Application Report

SLOU – August 2015

PurePath™ Smart Amp User Manual (TAS5766M/68M)

Jorge Arbona

High Performance Analog / Audio

ABSTRACT

PurePath™ Smart Amp provides powerful tools to make speaker characterization and tuning easy.

Smart Amp replaces traditional continuous power design principles and hardware based speaker protection methods with algorithms that allow significant increases in peak power output, loudness and sound quality relative to conventional amplifiers.

Smart Amp tools allow developers to understand how speakers are performing in the system and then make informed decisions to improve performance. The algorithms, characterization and tuning tools allow developers to overcome a wide variety of audio challenges.

WARNING: EXPORT NOTICE

Recipient agrees to not knowingly export or re-export, directly or indirectly, any product or technical data (as defined by the U.S., EU, and other Export Administration Regulations) including software, or any controlled product restricted by other applicable national regulations, received from Disclosing party under this Agreement, or any direct product of such technology, to any destination to which such export or re-export is restricted or prohibited by U.S. or other applicable laws, without obtaining prior authorization from U.S. Department of Commerce and other competent Government authorities to the extent required by those laws. This provision shall survive termination or expiration of this Agreement.
According to our best knowledge of the state and end-use of this product or technology, and in compliance with the export control regulations of dual-use goods in force in the origin and exporting countries, this technology is classified as follows:

US ECCN: 3E991
EU ECCN: EAR99

And may require export or re-export license for shipping it in compliance with the applicable regulations of certain countries.

Contents

Figures

Tables

1 General Overview

Chapter 1 describes Smart Amp, its features and the hardware and software tools required for evaluation and development.

Chapter 2 describes how to install the *PurePath™ Console 3* software.

Chapter 3 discusses how to perform a speaker characterization.

Chapter 4 explains how to tune loudspeakers with Smart Amp.

Chapter 5 helps the design engineer understand how to assess current requirements, and then determine the supply voltage levels and the current needed for Smart Amp. With that knowledge the designer can adapt conventional power systems to peak output power needs of Smart Amp designs.

Chapter 6 explains how to transfer the Smart Amp code into a microcontroller so that coefficients can be downloaded into Smart Amp in the end-system at start-up.

For support, please refer to Chapter 7.

For a list of supported speaker configurations, refer to Appendix A.

1.1 Supported Smart Amp Devices

This document supports the following *Smart Amp* devices:

1. TAS5766M
2. TAS5768M

1.2 Migrating from Previous Versions of Smart Amp

PurePath™ Console 3 can import snapshot (.cnsss3) files from PurePath™ Console 1. Snapshot files can be created from .ccdef files, if necessary. Refer to the PurePath™ Console 1 Quick Start User's Guide for details on how to create snapshots.

1.3 What is Smart Amp?

Conventional hardware-based speaker protection matches the continuous power output of the audio amplifier with the speaker output rating and sometimes incorporates high pass filtering to prevent over-excursion. *PurePath™ Smart Amp* replaces hardware based speaker protection methods with predictive algorithms, speaker characterization tools and real-time signal monitoring to increase the peak output of the speaker without damage.

The first implementation step of *Smart Amp* based audio solutions is characterizing the speaker with TI's *PurePath™ Console 3* and the *PurePath™ Learning Board*. These are powerful, easy-to-use tools designed specifically to simplify system level characterization, tuning and implementation. The characterization process creates a digital model of the speaker based on thermal, electro-mechanical and acoustic parameters.

The output of the characterization process is an initial set of coefficients that define the *Safe Operating Area* (SOA) which establishes the boundaries of maximum speaker diaphragm excursion and voice-coil temperature during operation. If the *SOA* is set correctly, the audio engineer need not worry about speaker damage during the audio tuning process – depending on how hard the system is pushed audio might sound more or less desirable, but speaker safety is ensured if configured properly.

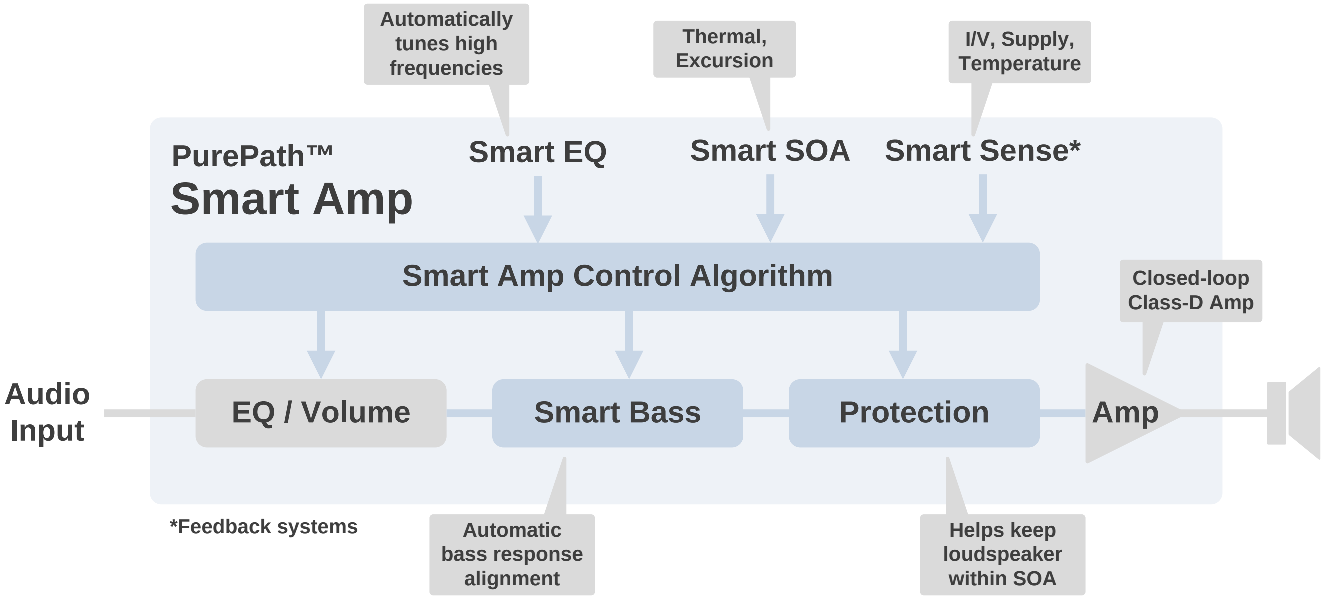
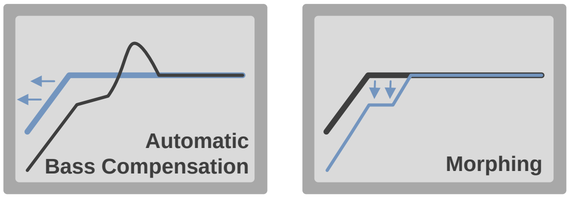


Figure 1. PurePath™ Smart Amp Block Diagram

PurePath™ Smart Amp technology enables significant sound quality and system reliability improvements while reducing component size and cost. The *PurePath™ Console 3* graphical user interface and *Learning Board* speaker characterization hardware provide simple configuration of advanced properties fully describing an audio system's acoustical, electrical, thermal and reliability capabilities and simplifying system level characterization, tuning and integration.

1.4 Smart Amp Features

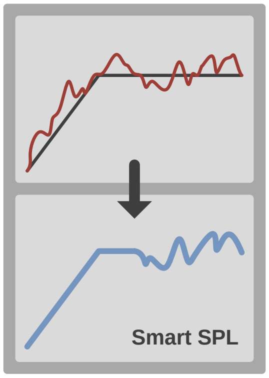


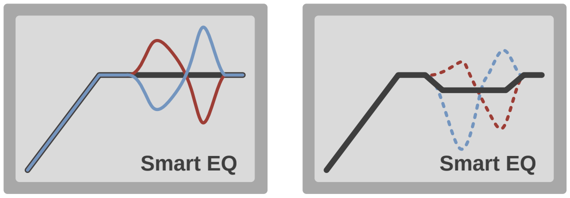
Smart Bass

Bass can easily be *extended* into any alignment automatically. As signal amplitude is increased in the bass region, Smart Bass automatically *morphs* the response to accommodate for larger excursion.

Smart SPL

High-frequency behavior of the loudspeaker diaphragm cannot be obtained electrically. Similarly, it is difficult to obtain accurate low-frequency acoustical measurements without an expensive anechoic chamber. Smart SPL automatically merges electrical and acoustical measurements to create a *full picture of the SPL response*.



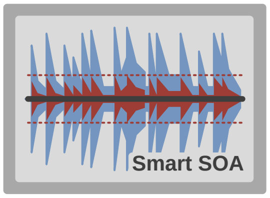


Smart EQ

Automatically tunes *high frequencies* to deliver a *flat response* or match a *target curve* in seconds. And it does it very efficiently, too.

Thermal and Excursion Protection

The Smart Amp algorithm understands the thermal and excursion limitations of the speaker. This allows to drive it at peak levels much louder than conventional amplifiers while keeping the voice coil temperature and excursion within the specified limits. This results in louder audio playback.



1.5 Smart Amp Development Overview

The following steps summarize Smart Amp evaluation, planning, characterization, tuning and integration:

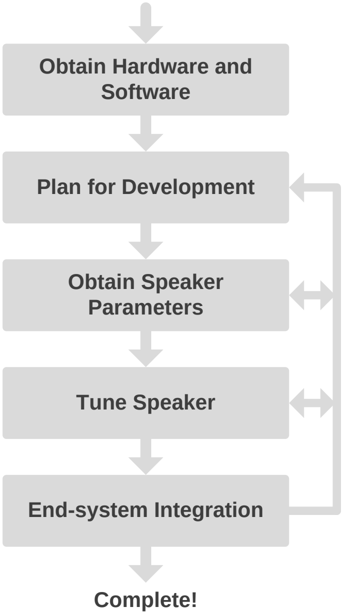


Figure 2. Smart Amp Development Overview

1. **Obtain Hardware and Software** – Speaker characterization and tuning are performed using the *PurePath™ Console 3* software. The *TI Learning Board* and the *Smart Amp Target EVM* are needed in order to fully evaluate and develop with *Smart Amp*. Refer to Section 1.6 for details.

2. **Plan for Development** – Developing *Smart Amp* based systems for the first time can be different than working with conventional amplifiers. Information obtained during the speaker characterization process often leads to changes to the speaker or enclosure to maximize output and quality. See Section 1.7 for tips on development.

3. **Obtain Speaker Parameters** – The next step is to understand the characteristics of the speaker to be tested. This is covered in Chapter 3. Once a speaker is characterized, the ppc3 file obtained from this step will be used on the next step.

4. **Tune Speaker** – Once the speaker data is obtained, a speaker can be tuned using the *Learning Board* or the *Target EVM* by importing a ppc3 file. Refer to Chapter 4.

5. **End-system Integration** – Smart Amp fundamentally shifts how audio systems are designed. Using the Smart Amp tool set, a designer gathers an in-depth understanding of speaker electro-mechanical, thermal and acoustic parameters. Based on these parameters, Smart Amp algorithms deliver high peak voltage and current to the speaker while protecting the speaker from excessive heat or movement. Increased voltage and current levels lead to changes in the system power design. For these reasons it is important to understand the power supply requirements early in the design. This is covered in Chapter 5. Software implementation is discussed in Chapter 6.

1.6 What's Needed

The table below shows the hardware and software requirements organized by configuration type (e.g. 2.0, 2.1 or 4 channel). All the configurations require the Learning Board for characterization. The Target EVM is required to generate code to be implemented in the end-system.

Table 1. Equipment and Software

	2 Channel Mid-Power Smart Amp	**2.1 Channel Mid-Power Smart Amp**	**4 Channel Mid-Power Smart Amp**
Hardware Tools (1.6.1)			
PurePath™ Smart Amp Learning Board	x	x	x
PurePath™ Console Motherboard	x	x	x
Target EVM (1.6.1)			
2 Channel TAS5766M EVM (TSSOP)	x		
4 Channel TAS5766M EVM (QFN)		x	x
Miscellaneous Equipment (1.6.2)			
Weigh Scale	o	o	o
Adhesive Putty	o	o	o
Metric Ruler	o	o	o
24V Power Supply	x	x	x
Microphone	p	p	p
Software Tools (1.6.1)			
PurePath™ Console 3	x	x	x

***x** = required, **o** = optional, **p** = required for Passive Radiator / Smart SPL*

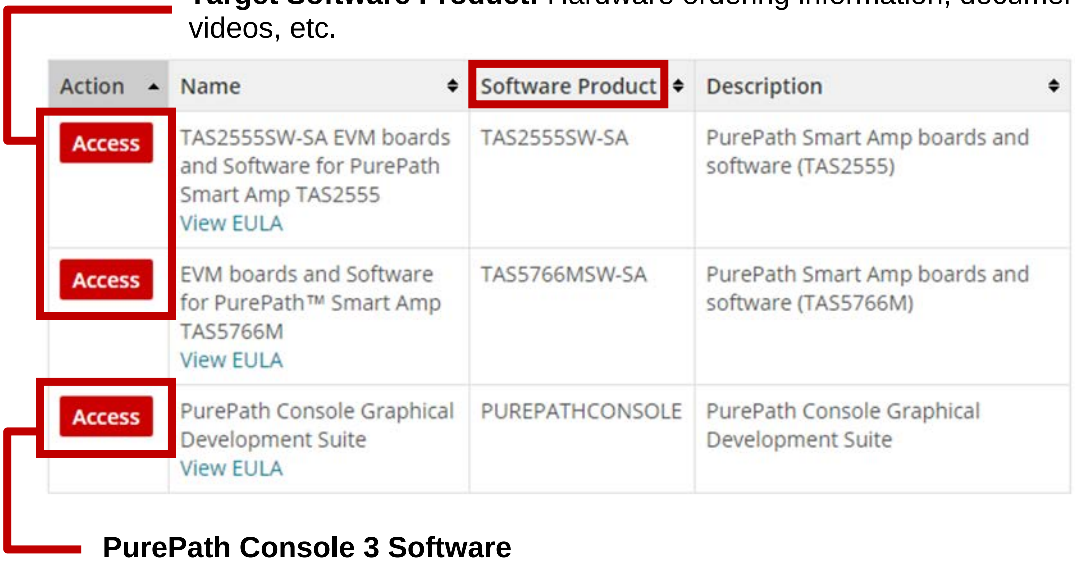
1.6.1 *Obtaining TI Hardware and Software*

TI approval is required to order Smart Amp-related hardware and software. Click the More Info button in the Target EVM web site to begin the approval process. Once approved, the user will receive an email confirming access to the mySecureSoftware web site and hardware ordering information.

The PurePath™ Console 3 software can be downloaded from the *PUREPATHCONSOLE Software Product* page in mySecureSoftware, as shown below. There is no need to download separate plugins – Smart Amp Apps will automatically be available within PurePath™ Console 3 once the user is given access to the proper Software Product (e.g. *TAS5766MSW-SA*, etc.).

The *Target Software Product*, as shown in Figure 3, contains hardware ordering information, documentation, videos, etc.

Target Software Product: Hardware ordering information, documentation, videos, etc.



Action	Name	Software Product	Description
Access	TAS2555SW-SA EVM boards and Software for PurePath Smart Amp TAS2555 View EULA	TAS2555SW-SA	PurePath Smart Amp boards and software (TAS2555)
Access	EVM boards and Software for PurePath™ Smart Amp TAS5766M View EULA	TAS5766MSW-SA	PurePath Smart Amp boards and software (TAS5766M)
Access	PurePath Console Graphical Development Suite View EULA	PUREPATHCONSOLE	PurePath Console Graphical Development Suite

PurePath Console 3 Software

Figure 3. mySecureSoftware Page

Please contact audio_software@ti.com for any issues related to software access.

