



mpideXcode

Embedded Computing Template on Xcode 4.2

Installation Guide



© Rei VILO, 2010—2012

Website <http://sites.google.com/site/vilorei/>

GitHub repository <https://github.com/rei-vilo/embedXcode>

Wunderkit workspace <http://www.wunderkit.com>

How to Help!



Contribute to the project on [GitHub](#).



Help me buy books on Xcode through my [amazon Wish List](#).

Summary

1. Installation	3
2. Create a New Project	4
3. Project Configuration	6
3.1. Declare Sketch .pde File as C++ File	6
3.2. Declare User's Sketchbook.....	8
3.3. Add User's Libraries	9
3.4. Declare Sources for Code-Sense	11
4. Change the Board	14
5. Re-Indexing Keywords	15
6. Add a File	19
7. Code-Sense Features	19
8. References	20
9. Version History	20

1. Installation

Before installing the template, here are the pre-requisites:

For all boards:

- Install Arduino 0023.
- Launch it.
- Define the path of the sketchbook in the menu Arduino > Preferences > Sketchbook location.

Arduino provides the reference for code-sense.

As an option:

- If chipKIT boards are to be used, install Mptide 0023.
- Launch it.
- Define the path of the sketchbook in the menu Mptide > Preferences > Sketchbook location.

For other boards with a Processing-based IDE, same procedure:

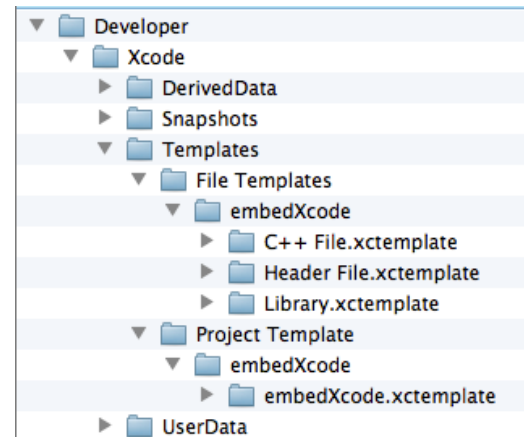
- Install the corresponding Processing-based IDE.
- Launch it.
- Define the path of the sketchbook.
- Additionally, develop a specific makefile and adapt the Step1 makefile.



Now, the template can be installed:

Check and create `~/Library/Developer/Xcode`

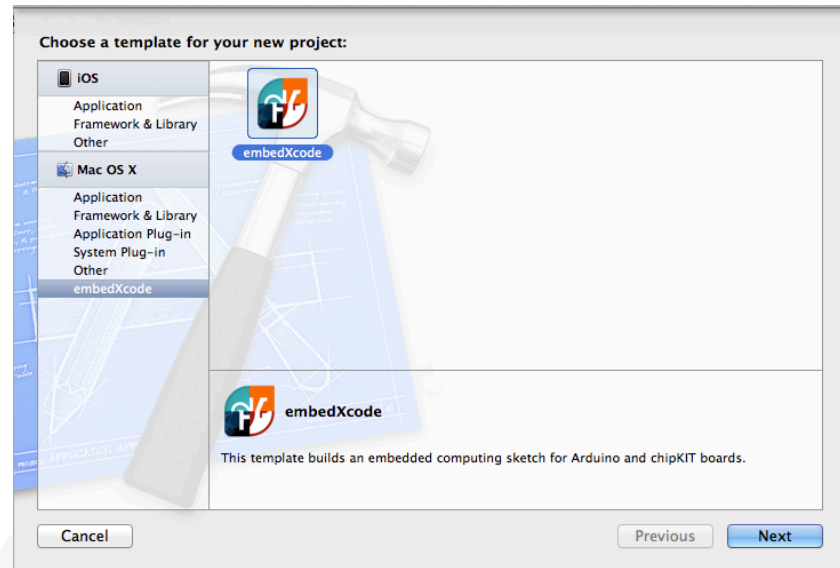
Copy the folder Templates into `~/Library/Developer/Xcode`



2. Create a New Project

Call the menu `File > New > New Project...` or press `⌘N`.

