de una dirección virtual, considerando que el procesador ARM ve el segmento de memoria compartida como direcciones virtuales y el procesador DSP como direcciones físicas.

La implementación de los algoritmos de mundo gris y la teoría Retinex permite eliminar la saturación a nivel global de la imagen HDR con mapeo de tonos y problemas en regiones blancas, respectivamente.

La implementación en coma flotante en el procesador de DSP (arquitectura coma fija) es un 74,94 % mayor que la implementación en el procesador ARM (arquitectura coma flotante).

La implementación de las funciones *applyResponse* y *responseLinear* en coma fija en en DSP se reduce en 23,34 % y 38,60 % respectivamente en comparación a la implementación en coma flotante del ARM.

Portar el código de coma flotante a coma fija del complemento *gsthdriimagedsp* permitiría optimizar el tiempo de ejecución de la aplicación de generación de imágenes con mapeo de tonos. Adicional a esta optimización se pueden realizar optimizaciones con técnicas como: reducción de la sobrecarga de ciclo, *restrict qualifiers*, utilización de estructuras por referencia, utilización del pragma *MUST_ITERATE()* y *_nassert()*, optimización en declaraciones *if* y el uso de intrinsics.

La generación de imágenes HDR con mapeo de tonos se restringe a escenas estáticas para evitar los *Ghost Artefacts*. En los sistemas de captura en los que el usuario interviene físicamente con la activación de la cámara se producen leves movimientos en la escena generando píxeles defectuosos en los bordes de la imagen HDR con mapeo de tonos; una alternativa para evitar el problema de movimiento en el módulo de captura es la implementación de un módulo de activación remota.

REFERENCIAS

- [1] Beagleboard.org. (2011) Beagleboard-xm product details. [Online]. Available: <http://beagleboard.org/hardware-xM>
- [2] M. A. Robertson, S. Borman, and R. L. Stevenson, "Estimation-theoretic approach to dynamic range improvement using multiple exposures," in *Journal of Electronic Imaging*, vol. 12, no. 2, April 2003, pp. 219–228.
- [3] E. Reinhard, G. Ward, S. Pattanaik, and P.Debevec, *High Dynamic Range Imaging: Acquisition, display and image-based lighting*. Morgan Kaufmann Publisher, 2005.
- [4] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes," in *Computer Graphics Forum, proceedings of Eurographics 2003* 22(3), 2003, p. 419–426.
- [5] TI. (2010) Cmem-overview. [Online]. Available: http://processors.wiki.ti.com/index.php/CMEM__Overview
- [6] T. I. Incorporated, *xDAIS-DM (Digital Media) User Guide. Literature Number: SPRUEC8B*. Texas Instruments, 2007.
- [7] TI. (2010) Getting started with iuniversal. [Online]. Available: http://processors.wiki.ti.com/index.php/Getting_started_with_IUNIVERSAL/
- [8] T. I. Incorporated, *Codec Engine Algorithm Creator Users Guide. Literature Number: SPRUED6C*. Texas Instruments, 2007.
- [9] A. Imaging. (2011) Image sensors. [Online]. Available: http://www.aptina.com/products/image_sensors/
- [10] (2011) Gstreamer, open source multimedia framework. [Online]. Available: <http://gstreamer.freedesktop.org/>
- [11] G. Krawczyk and R. Mantiuk. (2011) pfstools about. [Online]. Available: <http://pfstools.sourceforge.net/>
- [12] T. Aydin, M. Cadík, M. Goesele, V. Havran, G. Krawczyk, R. Mantiuk, K. Myszkowski, and A. Yoshida. (2011) High dynamic range image and video processing. [Online]. Available: <http://www.mpi-inf.mpg.de/resources/hdr/index.html>
- [13] T. I. Incorporated. (2008) Tms320c64x+ iqmath library. [Online]. Available: <http://www.ti.com/litv/pdf/sprugg9>



Agile Embedded Systems Design Capstone Course

The MAVRK Experience

Antonio F Mondragon-Torres, Troy Hawley, Alan Lee, and Ryan Masicci,
Electrical Computer and Telecommunications Engineering Technology
Rochester Institute of Technology
Rochester, NY, USA
afmiec@rit.edu

Abstract— This work is an example of a capstone senior design project in the last of a sequence of three Embedded Systems Design courses. The innovation on the presented work strives on the use of Agile project management techniques, open ended projects, ideation, market targeting, open source platforms, enhancing communications skills, and interfacing with engineers in the field. The course is very dynamic and there are just 10 weeks from product ideation to prototyping and final demonstration. The students have to demonstrate their product, upload a video demonstration, give a formal presentation before an audience, and make their design completely reusable by properly documenting and making use of version control repositories.

Embedded Systems Design; Agile Project Management; Open Source Platforms;

I. INTRODUCTION

For the last of a sequence of embedded systems design courses equivalent to a senior design course, we took a non-conventional approach in which the students work as a team to create an innovative open ended project targeted to a particular market segment or market need.

What is different to a traditional course is that students were exposed to Agile Project Management (PM) methodologies and we started using an open-source platform denominated MAVRK[1] (Modular and Versatile Reference Kit) that at the time was still under development. The students were able to contribute to the development of the platform as well as to the creation of a Weather Station platform for college use. The design included an FPGA platform with a graphical user interface and touchscreen that was used to control and monitor weather variables, it communicated serially with the remote MAVRK sensing and control platform.

II. CAPSTONE COURSE REALIZATION

A. Non-Conventional Approach

For the offering of this course, we have taken a different approach to what a senior capstone project traditionally is. While in a senior capstone project, a more traditional approach to project management and product development is taken, in this course the teams are structured like a small start-up company where the students will create an innovative product that will have differentiation to commercial available products and will target a specific market segment.

Another differentiation is IP reuse and collaboration across teams working on different projects. All that teams are required is to cite external consulting at the time of developing their budgets. Last but not least, we apply Agile project management techniques [2-5] which lead to innovation by allowing the students to experiment and mature their ideas. A previous version of this course was offered, but in that particular offering all teams worked competitively on an MP3 player design[2], while in this course each team choose a unique project¹. A key component in this effort has been the participation of different companies as “Product owners” which help to specify, shape and collaborate during the project.

B. Innovation

There is a big university effort towards innovation lectures we talk about global competition and first to market constraints as well as product differentiation. We encourage component reuse in order to be more efficient and to spend time and resources on those aspects of the product that will make a differentiation factor with the competition.

C. Team Work

Students are encouraged to collaborate within and across teams; the only constraint imposed is to cite collaboration. In cases where students need to modify code or documentation done by previous teams in other offerings, they have to comment on the changes made and make both documentation and code available for future generations.

D. Open Ended

The project goals are open ended, this means that there is an initial vision of what the product will be, but the product is allowed to change direction at every “Sprint” (Sprint is equivalent to a “Milestone” in traditional project management) based on feedback from the team and the product owners/customers. At the end of each Sprint, the product should be functional to a certain degree of completion. This is a two edge sword in the sense that students could misinterpret the concept and do too little effort instead of trying to progress as much as they can at every Sprint.

E. Target Market

Project specifications should be focused and we do not need to reinvent the wheel every time we work on an engineering project. The market is really dynamic and those who come first

¹ <http://antoniofmondragont.wordpress.com/student-projects-embedded-systems-design-iii-spring-2010/>



with an innovation are likely to remain the leaders. Followers need to catch up and sometimes they could vanish in this quest.

We encourage students to think a particular market that could benefit from their product. For example, it may be targeted for the masses, but that could be difficult to achieve. Students rather try to find a particular problem to be solved and attack it in an innovative way. One simple example is to offer products and services to the “Baby Boomers” generation which are currently transitioning to retirement, are wealthy and require specific services not offered before.

F. New Open Source Platform

As was mentioned in the introduction, students benefit to work on open source platforms since these can be assimilated easily and could start contributing to the development in a very natural and efficient form. While semiconductor companies’ technical support is tied to big customers, students do not have any influence on product definitions and thus are not considered for technical support. If a company offers an open source platform and allows global development, students could start participating immediately to the development and maturity of the platform. There are countless open source projects that are generating a tsunami in embedded systems design such as the Arduino[6], mbed[7] and BeagleBoard[8] communities to mention some of the ones that the authors are familiar with.

G. No User Guides or Kitchen Recipes

While students require a positive experience in a capstone course, the objective of this course is more of a reality check experience while they are still taking courses. It is very simple for students to go through instructions given in the laboratory manuals, or ask the instructor who has already solved the problem and knows the solutions. When there is no right or wrong solutions, students are exposed to real life situations, where a manager is going to ask them to perform a particular task, and in most of the cases they will require to learn the skills required to accomplish the assigned task. On the other hand a small but important number of these students could start a venture where all the innovation and ideation concepts could be applied to create the next wave of new products or services. These are some of the reasons why in this course there are no guidelines and it is valid to reuse intellectual property (IP) blocks created by former students or available as open source.

H. Contribution to the Platform

As mentioned before, the platforms chosen are open source and in constant evolution. Our students are involved in discovering alternative ways to use the platform according to their needs, in some cases this involves software, but in others it involves hardware. All the work performed by the students is saved in a version control repository (e.g. SubVersion), auto-documented with doxygen[9], and with instructions on how to recreate the project.

I. Industrial Experience

While it could seem that the open ended projects could be not very demanding, an emphasis on getting the best project achievable is given to students by encouraging them to develop professional projects that they could use to promote themselves while looking for a job. Nowadays it is very important for students to be able to push their resume to the top of the pile,

by proving to employers that they are capable of producing high quality, independent and focused products.

J. One to One Interaction with Engineers

One of the key outcomes of the course is on enhancing communication skills and to be able to express their ideas. While this project could be determined by students initiative, it could also solve a particular problem proposed by the “Product Owner” which could be internal (a professor) or external (an engineer at a company). It has been very important that students interact among themselves and also with external engineers already in the field and learn how to communicate and express their ideas in a concise and effective form. They should take into account that a company is a very dynamic environment and the engineers that are sponsoring the project are very busy and need to be presented with to the point information and clear requests.

K. Real Remote Project Development and Collaboration

For some of the projects we have been lucky that the students have been able to have remote projects kick-off meetings, as well as end of Sprint meetings to update the product owners on the status of the projects.

III. THE PROJECT – A WEATHER STATION

A. Background

Students have been exposed at different levels to the Altera DE-2 board for four courses in the program. The Altera DE-2 board is an educational system with a Cyclone II FPGA running at a nominal frequency of 50 MHz. In addition to performing digital design on the FPGA using VHDL as the primary hardware description language, the previous two courses have been based on the integration of the soft core NIOS II processor to be able to create programmable Systems on a Chip (SoCs). In a previous offering of the course[2], teams worked for the first time on bringing up an LCD Touchscreen which has not been used in the past on any course. Students in this course were able to leverage on previous IP and documentation elaborated by former students. This is shown in Figure 1 where the actual GUI of the weather station is shown.

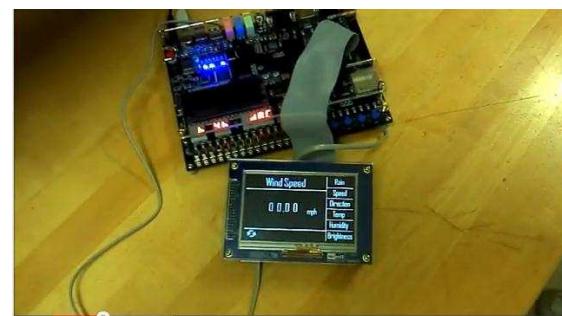


Figure 1. Altera DE-2 board used as part of the Embedded Systems Design courses with LCD touchscreen.

Students have been exposed in their sophomore year to a microcontrollers course and some of them have already been exposed to the MSP430 microcontroller used in the MAVRK platform. The MAVRK is a rapid prototyping platform which is targeted to allow engineers in the field to come up with fast implementation of their designs by using evaluation modules

(EVMs). The MAVRK platform and the actual project implementation are shown in Figure 2.



Figure 2. The MAVRK platform used at the project that is presented in this article. Additional PCB boards were designed as part of the project.

B. Specifications

Students came with the idea to design a weather station that could be used as a platform to interface with different embedded systems. A weather station able to measure parameters such as:

- Wind speed
- Wind direction
- Rain precipitation
- Temperature
- Ambient light
- Humidity

For this project, an inexpensive weather vane was available for the team to instrument as shown in Figure 3.



Figure 3. The weather station vane used was a commercially available product and the project involved the interfacing to the MAVRK platform.

C. Agile Project Management

While in this paper we cannot give a complete explanation of agile methodologies, we will give an example of one of the tools students used to track their project. The first stage in the project is to define a wish list with all the features they want to implement. Traditionally this can be done with post-its but in this case we used an issue tracker program called Jira² where students can list their wish lists on a planning board as seen in Figure 4. Students need to classify the tasks according to the Sprints, in our case we have 10 weeks and we divide this in two week Sprints 0.1, 0.3, 0.5, 0.8 and 1.0. The wish list become

tasks and is scheduled in a particular Sprint. Each task should be accomplishable in one Sprint. If it is predicted that a particular task will not be finished in one Sprint, it should be further divided into subtasks. This backlog of tasks will be initially be filled and will continue to be updated for the life of the project, as some tasks are given lower priorities as new tasks are discovered. Once the Sprint start, the task board shows the “To Do” tasks, the “In Progress” tasks and the “Done” tasks. This gives a very clear view of the state of the Sprint and anybody can go to the “To Do” column and self-assign a task. The task board is shown in Figure 5.

A very useful graphical tool is the “Burndown” chart that gives information on the initial number of tasks at the beginning of the Sprint, the predicted burndown rate, the number of new created tasks, the remaining values and a guideline. As can be observed in Figure 6, for this particular Sprint, students were not able to achieve their goals in terms of the number of tasks completed within the Sprint. In theory the red line and the green line should intersect at the end of the Sprint and the green has to follow the red. The orange line gives you the rate at which the tasks should be completed in one day in order to meet the burndown rate and this match the guideline. We can also observe that the blue line indicated that new tasks were added in the middle of the Sprint. This should not have happened, but it means that students discovered something that they did not realize before and then had to be included.

Does this mean that Agile project management methodology does not work? No, it means that the students are been exposed to this methodology for the first time and also they are working on their own for their first time by defining and creating their own tasks. This means that they do not have the skills required to predict accurately the time duration of particular tasks and also that they could not define perfectly all the tasks to be done. These situations are a great learning experience and helps with documenting and keeping track of team activities.

D. Hardware

In this particular case the platforms to be interfaced would be the DE-2 FPGA board with a LCD touch-screen as the user interface and control and the MAVRK motherboard and EVM modules.

The DE-2 platform will be in charge of the user interface and light control. For this purpose Altera SOPC builder was used to create the required hardware to control the LCD touch-screen, a simulated home light dimmer and the serial communication interface with the MAVRK. A Nios II processor was instantiated and programmed to accomplish these tasks. Additional hardware components were created in VHDL and added as custom components to the NIOS II system

The MAVRK platform is in charge of all measurement and control functions, both analog and digital. The boards will communicate through an asynchronous serial RS-232 communications link and a protocol with a simple packet structure was devised to accomplish this task. Asynchronous serial communications were chosen in order for the students to realize their own rules and finite state machines to handle inter

² <http://www.atlassian.com/>



platform communications. There are some other communication standards and protocols that could have been used such as I2C and SPI, but in this case we wanted students to realize the efforts required to communicate two asynchronous devices. In addition to this decision, all embedded systems usually have this type of interface either as a direct link or as a backdoor for debugging. Moreover, a printed circuit board was designed on-site to create an RS-232 interface to the MAVRK platform. This exposed students to learn on how to adapt their ideas to the selected platform.

E. Software

Programs for the NIOS II processor and for the MAVRK were all done in embedded C language. High reutilization was achieved in both platforms by leveraging on previous projects for the Altera platform and on libraries available on the MAVRK platform. This allowed the students to focus on the differentiation on their products by creating a more advanced graphical user interface and to communicate the platforms using serial communications.

F. Prototyping

1) Common Platform

As mentioned before, students have been exposed for four quarters to the DE-2 and are comfortable designing with this platform. The constraints are that this platform does not contain any available analog interfacing ports that could be used to extract information from the weather sensors. Also the touch screen is to be installed on a convenient place for the user.

2) Open Source Platform

The MAVRK platform has a series of slots that allow different types of components to be integrated. Following a traditional analog processing signal chain these are:

- Signal Conditioning (SC) slots
- Analog Front End (AFE) slots
- Radio Frequency (RF) slots

The MAVRK is based on the MSP430 and in particular, a MSP430F5438A is used as the main processor. Each slot has a combination of analog signals, digital signals and communication interfaces. So in addition to the traditional analog signal processing done on senior projects, the students are exposed to serial protocols and interfaces such as:

- Serial Parallel Interface (SPI)
- Inter-Integrated Communications I²C bus
- Universal Asynchronous Receiver and Transmitter (UART)

3) Putting all together

The Agile methodology calls for self-organization. One student specialized in the MAVRK platform, both hardware/software and serial communications. Another student on the graphical user interface (GUI) and serial DE-2 serial communications, while the other student was more on the hardware components to be interfaced to the NIOS II system.

Even that each student had a different specialization; the three students were collaborating where it was needed, thus

maximizing their time and achieving better results at the end of each Sprint.

G. Demonstration and Dissemination

In order for students to get a passing grade in the course, the students have to:

- 1) Give an In Situ demonstration of their working project before their final presentation to an audience of students, faculty, and family.
- 2) Upload a YouTube video with a presentation of their final product.
- 3) Present their work formally to an audience using PowerPoint and the video uploaded. In this case several engineers who were “Product Owners” joined the presentations remotely and participated in the Q&A.
- 4) Prepare documentation on how their project could be reproduced and lead the course instructor on how to reproduce their project based solely on documentation and code available on the version control repository.

IV. RESULTS

In particular, this project was very successful and the outcomes were recognized by Texas Instruments in their internal communications and external communications. Students were very motivated to see all their work recognized and prepared them for the workforce by applying the workable skills obtained through the program that led them succeed in this project. The final presentation and YouTube video are available at:

<http://www.youtube.com/watch?v=Sa7ykLBKmT4>
<http://www.slideshare.net/afmjee/weather-meter>

V. LESSONS LEARNED

A. Creativity and work under pressure

Students are capable of being creative at the same time dealing with time pressure due to the finite length of the course. In this particular course there are just 10 weeks available from idea to demonstration and that is the reason component and IP reuse is the key to succeed. Students can concentrate on the main product differentiation by gathering usable blocks.

B. Professional Communication

Students were able to communicate effectively with design engineers in the field to accomplish their goals that were done through remote meetings, email and version controlled repositories.

C. Reality Check

While we present here that everything is perfect and all students are motivated, there have been other teams that were not able to organize during the 10 available weeks or that faced road blocks that were not being able to overcome. Traditionally throughout the program, students are given laboratory exercises which guide them step by step and let them know what the desired results are. In courses like the presented here some students could be overwhelmed by not having the experience to take decisions by themselves, to propose innovative ways to perform the tasks or how to organize among themselves. In this



case as mentioned Agile PM methodologies can be a double edge sword where students do not know even how to begin.

VI. CONCLUSIONS AND FUTURE WORK

By using a new platform, there are no user guides, kitchen recipes or cookie cutter laboratory exercises; all development is new and limited by student's initiative and engagement. In this particular experience, a one to one interaction with real development engineers had to be performed in order to bring the project to a reasonable state of completion.

The students enjoyed the interaction with their industrial colleagues, contributed to the platform development and were exposed to a real remote project development and collaboration. The lessons learned from this open ended project are that: students are capable of being creative at the same time dealing with time pressure due to the finite length of the course, and are able to communicate effectively with design engineers in the field. Students made a formal presentation of their project, give live demonstrations as well as prepared and edited a video for further dissemination and to set precedence on their experience. The company used this video internally to demonstrate the platform capabilities.

Based on this work a new project has been proposed and students are currently working on the wireless implementation of the weather station by leveraging on the wireless communications between the main MAVRK board and a smaller µMARK board with RF capabilities. It is intended that the new team will install permanently the weather station and this will allow us to perform continued interfacing with the platform for future projects or courses.

ACKNOWLEDGMENT

This work is the result of the support from multiple people and multiple companies who have supported the course. We

would like to acknowledge Texas Instruments MAVRK team for providing the resources both material and human to be able to participate in the early stages of MAVRK development. We would like to acknowledge Altera University program that has been supporting this course by providing hardware and software components to expose students to advanced FPGA design. We would like to thank Atlassian for granting the licenses for their Agile project management suite which made this course possible. We would also like to thank the ECTET department at RIT for providing the computer and financial resources to build the product.

REFERENCES

- [1] Texas-Instruments. (2012, 04/22/2012). MAVRK Introduction - Texas Instruments Embedded Processors Wiki. Available: http://processors.wiki.ti.com/index.php/MAVRK_Introduction
- [2] A. F. Mondragon-Torres, A. Kozitsky, C. Bundick, E. Mc Kenna Jr., E. Alley, M. Lloyd, P. Stanley, and R. Lane, "Work in progress: An Agile Embedded Systems Capstone Course," presented at the 2011 Frontiers in Education, Rapid City, SD 2011.
- [3] J. A. Highsmith, "Agile project management creating innovative products / Jim Highsmith," ed: Upper Saddle River, NJ Addison-Wesley, c2010.
- [4] C. Larman, Agile and Iterative Development: A Manager's Guide: Addison-Wesley Professional, 2003.
- [5] R. Goering. (2010). Can Agile Software Development Methods help SoC design? Available: <http://www.cadence.com/Community/blogs/ii/archive/2010/08/18/can-agile-software-development-methods-help-soc-design.aspx>
- [6] (2012). Arduino - HomePage. Available: <http://www.arduino.cc/>
- [7] (2012). Rapid Prototyping for Microcontrollers | mbed. Available: <http://mbed.org/>
- [8] (2012, 04/22/2012). BeagleBoard.org - default. Available: <http://beagleboard.org/>
- [9] (2012, 04/22/2012). Doxygen. Available: <http://www.stack.nl/~dimitri/doxygen/manual.html>

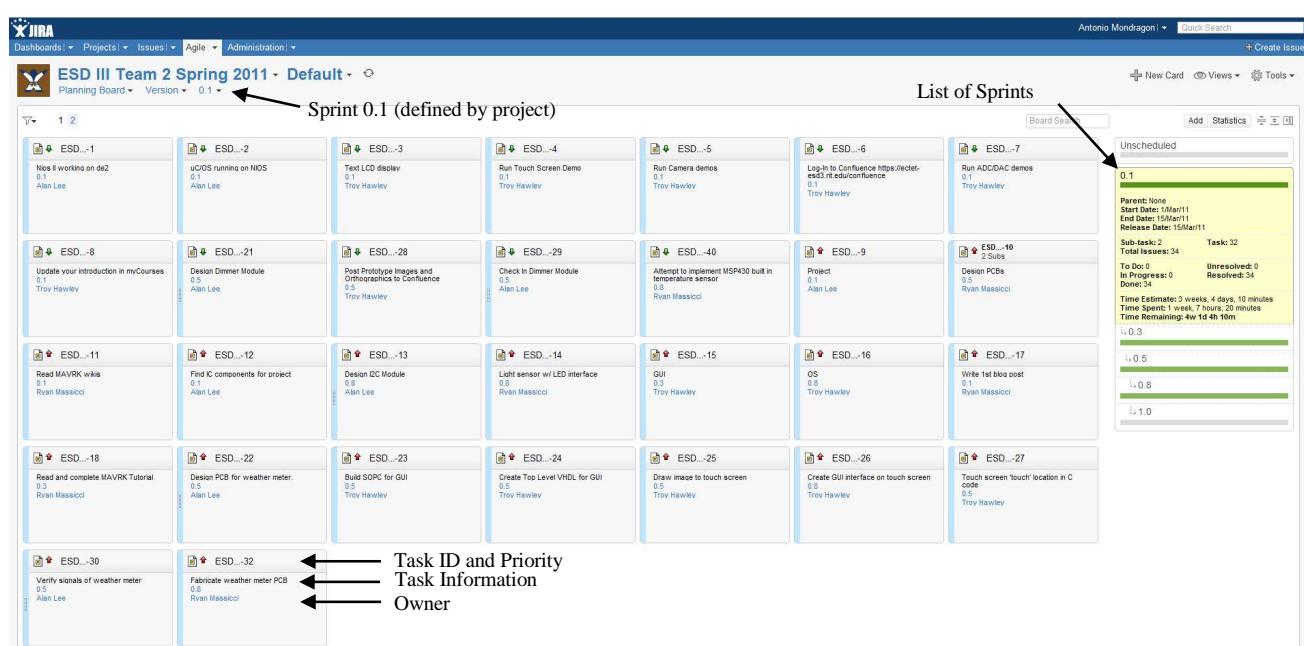


Figure 4. Jira planning board. All the tasks are entered in this view. At each Sprint tasks are prioritized and scheduled. If a task cannot be completed in one Sprint, then it should be divided into subtasks.

