Developer Source Tree Guide

Version 1.04

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 tool-chains.

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. However, once the tree is understood 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.

Summary

The schematic below shows HCC-Embedded's source tree structure. For each element in the structure there is a hyperlink to an explanatory paragraph.

Throughout this text the term [module] is used to represent a particular software component provided by HCC-Embedded (e.g., 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.

```
/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
```

HCC Root

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

Source Code

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

Application Programming Interfaces

This directory contains all the header files for each of the included HCC software modules 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. The user of the module will then have 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

This directory contains all the files containing configuration settings for the included modules. It should not be necessary 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 other types of files may also be included for specific products 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

For each HCC software component, all the source code for that component resides in a set of sub-directories under a module root directory (e.g., /hcc/src/fat for the FAT file system). The only files from that component not under this directory are the API, configuration and version files.

OS Abstraction Layer

For each embedded software component there are supporting infrastructures. One element of this is an operating system, which could be anything from a super-loop to a full-featured RTOS. This 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 of these components specifies which functions are required by that component. 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

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

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

Version Numbers

This directory contains all the files containing version numbers for the included modules. The files in this directory 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/config/ver_fat.h.

History Files

This directory contains the history files for each of the included modules. Each file will have the name [module].txt; for the FAT file system this file is /hcc/src/history/fat.txt.

Documentation

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 will be of the form [module]_[type].[Ext], where the type describes the particular document and the extension would typically be .pdf but could also be in other formats such as .docx or .txt. For the FAT file system the included documents are (1) /hcc/src/doc/fat_api.pdf (2) /hcc/src/doc/fat_mdriver.pdf and (3) /hcc/src/doc/fat_sg.pdf which are respectively the API guide, Media Driver guide and System guide for the FAT file system.

Library Modules

If libraries are included in a project these are placed in the /hcc/lib directory.

Package Information

This directory contains package information files for each of the HCC packages that are 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.

Drivers

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.

Utilities

This directory contains any utility programs provided with the package that aid with development or implementation of that package in a system. Not all packages have associated utilities.

Project

This directory contains an actual project. The format of the project files under this directory is project-specific and will vary as a function of tool-chains and also from project to project. For any particular project there is normally a readme file in the /project/doc sub-directory.