STMicroelectronics SensorTile Tutorial: Introduction to STMicroelectronics Development Environment and DataLog Project Example for Apple Mac Platforms

Table of Contents

1. Introduction

The SensorTile is a new Internet of Things (IoT) system provided by STMicroelectronics integrating state-of-the-art processor, wireless interfaces, and sensor systems. The SensorTile can form the foundation for wearable consumer devices, wearable medical devices, residential IoT systems and vehicle IoT systems.

The SensorTile system provides an exceptionally powerful and well-supported platform for introduction to IoT technology. The SensorTile is remarkably compact as shown in Figure 1.

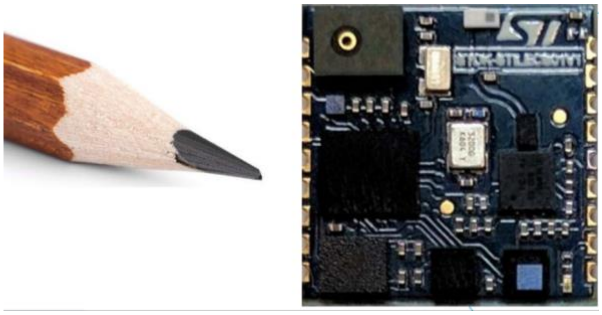


Figure 1. The STMicroelectonics SensorTile Platform with a pencil reference indicating its compact geometry.

The SensorTile includes these components:

1) The SensorTile Processor System is an STM32L4 microprocessor based on the ARM Cortex M4 system. This provides introduction to the ARM processor architecture that is deployed on nearly every smartphone on earth.
2) The SensorTile Sensors includes:
   a) The LSM6DSM combining microaccelerometer and microgyroscope.
   b) The LSM303AGR combining microaccelerometer and magnetometer for compass heading
   c) The LPS22HB barometric pressure sensor for determination of altitude and atmospheric pressure.
   d) The MP34DT04 microphone
3) The SensorTile also includes a Bluetooth Low Energy (Bluetooth Smart) wireless interface the BlueNRG-MS system.
4) The SensorTile also includes non-volatile flash storage that stores the executable code that enables IoT system operation.
5) The SensorTile also includes a cradle accessory with additional features including:
   a) SD Card Flash Storage System
   b) STC3115 Battery Monitor providing detailed energy monitoring for the SensorTile
   c) HTS221 Humidity and Temperature environmental sensors

2. Introduction to This Tutorial

This Tutorial introduces the development environment for the SensorTile system.

Development environments are essential to development of software for IoT systems and other products. These provide support to developers for both creation of systems, testing, debugging, and installation of software systems on platforms.

This development environment is referred to as an Integrated Development Environment (IDE). This includes all of the software tools required to create a software distribution for the SensorTile, compile this software system into the processor instruction set using a Build capability, execute this system using a Debug capability, and also create an “image” file that can be installed in the SensorTile non-volatile storage.

This tutorial is intended for users that have Apple Mac platforms, and will guide users through the tasks listed below. *Please note that there is a companion Tutorial for users that have personal computers with the Windows operating system.*

The Tutorial steps include:

1. Installing an Integrated Development Environment (IDE) on Mac.
2. Obtaining reference design example project software. This will specifically include a sensor Data Logging system.
3. Usage of the IDE to Import, Build, Run, Debug and Flash the SensorTile board to run the example Data Logging project.

For more information regarding the SensorTile board, please open the following link on a web-browser on your Mac.

www.st.com/sensortile

2.1. List of Required Equipment and Materials

1) 1x STMicroelectronics SensorTile kit.
2) 1x STMicroelectronics Nucleo Board.
3) 1x Mac 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.

3. Integrated Development Environment Installation

This portion of the document will guide users through the System WorkBench Integrated Development Environment (IDE) installation process.

3.1. Prerequisite Steps

1. Open the following link to register an account with the OpenSTM32 community.

   http://www.openstm32.org/tiki-register.php

2. Update your Mac's operating system (OS) to the latest version (OS X 10.10 Yosemite). However, please **do not update** to High Sierra due to the lack of support for tools needed here. (We will be working to resolve this in the future).

   Open the following link on a web-browser on your Mac for more details.

   https://support.apple.com/en-us/HT201475

3. Download and Install the latest version of Xcode from the App store. Open the following link on a web-browser on your Mac for more details.

   https://developer.apple.com/xcode/

4. Accept the license agreement by opening the terminal utility and issuing the following command.

   ```
   $ sudo xcodebuild -license
   ```

   When prompted to enter your password, enter the password used to log into your user account on your Mac.

   Press the **[Space]** key to scroll through the license.

   Once you reach the end of the license, type ***agree*** followed by the **[Enter]** key.

3.2. Install System WorkBench (IDE)

1. Open the following link on a web-browser on your Mac to download the System Workbench Installer.

   https://drive.google.com/open?id=1Tgtktoqpkq_5b71bX4CAHy1_hwkIde7T

2. Move the installer (*install_sw4stm32_macos_64bits-v2.2.run*) to Desktop.

3. Open your ***Terminal*** on Mac.

4. Navigate into Desktop directory.

   $ cd Desktop

5. Change the permission of the installer file.

   $ chmod 755 install_sw4stm32_macos_64bits-v2.2.run

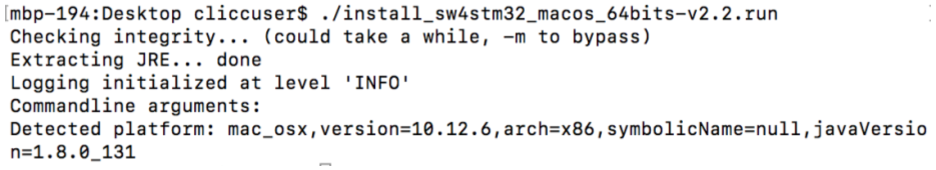
6. Disable the gate keeper restriction on your Mac.

   $ sudo spctl --master-disable

7. Install System WorkBench.

   $ ./ install_sw4stm32_macos_64bits-v2.2.run

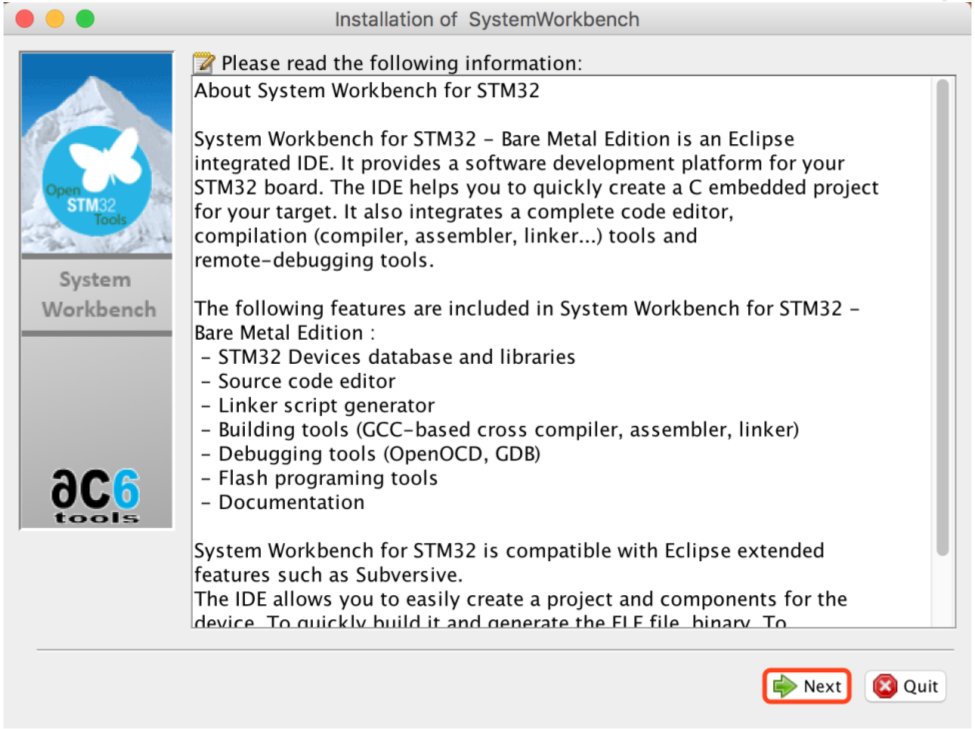
   Then, you should see the installation information in terminal like Figure 2.



```
[mbp-194:Desktop cliccuser$ ./install_sw4stm32_macos_64bits-v2.2.run
Checking integrity... (could take a while, -m to bypass)
Extracting JRE... done
Logging initialized at level 'INFO'
Commandline arguments:
Detected platform: mac_osx,version=10.12.6,arch=x86,symbolicName=null,javaVersio
n=1.8.0_131
```

Figure 2: System WorkBench Installation information in terminal.

8. Wait for a few seconds, you will see a GUI installation interface popup as Figure 3.



Please read the following information:

About System Workbench for STM32

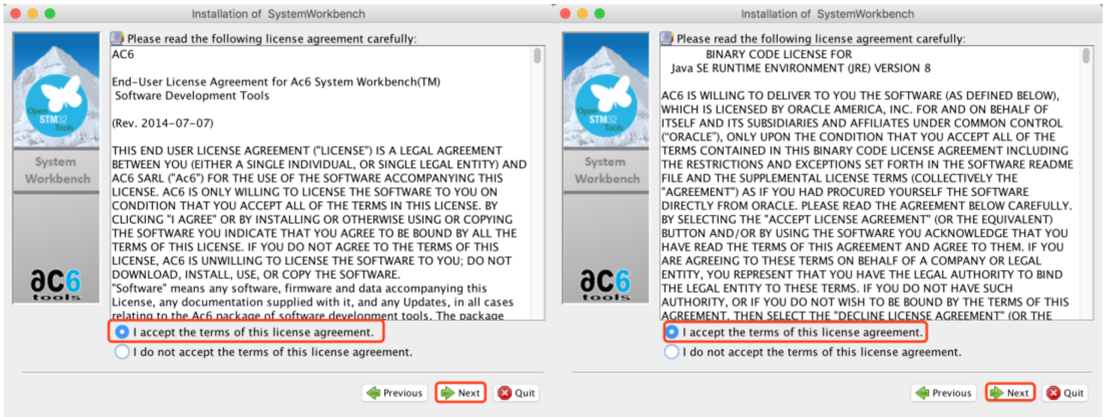
System Workbench for STM32 – Bare Metal Edition is an Eclipse integrated IDE. It provides a software development platform for your STM32 board. The IDE helps you to quickly create a C embedded project for your target. It also integrates a complete code editor, compilation (compiler, assembler, linker...) tools and remote-debugging tools.

The following features are included in System Workbench for STM32 – Bare Metal Edition :
- STM32 Devices database and libraries
- Source code editor
- Linker script generator
- Building tools (GCC-based cross compiler, assembler, linker)
- Debugging tools (OpenOCD, GDB)
- Flash programing tools
- Documentation

System Workbench for STM32 is compatible with Eclipse extended features such as Subversion.
The IDE allows you to easily create a project and components for the device. To quickly build it and generate the ELF file, binary. To

Figure 3: GUI installation interface of System WorkBench

9. Click on Next to continue. Accept all the license agreement and click next to continue as Figure 4.



Please read the following license agreement carefully:

AC6

End-User License Agreement for Ac6 System Workbench(TM)
Software Development Tools

(Rev. 2014-07-07)

THIS END USER LICENSE AGREEMENT ("LICENSE") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER A SINGLE INDIVIDUAL, OR SINGLE LEGAL ENTITY) AND AC6 SARL ("Ac6") FOR THE USE OF THE SOFTWARE ACCOMPANYING THIS LICENSE. AC6 IS ONLY WILLING TO LICENSE THE SOFTWARE TO YOU ON CONDITION THAT YOU ACCEPT ALL OF THE TERMS IN THIS LICENSE. BY CLICKING "I AGREE" OR BY INSTALLING OR OTHERWISE USING OR COPYING THE SOFTWARE YOU INDICATE THAT YOU AGREE TO BE BOUND BY ALL THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, AC6 IS UNWILLING TO LICENSE THE SOFTWARE TO YOU; DO NOT DOWNLOAD, INSTALL, USE, OR COPY THE SOFTWARE.
"Software" means any software, firmware and data accompanying this License, any documentation supplied with it, and any Updates, in all cases relating to the Ac6 package of software development tools. The package

● I accept the terms of this license agreement.
○ I do not accept the terms of this license agreement.

