

# Using Arduino

## What are the different file extensions for

While creating a sketch in the Arduino IDE, you have access to a number of different file extensions for use with your source code files. Each one has its own particular use which I'll explain below.

- **.ino**

This is the main extension for your sketch file(s). Your primary sketch file must be an `.ino` file named the same as the directory it is located in. You can create additional `.ino` files in your sketch named however you like. Before compilation, all additional `.ino` files are appended to the primary sketch file. They are copied in ascending order based on the file name, and all are run through the standard process of gathering includes and generating prototypes. For more information on what the IDE does before compiling your code, visit this FAQ: [What does the IDE change in my sketch?](#)

- **.pde**

This is the default extension for sketches written for the Arduino IDE prior to the release of version 1.0. If you use an IDE version of at least 1.0.0 or greater, you should rename your sketch file from `.pde` to `.ino`. Also keep your IDE up to date, pre 1.0 versions do not support many new libraries due to significant changes to the Arduino core API.

- **.h**

Header files, or files with the extension `.h` can be utilized in a number of ways. If your sketch uses a set of constants that multiple `.cpp` or `.ino` files may use, you can create a single location for your common variables. Functions that are defined in different `.cpp` files can have their declarations placed in a header for easy reuse.

- **.cpp**

This extension is a C++ source file. Sometimes a sketch becomes quite large and you can utilize a `.cpp` file to separate sections of your code. Definitions inside a `.cpp` file that you want to access elsewhere should have their declarations inside a header (`.h`) file. This allows your sketch to include the header, and its functionality. Also an important point is; your `.cpp` files do not go through the IDE pre-compilation modifications, and therefore have no prototypes generated. For more information on using multiple files, visit this article: [Breaking a sketch into multiple files.](#)

- **.c**

If you wish to write C code and use it within Arduino, you need to use a `.c` extension instead of a `.cpp` file. Using a `.c` file alone however is not the complete solution, visit [this FAQ](#) for more information.

Unique solution ID: #1036

Author: Christopher Andrews

Last update: 2014-05-12 14:26