

**PRÁCTICA EMPRESARIAL
GENERADOR MULTIVOLTAGE**

**EMPRESA MICRONAS GmbH
MÜNCHEN**

GERMAN MENDOZA AVENDAÑO

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2008**

**PRÁCTICA EMPRESARIAL
GENERADOR MULTIVOLTAGE**

**EMPRESA MICRONAS GmbH
MÜNCHEN**

GERMAN MENDOZA AVENDAÑO

**Informe final para optar al título de
Ingeniero Electrónico**

**Director(a)
CLAUDIA LEONOR RUEDA GUZMAN
Ingeniera Electrónica**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2008**

Nota de Aceptación

Jurado

Jurado

Bucaramanga, Enero 15 de 2008

Inicialmente agradezco a Dios por brindarme grandes oportunidades en mi vida personal y profesional.

Dedico este trabajo a mis Padres quienes han sido una fuente constante de apoyo, motivación y quienes me han formado y guiado para seguir un camino de responsabilidad, constancia y preparación para que pueda sortear y enfrentar cualquier tipo de situación que se presente en el futuro.

A mis familiares quienes también me han acompañado en esta experiencia brindándome su apoyo incondicional.

Finalmente a mis amigos, amigas y Profesores con quienes compartí en las aulas de clase y con quienes vivimos grandes experiencias.

German Mendoza Avendaño

AGRADECIMIENTOS

El autor expresa sus agradecimientos a:

Al ingeniero Alex Monclou por el apoyo y por su amistad.

A la Ingeniera Claudia Leonor Rueda Guzmán por su colaboración y por su paciencia.

Al Ingeniero Fabio Guzmán quien ha sido un gran amigo y guía durante el desarrollo de mi carrera profesional.

A mis compañeros de estudio Ximena Cardozo Moreno, Javier Fernando Avendaño Mendoza, Sergio Fabián Gaviria Méndez, Gustavo Andrés Aguillón, Carlos Reyes, Alfredo Sánchez, Erwin Villamizar, Jaime Viña y demás compañeros con quienes compartimos largas jornadas de estudio.

TABLA DE CONTENIDO

	<i>Pág.</i>
RESUMEN.....	14
INTRODUCCIÓN	15
DESCRIPCIÓN DE LA EMPRESA	16
DESCRIPCIÓN DE LA PRÁCTICA	18
OBJETIVOS	19
1. GENERADOR MULTIVOLTAJE.....	20
1.1 ¿QUÉ ES EL GENERADOR MULTIVOLTAJE?	20
2. FUNCIONES DEL PANEL DE CONTROL	21
2.1 FASE DE INICIALIZACIÓN	21
2.2 FASE DE CONFIGURACIÓN DE SEÑAL	22
2.3 FASE DE ADQUISICIÓN DE DATOS.....	28
2.4 FASE DE GENERACIÓN DE SEÑAL	29
2.5 FASE FINAL	29
3. PROGRAMACIÓN	30
3.1 INICIALIZACIÓN	30
3.2 CONFIGURACIÓN DE SEÑAL	33
3.3 FASE DE ADQUISICIÓN DE DATOS.....	38
3.4 FASE DE GENERACIÓN DE SEÑAL	40
3.4.1 GENERACIÓN DE SEÑAL - CONCEPTO BÁSICO UTILIZADO EN EL GENERADOR MULTIVOLTAJE.....	41
3.5 FASE FINAL	48
3.6 CÓMO DETENER EL GENERADOR MULTIVOLTAJE.....	49
4. MODIFICACIONES REALIZADAS	50
4.1 SALVADO Y RECARGA DE LA CONFIGURACION DE SEÑALES	50
4.2 REDUCCIÓN DE PROGRAMACIÓN A SUBVIS.	56
4.2.1 Spike Signal 2_5Subvi	56
4.2.2 Spike All Signals.....	58
4.2.3 Config-Signal5Subvi.....	59
4.3 NUEVOS TIPOS DE PARADAS STOP 1 Y STOP 2	61
4.4 BOTON DE PRESENTACIÓN.....	67
4.5 BOTÓN DE REPORTE RAPIDO	71
4.5.1 Funcionamiento del Programa	76
4.6 GENERACIÓN DE SECUENCIAS DE INICIO (START UP CREATOR).....	77
4.7 CREACION DE RIZOS	89
4.8 GENERADOR DE SECUENCIAS (CREATE SEQUENCE).....	96
4.9 RECARGA DE SEÑALES DESDE ARCHIVO AL TIEMPO	100
5. CORRECCION DE ERRORES.....	101
5.1 CORRECCIONES EN EL START UP CREATOR	101
5.1.1 Cambio de Rango de los controles.....	101

5.1.2 Mejoramiento de la función de Start Delay	102
5.1.3 Nombramiento de los archivos de Excel.....	104
5.1.4 Nombramiento de los archivos en secuencia.	107
5.1.5 Mejora en el botón de copy file.....	109
5.2 CORRECCIÓN DE LA GENERACIÓN DE LA RAMPA	112
5.3 CORRECCIÓN DEL READ FROM FILE	115
5.4 MEJORA EN EL BOTON SAVE SIGNAL.....	119
5.5 SALVADO AUTOMÁTICO EN EL CREATE SEQUENCE	120
5.6 MEJORAS EN EL RIPPLE CREATOR	121
6. ACTUALIZACIÓN LABVIEW 7.1 A LABVIEW 8.2	123
7. PRUEBAS CON CARGA EN EL GENERADOR MULTIVOLTAJE	127
7.1 PRUEBAS CON RESISTENCIAS	127
7.2 PRUEBAS CON BOARD	128
8. HARDWARE.....	135
8.1 AMPLIFICADOR KEPSCO.....	135
8.2 SCB – 68 (Shielded Connector Blocks).....	136
8.3 PCI 6733	139
8.4 PCI 6143	140
CONCLUSIONES	142
BIBLIOGRAFIA	144

LISTA DE FIGURAS

	Pág.
<i>Figura 1. Ejemplo de secuencias de inicio</i>	20
<i>Figura 2. Composición Generador Multivoltaje</i>	21
<i>Figura 3. Panel de control Generador Multivoltaje (Panel Principal)</i>	22
<i>Figura 4. Tab configuración de Señal (Señal 2)</i>	23
<i>Figura 5. Panel de Control para la creación de Rampas</i>	24
<i>Figura 6. Panel de Control de la función Create from Library</i>	25
<i>Figura 7. Panel de control de la función crear arbitrariamente</i>	25
<i>Figura 8. Panel de control de la función Edit Current</i>	26
<i>Figura 9. Funcionamiento botón Clear Channel</i>	26
<i>Figura 10. Procedimiento para la carga de archivo</i>	27
<i>Figura 11. Archivo Cargado</i>	28
<i>Figura 12. Funciones de adquisición de datos</i>	29
<i>Figura 13. Programación fase de inicialización</i>	30
<i>Figura 14. Initialize Array</i>	31
<i>Figura 15. AO config</i>	31
<i>Figura 16. AO Write</i>	32
<i>Figura 17. AO Start</i>	32
<i>Figura 18. DAQmx create channel</i>	32
<i>Figura 19. DAQmx Start task</i>	33
<i>Figura 20. Programación configuración de señal 2</i>	33
<i>Figura 21. Creación de una matriz de ceros de 5000 posiciones</i>	34
<i>Figura 22. Búsqueda de botones dentro de un cluster</i>	34
<i>Figura 23. Search 1D Array</i>	34
<i>Figura 24. Utilización del case structure para la selección de los botones del cluster</i>	35
<i>Figura 25. Selección del botón create from Sketch</i>	36
<i>Figura 26. Edit Current</i>	36
<i>Figura 27. Conversión a un factor de 5000</i>	37
<i>Figura 28. Programación para clarear el canal</i>	37
<i>Figura 29. Programación para el Read from File</i>	38
<i>Figura 30. Read from labview Measurement File</i>	38
<i>Figura 31. Programación fase de adquisición de datos</i>	39
<i>Figura 32. DAQmx Read</i>	39
<i>Figura 33. DAQmx Stop Task</i>	40
<i>Figura 34. DAQmx Clear Task</i>	40
<i>Figura 35. Simple Error Handler</i>	40
<i>Figura 36. Ejemplo de generación de forma de onda, programación básica utilizada en el Generador Multivoltaje</i>	42
<i>Figura 37. Programación cuando no se oprime el botón de empezar generación</i>	43

Figura 38. Delimitación del array de señal a 5000 muestras	43
Figura 39. Spike Signal.....	44
Figura 40. Guardar a un archivo LVM.....	44
Figura 41. Spike all Signals	45
Figura 42. Output Voltage level y re-escritura en el buffer.....	46
Figura 43. Read from file Señal 1	47
Figura 44. Programación fase final	48
Figura 45. AO – 1UP.....	48
Figura 46. AO Clear.....	48
Figura 47. Clareado de las variables booleanas.....	49
Figura 48. Detención del programa.....	49
Figura 49. Botones de recarga.....	50
Figura 50. Guardando varias señales en formato LVM.....	51
Figura 51. Save y Reload no se han presionado	51
Figura 52. Leyendo el archivo con las señales	52
Figura 53. Read From Spreadsheet File.....	53
Figura 54. Write from Spreadsheet File	53
Figura 55. Corrección al dar cancel con el File Dialog.....	53
Figura 56. Corrección al dar cancel al leer el archivo	54
Figura 57. Reloadsubvi.....	54
Figura 58. Controles de Labview para depurar el programa.....	54
Figura 59. Ajuste del SubVI en el programa principal.....	55
Figura 60. Spike signal para las señales 2 a la 5.....	57
Figura 61. Spike signal subVI	57
Figura 62. Spike all signals.....	58
Figura 63. Spike all signals SubVI	59
Figura 64. Modificación para crear el SubVI de configuración de señal	60
Figura 65. Ajuste de Configuración de señal en el programa principal.....	61
Figura 66. Ejemplo de Stop 1.....	62
Figura 67. Ejemplo de Stop 2.....	62
Figura 68. Get date\Time in seconds.....	63
Figura 69. Error 1 en el Stop 2.....	63
Figura 70. Error 2 en el Stop 2.....	64
Figura 71. Error 3 en el Stop 2.....	64
Figura 72. Stop1 y Stop2 sin oprimir.....	65
Figura 73. Programación Stop 2.	66
Figura 74. Programación para el Stop 1	67
Figura 75. Programación para el botón de presentación	67
Figura 76. Ejemplo maquina de estados primera diapositiva.....	68
Figura 77. Ejemplo máquina de estados última diapositiva.	69
Figura 78. Ejemplo diapositivas realizadas.....	69
Figura 79. Propiedades de un VI	70
Figura 80. Cambio de parámetros a un VI.....	70

<i>Figura 81. Ventana de configuración de apariencia de un VI</i>	<i>70</i>
<i>Figura 82. Funciones del Reporte rápido.</i>	<i>71</i>
<i>Figura 83. Configuración en el Programa Principal.....</i>	<i>72</i>
<i>Figura 84. Programación del SubVI de Reporte Rápido</i>	<i>73</i>
<i>Figura 85. New Report.....</i>	<i>74</i>
<i>Figura 86. Set Report Font</i>	<i>74</i>
<i>Figura 87. Set Report Orientation</i>	<i>74</i>
<i>Figura 88. Append Report Text.....</i>	<i>74</i>
<i>Figura 89. New Report Line</i>	<i>75</i>
<i>Figura 90. Append Control Image to Report</i>	<i>75</i>
<i>Figura 91. Image in Header or Footer.....</i>	<i>75</i>
<i>Figura 92. Save Report to File</i>	<i>75</i>
<i>Figura 93. Token Converter.....</i>	<i>76</i>
<i>Figura 94. Prompt User for Input.....</i>	<i>76</i>
<i>Figura 95. Introducción del nombre de usuario para el reporte.</i>	<i>76</i>
<i>Figura 96. Ejemplo de un reporte.....</i>	<i>77</i>
<i>Figura 97. Start up Creator en el programa principal</i>	<i>77</i>
<i>Figura 98. Start up Creator Panel Frontal.</i>	<i>78</i>
<i>Figura 99. Función Copy File.....</i>	<i>80</i>
<i>Figura 100. Lectura de archivo por primera vez.....</i>	<i>81</i>
<i>Figura 101. Lectura de archivo después del Refresh.....</i>	<i>81</i>
<i>Figura 102. Creación de rampa</i>	<i>83</i>
<i>Figura 103. Delay, inserción de rampa y offset.....</i>	<i>83</i>
<i>Figura 104. Array de posiciones.</i>	<i>84</i>
<i>Figura 105. Propiedades de cursor Waveform Graph.....</i>	<i>85</i>
<i>Figura 106. Lectura de posición de dos cursores.</i>	<i>85</i>
<i>Figura 107. Programación para guardar 5 archivos.....</i>	<i>86</i>
<i>Figura 108. Programación para salvar un archivo y refrescar</i>	<i>87</i>
<i>Figura 109. Programación para importar a Excel.....</i>	<i>88</i>
<i>Figura 110. Lectura desde Excel</i>	<i>89</i>
<i>Figura 111. Subvi de Creación de Rizos.....</i>	<i>89</i>
<i>Figura 112. Creación de rizos panel frontal</i>	<i>89</i>
<i>Figura 113. Programación completa Ripple Creator.....</i>	<i>91</i>
<i>Figura 114. Funciones de Clear, tamaño de array.....</i>	<i>92</i>
<i>Figura 115. Inserción del rizado.....</i>	<i>93</i>
<i>Figura 116. Programación de las propiedades del waveform graph</i>	<i>94</i>
<i>Figura 117. Generator SubVI.....</i>	<i>94</i>
<i>Figura 118. Panel Frontal Generator</i>	<i>95</i>
<i>Figura 119. Programación interna del Generator SubVI</i>	<i>95</i>
<i>Figura 120. Panel de control Generador de secuencias</i>	<i>96</i>
<i>Figura 121. Programación del Generador de secuencias en el Generador Multivoltaje.....</i>	<i>97</i>
<i>Figura 122. Programación interna generador de secuencias</i>	<i>98</i>

<i>Figura 123. Programación generador de secuencia un solo canal</i>	99
<i>Figura 124. Recarga de señales desde archivo</i>	100
<i>Figura 125. Función Select de labview</i>	101
<i>Figura 126. Controles de Offset, Start delay</i>	102
<i>Figura 127. SubVI para desplazar un array en dos direcciones</i>	102
<i>Figura 128. Desplazamiento a la derecha</i>	102
<i>Figura 129. Desplazamiento a la izquierda</i>	103
<i>Figura 130. SubVI Change extention</i>	104
<i>Figura 131. Programación interna SubVI Change Extention</i>	104
<i>Figura 132. SubVI String to array</i>	106
<i>Figura 133. Programación interna String to array SubVI</i>	106
<i>Figura 134. Corrección extensión en el nombre de un archivo</i>	107
<i>Figura 135. Corrección nombre de archivos guardados en secuencia</i>	108
<i>Figura 136. Adición de unidades al nombre de archivo</i>	109
<i>Figura 137. Corrección programación botón copy file</i>	111
<i>Figura 138. Función Copy</i>	111
<i>Figura 139. Corrección al dar cancel en Labview 8.2</i>	112
<i>Figura 140. Error 1 generación de rampa.</i>	112
<i>Figura 141. Comprobación tamaño de matriz.</i>	113
<i>Figura 142. Modificación de Spike Signal con variables globales</i>	113
<i>Figura 143. Configuración de señales con variables globales.</i>	114
<i>Figura 144. Error 2 de generación de rampa</i>	114
<i>Figura 145. Programación interna Read Measurement file</i>	116
<i>Figura 146. Modificación para leer archivos CVS.</i>	117
<i>Figura 147. SubVI Read File11</i>	118
<i>Figura 148. Programación interna SubVI Read file11</i>	118
<i>Figura 149. Nueva lectura de archivos LVM.</i>	119
<i>Figura 150. Corrección para el botón save file</i>	120
<i>Figura 151. Identificación ultimas 5000 muestras de un archivo</i>	121
<i>Figura 152. Ejemplo de rizados</i>	121
<i>Figura 153. Programación mejorada de rizado</i>	122
<i>Figura 154. Panel de control SubVI creación de rizos</i>	122
<i>Figura 155. Modificación programación para la rampa en Ripple Creator</i>	123
<i>Figura 156. File Dialog en Labview 7.1</i>	125
<i>Figura 157. File dialog en Labview 8.2</i>	126
<i>Figura 158. Board VCTPremiumD</i>	128
<i>Figura 159. Resultados prueba 1 board</i>	129
<i>Figura 160. Resultados Prueba 2 Board</i>	130
<i>Figura 161. Modificación 1 en el board de prueba</i>	130
<i>Figura 162. Modificación 2 en el board de pruebas.</i>	131
<i>Figura 163. Generador Multivoltaje</i>	131
<i>Figura 164. Ventana terminal de recepción de pulso</i>	132
<i>Figura 165. Prueba 3 en el Board</i>	132

<i>Figura 166. Prueba 3 en el board escala de tiempo ampliada</i>	133
<i>Figura 167. Prueba 4 en el board</i>	133
<i>Figura 168. Prueba 5 en el Board.</i>	134
<i>Figura 169. Amplificador KEPCOFigura 8.0.</i>	135
<i>Figura 170. Conector trasero PC 12</i>	136
<i>Figura 171. Board Interno SCB-68.</i>	137
<i>Figura 172. Diagrama de pines SCB – 68</i>	138
<i>Figura 173. Diagrama de pines SCB – 68 configurado para recibir datos</i>	139
<i>Figura 174. Tarjeta Generadora.</i>	140

LISTA DE TABLAS

	<i>Pág.</i>
Tabla 1. Orden de los elementos agregados a un cluster.....	35
Tabla 2. Requerimientos del sistema para instalar Labview 8.2 en Windows.....	124
Tabla 3. Puntos de prueba Board VCTPremiumD	127
Tabla 4. Conexiones SCB – 68 con la tarjeta PCI 6733	138

LISTA DE ANEXOS

	<i>Pág.</i>
ANEXO 1: DIAGRAMA DE FLUJO GENERAL DEL GENERADOR MULTIVOLTAJE	146
ANEXO 2: CONTINUACION DIAGRAMA DE FLUJO GENERAL DEL GENERADOR MULTIVOLTAJE (M).....	147
ANEXO 3: CONFIGURACIÓN DE SEÑALES (1)	148
ANEXO 4: CREATE FROM SKETCH AND EDIT CURRENT (1.1 Y 1.2)	149
ANEXO 5: RAMP UP (1.3).....	150
ANEXO 6: READ (FROM FILE) (1.4).....	151
ANEXO 7: CREATE RIPPLE (1.5).....	152
ANEXO 8: CREATE FROM LIBRARY (1.6).....	153
ANEXO 9: QUICK REPORT (2).....	154
ANEXO 10: BOTON DE INTRODUCCION RAPIDA (3)	155
ANEXO 11: RELOAD SUBVI (4).....	156
ANEXO 12: START UP CREATOR (5).....	157
ANEXO 13: CREATE SEQUENCE SUBVI (6).....	158
ANEXO 14: CONTINUACION CREATE SEQUENCE SUBVI.....	159
ANEXO 15: STOP 1 Y STOP 2 (7)	160
ANEXO 16: FACTOR DE 5000.....	161
ANEXO 17: READ LVM FILE 11 SUBVI.....	162

RESUMEN

La práctica empresarial fue realizada en Alemania en la ciudad de Munich en la empresa Micronas en el departamento de evaluación de chips, donde se encuentra un equipo en construcción conocido como el Generador Multivoltaje.

El equipo esta constituido en hardware por amplificadores marca Kepco, bloques conectores de 68 pines, tarjeta de adquisición y generación. En cuanto a software se encuentra programado bajo el software de National Instruments Labview. La programación inicial de sus funciones fue realizada con anterioridad por un practicante proveniente de la india. El generador multivoltage posee cinco canales de generación de señal capaces de generar 5000 muestras por segundo a través de su tarjeta PCI generadora. De la misma manera el generador esta en capacidad de adquirir una señal.

El principal interés durante el desarrollo de esta practica fue la implementación de nuevas características, es decir, mas funciones que permitan al usuario mayor capacidad de ejecución de tareas con el generador, como por ejemplo la posibilidad de modificar un archivo que contiene los datos de una señal, detener la señal de dos maneras posibles, crear un reporte, generar una secuencia completa de una señal cargando dos archivos diferentes, cargar toda una configuración de forma mas rápida, generar rizados de manera sencilla, entre otras características. También por supuesto se corrigieron errores en otros procesos como el de salvado y recarga de las señales.

Todo esto con el propósito de utilizar el generador para aplicar las señales que desde allí se generen a los circuitos en prueba y observar las respuestas que los circuitos, boards o circuitos integrados presentan ante diferentes tipos de señales.

INTRODUCCIÓN

Durante toda la historia de la electrónica han existido una gran variedad de adelantos científicos que han permitido que grandes circuitos se hayan podido reducir en tamaño, tanto que se pueden agrupar en una pastilla de silicio mejorando no solo parámetros como rendimiento, confiabilidad y manejo sino parámetros como estética y ahorro de espacio. De la mano con las nuevas técnicas de reducción también se han presentado retos de diseño y de ingeniería que han hecho que el hombre desarrolle mejoras en la tecnología para corregir los errores presentados permitiendo así también el desarrollo continuo de nuevos equipos de medición y prueba.

El generador multivoltage es un equipo en desarrollo cuyo principal objetivo es generar una serie de señales para ser aplicadas a los circuitos integrados y así poder observar y analizar los efectos ocasionados al polarizar un circuito integrado en diferentes situaciones, debido a que los eventos de energización de un dispositivo generalmente siempre son diferentes. Al generar o manipular los diferentes eventos de polarización o de energización de un chip o de un board se puede identificar parámetros críticos para el elemento en prueba o en fabricación, sin embargo, para ello se requiere de un sistema de fácil manejo.

Dentro del contenido de este informe el lector puede identificar y reconocer ciertas características como por ejemplo programación detallada de cada una de las fases, el diseño y construcción del proyecto Generador Multivoltage así como también aspectos básicos de funcionamiento que permiten al usuario interactuar con el programa, el cual ha sido implementado bajo el software de programación grafica Labview.

Durante el desarrollo y creación del Generador siempre se mantuvo la idea básica de realizar un sistema de fácil comprensión, apto para el manejo de cualquier usuario, tratando así, crear un dispositivo de arquitectura abierta. Esto le permite al sistema obtener características como configurar o acoplar el dispositivo a nuevos elementos e incluso añadir mas programación para ampliar mas las funciones para lo que fue creado.

El proyecto fue construido y pertenece a la empresa Micronas GmbH, empresa que se encarga de desarrollar, fabricar y vender componentes semiconductores, además de realizar aplicaciones para los campos de la electrónica de consumo, multimedia y la electrónica automotriz.

DESCRIPCIÓN DE LA EMPRESA

Micronas es una empresa que se encarga de desarrollar, fabricar y vender componentes semiconductores (microchips) en todas partes del mundo. Además de sensores para aplicaciones en campos de la electrónica de consumo, multimedia y electrónica automotriz.

Las principales áreas donde Micronas se enfoca son en los consumidores, industria automotriz y multimedia. Micronas posee Alrededor de 2200 empleados donde aproximadamente 165 se encuentran en Munich.

Productos

- Circuitos Integrados de Audio, Video y datos para aplicaciones en TV, VCR, productos multimedia.
- Procesadores de Sonido.
- Procesadores de Video.
- Controladores de video texto y servicio de datos.
- Procesadores de compresión de datos (Ejemplo para MP3)
- Controladores para automotores.
- Sensores de salón.

Micronas es líder del mercado en soluciones globales innovadoras del sistema de TV. Lidera con maestría en los nuevos mercados que emergen con la numeración del contenido de audio y video. Micronas sirve a todas las marcas de fábrica importantes del mundo, muchas de ellas en sociedades continuas que buscan éxito común.

Mientras que establecen la compañía que sostiene en Zurich (Suiza), las jefaturas operacionales se basan en Freiburg (Alemania). Actualmente, el grupo de Micronas emplea cerca de 2200 personas.

En el 2006 Micronas generó CHF 813/USD 648/EUR 515 millones en ventas. En el campo de la electrónica de consumo, el foco principal está en las soluciones de audio y video de próxima generación y sistemas virtuales para el uso en televisiones digitales.

Micronas tiene más de 20 años de experiencia en el proceso digital de las señales de televisión, que ha canalizado en sus productos actuales, permitiendo alcanzar el nivel del alto rendimiento ahora logrado.

Los productos de la nueva generación incluyen las soluciones para la recepción digital de la TV. También como productos de exhibición para apoyar los sistemas de la televisión de alta definición (HDTV) basados en el LCD, el plasma y la tecnología del DLP.

En 2005, Micronas adquirió WISchip, hoy Micronas USA. En Santa Clara, la compañía desarrolla de manera estable y ascendente productos que permiten a los consumidores archivar información en el mundo de negocios de hoy mientras se prepara para el cambio del mañana a IPTV, Set-Top Boxes, Networked Digital TV e IP Surveillance.

La división electrónica automotora de Micronas se está centrando en los sistemas altamente integrados para los usos interiores del coche y en los sistemas de sensores para una gran gama de usos. Micronas posee más de 30 años de experiencia automotora, desde el primer circuito electrónico empleado en el interior del coche - el IC del reloj.

DESCRIPCIÓN DE LA PRÁCTICA

La práctica empresarial fue realizada en el departamento de investigación de Micronas, más específicamente en el área de evaluación de chips donde se ejecutan diferentes tipos de proyectos los cuales en su mayoría son para el desarrollo y mejoramiento de los procesadores de imagen para televisores de plasma y LCD.

Dentro de este departamento hay una serie de profesionales bien capacitados en los manejos de software y equipos que se utilizan en los procesos de evaluación y mejoramiento de los circuitos integrados.

Entre los proyectos allí desarrollados se encuentra el generador Multivoltaje cuya finalidad, es decir, el propósito para el cual fue construido es aplicar diferentes señales a los circuitos integrados con el fin de analizar y estudiar las respuestas que estos presentan a diferentes tipos de polarizaciones.

El generador multivoltaje esta creado bajo software utilizando Labview 7.1, donde la primera fase de programación fue realizada por un practicante anterior proveniente de la India.

OBJETIVOS

OBJETIVO GENERAL

Comprender, analizar el funcionamiento del generador multivoltaje, para así permitir realizar cambios en su programación y hacer de este un dispositivo más efectivo y funcional.

OBJETIVOS ESPECIFICOS

- Identificar las partes del generador multivoltaje y a su vez comprender su funcionamiento.
- Adquirir conocimientos y destrezas en el manejo del software de programación Labview.
- Analizar y comprender las partes básicas y generales de la programación del generador multivoltaje.
- Desarrollar mejoras al generador multivoltaje con el fin de hacer de este un sistema más óptimo.
- Identificar los posibles problemas del generador multivoltaje en los experimentos que se efectúen al momento de aplicar señales a los circuitos integrados.
- Actualizar el programa de labview 7.1 a labview 8.2 con el propósito de obtener más posibilidades de mejoras en el futuro.

1. GENERADOR MULTIVOLTAJE

1.1 ¿QUÉ ES EL GENERADOR MULTIVOLTAJE?

El generador multivoltaje es un dispositivo que tiene como finalidad generar y adquirir señales. Las señales que se generen con este dispositivo son aplicadas a los bancos de pruebas de los circuitos integrados para analizar el comportamiento que estos presenten ante estas señales.

Cuando se energiza un circuito integrado se pueden presentar diferentes situaciones, como por ejemplo que la señal llegue atrasada, que presente rizos, picos, etc.

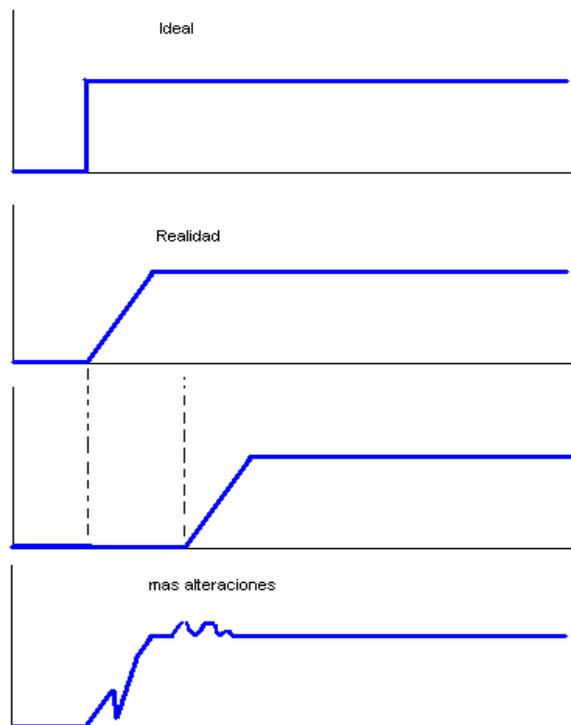


Figura 1. Ejemplo de secuencias de inicio

El generador multivoltaje esta compuesto en Hardware por un computador, cuatro amplificadores Bipolares marca KEPCO, dos SCB-68, dos Tarjetas PCI, PCI 6733 Tarjeta Generadora y PCI 6143 Tarjeta de adquisición de datos.

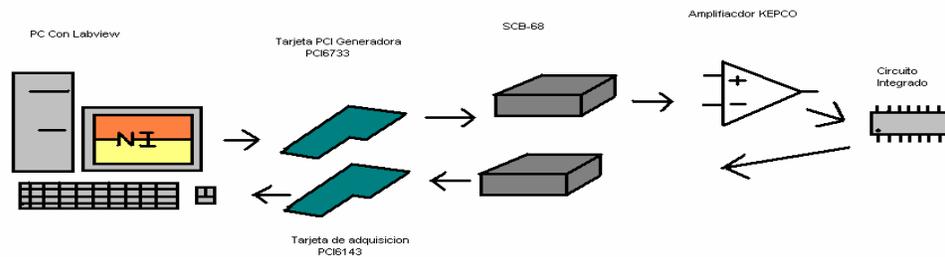


Figura 2. Composición Generador Multivoltaje

El generador Multivoltaje es capaz de generar 5000 muestras por segundo para cada una de las señales en las cuales está configurado y generar una señal sobre la otra hasta que las muestras sean reemplazadas por datos nuevos.

También es capaz de controlar los tiempos de rampa, pendiente, generar picos, generar señales en formato LVM, CVS y salvar en un archivo las señales.

En cuanto a software, el generador Multivoltaje está programado bajo Labview 7.1 o lenguaje de programación G cuya principal característica es un lenguaje gráfico y sencillo de manipular.

La programación está creada bajo cinco fases:

- 1- Inicialización
- 2- Configuración de Señal
- 3- Adquisición de datos
- 4- Generación de señal
- 5- Fase final

2. FUNCIONES DEL PANEL DE CONTROL

Como se mencionó anteriormente la programación del Generador Multivoltaje está realizada basada en cinco fases. Para entender de manera más clara la programación, a continuación se explicarán cada una de las funciones de cada fase y posteriormente se explicará la programación del diagrama de bloques de Labview.

2.1 FASE DE INICIALIZACIÓN

En esta fase se da lugar a la inicialización del sistema clareando todas sus variables y llevando los controles a utilizar a su estado inicial, el cual es de cero excepto para el botón de alimentación.

En esta fase también se deja la PCI 6733 (Tarjeta generadora) inicializada con un buffer de 5000 muestras y lista para la generación de las señales.

También se deja preparada la inicialización para la adquisición de datos. En la figura 3 se puede observar el panel de control cuando se inicia el sistema por primera vez.

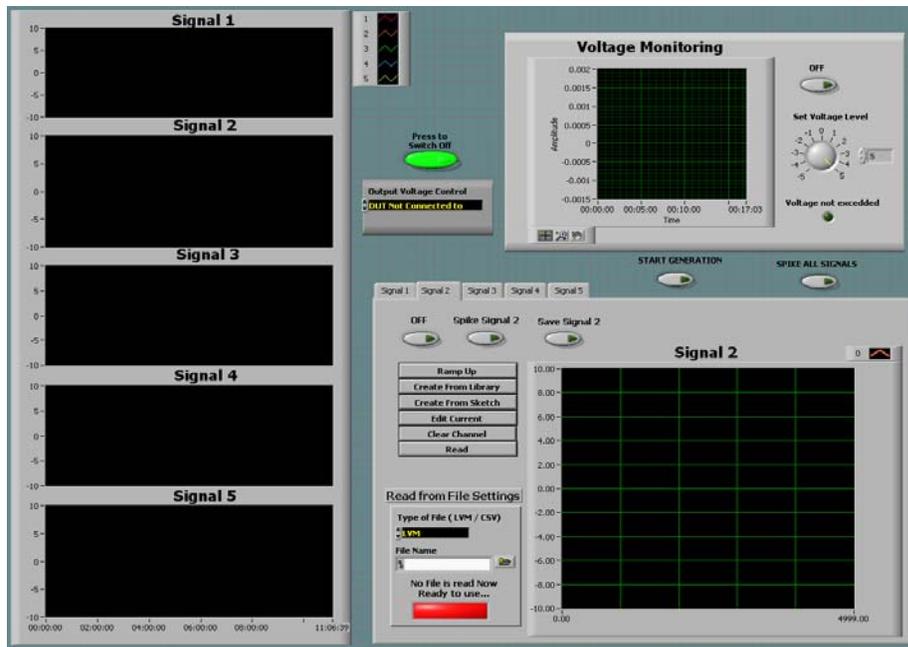


Figura 3. Panel de control Generador Multivoltaje (Panel Principal)

2.2 FASE DE CONFIGURACIÓN DE SEÑAL

En esta fase las señales son configuradas de manera individual. Aquí se puede observar en el panel de control que para la configuración de cada señal se hace uso de un Tab control, dentro del cual se encuentran los controles de encendido y apagado de la señal, los controles de creación de señal desde librería, crear arbitrariamente, editar la señal que se muestra, clearar canal, leer desde archivo, salvar señal y generar picos.

Así como también se puede visualizar un grafico de manera independiente de la señal que se desea generar.

Es importante recalcar que el usuario no puede activar ninguna de las funciones anteriormente nombradas sin antes haber oprimido el botón de encendido de la señal correspondiente que desea trabajar.

Funciones como crear arbitrariamente, crear desde librería, editar señal, corresponden a subrutinas o SubVIs que poseen su propio panel de control junto con sus propias funciones, de igual manera fáciles de utilizar y entender para el usuario.

La siguiente figura corresponde al Panel de control de labview donde se realiza la configuración de cada señal:

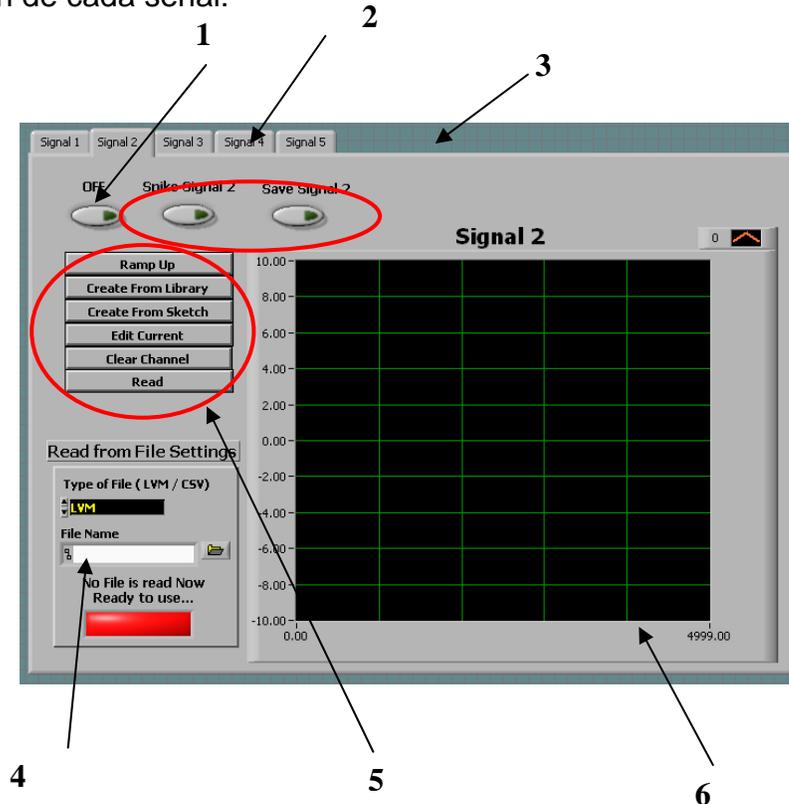


Figura 4. Tab configuración de Señal (Señal 2)

Controles:

- 1- ON/OFF Button, este botón se utiliza para encender o seleccionar la señal que se desea utilizar. Este botón es monitoreado constantemente para conocer su estado.
- 2- SPIKE SIGNAL Y SAVE SIGNAL. Estos controles por programación no están incluidos en la fase de configuración. Corresponden a la fase de generación, pero se hacen necesarios tener estos controles de manera independiente para cada señal.
- 3- Tab Control. Esta útil herramienta de Labview permite realizar de manera independiente cinco veces la configuración para cada señal sin abarcar

tanto espacio. Como se puede observar en la figura 2.1 tiene forma de capetas y se puede seleccionar la que se necesite.

- 4- Read from file Settings. En esta parte, se carga el archivo a leer. El archivo puede tener dos tipos de extensiones LVM o CVS. Se puede seleccionar la extensión que se desee, luego se coloca la dirección donde fue guardado el archivo y después presionar la opción de read. El archivo contiene una tabla de valores para dibujar una señal.
- 5- Aquí se encuentran los controles con los cuales se puede configurar la señal. Estos controles están reunidos todos en un cluster como más adelante se explicara en la programación. La función de cada uno de estos es la siguiente:

5.1 Ramp up. Al oprimir este botón se abre un nuevo subvi que da las opciones para generar una rampa. Como se puede observar en la figura 5 se tienen controles para especificar el tiempo de inicio de la rampa, la amplitud y la duración de esta, así como también se puede agregar picos de voltaje.

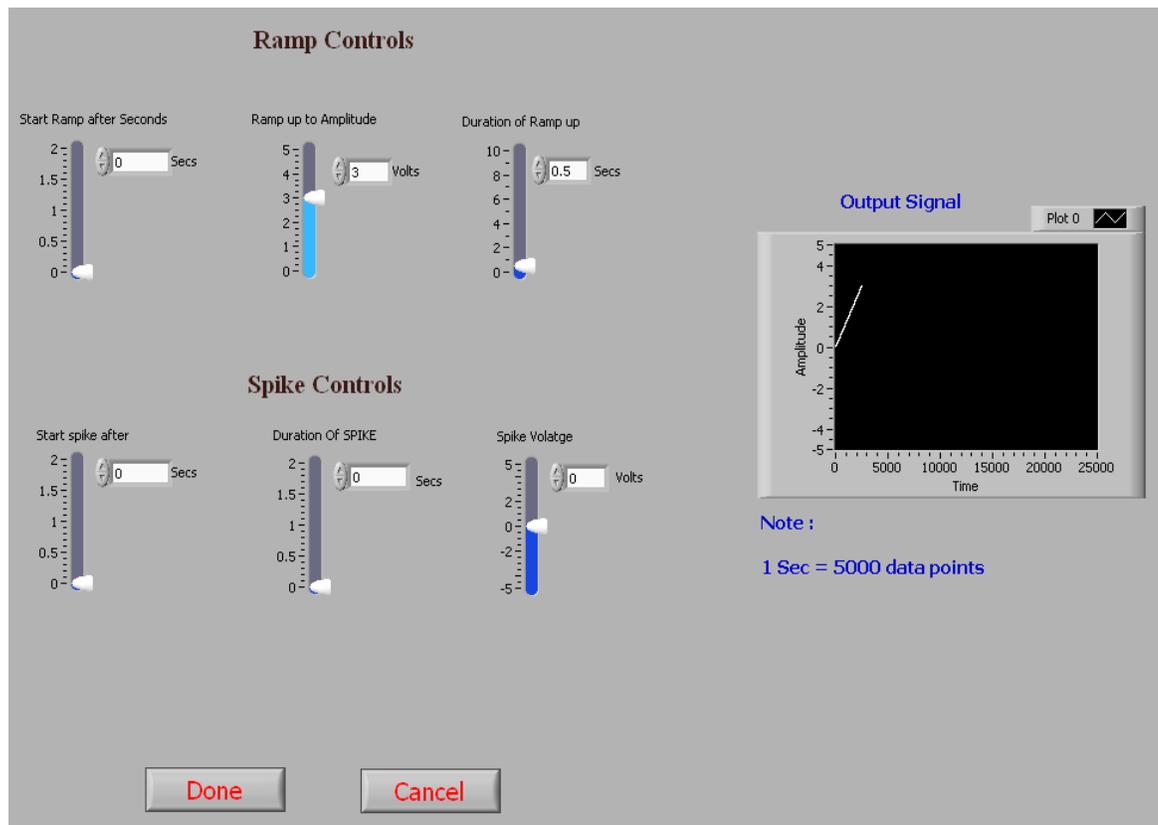


Figura 5. Panel de Control para la creación de Rampas

5.2 Create from library. Aquí se crean las formas de onda típicas como seno, triangular, cuadrada y diente de sierra. También se puede modificar algunos parámetros como la amplitud, ciclos por segundo, offset, etc.

En la siguiente figura se observan los controles que se manejan cuando se selecciona esta opción:

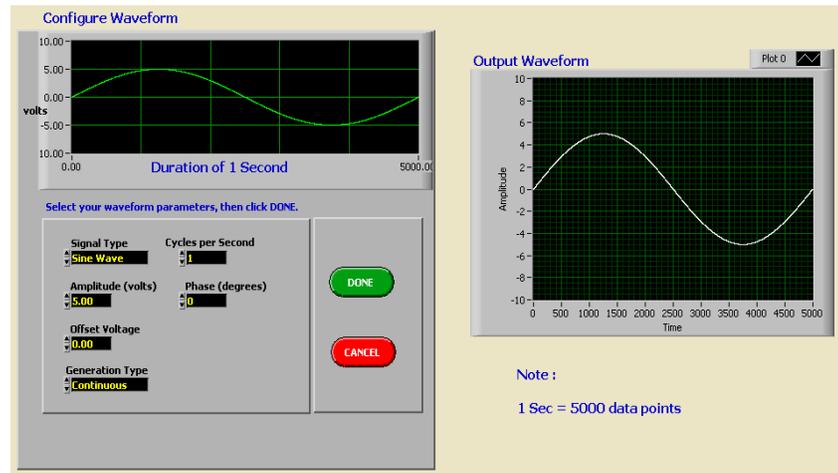


Figura 6. Panel de Control de la función Create from Library

5.3 Create from Sketch. Al oprimir este botón se llama nuevamente a otro subvi que permite al usuario crear una forma de onda de manera arbitraria dibujando esta con el cursor del Mouse como se puede ver en la siguiente figura:

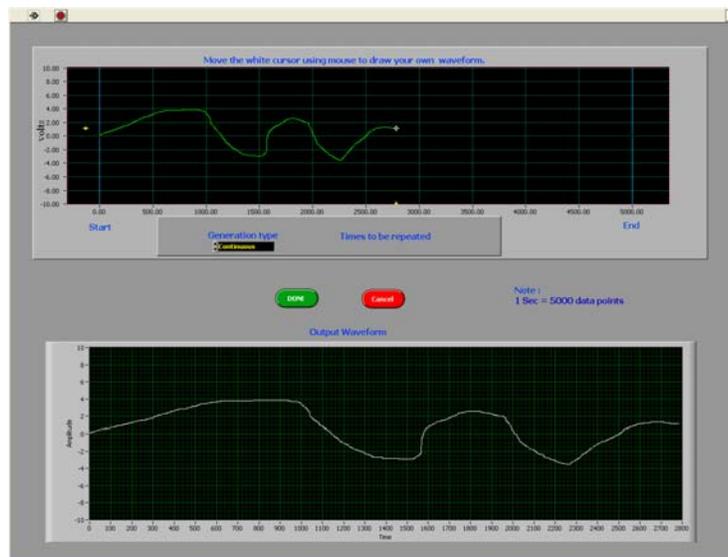


Figura 7. Panel de control de la función crear arbitrariamente

La figura 7 muestra que mientras se grafica la señal en el Waveform chart superior, en el inferior se puede observar la onda resultante simultáneamente.

5.4 Edit Current. Con esta opción se puede modificar una señal. Por ejemplo, el gráfico de abajo muestra una señal seno a la que se le puede agregar algunas irregularidades solo con mover el cursor del Mouse.



Figura 8. Panel de control de la función Edit Current

Esto es posible solo si se ha creado la señal previamente, utilizando los controles antes mencionados, es decir, utilizando la opción de create from Sketch, Read from file, Create from library o Ramp up.

5.5 Clear channel. Aquí la señal que se pensaba generar puede ser borrada. Es muy útil cuando por equivocación el usuario crea una señal en el canal que no desea.

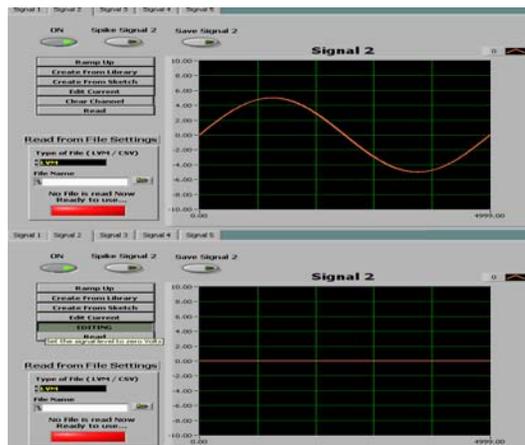
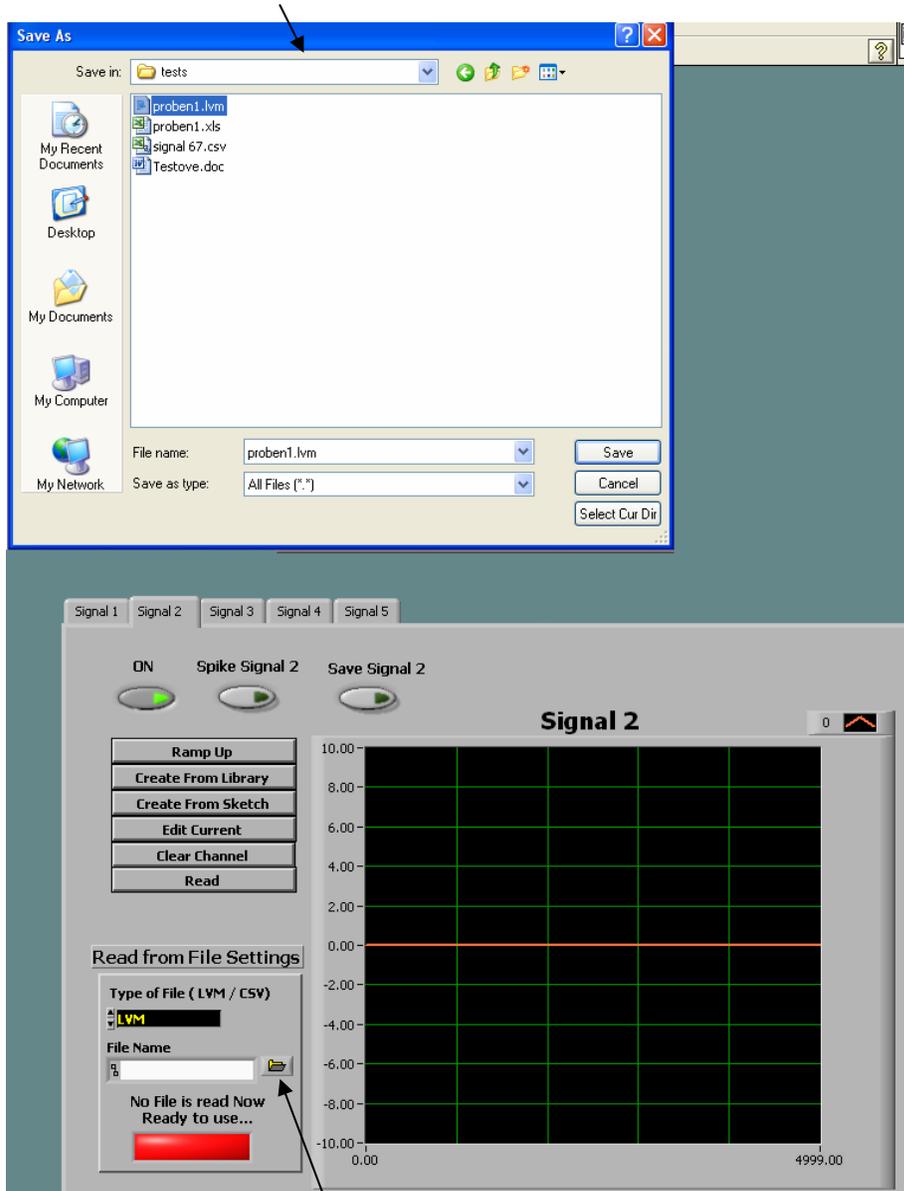


Figura 9. Funcionamiento botón Clear Channel

5.6 Read. Carga el archivo que se desea leer ya sea en formato LVM o CVS (valores separados por comas). Pero para ejecutarla primero se debe cargar la dirección del archivo como se muestra en la siguiente figura:

Seleccionar el archivo y dar click en save o hacer doble click sobre este.

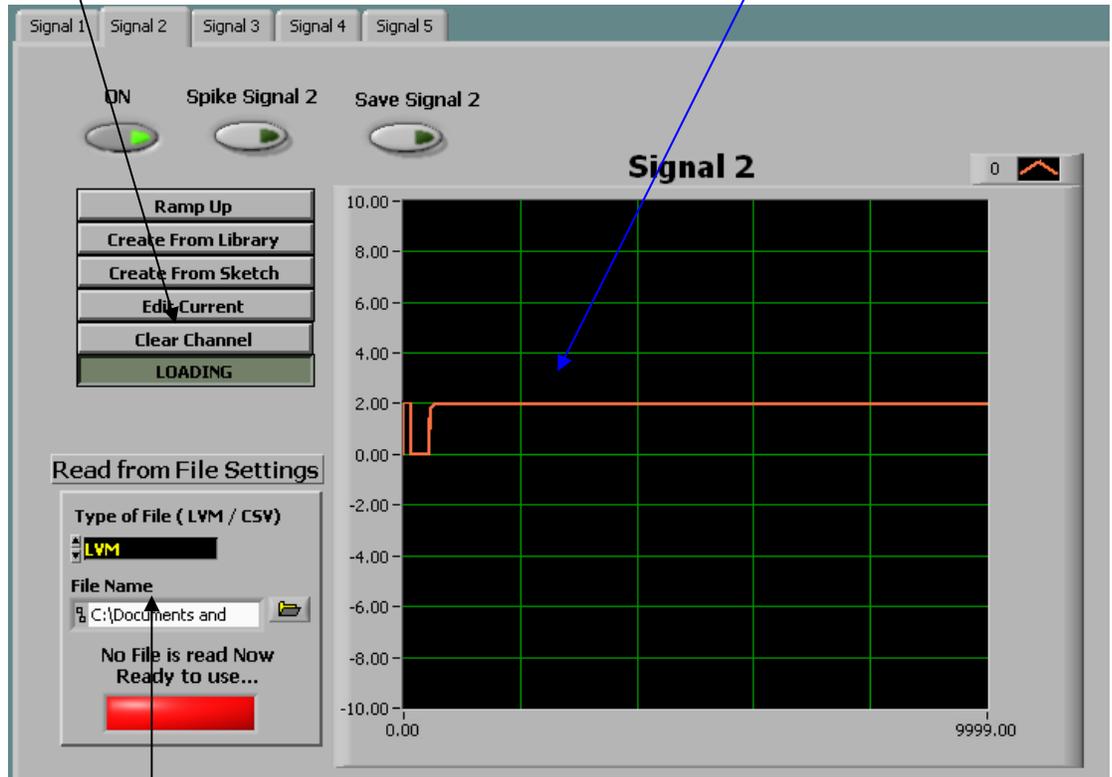


Primero presionar para cargar el archivo

Figura 10. Procedimiento para la carga de archivo

Posteriormente se presiona el botón Loading. El led de color rojo cambia a verde y cuando se carga la señal retorna a rojo

Señal cargada desde Archivo



Aparece la dirección (path) en donde se encuentra el archivo

Figura 11. Archivo Cargado

En esta fase de configuración de señal todas las señales tienen el mismo diseño a excepción de la señal 1, pues este canal es utilizado para generar señales con periodos de tiempo más grande.

2.3 FASE DE ADQUISICIÓN DE DATOS

En esta fase se adquieren las señales del mundo exterior haciendo uso de la tarjeta de adquisición de datos PCI 6143. En la figura 12 se observa el panel de control donde el usuario puede visualizar las gráficas de las señales obtenidas durante la adquisición.

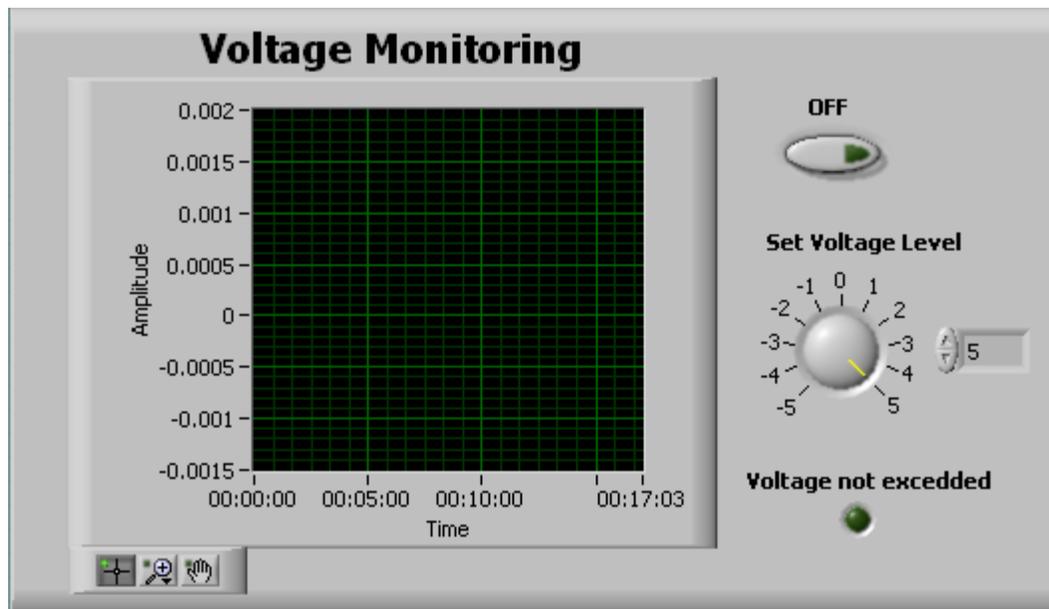


Figura 12. Funciones de adquisición de datos

Posee también un botón de encendido y apagado para la adquisición de datos, un waveform chart para la visualización de estos y una perilla para ajustar el voltaje de referencia así como un led que se enciende si el voltaje no se ha excedido.

