uEZ® New Project Creation Guide

Copyright ©2012, Future Designs, Inc., All Rights Reserved, Rev 1.0

FDI Future Designs, Inc.
Your Development Partner
2702 Triana Boulevard SW, Huntsville, AL 35805

Table of Contents

Information in this document is provided solely to enable the use of Future Designs products. FDI assumes no liability whatsoever, including infringement of any patent or copyright. FDI reserves the right to make changes to these specifications at any time, without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Future Designs, Inc. 2702 Triana Blvd, Huntsville, AL 35805.

For more information on FDI or our products please visit www.teamfdi.com.

NOTE: The inclusion of vendor software products in this kit does not imply an endorsement of the product by Future Designs, Inc.
© 2012 Future Designs, Inc. All rights reserved.

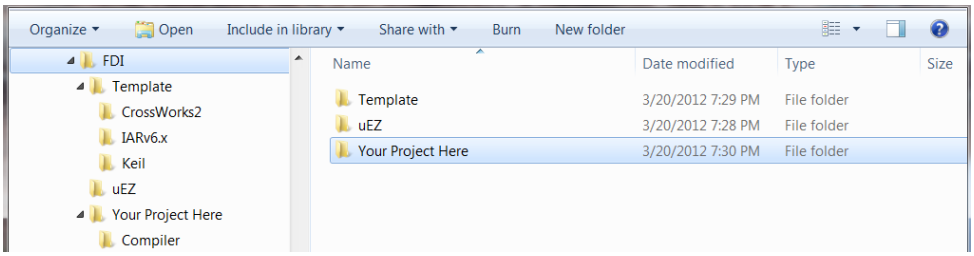
uEZ® is a registered trademark of Future Designs, Inc.
Microsoft, MS-DOS, Windows, Windows XP, Microsoft Word are registered trademarks of Microsoft Corporation.
Other brand names are trademarks or registered trademarks of their respective owners.

FDI PN: MA
Revision: 1.0, 5/22/2012
Printed in the United States of America

1. Background Information

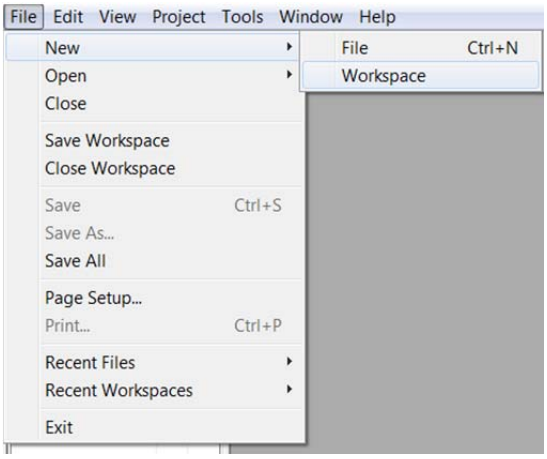
This guide covers creating new projects using the new uEZ v2.0 library setup. Projects for IAR, CrossWorks, and Keil are based on the uEZGUI-1788-43QWS.

It is recommended that the new project directory be in the same directory as the uEZ source code. For the purpose of this document we will be using the template directory to store the project created, for example an IAR project will created in “FDI/Template/IARv6.x”.

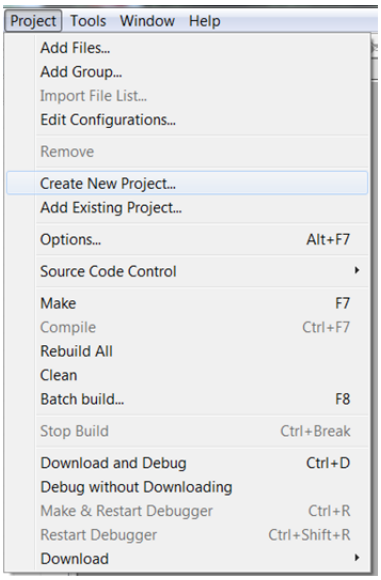


2. Creating a new IAR Project

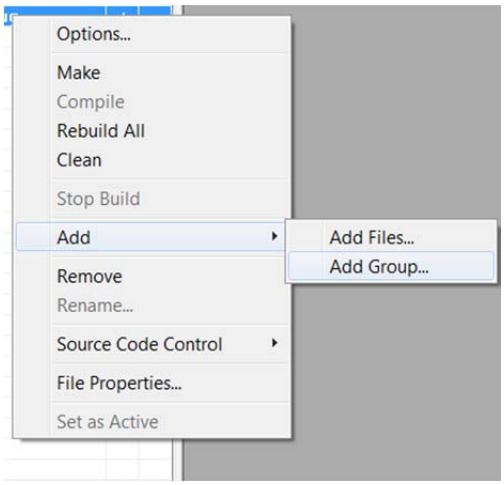
a. Create a new Workspace by clicking “File” -> “New” -> “Workspace”.



b. Create a new Project by clicking “Project” -> “Create New Project”.



