



STMicroelectronics SensorTile Tutorial: Firmware Programming



Table of Contents

1. INTRODUCTION TO THIS TUTORIAL.....	3
1.1. LIST OF REQUIRED EQUIPMENT AND MATERIALS.....	3
1.2. PREREQUISITE TUTORIALS.....	3
2. ADDING NEW SOURCE FILES, PATHS, AND PROJECT CONFIGURATION	4



1. Introduction to This Tutorial

The Tutorial steps provide:

1. An introduction to customize the starter firmware, STSW-STLKT01 package
2. Guidance to add new source files, create new linking paths, and properly configure a project.
3. Demonstration of the customized DataLog application, which will call a function from a newly added source to calculate the acceleration vector amplitude.

For more information regarding the SensorTile board, please open the following link.

www.st.com/sensortile

1.1. List of Required Equipment and Materials

- 1) 1x STMicroelectronics SensorTile kit.
- 2) 1x STMicroelectronics Nucleo Board.
- 3) 1x Personal Computer with two USB type-A inputs OR you must have a powered USB hub.
- 4) 1x USB 2.0 A-Male to Micro-B Cable (micro USB cable).
- 5) 1x USB 2.0 A-Male to Mini-B Cable (mini USB cable).
- 6) Network access to the Internet.

1.2. Prerequisite Tutorials

It is recommended that users have completed and are familiar with the contents of the following tutorials before proceeding.

1. Introduction to SensorTile and the System WorkBench Integrated Development Environment (IDE)
2. Sensor System Signal Acquisition, Event Detection and Configuration.



2. Adding New Source Files, Paths, and Project Configuration

It is not uncommon to split a complete project into multiple source and header files for elegance and clearness. You may have noticed that the STSW-STLKT01 package manages the projects exactly in this way. We will now go through the process of adding your own source file, linking the corresponding header file, and configuring your project appropriately. This enables you to customize your own project.

1. Navigate to the local folder containing the STSW-STLKT01 package “**../v1.2.0/**” and create a subdirectory “**Tutorial5**”, as shown in Figure 1.

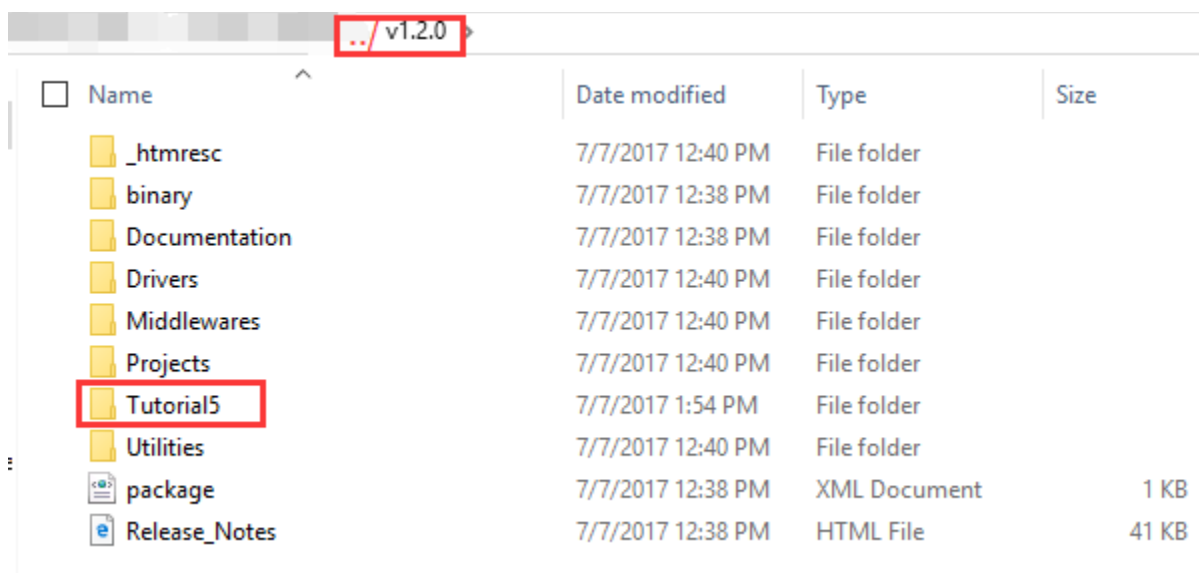


Figure 1: Add new directory into current project source file directory

2. Navigate to “**Tutorial5**” and download **SensorTile_abs_acc.c** and **SensorTile_abs_acc.h** into the folder. See Figure 2.

You can download these files from the link:

http://www.st.com/content/st_com/en/premium-content/sensortile-curriculum-SensorTile_abs_acc_zip.html

Fill out the form and a download link will be sent to your email.





