



Introducción al sistema de adquisición de datos iiDAQ

Recurso desarrollado para el proyecto PE101222

“La adquisición de datos como herramienta básica en el modelo mixto de
enseñanza-aprendizaje de Instrumentación y Control”

Apoyado por el programa UNAM - DGAPA - PAPIIME

Licencia



Ciudad Universitaria, CD.MX. 2022

Autores:

Lizbeth González Arrieta

Dr. Roberto Giovanni Ramírez Chavarría

Dra. Flor Lizeth Torres Ortiz

Ing. Areli Reséndiz Quintanar

Créditos

Todas las figuras contenidas en este tutorial son de elaboración propia, tomadas del sistema iiDAQ.

Índice general

1. Un vistazo general a los sistemas de adquisición de datos	5
1.1. Alcances	5
2. Descripción general	7
2.1. Hardware del sistema	7
2.2. Software del sistema	8
2.2.1. Modo gráfico	9
2.2.2. Modo de comandos	9
3. Implementación del sistema de adquisición de datos iiDAQ	11
3.1. Conexión de bloques para la adquisición de datos	11
3.2. Configuración del microcontrolador a partir de la IDE de Arduino	14
3.3. Instalación del Toolbox	16
3.3.1. Requerimientos del sistema	16
3.3.2. Instalación de interfaz gráfica y funciones especiales	16
4. Primeros pasos en la adquisición de datos con iiDAQ	19
4.1. Interfaz gráfica	19
4.2. Modo de comandos aplicados en scripts	21
4.2.1. Plantilla básica para la implementación de scripts	22
5. Prácticas complementarias para el uso del sistema iiDAQ	23
5.1. Práctica 1. Encender un led	26
5.1.1. Interfaz gráfica	26
5.1.2. Modo de comandos	27
5.1.3. Resultados esperados	29
5.2. Práctica 2. Modulación de ancho de pulso con un led	30
5.2.1. Interfaz gráfica	30
5.2.2. Modo de comandos	31
5.2.3. Resultados esperados	33
5.3. Práctica 3. Medición de voltaje en un circuito resistivo	35
5.3.1. Interfaz gráfica	35
5.3.2. Modo de comandos	37
5.3.3. Resultados esperados	38
5.4. Práctica 4. Generación de onda vista en un osciloscopio	42
5.4.1. Interfaz gráfica	42
5.4.2. Modo de comandos	43
5.4.3. Resultados esperados	45
A. Costos de dispositivos en el mercado actual	49
B. Driver para Arduino Nano 3.0	51

C. Consideraciones importantes para el sistema iiDAQ

57

Capítulo 1

Un vistazo general a los sistemas de adquisición de datos

La adquisición de datos (DAQ: *data acquisition*), es actualmente una de las herramientas más útiles e importantes en la instrumentación y control de todo tipo de procesos. Al ser una tendencia tecnológica tan significativa, muchos de los sistemas DAQ están disponibles sólo como herramientas de gran escala, principalmente enfocados a aplicaciones y soluciones industriales. Aunque existen sistemas DAQ comerciales para fines académicos, su principal limitante es el costo del producto y licenciamiento, el cual no es accesible para todas las instituciones de educación superior, ni para los estudiantes. Por esta razón, las actividades prácticas han sido reemplazadas por simulaciones, las cuales, permiten al estudiante tener apenas un mínimo acercamiento tangible al fenómeno real que se desea estudiar, presentando una gran problemática debido a la falta de elementos con los cuales el estudiante pueda experimentar y, al mismo tiempo, el docente pueda emplear para reforzar la enseñanza de los contenidos que inminentemente lo requieren.

Se cree que las propuestas de solución para estos sistemas deben tener como directriz herramientas de hardware y software de fácil acceso, que permitan a los alumnos, a partir de un entorno ya conocido, familiarizarse con el sistema de manera rápida e intuitiva. Para esto, existen múltiples plataformas hardware de acceso abierto, las cuales han cobrado relevancia en los últimos años para el desarrollo de proyecto y prototipos. Sin embargo, dichas plataformas suelen estar limitadas en cuanto a resolución, velocidad de muestreo y capacidad de interactuar con *software* especializado.

Así entonces, presentamos el sistema iiDAQ, un dispositivo de generación y adquisición de datos, accesible y de bajo costo, para ser controlado desde el entorno de programación de MATLAB™, inspirados en la problemática del proceso de enseñanza-aprendizaje de instrumentación y control, así como la adquisición de datos como herramienta fundamental del proceso. El objetivo principal de este tutorial es obtener la capacidad necesaria para reconstruir el sistema e instalar las herramientas necesarias para la correcta implementación del iiDAQ, y así, complementar de manera satisfactoria el aprendizaje de conceptos básicos fundamentales.

1.1. Alcances

Este tutorial está dirigido a profesores y estudiantes de ingeniería a nivel licenciatura, con enfoques en el área de electrónica, mecatrónica, control e instrumentación. El sistema será capaz de obtener

la información requerida por el usuario para la implementación de un gran número de aplicaciones. Dicho esto, se cuenta con un *toolbox*, el cual es un conjunto de herramientas virtuales en el ambiente MATLABTM para operar el sistema iiDAQ. Este estará conformado a partir de una interfaz gráfica, con el objetivo de realizar prácticas de enfoques básicos, y funciones de bajo nivel, para la implementación de prácticas específicas de mayor complejidad.

Capítulo 2

Descripción general

2.1. Hardware del sistema

iiDAQ es un sistema electrónico que permite adquirir y generar señales a partir de un microcontrolador y comunicación serial, mediante una computadora personal. La Figura 2.1 muestra el diagrama de bloques del sistema iiDAQ y sus principales componentes, de los cuales se destaca un módulo convertidor analógico-digital (*ADS1115*), un módulo convertidor digital-analógico (*MPC4725*) y un generador de funciones (*AD9833*), todos controlados a partir de una tarjeta de desarrollo Arduino Nano.

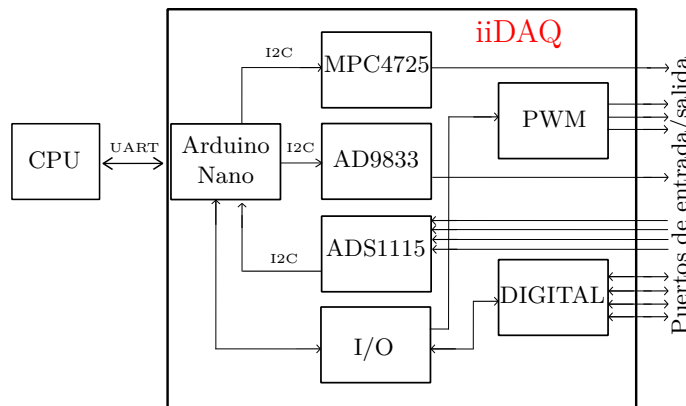







Figura 2.1: Diagrama a bloques del sistema iiDAQ.

Cuenta con ocho puertos de entrada y nueve puertos de salida, distribuidos en el sistema para diferentes funcionalidades (las cuales serán explicadas conforme se avance en la lectura de este tutorial), además de dos pines para suministro de alimentación simétrica. Los protocolos de comunicación utilizados serán I²C, SPI y UART, los cuales permiten obtener y mandar datos entre el hardware y software del sistema a partir del puerto serial disponible en nuestra computadora. En la Tabla 2.1, se resumen las características de los dispositivos¹ que componen el *hardware* del sistema iiDAQ, además de la descripción del funcionamiento de cada uno de los bloques implementados.

¹Puedes consultar una guía de costos en el anexo A de esta guía de usuario

Tabla 2.1: Dispositivos *hardware* del sistema iiDAQ.

Dispositivo	Descripción	Imagen
<i>Arduino Nano 3.0</i>	Microcontrolador base: Atmel ATmega328 de 8 bits @ 16 MHz. Esta tarjeta de desarrollo tiene disponibles 14 terminales digitales (seis proveen señales PWM) y cinco entradas analógicas. La tarjeta provee comunicación serial con una computadora personal mediante un conector USB Mini-B.	
<i>ADS1115</i>	Convertidor analógico-digital (ADC) Resolución: 16-bit con ganancia programable 2/3, 1, 2, 4, 8, 16. Permite obtener mediciones provenientes de dos canales diferenciales o cuatro canales single-ended, logrando obtener datos digitales de la señal analizada.	
<i>MCP4725</i>	Convertidor digital-analógico (DAC). Resolución: 12-bit (1.2 mV @ 5 V). Permite obtener una señal analógica a partir de un procesador digital. Este se controla a partir de I ² C.	
<i>AD9833</i>	Generador de señales sinusoidal, triangular y cuadrada. Frecuencia: 0.1 Hz a 10 MHz.	
<i>DD1718PA</i>	Convertidor ADC DAC para alimentación simétrica de circuitos. Voltaje: $\pm 5V$ para alimentación de circuitos de acondicionamiento de señales.	
<i>PWM</i>	Tres terminales para generar señales PWM (<i>pulse width modulation</i>), donde el control principal será a partir del ciclo de trabajo de la señal.	N/A
<i>Digital I/O</i>	Cuatro terminales digitales configurables como entrada o salida con valores lógicos de 0 y 1.	N/A

2.2. Software del sistema

Como elemento indispensable para la puesta en funcionamiento del sistema iiDAQ, se tiene, a la par, un conjunto de herramientas de *software* en el ambiente MATLABTM, denominado *toolbox*; estas permiten realizar toda la comunicación con el *hardware* del sistema, y así, obtener los datos requeridos dependiendo del caso de estudio. Así, cuenta con dos modalidades de funcionamiento disponible para interactuar: i) modo gráfico y ii) modo de comandos.

2.2.1. Modo gráfico

El principal objetivo del modo gráfico de operación es tener un primer acercamiento con el sistema de adquisición de datos iiDAQ. Así, el usuario puede explorar el funcionamiento del sistema y hacer uso de los módulos para aplicaciones de propósito general. Estas características, hacen del modo gráfico, un ambiente intuitivo y de gran apoyo para un aprendizaje rápido de conceptos relacionados con la adquisición de datos y generación de señales.

2.2.2. Modo de comandos

En esta modalidad, se programan funciones dedicadas a interactuar de forma particular con los módulos del *hardware* de iiDAQ. El objetivo principal de este conjunto de funciones es ser utilizadas en aplicaciones más específicas y/o avanzadas. Particularmente, la intención es desarrollar aplicaciones con algoritmos para la adquisición de datos, control y sistemas de medición.

Capítulo 3

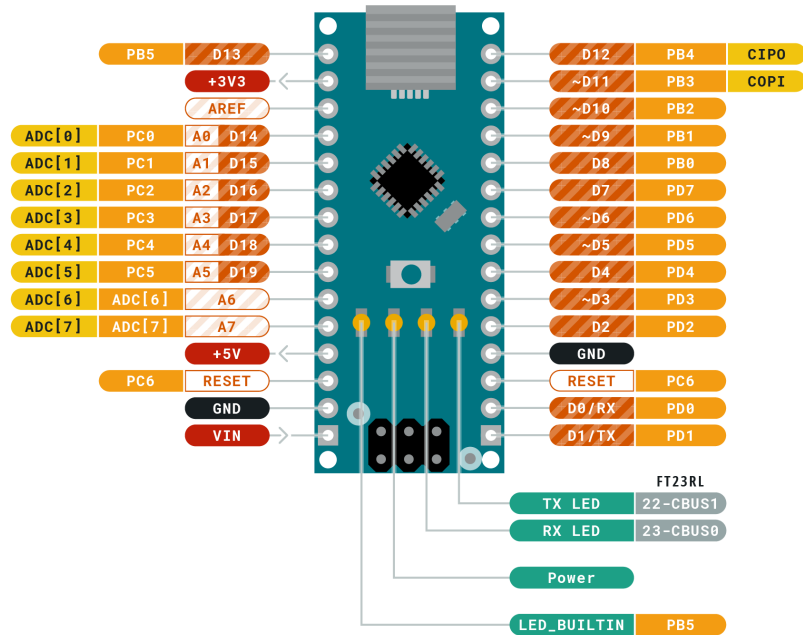
Implementación del sistema de adquisición de datos iiDAQ

3.1. Conexión de bloques para la adquisición de datos

En primer lugar, es importante conocer un poco más acerca del microcontrolador que estará manejando el funcionamiento de iiDAQ. El protocolo de comunicación que predomina es I²C; este es un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos, permitiendo el intercambio de información a una velocidad aceptable. La metodología de datos del bus I²C es en serie y sincrónica. Una de las señales del bus marcará el tiempo (pulsos de reloj) y la otra se utiliza para intercambio de datos. Por otro lado, tenemos el protocolo de comunicación SPI; este se trata de una interfaz serial síncrona, prevista para la comunicación entre dispositivos a corta distancia. No usa un protocolo estándar, y transfiere sólo paquetes de datos, lo que lo hace ideal para transferir flujo de datos extensos. En la Figura 3.1, se observa la disposición de cada uno de los pines de nuestra tarjeta de desarrollo, y con apoyo de la tabla 3.1, se pueden localizar cada uno de los puertos entrada/salida que requeriremos para el funcionamiento del sistema.



**ARDUINO
NANO**



Ground	Internal Pin	Digital Pin	Microcontroller's Port
Power	SWD Pin	Analog Pin	
LED	Other Pin	Default	

ARDUINO.CC

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Figura 3.1: Disposición de pines en Arduino Nano 3.0. ¹

Una vez identificados los puertos y los pines de conexión internas en el sistema, se deben conectar los módulos como se muestra en la Figura 3.2.

