Play Tones using eZdsp

EXPERIMENT 1.4

Propose of the experiment

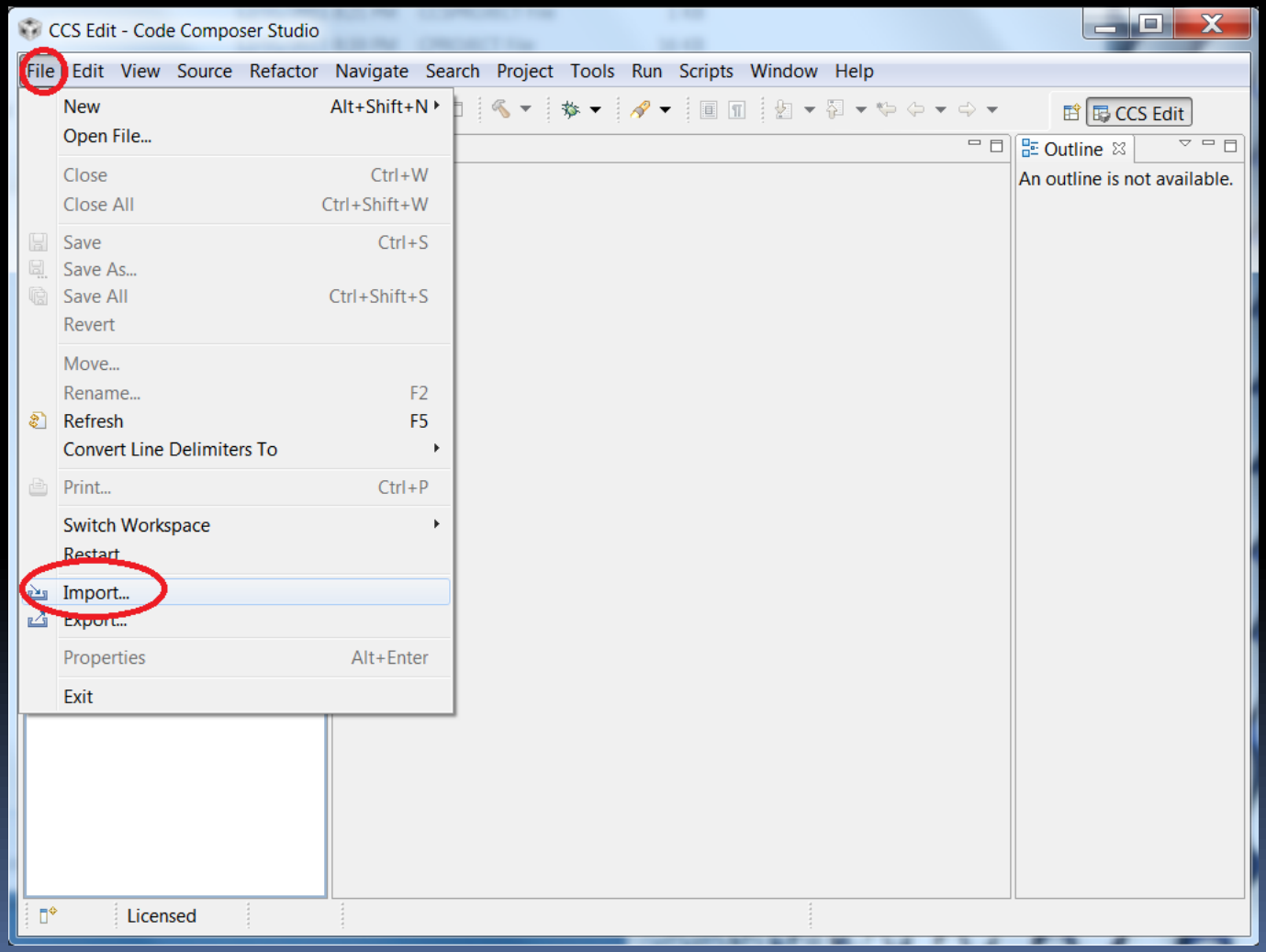
- Continue from previous experiments to get familiar with CCS environment
- Build a program that will play an audio tone through the audio output connector (Headphone output)
- Using the user interface (UI) from previous experiment to control the Play Tone program from CCS Console Window
- Introduction to the audio interface IC (AIC) used by C55x eZdsp USB Stick

