Cortex-R4 Build Setup

ARM Confidential

TABLE OF CONTENTS

1. Introduction 4
2. Pre-processor Macros 4
2.1. ARM_MATH_BIG_ENDIAN 4
2.2. ARM_MATH_MATRIX_CHECK 4
2.3. FPU_PRESENT 4
2.4. ARM_MATH_ROUNDING 4
2.5. UNALIGNED_SUPPORT_DISABLE 4
2.6. CCS 4
3. DSP Library Benchmarking on Cortex-R using MDK 5
3.1. Tools & Hardware used for Benchmarking 5
3.1.1. Cortex-R 5
3.2. Build the Library Project 5
3.3. Run the test Project 7
3.4. Cycle Measurement 8
4. Build steps on Cortex-R using CCS 12
4.1. Tools & Hardware used 12
4.1.1. Cortex-R 12
4.2. Open and Build the Library Project 12
5. Build steps on Cortex-R using DS-5 17
5.1. Tools used 17
5.1.1. Cortex-R 17
5.2. Open and Build the Library Project 17
6. Build steps on Cortex-R using GCC - CodeSourcery 22
6.1. Tools used 22

6.1.1. Cortex-R 22
6.2. Open and Build the Library Project 22
7. Build steps on Cortex-R using RVDS 27
7.1. Tools used 27
7.1.1. Cortex-R 27
7.2. Open and Build the Library Project 27
8. Build steps on Cortex-R using GCC - MDK 32
8.1. Tools used 32
8.1.1. Cortex-R 32
8.2. Open and Build the Library Project 32

1. Introduction

This document discusses how to build the Cortex-R DSP software library on different Tool chains like MDK, Code Composer studio, DS-5, MDK using the GCC compiler, GCC CodeSourcery and RVDS

2. Pre-processor Macros

Below are the different pre processor MACROS needs to define n the project options while building the library on different Tool chains. Define only the MACROS which ever applicable to supported features of hardware.

2.1. ARM_MATH_BIG_ENDIAN

Define macro ARM_MATH_BIG_ENDIAN to build the library for big-endian targets. By default library builds for little-endian targets.

2.2. ARM_MATH_MATRIX_CHECK

Define macro for checking on the input and output sizes of matrices

2.3. FPU_PRESENT

Define macro FPU_PRESENT when building on FPU supported Targets

2.4. ARM_MATH_ROUNDING

Define macro for rounding on support functions

2.5. UNALIGNED_SUPPORT_DISABLE

Define macro UNALIGNED_SUPPORT_DISABLE, if the silicon does not support unaligned memory access

2.6. CCS

Define macro if the code composer studio tool is used.

3. DSP Library Benchmarking on Cortex-R using MDK

3.1. Tools & Hardware used for Benchmarking

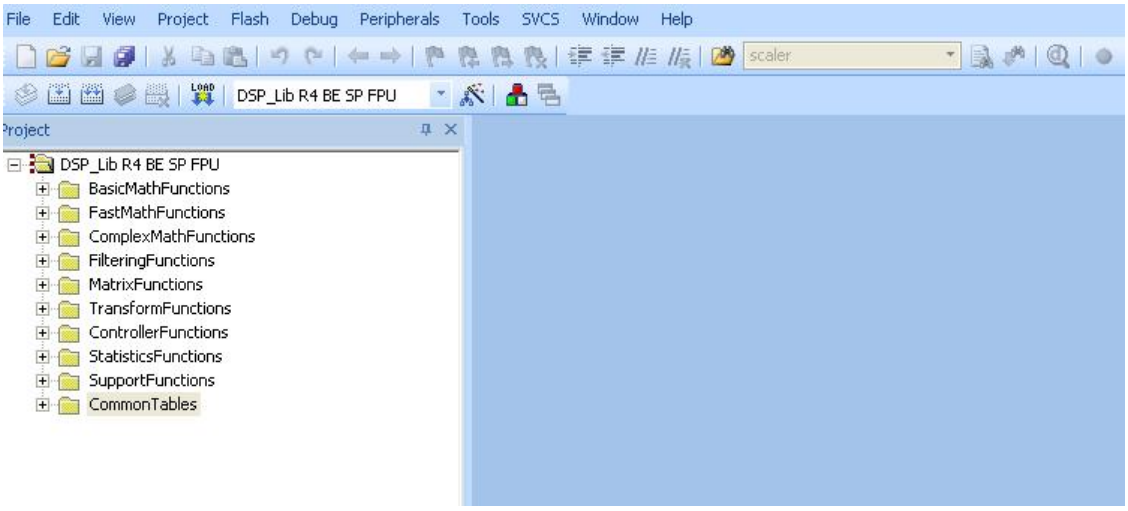
3.1.1. Cortex-R

Tool: MDK version 4.21
ARMCC Compiler version 4.10.561
Hardware: TMS570LS20216

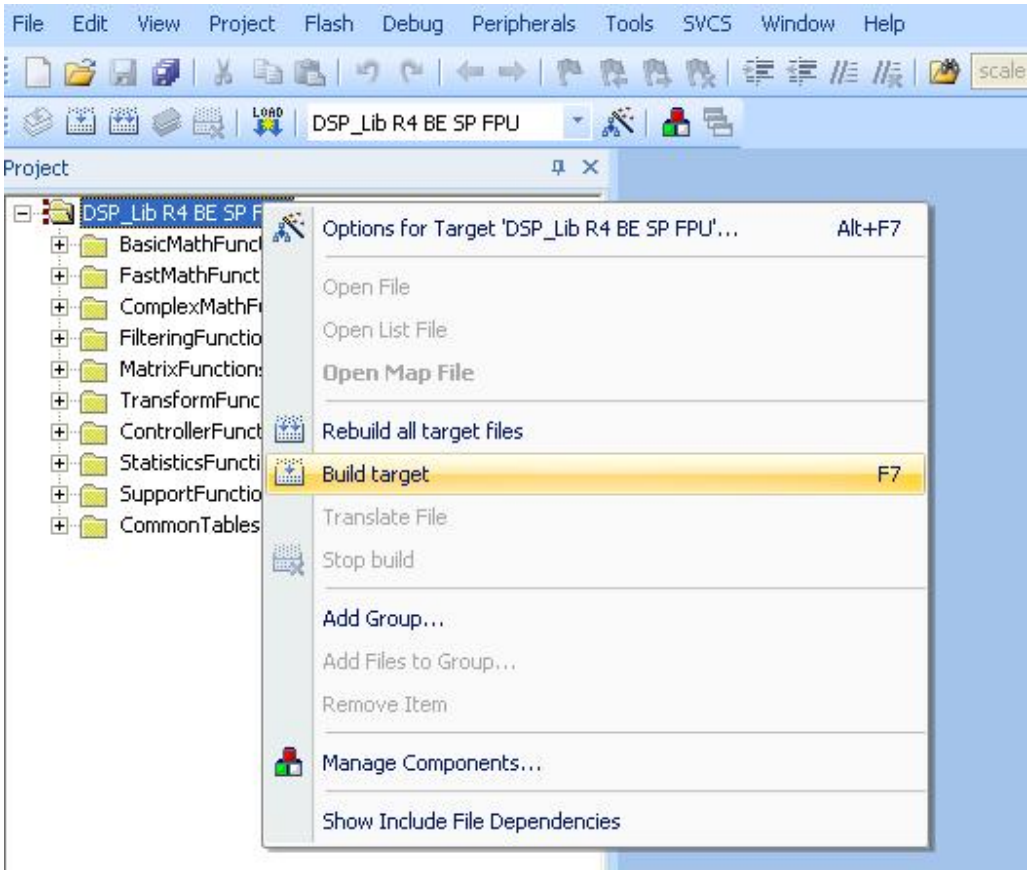
3.2. Build the Library Project

Double click on the arm_math_Cortex_R4_bspf.uvproj library project to open in the Keil.

The main window will appear as shown below.



To build the library project, Right click on DSP_Lib R4 BE SP FPU in the project window and select Build target (F7) as shown below.



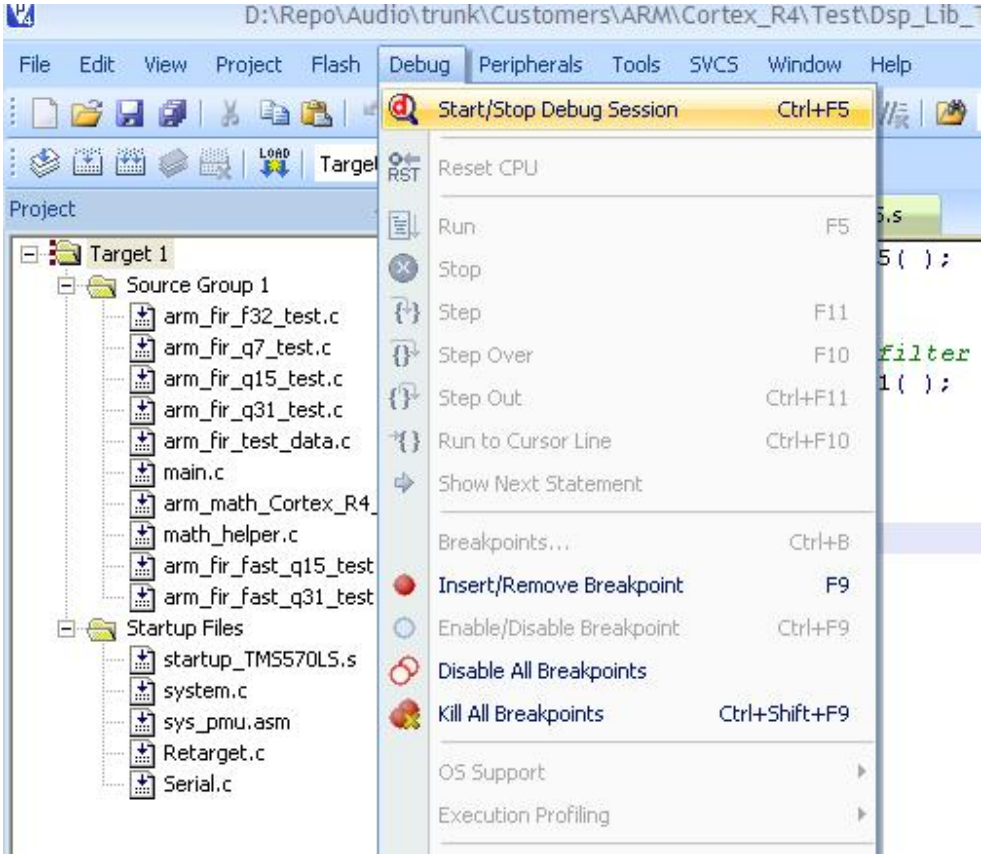
3.3. Run the test Project

Doubleclick on \Test\Dsp_Lib_Test\arm_fir_test \ARM\MDK\arm_fir_test.uvproj to open the FIR test project.

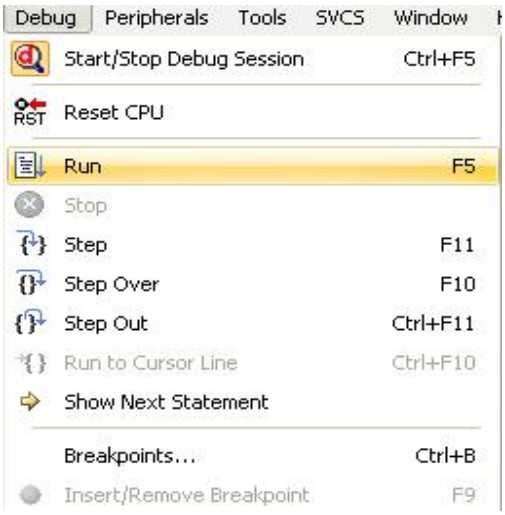
Build the test project (F7).