¹<https://docs.arduino.cc/hardware/nano>

Tabla 3.1: Funciones de los pines y puertos de entrada/salida del Arduino Nano 3.0, utilizados para la adquisición de datos en iiDAQ.

Pin	Función en el sistema
A5	<i>SCL (System Clock)</i> . Es la línea del bus I ² C que se encarga de los pulsos de reloj que sincronizan el sistema de adquisición. Los módulos que necesitarán de este pin serán el ADC y el DAC.
A4	<i>SDA (System Data)</i> . Es la línea del bus I ² C por la que se mueven los datos entre los dispositivos. Los módulos que necesitarán de este pin serán el ADC y el DAC.
D13	<i>SCLK (Serial Clock Input)</i> . Es el pulso que marca la sincronización del módulo. Con cada pulso de este reloj, se lee o envía un bit. El módulo que utiliza este pin es el AD9833.
D11	<i>SDATA (Serial Data Input)</i> . Es la línea por la que se mueven los datos seriales, en palabras de 16 bits. El módulo que utiliza este pin es el AD9833.
D10	<i>FSYNC (Active Low Control Input)</i> . Esta es la señal de sincronización para los datos de entrada. Cuando está en un valor bajo, se informa a la lógica interna que una nueva palabra entrará al dispositivo. El módulo que utiliza este pin es el AD9833.
D2, D3, D4, D7	<i>Pines digitales</i> . Configurables como entrada/salida.
D5, D6, D9	<i>Pines PWM</i> . Para generación de señales a partir del cambio en el ciclo de trabajo de la señal.
5V, GND	Fuente de alimentación para cada módulo del sistema y referencia GND para módulos internos y sistemas externos.

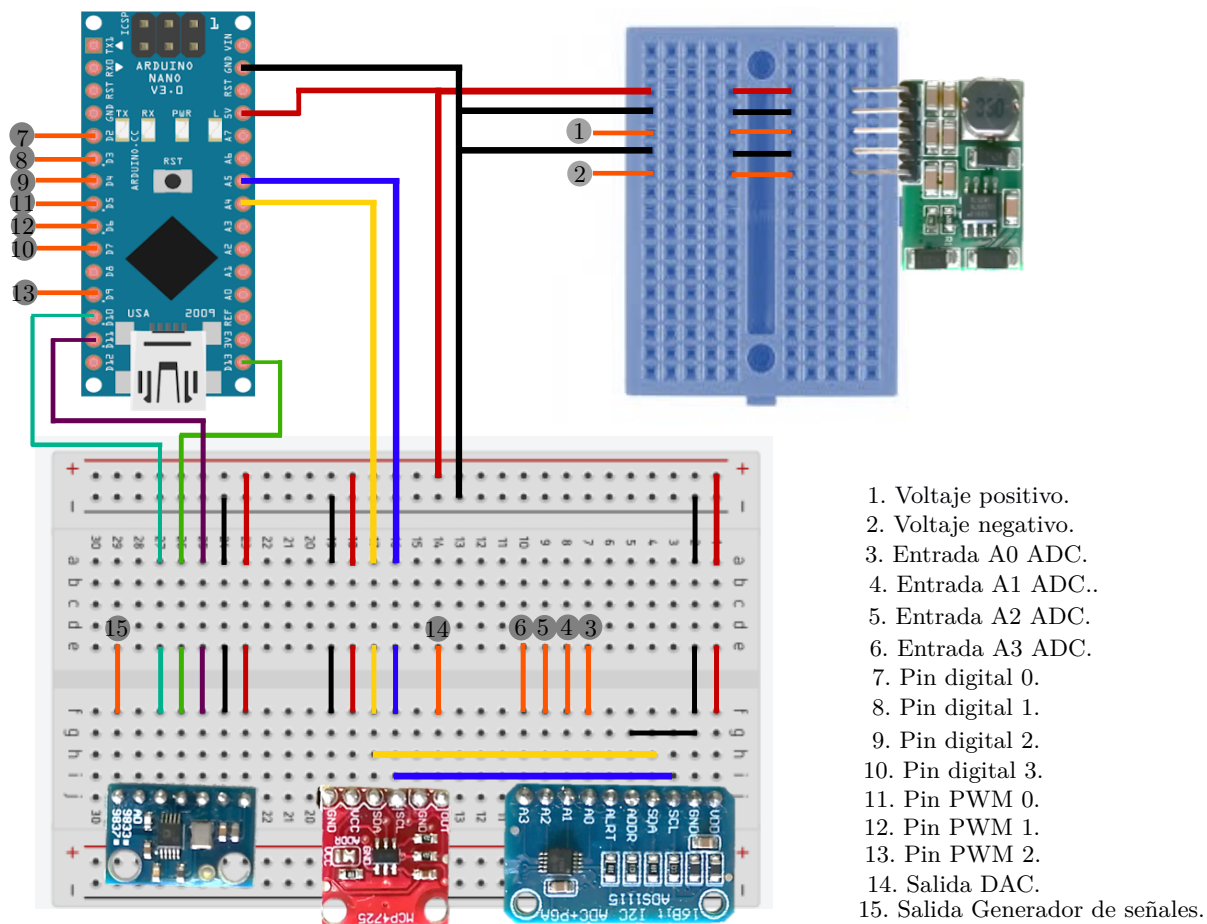


Figura 3.2: Diagrama de conexión para el sistema de adquisición de datos iiDAQ.

Es importante enfatizar que se debe seguir la conexión recomendada, puesto que el algoritmo se ha configurado para que funcione de esta manera.

3.2. Configuración del microcontrolador a partir de la IDE de Arduino

Para este paso, es necesario tener instalado el IDE Arduino ², en donde podremos compilar y subir el programa base a la tarjeta de desarrollo, todo para probar su funcionamiento. A continuación, realizar los siguientes pasos:

1. Abrir la aplicación IDE Arduino. Aparecerá una pantalla como la Figura 3.3.

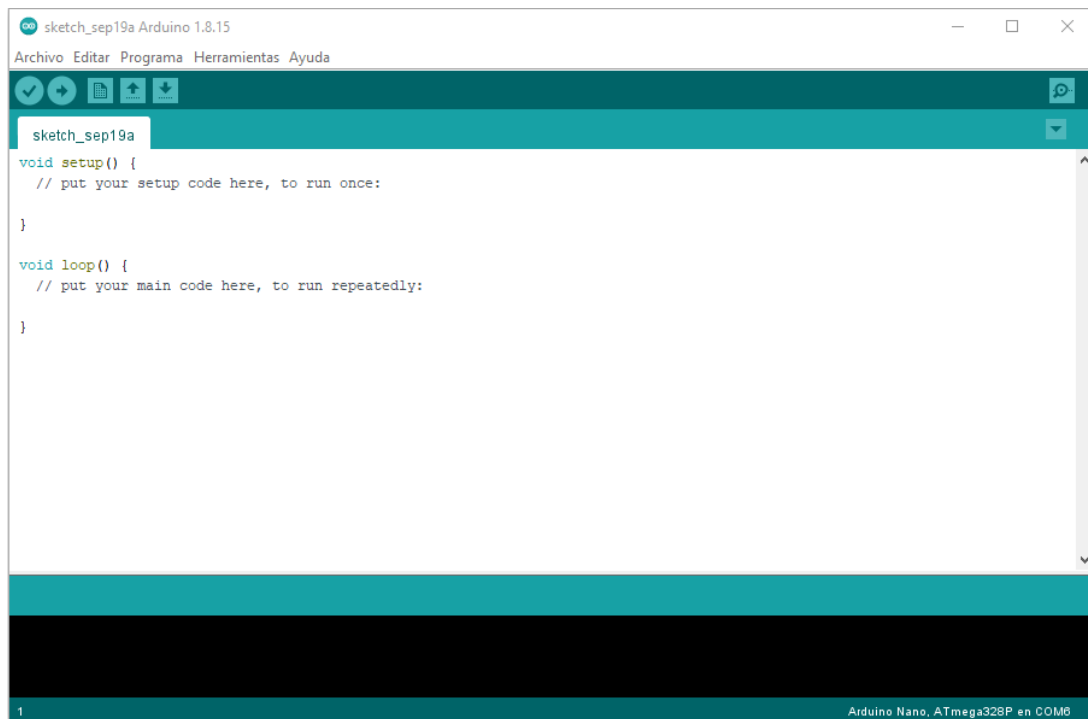


Figura 3.3: Ventana IDE Arduino.

2. En la esquina superior izquierda, dar clic en *Archivo*>*Abrir...* y buscar en la carpeta del proyecto *iiDAQ* el archivo con extensión *.ino* llamado *iiDAQ_Sketch*.

Nota importante

Una vez abierto, se recomienda no realizar ajustes o pruebas dentro del código, ya que puede modificar el funcionamiento del sistema o incluso provocar errores al momento de subir el archivo.

3. Se debe buscar el nombre de la placa y puerto en el que se está comunicando el sistema. Para esto, en la parte superior izquierda, dar clic en *Herramientas*>*Placa:*> *Arduino Nano* (Figura 3.4), que es el Arduino con el que estamos trabajando. Posteriormente, en la misma ubicación

²Este lo puedes encontrar en la página oficial de Arduino: <http://www.arduino.cc>

de la pantalla, dar clic en *Herramientas>Puerto>COM...* (Figura 3.5) y elegir el puerto donde está disponible la tarjeta de desarrollo.³

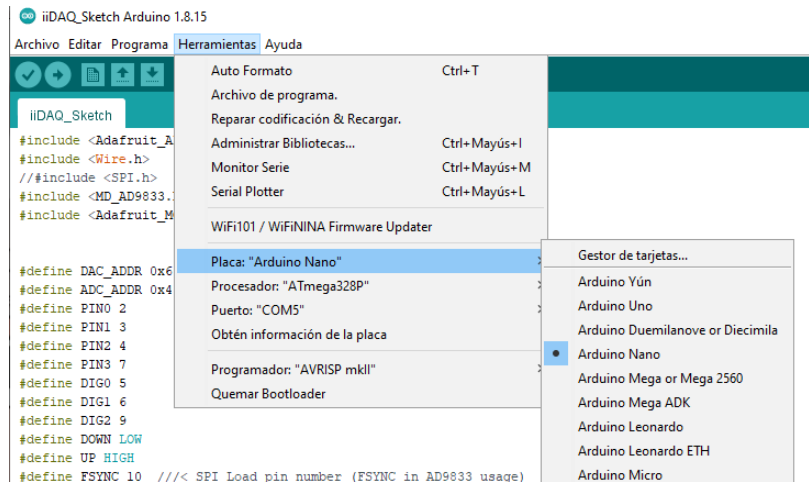


Figura 3.4: Selección de Placa.

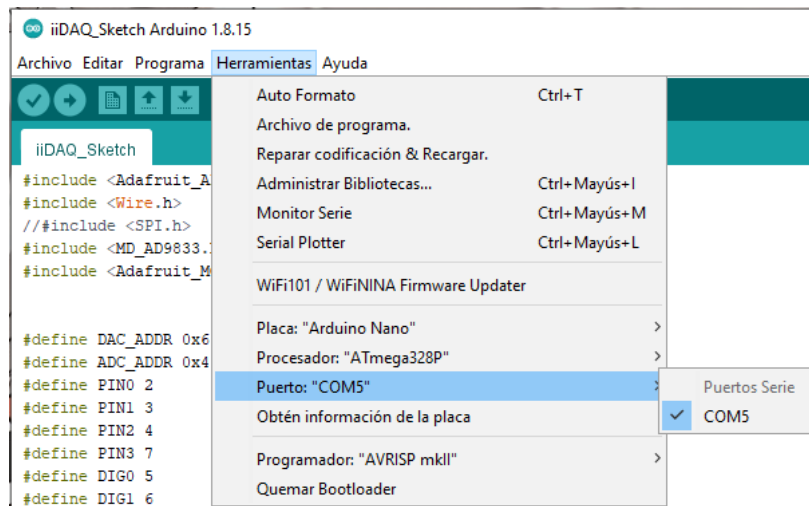


Figura 3.5: Selección de Puerto.

- En la esquina superior izquierda se muestra una barra de herramientas (Figura 3.6). Para subir el programa al sistema, dar clic en el botón de *Verificar*, lo que permitirá de cierta manera compilar el programa y revisar que no existan errores. Estará listo cuando en la parte inferior aparezca el siguiente mensaje: *Compilado*, tal como se muestra en la Figura 3.7.

³En caso de no saber el puerto COM, se debe abrir el administrador de dispositivos en la computadora y verificar en la pestaña de *Puertos (COM & LTP)* si está disponible. Si no aparece, aun cuando la tarjeta ya ha sido conectada, consultar el anexo B de este tutorial.

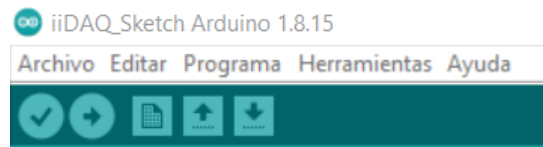


Figura 3.6: Barra para verificación y compilación del programa *iiDAQ_{sketch}*.

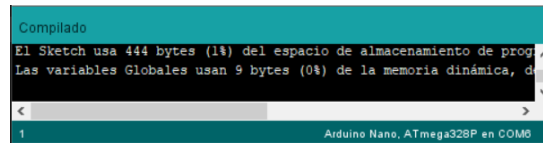


Figura 3.7: Aviso de compilación exitosa.