Esto porque la tarjeta de adquisición de datos solo adquiere en un rango de -5 a +5 voltios. En caso de que la señal supere este voltaje o los voltajes seleccionados dentro de este rango el sistema se detiene por seguridad.

2.4 FASE DE GENERACIÓN DE SEÑAL

Es en esta fase donde las señales son generadas y visualizadas en un Waveform chart. Además aquí también se realizan otras funciones como el control de picos Spike signal de cada una de las señales e incluso el de todas las señales. Aquí también se salvan las señales de manera independiente.

En esta fase también se encuentra el control de salida de nivel de voltaje. Este control sirve para cuando se desee solo simular la generación o para sacar las señales por los amplificadores KEPCO.

2.5 FASE FINAL

Aquí se clarea la generación y todos los canales son llevados a 0 voltios así como también los controles son llevados al estado de apagado.

3. PROGRAMACIÓN

3.1 INICIALIZACIÓN

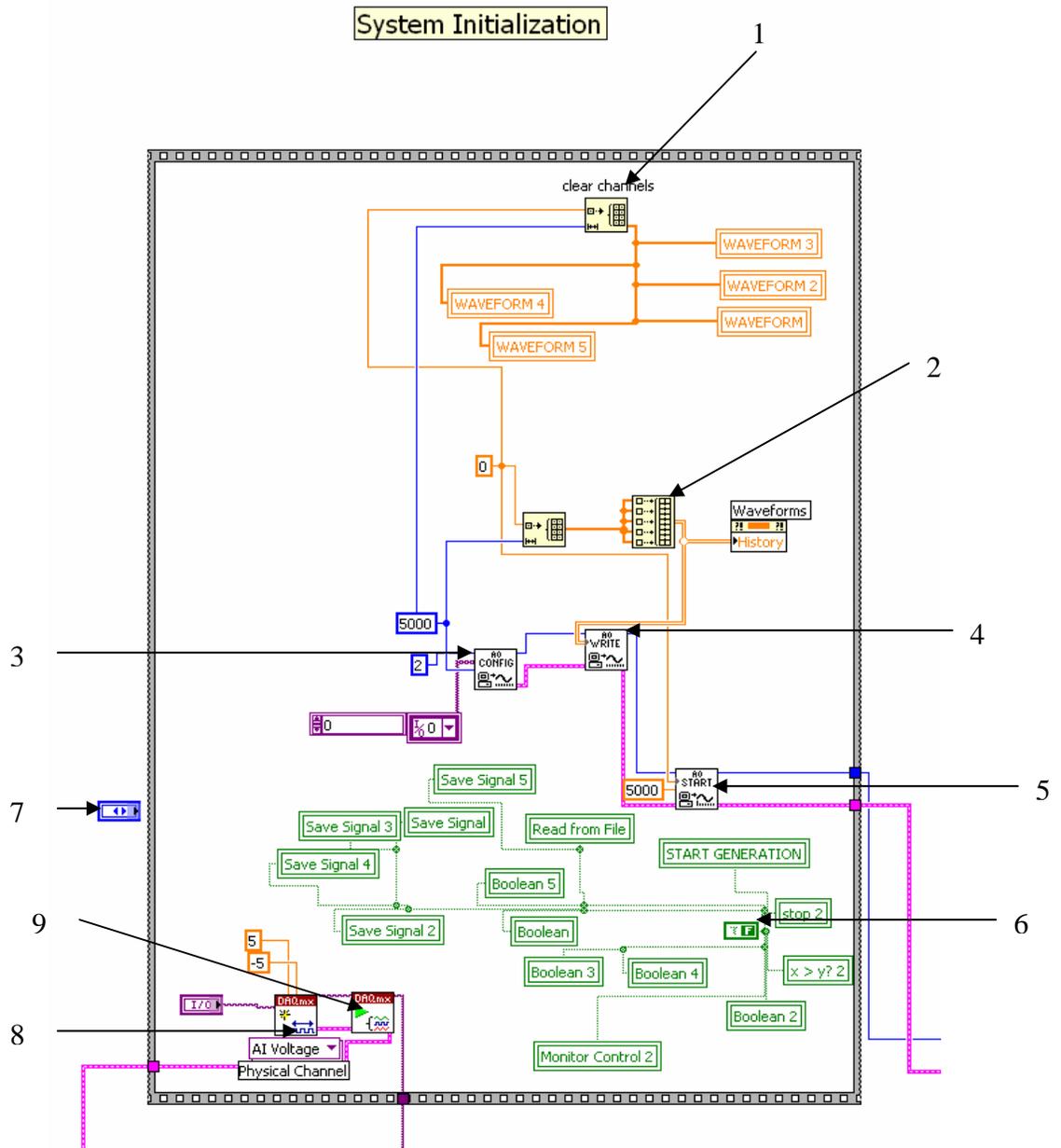


Figura 13. Programación fase de inicialización

- 1- Este elemento corresponde a un Initialize Array, esta herramienta se encuentra haciendo click derecho en el diagrama de bloques y seleccionando el icono de todas las funciones y posteriormente la opción de Arrays.

Esta opción permite iniciar un array con el valor que el usuario desea e incluso el tamaño que se desee.

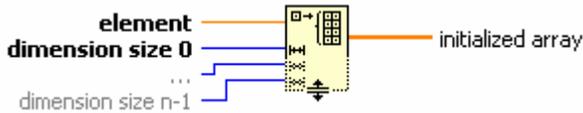


Figura 14. Initialize Array

En este caso como muestra la figura 13 se genera un array de 5000 posiciones inicializado con el valor de cero. Este tamaño del array es debido a que este es también el tamaño del buffer.

El array de ceros es mostrado en todos los Waveform Graphs que se encuentran dentro del Tab control.

- 2- Aquí se encuentra la herramienta de Build Array, la cual se encuentra en el mismo lugar donde esta Initialize Array. Esta herramienta permite crear una sola matriz a partir de otras matrices separadas.

Como se observa en la figura 13, se está creando una matriz a partir de cinco matrices inicializadas en cero y con 5000 posiciones también, esto con el fin de clarear el Waveform Chart colocando estos valores en el nodo propiedad en la opción History Channel e iniciar las condiciones iniciales de generación de señal para cada canal en cero voltios.

- 3- AO- Config. Esta función configura los canales de salida y ajusta los límites de salida, así como también permite configurar el tamaño del buffer para desarrollar las operaciones de salida analoga. Es en este punto donde se configura por programación la tarjeta de generación de datos.

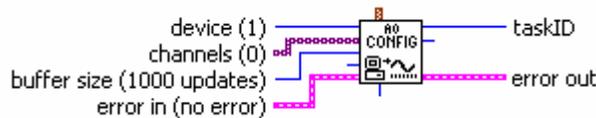


Figura 15. AO config

El AO - Config esta configurado de la siguiente manera:

- Tamaño del Buffer 5000
- Channels Constante por defecto.
- Elemento (Device) 2

En la parte de elemento, se considera como elemento número 2, puesto que al configurar las tarjetas de adquisición de datos utilizando el

Measurement & Automation Explorer, allí se programó la tarjeta PCI 6733, tarjeta generadora como elemento 2.

- 4- AO- Write. Con esta herramienta se escribe en el buffer los datos que se deseen estén en la salida de la tarjeta.



Figura 16. AO Write

En la figura 13 que correspondiente a la fase de inicialización, se escriben los primeros datos que corresponden a cinco matrices de 5000 puestos inicializadas en cero.

- 5- AO- Start. Este VI es el responsable de iniciar la generación. Aquí nuevamente se ajusta la taza de actualización del buffer, que para este caso es de 5000, pues se desean 5000 muestras por segundo.



Figura 17. AO Start

- 6- En este punto se colocan las condiciones iniciales para los botones del panel de control. Los botones que allí se muestran se van a inicializar en falso, es decir en la posición de apagado.
- 7- Este elemento corresponde al terminal del tab control en el diagrama de bloques. No se puede eliminar, de hacerlo las cosas contenidas dentro de él también se borran.
- 8- DAQmx Create Channel. Esta herramienta permite configurar la tarjeta de adquisición de datos. Aquí se crean los canales para la medición de voltaje.

De igual manera se configura por programación que la tarjeta de adquisición va a recibir señales de voltaje entre -5V a 5V.

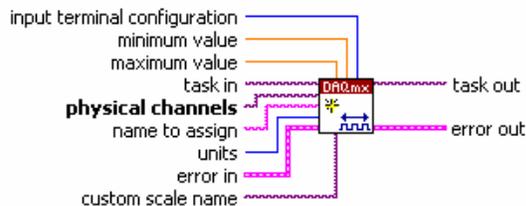


Figura 18. DAQmx create channel

9- DAQmx Start Task. Aquí es donde las tareas pasan al estado de ejecución para comenzar las mediciones.



Figura 19. DAQmx Start task

3.2 CONFIGURACIÓN DE SEÑAL

La configuración de señal es una programación bastante amplia, se encuentra dentro de un gran While loop aunque se puede mostrar por separado.

Básicamente la configuración de las cinco señales es la misma para todas excepto para la señal número uno que posee una opción menos.

Se empezará mostrando cómo es la programación de las señales 2 a la 5 debido a que es la misma programación con la diferencia que se cambian los nombres de las variables.

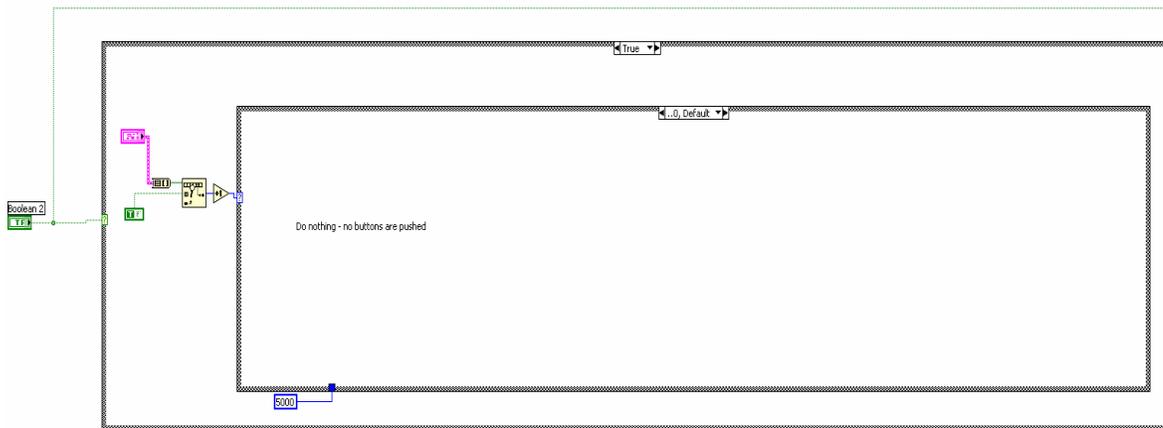


Figura 20. Programación configuración de señal 2

La figura 20 muestra la programación para las funciones de configuración de señal de cada una de las señales cuando se ha presionado el botón de encendido y no se ha seleccionado ninguna función de configuración. Estas funciones no se pueden ejecutar si no se presiona el botón de ON/OFF de cada señal independiente. El botón de On/Off del diagrama de bloques de Labview para la configuración de la señal 2 que se muestra en la figura anterior es el elemento en color verde nombrado Boolean2.

En el caso de que el botón de encendido de la señal no se haya presionado se muestra en el Waveform Graph correspondiente una matriz de ceros de 5000 posiciones.

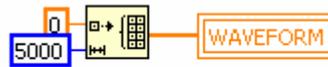


Figura 21. Creación de una matriz de ceros de 5000 posiciones

Si se presiona el Botón de encendido de la señal, se lee el cluster (donde se encuentran los botones de Ramp up, create from Library, Create from Sketch, Edit current y clear channel) para determinar cual función es la que se va a ejecutar.

Si no se presiona ningún botón del cluster entonces no se hace nada.

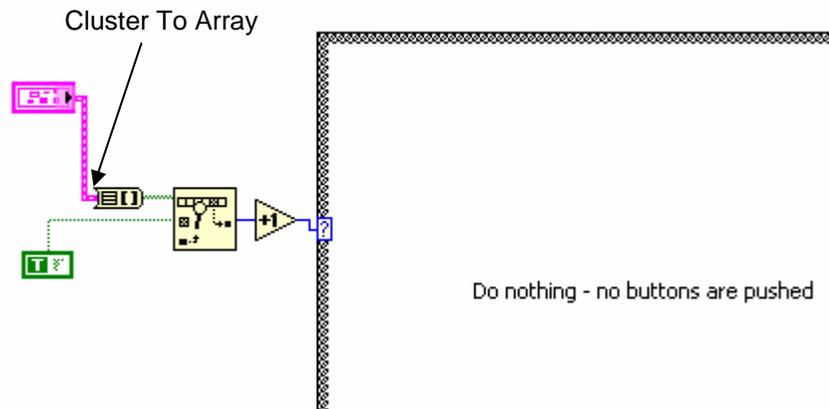


Figura 22. Búsqueda de botones dentro de un cluster

Para identificar que botón del cluster se presionó se utiliza la herramienta Search 1D Array. La cual busca cual de los botones se encuentra en verdadero a partir de un array booleano que es creado con la opción Cluster to array. La salida de esta opción es el valor de posición de la matriz donde se encuentra el valor de verdadero.

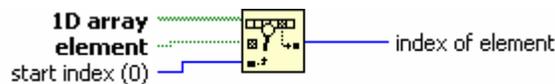


Figura 23. Search 1D Array

Cuando se utiliza un cluster, este ordena los elementos que se agreguen dentro de él dependiendo el orden con que se hallan introducido por defecto, es decir, por ejemplo si se tienen tres botones, el botón A, el botón B y C y se van a agregar al cluster de manera aleatoria primero el botón B, luego C y finalmente A, al utilizar la función cluster to array se crea un array que contiene los valores boléanos de estos en el mismo orden con que se introdujeron los botones B, C y A, aunque,

Labview permite también reordenar lo elementos del cluster como el usuario desee.

Para el Generador multivoltaje el orden es el siguiente:

Array	BOTON OPRIMIDO
0	No se oprimió ningún botón. No pasa nada
1	Crear desde librería
2	Crear arbitrariamente
3	Edit current
4	Ramp up
5	Clear channel
6	Leer desde archivo

Tabla 1. Orden de los elementos agregados a un cluster

Para determinar cual situación desde cero a seis es la que se debe ejecutar se utiliza un una estructura Case, que cambia al valor del número que se le coloque en la entrada.

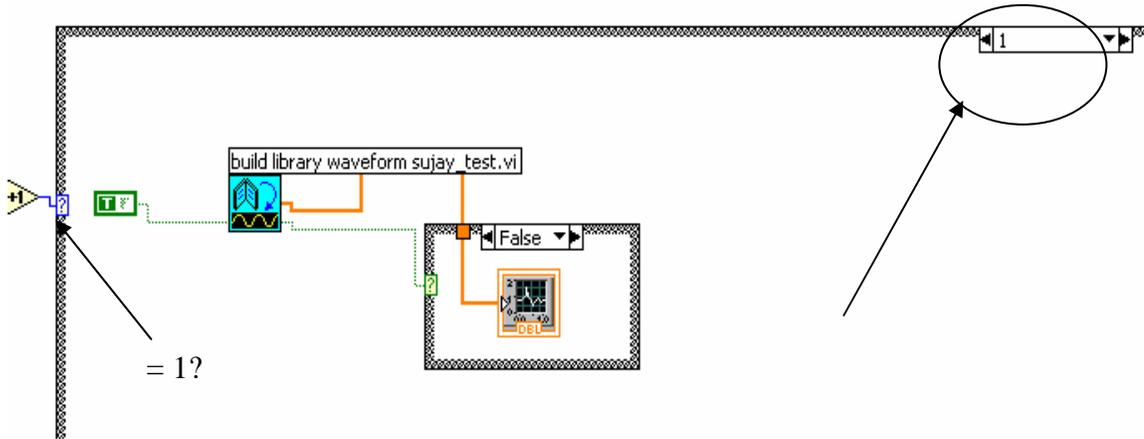


Figura 24. Utilización del case structure para la selección de los botones del cluster

Por ejemplo si el valor a la entrada del Case es igual a uno, este se posiciona en 1 luego quiere decir que se oprimió el botón de Create from library o crear desde librería.

En la figura 24 se observa que se llama a un subvi, donde se encuentra la programación para crear desde librería y posteriormente se envía una matriz con los valores de la señal seleccionada y modificada al terminal de la señal correspondiente que se este trabajando.

En caso de que en el subvi se cancele la operación no pasa nada.

Si el case se cambia a dos quiere decir que se oprimió el botón de crear arbitrariamente o Create from Sketch.

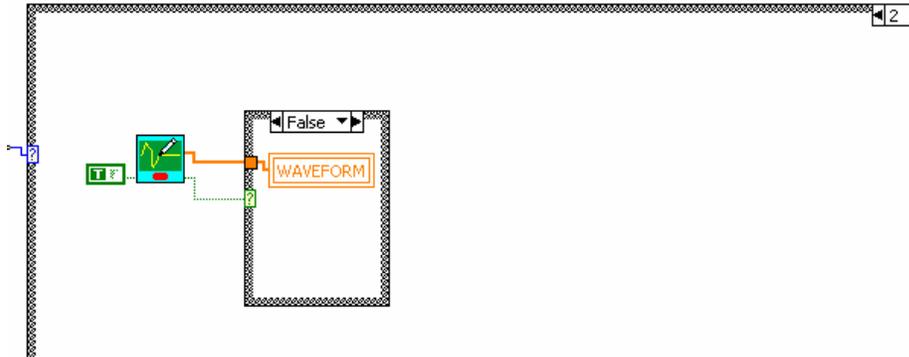


Figura 25. Selección del botón create from Sketch

De igual manera se llama a un subvi que realiza la programación para crear una gráfica utilizando el Mouse. Si la acción es cancelada entonces no pasa nada.

De manera similar ocurre con el botón edit current, que al igual que en el caso anterior utiliza el mismo subvi pero con una pequeña modificación que es la que hace la diferencia, ahora se toma la señal que se encuentra dibujada en el Waveform Graph para ser luego modificada en el subvi.

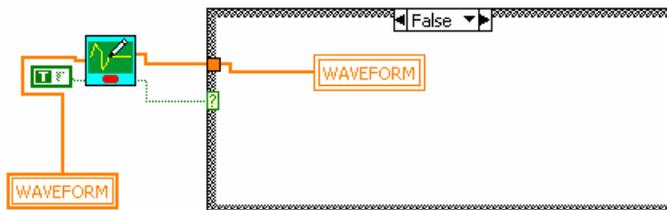


Figura 26. Edit Current

Al oprimir el botón de Ramp up, nuevamente a partir de otro SubVI el usuario tiene la posibilidad de crear una rampa haciendo uso de los controles allí presentes pero la señal de rampa que se origina en el subvi no se encuentra en un factor de 5000, por lo que se hacen arreglos para convertir la señal a este factor.

La figura 27 muestra la programación para convertir la señal en un factor de 5000, básicamente lo que realiza el programa es preguntar por el tamaño de la matriz inicial, si esta es menor que 5000, completa el array hasta 5000 relleno con el último valor de la matriz inicial, caso contrario si es mayor a 5000 completa un array con 5000 muestras y posteriormente las restantes las completa en otro array de 5000 muestras para reescribir en el buffer.

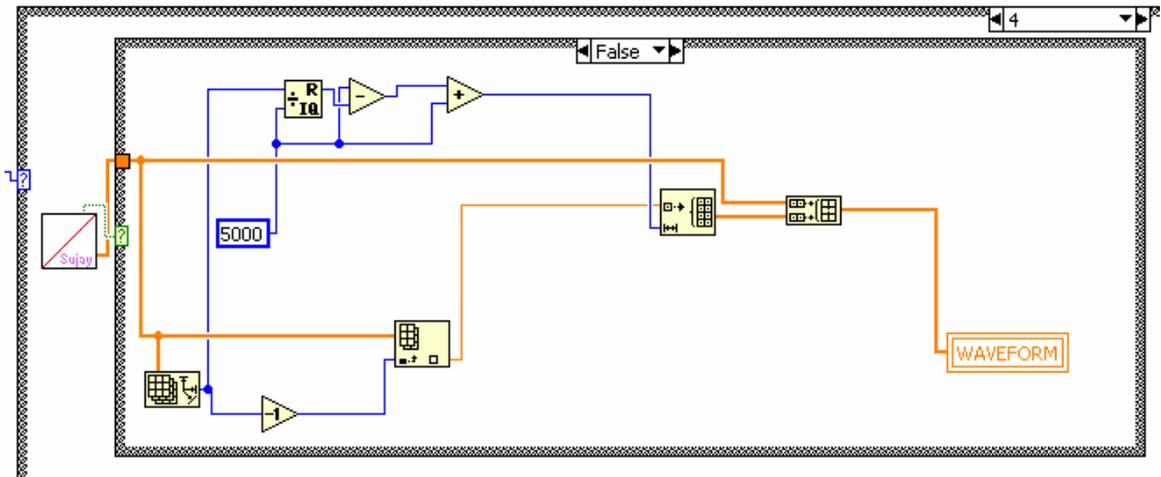


Figura 27. Conversión a un factor de 5000

Para Clear Channel, se crea una matriz de 5000 posiciones llena de ceros haciendo uso del Initialize Array. El array de salida es mostrado en el Graph de la señal correspondiente.

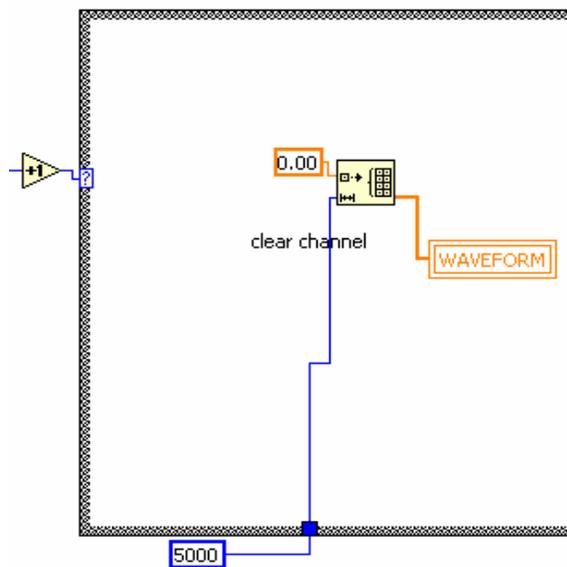


Figura 28. Programación para clarear el canal

Finalmente si el usuario desea leer una forma de onda contenida en un archivo, el programa pregunta por el tipo de archivo ya sea LVM o CVS, hecho esto el archivo es leído utilizando el Read Labview Measurement file, el cual dará la señal de que el archivo ha sido leído al colocar un valor verdadero en una de sus salidas llamada end of file (EOF) la cual se aprovecha para encender un led para avisar al usuario que se esta leyendo el archivo. El único inconveniente es que solo se puede leer el archivo solo una vez.

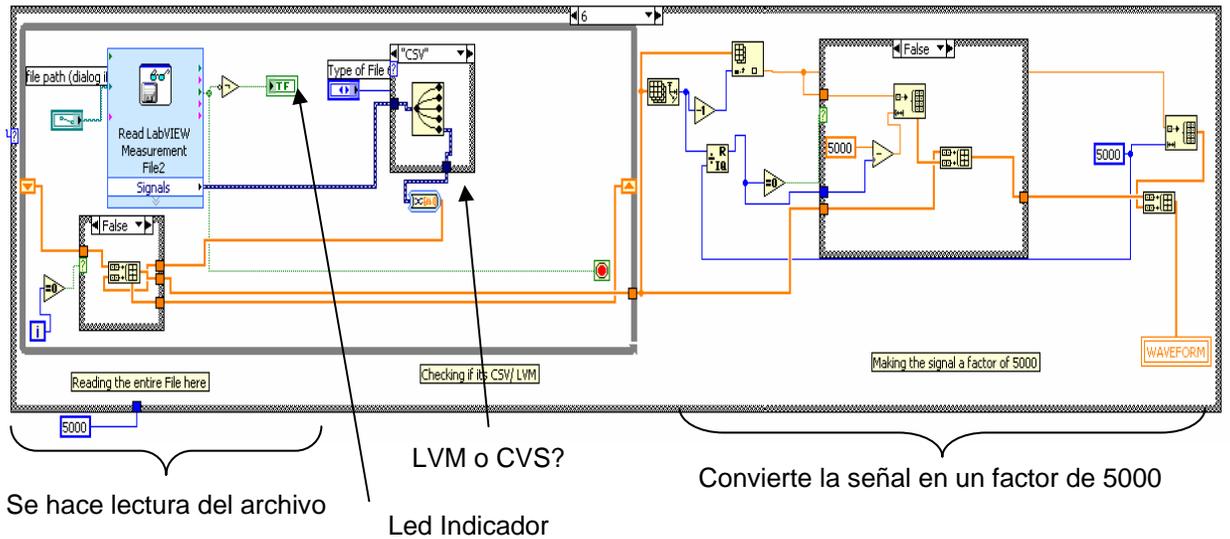


Figura 29. Programación para el Read from File

Como se observa en la Figura 30, para leer el archivo es necesario introducir la dirección o el path donde se encuentra el archivo, este es leído por el Labview Measurement file en su entrada de File Name.

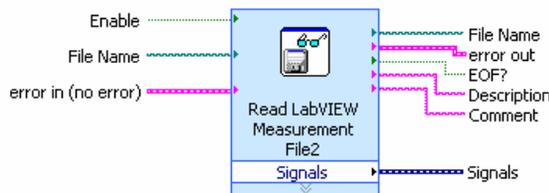


Figura 30. Read from labview Measurement File

Una leído los datos, estos son puestos en un array de una dimensión para luego convertirse a un factor de 5000 en caso de que el archivo no posea las 5000 muestras y después es visualizada por el Waveform Graph de la señal que se esta trabajando.

3.3 FASE DE ADQUISICIÓN DE DATOS

En esta fase se encuentra el código para el Voltage Monitoring. Aquí se visualizan los datos que se están adquiriendo. Posee su propio botón de encendido o de on/off que lleva a un case el cual por ejemplo si se encuentra encendido, adquiere el código fuente de la señal de voltaje de la tarjeta de adquisición de datos PCI6143 y lo compara con el valor de voltaje ajustado por el usuario en la perilla.

Hay que recordar que la adquisición solo se puede hacer dentro del rango de voltaje de -5V a +5V, en caso de que la señal se salga de este rango o supere el valor ajustado por el usuario el programa se detiene inmediatamente.

Esta fase se presenta una etapa adicional que es la etapa de clareo de adquisición, es allí donde se clarea la adquisición cuando ya no se necesita o no se esta utilizando la tarjeta de adquisición de datos.

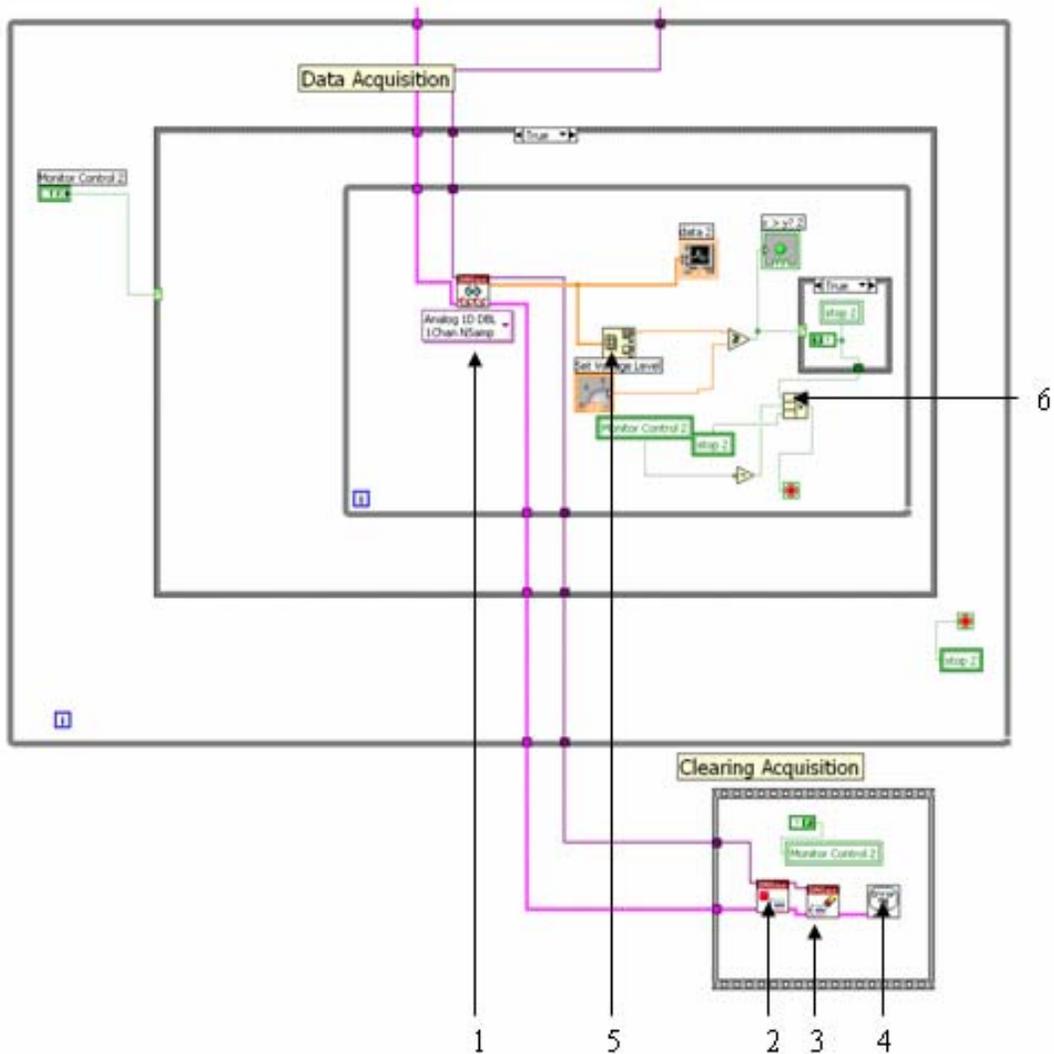


Figura 31. Programación fase de adquisición de datos

- 1- DAQmx Read. Lee una o mas muestras flotantes desde una tarea que contiene una sola entrada análoga.

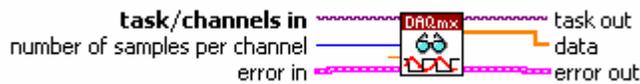


Figura 32. DAQmx Read

- 2- DAQmx Stop Task. Detiene la tarea de adquisición y la regresa al estado que se encontraba antes de que se corriera el DAQmx Start o el DAQmx Write con la entrada de autostart en verdadero.



Figura 33. DAQmx Stop Task

- 3- DAQmx Clear Task. Clarea la tarea. Antes de clarear, este Vi, detiene las tareas si es necesario y libera cualquier recurso que la tarea hubiese reservado. No se puede usar la misma tarea después de clarear a menos de que se vuelva a crear.



Figura 34. DAQmx Clear Task

- 4- Simple Error Handler. Indica cuando un error ha ocurrido. Si un error ocurre el VI regresa una descripción del error y opcionalmente muestra un cuadro de dialogo.

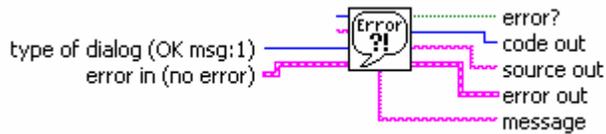


Figura 35. Simple Error Handler

- 5- Array Max & Min. Regresa el valor máximo y mínimo encontrado en un array junto con la posición de cada valor.
- 6- Compound Arithmetic. Ejecuta la aritmética de uno o más números, arrays, clusters o entrada booleana.

3.4 FASE DE GENERACIÓN DE SEÑAL

Aquí se generan las señales que ya fueron configuradas. Además de la generación de señal se realizan otras actividades como el control de picos (Spike signal) para cada señal independiente o para todas las señales al tiempo. También se revisa que la señal no se pase de 5000 muestras, se ejecuta la acción de salvado de la señal y se maneja el control para el voltaje de salida, un factor que hay que considerar cuando se usa el amplificador KEPCO o cuando no.

3.4.1 GENERACIÓN DE SEÑAL - CONCEPTO BÁSICO UTILIZADO EN EL GENERADOR MULTIVOLTAJE

Cuando el nivel de la señal en la salida es mas importante que la tasa de cambio a la cual los valores cambian, es porque se necesita generar valores de DC constantes. Se puede generar utilizando el VI Single point analog output, sin embargo solo se puede cambiar la salida tan rápido como labview llame a este VI.

Esta técnica es llamada Software Timing y se debe usar si no se necesita alta velocidad en la generación de un tiempo muy exacto.

Algunas veces en la ejecución de la salida análoga, la rata a la que se actualizan los datos ocurre justo en el nivel de voltaje que se considera que es importante.

Esto es llamado generación de forma de onda (Waveform Generation) o salida análoga bufereada (Buffers analog output), utilizada en el Generador Multivoltaje.

Para ejecutar esto es necesario por ejemplo almacenar un ciclo de la onda seno en un array y programar el elemento del DAQ para generar valores continuos en el array un punto al tiempo especificando la taza de generación¹.

La estructura con la que se programó el Generador Multivoltaje (Waveform Generation) esta conformada de la siguiente manera como se muestra en el siguiente ejemplo:

¹ Tomado de Data Acquisition Basic Manual. National Instruments. Pág. 10-1

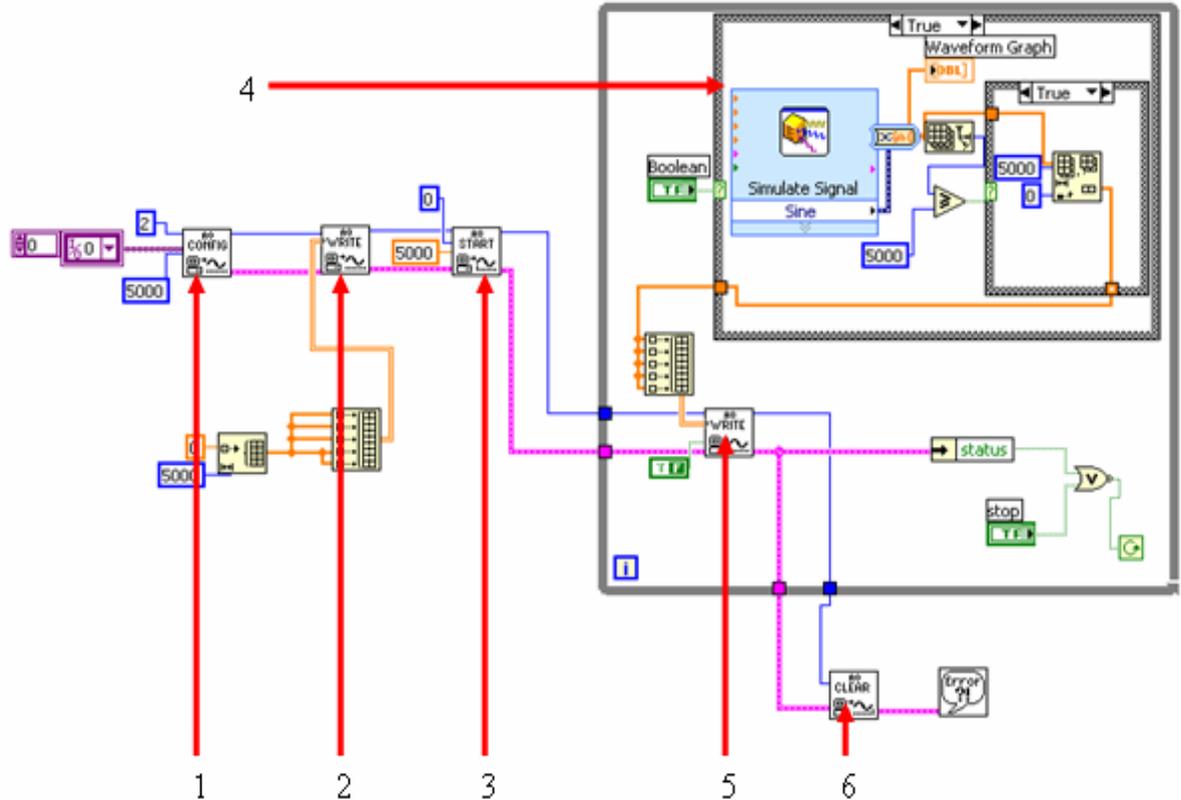


Figura 36. Ejemplo de generación de forma de onda, programación básica utilizada en el Generador Multivoltaje.

- 1- Se utiliza el AO config con el propósito de configurar la tarjeta de adquisición de datos. Allí se configura el número de muestras por segundo y los canales.
- 2- Con el AO write se escriben los datos que se van a generar, en este caso y en esta parte de la programación este funciona para escribir las condiciones iniciales.
- 3- Con el AO Start se inicia la generación.
- 4- Corresponde a un Case Structure dentro del cual se puede ejecutar la programación con la cual se desea modificar los datos de la señal. El while loop fuera del Case Estructure permite entregar dato por dato de manera continua para ser generado por la tarjeta PCI generadora hasta cuando el usuario desee o el programa sea detenido por algún motivo.
- 5- Usando AO Write nuevamente, se reescriben los nuevos valores que se quieren generar.

- 6- Finalmente, al terminar se clarea la salida de la tarjeta para que no se quede en el nivel de voltaje con el que se detuvo la generación.

En el programa principal del Generador Multivoltaje la generación de señal esta contenida dentro de un case Structure muy grande por lo que hace que la programación de esta fase sea demasiado extensa. La generación de datos solo se ejecuta cuando el botón de Start Generation es oprimido.

En caso de que no se oprima el botón se generan cinco matrices de 5000 posiciones de valor cero y se reescribe en el buffer haciendo uso de la herramienta AO- Write, como se muestra a continuación.

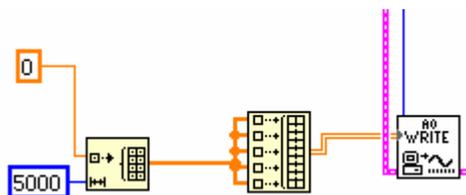


Figura 37. Programación cuando no se oprime el botón de empezar generación

Si el botón de generación de señal es oprimido, entonces el case cambia a verdadero y se realizan los siguientes pasos con el respectivo orden:

- 1- Determinar el tamaño de la matriz de la señal. Si la longitud de la señal es superior a las 5000 muestras, entonces se generan las primeras 5000 y posteriormente las siguientes 5000 en un ciclo subsiguiente y consecutivo para cada una de las cinco señales de manera independiente.

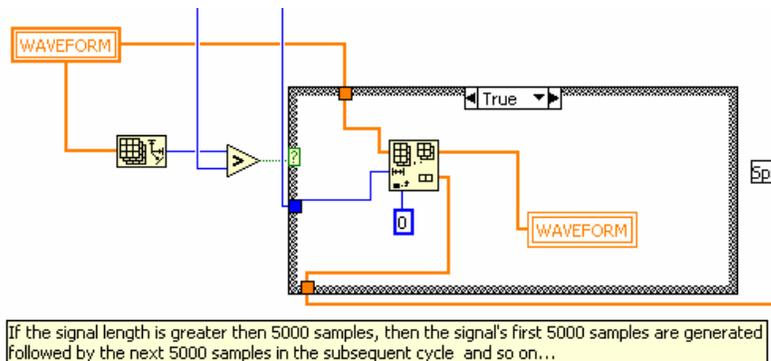


Figura 38. Delimitación del array de señal a 5000 muestras

- 2- Spike signal (De cada señal individual). Si se ha habilitado la opción de Spike signal de una de las señales, el Index Array toma solo el primer valor de la señal original y este será entregado al subvi de generación de rampa que en este caso se llama trigger, el cual entrega dos salidas una de Stop y la otra con los valores del nuevo array modificado para ser reemplazado posteriormente.

Si el botón de spike signal no se oprime no pasa nada.

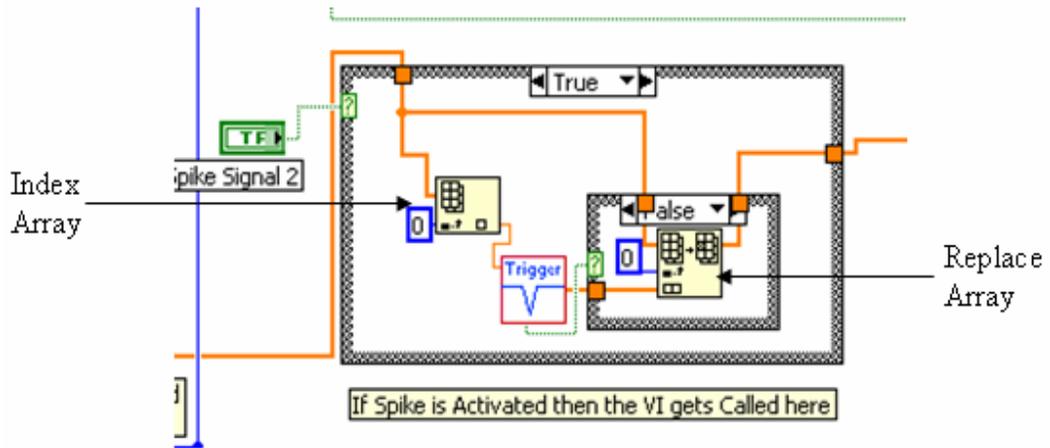


Figura 39. Spike Signal

- 3- Save signal Button. Es aquí donde se puede salvar la señal que se esta generando, para esto se utiliza el VI Write Labview Measurement file para guardar la señal en formato LVM.

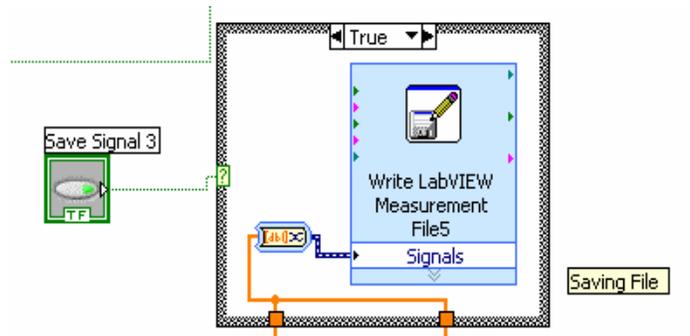


Figura 40. Guardar a un archivo LVM

- 4- Spike All Signals. Cuando se oprime este botón, las señales que están siendo generadas serán reemplazadas por los nuevos valores de matriz a generar. Esta opción no se ejecuta para aquellas señales que no estén encendidas en la fase de configuración de señal.

El subvi que realiza esta programación es similar al utilizado para realizar el Spike a una señal solo que posee nombre y forma diferente para diferenciar.

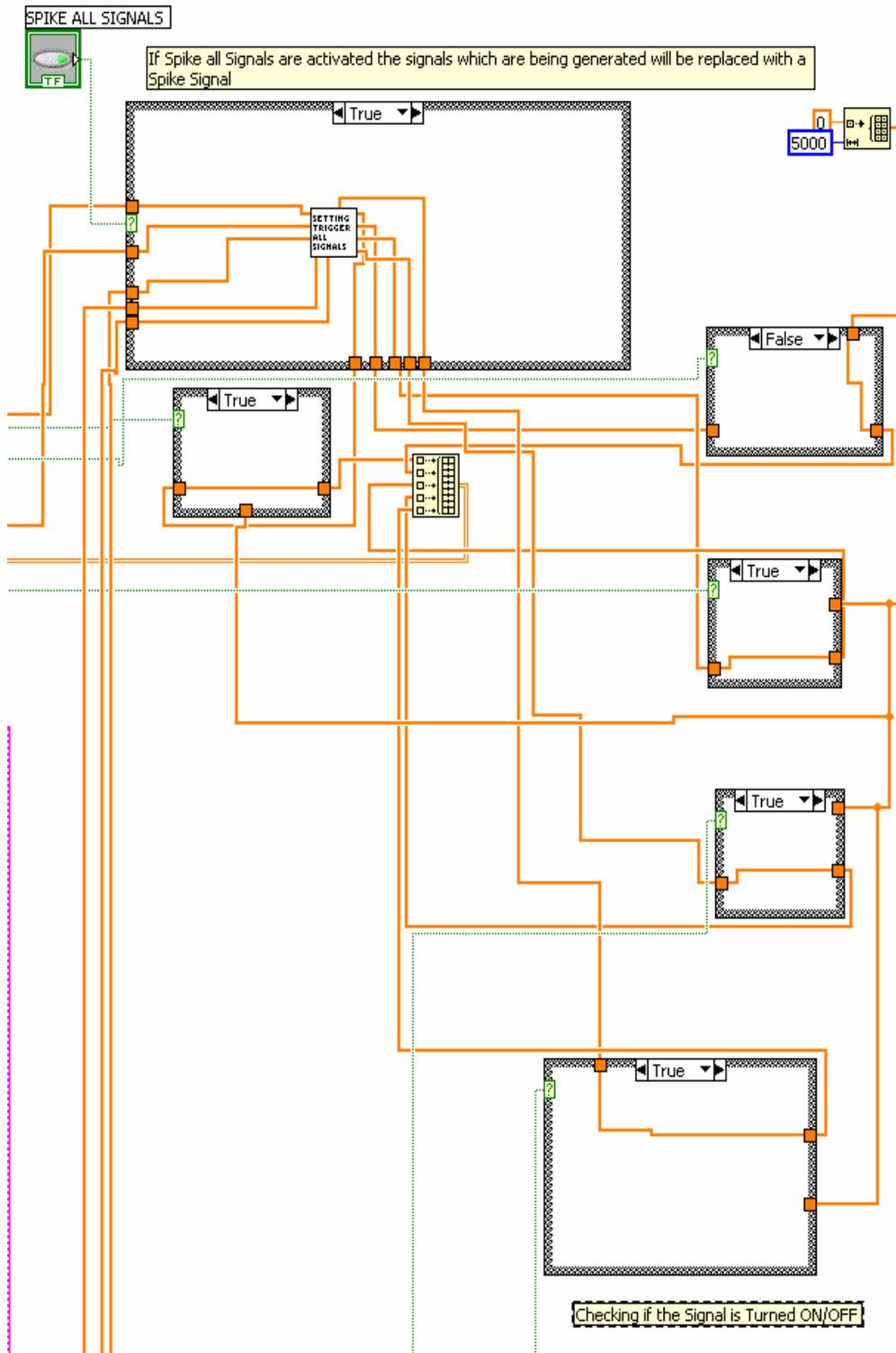


Figura 41. Spike all Signals

- 5- Output voltage Level. Con este control se determina si se va a usar el amplificador KEPCO o si solo se desea simular.

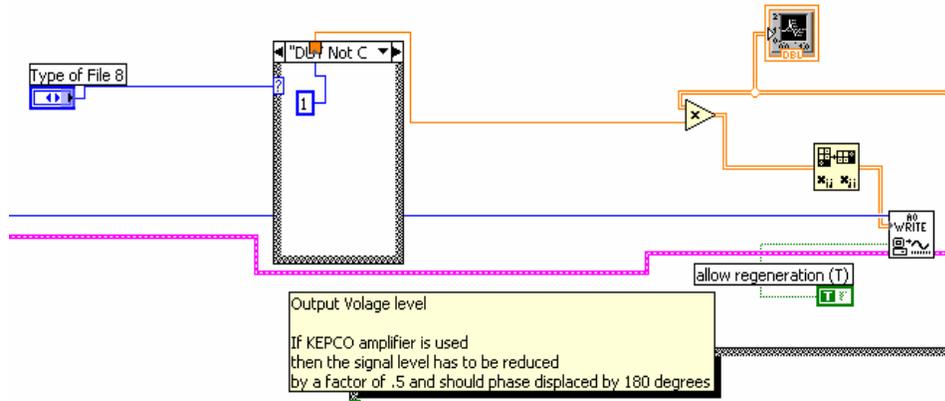


Figura 42. Output Voltage level y re-escritura en el buffer

El amplificador KEPCO tiene la característica que amplifica la señal al doble y la desfasa 180°, por lo que se debe multiplicar por -0.5 en la programación para entregar la forma de onda con los valores para los cuales fue configurada.

Cuando no se usa el amplificador KEPCO se multiplica por 1 para conservar la misma ganancia y fase y se puede simular la generación para tener una idea de cómo son las formas de onda y como van a actuar.

Se utiliza la herramienta de Transpose Array porque cuando se representa más de una gráfica en un mismo Waveform chart utilizando arrays, este interpreta por defecto las filas como gráficas diferentes. Pero si lo que interesa es que sean las columnas las gráficas diferentes, se utiliza este comando para convenir las columnas en filas.

Posteriormente se reescribe el buffer con los valores de las nuevas señales, utilizando para ello el AO Write nuevamente.

En esta fase también se ejecutan los comandos especiales de la señal 1. El cual consiste en generar la señal desde archivo en el momento de la generación.

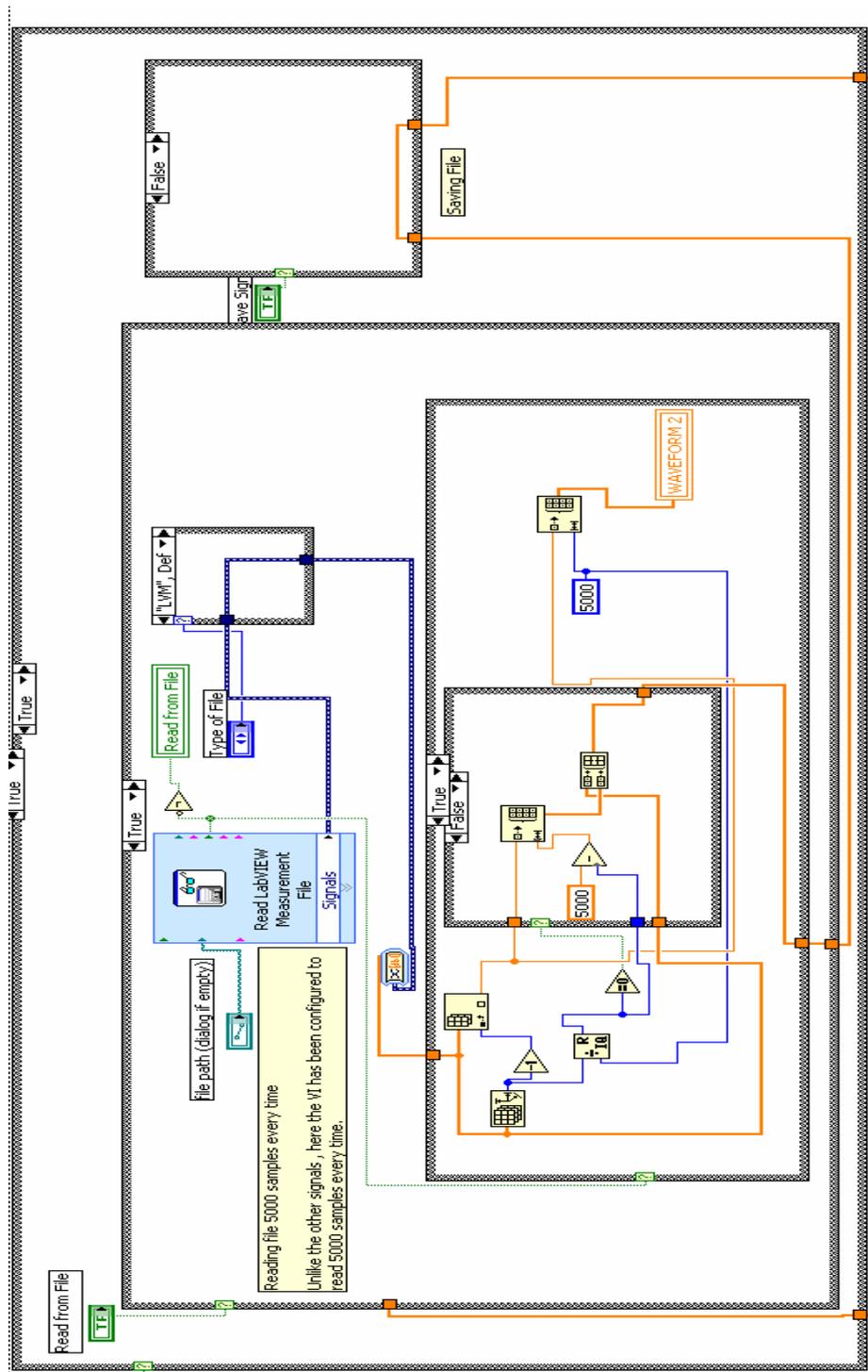


Figura 43. Read from file Señal 1

Aquí se lee el archivo de 5000 muestras todo el tiempo siempre y cuando el botón de generación de señal se encuentre oprimido. Una característica de esta forma de lectura de archivo es que permite leer un archivo de más de 5000 muestras.

3.5 FASE FINAL

Aquí la PCI 6733 tarjeta generadora de señal es clareada y todos los canales son llevados a cero voltios, así como también todos los controles son llevados al estado de apagado.