c. Next create the new project by selecting the "Empty project" option to create a blank project. (C/C++ options will create a new main.c/main.cpp that won't be needed)
d. Next, we will need at add the basic source file to start off the new project. Located in the uEZ download in "Template/IARv6.x", this sample project can be found there as well.
e. "main.c" – This file handles the platform startup, loads the Nonvolatile settings, starts the heartbeat task, and host the entry point to start a custom application.
f. "Config_Build.h" -- This files setup the display that will be used and a few other basic settings.
g. "Source/Calibration.c" and "Source/Calibration.h" – Handle calibration for the touch screen and stores the values in the NVSettings.
h. "Source/NVSetting.c" and "Source/NVSettings.h" – Handle loading and storing the NVSettings for the EEPROM.
i. "Source/Audio.c" and "Source/Audio.h" – are used to play simple tones with the PWM.
j. "Config_App.h" – Application and system settings are store here, including the project name and information for the LCD controller to fuction.
k. "Source/Hello_World.c" and "Source/Hello_World.h" – These file are the sample application and use the NXP SWIM library to turn on the backlight and put text on the screen.
l. "flash_loader" and "flash_loader NOR" – used for loading code in IAR only.
m. Next, we will add the groups and the source files to the project.
    i. To create a group right click on the project name in the workspace and click "Add" -> "Add Group". Create the following groups.



1. “Source”
2. “uEZFiles”
3. “Platform”
4. “uEZ Library”

n. Add the Platform and uEZ Library into the uEZFiles Group.

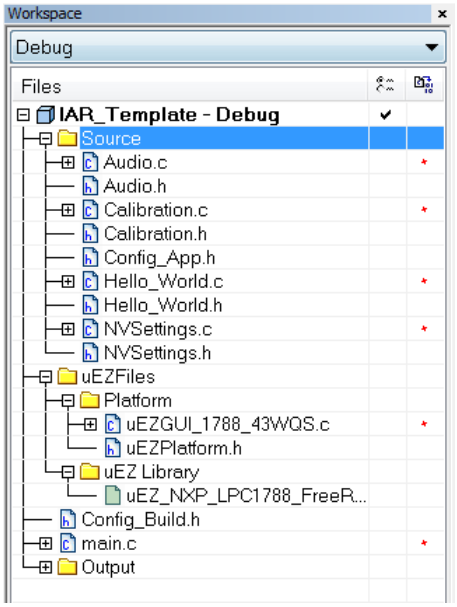
o. Now add the source file to the project by right clicking on the group that they should be added to, select “Add” -> “Add Files…”.

   i. Add “main.c” and “Config_Build.h” to the project.

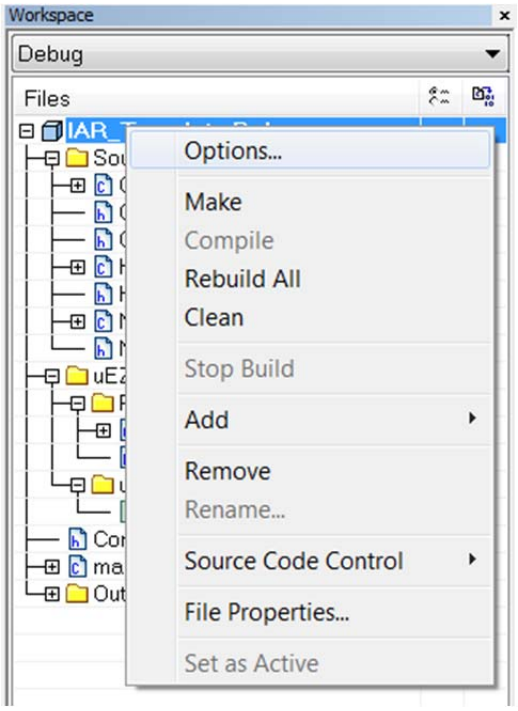
   ii. Add all of the files in the “Source” directory to the “Source” group.

   iii. Add “uEZ/Source/Platform/FDI/uEZGUI-1788-43WQS/uEZGUI-1788-43WQS.c” and “uEZ/Source/Platform/FDI/uEZGUI-1788-43WQS/uEZPlatform.h” to uEZFiles/Platform”.

   iv. Add the library located at “uEZ/Build/Generic/NXP/LPC1788/FreeRTOS/IAR6.x/Release/Exe/uEZ_NXP_LPC1788_FreeRTOS_IAR6_Release.a” to “uEZFiles/uEZ Library”. (make sure to select “Library/Object Files” from the drop down box for the file to be visible)



p. Next, setup the project setting by right clicking on the project name in the workspace and select “Options”. This will only cover the changes that need to be made for the debug project, the release version will be the same except for the Optimizations.



i. “General Options”
    1. Under “Target” change the device to “NXP LPC1788”

ii. “C/C++ Compiler”
    1. Under “Optimizations” change level to “None”.
    2. Under “Preprocessor” add the follow additional include directories.

