



UNIVERSIDAD NACIONAL DE COLOMBIA

# Diseño de Hardware y Firmware para un Sistema Inalámbrico de Adquisición de Datos (DAQ) de bajo costo.

Alejandro Suarez Ruiz

Universidad Nacional de Colombia  
Departamento de Ingeniería Eléctrica, Electrónica y Computación  
Manizales, Colombia  
2019



# Diseño de Hardware y Firmware para un Sistema Inalámbrico de Adquisición de Datos (DAQ) de bajo costo.

**Alejandro Suarez Ruiz**

Tesis o trabajo de grado presentada(o) como requisito parcial para optar al título de:  
**Magister en Automatización Industrial**

Director(a):  
Gustavo Adolfo Osorio Londoño

Línea de Investigación:  
Adquisición de datos y diseño electrónico.  
Grupo de Investigación:  
PCI-SHAC

Universidad Nacional de Colombia  
Departamento de Ingeniería Eléctrica, Electrónica y Computación  
Manizales, Colombia  
2019

# Hardware and Firmware design for a wireless and low cost data acquisition system (DAQ).

**Alejandro Suarez Ruiz**

Thesis or degree work presented as partial requirement to opt for the title:  
**Magister en Automatización Industrial**

Director:  
Gustavo Adolfo Osorio Londoño

Línea de Investigación:  
Electronic design and data acquisition.  
Investigation group:  
PCI-SHAC

Universidad Nacional de Colombia  
Departamento de Ingeniería Eléctrica, Electrónica y Computación  
Manizales, Colombia  
2019

## Dedicatoria

A mis padres, por su constante amor y apoyo.

La preocupación por el hombre y su destino siempre debe ser el interés primordial de todo esfuerzo técnico. Nunca olvides esto entre tus diagramas y ecuaciones.

Albert Einstein



# Agradecimientos

Aunque existen muchas personas que influyen en un proceso como estos, siempre habrá unas que dejan una marca especial. Por esto quiero agradecer a mis padres, quienes siempre me apoyaron a seguir adelante, y pusieron todo de sí para ello. Al profesor Gustavo, por acogerme en el grupo, tener la paciencia y el ánimo de enseñarme y corregirme. A la familia Jaramillo Villegas que siempre estuvo con gran disposición y apoyo hacia este proceso. Y a mis más cercanos amigos, que me brindaron su apoyo y confianza en los momentos más difíciles.





## Resumen

La adquisición de datos es un fenómeno de interés para las ciencias y la ingeniería. Un ejemplo práctico se presenta en las industrias, específicamente en el análisis y control de procesos. Los sistemas de adquisición de datos comerciales tienden a tener un costo elevado y a estar ligados a software propietarios. Esto dificulta el manejo de los datos adquiridos o la integración con otras aplicaciones. Además, aquellos que tienen funcionalidades de conexión inalámbrica aumentan de precio. Este trabajo se enfoca en el diseño de un sistema de adquisición de datos inalámbrico de bajo costo, de interacción amigable con el usuario y escalable. Para ello utiliza las técnicas de diseño electrónico, generando una PCB con componentes fáciles de integrar y costo asequible (Aproximadamente 20 dólares en total). Se utiliza un Firmware *open source* llamado Micropython, y se adapta para los periféricos de la tarjeta. Esto brinda una capacidad de programación y configuración de la tarjeta en un alto nivel. El resultado final está compuesto por una solución con ocho canales de adquisición analógico de 16 bits entre -5 y 5v, a una velocidad máxima de 1.2kS/s. Dos canales de salida analógica entre 0 y 5v, y un puerto I2C con el propósito de futuras expansiones. Cuenta también con comunicación inalámbrica WiFi y protocolo de sockets TCP. El trabajo deja abierta una línea de desarrollo que puede influir a nivel educativo y comercial en la región.

**Palabras clave:** Adquisición de datos, Hardware, Software, Firmware, Diseño electrónico, Conversión analógica-digital, Bajo costo, Micropython).

## Abstract

Data acquisition it's an interesting phenomenon for the science and the engineering. The industries is a practical example, specifically in the process control and analysis. The commercial data acquisition systems tends to have a high cost and be attached to proprietary software. This makes difficult the management of the acquired data or the integration with other applications. Also, those who have wireless connectivity increase the cost. The focus of this work is to design a low cost wireless data acquisition system, with a friendly user interaction and scalable architecture. This work use the electronic design techniques, generating a PCB with easy to assemble components at affordable costs (Aproximately 20 dollars in total). The used firmware its Micropython, an open source project, and its performed the adaptation to work with the board peripherals. This provides a high level programming and configuration of the board. The design it composed by eight analog input channels at 16 bits of resolution, a range between -5 and 5v and a maximum sample rate of 1.2kS/s. Two analog output channels between 0 and 5v and a I2C port with the aim of future expansions.

Also, counts with a WiFi and sockets TCP protocol. This work leave open a development line that can influence educational and commercial level in the region.

**Keywords: Data acquisition, Hardware, Software, Firmware, Electronic design, Analog to digital conversion, Low cost, Micropython)**

# Contenido

<b>Agradecimientos</b>	<b>VII</b>
<b>Resumen</b>	<b>IX</b>
<b>1. Introducción</b>	<b>2</b>
1.1. Propósito y motivación . . . . .	2
1.2. Estructura del documento . . . . .	4
<b>2. Marco teórico y estado del arte</b>	<b>6</b>
2.1. Fundamentos de la adquisición de datos . . . . .	6
2.1.1. Sensores y transductores . . . . .	7
2.1.2. Acondicionamiento de señal . . . . .	8
2.1.3. Conversión analógica - digital . . . . .	9
2.1.4. Elemento de cómputo . . . . .	11
2.2. Topologías de adquisición de datos . . . . .	12
2.2.1. Módulo I/O de conexión: . . . . .	12
2.2.2. Módulo I/O distribuido: . . . . .	13
2.2.3. Módulo independiente: . . . . .	14
2.3. Mercado e investigaciones relacionadas a la adquisición de datos . . . . .	15
<b>3. Propuesta de solución</b>	<b>17</b>
3.0.1. Beneficios . . . . .	17
3.0.2. Especificaciones del sistema . . . . .	18
3.0.3. Descripción del sistema . . . . .	18
3.1. Hardware . . . . .	20
3.1.1. Entradas . . . . .	20
3.1.2. Salidas . . . . .	23
3.1.3. Alimentación . . . . .	25
3.1.4. Estructura de cómputo . . . . .	27
3.1.5. Comunicaciones . . . . .	29
3.2. Resultados . . . . .	30
3.3. Pruebas y verificación técnica . . . . .	30

---

<b>4. Firmware</b>	<b>39</b>
4.1. Funcionamiento del dispositivo . . . . .	39
4.1.1. Lectura de entradas analógicas . . . . .	39
4.1.2. Control de las salidas analógicas . . . . .	40
4.2. Descripción del firmware . . . . .	41
4.2.1. Micropython . . . . .	42
4.2.2. Integración con la tarjeta . . . . .	42
4.2.3. Capacidades inalámbricas . . . . .	44
4.3. Casos de uso . . . . .	45
4.3.1. ADC con potenciómetro . . . . .	45
4.3.2. ADC con célula de carga . . . . .	47
4.3.3. DAC . . . . .	48
<b>5. Conclusiones</b>	<b>50</b>
<b>A. Anexo: Lista de materiales</b>	<b>54</b>
<b>B. Anexo: Esquemáticos</b>	<b>55</b>
<b>C. Anexo: Funciones y códigos</b>	<b>57</b>
<b>Bibliografía</b>	<b>64</b>

# 1. Introducción

## 1.1. Propósito y motivación

La adquisición de datos es el proceso mediante el cual, un fenómeno físico, del mundo real, es transformado en una señal eléctrica. Posteriormente es convertida a formato digital para su procesamiento, análisis y almacenamiento [32]. Se presenta como un tema de interés a nivel industrial e investigativo. Por ejemplo permite instrumentar y controlar procesos dentro de una empresa. También permite a nivel investigativo obtener los datos necesarios para analizar un fenómeno particular. El mercado responde a esto ofreciendo soluciones completas y robustas. Los problemas fundamentales de estas soluciones son el costo elevado y su relación con Software propietarios. Como es de esperarse, las funcionalidades como precisión, velocidad y estabilidad son proporcionales al costo. Existe además una gama de productos que se enmarcan dentro del bajo costo. En este trabajo se realiza una recopilación de algunas de estas alternativas que están entre los 99 y los 630 dólares. Esto sin incluir precios de importación y envío (Son de difícil adquisición en Colombia).

Este trabajo se enfocó en el desarrollo del hardware y el firmware para un sistema inalámbrico de adquisición de datos de bajo costo. Se partió de la referencia comercial para escoger los parámetros de funcionamiento del sistema, asegurando las modificaciones pertinentes para lograr una buena relación costo/beneficio/novedad. A nivel de hardware la tarjeta cuenta con ocho canales de adquisición analógicos de 16 bits a una velocidad máxima de 1.2kS/s, y con un rango de entradas entre los -5 y los 5v. Esto gracias al conversor ADS1120, una solución integrada para adquisición de Texas Instruments. Un factor clave respecto a este conversor es la compatibilidad en pines con el ADS1220, que tiene la misma cantidad de registros y comandos, pero cuenta con una resolución de 24 bits. Esto le permite a la tarjeta ser ensamblada con el componente más apropiado para la aplicación. Cuenta con dos canales de salida analógicos a una resolución de 12 bits, y un rango de 0 a 5v. El dispositivo empleado es el DAC7311 de Texas Instruments. Al igual que la sección de entradas analógicas tiene un conversor compatible en pines. El DAC8411 que cuenta con 16 bits de resolución.

La unidad de cómputo escogida es un microcontrolador de 32 bits llamado el ESP8266. Es un SOC de la empresa Espressif que implementa un procesador Tensilica L106. Los principales atractivos que brinda son: Un costo competitivo y la integración de conexión WiFi. Esto reduce aún más la lista de materiales de la solución a un total de 27 elementos. Se

---

utiliza un conversor de USB a UART para poder programar el chip. El FT232R es escogido para esta labor. Es un conversor muy completo, que requiere pocos componentes externos y cumple el estándar de USB2.0. La alimentación se puede presentar desde el mismo conector USB, o desde un jack para un cargador externo. Recibe voltajes entre los 4.5 y los 5.5v. Este voltaje se regula por medio del TLV1117, un LDO de Texas Instruments. Después se utiliza el Hx4002, un conversor DC/DC tipo bomba de carga que con solo 3 componentes externos eleva los 3.3v a 5v, pero de una manera regulada. Esto permite que los dispositivos analógicos tengan un comportamiento estable.

El firmware en este trabajo tiene una fuerte influencia del estado del arte y las nuevas tendencias. Proyectos como TinyGo y Micropython demuestran que se está trabajando en el desarrollo de soluciones de más alto nivel para programar microcontroladores. El principal beneficio es brindarle a los usuarios un tiempo de desarrollo y una curva de aprendizaje más rápida, perdiendo el menor rendimiento posible. TinyGo es un compilador basado en LLVM que traduce código de alto nivel (Go), a código binario, entendible y ejecutable por una unidad limitada como el microcontrolador. Hasta ahora las puertos sobre los cuales se puede correr son limitados, pero tiene un gran potencial. Por otro lado Micropython es una re-implementación de python 3 para correr sobre microcontroladores. Uno de sus principales atractivos es lo interactivo que puede llegar a ser, mediante su propia línea de comandos. Sobre todo para pruebas y depuración se vuelve particularmente útil. A diferencia de TinyGo, tiene puertos para la familia de micros ESP, lo cuál lo hace perfecto para implementar en el trabajo.

La conexión inalámbrica por medio de WiFi hace la solución particularmente escalable. Lo hace apto para esquemas de IIOT y domótica. Con la correcta implementación de arquitectura de redes y software, se podría realizar un monitoreo y control de procesos de manera remota y amigable. De seguir evolucionando la idea y la arquitectura, se podría llegar a un verdadero impacto regional, a nivel educativo y comercial. Se genera como una iniciativa para apoyar la producción local de soluciones electrónicas completas (De hardware y de software). Esto radica en precios más asequibles y mayor confianza de los consumidores locales. Factores que potenciarían drásticamente la distribución de la solución.

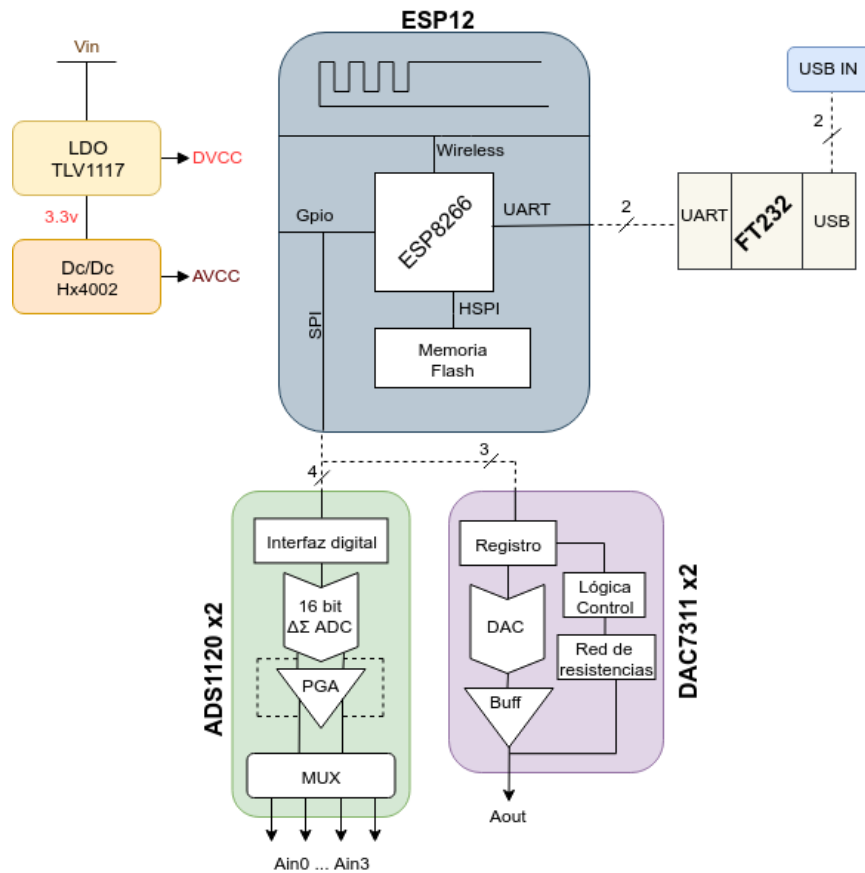


Figura 1-1.: Representación de la solución mediante diagrama de bloques.

## 1.2. Estructura del documento

El capítulo dos respecta al marco teórico y el estado del arte. Introduce los fundamentos de la adquisición de datos, de los bloques que lo componen y un par de referencias sobre las tendencias actuales del mercado en estos temas. También expone tres topologías principales en la adquisición de datos. Por último menciona equipos referentes en el mercado para la adquisición de datos a bajo costo, junto con el estado del arte a nivel investigativo en este tema. Exponiendo dos trabajos particularmente interesantes y una lista de otros más, muy relacionados también al caso de estudio.

El capítulo tres expone la propuesta de solución, muestra los beneficios y las características del sistema. Se explican bloque por bloque las razones por las que se han escogido los componentes del hardware de la solución propuesta, así como sus especificaciones técnicas. Se presenta los resultados y un par de pruebas y verificaciones técnicas que se le realizaron a la tarjeta.

El capítulo cuatro trata del Firmware. Expone detalladamente el funcionamiento de las entradas y las salidas. Se expone el firmware y se explica cómo puede interactuar con las entradas o salidas de la tarjeta. Menciona también las capacidades inalámbricas de la tarjeta y finalmente un par de casos de uso. Allí, se ejemplifican casos particulares de adquisición con el dispositivo.

Finalmente, el capítulo cinco muestra las conclusiones del trabajo, e introduce el cómo este trabajo puede influir al entorno regional a nivel educativo y comercial. También expone el trabajo futuro que queda abierto, y cómo se puede seguir mejorando la arquitectura y la implementación.

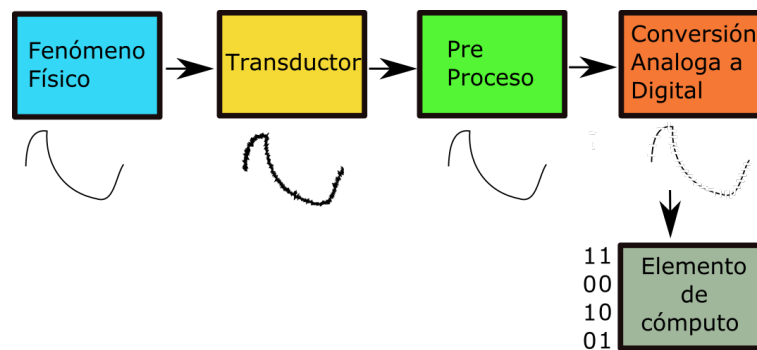


## 2. Marco teórico y estado del arte

Este capítulo habla sobre los conceptos claves en la adquisición de datos. Para ello se detiene en cada sección del proceso de adquisición de datos y da un pequeño resumen. Así mismo, busca dar luz sobre las alternativas actuales en el mercado para suplir cada uno de los temas. También, se mencionan las diferentes topologías existentes para la adquisición de datos. Por último se habla sobre los dispositivos comerciales de bajo costo para la adquisición de datos, y los trabajos científicos que se han desarrollado en esta área.

### 2.1. Fundamentos de la adquisición de datos

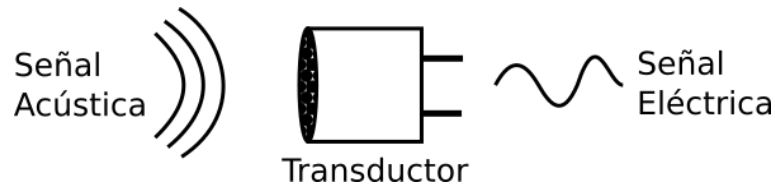
Constantemente a nuestro alrededor están sucediendo cientos de fenómenos físicos, unos que tienen más relevancia para nosotros que otros. Tenemos una capacidad limitada para observar y actuar frente a estos fenómenos, esto ha incentivado a buscar avances tecnológicos enfocados en esta área. El primer elemento clave en el esquema de adquisición son los sensores o transductores, cuya señal de salida es acondicionada y hecha viable para su posterior digitalización. La digitalización se realiza mediante conversores analógico-digitales, su salida en formato digital es entregada a una unidad de cómputo. Esta se encarga de realizar el procesamiento/análisis o gestionar su comunicación/almacenamiento. Cada uno de los pasos en este esquema será revisado a continuación.



**Figura 2-1.:** Diagrama de bloques del proceso de adquisición de datos.

### 2.1.1. Sensores y transductores

En general, son dispositivos designados para adquirir información de un objeto y transformarlo en una señal eléctrica [9]. Existen diferentes tipos, según la aplicación y según el tipo de señal analógica que tienen como salida (Los más comunes es cambio de resistencia y generación de pequeños voltajes).



**Figura 2-2.:** Acción de un transductor que convierte una onda acústica en una señal eléctrica.

También, gracias a las nuevas tendencias tecnológicas, vemos como han aparecido conceptos como sensores inteligente. La principal diferencia del sensor inteligente, es que tienen integrada toda la etapa de adquisición (Generalmente mediante un microcontrolador), y entregan la señal en formato digital [22].

Dentro de los fenómenos más comunes sujetos a medidas nos podemos encontrar con los siguientes:

- Presión.
- Nivel.
- Flujo.
- Temperatura y calor.
- Humedad.
- Iluminación.
- Movimiento y posición.
- Peso y fuerza.
- Campo magnético.
- Vibraciones.

