Workshop:
Audio Weaver on the STM32F407 Discovery Board

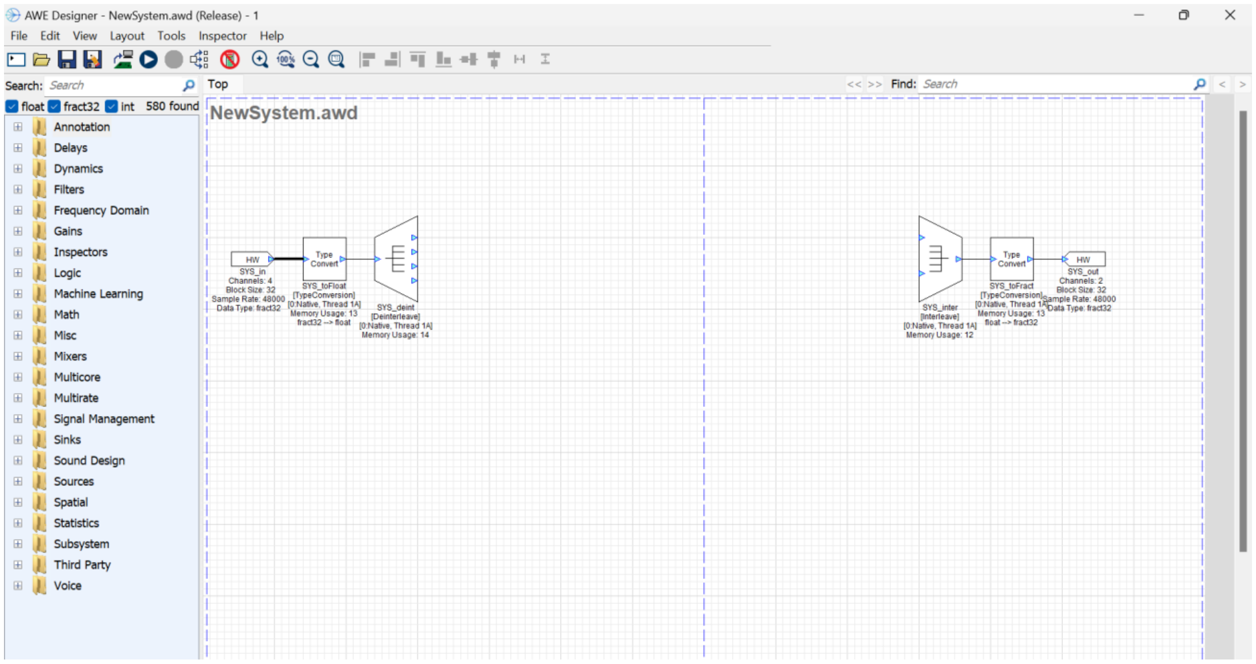
- Creating an Audio Weaver app using AWE Designer
- Developing Audio Weaver apps for PC
- Testing app on STM32F407
- Generating target files for STM32F407
- Implementing Audio Weaver app in STM32CubeIDE

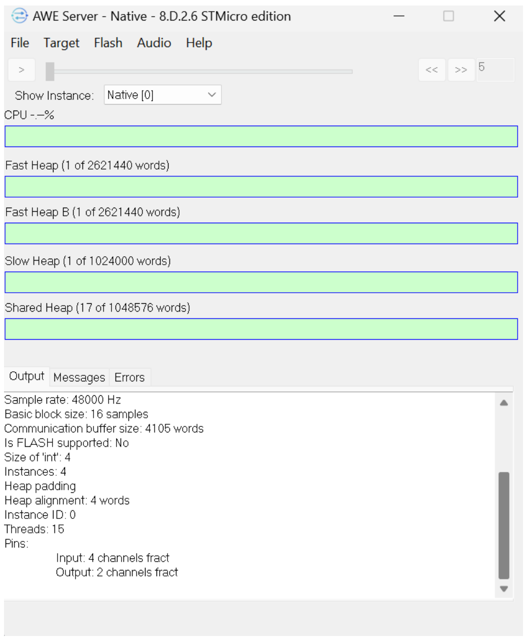
Creating an Audio Weaver app using AWE Designer

On your desktop PC, you will find an STMicro icon. This software is the ST Edition of AWE Designer, a subset of the Standard Edition, and free to users of STMicro products. Audio Weaver is a platform for developing audio applications in a graphical design framework that can generate code for standalone microcontroller projects.

STMicro
8.D.2.6

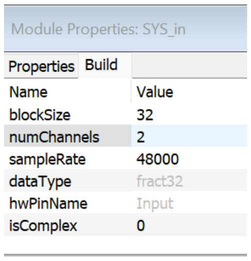
Opening AWE Designer launches two windows: the AWE Designer window and the AWE Server window.




AWE Server - Native - 8.D.2.6 STMicro edition
File Target Flash Audio Help
Show Instance: Native [0]
CPU -.--%
Fast Heap (1 of 2621440 words)
Fast Heap B (1 of 2621440 words)
Slow Heap (1 of 1024000 words)
Shared Heap (17 of 1048576 words)
Output Messages Errors

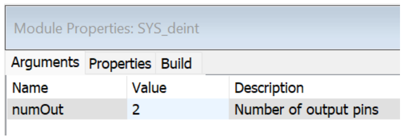
```
Sample rate: 48000 Hz
Basic block size: 16 samples
Communication buffer size: 4105 words
Is FLASH supported: No
Size of 'int': 4
Instances: 4
Heap padding
Heap alignment: 4 words
Instance ID: 0
Threads: 15
Pins:
        Input: 4 channels fract
        Output: 2 channels fract
```

To run a stereo application, the number of input channels should be 2, not 4. Right-click on the input HW block and choose View Properties. On the Build tab, change the numChannels to 2.



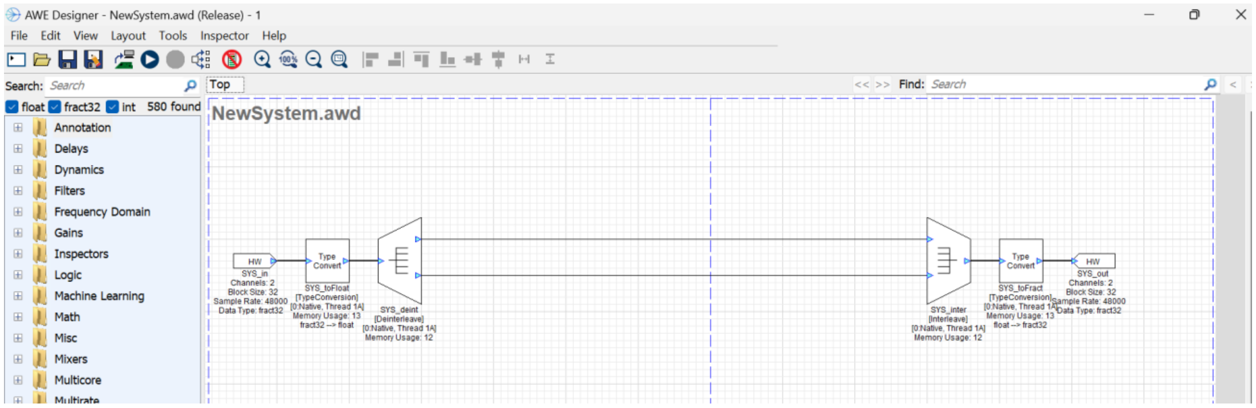
Name	Value
blockSize	32
numChannels	2
sampleRate	48000
dataType	fract32
hwPinName	Input
isComplex	0

The SYS_deint deinterleaver block separates left and right stereo channels so they can be processed individually. Right-click on this block and View Properties. On the Arguments tab, change numOut to 2.

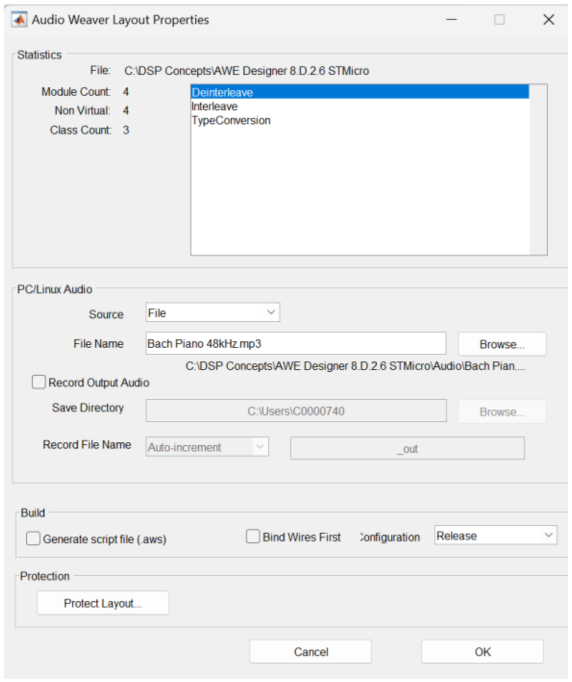


Name	Value	Description
numOut	2	Number of output pins

Connect two wires across from the deinterleaver block to the SYS_int interleaver block, to create a talk-through app.



Click Layout – Layout Properties. The input source is an .mp3 file containing Bach music.