1.6.2 Miscellaneous Equipment

In order to perform a full characterization the following hardware is recommended. Some of these items are not needed if loudspeaker parameters are known.

Table 2. Miscellaneous Equipment

Equipment	Description
Weight Scale	20g in 1mg increments preferred. The *Gemini-20 Portable Milligram Scale* by *American Weigh* is recommended. Not required if the force factor Bl is provided by the speaker manufacturer.
Adhesive Putty	Similar to *Blu Tack* or *Removable Adhesive Putty* by *Scotch*. Not required if the force factor Bl is provided by the speaker manufacturer.
Metric Ruler	Size depends on loudspeaker diaphragm area (S_d). Also used to measure port area (S_p). Plastic is preferred to prevent the magnet from pulling the ruler. Not required if S_d and S_p parameters are known.
24V Power Supply	A lab power supply capable of providing peak power to the speakers. A 24V, 6A supply addresses most circumstances. An external AC/DC supply could be used as well, such as the *MENB1100A2403F01* by *SL Power Electronics*.

A microphone is required for *Smart SPL* and characterizing *passive radiator* systems. Below is a list of microphones that have been tested for Smart Amp.

Table 3. Microphones Tested for Smart SPL and Passive Radiator Systems

Model	Manufacturer	Type
Mic: CME-1538-100LB	CUI, Inc.	Electret Condenser (requires external 1.8V bias)
Capsule: 4938 Preamp: Type 2670	Brüel & Kjær	¼” Diameter Pressure Field
Mic: EMM-6	Dayton Audio	½” Diameter Condenser (tested with Shure X2u USB Preamp)
Mic: iMM-6	Dayton Audio	Condenser (requires external 1.8V bias)

1.7 Things to Consider During Development

Characterization and tuning speakers provide previously unavailable information on the loudspeaker performance and how to improve it. Most customers find this leads to changes in the system. These changes to improve performance can affect the timeline if this isn't taken into account up front.

1.7.1 Leading up to Engineering Validation Test (EVT)

Do

- Create an acoustic mock-up of the audio system and use the *Learning Board* to start characterization and tuning.
- Request working/non-working prototypes of systems for use as acoustic test platforms.
- Start characterization as early as possible.
- Audition speakers in the acoustic mock up.
- Work out how the Smart Amp code will be downloaded during start up and how much time is needed to boot Smart Amp.
- Set clear performance targets.
- Use characterization and tuning results to work with speaker vendor to improve the sound or performance of transducers.

Do not

- Wait for first board build to start speaker characterization and tuning prior to EVT.

1.7.2 Leading up to Design Validation Test (DVT) and Mass Production (MP)

Do

- Start characterization and tuning of EVT systems as early as possible.
- Use EVT characterization and tuning results to work with speaker vendor to improve the sound or performance of transducers.
- Modify the EVT and DVT prototypes with external switches and wires so that the Smart Amp EVM can be connected to it.
- Set internal reviews of the tuning.
- To speed up response time, aim to provide common working prototypes to TI and speaker vendors, if necessary.
- Work to 'freeze' the acoustic elements by DVT. Acoustic freezes allow adequate time for fine tuning the sound for MP.

2 Software Installation

The following software is required to run Smart Amp.

1. PurePath™ Console 3
2. MATLAB Runtime

Smart Amp-related Apps are downloaded directly from PurePath™ Console 3 using its App Center (refer to Section 2.3).

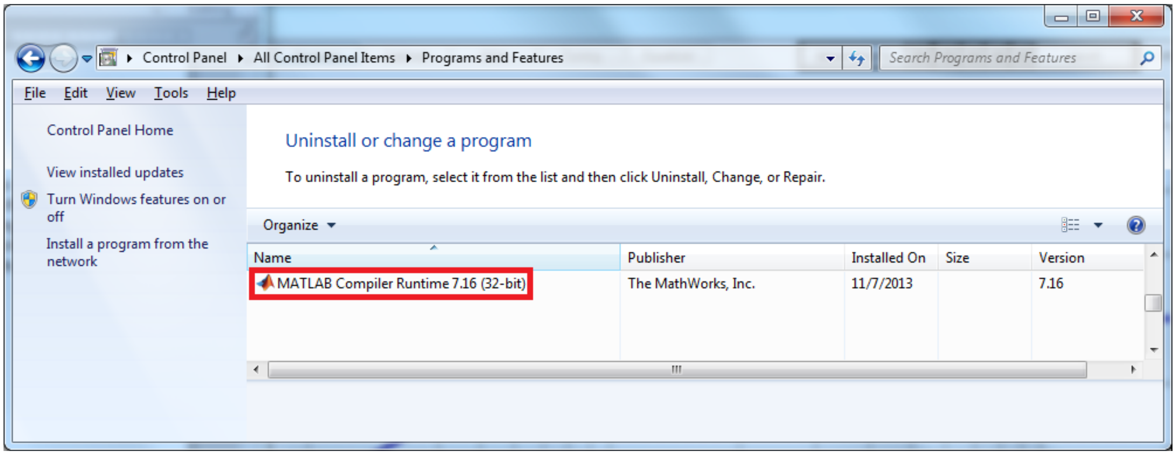
2.1 PurePath™ Console 3 Installation

Refer to the PurePath™ Console 3 documentation for installation details.

2.2 MATLAB Runtime Installation

The Smart Amp software requires the MATLAB Compiler Runtime (MCR) 7.16 engine in addition to the PurePath™ Console 3 installation to run. The installer for MCR should automatically install when installing *PurePath Console 3*. After installation, you may check the list of installed programs (On Windows 7, this is accessible via Control Panel > "Programs and Features") to verify that "MATLAB Compiler Runtime" is installed.

Refer to Appendix C for related troubleshooting information.


Control Panel ▸ All Control Panel Items ▸ Programs and Features
Search Programs and Features
File Edit View Tools Help
Control Panel Home
View installed updates
Turn Windows features on or off
Install a program from the network
Organize ▾

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Name	Publisher	Installed On	Size	Version
MATLAB Compiler Runtime 7.16 (32-bit)	The MathWorks, Inc.	11/7/2013		7.16

Figure 4. Verification of MATLAB Compiler Runtime Installation

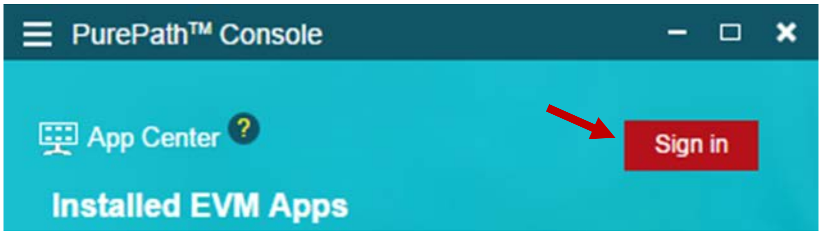
2.3 Getting Smart Amp Apps

PurePath™ Console 3 *Apps* are downloaded using the *App Center* .

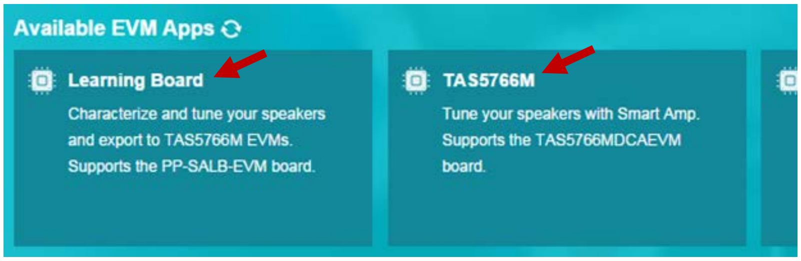
Follow these steps to download Smart Amp specific Apps:

1. Download the PurePath™ Console 3 software from the *PUREPATHCONSOLE* Software Product in mySecureSoftware. Refer to the PurePath™ Console 3 documentation for installation details.

2. Once PurePath™ Console 3 is installed, click the *Sign in* button as shown to the right.



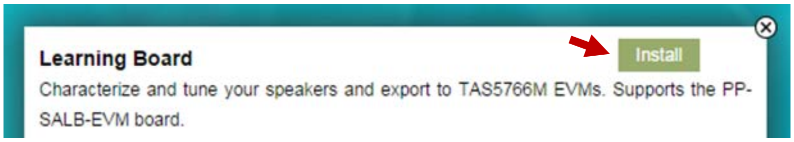
3. Smart Amp-related Apps will now appear in the *Available EVM Apps* section. If no Smart Amp Apps appear refer to Section 1.6.1.



Learning Board
Characterize and tune your speakers and export to TAS5766M EVMs. Supports the PP-SALB-EVM board.

TAS5766M
Tune your speakers with Smart Amp. Supports the TAS5766MDCAEVM board.

4. Click the desired App(s) to open the information window and click *Install*.



Learning Board
Characterize and tune your speakers and export to TAS5766M EVMs. Supports the PP-SALB-EVM board.

5. The App(s) will now appear in the *Installed EVM Apps* section.

3 Loudspeaker Characterization

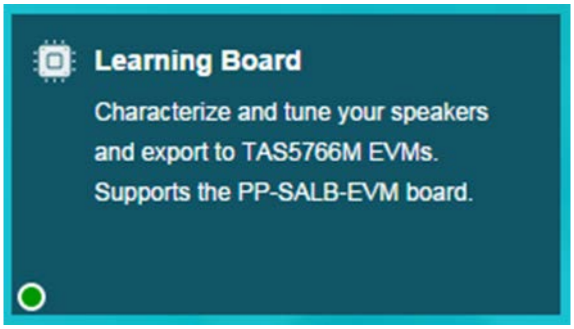
The main objective of the loudspeaker characterization is to obtain the *electro-mechanical* and *thermal* parameters and establish the safe operating area (SOA) of the loudspeaker system. The electro-mechanical and thermal parameters are obtained using the Learning Board.

The *Learning Board App* has a step-by-step wizard that guides the user through the entire loudspeaker characterization process.

3.1 Characterization Process

To perform a loudspeaker characterization:

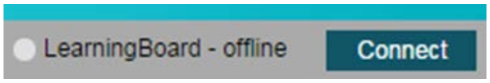
1. Connect the *Learning Board* to your PC using the USB cable.
    a. Provide a power supply.
    b. Do not connect the speakers yet.
2. In PurePath™ Console 3, open the **Learning Board** App and select **New**.



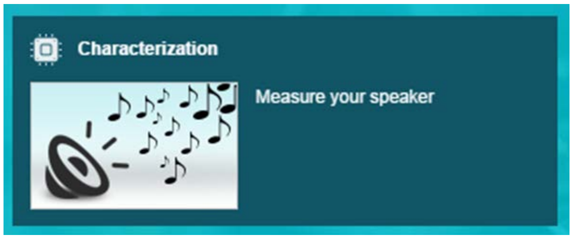
Learning Board

Characterize and tune your speakers and export to TAS5766M EVMs. Supports the PP-SALB-EVM board.

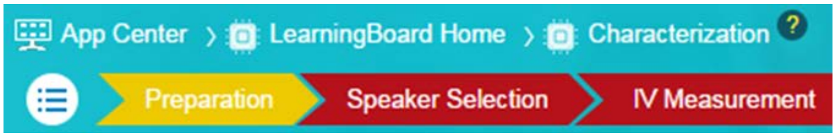
3. If the Learning Board is shown as offline, click **Connect**.



4. Click **Characterization**.



5. Follow the step-by-step **wizard** until the characterization is complete.



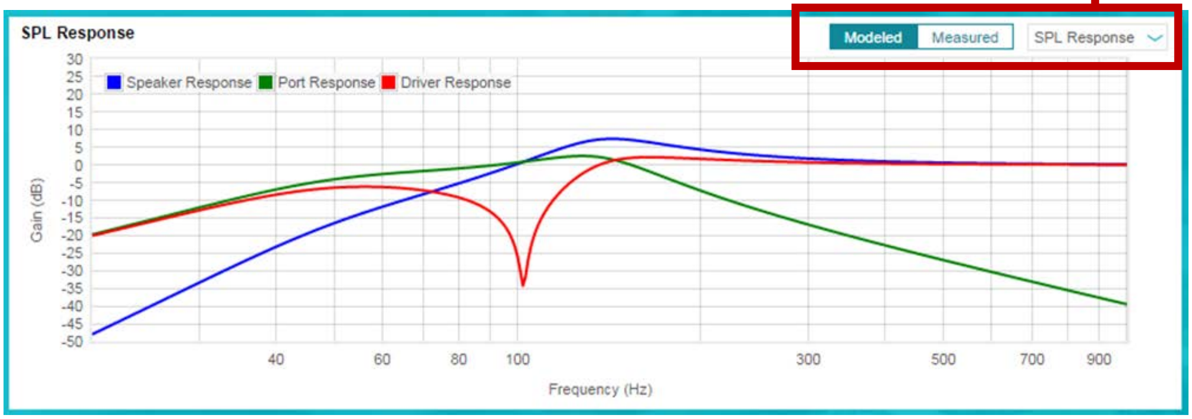
6. Once complete, the **Characterization Summary** page will be shown.

3.2 Characterization Summary Page

The *Characterization Summary* page shows the results of the loudspeaker characterization.

To verify the loudspeaker plots, use the controls to the top of the graph:

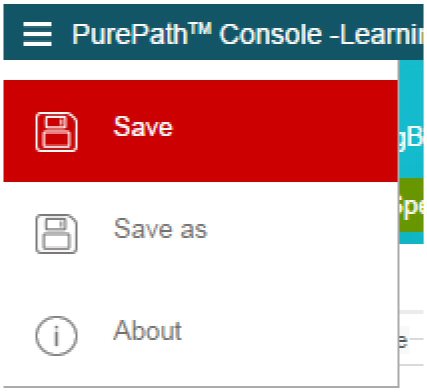
Selects between **Modeled** SPL and Excursion or **Measured** Impedance, Temperature and SPL plots.



Driver and enclosure parameters are also shown as well as the established safe operating area (SOA).

If desired, click the button to redo the characterization. The button will bring back the *Characterization Summary* page.

3.3 Saving a Characterization



The characterization data can be saved by clicking the ≡ button at the *Title Bar* and selecting *Save*. This will output a .ppc3 file.

This .ppc3 file can later be imported into the Target EVM App for tuning, as explained in Section 4.1.

4 Smart Amp Tuning

Tuning is a process involving both subjective and methodical approaches. This section provides guidelines to help establish a baseline to achieve the best possible tuning. The main objectives of the (iterative) audio tuning process, also referred to as 'voicing', are:

- Improve bass performance using **Smart Bass** controls (Section 0) by adjusting:
  - Bass Enhancement (Section 4.2.1)
  - Morphing Control (Section 4.2.2)
  - Harmonic Bass Alignment (Section 4.2.3)
- Improve high frequency response using **Smart EQ** (Section 4.3).

The Smart Amp Tuning Process is summarized in Figure 5.