../ v1.2.0 > Tutorial5				
<input type="checkbox"/> Name	Date modified	Type	Size	
 SensorTile_abs_acc.c	7/7/2017 12:56 PM	C Source	1 KB	
 SensorTile_abs_acc.h	7/7/2017 12:55 PM	C/C++ Header	1 KB	

Figure 2: Download required files to the new directory.

3. Open the IDE (Eclipse or System WorkBench) on your personal computer as instructed in the document labelled **STMicroelectronics SensorTile Tutorial: Introduction to STMicroelectronics Development Environment and DataLog Project Example**.

Select the same workspace as in Tutorial 1.

4. **Save and remove** all other active projects in System WorkBench. Import the **DataLog** project as instructed in the previous tutorials.



5. Once you successfully import the DataLog project again, select **STM32L4xx-SensorTile -> DataLog -> User** in the Project Explorer, **right-click on User**, and **select import**. Make sure that the User folder is highlighted in blue. See Figure 3.

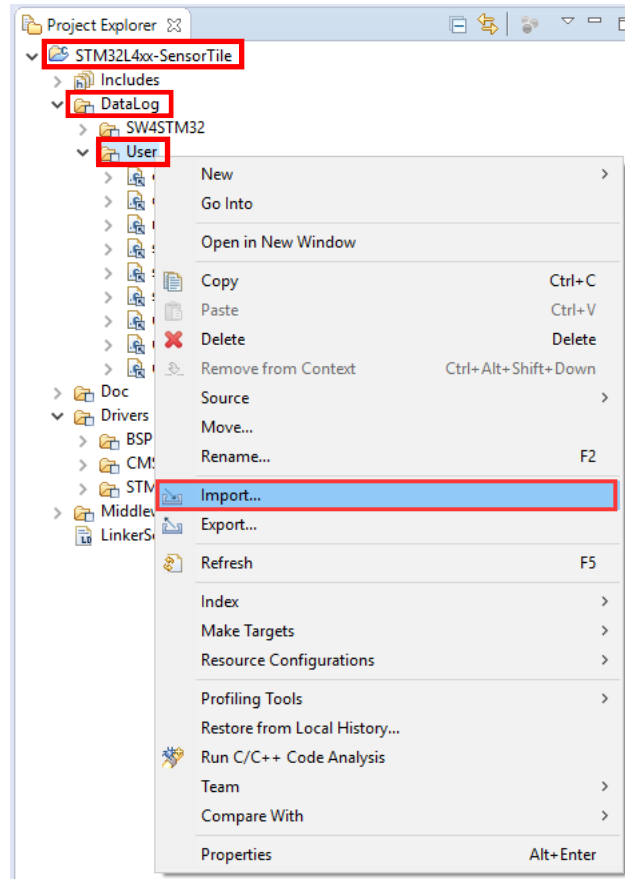


Figure 3: Importing a new source file



6. In the pop-up window, select **General -> File System** and then Press Next. See Figure 4.

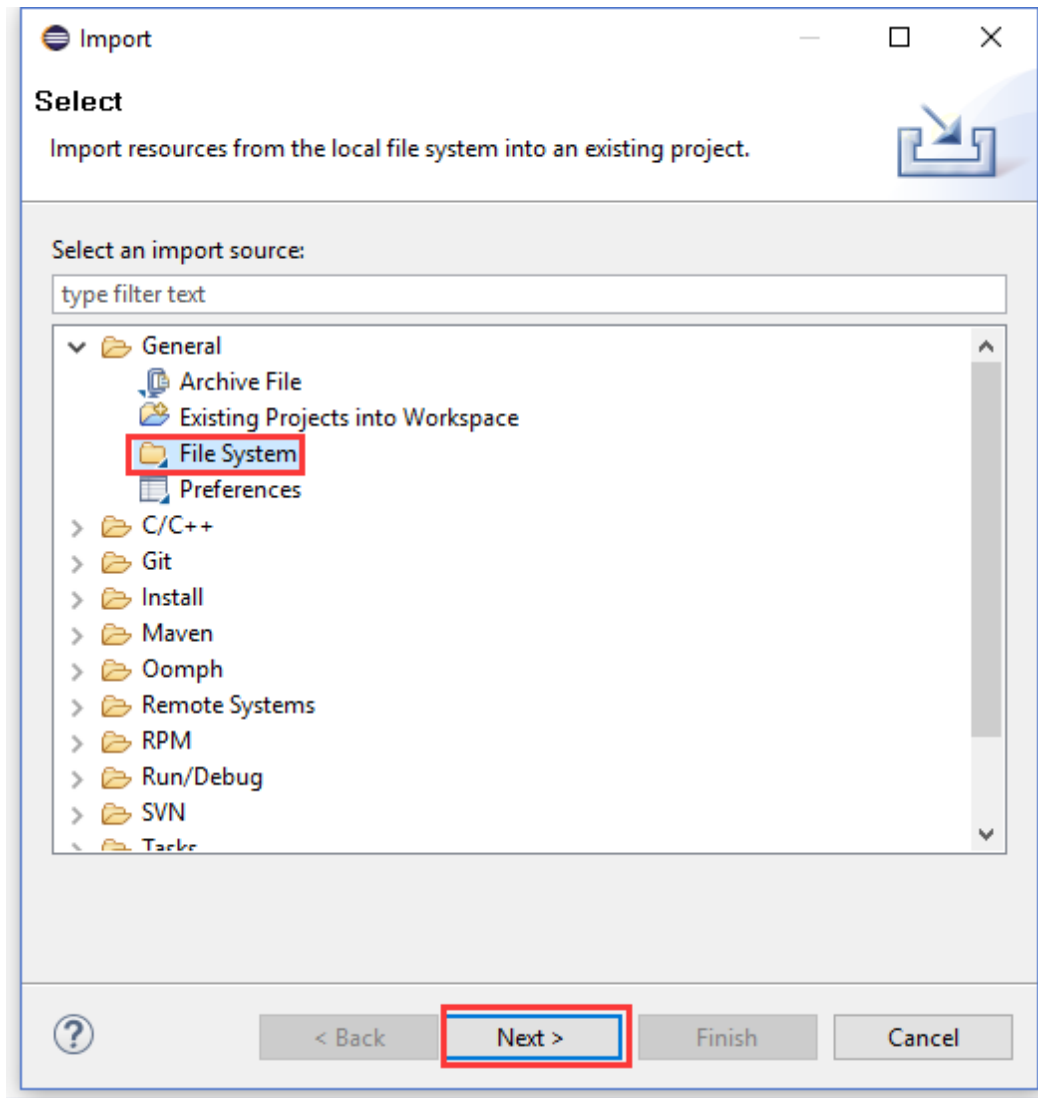


Figure 4: Importing a source file from File System.



7. In the next pop-up window, click on the top “Browse” button to select the “Tutorial5” folde, as shown in Figure 5. Make sure that the “Into folder” is correct, as shown in step 3 of Figure 5.

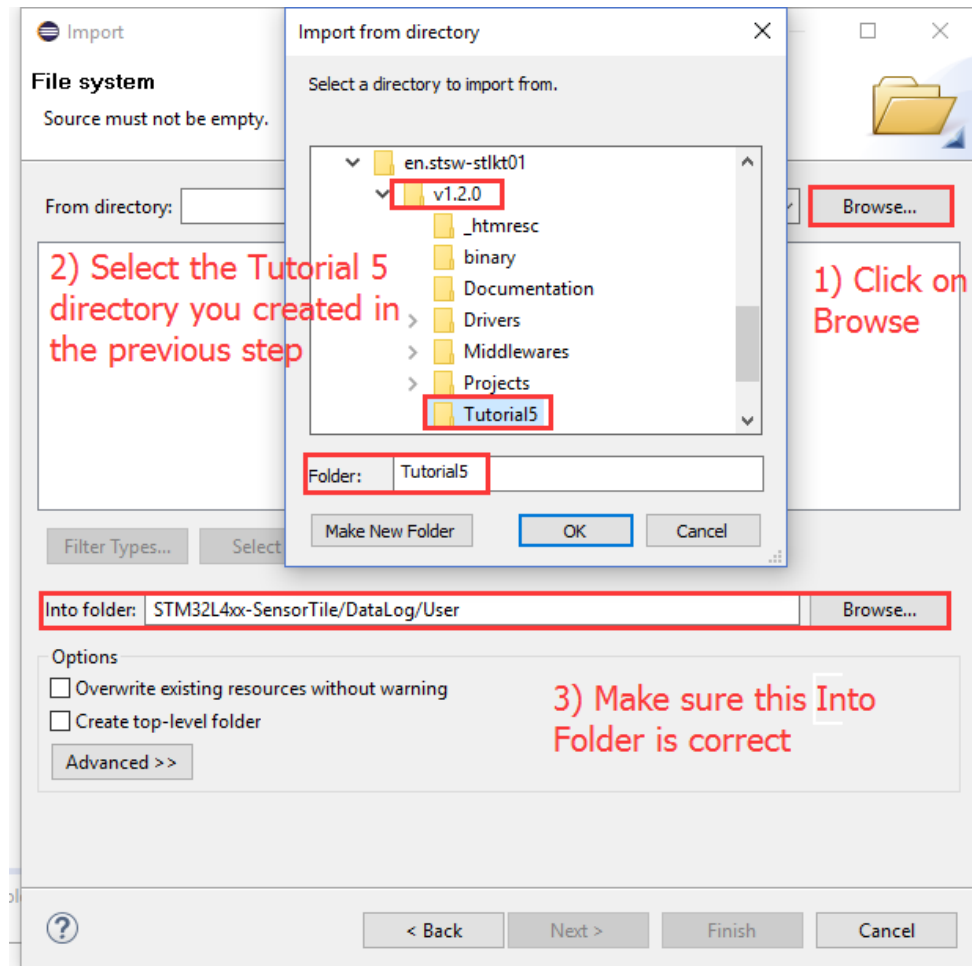


Figure 5: Selecting the directory containing the source code.



