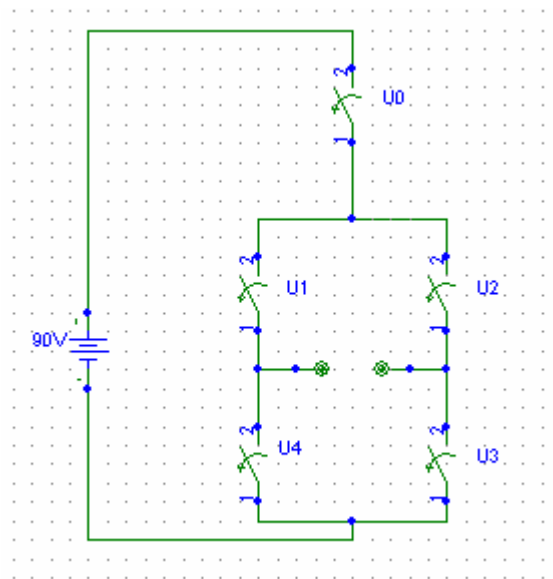


**CAPITULO 3 LabVIEW®**

**3.1 Introducción**

En el presente capítulo se da una breve revisión de las características de LabVIEW®. Se citan las características de la tarjeta de adquisición de datos, y se describe su proceso de instalación. Se explica también el funcionamiento del programa realizado y sus necesidades de operación, es decir, la secuencia que se debe cumplir al hacer operar un puente H con relevadores como dispositivos de apertura y cierre.



**Figura 3.1 Puente H con Seguridad**

En la figura 3.1 se ejemplifica como es el circuito diseñado para el proceso del control de giro en el motor; el relevador U0 se abre para que los relevadores, ya sea el par de los relevadores U1 y U3 o el par de los relevadores U4 y U2 se abran o se cierren, el relevador U0 se cierra para poder dejar pasar el voltaje suministrado al motor, una vez que las secuencias de los relevadores ha sido realizada. De esta forma se controla giro del motor para elevar el plano y se protege a la fuente de una sobrecarga.

Así mismo, en el presente capítulo se muestra la interfaz resultante para el usuario, es decir, el panel frontal de LabVIEW®.

### 3.2 LabVIEW®

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. Labview permite diseñar interfaces de usuario mediante una consola interactiva basada en software. Tiene la ventaja de que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos.

Al diseñar programas con LabVIEW® se trabaja siempre bajo algo denominado VI, es decir, un instrumento virtual. Se pueden crear VI a partir de especificaciones funcionales que sean diseñadas por el programador. Existen dos paneles, el panel frontal y el panel de programación ó diagrama de bloques; en el panel frontal se diseña la interfaz con el usuario y en el panel de programación se relacionan los elementos utilizados en la interfaz mediante operaciones que determinan en sí como funciona el programa o el sistema. [12]

### 3.3 Características de la DAQ USB-6008

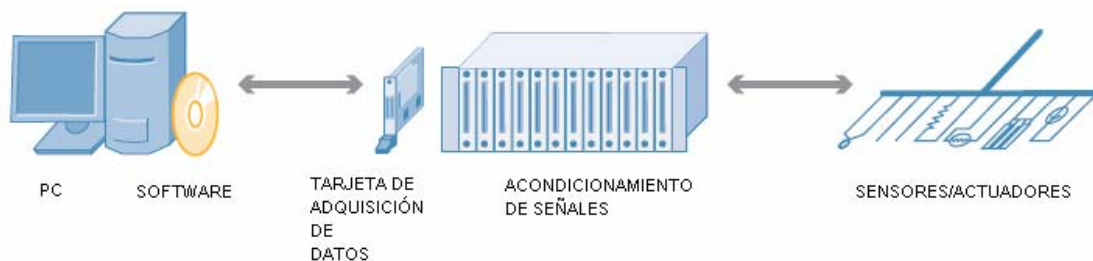
La tarjeta DAQ USB-6008 proporciona la adquisición de datos de forma confiable para proyectos de bajo presupuesto vía puerto USB. Es posible utilizar NI-DAQmx-DAQmx para desarrollar modificaciones particulares como requisitos para la adquisición de datos con los instrumentos de LabVIEW. Las entradas a la DAQ USB-6008 tienen terminales ajustables con tornillos para conectar fácilmente la señal, teniendo flexibilidad adicional al manejar configuraciones múltiples de cableado. El uso común es para proyectos donde sean esenciales la economía, el tamaño pequeño, y la simplicidad, como por ejemplo, el registro de datos y el uso académico en el laboratorio.

De acuerdo al apéndice B algunas de las características de la tarjeta DAQ-6008, son que la velocidad de las entradas analógicas es de 10K/muestras por segundo. Con respecto a la velocidad de las entradas digitales, esta es dada por la velocidad seleccionada desde el programa. La tarjeta trae un contador que permite contar pulsos

con una velocidad máxima de 5Mhz, se observó que la velocidad de operación de la DAQ-6008 baja considerablemente respecto a la DAQ-6024. Al realizar la adquisición de datos, en conjunto con las acciones de control desde el panel frontal, fue necesario reducir el ancho de banda de 10K para utilizar el programa, ya que no era suficiente el tiempo de procesamiento. La razón de esto es debido a que ésta es una tarjeta de bajo costo por lo que tiene un multiplexor que comparte el amplificador y ADC con todos los canales. La velocidad máxima de adquisición será 10K entre el número de canales ocupados, así que si solo se ocupa un canal se podrá adquirir a 10K, si se ocupan los 8 canales la máxima velocidad será de 1.25 K, para el caso de éste control se ocupan 4 canales y la velocidad resultante es de 2.5K. La capacidad de salida de voltaje es de 0 a 5V, corriente de salida de 5mA y cuenta con doce canales (8 para el puerto 0 y 4 para el puerto 1). Para entrada de voltaje acepta rangos de hasta  $\pm 20$  V, en modo diferencial.

Al considerar el DAQ USB de bajo costo, es porque probablemente no se requiere muestreo de alta velocidad o temporización ni sincronización avanzadas. Sin embargo, la presencia o ausencia de ciertas características pueden hacer la diferencia en la exactitud, confiabilidad y uso del dispositivo de medición basado en USB

En general, cuando se habla de un sistema de DAQ basado en una PC, se pueden identificar 5 partes principales: la PC, el software, hardware de adquisición de datos, acondicionamiento de señales y los sensores. Las características de cada uno de estos componentes determinarán en parte el desempeño adecuado del sistema. Las especificaciones de la entrada analógica sirven para determinar tanto las capacidades como la precisión del dispositivo de DAQ. Las especificaciones más básicas son: el número de canales, la máxima velocidad de muestreo, la resolución y el rango de entrada [11]



**Figura 3.2 Adquisición de Datos**

### 3.4 Descripción del programa

El programa para la parte de control (Figura 3.3), esencialmente consta de dos secciones. En la primera sección se inicializan los valores del arreglo booleano tanto en el puerto físico como en la variable local Array. En la segunda sección mediante una estructura de eventos se está monitoreando cuándo se presionan los botones de Subir, Bajar, Detener y Salir. En el caso “Subir” “Bajar” se corroborará mediante dos estructuras de casos concatenadas que sólo uno de los botones subir o bajar se haya presionado. Esto es, si el valor booleano de “Subir” está en verdadero y el de “Bajar” está en falso (que es el primer par de estructuras de casos concatenadas) lo que se hace es mandar al puerto el valor de 0, se tiene una pausa de 100ms y se manda al puerto el valor 101; lo mismo sucede en el caso contrario sólo que ahora se manda el valor 011 al puerto, en cada una de estas operaciones se actualiza el valor presente en el puerto en la variable local Array.



**Figura 3.3 Panel de control**

Para el evento que monitorea el botón Detener se manda un 0 al puerto y se actualiza el valor en la variable justo como en la primera sección del programa. El último evento que se monitorea es el cambio de estado del botón de Stop el cual manda a llamar a la función Stop la cual aborta la ejecución del programa. Todo este monitoreo

de los botones del panel frontal se realiza mediante un ciclo while con una pausa de 1ms entre cada una de sus iteraciones y cuya condición de paro se le ha puesto una constante de tal manera que la única manera de parar la ejecución del programa es mediante el evento del botón Stop. Esto es en general lo que está realizando el programa, cambiando el valor del puerto digital de la tarjeta según se vayan dando los eventos de cambio de valor de los botones en el panel frontal.