        $PROJ_DIR$/.
        $PROJ_DIR$/Source
        $PROJ_DIR$/../../uEZ/
        $PROJ_DIR$/../../uEZ/Include
        $PROJ_DIR$/../../uEZ/Source/Platform/FDI/uEZGUI_1788_43WQS
        $PROJ_DIR$/../../uEZ/Source/Processor/NXP/LPC1788
        $PROJ_DIR$/../../uEZ/Source/Processor/NXP/LPC1788/IAR/include
        $PROJ_DIR$/../../uEZ/Source/RTOS/FreeRTOS/include
        $PROJ_DIR$/../../uEZ/Source/RTOS/FreeRTOS/portable/IAR/ARM_CM3

iii. Under “Extra Options” select “Use command line options”
        --no_unaligned_access
        --enum_is_int

iv. “Assembler”
    1. Under “Preprocessor” add the same additional included from “C/C++ Compliler”

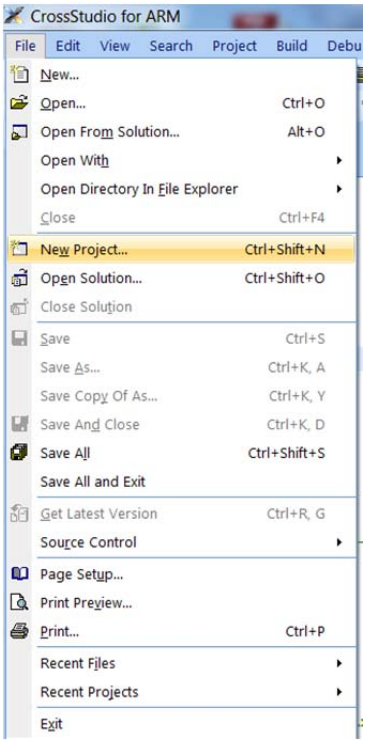
v. “Linker”
    1. Under “Config” select the “Override default” Linker configuration file and set it to “$PROJ_DIR$\uEZGUI_1788.icf”.

vi. "Debugger"
    1. Under "Setup" set the driver to "J-LinkJ/J-Trace" and set to use macro file(s) and use the file "$PROJ_DIR$\flash_loader\LPC177x_8x.mac".
    2. Under "Download" select to use flash loader(s) and Override default .board file and use the file "$PROJ_DIR$\flash_loader\uEZGUI_1788_70WVE.board".
vii. "J-Link/J-Trace"
    1. Under "Setup" change Reset to "Halt after bootloader".

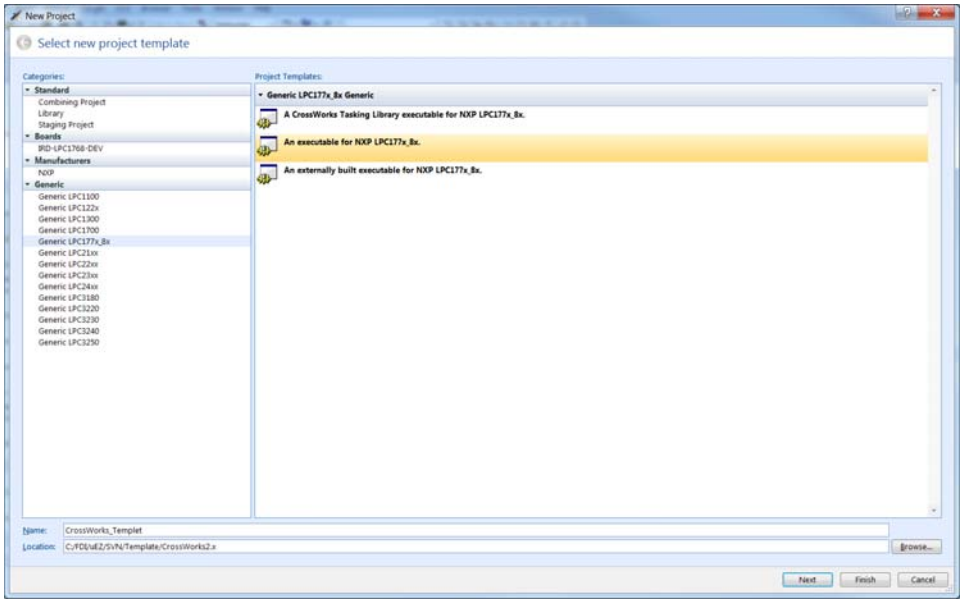
q. Click "File" then "Save All" to name and save the project. It is recommended that the projects are saved in a directory as the uEZ Source files, "<Project Name>/IAR".
r. The Project is now ready to be build and downloaded to the "uEZGUI-1788-43WQS". Other boards can be setup in a similar manner but using a different platform file.

3. Creating a new CrossWorks Project

a. Create a new project by clicking “File” -> “New Project”.