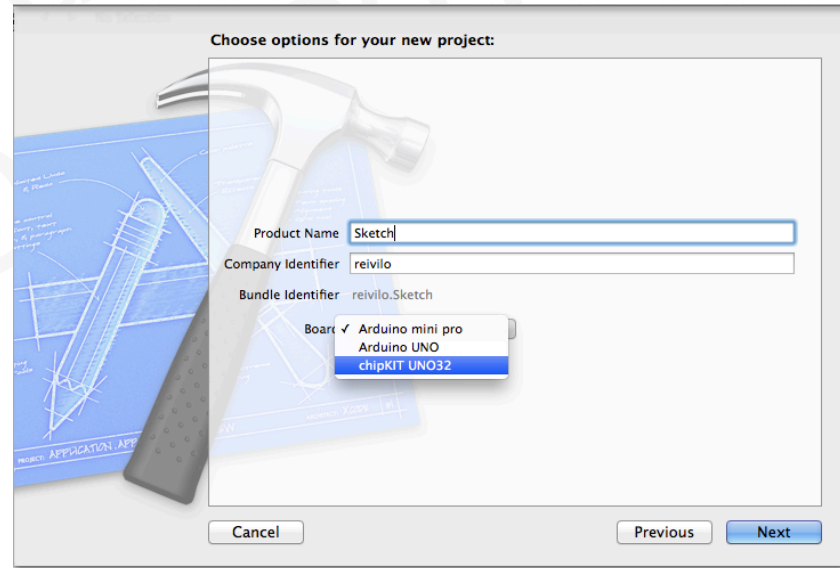
Select embedXcode > embedXcode.



Type in the name of the project.

Select the board on the drop-down list.

Click on Next to save.



3. Project Configuration

The template doesn't define all the parameters, so some of them need to be set manually.

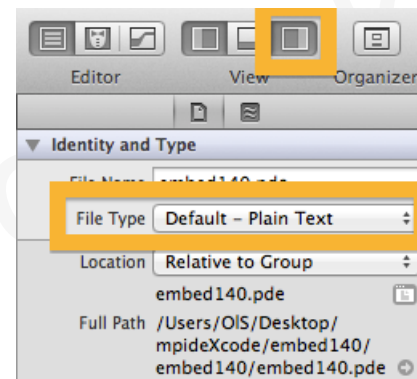
The goal is to have the following work done by the template, provided the right keywords are known.

3.1. *Declare Sketch .pde File as C++ File*

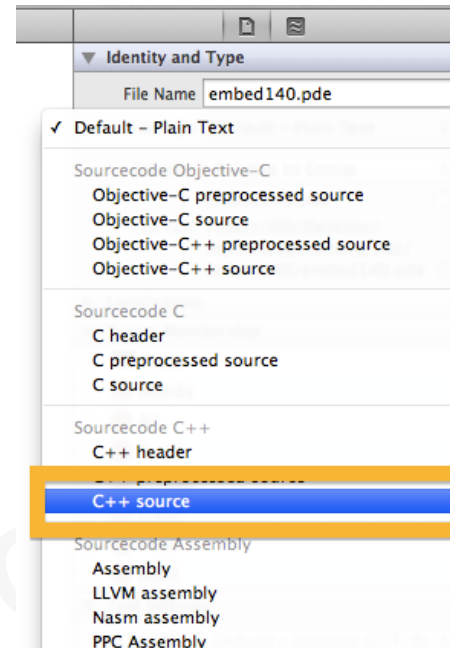
The sketch .pde file is considered as plain text. For code-sense, it should be declared as C++ file.

Select the sketch .pde file.

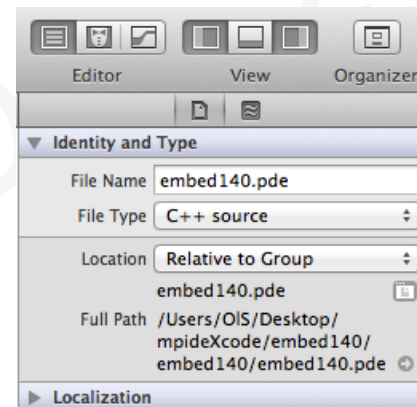
In the right-most column,



Click on the drop-down list of File Type.
Select C++ source.