De acuerdo a las necesidades de nuestros actuadores y de los tiempos que necesitan para su funcionamiento, es como se decide realizar el programa. En primera instancia se observa que se necesitan para controlar el motor 3 señales de control; una para activar y desactivar el relevador primario, y dos señales más para habilitar a los otros dos grupos de relevadores.

No son necesarias 4 señales de control ya que lo que se controla son relevadores de un puente H, y como la intención es el abrir dos relevadores y cerrar los otros dos, se utiliza la misma señal para un par de relevadores y la otra señal para el par restante. Es por eso que sólo se requieren de tres señales de control. También son necesarios cuatro botones: “Subir”, “Bajar”, “Detener” y “Salir del programa”. Cada botón deberá tener prediseñada su tarea a realizar.

El circuito diseñado abre al interruptor correspondiente del puente H cuando se aplica un voltaje de 5 Volts, es decir un “1” lógico y cierra al interruptor cuando no existe voltaje en la salida de la tarjeta o un “0” lógico. Al iniciar el programa se cuenta con la condición inicial de las salidas de forma que manda la secuencia “1 1 1”, ya que de no ser así, el motor comenzaría a trabajar fuera de la voluntad del usuario. De esta forma todos los relevadores quedan abiertos y no se tiene el riesgo de una sobrecarga en el sistema al no tener carga en el control. Para subir el plano se necesita primero una secuencia de “1 1 0”, para que se mantenga abierta la fuente y se hagan los cambios en los relevadores, después de un tiempo de un segundo, para realizar el cambio de cierre de los relevadores correspondientes, el programa manda la combinación “0 1 0”, el relevador que deja pasar la carga al circuito se cierra comportándose como corto circuito y mandando el voltaje a los relevadores cerrados adecuadamente.

De igual forma, para bajar el plano se necesita un cambio de secuencias automático es decir, que al presionar el botón de bajar, se mande una secuencia de “1 0 1” y al esperar un segundo se mande la secuencia “0 0 1” para controlar en cierre y apertura de los otros relevadores. Para detener el plano se necesita una secuencia de “1 1 1” para abrir el relevador que controla al paso de la fuente.

El punto que se debió resolver es que la fuente debe permanecer abierta mientras se realicen los cambios de apertura y cierre respectivos para no crear un corto hacia la fuente y hacia el motor. Se consideraron dos soluciones de programación; el uso de una estructura de un diagrama de estados llamada “*state diagram structure*” o una estructura llamada “*case structure*”. Ambas opciones se encuentran en la librería de LabVIEW®.

Las estructuras son representaciones gráficas de los lazos y de las declaraciones de caso de lenguajes de programación basados texto. Son utilizadas para repetir bloques de código y para ejecutar código condicional en un orden específico. Como otros nodos, las estructuras tienen terminales que las conectan con otros nodos del diagrama a bloques, se ejecutan automáticamente cuando los datos de entrada están disponibles, y proveen datos a las terminales de salida cuando la ejecución termina. Cada estructura tiene un límite distintivo, redimensionable para incluir la sección del diagrama de bloque que se ejecuta según las reglas de la estructura. [13]

Las estructuras utilizadas son la estructura case y la de secuencia. La estructura de secuencia automáticamente va cambiando los valores. En cuanto termina de ejecutar un cuadro pasa al siguiente. El cuadro de en medio tiene un retraso de aproximadamente de 1 segundo, pues es el tiempo requerido para asegurar que los relevadores se abran o se cierren, a pesar de que el tiempo en que los relevadores tardan en cerrar es de 10ms y el tiempo en que tardan en abrir es de 5ms por lo que aproximadamente un segundo de retardo se garantiza una buena coordinación de los elementos mecánicos. Se realizaron pruebas con un retardo de 100ms y el sistema no presentó fallas sin embargo un tiempo de retardo mayor en el diseño incrementa la protección. Luego pasa al tercer cuadro.



**Figura 3.4 Reloj del ciclo While**

El reloj que está en el ciclo *while* de 1ms (Figura 3.4), sirve para dar oportunidad al procesador de que pueda realizar otras tareas y dejar de realizar las rutinas del programa, de no ser así LabVIEW acaparará todos los recursos del sistema.

Se utiliza solamente una *task* para todo el programa, pues el puerto de salida que utiliza en ambos casos es el mismo. Si fueran dos las salidas requeridas, se necesita un puerto para cada uno y por ende una *task* para cada uno. Al declarar la *task*, después de haber especificado el puerto digital y nombrado la *task* se seleccionó la opción que aparece "*Invert all lines in port*" de esta forma se manda el "1 1 1" que se necesita como condición inicial. Asegurándose de invertir también el resto de las señales (de 101 a 010 por ejemplo). Para que el funcionamiento del programa fuera el adecuado, se cambió la acción mecánica de los botones de la siguiente forma.

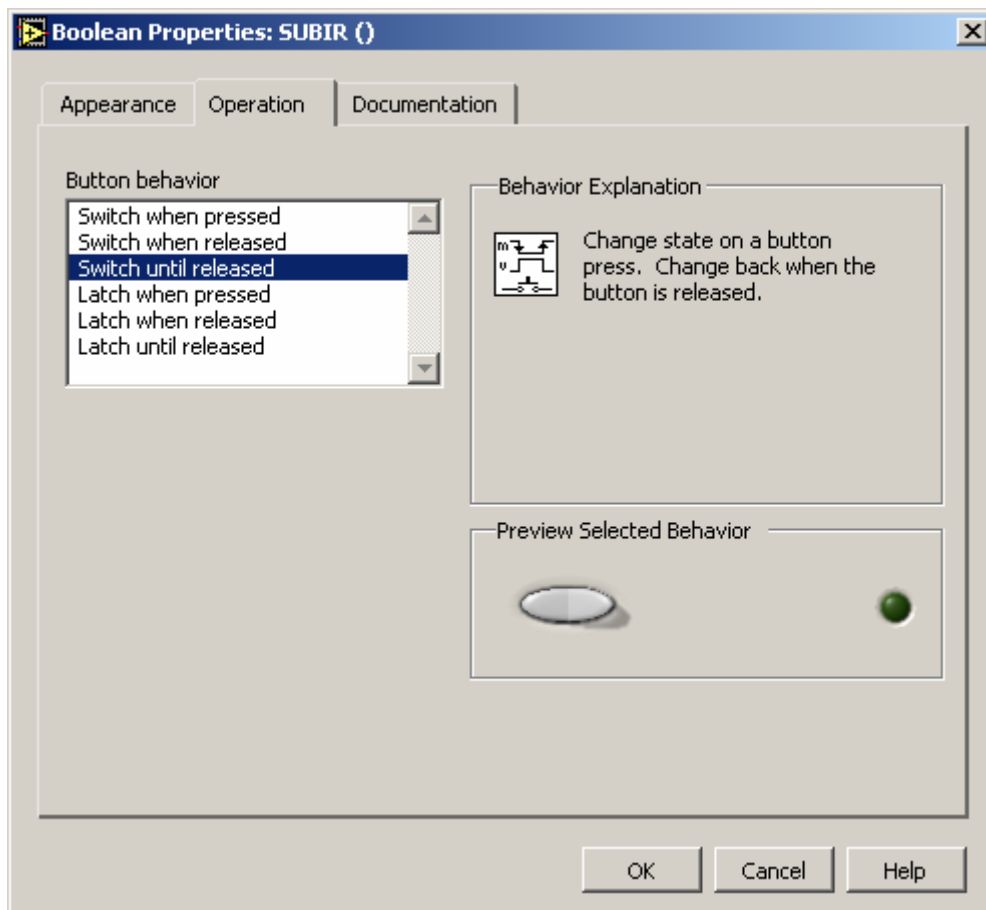


Figura 3.5 Comportamiento de los botones

El comportamiento del botón es que cambia de estado cuando es presionado. Regresa a su estado cuando es liberado y de esta forma se mantiene la secuencia en los relevadores.

