Multiband Noise Reduction Component for PurePath Studio Portable Audio Devices

Audio Converters

ABSTRACT

This application note describes the features, operating procedures and control capabilities of a noise cancellation solution for the Texas Instruments PurePath Studio Portable Audio products. This solution is developed as a simple to use drag and drop component in the PurePath Studio Portable Audio development environment. A variety of control features allow for customization of noise reduction parameters to optimize the solution for many applications. The noise reduction component is operational at a multitude of sample rates. This application note also provides an overview of PurePath Studio Portable Audio’s features and capabilities applicable to the customization of this noise reduction solution as well as instructions for tuning the noise reduction parameters.
Contents

1 Introduction ........................................................................................................................................... 3

2 Noise Reduction Overview .................................................................................................................. 4
  2.1 Noise Reduction Algorithm ........................................................................................................ 4
  2.2 Multi-band Filter Bank ............................................................................................................... 5
  2.3 Noise Gate .................................................................................................................................. 6
  2.4 Multi-band Summer ..................................................................................................................... 6

3 Noise Reduction Implementation ....................................................................................................... 7
  3.1 Processor Overview .................................................................................................................... 7
    3.1.1 AIC3254 ................................................................................................................................. 7
    3.1.2 AIC36 .................................................................................................................................. 7
    3.1.3 TSC2117 .............................................................................................................................. 7
  3.2 PPS Environment .......................................................................................................................... 7
    3.2.1 PPS Components ................................................................................................................... 8
    3.2.2 Build and Download Process Flow ..................................................................................... 8
    3.2.3 Loading Scripts ..................................................................................................................... 9
    3.2.4 Component GUIs ................................................................................................................... 9

4 Noise Reduction Process flow ............................................................................................................. 10
  4.1 Processor Frameworks ................................................................................................................ 10
  4.2 Noise Reduction ........................................................................................................................ 10
  4.3 Noise Reduction Control Properties ........................................................................................ 11
    4.3.1 Threshold Property .............................................................................................................. 11
    4.3.2 Attenuation Property .......................................................................................................... 12
    4.3.3 High Filter and Low Filter Frequency ................................................................................ 12
    4.3.4 Filter Coefficients .............................................................................................................. 12

5 Noise Reduction Application Example .............................................................................................. 13
  5.1 Noise Reduction Parameter tuning ........................................................................................... 13

6 Noise Reduction System Requirements ............................................................................................ 17

7 Conclusion ......................................................................................................................................... 18

8 References .......................................................................................................................................... 18

Figures

Figure 1: Algorithm Overview ............................................................................................................. 4
Figure 2: High Freq and Low Freq settings .......................................................................................... 5
Figure 3: Typical frequency response after summation ........................................................................ 6
Figure 4: PurePath Studio Features .................................................................................................. 8
Figure 5: PurePath Studio Tools menu ............................................................................................... 9
Figure 6: I2C Command Window ....................................................................................................... 9
Figure 7: PurePath Studio View menu ............................................................................................... 10
Figure 8: Noise Reduction Control Panel ........................................................................................ 11
Figure 9: Mono_Mixer_2 Runtime Properties Window .................................................................... 14
Figure 10: Tools/Options/Target menu -> Auto WriteState .............................................................. 14
Figure 11: Build/Write State menu .................................................................................................... 15

Tables

Table 1. Noise Reduction Resource Requirements ............................................................................... 17
Table 2. Available Processor Resource ............................................................................................... 17
1 Introduction

This application note describes the implementation of a multiband noise reduction solution available for the Texas Instruments PurePath Studio Portable Audio Devices. This solution was developed as a drag and drop component in the PurePath Studio Portable Audio development environment. It provides control properties to customize the implementation for various applications. This application note also provides an overview of PurePath Studio Portable Audio’s features and capabilities applicable to the customization of this noise reduction solution as well as guidelines for tuning the noise reduction parameters.