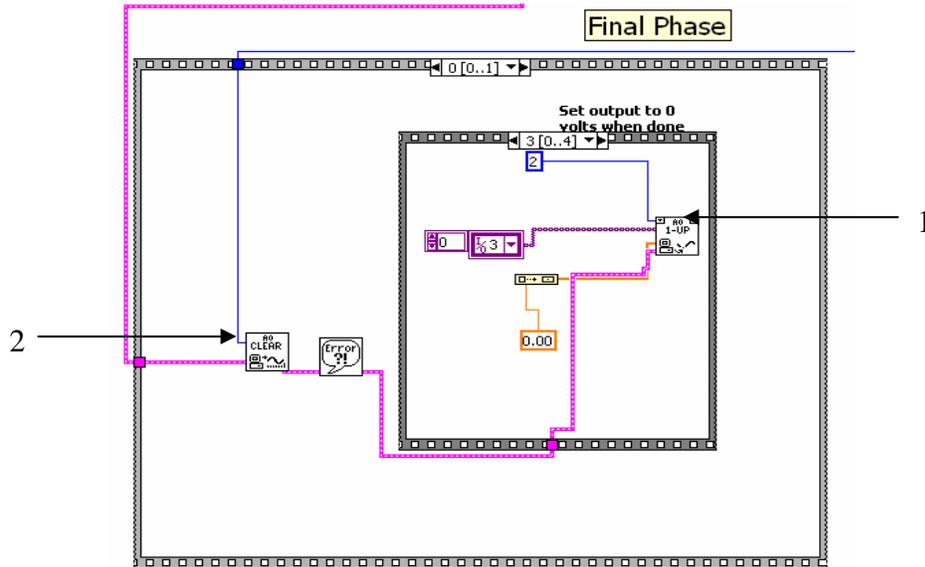


Figura 44. Programación fase final

- 1- AO 1-UP. Escribe una sola señal de voltaje para cada valor especificando los canales de salida.

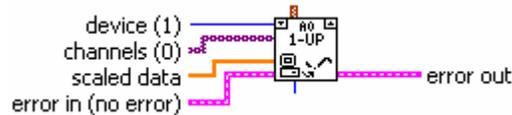


Figura 45. AO – 1UP

- 2- AO Clear. Borra las tareas de salida análogas.



Figura 46. AO Clear

La figura 47 muestra como son clareadas las variables booleanas, esto debido a que por seguridad se debe retornar los controles y los botones del generador a su estado de apagado antes de detener el programa.

Para realizar esto basta solo con conectar los botones o sus variables locales directamente al valor booleano de falso.

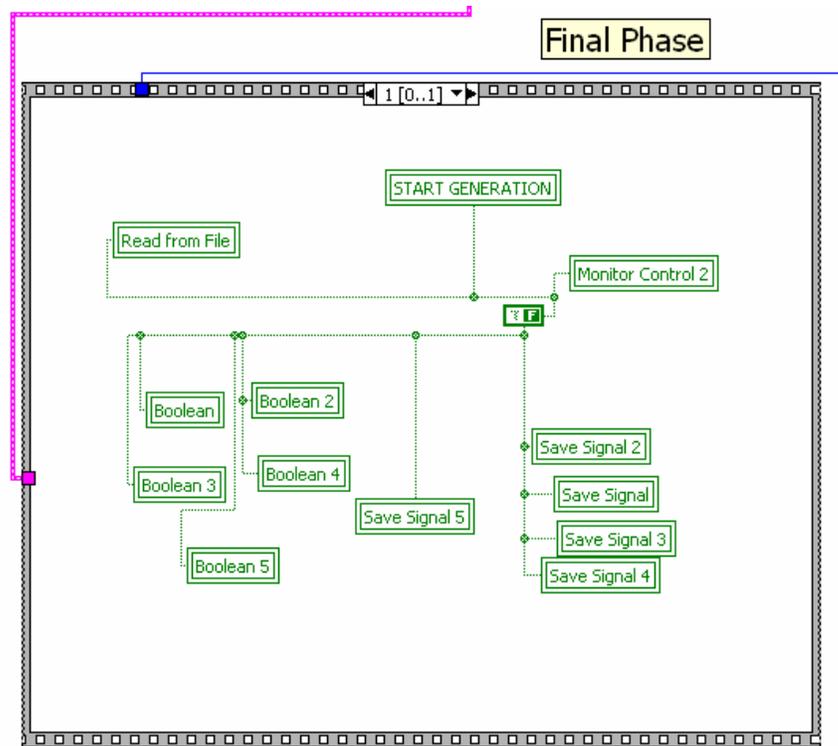


Figura 47. Clareado de las variables booleanas

3.6 CÓMO DETENER EL GENERADOR MULTIVOLTAJE

El sistema se detiene cuando hay algún error en el Voltage monitoring, que como se mencionó anteriormente este error se produce porque el voltaje adquirido supera el voltaje ajustado por el usuario o si se presiona el botón de encendido y apagado del sistema.

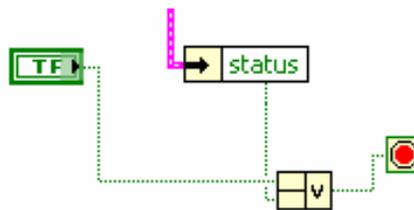


Figura 48. Detención del programa

4. MODIFICACIONES REALIZADAS

4.1 SALVADO Y RECARGA DE LA CONFIGURACION DE SEÑALES

Uno de los principales inconvenientes que existían en el generador multivoltaje, consistía que para recargar la configuración de las señales en caso de querer hacer la misma prueba consecutivamente o muchas veces había que configurar las señales una por una.

Para ello se implementaron dos nuevos botones. El botón Save y el botón Reload.

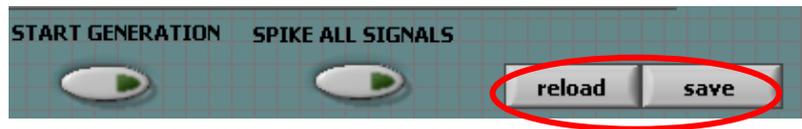


Figura 49. Botones de recarga

La idea principal, consistía en tomar las matrices con los valores de cada señal de la fase de configuración de señal y reunir las en una sola matriz para guardarlas en un solo archivo.

Cuando se quisiera leer se abriría el archivo y se separarían las matrices para recargarlas nuevamente en los Waveforms graphs de configuración de señal de forma correspondiente como se había configurado.

Inicialmente se utilizó un Measurement Read File y Save file, VIs Express para salvar y leer los archivos, aplicando la siguiente programación:

En la figura 50 se simula los datos de siete señales contenidos en uno solo array.

Estos valores entran al VI Express Write Labview Measurement file, para ser guardados.

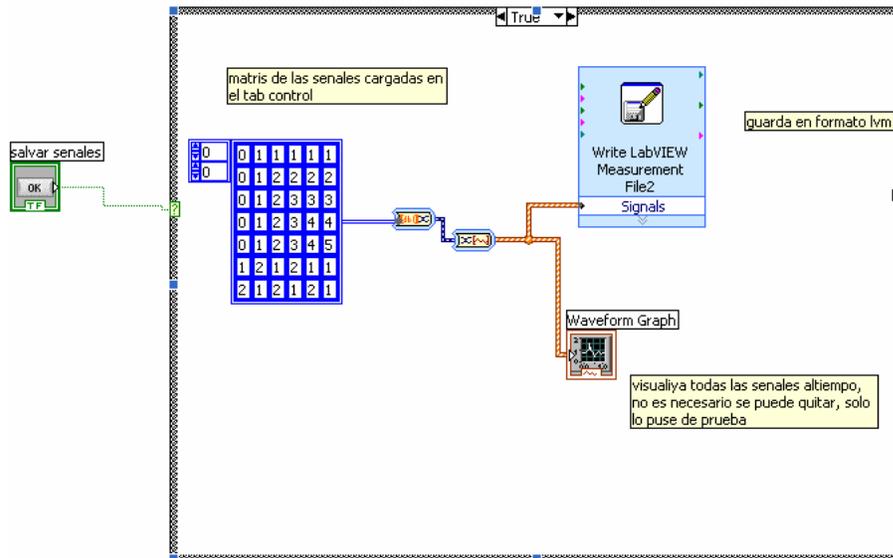


Figura 50. Guardando varias señales en formato LVM

En caso de que el botón de Salvar no se presione se pregunta por el botón de Recarga de las señales o Reload. Si este tampoco se oprime no pasa nada.

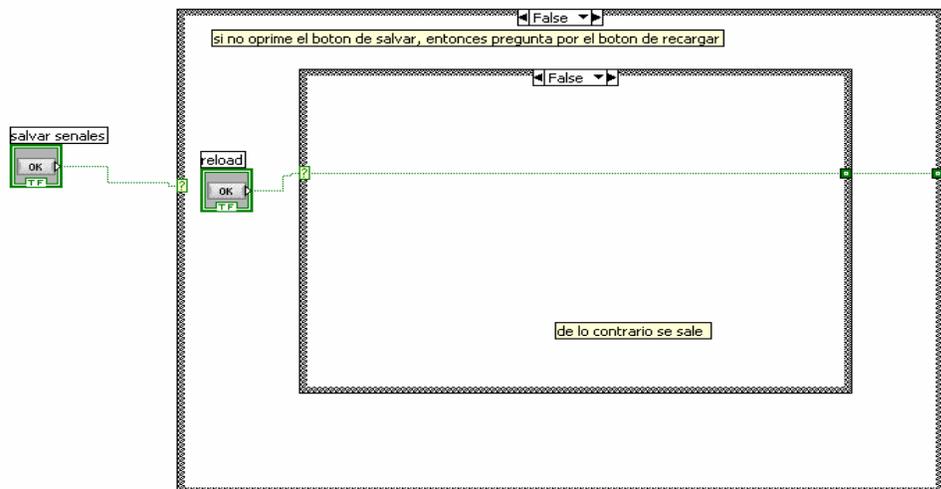


Figura 51. Save y Reload no se han presionado

Si por otra parte el usuario quisiera recargar las señales, haciendo uso del Read Labview Measurement file VI se leería el archivo y se separarían las señales utilizando la herramienta de Split signals para luego ser visualizadas en los Waveform Graphs de la fase de configuración de señal.

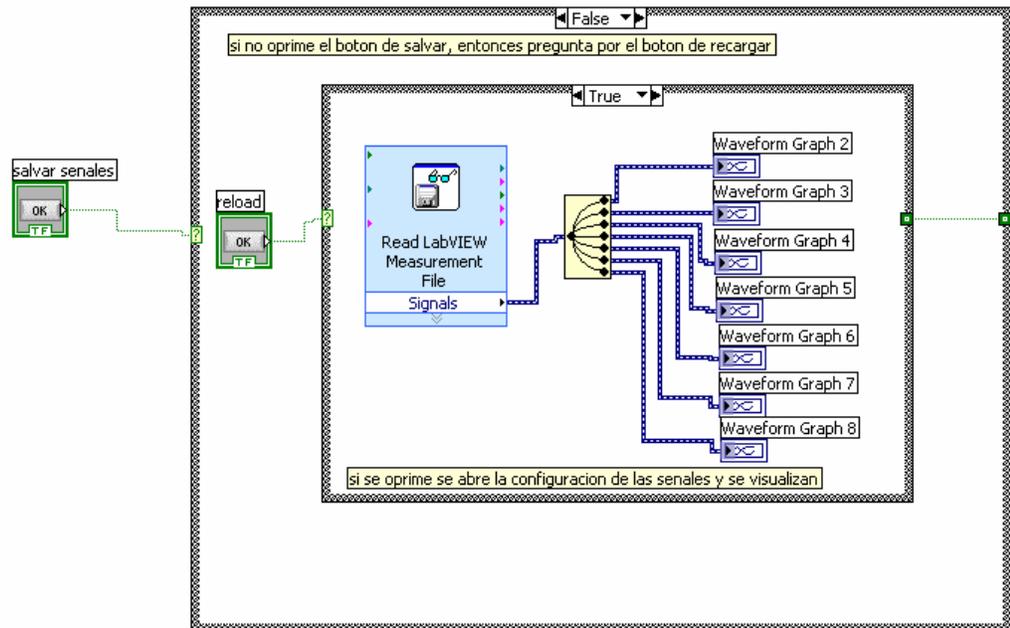


Figura 52. Leyendo el archivo con las señales

Con esta programación se presentaron dos errores. El primero consistía en que si se trataba de recargar otra configuración o incluso la misma que se había recargado, aparecía una ventana de error y se detenía el programa.

De igual manera si el usuario por ejemplo había oprimido por equivocación uno de los dos botones el de save o reload, y quería salir de allí presionando la opción de cancelar, se presentaba otro error nombrado por Labview como error 43, es un error por cancelación, "Operación cancelada por el usuario" presenta dos opciones la primera de continuar corriendo el programa y la segunda de detenerlo.

Al tratar de corregir los errores se encontró que debido a que se utilizan VI's Express para salvar y leer las señales, las opciones que estos presentan son muy limitadas y modificar su programación interna resulta en una tarea muy complicada pues este tipo de VIs, poseen internamente más SubVI.

Como solución se implementaron las herramientas de Write to Spreadsheet File.vi y Read from Spreadsheet File.vi. Estos elementos se pueden conseguir haciendo click derecho en el diagrama de bloques, seleccionar la opción de todas las funciones y allí seleccionar Archivos de entrada salida I/O.

- Read from Spreadsheet File.vi. Lee un número específico de líneas o filas de un texto numérico empezando en un carácter especificado y convierte los datos a dos dimensiones, o a un array de una dimensión o números.

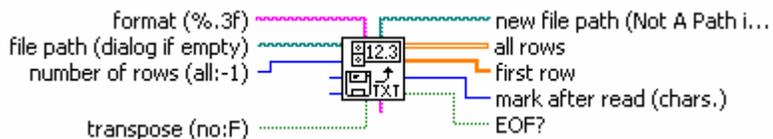


Figura 53. Read From Spreadsheet File

- Write from Spreadsheet File.vi.

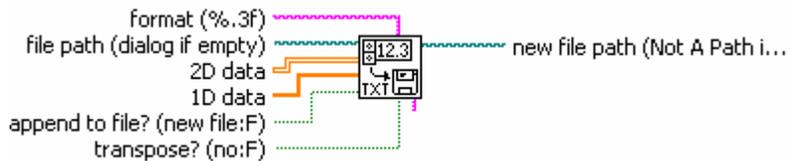


Figura 54. Write from Spreadsheet File

Se maneja el mismo concepto e idea de programación, solo se reemplazan los VI de escritura y lectura, aunque se poseían los mismos errores que VIs Express de Read/Write measurement file, con los read/write Spreadsheet File se hacia mas fácil corregirlos.

Para eliminar el error al dar cancelar se introduce el Write from Spreadsheet dentro de un case, el cual pasa a la posición de verdadero cuando recibe un valor booleano proveniente de un File dialog al oprimir cancelar. Y para salvar o reescribir en el mismo archivo se le coloca la opción de archivo nuevo o existente en el File dialog.

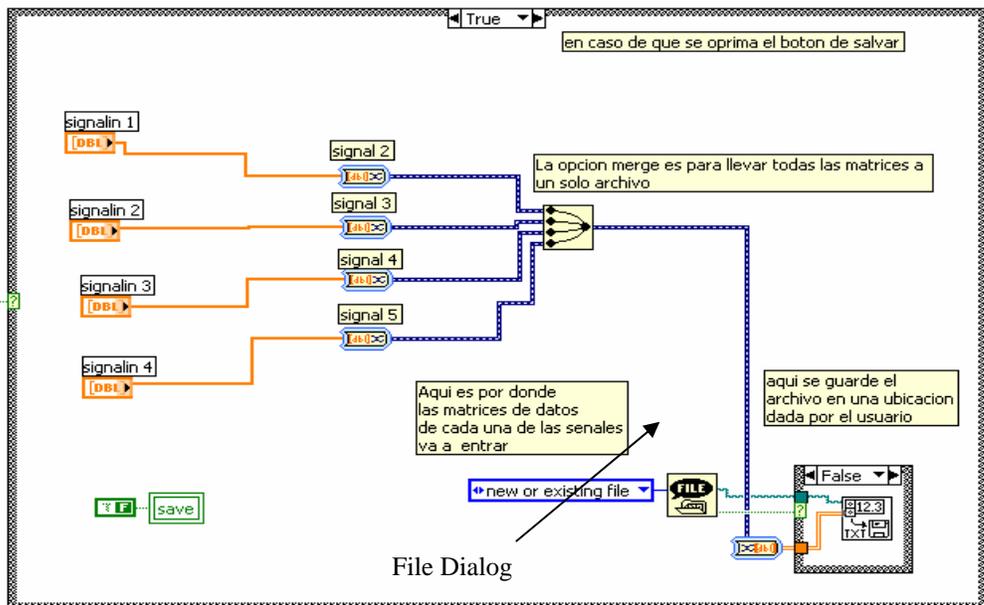


Figura 55. Corrección al dar cancelar con el File Dialog

De la misma manera que arriba a la hora de leer se utiliza la misma estrategia para corregir el error al dar cancelar, pero para abrir el mismo archivo se le coloca al file dialog la opción de Existing file o archivo existente.

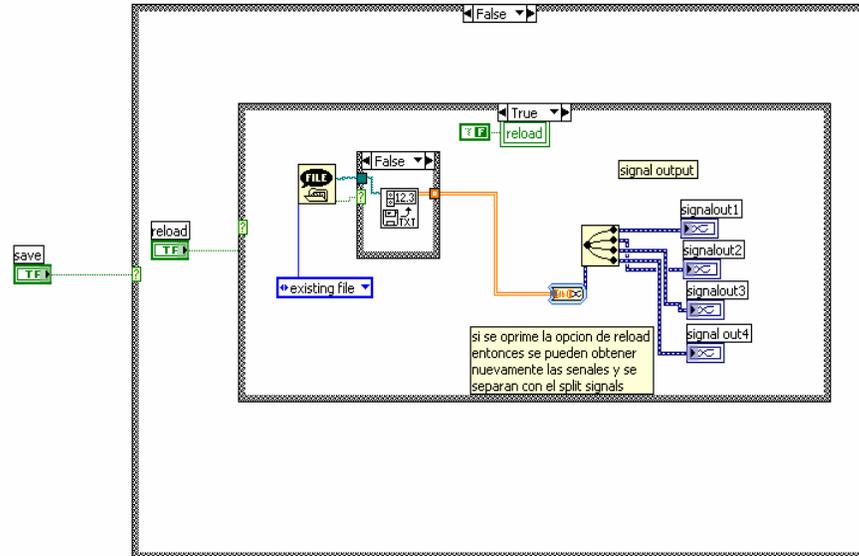


Figura 56. Corrección al dar cancel al leer el archivo

Para reducir el espacio y no agregar tanta carga de elementos a la programación del generador multivoltaje se decidió tomar esta programación e incluirla dentro de un Subvi al cual se nombró como Reloadsubvi.

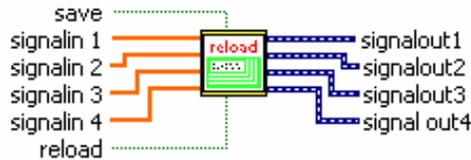


Figura 57. Reloadsubvi

Este SubVI posee entradas para las matrices de cada una de las señales creadas en el la fase de configuración de señal, posee las entradas para los botones de salvar y recargar, así como también las salidas por donde se va a entregar las señales previamente guardadas.

Este subvi debía situarse en la fase de configuración de señal. Al ejecutarlo por primera vez parecía no funcionar. Al probar el programa paso a paso utilizando la opción que presenta labview de Highligh se observó que el subvi si funcionaba, pero que los datos que éste



Figura 58. Controles de Labview para depurar el programa

Arrojaba a la hora de recargar, se borraban tan rápido que parecía que no funcionar. Esto debido a que el programa pasaba por el SubVI, recargaba la señal a la primera iteración pero para las siguientes iteraciones las matrices de valores de las señales se clareaban puesto a que las matrices de salidas se encuentran dentro de un case structure, es decir, para la primera iteración se cargan las matrices cuando el case structure esta en verdadero pues se oprimió el botón de recarga, pero para las siguientes iteraciones el botón no esta oprimido, luego el case structure cambia a falso y allí no hay nada por consiguiente se borran los datos de las matrices de salida.

Para ello se agregó algunas cosas adicionales en el programa principal, quedando finalmente de la siguiente manera:

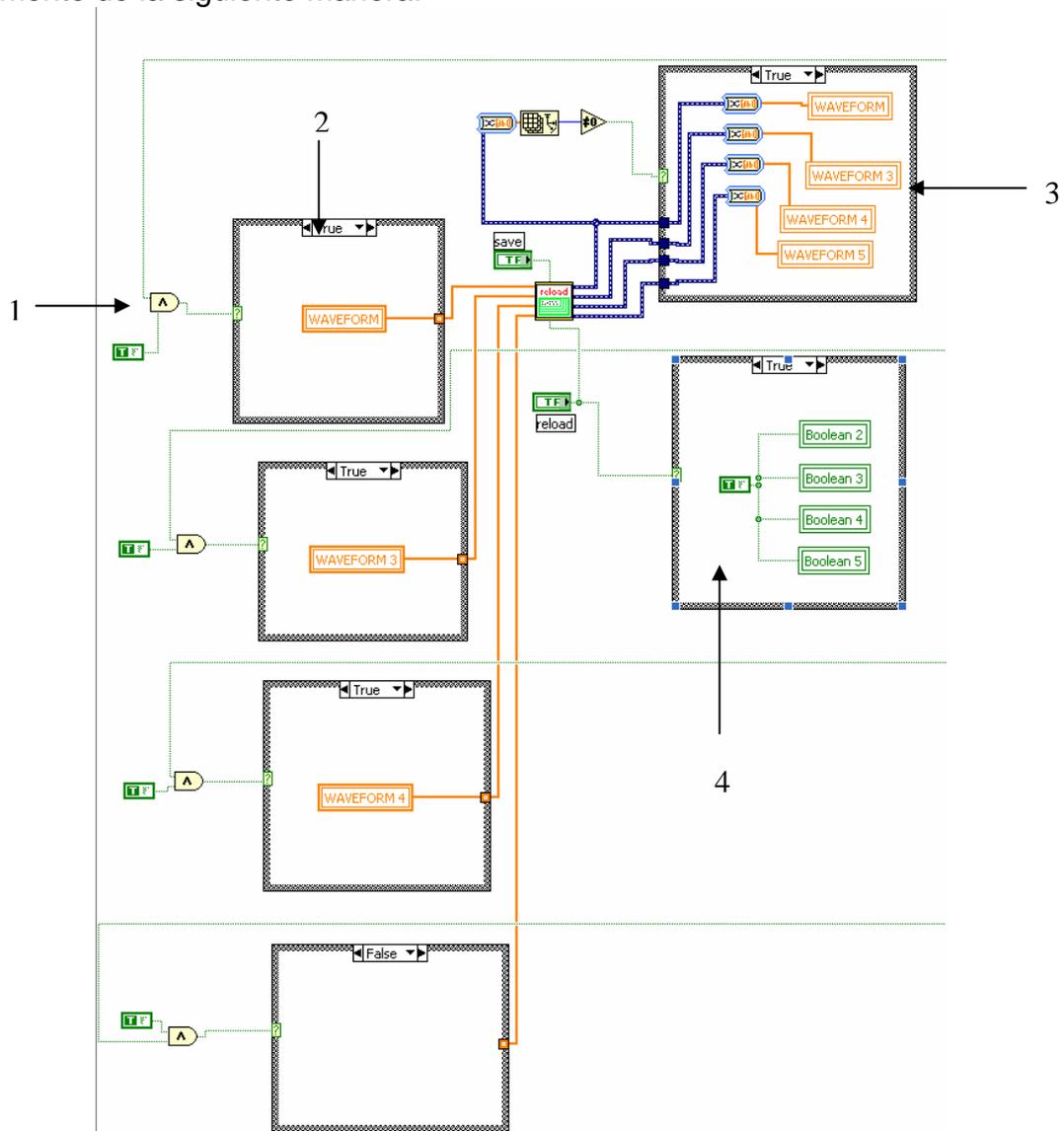


Figura 59. Ajuste del SubVI en el programa principal

- 1- Se asegura que el botón de ON/OFF de la configuración de señal de cada una de las señales este encendido, con el propósito de saber qué tipo de señales se van a utilizar.
- 2- En caso de que se haya configurado una señal que después se quiera utilizar, se toma del Graph los valores de la señal para ser introducidos en el Subvi y allí salvar las señales si se oprime el botón de save, de lo contrario no pasa nada.
- 3- En esta parte es donde se toman los valores de las señales leídas a la hora de realizar la recarga, y serán dibujadas nuevamente sobre los graphs de la fase de configuración de señal pero con la diferencia de que ahora no se van a borrar las graficas.
- 4- Se encienden los botones de On/Off de cada uno de las señales para visualizar las graficas que se quieren recargar.

4.2 REDUCCIÓN DE PROGRAMACIÓN A SUBVIS.

En esta ocasión se redujo partes de la programación a nuevos Subvis, con el fin de visualizar mejor la programación y de optimizar más el programa.

Se produjeron tres nuevos Subvis:

- Spike Signal 2_5Subvi
- Spike all signals
- Config-Signal5Subvi

4.2.1 Spike Signal 2_5Subvi

En este subvi se agrupa la programación que permite realizar el Spike para cada una de las señales de la dos hasta la cinco de la siguiente manera. Antes:

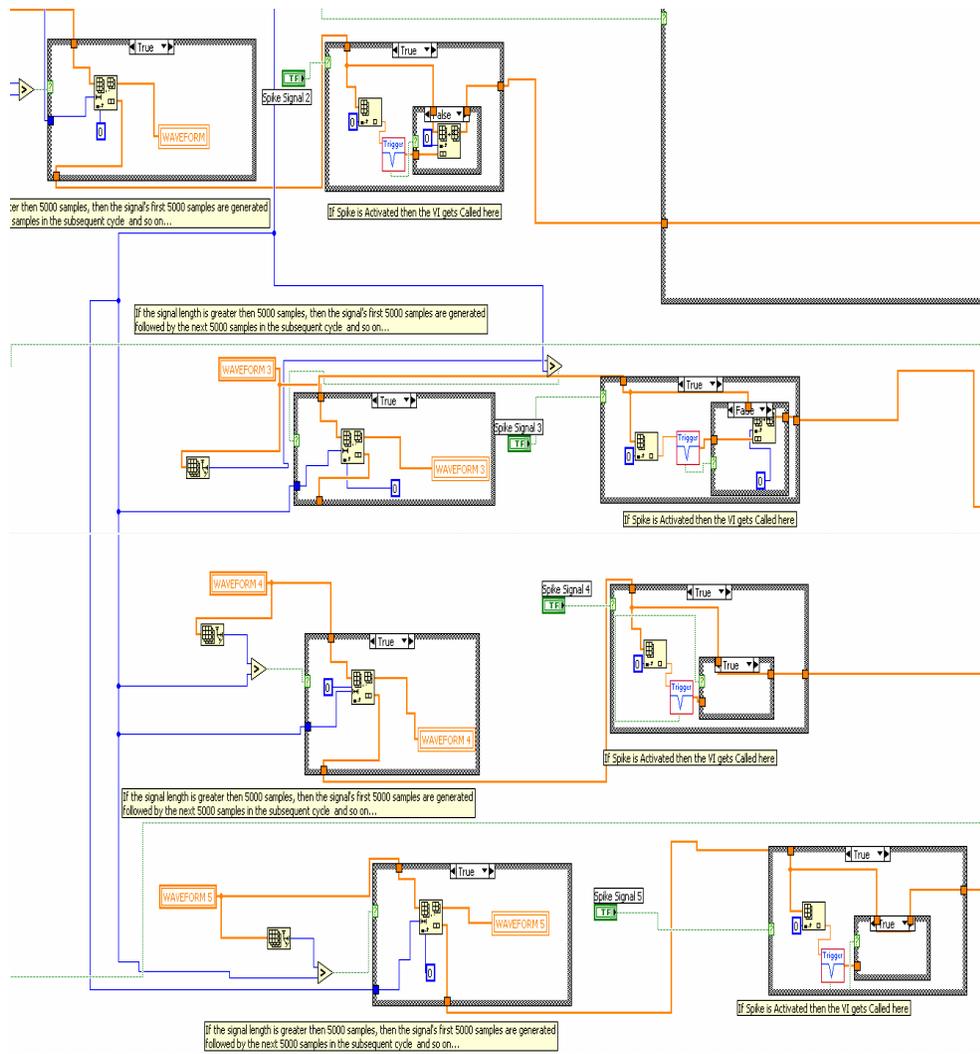


Figura 60. Spike signal para las señales 2 a la 5.

Ahora:

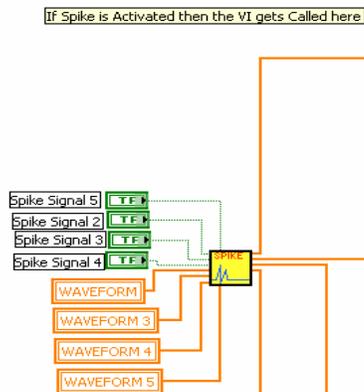


Figura 61. Spike signal subVI

Dentro de este subvi está contenida toda la programación anterior, pero ahora más fácil de interpretar. Se encuentra en la etapa de la generación de voltaje igualmente y se puede observar sus ocho entradas y sus cuatro salidas de forma más sencilla.

4.2.2 Spike All Signals.

Recordemos que esta función se ejecuta solo cuando se oprime el botón de generación de señal y se aplica a aquellas señales que se encuentran encendidas en la etapa de configuración de señal. La función que se realiza aquí es la de generar un pico de voltaje a todas las señales al mismo tiempo.

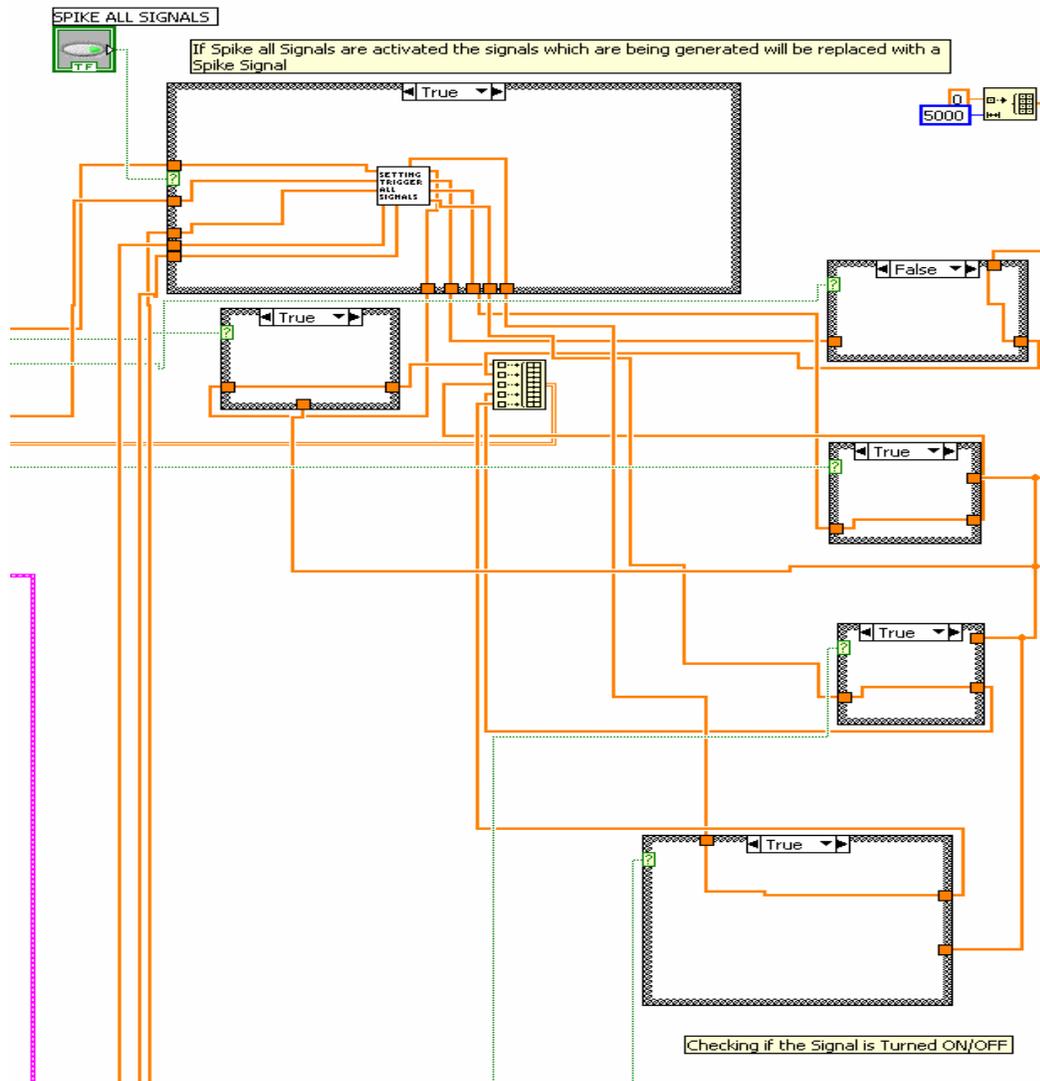


Figura 62. Spike all signals

Ahora:

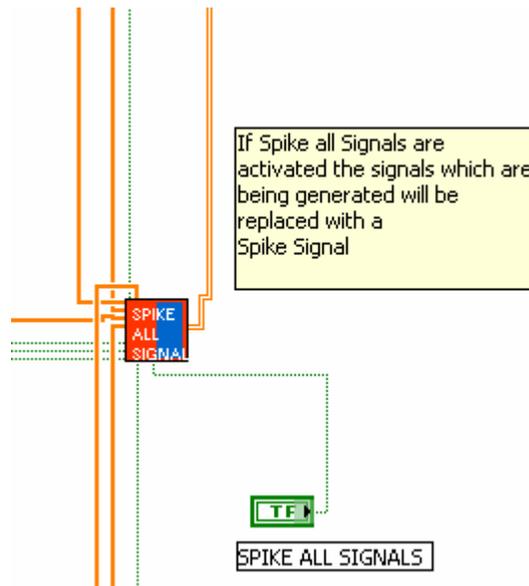


Figura 63. Spike all signals SubVI

Al igual que el anterior la programación quedo contenida en un solo subvi que es guardado en la librería del programa. Este SubVI posee once entradas y una sola salida.

4.2.3 Config-Signal5Subvi.

Fue tal vez la más difícil de los tres SubVIs que se querían cambiar, debido al manejo de todas sus variables internas. Inicialmente se pretendía reducir toda la programación correspondiente a la configuración de las cinco señales en un solo subvi, pero fue demasiado complicado el manejo de las variables, además de que se presentaban problemas con el número de entradas y salidas permitidas para el subvi.

Para evitar inconvenientes con el uso de las variables se decidió realizar un subvi, copiarlo y pegarlo en el programa principal para cada una de las señales. De esta manera las variables que manejaban en su interior para cada configuración de señal no se alterarían y además se ahorraría espacio.

Sin embargo, inicialmente se encontraron problemas en todas las funciones, si se recargaba alguna configuración no se podía visualizar en el los Graphs debido a que las variables internas del SubVI presentaban problemas.

Para superar este inconveniente se agregó dentro de uno de los Case de este SubVI una variable que funcionaba como un puente, es decir para cada uno de

los cambios realizados en la señal, la nueva matriz se almacenaría en esta nueva variable y esta a su vez entrega el resultado final, esto sin lugar a dudas permitió solucionar el problema y corregir las funciones del subvi a la perfección además se podía visualizar las señales en los Waveform Graphs de la etapa de configuración de señal.

El nuevo inconveniente era que al agregar esta variable en esa estructura case como se muestra en la figura 64 se dañaba la función del botón reload, debido a que la variable estaba ubicada justo en el momento en el que al mismo tiempo funcionaria la opción de recarga de señales.

Luego si se presionaba la opción de reload no se tenía nada, debido a esta incongruencia en la programación.

Se encontró que al ubicar la variable puente en la posición en que el SubVI no realizaba nada esta podría incluso ser modificada por la fase de configuración de señal y la función de recarga de señales no sería afectada.

Finalmente se obtuvo la siguiente programación para el SubVI:

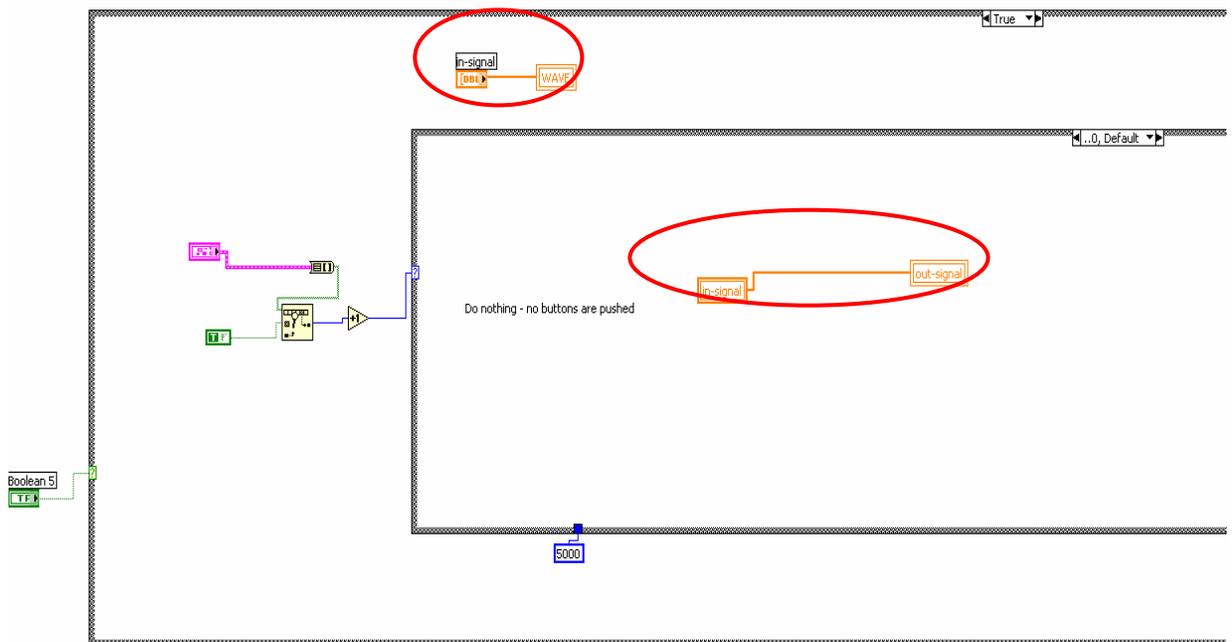


Figura 64. Modificación para crear el SubVI de configuración de señal

Se utilizó la variable Wave como puente y las variables in-signal y out signal para separar las matrices con los cambios realizados al aplicar las funciones de configuración de señal.

En el programa principal anteriormente se tenían cinco veces la misma programación ocupando mucho espacio y haciendo difícil entender lo que se hizo allí.

Para tener una idea, la programación de la figura 64 estaba cinco veces sobre el diagrama de bloques, para lo cual si se quería leer el programa había que hacer uso de la ventana de navegación de labview pues la programación era enorme.

Aunque solo se implemento para las señales 2, 3, 4 y 5, que son las más importantes, el espacio de programación en el diagrama de bloques se redujo casi a la mitad.

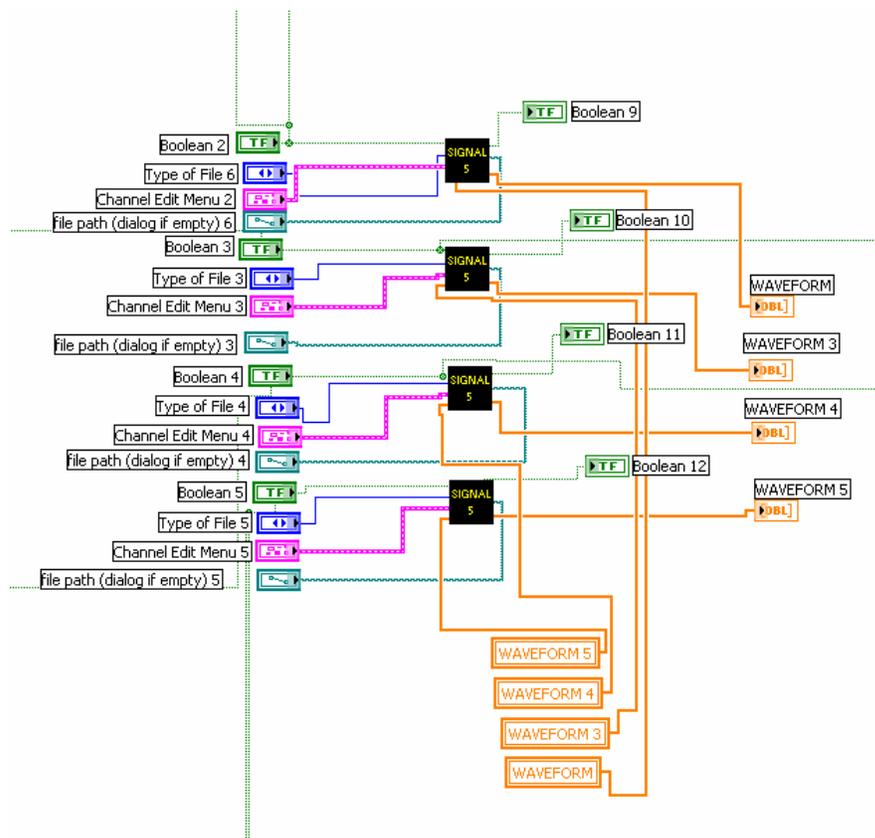


Figura 65. Ajuste de Configuración de señal en el programa principal

4.3 NUEVOS TIPOS DE PARADAS STOP 1 Y STOP 2

Debido a la necesidad de aplicar diferentes tipos de pruebas o realizar el cambio de un chip para aplicar la misma prueba de manera rápida, se agregaron dos botones para realizar dos tipos de paradas.

Los dos tipos de paradas debían realizarse durante la generación de la señal, y debían cumplir lo siguiente:

STOP 1: La señal debe correr, al oprimir Stop 1 debe irse a cero y al oprimir nuevamente el botón de Stop la señal debe continuar generándose.

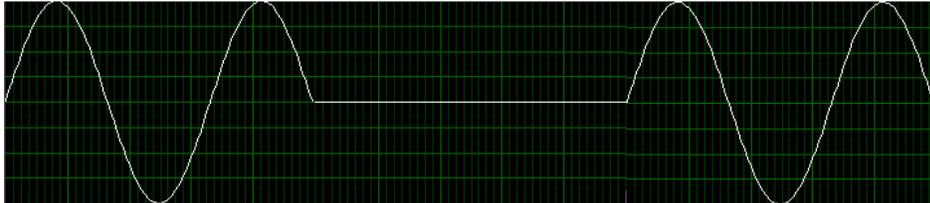


Figura 66. Ejemplo de Stop 1.

STOP 2: La señal debe generarse normalmente, al oprimir el botón de Stop 2, debe detener la generación, pero seguir generándose el voltaje en el nivel o en el punto en que la señal fue detenida y al oprimir nuevamente el botón la señal debe continuar la generación a partir del voltaje donde se detuvo inicialmente.

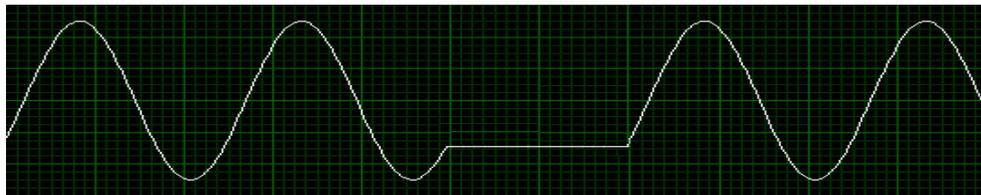


Figura 67. Ejemplo de Stop 2.

Si se retoma el programa completo del generador multivoltaje, se observa que estos pasos básicos están presentes durante la generación.

Ahora para el desarrollo de los botones de Stop, como se mencionó anteriormente la programación debe estar ubicada en la etapa de generación de señal más exactamente antes de reescribir los nuevos valores y debe seguir más o menos la siguiente estrategia de programación.

Inicialmente preguntar por el estado del botón de Stop 1, si este no se oprime preguntar por el estado del Stop 2, de igual manera si este no se oprime entonces no pasa nada y los datos de la matriz de la señal que se están generando pasan sin recibir algún cambio y son generados por la tarjeta.

Si el botón de Stop 1 es presionado, entonces utilizando un Inicializa Array, se crea una matriz de 5000 posiciones llena de ceros hasta soltar nuevamente el botón de Stop 1.

Si el Botón de Stop 2 es presionado, se debe determinar qué tantos valores de la matriz de la señal que se esta generando fueron los que se alcanzaron a generar en el momento de presionar el botón de Stop 2.

Para ello se hace necesario capturar el tiempo en el momento en que se inicia la generación de señal y el tiempo en el que se oprime el botón de Stop 2. Para ello se utiliza el elemento Get date\Time in seconds.

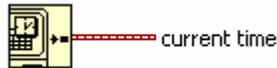


Figura 68. Get date\Time in seconds

Este elemento nos muestra el tiempo actual presente en el PC. Se necesitan dos, uno para el Start Generation Button y el otro para el Stop 2 button. La diferencia entre estos dos tiempos da como resultado un número que es equivalente al número de posiciones que se generaron de la señal original en el momento de aplicar Stop. Como la resta de estos tiempos no es un valor entero hay que redondear al valor más cercano.

Como también es posible que el valor que se obtenga sea mayor de 5000, entonces se crea una especie de bandera que setea nuevamente el Get Date\Time in seconds del Start Generation Button.

El valor resultante de la sustracción de los tiempos se utiliza para correr un For Loop y así capturar los datos que se alcanzan a generar, así como también el valor de la señal en el momento de presionar el botón de Stop 2, esto se realiza con las herramientas de matriz que se puede ver mas adelante en la figura 73.

Lo complicado fue hacer que el programa mostrara de manera consecutiva la transición desde el momento en que se oprime el botón de Stop 2, que se genere el valor constante y finalmente al soltar el botón de Stop 2 que continúe donde se encontraba la generación.

Inicialmente se dejo lo básico, presionar el botón de Stop 2 y generar inmediatamente el valor constante donde la señal fue detenida, pero la generación obtenida no era correcta se presentaban los siguientes errores:

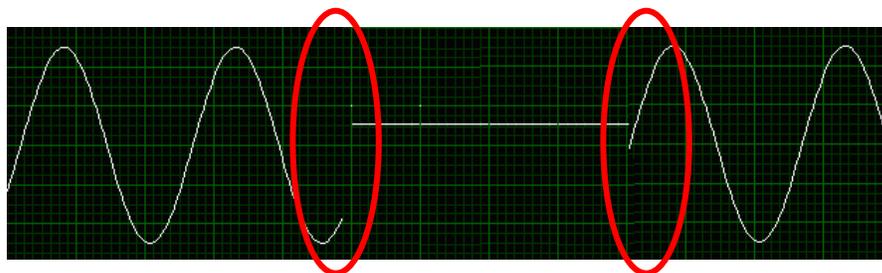


Figura 69. Error 1 en el Stop 2.

Para remediar esto, se separó la matriz que contiene todos los datos de la señal original, en dos partes. La cabeza de la matriz, que contiene los datos de la señal que se alcanzan a generar hasta que se oprime el botón de Stop 2 y la cola de la matriz que contiene los datos de la señal que deben generarse en el momento en que se suelta el botón de Stop 2.

Si a la cabeza de la matriz original se le insertan la matriz de valor constante y el resultado de este se reemplaza en la matriz original, se obtiene algo como lo siguiente:

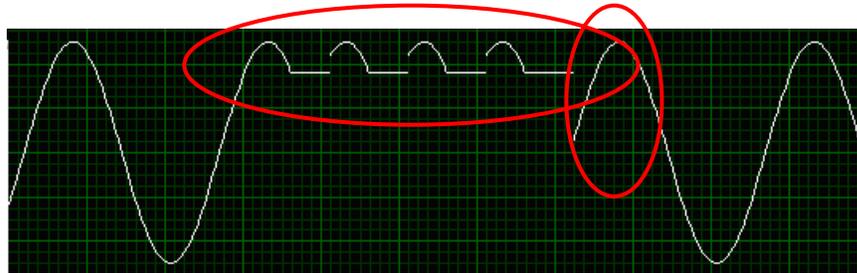


Figura 70. Error 2 en el Stop 2.

De esta manera se pudo corregir parte del error, debido a que se reemplaza consecutivamente la matriz cabeza con la matriz constante. Nuevamente se trato reemplazar la matriz de valores constantes sobre este resultado anteriormente visto, pero no parecía mejorar en gran manera, obteniéndose algo como lo siguiente:

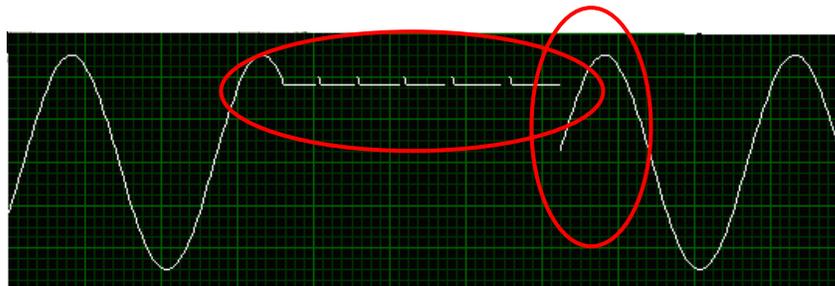


Figura 71. Error 3 en el Stop 2.

Como solución, se trato de mostrar cada situación por aparte, como si fuera una película de dos eventos, el primer evento cuando se presiona Stop 2 se crea la cabeza de la matriz y a esta se le incluye los valores constantes, y el segundo evento solo se generan los valores constantes.

La otra película también de dos eventos para cuando se suelta el botón de Stop 2 consiste en realizar el procedimiento contrario, es decir, se genera la señal con los valores constantes y la cola de la matriz de la señal y el segundo evento cuando solo se genera la señal normal. De esta manera se aplica una estructura

sequence dominada por un control numérico que se hace cambiar por programación.

Finalmente la programación obtenida es la siguiente:

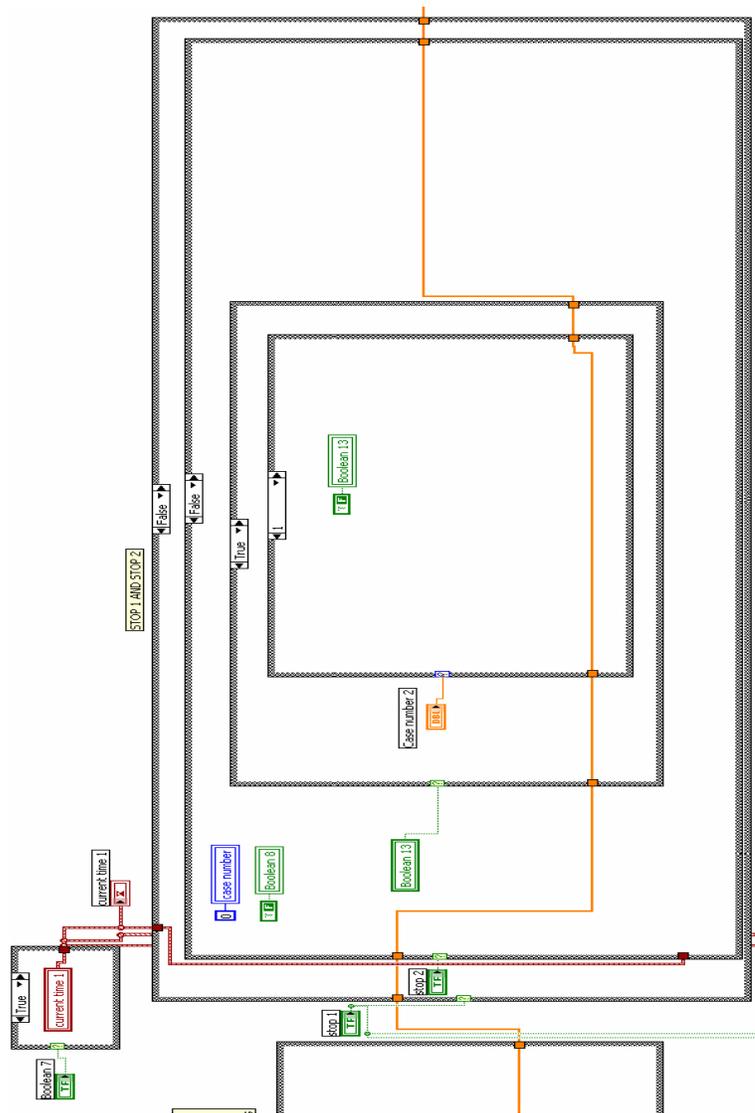


Figura 72. Stop1 y Stop2 sin oprimir

La figura 72 muestra la programación para cuando ninguno de los dos botones se ha oprimido. En la parte de arriba, dentro de un case hay una variable local de color café la cual corresponde al tiempo capturado al oprimir el botón de Start Generation.

La variable local Boolean 13, corresponde a la bandera que se activa cuando se sale del Stop 2, con el fin de diferenciar cuando es que se suelta el botón y cuando es que no se ha presionado ningún botón.

La siguiente figura corresponde a la programación para cuando se oprime el botón de Stop 2, allí nuevamente se captura el tiempo cuando se oprime el botón de Stop 2, se captura del ultimo valor y se ejecuta un caso de dos posiciones para corregir los errores antes mencionados.

