



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH**

PROYECTO FINAL DE CARRERA

AUTOMATIZACIÓN DE MEDIDAS

DE POTENCIA DE UN LÁSER

(AUTOMATED LASER POWER SYSTEM)

Grado: Ingeniería de sistemas de telecomunicaciones

Autor: Juan Carlos Pérez Ramos

Director: Alejandro Rodríguez

Año: 2018



Abstract

For the development of this Degree's thesis, it gets started from the realization of a past thesis which is responsible for monitoring the power measurements of a lidar system (Light Detection and Ranging) through different stages (power capture, converter, amplifier and AD conversion and processing) to subsequently send the processed measurements through a serial port and then, this data, will be treated in this project for its automatization. To do this, the graphic programming tool Labview is used. By working in a visual way, it makes easier to integrate measurement hardware from any supplier, represent a complex logic in the diagram and design custom user interface, and, it is perfect for the case in which we are.

Resum

Per el disseny d'aquest projecte, es parteix de la realització d'un projecte passat, el qual, s'encarrega de monitoritzar les mesures de potència d'un sistema lidar (Light Detection and Ranging) mitjançant diferents etapes (captació de potència, conversora, amplificadora y de conversió AD y processat) per, posteriorment, enviar les dades processades a través d'un port sèrie i les quals es tractaran en aquest projecte per la seva automatització. Per fer-ho, s'utilitza una eina de programació gràfica, Labview. Al treballar d'aquesta manera, fa que sigui molt més fàcil integrar hardware de mesures de qualsevol proveïdor, representar una lògica complexa al diagrama i dissenyar interfícies d'usuari personalitzades, lo que ho fa perfecte pel cas en el que ens trobem.

Resumen

Para el desarrollo de este proyecto, se parte de la realización de un proyecto pasado el cual, se encarga de monitorizar las medidas de potencia del un sistema lidar (Light Detection and Ranging) mediante diferentes etapas (captación de potencia, conversora, amplificadora y de conversión AD y procesado) para, posteriormente, mandar las medidas procesadas a través de un puerto serie y las cuales se tratarán en este proyecto para su automatización. Para ello, se utiliza la herramienta de programación gráfica, Labview. Al trabajar de una forma visual, hace que sea más fácil integrar hardware de medidas de cualquier proveedor, representar una lógica compleja en el diagrama y diseñar interfaces de usuario personalizadas, lo que lo hace perfecto para el caso en el que nos encontramos.



Agradecimientos

A mis padres, por su fe ciega en mi a lo largo de la vida, su confianza en mi y por creer que soy capaz de todo.

En cuanto al trabajo final de grado, agradecer a Alejandro Rodríguez la confianza para asignarme el proyecto y dejarme trabajar de una forma independiente. Así mismo, a los técnicos del laboratorio, que me hicieron un lugar sin problemas para poder trabajar y pusieron a mi disposición el material necesario.

Índice

Abstract	1
Resum	2
Resumen	3
Agradecimientos	4
Índice	5
Lista de Figuras	7
1. Introducción	8
1.1. Punto de partida	8
1.2. Objetivos del proyecto	9
2. Labview	10
2.1. Introducción a LabView	10
2.2. Principales características	10
3. Desarrollo del proyecto	11
3.1 Lectura por puerto serie	11
3.1.1 Configuración del puerto serie en el diagrama de bloques	11
3.1.2 Visualización en la interfaz de usuario	12
3.1.3 Espera a la entrada de datos	13
3.2 Bloque de lectura de datos y mostrado por pantalla	14
3.2.1 Bloque de lectura de datos y mostrado por pantalla	14
3.2.2 Visualización en la interfaz de usuario	16
3.3 Umbral mínimo de potencia	17
3.4 Estructura de media temporal por minuto	17
3.5 Estructura de media temporal horaria	20
3.6 Estructura de media temporal diaria	21
3.7 Estructura de escritura del fichero diario	21

3.8	Estructura de media temporal semanal.....	23
3.9	Estructura de media temporal mensual.....	24
3.9.1	Generación y activación de la estructura de un mes de 30 días	24
3.9.2	Generación y activación de la estructura de un mes de 31 días	25
3.9.3	Generación y activación de la estructura de febrero.....	26
3.9.4	Generación y activación de la estructura de febrero bisiesto	26
4.	Generación del gráfico de medias temporales	27
5.	Conclusiones	28
	Bibliografía:	30

Lista de Figuras

Fig. 1.1 Sistema de monitorización y transmisión de los datos hacia el PC	10
Fig. 3.1 VISA bloque de configuración del puerto serie.....	12
Fig. 3.2 Configuración del puerto serie en el diagrama de bloques de LabView	13
Fig. 3.3 Visualización de la configuración del puerto serie en la interfaz de usuario	14
Fig. 3.4 Bucle de espera a la entrada de datos.....	15
Fig. 3.5 Lectura de datos en el diagrama de bloques.....	15
Fig. 3.6 NI-VISA bloque de lectura de puerto serie	15
Fig. 3.7 Bloque de almacenado y muestreo por pantalla	15
Fig. 3.8 Extracción del valor de potencia	15
Fig. 3.9 Bloque de LabView Spreadsheet String to Array Function	16
Fig. 3.10 Bloque de Labview Index Array Function	16
Fig. 3.11 Visualización de las medidas en la interfaz de usuario.....	17
Fig. 3.12 Umbral mínimo de potencia	17
Fig. 3.13 Estructura de media temporal por minuto.....	18
Fig. 3.14 Obtención de la ruta para el fichero de medias temporales por minuto	18
Fig. 3.15 Estructura para medias temporales por minuto cuando el estado es true o activado.....	19
Fig. 3.16 Estructura para medias temporales por minuto cuando el estado es false o desactivado	19
Fig. 3.17 Estructura de media temporal horaria	20
Fig. 3.18 Estructura de media temporal diaria.....	21
Fig. 3.19 Fichero diario de las medias temporales.....	21
Fig. 3.20 Estructura de escritura del fichero diario	22
Fig. 3.21 Generación del fichero diario y activación de la estructura.....	22
Fig. 3.22 Escritura del fichero diario.....	23
Fig. 3.23 Estructura de generación del fichero semanal	23
Fig. 3.24 Estructura de escritura del fichero mensual	24
Fig. 3.25 Generación y activación de la estructura de un mes de 30 días.....	24
Fig. 3.26 Generación y activación de la estructura de un mes de 31 días.....	25
Fig. 3.27 Generación y activación de la estructura de febrero	26
Fig. 3.28 Generación y activación de la estructura de febrero bisiesto	26
Fig. 4.1 Bloque Merge Signals Function de LabView	27
Fig. 4.2 Generación del gráfico de medias temporales	27
Fig. 4.3 Gráfica de las medias temporales en la interfaz de usuario	28

1. Introducción

Este proyecto surge de la necesidad de procesar las medidas de potencia de un láser y almacenarlas mediante medias temporales en un fichero de texto para su posterior tratado en Matlab o programas similares. La intención de este trabajo es tener un control sobre el estado del láser, poder tener un histórico de los datos generados y poder detectar fallos o anomalías.

Para la realización del proyecto se necesita de software y hardware.

Hardware

Por la parte de hardware se necesita:

- Modulo de monitorización de potencia (mirar apartado 1.1, punto de partida de este trabajo).
- PC en el cual pueda estar ejecutándose continuamente el código.
- Adaptador Serie/USB dado a la falta de existencia de puertos series en los ordenadores actuales.
- Cable Serie.

Software

Por la parte de software se necesita:

- LabView con licencia activada.
- Sistema operativo compatible con el adaptador (no soporta Windows 10).

1.1. Punto de partida

Para poder empezar con este proyecto se parte de un trabajo final de grado anterior. Dicho trabajo, realizado por Jordi Bach Aguilà, se basa en un sistema de monitorización de la potencia del láser. Éste módulo, es el encargado de transmitir las medidas de potencia a través de un puerto serie y las cuales acaban llegando al programa de Labview generado en este trabajo. Acto seguido se procesan y almacenan según lo deseado.

Esquema del sistema de monitorización

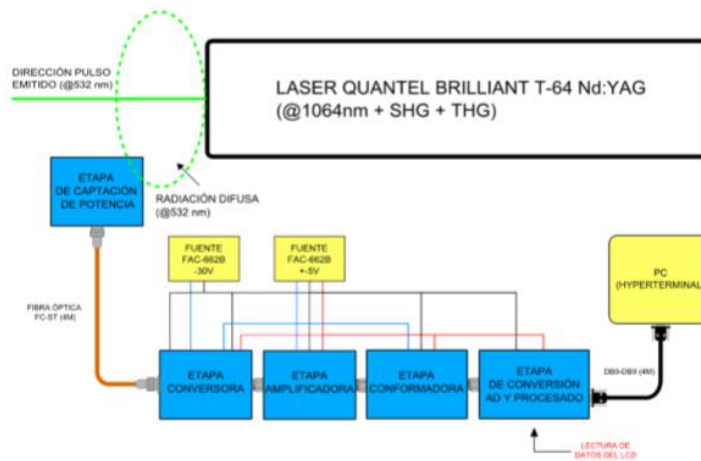


Fig. 1.1 Sistema de monitorización y transmisión de los datos hacia el PC

1.2. Objetivos del proyecto

Como se puede observar en la Fig. 1.1, no había existencia de un programa posterior encargado del procesado de los datos. Todo se observaba a partir de un Hyperterminal, lo que lo hacía poco óptimo para tener un control del comportamiento del láser y, de esto, nace este proyecto. Bajo la demanda de un almacenamiento de los datos y su procesado, se decidió que se necesitaba crear un sistema de automatización de datos de potencia y la herramienta destinada a este objetivo fue Labview.