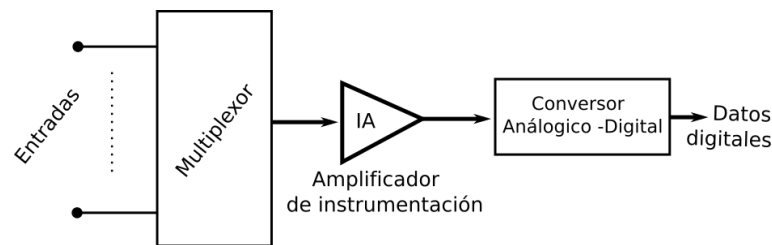
Cada tipo de medición tiene su forma de interacción y registrar el fenómeno. Una de las maneras más comunes de sensar los pequeños cambios de resistencia es mediante el puente de

wheatstone. Esto es gracias a tres resistencias fijas y una variable, donde los cambios sacan al puente del equilibrio, y se produce una diferencia de potencial en dos de sus extremos centrales [15]. Esta técnica es altamente utilizada en el sensado de fuerza y peso.

Una de las principales técnicas para el trabajo con sensores en la industria es mediante el lazo de corriente. Lo que busca es que en vez de medir una señal de voltaje, se mida una señal de corriente. Esto asegura que no hayan pérdidas de información (Que en este caso sería tensión) debido a la longitud y la resistencia de los cables. Finalmente, podemos resaltar que pueden estar sujetos a fenómenos de ruido, aún más cuando se encuentran en ambientes hostiles. Para lidiar con esto existen las etapas de acondicionamiento de señal.

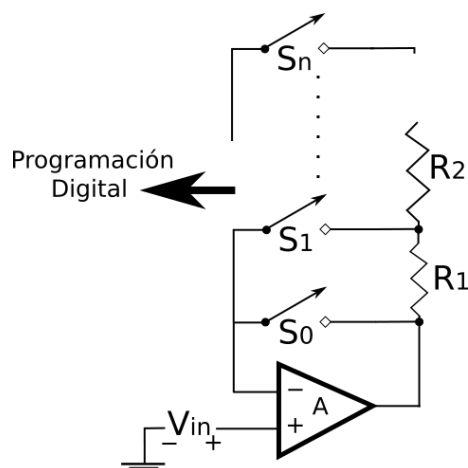
### 2.1.2. Acondicionamiento de señal

El acondicionamiento de señal se basa en, a partir de una salida de un sensor, llevar dicha señal a una formato adecuado para su transmisión, conversión o visualización [31]. Normalmente consiste de componentes electrónicos que realizan funciones tales como: Amplificación, atenuación, filtrado, manejo de impedancias, entre otros. El principal elemento que encontramos en el acondicionamiento es el **Amplificador de instrumentación**, que tiene grandes ventajas en el manejo de señales como la alta impedancia de entrada y el rechazo en modo común[23].



**Figura 2-3.:** Esquema simple de adquisición de datos compuesto por un multiplexor de entrada, un amplificador de instrumentación y un conversor analógico digital.

Una de las configuraciones más comunes para la adquisición de datos, es mediante un multiplexor y un amplificador de instrumentación, que van conectados directamente al conversor analógico-digital. Esto permite reducir los costos, y aumentar drásticamente el número de entradas. Otro dispositivo común es el amplificador de ganancia programable, que permite con un solo amplificador y una señal de control digital, seleccionar una ganancia deseada.



**Figura 2-4.:** Representación del amplificador de ganancia programable (PGA).

La etapa de filtrado es más dependiente del tipo de aplicación. Gracias a la potencia que logran los dispositivos actuales de cómputo, no es descabellado pensar en integrar filtros digitales, aunque siguen siendo muy común utilizar filtros analógicos. Dentro de estos filtros analógicos nos podemos encontrar con filtros tanto activos como pasivos. Es muy importante mencionar que muchos de los ADC actuales cuentan con algún tipo de acondicionamiento incluido, principalmente en la amplificación. La herramienta más común para ello es una PGA. Y mediante los mismos registros del conversor se puede seleccionar el valor de ganancia.

Aunque existen muchos fabricantes de amplificadores de instrumentación, hay dos compañías que dominan el mercado, por su amplia gama de productos y por sus precios competitivos. Estas empresas son **Texas Instruments** y **Analog Devices**. Es común encontrar empaquetados que contienen internamente uno, dos o hasta cuatro amplificadores. Ambos fabricantes ofrecen, dentro de su portafolio, alternativas de bajo ruido, bajo offset, bajo consumo de potencia o rango amplio de voltajes de entrada. Mientras mayores sean las prestaciones de los componentes mayor es el costo. Aunque existen dispositivos con precios superiores a los 15 USD, la gran mayoría se encuentran en el rango de los 5 USD.

### 2.1.3. Conversión analógica - digital

Una vez el fenómeno físico se haya registrado mediante una señal eléctrica gracias al transductor, y se haya ajustado esa señal mediante el acondicionamiento, se puede proceder a realizar la conversión a formato digital. Este dato digital, de naturaleza discreta, es una aproximación del valor analógico verdadero en un instante particular de tiempo (Esta sección va soportada en [17]). Aunque los parámetros que nos encontramos a la hora de escoger un ADC son muchos, algunos de los más importantes son: La resolución, el rango de adquisición, el

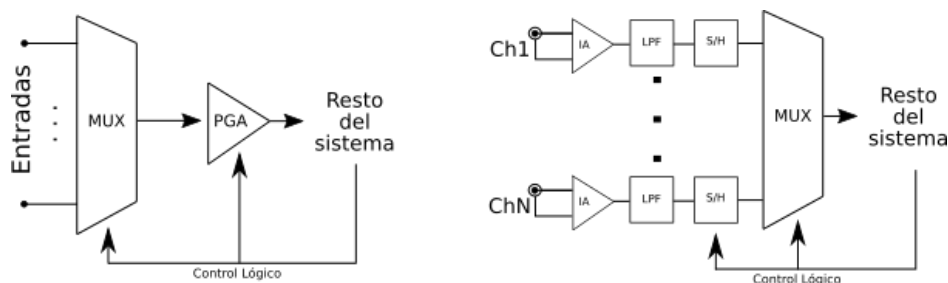
número de canales, la cantidad de muestras por segundo y el tipo de comunicación que ofrece.

La resolución representa la cantidad de dígitos binarios con los que realizará la conversión, esta cantidad se distribuye de manera discreta a través del rango de adquisición. Respecto a la cantidad de canales es importante hacer referencia al tipo de arquitectura utilizada para ello, si existe un convertor por cada canal, o un multiplexor y un solo convertor. Esto afecta directamente la cantidad de muestras adquiridas por segundo. Las comunicaciones normalmente se presentan en algún formato serial como I2C o SPI, aunque también existen convertidores con salida paralela.

Todos estos parámetros van enmarcados en el tipo de arquitectura del convertor. Es lógico pensar que a mayor resolución, es más difícil obtener un número elevado de muestras por segundo. Aunque existen muchas arquitecturas para realizar este proceso, las más utilizadas en el mercado son la **SAR: Registro de aproximaciones sucesivas**, y **delta-sigma**. A continuación nos encontramos con una pequeña comparativa entre ambas:

Arquitectura	Rango de resolución	Velocidad de conversión	Consumo
Delta-Sigma	16 - 24 bits	1 sps - 128ksps	Medio
SAR	8 - 16 bits	10ksps - 2Msps	Bajo

La tendencia es que encapsulados cada vez más pequeños integren más funcionalidades. Cada día se encuentran con dispositivos que requieren menos componentes externos para funcionar. Unos de los esquemas internos para las entradas que más nos encontramos actualmente son el de multiplexor con ganancia seleccionable y el de retención de muestra por cada canal:



**Figura 2-5.:** Esquemas internos comunes: Multiplexor con ganancia seleccionable y circuito para retención de muestra

Existen dispositivos para aplicaciones específicas, de bajo costo, que son muy utilizados en el mercado, como el HX711 (para celdas de carga). Es importante mencionar dos empresas muy fuertes en el mercado como lo son **Texas Instruments** y **Analog Devices**. Este par de empresas ofrecen muchas alternativas con variedad de empaquetados y funcionalidades, que permiten ajustarse a una amplia gama de presupuestos, además de una muy buena

documentación y guías de aplicación. Los dos valores de resolución que más nos encontramos es de 16 y 24 bits con arquitecturas respectivas SAR y Delta-Sigma. De ahí, tal vez el factor que más entra a jugar en el precio es la velocidad de adquisición. Los precios normales están al rededor de los 6 USD, pero muy fácilmente se pueden elevar a los 15 USD, en dispositivos con altas prestaciones y funcionalidades integradas.

#### 2.1.4. Elemento de cómputo

En el marco de las altas funcionalidades integradas, tal vez los elementos de cómputo son los mayores exponentes. El mercado está inundado de nuevos productos y tecnologías, que cada vez son más rápidas, pequeñas e interconectadas entre sí [1]. Específicamente para la adquisición de datos nos podemos encontrar tres plataformas muy comunes: Los MCU (unidades Micro-Controladoras), las FPGA (arreglo de compuertas programable) y los DSP (procesadores digitales de señal). Las principales características de cada uno se exponen a continuación:

Plataforma	Programación	Costo	Rendimiento	Consumo	Versatilidad
MCU	Sencilla	Bajo	Medio	Bajo	Media
FPGA	Compleja	Alto	Alto	Moderado	Alta
DSP	Intermedia	Intermedio	Alto	Moderado	Media

Cada una de las plataformas tiene posibilidad de comunicación digital (la más común es serial) con los ADC. Además, cada una tiene sus beneficios particulares, y sus grandes exponentes. Gracias al bajo costo y la capacidad de suplir la necesidad de este desarrollo en particular, la plataforma escogida mediante MCU.

Un microcontrolador es una unidad de cómputo con cantidad limitada de memoria y una variedad de entradas y salidas [2]. El método básico de programación de microcontroladores es mediante código ensamblador, aunque generalmente se trabaja en lenguaje C. Cabe resaltar que también existen alternativas de más alto nivel, a manera de firmware, que se ven reflejadas en un tiempo de prototipado más rápido.

Hasta hace unos cuantos años el tamaño del bus dentro de los Microcontroladores se limitaba a los 8 o tal vez 16 bits. Y aunque siguen siendo muy comunes en el mercado, las unidades con 32 bits han marcado una tendencia durante los últimos años. Una de las razones es los precios tan competitivos que tienen respecto al rendimiento que ofrecen. Por encima tal vez del tamaño del bus, un factor que ha revolucionado en la actualidad el mundo de los microcontroladores es la arquitectura. Una de las más reconocidas es la ARM, un computador con set de instrucciones reducido (RISC) [14]. Esta arquitectura ha ido evolucionando considerablemente con los años [45], y se han creado tres familias diferentes:

- La familia A, optimizada para aplicaciones de alto rendimiento.
- La familia R, diseñada para aplicaciones de tiempo real.
- La familia M, que busca una balance entre rendimiento, consumo y costo.

Algunos de los fabricantes que merecen ser resaltados por su variedad y gran reconocimiento en el mercado son **Texas Instruments**, **NXP**, **Microchips** y **ST Microelectronics**. Cada uno de ellos ofrece una gama de opciones que conjugan los factores de precio y prestaciones, y generando un buen espectro para escoger.

En estos tiempos donde la Industria 4.0 y el IOT son tendencia, fuera del procesamiento y la capacidad de cómputo se vuelve muy importante la conectividad. Varios fabricantes como **Silicon Labs** o **Espressif** se han dado cuenta de esto, generando soluciones de alta conectividad inalámbrica. En uno de sus artículos [5], Silicon Labs habla del diseño de sistemas con unidades MCU con capacidades inalámbricas. Menciona lo apropiadas que son las plataformas de 32 bits para este tipo de aplicaciones, gracias a su capacidad de cómputo y trabajo con unidades de punto flotante (FPU). Realza el hecho que los diseños se simplifican mucho gracias a la selección de componentes integrados de altas funcionalidades. Esta visión y concepto es altamente compartido y explotado en este trabajo. El precio de estos dispositivos es competitivo, y permite integrarlo muy bien en soluciones empotradas de costo asequible. Algunos módulos funcionales del mercado se pueden conseguir por precios inferiores a los 5 USD.

## 2.2. Topologías de adquisición de datos

Los sistemas de adquisición de datos en conjunto con las unidades de cómputo y procesamiento brindan una flexibilidad considerable en su configuración. Esto obliga a que sea el mismo número de sensores y actuadores requeridos, su ubicación física, el tipo de acondicionamiento de la señal requerido y la hostilidad del ambiente quien determinen la configuración apropiada para cada caso particular<sup>1</sup>. Algunas de las configuraciones más comunes son:

- Módulo I/O de conexión.
- Módulo I/O distribuido.
- Módulo independiente.

### 2.2.1. Módulo I/O de conexión:

Los módulos o tarjetas de conexión tienen como principal característica que se conectan directamente a la unidad de cómputo central (Por alguno de sus buses para conexión de

<sup>1</sup>Esta sección tiene fuerte influencia de [28], [12] y [34]

periféricos), lo cuál lo posicionan como la manera más rápida y fácil de adquirir datos en computadores. Estas tarjetas pueden ser de propósito específico, por ejemplo, de acondicionamiento y adquisición de datos analógicos, de salidas analógicas, de carácter digital, de instrumentación, de control de motores, entre otras. o multi-funcional, teniendo una combinación de algunos de estos. El esquema general para los multi-funcionales se caracteriza por entradas y salidas tanto analógicas como digitales. Los módulos de conexión generalmente se usan cuando el computador se encuentra cerca de los sensores y actuadores, o cuando existe un esquema de conexión remoto de transductores, mediante estándares industriales de transmisión de este tipo de señales.

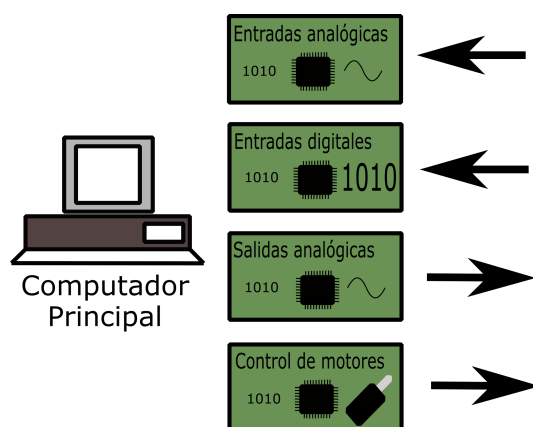


Figura 2-6.: Esquema de configuración: Módulos I/O de conexión.

### 2.2.2. Módulo I/O distribuido:

Cuando los sensores tienden a estar alejados, o en un ambiente más hostil, se generan varias implicaciones sobre las señales, en especial si son pequeñas señales como las entregadas por termocuplas o celdas de carga, incapaces de sobrevivir la transmisión sobre largas distancias. Es acá donde los módulos distribuidos juegan un papel importante, porque aunque tienden a tener un costo más elevado por su misma estructura, resuelven los problemas mencionados. Con módulos distribuidos se plantea la estrategia de ubicar para cada sensor, o conjunto de sensores, una unidad de adquisición y de transmisión digital. Esto genera distintas estaciones, bien sea de propósito general o de propósito particular, y es un medio digital quien interconecta todas con una unidad de interfaz, que ya está directamente conectada al computador principal. El medio digital comúnmente está soportado en estándares industriales de comunicación como lo son RS-232, RS-485 y ModBus.



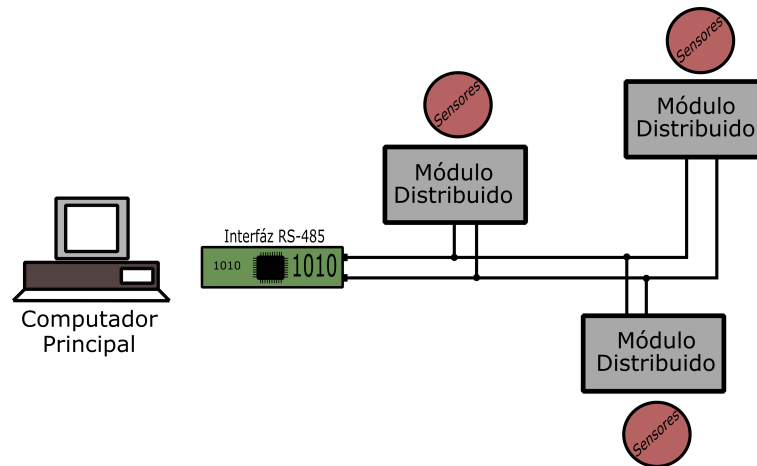


Figura 2-7.: Esquema de configuración: Módulos I/O distribuido.

### 2.2.3. Módulo independiente:

Como su nombre bien lo dice, un módulo independiente está diseñado para funcionar sin la necesidad de la conexión de un computador. Para ello, generalmente cuenta con métodos de programación y con salidas, analógicas o digitales según sea el caso, y poder controlar el proceso analizado. Estas características lo hacen particularmente adecuado en aplicaciones remotas o ambientes hostiles. Algunos de estos módulos independientes funcionan solamente para almacenar datos, y permiten extraer dicha información en alguna unidad de memoria externa para después ser llevada y analizada en un computador. Con los recientes avances en tecnologías inalámbricas también es común encontrar en estos módulos independientes esta funcionalidad, permitiendo extraer los datos sin necesidad de una conexión física, o subiéndolos constantemente a la nube.

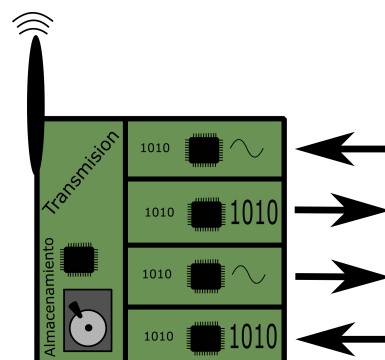


Figura 2-8.: Esquema de configuración: Módulos independientes.

## 2.3. Mercado e investigaciones relacionadas a la adquisición de datos

Lo primero a analizar, antes de establecer una propuesta de solución, es qué existe en el mercado actualmente para suplir las necesidades. A continuación se muestra una recopilación de algunos dispositivos de bajo costo para la adquisición de datos, cuyo precio se presenta en dólares(USD). A partir de la información entregada por esta comparativa, en la siguiente sección se plantea la solución, con las modificaciones correspondientes para lograr su ajuste.

Serie	Entradas analógicas			Salidas analógicas		Precio
	Canales	Resolución	Muestreo	Canales	Resolución	
USB-200	8se	12 bits	500kS/s	2	12 bits	99
USB-230	8se/4diff	16 bits	100kS/s	2	16 bits	260
MCC118	8se	12 bits	100kS/s	-	-	99
MCC152	-	-	-	2	12 bits	99
1208FS	8se/4diff	12-14 bits	50kS/s	2	10-12bits	129
1608FS	8se	16 bits	100kS/s	-	-	420
1608G	16se/8diff	16 bits	500kS/s	2	16 bits	420
E-1608	8se/4diff	16 bits	250kS/s	2	16 bits	505
1208LS	8se/4diff	12 bits	47kS/s	2	12 bits	189
Temp-tc	8diff	24 bits	2S/s/ch	-	-	375
E-tc	8diff	24 bits	4S/s/ch	-	-	505
2408	16se/8diff	24 bits	1kS/s	2	16 bits	630
1208HS	8se/4diff	13 bits	1MS/s	4	12 bits	525
DT9812	8se	12 bits	100kS/s	2	12 bits	415
DT9813	16se	12 bits	100kS/s	2	12 bits	495
DT9814	24se	12 bits	50kS/s	2	12 bits	570
DT9816	6se	16 bits	750kS/s	-	-	510
DAS08	8se	12 bits	40kS/s	-	-	280
DAS16	16se/8diff	16 bits	1kS/s	2	12 bits	525

Este es un resumen, el resto de las características, especificaciones y recursos se puede encontrar en [6]. Aunque son dispositivos de bajo costo, podemos observar que siguen siendo de precios que inician al rededor de los 100 USD, y que se elevan drásticamente en los dispositivos de altas prestaciones.

A raíz de esto, varias personas han trabajado y desarrollado artículos científicos en esta área. Al ser un tema mayormente comercial, la producción literaria es muy baja con respecto a la cantidad de productos en el mercado. Hay una marcada preferencia por las aplicaciones con fines educativos. A continuación se mencionan los trabajos relacionados que son relevantes

para el presente trabajo.