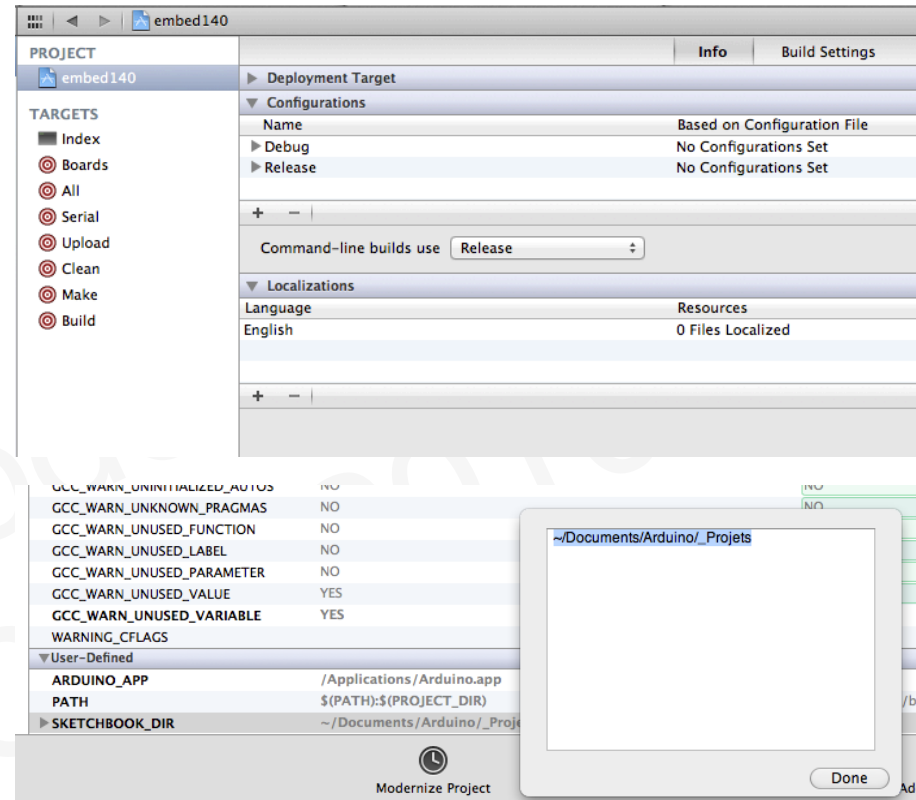
Now, the sketch is considered as C++ code for code-sense.



3.2. Declare User's Sketchbook

The user's sketchbook is a folder where the user's sketches are saved, among them the libraries in a dedicated sub-folder Libraries.

Select the project and the Build Settings pane.



At the very bottom, double-click on SKETCHBOOK_DIR and either type in the name of the folder or drag-and-drop it from a Finder window.

The ~ character is accepted.

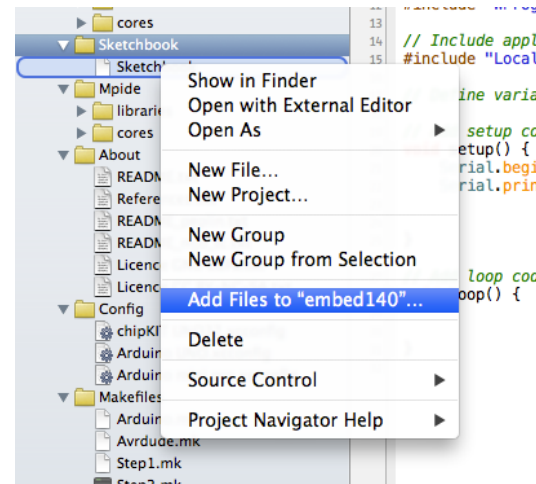
If no sketchbook is defined, SKETCHBOOK_DIR takes the value defined for Arduino or chipKIT during the installation process.

3.3. Add User's Libraries

Open the Sketchbook group on the project hierarchy.

Right-click to obtain the contextual menu.

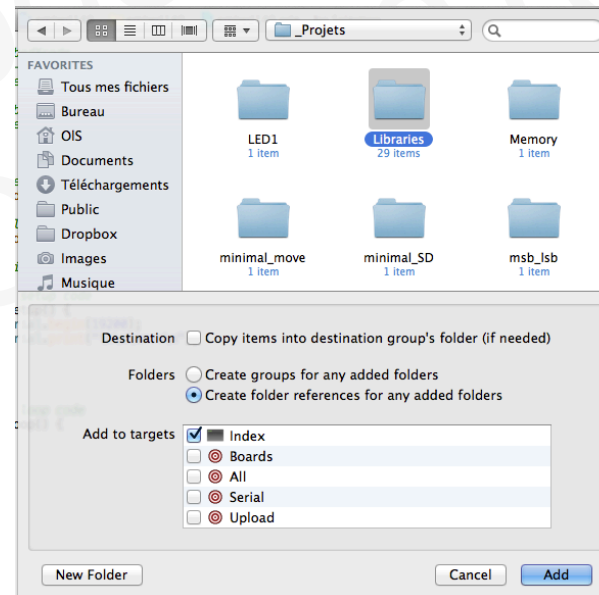
Choose Add file to...