Start the debug session by selecting

Debug->Start/Stop Debug Session (Ctrl+F5) as shown below.



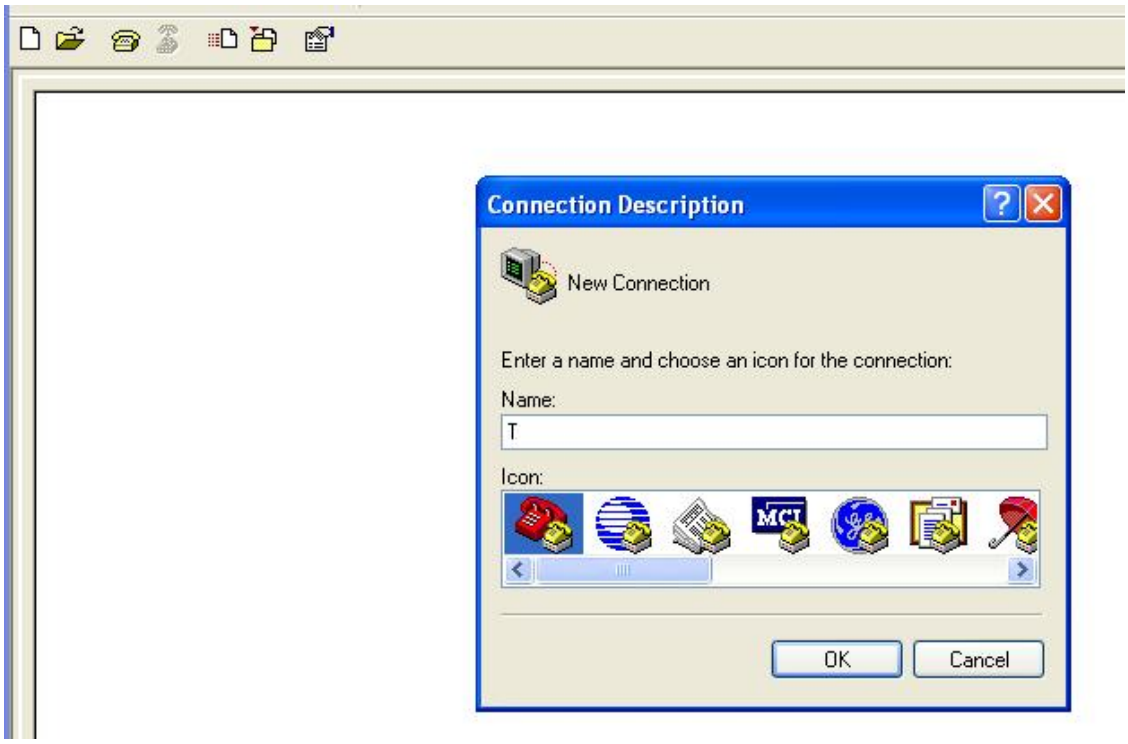
Run the project by selecting Debug > Run (F5).



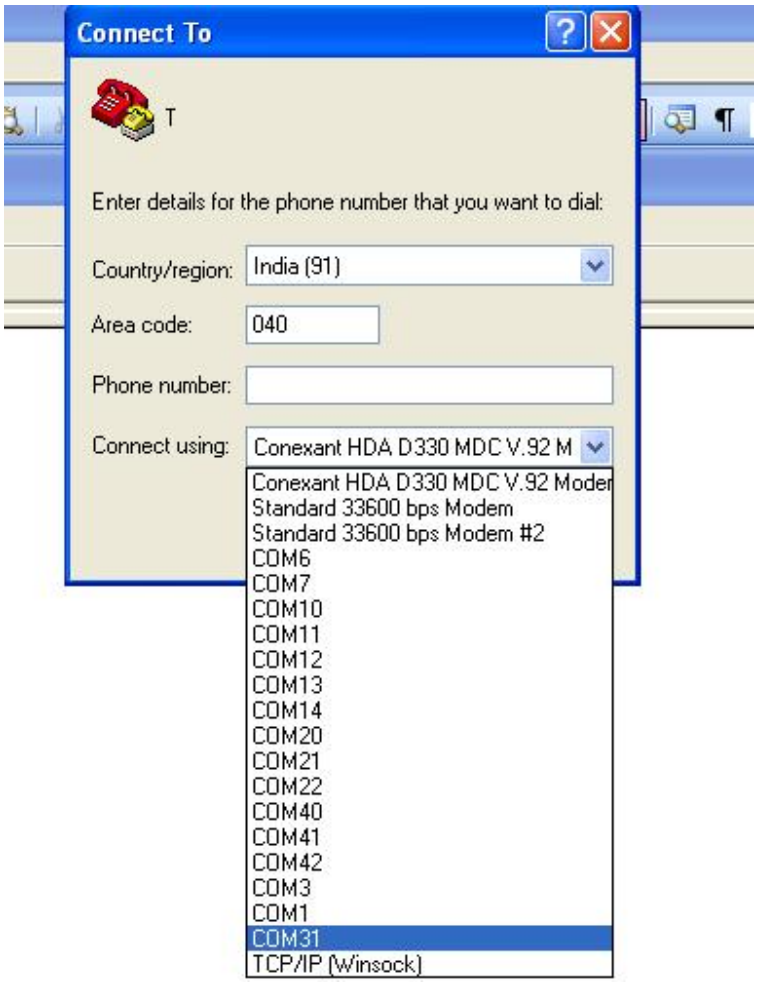
3.4. Cycle Measurement

Cycles are printed on Hyperterminal through UART.

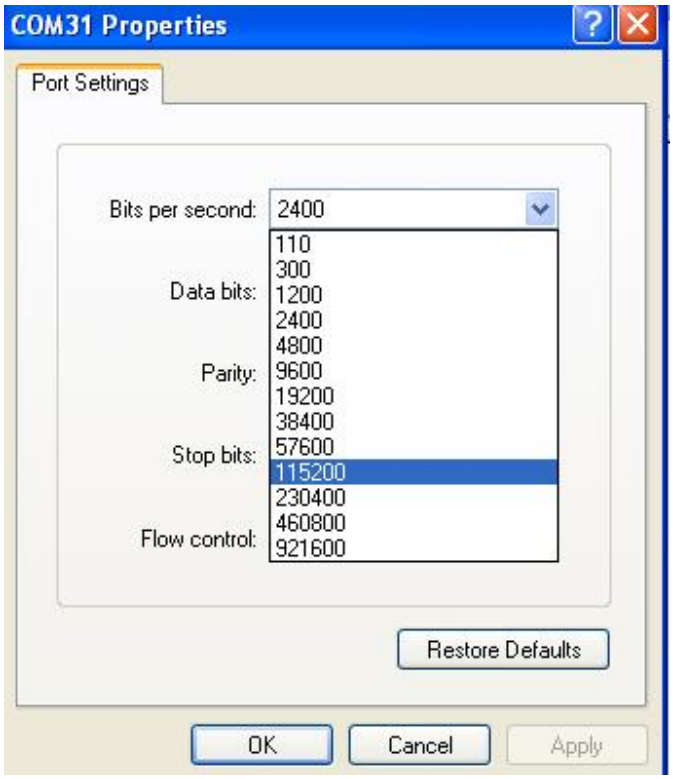
Open the hyperterminal and select name of the connection and click “ok”



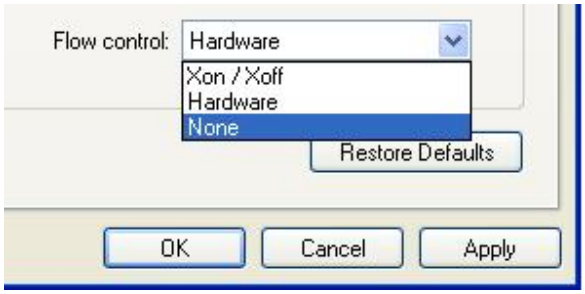
Select appropriate COM port as shown below and click “ok”



Select Bits per second as “115200”



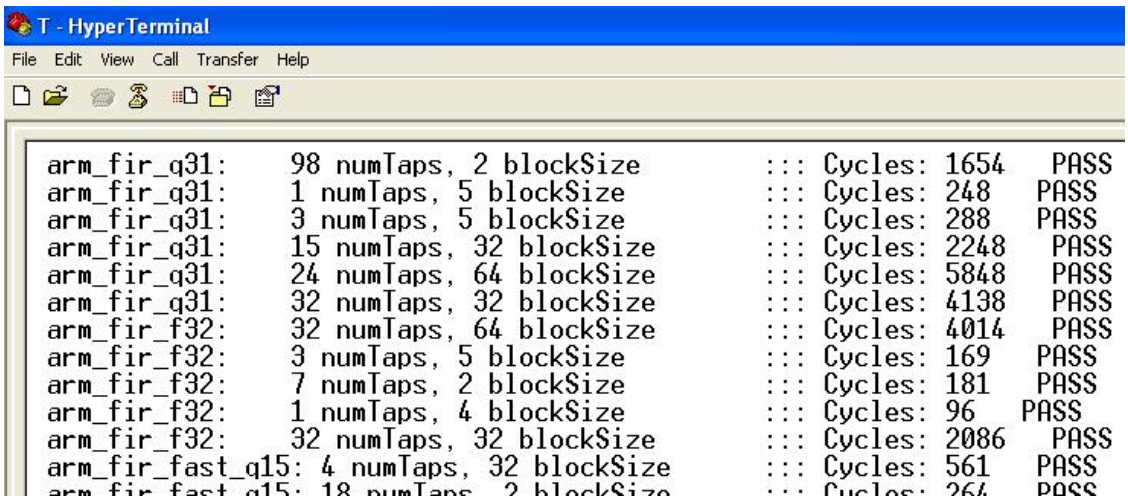
and Flow Control as “None”



Build the test project (F7).

