



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Máster en Ingeniería Industrial**

# **Comunicación con osciloscopio desde Labview.**

**Autor:**

**Sergio González Sánchez**

**Tutor:**

**Dr. Andrés Melgar Bachiller**

**Dpto. de Ingeniería Energética y Fluidomecánica**

**Valladolid, julio de 2018**



## Resumen /Abstract

Durante estos años, el estudio de bombas de combustión suponía un proceso laborioso que constaba de una fase de obtención de datos a través de una tarjeta de adquisición de datos y una fase de diagnóstico independiente. En este Trabajo de Fin de Máster (TFM en adelante) se elaborará un programa en LabView que permitirá adquirir los datos de un osciloscopio conectándolo a través del puerto USB del ordenador y analizarlos dentro del mismo programa haciendo de estos experimentos unas tareas más simples y que conlleven menos tiempo.

*During these years, the study of combustion pumps supposed a laborious process that consist in a data aobtainment phase through a data acquisition card and an independent diagnosis phase. In this final project (TFM) a program in LabView that allow to acquire the data of the oscilloscope and treated connecting toward the USB port and analyte them will be created. This try to save time in the execution and make the experiments easier tasks.*

## Palabras clave/ Keywords

LabView, Bomba, Osciloscopio, Diagnóstico.

*Labview, Bomb, Oscilloscope, Diagnosis.*



## INDICE

1. INTRODUCCIÓN.....	1
1.1. JUSTIFICACIÓN DEL ESTUDIO Y OBJETIVOS.....	1
1.2. UBICACIÓN Y ANTECEDENTES.....	2
1.3. EL OSCILOSCOPIO YOKOGAWA DL708E.....	3
1.3.1. LA INTERCONEXIÓN DE LOS EQUIPOS.....	7
1.4. EL SOFTWARE LABVIEW.....	8
1.4.1. PANEL FRONTAL.....	9
1.4.2. DIAGRAMA DE BLOQUES.....	10
1.4.3. PALETA DE CONTROLES.....	10
1.4.4. PALETA DE HERRAMIENTAS.....	11
1.4.5. PALETA DE FUNCIONES.....	12
1.4.6. EJECUCIÓN DEL PROGRAMA.....	13
2. PREPARACIÓN DE LOS ELEMENTOS.....	14
2.1. ADAPTADOR DEL PUERTO.....	14
2.2. CONFIGURACIÓN DEL PUERTO USB.....	15
3. ELABORACIÓN DEL SOFTWARE.....	18
3.1. ESTRUCTURACIÓN DEL PROGRAMA.....	18
3.2. CAPTACIÓN DE DATOS.....	19
3.2.1. SUBRUTINA “TRAER DATOS”.....	21
3.2.2. SUBRUTINA “INICIA PUERTO”.....	28
3.2.3. SUBRUTINA “CONFIGURA PUERTO”.....	29
3.2.4. SUBRUTINA “CONVERTIR DATOS”.....	30

3.3. GENERACIÓN DE FICHEROS .....	31
3.4. EJECUCIÓN DEL DIAGNÓSTICO.....	35
3.4.1. ARCHIVO .RES GENERADO .....	37
3.5. REPRESENTACIÓN GRÁFICA DE LOS RESULTADOS.....	37
3.5.1. SUBROUTINA “GRAFICAS” .....	38
4. SIMULACIÓN DEL PROCESO .....	43
5. CONCLUSIONES.....	47
6. LINEAS DE FUTURO .....	48
BIBLIOGRAFÍA .....	51

## 1. INTRODUCCIÓN

### 1.1. JUSTIFICACIÓN DEL ESTUDIO Y OBJETIVOS

Automatizar lo máximo posible la captación de datos, diagnóstico y representación de resultados en los ensayos de combustión de bombas de volumen constante puede suponer un ahorro de tiempo y recursos si dicho proceso se hace forma correcta.

Con la ayuda de softwares como LabView y elementos como el osciloscopio Yokogawa DL708E se llevará a cabo la captación de las presiones de la bomba durante su ciclo de combustión, se realizará el diagnóstico y se mostrarán las gráficas de los resultados obtenidos.

Los objetivos principales de este TFM son:

- Establecer una conexión fiable entre un osciloscopio Yokogawa DL708E y un ordenador que permita la transmisión de datos entre ellos.
- Originar un programa de LabView capaz de comunicarse con un osciloscopio Yokogawa DL708E para adquirir de este los datos recogidos por un captador piezoeléctrico introducido en una bomba de combustión a volumen constante.
- Transformar los datos facilitados por el osciloscopio en los valores de tensión equivalentes.
- Transformar los anteriores datos a los valores de presión que les corresponden.
- Generar un fichero con los datos de presión recogidos y otro con datos auxiliares del experimento que permitirán tener un histórico de los experimentos realizados, así como el posterior diagnóstico de los mismos.
- Realizar un diagnóstico de los datos obtenidos.
- Graficar los distintos datos del análisis de tal forma que se puedan interpretar por el técnico de laboratorio. Se representarán magnitudes que guarden relación dentro de la misma gráfica para que dicha interpretación sea más sencilla. Se mostrarán por pantalla los resultados obtenidos.

- Integrar dentro de un mismo programa todos los pasos anteriores.

## 1.2. UBICACIÓN Y ANTECEDENTES

En el presente Trabajo de Fin de Máster (TFM en adelante) se ha llevado a cabo en el laboratorio de motores térmicos de la Escuela de Ingenieros Industriales de la Universidad de Valladolid.

Anteriormente, en 2003, se realizó un proyecto de fin de carrera efectuado por Álvaro Lafuente Lafuente en el que se estudió el comportamiento de esta misma bomba de volumen constante. (1) La intención es realizar este mismo estudio a través de una nueva metodología y aparamenta posible debido a los grandes avances de la tecnología en estos 15 años.

Para la realización de este TFM en adelante se han necesitado los siguientes elementos:

- Bomba de volumen constante a analizar. Proporcionada por la universidad e instalada en el laboratorio antes descrito.
- Elementos auxiliares de la instalación: Válvulas para la entrada y salida de gases y sus displays. Gracias a ellos se podrá obtener datos que necesitaremos para nuestro programa. Dispuestos en el laboratorio.
- Captador piezoeléctrico: Será el encargado de captar las presiones del interior de la bomba y transformarlo en la tensión eléctrica correspondiente.
- Osciloscopio Yokogawa DL708E: Mostrará los datos aportados por el captador en su display. Además, será el elemento que conectaremos al ordenador y del cual cogeremos los datos. Su correcta configuración será vital para el éxito de este experimento
- Ordenador: Deberá ser lo suficientemente potente para poder soportar el software LabView. Durante este TFM se ha utilizado el ordenador del laboratorio para la generación y testeo del programa.

En la siguiente imagen se puede la disposición de los distintos elementos en el laboratorio a excepción de la bomba que estaba en reparación en el momento de la foto. (Fig.1.2.1)





Fig. 1.2.1. Laboratorio de Motores Térmicos de la Universidad de Valladolid (Elaboración propia)

### 1.3. EL OSCILOSCOPIO YOKOGAWA DL708E.

Es el corazón del experimento y el éxito radica en el buen funcionamiento de este.

Dispone de ocho canales para la captación de distintos datos y una configuración individual de cada canal. Dichas configuraciones pueden ser el rango y el offset de las mediciones.

Algunas configuraciones son específicas de cada uno de los canales mientras que otras pueden configurarse de forma global para todo el equipo según las necesidades del resto de aparataje o del experimento como los baudios de transferencia o los distintos bits de paro de los datos.

En las siguientes figuras se ven esquemas del osciloscopio utilizado donde se puede observar la complejidad del mismo. El gran número de teclas suponen una gran maniobrabilidad y flexibilidad por parte del usuario de la configuración del dispositivo (Fig. 1.3.1 y 1.3.2)

• Front panel

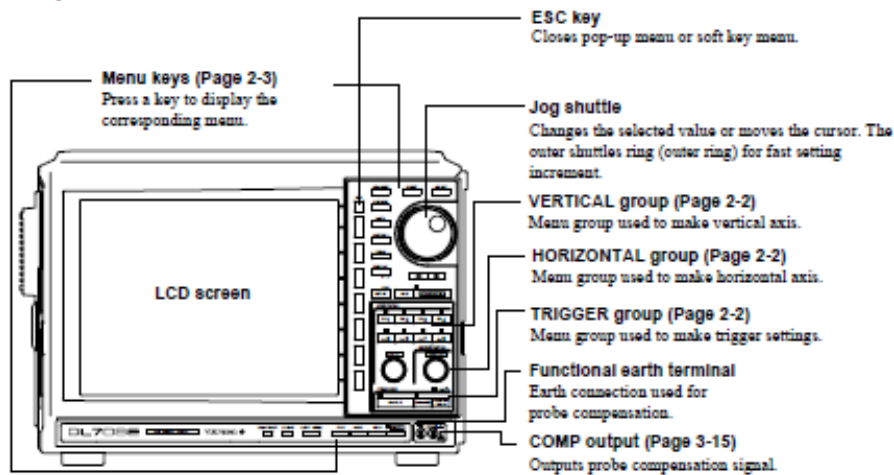


Fig. 1.3.1. Esquema de la vista frontal del osciloscopio Yokogawa DL708E (2)

• Side panel

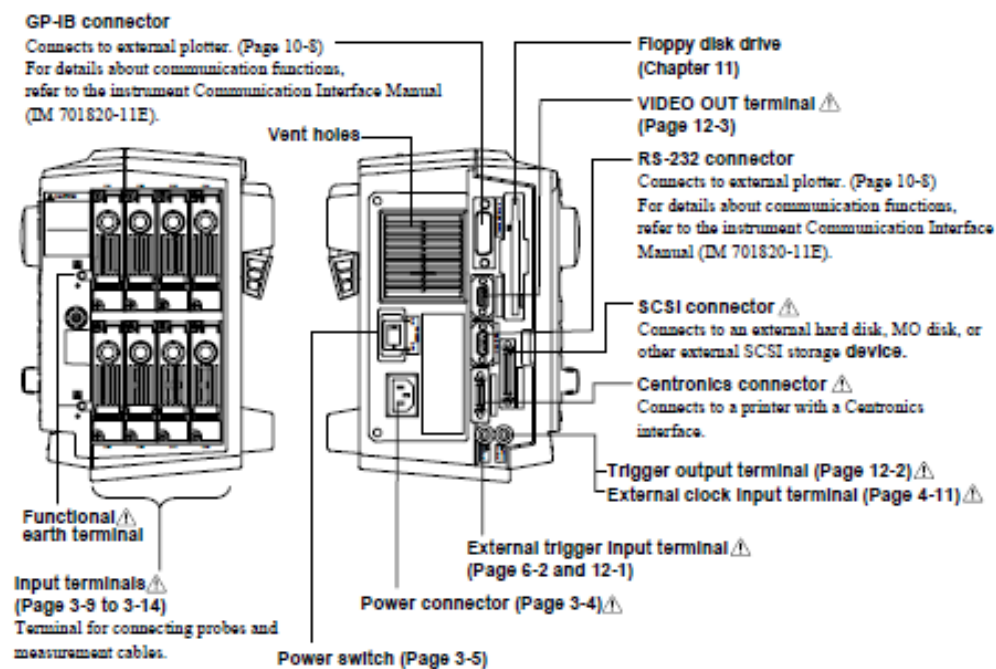


Fig. 1.3.2. Vistas laterales del osciloscopio Yokogawa DL708E (2)

De este dispositivo se explicarán las teclas y acciones necesarias para que el experimento funcione correctamente.

Antes de comenzar la experimentación hay que destacar que para la correcta ejecución de los experimentos se deben configurar el osciloscopio, el puerto USB y el programa del ordenador con los mismos parámetros. A continuación, se indicará como configurar el primero de estos elementos y más adelante se explicará cómo configurar los otros dos.

Para ello se accionará el menú MISC (Fig. 1.3.3) el cual muestra la configuración de la transmisión del puerto, del sistema y las acciones relativas al instante de disparo. Dicha tecla se encuentra en la parte inferior del panel frontal del osciloscopio.

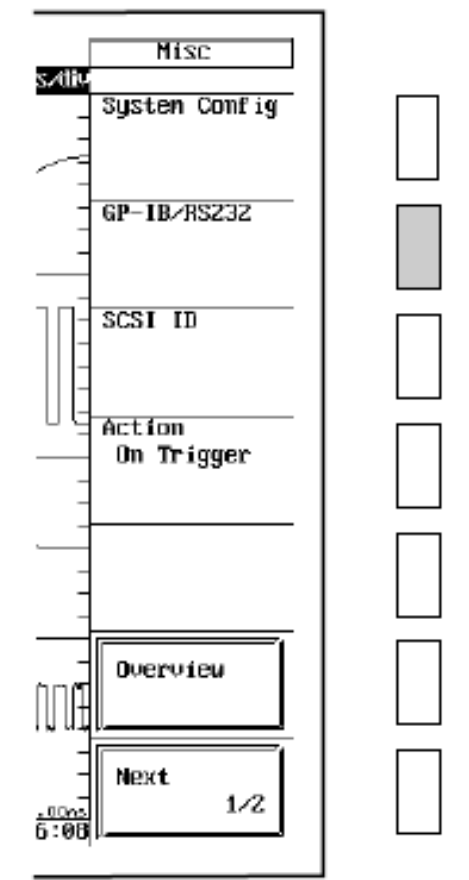


Fig. 1.3.3. Detalle menú MISC (2)

Dentro de este menú se selecciona la opción *GP-IB/R232* (como se muestra en la Fig. 1.3.3) ya que es la que nos permite acceder a la configuración del puerto. Una vez dentro se muestran los distintos parámetros para asignar el tipo de puerto y su configuración. (Fig. 1.3.4)

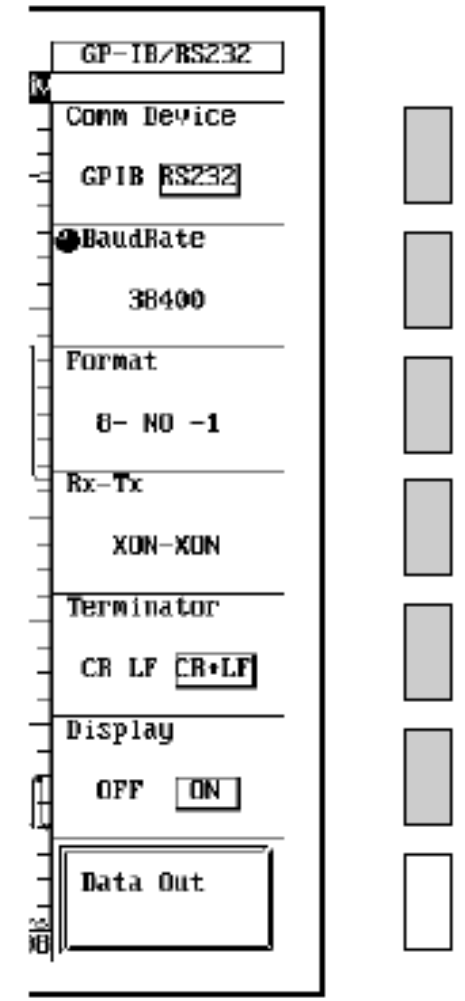


Fig. 1.3.4. Detalle menú GP-IB/R232 (2)

Los parámetros asignados serán los siguientes:

- Comm Device: Ofrece las opciones de GP-IB y R232 en función del puerto que se quiera configurar. Se seleccionará la opción R232 ya que es el tipo de puerto a través del cual se conecta el osciloscopio al ordenador.
- Baud rate: Establece en baudios la velocidad de transmisión entre los dispositivos. Se ofrecen las opciones de 1200, 2400, 4800, 9600, 19200. Para este proyecto se asignará una tasa de baudios de 19200.