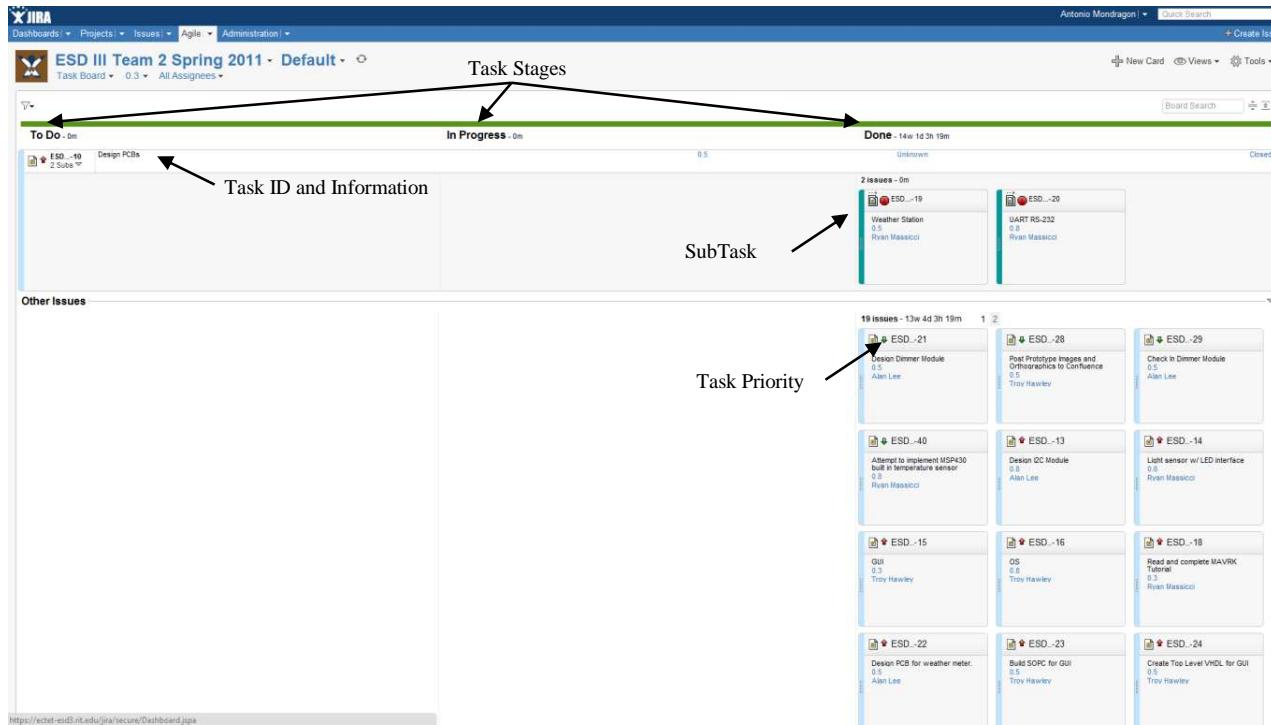


Figure 5. Jira Task Board example. As can be observed, all tasks for Sprint 0.3 have been completed. Also it can be observed that the task "Design PCBs" (upper left corner) was subdivided into two tasks. The tasks were assigned priorities previous to scheduling (represented by the arrow icons on each task).

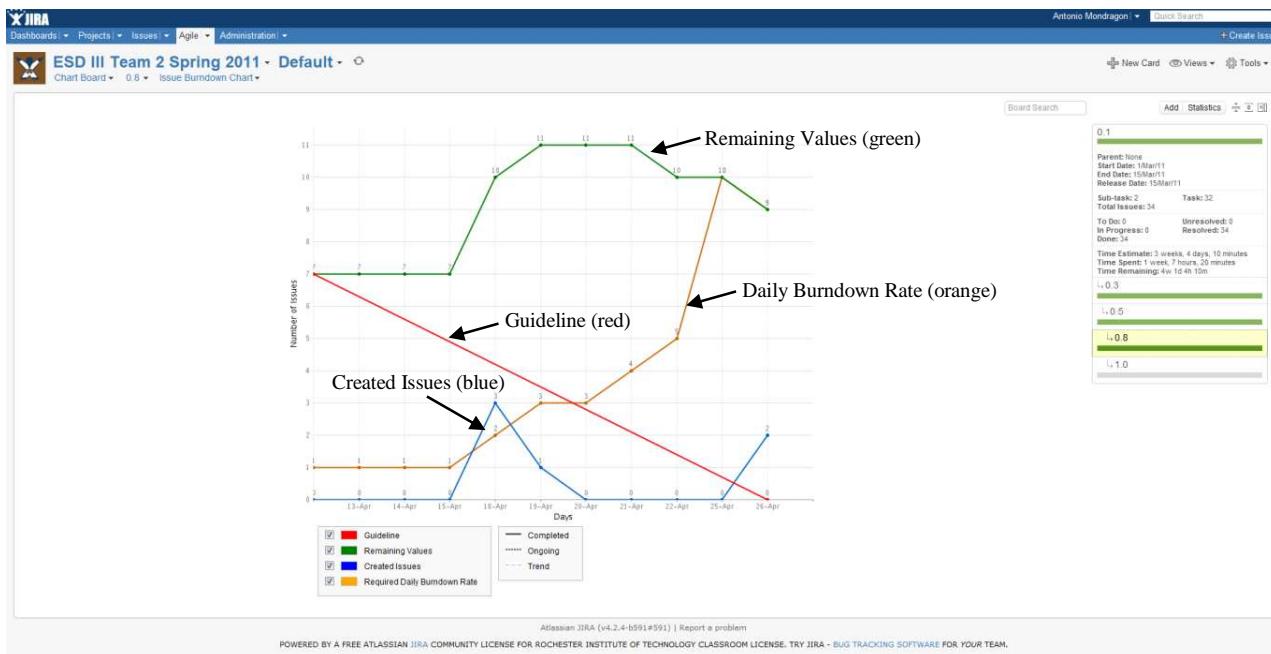


Figure 6. Jira issue burndown chart for Sprint 0.8. The green line represents the tasks in progress. The orange line represents the burndown rate, which is the number of tasks to be finish each day to meet the guideline (red line). The blue line represents the newly created tasks per day (In theory it should be zero).





Foro Tecnológico

Pósters

Implementación de Sistemas Embebidos





Control difuso basado en microcontrolador para la producción de biogás en digestión anaerobia tipo batch de fracción orgánica de residuos sólidos

Ilber Adonayt Ruge Ruge

Docente Universidad de Cundinamarca
Investigador grupo GITEINCO
Fusagasugá, Cundinamarca - Colombia
iruge@mail.unicundi.edu.co

Mauricio Hernandez Sarabia

Docente Universidad de Ibagué
Investigador grupo GMAE
Ibagué, Tolima - Colombia
Mauricio.hernandez@unibague.edu.co

Resumen—El manejo de los residuos sólidos generados por la población se convierte en un factor relevante por su impacto social y ambiental. Un manejo inadecuado de los mismos provoca entre otros aspectos enfermedades, contaminación de fuentes hídricas, del suelo y del aire por solo mencionar algunos. El biogás contiene un alto porcentaje en metano, CH₄ (entre 50-70%), por lo que es susceptible de un aprovechamiento energético mediante su combustión en motores, en turbinas o en calderas, bien sólo o mezclado con otro combustible. Un proceso de aprovechamiento óptimo requiere de la regulación de algunas variables como temperatura, pH, presión, agitación entre otras, para que el proceso de fermentación de los residuos de cómo resultado una buena concentración de CH₄ del biogás producido, lo cual se puede a llegar a conseguir si se piensa en la incorporación de elementos tecnológicos que permitan controlar correctamente dicho proceso. Este trabajo de investigación plantea el desarrollo de un sistema de control embebido basado en microcontrolador para la regulación de la temperatura de un digestor anaeróbico tipo batch para la generación de biogás a partir de la fracción orgánica de residuos sólidos orgánicos, con el propósito de mostrar una alternativa de desarrollo a bajo costo, así como aportar un conocimiento más en el campo de la digestión anaeróbica como alternativa energética y de saneamiento ambiental, pues en Colombia este tipo de investigaciones no ha sido ampliamente realizado.

Palabras claves: biogás, control difuso, microcontrolador, digestión anaerobia.

I. INTRODUCCION

El manejo de los residuos sólidos generados por la población se convierte en un factor relevante por su impacto social y ambiental. Un manejo inadecuado de los mismos provoca entre otros aspectos enfermedades, contaminación de fuentes hídricas, del suelo y del aire por solo mencionar algunos. Todos estos elementos afectan directamente el bienestar de la población y son alterados directamente por el crecimiento de la población y el aumento en la producción de residuos.

La digestión anaerobia es un proceso biológico en el que la materia orgánica, en ausencia de oxígeno, y mediante la

acción de un grupo de bacterias específicas, se descompone en productos gaseosos o “biogás” (CH₄, CO₂, H₂, H₂S, etc.), en digestato, que es una mezcla de productos minerales (N, P, K, Ca, etc.) y compuestos de difícil degradación.

El biogás contiene un alto porcentaje en metano, CH₄ (entre 50-70%), por lo que es susceptible de un aprovechamiento energético mediante su combustión en motores, en turbinas o en calderas, bien sólo o mezclado con otro combustible (IDAE, 2007).

Por lo anterior se plantea el desarrollo de este trabajo de investigación, en busca de aportar un conocimiento más en el campo de la digestión anaeróbica como alternativa energética y de saneamiento ambiental, y además como el uso de microcontroladores puede ser una excelente alternativa si se piensa en sistemas de control a bajo costo.

II. DESCRIPCIÓN DEL EXPERIMENTO

Se requiere someter a un proceso de digestión anaeróbica, una muestra de materia orgánica representada exclusivamente por los desechos de alimentos provenientes de restaurante, y evaluar el efecto de la temperatura en la calidad de la producción de biogás. Según datos mostrados por (W. Man-Chan, 2005), las fluctuaciones de temperatura en un proceso de digestión anaeróbica de residuos sólidos orgánicos municipales, puede influir en la producción de biogás y la concentración del CH₄ del mismo, por tanto la variable temperatura se convierte en la variable que se desea controlar para evaluar tales efectos en la calidad del biogás generado en el proceso de digestión anaerobia. La calidad del biogás se mide en términos de la Concentración Volumétrica del Metano (CH₄) y Dióxido de Carbono (CO₂), puesto que la calidad del biogás depende principalmente de la presencia de estos gases (T. Mandal, 1999; P. Vindis, 2008). Por otro lado, según lo expuesto por (Fernández, 2007), los Residuos Sólidos Urbanos de origen Domiciliario contienen un promedio del 70% de contenido orgánico, lo cual lo hace adecuado para el objeto de estudio.

Los desechos de alimentos fueron obtenidos del Restaurante Manila's Food del Centro Comercial Manila de la ciudad de Fusagasugá departamento de Cundinamarca. Los residuos de comida recolectados, consisten de cascara de papa común, cascara de vegetales, cascara de zanahoria y cascara de frutas. Las proporciones utilizadas para la preparación del sustrato del reactor son iguales debido a su igual proporción de generación en la fuente (Ver Tabla I).

La muestra de prueba fue preparada mezclando 400gr de la Fracción de Residuos Sólidos Total (FRST) y 1200 ml de agua de grifo, previamente reposada durante 2 horas para eliminar los efectos del cloro que pueden eliminar la población bacteriana. Para la inoculación de la materia orgánica se utilizó 20 gr. de Septitrim, el cual es un producto biológico basado en cultivo de bacterias no patogénicas especializadas, de uso común en pozos sépticos y sistemas de tratamiento de aguas anaeróbicas. El uso de Septitrim optimiza el funcionamiento garantizando una población bacteriana especializada y suficiente, capaz de degradar la carga orgánica que se deposita en el reactor anaeróbico. El sustrato obtenido es depositado en el reactor anaeróbico con capacidad de 3 Litros, y se dispuso de la mitad de su capacidad para almacenamiento de biogás generado en el proceso de fermentación. La relación de mezcla 1:3 entre el sustrato y agua, se toma según criterio dado por (K. Komemoto, 2009).

TABLA I. COMPOSICIÓN DE LA MATERIA ORGÁNICA A PARTIR DE DESECHOS DE COMIDA PROVENIENTES DE RESTAURANTE.

COMPONENTE	PORCENTAJE (FRST*-%)
Cascara de Frutas (Papaya, mandarina, banano, naranja)	25
Cascara de Vegetales (Pepino y Espinaca)	25
Cascara de Zanahoria	25
Cascara de papa	25

*FRST: FRACCIÓN DE RESIDUOS SÓLIDOS TOTAL

El pH de la muestra inicial depositada en el reactor anaeróbico fue medido con un pHmetro digital Checker by Hanna HI 98103, el cual registró un valor de 6.85. De acuerdo a lo descrito por (A. Guevara, 1996), este valor de pH se encuentra dentro del rango deseado (6.5–7.5), ya que si la muestra presenta un nivel de acidez alto (pH inferior a 6.5), esta es una de las causas de inhibición de las bacterias metanogénicas.

Para evaluar los efectos de la temperatura en la calidad del biogás, el experimento fue realizado durante un periodo de 40 días. El reactor fue sellado herméticamente y se mantuvo una agitación constante a 120 rpm una vez cada 24 horas.

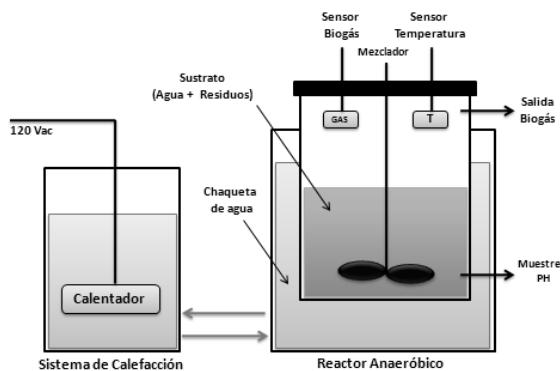


Figura 1. Diagrama de bloques del digestor anaeróbico.

A. Diseño de controlador difuso para temperatura

Los criterios de selección para optar por el diseño de un controlador difuso para la regulación de temperatura en el digestor anaeróbico fueron básicamente la relación no lineal de la temperatura con la actividad microbiana durante el proceso de digestión anaeróbica (J. Van Lier, 1995), y por otro lado, este método de control usa una caja negra (Black Box) enfocado a determinar la relación entre la variable manipulada (CV) y la variable del proceso (PV) sin incorporar ninguna información acerca de los detalles tanto físicos como químicos dentro del sistema anaeróbico (M. Huzmezan, 2002), que dada la complejidad de la composición de la materia orgánica y la variación misma de las reacciones dentro del digestor, la obtención de un modelo matemático se convierte en una tarea ardua y difícil de realizar.

La lógica difusa es más eficaz en sistemas de control en lazo cerrado y quizás más fácil de implementar en sistemas que van desde un microcontrolador embebido hasta una amplia red de computadoras con múltiples canales de adquisición de datos y sistemas de control (N. Kanagaraj, 2006).

La propuesta del sistema de control que permite mantener el punto de operación dentro del rango de temperaturas mesofílico (25°C – 45°C); por ser este el rango de temperatura seleccionado para evaluar los efectos de la temperatura en la calidad del biogás generado en proceso de digestión anaeróbico, se muestra en la Figura 2.

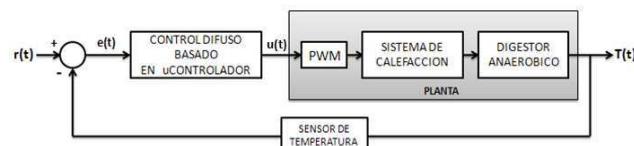


Figura 2. Control difuso basado en Microcontrolador.

El punto de operación inicial que se establece en el digestor anaeróbico es de 35°C . La variación del punto de operación dentro del rango de temperatura mesofílico, permitirá evaluar y registrar los efectos de la temperatura en la calidad del biogás generado en el reactor anaeróbico.



El sensor de temperatura utilizado es el DS18S20 de la Dallas Semiconductor, el cual es un termómetro digital que entrega la medida de la temperatura mediante un dato a 9 bits.

Para el monitoreo de las concentraciones volumétricas del gas Metano y el Dióxido de Carbono se utilizó el sensor de gas IR15TT-R de la empresa e2v, el cual posee dos canales para detección simultánea de CH₄ y CO₂ y cuenta con un rango de medición de 0 hasta 100%vol.

B. Control difuso con UNFuzzy

La señal de entrada al controlador difuso es la señal de error, correspondiente a la diferencia aritmética entre la señal de referencia y la señal de salida de temperatura, medida por medio del sensor DS18S20.

Las variables lingüísticas definidas para la señal de entrada **error** son:

- NG: Negativo Grande
- NP: Negativo Pequeño
- CERO: Cero
- PP: Positivo Pequeño
- PG: Positivo Grande

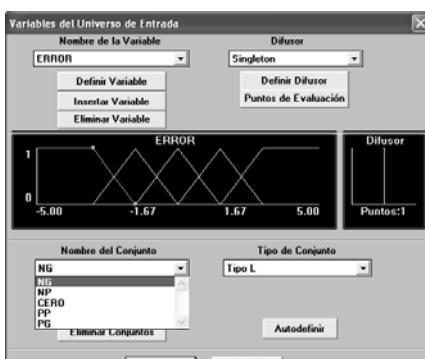


Figura 3. Definición de conjuntos difusos de la variable ERROR

Se cuenta con un calentador eléctrico manipulado mediante modulación de ancho de pulso PWM, el cual es controlado por la señal de control u(t) conformado por un dato digital de 8 bits suministrado por el bloque de control difuso basado en microcontrolador, y donde el dato digital d'0' corresponde a un suministro mínimo de voltaje RMS al elemento calefactor y el dato d'255' corresponde a un suministro máximo de voltaje RMS al elemento calefactor. Por tanto, el Universo de discurso para la variable **control** queda definida entre [0,255], dado que la resolución obtenida con un dato digital a 8 bits es de:

$$\text{Resolución} = 2^n - 1 = 2^8 - 1 = 255 \quad (1)$$

Donde **n** es el número de bits de la palabra de control.

Las variables lingüísticas definidas para la señal de salida control son:

- VOLTAJE NULO
- VOLTAJE BAJO
- VOLTAJE MEDIO
- VOLTAJE ALTO



Figura 4. Definición de los conjuntos difusos de la variable CONTROL.

El tipo de congresor (DeFuzzyfier) utilizado es el de tipo centro de gravedad, por ser este uno de los métodos más populares para calcular el valor concreto de salida en un Sistema de Lógica Difusa.

$$\text{defuzz} = \frac{\sum y_{mf} x_{mf}}{\sum y_{mf}} \quad (2)$$

Donde x_{mf} es el vector de valores de la función de membresía de entrada y y_{mf} es el vector de valores de la función de membresía del vector x_{mf} .

La base de reglas para el sistema de lógica difusa se definió considerando que el sistema de calefacción a través del sistema de recirculación constante hacia la chaqueta de agua que rodea el digestor, suministre la temperatura adecuada para mantener el error en estado estacionario igual a cero; es decir, que la temperatura dentro del reactor sea igual al valor establecido por el valor de referencia, el cual es establecido inicialmente en 35°C.

TABLA II. BASE DE REGLAS PARA CONTROLADOR DIFUSO

ERROR	NG	NP	CERO	PP	PG
CONTROL	VOLTAJE NULO	VOLTAJE NULO	VOLTAJE BAJO	VOLTAJE MEDIO	VOLTAJE ALTO

La función de transferencia del sistema de control difuso permite visualizar la relación existente entre la entrada (error) y la salida (control).

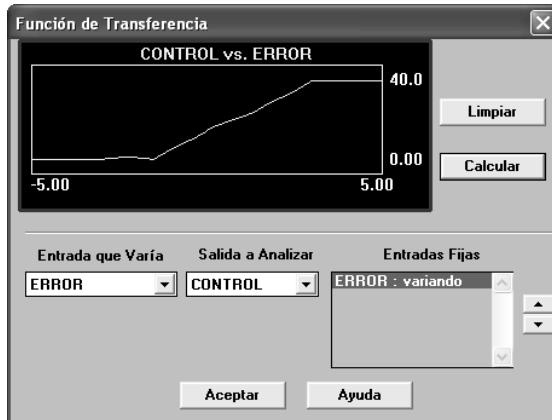


Figura 5. Superficie de control del sistema difuso

La figura 5 muestra una superficie de control con una relación proporcional positiva para un rango de la señal de error entre [-3,3], lo cual lo hace adecuado para su implementación. Por otro lado, la superficie de control no presenta cambios bruscos para ningún valor del Universo de Discurso de la señal de entrada.

C. Implementación del controlador difuso

Un microcontrolador de 8 bits PIC18F452 de Microchip fue seleccionado como procesador embebido, para implementar el algoritmo de control difuso para la regulación de temperatura en el digestor anaeróbico.

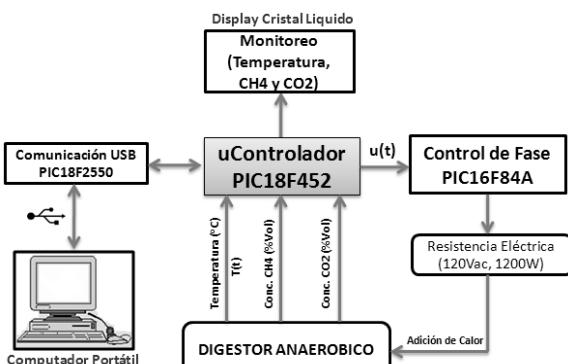


Figura 6. Diagrama de bloques del sistema de control de temperatura basado en microcontrolador.

El algoritmo de control difuso y manejo de periféricos del sistema en general fue escrito en Lenguaje C, utilizando el software de desarrollo CCS PIC C Compiler.

Para el bloque fuzzificador se utilizó funciones de pertenencia de tipo triangular y trapezoidal y el universo de discurso esta comprendido entre [-5.0 5.0] según se mostró en la Figura 3.

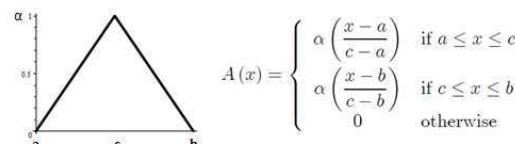


Figura 7. Definición de funciones de pertenencia de entrada.

El código fuente en lenguaje C para la definición de estas funciones de pertenencia se muestra en la siguiente figura:

```
-----  

// Funciones de pertenencia de la Entrada  

-----  

void trian(float x)  

{  

    if (x<=a) y=0;  

    if (x<=b&&x>a) y= (x-a)/(b-a);  

    if (x>c&&x<b)y =(c-x)/(c-b);  

    if (x>=c) y=0;  

}  

void trapmfiz(float x)  

{  

    if (x<=a) y=1;  

    if (x<=b&&x>a) y= (b-x)/(b-a);  

    if (x>b) y=0;  

}  

void trapmfder(float x)  

{  

    if (x<=a) y=0;  

    if (x<=b&&x>a) y= (x-a)/(b-a);  

    if (x>b) y=1;  

}
```

Figura 8. Código fuente para funciones de pertenencia triangular y trapezoidal.

Por tanto, la definición de los conjuntos difusos de entrada está definida como lo muestra la figura 9.

```
-----  

//Definicion Conjuntos Difusos de Entrada  

-----  

// Conjunto Borroso NG  

//  

a=-3.0;b=-1.5; trapmfiz(x); NG= y;  

//  

//Conjunto Borroso NP  

//  

a=-3.0;b=-1.5;c=0.0; trian(x); NP= y;  

//  

//Conjunto Borroso CERO  

//  

a=-1.5;b=0.0;c=1.5; trian(x); CERO= y;  

//  

// Conjunto Borroso PP  

//  

a=0.0;b=1.5;c=3.0; trian(x); PP= y;  

//  

// Conjunto Borroso PG  

//  

a=1.5;b=3.0; trapmfder(x); PG=y;
```

Figura 9. Código fuente para definición de conjuntos difusos de entrada.