Please read the following license agreement carefully:

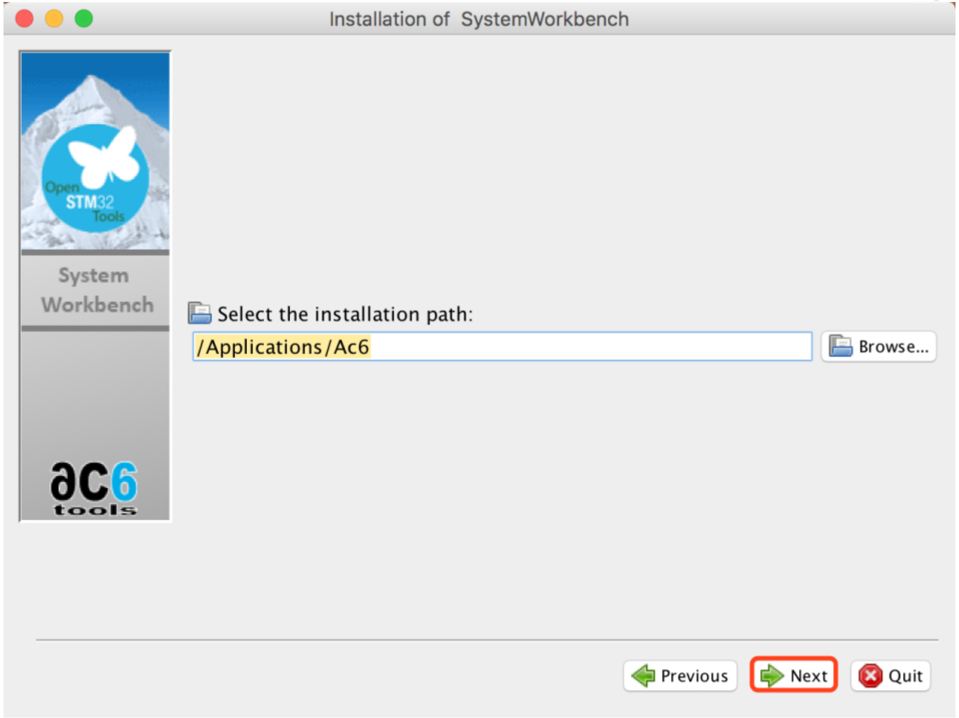
BINARY CODE LICENSE FOR
Java SE RUNTIME ENVIRONMENT (JRE) VERSION 8

AC6 IS WILLING TO DELIVER TO YOU THE SOFTWARE (AS DEFINED BELOW), WHICH IS LICENSED BY ORACLE AMERICA, INC. FOR AND ON BEHALF OF ITSELF AND ITS SUBSIDIARIES AND AFFILIATES UNDER COMMON CONTROL ("ORACLE"), ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT INCLUDING THE RESTRICTIONS AND EXCEPTIONS SET FORTH IN THE SOFTWARE README FILE AND THE SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY THE "AGREEMENT") AS IF YOU HAD PROCURED YOURSELF THE SOFTWARE DIRECTLY FROM ORACLE. PLEASE READ THE AGREEMENT BELOW CAREFULLY. BY SELECTING THE "ACCEPT LICENSE AGREEMENT" (OR THE EQUIVALENT) BUTTON AND/OR BY USING THE SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THE TERMS OF THIS AGREEMENT AND AGREE TO THEM. IF YOU ARE AGREEING TO THESE TERMS ON BEHALF OF A COMPANY OR LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE LEGAL AUTHORITY TO BIND THE LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO BE BOUND BY THE TERMS OF THIS AGREEMENT, THEN SELECT THE "DECLINE LICENSE AGREEMENT" (OR THE

● I accept the terms of this license agreement.
○ I do not accept the terms of this license agreement.

Figure 4: Agree the license agreement for System WorkBench.

10. Use its **default** installation directory and click next to continue. See Figure 5.



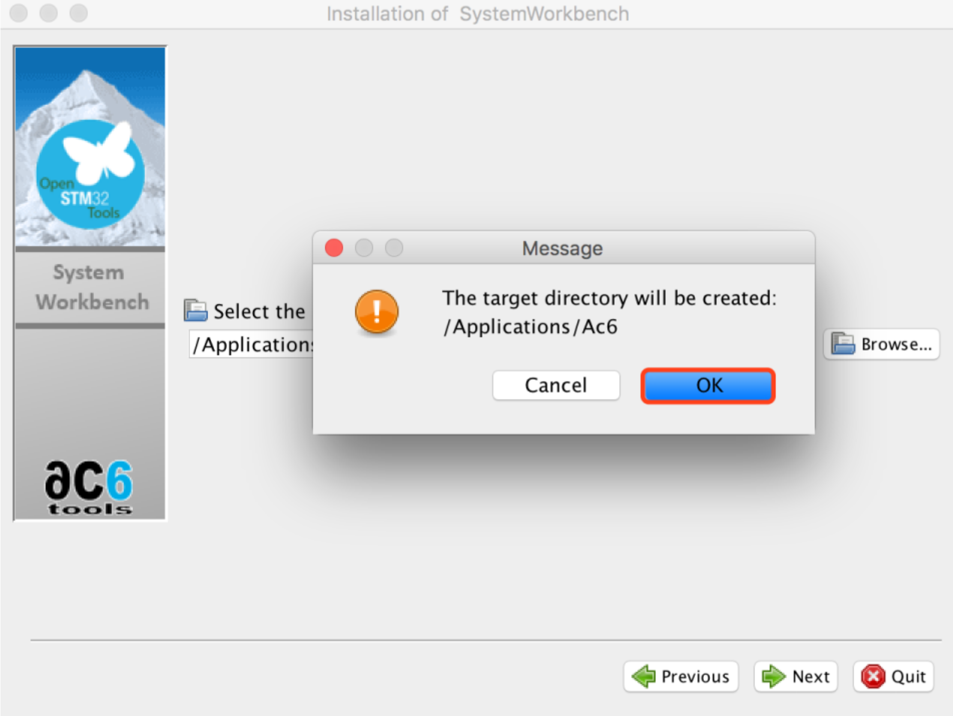
Installation of SystemWorkbench

Select the installation path:

/Applications/Ac6

Figure 5: Use default installation directory

11. Click OK in the pop-up window. See Figure 6.



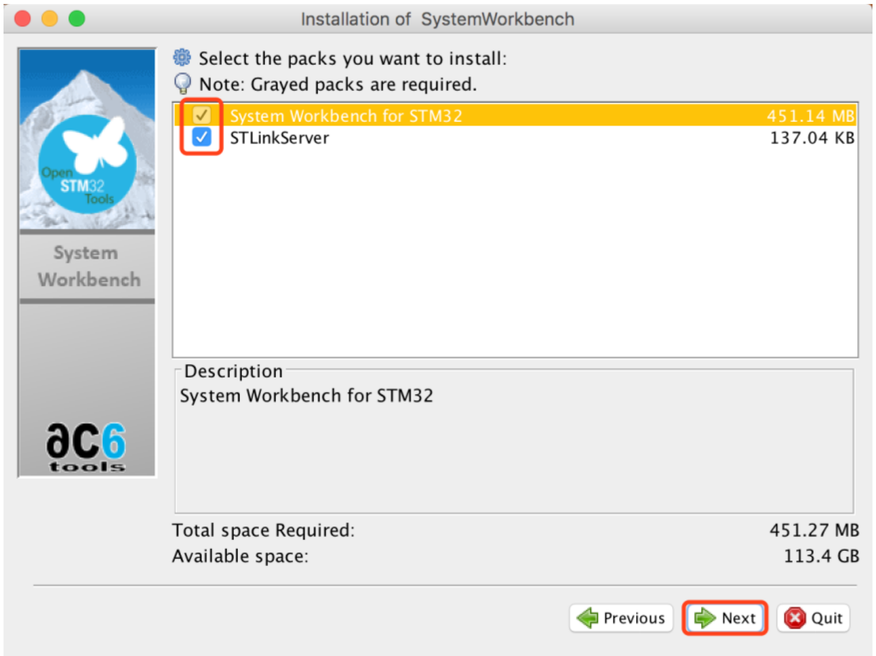
Installation of SystemWorkbench

Message

The target directory will be created:
/Applications/Ac6

Figure 6: Create the installation directory

12. Make sure that all installation options have been checked and click next to continue. See Figure 7.



Installation of SystemWorkbench

Select the packs you want to install:

Note: Grayed packs are required.

☑	System Workbench for STM32	451.14 MB
☑	STLinkServer	137.04 KB

Description

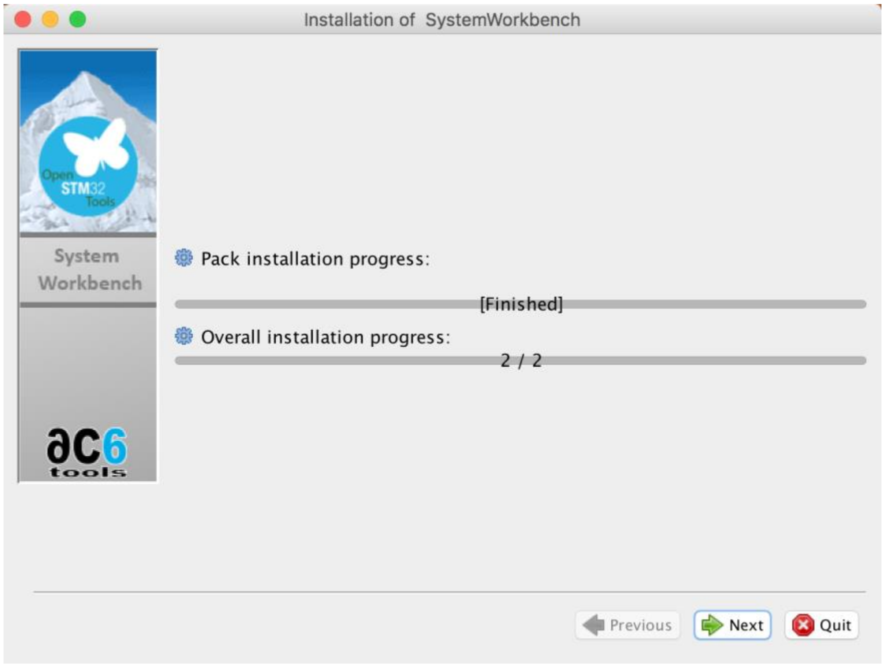
System Workbench for STM32

Total space Required: 451.27 MB

Available space: 113.4 GB

Figure 7: Install everything in the package

13. Wait until the installation has been finished. See Figure 8. ***IF THE INSTALLATION IS STUCK at 2/2, CHECK YOUR TERMINAL AND ENTER YOUR MAC PASSWORD IF IT REQUIRES.***



Installation of SystemWorkbench

Pack installation progress:

[Finished]

Overall installation progress:

2 / 2

Figure 8: Installation has been finished

14. Open your *Finder* and find the *System WorkBench* according to Figure 9 (*Applications-> Ac6 -> SystemWorkBench*).

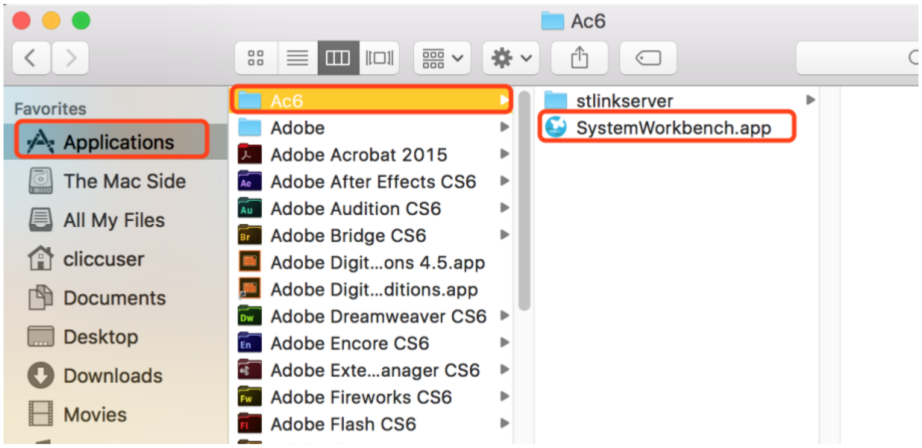
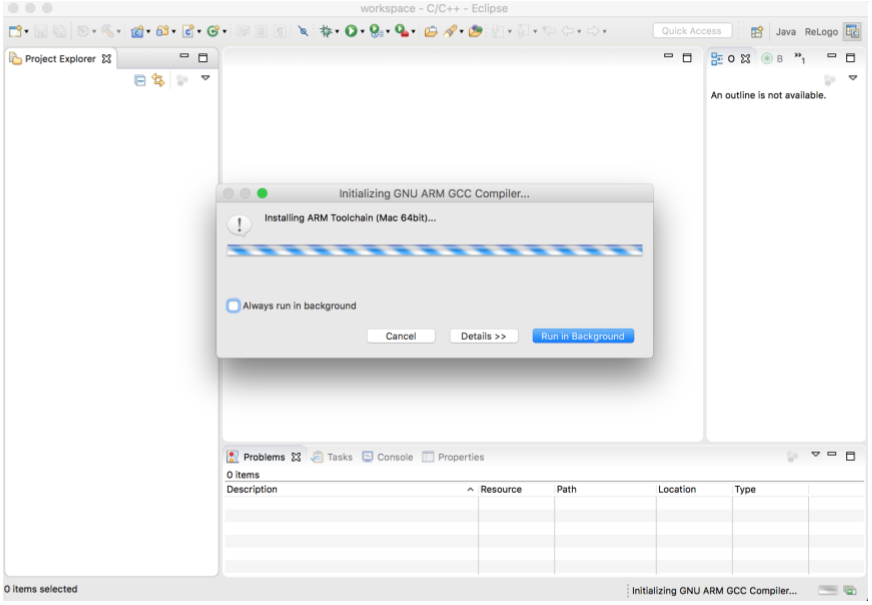


Figure 9: System WorkBench application location

15. Make sure your computer is connected to internet and double-click SystemWorkbench.app. Use the default workspace directory and wait for the final step of installation as Figure 10.



Initializing GNU ARM GCC Compiler...

Installing ARM Toolchain (Mac 64bit)...

□ Always run in background

0 items

Initializing GNU ARM GCC Compiler...

Figure 10: Last step of System Workbench (IDE) installation

16. After step 15, you System Workbench (IDE) is ready to go. System Workbench is an Eclipse with proper configuration and compiler tools.

4. ST-Link Utilities for STM32

This section will guide users through the process of downloading and installing a flash tool.

1. Open a terminal on your Mac. To do this, open spotlight search by clicking the magnifying glass in the top right-hand corner of your Mac toolbar (see Figure), and search for “Terminal”. Double click the result highlighted in the red box in Figure .

