

Sistema de adquisición de datos y control industrial basado en Arduino Due con hardware y software libre

Gustavo Ernesto Real
Universidad Nacional de
General Sarmiento - UNGS
Los Polvorines, Argentina
greal@ungs.edu.ar

María Florencia Jauré
Universidad Nacional de
General Sarmiento - UNGS
Los Polvorines, Argentina
fjaure@campus.ungs.edu.ar

Amado Osvaldo Vitali
Universidad Nacional de
General Sarmiento - UNGS
Los Polvorines, Argentina
avitali@ungs.edu.ar

Resumen—*Diversos requerimientos de automatización de procesos, realizados por investigadores y estudiantes universitarios, alentaron el diseño y desarrollo de una placa de adquisición de datos y control, con aplicación en automatizaciones y mediciones en laboratorios educativos y en entornos industriales, basado en software y hardware libre. En este trabajo se detalla el desarrollo realizado para dar respuesta a este requerimiento. Inicialmente, se presentan los distintos elementos constitutivos del sistema, el cual se diseñó centrado en la plataforma Arduino Due; posteriormente se profundiza en la descripción funcional de los mismos y, finalmente, se describen las pruebas iniciales realizadas sobre la placa de prueba. Estas evidenciaron el funcionamiento de los principales componentes, como así también, la necesidad de continuar trabajando con la placa prototipo definitivas para alcanzar los resultados esperados y, así, comprobar todas las interacciones de los componentes en su versión final.*

Palabras clave—*adquisición de datos, lazo de control, automatización industrial, hardware abierto, software libre*

I. INTRODUCCIÓN

A finales de 2015, este grupo de trabajo logró consolidar un sistema de adquisición de datos (SAD) diseñado para ser utilizado como herramienta didáctica en Laboratorios de Ciencias e Ingeniería de la Universidad Nacional de General Sarmiento (UNGS), basado en software y hardware libre [1]. Satisfactoriamente, este sistema se incluyó en el equipamiento de diversos trabajos desarrollados por estudiantes e investigadores de la UNGS, por ejemplo: (1) se utilizó para realizar el control de un seguidor solar instalado en una parábola para generación de energía eléctrica; (2) se incorporó en el sistema de automatización de una planta de compostaje desarrollado por estudiantes de esta casa de estudios; (3) se incluyó en diversos usos didácticos, particularmente, se utiliza en las prácticas habituales de la tecnicatura en automatización de la UNGS.

Sin embargo, teniendo en consideración numerosos requerimientos de distintos equipos de investigación y de estudiantes, se consideró necesario incrementar las prestaciones de este primer desarrollo ya mencionado.

De esta forma, el nuevo SAD, que aquí se presenta, continuará con su esencia de ser de diseño y uso libre -tanto sus componentes de software como de hardware, siguiendo la tendencia global de uso abierto [2][3]-, con el valor agregado de brindar mayores prestaciones, integrando la solución a una importante comunidad de usuarios y maximizando la posibilidad de intercambio y colaboración con pares de otras universidades y latitudes, abarcando distintos campos de investigación. Esta placa les posibilitará a los estudiantes universitarios incorporar conocimientos sobre microcontroladores de 32 bits a través de una aplicación práctica y concreta, tanto para uso en adquisición de datos, como en control.

Otro factor que estimula la continuidad en el desarrollo de diseños propios, teniendo en cuenta la experiencia ya realizada, es saber que ante cualquier problemática que presente el sistema o la necesidad de hacer algún cambio, se tiene el conocimiento y las herramientas para poder continuar, sin la necesidad y/o dependencia de empresas cuyo soporte técnico, muchas veces en Argentina, no se encuentra a la altura de las circunstancias.

El desarrollo del sistema está centrado en la utilización de la Plataforma Arduino Due [4], justificada esta decisión por su amplio desarrollo, estabilidad y disponibilidad de información. De esta forma, la incorporación fundamental, respecto al desarrollo previo, lo establece el uso de un microcontrolador de mayor poder de procesamiento. Esto, a su vez, permite disponer de una mayor cantidad de canales de entradas y salidas y, fundamentalmente, orientar el diseño a la medición y automatización industrial.

II. ESTRUCTURA Y COMPOSICIÓN DEL SISTEMA

Brevemente, el sistema consiste en un dispositivo de hardware -basado en un microcontrolador de 32 bits [5] (Arduino Due)-, el firmware que se ejecuta en el mismo, el conjunto de sensores o adaptadores de señal que transforman las magnitudes físicas de interés en tensiones eléctricas adecuadas y un software de computadora que se comunica con el dispositivo para configurarlo y obtener los datos de las mediciones, visualizando y almacenando sus

valores. También se prevé la comunicación del sistema con centrales de automatización a través de buses estándar. En la Fig. 1 se puede observar el esquema conceptual del sistema.