Mediante la realización del proyecto, se pretende almacenar medias temporales de las medidas de potencia emitidas por el láser y, al finalizar el día, almacenar los datos obtenidos en un fichero de texto para que se puedan procesar mediante Matlab o otra herramienta parecida.

Este proyecto también ofrece una interfaz de usuario amigable en la cual se pueden observar los valores de potencia actuales a tiempo real, así como los valores de las últimas medias temporales, ya sean cada minuto, cada hora, cada día, cada semana o cada mes.

2. Labview

2.1. Introducción a LabView

Labview (Laboratory Virtual Instrument Engineering) es un software desarrollado por National Instruments que te permite crear sistemas a partir de un lenguaje de programación visual gráfico o lenguaje G, como también es comúnmente conocido. Es recomendado para integrar hardware de medidas de cualquier proveedor, representar una lógica compleja en el diagrama, desarrollar algoritmos de análisis de datos y diseñar interfaces de usuario personalizadas.

Primeramente, este programa fue creado para funcionar sobre máquinas MAC, pero hoy en día está disponible para plataformas Windows, UNIX, MAC y GNU/Linux. Labview permite tanto el diseño del firmware de un instrumento de última generación, como la programación de alto nivel del mismo y, todo esto, con código abierto.

Los programas desarrollados en LabView se llaman instrumentos virtuales o VI, dado a su origen como control de instrumentos. En la actualidad no solo tiene esta funcionalidad, puesto que su uso se ha expandido ampliamente no sólo al control de todo tipo de electrónica sino también a su programación, comunicaciones, matemáticas, etc.

Otro punto a favor de LabView es que permite combinarse con todo tipo de hardware y software, tanto del propio fabricante como de otros.

2.2. Principales características

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. Todo esto gracias a su programación gráfica, la cual hace todo mucho más intuitivo y no se necesita el conocimiento de ningún lenguaje específico, dado a que se basa en la interconexión de bloques.

Para los amantes de lo complejo también hay lugar en este programa. Con LabView pueden crearse programas de miles de Vis para aplicaciones complejas, programas de automatización, proyectos para combinar nuevos VIs con Vis ya existentes e incluso existen buenas prácticas de programación para optimizar el rendimiento y la calidad de la programación.

Yo, como iniciación a esta herramienta (dado que no sabía de la existencia de ésta), puedo decir que es una herramienta bastante intuitiva, quizás es algo complejo ajustar la lógica

de programación adquirida en otros lenguajes típicos a un lenguaje de programación gráfico, pero todo es acostumbrarse. Después del desarrollo de este proyecto en LabView (aunque seguramente a un nivel muy básico), como primer contacto, considero que salgo con una opinión bastante positiva de esta herramienta.

En los siguientes módulos pasaré a explicar el desarrollo del proyecto, así como cada parte del diagrama de bloques y su influencia en la interfaz de usuario.

3. Desarrollo del proyecto

3.1 Lectura por puerto serie

Para la lectura de datos a través del puerto serie, se necesita tener instalado en LabView el software de National Instruments, VISA (Virtual Instrument Software Architecture). NI-VISA ofrece las funcionalidades necesarias para poder comunicarte a través de un puerto serie, pero también a través de otros protocolos como Ethernet, USB, etc.

A continuación, se pasará a mostrar la configuración de la lectura de datos por el puerto serie, así como los parámetros necesarios para establecer la comunicación y, también, como se ve reflejado en la interfaz de usuario y como puede cambiar los diferentes valores.

A partir de este punto hablaremos de diagrama de bloques como la ventana donde se interconectan los diferentes bloques de LabView y de interfaz de usuario como la pantalla en la cual se verán reflejados los datos de dichos bloques.

3.1.1 Configuración del puerto serie en el diagrama de bloques

Para empezar a configurar el puerto serie necesitamos un bloque de configuración que proporciona NI-VISA. Este bloque es el siguiente:

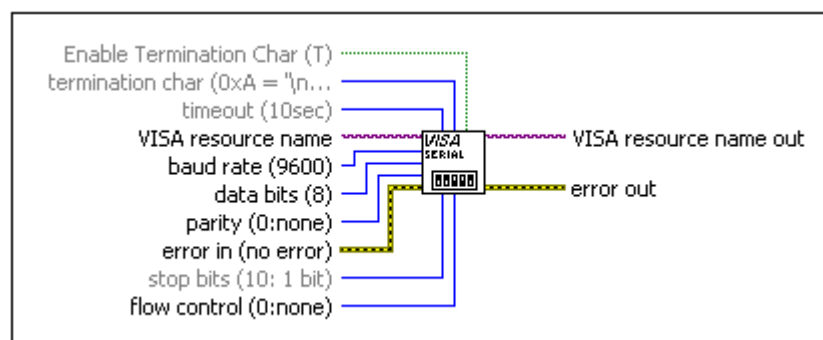


Fig. 3.1 VISA bloque de configuración del puerto serie

Como se puede observar en la Fig. 3.1, el bloque permite como entrada diferentes valores de la configuración del puerto serie, como pueden ser la velocidad de transmisión del enlace, los bits de datos, el bit de parada o el nombre del puerto que se está utilizando. El sistema de monitorización a partir del cual se trabaja establece una configuración para el puerto serie de 19200 a 8N1, esto es una velocidad de transmisión de 19200 bits por segundo, con 8 bits de datos, ningún bit de paridad para el control de errores y 1 bit de parada. Por lo tanto, con esto y el nombre del puerto a utilizar (COM en el que esté fijado el adaptador), podemos pasar a la configuración del puerto serie en el diagrama de bloques de la siguiente forma.

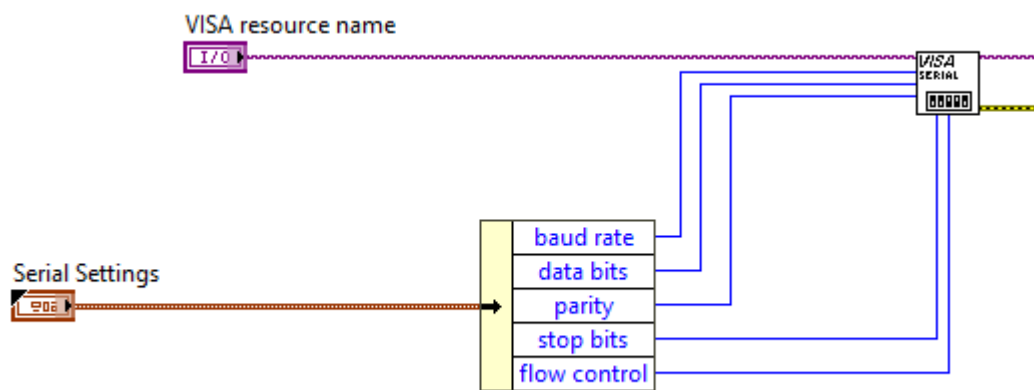


Fig. 3.2 Configuración del puerto serie en el diagrama de bloques de LabView

Como podemos ver en la figura, se ha utilizado una estructura de datos mediante una VI externa bajo el nombre Serial Settings que se encarga de crear un bloque de configuración para la interfaz de usuario que veremos en el siguiente punto. Además, se da la opción de selección del puerto serie a usar mediante el bloque VISA resource name. Una vez conectada toda la configuración, es hora de pasar al bucle principal de la aplicación, no sin antes pasar por la interfaz de usuario.

3.1.2 Visualización en la interfaz de usuario

Toda la configuración del punto anterior hace que en la interfaz de usuario quede de la forma siguiente.

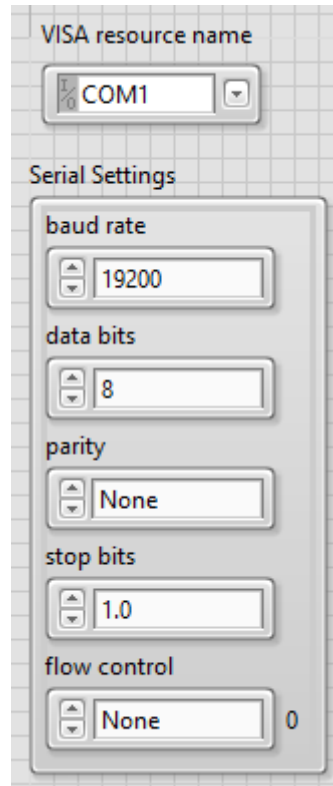


Fig. 3.3 Visualización de la configuración del puerto serie en la interfaz de usuario

Como se puede observar en la figura anterior, se muestra por defecto la configuración con la que debe trabajar el programa, pero siempre se puede cambiar haciendo click en las flechas en función de si se quiere subir o bajar el valor o bien haciendo click en el campo del valor, lo que hará que se abra un desplegable con las opciones a elegir.