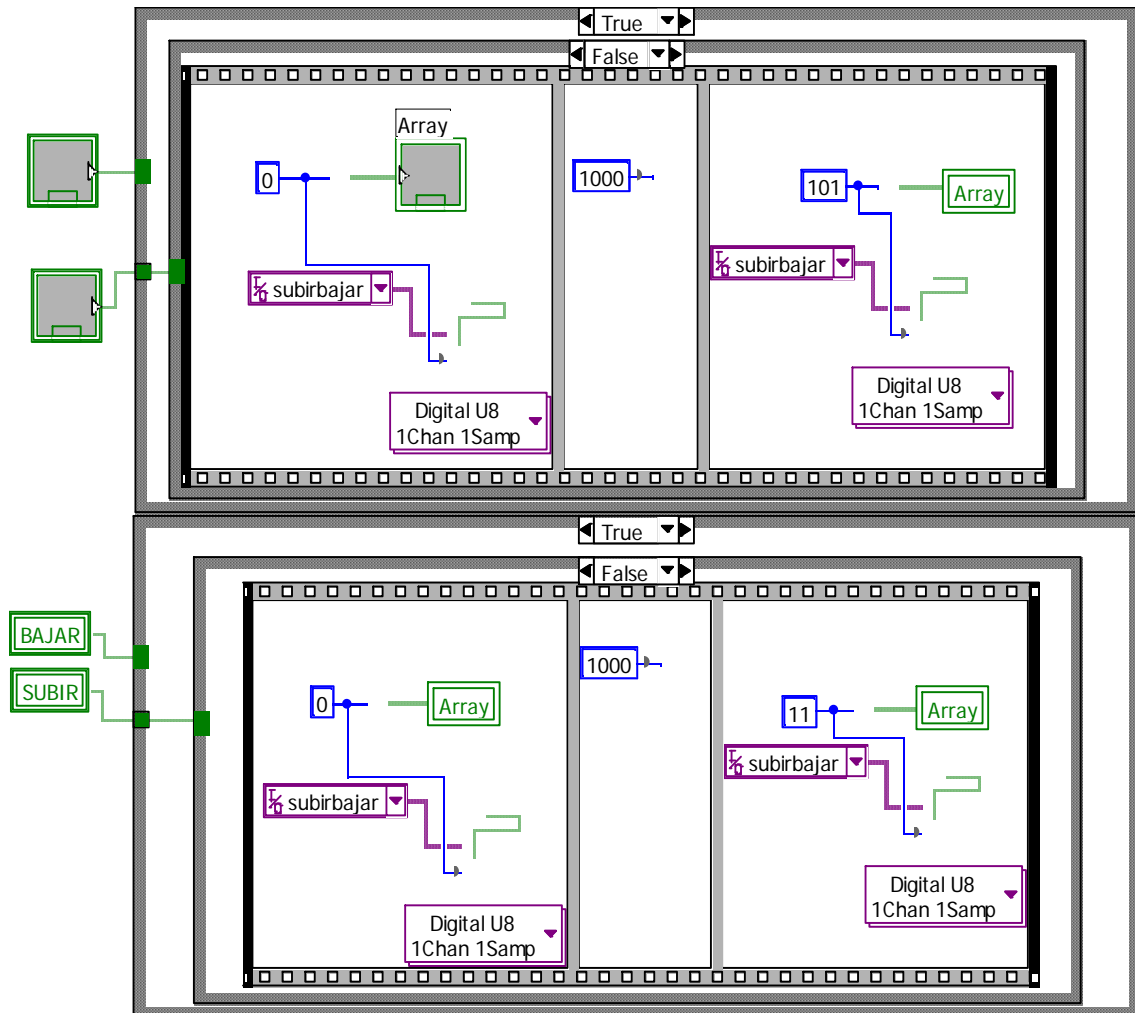


Figura 3.6 Secuencias Case y Event

En la figura 3.6 se tiene las cuatro secuencias del movimiento es decir en la parte superior izquierda de la figura se tiene en el arreglo una orden de salida de “0 0 0” pero por obvias razones el software lo abrevia como “0” es decir al presionarse el botón, lo que va a pasar es que a la salida del puerto se tendrá un “1 1 1” y como se había comentado, como se tiene lógica inversa, en lugar de cerrarse los relevadores, se abrirán todos no dejando pasar nada.



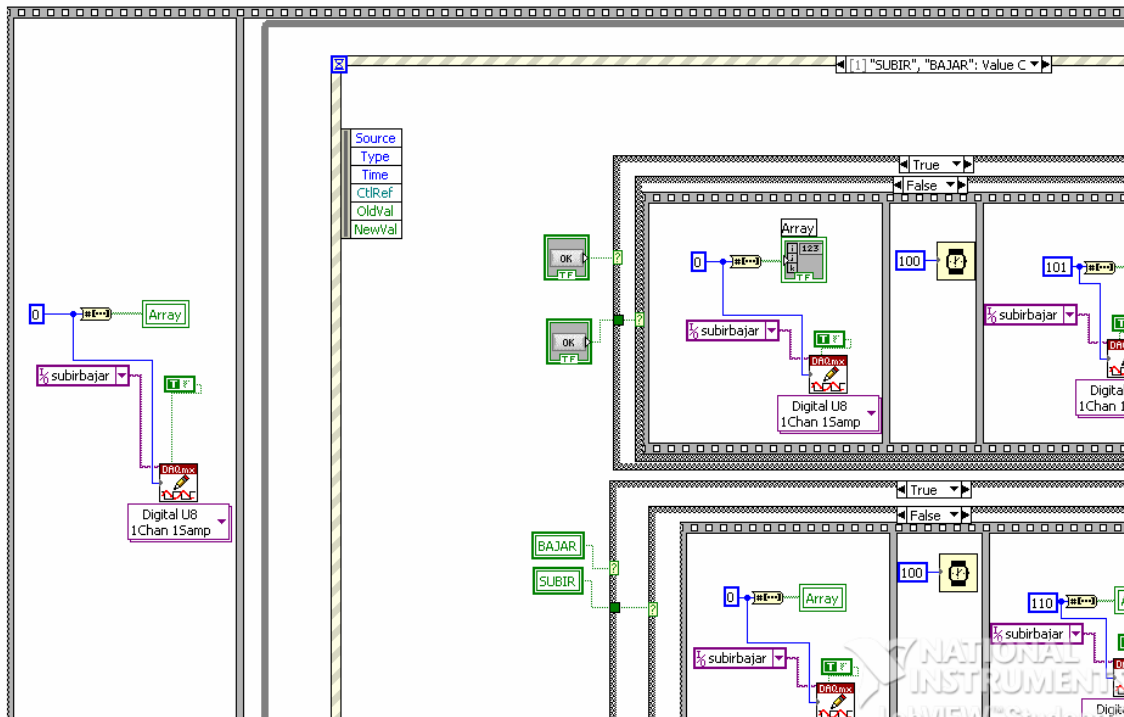


Figura 3.7 Evento 1

En la figura 3.7 se nota las secuencias del evento 1, es decir las secuencias de subir y bajar plano, así como las condiciones iniciales. En la figura 3.8 se tiene el caso del segundo evento considerando al botón de detener, que no es con el fin de detener el programa, sino detener el plano mandando la secuencia “0” es decir “1 1 1” si consideramos es lo que se necesita para detener el plano.

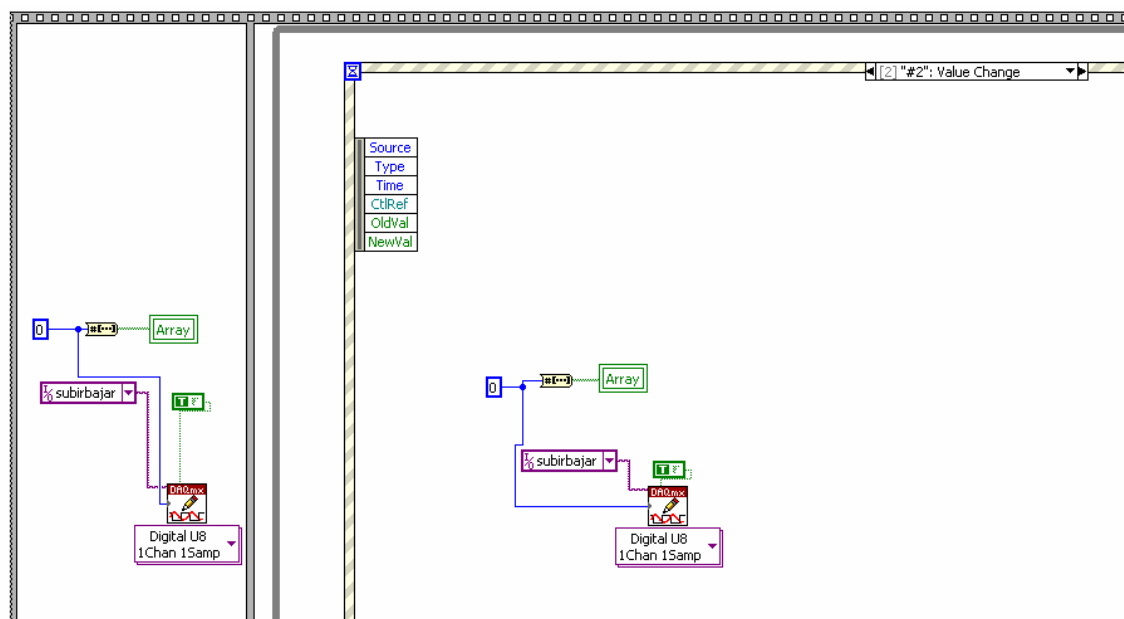


Figura 3.8 Evento 2

Como se mencionó el programa necesita de tres bits de control, uno para un solo relevador, y los otros para manejar a un par de relevadores cada uno. Para ejemplificar este comportamiento, se muestran en el panel frontal un arreglo que denota como es que van comportándose el programa al presionar los botones.