- Una vez que se ha verificado que no existen errores, dar clic en el botón *Subir* (Figura 3.8), el cual, finalmente, cargará el programa a la memoria de la tarjeta de desarrollo. Cuando en la parte inferior aparezca el mensaje *Subido*, el sistema es tará listo para interactuar con el toolbox de MATLAB™.



Figura 3.8: Botón para subir programa al sistema.

3.3. Instalación del Toolbox

Como se mencionó anteriormente, el sistema físico se complementa con una serie de herramientas virtuales que permiten la comunicación con cada uno de los módulos de *iiDAQ*. Dicho esto, se mostrarán los pasos a seguir para que este paquete se instale correctamente en la computadora y pueda comenzar con la interacción del sistema.

3.3.1. Requerimientos del sistema

Es importante que la computadora cumpla con los siguientes requerimientos mínimos.

- Tener instalada la aplicación de MATLAB™ $\geq 2020a$.⁴
- Memoria RAM de 4 GB (Recomendada: 8 GB).
- Disponibilidad de 350 KB en disco de almacenamiento.

3.3.2. Instalación de interfaz gráfica y funciones especiales

Instalar el *Toolbox* en el ambiente de MATLAB™ es muy sencillo, pues se ha configurado de tal manera que pueda funcionar a partir de los siguientes pasos:

⁴Es posible utilizar versiones anteriores, sin embargo, no se asegura que el funcionamiento sea el mismo en cuanto al uso de la interfaz gráfica.

1. Abrir la aplicación de MATLAB™.
2. En la carpeta del proyecto, dar doble clic al archivo con extensión *.mltbx* llamado *iiDAQ_Toolbox*. (Figura 3.9)

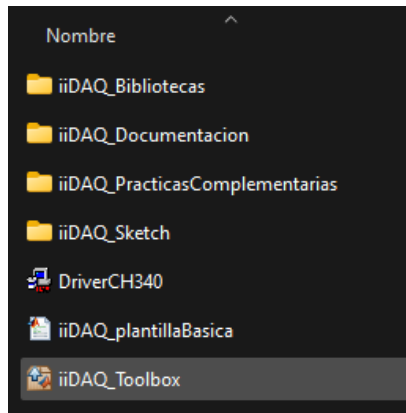


Figura 3.9: Archivo iiDAQ Toolbox en la carpeta de proyecto.

3. Una vez hecho el anterior paso, debe aparecer una pantalla como la Figura 3.10. Esto indica que la instalación se ha realizado correctamente. Dar clic en *Close*

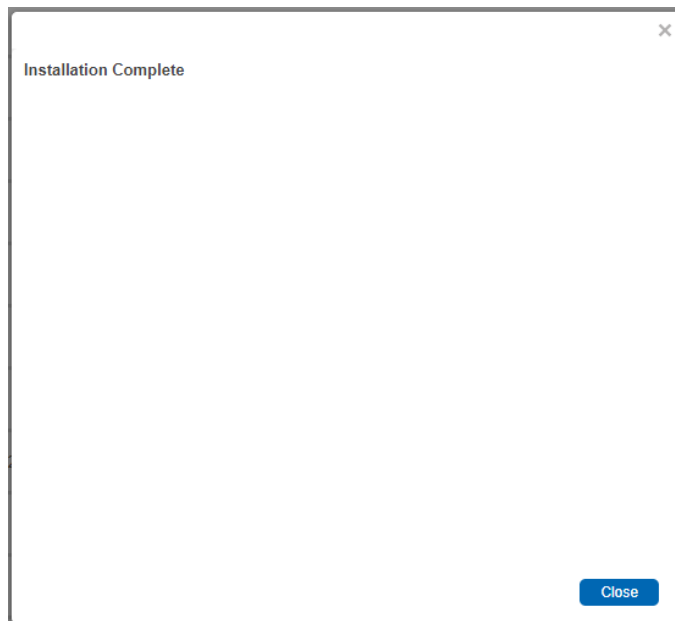


Figura 3.10: Ventana de Instalación Correcta.

4. Para verificar que se encuentre dentro de los archivos de MATLAB™, aparecerá la ventana de *Add On Manager* (Figura 3.11). Si al principio de la lista aparecen las aplicaciones de iiDAQ, todo está correcto.

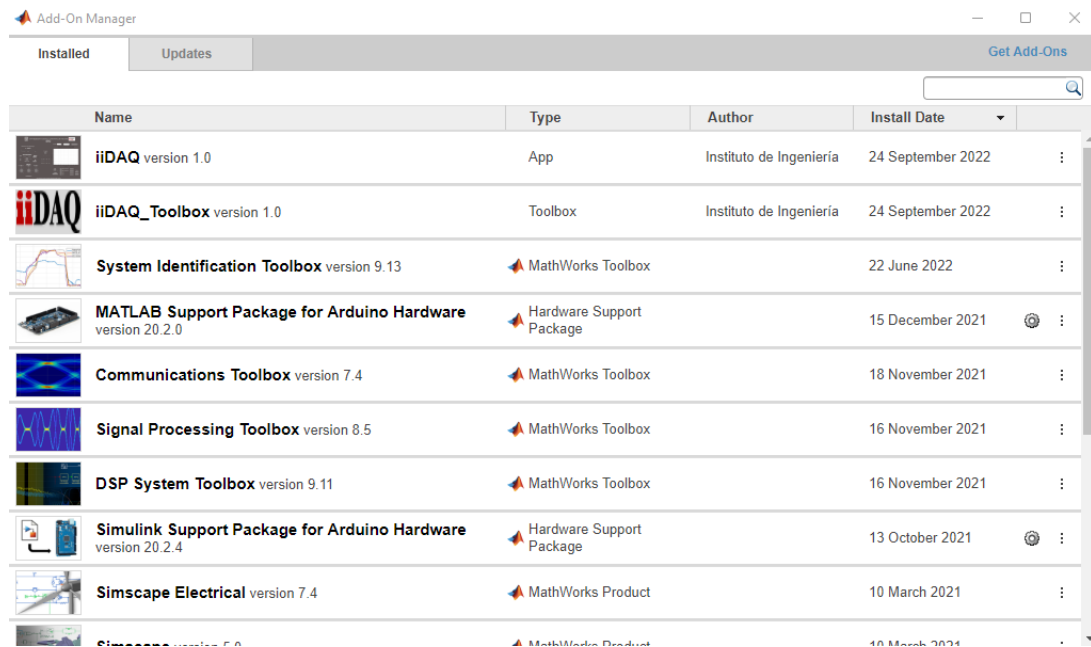


Figura 3.11: Ventana Add-On Manager. Si las aplicaciones se encuentran al inicio de la lista, estas se instalaron correctamente.

Capítulo 4

Primeros pasos en la adquisición de datos con iiDAQ

Hasta este momento, se han conocido las funcionalidades básicas que ofrece el sistema, sin embargo, es momento de conocer a fondo cuáles son las herramientas con las que cuenta el sistema para comenzar a aplicarlas en los casos de estudio que se deseen.

4.1. Interfaz gráfica

La Figura 4.1 muestra la interfaz gráfica de usuario del sistema iiDAQ, y los bloques con los que cuenta para su operación, los cuales se describen a continuación.



Figura 4.1: Interfaz gráfica de usuario para el funcionamiento básico del sistema iiDAQ.

1. *Bloque de conexión.* Configuración del puerto COM para la comunicación entre el sistema iiDAQ y la computadora, además de opciones para conectar y desconectar el sistema. En cuanto se conecta el dispositivo, se habilita el puerto COM disponible y es posible enlazar la comunicación entre la App y los módulos del sistema. Es importante elegir este puerto, ya que, de no hacerlo, se mostrará un mensaje de advertencia (Figura 4.2) y la comunicación no se enlazará. Una vez hecho el anterior paso, dar clic en *Conexión iiDAQ* y estará listo para utilizar el sistema con la interfaz gráfica. Cuando se hayan terminado las operaciones que el usuario requiera, se desconectará el sistema con ayuda del botón *Desconexión iiDAQ*, al dar clic en este.

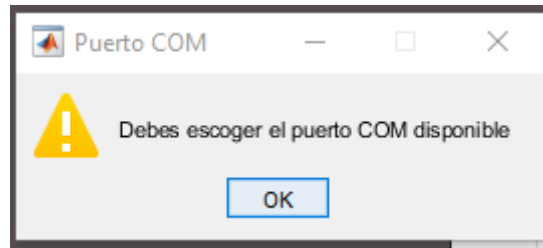


Figura 4.2: Alerta de puerto COM no disponible.

2. *Bloque módulo DAC:* Configuración de valor en bits, en un rango de 0 a 4095 para una salida de 0 a 5 volts. Se realizan cambios a partir de barra deslizable o *textbox* (caja de texto) para la alimentación de circuitos externos.
3. *Bloque módulo generador de señales:* Selecciona la forma de la señal de salida, ya sea sinusoidal, triangular o cuadrada, a partir de una perilla, además de configurar su frecuencia en Hertz (en un rango de 0 a 10 000 000) y su fase en grados (en un rango de 0 a 360), ambos a partir de una *textbox*. Disponible con dos canales para dos aplicaciones simultáneas.
4. *Bloque PWM:* Configuración del ciclo de trabajo (0% a 100%) para los pines de salida PWM. Se habilitan cada uno de los pines a partir de la casilla, dependiendo del momento en el que se usen, modificando el valor del ciclo tanto en la *textbox* como en la perilla.
5. *Bloque pines digitales I/O (Input/Output):* Configuración de pines como puertos de entrada o salida con ayuda del *switch*. Se obtienen o generan valores lógicos de 0 o 1, dependiendo el caso, a partir del cambio en el interruptor correspondiente.
6. *Bloque módulo ADC:* Configuración del modo de lectura de datos (*Single-Ended* o diferencial) y la ganancia operativa a partir de *switches*, además de la selección de los puertos se utilizarán para la adquisición de datos a partir de los botones *on/off*. Los datos obtenidos se muestran a partir de una *textbox* en su equivalencia a voltaje (ya sea en Volts o miliVolts, opción elegida a partir de una lista desplegable), además de la opción de un botón registrador de datos (*datalogger*).

Como se ha descrito anteriormente, cada uno de estos bloques ayudará al usuario a familiarizarse con las herramientas básicas del sistema iiDAQ, y así, entender el funcionamiento básico de la instrumentación y adquisición de datos.

4.2. Modo de comandos aplicados en scripts

En la Tabla 4.1, se describen las funciones disponibles para poder utilizarlas en la programación de *scripts* en MATLAB™. Para el caso del módulo ADC, es importante aclarar que el tipo de dato

Tabla 4.1: Funciones del *toolbox* para desarrollo de aplicaciones personalizadas. La variable **s** será el objeto del puerto serie.

Función	Descripción/Parámetros [valor]
connectiiDAQ(serialPort)	Conecta el sistema iiDAQ con el puerto serie disponible en la computadora. Esta función debe ser asignada a una variable s. <ul style="list-style-type: none"> ■ serialPort: Cadena para identificación de puerto serial disponible ['COM#'], donde '#' es el número de puerto.
disconnectiiDAQ(s)	Desconecta el sistema iiDAQ y cierra el puerto serie utilizado.
configureDAC (s, value)	Genera un voltaje usando el MCP4725, en el rango de 0 a 5 volts. <ul style="list-style-type: none"> ■ value: Entero [0-4095].
signalGenerator (s, channel, waveForm, Frequency)	Genera una señal a través del AD9833. <ul style="list-style-type: none"> ■ channel: Entero para seleccionar canal [0, 1] ■ waveForm: Cadena para selección de forma ["Sine", "Triangle", "Square"] ■ Frequency: Valor de tipo entero para frecuencia [0-1E7]
configurePWM (s, pin, dutyCycle)	Genera señales PWM en la terminal indicada y configura el ciclo de trabajo. <ul style="list-style-type: none"> ■ pin: Entero para seleccionar terminal [0, 1, 2] ■ dutyCycle: Entero para definir % ciclo de trabajo [0-100]
configureDigitalPin (s, digitalPin, value, type)	Genera señales digital de entrada o de salida en el pin requerido. <ul style="list-style-type: none"> ■ digitalPin: Entero para seleccionar pin [0, 1, 2, 3] ■ value: Entero para definir estado lógico bajo (0) o alto (1) ■ type: Entero para definir entrada (0) o salida (1)
configureADC(s, mode, gain)	Configura el ADS1115. <ul style="list-style-type: none"> ■ mode: Entero para seleccionar modo <i>single-ended</i> (0) o diferencial (1) ■ gain: Entero para definir ganancia [0-5]
analogRead(s, pin)	Adquiere un valor analógico del ADS1115. <ul style="list-style-type: none"> ■ pin: Entero para seleccionar pin [0, 1, 2, 3]

de salida será en bits, por lo tanto, es necesario multiplicar el valor por un factor si se desea obtener una salida en volts. Este valor es $\frac{3.3}{17000}$, el cual se observará con más detalle en el capítulo 5 de este tutorial.

Nota importante

Es posible que mientras se realicen pruebas en los programas realizados por el usuario, se cometan errores de programación y MATLAB™ marque un error, deteniendo la ejecución del programa. Esto provoca que volvamos a ejecutar el algoritmo una vez que se haya realizado la corrección; sin embargo, es importante que, antes de volver a ejecutarlo, implementes la función ***disconnectiiDAQ(s)*** en la ventana de comandos; así, se asegura que el puerto COM del dispositivo se haya cerrado correctamente y pueda volver a utilizarse en la siguiente ejecución.

