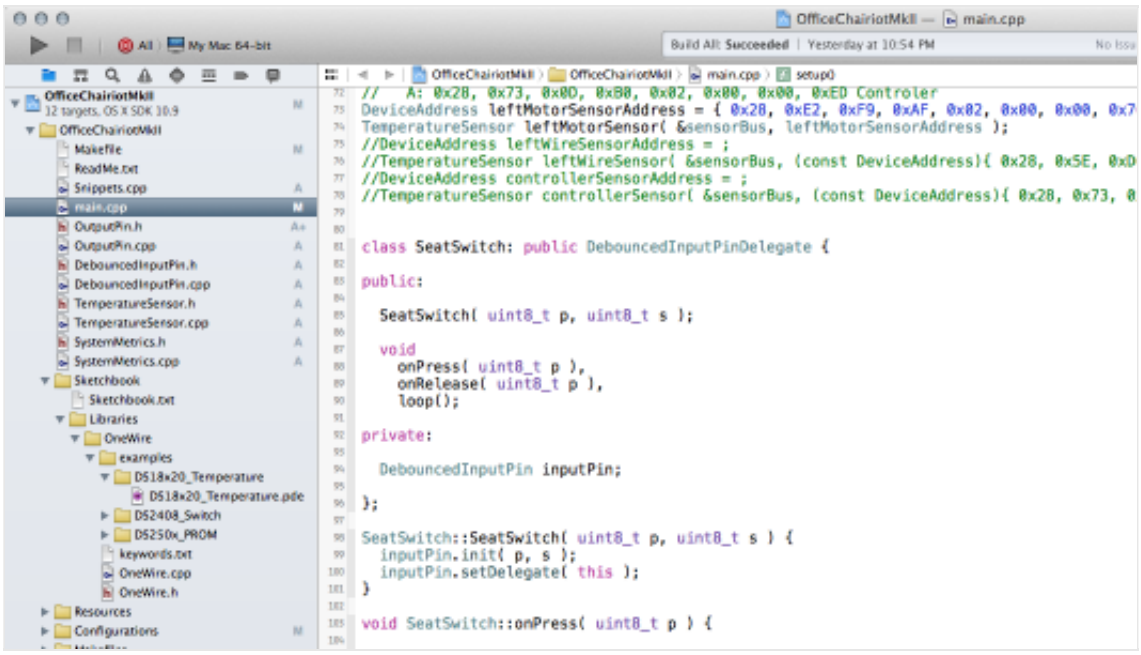
embedXcode: A Better Way to Develop for Arduino on the Mac using Xcode

If you are writing code for your Arduino on a Mac and you've previously written code using Apple's FREE Xcode IDE, you know that the standard Arduino development environment is lacking in functionality, especially those which professional software developers have had in their IDEs for years. My personal favorite is Apple's Xcode IDE, which is the primary IDE used in developing applications for the OS X on the Mac and for iOS applications on Apple's mobile platforms (which are technically ALSO running Apple's OS X operating system).



Arduino C++ development in Xcode on OS X using embedXcode+

To be clear, this article is NOT a knock on the Arduino environment. The Arduino IDE you download from arduino.cc is quite a capable little IDE for FREE. It's got just what most people need to get up and running with their shiny new Arduino board. Therein lies the rub: It's great for getting started. If you find yourself writing larger sketches (the term used by Arduino peeps for programs that run on Arduino board) and you're maintaining more and more libraries, it's easy to see the need for more advanced IDE features like code completion, live syntax checking, indexing and whole

project management (to name a few). The Arduino IDE was not designed for large embedded development projects.

