

MONITORIZACIÓN WEB DE SEÑALES INDUSTRIALES USANDO SISTEMAS EMBEBIDOS

(Web monitoring of industrial signals using embedded systems)

César Augusto Romero Molano¹

¹Escuela de Ingeniería, Universidad de los Llanos Unillanos. Grupo inv. Gitecx, cesar.romero@unillanos.edu.co

(Recibido el 24 de octubre de 2015 y aceptado el 1 de diciembre de 2015)

Resumen

El artículo presenta el diseño de *software* y *hardware* para un sistema de monitorización web de señales industriales. Este prototipo proporciona una interfaz web en la cual se puede observar en tiempo real el estado de un máximo de cuatro señales industriales del tipo *on-off*. En el diseño se utilizó un microcontrolador MSP432 encargado del muestreo y transmisión de las señales monitoreadas a una Raspberry PI, la cual recibe por un enlace UART los datos monitoreados del MSP432 y los presenta en el *front-end* de la aplicación web de manera inmediata. El prototipo diseñado fue verificado con una aplicación neumática formada por cuatro cilindros de efecto simple, y se observó una eficiente sincronización entre la ocurrencia del evento de activación o cambio en el estado de cualquiera de los cilindros monitoreados y su publicación web.

Palabras clave: msp432, mcu, node.js, Raspberry PI, sbc.

Abstract

The paper presents the design of software and hardware for a system of web monitoring of industrial signals. This prototype provides a web interface which can observe in real time the status of four industrial-type signal on-off. MSP432 microcontroller is used for sampling and transmitting monitored signals to a Raspberry PI which receives by a UART link the MSP432 monitored data and presents them immediately in the front-end web application. The prototype design was verified with a pneumatic application that consists of four single-acting cylinders and it was observed an efficient synchronization between the occurrence of the triggering event or change in status of any of the monitored cylinder and web publishing.

Keywords: msp432, mcu, node.js, Raspberry PI, sbc.

1. INTRODUCCIÓN

En Colombia, el Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Nacional es líder en el desarrollo de aplicaciones electrónicas. El ingeniero Bareño (2011) en el artículo “Metodología para transferencia tecnológica en la industria electrónica basada en software libre y hardware copyleft” plantea un panorama en el cual la industria colombiana está sometida a países desarrollados para el suministro de dispositivos electrónicos, a tal punto que depende totalmente de productos externos, en su mayoría

asiáticos, algunos de baja calidad que no suplen los requerimientos del mercado local, pero que se utilizan por sus bajos precios.

En el desarrollo de aplicaciones electrónicas existe el campo de monitorización de señales industriales, las cuales se implementan con soluciones como: plataformas de implementación (Camargo, Durán & Rosas, 2013), (Arrieta & Tarazona, 2014), implementación de PLCs en dispositivos lógicos programables FPGAs (Milik, 2016) y proyectos de software y hardware libre (CIAA, 2016), entre otros.

La monitorización de señales permite reducir las distancias y conocer de forma casi inmediata el estado de un sistema de control. Uno de los requerimientos indispensables para monitorear en tiempo real señales digitales es la validez lógica de los cálculos y su validez temporal.

En algunos sistemas de control críticos puede ser más importante el tiempo de respuesta que la validez lógica, pudiendo ser necesario elegir un método de cálculo aproximado más rápido, con el objeto de cumplir las restricciones temporales (Muñoz, 2010).

Por lo anterior, el autor se plantea el siguiente interrogante ¿es posible monitorear en tiempo real señales digitales usando sistemas embebidos?

2. MATERIALES Y MÉTODOS

Para el diseño del sistema de monitorización web se definieron tres bloques con objetivos y resultados definidos, los cuales son: adquisición y acondicionamiento de señal, muestreo y transmisión de datos, presentación web de los datos muestreados. Las etapas de la metodología del sistema de monitorización web se encuentran en la Figura 1.