El primero es un trabajo de maestría del instituto técnico de Lisboa, presentado por Francisco José Antunes Marques para optar por el título de magíster en Ingeniería Electrónica. El título del proyecto es: "Diseño e implementación de un sistema de adquisición de datos con propósitos de enseñanza" [11]. Allí se propone una única sección de entradas analógicas, que incluyen acondicionamiento. La unidad controladora es un dispositivo PIC, que, mediante una interfaz serial entrega los datos al computador. La solución cuenta con 16 bits de resolución para la adquisición, a una velocidad máxima de 500kS/s, aunque esto no se aclara del todo.

El otro se titula: "Sistema de adquisición de datos de bajo costo y open-source para medidas precisas" [27]. Se desarrolló en la universidad de Oviedo en Gijón, España. El sistema planteado cuenta con 8 entradas analógicas a 16 bits y con un voltaje máximo de 12V. La frecuencia de muestreo es de 20kS/s. Tiene un canal de salida analógica a 14 bits con un valor máximo de 4V. También posee un generador de señales. Las dimensiones son pequeñas, pero sigue teniendo un costo considerable, al rededor de los 270 dólares. Provee una serie de drivers en python para su configuración y operación, además de herramientas gráficas, desarrolladas también en python. Tal vez lo más relevante de este proyecto es que evolucionó hasta el punto de poder crear una propia línea de productos. Tienen una página web, con una tienda incluida, donde se presentan diferentes versiones de su producto denominado Open-DAQ [7], accesorios, y documentación relacionada a la tarjeta.

Este es un claro ejemplo de los alcances y las posibilidades que tiene este tipo de trabajos. De igual, aunque estos son los más relevantes en este caso particular, existen varios trabajos también relacionados [44] [37] [29] [16] [3] [39] [19] [24] [33] [40] [30] [35] [26] [21] [20].

## 3. Propuesta de solución

Se propone el diseño de un sistema de adquisición de datos bajo tres premisas fundamentales:

- Un esquema reconfigurable y versátil, amigable para el usuario.
- Conectividad inalámbrica, que permita la adquisición remota de datos en configuraciones de IOT e IIOT.
- Un bajo costo (Menor a los 50 USD), manteniendo niveles de resolución y velocidad de adquisición aptos para control industrial.

Para lograr esto, se hace uso de un microcontrolador con capacidades inalámbricas incluidas, y un Firmware Open-Source de alto nivel. La solución cuenta con ocho canales analógicos de entrada de 16 bits y 2 canales de salida analógicos de 12 bits. Además cuenta con un bus de expansión mediante el protocolo I2C.

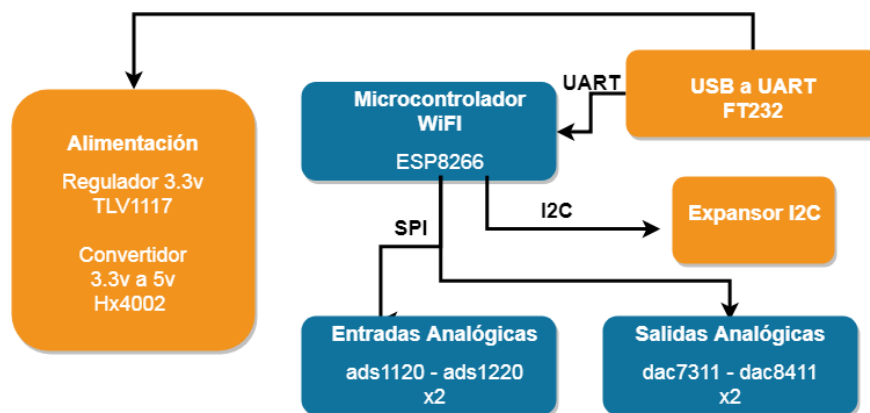


Figura 3-1.: Arquitectura de la solución.

### 3.0.1. Beneficios

- Alimentación por fuente unitaria de 5V.
- Conexión inalámbrica de Wi-Fi mediante una antena en la misma PCB.
- Bajo número de componentes (27 componentes).

- Bajo costo (Aprox 20 USD).
- Tamaño reducido de la tarjeta (Aprox: 6.7cm x 7.2cm).
- Firmware abierto y modificable.
- Ocho canales de adquisición analógica de 16 o 24 bits.
- Dos canales de salida analógica de 12 o 16 bits.

### 3.0.2. Especificaciones del sistema

Parámetro	Condición	Especificación			Unidad
		Min	Typ	Max	
Voltaje de alimentación	Normales	4.5	5	5.5	V
Número de canales de entrada	Fija	4 dif	8 se	8 se	-
Voltaje de entrada	Dif	-5	-	5	V
	Se	0	-	5	V
Resolución	Depende del conversor	16	16	24	bits
Ganancia	Normales	1	-	128	V/V
Frecuencia de muestreo	Normales	20	-	1.2k	Sps
Número de canales de salida	Fija	2			-
Voltaje de salida	Normales	0	-	5	V
Resolución	Depende del conversor	12	12	16	bits
Temperatura de operación	Normales	-40	-	85	grados c
Dimensiones	Fija	6.7 x 7.2			cm

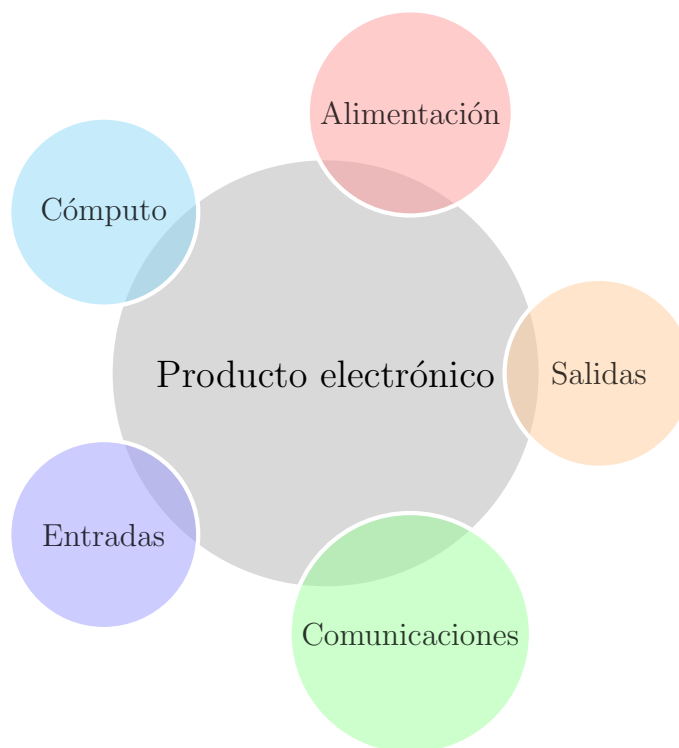
### 3.0.3. Descripción del sistema

Un sistema de adquisición de datos tiene como objetivo medir fenómenos del mundo físico. Para ello hace uso principalmente de elementos transductores, que convierten un fenómeno físico en una señal eléctrica medible, y conversores analógico-digitales, que se encargan de digitalizar esta señal. En el mercado existen diferentes soluciones de adquisición de datos, tanto para propósitos específicos, como para propósito general.

En la tabla comparativa de la sección anterior se pueden encontrar algunas alternativas de bajo costo para la adquisición de datos. Allí se presentan las características principales de estos dispositivos. A partir de esto se pueden construir unos parámetros de referencia para

definir las especificaciones del sistema a diseñar. La principal característica que no se presenta en ninguno de estos dispositivos y se quiere aprovechar en este diseño es la conectividad de red inalámbrica WiFi. Esto genera un valor añadido en estos tiempos donde resuenan fuertemente conceptos como el Internet de las cosas y la Industria 4.0.

La solución final consta de un capítulo de Hardware y uno de Firmware. En el hardware se exponen los componentes y el diseño que le brinda las características del sistema. En el firmware se expone cómo comunicarse con los periféricos, y realizar configuraciones más complejas. Para el diseño del hardware podemos partir de los conceptos fundamentales del diseño electrónico: Los bloques principales para generar un producto o diseño electrónico son la alimentación, la estructura de cómputo, las entradas, las salidas y las comunicaciones.



Al ser un sistema de adquisición de datos, debe existir un énfasis mayor en las entradas del sistema y la estructura de cómputo que las soporte. Para realizar la comunicación entre las interfaces de entrada/salida y la unidad de cómputo se opta por utilizar un protocolo de comunicación serial SPI. La representación de las conexiones físicas se expone a continuación:

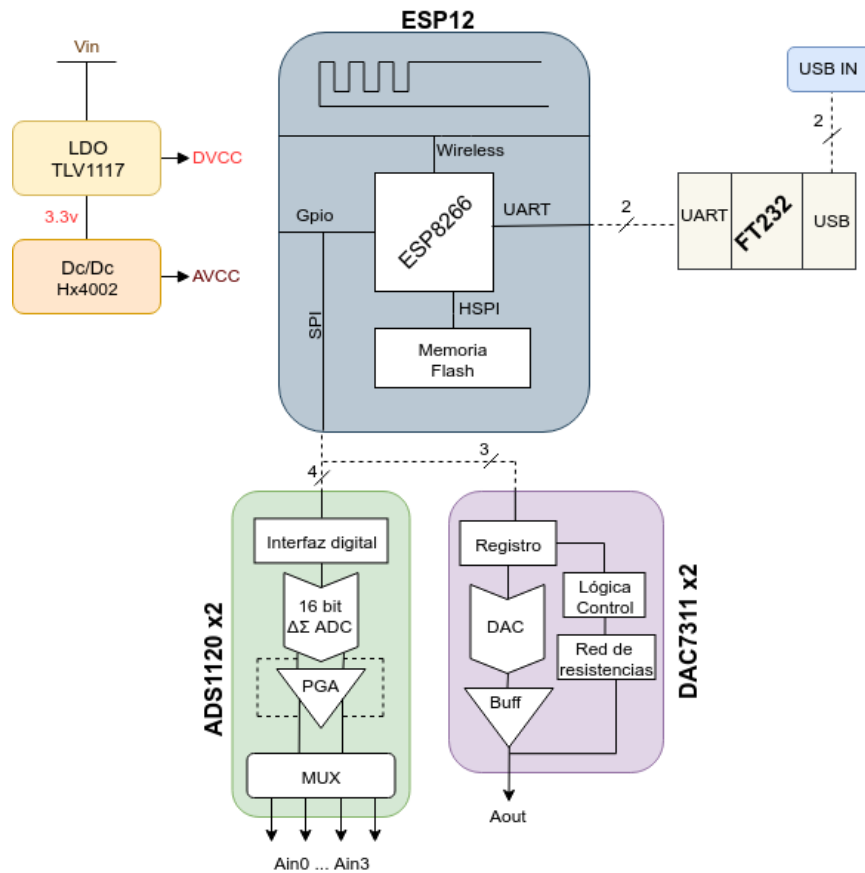


Figura 3-2.: Representación de la solución mediante diagrama de bloques.

## 3.1. Hardware

Como se mencionaba anteriormente, a nivel de hardware los bloques fundamentales son: Alimentación - Estructura de cómputo - Entradas - Salidas - Comunicaciones. A continuación se procede a explicar cada uno de ellos:

### 3.1.1. Entradas

Gracias a su naturaleza, es el principal factor en un sistema de adquisición de datos. En la solución propuesta las entradas son de carácter analógico. La adquisición de señales analógicas se fundamenta en la conversión a formato digital, realizada por un convertor analógico-digital. Los principales factores tomados en cuenta a la hora de elegir el convertor fueron: La resolución, el rango de tensión de entrada, la frecuencia de muestreo y el número de canales.

Según la comparativa de los dispositivos para adquisición de datos de bajo costo presente en la sección anterior, la resolución para la adquisición analógica está entre los 12 y los 16 bits.

Aunque existen algunos con 24 bits, suelen tener frecuencias de muestro mucho más bajas debido al cambio de arquitectura interna del convertor. Los rangos de tensión que manejan estos dispositivos suelen ser de 0-5V y 0-10V, ampliamente utilizados en aplicaciones generales e industriales. El número de canales se encuentra entre 4 y 8. También existen excepciones de 16 o más canales, pero tienden a elevar los costos. Respecto al rango de frecuencias varía entre 1kS/s y 500kS/s.

Partiendo de estas referencias comerciales se realiza la elección del convertor para este diseño. El dispositivo seleccionado es el ADS1120 de Texas Instruments<sup>1</sup>, un convertor de 16 bits, cuatro canales de entrada y una velocidad máxima de adquisición de 2kS/s. Un factor interesante sobre este dispositivo es que es compatible y reemplazable por el ADS1220, que ofrece características similares pero con 24 bits de resolución. Primero se presenta el ADS1120, que cuenta con las siguientes características relevantes:

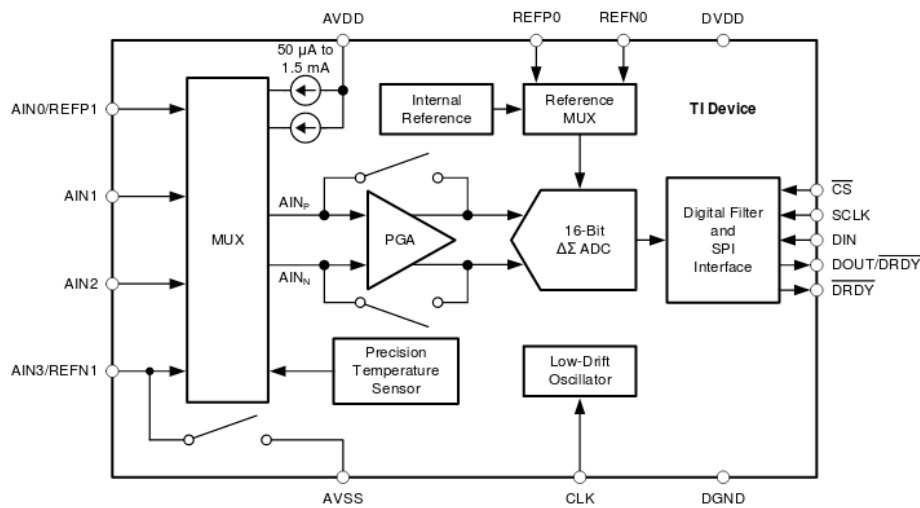
- Cuatro canales de entrada, que se pueden configurar como single-end, o como dos canales diferenciales.
- Comunicación mediante protocolo serial SPI.
- Fuente de alimentación unitaria entre 2.3V y 5.5V, referencia integrada y entradas en rangos estándares de 0-Vcc.
- Ganancia programable entre 1V/V y 128V/V, mediante PGA interna.
- Frecuencia de muestreo hasta de 2kS/s.
- Bajo consumo de corriente (Consumo típico de 240 $\mu$ A en modo normal), junto con rangos de temperatura industriales (-40 a 125 grados centígrados).

La estructura interna simplificada de este convertor es la siguiente:

---

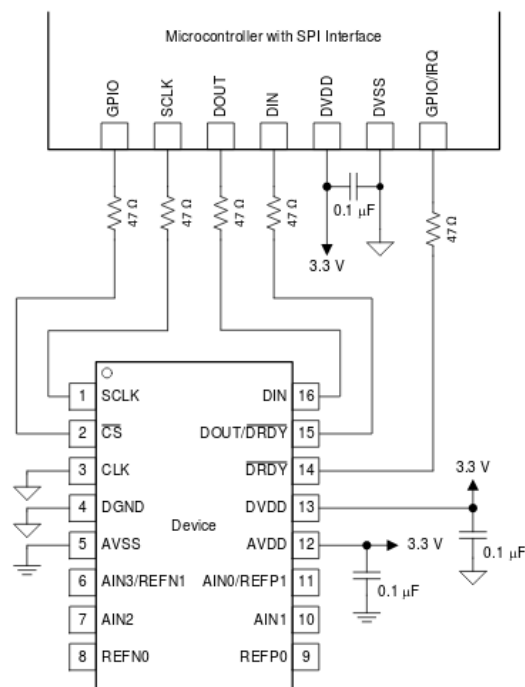
<sup>1</sup>Hoja de datos del dispositivo [43]





**Figura 3-3.:** Estructura simplificada del convertor A/D ADS1120.

El ADS1120 está concebido para que la cantidad de componentes externos requeridos para su funcionamiento sea mínima. Entre ellos se encuentra una fuente de alimentación apropiada y el respectivo soporte de comunicación en la estructura de cómputo para su correcta configuración y entrega de datos.



**Figura 3-4.:** Diagrama de conexión del convertor A/D ADS1120.

El conversor de reemplazo es el ADS1220 de la empresa Texas Instruments. Al ser de la misma familia, la estructura interna, las características y los componentes necesarios para su funcionamiento son los mismos. La diferencia principal radica en el hecho que este conversor es de 24 bits. Con las mismas palabras de configuración y registros, lo que habría que cambiar específicamente en el firmware del dispositivo es la función para leer un valor. Se debe ajustar para recibir 24 bits en vez de 16. La comunicación con esta familia de dispositivo es mediante interfaz SPI, con cuatro registros y seis comandos. Cada comando tiene una longitud de 8 bits. Los comandos son:

Comandos	Descripción	Bytes a enviar
Reiniciar	Reinicia el dispositivo	0000 011x
Conv	Inicia o reinicia la conversión	0000 100x
Apagar	El dispositivo entra en modo apagado	0000 001x
Rdata	Solicita el registro de datos	0001 xxxx
Rreg	Lee nn registros, iniciando en la dirección rr	0010 rrnn
Wreg	Escribe nn registros, iniciando en la dirección rr	0100 rrnn

En la tabla, donde se pone una  $x$  hace referencia a que ese bit no tiene importancia. Respecto a los registros de configuración podemos mencionar que son 4, necesitando un total de 2 bits para representarlos, iniciando en 00 y terminando en 11. También es importante resaltar que tienen una longitud de 8 bits. En cada registro se pueden definir diferentes configuraciones del dispositivo según la siguiente tabla:

Registro(Hex)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	Mux[3:0]			Ganancia[2:0]			PGA	
01h	DataRate[2:0]		Modo[1:0]		CM	TS	BCS	
02h	Vref[1:0]	Filtro[1:0]		PSW	IDAC[2:0]			
03h	I1Mux[1:0]		I2Mux[1:0]			DrdyM	0	

Para escribir en cualquier registro se debe iniciar enviando el comando de escribir (0100rrnn) junto con el identificador del registro ( $rr$ ). Luego se envían los 8 bits correspondientes a los datos o configuraciones en dicho registro. La tabla y descripción completa de cada registro se puede encontrar en la correspondiente hoja de datos.

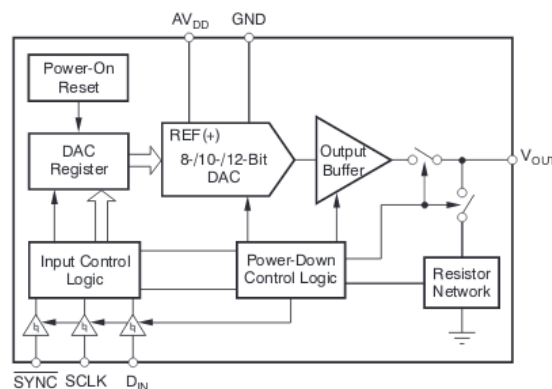
### 3.1.2. Salidas

Las salidas en el sistema también son de naturaleza analógica. La transformación de una señal digital a una analógica se realiza mediante un conversor digital-analógico(DAC). Los principales factores que se tomaron en cuenta a la hora de elegir un conversor fueron: La resolución, el rango, el tipo de salida y el número de canales.

Respecto a la comparativa comercial presentada en la sección anterior, muchos dispositivos no cuentan con salida analógica, y los que si tienen, manejan una resolución entre 12 y 16 bits. La salida se presenta en formatos de voltaje y en rangos de 0-5v y 0-10v. El número de canales es limitados, generalmente no más de dos. Como alternativa para realizar la salida analógica se propone el dispositivo DAC7311 de la empresa Texas Instruments<sup>2</sup>. Un convertor digital-analógico de 12 bits, que tiene dispositivos que lo pueden reemplazar totalmente. En este caso el posible reemplazo es el DAC8411, con la misma configuración de pines y arquitectura, pero con 16 bits de resolución a su salida. Las principales características del convertor DAC7311 son las siguientes:

- Convertor de un canal a una resolución de 12 bits.
- Buffer amplificador interno con operación Rail to Rail.
- Rango de alimentación entre 2 y 5.5V, con un bajo consumo de corriente.
- Comunicación digital mediante protocolo SPI.
- Rango de temperaturas industriales (-40 a 125 grados centígrados).