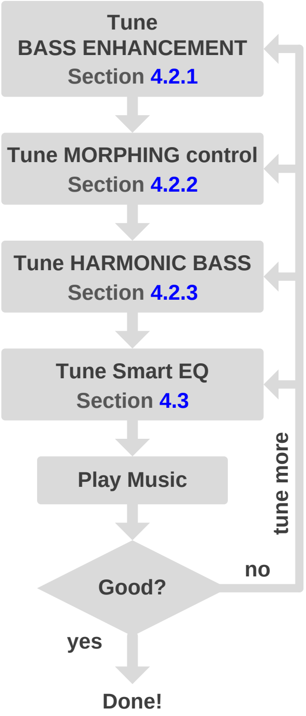


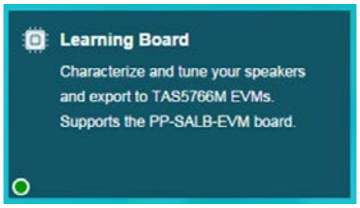
Figure 5. Smart Amp Tuning Process

4.1 Tuning Preparation

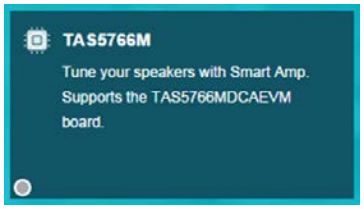
Tuning is performed using the Learning Board or the Target EVM.

1. **Connect** the *Learning Board* or *Target EVM* to your PC using the USB cable.
   a. Provide a power supply (matching the one to be used in the final system).
   b. **Connect** the loudspeaker.

2. In PurePath™ Console 3, open the **App** that corresponds to the board that was connected (*Learning Board* or *Target EVM*).

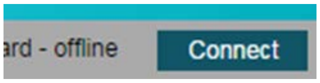


Characterize and tune your speakers and export to TAS5766M EVMs. Supports the PP-SALB-EVM board.

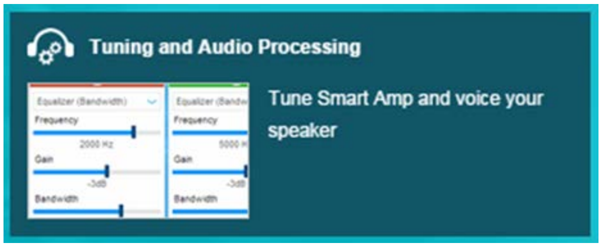


Tune your speakers with Smart Amp. Supports the TAS5766MDCAEVM board.

3. If the board is shown as offline, click **Connect**.



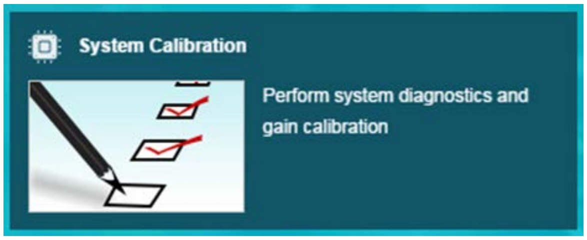
4. Click **Tuning and Audio Processing**.



Tune Smart Amp and voice your speaker

5. **Import** the characterization data that was obtained during the *Characterization* process (Chapter 3) by clicking the Import button.

6. Click and perform a **System Calibration**. This process ensures that the Smart Amp algorithm is properly scaled based on the amplifier output gain.

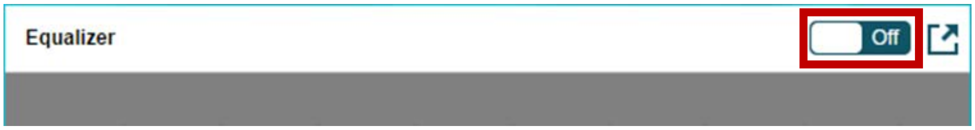


Perform system diagnostics and gain calibration

4.2 Smart Bass Tuning

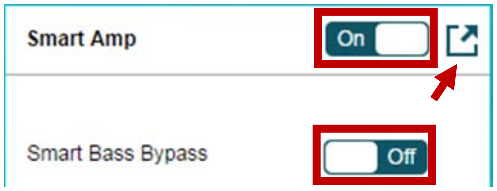
During the characterization process, the low-frequency SPL model of the loudspeaker was obtained. Based on this model, the response is automatically optimized to match popular loudspeaker alignment types (e.g. Butterworth, Linkwitz-Riley, etc.). This allows the audio engineer to focus on choosing the desired sound with just a few clicks.

When tuning *Smart Bass* for the first time, it is best to first disable the *Equalize*r (in the › Audio processing page):



Equalizer: Off

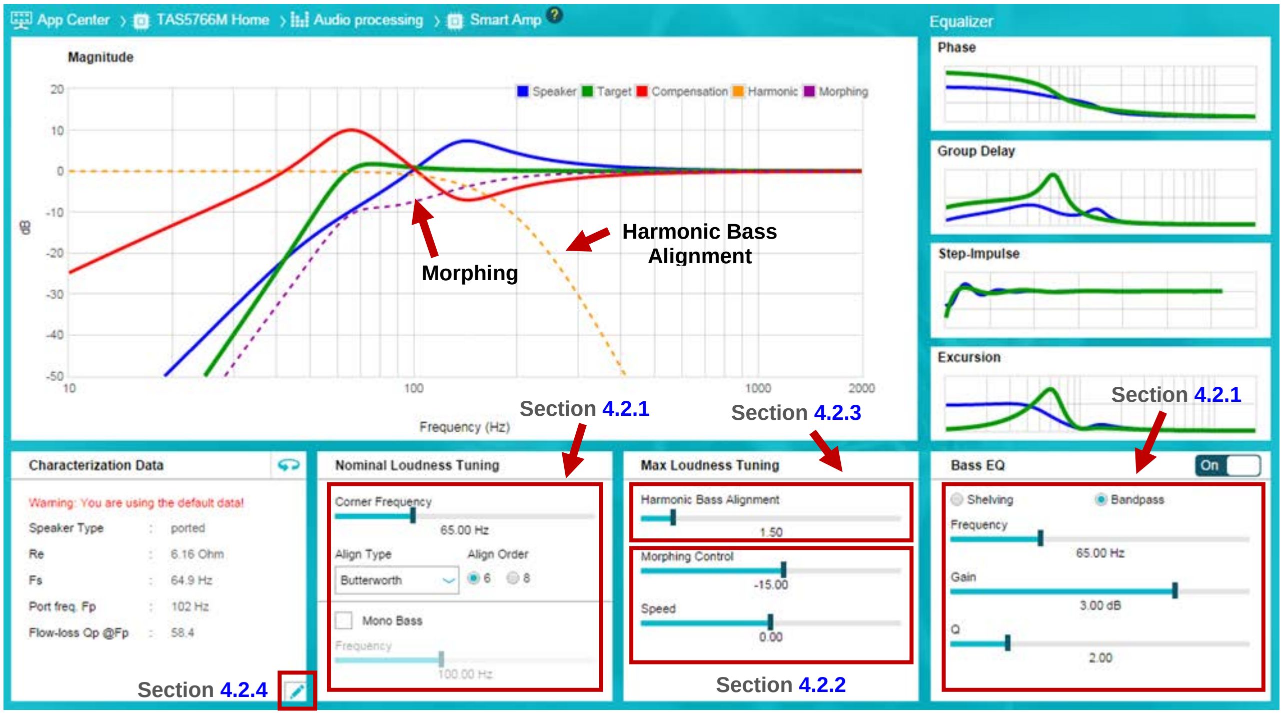
To start tuning *Smart Bass*, enable Smart Amp and click the Expand symbol ↗, as shown to the right.



Smart Amp: On

Smart Bass Bypass: Off

The *Smart Amp Tuning* page has all the controls needed for *Smart Bass* tuning. Several plots (such as *Excursion*) are provided as an aid to the tuning process.



Characterization Data

Warning: You are using the default data!

Speaker Type	:	ported
Re	:	6.16 Ohm
Fs	:	64.9 Hz
Port freq. Fp	:	102 Hz
Flow-loss Qp @Fp	:	58.4

Section 4.2.4

Nominal Loudness Tuning

Corner Frequency: 65.00 Hz

Align Type: Butterworth

Align Order: ● 6 ○ 8

□ Mono Bass

Frequency: 100.00 Hz

Max Loudness Tuning

Harmonic Bass Alignment: 1.50

Morphing Control: -15.00

Speed: 0.00

Section 4.2.2

Bass EQ: On

○ Shelving ● Bandpass

Frequency: 65.00 Hz

Gain: 3.00 dB

Q: 2.00

Figure 6. Smart Amp Tuning Page

4.2.1 Bass Enhancement (Low Volume Tuning)

The main objective is to maximize the bass response as much as possible and tune the bass as desired. During this phase it is important to listen at low volume levels only – this is to ensure that thermal and mechanical protection systems do not kick-in. Ensure that the **Morphing Control** (Section 4.2.2) is set at max during this phase.

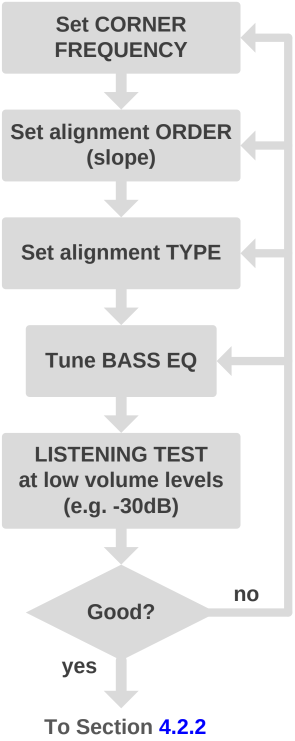
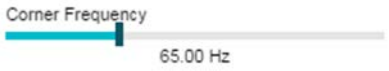


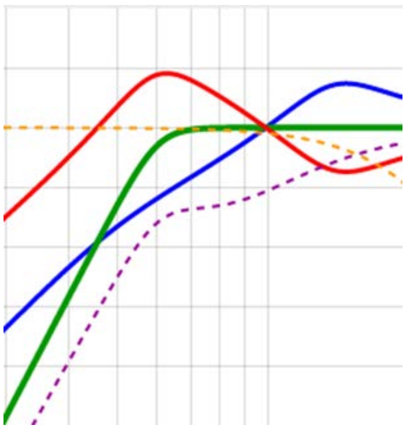
Table 4. Bass Enhancement Parameters

Field	Description
Corner Frequency	-3dB point of the target response. Smaller speaker drivers should use higher corner frequency.
Order (Slope)	Determines the sharpness of the roll-off towards lower frequencies. For lower corner frequency, choose higher order.
Type	Selects the alignment type.
Bass EQ	To further enhance bass, as needed, at low volume settings

Corner Frequency

The **Corner Frequency** indicates the -3dB point of a flat response target (indicated by the green curve). Selecting a proper Corner Frequency is important for the overall performance of the speaker system. If the cutoff is set too high, the speaker will have limited bass response. If set too low, energy will be wasted trying to drive frequencies that the speaker won't be able to reproduce and the excursion protection system will be overly active.

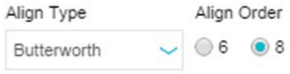

Corner Frequency
65.00 Hz



A 'getting started' setting is described in this section. It is recommended to do a series of listening tests while adjusting settings.

- Adjust **Corner Frequency** while watching the compensation (red curve) in the response plot window.
- Targeting between a 10dB to 20dB compensation (red curve) often provides the best results.
- Do not exceed the 20dB line (at least initially).
- In some situations it can be advantageous to leave some gain headroom in favor of a general Bass EQ (later step).

Alignment Order and Type



Align Type: Butterworth

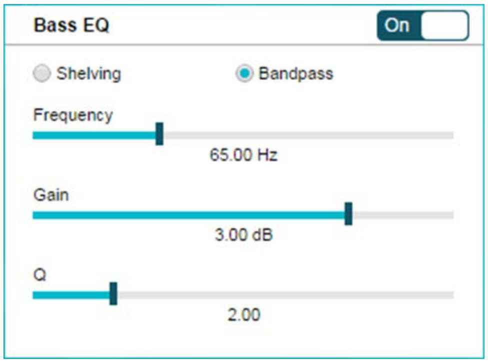
Align Order: ○ 6 ● 8

The **Order and Type** determine the bass roll-off. In other words, it determines what occurs below the corner frequency.

A high-order roll-off cuts bass faster saving power and limiting speaker excursion that won't produce much SPL. Likewise, Type has significant influence on the SPL and energy below the corner frequency.

- Select a higher order if the speaker handles excursion poorly.
- Select a lower order to leave small amounts of low-frequency content in the signal.
- Butterworth is suitable for most applications. For ported / passive radiator systems that can reproduce 60-80Hz, a Chebyshev alignment works well.

Bass EQ (Optional)



Bass EQ: On

○ Shelving ● Bandpass

Frequency: 65.00 Hz

Gain: 3.00 dB

Q: 2.00

It may be desired, for an even richer bass experience, to extend the target curve (green line) using a standard **Bass EQ** that can be shaped in amplitude, frequency and Q. Bass EQ works well in concert with the Morphing feature that dynamically cuts Bass as excursion and thermal headroom drops with increase in loudness. It is recommended to tune the EQ setting at low volume initially. In subsequent steps (e.g., Morphing adjustment) the Bass EQ setting may be revisited, especially if an aggressive boost has been selected initially.

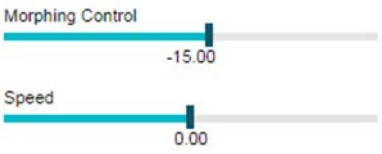
1. Set the **Bass EQ** to *On*.
2. Select **Bandpass**.
3. Select a Bass EQ **Frequency** within 1x to 2x that of the **Corner Frequency** setting.
4. Adjust **Gain** up, but keep the compensation (red curve) below 20dB (at least initially).
5. Tweak the **Q** value. Often, values between 0.5 and 3 work well.

Mono Bass Frequency

When enabled, **Mono Bass Frequency** is the cross over point below which the left and right channels are mixed, effectively creating a mono signal below that frequency on both channels. In systems with a common cabinet for left and right drivers this control ensures against the possibility of a drop in the bass level due to phase mismatch between the left and right channel signals.

4.2.2 Morphing Control (Mid- to High- Volume Tuning)

Morphing determines the headroom dependent balance between Bass Enhancement and Harmonic Bass Alignment. As excursion and thermal headroom drops with increase in music loudness, the **Morphing** feature gradually and dynamically reduces bass.

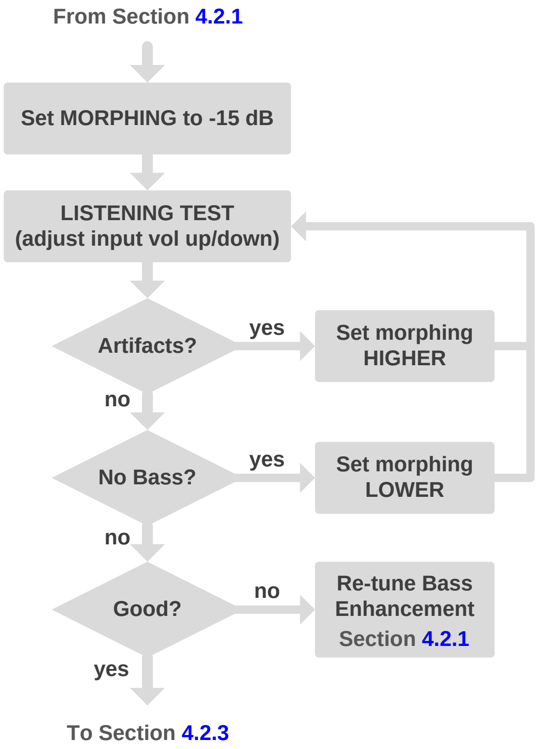


Morphing Control -15.00

Speed 0.00

Depending on **Speed** and **Harmonic Alignment** settings, some residue of bass harmonics might remain in the frequency spectrum creating a psychoacoustic bass enhancement effect.

- Audible artifacts indicate too high Morphing setting.
- Little but clean bass could indicate that the Morphing setting is too low.



NOTE:
This is an iterative process! It is important to listen to different types of music and at several volume settings (listening levels).