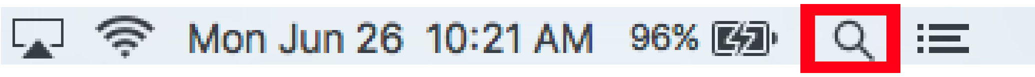


Figure 11: Opening Spotlight search on Mac.

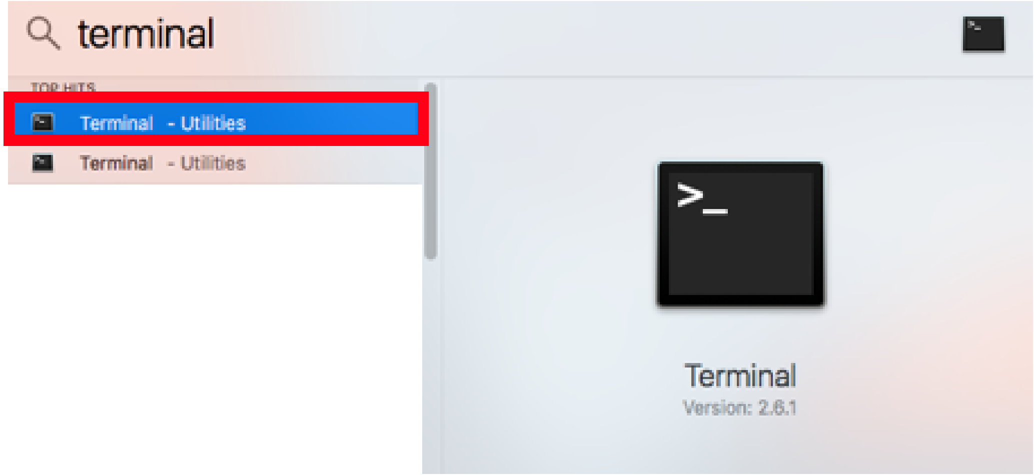
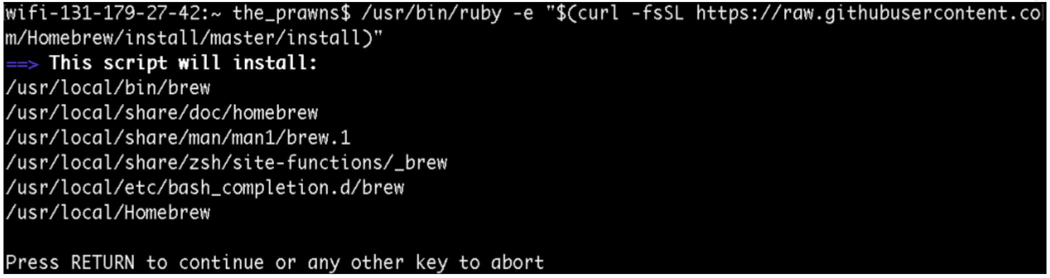


Figure 12: How to open the "Terminal" application on a Mac.

2. Follow the instructions online at https://brew.sh/ to install Homebrew (a package manager). Alternatively, issue the command shown below in the terminal session from step 1. Follow the onscreen prompts.

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Note that the above command appears as one line.



```
wifi-131-179-27-42:~ the_prawns$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.co
m/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew

Press RETURN to continue or any other key to abort
```

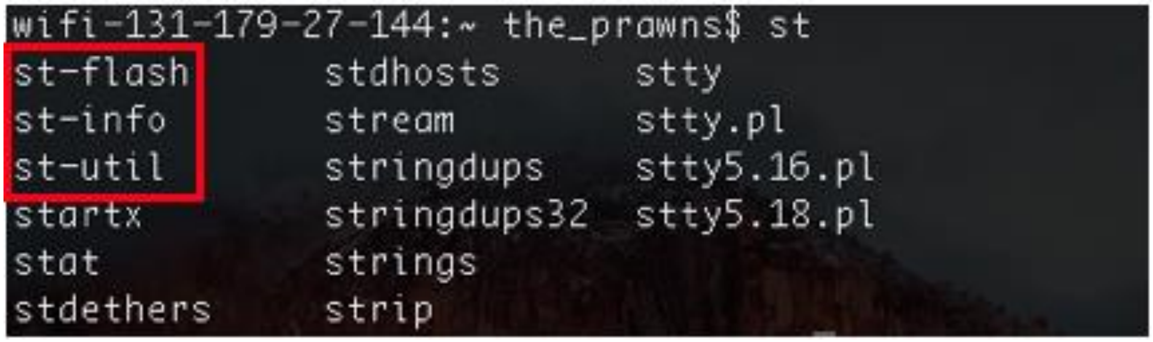
Figure 13: Installing Homebrew.

3. Issue the following command in the terminal session from step 1.

```
$ brew install stlink
```

4. To check for successful installation, type **st** into the terminal session from step 1 **without pressing [Enter]**.

5. While the prompt displays **st**, Press **[Tab]** until the terminal session lists some programs through the auto-complete feature. If the terminal session displays the following items (highlighted in the red box in Figure), the installation completed successfully.



```
wifi-131-179-27-144:~ the_prawns$ st
st-flash        stdhosts        stty
st-info         stream          stty.pl
st-util         stringdups      stty5.16.pl
startx          stringdups32    stty5.18.pl
stat            strings
stdethers       strip
```

Figure 14: Three st tools should appear when using the autocomplete feature.

5. Example Data Logging Project

5.1. Download STSW-STLKT01 v1.2.0

Follow the instructions below to download an existing data-logging project for the SensorTile.

First, please note that this Tutorial supported Version 1.2.0 of the SensorTile Firmware, STSW-STLKT01

Please download the Version 1.2.0 by clicking on this link or pasting this URL into your browser. When prompted, fill out the form and a download link will be sent to your email.

http://www.st.com/content/st_com/en/premium-content/sensortile-curriculum-stsw-stlkt01_zip.html

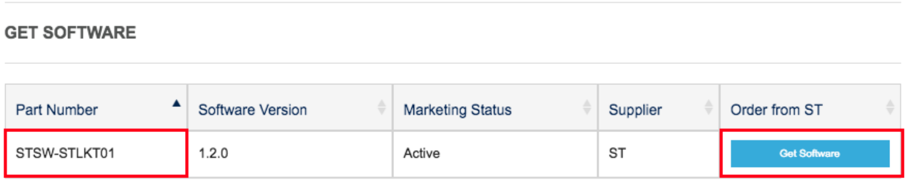
---

Instructions to download firmware from STMicroelectronics websites.

1. Open the following link on a web-browser on your Mac.

   http://www.st.com/content/st_com/en/products/embedded-software/evaluation-tool-software/stsw-stlkt01.html

2. Scroll to the bottom of the page, and click "Get Software" for the entry **STSW-STLKT01**.



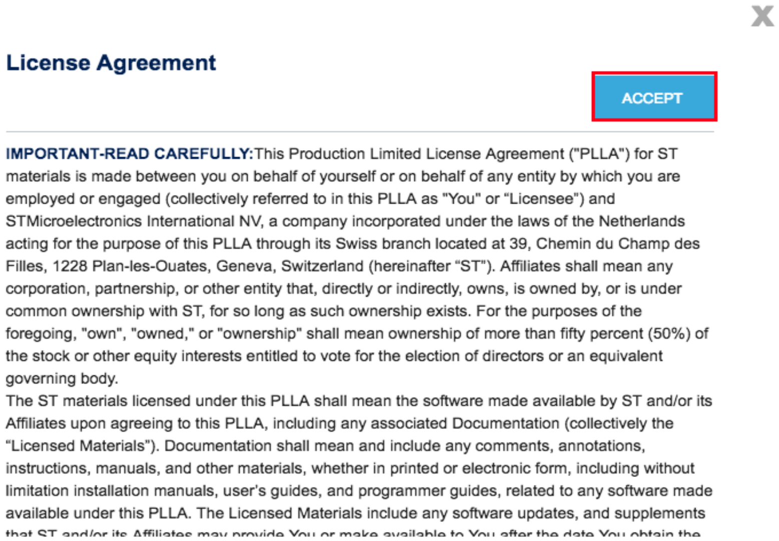
GET SOFTWARE

Part Number	Software Version	Marketing Status	Supplier	Order from ST
STSW-STLKT01	1.2.0	Active	ST	Get Software

Figure 15: This figure depicts what to click in order to acquire the example project files.

3. When the license agreement appears (Figure), read through it.

After reading the agreement, click “Accept” at the top of the page.



License Agreement

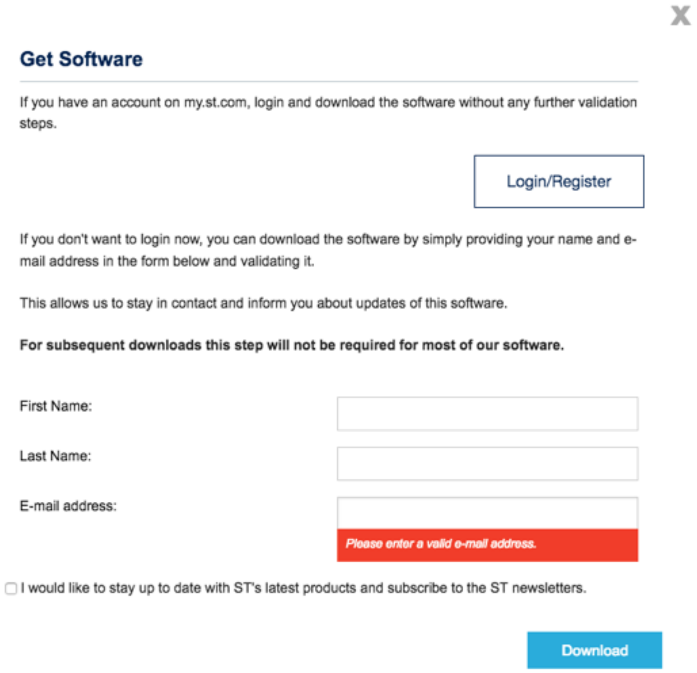
IMPORTANT-READ CAREFULLY: This Production Limited License Agreement ("PLLA") for ST materials is made between you on behalf of yourself or on behalf of any entity by which you are employed or engaged (collectively referred to in this PLLA as "You" or “Licensee") and STMicroelectronics International NV, a company incorporated under the laws of the Netherlands acting for the purpose of this PLLA through its Swiss branch located at 39, Chemin du Champ des Filles, 1228 Plan-les-Ouates, Geneva, Switzerland (hereinafter “ST”). Affiliates shall mean any corporation, partnership, or other entity that, directly or indirectly, owns, is owned by, or is under common ownership with ST, for so long as such ownership exists. For the purposes of the foregoing, "own", "owned," or "ownership" shall mean ownership of more than fifty percent (50%) of the stock or other equity interests entitled to vote for the election of directors or an equivalent governing body.

The ST materials licensed under this PLLA shall mean the software made available by ST and/or its Affiliates upon agreeing to this PLLA, including any associated Documentation (collectively the “Licensed Materials”). Documentation shall mean and include any comments, annotations, instructions, manuals, and other materials, whether in printed or electronic form, including without limitation installation manuals, user's guides, and programmer guides, related to any software made available under this PLLA. The Licensed Materials include any software updates, and supplements that ST and/or its Affiliates may provide You or make available to You after the date You obtain the

Figure 16: License agreement that appears once you click "Get Software".

4. Either register an account, or enter your name and email address when prompted by the popup depicted in Figure .

Then, click “Download” at the bottom of the popup screen.



Get Software

If you have an account on my.st.com, login and download the software without any further validation steps.

If you don't want to login now, you can download the software by simply providing your name and e-mail address in the form below and validating it.

This allows us to stay in contact and inform you about updates of this software.

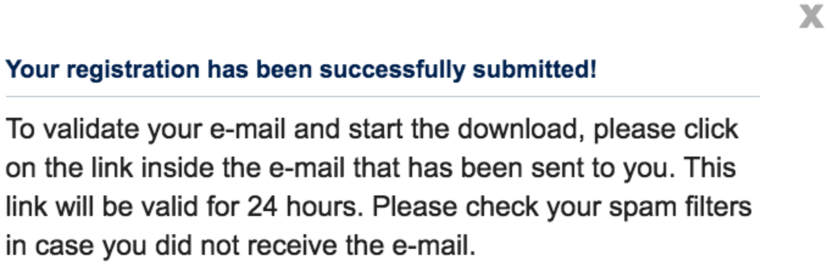
For subsequent downloads this step will not be required for most of our software.

First Name:	
Last Name:	
E-mail address:	*Please enter a valid e-mail address.*

□ I would like to stay up to date with ST's latest products and subscribe to the ST newsletters.

Figure 17: Dialog box that collects developer information prior to enabling acquisition of example project files.

5. The popup box should resemble Figure if all steps have proceeded successfully.



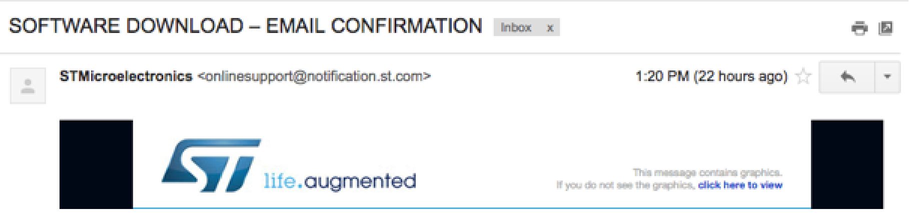
Your registration has been successfully submitted!

To validate your e-mail and start the download, please click on the link inside the e-mail that has been sent to you. This link will be valid for 24 hours. Please check your spam filters in case you did not receive the e-mail.

Figure 18: This is a successful registration notification box.