Run the project(F5)

Cycles are printed as shown below window, where each column tells the function name, Configuration for the test, cycles, status of test(Pass/Fail) respectively



4. Build steps on Cortex-R using CCS

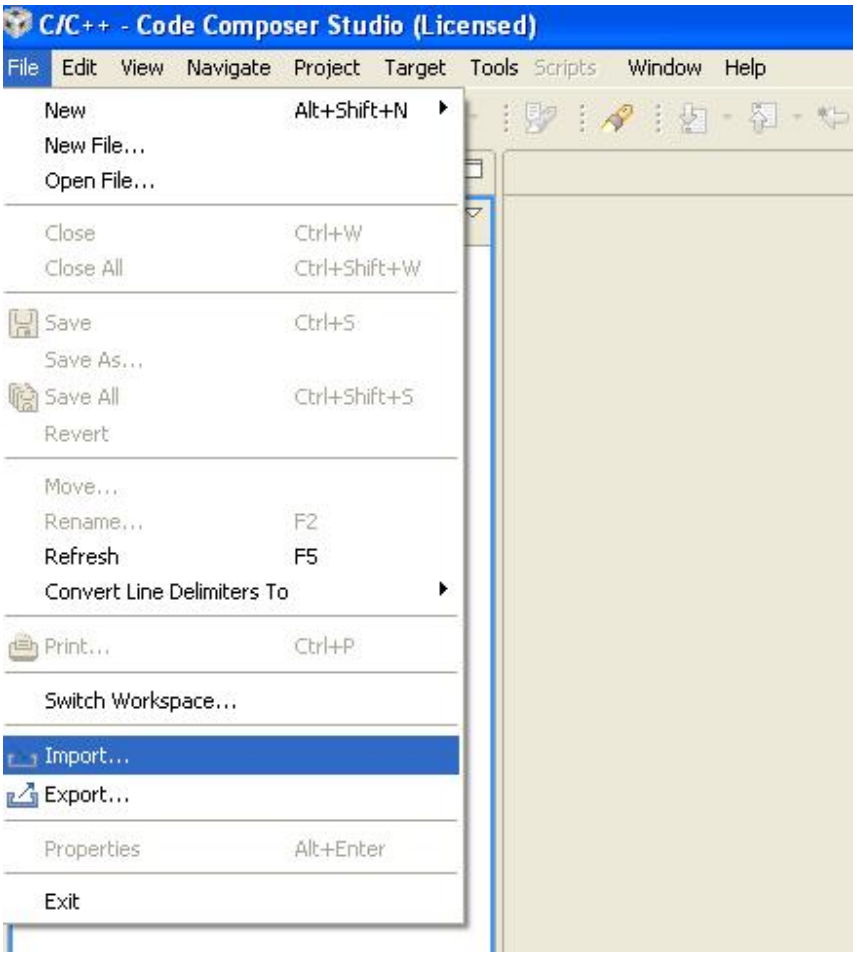
4.1. Tools & Hardware used

4.1.1. Cortex-R

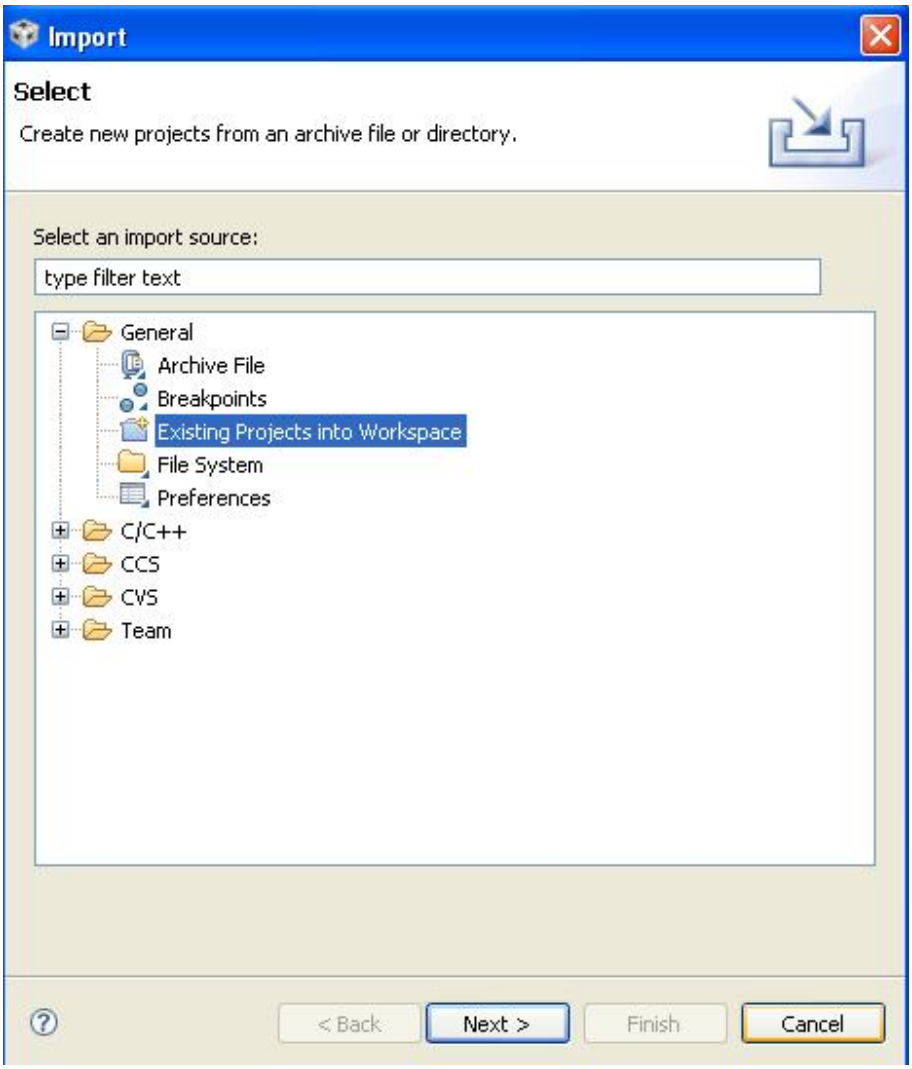
Tool: CCS version 4.2.3.00004
Hardware: TMS570LS20216

4.2. Open and Build the Library Project

The main window will appear as shown below. Select the option File->import

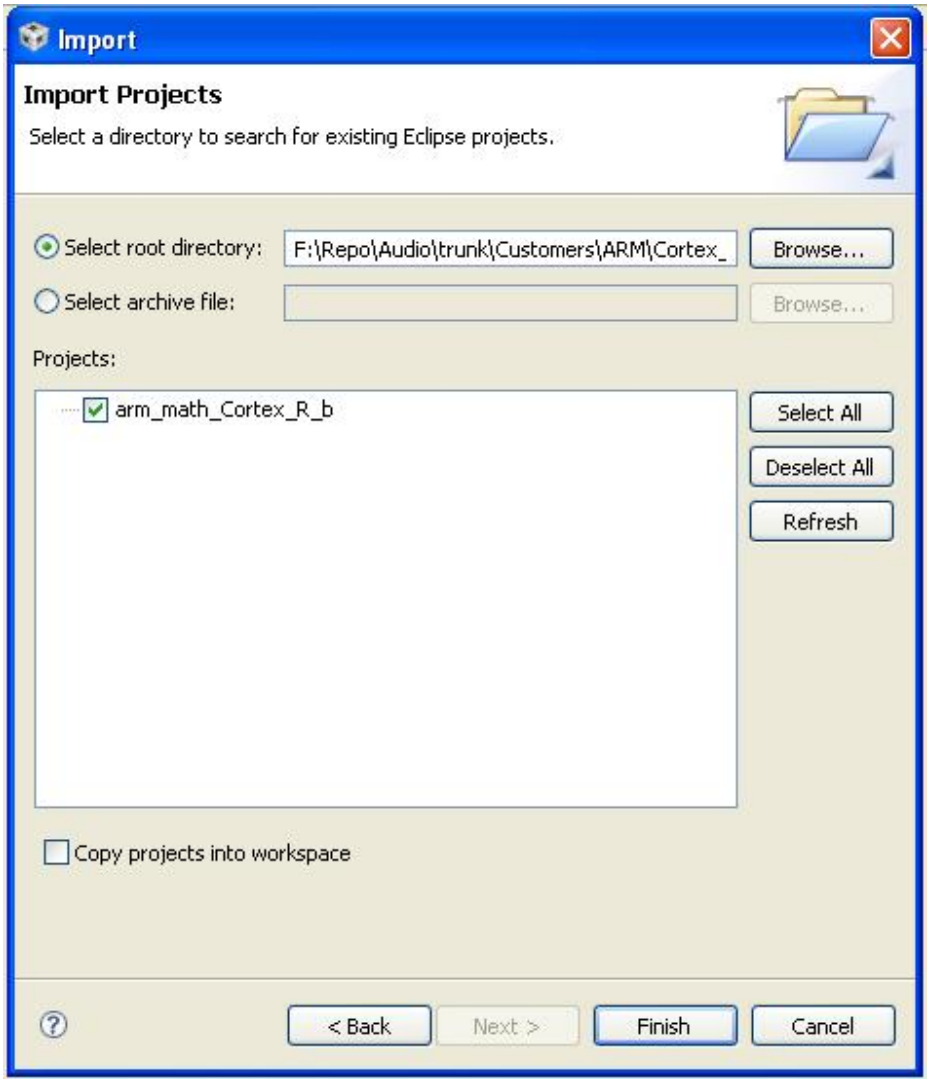


Select the General->Existing Projects into Workspace and then click “next”

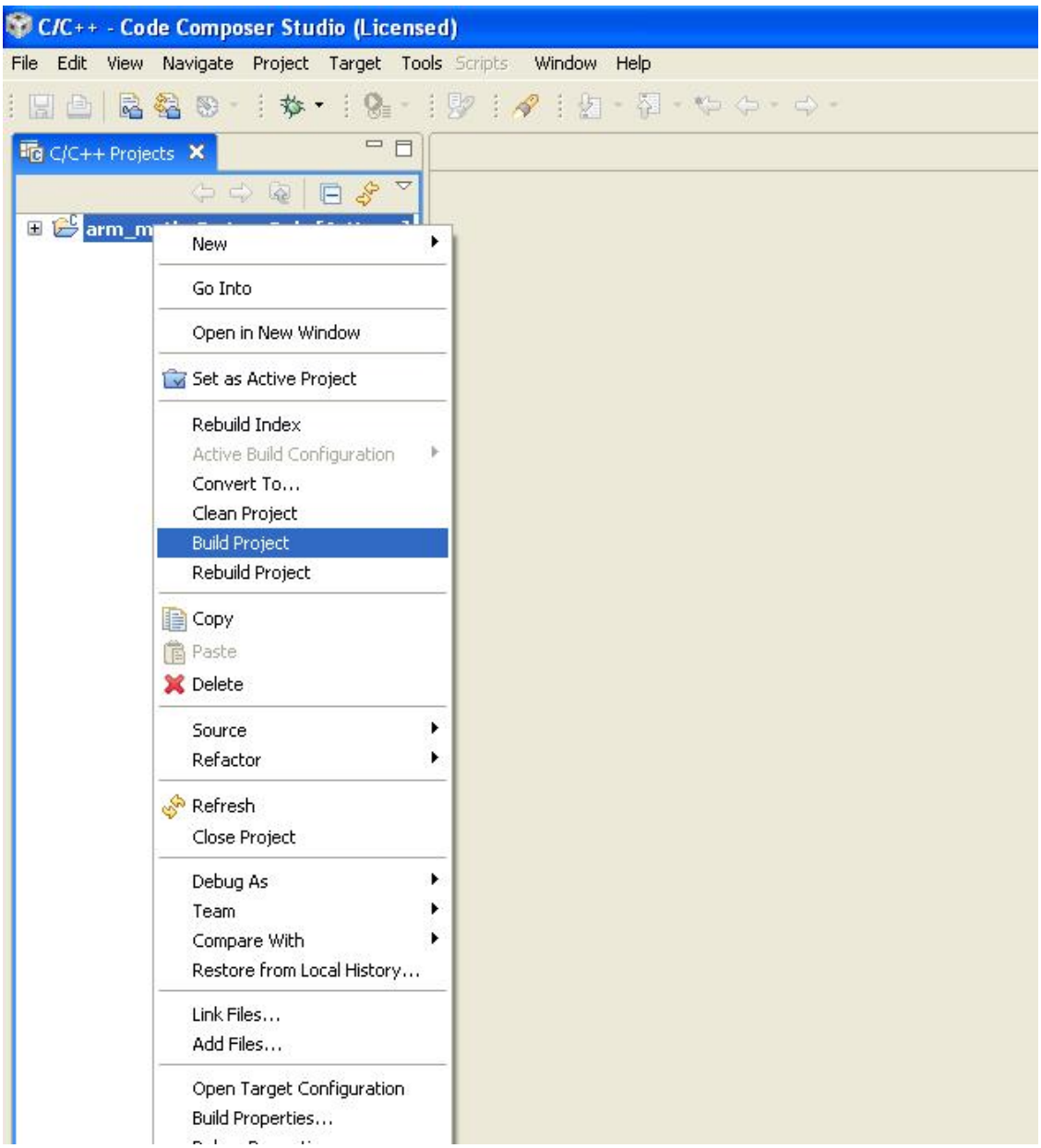


Select Browse and then go to Folder \Lib\TI\CCS\arm_math_Cortex_R_b and then click "ok".