Import an existing project

- Copy the zip file to work folder and unzip the file
- Start CCS
- Import the existing CCS Project Workspace as following steps

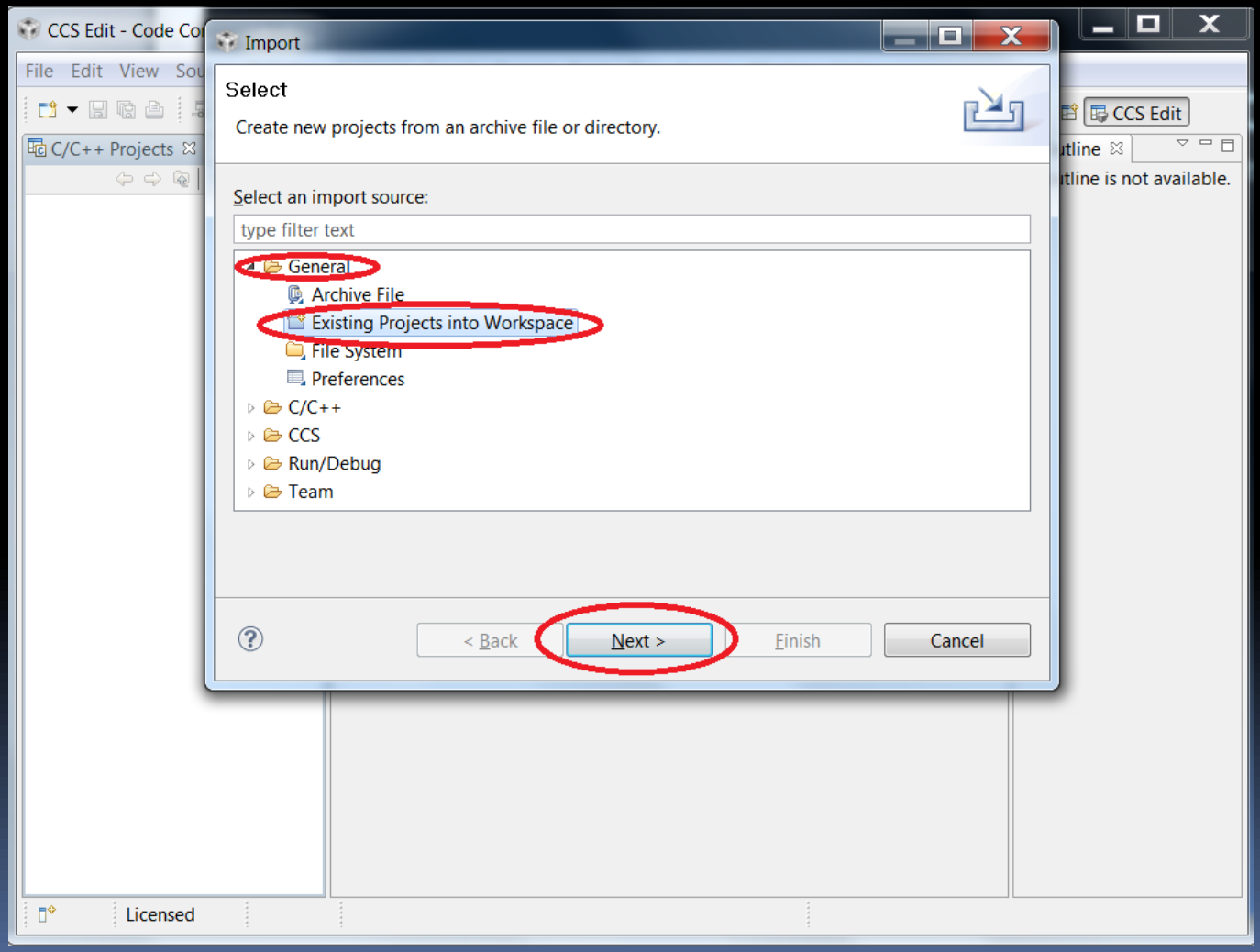
Import existing CCS project (1)