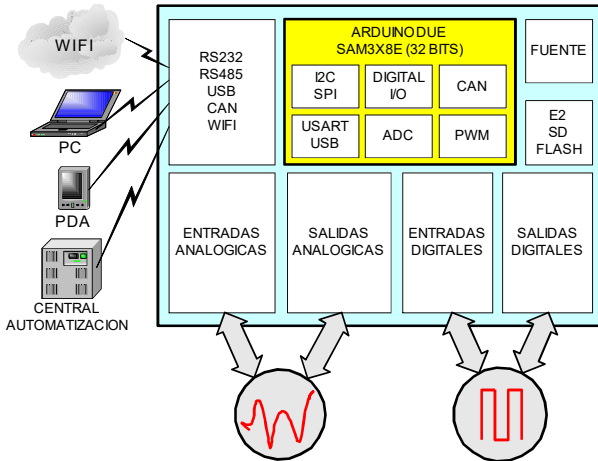


Fig. 1. Esquema conceptual del sistema

A continuación se describen con mayor detalle estos elementos.

A. Especificaciones

Dado que el desarrollo de este proyecto persigue dar soluciones o respuestas a las necesidades de la automatización industrial, la definición de los requerimientos para el hardware del sistema de adquisición de datos y control basado en Arduino Due se realizaron en función de esta premisa.

Por esta razón, esta solución provee de salidas analógicas que se utilizan para los lazos de control, ya que permite la modificación en forma continua -dentro del rango de las especificaciones- de dos actuadores que pueden intervenir en sendos lazos. Las entradas y salidas analógicas están entre 0-10 volts o 4-20 mA, seleccionable por el usuario. También se diseñó con buses RS485 y RS232 y comunicación WiFi; aspecto este último que no se orienta directamente a la automatización, pero sí provee de otras posibilidades al desarrollo. Por último, para ampliar los alcances del uso del sistema, se incluyó la comunicación CAN posibilitando, de esta forma, la vinculación con controles en el sector automotriz. Fundamentalmente, el microcontrolador de 32 bits facilita una gran variedad de prestaciones y un gran poder de procesamiento.

Habiendo ya analizado los aspectos generales del sistema desarrollado, a continuación se detallan los componentes utilizados en el hardware del sistema y sus respectivas especificaciones técnicas:

- PCB (Placa de circuito impreso)
 - Montaje superficial
 - Cuatro capas

- Material constitutivo: FR4
- Salidas
 - Cuatro salidas a relé – 24V @ 1A
 - Cuatro salidas Open Drain – 24V @ 0.5A
 - Cuatro salidas PWM – 0 a 3,3V @ 10mA
 - Dos salidas analógicas (DAC) – 0 a 10V - 0 a 20mA, seleccionable por jumper
- Entradas
 - Ocho entradas digitales optoacopladas – 0 a 24V
 - Ocho entradas analógicas (*single ended*) – 0 a 10V - 0 a 20mA, seleccionable con *jumpers*
 - Cuatro entradas analógicas de instrumentación para mediciones especiales – 0 a 3.3V @ .5mA, ganancia programable por programa
- Buses
 - Bus I2C, SPI, GPIO genérica (por ejemplo: para 1-wire, interrupciones y *triggers*)
- Comunicación
 - RS485
 - RS232
 - Serial USB (por placa Arduino Due)
 - CAN
 - Conector para placa de WiFi (basada en ESP8266)
- Almacenamiento
 - Conector para memoria SD
 - Memoria EEPROM (I2C)
 - Memoria Flash (SPI)
- RTC
 - Por I2C con pila de litio
- Frecuencia máxima de muestreo
 - 52 us @ 12 bits. Cada 52 us se muestran todos los canales elegidos
- Alimentación
 - Entrada por transformador: 9Vac+9Vac - 40VA
 - Entrada en continua: 15Vdc - 40VA

B. Diseño de la arquitectura

El diseño de la arquitectura implica, específicamente, la definición del funcionamiento de la solución. Analizando los objetivos y el marco del proyecto existían distintas opciones de diseño, sin embargo, teniendo en consideración las especificaciones iniciales ya detalladas, se consideró

pertinente que la solución realice parte de sus tareas de forma autónoma y otra parte que las realice conectado a otro equipamiento (por ejemplo: una PC, central de automatización, etcétera).