Background noise can interfere with the intelligibility of a conversation over communication devices and results from surrounding sources such as automobiles, planes, other people’s conversations, fans, lights, air conditioning, etc. Noise reduction is a technology that is applicable to reducing background noise in products such as mobile phones, hands-free speaker phones, headphones and voice recognition systems to mention a few. Low cost and low power solutions are most applicable in these cases and are where this noise reduction solution is targeted. Noise reduction systems are designed to remove background noise while minimally affecting the main audio content (i.e speech). This noise reduction design is available as a simple drag and drop component that can be combined with other components to create an effective audio processing system.
2 Noise Reduction Overview
This solution implements an effective multiband noise gate to remove undesirable noise. The algorithm splits the audio band into sub-bands. Each sub-band is then subjected to a noise-gate algorithm with variable attenuation and threshold, which enables automatic noise-floor tracking. The range over which the bands are spread can be set by the user, as can the amount of attenuation, and the threshold, which sets the point at which the differentiation is made between ‘signal’ and ‘noise’. The following describes briefly the algorithm.

2.1 Noise Reduction Algorithm
The algorithm consists of the following sequence of processes which are displayed in Figure 1.

1. The input audio signal is split into multiple frequency bands according to the user input. The center frequency for each of the high frequency and low frequency bandpass filters are variables controlled by the user.

2. Each frequency band is ‘gated’ so that only signals that are a certain threshold above the estimated noise floor is passed. Both the threshold and amount of attenuation level is user programmable.

3. The output from each of the frequency bands is summed to form the noise reduction output.

![Figure 1: Algorithm Overview](image)
2.2 Multi-band Filter Bank
The input signal is split into multiple bands by a filter bank built from 4\textsuperscript{th} order band pass filters. Each 4\textsuperscript{th} order band pass filter consists of a pair of cascaded second order band pass filters. The user controls ‘High Freq’ and ‘Low Freq’ to set the center frequencies of the lowest and highest bandpass filters as shown in Figure 2.

Figure 2: High Freq and Low Freq settings
The coefficients for each of the filters are automatically calculated according to the ‘High Freq’ and ‘Low Freq’ control settings. The ‘Q’ factor for each filter is managed automatically as well to obtain good frequency response flatness when the outputs from all filters are summed. Typical frequency response error after summation is less than 0.5dB. An example frequency response graph is shown in Figure 3.
2.3 Noise Gate
Each of the band-passed signals is processed by a separate noise gate. A noise gate allows a signal to pass through only when the signal is above the set threshold. In the Noise Reduction component a single threshold (controlled by a GUI) is applied to each of the noise gates. If a band-passed signal falls below the threshold that signal (and accompanying noise) is not allowed to pass or is attenuated by an amount corresponding to the “attenuation” control setting in the GUI. If a band-passed signal exceeds the threshold the noise gate is not applied to that particular band and the signal is passed unaffected.

2.4 Multi-band Summer
The output of each noise gate section is summed together generating the noise-reduced output signal. The summed output will generally have a flat frequency response between the GUI-programmed high and low frequency range as shown in Figure 3.

Figure 3: Typical frequency response after summation
3 Noise Reduction Implementation

3.1 Processor Overview
The noise reduction component can be implemented on any of the AIC3254, AIC36 or TSC2117 processors and adapts automatically to the precision and capabilities of each. The features, operation and interface are identical and although controlling the noise reduction is the same, the impact may vary depending on the precision of the processor.

3.1.1 AIC3254
The TLV320AIC3254 (sometimes referred to as the AIC3254) is a flexible, low-power, low-voltage stereo audio codec with programmable inputs and outputs, fixed/predefined and parameterizable signal processing blocks, integrated PLL, integrated LDOs and flexible digital interfaces. The AIC3254 incorporates dual miniDSPs (miniDSP_A and miniDSP_D) capable of meeting the needs of a variety of audio and communication applications. The miniDSP cores are fully software programmable and may be controlled via I2C or SPI interfaces. See the SLAS549A Technical Data Sheet for more details on this processor (2).

