

LabVIEW Driver Setup Guide

The AIO Test System provides API access to multiple software languages, allowing you to write your own audio test routines in your preferred language. The Echo AIO LabVIEW driver facilitates interfacing the Echo AIO hardware with LabVIEW. This guide assumes that you already have some experience programming with LabVIEW.

Installation

To install, run the Echo AIO LabVIEW driver installer and follow the prompts. The driver will add Echo AIO to your LabVIEW instr.lib folder. The dll files that are installed with the driver are specific to 64-bit or 32-bit LabVIEW—if you wish to install into both versions, be sure to run the installer twice rather than just copying the files. After installation, the driver files may be accessed from the LabVIEW Instrument I/O palette, from the file menu, or from the Echo AIO VI Tree.

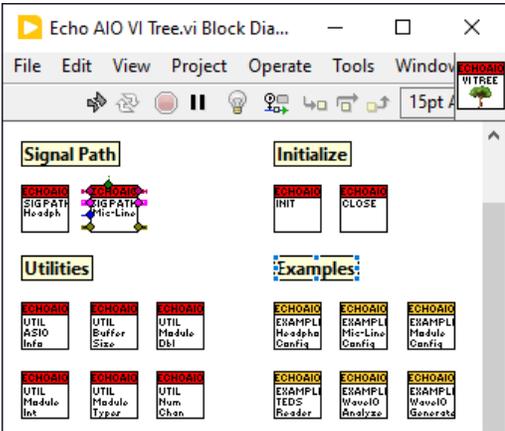


Fig 1. VI tree.

On first use, LabVIEW may recompile the VIs due to path variations on your computer. Mass Compile may be run on the Echo AIO folder to recompile all

files at once. Other than this, you should not make any changes directly to the driver VIs, as some VIs are interdependent, and they will get overwritten on future updates.

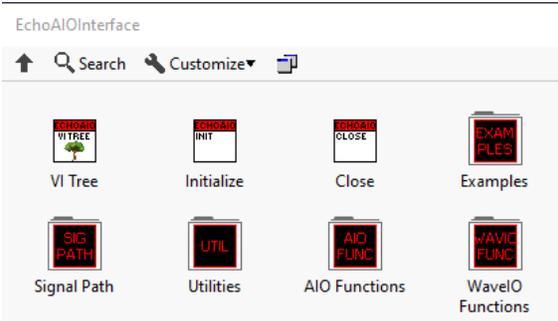


Fig 2. Instrument I/O > Echo AIO palette.

This driver is written in the version of LabVIEW that was current when released. Please check the Echo website for the latest version as it is periodically updated. LabVIEW code is not backwards compatible, and reverting to an earlier version may be possible but is not supported. Make sure to also install the latest version of the AIO Windows software package.

Signal Path and Utilities VIs

The Signal Path and Utilities folders contain the top-level VIs that are intended for utilization in your own projects.

The Signal Path VIs enable you to read, or write and read, of all the settings for each type of I/O connector available on the various AIO modules and duplicates the functionality of the AIO control panel. For each of these VIs, you will need to set whether it communicates with the inner or outer module slot (you can set this programmatically if you wish to from

the ModuleType data in the AIO class or the Module Type Out array of the AIO Initialize VI). Setting the Set toggle to True will write parameters as well as read them.

The Utilities VIs generally allow you to read, or write and read, general interface parameters, or parameters that apply to multiple types of modules. A VI to read TEDS transducer data is also included here.

In order to make use of any of the Signal Path or Utilities VIs, you must first initialize or open the connection using the Echo AIO Initialize VI, and afterwards close the connection using the Echo AIO Close VI. The Echo AIO object class wires must also be connected between the VIs. The class connection helps ensure that the VIs are wired correctly, and also passes key hardware parameters that are detected upon initialization.

