



OAL Template User Guide

Version 1.30

For use with OS Abstraction Layer (OAL) Template versions 2.02 and above

Exported on 06/07/2018

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	System Overview.....	3
1.1	Introduction	4
1.2	Feature Check	5
1.3	Packages and Documents	6
	Packages.....	6
	Documents	6
1.4	Change History	7
2	Source File List	8
2.1	Configuration File.....	8
2.2	Source Code	8
2.3	Version File	8
2.4	Platform Support Package (PSP) Files.....	9
3	Configuration Options	10
4	Integration.....	12
4.1	OS Abstraction Layer	12

1 System Overview

This chapter contains the fundamental information for this module.

The component sections are as follows:

- [Introduction](#) – describes the main elements of the module.
- [Feature Check](#) – summarizes the main features of the module as bullet points.
- [Packages and Documents](#) – the *Packages* section lists the packages that you need in order to use this module. The *Documents* section lists the relevant user guides.
- [Change History](#) – lists the earlier versions of this manual, giving the software version that each manual describes.

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 Real Time Operating System (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 module are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Integrated with the HCC OS Abstraction Layer (OAL).
- Fully MISRA-compliant.
- Allows almost any RTOS to be used with all HCC products.

1.3 Packages and Documents

Packages

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

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

Documents

For an overview of HCC RTOS software, see [Product Information](#) on the main HCC website.

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 OS Abstraction Layer (Base) User Guide

This document defines the OAL base system.

HCC OAL Template User Guide

This is this document.

1.4 Change History

This section describes past changes to this manual.

- To view or download earlier manuals, see [OAL PDFs](#).
- For the history of changes made to the package code itself, see [History: oal_os_template](#).

The current version of this manual is 1.30. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.30	2018-06-07	2.02	Corrected text on OAL_EVENT_FLAG configuration option. Added note to <i>Platform Support Package (PSP) Files</i> .
1.20	2017-06-28	2.02	New <i>Change History</i> format.
1.10	2016-09-13	2.02	Added <i>Change History</i> section.
1.00	2014-12-04	1.00	First online version.

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 files are in the directory `src/oal/os`. **These files should only be modified by HCC.**

File	Description
<code>oalp_defs.h</code>	Various definitions.
<code>oalp_event.c</code>	Source code for events.
<code>oalp_event.h</code>	Header file for events.
<code>oalp_isr.c</code>	Source code for ISRs.
<code>oalp_isr.h</code>	Header file for ISRs.
<code>oalp_mutex.c</code>	Source code for mutexes.
<code>oalp_mutex.h</code>	Header file for mutexes.
<code>oalp_task.c</code>	Source code for tasks.
<code>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 Platform Support Package (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
psp_isr.c	ISR functions source code.
psp_isr.h	ISR functions header file.

Note: In the package these files are offset to avoid overwriting an existing implementation. Copy them to the root **hcc** directory for use.

3 Configuration Options

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

Note: Set systemwide configuration options in the base package's configuration file; these allow you to disable certain functions or sets of functions. See the [HCC OS Abstraction Layer \(Base\) User Guide](#) for details.

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

OAL_EVENT_FLAG's usage depends on the type of event system an RTOS uses. There are two types:

- Event groups are supported independently of everything else in the system. In this case OAL_EVENT_FLAG does not matter.
- Each event group is directly controlled by a specific task. In this case all HCC stack internal events use the OAL_EVENT_FLAG as the event flag to set on the tasks event group. None of the tasks invoking HCC API calls should use OAL_EVENT_FLAG for signalling an event.

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

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