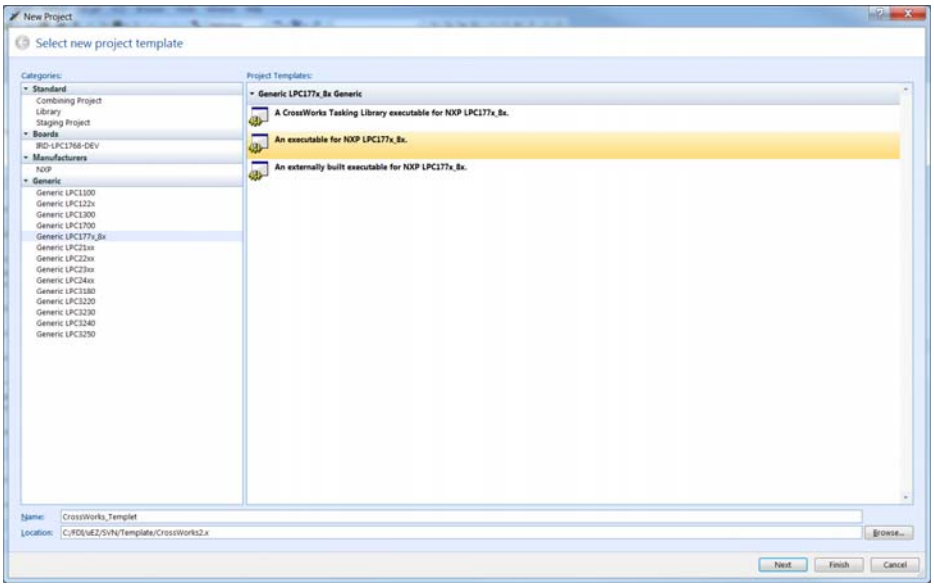


b. Select a “Generic LPC177x_8x” and “An executable for NXP LPC177x_8x”. Name the project and set the location to be in the same directory as the uEZ source files.

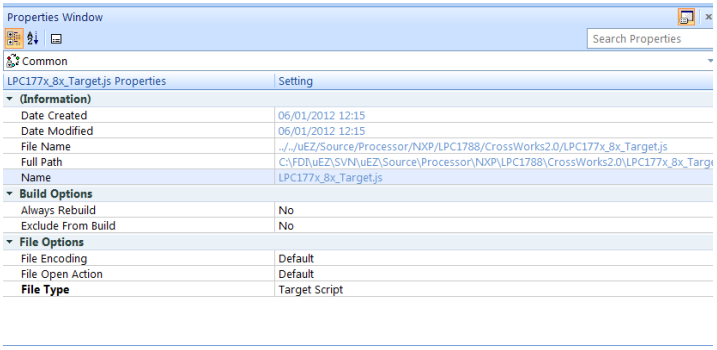


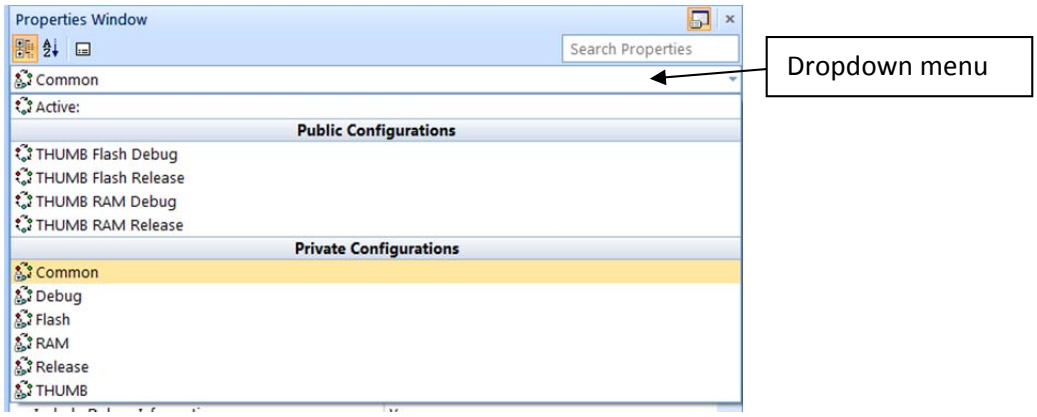
c. Click “next”. Leave the default setting on this page, click “next” again.

d. On the next page deselect all of the files under Links to system files.