Select project and click finish the Project will be opened into current workspace.



To build the project right click on project and select Build Project option



5. Build steps on Cortex-R using DS-5

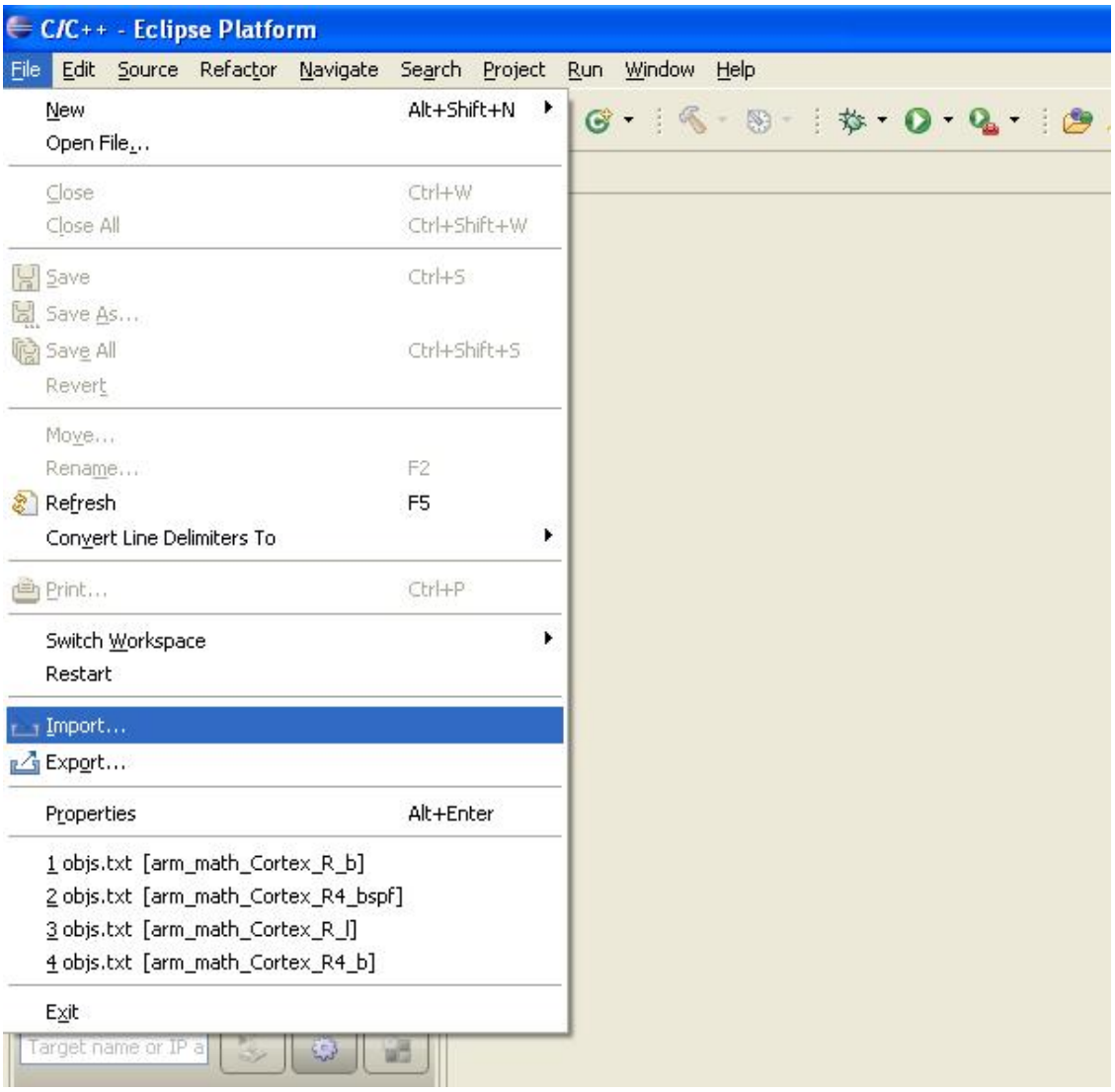
5.1. Tools used

5.1.1. Cortex-R

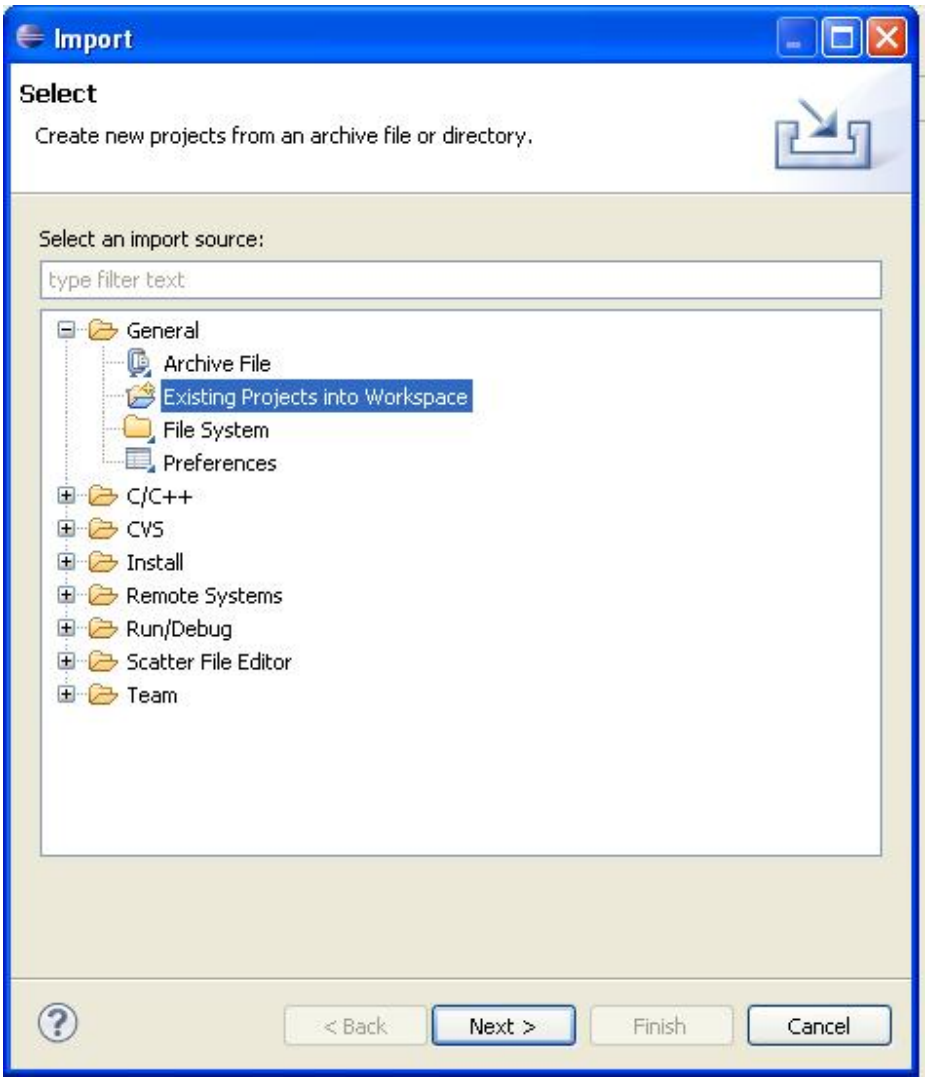
Tool: Eclipse Platform Version: 3.7.0 Build id: I20110613-1736

5.2. Open and Build the Library Project

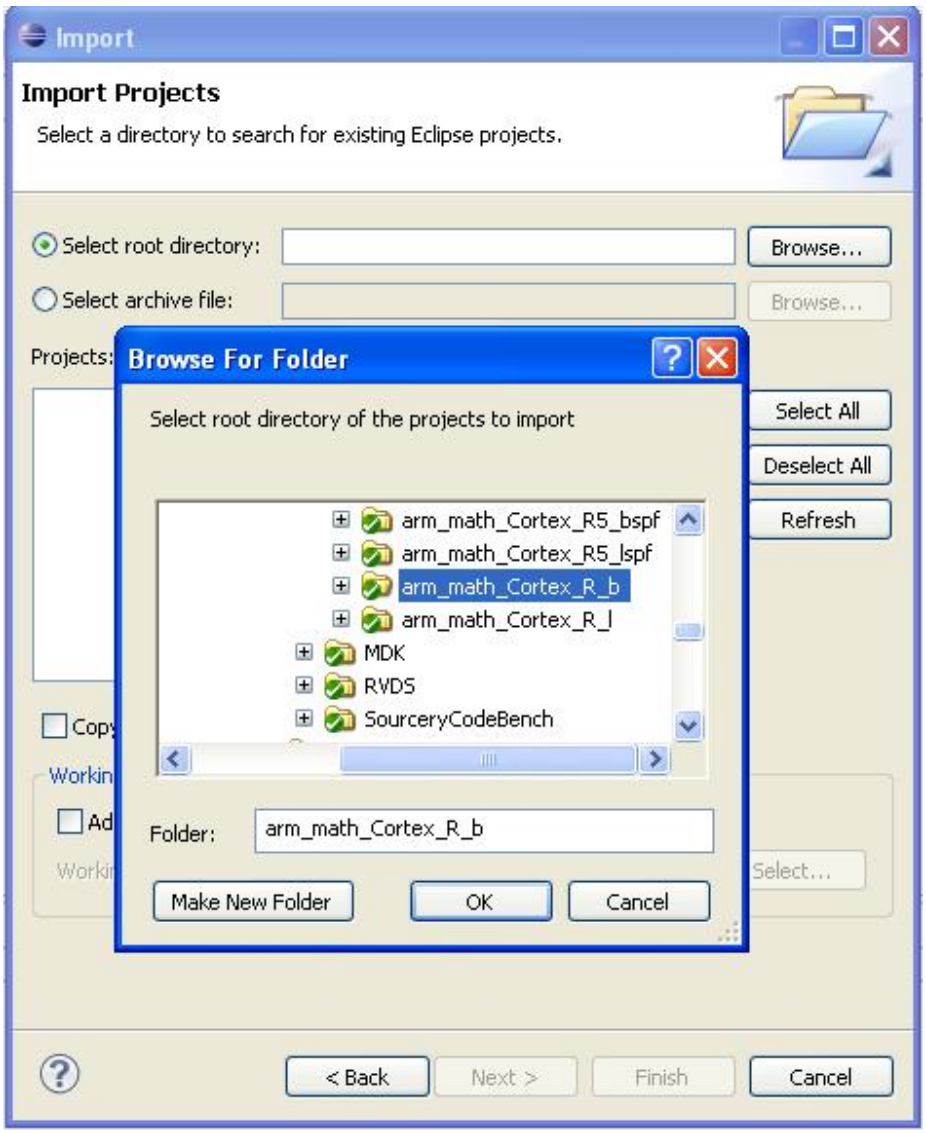
The main window will appear as shown below. Select the option File->import