### 3.5 Instalación de la DAQ

Para conectar la nueva DAQ existen consideraciones que hay que tomar en cuenta; existen ahora operaciones tanto para la parte digital como para la parte analógica, para enviar las señales digitales a la DAQ, es necesario utilizar un bloque de programación llamado "Write to DAQmx", incluido en el programa, este bloque llama a una tarea, que se tiene que crear para realizar el trabajo, primero, en el "measurement and automation" de LabVIEW, que se instala junto con la DAQ, declarar una "task" para los botones. En esta, definir qué tipo de puerto usar (en este caso el puerto digital de 8 bits puesto que los números se están enviando en binario) y se debe poner un nombre a la task.

Para declarar una task:

1. Abrir el *Measurement and Automation Explorer* y localizar la DAQ (que debe estar conectada a la computadora) en "Devices and Interfaces" y luego *NI DAQ mx Devices*.

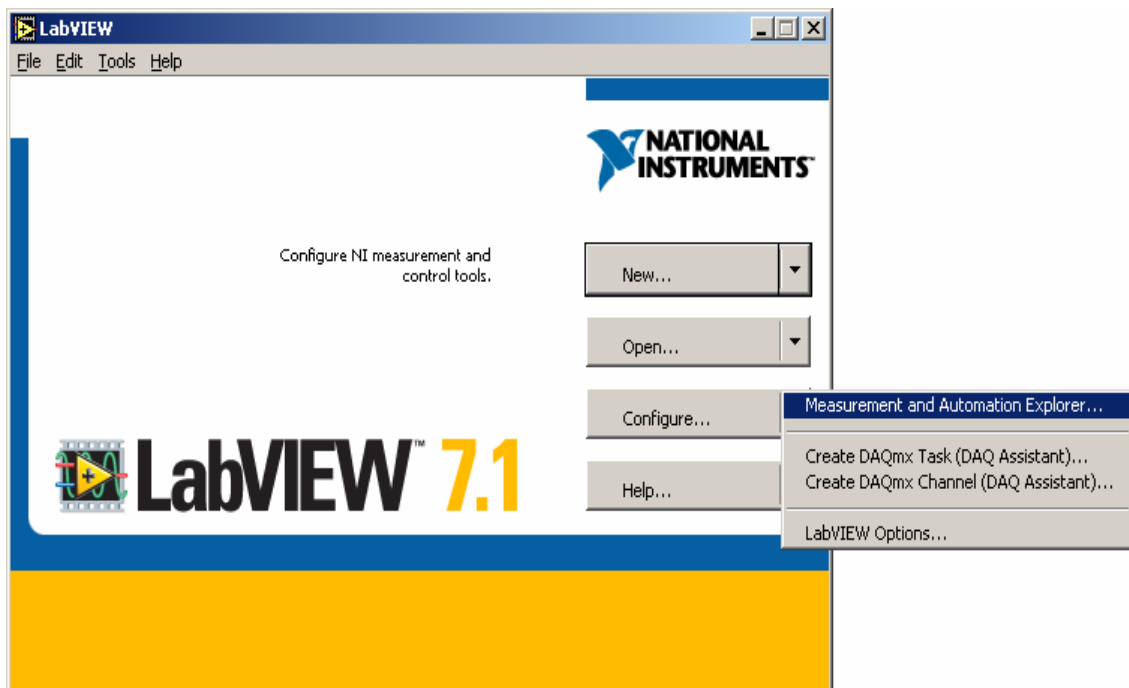


Figura 3.9 Measurement and Automation Explorer

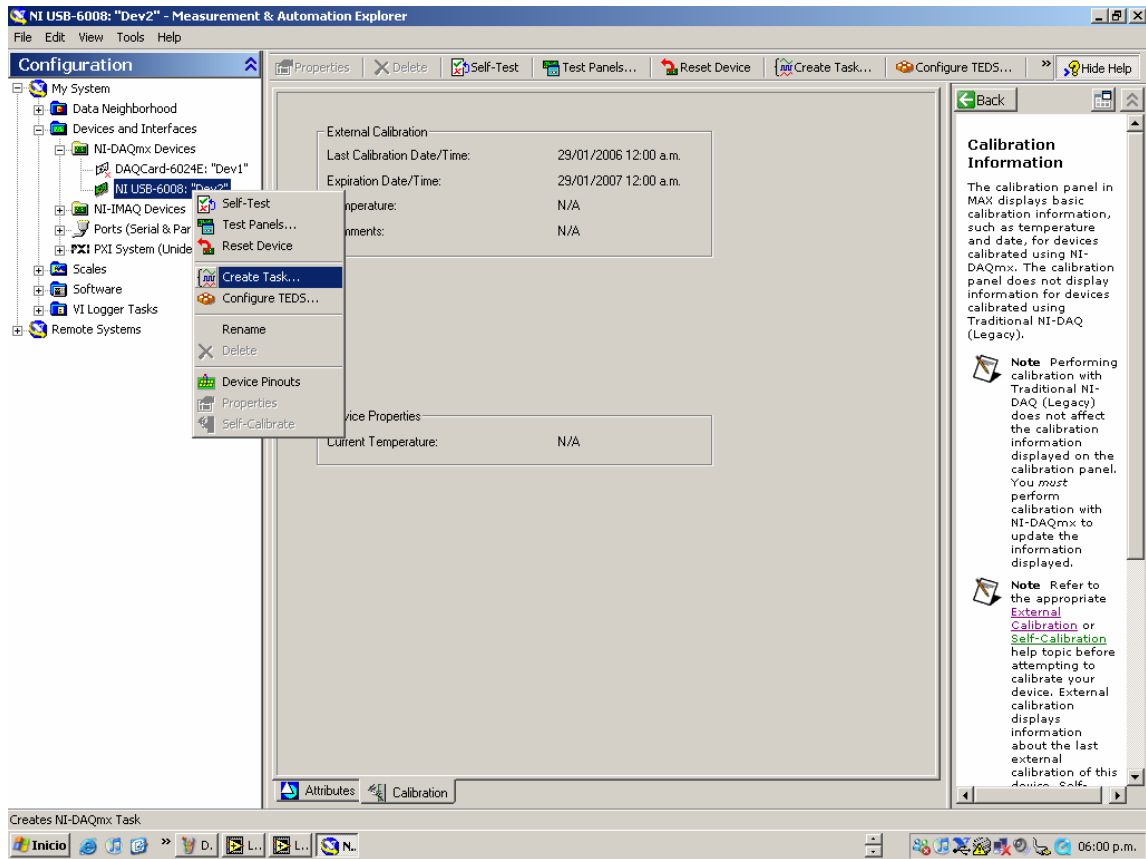


Figura 3.10 Asignar una tarea a la Tarjeta

2. Una vez que se haya localizado, se da click derecho en su nombre (si solo se tiene un dispositivo instalado éste debe ser "Dev 1", para este caso al realizar las pruebas, se tienen dos dispositivos. En la Figura 3.10 se muestra como "Dev 1" la tarjeta DAQ-6024 y "Dev 2" la tarjeta DAQ-6008) y seleccionar la opción "create task". (Figura 3.10). Posterior a este paso se muestran dos cuadros para seleccionar el tipo de salida o entrada que se quiere crear (Figuras 3.11 y 3.12). En este caso se seleccionó salida de puerto digital.