Como parte del análisis inicial de la arquitectura del hardware existían dos posibilidades: realizar un diseño modular o un diseño integrado¹. Si bien la modularidad podría simplificar cada placa en sí, al necesitar más conectores se podría incrementar los posibles puntos de fallas del sistema. Asimismo, como la placa está pensada para automatización, con una cantidad suficiente de entradas, salidas y comunicación, se llegó a la conclusión de avanzar por un diseño de tipo monolítico.

Definidas estas primeras características, se completó la arquitectura del sistema, la cual incluye los siguientes elementos de diseño:

a) *Diseño de hardware – Esquemático y PCB*

El diseño del hardware se inició con el trazado del circuito esquemático (Fig. 2).

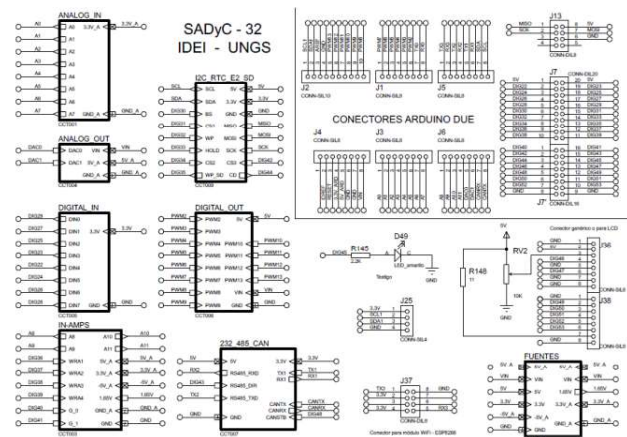


Fig. 2. Representación del circuito esquemático [6]

Pasando al diseño del PCB, inicialmente, se pensó en organizar todos los requerimientos en una placa de dos capas. Sin embargo, luego de desarrollar un diseño preliminar y de revisarlo en forma conjunta con el proveedor de circuitos impresos[7], se concluyó que la placa debía ser diseñada con cuatro capas.

En esta instancia, resulta importante indicar que el desarrollo de la placa correspondiente al sistema de control y adquisición de datos requirió el contacto con varios proveedores que realizaron o realizarán distintas tareas en el proceso de fabricación y armado de la placa prototipo definitiva. Se realizaron intercambios con proveedores para:

¹ Se debe tener en consideración que firmas internacionales muy reconocidas en el mercado desarrollan versiones con ambos criterios: versiones modulares y monolíticas.

(1) la fabricación de las placas [7], (2) la adquisición de los componentes necesarios [8] y, por último, (3) el armado de la misma. En este sentido, dado que la placa es de montaje superficial y que para realizar dicho montaje se requiere de un *expertise* diferenciado, se decidió que el montaje de los componentes sea realizado por un proveedor externo [9].

Cabe destacar que el desarrollo del proyecto sufrió alguna dificultad debido al retraso en la entrega de componentes provenientes del exterior del país. Por esta razón, y para evitar la demora de las sub tareas sucesivas planteadas en el cronograma inicial, se trabajó sobre una placa experimental de la firma Mikroelectronika [10], adaptando la placa Arduino a la misma, tanto en lo que respecta a la alimentación como a la señalización. Esto posibilitó la prueba de los módulos de firmware de la memoria SD, del reloj, de algunas entradas y salidas digitales, de la comunicación RS232, de la comunicación USB y de los módulos de registro. En lo que respecta a las pruebas de las entradas digitales, se utilizó otra placa de prueba confeccionada por este equipo de trabajo.

Una representación de lo que será la placa cuando esté terminada se puede observar en las Fig. 3 y Fig. 4, las cuales muestran las vistas superior y posterior, respectivamente, de la placa impresa con todos sus componentes posicionados.