4.2.1. Plantilla básica para la implementación de scripts

Con el objetivo de que los usuarios comiencen a familiarizarse con la implementación de las funciones en casos más específicos, se presenta la siguiente plantilla de inicio¹ fundamental para que los programas funcionen correctamente.

```

1  %Universidad Nacional Autonoma de Mexico
2  %Instituto de Ingenieria
3  %Sistema de adquisicion de datos iiDAQ
4
5  %Se descartan variables y objetos utilizados en programas anteriores.
6
7  clear all
8  close all
9  clc
10
11  %-----
12
13  %Programar en este apartado, en caso de ser necesario, la asignacion de
14  %variables o lineas de codigo necesarias antes de la conexion del sistema
15  %de adquisicion de datos iiDAQ.
16
17  %-----
18
19  %En este apartado, se procede a conectar el sistema.
20
21  s = connectiiDAQ();
22
23  %-----
24
25  %En este apartado, programar las funciones presentadas en el tutorial,
26  %para la configuracion y uso de los modulos y puertos disponibles
27  %en el sistema.
28
29  %-----
30
31  %En este apartado, se procede a desconectar el sistema.
32
33  disconnectiiDAQ(s);

```

¹Esta plantilla se puede encontrar dentro de la carpeta del proyecto *iiDAQ* como un archivo con extensión *.m* nombrado *iiDAQ_plantillaBasica*.

Capítulo 5

Prácticas complementarias para el uso del sistema iiDAQ

A continuación, se pondrá en práctica el sistema de adquisición de datos iiDAQ, a partir de los conocimientos adquiridos en este tutorial, con respecto a todas las herramientas que se tienen disponibles.

Para el caso de la interfaz gráfica, iniciar siempre con los siguientes pasos.

1. Abrir la aplicación de MATLAB TM.

2. En la pestaña *Apps*, dentro del apartado *Apps*, dar clic en el icono de la aplicación *iiDAQ* (Figura 5.1), aparecerá una ventana como la Figura 5.2.

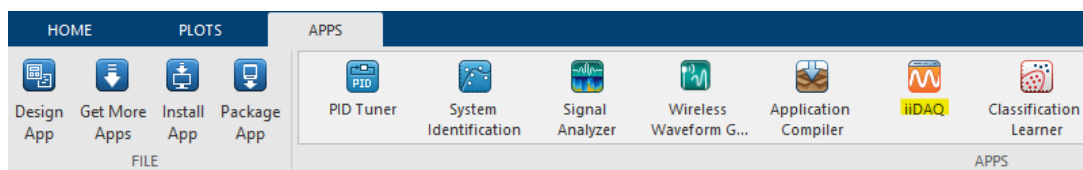


Figura 5.1: Ícono de aplicación iiDAQ.



Figura 5.2: Ventana de Sistema iiDAQ

Para el caso del modo de comandos, iniciar siempre con los siguientes pasos.

1. Abrir la aplicación de MATLAB™.
2. En la pestaña *HOME* > *New Script* (Figura 5.3), dar clic para abrir un Script y comenzar a programar las líneas de código necesarias para la adquisición de datos con iiDAQ. En su defecto, podemos adelantar el proceso con ayuda de nuestra plantilla básica, para esto, dar clic en *HOME* > *Open* (Figura 5.4) y buscar el archivo en nuestra carpeta de proyecto iiDAQ.

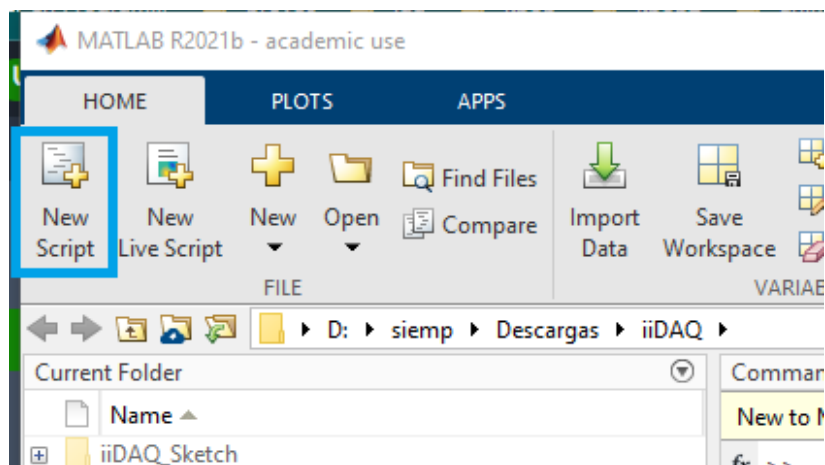


Figura 5.3: Nuevo Script

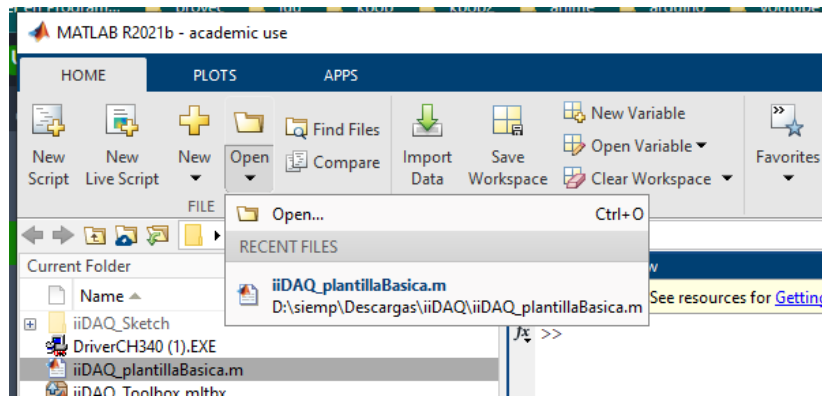


Figura 5.4: Abrir Proyecto

3. Guardar el archivo en *EDITOR*>*Save*>*Save As* y colocar el nombre que se desee, sin números ni caracteres especiales.

Con ayuda de estos pasos, es posible comenzar con cada una de las prácticas propuestas ¹ por este tutorial, con el objetivo de obtener una clara idea del funcionamiento del sistema y comenzar con aplicaciones más complejas orientadas a la instrumentación y control.

¹Para el modo de comandos, las prácticas complementarias se encuentran dentro de la carpeta del proyecto con el nombre *iiDAQ_PracticasComplementarias*.

5.1. Práctica 1. Encender un led

Para esta práctica, se requieren los siguientes materiales:

- Protoboard
- 4 Leds
- 4 resistencias de $220[\Omega]$ o $330[\Omega]$

Se presentan los pasos a seguir en esta práctica :

5.1.1. Interfaz gráfica

1. Conectar el sistema iIDAQ a la computadora y elegir el puerto COM disponible. Una vez hecho el anterior paso, dar clic en el botón *Conexión iIDAQ* (Figura 5.5).



Figura 5.5: Selección de puerto en el sistema iIDAQ

2. Construir el circuito de la Figura 5.6.

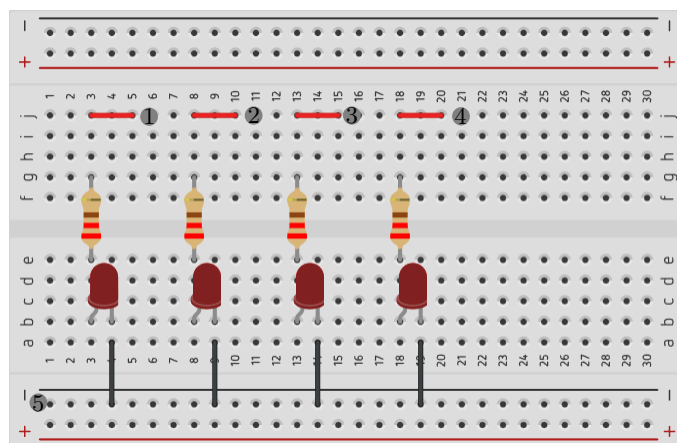


Figura 5.6: Circuito para análisis de Práctica 1. Las conexiones al sistema iIDAQ serán las siguientes: (1) Pin digital 0, (2) Pin digital 1, (3) Pin digital 2, (4) Pin digital 3, (5) GND del sistema.

- Ubicar el bloque de *Pines digitales* (Figura 5.7); asegurar que esté activada la opción de pin de salida (*Output*).

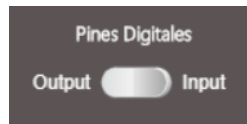


Figura 5.7: Pin de salida

- Ubicar el interruptor del pin 0 (*P0*) y dar clic (Figura 5.8). Se puede observar que el led se encenderá, esto debido a que el valor lógico que se está mandando es 1, equivalente a 5 [V] de alimentación.

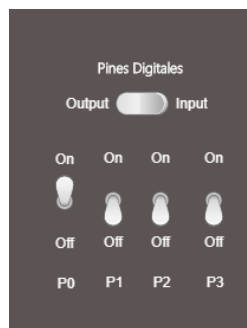


Figura 5.8: Interruptor P0

- Repetir desde el paso número 3 para la conexión de los demás leds y realizar las combinaciones que el usuario desee.

5.1.2. Modo de comandos

- Conectar el sistema iiDAQ a la computadora.
- Construir el circuito de la Figura 5.9.

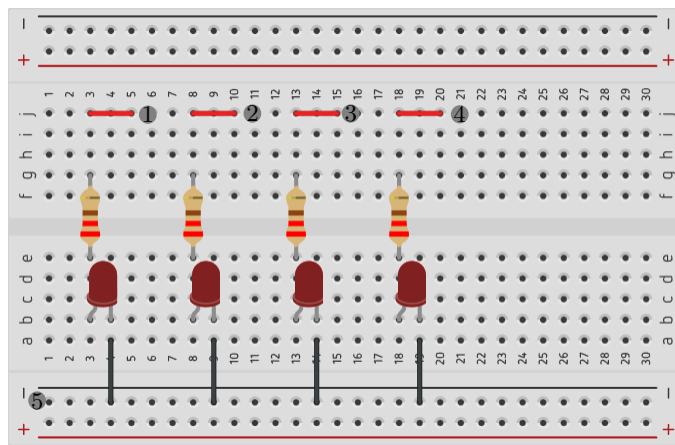


Figura 5.9: Circuito para análisis de Práctica 1. Las conexiones al sistema iiDAQ serán las siguientes: (1)Pin digital 0, (2)Pin digital 1, (3)Pin digital 2, (4)Pin digital 3, (5)GND del sistema.

3. En el script ya configurado, colocar las siguientes líneas de código.

```

1  %Universidad Nacional Autonoma de Mexico
2  %Instituto de Ingenieria
3  %Sistema de adquisicion de datos iiDAQ
4
5  clear all
6  close all
7  clc
8
9  %-----
10
11  i = 1;
12  digitalPin = 0;
13  value = 0;
14  typePin = 1;
15
16  %-----
17
18  s = connectiiDAQ();
19
20  %-----
21
22  %Aseguramos un valor inicial logico cero en todos los pines.
23
24  configureDigitalPin (s, 0, value, typePin);
25  configureDigitalPin (s, 1, value, typePin);
26  configureDigitalPin (s, 2, value, typePin);
27  configureDigitalPin (s, 3, value, typePin);
28
29  disp('Iniciando programa...');
30  pause(2);
31
32  while i == 1
33
34      digitalPin = input("Elegir el pin digital que se desea: ");
35      value = input("Colocar el valor logico de salida: ");
36      configureDigitalPin (s, digitalPin, value, typePin); %Configuramos el
37      pin digital para encender o apagar el led.
38      i = input("Para intentarlo de nuevo, escribir 1, en caso contrario,
39      escribir 0: ");
40
41      if i == 0
42          break;
43      end
44
45  end
46
47  %-----
48  disconnectiiDAQ(s);

```

4. Una vez listo el programa, dar clic en *EDITOR>Run* (Figura 5.10).

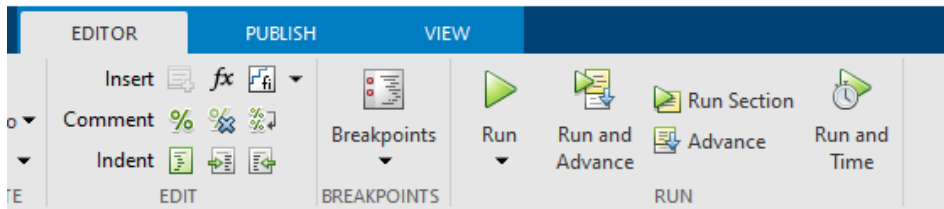


Figura 5.10: Botón de ejecución para la prueba del *script* en el sistema iiDAQ.

- Seguir las instrucciones en la ventana de comandos y reiniciar el programa las veces necesarias, con el objetivo de probar todas las combinaciones posibles en la configuración de los pines digitales.

5.1.3. Resultados esperados

En el caso del *modo de comandos*, los mensajes que visualizaremos en la ventana de comandos serán los siguientes:

```

Command Window
Dispositivo conectado
Configurando pin digital...
Pin digital 0, Tipo: Salida, Valor lógico: 0
Configurando pin digital...
Pin digital 1, Tipo: Salida, Valor lógico: 0
Configurando pin digital...
Pin digital 2, Tipo: Salida, Valor lógico: 0
Configurando pin digital...
Pin digital 3, Tipo: Salida, Valor lógico: 0
Iniciando programa...
  
```

Figura 5.11: Configuración de los módulos para inicio de programa.

```

Command Window
Elegir el pin digital que se desea: 1
Colocar el valor lógico de salida: 1
Configurando pin digital...
Pin digital 1, Tipo: Salida, Valor lógico: 1
Para intentarlo de nuevo, escribir 1, en caso contrario, escribir 0: 0
Dispositivo Desconectado
  
```

Figura 5.12: Configuración de los pines digitales por parte del usuario. Los valores pueden ser a consideración, repitiendo las pruebas las veces que se necesiten.