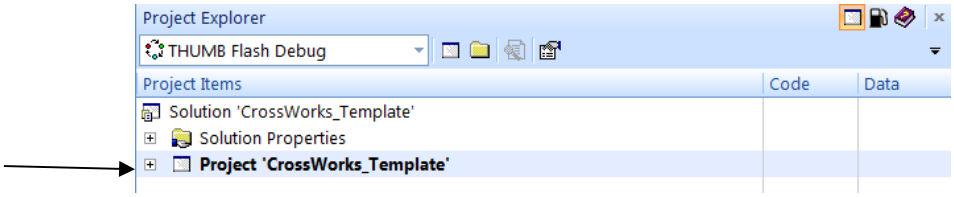


e. Click “next” and deselect “THUMB RAM Debug” and “THUMB RAM Release” and click finish.
f. The groups added by default can safely be deleted. (CMSIS, Source, and System Files)
g. Add the “new folders” by right clicking on the project and selecting “new folder”. (Folders can be added under other folders by right clicking on the folder and selecting “new folder”)
    i. Add the folders:
        1. “Source”
        2. “uEZ Files”
            a. “Platform”
            b. “uEZ Library”
            c. “Processor”
                i. “NXP”
                    1. “LPC1788”
                        a. “CrossWorks”
h. Add the source files to the project by right clicking and selecting “add existing file…”.
    i. Add “main.c” and “Config_Build.h” to the project.
    ii. Add all of the files in the “Source” directory to the “Source” group.
    iii. Add “uEZ/Source/Platform/FDI/uEZGUI-1788-43WQS/uEZGUI-1788-43WQS.c” and “uEZ/Source/Platform/FDI/uEZGUI-1788-43WQS/uEZPlatform.h” to uEZFiles/Platform”.
    iv. Add “uEZ/Source/Processor/NXP/LPC1788/CrossWorks2.0/LPC177x+8x_Target.js” to “uEZ Files/Processor/NXP/LPX1788/CrossWorks”.
        3. After adding the properties for this file will need to be setup. Highlight the file and select “Common” in the Properties Window drop down, under file options change the file type to “Target Script”.

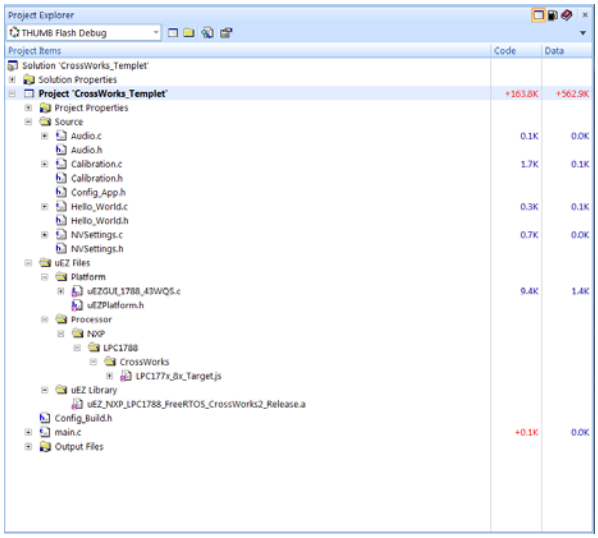




(Important note: when editing the properties below be sure that the project name is selected in the project explorer and the configuration is set to Common.)



v. Add the library located at "uEZ/Build/Generic/NXP/LPC1788/FreeRTOS/CrossWorks2.x/Release/ uEZ_NXP_LPC1788_FreeRTOS_CrossWorks2_Release.a" to the uEZ Library folder.

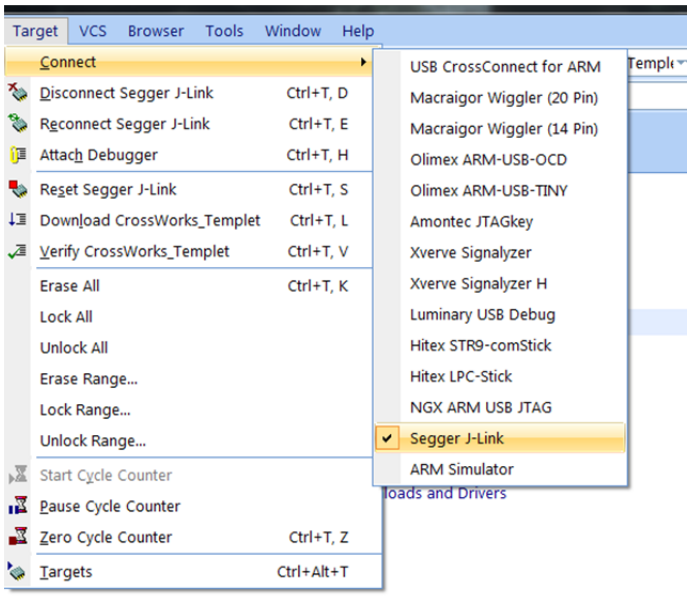


i. Next the project setting need to be setup for the code to compile. Highlight the Project in the Project Explorer and refer to the Properties Window, select "Common" in the drop down menu.

j. Memory Map File – Change to “uEZGUI-1788_MemoryMap.xml” in the project directory.
k. Section Placement File – change to “uEZGUI-1788_placement.xml”. This entry also needs to be set for THUMB Flash Debug and THUMB Flash Release.
l. Preprocessor Options – enter the options “USE_PROCESS_STACK”, “GCC_ARMCM3”, and “STARTUP_FROM_RESET” under “Preprocessor Definitions”.
m. User Include Directories – add the following user include directories. (under Preprocessor Options)