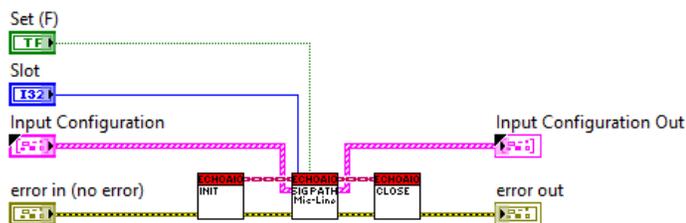


Fig 3. Initialize and close VIs with AIO class connection wired.

Example VIs

The example folder contains read-to-run examples that make use of various Signal Path and Utility VIs, with the Initialize and Close VIs already connected as required. Examples are also provided to demonstrate the use of the included WaveIO ASIO utility (see further explanation below).

Lower Level Function VIs

The VIs in the Functions folder each contain a specific call library function linked to the EchoAIOInterface dll. These give you direct access to all the calls in the API. Virtually all of these calls have already been utilized as sub-VIs in the Signal Path and Utility VIs, and in general it should not be necessary to call them directly. You may, however, make use of them if you wish to do your own customization.

WaveIO

Echo has licensed the WaveIO ASIO library for use by our customers in projects associated with the Echo AIO. WaveIO includes LabVIEW VIs that directly

access the ASIO interface, and therefore can read and write audio data to ASIO devices, such as the Echo AIO, installed on your computer. This allows you to generate and analyze signals and construct an entire audio test solution within LabVIEW utilizing the Echo AIO hardware.

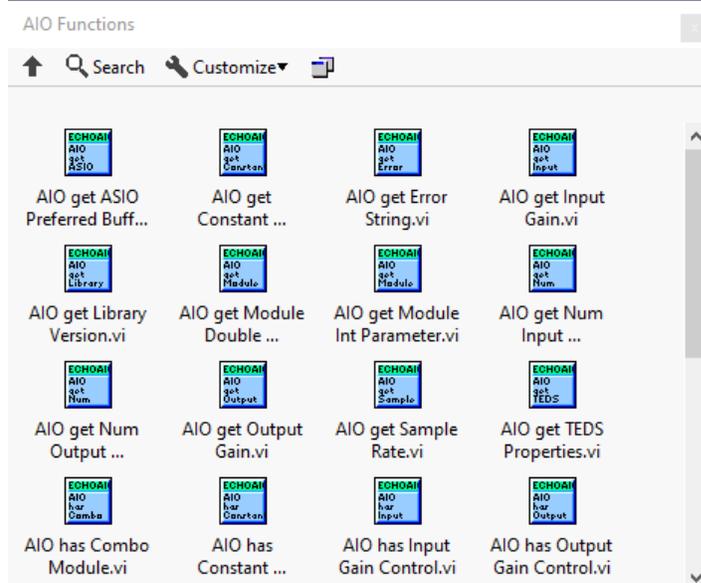


Fig 4. AIO function VIs.

WaveIO is completely independent from the Echo AIO LabVIEW driver, and it is not necessary to initialize the AIO driver to use WaveIO. WaveIO is used for generating and analyzing audio signals, and the AIO LabVIEW driver is used for configuring the AIO hardware. WaveIO is a product of Christian Zeitnitz (zeitnitz.eu/waveio). The complete license terms are contained in the AIO driver WaveIO subfolder.

We have provided examples utilizing WaveIO to generate and analyze signals with the Echo AIO in the Examples subfolder. Before using these, run the ASIO Info VI in the Utilities subfolder to determine the ASIO device number of the AIO instrument. The ASIO device number must be set in the front panel of the examples before running them. In addition, since ASIO is a digital interface, the correct AIO input or output connector type must be selected in order for the analog voltage settings or readings to be scaled correctly.

These examples provide a good starting point for making your own VIs utilizing WaveIO, as they contain all the basic building blocks. Make sure to save any modifications under a new name in a separate folder.

The low-level WaveIO functions are contained in the WaveIO Functions subfolder. As the WaveIO library

provides functions for both Windows WASPI audio and ASIO, some functions require you to select ASIO as a parameter, and some do not apply at all to ASIO but are retained to keep the full library intact. Low-level functions are also available for reading and writing ASIO data to disc.

If you have any questions or need assistance, please do not hesitate to contact Echo sales or support.