Morphing Speed (Optional)

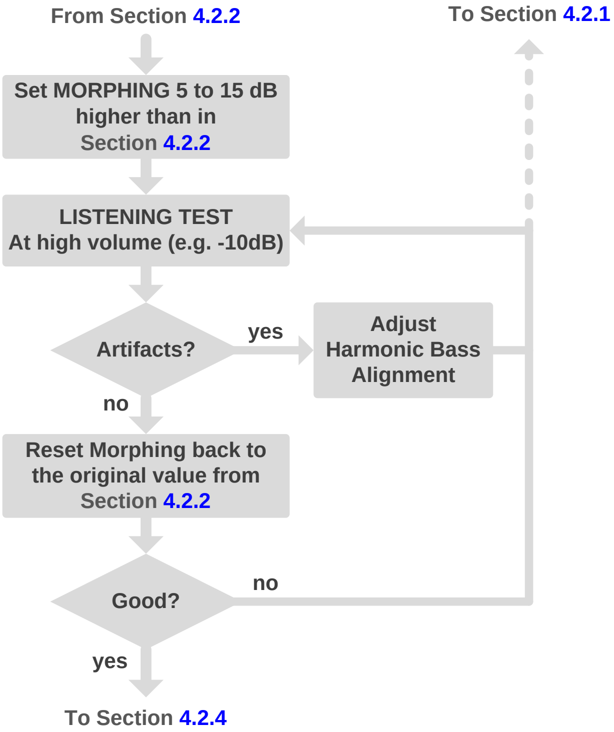
The **Morphing Speed** control determines the aggressiveness on which the Smart Amp algorithm adapts to a change of headroom.

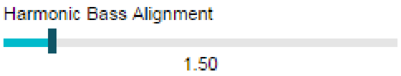
Speakers react very differently to morphing speed and there is unfortunately no universal guideline for how to tune this for best setting. It is recommended to experiment. This setting may also be left at the default value (0).

- Listen to different music types at moderate to high volume levels.
- Listen for audible artifacts such as:
    - **Bass region:** distortion, especially with high transients (such as a kick drum).
    - **Mid/high range**: distortion, modulation artifacts.

4.2.3 Harmonic Bass Alignment (Mid- to High- Volume Tuning)

The **Harmonic Alignment** control determines the aggressiveness of the excursion protection algorithm as speaker headroom is reduced. Some speakers sound great with an aggressive setting (high value) where other speakers, typically of lower quality, will sound harsh and distorted and will require less aggressive setting (lower value).


From Section 4.2.2
To Section 4.2.1
Set MORPHING 5 to 15 dB higher than in Section 4.2.2
LISTENING TEST At high volume (e.g. -10dB)
Artifacts?
yes
Adjust Harmonic Bass Alignment
no
Reset Morphing back to the original value from Section 4.2.2
Good?
no
yes
To Section 4.2.4

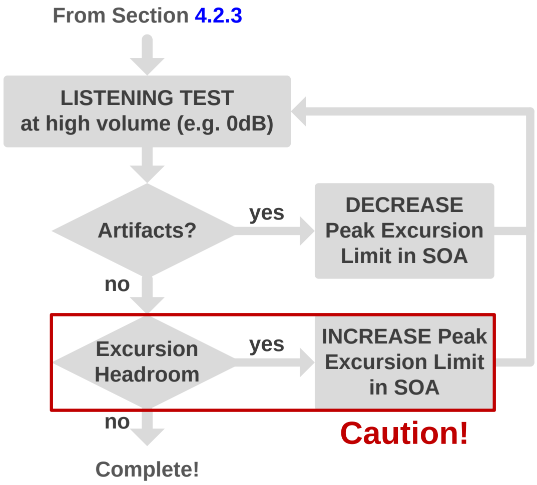

Harmonic Bass Alignment

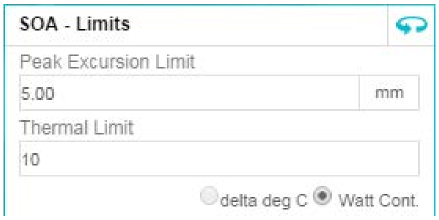
1.50

- Listen to different music types at moderate to high volume levels.
- Listen for audible artifacts such as:
  - **Bass region:** distortion, especially with high transients (such as a kick drum).
  - **Mid/high range**: distortion.

4.2.4 Excursion Tuning (Optional)

Depending on the type and quality of the speaker (as characterized in the measurement phase), the **Peak Excursion** SOA setting sometimes needs a post-audio-tuning adjustment for best sound quality. As a typical speaker approaches its Xmax the THD tends to rise quickly. This behavior can have an amplifying effect on artifacts.


From Section 4.2.3
LISTENING TEST at high volume (e.g. 0dB)
Artifacts?
yes
DECREASE Peak Excursion Limit in SOA
no
Excursion Headroom
yes
INCREASE Peak Excursion Limit in SOA
no
Caution!
Complete!


SOA - Limits

Peak Excursion Limit

5.00 mm

Thermal Limit

10

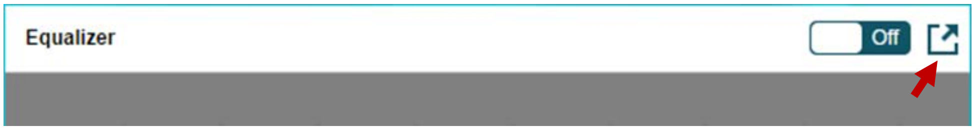
○ delta deg C ● Watt Cont.

- In the *Characterization Page*, change the peak excursion limit value while you listen to audio.
- Reduce it if there are Artifacts at high volume settings.

4.3 Smart EQ

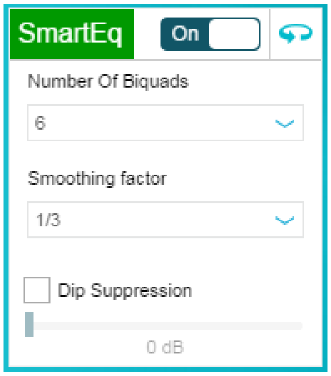
Smart EQ automatically tunes high frequencies to deliver a flat response or match a target curve. It is required measure SPL with a microphone during the characterization process in order to use the Smart EQ feature.

To configure *Smart EQ*, go to the › Audio processing page and click the Expand symbol next to the Equalizer section, as shown below.



Equalizer Off

1. Set **Smart EQ** to *On* to enable it. The compensation curve is calculated based on the SPL measurement obtained during the characterization process.



SmartEq On

Number Of Biquads

6

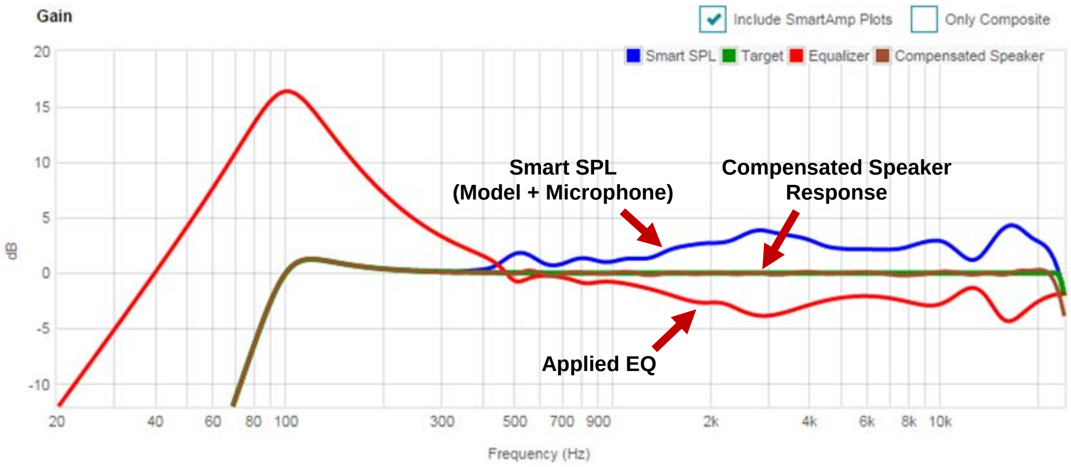
Smoothing factor

1/3

□ Dip Suppression

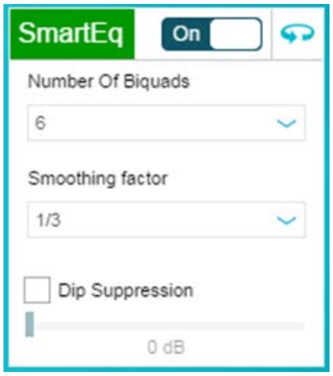
0 dB

2. A plot similar to the one below should appear. The Smart EQ algorithm effectively compensated to create a flat response at high frequencies. Because *Include Smart Amp Plots* is checked, the bass boost applied during the Smart Bass tuning is also shown.



The **SmartEQ** section has two panels that can be swapped by clicking the ⟲ button. The following controls are available:

- Number of Biquads
- Smoothing Factor
- Dip Suppression
- Spline Region
- SPL Offset

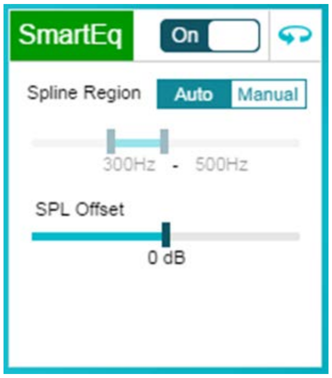

SmartEq On

Number Of Biquads: 6

Smoothing factor: 1/3

□ Dip Suppression

0 dB


SmartEq On

Spline Region: Auto Manual

300Hz - 500Hz

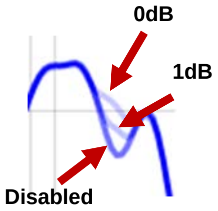
SPL Offset

0 dB

The **Number of Biquads** dropdown determines how many biquads are used to create the compensation curve. The typical value is 6 biquads.

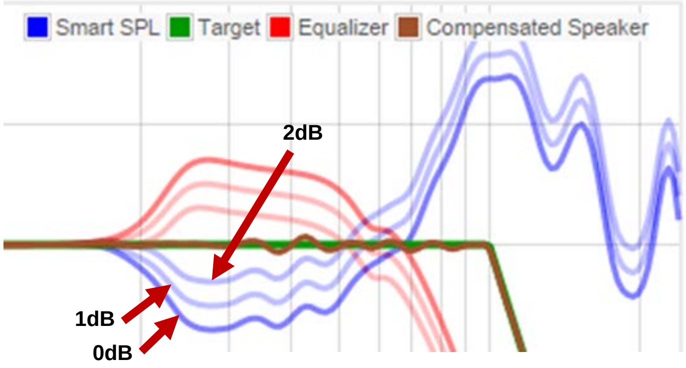
The **Smoothing Factor** smoothes the Smart SPL curve by 1/n octave (nth octave). Typical values are 1/3 and 1/6. Anything above will not provide much audible benefits, but can be used to observe a raw acoustic response.

The **Dip Suppression** feature controls how many dBs can the response dip. A 0dB setting means that dips are completely suppressed, as shown to the right. A 1dB setting means that the response is allowed to dip by 1dB.



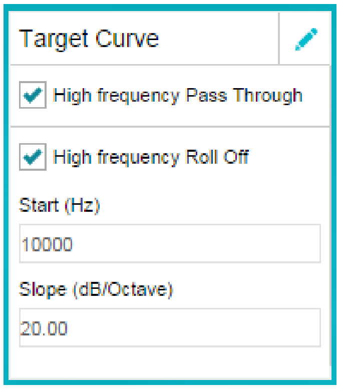
The **Spline Region** determines the frequency range where the transition from *Smart Bass* to the *Measured SPL* takes place. The end result is the *Smart SPL* curve. The **Auto** option calculates this range automatically, or can be manually changed by clicking **Manual** and selecting the desired range.

The **SPL Offset** controls how much offset is applied to the *Measured SPL* curve relative to the *Smart Bass* curve. For example, for a 2dB offset, the *Measured SPL* curve will shift up by 2dB. This can be useful to fine adjust the overall SPL response of the system.



The **Target Curve** section allows the user to create their own signature sound by focusing on the final target response.

The **High Frequency Pass Through** ignores the final dip of the Smart SPL response which would otherwise require a large amount of positive compensation that may not be audible and cause distortion.



Target Curve ✎

☑ High frequency Pass Through

☑ High frequency Roll Off

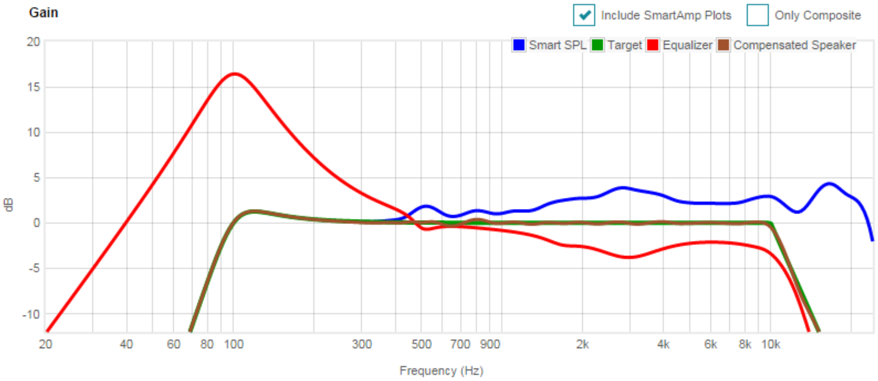
Start (Hz)

10000

Slope (dB/Octave)

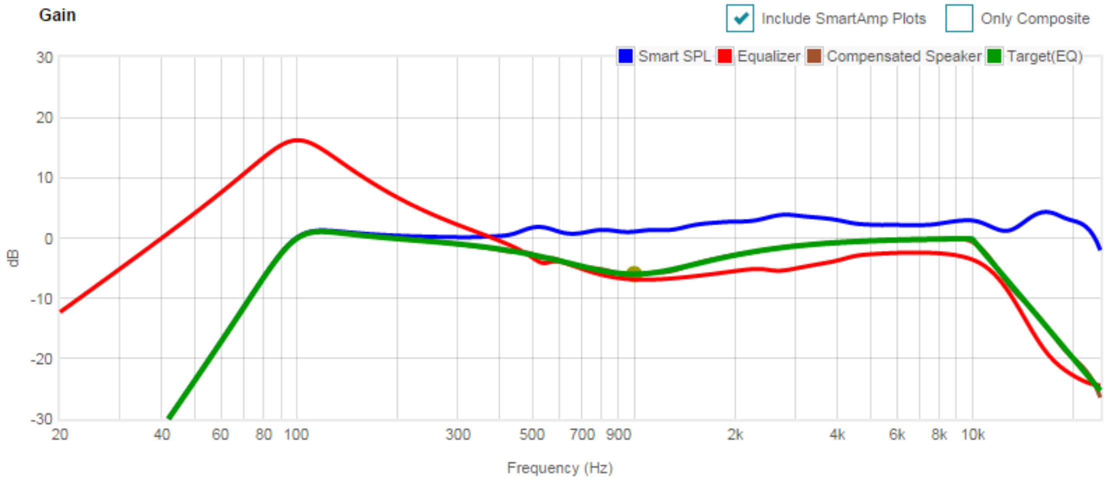
20.00

The example to the right shows a 20dB per octave rolloff at 10kHz using the **High Frequency Roll Off** feature.



☑ Include SmartAmp Plots ☐ Only Composite

A custom target can be created by clicking the ✎ button. The response below shows a 1kHz EQ filter with -6dB gain and 1kHz bandwidth. The Smart EQ algorithm will automatically calculate the proper filters to match that target.