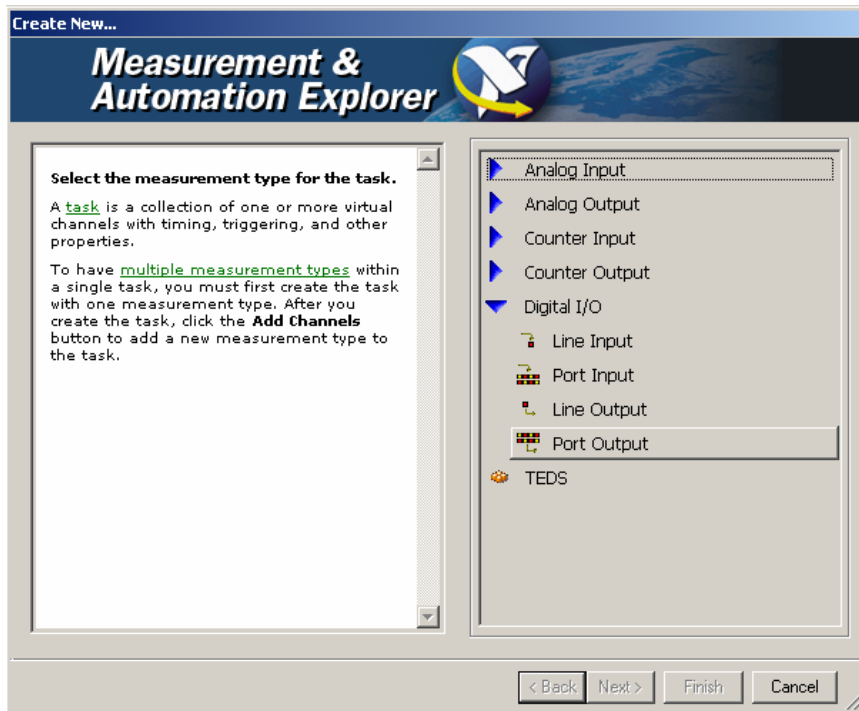


Figura 3.11 Selección de puerto de salida digital

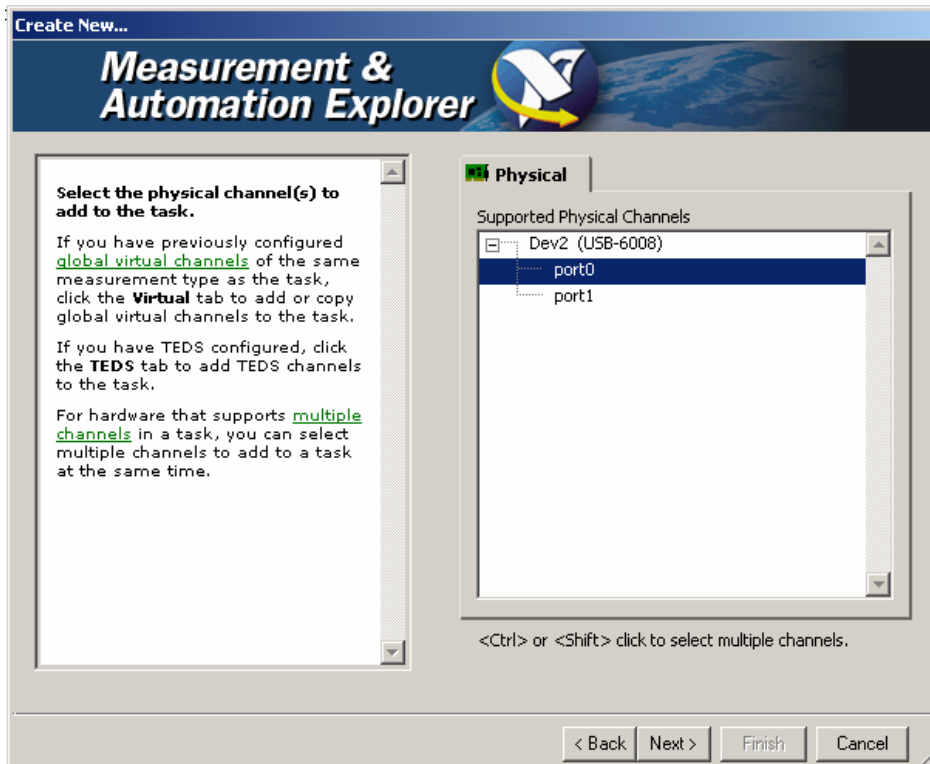
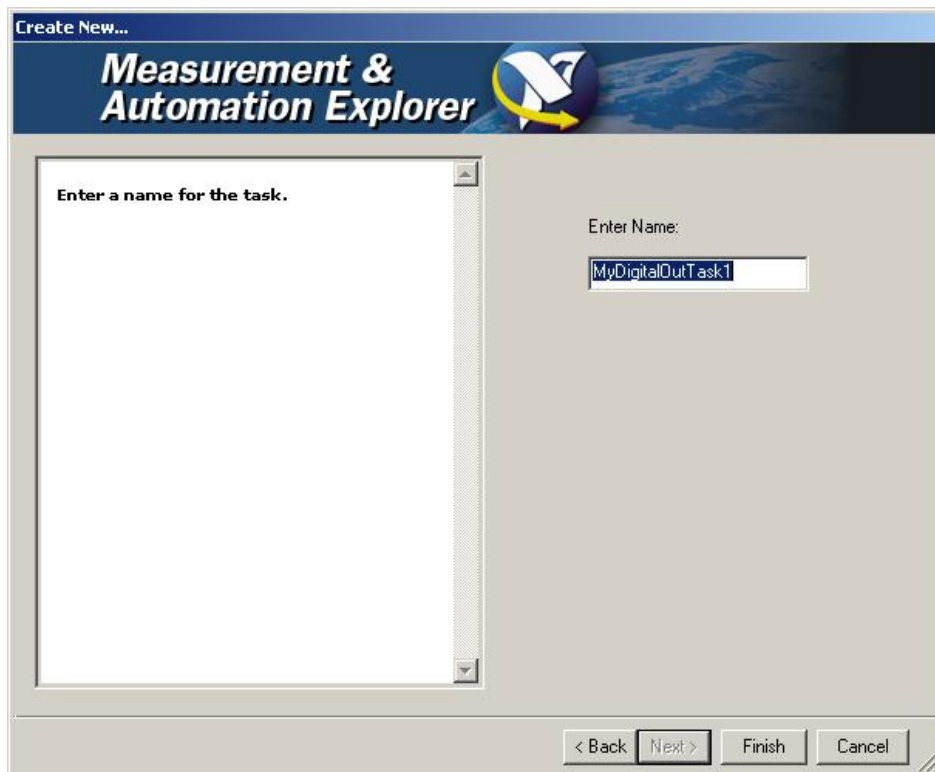


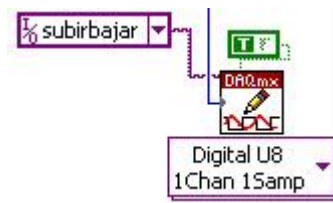
Figura 3.12 Selección del puerto de salida

3. Se selecciona si se quiere que sea salida o entrada y de qué tipo (Figuras 3.11-3.12), en este caso es salida digital para un puerto. Aparecerá un cuadro donde se nombra la *task* (Figura 3.13) Luego de haber seleccionado esto, se da *click* en *ok* y en el nuevo cuadro que aparecerá, se selecciona la opción "*invert lines*" (en este caso). Ahora, salir del Explorer.



**Figura 3.13 Nombre de la tarea**

4. En el programa, en los recuadros morados conectados al *Write to DAQmx* hay una flecha, se presiona y selecciona la *task* creada. Asegurarse de que en el *Write to DAQmx* esté la opción de "*Digital U8 1 Chan 1 Samp*".