3.1.3 Espera a la entrada de datos

En función de no almacenar datos no deseados ni hacer fallar el programa, es necesario que, en cada iteración del bucle principal y antes de realizar ninguna acción, se compruebe que existen datos en el puerto serie dispuestos a ser leídos y, si no es el caso, esperar a que existan. Para ello se utiliza un bucle de espera, del cual solo se saldrá si hay datos a leer o bien, si hay un error en el puerto serie, lo que parará completamente la aplicación. Dicho bloque es el siguiente.

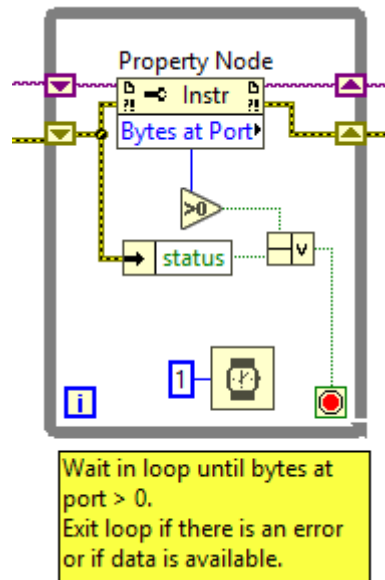


Fig. 3.4 Bucle de espera a la entrada de datos

Este bucle no tiene ningún impacto en la interfaz de usuario, es algo interno del código para el control de errores y garantizar correctamente su funcionamiento.

3.2 Bloque de lectura de datos y mostrado por pantalla

El siguiente bloque para comentar es el encargado de recoger los datos que van llegando al puerto serie e ir almacenándolos uno detrás de otro para ir mostrándolos por pantalla, así como extraer el valor de potencia de cada uno de ellos y mostrarlo en una gráfica. Además, dicho valor se utilizará para ir almacenándose en diferentes vectores para su posterior media temporal.

3.2.1 Bloque de lectura de datos y mostrado por pantalla

La configuración del bloque en el diagrama es la siguiente.

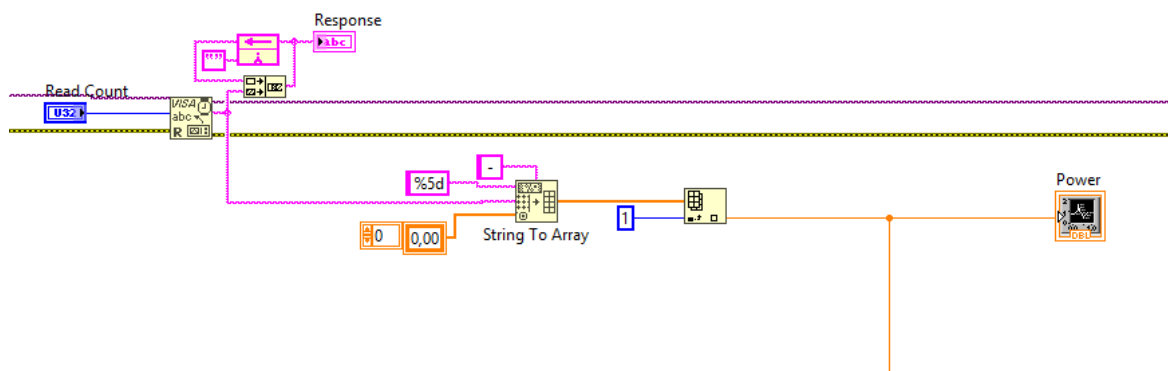


Fig. 3.5 Lectura de datos en el diagrama de bloques.

Primeramente, nos encontramos con el bloque de lectura del puerto serie de NI-VISA:

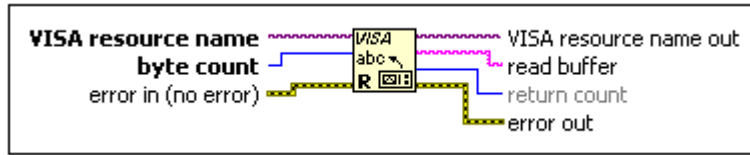


Fig. 3.6 NI-VISA bloque de lectura de puerto serie

A este bloque hay que pasarle como parámetros de entrada el puerto serie del cual se quiere leer la información, un bus de información sobre el estado del puerto y también el número de bytes máximos que queremos leer. Como parámetros de salida, seguiremos teniendo el nombre del puerto, si ha habido error o no, el número de bytes leídos y el búfer de lectura. Este búfer, es el que iremos almacenando para mostrar los datos por pantalla y del cual extraeremos los valores de potencia.

Para el almacenamiento de los datos y su mostrado por pantalla, utilizaremos el siguiente bloque.

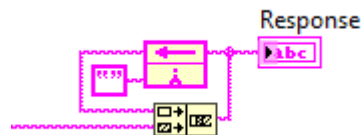


Fig. 3.7 Bloque de almacenado y muestreo por pantalla

Este bloque es bastante sencillo. Su función se basa en crear una cola de datos, en la cual creará una nueva línea por cada valor leído por el puerto serie y lo mandará al indicador Response, que es el encargado de mostrarlo por la interfaz de usuario. Como se muestra en la Fig. x.x.

Para la extracción del valor de potencia utilizaremos lo siguiente.

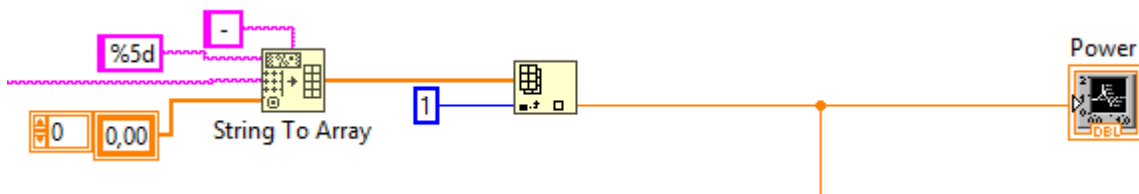


Fig. 3.8 Extracción del valor de potencia

Para la extracción del valor de potencia hay que tener en cuenta que el módulo de monitorización del cual parte este proyecto manda, para cada medida de potencia, un campo con el número de medida y otro campo con el valor de potencia, ambos de 5 dígitos y separados por un guion.

Teniendo en cuenta esto, utilizamos el bloque *Spreadsheet String To Array Function*, que se encarga de hacer un split del string que se le pase como entrada y mediante el formato que le digamos. El bloque en cuestión es el siguiente.

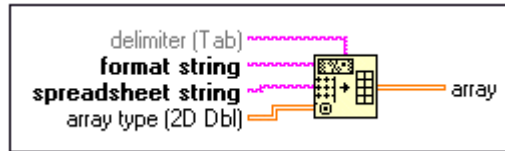


Fig. 3.9 Bloque de LabView Spreadsheet String to Array Function

De esta forma y como se puede observar en la Fig. 3.8, solo tenemos que indicarle que queremos que nos almacene en cada posición del vector 5 dígitos que vendrán separados por un guion. Esto se hace porque no nos interesa el valor del número de medida, solo necesitamos el valor de potencia para hacer las medias temporales. Por eso, acto seguido se utiliza el bloque *Index Array Function* que consultando la ayuda de LabView es el siguiente.

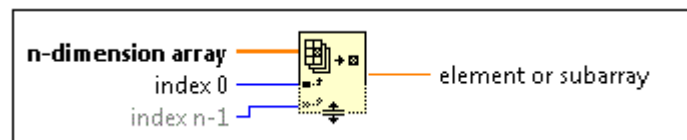


Fig. 3.10 Bloque de Labview Index Array Function

Este bloque necesita como parámetros de entrada el array de strings obtenido del bloque anterior y el índice del vector que queremos obtener. En nuestro caso, nos interesa el campo que contiene el valor de potencia y el cual es el segundo, por lo que le pasamos un uno y como elemento de salida, obtendremos el valor de potencia. Este valor se pasa a la gráfica en tiempo real de la potencia y se almacena en los diferentes vectores de medias temporales.

3.2.2 Visualización en la interfaz de usuario

Los valores leídos tal cual llegan por el puerto serie, es decir, en formato 5 dígitos – 5 dígitos, se mostrarán en la interfaz de usuario en el indicador bajo el nombre Response, mientras que el valor de potencia se verá reflejado en la gráfica bajo el nombre Power tal y como podemos ver en la siguiente imagen.

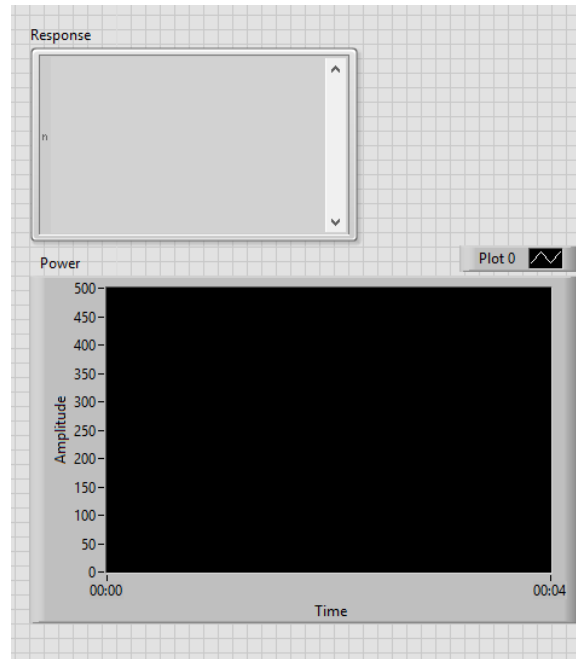


Fig. 3.11 Visualización de las medidas en la interfaz de usuario

3.3 Umbral mínimo de potencia

Como medida preventiva para no almacenar datos no deseados, existe un umbral mínimo de potencia, por debajo del cual, no se activa la estructura de medias temporales y, por lo tanto, se rechaza dicho valor. Esto es para evitar coger valores de potencia residuales captados por la fibra óptica y que no pertenezcan al láser. Dicho umbral se ve reflejado en la siguiente imagen.

