



OAL Template User Guide

Version 2.00

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

Table of Contents

1. System Overview	3
1.1. Introduction	4
1.2. Feature Check	5
1.3. Packages and Documents	6
1.4. Change History	7
2. Source File List	8
3. Configuration Options	10
4. Integration	11
4.1. OS Abstraction Layer	11
4.2. PSP Porting	11
psp_isr_install	12
psp_isr_delete	13
psp_isr_enable	14
psp_isr_disable	15
psp_int_enable	16
psp_int_disable	17
5. Version	18

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.

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.

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
<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

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 download this manual or a PDF describing an [earlier software version, 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 2.00. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
2.00	2020-02-12	2.02	New template.
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.

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](#).

Source Code

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

File	Description
oalp_defs.h	Various definitions.
oalp_event.c and .h	Event functions source code and header file.
oalp_isr.c and .h	ISR functions source code and header file.
oalp_mutex.c and .h	Mutex functions source code and header file.
oalp_task.c and .h	Task functions source code and header file.

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.

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.

Note:

- These are PSP implementations for the specific microcontroller and board; you may need to modify these to work with a different microcontroller and/or development board. See [PSP Porting](#) for details.
- In the package these files are offset to avoid overwriting an existing implementation. Copy them to the root **hcc** directory for use.

The files are as follows:

File	Description
psp_isr.c	ISR functions source code.
psp_isr.h	ISR functions header file.

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

4.2. PSP Porting

These functions are provided by the Platform Support Package (PSP) to perform various tasks. They are designed for a specific microcontroller and development board. You may need to port them to work with your hardware solution; they are designed to make porting easy.

The package includes samples in the **psp_isr.c** file.

Function	Description
psp_isr_install()	Initializes the ISR.
psp_isr_delete()	Deletes the ISR, releasing the associated resources.
psp_isr_enable()	Enables the ISR.
psp_isr_disable()	Disables the ISR.
psp_int_enable()	Enables global interrupts.
psp_int_disable()	Disables global interrupts.

These functions are described in the following sections.

psp_isr_install

This function is provided by the PSP to initialize the ISR.

Format

```
int psp_isr_install (
    const oal_isr_dsc_t * isr_dsc,
    oal_isr_id_t * isr_id )
```

Arguments

Argument	Description	Type
isr_dsc	The ISR descriptor.	oal_isr_dsc_t *
isr_id	The ISR ID.	oal_isr_id_t *

Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

psp_isr_delete

This function is provided by the PSP to delete the ISR, releasing the associated resources.

Format

```
int psp_isr_delete ( oal_isr_id_t isr_id )
```

Arguments

Argument	Description	Type
isr_id	The ISR ID.	oal_isr_id_t

Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

psp_isr_enable

This function is provided by the PSP to enable the ISR.

Format

```
int psp_isr_enable ( oal_isr_id_t isr_id )
```

Arguments

Argument	Description	Type
isr_id	The ISR ID.	oal_isr_id_t

Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

psp_isr_disable

This function is provided by the PSP to disable the ISR.

Format

```
int psp_isr_disable ( oal_isr_id_t isr_id )
```

Arguments

Argument	Description	Type
isr_id	The ISR ID.	oal_isr_id_t

Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

psp_int_enable

This function is provided by the PSP to enable global interrupts.

Format

```
int psp_int_enable ( void )
```

Arguments

None.

Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

psp_int_disable

This function is provided by the PSP to disable global interrupts.

Format

```
int psp_int_disable ( void )
```

Arguments

None.

Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

5. Version

Version 2.00

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