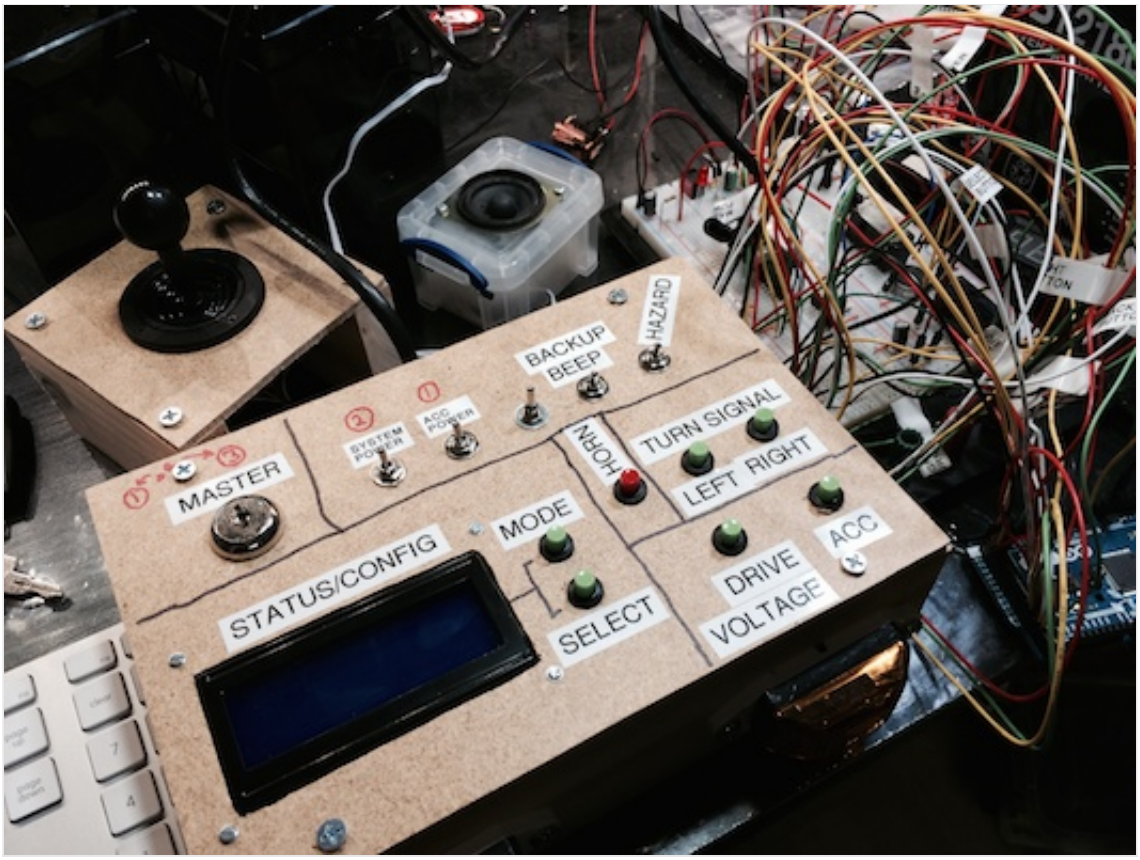
A very smart guy named Rei Vilo has written a plugin for Xcode called, "embedXcode" (and "embedXcode+") which VERY EASILY allows you to build Arduino projects (among other development boards in a continuously growing list) in a very advanced and solid IDE. You can download the standard embedXcode plugin system for FREE or a donation (although, I suggest you don't be a cheapskate and pony up *something* for his efforts). I donated at the professional level for about US$129 and got the embedXcode+ plugin. It was worth every penny. Xcode rocks. Arduino boards rock. Writing Arduino projects in Xcode super-double-rocks, thanks to Rei Vilo.

Also, coincidentally, as I was writing this I found a link to a podcast at the OpenSourceHardwareGroup.com website where they interviewed Rei about embedXcode. It's a good listen:

http://opensourcehardwaregroup.com/oshgroup-040-use-embedxcode-to-program-multiple-development-boards-with-arduino-code/

Quick on the Why

When you design larger projects around an Arduino, you start to realize the need for more modular code. Ideally, you separate pieces of code into modules that are more specific to tasks and specific pieces of hardware. For instance, I'm building a fairly complex control system for the latest version of my Office Chairiot (a stupidly fast motorized office chair). The control system has sounds, light indicators (turn signals, headlights, hazard blinkers, etc.), throttle control through a joystick to a robot motor controller and a security system, among other nutty modules. Why? A motorized office chair isn't silly enough without having an airliner-level cockpit control panel to run it.



