



TECNICAS DIGITALES III

Guía N° 3: *Labview*

Objetivos:

Introducir al alumno a la programación gráfica con Labview

Bibliografía recomendada:

LabVIEW Basics I Course Manual (Disponible en autogestión)

Enunciado:

3 Clusters, Waveforms e interfaz con placa de audio

3.1 Nodo de fórmula

- 3.1.1 Crear nuevo VI. En el panel frontal agregar 3 controles numéricos llamados *Muestras por segundo*, *Frecuencia* y *Amplitud*. Establecer valores iniciales para cada uno y luego control por control hacer click derecho y seleccionar **Data Operations** → **Make Current Value Default Value**.
- 3.1.2 En el diagrama en bloques seleccionar **Programming** → **Structures** → **For Loop**. Hacer click con el botón izquierdo y arrastrar para dimensionar el tamaño de la estructura.
- 3.1.3 Dentro del bucle FOR, colocar un nodo de fórmula, ubicado en **Programming** → **Structures** → **Formula Node**. Hacer click con el botón izquierdo y arrastrar para dimensionar el tamaño de la estructura. Hacer click derecho sobre el lateral izquierdo del nodo de fórmula, seleccionar **Add Input** y poner como nombre de variable de entrada *f*. Repetir para las entradas *a*, *i* y *N*.
- 3.1.4 Hacer click derecho sobre el lateral derecho del nodo de fórmula, seleccionar **Add Output** y poner como nombre *y_sine*.
- 3.1.5 El nodo de formula permite escribir ecuaciones utilizando la sintaxis de lenguaje C y llamadas a las funciones de la librería math.h. Se puede acceder a la sintaxis de escritura y funciones disponibles desde la ayuda detallada. Dentro del nodo de fórmula escribir el siguiente código:

```
y_sine=a*sin(2*pi*f*i/N);
```

- 3.1.6 Conectar *Muestras por segundo* a la entrada *N* del bucle FOR y la entrada *N* del nodo de fórmula, la entrada *Frecuencia* a la entrada *f*, la entrada *Amplitud* a la entrada *a* en la ecuación y el índice *i* del bucle FOR a la entrada *i* de la ecuación.
- 3.1.7 En el panel frontal, colocar un visualizador gráfico desde **Modern** → **Graph** → **Waveform Graph** y conectar la salida *y_sine* del nodo de fórmula a la entrada del visualizador.
- 3.1.8 Agrupar el código dentro de un bucle WHILE, temporizado cada 1 segundo y expandir el índice *i* (combinando los índices del bucle FOR y del bucle WHILE) de la fórmula.
- 3.1.9 Ejecutar y verificar funcionamiento.



TECNICAS DIGITALES III

- 3.1.10 Agregar las salidas *y_ramp*, *y_square*, *y_triangle*. Escribir las ecuaciones que describan cada una de estas salidas, utilizando la ayuda como referencia de comandos disponibles.
- 3.1.11 Crear un arreglo de señales y conectarlo al visualizador. Para ello conectar todas las salidas al nodo **Programming** → **Array** → **Build Array**, dimensionandolo apropiadamente. De este modo se pueden graficar múltiples señales sobre un mismo gráfico.