- Format: Establece la combinación de longitud de los datos y los bits de parada. De las opciones ofrecidas se escogerá 8-NO-1 lo que significa que el tamaño del paquete de datos será 8 bits y el número de bits de parada será 1.
- Rt-Tx: Selecciona los métodos de control para la recepción y transmisión de datos. Se escogerá la opción NO-NO que no establece ningún método de control suponiendo que los buffers son suficientemente grandes para los paquetes de datos transmitidos.
- Terminator: Establece el método a través del cual el osciloscopio emitirá el final de sus datos. Las opciones son LF (Line Feed o en ASCII “\n”), CR (Carriage Return o en ASCII “\r”) o una combinación de ambas. Para este proyecto se usará la opción LF.

Con esto se tendría la configuración del osciloscopio acorde a lo solicitado en el experimento.

### 1.3.1. LA INTERCONEXIÓN DE LOS EQUIPOS

Como se observa en las vistas laterales del dispositivo (Fig. 1.3.2) existen varias entradas y salidas del osciloscopio a través de los cuales se puede intercomunicarlo con otros dispositivos. En este experimento en particular será necesario conectar dos dispositivos a nuestro osciloscopio.

El primero de ellos es el captador piezoeléctrico encargado de recoger la presión del interior de la bomba. Se puede conectar a cualquiera de los ocho canales de los que dispone el osciloscopio. Se escogerá en puerto 1 para la conexión de este periférico.

A través del conector R232 del dispositivo se puede conectar un elemento que recoja los resultados una vez que estos han pasado por el osciloscopio. En este proyecto se usará este puerto para conectar nuestro ordenador.

Más adelante (apartado 2) se explicará cómo se ha llevado a cabo la interconexión del ordenador con el dispositivo ya que, en este caso, dicha conexión no es tan simple como usar un cable de puerto serie.

De cualquier manera, se recomienda realizar estas uniones con el osciloscopio apagado evitando así posibles daños tanto en el equipo como en el personal.

## 1.4. EL SOFTWARE LABVIEW.

LabView constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos. Las ventajas que proporciona el empleo de LabView se resumen en las siguientes:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

LabView es un entorno de programación destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo comerciales que utilizan el lenguaje C o BASIC. Sin embargo, LabView se diferencia de dichos programas en un importante aspecto: los citados lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabView emplea la programación gráfica o lenguaje G para crear programas basados en diagramas de bloques.

Para el empleo de LabView no se requiere gran experiencia en programación, ya que se emplean iconos, términos e ideas familiares a científicos e ingenieros, y se apoya sobre símbolos gráficos en lugar de lenguaje escrito para construir las aplicaciones. Por ello resulta mucho más intuitivo que el resto de los lenguajes de programación convencionales.

LabView posee extensas librerías de funciones y subrutinas. Además de las funciones básicas de todo lenguaje de programación, LabView incluye librerías

específicas para la adquisición de datos, control de instrumentación VXI, GPIB y comunicación serie, análisis presentación y guardado de datos.

LabView también proporciona potentes herramientas que facilitan la depuración de los programas. (3)

Los elementos principales de LabView son los que se muestran a continuación.

#### 1.4.1. PANEL FRONTAL

Se trata de la interfaz gráfica del VI con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc.

Cada uno de ellos puede estar definido como un control o un indicador. Los primeros sirven para introducir parámetros al VI, mientras que los indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación. (3)

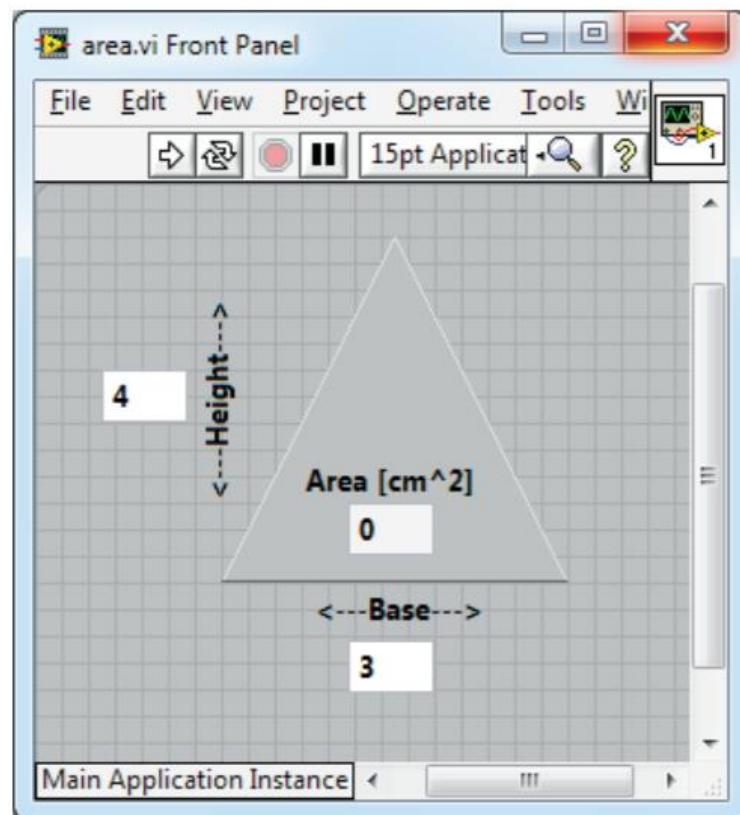


Fig. 1.4.1.1. Ejemplo de panel frontal de LabView (Elaboración propia)

### 1.4.2. DIAGRAMA DE BLOQUES

El diagrama de bloques constituye el código fuente del VI. En el diagrama de bloques es donde se realiza la implementación del programa del VI para controlar o realizar cualquier procesado de las entradas y salidas que se crearon en el panel frontal.

El diagrama de bloques incluye funciones y estructuras integradas en las librerías que incorpora LabView. En el lenguaje G las funciones y las estructuras son nodos elementales. Son análogas a los operadores o librerías de funciones de los lenguajes convencionales.

Los controles e indicadores que se colocaron previamente en el panel frontal se materializan en el diagrama de bloques mediante los terminales.

El diagrama de bloques se construye conectando los distintos objetos entre sí, como si de un circuito se tratara. Los cables unen terminales de entrada y salida con los objetos correspondientes, y por ellos fluyen los datos.

Los cables son las trayectorias que siguen los datos desde su origen hasta su destino, ya sea una función, una estructura, un terminal, etc. Cada cable tiene un color o un estilo diferente, lo que diferencia unos tipos de datos de otros. (3)  
(Fig. 1.4.2.1)

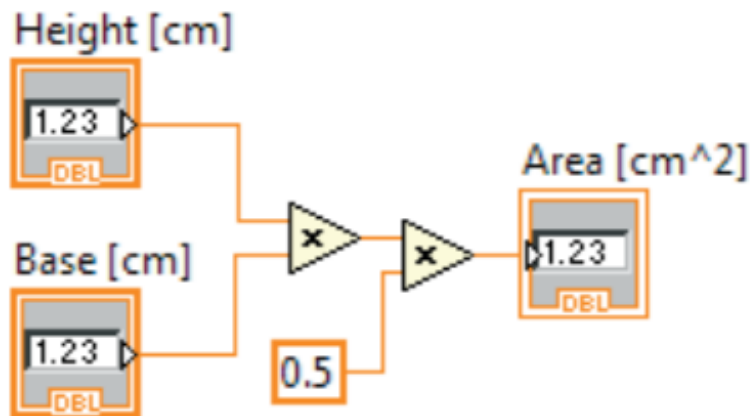


Fig.1.4.2.1. Ejemplo de diagrama de bloques de LabView (Elaboración propia)

### 1.4.3. PALETA DE CONTROLES

Se utiliza solo en el panel frontal y contiene todos los indicadores y controles a utilizar dentro de nuestro programa. (Fig. 1.4.3.1)



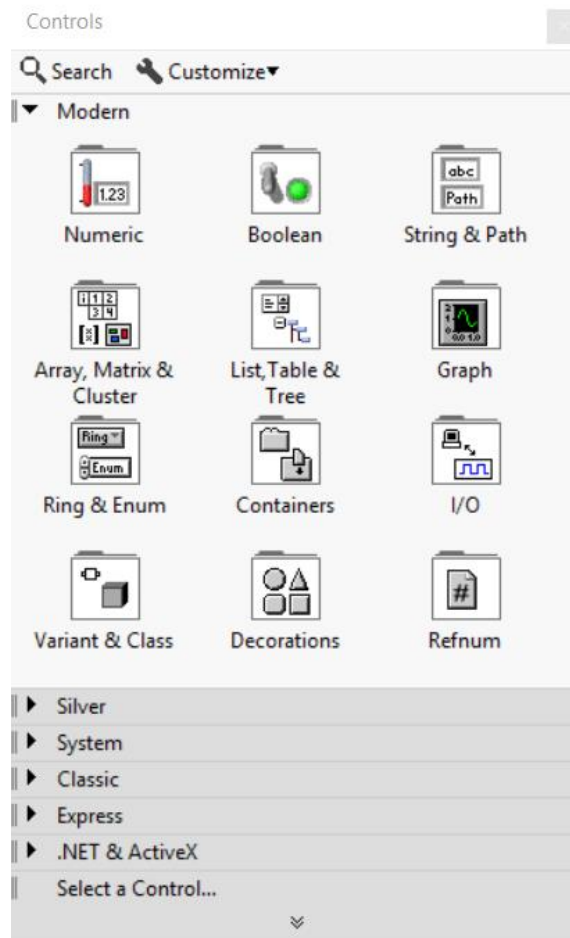


Fig. 1.4.3.1. Paleta de controles de LabView (Elaboración propia)

Se puede acceder a esta paleta desde el panel frontal siguiendo la siguiente ruta: *View* → *Controls Palette*.

Cualquiera de los menús de esta paleta puede desplegarse según sean necesarios unas variables u otras.

#### 1.4.4. PALETA DE HERRAMIENTAS

Se emplea tanto en el panel frontal como en el diagrama de bloques. Contiene las herramientas necesarias para editar y depurar los objetos tanto del panel frontal como del diagrama de bloques. (3) (Fig. 1.4.4.1)

Se destaca la herramienta *Probe* ya que ha sido utilizada para encontrar y subsanar los errores en el código.



Fig. 1.4.4.1. Paleta de herramientas de LabView (Elaboración propia)

Se puede acceder a esta paleta desde ambas pantallas siguiendo la siguiente ruta: *View* → *Tools Palette*.

### 1.4.5. PALETA DE FUNCIONES

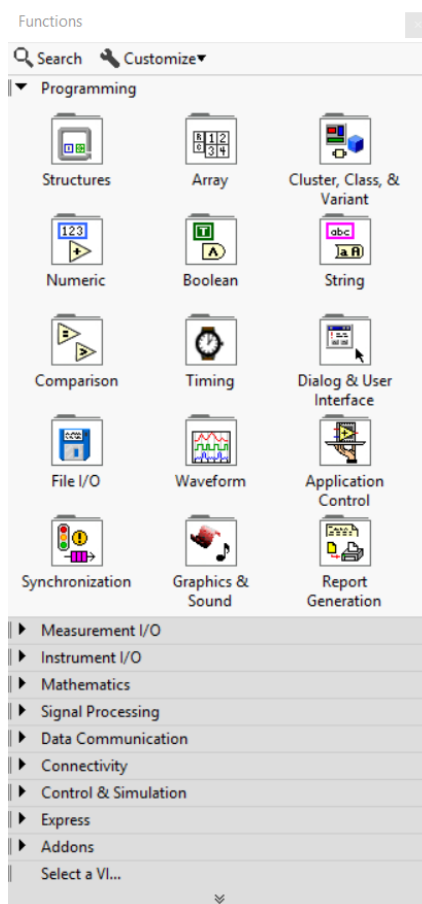


Fig. 1.4.5.1. Paleta de funciones de LabView (Elaboración propia)

Se emplea en el diseño del diagrama de bloques. La paleta de funciones contiene todos los objetos que se emplean en la implementación del programa del VI, ya sean funciones aritméticas, de entrada/salida de señales, entrada/salida de datos a fichero, adquisición de señales, temporización de la ejecución del programa, etc. (3) (Fig. 1.4.5.1)

Solo se puede acceder desde el diagrama de bloques de LabView siguiendo la ruta *View* → *Functions Palette*.

Para seleccionar una función o estructura concretas, se debe desplegar el menú *Functions* y elegir entre las opciones que aparecen.

#### 1.4.6. EJECUCIÓN DEL PROGRAMA

Una vez finalizada la programación se debe ir al panel frontal ya que es el que mostrará los resultados del experimento.

Desde ahí, se destaca la barra de herramientas. (Fig. 1.4.6.1)

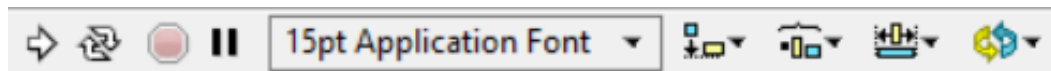


Fig. 1.4.6.1. Barra de herramientas encargada del arranque y parada del programa (Elaboración propia)

El primero de los botones es el botón *Run* e iniciará el programa. Para que el programa se inicie de forma continua pulsaremos el segundo botón (*Run Continuously*).

Una vez iniciado el programa el botón de *Stop* (el tercer botón desde la izquierda) se iluminará permitiendo de ese modo la finalización de este.

En cualquier momento podría pulsarse el botón *Pausa* que pausaría la ejecución permitiendo la continuación del mismo desde el punto donde se paró.

## 2. PREPARACIÓN DE LOS ELEMENTOS

Antes de ejecutar el programa de LabView se deben preparar los dispositivos para su correcta comunicación.

Los preparativos necesarios que llevar a cabo son la adaptación de puerto serie a USB del cable de conexión y la configuración del puerto USB del ordenador.

### 2.1. ADAPTADOR DEL PUERTO

El problema reside en que el cable de interconexión entre el osciloscopio y nuestro ordenador tiene dos entradas de puerto serie y nuestro ordenador no dispone de uno.

Se ha tenido que disponer de un convertidor de puerto serie a USB como el de la siguiente imagen. (Fig. 3.1.1)



Fig. 3.1.1. Adaptador puerto serie a USB (4)

Será necesaria la instalación de los drivers de este dispositivo. Si el dispositivo no se instala automáticamente la primera vez que se utiliza en un ordenador su configuración debe hacerse de forma manual.

Para esta configuración se descargará el driver correspondiente y en el administrador de dispositivos de nuestro ordenador se buscará el convertidor serie-USB y se instalará el driver descargado.

De esta forma quedaría configurado el convertidor y el ordenador podrá conectarse con el osciloscopio sin ningún problema

## 2.2. CONFIGURACIÓN DEL PUERTO USB

El objetivo de este paso consiste en configurar el puerto al que se conecta el osciloscopio de la misma forma que después se hará en nuestro programa de LabView y previamente hemos hecho en el osciloscopio a fin de no cometer errores con la lectura de datos.

Para ello se utilizará un software llamado *NI MAX*, un software desarrollado por National Instruments, empresa que desarrolla Labview.