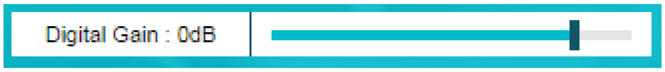


☑ Include SmartAmp Plots ☐ Only Composite

4.4 Digital Gain Control

For smaller loudspeakers it may be desired to use a *Digital Gain* setting lower than 0dB during playback at maximum volume. This is especially true when maximum SPL is achieved at much lower gain settings.

In the › Audio processing page, adjust the Digital Gain control down while playing heavily compressed music until the perceived maximum loudness starts to decrease.

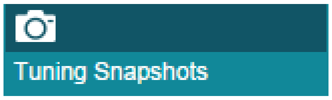


Digital Gain : 0dB

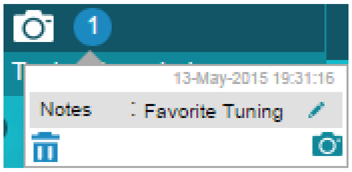
Figure 7. Digital Gain Control

Note that this gain control occurs before the Smart Amp algorithm (i.e. pre-processing). If it is discovered that the optimal Digital Gain level is below -6dB, then it may be a better option to adjust the post processing gain levels instead (requires re-calibration). This will achieve a better signal to noise ratio (SNR). Post processing considerations are discussed in Section 5.4.

4.5 Snapshots



Snapshots can be used to compare between different tunings. To create a Snapshot, click 📷. A 1 will appear.



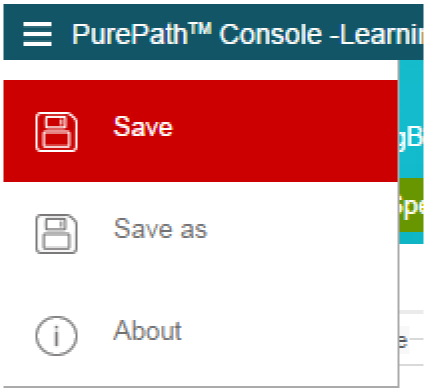
13-May-2015 19:31:16

Notes : Favorite Tuning

Snapshots can be named by clicking ✎ and deleted by clicking 🗑.

Click 📷 to replace the selected Snapshot or click 📷 to create a completely new one.

4.6 Saving a Tuning



The tuning data can be saved by clicking the ≡ button at the *Title Bar* and selecting *Save*. This will output a .ppc3 file that corresponds to the App being used.

5 Hardware Design

Smart Amp systems can operate at higher rail voltages than traditional systems while protecting loudspeakers. This is possible because Smart Amp understands the speaker characteristics. This allows control of excursion and temperature of the voice coil and the magnet while preserving the music peaks. For products where the rail voltage remains unchanged from a previous model, smaller speakers could be used to save cost and reduce form factor while preserving loudness.

If the maximum output voltage limit of a traditional system is based on the average power of a full-scale sinusoid, there is risk of voice coil overheating if a square wave is provided as an input. This is due to the fact that a square wave has 6dB higher average power than a sinusoid of the same peak amplitude as well as having the presence of higher frequency components. Conservative designs may then have to trade off sound pressure level (SPL) with reliability.

More advanced methods to control load power include the use of limiters and dynamic range compressors. These methods can protect the speaker however peaks may be clipped and/or greatly reduced, especially on source material with high peak-to-average ratios (PAR).

The graphs below are actual song clips comparing the traditional method (left) against Smart Amp (right) to control output power. The dashed lines correspond to the output limit of a traditional system. Note that the average power (P_{ave}) is increased while allowing peaks to cross the output limit.

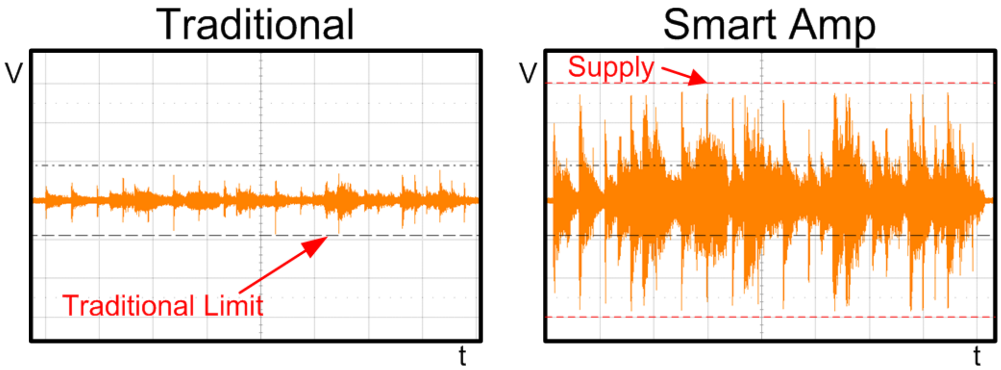


Figure 8. Audio Clip A, 22dB Peak to Average Ratio Source

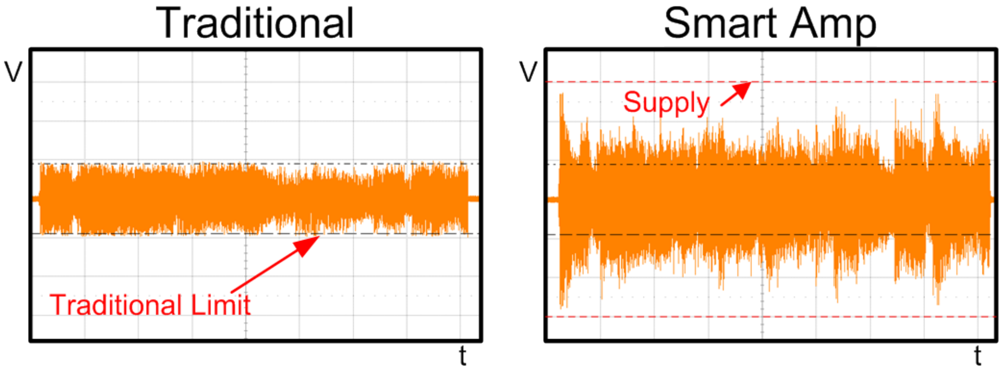


Figure 9. Audio Clip B, 9dB Peak to Average Ratio Source

5.1 Power Supply Rail Voltage

Traditionally, audio systems are designed to satisfy the power handling specification of a loudspeaker. There are several methods in which loudspeaker rated power is obtained, one of them discussed in the AES2 standard [1] [2]. Most tests involve a pink noise signal with a specified crest factor, bandwidth and roll-off over a period of time. The newest, being the AES2-2012, specifies a test signal peak-to-rms voltage ratio (or crest factor) of 12dB [2].

A traditional design would match the output power of the amplifier to the average power handling of the speaker. An amplifier maximum P_{ave} is determined with a sinusoid at a certain distortion level. For practical purposes, the rail voltage of a traditional design will be defined as below:

$$V_{dd,trad} = \sqrt{2P_{ave,spk} \times Z_L}$$

Where $P_{ave,spk}$ is the rated continuous P_{ave} of the speaker and Z_L is the rated speaker impedance. Note that the actual voltage observed at the load depends on factors such as $R_{DS(on)}$ of a Class-D amplifier among others.

As described in the previous section, Smart Amp systems can operate at higher rail voltages than traditional methods. To obtain the full benefits of Smart Amp, it is recommended to utilize a power supply rail that can provide 10 times $P_{ave,spk}$ (β = 10). The equation can be used to calculate the rail voltage for a Smart Amp system:

$$V_{dd,sa} = \sqrt{\beta \times 2P_{ave,spk} \times Z_L} = \sqrt{\beta} V_{dd,trad}$$

The table below compares the supply voltage that would be typically selected using a traditional amplifier ($V_{dd,trad}$) against the recommended supply voltage for Smart Amp ($V_{dd,sa}$) for several speaker load and power handling values. Although a β of 10 is recommended, smaller values could be used (e.g. 4) and still provide significant improvements. For the TAS5766M a 24V supply is preferred, although 12V is still considered optimal for 1W loads.

Table 5. Example Smart Amp Supply Rail Voltages for the TAS5766M

Speaker Nominal Impedance (ohms)	$P_{ave,spk}$ (W)	$V_{dd,trad}$ (V)	$V_{dd,sa}$ (V)
4	1	3	7.4 / 12 / 24
	5	6	12 / 24
	10	9	24
8	1	4	7.4 / 12 / 24
	5	9	12 / 24
	10	13	24

TIP: A 24V supply is suitable for most applications.

5.2 Power Supply Current at Audio Frequencies

An ideal power supply can deliver infinite amounts of current without a reduction in its output voltage. In reality, a power supply maximum current output depends on its series impedance. In practical audio applications, where the decoupling techniques are limited by cost and size, audio frequencies are mostly driven by the supply itself and not by decoupling/bulk capacitors. The time constants associated with audio frequencies (especially the bass and mid areas) would require a large amount of capacitance which may be deemed impractical for most applications.

Smart Amp will help protect the speaker against thermal failure associated with high currents for prolonged periods of time. This means that a large signal will be reduced in amplitude and settle into a steady level corresponding to a safe output power. The plot below shows an example $6V_{peak}$ 1 kHz signal that is attenuated to a safe level. The 24V power supply in this example has a 2-ohm series resistance and the amplifier is driving an 8-ohm load. The supply voltage drop and current for 100uF and 1000uF bulk capacitors are shown.

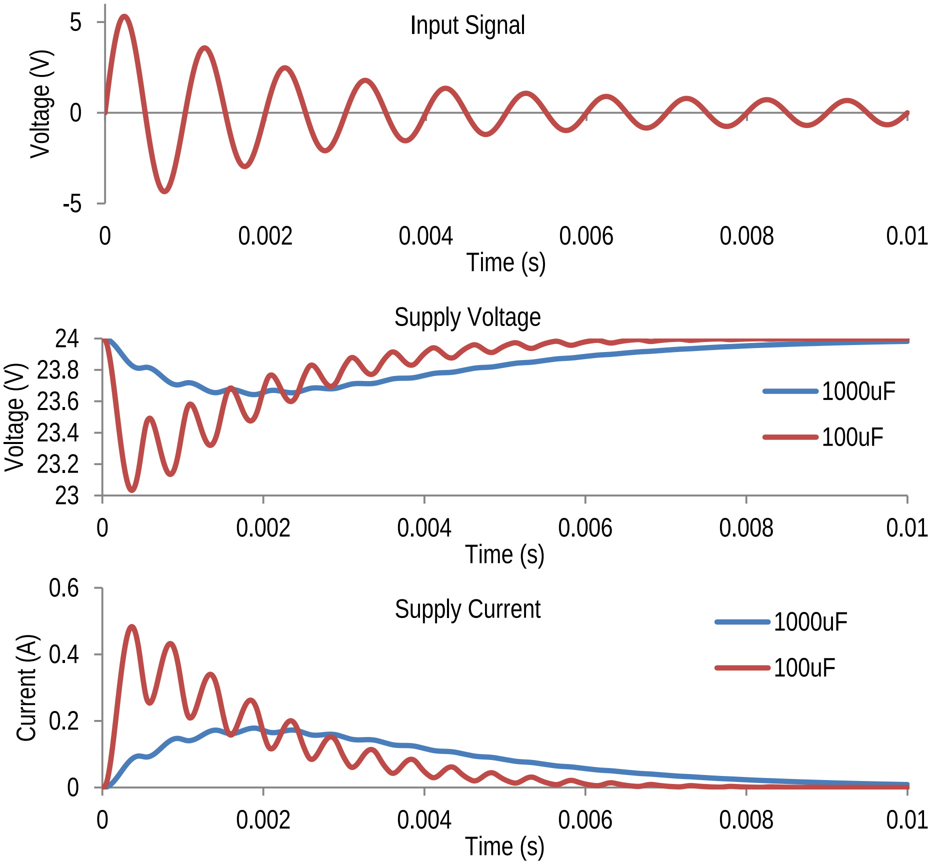


Figure 10. Effects of Decoupling for Audio Frequencies

Note that for a bridge-tied-load (BTL) amplifier, the current ripple at the supply is twice the frequency of the input signal.

A practical capacitor size will not help much with these peaks therefore it is recommended to design the power supply based on the expected peak currents. The peak current observed at the power supply can be approximated by calculating the maximum current into the load taking the amplifier efficiency into account.

$$I_{pk,in} = \frac{P_{pk,out}}{\eta \times V_{dd}} \times \#channels$$

The peak input current can also be calculated if the full-scale voltage is known, as shown below. Section 5.4 covers the output levels in more detail.

$$I_{pk,in} = \frac{V_{pk,FS}^2}{\eta \times V_{dd} \times Z_L} \times \#channels$$

The table below lists several peak supply current examples. Note that these calculations assume a resistive load. Speakers are inductive in nature and may cause higher peak currents as discussed in Section 5.3.

Table 6. Peak Input Current Examples (One Channel)

Speaker Nominal Impedance Z_L (ohms)	Full-scale Peak Voltage $V_{pk,FS}$ (V)	Efficiency η	Supply Rail V_{dd} (V)	Supply Current $I_{pk,in}$ (A)
4	3	0.92	12	0.2
	6	0.92	12	0.82
	12	0.92	24	1.63
	18	0.92	24	3.67
8	3	0.92	12	0.1
	6	0.92	12	0.41
	12	0.92	24	0.82
	18	0.92	24	1.83

The plot below shows measured power supply current into a TAS5766M. The dashed line corresponds to 6.4A delivered into two 3.76-ohm loads using a 12V supply.

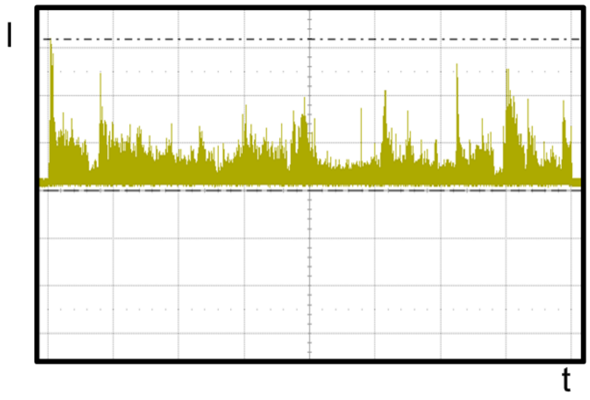


Figure 11. Power Supply Current Measurement

5.3 Power Supply Current at Higher Frequencies

Power supply current is limited by its inductance at higher frequencies, which is due to a combination of the power supply itself, passives and PCB layout. The PWM output stage of a Class-D amplifier switches at a high frequency (>100 kHz) which may cause the power supply voltage to drop if not decoupled properly affecting the amplifier and other circuits as well.

An LC filter at the output of a Class-D amplifier helps reduce high frequency switching current through the load and from the power supply while increasing efficiency. Some applications restrained by cost/space may use ferrite beads instead of the traditional LC filter, taking into advantage the series inductance of the loudspeaker. The design of Class-D filters is discussed in [3]. An LC filter is a second order filter, damped by the load impedance. The filter values are usually calculated based on the DC resistance of the loudspeaker. However, the filter may become underdamped due to the corner frequency of the filter and the inductance of the loudspeaker. An underdamped response will manifest as a dip in the impedance which may cause currents higher than the ones calculated for the rated loudspeaker impedance. For this reason, it is recommended to implement a snubber circuit in parallel to the filter capacitors. The device evaluation module (EVM) and datasheet are the best reference for these filters.