(File -> Import)



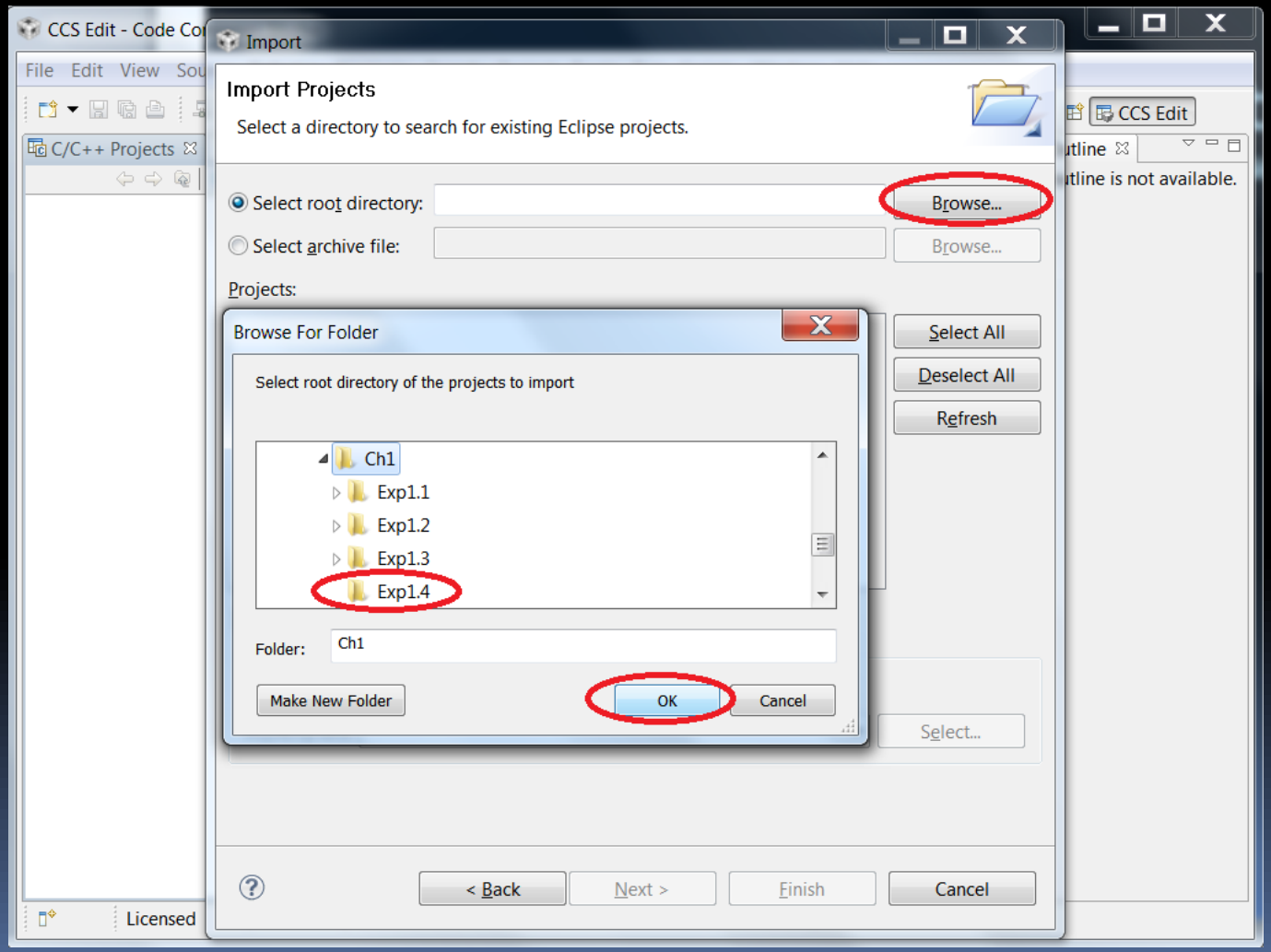
Import existing CCS project (2)