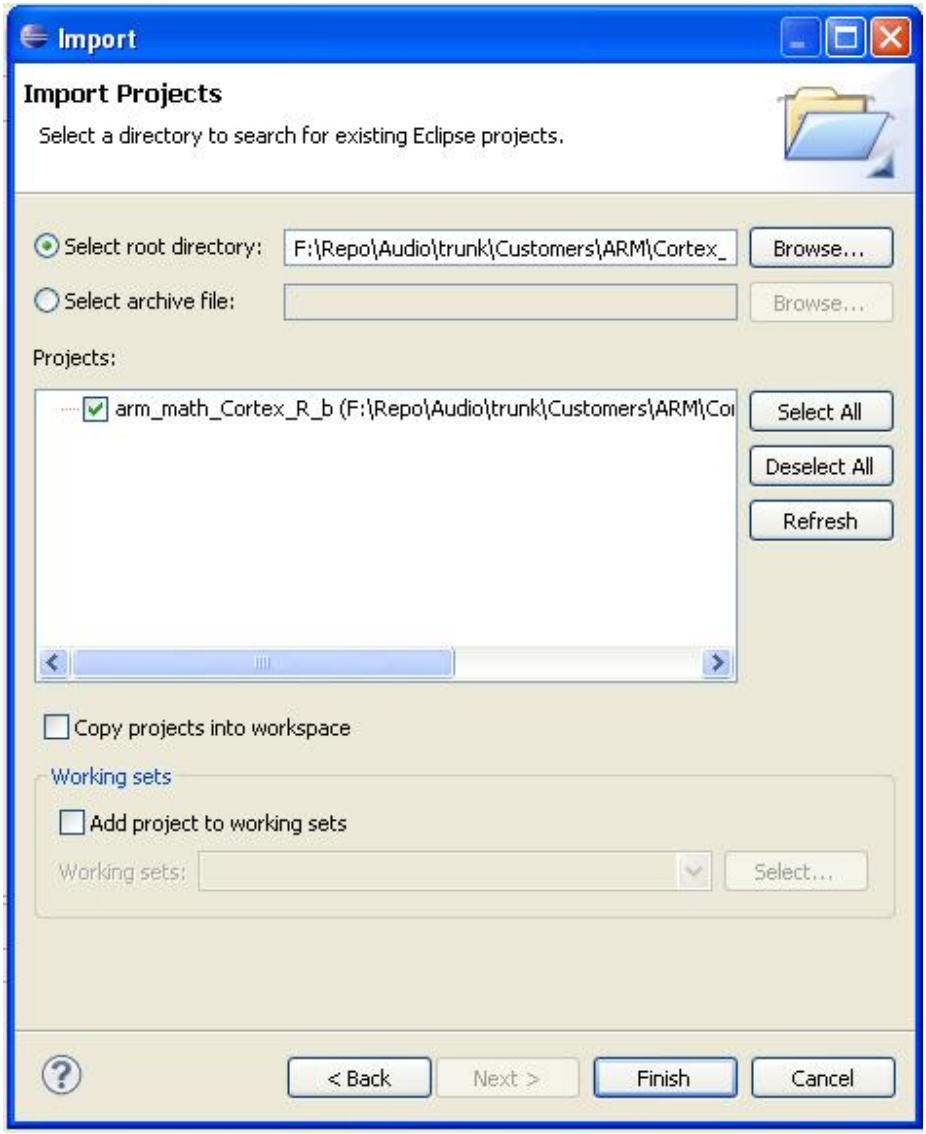
Select the General->Existing Projects into Workspace and then click “next”



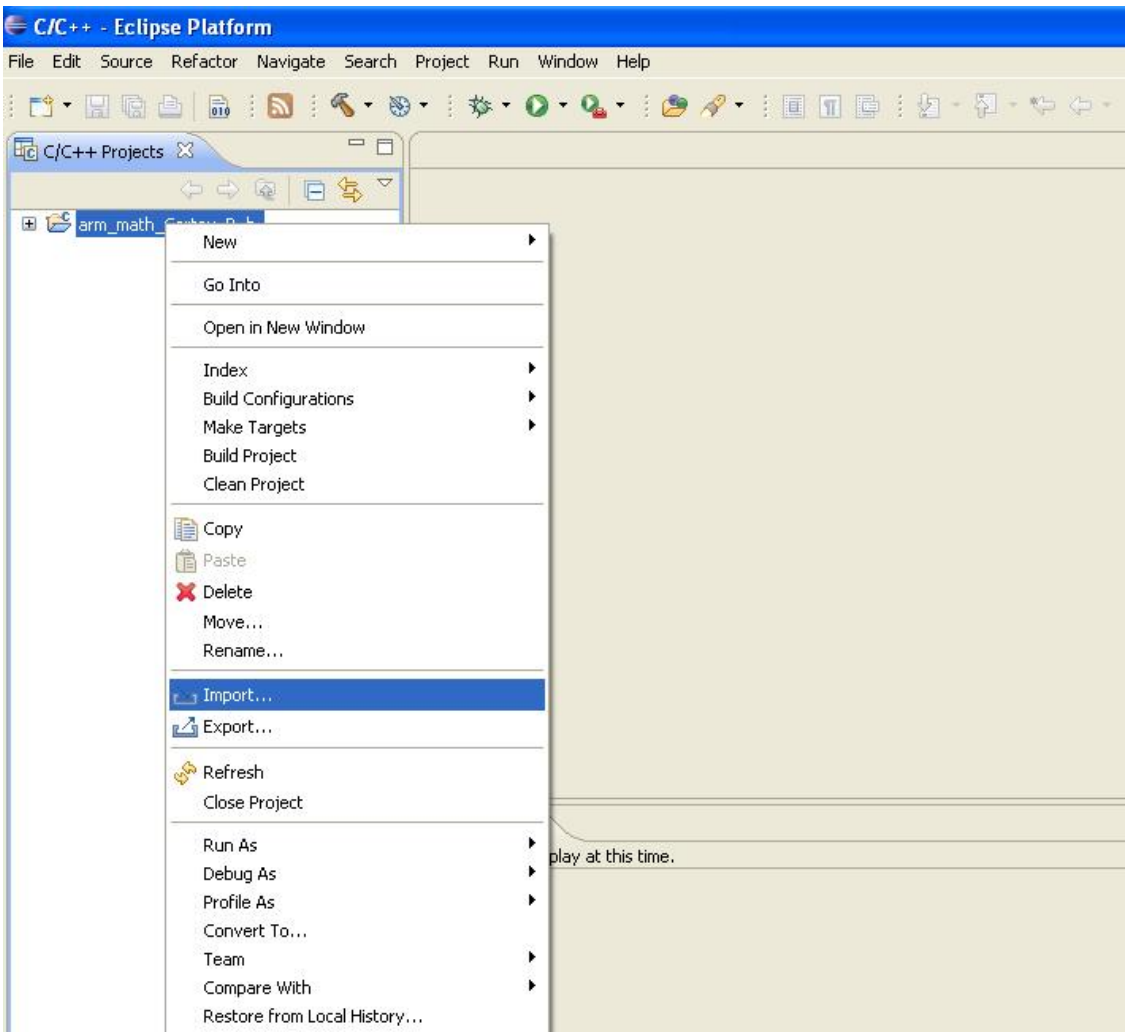
Select Browse and then go to Folder \Lib\ARM\DS5\arm_math_Cortex_R_b and then click “ok”.



Select library and Click finish the Project will be opened into current workspace.



To build the project right click on project and select Build Project option.