8. Check the file to be imported, click on **Advanced** and check **Create links in workspace**, and click **Finish**. See Figure 6.

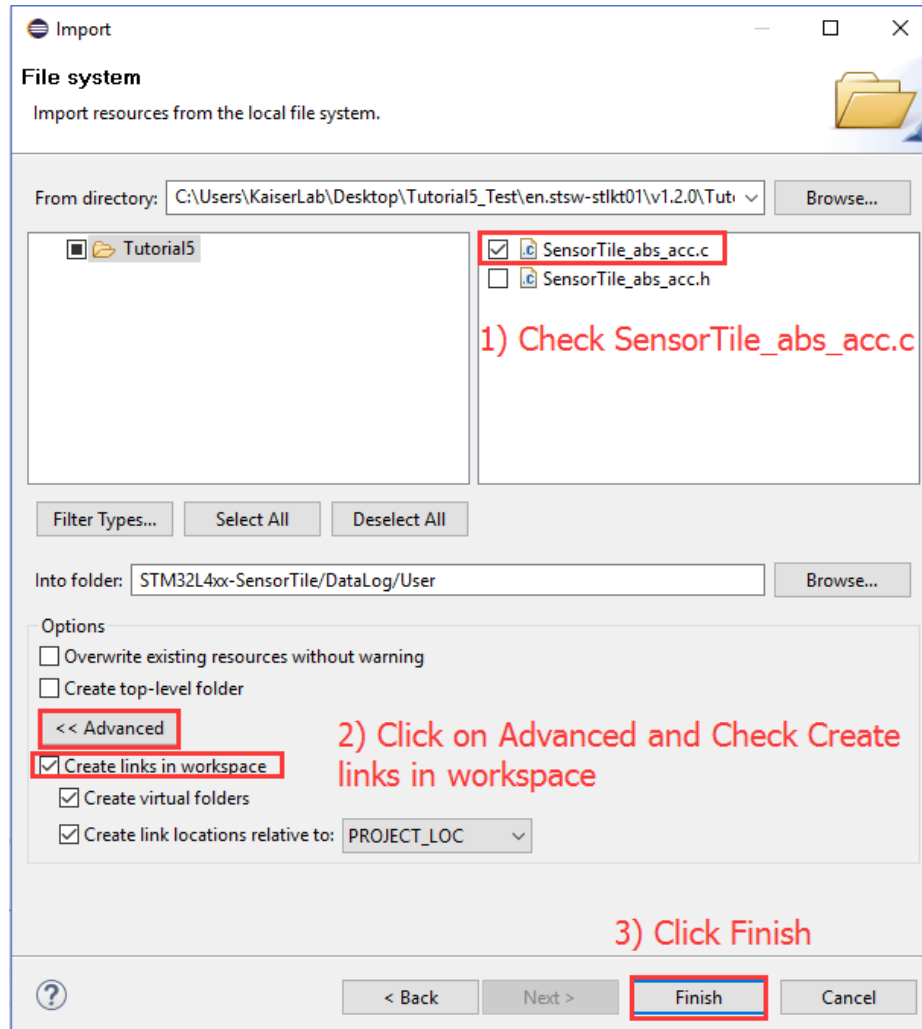


Figure 6: Checking import files and creating links in workspace in Advanced settings.