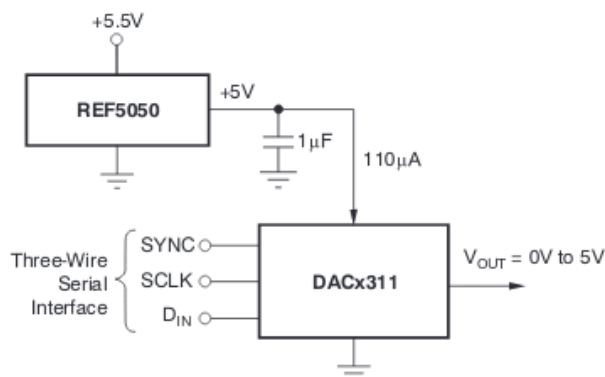
Con este dispositivo se puede implementar estrategias de control enviando la salida deseada en forma de voltaje. La estructura simplificada interna de este convertor es la siguiente:



**Figura 3-5.:** Estructura simplificada del DAC7311.

Este dispositivo proporciona las salidas en los niveles de 0V a Vdd. Siguiendo los lineamientos de implementación proveídos por su fabricante, este dispositivo cubre la sección de salidas analógicas. Los lineamientos vienen de la siguiente manera:

<sup>2</sup>Hoja de datos del dispositivo [42]

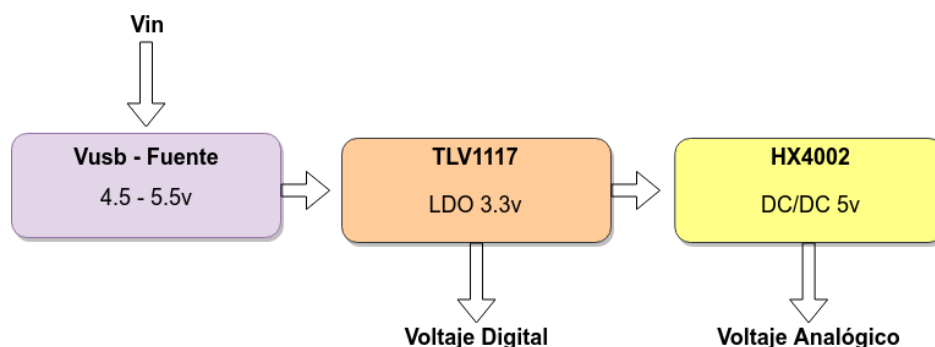


**Figura 3-6.:** Esquema de conexión del DAC7311.

La comunicación es mediante interfaz SPI. Este dispositivo no cuenta con registros de configuración, sólo requiere que se envíe un único paquete con la información correspondiente a la salida deseada. La información completa se encuentra también en la hoja de datos. Además, en la sección de firmware, se expande el formato y la manera de solicitar un valor de salida.

### 3.1.3. Alimentación

La sección correspondiente a la alimentación tiene una función principal: Entregar los rieles de voltaje apropiados para cada sección del diseño. El diagrama de alimentación es el siguiente:

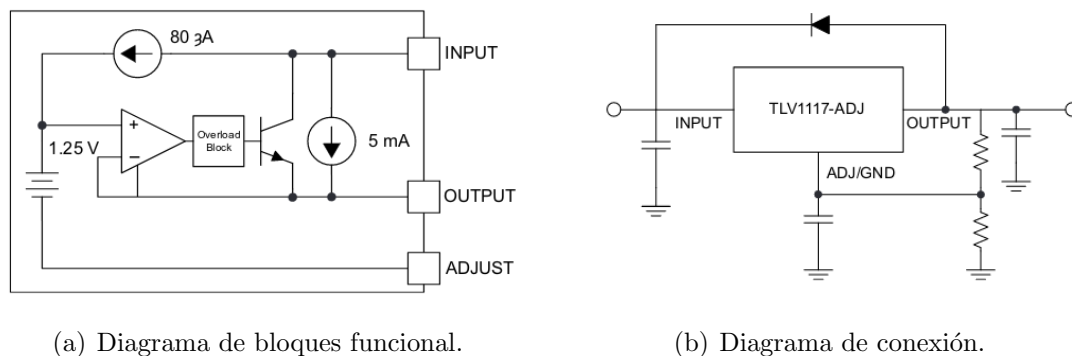


**Figura 3-7.:** Diagrama de alimentación del circuito.

La entrada principal del sistema es de 5v, de ahí se generan los rieles de alimentación analógicos y digitales. El riel de alimentación digital es de 3.3v. El circuito integrado encargado de generar esta salida es el TLV1117 de la empresa Texas Instruments<sup>3</sup>. Es un regulador de voltaje de baja pérdida que viene en diferentes versiones, con salida fija y ajustable. En este

<sup>3</sup>Hoja de datos del dispositivo [41]

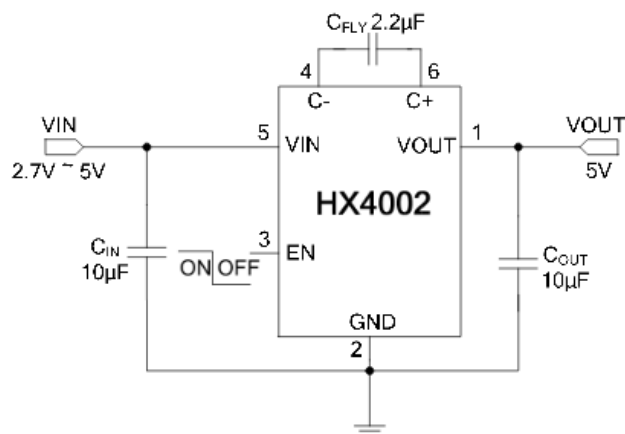
caso se escoge la versión de salida fija de 3.3v. Este regulador puede entregar una salida de corriente de hasta 800mA y presenta una caída máxima de voltaje de 1.3v a ese nivel de corriente. Como componentes externos requiere un capacitor de desacople a la entrada y uno a la salida.



**Figura 3-8.:** Diagrama de bloques funcional y de conexión del TLV1117.

Respecto al rieles que tienen como destino los dispositivos analógicos se requiere una salida de tensión de 5v. La calidad de esta alimentación determina el correcto y preciso funcionamiento de los conversores. Por esta razón se debe optar por una forma de regularla. Una solución sencilla y de pocos componentes es mediante el dispositivo HX4002 de la empresa Hexin<sup>4</sup>. Un convertor DC/DC tipo bomba de carga, de bajo ruido, con un voltaje de operación de entrada entre los 2.7V y los 5V, y una salida fija regulada de 5V y una corriente máxima superior a los 250mA. Tiene también protección contra corto circuito. Al ser tipo bomba de carga no requiere inductores, por lo que el espacio ocupado en la PCB es pequeño. Como componentes externos requiere los capacitores de desacople y el capacitor de carga. A continuación se presenta un diagrama de conexión del dispositivo:

<sup>4</sup>Hoja de datos del dispositivo [18]



**Figura 3-9.:** Diagrama de conexión del dispositivo HX4002.

Con estos dos rieles de alimentación, tanto el analógico como el digital se supe la alimentación total del sistema.

### 3.1.4. Estructura de cómputo

La estructura de cómputo debe encargarse de manejar la comunicación con las entradas y las salidas analógicas. Incluye también la sección de almacenamiento, ejecución de programa y gestión de conexión inalámbrica. Con el fin de reducir el número de conexiones de la estructura de cómputo con las entradas y las salidas se plantea una comunicación serial. Se escoge el estándar SPI, con el correspondiente *CS* (Del inglés *chip select*) hacia cada dispositivo. La estructura de cómputo planteada es un microcontrolador, que soporte velocidades de reloj por encima de los 80Mhz. Con esta velocidad se puede soportar un Firmware de alto nivel, y realizar las operaciones dentro de los tiempos de adquisición. Al gestionar las conexiones inalámbricas, la estructura de cómputo está fuertemente ligada con la sección de comunicaciones generales del sistema.

La tendencia de incluir varias funcionalidades dentro de un solo chip influye fuertemente en el desarrollo de nuevos microcontroladores. Esta ventaja se puede explotar a la hora de escoger la estructura de cómputo. Un fabricante que está invirtiendo mucho en esta área es Espressif. Una empresa de semiconductores establecida en el 2008 con sede en Shanghai. Se enfocan en el desarrollo de soluciones en el ámbito del IOT con tecnologías inalámbricas. Son los creadores de una familia de chips llamada ESP, soluciones en formato SOC (System on Chip), en forma de microcontroladores con conexión inalámbrica. Una de las intensiones principales de estos dispositivos es que sean soluciones altamente integradas, es decir, que la cantidad de componentes externos sea reducida. Gracias a estas características se decide implementar como estructura de cómputo un microcontrolador de esta familia. Se escoge el

ESP8266<sup>5</sup>, que cuenta con las siguientes características:

- Conexión WiFi bajo el protocolo 802.11 b/g/n e implementación TCP/IP.
- Posibilidad de conexión de antena en la misma PCB.
- Procesador TensilicaL106 de 32 bits.
- Interfaz de periféricos para UART/SPI/I2C/I2S entre otros.
- 9 pines GPIO disponibles para el usuario.
- Bajo consumo de potencia: Aproximadamente 80mA a un voltaje de 3.3v.
- Rango de temperatura industrial (desde -40 hasta 125 grados).

Dentro de la gama de aplicaciones para este dispositivo podemos encontrar el control industrial inalámbrico. El chip implementa un total de 17 GPIO que pueden ser designados para diferentes labores. Dependiendo del pin se pueden multiplexar con funciones como I2C, UART, PWM, entre otros. Este dispositivo integra unidades y controladores de memoria, aunque para el almacenamiento de programa debe tener una memoria flash SPI externa. Con la memoria conectada, la programación se realiza mediante una interfaz serial UART. La velocidad de transmisión por esta interfaz puede alcanzar los 4.5Mbps. Con el fin de cargar el programa mediante USB debe existir un conversor de USB a UART presente. En este caso se escoge el circuito FT232R, una interfaz muy completa<sup>6</sup>. Unos de sus principales atractivos es que tiene el cristal interno que genera la señal de reloj, lo cual hace que se requieran pocos componentes externos para funcionar. Además cumple totalmente con los estándares y especificaciones del USB 2.0.

---

<sup>5</sup>Hoja de datos del dispositivo [10]

<sup>6</sup>Hoja de datos del dispositivo [13]

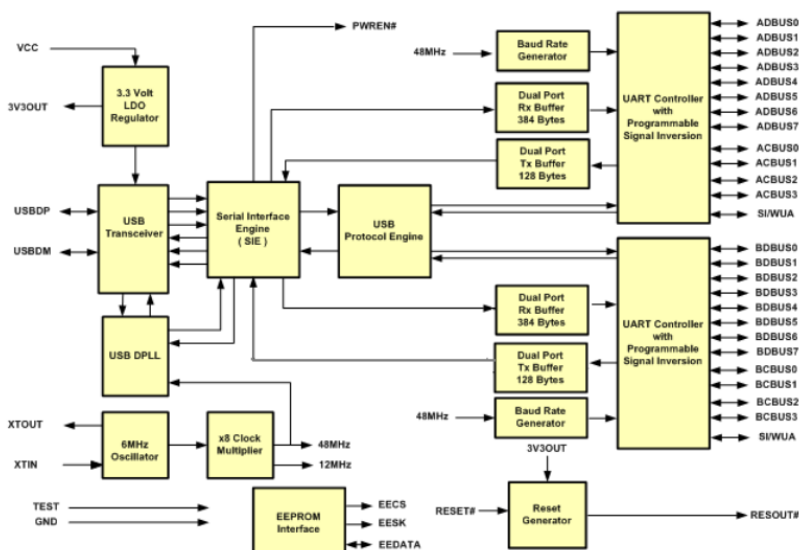


Figura 3-10.: Arquitectura del chip FT232R.

En el computador anfitrión se requiere un controlador específico para reconocer este circuito interfaz, una vez instalado, se puede acceder y empezar a comunicar con él.

Respecto al ESP8266, Espressif provee un SDK para el trabajo, con su propio firmware específico. Aquí se propone una manera novedosa de trabajo: Implementar un firmware Open Source, que se mencionará en la correspondiente sección. El ESP8266 viene además en diferentes módulos que ya integran funcionalidades como la memoria interna para la carga de programa y el cristal del oscilador. Estos módulos suelen ser de bajo costo y reducen aún más la lista final de materiales del sistema. Específicamente se escogió el ESP-12, del cual uno de sus principales atractivos es tener integrada ya la antena para las conexiones inalámbricas. Con una serie de resistencias y capacitores para el desacople, este dispositivo queda listo para trabajar.

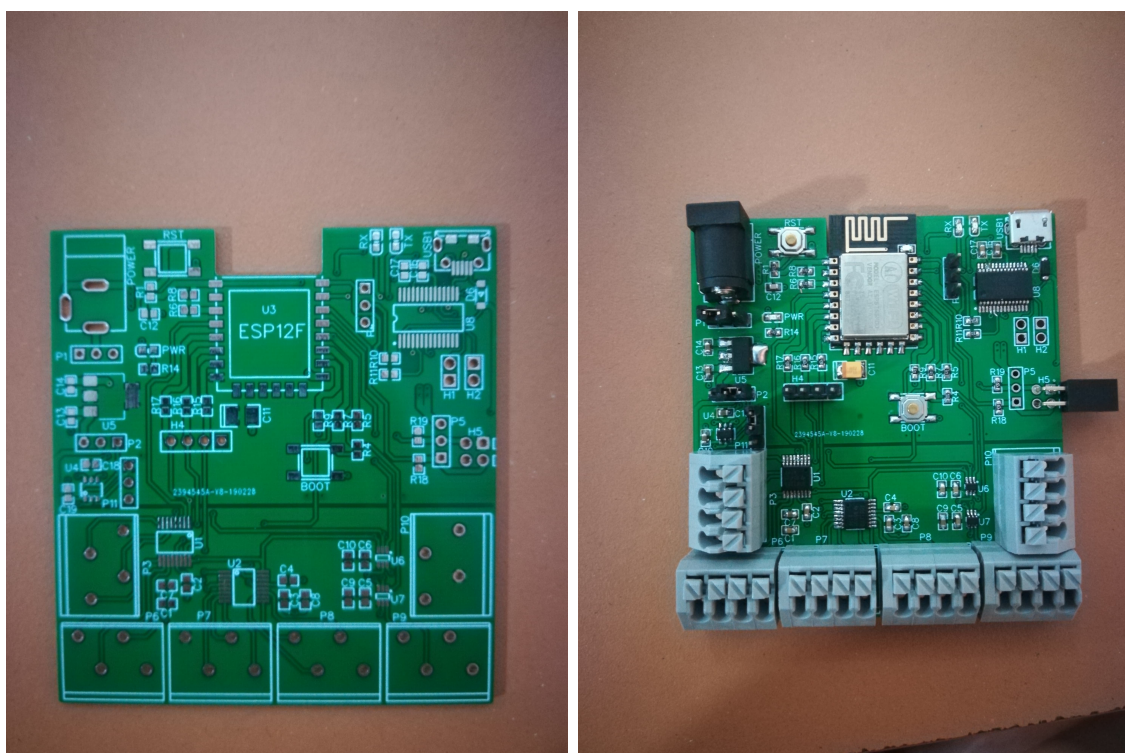
### 3.1.5. Comunicaciones

Las comunicaciones se pueden dividir en las conexiones físicas y el protocolo que las maneja. Las interfaces de comunicación externa son: Inalámbricas y USB. La parte inalámbrica quedó cubierta a nivel de chip con el dispositivo ESP8266 que cumple directamente con el protocolo 802.11 b/g/n de Wi-Fi en la banda de los 2.4GHz. Respecto al chip encargado de realizar la comunicación USB, es mencionado en la sección de Cómputo, el FT232. Referente a los protocolos de comunicación internos de la tarjeta se presentan en la sección de Firmware.



## 3.2. Resultados

Como resultado del diseño de hardware, se obtiene una tarjeta para la adquisición de datos. La tarjeta cumple con las especificaciones presentadas a lo largo del documento. La PCB se construyó por la compañía JLBPCB, un fabricante chino de tarjetas electrónicas que ofrece precios competitivos. Todo el ensamble se realiza a mano, gracias a los empaquetados elegidos a la hora de diseñar. El tiempo del proceso depende de la técnica que tenga la persona en el proceso de soldadura. Aunque es importante mencionar que la lista de materiales lo hacen mucho más sencillo (27 componentes).



(a) PCB.

(b) Tarjeta soldada.

Figura 3-11.: Resultados: PCB sin componentes y tarjeta final ensamblada.

## 3.3. Pruebas y verificación técnica

Para verificar el cumplimiento de las funcionalidades descritas, se realizan una serie de pruebas. Es importante mencionar que esta sección implementa todas las rutinas propuestas bajo el firmware mencionado en la siguiente sección. Como instrumento de medida calibrado para realizar las pruebas utilizaremos el dispositivo *Analog Discovery 2*, de la empresa Digilent. Se puede definir como un osciloscopio portable USB de un costo muy competitivo. Cuenta con dos canales de entrada en el osciloscopio capaces de muestrear a una velocidad de 100MS/s

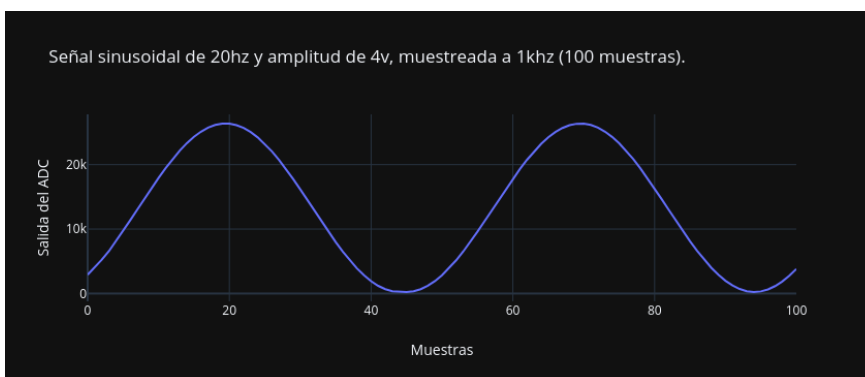
a una resolución de 14 bits. Dos canales para el generador de señales, también a 100MS/s y 14 bits de resolución. Analizador lógico de 16 canales entre otros beneficios.

Los tres parámetros principales a medir tanto en las entradas como en las salidas es: Precisión, frecuencia y rango. Empezaremos por las entradas. Para ello haremos uso del generador de señales del Analog Discovery. La primera prueba se realizó bajo condiciones normales. Se genera una señal sinusoidal de 20hz, una amplitud de 4v pico a pico, iniciando en cero. Se toma un segundo de la señal a una frecuencia de 1khz. El resultado es el siguiente:



**Figura 3-12.:** Señal sinusoidal, frecuencia 20hz, amplitud 4v, muestreo a 1khz.

Se ve una señal muy clara y limpia. De igual manera se muestra un acercamiento de 100 muestras.



**Figura 3-13.:** 100 muestras de una señal sinusoidal, frecuencia 20hz, amplitud 4v, muestreo a 1khz.

La segunda señal tiene una frecuencia un poco mayor. Es una señal cuadrada con frecuencia de 50hz y amplitud de 4v pico a pico, iniciando en cero. Se realiza un muestreo a una velocidad de 1khz. El resultado es el siguiente:



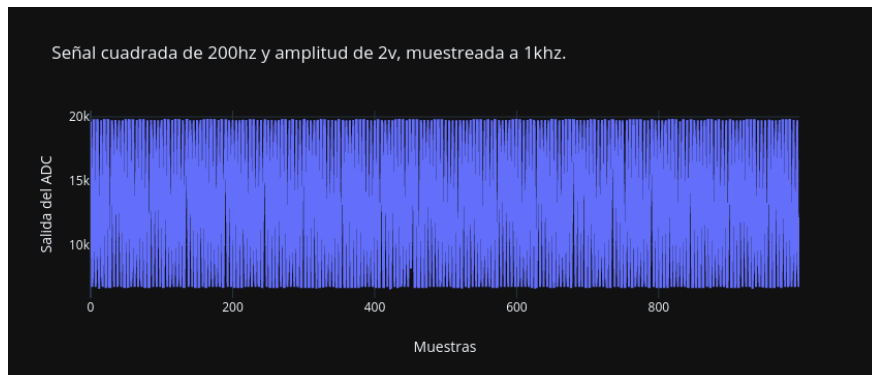
**Figura 3-14.:** Señal cuadrada, frecuencia 50hz, amplitud 4v, muestreo a 1khz.