Office Chairiot Mark II Control Panel Test Rig

To keep the very complex Office Chairiot Mark II control system code more readable and maintainable, I am writing individual classes to represent different hardware components, like the MP3 sound player board and the liquid crystal text display. As I build these classes, I collect more and more files in my project. When I collect more and more files and classes in my project, my tiny mind can't remember all the class functions and constants used to work with these modules, let alone the function parameter lists and whatnot. It all gets quite unwieldy very quickly. This is where Xcode shines! It was designed for this type of project.

The other thing that makes life easier with larger projects is source code management and version control. There are utilities and online applications for managing your large source code. If you've ever worked on a project and started down a path working on a feature you thought would be cool, but then later realized it wasn't, how did you backtrack to the previous know-working version? Unless you were saving off a copy of your working directory the every few minutes, you don't. Undo in your IDE only works so far back in history. Revision control does all that management for you and

it's stupid-simple with systems like Git. Older systems like Subversion and CVS did this, as well, but the new and easier way to do it is with Git and the web services around it like Github.com and Bitbucket.org (which is what I use primarily for personal projects).

Source control systems like Git allow you to store your code in a repository and that system keeps track of changes in your project over time, allowing you to roll back to older versions, if you want. they also allow you to try new things and toss them in the trash if you don't like them. They allow for the maintenance of multiple branches of development, say, for old and new version of your project. It's easy to get into. Google and learn.

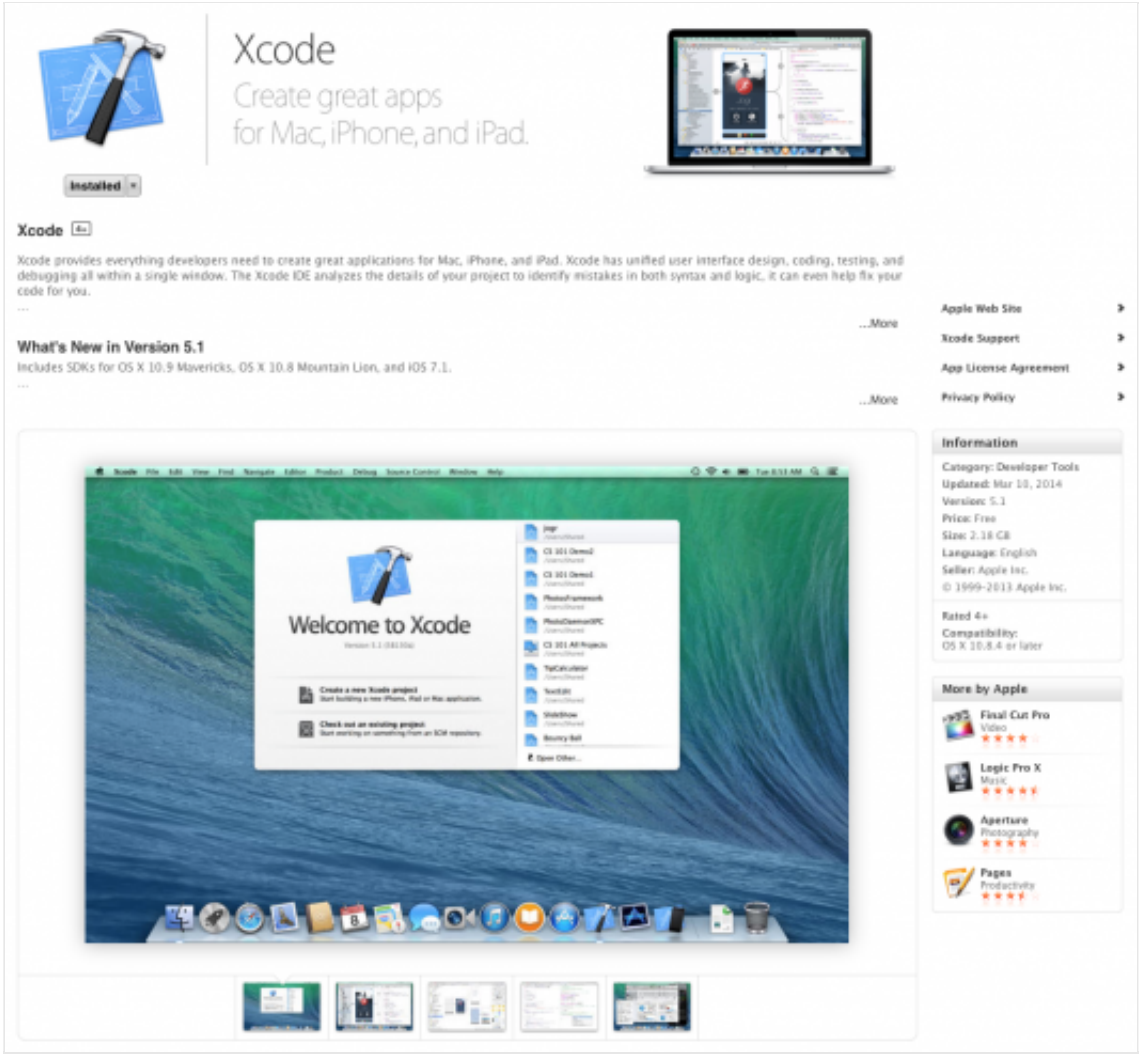
If I'm building a blinking LED project, using Xcode is completely silly. One file (sketch) to make a little LED blink is EASY to write and maintain and the Arduino IDE does this perfectly well. So, choose your tools and environments wisely, hoppergrass.