Las funciones de pertenencia de salida están definidas bajo el mismo fundamento matemático de las funciones de pertenencia de entrada.

```
-----  
// Funciones de Pertenencia de Salida  
-----  
void desbtrapder(float y)  
{  
    datos[num]=y;  
    W[num]=(y*(b-a))+a;  
}  
void desbtrapiz(float y)  
{  
    datos[num]=y;  
    W[num]=(y*(-b+a))+b;  
}  
void desbtriang (float y)  
{  
    datos[num]=y;  
    W[num]=(y*(b-a))+a;  
    num++;  
    datos[num]=y;  
    W[num]=(y*(-c+b))+c;  
}
```

Figura 10. Código fuente para definición de conjuntos difusos de salida

De acuerdo a la definición de la base de reglas mostrada en la tabla II, el código fuente correspondiente al motor de inferencia se muestra en la Figura 11.

```
-----  
// Base de Reglas del controlador para una sola Entrada  
-----  
  
for (num=0;num<8;num++)  
{  
    datos[num]=0;  
    W[num]=0;  
}  
  
num=0;  
  
if (NG>0) //APAGAR  
{  
a=28.0;b=96.0;y=NG;desbtrapiz (y);  
}  
if (NP>0) //APAGAR  
{  
a=28.0;b=96.0;y=NP;num++;desbtrapiz (y);  
}  
if (CERO>0) //MANTENER  
{  
a=48.0;b=96.0;c=144.0;y=CERO;num++;desbtriang (y);  
}  
if (PP>0) //MEDIO ENCENDER  
{  
a=96.0;b=144.0;c=192.0;y=PP;num++;desbtriang (y);  
}  
if (PG>0) //ENCENDER  
{  
a=144.0;b=220.0;y=PG;num++;desbtrapder (y);  
}
```

Figura 11. Código fuente para Base de reglas.

El código fuente correspondiente al método de desfuzzificación se muestra en la Figura 12.

```
-----  
// Metodo de Desborrification  
-----  
  
// Creating Custom Defuzzification Functions  
// Fuzzy Logic Toolbox User Guide page 108  
  
Mx=0;  
Vx=0;  
  
for (num=0; num<8; num++)  
{  
Mx=Mx+datos[num];  
Vx=Vx+datos[num]*W[num];  
}  
  
Salida= Vx/Mx;  
}
```

Figura 12. Código fuente para Desfuzzificación.

D. Respuesta del controlador difuso

El desempeño del sistema de control difuso fue probado experimentalmente para regular la temperatura en condiciones mesofílicas. Se estabilizó inicialmente el sistema a 35°C, y posteriormente se realizaron cambios de tipo escalón unitario

en el set point a valores de 40°C y 45°C para evaluar el comportamiento del controlador difuso basado en microcontrolador.

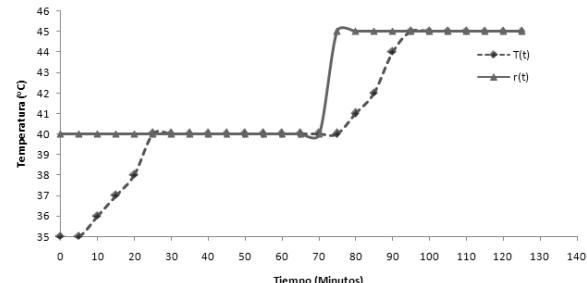


Figura 13. Respuesta del sistema de control difuso basado en microcontrolador.

Según los resultados, los parámetros de desempeño del controlador difuso basado en microcontrolador son:

- Máximo sobre pico: 0%
- Tiempo de estabilización: 25 min.
- Error en estado estacionario: 0%
- Tiempo muerto: 2.5 min.

Según lo mencionado por (K. Komemoto, 2009), el tiempo estimado para el proceso de digestión anaeróbica es de 30 días aproximadamente dado su capacidad volumétrica y características del sustrato. Bajo esta condición se puede determinar que el sistema de control es adecuado para las pruebas que se desean efectuar para determinar el efecto de la temperatura en la calidad del biogás generado dentro del digestor. Es importante resaltar el efecto positivo del proceso de agitación, el cual se realizó a 120 rpm durante 2 horas con una frecuencia de 2 veces cada 24 horas, dado que dicho parámetro permitió que la estabilización de la temperatura fuese más rápido con respecto a la ausencia de la agitación.

La calidad del biogás fue medida en términos del porcentaje volumétrico de la Concentración del Metano (CH_4) y Dióxido de Carbono (CO_2), puesto que la calidad del biogás depende principalmente de la presencia de estos gases (T. Mandal, 1999; P. Vindis, 2008). La frecuencia de muestreo para el registro del valor de la Concentración tanto del CH_4 como del CO_2 fue 2 muestras/día (muestreo cada 12 horas).

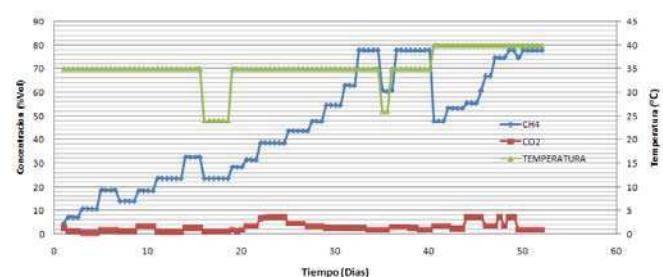


Figura 14. Efectos de la temperatura en la calidad del biogás.



De acuerdo a lo referenciado por el Instituto para la Diversificación y Ahorro de Energía del Ministerio de Industria, Turismo y Comercio del Gobierno de España en su revista “Biomasa - Digestores Anaerobios” (IDAE, 2007), la composición del producto gaseoso resultado de la digestión anaerobia puede estar comprendida entre 50-70% de CH₄, 30-40% de CO₂ y ≤ 5% de H₂, H₂S y otros gases. Por tanto, se considera que la concentración de CH₄ resultado de la digestión anaeróbica de la Fracción de Residuos Sólidos Orgánicos utilizado para el experimento se encuentra dentro del rango óptimo de producción. Por otro lado, se observa que el registro de Concentración de CO₂ presenta inconvenientes dado que no es un valor razonable dentro de los parámetros estándar de producción de CO₂ en el proceso de digestión anaeróbica.

III. ANALISIS DE RESULTADOS

El sistema de control difuso basado en microcontrolador para la regulación de la temperatura en el digestor anaeróbico resultó apropiado para los propósitos requeridos en el experimento de digestión anaeróbica, puesto que siempre mantuvo la temperatura en el punto de operación establecido con un error en estado estacionario de 0% y con un tiempo de estabilización de 20 minutos. El punto de operación fue inicialmente de 35°C y una vez estabilizada las concentraciones de Metano y Dióxido de Carbono en el proceso de fermentación, se estableció el punto de operación en 40°C para poder analizar los efectos de la temperatura en el rango mesofílico en los niveles de concentración de estos gases en el biogás generado en el reactor anaeróbico.

Este resultado es similar a lo mostrado por (P. Vindis, 2008) en su trabajo “Biogas production with the use of mini digester”.

IV. CONCLUSIONES

El diseño e implementación del controlador difuso basado en microcontrolador PIC18F452 de la familia microchip muestra una gran capacidad en la velocidad de respuesta, así como un bajo porcentaje de utilización de los recursos de memoria ROM para almacenamiento de programa (50%) y memoria RAM para datos (20%). Teniendo en cuenta que el algoritmo involucra procesamiento de datos de tipo flotante (float) y la arquitectura de este microcontrolador no posee hardware especializado para el manejo de datos de tipo flotante, se muestra la eficiencia y viabilidad de implementar sistemas embebidos a partir de este tipo de dispositivos programables para la obtención de controladores en tiempo real, que indudablemente mejora el desempeño del sistema controlado y disminuye los costos de implementación del sistema de control.

La calidad del biogás generado en el proceso de digestión anaeróbica a partir de una fracción de residuos sólidos

orgánicos provenientes de restaurante fue bueno, considerando que una concentración de 77%Vol de gas Metano es una concentración que para 1m³ de biogás puede llegar a contener una capacidad calorífica equivalente a la proporcionada por 0.8L de gasolina (IDEA, 2007). Algunos de los parámetros que se consideraron importantes durante el proceso para la obtención de este resultado fueron principalmente la variable de temperatura, la cual gracias al sistema de control difuso implementado siempre se mantuvo en el valor establecido, el valor de pH que siempre estuvo dentro del rango de operación recomendado de 6.5-7.5, el proceso de agitación con frecuencia de 2veces/día con duración de 2 horas cada una, puesto que este proceso permitió mantener una distribución de temperatura uniforme dentro del reactor, y una consistencia homogénea del sustrato, y la inoculación del sustrato mediante una población bacteriana especializada llamada Septitrim para aumentar el proceso de degradación de la fracción orgánica de residuos sólidos.

REFERENCIAS.

- [1] DUARTE, Oscar G. UNFUZZY: Fuzzy Logic System analysis, design, simulation and implementation software. 1998. Disponible en internet: http://www.eusflat.org/publications/proceedings/EUSFLAT-ESTYLF_1999/papers/251-duarte-2.pdf
- [2] Instituto para la Diversificación y Ahorro de la Energía IDAE. Biomasa: Digestores Anaerobios. Ministerio de Industria, Turismo y Comercio. Gobierno de España. ISBN: 978-84-96680-21-0. Madrid Octubre de 2007.
- [3] FERNÁNDEZ, Alejandro y SANCHEZ, Mayra. Guía para la Gestión Integral de los Residuos Sólidos Urbanos. Dirección Provincial de Servicios Comunales de la Ciudad de La Habana. Organización de las Naciones Unidas para el Desarrollo Industrial. 2007. Disponible en internet: http://www.unido.org/fileadmin/import/72852_Gua_Gestin_Integral_de_RSU.pdf.
- [4] FORSTER, T. The Effect of Temperature variations on the Performance of Mesophilic and Thermophilic Anaerobic Filter Treating a Simulated Papermill Wastewater. Process Biochemistry, Elsevier Science Ltda 2002.
- [5] GUEVARA, Antonio. Fundamentos Básicos para el Diseño de Biodigestores Anaeróbicos Rurales: Producción de Gas y Saneamiento de Efluentes. Centro Panamericano de Ingeniería Sanitaria y Ciencias del Ambiente. Lima. 1996. Disponible en internet: <http://www.bvsde.paho.org/bvsacd/scan2/031042/031042.pdf>.
- [6] HUZMEZAN, Mihai. Advanced Control of Batch Reactor Temperature. Proceedings of the American Conference. IEEE Xplore. 2002.
- [7] KANAGARAJ, N. An Embedded Fuzzy Controller for Real Time Pressure Control. IEEE Xplore. 2006.
- [8] KOMEMOTO, K. Effect of temperature on VFA's and biogas production in anaerobic solubilization of food waste, published by Elsevier Ltd, 2009.
- [9] LINDORFER, H. New data on temperature optimum and temperature changes in energy crop digesters, published by Elsevier Ltd, 2008.
- [10] MANDAL, T. Determination of the Quality of Biogas by Flame Temperature Measurement. Energy Conversion & Management, Elsevier Science Ltda. 1999.
- [11] NGUYEN, Hung T. A First Course in Fuzzy and Neural Control. Editorial Chapman & Hall/CRC. Chapter 3: Fuzzy Logic for Control. Pág. 87-132. 2003.
- [12] VAN LIER, Jules. Effect of Temperature on the Anaerobic Thermophilic conversion of Volatile Fatty Acids by Dispersed and Granular Sludge. Elsevier Science Ltda, 1995.
- [13] VINDIS, P. Biogas Production with the use of mini digester. Journal of Achievements in Materials and Manufacturing Engineering. Volume 28.



Osciloscopio digital basado en FPGA con conexión Ethernet e interfaz virtual para monitoreo de sistemas eólicos

Leonardo Flaviani

Universidad Tecnológica Nacional
Facultad Regional Mendoza
Email: flaviani1@yahoo.com.ar

Esteban Rivas

Universidad Tecnológica Nacional
Facultad Regional Mendoza
Email: estebanrrivas@gmail.com

Rodrigo González

Laboratorio de Computación Reconfigurable
Universidad Tecnológica Nacional
Facultad Regional Mendoza
Email: rodralez@frm.utn.edu.ar

Resumen—El presente trabajo describe el diseño y desarrollo de un osciloscopio digital con conexión Ethernet, con funciones de visualización y control a través de una computadora. El osciloscopio está constituido por un canal de entrada con ancho de banda de 300 KHz, una etapa analógica adaptadora de señal, una etapa de conversión analógico digital, una unidad de control y transmisión, la cual es un sistema en un chip basado en FPGA (field programmable gate array), y una memoria de almacenamiento. Las muestras tomadas de la señal de interés son enviadas mediante protocolo TCP-UDP/IP sobre Ethernet a una computadora para su visualización. Un instrumento virtual desarrollado en LabView permite una interacción amigable e intuitiva con el usuario ya que recrea el frente de un osciloscopio real. Se pueden realizar distintos tipos de análisis sobre las señales adquiridas como así también controlar y configurar las variables de funcionamiento del osciloscopio (nivel de disparo, modo de disparo, etc.). Se calibró el instrumento respecto a un osciloscopio HITACHI VC-6024 y a un contador de frecuencia Hewlett Packard 5325B. Se verificó que el osciloscopio desarrollado es de la misma clase que los instrumentos mencionados.

I. INTRODUCCIÓN

El instrumento desarrollado fue solicitado por la empresa IMPSA WIND en el 2010 a partir de la necesidad de efectuar mediciones de distintas señales provenientes de la góndola de un generador eólico, desde el pie de la torre del mismo (Fig. 1) en pleno funcionamiento. La normativa prohíbe por razones de seguridad que un operario realice mediciones en la góndola mientras el generador eólico está en funcionamiento.

El instrumento de medición debía ser voltmétrico de propósito general y de bajo costo, capaz de medir en el tiempo señales eléctricas de tensión con amplitudes de hasta 250 voltios y componentes de frecuencias en el rango industrial. Además, debía disponer de conexión Ethernet para efectuar mediciones desde el pie de la torre del generador y así utilizar el cableado Ethernet que contiene la misma. El software para la medición de señales y control del instrumento debía realizarse con el entorno de programación LabVIEW, por ser utilizado en los proyectos de la empresa solicitante.

Previo a su desarrollo, se hizo una búsqueda de dispositivos con estas características. La mayoría de los osciloscopios digitales solo tenían interfaz RS232 o USB, con protocolo de comunicación propietario. Si bien existían osciloscopios con

interfaz Ethernet, el costo de estos por lo general rondaba entre 10 y 15 veces el costo de nuestro equipo.

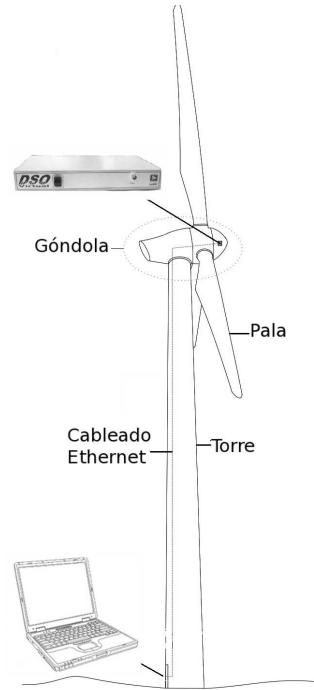


Figura 1. Ubicación del osciloscopio en el generador eólico.

II. DESCRIPCIÓN GENERAL

En la presente sección se detallan las partes que integran el osciloscopio digital. La Fig. 2 muestra un esquema del instrumento.

II-A. Atenuador calibrado y compensado

El atenuador está compuesto por una cadena de atenuación resistiva compensada en frecuencia, con atenuaciones programables de $\times 1$, $\times 10$ y $\times 100$. Cuenta además con un circuito de protección contra sobre tensiones, una etapa de adaptación de impedancia y una etapa sumadora. Esta última

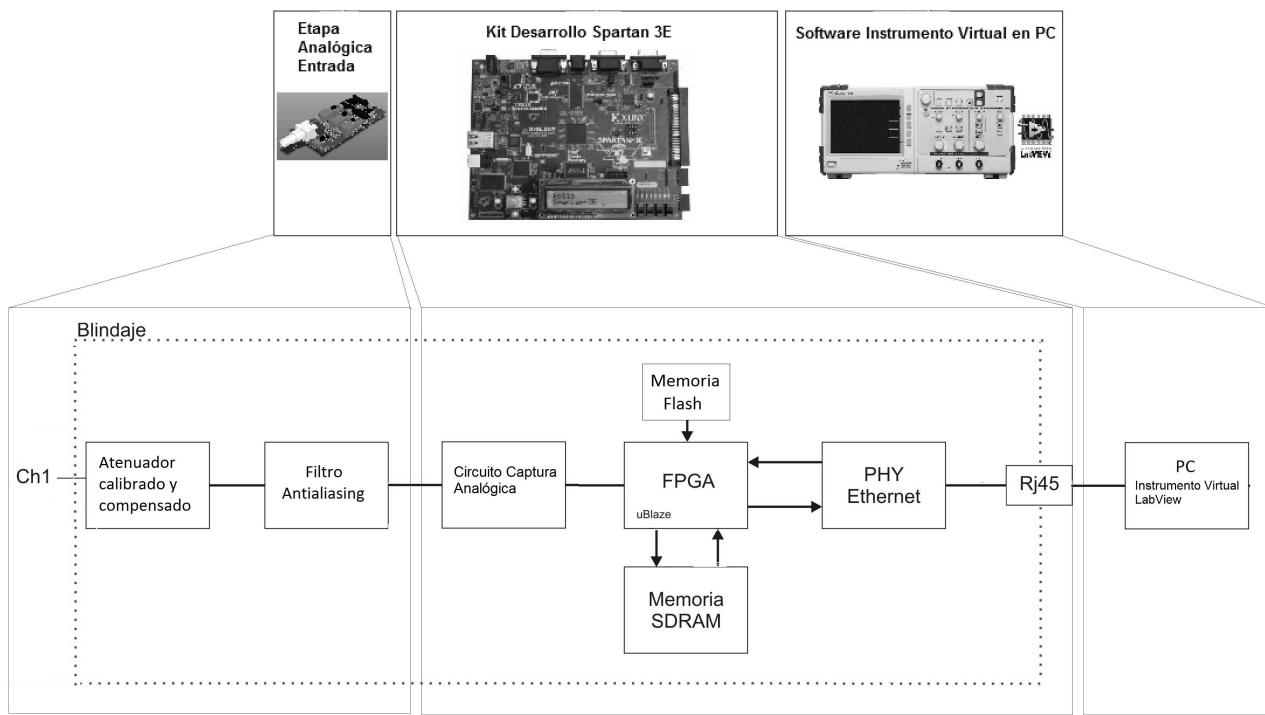


Figura 2. Diagrama del sistema.

es la encargada de agregar un nivel de offset de 1,25 voltios necesarios para atacar el conversor A/D. La selección del nivel de atenuación y del intercalado de un capacitor de desacople se lleva a cabo digitalmente mediante optoacopladores que son comandados por el controlador lógico (sec. II-D) desde los pines de propósito general del FPGA.

II-B. Filtro antialiasing

El filtro antialiasing limita la banda de frecuencia de la señal de entrada. Compuesto por un filtro activo pasa bajos tipo Butterworth de orden 8 [1]. Fue diseñado para un ancho de banda útil de 300 KHz, una frecuencia de Nyquist de 750 KHz y para una resolución del conversor A/D de 8 bits.

II-C. Conversor A/D

En esta etapa se utilizan un conversor analógico/digital LTC1407-1 y un amplificador de ganancia programable LTC6912, ambos de Linear Technology, incluidos en el kit de desarrollo utilizado [4] (sec. II-D).

El conversor posee de 2 canales, ambos con resolución máxima de 14 bits y frecuencia de muestreo máxima de 1,5 mega muestras por segundo. Se utilizó solo 1 canal con 8 bits de resolución. La elección de esta resolución se justifica en que el máximo nivel de señal de entrada permitido al conversor A/D es de 2,5 voltios. Por tanto, con 8 bits se obtiene una resolución de 10 mV por bit, suficiente para las especificaciones de diseño del osciloscopio. Se fijó la frecuencia de muestreo en 750 kilo muestras por segundo.

A la entrada de cada canal del conversor hay un amplificador de ganancia programable LTC6912. Su propósito es escalar el voltaje de entrada al conversor para maximizar el rango de conversión. Los amplificadores y el conversor poseen interfaz SPI para su programación. El controlador lógico (sec. II-D) es el encargado de configurar esta etapa, fijando el nivel de amplificación y la frecuencia de muestreo deseada por el usuario.

II-D. Arquitectura del sistema digital

En el corazón del sistema embebido del osciloscopio se encuentra una placa Spartan 3E Starter Board de Digilent [4] con un FPGA Xilinx Spartan-3E XC3S500E4CFG320. Dos motivos justifican el uso de tecnología reconfigurable en lugar de una solución basada en microprocesador. En primer término, 750 kilo muestras de 8 bit por segundo (sec. II-C) representan 750 KB de información por segundo. Esta elevada cantidad de información puede ser difícil de procesar en forma secuencial por un microprocesador de bajo costo, el cual además también debe atender el control y la comunicación del instrumento. Por otro lado, una solución basada en FPGA ofrece flexibilidad y mejor escalabilidad. Así, el sistema se podrá adecuar fácilmente a futuras necesidades, como por ejemplo, agregar otro canal de muestreo.

La placa en referencia posee una variedad de periféricos. Los utilizadas en este desarrollo fueron:

- Memoria DDR SDRAM de 64 MByte (512 Mbit).
- Memoria Flash NOR de 16 MByte (128 Mbit).
- Puerto Ethernet de 10/100 Mbps.



- Puerto serie RS-232, para depuración del sistema.
- Oscilador de 50 MHz.
- Conversor analógico digital y preamplificador programable (sec.II-C).
- Puerto USB para grabación del FPGA.
- Conectores de expansión de propósito general, para manejo de la etapa atenuadora (sec. II-A).

En el FPGA se implementa un sistema en un chip (SoC) basado en el soft-procesador MicroBlaze [5], el cual es provisto por el fabricante del FPGA. MicroBlaze es un microprocesador de 32 bits parametrizable. Cuenta con una gran variedad de periféricos que ejecutan funciones específicas. Los periféricos utilizados y conectados a MicroBlaze en este trabajo fueron los siguientes:

- Controlador Ethernet: diseñado para incorporar las características descriptas en el estándar IEEE Std. 802.3 Media Independent Interface (MII). Provee comunicación Ethernet a 10 Mbps y 100 Mbps con una mínima utilización de los recursos del FPGA.
- Timer: módulo con 2 timers de 32 bits con interrupciones y generación de eventos, un contador configurable y una salida de PWM.
- Controlador de interrupciones: concatena múltiples entradas de interrupciones de distintos periféricos en una sola línea de interrupción. Módulo es necesario dado que MicroBlaze posee solo una entrada de interrupción.
- Controlador RS-232: interfaz para las comunicaciones de datos asíncronas.
- Controlador lógico: desarrollado en lenguaje VHDL para el proyecto. Controla las funciones básicas del osciloscopio a través de la integración de cuatro módulos: 1) el módulo de captura que configura la ganancia y la frecuencia de muestreo y controla el preamplificador y el conversor A/D; 2) el módulo de disparo que recibe las muestras codificadas y en función del modo de disparo (recurrente, disparado o pre-disparo) y del nivel y flanco de disparo, controla el almacenamiento de las muestras en la memoria FIFO; 3) el módulo memoria FIFO constituida por 1024 palabras de 8 bits y 4) el módulo comunicación con Microblaze, el cual lee la memoria FIFO y entrega la información al microprocesador para ser transmitida a la computadora vía UDP/IP. El controlador lógico se comunica con Microblaze a través del bus punto a punto Fast Simple Link (FSL). Los parámetros para configurar el controlador lógico son definidos por el usuario desde la interfaz gráfica en la computadora, y son recibidos por Microblaze vía TCP/IP.

Justifica el uso de una memoria de adquisición FIFO el hecho de que la relativa baja velocidad de transferencia de datos disponibles a través de los canales de entrada/salida de una computadora (Ethernet, USB, etc.) comparada con la velocidad de muestreo del conversor A/D, hace imposible transferir datos directamente a una computadora. Además, el sistema operativo de una computadora de propósito general suele no ser de tiempo real, lo que empeora esta situación.

Para resolver este problema el bloque del controlador lógico posee una memoria interna tipo FIFO que almacena los datos capturados para luego ser transferidos secuencialmente a la computadora. La capacidad de la misma se define como una solución de compromiso entre el tiempo de captura, el tiempo de transmisión, la frecuencia de refresco de visualización y la cantidad de muestras a visualizar en pantalla. Se fijó su tamaño en 1024 palabras de 8 bits.