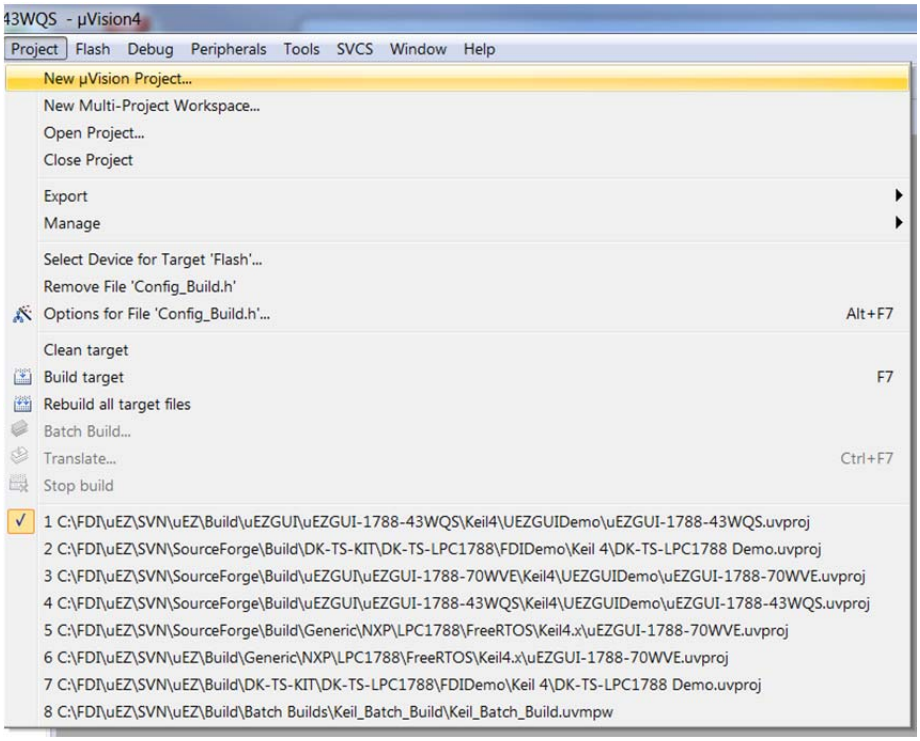
    .
    ../../.
    ./Source
    ../../uEZ
    ../../uEZ/Include
    ../../uEZ/Source/RTOS/FreeRTOS/include
    ../../uEZ/Source/Platform/FDI/uEZGUI_1788_43WQS
    ../../uEZ/Source/Processor/NXP/LPC1788
    ../../uEZ/Source/Processor/NXP/LPC1788/CrossWorks2.0/include

n. Heap Size – set to 500,000 bytes.
o. Main Stack Size – set to 2,048 bytes.
p. The project can now be built by pressing F7.
q. Connect to the Segger J-Link by clicking “Tartget” -> “Conncect” -> Segger J-Link”

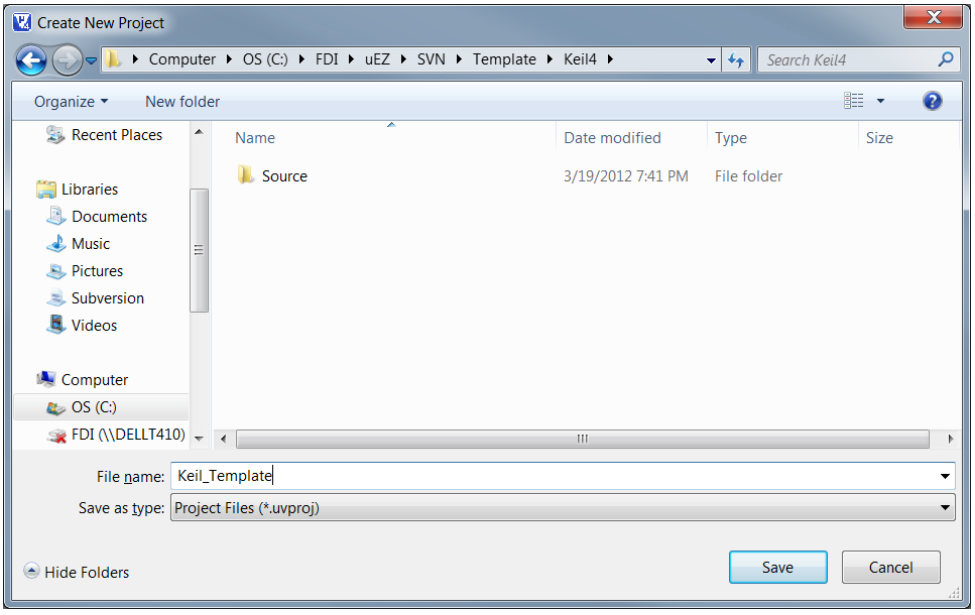


4. Creating a new Keil Project

a. Create a new Project by selecting “Project” -> “New uVision Project”.



