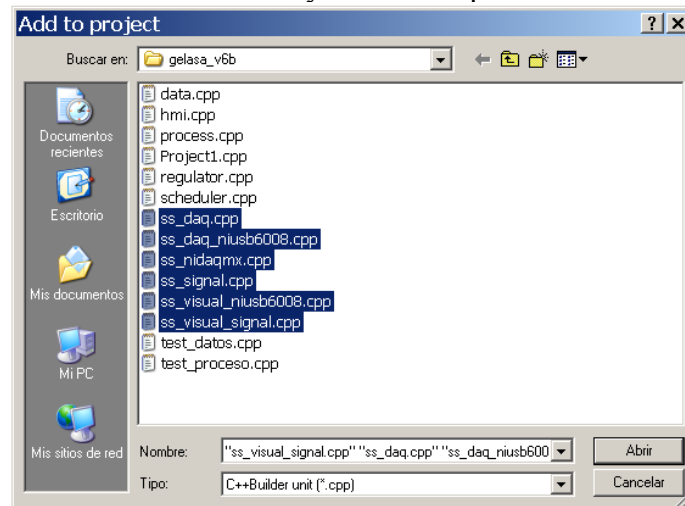


# Instrucciones rápidas de uso del Simulador SimSeny II v. alpha 0.0.1

## Preparación del proyecto

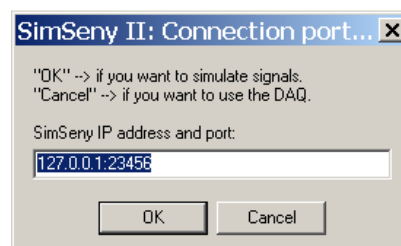
1. Descargar los archivos del simulador
2. Descomprimir en la carpeta del proyecto (son un montón de ficheros ss\_\*.\*)
3. Abrir nuestro proyecto con Borland C++ Builder
4. Añadir al proyecto todos los archivos ss\_\*.cpp utilizando la opción de la barra de menús “Project -> Add to project ...”. Con la tecla “Ctrl” y el ratón se pueden seleccionar todos de golpe.



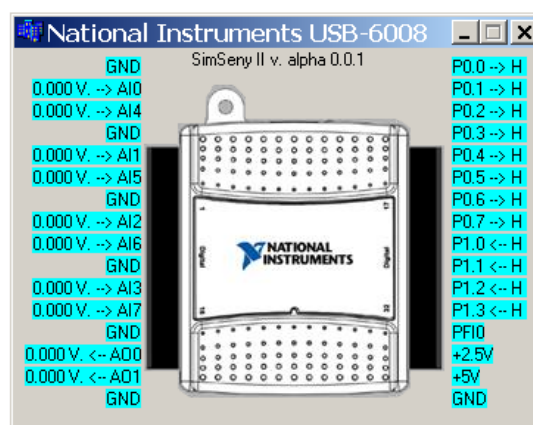
5. Comprobar la correcta integración utilizando la opción del Menú “Project -> Build All”
6. Modificar el módulo de proceso cambiando la cabecera “nidaqmx.h” por “ss\_nidaqmx.h”  
`#include "nidaqmx.h" --> #include "ss_nidaqmx.h"`
7. Comprobar de nuevo la correcta integración utilizando la opción del menú “Project -> Build All”

## Uso simulador

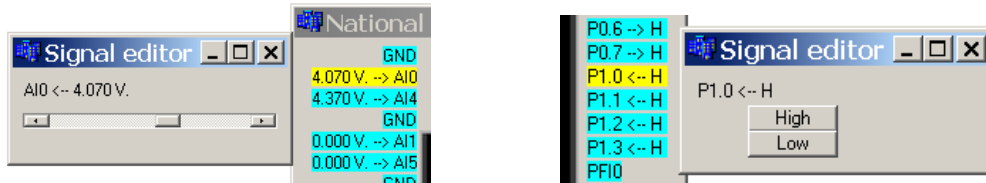
1. Ejecutar nuestro programa. Al ejecutarse la primera instrucción que emplee la tarjeta, el simulador preguntará si se desea usar la tarjeta real o simularla.



2. Pulsar “OK” para simular. Aparecerá un croquis de la tarjeta con el valor de los pines.



3. Pasar el ratón por encima de las señales de entrada para modificar el valor físico que deseamos aplicar a ese pin. Atención: que apliquemos 500 voltios al pin de la tarjeta no implica que esa lectura llegue a nuestra aplicación.



## Llevarse el proyecto a casa

Para que funcione la aplicación que hace uso de las librerías NiDAQmx de National Instruments es imprescindible que se instalen previamente dichas librerías del fabricante.

Para evitarlo, el simulador puede “engañar” al programa para que crea que dichas librerías están instaladas, pero será necesario introducir cambios.

Para llevarse a otro computador el proyecto (y el ejecutable) y funcione, se tiene dos opciones excluyentes:

- a) Descargar las librerías NiDAQmx de National Instruments disponibles en su web. En la web de la asignatura hay un enlace para localizar más fácilmente dónde descargar. Las librerías ocupan unos 300 MB e incluyen montones de ejemplos y manuales.
- b) Modificar el proyecto para que el simulador active el engaño. Para ello añadir la siguiente línea ANTES de la inclusión de la cabecera del simulador.  
A continuación eliminar NIDAQmx\*.lib de nuestro proyecto.

```
#define SS_FALSE_DAQ      // <-- AÑADIR ESTO
#include "ss_nidaqmx.h" //SimSeny header file
```

## Funciones NiDAQmx soportadas

Esta versión del simulador soporta las siguientes funciones NiDAQmx:

```
DAQmxCreateTask()
DAQmxStartTask(A)
DAQmxStopTask(A)
DAQmxGetErrorString(A,B,C)
DAQmxCreateAIVoltageChan(A,B,C,D,E,F,G,H)
    units=DAQmx_Val_Volts
DAQmxCreateAOVoltageChan(A,B,C,D,E,F,G)
    units=DAQmx_Val_Volts
DAQmxCreateDChan(A,B,C,D)
    lineGrouping=DAQmx_Val_ChannelsForAllLines
DAQmxCreateDOChan(A,B,C,D)
    lineGrouping=DAQmx_Val_ChannelsForAllLines
DAQmxReadAnalogScalarF64(A,B,C,D)
DAQmxWriteAnalogScalarF64(A,B,C,D,E)
DAQmxReadDigitalScalarU32(A,B,C,D)
DAQmxWriteDigitalScalarU32(A,B,C,D,E)
```