

Caracterización de componentes electrónicos utilizando una placa de audio

Micaela Toscani, Axel Lacapmesure y Guillermo Brinatti

13 de noviembre de 2018

1. Introducción

2. Desarrollo de la aplicación de adquisición

El control de la placa de adquisición fue llevado a cabo a partir de programar una biblioteca propia en lenguaje Python [1]. La misma incluyó funciones orientadas al usuario final destinadas a utilizar el dispositivo para la adquisición de datos y la generación de trenes de pulsos configurables, tanto en los modos de disparo único como de ejecución continua. Para ello empleamos la biblioteca *nidaqmx* propia del fabricante de la placa de adquisición [2], y que provee una API para la configuración del dispositivo y para su comunicación a través de los controladores del mismo.

Las funciones de usuario final se encargaron de la configuración de los canales de entrada analógicos y salidas de contadores, de la adquisición de una muestra finita de largo arbitrario, y de la adquisición continua de datos en simultáneo con la generación de pulsos de anchos variables. En la figura 1 se muestra un diagrama de flujo de una ejecución típica para la adquisición y generación de pulsos en modo continuo. Asimismo la biblioteca incluyó los elementos necesarios para la implementación del lazo de control que se discutirá en la sección 4, a saber, las estructuras que manejan la entrada de datos de sensado, que calculan la señal de error y que generan la señal de hacia el actuador.

El la adquisición de lectura y escritura se basa en un bucle que en cada iteración adquiere una ventana de datos con un número fijo de muestras, las procesa y por último escribe la configuración de los pulsos de salida, esto es, su frecuencia de repetición y su ancho. Para el funcionamiento de este esquema se asume que el tiempo de ejecución de las instrucciones dentro de cada iteración es menor al tiempo que demora la adquisición de las muestras, de manera que no se produzcan sobrecargas en el *buffer* de la placa de adquisición. Bajo el modo de disparo único, las sucesivas ventanas temporales se concatenan en una variable de salida preinicializada, y el bucle finaliza cuando el número de muestras adquiridas alcanza el número total

configurado. Bajo el modo de ejecución continua, las ventanas temporales se sobrescriben, y el bucle finaliza cuando el usuario interrumpe manualmente la ejecución del programa con el comando de teclado. En ambos casos, la finalización del bucle desencadena la etapa de cierre de comunicaciones y, eventualmente, el gráfico y guardado de los datos.

3. Caracterización de la placa de adquisición

4. Control de temperatura

4.1. Implementación de un lazo de control

4.2. Implementación de un lazo de control PID

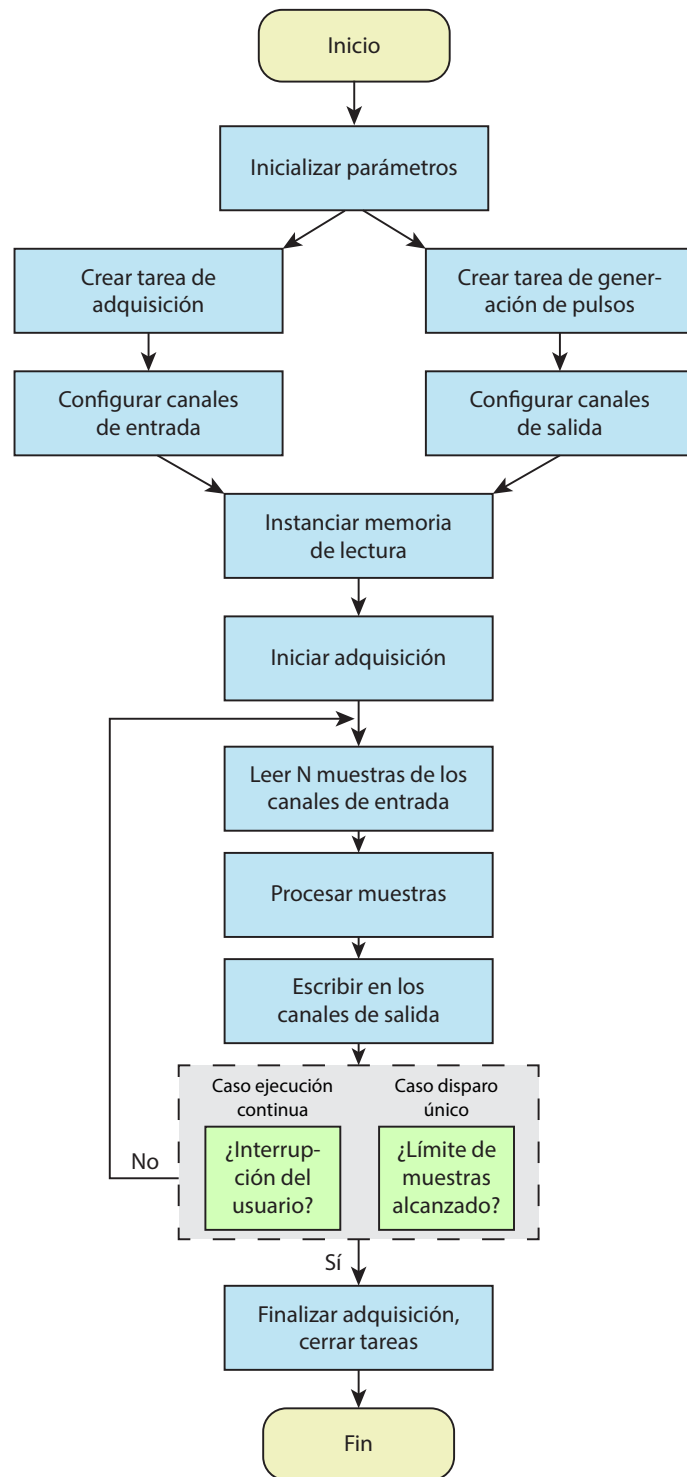


Figura 1: Diagrama de flujo para la aplicación de adquisición y generación de pulsos en modo de disparo único y ejecución continua.

Referencias

- [1] Repositorio en GitHub de la biblioteca propia utilizada en el trabajo, URL: <https://github.com/fotonicaOrg/daq>, actualizado el 7/11/2018. La biblioteca está programada en el archivo *daq.py*.
- [2] Documentación de la biblioteca nidaqmx, URL: <https://nidaqmx-python.readthedocs.io>, accedido el 7/11/2018.