Los recursos utilizados en el FPGA fueron:

- Número de *slices*: 3,880 de 4,656 (83 %).
- Número de RAMB16: 12 de 20 (60 %).
- Número de DCM: 2 de 4 (50 %).
- Número de BlockDSP: 3 de 20 (15 %).

II-E. Software

La herramienta de desarrollo utilizada para MicroBlaze es el Embedded Development Kit (EDK). Se usó la versión 12.4. EDK está compuesta por una serie de programas. El Xilinx Platform Studio (XPS) se utiliza para el desarrollo del hardware, y el Software Development Kit (SDK) para el desarrollo del software embebido para MicroBlaze. La programación del microprocesador se realiza en una versión reducida del lenguaje C, propiedad de Xilinx. SDK contempla la posibilidad de utilizar los drivers de los periféricos especificados en XPS y de dotar al microprocesador de un sistema operativo. El SO utilizado es el Standalone de Xilinx, monohilos, el cual es una plataforma de software simple y de bajo nivel que provee acceso directo a los recursos elementales del procesador. Para el uso de los protocolos TCP-UDP/IP se utilizó la biblioteca lwIP [3].

II-F. Instrumento virtual

LabVIEW [6] es un entorno de programación gráfico usado para desarrollar sistemas de medición, prueba y control. El lenguaje posee íconos y cables, su programación se asemeja a un diagrama de flujo. Los programas desarrollados con LabVIEW se llaman instrumentos virtuales (IV) [2].

La interfaz gráfica diseñada es muy similar a la de un osciloscopio digital convencional, a fin de que la interacción con el usuario fuera intuitiva y cercana a la de un instrumento real. La Fig. 3 muestra la interfaz del IV.

El IV es el encargado de la visualización en pantalla de las muestras tomadas en la góndola del generador eólico (Fig. 1) y de la configuración del osciloscopio digital. El IV configura el nivel de disparo, la pendiente de disparo y los modos de disparo. Los modos de disparo que soporta el osciloscopio son: inmediato, normal, automático, único y pre-trigger.

Las muestras son enviadas desde el sistema embebido vía protocolo UDP/IP sobre Ethernet. El instrumento virtual las toma y las procesa para ser mostradas por pantalla. El instrumento virtual también es el encargado de enviar los parámetros de configuración al sistema embebido vía protocolo TCP/IP sobre Ethernet, para garantizar su entrega.

Muestras	200mV/Div			500mV/Div			5V/Div		
	HIT [V]	Osc. [V]	Error [%]	HIT [V]	Osc. [V]	Error [%]	HIT [V]	Osc. [V]	Error [%]
1	0,384	0,390	-1,56	0,60	0,60	0,00	5,00	4,90	2,00
2	0,510	0,504	1,18	0,80	0,79	1,25	6,00	6,00	0,00
3	0,600	0,603	-0,50	1,00	1,01	-1,00	7,40	7,40	0,00
4	0,720	0,710	1,39	1,20	1,19	0,83	9,80	10,00	-2,04
5	0,800	0,799	0,13	1,40	1,39	0,71	12,60	12,30	2,38
6	0,899	0,900	-0,11	1,60	1,59	0,63	14,10	14,00	0,71
7	1,000	0,995	0,50	1,80	1,81	-0,56	16,60	16,50	0,60
8	1,100	1,099	0,09	2,00	1,99	0,50	18,00	17,90	0,56
9	—	—	—	2,20	2,19	0,45	19,80	19,60	1,01
10	—	—	—	2,40	2,40	0,00	22,00	22,10	-0,45

Tabla I
VALORES DE CALIBRACIÓN RESPECTO AL OSCILOSCOPIO HITACHI.

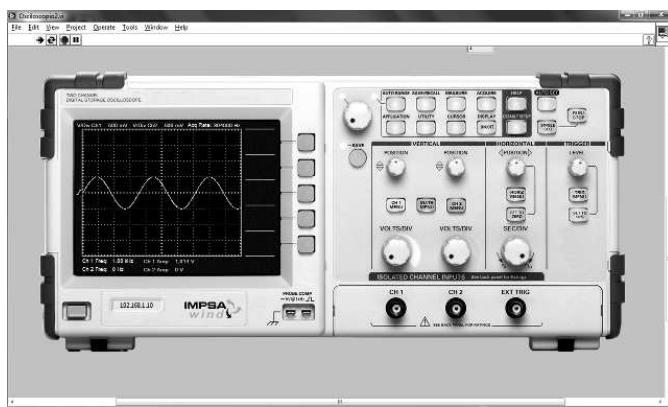


Figura 3. Interfaz gráfica del instrumento virtual en LabVIEW.

III. AJUSTE

El ajuste se realizó según la norma ISO 9001:2008, apartado 7.6. Se utilizó como instrumento patrón un osciloscopio de almacenamiento digital marca HITACHI VC-6024. La calibración fue hecha en amplitud de tensión y frecuencia. El procedimiento de ajuste consistió en modificar los factores de escalas de los distintos parámetros y rangos en el instrumento virtual, de tal forma que para una medición dada ambos instrumentos dieran prácticamente el resultado.

IV. CALIBRACIÓN

La norma ISO 9001:2008, apartado 7.6, establece que la calibración de un instrumento consiste en comparar las lecturas realizadas en él con las obtenidas de un instrumento de mejor jerarquía, denominado patrón. Se efectúa para definir la clase del instrumento bajo análisis.

Para la contrastación del osciloscopio digital se utilizaron como instrumentos patrones un osciloscopio de almacenamiento digital marca HITACHI VC-6024, utilizado también para su ajuste (sec. III), y un contador de frecuencia marca Hewlett Packard 5325B. El osciloscopio HITACHI tiene una precisión $\pm 3\%$ desde 100 mV/div a 10 V/div [7]. El contador por su método de medición de periodo tiene un error de ± 1 cuenta \pm exactitud base tiempo \pm error de disparo.

Ambos osciloscopios, patrón (HIT) y bajo análisis (Osc.), fueron conectados en paralelo a una fuente de señal senoidal.

Se fijaron 3 rangos para realizar las mediciones: 200 mV/div, 500mV/div y 5V/div. Se tomaron entre 8 y 10 muestras para cada rango. La tabla I muestra los valores registrados. Se observó que las mediciones de amplitudes de tensión de nuestro osciloscopio para los rangos seleccionados caen dentro del margen de error del instrumento patrón, $\pm 3\%$. Se concluye que se puede considerar a nuestro osciloscopio digital de la misma clase que el patrón.

En la calibración de medición de frecuencia observamos que todas las mediciones del osciloscopio digital presentan una desviación inferior al 1% en todo el rango de la medición respecto del instrumento patrón. Por tanto se considera que nuestro osciloscopio tiene la misma clase que el contador de frecuencia Hewlett Packard 5325B.

V. CONCLUSIONES

Este trabajo muestra el diseño y desarrollo de un osciloscopio digital con conexión Ethernet e interfaz virtual en LabVIEW. Se alcanzaron las especificaciones fijadas por la empresa solicitante. Se contrastó el instrumento con un osciloscopio y un contador de frecuencia de marcas reconocidas. Se verificó que nuestro osciloscopio es de la misma calidad que los instrumentos patrones.

AGRADECIMIENTOS

Agradecemos en primer lugar a nuestras familias por su apoyo incondicional. También queremos agradecer la dirección de los ingenieros Pedro Pérez y Gustavo Mercado, ambos de la UTN-FRM, y Oscar Ojeda de IMPSA Wind.

REFERENCIAS

- [1] Ashok Ambardar. *Procesamiento de señales analógicas y digitales*. Michigan Technological University, 2002.
- [2] National Instruments Corporation. *Getting Started with LabVIEW*. National Instruments Corporation, April 2003.
- [3] Adam Dunkels. lwIP, light-weight implementation of the TCP/IP protocol suite. Website: <http://www.sics.se/adam/lwip/>.
- [4] Digilent Inc. Spartan-3e Starter Board FPGA Development Board. UG230 (v1.0) march 9, 2006.
- [5] Xilinx Inc. Getting Started with the Microblaze Development Kit, Spartan-3E 1600E Edition. UG258 (v1.3). November 5. 2007.
- [6] National Instruments. LabVIEW System Design Software. Website: <http://www.ni.com/labview/>.
- [7] Hitachi Denshi Ltd. *Digital Storage Oscilloscope Model VC-6024. Operation Manual*.

Diseño y fabricación del sistema de interfaz y control de contadores de muones

D. Alejandro Almela^{1,2*}, Damián E. Alonso^{1,2**}, Alan E. Fuster^{1,2}, Matias R. Hampel^{1,2}, Adrián P. Sedoski^{1,3}
Federico Suarez¹, Oscar Wainberg^{1,2}

*alejandro.almela@iteda.cnea.gov.ar; **damian.alonso@iteda.cnea.gov.ar

¹Instituto de Tecnologías en Detección y Astropartículas (CNEA, CONICET, UNSAM)

²Universidad Tecnológica Nacional - Facultad Regional Buenos Aires

³Facultad de Ciencias Exactas y Naturales - Universidad de Buenos Aires

Resumen—El presente trabajo se focaliza en el diseño de un sistema embebido para proveer de interfaz y control a los módulos del contador de muones del proyecto internacional AMIGA (*Auger Muons and Infill for the Ground Array*) del Observatorio Pierre Auger, incluyendo la fabricación de prototipos e implementación de algunas aplicaciones. El desarrollo y verificación del desempeño es realizado por ITEDA (Instituto de Tecnologías en Detección y Astropartículas). El diseño brinda al contador alta eficiencia en la transferencia de los datos, continuo funcionamiento, control y monitoreo de las variables del entorno. Además, la capacidad de realizar operaciones comandadas a distancia, con fines de calibración y diagnóstico. Las características más importantes son su bajo consumo y alta confiabilidad.

Index Terms—Observatorio Pierre Auger, AMIGA, detección remota, single board computer, sistemas de adquisición, sistemas embebidos, *μCLinux*.

I. INTRODUCCIÓN

El Observatorio Pierre Auger [1], ubicado en Malargüe, Mendoza, Argentina. Originalmente ha sido concebido para estudiar los rayos cósmicos de energías mayores a los 10^{18} eV. Los rayos cósmicos son partículas que arriban desde el espacio exterior a las capas superiores de la atmósfera a partir de donde generan lluvias (o chubascos) de partículas secundarias que llegan hasta la superficie de la tierra. Estas lluvias tienen tres componentes fundamentales: la electromagnética (electrones y gamas), la hadrónica y la muónica (muones).

El Observatorio Pierre Auger consta de 27 telescopios de fluorescencia y 1666 detectores de radiación Cherenkov para el detector de superficie, instalados sobre una superficie de 3000 km^2 . Actualmente se está incorporando un tercer tipo de detector: los contadores de muones. Dichos contadores se incorporan en el marco de AMIGA [2] cuyos objetivos principales son mejorar el estudio de composición de los rayos cósmicos por medio del conocimiento de la componente muónica de las lluvias medidas por el Observatorio Pierre Auger (incorporando contadores de muones); y disminuir el umbral de energía hasta 10^{17} eV (mediante un *infill* o relleno de estaciones del detector de superficie) permitiendo el estudio de los rayos cósmicos en la zona del espectro donde se espera un cambio de fuentes galácticas a extragalácticas. En la figura 1 se muestra un esquema de las estaciones del Pierre

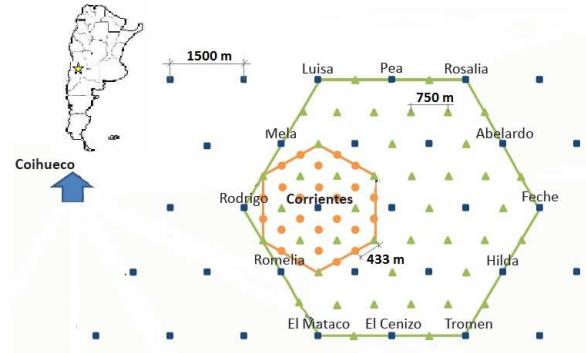


Figura 1. Mapa de ubicación del “Infill” de AMIGA, los puntos representan estaciones del detector de superficie. Cuadrado: arreglo principal, separados a 1500 m entre sí. Triángulo: “Infill” separados a 750 m entre si. Círculo: “Infill” separados a 433 m.

Auger donde se instalarán los contadores de AMIGA. En la figura 2 se ilustra el sistema del contador, que está formado por la electrónica de superficie, que sincroniza el contador con una estación del detector de superficie y realizará las comunicaciones con la central de datos del Pierre Auger mediante una red WI-FI, y la electrónica enterrada. Cada uno de los contadores de muones [3] de AMIGA están compuestos por 4 módulos centelladores enterrados a $\sim 2,25$ metros, cubriendo una superficie de 30 m^2 (ver figura 2). Estarán asociados a una estación del detector de superficie y cubrirán una superficie total de $\sim 23,5 \text{ Km}^2$. El diseño del módulo utiliza 64 centelladores plásticos recubiertos con TiO_2 de 4 cm de ancho, 1 cm de alto y 4 metros de largo, cada uno con un surco en la mitad de su cara superior donde se ubica una fibra óptica centelladora tipo WLS (*Wave-Length Shifting*) [4].

El centellador, al ser excitado cuando lo atraviesan muones, produce luz que es recogida, re-emitida y propagada por la fibra óptica, para luego ser convertida en pulsos eléctricos por un PMT (Photo Multiplier Tube) multianodo de 64 píxeles, obteniendo así pulsos de corriente de algunos nano-segundos de ancho en el ánodo. Las señales en cada uno de los

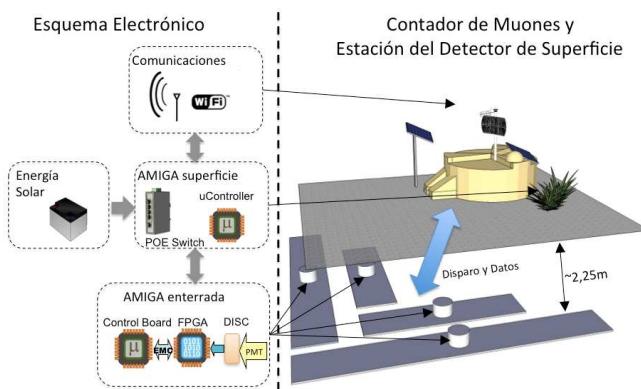


Figura 2. Cada estación del detector de superficie tendrá 4 módulos contadores de muones. En el esquema también se representa el sistema de alimentación de los módulos a través de un switch POE y la antena de comunicaciones WI-FI.

64 canales son amplificadas y, mediante comparadores de umbral programable, digitalizadas. Estas señales digitales son posteriormente muestradas a 320 MSps por un dispositivo lógico programable que almacena en memoria externa un total de 6,4 us por cada canal, conformando así un tren de “unos” y “ceros” correspondientes a presencia o ausencia de señales de los PMTs que hayan superado o no el nivel de umbral. Todo el sistema se alimenta con energía solar, siendo éste uno de los puntos críticos por lo que es necesario garantizar bajo consumo de la electrónica.

II. CONTROL INTERFACE BOARD

II-A. Requerimientos del sistema embebido

El diseño de la electrónica que se presenta en este trabajo tiene como objetivo obtener un sistema capaz de realizar el control, recolección, tratamiento y transferencia de datos. Todas estas tareas serán llevadas a cabo por la denominada Control Board. Además de ser autónomo, realizar los procedimientos y supervisión del sistema, este diseño es de muy bajo consumo (eliendo componentes especificados para tal fin, por ejemplo el DAC y el microcontrolador [7]) y alta confiabilidad, por ese motivo, a diferencia de una *single board computer*, este es un diseño a medida. Si bien la adquisición de datos se realiza por parte de un FPGA, un sistema embebido implementado, logra que el conjunto tenga cierta flexibilidad para el usuario, permitiéndole independizarse de la arquitectura utilizada. La funciones del detector de muones son totalmente programables en forma remota: configuración del sistema de disparo, programación de diferentes modos de adquisición de datos y de calibración, y la posibilidad de realizar cambios de bajo y alto nivel en el software y firmware.

II-B. Sistema Operativo

Teniendo en cuenta la duración prolongada del experimento AMIGA (~20 años incluyendo la fase de ingeniería y construcción), y los ciclos tecnológicos cada vez más acotados del hardware, resulta importante abstraer lo máximo posible el manejo de hardware de los programas a implementar en

el microcontrolador de la Control Board. Por lo tanto, se decidió usar un SO (sistema operativo) portable a distintos microcontroladores y arquitecturas para lograr esta abstracción.

II-B1. Gestor de arranque: Para poder utilizar un SO es necesario contar con un programa llamado *bootloader*, guardado en la ROM del dispositivo, cuya función es copiar a RAM externa y ejecutar el SO. Se utiliza el *bootloader* Das U-Boot [9], que es un software universal, flexible y de desarrollo activo. Dado que disponemos de un generoso ancho de banda para las comunicaciones, se decidió utilizar arranque por red, empleando el soporte que tiene el U-Boot para cargar imágenes vía TFTP. Esta operación tarda aproximadamente un minuto en cargar las imágenes del sistema operativo y las aplicaciones, todos ellos almacenados en el servidor del observatorio. Esto nos permite prescindir de la memoria flash, y facilita enormemente la actualización del software (ya que basta cambiar las imágenes de arranque).

II-B2. Kernel: El microcontrolador ARM7 utilizado no dispone de MMU (*Memory Management Units*), que realizaría la traducción de las direcciones lógicas o virtuales, a direcciones físicas o reales, la protección de la memoria y el control de cache; lo que implica una gran limitación para la implementación de cualquier SO. Se decidió utilizar *μCLinux* [10] (proyecto que realiza distintas versiones de Linux para dispositivos sin MMU) basado en el kernel de Linux 2.6.24.2, ya que es un sistema con una alta difusión, posee buen nivel de soporte, implementa una API estándar (POSIX), y permite brindar una interfaz de usuario familiar para los usuarios del dispositivo (científicos que participan en el proyecto AMIGA, que se encargarán de realizar los programas de calibración de los equipos antes de su puesta en marcha, y los programas de adquisición y análisis de los datos). Para el manejo de los periféricos que dispone el microcontrolador, se tomaron como base los drivers ya desarrollados, adaptándolos a las necesidades específicas de nuestro entorno [5]. En nuestro caso particular, implementamos los drivers de Ethernet, SPI, ADC, USB y UART.

II-B3. Entorno de Usuario: El entorno de usuario elegido para el SO es *busybox*. Este software presenta un entorno de usuario muy similar al de versiones de UNIX para PCs de escritorio, usando una cantidad acotada de memoria y siendo capaz de correr en entornos sin MMU. Para este proyecto, se compiló *busybox* para que, además de tener las utilidades estándar (*cp*, *ls*, etc), tenga herramientas de transferencia bidireccional de archivos (*tftp*, *wget*), y un servidor de acceso remoto (*telnetd*)¹.

II-C. Aplicaciones de usuario

A continuación se enumerarán las funciones que lleva a cabo la Control Board, para las cuales se programó el software correspondiente.

II-C1. Programación de FPGA vía JTAG: El front-end digital se basa en un FPGA, un Altera Cyclone III. Los FPGA son dispositivos volátiles, por lo que es necesario

¹No se utilizó un servidor SSH ya que la red de datos del proyecto es cerrada, y porque además aumenta notablemente el uso de CPU

reprogramarlos cada vez que se los enciende. En este caso, se optó porque sea el microcontrolador de la Control Board quien realice esta función. La interfaz elegida ha sido JTAG, basado en el programa JRunner, un software de código abierto ofrecido por el fabricante del FPGA, el cual fue originalmente concebido para correr bajo Microsoft Windows utilizando el puerto paralelo. Se eliminó el código correspondiente a bibliotecas de Windows y se lo adaptó a las de Linux, y se remplazaron las rutinas de bajo nivel que gobernaban el puerto paralelo por unas nuevas que utilizan 4 pines del GPIO del microcontrolador.

II-C2. Cliente: Se escribió el programa cliente de la Control Board para intercambiar información con el Centro de Cómputos. Éste utiliza un socket TCP/IP para implementar una comunicación segura, lo cual es posible gracias a la alta relación entre el ancho de banda disponible y el requerido. El programa cliente envía los datos de un evento (64 tramas de 2048 bits) cuando el servidor lo solicita.

II-C3. Interfaz con el dispositivo de adquisición: El sistema de adquisición basado en un FPGA almacena local y temporalmente los eventos en una memoria RAM. El FPGA se direcciona dentro del espacio de memoria del microcontrolador, y se accede a él utilizando el EMC (*External Memory Controller*). Por ende, acceder al FPGA equivale a leer o escribir registros ubicados en determinadas posiciones de memoria. Además, el FPGA fija el valor de los DACs (*Digital to Analog Converter*) que determinan el umbral de los discriminadores del front-end analógico, siendo el microcontrolador el que le indica con qué valores debe hacerlo. Por último se deberá verificar constantemente que el FPGA continúe funcionando correctamente, escribiendo en forma periódica un registro y leyendo en otro.

II-C4. Controlar la alta tensión del PMT: La regulación de la alta tensión del PMT es fundamental, pues determina parámetros importantes del mismo [6]. Esta tensión se ubica entre los 800 V y 1000 V. Para alcanzar estos valores, se utiliza una fuente que regula la alta tensión en función de la tensión de control, la cual es comandada por un DAC. Como el LPC2468 no cuenta con DAC propio [7], fue necesario utilizar uno externo, y se optó por usar uno controlado por SPI (periférico con el que sí cuenta el microcontrolador). Adicionalmente, para tener un correcto control del sistema, está previsto sensar constantemente la tensión del PMT, conectando la alta tensión mediante un divisor resistivo a uno de los ADCs (*Analog to Digital Converter*) del microcontrolador, y así monitorear que su valor sea el esperado.

II-C5. Monitoreo: La Control Board también tiene previsto realizar el monitoreo de diversas variables físicas presentes en el sistema, como ser las múltiples tensiones presentes en las placas, temperaturas, la ya mencionada alta tensión del PMT, etc. Estas variables son proveídas por distintos sensores ubicados en toda la electrónica enterrada, y son conectados a los ADCs del microcontrolador que encapsula los valores medidos dentro de la trama de datos que se comunican a la central de datos del Observatorio Pierre Auger cuando son requeridos.

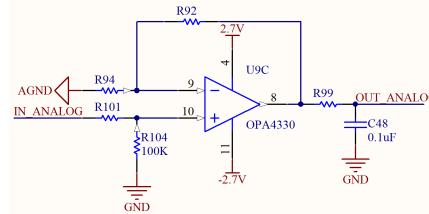


Figura 3. Circuito para separar las masas analógica y digital entre los ADCs y el microcontrolador.

II-D. Hardware

Para cumplir con las premisas fundamentales impuestas por el proyecto, el microcontrolador utilizado debe disponer de una interface para memoria externa para realizar la comunicación con el front-end de adquisición, y otra Ethernet para el comando y trasferencia de datos. Los microcontroladores ARM7TDMI de arquitectura RISC de 32 bits presentan, dentro del mercado mundial, un consumo nominal bajo ($\sim 0,28$ mA/MHz) respecto a otros microcontroladores de similares prestaciones. Es por ello que se seleccionó el modelo LPC2468 de NXP. Este modelo [7] cuenta con *dual AHB bus architecture, 512KB flash, USB 2.0 device/host/OTG, 10/100 Ethernet, SD/MMC, I2S, real-time emulation* y EMC. Los esquemas del microcontrolador en la Control Board están basados en las recomendaciones del fabricante. La memoria es una SDRAM de 256 Mbit a 133 MHz. La interfaz Ethernet necesita un controlador *PHY* y se optó por el integrado KSZ8721BLI de Micrel. La interfaz entre el microcontrolador y el FPGA se realiza a través de buffers. Para la alimentación de los circuitos se utiliza reguladores LDO (*low drop output*). Además cada dispositivo cuenta con filtros para las líneas de VCC. Para evitar lazos de masa e introducción de ruido digital en las variables de monitoreo analógicas, se separaron las masas de los circuitos externos (analógicos) de la propia masa de la placa (digital) en los ADCs, empleando un circuito diferencial como se muestra en la figura 3.

La memoria flash del microcontrolador se graba a través de un programador que se desarrolló para brindar una terminal serie y para la depuración del sistema durante la etapa de desarrollo.