The PWM signal must have good integrity in order to provide a good fidelity output signal. As mentioned previously, the power supply and PCB layout may not be capable of delivering the initial switching current required by the PWM signal. The worst-case scenario would involve calculating the bulk capacitor based on peak current delivered into the load, switching frequency and acceptable ripple. However, the ripple voltage may be dominated by the capacitor series resistance (ESR) in most cases [4]. Larger capacitor sizes may have a larger series inductance (ESL), so it is recommended to have smaller decoupling capacitor in parallel to compensate. As with the output filters, the device EVM and datasheet are the best reference for decoupling capacitor selection.

5.4 Post-processing Considerations

The Smart Amp signal chain is comprised of pre-processing, the Smart Amp algorithm and output gain controls (G_O). Pre-processing includes input volume controls (G_I) and custom equalization filters. Any changes that occur after the Smart Amp algorithm (post-processing) will affect how Smart Amp understands a speaker and acts on it.

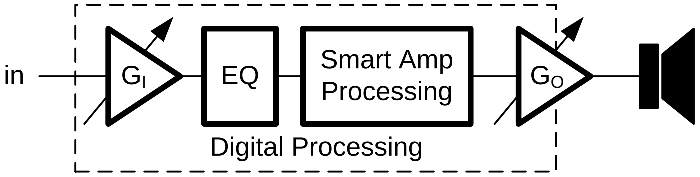


Figure 12. Smart Amp Signal Chain

Smart Amp devices have register-based output gain controls and may also have hardware pin gain control for the output amplifiers. It is recommended to set the gain level such that the full-scale voltage output ($V_{pk,FS}$) is close to the desired maximum power supply current per channel. In most circumstances, the gain should be set such that the peak output voltage is close to the supply voltage ($V_{dd,sa}$), but if the supply voltage is much higher, then the current can be controlled by lowering the post-processing gain.

The TAS5766M/68M has three post-processing gain controls, as shown in Table 7.

Table 7. TAS5766M/68M Post-processing Gain Controls

Gain Type	Control Type	Gain Values
DAC Fine Digital Gain Control (G_F)	Page 0 / Register 61 Page 0 / Register 62	-103dB to 0dB
DAC Analog Gain Control (G_A)	Page 1 / Register 2	-6dB 0dB
Power Stage Gain (G_P)	Controlled via GAIN pin of the device	14dB 20dB

The peak full-scale output voltage can be calculated as follows:

$$V_{pk,FS} = \sqrt{2} \times 2V_{rms} \times 10^{(G_F+G_A+G_P)/20}$$

For example, if $G_F = 0dB$, $G_A = 0dB$ and $G_P = 20dB$, then $V_{pk,FS} = 28.28V$(differential). Note that this level is slightly above the maximum supply voltage of the device. If it is desired to reduce the output current by half, the output swing could be reduced by reducing the output amplifier gain to 14dB.

IMPORTANT:

Any post-processing change requires re-calibration of the System Gain in the Smart Amp algorithm using the Smart Amp software.

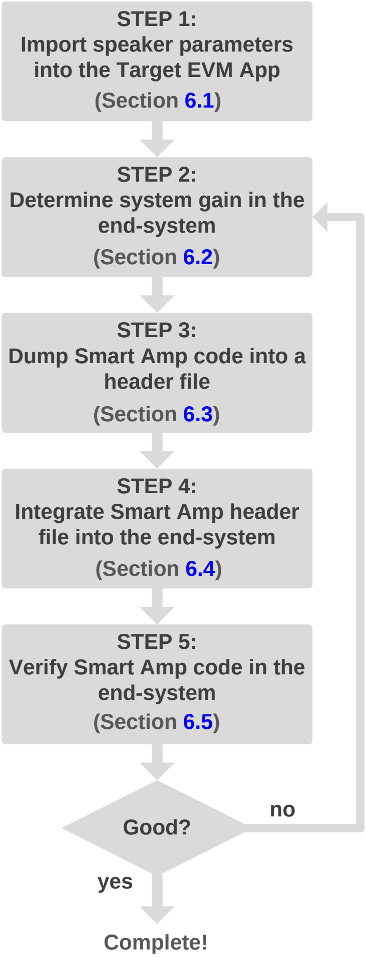
The Smart Amp software provides a means to re-calibrate the System Gain into the algorithm. More details are discussed in Chapter 6.

6 Software Implementation and Verification

Smart Amp first needs to understand the speaker and enclosure it will be connected to in the end product. TI provides a tool, called the *Learning Board* which extracts speaker parameters without the need of expensive equipment. Chapter 3 explains how to obtain these parameters in detail.

There are several Smart Amp *Apps*, one for each EVM, that share a common .ppc3 file extension. Speaker information obtained with the Learning Board App can be imported into any Smart Amp Target EVM App, regardless of the chip being used.

The process of integrating the Smart Amp tuning into the end-system is shown to the right. It is recommended to follow all these steps, in order, every time a new tuning is performed. A checklist following this process is provided in Appendix B and can be printed for record keeping.


STEP 1:
Import speaker parameters into the Target EVM App
(Section 6.1)
STEP 2:
Determine system gain in the end-system
(Section 6.2)
STEP 3:
Dump Smart Amp code into a header file
(Section 6.3)
STEP 4:
Integrate Smart Amp header file into the end-system
(Section 6.4)
STEP 5:
Verify Smart Amp code in the end-system
(Section 6.5)
Good?
no
yes
Complete!

The first step is to import the speaker parameters obtained from the Learning Board into the Target EVM App. For example, if the end-system uses the *TAS5766M*, then the speaker parameters must be imported into the *TAS5766M App* and loaded into the *TAS5766M EVM* in order to generate the required header files. This is explained in detail in Section 6.1.

The second step is to determine the system gain in the end-system. This is important because Smart Amp needs to know the relationship between the digital levels and the analog output levels in order to protect the speaker. See Section 6.2.

Once the Smart Amp *System Gain* is calibrated, the Smart Amp code can be dumped into a header file as explained in Section 6.3. The Smart Amp Target EVM (for example, the TAS5766M EVM) is required in order to generate this header file.

The fourth step is to integrate the header file from the previous step into the end-system microcontroller. See Section 6.4.

The last step is to verify that Smart Amp is working properly in the end-system. There are several easy tests that can be performed that ensure the gain calibration was implemented properly. This is covered in Section 6.5.

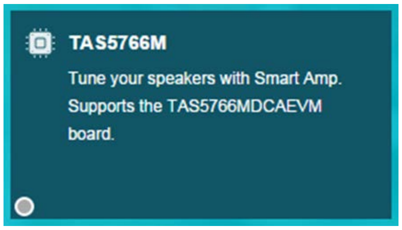
Section 6.6 describes how to change parameters on-the-fly in the end-system (not required).

6.1 Importing the Speaker Parameters into the Smart Amp Target EVM App

To generate a header file, it is necessary to first import the speaker characterization and tuning data from a .ppc3 file into the Target EVM App (e.g. TAS5766M App).

1. **Connect** the *Target EVM* to your PC using the USB cable.
   a. Provide a power supply (matching the one to be used in the final system).
   b. **Connect** the loudspeaker.

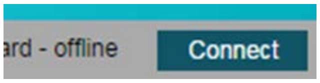
2. In the PurePath™ Console 3 App Center, open the **App** that corresponds to the Target EVM.



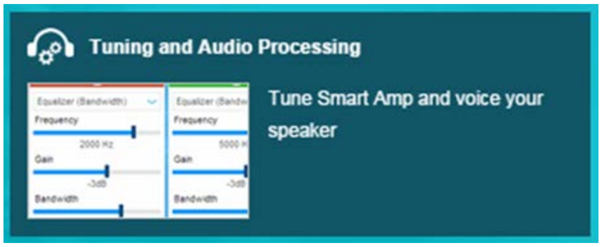
TAS5766M

Tune your speakers with Smart Amp. Supports the TAS5766MDCAEVM board.

3. If the board is shown as offline, click **Connect**.



4. Click **Tuning and Audio Processing**.



Tuning and Audio Processing

Tune Smart Amp and voice your speaker

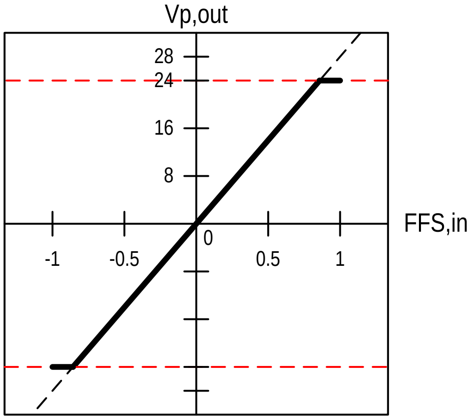
5. **Import** the characterization data that was obtained during the *Characterization* and Tuning processes (Chapters 3 and 4) by clicking the Import button.

6.2 Calibrating the System Gain in the End-System

Before integrating the Smart Amp code into the end-system, the *System Gain* must be determined and entered into the Smart Amp software.

The *System Gain* is a normalization factor that scales the voltage at the load to the digital level at the output of the DAC. Its units are *V/FS*, or *volts per full-scale*.

A *System Gain* of 28 means that a digital level of 1 FFS will be scaled up to 28 V_{pk}. However, it should be noted that this level will be limited by the power supply and the output impedance of the amplifier, as shown to the right.



During the *System Gain* calibration process, a -20dBFS 1 kHz tone is played and its RMS voltage measured at the output of the amplifier with a multi-meter. Based on this voltage, the *System Gain* is calculated as shown below.

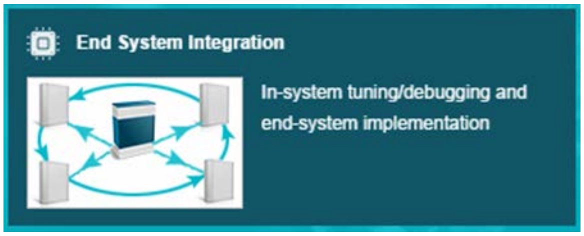
$$G_{system} = \sqrt{2}V_{L(rms)} \times 10$$

For example, if a $2V_{rms}$ signal is observed at the output, this means that the *System Gain* is approximately 28. Similarly, a $1V_{rms}$ measured signal will yield a *System Gain* of approximately 14.

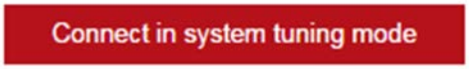
Obtaining the correct *System Gain* is extremely important – if under-calculated, Smart Amp will assume lower power is being delivered but in reality it might be exceeding the real thermal and excursion limits of the loudspeaker.

System Gain calibration is performed by downloading a *Tone Generator* code into the target device and measuring the RMS voltage with a multi-meter. The following steps show an example on how to obtain the *System Gain* of a *TAS5766M*-based end-system:

1. Ensure the *PurePath™ Console Motherboard* and the *Target EVM* are connected as described in Section 6.1. It is good practice to load the ppc3 file before obtaining the *System Gain* to ensure it is not overwritten later on.
2. Click **End-System Integration**.



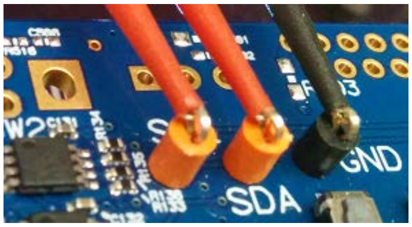
3. Select **In-System Tuning** and click **Connect in system tuning mode**. This will configure the App in *System Tuning* mode.



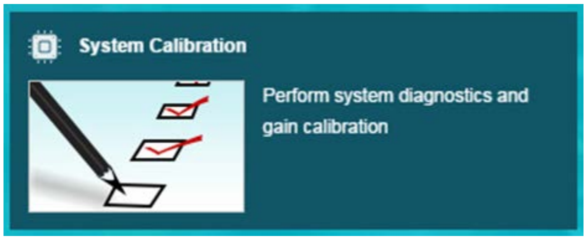
4. Ensure the target *End-System* (i.e. customer's board) is powered and the IO voltage of the I2C lines is 3.3V.

5. Ensure the target device hardware pins and external signals are configured correctly. For example, the *TAS5766M* requires the XSMT pin to be high and I2S signals to be present for the tone generator to work.

6. **Connect** the *SCL*, *SDA* and *GND* test points of the *PurePath™ Console Motherboard* to the I2C bus of the end-system, as shown to the right.



7. Click and perform a **System Calibration**. This process ensures that the Smart Amp algorithm is properly scaled based on the amplifier output gain. Having the loudspeaker connected provides more accurate results.


System Calibration

Perform system diagnostics and gain calibration

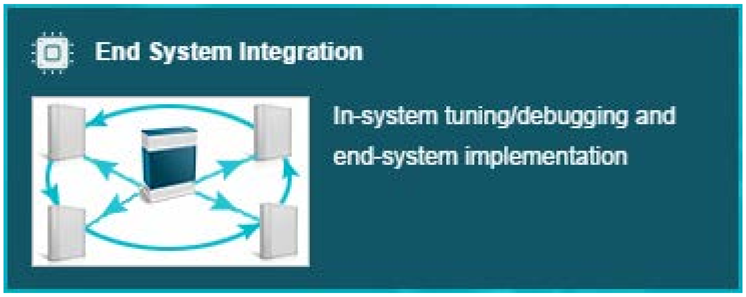
8. Save a .ppc3 file under a new name and disconnect the I2C bus from the end-system. This will save the *System Calibration* setting for future use.

If connecting I2C lines to the end-system is not possible, it can be estimated by performing the calibration with the Target EVM and calculating the gain. For example, if the *TAS5766M EVM* GAIN pin is configured for 20dB, but the target GAIN pin in the end-system is 14dB, then attenuate the *System Gain* obtained with the EVM by 6dB (half). This would yield 14 V/FS down from 28 V/FS. Refer to Section 5.4 for detailed calculations.

6.3 Smart Amp Register Dump into a Header File

Smart Amp devices have basic configuration registers as well as registers related its processing engine. By default, a Smart Amp device will boot in a simple, non-Smart Amp, predefined mode. The Smart Amp code would then need to be downloaded into Instruction RAM (I-RAM) and Coefficient RAM (C-RAM) of the processing engine. Note that in order to download this code, the DSP(s) must be turned off – although C-RAM can be accessed on-the-fly under some circumstances. Additional details are available at [5].

Smart Amp is programmed into the end-system device through I^{2}C commands. The typical way of achieving this is to compile a header file into the system code. This header file can be generated by using the *Dump* feature found in the *End-System Integration* Page.



End System Integration

In-system tuning/debugging and end-system implementation

Figure 13. End-System Integration

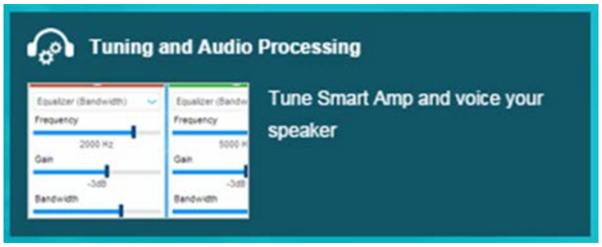
The *Dump* feature will read the device registers to compile an output header/cfg file based on the contents of a .ddf file. This .ddf file allows customizing which registers are actually generated, which is especially useful for systems using non-standard configurations (e.g. different audio interface formats, etc.).

A header file with the Smart Amp code can be dumped:

- With the *Target EVM* connected, or
- In *System Tuning* mode (i.e. connected to end-system)

Once connected and calibrated (as explained in Section 6.2):

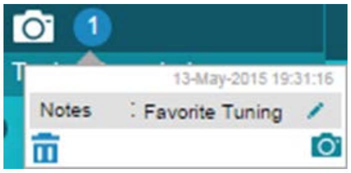
1. Open **Tuning and Audio Processing**. This will load the Smart Amp code to the target.