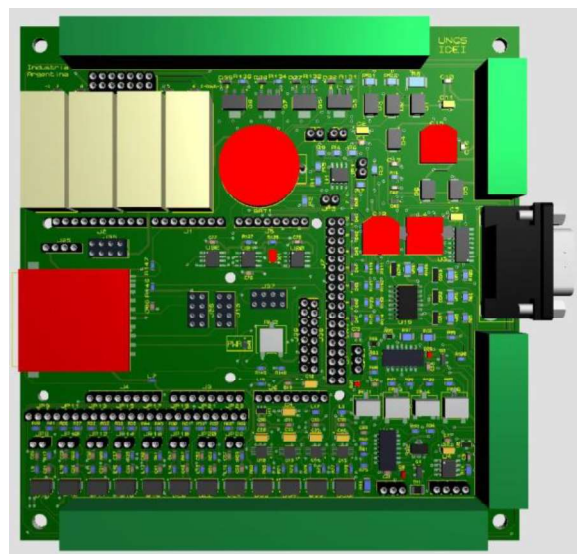


Fig. 3. TopView de PCB

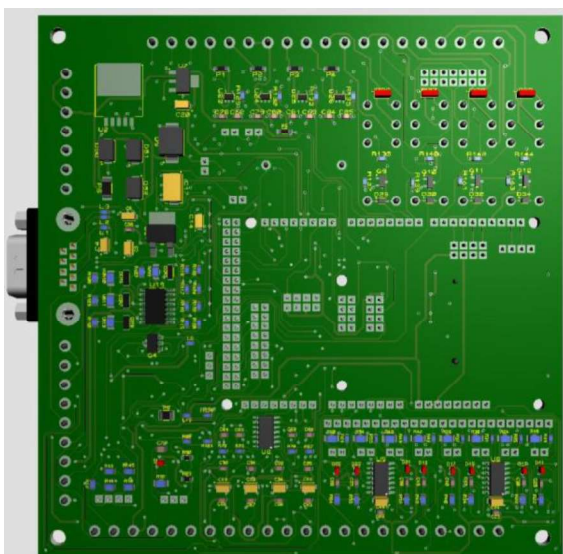


Fig. 4. BottomView de PCB

b) Versionado de firmware y software

En forma paralela a las tareas antes mencionadas, se realizó el seguimiento del desarrollo a través del uso de la herramienta Github [6] para realizar el versionado del firmware y software. El cual constituye una herramienta que permite la revisión sistemática de la evolución de estos elementos a medida que se van desarrollando.

c) Firmware

A los efectos de simplificar el uso de la placa se decidió que la placa se comunique hacia los periféricos a través de comandos. Cada comando responde a una comunicación específica, sin embargo, todos ellos se estructuran de la misma forma, logrando un cierto grado de estandarización, facilitando el agregado o eliminación de los mismos. De acuerdo a estas especificaciones, se definió como formato genérico de los comandos lo que se detalla a continuación en la Tabla I.

TABLA I. COMANDOS GENÉRICOS

Formato genérico del comando	
Esc	Atención
Carácter o número	Tipo de comando
Data	Datos del comando

De esta forma, no solo se logra una comunicación concreta y versátil, sino que cualquier especificación adicional que se quiera incorporar y que implique una relación con los periféricos, se podrá realizar de la misma forma, estandarizando así toda la comunicación presente y futura.

Al día de hoy, los comandos definidos para toda la comunicación se encuentran especificados en la Tabla II.

TABLA II. COMANDOS

1.- //Test de los chips de EEPROM instalados -(ESC 1 Valor CR) - Valor => 0 a 255
2.- //Test de la comunicación RS232 - (ESC 2 Char CR) - Char => cualquier carácter ASCII
A.- //Recibe cantidad de canales analógicos - (ESC A Cantidad CR) - Cantidad => 1 a 8
B.- //Recibe cantidad de canales analógicos tipo inAmp - (ESC B Cantidad CR) - Cantidad => 1 a 4
C.- //Seteo de la amplificación de los inAmp - (ESC C Entrada Amplificación CR) - Entrada => 09 a 12 - Amplificación => 0 a 3
D.- //Medición de tiempo entre eventos - (ESC D Pin1 Pin2 CR) - Pin1 y Pin2 => patas válidas, puede ser la misma. Patas válidas: 49, 50, 51, 52, 53 Devuelve medición en ASCII en microsegundos (variable Long)
E.- //Recibe el modo de trabajo del equipo - (ESC E Modo CR) - Modo => 1 a 5
F.- //Borrado EEPROM - (ESC F CR)
H.-//Seteo del reloj de tiempo real por I2c - (ESC H hhmmsDDMMAAAA CR)
M.- //Generación de PWM - (ESC M Pin Duty CR) - Pin => 02 al 13 - Duty => 0000 a 4095
N.- //Generación de señales analógicas en DACs - (ESC N Pin Steps CR) - Pin => 00 al 01 - Steps => 0000 a 4095
O.- //Transmite datos de un archivo en memoria SD - (ESC O Archivo CR) - Archivo en formato -> 15.3 máximo
P.- //Parada de los modos del equipo - (ESC P CR)
Q.- //Graba un dato fijo en un archivo en memoria SD - (ESC Q Archivo CR) - Archivo en formato -> 15.3 máximo
R.- //Registro temporizado de los canales analógicos seteados - (ESC R Tiempo CR) - Tiempo => 1 a 3600 segundos
S.- //Seteo de la salida digital - (ESC S Salida Estado CR) - Salida => 02 a 09 - Estado 0 o 1
T.- //Transmisión temporizada de los canales analógicos seteados - (ESC T Tiempo CR) - Tiempo => 1 a 10000000 microsegundos Tiempo=0 -> una transmisión (En los datos analógicos se transmite primero el byte alto y luego el bajo)
U.- //Transmisión de los canales digitales de entrada en un solo byte binario - (ESC U CR)
W.- //Seteo de Red y Password para WiFi - (ESC W Red 0xFF Pass 0xFF CR) - Red y Pass máximo 15 caracteres ASCII
d.- //Descarga de datos desde la EEPROM - (ESC d CR)
e.- //Transmite datos del equipo -(ESC e CR) - Responde con los datos básicos del proyecto
h.- //Transmite día y hora del RTC - (ESC h CR) - Responde con día y hora del equipo
Ref.: ESC → Escape (0x1b) CR → Carriage Return (0x0d)