Yadda, Yadda, Yadda... Shut Up and Set Up!

OK, so, to get this stuff up and running is actually VERY simple. Rei wrote a nice book to walk you through the setup of embedXcode. I'm not going to go into the nitty-gritty details of how to do that since the book already does it. But, here's the big overview step-by-step:

Step 1

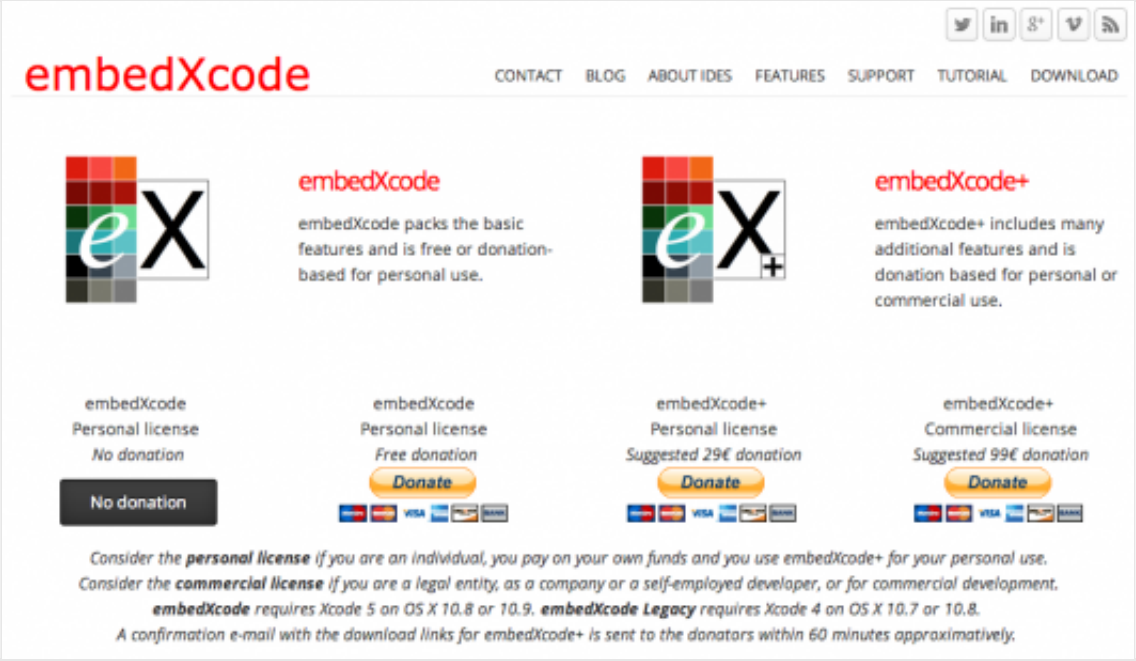
Download Xcode from Apple's developer website for FREE using the App Store app on your Mac.