Se realiza un zoom para visualizar mejor la señal. Se presentan las primeras 50 muestras:



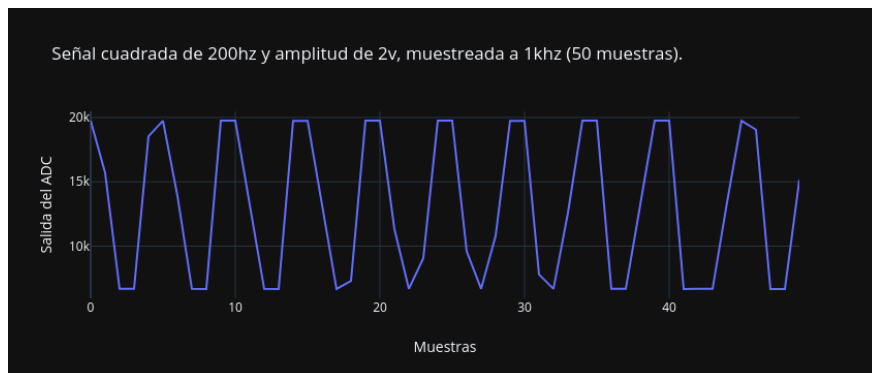
**Figura 3-15.:** 50 muestras de una señal cuadrada, frecuencia 50hz, amplitud 4v, muestreo a 1khz.

Se logra visualizar que aún está bien formada la señal pese al aumento de frecuencia. Se busca una señal aún más exigente por lo cuál se ingresa otra señal cuadrada, pero con una frecuencia de 200hz. La amplitud se reduce a 2 y se agrega un offset de 1v. La velocidad de muestreo continúa en 1khz.



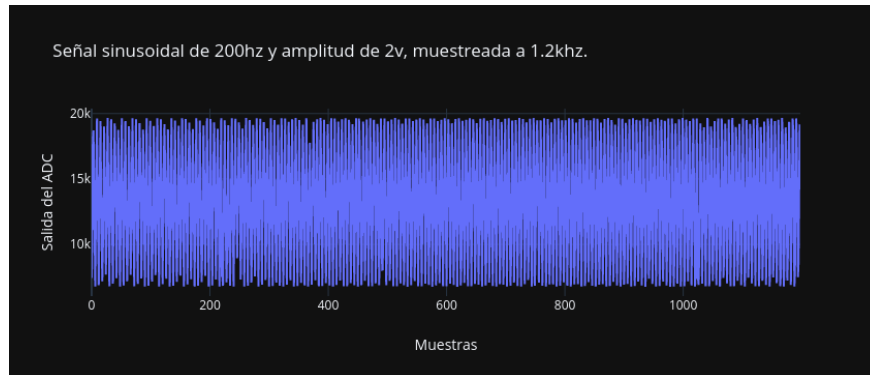
**Figura 3-16.:** Señal cuadrada, frecuencia 200hz, amplitud 2v, muestreo a 1khz.

Al acercar la imagen sobre las primeras 50 muestras se observa que a una frecuencia mayor de la señal y manteniendo la misma frecuencia de muestreo, se generan menos puntos de representación por cada periodo. La señal se empieza a deformar.



**Figura 3-17.:** 50 muestras de una señal cuadrada, frecuencia 200hz, amplitud 2v, muestreo a 1khz.

Al intentar realizar la misma prueba de 200hz pero con una señal sinusoidal se presentan problemas. Siendo una señal más compleja, y con tan pocos puntos de representación, se deforma mucho. Para esto se decide llevar la frecuencia de muestreo de la tarjeta al límite en 1.2kS/s.



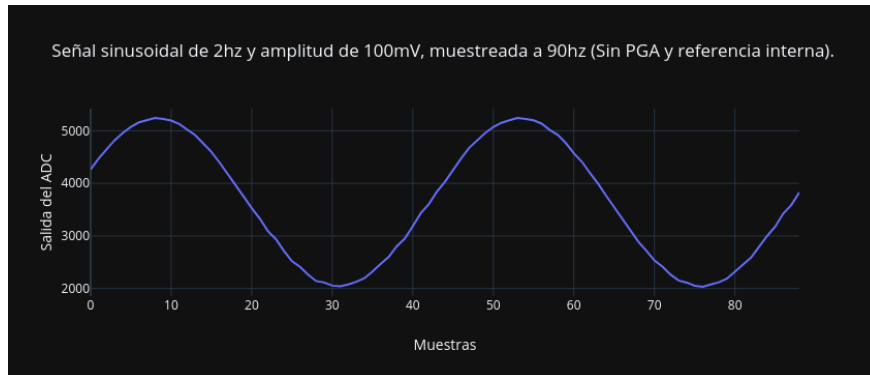
**Figura 3-18.:** Señal sinusoidal, frecuencia 200hz, amplitud 2v, muestreo a 1.2khz.

Para efectos de visualización se realiza un acercamiento sobre 50 muestras. Es muy similar a la señal cuadrada, aunque se alcanza a distinguir la forma de la senoide.



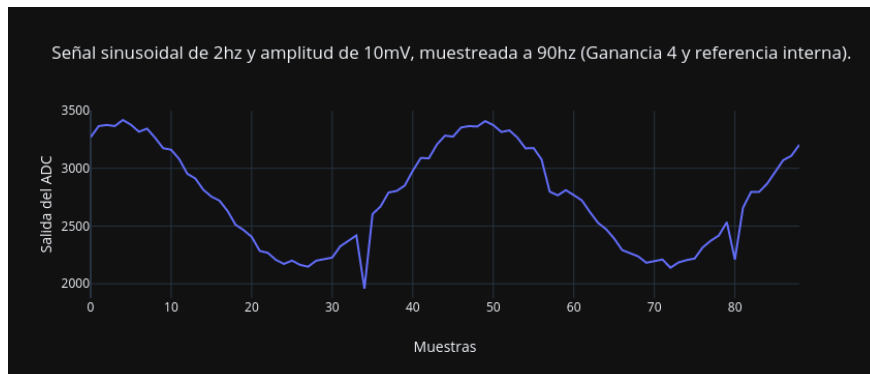
**Figura 3-19.:** 50 muestras de una señal sinusoidal, frecuencia 200hz, amplitud 2v, muestreo a 1.2khz.

Las pruebas finales se realizan con el fin de medir la precisión de lectura. Se utiliza una señal sinusoidal de una frecuencia de 2hz. La señal tiene una amplitud pico a pico de 100mV. Para reducir el ruido de lectura se decide disminuir la frecuencia de muestreo a 90hz. Gracias a que es una señal pequeña, se puede utilizar la referencia interna del ADC. El resultado es el siguiente:



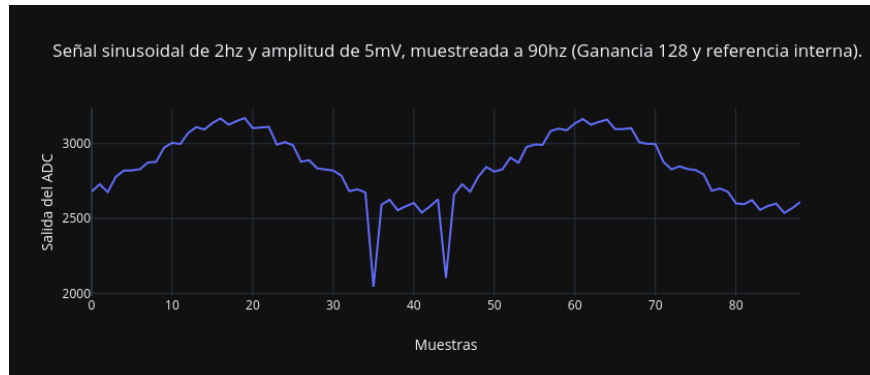
**Figura 3-20.:** Señal sinusoidal, frecuencia 2hz, amplitud 100mv, muestreo a 90hz.

Se visualiza la señal de una manera muy clara y definida. Se opta por reducir más la amplitud de la señal. Se selecciona sobre los 10mV pico a pico. La frecuencia de la señal se mantiene sobre los 2hz y la de muestreo en 90hz. Se utiliza una ganancia de 4, aún sin la PGA y se hace uso de la referencia interna.



**Figura 3-21.:** Señal sinusoidal, frecuencia 2hz, amplitud 10mv, muestreo a 90hz.

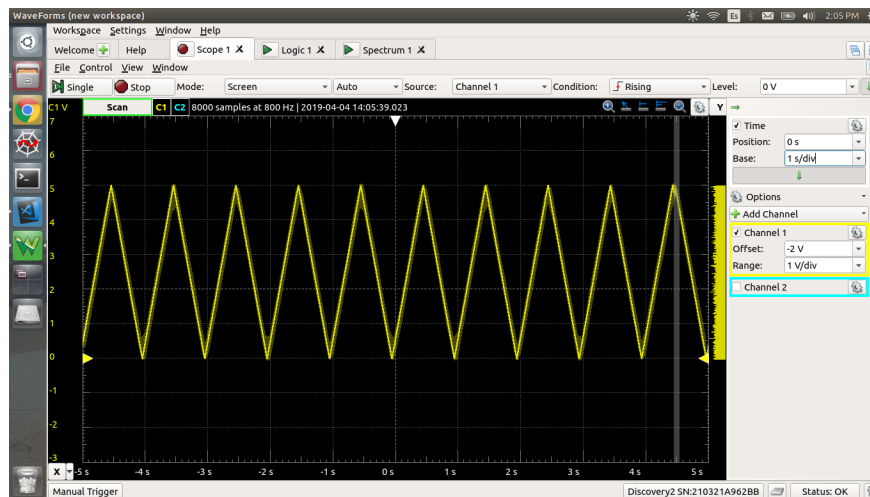
Se observa ya una señal menos definida, con más ruido y con un par de picos extraños. Para finalizar las pruebas de precisión se utiliza la amplitud mínima que ofrece el dispositivo Analog Discovery. Se presenta la señal con tan solo 5mv pico a pico y una frecuencia de 2hz. La frecuencia de muestreo se mantiene constante en 90hz y la referencia interna. Se utiliza la PGA con una ganancia de 128.



**Figura 3-22.:** Señal sinusoidal, frecuencia 2hz, amplitud 5mv, muestreo a 90hz.

Los puntos de representación para esta señal cada vez son más pocos. Se observa muy ruidosa, menos definida e igual con picos extraños. Cabe resaltar que se utilizan borneras y cables estándar para conectar la salida del generador a la tarjeta. Esto tiende a inducir mucho ruido en la misma señal. Con esta aclaración los resultados muestran mayor validez y entendimiento.

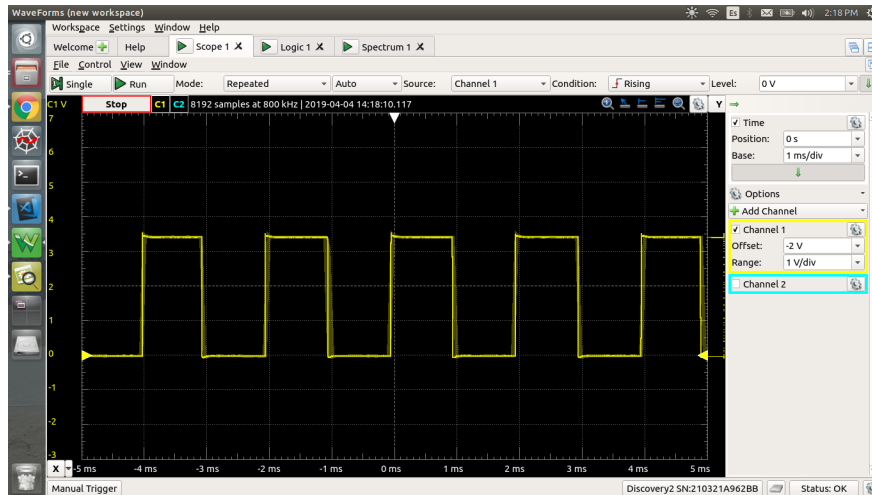
Con las entradas finalizadas se muestran los resultados de las salidas. Se utiliza el DAC para representar unos casos específicos, y se capturan los datos mediante unos de los canales de osciloscopio del Analog Discovery. El primer test representa el rango completo de la salida, generando una señal triangular de 5v pico a pico, empezando en 0. Se escoge una frecuencia de 1hz.



**Figura 3-23.:** Generación de una señal triangular con el DAC. Amplitud 5v y frecuencia 1hz.

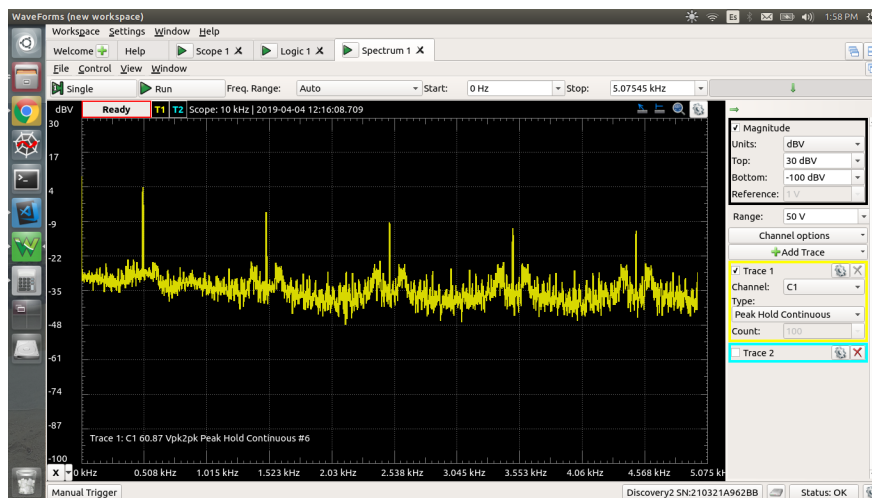
La explicación de cómo se genera este tipo de señal se presenta como caso de uso en la sección de Firmware. Los resultados son claros. Se decide cambiar el tipo de señal por una

onda cuadrada. Se aumenta la frecuencia a 500hz y se reduce la amplitud aproximadamente a 3.5v.



**Figura 3-24.:** Generación de una señal cuadrada con el DAC. Amplitud 5v y frecuencia 500hz.

Se presenta una señal clara y definida. Los tiempos son acordes. De igual manera se calcula el espectro de la señal y se presenta a continuación.

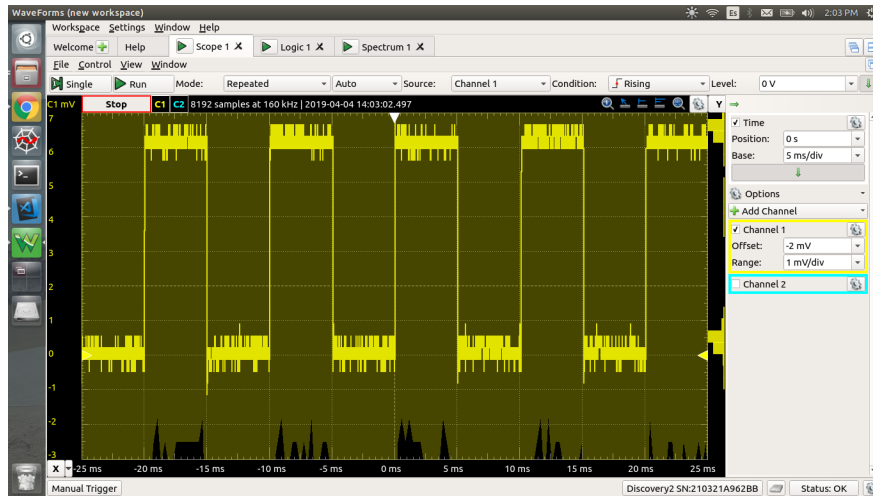


**Figura 3-25.:** Espectro de una señal cuadrada generada con el DAC, de amplitud 3.5v y frecuencia 500hz.

La frecuencia central se marca sobre los 500hz, y los armónicos se ven, como es de esperar, desde 3 veces la frecuencia de la señal en adelante. Una última prueba de precisión se realiza. Se mantiene una frecuencia de 500hz pero se decide reducir la amplitud. El DAC es de 12 bits, representados en 4096 divisiones. Si el voltaje de alimentación es de 5v, cada división



corresponde aproximadamente a unos 1.2mV. Se realiza un cambio en la salida del DAC entre 0 y 5 divisiones. La amplitud esperada de salida de la onda cuadrada sería entonces de 6mV. El resultado se muestra en la siguiente figura.



**Figura 3-26.:** Generación de una señal cuadrada con el DAC. Amplitud 6mV y frecuencia 500hz.

Se observa que los rangos esperados se cumplen. Se observa un pequeño ruido sobre el 0V y también sobre los 6mV. Esto es muy entendible considerando primero el rango sobre el que se encuentra, y segundo que la adquisición es mediante borneras y cables estándares. Como se mencionaba anteriormente esto induce un ruido en la señal. Con esta prueba concluye el análisis de funcionamiento de la tarjeta.

## 4. Firmware

El objetivo principal de este capítulo es exponer el firmware del dispositivo y mostrar la capacidad que tiene mediante unos pequeños casos de uso. Para entender el funcionamiento total, se empezará describiendo los protocolos de comunicación tanto con las entradas como con las salidas. También se presenta el Firmware general que puede usar la tarjeta, y cómo se puede modificar para las aplicaciones específicas. Por último se expondrán dos casos de uso de la tarjeta.

### 4.1. Funcionamiento del dispositivo

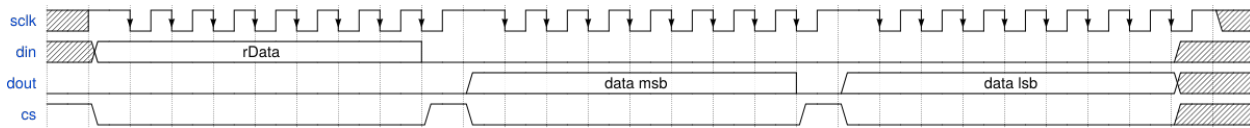
A nivel práctico, el funcionamiento del dispositivo se basa en la comunicación de la unidad de cómputo con las entradas y con las salidas. Para ello hace uso de un estándar de comunicación serial llamado SPI. Convencionalmente se basa de de un bus de 4 líneas y un esquema Maestro-Esclavo. Consta de una señal de reloj(SCK), una línea de comunicación del maestro hacia los esclavos(MOSI), una línea de los esclavos hacia el maestro(MISO) y una señal de activación del chip(CS). En este caso el ESP es el maestro, y los periféricos (Los dos ADC y los DAC) funcionan como esclavos. Es el ESP entonces quien da las ordenes de conversión y recepción de datos. A continuación se procede a detallar de una manera más explícita cómo es la comunicación con cada dispositivo.

#### 4.1.1. Lectura de entradas analógicas

En las entradas existen dos conversores, que en principio son ADS1120 y cuentan con 16 bits de resolución. Eventualmente se pueden reemplazar (Uno o Ambos) por el ADS1220, aumentando la resolución a 24 bits. El principio de lectura del ADS1120 se ve detallado en su hoja de datos. Mediante 6 comandos y 4 registros de configuración se puede realizar la interacción, estos fueron expuestos en la sección correspondiente de hardware. A continuación se expone el flujo o serie de pasos, a nivel general, para trabajar con el dispositivo. Se recomienda como primer paso realizar un reinicio mediante comando del dispositivo. El comando corresponde a 0x06(hexa), seguido de una pequeña espera que le permite al conversor completar la acción. Una vez reiniciado se deben configurar los canales a medir, la ganancia, los voltajes de referencia, los filtros, y demás parámetros requeridos. Esto se realiza mediante el comando para escritura en registro, debe enviarse la palabra de configuración apuntando

al registro, y posteriormente los 8 bits a asignar.