9. Make sure the file is imported successfully into the correct folder, as shown in Figure 7.

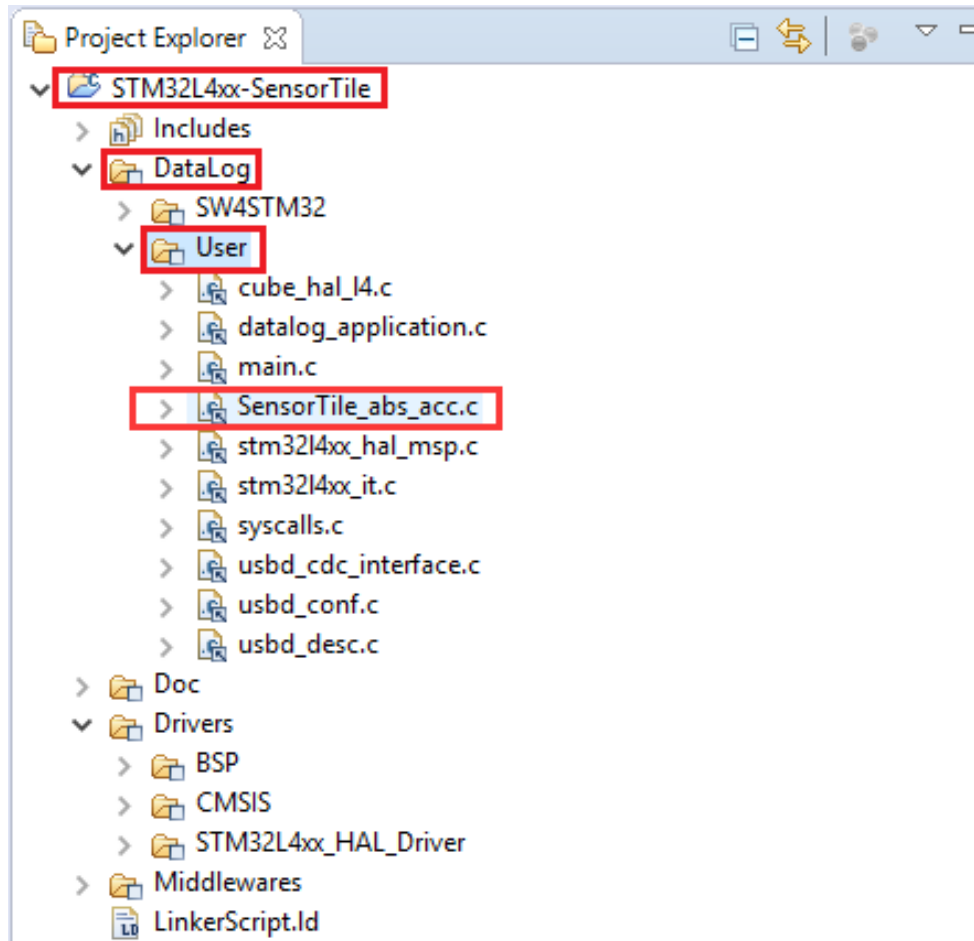


Figure 7: Imported SensorTile_abs_acc.c.