Iniciamos el software y nos encontramos la pantalla de inicio. (Fig. 3.2.1)

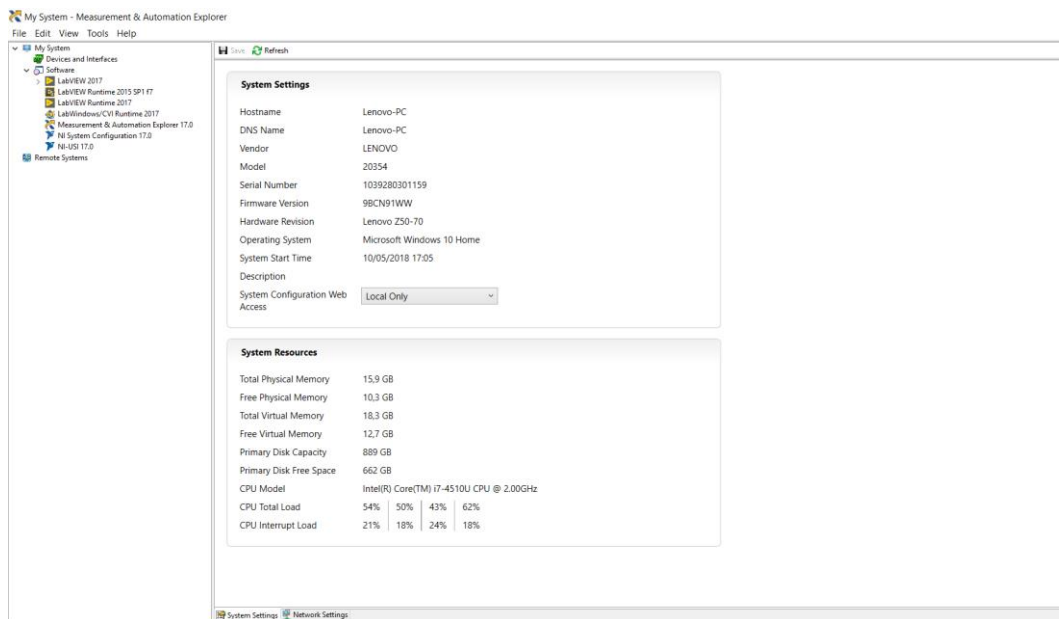


Fig. 3.2.1. Pantalla de inicio de NI MAX (Elaboración propia)

Cuando se tenga conectado el osciloscopio a través de un puerto USB del ordenador este aparecerá en el apartado “*Devices and Interfaces*” dentro del menú “*Configuración*”.

Seleccionando el dispositivo aparecerá una pantalla como la siguiente (Fig.3.2.2) en la que aparecerá por defecto el nombre del puerto al que está conectado. En este caso el osciloscopio está conectado al puerto 3 por lo que el nombre por defecto es “COM3”. No se realizarán cambios en este aspecto, pero se deberá recordar el nombre que se da al puerto para después configurarlo con el mismo nombre en el programa. (5)

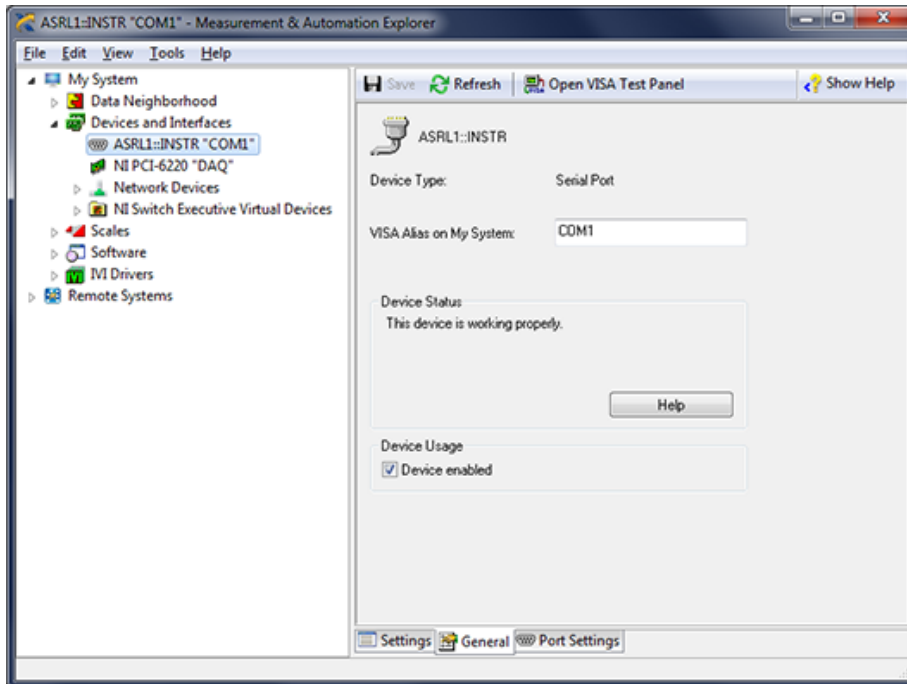


Fig. 3.2.2. Ventana de configuración del nombre del puerto USB (5)

Haciendo clic en la pestaña *Port Settings* (en la parte inferior de la columna central) se podrá confirmar que la razón de transferencia, tamaño del paquete, bits de paro y bits de paridad son compatibles con el instrumento usado en su aplicación. (5) (Fig.3.2.3)

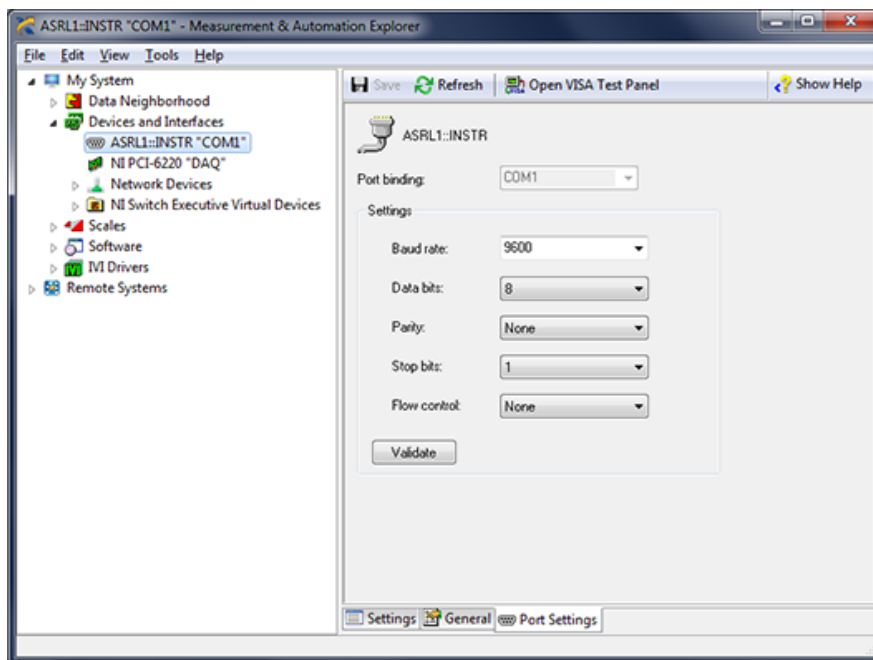


Fig. 3.2.3. Ventana de configuración de las características del puerto USB (5)

Para que la comunicación sea la correcta los valores deben ser los mismos que en la configuración del osciloscopio y la posterior configuración de LabView.

- Razón de transferencia: 19600
- Tamaño del paquete: 8
- Bits de paro: 1
- Paridad: Ninguna
- Control del flujo: Ninguno.

Tras validar estos apartados se tendrá el puerto configurado para la correcta transmisión de datos.

### 3. ELABORACIÓN DEL SOFTWARE

#### 3.1. ESTRUCTURACIÓN DEL PROGRAMA

El software generado en este TFM realizará cuatro tareas durante el diagnóstico de la bomba de volumen constante:

1. Captación de los datos adquiridos por el osciloscopio.
2. Generación de los ficheros para el análisis.
3. Diagnóstico de los datos.
4. Presentación de las gráficas resultantes del diagnóstico.

El programa de LabView está generado sobre un programa principal llamado *Diagnóstico* del que cuelgan varias subrutinas que siguen el siguiente esquema de funcionamiento. (Fig.4.1.1)

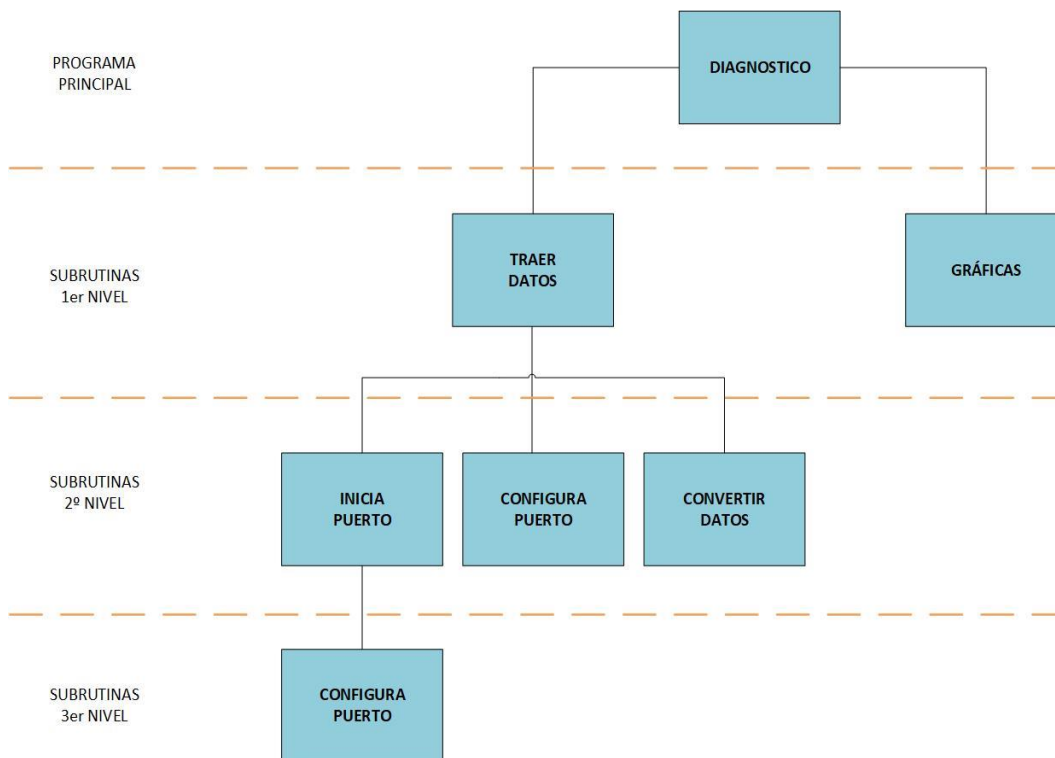


Fig. 4.1.1. Esquema de funcionamiento del programa de LabView. (Elaboración propia)

Asimismo, cabe destacar que el osciloscopio Yokogawa DL708E posee una serie de librerías que se puede descargar de la web para ayudarnos en la formación del programa.



Gracias a los manuales de uso del osciloscopio (6) (7) se pueden ver los comandos con los que se puede establecer comunicación con el osciloscopio y así adquirir los datos que el posee o enviarle comandos o preguntas desde el ordenador.

Para enviar ordenes o preguntar al osciloscopio se utilizará la herramienta VISA WRITE FUNCTION (Fig. 4.1.2) mientras que para recibir sus respuestas se usará VISA READ FUNCTION (Fig. 4.1.3).

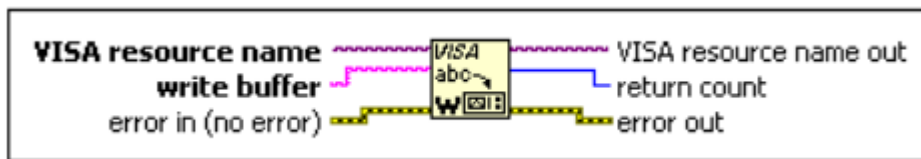


Fig. 4.1.2. VISA Write function. (8)

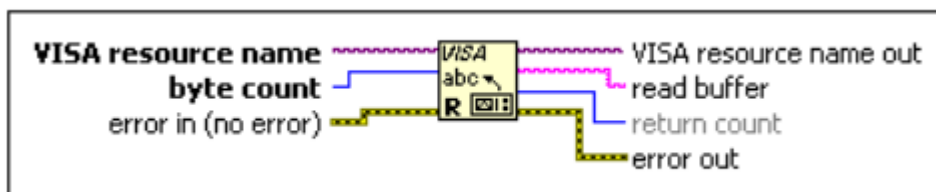


Fig. 4.1.3. VISA Read function. (8)

### 3.2. CAPTACIÓN DE DATOS

En este paso lo que se hará es establecer una comunicación con el osciloscopio para así poder adquirir los datos que previamente ha obtenido de la bomba de presión constante. Posteriormente se tratarán para su representación.

Será en este paso el único momento donde será necesario establecer conexión con el osciloscopio y por tanto en el único momento donde se necesitarán los comandos de intercomunicación propios del mismo.

Para iniciar la adquisición y preparación de los datos se deben introducir los siguientes parámetros:

- **Calibración:** Informa de la relación existente entre la diferencia de presión existente en el interior de la bomba y la diferencia de potencial que emite el captador piezoeléctrico y que adquiere nuestro osciloscopio. Permitirá entonces obtener los valores de presión reales en el interior de la bomba a través de los resultados adquiridos. Este

valor de calibración es de **2 bar/V** y se insertará a través de una variable con formato LabView DBL correspondiente a una variable en coma flotante de doble precisión que posee 64 bits de almacenamiento y puede dar una precisión máxima de 15 cifras decimales. (9)

- **Canal:** Como se vio anteriormente, el osciloscopio DL708E posee 8 canales para obtener información. Con este dato informaremos al programa de LabView de cuál será el canal por el que se recibirán los datos del osciloscopio. El canal escogido para este experimento de prueba será el **Canal 1**. Se introducirá a Labview a través de una variable U16 correspondiente a un número entero de 16 bits y sin signo. (9)

Este segundo parámetro será la única variable de entrada para nuestra subrutina *TRAER DATOS* cuyas salidas son las siguientes variables:

- **Datos Voltios:** Se trata de un vector con todos los valores de tensión que hemos traído del osciloscopio. Los datos están en formato DBL. (9)
- **Position Trigger:** Se trata de una variable que indica la diferencia de posición que hay entre el inicio del registro y la referencia temporal del experimento
- **Sampling rate:** Es la frecuencia de muestreo con la que el osciloscopio está captando los datos del interior de la bomba.

Una vez adquirido el vector *Datos Voltios* lo multiplicamos por el dato de calibración antes determinado para así obtener un vector de las presiones.

Por otro lado, se calcula la inversa del *Sampling rate* lo que proporcionará el periodo asociado a la frecuencia de muestreo. Con este valor se puede hallar el tiempo hasta instante de referencia del muestreo multiplicándolo por el valor de *Position Trigger*.

Se crea un registro o “Cluster” que es como se conoce en LabView y se le asigna primero un 0, después el valor del periodo de muestreo y por último todos los valores de presión obtenidos.

Por último, se genera una representación gráfica de este registro para que así muestre los valores de presión obtenidos frente al tiempo

En la Fig. 4.2.1 se puede ver como es el resultado final de la parte del programa principal *Diagnostico* encargada de captar los datos.