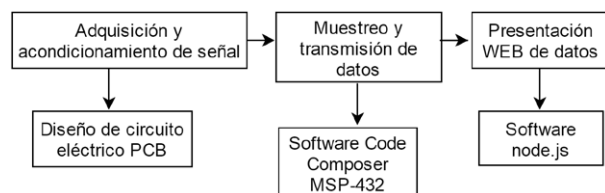


Figura 1. Etapas del sistema de monitorización

El enfoque utilizado permitió trabajar tareas en paralelo, las cuales se acoplan estratégicamente para dar cumplimiento progresivo a cada objetivo de etapa y así obtener un diseño del *software* y *hardware* necesario en el prototipo de adquisición de datos final.

Etapas 1 adquisición y acondicionamiento de señal. El objetivo de esta etapa es diseñar un módulo *hardware* que permita adquirir señales digitales industriales de tipo *on-off*.

Este objetivo se cumplió con el diseño de un módulo *hardware*, que captura señales digitales del tipo *on-off* que están en un rango de entrada de 0 a 30 voltios D.C.,

cumpliendo condiciones de protección como son: protección contra transitorios, polaridad invertida y tensiones negativas.

Estas señales de entrada se acondicionan a las características eléctricas del microcontrolador que se va a usar en la siguiente etapa. Para este caso se usó el MSP432, que es el dispositivo encargado del muestreo, por lo que se debe cumplir la condición de 3.0 voltios como voltaje máximo y 20 nano amperios como corriente máxima al pin usado como entrada digital (Texas Instruments, 2015). Como medida de protección cada entrada estará optoaislada.

El circuito eléctrico diseñado se observa en la Figura 2 y se basa en el uso del optoaislador PS2805, el cual tiene una alta capacidad de aislamiento (2500 voltios de valor real) en un encapsulado pequeño del tipo SSOP (Conti, Sgrigna, Zilpimiani & Assante, 2014).

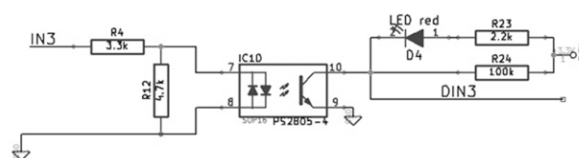


Figura 2. Circuito adquisición y acondicionamiento de señal.

Etapas 2 muestreo y transmisión de datos. El objetivo de esta etapa es diseñar un módulo encargado de muestrear y transmitir las señales que se acondicionaron en la etapa anterior.

El cumplimiento de este objetivo se logra, en el campo del *hardware*, usando los pines P3.0, P3.5, P3.6 y P3.7 del microcontrolador de Texas Instruments MSP432, configurados como pines de entrada digital, y los pines P3.2 y P3.3 como pines para la transmisión y recepción respectivamente de un enlace UART a 115200 baudios, 8 bits, sin bit paridad y un bit de parada entre el microcontrolador y la Raspberry PI esto al nivel del *hardware*.

Al nivel de *software* se utilizó Code Composer Studio 6.1.0 para realizar la gestión de los pines de entrada digital y de la UART del microcontrolador.