Este convertor cuenta con dos modos de lectura, el modo de disparo unitario y el modo de conversión continua. En el modo de disparo unitario se le debe enviar una señal de inicio para que realice el proceso de conversión. Después de la única conversión el dispositivo entra en modo de bajo poder. En el modo de conversión continua el dispositivo se encuentra siempre realizando lecturas, según la velocidad de muestreo, y carga en el registro de salida el valor de la última conversión.



**Figura 4-1.:** Diagrama de tiempos para lectura de datos con el ads1120.

Un punto que es pertinente aclarar respecto a la señal SPI que controla el funcionamiento del dispositivo es el de la polaridad y la fase. La polaridad corresponde al valor del reloj en el cuál no está activo. En el caso de estos convertores es con la señal en bajo. Por esta razón la polaridad es 0. La fase indica el flanco del reloj durante el cuál se captura el valor. Para esta familia de convertores es mediante el flanco de bajada, de acuerdo a esto la fase corresponde a 1.

El dispositivo cuenta con un pin indicador de la salida lista, que se puede usar como interrupción, pero esto implica un GPIO extra en el microcontrolador. Para evitar esto en el ESP8266, dada su limitante de pines, se puede utilizar el modo de indicación compartido, que configura el dispositivo para indicar simultáneamente por DRDY y MISO cuando el dato se encuentra listo. Para realizar se debe ajustar el byte 1 del registro 0x03h. Una vez configurado el modo de captura se envían 16 pulsos de reloj para obtener el valor de salida por el pin correspondiente (MISO). Cada pulso genera un bit. El proceso para reemplazo con el ADS1220 es muy similar, en cuanto a comandos y registros, lo único que cambia es la cantidad de pulsos que generarían una salida. En este caso serían 24 pulsos de reloj, correspondientes a cada bit a entregar.

### 4.1.2. Control de las salidas analógicas

Para realizar la interacción con las salidas se requiere también del estándar SPI, pero en este caso solo con 3 hilos. Se requiere la señal de reloj(SCKL), la entrada de datos(MOSI), y una señal de sincronización(CS). El DAC7311 cuenta con un registro de entrada de 16 bits, de los cuales, los primeros dos están reservados para el modo de operación, los siguientes doce representan la salida a obtener y los últimos dos no tienen relevancia. La secuencia de

escritura comienza poniendo el pin de SYNC en bajo, seguido de los 16 bits que se desean escribir en el registro. Con la última señal de reloj, el dispositivo se dispone a configurar el valor solicitado. En este punto es recomendable llevar SYNC a alto, para que se prepare para el valor posterior.

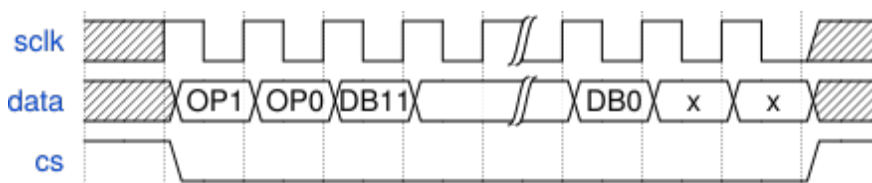


Figura 4-2.: Diagrama de tiempos para el dac7311.

La polaridad y la fase en la señal SPI también afecta a estos convertidores. El valor del reloj en el cual no está activo es en bajo, por lo tanto la polaridad es 0. El flanco del reloj en el cual se captura el valor es en el flanco de subida. La fase es 0 también.

En el caso del convertidor de reemplazo, el DAC8411 que cuenta con 16 bits de resolución, el funcionamiento es el mismo a excepción del tamaño del registro de entrada. La misma señal de sincronización da inicio a una escritura, esta vez de 24 bits, donde los primeros dos siguen siendo correspondientes al modo de operación. Los 16 bits siguientes indican el valor de la salida del convertidor. Los últimos 6 bits no tienen importancia.

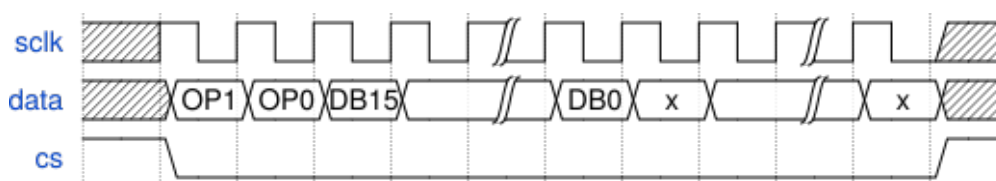


Figura 4-3.: Diagrama de tiempos para el dac8411.

## 4.2. Descripción del firmware

En el capítulo 3, específicamente se mencionaba que se buscaba implementar un novedoso firmware open-source. Este es **Micropython**, a continuación se procede a dar una breve introducción del mismo. Posterior, y gracias a la sección anterior que expresa cómo se debe interactuar con el dispositivo, se habla de la implementación de las funcionalidades de esta tarjeta en el firmware. Finalmente se habla sobre la posible integración de esas funcionalidades con la capacidad inalámbrica de la tarjeta.

### 4.2.1. Micropython

Micropython es una re-implementación liviana de python 3, diseñada específicamente para microcontroladores. Este Firmware es completamente open source. La principal ventaja es la abstracción de alto nivel que genera python. Esto lo hace mucho más amigable con el usuario. Otra ventaja notable a la hora de realizar pruebas es la consola de comandos(REPL). También presenta la posibilidad de combinar funciones nativas de C, pero implica construir de nuevo el Firmware. En términos generales, como alternativa para rápido prototipado es excelente. Al tener el Firmware cargado, y conectar dispositivo al computador, se habilita un puerto al cual podemos conectarnos.

En el sistema de almacenamiento de datos interno que genera el Micropython podemos encontrar el archivo *main.py*, el cuál será sujeto de las modificaciones de programa a ejecutar. Existen herramientas mediante consola de comandos como "ampy", diseñada por Adafruit, para enviar los archivos al microcontrolador portador del firmware Micropython. Además, sobre la plataforma específica del ESP8266 existe una herramienta llamada WebREPL: La consola interactiva pero mediante la WEB, que además del Debugg, permite transferir archivos al dispositivo.

En principio, el ESP viene con un firmware original llamado comandos AT. Para instalar el Micropython se debe iniciar la tarjeta en modo boot, limpiar la memoria flash, y cargar el RTOS. Para todo eso existe una librería de python llamada "esptool". Y la versión actual del Micropython se descarga directamente de la documentación oficial.

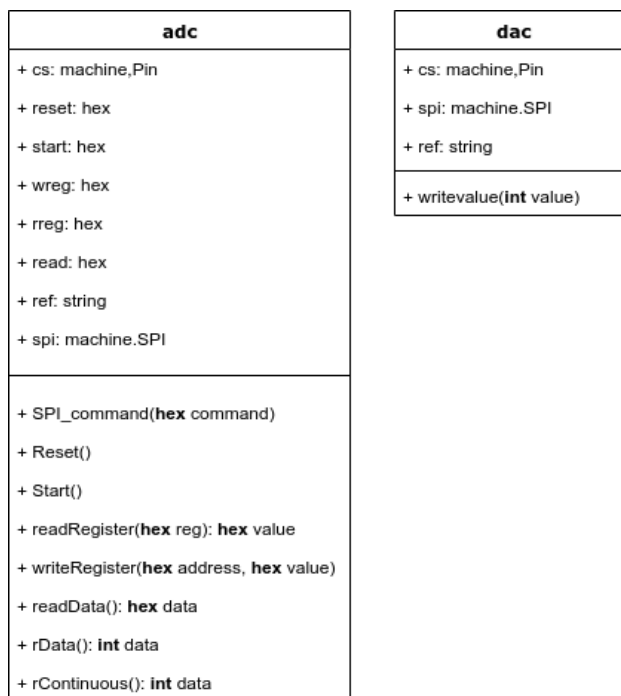
### 4.2.2. Integración con la tarjeta

Con el firmware precargado, nos permite realizar funciones y clases. La manera más elegante y ordenada de trabajar con la tarjeta es mediante clases. Aunque esto tiene sus restricciones: Al correr un sistema operativo en tiempo real y tener un intérprete, la incorrecta implementación de algunas funciones puede afectar directamente el rendimiento de la tarjeta. Para eso existen unas buenas prácticas y técnicas para optimizar el rendimiento del dispositivo. Aunque la referencia completa se encuentra en [8], es pertinente aclarar algunas de ellas:

- Evitar el uso de punto flotante: Aunque es una herramienta útil, las operaciones tienden a ser más lentas, aún más cuando no existe una unidad específica para su procesamiento. En el caso de los ADC, es mejor manejar una representación en entero mientras se pueda, y utilizar el punto flotante solo para el procesamiento de la señal.
- Utilizar arreglos: Los arreglos de bits son muy comunes y útiles, y pueden significar una mejoría en el rendimiento. Además pueden funcionar como buffers, recibiendo cadenas de datos.

- Usar el emisor de código nativo: Muchas funciones pueden ser sujetas del emisor de código nativo, esto mejora el rendimiento haciendo uso de códigos de operación nativos de la CPU.

Bajo estas consideraciones, se encuentran dos escenarios: generar una clase para los ADC y otra para los DAC, o, se puede condensar todo en una sola clase. En este trabajo se prefiere crear una clase para cada tipo de dispositivo. Todo se guarda dentro de un archivo llamado "DAQ.py". Lo principal es mirar cómo se maneja el protocolo SPI mediante Micropython. La tarjeta viene con una clase SPI por hardware y otra por software. Durante el desarrollo de la misma, se notó que el hardware SPI sólo funciona como esclavo. Esto igual podría servir, pero se deberían invertir en el diseño los pines MISO y MOSI. Por ahora se debe trabajar mediante el formato software SPI. La clase interna que maneja este estándar tiene varios métodos incluidos, pero en este caso se utilizan el de inicialización, que se ejecuta cuando se crea el objeto, el de lectura y el de escritura. Los métodos específicos que se deben crear respecto al ADC son para leer y escribir registros, y leer datos principalmente. Para el DAC, es simplemente escribir el valor que se desea como salida. A continuación se detallan:



**Figura 4-4.:** Diagrama de clases para el ADC y el DAC, contenidos dentro del archivo "DAQ.py".

La clase correspondiente al ADC viene preparada para funcionar tanto con el ads1120, como con el ads1220, se le debe especificar mediante el parámetro *ref* a cuál se hace referencia (Los dos posibles casos son 1120 y 1220). También se le debe informar cuál es el pin que funcionará como CS. Se le indica mediante un entero y la clase misma se encarga de crear el objeto

Pin, referente a ese valor. Al importar el archivo "DAQ.py", se crean automáticamente los objetos correspondientes al SPI, con la configuración correcta para cada dispositivo. Esto es tomado directamente por el objeto del ADC al ser creado. El resto de las variables que utiliza esta clase son constantes y se sacan a partir del comportamiento descrito en la hoja de datos.

En DAC es mucho más simple. Al no tener registros de configuración, sólo se debe generar una función para escribir el valor que se desea en la salida. En principio el objeto requiere un entero correspondiente al pin del CS. Igual que el ADC, toma el objeto SPI que se crea al importar el archivo, y que ya viene configurado de manera correcta. La función para escribir el valor recibe un entero. Como el conversor es de 12 bits, tiene en total 4096 divisiones, empezando desde 0 y terminando con 4095. El entero que se le entrega debe estar en este rango, de otra manera, la función no se ejecutará. La salida corresponde a la proporción de este valor en referencia a la escala completa. Como el voltaje de alimentación del DAC es de 5v, un valor de 2047 representa 2.5v, la mitad de la escala.

Por último es importante hablar sobre los **contadores** (*timers*), que se vuelven realmente útiles a la hora de adquirir datos síncronos, osea, a una velocidad fija. Cada función o método que se implementa, tiene un rango determinado de tiempo de ejecución (Al operar un sistema operativo en tiempo real). Este determina la velocidad máxima con la que funciona o adquiere el mismo. Esto, sumado a que opera un SPI mediante software, hace realmente importante determinar la velocidad real máxima a la cuál el sistema puede adquirir. El contador se debe encargar de generar la interrupción para la lectura de datos, y se debe cumplir que el periodo del contador debe ser mayor al tiempo que se toma completar la función de lectura. Este periodo del contador corresponde también a la frecuencia a la cuál se debe ajustar el conversor para muestrear. De igual manera, así la frecuencia necesaria sea menor a la máxima, los contadores ayudan a asegurar que estos intervalos se cumplan.

### 4.2.3. Capacidades inalámbricas

La versión del Micropython para el ESP8266 integra un par de librerías especiales para el trabajo con la red inalámbrica Wifi. En este caso hablaremos de dos, "Networking" "Sockets". La librería Networking tiene como principal objetivo configurar la conexión inalámbrica. Específicamente el objeto WLAN, se puede construir y configurar para acceder a una red determinada o modificar la funcionalidad de access point. Además se pueden modificar las dirección IP que obtiene el dispositivo al conectarse a una red existente. Para un esquema de tarjetas distribuidas, la mejor forma de conexión sería a una red existente, este podría ser un computador de placa unitaria (SBC), o un router del hogar, según se requiera.

La segunda librería se llama "sockets", estos se encargan de posibilitar una comunicación vía TCP sockets. Se implementa mediante un cliente o como servidor. El cliente tiene la posibi-





de adquisición. La tarjeta tiene dos ADC, pero sólo utilizaremos uno de ellos, específicamente el que está identificado como U2 en la tarjeta, y que tiene el correspondiente CS en el pin 2.

El canal utilizado es el AIN0. Para poder tener entonces la adquisición de este canal debemos ajustar los registros de configuración del ADC. Para ello nos referimos a la hoja de datos. El primer registro(0x00) configura la entrada del multiplexor, la ganancia y el desvío de la PGA. Deberíamos adquirir la señal del canal AIN0 respecto a tierra, el valor para esto es 1000. La ganancia se puede quedar en 1, no necesitaría amplificación en este caso, el valor es 000. Para reducir el consumo de potencia se puede deshabilitar la PGA con el valor de 1 en el campo del desvío. El valor completo que se debe enviar entonces al ADC es 10000001 en binario, o 0x81 en hexadecimal, como lo recibe la función.

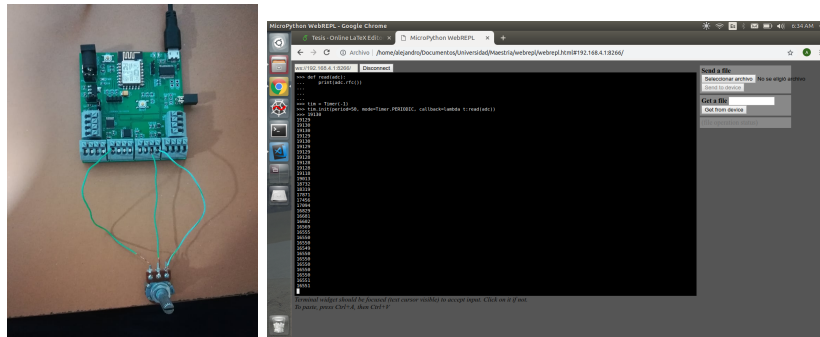
Lo siguiente es configurar la velocidad de muestreo, el modo de conversión y operación. Todo esto se configura mediante el segundo registro(0x01). Al ser un ejemplo donde se busca imprimir en pantalla el valor del potenciómetro, no necesita una velocidad alta de conversión, perfectamente podría operar a 20 muestras por segundo. Para ello debemos usar el valor 000 en data rate, y configurar el modo de operación en normal 00. De igual manera se debe poner en modo de conversión continua con un valor de 1 en el bit correspondiente. El sensor de temperatura y de corriente pueden estar desactivados mediante un 00 en los últimos dos bits. El valor que debemos enviar a este registro es 00000100 en binario, o 0x04 en hexadecimal.

Quedaría faltando ajustar el voltaje de referencia, esto se hace mediante el registro tres(0x02). Los primeros dos bits configuran este voltaje de referencia. Podríamos usar el voltaje de alimentación analógica como referencia. Para ello debemos escribir el valor 11 en el campo de referencia. El filtro notch, el switch y la corriente de salida no son usadas en este ejemplo. Por lo tanto el resto de bits en este registro quedarían en 0. El valor completo que debemos enviar a este registro es 11000000 en binario, o 0xc0 en hexadecimal.

El cuarto y último registro(0x03) que controla las salidas de corriente no nos interesa modificarlo. Por lo tanto se puede quedar tal cual lo deja el reset, o como verificación se puede escribir el valor 00000000 en binario, o 0x00 en hexadecimal. El flujo completo sería entonces:

- Inicializar el objeto del ADC.
- Realizar un reinicio mediante el comando Reset.
- Configurar los cuatro registros con los valores especificados anteriormente.
- Iniciar la adquisición de datos mediante el comando Start.
- Iniciar un ciclo con petición de lectura que se ejecute cada 50ms.
- Imprimir en pantalla cada resultado de conversión.

La manera más eficiente de implementar el ciclo es mediante un **contador**, que ejecute una interrupción a los 50ms programados, y que haga el llamado a una función que lea el valor y lo imprima en pantalla. El código final se encuentra en la sección de anexos. A continuación se expone una captura de pantalla del posible resultado.



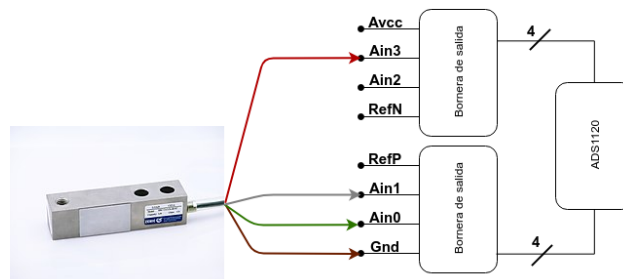
(a) Conexión física del potenciómetro.

(b) Lecturas mediante el WebREPL.

**Figura 4-6.:** Resultados: conexión física del potenciómetro y el resultado de la lectura mediante el WebREPL.

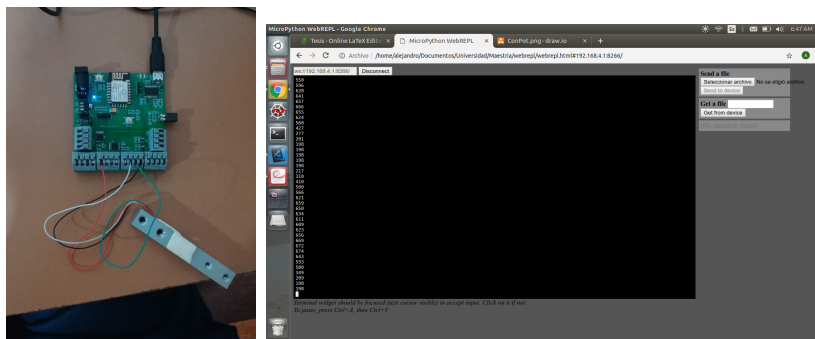
### 4.3.2. ADC con célula de carga

El segundo caso a analizar es el de una señal diferencial de entrada. Para ello haremos uso de una celda de carga. La celda de carga tiene galgas extensiométricas en configuración de puente de *wheatstone*. Los cables comunes que tiene de salida una de estas celdas son los de: Alimentación - Salida positiva - Salida negativa y tierra. Las convenciones de colores predefinidas son: Rojo - Verde - Blanco - Negro respectivamente. De nuevo, y en referencia al ejemplo anterior debemos configurar los registros de manera adecuada. Los parámetros a modificar serían la entrada del multiplexor y la ganancia, puesto que es una señal muy pequeña. El modo de operación, la tasa de conversión, el voltaje de referencia y los demás parámetros del dispositivo se pueden dejar tal cuál el ejemplo 1.



**Figura 4-7.:** Diagrama de conexión de la celda de carga a la tarjeta.

Entonces, el registro a modificar es el primero (0x00), la conexión se realizaría utilizando los canales *Ain0* y *Ain1*. Según la hoja de datos, esta configuración corresponde a 0000 en los bits del multiplexor. La ganancia se debe poner al máximo, para visualizar cambios significativos. Este valor corresponde a una ganancia de 128 voltios y una configuración de 111. Por último debemos habilitar la PGA, esto sería poniendo un 0 en el último bit. La cadena en binario completo sería 00001110 en binario, o 0x0e en hexadecimal. El flujo sería exactamente el mismo del ejemplo anterior, la única diferencia iría en la configuración del primer registro. A continuación se expone una captura de pantalla del posible resultado.