Los leds deberán encender y apagar de acuerdo con las opciones que el usuario indique.

5.2. Práctica 2. Modulación de ancho de pulso con un led

Para esta práctica, se requieren los siguientes materiales:

- Protoboard
- 3 Leds
- 3 resistencias de $220[\Omega]$ o $330[\Omega]$

Se presentan los pasos a seguir en esta práctica :

5.2.1. Interfaz gráfica

1. Conectar el sistema iiDAQ a la computadora y elegir el puerto COM disponible. Una vez hecho el anterior paso, dar clic en el botón *Conexión iiDAQ* (Figura 5.13).



Figura 5.13: Conexión del sistema

2. Construir el circuito de la Figura 5.14.

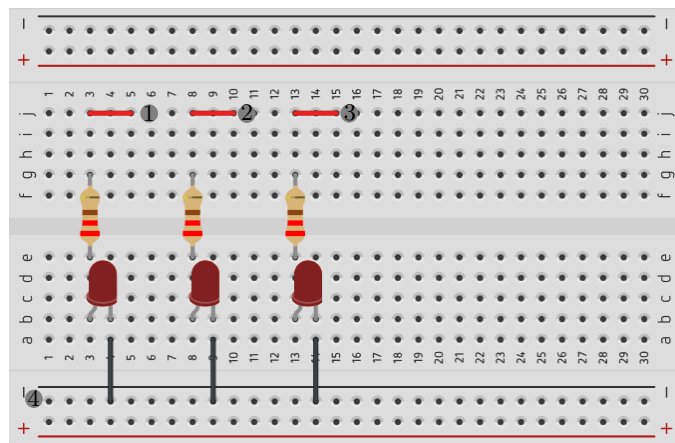


Figura 5.14: Circuito para análisis de Práctica 2. Las conexiones al sistema iiDAQ serán las siguientes: (1)Pin PWM 0, (2)Pin PWM 1, (3)Pin PWM 2, (4)GND del sistema.

3. Ubicar el bloque de *Señales PWM*; activamos la casilla θ , la perilla y la caja de texto se activarán (Figura 5.15).

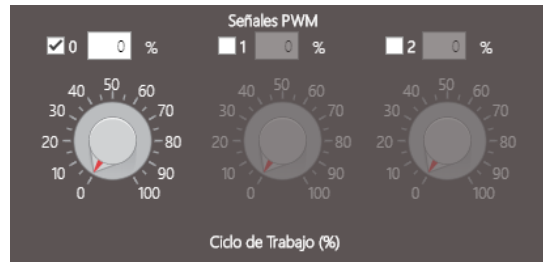


Figura 5.15: bloque de Señales PWM

4. Mover la perilla en los valores que se deseen o, también, colocar un valor en la caja de texto (Figura 5.16). Se observa cómo cambia la intensidad del led dependiendo del ciclo de trabajo de la señal, siendo tenue en los valores más bajos y brillante en los valores más altos.

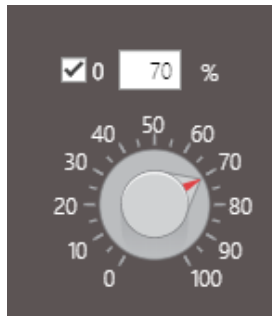


Figura 5.16: Selección de valor PWM

5. Repetir desde el paso número 2 para la conexión de los demás leds y realizar las combinaciones de luminosidad que el usuario desee.

5.2.2. Modo de comandos

1. Conectar el sistema iiDAQ a la computadora.
2. Construir el circuito de la Figura 5.17.

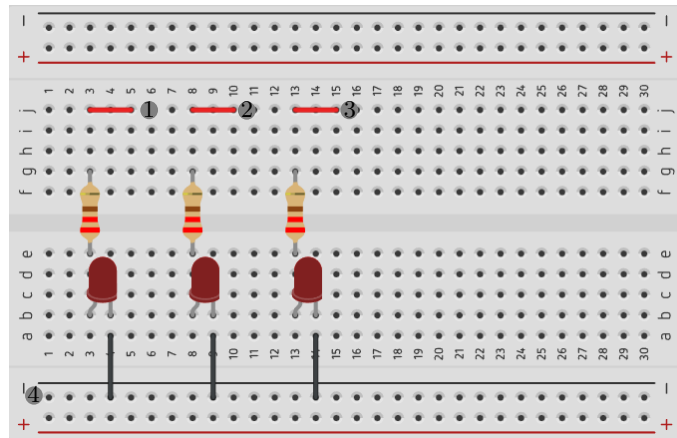


Figura 5.17: Circuito para análisis de Práctica 2. Las conexiones al sistema iiDAQ serán las siguientes: (1)Pin PWM 0, (2)Pin PWM 1, (3)Pin PWM 2, (4)GND del sistema.

3. En el script ya configurado, colocar las siguientes líneas de código.

```

1  %Universidad Nacional Autonoma de Mexico
2  %Instituto de Ingenieria
3  %Sistema de adquisicion de datos iiDAQ
4
5  clear all
6  close all
7  clc
8
9  %-----
10 i = 1;
11 pin = 0;
12 dutyCycle = 0;
13 %-----
14
15 s = connectiiDAQ();
16
17 %-----
18
19 %Aseguramos un valor inicial cero en el ciclo de trabajo de los pines PWM.
20
21 configurePWM (s, 0, dutyCycle);
22 configurePWM (s, 1, dutyCycle);
23 configurePWM (s, 2, dutyCycle);
24
25 disp('Iniciando programa...')
26 pause(2)
27 clc
28
29 while i == 1
30
31     pin = input("Elegir el pin PWM que se desea: ");
32     dutyCycle = input("Colocar el valor del ciclo de trabajo: ");
33     configurePWM (s, pin, dutyCycle); %Configuramos el pin PWM con el ciclo
        de trabajo correspondiente.

```

```

34     i = input("Para intentarlo de nuevo, escribir 1, en caso contrario,
35             escribir 0: ");
36
37     if i == 0
38         break;
39     end
40 end
41
42 %-----
43
44 disconnectiiDAQ(s);

```

4. Una vez listo el programa, dar clic en *EDITOR>Run* (Figura 5.18).

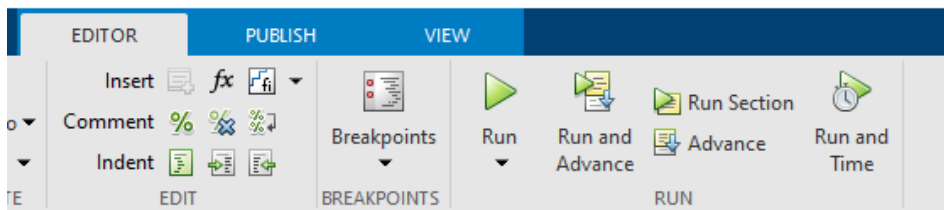


Figura 5.18: Botón de ejecución para la prueba del *script* en el sistema iiDAQ.

5. Seguir las instrucciones en la ventana de comandos y reiniciar el programa las veces necesarias, con el objetivo de probar todas las combinaciones posibles en la configuración de los pines digitales.

5.2.3. Resultados esperados

En el caso del *modo de comandos*, los mensajes que visualizaremos en la ventana de comando serán los siguientes:

```

Command Window
Dispositivo conectado
Pin PWM 0 configurado correctamente
Pin PWM 1 configurado correctamente
Pin PWM 2 configurado correctamente
Iniciando programa...

```

Figura 5.19: Configuración de los módulos para inicio de programa.

```
Command Window
Elegir el pin PWM que se desea: 2
Colocar el valor del ciclo de trabajo: 55
Pin PWM 2 configurado correctamente
Para intentarlo de nuevo, escribir 1, en caso contrario, escribir 0: 0
Dispositivo Desconectado
```

Figura 5.20: Configuración de los pines PWM por parte del usuario. Los valores pueden ser a consideración, repitiendo las pruebas las veces que se necesiten.

La intensidad de los leds debe variar de acuerdo con el valor de ciclo de trabajo que el usuario ingrese, repitiendo las pruebas las veces que sean necesarias.

5.3. Práctica 3. Medición de voltaje en un circuito resistivo

Para esta práctica, se requieren los siguientes materiales:

- Protoboard
- 1 resistencia de $1[k\Omega]$
- 1 Potenciómetro de $1[k\Omega]$

Se presentan los pasos a seguir en esta práctica:

5.3.1. Interfaz gráfica

1. Conectar el sistema iDAQ a la computadora y elegir el puerto COM disponible. Una vez hecho el anterior paso, dar clic en el botón *Conexión iDAQ*.
2. Construir el circuito de la Figura 5.21. Conectar la salida del DAC como fuente de alimentación y el pin 0 del ADC en la terminal señalada con el número (2) en la Figura 5.21.

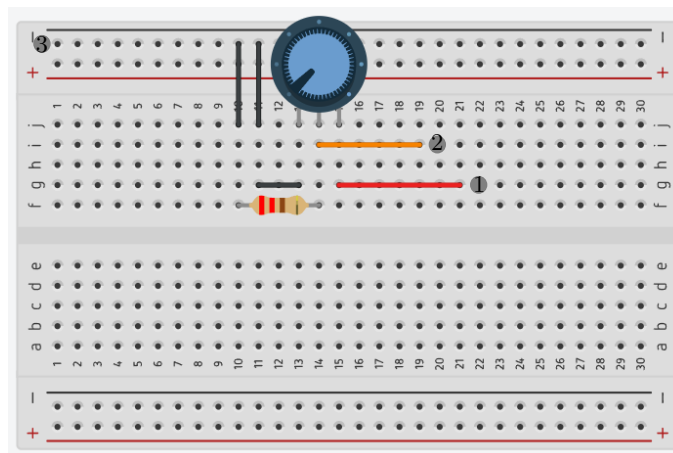


Figura 5.21: Circuito para análisis de Práctica 3. Las conexiones al sistema iDAQ serán las siguientes: (1)Puerto de salida DAC, (2)Entrada ADC pin A0 o A1, (3)GND del sistema.

3. Ubicar el bloque de *Módulo DAC*; en la caja de texto o en la barra deslizador, colocar el valor de 4095 equivalente a una fuente de alimentación de 5 Volts (Figura 5.22).

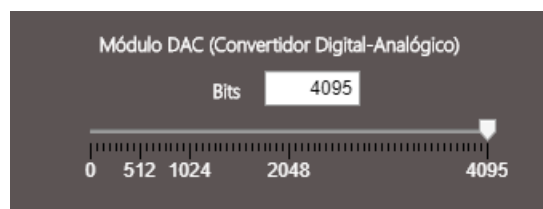


Figura 5.22: Selección de valor DAC.

4. Ubicar en el bloque *Módulo ADC*; asegurar que el modo esté habilitado en *Single* y la ganancia en 1 (Figura 5.23).

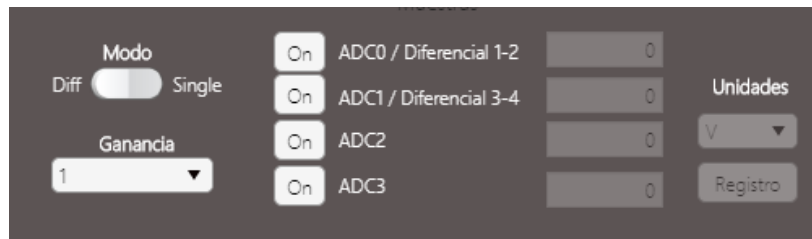


Figura 5.23: Selección de valor ADC.

5. Dar clic en el botón *On* del ADC0. Observar la gráfica y ver que el sistema leerá el valor de voltaje del potenciómetro en tiempo real (Figura 5.24); este valor cambiará de acuerdo con la variación que el usuario le dé. Elegir las unidades en las que se requieren obtener las lecturas (Volts o milivolts).

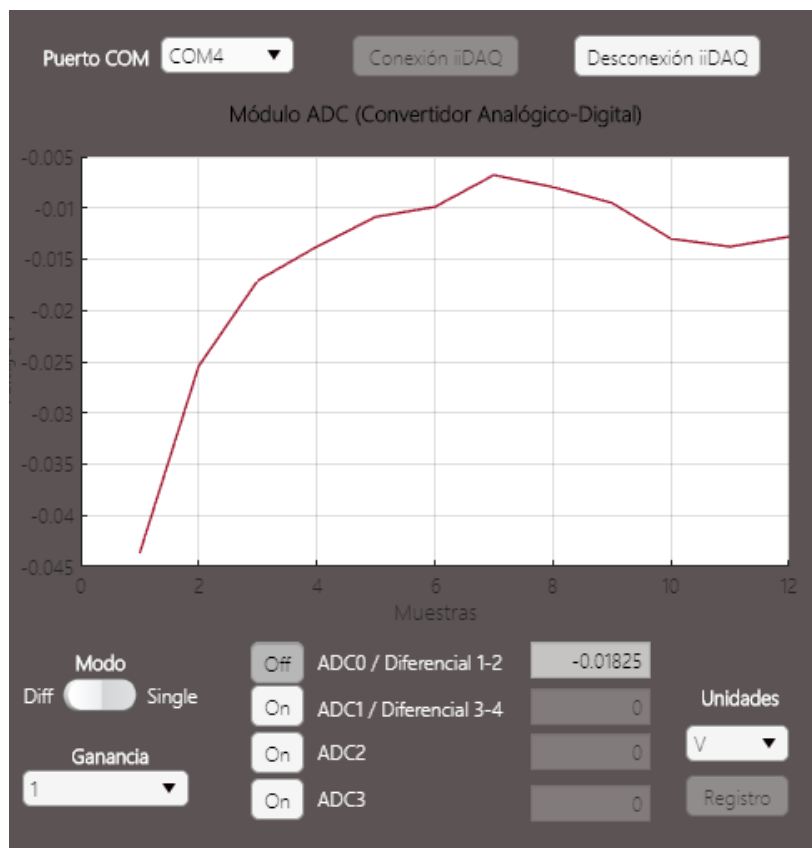


Figura 5.24: Lectura del modulo ADC.