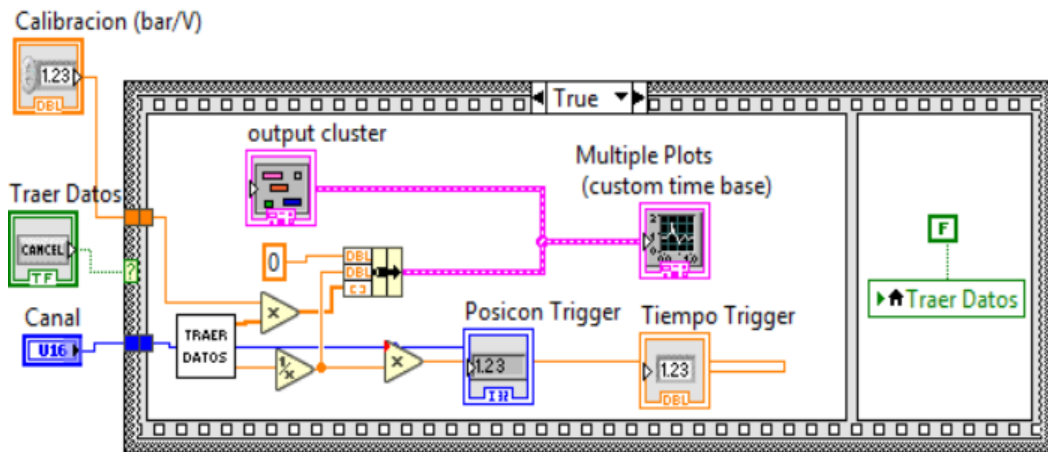


Fig. 4.2.1. Fracción del programa *Diagnostico* encargada de la captación de datos. (Elaboración propia)

### 3.2.1. SUBROUTINA “TRAER DATOS”

Esta subrutina se encargará de coger los datos de tensión del osciloscopio. Para ello, se llevarán a cabo una serie de tareas que son las siguientes:

➤ **Iniciación del puerto:**

Esta parte de la rutina se encargará de iniciar y configurar el puerto a través del cual se recibirán los datos. La llevará a cabo la subrutina *INICIA PUERTO*. Además, iniciará una sesión VISA, acción necesaria para poder establecer comunicación con el osciloscopio a través del puerto.

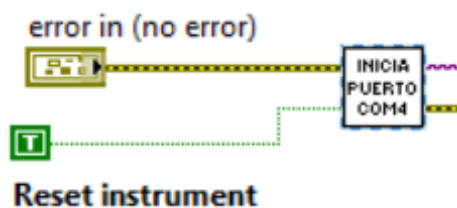


Fig. 4.2.1.1. Iniciación del puerto (Elaboración propia)

➤ **Selección del canal del osciloscopio.**

Una vez iniciado el puerto se debe configurar el canal a través del cual el osciloscopio está recibiendo los datos que nos interesan.

Para ello se usará el comando “WAV:TRAC\s” (6) y se le añadirá un listado con los canales disponibles. Por último, introduciremos una variable para que nos

permite seleccionar el canal. Se recuerda que en este experimento se usará el canal 1 del osciloscopio y que se podría asignar sin necesidad de crear el listado, pero haciéndolo de este modo se confiere al programa de una mayor flexibilidad. (Fig. 4.2.1.2)

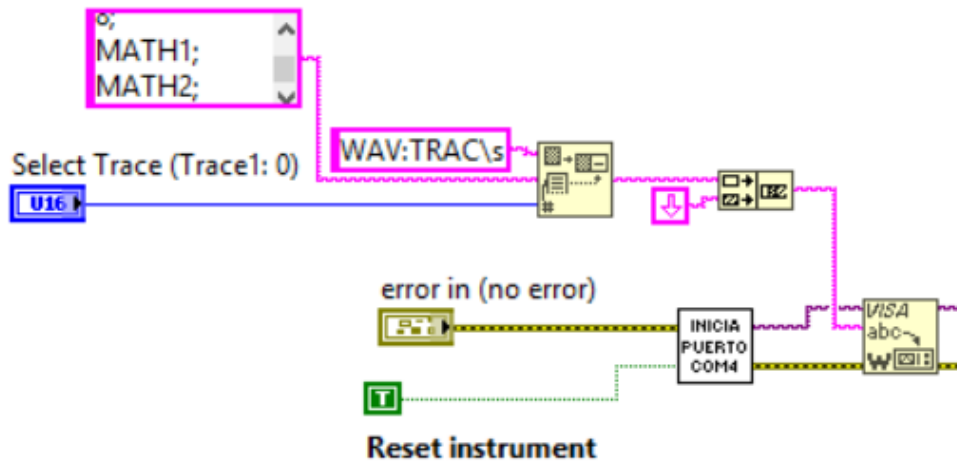


Fig. 4.2.1.2. Selección del canal del osciloscopio. (Elaboración propia)

➤ **Determinación del número de datos enviados.**

Con el comando “WAV:LENG?” (6) se le preguntará al osciloscopio cuál es el número de datos que va a enviar. A través del VISA Read obtendremos la respuesta del osciloscopio a través de una cadena de caracteres.

Gracias a DECIMAL STRING TO NUMBER FUNCTION se puede pasar esa cadena de caracteres a un número entero largo con signo (formato I32 de LabView) y así poderlo usar en el futuro. (Fig. 4.2.1.3)

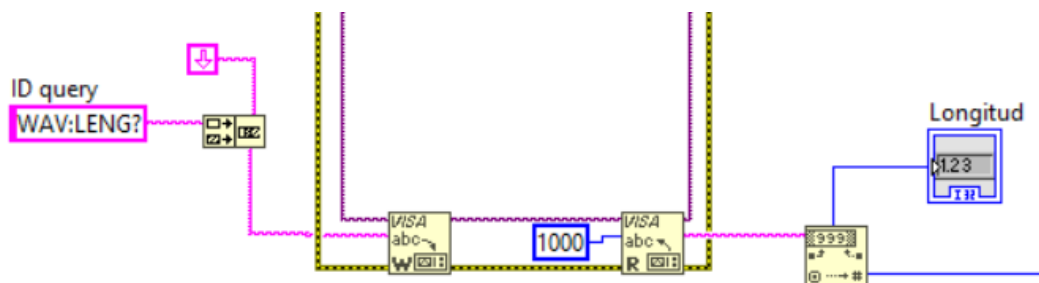


Fig. 4.2.1.3. Determinación del número de datos (Elaboración propia)

➤ **Determinación de la frecuencia de muestreo.**

Para la determinación de la frecuencia de muestreo se usa el comando “WAV:SRATE?”. (6)

El valor obtenido se pasará a formato DBL de LabView y como ya hemos dicho anteriormente será una de las salidas del sistema. (Fig. 4.2.1.4)

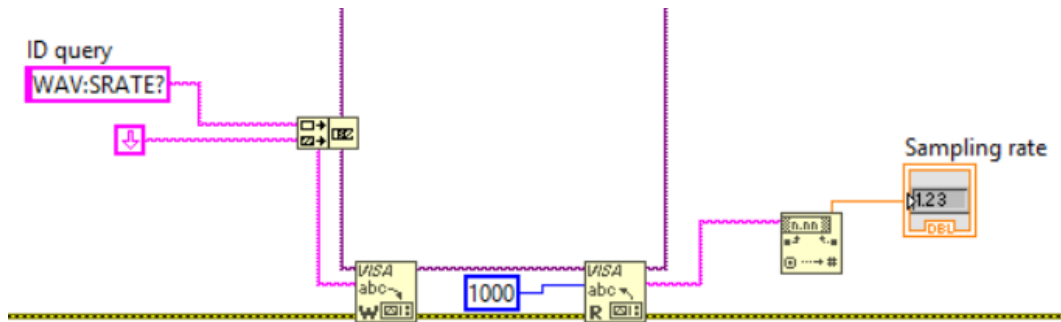


Fig. 4.2.1.4. Determinación de la frecuencia de muestreo (Elaboración propia)

➤ **Determinación de la posición del instante de disparo.**

Se trata de determinar la posición del instante de disparo con el comando “WAV:TRIG?”. (6).

La cadena de caracteres resultante deberá convertirse en un número entero que determina la posición del byte que indica el inicio de envío de datos por parte del osciloscopio. (Fig. 4.2.1.5)

La posición del instante de disparo será una de las salidas del sistema como se dijo anteriormente.

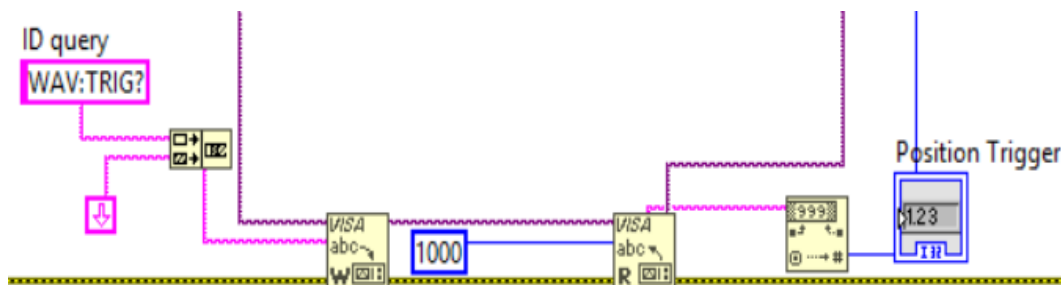


Fig. 4.2.1.5. Determinación de la posición del instante de disparo (Elaboración propia)

➤ **Configuración del tipo de datos a recibir por el ordenador.**

Este apartado es fundamental ya que una incorrecta programación del mismo supondría una captación de los datos que nada tiene que ver con la realidad. (Fig. 4.2.1.6)

Los datos que deben establecerse son los siguientes:

- Formato de los datos intercambiados: Se establecerán con formato palabra (Word) por lo que cada dato constará de dos bytes de información. El comando usado será “WAV:FORM\sWORD” (6)
- Orden de los bytes: Al estar formados por dos bytes, los datos podrían reportar valores totalmente distintos en función del orden en el que esos bytes sean leídos. Se configurará de tal forma que el byte más significativo sea leído primero (MSBF). Para ello a la expresión anterior se le añade “;BYT\sMSBF” (6)
- Número de la grabación: Este parámetro sirve para hacer varias mediciones dentro del mismo archivo. Por defecto nuestra medición será la 0. Con el comando “WAV:REC\s%7d” (6) permitirá asignar este número de grabación y este se introducirá a través de un entero largo (I32 en LabView).
- Inicio de la grabación: Se realiza con el comando “WAV:STAR\s%7d” (6). El valor escogido para el inicio de la grabación será *Position Trigger* ya que de este modo obviaremos los primeros bytes que ha enviado el osciloscopio y no aportan información sobre la medida.
- Fin de la grabación: Se realiza con el comando “WAV:END\s%7d” (6). El valor escogido para el fin de la grabación será *Longitud* ya que indica el último valor que ha de ser recogido.

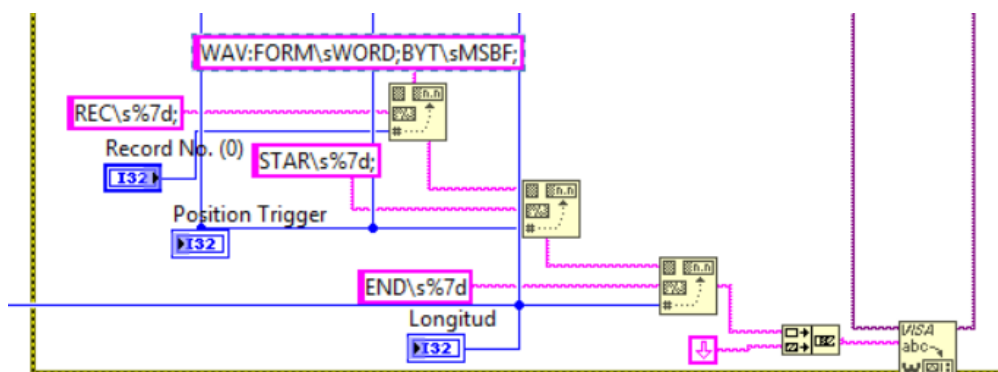


Fig. 4.2.1.6. Configuración de los datos (Elaboración propia)

➤ **Determinación del rango de los datos.**

El rango es un parámetro necesario que conocer del osciloscopio para poder convertir el bloque de datos en valores numéricos. Expone la variación que supone una subdivisión gráfica del osciloscopio.

Con el comando “WAV:RANG?” (6) le preguntamos al osciloscopio este valor y transformamos su respuesta de char (cadena de caracteres) en un número con formato LabView DBL para su posterior uso. (Fig. 4.2.1.7)

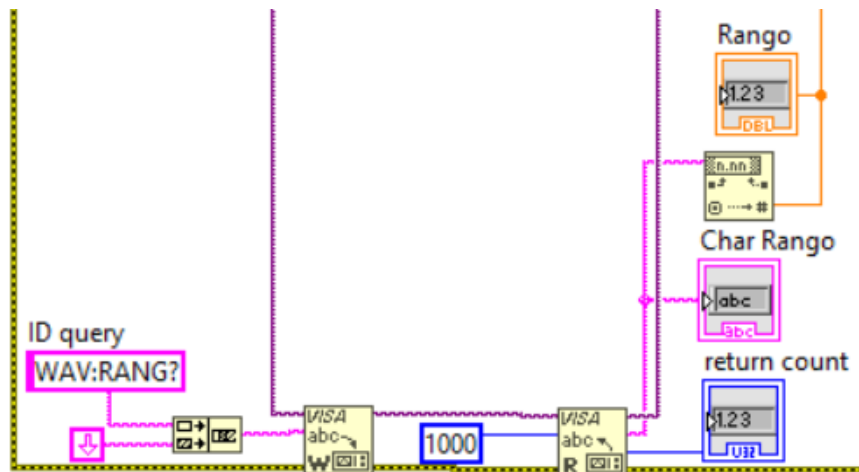


Fig. 4.2.1.7. Determinación del rango (Elaboración propia)

➤ **Determinación del offset de los datos.**

De igual modo que el rango, es un dato fundamental para la transformación de los datos.

Con el comando “WAV:OFFS?” (6) se le pregunta al osciloscopio este valor y se transforma su respuesta de char (cadena de caracteres) en un número con formato LabView DBL para su posterior uso. (Fig. 4.2.1.8)

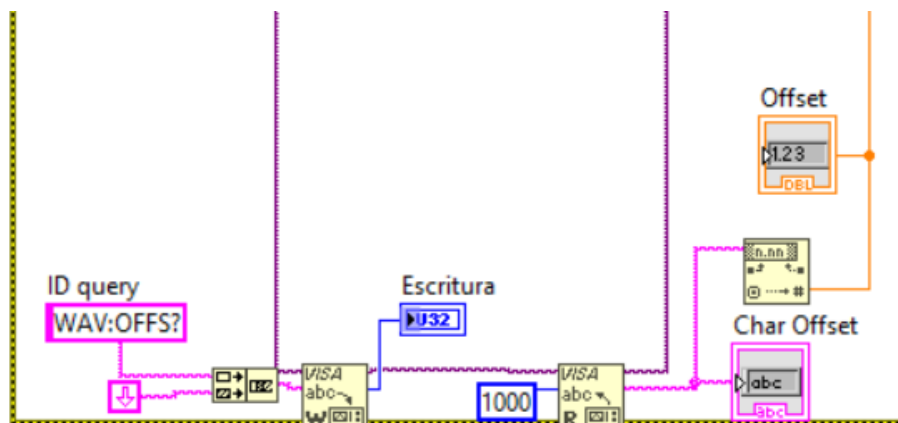


Fig. 4.2.1.8. Determinación del offset (Elaboración propia)

➤ Captación de la cadena de datos enviada por el osciloscopio.

Con el comando de LabView “WAV:SEND?” (6) le se puede preguntar al osciloscopio cual es la cadena de datos que ha obtenido de la bomba. (Fig.4.2.1.9).