(General, select Existing Project into Workspace, then Next)



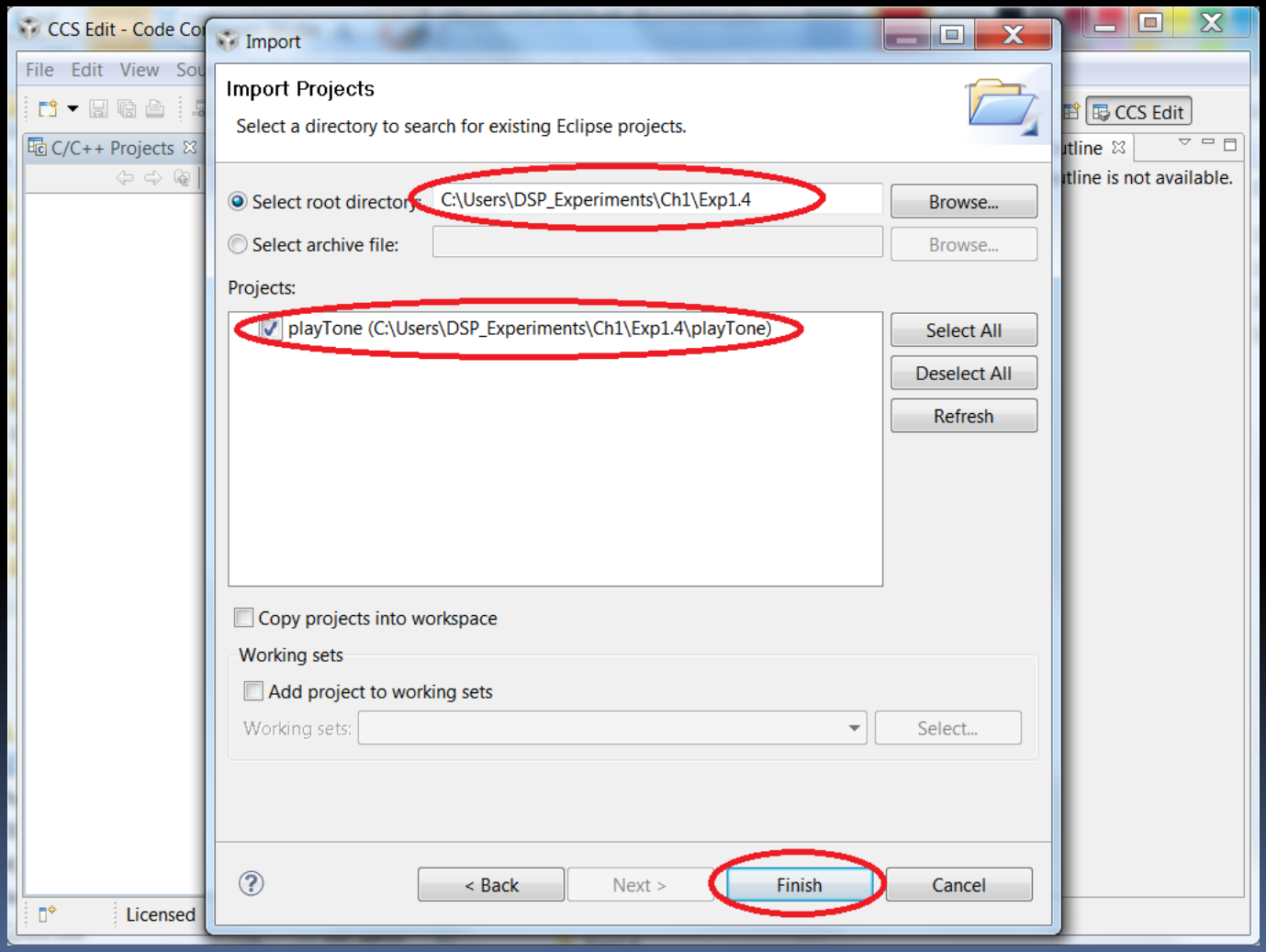
Import existing CCS project (3)