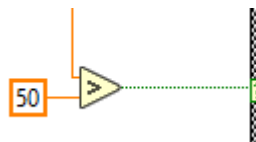


Fig. 3.12 Umbral mínimo de potencia

Para diseñar el umbral, simplemente se ha utilizado una operación lógica que comprueba que el valor de potencia extraído sea mayor al del umbral. Si se cumple, el resultado será true y activará la estructura de medias temporales.

3.4 Estructura de media temporal por minuto

El primer bloque de lógica a comentar en cuanto a media temporal es la media que se realiza cada minuto. Para ello, tenemos la siguiente estructura.

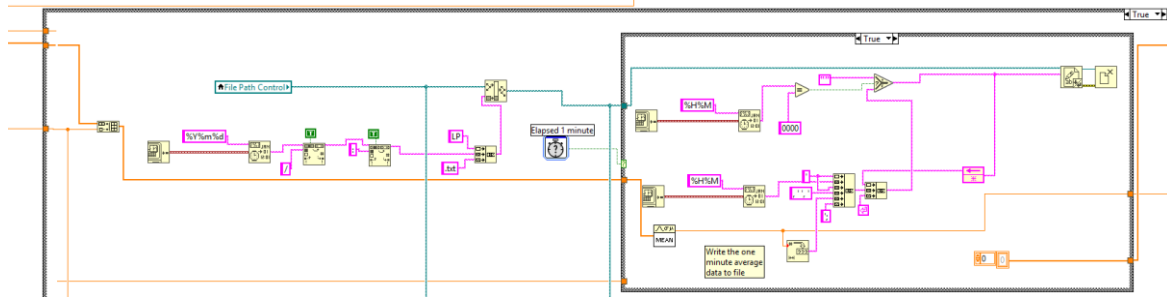


Fig. 3.13 Estructura de media temporal por minuto

Para hacerlo más entendible, dado a que la imagen es muy grande para ser observada, lo dividiremos en dos partes. La primera será la obtención de la ruta del fichero en el cual volcaremos la información de las medias temporales y la segunda, la lógica empleada para crear el contenido del fichero.

Para empezar con la primera parte, echaremos un ojo a la siguiente figura.

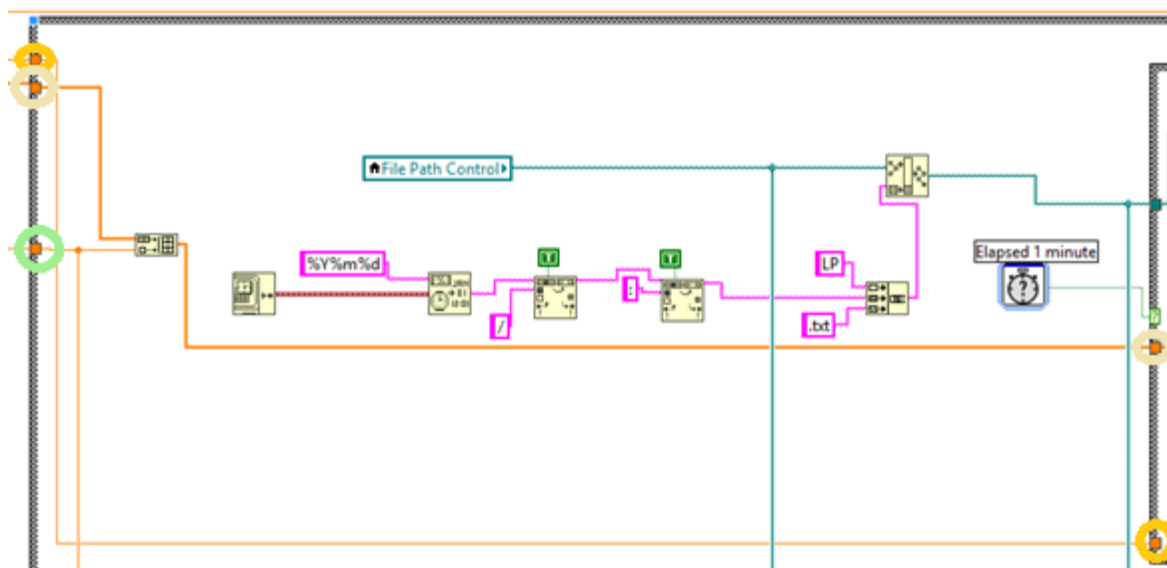


Fig. 3.14 Obtención de la ruta para el fichero de medias temporales por minuto

Marcado con círculos amarillos, tenemos un shift register con valor a 0. Un shift register, es una utilidad de LabView que permite mantener valores a lo largo de las iteraciones de un bucle. Este valor se pasa a la estructura de creación del fichero y que se comentará en la siguiente parte. En círculo gris, tenemos marcado el vector de medias temporales por minuto y al cual se va añadiendo, en una nueva posición, el valor de potencia marcado en círculo verde y que se le acabará pasando a la estructura de creación de fichero cuando el temporizador, bajo el nombre Elapsed 1 minute, detecte que ha pasado el tiempo establecido.

El resto se utiliza para la creación de la ruta del fichero en el cual se volcarán las medias temporales por minuto. En este caso se parte de la ruta establecida bajo la variable local File Path Control y en la cual se añadirá el fichero bajo el nombre generado a partir de la obtención de la fecha del día en el que nos encontremos en formato yyyyymmdd y la cual

será concatenada al string LP, que son las siglas de Laser Power. De esta forma, el fichero creado será LPyyyymmdd.txt y contendrá las medias temporales por minuto a lo largo de un día.

Para seguir con la segunda parte, tenemos las siguientes figuras.

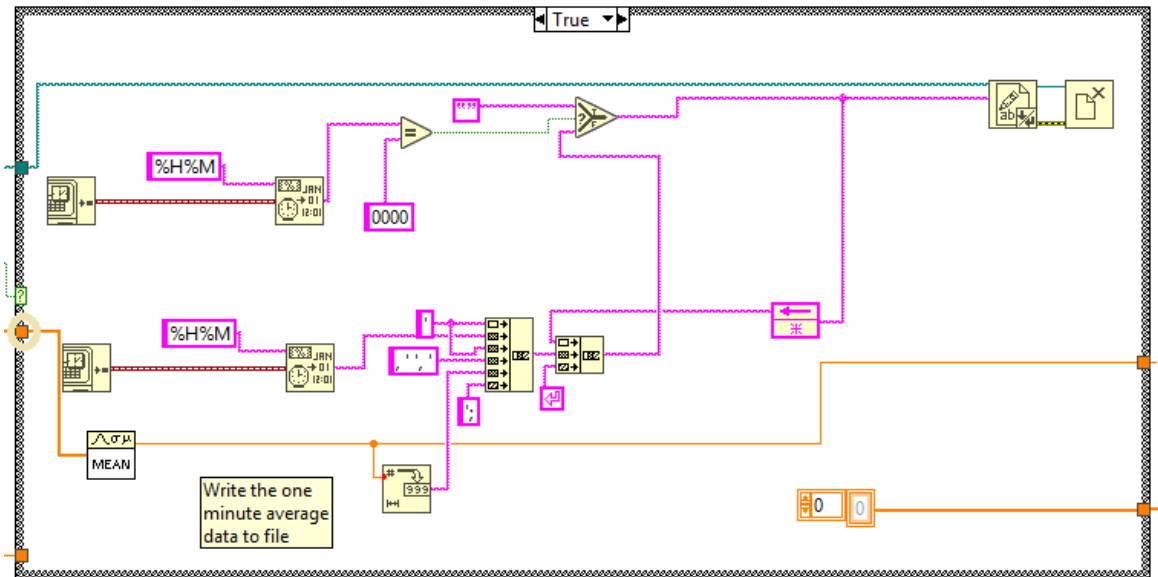


Fig. 3.15 Estructura para medias temporales por minuto cuando el estado es true o activado