6. Presionar el botón *Off* del ADC0 para que la lectura se detenga.
7. Repetir el paso 5 pero ahora en el ADC1 para obtener la misma lectura mientras se varía el valor del potenciómetro.
8. Repetir desde el paso número 3 para la variación del voltaje de alimentación diferente a 4095 (5 [V]) y con valores de resistencia variados.

5.3.2. Modo de comandos

1. Conectar el sistema iiDAQ a la computadora.
2. Construir el circuito de la Figura 5.25. Conectar la salida del DAC como fuente de alimentación y el pin 0 del ADC en la terminal señalada con el número (2) en la Figura 5.25.

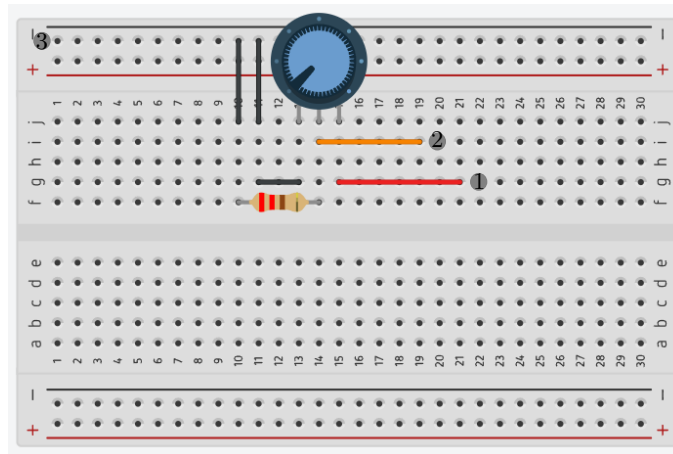


Figura 5.25: Circuito para análisis de Práctica 3. Las conexiones al sistema iiDAQ serán las siguientes: (1)Puerto de salida DAC, (2)Entrada ADC pin A0 o A1, (3)GND del sistema.

3. En el script ya configurado, colocar las siguientes líneas de código.

```

1
2  %Universidad Nacional Autonoma de Mexico
3  %Instituto de Ingenieria
4  %Sistema de adquisicion de datos iiDAQ
5
6  clear all
7  close all
8  clc
9
10 %-----
11
12 value = 0;
13 pin = 0;
14 mode = 0;
15 gain = 1;
16 N = 100; %Numero de muestras
17 ii = 1;
18 A0 = 0;
19
20 %-----
21
22 s = connectiiDAQ();
23
24 %-----
25
26 %Aseguramos un valor inicial cero en la salida del DAC

```

```

27 disp('Iniciando programa...')
28 configureDAC(s, value);
29
30 value = input("Elegir el voltaje de salida para el DAC en bits: ");
31 configureDAC(s, value); %Configuramos el voltaje de salida.
32
33 %A continuacion, se realizaran las lecturas del circuito
34 clc
35 disp("Obteniendo mediciones del circuito...");
36
37 for ii=1:N
38     VR1(ii) = ((analogRead(s, A0)*3.3)/17000); %Se multiplica por
39                                               %factor de
40                                               %conversion
41     plot(VR1)
42     drawnow
43 end
44
45 disp("Mediciones obtenidas");
46 pause(1);
47
48 figure
49 plot(VR1, 'LineWidth',1, 'Color', '#A2142F');
50 ylabel('Voltaje_(V)')
51 xlabel('Muestra')
52 legend('Voltaje_Potenci metro', 'Location', 'Best')
53 %-----
54
55 disconnectiiDAQ(s);

```

4. Una vez listo el programa, dar clic en *EDITOR>Run* (Figura 5.26).

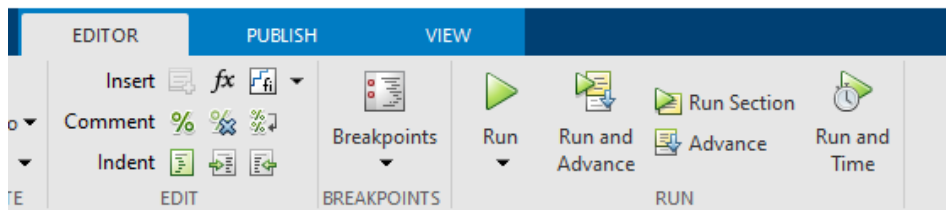


Figura 5.26: Botón de ejecución para la prueba del *script* en el sistema iiDAQ.

5. Seguir las instrucciones en la ventana de comandos y reiniciar el programa las veces necesarias, con el objetivo de obtener diferentes gráficas y comportamientos.

5.3.3. Resultados esperados

En el caso de la *interfaz gráfica*, podemos obtener diferentes lecturas de acuerdo con el valor del potenciómetro que se mida en ese momento, obteniendo resultados como las Figuras 5.27 y 5.28.

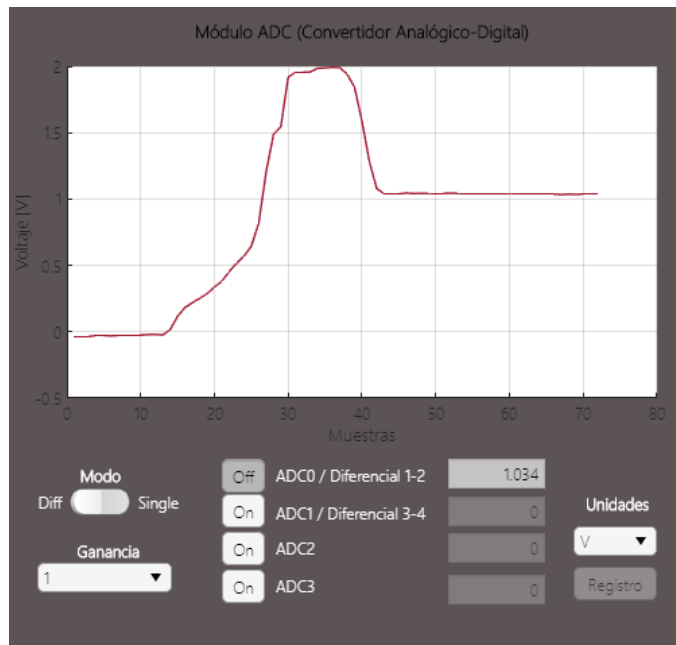


Figura 5.27: Prueba 1. Resultado de medición de voltaje (Volts) en potenciómetro.

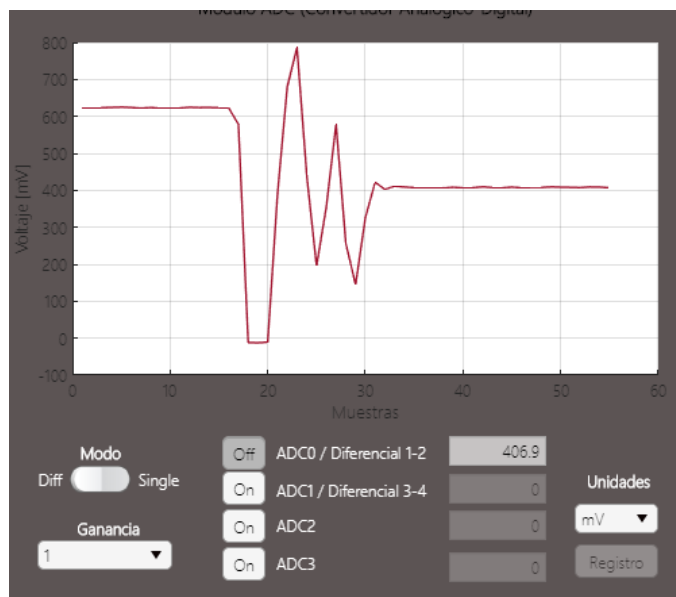


Figura 5.28: Prueba 2. Resultado de medición de voltaje (miliVolts) en potenciómetro.

En el caso del *modo de comandos*, los mensajes que visualizaremos en la ventana de comando serán los siguientes:

```

Command Window
Dispositivo conectado
Iniciando programa...
fx Elegir el voltaje de salida para el DAC en bits: 4095

```

Figura 5.29: Configuración de los módulos para inicio de programa.

```

Command Window
Obteniendo mediciones del circuito...
Mediciones obtenidas
Dispositivo Desconectado

```

Figura 5.30: Lectura de mediciones obtenidas del pin A0. Mientras se obtienen las muestras, se mostrará este mensaje y, una vez finalizada, el dispositivo se desconectará.

Mientras las lecturas se estén tomando, una gráfica se mostrará durante el periodo de tiempo que se tomen las muestras indicadas en el código (Figura 5.31). El número de muestras puede cambiar a consideración.

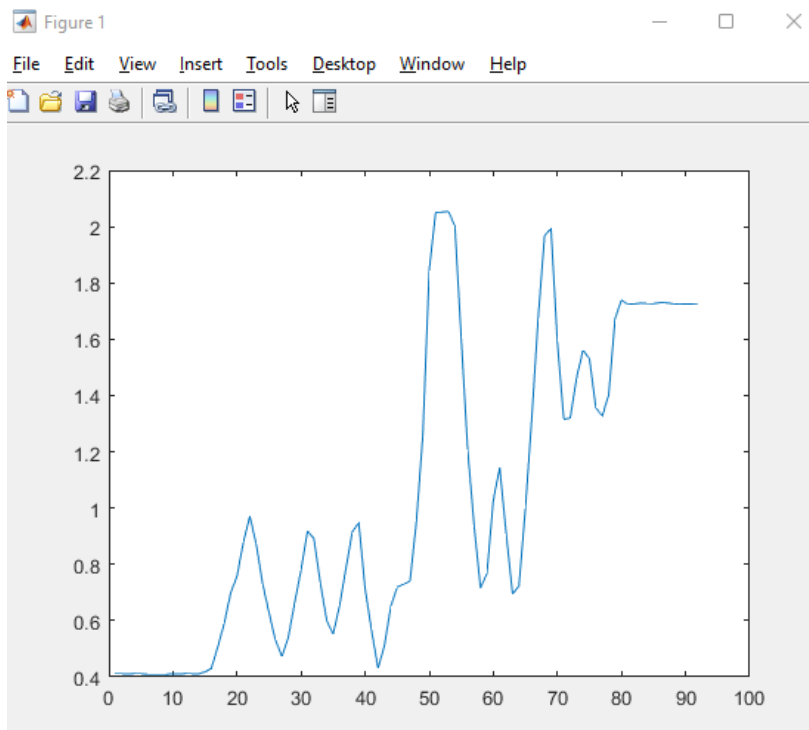


Figura 5.31: Gráfica obtenida en tiempo real. Permite visualizar los cambios en la lectura del pin A0.

Una vez que se haya finalizado la medición, obtendrás una gráfica resultante (Figura 5.32).

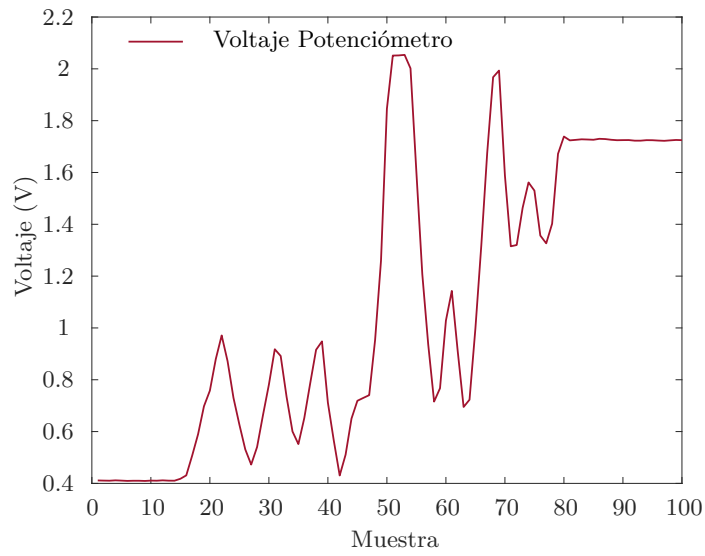


Figura 5.32: Gráfica final. Permite visualizar y analizar los datos obtenidos una vez finalizadas las mediciones.

5.4. Práctica 4. Generación de onda vista en un osciloscopio

Para esta práctica, se requieren los siguientes materiales:

- Osciloscopio

Se presentan los pasos a seguir en esta práctica:

5.4.1. Interfaz gráfica

1. Conectar el sistema iiDAQ a la computadora y elegiremos el puerto COM disponible. Una vez hecho el anterior paso, dar clic en el botón *Conexión iiDAQ*.
2. Conectar la salida del generador de funciones a la entrada de nuestro osciloscopio, tal como se muestra en la Figura 5.33.