6. Build steps on Cortex-R using GCC - CodeSourcery

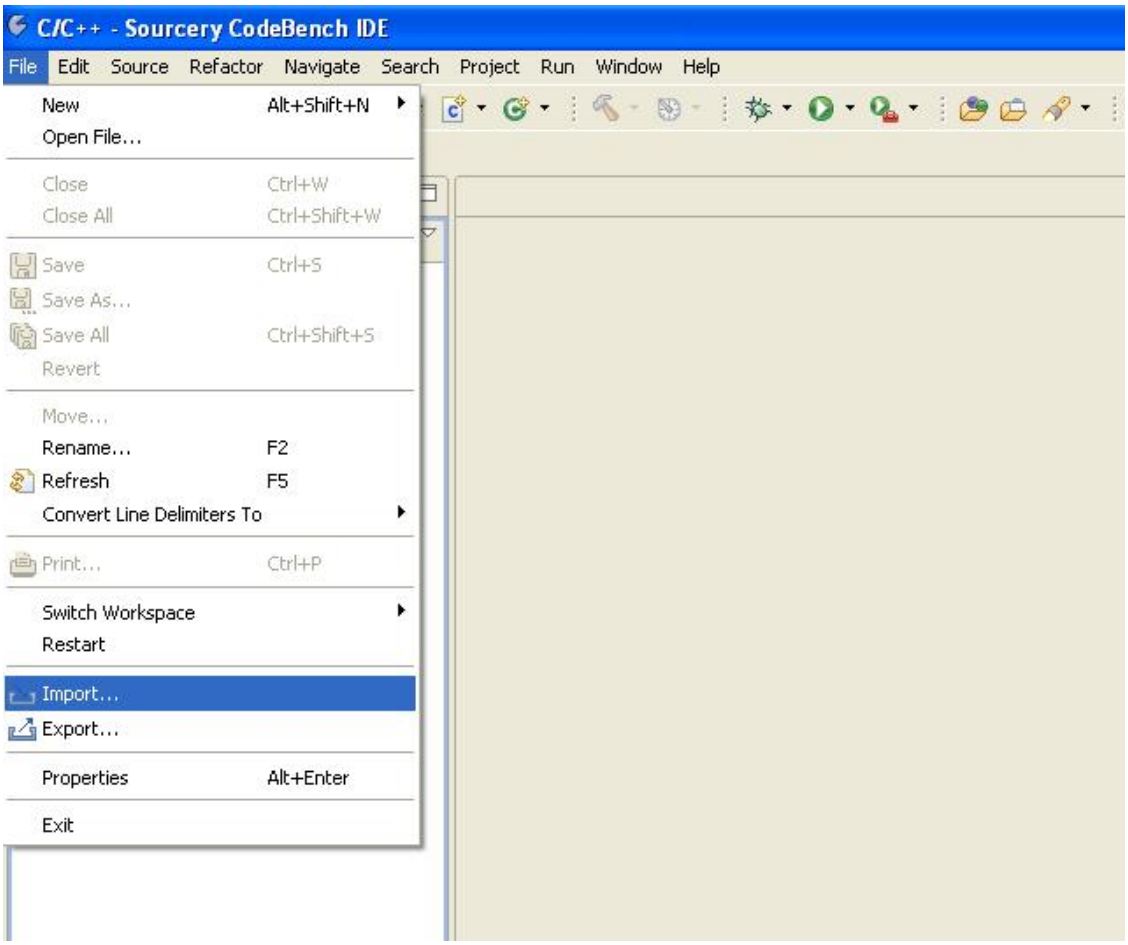
6.1. Tools used

6.1.1. Cortex-R

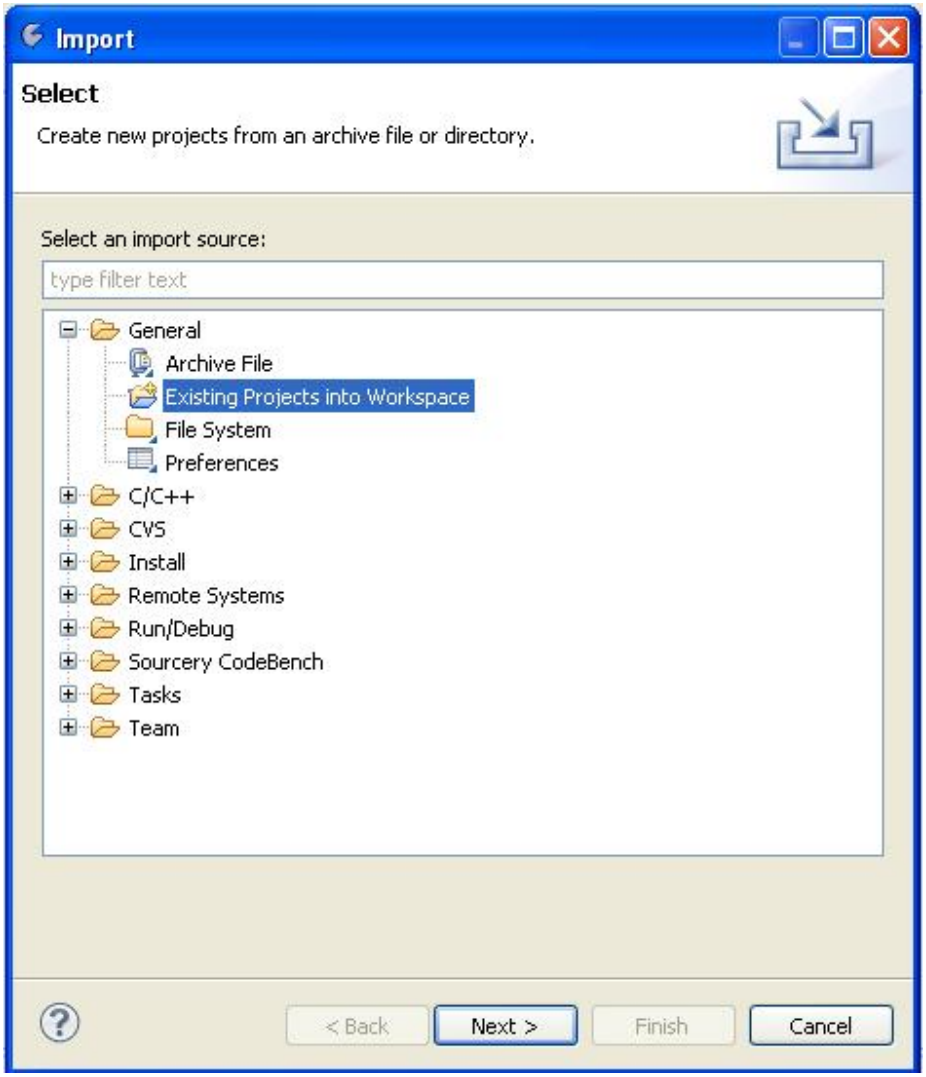
Tool: Sourcery CodeBench IDE version: 2011.90-60

6.2. Open and Build the Library Project

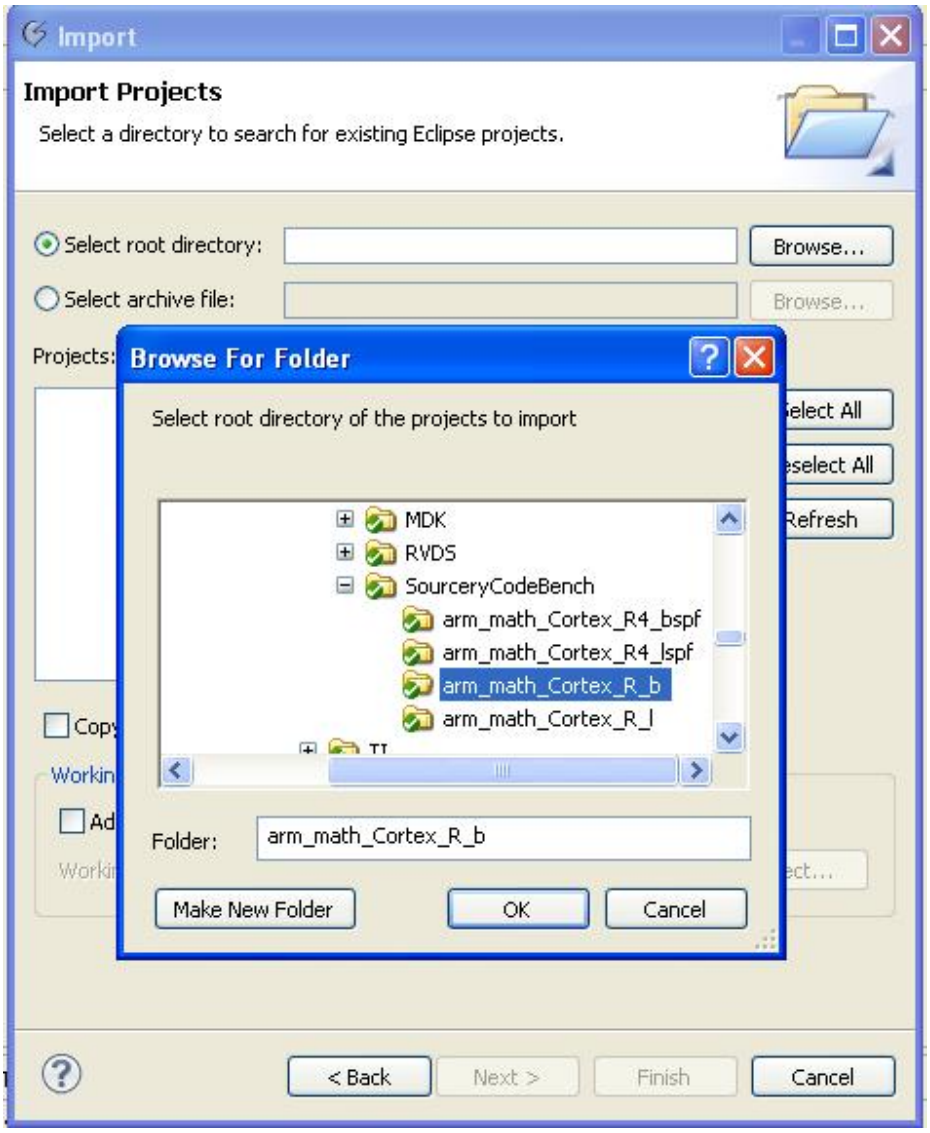
The main window will appear as shown below. Select the option File->import



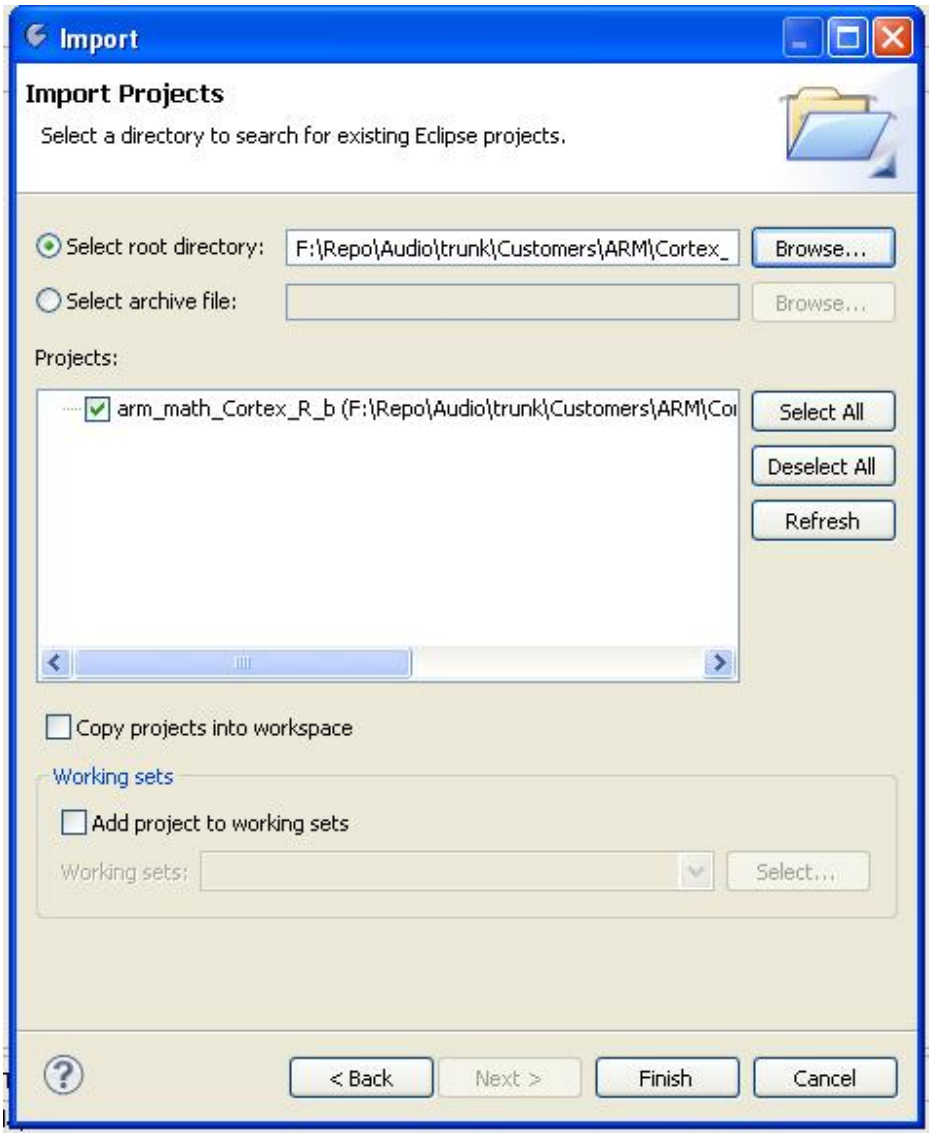
Select the General->Existing Projects into Workspace and then click “next”