6. Check your email inbox and spam folders for the email. Follow the link that you were sent to download the example code. The email should resemble Figure .

Note: this link will not function if your web-browser is in “Incognito” or “Private” mode.



SOFTWARE DOWNLOAD – EMAIL CONFIRMATION Inbox x

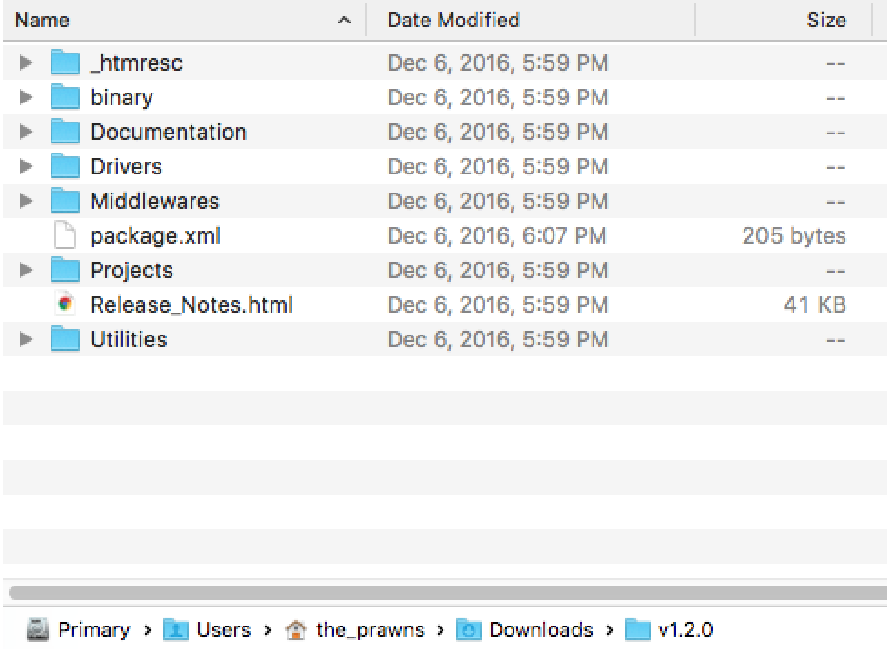
STMicroelectronics <onlinesupport@notification.st.com> 1:20 PM (22 hours ago)

life.augmented
This message contains graphics.
If you do not see the graphics, click here to view

Figure 19: Email from STMicroelectronics containing weblink to enable access to download example project.

7. Extract the contents of the archive. The default output folder name should be **v1.2.0**. For consistency with this tutorial, please extract the archive into the Downloads directory.

   The full path of the directory should resemble "/Users/<username>/Downloads/v1.2.0".



Name	Date Modified	Size
_htmresc	Dec 6, 2016, 5:59 PM	--
binary	Dec 6, 2016, 5:59 PM	--
Documentation	Dec 6, 2016, 5:59 PM	--
Drivers	Dec 6, 2016, 5:59 PM	--
Middlewares	Dec 6, 2016, 5:59 PM	--
package.xml	Dec 6, 2016, 6:07 PM	205 bytes
Projects	Dec 6, 2016, 5:59 PM	--
Release_Notes.html	Dec 6, 2016, 5:59 PM	41 KB
Utilities	Dec 6, 2016, 5:59 PM	--

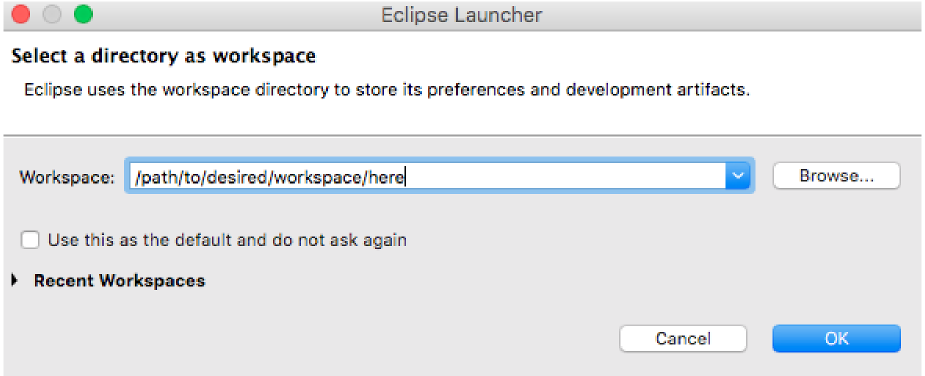
Figure 20: Contents of folder "v1.2.0".

5.2. Import

Follow the instructions below in order to import the example project into the System WorkBench IDE. Note: while we call this System WorkBench, it will be listed as "Eclipse" in your applications.

1. Open the System WorkBench application.

2. Use the default workspace directory, and click "Ok". See Figure 21.



Eclipse Launcher

Select a directory as workspace

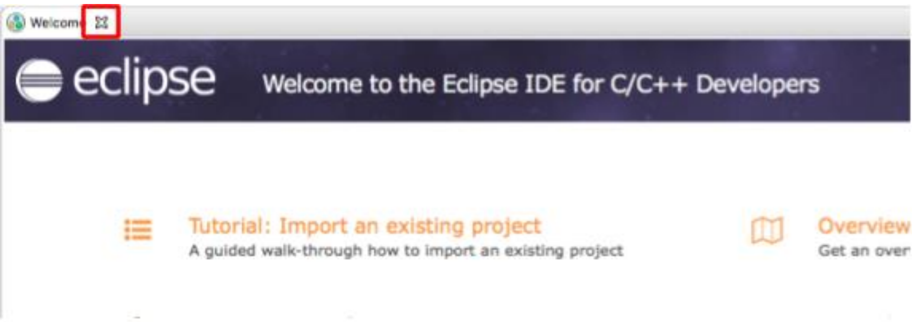
Eclipse uses the workspace directory to store its preferences and development artifacts.

Workspace: /path/to/desired/workspace/here

☐ Use this as the default and do not ask again

Figure 21: Selecting your workspace directory. It is recommended that you use the default location. It will resemble the form "/Users/<username_here>/Documents/workspace".

3. If you see a welcome screen, close it by clicking the "x" in the top left corner. See Figure .



Welcom
eclipse
Welcome to the Eclipse IDE for C/C++ Developers
Tutorial: Import an existing project
A guided walk-through how to import an existing project
Overview
Get an over

Figure 22: Closing the welcome splash page.

4. Once you are on the main screen of System WorkBench, click "File > Import …" to begin importing the project. See Figure .

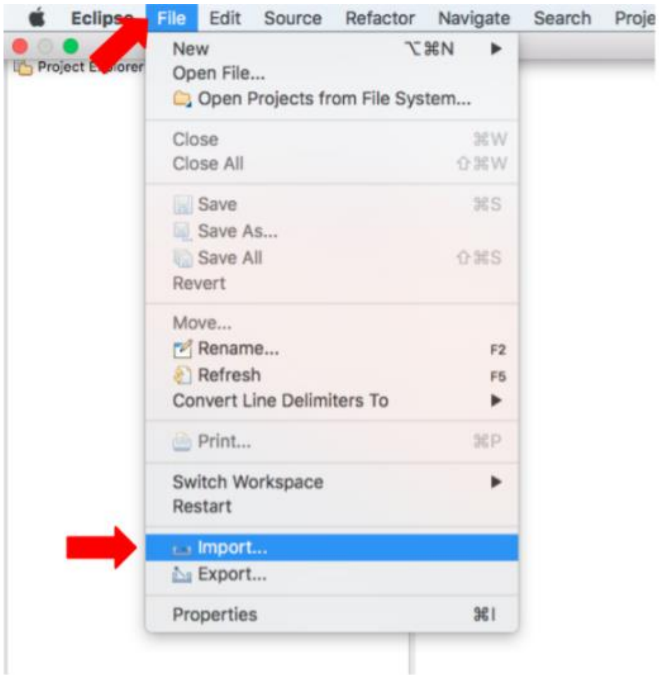
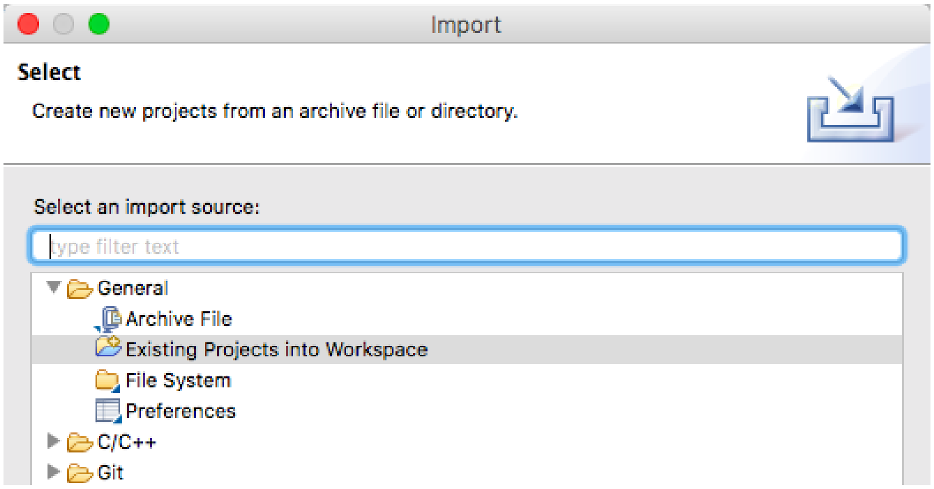


Figure 23: Importing an existing project into the workspace.

5. When the new window popup appears, click double click the entry “General”. See Figure .

6. Double click the “Existing Projects into Workspace” option that appears under “General”. See Figure .



Import

Select

Create new projects from an archive file or directory.

Select an import source:

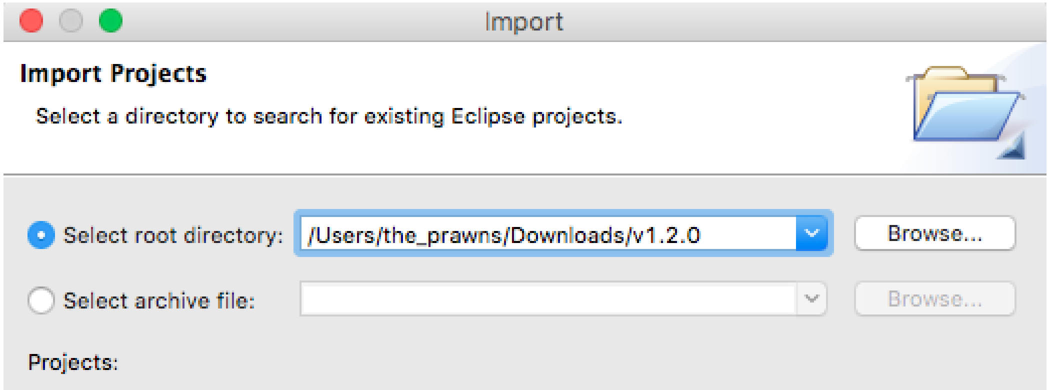
type filter text

Figure 24: Instructions on the "Select" dialog box.

7. **<u>DO NOT CLICK FINISH UNTIL YOU COMPLETELY READ THROUGH STEP 8</u>**.

   There are three projects in the v1.2.0 folder. If you import more than one project at a time, compilation will fail. This is because the three projects are dependent on other files in the folder.

   On the new popup window, enter the path to the folder **v1.2.0** in the text field "Select root directory". **<u>DO NOT CLICK FINISH</u>**.



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: /Users/the_prawns/Downloads/v1.2.0 [Browse...]

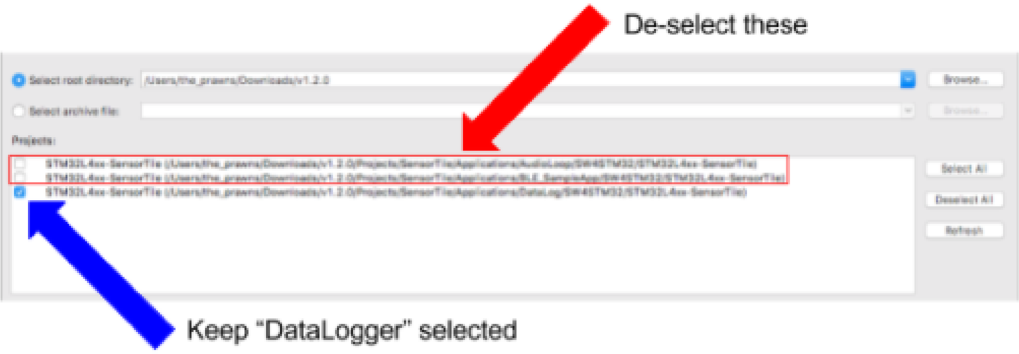
○ Select archive file: [Browse...]

Projects:

Figure 25: Selecting the folder where the existing project was extracted.

8. The projects box should be populated with 3 projects.
   8.1. AudioLoop (DE-SELECT THIS).
   8.2. BLE_SampleApp (DE-SELECT THIS).
   8.3. DataLog (Keep this selected).

   Uncheck the boxes next to AudioLoop and BLE_SampleApp.



De-select these

● Select root directory: /Users/the_prawns/Downloads/v1.2.0 [Browse...]

○ Select archive file: [Browse...]