3.1.2 AIC36
The TLV320AIC36 (sometimes referred to as the AIC36) is also a flexible, low-power, low-voltage stereo audio codec with programmable inputs and outputs, fully programmable miniDSP, fixed/predefined and parameterizable signal processing blocks, integrated PLL, integrated LDOs, and flexible digital interfaces. It differs from the AIC3254 in memory size, memory width, and processing speed. See the SBAS387 Technical Data Sheet for more details on this processor (6).

3.1.3 TSC2117
The TSC2117 is a low-power, highly integrated high performance codec and touch screen controller, which supports stereo audio DAC, mono audio ADC, and SAR ADC. The device integrates several analog features such as a digital microphone interface, stereo headphone drivers, and stereo Class-D speaker drivers. The TSC2117 has fully programmable audio. See the SLAS550A Technical Data Sheet for more details on this processor (5).

3.2 PPS Environment
The development environment for the noise reduction implementation is based on the PurePath Studio (PPS) tool set which incorporates a drag and drop style Graphical Development Environment (GDE) for simple development of process flows (i.e. systems) based on standard and custom components as described in the following sections.

PurePath Studio incorporates a Graphical Development Environment (GDE) as shown in the Figure 4. This environment is used to create systems by dragging components from the palette onto the drawing area (environment) and connecting with lines to create system solutions.
The environment defines various system parameters such as the sample rate. These can be viewed in the configuration window by clicking on the drawing area.

### 3.2.1 PPS Components
By right clicking on a component the user can access either a help file describing the operation of the component or view available run time properties and/or design properties which are displayed on the right side of the drawing area. The noise reduction component does not have any design time properties, although it does have multiple run time properties. Run time properties can be changed while the processor is running, but design properties can only be changed before building the process flow. To change the value of properties, right click on the component and choose Properties option. The component properties are displayed on the right of the drawing area, and those properties that can be changed are displayed in bold type. However, it is required that for the Noise Reduction component that the user control the Noise Reduction algorithm using only the GUI which calculates all the coefficients and not via the properties window. Components are linked together to build systems such as a noise reduction enabled cell phone.

### 3.2.2 Build and Download Process Flow
A build is performed when the process flow graph has been completed by pulling down the “Build” menu as shown in the Figure 4. Choose “Generate Code” to build the system and then follow up with ‘Download code’ to load the software to the target board via the USB connection. The user can also choose “Download Code” initially, in which case both build and download will be performed. The code is loaded and automatically starts running. The user is requested to save the process flow. This creates a process flow file with the extension *.pfw. Process flow files may also be loaded from the “File” menu using the “Open” option.
3.2.3 Loading Scripts
Once the code has been loaded and is running, changes to the code, coefficients or registers may be made dynamically using scripts. Scripts may be loaded by pulling down the “Tools” menu and choosing the “I2C Command” File option as shown in Figure 5. Scripts are text files formatted to be read by the AIC3254. Script files may be edited using any text editor.

![Figure 5: PurePath Studio Tools menu](image)
The following window opens and the user can browse to locate the desired script file and click on “Execute now”. This will load the script to the target processor.

![Figure 6: I2C Command Window](image)

3.2.4 Component GUIs
Once the code has been built and loaded, the runtime controls, if available may be controlled by the runtime window on the right side of the GDE window. Some components provide a Graphical User Interface (GUI) for simpler manipulation of the component. GUIs may include sliders, buttons or check boxes. The noise reduction component implements a GUI shown in Figure 9 for control. Operation of the noise reduction GUI is described in Section 4.
4 Noise Reduction Process flow
The noise reduction component is implemented in a process flow consisting of at minimum input and output components (i.e. I2S input, I2S output, decimation or Interpolation). Other components may be used as well to build a solution for a variety of noise reduction applications. The Noise Reduction component must reside in miniDSP_D (except when implemented on the AIC3254) due to memory size limitations in the miniDSP_A. See Section 6 for resource availability.