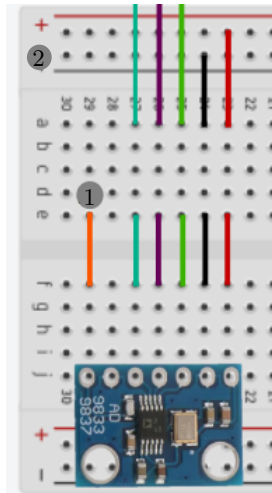


Figura 5.33: Circuito para análisis de Práctica 4. En este no se aplicará ningún circuito externo, ya que será la medida directa del módulo generador de señales. Las conexiones del sistema serán las siguientes: (1)Punta positiva del canal 1 del osciloscopio, (2)Referencia a tierra del canal 1 del osciloscopio.

3. Ubicar el bloque de *Módulo Generador de señales*; ubicar la perilla *Forma de onda* y colocar en la posición de *Sinusoidal* (Figura 5.34).

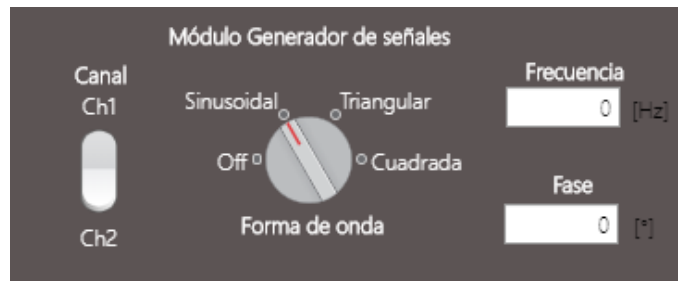


Figura 5.34: Módulo Generador de Señales

4. Asegurar que esté seleccionado el canal *Ch1*. Se verá en la pantalla del osciloscopio el cambio en la forma de la señal. Así, es posible probar las diferentes formas de onda en la perilla.
5. En el textbox de frecuencia, colocar un valor de 100 y presionar en el osciloscopio el botón de *Auto-set*, el resultado será una señal con una frecuencia de 100 [Hz] (Figura 5.35).

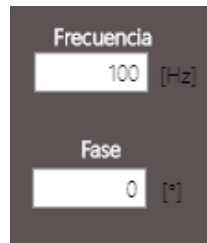


Figura 5.35: Valor de Frecuencia

6. Cambiar la forma y frecuencia al valor que se desee y visualizar los resultados en el osciloscopio las veces que se requieran.

5.4.2. Modo de comandos

1. Conectar el sistema iiDAQ a la computadora.
2. Conectar la salida del generador de funciones a la entrada de nuestro osciloscopio, tal como se muestra en la Figura 5.36.

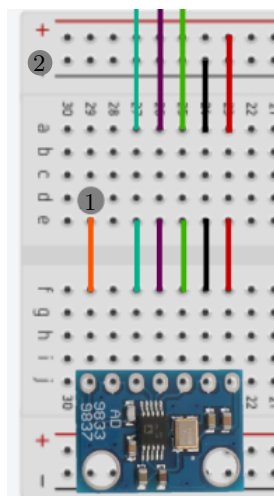


Figura 5.36: Circuito para análisis de Práctica 4. En este no se aplicará ningún circuito externo, ya que será la medida directa del módulo generador de señales. Las conexiones del sistema serán las siguientes: (1)Punta positiva del canal 1 del osciloscopio, (2)Referencia a tierra del canal 1 del osciloscopio.

3. En el script ya configurado, colocar las siguientes líneas de código.

```

1  %Universidad Nacional Autonoma de Mexico
2  %Instituto de Ingenieria
3  %Sistema de adquisicion de datos iiDAQ
4
5  clear all
6  close all
7  clc
8
9  %-----
10
11 channel = 0;
12 waveForm = "";
13 Frequency = 0;
14
15 %-----
16
17 s = connectiiDAQ();
18
19 %-----
20
21 %Aseguramos una configuraci n inicial
22
23
24
25 signalGenerator (channel, "Sine", Frequency);
26 disp('Iniciando programa...');
27 pause (2);
28 clc
29
30 while i == 1
31
32     channel = input("Elegir el canal de salida: ");
33     waveForm = input ("Selecciona la forma de onda: ");
34     Frequency = input("Escribir el valor de frecuencia en la onda: ");
35
36     signalGenerator(s, channel, waveForm, Frequency); %Se configura el
37         generador de se ales para obtener el resultado en el osciloscopio.
38
39     i = input("Para intentarlo de nuevo, escribir 1, en caso contrario,
40         escribir 0: ");
41
42     if i == 0
43         break;
44     end
45
46 end
47
48 %-----
49
50 disconnectiiDAQ(s);

```

4. Una vez listo el programa, dar clic en *EDITOR>Run* (Figura 5.37).

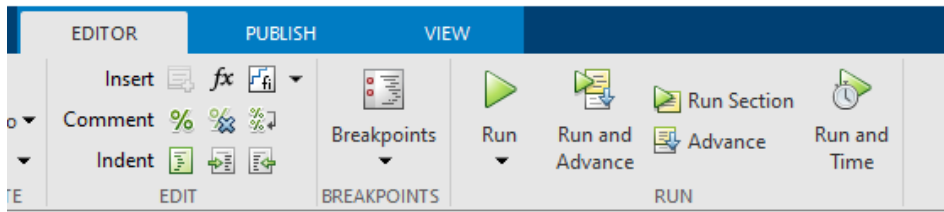


Figura 5.37: Botón de ejecución para la prueba del *script* en el sistema iiDAQ.

- Seguir las instrucciones en la ventana de comandos y reiniciar el programa las veces necesarias, con el objetivo de obtener diferentes señales en el osciloscopio.

5.4.3. Resultados esperados

En el caso de la *interfaz gráfica*, podemos obtener diferentes resultados en el osciloscopio. Con la frecuencia de 100 [Hz] indicada en los primeros pasos, los resultados obtenidos se muestran en las Figuras 5.38, 5.39 y 5.40.

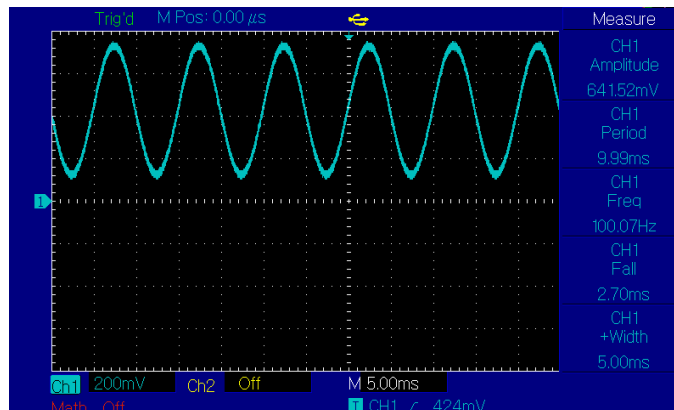


Figura 5.38: Prueba 1. Generación de onda sinusoidal con frecuencia de 100 [Hz].

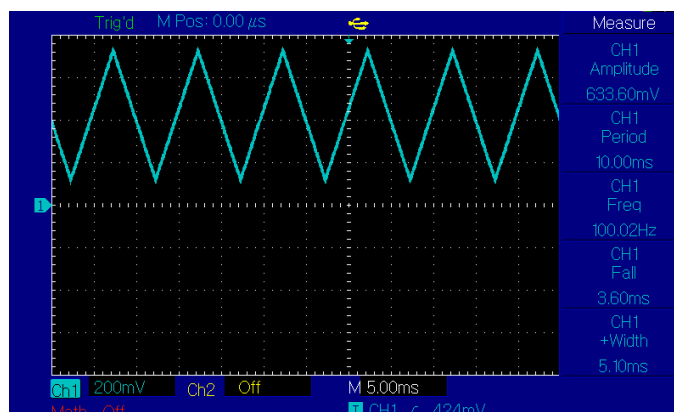


Figura 5.39: Prueba 2. Generación de onda triangular con frecuencia de 100[Hz].

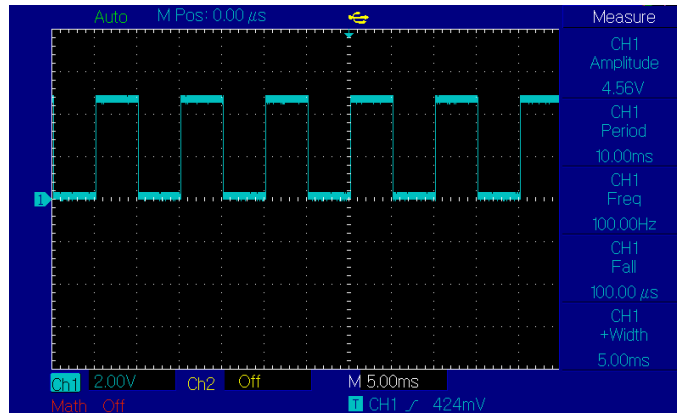


Figura 5.40: Prueba 3. Generación de cuadrada con frecuencia de 100[Hz].

En el caso del *modo de comandos*, los mensajes que visualizaremos en la ventana de comando serán los siguientes:

```
Command Window
Dispositivo conectado
Iniciando programa...
```

Figura 5.41: Configuración de los módulos para inicio de programa.

```
Command Window
Elegir el canal de salida: 1
Selecciona la forma de onda: "Square"
Escribir el valor de frecuencia en la onda: 10000
Escribir el valor de fase en la onda: 0
Para intentarlo de nuevo, escribir 1, en caso contrario, escribir 0: 0
Dispositivo Desconectado
```

Figura 5.42: Configuración de la forma de onda y frecuencia por parte del usuario. Se puede repetir el proceso las veces que sean necesarias.

Los resultados que se pueden obtener en el osciloscopio son variados, teniendo como ejemplos las Figuras 5.43 y 5.44.

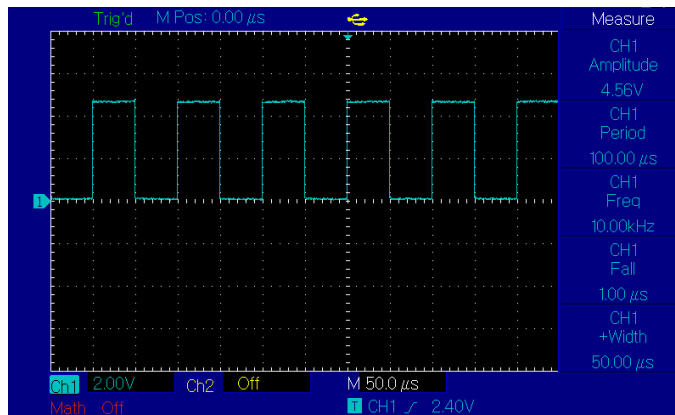


Figura 5.43: Prueba 4. Generación de onda cuadrada con frecuencia de 10 [kHz].

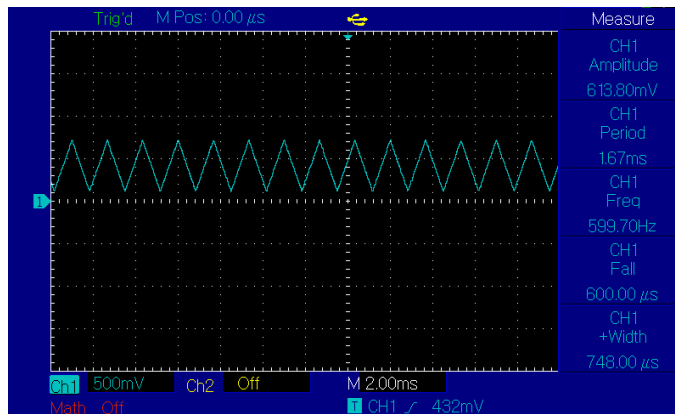


Figura 5.44: Prueba 5. Generación de onda triangular con frecuencia de 600 [Hz].

Apéndice A

Costos de dispositivos en el mercado actual

Para una mejor referencia, y por tratarse de módulos donde los alumnos y profesores se están familiarizando, se presenta en la tabla A.1 un aproximado de los costos que se manejan en el mercado nacional para cada uno de los módulos.

Tabla A.1: Costos actuales de los módulos requeridos para el sistema.

Dispositivo	Costo (MXN)
<i>Arduino Nano 3.0</i>	\$160
<i>ADS1115</i>	\$115
<i>MCP4725</i>	\$70
<i>AD9833</i>	\$215
<i>DD1718PA</i>	\$57
<i>Shield para Arduino Nano</i>	\$55
Total esperado	\$672

Apéndice B

Driver para Arduino Nano 3.0

Las tarjetas de desarrollo de Arduino tienen integrado un chip que sirve como interfaz de UART a USB, es decir, se podrá comunicar con la computadora por medio del cable USB. El modelo de chip varía según el fabricante de cada tarjeta; las versiones originales de Arduino por lo general tiene el chip *Ft232RL*, sin embargo, las versiones réplicas suelen tener el chip *CH340*.

Por esta razón, buscando que el sistema de adquisición de datos sea económico, es probable que la tarjeta de desarrollo tenga integrado el chip CH340, por lo tanto, se presentan los pasos para la instalación del driver correspondiente, logrando que el sistema *iiDAQ* pueda funcionar correctamente.

1. Abrir la carpeta del proyecto *iiDAQ* y dar doble clic en el archivo con extensión *.exe* llamado *DriverCH340*
2. Aparecerá la ventana para el requerimiento de permisos. Dar clic en *Sí* para permitir a la aplicación que realice los cambios necesarios en la computadora.
3. La siguiente ventana será la mostrada en la Figura B.1. Dar clic en el botón *Install* para comenzar la instalación del driver.

