Listen

Sound Measurements Make Sound Products

Application Note

External Control of SoundCheck via TCP/IP

Introduction

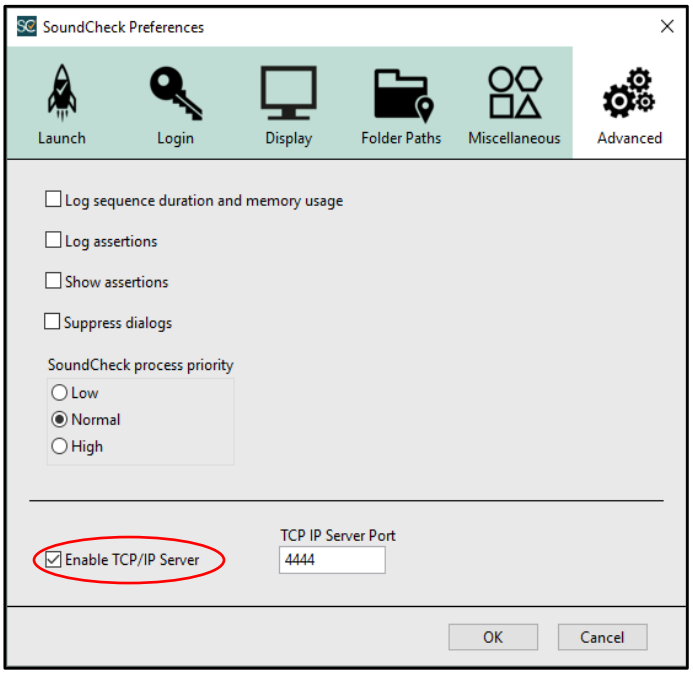
TCP/IP control of SoundCheck first appeared in SoundCheck 15, offering users the ability to connect to SoundCheck via any programming language, on any operating system, either locally or through a network. It also provides a more powerful and expandable command format for interacting with SoundCheck. This is extremely valuable for anyone who needs to control SoundCheck from an external program; for example, as part of an overall test plan or for factory automation. Using this feature, a single computer can control multiple SoundCheck systems, simplifying production line measurements.

SoundCheck contains external control examples in a variety of programming languages including C#, C++, MATLAB, VB.net, LabVIEW and Python. For the purpose of this note, we will focus on the Python examples as Python is one of the most widely used and popular programming languages in the world today and Listen has created an extensive library of Python commands to support this feature. See the accompanying PDF "**SoundCheck TCPIP Python Library**" for the current list of supported commands.

TCP/IP Setup

From the SoundCheck **Edit** menu:

- Select **Preferences**
- Click on the **Advanced** tab
- Check Enable **TCP/IP** Server



Security

The first time you enable TCP/IP in SoundCheck Preferences, you may get a Windows Security Alert prompting you to allow SoundCheck to communicate on the selected network. Click **Allow Access** and continue.

Manual Setup

The "Enable TCP/IP" and "Port #" settings are stored in the **SoundCheck *18*.ini** file found in the root of the SoundCheck folder.
[External Control]

- TCP IP SERVER ENABLED = TRUE or FALSE
- TCP IP SERVER PORT # = 4444

These settings can be modified manually if necessary.

Software Requirements

- SoundCheck 18
- Python 3.7

Hardware Requirements

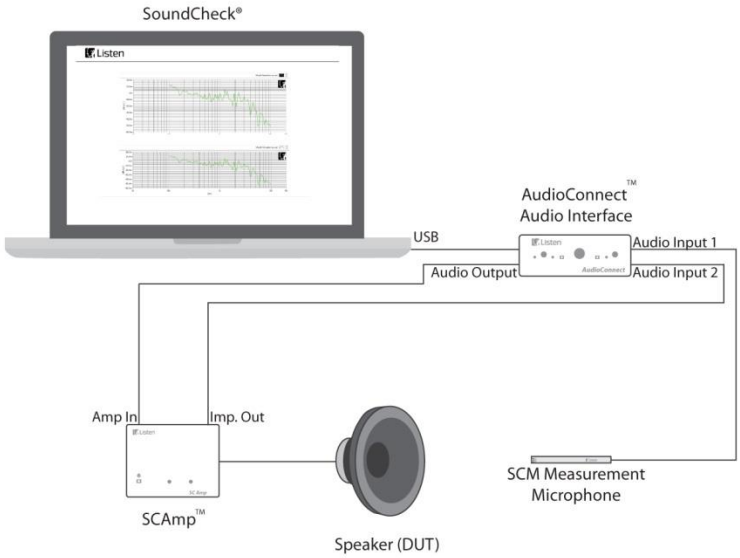
- Reference microphone – Listen SCM-3 or similar (Listen part number 4002)
- Audio interface/microphone power supply – Listen AudioConnect or similar (Listen part number 4050)
- Audio power amplifier – Listen SCAmp or similar (Listen part number 4060)
- Impedance box/impedance reference (optional)

Hardware Setup and Calibration

- Configure the hardware per the system diagram below
- Calibrate the Amplifier and microphone per the instructions in the SoundCheck user manual

You are now ready to run the SoundCheck external control example Python scripts.