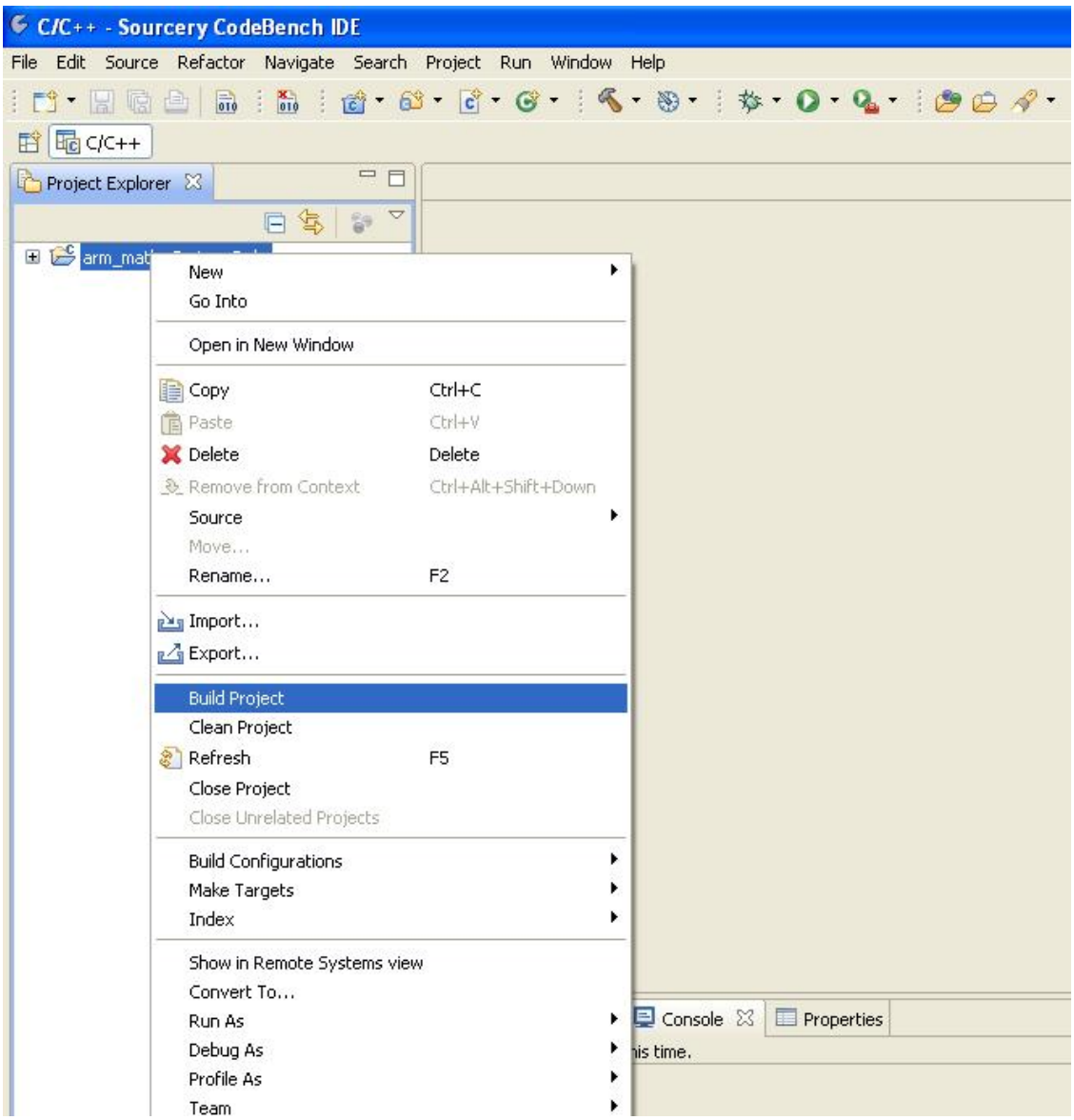
Select Browse and then go to Folder \Lib\ARM\SourceryCodeBench\arm_math_Cortex_R_b and then click “ok”.



Select library and Click finish the Project will be opened into current workspace.



To build the project right click on project and select Build Project option.



7. Build steps on Cortex-R using RVDS

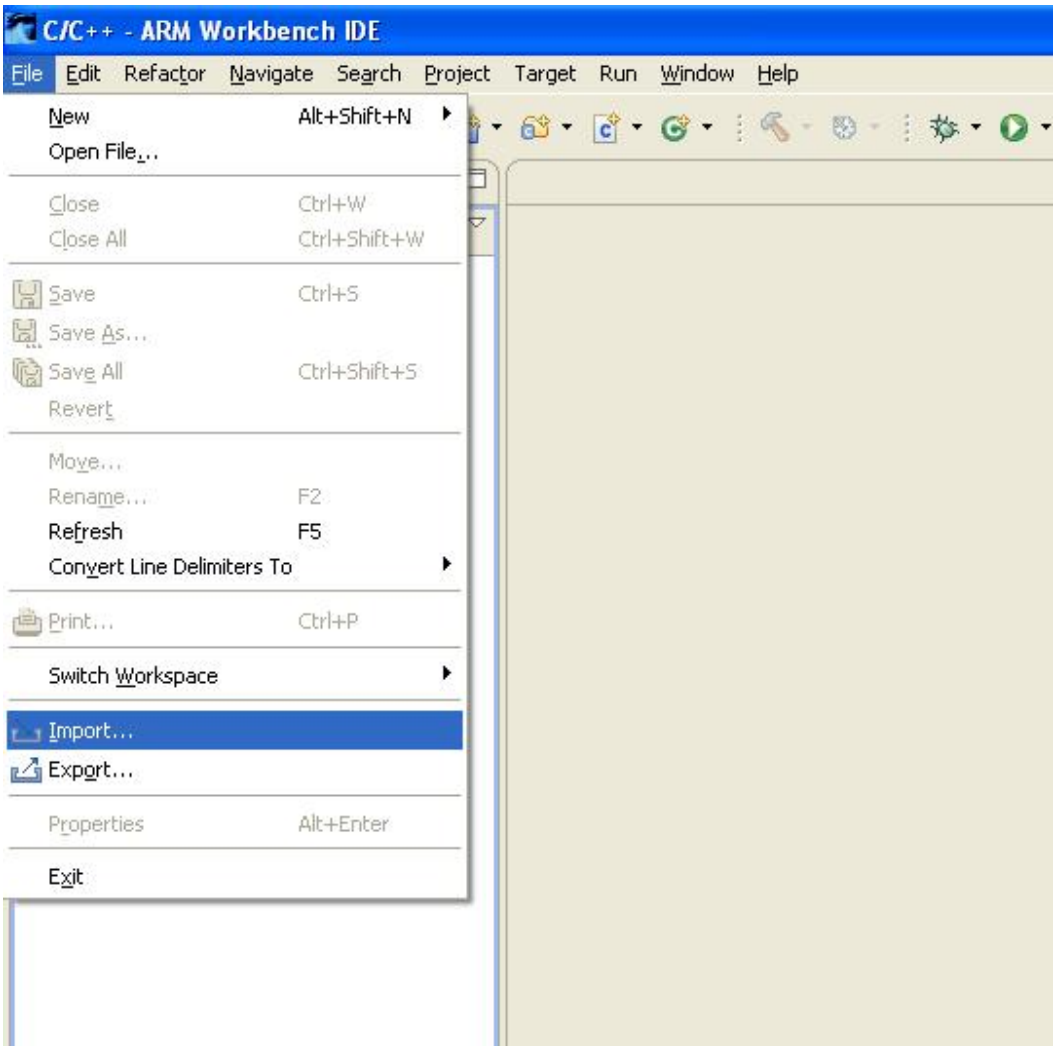
7.1. Tools used

7.1.1. Cortex-R

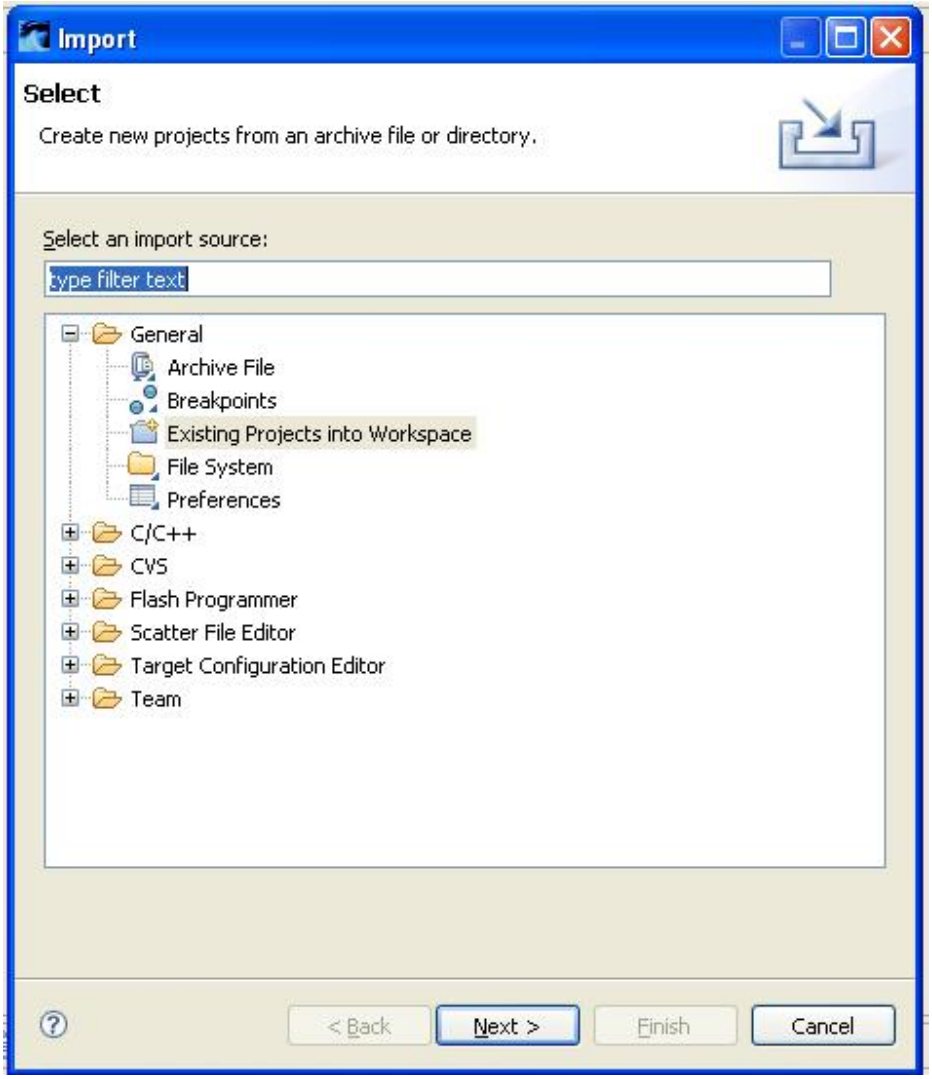
Tool: ARM Workbench IDE version: 4.0 Build Id: 159