En cuanto al funcionamiento, la placa trabaja por modos. Esto está directamente asociado con cada una de las prestaciones que dispone, de acuerdo con la medición y/o control que se quiera realizar. Dichos modos son mutuamente excluyentes.

En este sentido, se definió que la placa puede trabajar en los siguientes tres modos:

- Adquisición de datos (**ADQ**)
- Automatización (**AUI**)
- Medición de Tiempos (**MDT**)

Resulta importante tener en consideración que se deberá detener el modo de trabajo en curso para poder iniciar cualquier otro modo.

A continuación, en la TABLA III, se especifica el detalle de los modos de trabajo.

TABLA III. MODOS DE TRABAJO

Especificaciones básicas por modo	Descripción por modo
ADQ1 "On Line" (Modo 1): - Canales Analógicos de entrada - Máximo 8 - Canales Digitales de entrada - Máximo 8	La placa será consultada en tiempo real vía el canal de comunicación disponible (ej: RS232), acerca del estado de los canales analógicos y digitales elegidos al momento de configurar la medición.
ADQ1 "Registro" (Modo 2): - Canales Analógicos de entrada - Máximo 8 - Canales Digitales de entrada - Máximo 8	La placa registrará en alguna de las memorias disponibles el estado de los canales analógicos y digitales elegidos al momento de configurar la medición. En el instante en que se lo requiera, vía el canal de comunicación elegido, se podrán descargar los datos de la medición registrados y realizar el procesamiento correspondiente.
ADQ2 "On Line" (Modo 3): - Canales Analógicos de entrada - Máximo 4 - Tipo instrumentación	La placa será consultada en tiempo real vía el canal de comunicación disponible (ej: RS232), acerca del estado de los canales analógicos y digitales elegidos al momento de configurar la medición.
ADQ2 "Registro" (Modo 4): - Canales Analógicos de entrada - Máximo 4 - Tipo instrumentación	La placa registrará en alguna de las memorias disponibles el estado de los canales analógicos y digitales elegidos al momento de configurar la medición. En el instante en que se lo requiera, vía el canal de comunicación elegido, se podrán descargar los datos de la medición registrados y realizar el procesamiento correspondiente.
AUI (Modo 5): - Canales Analógicos de entrada - Máximo 8 - Canales Analógicos de salida - Máximo 2 - Canales Digitales de entrada - Máximo 8 - Canales Digitales de salida a relé - Máximo 4 - Canales Digitales de salida Open Drain - Máximo 4 - Salidas PWM - Máximo 2 - Comunicación RS232 y RS485	La placa deberá ser capaz de medir sobre todos los canales de entrada especificados en la configuración, y de acuerdo a la programación, actuar sobre los canales de salida correspondientes. Esto implica la existencia de un lazo de control, ya sea remoto (la toma de decisión está en un centro de automatización); o un lazo local, en donde la misma placa toma las decisiones correspondientes. NOTA: El presente proyecto no abarca la implementación de un lazo de control local, pero tiene todo lo necesario para implementar un lazo de control remoto.
MDT (Modo 6): - Entre 2 entrada - Sobre una misma entrada	La placa deberá ser capaz de medir tiempos entre eventos. Los mismos se manifiestan a través de un cambio de estado en una misma entrada digital, o entre dos entradas digitales distintas. El resultado se informará a través del canal de comunicación que corresponda.