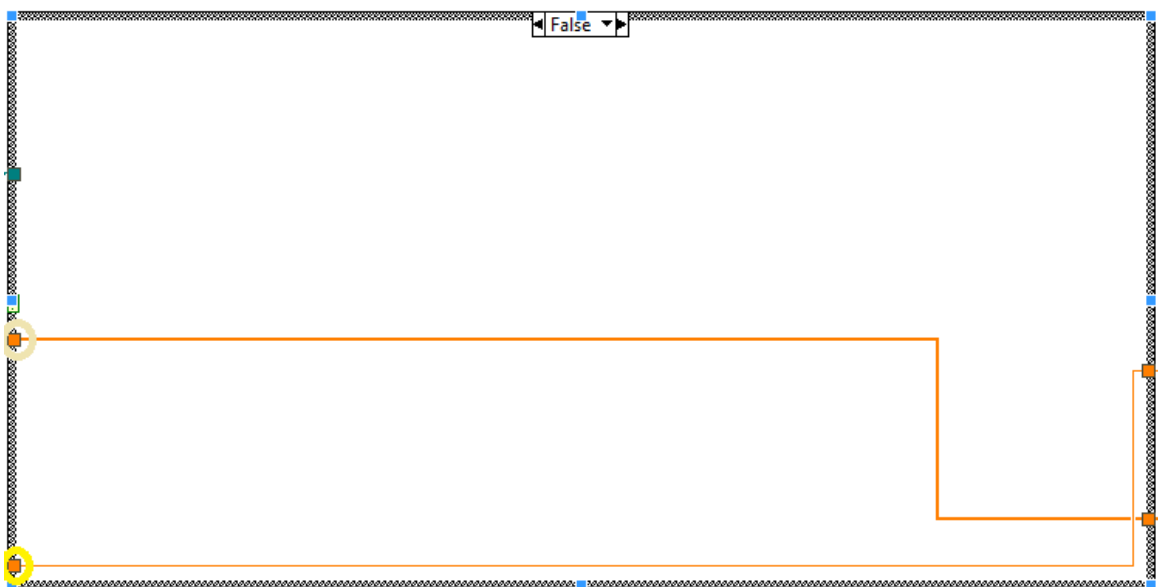


Fig. 3.16 Estructura para medias temporales por minuto cuando el estado es false o desactivado

Como podemos ver en la Fig. 3.15 marcado con un círculo gris, llega el vector con las diferentes medidas de potencia, a las cuales se le hace la media con un bloque MEAN de Labview. Una vez la media está hecha, se convierte a string para poder escribirla en el fichero en cuestión. En el fichero de texto, por cada valor de media a lo largo de un día, se escribe una línea que contendrá la hora y el minuto en el que ha sido realizada y el valor

de la media en formato 'HHMM', ' ', 'valor de la media';. Una vez se realiza la media se resetea el shift register del vector de medias correspondiente para que en la siguiente iteración no contenga valores ya utilizados. En la parte superior de la Fig. 3.15, podemos ver como se detecta el cambio de día comprobando la hora en la que nos encontramos. Si hay cambio de día, se resetea la cola de valores de medias para que, en la siguiente iteración, no se incluyan valores ya almacenados.

En la Fig. 3.16, podemos ver marcado en círculo amarillo, el shift register con valor 0 comentado anteriormente. Este shift register se utiliza porque, mientras no se produzca la primera media temporal, necesitamos poner a 0 el valor para indicar que todavía no existen valores en la gráfica de medias temporales de la interfaz de usuario.

Cuando todas las condiciones se satisfacen, se procede a escribir en el fichero mediante el bloque *Write to Text File Function* de LabView y, posteriormente, se cierra el fichero.

3.5 Estructura de media temporal horaria

El segundo bloque de lógica a comentar es el de las medias temporales horarias. Este bloque está formado por la siguiente estructura.

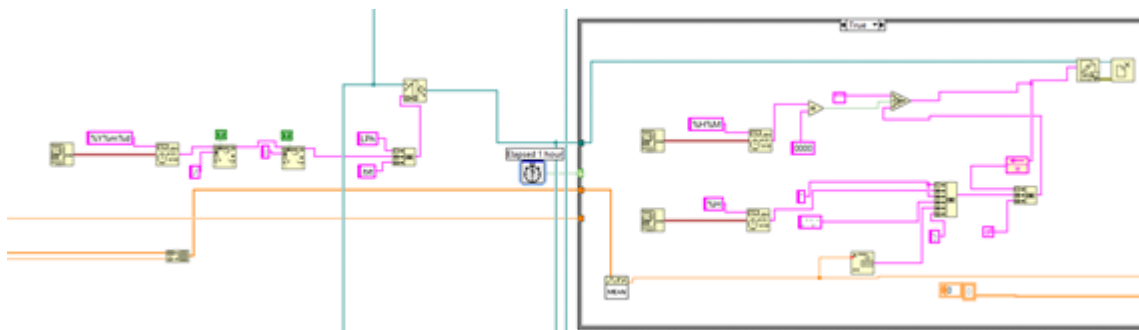


Fig. 3.17 Estructura de media temporal horaria

De la misma forma que en el caso anterior, tenemos las dos partes a destacar. La primera es la generación del fichero donde escribir las medias. La lógica es la misma que en el caso anterior, lo único diferente es que, a la hora de generar el nombre del fichero, se creará bajo el formato LPhyyyymmdd, es decir, se incluirá una 'h' para indicar que se trata de las medias temporales horarias.

En cuanto a la lógica de escritura del fichero es idéntica a la anterior. Sólo que esta se activará cuando el temporizador de 1 hora marque que el tiempo ha pasado mientras que en el caso anterior era de 1 minuto. Además, las líneas del fichero solo indicarán la hora en la que se realizó la media temporal sin indicar los minutos, puesto que solo habrá una medida cada hora. De esta forma, el formato de cada fila del fichero será 'HH', ' ', 'valor de la media';.

3.6 Estructura de media temporal diaria

De la misma forma que en los casos anteriores, se trabajará bajo un bloque de la misma lógica. En este caso, es el siguiente.

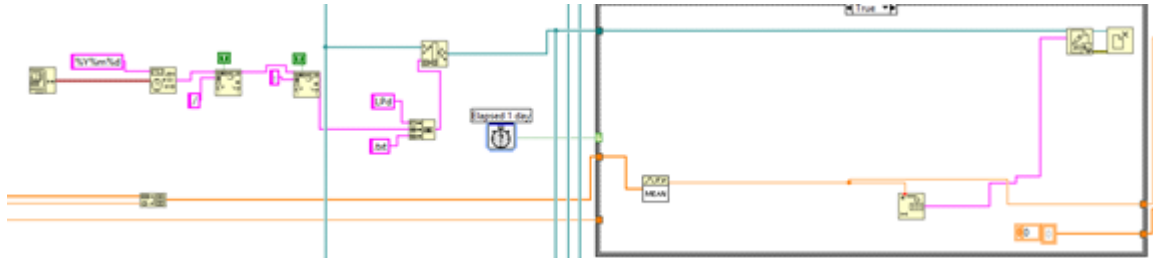


Fig. 3.18 Estructura de media temporal diaria

En cuanto a la generación del fichero, en este caso se crearán bajo el nombre LPdyyyyymmdd, de la misma forma que en el caso anterior, la 'd' sirve para diferenciar que se trata de medias diarias.

En cuanto a la escritura del fichero, la lógica es algo más sencilla. Cuando el temporizador de un día indique que el tiempo ha pasado, se activará la estructura, realizará la media diaria y escribirá el valor directamente en el fichero. Una vez escrito, se cerrará.

3.7 Estructura de escritura del fichero diario

Esta estructura se encarga de, una vez lleguen las 23:59:59 de cada día, generar un fichero que contenga todas las medias temporales obtenidas en ese día, englobando bajo variables diferentes el contenido de cada uno de los ficheros de medias temporales bajo el siguiente formato.

```
LP20180424= [ ...
    '0850', ' ', '01275'; ...
    '0851', ' ', '01276'; ...
    '0852', ' ', '01274'; ...
];

LPh20180424= [ ...
    '08', ' ', '01275'; ...
    '09', ' ', '01280'; ...
];

LPd20180424= '01279';
```

Fig. 3.19 Fichero diario de las medias temporales

Para la generación de estos ficheros se utilizará la siguiente estructura.

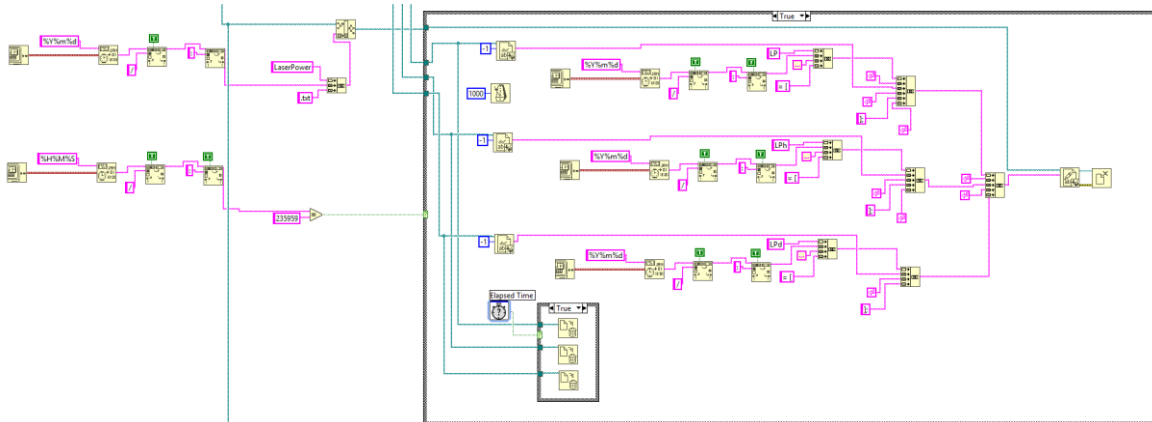


Fig. 3.20 Estructura de escritura del fichero diario

En este caso, también se dividirá en dos imágenes para poder observar de una forma más sencilla. En la siguiente figura se muestra la primera parte.