Statistics

File: C:\DSP Concepts\AWE Designer 8.D.2.6 STMicro

Module Count: 4
Non Virtual: 4
Class Count: 3

Deinterleave
Interleave
TypeConversion

PC/Linux Audio

Source: File
File Name: Bach Piano 48kHz.mp3
C:\DSP Concepts\AWE Designer 8.D.2.6 STMicro\Audio\Bach Pian...
☐ Record Output Audio
Save Directory: C:\Users\C0000740
Record File Name: Auto-increment _out

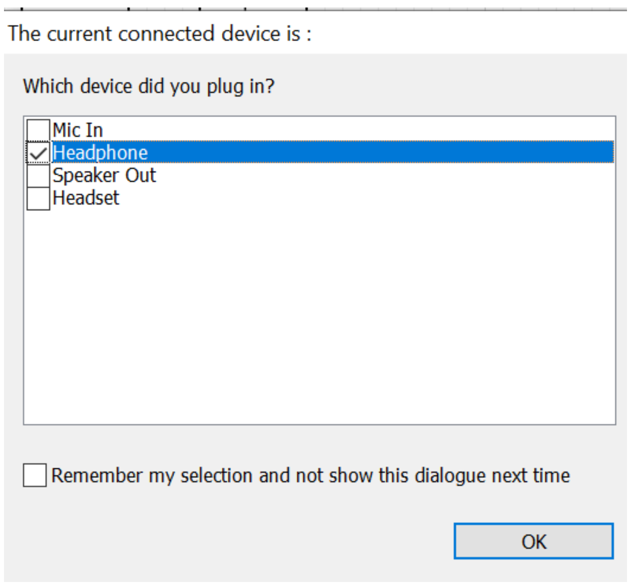
Build

☐ Generate script file (.aws) ☐ Bind Wires First Configuration: Release

Protection

Click OK.

Plug a set of headphones into your PC’s headphone jack. If your PC asks, select Headphone.


The current connected device is :

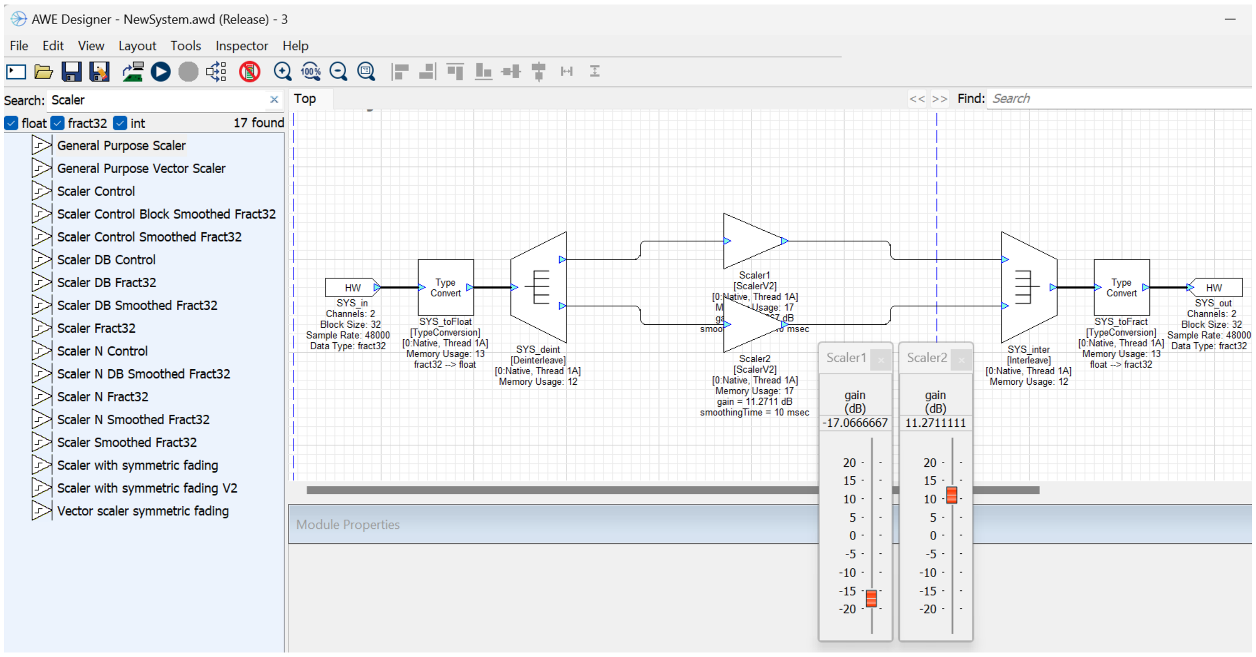
Which device did you plug in?

- [ ] Mic In
- [✓] Headphone
- [ ] Speaker Out
- [ ] Headset

- [ ] Remember my selection and not show this dialogue next time

In Audio Weaver, click the blue arrow to Run your app. You should hear Bach music. Click the red square to Halt.

In the left pane of AWE Designer, type Scaler into the Search box. Add a General Purpose Scaler for each channel. Double-click to open. Run your app. Vary the gains for each channel as you listen. One excellent feature of Audio Weaver is that you can change values of variables in real time. You can apply an extremely low gain to one of the channels to effectively remove it from the output. With high gain you will hear distortion in the output.


AWE Designer - NewSystem.awd (Release) - 3

Search: Scaler

[✓] float [✓] fract32 [✓] int   17 found

- General Purpose Scaler
- General Purpose Vector Scaler
- Scaler Control
- Scaler Control Block Smoothed Fract32
- Scaler Control Smoothed Fract32
- Scaler DB Control
- Scaler DB Fract32
- Scaler DB Smoothed Fract32
- Scaler Fract32
- Scaler N Control
- Scaler N DB Smoothed Fract32
- Scaler N Fract32
- Scaler N Smoothed Fract32
- Scaler Smoothed Fract32
- Scaler with symmetric fading
- Scaler with symmetric fading V2
- Vector scaler symmetric fading

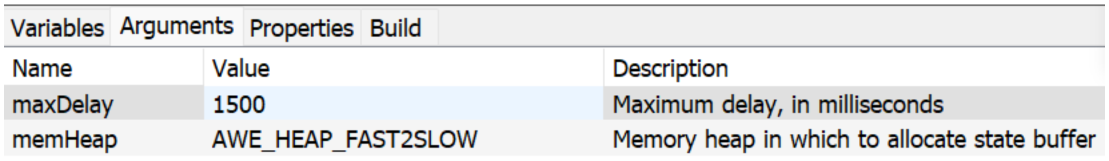
Scaler1: gain (dB) -17.0666667

Scaler2: gain (dB) 11.2711111

Remember to save any Audio Weaver app that you may want to use again later. By default apps are saved to `C:\DSP Concepts\AWE Designer 8.D.2.6 STMicro`.

Developing Audio Weaver apps for PC

AWE Designer has many processing blocks to choose from. For example, you can also experiment with the Millisecond Delay block. In the properties for the block change the default Maximum Delay from 500 msec to a larger number, perhaps 1500 msec.

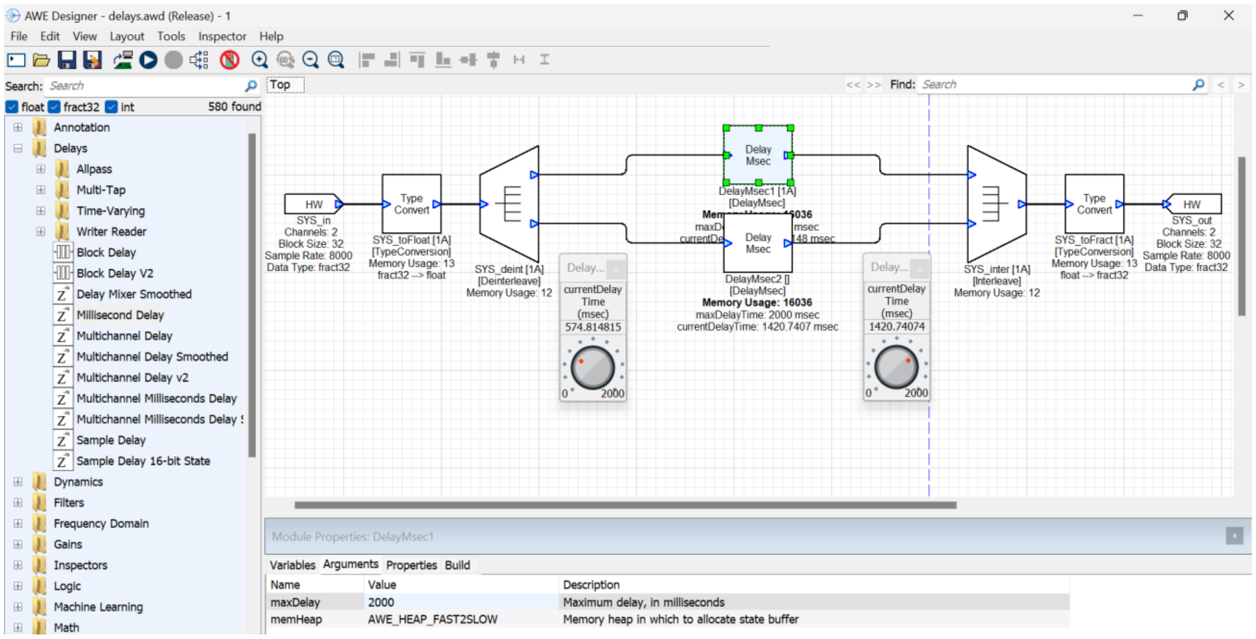


Variables | Arguments | Properties | Build

Name	Value	Description
maxDelay	1500	Maximum delay, in milliseconds
memHeap	AWE_HEAP_FAST2SLOW	Memory heap in which to allocate state buffer

Place one Millisecond Delay block for each channel. Double-click the block to open it. Set different delays for each channel. Run your app. Can you hear different delays for left and right channels? Try varying the delays for each channel while you listen.