Para la lectura necesitamos realizar dos acciones previas:

- Calcular el número de bytes de información que se han de leer: Restando el valor de *Position Trigger* al de *Longitud* se obtiene el número de datos que posee el osciloscopio. Debemos multiplicar dicho resultado por dos debido a que los datos están en formato palabra lo que ocupa dos bytes de información. Asimismo, se le añade un pequeño espacio de 12 bytes con el fin de asegurar que todos los datos sean recogidos.
- Desactivación de la combinación de parada hasta el fin de la lectura: Se decidió llevar esta acción a cabo al encontrarse con el problema de que alguno de los datos o la combinación de ellos proporcionaba la combinación de parada asociada a este programa por lo que varios datos quedaban sin recoger. La desactivación de esta combinación se lleva a cabo gracias a una subrutina llamada *CONFIGURA PUERTO* cuyo funcionamiento se explicará más adelante. Dicha rutina debe ser colocada entre el VISA Write que emite la petición de datos y el VISA Read encargado de recibirlos.

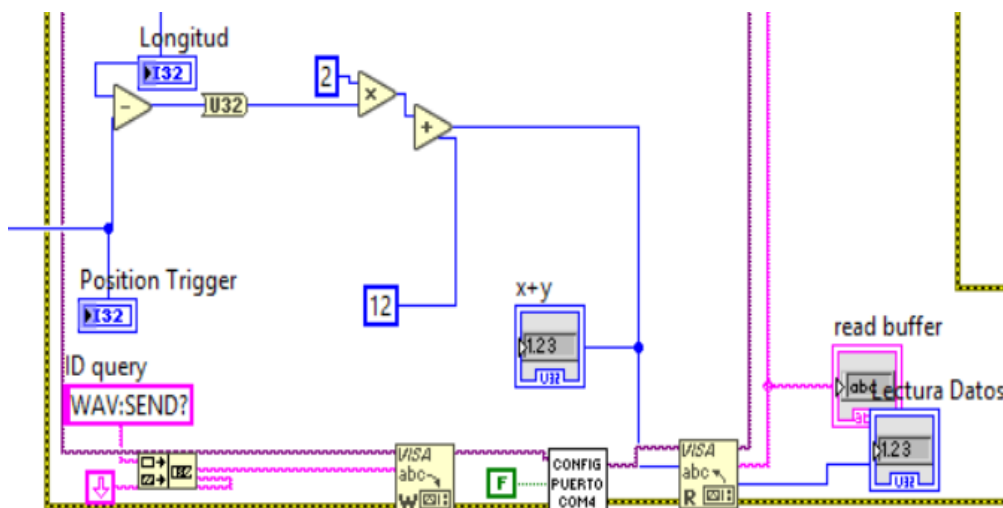


Fig. 4.2.1.9. Captación de la cadena de datos del osciloscopio (Elaboración propia)



➤ **Conversión de la cadena de datos a valores numéricos.**

Gracias a este paso se conseguirá pasar la cadena de caracteres a valores útiles para nuestro proyecto.

Este proceso se llevará a cabo dentro de una subrutina llamada CONVERTIR DATOS. Esta posee 3 entradas que son la cadena de datos, el valor de rango y el valor de offset. Tras su actuación proporciona 3 salidas que son el vector de los datos en voltios en formato DBL, la precisión y el número de byte de dato, aunque estos dos últimos no se usarán. (Fig. 4.2.1.10)

El vector de los datos en voltios será además una de las salidas de la rutina TRAER DATOS al programa principal.

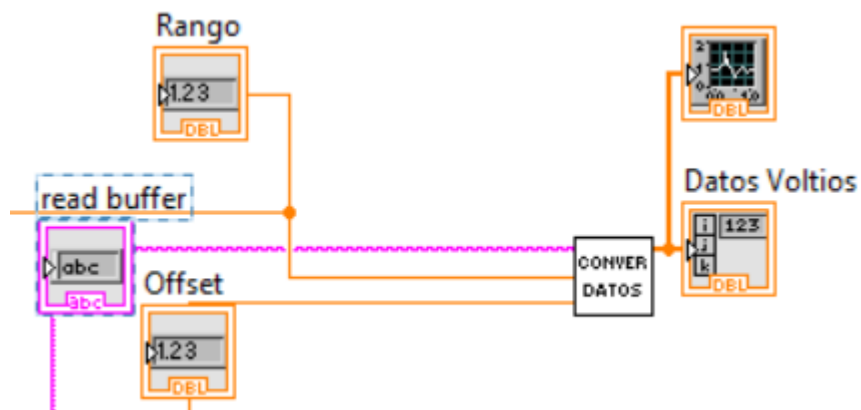


Fig. 4.2.1.10. Conversión de la cadena de datos a valores numéricos (Elaboración propia)

➤ **Cierre de la sesión VISA**

Es importante cerrar la sesión VISA abierta en el primer paso para evitar errores de comunicación en el futuro.

Existe una función llamada VISA CLOSE que se encarga de cerrar la sesión que se le aporte por su entrada. (Fig 4.2.1.11)



Fig. 4.2.1.11. Cierre de la sesión VISA (Elaboración propia)

Ordenando todos los pasos descritos anteriormente quedaría resuelta nuestra subrutina para traer datos cuya apariencia sería como la siguiente imagen. (Fig.4.2.1.12)

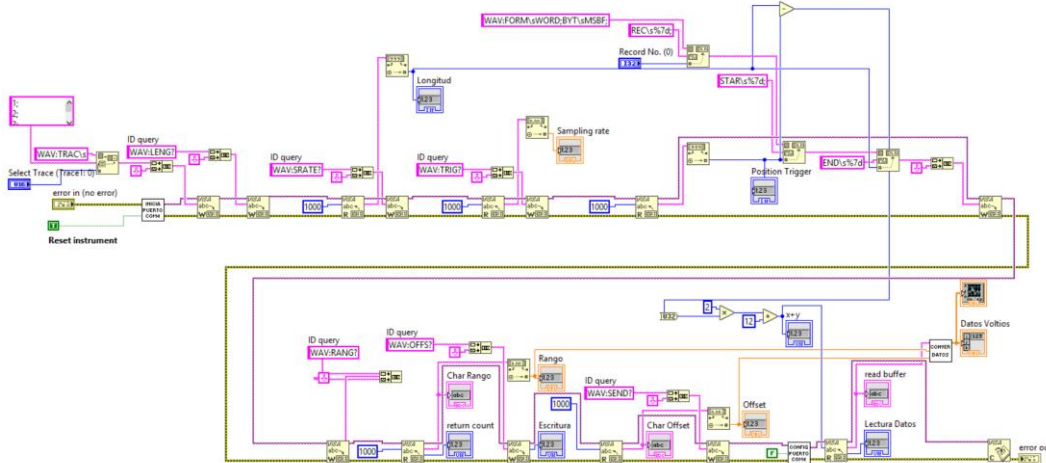


Fig. 4.2.1.12. Esquema completo de la subrutina TRAER DATOS (Elaboración propia)

### 3.2.2. SUBROUTINA “INICIA PUERTO”

Es una subrutina de que cuelga de TRAER DATOS y que se encarga de iniciar una sesión VISA y además de configurar el puerto USB para la correcta transmisión de datos. (Fig. 4.2.2.1)

El inicio de sesión de VISA se lleva a cabo con una función propia de LabView llamada VISA Open Function la cual dándole el nombre que le dimos al puerto USB en su configuración nos abre una sesión VISA para ese recurso.

La configuración del puerto se llevará a cabo a través de la subrutina CONFIGURA PUERTO.

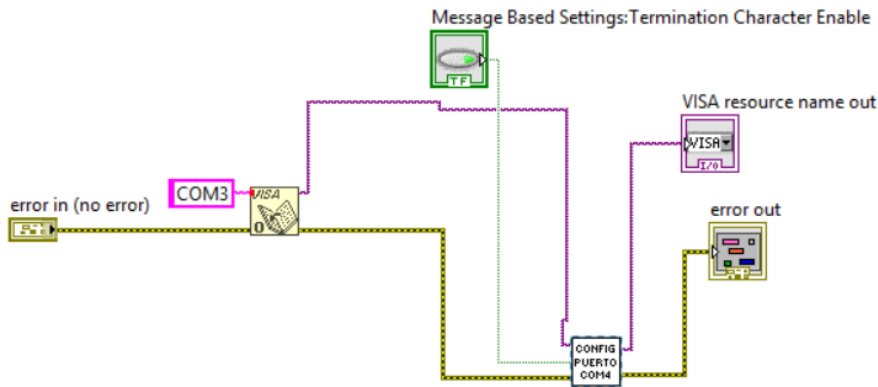


Fig. 4.2.2.1. Esquema completo de la subrutina INICIA PUERTO

### 3.2.3. SUBROUTINA “CONFIGURA PUERTO”

Esta subrutina será utilizada dos veces a lo largo del programa. La primera de ellas en la configuración del puerto que se lleva a cabo en la subrutina INICIA PUERTO y la segunda vez para desactivar el bit de parada en la lectura de datos del osciloscopio perteneciente a la subrutina TRAER DATOS. (Fig. 4.2.3.1)

Dentro de esta subrutina lleva a cabo la configuración de las distintas características del puerto USB a través del cual se transmitirán los datos.

Los valores de las características deberán coincidir con los aportados en la fase de configuración del puerto a través de NI MAX.

La configuración del puerto se hará gracias a la herramienta VISA Configure Serial Port VI. Los valores deberán ser:

- Combinación de paro: Establece la combinación de bits necesaria para terminar la lectura de datos. Escribiendo una A establecemos que la combinación de parada será “\n”. (8)
- Interruptor de habilitación de la combinación de paro: Se controla con una variable booleana (True o False) y con ella se podrá habilitar o no la combinación de parada. (8) Cuando configuremos el puerto está variable habilitará y cuando lo estemos usando en la lectura de datos la deshabilitaremos.
- Velocidad de transmisión: Indica en baudios (bits/s) la velocidad de transmisión de la información a través del puerto USB. (8) El valor debe ser de 19200 baudios.
- Timeout: Indica el tiempo en milisegundos que hay para escribir o leer del puerto. Establecemos 12 segundos (12000 milisegundos).
- Bits de datos: Establece la cantidad de bits que tiene cada dato. Se puede configurar de 5 a 8 bits. (8) Se utilizarán 8 bits.
- Paridad: Indica si hay paridad entre los datos leídos o enviados. (8) Se establecerá que no hay paridad.
- Bits de parada: Establece el número de bits de paro para cada paquete de datos. Puede escogerse 1, 1.5 o 2 bits de parada. (8) Se establecerá 1 bit de paro para este proyecto.

- Control del flujo: Establece el tipo de control asumido para la transmisión de datos. No se asignará ningún control dando por hecho que los buffers son suficientemente grandes como para que acojan todos los datos. (8)

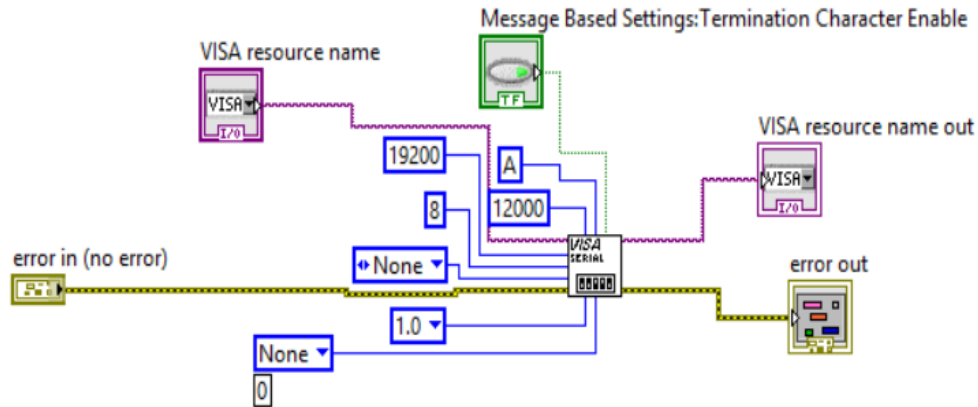


Fig. 4.2.3.1. Esquema completo de la subrutina CONFIGURA PUERTO (Elaboración propia)

### 3.2.4. SUBROUTINA “CONVERTIR DATOS”

Esta subrutina se encargará de convertir la cadena de caracteres resultante de la lectura de datos del osciloscopio a valores numéricos.

Accedemos a esta subrutina desde TRAER DATOS por lo que se considera una subrutina de nivel 2.

Existe una expresión dentro del manual del osciloscopio para convertir los datos que es la siguiente: (6)

$$\text{Voltaje} = \frac{\text{Rango} * \text{Datos} * 8}{\text{División}} + \text{Offset} \quad [1]$$

Donde *DIVISIÓN* es 250 si el formato de los datos es BYTE y de 64000 si es WORD.

Tanto el valor de rango como el de offset son valores obtenidos anteriormente en la subrutina TRAER DATOS.

Introducida la cadena de caracteres en la rutina se selecciona el segundo carácter ya que con él se podrá saber la precisión que hay tras transformarlo en un entero de formato palabra.

También se puede obtener el número de byte de datos escogiendo desde el tercer carácter con una longitud igual a la precisión.

Con ayuda de estos dos datos se irá leyendo el vector de datos y transformando sus valores en número con formato DBL.

Cada valor obtenido de la cadena de caracteres deberá ser convertido en un valor de tensión con la expresión [1].

En la siguiente figura se ve un esquema completo de nuestra rutina CONVERTIR DATOS. (Fig. 4.2.4.1)

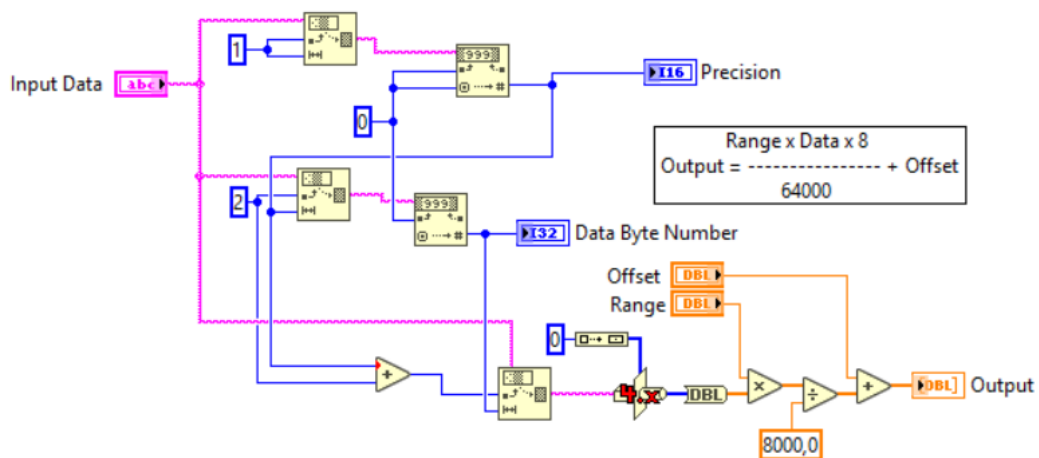


Fig. 4.2.4.1. Esquema completo de la subrutina CONVERTIR DATOS (Elaboración propia)

### 3.3. GENERACIÓN DE FICHEROS

Una vez que los datos son adquiridos y transformados en presión debemos guardarlos en archivos específicos para su diagnóstico.

En este apartado se crearán dos archivos distintos, uno con los datos obtenidos del osciloscopio y tratados para transformarlos en presión y otro con valores auxiliares necesarios para el diagnóstico.