(Browse…, go to your folder, then OK)



Import existing CCS project (4)

(Select the path, the project, then click Finish)



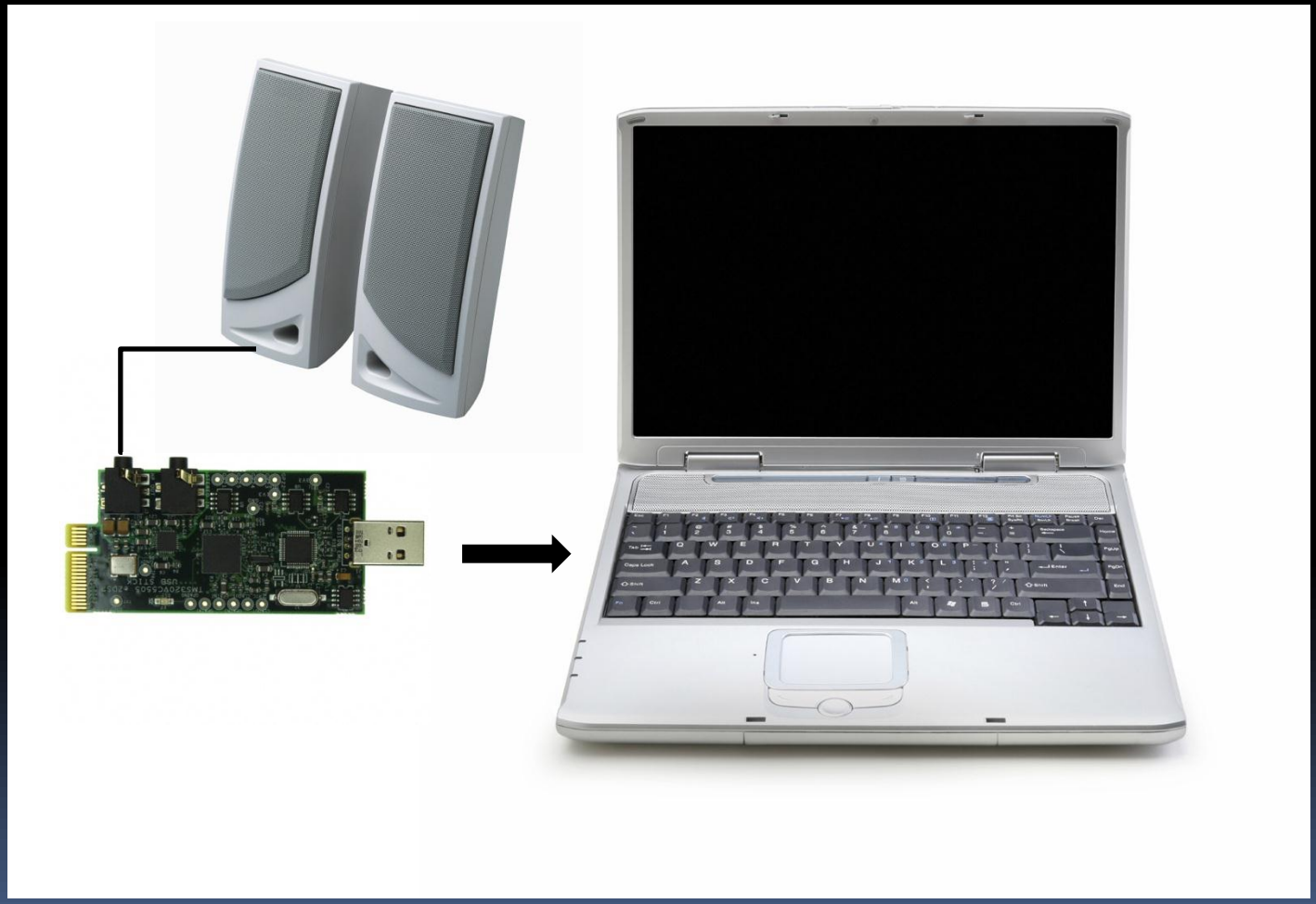
Experiment preparation

- Start CCS
- Import workspace Exp1.4 *playTone*
- Use *Build All* command to rebuild the experiment
- Connect eZdsp to computer
- Connect stereo speaker or headphone to eZdsp's HP Out jack
- From CCS *View->Target Configurations* to open the *Target View* window, locate the *playTone.ccxml*, launch and connect eZdsp
- Load the program *playTone.out* and run the experiment
  - Using different gain values, sampling frequencies, and time durations