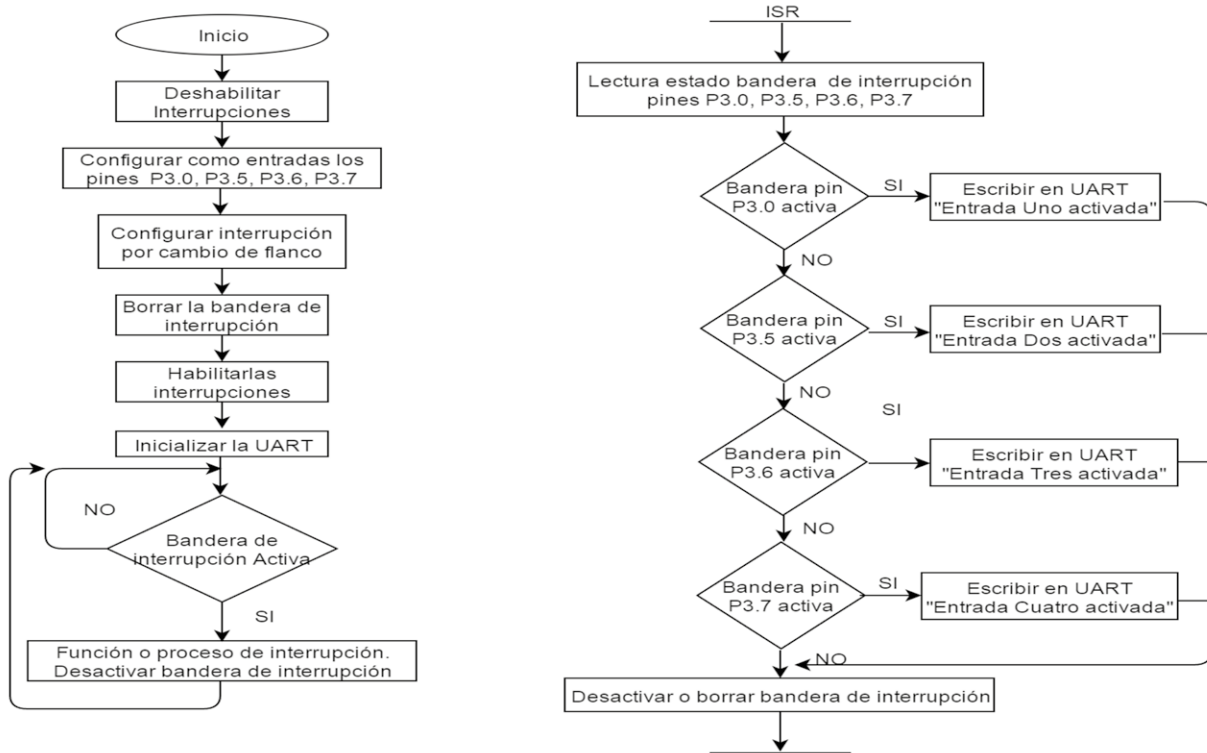


Figura 3. Flujograma.

En el flujograma se observa que en el programa principal (flujograma izquierdo) se realiza únicamente la configuración de los pines digitales que van a usarse, la inicialización y configuración de la UART y la habilitación del uso de interrupciones activadas por flanco, para dejar al microcontrolador a la espera de la ocurrencia de un llamado de servicio por interrupción por cualquiera de los pines configurados para tal fin. La rutina de servicio ejecutada en el llamado de una interrupción (flujograma derecho) verifica cuál pin fue el que presentó el cambio en el flanco de alto a bajo y luego hace la notificación vía UART a la Raspberry PI encargada de ejecutar la publicación web de la ocurrencia del evento.

Etapa 3 presentación web de los datos muestreados. El objetivo de esta etapa es diseñar el *software* necesario para la presentación web de los datos muestreados.

Este objetivo se cumplió diseñando dos servidores web que usan JavaScript y node.js.

La Figura 4 muestra un diagrama de bloques para ilustrar todas las conexiones de la aplicación desarrollada.

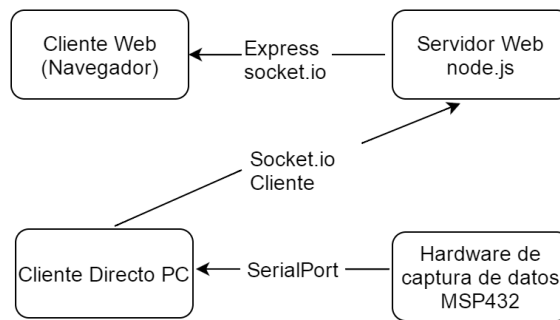


Figura 4. Diagrama de bloques de la aplicación web.