En primer lugar, debemos aportar al programa tanto la ruta del archivo, que debe ser la misma que el software de diagnóstico, como el nombre que le daremos a nuestros archivos.

Aclarar que los dos archivos creados tendrán el mismo nombre, pero distinta extensión. El registro de los datos de presión se guardará en una extensión .prs mientras que los datos auxiliares se aportarán mediante una extensión .cfg.

Una vez aportados la ruta y nombre del fichero se procede a abrir un archivo. La premisa es crear un archivo nuevo o sobrescribir en caso de que ya exista un archivo con el mismo nombre.

Con OPEN/CREATE/REPLACE FILE FUNCTION de LabView se puede llevar a cabo esta acción. Una de las salidas de esta función es la opción de cancelado que en caso de ser False abrirá el archivo. Pasaremos esa respuesta por una puerta NOT para que la respuesta válida sea True.

Realizada la apertura de nuestros archivos debemos preparar los datos que se escribirán en ellos.

Tomando el registro “*Output Cluster*” creado en el apartado anterior se deshace en las tres partes que lo conformaban. De él se toman los valores de presión obtenidos y se reconvertirán en una cadena de caracteres para su escritura en el archivo prs.

Para el otro fichero se debe integrar en un único archivo los datos iniciales sobre la bomba y el experimento, los datos de cálculo necesarios y los datos de composición del gas.

Se incluye en un único vector y por este orden los siguientes parámetros.

- Periodo de muestreo: Obtenido de la disgregación del registro “*Output Cluster*”. Será el periodo con el que se ejecutarán los cálculos y determinará el número de puntos calculados.
- Temperatura inicial: Se trata de la temperatura que hay en el interior de la bomba antes de la chispa. Se utilizará para el cálculo de la presión calculada en el momento del análisis. En futuros experimentos se podría medir la temperatura en todo momento en lugar de calcularla por lo que este dato no sería necesario.
- Presión inicial: Presión antes del inicio del experimento. Al igual que la temperatura se utilizará en la fase de diagnóstico.
- Radio de la esfera en metros: Radio equivalente al que aproximamos nuestra bomba
- Registro de datos de cálculo: Variables matemáticas necesarias para el cálculo.
- Registro de composición: Aportará los datos de la mezcla de gases introducida.

El registro de los datos de cálculo contendrá los datos de la siguiente imagen respetando su orden. (Fig. 4.3.1)

Datos Calculo

Npts_Reg	50
t_max	0,07
Orden_Pol	4
Orden_Pol_ini	3
Der_Nula_ini	1
Npts_Reg_Ini	75

Fig. 4.3.1. Registro de datos de cálculo (Elaboración propia)

El registro de composición de gases incluye las medidas de los gases de la siguiente imagen. (Fig. 4.3.2.)

Composición

N2	0
O2	0
CO2	0
H2O	0
CO	0
H2	0
CH4	0

Fig.4.3.2. Registro de composición. (Elaboración propia)

Una vez obtenido el vector de todos los datos debemos pasarlo de formato DBL a cadena de caracteres.

Para finalizar, se pondrá este vector tras otro que titulará lo que es cada dato. (Fig. 4.3.3)



Fig. 4.3.3. Titulación del vector de datos. (Elaboración propia)

Quedará por tanto una matriz de dos columnas con los datos para los cálculos y sus correspondientes títulos.

De este modo se tendrían preparados los datos para el fichero cfg.



Gracias a la función WRITE DELIMITED SPREADSHEET VI se podrán escribir los archivos prs y cfg correspondientes.

Por último, se rodeará toda esta rutina para que sea activada al pulsar el botón de salvar datos y dentro de ella se hará que dicho botón se apague al terminar la creación de nuestros ficheros.

En la siguiente figura se puede ver un esquema completo de la parte del programa responsable de la generación de ficheros. (Fig. 4.3.4)

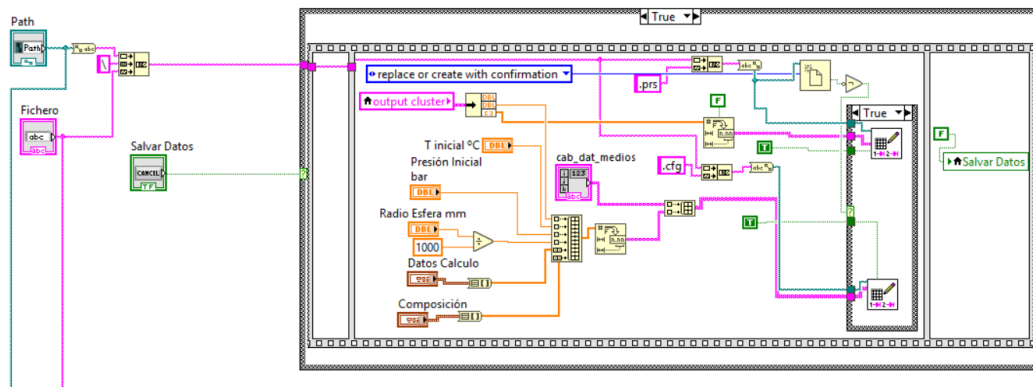


Fig. 4.3.4. Fracción del programa *Diagnóstico* encargada de la generación de ficheros (Elaboración propia)

### 3.4. EJECUCIÓN DEL DIAGNÓSTICO

Creados los archivos necesarios con toda la información sobre la bomba y las condiciones termodinámicas se procede a su diagnóstico.

El archivo de diagnóstico es *diagbomba\_polinom.exe* y contiene un programa basado en los cálculos antes descritos.

Como se dijo anteriormente, el ejecutable encargado del diagnóstico debe encontrarse en la misma carpeta que los archivos creados. Por lo tanto, la ruta escrita para ellos valdrá para encontrar este archivo.

Se introducirá el siguiente comando en LabView para diagnosticar los archivos creados.

***cmd/c (Ruta del archivo)\diagbomba\_polinom (nombre del archivo)***

El anterior comando abrirá la ventana de comando de Windows por lo que debe ser abierto con SYSTEM EXEC VI. (Fig. 4.4.1)

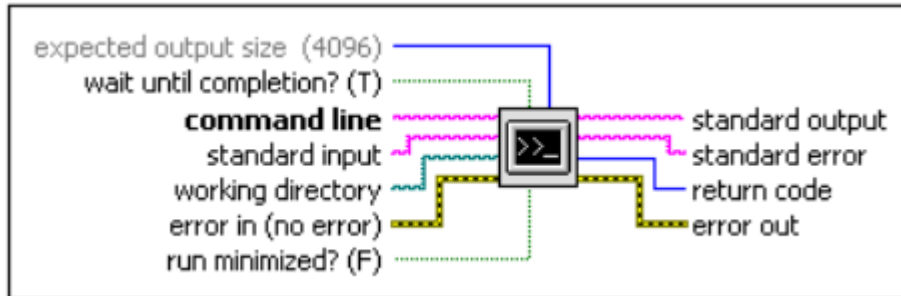


Fig. 4.4.1. Función System Exec VI (8)

Hay que añadirle también la entrada del directorio de trabajo que es la ruta de los archivos ya que se desea que el resultado se almacene en la misma carpeta. También se añade la salida de error para, en caso de fallo, poder visualizarlo y repararlo. Al ser la variable booleana que permite esperar a la ejecución de este diagnóstico TRUE por defecto, no se añade ninguna acción más.

Faltaría solo una variable booleana que activase la ejecución del análisis, aunque en este caso también activará la representación gráfica posterior.

En la siguiente imagen se muestra un esquema completo de la etapa de diagnóstico. (Fig. 4.4.2)

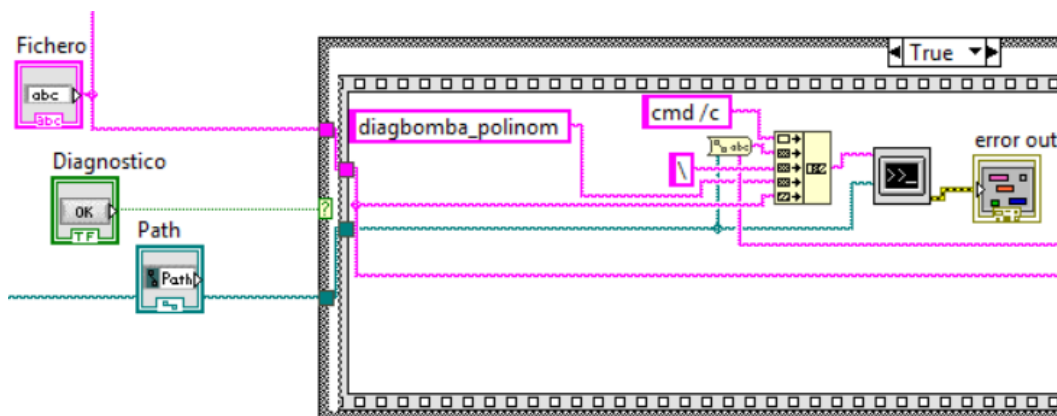


Fig. 4.4.2. Fracción del programa *Diagnóstico* encargada el análisis de los ficheros. (Elaboración propia)

El resultado del diagnóstico se guardará en la ruta seleccionada en un fichero cuyo nombre será igual al de los ficheros analizados, pero con extensión res.

### 3.4.1. ARCHIVO .RES GENERADO

Hemos abierto el archivo creado con el bloc de notas y su estructura es la siguiente. (Fig. 4.4.1.1)

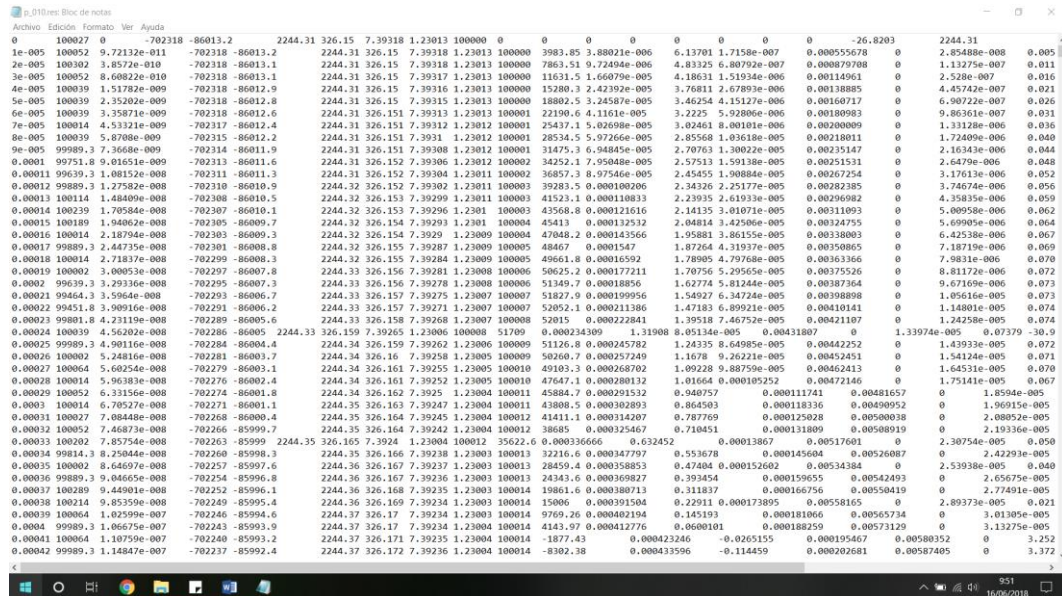


Fig. 4.4.1.1. Estructura archivo .res generado por nuestro software (Elaboración propia)

El archivo consta de 20 columnas que representan cada una de las variables que resultan del diagnóstico como temperaturas, presiones y velocidades de combantion y varias filas que representan su evolución en el tiempo.

### 3.5. REPRESENTACIÓN GRÁFICA DE LOS RESULTADOS

En esta etapa se llevarán a cabo dos funciones distintas:

- Lectura y transformación del archivo de compilación en un array en formato DBL.
- Captación y representación gráfica de los datos que necesitamos de ese array.

Para el primer paso se usa una subrutina interna de LabView llamada READ FROM SPREADSHEET.VI a la cual debemos introducir la ruta completa y extensión del archivo que queremos leer.

Asimismo, se ofrece la opción de transponer la lectura leída añadiendo un true a la entrada que lo habilita. Se hará uso de esta herramienta de tal forma que nuestra matriz resultante tenga una fila por cada parámetro extraído del análisis.

Por último, se puede introducir el número de filas que deben ser leídas en lo que escogeremos todas introduciendo un -1.

El resultado de este proceso proporcionará una salida en forma de matriz en formato LabView DBL que posteriormente se usará para graficar los resultados.

En la siguiente imagen se ve cómo es el funcionamiento interno de esta subrutina. (Fig. 4.5.1)

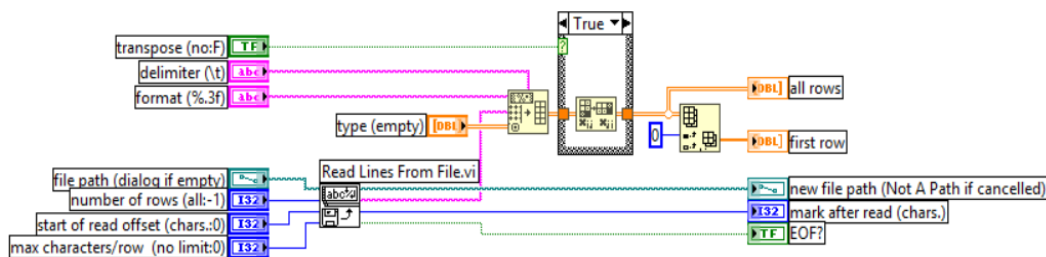


Fig. 4.5.1. Funcionamiento interno de la subrutina Read from spreadsheet.vi (Elaboración propia)

Una vez generada la matriz se procede a realizar las gráficas que se mostraran por pantalla como resultado del análisis. Para ello se creará una subrutina llamada GRAFICAS.

### 3.5.1. SUBRUTINA “GRAFICAS”

La única entrada a esta subrutina será la matriz en formato DBL generada en el paso anterior.

El objetivo será representar una serie de variable respecto al tiempo como la presión la fracción de calor liberada y el error cometido y otras frente a la temperatura de los no quemados como la velocidad de combustión y la temperatura adiabática y la temperatura de quemados.

Para ello se iran recolectando las filas correspondientes a estas magnitudes y las juntaremos con sus pares de representación (tiempo o temperatura de no quemados). Además, se juntarán aquellas que guarden relación para que la comparativa de sus evoluciones se vea de forma más sencilla.

Las filas representativas son las siguientes:

- Fila 0 → Tiempo (s)
- Fila 1 → Presión experimental. (bar)
- Fila 5 → Temperatura quemados (K)
- Fila 6 → Temperatura sin quemar (K)
- Fila 9 → Presión calculada (bar)
- Fila 10 → Derivada de la presión
- Fila 12 → Velocidad de combustión (m/s)
- Fila 16 → Fracción de calor liberado.
- Fila 17 → Derivada de la fracción de calor liberado.
- Fila 18 → Error (Pa)
- Fila 19 → Temperatura adiabatica (K)

Con ayuda de la función *Index Array* se irán seleccionando las filas anteriores de inicio a fin (sin especificar un límite de columnas). (8)

Posteriormente se usará la herramienta *Bundle Function* para crear los distintos emparejamientos de representación antes mencionados. Para evitar errores insertaremos primero en los registros las variables del eje de abscisas (tiempo y T<sup>a</sup> sin quemar) y después las variables del eje de ordenadas (presiones, velocidades de reacción, etc). (8)

A continuación, con la ayuda de *Build Array Function* se juntarán las medidas que guardan algo de relación para que su representación sea conjunta como las distintas presiones o algunas magnitudes y sus derivadas. A fin de evitar confusiones solo se unirán magnitudes que estén enfrentadas a una magnitud igual (tiempo o T<sup>a</sup> sin quemar). (8)

Por último, las salidas de cada registro se representarán de forma simultánea en gráficas distintas, pero dentro de la misma ventana.