10. Right-click on **STM32L4xx-SensorTile** and select **Properties**. See Figure 8.

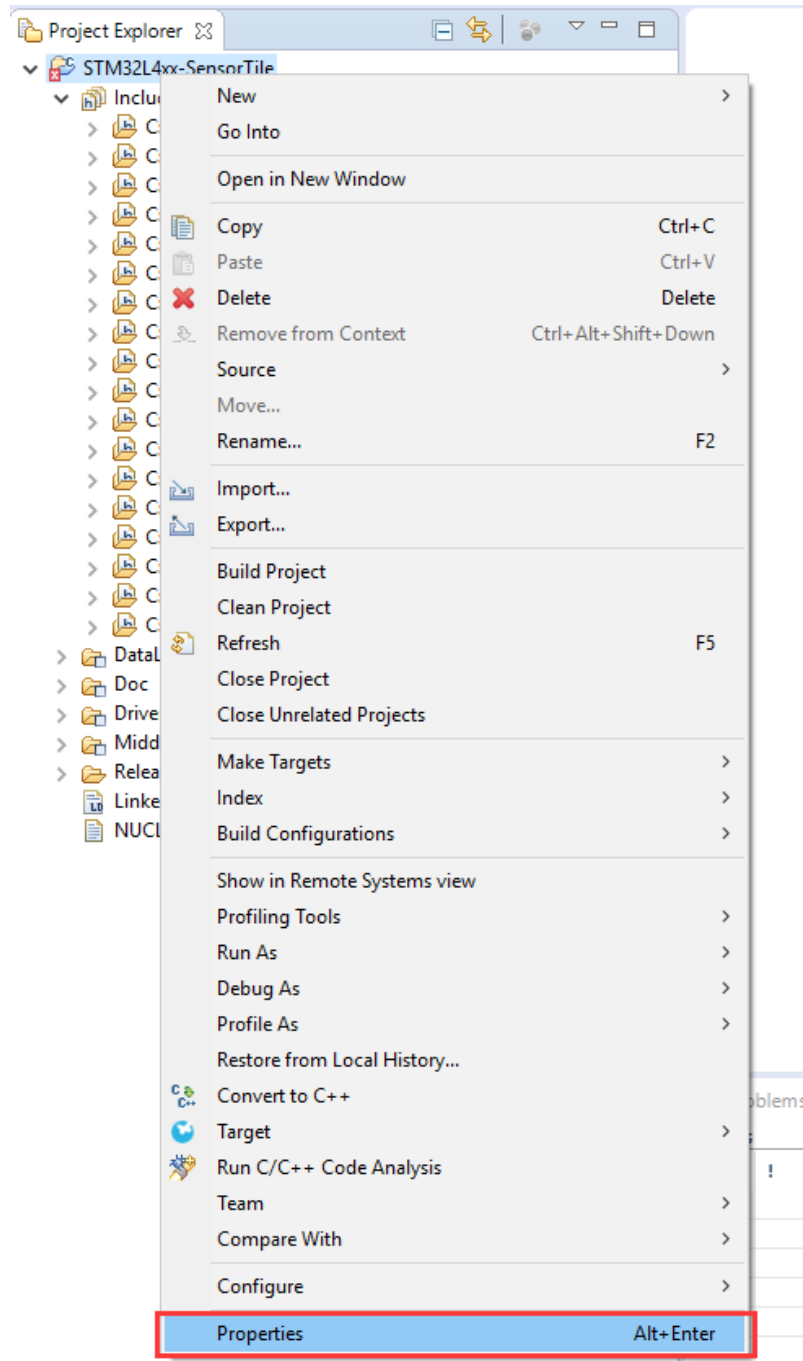


Figure 8: Open project properties.



11. Add the header to the includes paths as instructed in Figure 9. This enables System WorkBench to find the dependent header files for the related sources files.

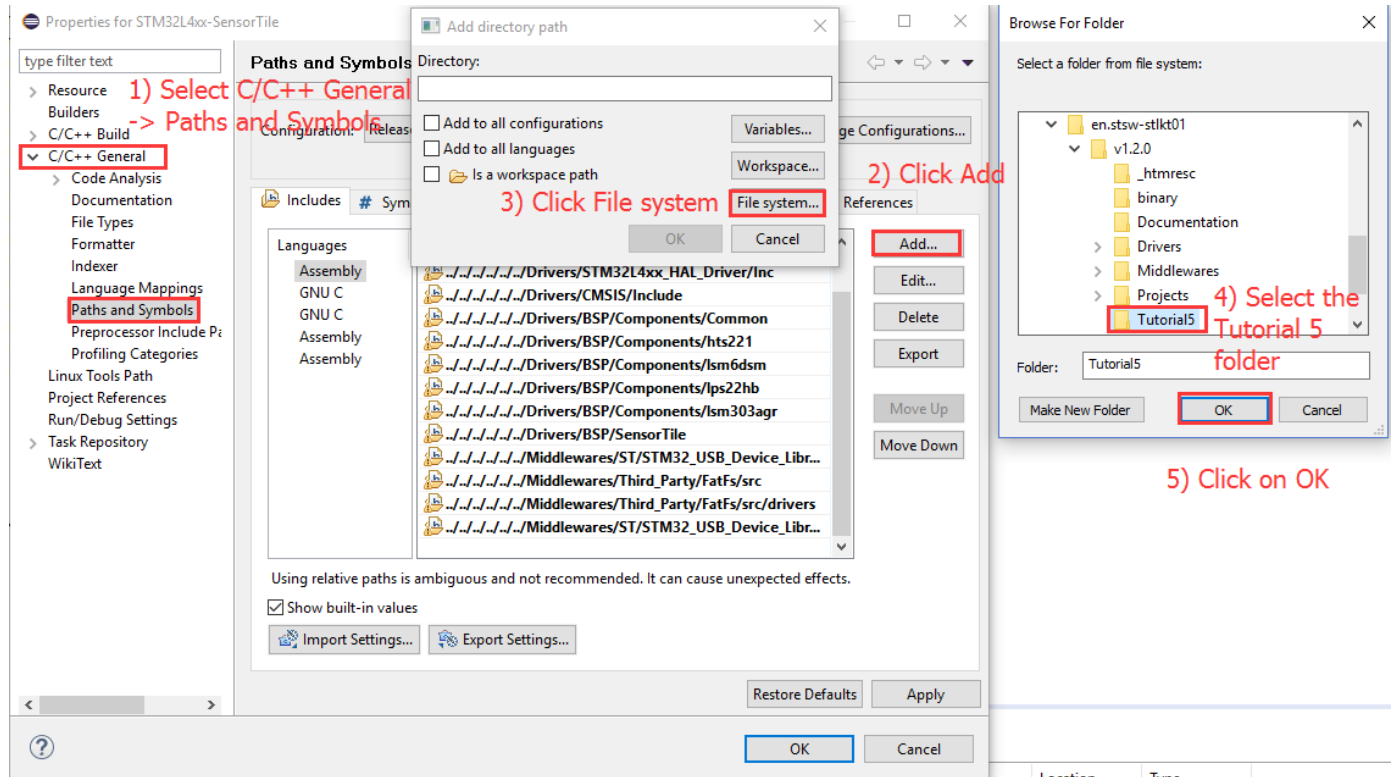


Figure 9: Adding a new path.