4.1 Processor Frameworks
A Framework component is required for every process work flow. The Framework determines register settings for the processor including clocking, I/O and power supply operation. Noise reduction may be implemented using Frameworks that provide the minimum required resources as defined in Section 6.

4.2 Noise Reduction
The Noise Reduction component performs the process of reducing undesirable background noise from an audio stream. The noise reduction process does not require any additional components other than I/O components as noted above. However, a process work flow may include additional components as needed to implement a system incorporating noise reduction with additional signal processing such as for cell phones, speaker phones, etc., as long as processor resources will support all components. See Section 6 regarding available resources.

After linking in new components the available resources can be evaluated after a build. The resource window can be requested via the View menu as shown in the example displayed in Figure 7 below. The resources window shares the right pane with the properties window. Click on the Resources menu tab at the bottom of the right window to open the Resources window.

![Image of PurePath Studio View menu]

Figure 7: PurePath Studio View menu
4.3 Noise Reduction Control Properties

The Noise Reduction Component provides several programmable properties that allow the customization of the noise reduction implementation. The Noise Reduction component does not implement design time properties; however, there are several run time properties that can be customized while running a PurePath Studio process flow allowing for real-time changes and evaluation. These run time properties are displayed in the properties window on the right when highlighting the Noise Reduction component by clicking on the component. It is important to note that the properties displayed should not be edited in the run time properties window as these are mainly for display only. Noise Reduction Properties are only to be modified via the provided GUI.

The GUI can be accessed by double-clicking on the Noise Reduction component or by right-clicking on the component and selecting “NR Gui”. The control panel is shown in Figure 8. It is required that the user control the Noise Reduction algorithm using only the GUI which calculates all the coefficients. These coefficients are updated to run-time properties and displayed in the run time properties window.

![NR GUI](image)

**Figure 8: Noise Reduction Control Panel**

### 4.3.1 Threshold Property

The Threshold property is a parameter that defines the level above the estimated noise floor the signal must be (within it's band) to be considered ‘signal’ as opposed to ‘noise’. It spans from 0 dB to 20 dB and is used along with the Attenuation property to differentiate between ‘signal’ and ‘noise’. Increasing the threshold increases the level (above the automatically estimated noise floor) which the input must rise above in order to be considered ‘signal’. For example, for an input whose averaged noise floor is at -46dB, and with a threshold set of 10dB, the input (within the frequency band) must rise above -36dB to be considered ‘signal’. Signals below this threshold are considered to be ‘noise’ and are thus attenuated. The noise floor estimation is performed in real-time and is dependant upon the recent history of the input. The noise floor estimate is therefore...
dynamically (and automatically) calculated during operation. For this reason, the Attenuation control is not absolute in terms of level, but relative to this estimated floor.

Higher threshold levels may also impact the desired signal level; thus it must be applied judiciously to generate the maximum amount of noise reduction while maintaining adequate signal levels. The linear Threshold value is displayed in the properties window. However, it is not adjustable from the properties window. It is required that the user control the Noise Reduction algorithm using only the GUI which calculates all the coefficients.

4.3.2 Attenuation Property
The attenuation property is a parameter that defines how much noise attenuation will be applied. It spans from 0 dB to 20 dB and is used along with the Threshold property to differentiate between ‘signal’ and ‘noise’. Increasing the attenuation increases the amount of noise reduction applied. Higher levels, however may impact the desired signal as well; thus it must be applied judiciously to generate the maximum amount of noise reduction while minimizing any distortion of the desired signal. The linear Attenuation value is displayed in the properties window. However, it is not adjustable from the properties window. It is required that the user control the Noise Reduction algorithm using only the GUI which calculates all the coefficients.