El resultado de la formación de la subrutina GRAFICAS se ve reflejado en la siguiente imagen. (Fig. 4.5.1.1)

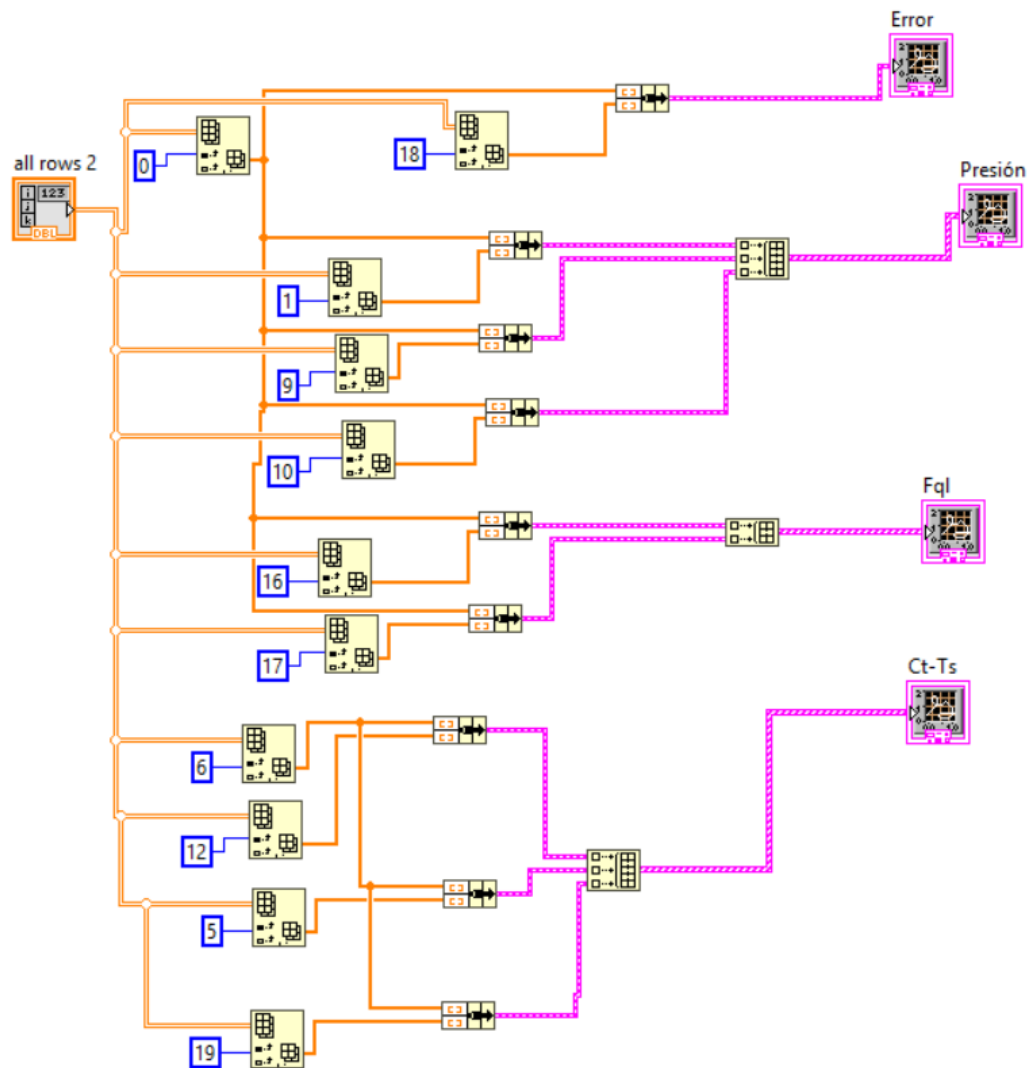


Fig. 4.5.1.1. Esquema completo de la subrutina GRAFICAS (Elaboración propia)

Como dijimos anteriormente, esta parte del programa se activa con el mismo actuador que el diagnóstico. No obstante, la ejecución de esta tarea se hará después de que la anterior etapa finalice hecho que se consigue simplemente insertándolo en otro recuadro distinto tras la etapa de análisis.

Aprovechamos también esta etapa para deshabitar el actuador que inicia el diagnóstico dejándolo así preparado para un futuro proyecto.

En la siguiente imagen se muestra como es la parte del programa encargada de la representación gráfica. (Fig. 4.5.1.2)

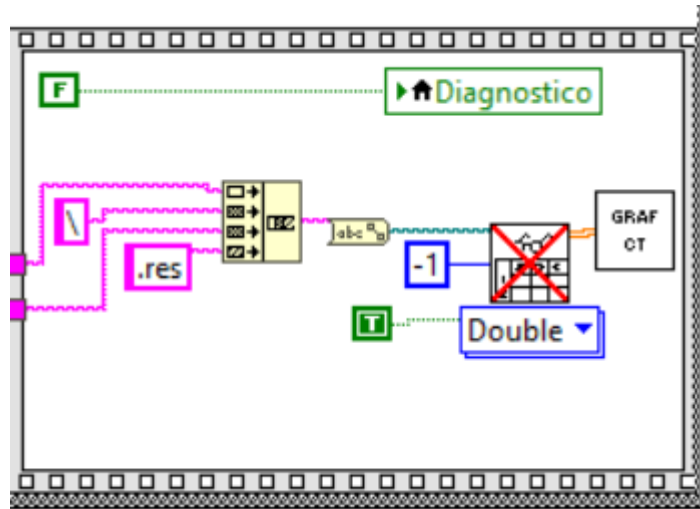


Fig. 4.5.1.2. Fracción del programa *Diagnóstico* encargada de la representación gráfica (Elaboración propia)

Conjuntamente, el análisis y representación de resultados que se activaría con el actuador de diagnóstico seguiría el siguiente esquema. (Fig. 4.5.1.3)

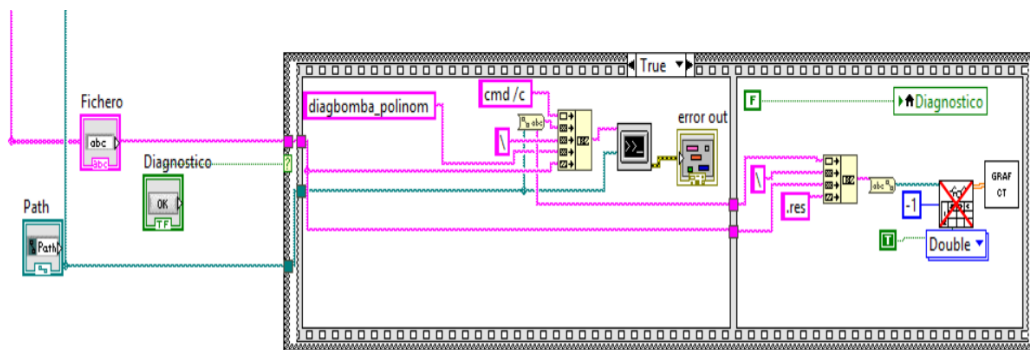


Fig. 4.5.1.3. Representación conjunta de las partes de análisis y representación de datos de programa *Diagnostico* (Elaboración propia)

Con esto estaría finalizado el diseño del programa al que solo faltaría añadirle un botón de reset que detenga la ejecución del programa en caso de que sea necesario y una herramienta llamada WAIT UNTIL NEXT MS MULTIPLE FUNCTION que ayudará con la sincronización del sistema evitando que los distintos experimentos o acciones se solapen.

Para que todo esto funcione se debe encerrar el programa dentro de un bucle while que permanecerá activo siempre que la señal de reset este deshabilitada.

En la siguiente imagen se muestra un esquema completo de nuestro programa. (Fig. 4.5.1.4)

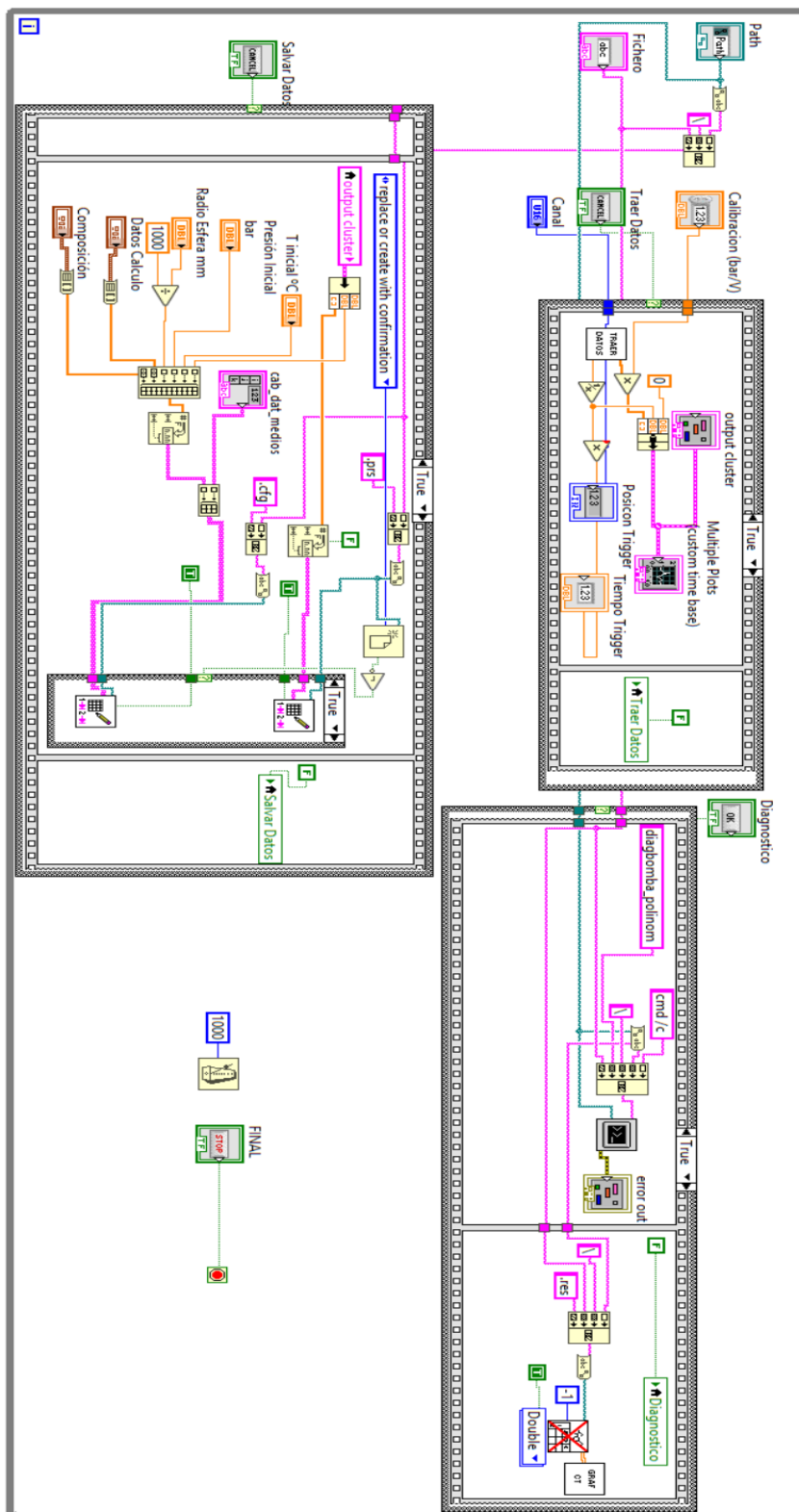


Fig. 4.5.1.4. Esquema completo del programa *Diagnostico* (Elaboración propia)



## 4. SIMULACIÓN DEL PROCESO

Todos los pasos antes descritos se generan en la pantalla de diagrama de bloques. Su programación genera una serie de una serie de elementos en el panel frontal que ordenados quedan de la siguiente manera. (Fig.5.1)

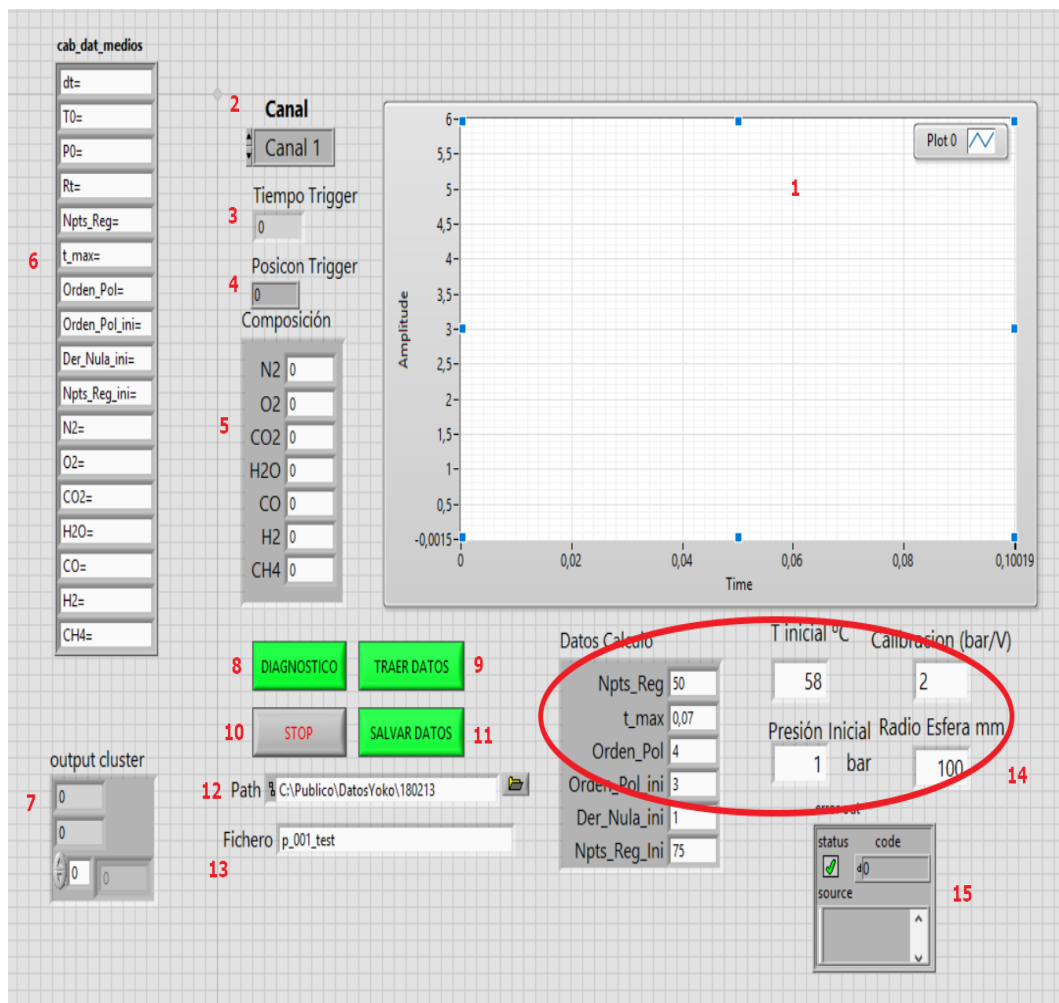


Fig. 5.1. Panel frontal del programa *Diagnóstico* (Elaboración propia)

Los elementos son los siguientes:

1. Representación gráfica de la presión captada.
2. Selector del canal del osciloscopio.
3. Resultado de tiempo hasta el instante de disparo.