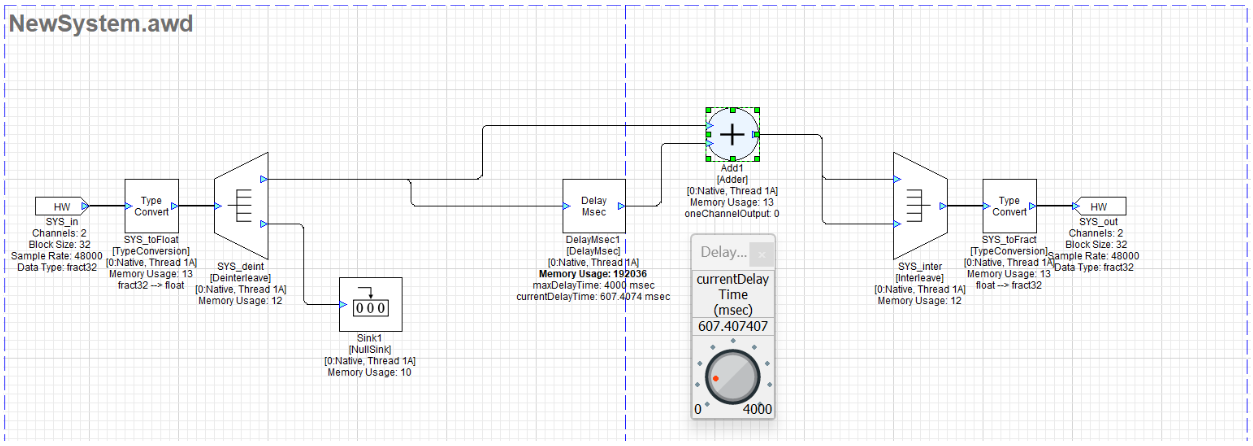
Save your delay app.


AWE Designer - delays.awd (Release) - 1
File Edit View Layout Tools Inspector Help
Search:
float fract32 int 580 found
Annotation
Delays
Allpass
Multi-Tap
Time-Varying
Writer Reader
Block Delay
Block Delay V2
Delay Mixer Smoothed
Millisecond Delay
Multichannel Delay
Multichannel Delay Smoothed
Multichannel Delay v2
Multichannel Milliseconds Delay
Multichannel Milliseconds Delay
Sample Delay
Sample Delay 16-bit State
Dynamics
Filters
Frequency Domain
Gains
Inspectors
Logic
Machine Learning
Math
Top
Find:
HW
SYS_in Channels: 2 Block Size: 32 Sample Rate: 8000 Data Type: fract32
Type Convert
SYS_toFloat [1A] [TypeConversion] Memory Usage: 13 fract32 --> float
SYS_deint [1A] [Deinterleave] Memory Usage: 12
Delay Msec
DelayMsec1 [1A] [DelayMsec]
DelayMsec2 [] [DelayMsec] Memory Usage: 16036 maxDelayTime: 2000 msec currentDelayTime: 1420.7407 msec
currentDelay Time (msec) 574.814815
0 2000
currentDelay Time (msec) 1420.74074
0 2000
SYS_inter [1A] [Interleave] Memory Usage: 12
SYS_toFract [1A] [TypeConversion] Memory Usage: 13 float --> fract32
SYS_out Channels: 2 Block Size: 32 Sample Rate: 8000 Data Type: fract32
Module Properties: DelayMsec1

Variables | Arguments | Properties | Build

Name	Value	Description
maxDelay	2000	Maximum delay, in milliseconds
memHeap	AWE_HEAP_FAST2SLOW	Memory heap in which to allocate state buffer

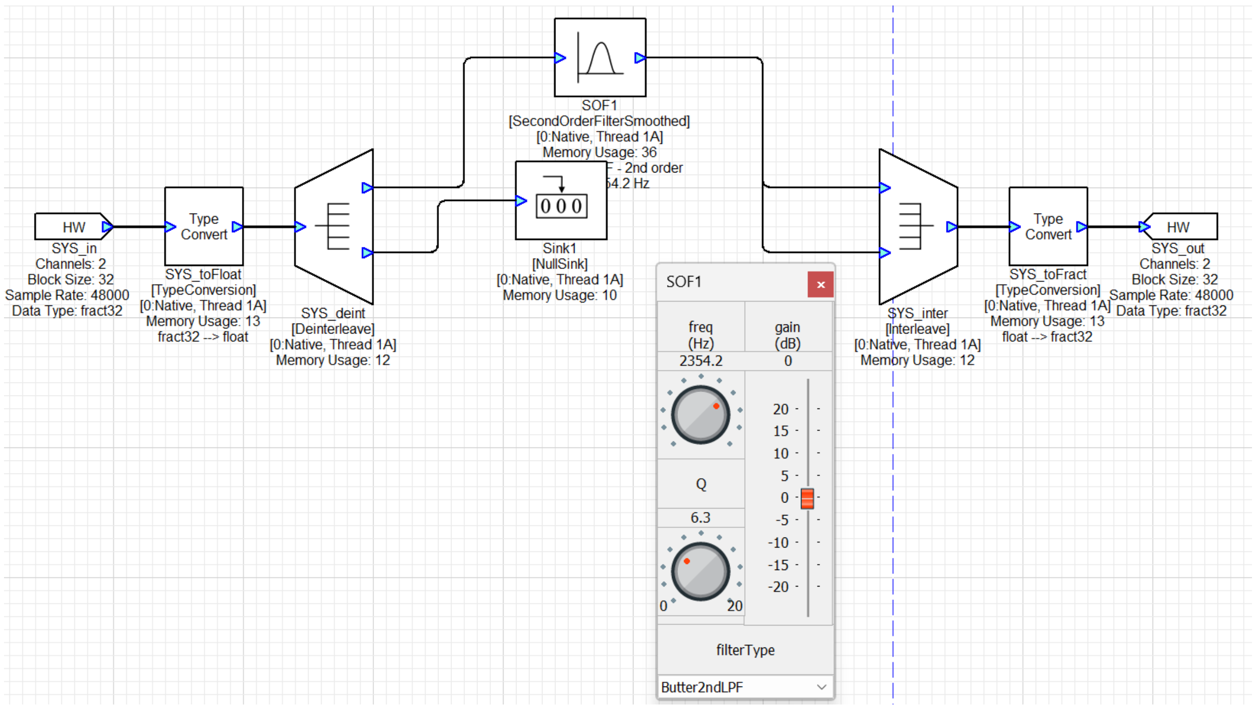
You can create an echo of an input by summing the original signal with a delayed version of itself. In this example, a Null Sink is used to remove the right channel, an Adder is used to sum two signals, and the Adder output is sent to both left and right outputs. Additional echoes can be created with more delay blocks, and the properties of the Adder block can be modified to permit more inputs.



Save your echo app.

Set up a new app. The Second Order Filter block is a useful block to experiment with. Add a filter block for the left channel. Send the right channel to Null Sink so you can hear the filter effects more clearly. Initially the Filter Type is set to Bypass. Change Filter Type to Butter2ndLPF (Butterworth second order low pass filter). Run your app. Vary the frequency, which is the cutoff frequency of your filter. When the cutoff frequency is low, the low pass filter is removing all of the signal except very low frequencies. You should hear very muffled sound. When the cutoff frequency is high, very little of the signal is removed by the low pass filter.

Halt your app. Change Filter Type to Butter2ndHPF (Butterworth second order high pass filter). Run your app. The high pass filter removes the frequencies above the cutoff frequency. When the cutoff frequency is low, the filter will have very little effect on your signal; when the cutoff frequency is high, you should hear only very high frequency remains of your signal.



Save your filtering app.