Download and install Xcode via the App Store app in OS X

Step 2

Download and run the package installer for embedXcode or embedXcode+ from Rei Vilo's site:

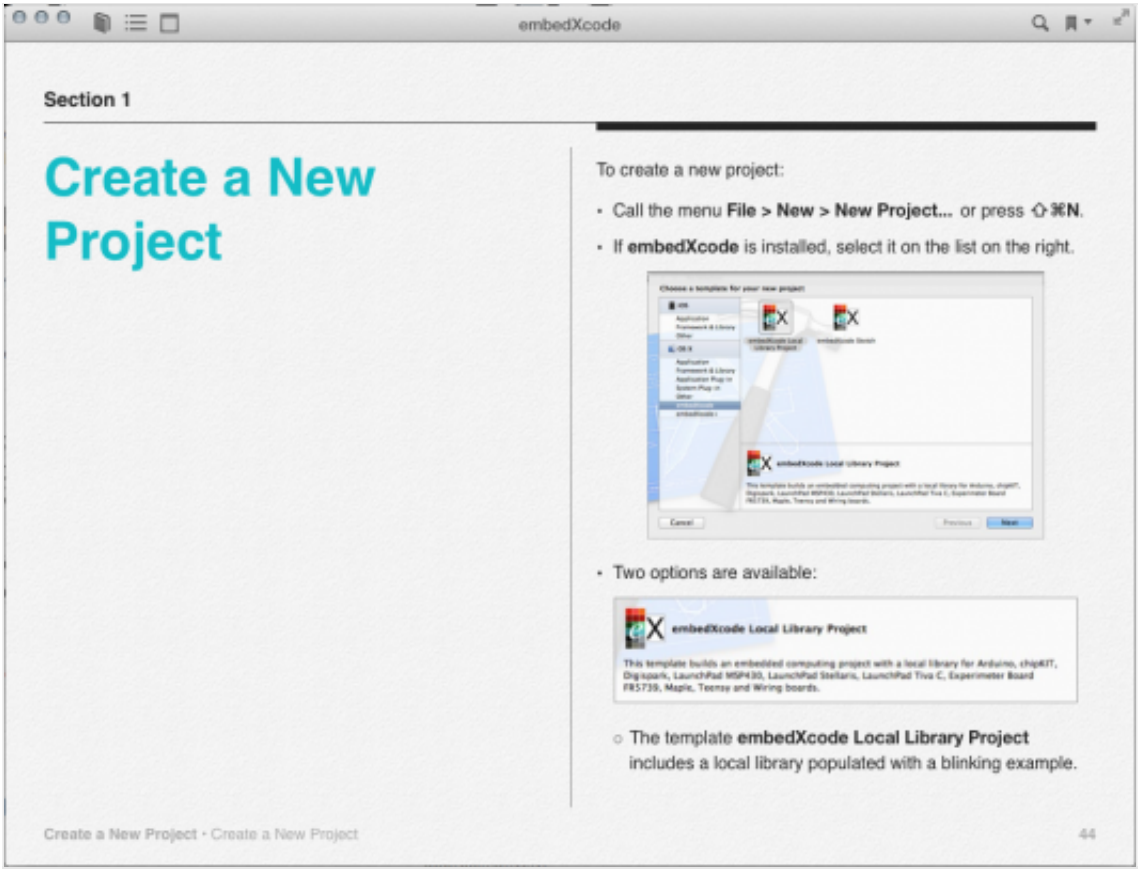


Download embedXcode

http://embedxcode.weebly.com/download.html

Step 3

Follow the directions in the embedXcode User Manual for creating your first Arduino project using Xcode and the embedXcode plugin!



User Manual for embedXcode

Boom! Bob's your uncle!

In Summary...

It really is that simple. Rei has worked through all the complexities of customizing projects in Xcode for specific tool chains and made our lives WAY easier for developing Arduino and othe embedded firmware on the Mac. Many, MANY thanks to Rei Vilo for his hard work and his continued dedication to keep the plugin updated. I can't recommend this plugin enough if you're getting more serious with your Arduino and AVR projects.

Happy embedded coding!

Rating: 0.0/**5** (0 votes cast)