4.3.3 High Filter and Low Filter Frequency
The high filter and low filter properties define the bandwidth over which the noise reduction filter bank is to be applied. The “Low Freq” control spans from 100 Hz to 1000 Hz. The “High Freq” control spans from 1000 Hz to 18,472.5 Hz. The filter bank is built from 4th order band pass filters. Each 4th order band pass filter is made from a cascade of a pair of second order band pass filters. The user controls ‘High Freq’ and ‘Low Freq’ to set the center frequencies of the lowest and highest bandpass filters as shown in Figure 2. The High and Low Center Frequency values are displayed in the properties window. However, do not adjust these settings using the properties window. It is required that the user control the Noise Reduction algorithm using only the GUI which calculates all the coefficients.

4.3.4 Filter Coefficients
The filter coefficients utilized for the band filters are displayed in the properties window. When the “Low Freq” and/or “High Freq” sliders are changed the filter coefficients will be updated. Do not adjust these coefficients directly. The implementation will operate in a suboptimal manner. It is required that the user control the Noise Reduction algorithm using only the GUI which calculates all the coefficients.
### 5 Noise Reduction Application Example

When built and downloaded, the Noise Reduction is automatically enabled and runs on the respective EVMs using the provided components, process flows and scripts. If utilizing the AIC3254, the EVM must initially be set up as described in Section 3 of the EVM User’s Guide, TLV320AIC3254EVM-K, SLAU264-October 2008. It is important to note the EVM sample rate. The EVM should be initially programmed for 44.1 kHz, but may be flashed to support a different sample rate by using the procedure found in the AIC3254 EVM Firmware Programming document. The PPS environment should configure for the same sample rate. The environment configuration menu will appear under the properties window by a click on drawing area. The configuration menu will appear as shown in the figure below. Open the sample rates tree and add the desired sample rate by setting it to “True”. A warning window will open telling you that a new sample rate has been chosen. You must change the “Current sample rate” by choosing it from the current rate menu at the end of the sample rate options. This will configure the PPS environment for the appropriate sample rate. This is necessary for components that calculate sample rate dependent properties such as filter coefficients. Connect the USB cable from the PC to the EVM. Open PurePath Studio to continue running the example.

In the example process work flow, data is input via the I2S port, filtered in miniDSP_D by a multi-biquad filter and split into a noise reduction path and a bypass path. A mux component is used to choose the signal path that will be output from the on-chip DAC via the interpolator. A volume control is also provided to control the output gain/attenuation.

#### 5.1 Noise Reduction Parameter tuning

The Noise Reduction implementation should begin running with the default values after the provided process work flow is opened, built and downloaded to the EVM. The process work flow allows the user to switch between the Noise Reduction path and the Bypass path using the Mono_Mux_1 run time properties. Click on the Mono_Mux_1 component and the MuxSelect property is displayed. Change the MuxSelect to 1 and enable the Noise Reduction path. Change it to 2 and enable the Bypass path.
To properly evaluate the difference between the noise reduced signal and the original bypassed signal it is recommended that “Auto WriteState” be disabled in the Tools/Options/Target window as shown below.

This means that property changes are not made independently and automatically, but rather all property changes (either made in the properties window or via the GUI) are applied at the same time when Build/Write State is selected. Thus, the Mux selection occurs exactly at the time the “Write State” selection is made. This makes for a more effective A/B comparison test.
The biquad component is used to bandlimit the input signal to match the bandwidth of the application (i.e. cell phone). For broadband signals the biquad filter may be removed. It is initially set up as a bandpass filter with lower and upper cut-off frequencies of 300 Hz and 3400 Hz respectively.