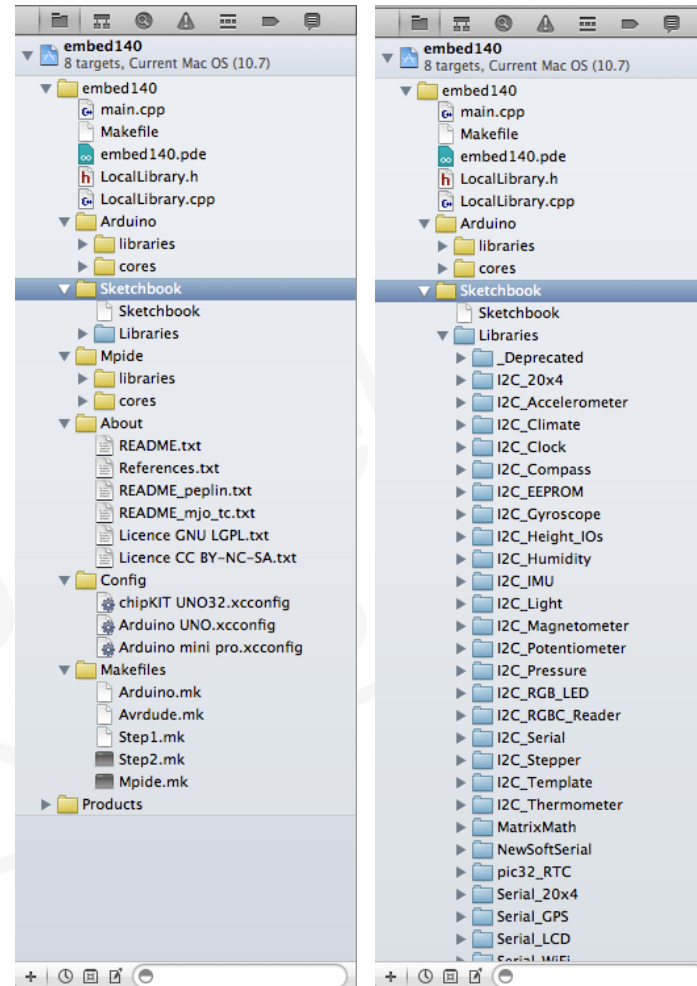
Select the Library sub-folder on the sketchbook folder, tick Add to target > Index and validate with Add.

Both Create group for any added folders and Create folder references for any added folders are relevant.

Don't tick Copy items into destination group's folder (if needed) folder (in needed) to avoid duplicating files.



The project hierarchy shows all your libraries.

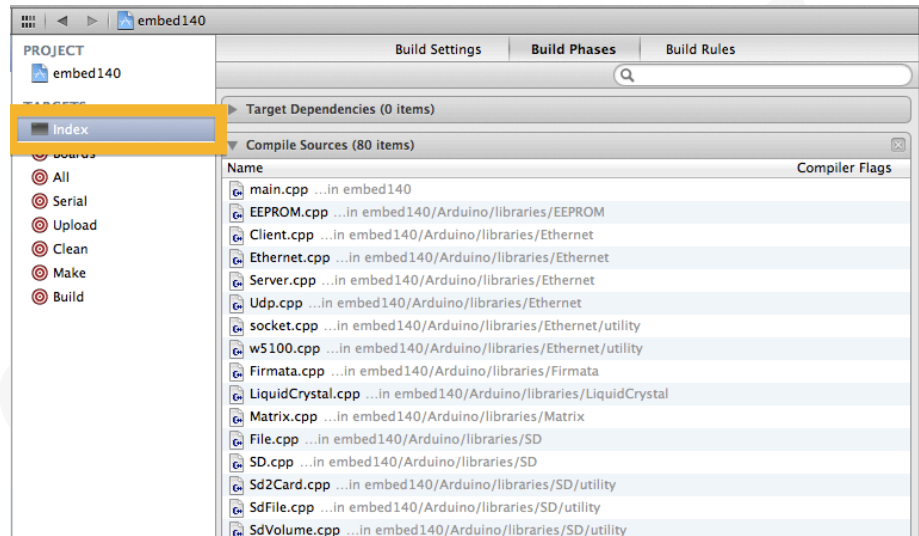


3.4. Declare Sources for Code-Sense

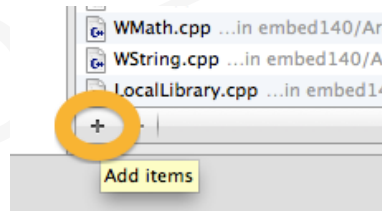
Standard C++ keywords are already known, but not the Arduino and user's library keywords.