Un primer servidor alojado en la Raspberry PI es el encargado de recibir los datos que provienen del MSP432 haciendo uso de Serialport. El módulo Serialport permite crear un enlace UART entre el MSP432 y la Raspberry PI, ejecutándose directamente en esta y no mediante un navegador.

Una vez establecido el enlace UART, es preciso enviar estos datos al segundo servidor web, el cual está alojado en una máquina remota de cloud9 IDE, cuya función es presentar de manera inmediata los datos muestreados

en el *front-end* de la aplicación. En este servidor se encuentra en ejecución el módulo Express, encargado de enviar el archivo HTML al cliente, cuando este realice solicitudes al portal, y el módulo socket.io, que se dedica a manejar el tráfico de información entre el cliente y el servidor (variables para monitorear y comandos desde el cliente).

3. RESULTADOS

El sistema de adquisición y monitorización diseñado para *hardware* se presenta en la Figura 5, donde se aprecia que está formado por tres bloques: entrada de datos, microprocesador MSP432 y Raspberry PI, los cuales articulan con la metodología propuesta y permiten la implementación del sistema de adquisición y monitorización web.

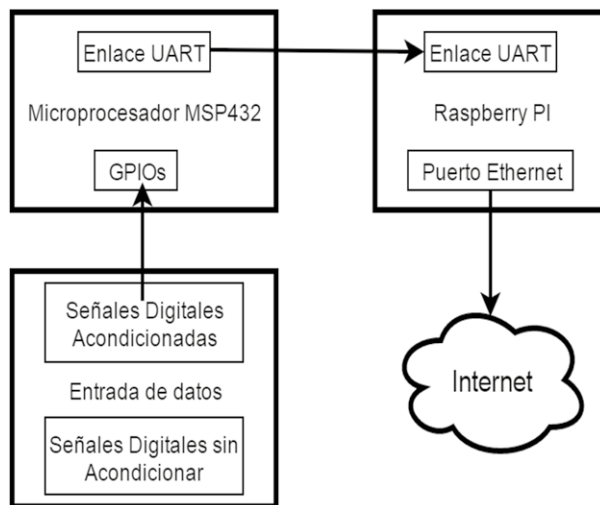


Figura 5. Diagrama de bloques del sistema.

Una vez integrados los bloques *hardware* y *software* del prototipo, se observa la publicación web del estado de cada señal monitoreada de forma adecuada.

Esta publicación web se lleva a cabo en 151.19 milisegundos, lo cual es un tiempo adecuado para aplicaciones industriales que presenten cambios en sus señales en un tiempo mayor al del sistema de monitorización diseñado, lo que garantizaría que no existe la posibilidad de pérdida de información. En pocas palabras, se puede considerar como un sistema que publica en tiempo real los eventos de la aplicación monitorizada.

Los resultados obtenidos en cada una de las etapas del sistema se detallan a continuación:

Módulo Entrada de datos.

En esta etapa se logró el diseño de una PCB rígida en fibra de vidrio FR4 figura 6, la misma fue diseñada con la herramienta Altium Designer 14.3 cumpliendo la norma IPC2221.