Configuring Noise Reduction Properties is dependent upon the application and the expected noise content; however there are some basic guidelines to help determine optimal settings. Because there is no easy objective measurement to obtain the optimal settings for a given application, best results may be determined by subjective evaluation. Subjective evaluation involves listening for signal clarity and ambient noise level provided a representative test signal. When performing an evaluation to establish noise reduction property settings, it is beneficial to utilize real-world input signals (containing signal and noise) which are representative of the expected use case.
The following are guidelines to determining optimal noise reduction property settings:

1. Configure the “Low Freq” and High Freq” properties based on the application requirements. Most speech applications require a bandwidth of about 300 Hz to 3400 Hz while consumer and portable audio applications demand the entire 100Hz to 20 kHz bandwidth. Also, if the biquad filter is used as in the provided process work flow its bandwidth must be adjusted to match the application.

2. Initialize the threshold: As previously noted, threshold relates to the difference between the estimated noise floor and how much above that noise floor the input must be in order to be treated as ‘signal’ rather than ‘noise’. It is desirable to establish a threshold setting such that the signal of interest (i.e. speech in a cell phone) commonly lies above it. A signal level that falls below the threshold relative to the estimated noise floor will be attenuated. This is undesirable. Thus, if you have high level ambient noise there will be little separation between the signal and estimated noise floor. Thus, the threshold should be set low since we always want the signal of interest to tend towards lying above the threshold. If you have low level ambient noise, the threshold can be set high since the signal is relatively higher and will tend to lie above the threshold. The idea is to best match the threshold to the relative level of the signal above the noise floor.

3. Initialize the attenuation: It is desirable to have established a threshold setting prior to finalizing the attenuation setting. The attenuation property determines how much noise reduction to apply when the input signal falls below the threshold. It is recommended to start the attenuation in the middle (10 dB). This should reduce noise by a modest amount. If the attenuation level is set too high, distortion of the signal may occur. It is important to evaluate the impact on both the signal and noise content to determine the optimal attenuation setting. If the signal quality is not impacted at this level the attenuation may be increased. Increase the attenuation setting to generate the optimal amount of noise reduction, but reduce it if signal distortion becomes unacceptable. The process to establish the optimal attenuation setting becomes an iterative process whereby the noise attenuation is balanced against the signal distortion for each subsequent modification of the attenuation setting. As with the threshold, it is best to ultimately set the Attenuation at a level that provides the maximum noise reduction, but has little to no impact on the signal quality.
6 Noise Reduction System Requirements

The Noise Reduction component has the following resource requirements.

Table 1. Noise Reduction Resource Requirements

<table>
<thead>
<tr>
<th>NR</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>miniDSP_D</td>
<td>Instruction Words 546</td>
</tr>
<tr>
<td></td>
<td>Data RAM 144</td>
</tr>
<tr>
<td></td>
<td>Coefficient RAM 48</td>
</tr>
<tr>
<td></td>
<td>Cycles/Frame 458</td>
</tr>
</tbody>
</table>

The following displays the available resources on the target processors.

Table 2. Available Processor Resource

<table>
<thead>
<tr>
<th>NR</th>
<th>TSC2117</th>
<th>AIC36</th>
<th>AIC3254</th>
</tr>
</thead>
<tbody>
<tr>
<td>miniDSP_A</td>
<td>Instruction Words 384</td>
<td>512</td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td>Data RAM 256</td>
<td>256</td>
<td>896</td>
</tr>
<tr>
<td></td>
<td>Coefficient RAM 128</td>
<td>128</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>Cycles/Frame 384</td>
<td>512</td>
<td>904</td>
</tr>
<tr>
<td>miniDSP_D</td>
<td>Instruction Words 1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td>Data RAM 896</td>
<td>896</td>
<td>896</td>
</tr>
<tr>
<td></td>
<td>Coefficient RAM 512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>Cycles/Frame 1024</td>
<td>1024</td>
<td>32768</td>
</tr>
</tbody>
</table>
7 Conclusion

A Noise Reduction solution has been developed for the AIC3254, AIC36 and TSC2117 miniDSP processors operating in the PurePath Studio environment. The implementation provides an extensive number of parameter controls to customize the Noise Reduction characteristics for a multitude of applications. Additionally, an extensive amount of additional resources are made available for inclusion of supplementary processing features.

8 References