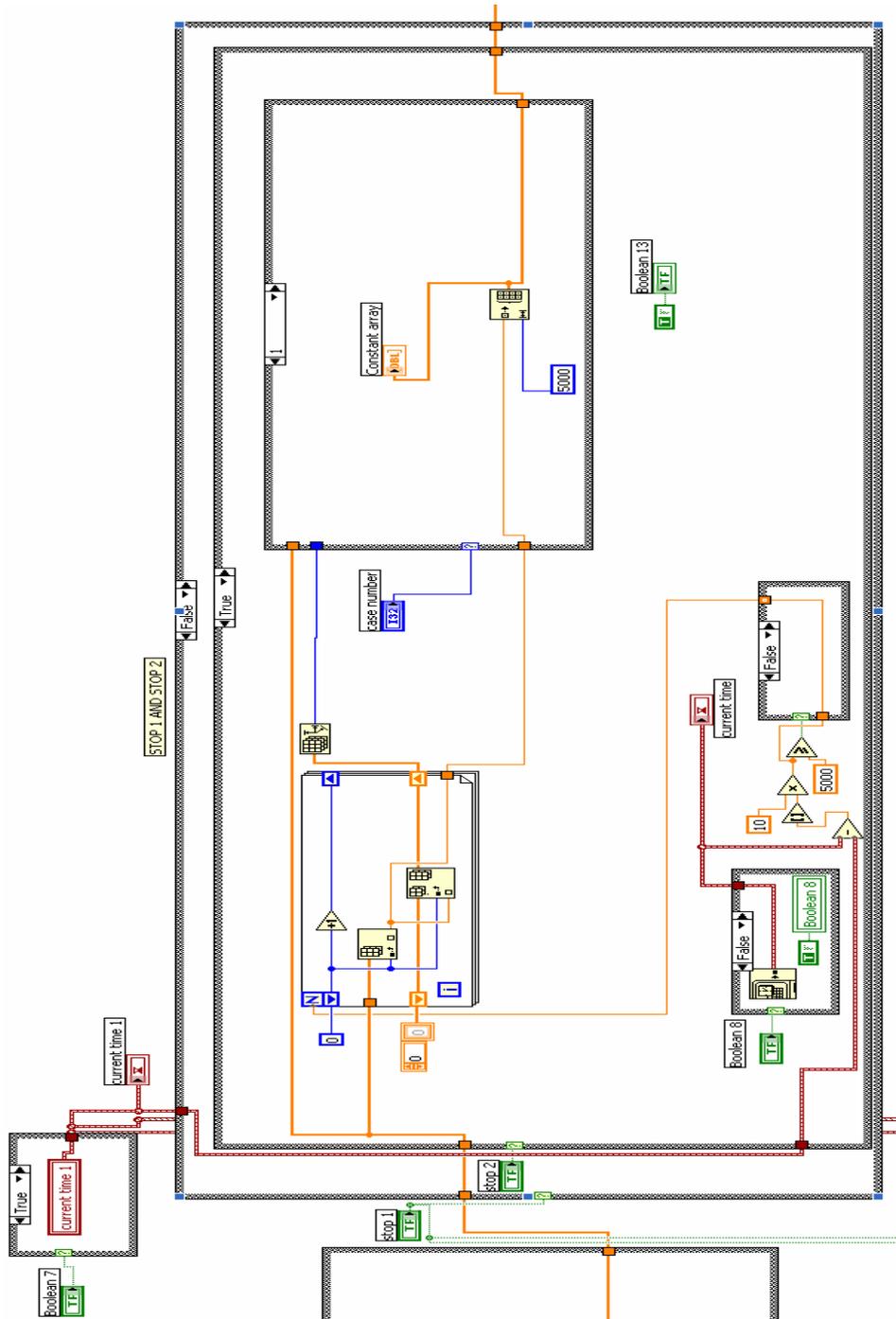


Figura 73. Programación Stop 2.

En la figura 74 se observa la programación para cuando se ha oprimido el botón de Stop 1, aquí se genera una matriz de 5000 posiciones con el valor de cero de la misma manera que en la figura 28.

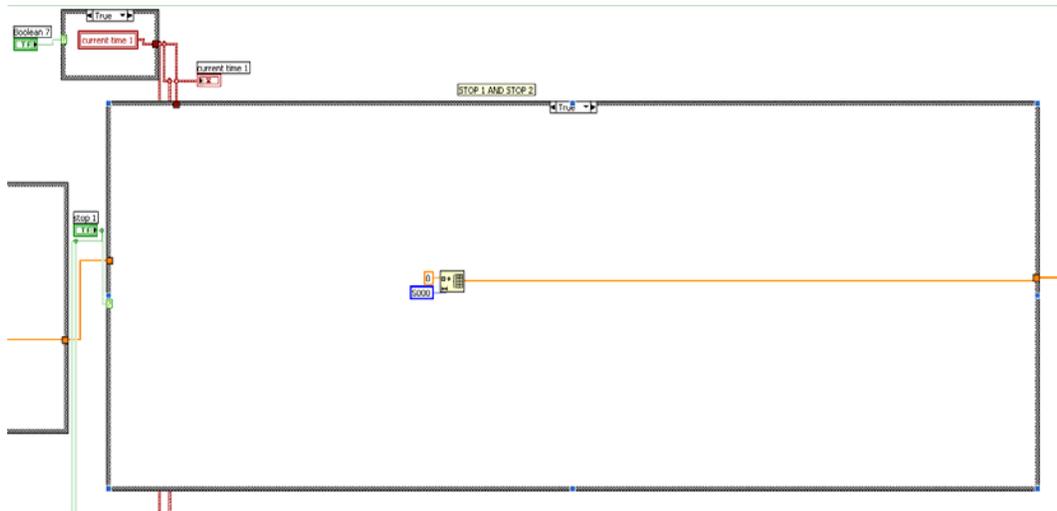


Figura 74. Programación para el Stop 1

Las dos películas se efectúan en dos momentos diferentes una al presionar el botón y la otra al liberarlo o que es lo mismo cuando no esta presionado, para identificar cuando es que es liberado de cuando no se ha presionado se implementa una nueva bandera que se activa cuando se presiona el botón.

4.4 BOTON DE PRESENTACIÓN

Este botón fue creado pensando en los usuarios que van a manejar el Generador Multivoltaje. Al presionar este botón, se abre un nuevo SubVi, el cual contiene una rápida presentación acerca del Generador Multivoltaje y sus funciones.

En el programa principal, esta opción se ubicó en la fase de configuración, de la siguiente manera:

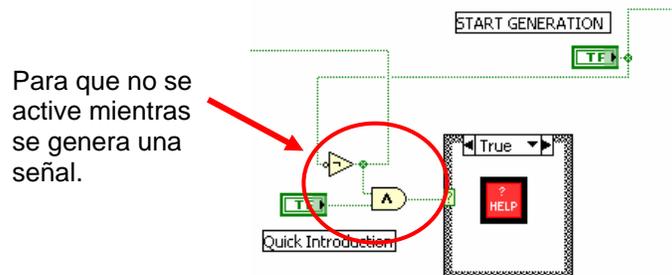


Figura 75. Programación para el botón de presentación

En el SubVi encontramos la programación la cual no es más que una sencilla maquina de estados, que permite al usuario a través de algunos botones desplazarse entre los Tabs de un Tab Control, como se puede ver a continuación.

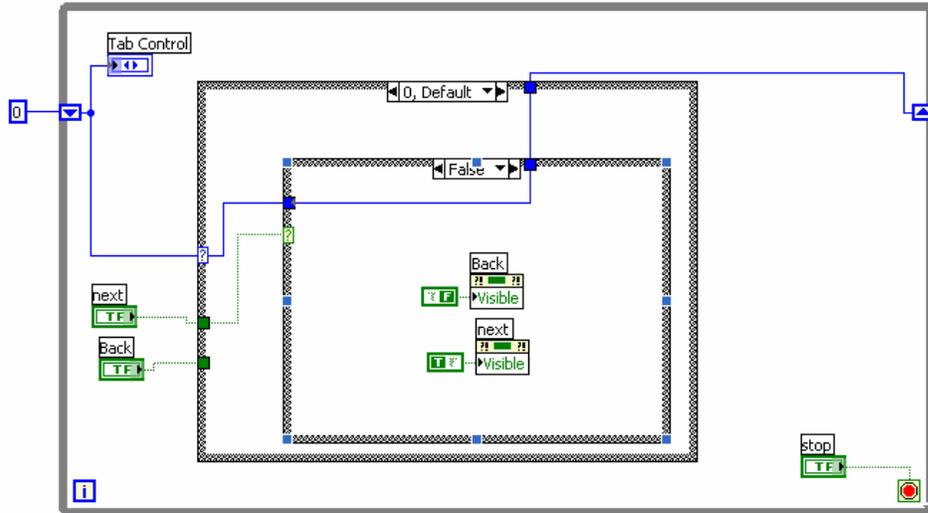


Figura 76. Ejemplo maquina de estados primera diapositiva

En el anterior grafico, se puede observar la estructura básica de una programación de Maquina de Estados, Un While Loop, con un Shift register o registro de desplazamiento, y un case.

El Registro de desplazamiento lo que hace es guardar por un momento o por una iteración un valor hasta ser reemplazado en la siguiente iteración. En este caso podemos observar que el registro de desplazamiento empieza con valor de cero y como nos encontramos en la situación en donde no se ha oprimido ningún botón, este valor de cero no cambia. En el caso en que se oprima el Botón de Next, el valor del registro de desplazamiento se incrementa en 1.

Podemos observar también el uso de Nodos Propiedad con el propósito de ocultar por un momento los botones de Next y de Back, por aquello de que por ejemplo en la primera diapositiva no se necesita el botón de back y en la última no se necesita el botón de Next.

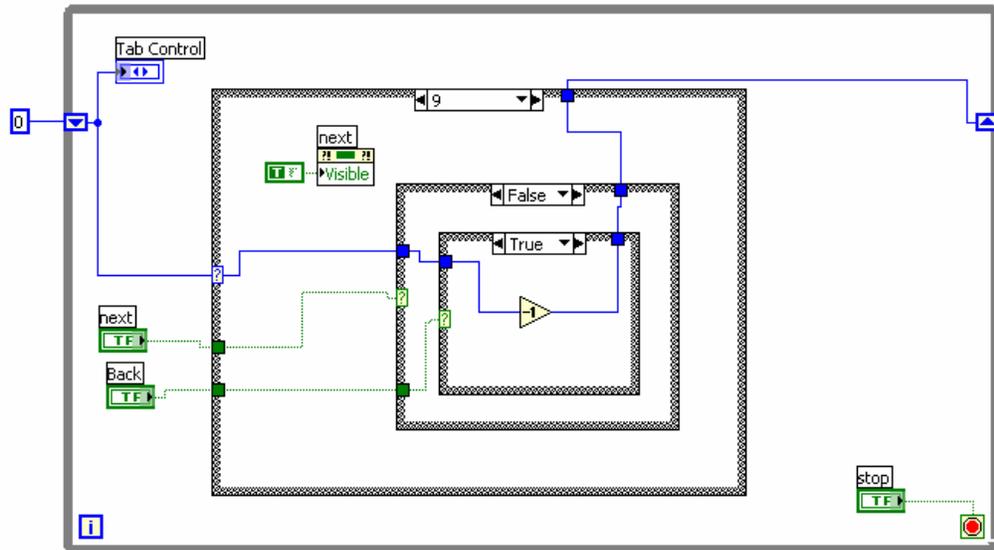


Figura 77. Ejemplo máquina de estados última diapositiva.

En la situación en que se oprima el botón de Back con el propósito de devolverse a la diapositiva anterior, se pregunta por el estado de este botón y se resta uno al conteo que se lleva en el shift register.

Se crearon diapositivas donde se da una rápida introducción acerca del Generador Multivoltaje y sus funciones Principales.



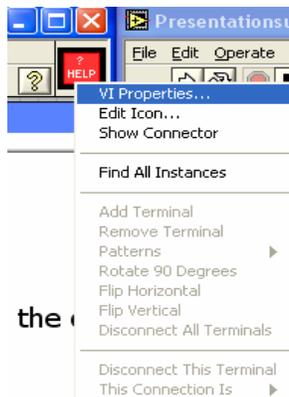
Esta es la Primera diapositiva. Se puede observar que el botón de Back no aparece.

De igual manera en la última diapositiva el botón de Next desaparece y solo se pueden observar los botones de stop y Back.

Figura 78. Ejemplo diapositivas realizadas

Para lograr que Labview permita visualizar el front panel de un SubVi, se deben realizar los siguientes pasos:

1. En el Front Panel de Labview aparece un dibujo pequeño en la parte superior derecha el cual se utiliza o se modifica para diferenciar un SubVi de otro. Allí con el Mouse se da click derecho y se selecciona la opción de Propiedades del VI.



the

Figura 79. Propiedades de un VI

2. Aparecerá una ventana, la cual posee una pestaña que al hacer click despliega un menú con varias opciones. Se selecciona la Opción de Window Appearance o aparición de ventana.

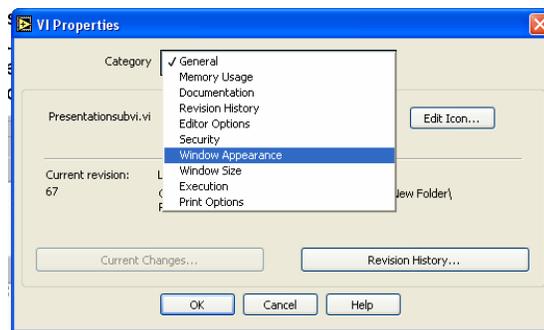


Figura 80. Cambio de parámetros a un VI

3. Una vez hecho esto, se procede a dar click en el botón de Customize. Donde aparece una pequeña ventana como la que se observa a continuación:



Figura 81. Ventana de configuración de apariencia de un VI

Aquí seleccionamos la opción de Show front panel when called (Mostrar el panel de control cuando se llame el SubVi) y Close afterwards if originally closed (Cerrar completamente si es detenido) y después dar click en aceptar.

En la ventana de VI Properties se puede también cambiar otro tipo de atributos como el tamaño de la ventana e incluso podemos tener información acerca del peso del programa, entre otras.

4.5 BOTÓN DE REPORTE RAPIDO

Con el objeto de tomar nota acerca de un caso inusual, se creó un SubVi con el nombre de report generation. Este SubVi es llamado desde el programa principal y despliega la siguiente ventana:



Figura 82. Funciones del Reporte rápido.

1. Muestra la grafica obtenida durante la Adquisición de la señal.
2. Botón para generar el reporte.
3. Waveform Chart, donde se pueden visualizar todas las graficas de las señales a generar.
4. Botón de Hecho. Para salir del SubVi.
5. String Text. Para realizar comentarios o escribir observaciones.

En el programa principal del Generador Multivoltaje, se realizó la siguiente programación para incluir el SubVi de generar el reporte:

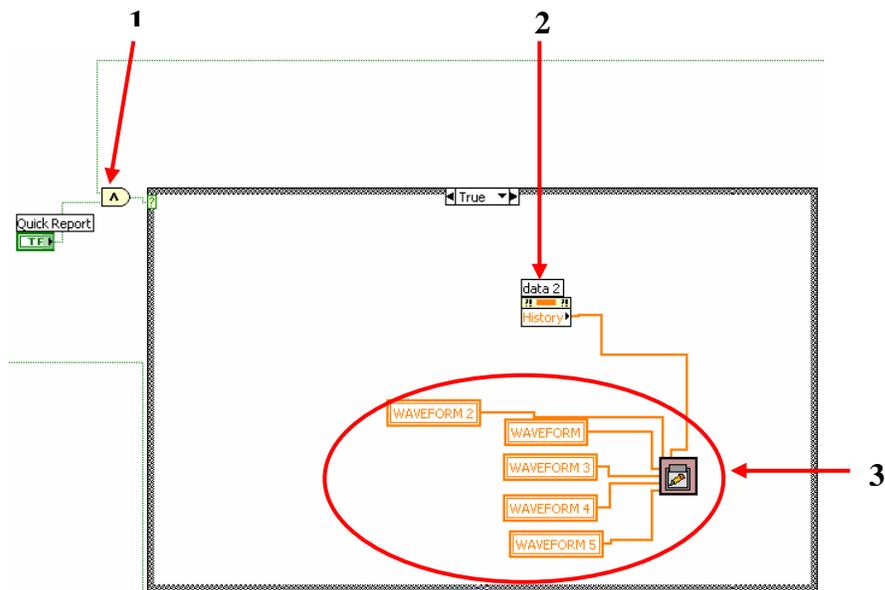


Figura 83. Configuración en el Programa Principal

- 1- Esta compuerta permite que el programa de generar el reporte no se ejecute mientras se esta realizando la generación. Por supuesto la línea verde que vemos arriba que sale de la compuerta y se desaparece a la derecha viene de una compuerta Not que niega el estado del botón de start generation.
- 2- Corresponde a un nodo Propiedad que contiene el Historial de los valores adquiridos.

¿Por que un nodo Propiedad y no una matriz simplemente? La respuesta va de la mano a la siguiente explicación:

Existen muchas maneras de visualizar una grafica, utilizando el Waveform Chart o el Waveform Graph son dos de las opciones más comunes que se encuentran en Labview. Al momento de adquirir una señal no es recomendable usar un Waveform Graph, ya que éstos últimos dibujan

totalmente la gráfica cuando llegan nuevos datos. Sin embargo los Waveform Chart anexan el nuevo dato a los existentes, es decir, tienen una especie de memoria, por decirlo de alguna manera.²

Para acceder a esta memoria del Waveform Chart se utiliza el nodo propiedad configurado en la propiedad de History. Se puede configurar también el número de datos que se desean visualizar. Para ello basta con hacer click derecho sobre el waveform chart y seleccionar la opción change history length.

- 3- Aquí se puede observar las variables locales que contienen las matrices de cada una de las señales que se desean generar para después llevarlas al Subvi de Generación de reporte.

Dentro del Subvi encontramos la siguiente programación:

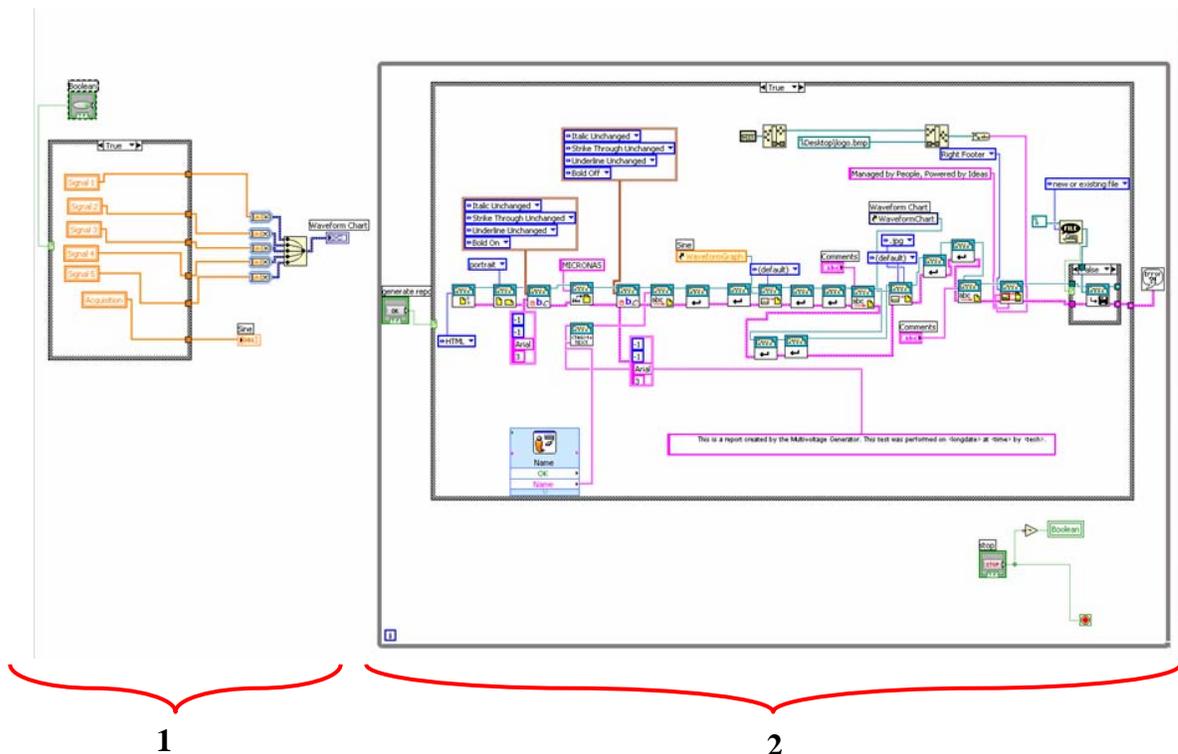


Figura 84. Programación del SubVI de Reporte Rápido

- 1- En esta sección, se capturan las matrices de valores de las graficas del Generador Multivoltaje. Para no perder los valores se conmuta un case haciendo uso de un control boleano que se cambia de estado por programación, posteriormente se llevan las graficas a un Merge Signals y

²Tomado de G programming reference manual pagina 581

finalmente al Waveform chart que permite visualizar las cinco señales por separado en un solo elemento. De forma independiente se muestra la Grafica de la señal adquirida en un Waveform Graph.

- 2- En esta fase se encuentra la programación para la generación del reporte. Solo se activa hasta que el Botón de generate report se presiona. Inicialmente para generar el reporte, se configura la hoja, se anexa un titulo, se configura el texto, se le agrega una leyenda, se agregan un par de renglones y el grafico de la señal adquirida.

Después se agregan otros dos renglones de separación y el comentario después la grafica de las señales generadas, posteriormente se coloca un pie de página y finalmente se salva el reporte.

Algunas de las Herramientas Utilizadas para este programa son:

- New Report. Crea un Reporte nuevo.

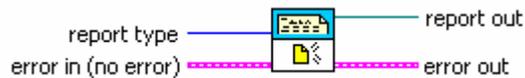


Figura 85. New Report

- Set Report font. Ajusta las propiedades de la fuente del reporte incluyendo a los textos del encabezado y pie de página.

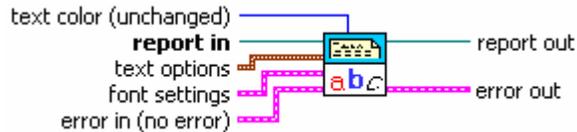


Figura 86. Set Report Font

- Set Report orientación. Determina la Orientación de la hoja, Horizontal o vertical.

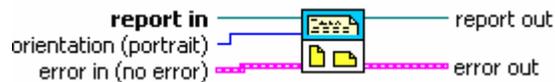


Figura 87. Set Report Orientation

- Append Report Text. Agrega Texto al reporte.

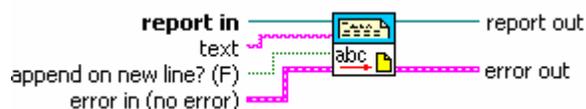


Figura 88. Append Report Text

- New Report Line. Comienza una nueva línea o Nuevo renglón en el reporte.



Figura 89. New Report Line

- Append Control Image to Report. Crea una imagen de los objetos de panel frontal y los adjunta al reporte.

Para identificar que objeto que se desea se dibuje en el reporte, damos click derecho sobre el objeto, seleccionamos create y allí la opción de Reference.

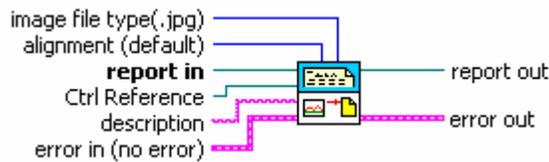


Figura 90. Append Control Image to Report

- Image in header or Footer. Agrega una imagen al encabezado o al pie de página dependiendo de lo que se desee.

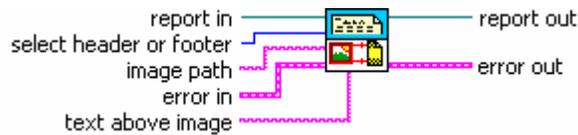


Figura 91. Image in Header or Footer

- Save Report to file. Salva el reporte especificando la ruta. Se usa solo para reportes en HTML no funciona para reportes tipo estándar.



Figura 92. Save Report to File

- Token Converter. Es un SubVi que introduce una leyenda a la cual se le puede cambiar algunas palabras. La leyenda escrita fue la siguiente:

This is a report created by the Multivoltage Generator. This test was performed on <longdate> at <time> by <tech>.

En <longdate> se agrega la fecha actual, <time> la hora que fue creado y en <tech> la persona que lo creo, este ultimo se completa pidiendo el nombre con el Prompt User for Input.

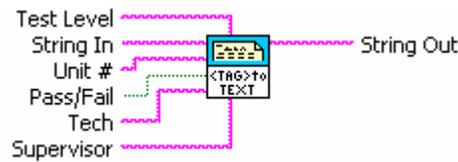


Figura 93. Token Converter.

- Prompt User for Input. Es un VI Express que muestra un cuadro de diálogo para entrar información del usuario como el nombre de usuario o una contraseña.

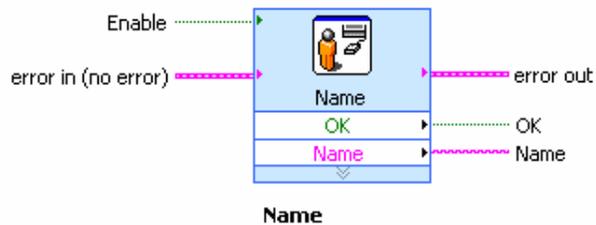


Figura 94. Prompt User for Input

4.5.1 Funcionamiento del Programa

1. En el panel de control del Generador Multivoltaje se hace click en el Botón de Quick report.
2. Una nueva Ventana se Abrirá (ver figura 82). Allí se observan las graficas que se adjuntaran al reporte. También se pueden agregar comentarios. Cuando se considera que el Reporte ya esta listo, basta con presionar el botón de Generar Reporte de color amarillo.
3. Inmediatamente el programa preguntara la ubicación de donde se desea guardar el reporte y el nombre de este.
4. Una nueva ventana aparecerá pidiendo el nombre Completo del Usuario.



Figura 95. Introducción del nombre de usuario para el reporte.

5. Dar click en Done. Posteriormente se puede buscar la ubicación donde guardó el reporte y abrirlo.

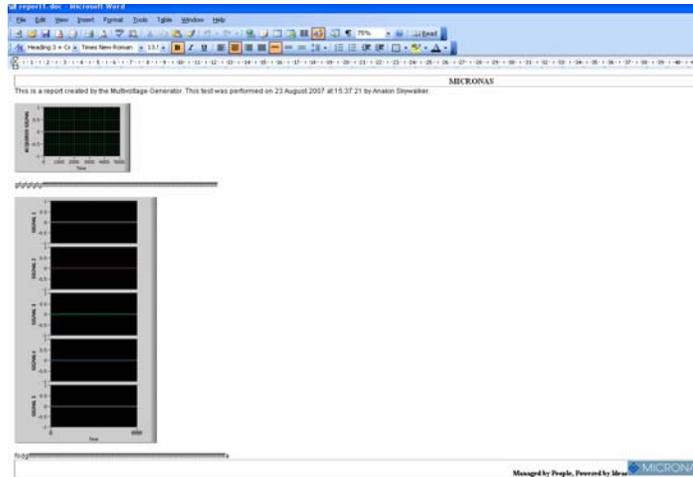


Figura 96. Ejemplo de un reporte.

4.6 GENERACIÓN DE SECUENCIAS DE INICIO (START UP CREATOR)

Debido a la necesidad de estudiar el comportamiento de un circuito integrado ante diferentes secuencias de inicio se creo un subVi que permite modificar una señal previamente guardada en formato LVM. Nuevas características fueron agregadas a la programación como por ejemplo copiar el archivo existente a otra ruta o carpeta que el usuario desee, exportar el archivo a Excel para modificarlo desde allí, así como también leer el archivo de Excel en labview.

Otras funciones que se agregaron fueron que el usuario pudiera cambiar cada uno de los valores de la señal desde labview, agregar zoom a la señal visualizada, salvar la señal a un solo archivo y a cinco archivos que posean un desplazamiento entre ellos, agregar una rampa en cualquier punto de la señal y permitir cambiar el ultimo valor de voltaje.

Como se mencionó anteriormente todas estas funciones se incorporaron en un subVI al cual se llamo Start_up_creator1.vi, y se ubico en el programa principal del generador Multivoltaje en la fase de configuración de señal de la siguiente manera.

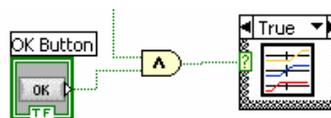


Figura 97. Start up Creator en el programa principal

Como se observa el subVI se encuentra dentro de un case Structure al cual le precede una función AND, esta es necesaria para que el subVI solo corra en la fase de configuración de señal y no en la fase de generación. El cable verde que no tiene ningún control, es el valor del estado del botón de Start Generation negado y el botón que si se puede observar es el botón que permite que el Subvi corra. Al oprimir este botón el usuario se encontrara con el siguiente panel frontal:

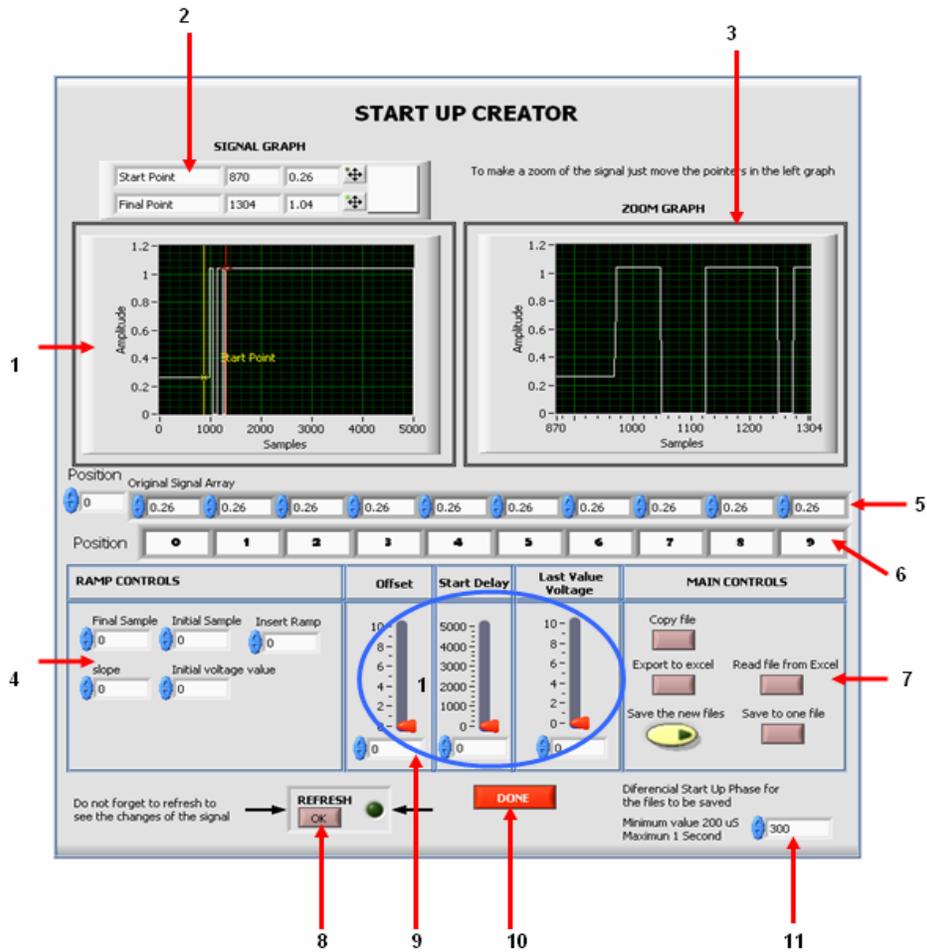


Figura 98. Start up Creator Panel Frontal.

- 1- Signal Graph. Aquí se puede observar la señal que se carga y los cambios que se realicen. Este waveform tiene la característica que esta configurado de tal manera que se permite ver dos cursores de diferente color.

Para crear los cursores basta simplemente en hacer click derecho sobre el waveform graph y dar click en propiedades, allí en la sección de cursores se da click en agregar, y se le colocan los atributos como la forma del puntero, el nombre, el color, entre otros.

El primer cursor de color amarillo es llamado Start Point y el segundo cursor de color rojo Final Point, estos cursores son creados con el propósito de tomar una sección de la grafica y realizar un zoom para visualizar con mejor detalle la forma de onda.

- 2- Corresponde al cursor legend. Se puede visualizar haciendo click sobre el waveform graph y en la sección de mostrar seleccionar cursor legend.

Aquí se muestran los nombres de los cursores y las posiciones en las cuales se encuentran ubicados. Desde aquí también se puede cambiar la posición con solo escribir un punto en donde se quiera que el cursor se sitúe.

- 3- Zoom Graph. Como su nombre lo indica es aquí donde se puede visualizar el zoom de la señal en la sección donde el usuario sitúo los cursores. También se puede observar el rango en donde los cursores se encuentran situados.
- 4- Ramp Controls. Aquí se configura la rampa que se desee agregar, es importante definir el punto inicial, el punto final, el lugar donde se pretende insertar la rampa, la pendiente y el voltaje inicial en donde se desea que empiece la rampa.
- 5- Corresponde al Array de valores de la señal que se cargo. Se visualizan cada diez posiciones. Y se pueden cambiar con el control que se encuentra a la izquierda del array.
- 6- Aquí se visualiza de manera automática el valor de la posición de los valores del array que contiene los valores de la señal. Por ejemplo para la grafica que se observa en los waveform graphs de la figura 98, se ven los diez primeros valores de la señal y el array nos muestra sus posiciones en el array de cero a nueve y puede cambiar la posición también usando el control del array que contiene los valores de la señal.
- 7- Main Controls. Los controles principales del SubVI corresponden a aquellas funciones que permiten salvar, exportar a Excel, leer desde Excel, salvar a un archivo y salvar a múltiples archivos.
- 8- Refresh Button. Este botón fue creado para refrescar los datos del array de la matriz que se cargo y que se visualiza en labview con el propósito de agregar más modificaciones.
- 9- Aquí se encuentran los controles para modificar el Start delay, el último valor y el Offset de la señal.
- 10- Done Button. Este botón termina la ejecución del subVi.

11- Aquí se escribe el valor que se quiere que las cinco señales que se van a guardar se queden desplazadas en el tiempo teniendo en cuenta la relación de que un segundo corresponde a 5000 muestras.

En cuanto a la programación se refiere debido a que es algo extensa, es necesario mostrar por aparte cada una de las funciones realizadas.

De manera general, la programación esta basada por decirlo así en una pequeña maquina de estados. Dentro de un While Loop grande y con un Shift register se incrementa el valor de un Case Structure, en donde en su primera posición se pregunta por el botón de copy file de la siguiente manera.

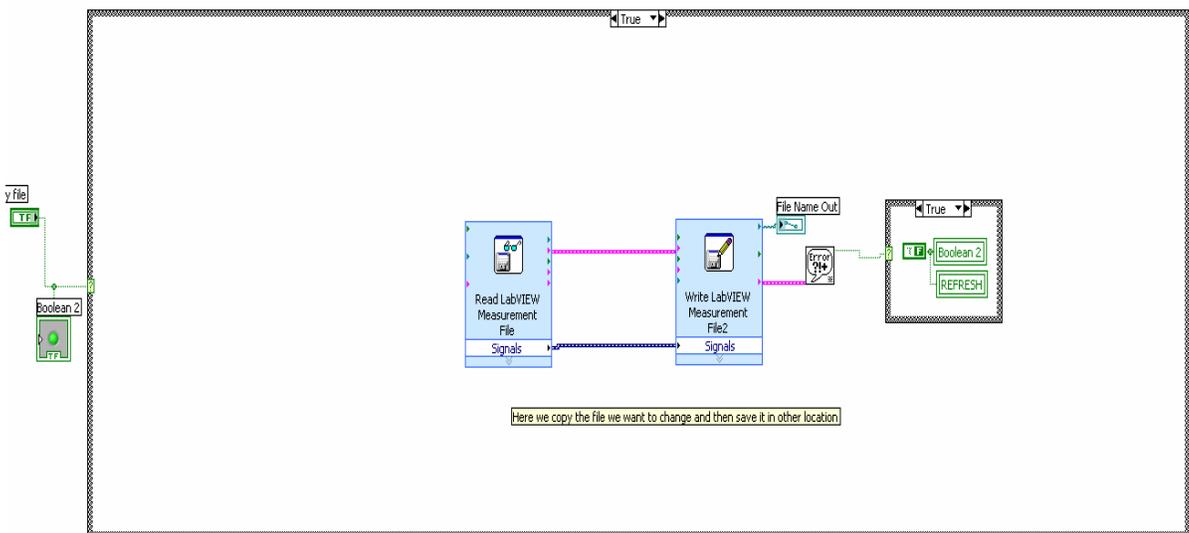


Figura 99. Función Copy File.

Haciendo uso de Read Measurement file se lee el archivo LVM, y junto con el Write Labview Measurement file, se escribe la nueva posición donde se guardará una copia del archivo. También hay una pequeña estructura de error en caso de la lectura o la escritura falle, para lo cual se activa un case structure que establece qué funciones no se deben ejecutar cuando hay error.

En el segundo estado del case estructura, es decir case 1, se lee el archivo por primera vez haciendo uso de un read Labview Measurement file y el archivo es puesto en un array llamado *original Signal Array*.

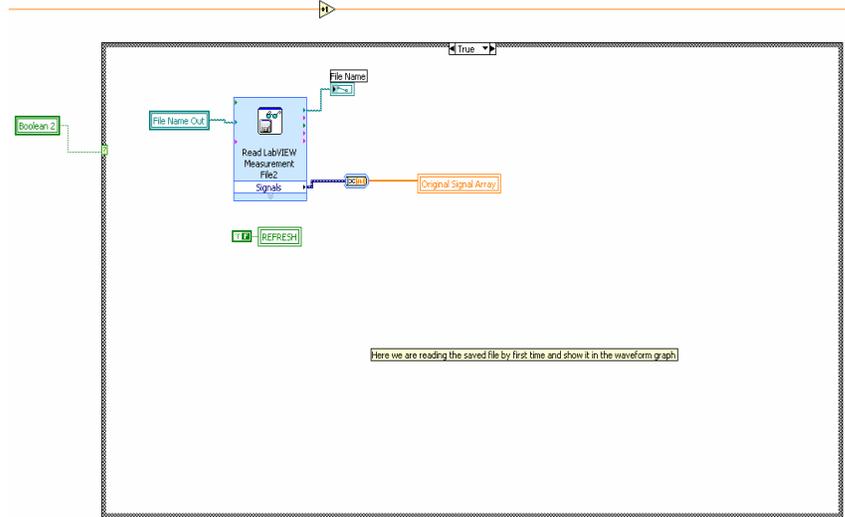


Figura 100. Lectura de archivo por primera vez

Pero debido a la necesidad de refrescar los valores del array de la señal original y debido a que el vi Express de Read Labview Measurement File solo permite leer una vez, pues a las siguientes veces muestra error, fue necesario implementar una nueva forma de leer los archivos LVM a partir de un Read from spreadsheet array, haciendo uso de unas cuantas modificaciones como se muestra en la siguiente grafica.

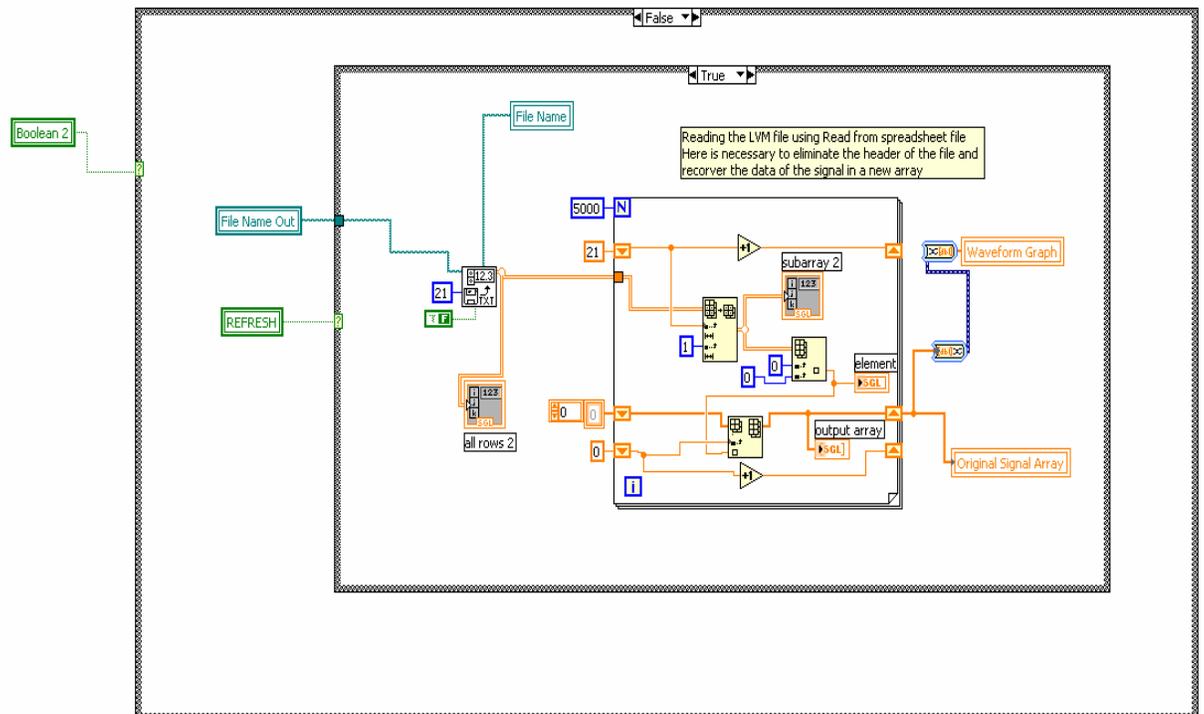


Figura 101. Lectura de archivo después del Refresh

Como se observa esta opción no se ejecuta si el botón de Refresh ha sido oprimido, en este caso, se lee el archivo LVM con el Spread Sheet File, configurado para que no realice transpuesta y para que lea el archivo a partir de la línea 21, puesto que a partir de esa línea se encuentran los datos de la señal y antes se encuentra el encabezado del archivo.

Como el Read from Spreadsheet entrega una matriz de dos dimensiones, con dos columnas, donde la primera columna corresponde a la posición de la muestra y la segunda columna a los valores de la señal los cuales son los que se necesitan extraer, se utiliza un For loop.

Dentro de este se encuentran las estructuras array subset, index array e insert into array.

Con el Array subset se toma solo la columna que interesa, la de los datos de la señal, pero como el resultado sigue siendo una matriz de dos dimensiones, utilizando un index array se extraen cada uno de los valores y junto con el insert into array, los datos son guardados en un nuevo array de una sola dimensión.

Finalmente se sacan los valores para ser cargados en el array de la señal original y el waveform graph. Lo mas importante es que se puede oprimir el botón de refresh cuantas veces se desee y no habrá ningún inconveniente como si se usara el Read Labview Measurement File.

En el case 2, es donde se encuentra la mayor parte de la programación, allí se realiza la programación para crear la rampa, haciendo uso de un nodo formula y de la ecuación de la recta de la siguiente manera:

Dentro de un For Loop que se ejecuta un número de veces correspondiente a la diferencia entre el punto superior y el punto inferior del eje de las ordenadas, hay un case structure en cuyo interior se encuentra alojado el nodo formula con la ecuación de la recta " $Y_f = m \cdot (x_f - x_o) + y_o$ " donde Y_f corresponde al valor final, m a la pendiente, x_f el punto inicial de la muestra, x_o el punto final de la muestra.

El nodo formula se ejecuta hasta que la muestra inicial sea mayor que la muestra final en el eje x, estos valores arrojados por el nodo formula son insertados en un nuevo array el cual será posteriormente insertado en un punto del array original de la señal que se carga.

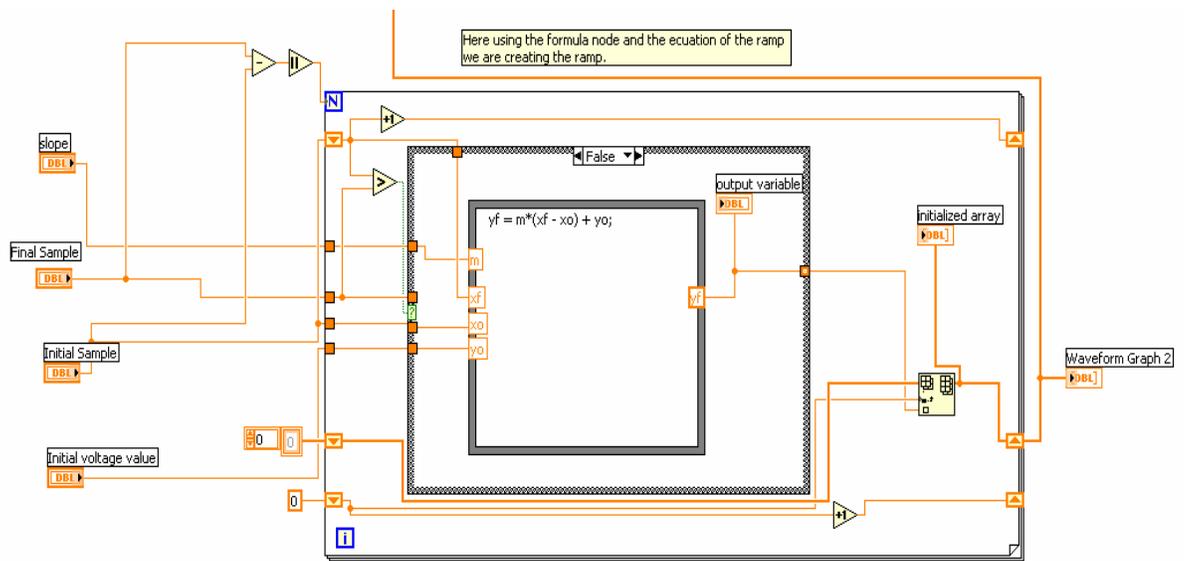


Figura 102. Creación de rampa

Del array de valores de la señal que se carga inicialmente no solo se le pueden cambiar sus valores directamente desde ese array que esta configurado como control sino que también se puede insertar la rampa, agregar un desplazamiento, offset y cambiar el ultimo valor de la siguiente manera.

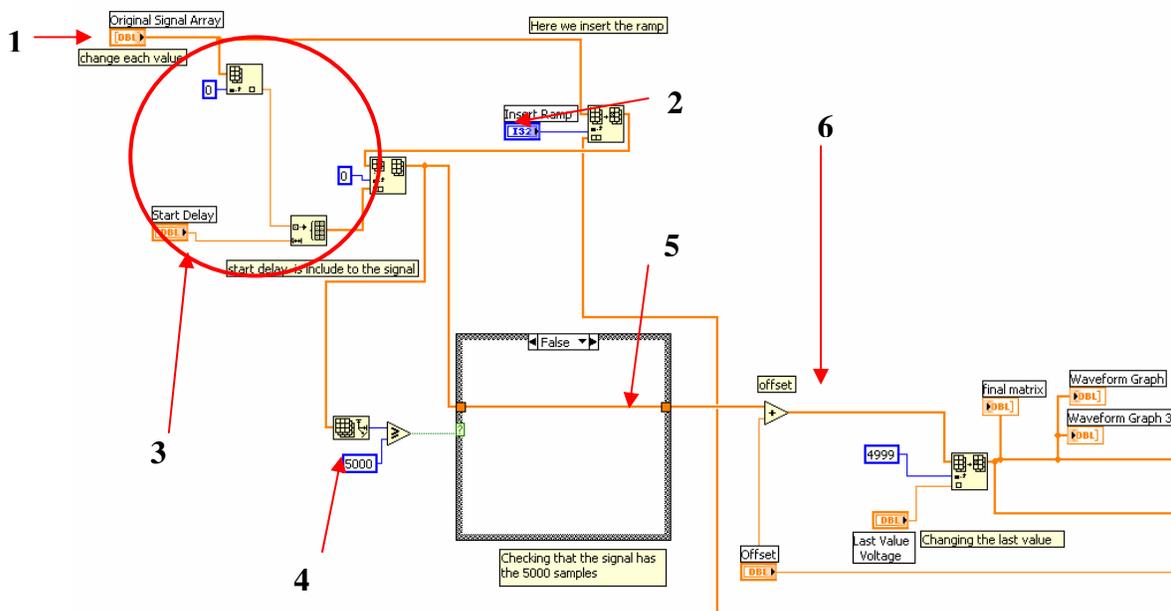


Figura 103. Delay, inserción de rampa y offset

- 1- Corresponde a la matriz de la señal original, la que podemos observar en el Panel de control y esta configurada como control así que es posible cambiar los valores de la señal en cada punto desde allí.

- 2- Corresponde a un insert into array y es allí donde la matriz de valores de la rampa son insertados en un punto de la señal original determinado por el usuario.
- 3- Aquí se inserta el delay, para ello se hace necesario inicializar un array con un número de posiciones igual al delay que se desea generar y posteriormente insertarlo al array original.

Se hace uso de un index into array porque se desea que la señal se desfase pero conserve el mismo valor de voltaje inicial que posee.

- 4- Aquí se pregunta si la matriz tiene un tamaño superior a 5000 muestras, en caso de que sea mayor solo se toman las primeras 5000.
- 5- En este punto se agrega el offset, al cual no es más que una suma entre los valores del array de la señal y un valor constante.
- 6- Finalmente haciendo uso de un Replace en la última posición de la señal se cambia el valor y después el array resultante es mostrado en los waveform graphs del panel de control.

Para mayor comodidad del usuario y poder entender mejor en que posición se encuentra en el array de la señal original, fue necesario insertar un nuevo array que mostrara de manera automática los valores de la posición del array original.

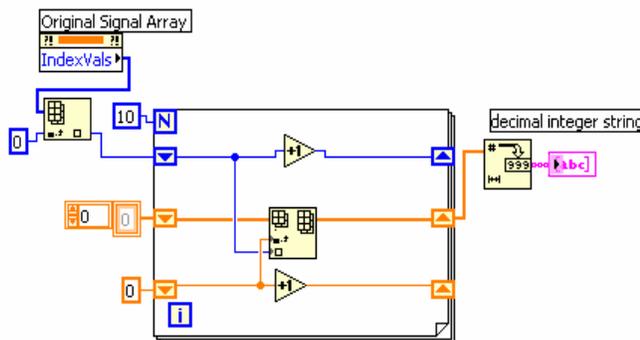


Figura 104. Array de posiciones.

Inicialmente es necesario conocer la posición que se está apuntando en el array para ello se usa un Property node o nodo propiedad. Al valor dado se le suma 1 diez veces para insertar los valores en un array nuevo a partir de la posición que se indica y nueve valores más.

Finalmente haciendo uso del number to decimal string se convierte el array de números a un array de texto de números. Este último paso no es necesario. Se puede colocar un array normal configurado como indicador.

También con el propósito de poder observar bien los cambios que se realicen en la grafica fue necesario implementar un graph aparte para realizar un zoom.

Para tomar la sección de la señal a la cual se le desea hacer zoom se utilizaron dos cursores como se menciono anteriormente.

El primero corresponde al punto inicial, y el segundo al punto final. Otra manera de colocar cursores en un waveform graph es haciendo click derecho sobre el waveform al cual se le quiere colocar cursores, luego seleccionar propiedades y allí dirigirse a la opción de cursores.

En esta opción se pueden modificar los parámetros del cursor como su color, nombre, estilo de línea, grosor de línea, entre otras características como se puede ver en el siguiente grafico.

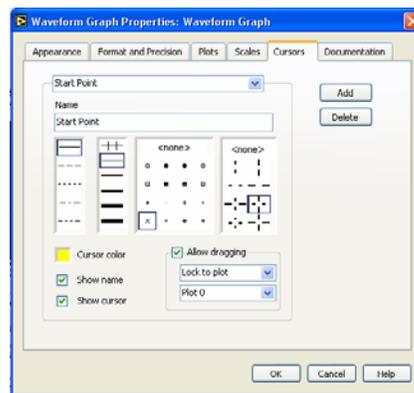


Figura 105. Propiedades de cursor Waveform Graph.

Por otra parte en cuanto a la programación para realizar el zoom se tiene lo siguiente:

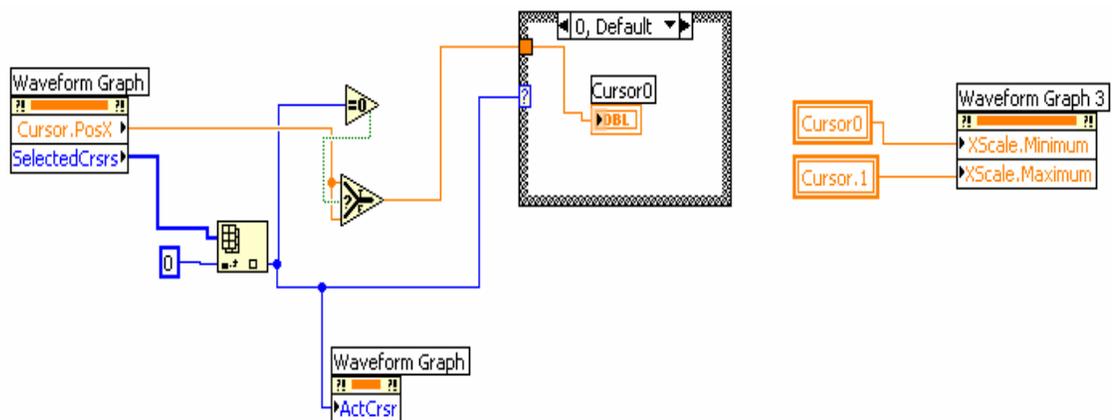


Figura 106. Lectura de posición de dos cursores.

Con los nodos propiedad se puede capturar la posición de uno de los cursores. Para determinar cual cursor se seleccionó es necesario usar la propiedad Selected cursors, esta cambia según el número del cursor, si es el cursor 0, selected cursors se pone en cero, si es el cursor uno, selected cursors se pone en uno, de tal manera que se puede usar para manipular un case structure dentro del cual hay dos indicadores numéricos que capturan el valor de los cursores y haciendo uso de sus variables locales se introducen estos valores dentro de otro nodo propiedad pero del Waveform graph donde se desea realizar el zoom configurado en la propiedad de valor de escala mínimo y máximo del eje X.