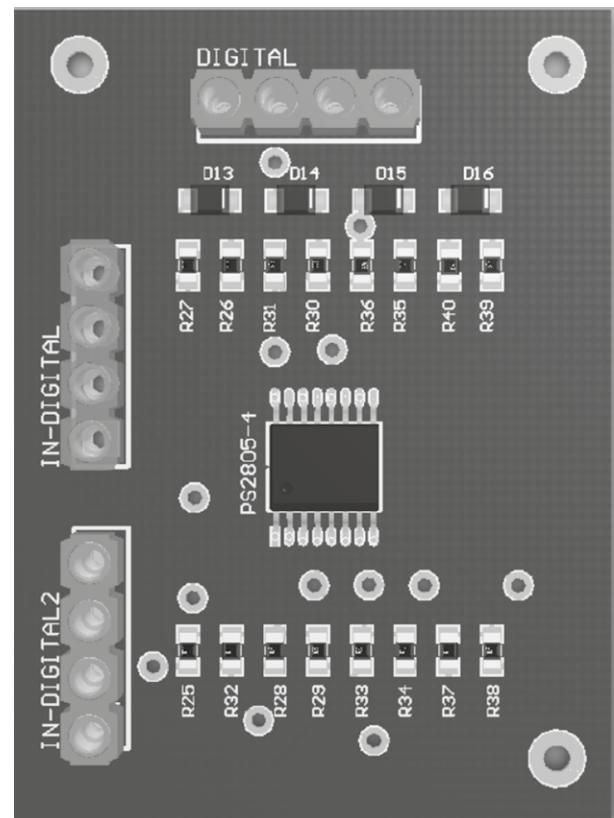


Figura 6. PCB diseñada.

Esta PCB se basa en el uso del optoaislador PS2805, el cual posee un *turn-on* de 10 microsegundos (Renesas, 2013), que será el tiempo máximo teórico de procesamiento de esta etapa.

Módulo microprocesador MSP432. En esta etapa se logró el diseño de un código escrito en lenguaje C, el cual utiliza interrupciones activadas por flanco e interrupción por transmisión de dato válido en UART. El tiempo máximo para el procesamiento en esta etapa es el que se requiere para enviar los datos vía UART a la Raspberry PI, i.e., 115200 bits/segundo.

Como el mensaje que se envía del MSP432 a la Raspberry PI es (salto INT pin X) el que usa 15 caracteres más los dos caracteres especiales de retorno de carro y fin de línea, esto nos entrega un tiempo de transmisión teórico de 1.18 milisegundos para esta etapa.

Módulo Raspberry PI. En esta etapa se logró el diseño de dos servidores web escritos en JavaScript y el *framework* node.js.

El primer servidor alojado en la Raspberry PI se observa en ejecución en la Figura 7.

```
pi@raspberrypi ~/ceroma $ node serial.js
open
From MSP: Linea...**
From MSP: salto INT pin 0
From MSP: salto INT pin 6
```

Figura 7. Servidor Raspberry.

La cual es una instantánea del servidor en modo de ejecución, que evidencia el envío del texto salto INT pin 0, cada vez que sucede el evento de interrupción en un pin del GPIO del MSP432.

El segundo servidor alojado en cloud9 IDE se observa en ejecución en la Figura 8.

```
bash - "ubuntu@x
ceroma175:~/workspace $ node app.js
listening on *:3000
user connected
user connected
Linea...**
Linea...**
salto INT pin 0
salto INT pin 0
salto INT pin 6
```

Figura 8. Servidor cloud9.

En la cual se verifica que se está ejecutando una máquina remota tipo Linux, que ejecuta el servidor node.js encargado de realizar la publicación web del evento recibo.

El tiempo calculado entre la captura de los datos y la presentación en el *front-end* de la aplicación, se encuentra en 150 milisegundos, usando una conexión a internet de 5MB. Este desfase de tiempo depende en gran

medida de la velocidad del canal de internet que use la aplicación.

4. ANÁLISIS DE RESULTADOS

Para la verificación del prototipo diseñado, se lleva a cabo el montaje de una aplicación neumática cíclica, formada por cuatro cilindros de efecto simple, cuatro finales de carrera y un PLC easy500 encargado de la activación de los cilindros. La Figura 9 muestra un fragmento del código en diagrama de contactos para el PLC.