So Xcode needs to be taught where to find them.

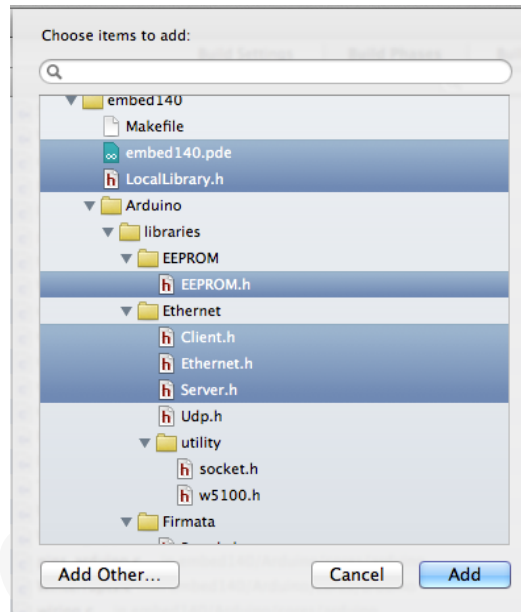
Select the target Index and the Build Phases pane.



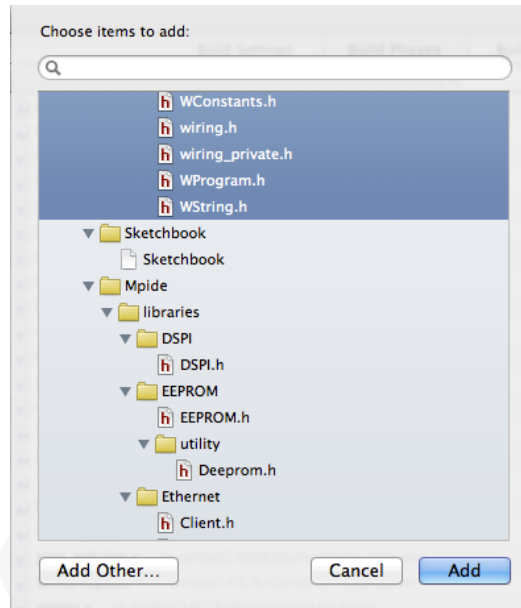
Go a the bottom of the list and click on the + button.



A list shows up.



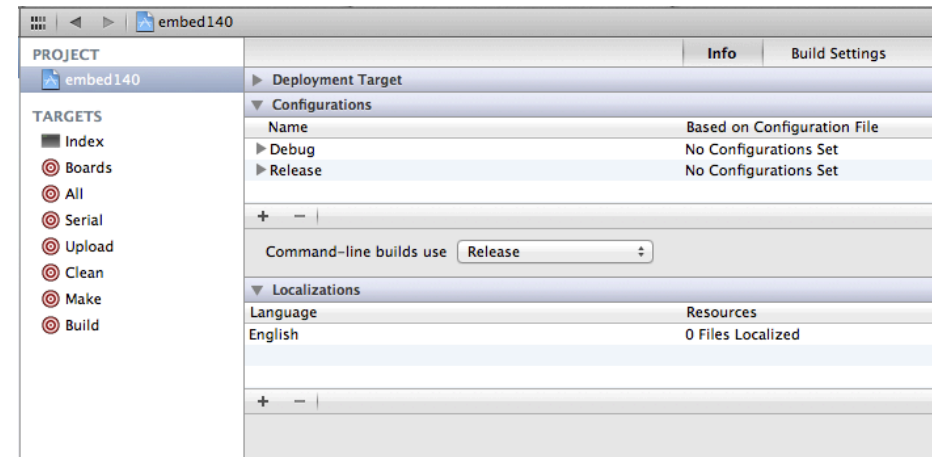
Select all the .h and .cpp files and click on Add.