Para grabar la señal resultante a cinco archivos desplazados entre si cierta cantidad de muestras, se utiliza la siguiente programación:

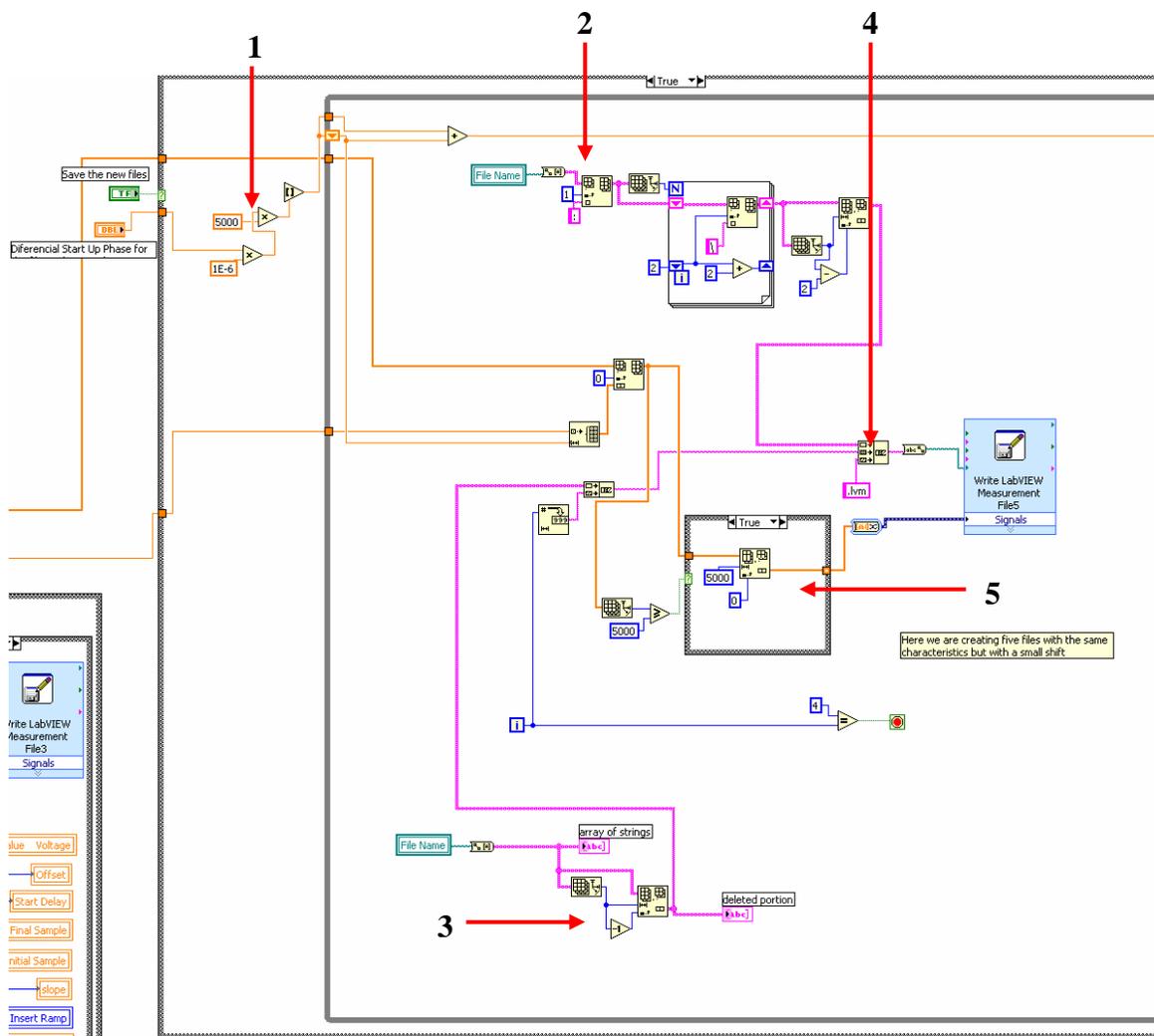


Figura 107. Programación para guardar 5 archivos.

Se puede observar que esta opción no funciona sin antes haberse presionado el botón de save the new files, entonces con el valor numérico introducido en el control numérico:

- 1- se convierte el valor dado a microsegundos multiplicando por 0,000001, luego se multiplica por 5000 para determinar el número de muestras que se deben desplazar, estos valores entran a un while loop.
- 2- Se captura la dirección o el path donde se almacenó el archivo, se convierte a array y allí se elimina el nombre del array.
- 3- Del mismo path se captura solo el nombre del archivo y con un bundle y un number to string se le agrega al nombre de archivo un número en secuencia para diferenciarlos.
- 4- Con este bundle se coloca en un solo string el path con el nombre cambiado y posteriormente con la herramienta string to path se convierte a path para ser introducida en el write to spreadsheet file.
- 5- La señal es desplazada según el valor dado de tiempo y se revisa nuevamente que no sobrepase las 5000 muestras.

Para refrescar el array de la señal original, se hace el procedimiento de salvado de la señal y posteriormente lectura que ya se explicó mas arriba en este documento. En el momento de salvar es importante que los valores de los controles de rampa, offset y start delay se ajusten a cero, como se muestra en la figura:

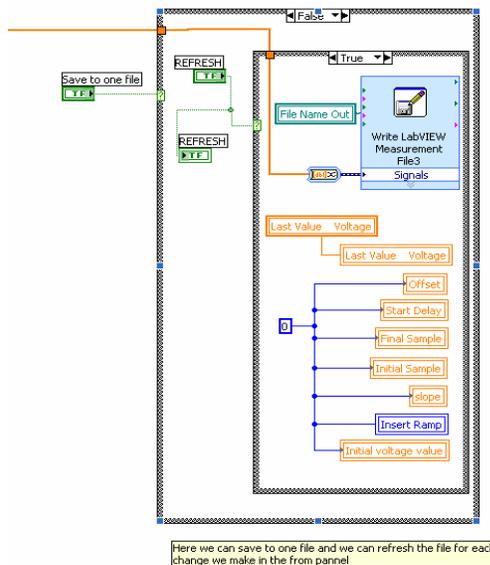


Figura 108. Programación para salvar un archivo y refrescar

Podemos observar que el programa pregunta por el estado del botón de refresh dentro de un case estructura mas grande. Este corresponde al case de la opción de salvar a un solo archivo, para lo cual se hace uso de un write labview measurement file simplemente.

Como una característica adicional se colocó la opción de importar y exportar a Excel. Para exportar a Excel, lo único que hay que hacer es salvar la señal en formato xls, para ello se captura el path donde esta guardado el archivo y allí se le introduce la extensión.

También se genera un array de dos dimensiones, en donde la primera dimensión corresponde a las posiciones de 0 a 4999 y la segunda columna a los cinco mil valores de la señal puestos en orden, tal y como se muestra en la siguiente figura:

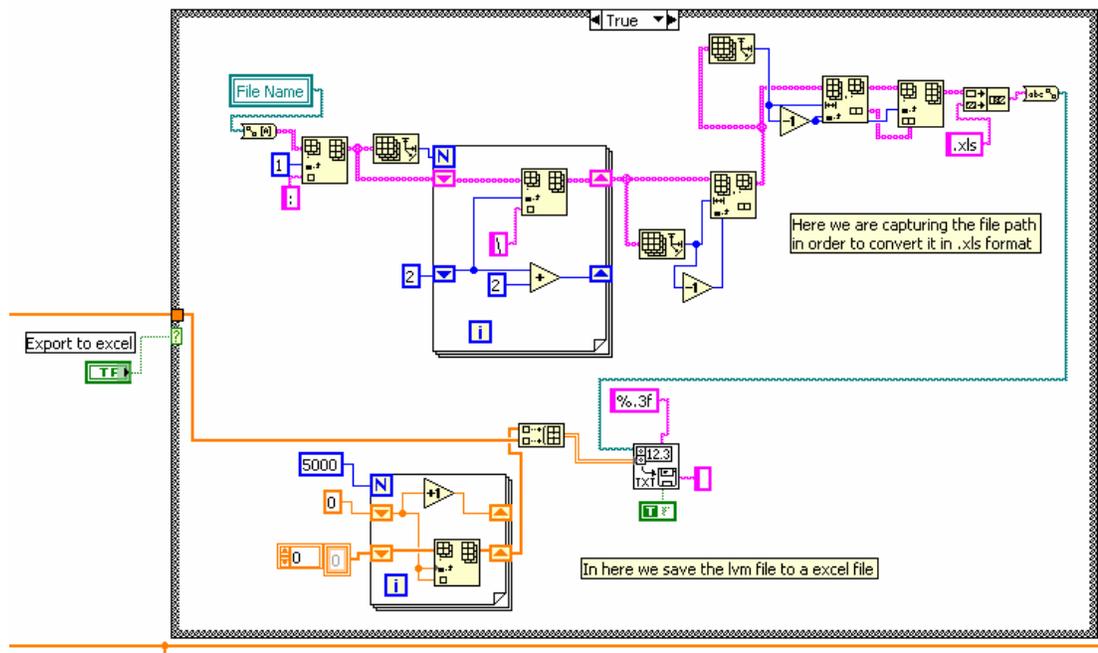


Figura 109. Programación para importar a Excel

Para Importar desde Excel, o para leer el archivo de Excel, se realiza el paso contrario, es decir, se extraen los valores del archivo pero estos vienen en un array de dos dimensiones el cual debe ser transformado en un array de una dimensión como se muestra en la figura numero 110.

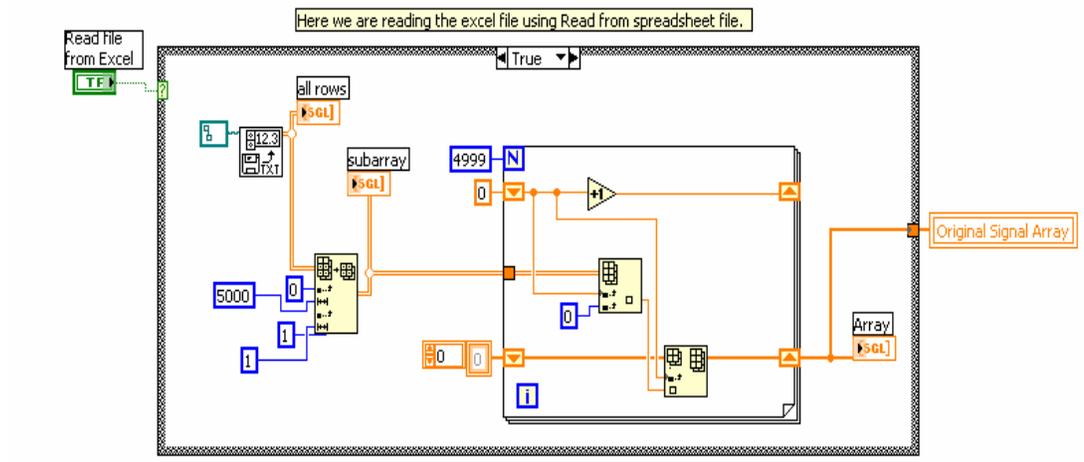


Figura 110. Lectura desde Excel

4.7 CREACION DE RIZOS

Con el propósito de crear de manera más sencilla una señal con rizos se creó un SubVI llamado Create Ripple. El subVI tiene la siguiente apariencia:



Figura 111. Subvi de Creación de Rizos

El SubVI solo arroja dos salidas, la primera no es mas que sino el array de la señal con el rizo y la segunda el valor boleano al dar cancel. La interfaz grafica o un panel de control para el usuario el cual es el siguiente:

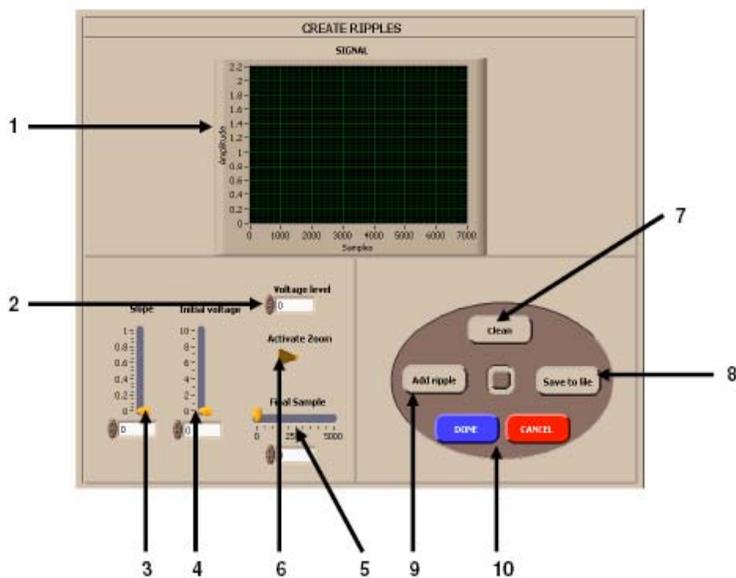


Figura 112. Creación de rizos panel frontal

- 1- Waveform graph donde se visualiza la señal.
- 2- Concierne al nivel de voltaje que se desea tenga el escalón.
- 3- Slope, corresponde al valor de pendiente de la señal.
- 4- Voltaje inicial, corresponde al nivel de voltaje inicial que se desea tenga la pendiente.
- 5- Final sample, es el punto final de la pendiente en el eje de las muestras.
- 6- Activate Zoom. Como su nombre lo indica, permite activar el zoom en la grafica. Al hacer click sobre este botón se despliegan dos waveform graphs, uno donde el usuario puede seleccionar la porción de la señal sobre la cual desea hacer zoom haciendo uso de dos cursores y el segundo waveform graph donde se visualiza el zoom de la porción.
- 7- Clean, borra o limpia los datos.
- 8- Save to File, guarda la señal creada a un archivo de extensión LVM.
- 9- Add Ripple, al oprimir este botón se abre un nuevo front panel donde el usuario puede seleccionar el tipo de rizado que quiera, así como la frecuencia de este y el voltaje.
- 10-Done. Para salir.

En cuanto a la programación se refiere se tiene lo siguiente:

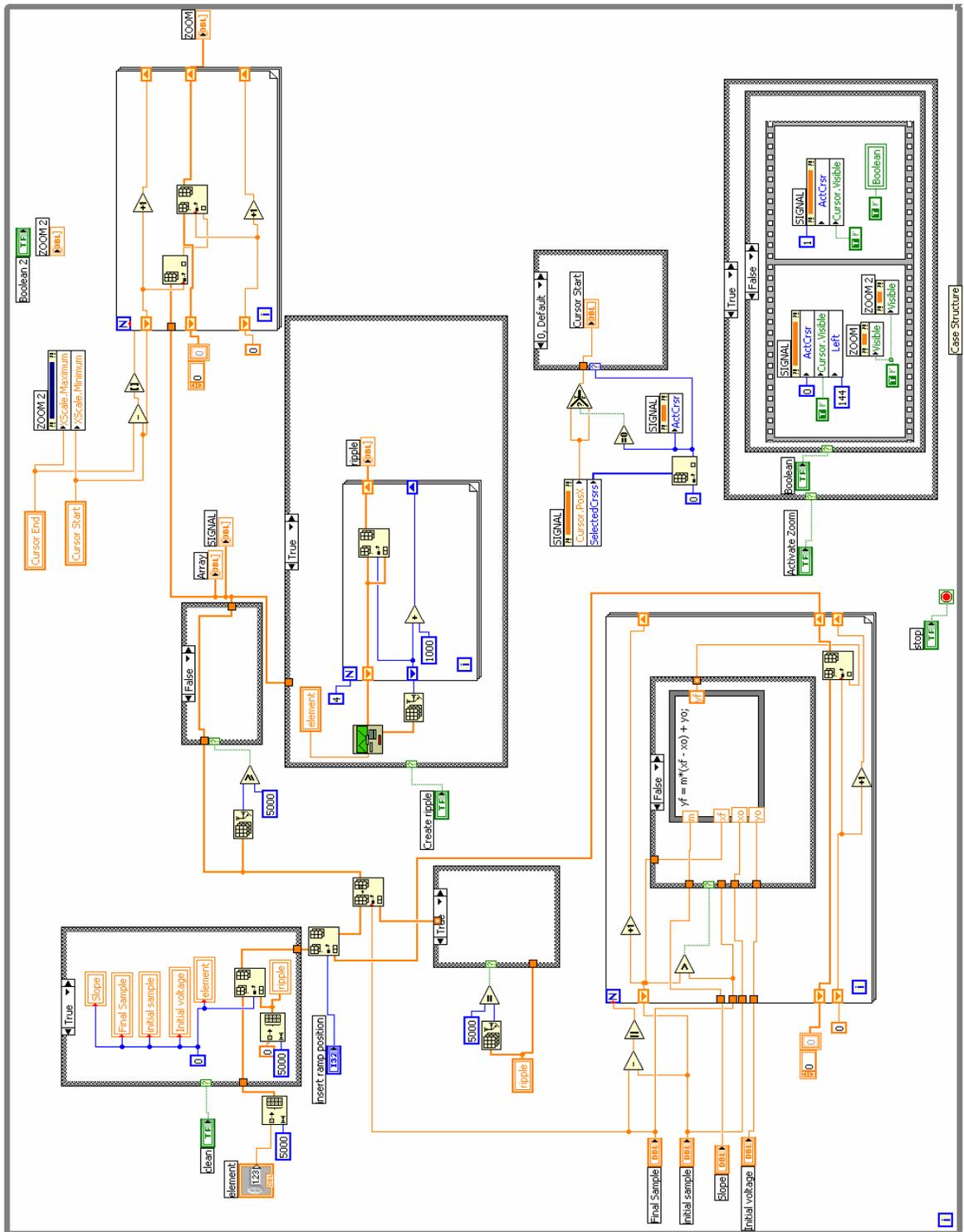


Figura 113. Programación completa Ripple Creator

Para entender mejor la programación, se va a explicar esta por secciones:

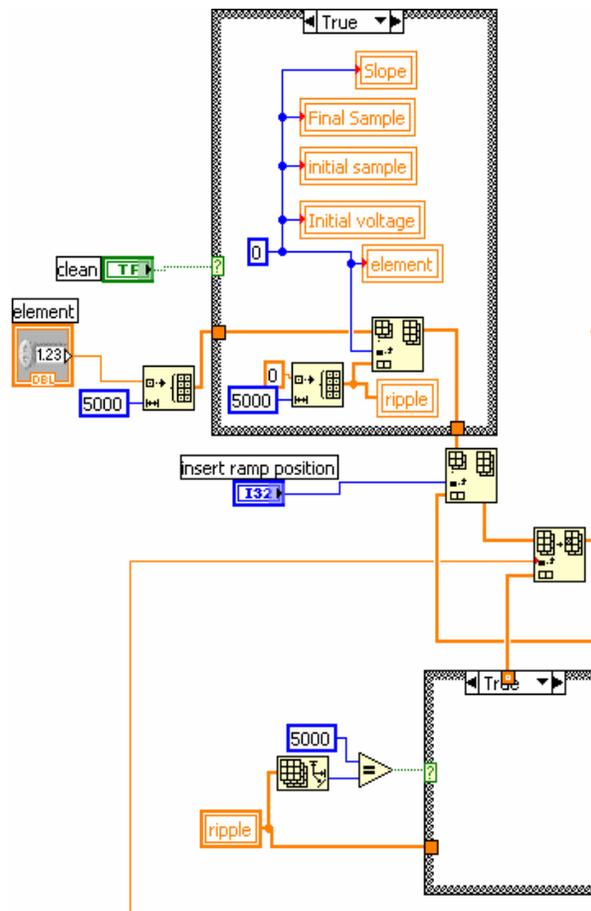


Figura 114. Funciones de Clear, tamaño de array

En esta primera parte, el usuario introduce el valor numérico o voltaje del step o de una señal dc constante. Este valor es introducido a un initialize array el cual va a generar un array de cinco mil posiciones iguales con este valor.

Posteriormente se tiene un case estructura, allí se pregunta si el usuario oprimió el botón de clean. Si no lo ha hecho, no pasa nada y se sigue trabajando la señal pero por el contrario si el usuario oprimió el botón de clean, los controles usados en el Front panel de este SubVI serán clareados puestos en cero al igual que la señal que se pretende crear.

Hecho esto, se introduce el array que contiene la rampa y también el array con los valores de rizo, claro que para este ultimo hay que revisar que no supere el tamaño de 5000 muestras, si lo supera se toman las primeras cinco mil muestras y las siguientes se borran.

En la siguiente grafica se observa la programación correspondiente para la creación de la rampa. Se utiliza un sencillo formula node y la ecuación de la recta evaluada en un punto.

Utilizando el For Loop se evalúa la ecuación en todos los puntos correspondientes desde el valor inicial en el eje X (Muestras) hasta el valor final y los resultados de estos, se almacenan en un array nuevo tal y como se realizo para el SubVI de Start Up Creator. (Ver Figura 102)

Para la creación del rizado se ejecuta un SubVI que es llamado cuando el usuario oprime el botón de Create Ripple este SubVI entrega como resultado una matriz de valores de 1000 muestras.

Estas 1000 muestras son tomadas y copiadas cuatro veces más para ser introducidas en un array ahora con 5000 muestras de la señal.

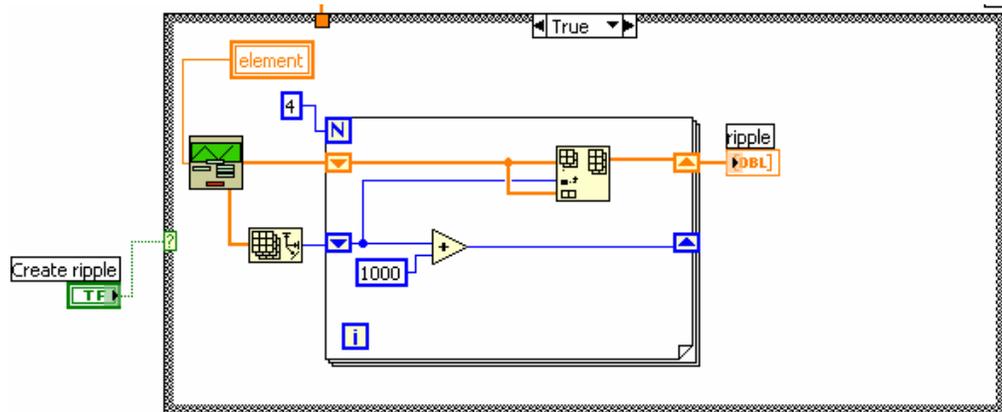


Figura 115. Inserción del rizado

Para lograr que el usuario pueda realizar zoom, se implemento el sistema de zoom en él se selecciona una porción de la grafica y en un waveform graph diferente se visualiza la porción deseada.

Primero es necesario capturar los valores de posición de los cursores, y también detectar que cursor es el que se activó, la forma más sencilla de hacer esto es utilizar las propiedades de Property node del waveform graph que va a contener los cursores.

En la siguiente figura, se puede observar la programación de captura de los nodos propiedad, esta no difiere en nada a la programación usada antes en el SubVI de Start Ups creator.

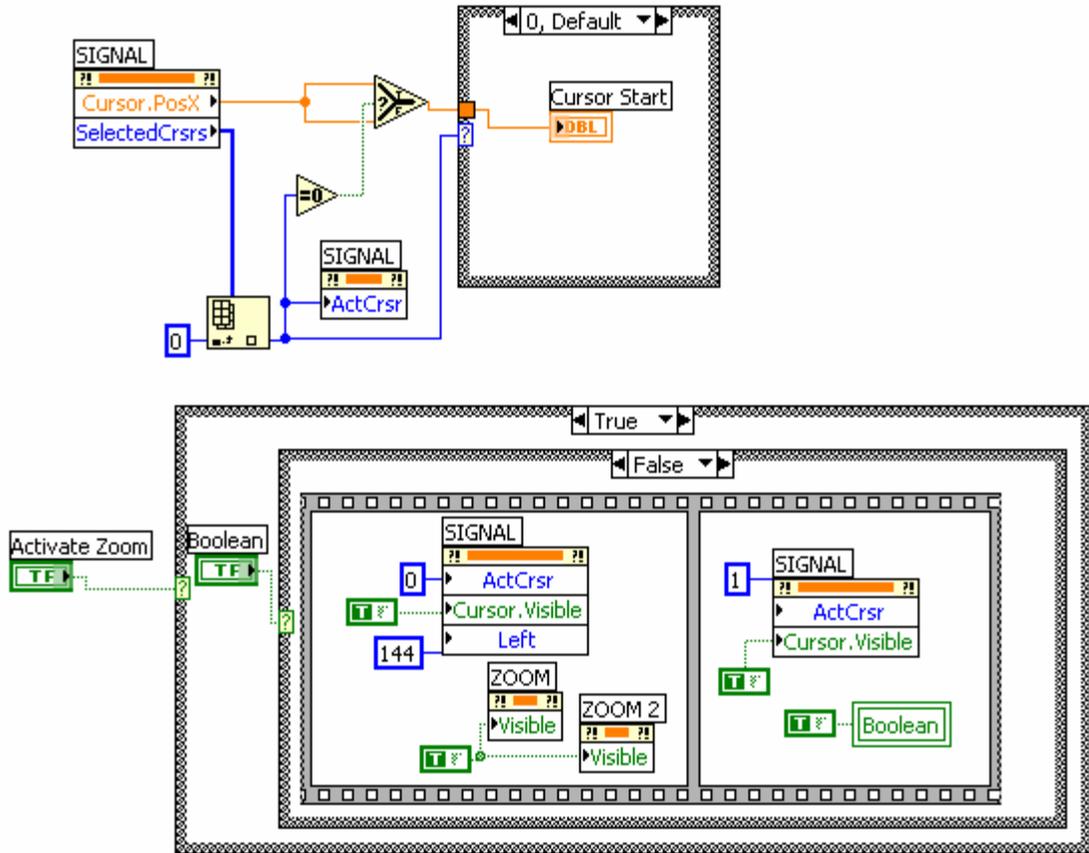


Figura 116. Programación de las propiedades del waveform graph

Lo realmente nuevo es la inclusión de nuevas propiedades para los waveform graphs, que permiten que desaparezcan o aparezcan no solo en una sola posición sino en dos posiciones diferentes.

Al oprimir el botón de zoom el waveform Graph que se encuentra centrado en la mitad del front panel, se corre hacia la izquierda, y a la derecha se introduce otro waveform graph donde se visualizara el zoom.

También se ocultan los cursores para cuando el usuario no desea hacer zoom.

Como se mencionó anteriormente para crear el rizado, se introdujo un nuevo SubVI al cual se le llamo Generator, este SubVI tiene la siguiente forma:



Figura 117. Generator SubVI

Este VI tiene como entrada el valor de offset que corresponde al nivel de voltaje en dc que el usuario seleccionó inicialmente para crear la rampa. La salida de este corresponde únicamente a la señal seleccionada con los parámetros configurados por el usuario, de igual manera y como se mencionó anteriormente este vi entrega un array de 1000 posiciones.

Al ejecutar el SubVi, este despliega un Front Panel como el que se muestra a continuación:

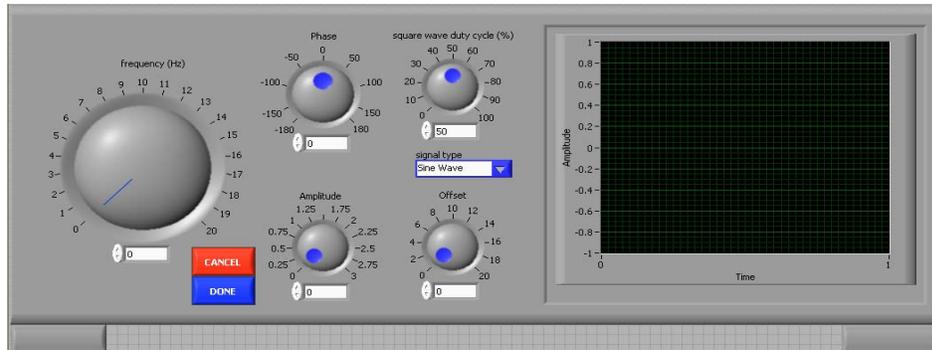


Figura 118. Panel Frontal Generator

A simple vista se puede observar que es un generador de señales. Posee perillas para ajustar la frecuencia, amplitud, offset, fase, e incluso el ciclo de trabajo para una señal cuadrada, así como también se puede seleccionar el tipo de señal que se desea trabajar.

En cuanto a la programación se tiene:

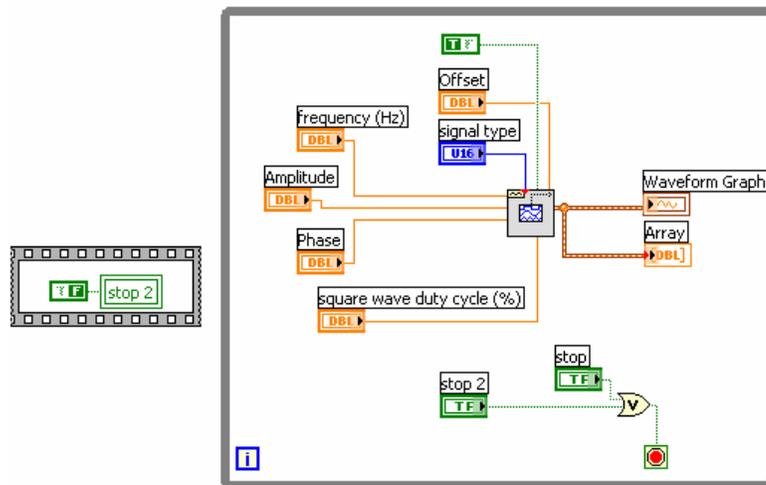


Figura 119. Programación interna del Generator SubVI

Como se observa en la figura anterior, toda la programación esta contenida en un solo VI, que se conoce como Basic Function Generator y su función como su nombre lo indica es crear una señal de salida basándose en el tipo de señal seleccionada por el usuario.

Finalmente, esta opción de crear señales con rizados fue incluida en el la fase de configuración de señal.

4.8 GENERADOR DE SECUENCIAS (CREATE SEQUENCE)

Con el objetivo de lograr generar una secuencia completa de encendido y apagado o cualquier combinación posible, se ha creado un SubVI, desde el cual se pueden cargar dos archivos de formato LVM.

Los archivos previamente guardados en una librería, permiten al usuario por ejemplo generar una secuencia de encendido y de apagado de manera automática sin necesidad de cargar por separado los archivos.

De igual manera el usuario podrá almacenar el archivo de forma completa en formato LVM para después generarlo por aparte por el canal 1, el cual permite generar archivos de mayor tamaño.

El usuario puede acceder a este nuevo subvi con presionar solo un botón en el panel de control del programa principal del Generador Multivoltage. Hecho esto el usuario se encuentra con el siguiente panel de control:

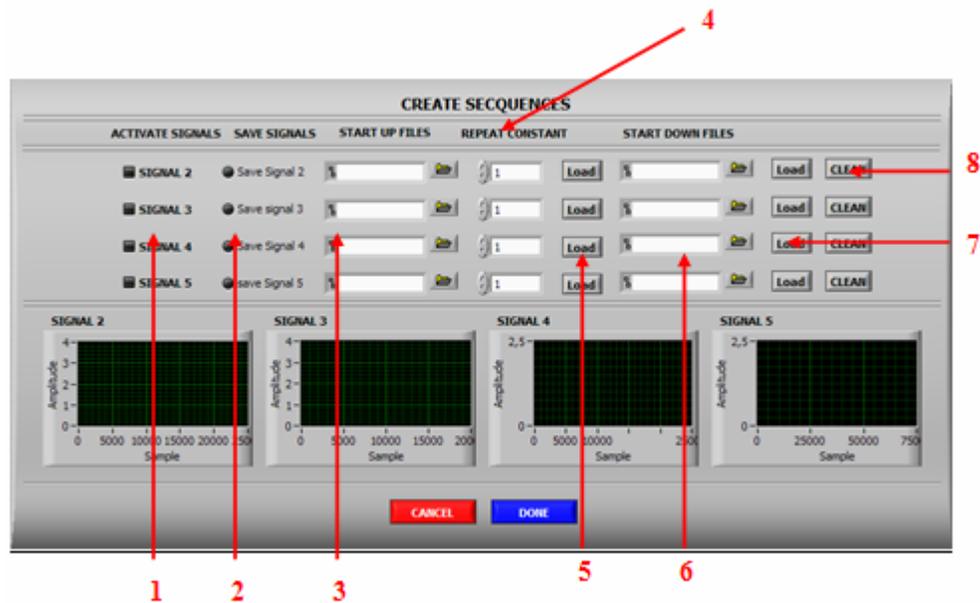


Figura 120. Panel de control Generador de secuencias

Como se puede observar en el grafico anterior, el usuario se encuentra con una serie de funciones de fácil manejo que le permitirá, manipular los canales de generación desde las señales 2 hasta la señal número 5 de la siguiente manera:

- 1- Aquí el usuario puede encender el canal que desea utilizar para generar la secuencia, por defecto estos se encuentran apagados.
- 2- En este punto el usuario puede decidir de igual manera que señal quiere guardar. Como se mencionó anteriormente la señal se guarda en un formato LVM para después ser generada por el canal 1.
- 3- Corresponden a los Path o rutas donde el archivo inicial se encuentra guardado, como por ejemplo una secuencia de encendido.
- 4- El programa también permite al usuario generar un valor constante correspondiente al último valor del primer archivo guardado cuantas veces el usuario lo desee.
- 5- El botón de Load 1 actualiza la carga del archivo así como también la generación repetitiva del último valor de este archivo.
- 6- Corresponde a un segundo Path para generar un archivo consecutivo, como por ejemplo una secuencia de apagado.
- 7- Load 2, carga el segundo archivo de la secuencia.
- 8- Clear, botón que permite borrar lo que se haya realizado.

De igual manera el usuario cuenta con unos pequeños Waveforms graphs donde puede observar la señal que esta desarrollando, así como el botón de cancel para salir del SubVI sin alterar alguna ejecución anterior y el botón de Done para ejecutar lo realizado en el SubVI.

En cuanto a la programación, el SubVI esta localizado en la fase de generación de señal, es decir para que este SubVI se ejecute se necesita que el botón de Start Generation se encuentre oprimido de lo contrario el programa no se ejecuta.

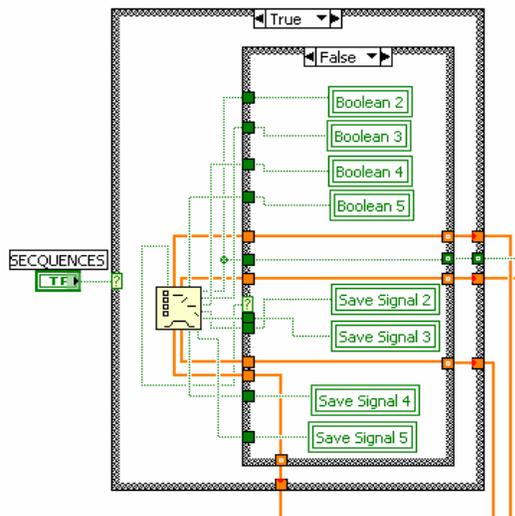


Figura 121. Programación del Generador de secuencias en el Generador Multivoltaje

Como se observa en la figura 121 el SubVi de Generación de secuencias funciona al oprimirse el botón de Sequences, en el caso de que se oprima cancel desde el SubVi el segundo case cambia a verdadero donde el programa no hace ningún cambio.

Así de igual forma si el botón de Sequences no se oprimió nunca el case mas grande se cambia a falso donde se encuentran las variables locales de cada señal para generarse. Dentro del SubVi encontramos la siguiente programación:

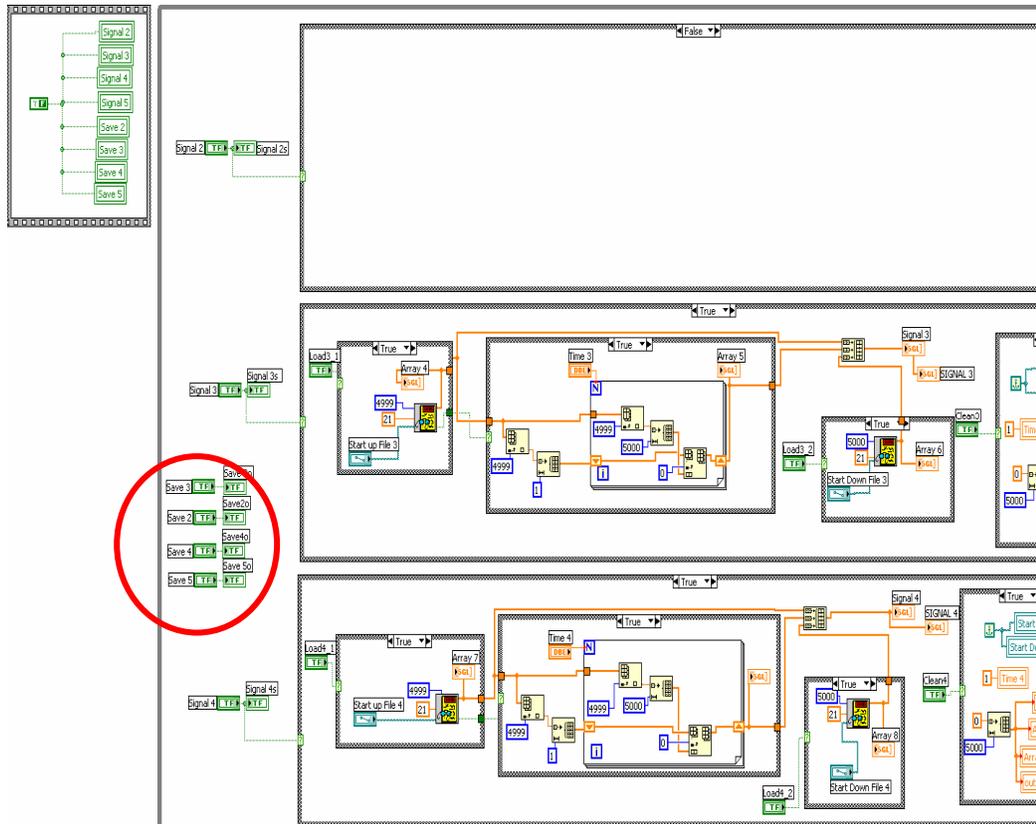


Figura 122. Programación interna generador de secuencias

En la grafica anterior se puede observar la programación del subVI aunque no completa pues solo se encuentra la programación de las señales 2, 3 y 4, sin embargo, la programación de la señal cinco es igual que de las señales antes mencionadas, solo difieren de sus variables locales.

La grafica muestra un Flat sequence, en la parte superior izquierda, dentro del cual se encuentran las variables locales para los botones de encendido y salvado de las señales conectados a una constante booleana en estado de false con el propósito de establecer las condiciones iniciales del SubVI.

De igual forma encerrados dentro de un círculo se encuentran los controles de los botones de salvar señal de todas las señales conectados a un Led indicador, el cual no se visualiza en el SubVI, pues solo se usa con el propósito de establecer el estado del botón y ejecutar este estado como una salida del SubVI.

Se observa también que la ejecución de las funciones de cada señal se encuentran dentro de un case Structure el cual obviamente es controlado por el botón de encendido de cada señal, si el botón no es presionado, no sucede nada, en caso contrario se tiene algo como lo siguiente:

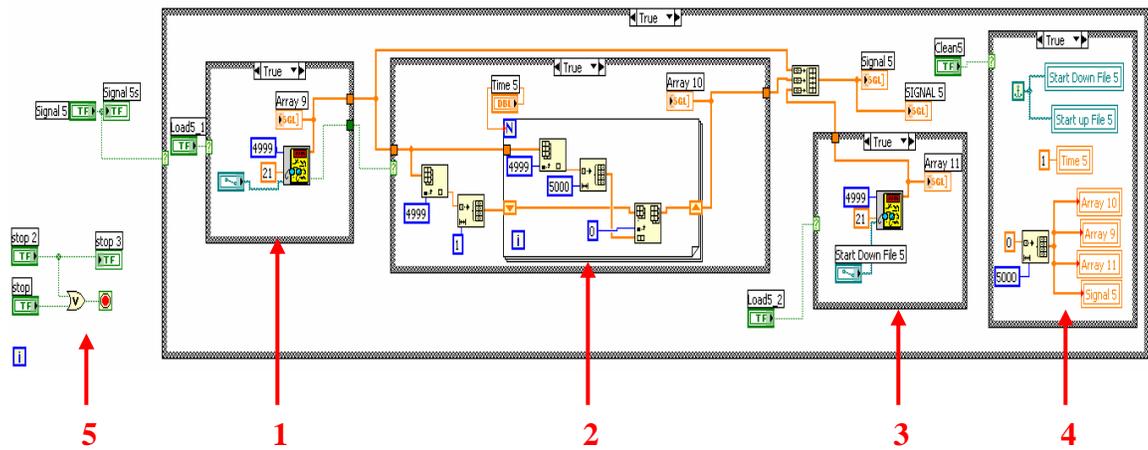


Figura 123. Programación generador de secuencia un solo canal

- 1- Dentro de este case structure que esta controlado por el botón de Load 1, se encuentra el SubVI Read LVM, el cual como su nombre lo indica permite leer un archivo LVM. El subVi esta configurado para que lea 5000 muestras a partir del renglón 21 del archivo LVM.
- 2- Aquí se captura el ultimo valor del primer archivo de 5000 muestras para generar un segmento de señal de valor constante con este valor tantas veces el usuario lo desee generar.
- 3- Aquí se lee el segundo archivo LVM para luego junto con lo antes mencionado agrupar las señales dentro de un concatenate array y así poder generar la secuencia.
- 4- Esta corresponde a la programación para cuando el botón de clear es presionado. Aquí se borran los array que contienen los datos de las señales así como también se borran los paths donde se escribe la ruta para abrir los archivos.
- 5- Finalmente, la programación para los botones de Cancel y Done. En el caso de Done el programa se sale normalmente, para el caso de cancel ocurre lo

mismo pero para que ningún cambio realizado en el SubVI surta efecto se coloca un indicador el cual permite en el programa principal identificar el cancel, y así hacer que en el programa principal no ocurra nada.

4.9 RECARGA DE SEÑALES DESDE ARCHIVO AL TIEMPO

La programación de la figura 124 corresponde a una nueva característica del generador multivoltaje para recargar archivos cvs o lvm, una vez que estos se hallan cargado en la barra de path de cada señal en la fase de configuración de señal.

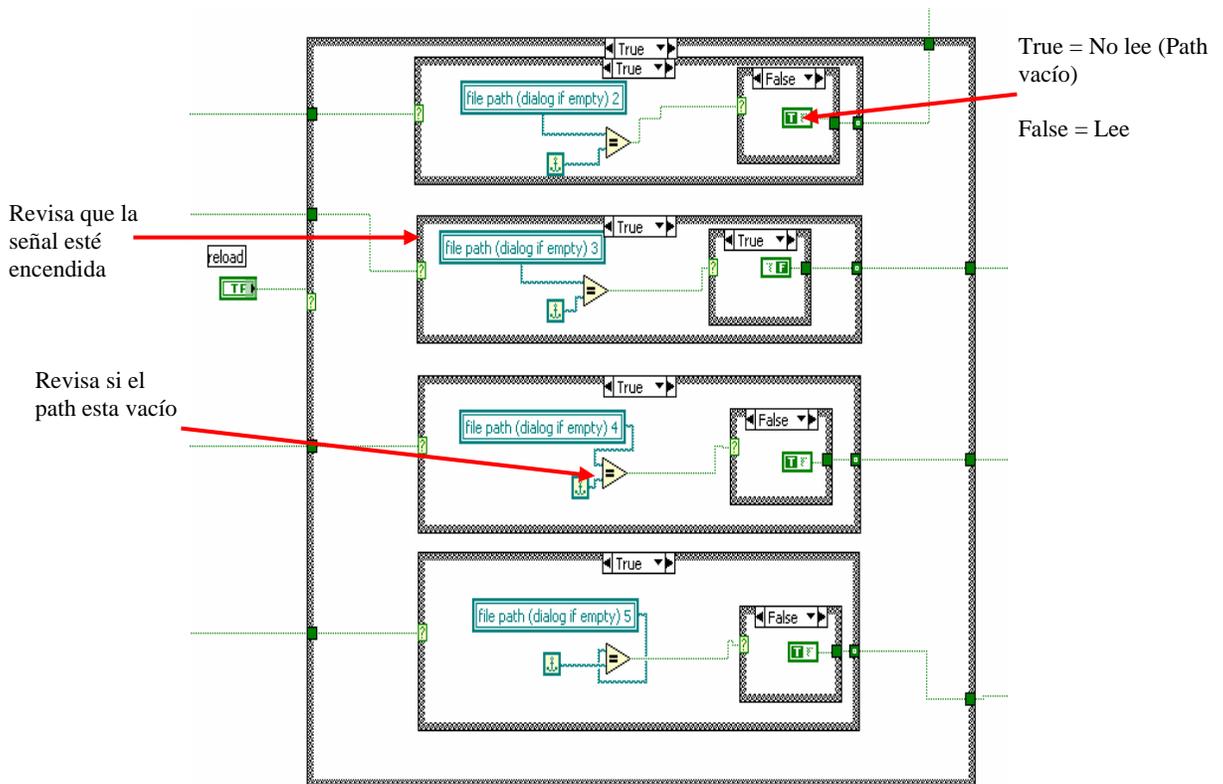


Figura 124. Recarga de señales desde archivo

Como se observa en la figura, primero se compara el file path de cada canal con una constante de path vacía, esto en caso de que si el path esta vacío, no se ejecute ninguna lectura. También antes de ello se mira que los canales para leer se encuentren encendidos.

La figura 125 muestra como se acciona la función de lectura, en la fase de configuración de señal. Allí el select cuando esta en verdadero, coloca el valor

numérico de 6 en el Case structure, el cual a su vez ejecuta el código para lectura de archivo.

Si el case se encuentra en falso, la programación que se ejecuta para buscar el valor de verdadero de alguno de los botones del Cluster funciona normalmente.

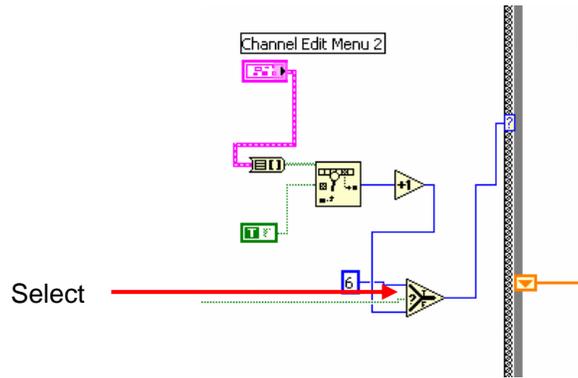


Figura 125. Función Select de labview

5. CORRECCION DE ERRORES

5.1 CORRECCIONES EN EL START UP CREATOR

Se encontró que algunas funciones del Start Up creator se encontraban restringidas a ciertos valores para su funcionamiento. Con el propósito de mejorar y ampliar la funcionalidad de este VI se realizaron algunos cambios sobre los siguientes parámetros:

- Cambio de Rango de los controles.
- Mejoramiento de la función de Start Delay.
- Nombramiento de los archivos de Excel.
- Nombramiento de los archivos en secuencia.

5.1.1 Cambio de Rango de los controles.

En el caso de la opción de Offset, Stara delay y last value voltage se cambio la escala de valores para que se puedan introducir valores negativos, solo modificando el valor inferior de la escala de cada uno de los controles.

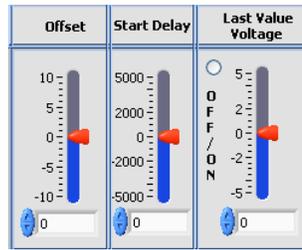


Figura 126. Controles de Offset, Start delay

5.1.2 Mejoramiento de la función de Start Delay

En el Start delay si se quería desplazar la señal hacia la izquierda no era posible porque solo se había desarrollado el programa pensando en que las señales siempre se desplazarían hacia un solo sentido. Para poder desplazar las señales en ambos sentidos a la derecha y a la izquierda se agrego un subVI el cual fue nombrado Shift array.

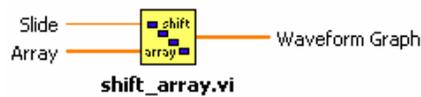


Figura 127. SubVI para desplazar un array en dos direcciones

El SubVI solo necesita como entrada el control numérico o Slide para cambiar el desplazamiento y el array a desplazar. Su Programación esta constituida de la siguiente manera:

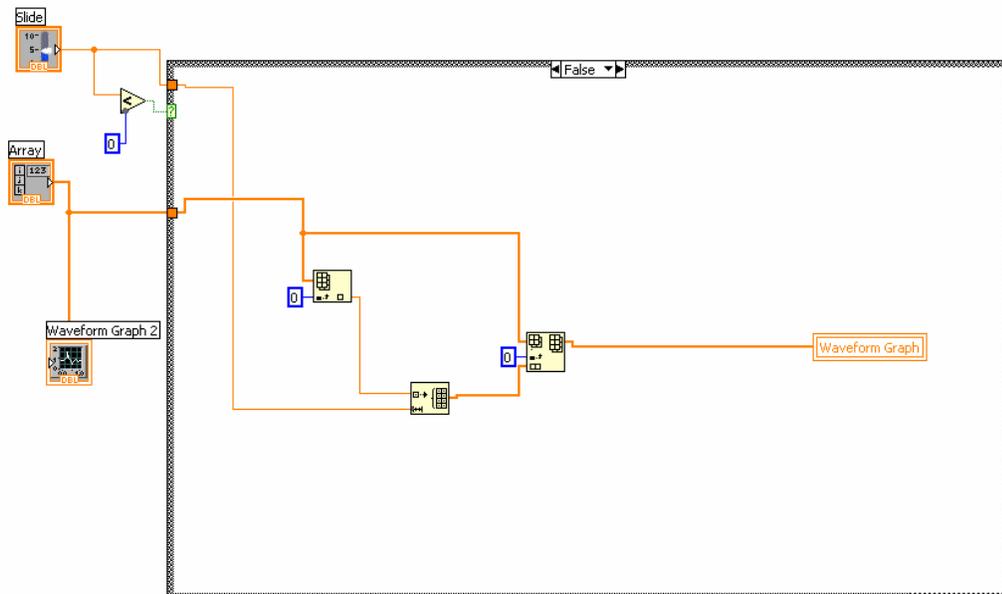


Figura 128. Desplazamiento a la derecha

Inicialmente es primordial preguntar si el valor numérico es mayor que cero para determinar si el delay es de valor positivo o negativo. Si el valor numérico es positivo se tendrá una programación como la que se muestra en el grafico de arriba.

Simplemente se toma el primer valor del array de la señal que se desea desplazar y se inserta n veces dependiendo del valor que se halla puesto en el control.

Caso contrario es necesario rotar el array de la señal tantas veces se quiera hacer el delay. Para ello se usa un rotate array, pero debido a que el rotate array coloca en la cola del array el primer valor rotado y así sucesivamente hace necesario al nuevo array rotado eliminarle tantas posiciones como se halla rotado a la cola de este y para ello se utiliza un delete from array.

También es importante añadir al nuevo array el valor numérico de su última posición. Para capturar nuevamente este valor se usa otro delete array.

Finalmente se introducen n veces este valor numérico en el array rotado a partir de su última posición como se muestra en la siguiente figura.

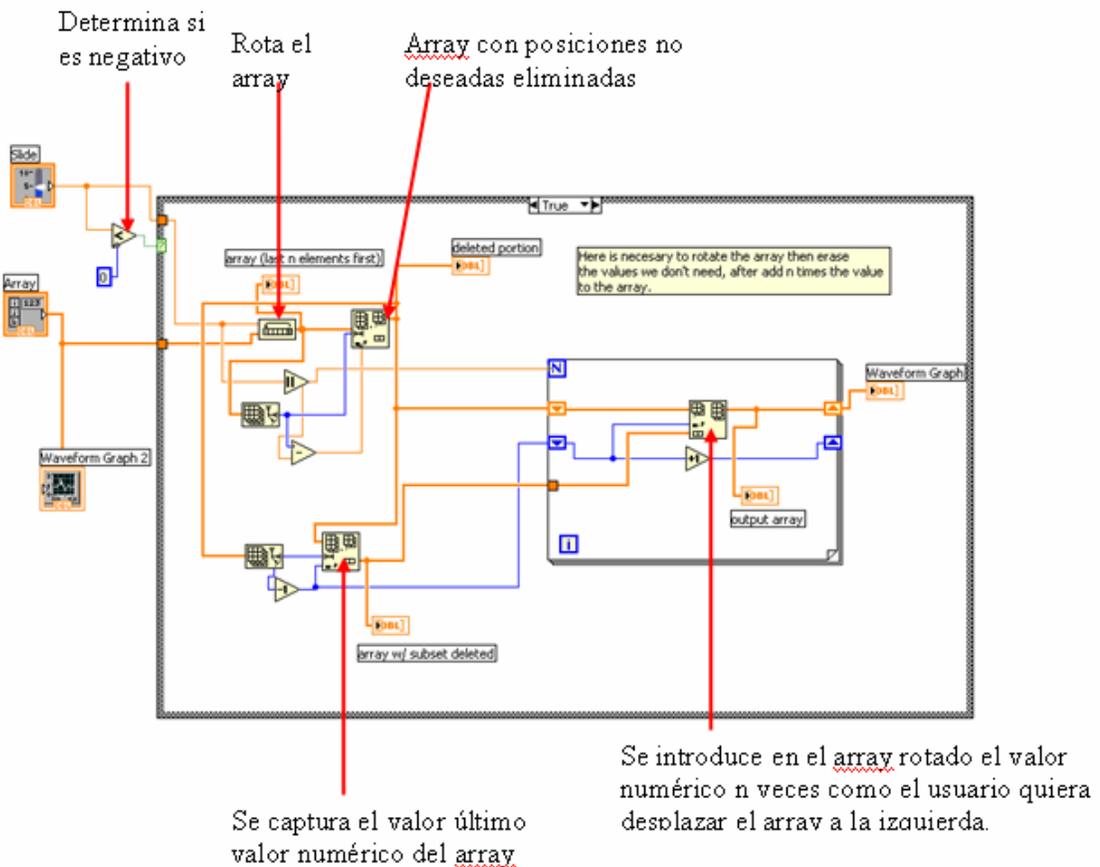


Figura 129. Desplazamiento a la izquierda

5.1.3 Nombramiento de los archivos de Excel

Otra características que se modificó, fue la forma como se guarda el nombre para cuando se desea exportar a Excel debido a que el archivo se guardaba con la antigua extensión y con la extensión de Excel.

Por ejemplo; suponga que el nombre del archivo era PRUEBA y como es un archivo de medición de Labview su extensión es .LVM de tal manera que el nombre completo del archivo es PRUEBA.LVM. Ahora bien, suponga que se va a exportar a Excel para lo cual se cambia la extensión del archivo a .XLS.

Así cuando se exportaba el nombre completo quedaba PRUEBA.LVM.XLS, de igual manera Excel lo abriría y lo entendería pero no es muy cómodo para el usuario ver un nombre de esta manera, puede llegar a ser confuso.