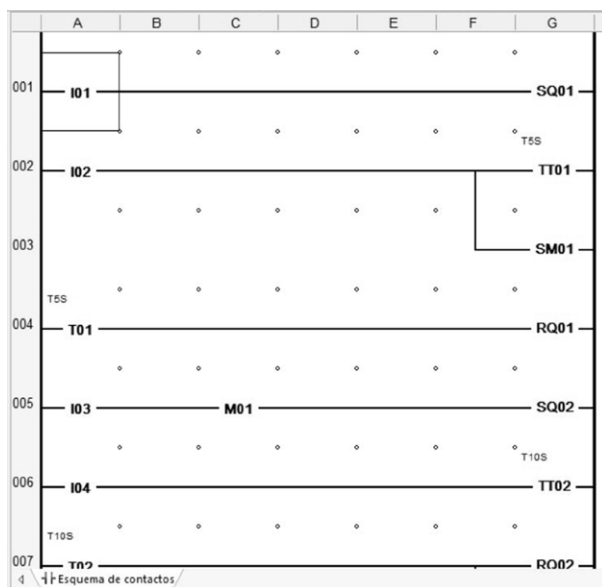


Figura 9. Código diagrama de contactos PLC.

El objetivo de esta aplicación neumática es simular el comportamiento de una aplicación industrial real, en la cual se activarán los cilindros de forma cíclica cada segundo.

La prueba de verificación inicia tomando las señales entregadas por los cuatro sensores o finales de carrera, que tienen un comportamiento *on-off* con un nivel de voltaje de 0 voltios para el *off* y 30 voltios para el *on*. Estas cuatro señales son ingresadas después directamente al módulo entrada de datos, diseñado para su acondicionamiento, muestreo y publicación web.

El tiempo que se requiere para realizar esta publicación web punto a punto, es de solo 150 milisegundos, sumando todos los retardos de tiempo en todas las etapas del sistema. El tiempo requerido para la sola publicación

web fluctúa entre 80 y 100 milisegundos, lo cual se puede verificar en la Figura 10.

```
pi@raspberrypi ~/ceroma $ ls
node_modules serial.js tes.js
pi@raspberrypi ~/ceroma $ node tes.js
From Server: Hi back 854 ms
From Server: Hi back 0 ms
From Server: Hi back 106 ms
From Server: Hi back 91 ms
From Server: Hi back 93 ms
From Server: Hi back 93 ms
From Server: Hi back 92 ms
From Server: Hi back 101 ms
From Server: Hi back 109 ms
From Server: Hi back 108 ms
From Server: Hi back 89 ms
From Server: Hi back 101 ms
From Server: Hi back 117 ms
From Server: Hi back 80 ms
From Server: Hi back 85 ms
```

Figura 10. Test de publicación web.

El comportamiento del sistema de monitorización diseñado, comparado con el trabajo de Antony, Mahato, Sharma y Chitranshi (2011), en el cual diseñan un PLC web, y con el trabajo de Ahmad, Mamat, Juhari, Daud & Arshad (2008), en el que se diseña un sistema de adquisición de datos inalámbrico en board de 32 bit, tiene como ventaja la arquitectura del sistema, ya que se usan plataformas para desarrollo de bajo costo, las cuales tienen el carácter de *hardware* libre y le permiten al usuario una fácil adaptación a cualquier aplicación industrial con un tiempo de respuesta del orden de milisegundos, lo cual se puede considerar como tiempo real para ciertas aplicaciones de tipo industrial.

Continuando con el análisis del prototipo, pero ahora orientado en aplicaciones que utilicen o requirieran hacer uso del concepto de tiempo real, se usa como referente de análisis el trabajo de Greitans, Grunde, Jakovics & Gendelis (2013), en el que se monitorea la eficiencia energética de unos bancos de prueba. El prototipo diseñado se destaca frente al analizado, en la forma como se realiza la publicación, ya que este usa el concepto de *socket* en *node.js*, lo cual garantiza una publicación web instantánea, solo retrasada por la eficiencia del medio de comunicación.

La principal desventaja del sistema diseñado es la de no poseer una interface web que presente de forma gráfica los datos monitoreados.

5. CONCLUSIONES

Desde el punto de vista técnico, el prototipo presentó robustez tecnológica y usabilidad, por lo cual fue posible acondicionarlo a un sistema formado por cuatro cilindros en funcionamiento y acoplarlo a protocolos de comunicación estándar, para la publicación web del estado de estos en tiempo real.