Tuning and Audio Processing

Tune Smart Amp and voice your speaker

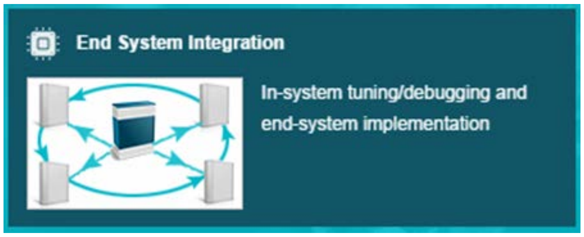
2. Select the desired **Snapshot**.



13-May-2015 19:31:16

Notes : Favorite Tuning

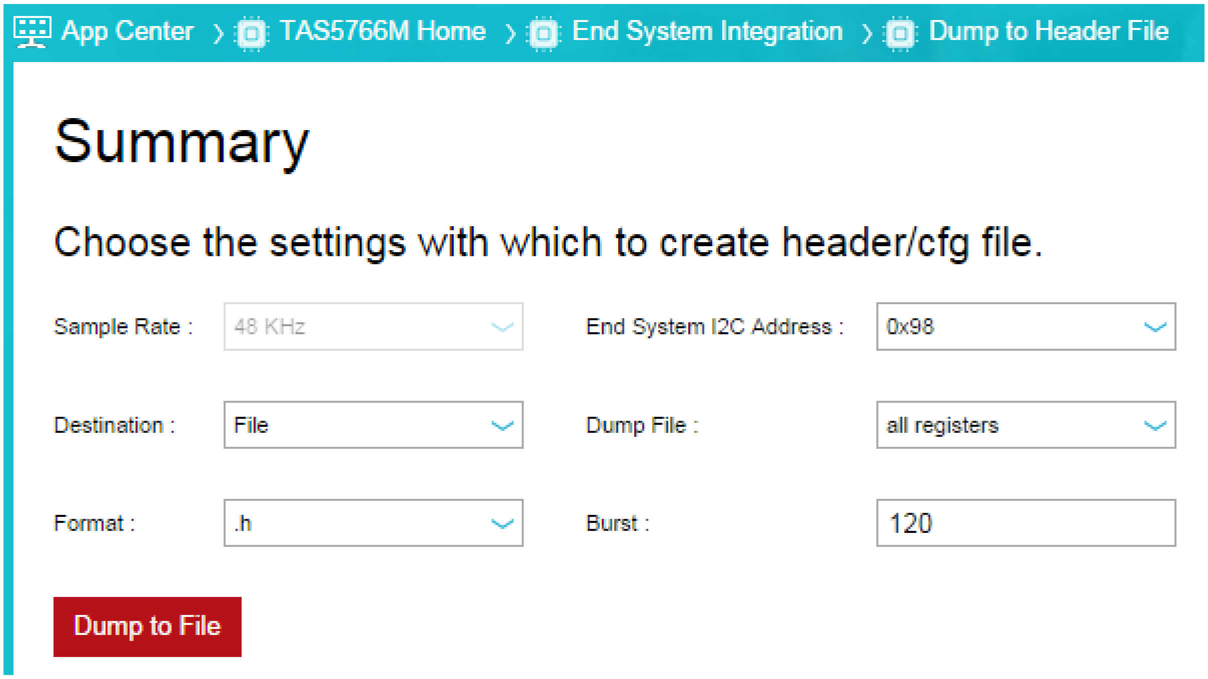
3. In the **End-System Integration** page, select **Dump Current State into a Header File**.



End System Integration

In-system tuning/debugging and end-system implementation

4. Click the **Dump to File** button, as shown below.


App Center › TAS5766M Home › End System Integration › Dump to Header File
Dump to File

Summary

Choose the settings with which to create header/cfg file.

Sample Rate : 48 KHz

End System I2C Address : 0x98

Destination : File

Dump File : all registers

Format : .h

Burst : 120

Figure 14. Dump to Header File Page

Setting *Burst* to 120 can help reduce the end-system download times and required code space significantly.

The sample *.ddf* file used for these steps simply dumped coefficients (C-RAM), instructions (I-RAM) and device settings. For this particular device, it places the device into standby, which allows C-RAM and I-RAM access through I^2C in the end-system. An example header file output snippet for the TAS5766M is shown in Figure 15.

```
cfg_reg registers[] = {
.
.
.
// -----------------------------------------------------------------------------
// Standby (allows I2C access to C-RAM and I-RAM)
// -----------------------------------------------------------------------------
// Select Page 0
    { 0x00, 0x00 },
// Set the device into Standby
    { 0x02, 0x10 },
// -----------------------------------------------------------------------------
// Begin Coefficient Memory (C-RAM_D), Buffer A Dump
// -----------------------------------------------------------------------------
// Page 44 (0x2C) Dump

    { 0x00, 0x2c },
    { CFG_META_BURST, 121 },
        { 0x08, 0x40 },
        { 0x26, 0xe7 },
.
.
.
// -----------------------------------------------------------------------------
// Wake
// -----------------------------------------------------------------------------
// Select Page 0
    { 0x00, 0x00 },
// Wake from Standby
    { 0x02, 0x00 },
};
```

Figure 15. Example Header File Output Snippet

6.3.1 Working with Non-Standard Configurations

As mentioned previously, custom *.ddf* files can be created as needed. If the Smart Amp device design in the end-system is the same as in the EVM schematic and clocking is similar then there is no need to edit this file. However, if there are changes from the EVM schematic design such as different clock configurations, then this *.ddf* file may need to be modified. Details on how to create custom DDFs can be found in the Appendix sections of the *PurePath™ Console 3 User Manual* [7].

The default DDF files can be found by searching **.ddf* in the folder below:

"*C:\Program Files (x86)\Texas Instruments\PurePath Console 3*"

6.4 Integrating the Output Header File into the End-System

The *PurePath™ Console 3 User Manual* [7] provides detailed information an example code on how to parse a header file in the Appendix sections.

The general concept is simple. The header file obtained from Section 6.3 contains an array of two element structures in which the first member is the register address (or offset) and the second member is the data written to that register. To distinguish between an actual register write and special commands in the parser, the *cfg_reg* is defined as a union, as shown in Figure 16. The parser would interpret a value of 128 and above as a command, rather than an offset. This does not cause any conflicts with target devices because none have registers above Register 127.

```
// Definition of the cfg_reg type
typedef union
{
    struct
    {
        cfg_u8 offset;
        cfg_u8 value;
    };
    struct
    {
        cfg_u8 command;
        cfg_u8 param;
    };
} cfg_reg;
```

Figure 16. Example Header File Output Snippet

The microcontroller would simply need to parse an array similar to the one shown in Figure 15 and generate I^2C commands.

6.5 Smart Amp Verification in the End-System

Once the Smart Amp code (header file) is implemented in the end-system, basic verification can begin.

Basic Verification:

1. Ensure power supplies for Smart Amp are proper.
2. Ensure audio clocks are present. Depending on the configuration (as defined by the .ddf file used to generate the header file) audio clocks may be generated by the Smart Amp device or may be input from an external source.
3. Ensure the *word clock* frequency matches the sample rate of the Smart Amp plug-in by measuring with an oscilloscope or multi-meter.
4. Set the volume of the host processor to a low level (e.g. -20dB to -40dB).
5. Unmute the Smart Amp device. For example, set the *XSMT* pin of the *TAS5766M* high.
6. Play a music file.

Once it is verified that audio is working, the Smart Amp *System Gain* calibration can be verified.

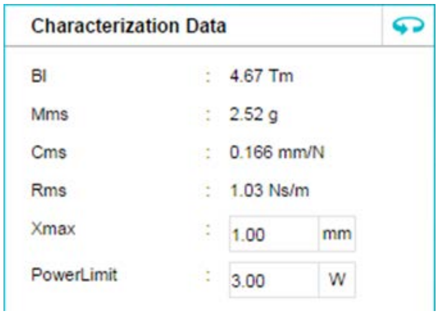
System Gain Verification:

1. Calculate the expected thermal limit in volts RMS using the following equation:

   $$V_{th\_limit} = \sqrt{P_{th\_limit} \times R_e}.$$

   - The P_{th_limit} value is obtained from the Characterization Data panel of the *Tuning and Audio Processing* page.
   - Click ↺ to obtain R_e.

   For example, $P_{th\_limit} = 3\ W$ and $R_e = 3.96\ \Omega$ would correspond to 3.45 V_{rms}.



Characterization Data

Bl	:	4.67 Tm
Mms	:	2.52 g
Cms	:	0.166 mm/N
Rms	:	1.03 Ns/m
Xmax	:	1.00 mm
PowerLimit	:	3.00 W

2. Wear hearing protection equipment.
3. Connect the positive and negative terminals of the loudspeaker to a multi-meter (volts AC setting).
4. Set the volume of the host processor to a low level (e.g. -20dB to -40dB).
5. Play the same excitation frequency that configured during the thermal characterization. A 20 kHz sine tone is typical.
6. Slowly increase the tone amplitude. The measured voltage should settle near V_{th_limit} even as the amplitude is increased to 0dB.

IMPORTANT:

These verification steps only confirm that the algorithm has been correctly calibrated. It is the responsibility of the customer to perform additional verification and stress tests to ensure reliability of the loudspeaker system.

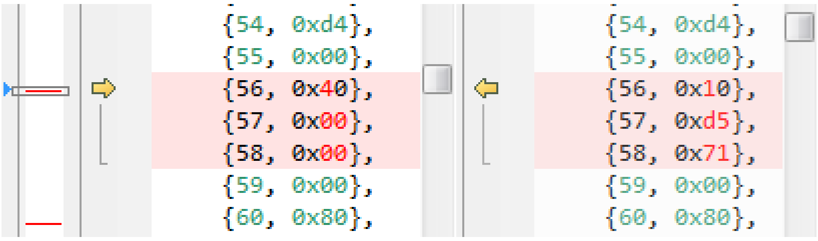
6.6 Changing Device Parameters On-the-fly in the End-System

If it is desired to change a parameter once the code is inside the end-system on-the-fly, its coefficient location must be determined. For example, if the designer wants to change the input (pre-processing) volume level, then he/she can generate two header files and use a compare tool to determine the coefficient location.

The image below shows the differences between the two files. Two things can be determined by observing these differences: the page/register location and the value.

In the left column, a value of 0x400000, or 0dB, is written to Page 46, Registers 56-58 (Buffer A). The same data is written to the same registers in Page 64 (Buffer B). This is because the miniDSPs are off during initialization hence both buffers being exposed to the I^{2}C/SPI bus. More details are explained in [5].

The right column shows a value of 0x10D571, or -11.6dB.



```
{54, 0xd4},        {54, 0xd4},
{55, 0x00},        {55, 0x00},
{56, 0x40},        {56, 0x10},
{57, 0x00},        {57, 0xd5},
{58, 0x00},        {58, 0x71},
{59, 0x00},        {59, 0x00},
{60, 0x80},        {60, 0x80},
```

Figure 17. Volume Control Change Comparison

Note that the miniDSP code writes to some C-RAM locations directly, so it is normal for some locations to change from dump to dump.

7 Support

Smart Amp is supported through the Audio Amplifiers Support Forum. Please include the words *Smart Amp* and the part number in the subject line. Contact your local TI sales representative for additional details.

References

[1] **AES2-1984**: *AES Recommended Practice Specification of Loudspeaker Components Used in Professional Audio and Sound Reinforcement*. Audio Engineering Society, New York, NY., US

[2] **AES2-2012**: *AES standard for acoustics - Methods of measuring and specifying the performance of loudspeakers for professional applications - Drive units*. Audio Engineering Society, New York, NY., US

[3] *Class-D LC Filter Design* application report (SLOA119)

[4] *Design Considerations for Class-D Audio Power Amplifiers* application report (SLOA031)

[5] *Coefficient RAM Access Mechanisms* application report (SLAA425)

[6] Leach, Marshall W., Jr., *Introduction to Electroacoustics and Audio Amplifier Design*. Kendall Hunt Publishing, 2009.

[7] PurePath™ Console 3 User Manual (SLOU408)

Appendix A. Supported Speaker Configuration Types

Supported speaker configurations types are described in this section.

The speaker membrane area Sd of a circular speaker is obtained by measuring the diaphragm diameter with a metric ruler from center point to center point of the surround, as shown below. Diaphragm area for other speaker shapes, such as 'racetrack' speakers, can be calculated by either approximating the speaker as a square or by adding the area of the half-ellipses at both ends plus the area of the square in the center.

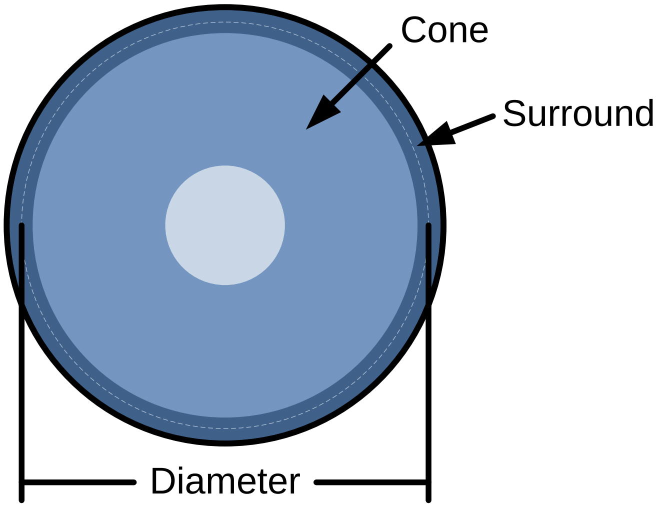


Figure A-2. Obtaining the Loudspeaker Diameter

Below are the basic speaker elements that are shown in the subsequent subsections.

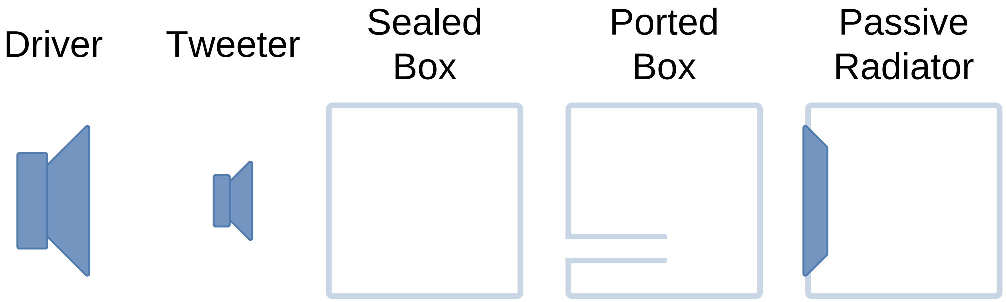


Figure A-3. Speaker Element Types

A.1 Free Field / Single Speaker in Box

Electro-mechanical Characterization

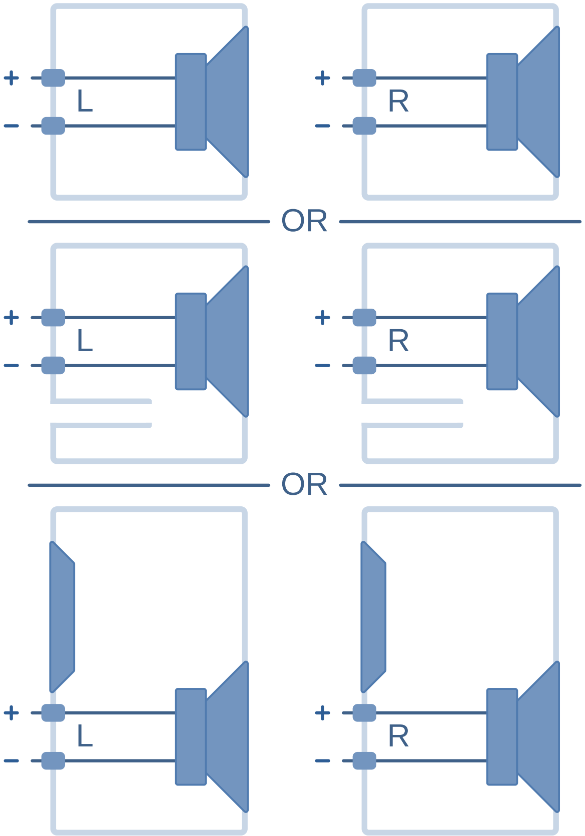
- **Connections** - Left Channel Speaker to the L+/L- (TEST) outputs of the *Learning Board*.
- **Bl** - Enter the mass placed at the speaker when prompted.
- **Sd** – Enter the diaphragm area as measured.
- **Sp (if applicable)** – Enter the total port or passive radiator area as measured.
- **In-box SPL for Passive Radiator** – Follow the steps in the wizard.