Para contrarrestar esto se creó un SubVI llamado Change extension.vi que tiene la siguiente apariencia:



Figura 130. SubVI Change extention

Dentro de este SubVI podemos encontrar la siguiente programación:

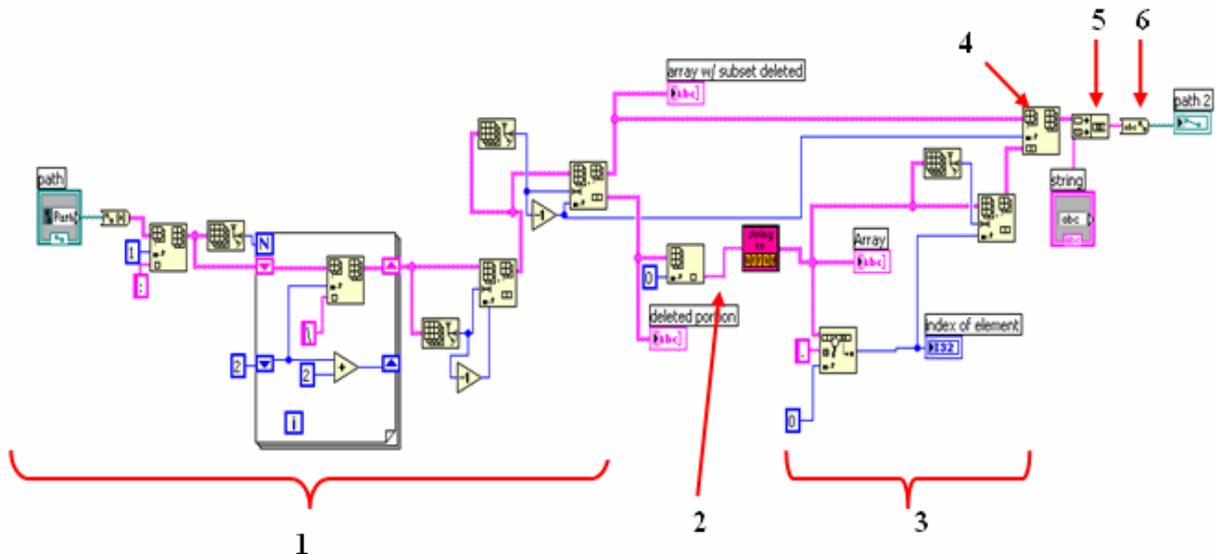


Figura 131. Programación interna SubVI Change Extension

- 1- Inicialmente se toma el Path o la dirección completa de donde se quiere guardar el archivo, por ejemplo, C:\Documents and Settings\mendoza\Desktop\test\prueba.lvm. Tenemos que el archivo prueba es de extensión lvm y se encuentra guardado en el escritorio en la carpeta Test.

Esta dirección para labview corresponde a un tipo de dato completamente diferente a String, o dato numérico, entre otros. Es necesario entonces cambiar el dato y para ello se usa la opción de path to array of strings que convierte la dirección a un array de string o de letras. Por ejemplo:

Path: C:\Documents and Settings\mendoza\Desktop\test\prueba.lvm

array: [C – Documents and Settings – mendoza – Desktop – test – prueba.lvm]

Como se observa en el ejemplo anterior se crea un array pero dentro de él no se encuentran los símbolos de dos puntos (:) y los símbolos de slash (\).

Para insertar estos símbolos, después de la primera posición del array se introducen los dos puntos (:) y usando un for loop se introducen los slashes quedando así para el anterior ejemplo un nuevo array como el siguiente:

[C - : - \ - Documents and Settings - \ - mendoza - \ - Desktop - \ - test - \ - prueba.lvm - \]

Pero debido a que un slash se ha agregado de más es necesario borrarlo y para ello se usa un delete from array dejando finalmente el array en las siguientes condiciones:

[C - : - \ - Documents and Settings - \ - mendoza - \ - Desktop - \ - test - \ - prueba.lvm]

Para cambiar la extensión hay que capturar el nombre del array y para capturarlo se usa nuevamente un delete from array separando el array del ejemplo de la siguiente manera:

Array1 = [C - : - \ - Documents and Settings - \ - mendoza - \ - Desktop - \ - test - \]

Array2 = [prueba.lvm]

- 2- Tomando el array 2 e introduciéndolo en el subvi de String to array obtenemos el siguiente resultado:

Array2 = [p – r – u – e – b – a – . – l – v – m]

- 3- Para eliminar la antigua extensión hay que saber a partir de que posición se encuentra. No se puede eliminar una extensión a partir de un valor fijo porque no todos los nombres tienen el mismo tamaño. La solución mas sencilla es buscar en qué posición del array del nombre se encuentra el punto que separa el nombre de la extensión y borrar a partir de allí. Para buscar el punto se uso el Search 1D array y para eliminar nuevamente el delete from array.

Para el ejemplo anterior el nuevo array quedaría:

Array2 = [p - r - u - e - b - a]

- 4- Ahora es necesario insertar al array que contiene el path completo en donde se almacena el archivo el nombre del archivo sin extensión así:

[C - : - \ - Documents and Settings - \ - mendoza - \ - Desktop - \ - test - \ - p - r - u - e - b - a]

- 5- Haciendo uso de un concatenate String se agrega la extensión .xls
- 6- Finalmente usando la opción array of strings to path convertimos nuevamente la dirección con la extensión del archivo cambiada a path, quedando finalmente:

C:\Documents and Settings\mendoza\Desktop\test\prueba.xls

Anteriormente se mencionó el uso de otro SubVI llamado string to array, este SubVi se creo exclusivamente para llevar cada letra de una palabra a un array.



Figura 132. SubVI String to array

La programación es la siguiente:

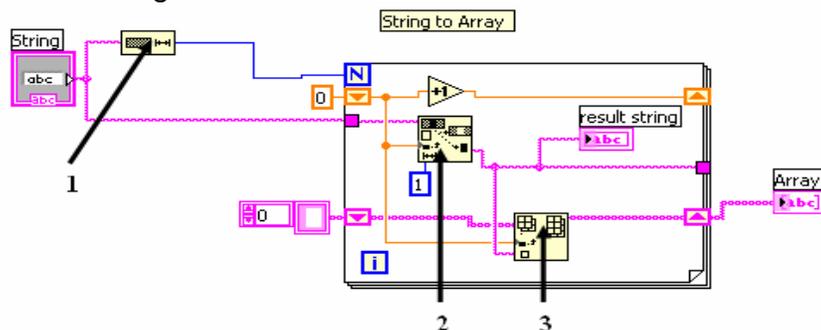


Figura 133. Programación interna String to array SubVI

- 1- Inicialmente se introduce la palabra y se mide la longitud de esta.
- 2- Dentro de un for loop se ejecuta un Replace substring, el cual inserta, borra o reemplaza un substring en un punto especificado por el usuario a un string ya existente. Para este caso se configuró para que extrajera letra por letra del array.
- 3- Haciendo uso de un insert into array se introducen las letras a un nuevo array.

Por ejemplo si la palabra es PRUEBA el nuevo array queda [P – R – U – E – B – A]

5.1.4 Nombramiento de los archivos en secuencia.

Otro inconveniente ocurría cuando se guardaban los cinco archivos de manera consecutiva. Por ejemplo, si se crearan los cinco archivos desplazados entre si un cierto número de muestras el nombre del archivo se guardaba de la siguiente manera:

Prueba.lvm0.lvm
 Prueba.lvm1.lvm
 Prueba.lvm2.lvm
 Prueba.lvm3.lvm
 Prueba.lvm4.lvm

Como se observa se guardaba usando dos veces la extensión, para corregir esto se realizó lo siguiente:

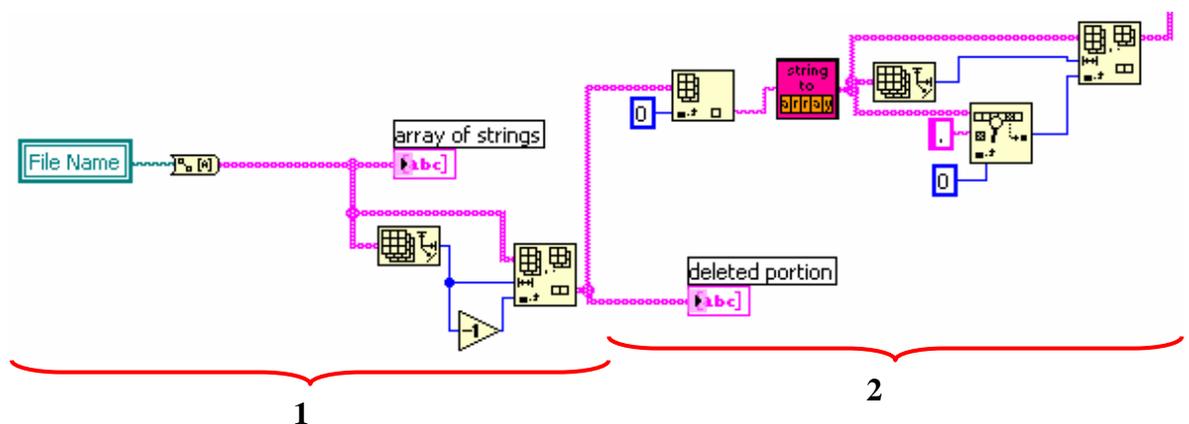


Figura 134. Corrección extensión en el nombre de un archivo

- 1- Inicialmente la programación llegaba hasta este punto, se tomaba el path se convertía a un string array y se separaba el nombre con la extensión para

después enviarlo a un concatenate array en donde se agrega un número en secuencia y el path completo junto con la extensión.

- 2- Esta sección se agregó con la finalidad de eliminar la extensión ya existente, para ello se uso el SubVI String to array y con un search 1D array se buscó el punto que separa la extensión del nombre del archivo y a partir de ahí se borra la extensión haciendo uso de un delete from array. Finalmente el nombre sin la extensión es llevado al concatenate array.

Otra de las mejoras realizadas a este SubVI fue la implementación de de un mejor sistema para nombrar las cinco señales que se van a generar cuando se presiona el botón de crear, debido a que cuando se crean los cinco archivos LVM desplazados entre si determinado tiempo, se guardan o se nombran de manera consecutiva, es decir, si el nombre del archivo es Prueba, los cinco archivos serian nombrados como Prueba0.lvm, Prueba1.lvm, Prueba2.lvm, prueba3.lvm y Prueba4.lvm.

Pero para un usuario cualquiera que vea por ejemplo el archivo prueba4.lvm no podría decir que el archivo esta desplazado en el tiempo por ejemplo 200uS, para ello se implementa la siguiente programación en la sección donde se guardan consecutivamente los archivos:

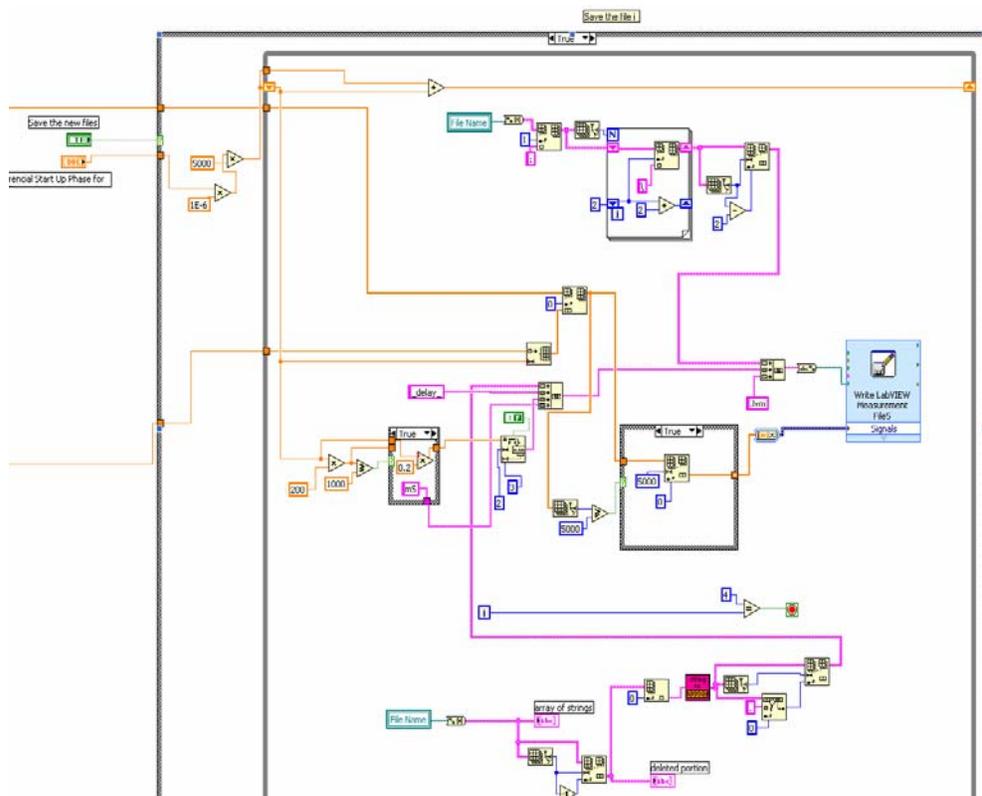


Figura 135. Corrección nombre de archivos guardados en secuencia

Como se puede observar en el grafico anterior, se toman los valores de señal que se han desplazado en el tiempo las señales para nombrar los archivos.

Primero se convierte a microsegundos el número de muestras multiplicando por 200, una vez hecho esto se pregunta si el archivo es mayor e igual a 1000 que es lo mismo que 1000uS para hacer referencia, si es menor que 1000 al nombre del archivo se le inserta la palabra delay y el valor en microsegundos.

Pero si el valor numérico es superior a 1000 o 1000uS se multiplica el número de muestras que se desplaza la señal por 0.2 para convertir a milisegundos, y al nombre del archivo se le agrega la palabra delay y el valor en milisegundos.

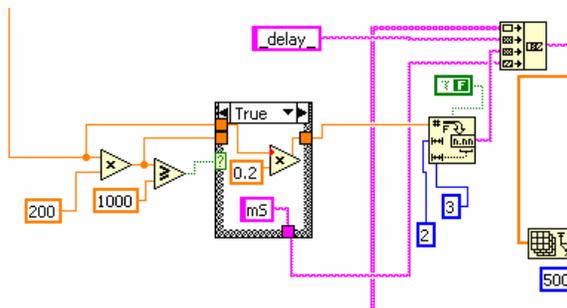


Figura 136. Adición de unidades al nombre de archivo

Como se puede observar también, se cambia la función numeric to decimal string, por numeric to fractional string, debido a que hay un producto por un decimal, para ello es necesario configurar la precisión así como el numero de decimales a utilizar.

Con esto para el nombre de los archivos del ejemplo cambiarían de la siguiente forma:

Prueba0.lvm	a Prueba_delay_200.000uS.lvm
Prueba1.lvm	Prueba_delay_400.000uS.lvm
Prueba2.lvm	Prueba_delay_600.000uS.lvm
Prueba3.lvm	Prueba_delay_800.000uS.lvm
Prueba4.lvm	Prueba_delay_1.000mS.lvm

5.1.5 Mejora en el botón de copy file.

Hay que recordar que el Start Up Creator es un SubVI que permite modificar las características o los valores de un archivo LVM. Dentro de este SubVI se encuentra como una de las funciones principales el Botón de Copy file que es el que permite como su nombre lo indica copiar un archivo LVM desde una ubicación específica a una nueva ubicación según desee el usuario.

Sin embargo, cuando se va a copiar el archivo se muestran dos ventanas, la primera que permite seleccionar el archivo y la segunda que permite cambiar el nombre y la ubicación del archivo copiado.

Estas dos ventanas presentan las opciones de cancelar, aunque no es muy usual la ejecución de esta orden, pues por lo general el usuario no se equivoca en la selección del archivo, es mas agradable para el usuario tener la posibilidad de seguir corriendo el programa sin que se presenten errores en este.

Para ello se reemplazaron los dos SubVIs Read LVM subVi y Write Measurement File por un nuevo código mas efectivo como se narra en el siguiente párrafo.

Obviamente al igual que antes se pregunta por el estado del botón copy. Si no es oprimido no sucede nada, pero si por el contrario el usuario lo oprime, una primera ventana aparecerá preguntando por el nombre del archivo que se desea copiar. Allí se presentan las opciones de aceptar y cancelar.

Sin ningún manejo de error, el programa automáticamente presenta una ventana de error cuando por cuestiones ajenas, el usuario quiere cancelar la función de copiar el archivo. El error allí presentado es clasificado por Labview como error 43, operación cancelada por el usuario. Para evitar esto, se maneja el error identificándolo con un unbundle by name y comparando el resultado de este con el numero 43, así cuando sea igual, el error es borrado.

Por otra parte, si el usuario decide copiar el archivo, una segunda ventana aparecerá preguntando por el nuevo nombre del archivo y ubicación. Allí de igual manera, el File dialog que es la estructura que se utiliza para preguntar por el archivo presenta la opción de dar cancelar arrojando como resultado el mismo error 43, si no existe ninguna programación como en el caso anterior.

Igual que antes, la solución para quitar el error es la misma, aunque se presenta un segundo error cuando el usuario da el mismo nombre del archivo a copiar y además lo copia al mismo lugar.

Labview nombra este error como error 10, duplicate path o path duplicado, de igual manera para evitar cancelar la operación del programa, se elimina el error aplicando la misma programación que en los casos anteriores aunque ahora obviamente se pregunta si el error es igual a 10.

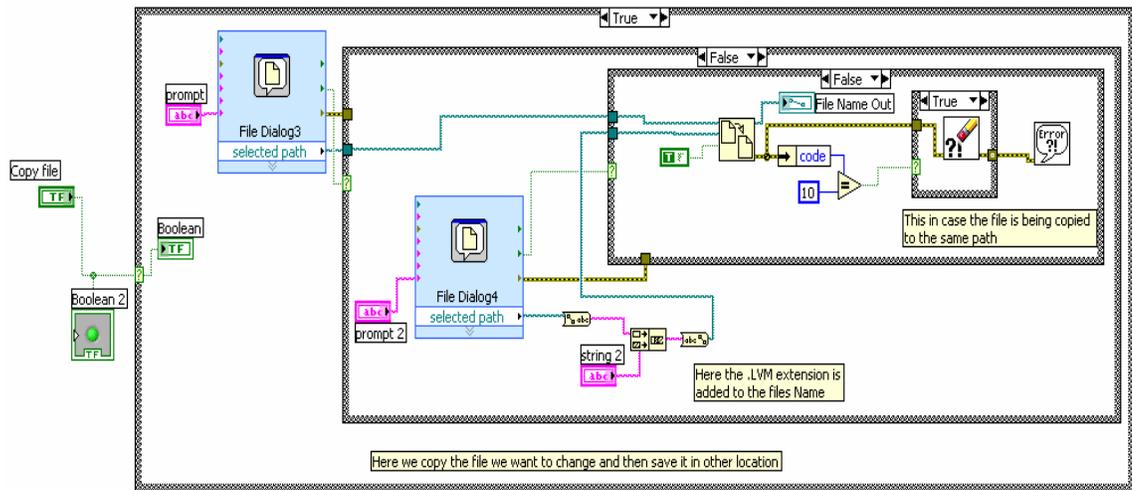


Figura 137. Corrección programación botón copy file

La figura anterior muestra la programación para cuando el botón de copy es oprimido. Adicionalmente a la hora de copiar el archivo y nombrar este no se puede guardar con la extensión .lvm pues la ventana del File dialog no se puede configurar para que se guarde con la extensión.

Por tal motivo, este procedimiento se realiza por programación, agregando al path la extensión usando un concatenate strings, aunque para ello primero es necesario convertir el path a string para así agregar la extensión y posteriormente realizar el proceso contrario para guardar el archivo.

Como se observa también en la figura anterior, el Read LVM y el write labview measurement file son reemplazados por la función de copy de la paleta de archivos de entrada y salida de Labview que se muestra a continuación.

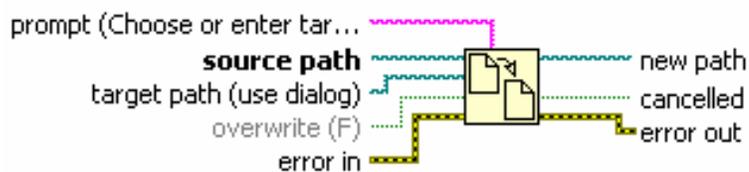


Figura 138. Función Copy

La figura 5.13 muestra la programación a la hora de corregir el error 43. También se hace necesario setear la bandera de Boolean 2 que es la que hace que el archivo sea leído por primera vez para trabajar sobre el, así como también la bandera de refresh que es la que sobrescribe un archivo en el que se esta trabajando.

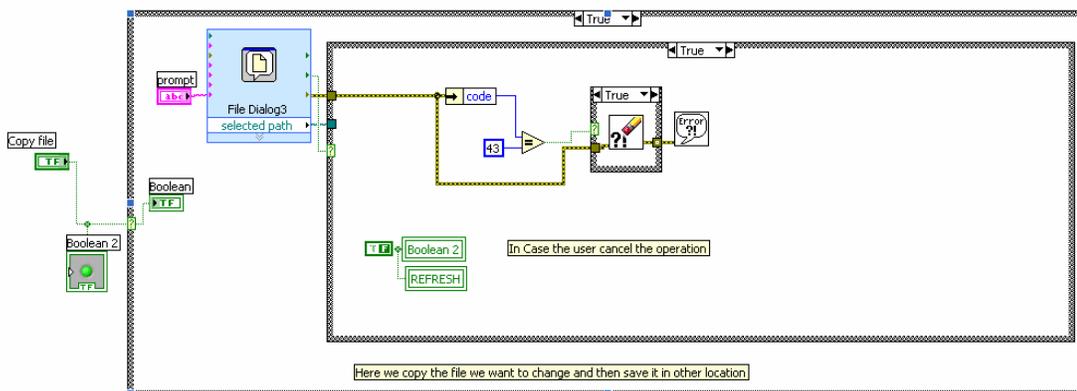


Figura 139. Corrección al dar cancel en Labview 8.2

5.2 CORRECCIÓN DE LA GENERACIÓN DE LA RAMPA

Tras seguir realizando más pruebas con el programa, se encontró que el programa fallaba a la hora de generar una señal rampa, puesto que este generaba un ciclo repetitivo de señales rampa más no generar la rampa y quedarse en el valor constante. El error es como se muestra en la siguiente figura:

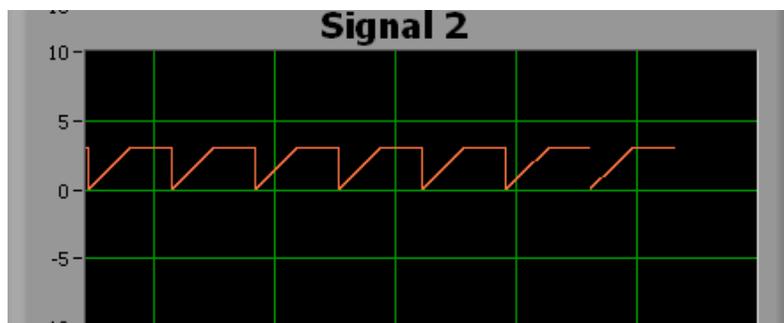


Figura 140. Error 1 generación de rampa.

Se encontró que el problema se localizaba, en el subvi de Spike_signals2_5subvi.vi nombrado anteriormente (ver figura 61), debido a que el funcionamiento de este se basa en preguntar primero el tamaño de la matriz de la señal, es decir, por ejemplo si la señal es menor que 5000 muestras no pasa nada y es generada.

Pero si la señal supera las 5000 muestras entonces haciendo uso de un delete from array el programa toma las primeras 5000 muestras y las restantes las almacena en una variable local de igual nombre que la señal de entrada, es decir, la retroalimenta para después generar las muestras restantes en porciones de 5000 muestras debido a que la tarjeta generadora funciona de esa manera.

Sin embargo al ejecutar paso por paso el programa se encontró que cuando la matriz de la señal pasaba por allí el subvi si cortaba las primeras cinco mil muestras, pero a la siguiente iteración, la señal de entrada no eran las muestras restantes sino nuevamente la señal completa sin cortar, es decir, cada vez que se ejecutaba el subvi, este solo cortaba las primeras cinco mil muestras de la señal y las generaba como se muestra en la figura 140.

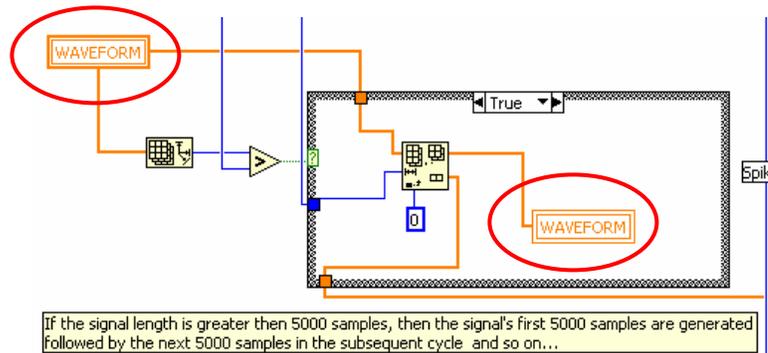


Figura 141. Comprobación tamaño de matriz.

La figura 141 muestra la configuración original, de una parte del subvi, puesto que allí también se genera el spike, pero antes se revisa el tamaño de la señal.

Como una de las primeras soluciones se intento cambiar la variable local que se encuentra dentro del case de la figura anterior y recargar la señal de entrada haciendo uso de una variable global, es decir, utilizando una variable que permita servir como puente entre dos subvis. Para ello el programa principal se modificaría de la siguiente manera.

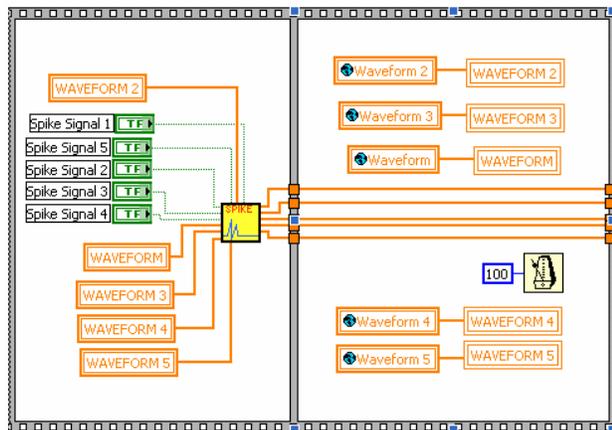


Figura 142. Modificación de Spike Signal con variables globales

Inicialmente se recargan las señales provenientes de la fase de configuración de señal que entran al subvi, donde se pregunta por el tamaño de la matriz de la

señal. Si este es superior a cinco mil, el subvi toma las primeras cinco mil muestras y las restantes son guardadas en una variable global.

Como se muestra en la figura 142, se reescribe la señal de entrada para la siguiente iteración, sin embargo el error persistía y además no permitía que la señal fuera clareada y que en la fase de configuración de señal al detener la generación siempre se observara una línea horizontal en un valor diferente de cero. Como solución a esto se agregaron también las variables globales a la salida de la fase de configuración de señal quedando de la siguiente manera:

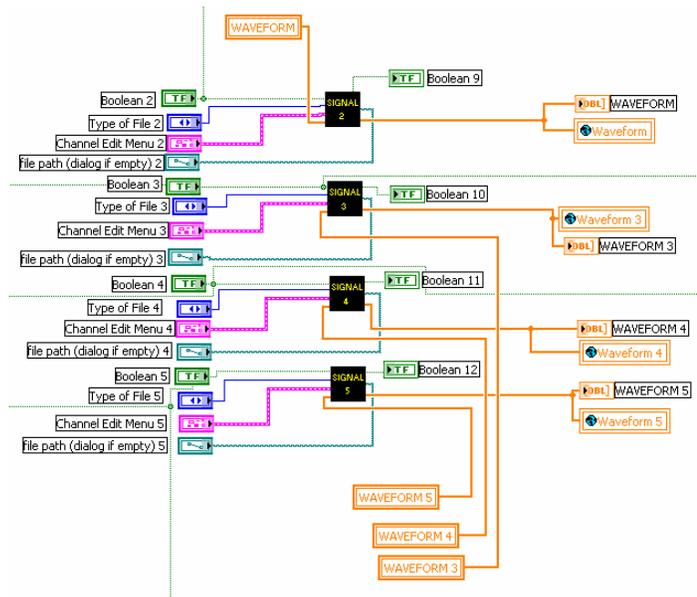


Figura 143. Configuración de señales con variables globales.

Sin embargo a la hora de generar la señal rampa, aunque el resultado mejoro para las rampas que partían desde el punto cero, para aquellas rampas desplazadas en el tiempo o con picos o perturbaciones se presentaban errores como el que se muestra en la figura de abajo llegando algunas veces a ser mas critico y realizar el mismo comportamiento que al principio durante un tiempo considerado.

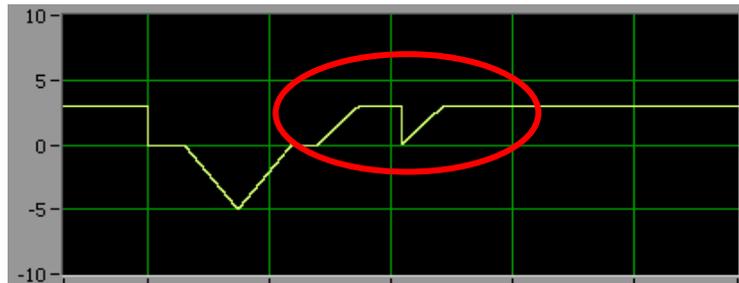


Figura 144. Error 2 de generación de rampa

Tras varios intentos, agregando variables y eliminando otras, buscando por muchos medios, se encontró que el error no solo estaba en el subvi de Spike_signas2_5.subvi sino que también se encontraba en el subvi de configuración de señal, pues este a la salida muestra la señal resultante y que se desea generar pero al ejecutar el programa paso a paso se encontró que por cada iteración donde se muestra la señal que se quiere generar, el subvi alcanza en la siguiente iteración cambiar a la posición en que la fase de configuración de señal se apaga y genera una señal de cinco mil muestras cargada de ceros.

Este switcheo de señales no permitían el correcto funcionamiento del programa y esto debido a que la programación contenida en los subvis de configuración de señal y spike_signas12_5 es una programación constante que se ejecuta durante todo el tiempo, lo cual hace que se complique la programación para un subvi si se desea que se ejecute todo el tiempo.

La verdadera función de un subvi es que se ejecute cuando sea llamado y después el programa retorne donde iba, aunque se puede hacer lo contrario sin embargo resulta en reparos como programación adicional y redundante.

Como solución final se decidió retornar la programación como se encontraba y eliminar estos dos subvis, encontrando también sorprendentemente que la velocidad de ejecución del programa se incremento un poco.

5.3 CORRECCIÓN DEL READ FROM FILE

Otro error que se corrigió fue al leer una señal en la fase de configuración de señal, es decir, a la hora de recargar una señal lvm o cvs, y presionar el botón de read. Esta se cargaba pero después de realizar la prueba y si el usuario o la persona que estuviera ejecutando el programa quisiera nuevamente recargar la señal, se presentaba un error en labview nombrado como error 4 acción detenida por el usuario. Esto pues la lectura del archivo se estaba realizando con el VI Express read labview measurement file.

Debido a la cantidad de inconvenientes con el vi Express de Read labview measurement y la necesidad de uso de este, se encontró que en su programación interna hay una bandera llamada first call, que se ejecuta cuando se lee o se llama a un archivo por primera vez.

Además, esta bandera se ejecuta por una orden dada al comparar el path de la señal en cada iteración, si el path es diferente en cada iteración quiere decir que se abre o se llama a un archivo nuevo pero si el path es igual, quiere decir que el archivo que se va a abrir es el mismo, esto con el fin de que después de cargarse

por primera vez, el puntero de final de archivo EOF (End of File) se setea para leer una nueva señal, pero si es la misma el puntero no se setea, y se muestra el error. Dos maneras de solucionar esto es engañar al VI Express dentro de su programación para que siempre compare el path con un path constante o activar de alguna manera la bandera de first call.

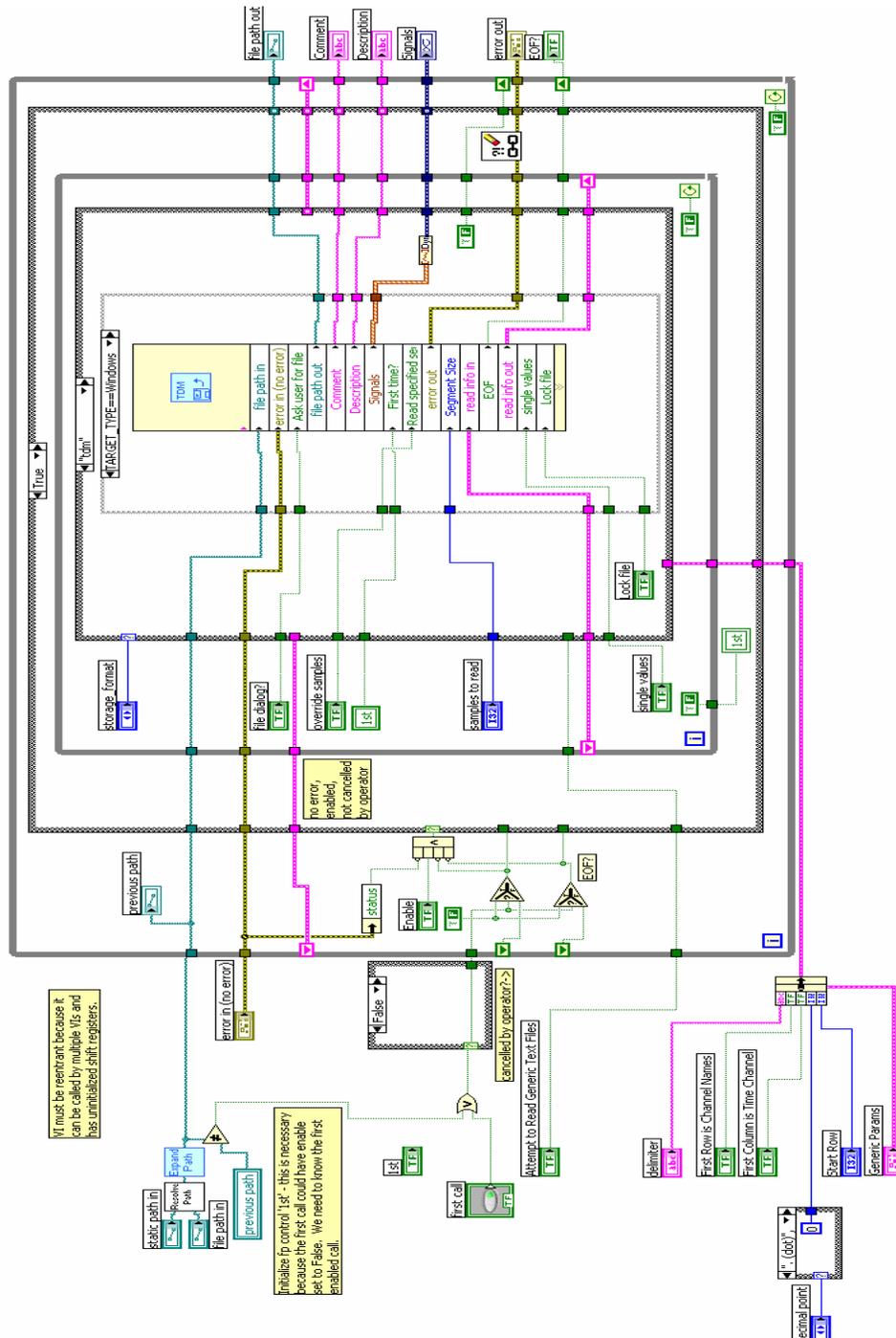


Figura 145. Programación interna Read Measurement file

Se decidió por activar la bandera agregando una entrada al Vi Express. En la programación principal del generador multivoltaje, después de leer por primera vez, se maneja el error para eliminarlo y después con una variable local activar el botón de first call como se muestra en la siguiente grafica.

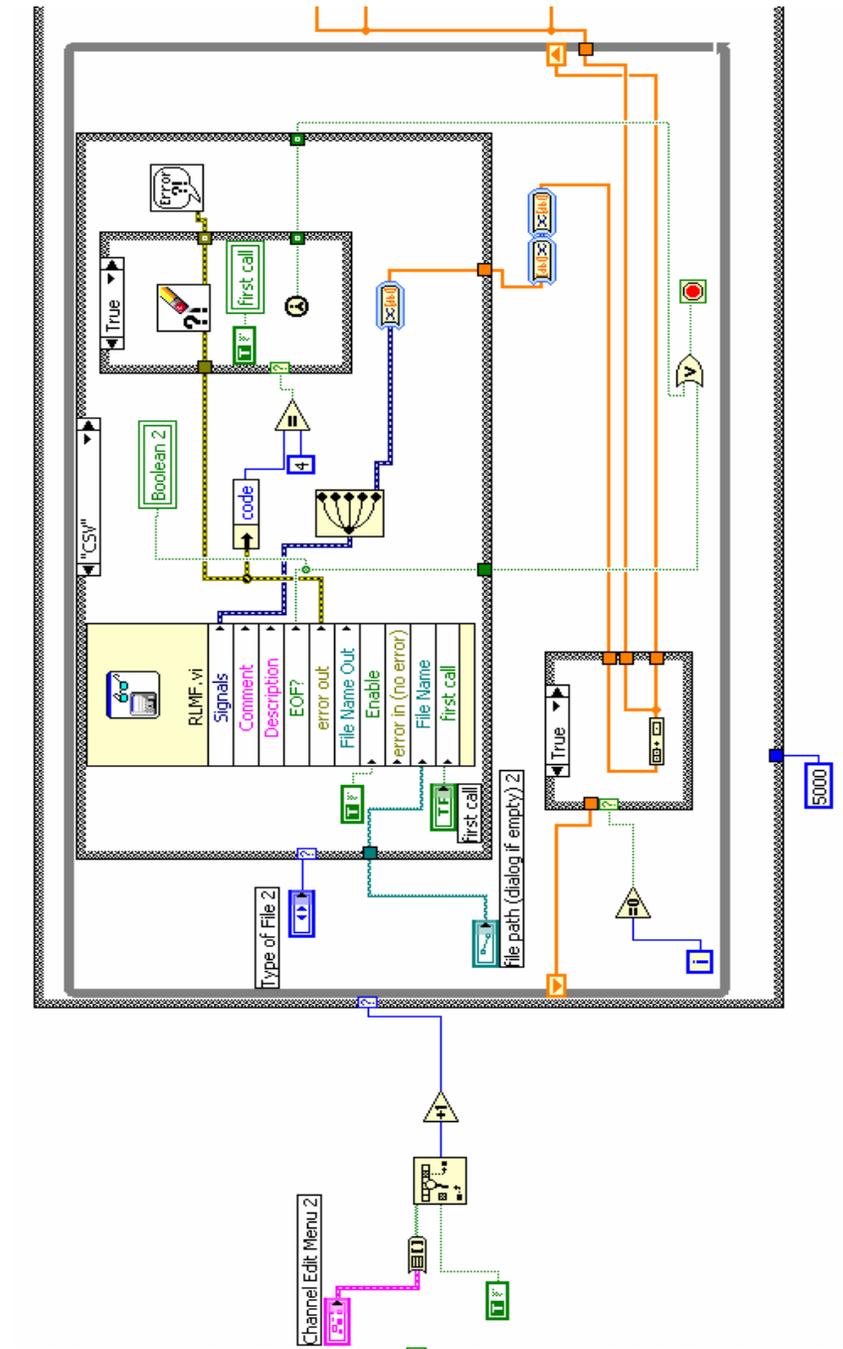


Figura 146. Modificación para leer archivos CVS.

Así también como se observa en la figura, esta es solo la programación para leer un archivo cvs. Para los archivos LVM, se utilizo un subvi, el cual internamente esta compuesto por un read from spreadsheet file.

Este subvi lo que hace es determinar el tamaño del encabezado del archivo lvm, y leer a partir de allí los datos.

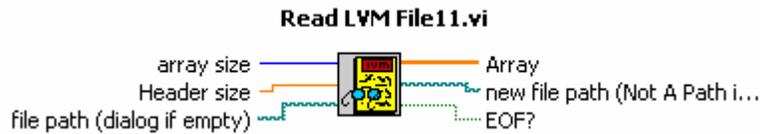


Figura 147. SubVI Read File11

Debido a que el resultado que se lee es una matriz de dos dimensiones, que contiene el la posición del dato y el dato como tal, y solo se quiere el dato, se extrae haciendo uso de un index array y un replace array. También se detecta cuando termina de leer el último dato.

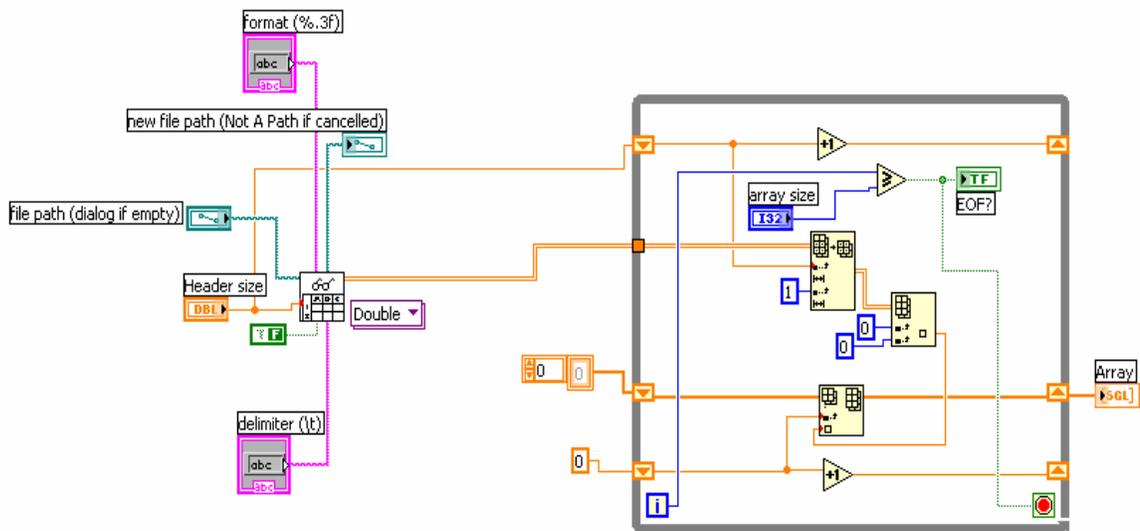


Figura 148. Programación interna SubVI Read file11

En el programa principal solo se configuran las entradas y se conectan las salidas correctamente como se muestra en la figura:

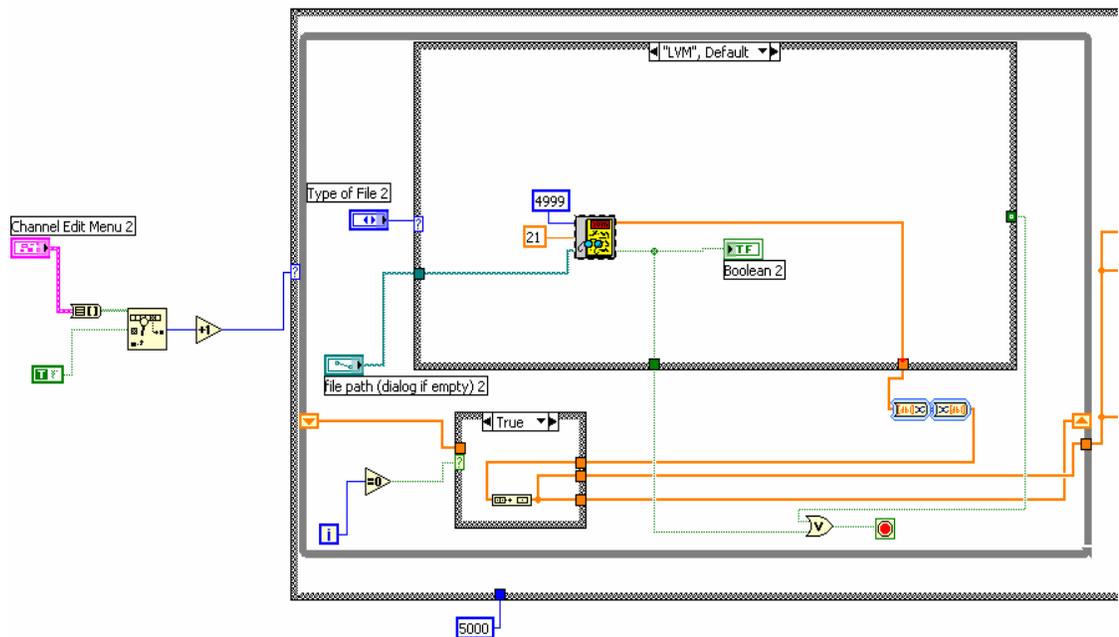


Figura 149. Nueva lectura de archivos LVM.

5.4 MEJORA EN EL BOTON SAVE SIGNAL

Recordemos que la función de Save signal se aplica de manera independiente para cada señal cuando es seleccionada por el usuario en la fase de configuración de señal. Recordemos también que aunque se encuentra dentro del TAP Control que ejecuta las funciones de la fase de configuración de señal, esta función no pertenece a esta fase y sólo se ejecuta en la fase de generación de señal.

Esta función permite salvar la señal que se esta generando, sin embargo al igual que en el caso anterior en el momento de salvar la grafica una ventana aparece preguntando el nombre del archivo. La ventana no solo permite guardar el archivo sino que también permite cancelar la acción al oprimir cancelar y al igual que antes al presionar cancelar se generaba un error pero ahora nombrado por Labview como error 1, mgArgErr un parámetro de entrada es invalido.

De la misma manera se corrigió el error aplicando la técnica mencionada anteriormente en este documento, es decir, manejando el error y comparando con el valor del error para después borrarlo. Sin embargo se agregó la posibilidad de preguntar por segunda vez por el nombre del archivo.

Para lograr esto se agregó una bandera que setea el Write Measurement file como se observa en la figura de abajo. Para este ejemplo la bandera se llama Flage 4.

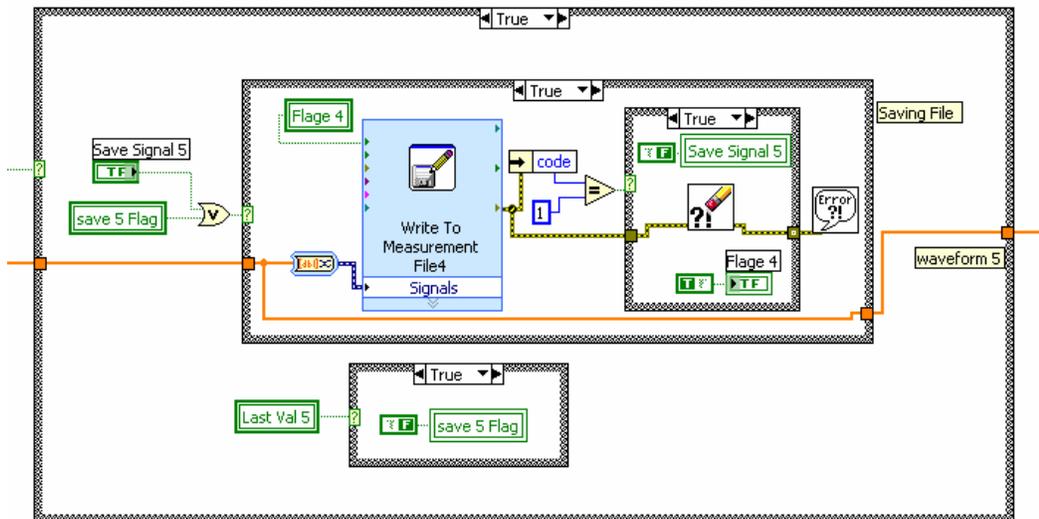


Figura 150. Corrección para el botón save file

5.5 SALVADO AUTOMÁTICO EN EL CREATE SEQUENCE

Create Sequence, es otra función que solo se ejecuta en la fase de generación de señal. Este subvi permite al programa desarrollar una secuencia mas larga abriendo dos archivos y además permite guardar la señal para generarla por el canal 1, sin embargo la programación inicial permitía guardar el archivo pero había que estar pendiente de la ejecución del programa para detener el salvado de este.

Para evitar inconvenientes entre diferenciar de salvar la señal al oprimir el botón de save file o salvar desde el subvi, se cambio a la salida de este ultimo las variables locales que ejecutaban la acción directa sobre el botón de salvar la señal por banderas que activan la opción de salvar como se muestra en la figura 3.

También hay que identificar cuando el programa ejecuta las ultimas cinco mil muestras para lo que se agregó una bandera a la que se llamó last Val, esta bandera se pone en verdadero cuando se generan las últimas cinco mil muestras y como se muestra en el gráfico anterior cancela el proceso de salvado, sin necesidad de que el usuario este pendiente de hacerlo.

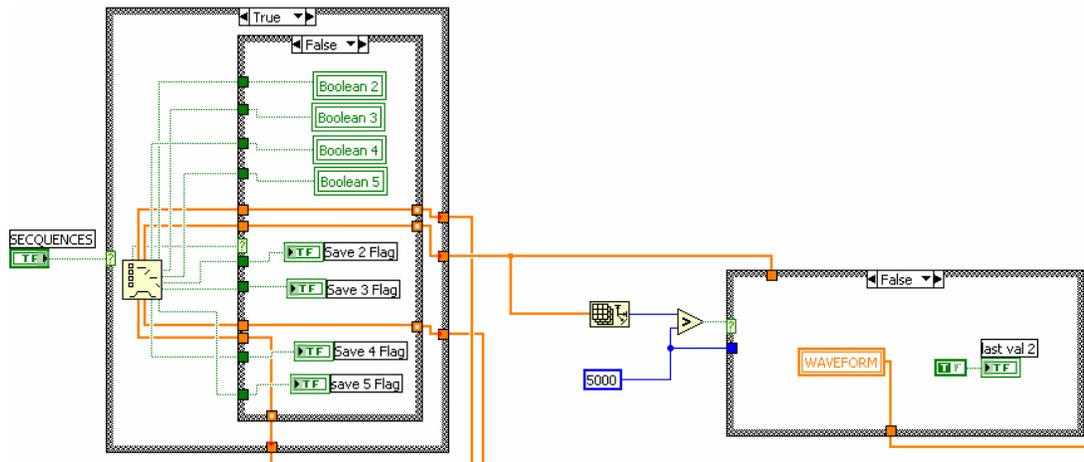


Figura 151. Identificación ultimas 5000 muestras de un archivo

5.6 MEJORAS EN EL RIPPLE CREATOR

Debido a que el ripple creator generaba solo una pequeña porción de señal con rizado, se mejoro la programación de este para que generara el rizado durante todo el tiempo. Sin embargo se puede obtener un resultado como el que se muestra en la señal 2, si se guarda el archivo, y se recarga.

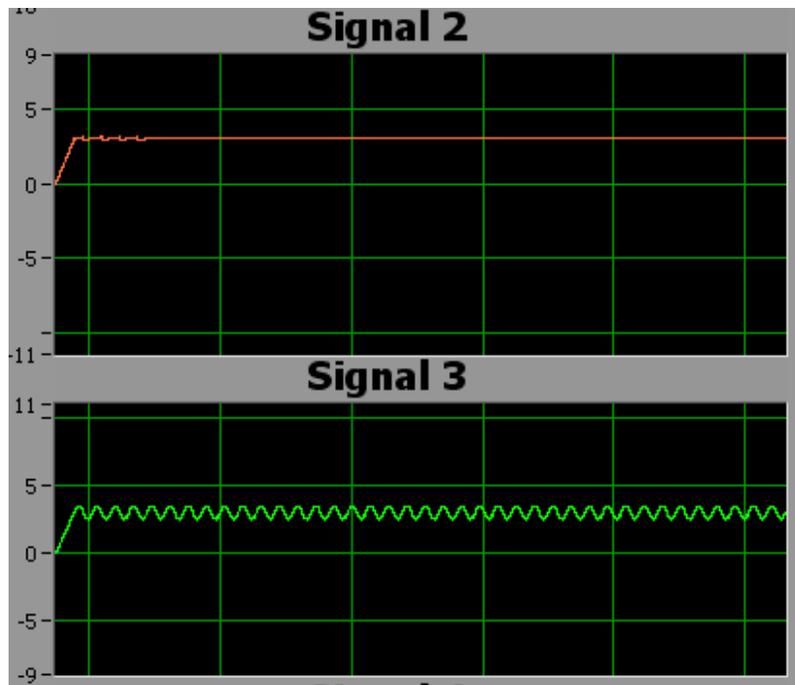


Figura 152. Ejemplo de rizados

Para lograr generar un rizado durante todo el tiempo, fue necesario cambiar la programación a la salida del Ripple creator SubVI. Para ello, primero se revisa que el tamaño de la señal no supere las 5000 muestras, luego se captura el array que contiene solo el rizado. Como el array del rizado no es un array de 5000 muestras, se agrega a este array las muestras que hacen falta para completar 5000. Hecho esto se agregan las muestras a las señal de contiene la rampa con el rizado.

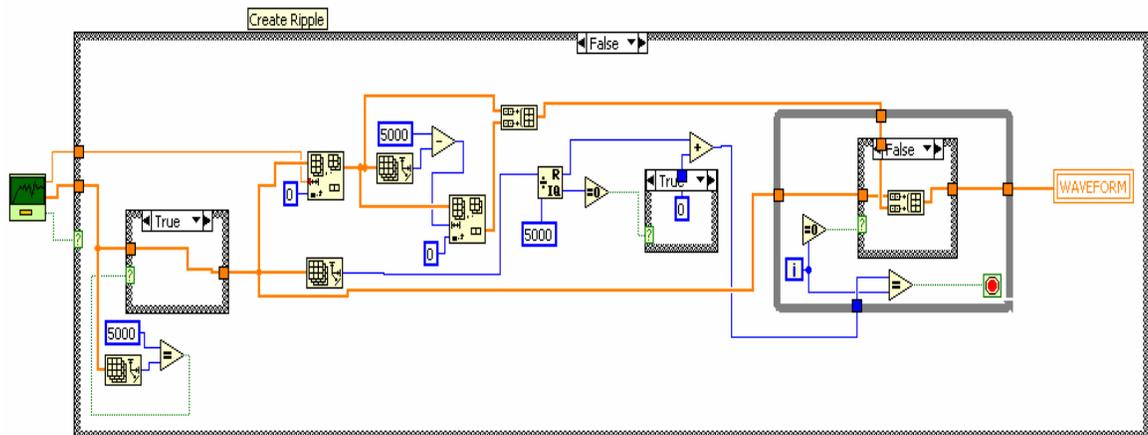


Figura 153. Programación mejorada de rizado

También como mejora a este SubVI, se eliminaron algunas funciones para el usuario, haciendo ahora más sencillo la creación de la rampa.

Las funciones de zoom, clean, save to file y Add Ripple siguen funcionando normalmente, aunque la función de crear rampa ahora es mas sencilla, solo se ajusta el nivel de voltaje de la señal, y la muestra hasta donde se quiere que la rampa se ejecute.