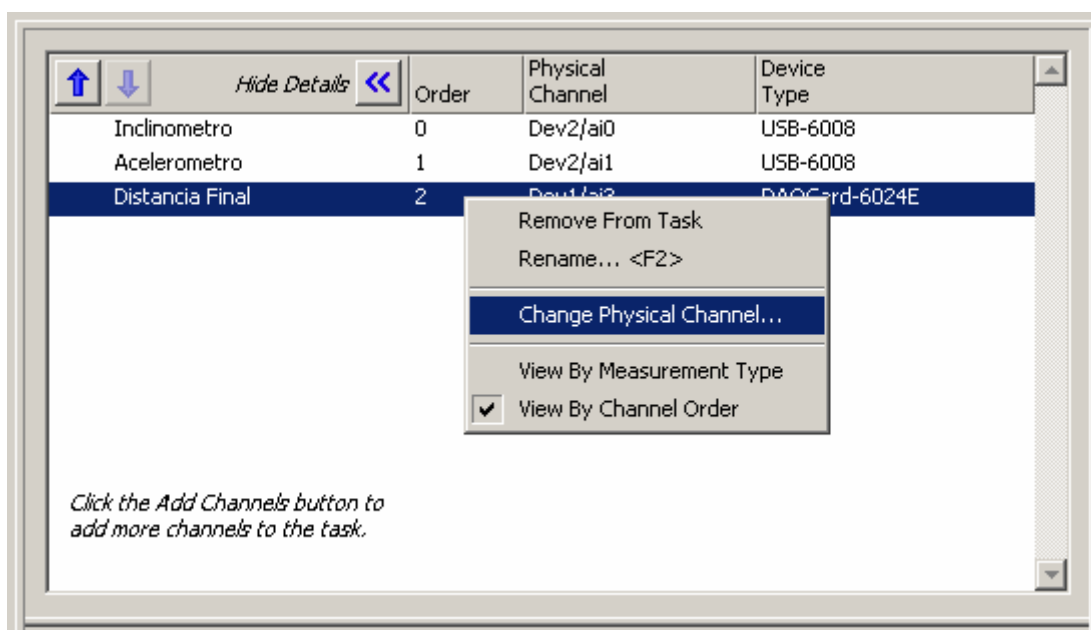


**Figura 3.14 Write to DAQmx**

La ventaja de declarar una *task* es que ya no es necesario poner el *DAQassistant*, ya que se está manejando las salidas del puerto directamente del DAQmx. Con poner

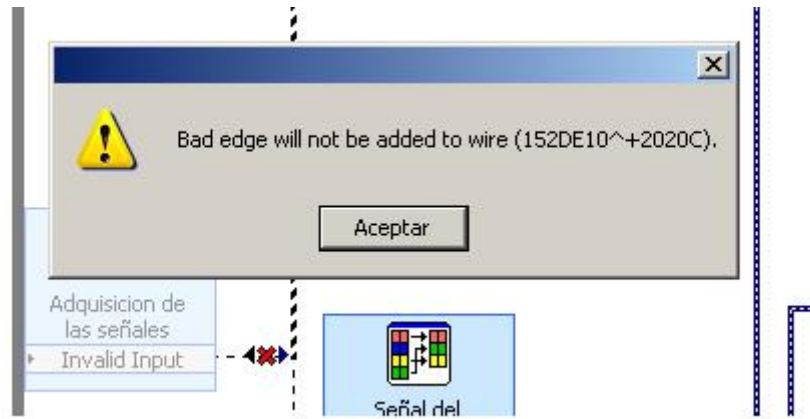
únicamente el bloque de *DAQmx write* incluyendo las terminales de referencia de tarea, el dato a mandar y el *autostart* con una constante de *true* basta para sacar el dato.

Para la parte analógica fue necesario cambiar el canal físico de la adquisición de datos. Esto se hace directamente en el programa ya que este cuenta con bloques tipo *DAC assistant* y que hace la comunicación entre el programa y la tarjeta. Fue necesario cambiar el canal físico de las cuatro señales a considerar: Inclinómetro, Acelerómetro, Distancia Inicial y Distancia Final. (Figura 3.15)



**Figura 3.15 Cambio físico de canales**

Estos canales físicos se cambiaron en las propiedades del *DAQ assistant* en la figura 3.15 se puede observar que se tenía declarada la tarjeta 6024 y que fue necesario remover del canal. Al momento que se hacen los cambios, el bloque anterior se borra y se crea otro distinto pero las conexiones del diagrama a bloques permanecen, es por eso que marcaba errores de tener una mala ruta de conexión (Figura 3.16). Para eliminar este error solo se tuvo que borrar el mal cableado con el comando `ctrl.>>E` que es el asistente de borrador de mal cableado de LabVIEW, y cablear nuevamente los elementos.



**Figura 3.16 Error de mala conexión**

Se migraron los puertos de la DAQ-6008 a los puertos de la DAQ-6024. Los puertos utilizados por la DAQ-6024 se muestran en la tabla 3.1

Señal proveniente de:	Pin en el módulo SCXi-1302	Canal físico de la tarjeta DAQ-6024E
Inclinómetro	3	ai0
Acelerómetro	5	ai1
Distancia inicial	7	ai2
Distancia final	9	ai3
Tierra común	19	aisense

**Tabla 3.1 Conexión física de los elementos [1]**

La migración de estos puertos se realizó desde el programa en las propiedades del *DAQ assistant*. Las figuras 3.17 y 3.18 muestran como el canal físico se cambió para la DAQ-6008 tomando en cuenta la tabla 3.1.