  - Once the program stops, go to Run->Restart, the Resume to rerun the experiment

Note this experiment includes several folders

- ***src*** *– source program folder, containing experiment programs*
- ***C55xx_csl*** *–contains all the header files for C55x CSL (chip select library)*
- ***USBSTK_bsl*** *–contains all the header files for the eZdsp BSL (board support library)*

Connect Speaker to eZdsp



The target configuration

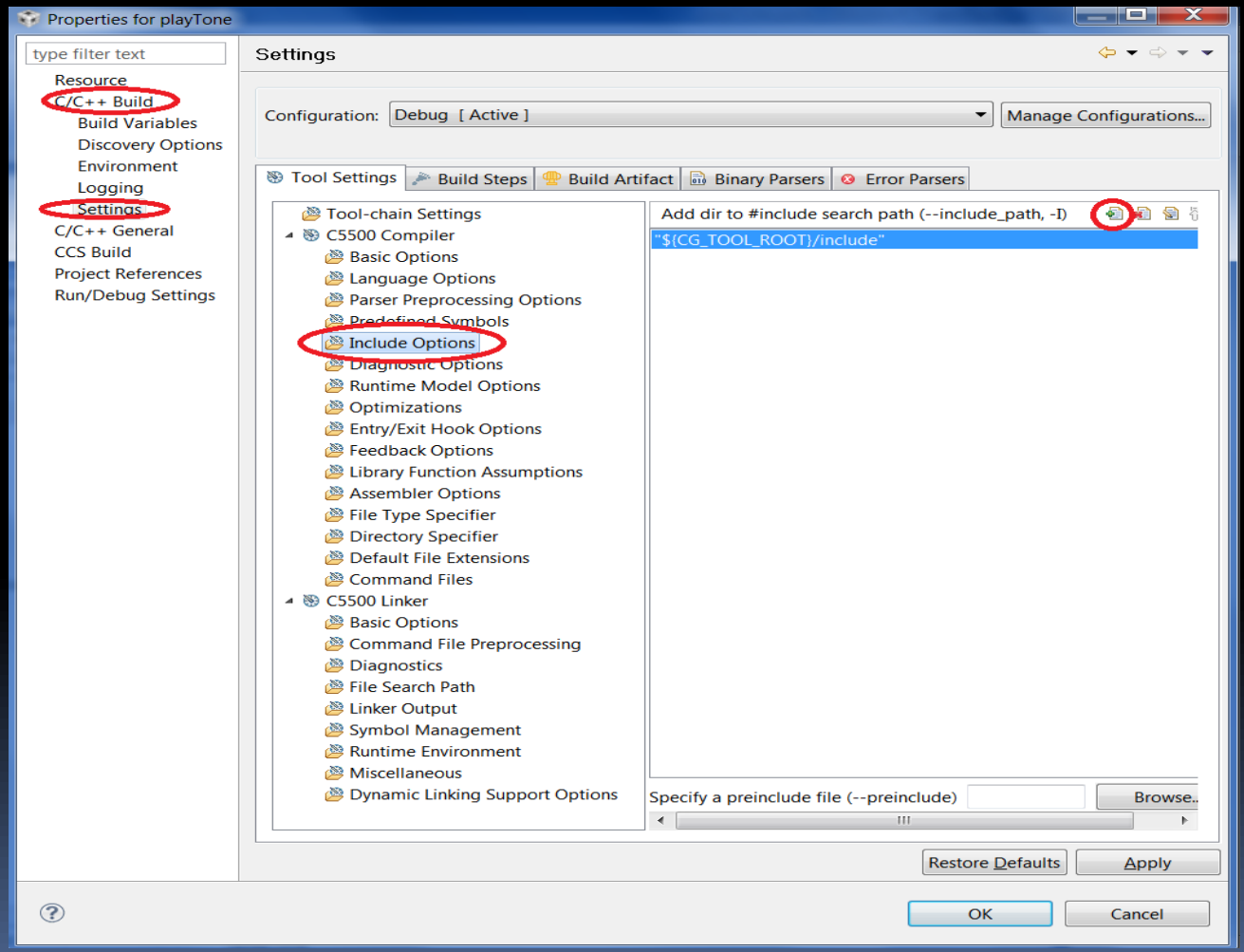
- Target configuration file name
  - *playTone.ccxml*
  - Texas Instruments XDS100v2 USB Emulator
  - Set for use USBSTK5505

