



Firmware Quick Start Guide

Version 1.09

Exported on 05/08/2018

All rights reserved. This document and the associated software are the sole property of HCC Embedded. Reproduction or duplication by any means of any portion of this document without the prior written consent of HCC Embedded is expressly forbidden.

HCC Embedded reserves the right to make changes to this document and to the related software at any time and without notice. The information in this document has been carefully checked for its accuracy; however, HCC Embedded makes no warranty relating to the correctness of this document.

Table of Contents

1	Getting Started.....	3
1.1	Full Project Delivery	3
1.2	New Package Delivery.....	3
1.3	Package Update Delivery.....	5
2	General Development Guidelines	6
2.1	Code Modification	6
2.2	Project Identification	6

1 Getting Started

This document provides a short set of instructions for installing HCC Embedded firmware products. This covers three possible situations:

- **Full Project Delivery** – this is when the complete development environment has been defined (including development board, toolchain, RTOS, HCC components, and so on) and HCC has built a complete tested project to this specification. In this case you will receive a full project.
- **New Package Delivery** – this is when the customer takes a set of HCC products and ports them to their development environment. In this case you will receive a set of new packages for creating your own project. These products will be delivered as packages of code that are wholly independent of compiler or toolchain. Core modules will also be independent of target; the areas of target dependency are clearly separated in the Platform Support Package and these will need to be ported.
- **Package Update Delivery** – this is where the customer already has a working project, either developed by HCC or by the customer, and HCC is providing an update to some of the components used in that project. In this case, you will receive one or more update packages for a project component.

Follow the procedure below for your case.

1.1 Full Project Delivery

If you have had a full project delivered, do the following:

1. Extract the zipped project file provided to your chosen working directory for this project. (This is referred to as <PROJECT_ROOT> below.)
2. Refer to the documents for the products and source tree in <PROJECT_ROOT>/**hcc/doc**.
3. Configure the configuration files in <PROJECT_ROOT>/**hcc/src/config** in accordance with your requirements and after referring to the product manuals.
4. Read the readme file for the project. It is in <PROJECT_ROOT>/**project/doc**.

1.2 New Package Delivery

If you have received a set of new packages to create your project from, do the following:

1. Extract all the supplied zipped package files to your chosen working directory for this project, allowing overwrite of existing files. (This directory is referred to as <PROJECT_ROOT> below.)
2. Read the *HCC Source Tree Guide*. It is vital that you understand this short guide before you build your project.
3. Read the documents for the products; these are in <PROJECT_ROOT>/**hcc/doc**.
4. If it does not already exist, create the directory <PROJECT_ROOT>/**hcc/src/config**.
5. Copy (without allowing existing files to be overwritten) all files from <PROJECT_ROOT>/**hcc/src/config_template** to <PROJECT_ROOT>/**hcc/src/config**.

6. Copy (without allowing existing files to be overwritten) all files from `<PROJECT_ROOT>/hcc/src/psp_template` to `<PROJECT_ROOT>`.
7. Modify the following line in the file `ver_psp.h`, so that it reflects your environment:

```
PSP_ (PROC) _(BOARD) _(RTOS) _(COMPILER)
```

Note: You change this to describe your platform uniquely. This allows you to build your own PSP code for your system, knowing that it is always protected by these safeguards. All target-specific code should be protected by this definition; this ensures the code is only associated with the intended target.

Remove the spaces and brackets from the line and substitute your CPU, BOARD, RTOS and COMPILER. Use upper case only. For example, you might change the line to the following:

```
PSP_STM32F103_MYBOARD_ETASKSYNC_IAR
```

8. Port the Platform Support Package functions in `<PROJECT_ROOT>/hcc/src/psp/target` to your target, as indicated in the product manuals.
9. Configure the OS Abstraction Layer (OAL). Use the following table to configure all the required OAL elements in the file `<PROJECT_ROOT>/hcc/src/config/config_oal.h`. An "x" indicates that an OAL element is required. For example, for TINY just one element is needed.

	USB Host	USB Device	FAT/ SafeFAT	SafeFLASH	SafeFTL	TINY	TCP/IP
OAL_TASK_SUPPORTED	x	x					x
OAL_TASK_GET_ID_SUPPORTED	x	x	x	x			x
OAL_TASK_SLEEP_SUPPORTED	x						x
OAL_MUTEX_SUPPORTED	x	x	x	x	x	x	x
OAL_EVENT_SUPPORTED	x	x					x
OAL_ISR_SUPPORTED	x	x					x

Note: No OAL elements are required if the system is used without an OS and THIN or SuperTHIN is the only file system used.

10. Configure the configuration files in `<PROJECT_ROOT>/hcc/src/config` in accordance with your requirements and after referring to the product manuals.

1.3 Package Update Delivery

If you have received one or more update packages for a project component, do the following:

1. Back up your existing project.
2. For each package, extract the package from the **.zip** file to the project's working directory. (This is referred to as <PROJECT_ROOT> below.) In each case, overwrite the existing package.
3. For any package that has changed, merge the new configuration files under <PROJECT_ROOT>/**hcc/src/config_template** with your existing configuration settings in <PROJECT_ROOT>/**hcc/src/config**.
4. For any PSP package that has changed,, merge the new PSP files under <PROJECT_ROOT>/**hcc/src/psp_template** with your existing PSP files in <PROJECT_ROOT>/**hcc/src/psp**.

Note: There are two easy methods to determine which packages have changed:

- Perform a **diff** on all the files in your backup **/hcc/src/version** directory against the updated directory. If any file shows a difference this means that its package has been updated.
- Perform a **diff** on all the files in your backup **/hcc/history** directory against the updated directory. If any file shows a difference this means that its package has been updated.

For all packages that have been updated, the configuration files and PSP files must be merged with your versions of these files.

2 General Development Guidelines

2.1 Code Modification

You should **not** modify any files in the **hcc** directory, except for the following:

- Files in the **hcc/src/config** directory, to configure your platform.
- Files in the **hcc/src/psp** directory, to meet the requirements of your specific platform.

In order to access all HCC modules, your application only needs to include files from the **hcc/src/api** directory.

2.2 Project Identification

All HCC projects and package deliveries are assigned a unique ID that is the reference for you to use. This is sent to you by email when the project is started by HCC.

This Project ID directly links your delivery with the HCC source code and project management system. It takes the form **10-xx-xxx-n**, where:

- **xx-xxx** is your unique customer identifier.
- **n** is your unique project identifier.

It is important to include this Project ID in all correspondence to enable HCC to provide the most efficient and accurate support.

Note: If you have any problems contact support@hcc-embedded.com, including the Project ID (provided in the delivery) in the Subject line.