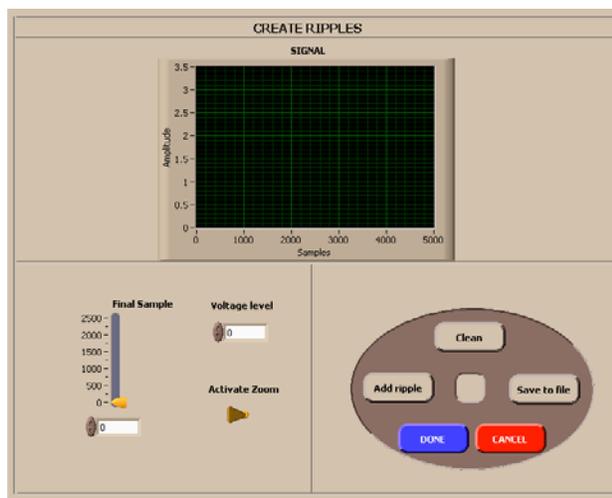


Figura 154. Panel de control SubVI creación de rizos

Para ello en la programación se corrigieron un par de cosas. Primero se ocultaron los controles que ya no son necesarios y segundo, se calcula la pendiente como el cociente entre el nivel de voltaje y la muestra final. También se revisa dentro del programa que la señal resultante no supere las 5000 muestras.

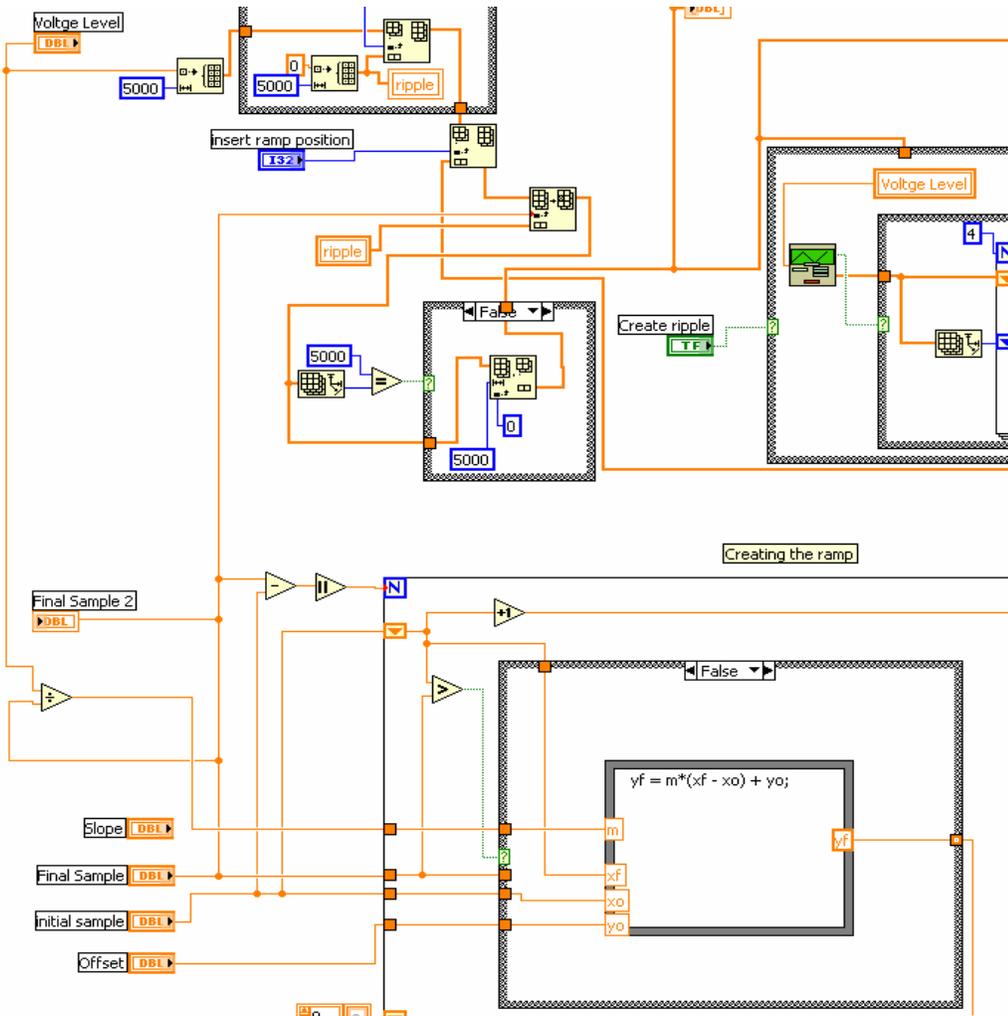


Figura 155. Modificación programación para la rampa en Ripple Creator

6. ACTUALIZACIÓN LABVIEW 7.1 A LABVIEW 8.2

En miras a mantener un programa actualizado y con mayores posibilidades en el futuro para desarrollar nuevas características, se actualizó la versión de Labview 7.1 a Labview 8.2.0

Las nuevas características que incluye Labview 8.2.0 son las siguientes:

- Mayores capacidades matemáticas y de análisis con el MathScrip – Lenguaje de programación textual generalmente compatible con la ampliamente usada sintaxis m-file.
- Fácil integración de código externo y servicios.
- Control de dibujos en 3D – Herramientas Open GL para rendimiento sofisticado tridimensional.
- Mejoramiento de la funcionalidad de almacenamiento de datos.
- Mejoramientos en el diagrama de bloques
- Directivas de programación orientadas a objetos para modular y reutilizar código
- Asistente de Importación para DLLs y bibliotecas compartidas
- Llamado dinámico de DLLs y soporte para llamadas definidas por el usuario
- Asistente de importación para Servicios Web de .NET
- Herramientas de escritura de datos a archivo TDM de alta velocidad
- Respaldo y recuperación automáticas de VIs
- Asistente de exportación de controladores de instrumentos
- Control simultáneo de múltiples usuarios de aplicaciones de LabVIEW a través de navegadores de Web

Además de las características de desarrollo de Labview, también permite nuevas mejoras en sus módulos Real time, FPGA, DSC y PDA.

Antes de actualizar Labview primero hay que tener en cuenta los requerimientos del sistema y si se esta actualizando Labview de una versión anterior. Los requerimientos del sistema para instalar Labview 8.2 son los siguientes:

Windows		
	Minimum	Recommended
Processor	Pentium III/Celeron 866 MHz or equivalent	Pentium 4/M or equivalent
RAM	256 MB	512 MB
Screen Resolution	1024 x 768 pixels	1024 x 768 pixels
Operating System	Windows Vista/XP/2000	Windows Vista/XP
Disk Space	1.2 GB	1.2 GB (Note -- Includes default drivers from NI Device Drivers CD)

Tabla 2. Requerimientos del sistema para instalar Labview 8.2 en Windows

Es recomendable primero desinstalar el programa de versión anterior, para ello basta con ejecutar el uninstall desde el panel de control de Windows XP en la opción de agregar o quitar programas.

Allí, seleccionar la opción de National Instruments Software y dar click en quitar. Una nueva ventana de National Instruments se abre mostrando todos los programas de National Instruments instalados en el PC, allí se busca Labview 7.1, posteriormente click en desinstalar.

Al finalizar es recomendable reiniciar el PC para que los cambios hechos surtan efectos. Luego instalar Labview 8.2 siguiendo los pasos que allí se presentan.

Finalmente reinstalar los drivers de las tarjetas o dispositivos externos que se estén utilizando que para el Generador Multivoltage son las tarjetas PCI 6733 y PCI 6443.

Por cuestiones de seguridad también es recomendable realizar una copia del programa que se estaba editando en Labview 7.1 en caso de que la actualización dañe o cambie en algo el comportamiento del VI.

Al iniciar Labview 8.2 y abrir el programa del Generador multivoltaje en esta versión, Labview automáticamente realiza los cambios necesarios para que el VI no sufra ninguna alteración en cuanto a su comportamiento se refiere, es decir para que las funciones para las que fue programado no se dañen.

Por ejemplo en el caso de los botones reload y save en labview 7.1 La programación tenía la siguiente apariencia:

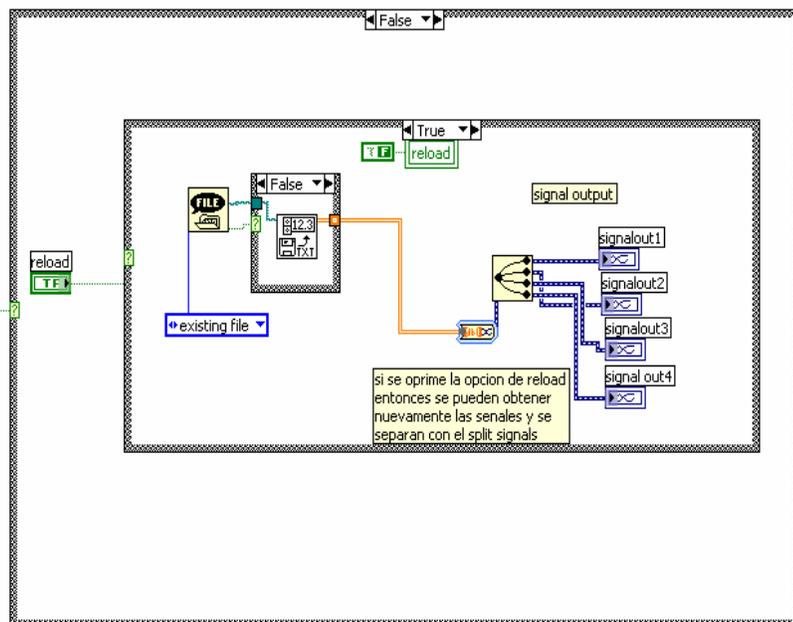


Figura 156. File Dialog en Labview 7.1

En Labview 8.2:

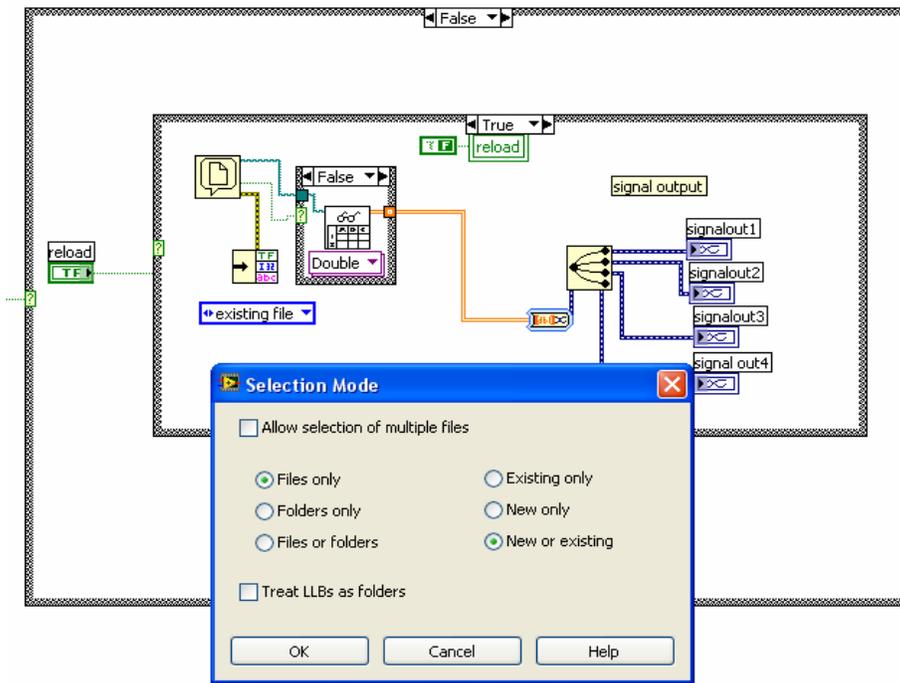


Figura 157. File dialog en Labview 8.2

Como se puede observar los VI utilizados han cambiado, por ejemplo el read from Spreadsheet file posee una pestaña abajo donde se puede seleccionar el tipo de archivos que se quieren leer.

Otra función que también cambio fue el Open File que ahora en Labview 8.2 corresponde a un VI Express.

El VI que se muestra en la figura anterior es también un open file, sin embargo es un VI que Labview colocó de manera automática a la hora de abrir el programa del Generador Multivoltaje por primera vez después de actualizar.

7. PRUEBAS CON CARGA EN EL GENERADOR MULTIVOLTAJE

7.1 PRUEBAS CON RESISTENCIAS

Function	Nominal Voltage [V]	Max Power [mW]	Current [mA]	Max allowable IR- Drop 2,5%	R[Ω]	≈
CADC+STBY-IO	3.3	79.2	24	83	137.5	140
FLASH - STDBY	3.3	50	15.2	83	217.1052632	220
STDBY - Core	1	50	50	25	20	20
Core	1	1450	1450	25	0.689665172	1/2W
IO - PADS	3.2	241	73	83	45.186	45
IO-PADS_VIDEO	2.5	84	33.6	63	74.4047619	75
AFE HD	1.2	276	230	30	5.217391304	5
AFE SD	1.2	228	190	30	6.315789474	6
AFE HD + TL/2	3.3	136	41.2	83	80.09708738	80
AFE SD + TL/2	3.3	123	37.3	83	88.47184987	90
AUD	3.3	100	30.3	83	108.9108911	110
SIF	1.2	10	8.3	30	144.5783133	144
	1.2	55	45.8	30	26.20087336	26
SPDIF	3.3	13.2	4	83	825	825
VDAC	3.3	100	30.3	83	108.9108911	110
DDR	1.8	525	291.7	45	6.170723346	6
LVDS A	2.5	257	102.8	63	24.31906615	24
LVDS B	2.5	222	88.8	63	28.15315315	30
LVDS A PLL	2.5	35	14	63	178.5714286	180
HDMI A	3.3	60	18.2	83	181.3186813	180
	1	230	230	25	4.347826087	4
HDMI B	3.3	60	18.2	83	181.3186813	180
	1	230	230	25	4.347826087	4
HDMI PLL	1	20	20	25	50	50
USB	3.3	140	42.4	83	77.83018868	80
Main PLL	1	10	10	25	100	100
DDR PLL	1	10	10	25	100	100

Tabla 3. Puntos de prueba Board VCTPremiumD

Para comprobar que el Generador ejecute todas sus funciones de manera correcta, se desarrollaron algunas pruebas con carga. Inicialmente se quería ver que aplicar una señal desde el generador es seguro, que no exista saltos de voltaje, revisar que tanto se cae el voltaje para los posibles puntos de prueba mostrados en la tabla de arriba.

Donde para estos se cálculo el valor de resistencia que se debía colocar al generador.

Una vez realizadas las pruebas con los valores de resistencias que simulan la carga ejercida en los puntos de la board, se crearon las librerías usando para ello el programa de Start Up Creator del generador Multivoltaje.

Al probar con los resistores, se encontró que para aquellos puntos como el core, el voltaje se cae un poco mas por debajo del valor permitido.

7.2 PRUEBAS CON BOARD

Una vez hecho esto se procedió a ejecutar las pruebas en la board de pruebas VCTPremiumD Evaluation Board.

Allí se quería visualizar o revisar el cuadro de inicio de las señales al alimentar la board, es decir determinar que señal esta mas adelantada que otra y como es su comportamiento.

La board fue diseñada para evaluar el comportamiento y el funcionamiento de un nuevo chip de Micronas, que mejora funciones para los televisores.



Figura 158. Board VCTPremiumD

Para desarrollar las pruebas en la board fue necesario además de dos canales del generador Multivoltaje, un multímetro, Osciloscopio (Tektronix TDS784d), un cable serial, una fuente de alimentación.

Las primeras pruebas fueron realizadas para observar el comportamiento de las señales y clasificar las señales, es decir, para identificar que señales se inician con la fuente de poder.

Puntos de prueba Board VCTPremiumD Evaluation Board

D25_LVDS_PLL	1V8_VCTH	VDD33_AAU
VDD25_LVDSA	VDD10_DDRPLL	VDD12_AFE
VDD25_LVDSB	VDD10_MPLL	VDD33_AFE
VDD25_IO	VDD33_STDBY	VDD33_IO
VDD33_USB	VDD33_FLASH	VDD12_SIF
VDD10_HDMI	VDD10_STDBY	VDD10_Core
VDD33_HDMI	VDD33_VDAC	

Observando más detenidamente como es el comportamiento del board al energizarse se realizó la siguiente prueba en el osciloscopio, usando solo la fuente de poder sin el generador multivoltaje:

Ch1: (trig) = VDD10_HDMI
Ch2 = VDD25_LVDS_PLL
Ch3 = VDD25_LVDSA
Ch4 = VDD25_LVDS

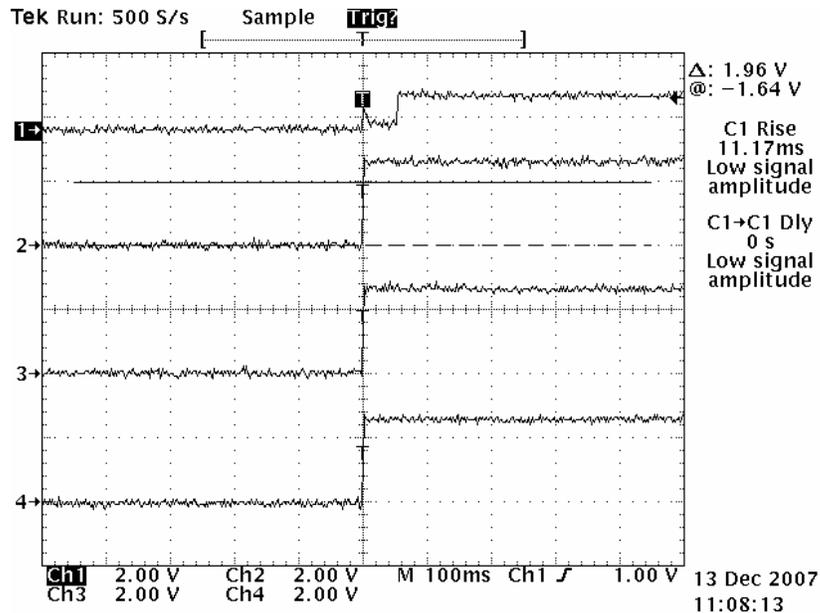


Figura 159. Resultados prueba 1 board

Una segunda prueba, en otros puntos y mejorando la escala de tiempo en el Osciloscopio muestra lo siguiente:

Ch1: (trig) = VDD10_HDMI
Ch2 = VDD33_FLASH
Ch3 = VDD12_AFE
Ch4 = VDD10_STDBY

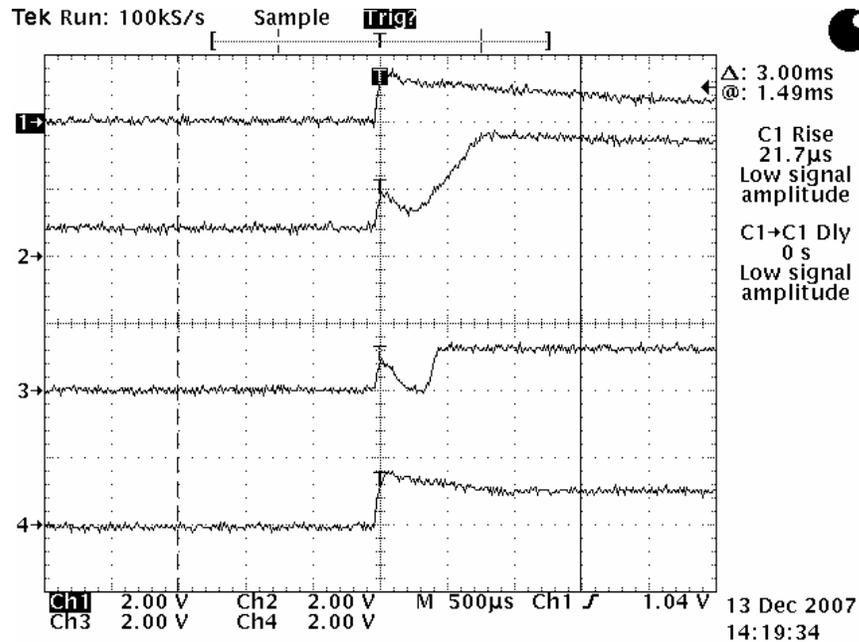


Figura 160. Resultados Prueba 2 Board

Para utilizar el generador Multivoltaje fueron necesarios realizar un par de cambios sobre la Board.

La primera modificación, consistió en remover una bobina para luego insertar dos pines en el VDD_CORE con el fin de alimentar este punto con el Generador Multivoltaje.

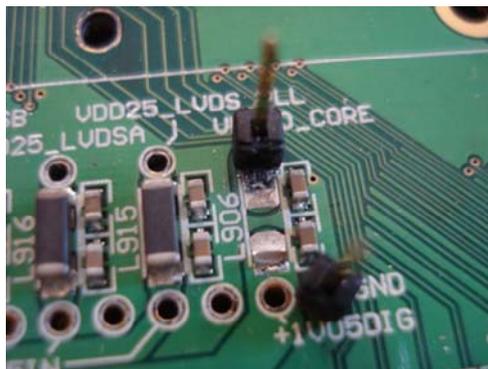


Figura 161. Modificación 1 en el board de prueba

La segunda modificación consistió en remover un dispositivo encargado de generar una señal de reset cuando la board es encendida, dando a conocer así un estatus conocido de inicialización. Allí se soldó una regletilla de 3 pines para insertar nuevamente el dispositivo, solo que el tercer pin fue conectado

directamente al segundo punto donde se insertaría la segunda señal del Generador Multivoltaje.

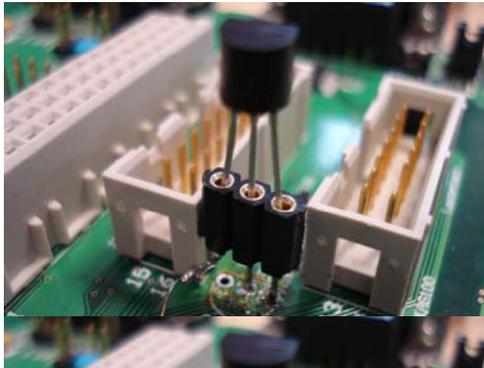


Figura 162. Modificación 2 en el board de pruebas.

Se utilizo el canal 2 del generador multivoltaje para insertar una señal por el punto VDD10_CORE. Este punto requiere de alrededor 1.5 amperios para alimentar el núcleo del chip.

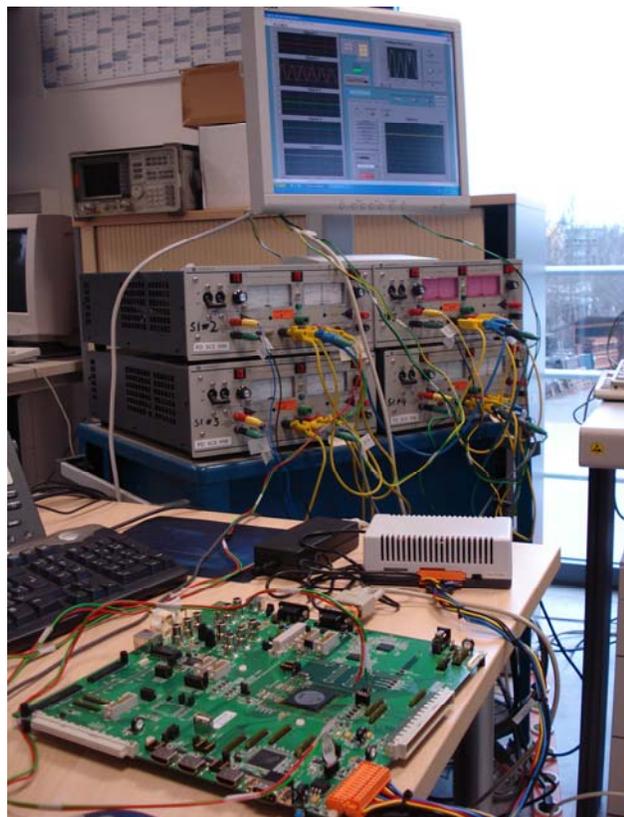


Figura 163. Generador Multivoltaje

La board una vez el core es energizado envía una palabra a un terminal usando el puerto serial, de la siguiente manera:

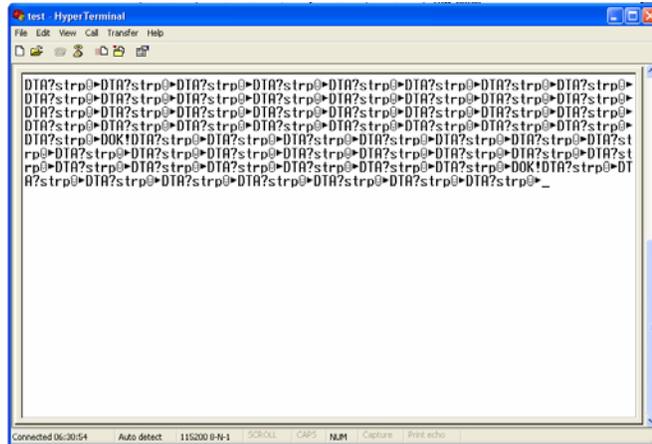


Figura 164. Ventana terminal de recepción de pulso

Cada vez que se energiza correctamente, aparece la palabra DTA?STRP.

Pero para apreciar el delay entre las señales que no se generan directamente desde la fuente fue necesario alimentar la board desde un segundo punto a 3.3 voltios.

Se realizaron las siguientes pruebas con el Generador Multivoltaje alimentando la board:

Ch1: (trig) = VDD10_CORE
Ch2 = VDD33_AFE

Ch3 = VDD12_SFE
Ch4 = VDD33_AAU

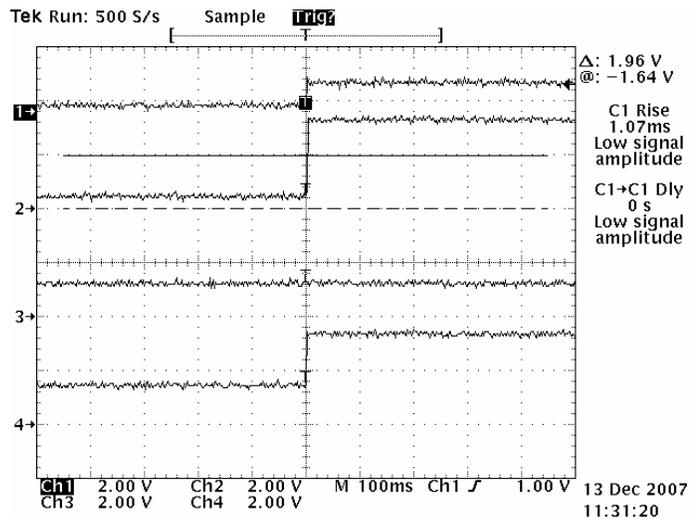


Figura 165. Prueba 3 en el Board

Mejorando la escala del tiempo:

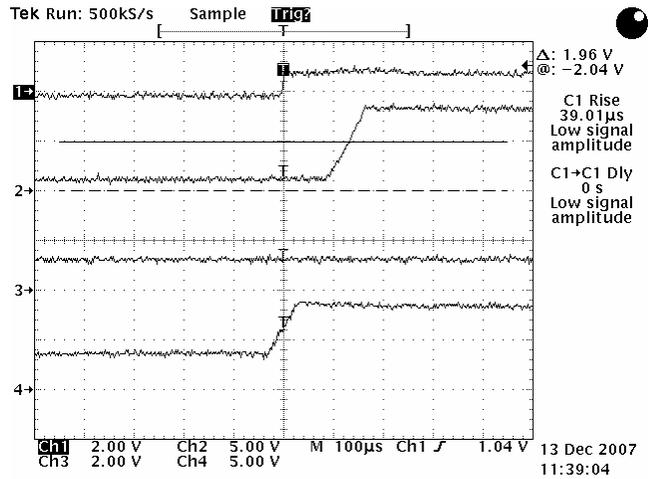


Figura 166. Prueba 3 en el board escala de tiempo ampliada

Una segunda prueba fue realizada

Ch1: (trig) = VDD10_CORE
Ch2 = VDD33_FLASH
Ch3 = VDD10_STDBY
Ch4 = VDD33_VDAC

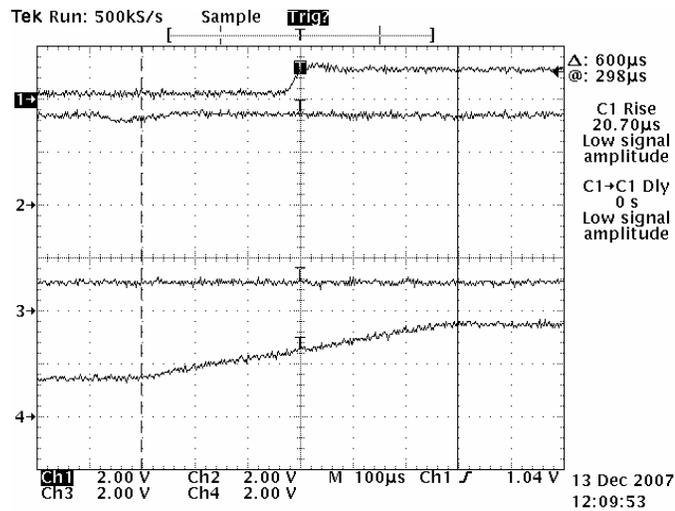


Figura 167. Prueba 4 en el board

Como se muestra en las graficas anteriores, se capturan las señales pero no se puede determinar bien en que momento del tiempo son generadas, debido a que la señal generada en el core no posee ningún desplazamiento y es solo un Step o una rampa brusca de 200uS.

Para determinar de mejor manera el funcionamiento o como se están generando las señales, para ello se realizó otra prueba de la siguiente manera:

Ch1: (trig) = VDD10_CORE
 Ch2 = VDD33_STDBY
 Ch3 = VDD33_AFE
 Ch4 = VDD33_AAU

Ahora la señal de VDD10_CORE posee un delay de 2mS

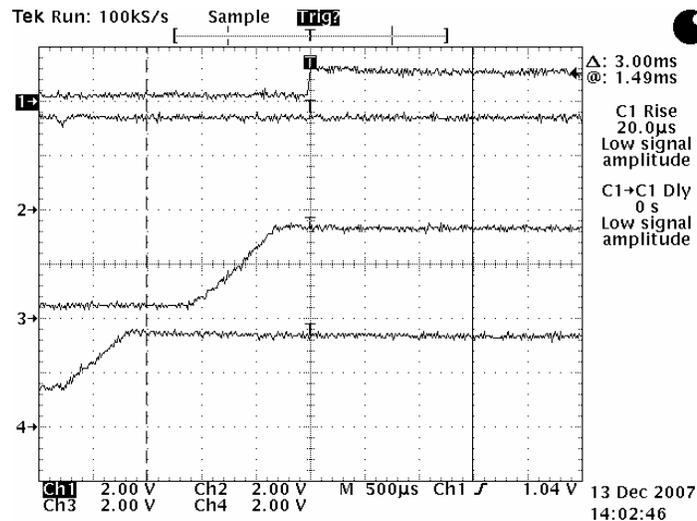


Figura 168. Prueba 5 en el Board.

Como se observa, las señales VDD33_AFE y VDD33_AAU, se energizan con anterioridad, que el core.

Se encontró que las siguientes señales se inicializan directamente con la fuente, estas señales son:

VDD25_LVDS_PLL	VDD10_MPLL
VDD25_LVDSA	VDD12_SIF
VDD25_LVDSB	VDD12_AFE
VDD25_IO	VDD10_STDBY
1V8_VCTH	VDD33_FLASH
VDD10_DDRPLL	

Las siguientes señales se visualizan con el core:

VDD33_USB	VDD33_VDAC
VDD33_HDMI	VDD33_IO
VDD33_AFE	VDD33_AAU

8. HARDWARE

8.1 AMPLIFICADOR KEPCO

Si las señales a generarse van a ser utilizadas para realizar test de chips o boards, entonces es recomendable utilizar el amplificador KEPCO o también llamado BOP (Bipolar Operacional Power Amplifier) como circuito limitante de corriente y voltaje. El amplificador BOP puede ser usado en una gran variedad de aplicaciones, configurado en los siguientes modos:

- Canal de control de voltaje.
- Canal de control de corriente.
- (+/-) Circuito limitante de Voltaje.
- (+/-) Circuito limitante de corriente.

El amplificador KEPCO es un amplificador bipolar. Tiene dos canales de control bipolar (modo voltaje o modo corriente), seleccionables de manera individual o por señales remotas.

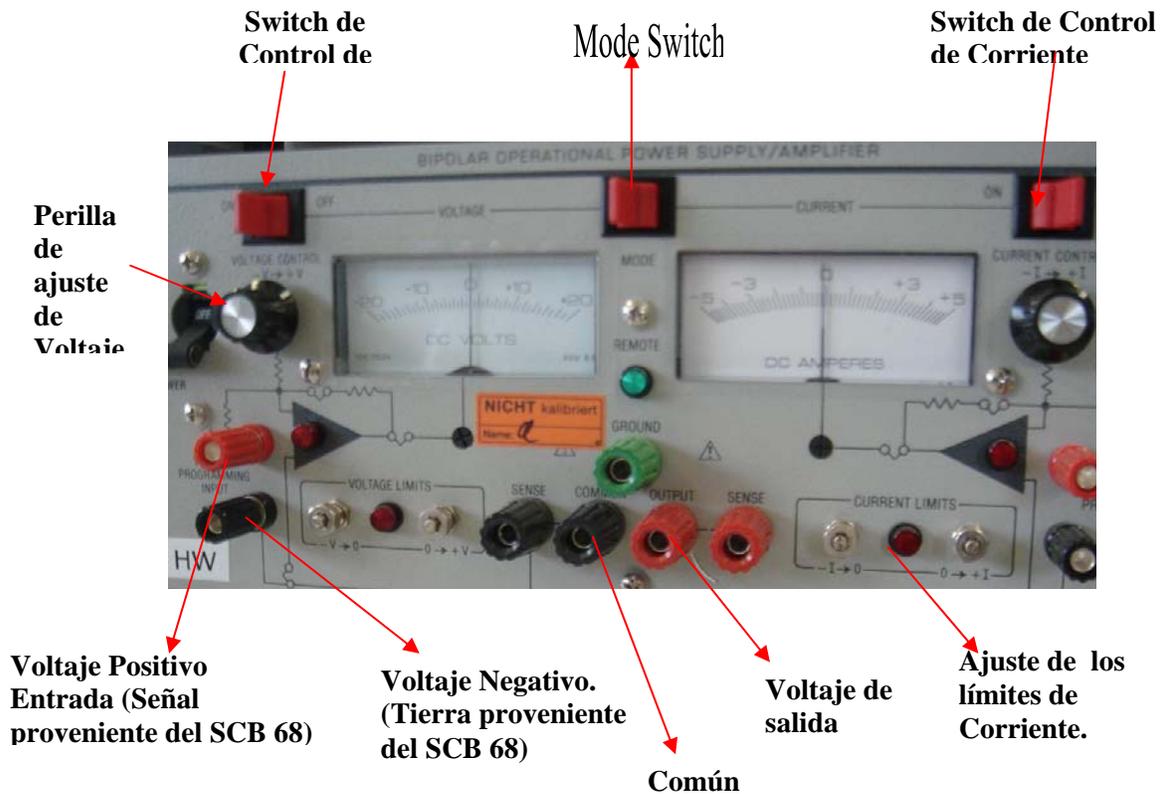


Figura 169. Amplificador KEPCO Figura 8.0.

El amplificador posee un conector trasero llamado PC 12 sin el cual el amplificador no puede funcionar. El conector PC 12 debe ser conectado en la ranura trasera del amplificador.



Figura 170. Conector trasero PC 12

8.2 SCB – 68 (Shielded Connector Blocks)

Para conectar las tarjetas de adquisición y de generación de datos al mundo exterior es necesario de un bloque conector. Esto se realiza utilizando cables especiales desde el Bloque conector SCB a las conexiones que pueden ser realizadas a otros dispositivos.

Hay dos tipos de SCBs basados en el número de pines que contienen y son:

- SCB 68
- SCB 100

El SCB es una board acorazada de 68 terminales de fácil conexión para los productos de National Instruments de 68 pines. Contiene en su interior una board para agregar resistencias, capacitancias, filtros (RC), un atenuador, sensores de corriente de 4-20mA así como también el uso de termocuplas, para lo cual también viene equipado con un sensor de temperatura de junta fría para usarse con las termocuplas.

Como las tarjetas PCI 6733 y PCI 6143 son dispositivos de 68 pines es apropiado el uso del SCB-68.

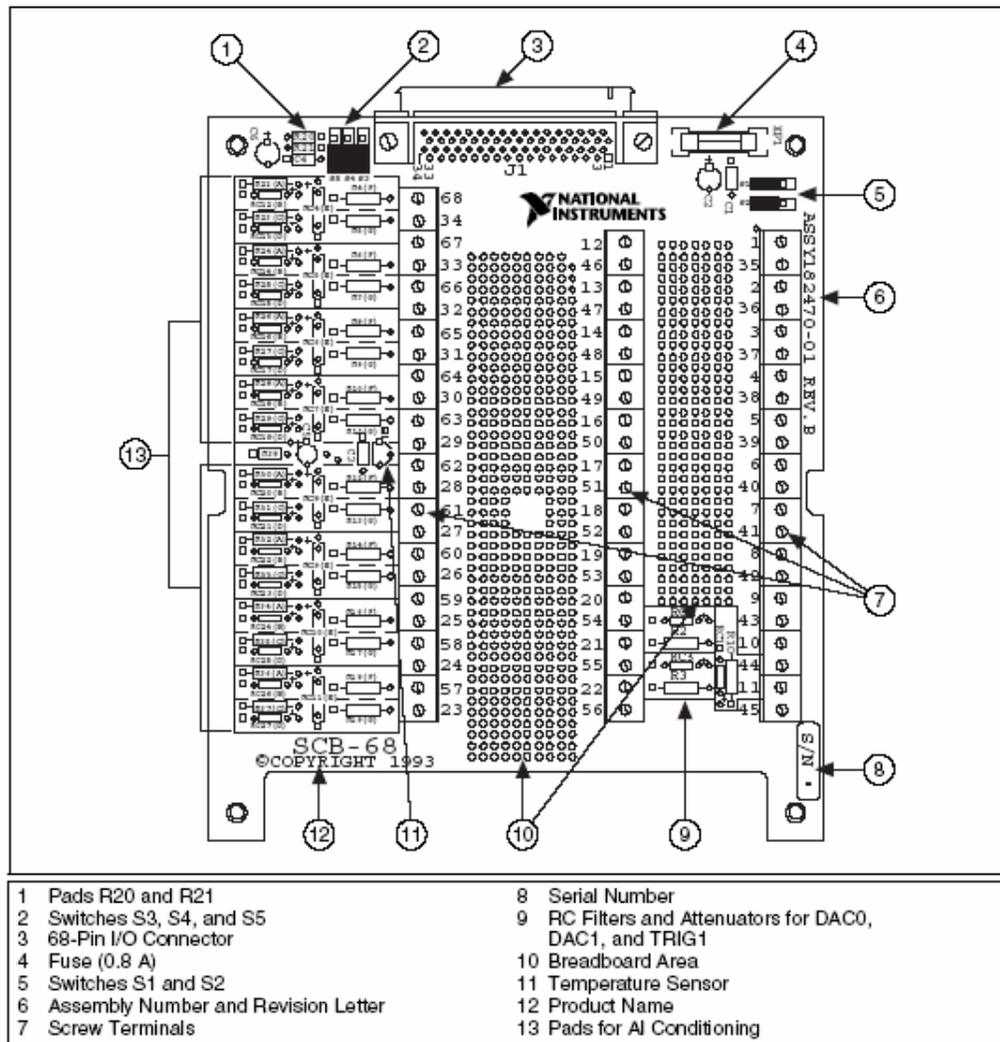


Figura 171. Board Interno SCB-68

El bloque conector SCB-68 posee cinco switches que deben ser apropiadamente configurados para funcionar correctamente con las tarjetas.

Cuando se usa una PCI 6733 los switches deben ser configurados como se muestra en la figura de abajo.

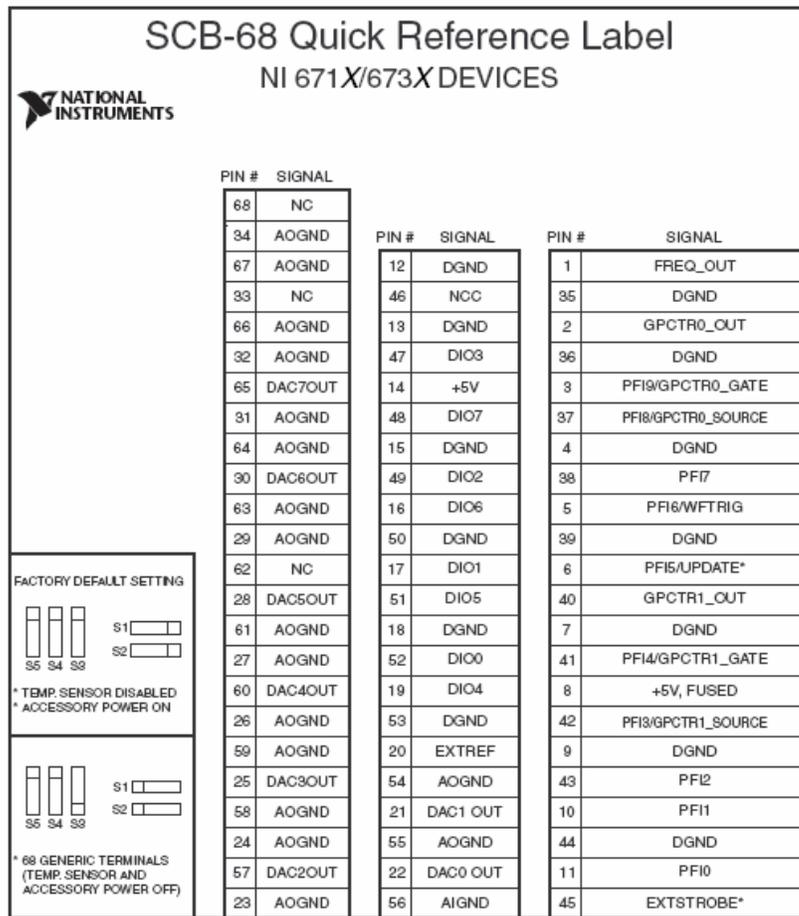


Figura 172. Diagrama de pines SCB – 68

Cuando se ejecuta el generador multivoltaje las señales de salida son generadas a través de los siguientes terminales, para la Tarjeta PCI 6733.

Signal Number in VI	DAQ Device Channel Used	Pin Number in SCB 68
Signal 1	DAC0 OUT	22
Signal 2	DAC1 OUT	21
Signal 3	DAC2 OUT	57
Signal 4	DAC3 OUT	25
Signal 5	DAC4 OUT	60
	AOGND	54,34,67,66,32.....Many Pins available for AOGND.

Tabla 4. Conexiones SCB – 68 con la tarjeta PCI 6733

Para el uso de la Tarjeta de adquisición de datos PCI 6143, solo un canal de voltaje es usado para este propósito y el VI ha sido programado para usar como entrada el canal AI 0 de la tarjeta de adquisición. Esta entrada corresponde al pin 68 del SCB-68 mientras que el pin 34 debe ser conectado a tierra.

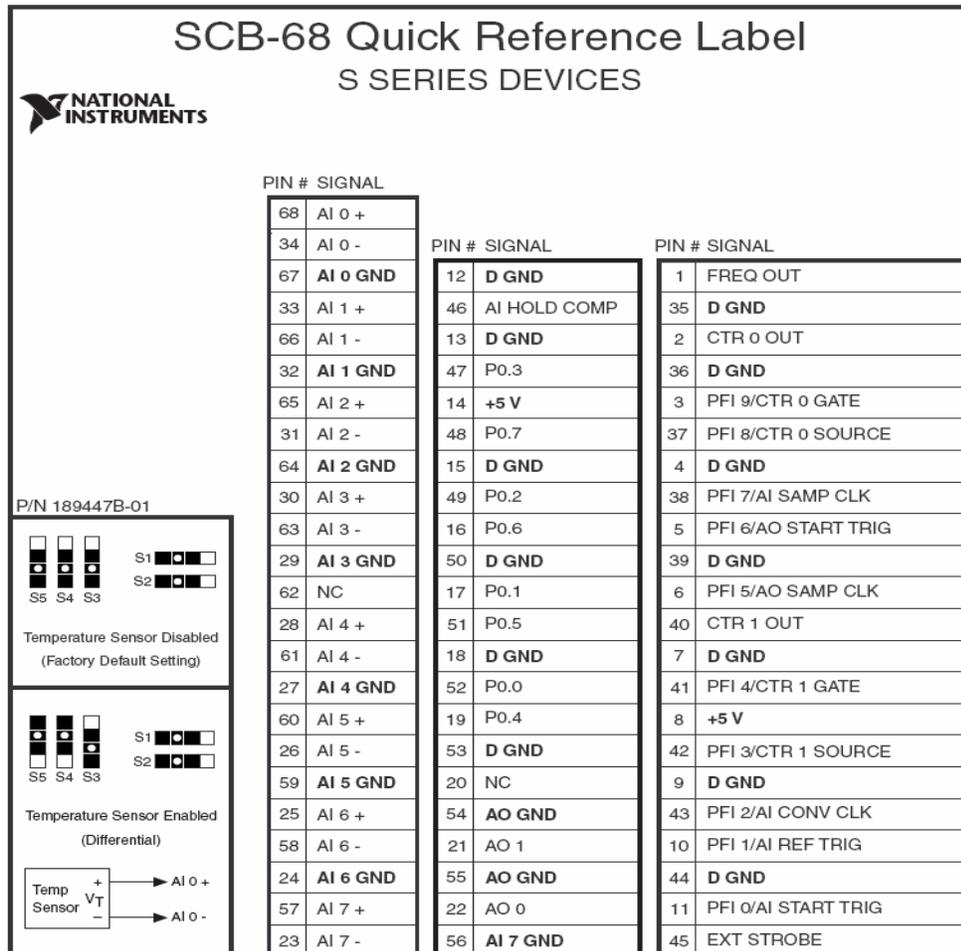


Figura 173. Diagrama de pines SCB – 68 configurado para recibir datos

8.3 PCI 6733

La PCI 6733 es una tarjeta generadora de señales de National Instruments. Posee las siguientes características:

- Bajo costo y generación arbitraria de formas de onda.
- Alta densidad de canal.
- Sincronización integrada del bus multielementos
- Easy real-time control with Labview Real-Time
- Trigger digital y reloj externo.

- Actualización simultanea.
- 8 digital I/O lines (TTL/CMOS)
- Two 24-bit counter/timers
- Servicios de medición que simplifican la configuración y mediciones.

Puede ser utilizada en sistemas operativos como Windows 2000/NT/XP, Mac OS X y Linux. Es recomendable desarrollar sus aplicaciones utilizando LabVIEW 7.x o versiones mas nuevas, LabWindows/CVI 7.x o recientes, Measurement Studio 7.x o recientes.

Esta tarjeta tiene la capacidad para tomar muestras a 1MS, dando la posibilidad de generar formas de ondas hasta por encima de los 500KHz. Cuando se utiliza este dispositivo se tiene control de cada punto de datos generado en la salida para cada canal. Esta característica es importante porque se puede definir no solo las formas de onda más comunes como cuadrada, seno, o diente de sierra sino que también formas de ondas más complejas.



Figura 174. Tarjeta Generadora.

8.4 PCI 6143

La PCI 6143 es una tarjeta de adquisición de datos de Nacional Instruments y posee características como:

- 8 entradas análogas diferenciales de 16-bits
- 250 kS/s de entrada análoga por canal.
- Muestreo Simultáneo
- 8 Líneas de Entrada salida digital (5 V TTL/CMOS).
two 24-bit counter/timers
- Digital triggering

- Rango de entrada análoga ± 5 V

La tarjeta de adquisición de Nacional Instruments PCI6143 combina lo último en tecnologías de computadores para desarrollar un muestreo simultáneo para aplicaciones con un alto conteo en el canal a un bajo costo. Estos elementos son usados en una gran variedad de aplicaciones, incluyendo:

- a. Física
- b. Pruebas de ultrasonido y sonar
- c. Balística.

Esta tarjeta es ideal también para aplicaciones que requieran una alta velocidad de adquisición de datos, sincronización entre las entradas análogas, salidas análogas, entradas y salidas digitales, operaciones de conteo, entre otras. Estas boards tienen también gran aplicación cuando se requiere más de un canal y mayor procesamiento por canal que una arquitectura multiplexada puede proveer.

CONCLUSIONES

- El generador Multivoltaje es un dispositivo cuya finalidad es generar y adquirir señales, con el propósito de estudiar las respuestas de los circuitos integrados a diferentes tipos de señales aplicadas, para ello puede generar cinco señales diferentes de 5000 muestras por segundo cada una.
- Los SCB – 68 son dispositivos conectores de 68 pines que actúan como interfases entre las tarjetas de adquisición y generación de datos y otros dispositivos.
- Los amplificadores KEPCO, también conocidos como amplificadores bipolares actúan como un circuito limitante de voltaje y de corriente. Están configurados de manera que funcionan como un amplificador inversor de ganancia 0.5, lo cual se debe tener en cuenta a la hora de programar.
- La programación del Generador Multivoltaje esta conformada en cinco fases principales, la fase de inicialización donde se encuentran las condiciones iniciales del sistema, la fase de configuración de señal donde se crean y modifican las señales, la fase de generación de señal que como su nombre lo indica se generan las señales, la fase de adquisición de datos y la fase final donde se devuelven las condiciones iniciales al sistema.
- En la fase de configuración de señal se crean y modifican las cinco señales del generador multivoltaje y todas las funciones de esta fase exceptuando la del canal uno son iguales, esto porque el canal uno es utilizado para cargar leer y generar señales desde la generación durante todo el tiempo.
- Se introdujeron nuevas funciones al Generador Multivoltaje como la posibilidad de guardar la configuración de las señales y recargarlas, así como también dos diferentes tipos de paradas con la finalidad de realizar diferentes tipos de pruebas o realizar la misma prueba muchas veces sin tener que recargar nuevamente o detener el sistema.
- Se introdujo también un botón de presentación con la finalidad de mostrar a los usuarios las principales funciones del generador Multivoltaje y sus controles.
- Se logró generar un reporte dentro del cual se pueden observar las graficas de las señales a generar y la grafica de la señal adquirida.

- Se logró manipular los datos de un archivo LVM, cambiando sus valores directamente desde un VI de Labview o desde una hoja de cálculo en Excel, así como también agregar rampas, offsets y mas atributos al archivo y la generación de archivos consecutivos con el propósito de crear librerías de manera mas rápida y sencilla.
- Se agregaron dos nuevas funciones al Generador Multivoltaje, la primera permite generar rampas con un pequeño rizo y la segunda función permite cargar dos archivos LVM para generar una secuencia completa de con la combinación de dos señales.
- Se encontró que la principal razón por la que el VI Express Read From Measurement File no permite leer el mismo archivo consecutivamente, es debido a que internamente posee una bandera llamada first call, la cual activa solo cuando se abre un archivo por primera vez.
- Se encontró también que el uso de los SubVI debe ser apropiado, no se deben utilizar SubVIs en rutinas de programación que se ejecutan todo el tiempo o en las que los datos dependen seriamente de las variables que se manejan en este tipo de rutinas, además puede resultar en programación redundante.

BIBLIOGRAFIA

Multivoltage Generator, Installation guide and user manual. Micronas GmbH.