II-E. PCB

Como parámetro de diseño para el circuito impreso se midió que la frecuencia de rodilla (knee frequency) es de ~ 200 MHz, siendo el ancho de banda mínimo necesario para transmitir la mayor cantidad de energía de la señal digital [5]. Se utilizó tecnología FR4 de 6 capas con un espesor de 1.6 mm. Las reglas de diseño del layout son: 7 mil de separación mínima entre pistas; 8 mil de ancho mínimo de cada pista; vías y pads con 28 mil de corona y 16 mil de agujero interior. El esquema de la pila de capas en orden descendente es: señal, GND, señal, señal, VCC, señal. Puede observarse la placa fabricada en las figuras 4 y 5.

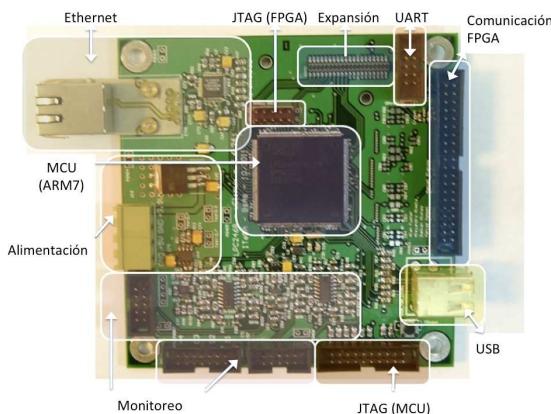


Figura 4. Vista superior de la Control Board.

II-F. Verificación

Para la verificación experimental de la correcta escritura/lectura de los registros del FPGA se conectó un analizador lógico Agilent 16821A en el bus de datos entre el microcontrolador y el FPGA. También se han verificado con éxito las funciones de programación y configuración de la FPGA, y las salidas de los DACs utilizados para la configuración de los niveles de umbral de los discriminadores y del alto voltaje para el PMT. Está previsto verificar la correcta lectura de las variables de monitoreo. Esto se realizará por medio de la inserción de la Control Board en un módulo contador de muones prototipo que ya ha sido construido y es actualmente utilizado en ITeDA para el estudio de la técnica de calibración y el software para análisis de datos. Este módulo prototípico cuenta con todo el resto de la electrónica del contador por lo que se puede realizar una completa prueba operativa de integración y simular distintos tipos de fallos y condiciones de trabajo posibles del verdadero contador de muones.

III. CONCLUSIONES Y TRABAJO FUTURO

Actualmente se encuentran construidos 5 prototipos de la Control Board que fueron completamente diseñados en ITeDA

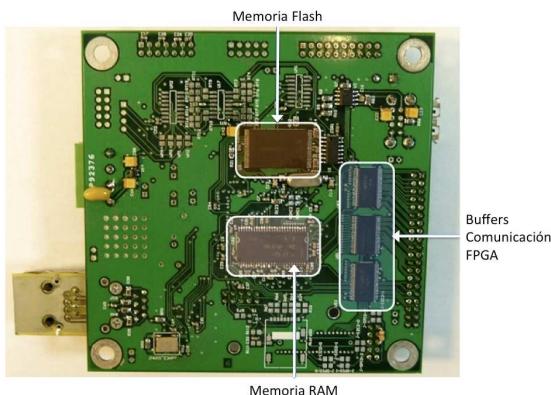


Figura 5. Vista inferior de la Control Board.

y cuyas PCBs (un total de 20) se hicieron fabricar en Advanced Circuits en los Estados Unidos. Se ha portado con éxito el SO y se han escrito programas de debug independientes para probar cada una de las funciones de la Control Board. En la actualidad se está trabajando en el desarrollo de las aplicaciones finales para que el SO ejecute todas las tareas simultáneamente. Una vez que se haya completado el software final de la placa, se la conectará a la electrónica de adquisición de uno de los contadores de muones para probar el sistema completo en laboratorio, validando así su comportamiento, y luego se instalarán los contadores de muones en el Observatorio Pierre Auger. A continuación se detallan algunas de las tareas que serán llevadas a cabo en el futuro, a fin de seguir mejorando el sistema diseñado.

- Integración de la Control Board y la placa del FPGA: Al momento de comenzar este trabajo, la placa del FPGA ya se encontraba fabricada, por lo que la Control Board fue diseñada como una placa aparte. En el futuro, y previo a la producción en masa de la electrónica de todos los contadores de muones que van a ser instalados, se prevé integrar ambas placas en una sola, con el objetivo de disminuir costos y espacio. Para lograr esta meta, debido a la complejidad de esta placa, será necesario realizar un estudio de compatibilidad electromagnética más profundo, empleando técnicas como la simulación por elemento finito.
- Servidor web centralizado de datos: deberá operar en el centro de cómputos, con el objetivo de poder visualizar las mediciones de monitoreo obtenidas por todos los contadores en cualquier posición del Observatorio.

REFERENCIAS

- [1] The Pierre Auger Collaboration, Nucl. Inst. and Meth. A523 (2004) 50.
- [2] A. Etchegoyen for the Pierre Auger Collaboration. "AMIGA, Auger Muon and Infill for the Ground Array". (2007) Proc. 30th ICRC (Mérida, México), 1307.
- [3] Brian Wundheiler, Rodolfo F. Gamarra, Alejandro Krieger, Alejandro Almela, Federico Sánchez, Alberto Etchegoyen. "Muon Counting: Signal Simulation, Measurements and Analyses". (2010). XI ICFA School on Instrumentation in Elementary Particle Physics (San Carlos de Bariloche, Argentina).
- [4] M. Platino, F. Suarez, M.R. Hampel, D.A. Almela, A. Krieger, D. Gorbeña, A. Kakazu, F. Gallo, A. Ferrero, G. De La Vega, A. Lucero, M. Videla, O. Wainberg, A. Etchegoyen, P.O. Mazur. "Fabrication and testing system for plastic scintillator muon counters used in cosmic showers detection" 32nd International Cosmic Ray Conference, Beijing 2011.
- [5] H.W.Johnson, M. Graham. "High-Speed Digital Design - A Handbook of Black Magic". Editorial: Prentice Hall. ISBN-13: 978-0133957242.
- [6] Federico Suarez, Agustín Lucero, Alberto Etchegoyen, Alejandro Almela, Carlos Reyes, Damían Alonso, Emmanuel Ponsone, Federico Barabas, Gonzalo De La Vega, Mariela Videla, Manuel Platino, Oscar Wainberg "A Fully Automated Test Facility for Multi Anode Photo Multiplier Tubes". (2011) 32nd International Cosmic Ray Conference, Beijing 2011.
- [7] NXP. "UM10237 LPC24XX User manual". (2009)
- [8] NXP. "AN10950: LPC24XX external memory bus example".(2010).
- [9] Das U-Boot - Universal Bootloader. Web para descarga: <http://sourceforge.net/projects/u-boot/>, Wiki: <http://www.denx.de/wiki/U-Boot>
- [10] µCLinux - Embedded Linux/Microcontroller Project. Web: <http://www.uclinux.org/>.
- [11] Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman. "Linux Device Drivers", Tercera edición.



Computación híbrida y colaborativa en acción

Un caso de estudio utilizando FPGA y PC

Eduardo A. Sanchez¹, Pablo A. Ferreyra^{2,3}, Carlos A. Marqués¹

1: Facultad de Matemáticas, Astronomía y Física – U.N.C.

2: Facultad de Ciencias Exactas, Físicas y Naturales –U.N.C.

3: Posgrado de Sistemas Embebidos – I.U.A.

Córdoba, Argentina

{esanchez, marques, ferreyra}@famaf.unc.edu.ar

Resumen—En la actualidad abundan los problemas que requieren el cálculo de algoritmos complejos, que dada la naturaleza del diseño de los procesadores modernos, no pueden ser computados de una manera eficiente. En muchos casos, el algoritmo implementado completamente en *hardware*, tampoco es una solución al problema. Es aquí donde la computación híbrida es capaz de tomar lo mejor de cada mundo. En este trabajo se presentará una metodología de cómputo híbrido entre la PC y una *Field Programmable Gate Array* (FPGA), tomando como caso de estudio el cálculo colaborativo de *cyclic redundancy check* (CRC). A lo largo de este trabajo, se describirá un método que demuestre la factibilidad del cómputo híbrido mediante un caso de estudio real, presentando los resultados obtenidos.

Palabras Claves: *FPGA, computación híbrida, lwIP, CRC, CRC-CCITT, Microblaze, ipcore*

I. INTRODUCCIÓN

Día a día aumentan los problemas que requieren el cálculo de algoritmos complejos específicos y que dada la naturaleza del diseño de los procesadores modernos es altamente probable que no sean computados de manera eficientes haciendo que su cómputo tome demasiado tiempo o recursos. En muchos casos, el algoritmo implementado completamente en *hardware* (en una FPGA, por ejemplo), tampoco es una solución viable al problema; ya que se desperdician recursos valiosos o el tamaño necesario para implementar la lógica propia en *hardware* lo hace privativo. Como una solución al problema, se propone el uso de computación híbrida: es decir, una combinación de la PC y la FPGA; tomando lo mejor de ambos mundos; al unir la flexibilidad de la comodidad del microprocesador (μ p), y el poder del *hardware* reconfigurable de la FPGA [1]. La PC se utiliza para trabajar con tareas no paralelizables, manejar fácilmente grandes volúmenes de datos, integrar los resultados, etc. mientras que la FPGA puede hacer los cálculos rápidamente, de manera eficiente, utilizando recursos específicos y consumiendo menos energía [2]. Si bien existen trabajos previos que analizan y hasta comprueban la factibilidad de la computación híbrida [3,5], utilizando FPGAs y PCs; no escapan a utilizar, parcial o totalmente, herramientas propias "cerradas" que no son de distribución libre y gratuita (como por ejemplo, el *SRC Software Development Kit* (SDK) [3] o *Mitriion SDK*; donde el código se compila a un procesador virtual propietario [6]). El presente trabajo tiene por objetivo

demonstrar empíricamente la factibilidad de dividir el procesamiento de algoritmos complejos, procesando parte en PC y parte en FPGA, de manera que combinados produzcan el resultado esperado. Se planea aplicar una adaptación de la técnica *divide-and-conquer* de un algoritmo escrito en lenguaje de alto nivel, que se encuentra corriendo en una PC; utilizando una modificación de la técnica propuesta en [3], de modo que sea transparente al algoritmo original que corría en la PC, tanto para el envío de datos desde/hacia la FPGA como para las llamadas a sus funciones. Como se menciona, no se utiliza el *SRC SDK* [7] ni *Mitriion SDK* [5] para los intercambios de datos entre PC-FPGA y viceversa, sino que se creará un servidor *Transmission Control Protocol / Internet Protocol* (TCP/IP) embebido para ello. Es decir, en base a un *stack TCP* dentro de la FPGA. Ésta, entonces, se comporta como un co-procesador, tomando ventaja de *hardware* específico [2]. La PC le envía datos vía un *frame ethernet*; ésta recibe esos datos, los procesa, los empaqueta y se los envía nuevamente a la PC que se los haya enviado. Como beneficio adicional, obtenemos una transparencia completa en cuanto a la ubicación de los recursos, ya que el sistema embebido y el cliente en la PC pueden estar en ubicaciones diferentes, y en cuanto a quien provee el servicio; ya que el cliente no sabe si es el servidor local o el servidor embebido el que lo está proveyendo. Cabe destacar que si bien la PC-FPGA funcionan como un sistema computacional integral, al tener una división clara entre el cliente y el servidor, se agrega un nivel de abstracción más; permitiendo a otros clientes (aplicaciones heterogéneas), sin el conocimiento específico del *hardware*, acceder a él mediante una simple conexión al servidor.

El resto del trabajo está organizado de la siguiente forma: La sección II presenta la metodología propuesta de trabajo para demostrar la factibilidad de la computación híbrida y su desarrollo en cada una de las subsecciones. La sección III muestra el funcionamiento del sistema colaborativo y los resultados obtenidos. Por último, en la sección IV se presentan las conclusiones y mejoras futuras del presente trabajo.

II. DESCRIPCIÓN DEL PROCESO

Si bien existen varios caminos posibles, se propone seguir el siguiente proceso para demostrar computación híbrida:

Se agradece a la Secretaría de Ciencia y Tecnología de la Universidad Nacional de Córdoba por otorgar becas y subsidios para permitir el desarrollo de este trabajo.

- Elegir e implementar un algoritmo para demostrar la factibilidad.
- Dividir el algoritmo, en cliente y servidor por *software*.
- Verificar y validar el correcto funcionamiento de ambos.
- Crear el bloque de *hardware* específico al problema.
- Crear un sistema embebido base.
- Crear el *ipcore* para ese *hardware*.
- Agregar el *ipcore* al sistema embebido base.
- Modificar el *software* en el Microblaze.

A. Elección e implementación de un algoritmo para demostrar la factibilidad

Se propone computar de manera híbrida el cálculo de CRC de archivos completos. El CRC es un código de detección de errores usado comúnmente en redes digitales y dispositivos de almacenamiento para detectar cambios accidentales en los datos; proveyendo la primer línea de defensa contra la corrupción de datos y la detección de errores en redes embebidas y otras aplicaciones [8]. En particular, el CRC elegido es el llamado CRC-CCITT, también conocido como CRC-16-CCITT; cuyo polinomio es el $x^{16}+x^{12}+x^5+1$.

Se plantea tomar un desarrollo legado en funcionamiento [9] y realizar la menor cantidad de modificaciones al código. En particular, como parte de la estrategia de *divide-and-conquer*, se debe modificar la función *crcFast* original. Dicha función calcula el CRC-CCITT de un mensaje completo:

Ej: *crcFast*("123", 0xFFFF) = 0x5BCE

Se sugiere agregar una nueva función que permita calcular sólo un paso del CRC-CCITT, a partir de un valor de CRC inicial dado. Esto permite ir obteniendo resultados parciales; sólo basta con darle un mensaje y un CRC inicial, para que la función compute el resultado intermedio. Luego, el resultado de la aplicación sucesiva de la función *crcFastPartial* a los segmentos del archivo, dan como resultado el valor CRC original calculado por la función *crcFast*; tal como se aprecia en el ejemplo de la Fig. 1.

B. División del algoritmo en cliente y servidor por software

Hasta el momento, se realiza el cálculo completo en la PC. El paso siguiente es dividirlo según el modelo cliente/servidor. Ambos pueden seguir corriendo en la misma PC. Cabe destacar que el cliente no debe sufrir ningún cambio cuando el servidor en la PC es remplazado por el servidor en la FPGA.

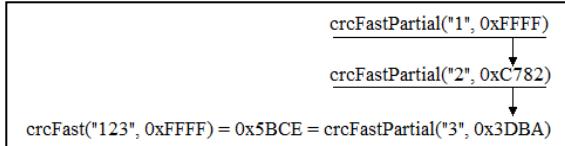


Figura 1. Relación entre funciones *crcFast* y *crcFastPartial*

Es responsabilidad del cliente abrir una conexión al servidor, leer una porción del archivo, empaquetarla en una estructura, enviarla al servidor y esperar por el resultado. Una vez que recibe el *output* del servidor, prepara un nuevo paquete incluyendo el CRC previamente calculado. Y repite el ciclo hasta completar el archivo.

A su vez, el servidor espera pasivamente que un cliente se conecte, y le envíe los datos a calcular el CRC. El servidor, entonces, llama a la función *crcFastPartial*, prepara el paquete de retorno y devuelve el resultado al cliente. En la estructura que le manda el cliente, se puede indicar al servidor, vía un *flag*, que el actual es el último paquete a calcular, por lo que luego de enviar los resultados, debe cerrar la conexión. En este caso, el servidor se queda pasivo esperando nuevas conexiones. Cabe destacar que al utilizar funciones *send* y *recv* bloqueantes, no hay problemas de sincronismo, ya que el cliente espera bloqueado por los datos del servidor y viceversa.

En este caso en particular, el servidor (tanto en la PC como en la FPGA) solo es capaz de calcular el CRC de una pequeña sección del archivo, siendo el cliente (en la PC) responsable de dividir el archivo en porciones e ir enviándolas con el CRC inicial. Más aún, sin el soporte del cliente, el servidor no puede calcular el CRC de archivos de más de 1430 bytes (máximo *payload* en un *frame ethernet*).

C. Verificación y validación del cliente-servidor por software

Existen numerosas herramientas en la red que calculan el CRC dado un mensaje inicial y el polinomio deseado; tal es el caso de [10]. Utilizando estas herramientas, fácilmente se pueden corroborar los resultados de la Tab. 1 tanto en el modelo *standalone* modificado de [9] (sección II.A) como en el modelo cliente/servidor. La validación y verificación por *software* tanto el cliente como el servidor no sólo permiten diseñar las funciones y estructuras necesarias; sino también corregir cualquier problema de comunicación o *endianness* antes de sintetizar o implementar en *hardware*, donde la corrección es mucho más costosa.

D. Creación de un bloque de hardware específico

El paso siguiente es crear un bloque *hardware* en *Very High Speed Integrated Circuit - Hardware Description Language* (VHDL) o Verilog, que permita el cálculo CRC-CCITT.

TABLA I. MENSAJES Y CRC-CCITT RESULTANTES

MENSAJE	Tamaño en bytes	CRC-CCITT (0xFFFF)
123456789	9	0x29B1
abcde...vwxyz.	26	0x53E2
6K de la Iliada de Homero ^a	6144	0x30A5
La Iliada de Homero ^a	880404	0x17DE
Las Mil y Una Noches ^a	1281772	0xD9C3

a. Fuente: <http://libroteca.net>

Tal como se utilizó el código legado de [9] puede utilizarse la herramienta generación de código automática de [11], la cual con sólo seleccionar el polinomio adecuado y el ancho del *bus* de datos de entrada, genera el código VHDL necesario para calcular el CRC a partir de esa selección. Se obtiene como resultado una nueva entidad y su *testbench* con la firma de la Fig. 2. Esta nueva entidad permite “secuenciar” por *hardware* el procesamiento de CRC siendo equivalente a la función por *software* *crcFastPartial*. Creado el bloque en Xilinx *Integrated Software Environment* (ISE) y testeado, hay que encapsularlo en un *ipcore* para ser utilizado por un sistema embebido más complejo. Pero antes, se debe crear un sistema embebido base.

E. Creación de un sistema embebido base

Utilizando el *Embedded Design Kit* (EDK) de Xilinx, se crea un sistema embebido base para la placa de desarrollo (en este caso, XUPV5); a través del *base system builder wizard*. Dado que debe ser capaz de alojar un servidor embebido, es necesario un sistema que contenga mínimamente los siguientes recursos: *μp Microblaze*, memoria externa DDR2, interfaz MAC, *timer*, controlador de interrupciones y UART. Aproximadamente, y sin entrar en detalles de implementación, el sistema embebido base tiene un 26% de *slices* ocupadas (FPGA Virtex-5), por lo que el diseño puede crecer quedando espacio de sobra suficiente para el desarrollo propio de *hardware* que se necesite agregar.

Se coloca en el único hilo de procesamiento el *stack TCP* llamado lwIP - “A Lightweight TCP/IP stack”. lwIP es una implementación pequeña y liviana del protocolo TCP/IP, capaz de correr directamente sobre el *μp Microblaze* sin la necesidad de un sistema operativo que lo controle. El foco en la implementación de lwIP es reducir al mínimo, el uso de los recursos manteniendo TCP [12], haciendo posible la integración en sistemas embebidos. Cabe destacar que lwIP es utilizado vastamente por muchos fabricantes en la industria de sistemas embebidos (Altera, Xilinx, Analog Devices, etc.); integrándolo en el *software* que corre en sus placas de desarrollo. Una vez integrado el *software* de lwIP con el *hardware*, se debe verificar el correcto funcionamiento de ambos mediante la aplicación de ejemplo “echo” incluida en la distribución de lwIP.

F. Creación del ipcore para el hardware específico e inclusión en el sistema embebido base

En base al *hardware* específico generado y testeado en II.D, se debe generar un *ipcore* que sea capaz de interactuar directamente con las señales del *Processor Local Bus* (*PLB*). Es decir es necesario generar un *wrapper* que encapsule todas

```
entity crc_ccitt_easics is
  Port (
    clk: in STD_LOGIC;
    data_in : in STD_LOGIC_VECTOR (31 downto 0);
    crc_in : in STD_LOGIC_VECTOR (31 downto 0);
    crc_out : out STD_LOGIC_VECTOR (31 downto 0)
  );
end crc_ccitt_easics;
```

Figura 2. Nueva entidad generada automáticamente

las señales de la entidad *crc_ccitt_easics* y que “conecte” las señales necesarias provenientes del *bus PLB* como puede observarse en la Fig. 3. Luego de hacer las modificaciones correspondientes, se debe agregar el *ipcore* al sistema embebido base, y conectarlo al *bus PLB*. Con la nueva incorporación, no hay cambios significativos en la cantidad de recursos utilizados por la placa, 27% de *slices* ocupadas en total.

En este punto, se posee todo el *hardware* necesario para computar el CRC-CCITT de forma colaborativa e híbrida, tal como puede observarse en el diagrama en bloques de la Fig. 4. La UART se utiliza para *debugging* general, mientras que los bloques *PHY Ethernet*, *Timer*, *INT Cont* (controlador de interrupciones) y DDR2 son necesarios para que el *μp* pueda correr el *stack lwIP*.

G. Modificación del software en el Microblaze

Ya que el presente trabajo es únicamente una prueba de concepto, se puede reutilizar la mayor parte del código “echo”. Basta con modificar la función *recv_callback*, de modo tal que sea capaz de manejar los datos y CRC inicial que envía el cliente y escribirlos en las posiciones de memoria correctas. Dado que el CRC es calculado mediante lógica combinacional pura, se puede obtener el resultado parcial en la misma función y utilizarlo como CRC inicial para la próxima porción de datos. Una vez que todos se terminan de consumir, deben retornarse los resultados al cliente. Terminadas las modificaciones de *software*, se debe bajar el *bitstream* junto con el archivo *Executable and Linkable Format* (ELF) a la placa; permitiendo la conexión al servidor que calcula el CRC-CCITT dentro de la FPGA.

III. FUNCIONAMIENTO DEL SISTEMA HÍBRIDO Y RESULTADOS OBTENIDOS

Una vez terminado el desarrollo, para poder testear el sistema completo, se levantó una máquina virtual con el sistema linux *OpenSuse*.

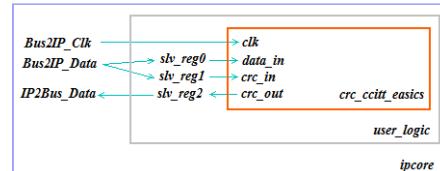


Figura 3. Diagrama de conexiones internas de las señales

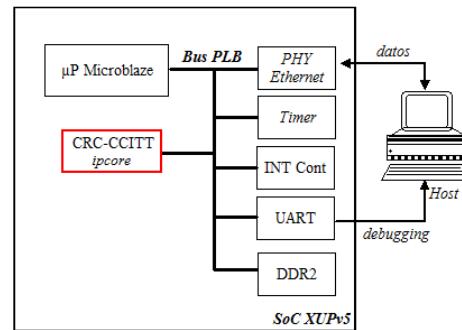


Figura 4. Diagrama en bloques del SoC desarrollado



Esto permitió compilar fácil y rápidamente el cliente, brindando la posibilidad de *debugging* con la herramienta *Wireshark* los paquetes enviados desde y hacia la placa de desarrollo. De hecho, ayudó a encontrar y solucionar rápidamente un pequeño problema de alineación de *bytes*. En el cliente se agregó la funcionalidad de calcular el CRC localmente, de modo que se pudieran corroborar los resultados fácilmente. Todas las pruebas fueron verificadas y validadas, coincidiendo en todos los casos con los resultados retornados por el servidor local, como puede apreciarse en la Fig. 4.