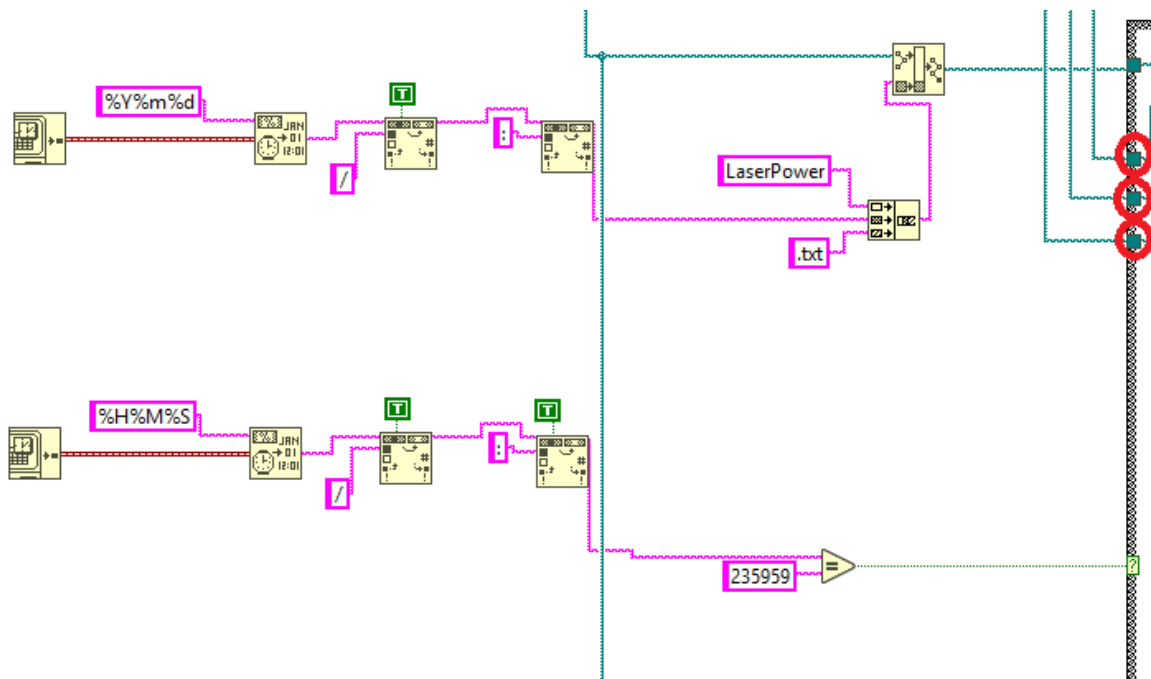


Fig. 3.21 Generación del fichero diario y activación de la estructura

En este caso, la generación del fichero sigue la misma lógica que en los casos anteriores, solo que el fichero se generará bajo el nombre LaserPoweryyyyymmdd.txt y contendrá la estructura mostrada en la Fig. 3.19. Además, se añade la lógica de activación de la estructura en el momento en el que la hora sea las 23:59:59. En círculos rojos se marca que, como parámetros de entrada, esta estructura, tiene la ruta de los diferentes ficheros de las medias temporales. Esto es porque se accederá a ellos dentro de la estructura para poder construir el fichero diario como se puede ver en la figura siguiente.

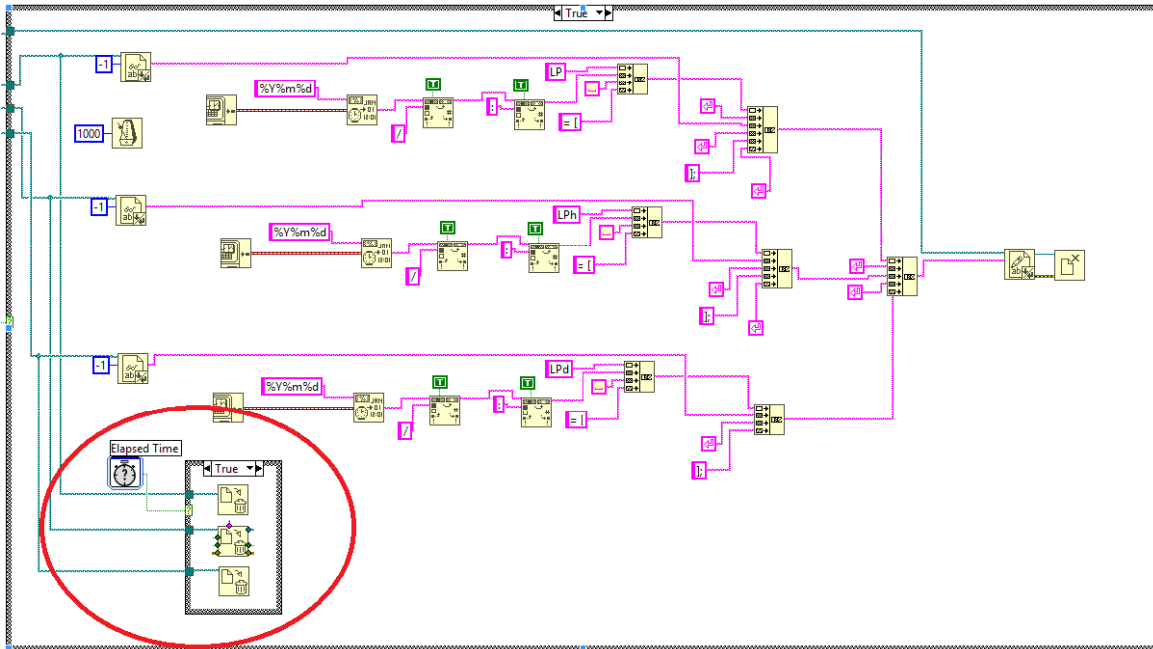


Fig. 3.22 Escritura del fichero diario

En cuanto a la escritura del fichero, se sigue una lógica idéntica a la de los ficheros individuales generados anteriormente, solo que dándole el formato deseado en la Fig. 3.19. En círculo rojo se destaca la eliminación de los ficheros individuales generados en las estructuras anteriores. Esto es porque ya no se necesitan dado a que los valores están escritos en el fichero diario. Cabe destacar que antes de eliminar estos ficheros, hay que esperar un tiempo mínimo, porque sino se eliminarían los ficheros antes de ser leídos completamente.

3.8 Estructura de media temporal semanal

De la misma forma que en la generación de los ficheros de medias por minuto, horaria y diaria, se genera el fichero de las medias semanales con la siguiente estructura.

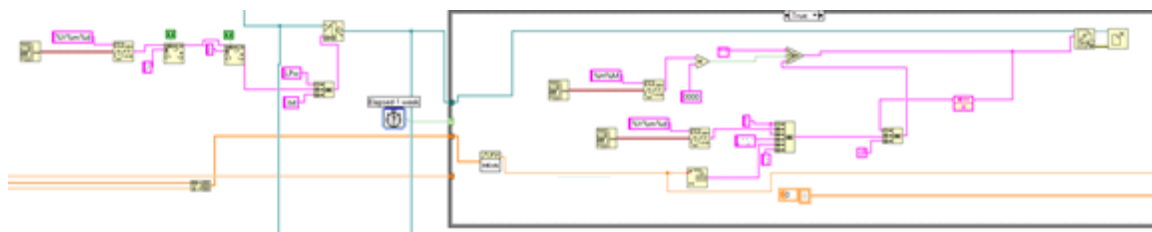


Fig. 3.23 Estructura de generación del fichero semanal

Para este caso no hay mucho nuevo a destacar, lo diferente al resto de los casos mencionados es lo mismo. En este caso el nombre del fichero generado será LPwyyyymmdd para diferenciar que se trata de medias semanales y las filas escritas en el fichero de texto contendrán los datos en el formato 'yyyymmdd', ' ', 'valor de la media';.

3.9 Estructura de media temporal mensual

Para este caso, la dificultad está en la lógica que hay que generar para la creación de los ficheros, así como la interacción de los timers entre ellos. Primero pasaré a comentar la estructura de escritura de los ficheros dado que es la misma para los 4 casos y, posteriormente, pasaré a explicar la lógica de la generación de ficheros para cada caso.