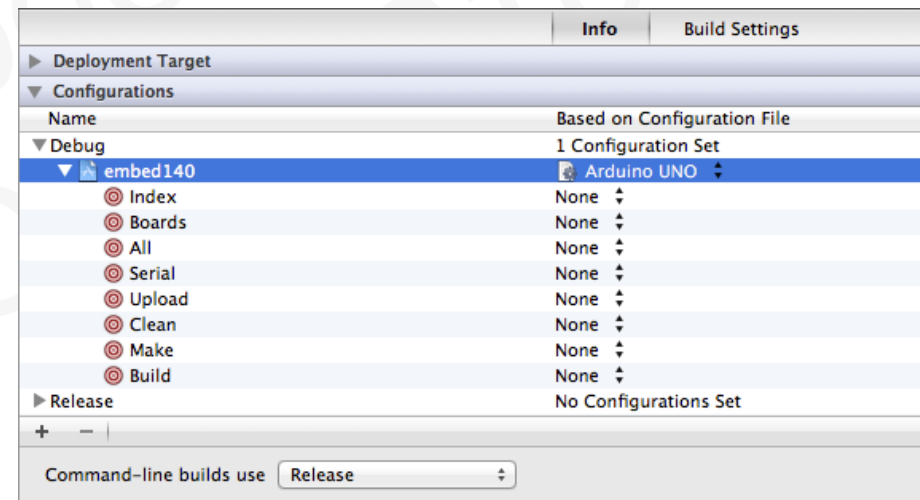
You're ready now!

4. Change the Board

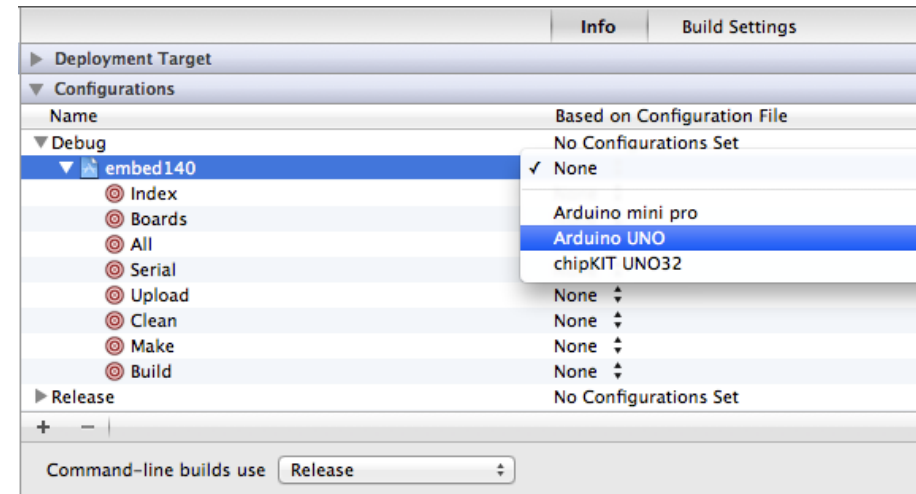
To change the board, select the project and the Info pane.



A drop-down list shows the boards available.



Just select one.

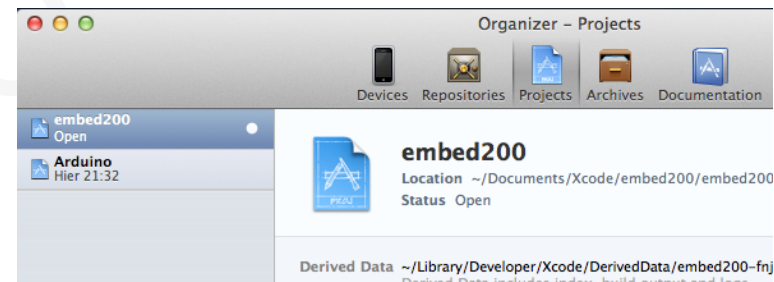


5. Re-Indexing Keywords

If code-sense doesn't work, we need to force a re-indexing of the key words.

To do so, first close the project.

Call the menu Window > Organiser and select the Projects pane.