Projects:

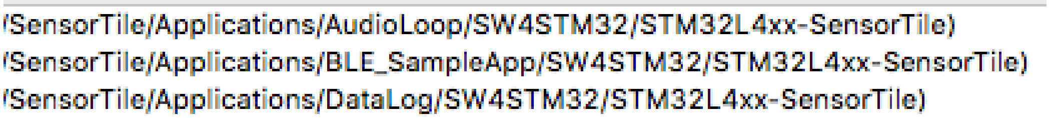
□ STM32L4xx-SensorTile (/Users/the_prawns/Downloads/v1.2.0/Projects/SensorTile/Applications/AudioLoop/SW4STM32/STM32L4xx-SensorTile)
□ STM32L4xx-SensorTile (/Users/the_prawns/Downloads/v1.2.0/Projects/SensorTile/Applications/BLE_SampleApp/SW4STM32/STM32L4xx-SensorTile)
☑ STM32L4xx-SensorTile (/Users/the_prawns/Downloads/v1.2.0/Projects/SensorTile/Applications/DataLog/SW4STM32/STM32L4xx-SensorTile)

[Select All] [Deselect All] [Refresh]

Keep "DataLogger" selected

Figure 26: Only selecting one project from the existing files to import into the workspace.

If you have selected more than one project and clicked "Finish", you will need to open your workspace directory on the Finder application on your Mac and manually delete all of the files.

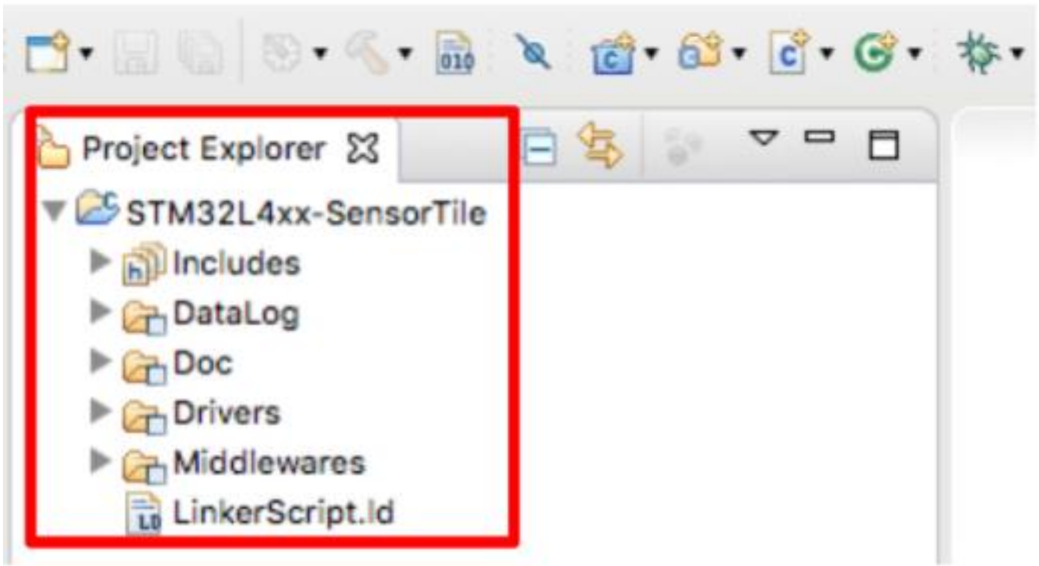


/SensorTile/Applications/AudioLoop/SW4STM32/STM32L4xx-SensorTile)
/SensorTile/Applications/BLE_SampleApp/SW4STM32/STM32L4xx-SensorTile)
/SensorTile/Applications/DataLog/SW4STM32/STM32L4xx-SensorTile)

Figure 27: Close-up image of text from Figure.

Click “Finish” with **only one project** selected.

9. Your screen should resemble the screenshot depicted below.



Project Explorer

- STM32L4xx-SensorTile
  - Includes
  - DataLog
  - Doc
  - Drivers
  - Middlewares
  - LinkerScript.ld

Figure 28: Contents of "Project Explorer" after successfully importing the example project files.

5.3. Build

This portion of the document will guide users through the process of compiling the C-source code into a binary file that can be loaded onto the SensorTile board.

1. Click Project -> Build Project (see Figure).

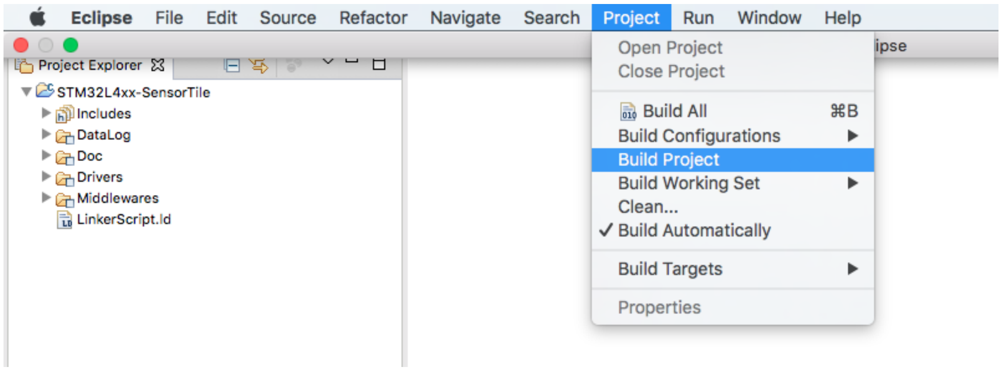


Figure 29: Compiling the source code for the example project.

2. If step 1 completed successfully, a folder named "Release" should appear and contain a **.bin** file. See Figure .

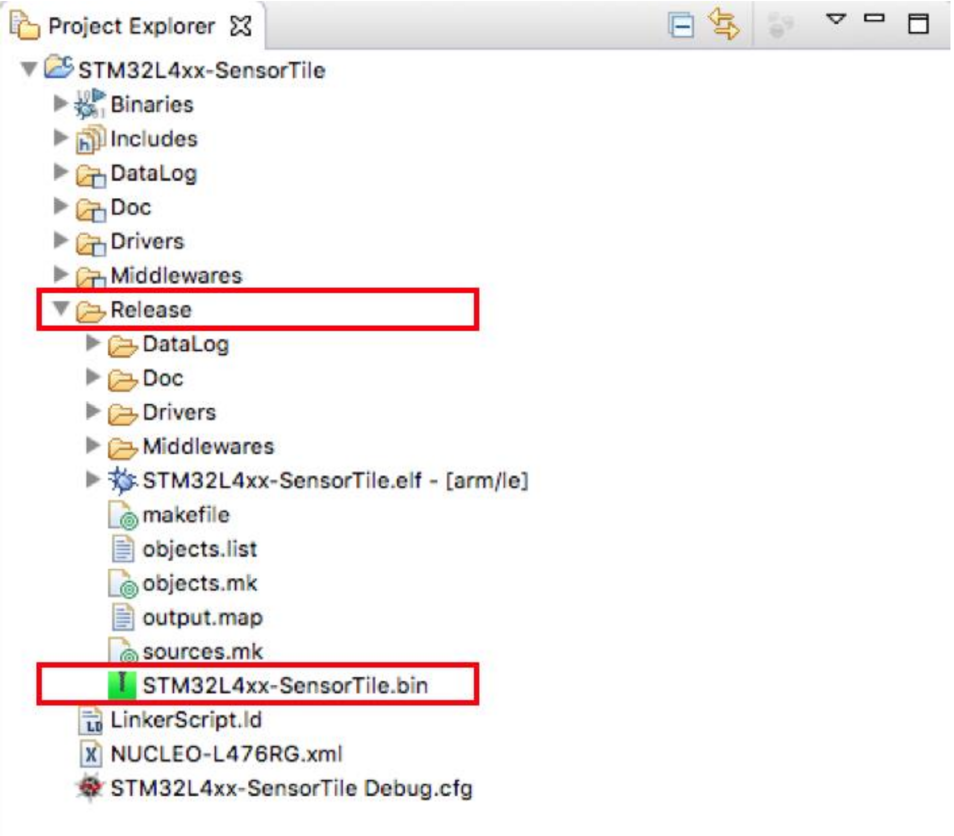


Figure 30: Project directory contents after successful compilation of project source code.

5.4.SensorTile Hardware Platform

This section describes how to configure the hardware. Be very careful in this section, if the wire connections are not configured correctly, the boards could be permanently damaged. Do not proceed with the tutorial until an instructor has verified that your board is correctly configured.

1. Remove the Nucleo - L476RG board from its packaging.

2. Remove the CN2 Jumpers. (Remove both of them). See Figure .

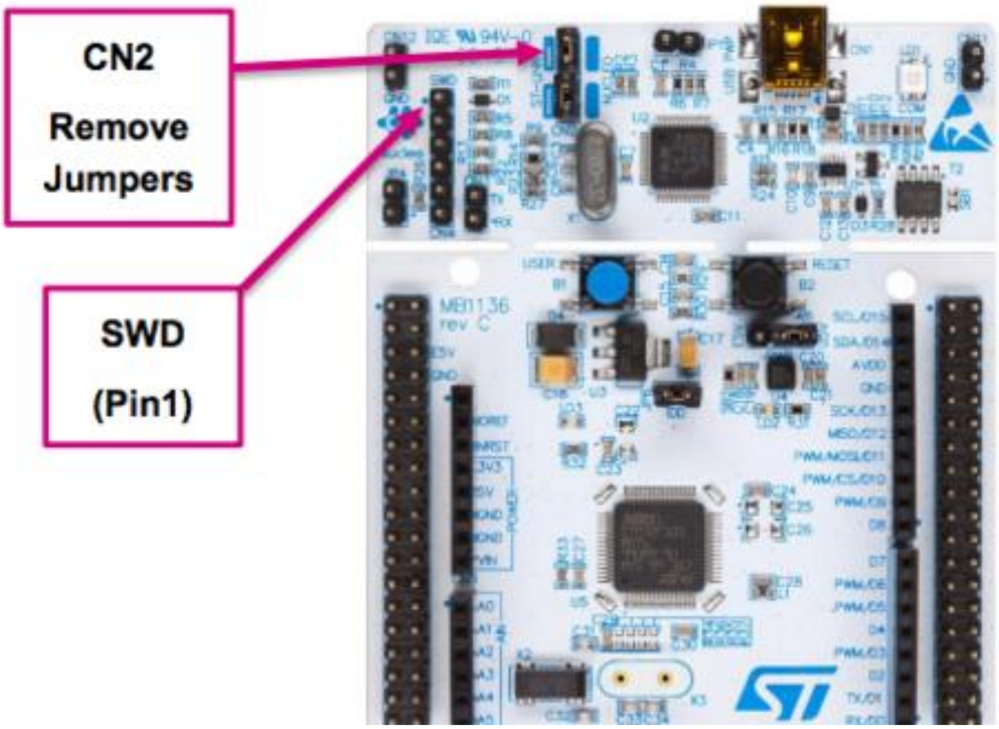


Figure 31: Removing the CN2 Jumpers from the Nucleo-L47RG board.

3. Place the SensorTile on the larger evaluation board. Ensure the orientation of the SensorTile matches Figure .

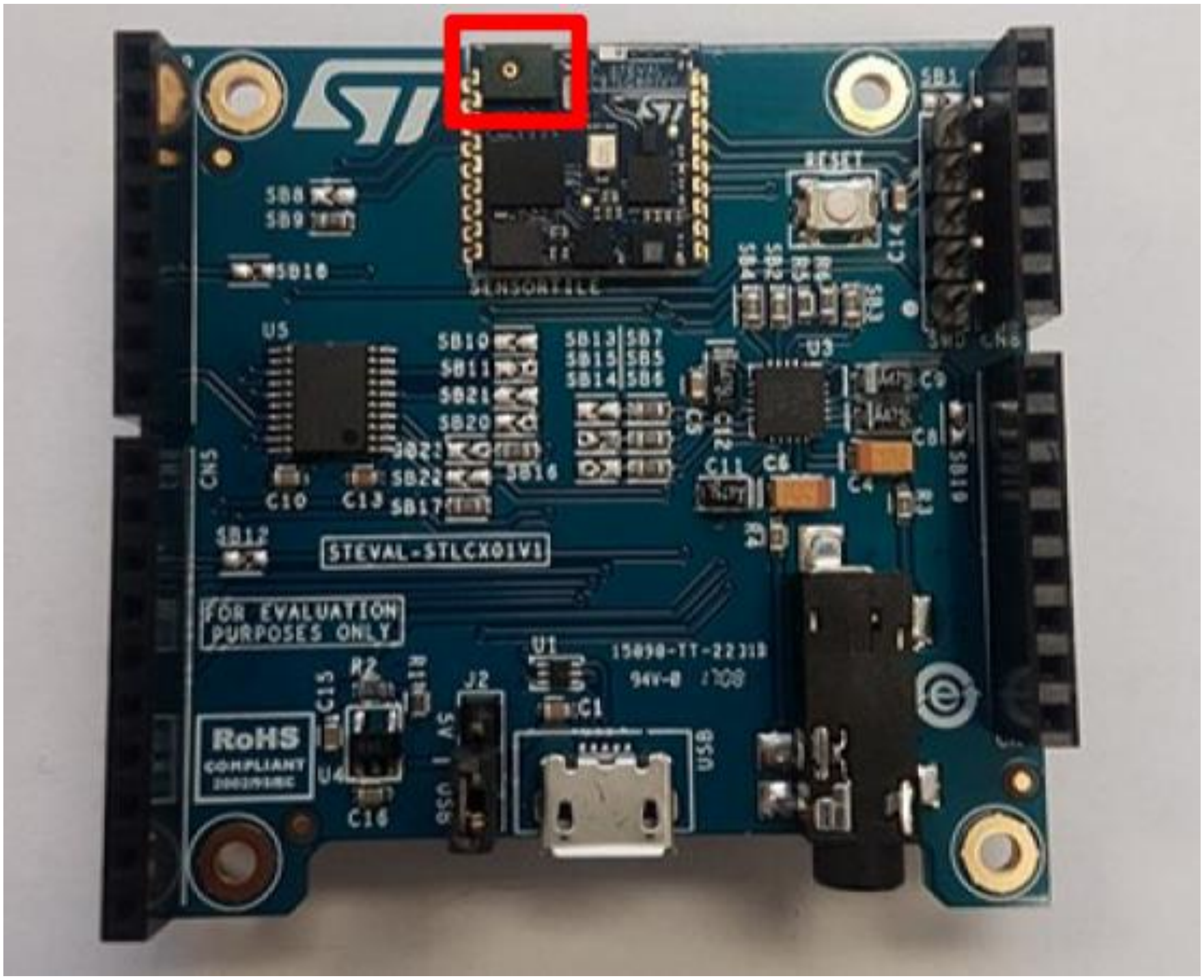


Figure 32: Ensure the orientation of the SensorTile on the larger evaluation board matches this figure. There should be a green protrusion with a little metalic hole right next to the ST logo.