Testing app on STM32F407

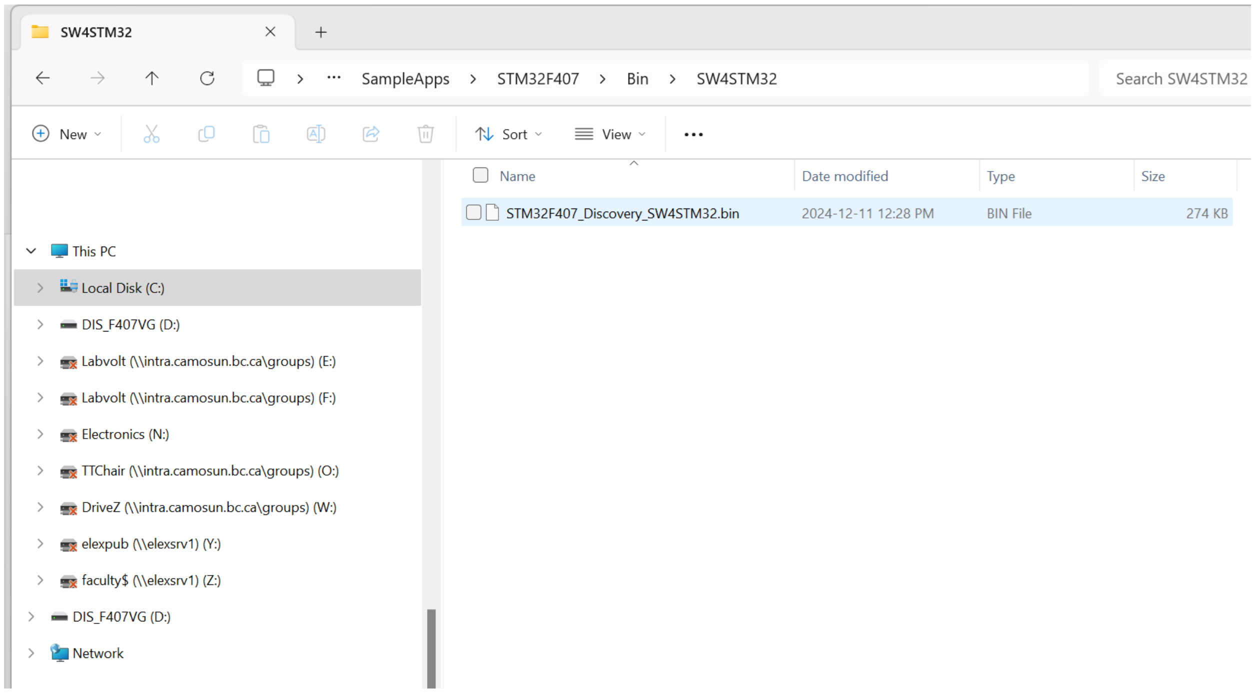
Up until now, Audio Weaver apps have been running on the PC. This is called Native connection. The apps can also run standalone on the STM32F407 Discovery board.

As an intermediate step to standalone operation, Audio Weaver can run code on the Discovery board, but still allow real time control from the PC. The value of this step is to verify that the app fits within the memory available on the board.

Plug a micro USB cable into one end of the Discovery board and a mini USB cable into the other end. Plug both USB cables into your PC.

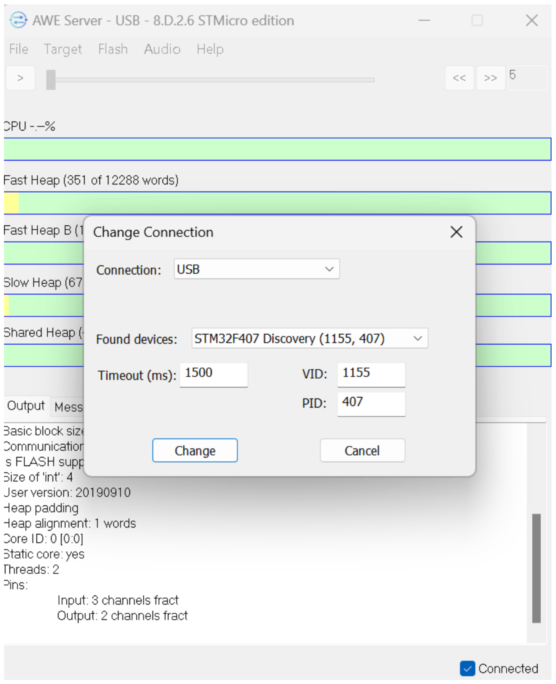
In the File Explorer on your PC, navigate to:
`C:\DSP Concepts\AWECore ST_EVAL_CortexM4 Release-8.D.2.7\SampleApps\STM32F407\Bin\SW4STM32`

Drag the file `STM32F407_Discovery_SW4STM32.bin` to the icon for the Discovery board in the left pane, `DIS_F407VG`.


SW4STM32
SampleApps > STM32F407 > Bin > SW4STM32
Search SW4STM32
New
Sort
View
This PC
Local Disk (C:)
DIS_F407VG (D:)
Labvolt (\\intra.camosun.bc.ca\groups) (E:)
Labvolt (\\intra.camosun.bc.ca\groups) (F:)
Electronics (N:)
TTChair (\\intra.camosun.bc.ca\groups) (O:)
DriveZ (\\intra.camosun.bc.ca\groups) (W:)
elexpub (\\elexsrv1) (Y:)
faculty$ (\\elexsrv1) (Z:)
DIS_F407VG (D:)
Network

Name	Date modified	Type	Size
STM32F407_Discovery_SW4STM32.bin	2024-12-11 12:28 PM	BIN File	274 KB

To run your Audio Weaver app on the Discovery board, go to the AWE Server window and choose Target – Change Connection. Select USB.



```
CPU -.–%
Fast Heap (351 of 12288 words)
Fast Heap B (1
Slow Heap (67
Shared Heap (
```

Change Connection

Connection: USB

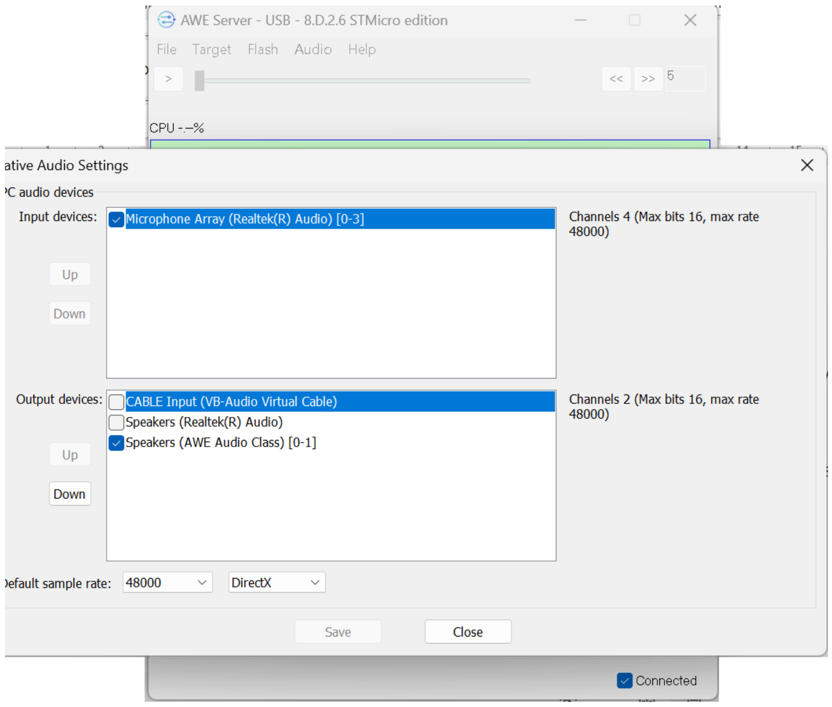
Found devices: STM32F407 Discovery (1155, 407)

Timeout (ms): 1500   VID: 1155

PID: 407

```
Basic block size
Communication
s FLASH supp
Size of 'int': 4
User version: 20190910
Heap padding
Heap alignment: 1 words
Core ID: 0 [0:0]
Static core: yes
Threads: 2
Pins:
        Input: 3 channels fract
        Output: 2 channels fract
```

Still in AWE Server, choose File – Native Audio Settings. Speakers (AWE Audio Class) and a microphone input should be checked off. The Discovery board appears to your PC like speakers. Uncheck other outputs.


AWE Server - USB - 8.D.2.6 STMicro edition
File Target Flash Audio Help
CPU -.-%
Up
Down
Save
Close
Connected

ative Audio Settings

PC audio devices

Input devices: ☑ Microphone Array (Realtek(R) Audio) [0-3] — Channels 4 (Max bits 16, max rate 48000)

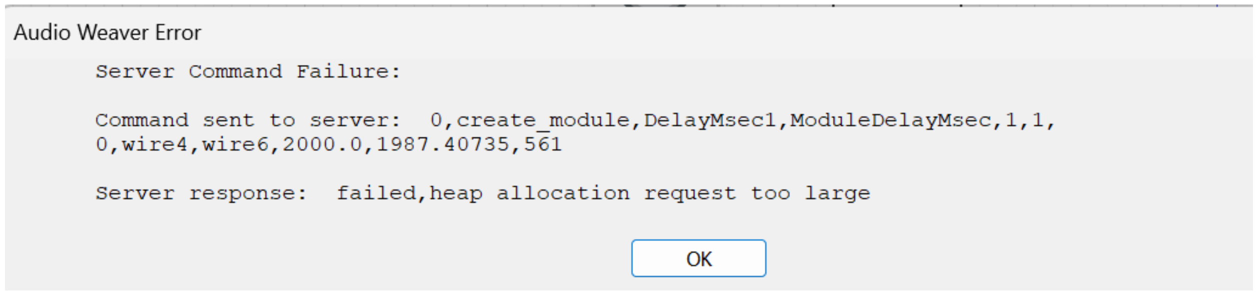
Output devices: ☐ CABLE Input (VB-Audio Virtual Cable) — Channels 2 (Max bits 16, max rate 48000)
☐ Speakers (Realtek(R) Audio)
☑ Speakers (AWE Audio Class) [0-1]

efault sample rate: 48000 DirectX

Plug a set of headphones into the jack on the Discovery board.