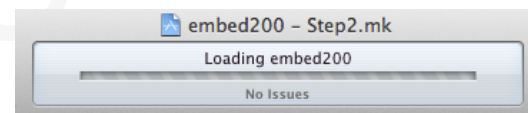
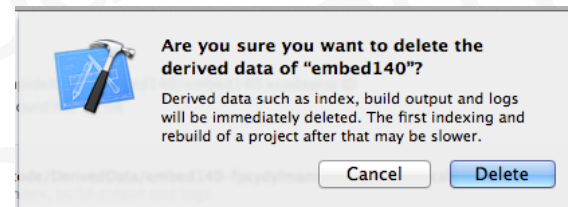
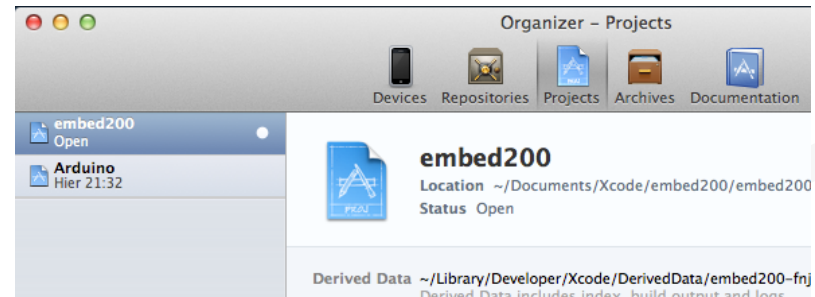


Select then the project.

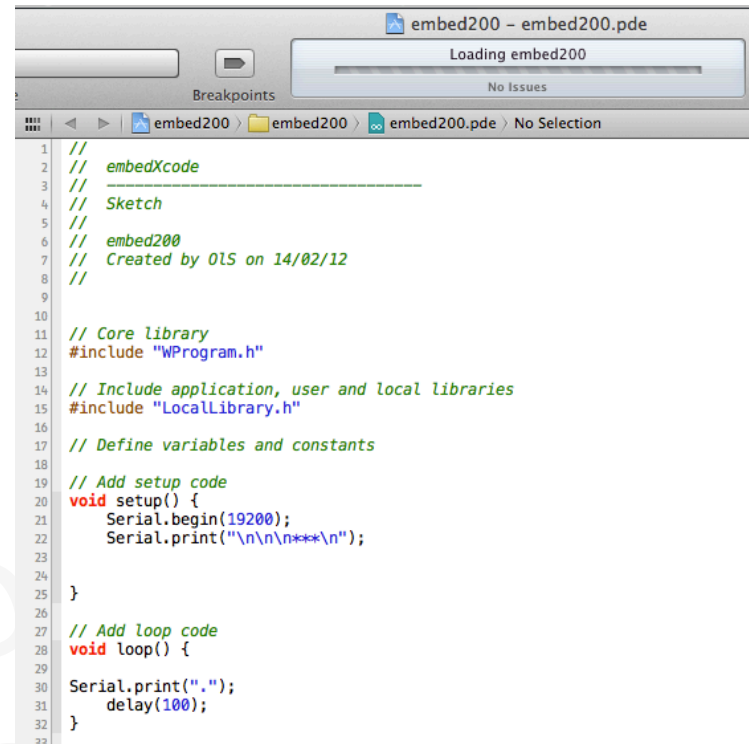
The index is saved within the Derived Data folder.
Click on the Delete button to delete them.

Confirm the deletion.

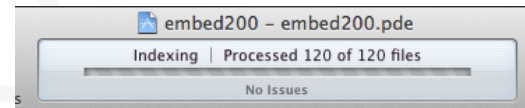
Load the project.



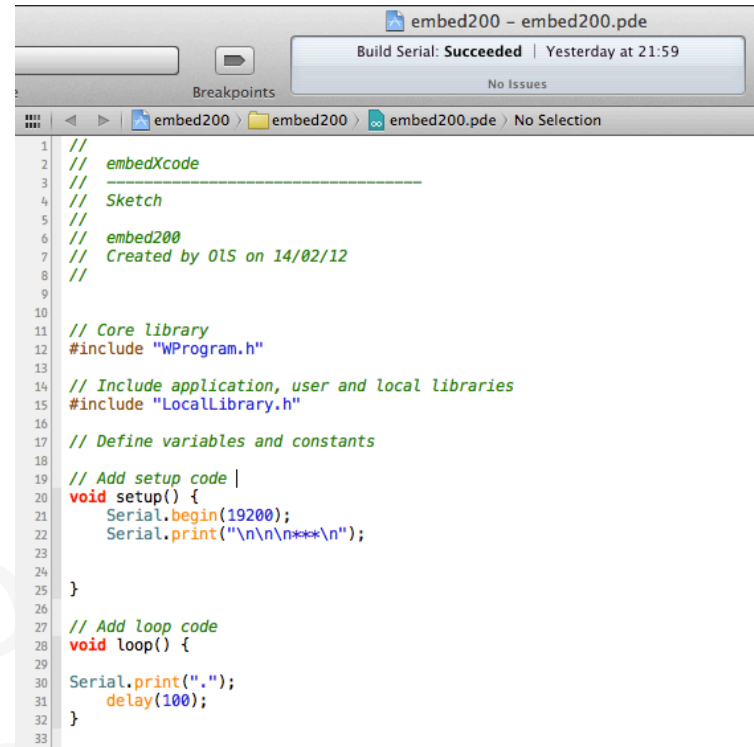
There's no code-sense yet: everything is in black-and-white, except standard C++ keywords.