Getting Started with Labview. National Instruments Corporation. April 2003.
Disponibile en: <http://www.ni.com/pdf/manuals/323427a.pdf>

Labview measurement Manual. National Instruments Corporation. July 2000.
Disponibile en: <http://www.ni.com/pdf/manuals/322661a.pdf>

Instruction Manual BOP 20-5M. KEPCO INC

Labview Data Adquisition Basic Manual- National Instruments Corporation.
January 2000. Disponibile en: <http://www.ni.com/pdf/manuals/320997e.pdf>

NI-DAQ User Manual for PC compatibles. National Instruments Corporation.
January 2000. Disponibile en: http://cires.colorado.edu/jimenez-group/QAMSResources/Docs/NIDAQ_User.pdf

Labview Installation guide Manual. National Instruments corporation

Labview Upgrade Notes. National Instruments Corporation. August 2006.
Disponibile en: <http://www.ni.com/pdf/manuals/371780b.pdf>

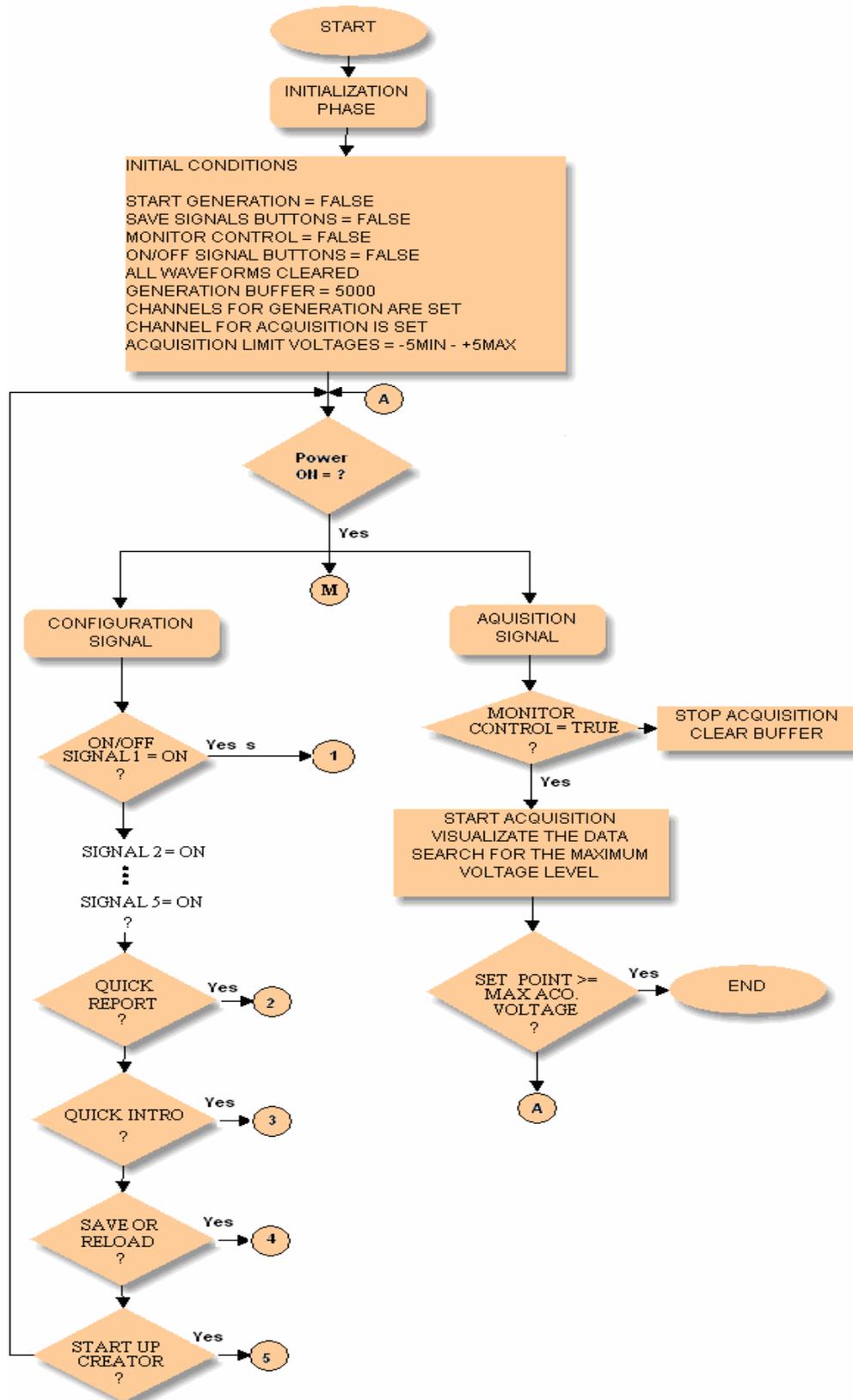
NI 6143 Specifications. National Instruments Corporation. February 2004.
Disponibile en: <http://www.ni.com/pdf/manuals/370835a.pdf>

NI 6731/6733 Specifications. National Instruments Corporation. June 2007.
Disponibile en: <http://www.ni.com/pdf/manuals/371232b.pdf>

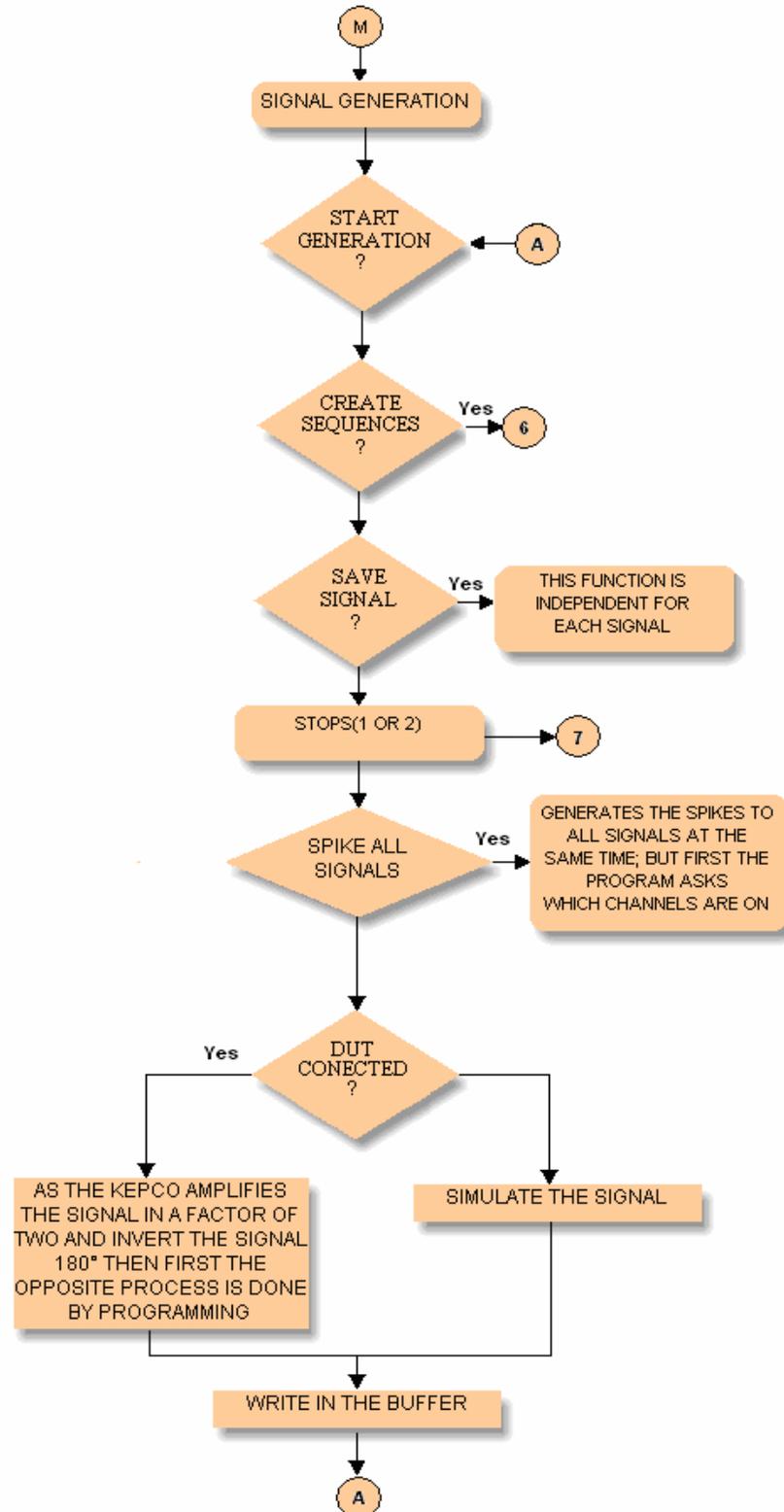
SCB -68 68-Pin Shielded Connector Block Installation Guide. National Instruments Corporation. January 1994.
Disponibile en: <http://www.ni.com/pdf/manuals/320745.pdf>

ANEXOS

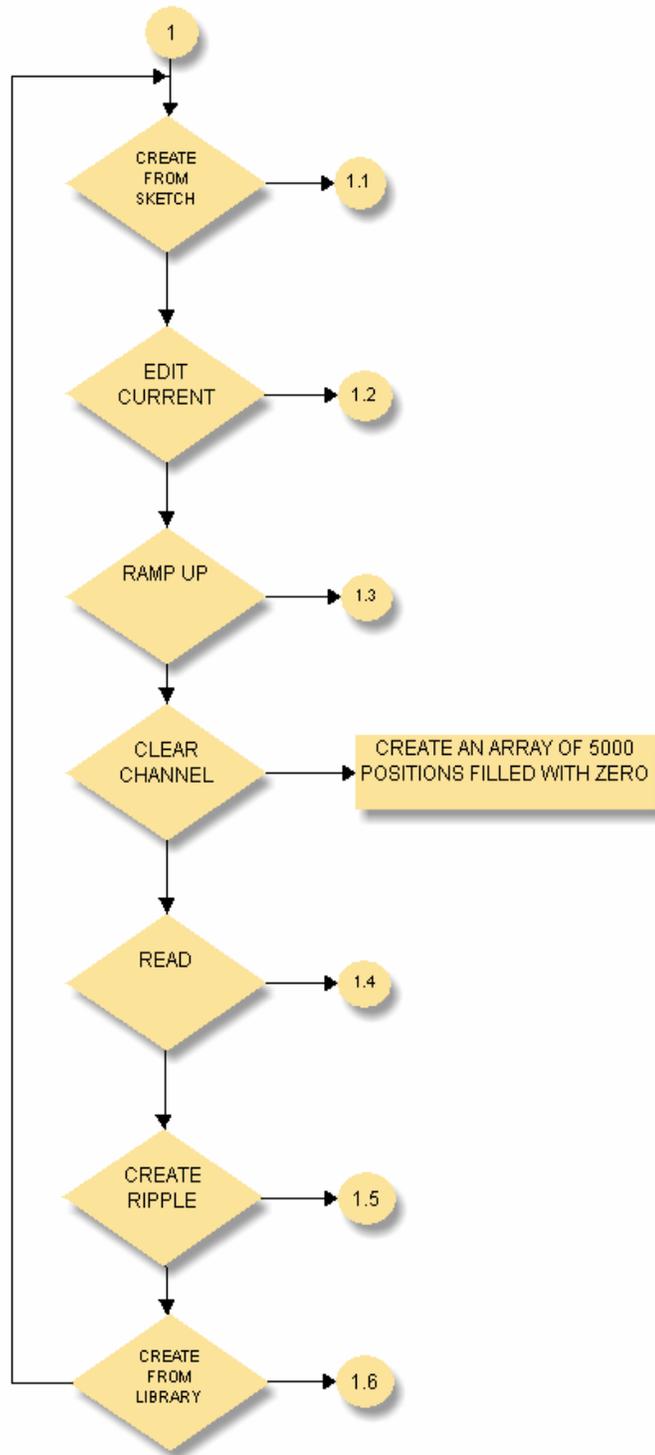
ANEXO 1: DIAGRAMA DE FLUJO GENERAL DEL GENERADOR MULTIVOLTAJE



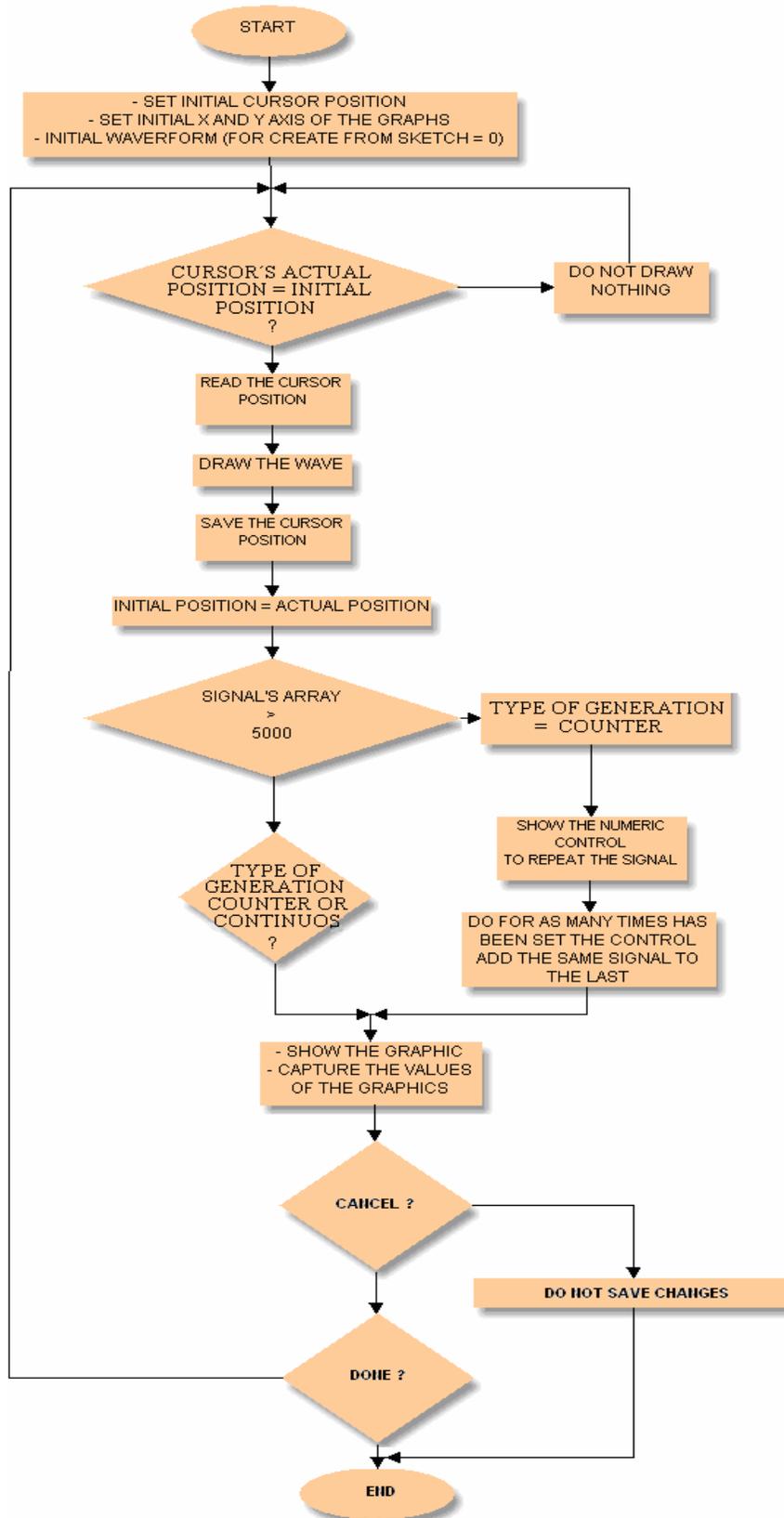
ANEXO 2: CONTINUACION DIAGRAMA DE FLUJO GENERAL DEL GENERADOR MULTIVOLTAJE (M)



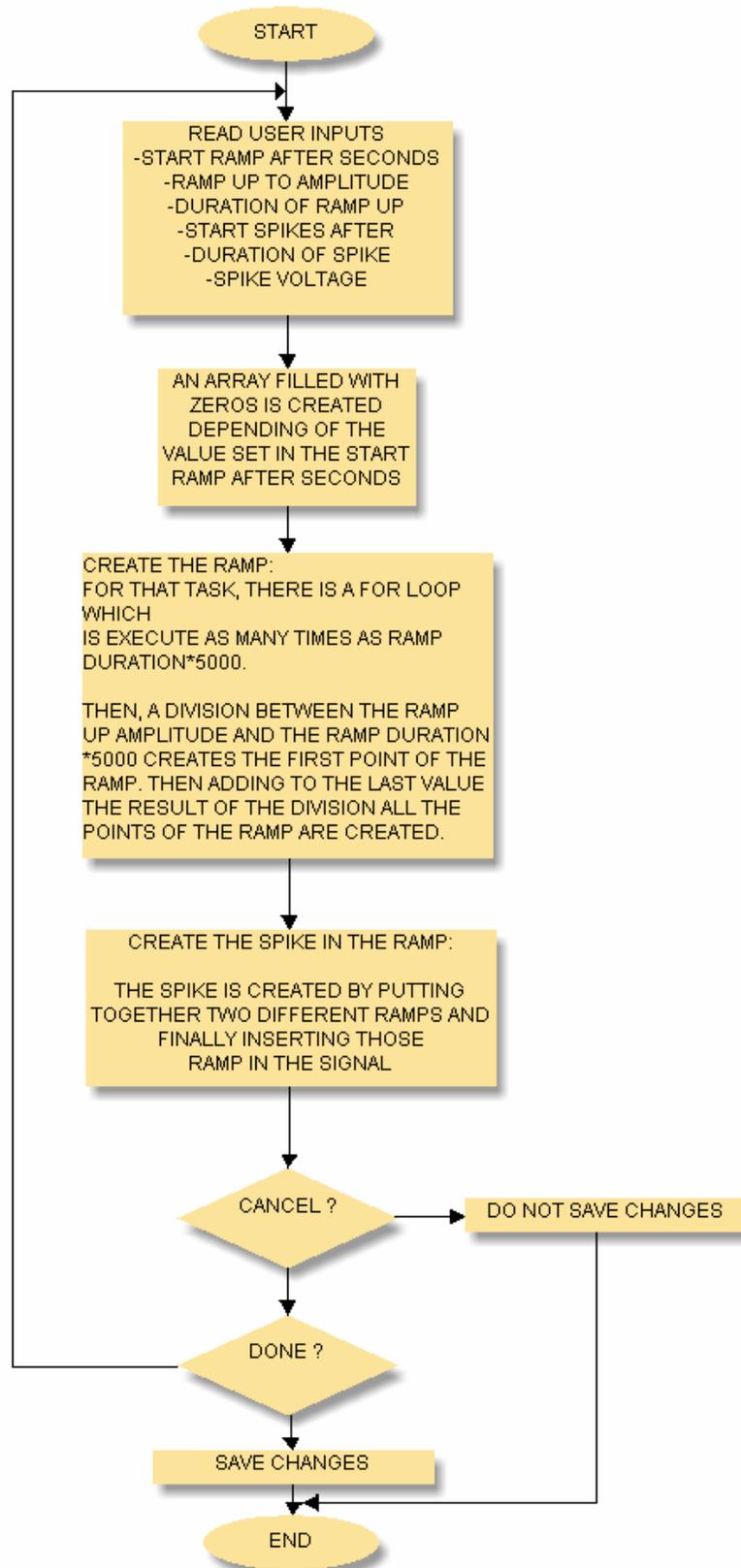
ANEXO 3: CONFIGURACIÓN DE SEÑALES (1)



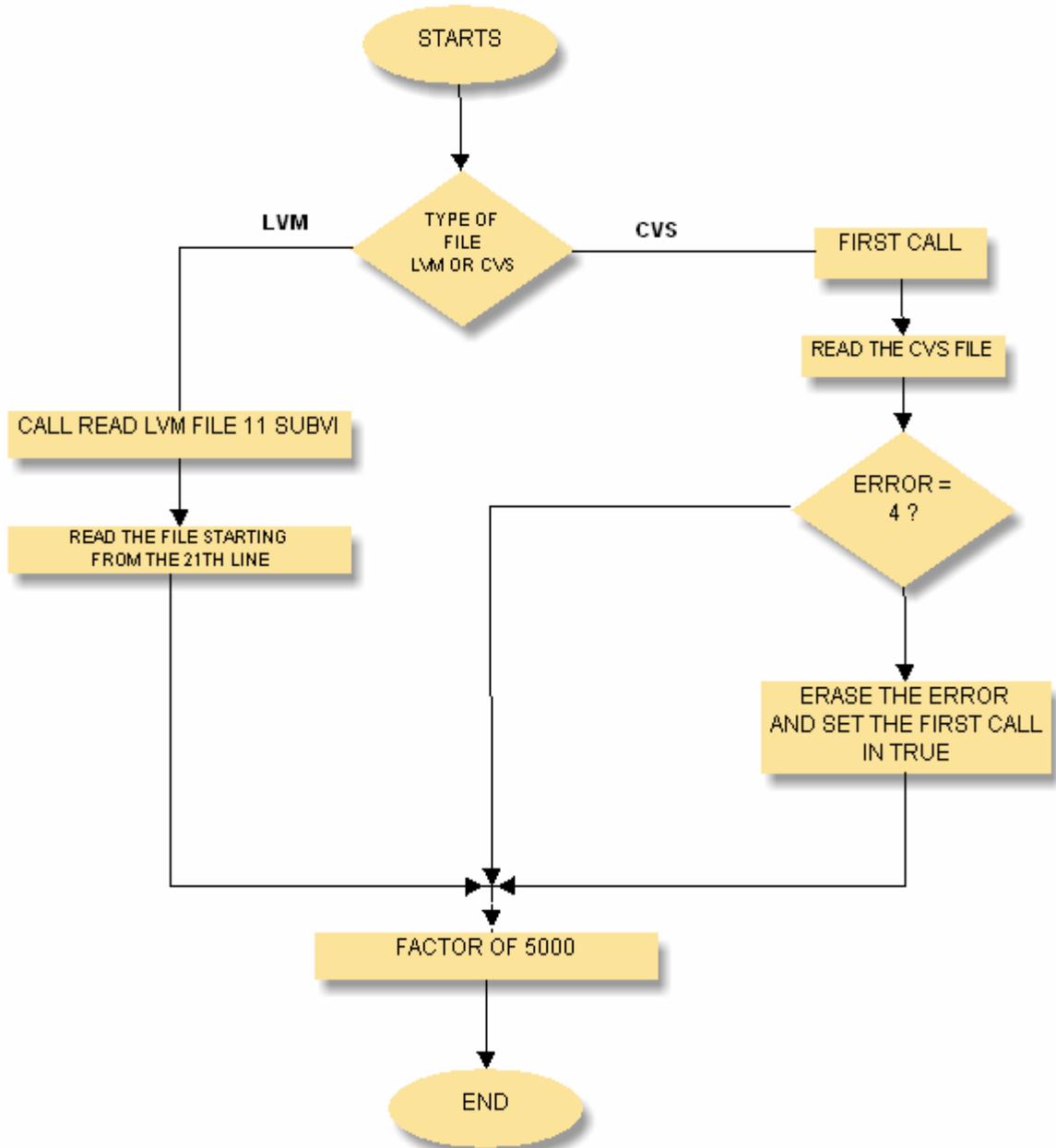
ANEXO 4: CREATE FROM SKETCH AND EDIT CURRENT (1.1 Y 1.2)



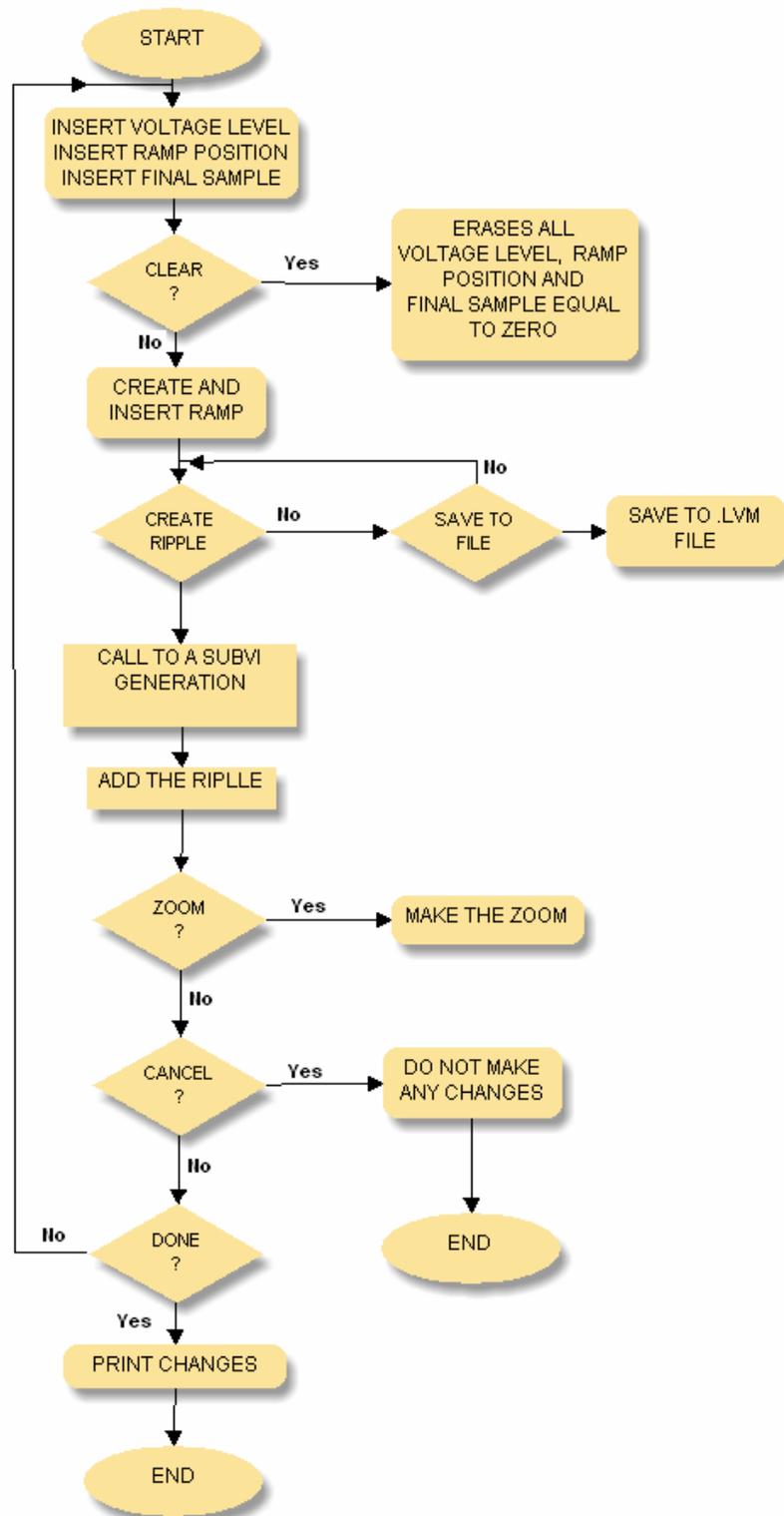
ANEXO 5: RAMP UP (1.3)



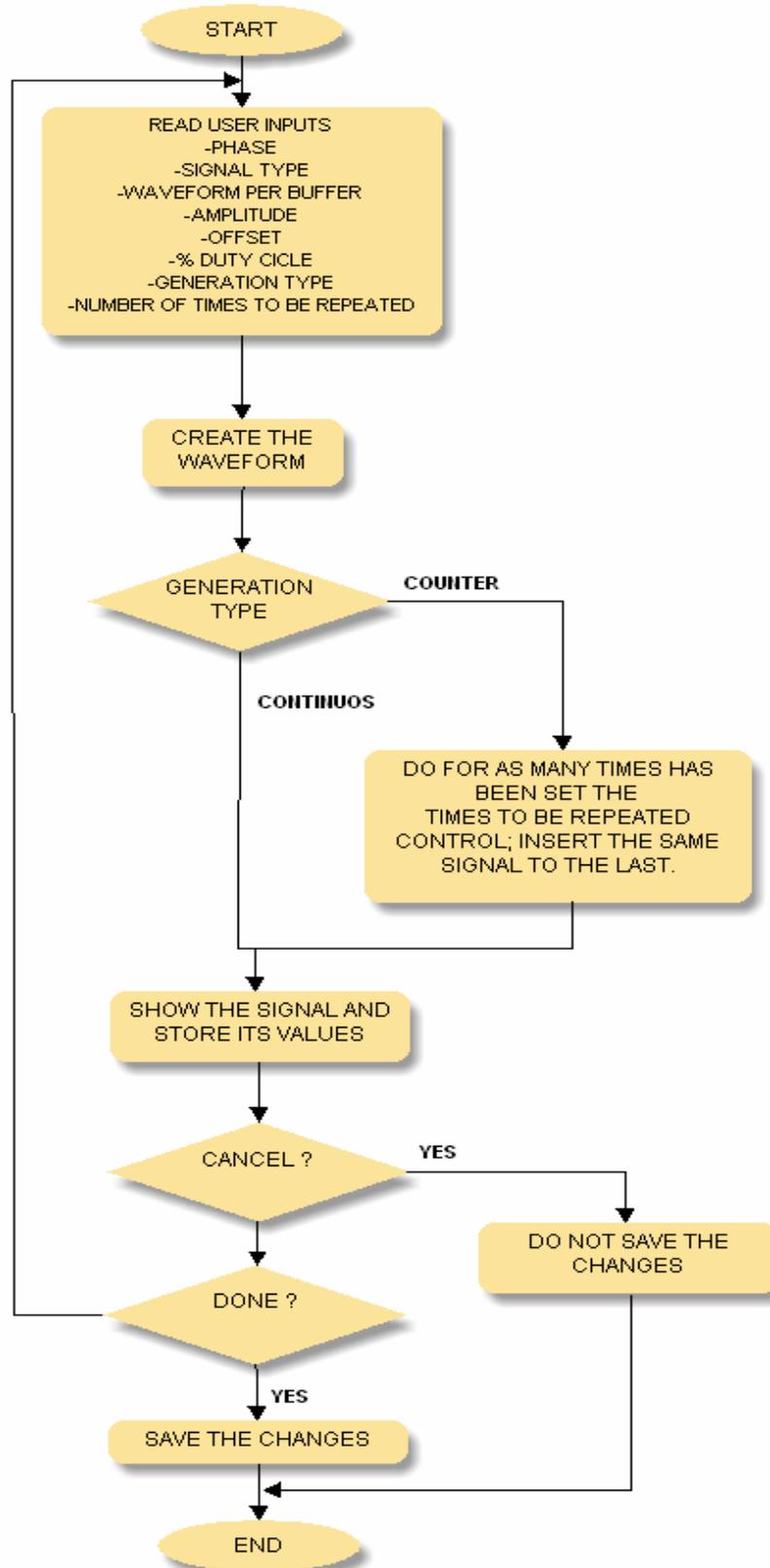
ANEXO 6: READ (FROM FILE) (1.4)



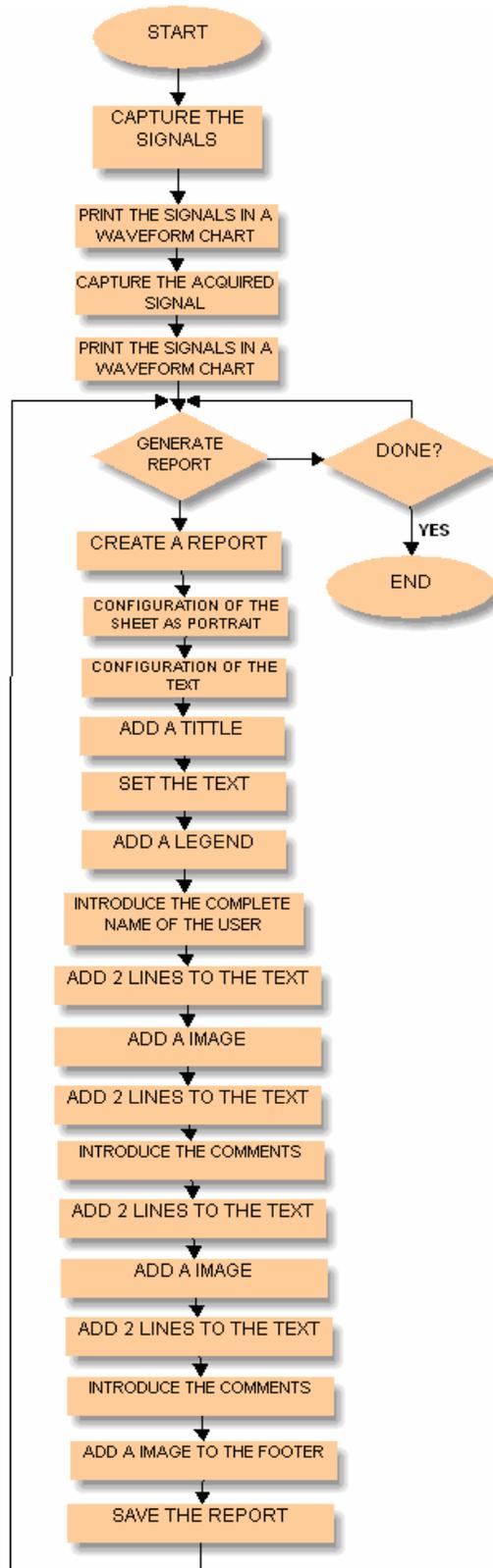
ANEXO 7: CREATE RIPPLE (1.5)



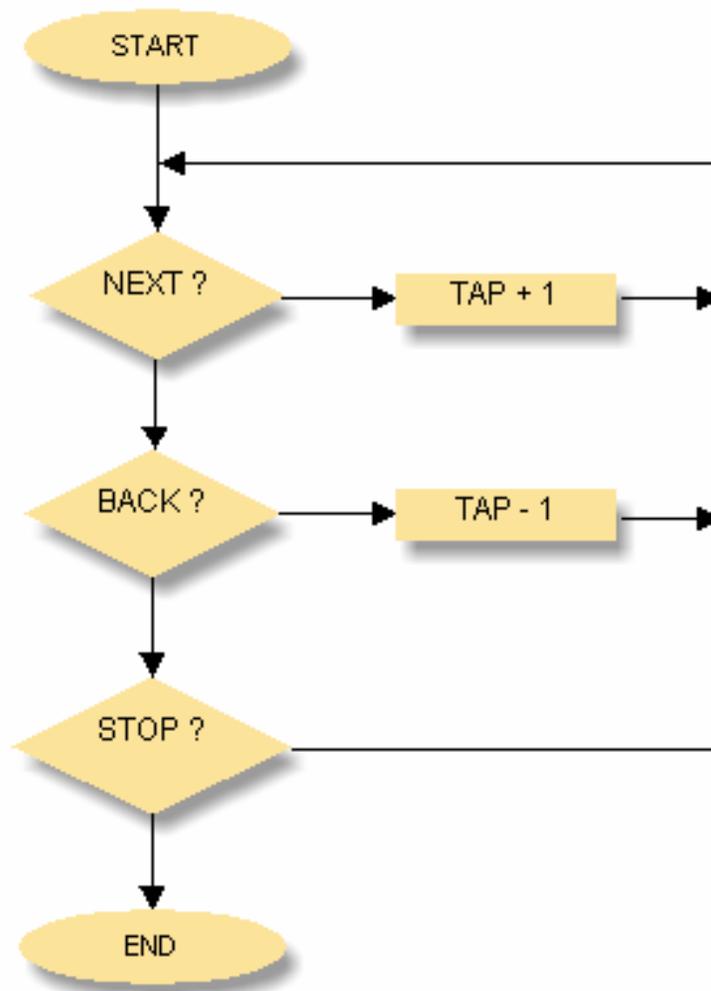
ANEXO 8: CREATE FROM LIBRARY (1.6)



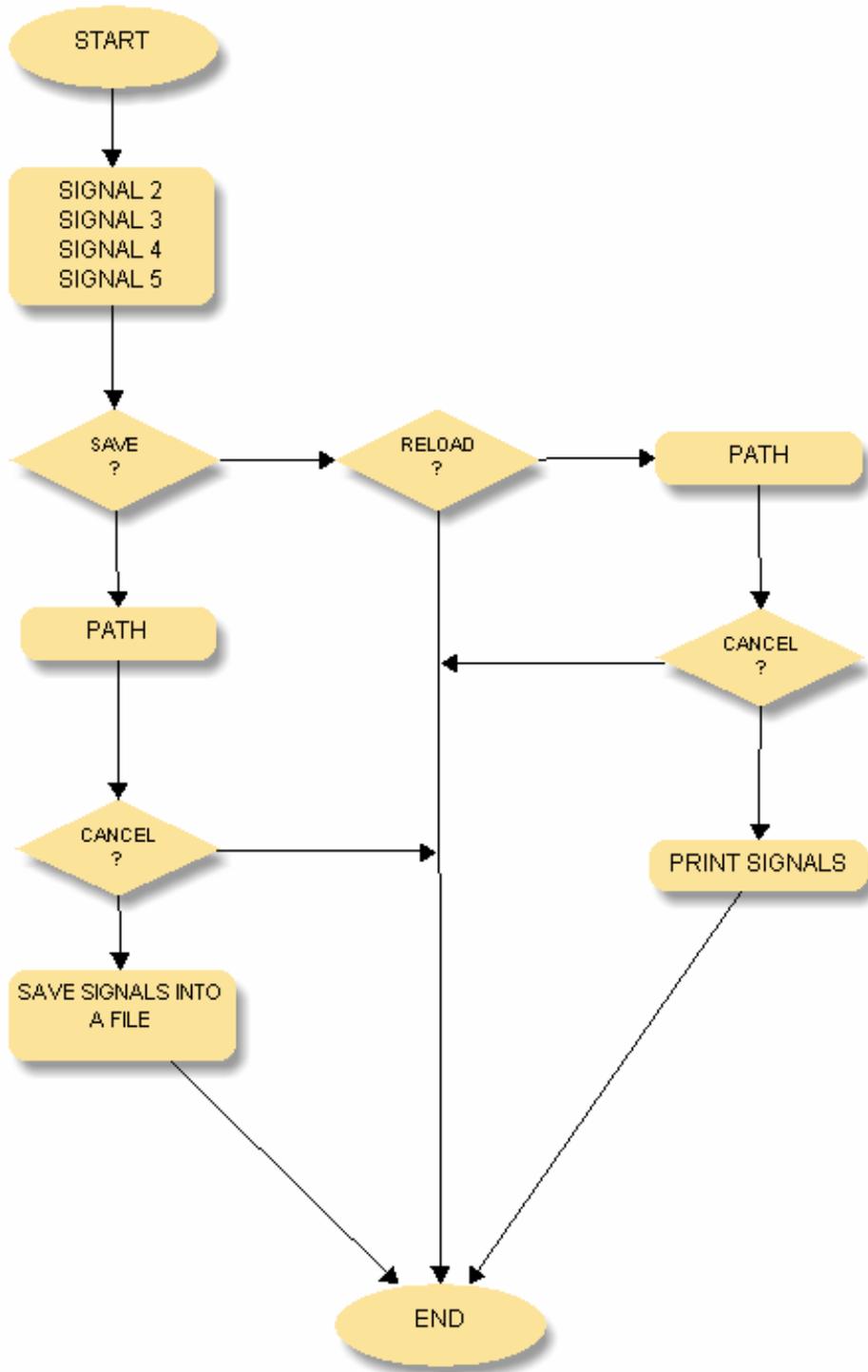
ANEXO 9: QUICK REPORT (2)



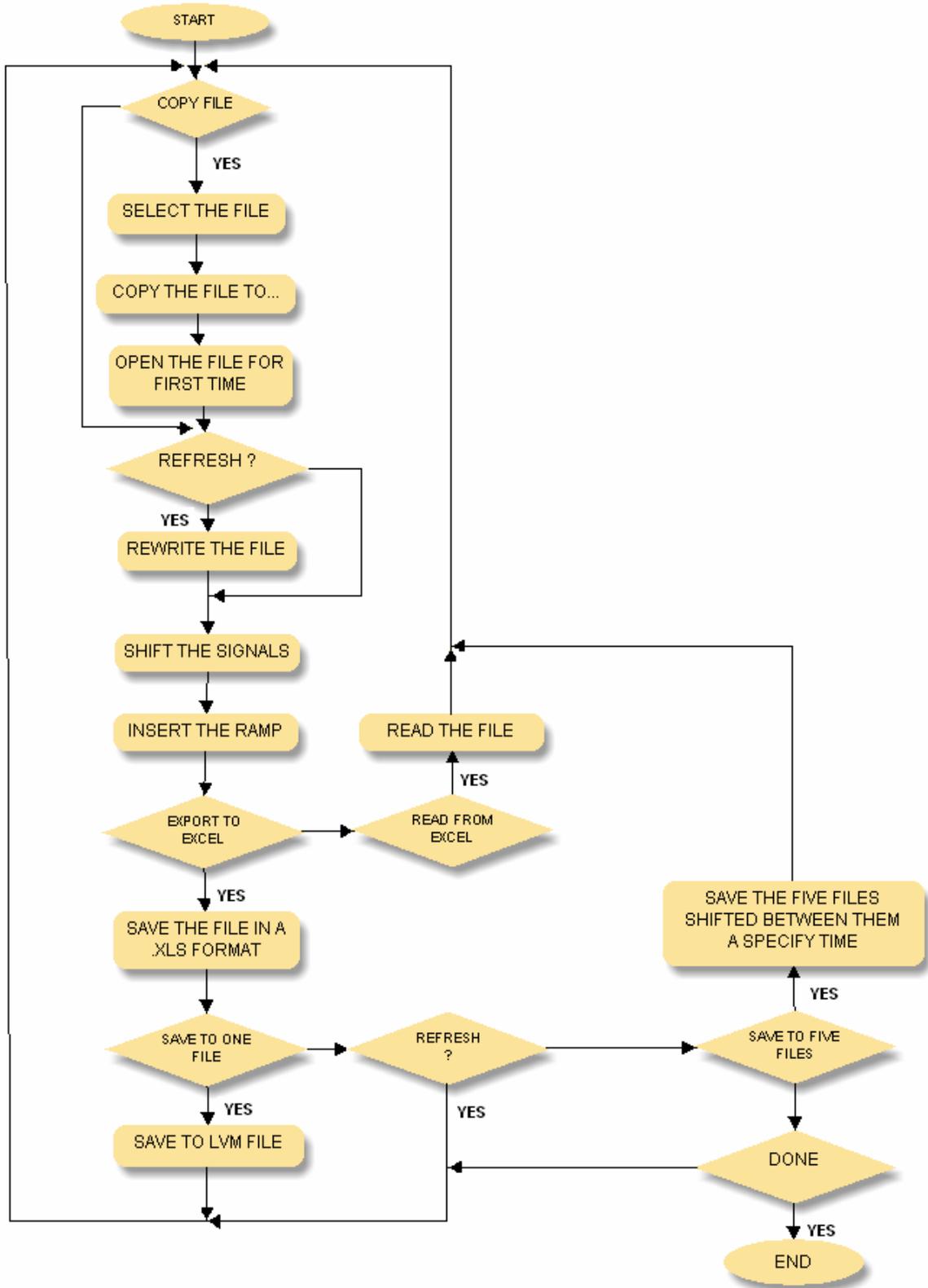
ANEXO 10: BOTON DE INTRODUCCION RAPIDA (3)



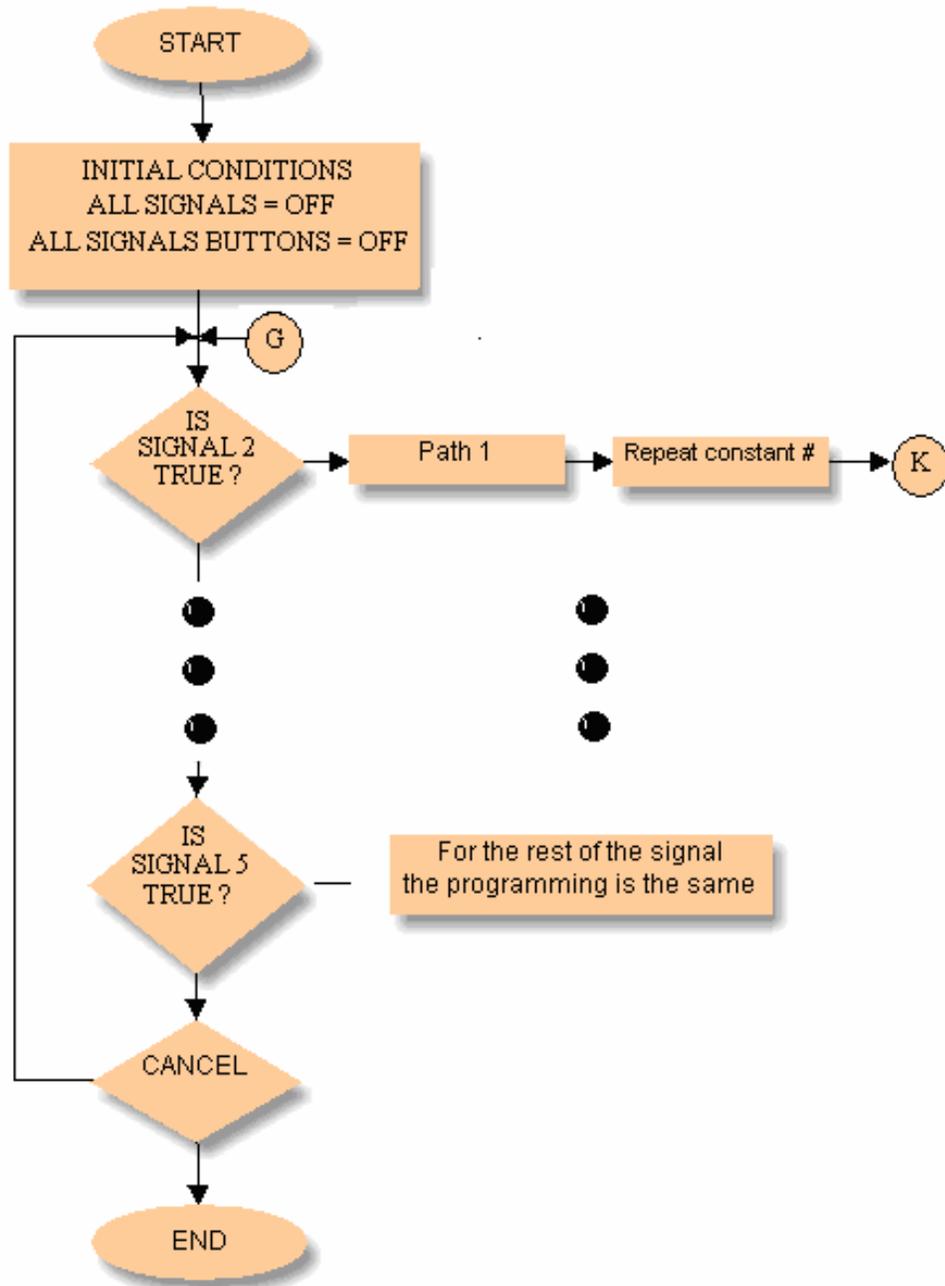
ANEXO 11: RELOAD SUBVI (4)



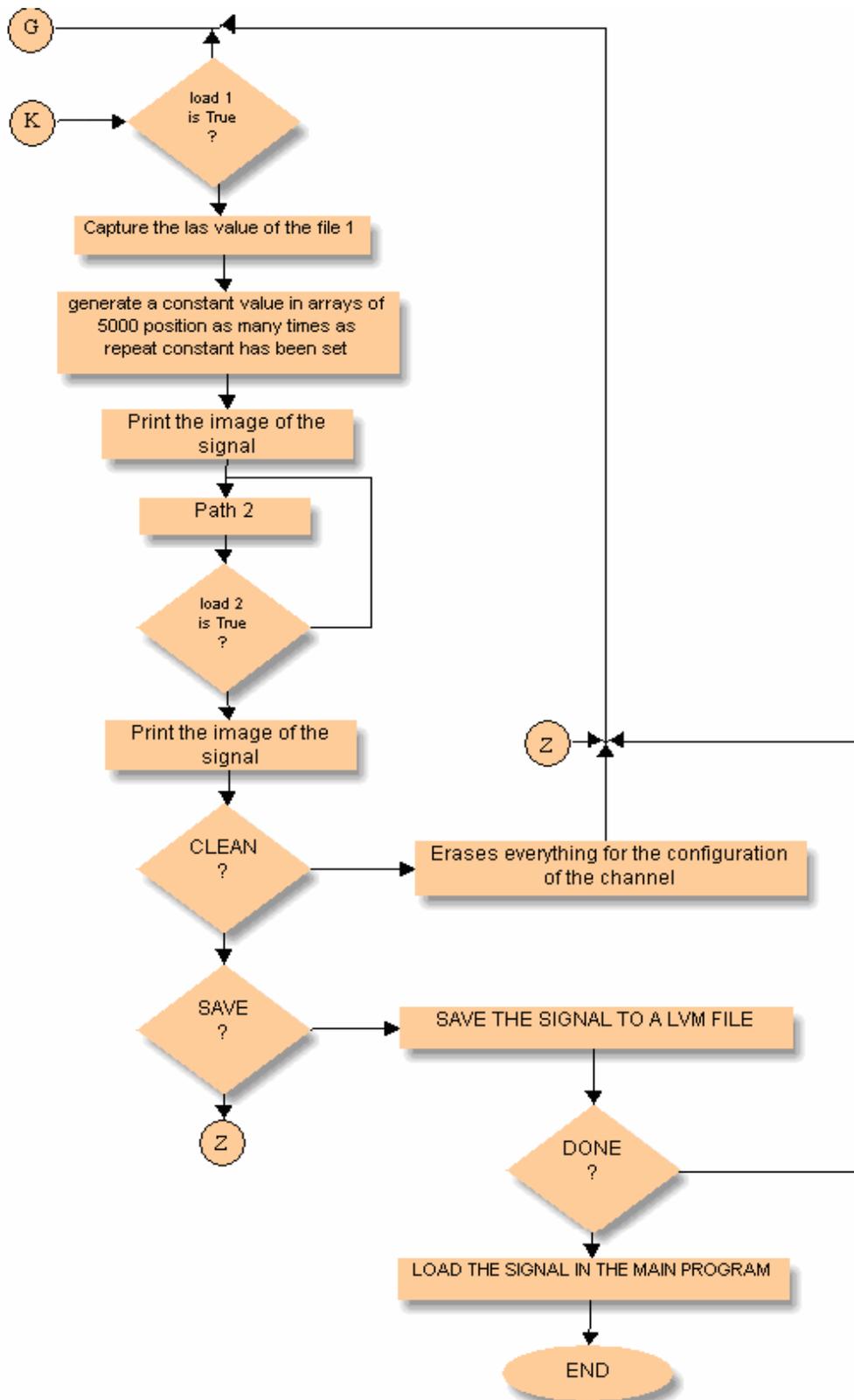
ANEXO 12: START UP CREATOR (5)



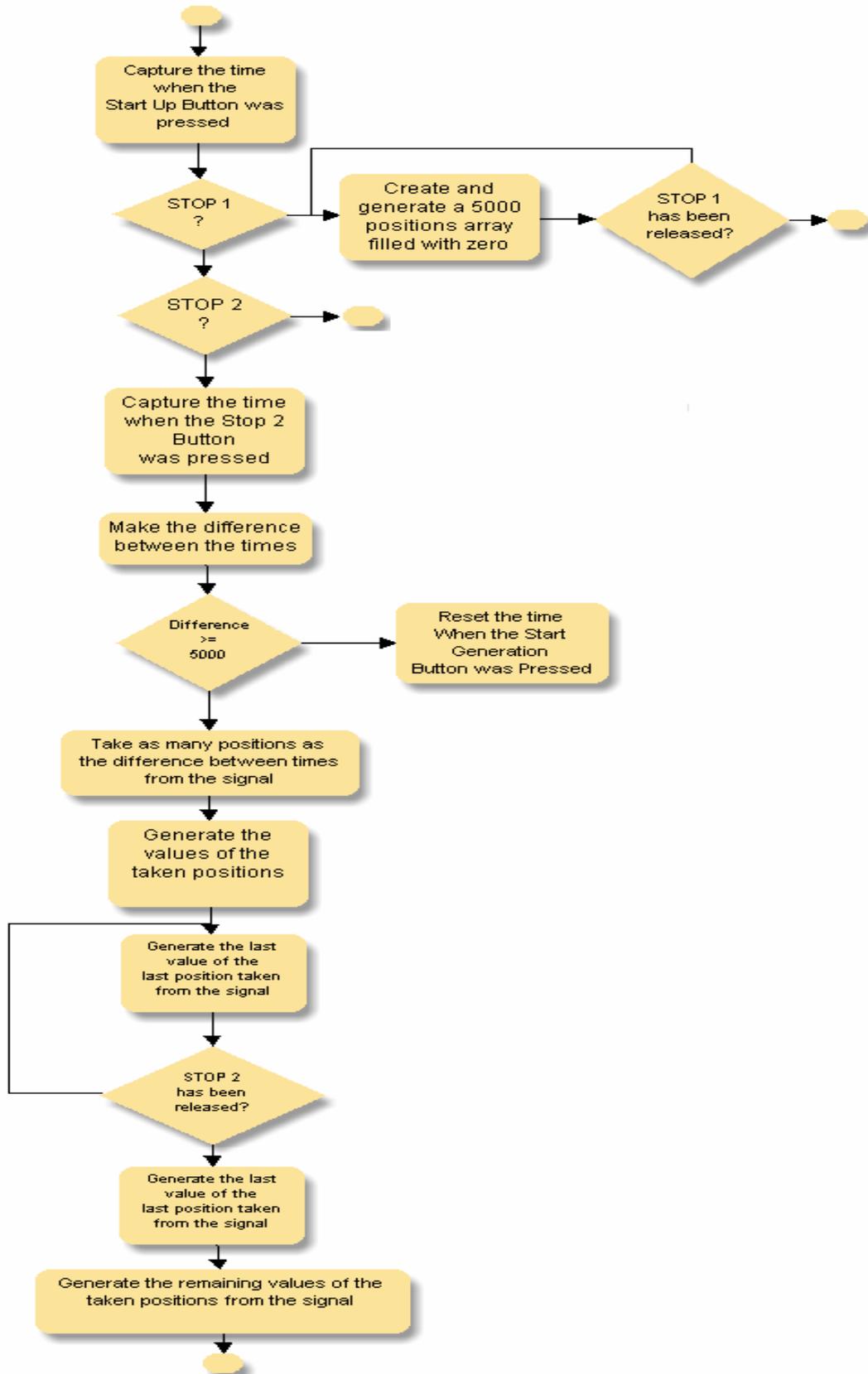
ANEXO 13: CREATE SEQUENCE SUBVI (6)



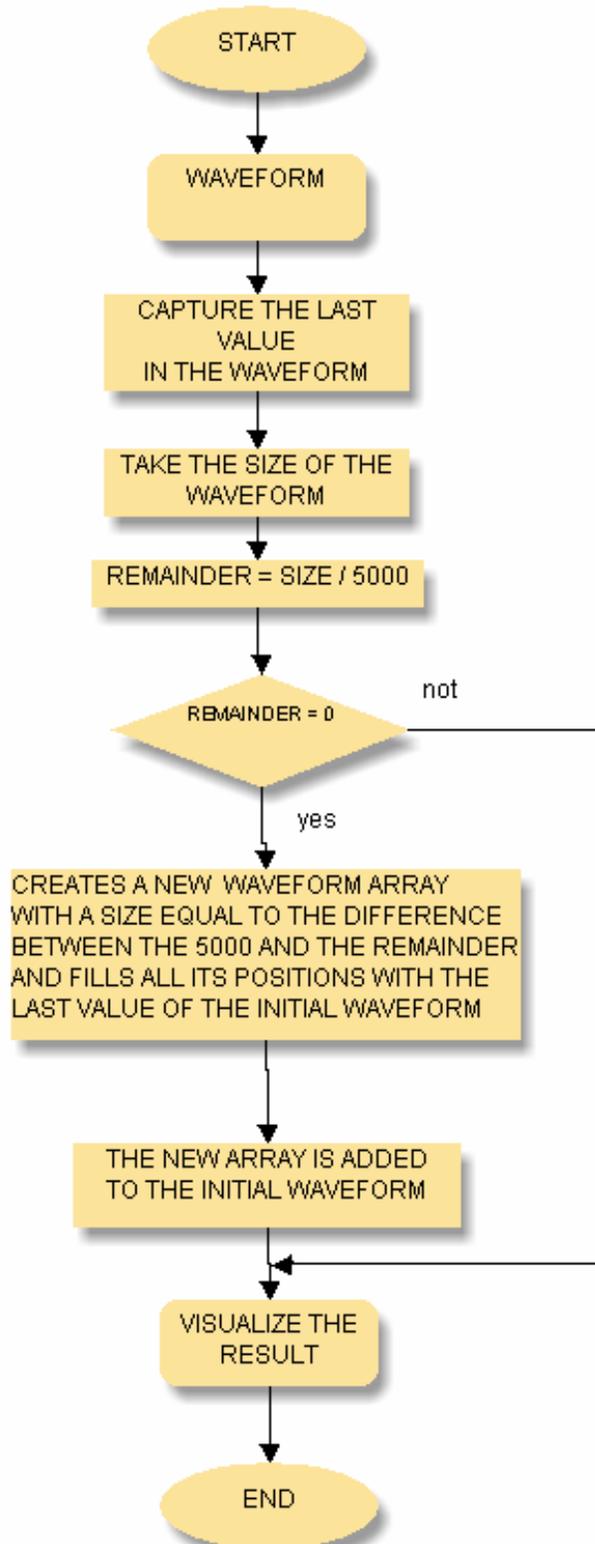
ANEXO 14: CONTINUACION CREATE SEQUENCE SUBVI



ANEXO 15: STOP 1 Y STOP 2 (7)



ANEXO 16: FACTOR DE 5000



ANEXO 17: READ LVM FILE 11 SUBVI