Project building environment

- View build environment
  - Right click on project "PlayTone" then select Property
  - Select and expand C/C++ Build option
  - Select Settings, then Runtime Options
- Include path
  - The header files are needed by experiment programs. These header files are in sub-folders of the project folder, *C55xx_csl*, and *USBSTK_bsl*

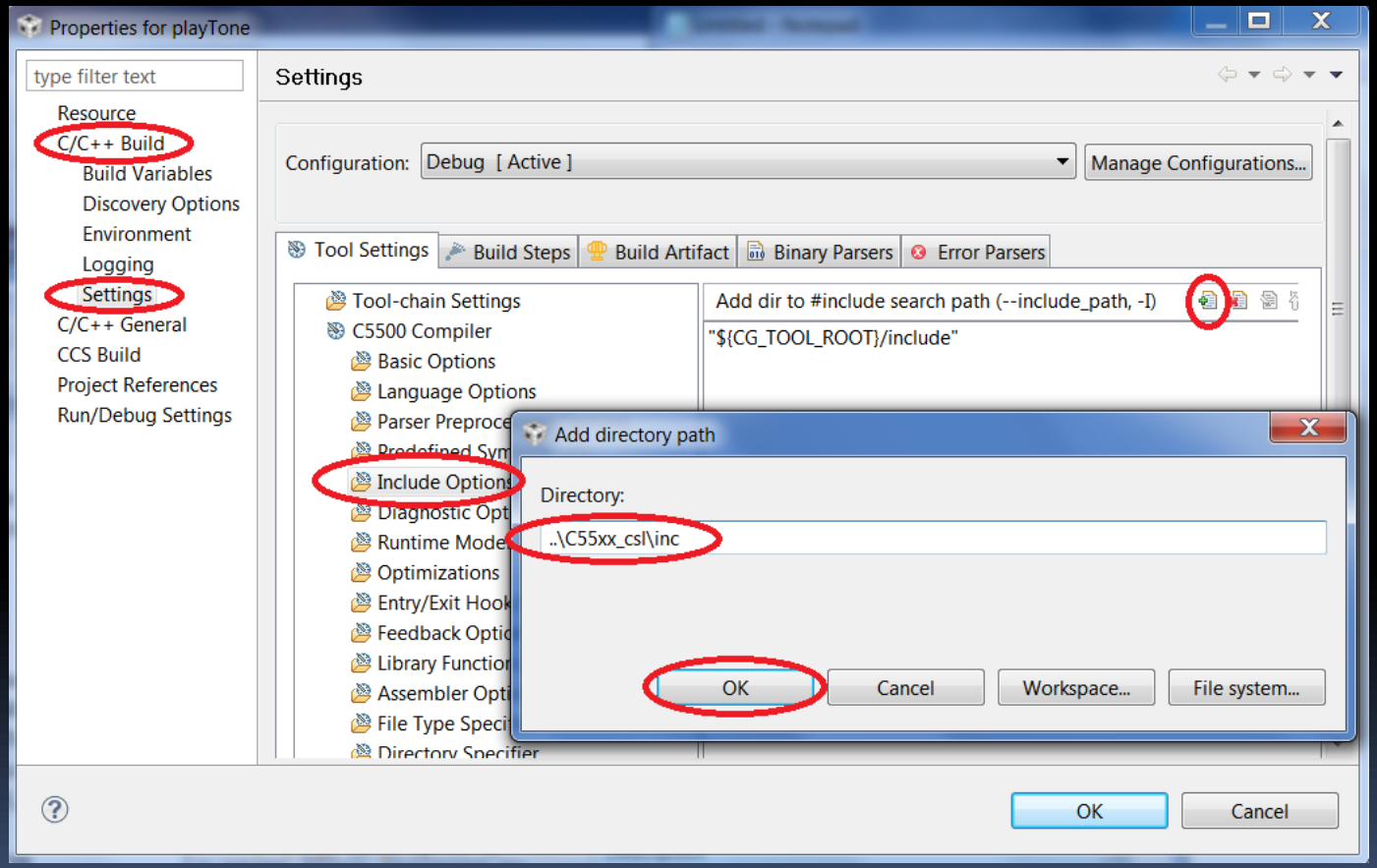
Setup Dependency Search Path

(Example: Property->C/C++ Build->Setting->Include Options)



Add Path for Header Files

(Example: add ..\C55xx_csl\inc and USBSTK_bsl\inc)



Build and run the program

- Build the project (use *Build All* or *Clean*)
- Load the program
- Connect a headphone or PC speaker to the eZdsp HP jack (3.5mm jack at the far end of the eZdsp board)
- Run the program using the following parameters:
  - Gain = 0 dB
  - Sampling Frequency = 48000 Hz
  - Playtime = 5 seconds
- The 1000 Hz tone will be played for 5 seconds
- Rerun the experiment with different settings, gain, sampling frequency, and playtime and verify the result

Set Sampling Frequency

- C5505 eZdsp USB Stick uses a 12MHz crystal.
- The AIC3204 setting will be based on this crystal's frequency, that is MCLK=12000000
- The DAC (AIC output digital-to-analog) Sampling Frequency can be calculated by

  SF = (MCLK*J.D*R)/(P*NDAC*MDAC*DOSR)

  where J, D, R, P, NDAC, MDAC, DOSR are AIC3204 registers
- Below is a table shows the combination of the register settings for different sampling frequency for DAC

SF	J.D	R	MCLK	NDAC	MDAC	P	DOSR