4. Connect the boards by attaching an SWD connector from the Nucleo board to the SensorTile board. Be **very** careful in this step. Please examine the figures Figure - Figure . Do not proceed with this tutorial until you have verified with an instructor that the hardware is correctly configured.

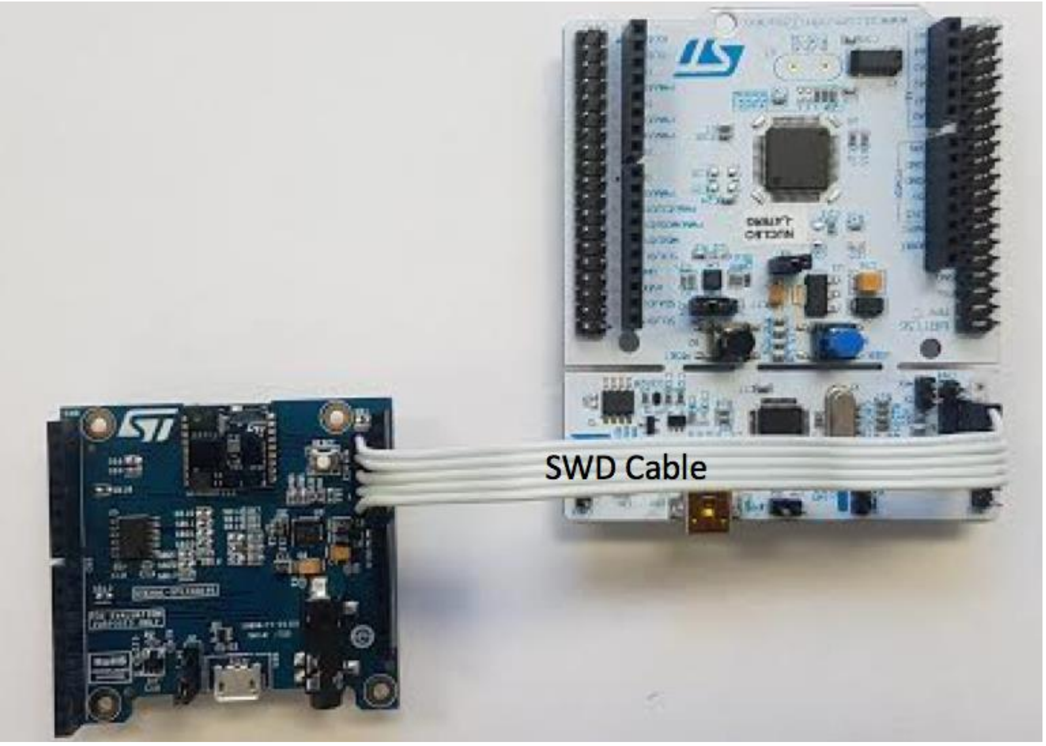
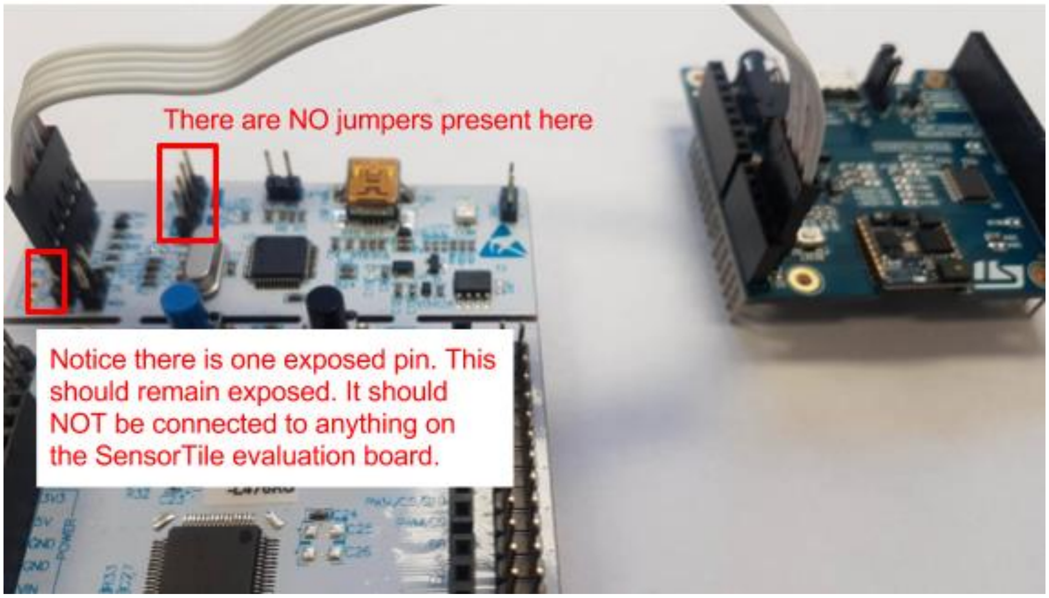


Figure 33: First figure depicting correct hardware configuration. Ensure that the SWD cable is oriented such that the pins marked "SWD" are connected via the same wire.



Notice there is one exposed pin. This should remain exposed. It should NOT be connected to anything on the SensorTile evaluation board.

Figure 34: Second figure depicting correct hardware configuration.

5.5. Debug

This section will guide users through the process of running the **DataLog** application in debug mode on the SensorTile board.

1. Attach a mini-USB cable from the **Nucleo** to your Mac. Attach a micro-USB cable from the **SensorTile** to your Mac. See Figure .

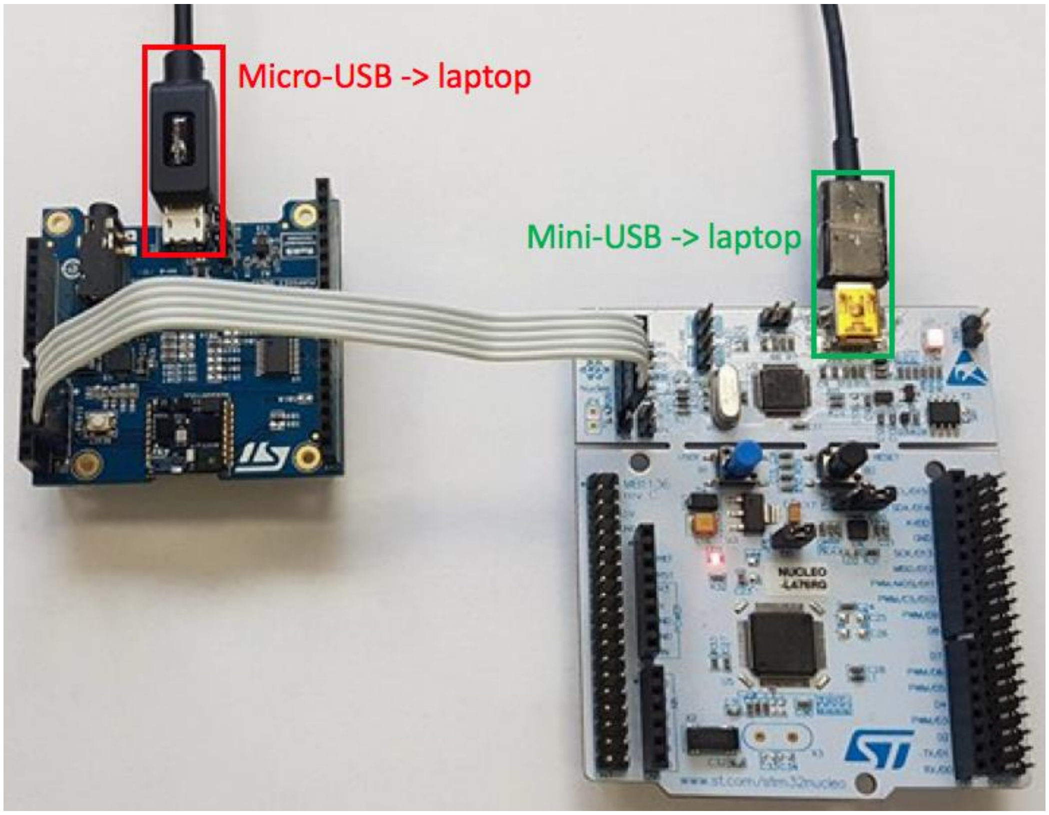
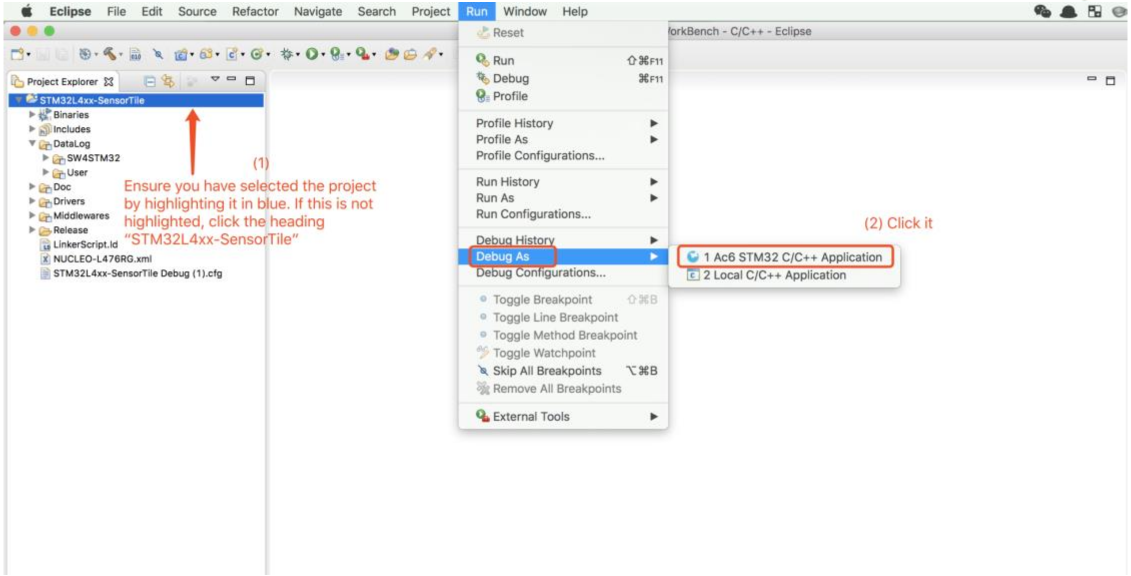


Figure 35: Establish a USB-wireline connection between each board and your Mac.

2. Click “Run -> Debug As -> AC6 STM32 C/C++ Application”. See Figure .



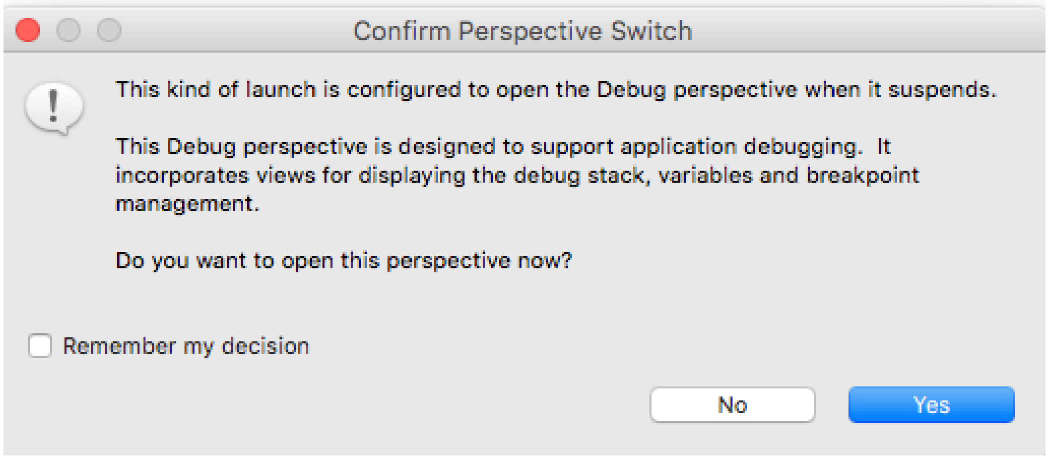
(1)

Ensure you have selected the project by highlighting it in blue. If this is not highlighted, click the heading "STM32L4xx-SensorTile"

(2) Click it

Figure 36: Launching the DataLog program in debug mode.