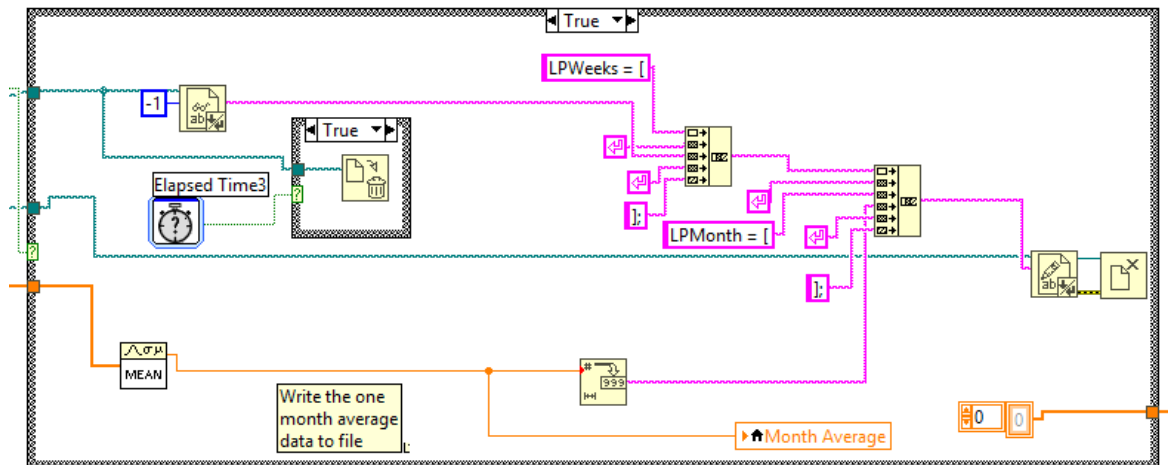


Fig. 3.24 Estructura de escritura del fichero mensual

De una forma parecida a la escritura del fichero diario, esta estructura recibe como parámetro de entrada el fichero de las medias semanales, genera una estructura como la de la Fig. 3.19 pero con las medias semanales y la media mensual. Elimina el fichero de las medias semanales y escribe el fichero mensual.

3.9.1 Generación y activación de la estructura de un mes de 30 días

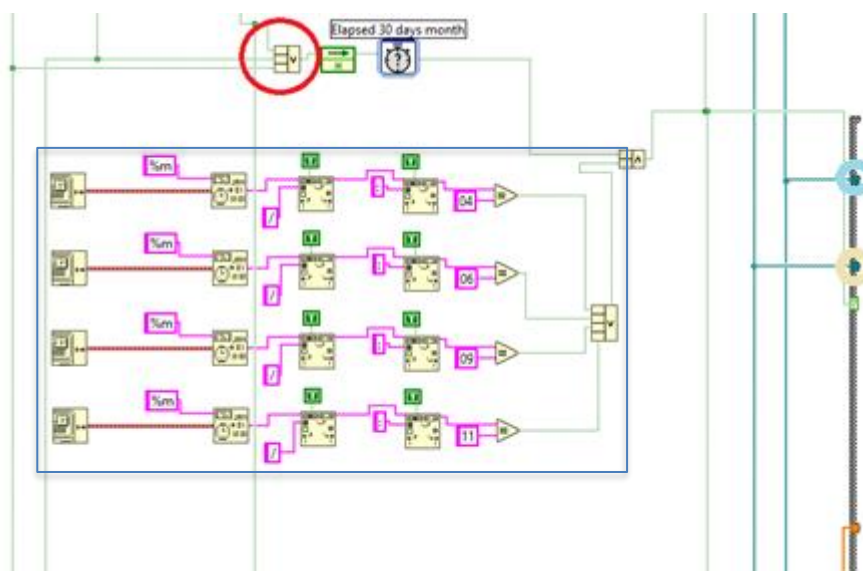


Fig. 3.25 Generación y activación de la estructura de un mes de 30 días

En el rectángulo se marca la lógica que se encarga de comprobar que nos encontramos en un mes de 30 días, la puerta OR de 4 entradas comprueba que estamos en el mes de abril, junio, septiembre o noviembre. Si se cumple la condición, manda un true hacia la puerta AND que debe verificar si el temporizador de un mes de 30 días ha mandado true. Aquí es donde entra la lógica de interacción entre temporizadores mensuales. Los 4 temporizadores deben estar conectados entre sí, de tal forma que, si una de las 4 condiciones AND mensuales se cumple, debe reiniciar el temporizador del resto de casos mensuales. Si esto no se hiciera, se guardarían medidas erróneas.

En círculo gris se marca la ruta del fichero mensual que entra como parámetro a la estructura, mientras que la ruta del fichero semanal se marca en círculo azul.

3.9.2 Generación y activación de la estructura de un mes de 31 días

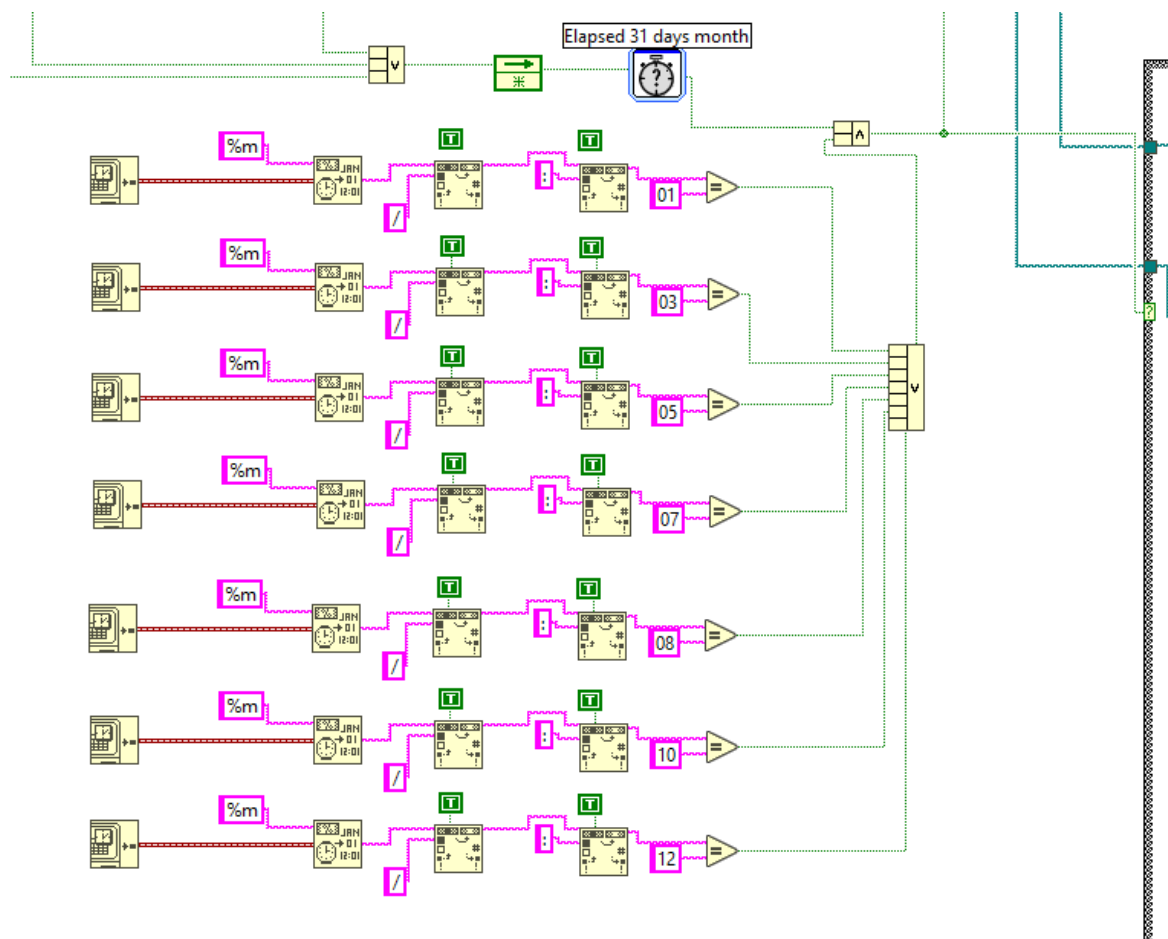


Fig. 3.26 Generación y activación de la estructura de un mes de 31 días

Este caso funciona de la misma forma que el anterior, las diferencias están en hay que comprobar que el mes en el que nos encontremos pertenezca al grupo de los meses de 31 días, además el temporizador tendrá el valor equivalente a estos.

3.9.3 Generación y activación de la estructura de febrero

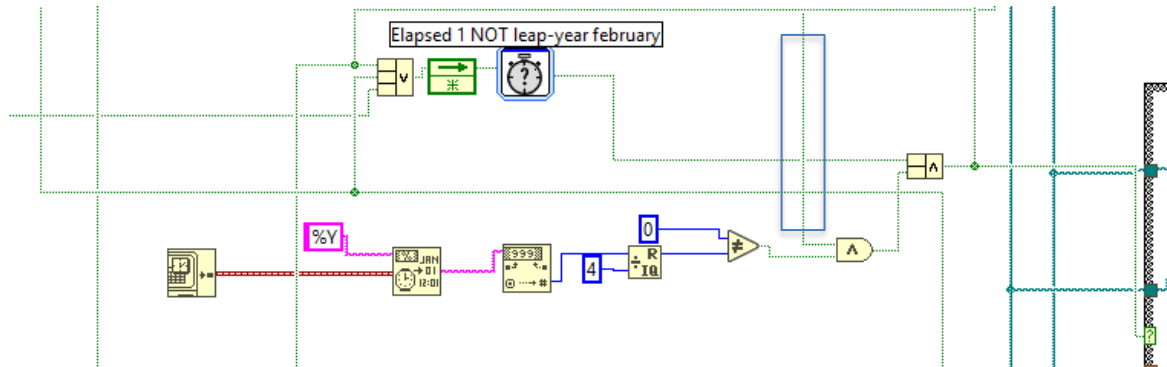


Fig. 3.27 Generación y activación de la estructura de febrero

Igual que en los casos anteriores la lógica es la misma, salvo que aquí solo hay que comprobar que nos encontramos en el mes de febrero (marcado con el rectángulo, viene del bloque que se explicará a continuación) y en un año que no sea bisiesto. Esto se hace mediante la obtención del año en el que nos encontramos y comprobando que, al dividirlo entre 4, el residuo no sea cero. Además, el temporizador tiene un tiempo equivalente al mes de febrero.