Thermal Characterization

- **Connections** - Left Channel Speaker to the L+/L- (TEST) outputs of the *Learning Board*.

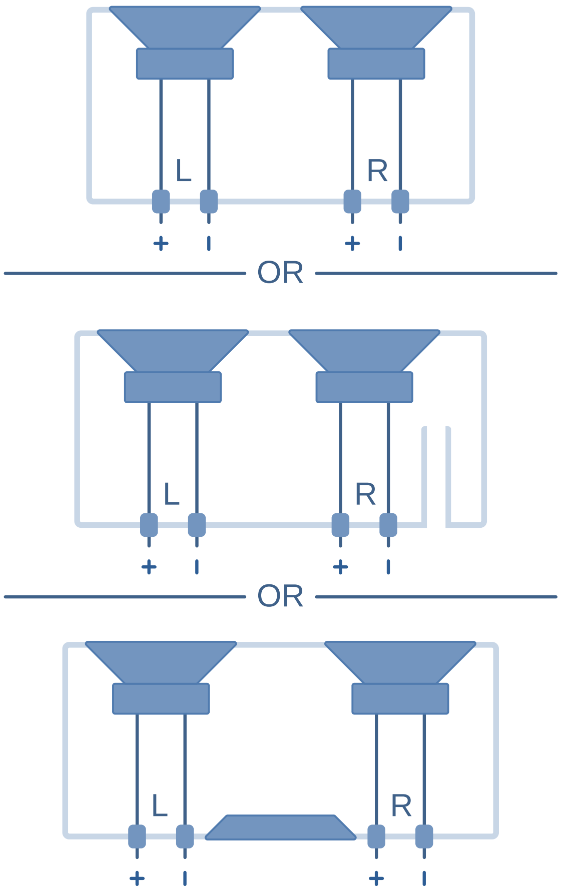
Tuning

- **Connections**
  - Left Channel Speaker to the L+/L- (TEST) outputs of the *Learning Board*.
  - Right Channel Speaker to R+/R- outputs of the *Learning Board*.



A.2 Left and Right Drivers in the Same Box

Characterization



- **Connections**
  - Left Channel Speaker to the L+/L- (TEST) outputs of the *Learning Board*.
  - Right Channel Speaker to R+/R- outputs of the *Learning Board*.
- **Bl** - Add mass m_L to the left speaker and mass m_R to the right speaker, where $m_L = m_R$. Enter the mass of the left speaker m_L when prompted.
- **Sd** – Enter the diaphragm area of a single speaker.
- **Sp (if applicable)**
  - **Single Port or Passive Radiator** – If the box has only one of these elements with area S'_p, enter the area of that element divided by 2. In other words, $S_p = S'_p \div 2$.
  - **Dual Port or Passive Radiator** – If both elements have the same area S'_p, then enter the area of only one of the elements. In other words, $S_p = S'_p$. Otherwise, enter the total area $S''_p = S'_{p1} + S'_{p2}$ divided by 2. In other words, $S_p = S''_p \div 2$.
- **In-box SPL for Passive Radiator** – Follow the steps in the wizard.

Thermal Characterization

- **Connections** – *Only* the Left Channel Speaker to the L+/L- (TEST) outputs of the *Learning Board*.

Tuning

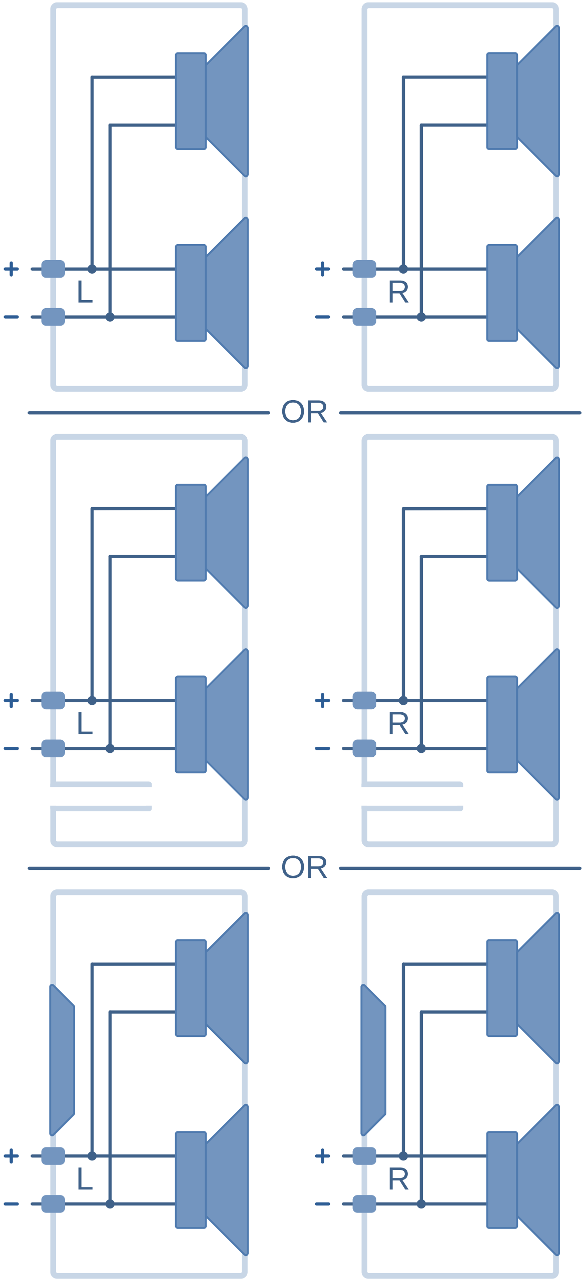
- **Connections**
  - Left Channel Speaker to the L+/L- (TEST) outputs of the *Learning Board*.
  - Right Channel Speaker to R+/R- outputs of the *Learning Board*.

IMPORTANT:
For this configuration, enable Mono Bass mix as described in Section 4.2.1. Set the frequency of the Mono Bass to 2x the highest impedance peak frequency.

A.3 Parallel/Series Drivers in the Same Box (no analog crossover)

Electro-mechanical Characterization

- **Connections** - Left Channel Speakers to the L+/L- (TEST) outputs of the *Learning Board*.
- **Bl** – Add mass m_1 to one speaker and mass m_2 to the other speaker of the same cabinet, where $m_1 = m_2$. Enter the sum of the masses $m_1 + m_2$ when prompted. If Bl is entered manually:
    - **Series** – enter $2 \times Bl$ of a single driver.
    - **Parallel** – enter Bl of a single driver.
- **Sd** – Enter the sum of the diaphragm area of both speakers in the same cabinet $S_{d1} + S_{d2}$.
- **Sp (if applicable)** – Enter the total port or passive radiator area as measured.
- **In-box SPL for Passive Radiator** – Follow the steps in the wizard.
- **Peak Excursion**
    - **Series** – enter $2 \times X_{max}$ of a single driver.
    - **Parallel** – enter X_{max} of a single driver.



Thermal Characterization

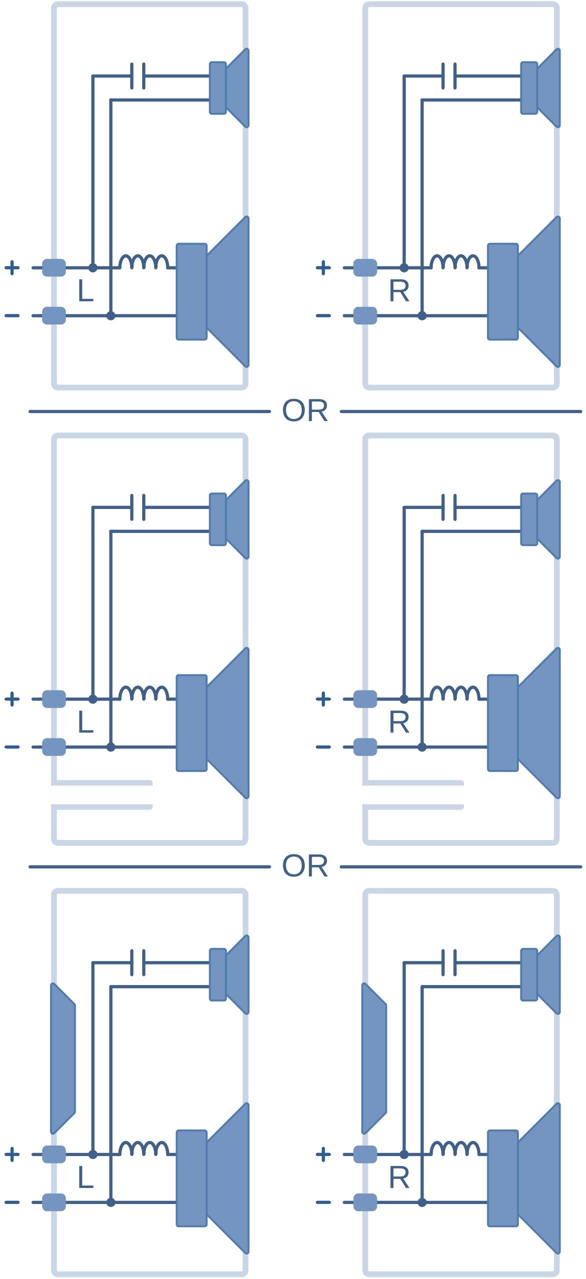
- **Connections** - Left Channel Speakers to the L+/L- (TEST) outputs of the *Learning Board*.
- **Power Handling** – if thermal characterization is skipped, provide 2x the maximum power handling of a single speaker.

Tuning

- **Connections**
    - Left Channel Speakers to the L+/L- (TEST) outputs of the *Learning Board*.
    - Right Channel Speakers to R+/R- outputs of the *Learning Board*.

A.4 Parallel Drivers in the Same Box (analog crossover)

A speaker cabinet with a bass driver and a tweeter is supported by employing excursion protection exclusively to the bass driver. The analog crossover frequency is assumed to be high enough such that the tweeter is not damaged by excursion. Thermal protection may be employed to the bass driver or the tweeter – whichever driver has the worst-case thermal model. To determine the worst case, both bass and tweeter drivers need to be characterized thermally. The driver that results in the lowest power handling (as shown in the SOA) should be used for thermal protection. If the result is too conservative, the bass thermal model could be used. However, the tweeter may not be protected for sinusoidal signals which would require stress/reliability tests by the customer.



Electro-mechanical Characterization

- **Connections** – Remove the crossover network and connect the bass driver directly to the L+/L- (TEST) outputs of the *Learning Board*.
- **Bl** - Enter the mass placed at the bass driver when prompted.
- **Sd** – Enter the diaphragm area of a bass driver.
- **Sp (if applicable)** – Enter the total port or passive radiator area as measured.
- **In-box SPL for Passive Radiator** – Follow the steps in the wizard.

Bass Driver Thermal Characterization

- **Connections** – Bass driver to the L+/L- (TEST) outputs of the *Learning Board*.

Tweeter Thermal Characterization

- **Connections** – Tweeter to the L+/L- (TEST) outputs of the *Learning Board*.

Tuning

- **Connections** – Reconnect the crossover network for both cabinets.
  - Left Channel Speakers to the L+/L- (TEST) outputs of the *Learning Board*.
  - Right Channel Speakers to R+/R- outputs of the *Learning Board*.

Appendix B. Smart Amp End-System Implementation Checklist

Use this checklist to ensure all Smart Amp end-system implementation steps are followed and to keep a history of project tunings, versions and filenames.

PROJECT NAME: ______________________________ TUNING VERSION: ____________

□ Ensure end-system power supply current is adequate (Section 5.2).

□ Import speaker parameters from Learning Board into the Smart Amp Target App and tune Smart Amp (Section 6.1).

□ Determine the system gain of the end-system (Section 6.2).

□ Dump the Smart Amp header file using the Target App (Section 6.3).

□ Integrate the Smart Amp header file into end-system (Section 6.4).

□ Perform *Basic and System Gain Verification* in the end-system (Section 6.5).

□ Perform stress and reliability tests of the loudspeaker system.

NOTES: ____

____

____

____

____

____

____

____

NAME: ____ DATE: _______________

TEXAS INSTRUMENTS
PurePath Smart Amp

Appendix C. MATLAB Compiler Runtime Troubleshooting

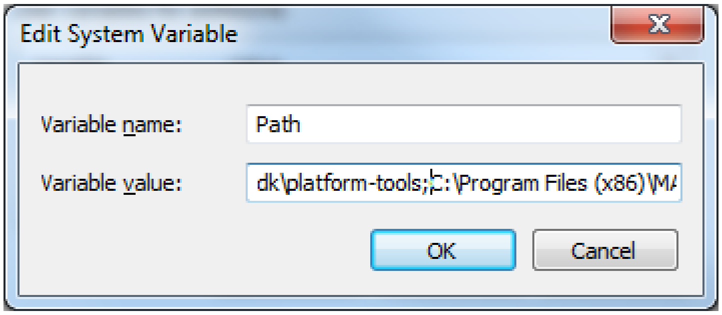
If an error message appears when launching the Smart Amp software:

1. Ensure MATLAB Compiler Runtime (MCR) is installed as described in Section 2.2.
2. Close PurePath™ Console 3.
3. If the MCR is already installed, browse to *C:\Users\[user]\AppData\Local\Temp\[user]* using Windows Explorer and delete *mcrCache7.16*. The *AppData* folder may require un-hiding files in the Windows system settings.
4. Open PurePath™ Console 3, load the Smart Amp App.
5. If issues persist, refer to Chapter 7 for support contact information.

Some antivirus software may prevent the MCR installation from setting Windows environment variables. If this is the case:

1. Go to the Windows *Control Panel* > *System* > *Advanced System Settings*
2. Click *Advanced* > *Environment Variables*.
3. Under *System variables*, click the Path variable and click *Edit*.
4. Add a semicolon "*;*" after the last value and paste

   "*C:\Program Files (x86)\MATLAB\MATLAB Compiler Runtime\v716\runtime\win32*" after the semicolon as shown below.



Variable name: Path

Variable value: dk\platform-tools;C:\Program Files (x86)\MA

Figure C-1. Editing Windows System Variables

Revision History

This revision history highlights the changes made to this application report.

Revision History

Version	Date	Author(s)	Additions/Modifications/Deletions
SLOU	20150820	J. Arbona	Added additional details to the Smart SPL section.
SLOU	20150715	J. Arbona, H. Rajagopala, C. Neesgaard, S. Kandadai, M. Spence, M. Kucic, T. Christiansen	Initial Release.

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Trademarks

All other trademarks are the property of their respective owners.