El detalle de cómo fue estructurado y codificado el firmware se encuentra disponible en la página del proyecto: <https://github.com/SADyCUNGSIDEI/Firmware>

Si bien algunas de las bibliotecas disponibles para uso en el entorno de programación Arduino podrían no estar lo suficientemente probadas, planteando dudas relativas a la fiabilidad y/o estabilidad del firmware, cabe destacar que como nuestro firmware es abierto, cualquier interesado puede reescribirlo con las bibliotecas que considere adecuadas. Es decir, si la estabilidad o fiabilidad no es suficiente, se puede reescribir la biblioteca que está generando el problema o, en su defecto, todo el firmware.

d) Software

Para proveer a los usuarios de una primera² interfaz de comunicación con el sistema de adquisición y control, se desarrolló un software utilizando lenguaje Python 2.7 [11][12]. La selección de este lenguaje se justifica principalmente por: (1) ser de código abierto, (2) cuenta con soporte de una comunidad activa de usuarios y programadores, (3) está muy documentado, (4) los programas se ejecutan en múltiples plataformas, prácticamente, sin modificaciones en el código fuente, (5) existía el *know-how* en el equipo de trabajo, (6) soporta múltiples paradigmas de programación, (7) permite desarrollar aplicaciones complejas, y (8) dispone de numerosas bibliotecas de código abierto para aplicaciones específicas como gráficos 2D y 3D, comunicaciones, interfaces gráficas de usuario, cálculo científico y desarrollo web.

Las bibliotecas gráficas utilizadas son: PyQt [14] para el diseño de ventanas y PyQtGraph [14] para el plotteo de los datos en gráficos en dos ejes cartesianos.

La diagramación de este software se presenta en tres capas:

- la capa de presentación, en donde la vista es creada en archivos .xml de QtDesigner [15] y controlada mediante Python (que hace uso de los archivos .xml de la vista);
- la capa de lógica, en donde parte de la lógica de la placa es representada, así como su estado; y
- la capa de acceso a datos, en donde se guarda información en el disco duro y se realiza la comunicación con el puerto serie de la placa.

El orden de las capas es: vista – lógica – acceso a datos. Cada capa solo hace uso de la capa inmediatamente inferior, es decir, la capa de vista solo ve a la capa lógica y esta última ve solo a la de acceso a datos.

² Dado que el sistema se diseñó considerando el uso de hardware y de software libre, la intención de proveer de una primera interfaz está lejos de ser una limitante o de considerar, de esta forma, cerrado este aspecto. Esta interfaz es el primer elemento de intercambio, el desarrollo posterior de una HMI o de una PDA resultan ejemplos de posibles interfaces a desarrollar en el futuro.

En la fig. 5 se muestra una pantalla como ejemplo de una de las visualizaciones ya generadas, representando una variable aleatoria simulada por el mismo software.

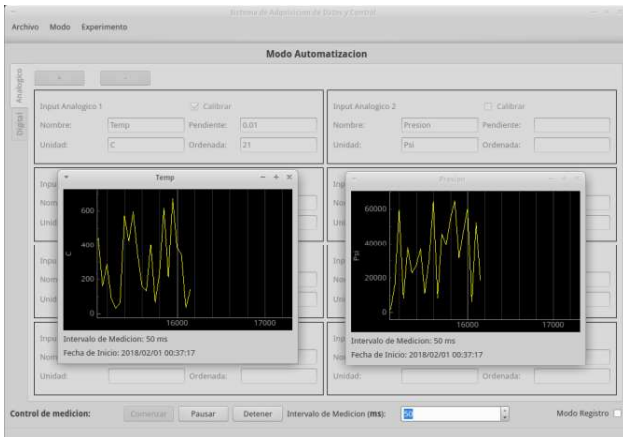


Fig. 5. Captura de pantalla de la interfaz gráfica del software

III. PRUEBAS PRELIMINARES

Ya se ha mencionado anteriormente la dificultad transitada respecto de la disponibilidad de los componentes para el montaje de la placa prototipo. Por esta razón, se tomó la decisión de realizar los primeros ensayos sobre placas de prueba y, de esta forma, analizar aspectos fundamentales, tales como: el comportamiento de los *drivers* de los dispositivos, la velocidad de conversión del ADC, la comunicación por USB y RS232 y la medición de señales sin la etapa de conformación y/o amplificación.

Si bien se tenía absoluta conciencia de que las mediciones de este tipo no demuestran cómo se va a comportar el equipamiento definitivo, pueden servir para ajustar el funcionamiento individual de componentes como los ya mencionados.

Hasta este momento, ya se ha probado todo el set de comandos, se logró establecer una comunicación satisfactoria entre una PC, en dónde funcionaba el software, y el firmware del equipo por USB y RS232; probándose inclusive la modalidad de transmisión continua de datos. En cuanto a la comunicación por Wifi, solo se probaron comandos básicos, ya que la influencia del limitado ancho de banda, no ha posibilitado realizar mediciones de transmisión continua. También se realizaron pruebas de transferencia de datos vía RS485 sin protocolo ModBus, ya que solo se quería probar hasta la capa de conectividad pero no entrar todavía en el uso de protocolos que complejizaban estas primeras pruebas.