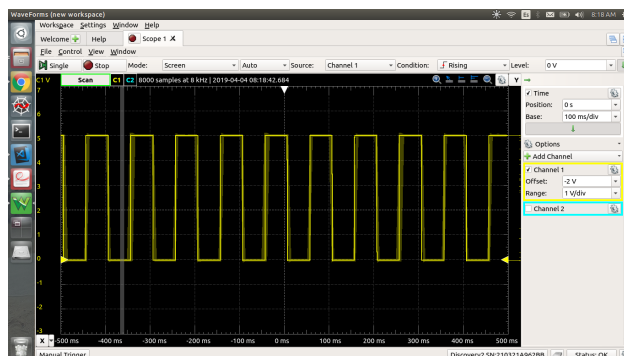
(a) Conexión física de la celda de carga.

(b) Lecturas mediante el WebREPL.

**Figura 4-8.:** Resultados: conexión física de la celda de carga y el resultado de la lectura mediante el WebREPL.

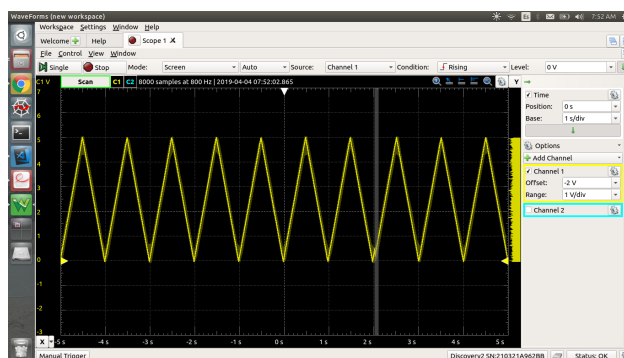
### 4.3.3. DAC

La implementación con el DAC será mediante el dispositivo DAC7311. Un elemento más simple de trabajar, pero que puede tener aplicaciones interesantes. El primer caso es la generación del tren de pulsos a una frecuencia determinada. La señal utilizará el rango completo del convertor, osea, de 0 a 5v. Ya que el DAC solo requiere el envío del valor deseado, sin interacción de registros o comandos, es mucho más sencillo el flujo necesario. El convertor es de 12 bits, osea 4096 divisiones. El tren de pulsos se genera variando la salida del convertor entre 0 y 4095 a una frecuencia determinada. Un ciclo completo sería llevar la señal de 0 a 5v, y después bajarla de nuevo a 0v, manteniendo un periodo fijo entre cambios. La implementación es mediante un contador, que tiene el doble de la frecuencia deseada, y cada que se ejecuta realiza el cambio del valor en la salida. La frecuencia deseada es de 10hz, por lo tanto el contador oscila a una frecuencia de 20hz. Para capturar la salida se utiliza el dispositivo *Analog Discovery 2*, de la empresa digilent. Es un osciloscopio, analizador lógico y generador de señales portable, y alimentado por USB. El código completo se encuentra en los anexos, pero a continuación se adjunta el resultado de la operación:



**Figura 4-9.:** Tren de pulsos a una frecuencia de 10hz, generada con la tarjeta de adquisición.

El segundo caso es la generación de señal triangular de una frecuencia determinada. Al tener solo una salida positiva entre 0 - 5v se realizará un acercamiento de amplitud fija que utilice todo el rango de salida. Para ello debe existir un offset en la señal de 2.5V y una amplitud también de 2.5V. Un ciclo de la señal triangular corresponde a recorrer dos veces el rango completo, osea 8192 divisiones. La frecuencia recomendad de cambio máxima para el DAC en la tarjeta es de 1000 divisiones por segundo, o 1khz. Vamos a representar una señal de 1hz de frecuencia. La manera más fácil de hacerlo es configurar un contador a una frecuencia de 819, y cada que se ejecute, aumentar o disminuir el valor del DAC en 10 según sea necesario. Se utiliza el mismo dispositivo que en el ejemplo anterior para capturar la señal. A continuación se adjunta el resultado que presenta:



**Figura 4-10.:** Señal triangular a una frecuencia de 10hz, generada con la tarjeta de adquisición.

En cada figura, la escala de tiempo se encuentra en la parte inferior. Se logra determinar que cada una cumple con la frecuencia propuesta de 10hz y 1hz respectivamente.

## 5. Conclusiones

Existen sistemas para adquisición de datos de propósito general o para fenómenos particulares. En este trabajo se abundó en sistemas de propósito general. Se plantearon tres objetivos fundamentales: Llegar a un sistema de bajo costo, amigable con el usuario y escalable. Todos se cumplieron con el desarrollo de la tarjeta. Un precio asequible de 20 dólares y una lista de materiales de apenas 27 componentes lograron resolver la tarea. Un firmware de alto nivel y open source lograron una versatilidad y una facilidad de manejo que se ven reflejadas directamente en un tiempo corto de prototipado y desarrollo. Por último la capacidad de conexión inalámbrica WiFi permite que no sólo la tarjeta, sino que la arquitectura de solución sea escalable, y con esto se genera la posibilidad de resolver tareas más complejas.

Siendo más detallados, el trabajo y la investigación da a relucir un fenómeno ya conocido: Cada vez los fabricantes de componentes electrónicos realizan esfuerzos por generar dispositivos más pequeños y con mayores funcionalidades integradas. Esto se ve reflejado una menor lista de materiales y PCB más pequeñas para los diseñadores electrónicos. El precio también se ve afectado, logrando eficiencia y accesibilidad a estas nuevas tecnologías. Otro tema relevante es el de conectividad. Términos como el de industria 4.0 e IOT han hecho que la mayoría de fabricantes opte por incluir en su línea de productos soluciones inalámbricas. Esto también le ha dado posibilidad a empresas alternativas como la misma Espressif de entrar a competir en el mercado. Aunque la confiabilidad y robustez de marcas de renombre es mayor, también es cierto que se presenta un gran impacto en las alternativas.

A nivel de Firmware, también vemos que diferentes exponentes llegan con sus propuestas a revolucionar el mercado. El primero fue Arduino, hace un par de años, permitiéndole a aficionados y personas con conocimiento no tan técnico dar sus primeros pasos en la electrónica. De igual manera a algunos profesionales desarrollar soluciones rápidas para problemas más simples o que la arquitectura lo permita. Una promesa que ha venido evolucionando durante estos años, es Micropython. Una reimplementación de python 3 para microcontroladores. Como sus principales ventajas tenemos que tiende a ser un lenguaje de programación muy amigable y clara. Esto igual se ve contrastado con un rendimiento más lento o pesado respecto a otros lenguajes compilados. Independiente de sus ventajas o restricciones, se ha evidenciado con tarjetas como Microbit(que surge gracias a Micropython) que puede llegar a tener altos impactos pedagógicos, y puede servir como puente para el acercamiento de las personas a la programación de sistemas embebidos. Su migración a puertos como

el ESP8266 le ha dado la posibilidad a nivel general de incluirse en proyectos de IOT. El performance comercial no está del todo documentado, pero en ambientes de desarrollo como el de este trabajo ha presentado muy buenos resultados, mostrándose muy estable y robusto.

Tal vez uno de los factores que más importantes en este trabajo, por encima de la tarjeta, es la arquitectura planteada. Una arquitectura que brinda la posibilidad de escalar, con modificaciones para tareas particulares. Con la capacidad de adquirir datos del entorno, pero también de interactuar. Gracias a una programación abierta y transparente, el usuario final puede tener corriendo lazos de control o algoritmos de análisis de datos y comunicando resultados y reportes de manera remota. Esta accesibilidad de la información es uno de los pilares de la revolución industrial y tecnológica que estamos viviendo.

Se puede influir en el entorno regional a nivel educativo y comercial de diferentes maneras. Tal vez las más relevantes son:

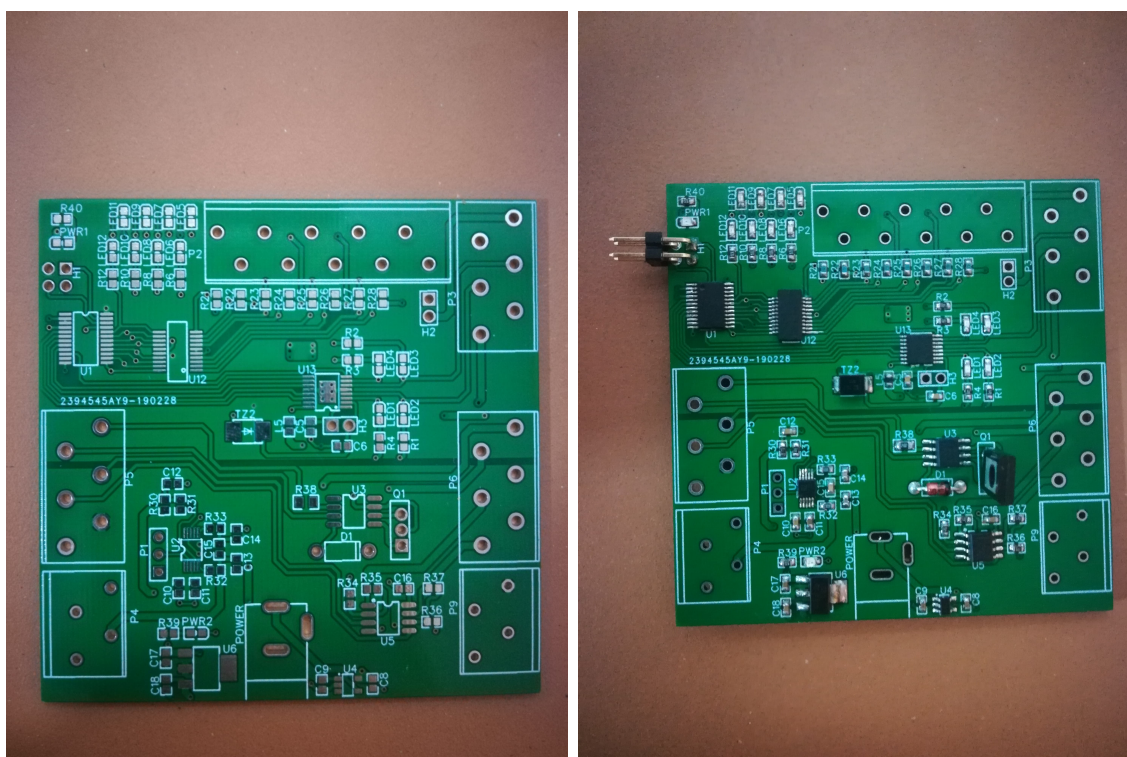
- Automatización a bajo costo: Las empresas en vías de crecimiento (pequeñas y medianas) muchas veces no cuentan con el dinero suficiente para hacer una gran inversión en automatización y estandarización de procesos. En esta región la mayor cantidad de empresas pertenecen a este gremio. Con soluciones como esta se puede brindar alternativas con tecnología de punta que se acomoden a los presupuestos y las necesidades de estas empresas.
- Domótica: Puede entrar perfectamente en aplicaciones inteligentes del hogar. Desde la medición de los consumos energéticos, hasta el accionamiento de mecanismos y luces.
- Emulación de procesos (HIL): Hablando desde la experiencia en la sede Manizales de la Universidad Nacional de Colombia, varias prácticas de automatización se quedan en simulación, o incrustando señales pasivas como potenciómetros. El poder utilizar procesos reales es complejo debido al costo de implementación. Una alternativa viable es mediante emulación, con técnicas como "Hardware In The Loop", donde, se puede montar el comportamiento de una planta en una tarjeta como la desarrollada en este trabajo. El trabajo del estudiante sería controlar el comportamiento de esa planta, mediante por ejemplo, un PLC. Esto podría llegar a ser mucho más cercano a lo que se enfrentarían a nivel profesional trabajando en una empresa. Para esto, es relevante ver [25] [36]

Finalmente podríamos decir que Colombia, a nivel económico es un consumidor de tecnología, y está lejos de ser productor. Sumado a esto, agregamos el hecho que el desarrollo de hardware es muy bajo. Gracias a la accesibilidad de la información en Internet, podemos estar enterados de las últimas tendencias y cambios tecnológicos. Es nuestro deber entonces promover el desarrollo local de soluciones electrónicas, aprovechando ese gran potencial humano e intelectual que se genera en esta tierra. Esto a fin de cuentas, podría potenciar el

desarrollo industrial y de las tecnologías de la información y las comunicaciones, generando mejores condiciones laborales y de vida. Algunas referencias importantes en este tema son [38] [4].

## Trabajo futuro

La investigación y el desarrollo deja abierto el camino tanto para mejoras como para modificaciones. Una de las principales mejoras que se podría realizar está directamente sobre la etapa de acondicionamiento. Esto con el fin de obtener señales y niveles industriales tanto a la entrada como a la salida. En otras palabras, manejar estándares como el de 0-10v o el de 4-20mA. Además, por medio del expansor I2C, se pueden tener tanto entradas como salidas Digitales, brindándole aún más flexibilidad al sistema. En particular, estas dos mejoras ya se están trabajando con una tarjeta de expansión. Esta permite 8 canales de entrada digitales a 24V, y cuatro canales de salida en función de switch, con un rango de voltaje de los 5 - 60V.



(a) PCB.

(b) Tarjeta parcialmente soldada.

**Figura 5-1.:** Resultados: PCB sin componentes y tarjeta parcialmente ensamblada del módulo de expansión digital y acondicionamiento analógico.

Al momento de presentar este documento ya se cuenta con la tarjeta de expansión pero no se ha terminado de realizar las pruebas y ajustes necesarios. Por eso mismo no se incluye en el

documento. De terminar sus pruebas con éxito se podría combinar, en una sola tarjeta con más beneficios como aislamiento digital y filtrado anti-aliasing, brindándole mayor robustez al sistema. Se podría pensar también en modificaciones de la arquitectura que lo hagan apto para diferentes tipos específicos de adquisición.

También sería conveniente realizar las pruebas de rigor en los tres contextos fundamentales presentados en las conclusiones: Automatización de bajo costo, domótica y emulación de procesos mediante HIL. Esto lleva directamente a acabar de madurar la idea, el proyecto, y llevarlo como producto final al contexto.

Por último, se deja en consideración la inclusión de la solución en plataformas existentes de IOT. Esto indica trabajo a nivel de controladores y compatibilidad, pero permite integrar la tarjeta a soluciones ya existentes y probadas. Puede acelerar drásticamente los tiempos de producción final, y sobre todo, puede ampliar drásticamente el panorama de esas plataformas haciendo una combinación versátil.



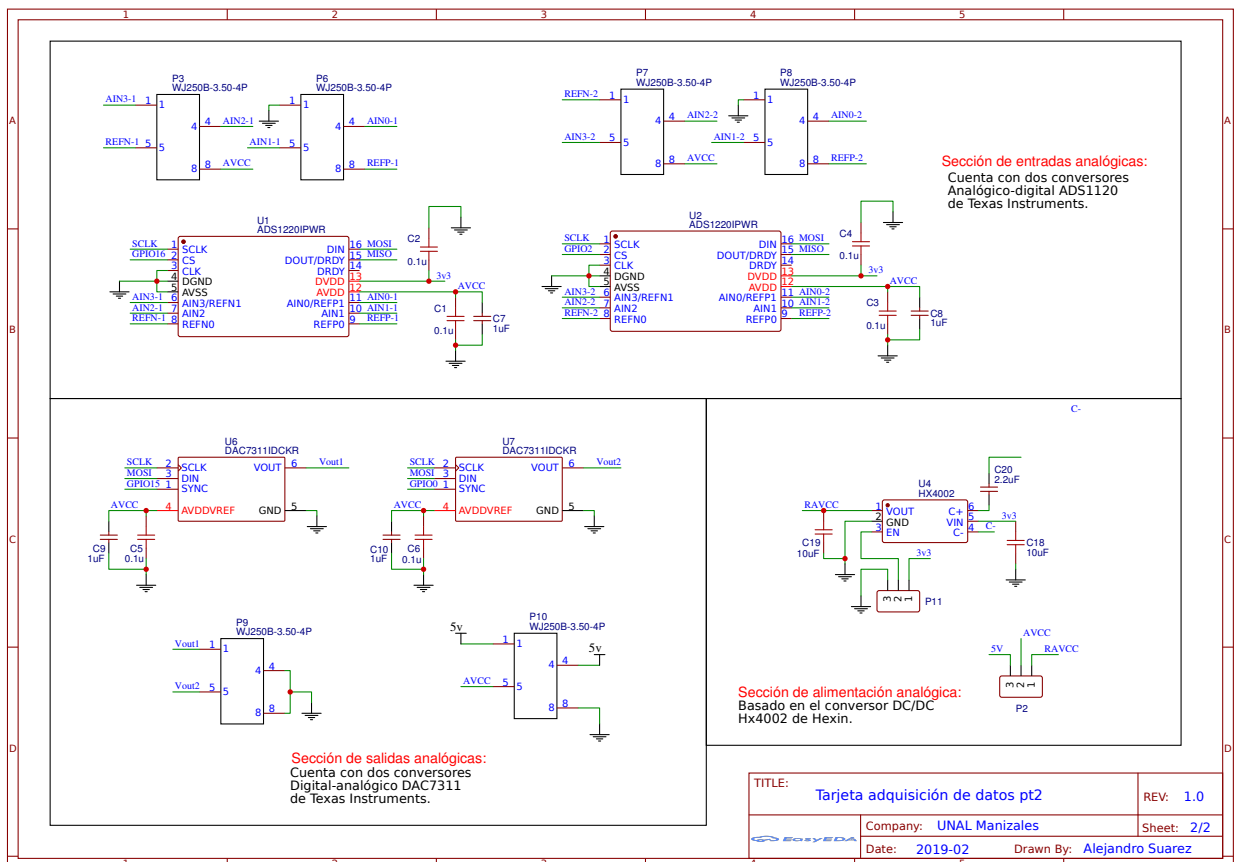
# A. Anexo: Lista de materiales

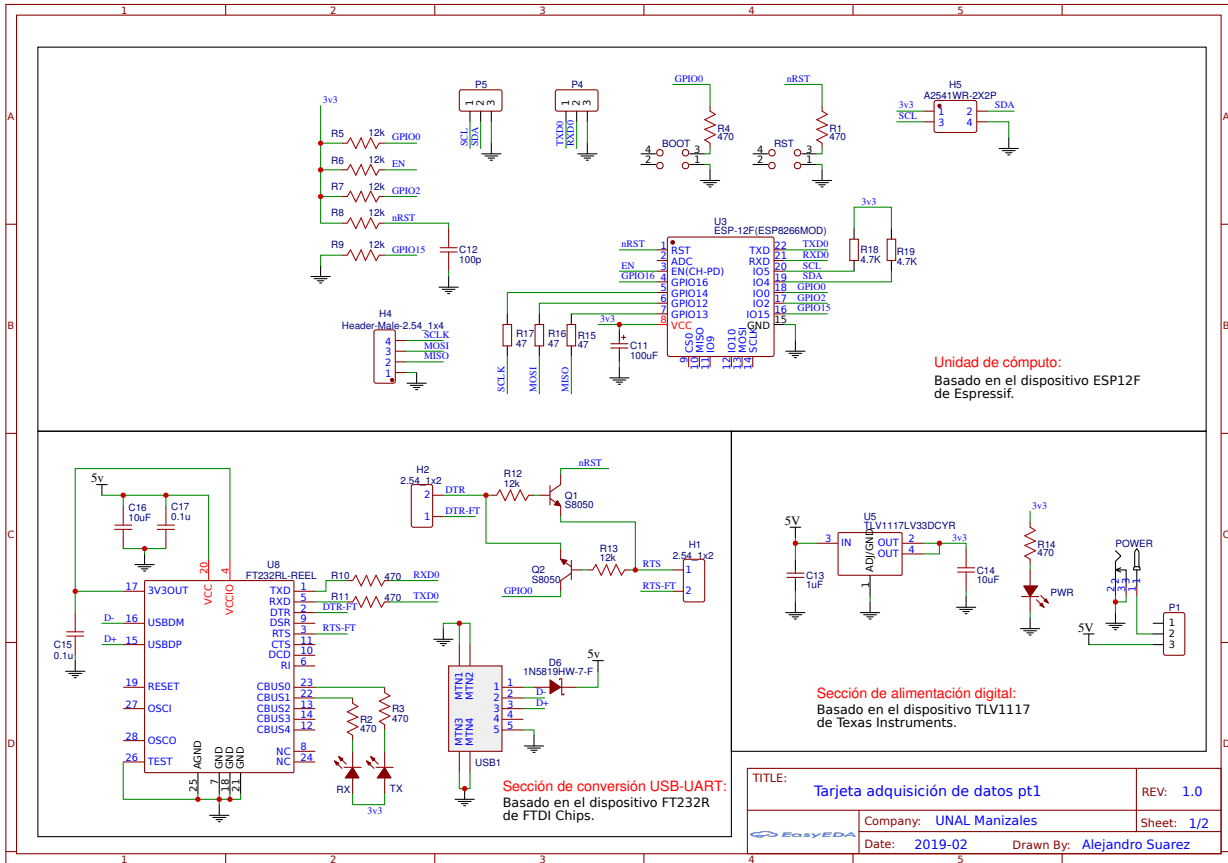
En este apéndice se presenta la lista de materiales del sistema, la descripción que tienen de proveedor (LCSC), y el costo de cada uno.