La publicación web del estado de señales industriales aporta al desarrollo de sistemas *hardware software* orientados al internet de las cosas, ya que cosas (actuales industriales) proporcionan datos en la web para su tratamiento y análisis.

El uso del optoaislador PS2805 permitió acondicionar adecuadamente señales con características industriales a las características estándar de microcontroladores genéricos de 32 bits, proporcionando un adecuado aislamiento A.C. D.C. de estas señales que se están muestreando.

6. REFERENCIAS

- Ahmad, R. B., Mamat, W. M. A., Juhari, M. R. M., Daud, S., & Arshad, N. W. (2008). Web-based wireless data acquisition system using 32bit single board computer. In *2008 International Conference on Computer and Communication Engineering* (pp. 777–782). IEEE. DOI: <http://doi.org/10.1109/ICCCE.2008.4580711>.
- Álvaro Ángel, A.-A., & Rosa Liliana, T.-C. (2014). Sistema multipotenciostato basado en instrumentación virtual. *Ingeniería, Investigación y Tecnología*, 15(3), 321–337. DOI: [http://doi.org/10.1016/S1405-7743\(14\)70344-0](http://doi.org/10.1016/S1405-7743(14)70344-0).
- Antony, J., Mahato, B., Sharma, S., & Chitranshi, G. (2011). A Web PLC Using Distributed Web Servers for Data Acquisition and Control: Web Based PLC. *2011 International Conference on Information Science and Applications*, 1–4. DOI: <http://doi.org/10.1109/ICISA.2011.5772427>.
- Bareño, C. I. C. (2011). *Metodología para la Transferencia Tecnológica en la Industria Electrónica Basada en Software Libre y Hardware Copyleft*. (Universidad Nacional de Colombia, Ed.). Bogotá.

- Camargo B., C. I., Durán B, L. K., & Rosas J., N. F. (2013). Plataforma hardware/software abierta para aplicaciones en procesos de automatización industrial. *Ingenium Revista de La Facultad de Ingeniería*, (c), 76–85.
- CIAA. (2016). Computadora Industrial Abierta Argentina. Retrieved January 30, 2016, from <http://www.proyectociaa.com.ar/devwiki/doku.php?id=start>
- Conti, L., Sgrigna, V., Zilpimiani, D., & Assante, D. (2014). Method for signal conditioning and data acquisition system, based on variable amplification and feedback technique. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 756, 23–29. DOI: <http://doi.org/10.1016/j.nima.2014.04.009>.
- Greitans, M., Grunde, U., Jakovics, A., & Gendelis, S. (2013). Web-based real-time data acquisition system as tool for energy efficiency monitoring. *2013 21st Telecommunications Forum Telfor (TELFOR)*, 553–556. DOI: <http://doi.org/10.1109/TELFOR.2013.6716289>.
- Isolation, H., Ac, V., Response, I., & Ssp, T. (2013). Ps2805c-1, ps2805c-4. Retrieved January 30, 2016, from http://documentation.renesas.com/doc/YOUSYS/document/r08ds0074ej0300_nepoc.pdf.
- José, Daniel, M. F. (2010). Summary for Policymakers. In Intergovernmental Panel on Climate Change (Ed.), *Climate Change 2013 - The Physical Science Basis* (Vol. 1, pp. 1–30). Cambridge: Cambridge University Press. DOI:<http://doi.org/10.1017/CBO9781107415324.004>.
- Milik, A. (2016). On Hardware Synthesis and Implementation of PLC Programs in FPGAs. *Microprocessors and Microsystems*. DOI: <http://doi.org/10.1016/j.micpro.2016.02.003>.
- Texas Instruments. (2015). MSP432P401x Mixed-Signal Microcontrollers. Retrieved January 30, 2016, from <http://www.ti.com/lit/ds/slas826a/slas826a.pdf>.