3. A dialog box may appear. Click "Yes" on this dialog box.



Confirm Perspective Switch

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

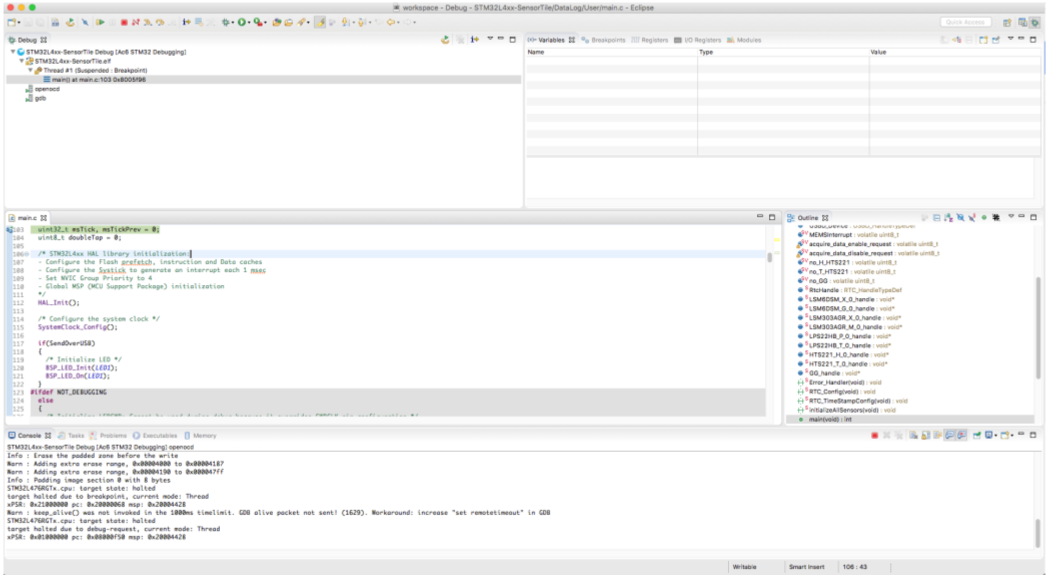
This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

☐ Remember my decision

Figure 5: Click "Yes" on this dialog box.

4. A debugging interface should appear Figure .

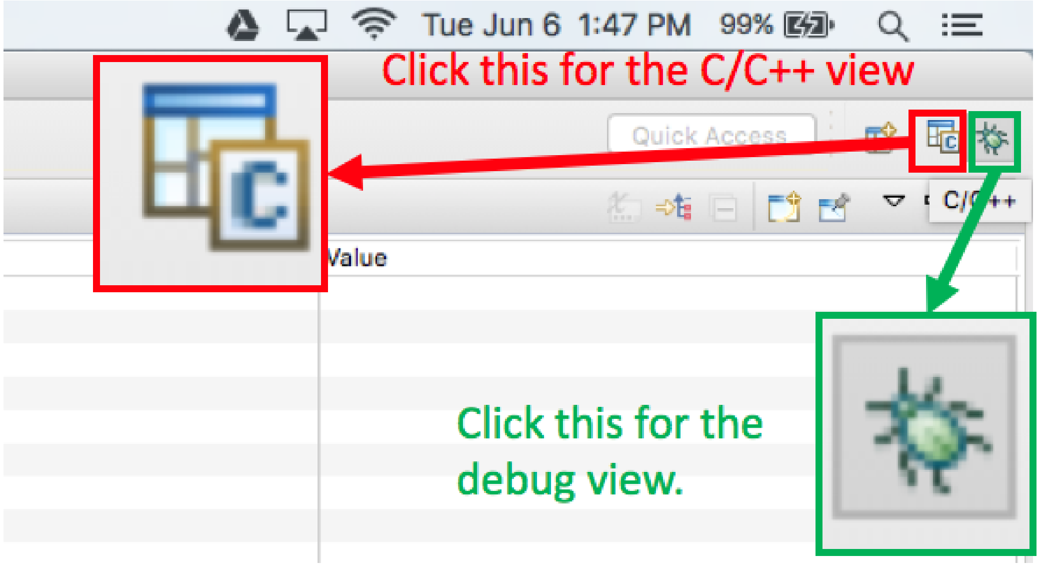

workspace - Debug - STM32L4xx-SensorTile/DataLog/User/main.c - Eclipse
Debug
STM32L4xx-SensorTile Debug [Ac6 STM32 Debugging]
STM32L4xx-SensorTile.elf
Thread #1 (Suspended : Breakpoint)
main() at main.c:103 0x8005f98
openocd
gdb
Variables
Breakpoints
Registers
I/O Registers
Modules
Name
Type
Value
main.c
Outline
MEMSInterrupt : volatile uint8_t
acquire_data_enable_request : volatile uint8_t
acquire_data_disable_request : volatile uint8_t
no_H_HTS221 : volatile uint8_t
no_T_HTS221 : volatile uint8_t
no_GG : volatile uint8_t
RtcHandle : RTC_HandleTypeDef
LSM6DSM_X_0_handle : void*
LSM6DSM_G_0_handle : void*
LSM303AGR_X_0_handle : void*
LSM303AGR_M_0_handle : void*
LPS22HB_P_0_handle : void*
LPS22HB_T_0_handle : void*
HTS221_H_0_handle : void*
HTS221_T_0_handle : void*
GG_handle : void*
Error_Handler(void) : void
RTC_Config(void) : void
RTC_TimeStampConfig(void) : void
initializeAllSensors(void) : void
main(void) : int
Console
Tasks
Problems
Executables
Memory
Writable
Smart Insert
106 : 43

```
  uint32_t msTick, msTickPrev = 0;
  uint8_t doubleTap = 0;

  /* STM32L4xx HAL library initialization:
  - Configure the Flash prefetch, instruction and Data caches
  - Configure the Systick to generate an interrupt each 1 msec
  - Set NVIC Group Priority to 4
  - Global MSP (MCU Support Package) initialization
  */
  HAL_Init();

  /* Configure the system clock */
  SystemClock_Config();

  if(SendOverUSB)
  {
    /* Initialize LED */
    BSP_LED_Init(LED1);
    BSP_LED_On(LED1);
  }
#ifdef NOT_DEBUGGING
  else
  {
```

```
STM32L4xx-SensorTile Debug [Ac6 STM32 Debugging] openocd
Info : Erase the padded zone before the write
Warn : Adding extra erase range, 0x00004000 to 0x00004187
Warn : Adding extra erase range, 0x00004190 to 0x000047ff
Info : Padding image section 0 with 8 bytes
STM32L476RGTx.cpu: target state: halted
target halted due to breakpoint, current mode: Thread
xPSR: 0x21000000 pc: 0x20000068 msp: 0x20004428
Warn : keep_alive() was not invoked in the 1000ms timelimit. GDB alive packet not sent! (1629). Workaround: increase "set remotetimeout" in GDB
STM32L476RGTx.cpu: target state: halted
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08000f50 msp: 0x20004428
```

Figure 38: Debugging interface.

If you wish to switch back to the C/C++ view, click the top right corner button labeled "C/C++". See Figure .



Click this for the C/C++ view

Click this for the debug view.

Figure 39: To enter the "C/C++" view, click the button highlighted in the red box. To enter the "debug mode" view, click the button highlighted in the green box.

5. Examine your SensorTile device. Notice how none of the LED's are activated.

6. Press the "green arrow with the yellow bar on its left" button that indicates "Resume" when you mouse over it. See Figure .

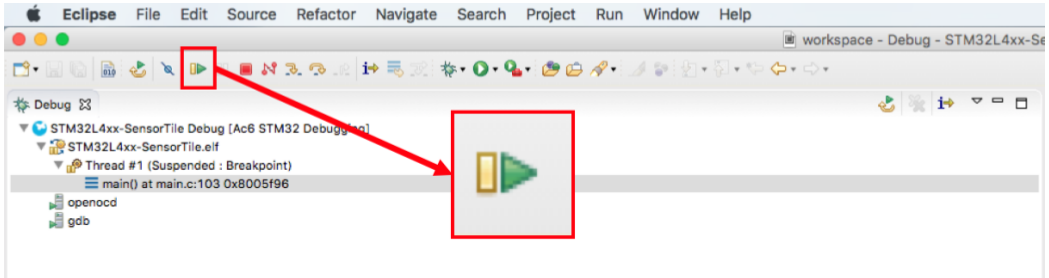
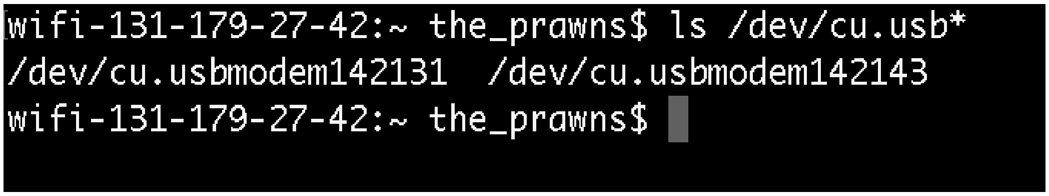

Eclipse
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
workspace - Debug - STM32L4xx-Se
Debug
STM32L4xx-SensorTile Debug [Ac6 STM32 Debugging]
STM32L4xx-SensorTile.elf
Thread #1 (Suspended : Breakpoint)
main() at main.c:103 0x8005f96
openocd
gdb

Figure 40: Starting the application in debug mode.

7. Examine your SensorTile device. Notice how the orange LED is rapidly blinking.

8. Open a terminal and issue the command shown in Figure .

```
$ ls /dev/cu.usb*
```



```
wifi-131-179-27-42:~ the_prawns$ ls /dev/cu.usb*
/dev/cu.usbmodem142131  /dev/cu.usbmodem142143
wifi-131-179-27-42:~ the_prawns$
```

Figure 41: Inspecting the devices connected via USB.

Notice how there are two devices listed.

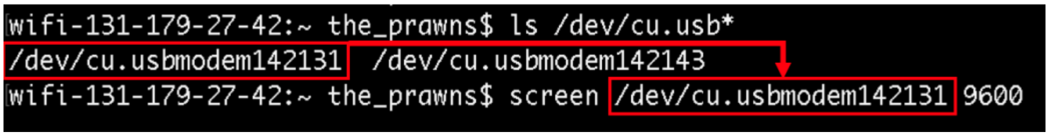
The device ending in “3” is the Nucleo board.

The device ending in “1” is the SensorTile.

9. **<u>If you receive an error in this step, please proceed to step 10.</u>**

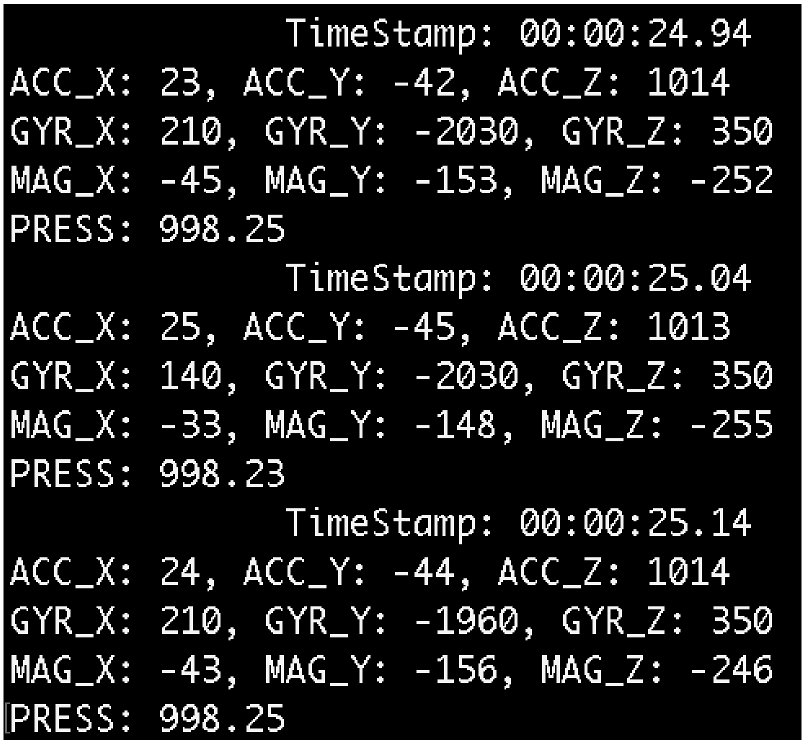
Inspect the data the SensorTile is sending over serial wireline USB connection by issuing the command shown in Figure . Make sure this command is edited to suit the name of the device as it appears in your terminal session. The output should resemble Figure .

```
$ screen /dev/cu.usbmodem<SENSORTILE_ID> 9600
```



```
wifi-131-179-27-42:~ the_prawns$ ls /dev/cu.usb*
/dev/cu.usbmodem142131  /dev/cu.usbmodem142143
wifi-131-179-27-42:~ the_prawns$ screen /dev/cu.usbmodem142131 9600
```

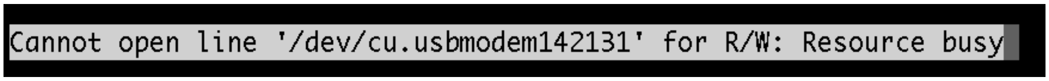
Figure 42: Examining the output from the SensorTile.



```
                   TimeStamp: 00:00:24.94
ACC_X: 23, ACC_Y: -42, ACC_Z: 1014
GYR_X: 210, GYR_Y: -2030, GYR_Z: 350
MAG_X: -45, MAG_Y: -153, MAG_Z: -252
PRESS: 998.25
                   TimeStamp: 00:00:25.04
ACC_X: 25, ACC_Y: -45, ACC_Z: 1013
GYR_X: 140, GYR_Y: -2030, GYR_Z: 350
MAG_X: -33, MAG_Y: -148, MAG_Z: -255
PRESS: 998.23
                   TimeStamp: 00:00:25.14
ACC_X: 24, ACC_Y: -44, ACC_Z: 1014
GYR_X: 210, GYR_Y: -1960, GYR_Z: 350
MAG_X: -43, MAG_Y: -156, MAG_Z: -246
PRESS: 998.25
```

Figure 43: Data being captured by the SensorTile.