Referencia en LCSC	Fabricante	Número del fabricante	Cantidad	Precio Unitario(USD)	Precio Línea
C82891	Ai-Thinker	ESP-12F(ESP8266MOD)	1	2,4452	2,4452
C105433	CJ	S8050	2	0,0159	0,0318
C103249	RALEC	RFT031202FTP	7	0,0014	0,0098
C82544	DIODES	1N5819HW-7-F	1	0,0323	0,0323
C101061	LIZ	CR0603FA4700G	7	0,0017	0,0119
C1590	SAMSUNG	CL10B104KA8NNNC	8	0,006	0,048
C1691	SAMSUNG	CL10A106MQ8NNNC	4	0,0091	0,0364
C7195	AVX	TAJB107M006RNJ	1	0,1218	0,1218
C1592	SAMSUNG	CL10A105KO8NNNC	5	0,0071	0,0355
C223821	LCSC	K2-1187SQ-D4SW-06	2	0,0657	0,1314
C16214	BOOMELE	DC-005 5.5-2.0MM	1	0,0388	0,0388
C49257	BOOMELE	Header2.54mm 1*3P	5	0,0105	0,0525
C1635	FH	0603CG101J500NT	1	0,0044	0,0044
C15578	TI	TLV1117LV33DCYR	1	0,175	0,175
C8690	FTDI	FT232RL-REEL	1	3,2407	3,2407
C86471	TE Connectivity	826629-2	2	0,1941	0,3882
C138541	BrLed	BL-HB336G-TRB	3	0,0213	0,0639
C10418	ValuePro	micro USBFemale	1	0,0711	0,0711
C124378	Ckmtw	210S-1*4P L=11.6MMGold-plated black	1	0,0254	0,0254
C114623	YAGEO	RC0603FR-0747RL	3	0,0012	0,0036
C99782	YAGEO	RC0603FR-074K7L	2	0,0014	0,0028
C239333	Changjiang Connectors	A2541WR-2x2P	1	0,0735	0,0735
C48263	TI	ADS1220IPWR	2	3,9884	7,9768
C128601	TI	DAC7311IDCKR	2	1,5623	3,1246
C172354	HEXIN	HX4002	1	0,0744	0,0744
C23630	SAMSUNG	CL10A225KO8NNNC	1	0,0102	0,0102
C8455	ReliaPro	WJ250B-3.50-4P	6	0,1511	0,9066
				<b>Precio total(USD):</b>	<b>19,1366</b>

# B. Anexo: Esquemáticos

En este apéndice se presentan los esquemáticos de la tarjeta.





## C. Anexo: Funciones y códigos

A continuación se adjuntan los códigos generados y utilizados durante el desarrollo de este documento. El primer archivo corresponde a "DAQ.py":

```
from machine import SPI, Pin
import time
```

```
""" Archivo con las clases que manejan la tarjeta de adquisicion.
Contiene las clases para manejar el ADC y el DAC, importa las
librerias necesarias e inicializa las variable. la manera de uso
es mediante: import DAQ."""
```

```
spiADC = SPI(baudrate=10000000, miso=Pin(13), mosi=Pin(12), sck=Pin(14), po
spiDAC = SPI(baudrate=10000000, miso=Pin(13), mosi=Pin(12), sck=Pin(14), po
```

```
class adc:
```

```
    """ Clase que maneja el ADC.
```

```
    Requiere un entero que representa el pin de Chip Select, y la
    referencia del conversor, si es el ADS1120 la referencia seria un
    string con el valor 1120."""
```

```
    def __init__(self, PinCS, ref="1120", spi=spiADC):
        self.cs = Pin(PinCS, Pin.OUT)
        self.cs.on()
        self.reset = 0x06
        self.start = 0x08
        self.wreg = 0x40
        self.rreg = 0x20
        self.read = 0x10
        self.ref = ref
        self.spi = spiADC
        self.cmdRead = bytearray(3)
        self.cmdRead[0] = 0x10
        self.cmdRead[1] = 0x00
        self.cmdRead[2] = 0x00
```

```
def SPI_command(self, data_in):
    """Comando por SPI.
    Requiere un valor en hexadecimal o decimal
    Envia dicho valor por el SPI hacia el ADC"""
    self.cs.off()
    data = bytearray(1)
    data[0] = data_in
    self.spi.write(data)
    self.cs.on()

def Reset(self):
    """ Reinicio del ADC.
    Comando para reiniciar el ADC."""
    self.SPI_command(self.reset)

def Start(self):
    """ Inicio de conversion.
    Este comando el indica al ADC que debe iniciar la conversion."""
    self.SPI_command(self.start)

def readRegister(self, address):
    """ Leer registro.
    Comando para solicitar un registro especifico al ADC
    Requiere el valor del registro en hexadecimal."""
    self.cs.off()
    data = bytearray(1)
    data[0] = self.rreg | (address << 2)
    self.spi.write(data)
    answer = self.spi.read(1)
    self.cs.on()
    return answer

def writeRegister(self, address, value):
    """ Escribir registro.
    Comando para escribir un registro especifico en el ADC
    Requiere el valor del registro en hexadecimal y el
    valor del valor, tambien en hexadecimal."""
    self.cs.off()
    data = bytearray(1)
```

---

```
        data[0] = self.wreg|(address<<2)
        self.spi.write(data)
        data[0] = value
        self.spi.write(data)
        self.cs.on()

def readData(self):
    """Comando para leer un dato del ADC.
    Con este comando se lee un dato, la respuesta es un
    bytearray. Se requiere la conversion a decimal."""
    self.cs.off()
    data = bytearray(1)
    data[0] = self.read
    self.spi.write(data)
    if(self.ref=="1120"):
        answer = self.spi.read(2)
    else:
        answer = self.spi.read(3)
    self.cs.on()
    return answer

def rShot(self):
    """Solicitar dato en modo Single Shot.
    Funcion para solicitar al ADC que realice una lectura, leer
    la respuesta y convertirla a un valor decimal."""
    self.Start()
    data = self.readData()
    if(self.ref=="1120"):
        number=(data[0]<<8)|data[1]
    else:
        number = (((data[0]<<8)|data[1])<<8)|data[2]
    return number

def rCont(self):
    """Solicitar dato en modo Conversion Continua.
    Funcion para leer el registro de salida del ADC
    y convertirlo a un valor decimal."""
    data=self.readData()
    if(self.ref=="1120"):
        number=(data[0]<<8)|data[1]
```

```

else:
    number = (((data[0]<<8)|data[1])<<8)|data[2]
return number

```

```

@micropython.native
def rfc(self):
    """Funcion para lectura rapida.
Funcion para leer datos en modo de Conversion Continua,
optimizado para tener un mejor rendimiento."""
    csPin = self.cs
    ans = bytearray(3)
    csPin.off()
    self.spi.write_readinto(self.cmdRead, ans)
    csPin.on()
    return (ans[1]<<8)|ans[2]

```

```

class dac:
    """Clase que maneja el DAC.
Requiere un entero que representa el pin de Chip Select, y la
referencia del conversor, si es el DAC7311 la referencia seria un
string con el valor 7311."""
    def __init__(self, PinCS, ref="7311", spi=spiDAC):
        self.cs = Pin(PinCS, Pin.OUT)
        self.cs.on()
        self.spi = spi

    @micropython.native
    def writeValue(self, value):
        """Escribir valor.
Comando para solicitarle al DAC que genere un valor, requiere un
entero entre 0 y 4095, de otra manera no funcionara"""
        if((value>=0)&&(value<4095)):
            self.cs.off()
            data = bytearray(2)
            data[0] = (value>>6)
            data[1] = (value&0x03f)<<2
            self.spi.write(data)
            self.cs.on()

```

---

El segundo corresponde a **"dacSquare.py"**:

```
from machine import Timer
import time
import DAQ

""" Archivo para demostrar como generar una senal cuadrada de rango
completo mediante el DAC. La frecuencia de la senal es la mitad de
la frecuencia del timer."""

dac = DAQ.dac(15)
opt = [False]

def write(dac):
    v = opt[0]
    if(v):
        dac.writeValue(4095)
    else:
        dac.writeValue(0)
    opt[0] = not(v)

tim = Timer(-1)
tim.init(freq=100, mode=Timer.PERIODIC, callback=lambda t:write(dac))
```

El tercero corresponde a **"dacTriang.py"**:

```
from machine import Timer
import time
import DAQ

""" Archivo para demostrar como generar una senal cuadrada de rango
completo mediante el DAC. La frecuencia de la senal depende del timer
y del valor con el cual se incrementa la variable v."""

dac = DAQ.dac(15)
opt = [2048,1]

def write(dac):
    v = opt[0]
    s = opt[1]
    if(v>4085):
        s = 0
```



```

elif (v<10):
    s = 1
if (s):
    v+=10
else:
    v-=10
dac.writeValue(v)
opt[0] = v
opt[1] = s

```

```

tim = Timer(-1)
tim.init(freq=819, mode=Timer.PERIODIC, callback=lambda t:write(dac))

```

El cuarto y último corresponde a **"lecturaADC.py"**:

```

from machine import Timer
import time
import DAQ

```

```

""" Archivo para demostrar como se adquiere una senal del ADC
funciona mediante un timer, y guarda el resultado en un archivo
de texto llamado res.txt."""

```

```

adc = DAQ.adc(2,"1120")
adc.Reset()
adc.writeRegister(0x00,0x0e)
adc.writeRegister(0x01,0x04)
adc.writeRegister(0x02,0x00)
adc.writeRegister(0x03,0x00)
adc.Start()
data = []

```

```

def read(adc):
    """ Funcion que se ejecuta como llamado del timer.
    Agrega a un arreglo vacio llamado data el valor de cada lectura """
    data.append(adc.rfc())

```

```

tim = Timer(-1)

```

```

t1 = time.ticks_ms()
tim.init(freq=90, mode=Timer.PERIODIC, callback=lambda t:read(adc))

```

```
time.sleep_ms(1000)
tim.deinit()
t2 = time.ticks_ms()

print(" Finished...")
print(" Time elapsed: "+str(t2-t1))

f = open('res.txt', 'w')
for i in range(len(data)):
    f.write(str(data[i]))
    f.write("\n")
f.close()

print(" Data saved...")
```

# Bibliografía

- [1] ADAMS, Leon ; MARKETING, Strategic: Choosing the right architecture for real-time signal processing designs. En: *Texas Instruments, Document Number SPRA879* (2002)
- [2] ALLEY, Peter J.: *Introductory Microcontroller Programming*. (2011)
- [3] BAO, Shuang ; YAN, Hairong ; CHI, Qingping ; PANG, Zhibo ; SUN, Yuying: FPGA-based reconfigurable data acquisition system for industrial sensors. En: *IEEE Transactions on Industrial Informatics* 13 (2017), Nr. 4, p. 1503–1512
- [4] BAREÑO, Carlos Iván C.: Transferencia tecnológica y de conocimientos en el diseño de sistemas embebidos. En: *Tesis Doctoral. Universidad Nacional de Colombia-Sede Bogotá* (2011)
- [5] CHAR, Kavita: Internet of things system design with integrated wireless MCUs. En: *Silicon Labs, ARM, Tech. Rep* (2015)
- [6] MEASUREMENT COMPUTING, MC. *Low Cost DAQ Devices*. <https://www.mccdaq.com/data-acquisition/low-cost-daq>. 2019
- [7] OPEN DAQ. *openDAQ*. <https://www.open-daq.com/>. 2019
- [8] DOCUMENTATION, Micropython. *Maximising MicroPython Speed*. [http://docs.micropython.org/en/v1.9.3/esp8266/reference/speed\\_python.html](http://docs.micropython.org/en/v1.9.3/esp8266/reference/speed_python.html). 2019
- [9] DUNN, William C.: *Introduction to instrumentation, sensors, and process control*. Artech House Publishers, 2017
- [10] Espressif Systems: *ESP8266EX*. 11 2017
- [11] ESTEVES, Francisco José Antunes M.: *Development and Implementation of a Data Acquisition System for Teaching Purposes*, Técnico Lisboa, Tesis de Grado, 2015
- [12] FARALDOS, Marisol ; GOBERNA, Consuelo: *Técnicas de análisis y caracterización de materiales*. Editorial CSIC Consejo Superior de Investigaciones Científicas, 2011
- [13] Ftdi chips: *FT232R USB UART IC*. 7 2005
- [14] FURBER, Stephen B.: *ARM system-on-chip architecture*. pearson Education, 2000

- [15] GARCÍA, Miguel Ángel P.: *Instrumentación electrónica*. Ediciones Paraninfo, SA, 2014
- [16] GATTU, Anil Kumar R. ; KUMAR, P S. ; SHRIRAMSHETTY, Madhu: Ethernet based multifunction DAQ using pic micro controller. En: *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECOT)* IEEE, 2017, p. 1–7
- [17] HANDBOOK, Data A.: A Reference for DAQ and Analog & Digital Signal Conditioning. En: *Measurement Computing Corporation (2004-2012)* (2012)
- [18] Hexin: *Hx4002 Low noise, regulated charge pump DC/DC converter*
- [19] IMREK, J ; NOVAK, D ; HEGYESI, Gy ; KALINKA, G ; MOLNAR, J ; VEGH, J ; BALKAY, L ; EMRI, M ; MOLNAR, G ; TRON, L [u. a.]: Development of an FPGA-based data acquisition module for small animal PET. En: *IEEE transactions on nuclear science* 53 (2006), Nr. 5, p. 2698–2703
- [20] JUSOH, W. N. S. E. W. ; HANAFIAH, M.a. M. ; GHANI, M.r. A. ; RAMAN, S.h.: Remote terminal unit (RTU) hardware design and implementation efficient in different application. En: *2013 IEEE 7th International Power Engineering and Optimization Conference (PEOCO)* (2013)
- [21] KAPTE, Jyoti J. ; UBALE, Ganesh A.: Design of on-line interactive data acquisition and control system for embedded real time applications. En: *2018 2nd International Conference on Inventive Systems and Control (ICISC)* (2018)
- [22] KIRIANAKI, Nikolay V. ; YURISH, Sergey Y. ; SHPAK, Nestor O. ; DEYNEGA, Vadim P.: *Data acquisition and signal processing for smart sensors*. Wiley New York, 2002
- [23] KITCHIN, Charles ; COUNTS, Lew: *A designer's guide to instrumentation amplifiers*. Analog Devices, 2004
- [24] LI, Shancang ; DA XU, Li ; WANG, Xinheng: Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. En: *IEEE Transactions on Industrial Informatics* 9 (2013), Nr. 4, p. 2177–2186
- [25] LOGREIRA, Victor Zabala C. ; MUSA, Ronald Z.: Análisis y actualización del programa de la asignatura Automatización Industrial en la formación profesional de ingenieros electrónicos. En: *Revista Educación en Ingeniería* (2016), p. 39–44
- [26] MANIVANNAN, M ; KUMARESAN, N: Design of on-line Interactive Data Acquisition and Control System for embedded real time applications. En: *2011 International Conference on Emerging Trends in Electrical and Computer Technology* (2011)

- [27] MARTÍN, FJ F. ; LLOPIS, M V. ; RODRÍGUEZ, JC C. ; GONZÁLEZ, JR B. ; BLANCO, J M.: Low-cost open-source multifunction data acquisition system for accurate measurements. En: *Measurement* 55 (2014), p. 265–271
- [28] MIGUEL, Mercedes G. ; BOLADO, Elena M.: *Instrumentación electrónica: transductores y acondicionadores de señal*. Vol. 15. Ed. Universidad de Cantabria, 2010
- [29] MIN, Peng ; SHI, Chang ; CHEN, Zhi M.: Design of Industrial Control Data Acquisition System Based on Embedded System. En: *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)* IEEE, 2018, p. 279–281
- [30] NIU, Ling: Design of Remote data acquisition system based on Internet of Things. (2014)
- [31] PALLAS-ARENY, Ramon ; WEBSTER, John G.: *Sensors and signal conditioning*. John Wiley & Sons, 2012
- [32] PARK, John ; MACKAY, Steve: *Data acquisition for instrumentation and control systems*. 2003
- [33] PENG, Daogang ; ZHANG, Hao ; ZHANG, Kai ; LI, Hui ; XIA, Fei: Research and Development of the Remote I/O Data Acquisition System Based on Embedded ARM Platform. En: *2009 International Conference on Electronic Computer Technology* (2009)
- [34] PÉREZ ROQUE, Felipe ; VALDÉS ZALDÍVAR, Enrique ; ARIAS DE FUENTES, Olimpia: Sistema de Adquisición de Datos con comunicación inalámbrica. En: *Ingeniería Electrónica, Automática y Comunicaciones* 34 (2013), Nr. 3, p. 63–73
- [35] PRASATH, A. ; SATHEESH, A.: Implementation of real time data acquisition system with ARM. En: *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (2015)
- [36] RAMÍREZ HEREDIA, Ricardo E. ; PALACIOS, Víctor Hugo G.: Desarrollo de un simulador de procesos industriales bajo configuración Hardware-in-the-Loop para la práctica-enseñanza de control lógico y regulatorio mediante un PLC. En: *Tesis Mestría. Universidad Nacional de Colombia*. (2015)
- [37] REAL, Gustavo E. ; JAURÉ, María F. ; VITALI, Amado O.: Data acquisition and industrial control system based on Arduino Due using open-source hardware and software. En: *2018 XIII Technologies Applied to Electronics Teaching Conference (TAEE)* IEEE, 2018, p. 1–7

- 
- [38] RUEDA, Jose Alejandro D.: Metodología integral para el emprendimiento basado en sistemas embebidos digitales en Colombia. En: *Tesis Maestría. Universidad Nacional de Colombia-Sede Bogotá* (2018)
- [39] SHARIFF, Farihah ; RAHIM, Nasrudin A. ; HEW, Wooi P.: Zigbee-based data acquisition system for online monitoring of grid-connected photovoltaic system. En: *Expert Systems with Applications* 42 (2015), Nr. 3, p. 1730–1742
- [40] SU, Hong J.: The Embedded Image Acquisition System Based on the ARM. En: *Journal of Convergence Information Technology* 8 (2013), Nr. 9, p. 845–852
- [41] Texas instruments: *TLV1117 Adjustable and Fixed Low-Dropout Voltage Regulator*. 12 2004
- [42] Texas instruments: *DACx311 2-V to 5.5-V, 80-A, 8-, 10-, and 12-Bit, Low-Power, Single-Channel, Digital-to-Analog Converters in SC70 Package*. 7 2008
- [43] Texas instruments: *ADS1120 4-Channel, 2-kSPS, Low-Power, 16-Bit ADC with Integrated PGA and Reference*. 7 2013
- [44] YAOYUAN, Chen ; JINSONG, Yu ; DIYIN, Tang ; HAO, Liu: Design and implementation of a LXI bus based multi-functional, portable and universal DAQ instrument. En: *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)* IEEE, 2017, p. 2038–2042
- [45] YIU, Joseph: *The Definitive Guide to ARM® Cortex®-M0 and Cortex-M0+ Processors*. Academic Press, 2015