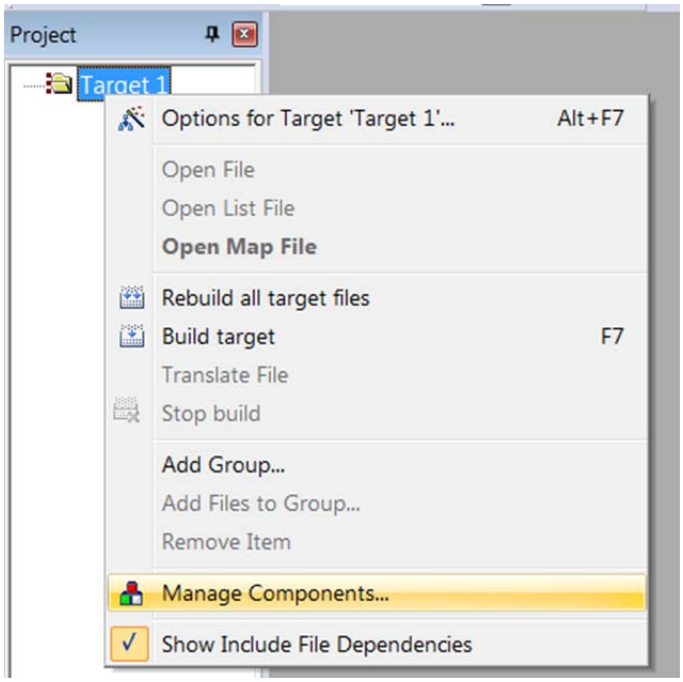
b. Name the project and save it in a directory with the uEZ source code.



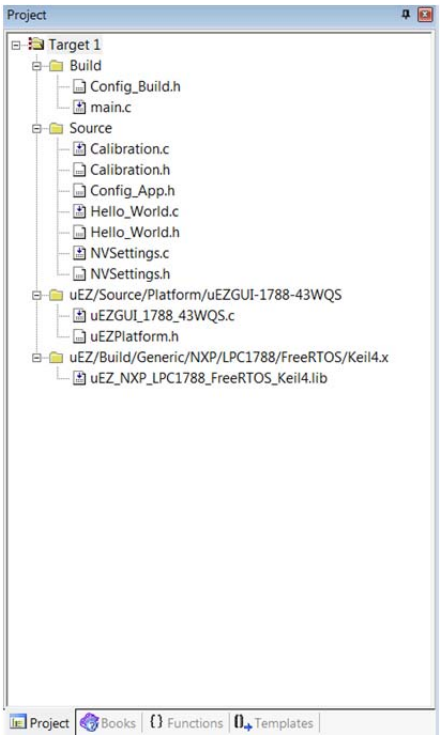
c. Select the NXP LPC1788 and click “OK”.

d. Click no to copy the startup code to the project folder.

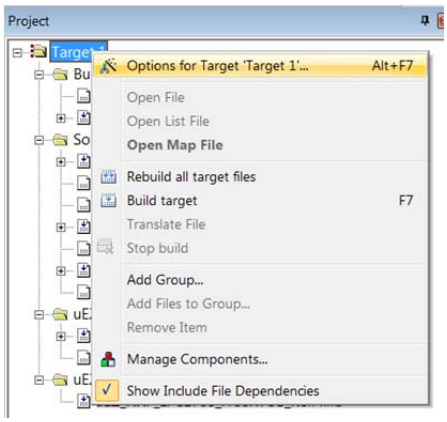
e. Added the source files and groups by right clicking Target 1 or Debug and selecting “Manage Components”.



f. Create the following groups.
   i. “Build”
   ii. “Source”
   iii. “uEZ/Source/Platform/uEZGUI-1788-43WQS”
   iv. “uEZ/Build/Generic/NXP/LPC1788/FreeRTOS/Keil4.x”
g. Add the source files to the new goups.
   i. “Build”
      4. “main.c” and Config_Build.h”
   ii. “Source”
      5. Add all the files from the Source directory.
   iii. “uEZ/Source/Platform/uEZGUI-1788-43WQS”
      6. Add the files “uEZ/Source/Platform/FDI/uEZGUI-1788-43WQS/uEZGUI-1788-43WQS.c” and “uEZ/Source/Platform/FDI/uEZGUI-1788-43WQS/uEZPlatform.h” to uEZ/Source/Platform”
   iv. “uEZ/Build/Generic/NXP/LPC1788/FreeRTOS/Keil4.x”
      7. Add the file “uEZ/Build/Generic/LPC1788/FreeRTOS/Keil4.x/Debug/uEZ_NXP_LPC1788_FreeRTOS_Keil4.lib”



h. Right Click Target 1 or Debug and select options.