IV. CONCLUSIONES Y POSIBLES MEJORAS

Se demostró la factibilidad de crear un sistema híbrido, procesando parcialmente el algoritmo en la PC y en la FPGA, de manera transparente a la PC Host (que realizaba el procesamiento completo). Se creó y/o modificó el *hardware* y el *software* mínimo, necesario y suficiente para procesar de manera colaborativa e híbrida el algoritmo CRC-CCITT. Cabe destacar que el *hardware* que realiza el procesamiento de CRC-CCITT, se escogió por motivos didácticos, sólo para demostrar la viabilidad del sistema y no por ser paralelizable u óptimo. Si bien existen trabajos donde se muestra la factibilidad de la computación híbrida colaborativa entre una PC y una FPGA, los mismos se basan fundamentalmente en el uso de herramientas comerciales totalmente o parcialmente cerradas [3][5]. Este trabajo contribuye con la descripción de un nuevo proceso, que salvo las herramientas integradas de Xilinx o Altera, demuestra la factibilidad de la computación híbrida colaborativa por medio de *software* libre. Otra contribución muy importante, que los demás trabajos no permiten, es obtener una transparencia en la ubicación de los recursos. Esto último posibilita que el servidor pueda estar apartado del cliente, que se pueda remplazar el *software*, el *hardware* e incluso la placa de desarrollo sin la necesidad de modificar y/o notificar al cliente que consume el servicio. Cabe destacar que un nivel de abstracción mayor (mediante el modelo cliente/servidor) posibilita compartir el recurso, que computa el algoritmo en la FPGA, con otras aplicaciones sin costo adicional. Este trabajo no buscó elevar la velocidad o mejorar la performance en el procesamiento sin embargo, dependiendo del algoritmo a atacar, puede ser necesario hacerlo. La comunicación puede ser un cuello de botella cuando hay que mover una gran cantidad de datos entre la PC y la FPGA: TCP necesita mucho *handshake* en ambos extremos (cliente y servidor) para determinar que no se hayan perdido paquetes o que los datos hayan llegado corruptos. Para atacar este problema, por el momento se descarta utilizar PCI Express (PCIe) pese a que hay trabajos al respecto [1], dado que se perdería la transparencia en los recursos y aunque PCIe representa la solución más rápida del mercado, también puede ser la más costosa [13] ya que no siempre están disponibles los

drivers de bajo nivel necesarios [13][1]. La combinación de IPv4 con *User Datagram Protocol* (UDP) representa, entonces, una solución más atractiva en términos de requerimientos de recursos *hardware* para mejorar la transmisión de datos entre FPGA y PC [13]. Llegado el caso, está disponible en versión libre y gratuita, un *ipcore* UDP/IP que implementa una arquitectura de comunicación FPGA-PC eficiente [14]. En mejoras futuras se planea la exploración de la computación híbrida, utilizando el diseño e implementación propuestos por este trabajo, en algoritmos más complejos [4][15,16] que exploten el paralelismo y los recursos propios fijos de las FPGAs. Podemos concluir que si bien se han utilizado herramientas de Xilinx tales como ISE y EDK; no es condición *sine qua non* ya que fabricantes como Altera ponen a disposición del desarrollador herramientas muy similares como *Quartus*, con las mismas características.

REFERENCIAS

- [1] M. Leonov; V.V. Kitaev. "Feasibility Study of Implementing Multi-Channel Correlation for DSP Applications on Reconfigurable CPU+FPGA Platform". IEEE Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies. pp. 159-166. 2008.
- [2] Mitronics. "Low Power Hybrid Computing for Efficient Software Acceleration". White Paper. Disponible online en <http://www.mitronics.com/?document=Hybrid-Computing-Whitepaper.pdf>. 2008. [2012].
- [3] Alam, S.R.; Agarwal, P.K.; Smith, M.C.; Vetter, J.S.; Caliga, D., "Using FPGA Devices to Accelerate Biomolecular Simulations", IEEE Computer Society, vol. 40, pp. 66-73, Marzo 2007.
- [4] D. Isaacs; E. Trexel; B. Karsten, "Accelerate system performance with hybrid multiprocessing and FPGAs", Embedded Systems Design, Vol. 20 No. 8, pp. 38-45, Agosto 2007.
- [5] Mitronics Inc. website disponible en <http://www.mitronics.com>. 2011.
- [6] V.V. Kindratenko; R.J. Brunner; A.D. Myers. "Mitron-C Application Development on SGI Altix 350/RC100". IEEE Symposium on Field-Programmable Custom Computing Machines. pp.239-250. 2007.
- [7] SRC Computers Inc. Colorado Springs. Website disponible en <http://www.srccomp.com/>. [2012].
- [8] Koopman, P.; Chakravarty, T., "Cyclic redundancy code (CRC) polynomial selection for embedded networks", Dependable Systems and Networks, 2004 International Conference on, pp. 145-154, 2004.
- [9] Barr M., "CRC Implementation Code in C", Internet: <http://www.netrino.com/Embedded-Systems/How-To/CRC-Calulation-C-Code>, Jan. 2000 [Nov. 2011].
- [10] Bies L, "On-line CRC calculation and free library", Internet: <http://www.lammertbies.nl/comm/info/crc-calculation.html>, Aug. 2011 [Oct. 2011].
- [11] Easics, "CRC Tool", Internet: <http://www.easics.com/webtools/crctool>, 2009 [Feb. 2012].
- [12] Dunkels A., "lwIP - A Lightweight TCP/IP stack", Internet: <http://savannah.nongnu.org/projects/lwip> Oct. 2002 [Dec. 2011].
- [13] N. Alachiotis; S.A. Berger; A. Stamatakis. "Efficient PC-FPGA Communication over Gigabit Ethernet". IEEE 10th International Conference on Computer and Information Technology. pp. 1727-1734. 2010.
- [14] UDP/IP Core, http://opencores.org/project.udp_ip_core. 2010. Updated 2011. [2012]
- [15] Oviedo M.; Ferreyra P.; Marqués C., "Sistemas en Chip acelerados por hardware: comparación de performance en aplicaciones criptográficas", Case 2011, Marzo 2011.
- [16] Wohlmuth, Otto, "High performance computing based on FPGAs", IEEE Field Programmable Logic and Applications, FPL, 2008

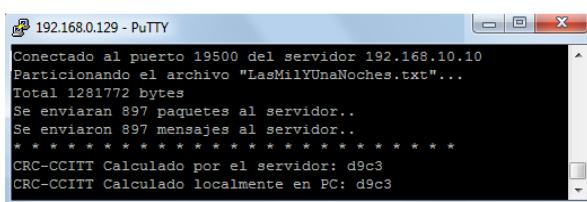


Figura 5. Cálculo híbrido de CRC-CCITT de "Las Mil y Una Noches"



Medición de vibraciones en pequeños motores eléctricos

Cascallares, Ivan; Castellini, Ariel; Panasiuk, Juan Manuel; Tantignone, Hugo; Zaradnik, Ignacio José
Cátedra de Tecnología Electrónica (383), Departamento de Ingeniería e Investigación Tecnológica
Universidad Nacional de la Matanza. Buenos Aires, Argentina.

inac89@hotmail.com; arielcastellini@gmail.com; jmp7788@gmail.com; htantignone@gmail.com;
izaradnik@unlam.edu.ar

Abstract—En el presente trabajo se pretende explicar el diseño de un sistema de medición de vibraciones para pequeños motores eléctricos. El diseño incluye la definición de características del dispositivo, la selección de componentes, la implementación del hardware, firmware y software asociado, y las pruebas y ensayos. Se hará referencia también, a los criterios de procesamiento digital de señal tenidos en cuenta y a las normativas asociadas a este tipo de medición.

Keywords- vibraciones; acelerómetros; microcontrolador; MEMS; transformada rápida de fourier (FFT); Matlab

I. INTRODUCCIÓN

En todas las instalaciones y plantas industriales donde existe maquinaria de producción con elementos dotados de movimiento rotativo o alternativo, se hace necesario efectuar un mantenimiento de estas máquinas para conservarlas en correcto estado de servicio y garantizar la seguridad y fiabilidad de la planta. Para lograr estos objetivos se puede realizar un mantenimiento preventivo, el mismo implica un desmontaje total o parcial de la maquinaria para su inspección y en casos de que existan defectos, proceder a la reparación de los mismos. Este método presenta una serie de inconvenientes, los cuales pueden ser soslayados por un mantenimiento predictivo por vibraciones mecánicas [1].

En la práctica la vibración aparece producto de la transmisión normal de fuerzas cíclicas a través de los mecanismos. Parte de la energía que se entrega a la máquina se disipa en sus elementos, dando a la estructura una vibración característica. La condición de la máquina está íntimamente ligada con las vibraciones que ella produce, esto hace que la medición y el análisis de vibraciones sean las herramientas básicas del mantenimiento predictivo por vibraciones mecánicas [1].

Un ejemplo de lo antes dicho se puede encontrar en el artículo Análisis vibrodinámico de motores eléctricos [6], el cual expone los trabajos realizados en las diferentes etapas de implementación del mantenimiento y el diagnóstico de defectos mediante la utilización del análisis espectral de vibraciones. En el mismo se podrán observar espectros del motor con problemas y en funcionamiento normal.

Para el diagnóstico de los procesos y de la condición de las máquinas, a través del procesamiento de las vibraciones, se puede utilizar diferentes técnicas, como ser el análisis

espectral, cepstral, de envolvente, de tiempo-frecuencia, estadístico de orden superior y cicloestacionario. Aunque la técnica de análisis espectral es considerada la más importante, y la cual se implementara en este trabajo, el resto de las técnicas presenta características que lo hacen especiales para determinadas aplicaciones [7].

El trabajo que se describe a continuación fue realizado como un proyecto de alumnos dentro de la cátedra de Tecnología Electrónica de la carrera de Ingeniería Electrónica, el mismo no buscó el diseño de un producto comercial, sino la utilización de dispositivos MEMS y técnicas de procesamiento digital de señales en una aplicación con implicancias en la industria.

II. DEFINICIÓN DE PRODUCTO

A. Campo de aplicación

Lo ideal en todo buen desarrollo es definir las especificaciones en función de las necesidades y diseñar el producto de forma tal que las cumpla. Es decir, el elemento sensor debería ser determinado por el rango de vibraciones que se desea medir. En nuestro caso y por tratarse de un proyecto de alumnos, se seleccionó un sensor de fácil adquisición en el mercado y en función de este se definieron las especificaciones y por lo tanto las posibles aplicaciones.

El elemento sensor utilizado es un acelerómetro modelo MMA 7361 de empresa Freescale Semiconductor, el cual en función de sus características permite el análisis de vibraciones en las siguientes aplicaciones [2][3]:

- Lavadoras.
- Aspiradoras.
- Licuadoras.
- Herramientas eléctricas.
- Sistemas compresores.

Es decir pequeños motores eléctricos.

B. Especificaciones

La norma ISO 2372, que determina las bases para la especificación de la normativa de evaluación de vibraciones mecánicas de máquinas con velocidades de operación entre 10



y 20 rev/s, y la norma ISO 2373, una adaptación especial para motores eléctricos, especifican los siguientes parámetros como medición de las vibraciones:

- Desplazamiento [mm, um]
- Velocidad [cm/s, mm/s]
- Aceleración [m/s², g]

Las especificaciones del sistema diseñado están determinadas por las del acelerómetro y el microcontrolador utilizados [2][3][4].

- Rango de aceleración 0 a 58.8 m/s²
- Rango de frecuencia 0 a 1 kHz
- Resolución 0.882 m/s²

III. DISEÑO

A. Diagrama en bloques

En la figura 1 se puede ver el diagrama en bloques de nuestro sistema, en el que se pueden destacar principalmente tres bloques, Sensor, Microcontrolador y CPU con interfaz gráfica, los cuales serán detallados más adelante.

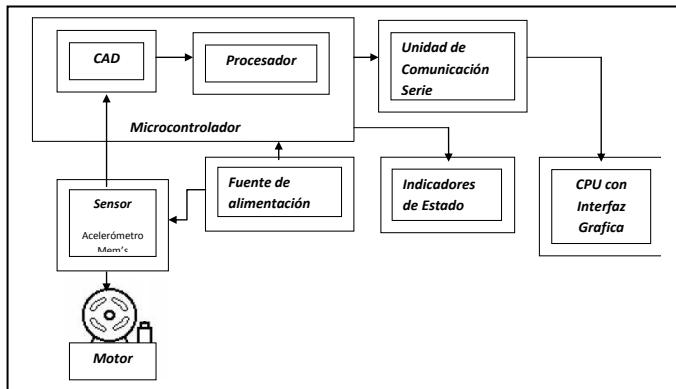


Figura 1. Diagrama en bloques.

Básicamente el funcionamiento es el siguiente: el acelerómetro, adosado rígidamente al cuerpo del motor, sensa las vibraciones que en este se producen. Convierte este movimiento mecánico en una tensión correspondiente a la aceleración de la superficie, el cual es tomado por el CAD del microcontrolador en forma de muestras que luego son magnificadas y codificadas digitalmente. A los datos obtenidos se les aplica una Transformada Rápida de Fourier, procesamiento que permite obtener la composición espectral de la señal muestreada, lo que nos posibilita ver las características, amplitudes y frecuencias, de las distintas vibraciones a la que se encuentra afectado el motor. Esta información es luego enviada a través de una comunicación serie hacia una PC, la cual a través de un software, desarrollado en Matlab, presenta los resultados en un gráfico de barras ordenado en función de la frecuencia, permitiendo la fácil visualización de las componentes frecuenciales de la vibración.

La interpretación que se hace de esta información depende puntualmente del motor estudiado y también del sistema en el

que ese motor actúa; pero en líneas generales, consiste en un contraste entre el espectro medido y el espectro normal de funcionamiento del motor, la presencia de una anomalía puede indicar el comienzo de una falla. El espectro vibratorio de una máquina es tan característico de ella, que puede saberse el tipo de falla que se está dando, de acuerdo a la anomalía puntual que se detecte en su espectro.

B. Selección de componentes

- Acelerómetro MMA7361. Tal como se comentó anteriormente, se buscó un dispositivo de fácil obtención en el mercado local. Analizadas alternativas de Freescale, ST y Analog Devices, se optó por la de Freescale ya que poseía las especificaciones técnicas necesarias para nuestra aplicación, había dos proveedores locales y existía una placa de evaluación, KIT3376MMA73x1L [5], figura 2, la cual nos facilitaba el montaje del mismo.

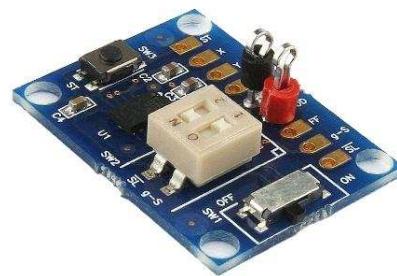


Figura 2. KIT3376MMA73x1L

- Microcontrolador ATmega8. Para el diseño del dispositivo se necesitaba contar con un microcontrolador que tuviera al menos 3 entradas analógicas, una para cada uno de los ejes del acelerómetro y un interfaz tipo UART para la comunicación con la PC. Ambas características soportadas por el ATmega8. Además debía tener la posibilidad de muestrear la señal del acelerómetro al menos al doble de la frecuencia máxima especificada, lo cual está determinado por el teorema de muestreo [5]. El ATmega8 puede muestrear la señal hasta 15khz [4], frecuencia por demás satisfactoria. Si bien existen en el mercado microcontroladores con un desempeño mayor, el ATmega8 puede realizar hasta 16 MIPS, este es suficiente para el tipo de aplicación que se está implementando.

C. Firmware del microcontrolador.

Para realizar distintas operaciones de procesamiento digital de señal como el filtrado digital, el análisis de la correlación o el análisis espectral, procesamiento el cual es de nuestro interés para nuestra aplicación, es necesario la implementación de la transformada discreta de Fourier (DTF). Básicamente, el problema del cálculo de la DTF es calcular la secuencia $\{X[k]\}$ de N números complejos dada la secuencia de datos $\{x[n]\}$ de longitud N según la ecuación 1 [8].

$$X[k] = \sum_{n=0}^{N-1} W_N^{kn} x[n] \quad 0 \leq k \leq N - 1 \quad (1)$$

Donde

$$W_N = e^{\frac{-j2\pi}{N}} \quad (2)$$

El cálculo DTF se lleva a cabo a través de una función especialmente desarrollada para micros de 8 bits [9].

Básicamente el algoritmo trabaja brindándole los siguientes parámetros:

- Dos vectores, uno con la parte real y otro con la imaginaria, la cual es cero, de los valores obtenidos por el acelerómetro.
- Cantidad de muestras.
- Tipo de transformada discreta a realizar.

La figura 3 muestra un diagrama de flujo de lógica implementada dentro del microcontrolador.

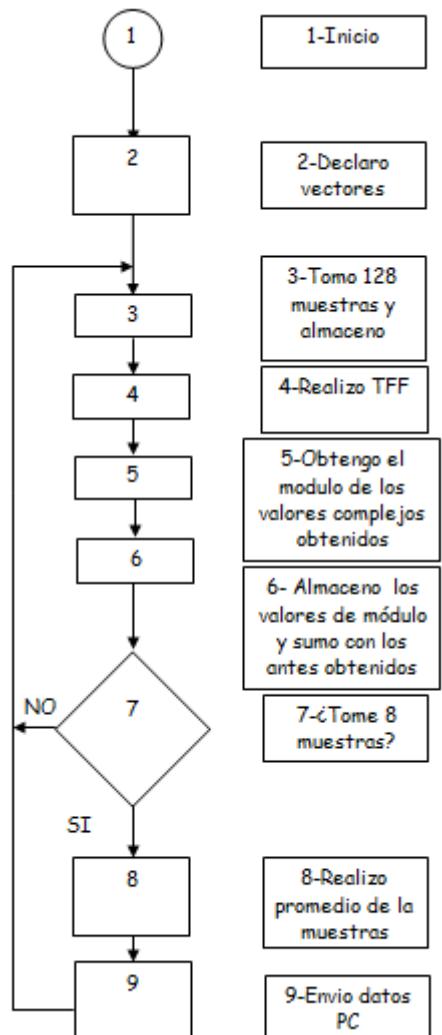


Figura 3. Diagrama de flujo.

El algoritmo de DTF nos entregará un valor complejo, una parte real y una parte imaginaria, asociado a cada frecuencia de estudio.

De cada uno de estos valores se obtendrá el módulo, el cual será utilizado para representar la amplitud de la componente de frecuencia en el gráfico.

Esta operación, la obtención de los coeficientes y el módulo de este valor complejo, se repite 8 veces para obtener un promedio y así poder acotar el error de las aproximaciones con números enteros.

Finalmente el resultado del promedio de cada uno de los valores obtenidos será transmitido a la PC donde se graficará el espectro.

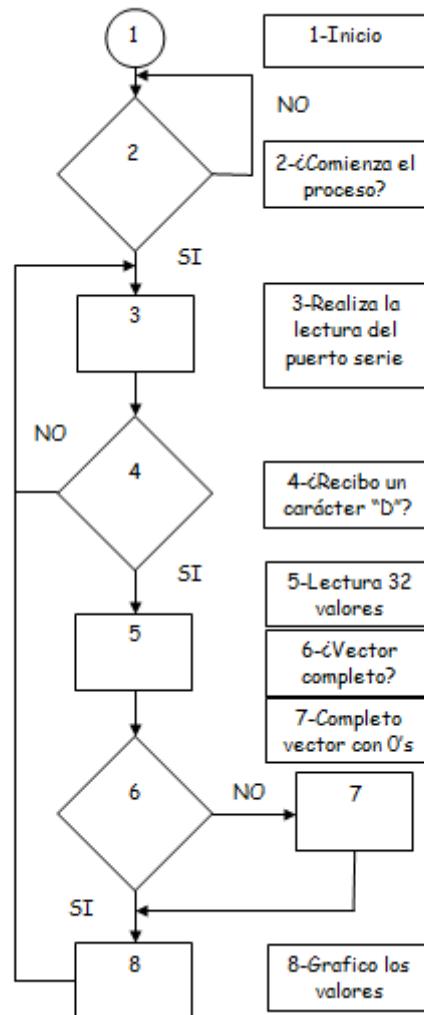


Figura 4. Diagrama de flujo del software.

D. Software de PC

La figura 4 muestra un diagrama de flujo de la interfaz gráfica implementada en la PC.

La figura 5 muestra la interfaz gráfica.

IV. CONCLUSIONES

El presente trabajo ha representado una muy buena herramienta para explicar conceptos relacionados a la cátedra donde el proyecto fue realizado (MEMS). Además de brindar



un acercamiento a los alumnos a aplicaciones industriales y a los sistemas embebidos. Si bien existen varios puntos a mejorar, entre los que podemos nombrar la presentación gráfica, el comportamiento del dispositivo ha tenido un desempeño satisfactorio cumpliendo las expectativas.

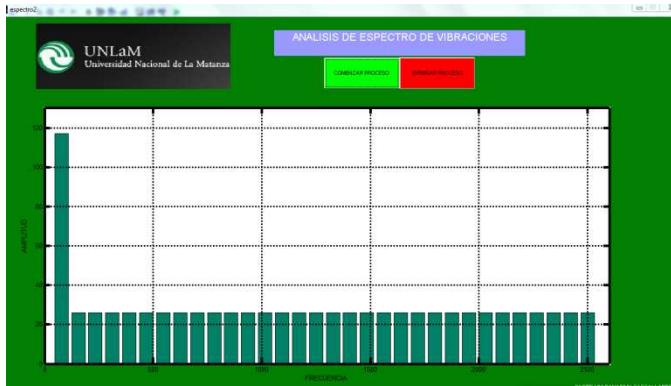


Figura 5. Interfaz gráfica.

REFERENCIAS

- [1] Las vibraciones mecánicas y sus aplicaciones al mantenimiento predictivo. G Mosquera, M de la Victoria Piedra Diaz, R. Armas Cardona. ISBN 980 00-1490-X 980 07 5678-7.
- [2] Document Number: MMA7361LRev 0, 04/2008.
- [3] Frequency Analysis in the Industrial Market Using Accelerometer Sensors, AN3751 Rev 0, 07/2008
- [4] Datasheet ATmega8(L), 2486Z-AVR-02/11.
- [5] http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=KT3376MMA73x1L#
- [6] Análisis vibrodinámico de motores eléctricos. Roberto Manuel Torres-Rodríguez, Carlos Ramón Batista-Rodríguez. Ingeniería Mecánica. Vol. 13. No.1, enero-abril de 2010, pag. 9-18 ISSN 1815-5944.
- [7] Aplicación de técnicas clásicas y avanzadas de procesamiento de vibraciones al diagnóstico de cojinetes. Análisis experimental. F. E. Hernández Montero, V. Atxa Uribe. Ingeniería Mecánica, 1 (2007) 71-82.
- [8] Proakis, J. G. y Manolakis, D. G., 2003. Tratamiento Digital de Señales. Tercera Edición. Prentice Hall. Madrid.
- [9] Written by: Tom Roberts 11/8/89, Made portable: Malcolm Slaney 12/15/94 malcolm@interval.com, Enhanced: Dimitrios P. Bouras 14 Jun 2006 dbouras@ieee.org. Modified for 8bit values David Keller 10.10.2010.



Desarrollo de un dispositivo inalámbrico para monitoreo de variables utilizando The Contiki OS

J. Morant, S. Marín, J. Velásquez

Departamento de I+D+I

Netux Tecnología Aplicada, Medellín, Colombia.

juan.morant@netuxtecnologia.com, info@netux.com

Abstract—Se describe el diseño, construcción y funcionamiento de un dispositivo de alta escala de integración con la capacidad de obtener datos de diversas variables del entorno y transmitirlas de manera inalámbrica a una puerta de enlace para su correcta visualización y monitorización. El nodo sensor de ultrabajo consumo corre el sistema operativo The Contiki, diseñado especialmente para sistemas limitados en capacidades y que fue portado completamente para la plataforma de hardware durante el proceso de diseño.

Index Terms—MOTE, The Contiki OS, Wireless Sensor Networks.

I. INTRODUCCIÓN

En el desarrollo de este artículo se presentan las etapas de especificación, diseño, desarrollo e implementación de un sensor inalámbrico con características convenientes para su implementación en una red de alta funcionalidad y bajo consumo energético. En la primera etapa del desarrollo del artículo se presenta un breve estado del arte acerca de los diversos protocolos de comunicación, sistemas operativos y plataformas de hardware disponibles que fueron considerados en el desarrollo del proyecto con el objetivo de obtener un dispositivo compacto, de alta flexibilidad y con un consumo energético óptimo. Luego, se describe la forma de crear un entorno de desarrollo apropiado para compilar y cargar una aplicación y finalmente se describe la plataforma de hardware construida y la funcionalidad obtenida.

II. GESTACIÓN DEL SISTEMA

Durante la etapa de diseño del sistema se revisó la disponibilidad de plataformas de hardware, protocolos de comunicación y existencia de sistemas operativos para obtener una plataforma integral, versátil, completamente funcional y eficiente, para resolver la problemática planteada. La selección de los componentes de hardware y software se define a partir de las especificaciones funcionales y los criterios de diseño y usualmente hay que hacer una negociación entre las diferentes condiciones para sintonizar el producto, ya que normalmente, la mejora de una de las características produce efectos negativos en otra, algunos de los casos más comunes son las relaciones consumo-rendimiento o tamaño-costo [1].