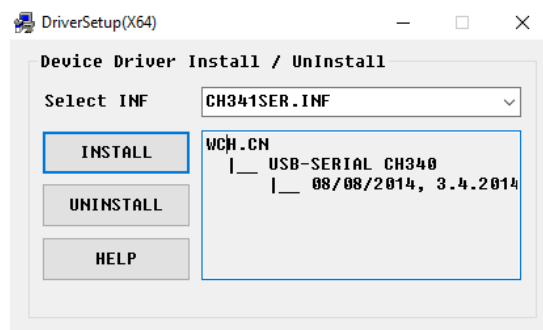


Figura B.1: Instalación de driver.

4. Si la instalación se llevó a cabo correctamente, la ventana que aparecerá será la de la Figura B.2. Los controladores se actualizarán automáticamente cuando se conecte la tarjeta de desarrollo Arduino Nano.

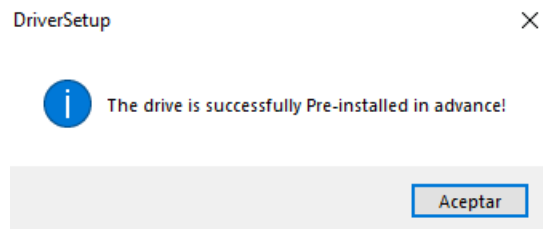


Figura B.2: Instalación correcta

Si los controladores no se actualizan automáticamente, se deben seguir los siguientes pasos:

1. Abrir el administrador de dispositivos en la computadora.
2. Abrir la pestaña *Puertos (COM & LTP)* y dar clic derecho en *USB-SERIAL CH340* (Figura B.3).

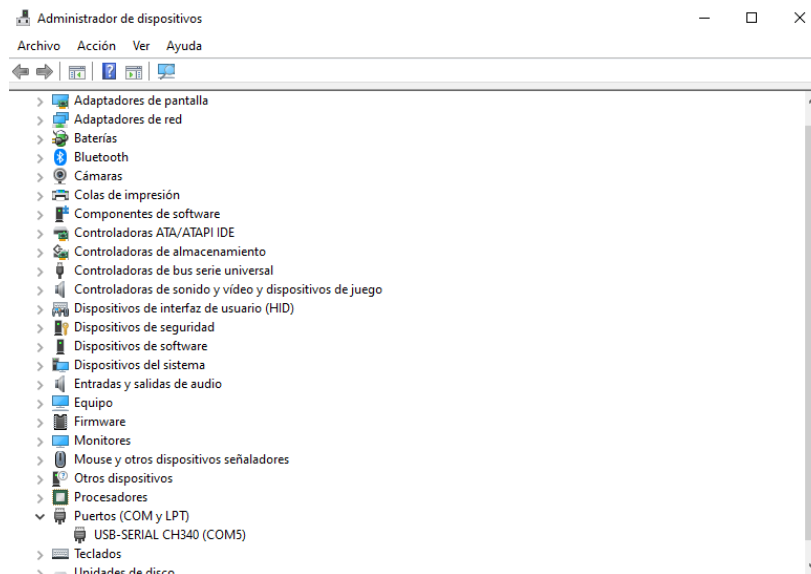


Figura B.3: Puertos COM y LPT

3. A continuación, dar clic en *Actualizar controlador* (Figura B.4).

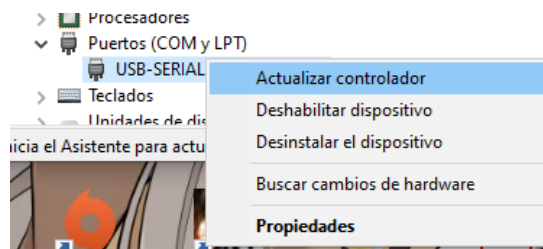


Figura B.4: Actualizar controlador

4. Aparecerá la ventana de la Figura B.5. Dar clic en la opción *Buscar controladores en mi equipo*.

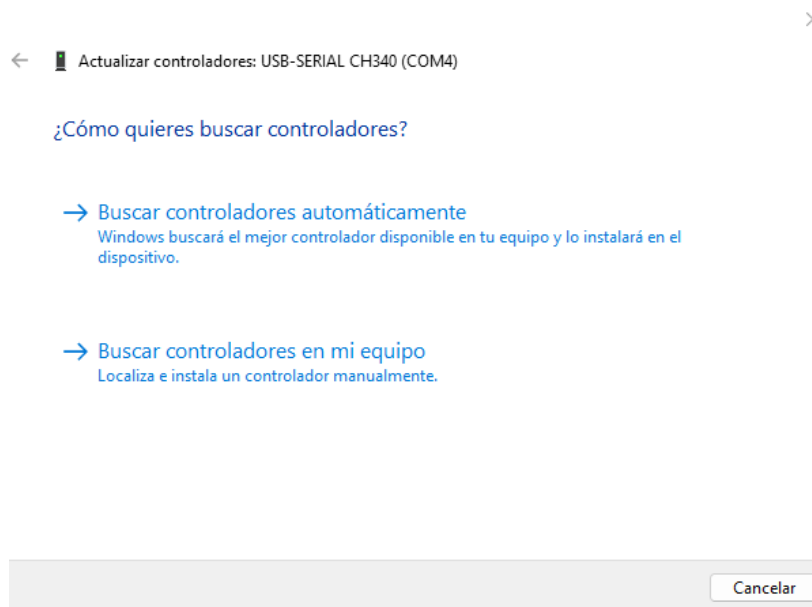


Figura B.5: Opciones de búsqueda para el controlador de Arduino Nano.

5. Aparecerá la ventana de la Figura B.6. Dar clic en el botón *Examinar...* para la búsqueda del controlador.

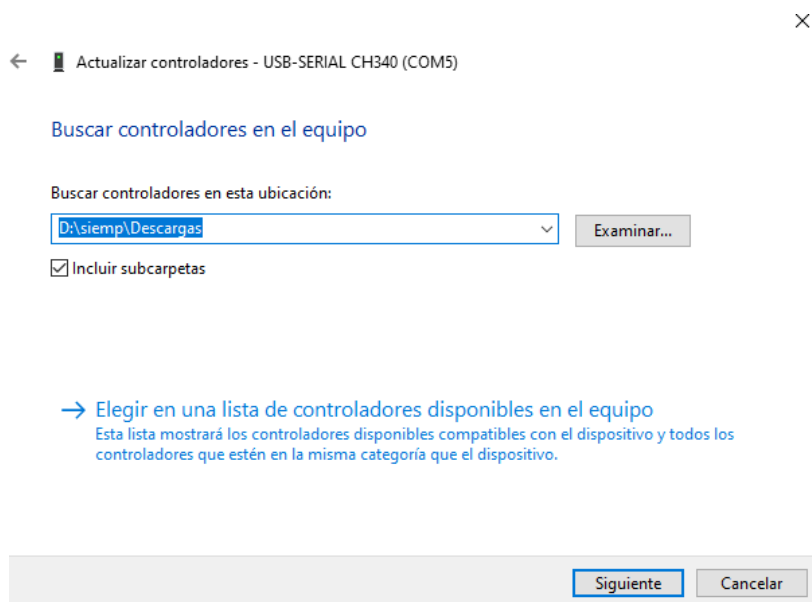


Figura B.6: Examinar computadora para localización de controlador.

6. Seleccionar la carpeta del proyecto *iiDAQ* (Figura B.7). Posteriormente, dar clic en *Aceptar*.

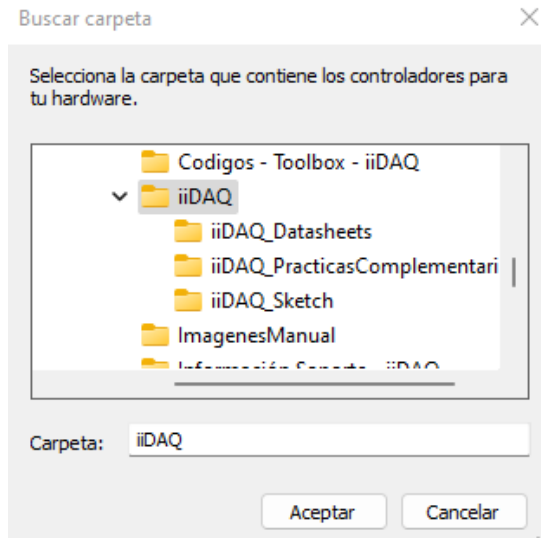


Figura B.7: Búsqueda y selección de carpeta *iiDAQ*.

7. Finalmente, el sistema regresará a la ventana de la Figura B.6. Dar clic en el botón *Siguiente* y aparecerá una ventana de espera (Figura B.8).

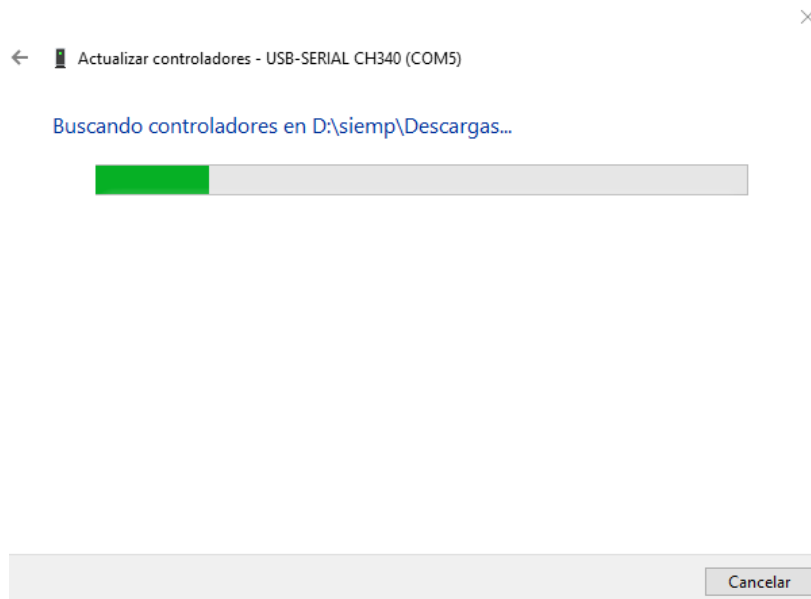


Figura B.8: Búsqueda de controlador en carpeta y actualización. En este paso, los controladores se estarán actualizando para poder utilizar la tarjeta de desarrollo.

8. Si la actualización se realizó correctamente, la ventana final será como la Figura B.9.

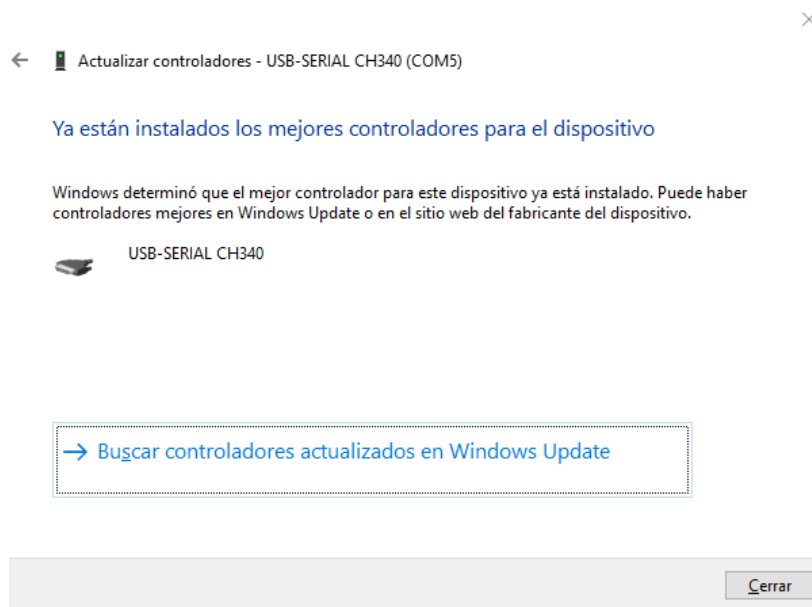


Figura B.9: Actualización de controlador correctamente instalada.

Apéndice C

Consideraciones importantes para el sistema iiDAQ

El sistema iiDAQ se encuentra en su etapa de prueba y retroalimentación por parte de los usuarios, siendo posible que se presenten algunos errores o problemas técnicos menores en la ejecución de la interfaz y funciones del sistema de adquisición de datos. Para esto, se presentan las siguientes recomendaciones a la par del desarrollo de la versión final de iiDAQ, la cual podrá preveer cualquier tipo de error a futuro.

1. Es importante que se sigan los rangos de valores determinados en este tutorial, ya que de no hacerlo, se pueden provocar errores de compilación en los scripts.
2. Los botones de *Registro de datos* (ADC), *Fase* y *Canal* (Generador de señales) siguen en etapa de desarrollo, por lo que aún podrían presentar errores en ciertos casos de estudio.
3. Se recomienda no desconectar el dispositivo en medio del proceso, esto para no generar problemas en el funcionamiento del sistema.
4. Asegurate siempre de cerrar el puerto COM cuando la compilación y ejecución del *script* en MATLAB™ falle.
5. Se está trabajando para que el sistema considere todos los errores posibles y un mayor número de aplicaciones con optimización, por lo tanto, es de gran apoyo que los usuarios compartan su experiencia para lograr la mejora de este modelo educativo orientado en sistemas de instrumentación y control. El sistema iiDAQ es para el beneficio de profesores y estudiantes en el desarrollo de su profesión, tu opinión es importante.

Para dudas, quejas y comentarios, enviar un correo electrónico a la dirección

iidaq.soporte@gmail.com

En breve, tu solicitud será atendida, brindando la mejor asesoría para mejorar el sistema de adquisición de datos iiDAQ.