Source Tree Guide

Version 1.08

Table of Contents

1. Introduction

HCC Embedded offers a broad range of software components for use in embedded systems. These components are designed to be used in varied combinations, in many different target environments, and with many different toolchains.

To facilitate this, and to enable HCC to provide all software components in a consistent way, a source tree structure has been designed and rigorously applied to the entire product line. There may be a very short initial learning period as you start using the tree but, once you understand the tree, the system is very flexible and easy to use.

The system ensures the robustness of HCC software components over a wide variety of target environments. The base code is not altered between projects; only configuration modules are altered.

2. Summary

The schematic below shows HCC Embedded's source tree structure. Click on any element in the directory structure to view a short description of it.

> **Note:** Throughout this text the term "[module]" is used to represent a particular software component provided by HCC Embedded (for example, a particular file system or a USB host stack). The source tree can accommodate as many of these modules as required by the specific project.

```
/3rd_Party
/hcc
    /doc
        [module].pdf
    /driver
    /history
        [module].txt
    /info
    /lib
    /src
        /api
            [module].h
        /config
            [module].h
        /[module]
        /oal
        /psp
        /version
            [module].h
    /util
 /project
        /doc
```

3. Directory Usage

Third Party files (/3rd_party)

This directory contains any third party software. Software referenced by HCC modules, such as third party RTOSes and micro-controller manufacturer specific libraries, is placed here.

HCC Root (/hcc)

This directory contains the sub-directories which contain all the product components provided by HCC. A detailed explanation of each sub-directory is given below.

Documentation (/hcc/doc)

This directory contains all the documents for each of the included software components. There may be more than one document associated with a particular software component. All names are of the form: "**HCC[module] [type] [version].[ext]"**, where:

- **module** refers to the software module it covers, for example "FAT and SafeFAT".
- **type** describes the document type, for example "System Guide".
- **version** is the version of the guide, for example "v1.05".
- **ext** is typically **.pdf**, but may be another format such as **.docx** or **.txt**.

Drivers (/hcc/driver)

This directory may contain driver files associated with a particular package, for example, Windows **.inf** files to install a Windows driver required for communication with the embedded firmware package.

History Files (/hcc/history)

This directory contains the history file for each included module. Each file has the name **[module].txt**. For example, for the FAT file system this file is **/hcc/history/fat.txt**.

Package Information (/hcc/info)

This directory contains package information files for each HCC package that is included in the system. Information contained in these files describes the package, its version, dependencies and other useful information. These files are not directly used in the project – they are designed to help with the efficient management of HCC packages.

Library Modules (/hcc/lib)

This directory contains any libraries which are included in a project.

Source Code (/hcc/src)

This directory contains all the source code. It is organized into the following logical set of sub-directories.

Application Programming Interfaces (/hcc/src/api)

This directory contains, for each included HCC software module, all the header files required to completely define the API for that module. Any external code using an HCC module should include only the appropriate module header file in this directory. This gives the user of the module access to all externally visible elements of that module.

All files in this directory are based on the software module name. For example, the API file for the FAT file system is: **/hcc/src/api/api_fat.h**.

Configuration Files (/hcc/src/config)

This directory contains all the files containing configuration settings for the included modules. You should not need to modify any files in a software component that are not in this directory. Generally the files in this directory have names of the form **config_[module].h**, but for specific products other types of file may also be included, such as **.xml** or **.c** files. For example, the configuration file for the FAT file system is **/hcc/src/config/config_fat.h**.

Source Modules (/hcc/src/[module])

All the source code for each HCC software component resides in a set of sub-directories under a module root directory (for example, **/hcc/src/fat** for the FAT file system). The only files for a component that are not under this directory are the API, Configuration, and Version files.

OS Abstraction Layer (/hcc/src/oal)

Each embedded software component has supporting infrastructures. One element of these is an operating system, which could be anything from a super-loop to a fully featured RTOS. The **oal** directory contains an OS abstraction layer that can be used with or without an OS. Each software component has its own requirements of a system; the system user guide for each component specifies which functions it requires.

HCC provides a default template in this directory that can be used by systems without an OS. HCC also provides many OS abstractions for standard RTOSes, the code for which is available from HCC for placement in this directory.

Typical objects and functions abstracted in this directory are mutexes, events, and interrupt handling.

Platform Support Package (/hcc/src/psp)

For each embedded software component there are some supporting functions that may be platform-dependent or toolchain-dependent. Any functions of this type are placed in the PSP directory. The system user guide for each software component specifies what it requires from the PSP.

Typical activities present in the PSP include memcpy functions, string functions, and endianness handling.

Packages may contain target-specific PSP variants and these can be used directly, without porting the PSP.

Version Numbers (/hcc/src/version)

This directory contains all the files containing version numbers for the included modules. The files in it have names of the form **ver_[module].h**. All source files in the related software component check this version number at build time to ensure that the system is consistent. For example, the version file for the FAT file system is: **/hcc/src/version/ver_fat.h**.

Utilities (/hcc/util)

This directory contains any utility programs provided with a package to aid development or implementation of that package in a system. For example, it may contain the USB device configuration tool, HID PC demo, or PC bootloader application.

Project (/project)

This directory contains an actual project. The format of the project files here is project-specific and varies as a function of toolchains and also from project to project. For any particular project there is normally a readme file in the **/project/doc** sub-directory.

Documents (/project/doc)

This directory contains documentation describing the specific project.

4. Version

Version 1.08