System Diagram



Python Setup

Important! Prior to running the Python external control example scripts, please do the following:

- Make sure that SoundCheck 18 and Python 3.7 (or higher) are installed on your SoundCheck PC
- In your SoundCheck 18\External Control Examples\Python folder, follow the Installation Notes contained in the readme.txt file

SoundCheck Python Example Scripts

There are three example scripts included with SoundCheck 18.

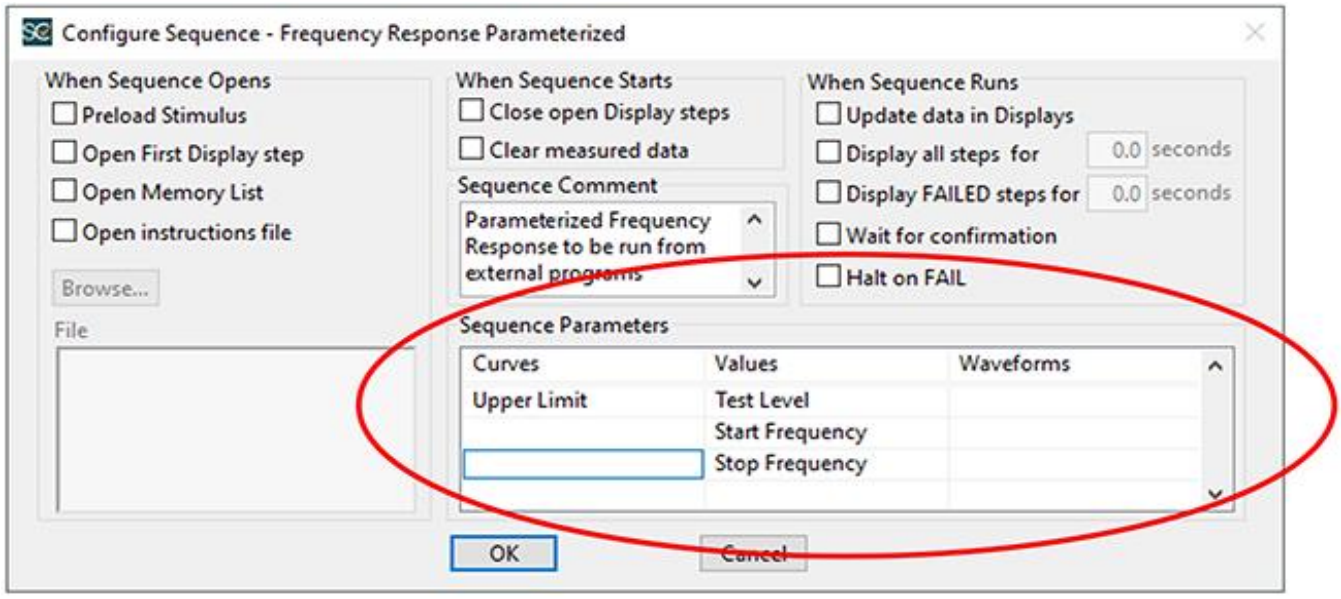
1. SimpleSoundCheckExample.py

This script will launch SoundCheck, open and run the Complete Test example sequence, close SoundCheck and display the Fundamental response of the DUT on an x-y graph.

To run this script from a command line, copy and paste the following: python SimpleSoundCheckExample.py

2. FrequencyResponse.py

This script will run a simple frequency response example sequence from a command line interface. The example sequence utilizes the new MemoryListSet command which allows the user to create placeholder curves, values, results and waveforms in the MemoryList and pass values into the placeholders via external control. The placeholders are created in the Configure Sequence editor as shown below:



To run this script from a command line, copy and paste the following:

python FrequencyResponse.py --launch --fstart=100000 --fstop=1000

This will launch SoundCheck, open the Frequency Response Parameterized sequence, Pass Start (10kHz) and Stop (1kHz) values into the Memory List for Stimulus range, run the sequence, close SoundCheck and show the Frequency Response Limit Pass/Fail verdict and margin in the terminal window.

3. WAVspectrum.py

This script will read any WAV file and use it as a stimulus for performing an FFT Spectrum measurement in SoundCheck.

To run this script from a command line, copy and paste the following:

python WAVSpectrum.py "C:\SoundCheck 18\wav files\itu real speech.wav" --launch --channel=1

This will launch SoundCheck, open the Spectrum Arbitrary Stimulus sequence, scale the WAV file of interest and pass it into the Memory List placeholder “”Stimulus”, run the sequence, close SoundCheck.

Notes

- The sequences and Python scripts referenced in this note are all included in the SoundCheck 18 install.
- No sequences or scripts are included in this distribution folder.
- The accompanying PDF “SoundCheck TCPIP Python Library” is also available in the SoundCheck 18 installation under SoundCheck 18\External Control Examples\Python