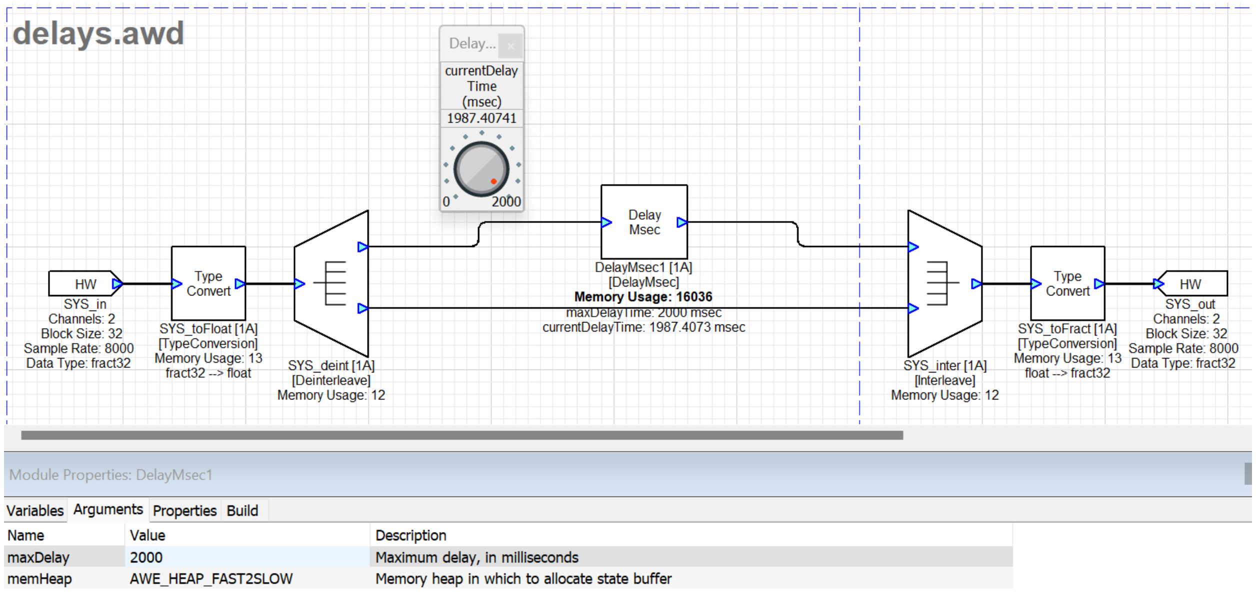
Play something on YouTube. Because you are running on the Discovery board, inputs can no longer come from the Bach piano file.

Run your Audio Weaver delay (or echo) app. You may find that your app, with a sampling frequency of 48 kHz, and maximum delays of 2 seconds on each channel, will not run. You may get an error declaring that the heap allocation request is too large.


```
Audio Weaver Error
        Server Command Failure:

        Command sent to server:  0,create_module,DelayMsec1,ModuleDelayMsec,1,1,
        0,wire4,wire6,2000.0,1987.40735,561

        Server response:  failed,heap allocation request too large
```

An adjusted layout might delay just one channel and reduce the sampling frequency to 8 kHz, or delay both channels but reduce maximum delay for each. Make some changes to your layout until it will run on the Discovery board without errors. This is not yet standalone operation, so you will still be able to control the delays from your PC.

Change your sampling rate in the properties for the HW input block. The sampling rate for the output block will change as soon as you re-run your app.

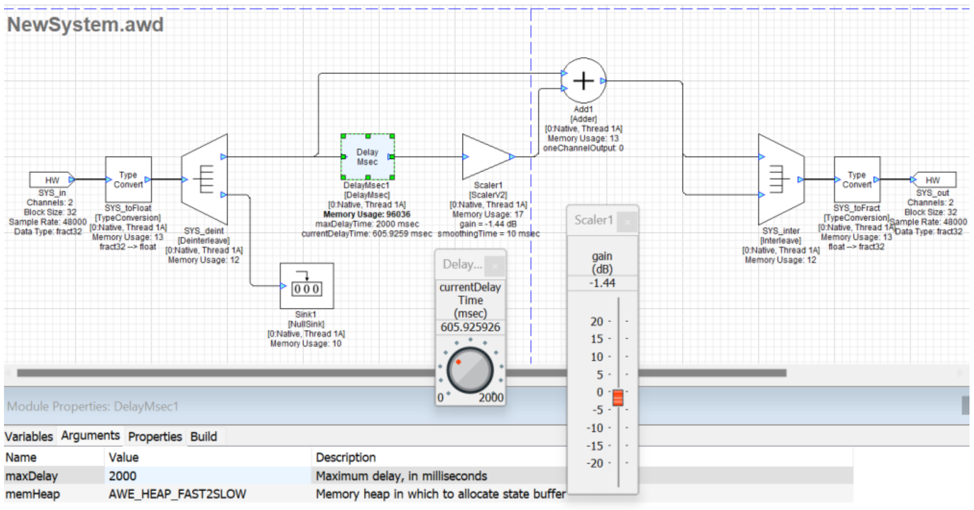


Name	Value	Description
maxDelay	2000	Maximum delay, in milliseconds
memHeap	AWE_HEAP_FAST2SLOW	Memory heap in which to allocate state buffer

If the volume of your output is too high, add an overall Scaler and set the gain below 0 dB.

While you are testing, be sure something is playing on YouTube. If you are using a filter, an online tone generator https://www.szynalski.com/tone-generator/ might be useful. To hear delays and echoes, spoken sources might be more helpful than music, for example, a newscast.

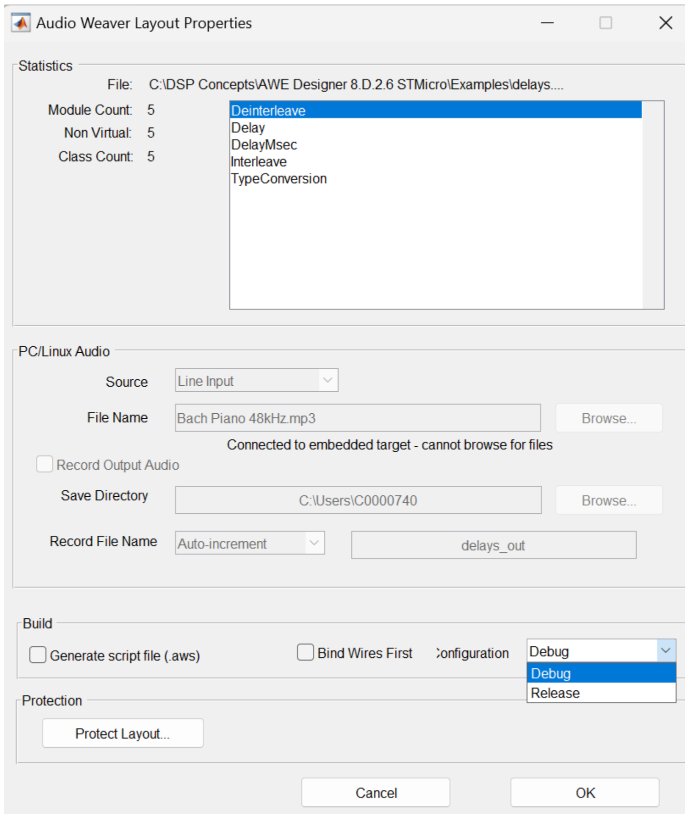
Experiment with running your delay, echo or filtering app on the Discovery board until you have strong and obvious effects in the output signal, e.g. noticeable delays, low cutoff frequency for low pass filter, high cutoff frequency for high pass filter. You will generate code for standalone operation, but once the app code is on the board, the variables are fixed at the values they had when you generated code, and they can no longer be varied in real time. The ability to change the variables of your app on the Discovery board in real time is the topic of another workshop.



Name	Value	Description
maxDelay	2000	Maximum delay, in milliseconds
memHeap	AWE_HEAP_FAST2SLOW	Memory heap in which to allocate state buffer

Generating target files for STM32F407

In AWE Designer, go to Layout – Layout Properties. Change Build Configuration from Release to Debug, and click OK.


Audio Weaver Layout Properties

Statistics
File: C:\DSP Concepts\AWE Designer 8.D.2.6 STMicro\Examples\delays....
Module Count: 5
Non Virtual: 5
Class Count: 5

Deinterleave
Delay
DelayMsec
Interleave
TypeConversion

PC/Linux Audio
Source: Line Input
File Name: Bach Piano 48kHz.mp3
Connected to embedded target - cannot browse for files
Record Output Audio
Save Directory: C:\Users\C0000740
Record File Name: Auto-increment delays_out

Build
Generate script file (.aws)
Bind Wires First
Configuration: Debug
Debug
Release

Protection
Protect Layout...

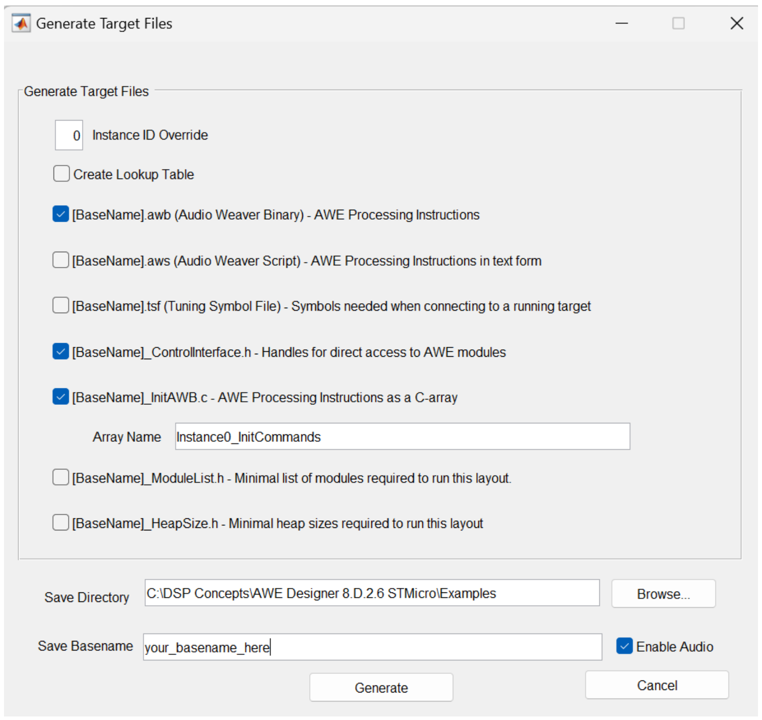
Cancel OK

Next, choose Tools – Generate Target Files.

