Far Horizons Project Directory Organization

Rev 2.0 August 25, 2013 Lou Nigra

Document Formats

Routine documentation should be in one of the following open formats compatible with free and open source viewers/editors such as OpenOffice, LibreOffice and web browsers, as well as MS Office:

Text-based:

.txt .rtf .docx .html .pdf

Spreadsheet:

.xlsx

Presentation:

.pptx

NOTE ON FILE SIZE:

With figures added to documents, file size can get into the several MB and higher (especially with presentations). Avoid making any document that will be put up in the public domain larger than it has to be. A rough limit might be 1 or 2 MB. You can take forever optimizing this, but just paying attention to it by taking the time to crop and resize images to the actual size needed for real-life viewing can pay off dramatically. Hyperlinks, of course are not a problem.

This DOES NOT apply to internal documents such as presentations (go crazy on those!) and reports. If we publish these to the web after the fact, we can take the time to make a lightweight version appropriate for the web.

The naming of things...

Code Content:

For a given programming language, please google "naming conventions" for code written in that language (e.g., "arduino naming conventions") and try to follow this as best you can in the code. In any case, be self-consistent.

Code Source Files:

Source file names sometimes have conventions as well, but generally not.

Other Files:

For non-code files that you create, the actual name portion of file names should use "CamelCase" convention, where each descriptive word is capitalized rather than separated by spaces or underscores or the like.

After the name part, which should be generic and descriptive, you should add something to distinguish it from other files of the same basic name separating it from the name with a dash. Where that needs further separation, we use underscores. Finally, there's the document suffix that is always separated by a period. That should be the ONLY period.

Every document should have a generic descriptive name and files of the same type should be distinguished from each other by a sequence number, a version number or a date:

a) A simple sequence number: SomeDescriptiveName-34.jpg

b) A version number with two levels (no more, no less): SomeOtherName-V3_12.docx

c) A date: OneReallyAmazingVideo-2013_06_01.mov NOTE: the format is NOT the American “dmy” convention, but the sensible standard the rest of the world uses in its reversed form: yyyy_mm_dd. Add "_ss" if seconds are necessary.

d) A person's initials can be optionally added to any filename: TheBestDesignEver-V943_1_LN.docx

Directories:

Preferably a simple, single-word lowercase name if it's a category like the directories described below. If it has to be more specific, like a directory for a particular program (as with Arduino sketches) then use CamelCase.

The placing of things...

The project directory is organized hierarchically. In the descriptions below, "./" represents the project directory and directories have a trailing "/", while files don't.

./data/

Raw(-ish) data that is large and doesn't have much meaning on its own as documentation, such as raw and high resolution images that need to be kept, files of collected data samples, etc. Organize this into folders if you can and as you see fit.

This directory is will not be tracked on GitHub.

./doc

Documentation that relates to the system design rather than hardware or software design material. An example would might be project design specifications, notes and analysis.

./hardware/

Source files required to re-create and build the hardware design if the project involves hardware design. That would include Eagle schematics and board layouts, mechanical sketches, drawings, etc. The project might include distinct pieces or "modules" that would lend themselves to having separate sub-directories. If there are a lot of files for a single module, you might consider further organizing it into subdirectories such as 'electrical' and 'mechanical' for example.

./hardware/doc/

Hardware-specific design notes, analysis and documentation.

./scratchpad/

This is for any material (programs, schematics, notes, analysis, etc.) that isn't part of the actual design or is still a work in progress. For instance, when you first start on a project, you might try some experimental things to get started or maybe check out some alternative approach to some aspect along the way.

./software/

Source code for any software developed in the project. If there are separate distinct software programs.

.software/doc/

Software-specific design notes, analysis and documentation.

./doc/

Project-level notes, analysis and documentation.

./reference/

Reference material for the project, organized into subdirectories if necessary. This is for things like documents from online sources, designs and code we borrowed from someone else, etc.

The "references.txt" file describes where you got this material. It can also include a description of aother things you used even if there are no associated files, for example, a description/acknowledgment of someone outside of Far Horizons who made a significant contribution. This might have been help with troubleshooting, giving advice or conversations having an impact on the project.

Nothing in this directory except the "references.txt" file should be modified in any way.

./ProjectDirectoryOrganization.pdf

This file.

./README.md

This is visible on the GitHub project page. See the file for a description what should be in it.

./.git/

This is the directory used by git. DO NOT modify it or any of its contents.

./.gitignore

This is a hidden file that tells the git program which directories and file types to ignore in tracking with the GitHub repository. They won't get uploaded to GitHub and are not tracked for changes.

DO NOT edit this file unless you are absolutely sure of what you are doing. Consult with a Far Horizons Engineer if you think files are being suppressed that shouldn't be or if you think some file types or directories should be added to the list.