7.2. Open and Build the Library Project

The main window will appear as shown below. Select the option File->import

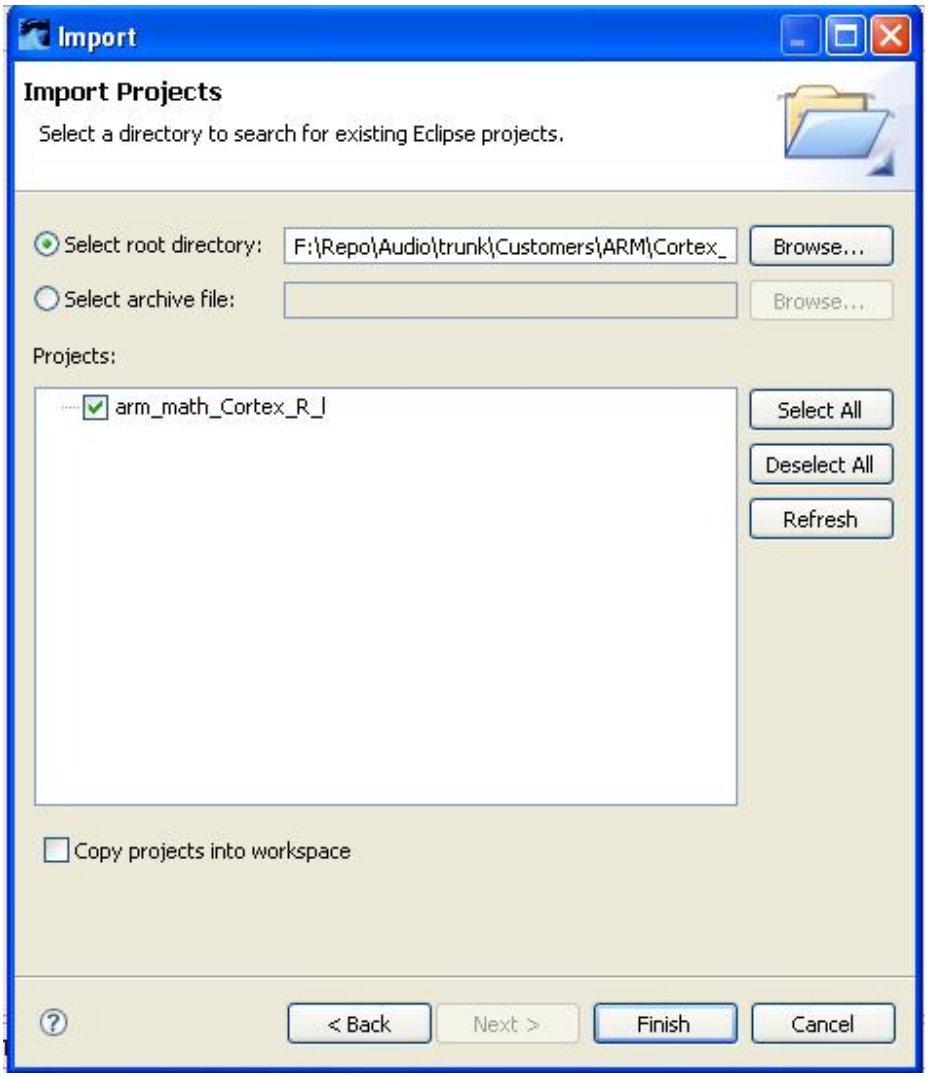


Select the General->Existing Projects into Workspace and then click “next”

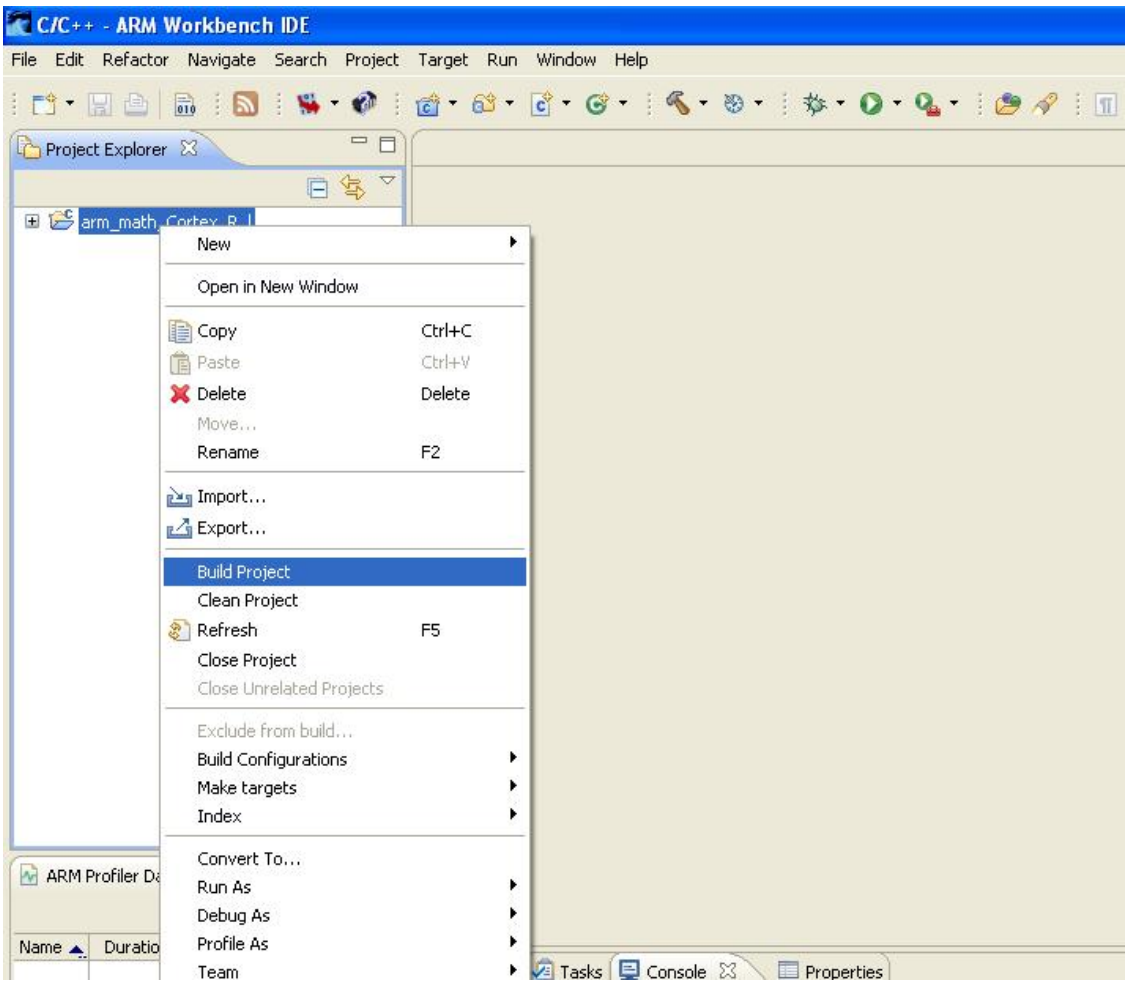


Select Browse and then go to Folder \Lib\ARM\RVDS\arm_math_Cortex_R_l and then click “ok”.

Select library and Click finish the Project will be opened into current workspace.



To build the project right click on project and select Build Project option.



8. Build steps on Cortex-R using GCC - MDK

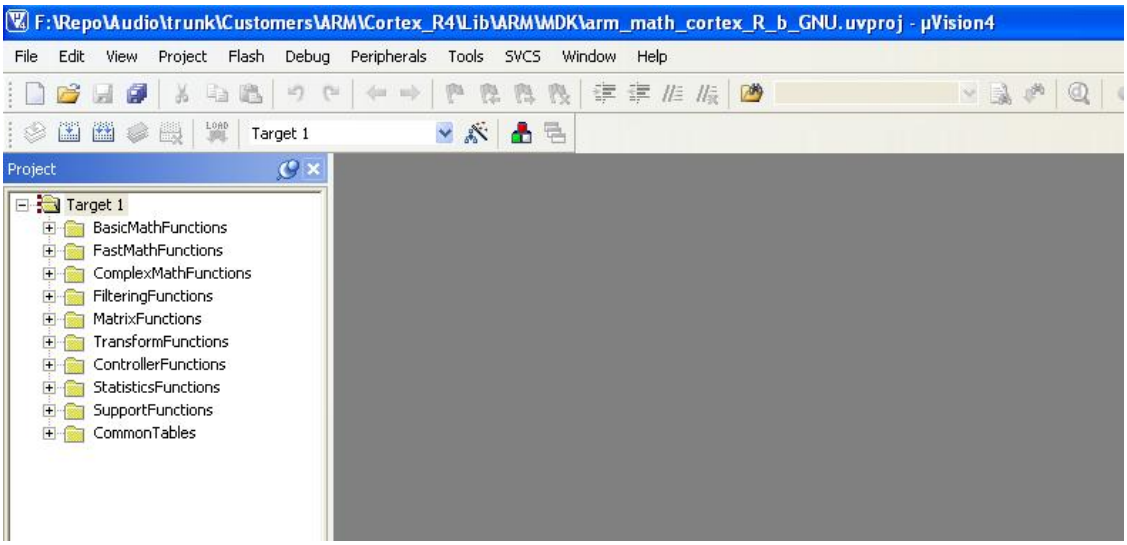
8.1. Tools used

8.1.1. Cortex-R

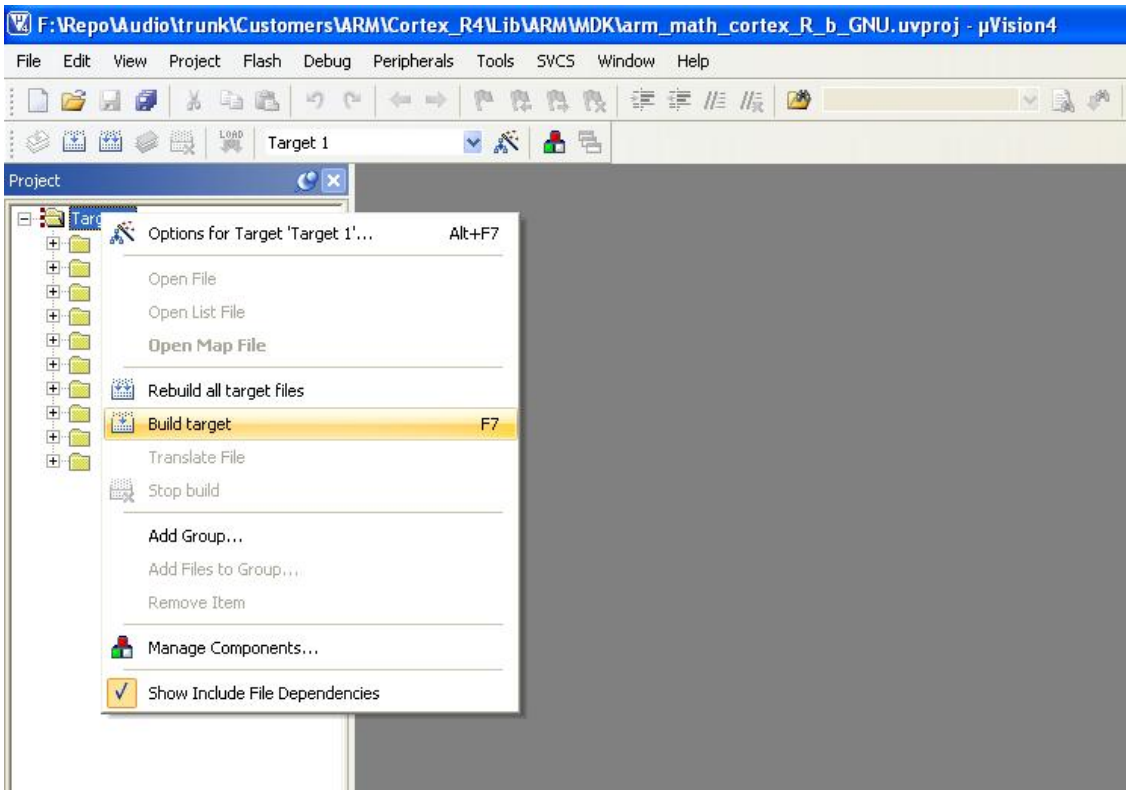
Tool: MDK version 4.21
ARM-ELF-GCC Compiler

8.2. Open and Build the Library Project

Double click on the arm_math_Cortex_R_b_GNU.uvproj library project to open in the Keil.



To build the library project, Right click on Target 1 in the project window and select Build target (F7) as shown below.


F:\Repo\Audio\trunk\Customers\ARM\Cortex_R4\Lib\ARM\MDK\arm_math_cortex_R_b_GNU.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Options for Target 'Target 1'... Alt+F7
Open File
Open List File
Open Map File
Rebuild all target files
Build target F7
Translate File
Stop build
Add Group...
Add Files to Group...
Remove Item
Manage Components...
Show Include File Dependencies