4. Posición del instante de disparo.
5. Composición de los gases de entrada.
6. Título de los elementos para el archivo cfg (datos auxiliares).
7. Registro creado para la representación de la presión captada.
8. Activador de la parte de diagnóstico y representación de resultados.
9. Activador de la parte de captación de datos.
10. Botón de reset.
11. Activador de la parte de generación de ficheros.
12. Ruta de los archivos.
13. Nombre de los archivos.
14. Otros datos necesarios para el cálculo.
15. Registro de los posibles errores durante la ejecución.

Se comenzaría la experimentación introduciendo los datos de la composición del aire introducido en la bomba con la ayuda de los displays instalados en el laboratorio.

También se calibrarán los datos auxiliares de temperatura y presión iniciales, calibración del osciloscopio y radio de la esfera.

A continuación, se añaden la ruta del archivo y el programa de diagnóstico y el nombre que los archivos recibirán para el experimento.

Una vez determinados todos los datos se procede a la captación de las presiones recogidas por el osciloscopio clicando en el botón *TRAER DATOS* de la figura 5.1.

Durante la ejecución de esta etapa el panel frontal debería lucir como en la Fig.5.2.

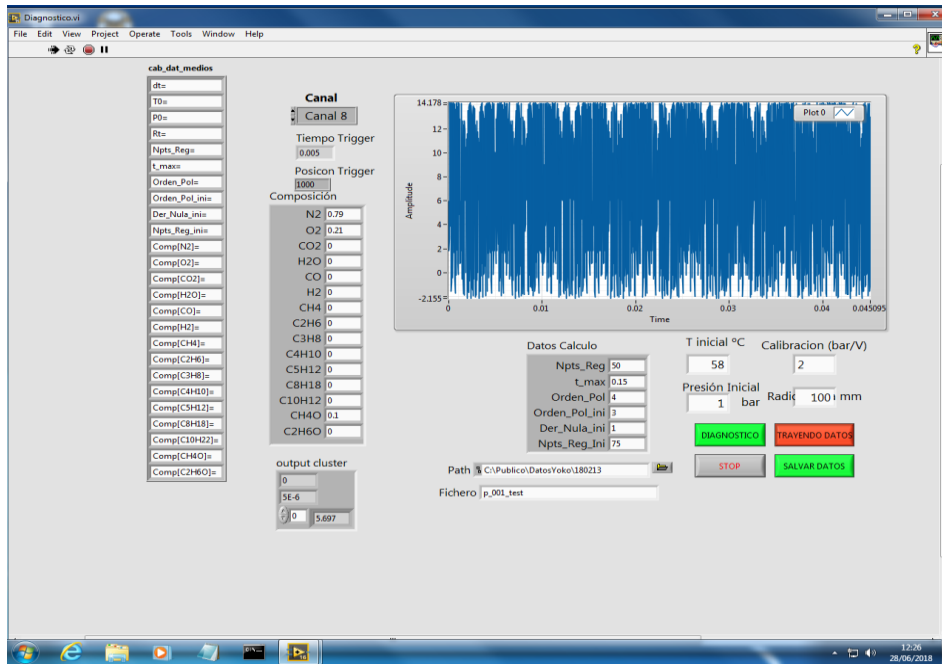


Fig. 5.2. Panel frontal durante la captación de datos (Elaboración propia)

Finalizado este proceso se mostrará en la representación gráfica del panel frontal (elemento 1 de la Fig. 5.1) la evolución de la presión captada durante la combustión por el osciloscopio. (Fig. 5.3)

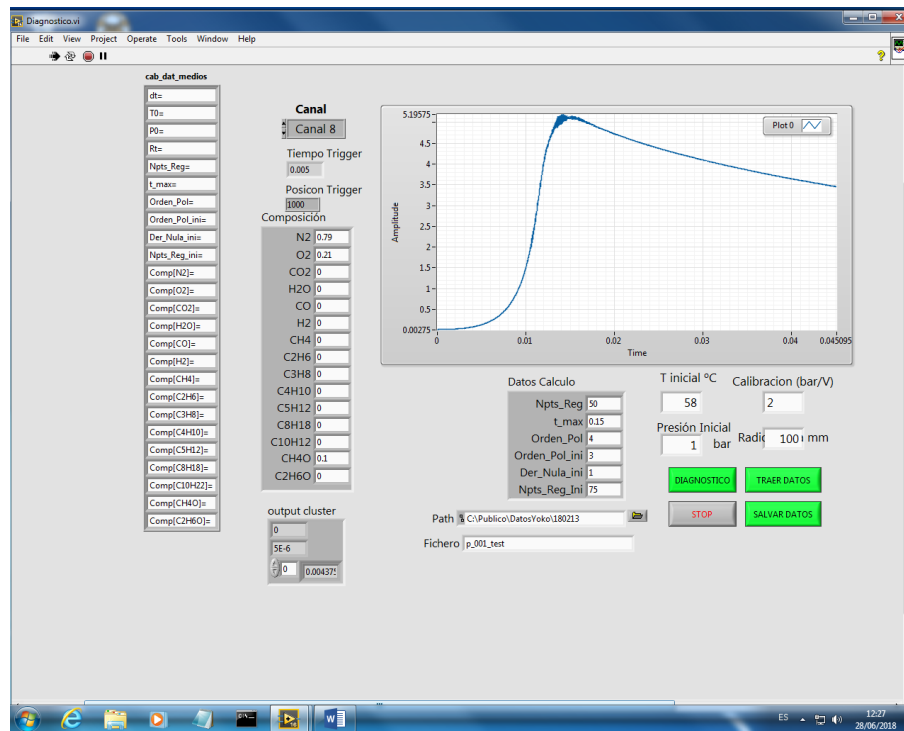


Fig. 5.3. Panel frontal tras la captación de datos (Elaboración propia)

El siguiente paso sería la generación de los ficheros para el diagnóstico. Para ello se hará clic sobre el botón *SALVAR DATOS*.

Una vez el botón vuelva a ser verde y contenga “salvar datos” en su interior significará que se han creado los archivos.

El último paso sería realizar el diagnóstico de los ficheros creados activando el botón *DIAGNOSTICO*.

Como ya dijimos este botón activa el análisis y la representación gráfica de los resultados. Una vez finalizada la etapa se mostrarán las gráficas correspondientes (Fig.5.4)

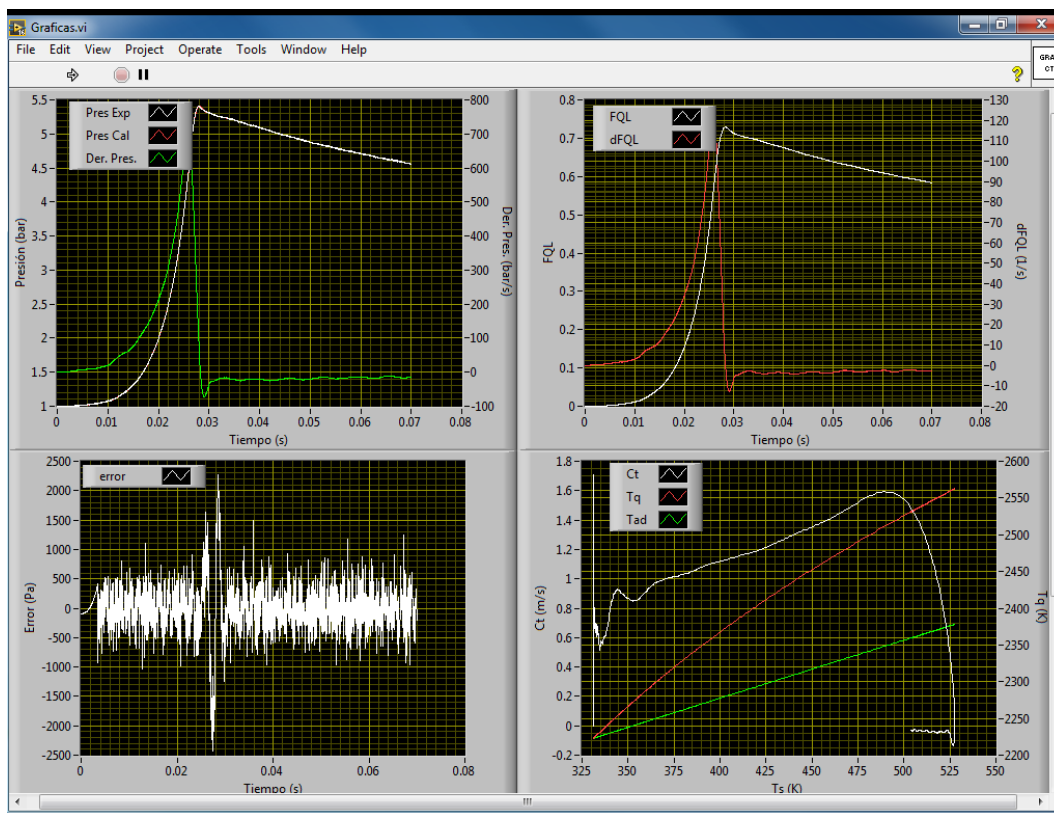


Fig.5.4. Gráficas resultantes del análisis y representación de resultados. (Elaboración propia)

La ejecución de todo el proceso podría llevar en torno a un minuto en función de los parámetros que haya que modificar de un experimento a otro.

## 5. CONCLUSIONES

Comparativamente con el trabajo en el que se ha basado realizado en septiembre de 2003 por D. Álvaro Lafuente Lafuente se han podido realizar mejoras importantes.

El beneficio fundamental se debe al tiempo necesario para la captación de datos habiéndose visto reducido de varios minutos a unos pocos segundos.

El hecho de integrar todas las etapas dentro de un mismo software traerá facilidades a la hora de su uso ya bien sea en entendimiento de las etapas como en la comodidad en la ejecución de varios experimentos.

Todos estos beneficios se ven incrementados teniendo en cuenta que las aplicaciones prácticas del mismo podrían conllevar un número importante de ensayos por lo que los beneficio sobre todo en tiempo serían muy importantes.

Como la mayoría de los beneficios estos tendrían un impacto económico importante reduciendo costes de mantenimiento y de ejecución del experimento. Dicha reducción de costes haría más viable esta experimentación y podría traer consigo mejoras en los futuros experimentos llevados a cabo en este laboratorio.

## 6. LINEAS DE FUTURO

A pesar de proponer grandes mejoras respecto a la metodología anterior, esta forma de proceder podría ser mejorada provocando beneficios tanto en la exactitud de los resultados como en la maniobrabilidad de la aparata. Si bien es cierto, estas mejoras podrían requerir de un instrumental mucho más avanzado y, por tanto, más caro. Debería estudiarse la viabilidad de las siguientes medidas antes de adoptarlas.

- Posibilidad de manipular los distintos parámetros del osciloscopio como el rango, offset o calibración desde el ordenador: Supondría una mayor programación en LabView de la expuesta en este TFM ya que faltaría una rutina que se encargase de dar órdenes al osciloscopio. En principio la aparata necesaria para esta mejora no debería ser modificada. En cuanto al beneficio, el menor número de desplazamientos al osciloscopio para su maniobrabilidad hará de la experimentación una actividad más breve y cómoda.
- Dotar al sistema de la capacidad tanto de captar como modificar los caudales de los gases que entran la bomba: Esta mejora podría requerir una mejora de los componentes del experimento como el osciloscopio (si no tiene un número de canales suficiente para el control de estas magnitudes) o las válvulas de admisión (si no permiten su programación para el control de los caudales). Además, supondría un programa más desarrollado que debería incluir las limitaciones del sistema para no incurrir en errores (por ejemplo: caudales máximos asumibles). Los principales beneficios de esta medida sería una mayor rapidez en la modificación de estos parámetros y una mayor precisión en los cálculos debido a la mayor precisión en la introducción de los mismos.
- Introducir los datos de temperatura de forma experimental: Sería necesario usar un canal del osciloscopio en cuyo caso no sería necesario cambiarlo. Se requeriría un captador de la temperatura adicional al que ya tenemos de presión. El programa de LabView sería más complejo y el software de diagnóstico debería tener en cuenta que esta magnitud sería introducida manualmente y no necesitaría ser calculada. El beneficio principal sería una mayor precisión en los resultados ya que se tendría en cuenta los posibles efectos que hagan que el cálculo de la temperatura de forma teórica cometa errores.

- Una metodología de cálculo del software de diagnóstico que cometa menos error: Como se observa en la Fig. 5.4. el error cometido en algunas partes del cálculo es bastante importante por lo que un método más exacto mejoraría la fiabilidad de los resultados del mismo. Conllevaría un mayor tiempo de investigación y de programación de la nueva metodología.
- Generación de informes sobre los resultados: La interpretación de los resultados sería mucho más sencilla que la actual ya que ahorraría el estudio de las gráficas de resultado. Se ganaría en rapidez y permitiría de una forma más sencilla la creación de un histórico de resultados. La programación necesaria sería mucho más compleja que la actual y dependiendo del nivel de profundidad de los informes puede que fuese necesario un procesador más potente que el actual para que el ahorro de tiempo sea determinante.





## BIBLIOGRAFÍA

- (1) Á. Lafuente Lafuente, “Metodología para el diagnóstico de la velocidad de combustión laminar de mezclas de gases combustibles a partir de la medida de presión instantánea en una bomba de combustión a volumen constante.,” Universidad de Valladolid, 2003.
- (2) Yokogawa Electric Corporation, “Digital Scope,” IM701820-01E, 1999.
- (3) “Tutorial de labview.” [Online]. Available: <http://www.esi2.us.es/~asun/LCPC06/TutorialLabview.pdf>. [Accessed: 14-Jul-2018].
- (4) RS, “Cable eval. USB-RS232, FT232RL,UC232R-10.” [Online]. Available: [https://es.rs-online.com/web/p/products/0429252/?grossPrice=Y&cm\\_mmc=ES-PLA-DS3A-\\_google-\\_PLA\\_ES\\_ES\\_Semiconductores\\_-\\_Kits\\_De\\_Desarrollo\\_Para\\_Semiconductor%7CKits\\_De\\_Desarrollo\\_De\\_Interfaz-\\_PRODUCT+GROUP&matcht](https://es.rs-online.com/web/p/products/0429252/?grossPrice=Y&cm_mmc=ES-PLA-DS3A-_google-_PLA_ES_ES_Semiconductores_-_Kits_De_Desarrollo_Para_Semiconductor%7CKits_De_Desarrollo_De_Interfaz-_PRODUCT+GROUP&matcht). [Accessed: 14-Jul-2018].
- (5) National Instruments, “Conectar Instrumentos vía Serial.” [Online]. Available: <http://www.ni.com/getting-started/set-up-hardware/instrument-control/esa/serial-connect>. [Accessed: 14-Jul-2018].
- (6) Yokogawa Electric Corporation, “DL708E Digital Scope Communication Interface,” IM701820-11E, 1998.
- (7) Yokogawa Electric Corporation, “DL708E Digital Scope Communication Interface,” IM701820-11E\_12, 2005.

- (8) National Instruments, "LabView help," 2017. Available: LabView 2017 [Accessed: 14-Jul-2018].
- (9) National Instruments, "Numeric Data Types Table." [Online]. Available: [http://zone.ni.com/reference/en-XX/help/371361G-01/lvhowto/numeric\\_data\\_types\\_table/](http://zone.ni.com/reference/en-XX/help/371361G-01/lvhowto/numeric_data_types_table/). [Accessed: 14-Jul-2018].