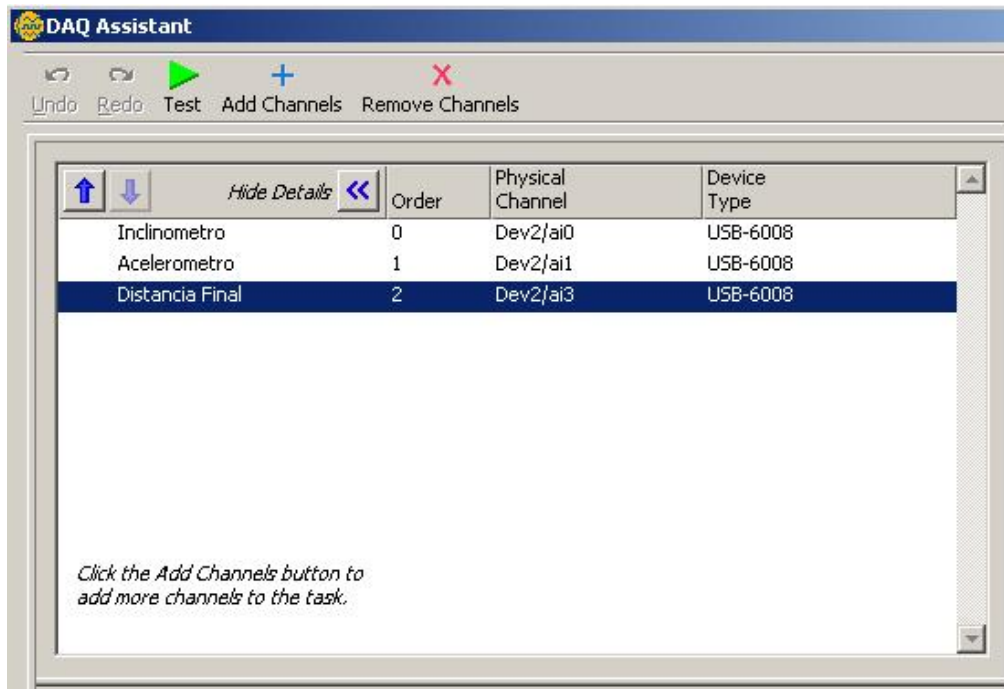


Figura 3.17 Migración de Puertos

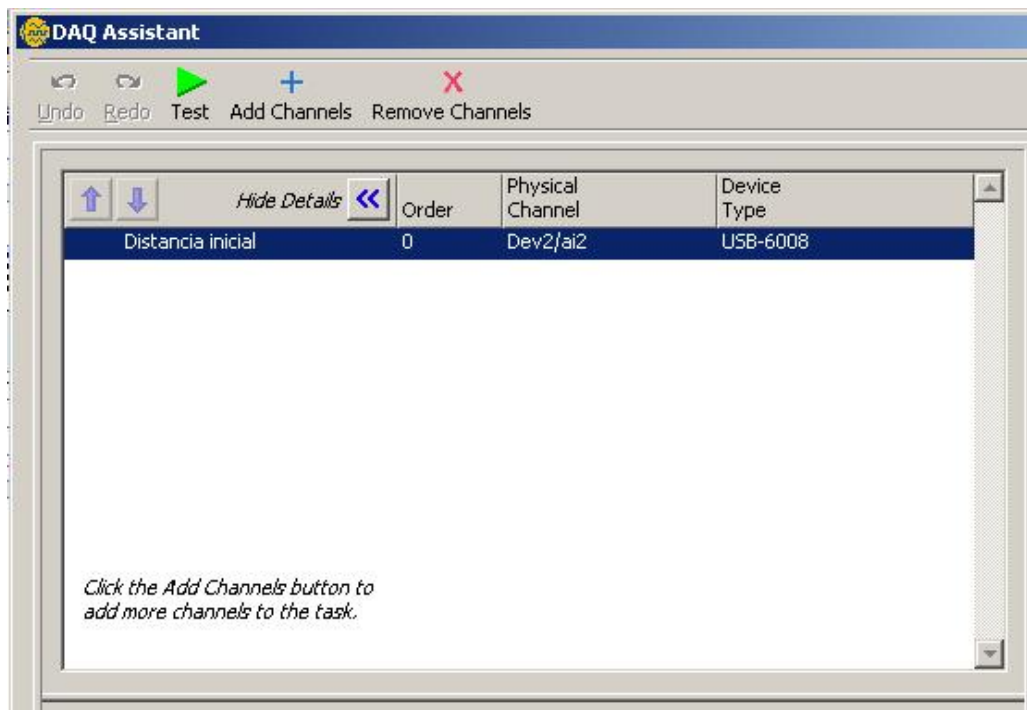


Figura 3.18 Migración de Puertos

Para la parte de la conexión física de los sensores a la DAQ-6008, se eliminaron las conexiones que se tenían que realizar anteriormente hacia la DAQ-6024, ya que se investigó que los módulos de la serie SCXI sólo son conexiones que garantizan la conexión entre dispositivos, y las conexiones hacia la tarjeta DAQ-6008 aceptan cables



directamente la conexión para la tarjeta fue directa hacia los sensores, considerando primero que los rangos de voltaje estuvieran dentro del rango aceptable de voltaje de entrada hacia la tarjeta (Figuras 3.19-22).

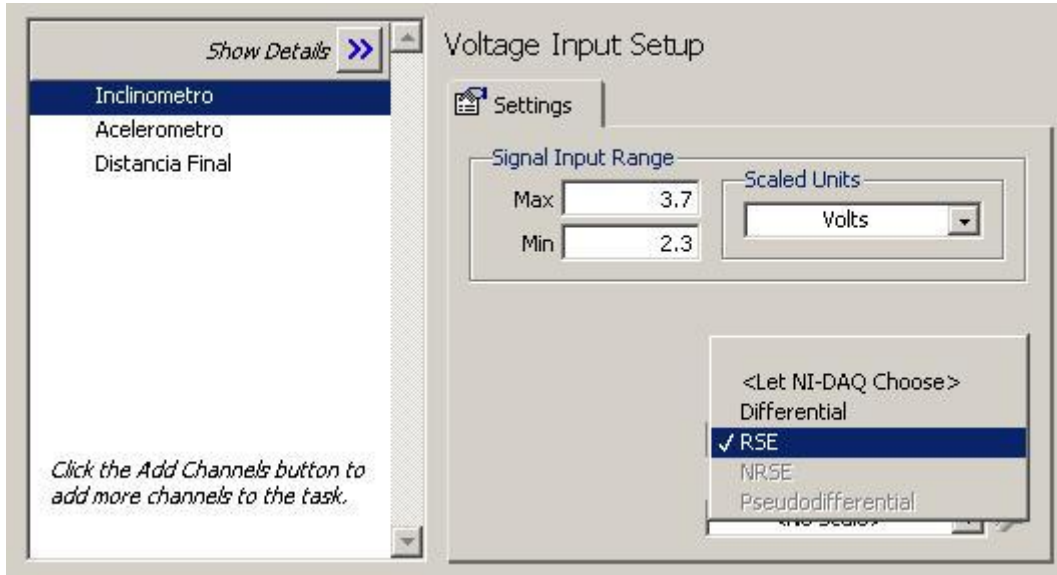


Figura 3.19 Rango de voltajes del Inclínómetro

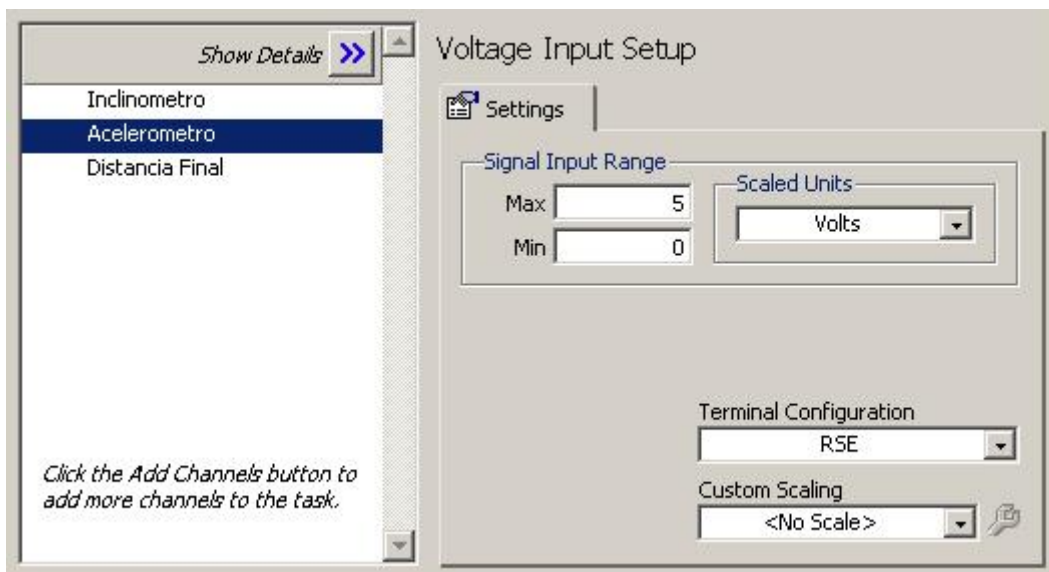
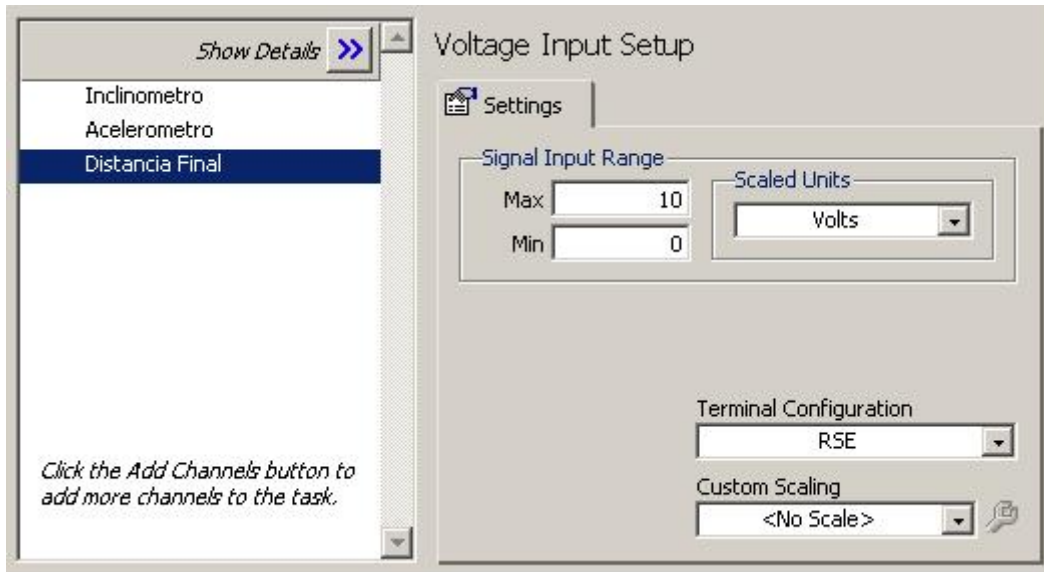
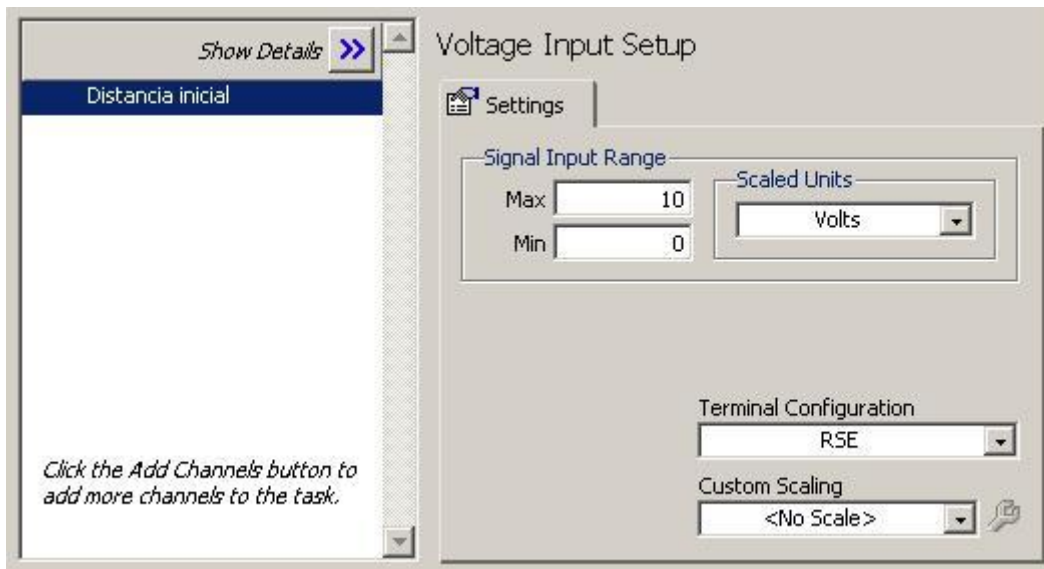