Be sure `BaseName.awb`, `BaseName_ControlInterface.h`, and `BaseName_InitAWB.c` are all selected. Also ensure that Enable Audio is checked. The Save Directory should be:

```
C:\DSP Concepts\AWE Designer 8.D.2.6 STMicro\Examples
```

Choose a `Basename` that will help you recognize the files you are looking for.


Generate Target Files

Generate Target Files

0 Instance ID Override

☐ Create Lookup Table

☑ [BaseName].awb (Audio Weaver Binary) - AWE Processing Instructions

☐ [BaseName].aws (Audio Weaver Script) - AWE Processing Instructions in text form

☐ [BaseName].tsf (Tuning Symbol File) - Symbols needed when connecting to a running target

☑ [BaseName]_ControlInterface.h - Handles for direct access to AWE modules

☑ [BaseName]_InitAWB.c - AWE Processing Instructions as a C-array

Array Name: Instance0_InitCommands

☐ [BaseName]_ModuleList.h - Minimal list of modules required to run this layout.

☐ [BaseName]_HeapSize.h - Minimal heap sizes required to run this layout

Save Directory: C:\DSP Concepts\AWE Designer 8.D.2.6 STMicro\Examples

Save Basename: your_basename_here

☑ Enable Audio

Click Generate. This will produce a group of files in your Save Directory. For example, with basename = `your_basename_here` you will see:

- your_basename_here.awb
- your_basename_here_ControlInterface.h
- your_basename_here_ControlInterface_AWE6.h
- your_basename_here_InitAWB.c
- your_basename_here_InitAWB.h

After successful generation of target files, go to your Save Directory. Copy the five files to:

```
C:\DSP Concepts\AWECore ST_EVAL_CortexM4 Release-
8.D.2.7\SampleApps\STM32F407\Source
```

If a previous set of target files are present, they may be removed.

Implementing Audio Weaver app in STM32CubeIDE

STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers.

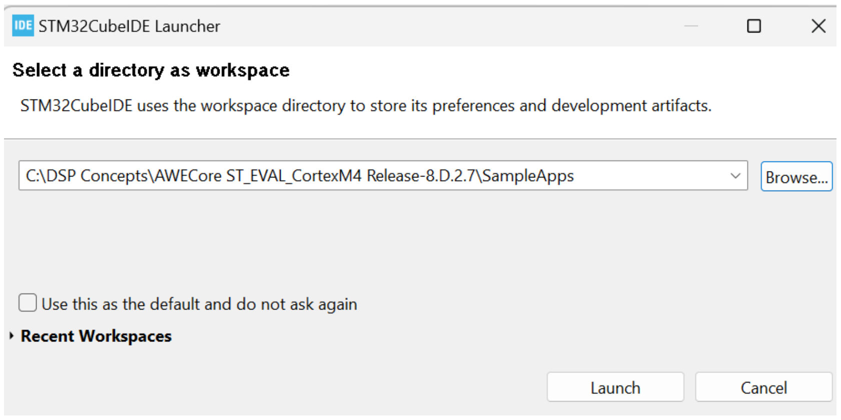
Ensure a micro USB cable is plugged into one end of the Discovery board and a mini USB cable is plugged into the other end. Plug both USB cables into your PC.

Open STM32CubeIDE by clicking on the STM32CubeIDE 1.17.0 icon.

For the workspace directory, enter:

```
C:\DSP Concepts\AWECore ST_EVAL_CortexM4 Release-8.D.2.7\SampleApps
```

Click Launch.

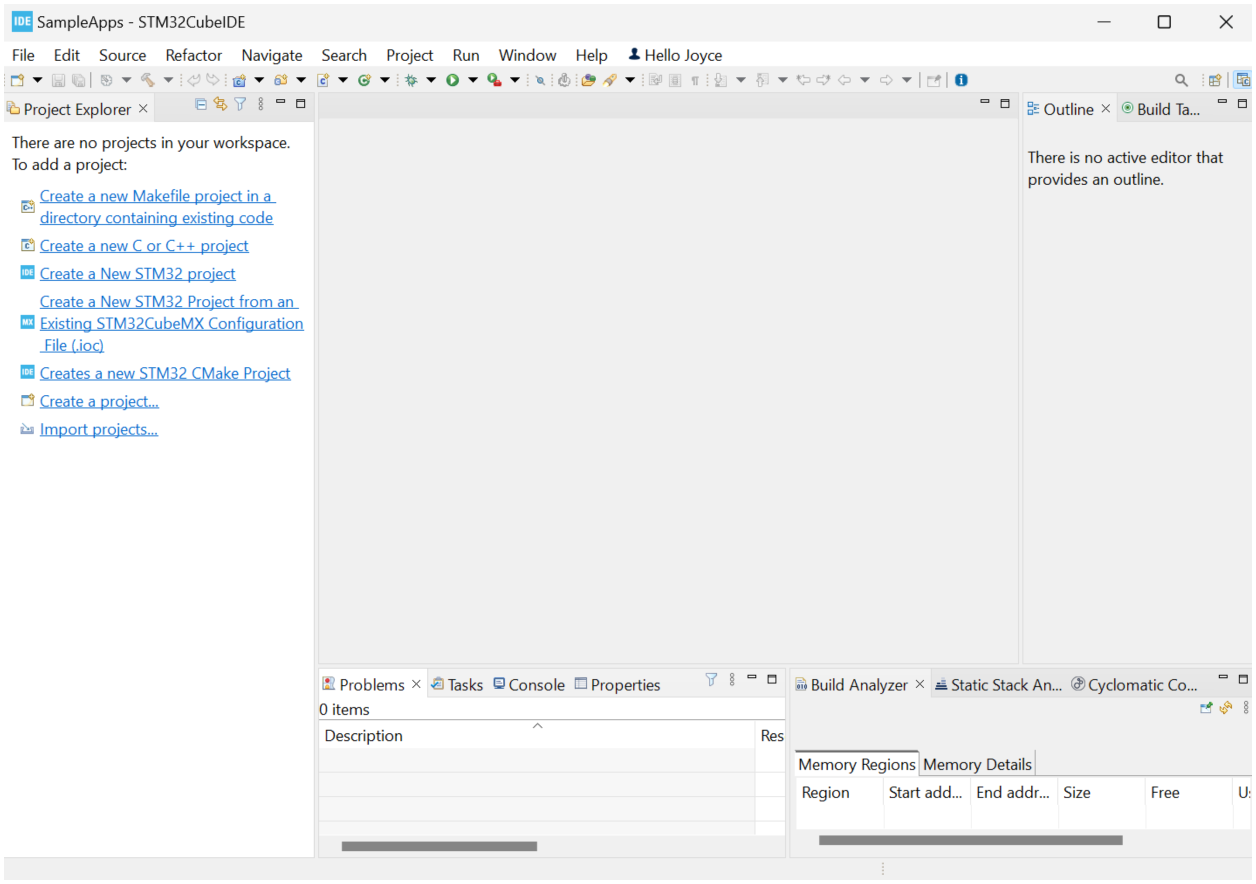

STM32CubeIDE Launcher

Select a directory as workspace

STM32CubeIDE uses the workspace directory to store its preferences and development artifacts.

C:\DSP Concepts\AWECore ST_EVAL_CortexM4 Release-8.D.2.7\SampleApps

Click x to close the Information Center window. Your window now looks like:



There are no projects in your workspace.
To add a project:

- Create a new Makefile project in a directory containing existing code
- Create a new C or C++ project
- Create a New STM32 project
- Create a New STM32 Project from an Existing STM32CubeMX Configuration File (.ioc)
- Creates a new STM32 CMake Project
- Create a project...
- Import projects...

There is no active editor that provides an outline.