A continuación se describen los parámetros a partir de los cuales se realizó la selección de los diferentes componentes principales del sistema, además se incluye información relevante a partir de la cual se realizó la toma de decisiones.

Los elementos periféricos o accesorios se fueron decidiendo en las etapas de diseño y desarrollo, por lo cual, su elección se detalla en secciones posteriores.

A. Elección del sistema de comunicaciones

Para elegir la solución más adecuada para cierta aplicación específica hay que tener en cuenta ciertos parámetros que tendrán un peso relacionado directamente con el uso final de la implementación, entre los más populares se encuentran [2]:

- 1) El consumo (los transmisores son los mismos sólo se diferencian en cómo se usan).
- 2) El overhead que supone la arquitectura de comunicaciones (capas intermedias, longitud de cabeceras, necesidad de memoria y tiempos de CPU).
- 3) Prestaciones de QoS como tiempos de respuesta, anchos de banda, eficiencia, escalabilidad, protocolos de enrutamiento/enmallado disponibles, seguridad, etc.
- 4) La movilidad
- 5) La madurez tecnológica (si se encuentra o no soporte técnico) o incluso la interoperabilidad con otros estándares. Sobre todo si la solución es propietaria y no del IETF (The Internet Engineering Task Force). Puede ser que la tecnología esté madura pero que requiera una curva de adopción muy superior.

Además de las anteriores se debe verificar la cantidad de protocolos de comunicación que utilicen la misma frecuencia, ya que existen bandas saturadas en las que se dificulta la comunicación, las bandas ISM son las más comunes usualmente, ya que no requieren permisos especiales para su uso a baja potencia, siendo los 900MHz y en especial los 2.4GHz , las que más albergan protocolos.

También existen características inherentes a la frecuencia de operación, por ejemplo a medida que la frecuencia se incrementa:

- Se atenúa más fácilmente (no puede viajar tan fácil a través de obstáculos como copas de árboles, muros, etc) porque la onda es afectada en mayor medida por reflexiones y refracción. Por ejemplo, operar a 900 MHz aumenta el rango máximo de transmisión aproximadamente en dos veces comparado con la operación a 2.4 GHz .



- Capaz de transmitir mayor ancho de banda (mayor tasa de transmisión de datos)
- El tamaño de la antena y los otros componentes de RF disminuye, ya que depende de la longitud de onda que es inversamente proporcional a la frecuencia.
- El diseño electrónico se vuelve más complejo en la medida en el que los componentes discretos y los materiales utilizados cambian sus propiedades eléctricas.
- La Zona de Fresnel disminuye

En base a las características enunciadas previamente, se ha establecido la frecuencia de $900MHz$ como la más apropiada para transmitir en la aplicación propia, dentro de las principales opciones se encuentran:

- 1) Wireless m-bus
- 2) Dash7
- 3) WirelessHart
- 4) Zwave
- 5) SimpliciTI
- 6) Zigbee
- 7) 6LoWPAN

En los requerimientos principales para el diseño se tuvo en cuenta la posibilidad de conectarse a internet sin aumentar significativamente el *overhead* y disminuir el consumo energético.

B. Elección del sistema operativo

Se indagó acerca de los sistemas operativos para aplicaciones embebidas que soportaran 6LowPAN, dentro de las opciones, el más maduro y desplegado hasta el momento es The Contiki OS [3], que es un sistema operativo de código abierto, altamente portable, multitareas, para sistemas embebidos en red eficientes en memoria y redes de sensores inalámbricos.

Está diseñado para microcontroladores con pequeñas cantidades de memoria. Una configuración típica de The Contiki tiene 2KB de RAM y 40KB de ROM [4].

C. Elección del controlador

A partir del protocolo de comunicaciones y del sistema operativo también se debe escoger una plataforma de hardware compatible. El microcontrolador del sistema es el componente de hardware principal, ya que éste debe coordinar los periféricos para la adquisición de datos, realizar el procesamiento y enviar la información por medio del radio para ser visualizada o monitoreada remotamente. Sin embargo, el desempeño del sistema depende en igual grado de una configuración de firmware que optimice las características de hardware, las experiencias previas con redes de sensores inalámbricas (tmote sky, zolertia, etc) han utilizado microcontroladores de Texas Instruments por su bajo consumo, encapsulado reducido (QFN) incluyendo en algunos casos un transceiver como el cc2420 y por su costo competitivo; por lo tanto, existen desarrollos en firmware adelantados para este tipo de plataformas que podrían reducir el esfuerzo del área de diseño y desarrollo, además del tiempo

al mercado o *Time-to-market* [5] que representa una ventaja competitiva para el producto.

Se ha escogido el SoC CC430F5137 de Texas Instruments que tiene la arquitectura de la familia MSP430 y un radio CC1101, lo cual permite desarrollar aplicaciones de muy bajo consumo (modo activo $160\mu A/MHz$, modo de espera $2.0\mu A$, modo apagado con retención de RAM $1.0\mu A$) y gracias al periférico de mapeo de puertos, se puede alcanzar gran flexibilidad en el diseño del hardware, porque se pueden utilizar el puerto 1 o el puerto 2 con cualquier periférico mapeable (I2C, SPI, UART, A/D, etc) lo que permite obtener también, un dispositivo de tamaño reducido.

III. THE CONTIKI OS

Contiki consiste en un *kernel* manejado por eventos sobre el cual las aplicaciones son cargadas y descargadas dinámicamente en tiempo de ejecución. Los procesos de *Contiki* usan *protothreads* livianos que proveen un estilo lineal similar a los hilos sobre el *kernel* manejado por eventos. Además, tiene compatibilidad para desarrollar aplicaciones con IPv6.

A. Características de The Contiki

Parámetros de *The Contiki*:

- Preemption opcional por aplicación
- Comunicación interprocesos usando paso de mensajes a través de eventos
- Subsistema GUI opcional local o remotamente
- Kernel multitareas
- Protothreads
- Creación de redes TCP/IP incluyendo IPv6
- Navegador WEB
- Servidor WEB personal
- Cliente telnet simple

IV. ADECUACIÓN DEL ENTORNO DE DESARROLLO

Para desarrollar firmware para The Contiki y compilarlo, se requiere una cadena de herramientas apropiada, existen varias opciones disponibles para lograr el acometido.

En la web se encuentran diversidad de links que brindan información básica con la manera de instalar *Instant Contiki*, que es un disco virtual para VMware que contiene un Ubuntu OS preconfigurado para The Contiki.

Instant Contiki es un solo archivo que contiene todas las herramientas necesarias para el desarrollo de software con The Contiki, que son: El conjunto de herramientas compiladoras, el simulador de red de The Contiki *Cooja*, el simulador de red *netsim*, y programas para cargar un sistema compilado The Contiki a tarjetas *Tmote Sky* conectadas.

Lamentablemente la versión disponible de *Instant Contiki* no contiene algunas utilidades fundamentales para el caso propio, como la versión adecuada del *mspgcc* o del *mspgdbproxy*.



(o en su defecto mspdebug para depurar) para el microcontrolador cc430, ni los drivers del MSP-FET430UIF necesarios para descargar el *firmware* en el cc430. Por lo tanto se decidió crear una máquina virtual propia llamada *NetuxContiki* que se construyó a partir de Ubuntu con los pasos que se indican a continuación:

1) *Instalación de Ubuntu*: Se decidió instalar Ubuntu 10.10 LTS nativamente en el computador personal para evitar otros posibles problemas asociados al funcionamiento de algunos periféricos del mismo usados por las herramientas de depuración cuando se utilizan máquinas virtuales.

2) *Instalación de paquetes adicionales*: Antes de proseguir se debe contar con los siguientes paquetes [6]:

- subversion
- texinfo
- patch
- libncurses5-dev
- zlib
- zlib1g-dev
- libx11-dev
- libusb-dev
- libreadline6-dev

3) *Instalación del MSPGCC4*: La versión incluida del compilador en *InstantContiki* no soporta al CC430, por lo que se debe adecuar para poder compilar y luego cargar el firmware.

Se ha diseñado un parche que incorporándose al mspgcc 3.2.3 daría soporte para los SoC. A partir de allí se integraron los parches de binutilus, gcc4 y msp430-libc a el último *release* del mspgcc4.

Los paquetes para *Debian* se pueden descargar de <ftp://ftp.openosian.net/mspgcc4/> hay tres que necesitan ser instalados: GCC (el compilador), libc (librería del compilador) y binutilus (el ensamblador y linker).

4) *Herramientas adicionales necesarias*: Es necesario instalar un bootstrap loader para cargar los archivos binarios del computador al MSP430 mediante el FET (Flash Emulation Tool). Se decidió compilar y cargar el código con msp430-gdb y mspdebug. Básicamente se compila el programa con el comando *make* del mspgcc y luego se carga y ejecuta el programa con el *mspdebug* de la siguiente manera:

```
make clean
make TARGET=netux430 savetarget
make application.ihex
mspdebug -d /dev/ttyUSB0 -j uif
(mspdebug)prog /pathdelarchivoejecutable/
application.ihex
(mspdebug)run
```

V. PORTE DEL SISTEMA OPERATIVO A LA PLATAFORMA DE HARDWARE

Para compilar una aplicación con *The Contiki* se requieren tres archivos [7]: El archivo de aplicación que contiene la

implementación, el archivo de definición de la aplicación (si se necesita) y el *Makefile* que permite compilar la aplicación a partir de código fuente especificando un conjunto de reglas y rutas a archivos necesarios en la aplicación.

A. Diseñar los drivers de la plataforma de hardware

Los divers son el medio por el cual el sistema operativo (*The Contiki*) puede interactuar con las herramientas de hardware. Se deben crear entonces archivos en *c* con sus respectivas cabeceras que contengan las funciones generales para el manejo de los diferentes periféricos.

Una manera apropiada para portar el sistema operativo es partir de una plataforma “nativa” y modificar los archivos y directorios de ésta plataforma adecuadamente [8]. Para la aplicación específica se van a utilizar LEDs, UART, el radio, I2C, ADC, etc.

Para portar una nueva plataforma hay básicamente dos directorios relevantes [8]: /platform y /cpu. Se debe crear un subdirectorio en el directorio /platform y mover los archivos de /platform/native ahí.

Se procede a crear un subdirectorio llamado “netux430” (plataforma de hardware) dentro de la carpeta /platform y se crean allí los archivos necesarios para tener una plataforma completamente funcional a partir de las carpetas /platform/icwcs [8] y /platform/z1 de Contiki2.x. En el archivo /platform/netux430/contiki_conf.h se definen las opciones específicas de configuración, tipos C, compilador, configuración, configuración del reloj, configuración de uIP, direccionamiento de memoria de bajo nivel, configuración de pines, etc.

Además se debe crear una nueva cpu (/cpu/cc430) utilizando los archivos de /cpu/msp430 para adecuarlos a los requerimientos propios, por ejemplo: Cambiar el archivo button.c para escoger el puerto adecuado, modificar el archivo flash.c para escoger el microcontrolador indicado (se debe añadir el archivo cc430f5137.h a /usr/msp430/include).

Después de esto se crean los makefiles de la nueva plataforma y de la CPU, además del makefile específico para cada aplicación:

B. Modificar los Makefiles

Se debe entonces modificar el *Makefile* (dentro de la carpeta de la aplicación específica) que incluye al *Makefile.include* para que se incluyan los archivos necesarios al proyecto, luego se debe modificar el *Makefile.target* (dentro de la carpeta de la plataforma) incluyendo los drivers de la plataforma de hardware específica y el tipo de microcontrolador (se incluye el *Makefile.cpu* que define los parámetros de compilación propios del microcontrolador).



El archivo `/platform/netux430/Makefile.netux430`, especifica qué archivos se van a incluir en el archivo por defecto `Makefile.include`, el cual incluye todas las otras opciones que se necesitan.

El otro *makefile* que se debe tener en cuenta es el específico de la CPU localizado en el directorio `/cpu/cc430` que toma en cuenta las reglas y definiciones del *cross-compiler* y especifica la ruta dónde buscar los archivos fuente.

Finalmente para correr *The Contiki* el archivo `contiki-netux430-main.c` se debe crear. La función principal de este archivo inicializa el hardware y luego programa y corre los procesos definidos.

VI. ESPECIFICACIONES DEL SISTEMA

A. Funcionalidad

Se requiere construir un nodo que cuente con las siguientes interfaces:

- ADC
- I2C
- Puertos de entrada salida de propósito general
- Módulo de captura de pulsos
- Módulo SPI
- Alimentación

De esa manera se tiene gran flexibilidad y se es compatible con la mayor parte de los sensores electrónicos del mercado, la primera aplicación para verificar el funcionamiento del dispositivo incluye una red capaz de sensar las siguientes variables:

- 1) Corriente
- 2) Temperatura
- 3) % Humedad
- 4) Contactos secos como pulsadores y puertas

Para cada una de los tipos de sensores se ha diseñado un driver específico, que permite utilizar un método de desbloqueo de tareas por mensajes enviados por los drivers que son activados cuando ocurren interrupciones asociadas a los periféricos. Esta implementación permite ahorrar energía en procesamiento pues el sistema operativo envía el microcontrolador a un modo de bajo consumo hasta que se desbloquee una tarea o “hilo” determinado.

B. Restricciones

- Alimentado por baterías de 3V
- Antena en cable
- Ajustar el tamaño de la tarjeta a una de las cajas disponible
- Diseño en 4 capas
- Encapsulado para componentes misceláneos 0603

C. Criterios de diseño

Diseñar un dispositivo portable, de bajo consumo, flexible en cuanto a variables a medir y de bajo costo de producción.

VII. DISEÑO DEL SISTEMA

A. Generalidades de la tarjeta

Las dimensiones son 35x35x10 mm. Se han utilizado 4 capas, de las cuales las capas intermedias corresponden al plano de tierra y el plano de alimentación.

B. Diseño de la antena y la etapa de radio

Se ha utilizado una antena tipo cable, que es equivalente a una tipo monopolo ($\lambda/4$), se debe tener en cuenta que en este tipo de antenas, el rendimiento es dependiente del tamaño de la tierra que es la encargada de completar el dipolo [9].

El cálculo para determinar la longitud de onda se realiza de la siguiente manera:

$$\lambda = \frac{2.99792458E8m/sec}{f(GHz)} \quad (1)$$

Teniendo en cuenta que la frecuencia de operación es de $915MHz$, entonces $\lambda/4 = 8.2cm$

Nota: La sintonización de la antena se realiza en el entorno en el que va a tener su funcionamiento [9], en el documento de referencia para el diseño electrónico [10], se plantean algunas buenas prácticas para el trazado del filtro y Balun:

- Las pistas para el trazado de RF deben ser simétricas
- Un plano de tierra sólido debe ser implementado bajo la circuitería de RF
- Se recomienda que la distancia entre la capa 1, con la circuitería de RF, y la tierra sea de $410\mu m$. Cambiar el grosor de la tarjeta cambia la inductancia de las vías que, en serie con el condensador de desacople, podría afectar negativamente el rendimiento.
- El diseño se debería implementar en un sustrato FR4. Cambiar el sustrato afectará la impedancia de las pistas del PCB.
- Los condensadores de desacople deben ser usados y se deben ubicar lo más cerca de los pines posible. Se deben colocar vías cerca de los condensadores de desacople para asegurar buena conexión al plano de tierra debajo.

C. Selección de la alimentación

Para la alimentación se ha elegido una batería tipo moneda de 20mm de 3V fabricada por FDK America INC referencia CR2032. Las especificaciones de la batería se muestran en la siguiente tabla:

Voltaje nominal (V)	3
Capacidad Nominal (mAh)	240
Corriente de descarga estándar (mA)	0.3
Máxima descarga continua (mA)	4
Máxima descarga pulsátil (mA)	20

También se ha colocado un condensador de reserva en paralelo con la batería, para ayudarle a suministrar la corriente necesaria cuando ocurren demandas fuertes, como en el caso de transmisión de información a través del radio.



D. Diseño del conector de los periféricos

Se ha hecho una búsqueda exhaustiva de conectores teniendo en cuenta algunos de los fabricantes más importantes como Molex, ERNI, hirose, FCI, JAE Electronics, entre otros.

Los criterios de selección fueron:

- Tamaño
- Cantidad de pines
- Robustez de la fijación

La opción elegida fue usar conectores microUSB que son más baratos con buena fijación, estándares y pequeños.

La interfaz de conexión con periféricos externos se ha diseñado de tal manera que se agreguen o quiten componentes dependiendo si se va a utilizar SPI, I2C, entradas analógicas, etc.

Se deberán hacer entonces dos ajustes, dependiendo del caso:

- Ajuste de software: Utilizando el módulo portmap mapear los pines de entrada de los periféricos a los pines de entrada correspondientes del puerto microUSB.
- Ajuste de hardware: De acuerdo al tipo de periférico realizar algunas modificaciones en la etapa de ensamble predefinidas desde el diseño.

E. Elección de componentes varios

Para la elección del cristal oscilador de 26MHz se hizo una comparación entre el sugerido por Texas (ASX 531 de Aker - Reemplazado por ASX5 ANSEN) y los diversos osciladores que se consiguen en Digikey.

Se decidió utilizar uno fabricado por NDK porque cuenta con características eléctricas similares, hay buena cantidad en stock y es más compacto.

F. Apariencia de la tarjeta electrónica

En las figuras 1 y 2 se muestra el dispositivo visto desde su parte superior e inferior respectivamente.

VIII. CONCLUSIONES

En la actualidad se han hecho diversidad de pruebas con la tarjeta electrónica obteniéndose resultados en todas las etapas, se ha medido el patrón de radiación usando *SmartRF Studio* de Texas Instruments y se han comparado las pérdidas con el módulo experimental que contiene el mismo microcontrolador usado y se ha llegado a la conclusión de que se tiene un muy buen desempeño, alcanzándose distancias de radiación con obstáculos de hasta una cuadra.

Se ha construido una red ad-hoc enviando datos con el driver diseñado y utilizando The Contiki OS satisfactoriamente.

En la primera versión de la aplicación se construyó una tarjeta hija con un módem celular que se comunica por un puerto serial con el coordinador de la red y permite enviar la información centralizada a una aplicación basada en la nube, en donde se pueden visualizar las diversas variables.

En la actualidad se está trabajando en el despliegue de las capas superiores del modelo OSI para darle conectividad a

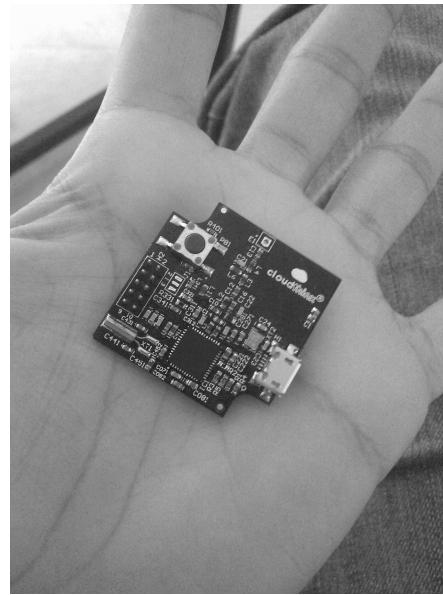


Fig. 1. Vista superior de Cloudthinx

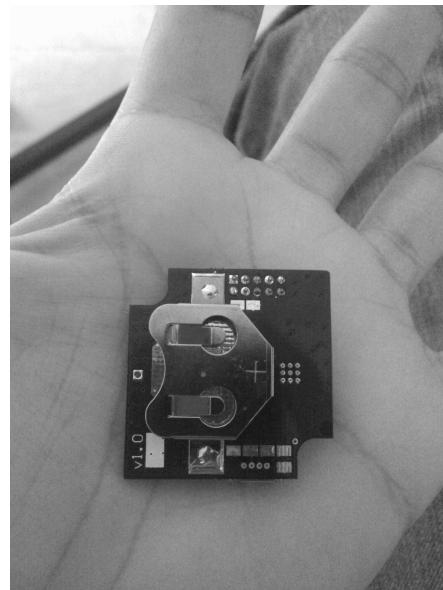


Fig. 2. Vista inferior de Cloudthinx

internet a cada módulo y realizar el monitoreo desde una plataforma web.

REFERENCES

- [1] P. A. SANDBORN and M. VERTAL, "Analyzing packaging trade-offs during system design," *IEEE DESIGN & TEST OF COMPUTERS*, pp. 10–19, 1998. [Online]. Available: <http://enme.umd.edu/ESCML/Papers/ddt.pdf>
- [2] G. Ravera, "Zigbee o 6lowpan." [Online]. Available: <http://blogs.salleurl.edu/networking-and-internet-technologies/2010/04/14/zigbee-o-6lowpan/>



- [3] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," *Swedish Institute of Computer Science*. [Online]. Available: <http://www.sics.se/~adam/dunkels04contiki.pdf>
- [4] A. Dunkels, "The contiki os the operating system for the internet of things," Tech. Rep. [Online]. Available: <http://www.contiki-os.org/p/about-contiki.html>
- [5] M. A. Cohen, J. Eliashberg, and T.-H. Ho, "New product development: The performance and time-to-market tradeoff," *Management Science*, vol. 42, pp. 173–178, 1996. [Online]. Available: <http://www.jstor.org/discover/10.2307/26329997?uid=3737808&uid=2&uid=4&sid=56254411973>
- [6] A. Aguilar, *Ubuntu Tutorial: MSPGCC and Eclipse*, August 2010. [Online]. Available: http://www.43oh.com/wp-content/uploads/2010/11/MSP430_MSPGCC_Eclipse_gdb.pdf
- [7] Zolertia. [Online]. Available: http://zolertia.sourceforge.net/wiki/index.php/Mainpage:Contiki_Lesson_0
- [8] C. Z. TAYSI, "Contiki port for msp-exp430f5438," Yildiz Teknik Universitesi, Tech. Rep., 2010. [Online]. Available: <http://www.yildiz.edu.tr/~cihan/contiki.php>
- [9] R. Wallace, "Antenna selection guide, application note an058," Texas Instruments, Tech. Rep., 2010.
- [10] M. Morales, "Em430f6137rf900 reference design guide," TEXAS INSTRUMENTS, Tech. Rep., 2010.



“Embedding a bridge Internet to X-10 and a web server in microcontrollers”

Martin Perez, Pablo Agüero, Alejandro Uriz, Juan Carlos Bonadero, Juan Carlos Tulli and
Esteban Gonzalez
Facultad de Ingeniería
Universidad Nacional de Mar del Plata
Mar del Plata, Argentina

Domotics contributes to enhance life quality, making the house a more functional place. The proliferation of wired and wireless communications imposes a new direction for domotics: enabling remote home actions. X-10 is a communication protocol designed for sending signaling and control over 220 VAC wiring for plug-in modules available from various vendors, by using the AC wiring already installed within a home.

This paper shows the implementation using Microchip’s microcontroller of an interface between X-10 communication protocol and Ethernet protocol to let a person command his home using any device with TPC/IP protocol, such as cell phones.

The embedded web server is running into a microcontroller device, and web pages are sent to the user to show all the available X-10 managing options. This web server may be connected to an ADSL router and may be accessed from any part of the world using a public IP. The proposed embedded web server is implemented using PIC24HJ128GP502 and the Microchip TCP/IP Stack, a suite of programs that provides services to standard TCP/IP-based applications.

The X-10 transceiver is also implemented in a PIC24H microcontroller. This transceiver makes the conversion of the orders sent by the embedded web server into X-10 messages. The communication between the X-10 transceiver and the embedded web server is performed using a UART. This communication was chosen due to the low requirements in complexity and speed of the application.

“Diseño de HMI WVGA táctil con comunicación Modbus TCP”

Muro, Gustavo¹, Ing. Gentile, Aldo²

Facultad de Ciencias Exactas, Ingeniería y Agrimensura - UNR
Rosario - Argentina

¹ gmuro@fceia.unr.edu.ar, ² agent@eie.fceia.unr.edu.ar

En múltiples aplicaciones de control industrial es necesario conocer y modificar el estado de variables relacionadas con el proceso controlado.

La HMI diseñada es capaz de mostrar información y actuar sobre un controlador a través de un protocolo de comunicación estándar (Modbus TCP) [1], ampliamente utilizado en la industria. La pantalla utilizada es de tipo TFT [2] con una resolución de 800x480 píxeles, 65535 colores, un tamaño de 7 pulgadas, con touch screen resistivo [3]. La misma incorpora un controlador gráfico sin aceleración por hardware (SSD1963). El diseño comienza desde el circuito impreso hasta el desarrollo del software. En la Figura 1 se muestra el diagrama en bloques simplificado del circuito impreso diseñado.