12. Check **Add to all configuration** and **Add to all languages**. Press OK. See Figure 10.

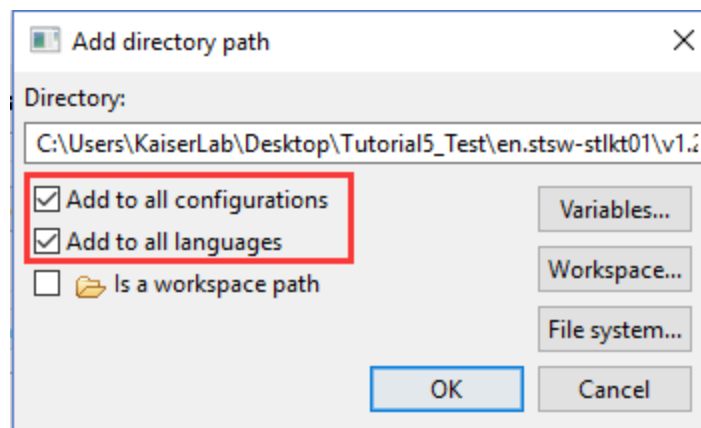


Figure 10: Adding directory path to all configurations and all languages.



13. You can see that Tutorial 5 directory is successfully added into the project path. See Figure 11.

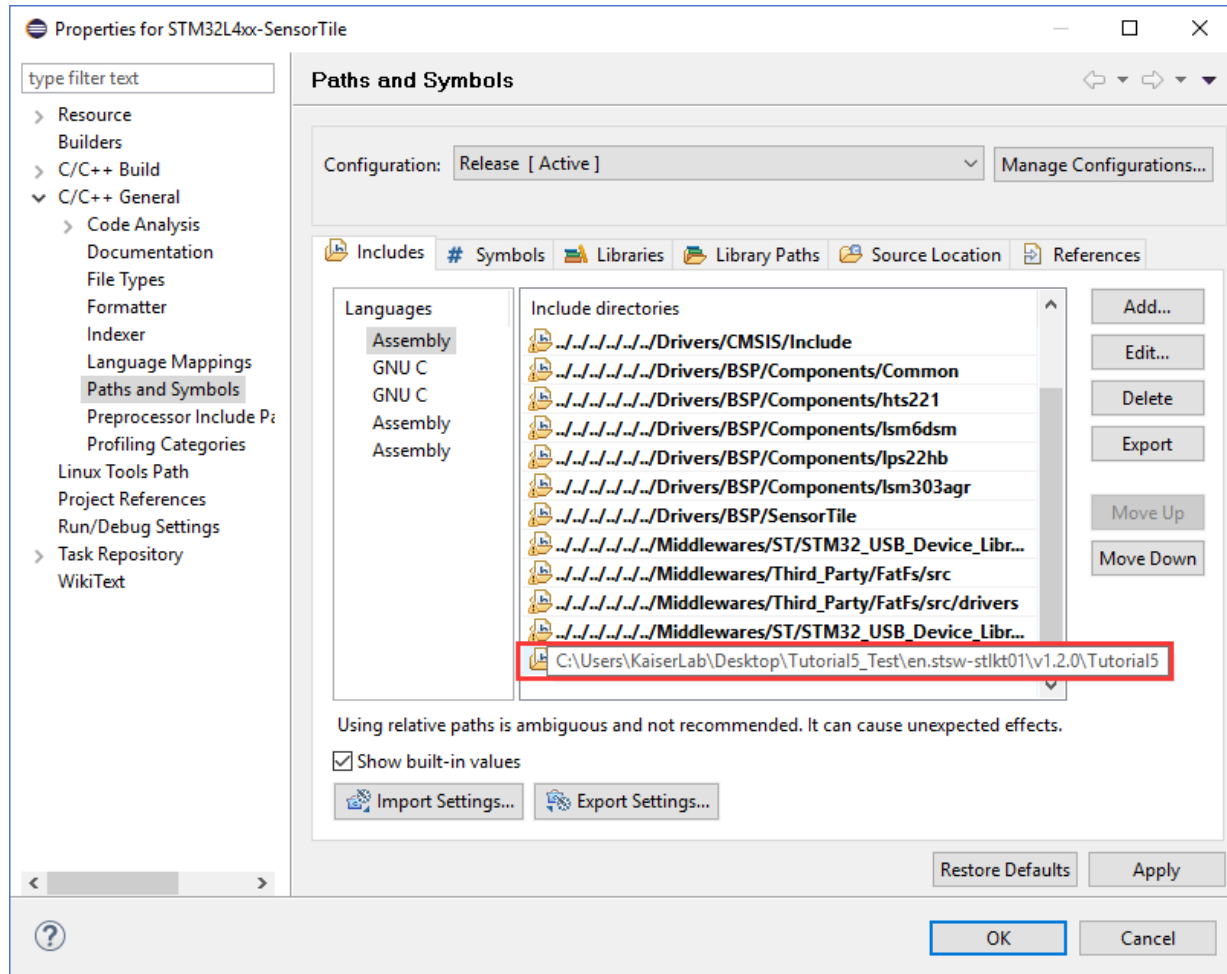
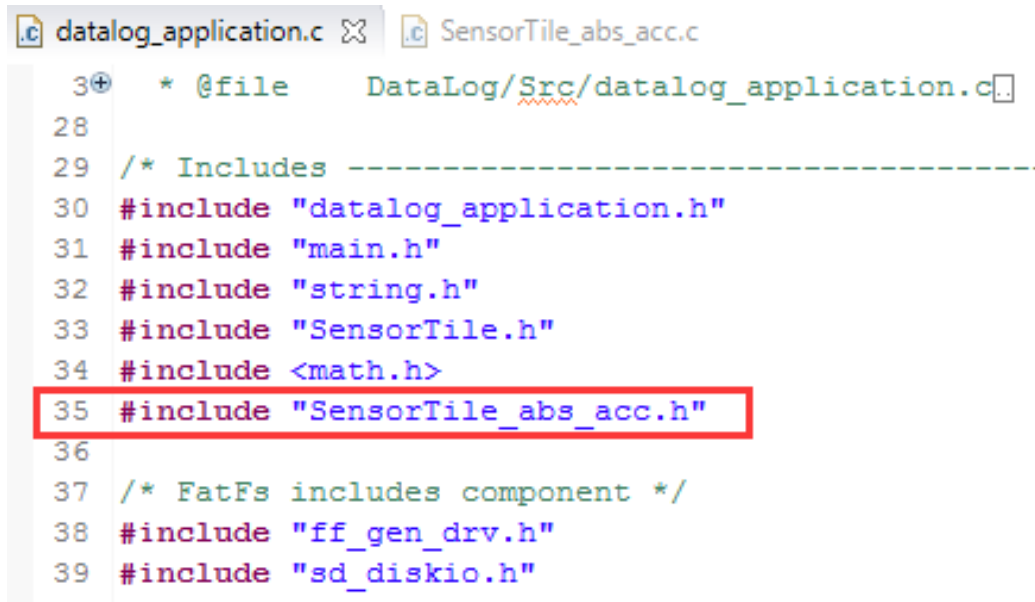