3.2 Clusters y Waveforms – Salida por placa de sonido

- 3.2.1 Crear un nuevo VI a partir del anterior haciendo **File** → **Save As** → **Copy** → **Open Additional Copy**. Remover la temporización de 1 segundo del bucle **WHILE**.
- 3.2.2 Insertar un nodo de configuración de salida de audio desde **Programming** → **Graphic & Sound** → **Sound** → **Output** → **Sound Output Configure**. Colocarlo fuera y a la izquierda del bucle **WHILE**.
- 3.2.3 En el panel frontal agregar un contenedor cluster desde **Modern** → **Array, Matrix & Cluster** → **Cluster**. Los contenedores clusters pueden contener datos de distinto tipo, de manera similar a las estructuras en el lenguaje C. Dentro de este contenedor agregar tres controles numéricos llamados *sample rate (S/s)*, *number of channels*, *bits per sample*, en ese orden. Establecer valores por defecto en 10000, 2 y 16, respectivamente. Haciendo click derecho sobre los controles *number of channels* y *bits per sample*, seleccionar **Representation** y hacer click en **I32**.
- 3.2.4 En el diagrama en bloques conectar el cluster con la entrada *sound format* del nodo de configuración de salida de audio.
- 3.2.5 Extraer el dato *sample rate (S/s)* desde el cluster anterior utilizando el nodo **Programming** → **Cluster, Class & Variant** → **Unbundle by Name**. Reemplazar el control *Muestras por segundo* con la salida de este nodo. También conectar la entrada *number of samples/ch* del nodo de configuración de salida de audio a esta señal.
- 3.2.6 Dentro del bucle **WHILE**, colocar el nodo **Programming** → **Graphic & Sound** → **Sound** → **Output** → **Sound Output Write**. Fuera y a la derecha del bucle **WHILE** colocar el nodo **Programming** → **Graphic & Sound** → **Sound** → **Output** → **Sound Output Clear**. Conectar la salida *device ID* y la salida *error out* del nodo de configuración a las entradas correspondientes del nodo de escritura de audio y luego conectar las salidas *device ID* y *error out* de este al nodo de blanqueo de audio.
- 3.2.7 Crear una señal de tipo Waveform, utilizando el nodo **Programming** → **Waveform** → **Build Waveform**. Configurar el nodo para mostrar las entradas *Y*, *t0* y *dT*. Conectar la entrada *Y* a la salida *y_sine*, por fuera del bucle **FOR**. Crear una constante para *t0* y conectar la entrada *dT* a *1/N*. Conectar la salida del nodo a la entrada *data* del nodo de salida de audio. Probar el funcionamiento del VI.
- 3.2.8 Crear un arreglo con dos señales de tipo Waveform usando el nodo **Programming** → **Array** → **Build Array**. Para la segunda señal waveform crear otra senoidal de distinta frecuencia y amplitud. Reemplazar la conexión de entrada al nodo de escritura de audio por esta señal. Esto nos dará los dos canales estéreo.
- 3.2.9 Modificar el programa para que la salida se actualice cada 100ms, manteniendo la misma cantidad de muestras totales por segundo. Ayuda: Separar el sampling rate del tamaño del buffer de audio y dividir el bucle **FOR** en dos bucles anidados y actualizar el buffer en el bucle exterior.



TECNICAS DIGITALES III

3.3 Entrada por placa de audio

- 3.3.1 Crear un nuevo VI. Agregar un bucle WHILE sin temporización.
- 3.3.2 Insertar un nodo de configuración de entrada de audio desde **Programming** → **Graphic & Sound** → **Sound** → **Output** → **Sound Input Configure**. Crear una control para la entrada de configuración de formato y establecer los valores en 44100, 1 y 16.
- 3.3.3 Dentro del bucle WHILE, colocar el nodo **Programming** → **Graphic & Sound** → **Sound** → **Input** → **Sound Input Write**. Fuera del bucle WHILE colocar el nodo **Programming** → **Graphic & Sound** → **Sound** → **Input** → **Sound Input Clear**. Conectar la salida *device ID* y la salida *error out* del nodo de configuración a las entradas correspondientes de los nodos de lectura y blanqueo de audio, en cascada.
- 3.3.4 Crear un control llamado *Buffer de lectura*, y conectarlo a la entrada de *number of samples/ch* del nodo de configuración de entrada de audio y del nodo de lectura. Establecer por defecto el valor de 11025.
- 3.3.5 Conectar la salida del nodo de lectura de audio al nodo **Programming** → **Array** → **Index Array**. Establecer el índice en 0 para seleccionar el canal 1.
- 3.3.6 Conectar la salida del nodo de selección de arreglo al nodo **Programming** → **Waveform** → **Get Waveform Component**. Colocar un visualizador gráfico y conectar la salida.
- 3.3.7 Determinar si la señal cruza un umbral programable (establecido mediante un control) y señalizarlo con un indicador led. Sugerencia: Ver Guía 1.
- 3.3.8 Sobre el lado derecho del bucle WHILE, hacer click derecho y seleccionar **Add Shift Register**. Este es un elemento de memoria que nos permitirá guardar el valor de una iteración escribiendo del lado derecho y leer el valor de la iteración anterior. Usar esto para encender el led de manera permanente al cruzar el umbral.