The index is being built.



When the index is built, code-sense shows pretty colours.



The screenshot shows the Arduino IDE interface. At the top, a status bar indicates 'Build Serial: Succeeded' and 'Yesterday at 21:59'. Below this, a breadcrumb trail shows the file path: 'embed200 > embed200 > embed200.pde'. The main editor window displays a sketch with the following code:

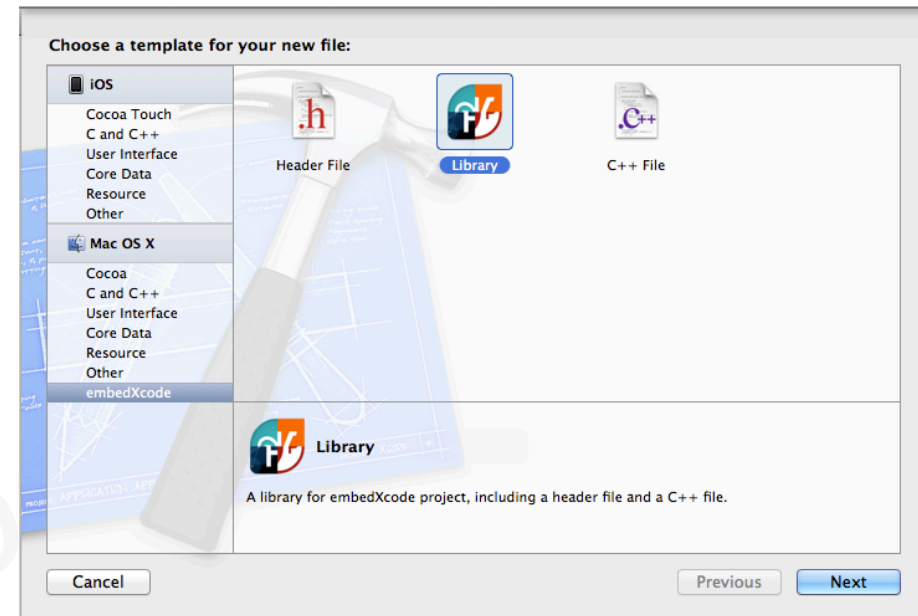
```
1 //  
2 // embedXcode  
3 // -----  
4 // Sketch  
5 //  
6 // embed200  
7 // Created by OLS on 14/02/12  
8 //  
9 //  
10 //  
11 // Core library  
12 #include "WProgram.h"  
13 //  
14 // Include application, user and local libraries  
15 #include "LocalLibrary.h"  
16 //  
17 // Define variables and constants  
18 //  
19 // Add setup code |  
20 void setup() {  
21   Serial.begin(19200);  
22   Serial.print("\n\n\n***\n");  
23 }  
24  
25  
26 // Add loop code  
27 void loop() {  
28   Serial.print(".");  
29   delay(100);  
30 }  
31  
32  
33
```

The code is syntax-highlighted: comments are green, keywords like 'void' are red, and strings are blue. A large, faint watermark 'embedXcode © Rei VILO' is visible across the lower half of the image.

6. Add a File

Call the menu File > New > New File... or press ⌘N

Select embedXcode and then Header File, C++ file or Library.



Library creates a header file and a C++ code file with the `#include` statement ready!

```
#include "LocalLibrary.h"
```

7. Code-Sense Features

Apart from pretty colours on the code and enhanced visibility, code-sense brings:

- auto-completion,

```
13 // Include application, user and local libraries
14 #include "Locallibrary.h"
15 #include "I2C_Serial.h
16 I2C_Serial.h
17 I2C_Stepper.h
18 // Define constants
19 void setup() {
20   Serial.begin(19200);
21   Serial.print("\n\n\n\n\n");
```

- code-snippets and check-as-you-type code monitoring,
- click-to-definition



8. References

See documents on the About folder for additional contributions and detailed references.

9. Version History

Release	Date	Content
1	Feb 15, 2012	Initial release
2	Feb 18, 2012	embedXcode 2 release