Figure 11: Newly added directory path.



14. Now we need to navigate to `datalog_application.c` and include the header file. See Figure 12.



```

3+  * @file      DataLog/Src/datalog_application.c
28
29  /* Includes -----
30  #include "datalog_application.h"
31  #include "main.h"
32  #include "string.h"
33  #include "SensorTile.h"
34  #include <math.h>
35  #include "SensorTile_abs_acc.h"
36
37  /* FatFs includes component */
38  #include "ff_gen_drv.h"
39  #include "sd_diskio.h"

```

Figure 12: Add header files.

15. Open the declaration for the function labelled **Accelero_Sensor_Handler** as instructed in the document labelled **STMicroelectronics SensorTile Tutorial: Sensor System Signal Acquisition, Event Detection and Configuration**.



16. Modify the code as shown in Figure 13.

```
if(SendOverUSB) /* Write data on the USB */
{
    sprintf( dataOut, "\n\rACC_X: %d, ACC_Y: %d, ACC_Z: %d, |ACC|: %d",
            (int)acceleration.AXIS_X, (int)acceleration.AXIS_Y, (int)acceleration.AXIS_Z,
            get_abs_acc(acceleration.AXIS_X, acceleration.AXIS_Y, acceleration.AXIS_Z));
    CDC_Fill_Buffer(( uint8_t * )dataOut, strlen( dataOut ));

    if ( verbose == 1 )
    {
        if ( BSP_ACCELERO_Get_WhoAmI( handle, &who_am_i ) == COMPONENT_ERROR )
        {
            sprintf( dataOut, "WHO AM I address[%d]: ERROR\n", id );
        }
        else
        {
            sprintf( dataOut, "WHO AM I address[%d]: 0x%02X\n", id, who_am_i );
        }

        CDC_Fill_Buffer(( uint8_t * )dataOut, strlen( dataOut ));

        if ( BSP_ACCELERO_Get_ODR( handle, &odr ) == COMPONENT_ERROR )
        {
            sprintf( dataOut, "ODR[%d]: ERROR\n", id );
        }
        else
        {
            floatToInt( odr, &d1, &d2, 3 );
            sprintf( dataOut, "ODR[%d]: %d.%03d Hz\n", (int)id, (int)d1, (int)d2 );
        }
    }
}
```

Figure 13: Calling the `get_abs_acc()` function to calculate acceleration vector magnitude.

17. Terminate and remove all previous applications from the SensorTile board.
18. Compile and run the DataLog application on the SensorTile board in debug mode.
19. Examine the data transmitted over the Serial USB connection to your personal computer. Take a screenshot.