

# OAL Template User's Guide

Version 1.00

For use with OS Abstraction Layer (OAL) Template  
versions 1.00 and above

**Date:** 04-Dec-2014 15:33

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

---

System Overview	3
Introduction	3
Feature Check	4
Packages and Documents	4
Packages	4
Documents	4
Source File List	5
Configuration File	5
Source Code	5
Version File	5
PSP Files	6
Configuration Options	7
Integration	8
OS Abstraction Layer (OAL)	8

# 1 System Overview

## 1.1 Introduction

---

This guide is for those who wish to create an HCC Embedded-compatible RTOS abstraction for their embedded systems. This enables all HCC Embedded firmware components to be used seamlessly with the target RTOS.

This template package provides a set of empty functions for you to insert code into. You will only have this if you are implementing your own RTOS within this system. Your RTOS provides the required functions, so you must create "hooks" to call its functions from the HCC abstractions. You use the HCC template to port to the RTOS.

**Note:**

- HCC provides a wide range of OALs for the most commonly used embedded RTOSes. If yours is not included, these OALs may be used as a reference.
- HCC provides a porting service and will provide an OAL for unsupported RTOSes on request.

## 1.2 Feature Check

---

The main features of the OAL template are the following:

- It conforms to the HCC Advanced Embedded Framework.
- It is integrated with the OAL base package.
- It allows almost any RTOS to be used with all HCC products.
- It provides a standard interface for HCC tasks.
- It provides a standard interface for HCC mutexes.
- It provides a standard interface for HCC events.

## 1.3 Packages and Documents

---

### Packages

The table below lists the packages that you need in order to use this module:

Package	Description
<code>hcc_base_doc</code>	This contains the two guides that will help you get started.
<code>oal_base</code>	The OAL base package which you use with the template.
<code>oal_os_template</code>	The OAL template package described by this document.

### Documents

Readers should note the points in the [HCC Documentation Guidelines](#) on the HCC documentation website.

#### HCC Firmware Quick Start Guide

This document describes how to install packages provided by HCC in the target development environment. Also follow the *Quick Start Guide* when HCC provides package updates.

#### HCC Source Tree Guide

This document describes the HCC source tree. It gives an overview of the system to make clear the logic behind its organization.

#### HCC HCC OS Abstraction Layer (Base) User's Guide

This document defines the OAL base system.

#### HCC OAL Template User's Guide

This is this document.

## 2 Source File List

This section describes all the source code files included in the system. These files follow the HCC Embedded standard source tree system, described in the *HCC Source Tree Guide*. All references to file pathnames refer to locations within this standard source tree, not within the package you initially receive.

**Note:** Do not modify any of these files except the configuration file and PSP files.

### 2.1 Configuration File

The file `src/config/config_oal_os.h` contains all the configurable parameters. Configure these as required. For details of these options, see [Configuration Options](#).

### 2.2 Source Code

These are the source code files. **These files should only be modified by HCC.**

File	Description
<code>src/oal/os/oalp_defs.h</code>	Various definitions.
<code>src/oal/os/oalp_event.c</code>	Source code for events.
<code>src/oal/os/oalp_event.h</code>	Header file for events.
<code>src/oal/os/oalp_isr.c</code>	Source code for ISRs.
<code>src/oal/os/oalp_isr.h</code>	Header file for ISRs.
<code>src/oal/os/oalp_mutex.c</code>	Source code for mutexes.
<code>src/oal/os/oalp_mutex.h</code>	Header file for mutexes.
<code>src/oal/os/oalp_task.c</code>	Source code for tasks.
<code>src/oal/os/oalp_task.h</code>	Header file for tasks.

### 2.3 Version File

The file `src/version/ver_oal_os.h` contains the version number of this module. This version number is checked by all modules that use this module to ensure system consistency over upgrades.

## 2.4 PSP Files

---

These files in the directory **src/psp/target/isr** provide functions and elements the core code needs to use, depending on the hardware. Modify these files as required for your hardware.

File	Description
<b>psp_isr.c</b>	ISR functions source code.
<b>psp_isr.h</b>	ISR functions header file.

## 3 Configuration Options

**Note:** Systemwide configuration options which allow you to disable certain functions or sets of functions are set in the base package's configuration file. See the *HCC OS Abstraction Layer (Base) User's Guide* for details.

Set the configuration options in the file `src/config/config_oal_os.h`. This section lists the available configuration options and their default values.

### OAL\_MUTEX\_COUNT

The maximum number of mutexes. This option may not be required if `oal_mutex_t` statically allocates space for the mutex. The default is 16.

### OAL\_EVENT\_COUNT

The maximum number of events. This option may not be required if `oal_event_t` statically allocates space for the event. The default is 16.

### OAL\_HIGHEST\_PRIORITY, OAL\_HIGH\_PRIORITY, OAL\_NORMAL\_PRIORITY, OAL\_LOW\_PRIORITY, OAL\_LOWEST\_PRIORITY

Lower numbers mean a higher priority. By default these are respectively 5, 10, 15, 20, and 25.

### OAL\_EVENT\_FLAG

The event flag to use for user tasks invoking internal functions. For example, this flag is used where one task calls an internal function that needs to wait for an event.

The value of this flag should be greater than 0x80 because lower bits might be used by internal tasks. The default is 0x100.

**Note:** The following options are by default commented out. Activate them if required.

### OAL\_TASK\_COUNT

The maximum number of tasks. Use this if an OS has any parameter that must be predefined in order to create a task. In this case this option can define the number of required entities. The default is 8.

### OAL\_ISR\_COUNT

The maximum number of interrupt sources. This is optional, depending on the OS. For example, if every ISR needs a wrapper function, these have to be prewritten somewhere so you must define this option. The default is 4.

## 4 Integration

This section specifies the elements of this package that need porting, depending on the target environment.

### 4.1 OS Abstraction Layer (OAL)

---

All HCC modules use the OS Abstraction Layer (OAL) that allows the module to run seamlessly with a wide variety of RTOSes, or without an RTOS.

This module requires the following OAL elements:

OAL Resource	Number Required
Tasks	1
Mutexes	1
Events	1
ISRs	1