Para las pruebas de muestreo de señales analógicas se colocó un generador de señal Rigol DG1022 y un Osciloscopio Tektronix TDS2001C a la entrada, de forma tal de disponer de toda la información de la señal inyectada a la placa de prueba (Figs. 6 y Fig. 7). Por otro lado, se solicitó al software de la PC que mostrara en tiempo real la

señal detectada en el canal correspondiente, recabando dichos datos a través de una comunicación USB.



Fig. 6. Ensayo con placa de prueba



Fig. 7. Instrumental de medición del ensayo

Los resultados fueron satisfactorios para las señales sinusoidal y cuadrada, en frecuencias hasta 100Hz (Fig. 8), donde se pudo verificar el correcto funcionamiento de todos los componentes involucrados en la prueba. Es importante destacar que las curvas obtenidas estaban dentro de los resultados esperados, pero no se procedió a realizar una medición válida del error total del sistema dado que no se dispone de la placa definitiva, por lo que una evaluación de este tipo no aportaría datos relevantes en esta instancia.

Cuando el mismo ensayo se llevó a cabo a frecuencias más elevada ($>1\text{KHz}$), comenzaron a manifestarse dos problemáticas: (1) el software tuvo problemas de estabilidad, colapsando las colas asignadas a la recepción de datos y graficación; (2) en el firmware se detectó una incompatibilidad entre el modo de manejo del ADC y la atención de los eventos seriales.

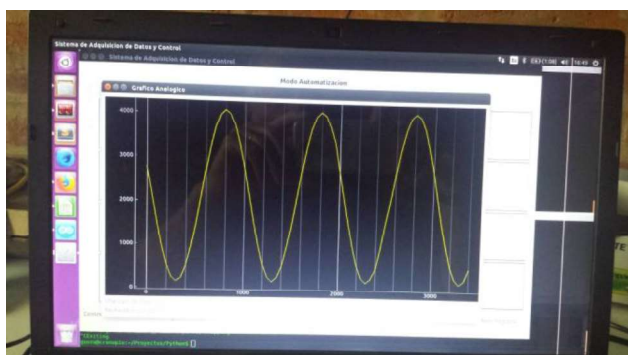


Fig. 8. Imagen de salida del sistema durante el ensayo, a 100Hz

Luego de un análisis detallado de la problemática del firmware surgida en estos primeros ensayos, se llegó a la conclusión que había que reestructurar la tarea de atención del ADC para poder bajar los tiempos de muestreo por debajo de 1ms. La consecuencia de este análisis llevó a la reescritura del driver del ADC y la eliminación del uso del *Scheduler* (biblioteca de Arduino) [16], con lo cual se superó el escollo de los 100Hz.

IV. DESARROLLO DE RECURSOS HUMANOS

El desarrollo de la placa prototipo que se detalló previamente, supuso la formación específica de dos estudiantes universitarios.

Inicialmente se unió el estudiante de la Carrera de la Licenciatura en Sistemas, Leandro Funes, como adscripto al proyecto; quién trabajó principalmente en el desarrollo integral del Software del sistema. Esta dedicación posibilitó el trabajo conjunto no sólo con el estudiante Funes, sino también con toda el área de Sistemas, ya que debimos definir criterios, tanto para el armado de un software adecuado a los requerimientos definidos, como así también, la realización de una tarea que implique un desarrollo formativo incremental para el estudiante. Este trabajo le brindó a Funes la posibilidad de escribir un software en un lenguaje de programación distinto a los ya estudiados en su carrera.

También se incorporó, como participante del proyecto, el estudiante de la Tecnicatura Superior en Automatización y Control, Gonzalo Ribera, quien contribuyó en el desarrollo de la modularización del firmware. Esto le permitió tomar contacto directo con microcontroladores de 32 bits, conocer su arquitectura y la realización de bibliotecas para los respectivos *drivers*.

Finalmente, se incorporó como adscripto del proyecto, Agustín Alexander, estudiante de la Licenciatura en Sistemas, quién realizará actividades vinculadas con los usos posteriores que ya se prevén para la placa desarrollada. Inicialmente, trabajará en la asistencia del uso de la placa para la automatización y el control automático del movimiento de una parábola para mantener el sol en el foco

de la misma, el cual accionará un motor Stirling para la producción de energía eléctrica en pequeña escala.