Figura 3.20 Rango de voltajes del Acelerómetro



**Figura 3.21** Rango de voltajes de la Distancia Final



**Figura 3.22** Rango de voltajes de la Distancia Inicial

Conociendo los rangos de los voltajes de entrada, se procedió a verificar que estos fueran los deseados en los canales de entrada de la DAQ-6008. Los canales de entrada hacia la DAQ-6008 se muestran en la tabla 3.2 y los canales analógicos se muestran en la tabla 3.3. Basándose en estas tablas es como se procede a conectar el sistema.

NI USB-6008

GND	1	17	P0.0
AI 0/AI 0+	2	18	P0.1
AI 4/AI 0-	3	19	P0.2
GND	4	20	P0.3
AI 1/AI 1+	5	21	P0.4
AI 5/AI 1-	6	22	P0.5
GND	7	23	P0.6
AI 2/AI 2+	8	24	P0.7
AI 6/AI 2-	9	25	P1.0
GND	10	26	P1.1
AI 3/AI 3+	11	27	P1.2
AI 7/AI 3-	12	28	P1.3
GND	13	29	PFI 0
AO 0	14	30	+2.5 V
AO 1	15	31	+5 V
GND	16	32	GND

Figura 3.23 Pines de la DAQ

SEÑAL DE SALIDA	NÚMERO DE PIN	PUERTO	COLOR
ACTIVAR GIRO DEL MOTOR EN PRO DE MANECILLAS DEL RELOJ	17	P0.0	AZUL
ACTIVAR GIRO DEL MOTOR EN CONTRA DE LAS MANECILLAS DEL RELOJ	18	P1.1	NARANJA
APERTURA Y CIERRE DE RELEVADOR PRIMARIO	19	P2.2	VERDE
GND	32	GND	BLANCO
GND	32	GND	BLANCO

Tabla 3.2 Conexiones Digitales de la DAQ

SEÑAL DE ENTRADA	NÚMERO DE PIN	COLOR	PUERTO
INCLINÓMETRO	2	ROJO	AI0
ACELERÓMETRO	5	AZUL	AI1
DISTANCIA INICIAL	8	NARANJA	AI2
DISTANCIA FINAL	10	NARANJA	GND
TIERRA	1	NEGRO	GND

**Tabla 3.3 Conexiones analógicas de la DAQ-6008**

Respetando la configuración y diseño, se conectan los cables provenientes de los sensores para la adquisición de datos en la bornera análoga de la tarjeta y en las entradas digitales para el control del plano.

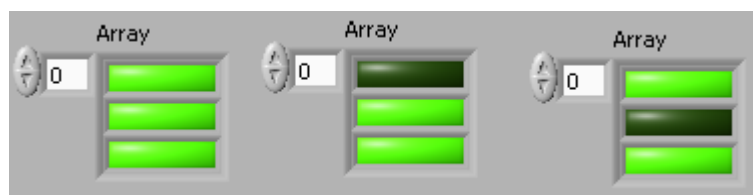
### 3.6 Interfaz con el usuario

La interfaz de control hacia el usuario resultante (Figura 3.26) acopla ambos programas para el fin específico de cada uno. Con esto se logró simplificar la forma de ejecución de la interfaz, ya que es posible el trabajar con ambos programas simultáneamente, es decir se puede mover el plano y al mismo tiempo es posible conocer los datos en tiempo real. El panel frontal hacia el usuario que se muestra en la figura 3.24 contiene los botones necesarios, con sus debidas configuraciones de operación, para subir, bajar, detener el plano y además cuenta con un botón para salir del programa. Un punto importante a señalar es que al inicio y al salir del programa, se cuentan con condiciones iniciales que permiten desconectar a la fuente de alimentación para evitar un accidente, al evitar que el control presente voltaje en sus terminales y no suministre voltaje al motor.



**Figura 3.24 Panel de Control**

Los arreglos que se presentan en la figura 3.25 es la forma de operación de los botones, si se inicia el programa, se presenta el primer caso a la izquierda de la imagen, al presionar subir, se presenta por un segundo el primer caso y luego cambia al caso intermedio, al presionar bajar se presenta nuevamente el caso de la izquierda para luego cambiar al último caso.



**Figura 3.25 Comportamiento bits de control**

El panel frontal de LabVIEW®, es para controlar y que el usuario por medio de este logre conocer y analizar los datos que calcula y recibe por medio de la tarjeta de adquisición de datos.

Se tiene en este panel frontal trabajando al mismo tiempo, las partes tanto analógica como digital de la tarjeta de adquisición de datos, ya que para la parte de la

instrumentación de la primera fase se tienen los sensores, y para la parte digital, los relevadores.

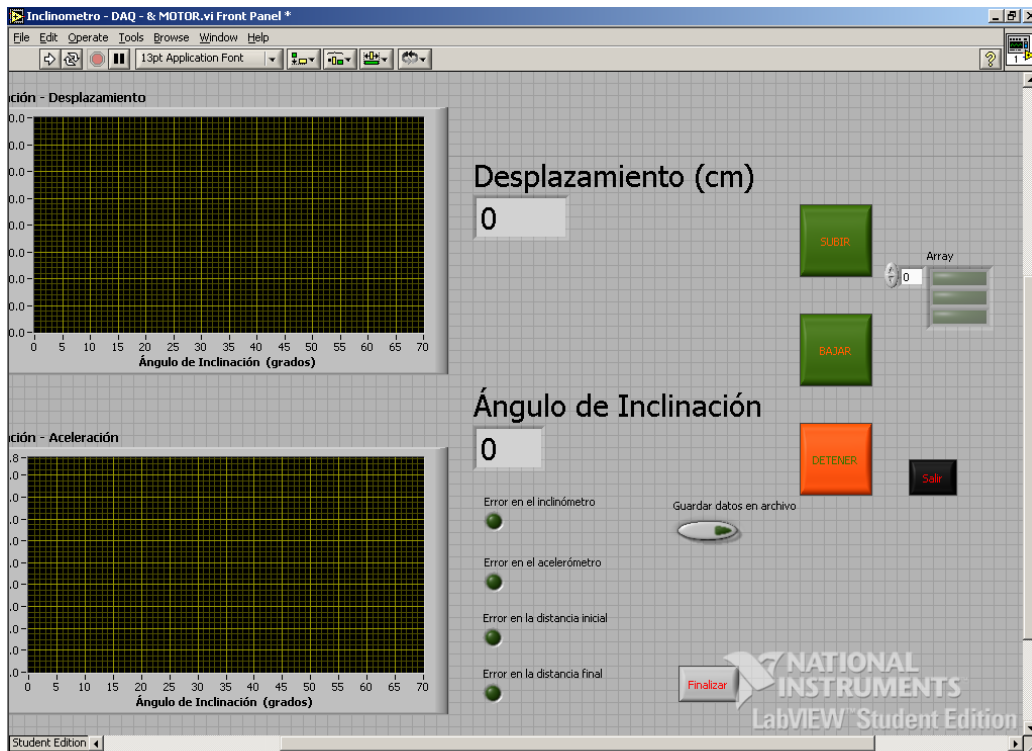


Figura 3.26 Programas en el panel frontal