0 items

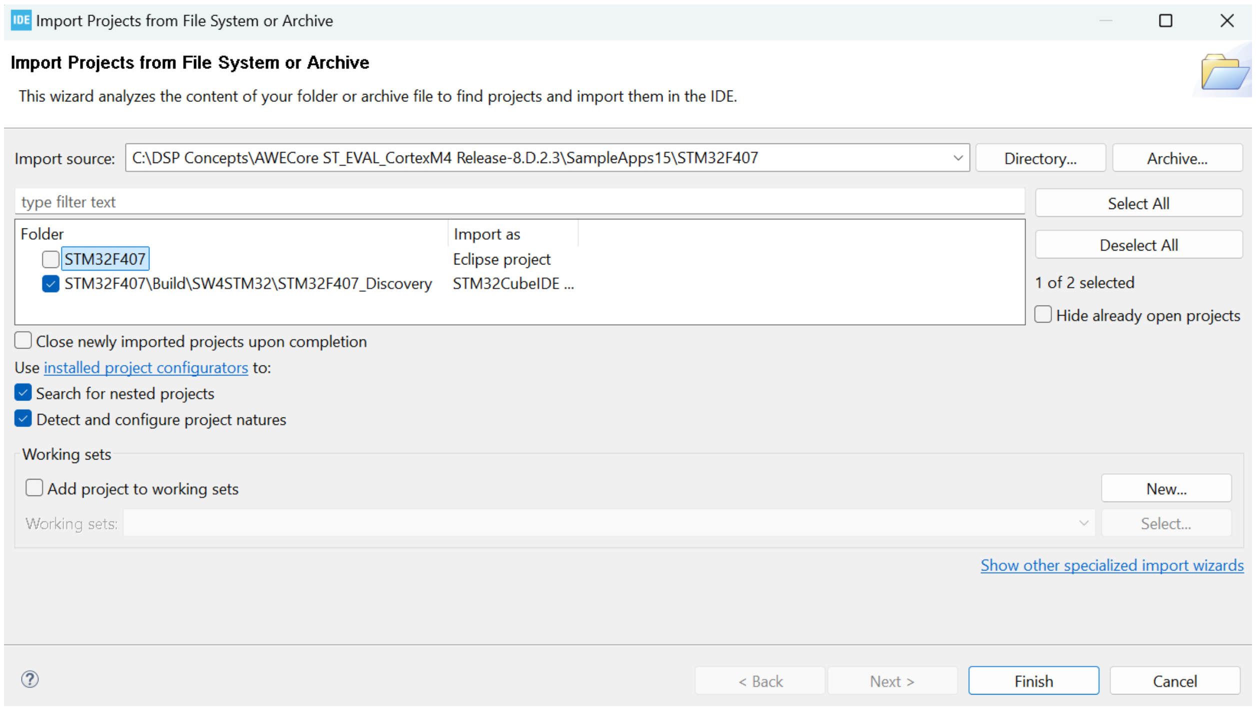
Description	Res

Region	Start add...	End addr...	Size	Free	U:

Choose File – Open Projects from File System. For Import Source, choose:

```
C:\DSP Concepts\AWECore ST_EVAL_CortexM4 Release-
8.D.2.7\SampleApps\STM32F407
```

Uncheck all folders except STM32F407_Discovery\Build\SW4STM32\STM32F407_Discovery. At the bottom of the window, click Finish.


Import Projects from File System or Archive

Import Projects from File System or Archive

This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.

Import source: C:\DSP Concepts\AWECore ST_EVAL_CortexM4 Release-8.D.2.3\SampleApps15\STM32F407

type filter text

Folder	Import as
☐ STM32F407	Eclipse project
☑ STM32F407\Build\SW4STM32\STM32F407_Discovery	STM32CubeIDE ...

1 of 2 selected

☐ Hide already open projects

☐ Close newly imported projects upon completion

Use installed project configurators to:

☑ Search for nested projects

☑ Detect and configure project natures

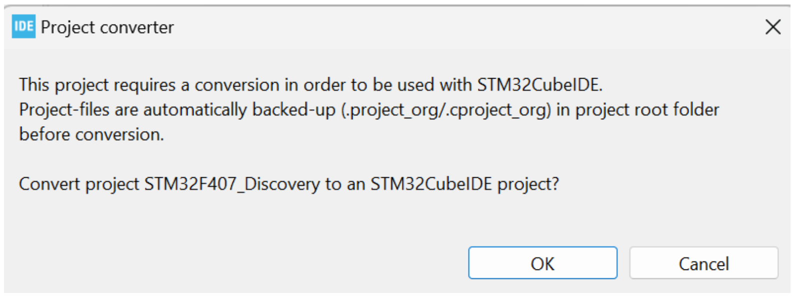
Working sets

☐ Add project to working sets

Working sets:

Show other specialized import wizards

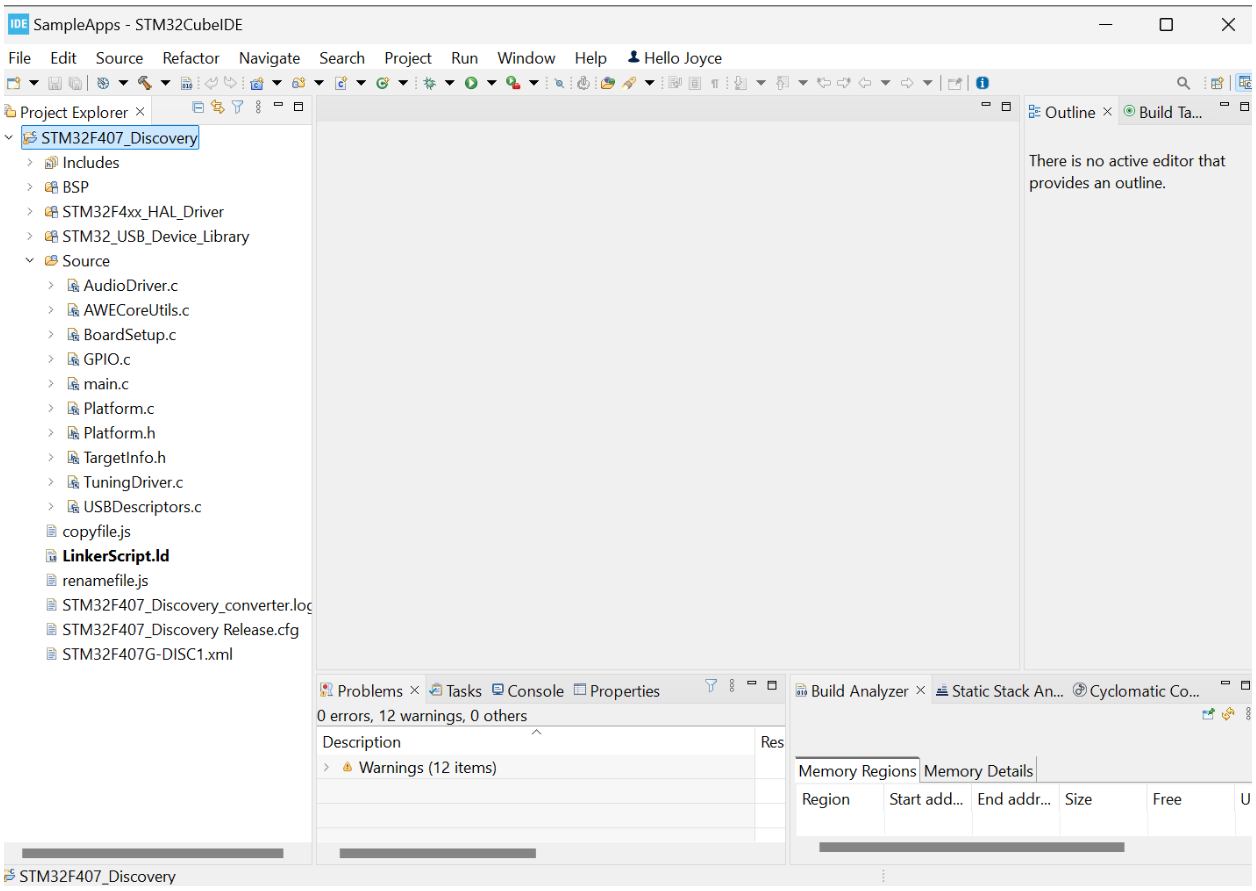
Agree to convert the project.


Project converter

This project requires a conversion in order to be used with STM32CubeIDE.
Project-files are automatically backed-up (.project_org/.cproject_org) in project root folder before conversion.

Convert project STM32F407_Discovery to an STM32CubeIDE project?

In the Project Explorer at the left, double-click on the project name STM32F407_Discovery to expand the project contents. Double-click the Source folder to open it.



STM32F407_Discovery
- Includes
- BSP
- STM32F4xx_HAL_Driver
- STM32_USB_Device_Library
- Source
  - AudioDriver.c
  - AWECoreUtils.c
  - BoardSetup.c
  - GPIO.c
  - main.c
  - Platform.c
  - Platform.h
  - TargetInfo.h
  - TuningDriver.c
  - USBDescriptors.c
- copyfile.js
- **LinkerScript.ld**
- renamefile.js
- STM32F407_Discovery_converter.log
- STM32F407_Discovery Release.cfg
- STM32F407G-DISC1.xml

There is no active editor that provides an outline.