Específicamente una de las placas prototipo se utilizará en las prácticas de la materia Desarrollo Avanzado con Microcontroladores, de la Tecnicatura Superior en Automatización y Control.

V. CONCLUSIONES Y TRABAJO FUTURO

En principio, se han alcanzado ciertos hitos que indican que la investigación se encuentra en el camino correcto, ya que:

- se han diseñado todos los módulos de hardware, firmware y software del sistema;
- se dispone de las placas (PCB) en condiciones de ser montados los componentes;
- se probaron el 70% de los módulos en los ensayos con placas de prueba acondicionadas a tal efecto.
- los primeros resultados de las pruebas fueron satisfactorios;
- se corrigieron errores de firmware que serán clave para la correcta medición de las señales analógicas.

Por otro lado, el trabajo que queda por realizar en lo inmediato tiene distintas aristas, de las cuales las más destacadas son:

- realizar el montaje final de componentes en las placas previstas (dado que se ha vuelto a retrasar la entrega de los componentes, se estima que las placas estén finalizadas para los primeros días del mes junio de 2018);
- verificar el correcto funcionamiento del conjunto, tanto en sus aspectos de medición, como en los de comunicación;
- caracterizar el error de medición en cada uno de los modos de funcionamiento, contrastando los resultados con los entregados por un instrumento patrón;
- documentar exhaustivamente la arquitectura y el uso del sistema facilitando a los usuarios su participación en el proyecto y el uso de este sistema.

Finalmente, una vez avanzado en estos últimos ítems y, con la consecuente publicación en la página web de toda la información y los resultados de las pruebas, se torna importante la adhesión que tenga de la comunidad universitaria y de investigación para enriquecer un sistema que está pensado para ser mejorado, ampliado y enriquecido con nuevos componentes, tanto de hardware como de software.

De acuerdo a los valores manejados hasta el momento por la fabricación de la placa y componentes, se verifica que los costos están dentro de los valores estimados; de esta forma, se puede extrapolar que la versión final del sistema tendrá un costo menor a un "enlatado" del mercado, con prestaciones equivalentes [17][18][19].

VI. AGRADECIMIENTOS

Agradecemos muy especialmente, por la dedicación y el compromiso mostrado por los dos estudiantes que trabajaron como adscriptos en el proyecto de investigación: Leandro Funes, estudiante de la Licenciatura en Sistemas y Gonzalo Ribera, estudiante de la Tecnicatura en Automatización y Control.

En el marco de las pruebas de laboratorio, queremos expresar nuestro especial agradecimiento al personal del Laboratorio de Ingeniería del Instituto de Industria de la Universidad Nacional de General Sarmiento por la asistencia técnica, la provisión de instrumentos y la paciencia.

VII. REFERENCIAS

- [1] https://www.researchgate.net/publication/305305142_Data_acquisiti_on_system_for_didactic_laboratories_based_on_open-source_hardware_and_free_software
- [2] A. Gibb, *Building Open Source Hardware: DIY Manufacturing for Hackers and Makers*, Addison-Wesley Professional, 2015
- [3] J. M. Pearce, *Building Research Equipment with Free, Open-Source Hardware*, *Science*, 337, 1303 – 1304, 2012.
- [4] <https://store.arduino.cc/usa/arduino-due>
- [5] <http://www.microchip.com/wwwproducts/en/ATsam3x8e>
- [6] <https://github.com/SADyCUNGSIDEI/>
- [7] Circuitos impresos: <http://www.maverpcb.com/>
- [8] Componentes electrónicos: <http://www.farnell.com/>
- [9] Montaje y ensamble de placas electrónicas: <http://www.assisi.com.ar/>
- [10] Placa de prueba: <https://www.mikroe.com/uni-ds>
- [11] <http://python.org>
- [12] J.M. Hughes, *Real World Instrumentation with Python*, Sebastopol: O' Reilly, 2010.
- [13] <http://goo.gl/jFU0By>
- [14] <http://www.pyqtgraph.org/>
- [15] <http://doc.qt.io/archives/qt-4.8/designer-manual.html>
- [16] <https://www.arduino.cc/en/Reference/Scheduler>
- [17] J. M. Pearce, *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs*, Oxford: Elsevier, 2014.
- [18] M. Gopalakrishnan y M. Gühr, *A low-cost mirror mount control system for optics setups*, *American Journal of Physics*, 83, 186, 2015.
- [19] K. Zachariadou et al., *A low-cost computer controlled Arduino based educational laboratory system for teaching the fundamentals of photovoltaic cells*, *European Journal of Physics*, 33, 1599, 2012.