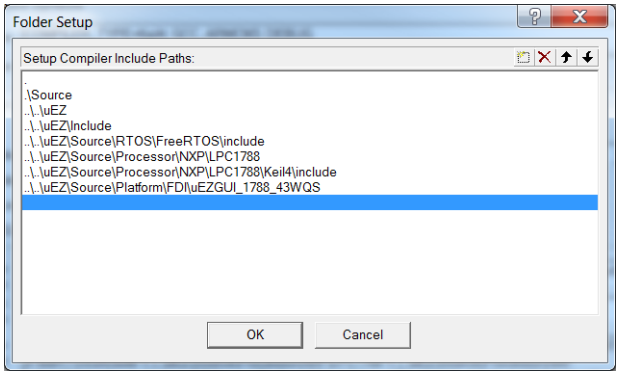


v. Target – Check Use MicroLIB.

vi. Output – Here you can change the folder for objects to be a directory inside the project if you want to keep the project folder a little cleaner.

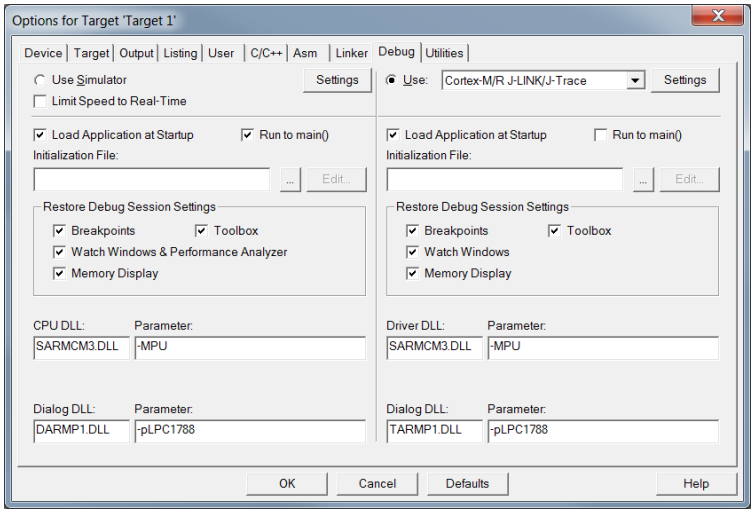
vii. C/C++ -- Change the Preprocessor Sybmbols Define: to “COMPILER_TYPE=Keil4, GCC_ARMCM3, DEBUG”.

viii. C/C++ -- change the Include Paths to.

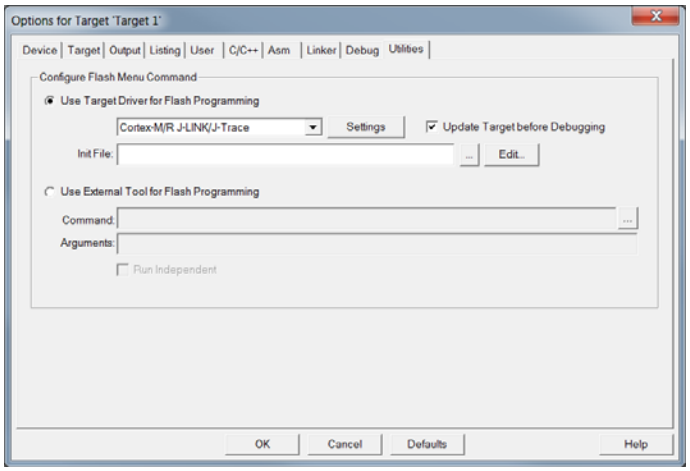


ix. Linker – deselect Use Memory Layout from Target Dialog and enter ".\uEZGUI-1788.sct for the Scatter File.

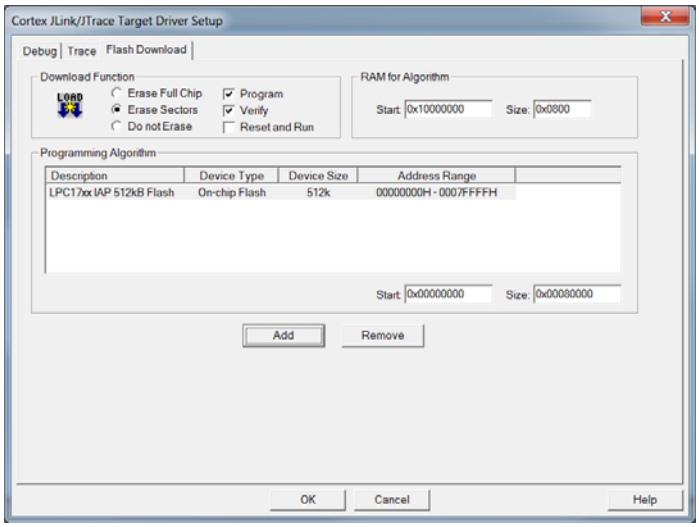
x. Debug – Select to use the Cortex-M/R J-Link/J-Trace.



xi. Utilities – Change the drop down box to be Cortex-M/R J-Link/J-Trace.



8. Click Settings next to the dropdown box.
9. Add the Programming Algorithm LPC17xxIAP 512kB Flash.



i. Click OK to exit the options menus.
j. Press F7 to compile the code.