Implementa las funciones de lectura y escritura del protocolo de comunicación Modbus Master y el transporte de los mensajes se realiza por medio de Socket TCP disponibles en la librería TCP/IP de Microchip, cumpliendo de esta manera con el estándar Modbus TCP. Para realizar el despliegue de objetos gráficos se utilizó el paquete de software “Microchip Graphics Library”.

La librería gráfica, el Stack TCP/IP, el protocolo Modbus y la aplicación se ejecutan bajo la gestión de un sistema operativo de tiempo real, el mismo es el FreeRTOS [4] según se muestra en la Figura 2. Además, se integró al sistema la librería FAT32 [5] para el manejo de archivos en una tarjeta de memoria SD y de esta manera poder acceder a imágenes BMP/JPEG para ser mostradas en la pantalla. El software se ejecuta sobre un microcontrolador de 32 bits de 80Mhz con 512KB de Flash y 32 KB de RAM (PIC32MX340F512H) [6].

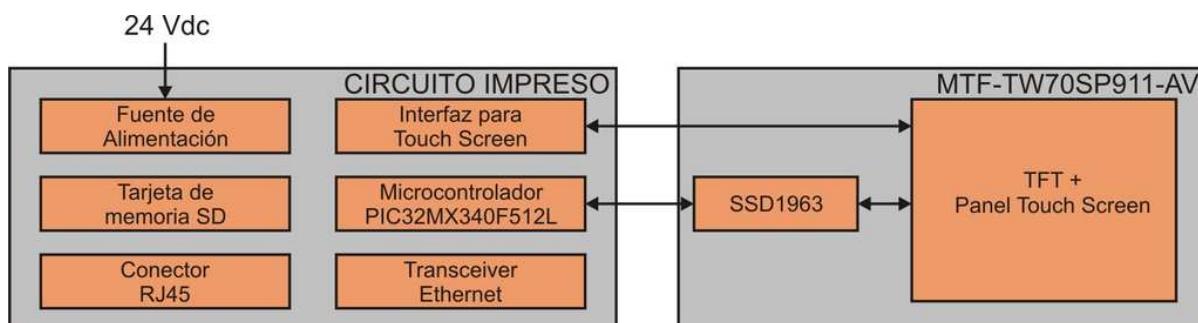


Figura 1: Diagrama en bloques simplificado del circuito impreso y la conexión con el display

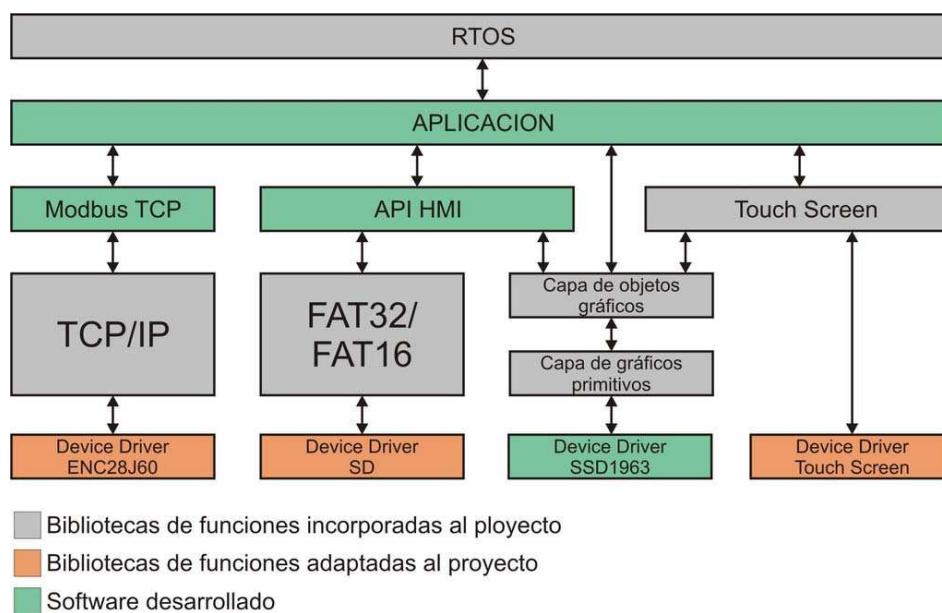


Figura 2: Estructura general del software de la HMI

En la Figura 3 se puede observar el prototipo mostrando la pantalla de configuración de la HMI.

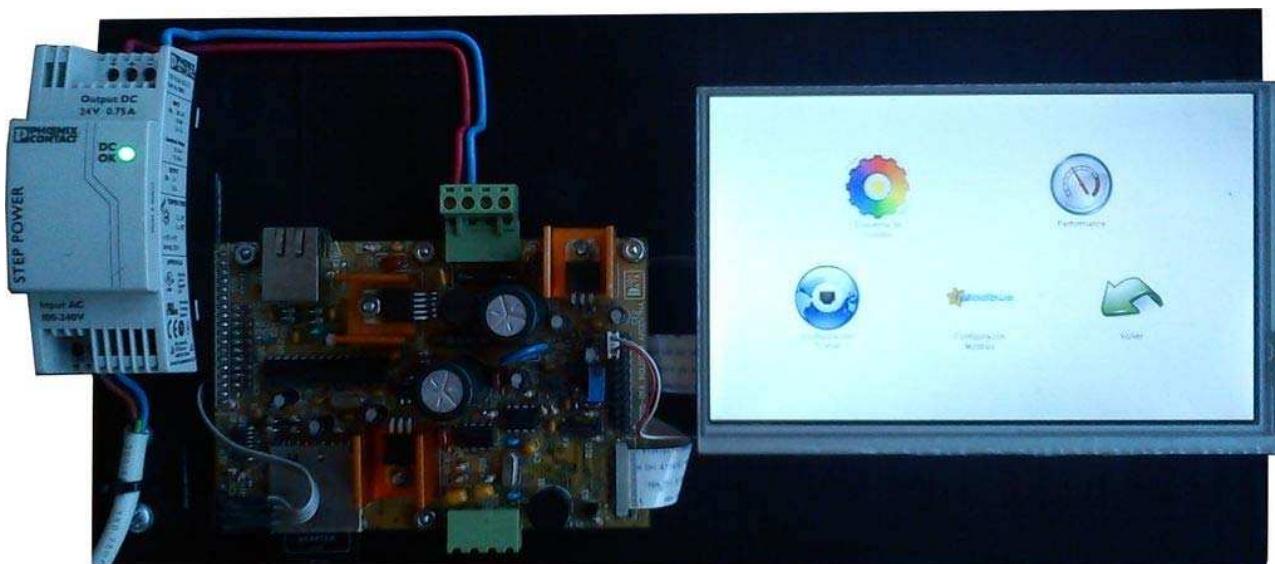


Figura 3: HMI: Fuente de alimentación de 24Vcc, la placa diseñada que aloja microcontrolador, comunicación Ethernet y tarjeta de memoria SD, el display TFT.

Bibliography

- [1] Diego M.Romero, "Introducción a Ethernet Industrial," *Schneider Electric, IEEE*, 2005.
- [2] Takashi Unagami, "EL Characteristics of a TFEL/TFT Stacked Structure Display Device Driven by a HV-Si TFT Circuit," *IEEE TRANSACTIONS ON ELECTRON DEVICES*, vol. 47, no. 9 Sept.2000.
- [3] R.N.Aguilar and G.C.M.Meijer, "Fast Interface Electronics for a Resistive Touch-Screen," *IEEE*, pp. 1360-1363, 2002.
- [4] K.Andersson and R.Andersson, "A comparison between FreeRTOS and RTLinux in embedded real-time systems," <http://www.streambag.se/files/rtproj.pdf>, 2005.
- [5] Moonsoo Choi, Heemin Park, and Jaewook Jeon, "Design and Implementation of a FAT File System for Reduced Cluster Switching Overhead," *International Conference on Multimedia and Ubiquitous Engineering*, pp. 355-360, 2008.
- [6] Darren Wenn, "Integrating Microchip Libraries with a Real-Time Operating System," *Microchip Technology Inc.*, 2009.



“Diseño de un TAG RFID integrado en un proceso CMOS de $0,5 \mu\text{m}$ ”

Alcalde Bessia, F.; Fanego, D.; Makar, G.

f@lcald.com.ar; dfanego@gmail.com; guillermomakar@gmail.com

Seminario de diseño y fabricación de circuitos integrados en tecnología CMOS

Departamento de electrónica, Facultad de ingeniería, Universidad de Buenos Aires

En los últimos años los dispositivos de identificación automática se han hecho muy populares en fábricas, industrias de servicios y logística, ya que brindan información acerca de personas, animales y bienes de forma rápida y eficaz. En muchas de estas aplicaciones es necesario que los dispositivos usados sean simples, robustos y de bajo costo.

En el presente trabajo se realizó el diseño de un prototipo integrado de un TAG de Identificación por Radio Frecuencia (RFID) implementado completamente en un proceso CMOS. El dispositivo obtiene la energía necesaria para su funcionamiento a partir de la señal de RF que provee el lector, captada mediante un inductor externo al circuito integrado, y envía un código de identificación (ID), sin necesidad de contar con baterías, como es habitual en estos TAGs. El circuito diseñado y enviado a fabricar comprende módulos de rectificación y regulación de la tensión de entrada, una memoria ROM de 16 bytes y una máquina de estados para sincronizar los distintos bloques. La memoria, junto con sus decodificadores, se implementaron con celdas de lógica dinámica, lo que permitió reducir considerablemente el tamaño, consumo y complejidad del diseño.

El prototipo fue fabricado con el proceso CMOS de ON Semiconductor de $0,5 \mu\text{m}$, al que se accede a través del consorcio MOSIS. La verificación de funcionamiento del circuito presentó resultados satisfactorios, similares a los obtenidos por simulación. Varios de los bloques que componen el circuito pueden ser utilizados independientemente, facilitando así tanto las mediciones como su utilización en futuros proyectos de nuestra Facultad.



“Desarrollo de un sistema de adquisición y procesamiento de señales ECG para la monitorización de pacientes de alto riesgo”

Dell'Aquila C.; Cañadas G.; Segura M.; Sisterna C.; Laciár E.

Departamento de Electrónica y Automática

Universidad Nacional de San Juan

{carlos.dellaquila, gcanadas, msegura, cristian}@unsj.edu.ar,

laciár@gateme.unsj.edu.ar

En este trabajo se ha desarrollado un sistema capaz de adquirir y procesar señales electrocardiográficas (ECG) en personas con afecciones coronarias.

Debido a su baja amplitud (1-3 mV), la señal ECG debe ser acondicionada antes de realizar su conversión analógica a digital. Para ello se ha utilizado el Front-End ADS1298 de la firma Texas Instruments, que permite acondicionar hasta 8 canales de ECG y muestrearlos a una frecuencia de 500 Hz. La ventaja de este chip es que permite integrar la etapa de amplificación de la señal ECG con un conversor A/D delta-sigma de 24 bits de resolución. Debido a la alta resolución del último no es necesario que utilizar ganancias elevadas en la etapa de amplificación. Otra ventaja del ADS1298 es su bajo consumo del orden del 0.75 mW por canal.

La etapa de procesamiento se diseñó con un microcontrolador MSP430 de 16 bits de Texas Instruments. El microcontrolador elegido permite diferentes modos de operación logrando un consumo muy reducido de potencia (<0.23 mW), aspecto que es muy importante para el desarrollo de un sistema portátil alimentado con baterías.

El sistema desarrollado ha sido probado en 3 sujetos masculinos y con señales ECG con diferentes arritmias generadas por el simulador multiparamétrico FLUKE PS420. En ambos casos se registraron simultáneamente las señales con un equipo convencional de ECG (de la firma Ecossur) para validar el sistema desarrollado. Los resultados obtenidos indican que existe una total concordancia entre las señales cardíacas registradas por el sistema desarrollado y por el equipo comercial de ECG. En un futuro se prevé su implementación en un sistema portátil que permita la monitorización de pacientes cardíacos de algo riesgo.



“Automedida de consumo en sistemas embebidos”

Oreggioni J.; Fernández S.; Steinfeld L.

IIE, Facultad de Ingeniería, Universidad de la República

Montevideo, Uruguay

{juliano, sebfer, leo}@fing.edu.uy

El dotar de la capacidad de conocer el consumo en tiempo real a un nodo que forma parte de una red de sensores inalámbricos, permite por ejemplo incorporar en la lógica de operación, decisiones que dependen del consumo efectivo del nodo: modificar la frecuencia de muestreo de sensores, ajustar el ciclo de trabajo de una radio o afectar el ruteo de los paquetes.

En el presente trabajo se busca encontrar una solución que cumpla con las siguientes características: bajo consumo, debe utilizar componentes de bajo consumo y/o bajo ciclo de trabajo; poco invasivo, debe utilizar el menor tiempo de procesador posible de forma de no afectar el funcionamiento de la aplicación y que el consumo asociado no afecte en forma significativa la medida; modular y con drivers, debe abstraerse de las particularidades del Hardware; fácil adopción, se busca que sea fácilmente incorporado a sistemas existentes.

Se propone utilizar el método de medida de consumo basado en la descarga de capacitores similar al planteado en [1] para medir el consumo de terceros, pero modificado para utilizarlo en forma de auto-medida y aumentar su rango dinámico de medición. El principio de funcionamiento radica en alimentar el sistema sobre el cual se desean realizar las medidas desde dos capacitores, en lugar de alimentarse directamente de una fuente de voltaje. Durante un ciclo de trabajo, el sistema consume carga de un capacitor, mientras el otro capacitor se carga. En el ciclo siguiente de trabajo se intercambian los capacitores, pasando el primero a cargarse y el segundo a alimentar el sistema. La alimentación proporcionada por los capacitores, está monitoreada todo el tiempo por un comparador contra un voltaje de umbral. Una vez que el voltaje cae hasta ese valor umbral, se produce una interrupción y el módulo de software intercambia los capacitores e incrementa la cuenta de capacitores. Para conocer el consumo en un intervalo de tiempo dado, basta con tener el número de veces que se intercambiaron los capacitores. Este método de medida cuenta descargas de capacitores completas, con lo cual el error de cuantización, dependiendo de los parámetros del sistema y de la aplicación en particular, puede llegar a ser alto. Para mejorar la precisión de la medida se propone un método complementario, que consiste en agregar un conversor AD que permita medir el voltaje de los capacitores en cualquier momento.

La solución se implementó y probó en un sistema basado en el microcontrolador MSP430F2274 de Texas Instruments, realizando pruebas exitosas en configuración de automedida y midiendo el consumo a otro dispositivo, logrando validar su funcionamiento.

La descarga de capacitores como método de medida plantea una ventaja importante: no es necesario estar continuamente midiendo el voltaje que cae sobre un resistor, para integrarlo después y obtener el valor de la carga consumida. Desde este punto de vista, la descarga de capacitores involucra menos recursos y menos consumo. Por otra parte, presenta dos desventajas que dependiendo de la aplicación no son tales. En primer lugar genera un rizado en la fuente. En segundo lugar, el sistema de medida se pone en serie entre la fuente y la aplicación, con lo cual, cualquier mal funcionamiento del sistema de medida puede resultar crítico para la aplicación.

[1] J. Andersen and M. Hansen, “Energy bucket: A tool for power profiling and debugging of sensor nodes,” in Third International Conference on Sensor Technologies and Applications, 2009.



“Medición remota del contenido de agua en el suelo mediante sensores capacitivos”

Askenazi, Javier^{†*}; De Marziani[#], Carlos; Scholz, Fabián^{†*}

(javier.askenazi@gmail.com ; marziani@unpata.edu.ar ; fgscholz@yahoo.com)

† CONICET, Comisión Nacional de Investigaciones Científicas y Técnicas

* GEBEF, Grupo de Estudios Biológicos y Eco-fisiológicos, Departamento de Biología, Facultad de Ciencias Naturales, Universidad Nacional de la Patagonia San Juan Bosco

#Departamento de Electrónica, Facultad de Ingeniería, Universidad Nacional de la Patagonia San Juan Bosco

Determinar el contenido de agua en el suelo es de especial interés en diferentes campos relacionados con la eco-fisiología vegetal y la agricultura entre otras actividades. Su conocimiento es de importancia para la realización de balances hídricos o la implementación de sistemas automáticos de riego en cultivos. Existen varios métodos para medirlo: reflectometría en el dominio del tiempo (TDR), sensor en el dominio de la frecuencia, sondas capacitivas, entre otras.

Las sondas comerciales presentan elevado costo, lo cual es impedimento para su utilización en el monitoreo de grandes áreas donde se necesita un alto número de réplicas. Por otro lado resulta beneficioso realizar el monitoreo de variables de manera inalámbrica ya que se puede ampliar la zona de cobertura eliminando el gran cableado que de otra forma sería necesario.

En este trabajo se presenta el diseño de una sonda de bajo costo a partir de principios capacitivos para la medición de contenido volumétrico de agua en el suelo. La sonda prototipo se realizó sobre dos placas para circuitos impresos, con una capa intermedia de material conductor (cobre). Las señales resultantes son procesadas mediante dispositivos analógicos-digitales reconfigurables, que permiten realizar el acondicionamiento y procesamiento de señales con un mismo circuito integrado sin necesidad de incrementar excesivamente el hardware a implementar. Finalmente los datos son transmitidos a un sistema central a través de módulos de RF Zigbee. Las pruebas experimentales realizadas en laboratorio demuestran que la arquitectura propuesta permite obtener una adecuada precisión en la determinación del contenido de agua en el suelo.

Diseño de un sistema de gestión y comunicación para mediciones de variables en procesos remotos y locales usando un sistema de desarrollo.

Melo Sevilla Luis Fernando
Universidad Santiago de Cali
Cali, Colombia
fdomelo@hotmail.com

Melo Sevilla Raúl Emilio
Facultad de Ingenierías
Universidad Santiago de Cali
Cali, Colombia
raulmelo@hotmail.com

Esta investigación pretende realizar un desarrollo de un sistema de captura, trasmisión, gestión e integración de datos con una arquitectura abierta, que permita a las empresas dar el uso que requieran a dicha información, ver figura 1.

El sistema desarrollado usa Arduinoⁱ por ser un sistema de arquitectura abierta y fácil de conseguir junto con otros módulos tales como Ethernet y GPRS.

Los datos son enviados usando el protocolo HTTP, por medio de un módulo GPRS y/o Ethernet y se garantiza la integridad y seguridad por medio de CRC 16(Código de redundancia cíclica), dentro de la trama a la que se le calcula el CRC se le añade una palabra clave que no se trasmite y es conocida únicamente por el receptor para garantizar que el que envía si es quien dice ser.

En la industria de servicios públicos domiciliarios de acueducto y alcantarillado de muchas ciudades de Colombia tienen a la vez una gran necesidad y un gran rezago en el cumplimiento de la ley, ya que deben garantizar una macro-medición del orden del 95% y solo se está cumpliendo en un 42%.

Con este trabajo se pretende solucionar algunas de las actuales limitaciones que se tienen en el país en la implementación de sistemas de captura de valores de variables de procesos remotos, muchas de estas tecnologías desarrolladas para la captura de datos, son propietarias y cerradas y no permiten

integrar de forma automática esta información con los sistemas de información de las empresas

La solución de software se desarrolló en la nube <http://www.lufemes.com/sensor/01> en PHP y mysql, permitiendo ser accedida desde cualquier parte del mundo y permitiendo acceder a los datos directamente por base de datos o la aplicación web.

Este software en la nube permite manejar el inventario de dispositivos de comunicación, de sensores, recibe los datos, los procesa, almacena, permitiendo monitorear, gestionar y compartir la información. Adicionalmente envía alarmas por medio de mensajes a celular usando un módulo Arduino/GPRS desarrollado para este fin. Cumpliendo el objetivo de este trabajo.

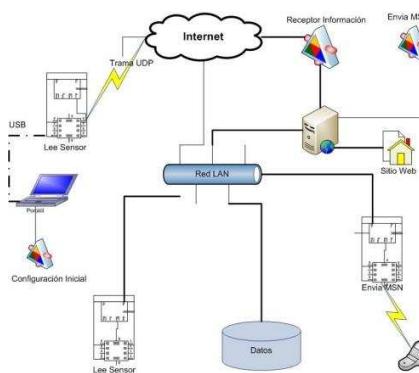


Figura 1 Arquitectura del sistema

I. Referencias

ⁱ Página oficial de Arduino, <http://arduino.cc/es/>



“Aplicaciones de control de acceso biométrico con sistemas embebidos de baja gama”

Caamaño B. Mauricio; Chan Z. Carmen; Román M. Frank; Carvajal G. Johan.

Tecnológico de Costa Rica
Cartago, Costa Rica

maucaamano@gmail.com, ckc004@gmail.com, fromanlda@gmail.com, johan.carvajal.g@gmail.com

El presente trabajo consta de un sistema de seguridad que emplea la autenticación biométrica de un usuario para el control de acceso hacia un lugar. El método de identificación utilizado por el sistema consta de insertar una clave personal sólo conocida por el usuario y la colocación del dedo de éste para obtener su huella dactilar.

Con base en este sistema, se diseñaron dos aplicaciones de control de acceso. La primera consiste en restringir el paso de usuarios no registrados hacia un laboratorio del Tecnológico de Costa Rica y la segunda consiste en permitir o no el acceso de usuarios hacia el uso de unas bicicletas eléctricas de una estación.

El sistema se implementó en una tarjeta de desarrollo, la cual cuenta con un microcontrolador de la serie PIC32MX. Se conecta el sensor vía USB con el microcontrolador. El sensor dactilar es un dispositivo de la clase USB Diagnostic Class, por lo cual se diseñó un controlador de esta clase para el microcontrolador. Además, se implementó una base de datos con la información de los usuarios, para la cual se escribieron rutinas de manejo de memoria Flash y EEPROM. Por último, en el diseño se emplea una pantalla táctil que es parte de la tarjeta de desarrollo que despliega las instrucciones a los usuarios así como también funciona como un teclado para la introducción de la clave personal.

Para el manejo de los módulos mencionados anteriormente, se implementa en el microcontrolador un RTOS desarrollado por estudiantes del Tecnológico de Costa Rica. La implementación del RTOS facilita el manejo de los módulos a nivel de programación.

“CONTROL INALÁMBRICO DE UN SISTEMA MÓVIL MEDIANTE UNA PANTALLA TÁCTIL Y LA RED GSM CELULAR”

Juan José Carazo Céspedes
juanjosecarazo@gmail.com

Mauricio Caamaño Bedoya
maucaamano@gmail.com

Miguel Rodríguez Escudé
mrescude@gmail.com

Profesor Asesor Johan Carvajal Godínez
johcarvajal@itcr.ac.cr

El proyecto consiste en el control de un dispositivo móvil, mediante el protocolo MiWi utilizando transceptores de RF y la red celular GSM. La implementación consta de un carro a escala, y pretende demostrar, mediante la integración de tecnologías de redes de sensores, la gran disponibilidad y facilidad que estas ofrecen en áreas como domótica, monitorización en sistemas ambientales y control de infraestructura e industrias.

Como sistema de entrada de datos (nodo central) se utilizó una pantalla táctil y un módulo para la recepción de SMS. Para el primer caso se desarrolló un panel de navegación, el cual permite al usuario acceder por separado a las ventanas respectivas para control de motores, luces y música. Para la conexión con la red celular se utilizó un DataCard y los comandos AT, de modo que se configuró el microcontrolador como un host para el uso de su puerto USB. Este sistema permite la validación del SMS, aprobando exclusivamente las instrucciones que se envíen con la clave preestablecida en el código del microcontrolador. Todas las instrucciones se envían desde el nodo central hacia el carro a escala por medio de transceptores RF.

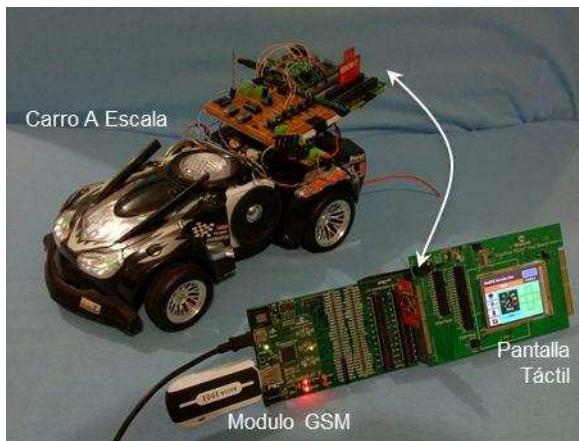


Figura 1. Foto de la implementación.

Para el proyecto se emplearon PIC32 Starter Kits, en los cuales se utilizó el sistema operativo en tiempo real SIWA-RTOS, el cual permite la ejecución de tareas mediante hilos de trabajo.