0 errors, 12 warnings, 0 others

Description	Res
Warnings (12 items)	

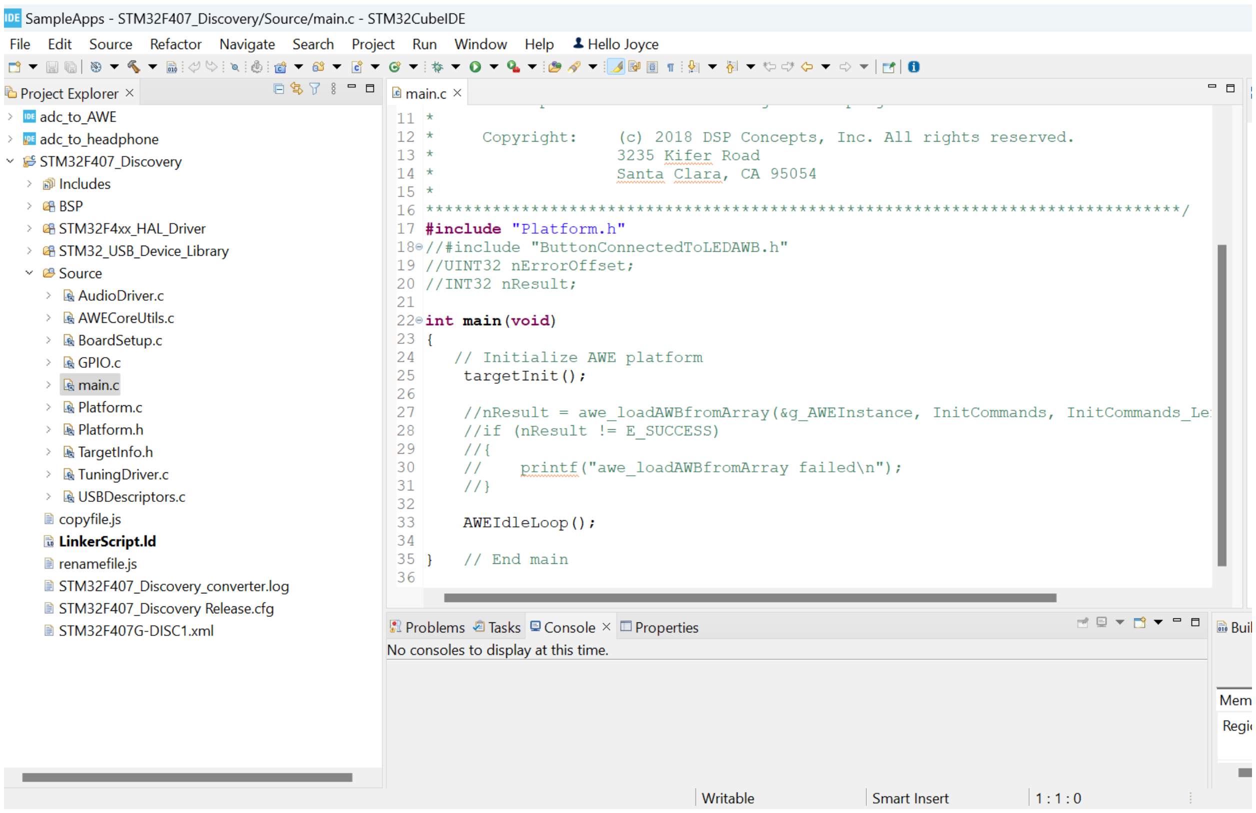
Memory Regions | Memory Details

Region	Start add...	End addr...	Size	Free	U

STM32F407_Discovery

At the moment your generated target files are not in this Source folder. Locate your five generated target files using your File Explorer. Copy and paste them into the Source folder in STM32CubeIDE. You should see your target files in the left pane. (Remove any target files from previous projects.)

Go to `main.c` in the Source folder in the left pane of STM32CubeIDE. Double-click on it to open it.



```
 *
 *     Copyright:    (c) 2018 DSP Concepts, Inc. All rights reserved.
 *                   3235 Kifer Road
 *                   Santa Clara, CA 95054
 *
 ******************************************************************************/
#include "Platform.h"
//#include "ButtonConnectedToLEDAWB.h"
//UINT32 nErrorOffset;
//INT32 nResult;

int main(void)
{
   // Initialize AWE platform
    targetInit();

    //nResult = awe_loadAWBfromArray(&g_AWEInstance, InitCommands, InitCommands_Le
    //if (nResult != E_SUCCESS)
    //{
    //    printf("awe_loadAWBfromArray failed\n");
    //}

    AWEIdleLoop();

}   // End main
```

No consoles to display at this time.

In `main.c`, after the line `#include "Platform.h"` add the line (with your basename):

```
#include "your_basename_here_InitAWB.h"
```

Uncomment:

```
UINT32 nErrorOffset;
INT32 nResult;
```

In the main function, uncomment the lines:

```
nResult = awe_loadAWBfromArray(&g_AWEInstance, InitCommands,
InitCommands_Len, &nErrorOffset);
if (nResult != E_SUCCESS)
{
    printf("awe_loadAWBfromArray failed\n");
}
```

Edit the first of the lines to read:

```
nResult = awe_loadAWBfromArray(&g_AWEInstance, Instance0_InitCommands,
Instance0_InitCommands_Len, &nErrorOffset);
```

File – Save.

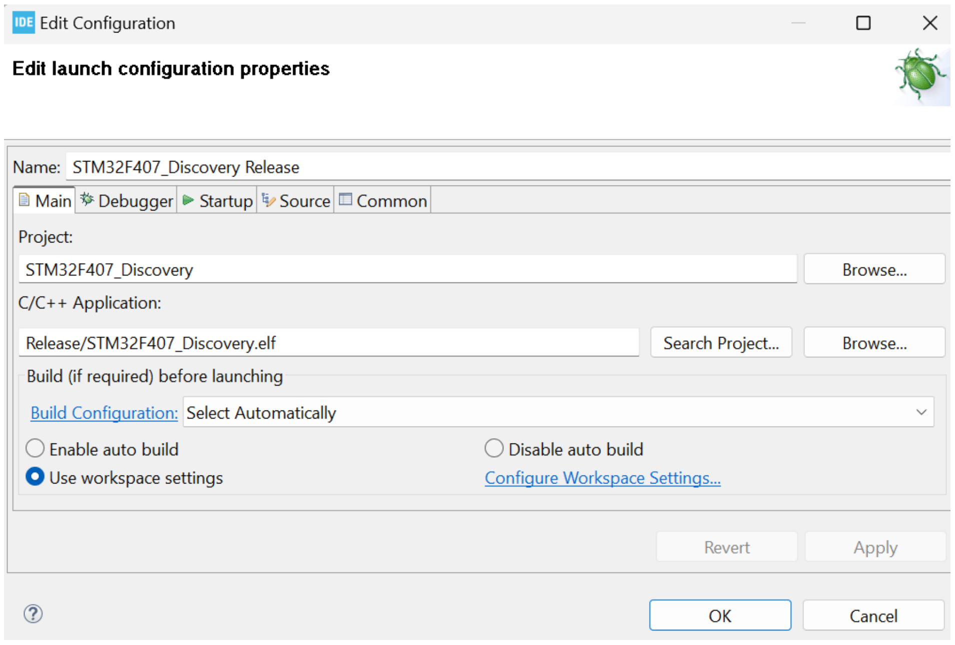
Make sure that `main.c` now looks like this:

```
#include "Platform.h"
#include "your_basename_here_InitAWB.h"
```

```
//#include "ButtonConnectedToLEDAWB.h"
UINT32 nErrorOffset;
INT32 nResult;

int main(void)
{
   // Initialize AWE platform
     targetInit();

    nResult = awe_loadAWBfromArray(&g_AWEInstance,
Instance0_InitCommands, Instance0_InitCommands_Len, &nErrorOffset);
    if (nResult != E_SUCCESS)
    {
        printf("awe_loadAWBfromArray failed\n");
    }

    AWEIdleLoop();

}   // End main
```

Click on the hammer to compile your project. Don't be concerned about warnings.

Choose Run – Debug. It will take a few minutes for the debugger to load the code onto the Discovery board. You can agree to Edit Launch configuration properties.



Name: STM32F407_Discovery Release

Project: STM32F407_Discovery

C/C++ Application: Release/STM32F407_Discovery.elf

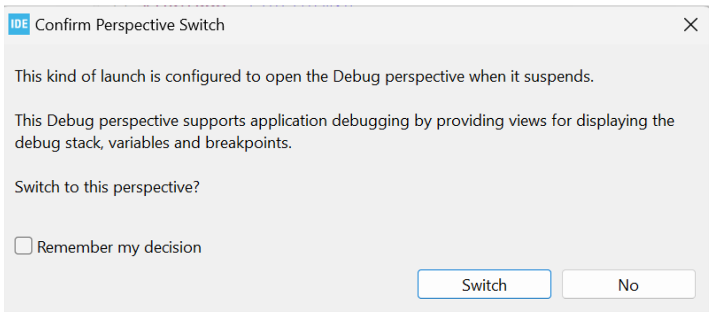
Build (if required) before launching

Build Configuration: Select Automatically

( ) Enable auto build ( ) Disable auto build

[✓ Use workspace settings] Configure Workspace Settings...

Confirm perspective switch.



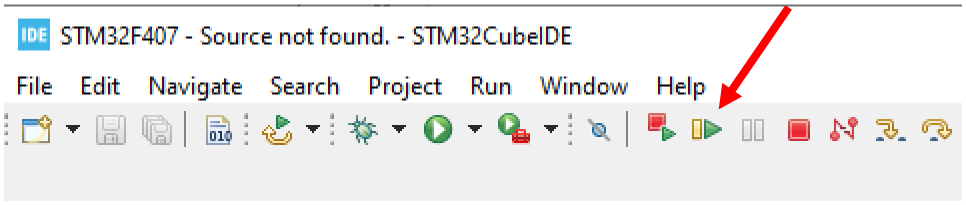
Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.

Switch to this perspective?

Once program download is successful, noted in the Console window at the bottom, click Play – Resume



STM32F407 - Source not found. - STM32CubeIDE

Make sure YouTube is playing something. You should be able to hear the effects of your app. The code is now standalone, except that power and YouTube music is coming from the PC. These could be replaced by a power bank and a smartphone, for example. You should be able to close Audio Weaver and still hear the effects.