10. If you receive the error shown in Figure , pause the program by clicking the two parallel yellow bars next to the run button (see Figure), unplug the SensorTile board from the Mac, wait 10 seconds, and reconnect it. Once the cable is reconnected, repeat steps 6-9.



```
Cannot open line '/dev/cu.usbmodem142131' for R/W: Resource busy
```

Figure 44: Common error message when examining data from the SensorTile using "screen".

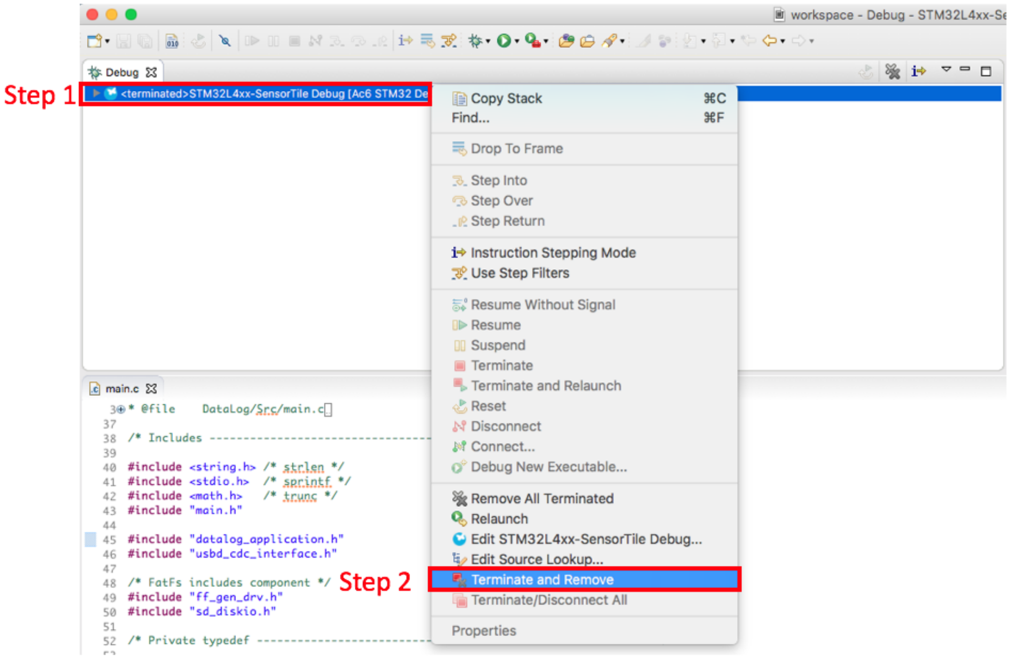
Figure 45: Pause button to temporarily halt execution of code on the SensorTile board.

5.6. Flash

This section will guide users through the process of uploading a compiled binary file onto the SensorTile for execution. Once the program is uploaded to the board, it will run every time the SensorTile is supplied with power. The SensorTile will no longer need to be connected to the Nucleo board, nor will users have to interface with System WorkBench (the IDE).

1. Terminate and remove all existing applications on the SensorTile board as shown in Figure .

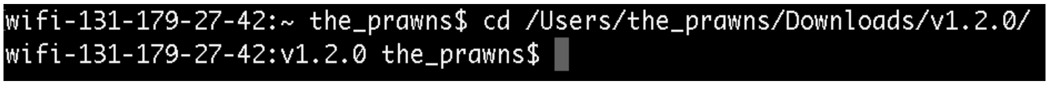
   Note: Ensure to remove ALL existing applications. Figure only contains one application. If there are more, delete all of them.



```
* @file    DataLog/Src/main.c

/* Includes ------------------------------------

#include <string.h> /* strlen */
#include <stdio.h>  /* sprintf */
#include <math.h>   /* trunc */
#include "main.h"

#include "datalog_application.h"
#include "usbd_cdc_interface.h"

/* FatFs includes component */
#include "ff_gen_drv.h"
#include "sd_diskio.h"

/* Private typedef ----------------------------
```

Figure 46: Terminating and removing existing applications on the SensorTile board.

2. Open a terminal session, and navigate to the **v1.2.0** directory.

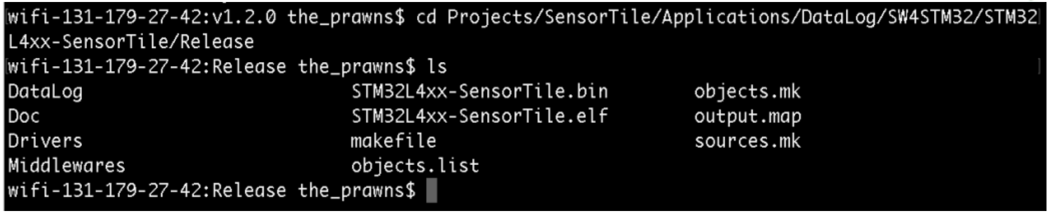


```
wifi-131-179-27-42:~ the_prawns$ cd /Users/the_prawns/Downloads/v1.2.0/
wifi-131-179-27-42:v1.2.0 the_prawns$ 
```

Figure 47: Navigating to the v1.2.0 directory.

3. Issue the command shown below and depicted in Figure .

```
$ cd Projects/SensorTile/Applications/DataLog/SW4STM32/STM32L4xx-SensorTile/Release
```



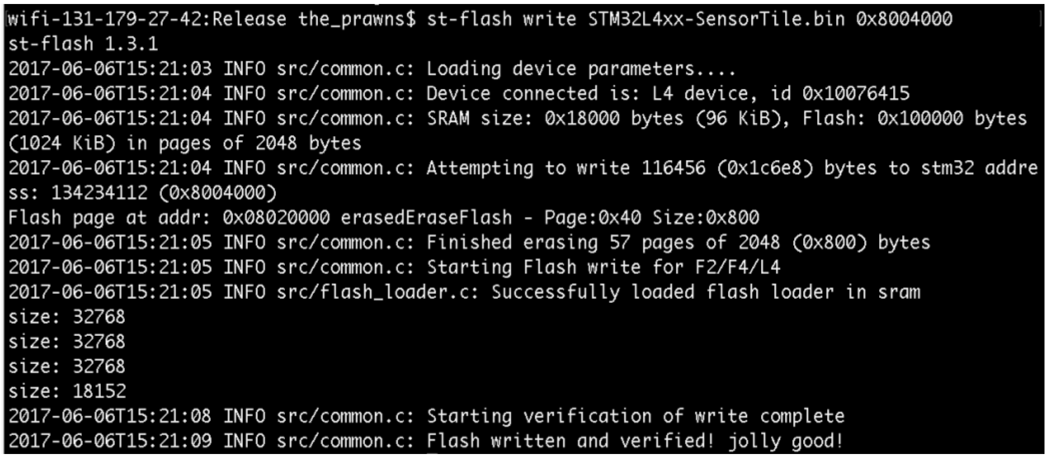
```
wifi-131-179-27-42:v1.2.0 the_prawns$ cd Projects/SensorTile/Applications/DataLog/SW4STM32/STM32
L4xx-SensorTile/Release
wifi-131-179-27-42:Release the_prawns$ ls
DataLog                         STM32L4xx-SensorTile.bin        objects.mk
Doc                             STM32L4xx-SensorTile.elf        output.map
Drivers                         makefile                        sources.mk
Middlewares                     objects.list
wifi-131-179-27-42:Release the_prawns$
```

Figure 48: Navigating to the directory containing the binary file compiled earlier in this tutorial.

4. Ensure that the directory contains the same files as seen in Figure .

5. Issue the command shown below and depicted in Figure .

```
$ st-flash write STM32L4xx-SensorTile.bin 0x8004000
```



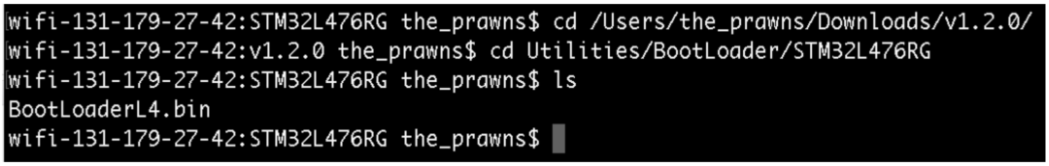
```
wifi-131-179-27-42:Release the_prawns$ st-flash write STM32L4xx-SensorTile.bin 0x8004000
st-flash 1.3.1
2017-06-06T15:21:03 INFO src/common.c: Loading device parameters....
2017-06-06T15:21:04 INFO src/common.c: Device connected is: L4 device, id 0x10076415
2017-06-06T15:21:04 INFO src/common.c: SRAM size: 0x18000 bytes (96 KiB), Flash: 0x100000 bytes
(1024 KiB) in pages of 2048 bytes
2017-06-06T15:21:04 INFO src/common.c: Attempting to write 116456 (0x1c6e8) bytes to stm32 addre
ss: 134234112 (0x8004000)
Flash page at addr: 0x08020000 erasedEraseFlash - Page:0x40 Size:0x800
2017-06-06T15:21:05 INFO src/common.c: Finished erasing 57 pages of 2048 (0x800) bytes
2017-06-06T15:21:05 INFO src/common.c: Starting Flash write for F2/F4/L4
2017-06-06T15:21:05 INFO src/flash_loader.c: Successfully loaded flash loader in sram
size: 32768
size: 32768
size: 32768
size: 18152
2017-06-06T15:21:08 INFO src/common.c: Starting verification of write complete
2017-06-06T15:21:09 INFO src/common.c: Flash written and verified! jolly good!
```

Figure 49: 'st-flash' command and resulting output to terminal.

6. Navigate back to the **v1.2.0** directory as seen in step 2.

7. Issue the command below and depicted in Figure . Ensure the contents match Figure .

```
$ cd Utilities/BootLoader/STM32L476RG
```

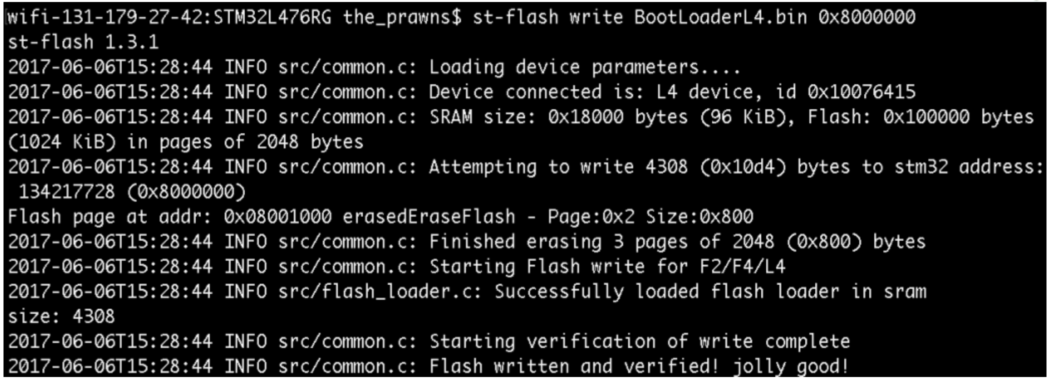


```
wifi-131-179-27-42:STM32L476RG the_prawns$ cd /Users/the_prawns/Downloads/v1.2.0/
wifi-131-179-27-42:v1.2.0 the_prawns$ cd Utilities/BootLoader/STM32L476RG
wifi-131-179-27-42:STM32L476RG the_prawns$ ls
BootLoaderL4.bin
wifi-131-179-27-42:STM32L476RG the_prawns$
```

Figure 50: Navigating to ".../v1.2.0/Utilities/BootLoader/STM32L476RG". Contents of said directory.

8. Issue the command below and depicted in Figure .

```
$ st-flash write BootLoaderL4.bin 0x8000000
```



```
wifi-131-179-27-42:STM32L476RG the_prawns$ st-flash write BootLoaderL4.bin 0x8000000
st-flash 1.3.1
2017-06-06T15:28:44 INFO src/common.c: Loading device parameters....
2017-06-06T15:28:44 INFO src/common.c: Device connected is: L4 device, id 0x10076415
2017-06-06T15:28:44 INFO src/common.c: SRAM size: 0x18000 bytes (96 KiB), Flash: 0x100000 bytes
(1024 KiB) in pages of 2048 bytes
2017-06-06T15:28:44 INFO src/common.c: Attempting to write 4308 (0x10d4) bytes to stm32 address:
 134217728 (0x8000000)
Flash page at addr: 0x08001000 erasedEraseFlash - Page:0x2 Size:0x800
2017-06-06T15:28:44 INFO src/common.c: Finished erasing 3 pages of 2048 (0x800) bytes
2017-06-06T15:28:44 INFO src/common.c: Starting Flash write for F2/F4/L4
2017-06-06T15:28:44 INFO src/flash_loader.c: Successfully loaded flash loader in sram
size: 4308
2017-06-06T15:28:44 INFO src/common.c: Starting verification of write complete
2017-06-06T15:28:44 INFO src/common.c: Flash written and verified! jolly good!
```

Figure 51: Flashing the bootloader.

9. Disconnect the USB cables connecting the SensorTile and the Nucleo board's to the Mac. This will power down the devices, enabling us to safely disconnect the SWD connection between the SensorTile board and the Nucleo board.

10. Disconnect the SWD cable connecting the SensorTile board to the Nucleo board.

11. Reconnect SensorTile board to your Mac. Do not reconnect the Nucleo board to your Mac.

12. Notice that the SensorTile device immediately starts streaming data over serial USB connection to your Mac by examining the blinking LED.

13. Examine the data transmitted from the SensorTile by repeating step 9 from 5.5 Debug.

14. You will now observe complete system operation with a stand-alone, automatically operating SensorTile IoT system.