48000 Hz	7.168	1	12000000	2	7	1	128
24000 Hz	7.168	1		2	7	2	128
16000 Hz	7.168	1		2	7	3	128
12000 Hz	7.168	1		2	7	4	128
8000 Hz	7.168	1		2	7	6	128

New experiment assignments

- Write a program that will
  - Configure eZdsp to play the *DTMF123.wav* provided with the experiment software at 8KHz sampling rate with DAC gain at -3 dB
  - Rub the eZdsp to play back the audio and compare it with the original *DTMF123.wav* playing back from a computer.

    Q1: Do you hear the same result? If not, find the problem and correct it. (hint: the .wav file has a header)

Programming quick review

- In order to make modular designs, in this experiment, we have written 3 files, *tone.c*, *playTone.c*, and *initAIC3204.c*. Each C file contains one or more functions
- Using a modular programming methodology, one can write a function for different uses. This is a good practice for software reuse, that is the functions that have been written and tested can be reused again and again.
- *tone.c* – this file contains the main function of the experiment. It uses the UI interface program we developed in previous experiment to provide gain, sampling frequency, and tone play back duration
- *InitAIC3204.c* – this file has the program that is used to initialize the control registers of the audio interface IC, AIC3204.
- *playTone.c* – this program calls *Init_AIC3204( )* to set up AIC3204 and play a tone for the time duration given by the user

References

- Ultra Low Power Stereo Audio Codec, by Texas Instrument, SLOS602A – OCT., 2008