3.9.4 Generación y activación de la estructura de febrero bisiesto

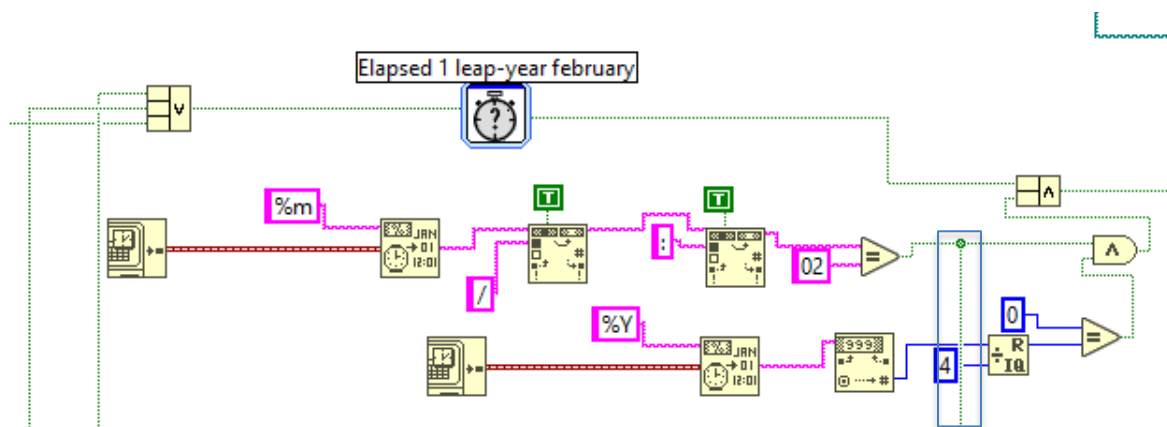


Fig. 3.28 Generación y activación de la estructura de febrero bisiesto

Igual que en el caso anterior, hay que comprobar que nos encontramos en el mes de febrero (marcado en rectángulo la línea de comprobación que va hacia la estructura anterior) y en un año bisiesto. En este caso hay que hacer el caso contrario al anterior. Además, el temporizador tiene el tiempo equivalente a un mes de febrero bisiesto.

4. Generación del gráfico de medias temporales

Una vez realizadas todas las medias de forma correspondiente, es hora de mostrarlas en una gráfica conjunta. Para ello haremos uso del bloque *Merge Signals Function* que tiene la siguiente forma.

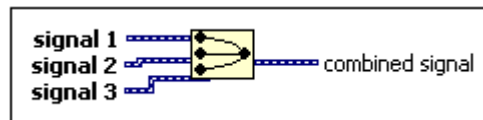


Fig. 4.1 Bloque Merge Signals Function de LabView

La conexión de este bloque con el gráfico de medias temporales quedará de la siguiente forma.

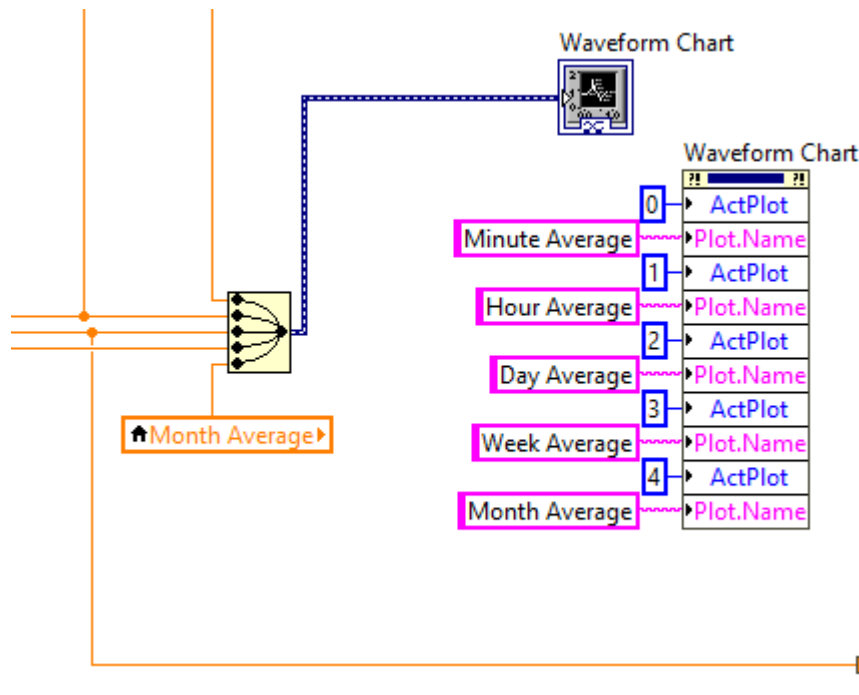


Fig. 4.2 Generación del gráfico de medias temporales

Como podemos observar todas las medias temporales acaban unidas en una misma señal y mostradas en la gráfica bajo el nombre Waveform Chart. Además, se ha añadido una leyenda para poder diferenciar las medias temporales fácilmente. En la interfaz de usuario queda de la siguiente forma.

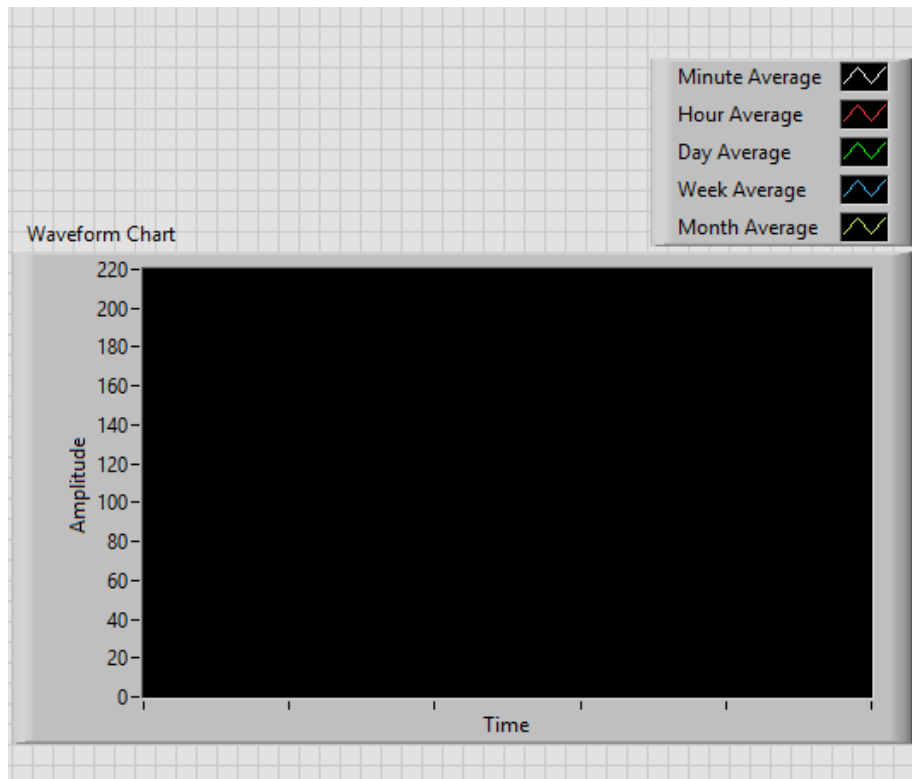


Fig. 4.3 Gráfica de las medias temporales en la interfaz de usuario

Como podemos ver, las últimas medias temporales por minuto se mostrarán en blanco, mientras que las horarias en rojo, las diarias en verde, las semanales en azul y las mensuales en amarillo. De esta forma no habrá problemas para reconocer cada una y no habrá que acceder a los ficheros para comprobar los últimos valores de cada una.

5. Conclusiones

Como primer contacto con la aplicación de LabView, he de decir que salgo bastante satisfecho en cuanto a las posibilidades que ofrece y el rendimiento que tiene. Es difícil adaptar la lógica de programación de otros lenguajes como C# o Java a una programación gráfica, pero una vez desenvuelto, empiezas a ver que es una forma fácil de trabajar y bastante potente.

Una de las motivaciones de entrar a este proyecto fue precisamente tomar contacto con esta aplicación en el desarrollo de un proyecto real y consideraba que no había mejor oportunidad que aprovechar el trabajo de fin de grado. Posiblemente, este trabajo se podría haber realizado de una forma óptima, pero, como usuario principiante, estoy bastante satisfecho con los resultados obtenidos.

Ahora, se ofrece la posibilidad de tener un control mayor sobre el láser automatizando las medidas de potencia. Mediante el uso de una interfaz de usuario simple y amigable, se

pueden observar y almacenar los datos de potencia emitidos por el láser, lo que hace tener control, en todo momento, del estado actual y pasado del láser, así como detectar posibles anomalías o fallos en su funcionamiento.

Bibliografía:

- [1] National Instruments. Foro de discusión. Disponible en: <https://forums.ni.com/t5/LabVIEW/bd-p/170>.
- [2] National Instruments. Ejemplos de LabView. Disponible dentro de LabView: Help - > find examples.